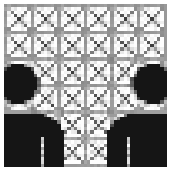
Diplomarbeit

PENETRATIONSTEST

Möglichkeiten und Grenzen

Universität Hamburg



Fachbereich Informatik

Nils Michaelsen

Betreut durch:

Prof. Dr. Klaus Brunnstein

Dr. Hans-Joachim Mück

Durchgesehene Auflage

Hamburg, im Januar 2004

Vorwort zur durchgesehenen Auflage

Kleinere Fehler sowie neue Formulierungen ließen die hier vorliegende durchgesehene Auflage entstehen. Die Diplomarbeit wurde am 28. Oktober 2003 abgegeben. Seitdem wurden die Abschnitte 2.4.2., 2.4.3, 3.4.3., 5.5. und 7.1.3. hinsichtlich ihrer Formulierung überarbeitet. Einfluss auf die Überarbeitung nahm auch das Erscheinen einer Studie des Bundesamtes für Sicherheit in der Informationstechnik zu diesem Thema im November 2003.

Hamburg, im Januar 2004

Nils Michaelsen

Danksagung

An dieser Stelle möchte ich all jenen danken, die mich während der Bearbeitungszeit unterstützt haben. Danken möchte ich insbesondere meinen Betreuern

Prof. Dr. Klaus Brunnstein und
Dr. Hans-Joachim Mück.

Außerdem bedanke ich mich für seelische Beistand und Korrekturhilfen bei Alma Reddel, Simone Schuster und meinen Eltern. Besonderer Dank gilt Jan Menne für die Zusammenarbeit sowie für kritische Hinweise.

Markenzeichen

Die Rechte an den in dieser Arbeit verwendeten Markenzeichen halten deren Eigentümer.

Hinweis für den Leser

Sollten Sie Fragen oder Anregungen haben, oder einen Fehler gefunden haben, können Sie eine Email an dip@nmhh.de schicken.

Inhalt

1. KAPITEL:

EINLEITUNG

1.1. Motivation und Zielsetzung

Der Begriff Penetrationstest wird häufig in Verbindung mit Überprüfungen im Rahmen der IT-Sicherheit verwendet. Beschäftigt man sich mit der Beantwortung von Fragen wie

- „Wie sicher ist mein Netz?“ oder
- „Habe ich meine Firewall gut konfiguriert?“,

ist der Kontakt mit dem Begriff unvermeidlich.

Wer sich mit IT-Sicherheit beschäftigt, meint zu wissen, was der Begriff bedeutet. Doch leider weiß jeder etwas anderes über die Bedeutung und dieses Wissen ist häufig nicht korrekt. Dieser Umstand wird aus der Diskussion auf einer Mailingliste deutlich, deren Inhalt zur eigentlichen Motivation des Themas zu Beginn des Kapitels 3 behandelt wird.

Zudem kennen die meisten IT-Sicherheitsexperten die Sichtweise der Verteidigung, nicht aber die des Angriffs. Motivation, auch diese Seite einmal zu betrachten bietet folgendes Zitat:

> “Knowing your enemy is the key to winning the battle.”
>
> Sun Tzu, The Art of War
> (aus [Cole02:19])

Diese Arbeit soll daher die Bedeutung des Begriffs „Penetrationstest“ klären. Zudem sollen die Möglichkeiten des Penetrationstests gezeigt, wegen der Risiken aber auch die Grenzen offenbart werden. Um einen Einblick in die Sichtweise des Penetrationstests zu erlangen, werden auch praktische Versuche innerhalb von Szenarien durchgeführt. Dabei wird aber auf eine genaue Beschreibung der betrachteten Angriffe verzichtet, da diese Arbeit kein Lehrbuch für Hacker sein soll.

1.2. Vorgehen

Zunächst wird eine gemeinsame Basis der Hintergründe der IT-Sicherheit geschaffen, indem in Kapitel 2 die Grundlagen behandelt werden. Kapitel 3 behandelt die Hintergründe des Penetrationstests an sich. Ein wesentliches Merkmal der Penetration ist die Durchführung von Angriffen, welche in Kapitel 4 näher betrachtet werden. Aus der Vorgehensweise eines Angreifers wird in Kapitel 5 eine Vorgehensweise des Penetrationstests entwickelt. Mittels dieser Vorgehensweise werden Versuche durchgeführt. In Kapitel 6 wird mit der entwickelten Vorgehensweise die Sicherheit eines Webservers betrachtet. Kapitel 7 behandelt den Einsatz des Penetrationstests bei der Überprüfung einer Firewall. In Kapitel 8 wird der Fachbereich Informatik mit den Methoden der Reconnaissance betrachtet. Kapitel 9 bietet eine Schlussbetrachtung, in der die Möglichkeiten und Grenzen sowie die Anwendbarkeit des Penetrationstests diskutiert werden.

Bei der Versuchsdurchführung wird eng mit dem Incident-Response Projekt des Arbeitsbereiches AGN am Fachbereich Informatik der Universität Hamburg zusammengearbeitet. Im Rahmen dieses Projekts schreibt zeitgleich Jan Menne eine Diplomarbeit (vgl. [Menne03]) mit dem Titel „Methoden der Vorfallserkennung und -analyse“, in der die Ergebnisse der Penetrationen verwertet werden. Hierbei soll mit dem von Jan Menne entwickelten Tool „CompareSys“ die durch die Penetration entstandenen Veränderungen an den Systemen erkannt werden.

2. KAPITEL:

GRUNDLAGEN

Eine Arbeit über das Thema Penetrationstest ist im Bereich IT-Sicherheit angesiedelt. Als Grundlage für eine solche Arbeit ist theoretisches Hintergrundwissen notwendig. Es dient zum einen als Grundlage für die Texte ab Kapitel 3, zum anderen besteht das Problem, dass es – wie in anderen wissenschaftlichen Gebieten auch – verschiedene Verständnisweisen für einen Fachbegriff gibt. Daher soll dieser Teil vor allem dazu dienen, ein gemeinsames Verständnis zu erlangen.

2.1. Abhängigkeit von der IT

Um den Schutzbedarf der Informationstechnologie darzustellen, wird in einem einführenden Exkurs die Bedeutung der Informationstechnologie in der Betriebswirtschaft erläutert.

„Aussicht auf einen Mehrwert stellt Motivation für unternehmerisches Handeln dar“ [Haller97:65]. Ist ein solcher subjektiv empfundener und quantifizierbarer Mehrwert geschaffen, so wird von einer Wertschöpfung (engl.: value added) gesprochen, die aus der Erstellung eines Produktes oder einer Dienstleistung entstanden sein kann. Ziel einer Unternehmung ist es, die Wertschöpfung zu steigern und zu erhalten (vgl. [Keuper01:201]).

Die gesamte Wertschöpfung eines Unternehmens kann nach Keuper (vgl. [Keuper01:200]) als die Gesamtheit aller Wertzuwächse der einzelnen Geschäftsprozesse verstanden werden. Geschäftsprozesse sind Abläufe, „die in Wirtschaftsorganisationen durchgeführt werden und dem Betriebszweck dienen“ [Keuper01:33].

Ein Instrument zur Untersuchung des Wertschöpfungsprozesses ist Porters Wertkette, auch Wertschöpfungskette genannt (vgl. [Keuper01:201]), die in Abbildung 1 dargestellt ist. Durch diese Kette kann untersucht werden, wo Wettbewerbsvorteile entstehen, welche die Wertschöpfung und somit das Überleben eines Unternehmens ermöglichen.

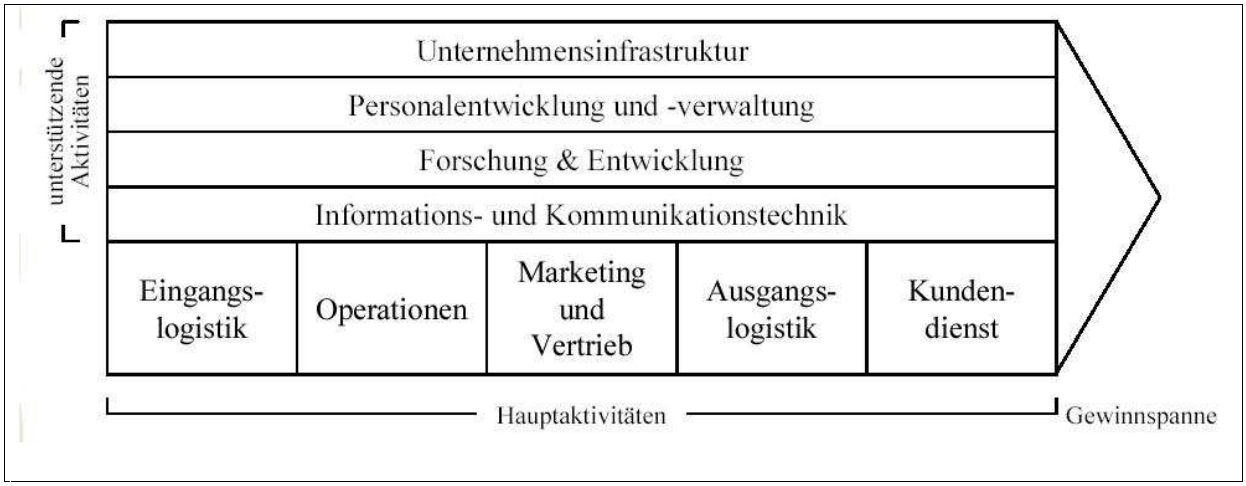


Abbildung 1: Portersche Wertschöpfungskette nach [Keuper02:8]

Aus der Wertschöpfungskette wird ersichtlich, dass die Informationstechnologie, in Abbildung 1 Informations- und Kommunikationstechnik genannt, ein fester Bestandteil der Wertschöpfung geworden ist. So werden immer „mehr Geschäftsprozesse […] auf die IT verlagert oder mit ihr verzahnt“ [GSHB02:Kapitel 1.3]. Auch ist Computer Integrated Manufacturing, kurz CIM (vgl. [Keuper01:329ff.]), ein Schlagwort in der betrieblichen Produktion geworden.

Die zur Wertschöpfung führenden Geschäftsprozesse bauen auf den Werten einer Organisation auf. Für die Werte einer Organisation hat sich der englische Fachbegriff Asset[1] etabliert. Assets wie Hardware und Daten sind Bestandteil der IT in der Wertschöpfungskette und somit ein Wertschöpfungsfaktor.

Für den Fortbestand des Betriebs (engl.: business continuity) ist eine andauernde Wertschöpfung notwendig. Daher müssen die Geschäftsprozesse mit den beteiligten Assets geschützt werden, um ferner die Produktivität des Unternehmens zu sichern.

Ohne Schutz kann es zum teilweisen Stillstand der Produktion kommen. Dadurch können neben finanziellen Schäden durch den Produktionsausfall auch andere Konsequenzen wie die Schädigung des Images eines Unternehmens entstehen. Daraus kann folgen, dass Kunden zur Konkurrenz wechseln oder die Kreditwürdigkeit leidet, wodurch der Fortbestand des Betriebs gefährdet ist.

Der eben dargestellte Sachverhalt gilt nicht nur für Unternehmen, sondern auch für andere Organisationen im institutionalen Sinne. Sie haben Nutznießer (engl.: beneficiaries), die den Bestand der Organisation sicherstellen (vgl. [Pfleeger00:460]). So sind beispielsweise die Studenten die Nutznießer einer Universität, welche die Leistungen für die Studenten aufrechtzuerhalten hat.

Um die Assets schützen zu können, müssen die Gefahren betrachtet werden, denen sie ausgesetzt sind. Einen Überblick über diese Gefahren bietet der folgende Abschnitt.

[1] Assets sind nach [Duden-Oxford90:64] Vermögenswerte.

2.2. Gefahren

Aus den oben genannten wirtschaftlichen Gründen sind Assets vor der Gefahr eines Schadens zu schützen. Die Gefahr geht von den Bedrohungen aus, denen die Assets ausgesetzt sind. Dabei ist nach Brunnstein (vgl. [Brunnstein02:43201]) zwischen zufälligen (engl.: accidential threats) und vorsätzlichen Bedrohungen (engl.: deliberate threats) zu unterscheiden. Während zufällige Bedrohungen durch natürliche Ereignisse wie Blitzeinschlag oder Flut, aber auch durch eine Fehlbedienung eintreten können, werden vorsätzliche Bedrohungen unter anderem durch Hacker oder Malware absichtlich herbeigeführt. Gerade die vorsätzlichen Bedrohungen sind für einen Penetrationstest von Bedeutung und bedürfen daher einer genaueren Betrachtung.

Vorsätzliche Bedrohungen gehen häufig von Hackern aus. Der Begriff „Hacker“ ist in den Medien immer negativ belegt. Ursprünglich wurde der Begriff „hack“ für die elegante Lösung eines technischen Problems verwendet (vgl. [Gröndahl00:40]). In dieser Arbeit soll ein Hacker nach der heutzutage üblichen Sichtweise als eine Person verstanden werden, die Angriffe auf IT-Systeme ausübt. Der Hacker wird dabei nach [Honeynet03] in blackhat und whitehat unterschieden. Den Ursprung hat diese Unterscheidung in den Western-Filmen in den frühen Tagen des Films. Damit der Gute und der Böse auf einem Schwarz-Weiss Bildschirm unterschieden werden konnten, erhielt der Gute einen weißen, der Böse einen schwarzen Cowboy-Hut. Dementsprechend verfolgt ein blackhat Hacker eine böse Intention, während ein whitehat Hacker an der Auffindung von Schwachstellen interessiert ist, um diese danach zu schließen.

Der Angriff wird nach [Howard98:12] als eine Serie von Schritten verstanden, die zu einem unautorisierten Ergebnis führt. Angriffe werden in Kapitel 4 näher betrachtet. Bei der Durchführung eines Angriffs nutzen Angreifer Schwächen (engl.: weaknesses) in einem IT-System aus. Schwächen sind Fehler oder Mängel (engl.: flaw) in der Konzeption, Implementation und Konfiguration eines Systems. Eine weitere Schwäche der IT ist der Mensch, der Fehler durch die Bedienung des Systems herbeiführen kann. Kann eine Schwäche zu einer ungewünschten Aktion führen, deren Ergebnis ein Schaden ist[2], so wird von einer Schwachstelle (engl.: vulnerability) gesprochen.

Die Zahl der Berichte über neu erkannte Schwachstellen ist von Jahr zu Jahr gestiegen, was aus der in Abbildung 2 dargestellten Statistik des CERT®/CC[3] zu erkennen ist. Die Schwachstellen sind schon vor der Erkennung in der Software enthalten. Daher sind sie in potentielle und bekannte Schwachstellen zu differenzieren. Eine potentielle Schwachstelle ist dabei noch nicht entdeckt worden und somit noch nicht bekannt.

[2] Definition beruht auf [Pfleeger00:3] und [Howard98:19]
[3] CERT®/CC an der Carnegie Mellon University in Pittsburgh, Pennsylvania, USA ging aus dem ursprünglichen Computer Emergency Response Team hervor. Da er heutzutage eine große Anzahl von CERTs gibt, hat das CERT®/CC die Koordination aller CERTs übernommen, weshalb es den Titel „Coordination Center“ trägt.

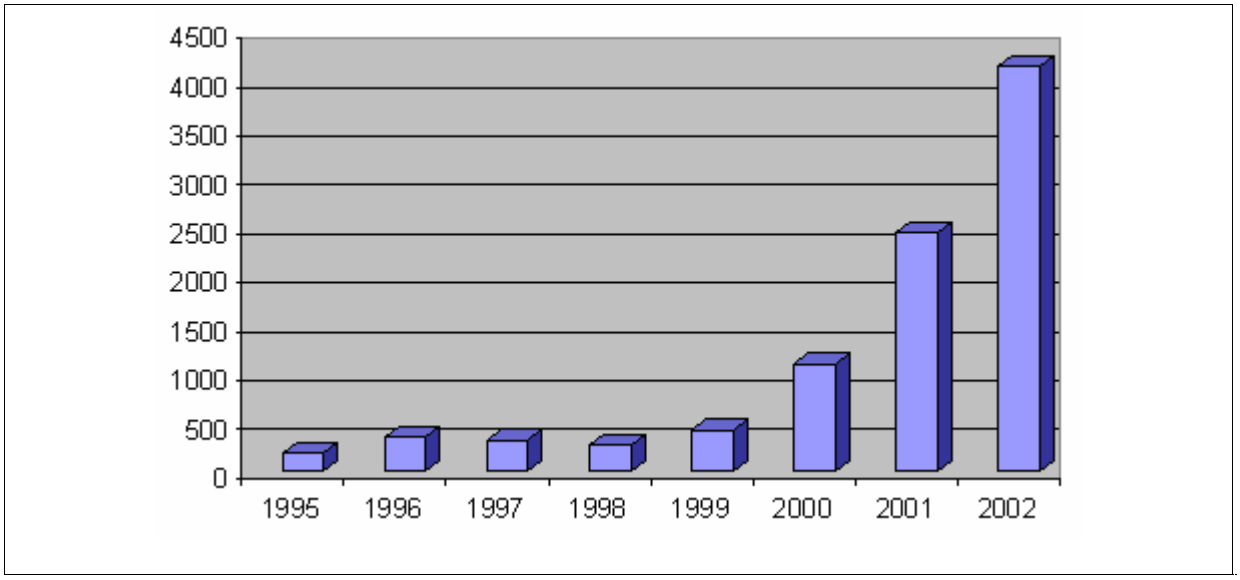


Abbildung 2: Entdeckte Schwachstellen nach Statistiken des CERT®/CC [Cert03]

Die Gefahr der Schwachstelle besteht durch das Einwirken einer Bedrohung auf eine Schwachstelle, das nach [SP800-30:15] in Abhängigkeit von der Art der Bedrohung unterschieden wird. Eine zufällige Bedrohung triggert eine Schwachstelle, während eine absichtliche Bedrohung eine Schwachstelle ausnutzt. Die Beschreibung einer Methode, mit der die Ausnutzung einer Schwachstelle verwirklicht werden kann, wird dabei als Exploit bezeichnet. Die Beschreibung kann dabei in natürlichsprachiger Form oder automatisiert in Form eines Computerprogramms vorliegen.

Neben den Angriffen stellt auch maliziöse Software, kurz Malware, eine große Bedrohung dar. Malware ist nach Brunnstein (vgl. [Brunnstein99:7]) ein Oberbegriff für sämtliche Software, die intentionale Dysfunktionen aufweisen und eine Evidenz gegeben ist, dass die Dysfunktionen zu einer Störung des normalen Verhaltens einer Software führen. Eine Dysfunktion ist eine Funktion, die von der Spezifikation abweicht und damit ein unbeabsichtigtes Merkmal einer Software ist. Hat eine solche Funktion eine schädliche Eigenschaft, so wird sie als Schadfunktion oder „payload“ bezeichnet. Malware, die sich von selbst verbreiten kann, wird häufig als viral bezeichnet. Alle anderen Formen von Malware werden dementsprechend als nicht viral bezeichnet.

Malware tritt in unterschiedlichen Formen auf, wobei hier die am weitesten verbreiteten Formen Virus, Wurm und Trojaner behandelt werden. Ein Virus bezeichnet jegliche Software, die sich mit Hilfe eines Wirtsprogramms, auch Host genannt, reproduzieren kann. Dazu enthält der Virus eine Funktion zur Reproduktion, die Infektion genannt wird, und eine Schadfunktion (vgl. [Brunnstein99:8]). Würmer hingegen verbreiten sich als eigenständiges Programm über ein Netzwerk und sind somit unabhängig von einem Wirtsprogramm. An der Propagation genannten Verbreitung des Wurms sind mindestens zwei Knoten eines Netzwerkes beteiligt (vgl. [Brunnstein99:8]). Zudem ist jeder Wurm dadurch charakterisiert, dass seine Instanzen untereinander kommunizieren können. Trojanische Pferde verfügen nach Brunnstein (vgl. [Brunnstein99:9]) neben nützlichen, dem Benutzer bekannten Funktionen, auch über mindestens eine schadhafte Funktion, welche dem Benutzer verborgen ist (vgl. [Brunnstein99:9]).

Allerdings verschwimmt durch Würmer und Trojaner die Grenze zu den Angriffen. Würmer benutzen zur Propagation Angriffsmethoden, die Schwachstellen ausnutzen. Des Weiteren können Trojaner von einem Hacker bei einem Angriff verwendet werden.

Ist ein Angriff gegen ein IT-System oder die Infektion durch Malware erfolgreich gewesen, so ist ein Vorfall (engl.: incident) entstanden. Unter einem Vorfall wird nach [SP800-3:1] jedes widrige Ereignis verstanden, das einen Aspekt der IT-Sicherheit schädigt. Der Vorfall kann dabei das Ergebnis eines Angriffes oder auch einer Fehlbedienung sein. Nach einem Vorfall sind Daten und Systeme nach [RFC2828:41] kompromittiert (engl.: compromised) worden, wenn sie dem unautorisierten Zugriff ausgesetzt sind.

Das Ausmaß der Gefahr nimmt im Laufe der Zeit enorm zu. So stellt die in Abbildung 3 dargestellte Statistik des CERT®/CC eine exponentielle Zunahme der Vorfälle dar. Nach einer Pressemeldung (vgl. [Heise03]) sei die Anzahl der Angriffe über das Internet im ersten Quartal 2003 um 84% gestiegen.

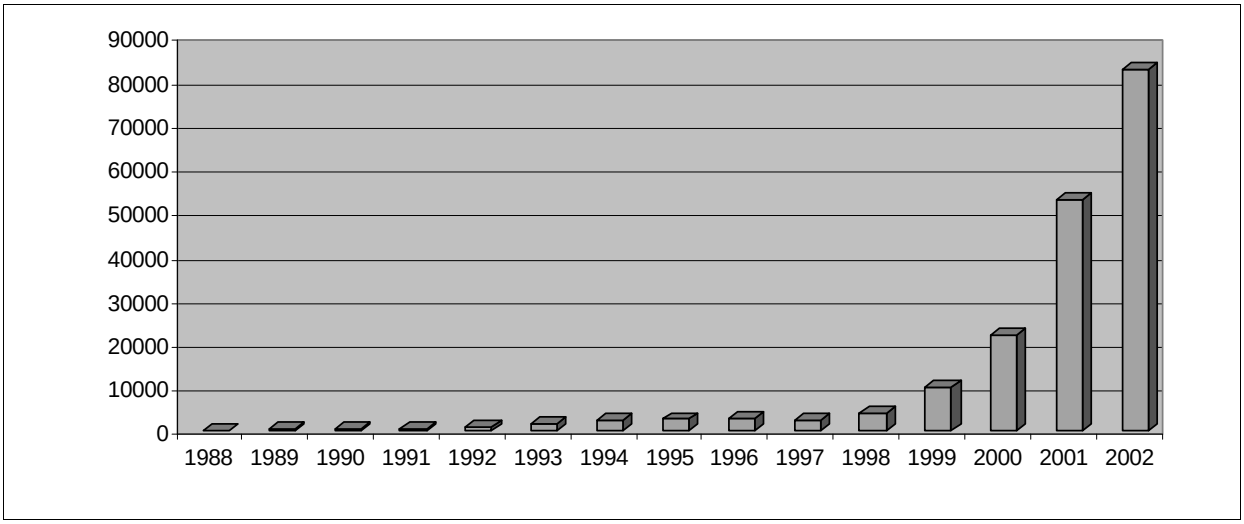


Abbildung 3: Entwicklung der Vorfälle nach Statistiken des CERT®/CC [Cert03]

Gründe für das Ausmaß der Gefahrenlage sind unter anderem folgende Sachverhalte (vgl. [Pfleeger00:449] und [Brunnstein01:1.1a]):

- Die Komplexität, die durch die Gesamtheit aller Zustände eines Systems entsteht, ist weder von dem Hersteller noch von dem Bediener eines Systems überschaubar, wodurch Fehler und weitere Schwachstellen entstehen.
- Es fehlt an Bewusstsein der Problematik.
- Die Administratoren haben eine Übermacht und halten sich für unfehlbar.
- Die Administratoren sind jedoch schlecht ausgebildet.
- Es fehlen Maßnahmen wie Backup, Logging oder die Entfernung unbenutzter Dienste.

Die Gefahrenlage macht deutlich, dass ein Schutz der Assets notwendig ist. Mit diesem Schutz beschäftigt sich die IT-Sicherheit, deren Bedeutung im folgenden Abschnitt geklärt wird.

2.3. IT-Sicherheit

Grundlage für die hier angefertigte Arbeit ist das Thema Sicherheit. Dabei kommt zunächst die Frage auf, was Sicherheit überhaupt bedeutet. Vielfach hört man Argumentationen wie „Kürzlich haben wir eine Firewall installiert. Nun sind wir gegen Angriffe aus dem Internet geschützt“ [Veit99:1]. In [Wiele02] wird in diesem Zusammenhang von einem Traum gesprochen, nach dem IT Sicherheit „[…] lange als ein Zustand der Unverletzlichkeit eines Netzwerks, der mit Produkten wie Firewalls und Virenscannern ein für allemal hergestellt werden sollte“ galt. Fehler in der Software und Konfiguration der Sicherheitsprodukte wie Firewall und Virenscanner führen jedoch dazu, dass ein Schutz durch das reine Installieren der Sicherheitsprodukte nicht gewährleistet werden kann.

Die Bedeutung der IT-Sicherheit wird in den Common Criteria deutlich: „Security is concerned with the protection of assets from threats“. Bedrohungen wirken auf Schwachstellen ein, weshalb auch diese zum Schutz der Assets betrachtet werden müssen. Die Verknüpfung von Bedrohung und Schwachstelle wird dabei als Risiko (vgl. [Krallmann89:35]) bezeichnet. Die IT-Sicherheit ist bestrebt, die Risiken, denen die Assets ausgesetzt sind, auf ein minimales Maß zu reduzieren.

Der englische Begriff „security“ ist dabei abgegrenzt von dem englischen Begriff „safety“, wobei letzterer gewährleisten soll, dass ein System im Sinne der Zuverlässigkeit nur prognostizierbare Ergebnisse liefert (vgl. [Strauss91:40]). Somit kann durch die „safety“ auch gewährleistet werden, dass ein System von sich aus keine Gefahr darstellt[4]. Die IT-Sicherheit im Sinne des Begriffes „security“ hingegen befasst sich mit dem Schutz der Assets gegen die Risiken. Somit lässt sich die „security“ auch vom Datenschutz abgrenzen, der den Schutz der Betroffenen vor den Folgen der Erhebung, Speicherung, Verarbeitung und Weitergabe gewährleisten soll. In dieser Arbeit wird die IT-Sicherheit im Sinne des Begriffes „security“ behandelt.

Ebenso wie im strategischen Management die Erfolgsfaktoren Qualität, Kosten und Zeit im Wettbewerb den angestrebten Erfolg beschreiben, stellen in der IT-Sicherheit die Sicherheitsfaktoren Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität das geforderte Maß an Sicherheit dar. Dieses Schutzmaß stellt die Anforderungen an die Sicherheit dar und kann durch die Kombination der im Folgenden beschriebenen Sicherheitsfaktoren definiert werden.

- Vertraulichkeit

  Vertraulichkeit (engl.: *confidentiality* oder auch *secrecy*) ist genau dann gewährt, wenn Informationen nur berechtigten Personen oder -gruppen zugänglich sind. Somit soll eine Offenlegung (engl.: disclosure) der Informationen gegenüber unbefugten Personen vermieden werden.

[4] So muss ein Atomkraftwerk nicht nur „secure“ sondern auch „safe“ sein, da bei unvorhersehbaren Ergebnissen, wie z. B. das plötzliche Abstellen der Kühlung, eine ungeheure Gefahr für die Bevölkerung von dem Atomkraftwerk ausgeht.

- Integrität

  Die Integrität (engl.: *integrity*) bezeichnet die Freiheit der Information von unberechtigten oder ungewollten Modifikationen. „Every piece of data is as the last authorized modifier left it." ([Schneier00:122])

- Verfügbarkeit

  Unter Verfügbarkeit (engl.: *availability*) wird der Umstand verstanden, dass ein Dienst und die durch ihn bereitgestellten Informationen jederzeit und ohne Verzögerung zugänglich sind (vgl. [Schneier00:122]).

- Authentizität

  Von Authentizität (engl.: *authenticity*) wird genau dann gesprochen, wenn sich die Echtheit der Quelle verifizieren lässt. Im Allgemeinen wird die Echtheit durch die Überprüfung der Quelle einer Nachricht bzw. der Identität des Autors sichergestellt.

In der Fachliteratur werden meist nur die Sicherheitsfaktoren Vertraulichkeit, Integrität und Verfügbarkeit[5] genannt. Diese sind die „traditionellen Pfeiler" (vgl. [Schneier00:122]) der IT-Sicherheit. Die Vertraulichkeit ist die maßgebliche Eigenschaft eines sicheren Systems im Sinne des Orange Books. Obwohl im Orange Book die Integrität bereits definiert ist (vgl. [TCSEC85:107]), hat sie dort keinen Einfluss auf den Vertrauensgrad. Erst im German Green Book des BSI werden Integrität (Klasse F6 und F8) und die Verfügbarkeit (Klasse F7) in die Ergebnisse der Evaluation mit einbezogen. Authentizität ist in der heutigen IT-Sicherheit besonders in der Verbindung mit E-Business und E-Goverment notwendig, weshalb sie in dieser Arbeit die traditionellen Faktoren ergänzt.

IT-Sicherheit kann allerdings kein Zustand sein. Jeder Bedrohung kann zwar durch eine Maßnahme begegnet werden, aber „diese spezielle Sicherheitsmaßnahme [wird] wiederum Kräfte auf den Plan rufen, die alles daran setzen werden [...] diese zu «knacken»" [Lessing98:3] So entstehen stetig neue Bedrohungen während gleichzeitig immer neue Schwachstellen identifiziert werden. IT-Sicherheit muss somit ein Prozess sein, der sich der ständig ändernden Risikolage anpasst. Dieser Prozess soll das geforderte Schutzmaß bewahren. Die gewonnenen Erkenntnisse sollen nun in einer Definition festgehalten werden.

DEFINITION

Sicherheit ist ein Prozess, der die Risiken minimiert, denen die Assets ausgesetzt sind. Das Minimum der Risiken ist durch das angestrebte Schutzmaß gegeben. Ziel des Prozesses ist, das durch die Sicherheitsfaktoren gebildete Schutzmaß zu wahren.

Vollständige Sicherheit gibt es nicht, da manche Risiken nicht minimiert werden können. Mit dieser Tatsache beschäftigt sich das Vertrauen in Abschnitt 2.5. Zudem muss auch bedacht werden, dass die Gegenmaßnahmen Fehler enthalten und damit selbst zu einem Risiko werden können. In der IT-Sicherheit gilt es also einen Prozess zu etablieren, mit denen die Maßnahmen zur Begegnung des Risikos geplant, etabliert und kontrolliert werden können. Dieser als Sicherheitsmanagement bezeichnete Prozess wird im Folgenden beschrieben.

[5] Im Englischen: Confidentiality, Integrity, Availability; daher häufig mit CIA abgekürzt

2.4. Sicherheitsmanagement

Zentrales Wesen des Sicherheitsmanagements ist die Erstellung, Umsetzung und Weiterentwicklung (vgl. [Nedon00:9]) eines Sicherheitskonzeptes, was der „Ausgangspunkt jeglicher Aktivitäten im Bereich der IT-Sicherheit“ [Nedon00:10] sein muss. Dabei ist das Sicherheitsmanagement

- im *funktionalen Sinne* „eine Managementaufgabe, die die Erkennung, […] Erfassung , sowie die Bewältigung von Risiken im IT-Bereich umfasst“ (vgl. [Oppliger97:21]).

- im *institutionalen Sinne* eine Bezeichnung für eine „Personengruppe, die für den IT-Sicherheitsprozess innerhalb einer Organisation verantwortlich ist“ (vgl. [GSHB02:Kapitel 3.0]).

In dieser Arbeit soll im Folgenden der Begriff Sicherheitsmanagement im funktionalen Sinne gemeint sein, sofern er ohne Anmerkung gebraucht wird.

Das Sicherheitsmanagement soll dabei bewirken, dass die Maßnahmen des Sicherheitskonzeptes effektiv und effizient sind. Die Effektivität, die im Allgemeinen den Wirkungsgrad beschreibt, zielt darauf ab, dass die zur Risikoreduktion gewählten Maßnahmen nicht lückenhaft und unvollständig sind (vgl. [Strauss91:22]). Die Effizienz hingegen beschreibt den Mitteleinsatz. Sie soll erreichen, dass der Aufwand im Sinne von eingesetzten Mitteln für die Sicherung dem möglichen Schaden entspricht und kein wertloser Überschutz entsteht, der dem Betrieb keinen Nutzen bringt.

Bevor ein Sicherheitsprozess initiiert werden kann, sind nach Krallmann (vgl. [Krallmann89:19]) folgende Voraussetzungen zu treffen:

1. Motivation des Managements
2. Abgrenzung von Aufgaben und Kompetenzen
3. Festlegung von Verantwortlichkeiten zur Bildung eines Sicherheitsteams

Richtlinien zur Bildung eines Sicherheitsteams finden sich in [BS7799-1_99:Kapitel 4]. Während der Abgrenzung der Aufgaben werden auch die Ziele festgelegt, die der Sicherheitsprozess erreichen soll. Der Sicherheitsprozess selbst besteht aus den folgenden Phasen:

1. *Risikoerkennungsphase:*

   Durchführung von Wertermittlung, Bedrohungsanalyse, Schwachstellenanalyse

2. *Risikobewertungsphase:*

   Subjektive Bewertung des Risikos

3. *Planungs- und Entscheidungsphase:*

   Durch Planung und Auswahl von Maßnahmen wird das Risiko bewältigt. Dabei muss auch eine Restrisikoanalyse erfolgen.

4. *Realisationsphase:*

   Festschreibung und Implementation der ausgewählten Maßnahmen

5. *Kontrollphase:*

   permanente Kontrolle durch Controlling und Revision

Die Phasen 1 und 2 werden dabei als Risikoanalyse, die Phasen 3-5 werden als Risikomanagement[6] bezeichnet. Im Folgenden sollen die Phasen des Sicherheitsprozesses, dessen Durchführung der Zweck des Sicherheitsmanagements ist, näher erläutert werden.

2.4.1. Risikoerkennungsphase

Eine Risikoanalyse soll u. a. folgende Fragen beantworten (in Anlehnung an [Krallmann89:22]):

- In wie weit bestehen Abhängigkeiten von der IT?
- Welche Betriebsgeheimnisse sind auf IT-Anlagen gespeichert?
- Welche Bedrohungen gibt es?
- Wo sind die Assets verwundbar?

Zum Erreichen von Sicherheit müssen die Risiken für die Assets minimiert werden. Dazu werden zunächst im Rahmen einer Risikoanalyse alle Assets inventarisiert. Da ein Risiko dort entsteht, wo Bedrohung und Schwachstelle zusammentreffen (vgl. [Krallmann89:35]), sind auch die Bedrohungen und Schwachstellen der Assets zu analysieren.

Die Risikoanalyse kann informal durchgeführt werden (vgl. [Nedon00:11f]). Dabei werden Experten zu Rate gezogen, die ihnen bekannte Bedrohungen, Schwachstellen sowie die daraus resultierenden Risiken aufzeigen und Lösungsmöglichkeiten darstellen.

In einer detaillierten Risikoanalyse werden zunächst die Assets analysiert und der Schutzbedarf festgestellt. In den Internationalen Accounting Standards sind Assets Vermögensgegenstände, von denen ein Nutzen zu erwarten ist (vgl. [Hahn03:Kapitel 8]). Assets können in physikalische und nicht physikalische Werte unterteilt werden (vgl. [Brunnstein02:43102]), andere Autoren unterscheiden sie auch in tangierbar und untangierbar (vgl. [Garfinkel96:28]). Die Tabelle 1 enthält Beispiele für Assets, die in Verbindung mit der Informationstechnologie stehen:

[6] Manche Autoren bezeichnen das gesamte Sicherheitsmanagement als Risikomanagement (vgl. [SP800-30:4]).

Physikalische Assets	Nicht physikalische Assets
- Gebäude - Fuhrpark - Maschinen - I&K-Anlagen - Versorgungseinrichtungen (Stromleitungen, Klimaanlage) - Ressourcen (im Sinne von Hilfs- und Betriebsmittel)	- Informationen und Daten (Dateien, Datenbanken, Dokumentationen, Betriebshandbücher etc.) - Software - Dienstleistungen (IT- und Versorgungsdienstleistungen

Tabelle 1: Beispiele für Assets

Oberstes Ziel ist dabei der Schutz der Geschäftsprozesse (vgl. [Barman02:11]), um damit die Business Continuity zu gewährleisten. Somit werden im Rahmen einer Prozessanalyse[7] (vgl. [Keuper01:218ff.]), die sich aus Prozessidentifikation und -dekomposition zusammensetzt, die Prozesse und die an ihnen beteiligten Assets identifiziert. Diese Methode erfordert einen sehr hohen Aufwand. Ist der Aufwand nicht tragbar, so können durch eine Strukturanalyse (vgl. [GSHB02:Kapitel 2.1]) die beteiligten Systeme und die auf ihnen gespeicherten Daten ermittelt werden.

Die identifizierten Assets werden in einem Asset-Inventar zusammengestellt. Zur Übersicht können gleichartige Assets auch zu Gruppen zusammengefasst werden (vgl. [Yazar02:3]). Werden nicht alle Assets identifiziert, so besteht die Gefahr, dass eine Schädigung eines unbeachteten Assets zu Konsequenzen in Form von Betriebsausfällen oder Imageverlust führen kann. Daneben muss ein Asset jederzeit der Kontrolle seines Besitzers unterliegen, was eine ständige Pflege des Assets-Inventars voraussetzt. Somit kann auch einem „Missing Server“[8] begegnet werden.

Im folgenden Schritt werden alle Bedrohungen identifiziert, denen die im Inventar enthaltenen Assets ausgesetzt sind. Eine Bedrohung wird als die Einwirkung verstanden, deren Resultat die Schädigung eines Assets ist (vgl. [Krallmann89:32]). Sie kann dabei in zufällige Bedrohung (engl.: accidential threat) und vorsätzliche Bedrohung (engl.: deliberate threat) unterschieden werden. Tabelle 2 zeigt mögliche Bedrohungen auf.

[7] Die Prozessanalyse ist Bestandteil einer Prozessorganisation, welche eine Unternehmensstruktur an die Prozesse anpasst, oder des Business Process Reengineering, das ein Ansatz der radikalen Reorganisation ist (vgl. [Keuper01:242]).

[8] In einigen Organisationen ist es bereits vorgekommen, dass der Standort eines Servers nicht bekannt war. Dies birgt die Gefahr, dass sich ein Angreifer unbemerkt physikalischen Zugriff verschaffen und das System kompromittieren kann.

Zufällige Bedrohungen *Accidental threats*	Vorsätzliche Bedrohungen *Deliberate Threats*
Höhere Gewalt Wasserschaden, Naturkatastrophen, Feuer, Unfall von Personen Menschliches Versagen Unbeabsichtigte Fehler von Benutzern, Administratoren, Servicepersonal Technisches Versagen Fehlfunktionen von Hardware, Software, Speichermedien; Stromausfall	System Infiltration Versuch des Zugriffs auf Daten mit dem Ziel der - Modifikation - Zerstörung - Enthüllung - Störung der Verfügbarkeit der Daten Missbrauch Ausnutzung spezieller Rechte z.B. für private Interessen, die zur Störung der eigentlichen Tätigkeiten führt; auch in Kombination mit System Infiltration Betrug z. B. Unterschlagung von Geldern Diebstahl Physikalisch, nicht physikalisch (z.B. Belauschen) Absichtliche Zerstörung Vandalismus, direkte physikalische Zerstörung, Brandstiftung

Tabelle 2: Bedrohung in Anlehnung an [Brunnstein02:43201]

Angriffe (vgl. Kapitel 4) und Malware sind Beispiele für die „System Infiltration“ und daher eine vorsätzliche Bedrohung.

Bedrohungen wirken auf Schwachstellen ein. Eine zufällige Bedrohung triggert eine Schwachstelle (vgl. [SP800-30:15]), während eine absichtliche Bedrohung eine Schwachstelle ausnutzt. Daher müssen neben den Bedrohungen auch die Schwachstellen in einer IT-Umgebung analysiert werden.

Die Schwachstellenanalyse verfolgt dabei das Ziel, *alle* Schwachstellen zu identifizieren (vgl. [Krallmann89:35]), die im Kontext des laufenden Betriebs der bestehenden IT-Umgebung vorhanden sind, bevor eine Bedrohung auf eine Schwachstelle einwirken kann.

Um Schwachstellen erkennen zu können, werden Informationen über mögliche Schwachstellen für alle im Inventar verzeichneten Assets gesammelt. Quellen für solche Informationen sind

- Herstellerinformationen über entdeckte Schwachstellen in deren Produkten.
- Advisories[9] und Schwachstellen Informationen aus Archiven wie z.B. CERT®/CC[10], BugTraq[11] von SecurityFocus oder Xforce[12] von Internet Security Systems. Sinnvoll ist dabei auch die Nutzung des Meta-Archives ICAT[13] des National Institute of Standards and Technology, kurz NIST.
- Informationen aus Fachliteratur, um sich über generelle Fehler und insbesondere Konfigurationsfehler zu informieren.
- Diskussionen in Foren und Maillinglisten.
- Ergebnisse vergangener Risikoanalysen, Kontrollen oder Revisionen (vgl. [SP800-30:16f], da dort gefundene Schwachstellen nicht immer behoben wurden.
- Ergebnisse von Evaluationen, wie z.B. die Common Criteria [CC99].

Mit Hilfe von Interviews und Netzplänen kann die bisherige Umsetzung bereits erkannter Maßnahmen betrachtet werden, um dabei vorhandene organisatorische Schwachstellen zu erkennen (vgl. [SP800-30:19]). Dabei kann auch fehlende Effektivität der Maßnahmen eine Schwachstelle des Gesamtkonzeptes sein. Mittels funktionalem Test und dem Penetrationstest können bereits an dieser Stelle Schwächen bei der Implementation von Maßnahmen erkannt werden.

Das Risiko entsteht dort, wo Bedrohung und Schwachstelle zusammentreffen. Da sowohl Betrohung als auch Schwachstelle zu einem Schaden führen, kann das Risiko als die Möglichkeit eines Schadens angesehen werden. Durch die Verknüpfung von Bedrohung und Schwachstelle kann betrachtet werden, welchen möglichen Schäden und damit welchen Risiken die inventarisierten Assets ausgesetzt sind. Damit ist die Risikoerkennungsphase abgeschlossen.

2.4.2. Risikobewertungsphase

Ist das Risiko erkannt, erfolgt eine Bewertung, wodurch ermittelt werden soll,

- welche Bereitschaft besteht, einen Schaden zu tragen,
- ob der Mitteleinsatz für eine Maßnahme adäquat ist und
- mit welcher Priorität eine Maßnahme realisiert werden soll.

Die Priorisierung der Maßnahmenrealisation dient vor allem der Beschleunigung des Sicherheitsmanagments durch die gleichzeitige Durchführung mehrere Phasen. So kann eine vollständige Risikoanalyse viel Zeit in Anspruch nehmen. Durch die Priorisierung soll für große Risiken das Risikomanagement durchlaufen werden, während kleinere Risiken noch analysiert werden. Diese Vorgehensweise soll das Sicherheitsmangement beschleunigen, um große Schäden zu vermeiden. Dazu ist zunächst der Schaden, der in englischsprachigen

[9] Advisories sind Empfehlungen, die befolgt werden sollen.
[10] http://www.cert.org/
[11] http://www.securityfocus.com/bid
[12] http://www.iss.net/security_center/
[13] http://icat.nist.gov

Dokumenten häufig als „impact“ bezeichnet wird, zu betrachten. Beispiele für die aus den Risiken resultierenden Schäden sind

- finanzieller Schaden
- Ausfall des Produktionsprozesses
- Kosten für Wiederherstellung (engl.: Recovery)
- Imageverluste (engl.: public embarrassment)
- Vertraulichkeitsverlust
- Verlust von kundenseitigem Vertrauen

Um einen möglichen Schaden genauer analysieren zu können, sind Baumstrukturen entwickelt worden. Einen Zergliederungsprozess verfolgt Krallmann mit Hilfe der Fehlerbaummethode (vgl. [Krallmann89:34f.]). Hierbei werden Funktionen in einzelne Teilfunktionen zergliedert, um das Ergebnis besser analysieren zu können. Bei sorgfältiger Vorgehensweise soll durch den Fehlerbaum sichergestellt werden, dass keine Ausfallart bei der Analyse ausgelassen werden kann. Auch Schneier (vgl. [Schneier00:318ff.]) entwickelte eine Baummethode. Bei seinen Angriffsbäumen (vgl. Abbildung 4) stellt die Wurzel das Ziel des Angriffes dar, das über die Blätter erreicht werden kann. Hierbei können auch die Kosten des Eintretens eines Schadens genauer analysiert werden.

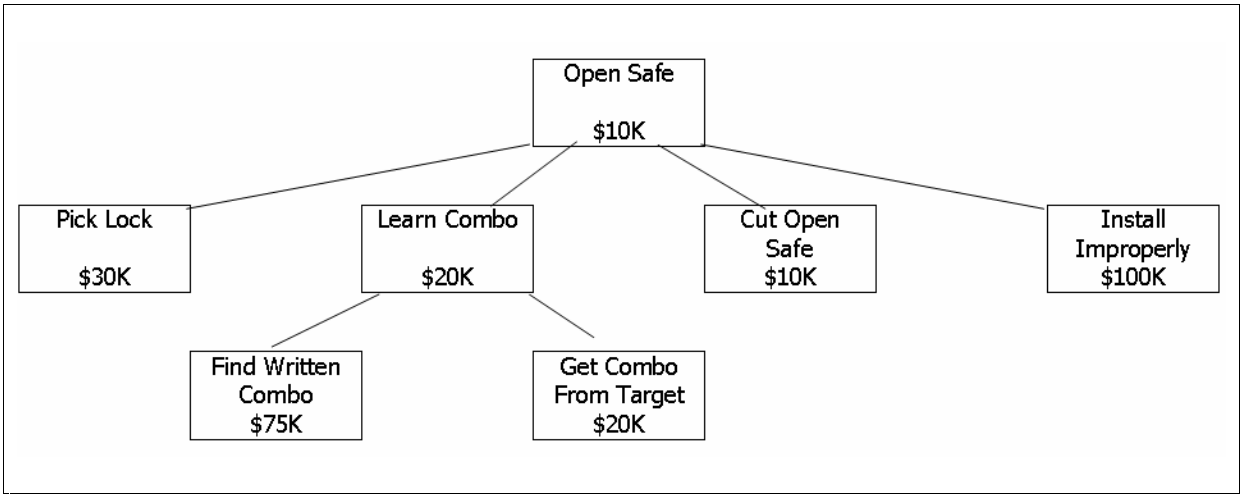


Abbildung 4: Angriffsbaum nach Schneier (vgl. [Schneier00:322])

Die Kosten des Eintretens eines Schadens ist ein in Geldeinheiten bewerteter Ausdruck der Schadenshöhe. Er ergibt sich nach Brunnstein (vgl. [Brunnstein02:43407]) aus den Kosten für die Wiederherstellung der IT, Kosten für die Daten-Recovery, dem erhöhten Personalaufwand und dem Umsatzverlust. So kann das Management zu einer Abschätzung des zu erwarteten Verlusts gelangen.

Die Schadenshöhe wird der Wahrscheinlichkeit des Eintretens eines Schadens (engl.: impact condition) gegenübergestellt. Sie ergibt sich nach [SP800-30:19] aus

- den Motiven und Fähigkeiten möglicher Täter[14],
- der Beschaffenheit einer Schwachstelle,
- der Existenz und Effektivität möglicher Gegenmaßnahmen,
- externe Dienstleister[15] und
- Vorfälle vergangener Jahre.

Bei der Betrachtung von vorfällen können ähnlich wie bei der Vorhersage von Aktienkursen Charts erstellt werden, mit denen Charts die Wahrscheinlichkeit des Eintretens eines Schadens abgeschätzt werden kann. Da häufig ausreichende statistische Informationen fehlen, gestaltet sich die Vorhersage besonders bei vorsätzlichen Bedrohungen als schwierig und ist daher fehleranfällig, so dass ein Risiko auf Grund falscher Daten als zu niedrig angesehen werden kann.

Sind die Informationen über mögliche Schäden und wahrscheinlichkeiten gewonnen, so ergibt sich das quantifizierte Risiko aus der Multiplikation voin Schaden und Wahrscheinlichkeit. Diese Rechnung wird durch die folgende Formel zur Bestimmung des Risikos detailliert dargestellt:

$$R = (C_r + C_{na} + C_a + C_{oi} + C_{in}) \bullet W \bullet P,$$

mit

R = Risiko [GE[16]]
C_r = Kosten für die Wiederherstellung (engl.: Recovery) des ursprünglichen Zustandes [GE]
C_{na} = Kosten für die Nicht-Verfügbarkeit (engl.: non-availability) von Anlagen, etc. [GE]
C_a = Kosten durch Angleichung (engl.: adaption) an Anforderungen [GE]
C_{oi} = Kosten für den Informationsabfluss (engl.: outflow of information) [GE]
C_{in} = Kosten durch Strafen oder Versicherungen (engl.: insurance) [GE]
W = Gewichtung (engl.: weigth) immaterieller Werte, z. B. Verlust öffentlichen Ansehens
P = Wahrscheinlichkeit (engl.: probability) des Eintritts des Schadens

Dieser Risikowert R gibt die in Geldeinheiten ausgedrückte jährliche Verlusterwartung (engl.: Annual Lost Exposure), kurz ALE, an. Auf Grund der genauen Berechnung des Risikowertes wird der eben erläuterte Ansatz als quantitativer Ansatz der Risikobewertung bezeichnet.

Der quantitative Bewertungsansatz hat neben dem hohen Aufwand und der Fehleranfälligkeit der Wahrscheinlichkeitsbestimmung auch den Nachteil, dass ein großer Schaden mit sehr geringer Wahrscheinlichkeit als geringes Risiko eingestuft werden kann. Um diesem Problem zu entgehen, kann die semi-quantitative Bewertung verwendet werden, bei der statt genauer Zahlen nur Klassifikationen wie gering, mittel, hoch verwendet werden. Dabei kann statt der ALE-Matrix das in Abbildung 5 dargestellte Dreieck verwendet werden. Hierbei wird davon

[14] Mögliche Täter, ihre Motive und Fähigkeiten werden von manchen Autoren als Bedrohung bezeichnet. In dieser Arbeit interessieren nur die möglichen Einwirkungen aller Täter, die zu einem Schaden führen können.
[15] Solche Dienstleister werden „Rating-Agenturen“ genannt. Sie helfen auch bei der bestimmung des möglichen Schadens.
[16] GE = Geldeinheiten

ausgegangen (vgl. [Strauss91:95f.]), dass die Eintrittswahrscheinlichkeit von Bedrohungen mit hohem Ausmaß schwer vorhersehbar sind, während geringe Bedrohungen häufig auftreten und somit gut vorhersehbar sind. Einem hohen Risiko soll dabei unabhängig von der Wahrscheinlichkeit begegnet werden.

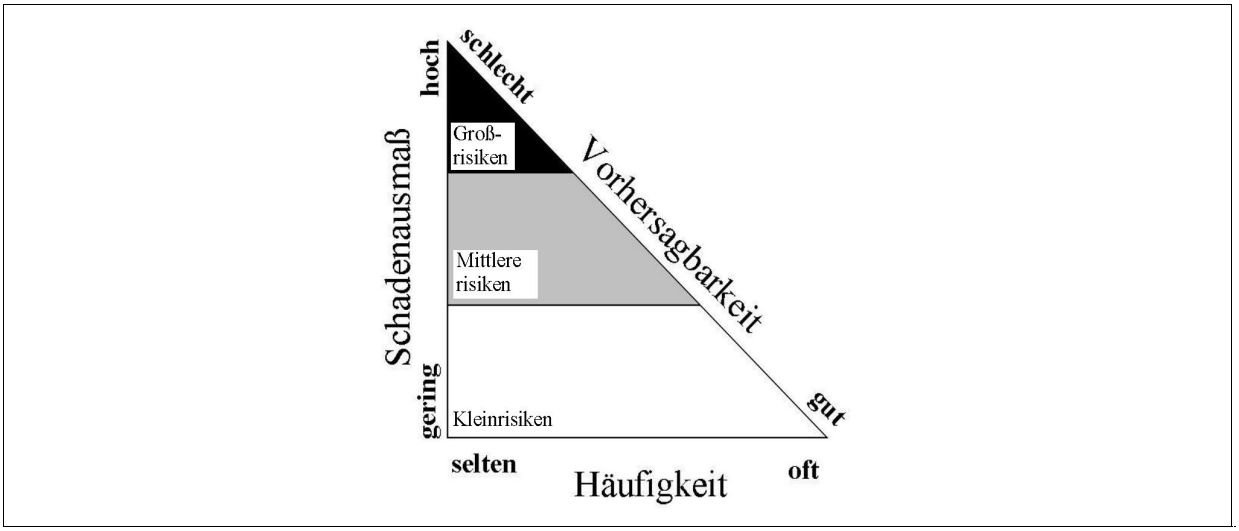


Abbildung 5: Risikoklassen der semi-quantitativen Risikoanalyse (aus [Strauss91:99])

Neben quantitativen und semi-quantitativen Ansatz existiert auch noch der qualitative Ansatz der Risikobewertung. Hierbei wird zunächst jedem Asset ein Schutzmaß, das sich aus dem Maß der geforderten Vertraulichkeit, Integrität und Verfügbarkeit herleitet, zugewiesen. Ist eine solche Bewertung noch nicht erfolgt, so ist sie an dieser Stelle nachzuholen[17]. Die Feststellung des Schutzmaßes ist abhängig von der Betrachtungsweise des Besitzers eines Assets. Daher ist das geforderte Schutzmaß durch Interviews mit den Besitzern zu bestimmen.

Ein Risiko schädigt das Schutzmaß. Um das Risiko qualitativ bewerten zu können, werden Assets, Bedrohung und Schwachstelle Werte aus einem begrenzten Wertebereich zugeordnet. Mögliche Wertebereiche sind beispielsweise (hoch, mittel, niedrig) oder eine Skala von 1 bis n. In der Risikoanalysesoftware CRAMM (vgl. [Yazar02:3]) werden Bedrohungen mit Werten einer Skala von eins bis sieben belegt, welche die Relation angeben, in der sie das Schutzmaß gefährden. Den Schwachstellen werden Werte zwischen eins und drei zugeordnet.

Das Risiko ist bei der qualitativen Risikoanalyse durch die folgende Formel gegeben, wobei der Wert des Assets den Schaden und die Verknüpfung von Bedrohung und Schwachstelle die Wahrscheinlichkeit ausdrücken (vgl. [Yazar02:4]):

$$\text{Risiko} = \text{Asset} \bullet (\text{Bedrohung} + \text{Schwachstelle})$$

Sämtliche Bewertungsmethoden sind subjektiv und hängen daher von der Betrachtungsweise der Asset-Besitzer ab. Die Bewertung bildet die Grundlage für die Auswahl der Gegenmaßnahmen in der nächsten Phase.

[17] Manche Autoren führen eine solche Bewertung bereits während der Erstellung des Asset-Inventars durch.

2.4.3. Planungs- und Entscheidungsphase

Nachdem die Risiken erkannt und bewertet wurden, wird ein Risikomanagement durchgeführt, das nach Brunnstein (vgl. [Brunnstein02:43102]) als „die Bestimmung möglicher Gegenmaßnahmen zum Zwecke der Risikominderung“ definiert ist. Zu dem Risikomanagement gehört die Risikobewältigung, bei der mögliche Maßnahmen geplant werden, eine Realisationsphase, in der die gewählten Maßnahmen implementiert werden, und eine den gesamten Sicherheitsprozess überwachenden Kontrollphase.

Die Planungs- und Entscheidungsphase ist hierbei der Teil der Risikobewältigung, da hier mögliche Gegenmaßnahmen geplant werden und durch die Entscheidung eine Auswahl der zu realisierenden Maßnahmen getroffen wird. Die ausgewählten Maßnahmen bilden dabei ein Maßnahmenportfolio.

Ist das Risiko bewertet worden, ist zu entscheiden, ob das bestehende Risiko akzeptiert werden kann. Ist dies nicht der Fall, besteht die Möglichkeit, das Risiko zu reduzieren. Das nach der Reduktion bestehende Restrisiko kann akzeptiert und überwälzt oder selbst getragen werden. Das entstandene, akzeptierbare Restrisiko ist das minimale Risiko, welches das Ziel der Risikobewältigung ist. Ist das Restrisiko jedoch nicht akzeptierbar, muss das Risiko vermieden werden, was bedeutet, auf die Funktionalität zu verzichten, aus der das Risiko entsteht.

Eine Risikoüberwälzung beinhaltet alle Formen der Versicherung (vgl. [Strauss91:114]). Wird ein Risiko selbst getragen, entsteht ein Wagnis, dessen potentieller Schaden durch finanzielle Reserven abgegolten werden muss.

Reduzierende Maßnahmen sind nach Strauss (vgl. [Strauss91:74]) „Geräte, Instrumente, Mittel, Abläufe, Standards oder Techniken zur Verhinderung und Entdeckung von Schadensfällen“. Sie können präventiver, erkennender und reaktiver[18] Natur sein. Beispiele für Maßnahmen sind das Einspielen von Patches, das Filtern von Emails, die Zugangskontrolle durch Passwörter und auch die Schulung der Benutzer und Administratoren[19].

Erstes Kriterium der Auswahl der Maßnahmen ist zunächst der Nutzen, den eine Maßnahme bringt. Ist das Risiko qualitativ bewertet worden, so werden die Risiken nach der Schwere geordnet. Dabei werden zunächst die größten Risiken auf ein tolerierbares Maß reduziert. Zur Reduktion des Risikos wird entweder der Bedrohung oder der Schwachstelle entgegengewirkt (siehe Abbildung 6). Der Nutzen einer Maßnahme ist dementsprechend der Wert, um den die Verlusterwartung gesenkt wird. Dieser Ansatz kann aber nicht die Effizienz betrachten.

[18] Durch die Reaktion auf einen Schadensfall können weitere Folgeschäden vermieden werden
[19] Schulungen mindern das Risiko einer Fehlbedienung

Abbildung 6: Reduktion der Risiken durch Bekämpfung der Ursachen

Bei dem quantitativen Ansatz werden die erwarteten Schäden in einer ALE-Matrix logarithmisch den Wahrscheinlichkeiten gegenübergestellt. Eine solche Matrix ist in Abbildung 7 dargestellt. Jedes Risiko kann durch eine Zelle in der Matrix quantifiziert werden. Die Verminderung des Risikos geschieht durch eine Auswahl von Gegenmaßnahmen, mit denen der mögliche Schaden sowie die Wahrscheinlichkeit gesenkt werden können. Das Risiko rückt anhand der in Abbildung 7 dargestellten Pfeile nach innen.

Kosten in € \ Wahrscheinlichkeit	Einmal in 30 Jahren	Einmal in 3 Jahren[14]	Einmal in 100 Tagen	Einmal in 10 Tagen	1 mal pro Tag	10 mal pro Tag	100 mal pro Tag
10			30	300	3.000	30.000	300.000
100		30	300	3.000	30.000	300.000	3.000.000
1.000	30	300	3.000	30.000	300.000	3.000.000	
10.000	300	3.000	30.000	300.000	3.000.000		
1.000.000	3.000	30.000	300.000	3.000.000			
10.000.000	30.000	300.000	3.000.000				

Abbildung 7: Risikoreduktion bei quantitativer Risikobewertung (nach [Brunnstein02:43407])

Der Wahrscheinlichkeit wird dabei durch vermindernde Maßnahmen entgegengewirkt. Dem Schaden kann auch durch die Risikoüberwälzung entgegengewirkt werden. So können die Kosten eines Brandschadens sowohl durch eine Sprinkleranlage als auch durch eine Feuerschutzversicherung gemindert werden.

Damit ein Maßnahmenportfolio effizient ist, müssen die Kosten einer Maßnahme kleiner des Nutzens sein. Der Nutzen ist dabei durch die Kosten im Falle eines Schadens bestimmt. So kommt der Betrieb beispielsweise im Falle eines Stromausfalls durch den Ausfall einer Komponente zum Erliegen, wodurch die betroffenen Mitarbeiter ihren Aufgaben nicht nachkommen können, so dass den Lohnkosten keine Leistungen gegenüber stehen. Zudem müssen die Mitarbeiter ihre Arbeiten durch Überstunden nachholen. Verursacht durch den Stromausfall können weiter Konzessionsstrafen bei nicht termingerechter Aufgabenerfüllung entstehen. Des Weiteren können Kosten durch das Hochfahren der Systeme nach dem Stromausfall entstehen, die unter Umständen nur mit kostenintensivem Fachpersonal möglich sind.

In dem beschriebenen Szenario des Stromausfalls werden beispielsweise Kosten in Höhe von 5.000,- € erwartet. Da ein Stromausfall auf Grund der Auswertung vergangener Ereignisse alle 10 Jahre erwartet wird, wird mit einem jährlichen Verlust von 500,- € pro Jahr gerechnet. Wird als Maßnahme zur Reduktion des Risikos eines Stromausfalls die Installation einer unterbrechungsfreien Stromversorgung (USV) gewählt, so müssen die durch die USV verursachten jährlichen Kosten unter 500,- € liegen, damit die Maßnahme effizient ist. Kosten einer USV entstehen vor allem durch Abschreibungen sowie der Wartung und Installation der Anlage.

Die Einhaltung der Effizienz gilt aber nicht nur für eine einzelne Maßnahme, sondern ist auch auf das gesamte Maßnahmenportfolio zu beziehen. So dürfen die Kosten des Portfolios die Reduktion der Verlusterwartung nicht übersteigen.

Bei der Auswahl der Maßnahmen muss aber nicht nur die Effizienz, sondern auch Nebeneffekte wie Benutzbarkeit und Funktionalität betrachtet werden. Zwar steigt durch die Realisation einer Maßnahme die Sicherheit eines Systems, jedoch ist dies nicht sinnvoll, wenn das System nach Realisation der Maßnahme nicht mehr die benötigte Funktionalität liefert. Leidet die Benutzbarkeit eines Systems durch eine Maßnahme, besteht die Gefahr, dass die Benutzer sich dem System verweigern oder alles versuchen, die Maßnahme zu umgehen, wodurch sie unnütz wäre. Diese kontraproduktiven Einflüsse werden durch das in Abbildung 8 gezeigte Dreieck veranschaulicht. Die verschiedenen Einflüsse müssen bei der Auswahl des Maßnahmenportfolios bedacht werden, bevor es in der nun folgenden Phase realisiert wird.

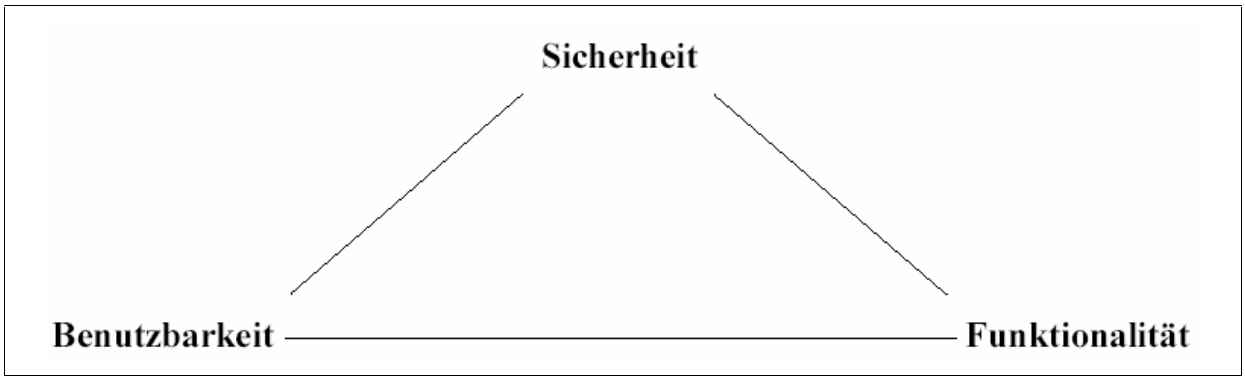


Abbildung 8: Kontraproduktive Einflüsse bei Realisation einer Maßnahme (nach [Nevers02:5])

2.4.4. Realisationsphase

Nachdem nun die Maßnahmen ausgewählt sind, müssen sie realisiert werden. Damit dies nicht planlos geschieht, was zu einer lückenhaften Sicherung oder zum Überschutz führt, wird zur Realisation der Maßnahmen innerhalb des Sicherheitsmanagements ein Sicherheitskonzept formuliert.

Erster Bestandteil eines jeden Managementkonzeptes ist eine Politik (engl. policy). Eine hier benötigte Sicherheitspolitik (engl.: security policy) ist jedoch nicht eindeutig definiert[20]. Nach

[20] „Policy means different things to different people“ [SP800-12:33]

[SP800-12:34] kann sie in organisatorische, themenbezogene und systembezogene Sicherheitspolitik differenziert werden.

Eine organisatorische Sicherheitspolitik[21], in [SP800-12:32] als „program policy“ bezeichnet, wird zur Initiierung des Sicherheitsprogramms einer Organisation verwendet. Sie ist die „Gesamtheit der Unternehmensgrundsätze, die ein Leitbild festlegen“ [Keuper01:2] und richtet damit die gesamte Organisation auf gemeinsame Ziele der Sicherheit aus. Eine solche Politik wird bereits vor der Risikoanalyse formuliert. Dabei können auch Grundsätze für die Risikoanalyse selbst enthalten sein.

Nach der Auswahl der Maßnahmen können mehrere themen- und systembezogene Sicherheitspolitiken formuliert werden. Sie bilden in ihrer Gesamtheit die IT-Sicherheitspolitik. In ihr werden neben den Anforderungen der Betriebsprozesse an einen Sachverhalt oder ein System auch abstrakte Maßnahmen definiert. Bei der Festlegung abstrakter Maßnahmen werden zunächst themenbezogene Politiken formuliert, die Themen wie Zugangskontrolle oder auch akzeptable Benutzung behandeln. Deren Anforderungen werden auf Systempolitiken heruntergebrochen, welche die Kriterien des Betriebs bestimmter Systeme wie Firewall, Webserver oder Workstation behandeln.

Die Politik soll alle Beteiligten über die Ziele informieren (vgl. [Fraser97:8]) und zu den Zielen leiten. Sämtliche Politiken regeln auch die generellen Verantwortlichkeiten innerhalb einer Organisation. Daher muss eine Politik für jeden verständlich sowie in natürlicher Sprache formuliert sein und darf keine technischen Details beinhalten. Dadurch wird die Politik auch langlebig und allgemein anwendbar, damit Änderungen an der IT-Infrastruktur im Rahmen der Politik bleiben. Deshalb sind die in der Politik verankerten Maßnahmen in abstrakter Weise zu formulieren.

Die IT-Sicherheitspolitik gibt auch Auskunft darüber, was beschützt wird, was erlaubt ist und welche Restriktionen durch die Maßnahmen gelten sollen. Die Maßnahmen besagen, „was man macht“ [Northcutt02:103], aber nicht wie, und stellen damit die Ziele dar, die es mit ihrer Implementation zu erreichen gilt. Die Politik hat dabei den Charakter einer Formel, welche die möglichen Handlungsalternativen verbindlich festlegt (vgl. [Strauss91:64]). In [McGinn-Combs02:2ff.] werden solche Maßnahmen auch als „meta-rules“ bezeichnet. Wie diese Maßnahmen ausgestaltet werden, wird in einer strategischen Ebene festgelegt. Für den Fall, dass eine Sicherheitspolitik nicht anwendbar ist, müssen in sehr speziellen Fällen Ausnahmen möglich sein.

Zusammenfassend sind hier die Zwecke einer Sicherheitspolitik aufgeführt (vgl. [Barman02:9])

- Sichert die geeignete Implementation der Maßnahmen
- Bietet Hilfe bei der Auswahl bestimmter Produkte
- Demonstriert die Unterstützung des Managements und vermittelt so Autorität
- Dokumentiert getroffene Maßnahmen und vermittelt Entscheidungen an die Beteiligten

[21] Ein Beispiel für eine solche Policy findet sich unter http://www.bsi.de/gshb/deutsch/aktuell/policy.pdf

Neben dem Konzept beinhaltet es in der strategischen Ebene die verfeinerte Ausgestaltung der Maßnahmen. Dabei helfen Standards, Richtlinien und Prozeduren.

Standards[22] (vgl. [SANS-PR03] und [SP800-12:34]) sind Anforderungen für einen einheitlichen Umgang mit Systemen oder Prozeduren. Ziel eines Standards ist eine erfolgreiche Anwendung der Sicherheit [Garfinkel96:36], die für alle Beteiligten verbindlich ist. Die Bindung der Standards ist schwächer als die einer Politik. Somit lassen Standards größere Ausnahmen zu. In [Garfinkel96:36] wird davon gesprochen, dass Standards befolgt werden sollen, im Gegensatz zur Politik aber nicht müssen.

Richtlinien (engl.: guidelines) geben ausführliche Empfehlungen für den optimalen Umgang mit Systemen und Prozeduren vor (vgl. [SANS-PR03]). Sie berücksichtigen die starke Änderung der Systeme über die Zeit, weshalb sie nicht immer anwendbar (vgl. [SP800-12:34]) und daher nur schwach bindend (vgl. [Großklaus99:59]) sind. „They are not requirements to be met, but are strongly recommended." [SANS-PR03]. Somit können Implementationsrichtlinien für bestimmte Systeme oder Prozeduren geschaffen werden.

Prozeduren (engl.: procedures) sind detaillierte Schritte, die von Nutzern, Administratoren und anderen Mitarbeitern zu befolgen sind, um eine bestimmte Aufgabe durchzuführen (vgl. [SP800-12:34]). Dabei wird eine Prozedur, die im Rahmen der Risikobewältigung als Gegenmaßnahme ausgewählt wurde, hier detailliert beschrieben. Neben der Detaillierung von Maßnahmen können Prozeduren auch die Einhaltung von Politiken, Standards und Richtlinien unterstützen (vgl. [SP800-12:34]).

Ist das Konzept formuliert, findet die Implementation der Systeme und organisatorischer Regelungen auf operationaler Ebene statt. Ein implementiertes Konzept muss einer ständigen Kontrolle unterliegen, die Aufgabe der nächsten Phase ist.

2.4.5. Kontrollphase

Durch die Sicherheitspolitik wird auf Basis der Risikoanalyse ein Sollzustand definiert, mit dem ein sicherer Betrieb gewährleistet werden soll. Dabei ist zu beachten, dass das Risikobild eine Momentaufnahme ist und der „Feind nicht schläft" [Krallmann89:74]. Daher muss der Sollzustand sich dem Risikobild anpassen, das sich durch neue Bedrohungen und Schwachstellen dynamisch ändert.

So muss der Istzustand durch Überprüfung auf Unregelmäßigkeiten sowie die Aufnahme von Verbesserungsvorschlägen und neuer Bedrohungen ständig kontrolliert werden. Daraus wird ein neuer Sollzustand definiert, um einen zukünftigen Erfolg des Sicherheitsmanagements zu gewährleisten. Eine solche zukunftsgerichtete Erfolgskontrolle entspricht nach Neuhof (vgl. [Krabbe98:173]) dem Aufgabenbereich des Controllings, zu dessen Aufgabe auch die Einleitung von Maßnahmen zur Verbesserung (vgl. [Krabbe98:175]) gehören.

Da jedes Risikobild eine Momentaufnahme ist, ist das Controlling in regelmäßigen Abständen zu wiederholen. Dazu kann das Controlling im Rahmen des in Abbildung 9 beispielhaft dargestellten unendlichen Kontrollprozesses durchgeführt werden.

[22] Nicht zu verwechseln mit (inter-)nationalen Standards wie z.B. ISO oder FIPS Standards.

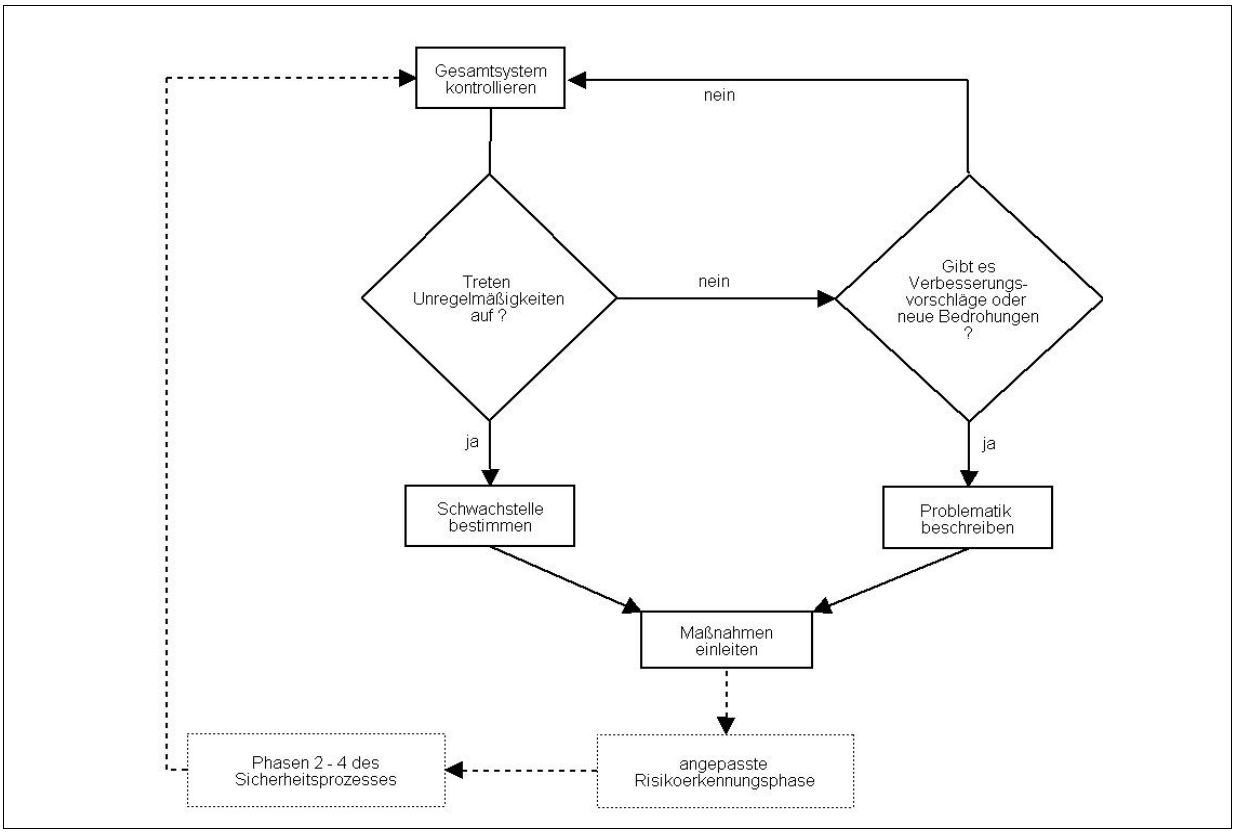


Abbildung 9: Kontrollprozess nach [Krallmann89:82]

Strauss [Stauss91:141f.] stellt zudem dar, dass Fehler auch im Entscheidungs- und im Implementierungsprozess auftreten können. Daher bedarf es der Überprüfung von Effektivität und Effizienz sowie der Feststellung, ob Entscheidungen und Implementation den Vorgaben entsprechen (vgl. [Strauss91:142]). Eine derartige Überprüfung hat den Charakter einer vergangenheitsorientierten Ordnungsmäßigkeitskontrolle, die nach [Krabbe98:173] in den Aufgabenbereich der Revision (engl.: auditing) fällt. Als Kontrollmaßstab der internen Revision „dient die oberste betriebliche Zielsetzung“ [Krabbe98:172]. Eine externe Revision benutzt dagegen außerbetriebliche Richtlinien, wie z. B. Bundesdatenschutzgesetz (BSDG), Vorschriften nach ISO oder die Grundsätze ordnungsgemäßer Buchführung (GoB) als Kontrollmaßstab. Bei positiv ausfallender externer Revision können auch Zertifikate erteilt werden. Eine Revision kann auch die Durchführung eines Penetrationstests beinhalten, weshalb sie in Abschnitt 3.4.3 näher betrachtet wird.

Wie das Controlling ist auch die Revision in regelmäßigen Abschnitten zu wiederholen. Dabei ist anzumerken, dass beide Kontrollinstanzen wegen der Erfordernis an großer Fachkenntnis nicht aufbauorganisatorisch getrennt werden können, obwohl die Revision „grundsätzlich von Personen durchgeführt [wird], die weder an der Vorgabe der Sollwerte noch an der Realisation der Istwerte beteiligt waren“ [Krabbe98:172]. Ein IT-Kontrolleur revidiert in regelmäßigen Abständen das Konzept. Zwischen den Zeitpunkten der Revision kontrolliert er das laufende Konzept im Sinne des Controlling.

Die Revision dient vor allem dem Aufbau von Vertrauen in die IT-Infrastruktur, mit dem sich der folgende Abschnitt beschäftigt.

2.5. Vertrauen

Vertrauen hat eine große Bedeutung. Organisationen pflegen wertschöpfende Geschäftsbeziehungen mit Geschäftspartnern wie z. B. Kunden. Wenn die Kunden den Systemen der Organisation nicht vertrauen können, aber den Systemen der Konkurrenz, werden sie zur Konkurrenz wechseln (vgl. [Klein01:1]). Vertrauen kann daher als Wettbewerbsvorteil gesehen werden, der zu einer Wertschöpfung führt.

„A computer is secure if you can depend on it and its software to behave the way that as you expect" [Garfinkel96:6]. Ein vollständig sicheres System muss sich verhalten, wie es von ihm erwartet wird. In wie weit sich ein System wie erwartet verhält, wird durch den Begriff Vertrauen (engl.: trust) beschrieben, das nach [Garfinkel96:26] als "[the] level of confidence that a computer system will behave as expected" definiert wird.

Vertrauen in ein System kann nach [Pfleeger00:270] durch Features[23] und Assurance gebildet werden. Diese Aufteilung beruht auf dem Orange Book, das durch die Evaluation einem System einen Vertrauensgrad (engl.: degree of trust) zuweist.

Die Features bilden den funktionalen Teil des Vertrauens. Sie beschreiben die Funktionalität, aus der die Gegenmaßnahmen gebildet werden. Der Vertrauensgrad steigt mit stärkeren Gegenmaßnahmen. Dabei muss auch sichergestellt werden, dass diese Gegenmaßnahmen auch den Erwartungen entsprechen. Das bezieht sich nicht nur auf die korrekte Funktionalität, sondern auch darauf, dass die gewählten Maßnahmen die Risiken wirklich minimieren. Nach Common Criteria benötigt der Besitzer eines Assets die Zuversicht (engl.: confidence), dass die Gegenmaßnahmen (engl.: countermeasures) die Risiken für die Assets minimieren. Die Zuversicht wird durch eine Zusicherung (engl.: assurance) erreicht. Abbildung 10 stellt diesen Sachverhalt dar. Der Grad der Zuversicht (engl.: „level of confidence") bildet nach der Definition das Vertrauen.

[23] Gemeint sind „security features", also Funktionalitäten wie logging oder Zugriffskontrolle, die Gegenmaßnahmen gegen Risiken bilden und somit zur Sicherheit beitragen.

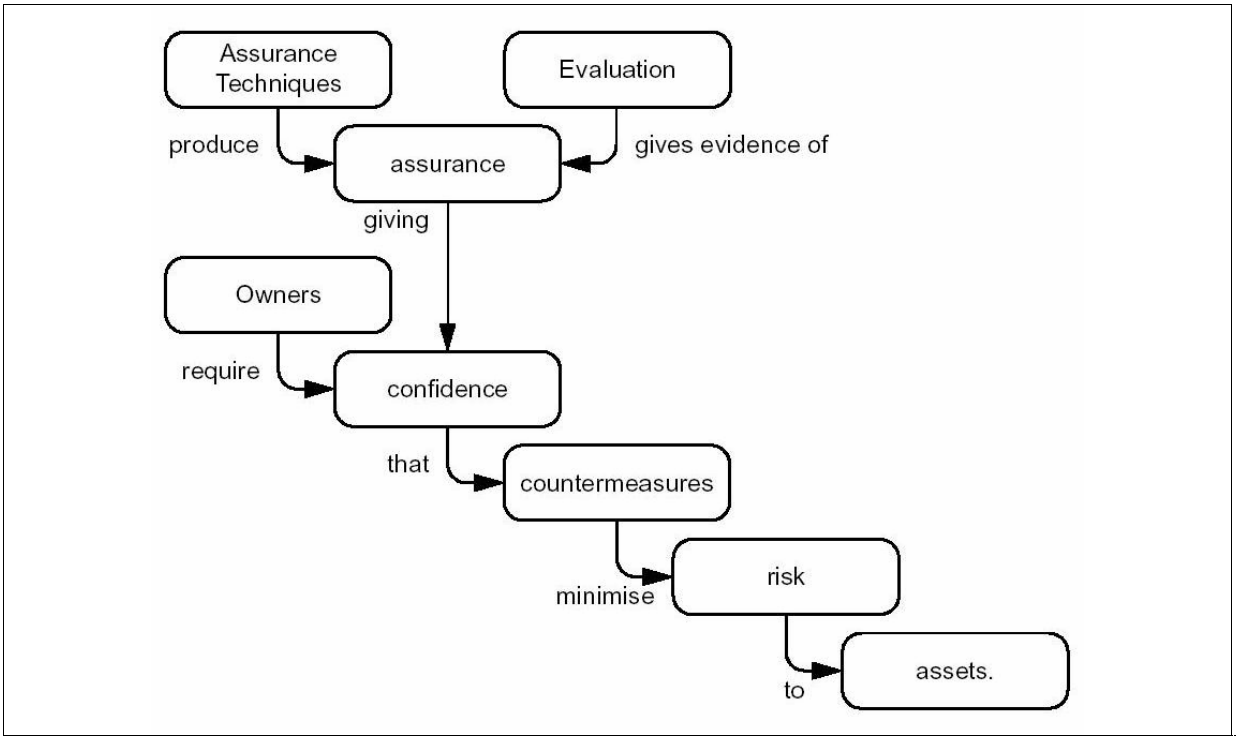


Abbildung 10: Zusammenhang der Fachbegriffe nach [CC99:15]

Techniken zum Erreichen der Zusicherung sind nach [Pfleeger00:308] Test, Verifikation und Validation, die im Folgenden beschrieben werden:

- Test

  Ein Test ist eine Demonstration, dass ein Testobjekt den Anforderungen aus bestimmten Testfällen genügt. Ein Testfall drückt die Anwesenheit einer bestimmten Eigenschaft aus, die durch einen Testvorgang überprüft wird. Ein potentieller Fehler wird nicht erkannt, wenn kein bestimmter Testfall erzeugt wird. E. Dijsktra sagte schon 1957 dazu: „Testen kann nur die Anwesenheit von Fehlern zeigen, niemals jedoch deren Abwesenheit" [Floyd00:Kapitel 16.7].

- Verifikation

  Eine Verifikation ist ein formaler Beweis, dass der Code den Anforderungen der Spezifikation entspricht. Dabei werden durch Beobachtungen an dem System Theorien über das System gebildet. Die Sätze der Theorie, die Theoreme, werden mit Hilfe formaler Methoden bewiesen. Somit können Implementationsfehler im Code ausgeschlossen werden. Wegen der hohen Komplexität eignet sich die Verifikation nicht immer und kann sogar fehlerhaft werden (vgl. [Pfleeger00:312]). Zudem ist fraglich, ob die Spezifikation keine Fehler enthält.

- Validation

  Eine Validation ist nach Pfleeger (vgl. [Pfleeger00:312]) ein Oberbegriff für den Nachweis von Korrektheit. Er beinhaltet neben der formalen Verifikation informale Methoden wie Differenzanalysen, Überprüfungen von Design, Implementation und Konfiguration durch Checklisten sowie Systemtests.

So wird deutlich, dass Vertrauen aus den Gegenmaßnahmen, die aus security features gebildet werden, und der Zusicherung, dass die Maßnahmen die erwartete Funktionalität aufweisen, entsteht. Die Bedeutung der Zusicherung wird bei Betrachtung der oberen Vertrauensstufen des Orange Book deutlich. In diesen Stufen kommen gegenüber den vorhergehenden Stufen kaum Features hinzu. ”Instead there is more Assurance“ (vgl. [Nedon00:7]).

Vertrauen hat aber Grenzen. Von vollständigem Vertrauen kann nur dann gesprochen werden, wenn ein Programm formal bewiesen ist. Der formale Beweis, der durch eine Verifikation erbracht wird, bezieht sich aber nur auf die korrekte und vollständige Funktionalität eines Programms hinsichtlich einer Spezifikation. Dabei wird weder gezeigt, dass die Spezifikation korrekt ist, noch dass das Programm nicht angreifbar ist. So zeigte Thompson (vgl. [Thompson84:1ff.] und [Garfinkel96:801f.]) in einem Experiment, dass in eine Verifikation auch Bibliotheken, Linker, Compiler[24], Assembler, Microcode und Loader mit einbezogen werden müssen, die ein Programm vor der Ausführung benutzt. Die Erkennung von Manipulationen des Microcodes sei nach Thompson fast unmöglich.

Im Experiment manipulierte Thompson den Befehl login von UNIX. Dabei manipulierte er nicht das Programm selbst, da eine solche Manipulation durch die Verifikation entdeckt werden könnte, sondern den Compiler. Bei Kompilation des korrekten Quellcodes von login manipuliert der Compiler die Ausgabe derart, dass auch die im System gespeicherten verschlüsselten Passwörter akzeptiert werden. Damit die Manipulation des Compilers nicht erkannt wird, wurde der Compiler auch dahingehend manipuliert, dass erst bei Kompilation des Compiler-Quellcodes der manipulierte Compiler erzeugt wird.

Das Experiment zeigt Grenzen des Vertrauens auf. Man kann nur dem Code vertrauen, den man selbst produziert hat (vgl. [Thompson84:5]). Zudem ist Sicherheit eine Folge von Aktionen und Gegenaktionen (vgl. [Garfinkel96:802]). Auf jede Aktion eines Angreifers kann wieder eine Gegenaktion erfolgen. Dieser Prozess kann endlos werden, so dass ab einem gewissen Punkt ein Vertrauen ohne fundierten Hintergrund hervorgebracht werden muss.

So wird dem Compiler ohne Zusicherung des korrekten Verhaltens vertraut, weshalb ein hundertprozentiges Vertrauen nicht erreicht werden kann. Dadurch muss immer ein gewisses Restrisiko getragen werden. Risiken können nur auf ein minimales Maß gesenkt werden. Hier liegt die Krux der IT-Sicherheit, weshalb es keine vollständige Sicherheit geben kann.

Die Grenzen machen ein Vertrauensmodell (engl.: Trust Model, vgl. [Schneier00:285]) notwendig. Dieses Konzept beschreibt, wie eine Organisation bestimmt, wem sie vertraut. So kann einem Angestellten allein dadurch Vertrauen geschenkt werden, dass er eine persönliche Magnetkarte besitzt. In manchen Vertrauensmodellen wird auch einem Provider blind vertraut, obwohl von diesem eine erhebliche Gefahr ausgeht (vgl. [Garfinkel96:809f.]). Ein solches Vertrauensmodell wird in einer Sicherheitspolitik festgeschrieben, deren Ausgangspunkt eine Risikoanalyse ist.

Um Vertrauen zu erreichen, muss ein Sicherheitskonzept erstellt werden, dass anschließend durch die Revision auf seine Ordnungsmäßigkeit überprüft wird. Die Revision ist ein Anwendungsgebiet des Penetrationstests, der im folgenden Kapitel näher betrachtet wird.

[24] Zur Kompilermanipulation vgl. [Garfinkel:800f]

3. KAPITEL:

PENETRATIONSTEST

Penetrationstests werden in der Praxis bereits verwendet. Dabei sind in der Literatur verschiedene Sichtweisen der Bedeutung des Penetrationstests sowie dessen Verwendungsmöglichkeiten zu finden. Diese sollen in diesem Kapitel vereinheitlicht werden, bevor in den folgenden Kapiteln eine Vorgehensweise entwickelt und angewendet wird.

3.1. Motivation

Motivation für das Thema ist eine Diskussion auf der „Firewalls Mailing Liste“ [ISC] mit dem Thema „Firewall Testing Recommendations“ [Chrichton01]. Begonnen wurde die Diskussion von David Chrichton, der zum Testen seiner Firewall einen Penetrationstest durchführen wollte. Dabei stellte er zunächst die Frage, welcher externe Dienst oder welches Tool am besten dafür geeignet sei.

In der Diskussion kamen neben der Abgrenzung des Begriffs Penetrationstest auch folgende Fragen auf:

- Ist ein Penetrationstest mehr als der Einsatz eines Vulnerability Scanners?
- Welche unterschiedlichen Varianten gibt es? Im Verlauf der Diskussion wurde von einem blinden Test und einem „Full Disclosure Test“ gesprochen, der auch die Konfigurationen des Systems testen soll.
- Was sind black-box und white-box Test?
- Wo liegen die Grenzen des Penetrationstests?
- Beinhaltet der Penetrationstest die Behandlung der Frage, ob ein System sich wie erwartet verhält, oder ist hier eine klare Abgrenzung zu sehen?
- Wie ist ein Penetrationstest zu einem Audit abgegrenzt?
- Beinhaltet der Test
    - die Betrachtung der Architektur des Netzwerkes?
    - eine Überprüfung der Befolgung von Prozeduren?
    - die Überprüfung, ob sich Systeme wie erwartet verhalten?
    - eine vollständige Überprüfung der Konfiguration eines Systems?
- Kann ein Penetrationstest in einem produktiven Netz durchgeführt werden?

Diese Fragen dienen der Motivation für die Behandlung des Penetrationstests. Sie sollen im weiteren Verlauf der Arbeit geklärt werden. Dazu werden zunächst Literaturquellen betrachtet, die das Thema Penetrationstest behandeln.

3.2. Penetrationstest in der Literatur

Die Älteste der hier behandelten Definitionen geht auf das Orange Book (vgl. [TCSEC85:110]) zurück. Dort wird unter Penetration Testing „the portion of security testing, in which the penetrators attempt to circumvent the security features of a system" verstanden. Das Security Testing ist Teil der Anforderungen zur Bildung der Zusicherung. Zum Nachweis der korrekten Funktionalität beinhaltet es den funktionalen Test und die Verifikation. Jedoch kann dadurch kein Nachweis der Nicht-Angreifbarkeit eines Systems erlangt werden. Um diese Lücke zu schließen, beinhaltet das Security Testing des Orange Books den Penetrationstest. Der Tester bekommt dabei sämtliche Dokumentationen des Designs und der Implementation eines Evaluationsobjektes. Begrenzt durch die Bedingungen, unter denen ein üblicher Benutzer mit dem System arbeitet, versucht er mit dem Penetrationstest, die Sicherheitsmaßnahmen des Systems zu umgehen.

Auch in den Common Criteria wird der Penetrationstest zum Erreichen der Assurance verwendet. Er ist Bestandteil der Assurance-Klasse „AVA:Vulnerability Assessment" (vgl. [CC99b:164ff.]). Aufgaben dieser Klasse sind nach [CC2002:12]

1. die Identifikation potentieller Schwachstellen eines Evaluationsgegenstandes
2. die Bestimmung, ob diese potentiellen Schwachstellen ausnutzbar sind.

Zur Bewältigung der ersten Aufgaben werden in einer methodischen Vorgehensweise Hypothesen über mögliche Schwachstellen im System gebildet und mögliche Angriffsmethoden entwickelt, mit denen die Schwachstelle ausgenutzt werden kann. Um zweitens zu bestimmen, ob eine angenommene Schwachstelle ausgenutzt werden kann, werden konkrete Angriffe durchgeführt. Die Durchführung dieser Angriffe wird als Penetrationstest verstanden. Bei erfolgreicher Penetration wird die Schwachstelle als „exploitable" bezeichnet. Eine ausnutzbare Schwachstelle ist ein Fehler, der zu einem Schaden führen kann und daher eine Schwachstelle im Sinne der Definition in Abschnitt 2.2. Ist ein solcher Test wegen der Erwartung eines zu hohen Schadens nicht durchführbar, so wird eine theoretische Beurteilung des Angriffes durchgeführt.

Ein Beispiel für ein solches Vorgehen ist die Hypothese, dass in einer Eingabefunktion die Grenzen des Eingabepuffers nicht geprüft werden, so dass es durch maliziöse Eingaben zu einem Pufferüberlauf und zur Ausführung von Code kommen kann. Um die Hypothese zu belegen, kann ein Angriff mit einem Perturbation Test durchgeführt werden. Hierbei wird eine Störung (engl.: perturbation) der Umgebung, wie z.B. fehlerhafte Eingaben, absichtlich erzeugt, um ein Fehlverhalten herbeizuführen (vgl. [Du00:4]).

Aber auch Schwachstellen, die bereits bekannt sind, werden in dieser Klasse der Common Criteria benutzt. Als Hypothesen einer Schwachstelle werden auch Schwachstellen des Produktes verwendet, die bis zum Zeitpunkt der Evaluation durch Herstellerangaben oder Informationen von unabhängigen Quellen wie z. B. CERT/CC bekannt geworden sind. Die Evaluation überprüft dabei, ob diese Schwachstellen im vorliegenden Evaluationsgegenstand ausnutzbar sind. Dieses Vorgehen wurde z.B. bei der Evaluation der Watchguard Firebox verwendet (vgl. [TTAP00:27]).

Auch im Grundschutzhandbuch des deutschen Bundesamtes für Sicherheit in der Informationstechnik (BSI) wird der Penetrationstest verwendet. Dabei wird „versucht, das

Angriffsverhalten eines vorsätzlichen Innen- oder Außentäters zu simulieren und zu ermitteln, welche vorhandenen Schwachstellen ausgenutzt bzw. welche potentiellen Schäden verursacht werden können“ [GSHB02:Kapitel 2.5]). Diese Vorgehensweise ist im Grundschutzhandbuch Teil einer „ergänzenden Sicherheitsanalyse“. Eine solche Analyse ist bei Assets mit hohem Schutzbedarf notwendig, bei denen die pauschale Zuordnung von standardisierten Maßnahmen nicht ausreicht. Dabei ist der Penetrationstest eine Methode zur Durchführung einer ergänzenden Sicherheitsanalyse. Weitere Methoden sind die in Abschnitt 2.4 besprochene detaillierte Risikoanalyse und die Differenz-Sicherheitsanalyse. Andere Autoren (vgl. [SP800-30:17] und [Barman02:8]) benutzen den Penetrationstest während einer Risikoanalyse.

Nach veit (vgl. [Veit99:3]) dienen Penetrationstests durch Simulation von Angriffsszenarien dem „Nachweis der Wirksamkeit der implementierten Mechanismen“, wobei sämtliche Komponenten den Angriffen widerstehen müssen. Die Simulation wird hierbei als Teil der Sicherheitsüberprüfungen von Maßnahmen verwendet. Diese Überprüfung ist eine Kontrolle im Sinne der Ordnungsmäßigkeit, also eine Revision. Auch andere Autoren (vgl. [Schultz96:1]) sehen den primären Zweck des Penetrationstests in der Methode zur Prüfung von Maßnahmen auf deren Effektivität, wobei die Effektivität den Wirkungsgrad darstellt. Dabei wird aus der Sicht eines Angreifers vorgegangen (vgl. [Moyer98:2]).

Mit diesen Einblicken in die Materie soll nun eine Begriffsbestimmung erfolgen.

3.3. Begriffsbestimmung

Der Begriff *Penetration* bedeutet nach [Duden1:519] „durchdringen“ oder „eindringen“. In der Geologie wird bei einem Penetrationstest der Boden durchdrungen, um durch Proben seine Beschaffenheit zu ermitteln (vgl. [Browner02]), bevor ein Bauwerk darauf errichtet wird. Die Medizin untersucht innerhalb der Gynäkologie bei einem Penetrationstest, ob Spermien den Gebärmutterschleim durchdringen können, um so auf die Fruchtbarkeit zu schließen. In der Informatik bedeutet der Penetrationstest analog das Durchdringen von Sicherheitsmaßnahmen eines Systems oder Netzwerkes, um dessen Resistenz gegen Angriffe vor Eintreten eines Vorfalls zu bestimmen. Da bei einem Penetrationstest auch so genannte Denial-of-Service Angriffe durchgeführt werden, wird an dieser Stelle das tatsächliche Eindringen in ein System mit dem Begriff Intrusion abgegrenzt. Ein Penetrationstest bedeutet daher die Durchführung von Angriffen, was mit dem Verständnis der oben genannten Beispiele aus der Literatur übereinstimmt.

Bei der Durchführung der Angriffe soll ein realer Angreifer nachgeahmt werden (vgl. [Moyer98:1]), so dass sein Verhalten simuliert wird und die Angriffe aus dessen Sichtweise erfolgen. Eine wesentliche Abgrenzung zu einem realen Angriff ist, dass die Angriffe von dem Besitzer des zu betrachtenden Systems oder Netzwerkes initiiert oder in Auftrag gegeben werden (vgl. [Robinson03:1]). Da ein Angriff Schäden anrichten und auch rechtliche Folgen mit sich bringen kann, müssen sämtliche Angriffe vom Initiator hinsichtlich ethischer und rechtlichter Bedenken kontrolliert werden.

Der Penetrationstest soll eine Grundlage bieten, um Aussagen über die Sicherheit eines bestimmten Kontextes machen zu können. Ein Leitbild für die Durchführung ist dabei ein Zitat von Sun Tzu aus seinem Buch “The Art of War”: „Knowing your enemy is the key to

winning the battle“ [Cole02:19]. Mit der Kenntnis seiner Vorgehensweise soll einem Gegner begegnet werden, bevor er einen Schaden anrichten kann. Auf Grund der dargestellten Überlegungen kann ein Penetrationstest wie folgt definiert werden:

DEFINITION

Penetrationstest	ist die kontrollierte Durchführung von Angriffen gegen ein Betrachtungsobjekt aus der Sichtweise eines Angreifers unter Einhaltung ethischer und rechtlicher Gesichtspunkte; die Durchführung wird von den Verantwortlichen des Betrachtungsobjektes initiiert oder in Auftrag gegeben.

Wegen der Durchführung von Angriffen unter Einhaltung ethischer Gesichtspunkte wird der Penetrationstest auch als „ethical hacking“ bezeichnet. Der Zweck dieser Vorgehensweise ist abhängig vom Anwendungsgebiet. Die verschiedenen Anwendungsgebiete werden im nächsten Abschnitt betrachtet.

3.4. Anwendungsgebiete

Der Penetrationstest wird in mehreren Gebieten verwendet. Aus der Literatur lassen sich fünf Anwendungsgebiete erkennen:

- Softwareentwicklung
- Risikoanalyse
- Revision
- Incident Response Teams
- Schwachstellenanalyse

Sie werden in den folgenden Abschnitten genauer betrachtet.

3.4.1. Softwareentwicklung

Im ersten Einsatzgebiet, der Softwareentwicklung, dient der Penetrationstest der Entdeckung bisher unbekannter, potentieller Schwachstellen in einem System. Auf Grund der gewonnenen Erkenntnisse werden Korrekturen an der Software durchgeführt. Am Markt vertretene Softwareunternehmen wie z.B. Macromedia (vgl. [Madar03]) setzen neben der Revision[25] des Quellcodes auch den Penetrationstest ein. Die Vorgehensweise eines Penetrationstests ist bereits bei der Betrachtung des Penetrationstests innerhalb der Common Criteria (vgl. Seite 36) erläutert worden.

3.4.2. Risikoanalyse

In der quantitativen Risikoanalyse werden Daten über zu erwartende Schäden und der Wahrscheinlichkeit bestimmt, aus denen das Risiko bestimmt wird. Zur Bestimmung dieser

[25] Die Revision wird meist nicht von den Softwareunternehmen selbst durchgeführt, sondern an externe Spezialisten abgegeben (vgl. [Madar03]).

Daten kann auch ein Penetrationstest helfen, in dem die Konsequenz der Ausnutzung bestehender Schwachstellen durch Darstellung der entstehenden Schäden aufgezeigt wird (vgl. [GSHB02:Kapitel 2.5]).

Des Weiteren kann der Penetrationstest die Vorhersage der Wahrscheinlichkeit der Ausnutzung einer Schwachstelle durch eine Bedrohung erleichtern. Die Wahrscheinlichkeit kann anhand der Beschaffenheit der Schwachstelle ermittelt werden (vgl. [SP800-30:21]). So kann aus den Ergebnissen des Penetrationstests erkannt werden, wie schwierig es war, eine bestimmte Schwachstelle auszunutzen. Zwar ist eine genaue Quantifizierung der Wahrscheinlichkeit kompliziert, jedoch lässt sie sich leicht in die Schadensklassen der semi-quantitativen Risikobewertung einordnen.

Während das BSI-Grundschutzhandbuch den Penetrationstest neben der Risikoanalyse als Ergänzung zu einer Sicherheitsanalyse benutzt, wird er in der Risikoanalyse nach NIST (vgl. [SP800-30:17]) innerhalb einer Schwachstellenanalyse verwendet. Demnach können mit den in Abschnitt 2.4.1 angegebenen Quellen lediglich Beobachtungen von Schwachstellen in den Systemen bestimmt werden. Um diese Beobachtungen zu validieren, werden Testverfahren benötigt. Als Testverfahren können neben Vulnerability Scannern auch weitere System Tests durchgeführt werden, um die Beobachtungen der Schwachstellen zu belegen. So können die Systeme auf Fehler getestet werden, die eine Schwachstelle darstellen. Weiter können hierbei auch Penetrationstests verwendet werden. Sie gehen von der Sicht des Angreifers aus. Dabei kann nicht nur erkannt werden, ob eine Schwachstelle tatsächlich vorhanden ist, sondern auch, ob die einer Schwachstelle entgegenwirkenden Maßnahmen ausreichen und das Risiko reduzieren (vgl. [SP800-30:17]). So können diese Testmethoden bereits vorausschauend angewendet werden, um Lücken in der Maßnahmenauswahl während der Risikobewältigung proaktiv zu vermeiden. Da die Überprüfung, ob eine Maßnahme ein Risiko ausreichend reduziert, zum Aufgabengebiet der Revision gehört, werden zum Teil in der Risikoanalyse im Rahmen der Schwachstellenanalyse nur Beobachtungen (engl.: observations) von Schwachstellen ermittelt (vgl. [SP800-30:19]). In diesem Fall ist für weitere proaktive Überprüfungen eine Revision im Anschluss an die Risikobewältigung nötig. Mit der Revision beschäftigt sich der folgende Abschnitt.

3.4.3. Revision

Ein implementiertes Sicherheitskonzept ist nicht frei von Fehlern. Todd (vgl. [Todd:2]) spricht in seinem Artikel über die Überprüfung einer Firewall davon, dass nach der Installation einer Firewall um dessen Sicherheit Sorge getragen werden muss. Daher bedarf es einer ständigen Kontrolle der Ordnungsmäßigkeit des Sicherheitskonzeptes, was Aufgabe der Revision ist (vgl. Abschnitt 2.4.5).

Die Sicherheitsrevision (engl.: security audit[26]) wird auch als Sicherheitsüberprüfung bezeichnet. Sie soll betrachten, ob die Gegenmaßnahmen des Konzeptes die Risiken für die Assets minimieren, wodurch Vertrauen (vgl. Abschnitt 2.5) in das Konzept geschaffen

[26] Hierbei kommt es zu einem Konflikt durch die doppelte Belegung des Begriffs „security audit“. Neben der Bedeutung der Revision wird in der Informatik unter dem Begriff auch das Erkennen, Aufzeichnen, Speichern und Analysieren von sicherheitsrelevanten Aktivitäten (vgl. [CC99a:17]) verstanden. In dieser Arbeit soll letztere Bedeutung mit dem aus dem Orange Book stammenden Begriff Audit Trail (vgl. [TCSEC85:106] abgegrenzt werden.

werden kann. Ein solches Vertrauen stärkt nicht nur das Vertrauen der Kunden in eine Organisation, sondern ist auch für Anteilseigner, Gläubiger und Mitarbeiter von Interesse. In Anlehnung an Strauss (vgl. [Strauss91:141f.]) werden dabei folgende Ziele verfolgt:

- Kontrolle auf Fehler in der Entscheidung bei der Auswahl von Maßnahmen zur Bildung des Maßnahmenportfolios, wobei auch die Aktualität der Maßnahmen zu kontrollieren ist
- Kontrolle der Implementation

Ziel ist nachzuweisen, dass die Maßnahmen die gewünschte Effektivität und Effizienz leisten.

Eine Vorgehensweise zur Revision kann aus den Common Criteria abgeleitet werden. Dort wird die Zuversicht bestimmt, dass die Gegenmaßnahmen die Risiken für die Assets minimieren (vgl. [CC99:15]). Der Grad der Zusicherung bildet nach der Definition (vgl. Seite 32) das Vertrauen.

Vertrauen entsteht nach [Pfleeger00:270] aus den Merkmalen (engl.: Features) und einer Zusicherung (engl.: Assurance). Abgeleitet auf ein Sicherheitskonzept bilden die Maßnahmen die Merkmale des Sicherheitskonzeptes. Dabei muss eine Revision das Konzept auf Fehler in Zusammensetzung des Maßnahmenportfolios kontrollieren.

Neben der Kontrolle des Portfolios muss die Revision auch die Zuversicht schaffen, dass die gewählten Maßnahmen die Risiken für die Assets minimieren. Dazu müssen Aussagen über die Qualität der Umsetzung getroffen werden, was eine Betrachtung der Effektivität bedeutet. Die Effektivität betrachtet dabei die Wirksamkeit der Maßnahmen.

Die Effektivität einer Maßnahme wird nach Strauss (vgl. [Strauss91:130]) durch das Maß definiert, um das eine Maßnahme das Risiko reduziert. Um Aussagen über die Effektivität der Maßnahmen eines implementierten Sicherheitskonzeptes zu tätigen, muss die Implementation betrachtet werden. Dabei wird betrachtet, ob die Maßnahmen bezüglich des Sicherheitskonzeptes vollständig und korrekt implementiert worden sind. Werden dabei Lücken in der Umsetzung erkannt, so kann eine Maßnahme die gewünschte Risikoreduktion nicht erreichen und somit nicht effektiv sein.

Nachdem das Sicherheitskonzept selbst auf Ordnungsmäßigkeit überprüft worden und die Implementation im Einklang mit dem Konzept ist, fehlt jedoch eine Zusicherung der Wirksamkeit der Maßnahmen, wobei die Maßnahmen dann wirksam sind, wenn sie die Risiken für die Assets auf ein tolerierbares Maß reduzieren. Zum Erreichen der Zuversicht soll innerhalb der Revision nachgewiesen werden, dass keine untolerierbaren Risiken in der IT-Infrastruktur bestehen.

Ausgehend von dem qualitativen Bewertungsansatz, bei dem das Risiko aus der Verknüpfung von Bedrohung und Schwachstelle entsteht, kann gezeigt werden, dass keine Schwachstellen in der Infrastruktur bestehen. Dazu kann mittels Systemtests und manuellen Methoden die Anwesenheit von Schwachstellen überprüft werden. Komplementär dazu wird gezeigt, welche Bedrohung auf die Assets einwirken kann. Dies kann erreicht werden, indem die Assets Angriffen ausgesetzt werden, was durch einen Penetrationstest erreicht werden kann.

Winkler (vgl. [Winkler00]) bezeichnet nur die Überprüfung des Sicherheitskonzeptes und den Abgleich der Implementation als Revision, während die Überprüfung auf Schwachstellen sowie der Penetrationstest von Winkler als alternative Methoden zur Revision dargestellt werden, da sie keinem Standard folgen. Dennoch sind sie Bestandteil der Revision, da sie zum einen zur Überprüfung der Ordnungsmäßigkeit genutzt werden, zum anderen zur Überprüfung der Einhaltung des Sicherheitskonzeptes verwendet werden, das als interner Standard zu sehen ist. So nennt Veit (vgl. [Veit99:3]) sämtliche bisher betrachteten Überprüfungen als Ziele der Revision. Aus diesen Überlegungen lässt sich die Revision in die folgenden drei Ebenen gliedern:

1. Features:

   Überprüfung des Sicherheitskonzeptes auf Vollständigkeit, Korrektheit und Konsistenz der Maßnahmen

2. Assurance I:

   Nachweis der Wirksamkeit der Maßnahmen durch Überprüfung auf Anwesenheit von Schwachstellen

3. Assurance II:

   Nachweis der Wirksamkeit der Maßnahmen gegen die Bedrohungen

Durch die Betrachtung von Features und Assurance kann durch die Gesamtheit der Revisionsebenen ein Vertrauen im Sinne von Pfleeger in die Maßnahmen erreicht werden, dass sie die Risiken für die Assets minimieren.

Die Ebene Features überprüft die Auswahl der Maßnahmen. Dabei wird hinsichtlich der Korrektheit[27] überprüft, ob eine Maßnahme ein durch sie abgedecktes Risiko reduziert. Hinsichtlich der Vollständigkeit wird überprüft, ob alle Risiken erkannt worden sind und für jedes aktuell bestehende Risiko eine Maßnahme existiert. Damit eine konsistente Implementation existieren kann, muss auch kontrolliert werden, dass sich zwei Maßnahmen gegenseitig nicht ausschließen.

Um die Überprüfung auf Korrektheit, Vollständigkeit und Widerspruchsfreiheit leisten zu können, müssen Risikoanalyse und –bewältigung revidiert werden. Im Extremfall werden Risikoanalyse und –bewertung vom Revisor erneut durchgeführt und deren Ergebnis mit den bestehenden Dokumenten verglichen (vgl. [Todd98:2]). Nach Kapp (vgl. [Kapp00:3]) solle die Sicherheitspolitik dabei als initiale Bedrohung angesehen werden. Somit können Fehler im Entscheidungsprozess gefunden werden. In dieser Phase kann auch die Einhaltung externer Standards wie z. B. ISO17799 oder das Grundschutzhandbuch geprüft und zertifiziert werden, was dem Charakter einer externen Revision entspricht.

In der hier durchgeführten Unterteilung der Revision ist der Abgleich der Implementation mit dem Konzept nicht explizit enthalten. Sie wird aber implizit von den beiden Assurance Ebenen geleistet. Dies wird bei der Betrachtung der Methoden zur Durchführung dieser beiden Ebenen deutlich, da es bei expliziter Forderung eines Abgleiches zu Überschneidungen kommen würde. So ist beispielsweise ein Backup, das trotz der Vorschrift

[27] Inhalt von Korrektheit und Vollständigkeit aus [Eschenbach02:6-2] hergeleitet

durch das Sicherheitskonzept nicht vorhanden ist, ein Fehler in der Implementation des Konzeptes, der zu einem Schaden führen kann und somit eine Schwachstelle der IT-Infrastruktur darstellt.

Weiter wird das Sicherheitskonzept durch Konfiguration der Systeme wie Firewall oder Webserver implementiert. Dabei können Konfigurationsfehler und somit Schwachstellen entstehen, wodurch ein Sicherheitskonzept nicht korrekt implementiert ist. Der Vermeidung dieser Schwachstellen dient die Ebene Assurance I.

In der Ebene *Assurance I* wird das Risiko einer fehlerhaften Implementation der Maßnahmen von den Schwachstellen betrachtet. Werden keine Fehler gefunden, so steigt das Vertrauen in die Wirksamkeit des implementierten Konzeptes. Dazu werden manuelle Überprüfungen und Testverfahren eingesetzt. So kann bei einer Durchsicht der Systemkonfigurationen nach Evidenzen von Schwachstellen gesucht werden. Dabei kann beispielsweise geprüft werden, ob alle Patches eingespielt sind. Zudem können auch Testverfahren verwendet werden, die eine korrekte Funktionalität der Implementation demonstrieren.

Ein Beispiel für ein solches Testverfahren ist ein Regeltest der Firewall, wobei ein Rechner verschiedene Pakete durch die Firewall schickt, welche ein Rechner auf der anderen Seite der Firewall aufzeichnet. Ein solches Testverfahren wird in [SP800-30:21] als „System Test and Evaluation“, kurz ST&E, bezeichnet. Der erste Schritt einer solchen Überprüfung ist die Erstellung eines Testplans. Ein Rahmenwerk für die Erstellung eines solchen Testplans und dessen Durchführung bietet das Open Source System Test Methology Manual [OSSTMM02]. Zur Verringerung des Aufwands können auch Vulnerability Scanner eingesetzt werden, die eine automatische Testmethode auf die Anwesenheit von Schwachstellen bieten. Allerdings sind dabei Methodik und Ergebnisse der Scanner zu hinterfragen, um falsche Ergebnisse der Überprüfung zu vermeiden.

Der Methodik mangelt es allerdings an den generellen Begrenzungen eines Testverfahrens. So kann lediglich die Anwesenheit eines Fehlers und somit einer Schwachstelle gezeigt werden, jedoch nicht die Abwesenheit. Um die Auswirkung dieser Einschränkung zu begrenzen, ist eine möglichst vollständige Testspezifikation zu erstellen. Dabei sind alle Schwachstellen mit einzubeziehen, denen durch das Konzept begegnet worden sein soll. Daher ist die Assurance I unter Verwendung des Sicherheitskonzeptes durchzuführen. Weiter ist bei einer gefundenen Schwachstelle gegenzuprüfen, ob diese bei korrekter Implementation des Konzeptes nicht vorhanden sein dürfte. Mit dieser Methodik kann die Konsistenz von Sicherheitskonzept und Implementation aufgezeigt werden. So wird die Wirksamkeit der Maßnahmen demonstriert, die zur Risikoreduktion den Schwachstellen entgegenwirken.

Die Ebene *Assurance II* bildet das Komplement der Assurance I. Hierbei werden Bedrohungen auf die Assets angewendet. Ziel dieser Phase ist zu testen, ob die vorhandenen Assets den Bedrohungen widerstehen und die Maßnahmen zur Reduktion des Risikos wirksam sind.

Mit dieser Methodik kann weiteren Schwächen der Ebene Assurance I entgegengewirkt werden. So können nur Evidenzen der Behebung einer Schwachstelle gefunden werden. Ist beispielsweise ein Patch eingespielt, ist zwar eine Evidenz der Behebung der zugehörigen Schwachstelle geschaffen worden. Dennoch kann der Patch Fehler enthalten oder fehlerhaft installiert sein, was durch die Anwendung des passenden Exploits überprüft werden kann.

Weiter ist die Wirksamkeit bestimmter Maßnahmen wie z. B. ein Intrusion Detection System (IDS) nur durch die Anwendung der Bedrohung zu testen. Durch einen strukturierten Penetrationstest können dabei die Bedrohungen auf die schutzbedürftigen Systeme angewendet werden und betrachtet werden, ob das IDS alle Angriffe erkennt und effektiv meldet. Auch Maßnahmen gegen Social Engineering können nur so effektiv überprüft werden. Es kann darauf geachtet werden, dass die Mitarbeiter daraufhin geschult werden. Ob die Mitarbeiter es auch einhalten, kann nur durch die Ausführung eines Social Engineering Angriffes überprüft werden. Zudem können mögliche Seiteneffekte eines Vorfalls nicht geprüft werden.

Eine Methode zur Demonstration der Widerstandsfähigkeit der Assets ist der Penetrationstest, mit dem kontrollierte Angriffe gegen die Assets durchgeführt werden. Die Entwicklung einer Vorgehensweise für einen Penetrationstest erfolgt in Kapitel 5. Ist einer der im Rahmen des Penetrationstests durchgeführten Angriffe erfolgreich, so ist gezeigt worden, dass eine zur Bedrohung gehörende Schwachstelle vorhanden ist. Somit können eventuell vorhandene Evidenzen der Behebung der Schwachstelle widerlegt werden.

Mit der Assurance II kann erkannt werden, ob die Maßnahmen korrekt implementiert wurden und ihre Wirksamkeit den Anforderungen des Konzeptes entsprechen. Somit können auch auf dieser Ebene Inkonsistenzen der Implementation zum Konzept aufgezeigt werden. Des Weiteren können die Auswirkungen einer nicht geschlossenen Schwachstelle demonstriert werden. Wenn Schwachstellen bereits in den vorhergehenden Ebenen erkannt worden sind, kann der Revisor nur das Vorkommen der Schwachstelle anzeigen und eine Empfehlung geben, wie diese zu beheben ist. Auf Grund der fehlenden Entscheidungsgewalt kann er jedoch nicht durchsetzen, dass eine Schwachstelle geschlossen wird. Er kann lediglich einen Anreiz bieten, indem er die möglichen Auswirkungen demonstriert, was durch die Assurance II geleistet werden kann.

Problematisch ist allerdings die Aussagekraft. Wenn es dem Revisor nicht möglich war, den Angriff erfolgreich durchzuführen, ist damit nicht gezeigt worden, dass die Bedrohung nicht auf die Assets einwirken konnte. Um daraus keine unzulänglichen Schlüsse zu ziehen, sollten beide Assurance Ebenen komplementär verwendet werden.

Die hier vorgestellte Revision wird nicht immer als ganzes gesehen. So stellt Kossakowski (vgl. [Kossakowski01]) unter anderem ein Review der Systeme hinsichtlich der Schwachstellen oder die Ausübung von Angriffen als verschiedene Methoden der Revision dar. Dies lässt den Eindruck entstehen, dass diese unabhängig angewendet werden können. Damit würden die Unzulänglichkeiten der einzelnen Methoden nicht abgedeckt werden, wie es bei der kombinierten Anwendung der Fall ist.

Als Beispiel für die Anwendung der Revision wird die Überprüfung eines Firewallkonzeptes betrachtet. In der Ebene Features kann der Revisor beispielsweise feststellen, dass SYN-Flood Angriffe (vgl. Abschnitt 4.7.7) nicht durch das Firewallkonzept abgedeckt sind. Ist nun eine Maßnahme zum Schutz gegen diese Angriffe getroffen, so wird in der Assurance I überprüft, ob an der Firewall der gewünschte Schutz konfiguriert ist. Hierzu kann der Revisor beispielsweise bei einer auf Linux und iptables basierenden Firewall prüfen, ob die SYN-Cookies aktiviert sind. In der Assurance II wird dann überprüft, ob nun die SYN-Flood Angriffe erfolgreich abgewehrt werden. Ein SYN-Flood Angriff, der im Rahmen der

Assurance II gegen die Firewall durchgeführt wird, ist dabei ein Penetrationstest. Widersteht die Firewall dem Angriff, so kann von einer Zuversicht gesprochen werden, dass die Firewall das Risiko der SYN-Flood reduziert hat.

Zudem kann die Revision auch die Effizienz bewerten, die sich nach Strauss (vgl. [Strauss91:138]) aus der Differenz der quantifizierten Effektivität[28] und den Kosten für die Realisation der Maßnahme ergibt. Entsteht ein negativer Wert, so ist die Maßnahme nicht effizient. Eine solche Ineffizienz kann auch entstehen, wenn bei fehlerhafter Realisation einer Maßnahme nicht die gewünschte Effektivität erreicht wird. Ein effizientes Sicherheitskonzept ist eine Forderung an das Sicherheitsmanagement (vgl. Abschnitt 2.4). Die Feststellung der Effizienz ist abhängig von der subjektiven Bewertung der Risiken, die nur der Besitzer der Assets tätigen kann. Daher soll die Effizienz in dieser Arbeit nicht weiter behandelt werden.

In der Ebene Features können auch Aussagen über die Aktualität der Maßnahmen zu einem bestimmten Zeitpunkt gemacht werden. Dies widerspricht nicht dem Prinzip der Revision, da sie das Konzept von der Vergangenheit bis zu einem Zeitpunkt betrachtet. Dabei entsteht innerhalb der Revision eine Momentaufnahme. Auf Grund der fehlenden Entscheidungsbefugnis des Revisors können durch die Revision keine Aussagen getroffen werden, ob die Aktualität in der Zukunft sichergestellt wird. Es kann nur eine Evidenz aufgezeigt werden, die besteht, wenn ein zukunftsgerichteter Kontrollprozess im Sinne des Controllings (vgl. Abschnitt 2.4.5) etabliert ist.

Das Ergebnis der Revision ist das Vertrauen, dass die Maßnahmen des Konzeptes die Risiken minimieren. Das Vertrauen steigt mit dem Grad der Zuversicht, der von den Merkmalen und der Zusicherung der Minimierung der Risiken durch die Merkmale abhängt. Die Merkmale eines Sicherheitskonzeptes sind die Maßnahmen. Je mehr Maßnahmen in die Überprüfung einbezogen werden, desto höher ist der Grad der Zuversicht in das Konzept. Von Vertrauen kann aber nur dann gesprochen werden, wenn der Grad der Zuversicht durch Test- und Validationsmethoden zugesichert ist. Je vollständiger diese Methoden sind, desto höher ist die Zuversicht und somit das Vertrauen. Ein volles Vertrauen ist nur dann gegeben, wenn die Korrektheit und Vollständigkeit formal bewiesen sind. Die Methode zum formalen Beweis ist die Verifikation. Auf Grund der Grenzen der Verifikation (vgl. Abschnitt 2.5) kann nicht alles formal bewiesen werden, so dass ein vollständiges Vertrauen nicht möglich ist. Ein Vertrauen kann aber nur dann erreicht werden, wenn die bei der Revision gefundenen Mängel behoben werden.

Die Ziele der hier vorgestellten Ebenen der Revision stellen nach der Ansicht von Veit (vgl. [Veit99:3]) eine Auswahlmöglichkeit dar. Auf Grund der gestellten Anforderungen an die Revision können Prioritäten gesetzt werden und die Revision auf Teilaspekte beschränkt werden. Beispiele sind die Bestimmung der Effektivität einer Firewall (vgl. Schultz96]) oder die Wirksamkeit eines Intrusion Detection Systems (vgl. [BSI02]).

3.4.4. Incident Response Teams

Im Rahmen der Revision können auch die Fähigkeiten eines Incident Response Teams durch den Penetrationstest überprüft werden (vgl. [Wai01:1]). Hierzu werden durch die Angriffe

[28] quantifizierte Effektivität E = erwartete Kosten Kr bei Eintreten des Risikos– erwartete Kosten Km bei Eintreten des Risikos, wenn Maßname realisiert

absichtlich Incidents erzeugt, die zu angemessenen Reaktionen führen sollen. Betrachtet wird, ob die während des Penetrationstests durchgeführten Angriffe an ein Team gemeldet werden und ob das Team effektiv auf die Angriffe reagiert. Dabei kann überprüft werden,

- wie lange ein Team zur Reaktion braucht,
- ob die Reaktion den Angriff eindämmt und
- ob das Team alle durch den Angriff verursachten Schäden beseitigt.

Der Penetrationstest kann auch zur Schulung von Incident Response Teams eingesetzt werden. Bei einem Projekt im WS2002/03 am Arbeitsbereich AGN des Fachbereichs Informatik der Uni Hamburg wurde als erster Schritt zur Erlangung von Incident Response Fähigkeiten ein Angriff mittels Würmern durchgeführt. Die Tests, die in den Szenarien ab Kapitel 6 durchgeführt werden, schließen an diesen Versuch an. Eine Methode zur Erkennung des durch die Angriffe erzeugten Incident wird in der Diplomarbeit von Jan Menne [Menne03] behandelt, die zeitgleich zu dieser Arbeit entsteht.

3.4.5. Mythos Schwachstellenanalyse

In der Literatur wird der Penetrationstest häufig als eine Methode zur Identifikation von Schwachstellen verstanden. Wai [Wai01:1] sieht den Penetrationstest als großartige Vorgehensweise, um Schwachstellen bei realisierten Maßnahmen zu finden. So wird in [Inside Security Glossar] der Penetrationstest als „Angriffsversuch in ein Netzwerk zur Feststellung von existierenden Schwachstellen von IT-Systemen“ definiert. Ebenso werde nach [Bräuer02:1] der „[...] Test auf mögliche Schwachstellen von Computern, also das Vulnerability Assessment, auch Penetrationstest genannt“. Weiter schreibt Bräuer: „Die Schwachstellenanalyse geht dabei vom Blickwinkel des Angreifers aus[…]“, womit Penetrationstest und Schwachstellenanalyse gleichgesetzt werden. Bei einer Internetsuche nach kommerziellen Angeboten ließ sich ferner feststellen, dass die meisten Firmen in ihren Leistungsbeschreibungen den Penetrationstest als eine Methode zum Aufspüren von Schwachstellen verstehen.

Das Ziel eines Penetrationstests ist die Ausnutzung von Schwachstellen (vgl. [Winkler00:2ff.]). Dazu müssen zwar zunächst Schwachstellen gefunden werden, aber nicht alle Schwachstellen identifiziert, da dies üblicherweise nicht das Ziel eines Angreifers ist. Ein Penetrationstest soll einen Angreifer nachahmen. Somit wird versucht, ein bestimmtes Ziel durch die Angriffe zu erreichen, für das aber nicht alle Schwachstellen identizifiert werden müssen. Die Schwachstellenanalyse hingegen soll nach [Krallmann89:35] alle Schwachstellen erkennen. Zwar könnte ein Penetrationstest mit dem Ziel durchgeführt werden, alle Schwachstellen zu erkennen, aber keine auszunutzen. Diese Methodik wäre sinnvoll, wenn das Ziel beispielsweise darin besteht, dass alle von einem externen Netz wie dem Internet erkennbaren Schwachstellen aufgezeigt werden sollen. Jedoch können mit einer solchen Vorgehensweise nicht alle bekannten Schwachstellen erkannt werden. Somit ist für eine Schwachstellenanalyse eine Durchsicht der Systeme wegen des zu erwartenden Aufwandes sinnvoller (vgl. [Kurtz00a:2]).

Da der Penetrationstest nicht alle Schwachstellen aufdeckt, ist er keine Schwachstellenanalyse. Auch die Verwendung eines Penetrationstests in der Schwachstellenanalyse ist wegen des Aufwandes nicht sinnvoll. Der Penetrationstest kann aber in der an die Analyse

anschließende Bewertung der Schwachstelle eingesetzt werden, da er die Auswirkung einer Schwachstelle aufzeigen kann.

Für die Schwachstellenanalyse können auch Vulnerability Scanner eingesetzt werden. In wie weit diese Tools in einem Penetrationstest Verwendung finden, ist Inhalt des folgenden Abschnitts.

3.5. Vulnerability Scanner

Neben dem Mythos, der Penetrationstest sei eine Schwachstellenanalyse, besteht auch die falsche Vorstellung, dass die Ausführung eines solchen Scanners ein Penetrationstest sei (vgl. [Kurtz00a:1]). Daher werden in diesem Abschnitt die Vulnerability Scanner näher betrachtet, um die Möglichkeiten ihres Einsatzes allgemein und in Verbindung mit einem Penetrationstest zu klären.

3.5.1. Begriffsabgrenzung

Ein Vulnerability Scanner, auch Security Scanner genannt, ist eine Software, die eine Anzahl von IT-Systemen automatisiert auf die Anwesenheit von Schwachstellen überprüft. Dazu suchen die Scanner im aktuellen Zustand eines Systems nach Anzeichen einer Schwachstelle.

Die Entwicklung der Vulnerability Scanner geht aus von einem Artikel von Dan Farmer [Farmer93] über die Verbesserung der Sicherheit eines Netzwerkes durch einen Einbruch hervor. Dabei behandelt Farmer die Beschaffung von Informationen über das System und die Erkennung von Schwachstellen im betrachteten System. Um diese zu automatisieren, schrieben Dan Farmer und Witse Venema ein Tool namens „Security Analysis Tool for Auditing Networks“, kurz SATAN. Dieses Tool stellte einen Prototyp dar und gilt daher als Urvater der Vulnerability Scanner.

Die Scanner können nach [SP800-6:11] passiv und aktiv sein. Passive Scanner schließen aus Eigenschaften, die im aktuellen Zustand des Systems vorhanden sind, auf die Anwesenheit einer Schwachstelle. Aktive Scanner finden Schwachstellen durch deren Ausnutzung, wodurch sie die Gefahr eines Schadens bergen. Würde sich ein aktiver Scanner selbstständig verbreiten können, so würde er die Eigenschaften eines Wurms aufzeigen (vgl. [SP800-6:12]).

Scanner können zudem Host-basiert oder Netzwerk-basiert sein. Host-basierte Scanner können nur ihr eigenes System untersuchen. Netzwerk-basierte Scanner hingegen testen ein System über die Netzwerkverbindungen auf Schwachstellen.

Heutige Scanner sind überwiegend passiv. Lediglich die Anwesenheit mancher Denial-of-Service- oder Passwortschwächen werden durch aktive Tests geprüft. Solche Schwächen können nicht durch passive Tests nachgewiesen werden, da beispielsweise die Passwörter verschlüsselt auf dem Host gespeichert sind.

3.5.2. Anwendung des Scanners

Vulnerability Scanner können bei der Analyse von Schwachstellen in Systemen helfen. Zu diesem Zweck können sie in einer Schwachstellenanalyse oder der Ebene Assurance I der oben dargestellten Revision eingesetzt werden.

Manche Autoren vermischen die Begriffe Vulnerability Scanner und Penetrationstest. Dabei wird ausgehend von der Argumentation, dass der Penetrationstest eine Schwachstellenanalyse sei, häufig auch argumentiert, ein Penetrationstest könne durch das reine Ausführen eines Vulnerability Scanners durchgeführt werden. In der Zeit vom 7. März bis 10.März 2003 wurde auf der Mailing Liste „pen-test“ bei Securityfocus.com das Thema „Penetration Testing or Vulnerability Scanning?“ besprochen, wobei Rizwan Ali Khan um eine Hilfestellung bei der Abgrenzung der beiden Begriffe bat.

Der Argumentation stehen zwei Überlegungen gegenüber. Zum einen wird die Argumentation in [Kurtz00a:1] als Mythos bezeichnet, da ein Vulnerability Scanner nicht von einem Angreifer verwendet würde. Ein solcher Scanner würde auf Grund des großen Verkehrs, den er erzeugt, sehr schnell erkannt werden. Eine ähnliche Begründung findet sich auch unter [RUS-CERT]. Zudem ist der Penetrationstest ein zielgerichteter Angriff, der nicht alle Schwachstellen erkennen soll, und somit keine Schwachstellenanalyse darstellt.

Die pure Anwendung eines Vulnerability Scanners ist somit kein Penetrationstest. Hinzu kommt, dass ein Penetrationstest durch die kontrollierte Durchführung von Angriffen charakterisiert ist. Da die Scanner zur Automatisierung von Vorgängen geschaffen worden sind, könnte ein Penetrationstest nur mit aktiven Vulnerability Scannern erreicht werden. Ein solcher Scanner würde aber unter den allgemeinen Problemen der Scanner leiden, die im Folgenden besprochen werden.

3.5.3. Probleme der Vulnerability Scanner

Die zwei wesentlichen Probleme der Qualität eines Vulnerability Scanners sind die fehlende Vollständigkeit und die Meldung von falschen Positivergebnissen (engl.: false positives), die zu einem fehlerhaften Ergebnisbericht führen.

Die passiven Scanner haben Signaturen, die mit dem aktuellen Zustand des zu prüfenden Systems abgeglichen werden. Daher ist ein Vulnerability Scanner wie ein Virenscanner ständig zu aktualisieren. Dennoch besitzen die meisten Scanner nicht für alle Schwachstellen eine Signatur, die zudem fehlerhaft sein kann. So können unvollständige Ergebnisse entstehen. In einem Test (vgl. [Forristal01]) fand keiner der betrachteten Scanner alle präparierten Schwachstellen.

Zudem kann ein Vulnerability Scanner falsche Positivergebnisse erzeugen, wobei er eine Schwachstelle meldet, die nicht in dem System vorhanden ist. Dies ist unter anderem in den Signaturen begründet. So bauen die meisten Scanner eine Verbindung zu einem Dienst auf und analysieren anhand des Banners, der Informationen über die installierte Software und deren Version enthält, welche Schwachstellen auf die Software des Systems zutreffen. Ob die Software wirklich installiert ist, prüft der Scanner nicht. Damit kann der Scanner durch ein manipuliertes Banner leicht getäuscht werden.

3.6. Parameter des Penetrationstests

In der Literatur wird der Penetrationstest unterschiedlich abgegrenzt. So vertritt Winkler (vgl. [Winkler00:2]) die Ansicht, dass ein Penetrationstest immer verdeckt erfolgt. Nach Payne (vgl. [Payne01:2]) hingegen kann ein Penetrationstest offen als auch verdeckt erfolgen. Um eine einheitliche Sichtweise entwickeln zu können, werden sämtliche Varianten durch die Parameter des Tests beschrieben. Dazu werden im Folgenden fünf Parameter des Penetrationstests vorgestellt.

Erster Parameter ist die Bekanntmachung (engl: notice). Er beschreibt, wem innerhalb einer Organisation die Durchführung des Penetrationstests bekannt ist. Die Bekanntmachung kann dabei die Werte covert oder overt (vgl. [Payne01:2]) annehmen. Hierbei bedeutet

- covert (deut.: verdeckt), dass die Betroffenen keine Kenntnis von dem Test haben, und
- overt (deut.: offenkundig), dass der Test den Betroffenen bekannt ist.

Die verdeckte Durchführung des Tests wird nach [SP800-42:16f] als Red Teaming bezeichnet, während der offenkundige Fall als Blue Teaming bezeichnet wird.

Zweiter Parameter ist das Wissen (engl.: knowledge). Das Wissen beschreibt die Menge der Informationen, die der Tester über das Testobjekt besitzt. Dabei wird in der Literatur zwischen den beiden Extremen zero knowledge und full knowledge unterschieden, wobei

- zero knowledge keine Kenntnis des Testers über das Testobjekt, und
- full knowledge die Kenntnis des Testers über sämtliche Details des Testobjektes bedeutet.

Um zu vermeiden, dass ein Tiger Team auf Grund fehlender Informationen das gewünschte Ziel nicht erreicht, sind auch Abstufungen möglich. So kann beispielsweise das zu testende Subnetz oder die Existenz eines bestimmten Servers vorgegeben werden.

Dritter Parameter ist der Standort (engl.: location) des Angreifers. Der Standort wird als

- internal bezeichnet, wenn der Angreifer von innerhalb des Netzes operiert, und als
- external bezeichnet, wenn der Angreifer von außerhalb des Netzes operiert.

Der vierte Parameter ist die Auswirkung (engl.: impact) des Penetrationstests auf das Testobjekt. Die Auswirkung stellt das Potential des Schadens dar, das durch den Penetrationstest erzeugt werden kann. In [Robinson03] wird die Auswirkung in drei Stufen unterteilt, wobei die erste Stufe nur eine Reconnaissance erlaubt, die zweite Stufe die Denial-of-Service Angriffe ausschließt und die dritte Stufe alle Angriffe erlaubt. Da in der zweiten Stufe nicht zwischen weiteren Angriffarten unterschieden werden kann, soll in dieser Arbeit die Auswirkung durch die Möglichkeit des Verlustes der Sicherheitsfaktoren Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität der Assets beschrieben werden.

Fünfter Parameter ist die Abdeckung (engl.: coverage). Die Abdeckung beschreibt die Menge der Assets, die von dem Test betroffen sind. Aus Gründen der Risikoeinschätzung der aus einem Penetrationstest resultierenden Schäden kann neben der Beschränkung der Einwirkung auch die Menge der einbezogenen Assets eingeschränkt werden. Beispielsweise wird nur der Perimeter in den Test einbezogen. Nur wenn alle Systeme in den Test einbezogen werden, kann von einer vollen Abdeckung (engl.: full coverage) gesprochen werden. Nach [Robinson03] kann die Abdeckung eines Netzwerkes auch durch die Aufteilung in Horizonte beschrieben werden. Eine solche Aufteilung ist in Abbildung 11 dargestellt.

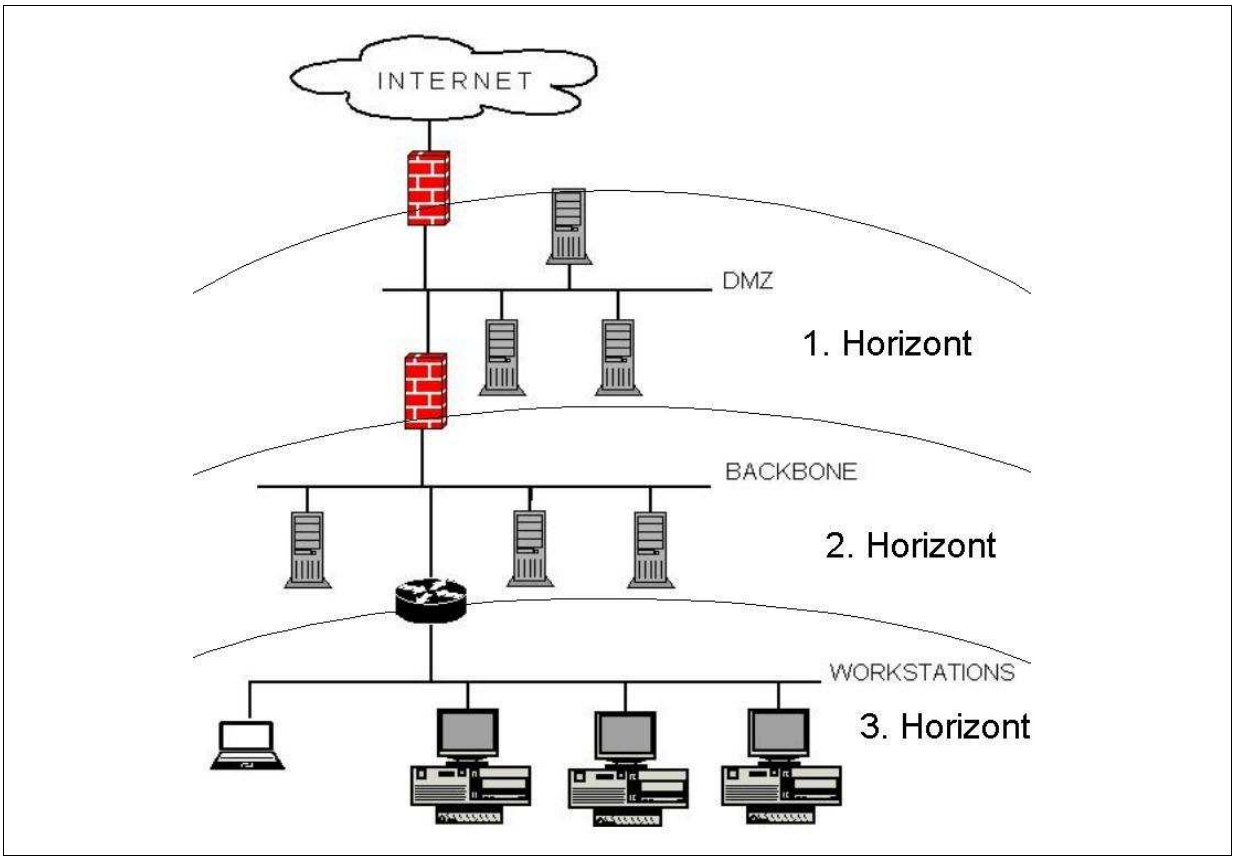


Abbildung 11: Ein in Horizonte unterteiltes Netzwerk

Neben den Unterscheidungen des Penetrationstests nach Parametern, werden sie in der Literatur auch auf Grund der Sichtweise auf das Testobjekt in black-box und white-box unterschieden. Diese Unterscheidung entspricht eine Kombination von Standort und Wissen (vgl. [Robinson03] und [GSHB02:Kapitel 2.5]), wobei

- ein black-box Test einem external, zero-knowledge Test entspricht und
- ein white-box Test einem internal, full-knowledge Test entspricht.

Von der Verwendung der Begriffe black-box und white-box wird in dieser Arbeit abgesehen, da die Begriffe auch mit den oben vorgestellten Parametern beschrieben werden können.

Die genaue Parametrisierung des Tests muss zu Beginn eines Penetrationstests mit dem Initiator des Penetrationstests abgesprochen und festgehalten werden.

3.7. Das Tiger Team

Der Begriff Tiger Team stammt vom Militär, wobei dort Mitglieder der eigenen Truppe versucht haben, die Sicherheitsvorkehrungen einer Militärbasis zu umgehen oder zu durchbrechen (vgl. [Wosnack01:1]). Im Sinne der Informationstechnologie werden als Tiger Team analog diejenigen Personen verstanden, die im Auftrag des Besitzers eines Systems oder Netzwerks versuchen, in dieses einzubrechen (vgl. [Wosnack01:1]). Ein Tiger Team ist

somit die Gruppe von Personen, die einen Penetrationstest unabhängig vom Anwendungsgebiet durchführen. Daher werden nicht nur diejenigen Gruppen als Tiger Team bezeichnet, die von außen in ein Netzwerk eindringen, sondern auch solche Gruppen, die im Anwendungsgebiet der Softwareentwicklung in neue Systeme eindringen, um potentielle Schwachstellen in neuer Software zu entdecken (vgl. [Wosnack01:1]).

Auf Grund des Begriffs Tiger Team wird der Penetrationstest auch als Tiger-Team-Analyse bezeichnet (vgl. [Pfleeger00:309]). Bei der Zusammensetzung und Auswahl des Teams sind aber einige Gesichtspunkte zu beachten.

Bei den Mitgliedern des Tiger Teams sollte es sich nicht um Hacker handeln. So kann das Tiger Team vertrauliche Daten oder Informationen über das Unternehmen erlangen, die von Hackern zu ihrem Zweck verwendet und in ihren Kreisen verbreitet werden können. In einem solchen Fall entsteht für den Initiator ein immenser Schaden. Zudem muss ein Tiger Team ethische und rechtliche Bedenken berücksichtigen, was weder in der Natur eines blackhat noch in der eines whitehat Hackers liegt.

Auch die Qualität der Ergebnisse ist abhängig von den Erfahrung der Mitglieder eines Tiger Teams (vgl. Abschnitt 5.3.5). Daher sollte ein Tiger Team nach Kurtz (vgl. [Kurtz00:2]) nur aus Personen bestehen, die über Ausbildung, Fähigkeiten und Erfahrung in den Bereichen IT-Sicherheit und Netzwerktechnologie verfügen. Sie ist notwendig, um durch eine strukturierte Vorgehensweise qualifizierte Aussagen über die Sicherheit eines Betrachtungsobjekts treffen zu können. Ein Hacker würde nur diejenigen Angriffe, die er in seinem Umfeld gesammelt hat, betrachten. Eine strukturierte, auf einer Risikoanalyse basierende Betrachtung aller in Frage kommenden Angriffe ist von ihm nicht zu erwarten. Daher sollte ein Tiger Team nur aus Sicherheitsexperten mit entsprechender Ausbildung bestehen.

Zusätzliche Gesichtspunkte bei der Auswahl eines Tiger Teams als Vertragspartner für einen Penetrationstest werden im Rahmen des Service Level Agreements, das einen Vertrag zwischen Initiator und Tiger Team darstellt, in Abschnitt 5.2 betrachtet. Zunächst werden Angriffe vertieft, die das wesentliche Merkmal eines Penetrationstests sind.

4. KAPITEL:

ANGRIFFE

Wesentliches Merkmal eines Penetrationstests ist die Durchführung von Angriffen, um daraus Aussagen über die Sicherheit eines Kontextes zu machen. In diesem Kapitel soll zunächst der Begriff *Angriff* bestimmt und an einigen Beispielen näher betrachtet werden. Darüber hinaus ist die Vorgehensweise eines Angreifers zu analysieren, um daraus im darauffolgenden Kapitel eine Vorgehensweise für einen Penetrationstest herzuleiten.

4.1. Begriffbestimmung eines Angriffes

Häufig wird von „Denial-of-Service-Angriffen“, „Flooding-Angriffen“ oder „Buffer-Overflow-Attacken“ gesprochen. Dabei wird deutlich, dass ein Angriff nach unterschiedlichen Gesichtspunkten klassifiziert sein kann. So ist ein Denial-of-Service das Ergebnis (engl.: result) eines Angriffes, während Flooding eine Aktion (engl.: action) ist, die zum Durchführen des Angriffes genutzt wird. Versuche, diese Angriffe in Klassen einzuordnen, bringen Schwierigkeiten zum Vorschein. Cohen (vgl. [Howard98:3]) sagt dazu: „…a complete list of the things that can go wrong with information systems is impossible to create.”

Diesem Problem soll in dieser Arbeit mit einem Modell aus [Howard98] begegnet werden, wobei ein Angriff (engl.: Attack) als „a series of steps taken by an attacker to achieve an unauthorized result.” (vgl.: [Howard98:12]) definiert ist. Das unautorisierte Ergebnis ist dabei aus der Sicht des Besitzers eines Assets zu sehen. Was autorisiert ist, ist in der IT-Security Policy festgehalten. Daher kann ein Angriff auch als ein Verstoß (engl.: violation) gegen die Policy gesehen werden.

Zudem werden Angriffe, die den Datenstrom nicht verändern, als passive Angriffe bezeichnet. Aktive Angriffe hingegen verändern den Datenstrom durch Einfügen, Löschen oder Veränderung der Nachrichten. Aktive Angriffe können dabei von Intrusion Detection Systemen oder durch Analyse des Audit Trails erkannt werden.

Die Abbildung 12 stellt einen Angriffsvektor nach [Howard98] bestehend aus fünf Elementen dar. Kern des Angriffes ist ein Ereignis (engl.: Event), unter dem eine gegen ein bestimmtes Ziel (engl: target) gerichtete Aktion verstanden wird. Eine Aktion entsteht dann, wenn eine Schwachstelle (engl: vulnerability) durch ein Tool ausgenutzt wird. Diese Schritte, die in Abbildung 12 dargestellt sind, führen zum Erreichen eines unautorisierten Ergebnisses (engl.: unauthorized result).

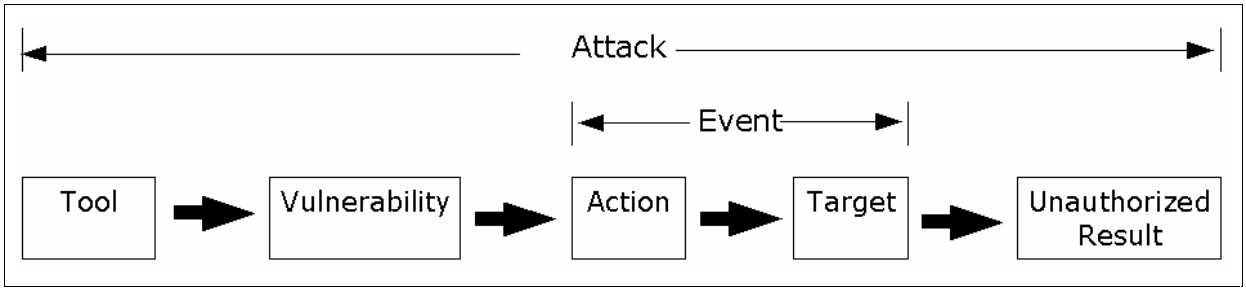


Abbildung 12: Elemente eines Angriffes nach [Howard98:12]

Ein Tool bedeutet nach [Howard98:13] den Weg der Ausnutzung einer Schwachstelle. Dieser Weg kann durch eine Methode zur Ausnutzung der Schwachstelle beschrieben werden, was der Definition eines Exploits entspricht (vgl. Seite 14). Ein Angreifer kann aber beispielsweise einen Wurm oder eine andere Art von Malware verwenden, um eine Aktion gegen ein Ziel durchzuführen. Daher wird statt des Begriffs Tool der Begriff Critter verwendet, der nach Kossakowski (vgl. [Kossakowski01:421f.]) den Oberbegriff von Exploit und Malware darstellt. Mit diesem Begriff können zudem sämtliche Tools bezeichnet werden, „die Personen einsetzen, um unbefugt in Systeme einzubrechen oder sich höhere Berechtigungsstufen zu verschaffen“ [Kossakowski01:421f.]). Daher ist in dieser Arbeit das erste Element des Angriffsvektors der Critter. Der in dieser Arbeit verwendete Angriffsvektor ist in Abbildung 13 dargestellt.

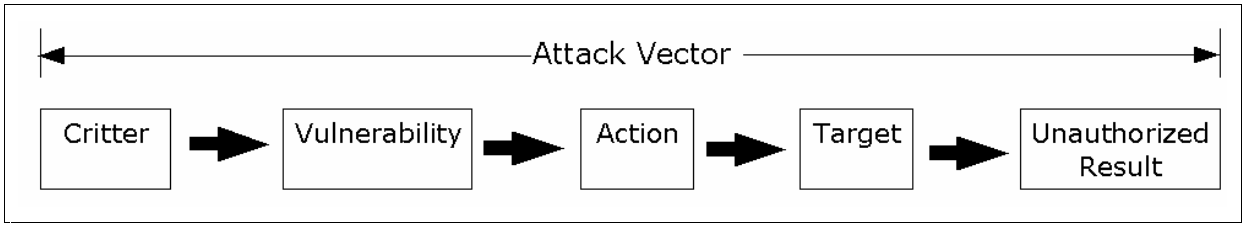


Abbildung 13: Ein Angriffsvektor für die Verwendung in dieser Arbeit

Welche Tools ein Angreifer neben eventuell automatisierten Exploits verwendet, kann dabei vernachlässigt werden. So kann der Angreifer, nachdem er sich auf einen Exploit und eine Schwachstelle festgelegt hat, einen Port-Scanner einsetzen, um die Anwendbarkeit des Exploits auf einem bestimmten Zielsystem festzustellen.

Ein Angreifer muss sein Ziel nicht erreicht haben, wenn er ein unautorisiertes Ergebnis erzielt hat. So kann er nach erfolgtem Zugriff auf ein System die Absicht haben, dort vorgefundene Daten zu zerstören. Zudem kann er eine Hintertür installieren, durch die er mit geringem Aufwand erneut Zugriff auf das System erlangen kann. Diese Tatsachen erfordern die Betrachtung der Vorgehensweise eines Angreifers.

4.2. Vorgehensweise eines Angreifers

Um sein Ziel zu erreichen, muss ein Angreifer eine Vorgehensweise entwickeln. Diese Vorgehensweise kann anhand von Stationen beschrieben werden, die ein Angreifer auf dem Weg zu seinem Ziel passiert.

Ein Beispiel für eine solche Vorgehensweise ist der von [Cole02:23] vorgeschlagene „Attacker's Process". Er besteht aus den folgenden sieben Stationen:

1. passive reconnaisance
2. active reconnaisance
3. exploiting the system
4. uploading programs
5. downloading data
6. keeping access by using backdoors and Trojan horses
7. covering tracks

Schneier [Schneier00:275] vereinfacht einen solchen Process durch „figuring out what to attack, figuring out how to attack, getting in, performing the attack, and getting out".

Um zunächst zu erkennen, was angegriffen werden soll, führt der Angreifer eine Reconnaissance durch. Unter Reconnaissance wird nach [Duden-Oxford90:590] die militärische Aufklärung bzw. Erkundung verstanden. Eine genauere Betrachtung der Reconnaissance erfolgt in Abschnitt 5.3.

Hat sich der Angreifer ein Bild über sein Ziel gemacht, so informiert er sich über bekannte Schwachstellen, die in dem als Ziel gewählten System vorhanden sein können. Durch die Identifikation der Schwachstellen hat er einen Startpunkt für den Angriff erlangt. Diese Station der Vulnerability Detection wird in Abschnitt 5.4 verfeinert.

In der nächsten Station nutzt der Angreifer Tools und sein Wissen, um eine gefundene Schwachstelle auszunutzen. Bei der Ausnutzung der Schwachstelle führt der Angreifer eine Aktion gegen das Ziel durch und erzeugt somit ein unautorisiertes Ergebnis. Diese Station ist die eigentliche Durchführung des Angriffes. Dabei erfolgt ein Eindringen in ein System, was auch mit dem Begriff Penetration bezeichnet wird. Dazu gehören auch Denial-of-Service Angriffe, deren Pakete in ein System eindringen. Die Station der Penetration wird in Abschnitt 5.5 verfeinert.

Mit einer Penetration hat der Angreifer sein Ziel nicht immer erreicht. Hat ein Angreifer ein unautorisiertes Ergebnis erreicht, so kann sein nächstes Ziel ein weiteres unautorisiertes Ergebnis sein. Dadurch breitet sich ein Angreifer weiter aus. Diese Ausbreitung wird nach [Steward99:3] in Anlehnung an die Ausbreitung von Metastasen bei Krebskrankheiten als „Distributed Metastasis" bezeichnet. Die Metastase ist dabei in zwei Komponenten unterteilt. Die erste Komponente ist die Consolidation, bei der ein Angreifer seinen Standpunkt festigt. Dazu versucht er, alle Anzeichen auf den Einbruch sowie seiner Anwesenheit zu vernichten und durch Hintertüren den erneuten Zugang zu dem System zu erleichtern. In der Continuation, der zweiten Phase der Metastase, versucht der Angreifer, Prozesse und Systeme anzugreifen, wodurch der Angriff einen iterativen Verlauf annimmt.

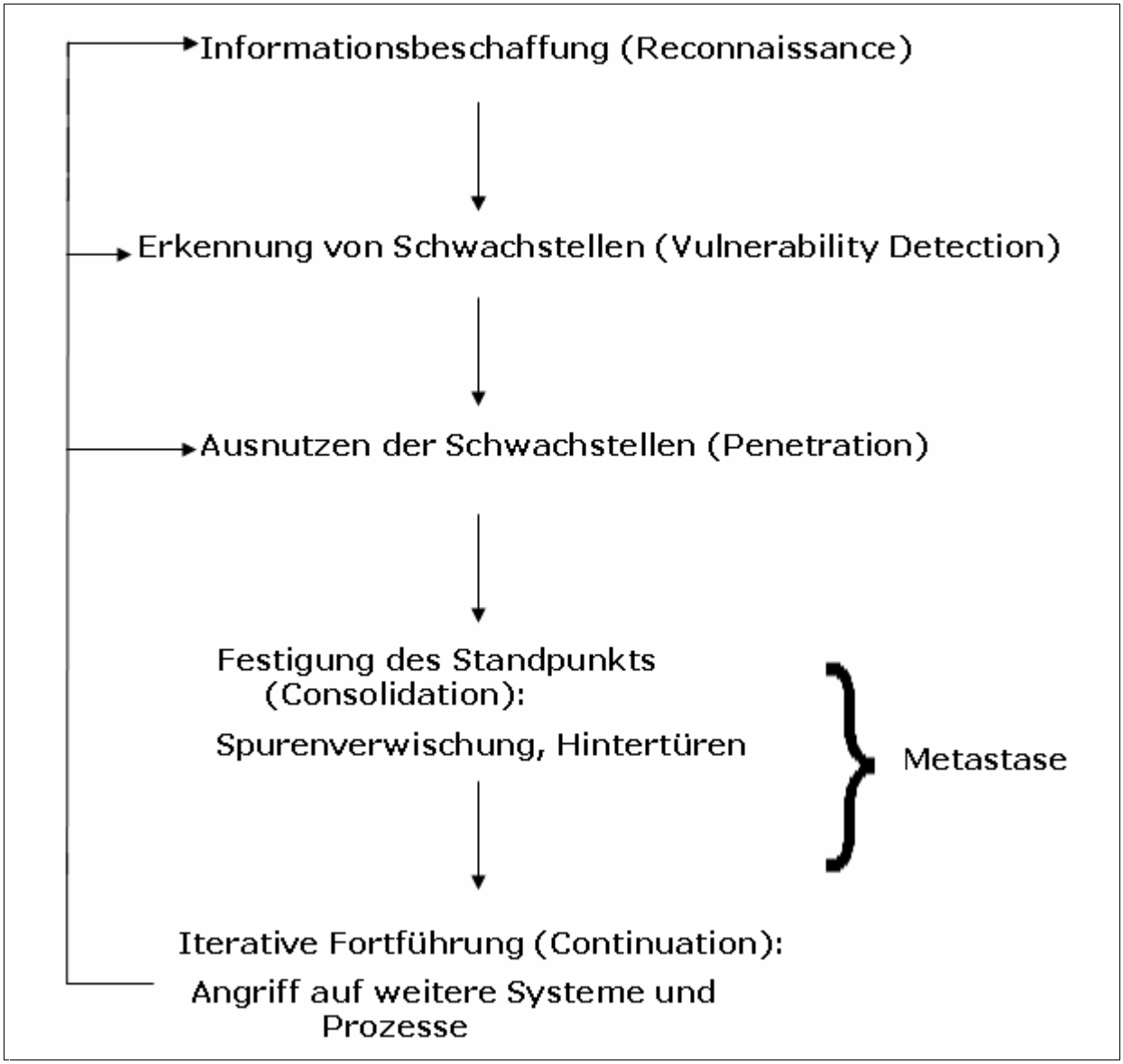


Abbildung 14: Iterative Vorgehensweise eines Angreifers

Abhängig von seinem Ziel kann ein Angreifer während seines Vorgehens versuchen, möglichst hohe Rechte zu erlangen. Die schrittweise Erhöhung der Rechte ist dabei eine Eskalation der Rechte (engl.: escalation of priviledges) und nach Steward (vgl. [Steward99:4]) Teil der Consolidation. Da sie aber einen erweiterten Zugriff bedeutet, ist die Eskalation ein iterativer Schritt eines Angriffes.

Aus der Abbildung 14 wird ersichtlich, dass der Angreifer nicht sequentiell vorgeht, sondern ein Angriff ein iterativer Prozess ist. In den nun folgenden Abschnitten wird auf einzelne Elemente eines Angriffes näher eingegangen, bevor die Vorgehensweise, angewendet auf einen Penetrationstest, im nächsten Kapitel verfeinert wird.

4.3. Schwachstellen

Eine Schwachstelle (engl.: vulnerability) ist nach Pfleeger (vgl. [Pfleeger00:3]) eine Schwäche in einem System, die zu einem Schaden führt. Die Schwäche (engl.: weakness) wird durch Fehler, in Konzept, Implementation und Konfiguration verursacht, wobei unter Konfiguration die Installation und Wartung der Systeme zu verstehen ist. Zusätzliche Ursache für eine Schwachstelle ist der Mensch. Nach dieser Differenzierung werden in diesem Abschnitt Schwachstellen anhand von Beispielen vorgestellt.

4.3.1. Designfehler

Von Neumann-Konzept

Die heute zum Einsatz kommenden Computer beruhen alle auf dem in den 1940er Jahren vorgestellten von-Neumann-Konzept. In diesem Konzept werden Daten und Programmcode in einem gemeinsamen Speicher gehalten. Dadurch können Datenbereiche auch Programmcode enthalten, der zu maliziösen Zwecken ausgeführt wird. Diese Schwachstelle wird bei der Methode des Buffer Overflows ausgenutzt.

Designfehler in den TCP/IP Protokollen

Der TCP/IP-Protokollstapel ist in den 1960er vom US-amerikanischen Militär für den Betrieb des ARPANETs entwickelt worden. Zu diesem Netz hatten nur Militärs und Wissenschaftler Zugang. Da diesen Personenkreisen voll vertraut wurde und die Rechenkapazität der damaligen Rechner sehr gering war, beschränkte sich der Entwurf der Protokolle auf die Funktionalität. Bei den heutigen Anwendern des aus dem ARPANET entstandenen Internets ist das Vertrauen nicht mehr gegeben. Durch fehlende Sicherheitseigenschaften der Protokolle können maliziöse Benutzer Schwachstellen der Protokolle für Angriffe gegen andere Benutzer ausnutzen.

4.3.2. Implementationsfehler

Unchecked Buffer: Implementationsfehler in C/C++

Eingaben werden in einen Speicherbereich geschrieben, der Puffer (engl.: buffer) genannt wird. Zum heutigen Zeitpunkt werden die meisten Programme in der Programmiersprache C[29] implementiert. Hierbei wird meist weder vom Compiler noch vom Programmierer geprüft, ob der Puffer bereits vollständig belegt ist. So können mehr Daten in den Puffer geschrieben werden, als dieser eigentlich fassen kann und es kommt zu einem Pufferüberlauf (engl.: Buffer overflow), der eine Angriffsmethode darstellt.

[29] Mit den durch C++ eingeführten Methoden wie String-Klassen könnte das Problem eingeschränkt werden. Bestehende Programme werden aber aus historischen Gründen weiter in reinem C geschrieben. Zudem ist C++ abwärtskompatibel zu C. Daher werden häufig anfällige C Konzepte in C++ Programmen weitergenutzt.

Microsoft Bob Passwort Schwäche

In Microsoft Bob (vgl. [Epstein95]) bestand eine Schwachstelle, bei der das Programm den Benutzer nach dreimaliger Fehleingabe des Passworts fragt, ob sein Passwort zurückgesetzt werden solle. Mit einem Klick auf „Yes“ konnte jedes Passwort auf diesem Weg geändert werden.

4.3.3. Konfigurationsfehler

Authentifizierung

Auch Fehler bei der Konfiguration der Authentifizierungsmethoden können zu Schwachstellen führen. So stellen fehlende Authentifizierung, auch Null-Sessions genannt, eine große Schwachstelle dar, wenn dadurch ein Zugriff auf schutzwürdige Assets erlangt werden kann.

Ist eine Authentifizierungsmethode vorhanden, so kann das Authentifizierungsmerkmal eine Schwachstelle darstellen. So reicht die IP-Adresse als Merkmal nicht aus. Das Vertrauen, das einem Host durch die Authentifizierung mit seiner IP-Adresse entgegengebracht wird, kann durch Spoofing leicht missbraucht werden. Auch Passwörter können ein großes Problem darstellen. Wegen des Ausmaßes der Schwachstelle werden die Passwörter in einem separaten Abschnitt behandelt.

Passwörter

Passwörter können Schwachstellen hervorrufen. So bietet ein leeres Passwort keinen Schutz. Aber auch andere Passwörter bilden eine Schwachstelle, da sie leicht erraten werden können. Um es sich leichter merken zu können, nehmen Menschen Wörter wie den Vornamen des Lebenspartners, die Angreifer aus dem Umfeld des Benutzers leicht erraten können. Aber auch andere Menschen, die keine sozialen Kontakte zu einem Benutzer haben, können solche Passwörter leicht erraten, da sie aus einem Wörterbuch stammen. Auch die Länge eines Passwortes ist relevant. Mit der Länge des Passwortes steigt der Aufwand zum Erraten des Passwortes exponentiell. Lange Passworte, die nicht in einem Wörterbuch vorkommen, sind kaum in endlicher Zeit zu erraten.

Nutzlose Dienste

Bei einer voreingestellten Installation eines Betriebssystems werden meist Dienste installiert und gestartet, die für den Benutzer nutzlos sind, aber dennoch einen Zugriff auf ein System erlauben. Da sie Schwachstellen enthalten, stellen sie ein Risiko dar, das auf Grund mangelnden Nutzens unnötig ist. Somit stellt ein solcher Dienst im Falle der Aktivierung einen Fehler dar, der zu einem Schaden führt, wodurch nutzlose Dienste eine Schwachstelle in einem System darstellen.

Falsche Rechte von Prozessen oder Nutzern.

Nutzer und Prozesse sollten nach dem least privilege Prinzip nur Rechte erhalten, die sie benötigen. Kann zum Beispiel ein Webserver Prozess auf einen Systemordner zugreifen, so kann das Ausnutzen einer Schwachstelle im Webserver einen signifikanten Schaden im gesamten System verursachen.

4.3.4. Der Mensch

Fehlbedienung

Ein Benutzer macht Fehlbedienung, aus denen ein Schaden entstehen kann. Um dieser Schwachstelle zu begegnen, wird in vielen Systemen daran gearbeitet, die Bedienung zu vereinfachen, was auch im Interesse der Nutzer ist. Dabei werden die Systeme allerdings komplexer. Hierdurch entstehen mehr Fehler und somit mehr Schwachstellen, wodurch das Risiko an anderer Stelle vergrößert wird.

Leichtgläubigkeit

Die Leichtgläubigkeit eines Menschen ist eine nicht zu verachtende Schwachstelle. Sie ist Grundlage der Social Engineering Angriffe und der Verbreitung von Malware. So können Menschen leicht zu einem Klick auf eine infizierte und ausführbare Datei verleitet werden, wenn ihnen eine Liebesbotschaft oder der Gewinn eines Handys suggeriert wird.

4.4. Ausgewählte Angriffsaktionen

In diesem Abschnitt werden Aktionen des Angriffes näher erläutert. Manche dieser Aktionen sind nicht alleine anwendbar, sondern führen nur in Kombination zu einem Ergebnis.

Die *Reconnaissance* bezeichnet die Aufklärung im militärischen Sinne. Im informatischen Sinne ist darunter die generelle Sammlung von Informationen durch einen Angreifer über ein Ziel zu verstehen, die noch keinen Schaden bewirkt. Nach [Pfleeger03:7-30] kann sie aber als Vorbote einer Bedrohung gesehen werden. Die Reconnaissance ist dabei keine direkte Angriffsmethode, sondern ein Oberbegriff für die Verkettung der Aktionen Open Information Gathering, Social Engineering, Scanning, Probing, Fingerprint und Sniffing.

Das *Open Information Gathering* beinhaltet eine Reihe von passiven Angriffen, wobei der Angreifer Informationen über sein Ziel erlangen will. Dabei durchsucht er Artikel im Internet oder Zeitschriften und wertet Anfragen an whois-Dienste und DNS-Server aus. Auch das so genannte *Dumbster Diving* gehört zu dieser Angriffsmethode, wobei ein Angreifer die Müllbehälter nach Dokumenten mit Informationen durchsucht.

Social Engineering ist eine Methode, bei der ein Angreifer Menschen durch soziale Fertigkeiten beeinflusst, um so den Angriff durchzuführen. Diese Angriffstechnik kann zum einen bei der Informationsbeschaffung genutzt werden, wobei ein Angreifer Benutzer anruft und sich als Mitarbeiter des Helpdesk ausgibt, um nach Informationen zu fragen. Zum anderen kann er die Benutzer auch dazu bringen, wichtige Dateien zu löschen, oder anderweitig einen Schaden zu erzeugen. Dabei kann der soziale Kontakt nicht nur per Telefon, sondern auch per Email oder durch Besuche zustande kommen. Eine Abwandlung stellt das Reverse Social Engineering dar, bei dem der Angreifer nach Informationen gefragt wird und so die Fragenden beeinflussen kann (vgl. [Granger01:1]).

Ein *Probe* ist der Zugriff auf ein Ziel, um seine Charakteristika zu bestimmen. Ein *Scan* ist der sequentielle Zugriff auf mehrere Ziele, um festzustellen, welches Ziel spezifizierte Charakteristika aufweist. Als Ziel wird dabei ein System, eine Netzwerk-Entität, oder eine physische Entität verstanden. Ein Zugriff ist dabei eine Kommunikationsbeziehung oder ein physischer Kontakt mit dem Ziel (vgl. [Howard98:8]). Eine Variante des Scans, bei der mehrere Ziele auf Anwesenheit genau einer Charakteristik wie beispielsweise einem Port geprüft werden, wird auch *Sweep* (vgl. [Jamieson01:2]) genannt.

Fingerprint ist eine Methode, um Hersteller, Version und andere Charakteristika eines Betriebssystems oder einer Anwendung zu erhalten. Bei der Erkennung von Client/Server-Applikationen wird ein beim Verbindungsaufbau übermittelter Indentifikationstext ausgewertet. Dieser Text wird Banner genannt, weshalb die Methode der Erkennung einer Applikation auch als *Banner Grabbing* bezeichnet wird.

Unter dem Begriff S*niffing* oder *Eavesdropping* wird das Abhören einer Kommunikation bezeichnet. Dabei können Anzeichen auf vorhandene Systeme oder Zugangsinformationen wie Benutzernamen und Passwörter erlangt werden. Zudem können Informationen erlangt werden, die nicht für den Lauscher bestimmt sind, wodurch die Vertraulichkeit der Dokumente geschädigt wird.

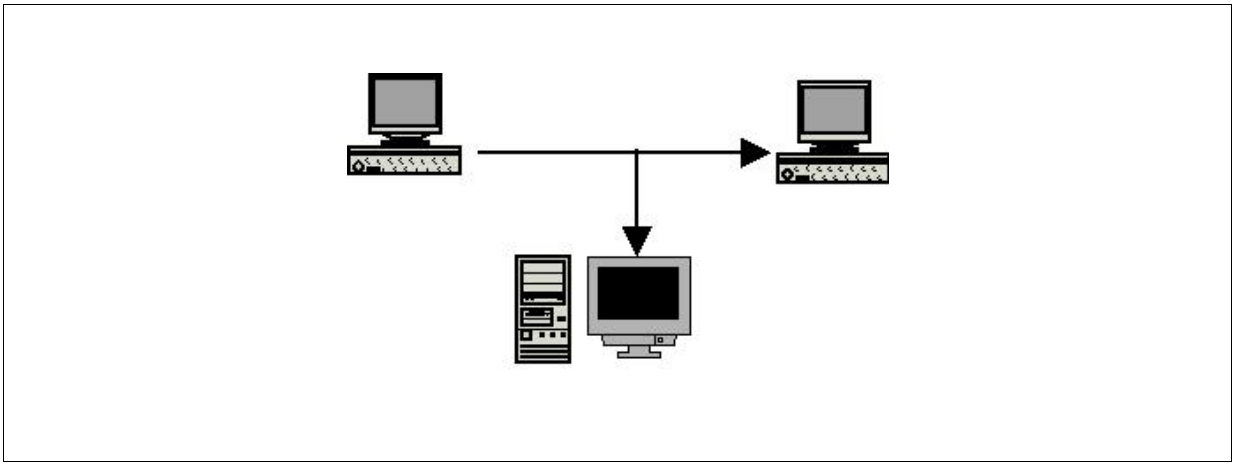


Abbildung 15: Darstellung eines sniffing-Angriffes

Die Ergebnisse des Abhörens einer Kommunikation können in einer *Verkehrsflussanalyse* (engl.: traffic flow analysis) interpretiert werden, die nach [Schneier00:34] eine Studie von Verkehrsmustern ist. Hierzu wird analysiert, wer mit wem, wann, wie lange und in welcher Frequenz kommuniziert. Bei dem Nutzen kommt es auf die Interpretation an. So kann ein Angreifer beispielsweise schließen, welche Systeme eines Netzes von großer Bedeutung sind.

Der *Bypass* beschreibt die Nutzung eines alternativen Weges zu einem Ziel. So können Authentikationsmethoden umgangen werden, in dem durch ein Buffer Overflow über eine Shell ein Zugriff auf eine Systemressource erfolgt. Ein weiteres Beispiel ist die Ausnutzung der Web Folder Traversal Vulnerability, wobei mittels anormaler http-Anfragen an einem IIS-Webserver das System über Kommandozeilenbefehle bedient werden kann.

Ein Puffer ist in Anlehnung an [Nelißen02:4] ein Segment im Speicher, das eine bestimmte Anzahl gleicher Datentypen enthält. Wird bei der Belegung des Puffers die Länge nicht geprüft, so entsteht die Schwachstelle der ungeprüften Pufferlänge, so dass der Puffer mittels einer zu langen Eingabe zum Überlauf gebracht wird. Diese Angriffsmethode wird *Pufferüberlauf* (engl.: *Buffer Overflow*) genannt.

Ein Pufferüberlauf kann einen Absturz des Programms und eventuell des Systems zur Folge haben, wenn das betroffene Programm Teil des Betriebssystems ist oder eng mit diesem verbunden ist. Des Weiteren kann durch einen Pufferüberlauf auch maliziöser Code eingeschleust und mit den Privilegien des betroffenen Programms ausgeführt werden. Diese Gefahr beruht zum einen auf dem Designfehler des von-Neumann Konzeptes, dass Code und Daten in einen Speicher geschrieben werden können, zum anderen auf einem Implementationsfehler. Zum anderen beruht der Pufferüberlauf auf der ungeprüften Pufferlänge, die im Folgenden am Beispiel einer Variante, dem Stack Overflow, genauer erläutert wird.

Obwohl für Code der untere Speicherbereich vorgesehen ist, kann auch in anderen Speicherbereichen wie den Stack, der für Parameter und lokale Variablen einer Funktion genutzt wird, Code abgelegt und ausgeführt werden. Im Falle eines Stack Overflows ist der Puffer ein lokales Array einer Funktion, das beispielsweise für eine Zeichenkette genutzt wird. Bei jedem Aufruf einer Funktion wird ein Stack genannter Speicherbereich reserviert, der nach dem Prinzip eines Stapels (engl.: stack) aufgebaut ist. Der nach dem LIFO-Prinzip

(Last In First Out) geordnete Stack wächst in Richtung der niedrigeren Speicheradressen. Nach der Initialisierung des Stacks werden nach den Parametern des Funktionsaufrufs auch die Return-Adresse sowie der Basepointer der aufrufenden Funktion auf den Stack gelegt. Bei der Return-Adresse handelt es sich um die Adresse, die vor dem Aufruf im CPU-Register Instruction Pointer enthalten war. Sie verweist auf die Adresse der nächsten Instruktion, die nach der Ausführung der Funktion ausgeführt wird. Der Basepointer zeigt auf den Speicherbereich der aufrufenden Funktion. Wird bei der Belegung des Puffers, der gegensätzlich zum Stack in Richtung der hohen Speicheradressen wächst, die Einhaltung der Grenze nicht überprüft, werden zunächst weitere lokale Variablen, dann Basepointer und die Return-Adresse überschrieben. So kann die Return-Adresse auch auf einen höheren Speicherbereich zeigen, in dem der Angreifer ausführbaren Code hinterlassen hat. Problematisch ist allerdings der Umstand, dass die Position der Return-Adresse und des ausführbaren Codes kaum vorhergesehen werden kann. Daher schleust der Angreifer die manipulierte Rückspringadresse mehrfach ein und beginnt den ausführbaren Code mit dem Befehl „No Operation“ (NOP), um Fehler bei der Vorhersage der ersten Codeadresse zu kompensieren. Diese Vorgehensweise wird in Abbildung 16 veranschaulicht. So kann es zu einer Ausführung von beliebigem Code kommen, der auch eine maliziöse Semantik aufweisen kann.

Neben dem hier beschriebenen Stack Overflow ist auch der komplexere Heap Overflow bekannt. Für weitere Details zum Thema Pufferüberlauf wird auf [Neliß en02] verwiesen.

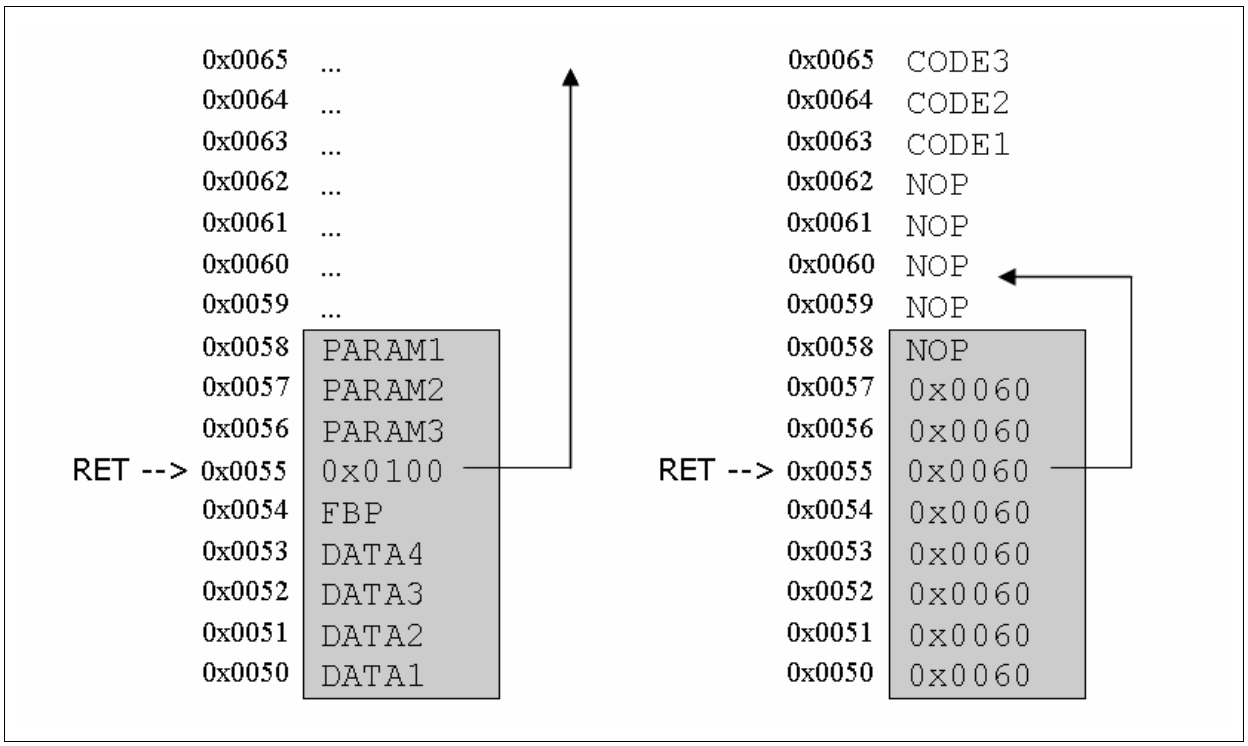


Abbildung 16: Aufbau des Speicherbereiches eine Funktion (grau) bei normaler Programmausführung (links) und nach einem Pufferüberlauf (rechts)

Brute-force bezeichnet die Methode, sämtliche Kombinationen von Zeichenketten zu erzeugen. Diese Methode wird zum Erraten von Passwörtern, Schlüsseln oder Klartexten verwendet.

Spoofing bezeichnet das Fälschen einer Adresse, wobei ein Angreifer sich als jemand anders ausgibt. Pfleeger (vgl. [Pfleeger03:7-36ff.]) unterscheidet dabei zwischen Spoofing, wenn der Angreifer eine andere Netzwerkentität vorgibt, und *Impersonation*, bei der der Angreifer eine Person vortäuscht. Da der Angreifer damit seine Identität verbirgt, wird diese Angriffsmethode auch *Maskerade* genannt.

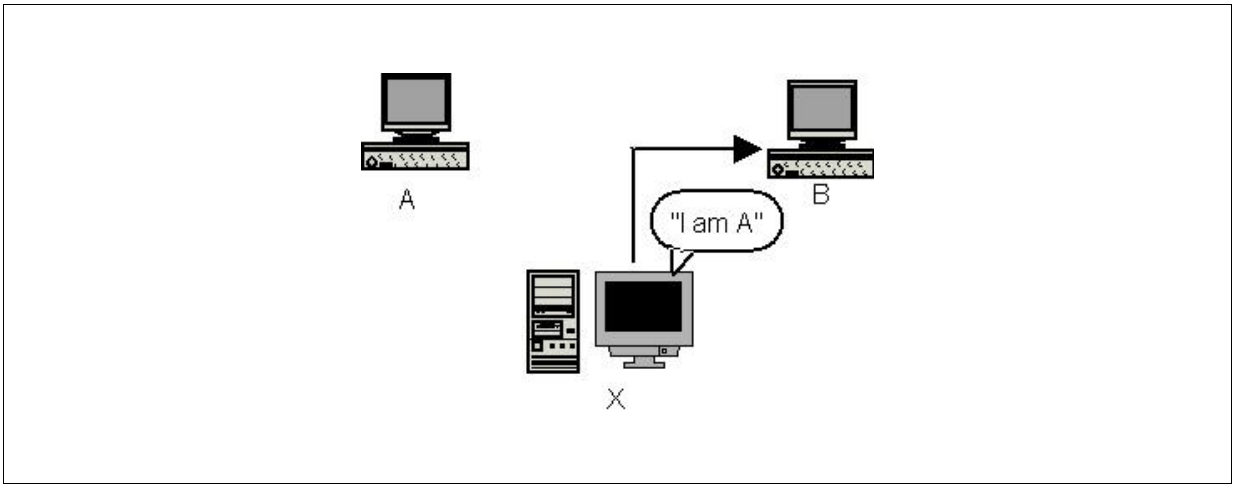


Abbildung 17: Darstellung eines Spoofing-Angriffes

Durch *Tunneling* können ungewünschte Kommunikationen über erwünschte Kommunikationsbeziehungen ausgetauscht werden. So entstehen Informationslecks, über die Informationen nach außen gelangen können.

Als *Hijack* wird die Übernahme einer bereits bestehenden Verbindung bezeichnet. Dabei fängt ein Dritter die Verbindung ab und führt diese mit dem Opfer weiter.

Mit einem *Man-in-the-middle* stellt sich ein Dritter zwischen zwei Kommunikationspartner. Dadurch hat dieser Dritte sämtliche Informationen über die Verbindung und kann dabei die Kommunikation verfälschen. Bei dynamischer Verschlüsselung besteht für den Angreifer sogar die Möglichkeit, verschlüsselte Kommunikationen zu belauschen oder zu verfälschen, da der Angreifer in der Mitte die Schlüsselaushandlung verfolgen kann.

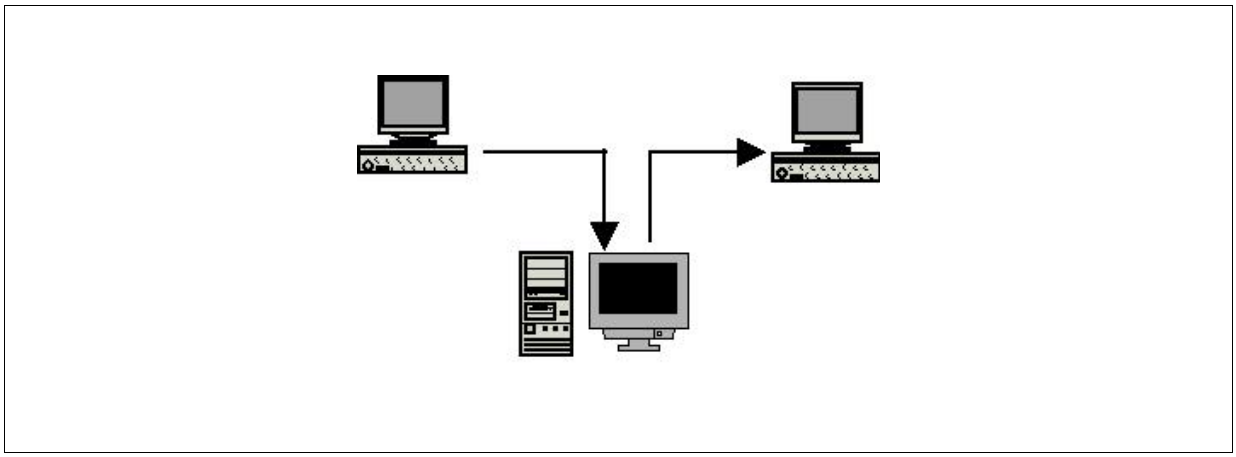


Abbildung 18: Grafische Darstellung eines Man-in-the-Middle Angriffes

Die *Redirection* stellt eine Umleitung einer Kommunikationsbeziehung dar. Hierbei soll erreicht werden, dass eine Kommunikation ein anderes Ziel oder einen anderen Weg nimmt.

Die Aktion *Replay* bezeichnet die Wiederholung gleicher Nachrichten. Dabei benutzt ein Angreifer Nachrichten, die bereits von einem anderen Ort gesendet wurden. Der Angreifer erhofft sich dabei, den Zweck der Nachricht erneut zu erreichen. So könnte er versuchen, eine Authentikation mittels abgehörter Pakete erneut durchzuführen.

Das *Flooding* stellt eine Überflutung einer Kommunikationsbeziehung mit Nachrichten dar. Dabei werden einem System mehr Nachrichten zugesendet, als es verarbeiten kann. Hierbei werden so viele Ressourcen verbraucht, dass der Zugriff auf ein System verzögert wird.

Die Bezeichnung *Salami-Attacke* leitet sich aus der Metapher der Mettwurstscheiben, die als ganzes eine Salami ergeben, ab. Ein Angreifer führt dabei nur Angriffe, die auf Grund ihres geringen Ausmaßes kaum bemerkt werden. Der Wert für den Angreifer ergibt sich aus der Masse der Angriffe. In einer berühmten Anwendung der Salami-Attacke überwies ein Angreifer Rundungsdifferenzen im hundertstel-cent Bereich auf sein Konto (vgl. [Pfleeger00:198]).

Bei der *Parametrisierung* genannten Angriffsmethode kann durch bestimmte Parametrisierungen von Eingaben und Paketen ein unerwartetes Ergebnis hervorgerufen werden. Dabei können auch Eingaben und Pakete außerhalb der Spezifikation erzeugt werden. Beispiele hierfür sind unzulässige Eingaben oder missgebildete Pakete (engl.: malformed packets) bzw. Deformationen.

Methoden zur Durchführung eines Angriffes können nicht nur von digitaler sondern auch von physikalischer Natur sein. So sind auch der *physikalische Einbruch* in ein Rechenzentrum und die *physikalische Zerstörung* eines Rechners Angriffsmethoden. Ein berühmtes Beispiel begab sich an der University of Texas, wo ein Student aus Wut über schlechte Noten mit einer Pistole auf einen Rechner schoß und ein Loch im Kernspeicher erzeugte.

4.5. Ergebnisse des Angriffes

Ergebnisse eines Angriffes sind nach [Howard98:12]

- Erweiterter Zugriff
- Enthüllung von Information
- Korruption von Informationen
- Denial-of-Service
- Diebstahl

Zu einem *erweiterten Zugriff* gehören der Einbruch sowie die Eskalation der Rechte. Durch einen Einbruch in ein System hat er einen Zugriff auf die Ressourcen eines Systems, wodurch der Angreifer seine Zugriffsmöglichkeiten durch den Angriff erweitert hat. Kann ein Angreifer nach seinem Einbruch auf Grund fehlender Rechte nicht auf bestimmte Ressourcen zugreifen, so erweitert er seinen Zugriff durch die Eskalation der Rechte.

Durch die *Enthüllung* (engl.: Disclosure) von Informationen erhält der Angreifer Kenntnis über Informationen, die nicht für ihn bestimmt sind. Diese Informationen kann er weiterverbreiten und so einer Organisation Schaden zufügen.

Die *Korruption* beschreibt die Gesamtheit aus Manipulation und Zerstörung von Informationen. Ein Beispiel für eine Korruption ist beispielsweise die Verunstaltung (engl.: Defacement) von Webseiten. Aber auch die Verfälschung (engl.: Falsification) von Nachrichten, bei der falsche Nachrichten erzeugt oder Nachrichten bei der Übertragung verfälscht werden, ist Teil der Korruption von Informationen.

Denial-of-Service, kurz DoS, bezeichnet die Störung der Verfügbarkeit, durch die der Zugriff auf ein Ziel verzögert oder vollständig unterbunden wird. Ein vollständig unterbundener Zugriff ist an einem System beobachtbar, wenn es als Folge eines Angriffes abstürzt oder neu startet. Bei einem *Distributed Denial-of-Service*, kurz DDoS, wird die Blockierung der Ziels durch einen Strom mehrerer verteilter Angreifer erreicht. Hierzu missbraucht ein Angreifer zunächst mehrere Rechner, indem auf ihnen meist ohne Wissen des Betreibers eine Angriffssoftware installiert wird. Die betroffenen Rechner werden Daemon-hosts genannt, die ein Angreifer von einem Master Host steuern kann (vgl. [Northcutt02a:436ff.]). So kann der Angreifer den Angriff steuern, den die Daemon-Hosts für ihn ausüben.

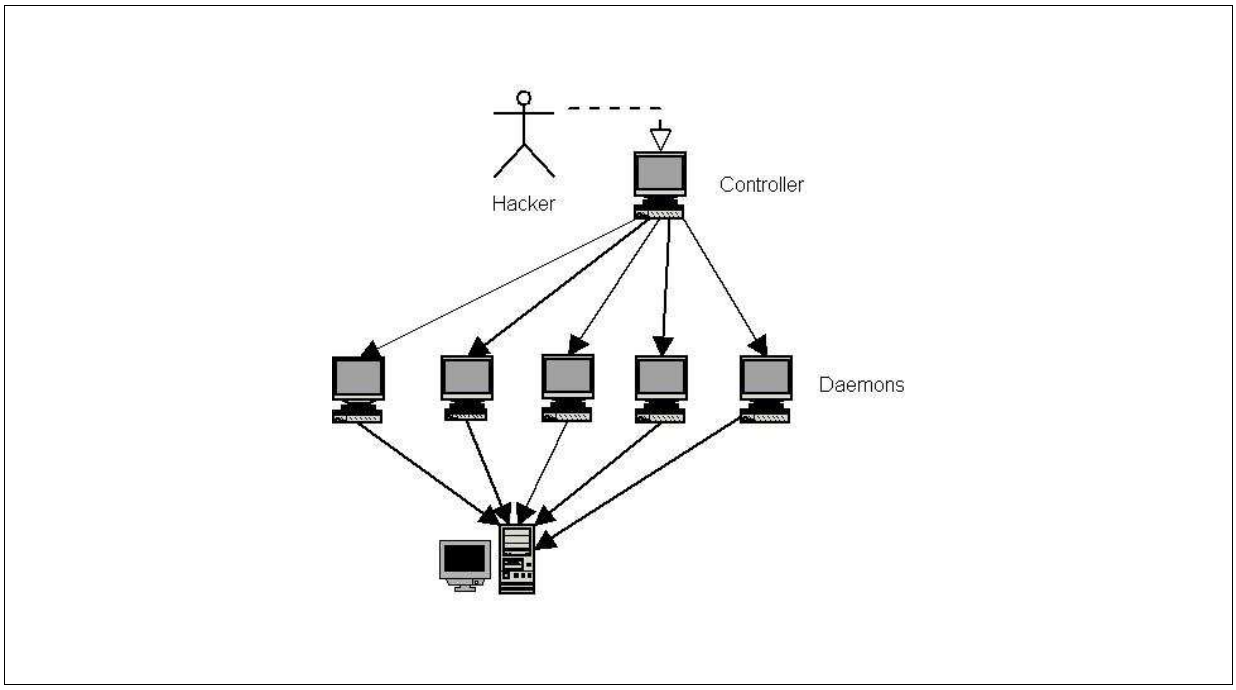


Abbildung 19: Prinzip eines Distributed Denial-of-Service Angriffes

Das Ergebnis eines Angriffes kann auch ein *Diebstahl* sein. Neben Daten können auch andere Ressourcen wie Speicherplatz und Rechenzeit gestohlen werden.

4.6. Risiken in TCP/IP Protokollen

In diesem Abschnitt werden Risiken in ausgewählten Protokollen des TCP/IP Protokollstapels betrachtet. Zuvor wird zum besseren Verständnis ein kurzer Überblick über den TCP/IP Protokollstapel gegeben.

4.6.1. Der TCP/IP Protokollstapel

Der Vorläufer des heutigen Internets ist das in 1960er Jahren entwickelte ARPANET des amerikanischen Verteidigungsministeriums. Bei dessen Entwicklung wurden folgende zwei Ziele verfolgt (vgl. [Tanenbaum98:53]):

1. Das Netz sollte bei Ausfall eines Knotens weiterexistieren können.
2. Das Netz sollte verschiedene Kommunikationskanäle wie Telefon, Satellit oder Computernetzwerke nutzen können.

Auf Grund dieser Ziele wurde der TCP/IP Protokollstapel entworfen, der aus vielen aufeinander aufbauenden Protokollen besteht und beliebig um weitere Protokolle erweitert werden kann. Da für die Beschreibung kein Referenzmodell geschaffen wurde, wird zur Beschreibung eines TCP/IP Netzes das ISO/OSI Referenzmodell gewählt, dessen Schichten in der Tabelle 3 dargestellt sind.

Schicht 7: Anwendungsschicht (engl.: Application Layer) stellt Funktionen zur Kooperation der Anwendungsprozesse bereit
Schicht 6: Darstellungsschicht (engl.: Presentation Layer) behandelt Syntax und Semantik der übertragenden Informationen. Ein Beispiel ist die Kodierung der zu übertragenden Daten in ASCII oder Unicode.
Schicht 5: Sitzungsschicht (engl.: Session Layer) dient der Steuerung der Kommunikation einer Sitzung. Dazu gehören Dialogsteuerung, Token Management und Synchronisation
Schicht 4: Transportschicht (engl.: Transport Layer) dient der Fehlerkontrolle zwischen Endsystemen sowie der Verwaltung von Verbindungen durch Bereitstellung von Funktionen von Verbindungsaufbau und -abbau und Multiplexen mehrerer Kanäle.
Schicht 3: Vermittlungsschicht (engl.: Network layer) ermittelt Routen durch das Vermittlungsnetz und dient der einheitlichen Adressierung.
Schicht 2: Datensicherungsschicht (engl.: Data Link Layer) erstellt Datenrahmen zur Übertragung und sichert die Fehlerfreiheit der Rahmen bei der Übertragung zwischen zwei direkt verbundenen Rechnern.
Schicht 1: Bitübertragungsschicht (engl.: Physical Layer) dient der Übertragung roher Bits über einen Kommunikationskanal.

Tabelle 3: Die Schichten des ISO/OSI-Referenzmodells

Da das ISO/OSI-Referenzmodell erst Anfang der 1980er Jahre entwickelt wurde, bietet es keine exakte Beschreibung des TCP/IP Protokollstapels. So werden in TCP/IP die Aufgaben der Sitzungs- und Darstellungsschicht von der Anwendung übernommen. Zudem existiert bei TCP/IP keine klare Trennung von Dienst und Protokoll (vgl. [Tanenbaum98:61]). So werden in den folgenden Abschnitten nur die Risiken von Protokollen betrachtet.

Die Protokolle des in Abbildung 20 dargestellten TCP/IP-Protokollstapels verfügen über keine Sicherheitseigenschaften, so dass der Einsatz dieser Protokolle Risiken birgt. Die Risiken sind vor allem darauf zurückzuführen, dass die Benutzung des ARPANETs ursprünglich nur einem exklusiven Nutzerkreis vorbehalten war (vgl. [Yilmaz02:2]). Daher konnten Verstöße gegen eine Nutzungsordnung verfolgt werden. Außerdem wurde den Nutzern vertraut, dass sie die Technik sorgfältig nutzen. Ferner wurden manche Protokolle spontan von Freiwilligen wie Studenten in deren Freizeit produziert, die ihre Implementation nicht sorgfältig durchdachten (vgl. [Tanenbaum98:62]).

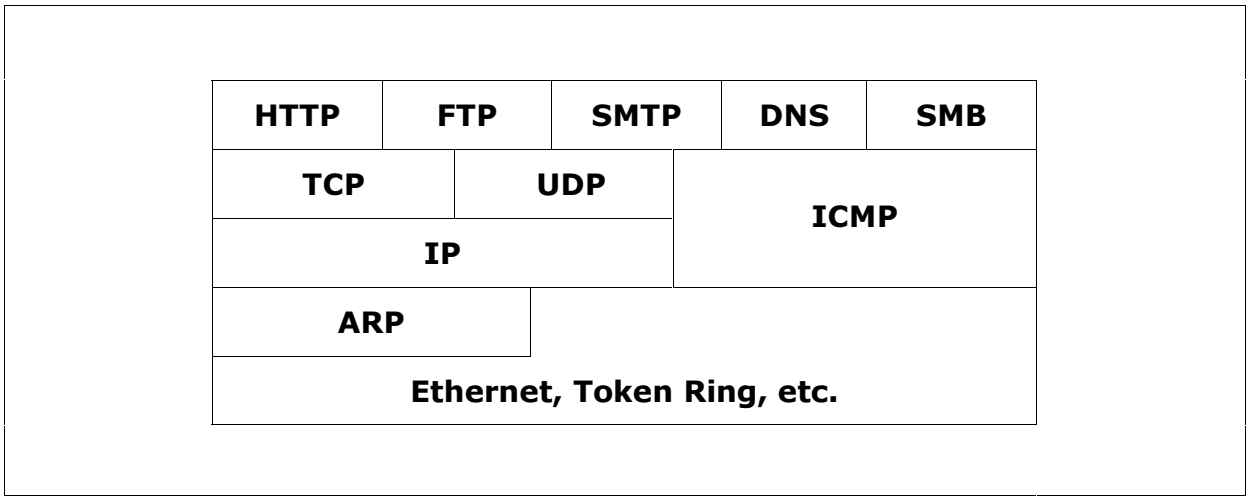


Abbildung 20: Der TCP/IP Protokollstapel

Die durch die Verwendung der Protokolle entstehenden Risiken werden in den folgenden Abschnitten erläutert. Dabei sollen nur die Probleme behandelt werden, die aus dem Konzept der Protokolle entstehen. In den Implementationen der Protokolle verankerte Risiken werden hier nicht behandelt, sondern sind der Kategorie Exploit zuzuordnen.

4.6.2. Address Resolution Protocol (ARP)

Das Address Resolution Protokoll (vgl. [RFC826]) dient der Übersetzung von IP-Adressen in die im Ethernet verwendeten MAC-Adressen. Diesem Protokoll mangelt es an fehlender Authentizität, so dass Spoofing möglich ist. Zudem werden Antworten angenommen und verarbeitet, ohne dass eine Anfrage erfolgt ist. In Verbindung mit Spoofing können so Man-in-the-Middle Angriffe realisiert werden.

4.6.3. Internet Protocol (IP)

Das Internet Protocol (vgl. [RFC791] und [RFC1349]) dient der ungesicherten Übertragung von Datenpaketen über ein Netz und verwaltet die Adressierung der an der Kommunikation beteiligten Rechner. Ihm mangelt es ebenfalls an fehlender Authentizität, wodurch Spoofing möglich ist. Zudem kann die Source-Routing Eigenschaft dazu missbraucht werden, Antworten einem anderen Rechner zugänglich zu machen. Somit sind Redirect- und Man-in-the-Middle Attacken möglich. Zusätzlich können Pakete mit gefälschter Absenderadresse durch das Source-Routing wieder zu ihrem originalen Absender gelangen, wodurch Spoofing Attacken für einen Angreifer interessanter werden.

Obwohl die Eigenschaft der Fragmentation von Paketen an sich kein Risiko darstellt, haben manche Implementationen Probleme beim Zusammensetzen der Fragmente. Dadurch kann es zum einen zu Abstürzen kommen, zum anderen können so Pakete durch eine Firewall geschleust werden, die aus dem Datenstrom herausgefiltert werden sollten (siehe [Northcutt02a:51]).

4.6.4. Internet Control Message Protocol (ICMP)

Das zustandslose Internet Control Message Protocol (vgl. [RFC792] und [RFC950]) dient dem Versenden von Kontrollnachrichten zur Wegesteuerung und Fehlerkontrolle über ein TCP/IP-basiertes Netz. Die Verwendung dieses Protokolls beinhaltet eine Vielzahl von Risiken.

Die Nachricht „Destination unreachable“ (Typ 3) kann zur Erzeugung eines Denial-of-Service verwendet werden (vgl. [vanEden01:4]). Wenn ein Opfer mit solchen Nachrichten mit gefälschtem Inhalt überflutet wird, würden im Extremfall auf Grund der Fehlerkontrolle sämtliche Pakete erneut gesendet, bis das Opfer aufgibt. Verbindungen würden abbrechen oder können nicht aufgebaut werden.

Eine “Redirect”-Nachricht (Typ 5) veranlasst den Empfänger, eine andere Route zum Ziel zu wählen. Dadurch können Man-in-the-Middle Angriffe realisiert werden (vgl. [vanEden01:4]).

Die Nachricht „Time exceeded“ (Typ 11) zeigt an, dass das Time-to-Live Feld des IP-Headers bei der Übertragung den Wert 0 erreicht hat. Dieses Verhalten wird für die Implementation von Traceroute (siehe Abschnitt 5.3.1) verwendet. Wird ein Host mit gefälschten Nachrichten eines solchen Typs überflutet, besteht die Gefahr, dass ein Denial-of-Service erreicht werden kann (vgl. [vanEden:5]).

Durch die Nachricht „Parameter Problem“ (Typ 12) wird ein Problem in der Parametrisierung eines IP-Paketes angezeigt. Auch hier kann durch eine Flut ein Denial-of-Service erreicht werden (vgl. [vanEden:5]).

Die Nachrichten „AM1: Address Mask Request“ (Typ 17) und „AM2: Address Mask Reply“ (Typ 18) (vgl.[RFC950:10]) können die Reconnaissance bei der Erkennung des Adressbereiches unterstützen.

Ein weiteres Problem ergibt sich aus dem Datenteil der „Echo-Request“ (Typ 8) und „Echo-Reply“ Nachrichten. Sie enthält einen Datenteil, dessen Länge in [RFC792:14f.] nicht definiert ist. Somit können mittels eines Pings Daten von bis zu 65507 übermittelt werden. Der Wert ergibt sich aus der maximalen Länge eines IP-Paketes (65535 Bytes) abzüglich der Headerlängen von IP- und ICMP-Paketen (20 bzw. 8 Bytes). Diese Schwachstelle wird unter anderem durch den Exploit „loki“ (vgl. Abschnitt 4.7.3) ausgenutzt.

4.6.5. Transmission Control Protocol (TCP)

Das Transmission Control Protocol (vgl. [RFC793], [RFC1122] und [RFC1323]) bietet eine zuverlässige Ende-zu-Ende Verbindung. Um die Einhaltung der Paketreihenfolge zu gewährleisten, benutzt dieses Protokoll Sequenznummern. Gelingt es einem Angreifer, diese zu erraten, so kann er Man-in-the-Middle- und Hijack-Angriffe durchführen (vgl. [Yilmaz02:13f.]).

4.6.6. Hypertext Transfer Protocol (HTTP)

Das Hypertext Transfer Protocol (vgl. [RFC1945]) ist ein zustandsloses Protokoll, das primär dem Transport von Webseiten dient. Während das Protokoll an sich als sicher gilt, bereiten die transportierten Inhalte Probleme. So können über das Protokoll mit Malware infizierte ausführbare Dateien transportiert werden, die auf der Clientseite zur Ausführung kommen.

Ein weiteres Problem stellt der Active Content dar, mit dem ausführbarer Code wie Java, JavaScript, VBScript und ActiveX-Controls auf der Client Seite bei reinem Betrachten der Seite ausgeführt werden kann. Enthält der Active Content maliziösen Inhalt, so kann dem Benutzer ein Schaden zugefügt werden. Mittels des Active Contents kann in einem neuen Fenster ein Browser vorgetäuscht werden, wodurch das Surfverhalten eines Nutzers beobachtet und manipuliert werden kann. Diese Art von Attacke wird als Web-Spoofing bezeichnet. Unter http://www.cs.dartmouth.edu/~pkilab/demos/spoofing/ ist ein Beispiel für ein Web-Spoofing zu finden. Eine weitere Möglichkeit des Webspoofing liegt in einem Problem mit der Adressierung von Webseiten. So führt der Link http://www.cnn.com:mainpage@129.170.213.101/~sws/0/ nicht nach cnn.com, sondern zu der IP-Adresse[30] hinter dem @-Symbol.

Zudem erlaubt HTTP Zugriffe serverseitige Skripte. Sie werden auf dem Server ausgeführt und Skripte können Schwachstellen enthalten, deren Ausnutzung eine Gefahr für den Web-Server und die auf ihm gespeicherten Daten darstellt.

4.6.7. File Transfer Protocol (FTP)

Das File Transfer Protocol (vgl. [RFC959], [RFC2228], [RFC2640], [RFC2773]) dient der Übertragung von Dateien, wobei der Zugriff auf die Dateien durch Authentifizierungs-mechanismen kontrolliert wird.

Ebenso wie bei HTTP kann mittels FTP Malware übertragen werden. Ein weiteres Problem stellt die Kommunikation dar, die in einen Kommandokanal und Datenkanal unterteilt ist. Bei der Übertragung von Informationen gibt es zwei Modi, den normalen und den passiven Modus (vgl. [Chapman00:456ff.]). Im normalen Modus gibt der Client dem Server über den Kommandokanal (Port 21/tcp) einen tcp-Port an, über den der Server eine Datenverbindung ausgehend von Port 20 aufbauen kann. Dies bedeutet, dass zumindest kurzfristig Verbindungen von Port 20 zu der Site möglich sind. Im passiven Modus gibt der Server einen zufällig gewählten Port vor, zu dem der Client eine Datenverbindung aufbauen muss. So ist auf Seiten des Servers ein Port offen. Diese Überlegungen müssen vor allem bei der Konfiguration einer Firewall bedacht werden.

[30] Hierbei handelt es sich um http://www.cs.dartmouth.edu/~sws/0/

4.6.8. Simple Mail Transfer Protocol (SMTP)

Das Simple Mail Transfer Protocol (vgl. [RFC821] und [RFC2821]) dient der Übertragung von Emails. Ein Problem des Protokolls ist die fehlende Verschlüsselung bei der Übertragung, so dass Emails mittels Sniffing oder auf einem an der Kommunikation beteiligten Server von beliebigen Personen gelesen werden können. Dieser Schwäche kann mittels weiterer Programme zur Verschlüsselung der Inhalte, beispielsweise PGP, begegnet werden.

Des Weiteren mangelt es dem Protokoll an fehlender Authentizität, so dass Absenderadressen gefälscht werden können. Auch dieser Schwäche kann durch eine digitale Signatur der Email begegnet werden.

Zudem kann SMTP innerhalb der Reconnaissance benutzt werden. So enthält jede Email den Pfad vom Sender zum Empfänger, wodurch Systeme entdeckt werden können. Des Weiteren bietet das Protokoll mittels der Befehle EXPN und VRFY (vgl. [RFC2821:20ff.]) die Möglichkeit, Benutzernamen und Gruppen, die für Mailinglisten benötigt werden, aufzufinden.

4.6.9. Domain Name System (DNS)

Das Domain Name System (vgl. [Albitz01]) übersetzt Rechnernamen in IP-Adressen und umgekehrt. So wird die Anfrage an einen Host mit der IP-Adresse beantwortet. Der Antwort mangelt es aber an Authentizität, wodurch ein DNS-Spoofing möglich wird. Hat ein Angreifer den Cache eines Servers manipuliert, enthält das Opfer eine falsche IP-Adresse.

Des Weiteren kann DNS die Reconnaissance unterstützen. So bietet sich die Möglichkeit des Zonentransfers an (vgl. Abschnitt 5.3.2), wobei ein Angreifer sämtliche auf einem DNS-Server gespeicherten Informationen über eine Zone erhält. Ferner stellen die Ressource Records HINFO und TXT eine Gefahr dar, wenn sie unnötige Informationen über einen Rechner enthalten, die einem Angreifer zum Erreichen seines Ziels helfen.

4.6.10. Server Message Block (SMB)

Das Server Message Block Protocol wurde von Microsoft, IBM und Intel entwickelt (vgl. [Chapman00:361]) und wird bei Microsoft Windows Systemen als Quasi-Standard für die Dateiübertragung verwendet. Seit Windows 2000 wird mit dem Common Internet File System (CIFS) eine erweiterte Implementierung des SMB Protokolls verwendet. Während das reine SMB nur auf der Schnittstelle namens NetBIOS abhängig ist, das nur durch die NetBT Protokollfamilie (vgl. [MS00-47]) über TCP/IP kommunizieren kann, kann das CIFS wahlweise auch direkt über TCP/IP kommunizieren. Die Zusammenhänge zwischen den verschiedenen Protokollen sind in Abbildung 21 dargestellt.

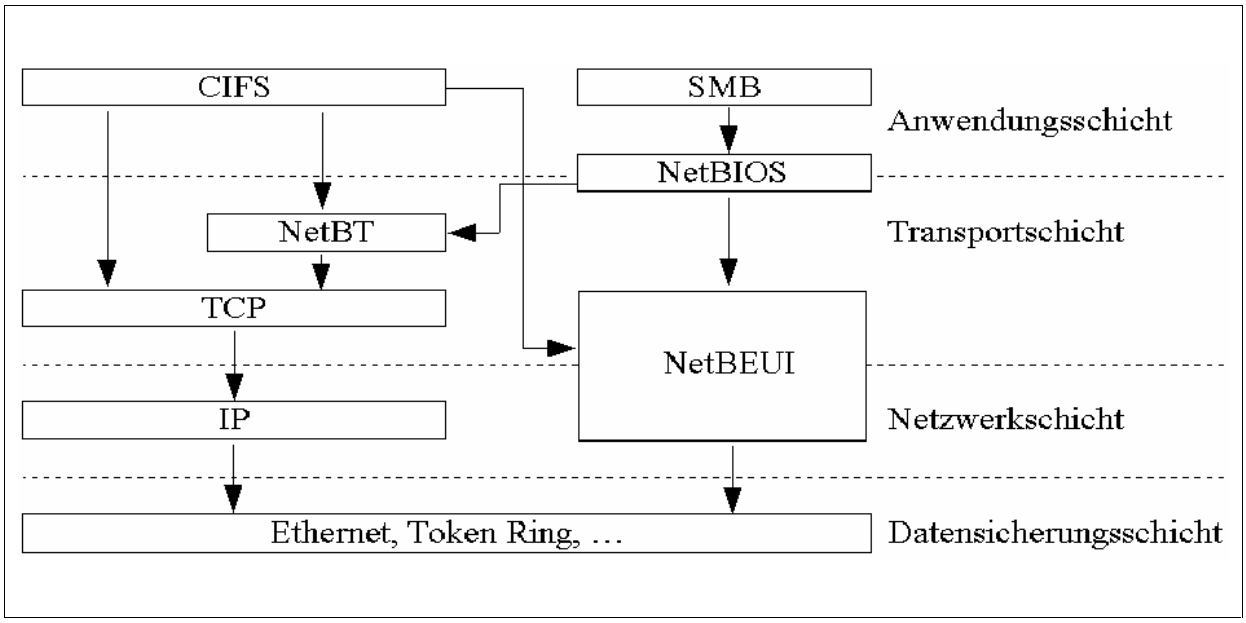


Abbildung 21: Zusammenhänge zwischen SMB und dazugehörigen Protokollen[31]

NetBIOS stellt eine tragende Rolle beim Austausch von Dateien über SMB dar, weshalb neben den Schwächen von SMB und CIFS auch Schwachstellen der über die NetBIOS-Schnittstelle angebotenen Dienste von Bedeutung sind.

Auch NetBIOS fehlt es an Authentizität der Rechnernamen, wodurch Spoofing der Rechnernamen möglich wird (vgl. [MS00-47]). Zudem lassen sowohl SMB als auch CIFS das Etablieren von Null-Sessions zu. Eine Nullsession ist eine Verbindung, für die weder ein Benutzername noch ein Passwort angegeben werden muss.

Von diesen Protokollschwächen sind sämtliche Systeme betroffen, die das Protokoll verwenden. So sind neben den Microsoft Windows Systemen auch Unix-Derivate betroffen, die Samba als Lösung zur Dateiübertragung gewählt haben.

4.7. Exploits

Der Exploit, der eine Methode zur Ausnutzung einer Schwachstelle ist, stellt das erste Element eines Angriffsvektors dar. Somit ist jeder Exploit eine Konkretisierung einer Bedrohung, deren Anwendung Bestandteil eines jeden Penetrationstests ist. In diesem Abschnitt werden einige näher erläutert.

[31] Grafik basiert auf [Chapman00:480] und [Solomon00:688]

4.7.1. ARP-Poisoning

Damit ein Rechner nicht für jedes IP-Paket einen ARP-Request durchführen muss, werden die Tupel bestehend aus MAC- und IP-Adresse in einem Cache, dem so genannten ARP-Cache, gespeichert, wodurch weniger Netzverkehr entsteht. Um den ARP-Cache zu pflegen, schickt ein Rechner eine Anfrage an die Broadcast Adresse ff:ff:ff:ff:ff:ff und fragt, welcher Host im Netzwerk die zu bestimmende IP-Adresse besitzt. Ist die IP-Adresse einem Host im Segment zugeordnet, so schickt dieser eine Antwort, die seine MAC-Adresse enthält.

Durch die Verwundbarkeit des ARP-Protokolls gegen Spoofing kann auch ein maliziöser Host ARP-Antworten senden, welche die MAC-Adresse des maliziösen Hosts und die IP-Adresse des Opfers enthält, obwohl ihm nicht die gesuchte IP-Adresse zugeordnet ist. Pflegt der Opferhost die maliziöse Antwort in seinen ARP-Cache ein, so entsteht eine falsche Zuordnung. Sämtliche Pakete für die gesuchte IP werden nun an den maliziösen Host geschickt. Auf Grund des „vergifteten“ ARP-Caches wird dieser Exploit „ARP-Poisoning“ genannt.

Zudem arbeiten manche Implementationen zustandslos, so dass ein Opfer ARP-Antworten in seinen Cache einpflegt, für die keine Anfrage gesendet worden ist. Somit kann ein Man-in-the-Middle Angriff realisiert werden.

4.7.2. Web Folder Traversal

Der Exploit für die Web Folder Traversal-Vulnerability benötigt keine speziellen Tools, sondern kann mittels eines Webbrowsers angewendet werden. Durch eine fehlerhafte URL kann ein Angreifer mit den Rechten des Webserver-users auf sämtliche Dateien eines Webservers zugreifen. Dadurch kann in Host eingedrungen und weitere Schäden wie z. B. ein Web-site Defacement verursacht werden. Die ausgenutzte Schwachstelle sowie notwendige Gegenmaßnahmen ist im Microsoft Security Bulletin [MS00-78] von Oktober 2000 beschrieben.

4.7.3. Loki

Loki (vgl. [Kurtz01:592]) besteht aus einem Client und einem Server, zwischen denen ein Tunnel mittels ICMP- oder UDP-Pakete erzeugt werden kann. Bei der Verwendung von ICMP für den Aufbau des Tunnels nutzt Loki den Datenteil der ICMP Echo-Request Pakete. Die Datenlänge ist dabei nach [RFC792:14] nicht definiert und somit nur durch die 65335 Bytes eines IP-Pakets begrenzt, wobei 20 Bytes für den IP-Header und 8 Bytes für den ICMP-Header benötigt werden.

4.7.4. Juggernaut

Mittels des für Linux verfügbaren Tools Juggernaut (vgl. [Cole02:152ff.]) ist ein Hijack-Angriff realisierbar. Das Tool erkennt sämtliche aktive TCP-Verbindungen und bietet eine Funktion zur automatischen Übernahme einer ausgewählten Verbindung. Neben diesem maliziösen Einsatzzweck kann es aber auch als erkennende Maßnahme eingesetzt werden, in dem nach Informationen gesucht werden kann, die auf einen Angriff schließen lassen. Juggernaut ist einer der ersten aus einer Reihe von Implementationen des Exploits zur Übernahme von Verbindungen.

4.7.5. Würmer

Auch Würmer stellen einen Exploit dar, da sie eine implementierte Methode zur automatischen Ausnutzung einer Schwachstelle sind. Beispielhaft wird im Folgenden der Morris-Wurm (vgl. [Kossakowski92:2.2.2]) beschrieben, welcher der erste Wurm war, der sich im Internet verbreitet hat.

Der Morris-Wurm ist nach seinem Erfinder Robert T. Morris jr., Sohn eines angesehenen Wissenschaftlers des National Computer Security Centers (NCSC), benannt worden. Die Verbreitung geschah über BSD-Unix Derivate auf den Systemen der Firmen DEC und SUN. Ein Ziel des Wurms war eine maschinenunabhängige Verbreitung, weshalb der Wurm im C-Quellcode übertragen und auf der entfernten Maschine kompiliert und ausgeführt wurde. Um die Ausführung auf DEC und SUN Systemen zu ermöglichen, wurden zwei Quellcodedateien übertragen, die für das jeweilige System optimiert waren. Beide Quellcodes wurden nacheinander kompiliert und ausgeführt. Hatte der Wurm Erfolg, so war ein Kompilat aktiv.

Zur Ausbreitung nutzte der Wurm die folgene, damals bereits bekannte Schwachstellen aus:

- Buffer overflow in fingerd

  Durch einen Buffer Overflow im Finger-Dämon konnte der Wurm eine Shell erzeugen, mit der eine Kopie des Wurmes übertragen und ausgeführt wurde.

- Remote shell

  Schwachstelle der Remote shell ist die fehlende Authentifizierung. So wird das gegenseitige Vertrauen durch Angabe der Rechnernamen in einer Datei bestimmt, so dass die in der Datei enthaltenen Rechner auf einem anderen Shells erzeugen und Befehle ausführen konnten. Diese Schwäche nutzte der Wurm für seine Verbreitung.

- DEBUG Befehl in Sendmaild

  Falls der DEBUG-Befehl des Email-Servers sendmail aktiviert war, konnte eine empfangene Email als Befehlsfolge interpretiert werden. So konnte der Wurm auf einem entfernten Host eine Shell starten und sich verbreiten.

- Schwache Passwörter

  Die vierte Schwachstelle sind schwache Passwörter, die der Wurm durch Zugriff auf die gespeicherten Benutzernamen und dem verschlüsselten Passwort auf einem System lokal erraten konnte. Mit den erratenen Passwörtern konnte der Wurm Shells auf weiteren Rechnern öffnen und sich verbreiten.

Aufgabe des Wurms war es, eine Reihe von Dateien vom Initiator-Prozess zu empfangen. Durch einen Fehler verbreitete er sich schneller als beabsichtigt und konnte noch ein Jahr später festgestellt werden.

Würmer stellen auch heute eine große Bedrohung dar. Opfer ihrer Aktivitäten sind vor allem Systeme, auf denen die Betriebssysteme Windows oder Linux installiert sind.

4.7.6. Smurf, Fraggle und Echo/Chargen

Ziel der Smurf Attacke (vgl. [Mojert01:80f.]) ist das Erreichen eines Denial-of-Service durch Flooding. Dazu werden Echo-Request Pakete (ICMP, Typ8), deren Absenderadresse durch Spoofing die des Opfers ist, an die IP-Broadcast[32] Adresse eines Netzsegmentes geschickt. Übersetzt ein in diesem Netzsegment vorhandener Router die IP-Adresse in die Broadcast Adresse der Datensicherungsschicht[33], so antworten alle im Netzwerk vorhandenen Systeme dem Opfer. Werden mehrere gespoofte Echo-Requests zur gleichen Zeit verschickt, so bricht das Opfer unter der Last zusammen.

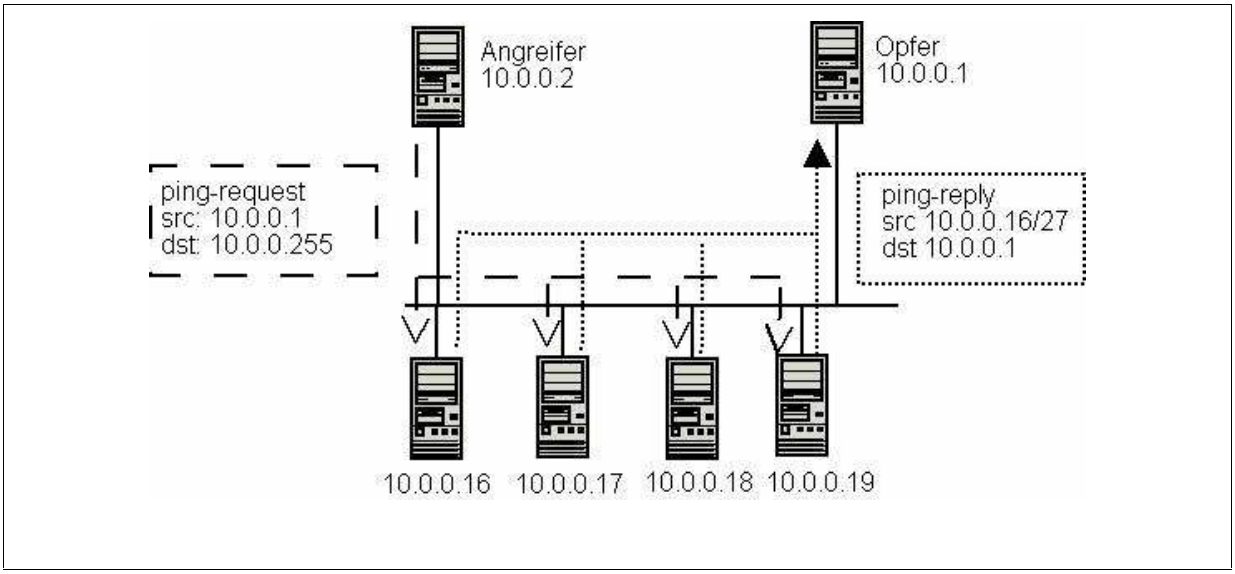


Abbildung 22: Die Smurf-Attacke

Eine Abwandlung von Smurf bildet Fraggle, der UDP Pakete anstelle von ICMP Paketen verwendet. Hierbei wird der auf Port 7/udp hörende Echo-Dienst mit Paketen überflutet. In einer als Echo/Chargen-Exploit (vgl. [Northcutt02a:429f.]) bezeichneten Variation sendet der Angreifer Pakete, deren Absenderadresse die des Opfers und der Quellport 7/udp des Echo Dienstes ist, an den Port 19/udp des Chargen Dienstes diverser Rechner. Der Chargen Dienst antwortet auf beliebige Pakete mit einer Zeichenkette, die er bei diesem Exploit an den Echo-Dienst des Opfers schickt. Der Echo-Dienst sendet jedes empfangene Paket an den Absender zurück, in diesem Falle an den Chargen Dienst. Dadurch entsteht ein Kreislauf, durch dessen Flut von Paketen die CPU-Last stark ansteigt und der mögliche Datendurchsatz stark abnimmt.

[32] Die IP-Broadcast Adresse ist die letzte Adresse eines durch IP-Adresse und Subnet-Maske angegebenen Subnetzes
[33] Beispielsweise FF:FF:FF:FF:FF bei Ethernet

4.7.7. TCP-SYN Flood

Der TCP-SYN-Flood (vgl. [Mojert01:51]) nutzt einen Fehler in der Konfiguration des TCP/IP Protokollstapels aus. Empfängt ein Rechner ein SYN-Paket, so wird ein neuer Speicherbereich reserviert, in dem alle benötigten Datenstrukturen der für die neue Verbindung benötigten Socket[34] gespeichert werden, und geht in den Zustand SYN_RCVD über, in dem er nach Mojert zwischen 75 Sekunden und 23 Minuten verharren kann. Ist die Absenderadresse des SYN-Pakets mittels Spoofing gefälscht, so erhält das Opfersystem kein ACK-Paket, wodurch der Verbindungsaufbau nicht vollendet wird.

Da die Implementationen lediglich über eine begrenzte Warteschlange für noch nicht vollständig aufgebaute Verbindungen verfügen, kann bereits mit wenigen SYN-Paketen eine Auslastung des Speichers erreicht werden, so dass der betroffene Host keine weiteren Verbindungen annehmen kann.

4.7.8. Tribe flood Network (TFN)

Tribe-Flood Network (vgl. [Northcutt02a:71]), kurz TFN, ist ein Exploit zur Realisation eines Distributed Denial of Service Angriffes. Wesentliches Merkmal eines solchen verteilten Angriffes ist die große Anzahl von beteiligten Rechnern.

TFN besteht dabei aus zwei Komponenten, einem Master Programm und einem Daemonprogramm. Das Daemonprogramm muss von dem Angreifer auf möglichst viele Rechner verteilt werden. Gesteuert wird es vom Masterprogramm mittels ICMP-Echo-Reply Nachrichten, wobei eine UDP-Flood, TCP-SYN-Flood, ICMP Echo-request Flood und die Smurf Attacke ausgelöst werden kann. Durch dieses System können alle Daemonen zeitgleich einen Angriff auf ein einziges Ziel durchführen, wodurch die Wirkung des Angriffes steigt. Ähnliche Exploits sind TFN2k, Trinoo und Stacheldraht.

4.7.9. Land

Die Land-Attacke (vgl. [Mojert01:96]) nutzt eine Parametrisierung der Pakete aus, um einen Denial-of-Service auf den Systemen zu erreichen. Dazu wird ein Paket, bei dem sowohl Quell- und Zieladresse als auch Quell- und Zielport übereinstimmen, an das Opfer geschickt.

4.7.10. Ping of Death und Teardrop

Durch die 16 Bit des Total-Length Feldes eines IP-Paketes (vgl. [RFC791:11]) beträgt die maximale Größe eines IP-Paketes 65335 Bytes. Abzüglich von 20 Bytes für den IP-Header und 8 Bytes für den ICMP Header beträgt die maximale Länge eines vom Ping-Programms verwendeten ICMP-Echo Request Paketes 65507 Bytes. Größere Pakete erzeugen bei einem verwundbaren System einen Absturz oder einen Neustart, wodurch das Ergebnis eines Denial-of-Service erzielt wird. Ursache dafür ist meist ein Überlauf von Variablen bei der Reassemblierung der Pakete, die auf Grund der Größe fragmentiert sind.

[34] Eine Socket ist nach [RFC793:5] die Konkatenation von IP-Adresse und Port. Durch zwei Sockets kann eine Verbindung eindeutig identifiziert werden, wodurch die Socket mit einem Connection End Point, kurz CEP, derISO/OSI-Modells vergleichbar ist.

Auch der Teardrop-Exploit (vgl. [Northcutt02a:54f.]) nutzt einen Fehler bei der Zusammensetzung fragmentierter IP-Pakete, um einen Denial-of-Service zu erreichen. Dabei werden Fragmente mit überlappenden Fragmentgrenzen (siehe Abbildung 23) erzeugt, bei deren Verarbeitung der Opferhost abstürzt.

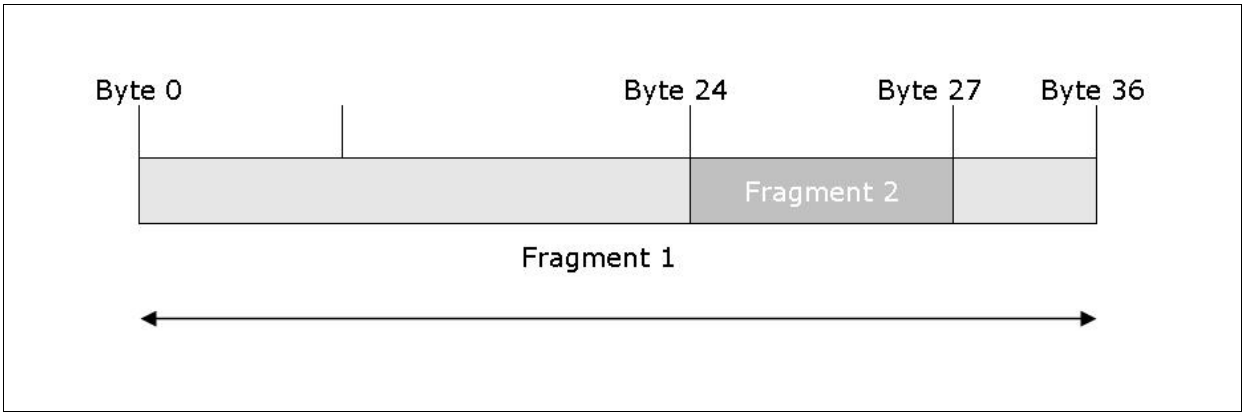


Abbildung 23: Fragmentüberlagerung bei Teardrop-Angriff nach [Northcutt02a:54]

5. KAPITEL:

VORGEHENSWEISE

Merkmale des Penetrationstests sind kontrollierte Angriffe aus der Sichtweise eines Angreifers. So soll die im letzten Kapitel vorgestellte Vorgehensweise eines Angreifers verfeinert werden, um eine Vorgehensweise für den Penetrationstest aus der Sicht eines Angreifers zu entwickeln. Auf Grund der Motivation beschränkt sich die Betrachtung auf die Anwendung eines Penetrationstests an dieser Stelle auf die Revision eines Netzwerkes.

5.1. Initiale Planung

Vor jedem Test ist ein Testplan zu erstellen. Ein solcher Plan besteht aus den Zielen und Rahmenbedingungen des Tests sowie einer Vorgehensweise. Dies ist notwendig, um eine systematische und eine per Definition notwendige kontrollierte Durchführung der Angriffe im Rahmen des Penetrationstests zu ermöglichen.

Zunächst hat ein Tester klare Ziele (vgl. [Kurtz00a:1]), die festlegen, was erreicht werden soll. So kann z. B. das Ziel sein, vertrauliche Dokumente zu enthüllen. Die Rahmenbedingungen des Tests können anhand der Parameter beschrieben werden.

Der Initiator, beispielsweise der Vorstand einer Aktiengesellschaft, kann den Penetrationstest an ein Tiger Team abgeben und kann damit die Angriffe auf seine Assets nicht kontrollieren. Mitglieder des Tiger Teams können ehemalige oder noch aktive Hacker sein, die zum Teil vorbestraft sind (vgl. [Kossakowski01]). Zudem besteht die Gefahr, dass manche der an den Tests beteiligten Personen nicht genügend qualifiziert sind. Auf Grund dieser Gefahren können ungeahnte Schäden entstehen und die Qualität des Tests leiden.

Die Qualität des Tests sollte aber vertraglich festgehalten werden. So kann der Initiator die Angriffe nicht vollständig kontrollieren, aber er kann die Grenzen des Tests vertraglich zusichern, für dessen Überschreitung das Tiger Team haftbar gemacht werden kann. Zudem kann sich der Initiator bei der Auswahl des Teams zusichern lassen, dass ein Team keine Hacker einsetzt. Dazu kann er beispielsweise polizeiliche Führungszeugnisse der Mitglieder verlangen. Auch kann der Initiator Referenzen fordern, welche den Erfahrungsgrad des Teams belegen. Auch diese Eigenschaften kann er sich vertraglich zusichern lassen. Ein solcher Vertrag, der die Qualität des Penetrationstests sichert, ist das Service Level Agreement, das im nächsten Abschnitt ausführlich behandelt wird.

Um eine Vorgehensweise zu entwickeln, muss ein Tiger Team die Schritte zum Erreichen der vorher definierten Ziele planen. Dazu gehört neben den Methoden auch die Wahl der passenden Tools. Dabei bietet sich nach Schultz (vgl. [Schultz96:4]) auch die Möglichkeit,

einige Tests als Skripte zu programmieren, die bei Bedarf in einem anderen Kontext wieder eingesetzt werden können.

Der Schaden, der durch einen Penetrationstest entstehen kann, ist bereits vor seiner Durchführung mit Mitteln der Risikobewertung festzustellen. Ist der mögliche Schaden zu groß, so kann die Umgebung simuliert werden. Unter einer Simulation wird nach [Page00:27] die „Nachbildung eines dynamisches Prozesses in einem Modell“ verstanden, wobei ein Modell die Abbildung charakteristischer Eigenschaften ist. Somit werden die betroffenen Assets in einem abgegrenzten Testfeld nachgebildet, in dem die Angriffe durchgeführt werden.

5.2. Service Level Agreement

Vor der Durchführung wird der Penetrationstest an der Schnittstelle zwischen dem Dienstleister und Kunden verbindlich spezifiziert, um so „potentielle Schwierigkeiten im Vorfeld erkennen und entschärfen“ [CZ27/03:10] und nachprüfbare Aussagen über eine verlässliche Qualität treffen zu können (vgl. [Röhrig02:186ff.]). Diese Schnittstelle bildet das Service Level Agreement.

Ein Service Level Agreement, kurz SLA, ist nach [Köppel99:1] eine formal ausgehandelte Einigung (engl.: agreement), welche die Ziele identifiziert, die Verantwortlichkeiten klar stellt und die Kommunikation zwischen Anbieter und Kunden erleichtert. Somit bildet das SLA einen rechtsgültigen Vertrag zwischen dem Tiger Team und dessen Kunden. Dabei wird definiert, wer welche Leistung in welchem Zeitraum an wen liefert. Vorteil dieser Einigung ist ihre Verbindlichkeit, auf Grund derer Vertragsstrafen bei Schäden und Nichterfüllung, sowie verbindliche Haftungsregelungen möglich sind (vgl. [CZ27/03:10]).

Im SLA sind die Rahmenbedingungen des Penetrationstests festgehalten, wobei auch die Parameter des Tests definiert werden. So kann aus einem SLA die durchgeführte Variante des Penetrationstests abgeleitet werden.

Die Bestimmung der Ziele beginnt mit der Festlegung der Aufgabe des durchzuführenden Penetrationstest. Beispielweise könnte ein Test überprüfen, ob personenbezogene Daten enthüllt werden können, oder ob ein Incident Response Team die Incidents bemerkt und die richtigen Maßnahmen ergreift. Aus den Aufgaben ergibt sich der Parameter Auswirkung (engl.: impact), welcher auch den möglichen Schaden impliziert.

Neben den Aufgaben beinhalten die Ziele auch die Auswahl der Systeme, die in den Test mit einbezogen werden können und dürfen, was die Definition des Parameters Abdeckung (engl.: coverage) ist. Die Entscheidung, ob ein System einem Test ausgesetzt wird, ist abhängig von dem mit der Penetration verbundenen Risiko (vgl. [Moyer98:2]). So müssen auch die zu erwartenden Schäden im SLA berücksichtigt werden. Dabei kann ein Dienstleister Zusicherungen über die Eigenschaften der Systeme während des Tests machen, wobei aber eine exakte Definition notwendig ist (vgl. [CZ27/03:10]). Ein Beispiel für eine ungenaue Definition ist eine zugesicherte Verfügbarkeit von 99,5 Prozent im Jahr. Sie würde eine durchgehende Ausfallzeit von 43,2 Stunden zulassen, die für den Kunden aber kaum tragbar ist. Zweck der exakten Definition ist der Schutz des Tiger Teams vor

Schuldzuweisungen seitens des Kunden, die bei Einhaltung der exakt definierten Eigenschaften der Systeme nicht gemacht werden können.

Aus der Abdeckung ist bereits erkennbar, dass bei der Festlegung der Ziele sehr sorgfältig vorzugehen ist, um auszuschließen, dass der Dienstleister trotz Einhaltung der Rahmenbedingungen ein unzureichendes Ergebnis liefert. So soll ausgeschlossen werden, dass ein Dienstleister die bloße Verwendung eines Vulnerability Scanners als Penetrationstest zu verkaufen versucht.

Weiter ist zu beachten, dass der betriebene Aufwand dem Nutzen gegenübersteht. Hierbei ist auch festzulegen, wann die Ziele erreicht sind. Wie in Abschnitt 4.2 dargestellt ist, kann ein Angriff durch die Metastase iterativ durchgeführt werden, wobei ein sehr großer Iterationsgrad erreicht werden kann, dessen Aufwand nicht gerechtfertigt ist. Auch kann das Tiger Team die Iteration weiter vorantreiben, als für den vom Kunden gewünschten Nutzen notwendig ist.

Innerhalb der Ziele kann auch definiert werden, ob der Dienstleister den Test von außerhalb oder innerhalb des Security Perimeters des Kunden durchführt, wodurch der Parameter Standort (engl.: location) festgehalten wird.

In einem SLA kann auch verbindlich festgelegt werden, welche Informationen ein Team zu Beginn des Tests bekommt, wodurch der Parameter Kenntnis (engl.: knowledge) behandelt wird. Dabei ist aber auch zu bedenken, dass bei zu ungenauer Kenntnis des Teams beispielsweise der falsche Webserver angegriffen werden kann. Daher ist eine sorgfältige Ausarbeitung notwendig. Gemeinsam wird auch der Parameter Bekanntmachung (engl.: notice) festgelegt. Hierbei kann vermieden werden, dass der Test durch Administratoren manipuliert wird, die beispielsweise schlecht gesicherte Systeme vom Netz trennen.

Zusätzlich zu den Zielen wird in einem SLA auch der Zeitraum des Tests bestimmt, da ein Test während der Durchführung nichts an der Sicherheitssituation des Kunden ändert. Zudem ist hier die Festlegung eines „Freezepoint“ sinnvoll, ab dem der Kunde keine Änderungen an der zu testenden IT-Infrastruktur vornehmen darf, da sonst die Ergebnisse des Penetrationstests verfälscht werden können. Ausgenommen von solchen Änderungen sind Aktionen wie das Ändern von Paßwörtern oder Einspielen von Patches, die zur Befolgung der Policy notwendig sind. Der Zeitraum kann dabei auch eine längere Zeit beinhalten, wobei die Tests dann unter den Bedingungen des SLA periodisch in einem bestimmten Abstand wiederholt werden.

Weiter muss geregelt werden, wie die per Definition des Penetrationstests geforderte Kontrolle der Angriffe erfolgen soll. Die Überwachung muss dabei permanent erfolgen, damit auftretende Schäden sofort eingedämmt werden können. Durch die permanente Überwachung kann der Test bei zu großen Schäden auch abgebrochen werden. Dazu müssen verantwortliche Personen bestimmt werden. Eine solche Person wird nach Moyer (vgl. [Moyer98:2]) als „cutout“ bezeichnet. Der cutout kann ein Mitarbeiter des Kunden sein, wobei dieser zusichern sollte, dass er zur Kontrolle fähig ist. So können Haftungsfragen im Nachhinein leichter geklärt werden. Dabei können auch Schulungen für den cutout durch das Tiger Team vereinbart werden (vgl. Moyer98:2]). Eine andere Alternative für einen cutout ist ein Mitglied des Teams, das während des Tests ständig beim Kunden ist. Er kann z. B. als neuer Mitarbeiter vorgestellt werden. Weiter kann die Kontrolle nach dem Vier-Augen-

Prinzip auch von einem internen als auch einem externen cutout zugleich durchgeführt werden. Der cutout hat auch die Aufgabe, die Mitarbeiter des Dienstleisters in Falle einer Verhaftung aus dem Gefängnis zu befreien (vgl. [Moyer98:1]).

Zudem soll das SLA sicherstellen, dass ethischen und gesetzlichen Vorschriften Genüge getan wird. Wenn der Kunde z. B. personenbezogene Daten verarbeitet und der Dienstleister durch den Test darauf Zugriff bekommen kann, sollte sich der Kunde versichern lassen, dass der Dienstleister die §§ 5 und 14 BDSG befolgt und die Daten weder nutzt noch weitergibt. Wenn ein Tiger Team Daten, die gegen unberechtigten Zugang gesichert sind, als Beweis für die Umgehung einer Sicherungsmaßnahme kopiert, würde es sich im Sinne des § 202a StGB strafbar machen, wenn es nicht dazu befugt ist. Da die Kopie solcher Daten zur Kontrolle des Tiger Teams genutzt werden kann, sollte das Tiger Team durch das SLA eine entsprechende Erlaubnis bekommen. Gleiches gilt auch für die Manipulation von Log-Dateien, die nach §§ 268-270 StGB eine strafbare Fälschung von Beweismitteln darstellen kann.

Für den Fall, dass der Dienstleister einen Schaden verursacht, ist durch das SLA im Voraus zu klären, wer welche Haftung übernimmt. Liegt die Haftung beim Dienstleister, so sollte dieser durch das SLA zu einer Versicherung verpflichtet werden, um die Schäden finanziell abdecken zu können (vgl. [Moyer98:2]). Des Weiteren kann der Dienstleister durch das SLA für Nichterfüllung des Vertrages haftbar gemacht werden. Um den Schaden zu begrenzen, sollte der Kunde durch das SLA zusichern, dass er eine Wiederherstellung (engl.: recovery) betroffener Systeme leisten kann (vgl. [Lowery02:2]). Um den Schaden weiter zu begrenzen, kann auch vereinbart werden, dass der Test nur simuliert wird. Dabei ist zu bedenken, ob das computerisierte Modell sämtliche Eigenschaften der realen Umgebung leisten kann.

Bei einem Penetrationstest kann die Datenverarbeitung des Kunden, der ein Betrieb oder eine Behörde sein kann, gestört werden. Eine solche Störung oder schon der Versuch einer Störung stellt eine Straftat im Sinne des § 303b StGB dar. Allerdings kann der Kunde Vertragsstrafen erwarten, wenn Dritte von der Störung betroffen sind und einen Schaden erleiden. Weiter sollte bedacht werden, dass Daten gelöscht, unterdrückt, unbrauchbar gemacht oder verändert werden können. Geschieht dies rechtswidrig, so ist im Sinne des § 303a StGB eine Straftat begangen worden. Hierbei sollte im SLA darauf geachtet werden, dass nicht nur der Kunde die Befugnis erteilen muss, sondern auch seine Geschäftspartner.

Zur Ausübung einer Penetration sind Besitz, Installation und Wartung von Tools erforderlich, mit denen Zugangskontrollen umgangen werden können. Da solche Tools sehr oft auch für die unerlaubte Nutzung eines entgeldlich erbrachten zugangskontrollierten Dienstes eingesetzt werden können, sind Besitz, Installation und Wartung dieser Tools nach § 3 Satz 2 ZKDSG (vgl. [ZKDSG02]) verboten und werden bei gewerblicher Nutzung als Ordnungswidrigkeit mit einem Bußgeld in Höhe von 50.000,- € verfolgt. Dieses Bußgeld sollte ein Tiger Team in seiner Kostenkalkulation beachten.

Jeder Dienstleister sollte nach [Röhrig02:2] „mit den Vorschriften vertraut sein und […] Fachkunde nachweisen“. Eine solche Kenntnis ist auch dringend erforderlich, da die Gesetze für solche Delikte hohe Geldstrafen und Freiheitsstrafen von bis zu fünf Jahren vorsehen.

Der Dienstleister sollte sich daher durch das SLA die notwendigen Befugnisse erteilen lassen und zudem zu einer Versicherung verpflichtet werden, um eventuelle Schäden finanziell abdecken zu können (vgl. [Moyer98:2]). Der Kunde hingegen sollte dem Tiger Team

zusichern, dass er der Gefahr der strafrechtlichen Verfolgung im Sinne der §§ 303a und 303b vorbeugt und zur Begrenzung eventueller Schäden eine Wiederherstellung (engl.: recovery) betroffener Daten und Systeme leisten kann (vgl. [Lowery02:2]). Weiter sollte der Kunde zu seinem eigenen Schutz überprüfen, dass er im Falle einer Störung seiner Datenverarbeitung nicht Vertragstrafen durch die Verträge mit Dritten zu erwarten hat.

Erweist ein Dienstleister die Kenntnis der Vorschriften, so kann davon ausgegangen werden, dass er auch den in § 263a StGB behandelten Computerbetrug kennt. Neben einem Vermögensvorteil darf sich ein Tiger Team hinsichtlich ethischer Richtlinien auch keinen allgemeinen Vorteil gegenüber dem Kunden verschaffen können. Dabei wird unter ethischen Richtlinien das verantwortungsvolle Handeln gegenüber anderen verstanden (vgl. [Duden5:220]).

Auf Grund der Notwendigkeit für den Dienstleister, ethische und rechtliche Grundsätze zu befolgen, sollten dem Tiger Team keine Hacker angehören. Das Tiger Team sollte aus Mitgliedern bestehen, die über eine fundierte Ausbildung Bereich IT-Sicherheit verfügen. So kann nicht nur sichergestellt werden, dass den Beteiligten ethische und rechtliche Prinzipien bewusst sind, sondern auch die notwendige Erfahrung und eine systematische Vorgehensweise vorausgesetzt werden. Dafür kommt ein Hacker nicht in Frage, da dieser meist nur gewisse Angriffe durchführen kann, die er beispielsweise durch Tutorials im Internet gelernt hat.

Als Dienstleister kommen ein aus internen Mitarbeitern bestehendes Team oder ein externes Team einer dritten Partei in Frage. Da interne Mitarbeiter meist nicht über die notwendige hohe Expertise verfügen, sind Schulungen notwendig, die Kosten verursachen und die Dauer des Tests verlängern. Aber auch bei der Wahl interner Mitarbeiter ist ein SLA sinnvoll, um die Handlungsspielräume und Haftungsgrenzen zu definieren, damit beide Parteien vor ungewollten Schäden wie z. B. dem Verlust des Arbeitsplatzes bewahrt werden.

Außerdem sollte im SLA auch der Inhalt des Abschlussberichtes festgehalten werden, da dieser zum einen vertrauenswürdige Informationen enthalten kann. Zum anderen sollte der Bericht den Testablauf inklusive Methoden, eingesetzter Tools und gefundener Probleme enthalten, womit er als Grundlage für die Forderung von Haftungsansprüchen besonders bei Nichterfüllung dienen kann. So könnte z.B. ein Kunde argumentieren, dass die Methoden und Tools die geforderten Ziele nicht hätten erreichen können. So sollte das SLA auch die Zahlungsmodalitäten behandeln, damit keine Partei im Nachhinein die finanzielle Entgeltung anzweifeln kann.

Mit dem SLA kann die Qualität des Penetrationstests vorausschauend festgehalten werden. Unter den Einschränkungen des Service Level Agreements wird der Penetrationstest durchgeführt, dessen Vorgehensweise Thema der folgenden Abschnitte ist.

5.3. Reconnaissance

Erstes Ziel eines Angreifers ist die Feststellung, was er angreift. Dazu beschafft er sich Informationen, indem er Systeme sucht und dessen Eigenschaften erkundet. Diese Station des Angriffes wird Informationsbeschaffung (engl.: information gathering) oder auch

Reconnaissance genannt. Unter Reconnaissance wird nach [Duden-Oxford:590] die militärische Aufklärung bzw. Erkundung verstanden.

Bei einem zero-knowledge Pentretrationstest, bei dem das Tiger Team kein Wissen über die zu betrachtende Organisation besitzt, ist die Reconnaissance eine Methode, um fehlende Informationen zu beschaffen. Sie ist auch bei vollständigem Wissen sinnvoll, da der Angreifer schon bei der Reconnaissance Schwachstellen ausnutzt. Ansonsten würden Schwachstellen übersehen werden. Zudem können die vom Kunden gestellten Informationen unvollständig sein, was durch die Reconnaissance überprüft und vervollständigt werden kann.

Im Zusammenhang mit der Reconnaissance wird häufig der Begriff „footprinting“ benutzt. Nach [Kurtz00:31] wird unter Footprinting die „systematische Vorgehensweise, die es einem Angreifer ermöglicht, ein vollständiges Profil der Sicherheitsvorkehrungen eines Unternehmens zusammenzustellen“ verstanden. Das vollständige Profil der Sicherheitsvorkehrungen kann aber erst nach Abschluss der gesamten Reconnaissance erstellt werden. Welche Teile der Reconnaissance zum Footprinting gehören, ist in der Literatur nicht eindeutig bestimmt. Aus dem Begriff lässt sich allerdings das Ziel ableiten, aus gesammelten „Spuren“ der Organisation auf vorhandene Systeme und Sicherheitsvorkehrungen zu schließen. Somit soll als Ziel der Reconnaissance die Bestimmung der zum Ziel gehörenden Systeme sowie deren Eigenschaften festgehalten werden, was einer Aufklärung und Erkundung im militärischen Sinne entspricht.

Ziele der Reconnaissance (vgl. [Cole02:64]) sind:

- Beschaffen initialer Information
- Adressraum des Netzwerkes bestimmen
- Aktive Systeme bestimmen
- Offene Ports und Zugriffspunkte bestimmen
- Die angebotenen Dienste bestimmen
- Das Netzwerk vermessen („map out the network“)
- Benutzernamen bestimmen

Die Reconnaissance ist in der Literatur in mehrere Phasen unterteilt. Diese Aufteilung wird von den Autoren unterschiedlich vorgenommen, wobei gleiche Begriffe anders bestimmt werden. In dieser Arbeit wird die Reconnaissance nach Payne (vgl. [Payne01:3ff.]) in die Phasen Discovery und Enumeration unterteilt, die in folgenden näher betrachtet werden.

5.3.1. Discovery

Die Discovery zielt darauf ab, vorhandene aktive Systeme einer Organisation zu entdecken. Dazu sucht der Angreifer mittels passiver Methoden, die nicht von dem Opfer erkannt werden können, nach Spuren in öffentlichen Informationsquellen sowie whois- und DNS-Datenbanken. Zudem kann mittels dieser Methoden der Adressraum des Netzwerkes erkannt werden. Durch die Verwendung von Traceroute und Ping-sweeps, die aktive Angriffs-methoden sind, kann der Angreifer innerhalb des erkannten Adressraumes weitere aktive Systeme im Netzwerk einer Organisation aufdecken. Die Ergebnisse werden in einem Netzwerkplan festgehalten.

Zu den Elementen der Discovery gehört daher:

- Öffentliche Quellen
- Whois-Datenbanken
- DNS
- Traceroute
- Ping Sweep

Diese Elemente werden im Folgenden näher erläutert.

Öffentliche Quellen

Für die Beschaffung von Informationen aus öffentlichen Quellen benutzt Cole den Ausdruck „Open Source Information“: „Open Source is general information about a company or its partners that anyone can obtain“ [Cole02:66]. Eine solche Informationssuche ist sowohl technisch als auch rechtlich nicht zu unterbinden, da das Ziel der Information die Informierung der Öffentlichkeit ist. Daher sind solche Informationen gewollt frei zugänglich und lassen sich nicht unterbinden. Es sollte aber genauestens überlegt werden, welche Informationen in öffentlichen Quellen preisgegeben werden dürfen.

Ein Angreifer versucht durch die gewonnenen Informationen mögliche Zugänge zu den Systemen und vorhandene Sicherheitsvorkehrungen zu finden. Außerdem können Daten von Personen erlangt werden, die zum Zweck eines Social Engineering Angriffes missbraucht werden können.

Quellen solcher Informationen können Zeitungen, Mailinglisten, Newsgroups, Webseiten, das Handelsregister oder auch Müll sein. So können auf der Website der Organisation Kontaktdaten oder auch Hinweise auf mögliche Systeme und Sicherheitsvorkehrungen erlangt werden. Weiter können Suchmaschinen bei Eingabe des Namens Informationen preisgeben. Durch dieses Vorgehen können Informationen auf mögliche Zeitungsartikel erschlossen werden, die nützliche Informationen enthalten können. So berichtet Cole (vgl. [Cole02:66]) von einem Zeitungsartikel, der die Anzahl redundanter Zugänge zum Netzwerk preisgab, was einem Angreifer das Erreichen eines Denial-of-Service erleichtert.

Im Internet sind auch Archive vorhanden, in denen Diskussionen von Administratoren und sonstigen Fachkräften über Probleme gespeichert sind. Das Durchforsten solcher Archive kann einem Angreifer nützliche Auskünfte über eingesetzte Systeme und Sicherheitsvorkehrungen sowie deren Schwachstellen liefern.

Die Preisgabe von Informationen kann auch durch das so genannte „Dumbster Diving“ verursacht werden. Hierbei durchsucht der Angreifer den Müll der Organisation nach nützlichen Hinweisen. Zudem kann ein Angreifer sich physikalisch vor der Organisation positionieren. Dabei kann er z. B. feststellen, wann welche Reinigungsfirma kommt. So kann sich der Angreifer als Mitarbeiter einer Reinigungsfirma verkleidet, Zutritt zum IT-Verbund der Organisation verschaffen.

Sinnvoll ist auch, während dieser Phase nach Informationen über die Unternehmensgeschichte zu suchen. So kann ein Unternehmen den Namen geändert oder

fusioniert haben. Dabei ist auch eine Suche nach alten Namen sinnvoll, um sich mehr Informationen beschaffen zu können. Weiter können auch Informationen über den Internet Service Provider oder auch Geschäftspartner gefunden werden. Da sich Geschäftspartner häufig vertrauen, könnte dieses Vertrauen für einen Angriff ausgenutzt werden.

Whois-Datenbanken

An dieser Stelle besitzt ein Angreifer entweder einen Domainnamen oder eine IP-Adresse. Da ein Netz mehrere Rechner mit Namen und IP-Adressen enthält, müssen Informationen über die Gesamtheit der Systeme des Netzwerkes gesammelt werden.

Da sich Menschen die IP-Adressen schlecht merken können, wurde eine Abbildung von Domainnamen auf IP-Adressen geschaffen. In den Anfängen des Internets geschah diese Abbildung durch Abfrage der `hosts` Datei, die lokal auf jedem Rechner gespeichert war. Um Änderungen und Ergänzungen zu synchronisieren, wurde eine zentrale Instanz geschaffen, die eine aktuelle `hosts` Datei zentral zur Verfügung stellt und Änderungen einpflegt. Diese Instanz wurde Network Information Center, kurz NIC, genannt, das beim Stanford Research Institute untergebracht war. Die Verwendung der `hosts` Datei wurde durch das Domain Name System, kurz DNS abgelöst[35]. Heute beschränkt sich die Aufgabe eines NIC auf die Registrierung von Domainnamen, um die Eindeutigkeit der Namen zu gewährleisten.

Eben wurde bewusst von einem NIC gesprochen, da es heute mehrere Network Information Center gibt, wobei jedes nur eine bestimmte Teilmenge der im Internet vorhandenen Domains pflegt. Meist pflegt ein NIC, wegen seiner Aufgabe auch Registrar genannt, eine Top-Level Domain. Während bei den meisten Top-Level Domains ein Monopol besteht, werden .com, .org und .edu Domains von mehreren Registraren gepflegt. Die Registrare sind zum Teil hierarchisch angeordnet, so dass es Registrare gibt, die Informationen über mehrere Top-Level-Domains besitzen.

Erste Aufgabe ist, den zur Domain gehörenden Registrar zu finden, um Informationen über die Domainnamen zu bekommen. Vereinfacht wird dies allerdings heute durch den Dienst Allwhois[36], bei dem Informationen über alle Domains des Internets abgefragt werden können. Während die Abfrage von Allwhois über eine Web-Oberfläche geschieht, bieten andere Registrare die Möglichkeit, Abfragen über den unter Port 43/tcp verfügbaren whois-Dienst durchzuführen. UNIX-Derivat liefern einen passenden Client mit, unter Windows kann dies mit Freeware-Tools geschehen. Die Ausgabe des whois-Tools unter UNIX für die Domain uni-hamburg.de ist in Abbildung 24 in gekürzter, anonymisierter Form dargestellt.

Aus den in Abbildung 24 dargestellten Informationen ist für einen Angreifer zunächst der Eintrag nserver interessant. Dieser Eintrag enthält den für die Zone uni-hamburg.de autorisierten Nameserver, der im Zweifelsfalle über die aktuellsten und vollständigsten Informationen einer Domain verfügt.

[35] Die `hosts` Datei wird heute noch in kleineren lokalen Netzen benutzt, nicht aber für das Internet.
[36] www.allwhois.com

```
# whois -h whois.denic.de uni-hamburg.de

domain:      uni-hamburg.de
descr:       Universitaet Hamburg; Regionales Rechenzentrum
nserver:     rzdspc1.informatik.uni-hamburg.de 134.100.9.61
status:      connect
changed:     20011126 150620
source:      DENIC

[admin-c][tech-c][zone-c]
Type:         PERSON
Name:         Hugo Maier[37]
Address:      Universitaet Hamburg
Address:      Regionales Rechenzentrum
Address:      Schlueterstr. 70
City:         Hamburg
Pcode:        20146
Country:      DE
Phone:        +49 40 42838 0[30]
Fax:          +49 40 42838 0[30]
Email:        mail@uni-hamburg.de[30]
Changed:      20011123 160405
Source:       DENIC
```

Abbildung 24: whois Eintrag für uni-hamburg.de (gekürzt und anonymisiert)

Weiter kann der Angreifer hier Kontaktinformationen entnehmen. Die Informationen enthalten die Adresse, wodurch der Angreifer dort nach öffentlichen Informationen suchen oder physikalische Angriffe starten kann. Zudem sind Personendaten mit Telefonnummer und Email-Adresse enthalten, die der Angreifer für Social Engineering Attacken missbrauchen kann. Dabei kann er mehr Informationen für seinen Angriff erhalten.

Um in der aktiven Reconnaissance möglichst alle Rechner erkennen zu können, ist es sinnvoll, den Adressbereich festzustellen. So kann der Angreifer gezielter nach Informationen suchen. Dazu können manche Whois-Server nach einer IP-Adresse gefragt werden. Ein häufig genanntes Beispiel ist der Server der „American Registry for Internet Numbers“, kurz ARIN[38]. Diese nicht-kommerzielle Organisation kümmert sich um die Verwaltung der IP-Adressen (vgl. [ARIN03]). Obwohl sie offiziell nur für Nord Amerika und für Teile Afrikas und der Karibik zuständig ist, lassen sich bei ARIN auch Informationen über IP-Adressen in anderen Teilen der Welt abfragen. Mittels des Unix Befehls

```
# whois –h rr.arin.net adresse
```

lassen sich der Besitzer der Adresse sowie der Adressbereich der Domain feststellen. Die Abbildung 25 stellt einen Auszug der Abfrage des Adressbereiches des Fachbereichs Informatik der Universität Hamburg dar. Es ist nur der öffentlich bekannte Name des Webservers nötig.

[37] Information geändert

[38] In der Literatur ist diese Aufgabe häufig unter dem Begriff ARIN zu finden, solche Anfragen funktionieren beispielsweise auch mit whois.ripe.net, nicht aber bei whois.denic.de.

```
# whois -h rr.arin.net ‘host www.informatik.uni-hamburg.de |\
 cut -f 4 -d " " | cut -f 1-4 -d "." -s‘

% ARIN Internet Routing Registry Whois Interface

inetnum:       134.100.0.0 - 134.100.255.255
netname:       UNIHH
descr:         Universitaet Hamburg campus net
country:       DE
```

Abbildung 25: Ausgabe der Abfrage des Adressbereiches von www.informatik.uni-hamburg.de (gekürzt)

Die Abfrage des Adressbefehls lässt sich auch automatisieren. Dies ist mit dem Skript addrrange möglich, dessen Quellcode im Anhang G1 enthalten ist. Die Anwendung des Skripts auf den Webserver des Fachbereiches Informatik ist in Abbildung 26 dargestellt.

```
# sh addrrange www.informatik.uni-hamburg.de
       134.100.0.0 – 134.100.255.255
```

Abbildung 26: Anwendung von addrrange auf www.informatik.uni.-hamburg.de

Mit den Informationen des autorisierten Nameservers und des Adressbereiches kann der Angreifer nun mittels DNS-Anfragen weitere Systeme identifizieren.

DNS

Das DNS abgekürzte Domain Name System entstand, da der Austausch der *host* Datei den Anforderungen an die maximal vertretbare Last sowie der Eindeutigkeit und Konsistenz der Namen nicht mehr entsprach. Es ist für die Auflösung von Domainnamen in die zugehörige IP-Adresse zuständig. Die Ermittlung des zu einer IP-Adresse gehörenden Domainnamens wird dabei Reverse DNS genannt.

Da die Verwaltung sämtlicher im Internet vorhandener Domainnamen auf einem Rechner zu aufwendig ist, können Teile des gesamten Namensraumes an andere Rechner delegiert werden. Teilbäume des Namensraumes werden Domains genannt (vgl. [Albitz01:16]). Der Namensraum ist dabei hierarchisch aufgeteilt, wobei Kinder des Wurzelknotens Top-Level Domains genannt werden.

Ein vollständig qualifizierter Domainname (engl.: fully qualified domain name, kurz FQDN) ist ein Name, der die Position der Domain im Baum ausgehend von der Wurzel angibt. So ist informatik.uni-hamburg.de. vollständig qualifiziert, was durch den letzten Punkt gekennzeichnet ist. Im Alltag werden Domainnamen meist nur bis zur Top-Level-Domain angegeben, da dies zur Wahrung der Eindeutigkeit genügt. Teile einer Domain werden Zone genannt. Während eine Domain ein Teilbaum ist und somit alle Kindsknoten eines Knotens beinhaltet, kann eine Zone nur einen oder mehrere bestimmte Knoten enthalten.

Die Abfrage des DNS ist ein wichtiger Schritt bei der Entdeckung von Systemen. Das DNS-System ist durch Server implementiert, wobei jeder Server für die Verwaltung einer Zone zuständig ist (vgl. Albitz01:24). In diesem Fall besitzt der Nameserver die Autorität für die

Zone und wird daher autorisierter Name-Server genannt. Ein Server kann dabei auch für mehrere Zonen zuständig sein.

Bei Anfragen eines Resolver genannten Clients an einen Server, kann der Server auch Antworten aus seinem Cache generieren. Da diese Daten unter Umständen nicht mehr aktuell sein können, sollten die Anfragen an den autorisierten Nameserver der Zone des Opfers gestellt werden. Die Adresse eines autorisierten Nameservers kann entweder aus den Daten der Whois-Abfragen oder aus den Ressource Record Typen „ns“ oder „soa“ gewonnen werden.

Ressource Records (vgl. [Albitz01:19]) sind die mit einem Domainnamen assoziierten Daten. Sie kommen in den folgenden Typen vor:

- Typ A (Address)

  Enthält die Adresse der Domain[39].

- MX (Mail Exchanger)

  Enthält die Adresse eines Mailservers der Domain. Zu einer Domain kann es mehrere MX-Einträge geben.

- CNAME (Canonical Name)

  Gibt den primären bzw. kanonischen Namen für einen Host aus. Besitzt der Webserver den Namen Server1, so können alle Anfragen an den www die Adresse von Server1 zurückgeben. Server1, der reale Name für www, ist dabei der kanonische Name.

- HINFO (Hostinfo)

  Hier können Informationen über Prozessortyp sowie das verwendete Betriebssystem enthalten sein.

- TXT (Text)

  Enthält beliebige Kommentare.

- NS (Nameserver)

  Gibt die Adresse der für die Zone autorisieten Nameserver an.

- SOA (Start of Authority)

  Enthält unter anderem folgende allgemeine Informationen über die Zone (vgl. [RFC1035:19f.] :
  - MNAME enthält den Server, von dem die Informationen bezogen wurden
  - RNAME enthält die Emailadresse des zuständigen Administrators (responsible Mail-adress), wobei das @ durch den letzten Punkt dargestellt ist
  - SERIAL gibt die Seriennummer

  Zusätzlich sind Informationen zum Update der Zoneninformationen enthalten.

[39] Eine Abfrage mit nslookup ohne Parameter führt eine Abfrage nach Typ A durch.

Die Abfrage der Daten kann mittels des Tools nslookup durchgeführt werden, das von jedem gängigen Betriebssystem mitgeliefert wird. Sie ist wichtig für die Erkennung von Systemen. Um Aufwand zu ersparen, können die Ressource Records A, MX und NS zusammen mittels des Typs ANY abgefragt werden. Der Typ HINFO kann einem Angreifer Informationen über ein System wie z. B. das verwendete Betriebssystem liefern, wodurch ihm die Auswahl des richtigen Exploits für den Angriff erleichtert wird. Daher wird der Ressource Record HINFO heute kaum noch verwendet.

Um einfacher an die Informationen zu kommen, kann ein Zonentransfer durchgeführt werden. Wenn aus Gründen der Lastverteilung mehrere autorisierte Nameserver existieren, braucht nur der primäre Server gepflegt zu werden. Die anderen für die Zone autorisierten Server erlangen diese Informationen über Zonentransfers, wobei sämtliche Informationen der Zone von einem Host zum anderen kopiert werden. Während für die Abfragen der Port 53/udp genutzt wird, nutzen Zonentransfers den Port 53/tcp. Sofern dieser nicht gesperrt ist, können die gesamten Informationen über die Zone vom autorisierten Nameserver auf den Rechner kopiert werden. Dies kann mittels des `ls` Kommandos des nslookup clients geschehen.

Ist ein Zonentransfer nicht möglich, so können Reverse-Abfragen über den gesamten Adressbereich durchgeführt werden, um alle im DNS eingetragenen Systeme zu identifizieren. Sind einem Namen mehrere IP-Adressen zugeordnet, so wird eine Anfrage aus Lastgründen von mehreren Maschinen bedient. Diese Information ist für einen Angreifer wichtig, da hierbei verschiedene Pakete eines Angriffes an unterschiedliche Maschinen geleitet werden, wodurch der Angriff erfolglos bleiben könnte.

In den gewonnenen Informationen sollten keine Systeme wie Router oder Firewall eingetragen sein. Sie würden einem Angreifer die Erstellung eines Netwerkplans erleichtern. Dennoch können diese Systeme durch Verwendung des Tools traceroute erkannt werden. Das Tool ist Inhalt des folgenden Abschnitts.

Traceroute

Traceroute ist ein Diagnose Tool, mit dem die Route zu einem Ziel identifiziert werden kann. Es ist eigentlich für administrative Zwecke gedacht. Bei der aktiven Reconnaissance kann diese Diagnose dazu genutzt werden, weitere Rechner zu bestimmen, die nicht mit den vorhergehenden passiven Methoden identifiziert worden sind. Daneben kann durch die Bestimmung der Route auch die Lage der Rechner in einem Netz festgestellt werden. Die hier gewonnenen Informationen dienen der Erstellung eines Netzwerkplans.

Technisch nutzt jede Traceroute-Implementation die Behandlung des Time-to-Live Feldes, kurz TTL, des IP-Headers. Auf dem Weg zu einem Ziel wird das TTL-Feld von jedem Router dekrementiert. Erreicht das Feld bei einem Router den Wert 0, so antwortet der Router mit der ICMP Meldung „ICMP TIME_EXCEEDED“ (Typ 11) und gibt sich damit zu erkennen. Das Traceroute-Programm generiert eine Reihe von Paketen, wobei das TTL-Feld des ersten Pakets den Wert 1 hat. Bei jedem weiteren versendeten Paket wird das TTL Feld inkrementiert, bis das Ziel erreicht ist.

Die von dem Traceroute-Programm generierten Pakete unterscheiden sich in den unterschiedlichen Implementationen. So generiert das mit Windows gelieferte `tracert`[40] ICMP_ECHO_REQUEST (Typ 8) Nachrichten. Die UNIX Variante `traceroute` generiert bei Standardeinstellungen UDP-Pakete, deren Port sich aus 33434[41]+n berechnet, wobei n die Nummer der verschickten Pakete ist. Somit wird jedes Paket an einen anderen Port geschickt. Mittels eines Patch[42] kann die Portnummer mit einem festen Wert belegt werden. Mit dem Schalter `-I` verwenden die lbl-Varianten die Funktionsweise des in Windows enthaltenen tracert.

In Abbildung 27 ist die Ausgabe von traceroute unter UNIX dargestellt. Hier ist zu erkennen, dass auf dem Weg zum Ziel 192.168.1.2 der Host mirror.irt.local als Gateway dient. Abbildung 28 stellt die Ausgabe von tcpdump auf dem Quellsystem (192.168.0.17) dar.

```
# traceroute 192.168.1.2
traceroute to 192.168.1.2 (192.168.1.2), 30 hops max, 40 byte packets
1  mirror (192.168.0.249)  5.992 ms  0.306 ms  0.281 ms
2  192.168.1.2 (192.168.1.2)  0.818 ms  0.605 ms  0.601 ms
```

Abbildung 27: Ausgabe von traceroute zum Ziel 192.168.1.2

```
192.168.0.17.34199 > 192.168.1.2.33435:  udp 12 [ttl 1]
mirror.irt.local > 192.168.0.17: icmp: time exceeded in-transit [tos 0xc0]
192.168.0.17.34199 > 192.168.1.2.33436:  udp 12 [ttl 1]
mirror.irt.local > 192.168.0.17: icmp: time exceeded in-transit [tos 0xc0]
192.168.0.17.34199 > 192.168.1.2.33437:  udp 12 [ttl 1]
mirror.irt.local > 192.168.0.17: icmp: time exceeded in-transit [tos 0xc0]
192.168.0.17.34199 > 192.168.1.2.33438:  udp 12
192.168.1.2 > 192.168.0.17: icmp: 192.168.1.2 udp port 33438 unreachable
192.168.0.17.34199 > 192.168.1.2.33439:  udp 12
192.168.1.2 > 192.168.0.17: icmp: 192.168.1.2 udp port 33439 unreachable
192.168.0.17.34199 > 192.168.1.2.33440:  udp 12
192.168.1.2 > 192.168.0.17: icmp: 192.168.1.2 udp port 33440 unreachable
```

Abbildung 28: tcpdump Ausgabe auf Gateway mirror.irt.local

Da ein Paket von der Quelle zum Ziel im Internet nicht immer denselben Weg nimmt, ist es ratsam, ein Traceroute zu einem System mehrmals auszuführen. Somit können mehr Systeme identifiziert werden. Eine weitere Methode zum Erkennen aktiver Systeme ist der Ping Sweep.

Ping Sweep

Ein Ping ist das Versenden eines ICMP_ECHO (Typ 8) Nachricht an ein anderes System. Ist dieses System verfügbar, so antwortet es mit einem ECHO_REPLY (Typ 0), anderenfalls mit

[40] Programmname ist wegen Kompatibilität zur 8.3-Namenskonvention abgekürzt.

[41] Dieser Port ergibt sich im Quellcode aus 32768+666. Die Wahl dieser Portnummer wird in der Manpage von `traceroute` begründet: „Traceroute hopes that nothing is listening on UDP ports base to base + nhops – 1 at the destination host (so an ICMP PORT_UNREACHABLE message will be returned to terminate the route tracing).”

[42] Verfügbar unter http://www.packetfactory.net/firewalk/dist/traceroute/

ICMP_DESTINATION_UNREACHABLE (Typ 3). Dieses Verfahren wurde von Mike Muuss, dem Autor des Ping Programms, analog zu einem Sonar entwickelt (vgl. [Muuss97]). Ein Sonar sendet dabei einen Ton, der sich für den Menschen wie das Wort „ping“ anhört. Auf Grund der Echos kann das Sonar ein Bild der Umgebung erzeugen.

Diese Verfahrensweise ist in den gängigen Betriebssystemen mit dem Befehl `ping` implementiert, der auf einen einzelnen Host angewendet werden kann. Ein Angreifer nutzt das Verfahren nicht nur für ein einzelnes System, sondern für einen ganzen Netzbereich. Durch solch einen systematischen Ping Scan, meist Ping Sweep genannt, kann der Angreifer feststellen, welche Systeme im Netzwerk zurzeit aktiv sind. Zudem kann er bisher unbekannte Systeme entdecken. Ein solcher Ping Scan kann mittels spezieller Tools durchgeführt werden. Manche Portscanner bieten ebenfalls diese Funktionalität.

Bei der Implementation eines Ping-Programms können neben der Verwendung von ICMP Nachrichten auch TCP oder UDP-Pakete verwendet werden (vgl. [Kurtz01:66ff.]). Dabei wird auf Anwesenheit eines bestimmten Ports geprüft. Neben Portscannern kann dazu auch das Tool hping verwendet werden.

Die bisherigen Ergebnisse der Discovery fließen in die Erstellung eines Netzwerkplans (engl.: network map) ein.

Network Map

Sind die oben genannten Schritte durchlaufen, so ist die Discovery abgeschlossen. Aus den gewonnenen Daten kann der Angreifer einen Plan des Netzwerkes (engl.: network map) erstellen, der ihm als Übersicht für weitere Schritte dient.

Mit dem Netzwerkplan erstellt der Angreifer für spätere Schritte eine Übersicht über das Netzwerk des Ziels. Die Erstellung des Netzwerkplans (engl.: network map) wird in der Fachliteratur mit dem Begriff „Network Mapping“ (vgl. [SP800-42:11])[43] bezeichnet. Dabei wird das Netz mittels Traceroute und Pingscans vermessen, weshalb die Erstellung des Plans nach [OSSTMM02:20] auch Netzwerkvermessung (engl.: network survey) genannt wird.

Der erstellte Plan ist aber noch nicht vollständig, da die Eigenschaften der erkannten Systeme noch nicht vollständig bekannt sind. Anhand der Einträge in der DNS-Datenbank kann auf die Eigenschaften mancher Systeme geschlossen werden. So kann durch das im MX-Record eingetragene System auf den Mailserver geschlossen werden. Über die Funktion der Systeme, die nur durch Name oder IP-Adresse bekannt sind, kann noch keine Aussage getroffen werden. Zudem sind die Annahmen dieser Phase zu verifizieren, wozu die Eigenschaften der in der Discovery erkannten Systeme ermittelt werden, was Aufgabe der Enumeration ist.

[43] In [SP800-42] beinhaltet der Begriff “Network Mapping“ auch die Anwendung der Methoden der Enumeration. In Anlehnung an [Kurtz00] und [Payne01] sollen Discovery und Enumeration voneinander abgegrenzt werden.

5.3.2. Enumeration

Nachdem ein Angreifer die Systeme erkannt hat, versucht er so viele Informationen wie möglich über jedes einzelne System herauszufinden (vgl. [Kurtz00:4]). Dabei sollen folgende Informationen erkannt werden:

- Angebotene Dienste und deren Ports
- Vorhandene Betriebssysteme
- Vorhandene Applikationen und Version
- Auf dem System vorhandene Benutzer

Aus diesen Daten will ein Angreifer einen Punkt finden, über den er in das Netzwerk eindringen kann.

Der Begriff Enumeration lässt sich von der „Turing Enumeration“ herleiten, bei der alle von einer Turing Maschine erzeugbaren Wörter aufgezählt werden. In der Reconnaissance werden alle erkennbaren Eigenschaften eines Systems aufgezählt, um den Netzwerkplan zu verfeinern. Dazu stehen dem Angreifer die folgenden Methoden zur Verfügung:

- OS Fingerprint
- Portscan
- Banner Grabbing
- User Information

Sie werden im Folgenden näher erläutert.

OS-Fingerprint

Um genauere Aussagen über die Funktionalität machen zu können, ist zunächst das Betriebssystem festzustellen. Damit können in späteren Schritten die angebotenen Dienste und die dafür verwendete Software näher bestimmt werden. Die Technik zum Erkennen des Betriebssystems heißt nach [Fyodor02] OS-Fingerprinting und kann in passives und aktives Fingerprinting unterschieden werden. Passives Fingerprinting wertet dazu die Daten aus, die beim Belauschen des Netzwerkverkehrs gesammelt wurden. Aktives Fingerprinting schickt Anfragen an das Zielsystem und wertet die Reaktionen aus, um auf das Betriebssystem schließen zu können. Das aktive Fingerprinting ist in [Fyodor02] genauer beschrieben.

Die Betriebssysteme verschiedener Hersteller und Versionen beantworten Paketanfragen auf unterschiedliche Art und Weise. So unterscheiden sich die Parametrisierungen der Pakete. Auf manche Pakete erfolgt bei bestimmten Betriebssystemen keine Reaktion. Solche Eigenheiten der Betriebssysteme werden beim Fingerprinting ausgenutzt. Genauer wird das Verhalten des TCP/IP-Stacks analysiert, weshalb die Methode auch Stack-Fingerprinting genannt wird (vgl. [Kurtz01:95ff.]). Dazu werden unter Anderem das Verhalten auf fehlerhafte Pakete, die Behandlung der Fragmentierung oder die Inhalte von Paketfeldern untersucht. So gibt es unterschiedliche Belegungen des Type-of-Service-Felds oder des TTL-Feldes des IP-Headers.

Für aktives Fingerprinting können Tools wie nmap und queso verwendet werden, passives Fingerprinting kann mit dem Tool siphon durchgeführt werden. Die von den Tools verwendeten Methoden erlauben lediglich einen Schluss auf das verwendete Betriebssystem. So kann nmap die Systeme Windows 2000, XP und ME nicht voneinander unterscheiden, was in Abbildung 29 dargestellt ist.

```
# nmap -P0 -O -p 134-135 192.168.0.1

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on www.irt.local (192.168.0.1):
(The 1 port scanned but not shown below is in state: closed)
Port       State       Service
135/tcp    open        loc-srv
Remote operating system guess: Windows Millennium Edition (Me), Win
XP

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Abbildung 29: OS-Fingerprinting eines Windows 2000 Rechners mit nmap

Eine andere Möglichkeit, das Betriebssystem genauer zu erkennen, bietet sich im Falle eines aktivierten SMNP Dienstes. Mittels des passenden Object Identifiers, kurz OID, kann die Systembeschreibung abgefragt werden, die das verwendete Betriebssystem enthält. Der OID für die Systembeschreibung heißt `sysDescr` und hat die Nummer 1.3.6.1.2.1.1.1. Die Abfrage und das Ergebnis mittels des Tools snmptool ist in Abbildung 30 dargestellt. Dabei sei angemerkt, dass SNMP-Dienst bei den meisten Betriebssystemen manuell installiert werden muss.

```
C:\>snmptool walk 192.168.0.1 public .1.3.6.1.2.1.1.1

SnmpTool - Simple Network Management Protocol Tool for Win32


Performing branch walk starting at OID 1.3.6.1.2.1.1.1


      Name: system.sysDescr.0
      OID: 1.3.6.1.2.1.1.1.0
      Type: OCTET STRING
      Length: 127
      Value: Hardware: x86 Family 15 Model 2 Stepping 4 AT/AT COMPATIBLE –
                  Software:  Windows  2000  Version  5.0  (Build  2195
Uniprocessor Free)


End of MIB branch
```

Abbildung 30: Abfrage des Betriebssystems mittels snmptool

Portscan

Unter einem Scan wird der systematische Zugriff auf eine Entität verstanden (vgl. [Howard98:10]). Ein Portscan ist dabei der systematische Zugriff auf die Ports. Dabei soll erkannt werden, ob ein Port offen oder geschlossen ist. Ist ein Port offen, so kann durch ihn eine Kommunikationsbeziehung aufgebaut werden. Somit dient ein Portscan dem systematischen Erkennen des Status der Ports eines oder mehrerer Systeme.

Der Zugriff kann dabei grundsätzlich auf TCP- und auf UDP-Ports erfolgen. Dabei gibt es verschiedene Scantypen, die nach der Art des Zugriffes auf den Port unterschieden werden. Im Folgenden wird eine Auswahl von Scantypen nach [Fyodor97] erläutert.

TCP-CONNECT Scan: Hierbei wird ein vollständiges 3-Way-Handshake durchgeführt. Ist der Verbindungsaufbau erfolgreich, so ist der TCP-Port offen. Dieser Scan-Typ ist der schnellste.

TCP-SYN Scan: Dieser Scantyp sendet lediglich ein SYN-Paket. Wird es mit SYN/ACK beantwortet, so ist der TCP-Port als offen erkannt und der Scanner sendet sofort ein RST-Paket, um die Verbindung zu beenden. Da das erste ACK-Paket vom Initiator der Verbindung nicht gesendet wird, wird der Scan auch als „Half-open-Scan bezeichnet“. Bei einem geschlossenen Port sendet der Empfänger ein RST/ACK.

TCP-FIN Scan: Bei diesem Typ wird ein FIN-Paket an einen TCP-Port geschickt. Einige Implementierungen des TCP-Protokollstapels senden nur bei einem geschlossenen Port ein RST-Paket zurück. Empfängt der Portscanner ein RST-Paket, so gilt der geprüfte Port als geschlossen. Diese Technik funktioniert nicht bei Windows-Systemen, da diese grundsätzlich ein RST-Paket zurückschicken.

TCP Xmas-Tree-Scan: Ein solcher Scan funktioniert ähnlich dem FIN Scan, wobei zusätzlich das URG und das PUSH-Flag gesetzt werden. Auch diese Scans funktionieren bei Windows Betriebssystemen nicht.

TCP Null-Scan: Charakteristika dieses Scantyps sind TCP-Pakete, bei denen alle Flaggen den Wert 0 haben. Hierbei antworten geschlossene Ports bei einigen Betriebssystemen mit einem RST. Wie TCP-FIN und TCP-Xmas-Tree-Scans funktionieren die TCP-Nullscans nicht bei Windows Betriebssystemen.

TCP-ACK-Scan: Dieser Scantyp prüft nicht den direkten Status eines Ports auf einem System, sondern gibt ein Anzeichen darauf, ob der Port durch eine Firewall gefiltert wird. Dazu wird ein TCP-ACK-Paket an den Host geschickt. Wird dieses nicht beantwortet, so kann daraus geschlossen werden, dass der Port durch eine Firewall gefiltert wird.

UDP-Scan: Wird ein UDP-Paket an einen geschlossenen UDP-Port geschickt, so antwortet dieser mit einem ICMP_DESTINATION_UNREACHANBLE (Typ 3). Werden diese ICMP-Meldungen allerdings gefiltert, so werden bei einem UDP-Scan alle UDP-Ports als offen angezeigt.

Ein Beispiel für die Durchführung eines Portscans ist in Abbildung 31 dargestellt.

```
#nmap –P0 –sS mirror.irt.local
Starting nmap 3.20 ( www.insecure.org/nmap/ ) at 2003-05-13 14:12 CEST
Interesting ports on mirror.irt.local (192.168.0.249):
(The 1608 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
80/tcp     open        http

Nmap run completed -- 1 IP address (1 host up) scanned in 0.682 seconds
```

Abbildung 31: Ausgabe eines Stealth-SYN-Scan mit nmap

An Hand der offenen Ports kann auf einen bestimmten Dienst geschlossen werden, den ein System bereitstellt. Da ein Dienst über jeden beliebigen Port angesprochen werden kann, kann an einem Port auch ein anderer Dienst betrieben werden. Daher müssen die Dienste verifiziert werden. Für einen erfolgreichen Angriff muss auch die Applikation, die den Dienst bereitstellt, identifiziert werden. Diese Aufgaben werden mit der „Service and Application enumeration“ bewältigt.

Netzwerkmaske erkennen

Zur Erstellung des Netzwerkplans kann der Adressraum verfeinert werden und die Subnetzadressen der einzelnen Hosts erkannt werden. So kann mittels der ICMP-Nachrichten „AM1: Address Mask Request“ (Typ 17) und „AM2: Address Mask Reply“ (Typ 18) die Netzwerkmaske erkannt werden (vgl. [Kurtz01:71f]).

Service and Application enumeration

Nachdem die offenen Ports identifiziert sind, kann anhand der Liste der so genannten „Well-known portnumbers“[44] auf den Dienst geschlossen werden, der auf dem Port hört. Die Liste besagt, auf welchem Port ein Dienst erreichbar sein soll. Sie findet sich bei UNIX in der Datei /etc/services und unter %WINDIR%/system32/drivers/etc/services bei Windows NT/2000/XP.

Allerdings ist nicht bekannt, welcher Dienst wirklich an den Port gebunden ist. So kann ein Webserver auch an einen beliebigen anderen Port gebunden werden. Demonstriert wird diese Möglichkeit durch das Programm echofake, dessen Quellcode im Anhang G2 enthalten ist. Echofake implementiert einen Echoserver, der allerdings an den Port 21/tcp gebunden ist. Dieser Port wird nach den „Well-known portnumbers“ von ftp genutzt. Abbildung 32 stellt die Ausgaben der Programme nmap und amap dar. Amap ist ein Tool, mit dem die an einen Port gebundene Anwendung erkannt werden kann. Während nmap den zum offenen Port 21/tcp als ftp-Dienst erkennt, identifiziert amap den an diesen Port gebundenen Echo-Dienst.

[44] http://www.iana.org/cgi-bin/sys-port-number.pl

```
# nmap -sS -p 21 -P0 mirror.irt.local

Starting nmap 3.20 ( www.insecure.org/nmap/ ) at 2003-05-23 12:20 CEST
Interesting ports on mirror.irt.local (192.168.0.249):
Port       State       Service
21/tcp     open        ftp

Nmap run completed -- 1 IP address (1 host up) scanned in 0.048 seconds

# amap -sT -v -1 mirror.irt.local 21
Using trigger file ./appdefs.trig
Using response file ./appdefs.resp
Total amount of tasks to perform: 16
Amap v2.1 started at Fri May 23 12:23:31 2003, stand back and keep
children away
Protocol on IP 192.168.0.249 port 21 tcp matches echo
1 responses received in total for 16 tasks.
Unidentified ports: none.
Amap v2.1 ended at Fri May 23 12:23:32 2003
```

Abbildung 32: Erkennung des Echo-Dienstes auf Port 21/tcp

Wichtiger für einen Angreifer ist die Erkennung des Softwareprodukts und dessen Version und welche Anwendung den Dienst bereitstellt. So kann zwar bei einem geöffneten Port 80 auf einem Windowssystem auf den IIS-Webserver geschlossen werden, jedoch ist auch der Apache unter Windows verfügbar. Auch die Version der Anwendung ist unbekannt. All diese Daten sind für den Schluss auf vorhandene Schwachstellen wichtig. Der Angreifer benötigt diese Informationen, um den korrekten Exploit anzuwenden.

Eine Methode zur Erkennung des an einen Port gebundenen Diensts sowie des Softwareprodukts und dessen Version ist das Banner Grabbing, worunter nach [SP800-42:15] der Prozess der Erlangung der Banner-Informationen verstanden wird. Das Banner ist ein Informationstext, den ein Server bei Verbindungsaufbau überträgt, der aber nicht immer für den Endnutzer sichtbar ist. Er enthält häufig Informationen über verwendete Anwendung und Version. Der Begriff Banner wurde bereits in der Cert Advisory CA-1988-1 verwendet.

Um diesen zu erlangen, kann mit Telnet eine Verbindung zu dem Port aufgebaut werden. Abbildung 33 stellt das Banner Grabbing mittels telnet für SMTP dar. Hierbei kann neben der Software sendmail 8.11.6 auch das Betriebssystem SuSE Linux erkannt werden. Abbildung 34 stellt das Banner Grabbing für http dar.

```
# telnet 192.168.0.249 25

Trying 192.168.0.249...
Connected to mirror.irt.local.
Escape character is '^]'.
220 mirror.irt.local ESMTP Sendmail 8.11.6/8.11.6/SuSE Linux 0.5; Tue, 13
May 2003 12:30:22 +0200
221 2.0.0 mirror.irt.local closing connection
```

Abbildung 33: Banner Grabbing mittels telnet bei SMTP

```
# telnet 192.168.0.249 80

Trying 192.168.0.249...
Connected to mirror.irt.local.
Escape character is '^]'.
HTTP/1.1 200 OK
Date: Tue, 13 May 2003 10:35:38 GMT
Server: Apache/1.3.20 (Linux/SuSE)
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
...
```

Abbildung 34: Banner Grabbing mittels telnet bei http

Das Banner Grabbing kann auch mit dem Tool Netcat durchgeführt werden. Der Befehl „`C:\>nc mirror.irt.local 80 | find /i Server`“ würde nur die Zeile `Server: Apache/1.3.20 (Linux/SuSE)`ausgeben. Netcat ist sowohl für Unix als auch für Windows verfügbar.

Für das Auslesen des Banners können auch spezielle Tools genutzt werden. Ein Tool ist amap, das weiter oben bereits erwähnt worden ist. Auch der Portscanner SuperScan führt das Banner Grabbing automatisch durch (vgl. Abbildung 35).

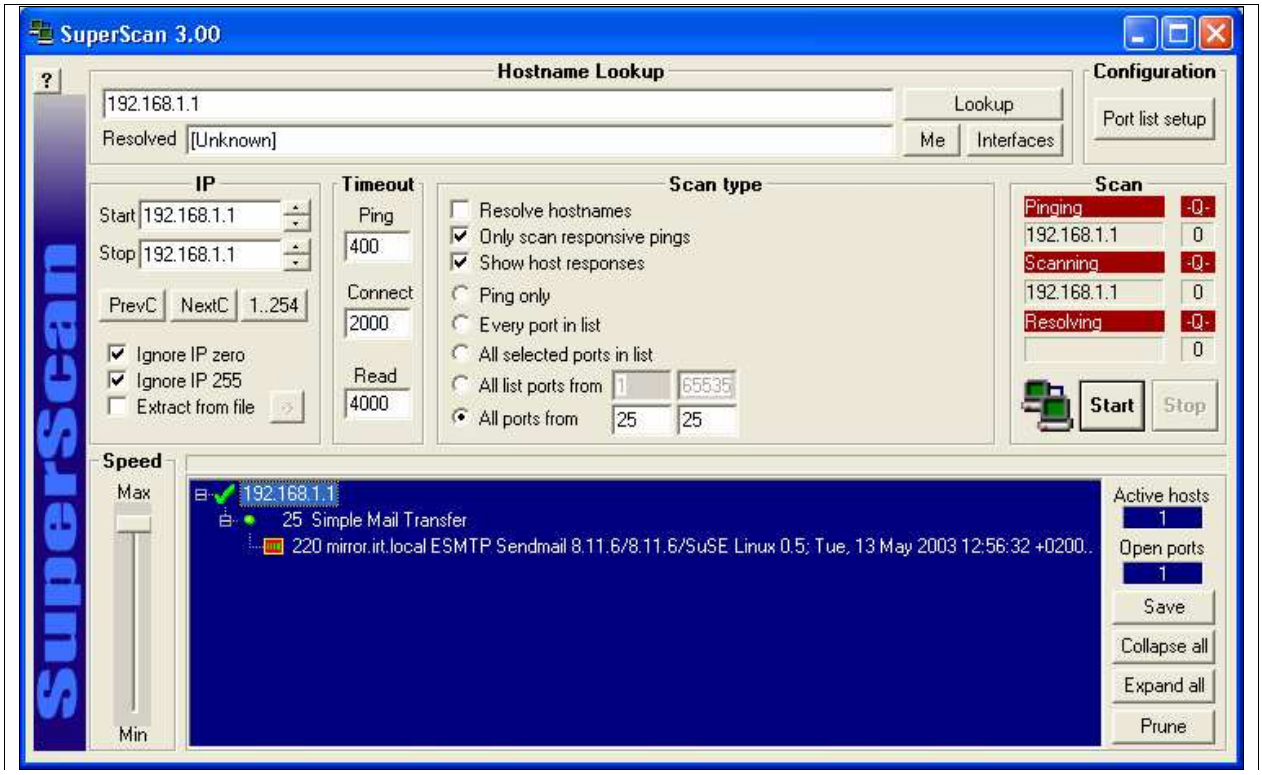


Abbildung 35: Automatisches Banner Grabbing mit dem Portscanner SuperScan

Nicht jeder Dienst kann durch ein Banner Grabbing erkannt werden, da einige Dienste über keinen Banner verfügen. Kann durch das Banner Grabbing der Dienst nicht erkannt werden, so besteht auf Windows-Systemen, auf denen ein SNMP-Dienst betrieben wird, die Möglichkeit, über einen SNMP-Request die von dem System betriebenen Dienste zu erkennen. Dadurch können neben lokalen Diensten auch Dienste erkannt werden, die ein Portscanner eventuell nicht erkannt hat. Dafür sind alle Kindsknoten des OID .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr2.server.svSvcTable.svSvcEntry.svSvcName (1.3.6.1.4.1.77.1.2.3.1.1) zu durchlaufen.

```
svSvcName.9.84.101.108.101.102.111.110.105.101 Telefonie
svSvcName.10.68.78.83.45.67.108.105.101.110.116DNS-Client
svSvcName.10.84.97.115.107.112.108.97.110.101.114    Taskplaner
svSvcName.11.68.72.67.80.45.67.108.105.101.110.116   DHCP-Client
svSvcName.11.80.108.117.103.32.38.32.80.108.97.121   "Plug & Play"
svSvcName.11.83.78.77.80.45.68.105.101.110.115.116   SNMP-Dienst
```

Abbildung 36: SNMP Abfrage nach Diensten (gekürzte Ausgabe)

Mit den bis hier gewonnenen Informationen kann die Aufgabe eines Systems bestimmt werden. Somit kann der Netzwerkplan vervollständigt werden. Zudem verfügt der Angreifer über weitere Informationen wie die Betriebssystemsoftware und -version. Hat der Angreifer das Ziel, in ein System einzubrechen, so ist die Kenntnis über Benutzernamen und Verzeichnisfreigaben notwendig, mit denen ein Angreifer direkten Zugang zu einem System erhalten kann.

Freigaben und Benutzernamen

Benutzernamen und Freigaben können ebenfalls über SNMP herausgefunden werden. Zudem kann der Rechnername des Zielsystems erkannt werden. Die zugehörigen Object Identifier finden sich in Tabelle 4.

OID Nummer	OID Bezeichnung	Zweck
1.3.6.1.2.1.1.5	.iso.org.dod.internet.mgmt.mib-2.system.sysName	Rechnername
1.3.6.1.4.1.77.1.2.27	.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr2.server.svShareTable	Freigaben
1.3.6.1.4.1.77.1.2.25	.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr2.server.svUserTable	Benutzernamen

Tabelle 4: Objekt Identifier für Aufgaben der Reconnaissance

Leider können mittels SNMP keine versteckten Freigaben erkannt werden. Solche Freigaben enden mit dem Zeichen $ und werden bei einer Standardinstallation für jedes Laufwerk angelegt. Daher ist eine Betrachtung der Möglichkeiten von NetBIOS und SMB notwendig.

Die Schwachstelle in SMB und seinem Nachfolger CIFS, durch die solche Informationen erkannt werden können, bilden die so genannten Null-Sessions, deren Charakteristikum die fehlende Authentifizierung ist. So muss weder ein Benutzername noch ein Passwort

angegeben werden. Ist die Null-Session aufgebaut, so können Informationen über Freigaben und Benutzernamen ausgelesen werden. So kann mittels des Windows Befehls `net view` die auf einem System verfügbaren Freigaben angezeigt werden. Dabei können aber keine versteckten Freigaben oder Benutzernamen angezeigt werden. Um dieses Ziel zu erreichen, werden weitere Tools benötigt, die frei im Internet verfügbar sind. Ein Beispiel für solche Tools ist `enum`[45], dessen Ausgabe in Abbildung 37 dargestellt ist.

```
C:\>enum -U –S -P -G –L www
server: www
setting up session... success.
password policy:
  min length: none
  min age: none
  max age: 42 days
  lockout threshold: none
  lockout duration: 30 mins
  lockout reset: 30 mins
opening lsa policy... success.
server role: 3 [primary (unknown)]
names:
  netbios: WWW
  domain: ARBEITSGRUPPE
trusted domains:
  indeterminate
netlogon done by a PDC server
getting user list (pass 1, index 0)... success, got 4.
  Administrator  Gast  IUSR_WWW  IWAM_WWW
enumerating shares (pass 1)... got 5 shares, 0 left:
  IPC$  Z$  test  ADMIN$  C$
Group: Administratoren
WWW\Administrator
Group: Benutzer
NT-AUTORITÄT\INTERAKTIV
NT-AUTORITÄT\Authentifizierte Benutzer
Group: Gäste
WWW\Gast
WWW\IUSR_WWW
WWW\IWAM_WWW
Group: Hauptbenutzer
Group: Replikations-Operator
Group: Sicherungs-Operatoren
cleaning up... success.
```

Abbildung 37: Gekürzte Ausgabe des Tools enum

In der in Abbildung 37 dargestellten Ausgabe sind der Name des Host 192.168.0.1 die Freigaben und Benutzernamen sowie die Benutzergruppen zu erkennen. Zudem ist die Passwordpolicy erkennbar.

Die Informationen über Benutzernamen kann der Angreifer in der Penetrationsphase dazu verwenden, zugehörige Passwörter zu erraten. Hat er das Passwort für den Administrator erlangt, so kann er über eine Verbindung zu einer erkannten Freigaben auf Dateisysteme des

[45] Quelle: razor.bindview.com

Opfersystems direkt zugreifen und die gespeicherten Daten manipulieren. Zudem kann er über die ADMIN$ Freigabe auf das System zur Durchführung administrativer Aufgaben zugreifen, wozu das Starten von Diensten oder Veränderungen an der Registry gehören.

Neben den besprochenen Methoden können Benutzernamen auch über RPC, LDAP, Active Directory, NIS oder finger ermittelt werden. Bei SMTP-Mailservern besteht durch die Befehle EXPN und VRFY die Möglichkeit, Benutzernamen zu erkennen. Diese Befehle sind heutzutage bei den meisten Mailservern abgestellt.

Auswirkungen der Reconnaissance

Die Reconnaissance an sich verursacht nur in seltenen Fällen Schäden. So kann es bei sehr wenigen Systemen vorkommen, dass es einem Port Scan nicht widersteht und abstürzt. Die Informationen, die in der Reconnaissance erkannt werden, können aber genutzt werden, um einen Ansatzpunkt für einen Angriff zu finden, der zu einem Schaden für die Assets führen kann. Daher sollten Maßnahmen gegen die Methoden der Reconnaissance in die Sicherheitskonzepte einfließen, um einem Angreifer sein Vorgehen zu erschweren.

5.4. Vulnerability Detection

In der Phase der Vulnerability Detection informiert sich der Angreifer über Schwachstellen, die in den während der Reconnaissance erkannten Systemen auf Grund ihrer Charakteristika vorhanden sind. Dabei bildet er eine Relation zwischen Systemcharakteristika und Schwachstellen. Die Relation ist jedoch keine Abbildung von Charakteristika auf Schwachstellen, wie es in [Kurtz00:4] dargestellt wird, da sonst ein Charakteristikum auf mehrere Schwachstellen abgebildet werden könnte, was von der Definition der Abbildungen (vgl. [Biggs89:27]) nicht möglich ist.

Die Bildung einer Relation ist jedoch notwendig. So kann der Angreifer sichergehen, dass er seine Mühen nicht in einen Exploit investiert, dessen zugehörige Schwachstelle in dem Opfersystem nicht vorhanden ist. Nach [Kurtz00:4] lässt sich ein solches Verhalten bei Skript Kiddies beobachten. Sie bilden keine Relation und wenden so Exploits, deren Aufgabe die Ausnutzung von Schwachstellen in Unix Systemen ist, gegen Microsoft Windows Systeme an.

Bei der Bildung der Relation ist der Inhalt der Menge der Schwachstellen näher zu betrachten. Alle in einem System vorhandenen Schwachstellen zu finden, ist nicht von einem einzelnen zu leisten. Dazu müssten sämtliche System-Zustände auf Fehler überprüft werden. Die Menge aller Systemzustände ist zu groß, um sämtliche Fehler zu finden.

Ob ein Angreifer potentielle Schwachstellen betrachtet, ist abhängig von seiner Motivation. So haben Geheimdienste sowohl die Motivation als auch die notwendigen Fähigkeiten hinsichtlich Personal, Zeit und Ausbildung. Ein Schutz gegen solche Angreifer bedarf eines ungemein höheren Aufwands als der Schutz gegen blackhat Hacker (vgl. Abschnitt 2.2) mit maliziöser Absicht, die von Moyer (vgl. [Moyer98:3]) als „Sports Intruders“ klassifiziert werden. Sie benutzen oder entwickeln Tools für Angriffe, die auch von einem Tiger Team

genutzt werden. Dabei verfügen die Hacker über keine Methodiken zum Auffinden von potentiellen, bisher unbekannten Schwachstellen. Sollten sie doch eine bisher unbekannte Schwachstelle entdecken, so tauschen sie sich über bestimmte Foren des Internets aus, um so beispielsweise eine Bestätigung zu bekommen. Auf diesem Wege wird die Schwachstelle bekannt, da auch Sicherheitsexperten diese Foren besuchen. Ein Angreifer, der dem Grad eines blackhat Hackers entspricht, nutzt daher nur bekannte Schwachstellen aus.

Quellen für die Informationen über bekannte Schwachstellen bieten die Schwachstellenarchive der Hersteller sowie unabhängiger Organisationen wie Nist, das mit ICAT[46] ein Meta-Archiv über die Archive anderer Organisationen zur Verfügung stellt. Neben solchen Archiven können auch Mailinglisten, Newsgroups oder Foren bestimmter Seiten zur Informationssuche genutzt werden. Auch Fachliteratur kann einem Angreifer Ansatzpunkte liefern. Eine Basis hat sich der Angreifer dabei schon in der Reconnaissance geschaffen, wo er bereits Schwachstellen ausnutzte, um sich über die Systeme und ihre Eigenschaften zu informieren. So bildete die Existenz der öffentlich zugängliche Community „public“ eine Schwachstelle im SNMP-Dienst, durch die Erkenntnisse über die Systemeigenschaften in der Reconnaissance gewonnen werden konnten. Solche Schwachstellen können zum Teil für weiterführende Ziele ausgenutzt werden.

Ein Angreifer betrachtet aber nicht alle bekannten Schwachstellen in einem System, sondern nur diejenigen, die ihm für das Erreichen seines Zieles nützen (vgl. [HiSolutions03:8]). So nützen ihm keine Schwachstellen, deren Ausnutzung zu einem Denial-of-Service führen, wenn ein Angreifer Zugriff auf vertrauliche Daten erlangen will. Zudem bedeutet die Ausnutzung einer Schwachstelle einen Aufwand für den Angreifer. Um den Aufwand so gering wie möglich zu halten, sucht der Angreifer den schwächsten Punkt, der durch die am einfachsten auszunutzende Schwachstelle gegeben ist. Hat er Erfolg, so bleiben alle anderen Schwachstellen vom Angreifer unbeachtet. Ein Angreifer führt daher keine Schwachstellenanalyse durch, deren Ziel die Erkennung *sämtlicher* Schwachstellen ist.

Während ein Penetrationstest in der Softwareentwicklung versucht, potentielle Schwachstellen aufzuspüren (vgl. Abschnitt 3.4.1), untersucht der hier betrachtete Penetrationstest zur Überprüfung eines Sicherheitskonzeptes die Auswirkungen eines Angriffes aus der Sichtweise eines Hackers, wobei nur öffentlich bekannte Schwachstellen behandelt werden.

Ein Tiger Team arbeitet nur selten unter realen Bedingungen. So arbeitet es meist in einem engeren Zeitrahmen (vgl. [Moyer98:2]), so dass es auffälligere Log-Einträge erzeugt, als es ein normaler Angreifer tun würde. Aus diesem Grund kann ein Tiger Team auch einen Vulnerability Scanner gegen ein Ziel einsetzen. Unter normalen Bedingungen dürfte der Einsatz eines Scanners auch nicht auffallen, denn Angreifer, die der Kategorie Skript Kiddy zuzuordnen sind, nennen solche Tools auch Angriffssimulatoren, obwohl sie meist nur nach Evidenzen einer Schwachstelle in einem System suchen (vgl. Abschnitt 3.5.1). Solche Skript Kiddies wenden die frei im Internet zu beziehenden Scanner auf produktive Systeme an. So ist der Einsatz eines Vulnerability Scanners durchaus sinnvoll, da seine Auswirkungen im realen Betrieb erkannt werden können. Mit der Ausführung eines solchen Tools ist der Penetrationstest aber noch nicht beendet, da der nächste Schritt, die im folgenden Abschnitt behandelte Penetration der Zielsysteme, noch nicht stattgefunden hat (vgl. [Kurtz00a:2]).

[46] http://icat.nist.gov

5.5. Penetration

Der Angreifer ist nun über eine Anzahl von Schwachstellen informiert, die im Netzwerk, welches Gegenstand seiner Aktivitäten ist, vorhanden sind und versucht nun, die Schwachstellen auszunutzen. Dazu beschaffen sich Angreifer Exploits oder programmieren diese gegebenenfalls selber. So können in dieser Phase Buffer Overflows erzeugt und versucht werden, Passwörter zu erraten.

Die Penetration erfordert einen sehr viel größeren Aufwand als die bisherigen Schritte Reconnaissance und Vulnerability Detection. So sind zum Teil Änderungen an den Exploits notwendig, da diese Fehler enthalten oder auf Anpassungen an das Opfersystem angewiesen sind. Dazu benötigt ein erfolgreicher Penetrator auch Erfahrung in der Programmierung. Dabei sind im Wesentlichen Kenntnisse der Programmiersprache C notwendig, in der zurzeit die meisten Exploits programmiert sind.

Bei einer Penetration durch ein Tiger Team sollte nicht nur ein Denial-of-Service oder ein unautorisierter Zugriff das Ziel sein, sondern auch versucht werden, Zugriff auf die Log-Einträge eines Systems zu bekommen und diese zu manipulieren. So kann der Teil der Metastase, bei dem der Angreifer seine Spuren verwischt, durch den Penetrationstest betrachtet werden. Dabei sollte der Penetrator jedoch die Log-Dateien sichern, wenn sie ihm als Nachweis für die Erbringung vertraglich zugesicherter Leistungen dienen sollen.

Wesentliche Bedeutung hat in dieser Phase eine strukturierte Vorgehensweise. Nur so kann eine ausreichende Aussagekraft über die Sicherheit eines Systems oder Netzwerkes erlangt werden. Ein Ansatz ist die Zusammenfassung einzelner Aufgaben zu Modulen, die auf verschiedene Betrachtungsobjekte angewendet werden können. Diese Vorgehensweise wurde im Open Source Security Testing Methodology Manual (vgl. [OSSTMM02]) verwendet. In einer Studie zum Thema Penetrationstest vom Bundesamt für Sicherheit in der Informationstechnik (vgl. [BSI03]) wurde der Ansatz für die Verwendung in einem Penetrationstest interpretiert.

Die Penetration soll hier nicht weiter ausgeführt werden. Zum einen sind die Exploits sehr systemspezifisch, weshalb die genauere Betrachtung der Penetration in den Szenarien ab Kapitel 6 durchgeführt wird. Zum anderen wird hier eine ethische Grenze erreicht. Ein Leser mit maliziöser Absicht könnte diese Beschreibung dazu verwenden, sich Hacker-Fähigkeiten anzueignen und diese mit der Folge eines Schadens anzuwenden. Damit würde diese Arbeit auch das Niveau der meisten im Internet verfügbaren Tutorien mit Titeln wie „So wird man ein Hacker“ übersteigen. Sie enden meist an dieser Stelle mit Sätzen wie „Lerne Linux und C++“.

Wird ein zu großer Schaden erwartet, so wird die Penetration innerhalb eines Penetrationstests unter charakteristischer Nachbildung der Umgebung simuliert. Die Simulation hat aber gewisse Grenzen, die in Abschnitt 9.2 im Rahmen der Grenzen eines Penetrationstests näher diskutiert werden wird. Die Ergebnisse der bisher besprochenen Phasen werden in einem Abschlussbericht festgehalten, der Inhalt des folgenden Abschnitts ist.

5.6. Abschlussbericht

Die während des Penetrationstests gewonnenen Ergebnisse werden in einem abschließenden Bericht festgehalten. Der Bericht sollte nach Schultz (vgl. [Schultz96:5]) in erster Linie lösungsorientiert sein. So soll der Bericht für jedes gefundene Problem auch einen Lösungsvorschlag bieten. Ferner sollte der Bericht auch die Zielsetzungen, Rahmenbedingungen und die verwendete Methodik des Tests beinhalten. Die Methodik stellt dabei alle durchgeführten Tests vor, damit die Ergebnisse dem Kunden transparenter werden. Damit gewinnt der Kunde zum einen mehr Vertrauen in die Ergebnisse und kann zudem auch nachprüfen, ob das Tiger Team die geforderten Dienste erfüllt hat. Im Anhang sollten dann im Sinne des Kunden die verwendeten Tools aufgeführt werden. So hat der Kunde die Möglichkeit zu überprüfen, dass ihm nicht nur ein reiner Scan als Penetrationstest verkauft worden ist.

Ein Beispiel für einen solchen Bericht ist [Bim03] zu entnehmen. Zwar wird dort kein Penetrationstest in dem hier definierten Sinne durchgeführt, dennoch ist die Struktur eines Berichts sehr gut zu erkennen. Interessant ist auch die folgende Erklärung, die sich auf Seite 2 des Musterberichts findet: „Trotz der Nutzung modernster und umfassendster Methoden und Werkzeuge kann eine IT-Struktur nicht zu 100% auf Sicherheit getestet werden. Aus diesem Grunde stellt dieser Bericht keine Garantie für vollständige Sicherheit dar und sollte auch nicht als solche aufgefasst werden“. Eine solche Aussage ist Grundlage eines Haftungsausschlusses, damit der Kunde den Dienstleister nicht haftbar machen kann, falls nach abgeschlossenem Test ein Vorfall eintritt.

Bei der Struktur eines solchen Berichtes muss aber bedacht werden, dass er von Mitgliedern der Führungsebene als auch von Mitarbeitern der IT gelesen und umgesetzt wird. Ein Manager hat nicht die nötige Fachkenntnis, um einen detaillierten Report zu verstehen. Daher sollten die gefundenen Probleme zunächst in einem Teil des Berichtes erfasst werden, der in natürlicher Sprache ohne technische Details die Probleme beschreibt, so dass ein Management diese Informationen verstehen und verwerten kann. Dabei sollte nach Winkler (vgl [Winkler00:2f.]) nicht aufgeführt werden, welche Angriffe Erfolg hatten, sondern welche Assets gefährdet und welcher Schaden daraus resultieren kann. Assets und resultierender Schaden sind die Grundlagen jeder Entscheidung innerhalb des Risikomanagements.

In einem weiteren Abschnitt sollten die Probleme näher aus technischer Sicht erklärt werden. Dabei bietet sich die Möglichkeit, Ausgaben der Testläufe im Anhang zur Verfügung zu stellen und darauf zu verweisen. Dabei ist aber auch möglich, zur Motivation des Kunden neben den Problemen auch positive Bemerkungen zu tätigen.

Auf Grund der gefundenen Probleme werden dann Maßnahmenempfehlungen gegeben. Die Entscheidung, ob die Empfehlungen umgesetzt werden, liegt aber nicht beim Tiger Team. Das Tiger Team tritt hierbei als Revisor auf, der keine Entscheidungsgewalt hat. Damit liegt die Entscheidung, ob eine Maßnahme realisiert wird, allein beim Kunden. Um den Kunden bei der Entscheidung zu unterstützen, können hier zum einen die Konsequenzen dargestellt werden, falls er die Maßnahmen nicht durchführt. Zum anderen kann der Kunde unterstützt werden, indem im Bericht die Maßnahmen detaillierter beschrieben werden. So könnte der Bericht beispielsweise im Anhang eine Liste der fehlende Patche beinhalten.

Aussagen über Probleme können nur über den Zeitraum gemacht werden, in dem der Penetrationstest stattgefunden hat. Somit entsteht eine Momentaufnahme, so dass zu späteren Zeitpunkten keine Aussagen über eventuelle Probleme gemacht werden können. Daher ist ein Penetrationstest nur bei einer periodischen Wiederholung sinnvoll, wobei ein Bericht Aussagen über die Änderungen seit dem letzten Test machen kann. So kann dargestellt werden, welche Probleme seit dem letzten Test hinzugekommen sind und welche schon beim letzten Test bestanden haben. Weiter kann der Bericht darstellen, welche Maßnahmen seit dem letzten Test realisiert worden sind.

Werden Maßnahmenempfehlungen gegeben, so kann nach der Realisation der Maßnahmen ein Nachtest durchgeführt werden, um zu bestätigen, dass die gefundenen Probleme nicht mehr existieren. Dabei kann das Tiger Team sich auf bestimmte Probleme fokussieren.

5.7. Der Prozess des Penetrationstests

Bisher wurden mit der Reconnaissance, der Vulnerability Detection und der Penetration drei Phasen in einer sequentiellen Abfolge vorgestellt. Dabei wurde die Metastase des Angriffes jedoch nicht berücksichtigt.

Gelingt es einem Tiger Team, in ein System einzudringen, so kann es von dem Standpunkt aus weitere Penetrationen durchführen, um zum Beispiel die Installation einer Backdoor zu ermöglichen. Weiter kann das Tiger Team von diesem Standpunkt aus möglicherweise mehr Systeme entdecken. So kann es nach erfolgreicher Penetration nicht nur weitere Penetrationen, sondern von diesem Standpunkt aus eine erneute Reconnaissance starten. Auch wenn keine Penetration möglich war, kann das Team erneut mit der Reconnaissance beginnen, um diesmal genauer oder unter anderen Voraussetzungen mehr Systeme und Charakteristika zu erkennen.

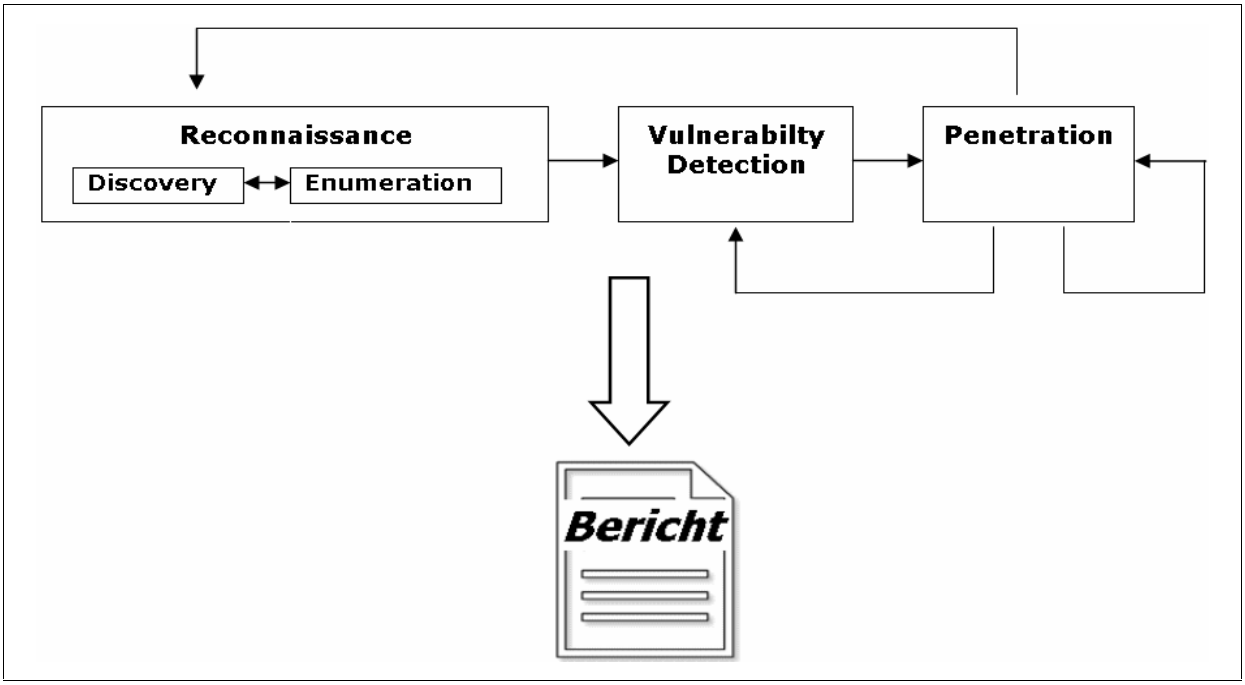


Abbildung 38: Prozess des Penetrationstests

Somit ist die Durchführung eines Penetrationstests ein unendlicher Prozess, der in Abbildung 38 in Form eines Automatenmodells dargestellt ist. Auch wenn keine Penetration geglückt ist, kann der Automat immer wieder in den Startzustand der Reconnaissance zurückkehren und den Prozess durchführen, wobei zwischendurch eine Penetration geglückt sein kann. So unterstützt dieses Modell auch einen zeitlich unbegrenzten Penetrationstest, der somit nicht dem Nachteil einer Momentaufnahme unterliegt. An beliebiger Stelle kann der Prozess unterbrochen und die Ergebnisse in einem Bericht festgehalten werden.

6. KAPITEL:

SZENARIO WEBSERVER

In den nun folgenden drei Kapiteln sollen die Möglichkeiten des Penetrationstests gezeigt werden. Jedes Kapitel beschreibt dabei ein Versuchszenario. Hierbei beschränkt sich die Betrachtung des Penetrationstests auf seine Anwendung im Aufgabengebiet der Revision, da die Revision das gebräuchlichste Anwendungsgebiet für Unternehmen ist, die ihre bestehenden Sicherheitskonzepte überprüfen wollen. Daher werden in diesem Kapitel nur die Auswirkungen der Ausnutzung von bekannten Schwachstellen betrachtet, um somit vermeidbare Vorfälle zu demonstrieren.

6.1. Beschreibung des Szenarios

In der New-Economy Zeit war die Erstellung einer Homepage vergleichbar mit einer Goldgrube, weshalb viele Kleinunternehmen diese Dienstleistung anboten und dies noch heute tun. In dieser Zeit entstanden auch Unternehmen, die auf ihrem Server Plattenplatz für die Webauftritte, so genannten Webspace, zur Verfügung stellen. Solche Unternehmen verfolgen dabei noch heute den Grundsatz, mit möglichst wenig Aufwand die geforderte Funktionalität zur Verfügung zu stellen. Gefordert war dabei nur, dass die Inhalte mittels eines Browsers abrufbar und per File Transfer Protocol aktualisierbar sind und zudem eine Email-Funktionalität bereitsteht. Dazu wurde meist eine Standardinstallation der Betriebssysteme und Software vorgenommen, die zur Erfüllung der Anforderungen genügte. Eine kontinuierliche Pflege der Installationen erfolgt dabei nicht. Solche Systeme sind noch heute im Einsatz und sollen diesem Szenario einen realistischen Hintergrund bieten.

Ausgangspunkt dieses Szenarios soll ein Webspace-Anbieter namens Arachnocolus[47] sein. Der zum Geschäftbetrieb notwendige Server dient dem Abrufen von Webpages per http, dem Ändern der Webseiten mittels ftp sowie dem Abruf und Empfang von Emails. Diese Funktionalitäten werden von den Microsoft Internet Information Services, kurz IIS, bereitgestellt. Zusätzlich soll der Server Unterstützung von PHP-4-Skripten und eine Datenbankfunktionalität mittels der Software MySQL bieten. Für ihren Webauftritt wird der Server unter anderem von der Firma Alarm Mayer[48] genutzt, die Sicherheitseinrichtungen für Gebäude vertreibt. Für ihren Web-Auftritt nutzt sie PHP und MySQL, um damit die Angebote und Preise zu verwalten. Auf Grund der auslaufenden Unterstützung von Windows NT 4.0, das bisher auf dem Server installiert war, entschied sich der Anbieter, seinen Web-Server auf Windows 2000 umzustellen. Da die IIS in Windows integriert sind, wurden unausweichlich auch die IIS von Version 4.0[49] auf Version 5.0 umgestellt

[47] Ähnlichkeiten mit real existierenden Firmen sind rein zufällig und nicht beabsichtigt.
[48] Der Name „Alarm Mayer“ ist frei erfunden. Ähnlichkeiten sind unbeabsichtigt und rein zufällig. Laut einer Abfrage der DENIC-Datenbank am 10.09.03 ist die Domain alarmmayer.de nicht vergeben.
[49] Die Version 4.0 der Internet Information Services ist im NT-Optionpack enthalten.

Bei der Installation von Windows 2000 wurde wie in der Vergangenheit auch eine Standardkonfiguration gewählt. Auf Grund von Unwissenheit werden zusätzliche Netzwerkdienste in die Konfiguration aufgenommen, um Fehler durch fehlende Komponenten des Betriebssystems zu vermeiden. Da der Inhaber von Arachnocolus durch seinen Freund, der einen Webserver in einer Großorganisation administriert, von den „hervorragenden Möglichkeiten“ und von „wahnsinnigen Vorteilen“ der Systemüberwachung mittels SNMP gehört hat und begeistert ist, ist jetzt die Überwachung des Webservers mit SNMP geplant, jedoch noch nicht realisiert. „Vorausschauend“ wurde aber schon SNMP installiert.

Zudem ist der Server direkt an das Internet angeschlossen. Eine Firewall ist dabei weder installiert, noch geplant. Arachnocolus ist auf die bestehenden Mängel seines Angebots bezüglich der IT-Sicherheit hingewiesen worden, sieht aber dennoch keinen Grund für den Mehraufwand zur Sicherung seines Servers. Begründet wird dies dadurch, dass in der Vergangenheit nichts passiert sei. Zudem vertritt der Anbieter die Ansicht, nicht attraktiv genug für einen Angreifer zu sein. Sollte dennoch ein Angreifer Erfolg haben, so können nach Ansicht des Anbieters der Schaden kein Ausmaß annehmen, das nicht durch die vorhandenen Backups begrenzt werden könne. Um diese Ansichten zu widerlegen, wird Arachnocolus nahe gelegt, sich eines Penetrationstests zu unterziehen, damit er das mögliche Ausmaß anhand der Demonstration eines Angriffes besser beurteilen kann. Dieser Aufforderung kommt Arachnocolus nach und stellt einen gespiegelten Server bereit, der in diesem Szenario nachgebildet wird.

Das Szenario beschränkt sich ausschließlich auf die Betrachtung des Servers. Auf weitere Hosts, wie beispielsweise die Workstations der Verwaltung oder von Entwicklern, die vor Ort Homepages entwerfen, kann auf Grund der fehlenden Firewall direkt zugegriffen werden. Wegen der Beschränkung an Zeit und Laborkapazitäten werden sie in diesem Szenario vernachlässigt. Zur Reduktion der Komplexität des in diesem Szenario durchgeführten Penetrationstests wird auf Risiken der in den IIS 5.0 enthaltenen FTP- und SMTP-Server nicht eingegangen.

Des Weiteren werden nur Risiken betrachtet, die von einem externen Angreifer ausgehen. Dieser externe Angreifer hat außer der IP und dem Namen des Servers keine weiteren Informationen über den Server. Risiken, die von Mitarbeitern der Firma Arachnocolus ausgehen, sowie Risiken, die vom physikalischem Zugriff ausgehen, werden nicht betrachtet.

Für den Penetrationstest wird ein außenstehender Angreifer angenommen, der die Angriffe über das Internet ausübt. Daher hat der Penetrationstest externen Charakter. Ziel der durchgeführten Angriffe ist ein Einbruch in das System. Bei den Angriffen sollen sowohl Integritätsverletzungen durch Manipulation von geschäftswichtigen Daten wie die Homepages der Kunden sowie deren Datenbanken sein. Das Bestehen eines entsprechenden Service Level Agreements wird angenommen.

Nach der Durchführung werden die Versuche von Jan Menne mittels des in seiner Diplomarbeit (vgl. [Menne03]) entwickelten Tools „CompareSys“ analysiert. Dazu wurden vor und nach den Angriffen Images der Festplatten mit der Software Norton Ghost erstellt, und mit CompareSys verglichen. Die Ergebnisse des Vergleichs werden in [Menne03:62ff.] behandelt.

6.2. Versuchsaufbau

Die Penetrationstests in diesem und dem nächsten Kapitel verwenden Angriffe, die Schäden an Systemen zur Folge haben können. Daher wird für die Durchführung der Szenarien Webserver und Firewall ein Testnetz in einem abgeschotteten Labor aufgebaut, damit durch die Penetrationstests keine unbeteiligten Systeme beeinträchtigt werden können. Das Testnetz ist nicht an das Internet angeschlossen, benutzt aber mittels eines TCP/IP-Netzwerkes die gleichen Techniken.

Da die bei den Versuchen eingesetzten Tools zum Teil nur für Linux, zum Teil nur für Microsoft Windows verfügbar sind, wurden zwei Angreifer Systeme installiert. Auf dem Angreifer Devil wurde Gentoo Linux 1.4rc2, auf dem Angreifer Belzebub Microsoft Windows 2000 Professional in deutscher Sprache installiert. Ist ein Beobachter erforderlich, so kann der jeweils untätige Angreifer den Netzverkehr aufzeichnen. Dazu steht für beide Betriebssysteme das frei erhältliche Tool Ethereal zur Verfügung. Da die eingesetzten Beobachter-Tools für beide Plattformen verfügbar sind, übernimmt der jeweils unaktive Angreifer-Rechner die Aufgabe des Beobachters. Damit der Beobachter den gesamten Netzverkehr aufzeichnen kann, wurde für die Verbindung der Rechner ein Ethernet-Hub gewählt.

Grundlage der Entscheidung für die Linux Distribution Gentoo ist die Begebenheit, dass die meisten Tools für einen Penetrationstest, die für Linux verfügbar sind, in der jeweils neuesten Version nur als Quellcode vorliegen. Zwar können Quellcodes auch auf anderen Distributionen kompiliert werden, jedoch kommt es dort wegen speziell an die jeweilige Distribution angepassten Bibliotheken zu Problemen. Um diese Probleme zu umgehen, wurde für diese Arbeit die Distribution Gentoo gewählt, da diese Distribution vollständig aus frei verfügbaren Quellen besteht, die erst bei der Installation auf der jeweiligen Maschine kompiliert werden.

Ein Problem der Distribution Gentoo stellt allerdings der Bezug der Quellen dar. Die Quellen, die auch im gepackten Zustand mehrere Megabyte groß sind, werden von Gentoo bei der Installation aus dem Internet heruntergeladen. Zwar besteht die Möglichkeit, die Quellen von CD-Rom einzuspielen, jedoch wäre dieser Weg wegen der Größe und der unübersichtlichen Abhängigkeiten der einzelnen Quellen zu umständlich. Daher wurde für die Versuche ein Rechner installiert, der die im Internet verfügbaren Dateien bereitstellt und somit das Internet spiegelt. Dieser Rechner hat den Namen mirror.irt.local. Auf ihm wurde Suse Linux 7.3. Professional installiert. Der Zugriff auf die Dateien via http ist durch den mitgelieferten Webserver Apache 1.3.20 möglich.

Als Opfersystem wurde der Rechner www3 mit Windows 2000 Professional Deutsch installiert. Das System verfügt über den Webserver der Microsoft IIS 5.0. Zudem wurde MySQL 3.23.56 und php 4.3.2rc3 installiert. Entsprechend der Vorraussetzung des Szenarios wurden zudem die Windowskomponenten „Netzwerkdienste/Einfache TCP/IP-Dienste“ sowie „Verwaltungs- und Überwachungsprogramme/SNMP“ installiert. Arachnolocus erhofft sich durch die Installation der Komponenten, dass im Betrieb keine Störungen durch fehlende Abhängigkeiten bei der Konfiguration der Dienste oder Installation weiterer Software auftreten. Zudem wurden das Packprogramm 7-zip 2.30 Beta 28 sowie der Browser Mozilla 1.3.1 installiert. Weitere Konfigurationen wurden nicht vorgenommen, so dass es sich bei

dieser Installation um eine Standardkonfiguration handelt. Ferner fehlen sämtliche Service Packs sowie Patches. Auch die Frontpage-Server-Erweiterungen wurden auf diesem Server installiert, obwohl sie nicht benötigt werden. Auf den Einsatz einer AntiViren-Software wurde verzichtet.

Als Webserver-Daten wurde die Homepage von Alarm Mayer installiert. Sie wurde auf einem anderen Rechner mittels UltraEdit erstellt. Sie benutzt PHP und MySQL, um die Produkte und Preise in der Datenbank zu speichern und auf der Webseite darzustellen. Installiert wurde die Webseite, indem die HTML- und PHP Dateien in das Verzeichnis c:\inetpub\wwwroot kopiert wurden. Weiter wurde das Datenverzeichnis der Mysql-Datenbank von dem Erstellungsrechner auf den Webserver kopiert, um die Datenbank auf dem Webserver verfügbar zu machen. Der Quellcode der Webseite ist dem Anhang G3 zu entnehmen.

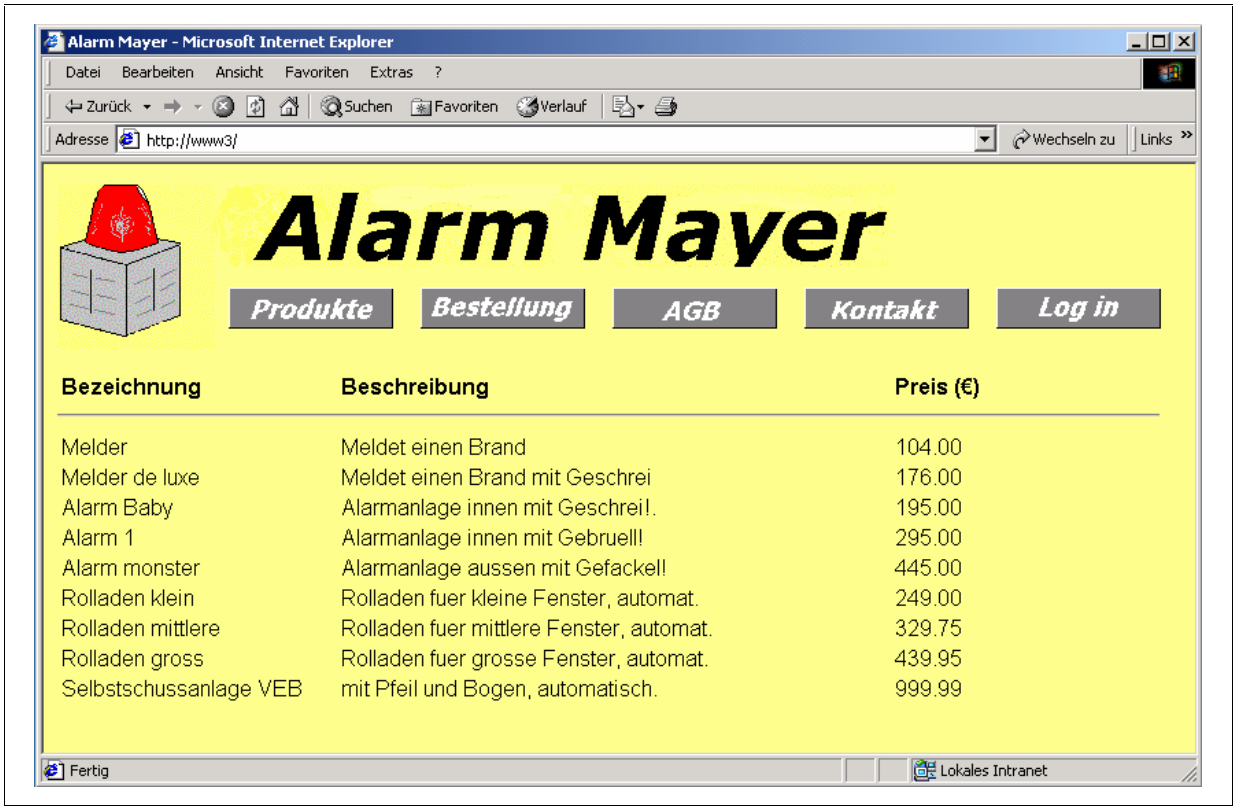


Abbildung 39: Die Homepage der Firma Alarm Mayer

Der Name des Opfersystems, www3, entstand durch die im Rahmen dieser Szenarien durchgeführten Versuche mit leicht geänderten Konfigurationen des Opfersystems, die aber wegen Seiteneffekten der Konfigurationen zu erheblichen Problemen führten. Ein Seiteneffekt war beispielsweise die Verwendung des Dateisystems NTFS für das Opfersystem, was durch die Rechteverwaltung zu Problemen bei der Analyse der Versuche mittels CompareSys (vgl. [Menne03]) führte. Somit unterliegt die Versuchsdurchführung der Einschränkung, dass das Dateisystem FAT32 benutzt wird, während in einer realen Umgebung NTFS verwendet werden würde.

Eine weitere Einschränkung stellt die Lizenz von Microsoft Windows 2000 dar. Für die Versuche im Labor stehen lediglich drei Windows 2000 Professional Lizenzen zur Verfügung. Eine Lizenz wird für den Angreifer Belzebub verwendet, so dass parallel zwei

Opfersysteme mit Windows 2000 betrieben werden können. Die Windows 2000 Professional Lizenz unterliegt gegenüber den Server Versionen zwei nennenswerten Einschränkungen. Zum einen sind die Internet Information Services in der Professional Version nicht in der Standardkonfiguration installiert und müssen daher manuell nachinstalliert werden. Zum anderen kann mit dem IIS der Professional Version nur eine Webseite betrieben werden, was allerdings keine Einschränkung für das Szenario bedeutet.

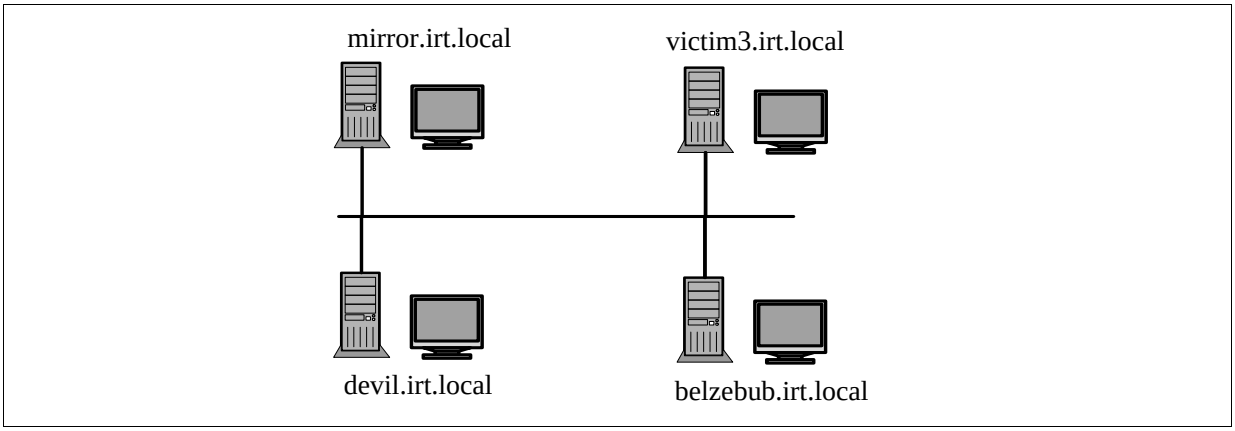


Abbildung 40: Topologie des Labornetzes

Rechnername[50]	Ergo	Kathy	Miraculix	IRT-Mirror
Systemname[51]	Belzebub	Devil	Opfer	Mirror.irt.local
Leistung	Pentium II-350; 128MB RAM	Pentium II-350; 128MB RAM	Pentium 4-1,8; 512MB RAM	Pentium233MMX; 64MB RAM
Plattennr / -größe	22 / 4,5GB SCSI	45 / 2,1GB SCSI	50 / 80GB IDE	- / 20GB IDE
Funktion	Angreifer	Angreifer	Opfer	Dateiserver
IP-Adresse	192.168.0.18	192.168.0.17	192.168.0.3	192.168.0.249

Tabelle 5: Übersicht der Rechnerkonfigurationen

6.3. Überlegungen zur Sicherheit eines Webservers

Bevor nun ein Penetrationstest auf den Webserver durchgeführt wird, sollen zunächst generelle Überlegungen zur Sicherheit eines Webservers behandelt werden. Die Konfiguration des umgangssprachlich als Webserver bezeichneten dedizierten Rechners, der zur Speicherung und Auslieferung von Webseiten dient, kann in Anlehnung an [Nevers02:4] in die drei Ebenen Anwendung, Server und Betriebssystem unterteilt werden. Abbildung 41 stellt diese Unterteilung dar.

[50] Bezeichnet die Namen der physikalischen Rechner mit Wechselrahmeneinschub
[51] Bezeichnet den logischen Namen einer Installation auf einer Wechselplatte

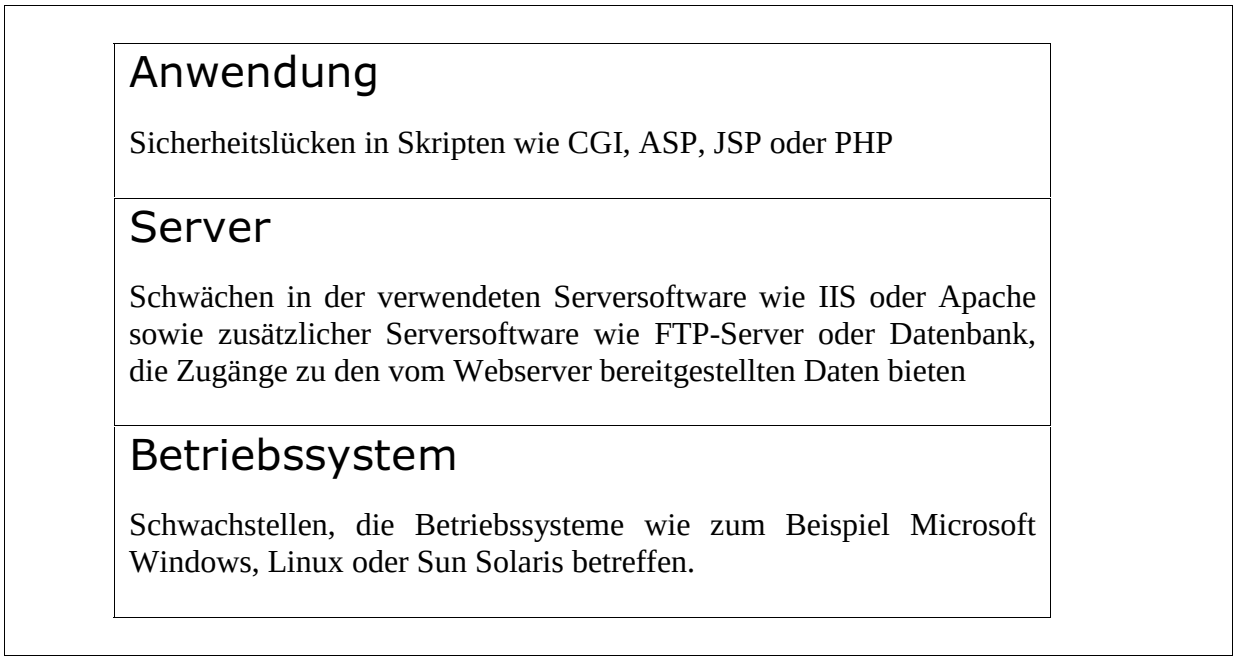


Abbildung 41: Ebenen der Sicherheit eines Webservers (in Anlehnung an [Nevers02:4])

Ziel der Versuche in diesem Szenario ist die Betrachtung der Sicherheit der Serversoftware und des Betriebssystems. Die Sicherheit der Web-Anwendung wird hier auf Grund des Aufwands vernachlässigt.

Die Probleme eines Webservers entstehen durch implementations- und konfigurationsbedingte Schwachstellen in der Software. Neben Fehlern in der Software selbst, die von einem Administrator durch die Installation von Patches behoben werden können, entstehen vor allem durch die Benutzung der Voreinstellungen der Installation, auch als Standardeinstellungen oder Standardkonfiguration bezeichnet, Sicherheitsprobleme, die im folgenden aufgezeigt werden:

- zu viele Dienste
- zu hohe Rechte des Webserver Prozesses
- Abhängigkeit des Webservers von anderen Komponenten und somit hohe Integration in das Betriebssystem
- Gefahren durch Fehler in Plugins
- Komplexität der Bedienung führt zu Fehlbedienungen
- Beispielanwendungen und Skripte

Die Standardkonfiguration eines Windows 2000 Servers enthält Dienste, die der Betreiber eines Systems nicht benötigt. So sind in der Installation der IIS auch ein SMTP-Server sowie Erweiterungen für die Unterstützung von Frontpage oder WebDAV enthalten. Solche Erweiterungen enthalten Schwachstellen, wodurch die Installation des Webservers wegen der unbenötigten Komponenten zu einem vermeidbar größeren Risiko führen. Daher sollten unbenötigte Komponenten deinstalliert oder deaktiviert werden.

Die Internet Information Services sind in die folgenden Prozesse aufgeteilt:

- IIS Admin-Dienst / IISADMIN
- WWW-Publishingdienst / W3SVC
- FTP-Publishingdienst / FTPSVC
- Simple Mail-Transportprotocol SMTP / SMTPSVC

Sie alle werden mit dem vordefinierten Dienste-Benutzerkonto (engl.: service account) LocalSystem ausgeführt, das auch nur SYSTEM genannt wird. Dieses Konto ist mit den höchsten Rechten versehen (vgl. [MSDN03]). Gelingt es einem Angreifer, eine Schwachstelle in einem der Dienste auszunutzen, um damit in einen Rechner einzudringen, so verfügt er mit hoher Wahrscheinlichkeit über diese Rechte. Dienste sollten keine zu hohen Rechte erhalten und somit, soweit möglich, mit den Rechten des Kontos LocalService betrieben werden (vgl. [MSDN03]).

Zudem bestehen Abhängigkeiten zwischen den IIS und anderen Softwarekomponenten von Microsoft Windows. So führen Fehler im Indexingdienst zu Schwachstellen, die auch die IIS betreffen (vgl. [MS01-033] und [MS01-044]). Diese Schwachstelle nutzte beispielsweise der Wurm CodeRed aus. Auch Fehler in der ntdll.dll, welche tief im Betriebssystem verankert ist, können über die WebDAV-Schnittstelle der IIS ausgenutzt werden (vgl. [MS03-007]).

Auch Fehler in Plugins, mit denen ein Webserver um Funktionalitäten wie beispielsweise das Ausführen von php-Skripten erweitert werden kann, bergen Risiken. Der IIS-Webserver kann durch so genannte ISAPI (Internet Server Application Programmable Interfaces) erweitert werden. In einer Standardinstallation der Microsoft IIS 5.0 sind eine große Anzahl mitgelieferter ISAPIs eingebunden, die für den gewöhnlichen Betrieb nicht benötigt werden. Da einige ISAPIs Schwachstellen enthalten, entsteht durch die per default installierten ISAPIs ein unnötiges Risiko. Ein Beispiel hierfür ist die Schwachstelle CVE-2001-0241 mit der Bezeichnung “IIS 5.0 –printer ISAPI Extention Buffer Overflow Vulnerability“, die im Microsoft Security Bulletin MS01-023 behandelt wird und bei BugTraq unter der Kennung BID 2674 verzeichnet ist.

Die Konfiguration der IIS erfolgt über die Microsoft Management Console, kurz MMC. Wie auch bei anderen Komponenten ist bei der Konfiguration des Webservers eine Vielzahl von Einstellungen möglich, wobei die Bedeutung aller Optionen in den meisten Fällen nicht bekannt ist. Durch die dabei entstehende Komplexität der Konfiguration kommt es zu Konfigurationsfehlern, die von einem Angreifer ausgenutzt werden können.

Anwendungen stellen eine weitere Gefahr dar, da sie direkt auf dem Server ausgeführt werden. Eine Anwendung kann von einem Benutzer durch Eingabefelder oder Parameter in der URL gesteuert werden. Dabei besteht die Gefahr, dass die Anwendung einen ungeprüften Puffer enthält, wodurch ein Angreifer die Möglichkeit hat, mittels eines Pufferüberlaufs beliebigen Code auf dem Server auszuführen. Auch Cross-site-Skripting Attacken, bei denen Anwender durch Skripte auf andere Seiten geleitet werden, sind durch solche Schwachstellen möglich. Web-Anwendungen können zudem durch Skripte realisiert sein, die im Quellcode auf dem Server gespeichert sind. Sie enthalten häufig Authentifizierungsmerkmale wie Benutzername und Passwort für eine Datenbank im Klartext. Gelingt es einem Angreifer, den Quellcode zu erlangen, so kann er ungehindert auf eine Datenbank zugreifen. In einer

Standardinstallation verfügt der IIS-Webserver über Beispielanwendungen, deren Sicherheitslücken Angriffe möglich machten.

Im Folgenden wird ein Penetrationstest nach der in Kapitel 5 dargestellten Vorgehensweise auf den oben beschriebenen Webserver angewendet.

6.4. Reconnaissance

Die Reconnaissance wurde in Abschnitt 5.3.1 in Discovery und Enumeration unterteilt. Eine Discovery, deren Ziel die Entdeckung eines Systems ist, ist bei diesem Versuchsaufbau nicht notwendig, da das Zielsystem bekannt ist. Außerdem ist eine Discovery in dem Testnetz nicht möglich, da keine Einträge über die Systeme in Whois- oder DNS-Datenbanken existieren. Es existieren zudem keine Netzwerksegmente, womit Methoden wie Traceroute nicht sinnvoll sind.

Aus den vorhergehenden Überlegungen wird in diesem Szenario nur eine Enumeration durchgeführt, mit der von den Angriffsrechnern aus die einzelnen Eigenschaften des Opfersystems ermittelt werden. Dabei wurden folgende Methoden verwendet:

- OS-Fingerprint
- Port Scan
- Banner Grabbing
- User Informationen

Das OS-Fingerprint wurde mit dem Tool nmap durchgeführt. Für die Durchführung benötigt der Scanner die Angabe eines offenen sowie eines geschlossenen Ports. Dazu wurden die Ports 134/tcp und 135/tcp gewählt, da Port 134/tcp meist ungenutzt ist, Port 135/tcp hingegen bei Windows als auch UNIX-Systemen zur Realisation der Implementation von Remote Procedure Call, kurz RPC, nach dem DCE-Standard[52] genutzt wird.

```
C:\>nmap -sS -O -P0 -p134-135 www3

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on www.irt.local (192.168.0.3):
(The 1 port scanned but not shown below is in state: closed)
Port       State       Service
135/tcp    open        loc-srv
Remote operating system guess: Windows Millennium Edition (Me), Win 2000,
or Win XP

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Abbildung 42: Ausgabe von nmap nach OS Fingerprint

[52] DCE ist eine Abkürzung für Distributed Computing Enviroment. Microsofts RPC-Protokoll ist eine Implementation des im DCE-Standards. Alternativ steht auf UNIX-Systemen die dort am weitesten RPC-Variante des Open Network Computing Standards zur Verfügung, die auf einer Entwicklung von Sun basiert (vgl. [Chapman00:349]). Für weiter Informationen siehe http://www.opengroup.org/dce/ und [Pfleeger00:415f.]

Das in Abbildung 42 dargestellte Ergebnis des Fingerprinting mittels nmap ist sehr ungenau. Genauer kann das Betriebssystem mit einer SNMP-Abfragen erfolgen, die bereits in Abschnitt 5.3.2 auf Seite 92 genauer beschrieben worden sind.

In einem darauf folgenden Portscan wurden zunächst die Ports identifiziert. Dazu wurde wieder das Tool nmap benutzt. In vorhergehenden Experimenten mit dem Tool nmap stellte sich heraus, dass sämtliche TCP-Varianten des Scanners die gleichen Ergebnisse erzeugten. Somit wurden nur die Optionen -sS für einen TCP-SYN-Scan und -sU für einen UDP-Scan verwendet. Die Ergebnisse sind in Abbildung 43 dargestellt.

 # Stealth TCP Portscan:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on www.irt.local (192.168.0.3):
(The 65320 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
17/tcp     open        qotd
19/tcp     open        chargen
21/tcp     open        ftp
25/tcp     open        smtp
80/tcp     open        http
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
445/tcp    open        microsoft-ds
1025/tcp   open        NFS-or-IIS
1026/tcp   open        LSA-or-nterm
3306/tcp   open        mysql
```

UDP Portscan:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on www.irt.local (192.168.0.3):
(The 65320 ports scanned but not shown below are in state: closed)
Port       State       Service
7/udp      open        echo
9/udp      open        discard
13/dup     open        daytime
17/udp     open        qotd
19/udp     open        chargen
135/udp    open        loc-srv
137/udp    open        netbios-ns
138/udp    open        netbios-dgm
161/udp    open        snmp
445/udp    open        microsoft-ds
500/udp    open        isakmp
520/udp    open        route
1027/udp   open        unknown
1028/udp   open        ms-lsa
3456/udp   open        IISrpc-or-vat
```

Abbildung 43: PortScan auf www3 mittels nmap

Auf dem Opfersystem sind insgesamt 30 Ports offen. Anhand der Portnummern kann auf die Dienste geschlossen werden. So gehören die Ports 137/udp, 138/udp und 139/udp, sowie 445/tcp und 445/udp der SMB-Protokollfamilie (vgl. Abschnitt 4.6.10) an.

Im Folgenden wird ein Banner Grabbing auf das System www3 angewendet. Ein manuelles Banner Grabbing, das im einfachsten Fall mittels des Befehls `telnet` erfolgen kann, ist bei einigen Diensten sehr aufwendig, da zunächst ein Befehl an den Dienst gesendet werden muss, damit der Dienst seinen Banner enthüllt. So muss bei dem Web-Dienst ein GET-Request erfolgen, dessen Antwort das Banner des Webservers enthält. Daher beschränkt sich der dem Szenario unterstellte Angreifer, der primär das Ziel verfolgt, in den Rechner einzudringen, um Daten zu manipulieren, auf einige wenige Ports. Dazu bieten sich auf den ersten Blick die Dienste FTP auf Port 21/tcp, Web auf Port 80/tcp und mysql auf Port 3306/tcp an. Auf ihnen wird mittels telnet ein Banner Grabbing durchgeführt, deren Ergebnisse in Tabelle 6 dargestellt sind. Das Banner des FTP-Dienstes enthüllt zudem den lokalen Rechnernamen.

Port	Banner
21/tcp	220 www3 Microsoft FTP Service (Version 5.0).
80/tcp	Server: Microsoft IIS/5.0
3306/tcp	3.23.56-max-debug•z!2x&RIV

Tabelle 6: Ergebnisse des manuellen Banner Grabbings

Wie in Kapitel 5.3.2 bereits erwähnt, kann das Banner Grabbing auch durch Tools wie amap und SuperScan automatisiert werden. Das Tool amap akzeptiert eine von nmap auf Grund des Schalters -oM erstellte Datei als Eingabe. Der Versuch, amap unter Verwendung der beim Portscan erstellten Datei amaptcp.log auf das System www3 anzuwenden, scheiterte in einer Endlosschleife. Die Ergebnisse des Tools SuperScan, dessen Porterkennung nicht die Vollständigkeit von nmap erreicht, sind in Abbildung 44 dargestellt.

Da der Port 161/udp offen ist, wird angenommen, dass der fiktive Angreifer versucht, Informationen über das verwendete Betriebssystem, Benutzernamen und Freigaben mittels der Verwendung von SNMP-Abfragen zu erschließen. Für sämtliche snmp Abfragen wurde die Software snmputil benutzt, die gegenüber dem Tool snmptool kompaktere Ausgaben erzeugt und nach [Kurtz01:122] Bestandteil des Microsoft Resource Kits sein soll. Zwar ist das Tool nicht unter den frei verfügbaren Resource Kit Tools, jedoch ist dessen Quellcode im Microsoft Plattform SDK enthalten, so dass die Binary mittels eines Compilers erzeugt werden konnte. Neben dieser kommandozeilenbasierten Methode bieten sich auch graphische Softwarelösungen an, die in Tools wie NSS (vgl. Abschnitt 6.6.1) oder WS_Ping ProPack enthalten sind.

```
* + 192.168.0.3
      |___     7  Echo
      |___     9  Discard
      |___    13  Daytime
            |___ 13:47:16 21.07.2003.
      |___    17  Quote of the Day
            |___  Gesichter  sind  die  Leseb.cher  des  Lebens.....(Federico
Fellini)..
      |___    19  Character Generator
            |___                                                !"#$%&'()*+,-
./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_‘abcdefg..!"#$%&'()*+,-
./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTU
      |___    21  File Transfer Protocol [Control]
            |___ 220 www Microsoft FTP Service (Version 5.0)...
      |___    25  Simple Mail Transfer
            |___ 220 www Microsoft ESMTP MAIL Service, Version: 5.0.2172.1
ready at  Mon, 21 Jul 2003 13:47:16 +0200 ..
      |___    80  World Wide Web HTTP
            |___   HTTP/1.1   200   OK..Server:   Microsoft-IIS/5.0..Content-
Location: http://192.168.0.1/index.html..Date: Mon, 21 Jul 2003 11:47:18 GM
      |___   135  DCE endpoint resolution
      |___   139  NETBIOS Session Service
      |___   443  https  MCom
      |___   445  Microsoft-DS
      |___  1025  network blackjack
      |___  1026  LSA-or-nterm (nmap)
      |___  3306  mysql (nmap)
            |___ +....3.23.56-nt.....ob|l&vD-.,.................
```

Abbildung 44: Ergebnisse der Anwendung von SuperScan auf www3

Mittels der in Kapitel 5.3.2 auf Seite 86 beschriebenen Methodik erfragt der Angreifer die Version des Betriebssystems, wobei er folgende Ausgabe erhält:

```
Software: Windows 2000 Version 5.0 (Build 2195 Uniprocessor Free)
```

Zwar erhält der Angreifer gegenüber den von nmap gelieferten Informationen nun genauere Daten über die Bezeichnung und Version des Betriebssystems, jedoch kann er auch so keine Informationen über eingespielte Servicepacks oder Patches erlangen und auch nicht erschließen, welche Schwachstellen in der Software bereits geschlossen worden sind.

Wesentlich wichtiger sind Informationen über vorhandene Benutzernamen, die für einen Einbruch genutzt werden können. Kennt der Angreifer die Benutzernamen, braucht er nur noch die Passwörter zu erlangen. Somit kommt ein Angreifer dem Ziel des Einbruchs ein entscheidendes Stück näher. Die Erlangung von Benutzernamen mittels snmp auf dem System www3 ist in Abbildung 45 dargestellt.

```
C:\>snmputil walk 192.168.0.3 public .1.3.6.1.4.1.77.1.2.25.1.1.
Variable   =   .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.
svUserTable.svUserEntry.svUserName.4.71.97.115.116
Value    = String Gast

Variable   =   .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.
svUserTable.svUserEntry.svUserName.8.73.85.83.82.95.87.87.87
Value    = String IUSR_WWW

Variable   =   .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.
svUserTable.svUserEntry.svUserName.8.73.87.65.77.95.87.87.87
Value    = String IWAM_WWW

Variable   =   .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-
2.server.
svUserTable.svUserEntry.svUserName.13.65.100.109.105.110.105.115.116.114.9
7.116.
111.114
Value    = String Administrator
```

Abbildung 45: Erkennung von Benutzernamen mittels snmp

Auch Freigaben lassen sich über snmp auslesen. Da auf dem System www3 keine Freigaben eingerichtet worden sind, liefert die in Abbildung 46 dargestellte Abfrage kein Ergebnis. Administrative Freigaben können auf diesem Wege allerdings nicht erkannt werden.

```
C:\>snmputil walk 192.168.0.3 public .1.3.6.1.4.1.77.1.2.27.
End of MIB subtree.
```

Abbildung 46: Erkennung von Freigaben mittels snmp

Neben dem Banner Grabbing besteht mit Hilfe bestimmter SNMP-Anfragen eine weitere Möglichkeit, die hinter den Ports betriebenen Dienste zu erkennen. Für das System www3 ergibt die Abfrage die in Abbildung 47 gekürzt dargestellte Ausgabe. In dieser Ausgabe ist zu erkennen, dass die IIS mit allen Komponenten sowie MySQL auf dem Server betrieben werden. Außerdem lässt das „TCP/IP-NetBIOS-Hilfsprogramm” auf die Verwendung von SMB schließen. Die Möglichkeit, Informationen über SNMP zu beschaffen, ist allerdings sehr begrenzt, da nur wenige im Internet verfügbare Systeme diesen Dienst anbieten.

```
C:\>snmputil walk 192.168.0.3 public .1.3.6.1.4.1.77.1.2.3.1.

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String Computerbrowser

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String IIS Admin-Dienst

Variable =.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String Ereignisprotokoll

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String FTP-Publishingdienst

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String WWW-Publishingdienst

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String Arbeitsstationsdienst

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2….
Value    = String TCP/IP-NetBIOS-Hilfsprogramm

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String Windows-Verwaltungsinstrumentation

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String Simple Mail-Transportprotokoll (SMTP)

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2…
Value    = String MySql
```

Abbildung 47: Auslesen der auf www3 laufenden Dienste mittels snmp (gekürzt)

Auf Grund des Portscans wurde geschlossen, dass die SMB-Protokollfamile aktiviert ist. Diese Information nutzt der Angreifer dazu, Informationen über Freigaben und Benutzerkonten mittels der Protokolle Netbios, SMB und CIFS herauszufinden. Dazu wurde das Tool enum verwendet, dessen Ausgabe in Abbildung 48 dargestellt ist.

```
C:\>enum -U -S -P -G -L www3
server: www3
setting up session... success.
password policy:
  min length: none
  min age: none
  max age: 42 days
  lockout threshold: none
  lockout duration: 30 mins
  lockout reset: 30 mins
opening lsa policy... success.
server role: 3 [primary (unknown)]
names:
  netbios: WWW
  domain: ARBEITSGRUPPE
quota:
  paged pool limit: 33554432
  non paged pool limit: 1048576
  min work set size: 65536
  max work set size: 251658240
  pagefile limit: 0
  time limit: 0
trusted domains:
  indeterminate
netlogon done by a PDC server
getting user list (pass 1, index 0)... success, got 4.
  Administrator  Gast  IUSR_WWW  IWAM_WWW
enumerating shares (pass 1)... got 5 shares, 0 left:
  E$  IPC$  F$  ADMIN$  C$
Group: Administratoren
WWW\Administrator
Group: Benutzer
NT-AUTORITÄT\INTERAKTIV
NT-AUTORITÄT\Authentifizierte Benutzer
Group: Gäste
WWW\Gast
WWW\IUSR_WWW
WWW\IWAM_WWW
Group: Hauptbenutzer
Group: Replikations-Operator
Group: Sicherungs-Operatoren
cleaning up... success.
```

Abbildung 48: Enumeration von www3 durch NetBIOS und SMB

Mit den aus der Anwendung der Enumeration gewonnenen Ergebnissen hat ein potentieller Angreifer bereits Ergebnisse gesammelt. So weiß er, dass auf dem Opfersystem das Betriebssystem Microsoft Windows 2000 Professional sowie die Serversoftware Internet Information Server 5.0 und MySQL 3.53 installiert sind. Diese Informationen sind Grundlage für die im nächsten Schritt folgende Vulnerability Detection, in der sich der Angreifer über Schwachstellen in der erkannten Software informiert.

Zudem kennt der Angreifer nun mögliche Zugangspunkte zu einem System. So wurde durch den Portscan eine Vielzahl offener Ports entdeckt. Außerdem verfügt der Angreifer nun über Benutzernamen und die Kenntnis, dass das System über administrative Freigaben verfügt.

Freigaben und Benutzernamen können dazu verwendet werden, mittels eines Brute-Force Angriffes Passwörter zu erraten.

Die in der Reconnaissance gewonnenen Ergebnisse fließen in die nun folgende Vulnerability Detection ein, mit der ein Angreifer oder ein Tiger Team Schwachstellen in einem System erkennen kann.

6.5. Vulnerability Detection

Nachdem der Angreifer sich in der Reconnaissancephase Informationen über das System beschafft hat, leitet er nun aus den Ergebnissen Schwachstellen des Servers ab, die innerhalb der Penetration ausgenutzt werden können. Bereits die Durchführung der Reconnaissance offenbart Schwachstellen, die auf Fehler in der Konfiguration des Servers beruhen, wodurch die Grenze zur Vulnerability Detection verschwimmt. Die Informationen, die im vorherigen Abschnitt von dem fiktiven Angreifer erlangt worden sind, hätten nicht derart umfangreich sein dürfen.

So fällt bereits bei einem Portscan auf, dass die Ports der Dienste Echo und Chargen offen sind. Dies deutet auf die Verwundbarkeit des Servers gegen eine Echo/Chargen Attacke hin. Der Konfigurationsfehler, der zu dieser Schwachstelle führte, ist die Installation der Windowskomponente „Netzwerkdienste/Einfache TCP/IP-Dienste“. Eine Konfigurationsrichtlinie, die eine Spezifikation zur Installation und Konfiguration eines bestimmten Systems darstellt, sollte daher beinhalten, dass die einfachen TCP/IP Netzwerkdienste nicht installiert werden dürfen.

Ebenso sollte der SNMP-Dienst nicht installiert sein, da seine Verwendung zwar geplant, aber nicht unbedingt benötigt wird. SNMP kann von Nutzen sein, wenn ein Netzwerk zentral überwacht wird. In diesem Falle besteht aber durch Zugang zu der Community „public“ ein Konfigurationsfehler. Diese Community sollte umbenannt werden[53]. Zudem sollte der Zugangspunkt zum SNMP-Dienst, der Port 161/udp, gefiltert werden, so dass der SNMP-Dienst nicht oder nur eingeschränkt über das Internet zugänglich ist.

Ein weiterer Konfigurationsfehler ist die Möglichkeit von Null-Sessions bei der Benutzung des SMB-Protokolls. Sie sind auf Grund der Voreinstellungen möglich, sollten aber deaktiviert werden, damit ein Angreifer keine weiteren Informationen wie Benutzernamen und Freigaben erhalten kann. So konnte in der Phase der Reconnaissance festgestellt werden, dass die administrativen Freigaben[54], die bei einer Standard-Konfiguration vorhanden sind, nicht deaktiviert worden sind. Sie können vom Angreifer für weitere Aktionen wie das Erraten von Passwörtern oder die Manipulation von Dateien auf dem Opfersystem verwendet werden. Zudem sollten RPC-Endpunkte gefiltert werden. Sie werden meist nur lokal auf einem Rechner aber nicht im Netzwerk benötigt.

Anhand der in der Reconnaissance gewonnenen Informationen, dass ein IIS 5.0 Server auf einem Windows 2000 System betrieben wird, kann der Angreifer sich über Schwachstellen

[53] Denkbar wäre auch eine Authentikation bei Zugriff auf den SNMP-Dienst mittels IPSec
[54] Beispiel \\Rechnername\c$

des Systems informieren. Dies schließt Fehler in der Software sowie häufig vorhandene Konfigurationsfehler der Software ein.

Abhängig von den Zielen eines Penetrationstests, kann sich ein Angreifer oder ein Tiger Team bei Quellen wie Bugtraq, CERT/CC® oder dem Hersteller Microsoft über Schwachstellen informieren. Ein Angreifer sucht dabei nur diejenigen Schwachstellen, die ihm zum Erreichen seines Zieles helfen. Einem Angreifer muss im Extremfall nur eine Schwachstelle bekannt sein, mit deren Ausnutzung er sein Ziel erreichen kann. Weitere Schwachstellen braucht der Angreifer nicht zu suchen, womit sich seine Vorgehensweise von der Schwachstellenanalyse unterscheidet.

Ein Tiger Team, das während eines Penetrationstests die Rolle eines Angreifers annimmt, sollte aber möglichst alle Schwachstellen kennen, um möglichst schnell auf die Kundenwünsche reagieren und mehrere Varianten einer Bedrohung darstellen zu können. In Anhang E ist daher eine Übersicht aller bisher bekannt gewordenen Schwachstellen der IIS 5.0 enthalten. Zur Identifizierung der Schwachstellen sind zu jeder Schwachstelle die CVE-ID, eine Referenz auf eine zugehörige CERT/CC®-Vulnerability Note oder Advisory, die jeweilige Bugtraq-ID sowie die Microsoft Advisory angegeben. Aus der Menge der Schwachstellen wird eine kleine Auswahl in der Phase Penetration verwendet werden. Die Durchführung einer solchen Schwachstellenanalyse ist sehr zeitaufwendig, dafür aber auch sehr genau.

Für eine genaue Schwachstellenanalyse eines Systems sind allerdings genauere Informationen über die Konfiguration des Systems notwendig. So gibt ein eingespielter Patch die Evidenz, dass eine Schwachstelle bereits geschlossen ist. Informationen über eingespielte Patches können in vielen Fällen nicht durch die beschriebenen Reconnaissance Methoden erlangt werden. Zwar kann der Patch in vielen UNIX-Programmen anhand der im Banner enthaltenen Versionsnummer erkannt werden, bei Windows-Systemen besteht diese Möglichkeit nicht. Hier sind Informationen über die Patches in der Registry im Schlüssel HKLM\Software\Microsoft\Windows NT\Currentversion\hotfix enthalten. Für den Zugriff auf diesen Schlüssel ist ein administrativer Zugang notwendig.

Ist ein administrativer Zugang nicht gegeben, kann ein Angreifer oder ein Tiger Team durch Angriffsmethoden wie das Erraten von Passwörtern versuchen, sich einen solchen Zugang zu einem System zu verschaffen. Damit ist eine Penetration vor der eigentlichen Schwachstellenanalyse notwendig, wodurch der Aufwand der Vorgehensweise steigt. Eine zweite Methode der Informationen ist die Abwesenheit eines Patches vorauszusetzen. Dabei werden zunächst Informationen über Schwachstellen des Systems unabhängig von vorhandenen Patches gesammelt und versucht, diese Schwachstelle zu penetrieren. Hat die Penetration Erfolg, so ist die Abwesenheit eines Patches nachgewiesen. Eine solche Vorgehensweise wird häufig von Scannern verwendet, welche auf die Anwesenheit der Web Folder Traversal Schwachstelle (vgl. Abschnitt 6.7.1) prüfen. Diese Vorgehensweise ist sogar genauer, da ein Patch fehlerhaft sein kann, so dass eine Schwachstelle trotz installiertem Patch vorhanden ist. Mittels der Penetration kann so die Anwesenheit einer Schwachstelle gezeigt werden, was einen Vorteil des Penetrationstests darstellt (vgl. Abschnitt 9.1).

Die Erkennung von Schwachstellen kann auch mittels Scanner automatisiert werden. Die Möglichkeiten des Einsatzes eines solchen Vulnerability Scanners werden im nächsten Abschnitt behandelt.

6.6. Automatisierte Erkennung mittels Scanner

Vulnerability Scanner bieten eine Möglichkeit, die Erkennung der Schwachstellen zu automatisieren. Sie werden vielfach im Zusammenhang mit dem Begriff Penetrationstest verwendet, weshalb in diesem Abschnitt ihre Verwendung in Verbindung mit einem Penetrationstest untersucht werden. Dabei ist festzuhalten, dass die Anwendung eines Vulnerability Scanners kein Penetrationstest ist (vgl. Abschnitt 3.5). Der Einsatz eines solchen Scanners ist zwar durch Log-Einträge und Intrusion Detection Systeme leicht zu erkennen und wird daher von Angreifern vermieden. Da ein Scanner aber unter normalen Bedingungen von anderen Nutzern des Internet eingesetzt wird, sollten auch ohne Einwirkung des Tiger Teams entsprechende Einträge in den Log-Dateien eines an das Internet angeschlossenen Rechners vorhanden sein. Zudem steht ein Tiger Team unter Zeitdruck, der durch die Unterstützung eines Vulnerability Scanners gemildert werden kann.

In diesem Abschnitt wird an Hand des Szenarios betrachtet, inwieweit die betrachteten Vulnerability Scanner das Tiger Team bei ihrer Arbeit unterstützen können. Die dabei eingesetzten Scanner lassen sich gemäß Abschnitt 3.5.1 als passive, netzwerkbasierte Scanner klassifizieren. Zwei Scanner sind auf Webserver spezialisiert, weshalb für einen Vergleich nur diejenigen Ergebnisse, die sich auf Schwachstellen im Web-Server beziehen, in die Auswertung der Scanner einbezogen werden.

Die Scanner können neben der Vulnerability Detection auch die Reconnaissancephase und somit die Arbeit des Tiger Teams vor der Penetrationsphase unterstützen. Im Folgenden werden konkrete Produkte betrachtet.

6.6.1. LANGuard Network Security Scanner 3.3

Bei dem Network Security Scanner[55], kurz NSS, handelt es sich um einen kommerziellen Scanner der Firma GFI, der für nicht-kommerzielle Zwecke kostenlos erhältlich ist. Der Scanner kann auf Microsoft Windows Systemen installiert werden und von dort aus Windows und Unix Systeme untersuchen. Durch eine mitgelieferte Scriptsprache lässt sich der Scanner um individuelle Tests erweitern.

Der NSS unterstützt die Discovery durch einen Whois-Client sowie der Möglichkeit, traceroute zu verwenden. Bei der Enumeration können sämtliche in Abschnitt 5.3.2 besprochenen Methoden vom NSS automatisiert verwendet werden.

Der Scanner weist zwei Besonderheiten auf. Zum einen bietet der Scanner in der kostenpflichtigen Version ein Patch-Management. Mit diesem lassen sich gefundene Schwachstellen durch den Scanner automatisiert beheben. Zurzeit kann diese Funktion nur für die englischsprachigen Versionen der Microsoft Produkte Windows, Office, Exchange, SQL-Server und ISA-Server genutzt werden. Zudem kann zur Enumeration von NetBIOS-Eigenschaften neben der Methode Null-Session auch eine authentifizierte Verbindung hergestellt werden. Dazu kann ein Benutzername und ein Passwort vorgegeben werden. Sind keine Anmeldedaten bekannt, so bietet der NSS auch die Möglichkeit, Passwörter durch einen Brute-Force Angriff zu erraten. Damit kann die Qualität der Ergebnisse verbessert werden.

[55] http://www.gfi.com/lannetscan/, Download am 29.7.2003

Die Qualität ist somit von einer bestehenden NetBIOS-Anmeldung abhängig. Nutzt der Scanner lediglich eine Null-Session, so kann der NSS nur sehr wenige Schwachstellen finden. Dennoch ist die Leistung der Enumeration gut. So werden administrative Freigaben entdeckt. Die NetBIOS bzw. SMB Tests können individuell ausgewählt werden, auch die Abfrage der SNMP-Daten kann in einen Scan integriert werden. Der Bericht des NSS mit einer Null-Session ist in Anhang I1 enthalten.

Ist eine NetBIOS-Verbindung vorhanden, verbessern sich die Ergebnisse. Der Scanner testet dabei auf die Anwesenheit von Microsoft Patches. So kann bei einem fehlenden Patch darauf geschlossen werden, dass die Schwachstelle vorhanden ist. Somit erzielt der NSS sehr zuverlässige Ergebnisse. Ein Bericht bei vorhandener Anmeldung ist in Anhang I enthalten.

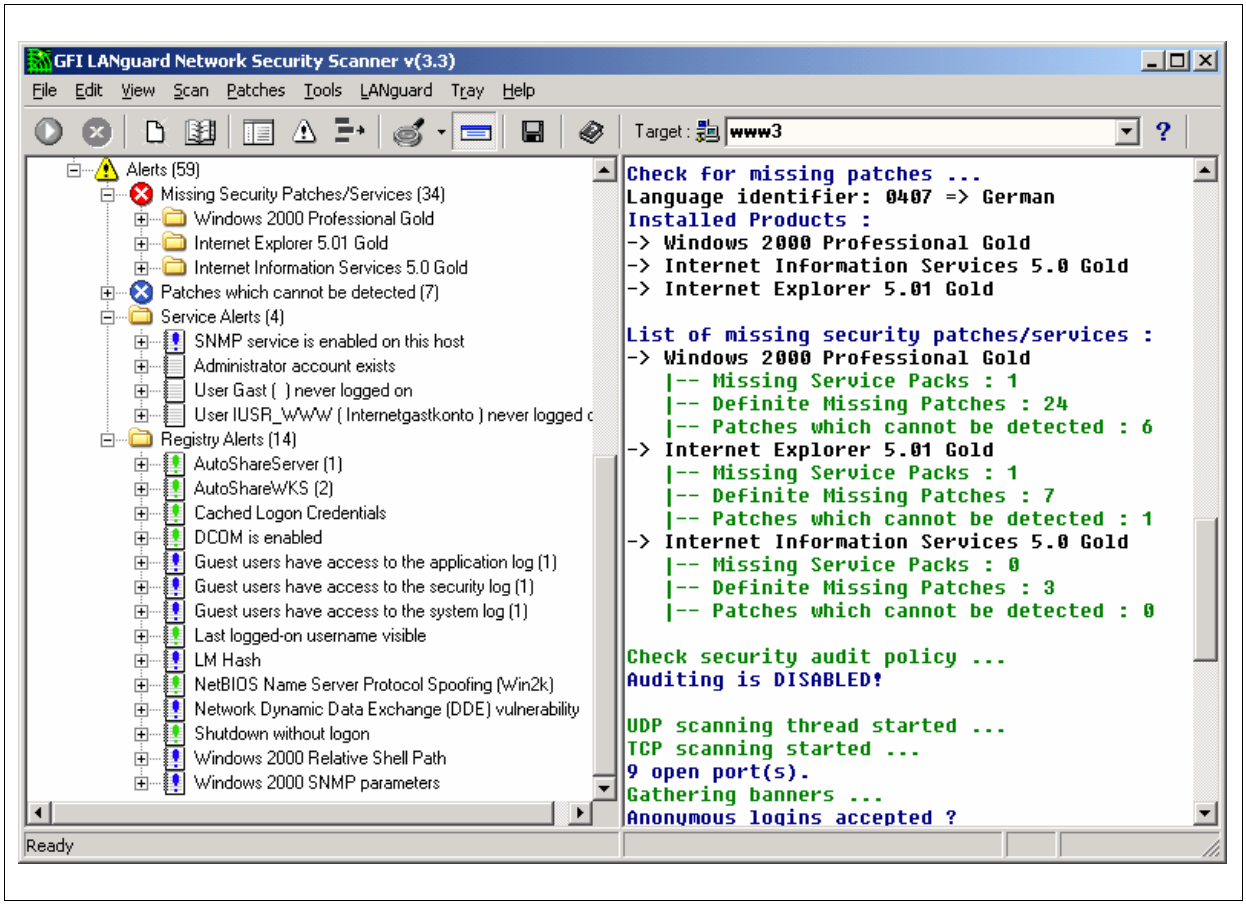


Abbildung 49: Der LanGuard Network Security Scanner

Der Scan ist nicht durch Betrachtung der Log-Dateien zu erkennen. Weder in Ereignis-, Anwendungs- und Sicherheitsprotokoll noch in den Log-Dateien des IIS-Webservers waren nach Durchführung eines Scans mittels des NSS neue Einträge hinzugekommen.

6.6.2. Nessus 2.0.7

Nessus[56] ist ein kostenloses Open-Source Produkt. Es verfügt über eine Server-Komponente für die Funktionalität und eine Client-Komponente zur Steuerung des Scanners. Während die Server-Komponente nur für UNIX-Systeme verfügbar ist, ist die Client-Komponente auch für Microsoft Windows verfügbar. Der Scanner wird seit 1998 unter der Leitung von Renaud Deraison entwickelt und wird ständig aktualisiert. Durch Plugins, die in der eigens für Nessus entwickelten Skriptsprache NASL geschrieben werden, kann Nessus flexibel erweitert werden. Für einige wenige Schwachstellen bietet der Scanner aktive Methoden, die aber besonders gekennzeichnet sind. Alle aktiven Methoden können durch eine Schaltfläche deaktiviert werden. Durch die Auswahl von Plugins können einzelne Testdurchläufe stark individualisiert werden. Abbildung 50 stellt die Auswahl der Plugins dar.

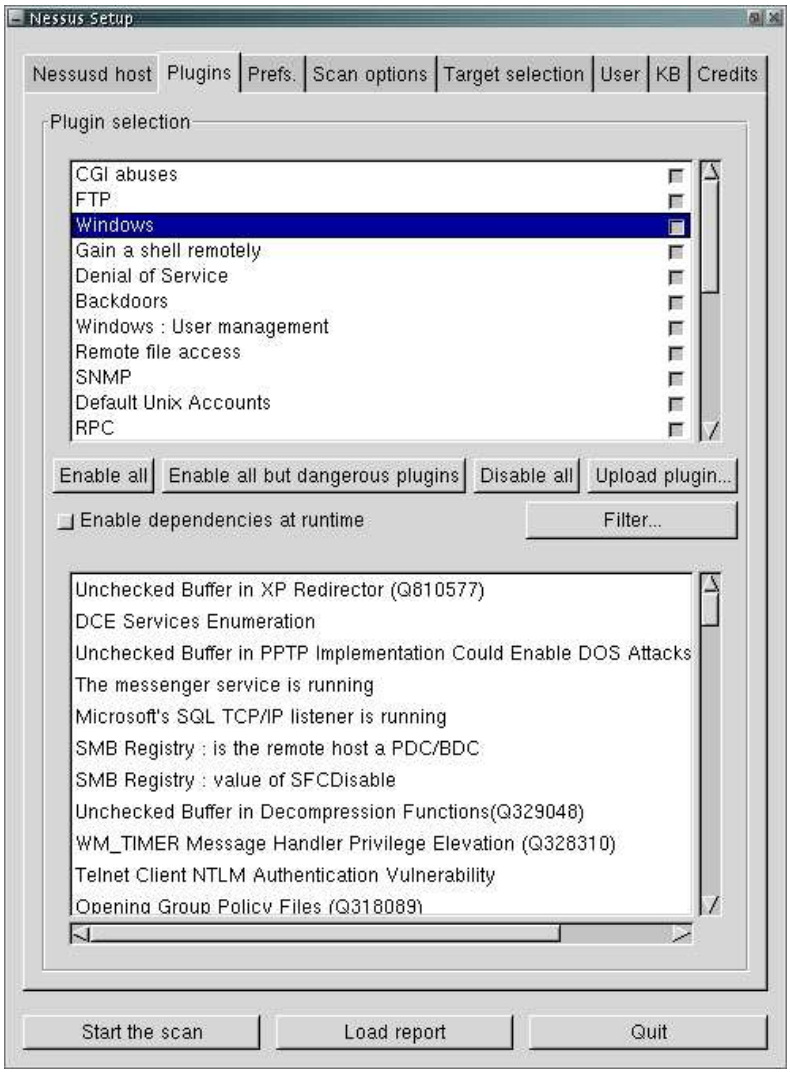


Abbildung 50: Nessus-Plugins (Quelle: http://www.nessus.org/demo/plugins.jpg)

[56] www.nessus.org, Download am 29.07.2003

Die Ergebnisse eines Scans werden in einem Report dargestellt. Sie sind nach der Portnummer sortiert und werden mit den Risikoklassen high, medium und low bewertet. Eine Warnung zum netbios-ns Dienst beinhaltet Informationen über Netbios-Namen, Arbeitsgruppe sowie der MAC-Adresse, die auch mittels des Tools enum erkannt werden konnten. Somit unterstützt der Scanner zwar die Enumeration, auf Grund der fehlenden Informationen über vorhandene Freigaben ist diese Unterstützung jedoch vernachlässigbar.

Der von Nessus generierte Bericht über die Schwachstellen des Opfersystems, der in Anhang I2 beigelegt ist, enthält 43 Schwachstellen, davon 8 mit hohem Risiko. Auf Port 80/www entfallen 17 Schwachstellen, davon 6 mit hohem Risiko. Eine Schwachstelle im Webserver (CA-2002-02) wurde doppelt gemeldet. Bei zwei Schwachstellen (CVE-1999-1011, CAN-2001-1408) handelt es sich um so genannte false-positives, die im betrachteten Opfersystem nicht enthalten sind. Bei einer Schwachstelle trifft der im Report enthaltene Hinweis, dass es sich um einen false-positive handeln könnte, nicht zu. Zwei Schwachstellen (CVE-2002-0333 und CVE-2001-0507) können mit dem Patch MS01-044 behoben werden. Daher sind sie im Bericht von Nessus zu einer Meldung zusammengefasst. Ein bereinigter Bericht über 14 im Opfersystem vorhandenen Schwachstellen, die in Verbindung zum Webserver stehen, ist in Tabelle 7 dargestellt.

Einer genaueren Erläuterung bedarf die Schwachstelle „Backup CGI download“. Da Skripte direkt auf dem Server ausgeführt werden, erhält der Aufrufer des Skriptes lediglich die Ausgabe des Skriptes. Der Quellcode des Skriptes ist jedoch nicht zugreifbar. Manche Editoren legen allerdings eine Backup-Datei im Verzeichnis ab. Emacs erzeugt von der Datei xyz die Backupdatei xyz~, UltraEdit die Datei xyz.bak. Ist eine solche Datei im Webverzeichnis, so kann der Quellcode erlangt werden. Die Gefahr besteht darin, dass der Quellcode sicherheitsrelevante Informationen wie beispielsweise Benutzernamen oder Passwörter für den Zugriff auf eine Datenbank enthält. So enthält die Datei /script/produkte.php der Alarm Mayer-Homepage die Zugangsdaten für die MySQL-Datenbank. Das gesamte Verzeichnis wurde von dem Rechner, auf dem die Homepage mit UltraEdit erstellt wurde, auf den Server www3 kopiert, wobei auch die von UltraEdit erstellte Backupdatei produkte.php.bak in das Verzeichnis gelangte. Sie konnte mittels eines beliebigen Webbrowsers auf einen Angreiferrechner kopiert und so die Zugangsdaten für die MySQL-Datenbank enthüllt werden. Diese Schwachstelle ist in den Archiven, die zur Erstellung der in Anhang E enthaltenen Schwachstellenanalyse des IIS 5.0 benutzt worden sind, nicht enthalten. Obwohl die Ergebnisse des Scanners unvollständig sind, sollte der Scanner der Vollständigkeit halber in die Erstellung einer Schwachstellenanalyse mit einbezogen werden. Zur Erkennung solcher Schwachstellen, die nicht in Schwachstellendatenbanken enthalten sind, eigenen sich Scanner wie Nessus. Hierbei sollte nur auf eine einzelne Schwachstelle getestet werden, damit der Einsatz des Scanners möglichst unentdeckt bleibt.

CVE	CERT/CC	Patch	Name / Description	Risk (Nessus)
CVE-2000-0884	-	MS00-078	Web Folder Travesal Vulnerability	High
CVE-2001-0507	-	MS01-044	System file listing privilege elevation	High
CVE-2001-0333	VU#789543	MS01-026 MS01-044	IIS Directory traversal vulnerability	High
-	-	-	Backup CGI-Download	High
	-	MS02-065	Buffer Overrun in MDAC could Lead to Code Excecution	High
-	-	-	WebDAV enabled	Medium
-	CA-2000-02	-	Cross Sote Scripting Vulnerability: Malicious HTML Tags Embedded in Client Web Requests	Medium
-	-	-	Cross Site Tracing	Medium
-	-	-	IIS sample-Application discloses physical path of web root	Low
CVE-2002-0074 CVE-2002-0075	CA-2002-02	MS02-018	Cross site Scripting Vulnerability in Help-file search facility of IIS	Medium
-	-	-	Cross site Scripting Vulnerability in the handling of overliong requests on an idc-file	Medium
CVE-2001-0500	CA-2001-13	MS01-033		Medium
CAN-2000-0071	-	Service Pack 3	Obtain Real pathname of document root by requesting non.-existent .ida or .idq-file	Medium
-	-	-	Banner discloses web sevrer type and version	Low

Tabelle 7: Ergebnisse des Nessus-Scan ohne False-positives

Weiter ist auch die Schwachstelle „WebDav enabled“ näher zu betrachten. Statt dieser generellen Meldung sollte für den Fall des fehlenden Patches auch auf die damit verbundene Gefahr der Ausnutzung eines Pufferüberlaufs, der in der Microsoft Advisory MS03-007 beschrieben ist, hingewiesen werden. Diese Schwachstelle ist bereits am 17. März 2003 veröffentlicht worden, so dass eine Nessus-Version vom 29. Juli 2003 auf diese Schwachstelle hinweisen sollte. Durch das Ausmaß dieser Schwachstelle sollte sie auch mit hohem statt mittlerem Risiko bewertet werden. Auch sollten die von Nessus gelieferten Informationen aktueller und vollständiger sein. Für die Reconnaissance bietet Nessus mittels Portscanner und NetBIOS-Enumeration nur geringe und unvollständige Unterstützung.

Nessus erzeugte beim Scan einige wenige Einträge im System-Ereignisprotokoll. Hierbei wurden aber lediglich Zugriffe auf nicht existierende Konten des FTp-Servers sowie die Anfrage an die cmd.exe über den Webserver gemeldet.

6.6.3. Sara 4.2.1e

Weiter wurde der Scanner Sara[57] betrachtet, der für Unix-Platformen verfügbar ist. Allerdings waren die Ergebnisse dieses Scanners, der nur bei etwa jedem fünften Aufruf korrekt startete, sehr enttäuschend. Der Scanner zeigte lediglich eine einzige webbasierte Schwachstelle auf. Die weiteren Informationen zu dieser Schwachstelle beschränkten sich auf die Angabe des kumulativen Patches, der diese Schwachstelle behebt, sowie der weiteren Schwachstellen, die ebenfalls von dem Patch behoben werden. Die Ergebnisse von Sara werden daher nicht weiter betrachtet.

6.6.4. Cerberus Internet Scanner 5.0.02

Der Cerberus Internet Scanner wird nicht mehr weiterentwickelt, da der Hersteller Cerberus von der Firma atstake übernommen wurde. Die aktuellste Scannerversion stammt aus dem Jahr 2000. Der Scanner liefert nur schlechte Ergebnisse. Die erkannten Schwachstellen beruhen vor allem auf Konfigurationsfehlern. Auch Informationen zum beheben der Schwachstellen fehlen. Die Leistungen der NetBIOS Enumeration sind gut, lassen sich allerdings auch mit dem Tool enum herausfinden. Die Ergebnisse über Web und NetBIOS sind in Anhang I3 enthalten.

6.6.5. Stealth-http 2.0

Stealth-http[58] ist ein auf Webbrowser spezialisierter Vulnerability Scanner. Die Entwicklung wird unter dem Namen N-Stealth von der Firma N-Stalker weiterentwickelt. Da das aktuelle N-Stealth 5.0 kostenpflichtig ist, wird in dieser Arbeit die allerdings veraltete Freeware-Version vom 20.2.2002 benutzt.

Der Scanner liefert keine überzeugenden Ergebnisse. Zwar erkannte Stealth-http 152 Schwachstellen, viele Meldungen sind allerdings nur Variationen einer bereits gemeldeten Schwachstelle. Allein die Web Folder Traversal Vulnerability wurde 55 mal gemeldet. Nach Bereinigung der Ergebnisse erkannte der Scanner nur 8 Schwachstellen. Des Weiteren

[57] www-arc.com/sara, Download am 29.07.2003
[58] http://www.devhood.com/tools/tool_details.aspx?tool_id=353, Download 30. Juli 2003

beinhaltet der Bericht, auch aussagelose Informationen, wie die in Abbildung 51 dargestellte Meldung. Auf Grund dieser Unzulänglichkeiten wird auf die Ergebnisse von Stealth-http nicht weiter eingegangen.

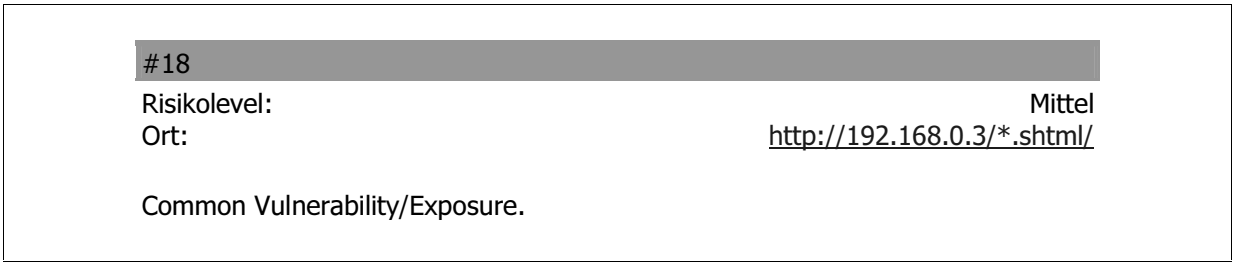


Abbildung 51: Aussageloses Ergebnis des Stealth-http Scanners

6.6.6. Nikto 1.30

Nikto[59], eine Weiterentwicklung von Whisker, ist ein in perl geschriebener Scanner, der auf Schwachstellen von Webservern spezialisiert ist. Das Ergebnis eines Scans ist ein sehr unübersichtlicher und zum Teil irreführender Bericht, da sowohl vorhandene als auch nicht vorhandene Schwachstellen dargestellt werden. So ist auch am Ende des Berichts ein markanter Satz enthalten: „server answering all requests with a "200 OK" message. You should manually verify your results“. Der Bericht, der Anhang I4 zu entnehmen ist, führt lediglich Schwachstellen auf, die über die URL zu erkennen sind.

6.6.7. Fazit der Vulnerability Scanner

Keiner der betrachteten Scanner kann alle Schwachstellen finden, weshalb für einen Angreifer oder ein Tiger Team eine manuelle Analyse notwendig ist. Bei der Reconnaissance ist nur der NSS als Tool einsetzbar. Nessus bietet einige Plugins wie der Test auf Backup-Dateien, die einzeln auf ein Testsystem angewendet werden sollten, um Schwachstellen zu entdecken, die nicht in den gängigen Archiven vorhanden sind. Werden sie separat angewendet, sollte der von Nessus erzeugte Verkehr sich in Grenzen halten.

Neben den hier betrachteten frei erhältlichen Scannern gibt es auch kommerzielle Produkte, die dem Autor nicht zur Verfügung standen. Ob deren Leistungen besser sind, muss daher offen bleiben. Wichtig wäre bei einem passiven Scanner, der zur Unterstützung der Phasen Reconnaissance und Vulnerability Detection verwendet wird, dass er ein vollständiges Ergebnis produziert. Soll ein solcher Scanner bei einem verdeckten Test eingesetzt werden, sollte ein Scan in kleinen, zeitversetzten Schritten durchgeführt werden können, damit der Scan auf Grund des erzeugten Verkehrs nicht allzu auffällig ist.

6.7. Penetration

Zur Reduktion des Aufwandes wurden zwei bestimmte Schwachstellen aus der in Anhang E gegebenen Liste ausgewählt und der Penetrationstest auf diese Schwachstellen beschränkt. Dabei soll die Wirkung einer sehr einfach auszunutzenden Schwachstelle, sowie eines

[59] http://www.cirt.net/code/nikto.shtml, Download 30. 7. 2003

typischen Pufferüberlaufs betrachtet werden. Diese Angriffe dienen einem fiktiven Angreifer zur Manipulation von Daten auf dem System.

Die bei den Versuchen betrachteten Schwachstellen sind bekannte Schwachstellen, die am heutigen Tage durch Patches geschlossen sein sollten. Dennoch haben Vorfälle der Vergangenheit, die unter anderem von Würmern wie W32/Nimda@mm erzeugt worden sind, gezeigt, wie verletzlich die Systeme trotz existierender Patches gegen längst bekannte Angriffe sind, so dass die Versuche noch heute einen realistischen Hintergrund haben. Zudem werden zukünftige Angriffe dieselben Prinzipien befolgen, so dass die im Folgenden durchgeführten Versuche prinzipiell auch zukünftige Verletzlichkeiten der Systeme und deren Auswirkungen demonstrieren können.

Wie einem Beispiel aus [Payne01:3] zu entnehmen ist, hören die meisten der im Internet erhältlichen Anleitungen an dieser Stelle auf. Da die Durchführung der Angriffe das wesentliche Merkmal des Penetrationstests ist, kann hier darauf nicht verzichtet werden. Allerdings bestehen an dieser Stelle ethische und rechtliche Bedenken, da die Dokumentation der Versuche für die Schulung von Hackern missbraucht werden könnte. Daher werden keine genauen Befehle angegeben. Die Penetration wird lediglich in groben Schritten erläutert. Eine genauere Beschreibung ist daher nur im Anhang K enthalten, der ausschließlich dem Autor selbst und den Betreuern vorbehalten ist. Ebenso wird mit den Log-Dateien der Zielsysteme vorgegangen, da die Gefahr besteht, dass die genaue Vorgehensweise der Versuche enthüllt werden kann.

6.7.1. Web Folder Traversal

Eine Traverisierung (engl.: traversal) ist nach [Duden5:739] eine Querverbindung oder eine Ausweichbewegung. Im Kontext eines Webservers ist es einem Angreifer durch Ausnutzen einer Schwachstelle metaphorisch gesehen möglich, den Sicherheitsmechanismen auszuweichen und so eine Querverbindung zu einem Verzeichnis außerhalb des Web-Verzeichnisses zu schaffen.

Eine solche Schwachstelle ist die am 10.10.2000 in einem Forum auf packetstorm[60] diskutierte Schwachstelle CVE-2000-0884, die durch das eine Woche später erschienene Microsoft Security Bulletin MS00-78 mit dem Titel „Web Server Folder Traversal“ behandelt wurde. Im Bulletin wird auf einen Patch verwiesen, der bereits im Security Bulletin MS00-057 behandelt worden ist. Bei Ausnutzung der Schwachstelle ist es dem Angreifer möglich, durch eine bestimmte URL mit den Privilegien des Internetgastkontos Kommandos auszuführen. Auf Grund der Zugehörigkeit des Internetgastkontos zu der Gruppe „Gäste“, kann der Angreifer auf sämtliche Dateien im Dateisystem Zugriff bekommen.

Hintergrund dieser Schwachstelle ist die fehlerhafte Behandlung von Unicode-Sequenzen in einer URL. Durch die Eingabe einer URL der Form http://www.ziel.local/../../autoexec.bat könnte theoretisch auf die Datei autoexec.bat im Hauptverzeichnis zugegriffen werden, sofern das Web-Verzeichnis das voreingestellte Verzeichnis C:\inetpub\wwwroot ist. Eine solche Anfrage wird aber durch einen Webserver innerhalb seiner Sicherheitskontrollen unterbunden, da der Client nur Zugriff auf das Web-Verzeichnis haben soll. Das Unterbinden der Sequenz ../ kann aber dadurch umgangen werden, in dem entweder das Zeichen ‚.’

[60] www.packetstormsecurity.nl

(Punkt) oder ‚/' (Slash) durch eine Unicode-Repräsentation ersetzt wird. Unicode kann wie der ASCII-Standard als eine Abbildung von binären Zahlen auf eine Menge von Zeichen verstanden werden, wobei Unicode zwecks besserer internationaler Unterstützung mehr Zeichen beinhaltet und somit ein Unicode-Zeichen durch 16 Bits repräsentiert ist. Die Umgehung ist möglich, da die Serversoftware die Dekodierung von Unicode Sequenzen erst nach den Sicherheitskontrollen durchführt.

Unter dem Begriff Web Folder Traversal ist im Allgemeinen nur die eben beschriebene und in [MS00-078] veröffentlichte Schwachstelle bekannt, deren Ursache in der Umwandlung der Unicode-Repräsentationen liegt. Sie ist jedoch nicht die einzige Schwachstelle, die ein Web Folder Traversal ermöglicht. Werden die Zeichen Punkt oder Slash nicht nur durch das zugehörige Unicodezeichen, sondern gleichzeitig ein Escapezeichen in die Repräsentation eingebaut, so ist immer noch eine Web Folder Traversal möglich. Die Umgehung der Sicherheitskontrollen mittels Escapesequenzen ist im Microsoft Security Bulletin MS01-026 beschrieben. Allerdings ist in MS01-026 keine Verbindung zum Bulletin MS00-078 vorhanden, obwohl auf Bugtraq[61] URLs mit Escapesequenzen als Exploits für die in MS00-078 beschriebene Schwachstelle bekannt waren.

Einer Web Folder Traversal liegen daher die folgenden zwei Schwachstellen zu Grunde:

1. Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability (BugTraq ID 1806, CVE-2000-0884, MS00-078)
2. Microsoft IIS/PWS Escaped Characters Decoding Command Execution Vulnerability (BugTraq ID 2708, CVE-2001-0333, MS01-026)

Die zwei Schwachstellen lassen sich auch anhand der Log-Einträge unterscheiden. In Abbildung 52 ist ein Auszug aus der Log-Datei eines IIS-Webservers dargestellt. Der erste Eintrag von 11:18:13 Uhr ist bei Ausnutzung der ersten Schwachstelle (CVE-2000-0884) erzeugt worden. Hier ist zu sehen, dass die ursprünglich eingegebene Unicodesequenz in ../.. umgewandelt worden ist. Der zweite Eintrag um 11:18:18 ist durch Ausnutzung der zweiten Schwachstelle (CVE-2001-0333) erzeugt worden. Hier wurde das Escapezeichen aus der URL entfernt.

[61] www.securityfocus.com/bid

```
 #Software: Microsoft Internet Information Services 5.0
 #Version: 1.0
 #Date: 2003-08-22 11:18:13
 #Fields: time c-ip cs-method cs-uri-stem sc-status
11:18:13 192.168.0.18 GET /scripts/../../winnt/system32/cmd.exe 200
11:18:18 192.168.0.18 GET /scripts/..%5c../winnt/system32/cmd.exe 200
```

Abbildung 52: Log-Einträge des IIS bei Ausnutzung der Schwachstelle

Die besondere Gefahr der Web Folder Traversal Schwachstelle ist die Leichtigkeit, mit der sie ausnutzbar ist. Nahezu jeder, der über einen Browser[62] verfügt und diesen bedienen kann, kann diese Sicherheitslücke ausnutzen, sofern der Webserver verwundbar ist. Benötigt werden somit nur geringe Fähigkeiten und keine speziellen Programme. So wurden die Schwachstellen auch durch die Würmer W32/CodeBlue.worm, W32/Nimda@mm und Solaris/Sadmind.worm[63] ausgenutzt. Trotz der Verfügbarkeit von Patches konnten sich die Würmer verbreiten und zum Teil Schäden anrichten.

In dem hier beschriebenen Szenario wird die zweite Schwachstelle ausgenutzt, um eine Penetration zu erreichen. Das Ziel ist, eine Verunstaltung (engl.: Defacement) der Webseite zu erreichen, sowie eine Hintertür zu installieren, mit dem das System von außen gesteuert werden kann. Dies soll dem Betreiber des Webservers dazu motivieren, die Sicherheit des Servers zu erhöhen.

Die Versuche zur Web Folder Traversal geschah unter Mitwirkung der Vorfallserzeugungsgruppe des IRT-Projektes. Sie bestand zu diesem Zeitpunkt aus Christopher Alm, Björn Bartels, Samira Razai und Torsten Sorger, wobei letzterer auch an der Installation der Systeme beteiligt war. Auch die Einschleusung einer Hintertür auf einem NTFS-System wurde von der Gruppe durchgeführt.

Zur Durchführung des Defacements wurde auf dem Angreiferrechner Belzebub eine HTML-Datei mit dem Namen index.html erzeugt, die im Browser nur den Text „This site was hacked“ ausgibt. In der Titelzeile des Browserfensters ist „Hacker on board“ angegeben. Die Ausgabe ist in Abbildung 53 dargestellt.

[62] Die in dieser Arbeit durchgeführten Penetrationstests zeigten auf, dass die Ausnutzung dieser Schwachstelle mit einem Netscape/Mozilla Browser Probleme machte

[63] Der Solaris/Sadmin.worm infiziert Solaris-Systeme, die dann verwundbare IIS-Systeme angreifen.

Abbildung 53: Defacement auf Grund eines Web Folder Traversal

Zunächst wurde die auf dem Opfersystem gespeicherte Webseite der Firma Alarm Mayer angeschaut, um zum einen die Verfügbarkeit des Servers festzustellen. Bei der Anwendung des Exploits steht der Angreifer jedoch vor dem Problem, dass ihm bekannt sein muss, in welches lokale Verzeichnis das Web-Verzeichnis ist. Dazu kann er sich der Schwachstelle CAN-2000-0071 bedienen, deren Ausnutzung das physikalische Verzeichnis enthüllt, sofern der Webserver verwundbar ist.

Abbildung 54: Enthüllung des Webverzeichnisses

Um festzustellen, ob der Exploit auf dem Opfersystem möglich ist, wurde mittels des Befehls `dir` das Webverzeichnis angeschaut. Daraufhin konnte geschlossen werden, dass die Hauptdatei der Webseite die Datei index.html ist, so dass die verunstaltete Datei als index.html gespeichert wurde. Damit die Datei auf das Opfersystem übertragen werden kann, wurde auf dem Angreifer Belzebub der TFTP-Server TFTPD32 2.0 von Philippe Jounin installiert und gestartet. Die verunstaltete index.html wurde in das Wurzelverzeichnis des TFTP-Servers kopiert.

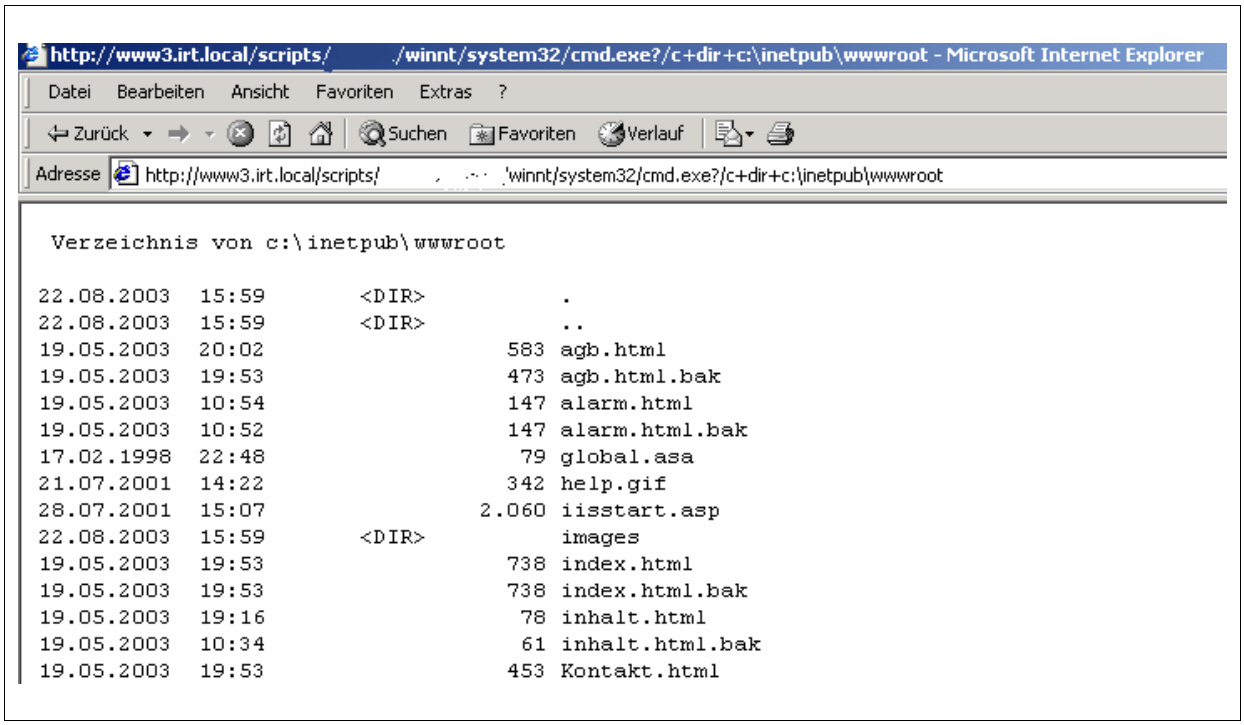


Abbildung 55: Ausnutzung der Web Folder Traversal zum Anschauen eines Verzeichnisses

Unter Ausnutzung der Schwachstelle wurde der in der Standardkonfiguration von Windows verfügbare tftp-client für die Kommandozeile verwendet, um die index.html des Opfersystems mit der index.html des Angreifers zu überschreiben. Um den Erfolg des Austausches zu kontrollieren, wurde die Webseite erneut angeschaut. Unter der Adresse http://www3.irt.local/ war nun die in Abbildung 53 dargestellte Webseite zu sehen.

Weiter wurde mittels des oben angewendeten Exploits eine Hintertür eingebaut, diese wurde mittels Back Orifice 2000, kurz BO2K, installiert. Obwohl Back Orifice 2000 häufig als solches bezeichnet wird, ist die Software kein Trojanisches Pferd, da es weder über Dysfunktionen verfügt und alle Funktionalitäten dem Benutzer bekannt sind. Dennoch kann es in die Klasse der Malware eingeordnet werden.

Nach [RFC2828:19] wird unter einer Hintertür (engl.: backdoor) ein Hardware- oder Softwaremechanismus verstanden,

- der Zugriff auf einem anderen als dem gewöhnlichen Wege bietet.
- der vorsätzlich von Systemdesignern oder dessen Wartungspersonal hinterlassen wurde.
- der gewöhnlich nicht allgemein bekannt ist.

Aus Sicht des Administrators eines Systems bietet die Installation eines BO2K-Servers einen Zugriff auf einem anderen als dem gewöhnlichen Wege, ist nicht allgemein bekannt und erfüllt somit das Kriterium einer Hintertür. Während manche Autoren Hintertür und Falltür (engl.: trap door) als äquivalente Begriffe ansehen (vgl. [Garfinkel96:329] und [Pfleeger00:179]), definiert Shirey [RFC2828:179] eine Falltür als [RFC2828:179] „a hidden computer mechanism (usually software) installed by an intruder, who can activate the trap door to gain access to the computer without being blocked by security services or

mechanisms“. Daher sollte Back Orifice 2000 genaugenommen als Falltür bezeichnet werden. Da die Installation von BO2K aber ein unbekannter Weg ist, Zugriff auf einem ungewöhnlichen Wege zu erlangen, soll ein für den Zweck des unautorisierten Zugriffs installiertes BO2K hier als Hintertür bezeichnet werden.

Für die Installation von BO2K auf dem Opfersystem wurde auf dem Angreifersystem Belzebub ein BO2K-Server vorbereitet, der über den Port 12345/tcp[64] gesteuert werden kann. Der Server wurde als bo2k.exe durch den TFP-Server auf Belzebub bereitgestellt. Durch die Fernsteuerung des Opfersystems über den Browser des Angreifersystems, wurde die Datei bo2k.exe mittels des Aufrufs des kommandozeilenorientierten tftp-clients des Opfersystems von dem Angreifer Belzebub auf das Opfersystem www3 kopiert. Jeder Aufruf einer ausführbaren Datei auf dem Opfersystem durch Ausnutzung der Schwachstelle wurde vom Browser mit der Fehlermeldung „CGI Error: The specified CGI application misbehaved by not returning a complete set of HTTP headers. The headers did return are:“ beantwortet. Der CGI-Meldung folgen eventuelle Fehlermeldungen des Programms.

Auf dem Opfersystem ist die Datei bo2k.exe mit einem Schreibschutz versehen. Dies behindert zwar nicht die Ausführung der Datei, erschwert dem Angreifer aber seine Spuren zu verwischen, so dass zunächst mit dem Befehl attrib die Attribute zurückgesetzt wurden. Im nächsten Schritt wurde die Datei bo2k.exe aufgerufen, bei deren Ausführung die Datei %systemroot%\umgr32.exe erzeugt und ausgeführt wurde. Somit konnte das Opfersystem mit dem in Abbildung 56 dargestellten Back Orifice 2000 Clients ferngesteuert werden. Allerdings wurde entgegen der Spezifikation von Back Orifice 2000 in der Registry unter dem Schlüssel HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run kein Eintrag für die Datei UMGR32.exe eingefügt, so dass Back Orifice 2000 nach einem Neustart nicht mehr startet. Ursache sind fehlende Rechte, da sämtliche Dateien, die durch ein Web Folder Traversal aufgerufen werden, mit den Rechten des Internetgastkontos IUSR_www3 ausgeführt werden. Dieses Konto wird von dem Server für Besucher von WebSeiten verwendet und ist Mitglied der Gruppe Gäste.

[64] Da viele Scanner nur auf Anwesenheit von BO2K schließen, wenn es auf seinem Standardport 54320/tcp hört, soll hier gezeigt werden, dass BO2K auch über einen andern Port erreichbar sein kann. In diesem Fall wurde zur Verwirrung von Scannern der voreingestellte Port des maliziös verwendbaren Tools NetBus gewählt.

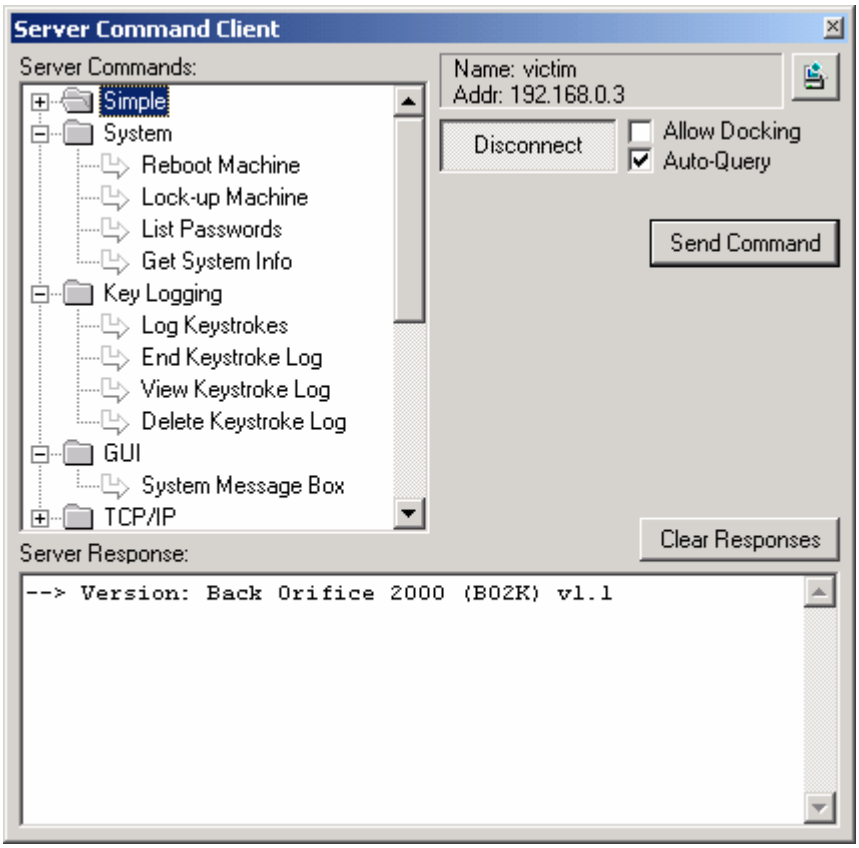


Abbildung 56: Back Orifice 2000 Client

An dieser Stelle ist darauf hinzuweisen, dass die mittels BO2K durchgeführten Versuche sehr unrealistisch sind, da kaum ein geschäftlich betriebenes Windows 2000 System das Dateisystem FAT32 verwendet, das auf Grund der Einschränkungen durch die Rechtestruktur beim Vergleich zweier Festplatten benutzt wurde. Auf einem NTFS System verfügt das Internetgastkonto zwar über genügend Rechte, die Datei bo2k.exe im wwwroot oder Hauptverzeichnis des Opfersystems zu speichern, jedoch fehlen ihm die Rechte, die zum Auszuführen der Datei benötigt werden. Somit wird der Angriff unter realistischen Bedingungen nicht so wie hier gezeigt möglich sein. Jedoch sind Schwachstellen bekannt, mit denen die Rechte des Angreifers eskaliert werden können, so dass Back Orifice 2000 unter Ausnutzung einer solchen Schwachstelle gestartet werden könnte. Ein Beispiel für eine Eskalations-Schwachstelle ist die „IIS Out of Process Privilege Escalation Vulnerability“ (CAN-2002-0869, MS02-062, BugTraq ID 6069).

6.7.2. ntdll.dll Overflow through WebDAV (MS03-007)

Am 17. März 2003 wurde das Microsoft Security Bulletin MS03-007 veröffentlicht. Inhalt des Bulletins ist eine Schwachstelle in der WebDAV-Komponente des bei Windows 2000 mitgelieferten IIS 5.0, durch die mittels eines Pufferüberlaufs ein Angreifer beliebigen Code von einem anderen Rechner aus auf einem betroffenen Server ausführen kann.

Nach weiteren Untersuchungen wurde aber erkannt, dass der Fehler nicht in der WebDAV – Komponente, sondern in der von der WebDAV-Komponente benutzten ntdll.dll enthalten war, die zum Kern des Betriebssystems gehört. Somit war die Ausnutzung der Schwachstelle nicht ausschließlich durch die Benutzung von WebDAV möglich, sondern auch über andere Programme, welche die ntdll.dll benutzen. Dabei wurde auch erkannt, dass neben Windows 2000 auch NT 4.0 und XP Systeme betroffen sind.

In diesem Szenario soll die oben beschriebene Schwachstelle eines ungeprüften Puffers in der ntdll.dll (CAN-2003-0109[65], BugTraq ID 7116) mittels WebDAV ausgenutzt werden. WebDAV (Web Distributed Authoring and Versioning) ist eine in [RFC2518] beschriebene Erweiterung des HTTP/1.1 Protokolls, die der Verwaltung der auf einem Webserver gespeicherten Inhalte von einem Client aus dient. Dies ist neben FTP eine weitere Möglichkeit für den Autor einer Homepage, um die Dateien auf den Server zu kopieren und dort zu pflegen.

Ziel der Angriffe in diesem Szenario ist neben der erneuten Verunstaltung der Webseite auch die Verwischung der Spuren, indem die Log-Dateien auf dem System gelöscht werden. Zudem soll auch ein administrativer Zugang geschaffen werden, der auch nach dem Vorfall besteht. Dazu wurde sich eines Exploits bedient, der einen Pufferüberlauf erzeugt. Generelle Vorgehensweise solcher Exploits ist das Einschleusen von Code, der dem Angreifer eine Shell über das Netzwerk bereitstellt. Als Shell wird bei Windows-NT[66] Betriebssystemen %systemroot%/system32/cmd.exe genutzt. Zur Bereitstellung der Shell nutzen die Exploits zwei Möglichkeiten. Erstere ist das Öffnen eines Ports auf dem Opfersystem, über den der Angreifer eine Verbindung öffnen kann, mit der die Shell bedient wird. Bei der zweiten Möglichkeit geht die Verbindung von dem Opfersystem aus. Hierbei öffnet der Angreifer auf dem Angreifersystem einen Port, was beispielsweise mit dem Tool Netcat von AtStake im Listenmodus geschehen kann. Der Port ist von dem Exploit vorgegeben. Wird der eingeschleuste Code auf dem Opfersystem ausgeführt, verbindet sich das Opfersystem mit dem Angreifersystem, und bietet dem Angreifer eine Shell, die er beispielsweise über Netcat bedienen kann.

Unter den für die in [MS03-007] beschriebene Schwachstelle existierenden Exploits gibt es neben einer Variante, die den Benutzer matt mit dem Passwort 1234 einrichtet, auch beide beschriebenen Shell-Varianten. Für dieses Szenario wurde ein im Internet verfügbarer Exploit benutzt, dessen Name aus ethischen und rechtlichen Gründen geändert wurde. Der Exploit wird in dieser Arbeit webdav-Exploit genannt. Als Dateiname wurde exwebdav gewählt.

[65] Die Schwachstelle hat bei CVE noch „Canidate“-Status. Eine Aufnahme in die Liste der Common Vulnerabilities and Exposures ist von einer Abstimmung abhängig, die noch erfolgen kann. In diesem Falle ist die Schwachstelle unter der Bezeichnung CVE-2003-0109 verzeichnet.
[66] Man beachte: Windows 2000 = Windows NT 5.0 und Windows XP = Windows NT 5.1

Der webdav-Exploit ist nur wirksam, wenn WebDAV auf dem Zielsystem aktiviert ist. Um dies festzustellen, genügt es, eine HTTP-OPTIONS-Anfrage an das Zielsystem zu schicken. Die Teile der in Abbildung 57 dargestellten Antwort, die ein aktiviertes WebDAV erkenntlich machen, sind fett markiert.

```
bash-2.05b# telnet 192.168.0.3 80
Trying 192.168.0.3...
Connected to 192.168.0.3.
Escape character is '^]'.
OPTIONS * HTTP/1.0

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 27 Aug 2003 15:39:01 GMT
Content-Length: 0
Accept-Ranges: bytes
DASL: <DAV:sql>
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL,
PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL,
PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
Cache-Control: private

Connection closed by foreign host.
```

Abbildung 57: Überprüfung eines Webservers auf Anwesenheit von WebDAV

Nach [Nelieβen02:15] ist das Erraten der richtigen Return-Adresse, die auf den eingespielten Code zeigt und zur Ausführung bringt, ein wesentliches Hindernis bei der Erzeugung eines Pufferüberlauf. So muss die Return-Adresse in dem verwendeten WebDAV-Exploit in der Kommandozeile angegeben werden.

Die Return-Adresse kann mit Hilfe eines Debuggers herausgefunden werden. Dazu ist der Prozess mit dem Debugger zu beobachten, wobei der Pufferüberlauf eine Exception verursacht. Bei Auslösung der Exception beendet der Debugger die Beobachtung im aktuellen Zustand, in dem die Exception verursacht wurde. Danach ist der eingeschleuste Shellcode im Speicher zu suchen. Wird er gefunden, so ist auch die Return-Adresse bekannt.

Es war mir unter Verwendung des Debuggers OllyDBG nicht möglich, den eingeschleusten Code zu finden. Daher wurden ein Brute-Force Versuch durchgeführt, um die Return-Adresse zu finden. Problematisch an dieser Methode ist der Umstand, dass der IIS-Webserver bei diesem Versuch abstürzt, das Skript aber weiter läuft. Um dieses Hindernis zu umgehen, wurde für das Finden der Return-Adresse mittels Brute Force Windows 2000 professional, Internet Information Services 5.0 sowie ein simpler Server auf einer Wechselplatte installiert. Dieser simple Server ist eine modifizierte Version des „Simple TCP/UDP Server“, der im Quellcode in den Beispielen des Microsoft Core Plattform SDK enthalten ist. Er wurde um den Befehl RESTART erweitert, der den Webserver-Prozess w3svc neu startet. Der modifizierte Quellcode ist in Anhang G5 beigelegt. Diese Installation diente als Zielsystem, auf dem zunächst kein Service Pack installiert war.

Das Skript, das auf dem Angreifer Belzebub unter Linux ausgeführt wurde, testet alle 15 Sekunden mittels eines Portscans, ob der Webserver erreichbar ist. Ist dies nicht der Fall, so startet es den Server neu und führt den Brute Force weiter.

Die Versuche wurden mit den Service Packs 1 bis 4[67] wiederholt. Dabei wurde erkannt, dass der Angriff nur auf Systeme mit Service Pack 2 oder Service Pack 3 möglich ist. Da die Konfiguration von www3 kein Service Pack enthielt, wurde auf dem Rechner www3 das Service Pack 3 für Windows 2000 installiert, um den Angriff durchführen zu können.

Mittels der bekannten Return-Adresse wurde der Exploit angewendet, so dass auf dem Angreifer System über Port 2512[68] auf eine Shell zugegriffen werden kann. Der Zugriff auf die Shell erfolgt über Telnet und ist in Abbildung 58 dargestellt. Durch den Befehl whoami[69] ist zu erkennen, dass die Shell über System-Rechte verfügt. Somit hat der Angreifer nicht die eingeschränkten Rechte des Internetgastkontos IUSR_WWW3, sondern die Rechte des IIS Dienstes, der in diesem Fall die Privilegien des LocalSystem-Kontos besitzt. Mit diesen Rechten hat der Angreifer einen uneingeschränkten Zugriff auf das System.

```
# telnet 192.168.0.3 2512
Trying 192.168.0.3...
Connected to 192.168.0.3.
Escape character is '^]'.
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>whoami
NT-AUTORITÄT\SYSTEM
```

Abbildung 58: Shell auf dem Opfersystem

Mit dieser Shell wurde nun mit den Befehlen `net user` und `net localgroup` ein Benutzerkonto `hacker` mit administrativen Rechten angelegt. Mit diesem Benutzerkonto wurde nun eine authentifizierte Verbindung zur Freigabe \\www3\IPC$ hergestellt. Dies ermöglicht eine Verbindung mit der Microsoft Management Console und dem Registry-Editor RegEdt32 zu dem System www3. Um weiterhin einen Shell-Zugriff zu haben, wurde der Telnet-Server eingerichtet. Dazu musste die NT-LANmanager Authentifizierung deaktiviert werden, wozu dem Schlüssel \\www3\HKLM\SOFTWARE\Microsoft\TelnetServer\1.0\NTLM der Wert 0 zugewiesen wurde. Nach der Änderung konnte über die von dem webdav-Exploit geöffnete Shell durch den Befehl net start der Telnet Server gestartet werden.

Nach der Einrichtung des Telnet-Zugangs wurde die vom webdav-Exploit geöffnete Shell beendet, da diese sehr instabil läuft und nach einem Absturz der Exploit bis zum Neustart des Opfersystems nicht mehr funktioniert. So wurde im Folgenden der Telnetzugang genutzt. Damit wurde das wwwroot nach php Dateien durchsucht. Die gefundene Datei produkte.php wurde mit dem Befehl type auf dem Bildschirm ausgegeben. Sie enthält die Benutzerdaten für den MySQL-Server, für die das MySQL-Konto root verwendet wird. So hat der Angreifer

[67] Das Service Pack 4 für Windows 2000 ist zum Zeitpunkt der Versuchsdurchführung das aktuellste.
[68] Port ist frei wählbar
[69] Teil des Windows 2000 Resource Kits, siehe http://www.microsoft.com/windows2000/techinfo/reskit/tools/default.asp

volle Rechte für den Zugriff auf die MySQL-Datenbank, die er aber nicht ausnutzt. Dennoch ist das Ziel der Manipulation der Datenbank ermöglicht worden.

Mittels des Befehls echo wurde von der Kommandozeile aus die alte Website überschrieben. Die neue Website hat den einfachen Inhalt „hacked by attacker“ und ist in Abbildung 59 dargestellt.

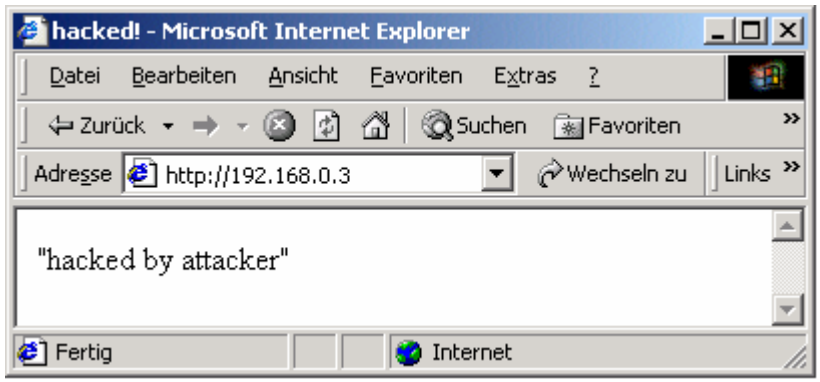


Abbildung 59: Webseite nach dem zweiten Defacement

Da das Opfersystem www3 über administrative Freigaben verfügt, wurde mit dem Benutzerkonto hacker eine SMB-Verbindung zu der Freigabe \\www3\c$ hergestellt. Über diese Verbindung wurden die logs des IIS gelöscht, die allerdings zuvor zu Dokumentationszwecken in dieser Arbeit auf den Angreifer Belzebub kopiert wurden. Über die bestehende SMB-Verbindung wurden die NT Tools von Jesper Lauritsen[70] im Verzeichnis c:\winnt\system32\microsoft\crypto\ntt installiert. So standen die Tool eldump und elsave zur Verfügung. So konnte mit eldump das Eventlog gesichert und über die Administrative Freigabe auf den Angreifer Belzebub kopiert werden. Danach wurde das Eventlog mit dem Tool elsave gelöscht.

6.8. Auswertung

6.8.1. Konsequenzen

Für die Auswertung eines Penetrationstests ist zunächst der entstandene Schaden von Interesse, der dem Anbieter einen Anreiz zur Etablierung von Sicherungsmaßnahmen bieten soll.

Durch den Einsatz der Scanner ist dem Anbieter kein direkter Schaden entstanden. Indirekt hat der Angreifer aber Informationen über das System erlangen können. Dabei können mittels des NSS auch Passwörter erraten werden, die später zu einem Schaden durch Folgeaktionen führen können.

[70] www.ibt.ku.dk/jesper/NTtools

Ein wesentlicher Schaden ist an den Kundendaten entstanden, die geändert worden sind. Dies kann für die Firma Alarm Mayer zu einem Imageschaden und zu einem Vertrauensverlust der Kunden führen, was wiederum zu finanziellen Einbußen führen würde. Diese Schäden gelten auch für den Webspace-Anbieter Arachnolocus, der für die Vorfälle verantwortlich gemacht werden kann. So kann er durch einen solchen Vorfall Kunden verlieren, wodurch seine Existenzgrundlage verloren gehen würde. Auch Haftungsansprüche des Kunden gegenüber Arachnolocus sind denkbar.

Das Argument des Anbieters, er sei für einen Angreifer nicht interessant genug, kann hinsichtlich Malware nicht gelten. So wurde die Web Folder Traversal Schwachstelle von mehreren Würmern wie Solaris/SadMind.worm, W32/CodeRed.worm und W32/Nimda@mm ausgenutzt. Für einen Wurm ist bei der Verbreitung jedes System interessant.

Weiter können die Angriffe jederzeit wiederholt werden, sofern die Schwachstellen nicht behoben sind. Werden nach dem Einbruch nur die ausgenutzten Schwachstellen behoben, kann ein Angreifer auch nach der Behebung die Angriffe wiederholen. So existiert nach dem zweiten Versuch durch den Start des Telnetservers ein weiterer Zugang. Außerdem hat der Angreifer die Zugangsdaten zu der MySQL Datenbank erlangt. Da er mit diesen Zugangsdaten über Schreibrechte verfügt, kann er auch zukünftig die gespeicherten Daten manipulieren. Die Sicherung des Webservers erfordert somit weiter Maßnahmen, die im nächsten Abschnitt behandelt werden.

6.8.2. Gegenmaßnahmen

Der Erfolg sämtlicher im Szenario durchgeführten Angriffe hätte durch eine korrekte Konfiguration des Webservers verhindert werden können. So sind alle durchgeführten Angriffe durch das Einspielen von Patches sowie die richtige Konfiguration des Systems vermeidbar gewesen. Daher wird im folgenden Abschnitt die Konfiguration von Microsoft Windows behandelt. Mit einer solchen Konfiguration kann den in diesem Kapitel durchgeführten Angriffen begegnet werden. Eine genauere Beschreibung der Gegenmaßnahmen erfolgt daher im folgenden Abschnitt.

6.9. Konfiguration des Webservers

Dieser Abschnitt behandelt die Konfiguration eines Webservers, der durch die Internet Information Services 5.0 und Windows 2000 implementiert ist. Im Folgenden werden lediglich Maßnahmen für die richtige Konfiguration eines auf den Microsoft Internet Information Services basierenden Webservers vorgestellt. Dabei ist allerdings zu beachten, dass es sich bei den vorgestellten Maßnahmen nur um eine Auswahl zur Vermeidung der oben beschriebenen Vorfälle handelt. Für weitere Quellen zur Sicherung eines IIS Webservers wird auf Anhang D verwiesen.

Neben der Etablierung von Backup-Prozeduren und der Installation sowie Konfiguration von Anti-Malware Produkten[71], ist die Konfiguration des Betriebssystems entscheidend für die Sicherheit des Systems. Im Folgenden werden Maßnahmen vorgestellt, die in der Literatur

[71] Zu weiteren Informationen über gängige Anti-Malware-Produkte siehe http://agn-www.informatik.uni-hamburg.de/vtc/index.html

unter dem Begriff „Hardening“ behandelt werden. Ein Hardening umfasst minimale Komponenten sowie die Umsetzung des „least-privilege“-Prinzips, das eine minimale Rechtevergabe fordert. Zudem sollte beim Hardening ein Prinzip beachtet werden, das als „never looks like usual“ bezeichnet werden kann. So sollten keine vorgegebenen Verzeichnisse wie „C:\WINNT“ für Windows benutzt werden, um einen Angriff zu erschweren.

Zum Hardening eines Windows-Systems gehören zunächst Einstellungen, die bei einem Windows-System über die Registry und die lokalen Sicherheitseinstellungen[72] vorgenommen werden, wobei bei vielen Einstellungen beide Wege möglich sind. So sind viele der als Gegenmaßnahme beschriebenen Wege nur eine Möglichkeit, um die gewünschte Systemeigenschaft einzustellen.

6.9.1. Windowskonfiguration

Die Internet Information Services sind so eng im Betriebssystem verankert, dass die Sicherheit des Webservers auch von der des Betriebssystems abhängig ist. So kann ein Zugriff auf den einen der IIS-Dienste mit der Windows-Anmeldung gekoppelt sein, so dass die Sicherheit der IIS von der Sicherheit der Benutzerkonten abhängig ist.

Sowohl systemweite als auch benutzerspezifische Einstellungen werden in einer Datenbank, der Registry, gespeichert. Die Registry verfügt dazu über eine Verzeichnisstruktur, wobei die Verzeichnisse Schlüssel heißen. Die direkt mit der Wurzel verbundenen Schlüssel werden als „Stammschlüssel“ bezeichnet, während untergeordnete Schlüssel „Unterschlüssel“ heißen. Für die Stammschlüssel werden in dieser Arbeit wie auch in der verwendeten Literatur Abkürzungen verwendet, die in Tabelle 8 aufgeführt sind. In jedem Schlüssel werden Werte gespeichert, die aus einem Namen, einem Typ und einem Datenteil[73] bestehen.

HKCR	HKEY_CLASSES_ROOT
HKCU	HKEY_CURRENT_USER
HKLM	HKEY_LOCAL_MACHINE
HKU	HKEY_USERS
HKCC	HKEY_CURRENT_CONFIG
HKPD	HKEY_PERFORMANCE_DATA

Tabelle 8: Abkürzungen der Registry-Schlüsselklassen nach [Solomon00:183]

Windows 2000 Betriebssysteme unterstützen die Dateisysteme NTFS in der Version 5 sowie FAT16, FAT32. Dabei unterstützt nur NTFS die Vergabe von Rechten für Dateien und Verzeichnisse, weshalb bei der Installation eines sicheren Windows 2000 Systems ausschließlich NTFS zu verwenden ist. Die richtige Vergabe von Rechten und deren Auswirkungen werden später im Zusammenhang mit den Benutzerkonten behandelt.

[72] Die „Security Policy“ wird in der englischen Version als „Local Security Settings“ bezeichnet und kann mittels des Aufrufes der Datei secpol.msc konfiguriert werden, die Nachfolger des Policy Editors früherer Windows NT Versionen ist.
[73] Der Datenteil wird in deutsprachigen Windowsversionen irreführend ebenfalls als Wert bezeichnet.

Ferner sind die Festplatten entsprechend zu partitionieren. Daten und Programme sollten auf separaten Partitionen gespeichert sein. Standardverzeichnisse wie das Windowsverzeichnis „C:\WINNT“ sollten im Sinne des „never look like usual“-Prinzips umbennant werden.

Einer Null-Session, die zur Enumeration von Freigaben und Benutzernamen verwendet wird, kann eine Registry-Einstellung entgegenwirken. Hierzu ist unter dem Schlüssel HKLM\SYSTEM\CurrentControlSet\Control\Lsa dem DWORD-Wert „RestrictAnonymous“ der Wert 2 zuzuweisen. Nach [SecWin2000:144] bedeutet diese Einstellung „No access without explicit anonymous permissions“, so dass nicht authentifizierte Benutzer bis zur Vergabe expliziter Privilegien keine Rechte besitzen.

Null-Sessions können aber auch auf anderer Ebene entstehen. So greifen manche Dienste, die mit den Privilegien des SYSTEM-Kontos ausgeführt werden, über eine Null-Session auf eine Ressource zu. Diese Eigenschaft eines Dienstes kann durch die Werte „RestrictNullSessAccess“, „NullSessionShares“ und „NullSessionPipes“ im Schlüssel HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters konfiguriert werden, die in [Q122702] näher beschrieben werden. Wenn möglich, sollten aber die Werte „NullSessionShares“ und „NullSessionPipes“ aus der Registry entfernt werden (vgl. [SecWin2000:188]).

Mit der Unterbindung von Null-Sessions kann die Enumeration von Benutzernamen und Freigaben unterbunden werden. In Abbildung 60 ist die Ausgabe von enum nach Anwendung der Einstellungen dargestellt.

```
C:\>enum -U -M -N -S -P -G 192.168.0.3
server: 192.168.0.3
setting up session... fail.
return 5, Zugriff verweigert
```

Abbildung 60: Ausgabe von enum nach Anwendung der Gegenmaßnahmen

Eine weitere Schwachstelle der Standardkonfiguration sind die administrativen Freigaben, mit denen die Festplattenpartitionen sowie das Windowsverzeichnis mittels des SMB-Protokolls über das Netzwerk zugreifbar sind. Die Namen solcher Freigaben enden mit dem $ Symbol und werden nicht im Explorer oder bei der Verwendung des Kommandozeilenbefehls `net view` angezeigt. Sie stellen eine Gefahr dar, da sie einem Angreifer einen Einstiegspunkt bieten und zudem häufig nicht bekannt sind.

Administrative Freigaben können bei Microsoft Windows 2000 im Computermanagement (vgl. Abbildung 61) entfernt werden. Bei Benutzung der Voreinstellungen werden die administrativen Freigaben bei Neustart des Server-Dienstes, der bei jedem Systemstart gestartet wird, erneut erzeugt. Um dies zu verhindern, sind zunächst die im Schlüssel HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\ eventuell vorhandenen Werte „AutoShareServer“ und „AutoShareWks“ vom Typ DWORD zu entfernen und im Schlüssel HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters\ mit dem Wert 0 erneut zu erzeugen.

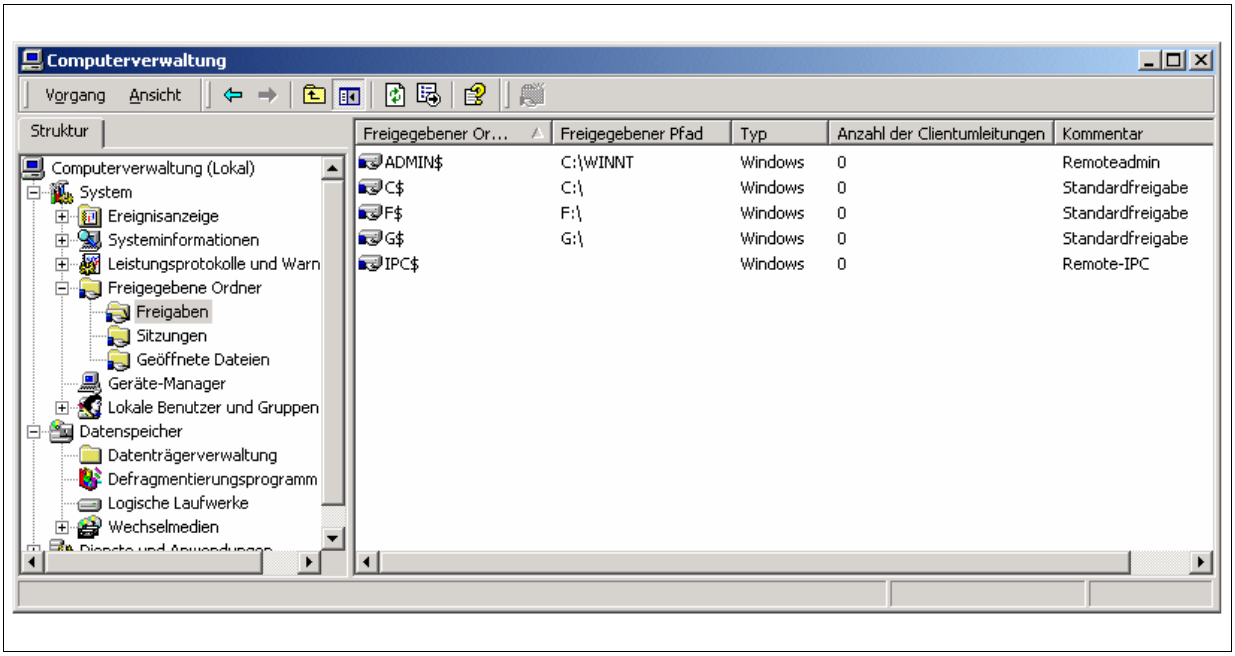


Abbildung 61: Verwaltung der Freigaben in Windows 2000

Für den Fall, dass das System über Terminal Services administriert wird, kann ein Benutzer auf diesem Weg das System ohne Anmeldung herunterfahren. Um dies zu verhindern, ist in den Lokalen Sicherheitseinstellungen der Eintrag „Sicherheitseinstellungen\Lokale Richtlinien\Sicherheitsoptionen\Herunterfahren: Herunterfahren des Systems ohne Anmeldung zulassen“ zu deaktivieren. Ebenso kann ein lokaler Angreifer oder ein solcher, der über die Terminal Services auf das System zugreifen kann, Benutzerkonten erfahren, wenn im Anmeldedialog der letzte Anmeldename gezeigt wird. Um dies zu verhindern, ist im selben Verzeichnis der Lokalen Sicherheitseinstellungen die Option „Interaktive Anmeldung: Letzten Benutzernamen nicht anzeigen“ zu aktivieren.

In diesem Szenario wurden bisher keine Denail-of-Service Angriffe betrachtet. Um diesen bereits jetzt vorzubeugen, werden die in [SecWin2000:175] vorgeschlagenen Werte für den Schlüssel HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\ übernommen. Die Werte sind in Tabelle 9 dargestellt.

Zusätzlich sollten nach [SecWin2000:175] für Anwendungen, welche die WinSock benutzen, folgende Werte des Schlüssels HKLM\System\CurrentControlSet\Services\AFD\Parameters\ übernommen werden, die in Tabelle 10 aufgeführt sind.

Name	Typ	Daten
EnableICMPRedirect	DWORD	0
SynAttackProtect	DWORD	2
EnableDeadGWDetect	DWORD	0
EnablePMTUDiscovery	DWORD	0
KeepAliveTime	DWORD	300 000
DisableIPSourceRouting	DWORD	2
TcpMaxConnectResponseRetransmissions	DWORD	2
TcpMaxDataRetransmissions	DWORD	3
PerformRouterDiscovery	DWORD	0
TCPMaxPortsExhausted	DWORD	5

Tabelle 9: TCP/IP-Einstellungen zum Eindämmen von DoS Angriffen

Name	Type	Daten
DynamicBacklogGrowthDelta	DWORD	10
EnableDynamicBacklog	DWORD	1
MinimumDynamicBacklog	DWORD	20
MaximumDynamicBacklog	DWORD	20000

Tabelle 10: Winsock-Einstellungen zum Eindämmen von DoS Angriffen

Neben den Systemeinstellungen gehören zu einer sicheren Konfiguration auch Patches, die Wartung der Benutzerkonten, sowie die Minimierung aktiver Dienste und vorhandener Dateien. Diese Elemente eines gehärteten Systems werden im Folgenden näher betrachtet.

6.9.2. Patches

Ein Patch ist die englische Bezeichnung für einen Flicken. Hierbei handelt es sich um kleine Updates, die Fehler einer Software korrigieren. Eine solche Korrektur, die meist einzelne Dateien des Systems austauscht, werden von Microsoft als Hotfix bezeichnet (vgl. [Q296861]). Ein Fehler in der Software gefährdet die Sicherheit eines Systems, wodurch es sich bei den Fehlern um Schwachstellen handelt. So ist es notwendig, alle verfügbaren Hotfixes einzuspielen, um die Schwachstellen zu beheben.

Interessant ist, dass manche Patches die Sicherheitslücke nicht beheben (vgl. [Zavdi01]). Zudem öffnen manche Patches neue Sicherheitslücken. So musste auf dem Webserver für den im Abschnitt 6.7.2 beschriebenen Angriff mindestens Service Pack 2 installiert sein, damit der Angriff erfolgreich war.

Ein Service Pack ist dabei eine Sammlung von Patches. Dabei soll ein Service Pack alle zum Zeitpunkt des Erscheinens verfügbaren Patches vereinen. So steht in [SecWin2000:276]: „Service packs are also cumulative — each new service pack contains all of the fixes in previous service packs, as well as any new fixes and system modifications that have been recommended since. You do not need to install a previous service pack before you install the latest one. Service packs may also contain a limited number of customer requested design changes or features, and because they are broadly distributed, they are tested heavily“. Trotz der spezifizierten Eigenschaft des Service Packs, beinhaltet es nicht alle vorangegangenen

Patches vollständig. So ließ sich der in Abschnitt 6.7.1 beschriebene Web Folder Traversal Angriff nach Einspielen des im Bulletin MS00-078 referenzierten Patches nicht mehr durchführen. Nach der Installation des im Mai 2001 erschienenen Service Packs 2 konnte dieser Angriff wieder erfolgreich durchgeführt werden. Jedoch war die Sicherheit eines Benutzers, der alle Patches installiert hat, zu keinem Zeitpunkt gefährdet, da bei Erscheinen des Service Pack 2 bereits ein pre-SP3 Patch verfügbar war, der die Schwachstelle endgültig behebt. Auf Grund dieser Beobachtungen kann jedoch nicht ausgeschlossen werden, dass in einem heutigen Servicepack eine bereits behobene Schwachstelle wieder durch einen altbekannten Angriff ausgenutzt werden kann.

Für das Einspielen der Patches ist ein Patch Management zuständig, das aus folgenden Komponenten besteht:

- Planungskomponente: Wie erfahre ich von neuen Patches und wie werden sie bezogen?
- Organisationskomponente: Wie werden neue Patches eingespielt?
- Kontrollkomponente: Wie wird die Installation des Patches kontrolliert?

In der Planungskomponente wird beispielsweise entschieden, ob ein Patch über das bei Windows beigegebene Windows-Update oder manuell bezogen wird[74]. Alternativ kann entschieden werden, sich mittels des Microsoft Notify Services[75] über aktuelle Patches zu informieren und diese manuell aus dem Internet herunterzuladen, wobei die Integrität der Patches vorher manuell überprüft wird.

In der Organisationskomponente muss bedacht werden, dass das Einspielen eines Patches einen Neustart erfordert. Hierbei muss geklärt werden, ob ein System während des Einspielens des Patches vom Netz genommen wird (vgl. SecWin2000:275ff.]). Zudem ist zu entscheiden, ob ein Patch vorher auf einem separaten System getestet wird. Auch Probleme bei der Hintereinanderausführung von Patches müssen bedacht werden (vgl. [Q296861]).

Um die Installation der Patche zu kontrollieren, kann nach Evidenzen der korrekten Installation gesucht werden. Dazu bieten sich Tools wie QFEcheck (vgl. [Q282784]), der Microsoft Baseline Security Analyser[76] oder der in Abschnitt 6.6.1 behandelte NSS an. Eine andere Möglichkeit der Kontrolle ist die erneute Durchführung eines Penetrationstests. Die Durchführung eines Penetrationstests im Zusammenhang mit eingespielten Patchen wird in Kapitel 9.1 näher besprochen werden.

Wären in diesem Szenario zu Beginn der Versuche im Juni 2003 alle bis dahin verfügbaren Patche installiert gewesen, hätte der fiktive Angreifer nicht auf dem beschriebenen Wege eindringen können.

[74] Die Patches werden in den Microsoft Security Bulletins veröffentlicht, die jeden Mittwoch aktualisiert werden.
[75] www.microsoft.com/technet/security/bulletin/notify.asp
[76] www.microsoft.com/technet/security/tools/Tools/mbsahome.asp

6.9.3. Rechtevergabe

Die Rechtevergabe ist abhängig von Benutzerkonten, denen Privilegien zugeordnet sind. Anhand dieser Privilegien entscheidet das Betriebssystem, ob und wie ein Benutzer auf Dateien oder Verzeichnisse zugreifen und welche administrativen Aufgaben ein Benutzer durchführen kann. Da Programme und Dienste im Kontext eines Benutzers laufen und über die Privilegien dieses Benutzers verfügen, sind Benutzerkonten selbst dann von großer Bedeutung, wenn ein Benutzer sich nicht in ein System einloggen kann.

Zunächst sind die zwei vordefinierten Konten „Administrator“ und „Gast“ zu beachten. Sie sind auf jedem System vorhanden und können nicht gelöscht werden. Daher kann ein Angreifer versuchen, die Privilegien der Konten zu erlangen, indem er beispielsweise die Passwörter rät. Da das Gastkonto nicht benötigt wird, ist es zu deaktivieren.

In diesem Zusammenhang sind auch die Passwörter der Benutzerkonten zu sichern. Dabei sollten möglichst 7 oder 14 Zeichen lange Passwörter bestehend aus großen und kleinen Buchstaben sowie Ziffern und Sonderzeichen verwendet werden. Die Passwörter sollten in kurzen und regelmäßigen Abständen ausgetauscht werden und dürfen sich nicht wiederholen. Richtlinien für Passwörter können in den lokalen Sicherheitseinstellungen konfiguriert werden. Weitere Informationen zur Konfiguration von Passwörtern sind in [SecWin2000:120ff.] enthalten.

Auch ein anonymer Besucher einer Homepage verfügt auf dem jeweiligen Server über ein Benutzerkonto. Dieses ist im Falle des IIS das Internetgastkonto IUSR_<computername>, das auf dem Rechner www3 dementsprechend IUSR_www3 heißt. Einem solch spezialisierten Benutzerkonto sollte nach dem least-privilege Prinzip (vgl. [Chapman00:59]) nur nie nötigsten Rechte gegeben werden. Beim Internetgastkonto darf daher nach [Walker02:9] das Passwort nicht vom Benutzer selbst geändert werden und läuft nie ab. Zudem sollte der Besitzer nur Rechte zum Lesen von Dateien haben.

Neben dem Internetgastkonto wird bei Installation auch das IIS-Prozesskonto erzeugt. Es ist für die Ausführung von Anwendungen und Skripten vorgesehen, damit ein Fehler in der Anwendung nicht den IIS-Webserver-Prozess beeinträchtigt (vgl. [Q236855]). Daher werden solche Anwendungen als „out-of-process“ bezeichnet. Das IIS-Prozesskonto hat den Namen IWAM_<computername> und somit in dem hier vorliegenden Szenario den Namen IWAM_www3. Auch dieses Konto ist zunächst Mitglied der Gruppe Gäste, sollte aber keiner Gruppe angehören. Zudem sollte das Internet Prozesskonto nur Rechte zum Ausführen von Dateien haben.

Wird der IIS auf einer Server Version von Windows 2000 installiert, so kann der Server mehrere Webseiten beherbergen. In diesem Fall ist es sinnvoll, die Benutzerkonten IUSR_www3 und IWAM_www3 zu entfernen. Für jede Webseite sollte ein eigener Benutzer jeweils für Besucher und für die Ausführung von Anwendungen geschaffen werden. So kann ein Angreifer bei einem erfolgreichen Angriff auf eine Webseite nicht auf Daten einer anderen Webseite zugreifen.

Auf die Verzeichnisstruktur ist das least-privilege Prinzip zu übertragen. So sollten Internetgastkonto und IIS-Prozesskonto nur auf das Verzeichnis zugreifen dürfen, in dem die Homepage gespeichert ist. Bei Nutzung der Voreinstellungen sind Internetgastkonto sowie

IIS Prozesskonto Mitglied der Gruppe Gäste, womit die Möglichkeit besteht, mit den Privilegien des Internetgastkontos auch auf andere Verzeichnisse zugreifen zu können. Daher sind beide Benutzerkonten aus der Gruppe Gäste zu entfernen. Die Benutzer sollten möglichst keiner Gruppe angehören, um u. a. zu verhindern, dass ausführbare Dateien auf Verzeichnisse mit statischem Inhalt zugreifen können.

Hintergrund des Erfolgs des Web Folder Traversal Angriffes ist nicht nur ein Implementationsfehler in der Software. Auch Fehler der Konfiguration der Systeme ermöglichen diesen Angriff. So hat bei Nutzung der Voreinstellungen die Benutzergruppe „Jeder“, die alle Benutzerkonten vereinigt und weder modifiziert noch gelöscht werden kann, zu viele Rechte. Jeder Benutzer kann im Hauptverzeichnis schreiben sowie das Windowsverzeichnis lesen und Dateien ausführen. Zudem hat bei Nutzung der Voreinstellungen auch die Gruppe „Benutzer“ Rechte zum Lesen des Windows-verzeichnisses. Zu dieser Gruppe gehören auch alle authentifizierten Benutzer, zu denen auch jeder anonyme Besucher einer Website gehört, da er sich beim Besuch der Webseite mit dem Internetgastkonto anmeldet.

Die Gruppe „Jeder“ sollte nur Lese- und Schreibrechte[77] auf das temporäre Verzeichnis bekommen. Bei allen anderen Verzeichnissen sollten beide Gruppen keine Rechte bekommen. Um dies zu ermöglichen, sollte in den Verzeichniseigenschaften im Reiter Sicherheit auf „Erweitert“ geklickt werden. Durch das Setzen des in Abbildung 62 gesetzten Hakens werden die Rechte für alle Unterverzeichnisse übernommen. Bei der Änderung der Rechte für die Gruppe „Jeder“ darf dieser Gruppe nichts verboten werden, da von solchen Verboten auch der Administrator und der von Diensten verwendete Benutzer „System“ betroffen sind.

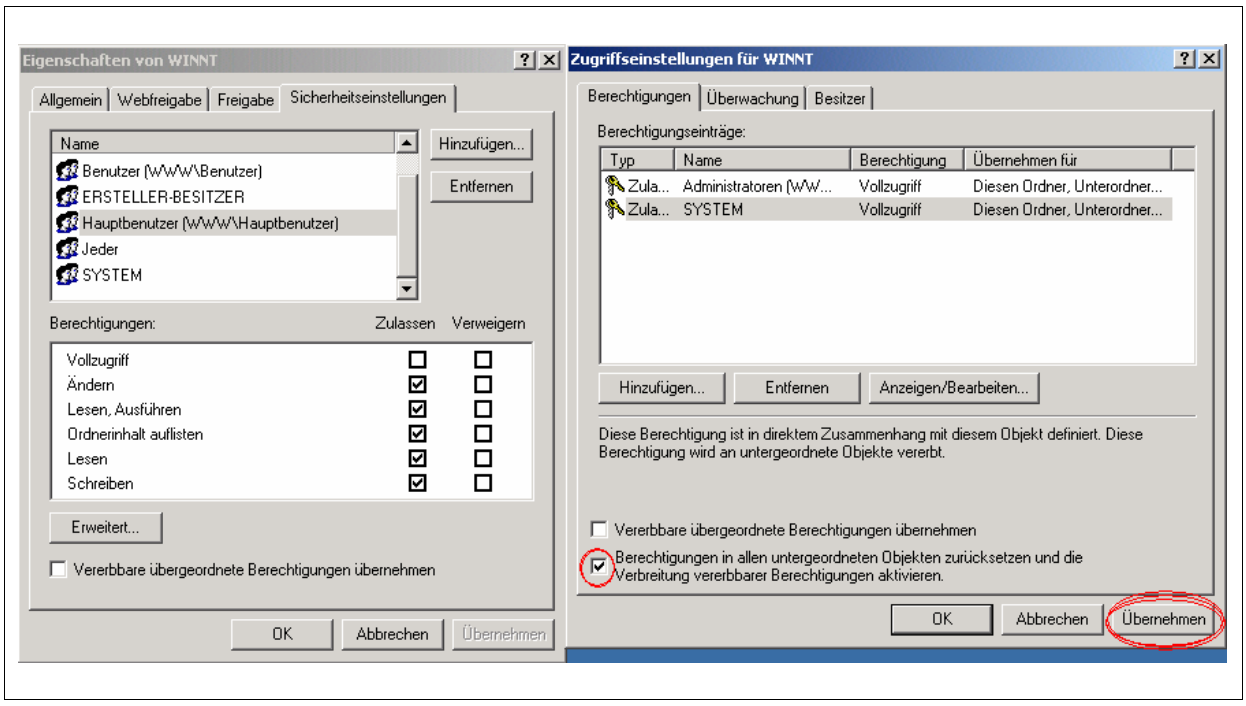


Abbildung 62: Effektive Rechtevergabe bei Windows 2000

[77] Bei der Installation von Software wird auch ein ausführendes Recht benötigt.

Auf der rechten Seite der Abbildung 62 ist zu erkennen, dass sämtliche Administratoren Zugriff auf das Verzeichnis haben. Dies schließt auch andere Administratoren der Domain ein und sollte auf den lokalen Administrator beschränkt werden (vgl. [Walker02:17]). Nach Walker (vgl. [Walker02:17]) genügen diese Einstellungen: „The default permission settings for this file are System and Administrators Full Access. Limiting the access to System and local Administrators provides good security […]“ [Walker02:17]. Allerdings haben Administrator und System volle Rechte und damit eine Übermacht. Erlangt ein Angreifer die Privilegien eines dieser beiden Benutzerkonten, so hat er vollen Zugriff auf das System. Die Auswirkungen konnten bei der Durchführung der Versuche des Abschnitts 6.7.2 beobachtet werden, bei denen der Angreifer durch einen Buffer Overflow Zugriff mit den Privilegien des System-Kontos erlangte.

Daher sollten auch die Rechte des Administrators weitestgehend beschränkt werden und Dienste sollten nicht mit den SYSTEM-Privilegien betrieben werden. Diese Aufgabe wird auf Grund ihrer Komplexität an dieser Stelle nicht behandelt werden. So kann es passieren, dass die für eine Anmeldung notwendigen Rechte nicht mehr bestehen und eine Neuinstallation notwendig wird. Jedoch sollten SYSTEM und Administrator keine Rechte auf die Verzeichnisse haben, welche die Webseite enthalten, um so einem Defacement vorzubeugen. Soll ein Mitarbeiter auf Seiten des Webspace-Anbieters Zugriff auf die Inhalte haben, so sollte dafür ein spezieller Benutzer eingerichtet werden.

6.9.4. Minimale Dienste

Ein Dienst stellt Funktionalitäten bereit, die von Applikationen genutzt werden können und „nicht an einen interaktiven Benutzer gebunden sind“ [Solomon00:199]. Mögliche Fehler in Design, Implementation oder Konfiguration des Dienstes führen zu Schwachstellen, welche wiederum zu Risiken hinsichtlich der IT-Sicherheit führen. Um die Risiken gemäß der IT-Sicherheit zu minimieren, sind alle nicht benötigten Dienste aus dem System zu entfernen, um so ein unnötiges Risiko zu vermeiden.

Bevor ein Dienst deaktiviert wird, muss überprüft werden, ob er wirklich nicht benötigt wird. Ist er unverzichtbar,

- so sollte der Zugriff auf den Dienst gefiltert,
- die Rechte des Dienstes möglichst weit beschränkt und
- wenn möglich die Funktionalität des Dienstes begrenzt werden.

Nicht benötigte Dienste können zum Teil durch die Deinstallation der Software, die diesen Dienst implementiert, entfernt werden. Solche Möglichkeit ist zum Beispiel bei den Internet Infomation Services gegeben, die durch eine Komponente von Microsoft Windows realisiert sind.

Manche Dienste können aber nicht deinstalliert werden. Solche Dienste sind nicht nur zu beenden, sondern zu deaktivieren. So kann der Dienst nicht mit `net start` gestartet werden, was in 6.7.2 mit dem Dienst Telnet getan wurde. Abbildung 63 stellt die Wirkung der Deaktivierung eines Dienstes dar.

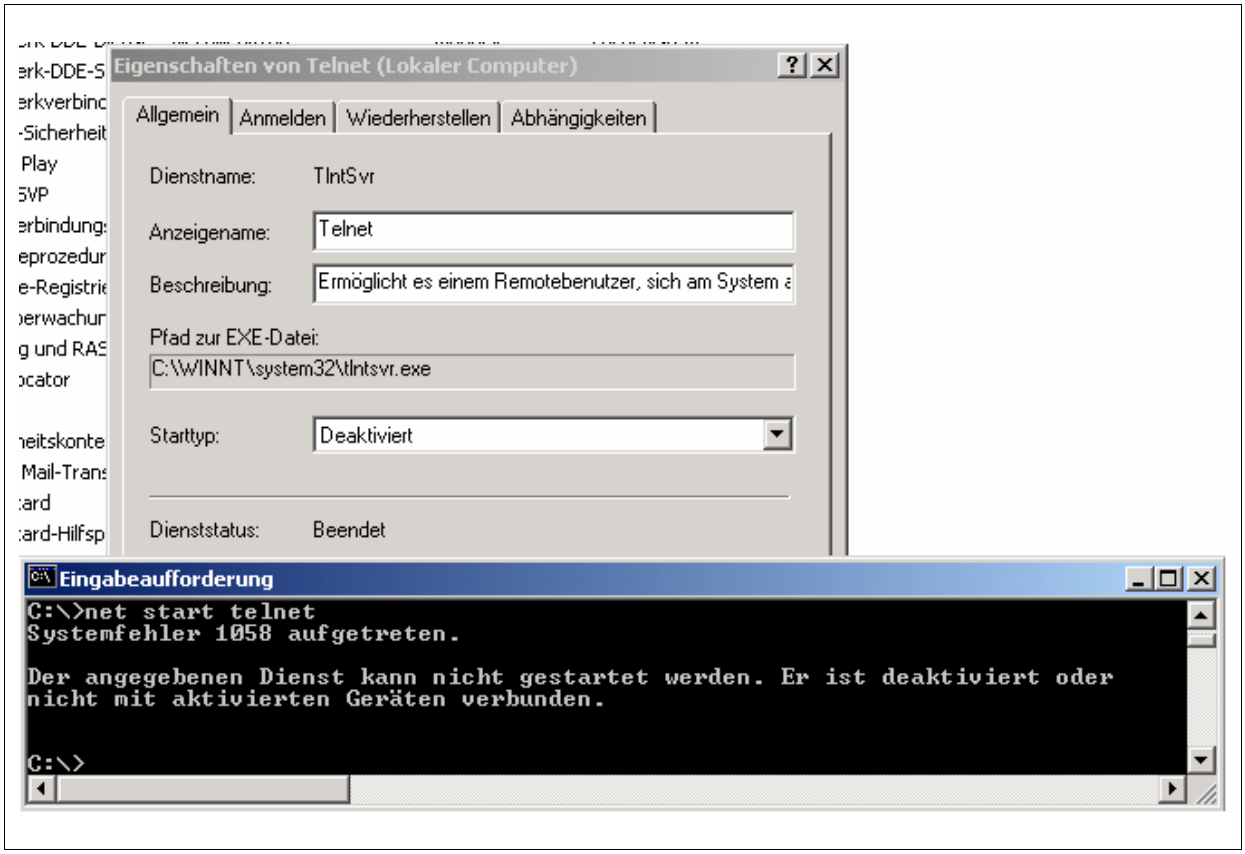


Abbildung 63: net start bei deaktiviertem Dienst

Um den Angreifer in der Reconnaissance zu verwirren, kann der Anzeigename der Dienste verändert werden, da diese bei den in Kapitel 5.3.2. und 6.4 gezeigten Methoden ausgegeben werden. Jedoch kann der Dienstname selbst nicht verändert werden, so dass ein Angreifer trotzdem den Status eines Dienstes abfragen kann.

Grundsätzlich ist ein Webserver auf einem eigenen Rechner zu installieren, damit ein Angriff auf den Webserver keine anderen Funktionen des Rechners in Mitleidenschaft ziehen kann und umgekehrt.

Der SNMP-Dienst, der das Simple Network Management Protokoll unterstützt, ist in einer Standardinstallation von Windows 2000 nicht enthalten. Wird er nicht benötigt, so ist er zu deinstallieren bzw. nicht zu installieren. Jedoch benötigen viele Tools zur Überwachung von Systemen wie beispielsweise MOM diesen Dienst. Auch Arachnoclus plant den Einsatz solcher Tools und hat den Dienst proaktiv installiert.

Wird der SNMP-Dienst benötigt, so ist den durch ihn entstehenden Risiken entgegenzuwirken. Dazu ist der Zugangspunkt zu diesem Dienst auf Port 161/udp durch eine Firewall zu filtern, so dass nur die überwachenden Rechner auf den Dienst zugreifen können. Zudem werden Informationen mit dem Simple Network Management Protocol im Klartext übertragen. Auch mangelt es dem Protokoll an Authentizität. Die einzige Authentifizierung ist die Kenntnis des Comunity Strings. In einer Standardkonfiguration existiert der Community String Public, über den ein Angreifer in der Reconnaissance Informationen über das System auslesen kann. Ist ein Schreibzugriff erlaubt, kann ein Angreifer sogar Einstellungen eines

Systems manipulieren. Die Voreinstellungen von Windows 2000 lassen nach [SecWin2000:172] allerdings keinen Schreibzugriff zu.

Zur Behebung dieser Schwachstellen ist zunächst der Community String Public umzubenennen. Dies erfolgt in der Registry durch das Ändern des Wertes Public im Schlüssel HKLM\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ValidCommunities. Zudem empfiehlt sich der Einsatz einer VPN mittels IPSec, das der Autor in [Heinzel03] genauer beschreibt. Dabei kann mittels des Authentication Headers die fehlende Authentifizierung implementiert werden. Mit der Encapsulation Security Payload kann der Netzwerkverkehr verschlüsselt werden.

Um die Benutzung von SMB über das Netzwerk zu unterbinden, ist die wirksamste Methode, die Bindung der Netzwerkkarten an diesen Dienst aufzuheben. Dazu muss für jede in den Eigenschaften der Netzwerkumgebung angezeigten Verbindung der „Client- für Microsoft-Netzwerke“ und die „Datei- und Druckerfreigabe für Microsoft Netzwerke“ deinstalliert werden, was in den Eigenschaften der Verbindungen durchgeführt wird. Wird dies für sämtliche Verbindungen durchgeführt, können auch die Dienste „TCP/IP-NetBIOS-Hilfe“ (LmHosts) und „Server“ (lanmanserver) deaktiviert werden, um die SMB-Funktionalität komplett zu deaktivieren.

Grundsätzlich sollten sämtliche Dienste, von denen nicht klar ist, ob sie zum Betrieb des Servers notwendig sind, zunächst deaktiviert werden. Ist dann eine gewünschte Funktion nicht mehr verfügbar, sollten die Dienste einzeln aktiviert und bei Misserfolg wieder deaktiviert werden.

6.9.5. Minimale Dateien

Neben der Deinstallation unbenötigter Dienste ist eine goldene Regel des Hardenings, ausführbare Dateien zu sichern. Dies bedeutet möglichst viele ausführbare Dateien aus dem System zu entfernen und benötigte Dateien umzubenennen oder im Zugriff zu beschränken.

Dabei geht es um Dateien, die als Kommandos von der Eingabeaufforderung aus verwendet werden können. Dies sind vor allem die in Tabelle 11 aufgeführten Programme. Solche Programme kann ein Angreifer verwenden, um weitere Informationen über das System zu erlangen oder das System zu manipulieren. Dabei sind vor allem die Konsolenwerkzeuge interessant, die ein Angreifer über eine Shell steuern kann. Die Möglichkeiten der Steuerung eines Systems über eine Shell sind bei dem in Abschnitt 6.7.2 durchgeführten Versuch dargestellt worden.

- arp.exe - append.exe - at.exe - attrib.exe - cacls.exe - chkdsk.exe - chkntfs.exe - comp.exe - cscript.exe - debug.exe - discperf.exe - edlin.exe - esentutl.exe - fc.exe - find.exe - finger.exe - forcedos.exe - ftp.exe - ftpqfe.exe - help.exe - hostname.exe - iexpress.exe - iisreset.exe - jview.exe - label.exe - lpq.exe - lpr.exe - makecab.exe	- mem.exe - migpwd.exe - mountvol.exe - mshta.exe - nbtstat.exe - nddeapir.exe - netstat.exe - net.exe - net1.exe - netsh.exe - nslookup.exe - ping.exe - print.exe - rasadmin.exe - rasautou.exe - rasdial.exe - rasphone.exe - replace.exe - rexec.exe - route.exe - rsh.exe - tftp - tlntadmn.exe - tlntsvr.exe - tlntsvr.exe - tracert.exe - xcopy

Tabelle 11: Einzuschränkende ausführbare Programme [78]

Es sollten möglichst alle Programme gelöscht werden, die nicht benötigt werden, oder deren Funktion auch über die grafische Benutzeroberfläche zur Verfügung steht. Manche Funktionen, wie beispielsweise arp.exe, können durch Programme dritter Anbieter ersetzt werden, die eine grafische Konfiguration bieten. Sollten dennoch Programme für die Kommandozeilen benötigt werden, so sind weitere Sicherheitsmaßnahmen zu treffen.

Zunächst sind die zugehörigen Dateien aus dem Verzeichnis %systemroot%\system32 in ein separates Verzeichnis zu verschieben. Auf dieses Verzeichnis sollte nur ein administrativer Benutzer lesenden und ausführenden Zugriff haben. Dem Benutzerkonto SYSTEM darf kein Zugriff erlaubt sein, da die Dienste in diesem Kontext ausgeführt werden und ein Einbruch über die Ausnutzung einer Schwachstelle in einem Dienst einem Angreifer eben diese Privilegien verschafft. Zu jedem Benutzerkonto existiert dabei ein numerischer Bezeichner namens Security ID. Sie endet für den Administrator auch bei Umbenennung mit dem

[78] Siehe auch http://www.horseplay.demon.co.uk/windows_2000_commands.html

Wert 500 (vgl. [Solomon00:422f.]). Daher sollte zur Ausführung der Kommandozeile und der zugehörigen Werkzeuge ein exklusiver Benutzer geschaffen werden. Weiter sollte die Umgebungsvariable path nicht auf das separate Verzeichnis verweisen. Für noch mehr Sicherheit kann auch eine Verschleierung des Verzeichnisses sorgen, indem alle Konsolenwerkzeuge unsinnige Namen erhalten.

Auch für Programme, die einem Resource Kit oder einem SDK[79] entstammen, sind solche Maßnahmen zu treffen. So könnte mit `xcacls` die Rechtestruktur manipuliert und mit `ipsecpol` die Filterung des Netzverkehrs aufgehoben werden. Zudem sollten alle unbenötigten Windows-Komponenten entfernt werden.

6.9.6. Security Templates

Die meisten der hier vorgenommenen Einstellungen müssen nicht von Hand vorgenommen werden, sondern können in einem so genannten Security Template zusammengefasst werden. Dies hat den Vorteil, dass diese Einstellungen automatisch auf verschiedene Systeme übertragen werden können und auch bei einer Neuinstallation zur Verfügung stehen.

Für derartige Aufgaben sind in der Microsoft Management Console, kurz MMC, die Snap-ins „Sicherheitskonfiguration und -analyse“ sowie „Sicherheitsvorlagen“ enthalten. Mit dem Snap-In „Sicherheitsvorlagen“ können Security Templates definiert werden, deren Konfigurationsmöglichkeiten in Tabelle 12 dargestellt sind. Templates werden im Verzeichnis %systemroot%\system32\Security\Templates gespeichert, wobei Windows einige Beispiele enthält. Unter [Q316347] stellt Microsoft ein Sicherheitstemplate für Webserver bereit.

Mit dem Snap-in „Sicherheitskonfiguration und -analyse“ können die Security Templates eingespielt und überwacht werden. Auch das im Resourcekit enthaltene Tool secedit kann zum Einspielen eines Security Templates verwendet werden. Für weitere Informationen über die Security Templates wird auf [Davis01:2f.] und [Q309689] verwiesen.

Kontorichtlinien	Einstellungen zu Benutzerkonten und Passwörtern wie Kennwortchronik oder minimale Kennwortlänge
Lokale Richtlinien	Einstellen von Überwachung, Benutzerrechten sowie Sicherheitsoptionen
Ereignisprotokoll	Einstellungen zum Ereignisprotokoll, wie z.B. maximale Größe
Eingeschänkte Gruppen	Verwaltung der Mitglieder von lokalen Gruppen
Systemdienste	Starteinstellung der Dienste
Registrierung	Berechtigungen und Überwachung der Registry
Dateisystem	Berechtigungen und Überwachung des Dateisystems

Tabelle 12: Konfigurationsmöglichkeiten eines Sicherheitstemplates

[79] So entstammt das Tool snmputil dem Core-SDK. Es musste lediglich kompiliert werden.

6.9.7. IIS Konfiguration

Neben der Konfiguration des Betriebssystems ist auch die Konfiguration der Internet Information Services von großer Bedeutung für einen Webserver. Da bei der Penetration die Unsicherheiten auf den Webserver beschränkt worden sind, wird die Sicherung der IIS auf die Web-Komponente beschränkt. Die Konfiguration der IIS erfolgt über den Internetdienste-Manager[80], der ein Snap-In der Microsoft Management Console und auch in der Computerverwaltung[81] enthalten ist.

Bei den mit einer Servervariante von Windows 2000 gelieferten IIS können mehrere Webseiten konfiguriert werden. Neben Einstellungen für eine bestimmte Webseite, können durch die Haupteigenschaften auch Vorgaben für alle Webseiten gegeben werden. Sie werden automatisch von allen Webseiten übernommen. Daher sind die Haupteigenschaften zuerst zu konfigurieren, wobei die Vorgaben sehr restriktiv sein sollten.

Manche Einstellungen im Betriebssystem führen zu der Notwendigkeit, Einstellungen des IIS zu ändern. So muss für den Fall, dass das Internetgastkonto umbenannt wird, diese Änderung auch im IIS eingestellt werden. Dies erfolgt im Reiter „Verzeichnissicherheit“ in der Bearbeitung von „Steuerung des anonymen Zugriffs und der Authentifizierung“. Wird dort die „Anonyme Anmeldung“ gewählt, so kann das dafür verwendete Konto bearbeitet werden. Gleichzeitig sollte hier die „Integrierte Windows-Authentifizierung“ abgestellt werden, sofern sie nicht benötigt wird. So fordert ein ungepatchter Server mit korrekten Verzeichnisrechten bei einem Web Folder Traversal Angriff den Client auf, sich zu authentifizieren. Dies kann ein Angreifer ebenfalls ausnutzen, um Benutzernamen und Passwörter zu erraten. Nach der Deaktivierung der Windows-Authentifizierung erfolgt bei einem versuchten Zugriff auf ein Verzeichnis wie das Windows Verzeichnis die Fehlermeldung „Sie sind nicht berechtigt, diese Seite anzuzeigen“. Diese in Abbildung 64 dargestellten Einstellung sollte für alle Webseiten gelten, die einen anonymen Zugriff erlauben.

Ist eine Authentifizierung erforderlich, so sollte keine „Standardauthentifzierung“ gewählt werden. Sie implementiert das Basic-Authentication-Vefahren (vgl. [Walker02:35]), bei dem Benutzername und Passwort im Klartext gesendet und somit abgehört werden können.

[80] %SystemRoot%\system32\inetsrv\iis.msc
[81] compmgmt.msc

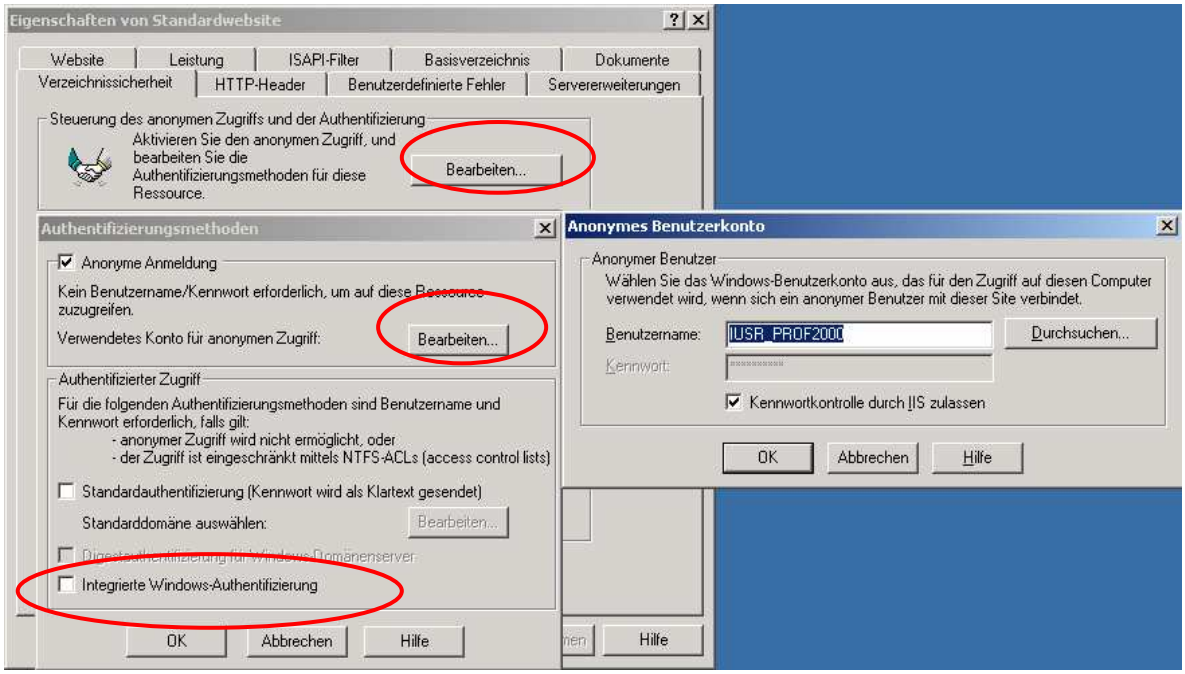


Abbildung 64: Umbenennung des Internetgastkontos und Deaktivieren der Windows-Authentifizierung

Bei der Sicherung eines IIS-Webservers müssen auch die ISAPI (Internet Server Application Programmable Interfaces) beachtet werden, da Fehler in diesen Erweiterungen zu weiteren Angriffsmöglichkeiten führen. Hierbei ist zwischen ISAPI-Erweiterungen[82] und ISAPI-Filtern zu unterscheiden (vgl. [IIS Insider 05/02]). Erweiterungen sind Anwendungen oder Skripte, die direkt über die URL aufgerufen werden können und eine spezifische Ausführ-Berechtigung sowohl im IIS als auch im Dateisystem benötigen. Filter hingegen, können sowohl den eingehenden als auch den ausgehenden Datenstrom des IIS modifizieren.

Auf Grund unterschiedlicher Schwachstellen, die bereits in verschiedenen ISAPI-Erweiterungen erkannt worden sind[83], sollten alle ISAPI-Erweiterungen entfernt werden, sofern sie nicht explizit benötigt werden. Auch auf ISAPI-Filter ist diese Regel anzuwenden. Die Funktionen der ISAPI-Filter einer Standardinstallation werden in Tabelle 13 beschrieben. In den Haupteigenschaften sollten keine der unten genannten ISAPI-Filter eingestellt sein. Soweit möglich, sollten auch alle ISAPI-Erweiterungen entfernt werden.

Sspfilt	Unterstützung für SSL
Compression	Möglichkeit der Datenkompression
md5filt	Unterstützung für MD5-Authentikation
fpexedll.dll	Kompatibilität mit FrontPage 97

Tabelle 13: Funktionen der voreingestellten ISAPI-Filter nach [IIS Insider 07/02]

[82] In englischsprachigen Versionen „ISAPI extensions“ genannt
[83] Beispiele: CAN-2000-0071, CVE-2001-0004, CVE-2001-0241 oder CAN-2002-0247

Für die als ISAPI-Erweiterung ausgeführten Anwendungen, die serverseitig Code ausführen, bietet der IIS einen Anwendungsschutz, der aus den drei Stufen „Niedrig (IIS-Prozess)“, „Mittel (zusammengefasst)“ und „Hoch (isoliert)“ besteht (vgl. Abbildung 65).

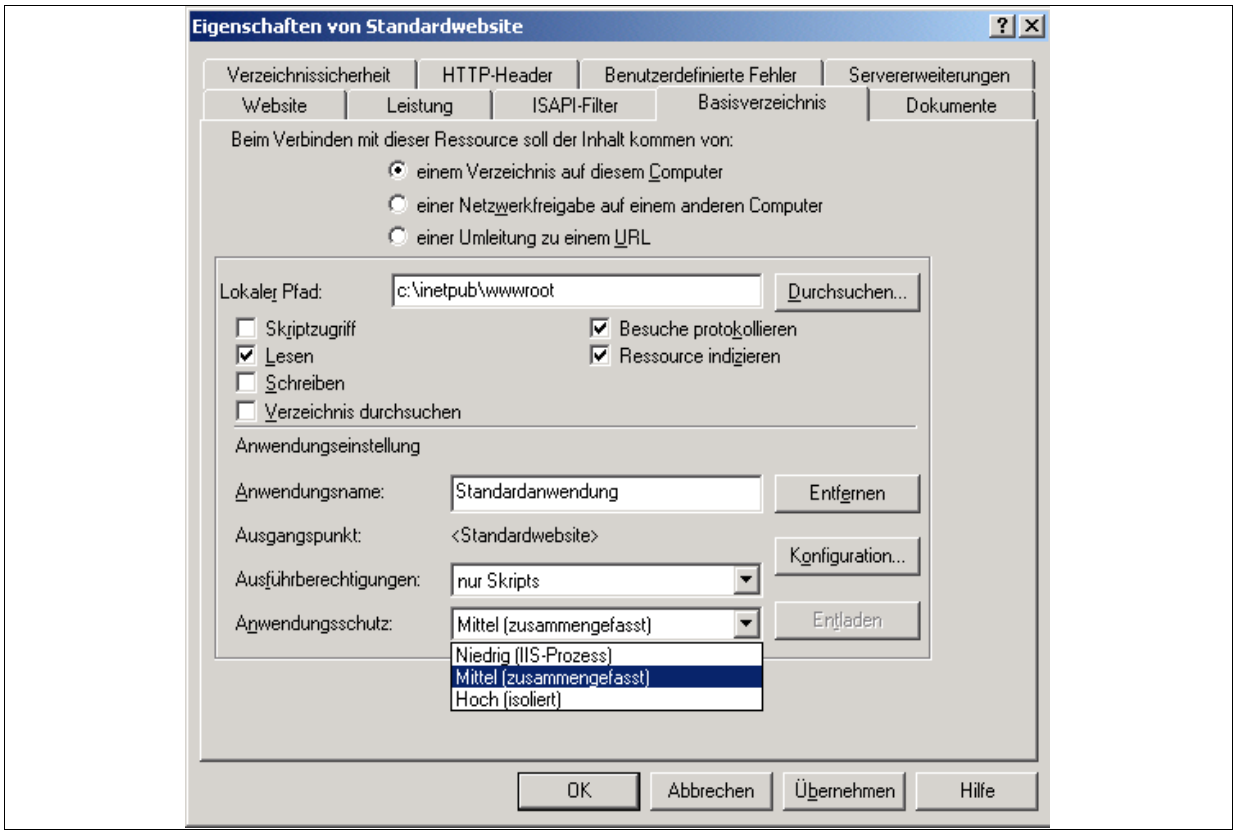


Abbildung 65: Stufen des Anwendungsschutzes im IIS

Bei niedrigem Anwendungsschutz wird eine Anwendung im Webserverprozess (inetinfo.exe) ausgeführt, weshalb diese Einstellung in vielen Artikeln der Microsoft Knowledgebase auch als „in-process“ bezeichnet wird. Hierbei kann allerdings durch eine Schwachstelle in der Anwendung der Webserverprozess kompromittiert werden. Die beiden anderen Anwendungsschutzstufen werden dementsprechend als „out-of-process“ bezeichnet. Hierbei werden die Anwendungen in einem von dllhost.exe erzeugten Prozess ausgeführt. Bei mittlerem Anwendungsschutz, der als „pooled out-of-process“ bezeichnet wird, werden alle Anwendungen in einer Instanz von dllhost.exe zusammengefasst (engl.: pooled). Bei hohem Anwendungsschutz wird für jede Anwendung ein eigener, isolierter Prozess erzeugt. Dabei wird für jeden Prozess eine Instanz von dllhost.exe aufgerufen, was sich negativ auf die Leistung des Webservers auswirkt. Beim Anwendungsschutz ist also ein klarer Gegensatz zwischen Sicherheit und Leistung zu erkennen. Niedriger Anwendungsschutz sollte allerdings nicht gewählt werden. Die Entscheidung zwischen mittlerer und hoher Anwendungssicherheit ist abhängig von den Anforderungen an die Sicherheit und Funktionalität eines Servers.

In der in Abbildung 65 dargestellten Maske können auch die Verzeichnisrechte geändert werden. Wird das Verzeichnis einer Homepage geändert, so kann dies ebenfalls in dieser Maske eingestellt werden. Dabei sollte die Standardeinstellung vermieden werden, was beispielsweise eine Maßnahme gegen ein Web Folder Traversal darstellt. Die Daten sollten generell auf einem anderen Laufwerk als die Programme gespeichert werden.

Dem Basisverzeichnis sollten gemäß dem Least-Privilege-Prinzip nur die nötigsten Rechte zugeordnet werden. In den Haupteigenschaften sind den Rechten im Reiter „Basisverzeichnis“ keine Rechte zuzuordnen.

Neben den bereits in Abschnitt 6.2 erwähnten Einschränkungen, können bei der Windows 2000 Professional Version keine Operatoren bestimmt werden, die eine bestimmte Website konfigurieren. Bei der Servervariante ist daher zu beachten, dass bei Nutzung die Gruppe Administratoren voreingestellt ist. Operatoren benötigen jedoch keine administrativen Rechte, weshalb die Rechte der Operatoren zu beschränken sind.

Um die Gefahr eines Denial-of-Service Angriffes zu mindern, kann im Reiter „Website“ die Anzahl der gleichzeitigen Verbindungen begrenzt werden und das Time-out gesenkt werden. Die dabei angegebenen Werte sollten einer realistischen Zugriffserwartung entsprechen, damit normale Besucher der Webseite nicht abgelehnt werden. Eine solche Einstellung ist auch für Anwendungen im Reiter „Optionen“ der Anwendungsoptionen zu konfigurieren. Die Anwendungsoptionen können durch die Schaltfläche „Konfigurieren“ im Reiter Website bearbeitet werden.

Neben den Einstellungen, die im IIS-Snapin vorgenommen werden, sind auch Einstellungen an den auf dem System gespeicherten Dateien vorzunehmen. So ist die Konfiguration des IIS in der Datei %systemroot%\winnt\inetsrv\metabase.bin gespeichert. Sie sollte umbenannt und in ein anderes Verzeichnis verschoben werden. Diese Änderung benötigt eine Anpassung der Registry, wobei Dateiname und Verzeichnis Wert „MetadataFile“ vom Typ REG_SZ im Schlüssel HKLM\Software\Microsoft\InetMgr\Parameters anzugeben ist. Zudem sind die Rechte dieser Datei auf die nötigsten zu beschränken.

Außerdem sollten keine Backup-Dateien von Skripten vorhanden sein. So kann ein Angreifer den Quellcode eines Skriptes erlangen und so beispielsweise die Zugangsdaten zu einer Datenbank erhalten (vgl. Abschnitt 6.7.2). Auch sind auf einem produktiven Server alle Beispielanwendungen zu entfernen, da diese ebenfalls Sicherheitslücken enthalten.

Pufferüberläufe, die einen Einbruch oder einen Denial-of-Service zum Ergebnis haben, werden häufig durch sehr lange Anfragen erzeugt. So existiert beispielsweise ein Exploit der in Abschnitt 6.7.2 beschriebenen Schwachstelle CAN-2003-0109, der durch eine bestimmte Anfrage mit mehr als 65535 Zeichen einen Denial-of-Service erzeugt. Solchen Angriffen kann durch die Begrenzung der maximalen Größe des http-Headers entgegengewirkt werden (vgl. [IIS Insider10/01]), die durch den DWORD-Wert MaxClientRequestBuffer im Schlüssel HKLM\CurrentControlSet\Services\W3SVC\Parameters in der Registry angegeben wird. Dabei ist zu beachten, dass der HTTP-Header nicht nur die gewünschte URL enthält. In [IIS Insider 10/01] wird 5000 als optimaler Wert empfohlen.

Viele der erkannten Schwachstellen des IIS existieren in der WebDAV-Komponente (vgl. Abschnitt 6.7.2), die bei Nutzung der Voreinstellungen aktiviert ist. Da sie nicht immer benötigt wird, sollte sie abgestellt werden. Dazu ist in der Registry im Schlüssel HKLM\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters dem DWORD-Wert „DisableWebDAV“ der Wert 1 zuzuweisen. Zudem sollte dem IIS-Webserver nach [SecWin2000:255] kein Zugriff auf die Datei %systemroot%\system32\inetsrv\httpext.dll erlaubt werden.

6.9.8. IIS Lockdown Tool und URL Scan

Viele der ebengenannten Einstellungen können auch durch die Installation des IIS-Lockdown Tools[84] vorgenommen werden. So werden unter anderem die unbenötigten ISAPI-Erweiterungen entfernt, WebDav deaktiviert und das Internetgastkonto der Gruppe "Web Anonymous Users" zugeordnet. Bei der Installation können Security Templates für mehrere Serverarten, unter denen auch ein IIS Webserver ist, installiert werden. Die vom Lockdown Tool installierten Security Templates sind jedoch nicht mit den auf Seite 151 erwähnten Templates zu verwechseln.

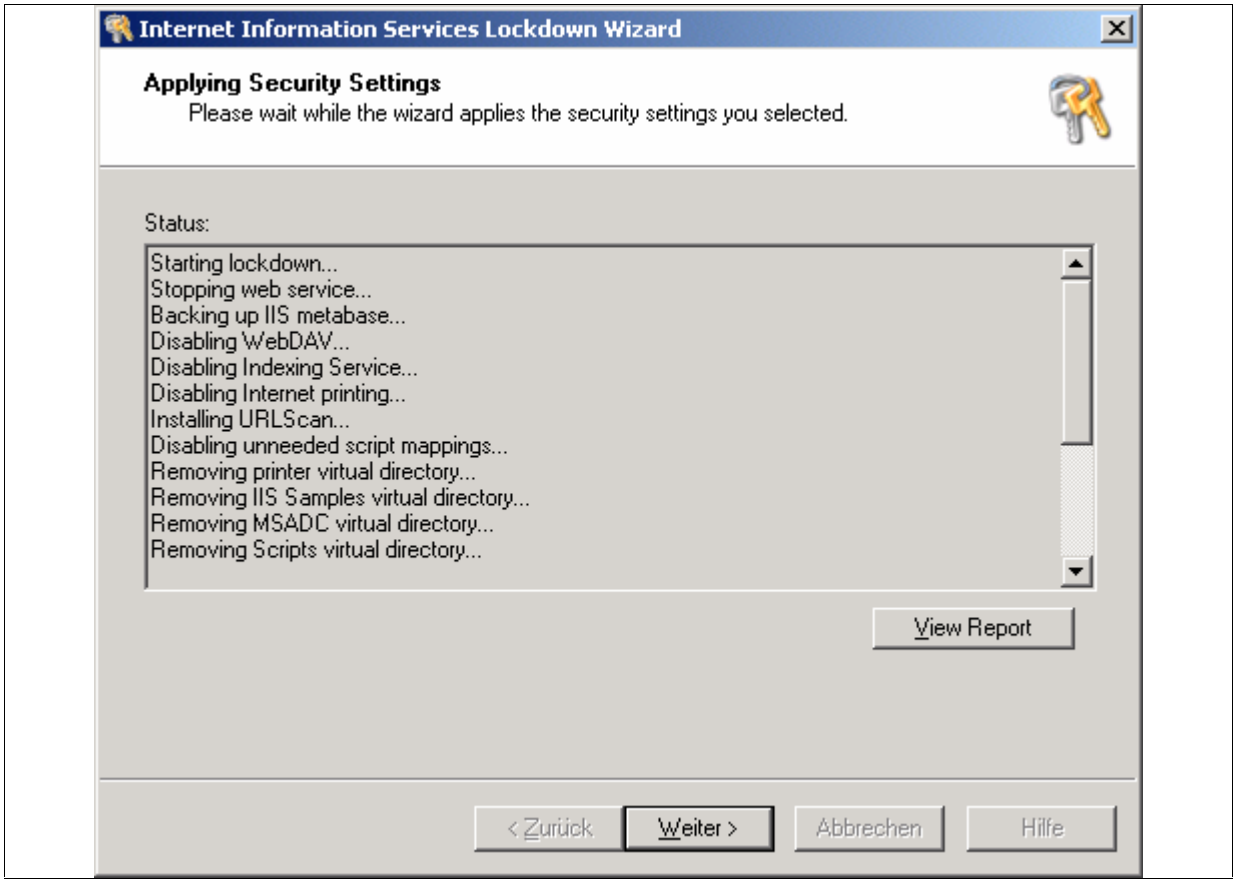


Abbildung 66: Installation des Lockdown Tools

Bei der Installation des Lockdown Tools kann auch der ISAPI-Filter URLScan[85] installiert werden. URLScan filtert eingehende http-Anfragen, bevor der Server sie verarbeitet. Somit können Restriktionen für die http-Anfragen dazu verwendet werden, um Unicode oder Escape-Zeichen herauszufiltern, wodurch einer Web Folder Traversal entgegengewirkt werden kann. Log-Einträge der Abwehr eines Web Folder Traversal Angriffes nach der Installation von URLScan sind in abgewandelter Form in Abbildung 67 zu erkennen.

[84] http://www.microsoft.com/technet/security/tools/tools/locktool.asp
[85] In früheren Versionen war URLScan ein eigenes Tool, das nicht in Lockdown enthalten war

URLScan.log:

```
[08-29-2003 - 16:10:34] Client at 192.168.0.18: URL normalization was not
complete after one pass. Request will be rejected.  Site Instance='1', Raw
URL='/scripts/..xxxxx../winnt/system32/cmd.exe'
```

IIS Log:

```
2003-08-29  23:10:58  192.168.0.18  -  192.168.0.3  80  GET  /<Rejected-By-
UrlScan>          ~/scripts/..xxxxx.../winnt/system32/cmd.exe          404
Mozilla/4.0+(compatible;+MSIE+5.01;+Windows+NT+5.0)
```

Abbildung 67: Abwehr eines Web-folder Traversal Angriffes mittels URL-Scan

Die Konfiguration von URLScan (vgl. [Q326444]) erfolgt durch das Bearbeiten der Datei %systemroot%\system32\inetsrv\urlscan\URLScan.ini mittels eines Editors. Bei der Konfiguration können Verben und Dateierweiterungen erlaubt oder verboten werden. Zudem kann die Canonicalasation, bei der die Verarbeitung bestimmter Zeichen wie die Unicodezeichen erfolgt, vor einem Scan konfiguriert werden. Dabei können auch beliebige Sequenzen in der URL verboten werden.

6.9.9. ServerMask

Ein weiteres nützliches Tool ist das kommerzielle Programm ServerMask der Firma Port80 Software Inc. Mittels dieses Tools können Modifikationen am http-Header vorgenommen werden. So kann der Angreifer in der Reconnaissance verwirrt werden, indem beispielsweise ein Apache Webserver vorgetäuscht oder gar kein http-Header gesendet wird.

6.9.10. Firewall

Eine Firewall kann durch Filterung des Netzverkehrs das bestehende Risiko reduzieren, indem der Zugriff auf Dienste unterbunden und beschränkt wird. So kann auch der Zugriff auf solche Dienste unterbunden werden, die ein Risiko bergen, aber nicht deaktiviert werden können. Ein Beispiel hierfür ist der RPC-Dienst auf Port 135/tcp.

Windows 2000 ermöglicht auch ohne Zusatzprodukte die Implementation eines einfachen Portfilters. Hierfür können mittels IPSec-Regeln eingehender und ausgehender Netzverkehr gefiltert werden.

Kommandozeilenbasiert kann die Konfiguration mit `ipsecpol` erfolgen, das zu denjenigen Dateien des Windows 2000 Resource Kits gehört, die frei heruntergeladen werden können. Die Konfiguration des Packet Screens, mit dem nur auf Port 80 des Webservers zugegriffen werden kann, ist mit den folgenden drei Befehlen zu realisieren (vgl. [SecWin2000:7]).

```
ipsecpol -w REG -p "Packet Filter" -r "HTTP" -f *+0:80:TCP -f *+0:135:UDP -n PASS
ipsecpol -w REG -p "Packet Filter" -r "All Inbound Traffic" -f *+0 -n BLOCK
ipsecpol -w REG -p "Packet Filter" –x
```

Allerdings ist die Installation des Kommandozeilen-Tools bedenklich. Sollte ein Angreifer eine Shell auf dem System öffnen können, so kann er mittels des Tools die Firewall umkonfigurieren und deren Schutzwirkung zunichte machen. Der Packet Screen lässt sich auch grafisch durch eine spezielle IP-Sicherheitsrichtlinie in den lokalen Sicherheitseinstellungen implementieren (vgl. [Northcutt02:274]). Generell unterliegt die Konfiguration eines Packet Screens durch IPSec in Windows 2000 den Einschränkungen, dass diese Richtlinie beim Systemstart erst spät geladen wird. Zudem wird Verkehr wie beispielsweise Kerberos erst bei Erstellung eines bestimmten Registrywertes gefiltert (vgl. [Q254728]). Ein einfacher Packet Screen, der weder zustandorientiert ist noch auf Applikationsebene filtern kann, bietet nur begrenzte Möglichkeiten der Filterung. Daher ist für die Implementation einer Firewall eine Gateway-Firewall auf einem separaten Host zu verwenden.

6.9.11. Angriffserkennung

Trotz getroffener Maßnahmen kann es immer noch zu Angriffen kommen. In diesem Fall sollten die Angriffe schnell erkannt und eingedämmt werden können, um die Schäden zu begrenzen. Des Weiteren sollte Material für eine kriminalistische Aufklärung eines Vorfalls, der Forensik, gesammelt werden, das auch als Beweismaterial in einer straf- und zivilrechtlichen Verfolgung verwendet werden kann.

Ein wichtiges Instrumentarium der Angriffserkennung und Forensik sind die Log-Dateien. Im Reiter „Website“ des IIS Snap-Ins kann die Log-Funktion eingestellt werden. Dies sollte in den Haupteigenschaften geschehen, damit die Funktion für alle Webseiten gilt. Dabei sollten möglichst viele Informationen geloggt werden. Allerdings treten dabei auch Schwachstellen auf. So kann bei einer großen Anzahl von Informationen die Log-Datei sämtlichen Speicherplatz verbrauchen, wodurch das System nicht mehr vollständig funktioniert und wertvolle Informationen verloren gehen können. Um Beweisunterlagen zu sichern, ist in regelmäßigen Abständen ein Backup der log-Dateien zu erstellen, und über einen längeren Zeitraum zu verwahren. Mit der Erstellung des Backups ist das alte Log zu löschen.

Ein weiteres Problem ist die Auswertung der Logs. Bei vielen Log-Einträgen können leicht Hinweise auf einen Angriff übersehen werden. Zudem birgt die richtige Deutung von Logeinträgen Schwierigkeiten. Dabei können Tools wie beispielsweise der Microsoft Free Log Analyser[86] helfen. Allerdings können auf diese Weise Angriffe erst nach dem Vorfall erkannt werden.

Um Angriffe bereits zum Zeitpunkt ihrer Durchführung erkennen und entsprechende Reaktionen einleiten zu können, empfiehlt sich die Installation eines Intrusion Detection Systems oder Programme zur Überwachung der Integrität. So prüft das Tool watznew[87] eine Webseite in regelmäßigen Abständen auf Änderungen und meldet diese dem Administrator. So können die in Kapitel 6.7 erzeugten Defacements der Website zum Zeitpunkt des Angriffs erkannt werden.

[86] http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=8CDE4028-E247-45BE-BAB9-AC851FC166
[87] http://www.watznew.com

6.9.12. MySQL

Für MySQL gelten gleiche Maßnahmen wie für einen Webserver. So werden Implementationsschwächen durch Patches behoben. Auch für die Datenbank sollte das Least-privileges Prinzip gelten. So wird in dem Szenario für das Auslesen der Produktpreise der Firma Alarm Mayer das Root-Konto der Datenbank benutzt. Der Kontoname und das Passwort stehen unverschlüsselt in der Datei produkte.php. Da ein Passwort in der php-Datei enthalten sein muss, ist zunächst mit den oben beschriebenen Sicherungsmaßnahmen zu verhindern, dass ein Angreifer lesenden Zugriff auf die Datei bekommen kann. Für den Fall, dass ein Angreifer dennoch Benutzername und Passwort erlangt, ist für den Zugriff durch .php-Dateien ein gesondertes Konto mit minimalen Privilegien zu schaffen.

6.9.13. Weitere Maßnahmen

Neben den beschriebenen sind auch weitere Maßnahmen zur Sicherung eines Webservers notwendig, auf die wegen des Umfangs in dieser Arbeit nicht näher eingegangen wird. Dazu gehören Backupkonzepte und ein effektiver Einsatz von Anti-Malware Produkten.

Weiter sind Daten auch bei der Übertragung über Netzwerke gegen Sniffing-Angriffe zu schützen. So kann der Übertragungsweg mittels der in IPSec enthaltenen Encapsulation Security Payload (vgl. [Heinzel03:22ff.]) durch die Verwendung von Verschlüsselungs-mechanismen gesichert werden.

Auch können bei der Verwendung von Authentifizierungsmechanismen Schwachstellen in eben diesen enthalten sein, denen entgegenzuwirken ist. Ein Beispiel ist die Verwendung von NTLMv2 statt LM oder NTLM sowie die Maßnahme, die Authentikation durch digitale Zertifikate zu unterstützen. Zudem sollten Server, die Dienstleistungen über ein WAN anbieten, nicht in einer Domain sein. Auch physikalische Sicherungsaspekte sind zu beachten.

6.9.14. Incident Response

Auf Grund fehlender oder unzulänglicher Maßnahmen kann es bereits zu einem erfolgreichen Angriff gekommen sein, dessen Ereignis und resultierender Schaden einen Vorfall (engl.: incident) darstellen. Die „Aufgaben und Funktionen, die mit der Reaktion auf Vorfälle [im] Zusammenhang stehen“ [Kosswakowski01:13] fallen unter den Begriff Incident Response. Dabei steht im technischen Sinne vor allem die Bewältigung und Beseitigung des Angriffes im Vordergrund.

Bei der Bewältigung und Beseitigung eines Vorfalls müssen neben dem Aufhalten des Angreifers alle durch den Angreifer erfolgten Änderungen rückgängig gemacht werden. Die letztere Aufgabe ist sehr komplex, da ein Angreifer nicht nur Dateien manipuliert haben kann, sondern auch Trojaner, Hintertüren oder Rootkits installiert sowie Benutzerkonten hinzugefügt haben kann. Zur erfolgreichen Reaktion auf einen Angriff empfehlen sich folgende Maßnahmen:

1. Der Rechner ist vom Netz zu nehmen.
2. Für forensische Zwecke ein Image des kompromittierten Servers erstellen.
3. Ein Backup einspielen, das vor dem Vorfall erstellt wurde. So wird vermieden, dass eine vom Angreifer erzeugte Änderung übersehen wird.
4. Die oben genannten Gegenmaßnahmen ergreifen, damit der Vorfall kein zweites Mal eintreten kann.
5. Benutzerkonten überprüfen, ob die Benutzer noch notwendig sind. Da ein Angreifer Benutzernamen und Passwörter erlangt haben kann, müssen die Passwörter gewechselt werden. Sofern es möglich ist, sollten auch Benutzernamen geändert werden.

Eine bisher nur beiläufig erwähnte Maßnahme ist die Installation einer Firewall auf einem Gateway, um den Zugriff auf den Webserver zu beschränken. Die Wirksamkeit dieser Maßnahme ist Inhalt des nun folgenden Kapitels.

7. KAPITEL:

SZENARIO FIREWALL

Der Penetrationstest wird in der Praxis bei der Kontrolle der Ordnungsmäßigkeit und wird somit auch bei der Revision einer Firewall verwendet. Ausgehend von dem in Kapitel 6 verwendeten Szenario soll durch den Penetrationstest die Wirksamkeit der Firewall bei der Sicherung gegen Angriffe betrachtet werden.

7.1. Theorie der Firewalls

Eine Firewall ist im Sinne der IT-Sicherheit ein Mechanismus, mit dem der Netzwerkverkehr an einem bestimmten Punkt zwischen mindestens zwei Netzen kontrolliert und eingeschränkt werden kann. Ist der Kontrollpunkt auf dem Rechner eines Endbenutzers, wobei nur der Netzverkehr zwischen eben diesem Rechner und dem angeschlossenen Netzwerk kontrolliert wird, so wird die Firewall als „Personal Firewall“ bezeichnet. Geschieht die Kontrolle auf einem Gateway im Netzwerk, wobei mehrere Rechner durch die Firewall geschützt werden, kann hingegen von einer Gateway-Firewall[88] gesprochen werden. Sie wird als Software auf einem separaten Host installiert, der auch als Firewall bezeichnet wird, wobei Hard- und Software auch in Form einer Appliance aus einer Hand geliefert werden können. Im Folgenden wird die Betrachtung auf die Gateway-Firewalls beschränkt, wofür lediglich der Begriff Firewall verwendet wird.

Der Umfang der Kontrolle ist in einer Policy definiert, die von der Firewall durchgesetzt (engl.: enforced) wird. Die Implementation der Firewall sollte zumindest die Prinzipien Single Point of Access und Default Deny umsetzen. Das Single Point of Access Prinzip besagt, dass die Firewall der einzige Zugangspunkt zu einem Netzwerk ist, da sonst unkontrollierte Kommunikationsbeziehungen zu einem unkontrollierten Netz bestehen können. Sind mehrere Zugangspunkte zu einem Netz notwendig, so sollten alle Zugangspunkte durch eine Firewall kontrolliert werden. Nach dem Default Deny Prinzip ist dabei jede Kommunikationsbeziehung verboten, die nicht ausdrücklich erlaubt ist, um unnötige Risiken durch einen nicht betrachteten Dienst oder dessen Protokoll zu vermeiden.

Für die Betrachtung der Firewalls in diesem Kapitel soll zunächst kurz auf die Komponenten einer Firewall eingegangen werden, die im Folgenden näher betrachtet werden.

[88] Hierbei ist nicht das Gateway-Firewall Konzept nach Mück (vgl. [Mück00:123]) gemeint, welches das Prinzip eines dual-homed host nach [Chapman00:123f.] beschreibt.

7.1.1. Packet Screen

Ein Packet Screen durchleuchtet ein- und ausgehende Pakete, um anhand von Informationen aus den Protokoll-Header der Vermittlungs- und Transportschicht zu entscheiden, ob ein Paket erlaubt oder verboten ist. Ein verbotenes Paket wird entweder verworfen (engl.: drop) oder zurückgewiesen (engl.: reject), wobei im letzten Falle ein entsprechendes Antwortpaket an den Sender zurückgeschickt wird. Die Felder eines Protokoll-Headers, die von einem Packet Screen betrachtet werden, sind in Abbildung 68 dargestellt.

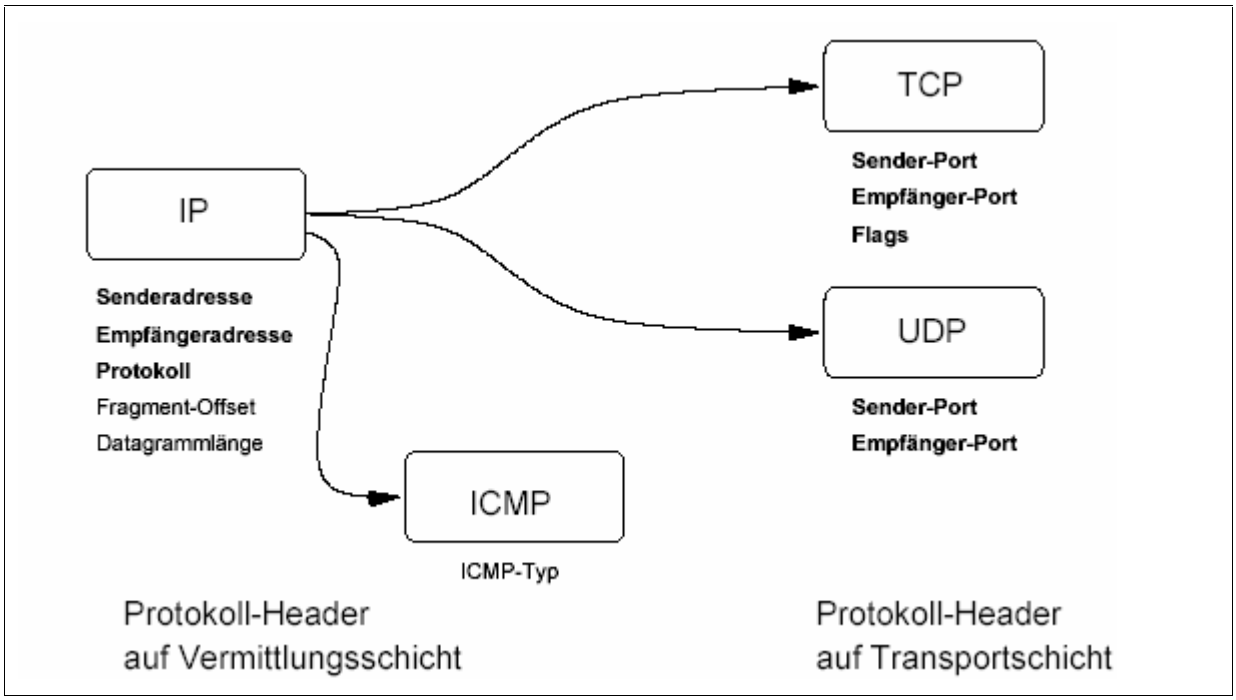


Abbildung 68: Von einem Packet-Screen verwertete Informationen (aus [Mück00:109])

Als Synonym für ein Packet Screen wird auch der Begriff Paketfilter (engl.: packet filter) verwendet, der streng genommen schon von dem Berkley Packet Filter, kurz bpf, belegt ist. Der Berkley Packet Filter kann jedoch lediglich Pakete filtern, die vom Netz abgehört[89] worden sind, und wird daher von vielen Netzwerkanalyse-Programmen wie tcpdump oder ethereal verwendet.

7.1.2. Stateful Packet Screen

Ein Stateful Packet Screen, nach Mück (vgl. [Mück00:109]) auch dynamischer Packet Screen genannt, ist ein um eine Zustandstabelle erweiterter Packet Screen. Ohne eine Zustandstabelle kann bei UDP auf Grund des fehlenden Verbindungsaufbaus nicht festgestellt werden, ob ein eingehendes UDP-Paket zum Aufbau einer neuen Kommunikationsbeziehung bestimmt ist, oder lediglich eine Antwort auf ein bereits gesendetes Paket ist. Ohne die Zustandstabelle sind die Client-Ports eines jeden auf dem UDP-Protokoll basierenden Dienstes in beide

[89] In diesem Zusammenhang wird häufig der Begriff capture benutzt.

Richtungen zu öffnen. Auch FTP benötigt die Unterstützung von Ports mit einer zufällig vergebenen Nummer. Mit Hilfe eines Stateful Packet Screens werden diese Ports erst bei Aufbau der UDP- oder FTP-Verbindung geöffnet.

7.1.3. Proxy und Application-Level-Gateway

Der Proxy, zu Deutsch Stellvertreter, ist ein Prozess, über den eine Verbindung ermöglicht wird. Ein Client schickt Anfragen an einen Proxy, der sie stellvertretend für den Client an den Server weiterschickt, wobei eine Kontrolle der Kommunikationsbeziehung erfolgen kann. Proxies existieren in zwei Varianten (vgl. [Chapman00:231f.]). Während der Circuit-Level-Proxy nur Informationen der Transportschicht betrachtet und die Nutzdaten der höheren Schichten unbeachtet lässt, arbeitet ein Application-Level-Proxy auf der Anwendungsschicht. Der Proxy kann dabei die Zugriffskontrolle durch Authentifizierung erweitern und bietet auf Anwendungsschicht Caching und Content-Filtering. Zudem kann ein Proxy genauere Log-Informationen erzeugen als ein Packet Screen.

Da durch den Proxy keine direkte Kommunikationsbeziehungen zwischen Client und Server besteht, können Angriffe auf den OSI-Schichten 3 und 4 von einem Proxy herausgefiltert werden. Auch Angriffe, die durch das Content-filtering[90] herausgefiltert werden, durchdringen den Proxy nicht. Im Gegensatz zu einem Gateway verarbeitet der Proxy die Protokolldaten nicht, sondern schickt stellvertretend für einen Client an einen Server weiter. Dies ist besonders auf der Anwendungsschicht ein Nachteil, da so maliziöse Befehle direkt an einen Server vermittelt werden und so ein Angriff den Proxy durchdringen kann. Bei einem Application-Level-Gateway, das keine kompatiblen Schichten aufweist, erfolgt die Vorverarbeitung der Dienstbefehle auf dem Gateway, so dass Angriffe von diesem abgefangen werden können. Daher muss das Application-Level-Gateway zumindest Teile der Serverfunktionen realisieren. Das Konzept des Applikation-Level-Gateways wird häufig auch als Relay-Konzept bezeichnet. Die bekannteste Form ist ein Mail-Relay, bei dem Emails von einem zweiten Mailserver verarbeitet werden, bevor sie an den primären Mailserver weitergeleitet werden.

7.1.4. Network Address Translation

Obwohl die Network Address Translation, kurz NAT, kein eigentliches Element einer Firewall ist, soll sie hier dennoch besprochen werden, da sie wie im Falle von iptables meist durch eine Firewallsoftware implementiert wird und mittlerweile eine Grundausstattung einer jeden Gateway-Firewall ist.

NAT (vgl. [RFC3022]) dient der Übersetzung von IP Adressen auf einem Gateway. Im traditionellen NAT ist nach [RFC3022:3] nur die Übersetzung der Adressen ausgehender Verbindungen möglich. Dabei werden die internen Adressen des Senders transparent in eine extern verfügbare Adresse übersetzt, wobei sowohl statische (1:1) als auch dynamische (n:m) Zuordnungen möglich sind. Ursprünglich sollte mit der Technik der Verknappung von IPv4-Adressen entgegengewirkt werden. Vorteilig für die IT-Sicherheit ist der Umstand, dass die Rechner hinter dem NAT-Gateway einem außenstehenden Angreifer verborgen sind. Bei

[90] Beim Content-filtering wird der Datenstrom mit Hilfe regulärer Ausdrücke nach Mustern wie das Vorkommen bestimmter Dateitypen durchsucht. Professionelle Firewalls können hierbei auch nach Mustern von Angriffen suchen (vgl. SmartDefense auf Seite 169).

Verwendung von privaten IP-Adressen (vgl. [RFC1918]) für Rechner im internen Netz kann ein Angreifer außerdem nicht direkt auf einen der Rechner über das Internet zugreifen.

Um die reale Adresse eines Servers zu verbergen, besteht für eingehende Verbindungen die Möglichkeit der statischen NAT, die eine private einer öffentlichen Adresse zuordnet. In der erweiterten Form, NAPT, werden neben der IP-Adresse auch die Adressierungsinformationen der Transportschicht, wie Ports im Falle von TCP/UDP und Typs im Falle von ICMP, übersetzt.

7.1.5. DMZ und Bastion

Die Grenze eines Netzwerks wird in der IT-Sicherheit als Security Perimeter bezeichnet. Wird diese Grenze durch ein separates Netzwerk zwischen internem und externem Netz gebildet, so wird eben dieses Netz als Perimeternetz bezeichnet. Ausgehend von der Demilitarisierten Zone zwischen den beiden Koreanischen Staaten, die weder zu Nord- noch zu Südkorea gehört, wird auch das Perimeter Netz als Demilitarisierte Zone, kurz DMZ, bezeichnet (vgl. Chapman00:103]). Aufgabe eines Perimeternetzes ist die Bildung von Sicherheitszonen, die mit Brandabschnitten eines Gebäudes im Brandschutz vergleichbar sind.

Eine Spezialform einer DMZ bildet das Screened Subnet. Bei einem Screened Subnet wird sämtlicher Verkehr zwischen DMZ, externen und internem Netz durch Packet Screens gefiltert. Die DMZ enthält Bastion Host und Server wie den Webserver. So soll der Angreifer im Falle einer Kompromittierung eines Systems der DMZ davon abgehalten werden, Angriffe auf das interne Netz durchzuführen. Grundprinzip der DMZ ist, dass externe nur auf Rechner in der DMZ zugreifen können, nicht aber auf Rechner im internen Netz. Verbindungen zwischen Rechnern der DMZ und dem internen Netz sollten nur von Systemen des internen Netzes aufgebaut werden können, so dass sich der Angriff nicht auf Rechner des internen Netzes ausweiten kann.

Ein Bastion Host wird allgemein definiert als ein besonders abgesicherter Host, der von außen zugreifbar ist und überwunden werden muss, um einen Rechner im internen Netz zu erreichen (vgl. [Chapman00:241] und [Mück00:120]). Auf dem Bastion Host sind Proxies und Application-Level-Gateways installiert. Über den Bastion Host wird auf Server zugegriffen, die ebenfalls von außen sichtbar sind und überwunden werden müssen, um in das interne Netz zu gelangen, die nach [Chapman00:243] als sekundäre Bastionen bezeichnet werden.

Die Art der DMZ sowie die Position des Bastion Hosts unterscheiden sich in den verschiedenen Architekturen. Für mehr Informationen zu Firewall-Architekturen wird auf [Chapman00:122ff.] und [Mück00:117ff.] verwiesen.

7.2. Bewertungsverfahren einer Firewall

Eine Firewall im Sinne der IT kann als eine Metapher der Brandmauer im Brandschutz eines Gebäudes angesehen werden. Ebenso, wie eine Brandmauer die Ausbreitung eines Brands im Gebäude vermeidet oder zumindest verlangsamt, vermeidet oder verlangsamt eine Firewall die Angriffe aus dem Internet. Das Böse im Internet kann daher metaphorisch als ein Flächenbrand des Bösen gesehen werden, vor dem eine Firewall ein Netzwerk schützen soll.

Bei der Bewertung von Brandmauern in Gebäuden sind Vorgaben von Brandschutzrichtlinien zu beachten, wie sie z. B. in [BMWA02] enthalten sind. Dabei muss beachtet werden, dass eine Brandmauer der Brandwiderstandsklasse F30 einem Feuer von 800°C mindestens 30 Minuten widerstehen muss. Des Weiteren sind auch die Art des Materials, die Einteilung von Brandabschnitten sowie Umgehungsmöglichkeiten eines Brandabschnittes zu betrachten. So darf die Lagerung brennbarer Materialien keine Brandabschnitte überbrücken.

Ähnliche Überprüfungen gelten auch für Firewalls. Bei der Überprüfung einer Firewall kann als Ziel betrachtet werden,

- ob die Firewall zu jeder Zeit den Anforderungen entspricht (vgl. [Todd98:1]) und
- ob die Firewall effektiv ist (vgl. [Schultz96:1]) und somit das „angestrebte Schutzniveau tatsächlich erreicht wurde“ [Veit99:2].

Die Anforderungen an eine Firewall lassen sich nach [Mück00:109] differenzieren in

- *Kommunikationsanforderungen*, welche durch die benötigten Kommunikationsbeziehungen abgegrenzt sind,
- *Sicherheitsanforderungen*, die durch die Risiken charakterisiert sind, die durch den Betrieb der Firewall und den erlaubten Kommunikationsbeziehungen bestehen, sowie
- *Performanceanforderungen*, welche durch den zu erwartenden Datendurchsatz beschrieben werden können.

Die Anforderungen an die Firewall werden in einer Sicherheitspolitik[91] festgelegt. Die erste der oben genannten Zielsetzungen, ob die Firewall zu jeder Zeit den Anforderungen entspricht (vgl. [Todd98:1]), ist durch die Überprüfung des Sicherheitskonzeptes zu bewältigen. Dabei wird nicht nur die Sicherheitspolitik sondern auch das Design der Firewall überprüft. Zum Design gehört nach Todd (vgl. [Todd98:5]) die Betrachtung der Komponenten der Firewall sowie ihre Architektur, die beispielsweise eine Screened-Subnet-Architektur sein kann. Diese Betrachtung ähnelt einer Begutachtung des Materials einer Brandmauer sowie der Einteilung der Abschnitte.

Grundlage des Sicherheitskonzeptes ist die Risikoanalyse, die bei der Überprüfung zu wiederholen bzw. an aktuelle Anforderungen wie Bedrohungen oder neue Kommunikationsbeziehungen anzugleichen ist. Diese Überprüfung gleicht der Ebene Features der Revision. Um weiter eine Zusicherung zu erhalten, dass die Firewall das Konzept richtig implementiert, wird die aktuelle Konfiguration hinsichtlich des Konzeptes verglichen. Fehler in der Konfiguration können zu einem Schaden führen und bilden somit eine Schwachstelle im Sinne der Definition (siehe Seite 13). Somit wird die Implementation auf Schwachstellen

[91] Zu Sicherheitspolitiken für Firewalls siehe [GSHB02:M2.070] und [SP800-41:33ff.]

untersucht, die neben Fehlern in der Software auch durch Fehler in der Konfiguration hinsichtlich des Sicherheitskonzeptes bedingt sein können. Dabei werden auch Tests der Funktionalität einer Firewall durchgeführt, um zu zeigen, dass Filter, Routing, Logging und Alert-Funktionen[92] wie erwartet funktionieren. Diese Vorgehensweise entspricht der Assurance I.

Durch den Penetrationstest, der die Assurance II darstellt, werden alle Elemente der in Abschnitt 3.4.3 besprochenen Revision im Kontext der Firewall durchgeführt. Dabei werden die Firewall und die von ihr zugelassenen Kommunikationsbeziehungen den Angriffen ausgesetzt. Zweck dieser Methode ist die Wirksamkeit der Firewall zu zeigen und somit zu betrachten, welchen Angriffen die Systeme widerstehen und welchen Angriffen durch weitere Schutzmaßnahmen vorgebeugt werden muss.

Problematisch ist allerdings, ob nur einige wenige Angriffe oder aber alle möglichen Angriffe betrachtet werden. Alle möglichen Angriffe zu betrachten, ist auf Grund verschiedener Variationen der Elemente eines Angriffsvektors unmöglich. Daher sollten lediglich Klassen von Angriffen betrachtet werden. Dabei ist auch zu hinterfragen, wie viele Informationen durch eine Reconnaissance beschafft werden können.

Mittels des Penetrationstests kann auch die Alarm- und Logging-Strategie überprüft werden. Diese Strategie ist meist schwer zu konzipieren und umzusetzen, da zu viele Logs leicht zum Überlauf der Log-Dateien führen können und schwer zu analysieren sind. Zudem können zu wenige Logs dazu führen, dass Angriffe nicht erkannt werden und beweiskräftige Informationen fehlen. Durch einen Penetrationstest kann einem Verantwortlichen bewusst werden, was einzelne Log-Einträge bedeuten, sowie die Fähigkeit erlernt werden, Angriffe anhand der Logs zu erkennen. Diese Aufgabe kann allerdings durch ein Intrusion Detection System erleichtert werden.

Ebenso wie brennbare Materialien zu einer Überbrückung von Brandabschnitten führen können, können andere Zugänge zum internen Netz zu einer Leckage führen. Zu solchen Leckagen zählen auch administrative Zugänge von außen zu der Firewall, die ein Angreifer zur Manipulation der Firewall ausnutzen kann, um ungehindert ins interne Netz zu gelangen. Bei einem effektiven Security Perimeter bestehen nach Schultz (vgl. [Schultz96:2]) keine Leckagen.

In dieser Arbeit soll allerdings nicht die gesamte Revision eines Firewallkonzeptes behandelt werden, sondern lediglich betrachtet werden, welche Möglichkeit der Penetrationstest dabei bietet. Die Vorgehensweise wird im nächsten Abschnitt genauer erörtert.

[92] Die Überprüfung dieser vier Funktionen stellt die Zeile der Firewallüberprüfung nach [Allen01:162] dar.

7.3. Versuchsbeschreibung

In diesem Abschnitt wird der Versuch konzipiert, mit dem ein Penetrationstest gegen eine Firewall und ein internes Netz durchgeführt werden soll.

7.3.1. Zielsetzung und Vorgehen

Zielsetzung des Versuchs ist die Betrachtung der Wirksamkeit einer aktuellen Firewall gegen Angriffe durch einen Penetrationstest zu betrachten. In Anlehnung an [Allen01:162] soll dabei neben den Filter- und Routingfunktionalitäten auch die Logging- und Alarmfunktionalitäten betrachtet werden.

Problematisch gestaltet sich dabei die Abgrenzung zu der Schwachstellenanalyse in der Assurance I. So ist eine falsche Filterregel ein Konfigurationsfehler, durch den eine Inkonsistenz zur Policy entsteht. Ein solcher Fehler kann durch den automatisierten Test der Regelbasis entdeckt werden, bei dem Pakete durch die Firewall geschickt werden, wobei jede Regel auf erwartungsgemäße Funktionalität geprüft wird. Dabei werden auch Pakete in den Netzwerkverkehr injiziert, die von der Firewall verworfen werden sollten. Solche Pakete könnten auf Grund der fehlerhaften Regel die Firewall durchdringen. Daher ist schon die Suche einer fehlerhaften Regel durch das Versenden spezieller Pakete der Versuch einer Penetration und kann daher auch Bestandteil des Penetrationstests sein. Zudem wird ein Angreifer, der bei einer zero-knowledge Penetration kein Wissen über die Konfiguration der Firewall hat, mittels eines solchen Tests versuchen, Kenntnisse über die Regelbasis zu bekommen.

Die Überprüfung, ob Filter oder Logging wie erwartet funktionieren, ist in manchen Fällen nur durch einen Penetrationstest möglich. So bieten die meisten Firewalls eine Filterung von SYN-Flood Angriffen. Um diese Funktionalität auf ihre korrekte Funktion zu überprüfen, ist ein SYN-Flood gegen ein durch die Firewall geschütztes System durchzuführen.

Auf eine genaue Zuordnung der eben behandelten Methoden zu einer der beiden Assurance-Phasen der Revision soll an dieser Stelle verzichtet werden, da die Methoden sich für beide Phasen eignen. Methoden wie der Test der Regelbasis oder der SYN-Flood Filterung sollen daher auch in den hier durchgeführten Penetrationstest aufgenommen werden.

Der Penetrationstest soll aber auf Grund des zu großen Aufwandes, der auch durch die gegebene Zeit nicht vertretbar ist, nicht im Sinne der Vollständigkeit die Penetration aller Schwachstellen beinhalten. Bei diesem Versuch sollen vorwiegend nur die im Szenario des vorherigen Kapitels benutzten Angriffe behandelt und so die Wirksamkeit der Schutzmaßnahme Firewall betrachtet werden. Da diese Angriffe zum Teil Schwachstellen der Applikationsschicht ausnutzen, soll eine Firewall betrachtet werden, die über Filterfunktionalitäten auf Applikationsschicht verfügt.

7.3.2. Versuchsaufbau

Die für diesen Versuch gewählte Firewall muss kostenfrei zur Verfügung stehen und die Filterung auf Anwendungsschicht bieten. Aus diesem Grund wurde das Produkt Firewall-1 der Firma Check Point Software Technologies LTD gewählt. Firewall-1 ist Bestandteil des Check Point Next Generation Feature Pack 3, das von der Firma Check Point in einer Evaluationsversion zur Verfügung gestellt wurde.

Für die Installation der Firewall wurde der Rechner Obelix gewählt, der baugleich dem Rechner Miraculix ist (vgl. Seite 109) und mit einer zweiten Netzwerkkarte ausgerüstet wurde. Die Firewall ist für Microsoft Windows und verschiedene UNIX-Derivate verfügbar, wobei allerdings nur die Windowsversion über eine grafische Benutzeroberfläche (GUI) verfügt. Um die Komplexität der Konfiguration durch die GUI zu reduzieren, wurde Microsoft Windows NT 4.0 Server als Betriebssystem gewählt. Vor der Installation wurden sämtliche für das Betriebssystem verfügbaren Patches eingespielt. Im Rahmen der Check Point Next Generation Feature Pack 3 wurden folgende Komponenten installiert:

- VPN-1&FireWall-1
- SMART Clients
- User Authority
- Policy Server
- SmartView Monitor
-

Neben dem Rechner Obelix wurden auch die Rechner Kathy, Ergo und Miraculix aus dem Versuch des vorherigen Kapitels in der in Abbildung 69 dargestellten Topologie verwendet.

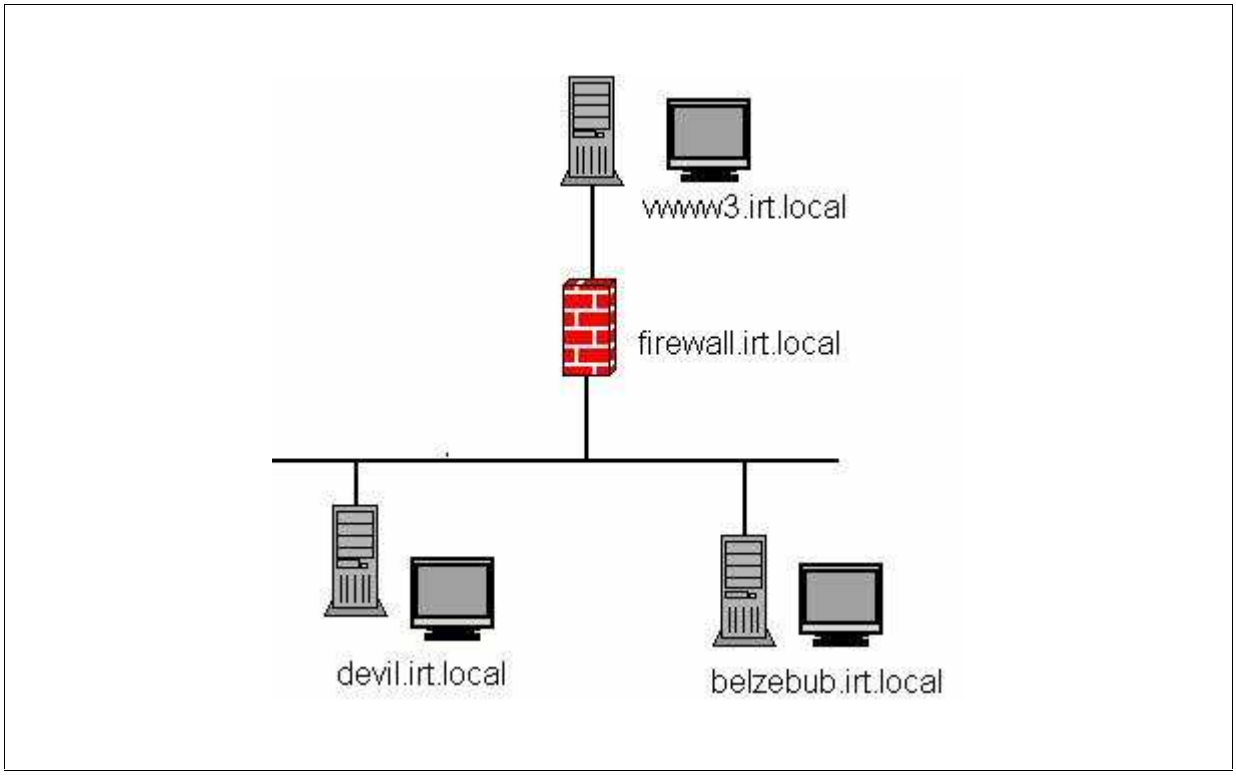


Abbildung 69: Topologie des Testnetzes mit Firewall

Durch die Firewall wird das Testnetz in zwei Subnetze gespalten. Der dem internen Netz zugeordnete Netzwerkadapter der Firewall erhielt die IP-Adresse 10.0.0.1/24, während dem externen Adapter die Adresse 192.168.0.254/24 zugewiesen wurde. So mussten die Angreifersysteme nicht umkonfiguriert werden. Dem Opfersystem wurde die IP-Adresse 10.0.0.3/24 zugeordnet. Mittels NAT erhielt es im externen Netz die IP-Adresse 192.168.0.42/24.

Somit mussten an den beiden Netzwerkadaptern der Firewall verschiedene IP-Adressen zugeordnet werden. Das interne Netz, dem das Opfersystem www3 angehört, erhielt die Adresse 10.0.0.0/24.

Im Rahmen der Vorgaben des in Abschnitt 6.2 spezifizierten Szenarios, sollen die Dienste http, ftp und smtp erlaubt sein. Zwar werden nur Angriffe auf den Webserver durchgeführt, jedoch sollen auch die anderen Dienste erlaubt sein, um die Wirksamkeit der Firewall gegen das Banner Grabbing untersuchen zu können. Dazu wurden die in Abbildung 70 dargestellten Regeln im SecureDashboard konfiguriert.

NO.	SOURCE	DESTINATION	IF VIA	SERVICE	ACTION	TRACK	INSTALL ON	TIME
1	* Any	webserver	* Any	TCP http	accept	Log	firewall	* Any
2	* Any	webserver	* Any	TCP ftp	accept	Log	firewall	* Any
3	* Any	webserver	* Any	TCP smtp	accept	Log	firewall	* Any

Abbildung 70: Regelbasis der Firewall

Die Regeln der dargestellten Regelbasis ermöglichen lediglich den Zugriff auf die erlaubten Dienste. Weitere Eigenschaften bestimmter Protokolle und Dienste können in den in Abbildung 71 dargestellten Einstellungen der SmartDefense bestimmt werden. Dort kann beispielsweise der Worm Catcher aktiviert werden, der den http-Verkehr nach bestimmten regulären Ausdrücken durchsucht, die auf einen Wurm schließen lassen. Für die Versuche wurde zunächst die Standardkonfiguration der SmartDefense verwendet, um die Wirksamkeit einer vorkonfigurierten Firewall-1 zu zeigen.

7.3.3. Vorgehen

Ausgehend von den Zielsetzungen soll der Penetrationstest zunächst betrachten, wie wirksam die Firewall den Methoden der Reconnaissance widerstehen kann. In einer Schwachstellenerkennung wird geprüft, ob die Regelbasis richtig konfiguriert ist. Die Penetration beinhaltet einen SYN-Flood Angriff sowie die Web Folder Traversal und der ntdll.dll Buffer Overflow Angriffe aus dem vorherigen Kapitel.

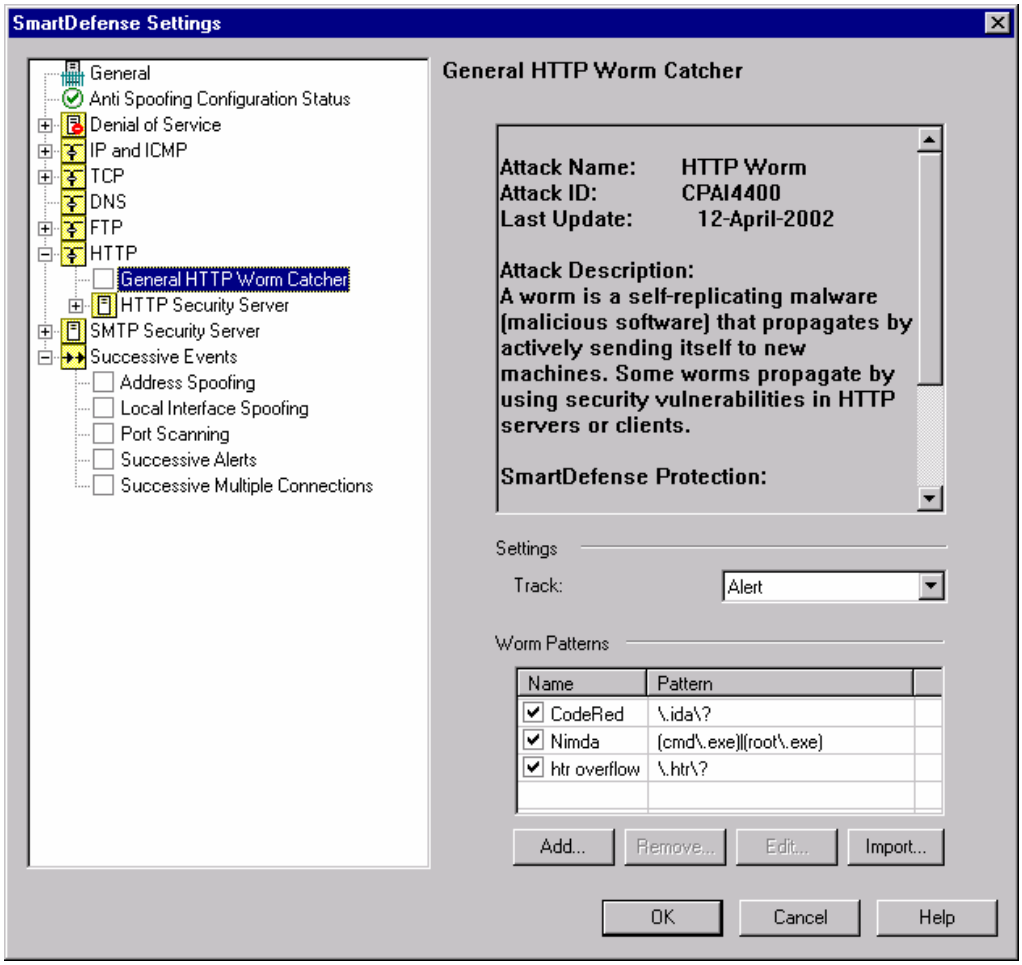


Abbildung 71: SmartDefense Einstellungen der Checkpoint Firewall-1

7.3.4. Einschränkungen

Auf Grund der gewählten Topologie ergeben sich Einschränkungen der Versuchsdurchführung. So sollte eine grafische Benutzeroberfläche, die GUI, nicht direkt auf dem Firewall-Host installiert sein, da Schwachstellen in der GUI die Sicherheit der Firewall gefährden können. Für den Fall, dass die GUI auf einem separaten Rechner installiert ist, ist in einer Sicherheitsüberprüfung die Verbindung zwischen GUI-Host und Firewall hinsichtlich Verschlüsselungs- und Authentifizierungsmethoden zu überprüfen. Gleiches gilt auch für die Log-Dateien, auf die ein Angreifer bei einem Einbruch in die Firewall keinen Zugriff erlangen soll. Diese Überprüfungen sind in diesem Versuch nicht nötig, da die GUI direkt auf dem Rechner „Firewall“ installiert wurde.

In der Literatur wird die Verwendung einer Screened-Subnet Architektur empfohlen, bei der die Firewall-1 die Funktion des Bastion Host erfüllt und in einem Screened-Subnet installiert ist, das zusätzlich durch Packet-Screens gesichert ist. In den Versuchen wird lediglich eine Dual-homed-host Architektur verwendet, deren Nachteile im Falle der Kompromittierung der Firewall in diesem Szenario nicht aufgezeigt werden.

Weitere Einschränkungen bilden die beschränkte Funktionalität des Opfersystems sowie das kleine Testnetz. Dabei sind nur wenige Dienste erlaubt, weshalb bei der Konfiguration wenige Erlaubnisregeln ausreichen. In großen Organisationen hingegen besteht die Regelbasis aus mehreren hundert Regeln, die über die Zeit verändert werden. Diese Änderungen sind eine mögliche Fehlerquelle. Zudem ist das Testnetz nicht groß genug, um zu überprüfen, ob noch andere Zugänge vorhanden sind. Eine solche Überprüfung kann durch eine Reconnaissance geleistet werden, die auf Grund der geringen Größe des Testnetzes an dieser Stelle vernachlässigt wird.

Zudem ist der Versuch auf die im Kapitel 6 behandelten Angriffe beschränkt. Angriffe, deren Ziel die Manipulation des Firewall-Systems ist, werden in dieser Arbeit nicht betrachtet werden.

7.4. Durchführung

Um das Ziel, die Wirksamkeit der Firewall gegen die betrachteten Angriffe zu überprüfen, werden diese systematisch gegen die konfigurierte Firewall durchgeführt. Begonnen wurde dabei mit einer Reconnaissancephase. Da eine Firewall eine öffentliche Informationsbeschaffung sowie whois-Abfragen beim Registrar nicht beschränken kann, wurden diese Phasen ebenso wie die DNS-Abfragen nicht beachtet. Es wird angenommen, dass der für die Zone autorisierte Nameserver in den Räumen des Providers steht und somit nicht zum Netz der Firma Arachnolocus gehört.

Unter diesen Prämissen werden in diesem Abschnitt die weiteren Elemente der Reconnaissance sowie die in Kapitel 6.7 behandelten Angriffe auf die Firewall angewendet.

In der Standardkonfiguration der Checkpoint Firewall-1 konnte weder ein Traceroute in allen Varianten noch ein Ping die Firewall oder den Webserver erreichen. Ein OS-Fingerprint führte ebenfalls zu keinem Ergebnis.

Bei einem Portscan mit dem Tool nmap wurde jeweils die Firewall selbst als auch der Webserver auf TCP- und UDP-Ports gescannt. Beim TCP-Portscan der Firewall wurde der Port 500/tcp als geschlossen und der Port 264/tcp als offen gemeldet. Über den auf Port 264/tcp hörenden Dienst FW1-Topo kann ein Kommunikationspartner von außen Informationen über die Topologie hinter der Firewall erhalten (vgl. [Checkpoint02:106ff.]). Durch die Topologieinformationen kann ein VPN-Partner erkennen, zu welchen Systemen im durch die Firewall verdeckten internen Netz er eine VPN aufbauen kann. Ein solcher Dienst birgt Risiken. Zwar ist die Kommunikation durch ein asymmetrisches Verschlüsselungsverfahren (vgl. [Checkpoint02:179]) verschlüsselt, jedoch wurden auf der Bugtraq-Mailingliste Schwachstellen in den Authentifizierungsmechanismen und die Verwundbarkeit gegen Flooding-Angriffe diskutiert, die an dieser Stelle nicht weiter erörtert werden. Alle anderen TCP-Ports sind im Status „filtered“. Dieser Status wird von nmap genau dann angegeben, wenn eine der drei folgenden Bedingungen erfüllt ist (vgl. [Kurtz01:581]):

- es wurde kein SYN/ACK-Paket empfangen
- es wurde kein RST/ACK-Paket empfangen
- es wurde eine ICMP_DESTINATION_UNREACHABLE – „Communication Administratively Prohibited“ (Typ 3, Code 13) empfangen

Bei dem hier durchgeführten Versuch wurden alle eingehenden Pakete verworfen.

Ein UDP-Portscan der Firewall meldete nur offene udp-Ports. Dies ist nicht verwunderlich, da nur bei einem geschlossenen Port ein ICMP_DESTINATION_UNREACHABLE an den Sender zurückgeschickt wird. Unterdrückt eine Firewall solche Pakete, so werden bei einem Portscan sämtliche Ports als offen gemeldet. Daher wurden auch bei einem UDP-Portscan des Webservers nur offene Ports gemeldet.

Bei den Portscans wurde mit Ethereal, das auf dem Rechner www3 installiert wurde, der Netzwerkverkehr beobachtet. Hierbei konnten auf www3 nur Pakete beobachtet werden, die an den Web-, FTP- oder Email-Server gerichtet waren. Somit ist festgestellt worden, dass die Regelbasis den Erwartungen entsprechend funktioniert.

Um die Funktionalität des Stateful Packet Screens zu demonstrieren, wurden der Firewall-1 zwei Regeln hinzugefügt, die ausgehende SNMP-Pakete zulassen und alle eingehenden UDP-Pakete zurückweisen. Das Zurückweisen wird durch die Aktion „REJECT“ erreicht. Für den Zeitraum einer ausgehenden SNMP-Verbindung ist durch die Firewall-1 der Client-Port für die SNMP-Antworten zu öffnen. Nach der Installation der Regeln wurden bei einem Portscan sämtliche UDP-Ports als geschlossen gemeldet. Der Scan wurde wiederholt, während der Webserver ein SNMP-walk über alle Einträge der OID iso.dod.org.internet.mgmt.mid-2 des Angreifers Belzebub durchführte, auf dem für diesen Zweck der SNMP-Dienst installiert wurde. Als Ergebnis wurde erkannt, dass während des Scans der Port 1035/udp geöffnet war. Die Portnummer änderte sich bei Wiederholung der Versuche. SNMP-Anfragen an www3 lieferten kein Ergebnis. Dadurch ist die erwartete Funktion des Stateful Packet Screens demonstriert worden.

Ein Banner Grabbing war jedoch auch mit der Check Point Firewall-1 möglich. Um das Banner Grabbing wirksam zu verhindern, sind somit auch auf dem Server Maßnahmen zu treffen.

Der Web Folder Traversal Angriff war bei Nutzung der Standardkonfiguration möglich. So konnte zunächst durch eine http-Anfrage nach der Datei blah.ida das Wurzelverzeichnis des Webservers erkannt werden. Auch die Verzeichnisstruktur konnte wie in Kapitel 6.7.1 über den Internet Explorer angesehen werden (vgl. Abbildung 55 auf Seite 132). Auf Grund des Outboundfilterings der Firewall war es jedoch nicht möglich, Dateien auf den Host www3 zu kopieren. So war weder die Änderung der Webseite noch das Installieren von Back Orifice 2000 möglich. Allerdings konnten im Web-Verzeichnis als auch im Hauptverzeichnis Dateien gelöscht werden, wodurch ein Schaden entstehen kann.

Einem Web Folder Traversal kann aber durch die korrekte Installation der Firewall begegnet werden. Dazu ist in den Einstellungen des http-Protokolls in SmartDefense der Worm Catcher (vgl. Abbildung 71) zu aktivieren. Hierbei sollte aber das Muster für Code-Red durch den Regulären Ausdruck *.ida ersetzt werden, um der Enthüllung des Web-Verzeichnisses zu begegnen. Nach Aktivierung des Worm Catchers wurde der Web Folder Traversal Angriff von der Firewall geblockt, was im Log mit der Information „URL filter pattern detected: /scripts/..\../winnt/system32/cmd.exe?/c+dir+c:\“ verzeichnet wurde.

Ein Angriff mit dem WebDAV Exploit wurde mit dem Log-Eintrag „message_info: Line in HTTP request too long“ verhindert. So kann die Firewall-1 generell vor Buffer Overflow Angriffen schützen.

7.5. Denial-of-Service

Während die bisherigen Versuche auf die Störung der Integrität zielten, soll in diesem Abschnitt auch eine Störung der Verfügbarkeit durch einen Denial-of-Service Angriff untersucht werden. Ein Weg zum Erreichen eines Denial-of-Service ist das Erzeugen eines Pufferüberlaufs in einem Dienst. Da ein Pufferüberlauf bereits durch den webdav-exploit betrachtet wurde, soll in diesem Abschnitt ein Flooding-Angriff durchgeführt werden.

Wesentliches Merkmal des Floodings ist das Überlasten eines Opfersystems durch Anfragen. Ein Fraggle-Angriff überlastet beispielsweise einen bestimmten UDP-Port. Da in dem Szenario der Zugriff auf sämtliche UDP-Ports durch eine Firewall unterbunden wird, ist das Hauptkriterium für die Auswahl des Angriffes die Möglichkeit, ein System hinter der Firewall zu erreichen. Daher ist der in Abschnitt 4.7.7 beschriebene TCP-SYN Flood Angriff Gegenstand des an dieser Stelle durchgeführten Versuchs, da SYN-Pakete an das Opfersystem weitergeleitet werden.

Zur Zeitersparnis wurde der Exploit nicht selbst programmiert, sondern ein im Internet verfügbarer Exploit benutzt. Bei der Verwendung solcher im Internet verfügbarer Exploits ist zu beachten, dass diese häufig Fehler enthalten. So funktioniert der hier verwendete Exploit nicht, da jedes von ihm gesendete SYN-Paket keine neue Socket initiierte.[93]

Zunächst wurde ein SYN-Flood-Angriff ohne den Schutz einer Firewall gegen verschiedene Systeme durchgeführt. Während auf einem Windows 95 SP1 und einem Windows NT 4.0 SP6a System, der Angriff erfolgreich war, konnte die Verfügbarkeit des Opfersystems www3 zunächst nicht gestört werden. Hintergrund ist der in Windows 2000 enthaltene Schutz gegen diese Attacke. Dieser Schutz kann aber deaktiviert werden, in dem im Registry-Schlüssel „\HKLM\System\CurrentControlSet\TcpIp\Parameters“ der Wert „SynAttackProtect“ vom Typ DWORD erzeugt wird, dessen Datenteil den Wert 0 zugewiesen wird. In diesem Fall verhält sich Windows 2000 wie die übrigen betrachteten Systeme. Das Ergebnis einer SYN-Flood-Attacke wird in Abbildung 72 gezeigt.

[93] Weitere Details zum verwendeten Exploit sind dem vertraulichen Anhang K zu entnehmen.

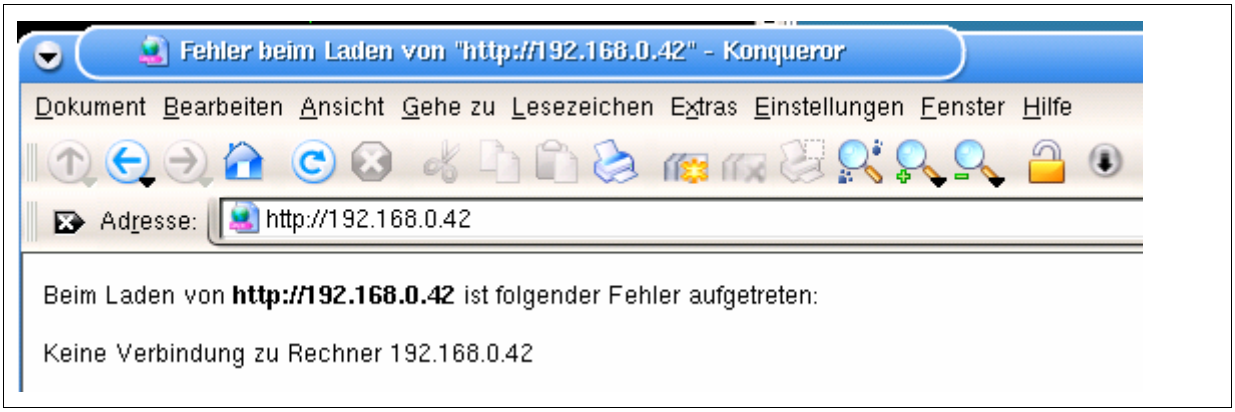


Abbildung 72: Ergebnis des SYN-Floods

Die Checkpoint Firewall-1 bietet zwei Möglichkeiten, den SYN-Flood-Schutz zu aktivieren. Die erste Möglichkeit ist die Aktivierung des SYN-Defenders, der in den Eigenschaften des „Network Objects“ der Firewall aktiviert werden kann. Zum anderen können seine Eigenschaften durch die „SYN-Attack Protection“ von SmartDefense aktiviert werden, wobei die Einstellungen des SYN-Defenders überladen werden. In diesem Versuch wurde die letztere Variante verwendet. Dabei konnte die SYN-Attacke gegen ein verwundbares Windows 2000 System erfolgreich abgewehrt werden. Im Log wurde der Eintrag „SYN Defender: Under SYN attack – Switching to active protection“ vermerkt.

7.6. Auswertung

Durch eine auf Applikationsebene filternde Firewall können Angriffe wirksam abgehalten werden. Das Vertrauen in eine solche Firewall kann aber nur dann bestehen, wenn die Firewall über entsprechende Funktionalitäten wie eine Längenbegrenzung verfügt. Mit der im Linux Kernel integrierten Firewall, die mit iptables konfiguriert wird, kann Angriffen auf Anwendungsschicht nicht entgegengewirkt werden.

Damit diese Funktionalitäten das Risiko reduzieren können, ist die Firewall richtig zu konfigurieren. So konnte bei der Durchführung des Penetrationstests unter Verwendung der Standardkonfiguration gezeigt werden, dass die reinen Funktionalitäten einer Firewall nicht ausreichen. Somit ist auch eine Zuversicht, dass die Firewall richtig konfiguriert ist und so die Risiken reduziert, notwendig. Sie kann durch einen Penetrationstest geschaffen werden. Der Grad der Zuversicht definiert das Vertrauen in die Firewall.

Auch eine hochwertige Firewall kann alleine keine Sicherheit bieten. Durch Würmer, die durch Emails die Firewall umgehen, und Angriffe, die zum Entwicklungszeitpunkt der Firewall noch nicht bekannt waren oder aus dem internen Netz initiiert werden, können Systeme im internen Netz trotz Firewall geschädigt werden. Somit ist ein Sicherheitskonzept notwendig, das weitere Sicherungsmaßnahmen wie das Hardening der Systeme beinhaltet.

Die Wirksamkeit einer Firewall gegen die Methoden der Reconnaissance in einer realen Umgebung ist Teil des nächsten Szenarios. Dabei wird im nächsten Kapitel eine Reconnaissance durchgeführt, die das Netz des Fachbereichs Informatik betrachtet.

8. KAPITEL:

SZENARIO RECONNAISSANCE

In diesem Kapitel werden die in Abschnitt 5.3 dargestellten Methoden der Reconnaissance auf das gesamte Netzwerk des Fachbereichs Informatik der Universität Hamburg angewendet. Ziel dieses Versuchs ist die Überprüfung der Wirksamkeit des gesamten Sicherheitskonzepts des Fachbereichs Informatik gegen die Schritte eines externen Angreifers in der Phase der Reconnaissance.

8.1. Vorbereitung

Vor Beginn der Reconnaissance wurde ein Service Level Agreement erstellt. Dabei wurde festgelegt, dass ein Angreifer simuliert werden soll, der über das Internet versucht, in das Netz des Fachbereiches Informatik der Universität Hamburg einzudringen. Zur Vermeidung eventueller Schäden soll lediglich die Reconnaissance durchgeführt werden, um festzustellen, welche für einen Einbruch notwendigen Informationen durch den Angreifer erlangt werden können. Zeitliche Aspekte sollen nicht beachtet werden. So könnte ein Angreifer seine Aktivitäten über Wochen verteilen, um in den Logs nicht aufzufallen.

Die Reconnaissance wird von einem privaten ADSL-Anschluss aus durchgeführt. Zur Kontrolle der Aktivitäten wurde vereinbart, dass vor jeder Aktivität eine Email an Herrn Mück geschickt wird, in der die verwendete IP-Adresse angegeben ist.

Die Ergebnisse der Reconnaissance sind in diesem Kapitel nur zusammengefasst dargestellt, wobei Details geändert und abstrahiert wurden, um einen Missbrauch zu vermeiden. Genauere Ergebnisse sind im vertraulichen Anhang K enthalten, dessen Inhalt nur dem Autor und den Betreuern bekannt ist.

Das geschlossene Service Level Agreement, das die Rahmenbedingungen der im nun folgenden Abschnitt erörterten Durchführung festlegt, ist der Arbeit im Anhang J beigelegt.

8.2. Durchführung

In diesem Abschnitt werden die in Abschnitt 5.3. erläuterten Methoden der Discovery und der Enumeration angewendet. Begonnen wird die Reconnaissance mit der öffentlichen Informationsbeschaffung.

Öffentliche Informationsbeschaffung

In der öffentlichen Informationsbeschaffung fiel bei einem Rundgang durch die Gebäude des Fachbereiches Informatik ein Aushang im Foyer des Hauses D auf. Dieser Aushang stellt die Anordnung der Firewall und der wichtigsten Router dar, sowie die Adressierung der Subnetze in den einzelnen Häusern. Ein potentieller Angreifer erhält durch diese Informationen einen groben Überblick über das Netzwerk am Fachbereich. Zwar werden keine IP-Adressen der Systeme angegeben, bedenklich an diesem Aushang ist jedoch die zum Teil genaue Typenbeschreibung der Router. Kann der Angreifer durch die Kombination der weiterer Ergebnisse, die er in späteren Phasen erlangt, eine Zuordnung von IP-Adresse und Typbezeichnung vornehmen, so kann er sich über Schwachstellen informieren und diese eventuell ausnutzen.

Um den ausgehängten Netzplan zu erhalten, muss ein Angreifer keinen physikalischen Weg nutzen. Er ist ebenfalls in einem im Internet verfügbaren Vortrag enthalten.

Whois

Bekannt ist der Name www.informatik.uni-hamburg.de . Mit der in Abbildung 25 auf Seite 86 dargestellten Anfrage, bei der eine whois-Anfrage mit der IP-Adresse des Host www.informatik.uni-hamburg.de an den Host rr.arin.net gestellt wird, wurde ermittelt, dass der Rechner zu dem „Universitaet Hamburg campus net“ gehört. Das Campusnetz besitzt den Adressbereich 134.100.0.0/16 und beinhaltet alle an der Universität Hamburg vorhandenen Rechner. Zusätzlich können durch eine Whois-Anfrage an den für die .de-Domains zuständige DeNIC weitere Informationen über die Verwalter herausgefunden werden (vgl. Abbildung 24 auf Seite 85).

Welche der Rechner des Netzes 134.100.0.0/16 zu der Domain informatik.uni-hamburg.de gehören, soll mittels spezieller DNS-Abfragen herausgefunden werden. Die DNS-Anfragen werden im nächsten Abschnitt betrachtet.

DNS

Zunächst wurde der autorisierte Nameserver für die Zone durch eine DNS-Abfrage vom Typ ns an den Webserver www.informatik.uni-hamburg.de herausgefunden. Um nun die Systeme der Domain informatik.uni-hamburg.de ausfindig zu machen, besteht die Möglichkeit einen Zonentransfer (vgl. Seite 81) durchzuführen. Da der Port 53/tcp am Nameserver nicht geöffnet ist, war dies jedoch nicht möglich. Um dennoch einen Zonentransfer durchzuführen, könnte ein anderer Nameserver im Internet gesucht werden, der einen Zonentransfer zulässt. Dabei würde aber eine Schwachstelle eines Systems gesucht werden, das nicht zu der Domain informatik.uni-hamburg.de gehört, was durch das SLA ausgeschlossen ist.

Um dennoch festzustellen, welche Systeme zu dem Netz 134.100.0.0/16 gehören, wurden DNS-Reverse-Lookups, die eine IP-Adresse in den zugehörigen DNS-Namen übersetzen, durchgeführt. Für diese Aufgabe wurde mit dem shell-Skript revinf, das im Anhang G6 enthalten ist, ein Reverse-Lookup für alle IP-Adressen des Netzes 134.100.0.0/16 durchgeführt. Die Ergebnisse wurden in die Datei logs/output.txt geschrieben, aus der

sämtliche Systeme der Domain informatik.uni-hamburg.de mit dem Befehl grep herausgefiltert und in der Datei dns.in gespeichert wurden.

Durch die Reverse-lookup-Anfragen wurden ca. 1500 Systeme erkannt. Bedenklich dabei ist, dass alle Systeme inklusive Router, Arbeitsstationen, Notebooks, Drucker und sogar Testsysteme im DNS eingetragen sind. Für einen Angreifer sind dies interessante Informationen. Auch wenn der Angreifer die Informationen bei Aktionen von außerhalb des Fachbereiches Informatik nicht nutzen kann, sind sie dennoch vorteilhaft, wenn der Angreifer in einen Rechner des Fachbereiches eingedrungen sein sollte.

Eine weitere interessante Methode bei einem DNS-Lookup sind die Resource Records HINFO und TXT, bei denen Administratoren beliebige Informationen zu den Systemen eintragen können. Zwar sollte jedem Administrator mittlerweile bewusst sein, solche Einträge nicht mehr zu nutzen. Es sollte jedoch auch bei bekannten Fehlern überprüft werden, ob die Gegenmaßnahmen befolgt werden.

Zur Abfrage der HINFO und TXT Felder wurden die Perl-Skripte hinfo.pl und txt.pl, deren Quelltext in Anhang G enthalten ist, benutzt. Wie erwartet, brachte die Abfrage des Ressource Records TXT bei allen Systemen keine Antwort. Allerdings enthielt der Ressource Record HINFO bei ca. 600 Systemen Daten wie „PENTIUM-III-500“ „Windows 2000“, „HP-LaserJet-4M/Plus" "HP_LJ4" oder „Cisco“ „7200“[94]. Auf diese Weise braucht das Betriebssystem und die Art des Rechners bei den betroffenen Systemen nicht erraten werden.

Erkennung der Netzstruktur

Nachdem einige Systeme erkannt worden sind, soll mit den Methoden der Reconnaissance ein Netzplan des Arbeitsbereiches AGN erstellt werden. Dazu soll mit dem Tool Traceroute sowohl ein ICMP-Traceroute als auch ein UDP-Traceroute (vgl. Seite 82) durchgeführt werden. Problematisch ist allerdings, dass die in einer Linux-Distribution enthaltene Variante die UDP-Portnummer bei jedem Knoten inkrementiert. Um dies zu verhindern, wurde eine modifizierte Traceroute-Variante der Version 1.4.a13[95] mit dem Schalter -S verwendet, durch den das Inkrementieren der Port-Nummer verhindert wird.

Unabhängig von der Variante ist das Ziel der Traceroutes, die Lage der Systeme rzdspc1, rzdspc2, rzdspc3, rzdspc77 sowie der Systeme des Arbeitsbereiches AGN zu bestimmen, wobei ICMP, Standard UDP sowie ein Traceroute mit nicht-alternierendem Port 53/udp verwendet wurden.

Die Systeme rzdspc1 und rzdspc77 antworteten bei Verwendung der ICMP-Variante nach dem atm-Router atm-informatik.informatik.uni-hamburg.de jeweils drei Mal. Demnach müssten die Netzwerkpakete zwei Schleifen von rzdspc1 zu rzdspc1 durchlaufen, um an ihr Ziel zu gelangen. Alle anderen Traceroutes erhielten nach dem atm-router keine Antworten. Diese Umstände lassen darauf schließen, dass nach dem atm-Router eine Firewall installiert ist, die ausgehende ICMP TIME_EXCEEDED-Nachrichten (Typ 11) verwirft. In den

[94] Die Daten wurden absichtlich geändert.
[95] Der Tarball auf dem Server ftp.ee.lbl.gov trägt unverständlicherweise die Versionsnumme 1.4a12. Das Programm meldet sich jedoch mit 1.4a13 .

speziellen Fällen wie rzdspc1 und rzdspc77 wird eine Maßnahme getroffen worden sein, welche die Informationen verfälscht und so einen Angreifer verwirrt. Dieses in Abbildung 73 dargestellte Verhalten wurde auch bei einer TCP-Variante von Traceroute unter Verwendung von Port 80/tcp festgestellt.

```
# traceroute –f 13 –I rzdspc1.informatik.uni-hamburg.de
traceroute to rzdspc1.informatik.uni-hamburg.de (134.100.9.61), 30 hops may, 40
bytes packets
13 atm-informatik.rrz.uni-hamburg.de (134.100.38.230) 55  ms  123 ms  91 ms
14 rzdspc1.informatik.uni-hamburg.de (134.100.9.61)  101 ms  117 ms  97 ms
15 rzdspc1.informatik.uni-hamburg.de (134.100.9.61)  185 ms  78 ms *
16 rzdspc1.informatik.uni-hamburg.de (134.100.9.61)  2782 ms  1775 ms  713 ms
```

Abbildung 73: Dreimaliges Antworten eines Servers bei Anwendung von traceroute

Die Verfälschung ist auch in einer Analyse des Netzverkehrs zu beobachten. So schicken die Hosts 14 und 15 eine ICMP-TIME_EXCEEDED Nachricht, während der wahre Host mit einem ICMP-ECHO_REPLY antwortet. Zudem haben die TTL-Felder der Antworten den Wert 50 im Falle des Host 14 bzw. 51 im Falle des Host 15, während die TTL-Felder der von Host 16 generierten Pakete den Wert 240 aufweisen. Dies bedeutet, dass die Hosts 14 und 15 mit der TTL 64 starten, der Host rzdspc1 allerdings mit der TTL 255. Da die TTL des Host 15 genau um 1 höher ist als die des Host 14, sind es zwei verschiedene Hosts zwischen rzdspc1 und dem ATM-Gateway, die allerdings nicht erkennbar sind. Daher kann angenommen werden, dass die Traceroutes von einer Firewall verfälscht werden.

Die Traceroutes konnten nicht zur Erstellung eines Netzplans verwendet werden. Ein Angreifer könnte sich lediglich mit dem im Foyer ausgehängten Netzplan einen groben Überblick verschaffen. Um weitere Informationen über die Lage der Systeme zu bekommen, müsste der Angreifer in ein System eindringen. Da das Eindringen in ein System das Ziel des Angreifers ist, wird er an dieser Stelle versuchen, einen Zugangspunkt zu finden. Dazu führt er ein Scan einzelner Systeme durch.

Scan nach bestimmten Systemen

Von dem Angreifer wird angenommen, dass er sich auf die in den vorherigen Kapiteln besprochene Angriffsarten spezialisiert hat. So möchte er wissen, ob es Systeme mit den folgenden Eigenschaften im Netz des Fachbereichs Informatik gibt:

- Systeme, auf denen ein Webserver (Port 80/tcp) installiert ist
- Systeme mit installiertem SNMP-Agenten (Port 161/udp)
- Systeme, die Zugriff über SMB (Port 139/tcp und Port 445/tcp) bieten
- Systeme, auf die mittels Telnet (Port 23/tcp) zugegriffen werden kann

Für die Scans wurde das Tool nmap verwendet, wobei bei TCP-Ports ein Stealth-Scan durchgeführt wurde. Die Scantypen sind auf Seite 93 näher beschrieben.

Banner Grabbing erkannter Webserver

Beim Scan der Webserver wurden ca. 25 Webserver gefunden, für die ein Banner Grabbing durchgeführt wurde. Dabei wurden überwiegend Apache und wenige Microsoft Systeme erkannt. Der Status der Patches auf den Microsoft-Systemen ist nicht zu erkennen, so dass durch das Banner keine Evidenzen für anwesende Schwachstellen erkannt wurden. Zur Behebung von Schwachstellen des Apache-Webservers werden keine Patche, sondern ein Versioning benutzt. So werden mit jeder neuen Version des Apache-httpd Schwachstellen behoben. So ist zum Beheben der Schwachstellen eine neue Version der Software zu beziehen und zu installieren[96]. Dies stellt allerdings kein Problem dar, da Open-Source Software frei verfügbar ist.

Auf Grund des Versioning können bei Apache-Webservern anhand des letzen Teils der Versionsnummer auf die Anwesenheit bekannter Schwachstellen geschlossen werden. Bei dem im Kontext des Fachbereiches Informatik durchgeführten Banner Grabbing wurde festgestellt, dass nur drei Systeme dem aktuellen Stand entsprechen. In einem extremen Fall wurde eine viereinhalb Jahre alte Apache-Version eingesetzt.

Auch melden manche Server die Versionen der geladenen Module sowie die Versionen der zugrunde liegenden Software. Wie auch die ISAPI-Erweiterungen des IIS-Webservers können auch sie Schwachstellen enthalten. So wurden zwei Webserver erkannt, die von dem Wurm Linux/Slapper.worm befallen werden können. Der Wurm benutzt eine Schwachstelle in OpenSSL <0.9.6e (vgl. [CA-2002-27]), das vom mod_ssl des Apache benutzt wird. Zudem erfüllte ein Webserver keinen universitären Zweck.

Neben den 25-Systemen, deren Ports beim Portscan als offen gemeldet wurden, meldeten - einige aber nicht - alle Router den Port 80/tcp als geschlossen. Alle restlichen Systeme zeigten für Port 80/tcp einen gefilterten Zustand an.

Systeme, die den SMB-Dienst nach außen hin anbieten, wurden nicht gefunden. Gleiches gilt auch für den Dienst SNMP. Der zugehörige Port 161/udp wurde von nmap bei allen Systemen als offen erkannt. Dies bedeutet aber nicht, dass der Port offen ist, sondern dass nmap keine Antwort empfangen hat. Auch wenn eine Firewall alle Pakete verwirft, so werden die gescannten Ports von nmap als offen gemeldet.

Um zu prüfen, ob die Ports wirklich offen sind, oder eine Firewall die Pakete verwirft, wurde mittels eines Perlskript versucht, von jedem System die Systembeschreibung der community „Public“ auszulesen (vgl. Abbildung 29, Seite 86). Das Skript ossnmp.pl, dessen Quellcode in Anhang G9 enthalten ist, benutzt dazu das Tool snmputil, weshalb der Scan unter Windows ausgeführt wurde. Während das System www3 bei einem Test des Scripts im Testnetz (vgl. Abschnitt 6.3) korrekt antwortete, hatte die Anfrage bei allen in DNS eingetragenen Systemen des Fachbereichs eine Fehlermeldung zur Folge. Somit konnten keine Informationen durch SNMP-Anfragen an die Community public gewonnen werden.

Der Scan nach offenen Telnet Ports hat zwei Ziele. Zum einen kann ein interaktiver Zugang erlangt werden, wenn ein Benutzername und ein Passwort bekannt sind. Das Passwort kann

[96] Bezieht sich auf Binaries. Für Source Code gibt es Patches in Form von diff-Dateien.

dabei erraten und, da es im Klartext übertragen wird, abgehört werden. Zum anderen werden in den Begrüßungsbotschaften häufig Informationen über die Systeme verraten.

In der Begrüßungsbotschaft zweier Server wurde auf Grund der unverschlüsselt übertragenen Passwörter vorbildlich davon abgeraten, Telnet zu benutzen. Das Betriebsystem wurde dennoch bekannt gegeben, wobei aber nur der Name und nicht die Versionsnummer angegeben ist.

Ebenso wie Telnet bieten auch SSH-Server die Möglichkeit eines interaktiven Zugangs. Allerdings werden hierbei Begrüßungsbotschaften erst nach einem erfolgreichen login angezeigt. Da das Erraten von Benutzernamen und Passwörtern kein Bestandteil der Reconnaissance ist, wird an dieser Stelle kein Scan nach ssh-Systemen durchgeführt.

Scan bestimmter Webserver

Aus den beim http-Scan gewonnenen Ergebnissen würde ein Angreifer bestimmte Webserver bestimmen, die ihm am verwundbarsten erscheinen. Bei den Apache-Webservern ist auf Grund des Versionings an Hand der Versionsnummer bestimmbar, welche Schwachstellen in der installierten Version enthalten sind. Das Banner eines IIS-Webservers gibt allerdings keine Auskunft über vorhandene Schwachstellen oder anwesende Patche.

Für die Betrachtung, welche Risiken für einen bestimmten Webserver existieren, ist als erstes zu bestimmen, welche Dienste durch den Server angeboten werden. Jeder weitere Dienst erhöht das Risiko eines erfolgreichen Angriffes. Dazu wird nach den Zugangspunkten des Dienstes auf einem System, den Ports, gescannt.

Auf Grund des hohen Zeitaufwands wurden beispielhaft sechs Systeme gescannt. Hierbei handelt es sich um vier Systeme, deren Banner erhebliche Schwachstellen erkennen lassen. Des Weiteren werden beispielhaft zwei Systeme gescannt, auf denen IIS-Webserver installiert sind.

Zur Durchführung wurden mit dem Tool nmap, das als Eingabe eine Liste der ausgewählten IP-Adressen erhielt, alle TCP-Ports gescannt. Bei manchen Systemen brach nmap den Scan mit der Meldung „Skipping host … due to host timeout“ ab. In diesem Fall lieferte nmap keine Ergebnisse, wobei auch der Port 80 des Webservers nicht als offen gemeldet wurde. So kann darauf geschlossen werden, dass Portscans durch die Firewall unterbunden werden. Trotz Unterbindung des Portscans wird der Angreifer nicht blockiert. Allerdings können bestimmte Systeme ohne Probleme gescannt werden. Ein sehr langsamer Portscan allerdings liefert für jedes System in jedem Fall die korrekten Ergebnisse.

Bei Auswertung der Ergebnisse ist auffällig, dass einige Systeme von Außen nur Zugriff auf den Webserver und gegebenenfalls auf einen FTP-Server bieten. Auf manchem Server jedoch ist eine Vielzahl von Ports offen. Da bereits der Webserver Schwachstellen enthält, ist zweifelhaft, ob die bekannten Schwachstellen anderer Dienste behoben sind. Daher ist der Umstand, dass auf die Ports zugegriffen werden kann, sehr bedenklich. Bei den zusätzlich angebotenen Diensten handelt es sich unter anderem um Dienste zu Remote-Administration oder RPC-Dienste.

Scan weiterer Systeme

Auf Grund der Einträge im DNS sollen auch Workstations, Testsysteme und Drucker betrachtet werden. Mittels eines Ping-Sweeps, der ebenfalls zu den Funktionalitäten des Tools nmap gehört, wurde zunächst geprüft, ob einzelne Systeme im Netz verfügbar sind. Antworten auf die Pings wurden nur von Servern, die auch einen öffentlich zugänglichen Dienst anbieten, sowie von wenigen Routern erhalten.

Für den Scan wurden kategorisch einige Systeme ausgewählt. Zwar konnten keine offenen Ports erkannt werden, jedoch ist auffällig, dass nur spezielle, wie die für den SMB-Dienst benötigten Ports gefiltert werden. Alle anderen Ports sind geschlossen.

8.3. Auswertung

Positiv ist zu vermerken, dass ein Angreifer keinen Zonentransfer durchführen kann. Dieser positive Effekt wird allerdings durch den Umstand wettgemacht, dass alle Systeme im DNS eingetragen sind und so durch Reverse-Lookups ein dem Zonentransfer gleichwertiges Ergebnis erzielt werden kann.

Durch die Firewall sind sämtliche Systeme verdeckt. Die Verfügbarkeit kann mittels eines Pings nur bei solchen Systemen geprüft werden, die externen Benutzern einen Dienst anbieten. So kann auch durch das Verbot der zum Zeitpunkt der Durchführung der hier besprochenen Reconnaissance keine positive Wirkung erzielt werden. Dabei sollte aber geprüft werden, ob die Konformität zu [RFC792] durch die Firewall geprüft wird, um Angriffe wie beispielsweise den Ping-of-Death zu vermeiden. Auch können mittels Traceroutes keine Informationen über die Netzstruktur erlangt werden.

Die Portscans werden zum Teil unterbunden. Positiv dabei ist, dass der Initiator des Portscans nicht geblockt wird. Manche Programme, die einen Portscan unterbinden sollen, prüfen lediglich, ob ein bestimmter, bei normalem Verkehr nicht angesprochener Port ein Paket erhält. Findet ein Angreifer diesen Port, so kann in Verbindung mit Spoofing ein Denial-of-Service erreicht werden. Wenn sogar die Adresse eines Routers vor der Firewall gefälscht wird und dieser kein ingress-Filtering durchführt, so kann ein Angreifer ein Netzwerk von sämtlichen Kommunikationspartnern abschneiden.

Neben den fehlenden Updates der Webserver ist auch bedenklich, dass bei einigen Webservern sehr viele Ports geöffnet sind. Hierbei sollte geprüft werden, ob die an den Port gebundenen Dienste wirklich benötigt werden und ob die Firewall ihre gewünschte Wirkung erfüllt.

Die öffentlich verfügbare Information der groben Netzstruktur kann einem Angreifer bei der Reconnaissance helfen. Bei einer Universität ist dies aber unbedenklich, da zum einen die Assets keinen hohen Schutzbedarf erfordern, zum anderen solche Informationen auch als Beispiel für die Lehre dienen können.

Neben der Überprüfung der Filterwirkung hinsichtlich der betrachteten Webserver, sollten auch die Webserver vor Ort auf Schwachstellen überprüft werden. Wird dies unterlassen, so

besteht erstens die Gefahr eines Angriffes durch einen Angreifer sowie zweitens die Gefahr eines von einem Wurm verursachten Vorfalls.

An dieser Stelle endet die Durchführung der drei Szenarien. Mit dem nun folgenden letzten Kapitel wird die Arbeit durch eine Schlussbetrachtung abgeschlossen.

9. KAPITEL:

SCHLUSSBETRACHTUNG

In diesem Kapitel werden die Möglichkeiten und Grenzen des Penetrationstests zusammenfassend erläutert. Diese Betrachtung bildet den Abschluss der Arbeit, die mit einem Fazit über die Verwendung der Methode Penetrationstest endet.

9.1. Möglichkeiten

Die Möglichkeiten des Penetrationstests sind abhängig von dem Anwendungsgebiet. So können mittels des Penetrationstests in der Softwareentwicklung neue Schwachstellen entdeckt werden. Auch kann der Penetrationstest zur Schulung von Incident Response Teams genutzt werden.

Schwerpunkt der Betrachtung in dieser Arbeit ist die Anwendung des Penetrationstests innerhalb einer Revision zur Überprüfung eines Sicherheitskonzeptes. Daher wird an dieser Stelle genauer auf die Möglichkeiten eines Penetrationstests bei der Revision eingegangen.

Durch die Durchführung von Angriffen kann ein Penetrationstest im Falle eines durch ihn erzeugten Vorfalls beweisen, dass ein System kompromittiert werden kann. In diesem Fall kann gezeigt werden, dass Maßnahmen fehlen oder fehlerhaft implementiert sind. Bei einer strukturierten Vorgehensweise kann auch gezeigt werden, dass ein System einer Bedrohung widersteht. Auch kann gezeigt werden, dass eine Maßnahme wie eine Firewall oder ein Intrusion Detection System fehlerfrei implementiert ist und effektiv vor den Bedrohungen schützen, in dem die Bedrohungen auf die Maßnahme angewendet werden (vgl. Abschnitt 3.4.3 und Kapitel 7).

Besonders geeignet ist der Penetrationstest im Fall der Überprüfung, ob menschliche Schwachstellen behoben worden sind. So können Mitarbeiter zur Vorbeugung von Social Engineering Angriffen zwar durch Schulungen dazu bewegt werden, gewisse Regeln bei Telefongesprächen zu befolgen. Ob die Mitarbeiter die Maßnahmen einhalten, ist jedoch fraglich. So können die Verantwortlichen die Mitarbeiter zwar befragen, ob sie die Maßnahmen auch einhalten. Die Antworten sind aber nur durch die Ausübung einer Social Engineering Attacke im Rahmen eines Penetrationstests zu überprüfen. Ähnliches gilt für die Verwendung starker Passwörter, wenn diese Maßnahme nicht durch technische Systeme erzwungen werden kann.

Das Aufzeigen der Widerstandsfähigkeit der Systeme sowie der Wirksamkeit von Maßnahmen schafft eine Zuversicht, dass den Risiken durch ein bestehendes Sicherheitskonzept effektiv entgegengewirkt wurde, wodurch das Vertrauen der Kunden, Manager und Anteilseigner in die Organisation gebildet oder gestärkt wird. Dies verbessert die Reputation

und schafft einen Wettbewerbsvorteil für ein Unternehmen, welches den Penetrationstest initiiert, so dass die Marktposition eines Unternehmens gestärkt wird (vgl. [Higgins01:3]). Dadurch trägt der Penetrationstest auch zur Sicherung der Überlebensfähigkeit im Markt bei.

Im Falle eines durch den Penetrationstest erzeugten Vorfalls können die Risiken und damit das Ausmaß des Schadens demonstriert werden. Dies kann in der Kontrollphase zu einem „Wachrütteln des Managements“ [Lessing98:16] führen, wenn ein gezeigter Schaden und dessen Ausmaß nicht erwartet worden ist. Dieser Effekt kann bereits in der quantitativen Risikobewertung genutzt werden, bei der Schäden und Wahrscheinlichkeiten abgeschätzt werden. Die Abschätzung leidet häufig an fehlenden empirischen Daten, welche durch einen Penetrationstest gesammelt werden können. So kann der Penetrationstest auch die Entscheidung über die Zusammensetzung des Maßnahmenportfolios im Falle einer Unsicherheit unterstützen, indem Risiko und Wirksamkeit der Maßnahme demonstriert werden. Die Demonstration des Risikos kann auch dazu verwendet werden, die Notwendigkeit eines Sicherheitskonzeptes sowie den Aufwand der Erstellung eines solchen zu rechtfertigen.

Weiter unterstützt der Penetrationstest die Beteiligten des Sicherungsprozesses, die Risiken zu verstehen. „[…] it surprises me how much we as security professionals know about defense but how little we sometimes know about what hackers are doing“ [Payne01:1]. So spielen Administratoren meist nur einen Patch ein, ohne die eigentlichen Auswirkungen der Bedrohung zu kennen (vgl. [Payne01:2]). Beim Web Folder Traversal Angriff konnte gezeigt werden, dass eine ordentliche Rechtevergabe auch vor dem Angriff schützt. Ist ein Administrator, der lediglich Patches einspielt, am 14. Mai 2001 der Annahme gewesen, das Service Pack 2 für Microsoft Windows 2000 würde alle bis zu diesem Tage bekannten Schwachstellen beheben, wäre sein System erneut gegen bestimmte Instanzen des Web Folder Traversal Angriffes verwundbar (vgl. Abschnitt 6.9.2). So versuchen die Administratoren metaphorisch gesehen immer noch die Türen zu sichern, haben aber keine Maßnahmen, die den Angreifer nach dem Durchbrechen einer Tür am Fortkommen hindern. Hierbei ist dringend zu empfehlen, das Defense-in-Depth Prinzip anzuwenden, was ein Verständnis der Sicherung jenseits des Patchmanagements erfordert. Dieses Verständnis kann ein Administrator bei der Durchführung eines Penetrationstests gewinnen. Ebenso kann er geschult werden, die Logs zu verstehen und richtig zu deuten. Weiter kann durch die Durchführung einer Reconnaissance die Notwendigkeit erkannt werden, einen Angreifer bereits in der frühesten Phase des Angriffes zu stoppen und daraufhin Gegenmaßnahmen zu entwickeln. So können Risikoanalyse und -management durch die im Penetrationstest gewonnenen Kenntnisse verfeinert werden.

Durch das Verständnis der Vorgehensweise eines Angreifers kann die Maßnahmenauswahl effektiver erfolgen, als bei einer reinen Schwachstellenanalyse, bei der eventuell Schwachstellen übersehen werden, da sie nicht in den Schwachstellenarchiven beschrieben sind. Solche Schwachstellen sind zum Beispiel falsche Rechte oder installierte Software, die einem Angreifer bei der Fortführung seines Angriffes helfen. Durch Maßnahmen wie ein effektives Hardening eines Servers, das beispielsweise in Abschnitt 6.9 beschrieben ist, kann auch 0-day Exploits besser vorgebeugt werden, bei denen der Exploit schon am Tage des Bekanntwerdens der Schwachstelle verfügbar ist, ein Patch zu diesem Zeitpunkt jedoch fehlt.

Weiter können durch das Verständnis der Vorgehensweise auch Irrtümer wie die Annahme, die Firewall schütze vor allen Angriffen aufgezeigt und demonstriert werden. So ist nach [SANS99] einer der Top 7 Fehler des Managements, dass ein Sicherheitskonzept hauptsächlich auf der Firewall basiert.

9.2. Grenzen

Der Penetrationstest ist unter anderem durch die Qualität des Tiger Teams begrenzt. „A penetration test is only as good as the people conducting it" (vgl. [Kurtz00:5]). Die Qualität des Tiger Teams ist hinsichtlich der technischen und ethischen Kompetenz zu differenzieren.

Die technische Kompetenz eines Tiger Teams ist vor allem durch das Hintergrundwissen charakterisiert, das für die Durchführung der Angriffe benötigt wird. So kann ein Hacker nicht unbedingt die technische Expertise vorweisen. Er kennt eventuell nur wenige Exploits, die er nur begrenzt einsetzen kann. Die Aussagekraft eines Hackers oder sonstigem ungeschulten Personal, ein System sei sicher, da nicht in das System eingebrochen werden konnte, ist sehr begrenzt. Begründet ist die Begrenzung durch die Möglichkeit, dass ein anderer Hacker in das System einbrechen kann.

Nur wenn der Penetrationstest mit einer systematischen Vorgehensweise durchgeführt wird, kann eine ausreichende Aussagekraft über die Sicherheit eines Systems erzielt werden. Eine strukturierte Vorgehensweise beginnt mit einer technisch orientierten Risikoanalyse, bei der für die betrachteten Systeme alle Schwachstellen und die dazugehörigen Bedrohungen, deren Instanzen die Exploits sind, ermittelt werden. Schon bei diesem Schritt ist eine detaillierte Kenntnis der Risiken sowie der technischen Details der Systeme erforderlich, die nur durch eine fundierte Ausbildung im Gebiet der IT-Sicherheit erlangt werden kann. Nur wenn in der Durchführung alle Kombinationen von Schwachstellen und Bedrohungen sowie alle möglichen Angriffswege betrachtet worden sind, kann eine Zuversicht geschaffen werden, dass die Risiken minimiert sind. Diese Zuversicht führt zu einem gesteigerten Vertrauen in ein bestehendes Sicherheitskonzept.

Begrenzt ist eine systematische Vorgehensweise durch den hohen Aufwand. So wurde im Webserverszenario mit einer Schwachstellenanalyse begonnen, die Durchführung jedoch auf zwei Exploits begrenzt. Schon die Penetration dieser beiden Exploits verursachte einen Zeitaufwand von mehreren Wochen. Allerdings ist dieser Zeitaufwand durch die geringe Erfahrung hinsichtlich der Penetration seitens des Autors begründet. Dies untermauert das Argument, dass das Tiger Team nur aus erfahrenen Mitgliedern bestehen sollte. Zur Begrenzung des Aufwands können die Ziele des Penetrationstests genauer umrandet werden.

Da ein Tiger Team nur schwer zu kontrollieren ist, ist zusätzlich zu einem SLA auch das Vertrauen des Kunden in das Tiger Team notwendig, dass es neben der technischen Kompetenz auch über eine nötige ethische Kompetenz verfügt. Ethik ist nach [Duden5:220] allgemeingültig definiert als die „Normen […], die sich aus der Verantwortung gegenüber anderen herleiten". Ein Tiger Team muss sich vor allem der Risiken bewusst sein, die bei einem Penetrationstest entstehen können. Neben den Schäden an Daten und Systemen, die vom Penetrationstest betroffen sind, können auch rechtliche Konsequenzen folgen.

Im Falle eines entstandenen Schaden an den Daten und Systemen können Schuldzuweisungen gegen das Tiger Team entstehen (vgl. [Lessing98:17]). Der Schaden kann im Sinne des § 303b Strafgesetzbuch (StGB) auch dann verfolgt werden, wenn die Verantwortlichen der betreffenden Organisation eingewilligt haben. So können mögliche Nutznießer der Organisation den Schaden anzeigen. Ein Staatsanwalt ist in diesem Falle verpflichtet, der Anzeige nachzugehen. Zudem sind der Besitz, die Wartung und die Installation von Umgehungsvorrichtungen, mit denen ein gegen Entgelt erbrachter zugangskontrollierter Dienst genutzt werden kann, zu gewerblichen Zwecken nach § 3 Satz 1 Zugangskontrolldiensteschutz-Gesetz (ZKDSG, vgl. [ZKDSG02]) verboten. Zuwiderhandlungen werden als Ordnungswidrigkeit mit einer Geldbuße von bis zu 50.000,- € geahndet. Sollte die Gefahr der Geldbuße bestehen, sollte ein Tiger Team die Geldbuße in die Teams Kostenkalulation einfließen lassen.

Auch die ethische Kompetenz ist ein Argument gegen den Einsatz eines Hackers im Tiger Team. Unabhängig davon, ob er als black-hat oder als white-hat klassifiziert werden kann, sind die Angriffe seine einzige Intention. Verantwortlichkeiten sind einem Hacker fremd. Nur bei einer fundierten Ausbildung in der IT-Sicherheit, zu der auch die Behandlung von Gesetztestexten und ethischen Richtlinien gehören, kann eine rechtliche und ethische Kompetenz vorausgesetzt werden.

Unethisch sind ebenfalls die Angebote einiger Firmen, die Schulungen anbieten, in denen Teilnehmer wie Administratoren in wenigen Tagen das Hacken lernen sollen (vgl. [Higgings01:3 und [Kurtz00:2]). Eine fundierte Ausbildung ist auf Grund der Komplexität in der Kürze der Zeit nicht möglich. Zudem kann nicht ausgeschlossen werden, dass die erworbenen Kenntnisse von den Kursteilnehmern unverantwortlich eingesetzt werden. So können diese Kurse keine ausreichende technische und ethische Kompetenz zur Durchführung eines Penetrationstests vermitteln. Da zu den Leistungen solcher Firmen häufig auch Penetrationstests gehören, zeigt das Angebot eines „Hacker-Workshops“, dass die Firmen sich auch unverantwortlich gegenüber ihrer Geschäftsgrundlage verhalten. Sollten die Teilnehmer auf Grund der Schulungen den Penetrationstest wirklich selber durchführen können, so kann der Anbieter der Schulungen weniger Penetrationstests verkaufen.

Auch ist die Verwendung von Viren und Würmern im Rahmen eines Penetrationstests unethisch. Während in einem Penetrationstest durch ein SLA auf gewisse Systeme beschränkt werden kann, ist die Ausbreitung von Viren und Würmern nicht kontrollierbar. Auch in einem abgeschotteten Labor kann nicht gewährleistet werden, dass infektiöse Malware nicht Datenträger befallen hat, welche das Labor verlassen.

Die Absicherungen des Tiger Teams durch das SLA kommen aber erst bei entstandenem Schaden zur Wirkung. Damit ein Schaden gar nicht entstehen kann, ist eine Simulation sinnvoll. Zwar wird der Penetrationstest in der Literatur häufig als simulierter Angriff bezeichnet, jedoch werden auch in einer Simulation Angriffe durchgeführt. Simuliert werden können das Verhalten eines realen Angreifers sowie die von den Angriffen betroffenen Systeme. Letzteres bietet die Möglichkeit, Schäden im Rahmen eines Penetrationstests proaktiv zu vermeiden. Hierbei werden nicht reale Systeme, sondern Kopien der Systeme beispielsweise in einem Testnetz installiert sein, das Ziel der Penetration ist.

Die Simulation der Systeme hat aber ebenfalls Grenzen, welche die Ergebnisse des Penetrationstests beeinträchtigen können. Unter einer Simulation wird nach [Page00:1-27] die „Durchführung von Experimenten an einem Modell, das anstelle des Originalsystems tritt verstanden“. Nach einem Simulationsmodell von Brunnstein wird dabei ein Abbild von der Realität $\mathcal{R}$ geschaffen, durch das die charakteristischen Eigenschaften von $\mathcal{R}$ auf das Modell $\mathcal{M}_{\vartheta i}$ übertragen werden. Das Modell betrachtet dabei einen Ausschnitt der Realität unter dem Blickwinkel ϑ_i. Das Modell $\mathcal{M}_{\vartheta i}$ wird weiter durch das computerisierte Modell $\mathcal{C}_{\vartheta i}(\mathcal{M}_{\vartheta i})$ implementiert. Durch die Komplexität der Realität ist nicht zu gewährleisten, dass das in der Komplexität reduzierte computerisierte Modell $\mathcal{C}_{\vartheta i}(\mathcal{M}_{\vartheta i})$ der Realität $\mathcal{R}$ entspricht. So kann schon der Blickwinkel unzureichend gewählt werden, so dass charakteristische Eigenschaften der Realität verloren gehen können. Zudem können bei der Implementation Eigenschaften der Realität verloren gehen. In Abbildung 74 wird dieser Verlust durch kleiner werdende Mengenkreise dargestellt. So können bei einem in-vitro Versuch, der am lebenden Objekt durchgeführt wird, Seiteneffekte auftreten, die in einem computerisierten Modell auf Grund der Komplexitätsreduktion nicht beobachtbar sein können.

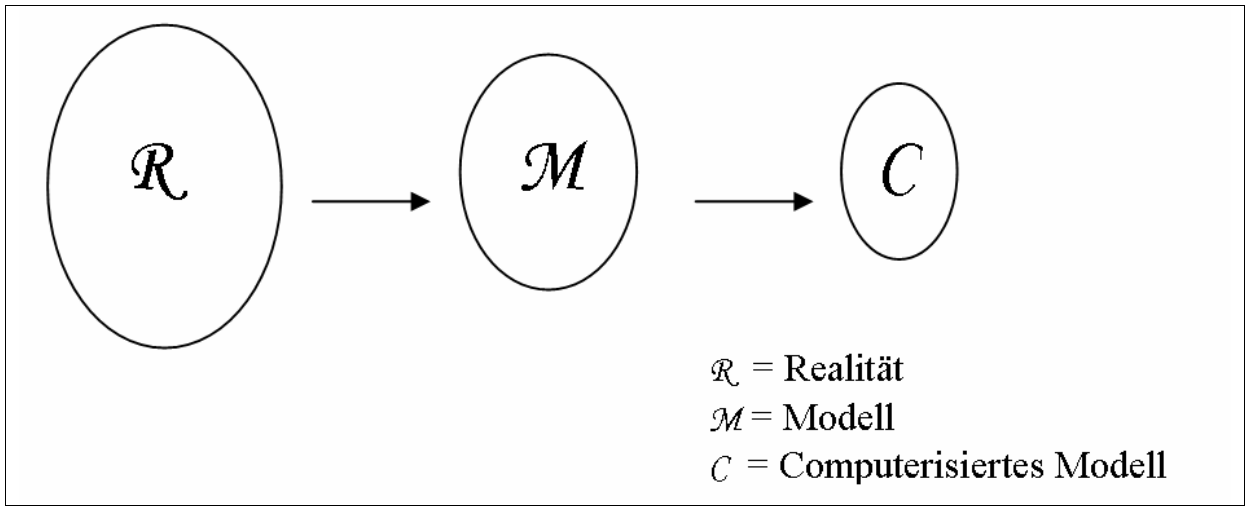


Abbildung 74: Simulationsmodell nach Brunnstein

Ein Beispiel für Seiteneffekte liefern ein Experiment von Shoch und Hupp (vgl. [Denning90:264ff.]) am XEROX Palo Alto Research Center (XEROX-PARC[97]) im Jahre 1982. Dabei entwarfen sie ein Programm, das sich selbstständig über das Netzwerk auf anderen Rechnern installieren konnte und Wurm genannt wurde. Dieses Programm gilt als der erste Wurm in der Geschichte. Neben dem Effekt, dass die Rechner bei der Infektion abstürzten, wurde in einem Fall beobachtet, dass der Wurm auf eine Festplatte zugriff, obwohl er keine Zugriffroutine enthalten durfte (vgl. [Denning90:267]). Auf Grund solcher Beobachtungen kann nicht ausgeschlossen werden, dass bei einer in-vitro Penetration eines Systems auch andere Systeme im Netzwerk auf Grund wenig bekannter Abhängigkeiten Fehlfunktionen aufweisen, die bei einer Simulation nicht erkannt werden würden.

Bei der Durchführung eines Penetrationstests ist somit zu wählen zwischen einer in-vitro Penetration oder einer Simulation der Systeme. Dabei ist bei der in-vitro Penetration immer mit Schäden an Systemen und Daten zu rechnen. Bei einer Simulation besteht die Gefahr,

[97] Seit geraumer Zeit nur noch PARC, siehe www.parc.xerox.com

dass Seiteneffekte übersehen werden. Die Entscheidung für eine in-vitro Penetration oder eine Simulation sollte aber von den Zielen des durchzuführenden Penetrationstests abhängen.
Auch ergeben sich Begrenzungen aus dem Standort des simulierten Angreifers. Wenn es keine Schwachstelle gibt, über die ein Tiger Team eindringen kann, so können die Auswirkungen eines eventuellen Einbruches nicht gezeigt werden. Um diesen Mangel zu beheben, sollte der Einbruch simuliert werden. Dies kann beispielsweise geschehen, wenn der Penetrationstest von innen durchgeführt wird.

Die Erkennung der Schwachstellen von außen ist ein weiteres Problem. Die Evidenz für die Anwesenheit einer Schwachstelle ist unter anderem die Abwesenheit eines Patches. Evidenzen für einen Patch können nicht immer aus einem Banner abgeleitet werden. So bleibt der Banner eines IIS-Webservers auch nach der Installation eines Patches unverändert. Um mögliche Schwachstellen definitiv von außen zu erkennen, muss das Tiger Team entweder in das System einbrechen, um dort nach Evidenzen eines Patches zu suchen, oder es versucht einen Exploit anzuwenden, ohne Kenntnisse über die tatsächliche Anwesenheit der Schwachstelle auf dem angegriffenen System zu haben. So ist eine Penetration bereits zur Erkennung von Schwachstellen notwendig.

Des Weiteren unterliegt der Penetrationstest zwei zeitlichen Begrenzungen. Zum einen soll ein Tiger Team einen realen Angreifer nachahmen, jedoch soll der Penetrationstest in begrenzter Zeit durchgeführt werden. Ein realer Angreifer hat allerdings mehr Zeit als ein Tiger Team, so dass die Aktionen zur Informationsbeschaffung eher in den Logs eines Angriffsziels auffallen. Auf Grund der begrenzten Zeit werden häufig die Ziele begrenzt (vgl. [Kurtz00a:2]). So kann wie in Kapitel 6 nur die Möglichkeit eines Einbruches von außen das einzige Ziel sein. In diesem Fall werden aber die Risiken von Denial-of-Service Angriffen unbeachtet bleiben.

Zum anderen ist der Penetrationstest nur eine Momentaufnahme. So kann nur versichert werden, dass ein System den zum Zeitpunkt des Penetrationstests bekannten Angriffen widersteht. Durch die permanente Entdeckung neuer Schwachstellen und Bedrohungen ist zum Aufrechterhalten der Zuversicht ein Penetrationstest in regelmäßigen Zeitabschnitten zu wiederholen.

Ein markantes Beispiel hierfür ist die Schwachstelle im RPCSS-Dienst des Betriebssystems Microsoft Windows, die erstmals im Security Bulletin MS03-026 am 16. Juli 2003 veröffentlicht wurde. Sie wurde von dem Wurm W32/Blaster bzw. Lovsan für automatische Angriffe ausgenutzt. Bei eingespieltem Patch können die entsprechenden Exploits in einem Penetrationstest nicht mehr erfolgreich verwendet werden. Jedoch wurden auch Exploits entwickelt, die trotz des eingespielten Patches erfolgreich waren. So wurde die Beschreibung der Schwachstelle am 10. September 2003 im Security Bulletin MS03-039 erweitert und ein neuer Patch bereitgestellt. Wäre ein Penetrationstest etwa im August 2003 durchgeführt worden, wäre ein betrachtetes System trotz der Zusicherung durch den Penetrationstest angreifbar.

Wie im Namen zu erkennen ist, ist auch der Penetrationstest lediglich ein Testverfahren. Jeder Test kann lediglich die Anwesenheit eines Fehlers aufzeigen, nicht aber dessen Abwesenheit. So kann auch der Penetrationstest lediglich nachweisen, dass ein System einem Angriff nicht widersteht. Widersteht ein System einem Angriff, so kann dieser auf Grund mangelnder technischer Kompetenz des Tiger Teams oder der generellen Grenze der Momentaufnahme zu

einem späteren Zeitpunkt möglich sein. Zudem können die Ziele des Penetrationstests zur Begrenzung des technischen und zeitlichen Aufwandes der Art definiert worden sein, dass nicht alle für ein System möglichen Angriffe betrachtet werden können. Da aber nur Gegenbeispiele der Nicht-Angreifbarkeit betrachtet werden, kann die Nicht-Angreifbarkeit nicht bewiesen werden. So ist die verbreitete Ansicht, ein System sei sicher, wenn das Penetration Team nicht einbrechen konnte, ein Mythos. Daher ist der Penetrationstest mit anderen Methoden einer Sicherheitsüberprüfung zu ergänzen.

Auch wenn ein Penetrationstest Mängel zu Tage fördert, muss dies keine Verbesserung der Sicherheit bewirken. Zwar kann das Management wachgerüttelt werden, dennoch muss es selbst entscheiden und Gegenmaßnahmen ergreifen, da das Tiger Team in Form eines Revisors keine Entscheidungsgewalt hat.

Nach der Betrachtung der Möglichkeiten und der Grenzen wird die Arbeit durch ein Fazit abgeschlossen.

9.3. Fazit

Der Penetrationstest ist eine Methode, die sich innerhalb einer Revision dazu eignet, die Auswirkungen eines Angriffes aufzuzeigen sowie zu zeigen, ob Systeme bei etabliertem Sicherheitskonzept bestimmten Angriffen widerstehen. Ist die Widerstandsfähigkeit in einer systematischen Vorgehensweise gezeigt worden, entsteht ein Vertrauen seitens aller Beteiligten in die Wirksamkeit und somit in die Effektivität des Sicherheitskonzeptes.

In dieser Arbeit konnte in Kapitel 6 gezeigt werden, welche Möglichkeiten ein Angreifer auf einem ungesicherten System hat. Allerdings war diese Betrachtung mit einem sehr hohen Aufwand verbunden. Aus diesem Grund wurde auf weitere Angriffe wie spezifische Denial-of-Service Angriffe oder das Erraten von Passwörtern verzichtet. Durch geeignete Sicherheitsmaßnahmen wie das betrachtete Hardening oder eine Firewall auf Applikationsschicht kann dem Risiko effektiv begegnet werden.

Wegen des hohen Aufwands und der mit dem Penetrationstest verbundenen Risiken kann der Penetrationstest nicht als *die* ultimative Methode zur Überprüfung der Sicherheit gesehen werden. Im praktischen Einsatz empfiehlt sich zunächst eine Risikoanalyse, die vorhandene Bedrohungen und Schwachstellen aufzeigt und gegenüberstellt. Auf Grund der Schwierigkeit, Schwachstellen ohne Penetration von außen zu erkennen, sollte die Risikoanalyse dabei vorwiegend von innen erfolgen. Zusätzlich können auch die Methoden der Reconnaissance verwendet werden, um Schwachstellen, die ein potentieller Angreifer zur Beschaffung von Informationen ausnutzen kann, zu vermeiden. So konnten im Kapitel 8 durch die alleinige Reconnaissance auf die Anwesenheit von Schwachstellen geschlossen werden.

Werden in der Risikoanalyse bei bestehendem Sicherheitskonzept Schwachstellen aufgezeigt, so zeigt schon dies, dass die bestehenden Maßnahmen nicht wirksam sein können. Schon eine solche Analyse kann das Vertrauen aller an den Betriebsprozessen Beteiligten stärken. Auf Grund der Ergebnisse können daraufhin weitere Maßnahmen dem verantwortlichen Management vorgeschlagen werden. Will das Management die Auswirkungen der erkannten Bedrohungen trotz der Analyse oder die Notwendigkeit einer vorgeschlagenen Maßnahme aus anderen Gründen nicht erkennen, sollte ein Penetrationstest erfolgen.

Der Aufwand des Penetrationstests verursacht auch Kosten für den Initiator. Sollte trotz der Erkenntnis durch die Risikoanalyse, dass bestehende Maßnahmen wirksam sind, zur Stärkung des Vertrauens ein Penetrationstest durchgeführt werden, so ist die Wirtschaftlichkeit zu beachten. So werden bei einer quantitativen Risikoanalyse die Kosten des Risikos R den Kosten für eine Maßnahme gegenübergestellt, wobei die Kosten der Maßnahme den Wert R nicht übersteigen dürfen. Wird die Effektivität der Maßnahme durch einen Penetrationstest überprüft, so dürfen nach Wilson (vgl. [Wilson03:3]) zum Zweck des Return on Security Investment (ROSI) die kumulierten Kosten der Maßnahme und des Penetrationstests den Wert R nicht übersteigen.

Ein Revisionsprozess arbeitet mit bereits erkannten Schwachstellen. Beinhalten die Information Assets auch eine selbst entwickelte Anwendung wie beispielsweise eine Web-Anwendung, so kann ein aus dem Anwendungsgebiet der Softwareentwicklung bekannter Penetrationstest eingesetzt werden, um bisher unentdeckte potentielle Schwachstellen zu erkennen und somit zur Stärkung der Sicherheit beizutragen. Hierbei kommt es zu einer Vermischung der Anwendungsgebiete Softwareentwicklung und Revision.

Das durchführende Tiger Team sollte nur aus Mitgliedern bestehen, die nicht der Organisation angehören, die den Penetrationstest in Auftrag gibt. Die Nutzung eines solchen externen Dienstes hat gegenüber einem internen Team die Vorteile, dass die Erfahrungen eines professionellen Tiger Teams genutzt und Manipulationen der Ergebnisse vermieden werden können. Zudem sollte bei der Wahl eines Tiger Teams auf die technische und ethische Kompetenz geachtet werden. Auch unqualifizierte Angebote, die etwa nur die Anwendung eines Vulnerability Scanners als Penetrationstest anbieten, sind zu beachten.

Vollständige Sicherheit kann allerdings auch ein Penetrationstest im Rahmen einer Revision nicht gewährleisten. Wegen der Komplexität der Tests kann ein Risiko bei falscher Durchführung trotz der Zusicherung durch den Penetrationstest bestehen. Begründet durch die fast tägliche Entdeckung neuer Schwachstellen und damit verbundener neuer Bedrohungen entstehen neue Risiken, die zum Zeitpunkt des Tests nicht betrachtet werden konnten. Interessanterweise kann zur Entdeckung neuer Schwachstellen ein im Anwendungsgebiet Softwareentwicklung betrachteter Pntrationstest helfen.

So gilt auch nach der Durchführung eines Penetrationstests, dass bekannte Schwachstellen behoben werden müssen, um vermeidbare Unfälle und Vorfälle zu verhindern. Wird dies nicht getan, so können weiter Hacker und Würmer die Systeme schädigen oder beispielsweise Stromausfälle verursachen. Solche Vorfälle sind durch qualitativ hochwertige Sicherheitsprozesse vermeidbar. Die Notwendigkeit der Behebung von Schwachstellen wird auch jenseits der IT deutlich. Werden Schwachstellen nicht behoben, so werden sich wieder Gepäckluken eines Flugzeuges auf Grund eines fehlerhaft konstruierten Schalters während des Fluges öffnen oder Flugzeuge wegen Schwachstellen in der Verkabelung im wahrsten Sinne des Wortes „aus heiterem Himmel“ explodieren. Ich wünsche einen angenehmen Flug.

A n h ä n g e

ANHANG A
QUELLENVERZEICHNIS

Anmerkung

Im Text ist hinter der Quellenangabe die Seitenzahl bzw. der Abschnitt angegeben, um die Referenz genauer angeben zu können. Bei Artikeln, die nur als Webseite verfügbar sind und somit über keine Seitenzahlen verfügen, bezieht sich die jeweilige Seitenzahl auf die Seitenzahl des Ausdrucks. Die Seitenzahl kann sich bei Verwendung eines anderen Browsers oder Druckers jedoch ändern.

[Albitz01] Paul Albitz, Cricket Liu: „DNS und BIND“, 2., korrigierte deutsche Auflage der 3., englischen Auflage, O’Reilly, Köln, 2001

[Allen01] Julia H. Allen: “The CERT® Guide to System and Network Security Practices“, Addison-Wesley, Boston, 2001

[ARIN03] American Registry for Internet Numbers: “About ARIN”, 2003, http://www.arin.net/about_us/index.html
Link geprüft: 26.10.2003

[Barman02] Scott Barman “Writing Information Security Policies”, New Riders, Indianapolis, 2002

[Biggs89] Norman L. Biggs: “Discrete Mathematics”, revised edition, Oxford University Press, New York, 1989

[Bim03] BIM Business Information Management GmbH: „Musterbericht Penetrationstest“, 2003,
Auf Anfrage von via.solution IT professional GmbH zu erhalten.
Siehe http://www.viasolution.de
Link geprüft: 26.10.2003

[BMWA02] Bundesministerium für Wirtschaft und Arbeit (Österreich): „Vorbeugender baulicher Brandschutz Dämmarbeiten / Ausführungsrichtlinien: Teil 1 – Haustechnik“, Juli 2002
http://www.bmwa.gv.at/BMWA/Themen/Tourismus/Baupublikationen/05_haustechnik_leifaden.htm
Link geprüft: 26.10.2003

[Bräuer02] Viola Bräuer: „Simulierter Einbruch“, 2002,
http://www.linux-magazin.de/Artikel/ausgabe/2002/03/pentest/pentest.html
Link geprüft: 26.10.2003

[Brouwner02] J. W. W. Brouwner: “Guide to Cone Pentration test“, 2002,
http://www.conepenetration.com
Link geprüft: 26.10.2003

[Brunnstein99] Klaus Brunnstein: “From AntiVirus to AntiMalware Software and Beyond: Another Approach to the Protection of Customers from Dysfunctional System Behavior“, 22nd NISSC Proceedings, Virginia, 1999,
http://csrc.nist.gov/nissc/1999/proceeding/papers/p12.pdf
Link geprüft: 26.10.2003

[Brunnstein01] Klaus Brunnstein: „Gestaltbarkeit und Beherrschbarkeit von Informatiksystemen (GBI)“, Vorlesung an der Universität Hamburg, Sommersemester 2001

[Brunnstein02] Klaus Brunnstein: „Risikoanalyse und Forensische Informatik“, Vorlesung an der Universität Hamburg, Wintersemester 2002/03

[BS7799-1_99] British Standards Institute: “Information security management – Part 1: Code of practice for information security management“, 1999

[BSI02] Bundesamt für Sicherheit in der Informationstechnik: „BSI-Leitfaden zur Einführung von Intrusion-Detection-Systemen“, 2002, http://www.bsi.de/literat/studien/ids02/lf_index.htm Link geprüft:

[BSI03] Bundesamt für Sicherheit in der Informationstechnik: „Durchführungskonzept für Penetrationstests“, 2003, http://www.bsi.de/literat/studien/pentest/index.htm Link geprüft: 21.01.2004

[CA-2002-27] CERT/CC®: “CERT® Advisory CA-2002-27 Apache/mod_ssl Worm“, 2002, http://www.cert.org/advisories/CA-2002-27.html Link geprüft: 26.10.2003

[CC99] Common Criteria: “Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model“, Version 2.l, 1999, http://www.commoncriteria.org/docs/PDF/CCPART1V21.PDF Link geprüft: 26.10.2003

[CC99a] Common Criteria: “Common Criteria for Information Technology Security Evaluation – Part 2: Security assurance requirements“, Version 2.l, 1999, http://www.commoncriteria.org/docs/PDF/CCPART2V21.PDF Link geprüft: 26.10.2003

[CC99b] Common Criteria: “Common Criteria for Information Technology Security Evaluation – Part 3: Security functional requirements“, Version 2.l, 1999, http://www.commoncriteria.org/docs/PDF/CCPART3V21.PDF Link geprüft: 26.10.2003

[CC2002] Common Criteria: “Supplement: Vulnerability Analysis and Penetration Testing”, CCIMB-2002-07-001, 2002, http://www.commoncriteria.org/review_docs/docs/2002-07-001.pdf Link geprüft: 26.10.2003

[Cert03] CERT® Coordination Center: “CERT®/CC Statistics 1988-2003”, Stand: 24.04.2003 http://www.cert.org/stats Link geprüft: 26.10.2003

[Chapman00] D. Brent Chapman, Simon Cooper, Elizabeth D. Zwicky: “Building Internet Firewalls”, 2. Auflage, O’Reilly, Sebastapol, 2000

[Checkpoint02] Check Point Software Technologies Ltd. : “Check Point Virtual Private Networks Guide“, Part No.: 700528, 2002

[Cole02] Eric Cole: “Hackers Beware”, New Riders, Indianapolis, 2002

[Crichton01] David Chrichton: “Firewall Testing Recommendation”, Diskusion auf [ISC], 7.9. – 11.9.2001, http://marc.10east.com/?l=firewalls-gc&m=99985736125767&w=2 Link geprüft: 26.10.2003

[CZ27/03] Achim Born: „Mangelhafte Verträge behindern Outsourcing”, Artikel in der „Computer Zeitung“, Ausgabe 27/2003

[Davis01] John Davis: “From Blueprint to Fortress A Guide to Securing IIS 5.0”, 2001, http://www.microsoft.com/technet/prodtechnol/iis/deploy/depovg/securiis.asp Link geprüft: 26.10.2003

[Denning90] Peter J. Denning (Hrsg.): “Computers under attack: intruders worms, and viruses”, ACM Press, New York, 1990

[Du00] Wenliang Du, Aditya P. Mathur: “Testing for Software Vulnerability Using Environment Perturbation”, 2000, http://www.cs.umd.edu/~waa/class-pubs/vuln-testing.pdf Link geprüft: 26.10.2003

[Duden1] Duden Verlag (Hrsg.): „Duden, Band 1, Rechtschreibung“, 19. Auflage, Duden Verlag, Mannheim, 1986

[Duden5] Duden Verlag (Hrsg.): „Duden, Band 5, Fremtwörterbuch“, 3. Auflage, Duden Verlag, Mannheim, 1974

[Duden-Oxford90] Duden Verlag, Oxford University Press (Hrsg.): „Duden-Oxford Groß wörterbuch englisch: englisch-deutsch, deutsch-englisch“, Duden Redaktion und Oxford Univ. Press, Mannheim/Oxford, 1990

[vanEden01] Lindsay van Eden: “The Truth about ICMP”, 2001, http://www.giac.org/practical/gsec/Lindsay_Eden_GSEC.pdf Link geprüft: 26.10.2003

[Epstein95] Jeremy Epstein: “Microsoft "Bob" passwords”, Beitrag zu: ” Forum on Risks to the Public in Computers and Related Systems, Volume 17: Issue 12” der ACM, 1995 http://catless.ncl.ac.uk/Risks/17.12.html#subj5 Link geprüft: 26.10.2003

[Eschenbach02] Carola Eschenbach, Rüdiger Valk: „Logik und Semantik (LOS)“, Vorlesung an der Universität Hamburg, Sommersemester 2002

[Farmer93] Dan Farmer: “Improving the Security of Your Site by Breaking into it”, 1993, http://www.fish.com/security/admin-guide-to-cracking.html Link geprüft: 26.10.2003

[Floyd00] Christiane Floyd, Horst Oberquelle: „Softwaretechnik und Softwareergonomie (STE)“, Vorlesung an der Universität Hamburg, Wintersemester 2000/01

[Forristal01] Jeff Forristal, Greg Shipley: “Vulnerabilty Assessment Scanners”, 2001,
http://www.networkcomputing.com/1201/1201f1b1.html
Link geprüft: 26.10.2003

[Fraser97] Barbara Fraser: “Site Security Handbook”, RFC2196, 1997,
ftp://ftp.rfc-editor.org/in-notes/rfc2196.txt
Link geprüft: 26.10.2003

[Fyodor97] Fyodor: “The Art of Port Scanning”, 1997,
http://www.insecure.org/nmap/nmap_doc.html
Link geprüft: 26.10.2003

[Fyodor02] Fyodor: “Remote OS detection via TCP/IP Stack FingerPrinting”, 2002,
http://www.insecure.org/nmap/nmap-fingerprinting-article.html
Link geprüft: 26.10.2003

[Garbars02] Kurt Garbars: “Implementing an Effective IT Security Program”, 2002,
http://www.sans.org/rr/paper.php?id=80
Link geprüft: 26.10.2003

[Garfinkel96] Simson Garfinkel, Gene Spafford: „Practical Unix & Internet Security“, 2. Auflage, O’Reilly, Sebastapol, 1996

[Granger01] Sarah Granger: “Social Engineering Fundamentals, Part I: Hacker Tactics”, 2001,
http://www.securityfocus.com/infocus/1527
Link geprüft: 26.10.2003

[Gröndahl00] Boris Gröndahl: „Hacker", Rotbuch, Hamburg, 2000

[Großklaus99] Axel Großklaus: „Policy, Vorfallsbearbeitung, Schwachstellen-analyse", in Mück, Benecke, Kelm (Hrsg.): „Sicherheit in vernetzten Systemen", Bericht 224, Universität Hamburg, Sommersemester 1999

[GSHB02] Bundesamt für Sicherheit in der Informationstechnik: „IT-Grundschutzhandbuch", Juli 2002, http://www.bsi.bund.de/gshb/deutsch/menue.htm Link geprüft: 26.10.2003

[Heise03] Heise Newsticker: „Internet-Sicherheit: Attacken nahmen um 84 Prozent zu", 7. April 2003, http://www.heise.de/newsticker/data/anw-07.04.03-006/ Link geprüft: 26.10.2003

[Hahn03] Heiner Hahn: „BWL3 – Bilanzierung", Vorlesung an der Universität Hamburg, Sommersemester 2003

[Haller97] Axel Haller: „Wertschöpfungsrechnung", Schäffer - Poeschel Verlag für Wirtschaft, Steuern, Recht, Stuttgart, 1997

[Heinzel03] Marcus Heinzel, Nils Michaelsen, Alexander Scheibe: „Virtual Private Networks", Seminararbeit im Seminar „Sicherheit in vernetzten Systemen", 2003, http://www.informatik.uni-hamburg.de/ RZ/lehre/18.415/seminararbeit/10_VPN.pdf Link geprüft: 26.10.2003

[Higgings01] Scott Higgings: "Physical Penetrations: The Art of Advance Social Engineering", 2001, http://www.giac.org/practical/gsec/Scott_Higgins_GSEC.pdf Link geprüft: 26.10.2003

[HiSolutions03] HiSolutions AG: “Security Auditing und Penetration Testing”, Vortrag im Rahmen der CEFIS auf der CeBIT 2003

[Honeynet03] The Honeynet Project: „Frequently Asked Question“, 2003
http://project.honeynet.org/misc/faq.html#faq3
Link geprüft: 26.10.2003

[Howard98] John D. Howard, Thomas A. Longstaff: “A Common Language for Computer Security Incidents“, Sandia Report, SAND98-8667, 1998
http://www.cert.org/research/taxonomy_988667.pdf
Link geprüft: 26.10.2003

[IIS Insider 10/01] Microsoft Corporation: “IIS Insider – October 2001”,
http://www.microsoft.com/technet/columns/insider/iisi1001.asp
Link geprüft: 26.10.2003

[IIS Insider 05/02] Microsoft Corporation: “IIS Insider – May 2002”,
http://www.microsoft.com/technet/columns/insider/iisi0502.asp
Link geprüft: 26.10.2003

[IIS Insider 07/02] Microsoft Corporation: “IIS Insider – July 2002”,
http://www.microsoft.com/technet/columns/insider/iisi0702.asp
Link geprüft: 26.10.2003

[Inside Security Glossar] Inside Security: “Glossar”, 2003,
http://www.inside-security.de/glossar_p.html
Link geprüft: 26.10.2003

[ISC] isc.org: “Firewalls Mailing list”
http://www.isc.org/services/public/lists/firewalls.html
Link geprüft: 26.10.2003

[Jamieson01] Shaun Jamieson: “The Ethics and Legality of Port Scanning”, 2001, http://www.sans.org/rr/paper.php?id=71 Link geprüft: 26.10.2003

[Kapp00] Justin Kapp: “How To Conduct A Security Audit”, 2000, http://www.pcsupportadvisor.com/nasample/t04123.pdf Link geprüft: 26.10.2003

[Keuper01] Frank Keuper: „Strategisches Management“, Oldenbourg-Verlag, München, 2001

[Keuper02] Frank Keuper, „Folien zur Vorlesung Strategisches Management“, Vorlesung an der Universität Hamburg, Sommersemester 2002

[Klein01] Nick Klein, “How to Conduct Vulnerability Assessment”, 2001, http://www.deloitte.com/dtt/whitepaper/ 0,2312,sid%253D2133%2526cid%253D3331,00.html Link geprüft: 26.10.2003

[Köppel99] Köppel: “How to Support the Negotiation of Service Level Agreements (SLAs) for Your Client/Server Application?”, 1999, http://www.cooperation-management.de/publikationen/ paper/isas99_Abeck-Boening-Koeppel.pdf Link geprüft: 26.10.2003

[Kossakowski92] Klaus-Peter Kossakowski: „Klassifikation und Abwehr von Computer-Würmern in Netzwerken“, Diplomarbeit, Universität Hamburg, 1992, http://www.kossakowski.de/wuermer.htm Link geprüft: 26.10.2003

[Kossakowski01] Klaus-Peter Kossakowski: „Information Technology Incident Response Capabilities”; Dissertation, Books-On-Demand, Hamburg, 2001

[Krabbe98] Elisa Krabbe (Hrsg.): „Leitfaden zum Grundstudium der Betriebswirtschaftslehre“, 6. Auflage, Deutsche-Betriebswirte-Verlag,1998

[Krallmann89] Hermann Krallmann: „EDV-Sicherheitsmanagement“, Erich Schmidt Verlag, Berlin, 1989

[Kurtz00] George Kurtz, Chris Prosise: “Penetration Testing Exposed”, 2000, http://www.infosecuritymag.com/articles/september00/features3.shtml Link geprüft: 26.10.2003

[Kurtz00a] George Kurtz, Chris Prosise: “Penetration Testing: Myth vs. Reality”, 2000, http://www.infosecuritymag.com/articles/september00/features4.shtml Link geprüft: 26.10.2003

[Kurtz01] George Kurtz, Stuart McClure, Jopel Scambray: „Das Anti Hacker Buch“, 2. Auflage, mitp, Landsberg, 2001; Original Titel: “Hacking Exposed”

[Lessing98] G. Lessing: „Penetrationstest in der Praxis“, 1998, http://www.lessing.de/pdf/061.pdf Link geprüft: 26.10.2003

[Lowery02] Jessica Lowery: “Penetration Testing: The Third Party Hacker”, 2002, http://www.sans.org/rr/paper.php?id=264 Link geprüft: 26.10.2003

[Madar03] Paul Madar: “Software Security at Macromedia”, 2003 http://www.macromedia.com/devnet/security/articles/mmsecurity.htm Link geprüft: 26.10.2003

[McGinn-Combs02] Dan McGinn-Combs : “Defining Policies Using Meta Rules”, 2002, http://www.sans.org/rr/paper.php?id=505 Link geprüft: 26.10.2003

[Menne03] Jan Menne: „Methoden der Vorfallserkennung und -analyse“, Diplomarbeit, Universität Hamburg, 2003
http://agn-www.informatik.uni-hamburg.de/papers/doc/diparb_jan_menne.pdf
Link geprüft: 26.10.2003

[Mojert01] Caroline Mojert: „Informationstechnische Gefahren für die Verfügbarkeit von Computernetzwerken“, Diplomarbeit, Universität Hamburg, 2001

[Moyer98] Philip R. Moyer: “What to demand from penetration testers”, 1998,

[MS00-047] Microsoft Corperation: “Microsoft Security Bulletin MS00-047: Patch Available for 'NetBIOS Name Server Protocol Spoofing' Vulnerability”,2000,
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-047.asp
Link geprüft: 26.10.2003

[MS00-078] Microsoft Corperation: “Microsoft Security Bulletin MS00-078: Patch Available for ‘Web Server Folder Traversal’ Vulnerbility”, 2000,
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-078.asp
Link geprüft: 26.10.2003

[MS01-033] Microsoft Corperation: “Microsoft Security Bulletin MS01-033: Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise”,2001,
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp
Link geprüft: 26.10.2003

[MS01-044] Microsoft Corperation: “Microsoft Security Bulletin MS01-044: 15 August 2001 Cumulative Patch for IIS”,2001, http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-044.asp Link geprüft: 26.10.2003

[MS03-007] Microsoft Corperation: “Microsoft Security Bulletin MS03-007: Unchecked Buffer In Windows Component Could Cause Server Compromise (815021) http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-007.asp Link geprüft: 26.10.2003

[MSDN03] Microsoft Corperation: “LocalSystem Account”, 2003, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/localsystem_account.asp Link geprüft: 26.10.2003

[Mück00] Hans-Joachim Mück: “Leitfaden zur Absicherung von Rechnersystemen in Netzen”, DFN-CERT, Hamburg, 2000, ftp://ftp.cert.dfn.de/pub/docs/leitfaden/leitfaden.pdf Link geprüft: 26.10.2003

[Muuss97] Mike Muuss: “The Story of the PING Program”, 1997 http://ftp.arl.mil/~mike/ping.html Link geprüft: 26.10.2003

[Nedon00] Jens Nedon: „Ein IT-Sicherheitskonzept für eine wissenschaftliche Einrichtung“, 2. Auflage, Diplomarbeit, Universität Hamburg, 2000 http://agn-www.informatik.uni-hamburg.de/papers/doc/diparb_jens_nedon_teil1.pdf Link geprüft: 26.10.2003

[Nelißen02] Josef Nelißen: “Buffer Overflows for Dummies”, 2002, http://www.sans.org/rr/paper.php?id=481 Link geprüft: 26.10.2003

[Nevers02] Frank A. Nevers: “Securing Internet Information Server”, 2002, http://webdev.indiana.edu/2002/powerpoint/ECO8.ppt Link geprüft: 26.10.2003

[Northcutt02] Stephen Northcutt, Lenny Zeltser, Scott Winters, Karen Kent Frederik, Ronald W. Ritchey: “INSIDE Network Perimeter Security”, New Riders, Indianapolis, 2002

[Northcutt02a] Stephen Northcutt, Jody Novak: “Network Intrusion Detection”, dritte Auflage, 2002

[Oppliger97] Rolf Oppliger: „IT-Sicherheit : Grundlagen und Umsetzung in der Praxis“, Vieweg Verlag, 1997

[OSSTMM02] Pete Herzog “Open Source Security Testing Methodology Manual”, Version 2.0, 2002, http://www.osstmm.org Link geprüft: 26.10.2003

[Packetstorm00] Anonymous: “IIS5 has a very big bug that let you execute arbitrary comma”, 10.10.2000, http://www2.packetstormsecurity.org/cgi-bin/cbmc/forums.cgi?authkey=anonymous&uname=anonymous&datopic=Windows&mesgcheck=defined&gum=474&editoron= Link geprüft: 26.10.2003

[Page00] Bernd Page: „Diskrete Simulation und Optimierung (DOS)“, Vorlesung an der Universität Hamburg, Wintersemester 2000/01

[Payne01] Patricia Payne: “A Model for Peer Vulnerability Assessment”, 2001, http://www.sans.org/rr/paper.php?id=263 Link geprüft: 26.10.2003

[Pfleeger00] Charles P. Pfleeger: “Security in Computing”, second edition with corrections, Prentence Hall PTR, Upper Saddle River, 2000

[Pfleeger03] Charles P. Pfleeger, Shari Lawrence Pfleeger: “Security in Computing”, third edition, Prentence Hall PTR, Upper Saddle River, 2003

[Q122702] Microsoft Knowledge Base Article: “Using the System Account as a Service in Windows NT 3.5”, http://support.microsoft.com/default.aspx?scid=kb;en-us;122702 Link geprüft: 26.10.2003

[Q236855] Microsoft Knowledge Base Article: “Changes to IWAM Account in IIS 5.0”, http://support.microsoft.com/default.aspx?scid=kb;en-us;236855 Link geprüft: 26.10.2003

[Q254728] Microsoft Knowledge Base Article: “IPSec Does Not Secure Kerberos Traffic Between Domain Controllers”, http://support.microsoft.com/default.aspx?scid=kb;en-us;254728 Link geprüft: 26.10.2003

[Q282784] Microsoft Knowledge Base Article: “Qfecheck.exe Verifies the Installation of Windows 2000 and Windows XP”, http://support.microsoft.com/default.aspx?scid=kb;en-us;282784 Link geprüft: 26.10.2003

[Q296861] Microsoft Knowledge Base Article: “How to Install Multiple Windows Updates or Hotfixes with Only One Reboot”, http://support.microsoft.com/default.aspx?scid=kb;en-us;296861 Link geprüft: 26.10.2003

[Q309689] Microsoft Knowledge Base Article: “HOW TO: Apply Predefined Security Templates in Windows 2000”, http://support.microsoft.com/default.aspx?scid=kb;en-us;309689 Link geprüft: 26.10.2003

[Q316347] Microsoft Knowledge Base Article: “IIS 5: HiSecWeb Potential Risks and the IIS Lockdown Tool”, http://support.microsoft.com/default.aspx?scid=kb;en-us;316347 Link geprüft: 26.10.2003

[Q326444] Microsoft Knowledge Base Article: “HOW TO: Configure the URLScan Tool“, http://support.microsoft.com/default.aspx?scid=kb;en-us;326444 Link geprüft: 26.10.2003

[RFC791] Jonathan B. Postel: “Internet Protocol”, 1981, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc791.txt.pdf Link geprüft: 26.10.2003

[RFC792] Jonathan B. Postel: “Internet Control Message Protocol”, 1981, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc792.txt.pdf Link geprüft: 26.10.2003

[RFC793] Jonathan B. Postel: “Transmission Control Protocol”, 1981, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc793.txt.pdf Link geprüft: 26.10.2003

[RFC821] Jonathan B. Postel: “Simple Mail Transfer Protocol”, 1981, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc821.txt.pdf Link geprüft: 26.10.2003

[RFC826] David C. Plummer: “An Ethernet Address Resolution Protocol”, 1982, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc826.txt.pdf Link geprüft: 26.10.2003

[RFC950] J. Mogul, Jonathan B. Postel: “Internet Standard Subnetting Procedure”, 1981,
ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc950.txt.pdf
Link geprüft: 26.10.2003

[RFC959] Jonathan B. Postel, J. Reynolds: “File Transfer Protocol”, 1985,
ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc959.txt.pdf
Link geprüft: 26.10.2003

[RFC1122] R. Braden: “Requirements for Internet Hosts – Communication Layers”, 1989,
ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc1122.txt.pdf
Link geprüft: 26.10.2003

[RFC1323] V. Jacobson, R. Braden, D. Borman: “TCP Extensions for High Performance”, 1992,
ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc1323.txt.pdf
Link geprüft: 26.10.2003

[RFC1349] P. Almquist: “Type of Service in the Internet Protocol Suite”, 1992,
ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc1349.txt.pdf
Link geprüft: 26.10.2003

[RFC1918] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear: “Address Allocation for Private Internets”, 1996,
ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc1918.txt.pdf
Link geprüft: 26.10.2003

[RFC1945] T. Berners-Lee, R. Fielding, H. Frystyk: “Hypertext Transfer Protocol - HTTP/1.0”, 1996,
ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc1945.txt.pdf
Link geprüft: 26.10.2003

[RFC2228] M. Horowitz, S. Lunt: “FTP Security Extensions”, 1997, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc2228.txt.pdf Link geprüft: 26.10.2003

[RFC2518] Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen: “HTTP Extensions for Distributed Authoring -- WEBDAV”, 1999, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc2518.txt.pdf Link geprüft: 26.10.2003

[RFC2640] B. Curtin: “Internationalization of the File Transfer Protocol”, 1999, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc2640.txt.pdf Link geprüft: 26.10.2003

[RFC2773] R. Housley, P. Yee, W. Nace: “Simple Mail Transfer Protocol”, 2000, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc2773.txt.pdf Link geprüft: 26.10.2003

[RFC2821] J. Klensin: “Simple Mail Transfer Protocol”, 2001, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc2821.txt.pdf Link geprüft: 26.10.2003

[RFC2828] R Shirey: “Internet Security Glossary”, Request for Comment 2828, 2000, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc2828.txt.pdf Link geprüft: 26.10.2003

[RFC3022] P. Srisuresh, K. Egevang: “Traditional IP Network Address Translator (Traditional NAT)”, 2001, ftp://ftp.rfc-editor.org/in-notes/pdfrfc/rfc3022.txt.pdf Link geprüft: 26.10.2003

[Robinson03] Andrew T. Robinson: “Validating Your Security Plan Using Penetration Testing**”,** 2003, www.nmi.net/pages/pentest.html Link geprüft: 26.10.2003

[Röhrig02] Ralf Röhrig: “Nachprüfbar Qualität”, Artikel in LanLine; Ausgabe März 2002

[RUS-CERT] RUS Cert: “Security Scanner”, 2003,
http://helpdesk.rus.uni-stuttgart.de/~rustomfi/Scanner/
Link geprüft: 26.10.2003

[SANS99] SANS Institute: „The 7 Top Management Errors that Lead to Computer Security Vulnerabilities“, As determined by computer security experts and managers meeting at the SANS99 and Federal Computer Security Conferences held in Baltimore May 7-14, 1999
http://www.sans.org/resources/errors.php
Link geprüft: 26.10.2003

[SANS-PR03] SANS policy ressource, 2003
http://www.sans.org/resources/policies/
Link geprüft: 26.10.2003

[Schneier00] Bruce Schneier: “Secrets and Lies”, Wiley Computer Publishing, New York, 2000

[Schultz96] Ph. D. E. Eugene Schultz: „How to Perform Effective Firewall Testing”, 1996,
http://www.spirit.com/CSI/Papers/how2test.htm
Link geprüft: 26.10.2003

[SecWin2000] Microsoft Corporation: „Microsoft Solution for Securing Windows 2000 Server“, 2003,
http://www.microsoft.com/technet/security/prodtech/windows/secwin2k/default.asp
Link geprüft: 26.10.2003

[Snader00] Jon C. Snader: “Effective TCP/IP Programming”, Addison-Wesley, Boston, 2000

[Solomon00] David A. Solomon, Mark Russinovich: “Inside Microsoft Windows 2000”, 3., deutsche Auflage, Microsoft Press, Unterschleißheim, 2000

[SP800-3] John P. Wack: “Establishing a Computer Security incident Response Capability (CSIRC)”, NIST Special Publication 800-3, 1991, http://csrc.nist.gov/publications/nistpubs/800-3/800-3.pdf Link geprüft: 26.10.2003

[SP800-6] W. Timothy Polk: “Automated Tools for Testing Computer System Vulnerability”, NIST Special Publication 800-6, 1992, http://csrc.nist.gov/publications/nistpubs/800-6/800-6.ps Link geprüft: 26.10.2003

[SP800-12] National Institute for Standards and Technology: “An Introduction to Computer Security: The NIST Handbook”, NIST Special Publication 800-12, 1996, http://csrc.nist.gov/publications/nistpubs/800-12/handbook.pdf Link geprüft: 26.10.2003

[SP800-30] Gary Stoneburner, Alice Goguen, Alexis Feringa: “Risk Management Guide for Information Technologie Systems”, NIST Special Publication 800-30, 2001 http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf Link geprüft: 26.10.2003

[SP800-41] John Wack, Ken Cutler, Jamie Pole: “Guidelines on Firewalls and Firewall Policy“, NIST Special Publication 800-41, 2001, http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf Link geprüft: 26.10.2003

[SP800-42] John Wack, Miles Tracy, Murugiah Souppaya“ Guideline on Network Security Testing”, NIST Special Publication 800-42, 2003, http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf Link geprüft: 26.10.2003

[Stewart99] Andrew J. Stuwart. „Distributed Metastasis: A Computer Network Penetration Methodology“, 1999, www.csee.umbc.edu/cadip/docs/NetworkIntrusion/distributed_metastasis.pdf Link geprüft: 26.10.2003

[Strauss91] Christiane Strauss: „Informatik Sicherheitsmanagement – Eine Herausforderung für die Unternehmensführung“, Teubner, Stuttgart, 1991

[Tanenbaum98] Andrew S. Tanenbaum: „Computernetzwerke“, 3., revidierte Auflage, Prentence Hall, Haar bei München, 1998

[TCSEC85] Department of Defense: “Trusted Computer Security Evaluation Criteria - Orange Book”, DoD 5200.28-STD, 1985, http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.pdf Link geprüft: 26.10.2003

[Thompson84] Ken Thompson: “Reflections on Trusting Trust”, 1984, http://www.acm.org/classics/sep95/ Link geprüft: 26.10.2003

[Todd98] Bennett Todd: “Auditing Firewalls: A Practival Guide”, 1998, http://www.itsecurity.com/papers/p5.htm Link geprüft: 26.10.2003

[TTAP00] Trust Technology Assessment Program: “Evaluation Technical Report - Watchguard Technologies: Watchguard Livesecurity System with FireBox II 4.1”, Version 1.0, 2000, http://niap.nist.gov/cc-scheme/TTAP-FER-0015.pdf Link geprüft: 26.10.2003

[Veit99] Thomas Veit: „Sicherheitsüberprüfung einer Internetanbindung“, BSI-Kongress März 1999, http://www.bsi.bund.de/literat/tagung/sikon99/vt_021.htm Link geprüft: 26.10.2003

[Wai01] Chan Tuck Wai: “Conducting a Penetration Test on an Organization”, 2001, http://www.sans.org/rr/paper.php?id=67 Link geprüft: 26.10.2003

[Walker02] William E. Walker IV: „Guide to Secure Configuration and Administration of Microsoft Internet Informationm Services 5.0®“, NSA Report Number: C4-057R-00, 2002, http://www.nsa.gov/snac/win2k/guides/w2k-14.pdf Link geprüft: 26.10.2003

[Wiele02] Dr. Johannes Wiele, Bettina Wesselmann: „Sicherheit heißt Risiken erkennen“, Network World Deutschland, Ausgabe 15/16 02,

[Wilson03] Marcia J. Wilson: „Demonstrating ROI for Penetration Testing (Part One)“, 2003, http://www.securityfocus.com/infocus/1715 Link geprüft: 26.10.2003

[Winkler00] Ira Winkler: “Audits, Assessments & Test (Oh, my)”, 2000, http://www.infosecuritymag.com/articles/july00/features4.shtml Link geprüft: 26.10.03

[Wosnack01] Nathan Wosnack: “Hypervivid Employees; Tiger Team Documentation v1.3”, 2001, http://www.hypervivid.com/TigerTeam1_3_2.pdf Link geprüft: 26.10.2003

[Yazar02] Zeki Yazar: „A Qualitative Risk Analysis and Management Tool – CRAMM“,2002,
http://www.sans.org/rr/paper.php?id=83
Link geprüft: 26.10.2003

[Yilmaz02] Ahmet Yilmaz: „Angriffe auf die Netzwerksicherheit“, Vortrag im Seminar „Sicherheit in vernetzten Systemen“an der Universität Hamburg, Wintersemester 2002/03

[Zavdi01] Moran Zavdi: ".htr bug still exist after applying MS patches", 2001,
http://archives.neohapsis.com/archives/bugtraq/2001-01/0502.html
Link geprüft:

[ZKDSG02] Bundesgesetzblatt: „Gesetz über den Schutz von zugangskontrollierten Diensten und von Zugangskontrolldiensten
(Zugangskontrolldiensteschutzgesetz – ZKDSG)“, Bonn, 2003,
http://217.160.60.235/BGBL/bgbl1f/BGBl102019s1090.pdf
Link geprüft: 26.10.2003

ANHANG B
ABBILDUNGSVERZEICHNIS

ANHANG C
TABELLENVERZEICHNIS

ANHANG D
LINKS ZUR IIS-SICHERHEIT

Dieser Anhang listet weitere Links auf, die Inhalte zu Sicherung der Microsoft Internet Information Services bieten. Sie wurden in Kapitel 6 nicht verwendet, sollen aber einen interessierten Leser bei der Vertiefung des Themas unterstützen.

Michael Howard: „Secure Internet Information Services 5 Checklist“
http://www.microsoft.com/technet/security/tools/chklist/iis5chk.asp

Übersicht der Themen zu den Internet Information Services 5 auf der Microsoft Homepage
http://www.microsoft.com/technet/prodtechnol/iis/default.asp

Freie Tools für Sicherheitsaspekte der Firma Microsoft
http://www.microsoft.com/technet/security/tools/tools.asp

Freie Tools des Windows 2000 Ressource Kits
http://www.microsoft.com/windows2000/techinfo/reskit/tools/default.asp

Artikel zum Hardening von Windows 2000
http://www.systemexperts.com/tutors/HardenW2K101.pdf

Diverse Artikel zur Sicherung von Windows 2000
http://www.labmice.net/Security/securewin2000.htm

Informationen zur sicheren Konfiguration von Microsoft Windows
http://www.cert.org/tech_tips/win_configuration_guidelines.html

Pat Schneider, Mahran Yanya : „Securing IIS Servers“
http://www.asu.edu/it/ac/uncel/projects/SecIIS.pdf

Artikel zur Sicherung eines öffentlichen Webservers
http://www.cert.org/security-improvement/modules/m11.html

Nist: „System Administrators Guidance for Windows 2000 Server“
http://csrc.nist.gov/itsec/guidance_W2Kpro.html

Leitfäden der NSA zur Sicherung von Windows 2000
http://www.nsa.gov/snac/win2k/index.html

Homepage zu unterschiedlichen, die IIS betreffenden Themen
http://www.iisfaq.com

ANHANG E
SCHWACHSTELLENANALYSE DES IIS 5.0

Auf der folgenden Tabelle werden alle Schwachstellen aufgeführt, die seit dem Erscheinen von Windows 2000 in dem zugehörigen IIS 5.0 entdeckt worden sind. Grundlage für die Sortierung sind die Microsoft Bulletins, denen ein passender Patch zu entnehmen ist. Die Bugtraq-Vulnerability Database bildet die Basis für die Benennung der Schwachstellen.

Zu Informationen zu den Commen Vulnearbilities and Exposures (CVE) siehe cve.mitre.org .

Zu Informationen zu CERT®/CC Advisories und Vulnerability Notes siehe www.cert.org.

Für Mircosoft Security Bulletins siehe
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/current.asp.

Zur Bugtraq Vulnerabity Database siehe www.securityfocus.com/bid

CVE	CERT/CC	BID	Patch	Name / Description
-	-	-	-	Backup CGIs download (NessusID:11411)
-	CA-2000-02	-	-	Malicious HTML Tags in Client Requests Cross Site Scripting Vulnerability
-	-	-	-	Cross Site Tracing
-	-	-	-	IIS Sample-Application discloses physical path of web root
-	-	-	-	Banner discloses web server type and version
CAN-1999-0450	-	0194	-	IIS ISAPI Extension Enumerate Root Web Server
-	-	2736	-	IIS WebDAV Lock Method Memory Leak DoS Vulnerability
-	-	5907	-	IIS Malformed HTTP HOST Header Field Denial Of Service Vulnerability
CAN-2001-0902	-	6795	-	IIS False Logging Weakness
-	-	7492	-	IIS User Existence Disclosure Vulnerability
CAN-2000-0649	-	1499	Q218180	IIS Internal IP Address Disclosure Vulnerability
-	-	1756	Q272079	IIS 5.0 Indexed Directory Disclosure Vulnerability
CVE-2000-0246	-	1081	MS00-019	IIS UNC Mapped Virtual Host Vulnerability
CVE-2000-0258	-	1101	MS00-023	IIS 4.0/5.0 Escaped Characters Vulnerability
CVE-2000-0408	-	1190	MS00-030	IIS 4.0/5.0 Malformed File Extension DoS Vulnerability
CVE-2000-0630	VU#28565	1488	MS00-031	IIS 4.0/5.0 Source Fragment Disclosure Vulnerability
CVE-2000-0304	-	1191	MS00-044	IIS 4.0/5.0 Malformed .htr Request Vulnerability
CVE-2000-0631	-	1476	MS00-044	IIS .htr Missing Variable Denial of Service Vulnerability
CVE-2001-0004	VU#264272	2313	MS00-044	IIS file Fragment Disclosure Vulnerability
CVE-2000-0770	-	1565	MS00-057	IIS 4.0/5.0 File Permission Canonicalization Vulnerability
CVE-2000-0778	-	1578	MS00-058	IIS 5.0 “Translate: f” Source Disclosure Vulnerability
CVE-2000-0746	-	1595	MS00-060	IIS Cross Site Scripting .shtml Vulnerability
CVE-2000-0884	VU#111677	1806	MS00-078	IIS and PWS Extended Unicode Directory Traversal Vulnerability
-	-	1832	MS00-080	IIS 4.0/5.0 Session ID Cookie Disclosure Vulnerability

CVE	CERT/CC	BID	Patch	Name / Description
CVE-2000-0886	-	1912	MS00-086	IIS Executable File Parsing Vulnerability
CAN-2000-1105	VU#829845	1933	MS00-098	Microsoft Indexing Services for Windows 2000 File Verification Vulnerability
CAN-2001-0146	VU#796584	2440	MS01-014	IIS 5.0 Multiple Invalid URL Request DoS Vulnerability
CVE-2001-0241	VU#516648	2674	MS01-023	IIS 5.0 .printer ISAPI Extension Buffer Overflow Vulnerability (vgl. [Nevers02])
CVE-2001-0151	-	2453	MS01-026	IIS WebDAV Denial of Service Vulnerability
CVE-2001-0333	VU#789543	2708	MS01-026	IIS/PWS Escaped Characters Decoding Command Execution Vulnerability
CVE-2001-0335	VU#137544	2719	MS01-026	Microsoft FTP searches all trusted domains for user accounts
CVE-2001-0500	CA-2001-13	2880	MS01-033	Indexing Service ISAPI Extension Buffer Overflow Vulnerability
CVE-2001-0504	VU#435963	2988	MS01-037	Windows 2000 SMTP Improper Authentication Vulnerability
CVE-2000-0457	VU#35085	1193	MS01-044	IIS 4.0/5.0 Malformed Filename Request Vulnerability
CVE-2001-0506	VU#630531	3190	MS01-044	IIS SSI Buffer Overrun Privilege Elevation Vulnerability
CVE-2001-0507	-	3193	MS01-044	IIS 5.0 In-Process Table Privilege Elevation Vulnerability
CVE-2001-0508	VU#959211	3194	MS01-044	IIS WebDAV Invalid Request Denial of Service Vulnerability
CVE-2001-0544	-	3195	MS01-044	IIS MIME Header Field Delimiter Buffer Overflow Vulnerability
CVE-2002-0071	-	4474	MS02-018	IIS HTR ISAPI Extension buffer Overflow Vulnerability
CVE-2002-0150	VU#454091	4476	MS02-018	IIS HTTP Header Field Delimiter Buffer Overflow Vulnerability
CVE-2002-0149	VU#721963	4478	MS02-018	IIS ASP Server-Side Include Buffer Overflow Vulnerability
CVE-2002-0072	-	4479	MS02-018	IIS ISAPI Filter Access Violation Denial of Service Vulnerability
CAN-2002-0073	VU#412203	4482	MS02-018	IIS FTP Connection Status Request Denial of Service Vulnerability
CVE-2002-0074	VU#883091	4483	MS02-018	Cross Site Scripting Vulnerability in Help-file Search Facility of IIS
CVE-2002-0079	VU#610291	4485	MS02-018	IIS chunked Encodng Transfer Heap Overflow Vulnerability
CVE-2002-0148	VU#886699	4486	MS02-018	IIS HTTP Error Page Cross Site Scripting Vulnerability
CVE-2002-0075	-	4487	MS02-018	IIS HTTP Redirect Cross Site Scripting Vulnerability
CVE-2002-0147	VU#66779	4490	MS02-018	IIS Chunked Encoding Head Overflow Variant Vulnerability

CVE	CERT/CC	BID	Patch	Name / Description
CAN-2003-0225	-	7733	MS02-018	IIS ASP Header Denial of Service Vulnerability
CVE-2002-0364	-	4855	MS02-028	IIS HTP Chunked Encoding Transfer Heap Overflow Vulnerability
-	-	5213	MS02-062	IIS SMTP Service Encapsulated SMTP Address Vulnerability
CAN-2002-0896	-	6069	MS02-062	IIS Out Of Process Privilege Escalation Vulnerability
CAN-2002-1182	-	6070	MS02-062	IIS WebDAV Denial of Service Vulnerability
CAN-2002-1180	-	6071	MS02-062	IIS Script Source Access File Upload Vulnerability
CAN-2002-1181	-	6072	MS02-062	IIS Administrative Pages Cross Site Scripting Vulnerability
CAN-2002-1142	VU#542081	6214	MS02-065	Microsoft Data Access Components RDS Buffer Overflow Vulnerability
CAN-2003-0109	CA-2003-09	7116	MS03-007	Microsoft Windows ntdll.dll Buffer Overflow Vulnerability
CAN-2003-0223	-	7734	MS03-018	IIS SSINC.DLL Server Side Includes Buffer Overflow Vulnerability
CAN-2003-0224	-	7735	MS03-018	IIS WebDAV PROPFIND and SEARCH Method Denial of Service Vulnerbility
CAN-2003-0226	-	8035	MS03-022	IIS ISAPI Extension for Windows Media Services Buffer Overflow Vulnerability
CAN-2000-0071	-	1065	SP3	Microsoft IIS UNC Path Disclosure Vulnerability
-	-	5900	SP3	IIS IDC Extension Cross Site Scripting Vulnerability
-	-	8092	SP4	IIS _VTI_BOT Malicious WebBot Elevated Permissions Vulnerability

ANHANG F
SOFTWARE

Dieser Anhang enthält eine Übersicht der verwendeten Software. Des Weiteren werden andere Programme angegeben, die zwar nicht verwendet werden, für das Thema Penetrationstest jedoch geeignet sind.

F1 Verwendete Software

Amap 2.41
www.thc.org/releases.php
Tool für automatisches Banner Grabbing

Cerberus Internet Scanner (CIS) 5.0.02
http://www.cerberus-infosec.co.uk/CIS-5.0.02.zip
Vulnerability Scanner für Windows

Checkpoint NG Feature Pack 3
www.checkpoint.com
Leistungsfähige Firewall und VPN Software

enum
http://www.cotse.com/tools/netbios.htm
Tool zur Enumeration des SMB-Dienstes

Gentoo Linux 1.4rc2
www.gentoo.org
Linux Distribution, die von den Quellen kompiliert wird.

ipsecpol
Enthalten im → Microsoft Windows 2000 Resource Kit

LanGuard Network Security Scanner 3.3
www.languard.com
Vulnerability Scanner der Firma LanGuard

Microsoft Platform SDK
www.microsoft.com/msdownload/platformsdk/sdkupdate
Software Developement Kit; bibliotuheken und Beispiele für die Entwicklung von Windows-Programmen

Microsoft Windows NT 4.0 Server
www.microsoft.com/germany
Weit verbreitetes Betriebssystem

Microsoft Windows 2000 Professional deutsch
www.microsoft.com/germany
Weit verbreitetes Betriebssystem

Microsoft Windows 2000 Resource Kit
http://www.microsoft.com/windows2000/techinfo/reskit/default.asp
Zusatzprogramme für das Betriebssystem

nessus 2.0.7
www.nessus.org
Open-Source Vulnerability Scanner

Netcat von atstake
www.atstake.com
Vielseitiges Tool für Netzwerke

Nikto
www.cirt.net/code/nikto.shtml
Auf Webserver spezialisierter Vulnerability Scanner

Nmap 3.00
www.insecure.org/nmap
Umfangreichster Portscanner; bietet auch Möglichkeiten des OS-Fingerprinting

nt tools von Jesper Lauritson
http://www.ibt.ku.dk/jesper/NTtools/
Verschiedene Tools für Windows NT

OllyDebug
http://home.t-online.de/home/Ollydbg/
Debugger für Windows

Sara 4.21
www-arc.com
Vulnerabilityscanner

Snmptool
http://www.jklein.de/techniker_arbeit/tech_data/snmptool.exe
Tool für SNMP-Abfragen

snmputil
Enthalten im → Microsoft Platform SDK

Stealth-http
www.devhood.com/tools/tool_details.aspx?tool_id=353
Auf Webserver spezialisierter Vulnerability Scanner

SuperScan 3.00
www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/freetools.htm
Portscanner

Suse Linux 7.3
www.suse.de
Linux Distribution, die auf dem Rechner mirror.irt.local verwendet wurde.

Tftp-Server
http://tftpd32.jounin.net/
TFTPD32 von Philippe Jounin

Traceroute 1.4a13
ftp://ftp.ee.lbl.gov/traceroute-1.4a12.tar.gz[98]
Traceroute-Variante mit Schalter –S, um Inkrementieren der Portnummer zu verhindern.

whoami
Enthalten im → Microsoft Windows 2000 Ressource Kit

F2 Weitere Programme

Tools für Reconnaissance

WS_Ping ProPack (http://www.ipswitch.com/Products/WS_Ping/)
Cheops (http://www.marko.net/cheops/)

Vulnerability Scanner

Eeye Retina Scanner (http://www.eeye.com/html/Products/Retina/)
IIS Internet Security Scanner (www.iis.net)
Qualys QualysGuard (www.qualys.com)

Packetgeneratoren

Nemesis
Sendip
Hping2

Password Tools (ohne Bezugsquelle)

Advanced NT Security Explorer
Brutus
Cain und Abel
Claymore
Crack

[98] Der Tarball auf dem Server trägt unverständlicherweise die Versionsnummer 1.4a12. Das Programm meldet sich jedoch mit 1.4a13

L0phtcrack
Pwddump
Lsasecrets

Weitere Links

www.monkey.org
www.Thc.org/releases.php

ANHANG G
QUELLCODES

G1 addrrange

```
#!/bin/sh
whois –h rr.arin.net `host $1 | cut -f 4 -d " " | cut -f 1-4 -d "." -s` |\
grep inetnum | cut -f 2 -d ":"
```

G2 echofake.c

```
/* echo-fake
 * May 23,2003: Nils Michaelsen
 * based on Jon C. Snader: "Effective TCP/IP Programming",
 * Addison-Wesley, 2000, p.14 ([Snader00])
 *
 * fakes echo-service on another port
 *
 * vulnerable to buffer overflow ?
 */

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>

int main (int argc, char **argv)
{
  struct sockaddr_in local;
  int s;
  int s1;
  int rc;
  int c;
  int i;
  int port = 21;
  char buf[80];
  char ch[1];

  printf ("echo-fake: listen on port %d\n",port);

  local.sin_family = AF_INET;
  local.sin_port = htons ( port );
  local.sin_addr.s_addr = htonl( INADDR_ANY );
  s=socket(AF_INET, SOCK_STREAM,0 );
  if ( s < 0)
```

```
  {
     perror( "socket call failed" );
     exit( 1 ) ;
  }
  rc = bind( s, ( struct sockaddr* )&local, sizeof( local ));
  if ( rc < 0)
  {
     perror( "bind call failure" );
     exit( 1 );
  }
  rc = listen (s, 5);
  if ( rc  )
  {
     perror( "listen call failed" );
     exit( 1 );
  }
  s1 = accept( s, NULL,NULL );
  if ( s1 < 0 )
  {
     perror( "accept call failed" );
     exit( 1 );
  }
  rc = recv( s1, buf, 80, 0);
  if ( rc <= 0 )
  {
     perror( "recv call failed" );
     exit ( 1 );
  }
  printf ("received: %s\n", buf );
  /*
   *  search for char13
   */
   i = 0;
   while (buf[i]!=13) {
        printf ("i = %d, buf[i]: %c %d\n",i,buf[i],buf[i]);
        ch[0] = buf[i];
        rc = send( s1,ch,1,0);
        if (rc<=0) {
            perror("send call failed");
            exit (0);
        }
        i++;
       if (i==80) { exit (0); };
   }
  exit(0);
}
```

G3 Alarm Mayer-Homepage

In alphabetischer Reihenfolge

agb.html

```
<HEAD>
```

```
<title>Alarm Mayer: AGB</title>
</HEAD>
<BODY bgcolor="FFFF88">
<p align="center"><span style="font-family:sans-serif">
<b>Alarm Mayer - AGB</b><BR><br>
<b>1. Regelungen</b><br>
Es gelten die gesetzlichen Regelungen nach §§474 - 479 BGB<br><br>
<b>2. Gewährleistung</b>
Es gilt die gesetztliche Gewährleistungspflicht von 24 Monaten. Garantien
werden nicht gegeben.
<b>3. Schadensersatz</b>
Schadensersatz ist ausgeschlossen.
<b>4. Inhalt</b>
Diese AGB ist Inhalt jedes Vertrages zwischen einem Kunden und Alarm
Mayer.
</span></p>
</BODY>
</HTML>
```

alarm.html

```
<HTML>
<HEAD>

</HEAD>
<body bgcolor="#FFFF88"  LINK="#004080" VLINK="#808080" ALINK="#ffff00">
<img src="images/alarm.gif">
</BODY>
</HTML>
```

index.html

```
<HTML>
<HEAD>
<title>Alarm Mayer</title>
</HEAD>
<frameset rows="135,*" bordercolor="#FFFF88" frameborder="0"
framespacing="0" border="0">
 <frameset cols="125,*" bordercolor="#FFFF88" frameborder="0"
framespacing="0" border="0">
    <frame src="alarm.html" name="alarm" scrolling="no">
    <frameset rows="75,*" bordercolor="#FFFF88" frameborder="0"
framespacing="0" border="0">
       <frame src="title.html" name="title" scrolling="no">
       <frame src="navigation.html" name="navigation" scrolling="no">
     </frameset>
 </frameset>
 <frame src="kontakt.html" name="inhalt">
 <noframes>
   Ihr Browser unterstützt keine Frames! Er kann diese Seite leider nicht
anzeigen!
 </noframes>
</frameset>
</HTML>
```

kontakt.html

```
<HTML>
<HEAD>
```

```
<title>Alarm Mayer</title>
</HEAD>
<BODY bgcolor="FFFF88">
<p align="center"><span style="font-family:sans-serif">
<b>Alarm Mayer</b><BR><br>
<i>Sicherheitssysteme f&uuml;r das Haus</i><br><br>
Inhaber Hugo Mayer<br>
Hamburger Strasse 12<br>
12345 Musterstadt<br><br>
Telefon: +49-00-123 45-678<br>
www.alarm-mayer.de<br>
<a href="mailto:info@alarm-mayer.de">info@alarm-mayer.de</a><br><br>

</span></p>
</BODY>
</HTML>
```

navigation.html

```
<HTML>
<HEAD>

</HEAD>
<BODY bgcolor="FFFF88">

<a href="script/produkte.php" target="inhalt">
<img src="images/button2.gif" width="120" height="30" border="0"
alt="Produkte">
</a>
   

<img src="images/button1.gif" width="120" height="30" border="0"
alt="Bestellung">
   

<a href="agb.html" target="inhalt">
<img src="images/button5.gif" width="120" height="30" border="0"
alt="AGB">
   
</a>
<a href="kontakt.html" target="inhalt">
<img src="images/button3.gif" width="120" height="30" border="0"
alt="Kontakt">
</a>
   

<img src="images/button4.gif" width="120" height="30" border="0"
alt="Login">
   

</BODY>
</HTML>
```

title.html

```
<HEAD>
</HEAD>
<BODY bgcolor="#FFFF88"  LINK="#004080" VLINK="#808080" ALINK="#ffff00">
```

```
<img src="images/title.gif">
</BODY>
</HTML>
```

Script\produkte.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Alarm Mayer: Produkte<title>
</head>
<body bgcolor="#FFFF88">
 <p><span style="font-family:sans-serif">
            <TABLE border="0">
      <colgroup>
        <col width="200">
        <col width="400">
        <col width="100">
      </colgroup>
      <TR>
      <td><b>Bezeichnung</b></td><td><b>Beschreibung</b></td><td><b>Preis
(&euro;)</b></td>
      </TR>
      </table><HR><table border="0">
      <colgroup>
        <col width="200">
        <col width="400">
        <col width="100">
      </colgroup>
      <TR><td></td><td></td><td></td></TR>
<?
            $error = 0;
        // Datenbankverbindung öffnen, auf Fehler prüfen
       if (! ($dbLink = mysql_pconnect ("localhost", "root", "geheim"))) {
         print ("Datenbankverbindung fehlgeschlagen!\n");
         print ("MySQL meldet: " . mysql_error() . "\n");
         $error++;
       }
       // Datenbank auswählen, auf Fehler prüfen
       if (! (mysql_select_db ("mayeralarm", $dbLink))) {
          print ("Fehler bei der Benutzung der Datenbank!\n");
          print ("MySQL meldet: " . mysql_error() . "\n");
          $error++;
       }
       $Query = "SELECT * FROM Produkte;";

       //Query durchführen
       if (! ($dbResult = mysql_query ($Query))) {
          print ("Fehler beim Ausführen der folgenden Abfrage !\n");
          print ("<pre>$Query[$i]</pre><BR>\n");
          print ("MySQL meldet: " . mysql_error() . "\n");
          $error++;
       }

     while ($row = mysql_fetch_object ($dbResult)) {
              print ("<TR><TD>$row->bez</TD><TD>$row->beschr</TD><TD>$row-
>preis <br></TD></TR>");
       }
```

```
?>
   </table>
   </span></p>
</body>
</html>
```

G4 exwebdav.pl

```
#!/usr/bin/perl -w

$target="192.168.0.3";
for ($i=0;$i < 256; $i++) {
  $retcode = sprintf("%x", $i);
  $exec = "./exwebdav $target 80 2512 " . $retcode . "04";
  printf("%s",$exec);
  system($exec);
  sleep(13);
  $exec="telnet 192.168.0.40 2512";
  system($exec);
  $exec="./restart.sh";
  system($exec);
}
```

G5 simples.c

```
/*************************************************************************\
* simples.c - Simple TCP/UDP server using Winsock 1.1
*       This is a part of the Microsoft Source Code Samples.
*       Copyright 1996 - 2000 Microsoft Corporation.
*       All rights reserved.
*       This source code is only intended as a supplement to
*       Microsoft Development Tools and/or WinHelp documentation.
*       See these sources for detailed information regarding the
*       Microsoft samples programs.
\*************************************************************************/
/*
 *  Modifiziert zwecks Diplomarbeit
 *
 *  Penetrationstest: Möglichkeiten und Grenzen
 *
 *  Nils Michaelsen, 5. August 2003
 */



#ifdef _IA64_
#pragma warning(disable:4127)
#endif
```

```
#define WIN32_LEAN_AND_MEAN
#include <winsock2.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#define DEFAULT_PORT 5001
#define DEFAULT_PROTO SOCK_STREAM // TCP

void Usage(char *progname) {
        fprintf(stderr,"Usage\n%s -p [protocol] -e [endpoint] -i
[interface]\n",
                progname);
        fprintf(stderr,"Where:\n\tprotocol is one of TCP or UDP\n");
        fprintf(stderr,"\tendpoint is the port to listen on\n");
        fprintf(stderr,"\tinterface is the ipaddr (in dotted decimal
notation)");
        fprintf(stderr," to bind to\n");
        fprintf(stderr,"Defaults are TCP,5001 and INADDR_ANY\n");
        WSACleanup();
        exit(1);
}
int main(int argc, char **argv) {

        /* MODIFIED : buffer size changed from 128 to 32 */
        char Buffer[32];
        /* added by nm */
        char Command[32];
        int cmdfin;

        char *interface= NULL;
        unsigned short port=DEFAULT_PORT;
        int retval;
        int fromlen;
        int i;
        int socket_type = DEFAULT_PROTO;
        struct sockaddr_in local, from;
        WSADATA wsaData;
        SOCKET listen_socket;
        SOCKET msgsock = INVALID_SOCKET;

        /* Parse arguments */
        if (argc >1) {
                for(i=1;i <argc;i++) {
                        if ( (argv[i][0] == '-') || (argv[i][0] == '/') ) {
                                switch(tolower(argv[i][1])) {
                                        case 'p':
                                                if (!stricmp(argv[i+1],
"TCP") )
                                                        socket_type =
SOCK_STREAM;
                                                else if (!stricmp(argv[i+1],
"UDP") )
                                                        socket_type =
SOCK_DGRAM;
                                                else
                                                        Usage(argv[0]);
                                                i++;
                                                break;
```

```
                                        case 'i':
                                                interface = argv[++i];
                                                break;
                        case 'e':
                            // NOTE on 64 bit, possible loss of data in cast
                                                port =
(USHORT)atoi(argv[++i]);
                                                break;
                                        default:
                                                Usage(argv[0]);
                                                break;
                                }
                        }
                        else
                                Usage(argv[0]);
                }
        }

        if ((retval = WSAStartup(0x202,&wsaData)) != 0) {
                fprintf(stderr,"WSAStartup failed with error %d\n",retval);
                WSACleanup();
                return -1;
        }

        if (port == 0){
                Usage(argv[0]);
        }

        local.sin_family = AF_INET;
        local.sin_addr.s_addr =
(!interface)?INADDR_ANY:inet_addr(interface);

        /*
         * Port MUST be in Network Byte Order
         */
        local.sin_port = htons(port);

        listen_socket = socket(AF_INET, socket_type,0); // TCP socket

        if (listen_socket == INVALID_SOCKET){
                fprintf(stderr,"socket() failed with error
%d\n",WSAGetLastError());
                WSACleanup();
                return -1;
        }
        //
        // bind() associates a local address and port combination with the
        // socket just created. This is most useful when the application is
a
        // server that has a well-known port that clients know about in
advance.
        //

        if (bind(listen_socket,(struct sockaddr*)&local,sizeof(local) )
                == SOCKET_ERROR) {
                fprintf(stderr,"bind() failed with error
%d\n",WSAGetLastError());
                WSACleanup();
```

```
                return -1;
        }

        //
        // So far, everything we did was applicable to TCP as well as UDP.
        // However, there are certain steps that do not work when the
server is
        // using UDP.
        //

        // We cannot listen() on a UDP socket.

        if (socket_type != SOCK_DGRAM) {
                if (listen(listen_socket,5) == SOCKET_ERROR) {
                        fprintf(stderr,"listen() failed with error
%d\n",WSAGetLastError());
                        WSACleanup();
                        return -1;
                }
        }
        printf("%s: 'Listening' on port %d, protocol %s\n",argv[0],port,
                (socket_type == SOCK_STREAM)?"TCP":"UDP");
        while(1) {
                fromlen =sizeof(from);
                //
                // accept() doesn't make sense on UDP, since we do not
listen()
                //
                if (socket_type != SOCK_DGRAM) {
                        if (msgsock == INVALID_SOCKET) {
                                msgsock = accept(listen_socket,(struct
sockaddr*)&from, &fromlen);
                                if (msgsock == INVALID_SOCKET) {
                                        fprintf(stderr,"accept() error
%d\n",WSAGetLastError());
                                        WSACleanup();
                                        return -1;
                                }
                                printf("accepted connection from %s, port
%d\n",
                                                inet_ntoa(from.sin_addr),
                                                htons(from.sin_port)) ;
                                }

                }
                else
                        msgsock = listen_socket;

                //
                // In the case of SOCK_STREAM, the server can do recv() and
                // send() on the accepted socket and then close it.

                // However, for SOCK_DGRAM (UDP), the server will do
                // recvfrom() and sendto()  in a loop.

                if (socket_type != SOCK_DGRAM)
                        retval = recv(msgsock,Buffer,sizeof (Buffer),0 );
                else {
                        retval = recvfrom(msgsock,Buffer,sizeof (Buffer),0,
                                (struct sockaddr *)&from,&fromlen);
```

```
                        printf("Received datagram from
%s\n",inet_ntoa(from.sin_addr));
                }

                if (retval == SOCKET_ERROR) {
                        fprintf(stderr,"recv() failed: error
%d\n",WSAGetLastError());
                        closesocket(msgsock);
                        msgsock = INVALID_SOCKET;
                        continue;
                }
                if (retval == 0) {
                        printf("Client closed connection\n");
                        closesocket(msgsock);
                        msgsock = INVALID_SOCKET;
                        continue;
                }


                /* MODIFIED: printf content of buffer */
                cmdfin=0;
                for (i=0;i<32;i++) {
                        if (Buffer[i]==-52) Buffer[i]=45;
                /*      printf("Buffer[%3d]= %3d ",i, Buffer[i]);
                        (((i%4)==3) ? printf("\n") : printf(" | "));*/
                        if (cmdfin==0) {
                                /* Command only compatible to bsd-style
telnet comes with linux */
                                if (Buffer[i]==13) {
                                        Command[i]=0;
                                        cmdfin=1;
                                } else {
                                        Command[i]=Buffer[i];
                                }
                        }

                }


                printf("Received %d bytes, data [%s] from
client\n",retval,Buffer);
                printf("Command: >%s<\n", Command);

                if (strncmp(Command,"quit",4)==0) exit (0);

                if (strncmp(Command,"restart",7)==0) {
                        i = system("net stop w3svc");
                        i = system("net start w3svc");
                }


/*              printf("Echoing same data back to client\n");
                if (socket_type != SOCK_DGRAM)
                        retval = send(msgsock,Buffer,sizeof(Buffer),0);
                else
                        retval = sendto(msgsock,Buffer,sizeof (Buffer),0,
                                (struct sockaddr *)&from,fromlen);
                if (retval == SOCKET_ERROR) {
```

```
                        fprintf(stderr,"send() failed: error
%d\n",WSAGetLastError());
                }*/
        }
}
```

G6 revinf.sh

```
#!/bin/bash
# reverse-anfrage für informatik.uni-hamburg.de
# Aug, 2003 Nils Michalsen
echo "Ausgabe von revinf `date`" >> logs/output.log
for ((i=$1;i<$2;i++)); do
  for ((j=0;j<256;j++)); do
    echo "nslookup -sil 134.100.$i.$j" >>logs/output.log
    nslookup -sil 134.100.$i.$j
  done
done
```

G7 hinfo.pl

```
#!/usr/bin/perl -w
#
#   *** hinfo.pl ***
#
#  Sep 2003, Nils Michaelsen
#
#  published under the terms of GPL
#
#  tries to get hinfo Resource Record of targets specified in hinfo.in


print ‘date‘;

open(IFH,"<hinfo.in");

while (<IFH>) {
$target = $_;
#$target = "rzdspc77.informatik.uni-hamburg.de";
@outtarget = split /\n/, $target;
print $outtarget[0], "  ";
$out = ‘nslookup -sil -query=hinfo $target‘;
@lines = split /\n/, $out;
#$i=0;
#foreach $l (@lines) {
#  print $i++," ", $l,"\n";
#}
push (@result, $target);
if ($lines[3] =~ /No answer/) {
  print "no answer \n";
```

```
  push (@result, "No answer");
} else {
  print $lines[3], "\n";
  push (@result, $lines[3]);
}

#foreach $l (@result) {
#  print  $l, "\n";
#}
}
close (IFH);

print (‘date‘);
```

G8 txt.pl

```
#!/usr/bin/perl -w
#
#   *** txt.pl ***
#
#  Sep 2003, Nils Michaelsen
#
#  published under the terms of GPL
#
#  tries to get txt Resource Record of targets specified in hinfo.in

$DEBUG = 0;
$cmd = "echo ‘date‘ >fbitxt.log";
$ret = ‘$cmd‘;

open(IFH,"<dns.in");
open(ILOG, ">fbitxt.log");

while (<IFH>) {
$target = $_;
#$target = "rzdspc77.informatik.uni-hamburg.de";
@outtarget = split /\n/, $target;
print $outtarget[0], "  ";
$out = ‘nslookup -sil -query=txt $target‘;
@lines = split /\n/, $out;
print ILOG $out;

if ($DEBUG) {
   $i=0;
   foreach $l (@lines) {
     print $i++," ", $l,"\n";
   }
   print "\n";
}

push (@result, $target);
if ($lines[3] =~ /No answer/) {
  print "no answer \n";
  push (@result, "No answer");
```

```
} else {
  print $lines[3], "\n";
  push (@result, $lines[3]);
}

}
close (IFH);
```

G9 ossnmp.pl

```
#!/usr/bin/perl -w
#
#   *** ossnmp.pl ***
#
#  Sep 2003, Nils Michaelsen
#
#  published under the terms of GPL
#
#  This program is expected to run with Win32 only !
#
#  tries to get os-version of many targets given in file "ip.in"
#  by use of snmp public community

if (@ARGV) {
  $infile = shift @ARGV;
  open (INFH, "<$infile");

  while (<INFH>) {
    $target = $_;
    @debug = split /\n/, $target;
    $target = $debug[0];
    $cmd = "snmputil walk $target public .1.3.6.1.2.1.1.1.";
    print $cmd, "\n";
    $result = `$cmd`;
    print $result, "\n\n";
    push @result, $target;
    push @result, " : ";
    push @result, $result;
    push @result, "\n";
  }
  close (INFH);
  open (OUTFH, ">snmp1.log");
  select OUTFH;
  foreach $r (@result) {
     print $r;
  }
  close (OUTFH);
} else {
   print "too few arguments\nossnmp <input ip-file>\n";
}
```

ANHANG H
GESETZESTEXTE

Dieser Anhang enthält die Gesetzestexte der in der Diplomarbeit verwendeten Paragraphen. Zum Zugangskontrolldiensteschutzgesetz (ZKDSG) siehe [ZKDSG02].

Verwendete Paragraphen des Bundesdatenschutzgesetzes (BDSG)

§5 Datengeheimnis

Den bei der Datenverarbeitung beschäftigten Personen ist untersagt, personenbezogene Daten unbefugt zu verarbeiten oder zu nutzen (Datengeheimnis). Diese Personen sind, soweit sie bei nicht-öffentlichen Stellen beschäftigt werden, bei der Aufnahme ihrer Tätigkeit auf das Datengeheimnis zu verpflichten. Das Datengeheimnis besteht auch nach Beendigung ihrer Tätigkeit fort.

§14 Datenspeicherung, -veränderung und –nutzung

1) Das Speichern, Verändern oder Nutzen personenbezogener Daten ist zulässig, wenn es zur Erfüllung der in der Zuständigkeit der speichernden Stelle liegenden Aufgaben erforderlich ist und es für die Zwecke erfolgt, für die die Daten erhoben worden sind. Ist keine Erhebung vorausgegangen, dürfen die Daten nur für die Zwecke geändert oder genutzt werden, für die sie gespeichert worden sind.

(2) Das Speichern, Verändern oder Nutzen für andere Zwecke ist nur zulässig, wenn

1. eine Rechtsvorschrift dies vorsieht oder zwingend voraussetzt,
2. der Betroffene eingewilligt hat,
3. offensichtlich ist, daß es im Interesse des Betroffenen liegt, und kein Grund zu der Annahme besteht, daß er in Kenntnis des anderen Zwecks seine Einwilligung verweigern würde,
4. Angaben des Betroffenen überprüft werden müssen, weil tatsächliche Anhaltspunkte für deren Unrichtigkeit bestehen,
5. die Daten aus allgemein zugänglichen Quellen entnommen werden können oder die speichernde Stelle sie veröffentlichen dürfte, es sei denn, daß das schutzwürdige Interesse des Betroffenen an dem Ausschluß der Zweckänderung offensichtlich überwiegt,
6. es zur Abwehr erheblicher Nachteile für das Gemeinwohl oder einer sonst unmittelbar drohenden Gefahr für die öffentliche Sicherheit erforderlich ist,

7. es zur Verfolgung von Straftaten oder Ordnungswidrigkeiten, zur Vollstreckung oder zum Vollzug von Strafen oder Maßnahmen im Sinne des § 11 Abs.1 Nr.8 des Strafgesetzbuches oder von Erziehungsmaßregeln oder Zuchtmitteln im Sinne des Jugendgerichtsgesetzes oder zur Vollstreckung von Bußgeldentscheidungen erforderlich ist,
8. es zur Abwehr einer schwerwiegenden Beeinträchtigung der Rechte einer anderen Person erforderlich ist oder
9. es zur Durchführung wissenschaftlicher Forschung erforderlich ist, das wissenschaftliche Interesse an der Durchführung des Forschungsvorhabens das Interesse des Betroffenen an dem Ausschluß der Zweckänderung erheblich überwiegt und der Zweck der Forschung auf andere Weise nicht oder nur mit unverhältnismäßigem Aufwand erreicht werden kann.

(3) Eine Verarbeitung oder Nutzung für andere Zwecke liegt nicht vor, wenn sie der Wahrnehmung von Aufsichts- und Kontrollbefugnissen, der Rechnungsprüfung oder der Durchführung von Organisationsuntersuchungen für die speichernde Stelle dient. Das gilt auch für die Verarbeitung oder Nutzung zu Ausbildungs- und Prüfungszwecken durch die speichernde Stelle, soweit nicht überwiegende schutzwürdige Interessen des Betroffenen entgegenstehen.

(4) Personenbezogene Daten, die ausschließlich zu Zwecken der Datenschutzkontrolle, der Datensicherung oder zur Sicherstellung eines ordnungsgemäßen Betriebes einer Datenverarbeitungsanlage gespeichert werden, dürfen nur für diese Zwecke verwendet werden.

Verwendete Paragraphen des Strafgesetzbuches (StGB)

§202a Ausspähen von Daten

(1) Wer unbefugt Daten, die nicht für ihn bestimmt und die gegen unberechtigten Zugang besonders gesichert sind, sich oder einem anderen verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

(2) Daten im Sinne des Absatzes 1 sind nur solche, die elektronisch, magnetisch oder sonst nicht unmittelbar wahrnehmbar gespeichert sind oder übermittelt werden.

§263a Computerbetrug

(1) Wer in der Absicht, sich oder einem Dritten einen rechtswidrigen Vermögensvorteil zu verschaffen, das Vermögen eines anderen dadurch beschädigt, daß er das Ergebnis eines Datenverarbeitungsvorgangs durch unrichtige Gestaltung des Programms, durch Verwendung unrichtiger oder unvollständiger Daten, durch unbefugte Verwendung von Daten oder sonst durch unbefugte Einwirkung auf den Ablauf beeinflußt, wird mit Freiheitsstrafe bis zu fünf Jahren oder mit Geldstrafe bestraft.

(2) § 263[99] Abs. 2 bis 7 gilt entsprechend.

[99] §263 StGB behandelt den Betrug im allgemeinen

§268 Fälschung technischer Aufzeichnungen

(1) Wer zur Täuschung im Rechtsverkehr

1. eine unechte technische Aufzeichnung herstellt oder eine technische Aufzeichnung verfälscht oder

2. eine unechte oder verfälschte technische Aufzeichnung gebraucht,

wird mit Freiheitsstrafe bis zu fünf Jahren oder mit Geldstrafe bestraft.

(2) Technische Aufzeichnung ist eine Darstellung von Daten, Meß- oder Rechenwerten, Zuständen oder Geschehensabläufen, die durch ein techniches Gerät ganz oder zum Teil selbsttätig bewirkt wird, den Gegenstand der Aufzeichnung allgemein oder für Eingeweihte erkennen läßt und zum Beweis einer rechtlich erheblichen Tatsache bestimmt ist, gleichviel ob ihr die Betimmung schon bei der Herstellung oder erst später gegeben wird.

(3) Der Herstellung einer unechten technischen Aufzeichnung steht es gleich, wenn der Täter durch störende Einwirkung auf den Aufzeichnungsvorgang das Ergebnis der Aufzeichnung beeinflußt.

(4) Der Versuch ist strafbar.

(5) § 267[100] Abs. 3 und 4 gilt entsprechend.

§269 Fälschung beweiserheblicher Daten

(1) Wer zur Täuschung im Rechtsverkehr beweiserhebliche Daten so speichert oder verändert, daß bei ihrer Wahrnehmung eine unechte oder verfälschte Urkunde vorliegen würde, oder derart gespeicherte oder veränderte Daten gebraucht, wird mit Freiheitsstrafe bis zu fünf Jahren oder mit Geldstrafe bestraft.

(2) Der Versuch ist strafbar.

(3) § 267[2] Abs. 3 und 4 gilt entsprechend.

§270 Täuschung im Rechtsverkehr

Der Täuschung im Rechtsverkehr steht die fälschliche Beeinflussung einer Datenverarbeitung im Rechtsverkehr gleich.

§303a Datenveränderung

(1) Wer rechtswidrig Daten (§ 202a Abs. 2) löscht, unterdrückt, unbrauchbar macht oder verändert, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft.

(2) Der Versuch ist strafbar.

[100] §267 StGB behandelt die Urkundenfälschung

§303b Computersabotage

(1) Wer eine Datenverarbeitung, die für einen fremden Betrieb, ein fremdes Unternehmen oder eine Behörde von wesentlicher Bedeutung ist, dadurch stört, daß er

1. eine Tat nach § 303a Abs. 1 begeht oder

2. eine Datenverarbeitungsanlage oder einen Datenträger zerstört, beschädigt, unbrauchbar macht, beseitigt oder verändert, wird mit Freiheitsstrafe bis zu fünf Jahren oder mit Geldstrafe bestraft.

(2) Der Versuch ist strafbar.

ANHANG I
REPORTS DER SCANNER

Der Anhang I enthält die Reports, die von den Scannern bei der Anwendung im Rahmen des Kapitels 6 generiert wurden. Die Reports von Sara und Stealth-http wurden auf Grund der fehlenden Aussagekraft vernachlässigt.

I1 Network Security Scanner

LANguard Network Security Scanner

Print this page

Scan target : **192.168.0.3** [**1** computers found]

IP Address	Details	Hostname	Username	Operating System
192.168.0.3		WWW3	WWW3	Windows 2000

192.168.0.3 [WWW3] Windows 2000

IP Address : **192.168.0.3**
Hostname : **WWW3**
Username : **WWW3**
MAC address : **00-04-75-CA-7A-BE** Vendor : 3 Com Corporation
LAN Manager : **Windows 2000 LAN Manager**
Domain : **ARBEITSGRUPPE**
Operating System : **Windows 2000**
Time to live : **128**

Browse list
WWW3 - Workstation Service
WWW3 - File Server Service
ARBEITSGRUPPE - Domain Name
INet~Services - IIS
WWW3 - Messenger Service
ARBEITSGRUPPE - Browser Service Elections
IS~WWW3 - Workstation Service

TCP ports - 8 open ports
21 [Ftp => File Transfer Protocol]
220 www3 Microsoft FTP Service (Version 5.0).
25 [Smtp => Simple Mail Transfer Protocol]
220 www3 Microsoft ESMTP MAIL Service, Version: 5.0.2172.1 ready at Tue, 29 Jul 2003 18:55:42 +0200
80 [Http => World Wide Web, HTTP]
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Tue, 29 Jul 2003 16:55:49 GMT
Content-Type: text/html
Content-Length: 79
135 [epmap => DCE endpoint resolution]
139 [Netbios-ssn => NETBIOS Session Service]
443 [HttpS => Secure HTTP]
445 [Microsoft-Ds]
3306 [MySQL]
+___3.23.56-nt_____y'SNkP.4_,_____

UDP ports - 4 open ports
135 [epmap => DCE endpoint resolution]
137 [Netbios-NS => Netbios Name Service]
138 [Netbios-DGM => Netbios Datagram Service]
445 [Microsoft CIFS => Common Internet File System]

Alerts
CGI abuses
Possible RDS exploit (msadcs.dll) RFP9902
Run arbitrary commands (SYSTEM level privileges)
http://www.securityfocus.com/bid/529

Frontpage check (1)
Frontpage extensions are installed on this computer

FTP alerts
FTP anonymous access allowed
It is recommended to disable anonymous logins

Informational alerts
MySQL (open source database) running
MySQL is running on this computer.

Dienstag, 29 Juli 2003 - 07:04 PM

Generated by **LANguard Network Security Scanner v(3.2)**
Copyright © 2001-2002 GFI Software Ltd.
www.gfisoftware.com/lannetscan
Pentest - Nils Michaelsen, 29.7.2003 - Start Scan from belzebub

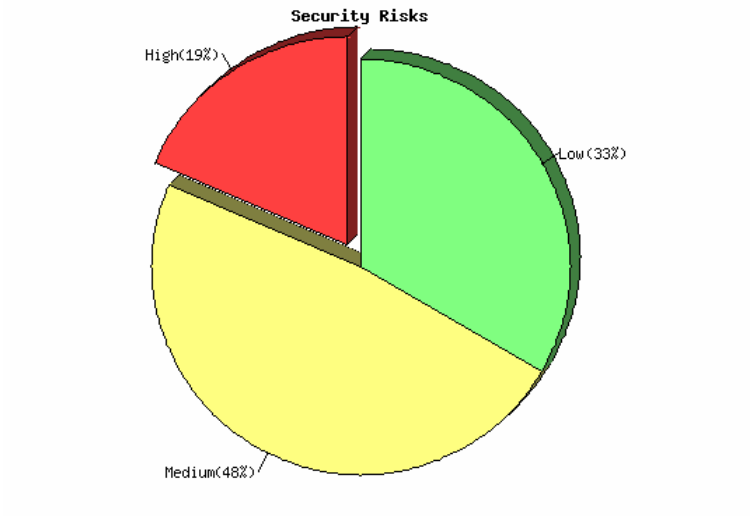
I2 Nessus

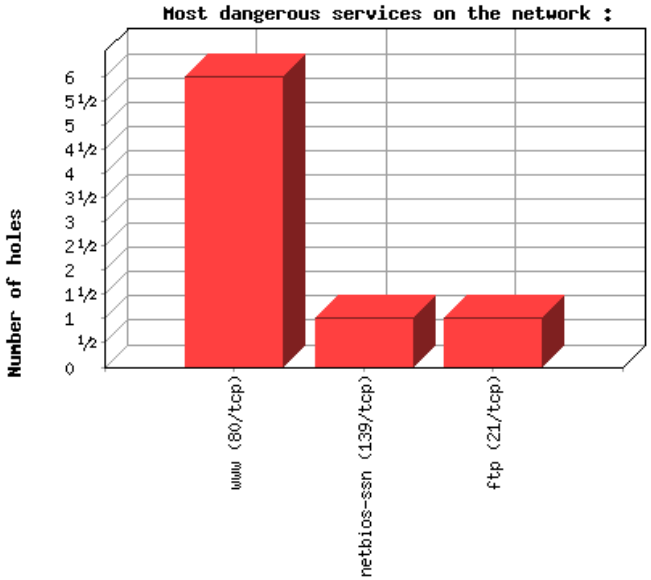
Nessus Report

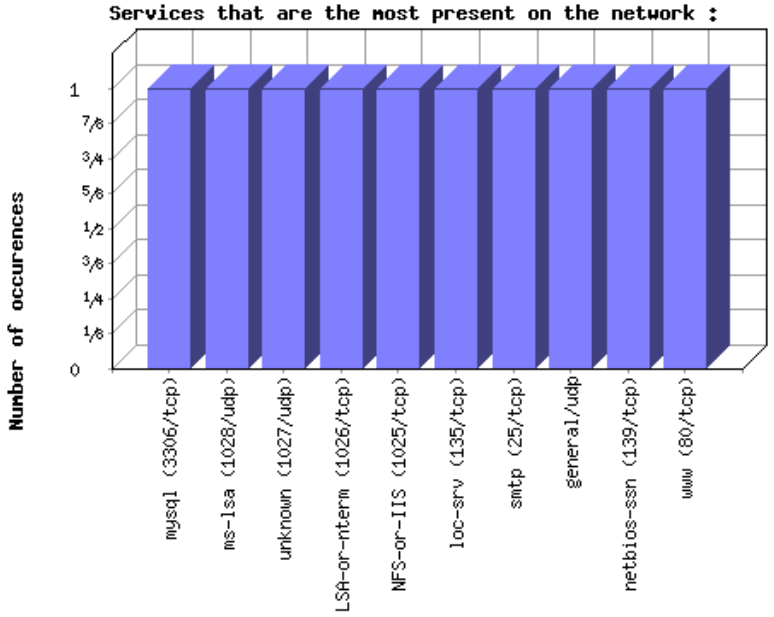
The Nessus Security Scanner was used to assess the security of 1 host

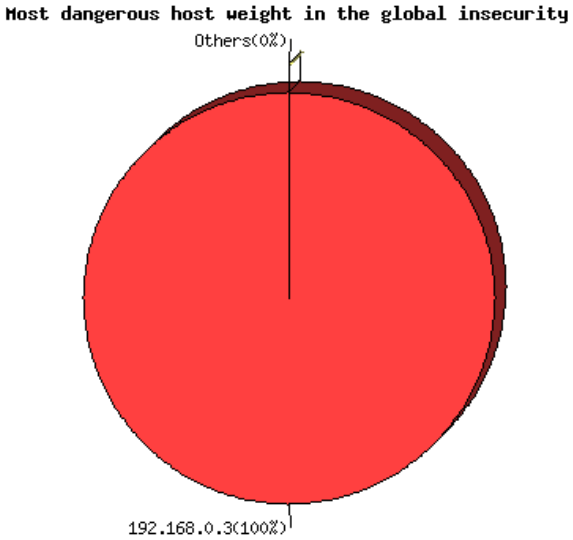
- **8 security holes have been found**
- **22 security warnings have been found**
- **13 security notes have been found**

Part I : Graphical Summary :









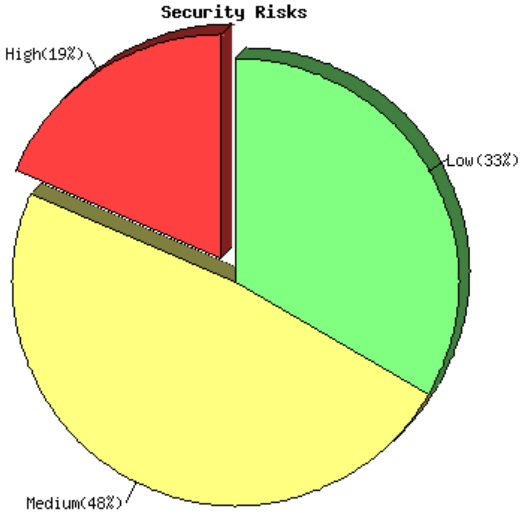
Part II. Results, by host :

Host name Notes 192.168.0.3
(found 8 security holes)

This file was generated by Nessus*, the open-sourced security scanner.*

192.168.0.3

Repartition of the level of the security problems :



[Back to the index]

List of open ports :

- *microsoft-ds (445/tcp)* **(Security notes found)**
- *netbios-ns (137/udp)* **(Security warnings found)**
- *ftp (21/tcp)* *(Security hole found)*
- *www (80/tcp)* *(Security hole found)*
- *netbios-ssn (139/tcp)* *(Security hole found)*
- *general/udp* **(Security notes found)**
- *smtp (25/tcp)* **(Security warnings found)**
- *loc-srv (135/tcp)* **(Security warnings found)**
- *NFS-or-IIS (1025/tcp)* **(Security notes found)**
- *LSA-or-nterm (1026/tcp)* **(Security notes found)**
- *unknown (1027/udp)* **(Security notes found)**
- *ms-lsa (1028/udp)* **(Security notes found)**
- *mysql (3306/tcp)* **(Security notes found)**

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

A CIFS server is running on this port
Nessus ID : 11011

[back to the list of ports]

Warning found on port netbios-ns (137/udp)

. The following 7 NetBIOS names have been gathered :
WWW3 = This is the computer name registered for workstation services by a WINS client.
WWW3
ARBEITSGRUPPE = Workgroup / Domain name
INet~Services = Workgroup / Domain name (Domain Controller)
WWW3 = Computer name that is registered for the messenger service on a computer that is a WINS client.
ARBEITSGRUPPE = Workgroup / Domain name (part of the Browser elections)
IS~WWW3 = This is the computer name registered for workstation services by a WINS client.
. The remote host has the following MAC address on its adapter :
0x00 0x04 0x75 0xca 0x7a 0xbe

If you do not want to allow everyone to find the NetBios name of your computer, you should filter incoming traffic to this port.

Risk factor : Medium
CVE : CAN-1999-0621
Nessus ID : 10150

[back to the list of ports

Vulnerability found on port ftp (21/tcp)

It may be possible to make the remote FTP server crash by sending the command 'STAT *?AAA...AAA.

An attacker may use this flaw to prevent your site from distributing files

*** Warning : we could not verify this vulnerability.
*** Nessus solely relied on the banner of this server

Solution : Apply the relevant hotfix from Microsoft

See:http://www.microsoft.com/technet/security/bulletin/ms02-018.asp

Risk factor : High
CVE : CVE-2002-0073
BID : 4482
Nessus ID : 10934

[back to the list of ports]

Warning found on port ftp (21/tcp)

This FTP service allows anonymous logins. If you do not
want to share data with anyone you do not know, then you should deactivate
the anonymous account, since it can only cause troubles.
Under most Unix system, doing :
echo ftp >> /etc/ftpusers
will correct this.

Risk factor : Low
CVE : CAN-1999-0497
Nessus ID : 10079

[back to the list of ports]

Information found on port ftp (21/tcp)

Remote FTP server banner :
220 www3 Microsoft FTP Service (Version 5.0).

Nessus ID : 10092

[back to the list of ports]

Vulnerability found on port www (80/tcp)

The remote IIS server allows anyone to execute arbitrary commands
by adding a unicode representation for the slash character
in the requested path.

Solution: See http://www.microsoft.com/technet/security/bulletin/ms00-078.asp
Risk factor : High
CVE : CVE-2000-0884
BID : 1806
Nessus ID : 10537

[back to the list of ports]

Vulnerability found on port www (80/tcp)

When IIS receives a user request to run a script, it renders
the request in a decoded canonical form, then performs
security checks on the decoded request. A vulnerability
results because a second, superfluous decoding pass is
performed after the initial security checks are completed.
Thus, a specially crafted request could allow an attacker to
execute arbitrary commands on the IIS Server.

Solution: See MS advisory MS01-026(Superseded by ms01-044)
See http://www.microsoft.com/technet/security/bulletin/ms01-044.asp

Risk factor : High
CVE : CVE-2001-0507, CVE-2001-0333
BID : 2708
Nessus ID : 10671

[back to the list of ports]

Vulnerability found on port www (80/tcp)

The web server is probably susceptible to a common IIS vulnerability discovered by
'Rain Forest Puppy'. This vulnerability enables an attacker to execute arbitrary
commands on the server with Administrator Privileges.

*** Nessus solely relied on the presence of the file /msadc/msadcs.dll
*** so this might be a false positive

See Microsoft security bulletin (MS99-025) for patch information.
Also, BUGTRAQ ID 529 on www.securityfocus.com (http://www.securityfocus.com/bid/529)

Risk factor : High
CVE : CVE-1999-1011
BID : 529
Nessus ID : 10357

[back to the list of ports]

Vulnerability found on port www (80/tcp)

It seems that the source code of various CGIs can be accessed by requesting the CGI name with a special suffix (.old, .bak, ~ or .copy)

Here is the list of CGIs Nessus gathered :
/script/produkte.php.bak

You should delete these files.
Nessus ID : 11411

[back to the list of ports]

Vulnerability found on port www (80/tcp)

IIS web server may allow remote users to read sensitive information from .cnf files. This is not the default configuration.

Example, http://target/_vti_pvt%5csvcacl.cnf, access.cnf, svcacl.cnf, writeto.cnf, service.cnf, botinfs.cnf, bots.cnf, linkinfo.cnf and services.cnf

See: http://www.safehack.com/Advisory/IIS5webdir.txt

Solution: If you do not need .cnf files, then delete them, otherwise use suitable access control lists to ensure that the .cnf files are not world-readable by Anonymous users.
Risk factor : Medium
BID : 4078
Nessus ID : 10575

[back to the list of ports]

Vulnerability found on port www (80/tcp)

The remote DLL /msadc/msadcs.dll is accessible by anyone. Several flaws have been found in it in the past, we recommand you restrict access to MSADC only to trusted hosts.

*** Nessus did not test for any security vulnerability
*** but solely relied on the presence of this resource
*** to issue this warning

Solution:
- Launch the Internet Services Manager
- Select your web server
- Right-click on MSADC and select 'Properties'
- Select the tab 'Directory Security'
- Click on the 'IP address and domain name restrictions' option
- Make sure that by default, all computers are DENIED access to this resource
- List the computers that should be allowed to use it

See also: MS advisory MS02-065
Risk factor: High
CVE : CAN-2002-1142
Nessus ID : 11161

[back to the list of ports]

Warning found on port www (80/tcp)

The remote server is running with WebDAV enabled.

WebDAV is an industry standard extension to the HTTP specification.
It adds a capability for authorized users to remotely add and manage
the content of a web server.

If you do not use this extension, you should disable it.

Solution : If you use IIS, refer to Microsoft KB article Q241520
Risk factor : Medium
Nessus ID : 11424

[back to the list of ports]

Warning found on port www (80/tcp)

The remote web server seems to be vulnerable to the Cross Site Scripting vulnerability (XSS). The vulnerability is caused
by the result returned to the user when a non-existing file is requested (e.g. the result contains the JavaScript provided
in the request).
The vulnerability would allow an attacker to make the server present the user with the attacker's JavaScript/HTML code.
Since the content is presented by the server, the user will give it the trust
level of the server (for example, the trust level of banks, shopping centers, etc. would usually be high).

Risk factor : Medium

Solutions:

. Allaire/Macromedia Jrun:
- http://www.macromedia.com/software/jrun/download/update/
- http://www.securiteam.com/windowsntfocus/Allaire_fixes_Cross-Site_Scripting_security_vulnerability.html
. Microsoft IIS:
- http://www.securiteam.com/windowsntfocus/IIS_Cross-Site_scripting_vulnerability__Patch_available_.html
. Apache:
- http://httpd.apache.org/info/css-security/
ColdFusion:
- http://www.macromedia.com/v1/handlers/index.cfm?ID=23047
. General:
.http://www.securiteam.com/exploits/Security_concerns_when_developing_a_dynamically_generated_web_site.html
- http://www.cert.org/advisories/CA-2000-02.html
BID : 5305, 7353, 7344, 8037
Nessus ID : 10815

[back the list of ports]

Warning found on port www (80/tcp)

Your webserver supports the TRACE and/or TRACK methods. It has been shown that servers supporting this method are subject to cross-site-scripting attacks, dubbed XST for 'Cross-Site-Tracing', when used in conjunction with various weaknesses in browsers.

An attacker may use this flaw to trick your legitimate web users to give him their credentials.

Solution: Disable these methods.

If you are using Apache, add the following lines for each virtual host in your configuration file :

RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
RewriteRule .* - [F]

If you are using Microsoft IIS, use the URLScan tool to deny HTTP TRACE requests or to permit only the methods needed to meet site requirements and policy.

See http://www.whitehatsec.com/press_releases/WH-PR-20030120.pdf
http://archives.neohapsis.com/archives/vulnwatch/2003-q1/0035.html

Risk factor : Medium
Nessus ID : 11213

[back to the list of ports]

Warning found on port www (80/tcp)

/base/webmail/readmsg.php was detected.
Some versions of this CGI allow remote users to read local files with the permission of the web server.
Note that if the user has a shell access, this kind of attack is not interesting.

*** Nessus just checked the presence of this file
*** but did not try to exploit the flaw.

Solution : get a newer software from Cobalt

Reference : http://online.securityfocus.com/archive/1/195165

Risk factor : Low
CVE : CAN-2001-1408
Nessus ID : 11073

[back to the list of ports]

Warning found on port www (80/tcp)

A sample application shipped with IIS 5.0 discloses the physical path of the web root. An attacker can use this information to make more focused attacks.

Solution: Always remove sample applications from productions servers.
In this case, remove the entire /iissamples folder.
Risk factor : Low
Nessus ID : 10573

[back to the list of ports]

Warning found on port www (80/tcp)

This IIS Server appears to vulnerable to one of the cross site scripting attacks described in MS02-018. The default '404' file returned by IIS uses scripting to output a link to
top level domain part of the url requested. By crafting a particular URL it is possible to insert arbitrary script into the
page for execution.

The presence of this vulnerability also indicates that you are vulnerable to the other issues identified in MS02-018 (various remote buffer overflow and cross site scripting attacks...)

References:
http://www.microsoft.com/technet/security/bulletin/MS02-018.asp
http://jscript.dk/adv/TL001/

Risk factor : Medium
CVE : CVE-2002-0074
BID : 4483
Nessus ID : 10936

[back to the list of ports]

Warning found on port www (80/tcp)

This IIS Server appears to be vulnerable to a Cross
Site Scripting due to an error in the handling of overlong requests on
an idc file. It is possible to inject Javascript
in the URL, that will appear in the resulting page.

Risk factor : Medium

See also : http://online.securityfocus.com/bid/5900
http://www.ntbugtraq.com/default.asp?pid=36&sid=1&A2=ind0210&L=ntbugtraq&F=P&S=&P=1391

BID : 5900
Nessus ID : 11142

[back to the list of ports]

Warning found on port www (80/tcp)

The IIS server appears to have the .IDA ISAPI filter mapped.

At least one remote vulnerability has been discovered for the .IDA (indexing service) filter. This is detailed in Microsoft Advisory MS01-033, and gives remote SYSTEM level access to the web server.

It is recommended that even if you have patched this vulnerability that you unmap the .IDA extension, and any other unused ISAPI extensions if they are not required for the operation of your site.

Solution:
To unmap the .IDA extension:
1.Open Internet Services Manager.
2.Right-click the Web server choose Properties from the context menu.
3.Master Properties
4.Select WWW Service -> Edit -> HomeDirectory -> Configuration and remove the reference to .ida from the list.

Risk factor : Medium
CVE : CVE-2001-0500
BID : 2880
Nessus ID : 10695

[back to the list of ports]

Warning found on port www (80/tcp)

IIS 4.0 allows a remote attacker to obtain the real pathname of the document root by requesting non-existent files with .ida or .idq extensions.

An attacker may use this flaw to gain more information about the remote host, and hence make more focused attacks.

Solution: Select 'Preferences ->Home directory ->Application', and check the checkbox 'Check if file exists' for the ISAPI mappings of your server.

Risk factor : Low
CVE : CAN-2000-0071
BID : 1065
Nessus ID : 10492

[back to the list of ports]

Warning found on port www (80/tcp)

The script /iissamples/sdk/asp/interaction/Form_JScript.asp (or Form_VBScript.asp) allows you to insert information into a form field and once submitted re-displays the page, printing the text you entered. This .asp doesn't perform any input validation, and hence you can input a string like:
<SCRIPT>alert(document.domain)</SCRIPT>.

More information on cross-site scripting attacks can be found at:

http://www.cert.org/advisories/CA-2000-02.html

Solution: Always remove sample applications from productions servers. In this case, remove the entire /iissamples folder.
Risk factor : Low
Nessus ID : 10572

[back to the list of ports]

Information found on port www (80/tcp)

The following directories were discovered:
/_private, /_vti_log, /iissamples, /images, /script, /service, /webpub
The following directories require authentication:
/printers
Nessus ID : 11032

[back to the list of ports]

Information found on port www (80/tcp)

The remote web server type is :

Microsoft-IIS/5.0

Solution : You can use urlscan to change reported server for IIS.
Nessus ID : 10107

[back to the list of ports]

Vulnerability found on port netbios-ssn (139/tcp)

. It was possible to log into the remote host using a NULL session. The concept of a NULL session is to provide a null username and a null password, which grants the user the 'guest' access

To prevent null sessions, see MS KB Article Q143474 (NT 4.0) and Q246261 (Windows 2000).
Note that this won't completely disable null sessions, but will prevent them from connecting to IPC$
Please see http://msgs.securepoint.com/cgi-bin/get/nessus-0204/50/1.html

. All the smb tests will be done as "/" in domain ARBEITSGRUPPE
CVE : CAN-1999-0504, CAN-1999-0506, CVE-2000-0222
Nessus ID : 10394

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

The host SID can be obtained remotely. Its value is :

WWW3 : 5-21-1645522239-436374069-1343024091

An attacker can use it to obtain the list of the local users of this host
Solution : filter the ports 137 to 139 and 445
Risk factor : Low

CVE : CVE-2000-1200
BID : 959
Nessus ID : 10859

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

The host SID could be used to enumerate the names of the local users of this host.
(we only enumerated users name whose ID is between 1000 and 2000 for performance reasons)
This gives extra knowledge to an attacker, which is not a good thing :
- Administrator account name : Administrator (id 500)
- Guest account name : Gast (id 501)
- IUSR_WWW3 (id 1000)
- IWAM_WWW3 (id 1001)

Risk factor : Medium
Solution : filter incoming connections this port

CVE : CVE-2000-1200
BID : 959
Nessus ID : 10860

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

The following local accounts have never logged in :

Gast

Unused accounts are very helpful to hacker
Solution : suppress these accounts
Risk factor : Medium
Nessus ID : 10915

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

The following local accounts have never changed their password :

IUSR_WWW3
IWAM_WWW3

To minimize the risk of break-in, users should change their password regularly
Nessus ID : 10914

[back to the list of ports]

Warning found on port netbios-ssn (139/tcp)

The following local accounts have passwords which never expire :

Administrator
Gast
IUSR_WWW3
IWAM_WWW3

Password should have a limited lifetime
Solution : disable password non-expiry
Risk factor : Medium
Nessus ID : 10916

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

The remote native lan manager is : Windows 2000 LAN Manager
The remote Operating System is : Windows 5.0
The remote SMB Domain Name is : ARBEITSGRUPPE

Nessus ID : 10785

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

The following local accounts are disabled :

Gast

To minimize the risk of break-in, permanently disabled accounts
should be deleted
Risk factor : Low
Nessus ID : 10913

[back to the list of ports]

Information found on port general/udp

For your information, here is the traceroute to 192.168.0.3 :
192.168.0.3

Nessus ID : 10287

[back to the list of ports]

Warning found on port smtp (25/tcp)

The remote SMTP server is vulnerable to a flaw in its authentication process.

This vulnerability allows any unauthorized user to successfully authenticate and use the remote SMTP server.

An attacker may use this flaw to use this SMTP server
as a spam relay.

Solution : see http://www.microsoft.com/technet/security/bulletin/MS01-037.asp.

Risk factor : High
CVE : CVE-2001-0504
BID : 2988
Nessus ID : 10703

[back to the list of ports]

Warning found on port smtp (25/tcp)

It is possible to make the remote SMTP server fail
and restart by sending it malformed input.

The service will restart automatically, but all the connections
established at the time of the attack will be dropped.

An attacker may use this flaw to make mail delivery to your site
less efficient.

Solution : http://www.microsoft.com/technet/security/bulletin/MS02-012.asp
Risk factor : Medium
CVE : CVE-2002-0055
BID : 4204
Nessus ID : 10885

[back to the list of ports]

Warning found on port smtp (25/tcp)

It is possible to authenticate to the remote SMTP service
by logging in as a NULL session.

An attacker may use this flaw to use your SMTP server as a
spam relay.

Solution : http://www.microsoft.com/technet/security/bulletin/MS02-011.asp
Risk factor : Medium
CVE : CVE-2002-0054
BID : 4205
Nessus ID : 11308

[back to the list of ports]

Information found on port smtp (25/tcp)

This server could be fingerprinted as being Microsoft ESMTP MAIL
Service, Version 5.0.2195
Nessus ID : 11421

[back to the list of ports]

Warning found on port loc-srv (135/tcp)

DCE services running on the remote can be enumerated
by connecting on port 135 and doing the appropriate
queries.

An attacker may use this fact to gain more knowledge
about the remote host.

Solution : filter incoming traffic to this port.
Risk factor : Low
Nessus ID : 10736

[back to the list of ports]

Information found on port NFS-or-IIS (1025/tcp)

Here is the list of DCE services running on this port:
UUID: 1ff70682-0a51-30e8-076d-740be8cee98b, version 1
Endpoint: ncacn_ip_tcp:192.168.0.3[1025]

UUID: 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1
Endpoint: ncacn_ip_tcp:192.168.0.3[1025]

Nessus ID : 10736

[back to the list of ports]

Information found on port LSA-or-nterm (1026/tcp)

Here is the list of DCE services running on this port:
UUID: 82ad4280-036b-11cf-972c-00aa006887b0, version 2
Endpoint: ncacn_ip_tcp:192.168.0.3[1026]

UUID: 8cfb5d70-31a4-11cf-a7d8-00805f48a135, version 3
Endpoint: ncacn_ip_tcp:192.168.0.3[1026]

UUID: bfa951d1-2f0e-11d3-bfd1-00c04fa3490a, version 1
Endpoint: ncacn_ip_tcp:192.168.0.3[1026]

Nessus ID : 10736

[back to the list of ports]

Information found on port unknown (1027/udp)

Here is the list of DCE services running on this port:
UUID: 5a7b91f8-ff00-11d0-a9b2-00c04fb6e6fc, version 1
Endpoint: ncadg_ip_udp:192.168.0.3[1027]
Annotation: Messenger Service

Nessus ID : 10736

[back to the list of ports]

Information found on port ms-lsa (1028/udp)

Here is the list of DCE services running on this port:
UUID: bfa951d1-2f0e-11d3-bfd1-00c04fa3490a, version 1
Endpoint: ncadg_ip_udp:192.168.0.3[1028]

Nessus ID : 10736

[back to the list of ports]

Information found on port mysql (3306/tcp)

Remote MySQL version : 3.23.56-nt
Nessus ID : 10719

[back to the list of ports]

This file was generated by Nessus, the open-sourced security scanner.

I3 Cerberus Internet Scanner

NetBIOS Session Service

Share Information

Share Name :E$
Share Type :Default Disk Share
Comment :Standardfreigabe

Share Name :IPC$
Share Type :Default Pipe Share
Comment :Remote-IPC

WARNING - Null session can be established to \\192.168.0.3\IPC$

Share Name :F$
Share Type :Default Disk Share
Comment :Standardfreigabe

Share Name :ADMIN$
Share Type :Default Disk Share
Comment :Remoteadmin

Share Name :C$
Share Type :Default Disk Share
Comment :Standardfreigabe

Group Information

Group Name : **Kein**
Users
Administrator
Gast
IUSR_WWW3
IWAM_WWW3

Account Information

Account Name :Administrator

The Administrator account is an ADMINISTRATOR, and the password was changed 5 days ago. This account has been used 4 times to logon.
The default Administrator account has not been renamed. Consider renaming this account
and removing most of its rights. Use a different account as the admin account.

Comment :Vordefiniertes Konto für die Verwaltung des Computers bzw. der Domäne
User Comment :
Full name :

Account Name :Gast

The Gast account is a GUEST, and the password was
changed 0 days ago. This account has been used 0 times to logon.

Comment :Vordefiniertes Konto für Gastzugriff auf den Computer bzw. die Domäne
User Comment :
Full name :

Account Name :IUSR_WWW3

The IUSR_WWW3 account is a GUEST, and the password was changed 5 days ago. This account has been used 0 times to logon.

Comment :Vordefiniertes Konto für anonymen Zugang zu Internet-Informationsdienste
User Comment :Vordefiniertes Konto für anonymen Zugang zu Internet-Informationsdienste
Full name :Internetgastkonto

Account Name :IWAM_WWW3

The IWAM_WWW3 account is a GUEST, and the password was changed 5 days ago. This account has been used 2 times to logon.

Comment :Vordefiniertes Konto für Internet-Informationsdienste aus Prozessanwendungen heraus starten
User Comment :Vordefiniertes Konto für Internet-Informationsdienste aus Prozessanwendungen heraus starten
Full name :IIS-Prozesskonto starten

I4 Nikto

- Nikto v1.30/1.14

+ Target IP: 192.168.0.3
+ Target Hostname:
+ Target Port: 80
+ Start Time: Wed Jul 30 15:34:50 2003

- Scan is dependent on "Server" string which can be faked, use -g to override
+ Server: Microsoft-IIS/5.0
- Retrieved X-Powered-By header: PHP/4.3.2RC3
+ IIS may reveal its internal IP in the Content-Location header. The value is "http://192.168.0.3/index.html". http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0649.
+ Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK
+ HTTP method 'PROPFIND' may indicate DAV/WebDAV is installed. This may be used to get directory listings if indexing is allowed but a default page exists.
+ HTTP method 'SEARCH' may be used to get directory listings if Index Server is running.
+ HTTP method 'TRACE' is typically only used for debugging. It should be disabled.
+ Microsoft-IIS/5.0 is outdated if server is Win2000 (4.0 is current for NT 4)
+ / - TRACE option appears to allow XSS or credential theft. See http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACE)
+ / - TRACK option ('TRACE' alias) appears to allow XSS or credential theft. See http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACK)
+ /<script>alert('Vulnerable')</script>.shtml - Server is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html. (GET)
+ /admin.php - This might be interesting... (GET)
+ /admin.php?en_log_id=0&action=config - EasyNews from http://www.webrc.ca version 4.3 allows remote admin access. This php file should be protected. (GET)

+ /admin.php?en_log_id=0&action=users - EasyNews from http://www.webrc.ca version 4.3 allows remote admin access. This php file should be protected. (GET)
+ /admin/login.php?action=insert&username=test&password=test - phpAuction may allow user admin accounts to be inserted without proper authentication. Attempt to log in with user 'test' password 'test' to verify. (GET)
+ /admin/phpinfo.php - Immobilier allows phpinfo to be run. See http://www.frog-man.org/tutos/Immoblier.txt (GET)
+ /admin/phpinfo.php - phPay v2.02 information disclosure via phpInfo() script. http://phpay.sourceforge.net/. (GET)
+ /admin/system_footer.php - myphpnuke version 1.8.8_final_7 reveals detailed system information. (GET)
+ /administrator/gallery/uploadimage.php - Mambo PHP Portal/Server 4.0.12 BETA and below may allow upload of any file type simply putting '.jpg' before the real file extension. (GET)
+ /agentadmin.php - Immobilier may allow php files to be included from remote sites. See http://www.frog-man.org/tutos/Immoblier.txt (GET)
+ /anthill/login.php - Anthill bug tracking system may be installed. Versions lower than 0.1.6.1 allow XSS/HTML injection and may allow users to bypass login requirements. http://anthill.vmlinuz.ca/ and http://www.cert.org/advisories/CA-2000-02.html (GET)

+ /b2-include/b2edit.showposts.php - Some versions of B2 (cafelog.com) are vulnerable to remote inclusion by redefining $b2inc to a remote php file. Upgrade to a version higher than b2.06pre2. This vulnerability could not be confirmed. (GET)
+ /catalog/includes/include_once.php - This phpWebSite script may allow inclusion of remote scripts by adding ?inc_prefix=http://YOURHOST/ (GET)
+ /cbms/cbmsfoot.php - CBMS Billing Management has had many vulnerabilities in versions 0.7.1 and below. none could be confirmed here, but they should be manually checked if possible. http://freshmeat.net/projects/cbms/ (GET)
+ /cbms/changepass.php - CBMS Billing Management has had many vulnerabilities in versions 0.7.1 and below. none could be confirmed here, but they should be manually checked if possible. http://freshmeat.net/projects/cbms/ (GET)
+ /cbms/editclient.php - CBMS Billing Management has had many vulnerabilities in versions 0.7.1 and below. none could be confirmed here, but they should be manually checked if possible. http://freshmeat.net/projects/cbms/ (GET)
+ /cbms/passgen.php - CBMS Billing Management has had many vulnerabilities in versions 0.7.1 and below. none could be confirmed here, but they should be manually checked if possible. http://freshmeat.net/projects/cbms/ (GET)
+ /cbms/usersetup.php - CBMS Billing Management has had many vulnerabilities in versions 0.7.1 and below. none could be confirmed here, but they should be manually checked if possible. http://freshmeat.net/projects/cbms/ (GET)
+ /config.php - PHP Config file may contain database IDs and passwords. (GET)
+ /contents.php?new_language=elvish&mode=select - Requesting a file with an invalid language selection from DC Portal may reveal the system path. (GET)
+ /dostuff.php?action=modify_user - Blahz-DNS allows unauthorized users to edit user information. Upgrade to version 0.25 or higher. http://blahzdns.sourceforge.net/ (GET)
+ /filemanager/filemanager_forms.php - Some versions of PHProjekt allow remote file inclusions. Verify the current version is running. See http://www.securiteam.com/unixfocus/5PP0F1P6KS.html for more info (GET)
+ /forums/@ADMINDIRSconfig.php - PHP Config file may contain database IDs and passwords. (GET)
+ /forums/config.php - PHP Config file may contain database IDs and passwords. (GET)
+ /gb/index.php?login=true - gBook may allow admin login by setting the value 'login' equal to 'true'. (GET)
+ /guestbook/admin.php - Guestbook admin page available without authentication. (GET)
+ /inc/common.load.php - Bookmark4U v1.8.3 include files are not protected, and may contain remote source injection by using the 'prefix' variable. (GET)
+ /inc/config.php - Bookmark4U v1.8.3 include files are not protected, and may contain remote source injection by using the 'prefix' variable. (GET)

+ Over 30 "OK" messages, this may be a by-product of the
    + server answering all requests with a "200 OK" message. You should
    + manually verify your results.
+ /inc/dbase.php - Bookmark4U v1.8.3 include files are not protected, and may contain remote source injection by using the 'prefix' variable. (GET)
+ /instantwebmail/message.php - Instant Web Mail (http://understroem.kdc/instantwebmail/) is installed. Versions 0.59 and lower can allow remote users to embed POP3 commands in URLs contained in email. (GET)
+ /mambo/administrator/phpinfo.php - Mambo Site Server 4.0.11 phpinfo.php script reveals system information. (GET)
+ /manual.php - Does not filter input before passing to shell command. Try 'ls -l' as the man page entry. (GET)
+ /modsecurity.php - This phpWebSite script may allow inclusion of remote scripts by adding ?inc_prefix=http://YOURHOST/ (GET)

+ /modules.php?name=Members_List&sql_debug=1 - The PHP-Nuke install may allow attackers to enable debug mode and disclose sensitive information by adding sql_debug=1 to the query string. (GET)
+ /php/index.php - Monkey Http Daemon default php file found. (GET)
+ /phpBB/phpinfo.php - phpBBmod contains an enhanced version of the phpinfo.php script. This should be removed as it contains detailed system information. (GET)
+ /phpBB2/includes/db.php - Some versions of db.php from phpBB2 allow remote file inclusions. Verify the current version is running. See http://www.securiteam.com/securitynews/5BP0F2A6KC.html for more info (GET)
+ /phpEventCalendar/file_upload.php - phpEventCalendar 1.1 and prior vulnerable to file upload bug. (GET)
+ /phpinfo.php - Contains PHP configuration information (GET)
+ /phpshare/phpshare.php - Several serious security holes pre 0.6b2. Several minor security holes pre 0.6b3 (GET)
+ /project/index.php?m=projects&user_cookie=1 - dotProject 0.2.1.5 may allow admin login bypass by adding the user_cookie=1 to the URL. (GET)
+ /pvote/ch_info.php - PVote administration page is available. Versions 1.5b and lower do not require authentication to reset the administration password. (GET)
+ /simplebbs/users/users.php - Simple BBS 1.0.6 allows user information and passwords to be viewed remotely. (GET)
+ /smssend.php - PhpSmssend may allow system calls if a ' is passed to it. http://zekiller.skytech.org/smssend.php (GET)
+ /splashAdmin.php - Cobalt Qube 3 admin is running. This may have multiple security problems as described by www.scan-associates.net. These could not be tested remotely. (GET)
+ /thebox/admin.php?act=write&username=admin&password=admin&aduser=admin&adpass=admin - paBox 1.6 may allow remote users to set the admin password. If successful, the 'admin' password is now 'admin'. (GET)
+ /typo3conf/localconf.php - Typo3 config file found. (GET)
+ /userlog.php - Teekai's Tracking Online 1.0 log can be retrieved remotely. (GET)
+ /wikihome/action/conflict.php - Some versions of this script allow external source to be included/run by appending ?TemplateDir=http://my.host/ to requests. (GET)
+ /admin/config.php - PHP Config file may contain database IDs and passwords. (GET)
+ /adm/config.php - PHP Config file may contain database IDs and passwords. (GET)
+ /webcgi//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgi-914//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgi-915//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /bin//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgi//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /mpcgi//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgi-bin//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgi-sys//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgi-local//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /htbin//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgibin//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgis//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /scripts//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /cgi-win//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /fcgi-bin//nimages.php - Alpha versions of the Nimages package vulnerable to non specific 'major' security bugs. (GET)
+ /webcgi/admin.php - This might be interesting... (GET)
+ /cgi-914/admin.php - This might be interesting... (GET)
+ /cgi-915/admin.php - This might be interesting... (GET)
+ /bin/admin.php - This might be interesting... (GET)
+ /cgi/admin.php - This might be interesting... (GET)
+ /mpcgi/admin.php - This might be interesting... (GET)
+ /cgi-bin/admin.php - This might be interesting... (GET)
+ /cgi-sys/admin.php - This might be interesting... (GET)
+ /cgi-local/admin.php - This might be interesting... (GET)
+ /htbin/admin.php - This might be interesting... (GET)
+ /cgibin/admin.php - This might be interesting... (GET)
+ /cgis/admin.php - This might be interesting... (GET)

+ /scripts/admin.php - This might be interesting... (GET)
+ /cgi-win/admin.php - This might be interesting... (GET)
+ /fcgi-bin/admin.php - This might be interesting... (GET)
+ /webcgi/code.php - This might be interesting... (GET)
+ /cgi-914/code.php - This might be interesting... (GET)
+ /cgi-915/code.php - This might be interesting... (GET)
+ /bin/code.php - This might be interesting... (GET)
+ /cgi/code.php - This might be interesting... (GET)
+ /mpcgi/code.php - This might be interesting... (GET)
+ /cgi-bin/code.php - This might be interesting... (GET)
+ /cgi-sys/code.php - This might be interesting... (GET)
+ /cgi-local/code.php - This might be interesting... (GET)
+ /htbin/code.php - This might be interesting... (GET)
+ /cgibin/code.php - This might be interesting... (GET)
+ /cgis/code.php - This might be interesting... (GET)
+ /scripts/code.php - This might be interesting... (GET)
+ /cgi-win/code.php - This might be interesting... (GET)
+ /fcgi-bin/code.php - This might be interesting... (GET)
+ /blahb.ida - Reveals physical path. To fix: Preferences -> Home directory -> Application & check 'Check if file exists' for the ISAPI mappings. http://www.microsoft.com/technet/security/bulletin/MS01-033.asp. (GET)
+ /blahb.idq - Reveals physical path. To fix: Preferences -> Home directory -> Application & check 'Check if file exists' for the ISAPI mappings. http://www.microsoft.com/technet/security/bulletin/MS01-033.asp. (GET)
+ /iissamples/sdk/asp/docs/codebrws.asp - This is a default IIS script/file which should be removed. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0738. http://www.microsoft.com/technet/security/bulletin/MS99-013.asp. (GET)
+ /xxxxx.htw - Server may be vulnerable to a Webhits.dll arbitrary file retrieval. Ensure Q252463i, Q252463a or Q251170 is installed. http://www.microsoft.com/technet/security/bulletin/MS00-006.asp. (GET)
+ /msadc/../../../../winnt/system32/cmd.exe?/c+dir - IIS is vulnerable to a double-decode bug, which allows commands to be executed on the system. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333. http://www.securityfocus.com/bid/BID-2708. (GET)
+ /msadc/../../../../winnt/system32/cmd.exe?/c+dir - IIS Unicode command exec problem, see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884 (GET)
+ /msadc/msadcs.dll - See RDS advisory RFP9902, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1011, http://www.microsoft.com/technet/security/bulletin/MS98-004.asp, http://www.microsoft.com/technet/security/bulletin/MS99-025.asp RFP-9902 BID-29 (http://www.wiretrip.net/rfp/p/doc.asp/i2/d1.htm), CIAC J-054 http://www.ciac.org/ciac/bulletins/j-054.shtml www.securityfocus.com/bid/529 (GET)
+ /msadc/msadcs.dll - See RDS advisory, RFP9902 (wiretrip.net), http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1011, http://www.microsoft.com/technet/security/bulletin/MS98-004.asp, http://www.microsoft.com/technet/security/bulletin/MS99-025.asp, CIAC:J-054, ISS 19990809, BID-529 (GET)
+ /scripts/../../winnt/system32/cmd.exe?/c+dir - IIS is vulnerable to a double-decode bug, which allows commands to be executed on the system. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0333. http://www.securityfocus.com/bid/BID-2708. (GET)
+ /scripts/../../winnt/system32/cmd.exe?/c+dir - IIS Unicode command exec problem, see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884 (GET)
+ /scripts/samples/search/qfullhit.htw - Server may be vulnerable to a Webhits.dll arbitrary file retrieval. http://www.microsoft.com/technet/security/bulletin/MS00-006.asp. (GET)
+ /scripts/samples/search/qsumrhit.htw - Server may be vulnerable to a Webhits.dll arbitrary file retrieval. http://www.microsoft.com/technet/security/bulletin/MS00-006.asp. (GET)
+ /_vti_inf.html - FrontPage may be installed. (GET)
+ /_vti_pvt/access.cnf - Contains HTTP server-specific access control information, remove or ACL if FrontPage is not being used. (GET)
+ /_vti_pvt/linkinfo.cnf - IIS file shows http links on and off site. Might show host trust relationships and other machines on network. (GET)
+ /_vti_pvt/service.cnf - Contains meta-information about the web server, remove or ACL if FrontPage is not being used. (GET)
+ /_vti_pvt/services.cnf - Contains the list of subwebs, remove or ACL if FrontPage is not being used. May reveal server version if Admin has changed it. (GET)
+ /_vti_pvt/writeto.cnf - Contains information about form handler result files, remove or ACL if FrontPage is not being used. (GET)

```
+ Over 30 "OK" messages, this may be a by-product of the
        +   server answering all requests with a "200 OK" message. You should
        +   manually verify your results.
+ 11133 items checked - 117 items found on remote host
+ End Time:       Wed Jul 30 15:36:52 2003 (122 seconds)
---------------------------------------------------------------------------

Test Options: -allcgi -host 192.168.0.3 -output nikto.txt
```

Anmerkung des Autors:

Der von Nikto generierte Bericht wurde geändert, da bestimmte Einträge eine Nachahmung von Angriffen ermöglichen.

ANHANG J

S e r v i c e L e v e l A g r e e m e n t

Die folgenden Paragraphen bilden ein Service Level Agreement (im Folgenden SLA genannt) zwischen Nils Michaelsen und dem Rechenzentrum des Fachbereiches Informatik, vertreten durch Herrn Dr. Hans-Joachim Mück, über eine Reconnaissance im Rahmen der von Nils Michaelsen anzufertigen Diplomarbeit mit dem Titel „Penetrationstest: Möglichkeiten und Grenzen“ (im Folgenden Diplomarbeit genannt)

§ 1 Definitionen

Definitionen sind der Diplomarbeit zu entnehmen.

§ 2 Ziele

I. Das Ziel der Reconnaissance ist die Betrachtung der Wirksamkeit der Maßnahmen, die einen Angreifer an der Beschaffung von Informationen hindern. Als Angreifer wird eine Person angenommen, welche die Intention verfolgt, sich über das Internet Zugriff auf das Rechenzentrum des Fachbereichs Informatik zu verschaffen.

II. Außer der Verwendung gleicher Tools soll die Reconnaissance keine weiteren Eigenschaften eines Angreifers wie die Verwendung bestimmte Zeitfenster oder Geschwindigkeit der durchzuführenden Aktionen nachahmen.

§ 3 Durchführung, Abdeckung und Standort

I. Die Reconnaissance wird von Nils Michaelsen durchgeführt

II. Die Reconnaissance deckt alle zur Domain „informatik.uni-hamburg.de“ gehörenden Systeme ab.

III. Die Reconnaissance wird über einen ADSL-Anschluss der Firma Hansenet aus dem Wohnbereich von Nils Michelsen durchgeführt.

§ 4 Bekanntmachung und Wissen

I. Der Test ist nur Nils Michaelsen Sowie den Betreuern Herrn Prof. Dr. Klaus Brunnstein und Herrn Dr. Hans-Joachim Mück bekannt.

II. Die Domain „informatik.uni-hamburg.de“ ist die einzige Information, die Nils Michaelsen zur Verfügung gestellt wurde.

§ 5 Auswirkung

I. Die Vertraulichkeit sämtlicher auf den Systemen des Fachbereichs Informatik gespeicherter Daten des darf durch die Reconnaissance nicht geschädigt werden. Ausnahmen sind nicht möglich.
II. Absatz I gilt entsprechend für die Integrität.

III. Die Verfügbarkeit der in Absatz I genannten Daten sowie aller Systeme des Fachbereiches Informatik darf zu keinem Zeitpunkt gestört werden. Ausnahmen sind nicht möglich.

§ 6 Kontrolle

I. Datum, Uhrzeit und die jeweilige öffentliche IP-Adresse, die Nils Michaelsen durch die Firma Hansenet zugeordnet worden ist, sind zu dokumentieren.

II. Sämtliche Aktionen, die bei der Reconnaissance durchgeführt werden, sind Herrn Dr. Mück vorher per Email anzuzeigen. In der Email ist die in Absatz I erwähnte IP-Adresse anzugeben.

III. Herr Dr. Mück übernimmt die Funktion des „Cutouts“.

§ 7 Ergebnisse

I. Die Ergebnisse dürfen nur zu der Erstellung der Diplomarbeit verwendet werden.

II. Sämtliche Ergebnisse sind vertraulich zu behandeln und sind Herrn Dr. Mück zu melden.

III. Ergebnisse dürfen nur mit Einverständnis von Herrn Dr. Mück in der Diplomarbeit veröffentlicht werden. Vertrauliche Ergebnisse werden in einem vertraulichen Anhang dokumentiert, der nur Nils Michaelsen, Herrn Prof. Dr. Brunnstein und Herrn Dr. Mück zugänglich gemacht werden darf.

§ 8 Haftung und Nichtdurchführung

I. Nils Michaelsen übernimmt keinerlei Haftung

II. Eine Nichtdurchführung wird in die Bewertung der Diplomarbeit berücksichtigt werden.

Hamburg, den _____

_____ _____

(Nils Michaelsen) (Dr. Mück)

Ich versichere, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich und sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Hamburg, den 28. Oktober 2003

(Nils Michaelsen)