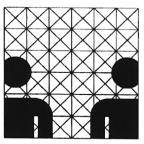
INTRUSION DETECTION SYSTEME IN FIREWALLS

2., überarbeitete Auflage

Baccalaureatsarbeit

Fachbereich Informatik

Universität Hamburg



Benjamin Hoherz

Silvio Krüger

Jan Menne

Nils Michaelsen

Betreuer: Prof. Dr. Klaus Brunnstein

Hamburg, im April 2002

VORWORT ZUR ZWEITEN AUFLAGE

Die Urfassung dieser Arbeit entstand innerhalb der zeitlichen Vorgaben der Prüfungsordnung für den Baccalaureus Scientiae – also binnen 6 Wochen. Wer schon einmal eine vermeidlich wissenschaftliche Arbeit in dieser kurzen Zeit geschrieben und mit drei weiteren koordiniert hat, kann vermutlich erahnen, dass die erste Auflage nicht nur mit kleinen Rechtschreibfehlern, sondern zum Teil auch mit inhaltlichen Irrtümern gespickt war.

Dank einiger Hinweise – besonders von Matthias Finck und Karin Pape – und viel Engagement vor allem von Nils ist diese zweite, überarbeitete Auflage entstanden.

Hamburg, im April 2002

VORWORT ZUR ERSTEN AUFLAGE

Wieder einmal ist eine Arbeit nicht voll und ganz ohne fremde Unterstützung entstanden. Unser besonderer Dank gilt Martin Becker für den ersten Überblick über dieses Thema, Dietmar Krüger und Karim Senoucci für die technische Unterstützung, Astrid Fethke und Mathias Meyer für die Hilfen bei den Recherchen und Andreas Hoherz, Bastian Koos, Tanja und Georg Menne sowie Lars Orta für das konstruktiv kritische Korrekturlesen.

Hamburg, im Juli 2001

INHALTSVERZEICHNIS

0 EINFÜHRUNG

Diese Baccalaureatsarbeit befasst sich mit dem Thema „Intrusion Detection Systeme in Firewalls“ als eine mögliche Maßnahmenkombination, um sich vor gezielten Angriffen aus Netzen wie dem Internet wegen der Mängel der heutigen Systeme nachträglich zu schützen.

In allen Organisationen gewinnen Informationen an Bedeutung. Offensichtlich wird dies im Banken- und Versicherungswesen, wo die Verwertung von Informationen über den Erfolg und Misserfolg des Unternehmens entscheidet (zum Beispiel beim Kauf oder Verkauf von Aktien). Aber auch die Industrie lebt von Informationen, denn die Schaltbilder von neuartigen Prozessoren, das Design eines Flugzeuges und der Quellcode einer Software gehören zu den höchsten Gütern eines Unternehmens. Die Werte, welche die Informationen darstellen, gilt es unter allen Umständen vor unberechtigtem Zugriff zu schützen.

Dennoch beläuft sich nach Angaben des TÜV der jährliche wirtschaftliche Schaden in Deutschland, der durch Hacker verursacht wird, auf über 10 Milliarden Euro ([FAZ 17.2.00]). Vor allem der Missbrauch, die Verfälschung oder der Verlust der Informationen kann für Unternehmen aller Art Existenz gefährdend sein. Vor allem kleine Unternehmen scheuen noch die Investitionen in Schutzmaßnahmen, während die Global Player bereits Sicherheitsmechanismen verwenden. Dennoch sind auch jetzt schon große Unternehmen Opfer massiver netzbasierter Angriffe. So waren die auf Online-Diensten spezialisierten Unternehmen, wie die Buchhandlung Amazon.com, das Auktionshaus eBay.com und das Portal Yahoo.com im Februar 2000 Opfer so genannter verteilter (distributed) Denial-of-Service-Attacken. Diese Attacken überforderten die Server der besagten Unternehmen, so dass ihre Dienste von nur sehr unzuverlässig bis gar nicht mehr von den eigentlichen Kunden dieser Firmen in Anspruch genommen werden konnten.

Eine große Gefahrenquelle stellen hierbei öffentliche Netzwerke dar. Um konkurrenzfähig zu bleiben, müssen Organisationen mittlerweile schon im Internet vertreten sein, denn sonst wenden sich vor allem die jüngeren Kunden an die Mitstreiter. Aber auch die heutigen Unternehmensstrukturen verlangen externe Zugänge zur IT-Infrastruktur. So will ein Mutterkonzern auch Zugriff auf die Daten der Töchterunternehmen haben, Mitarbeiter sollen auf Reisen vom Hotel aus sich in das Unternehmen einloggen können und der neuste Trend – E-Business – lässt sich nur über Netzwerke realisieren.

Diese Verbindungen zur Außenwelt öffnen aber auch Tür und Tor für die Computerkriminalität. Da durch mangelhafte Konzepte und deren Umsetzung in den heutigen Betriebssystemen und Anwendungsprogrammen es immer häufiger zu unberechtigten Zugriffen auf wertvolle

Informationen kommen kann, gibt es neben dem Kappen der Verbindungen zur Außenwelt zwei Lösungsansätze.

- Zum einen kann die Sicherheit der einzelnen Systeme (Betriebssysteme, Softwarekomponenten, Workstations, Kommunikationsprotokolle etc.) erhöht werden. Sinnvoll umgesetzt erfordert dieses Verfahren aber die völlige Neugestaltung aller Systeme – ein Aufwand, der bei den Herstellern liegt, dort aber gescheut wird.
- Zum anderen werden durch „Bastellösungen“ (K. Brunnstein) die heutigen Systeme erweitert. Firewalls und Intrusion Detection Systeme sind solche nachträglichen Erweiterungen der ursprünglich unsicheren Systeme.

An dem möglichst einzigen Zugangspunkt wird der Datenverkehr zum Beispiel mit Hilfe einer Firewall eingeschränkt. Dennoch wird es einem Einbrecher durch fehlerhafte Programme, Fehlkonfigurationen oder unbedachte Verhaltensweisen der Benutzer immer noch möglich sein, die Firewall zu überlisten und der Organisation Schaden zuzufügen. Doch dabei hinterlässt der Eindringling Spuren. Anhand dieser Spuren kann ein so genanntes Intrusion Detection System (kurz: IDS) solche Vorfälle erkennen und Alarm schlagen.

Die Wirkungsweise einer Firewall in Verbindung mit einem IDS soll in dieser Baccalaureatsarbeit untersucht werden. Kapitel 1 bietet einen geschichtlichen Abriss über das Aufkommen der Cracker, welche durch Angriffe gezielt Schaden anrichten wollen. In Kapitel 2 wird die Motivation für diese Arbeit, sowie technische Grundlagen geklärt. Danach wird in Kapitel 3 näher auf Firewalls und in Kapitel 4 auf Intrusion Detection Systeme eingegangen.

Für die konkrete Untersuchung der Wirkungsweise von Firewall und IDS standen uns im Labor des Arbeitsbereichs Anwendungen der Informatik in Geistes- und Naturwissenschaften (AGN) des Fachbereichs Informatik an der Universität Hamburg drei Rechner zur Verfügung. Des Weiteren bekamen wir die Firewall Gauntlet NT 5.5 und den IDS CyberCop Monitor freundlicherweise von Network Associates, Inc. zur Verfügung gestellt, wofür wir an dieser Stelle herzlich danken möchten. Kapitel 5 ist eine Beschreibung der durchgeführten Versuchsreihen:

- Angriff ohne Schutz,
- Angriff mit Firewall und
- Angriff mit Firewall in Verbindung mit einem Intrusion Detection System.

Kapitel 6 behandelt die Zusammenfassung und liefert unser Fazit zum Thema „Intrusion Detection Systems in Firewalls“.

In dem Projekt, über das diese Baccalaureatsarbeit berichtet, gab es keine feste Rollenverteilung. Die beteiligten Autoren haben bei allen Aufgaben mitgeholfen. Dennoch wurden vor allem auch für diesen Projektbericht Hauptverantwortliche benannt. Dabei wurden die Aufgaben nicht voneinander getrennt, sondern nur geklärt, wer für eine Aufgabe besonderes Fachwissen sich aneignen und dieses später auch in den entsprechenden Kapiteln dokumentieren soll. Folgende Einteilung wurde vorgenommen:

Jan Menne beschäftigt sich mit dem geschichtlichen Hintergrund der Angreiferkultur. Des Weiteren wählt er die Angriffsarten und die entsprechende Software und führt damit die Angriffe im Versuchsnetz durch. Dabei werden die Versuchsergebnisse ohne Schutz von ihm protokolliert und ausgewertet. Formal beschreibt er den im Internet verwendeten TCP/IP-Protokollstapel.

Benjamin Hoherz und Nils Michaelsen kümmern sich hierbei um den Bereich Firewalls. Benjamin Hoherz beschäftigt sich formal mit den Paketfiltern und den Einsatzgebieten von Firewalls. Innerhalb der Versuche protokolliert er den Versuch mit Firewall und wertet diesen anschließend aus. Formal beschäftigt sich Nils Michaelsen mit den Applikationsfiltern und Problematiken der Firewalls. Praktisch ist er für die Konfiguration der Firewall verantwortlich. Des Weiteren beschreibt er den formalen Hintergrund der Sicherheitspolitiken.

Silvio Krüger beschäftigt sich mit dem Intrusion Detection System. Sowohl der formale Anteil als auch der praktische Versuch werden von ihm durchgeführt. Zusätzlich beschreibt er den theoretischen Hintergrund des ISO OSI-Modells.

Hamburg, im Juli 2001

Benjamin Hoherz
Silvio Krüger
Jan Menne
Nils Michaelsen

1 GESCHICHTLICHE BETRACHTUNG

„Im Kriege sollte man seine Feinde gut kennen. Da hilft es, sich mit der Geschichte auseinander zu setzen“.

Die Kenntnisse über die Historie der Gegner könne deren Verhaltensmuster leichter erkennbar machen und somit können bessere Verteidigungsmaßnahmen entwickelt werden. Aus diesem Grund beschäftigt sich dieses Kapitel mit der Geschichte der Hacker, obwohl, wie wir sehen werden, nicht alle Hacker ein gemeinschaftliches Geschichtsbewusstsein entwickelt haben und mangels dessen auch keine traditionellen Vorgehensweisen kennen. Dennoch können aus der Geschichte die Motivation der verschiedenen Hackergenerationen erkannt werden.

Seitdem die Menschheit existiert, gibt es auch Geschichten von heldenhaften Taten bzw. arglistigen Verstößen. Die Sage der List des Odysseus in dem letzten trojanischen Krieg ist wohl eine der ersten Überlieferungen. Odysseus hatte nach der Überlieferung damals ein hölzernes Gottesopfer in Form eines Pferdes bauen lassen. Darin versteckte er griechische Kämpfer. Als die restlichen griechischen Soldaten in See stachen, holten die Trojaner das „Geschenk“ durch die sonst verschlossenen Tore in ihre Stadt. Nachts krochen die griechischen Kämpfer aus dem Pferd und öffneten die Stadttore für ihre bereits wartenden Kameraden. Troja wurde daraufhin im zehnten Jahre des Trojanischen Krieges zerstört (nach [Grill 92]).[1]

Nach dem Jargon File ([Raymond 01]) ist ein „Hacker unter anderem jemand, der Spaß daran hat, auf intellektuelle Herausforderungen einzugehen, vor allem wenn er dabei kreativ Grenzen über- oder umgeht“. Odysseus könnte deshalb mit seiner kreativen Art, die Schutzmauern Trojas zu überwinden als erster Hacker angesehen werden.

Im Allgemeinen haben Hacker keine negative Intention, wie zum Beispiel das Zerstören einer Stadt oder in der heutigen Zeit Computersabotage. Hacker versuchen ihrem Eigenverständnis nach nur Sicherheitslücken aufzuweisen und bekannt zu machen. Hacker distanzieren sich deshalb von den so genannten „Crackern“. Obwohl die Medien beide Gruppen häufig nicht differenzieren, sind die Cracker vorwiegend darauf erpicht, an alle Arten von Ressourcen zu gelangen oder deren Verfügbarkeit zu reduzieren. Die eingesetzten Techniken der Cracker und der Hacker sind dabei dieselben, weshalb eine Unterscheidung der beiden Gruppen ohne Berücksichtigung ihrer Intentionen schwer fällt. Es kommt allerdings vor, dass die Cracker eine technisch versierte Lösung der Hacker für ihre Zwecke übernehmen und einsetzen.

[1] Vor allem in letzter Zeit haben sich die Indizien jedoch verdichtet, dass es nie zu einem trojanischen Krieg kam, insofern ist diese Überlieferung anzuzweifeln.

Die Hacker-Gemeinschaften – entstanden an unterschiedlichen Orten aus verschiedenen technischen Entwicklungen – haben sich mittlerweile zu einer eigenen Kultur entwickelt. Diese soll im Folgenden etwas genauer beleuchtet werden, um ein besseres Verständnis über die Notwendigkeit des Schutzes durch Firewalls und Intrusion Detection Systeme zu gewinnen.

Hackerangriffe setzten am Anfang ihrer Geschichte immer tiefe Kenntnisse in den jeweiligen technischen Gebieten voraus. Eine der ersten elektrotechnischen Entwicklungen, die bekanntermaßen nicht nur in ihrem eigentlichen Sinne genutzt wurde, war die Telephonie. Am Anfang waren es vorwiegend männliche Jungendliche, die den Vermittlungsdienst in den Telefonschaltzentralen gewährleisteten. Viele machten sich einen Spaß daraus, falsche Telefonpartner zu vermitteln, Gespräche zu belauschen, diese frech zu kommentieren und das Telefonnetz zu testen, indem sie eine Telefonleitung quer über den Kontinent und wieder zurück schalten ließen, um mit dem Tischnachbarn zu reden. Langeweile und Neugier waren die treibenden Kräfte dieser Telephoniemissbräuche. Nach Bruce Sterlings „Hacker Crackdown“ ([Sterling 93]) war das Benehmen der Jungendlichen der Grund für AT&T nur noch Frauen für den Vermittlungsdienst einzustellen.

1.1 HACKER-BEWEGUNGEN

Die erste Generation der Computerhacker war aufgrund der damaligen Gegebenheiten an den Computerentwicklungsstätten zu finden. Da in den Anfängen der Computerentwicklung Rechner wegen der hohen Produktionskosten auch sehr hohe Verkaufspreise erzielten, gab es sie häufig nur an öffentlichen oder privaten Forschungseinrichtungen. So waren die ersten Hacker an einer der renommiertesten Universitäten der USA zu finden. Das Signal und Energie Subkomitee (Signal and Power – S&P) des Tech Model Railroad Clubs (TMRC) am Massachusetts Institute of Technology (MIT) nutzte als erstes den Begriff „Hack“ für eine im technischen Sinne besonders elegante Lösung eines Problems.

Vor allem aus dieser Gruppe von Studenten entsprangen die ersten Computerhacker, als 1961 das MIT im Gebäude 26 des Campus einen „Programmed Data Processor“ der Firma Digital Equipment Corporation (DEC) PDP-1 erhielt. Es war ebenfalls dieses Subkomitee, das später den Kern des Labors für künstliche Intelligenz am MIT bildete und somit Computerzugang erhielt.

1.1.1 HACKEN DER TELEFONNETZE

Zu den ersten Hacks des TMRC S&P zählte die Programmierung einer Schnittstelle zum Telekommunikationsnetz. Diese missbrauchten die Studenten, um unerlaubten Zugriff auf die Vermittlungsstellen zu bekommen. Zum einen wurden darüber kostenlose Telefonate für den einzelnen vermittelt, zum anderen wurden aber auch nur „spaßeshalber“ möglichst wirre interkontinentale Verbindungen geschaltet.

Das Hacken von Telefonnetzen hatte seinen Höhepunkt im Vietnamkrieg erreicht, als die US-Regierung zur Finanzierung des Krieges die Telefonsteuer auf 10% erhöhte, Daraufhin wurde das Hacken der Telefonnetze in der Gesellschaft als ziviler Ungehorsam angesehen. Nach Boris Gröndahl's „Hacker“ ([Gröndahl 00]) werden heutzutage Telefonhacks von entweder fast mittellosen „Phone Phreaks“ oder von der organisierten Kriminalität durchgeführt.

1.1.2 COMPUTERSPIELE

Neben dem Programmieren einer Schnittstelle zum damaligen Telefonnetz missbrauchten die studentischen Mitglieder der KI-Forschung am MIT die Rechner zur Programmierung von Spielen. Die PDPs sollten im Gegensatz zur Nutzung heutiger Personal Computers (PCs) der wissenschaftlichen Problemlösung dienen und nicht dem reinen Vergnügen der Anwender. Dennoch war in der Programmierung eines Spiels der erste Achtungserfolg der KI begründet, als ein Schachprogramm von Richard Greenblatt auf einem PDP-6 den Sozialwissenschaftler Herbert Dreyfus schachmatt setzte.

1.1.3 DAS ARPANET

Am Anfang der Computerentwicklung standen große und teure Mainframe-Rechner. Diese wurden im Timesharing-Verfahren von mehreren Anwendern seriell genutzt. Programmieren war noch stark mit mathematischen und physikalischen Kenntnissen verbunden, und der Markt wurde von dem Hersteller DEC dominiert. Selbst die Knoten des ARPANET (Vorläufer des heutigen Internets) waren fast alle von DEC. Es handelte sich hierbei meist um die PDP-10. Auch das MIT hatte PDP-10s entwickelte aber ihr Betriebssystem „Incompatible Timesharing System“ (ITS) selbst, aus dem dann so bekannte Software-Produkte wie der Emacs-Editor hervorgingen. Alles selber oder zumindest nicht von einer Firma gelenkt zu entwickeln, war eine typische Eigenschaft der damaligen Hacker. Aufgrund dieser Vorgehensweise wurde wesentlich zu der Vielfalt des Internets beigetragen, zumal viele der von Hackern entwickelten Lösungen nicht angeboten wurden.

Das ARPANET wurde vom US-Verteidigungsministerium (Department of Defense – DoD) aufgebaut und bot Universitäten und Studenten einen Netzzugang. Eigentlich sollte dieser Zugang den Wissenschaftlern als Kommunikationsbasis dienen, doch die Studenten entdeckten das Netz für sich. Sicherlich war das DoD nicht davon ausgegangen, dass zum Beispiel Verteilerlisten nicht hauptsächlich zum wissenschaftlichen, sondern zum freizeitlichen Nachrichtenaustausch genutzt wurden. Bekannt wurde vor allem die Liste „SF-Lovers“, über die jahrelang Neuigkeiten der Science Fiction verbreitet wurden.

Dieses Datennetz bot aber nicht alle Möglichkeiten, die Hacker sich gewünscht hatten. Da auch Standardisierungsinstitutionen per se von Hackern nicht besonders gemocht wurden, entwickelten sie verschiedene Protokolle und stellten diese in „Request for Comments“ (RFC, s. [RFC]) zur Diskussion. Auch heute werden viele der im Internet verwendeten Protokolle durch RFCs festgelegt. Diese Protokolle nutzen die zugrunde liegende Technik in einer Art und Weise aus, die das DoD nicht vorgesehen hatte. Somit handelt es sich bei den Protokollen im eigentlichen Sinne um Hacks.

1.1.4 HARDWARE-HACKER

Die bisher betrachtete Hacker-Geschichte spielte sich hauptsächlich an der Ostküste der USA ab. An der kalifornischen Westküste entstand eine andere Art von Hackern – die „Hardware-Hacker“ (Boris Gröndahl). Sie beabsichtigten aus den bisherigen Mainframes kleinere Personal Computer zu entwickeln, die jeder nutzen könnte.

Erfüllt wurde diese Forderung durch den Altair 8800 von Mits, einer vor der Insolvenz stehenden Firma, die verschiedene Bausätze für Elektrotechnik-Fans herstellte. Die Ein- und Ausgabe erfolgte durch Schalter und Dioden, basierte also nicht auf Lochstreifen oder ähnlichem.

Die Fans des Altair vereinten sich im Homebrew Computer Club, der sehr bald enge Beziehungen zur Stanford University pflegte und auch dort tagte. Der Gründer des Homebrew Computer Clubs, Fred Moore, sagte: „Hier war der Ursprung der PC-Industrie. Sie entstand nicht bei Texas Instrument, IBM oder Fairchild. Sie entstand unter Leuten am Rand einer alternativen Vision“ (aus Freiberger und Swaine: Fire in the Valley, New York 1984, abgedruckt in [Gröndahl 00]).

Mit der Zeit entwickelten sich die Altair-Begeisterten zu Firmengründern. Der Homebrew Computer Club, der zuerst von den großen Firmen den Zugang für jeden zu Computern forderte, löste sich allmählich aufgrund der vielen aus ihm hervorgegangenen Firmengründungen

auf. Die wohl bekanntesten Firmengründer, die im Homebrew Computer Club zueinander fanden, waren Steve Jobs und Stephen Wozniak, die die Firma Apple Computer aufbauten.

Auch wenn nur wenige Hacker es wahr haben wollen, ging Microsoft aus dieser Entwicklung hervor. Bill Gates und Paul Allen programmierten zuerst Softwarepakete auf den Altairs. Im Gegensatz zu Apple machte sich Microsoft aber bald nach der Firmengründung bei den Hackern unbeliebt. In dem „Offenen Brief an Computerhobbyisten“ von Bill Gates aus dem Jahre 1976 warf er den damaligen Computerfans vor, sie würden sich die Software zusammenstehlen – ein Vorwurf, der auch heute noch häufig vom Microsoft-Konzern zu hören ist. Mit den zahlreichen Firmengründungen und dem nahenden Ende des Homebrew Computer Clubs endete auch diese Entwicklung der Hacker-Szene.

1.1.5 UNIX

In den Bell Labs entstand noch eine weitere Richtung der Hacker. Ken Thomson arbeitete bei Bell an der Programmierung des Betriebssystems Multics. Es basierte auf ITS und dem Grundgedanken, dass die Komplexität des Systems für den Endanwender nicht offensichtlich sein muss. Multics wurde 1969 von den Bell Labs aufgegeben, später aber von Honeywell vertrieben. Zwischenzeitlich vermisste Thomson seine Multics-Umgebung und begann privat ein ähnliches Betriebssystem zu entwickeln. Er nannte es Unix. Zusammen mit dem Erfinder der Programmiersprache C, Dennis Ritchie, erkannte er, dass die Compiler für C mittlerweile so gut waren, dass Unix in C geschrieben werden konnte. Vorherige Betriebssysteme waren immer in Assembler geschrieben worden und konnten daher auch nur auf baugleichen Computern eingesetzt werden. Mit dem auf C basierenden Unix konnte nun erstmals ein Betriebssystem Hardware übergreifend hergestellt werden.

Über Richard Stallmanns „GNU“ (für „GNU’s not Unix“) und Linus Torvalds’ „Linux“ wurde mit „Unix“ auch der Open-Source-Gedanke – also die länger in bestimmten Kreisen bekannte Offenlegung des Quellcodes – propagiert. Neben den Betriebssystemen entstanden so Tausende von Softwarepaketen, die über die Netze frei verfügbar waren.

1.1.6 ZWEI HACKER-GRUPPEN VEREINIGEN SICH

1983 hatte DEC beschlossen, keine weiteren Nachfolger der PDP-Reihe zu bauen. Die DEC wollte sich nur noch auf VAX konzentrieren, auf denen UNIX und nicht mehr das von PDP-10-Fans bevorzugte ITS lief. Da ITS in Assembler programmiert war und eine Übernahme von ITS auf die VAX schwer machbar erschien, wechselten die meisten Hacker auf Unix.

Mit der Entwicklung und Ausbreitung des Internets konzentrierten sich Masse der Hacker auf Weiterentwicklungen von und für Linux. Natürlich gibt es auch Hacker, die sich auf andere Gebiete (zum Beispiel Datenübertragung, Hardware, BIOS etc.) spezialisiert haben, zumal es dem vorurteilbehafteten Naturel eines Hacker entspricht: „Nicht mit dem Strom schwimmen!“

1.2 CRACKER-SUBKULTUR

Wie anfangs schon erwähnt wurde, unterscheiden sich die Cracker von den Hackern und verdienen eine gesonderte Betrachtung.

1.2.1 DER CCC

In der bisherigen Geschichte der Hacker spielte Deutschland, wie ganz Europa, fast keine Rolle. Weltweite Berühmtheit erlangten deutsche Hacker, eigentlich Cracker, mit der Datenspionage für den sowjetischen Geheimdienst KGB.

Der Grund für die anfänglich geringe Zahl deutscher Hacker lag darin, dass ihnen der Zugriff auf die Datennetze erschwert wurde, da diese zunächst nur für die Wirtschaft vorgesehen waren. Deshalb wurde der Chaos Computer Club (CCC) als Vereinigung von Hackern zuerst durch seine Antipartie zur Deutschen Bundespost (DBP) bekannt. Der CCC erklärte in den 80ern in seiner Zeitschrift, wie Modems gebaut und an das Telefonnetz angeschlossen werden konnten. So war auch die DBP erstes und hauptsächliches Opfer. Vor allem der Bildschirmtext (BTX) wurde häufig gehackt. Anfänglich wurden Trickfilmdarstellungen auf BTX-Seiten erzeugt, dann wurde nachgewiesen, dass abgesendete BTX-Mitteilungen nachträglich geändert werden können. Bekannt wurde der CCC auch durch die Erstellung kostenpflichtiger BTX-Seiten, deren Aufrufe zu Lasten der Hamburger Sparkasse in Höhe von DM 135.000 gingen.

Nach der Öffnung der Datennetze trat das Ausnutzen von Sicherheitslücken des BTX der DBP in den Hintergrund, da nun ein größeres und interessantes Netz zur Verfügung stand. Aufgrund der vielen bekannten Sicherheitslücken und sicherheitsrelevanten Standardeinstellungen, wie zum Beispiel Benutzer „System“ mit Passwort „Manager“, wurden die VAX von DEC bevorzugtes Angriffsziel, um dann darüber weitere Netze zu „erkunden“. Wie das MIT bei Hackern war das Forschungszentrum CERN in Genf erste Anlaufstelle für die Cracker Europas. Vor allem die zahlreich vorhandenen VAX boten Crackern, wie Hackern, die Möglichkeit, sich unrechtmäßig in fremden Netzen auszubreiten. Diese Netze waren teilweise so sicherheitskritisch, dass sich die Cracker aus Angst vor Verfolgung an den CCC wandten.

Am 15. September 1987 veröffentlichte der CCC eine Liste mit 138 gehackten Computer / -netzen. Darunter war nicht nur das CERN, sondern auch die NASA und ESA, sowie viele Universitäten.

Im gleichen Zeitraum, als sich die Medien mit der Liste beschäftigen, wurde ein weiterer, viel relevanterer Hack durchgeführt. Erst aus spielerischer Neugier, dann im Auftrag des KGB gelangte ein Quintett von Hackern an sensitive Daten. Sie hatten sich vorwiegend über den CCC kennen gelernt und waren hauptsächlich in Hannover vertreten. Auf Wunsch des KGB's sollten sie vor allem amerikanische Einrichtungen des Militärs oder damit kooperierende Wissenschaftslabors ausspionieren. Einige ihrer Versuche liefen über die Computer des Lawrence Berkeley Laboratory in Kalifornien, das von Clifford Stoll administriert wurde. Seiner Genauigkeit ist es zu verdanken, dass die Spionageversuche schließlich aufgedeckt wurden ([Stoll 99]).

In den 90ern hat sich der CCC zu einer Interessensvertretung der echten Hacker und zur Anlaufstelle für Cracker, die ihre Aktivitäten bereuen und publik machen wollen, entwickelt. Der CCC ist dafür gesellschaftlich anerkannt und wird von der Informationsgesellschaft und der Industrie gerne um Rat gebeten. Trotzdem hält der Sprecher des CCC, Andy Müller-Maguhn, Cracker immer noch für „Befreier von Bits".

1.2.2 SOFTWAREPIRATERIE UND MALWARE

Den größten wirtschaftlichen Schaden verursachen zurzeit wohl die Malware-Programmierer und die Cracker von Software. Sie werden zusammen mit den Hackern von Netzen als eigentliche Hacker in der breiten Masse und den Medien angesehen.

Die Cracker von Software wollen vor allem die „lästigen" Kosten bei der Anschaffung von Software sparen. Dazu programmieren sie Key-Generators für Software, die über eine Registrierung frei geschaltet werden muss, „Zeitbeschränkungsverlängerer" für Shareware, die zum Beispiel 30 Tage lang getestet werden kann und sich sonst nach Ablauf der Zeit eigenständig sperren würde, und Werkzeuge zur Umgehung eines Kopierschutzes. Alle diese so genannten „Toolz" dienen der eigenen Bereicherung, denn durch diese Programme werden die teuer entwickelten Softwarepakete kostenlos erschlichen. Der dadurch entstehende Schaden für die Softwarebranche wird auf mehrerer Milliarden Mark pro Jahr in Deutschland geschätzt.

Aber nicht nur die Cracker von Software sorgen für wirtschaftlichen Schaden, auch die Malware-Programmierer verursachen solchen. Ihre Motivation reicht von Hass auf den (ehemaligen) Arbeitgeber bis zu reiner Experimentierfreudigkeit. Ihre Ergebnisse sind Viren, Würmer, Trojaner und ähnliche zu dem Begriff „Malware" zusammengefasste Produkte. Diese Art der

Cracker vereinigt sich häufiger zu international operierenden Gruppen. Sie tragen exotische Namen wie „Cult of the dead Cow“, „Doomsday“ und „Underground Empires“.

1.2.3 CRACKER HEUTE

In den letzten Jahren ist eine starke Entwicklung der Cracker-Szenen in Russland zu entdecken. Das kann auf mehrere Faktoren zurückgeführt werden: Russland leidet zurzeit ökonomisch immer noch an dem Zusammenfall der Sowjet Union. Die Armut in der Bevölkerung ist groß und Softwarepakete im Verhältnis zu den durchschnittlichen Einkommen zu teuer. Des Weiteren fehlen in Russland entsprechende Computerkriminalitätsgesetze, bzw. es mangelt an deren Umsetzung.

Trotz des Schadens, den Cracker verursachen, sind sie auch in der Wirtschaft heutzutage sehr gefragt (als Beispiele seien die Einstellungen von diversen Virenprogrammierern genannt), da sich die Cracker sehr gut in der Informationstechnologien auskennen. Die Unternehmen hoffen, dass Cracker aufgrund entsprechender Gehälter nicht kriminell, sondern Profit bringend arbeiten. Des Weiteren haben sich auch schon Agenturen wie hacker-for-hire.com und rent-a-hacker.com gebildet, die über das Internet Hackerdienstleistungen anbieten.

Im Großen und Ganzen ist allerdings festzustellen, dass immer mehr wenig ausgebildete Cracker vor allem bei netzbasierten Angriffen auf sich aufmerksam machen. Aufgrund des Internets werden neue Angriffstechniken, kürzlich aufgedeckte Sicherheitslücken oder sonstige Fehler (sog. Exploits) sehr schnell bekannt. Die Hersteller der Systeme reagieren auch meistens relativ schnell und stellen ihren Kunden Patches, Upgrades, Service Releases oder ähnliches zur Verfügung. Seit den Anfängen von E-Business sind die IT-Infrastrukturen von Kunden und Lieferanten aber so eng miteinander verknüpft, dass die Patches etc. erst nach einer Testphase eingeführt werden. Während dieser Testphase sind die betroffenen Organisationen verwundbar und können leicht Opfer der neuen Angriffstechnik werden. Es kommt auch vor, dass ein Administrator von dem neuen Patch oder der Sicherheitslücke nichts erfährt und deshalb sein System nicht entsprechend schützt.

Der leichte Zugang zum Internet, das immer noch den Ruf eines rechtsfreien Raumes hat, erleichtert es, Angreifern netzbasierte Hacks durchzuführen. Ein Angreifer muss kein tieferes Verständnis für die Technik haben, denn zum einen werden auf den einschlägigen Webseiten die Sicherheitslücken im Detail erklärt, und zum anderen werden auch gleich fertige Programme, die die Exploits ausnutzen können, zur Verfügung gestellt. Für Malware-Hersteller, die sich dieser Programme aus dem Internet bedienen, gibt es mittlerweile den Spitznamen „Script-Kiddies“, da diese meist sehr jungen Programmierer vorwiegend Script-Würmer herstellen.

2 GRUNDLAGEN DER NETZSICHERHEIT

Dieses Kapitel befasst sich mit der Motivation für das Projekt „Intrusion Detection Systeme in Firewalls“ und erläutert einige der Grundlagen der Netzsicherheit, sowie der technischen Grundlage, wie den verwendeten Protokollen.

2.1 SCHUTZBEDARF

Der nun folgende Abschnitt soll darstellen, wozu Schutzmaßnahmen erforderlich sind. Er geht darauf ein, warum spezieller Schutz gebraucht wird und wie groß die Gefahr bei der Anbindung an öffentliche Netze, wie das Internet, ist.

2.1.1 RISIKEN DER IT-SYSTEME

Das Internet, ein auf dem Kommunikationsprotokollstapel TCP/IP basierendes globales Netz aus Netzen, erfreut sich seit seiner Entstehung immer größerer Beliebtheit. In den 80er Jahren des 20. Jahrhunderts schlossen sich immer mehr Unternehmen, Behörden und Lehranstalten an das Internet an. Mit den 90er Jahren begann die Anzahl der privaten Nutzer drastisch zu steigen. So explodierte die Zahl der Teilnehmer schlagartig. Mit ihnen nahm auch die Zahl derer mit böswilligen Absichten, die aus Kapitel 1 bekannten Cracker, zu. Diese haben ein leichtes Spiel. Die Gründe dafür sind vielschichtig und werden durch die Schwächen des TCP/IP-Stacks mit dem Internet Protokoll Version 4 (IPv4) begünstigt.

Neben den Sicherheitslücken in den Protokollen führen Mängel in den Applikationen zu erheblichen Risiken. Diese Mängel ergeben sich z.B. aus Fehlimplementationen, die Buffer Overflows oder Denial-of-Service ermöglichen. Grund für das Vorkommnis dieser Fehler ist die hohe und damit nicht überschaubare Komplexität der Software und Netze.

Des Weiteren ist das mangelnde technische Verständnis der Benutzer ein nicht zu unterschätzendes Sicherheitsproblem, zumal die Software-Ergonomie bei vielen Produkten noch in den Kinderschuhen steckt und die Programme somit technisches Verständnis voraussetzen.

2.1.2 NETZBASIERTE ANGRIFFE

Dieser Abschnitt soll einen kurzen Überblick über die möglichen Angriffe geben, die in einem Netzwerk möglich sind. Die Angriffsarten lassen sich nach [Chapman 00] in folgende Kategorien unterteilen:

- Einbruch

  Ein Einbruch in ein System lässt den Angreifer das System ganz oder teilweise unter Kontrolle bringen. Möglich ist dies über die Angriffstechniken Hijacking, Buffer Overflow und Spoofing.

  Bei einem Hijack schaltet sich ein Angreifer zwischen zwei Kommunikationspartnern und bringt die Verbindung unter seine Kontrolle. So kann er die Identität eines Kommunikationspartners ausnutzen.

  Um einen Buffer Overflow auszunutzen, bekommt eine bestimmte Applikation mehr Daten, als es in einem Puffer zwischenspeichern kann. Enthalten diese Daten ausführbaren Code, so kann somit ein fremdes Programm auf dem angegriffenen Rechner ausgeführt werden.

  Für einen Spoofing-Angriff gibt sich der Angreifer als jemand anders aus, indem er die Absenderadresse fälscht. Um diese Angriffsart für einen Einbruch zu nutzen, muss der wirkliche Besitzer der Absenderadresse isoliert werden, damit die Antworten empfangen werden können.

  Ein Einbruch kann aber auch über reguläre Zugänge erfolgen, die durch Accounts mit Passwörtern gesichert sind. Hierbei werden Accountname und Passwort erraten oder zum Beispiel über Trojaner ausspioniert, um so unberechtigt Zugriff auf das System mit seinen Daten zu erlangen.

- Informationsdiebstahl

  Hierbei unterscheidet man zwischen aktiven und passiven Informationsdiebstahl. Im aktiven Fall erfolgt der Diebstahl mit Hilfe eines Einbruchs, um so Informationen nach außen zu schaffen.

  Der passive Diebstahl erfolgt durch das Abhören von Datenpaketen. Diese Technik wird als Sniffing bezeichnet.

- Denial-of-Service

  Ein Rechner bekommt Anfragen, die aus zu großen Mengen oder missgebildeten Datenpaketen bestehen, die zur Einstellung der Leistung eines Dienstes oder des ganzen Systems führen.

  Ein Beispiel zur Durchführung einer Denial-of-Service-Attacke ist die Technik des Floodings. Hierbei wird ein Rechner mit Datenpakten überschwemmt. Auch diese Angriffstechnik kann mit Spoofing kombiniert werden. Dies wird bei Smurfattacken eingesetzt, wobei ICMP-Echo-Request-Pakete mit einer gefälschten Absenderadresse an alle Hosts in einem Netzwerksegment geschickt werden. Der reale Host der Adresse wird dann mit Antworten überlastet.

2.1.3 WAS ES ZU SCHÜTZEN GILT

Die Angriffe versuchen vor allem die Schädigung der Assets. Die Assets sind Vermögenswerte und die damit verbundenen Eigenschaften einer Person oder Organisation. Im Hinblick auf die IT-Sicherheit sind dies die Daten, die Ressourcen und auch der Ruf einer Person oder Organisation, die es zu Schützen gilt.

Im Hinblick auf die Kompromittierung von Daten heißt dies, die Wahrung der

- Integrität
  Die Korrektheit der Daten muss gewährleistet werden, die Daten müssen daher gegen Manipulation, Modifikation und Zerstörung geschützt werden.

- Vertraulichkeit (inkl. Geheimhaltung)
  Vertraulichkeit bedeutet, dass Daten vor unberechtigtem Einblick geschützt werden müssen. Dazu ist es nötig, sowohl die betreffenden Systeme als auch die Übertragungswege zu schützen. In der Literatur wir dies auch unter dem Begriff der Geheimhaltung erwähnt.

- Verfügbarkeit
  Die Daten müssen jederzeit abrufbar sein.

Die Begriffe Integrität, Vertraulichkeit und Verfügbarkeit werden als die Sicherheitsmerkmale der Daten aufgefasst, denen eine vertrauenswürdige Verarbeitung genügen soll.

Nicht nur die Daten, sondern auch das System an sich mit seinen Ressourcen gilt es zu schützen. Die Systemressourcen werden von manchen Autoren als Teil der Verfügbarkeit der Daten aufgefasst. Wenn ein Angreifer die Maschine außer Gefecht setzt, so sind auch die Daten nicht mehr verfügbar. Die generelle Verfügbarkeit ist eine Eigenschaft des Systems, die gesichert werden soll.

Auch die Ressourcen selbst gilt es bei einem System zu sichern. Sie beinhalten u. a. Prozesszeit und Speicherkapazität. Manche Angriffe verfolgen das Ziel, die Rechenleistung oder den Speicherplatz und -inhalt einer fremden Maschine für ihre eigenen Zwecke widerrechtlich zu nutzen. Wenn dies gelingt, sind die Ressourcen für den Besitzer der Maschine nicht verfügbar. Er möchte jedoch jederzeit die vollen Kapazitäten seiner Maschine nutzen können. Daher werden die Ressourcen hier separat aufgeführt.

Der letzte hier besprochene Schaden, den ein Angreifer anrichten kann, ist die Schädigung des Rufes einer Person oder Organisation. So kann zum Beispiel ein Professor eine von seiner

Adresse stammende E-Mail mit rassistischen Äußerungen seinen Arbeitsplatz und den Ruf kosten, wenn er die Fälschung nicht widerlegen kann. Wenn diese E-Mail nun von irgendwo anders her abgeschickt wurde, ist die Fälschung leichter zu beweisen, da Audit-Systeme den Weg der Mail offen legen können. Hat ein Angreifer die Nachricht von dem eingenommenen Rechner des Professors abgeschickt, so ist der Nachweis schwerer, denn die E-Mail ist einen zu erwartenden Weg vom PC des Professor aus gegangen, wurde aber über andere Datenpakete vom Hacker auf dem angegriffenen Computer erzeugt. Der Betroffene wird in diesem Fall Schwierigkeiten haben zu beweisen, dass die E-Mail doch nicht von ihm stammt (jur. Nicht-Abstreitbarkeit). Eine langfristige Rufschädigung ist dann nicht auszuschließen. Schutz gegen dieses Szenario bieten aber zum Beispiel digitale Unterschriften.

2.1.4 AUSMAß DER GEFAHR

Die Einnahme anderer Rechner kann auch den Ruf ganzer Organisationen schädigen. Von einem unter seiner Kontrolle stehenden Rechner kann ein Angreifer andere Rechner angreifen. So geschehen ist es zum Beispiel Anfang 2000, als die Webpages bekannter Unternehmen wie Yahoo durch Distributed Denial-of-Service-Attacken nicht mehr erreichbar waren. Dort wurde eine Vielzahl anderer Rechner eingenommen, die auf Kommando einen Angriff gegen diese Homepages durchführten. So ist es auch vorstellbar, dass ein Angreifer von einem Rechner eines Unternehmens aus ein Konkurrenzunternehmen angreift, um so die Unternehmen gegeneinander aufzubringen.

Nun gibt es immer noch einige Nutzer, die behaupten, dies treffe auf sie nicht zu. Ihnen ist zwar bewusst, dass es Gefahren gibt, halten aber das Risiko für zu gering. Dies begründen die Nutzer damit, dass ihr Bekanntheitsgrad nicht ausreiche, um von den Gefahren betroffen zu sein. Dies ist eine besonders lässige Interpretation des Prinzips der *Security through Obscurity*; man versteckt oder verschleiert etwas, um es zu sichern. Die Existenz des zu sichernden Objektes ist hierbei anderen nicht bekannt. So hoffen die Nutzer, ihre Computer seien in der Masse der ans Internet angeschlossenen Maschinen verborgen, solange niemandem etwas über die Existenz des Rechners berichtet.

Dieses Prinzip ist bei Weitem nicht ausreichend. Angreifer suchen immer neue Opfer und tasten ganze Provider nach ungesicherten Maschinen ab. Dies tun sie erst recht, wenn es sich um so genannte Scorekeepers handelt, die mit der Masse der von Ihnen geschädigten Rechner prahlen [Chapman 00]. Dass ein Schutzbedarf besteht, wird auch von behördlichen Organisationen bestätigt.

Ein Blick in die „Polizeiliche Kriminalstatistik 2000“ ([BMI 01]) verrät, dass seit 1994 mit Ausnahme von 1999 eine Steigerung der erfassten Computerkriminalitätsfälle von jährlich mindestens 15% zu verzeichnen war, während im gleichem Zeitraum die Kriminalitätsrate insgesamt um etwa 7% fiel. Hierbei ist allerdings zu bemerken, dass 80% der Computerkriminalitätsfälle Kreditkartenbetrug und weitere 4% Software-Piraterie waren. Viele der restlichen 10.117 Kriminalitätsfälle des Jahres 2000 (Computerbetrug, Fälschung von Daten, Datenveränderung, Computersabotage, sowie Ausspähen von Daten und Betrug mittels Zugangsberechtigungen zu Kommunikationsdiensten) hätten wahrscheinlich mittels Firewalls und Intrusion Detection Systeme erkannt oder sogar verhindert werden können. Diese Sicherheitsmechanismen haben aber zumindest aufgrund der integrierten Auditing Systeme dafür gesorgt, dass Angriffe wenigstens nachgewiesen werden konnten. Ob dies zu einer erfolgreichen Strafverfolgung geführt hat, ist aber aus der Polizeilichen Kriminalstatistik 2000 nicht ersichtlich und wegen der aktuellen Rechtslage und den üblicherweise verwendeten Protokollen auch eher unwahrscheinlich, denn ein lückenloser Nachweis ist sehr schwierig.

Namhafte Unternehmen scheuen vermutlich auch das Erstatten von Strafanzeigen wegen Computerkriminalität, weil sie befürchten müssen, dass geglückte Hackerangriffe ihrem Image und der Aktie mehr Schaden zufügen, als der Cracker verursacht haben könnte.

Der geschätzte Schaden durch Hacker und Cracker geht in die Millionen DM. So berichtete Lutz Reichert, Director Consultant der META Group Deutschland GmbH am 19. Juni 2001 auf dem Innovation-Meeting „IT-Security & Services im Web“ der Siemens Business Services GmbH & Co. OHG in Kiel von dem Ergebnis einer Umfrage. Danach komme durchschnittlich auf jedes Unternehmen pro Jahr ein IT-Schaden in Höhe von 500.000,- US-$ zu. Davon entfallen 33% auf Datendiebstahl, 33% Fälschung, 17% auf Beschädigung, 11% auf allg. Diebstahl und 6% auf Fremdzugriffe.

Die Politik hat in den letzten Jahren die Problematik „Computerkriminalität“ verstärkt in den Augenschein genommen. Das Ergebnis ist die Novellierung der Gesetze nicht nur auf Bundes-, sondern auch auf EU-Ebene. Diese neuen Gesetze schützen vor Hackerangriffen aber nur, wenn der Angreifer sich vor den Rechtsfolgen einschüchtern lässt. Geschieht dies nicht, müssen sich die Netzbetreiber und Computer-Benutzer selber schützen. Zwei der möglichen Schutzmaßnahmen werden in den Kapiteln 3 und 4 vorgestellt und in Kapitel 5 getestet.

In den beiden folgenden Abschnitten wird die Aufstellung eines Sicherheitskonzeptes behandelt, das den nötigen Schutz sicherstellen soll.

2.2 SICHERHEITSKONZEPT

Wie eben betrachtet, ist es zwingend erforderlich, sich zu schützen. Nun stellt sich die Frage, was und vor allem gegen was man sich schützen will. Dazu muss ein Plan entworfen werden, um den Schutz zu erreichen. Dies führt zu einem Sicherheitskonzept, das die Sicherheit gewährleisten soll.

2.2.1 BESTANDTEILE DES SICHERHEITSKONZEPTES

Bei einem Konzept handelt es sich nach Duden um den Entwurf eines Werkes. Das Werk ist im Kontext IT-Sicherheit ein IT-System, in das man genügend vertrauen hat, dass es Integrität, Verfügbarkeit und Vertraulichkeit durchsetzt. Zu dem Konzept gehört eine Politik, die Richtlinien für einen sicheren Betrieb festlegt, und eine Umsetzung der Politik. Beide Punkte werden hier näher erläutert.

Des Weiteren gehört zu einem Konzept ein Verfahren, wie auf den Fall des Versagens des Konzeptes zu reagieren ist. Ein Solcher Plan wird im Allgemeinen als Incident Response Plan bezeichnet. Ein solcher Plan ist besonders bei der Planung der Response in Verbindung mit den in Kapitel 4 beschriebenen IDS und IRS sinnvoll. Auf die Details dieses Plans wird in dieser Arbeit aber nicht näher eingegangen.

Häufig werden die Begriffe Politik und Konzept durcheinander gebracht. Bei einem Konzept handelt es sich nicht nur um eine Zielsetzung, die in der Politik festgehalten ist, sondern beinhaltet auch die Umsetzung und Überprüfung. Deutlich wird der Umfang des Konzeptes in [RFC2196], wo die Durchsetzung als Plan auf der Abstraktionsebene über der Politik dargestellt wird.

2.2.2 DER PROZESS DER SICHERHEIT

Die Entwicklung eines Sicherheitskonzeptes ist kein einmaliger Vorgang. Es ist ein sich ständig wiederholender Prozess, damit das Konzept den sich ständig ändernden Anforderungen weiter entspricht. Dabei muss das Konzept hinsichtlich der sich ständig ändernden Einsatzumgebung als auch der sich täglich wachsenden Zahl der Gefahren angepasst werden. Sowohl die Umgebung, auf die das Konzept abzielt, als auch die Gefahren ändern sich ständig. Das Konzept soll die Risiken der Gefahren minimieren und so einen ausreichenden Schutz gewähren.

Einen Leitfaden dazu gibt die ISO/IEC 1333. Er wurde in [Brunnstein 01a] vorgestellt und umfasst folgende Punkte:

1) Zielbestimmung
   Bestimmung der unternehmerischen Ziele, welchen Schutz es zu erreichen gilt.

2) Anforderungspolitik
   Festlegung der Informationen und Werte, die es zu Schützen gilt.

3) Bedrohungsanalyse
   Untersuchung der Gefahren, denen die Unternehmenswerte ausgesetzt sind.

4) Risikoanalyse
   Abschätzung des Risikos, dem das Unternehmen ausgesetzt ist, falls keine Sicherheitsmassnahmen erfolgen.

5) Risikovorsorgeplanung
   Festlegung der Maßnahmen zur Minimierung des Risikos.

6) Durchsetzung
   Anwendung der Maßnahmen im gesamten Kontext der Politik. Die Durchsetzung erfordert in der IT auch eine Umsetzung auf den Computer.

7) Schulung von Usern und Administratoren
   Allen Beteiligten Hilfe bei der Umsetzung geben.

8) Störfall
   Konzepte zum Umgang mit einem nicht vorhergesehenen Schaden im Falle eines Fehlers in der Politik. Dies ist Teil des Incident Response Plan.

9) Revision
   Kontrolle auf Korrektheit des Konzeptes.

Dieser Leitfaden ist ein empfohlener Plan zur Erstellung eines Konzeptes. Wichtig hierbei ist, dass dieser Plan immer wieder durchlaufen wird, da sich besonders die Punkte 2 bis 4 im Laufe der Zeit ändern. Der Plan umfasst die Erstellung einer Sicherheitspolitik und ihre Durchsetzung. Beide Punkte werden in den nun folgenden Abschnitten besprochen.

2.3 SICHERHEITSPOLITIK

2.3.1 WAS IST EINE SICHERHEITSPOLITIK?

Auf die Frage, was denn überhaupt eine Sicherheitspolitik ist, liefert die Literatur viele Antworten. Diese widersprechen sich zum Teil, vor allem auf der konzeptionellen Ebene. Manche Autoren erarbeiten bereits Umsetzungen, bevor sie die Politik überhaupt aufgestellt haben. Daher werden hier erst einmal einige Definitionen betrachtet, die zu einer eigenen führen werden.

Christiane Strauss definiert die Sicherheitspolitik in ihrer Dissertation (nach: [Nedon 00]) folgendermaßen: „Unter einer Sicherheitspolitik ist ein System von gegenseitig und auf die allgemeine Unternehmenspolitik abgestimmten Grundsatzentscheidungen [...], die ein Sicherheitsniveau festlegen, das es zu erreichen gilt und die sicherheitspolitischen Zielsetzungen bis auf die operationale Ebene einer Unternehmenshierarchie hinunterträgt.“.

B. Fraser definiert in [RFC2196] die Politik als „[...] a formal set of rules by which people who are given access to an organization’s technology and information assets must abide.”

In [Chapman 00] wird die Politik im militärischen Sinne definiert: „A policy is what determines what wars you’re going to fight and why.“

Die Definitionen haben gemeinsam, dass es sich bei der Politik um eine Zielsetzung handelt. Fraser schreibt über die Notwendigkeit einer Sicherheitspolitik: „However, you cannot make good decisions about security without first determine what your security goals are“. Die Politik betrachtet zur Festlegung der Sicherheitsziele die Werte der Organisation, in dessen Umfeld die Sicherheitspolitik angewandt wird. Dazu gilt es, die Werte zu identifizieren. Zur Festlegung des Sicherheitsniveaus ist es zudem nötig, die Gefahren zu erkennen, denen die Werte ausgeliefert sind. Die Gefahren wurden bereits in 2.1. betrachtet. Die Identifizierung der Werte und die Analyse der Gefahren sind im Vorgang des Risk Assessment zusammengefasst. Dabei wird auch der mögliche Schaden betrachtet, und ob dieser im Falle eines Auftretens für den Einsatzkontext zu verkraften ist. Dieser Vorgang ist Teil der Erstellung einer Sicherheitspolitik.

Unter einer Sicherheitspolitik definieren wir:

> Eine Sicherheitspolitik ist eine Zielsetzung, die durch eine Menge von abstrakten Regeln festlegt, welche Sicherheit beim Umgang mit Informationen erreicht werden soll. Die Festlegung der Sicherheit bedeutet zu wissen, welche Werte gegen welche Gefahren geschützt werden sollen und welche Risiken mit den nicht betrachteten Gefahren eingegangen werden.

Die wesentlichen Aufgaben der Politik sind die Festlegung der Ziele und der Maßnahmen zur Erreichung der Ziele. J. Nedon schreibt dazu in [Nedon 00]: „Die Gesamtheit aus den Zielentscheidungen und den Maßnahmen zur Zielerreichung, angepasst auf das aktuelle Umfeld, wird als Sicherheitspolitik verstanden."

Zu bemerken ist, dass eine gute Sicherheitspolitik kein Garant für absolute Sicherheit ist. Je besser die Politik, desto höher ist die Zusicherung, dass man den hier aufgestellten Sicherheitsmaßnahmen vertrauen kann. Der Umfang der Sicherheit wird aber auch durch den Einsatzkontext beeinflusst, der zu Kompromissen führt.

2.3.2 ANFORDERUNGEN AN DIE SICHERHEITSPOLITIK

Der Hauptzweck der Politik ist die Informierung der IT-Benutzer über die notwendigen Maßnahmen für einen sicheren Umgang mit der Informationstechnologie [RFC2196]. Mit den Benutzern werden auch Administratoren und Führungskräfte angesprochen. Anwender der IT-Systeme sind diejenigen, die mit und nach den Einschränkungen der Politik leben. Daher ist es wichtig, dass sie die in der Politik aufgestellten Regeln auch einhalten. Um dies zu erreichen, muss die Politik für die Anwender verständlich gehalten werden. Dazu ist es nach [Chapman 00] wichtig, die Politik umgangssprachlich zu verfassen und Erklärungen beizufügen. Des weitern ist es sinnvoll, alle Anwender im Umgang mit den Einschränkungen zu schulen.

Die Politik soll über einen langen Zeitraum angewandt werden. Dazu ist es nötig, sie flexibel gegenüber Änderungen des Einsatzkontextes zu halten. Dies wird nicht nur bei organisatorischen Zwecken wie die Umstrukturierung der Hierarchie deutlich, sondern besonders bei der verwendeten Hardware und Software. Dazu ist es sinnvoll, keine technischen Details in die Politik aufzunehmen (nach [Chapmann 00]). [steht doch schon im Satz zuvor]

Unter gewissen Umständen kann es vorkommen, dass die Politik nicht anwendbar ist. So kann es sein, dass ein Administrator auf Benutzerdaten zugreifen muss, obwohl die Politik verbietet, auf andere als seine eigenen Daten zuzugreifen. Wann eine Ausnahme gemacht werden kann, wird durch eine befugte Instanz entschieden. Dazu ist es notwendig, Zuständigkeiten zu definieren, wobei wie im Sinne einer flexiblen Politik auch von konkreten organisatorischen Details abstrahiert werden sollte. Dazu ein Zitat aus [Garfinkel 96]: „What the policy should *not* do is list specific threats, machines, or individuals by name [...]".

Die in der Sicherheitspolitik aufgestellten Anforderungen müssen des weitern durchsetzbar sein. Dabei kann es aber einem Konflikten zwischen Benutzbarkeit und Sicherheit kommen. Der Konflikt ist nach [RedHat01] das Dilemma der Sicherheit, für dessen Lösung es keinen defini-

tiven Weg gibt. Hier ist ein Kompromiss zu finden. Auch zwischen Kosten und Sicherheit ist ein Kompromiss zu finden, wobei die Politik aber das Minimum der Sicherheit festlegt.

Nicht in die Sicherheitspolitik gehören nach [Chapman 00] neben den bereits im Zuge der Flexibilität angesprochenen technische Details, die Sichtweisen Außenstehender und Angelegenheiten, die nichts mit der Sicherheit im Allgemeinen zu tun haben.

2.3.3 STRUKTUR DER SICHERHEITSPOLITIK

Eine Sicherheitspolitik kann sowohl aus einem zentralen als auch aus mehreren Dokumenten bestehen. Nach [Garfinkel 96] gibt es bei der Formulierung der Sicherheitspolitik drei Ansätze. Der erste Ansatz ist eine allgemeine Sicherheitspolitik, die gängige Zielssetzungen in Bezug auf die Sicherheit zu umfasst. Beim zweiten Ansatz besteht die Sicherheitspolitik aus verschiedenen Teilpolitiken, die nach Fraser ([RFC2196]) Komponenten genannt werden. Es handelt sich hierbei nach [Nedon 00] um Politiken der unteren Ebenen. Die Komponenten sind z.B. eine Zugriffspolitik, eine Authentikationspolitik, eine Beschaffungspolitik oder eine Firewallpolitik. Diese spezifischen Teilpolitiken sind aber nicht als Auswahl zu verstehen, sondern als Teile der gesamten Sicherheitspolitik. Der dritte Ansatz nach [Garfinkel 96] besteht aus einer allgemeinen Sicherheitspolitik, die durch Standards und Richtlinien verfeinert wird. Die Standards definieren nach [SANS 01] Anforderungen für Systeme und Prozeduren, die eingehalten werden müssen, während die Richtlinien Vorschläge sind. Richtlinien sind aber keine Festlegung, können daher bei der Umsetzung unbeachtet bleiben. In diesem Teil können daher unter Beachtung der Flexibilität technische Details mit einfließen.

Bei jedem Ansatz muss die Sicherheitspolitik das gesamte Einsatzumfeld umfassen (nach [Nedon 00]).

2.3.4 SICHERHEITSPOLITIK IM BRENNPUNKT FIREWALL

Im Rahmen des Themas dieser Arbeit wird der Einfluss der Sicherheitspolitik auf Firewalls betrachtet. Zu den angesprochenen Teilpolitiken kann nach [BSI 99] auch eine Firewallpolitik gehören. Im Folgenden soll ein grober Überblick über den Umfang der Firewall-Politik des BSI gegeben werden Dieser Überblick richtet sich nach Punkt M2.71 des IT-Grundschutzhandbuch, das vom Bundesamt für Sicherheit in der Informationstechnik (BSI) [BSI 99] herausgegeben wurde. Es werden dabei Anforderungen an die Kommunikationsbeziehungen aufgestellt. Der Punkt M2.71 stellt dazu folgende Fragen heraus, die bei der Festlegung der Anforderungen betrachtet werden sollten:

- Welche Informationen dürfen durch die Firewall nach außen hindurch bzw. nach innen hineingelassen werden?
- Welche Informationen soll die Firewall verdecken?
- Welche Authentisierungsverfahren sollen innerhalb des zu schützenden Netzes bzw. für die Firewall benutzt werden?
- Welche Zugänge werden benötigt?
- Welcher Datendurchsatz ist zu erwarten?

Dies sind Anforderungen an die Kommunikationsbeziehungen, also eine Zielsetzung, die der Kommunikationspolitik zugerechnet wird. Dabei sollte von der Firewall abstrahiert werden.

Nachdem die Anforderungen an die Kommunikationsbeziehungen in der Sicherheitspolitik aufgestellt wurden, können die nötigen Dienste gewählt werden. Die Wahl der Dienste kann anhand der nötigen Informationen, die die Firewall passieren dürfen, erfolgen. Dabei ist aber zu beachten, dass die Sicherheitspolitik die Dienste und keine Protokolle betrachtet. Protokolle sind technische Details, die sich trotz Verwendung des gleichen Dienstes ändern können.

Des Weiteren sind auch organisatorische Maßnahmen festzulegen. So ist zu klären, wer für die Implementierung, Pflege und Kontrolle des Betriebes verantwortlich ist. Auch ist festzulegen, welche Restrisiken zu erwarten sind und welcher Schaden daraus resultieren kann. Dabei ist zu betrachten, ob der maximal auftretende Schaden verkraftbar ist. Sollte dies nicht der Fall sein, so müssen zusätzliche Maßnahmen ergriffen werden, um das Risiko zu minimieren. Zudem sollte geklärt werden, wie weit die Benutzbarkeit durch die Firewall eingeschränkt wird, und in wie weit diese Einschränkungen durchsetzbar sind.

Durch die Implementation der Firewall erfolgt die Umsetzung der Sicherheitspolitik. Dabei sei zu bemerken, dass die Regelmenge der Firewall eine Implementation der Policy, aber keine Policy an sich ist.

2.4 UMSETZUNG DER SICHERHEITSPOLITIK

Nachdem nun besprochen wurde, was eine Sicherheitspolitik ist, beschäftigen sich die nächsten Abschnitte damit, wie ein Konzept zur Umsetzung der Politik aussehen soll. Zur Durchsetzung der Politik mit technischen Mitteln, muss ein Abbild der Politik auf eine technische Repräsentation gefunden werden. Zur Unterstützung der Umsetzung hilft ein Sicherheitsmodell. Innerhalb der Umsetzung werden Strategien angewandt, die zu einem am Computer implementierten Abbild der Politik führt. Die dabei verwendeten Grundsatzstrategien der IT-Sicherheit werden ebenfalls vorgestellt.

Wegen der Thematik dieser Arbeit, wird in diesem Abschnitt vorrangig die Durchsetzung der in der Sicherheitspolitik aufgestellten Ziele mit Hilfe einer Firewall.

2.4.1 SICHERHEITSMODELLE

Das Modell ist die explizite Festlegung der Sicherheitsbedürfnisse eines IT-Systems, in dem sich die Sicherheitspolitik widerspiegeln soll. Es unterstützt dabei die Mechanismen zur Umsetzung auf ein bestimmtes System bei der Einhaltung der in der Sicherheitspolitik aufgestellten Ziele. Die Sicherheitspolitik an sich ist nicht im Modell enthalten, sondern steht außerhalb des Modells [Russell 92]. Konkret hilft es nach [Pfleeger 97] bei

- dem Test auf Vollständigkeit und Konsistenz,
- dem Dokumentieren der Sicherheitspolitik,
- der Implementation der Politik und
- der Überprüfung der Implementation in Bezug auf die Anforderungen.

Die erstellte Politik wird innerhalb des Prozesses der Sicherheit durch die Revision auf Konsistenz und Vollständigkeit überprüft. Damit kann ausgeschlossen werden, dass ein Fehler in der Politik eine bereits bestehende Umsetzung unbrauchbar macht.

Die Dokumentation der Sicherheitspolitik erläutert genauer die Mechanismen, die in der Sicherheitspolitik zur Erreichung der Zielsetzung beschrieben werden. Dabei wird näher auf die Eigenschaften des Systems eingegangen, dass die Sicherheitspolitik implementieren wird.

Die Implementation ist die Durchsetzung der Sicherheitspolitik durch das System. Dabei werden die Mechanismen der Sicherheitspolitik bereits durch die Richtlinien der Dokumentation auf Systemeigenschaften abgebildet.

Durch die Sicherheitspolitik und ihre Dokumentation sind Sicherheitsziele und die Eigenschaften des Systems spezifiziert. Nach der Implementation stellt sich die Frage, ob die Implementation die spezifizierten Anforderungen wirklich implementiert. Dies wird durch Testverfahren überprüft. E. Dijkstra sagte dazu: „Testing can reveal the presence of bugs, but not their absence.“

Die Implementation folgt den vorher aufgestellten Methoden und muss natürlich für einen fehlerfreien Ablauf kontrolliert werden.

2.4.2 DOKUMENTATION

Eine Dokumentation ist immer dazu da, Sachverhalte näher zu erklären. Eine Dokumentation der Sicherheitspolitik sollte in Handbüchern zur Sicherheitspolitik verfasst werden. In diesen werden nach [Nedon 00] „die in der Sicherheitspolitik festgelegten Verfahren und Handlungsanweisungen“ näher beschrieben. Dabei werden auch „technische Details erläutert, von denen in der IT-Sicherheitspolitik abstrahiert wurde“.

Das Orange Book sieht diese Dokumentation innerhalb der Security Features User Guide vor. Hierbei werden die durch das System, im Fall des Orange Books die Trusted Computing Base, gebotenen Sicherheitsmechanismen und die Richtlinien bei der Benutzung der Mechanismen beschrieben. Des Weiteren wird die Interaktion zwischen Mechanismen und Richtlinien geklärt.

Morrie Gasser schreibt in [Gasser 88] „Strictly speaking, the computer obeys security properties while people obey a security policy.” Ein Computersystem, wie eine Firewall, verfügt über Sicherheitseigenschaften. Innerhalb der Dokumentation werden die Ziele der Sicherheitspolitik durch Richtlinien in Systemeigenschaften abgebildet.

Mit der Dokumentation der Systemeigenschaften ist die Dokumentation aber nicht vollständig. Ebenfalls zu dokumentieren sind die Prozessphasen „Implementation“ und „Test“.

M. Raepple schlägt dazu in [Raepple 01] vor, die Beschreibung der technischen Maßnahmen in Sicherheitsarchitektur und Implementationsvorschrift zu trennen. Hierbei werden die technischen Mittel zur Durchsetzung identifiziert, spezifiziert und Vorgaben an einen Test zur Überprüfung ihrer Leistung festgelegt. Des Weiteren schlägt er ein Betriebskonzept vor, in der die Dokumentation der organisatorischen Maßnahmen festgelegt wird.

Dadurch sind bin [Raepple 01] sowohl die technischen als auch die organisatorischen Maßnahmen dokumentiert. Dabei integriert er die Phasen „Implementation“ und „Test“, die im Folgenden besprochen werden.

2.4.3 IMPLEMENTATION

Wie in Abschnitt 2.2 besprochen, müssen die Eigenschaften der Sicherheitspolitik in ein auf Computer abbildbares Modell umgesetzt werden. Dazu muss eine gewisse Vorgehensweise entwickelt werden, wie diese Umsetzung durchgeführt werden soll.

Nach [Chapman 00] wird im militärischen Sinne zwischen den drei Begriffen Politik, Strategie und Taktik unterschieden.

„A policy is what determines what wars you are going to fight and why. A strategy is the plan for carrying out the war. A tactic is a method for carrying out the strategy.“

Die Strategie ist ein Plan für die Umsetzung der Sicherheitspolitik. Ein solcher Plan sollte nach Fraser [RFC2196] auf einer höheren Ebene als die Politik stehen und als ein Rahmenwerk von Richtlinien bestehen, in den die Politik passt. Die Strategie ist also Teil der Richtlinien innerhalb der Dokumentation der Sicherheitspolitik. Hiermit wird der Umfang der Richtlinien klar. Er umfasst die Basisstrategien, mit denen das in der Politik definierte Niveau der Sicherheit erreicht werden kann. Mögliche Basisstrategien werden im folgenden Abschnitt besprochen. Zudem besagen die Richtlinien, welche konkreten Dienste von dem System angeboten werden, und welche Benutzergruppen diese Dienste nutzen dürfen.

Die Taktik ist die Methode zur Umsetzung der Strategie. Im Blickpunkt der Firewall umfasst die Taktik die Erstellung und Pflege einer Firewall bis ins kleinste technische Detail. Die Firewall mit ihrer Regelmenge ist eine Implementierung der Sicherheitspolitik.

Manche Quellen, insbesondere die technischen Handbücher der Firewalls, betrachten die Regelmenge der Firewall als die Sicherheitspolitik. Die Regelmenge ist aber nur ein Ausdruck der Sicherheitspolitik. So weist Lance Spitzner in [Spitzner 00] darauf hin: „Management defines the security policy, which states what is to be enforced. The firewall is a technical tool, which is how the policy gets enforced”.

Die Mittel, die den Administratoren zur Bildung der Firewall zur Verfügung stehen werden in Kapitel 3 betrachtet.

2.4.4 BASISSTRATEGIEN DER SICHERHEIT

Die Strategie ist ein mögliches Handeln, um die in der Sicherheitspolitik aufgestellten Sicherheitsanforderungen umzusetzen. Dabei gibt es mehrere grundlegende Prinzipen der Sicherheit, die in die Strategie zur Durchsetzung von Sicherheit eingearbeitet werden können. Diese sind:

- Defense in Depth
  Mehrer Schutzmechanismen werden zusammengeschaltet. Dabei verstärken und sichern sie sich gegenseitig. Durch die Verstärkung können verschiedene Mängel mehrfach geprüft werden, um so dem Angreifer eine größere Hürde entgegenzustellen. Das Zusammenschalten gleicher Mechanismen ist möglich.

- Diversity of Defense

  Auch dieses Prinzip beinhaltet die Zusammenschaltung mehrerer Schutzmechanismen. Im Unterschied zu Defense in Depth werden hier aber verschiedene Schutzmechanismen zusammengeschaltet. Damit soll zum einen ein breiteres Band von Risiken gedeckt werden, zum anderen soll eine mögliche Schwäche eines Schutzmechanismus durch einen anderen ausgeglichen werden.

- Choke Point, Common Point of Trust

  Gefahren kommen auf mehreren Wegen in die zu sichernde Umgebung. Ziel ist es, diese Wege zu einem schmalen Pfad zusammenzuführen, um sie an einem Kopplungspunkt, dem Common Point of Trust, der Kontrolle zu unterziehen. Eine besondere Form dieses Prinzips ist der Single Point of Access. Hierbei beschränkt sich der Pfad auf einen einzigen Kopplungspunkt.

- Weakest Link

  „Eine Kette ist immer nur so stark, wie ihr schwächstes Glied." Diese Weisheit gilt auch für die Sicherung einer Umgebung. Angriffe auf diese richten sich meist auf das schwächste Glied der Sicherungsmaßnahmen. Ziel ist es diese, falls möglich, zu vermeiden und zur Not besonders zu beobachten.

- Grundhaltung des generellen Verbots

  In dieser Grundhaltung ist alles, was nicht erlaubt ist, grundsätzlich Verboten. Diese Einstellung gewährleistet eine höhere Sicherheit als eine generelle Erlaubnis, denn man kann sich mit jeglichen Risiken der erlaubten Eigenschaften vertraut machen.

- Grundhaltung der generellen Erlaubnis

  Diese Grundhaltung ist das genaue Gegenteil zum generellen Verbot. Alles, was nicht ausdrücklich verboten ist, ist erlaubt. Für die Benutzer kann dies in Bezug auf die Nutzung neuer Dienste vom Vorteil sein, da hier kein administrativer Aufwand nötig ist. Doch in dieser Haltung liegt eine große Gefahr. Nicht alle Sicherheitslücken sind heute bekannt. Daher kann es vorkommen, dass eine neue Angriffstechnik eventuell noch nicht von einer Verbotsregel abgedeckt ist. Somit stellt diese Lücke ein potentielles Sicherheitsrisiko dar.

Zu beachten ist hierbei, dass die Basisstrategien zur Sicherheitspolitik und dem Ergebnis des Risk Assessment passen müssen.

2.5 GRUNDBEGRIFFE DER NETZWERKTECHNOLOGIE

Bevor auf die zurzeit wohl am häufigsten verwendete Protokoll-Familie TCP/IP eingegangen wird, folgt nun eine kurze Beschreibung des theoretischen Kommunikationsmodells der International Standard Organisation (ISO).

2.5.1 OSI-ARCHITEKTURMODELL

Als Grundlage für die Kommunikation zwischen Rechnernetzwerken dient das OSI-Architekturmodell, welches von der ISO erarbeitet wurde. OSI ist ein theoretischer Ansatz, um zu verdeutlichen wie Computer miteinander kommunizieren können. Der folgende Abschnitt ist eine kurze Darstellung des OSI-Architekturmodells und bezieht sich auf die Beschreibung im [Kerner 95]. Sie ist nur insoweit ausgeführt, wie sie für das Verständnis dieser Baccalaureats-Arbeit notwendig ist.

Die Abkürzung OSI steht für Open Systems Interconnection. Dieser Standard unterteilt die Netzwerkarchitektur in 7 Schichten. Jede dieser Schichten erfüllt einen bestimmten Dienst oder eine bestimmte Aufgabe.

- Schicht 1
  Die Bitübertragungsschicht stellt ungesicherte Verbindungen zwischen Systemen für die Übertragung von Bits zur Verfügung.

- Schicht 2
  Die Sicherungsschicht dient der Fehlerkontrolle zwischen zwei direkt verbundenen Rechnern.

- Schicht 3
  Die Vermittlungsschicht ermittelt Wege durch das Vermittlungsnetz.

- Schicht 4
  Die Transportschicht dient der Fehlerkontrolle zwischen den Endsystemen.

- Schicht 5
  Die Kommunikationssteuerungsschicht wird für Dialogfunktionen gebraucht, zum Beispiel setzt sie Synchronisationspunkte.

- Schicht 6
  Die Darstellungsschicht ermöglicht die Behandlung unterschiedlicher Datendarstellungen, zum Beispiel von Zahlen im ASCII- und Unicode-Format.

- Schicht 7

  Die Anwendungsschicht stellt Mittel zur Kooperation zwischen zwei verteilten Anwendungsprozessen zur Verfügung, zum Beispiel für den Dateitransfer.

In jedem offenem System entsteht so eine Struktur aufeinander aufbauender Schichten, wie die Abbildung zeigt:

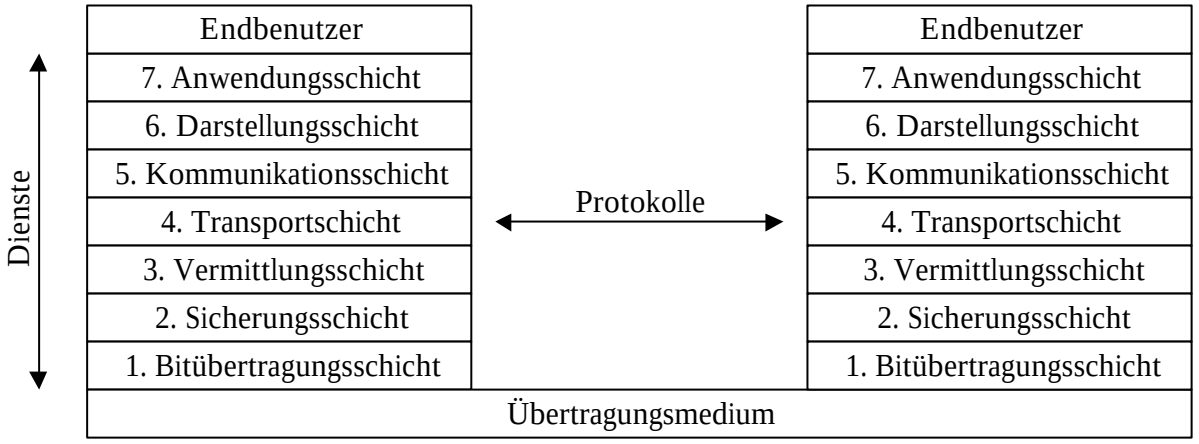


Abb. 1: OSI-Schichtenmodell

Unter der Bitübertragungsschicht liegt das Übertragungsmedium, zum Beispiel Glasfaserkabel oder Kupferdraht.

Das OSI-Architekturmodell hat sich in der praktischen Anwendung so nicht durchgesetzt. Zum Beispiel werden einige Schichten zu einer zusammengefasst, so dass die Schichtung so nicht auftritt. Deshalb wird das Architekturmodell von B. Wolfinger als „Referenzmodell“ bezeichnet.

Die statische Struktur

Jede Schicht benutzt die darunter liegende und unterstützt die darüber liegende Schicht. Die Leistung, die eine Schicht einer darüber liegenden Schicht anbietet, wird als Dienst bezeichnet. In den Diensten sind die Leistungen der darunter liegenden Schichten inbegriffen. Außerdem gibt es Dienstelemente (Service Primitives), welche Kommandos zur Inanspruchnahme von Diensten und Ergebnismeldungen sind. Die Schnittstelle zwischen zwei Schichten, über welche Dienste mittels der Dienstelemente angefordert und bereitgestellt werden, heißen Dienstzugangspunkte (Service Access Point, kurz SAP).

Die Aufgaben, die jede Schicht zu erfüllen hat, werden Instanzen genannt. Instanzen verkehren sowohl mit Instanzen der darunter oder darüber liegenden Schicht (d. h. vertikal), als auch mit Instanzen der gleichen Schicht eines anderen Rechners (d. h. horizontal). Beim Aufruf einer Instanz wird eine Kopie von ihr erzeugt, welche als Vorkommnis bezeichnet wird. Ein Vorkommnis wird nur während des Betriebes erzeugt und gehört deshalb zur dynamischen

Struktur. Im normalen Netzwerkbetrieb hat ein Rechner mehrere Verbindungen gleichzeitig zu anderen Rechnern aufgebaut. Deshalb gibt es auch mehrere Verbindungen zwischen den Schichten und somit hat eine Instanz mehrere Vorkommnisse. Zur Unterscheidung der Verbindungen und der dazugehörigen Vorkommnisse gibt es die Verbindungsendpunkte (Connection End Points, kurz CEP).

Für die Kommunikation durch Austausch von Dateneinheiten zwischen zwei oder mehreren Kommunikationspartnern werden Regeln gebraucht. Diese Regeln werden in ihrer Gesamtheit als Protokoll bezeichnet. Jede Schicht braucht ein Protokoll, um mit der gleichen Schicht eines anderen offenen Systems zu verkehren. Protokolle dienen im Wesentlichen dem Verbindungsauf- und -abbau und der Datenübertragung.

Die Protokoll- und Nutzdaten werden mittels Paketen übertragen. Wenn die Pakete zum Beispiel der Schicht 7 horizontal von einer Partnerinstanz zu einer anderen übertragen werden, durchlaufen die Pakete zuerst vertikal alle darunter liegenden Schichten, bis das Übertragungsmedium erreicht wird. Auf dem Übertragungsmedium werden die Pakete zu dem Zielrechner weitergeleitet. Beim Adressaten werden die Pakete wiederum Schicht für Schicht nach oben bis zur Zielschicht gereicht.

Im einzelnem sehen die Pakete folgendermaßen aus: Beim vertikalen Austausch von Paketen zwischen zwei Schichten hat jedes Paket einen Kommandoteil, auch Interface Control Information (ICI) genannt. Des weiteren enthalten die Pakete Nutzdaten, welche Service Data Unit (SDU) heißen. Zusammen bilden der Kommandoteil und die Nutzdaten ein Dienstelement. Wenn dieses zum Beispiel zum Aufbau oder Abbau einer horizontalen Verbindung dient, enthält es nur Steuerinformationen aber keine Nutzdaten. Um die Nutzdaten an die Folgeinstanz im eigenen System zu übergeben, befinden sich die Steuerinformationen wohl getrennt von den Nutzdaten im ICI. Steuerinformationen sind zum Beispiel der Zielort des Paketes. Die SDU wird unverändert an die Folgeinstanz übergeben. Zusammen bilden ICI und SDU eine Interface Data Unit (IDU). Werden die Pakete horizontal zwischen zwei Protokollpartnern ausgetauscht, dann enthält die Protocol Data Unit (PDU) weiterhin eine SDU und eine Protcol Control Information (PCI). Aber beim Weiterleiten von einer Schicht zur darüber oder darunter liegenden Schicht werden statt der ICI noch Steueranweisungen für den Protokollpartner in der PCI hinzugefügt. Dabei ergänzt jede Schicht, die ein Paket bekommt, die PDU um weitere Steuerinformationen bis das Paket das Übertragungsmedium erreicht. Dort wird es an den Empfänger weitergeleitet. Beim Empfänger sieht sich jede Schicht die äußeren Steuerinformationen an, schneidet diese vom Paket ab und übergibt das Paket an die nächst höhere Schicht. Also versieht jede Schicht beim Sender die Nutzdaten mit Steuerinformationen für die entsprechende Schicht beim Empfänger.

Beim Empfänger wird das Datenpaket aus Nutzdaten und Steuerinformationen wieder Schicht um Schicht ausgepackt, so dass jede Schicht auf der Senderseite die entsprechende Schicht auf Empfängerseite mit den für sie notwendigen Steuerinformationen versorgen kann. Soweit zur statischen Struktur, als nächstes folgt die dynamische Struktur des OSI-Architekturmodells.

Die dynamische Struktur

Die dynamische Struktur beinhaltet den Betrieb auf der statischen Struktur, der Ablaufen muss, um Datenpakete zwischen zwei Kommunikationspartnern auszutauschen. Wie die Pakete von einer Schicht zur nächsten Schicht übergeben werden, wurde bereits gezeigt, deshalb wird im Folgenden davon abstrahiert und nur die Erbringung eines Dienstes an den Dienstbenutzer betrachtet.

Die Kommandos, die dafür benutzt werden, stehen, wenn sie von einer Schicht zur darunter oder darüber liegenden Schicht (vertikal) übergeben werden, im ICI. Für die Übertragung zwischen den Partnerschichten (horizontal) werden die Kommandos in einer PCI zusammengefasst und mit der SDU in einer PDU gesendet. Dazu schickt der eine Kommunikationspartner, der den Verbindungsaufbauwunsch hat, ein „Connect Request“ an den gewünschten Kommunikationspartner. Dieser erhält eine „Connect Indication“ und weiß nun, dass ein Verbindungsaufbauwunsch durch den anderen besteht. Wenn er bereit ist, die Verbindung aufzubauen, schickt er als Antwort ein „Connect Response“. Dies kommt als „Connect Confirm“ beim anfragenden Kommunikationspartner an. Somit ist die Verbindung aufgebaut und die Daten werden ausgetauscht. Ist der Kommunikationspartner nicht bereit eine Verbindung aufzubauen, schickt er ein „Disconnect“ an den anderen und dieser weiß, dass der Verbindungsaufbau gescheitert ist. Der Abbau folgt dem gleichen Muster. Nur das hierbei die Kommandos: „Disconnect Request“, „Disconnect Indication“, „Disconnect Response“ und „Disconnect Confirm“ heißen.

Den Austausch von Kommandos (Request, Indication, Response, Confirm) zum Aufbau oder Abbau einer Verbindung zwischen zwei Partnervorkommnissen wird Hand-Shake genannt. Der eben beschriebene Kommandoaustausch ist ein „Zweiwege-Hand-Shake“. Dies ist der einfachste Fall für einen gegenseitig akzeptierten Verbindungsaufbau.

Häufig wird auch eine Verbindung aufgebaut, bei der Synchronisationspunkte gesetzt werden. Dabei wird dann statt eines Connect Request als erstes ein „Syn“ zur Synchronisation der beiden Kommunikationspartner gesendet. Der Kommunikationspartner antwortet darauf ebenfalls mit „Syn“ und vorher als Bestätigung des Verbindungsaufbaus mit „Ack“. Werden „Syn“ und „Ack“ zusammen geschickt liegt der Spezialfall des „Dreiwege-Hand-Shakes“ vor. Am Ende des

Verbindungsaufbaus wird das empfangene „Syn“ und „Ack“ noch einmal durch ein „Ack“ bestätigt. Den „Zweiwege-Hand-Shake“ und den „Dreiwege-Hand-Shake“ soll folgende Abbildung verdeutlichen:

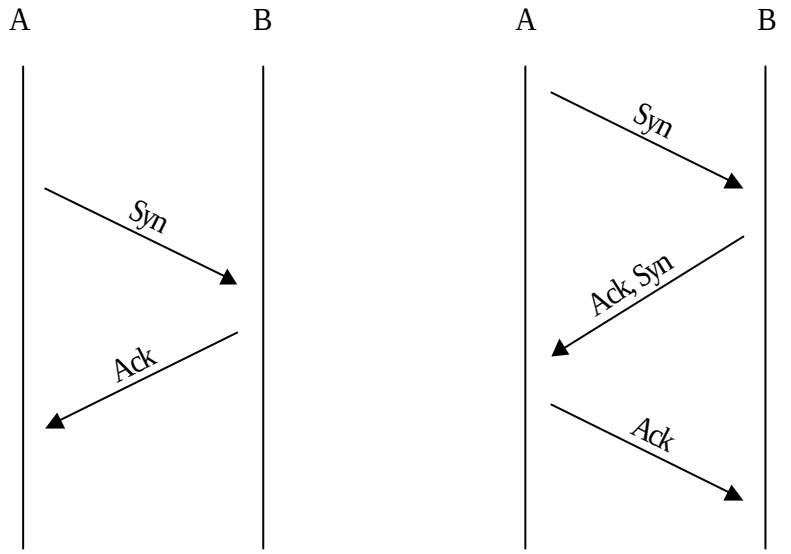


Abb. 2: Verbindungsaufbauvarianten

2.5.2 INTERNET-PROTOKOLLE

Die Internet-Protokollfamilie wird häufig „TCP/IP“ genannt. Die beiden Protokolle IP und TCP sind zwar sehr relevant für die Kommunikation im Internet, doch beinhaltet der TCP/IP-Stack auch noch andere Protokolle. Auch die folgende Abbildung zeigt nur eine Auswahl:

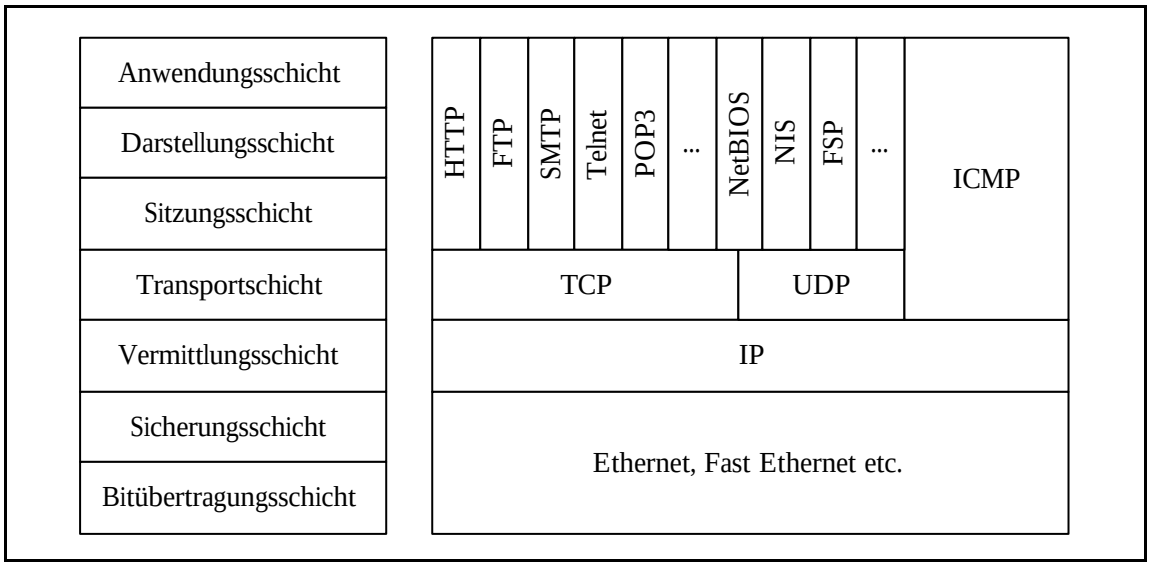


Abb. 3: Vergleich des OSI-Modells mit der TCP/IP-Protokollfamilie nach [Tanenbaum 98]

Im Folgenden werden jetzt einige für die Versuchsreihe relevante Protokolle erläutert (nach [Bonnard 97]):

Internet Protocol (IP)

Das Internet Protocol, dessen Version 4 bereits in [RFC791] 1981 festgelegt wurde, regelt die Kommunikation zwischen zwei Computern verbindungslos, das heißt ohne Aufbau einer Verbindung insbesondere ohne Hand-Shake. IP geht nicht davon aus, dass die unteren Schichten Fehlererkennung durchführen. Aus diesem Grund sind die Header-Informationen von IP mit Checksummen versehen, nicht aber die Protokolldateneinheit. Probleme der Verlässlichkeit und Flusssteuerung überlässt IP der Transportschicht.

IP bietet den höheren Schichten diverse Routing-Dienste (*IP-Source-Routing, IP-Default-Routing, Routing-Operations* und *Route-Recording*) an. Des Weiteren können *Time-Stamps* beim Durchlaufen von Routern zu den Datenpaketen hinzugefügt werden, um eine bessere Netzadministration zu ermöglichen. Als ganz wesentlichen Dienst bietet IP noch das *Fragmentieren* und *Reassemblieren* an. Damit können große Datenpakete in kleinere zerlegt und wieder zusammengesetzt werden. Dieses wird zum Beispiel auf Teilstrecken mit geringer Durchsatzrate verwendet, um eine Gleichberechtigung des Datenverkehrs auf diesem Teilstück zu ermöglichen.

IP verwendet 32 Bit lange Adressen, die aus Netz- und Knotenadressen zusammengesetzt sind. Dabei bestimmt die Klasse der IP-Adresse die Anzahl verfügbarer Knotenadressen. Die in späteren Versuchen verwendeten Adressen entstammen der Klasse C und beginnen dementsprechend mit 192.168.x.y (x für 256 mögliche Netze, y für 254 mögliche adressierbare Netzkomponenten je Netz).

Transmission Control Protocol (TCP)

Das Transmission Control Protocol wurde als Protokoll der Transportschicht von dem Verteidigungsministerium der USA (DoD) extra für IP entwickelt und in [RFC793] aber so allgemein festgeschrieben, dass es auch auf anderen Protokollen der Vermittlungsschicht aufbauen kann. Es arbeitet verbindungsorientiert, aber auf einem verbindungslosen Protokoll der Vermittlungsschicht; damit muss es die Flusssteuerung inklusive Verbindungsauf- und -abbau und die Verlässlichkeit der Verbindung selber regeln. TCP nutzt dazu beim Verbindungsauf- und -abbau den Dreiwege-Hand-Shake-Algorithmus. Auch alle anderen Datenpakete (bis auf eine Quittierung selber) werden beim Empfang durch ACK für Acknowledge bestätigt. Dazu werden den Paketen beim Absenden Nummern beigefügt, die um eins erhöht im Acknowledge-Paket (kann auch Bestandteil eines anderen Datenpaketes sein) wieder zurückgesendet werden. Über die Nummerierung ist auch eine Erkennung verlorener Datenpakete möglich, wodurch überhaupt erst verbindungsorientierte Kommunikation ermöglicht wird.

TCP bietet für die verbindungsorientierte Kommunikation den oberen Schichten folgende Dienste an:

- *Connection-oriented Data Management*, für die eigenverantwortliche Umsetzung eines verbindungsorientierten Datenaustausches
- *Reliable data transfer* wird realisiert durch den Neuversand der Datenpakete, bis ein Acknowledge für dieses Paket eingetroffen ist
- *Push function* zur Erzwingung der unmittelbaren Absendung eines TCP-Paketes durch eine Anwendung, die dieses Protokoll benutzt
- *Flow Control* zur Begrenzung der Übertragungen gemäß den Pufferkapazitäten
- *Resequencing* zur Analyse der Zulässigkeit eines TCP-Pakets, da beide Kommunikationspartner durch den Sequenznummernvergabealgorithmus wissen, welche Sequenznummern als Teil eines Pakets sie als Nächstes erwarten dürfen
- *Multiplexing*, um über eine Verbindung mehrere Sitzungen zu übertragen
- *Full-duplex transmission* für eine gleichzeitige bidirektionale Kommunikation ohne Übergabe eines Sendetokens (also im Gegensatz zu Halbduplex)
- *Graceful close* zur beiderseitig bestätigten Beendigung einer Verbindung erst nach Empfang aller erwarteten Acknowledges
- *Passive* & *active open* zum Aufbau von Verbindungen – *passive open*, wenn die höhere Schicht einen Verbindungsaufbau von dem Kommunikationspartner erwartet, und *active open* bei Wunsch eines aktiven Verbindungsaufbaus des Hosts durch TCP
- *Transmission Control Blocks* zur Zwischenspeicherung diverser TCP-spezifischen Merkmale, wie die verwendeten Sockets, Zeiger auf die verwendeten Puffer, Datenpakete, die eventuell noch einmal mangels Quittierung durch den Empfänger gesendet werden müssen.

In der Struktur eines TCP-Segments (PDU nach OSI) gibt es die Möglichkeit ein solches Paket als dringend (engl. *urgent*) zu deklarieren. Dieses Flag wird häufig bei zeitkritischen Übertragungen, wie zum Beispiel Videotelephonie, verwendet. Wie auf ein so markiertes Paket reagiert wird, hängt von den Implementierungen des Protokolls ab.

Sofern Hacker-Angriffe auf den einzelnen Diensten dieses Protokolls oder anderer basieren, werden diese noch einmal detaillierter bei den einzelnen Angriffstechniken betrachtet.

User Datagram Protocol (UDP)

Im Gegensatz zu TCP ist das User Datagram Protocol verbindungslos. Damit entfallen auch alle Mechanismen, wie Hand-Shake, Durchnummerierung der Pakete und damit verbundene „Nachforderungsalgorithmen“. Den einzigen Dienst, den UDP anbietet, ist *Multiplexing*. Auf Sicherheitsmechanismen zum Garantieren der Übertragung von Datenpakete verzichtet es, wie in [RFC782] beschrieben, gänzlich. Somit ist UDP ein sehr einfaches Protokoll mit wenig Overhead.

Internet Control Message Protocol (ICMP)

Das ICMP wird im Detail im [RFC792] beschrieben. Es wird sowohl von UDP und TCP als auch von IP verwendet, um Kontrollinformationen (verbindungslos) zu übermitteln. Dabei bietet es folgende Dienste an:

- *Time Exceeded*, für IP-Pakete, deren Schwellzeitwert überschritten ist
- *Unintelligible*, für empfängerseitige Dekodierungsprobleme des IP-Headers
- *Destination Unreachable*, für die Unerreichbarkeit der Zieladresse (Rechner, Port etc.) mit detaillierter Angabe der Gründe
- *Source Quench*, zur Flusssteuerung bei zu kleinen Puffern auf einem Gateway oder Zielhost
- *Echo Request*, zur Anforderung einer Antwort einer Netzkomponente, worüber dann die Verfügbarkeit dieser Komponente analysiert werden kann (auf der Anwendungsschicht kann dieser Dienst bei vielen Betriebssystemen durch den Befehl ping angesprochen werden; s. a. Ping of Death)
- *Echo Reply*, zur Beantwortung eines Echo Requests
- *Redirect*, für adaptives Routing
- *Timestamp* (wie bei IP beschrieben)
- *Information Request* und *Reply*, zur Bestimmung der Adresse des angeschlossenen Netzes
- *AddressMask*, zur Übermittlung der Subnetzmaske

Diese Dienste dienen eigentlich dem Versand von Kontrollinformationen. Da Kontrollinformationen aber auch häufig sicherheitsrelevant seien können, gibt es verschiedene Hacker-Angriffe, die auf ICMP basieren. Dazu zählen die Redirect-Angriffe, die nicht nur Sniffing, sondern auch Spoofing erlauben. Auch der später noch betrachtete, bereits erwähnte Ping of Death basiert auf den ICMP-Diensten.

HyperText Transfer Protocol (HTTP)

Das HyperText Transfer Protocol ist wohl das bei den Anwendern bekannteste, da es Teil der in der Werbung genannten Webadressen ist. Die Definition der Version 1.1 des HTTP ist in [RFC2616] beschrieben.

Im wesentlichen basiert HTTP auf einem verbindungsorientierten Pull-Mechanismus, bei dem ein Client zunächst einen Verbindungsaufbau über die Operation *Connection* initiiert, dann über *Request* die gewünschte Datei anfordert, diese per *Response* vom Server erhält und dann die Verbindung mittels *Close*-Operation beendet.

File Transfer Protocol (FTP)

FTP wurde für den Datenaustausch zwischen zwei Rechnern entwickelt und in [RFC959] beschrieben. Bei der Entwicklung dieses Protokolls sollte eigentlich nur eine einfache Möglichkeit des Dateienaustausches für Anwendungen beschrieben werden, jedoch hat die Realität gezeigt, dass FTP auch von den Endanwendern direkt genutzt wird.

FTP bietet dazu die Dienste:

- plattformunabhängige Verfügbarkeit von Dateien für mehrere Benutzer und
- *Third Party Transfer* zur Übertragung von Dateien zwischen FTP-Servern koordiniert oder zumindest initiiert von einem FTP-Client.

Diese Dienste ermöglichen auch eine verteilte Datenhaltung und bei entsprechender Nutzung des Third Party Transfers auch eine RAID1-ähnliche Spiegelung wichtiger Dateien.

Simple Mail Transfer Protocol (SMTP)

Das Simple Mail Transfer Protocol gehört zu den am häufigsten verwendeten Protokollen auf der Anwendungsschicht, wenn man den momentanen Gebrauch des ElectronicMail-(E-Mail-)Versandes betrachtet. Die Definition von SMTP ist aktuelle in [RFC1123] und [RFC2821] festgehalten. In der Beschreibung dieses Protokolls wird von verbindungsorientierten Diensten der unteren Schichten ausgegangen, vor allem wird TCP erwähnt.

Zur Übermittlung einer Mail wurde SMTP entwickelt. Dabei basiert es auf folgendem Sendemechanismus: Zunächst wird von einem SMTP-Client eine zumindest Halbduplex-verbindung zu einem Server per einfaches Hand-Shake aufgebaut. Als nächstes wird die Empfängeradresse (zum Beispiel 8Student@informatik.uni-hamburg.de) und dann die Absenderadresse übermittelt. Jede dieser Übermittlungen bedarf der Quittierung durch den SMTP-Server. Danach wird der E-Mail-Inhalt übertragen, quittiert und die Verbindung wieder

per Hand-Shake geschlossen. Die Verantwortung der Weiterleitung der Mail an den Empfänger liegt nun beim Server, der gegebenenfalls auch eine Unerreichbarkeitsmeldung des Empfängers an den Absender schicken kann (aber nicht muss).

Zur weiteren Übertragung der E-Mail nutzt der SMTP-Server, sofern ihm der Empfänger nicht bekannt ist, Domain Name System (DNS-) Anfragen, um den nächsten SMTP-Server auf dem Weg zum Empfänger zu ermitteln.

NetBIOS

Das NetBIOS-Protokoll, von IBM „Technical Reference PC Network“ genannt, wurde als erstes unter diesem Namen von Microsoft im Betriebssystem DOS implementiert. Es findet bei der Kommunikation kleiner Rechnergruppen (LANs) Verwendung, die gegenseitig auf die gemeinsamen Ressourcen, wie Dateien, Directories und Drucker zugreifen sollen. NetBIOS kann laut [RFC1001] sowohl verbindungsorientiert als auch verbindungslos operieren. Zusätzlich bietet es Peer-to-Peer-Kommunikation, sowie Multi- und Broadcast.

Es verfügt über Dienste, mit deren Hilfe es Ressourcen lokalisieren, Daten (zum Beispiel Dateien und Druckaufträge) senden und empfangen und zur sequenzierten, verbindungsorientierten Vollduplexkommunikation Verbindungen auf- und abbauen (*Session Service* im Gegensatz zum verbindungslosen *Datagram Service*) kann. Dabei bietet es die Möglichkeit, die Ressourcen über den *Name Service* in dem entsprechenden Netz zwar eineindeutig, aber dennoch frei zu benennen, wobei Kombinationen von Rechnername und Ressourcenamen üblich sind (zum Beispiel Kathy\c$ oder Ergo\Printer).

3 FIREWALLS

Im letzten Kapitel haben wir die Grundlagen der Netzsicherheit erläutert und sind auf Sicherheitspolitiken eingegangen. Im Folgenden befassen wir uns mit den Mechanismen, wie diese Grundlagen im Netz, oder in Spezialfällen auch auf einzelnen Systemen, umgesetzt werden. Dieses Kapitel behandelt Filtermechanismen, die als Firewalls realisiert werden. Im Kapitel darauf werden dann die IDS beschrieben. Beides sind technische Hilfsmittel, die die Sicherheitspolitik technisch für das Netz umzusetzen.

3.1 WAS IST EINE FIREWALL

Um der drohenden Angriffsgefahr aus dem Internet zu begegnen, werden in den letzten Jahren verstärkt Filtermechanismen, so genannte Firewalls, eingesetzt, um das lokale Netzwerk oder den eigenen Computer zu schützen. Was genau man unter einer Firewall zu verstehen hat, wird in der Literatur höchst unterschiedlich definiert. Viele dieser Definitionen sind sehr speziell. Oft wird der Begriff der Firewall mit dem eines reinen Paketfilters gleichgesetzt, der Datenpakete zum Beispiel anhand ihres IP-Headers analysiert. Diese Analyse findet auf Vermittlungs- oder Transportebene (nach ISO OSI-Schichtenmodell, siehe 2.5.1) statt und beachtet nicht die „Semantik“ der Daten. Andere Quellen nennen Proxies oder Application-Level-Gateways als ein Synonym für Firewalls, bei denen die Filterung auf Anwendungsebene geschieht. Diese Definitionen betonen jedoch bereits eine bestimmte Art, wie die Firewall technisch und konzeptionell realisiert wurde. Um eine möglichst allgemeine Definition der Firewall zu geben, muss man zunächst ihre eigentlichen Aufgaben und nicht ihre Funktionsweise untersuchen.

Eine Firewall dient zum Schutz vor Angriffen. Zu diesem Zweck filtert sie den Datenverkehr, der über sie geleitet werden soll. Anhand bestimmter Regeln wird entschieden, welche Datenpakete die Firewall passieren dürfen und welche nicht. Diese Regeln können normalerweise vom Benutzer der Firewall konfiguriert werden, um eine strenge oder weniger strenge Filterung zu erreichen. Dies ist eigentlich bereits alles, was zur grundlegenden Definition der Firewall gehört. Wir halten also als Definition fest:

> Als Firewall bezeichnet man eine Vorrichtung, die den Datenverkehr an einem Punkt der Datenleitung anhand bestimmter Regeln analysiert und filtert. Dabei können ausgefilterte Datenpakete die Firewall nicht passieren.

Wie bereits gesagt, kann die Art und Weise, wie die Filterung vonstatten geht, von Firewall zu Firewall unterschiedlich sein. Bei der Konfiguration einer Firewall sollte sich der Benutzer deshalb im Klaren sein, wie die verwendete Firewall arbeitet. In den folgenden Abschnitten

werden wir die beiden bekanntesten Lösungen für Firewalls näher betrachten: Paketfilter auf Transportebene und Proxies. Untersuchen wir aber zunächst, für welche Aufgaben Firewalls nun aber genau geeignet sind und für welche nicht. Bei dieser Untersuchung stützen wir uns auf [Chapman 00].

Eine Firewall fokussiert das Sicherheitsproblem eines lokalen Netzwerks auf eine bestimmte Stelle. Wenn man davon ausgehen kann, dass das Netzwerk nur von außen bedroht wird und sich keine Angreifer innerhalb des Netzes aufhalten, so braucht man lediglich allen Verkehr von und nach draußen über die Firewall zu leiten. Andernfalls müsste man jeden einzelnen Rechner mit Sicherheitsvorkehrungen ausstatten, deren Anschaffung und Wartung viel aufwendiger wäre. Durch diese Fokussierung kann zudem im gesamten Netzwerk ein einheitliches Sicherheitsniveau erreicht werden. Ohne Firewall hätte vielleicht jeder Rechner im Netz andere Sicherheitsmaßnahmen, die eventuell nicht einmal konsistent wären. Da aber aller Verkehr über die Firewall geleitet wird, ist der Standard auf allen Rechnern im Netz gleich. In Firmennetzen kann diese Eigenschaft noch zu etwas anderem genutzt werden: Die Firewall kann einzelne Bereiche des Netzwerks voneinander abgrenzen. Dadurch kann beispielsweise verhindert werden, dass sensible Daten, auf die etwa nur die Firmenleitung zugreifen soll, im Local Area Network (LAN) verbreitet werden, oder dass sich Probleme schnell im ganzen LAN ausbreiten. Wenn zum Beispiel doch ein Angriff auf das Netzwerk erfolgreich ist, kann durch die Abschottung einzelner Bereiche der Schaden gut begrenzt werden. Ein weiterer Nutzen vieler Firewalls besteht darin, den Netzverkehr von und nach draußen zu protokollieren. Dies umfasst je nach Firewall von einfachen statistischen Daten wie Auslastungsmessungen bis zum Aufzeichnen von Angriffsversuchen alle Arten der Protokollierung.

Dies sind also die Möglichkeiten beim Einsatz einer Firewall. Viele Benutzer machen jedoch den Fehler, Erwartungen an eine Firewall zu stellen, die von dieser nicht erfüllt werden können. Deshalb werden wir nun darstellen, welche oft gewünschten oder unterstellten Eigenschaften eine Firewall nicht hat.

Eine Firewall schützt nur vor Angriffen von außen. Wenn das eigene Netzwerk von innen heraus bedroht wird, so ist die Firewall nutzlos, da netzinterner Verkehr nicht von ihr gefiltert wird. Außerdem schützt sie nur vor Angriffen, die über das Netz ablaufen. Wenn maliziöse Programme beispielsweise über Disketten eingeschleppt werden, kann die Firewall dagegen wenig tun. Zudem gibt es Administratoren, die sich eine Hintertür offen halten, wie zum Beispiel eine ungefilterte Modemverbindung nach außen (um das Netz auch von außen „unkompliziert“ administrieren zu können). Da der Verkehr über diese Verbindung nicht über die Firewall läuft, bietet sie hier keinen Schutz.

Viele Benutzer sind der Meinung, ihre Firewall schütze sie auch vollständig vor Viren. Dies ist nur eingeschränkt der Fall, denn die Firewall müsste dann in den Datenpaketen feststellen, dass diese Teile eines Programms sind, was dieses Programm tun sollte und dass es sich bei dem Programm um einen Virus handelt. Da Programme aber im Regelfall fragmentiert verschickt werden und Viren zudem sehr unterschiedliche Gestalten haben können, ist dies eine fast unlösbare Aufgabe, wenn der Datenverkehr nicht hoffnungslos verlangsamt werden soll. Werden die Daten zusätzlich noch komprimiert verschickt, wird diese Aufgabe noch rechenintensiver. Natürlich ist in viele Firewalls ein Virenscanner integriert, aber dieser liefert aus den eben genannten Gründen nur einen begrenzten Schutz.

Ein weiterer Punkt kommt noch hinzu: Die Firewall arbeitet nach vorher definierten Regeln. Sie kann nur bekannte Angriffsarten abwehren, nicht etwa solche, die der Angreifer sich erst nach Erstellung der Firewall ausgedacht hat. Zudem kann die Firewall sich nicht selbst optimieren. Wenn sie nicht von Anfang an nach den Bedürfnissen der Benutzer konfiguriert wurde, filtert sie vielleicht die ganze Zeit die falschen Datenpakete und stört so den Netzbetrieb. Aber auch eine korrekt konfigurierte Firewall ist auf jeden Fall ein Verlust an Performanz, da die Filterung Zeit in Anspruch nimmt.

Soviel also zu den Vor- und Nachteilen einer Firewall. Im Folgenden wird nun die Filterung auf Transport- und Anwendungsebene besprochen.

3.2 ARBEITSWEISEN

Dieses Kapitel beschäftigt sich mit den Arbeitsweisen einer Firewall, um den Datenverkehr zu filtern. Dazu werden die Paketfilter auf Transport- und die Application-Level Gateways mit ihrem Spezialfall Proxy auf der Anwendungsschicht betrachtet.

3.2.1 PAKETFILTER

Ein Paketfilter ist die einfachste Form einer Firewall und basiert auf dem Konzept eines Routers auf IP-Ebene. Die Aufgabe des Routers besteht darin zu entscheiden, wohin ein IP-Paket mit einer bestimmten Zieladresse als nächstes geleitet werden soll. Auf diese Weise wird die für das IP-Protokoll typische verbindungslose Kommunikation realisiert. Ein Paketfilter erweitert das Routing dahingehend, dass er nicht nur entscheidet, wohin das Paket geschickt werden soll, sondern auch, ob es überhaupt weitergeleitet werden soll. Diese Entscheidung trifft der Paketfilter anhand von konfigurierten Regeln, in denen die der Firewall zugrunde liegende Sicherheits-

politik verankert ist. Dieses Kapitel betrachtet nun die Einzelheiten des Paketfilterkonzepts. Dabei orientieren wir uns erneut an [Chapman 00].

Welche Aufgaben kann ein Paketfilter erfüllen? Das hängt davon ab, wie aufwendig der Filter gestaltet wird. Im Prinzip kann ein Paketfilter nahezu jede noch so komplizierte Filterregel umsetzen, wenn nur genug Informationen aus dem Paketfluss gezogen werden. Je mehr dieser Informationen genutzt werden, desto aufwendiger wird aber der Filter, was eventuell zu Performanzeinbußen führen kann. Im einfachsten Fall werden die Filterentscheidungen nur aufgrund von drei Einträgen in einem einzelnen Paket getroffen:

- der Quelladresse des Paketes
- der Zieladresse des Paketes
- den angesprochenen Service Access Points (SAP). bzw. deren Instanziierungen Connection End Points (CEP)

Diese einfache Form der Filterung betrachtet insbesondere nicht das Datenfeld der Pakete. Ebenso wenig wird das Paket im Kontext mit anderen Paketen untersucht. Man kann mit solch einem einfachen Filter beispielsweise Verbindungen zwischen bestimmten Rechnern erlauben oder sperren. Es ist jedoch nicht möglich, bestimmten Anwendern den Aufbau einer Verbindung zu erlauben und anderen nicht, oder das ausschließliche Versenden bestimmter Dateitypen, weil der Filter das Datenfeld ja überhaupt nicht betrachtet.

Ein etwas leistungsfähigerer Paketfilter lässt sich bereits dadurch realisieren, dass man nicht nur einzelne Pakete betrachtet, sondern die Pakete im Kontext mit anderen Paketen untersucht. Dadurch ist es beispielsweise möglich, den Vorgang eines Verbindungsaufbaus festzuhalten und Datagramme von außen nur dann durchzulassen, wenn sie Antworten auf Datagramme von innen sind. Diese Form der Filterung nennt man *dynamische Paketfilterung*, weil das Verhalten des Filters nicht nur auf den konfigurierten Regeln, sondern auch auf dem protokollierten Systemzustand beruht. Leider bringt das dynamische Paketfiltern auch Nachteile mit sich. Es erhöht die Last, die von der Firewall verkraftet werden muss und ermöglicht so Denial-of-Service-Attacken. Zudem muss der Filter gesammelte Statusinformationen zu einem geeigneten Zeitpunkt wieder löschen können. Da nicht alle Datagramme beantwortet werden, muss der Filter die Regel, die auf die Antwort wartet, irgendwann wieder löschen. Einen Zeitpunkt für die Löschung festzulegen, ist aber sehr schwierig. Wenn die dynamisch erzeugte Regel zu früh gelöscht wird und doch eine Antwort auf das Paket erfolgt, so wird diese abgelehnt, obwohl sie eigentlich den Filter passieren sollte. Werden solche Regeln sehr lange nicht gelöscht, dann wird die Anzahl der

Regeln unüberschaubar und die Performanz des Filters verringert sich wegen der vielen Regeln, die er für ein Datenpaket anwenden muss.

Neben der dynamischen Paketfilterung gibt es noch eine andere Möglichkeit, die Fähigkeiten eines einfachen Paketfilters zu erweitern: Das Überprüfen von Protokollformaten. Dabei muss das Datenfeld der Pakete, die Packet Data Unit (PDU), untersucht werden, um an die Protokollinformationen der höheren Protokollschichten (Protocol Control Information) zu gelangen. Dies ist besonders dann wirkungsvoll, wenn es mit der dynamischen Filterung kombiniert wird. Ein guter Filter mit Protokollprüffunktion kann Regeln umsetzen, die etwa alle File Transfer Verbindungen verbieten, bei denen der Username unbekannt ist. Ein solcher Filter muss viele Informationen über die höher liegenden Protokollschichten haben und kann in der Regel nur einige wenige Protokolle erkennen. Dabei handelt es sich dann um die bekanntesten Protokolle wie beispielsweise HTTP. Aber selbst wenn man sich beim Bau des Filters auf wenige bekannte Protokolle beschränkt, wird die in der Firewall abgelegte Datenmenge sehr groß.

Alle Paketfilter haben gemeinsam, dass die Pakete anhand bestimmter Regeln gefiltert werden, die in der Firewall abgelegt sind. Diese müssen vom Benutzer der Firewall konfiguriert werden (eventuell werden später von der Firewall automatisch weitere Regeln hinzugefügt, wie etwa beim dynamischen Filtern).

Kommen wir nun zu der Frage, wie ein Paketfilter eigentlich konfiguriert wird. Die grundlegende Aufgabe besteht darin, die eigene Sicherheitspolitik festzulegen und diese dann in Regeln umzusetzen. Das Problem dabei ist, dass die Regeln sich lediglich auf die Pakete beziehen, während die Sicherheitspolitik solche Details in der Regel nicht enthält. Deshalb müssen die Sicherheitskonzepte zunächst auf die Paketebene heruntergebrochen werden, was nicht selten recht problematisch ist. Wenn wie im obigen Beispiel alle FTP Verbindungen mit unbekanntem Usernamen in der Sicherheitspolitik verboten sind, dann muss dieses Konzept in Paketfilterregeln umformuliert werden. Die erste Entscheidung, die getroffen werden sollte, ist die zwischen generellem Verbot und genereller Erlaubnis (siehe 2.3.3). Im Normalfall wird das generelle Verbot verwendet, da der Benutzer hier lediglich die benötigten Verbindungstypen festlegen und in den Regeln verankern muss. Der Administrator definiert einfach eine (in der Regel recht kleine) Menge von sicheren oder benötigten Verbindungen, die dann von der Firewall erlaubt werden. Bei Bedarf kann dieser Katalog im Nachhinein noch erweitert werden, wofür einfach einige Regeln im Katalog ergänzt werden.

Wenn die Regeln festgelegt werden, muss auf jeden Fall bedacht werden, dass Protokolle meist bidirektional ablaufen. Wenn eine Firewall-Konfiguration den File Transfer regelt, sollte

der Administrator sowohl Regeln für Verbindungen nach außen als auch für Verbindungen nach innen festlegen. Dieser Grundsatz gilt sowohl für Erlaubnisregeln wie für Verbotsregeln. Auch wenn eine Firewall das Prinzip der generellen Erlaubnis praktiziert, müssen Regeln für beide Kommunikationsrichtungen festgelegt werden, da einige Attacken nur unidirektionalen Verkehr erfordern und somit nicht unterbunden werden, wenn lediglich der Verkehr in die andere Richtung geblockt wird.

Eine weitere grundsätzliche Frage, die bei der Konfiguration geklärt werden muss, besteht darin, ob beim Blocken eines Paketes irgendeine Form der Rückmeldung erfolgen soll. Eine solche Rückmeldung kann ein Fehlercode an denjenigen sein, der die unerlaubte Verbindung aufbauen wollte. Die Fehlercodes können nützlich sein, wenn ein User irrtümlich eine unerlaubte Verbindung aufbauen wollte. Aufgrund der Rückmeldung wird er dies vermutlich nicht mehr versuchen und die Last der Firewall senken. Wenn der Fehlercode jedoch einen potentiellen Angreifer erreicht, kann dieser den Code nutzen, um die Regeln der Firewall systematisch auszutesten, wodurch er die verwundbaren Stellen ausfindig machen kann. Wenn zudem für jedes geblockte Paket ein Fehlercode erzeugt wird, kann der Angreifer Denial-of-Service-Attacken gegen die Firewall starten, indem er große Mengen unerlaubter Pakete schickt. Die Firewall generiert dann nur noch Fehlercodes, wodurch sie in ihrer eigentlichen Arbeit stark behindert wird.

Wie sollte nun den Regelsatz der Firewall an sich behandelt werden? Die Regeln für den Filter sind hochsensible Daten, denn auf ihnen basiert ja der ganze Filtermechanismus. Es muss nicht nur ihre korrekte Arbeitsweise und ihre Konsistenz gewährleistet sein, sondern die Regeln selbst können auch das Ziel von Angriffen werden. Wenn die Firewall gerade aktiv ist, ist es oft schwierig, neue Regeln hinzuzufügen oder bestehende zu löschen, da der Benutzer den Einfluss dieser Aktionen auf bestehende Verbindungen nicht kennt. Deshalb sollten die Regeln nur offline editiert, und der neue Regelsatz sollte als ganzes in die Firewall eingespielt werden. Wenn die Regeln editiert wurden, sollte der neue Regelsatz den alten überschreiben. Wenn am Regelsatz in der Firewall Änderungen direkt (also während des Betriebes) vorgenommen werden, können vor allem bei dynamischem Filtern Komplikationen auftreten. Deshalb sollte der aktuellen Regelsatz immer offline zur Verfügung stehen. Dieser Regelsatz ist frei von dynamisch erzeugten Regeln und enthält somit keine verwirrenden Einträge, die zu Störungen mit den neuen Regeln führen könnten. Ein eingespielter Satz von Regeln muss zudem vor unerlaubten Änderungen geschützt werden. Ein Angreifer wird immer versuchen, die Filterregeln der Firewall in die Hand zu bekommen, weil er das Filtersystem dann einfacher umgehen kann. Deshalb müssen die Regeln vor Angreifern über das Netz geschützt werden. Die einfachste Lösung wäre natürlich, Netzzugriff auf die Regeln gar nicht zu ermöglichen. Wenn dies geschieht, können neue Regelsätze aber nicht

mehr über das Netz eingespielt werden, was die Aufgaben des Administrators komplizierter macht. Neben dem Schutz vor unerlaubtem Zugriff müssen auch die Regeln selbst möglichst frei von Schwachstellen sein.

Ein beispielhaftes Problem für die Regelkonfiguration ist dabei, ob nach IP-Adresse oder Hostname gefiltert werden soll. Will man zum Beispiel Zugriffe auf eine bestimmte Website verhindern, kann man die IP-Adresse sperren. Bei den Anbietern illegaler Seiten ist es aber nicht unüblich, ihre IP-Adresse von Zeit zu Zeit zu ändern und diese Änderung dem DNS bekannt zu geben. So kann die Seite wieder besucht werden, indem der Hostname im Browser eingegeben wird. Filtert man aber nach dem Hostnamen, so kann es sein, dass die Anfrage an einen externen DNS-Server manipuliert wird und somit die Anfrage an einen vertrauten Host dem Angreifer eine Verbindung zu seinem Host verschafft.

Die vorangegangenen Abschnitte haben gezeigt, dass der Aufbau eines korrekt konfigurierten Filtersystems eine schwierige Aufgabe ist. Da es so viele unterschiedliche Paketfilter mit unterschiedlichen Leistungsmerkmalen und Vor- und Nachteilen gibt, ist es oft eine schwere Wahl, welchen der vielen Paketfilter man für seine Firewall am Besten einsetzt. Wenn ein Netzbetreiber sich entschieden hat, wartet noch die große Aufgabe der Konfiguration auf ihn. Und letztlich kann er sich auch nie sicher sein, ob er alles richtig gemacht hat. Der nächste Teil dieses Kapitels stellt Kriterien vor, nach denen der richtige Paketfilter ausgewählt werden kann, der für die gegebene Situation angemessen ist.

Performanz

Das Kriterium für einen Paketfilter, das oft an erster Stelle genannt wird, ist die Performanz. Mit Performanz ist die Geschwindigkeit gemeint, mit der die Firewall die Pakete filtert und dann entweder weiterleitet oder blockt. Vom Hersteller wird diese Geschwindigkeit oft in einer Pakete/Sekunde Rate angegeben, was jedoch zu Missverständnissen führen kann. Zwei Pakete können unterschiedlich groß sein, und wenn zwei Firewalls die gleiche Pakete/Sekunde Rate haben, können diesem Wert zwei vollkommen unterschiedliche Datenraten der Einheit Bit/Sekunde zugrunde liegen, je nachdem, welche durchschnittliche Paketgröße der Hersteller bei der Pakete/Sekunde Angabe angenommen hat. Wenn die tatsächliche Datenrate relevant ist, muss diese angenommene Durchschnittsgröße auf jeden Fall bekannt sein. Die Performanz der Firewall ist außerdem nicht immer konstant. Sie hängt in hohem Maße von der Komplexität der eingespielten Regeln ab. So hat ein dynamischer Paketfilter eine höhere Performanz, wenn er nicht viele dynamische Regeln erzeugt hat. Sind dagegen bereits viele Pakete protokolliert und ist somit der Systemzustand komplex, sinkt die Performanz. Zudem hängt die Performanz vom Rechner ab, auf dem die Firewall installiert wurde. Vor allem der Speicher des Rechners hat

hohen Einfluss auf die Geschwindigkeit. Hingegen ist die Prozessorgeschwindigkeit in der Regel nicht für die Performanz ausschlaggebend.

Inwieweit die Performanz wirklich eine Rolle spielt hängt auch von der vorhandenen Netzverbindung ab. Wenn das Netz ohnehin nur eine geringe Datenrate hat, braucht die Firewall auch keine hohe Performanz zu haben. Wird hingegen ein Hochgeschwindigkeitsnetz betrieben, so kann eine langsame Firewall zum Nadelöhr für den Verkehr werden.

Einsatzrechner

Soll für die Firewall ein eigener Rechner eingesetzt werden, der lediglich die Filter- und Routing-Funktionalität hat? Oder kann vielleicht auf einen gewöhnlichen Rechner, der noch andere Aufgaben hat, der Paketfilter aufgesetzt werden? Die Antwort dieser Fragen hängt vom Netzbetrieb ab. Wenn für den Paketfilter ein eigener Rechner zur Verfügung steht, ist dieser natürlich wesentlich leistungsfähiger als wenn er sich den Rechner mit anderen Programmen teilen muss. Wenn eine große Anzahl von Netzen mit vielen unterschiedlichen Protokollen über die Firewall verbunden werden soll, so kann auf die Geschwindigkeit eines einzelnen Firewall-Rechners sicher nicht verzichtet werden. Wenn man hingegen nur eine einfache Internetverbindung filtern möchte, kann der Filter auch auf einen gewöhnlichen Arbeitsrechner aufgesetzt werden, der eventuell nicht einmal neu angeschafft werden muss.

Einfache Regelspezifikation

Das Konfigurieren der Firewall ist eine schwierige Sache. Damit dieser Vorgang nicht noch komplizierter wird, als er ohnehin schon ist, sollte aufgrund der Sicherheitsrelevanz die Benutzerfreundlichkeit des Filters hoch sein, also die Möglichkeit bieten, die Regeln auf der richtigen Abstraktionsebene zu formulieren. Die Pakete sollen vom Filter nicht einfach als Bitfolge betrachtet werden, ohne Rücksicht auf irgendwelche Protokollformate. Niemand möchte Regeln der Form formulieren wie „Leite ein Paket nur dann weiter, wenn die Bits 12 und 34 gesetzt sind“. Ein solcher Filter hätte ganz bestimmt eine viel zu niedrige Abstraktionsebene. Die Abstraktionsebene sollte aber auch nicht zu hoch angesetzt sein. Manche Paketfilter verdecken so viele Details, dass beispielsweise Regeln, die sich auf bestimmte Ports beziehen, nicht formuliert werden können. Solche Regeln werden aber gebraucht, um zum Beispiel die Standardports bekannter Angriffssoftware zu überwachen.

Flexible Regeln

Die Regeln müssen so formuliert werden können, dass sie nicht nur Informationen aus dem Paket-Header (und zwar *alle* diese Informationen), sondern auch sog. Metainformationen berücksichtigen. Zu den Metainformationen gehören alle Informationen, die nicht direkt aus dem Paket

gewonnen werden. Die Eingangsleitung (Interface), auf der das Paket den Filter erreicht hat, ist beispielsweise eine Metainformation. Zu den wichtigen Informationen aus dem Paket-Header gehören Start- und Zieladresse, Optionen, Protokollinformationen (TCP, UDP...) sowie die Flags. Wenn solche elementaren Informationen nicht in den Regeln abgefragt werden können, entstehen Sicherheitslücken, die nicht vom Filter abgedeckt werden können.

Reihenfolge der Regeln

Für den Administrator der Firewall muss vorhersagbar sein, in welcher Reihenfolge der Filter die Regeln auf ein Paket anwendet. Im einfachsten (und häufigsten) Fall ist dies lediglich die textuelle Reihenfolge, in der die Regeln eingegeben wurden. Manche Filter ändern diese Reihenfolge aber automatisch, um so eine höhere Effizienz zu erzielen. Dieser Vorgang macht die Anwendungsreihenfolge aber undurchschaubar, was zu einigen Problemen führt:

- Wenn die textuelle Reihenfolge geändert wird, kann nicht vorhergesagt werden, wie die Regeln endgültig angeordnet werden.
- Der Algorithmus, der die Regeln umsortiert, kann fehlerhaft sein.
- Das Umsortieren kann die Semantik eines Regelsatzes verändern, so dass anders gefiltert wird, als in der Sicherheitspolitik festgelegt wurde.

Es gibt zwei Auswege aus dieser Misere. Entweder verzichtet der Administrator darauf, dass der Filter die Reihenfolge der Regeln automatisch ändert, oder er führt Metaregeln ein, in denen die Ausführungsreihenfolge der Regeln festgelegt wird.

Regeln für ankommende und verlassende Pakete

Manche Paketfilter erlauben Filterregeln nur für Pakete, die den Filterrechner verlassen, nicht aber für solche, die bei ihm ankommen. Dies führt zu drei Problemfeldern.

Zunächst kann der Firewall-Rechner selbst attackiert werden, wenn an ihn gerichtete Pakete nicht gefiltert werden. Wenn auf dem Rechner angreifbare Dienste laufen, kann diese Schwachstelle von Angreifern ausgenutzt werden: Die Firewall steht dann „außerhalb“ ihrer eigenen Regeln.

Der zweite Punkt ist, dass bestimmte Adressfälschungen von einem solchen Filter nicht erkannt werden. Meist geht ein Adressfälscher so vor, er von außen ein Paket schickt, dass eine Quelladresse innerhalb des eigenen Netzes hat. Solche Pakete werden von Filtern abgefangen, wenn sie von außen in den Filter eindringen, denn ein Paket von außen kann normalerweise keine Quelladresse innerhalb des Netzes haben (Metainformationen, s.o.). Wenn der Filter aber nur abgehende Pakete prüft kann er ohne Auswertung von Metainformationen nicht feststellen,

ob das Paket tatsächlich von „innen“ kam oder ob eine Adressfälschung vorliegt. Das dritte Problem tritt dann auf, wenn mehr als zwei Netze über den Filter verbunden werden sollen. Wenn beispielsweise drei Netze (A, B und C) über eine Firewall verbunden werden sollen und der Filter nur abgehende Pakete ohne Auswertung von Metainformationen prüft, so kann er bei Paketen, die ins Netz A geleitet werden sollen nicht feststellen, ob diese aus Netz B oder Netz C stammen.

Log Funktionen

Paketfilter besitzen meist umfangreiche Log-Funktionen. Dabei kann das Log sowohl geblockte Pakete protokollieren als auch ausgewählte Pakete, die nicht geblockt wurden. Ersteres ist notwendig, um Informationen darüber zu gewinnen, ob die Netzanwender eventuell Verstöße gegen de Sicherheitspolitik unternommen haben oder ob das Netz angegriffen wurde. Das Protokollieren von legalen Paketen kann aus zweierlei Gründen wichtig sein. Einerseits hilft es bei der Fehlerkorrektur in der Konfigurationsphase. Andererseits können nicht abgewehrte Angriffe besser analysiert werden, um geeignete Gegenmaßnahmen auszuarbeiten. Ohne eine solche Protokollierung würde der erfolgreiche Angriff überhaupt nicht aufgezeichnet, und der Netzbetreiber könnte die Angriffsstrategie nicht nachvollziehen.

Wichtig ist ebenfalls die Komplexität der Daten, die protokolliert werden. Von größter Bedeutung sind Angaben darüber, welches Paket und welche Regel den Protokolleintrag verursacht haben. Ebenfalls müssen nähere Informationen über das Paket in den Informationen enthalten sein. Vor allem Start- und Zieladresse sowie der verwendete Protokolltyp (beispielsweise TCP) müssen aufgezeichnet werden. Wünschenswert sind auch Angaben über die angesprochenen CEPs. In Einzelfällen kann sogar die Protokollierung des ganzen Paketes sinnvoll sein.

Das System zum Auditing ist häufig flexibel. Das Audit-Trail kann nicht nur auf dem Bildschirm angezeigt, sondern auch in einer geeigneten Log Datei abgelegt werden. Eventuell sollten diese Log Dateien automatisch nach bestimmten Kriterien sortiert werden, wie etwa der Paketrichtung. Viele Auditing-Systeme verfügen zudem über eine optionale Funktion, mit der die Log Dateien automatisch per E-Mail an den Administrator weitergeleitet werden können. Die Log Funktion ist dabei konfigurierbar. Manche Angreifer produzieren einfach einen Datenstrom, der die Log Datei anschwellen lässt, bis die Festplatte voll ist. Deshalb hat die Log Datei eine maximale Größe, die einstellbar ist. Wenn diese Größe überschritten wird, werden die alten Log Einträge automatisch gelöscht. Die richtige Größe für die Datei zu finden ist eine schwierige und wichtige Aufgabe. Wenn sie zu groß ist, belegt sie wertvollen Platz auf der Platte. Wenn sie zu klein ist, werden Angriffe eventuell nicht vollständig dokumentiert.

Gute Testbarkeit

Natürlich sollte ein Paketfilter sich einfach testen lassen, denn die Firewall ist ein wichtiger Baustein des Netzes und muss einwandfrei funktionieren, doch werden vom Hersteller nur selten Testszenarien mitgeliefert. Beim Testen müssen zwei Fragen vom Administrator geklärt werden. Einerseits muss sichergestellt werden, dass der Paketfilter gemäß der zugrunde liegenden Sicherheitspolitik korrekt konfiguriert wurde. Andererseits muss gewährleistet sein, dass der Filter sich auch gemäß seiner Konfiguration verhält. Die aus diesen beiden Fragen resultierenden Tests können die korrekte Arbeitsweise zwar nicht formal beweisen, liefern aber einen Indiz dafür. Ob Indizien für die Sicherheitspolitik einer Organisation ausreichen, ist von den dortigen Verantwortlichen zu prüfen.

Obwohl viele Hersteller von Paketfiltern keine Testumgebungen bereitstellen, in denen der Filter auf diese beiden Fragen hin abgeklopft werden kann, gibt es immerhin für einige Produkte Paketgeneratoren, mit denen Testpakete erzeugt werden können. Mit diesen Paketgeneratoren ist aber noch längst kein wirklich ausreichendes Testen möglich, weshalb die Korrektheit des Filters und der Konfiguration in der Regel nicht nachgewiesen werden können.

Die bisherigen Ausführungen haben die Vor- und Nachteile von Paketfiltern sowie die Kriterien zu ihrer Bewertung offen gelegt. Zum Abschluss des Kapitels wollen wir untersuchen, wo der Einsatz von Paketfiltern sinnvoll ist und wie man diese am Besten in der gegebenen Netzarchitektur einsetzt. Zudem werden die wichtigsten Regeln aufgeführt, die der Regelsatz des Paketfilters immer enthalten sollte.

Paketfilter sind erweiterte Router. Deshalb kann Paketfilterung überall dort eingesetzt werden, wo auch Router einsetzen können. In einem busbasierten Netz, das über einen einzelnen Router mit dem Internet verbunden ist, kann Paketfilterung beispielsweise nur auf diesem Router eingesetzt werden. In einem größeren LAN, das über viele Router verfügt, kann prinzipiell auf jedem dieser Router auch ein Paketfilter aufgesetzt werden. Wichtig ist vor allem, dass die „Single Point of Access“ Strategie eingehalten wird. Wenn Paketfilterung zum Schutz eines Netzes eingesetzt wird, dann sollte jedes Paket von außen an mindestens einem Paketfilter untersucht werden. Dies wird am einfachsten gewährleistet, indem der gesamte von außen ankommende Netzverkehr über einen einzelnen Router mit Paketfilter geleitet wird (Single Point of Access). Weitere Paketfilter innerhalb des eigenen Netzes können ebenfalls sinnvoll sein. Beispielsweise kann dadurch erreicht werden, dass ein Server, der nur bestimmte Dienste anbietet, auch nur Pakete erhält, die für diese Dienste notwendig sind. Paketfilter können zudem kaskadiert eingesetzt werden, um das Netz in Bereiche mit unterschiedlichen Sicherheitsniveaus einzurichten.

So wird beispielsweise ein Rechner mit hochsensiblen Daten auf ein höheres Sicherheitsniveau gestellt als die „normalen“ Rechner im Netz.

Der übermäßige Einsatz von Paketfiltern führt aber auch leicht zu erheblichen Einbußen in der Performanz. Dies kommt vor allem daher, dass die Geschwindigkeitsanforderungen innerhalb des eigenen Netzes in der Regel höher sind als die für die Verbindung zum Internet. Innerhalb eines Firmennetzes wird beispielsweise sehr viel an Tagewerk über das Netz abgewickelt, was aber mit der Verbindung zum Internet nichts zu tun hat. Ebenso können zu viele Filter eine Abschottung einzelner Netzbereiche oder sogar einzelner Rechner verursachen, die gar nicht geplant war. Je mehr Paketfilter innerhalb des Netzes eingesetzt werden, umso größer wird der Verwaltungsaufwand. Die Regeln für die einzelnen Paketfilter müssen konsistent bleiben, nicht nur innerhalb eines Filters, sondern auch zwischen verschiedenen Filtern. Die Konsistenz der Regelsätze ist vor allem dann bedroht, wenn ein neuer Regelsatz eingespielt wird, während Betrieb im Netz herrscht. Während die neuen Regeln bereits auf einigen Filtern laufen, gelten auf anderen Filtern eventuell noch die alten, wodurch der reibungslose Ablauf des Netzverkehrs gefährdet werden kann.

Welche Regeln sollte der Regelsatz nun enthalten? Diese Frage muss der Netzbetreiber weitgehend selbst entscheiden, denn nur er weiß, welche Arten von Netzverkehr er gemäß seiner Sicherheitspolitik zulassen will. Es gibt jedoch einige elementare Regeln, die der Regelsatz des Paketfilters auf jeden Fall enthalten soll. Diese werden im Folgenden erläutert.

Zunächst sollte man, wie bereits erwähnt, das Prinzip des generellen Verbotes seinen Regeln zugrunde legen, damit nicht eventuell eine Art unerwünschten Netzverkehrs übersehen werden kann. Dies erreicht man dadurch, dass man in einem Regelsystem mit textueller Abarbeitungsreihenfolge zunächst alle Erlaubnisregeln formuliert und ans Ende des Regelsatzes eine generelle Verbotsregel. Für jedes Paket werden nun zunächst alle Erlaubnisregeln geprüft. Wenn eine dieser Regeln auf das Paket anwendbar ist, erfüllt es eine Spezifikation für zuverlässigen Netzverkehr und wird weitergeleitet. Ist das nicht der Fall, so greift am Ende der Prüfung die generelle Verbotsregel, deren Erkennungsmuster auf alle Pakete passt. Somit werden alle Pakete, denen nicht zuvor durch eine Erlaubnisregel das Passieren gestattet wurde, durch diese Generalregel aussortiert. Einige Firewalls haben dieses Prinzip von vorne herein implementiert. Dort sind nur die Erlaubnisregeln zu erstellen.

Ferner sollte von außen kommender Verkehr mit einer Quelladresse innerhalb des Netzes immer abgeblockt werden, da es sich dabei mit hoher Wahrscheinlichkeit entweder um Adressfälschung (Spoofing) oder einfach um ein Produkt fehlerhafter Netzkonfiguration handelt. Das

gleiche gilt für Netzverkehr aus dem inneren Netz heraus, der eine Quelladresse außerhalb des eigenen Netzes hat. Ebenso sollten alle Pakete abgeblockt werden, die eine nicht zulässige oder nicht nachvollziehbare (also unbekannte) Quelladresse haben, egal in welche Richtung das Paket gesendet wird. Dies gilt erst Recht, wenn es sich bei dem Paket um eine Broadcast-Sendung handelt. Wenn bei einem Paket einige der Protokollparameter benutzt werden, sollte es ebenfalls geblockt werden, da die Parameter in der Regel nicht benutzt werden und auch nicht benutzt werden müssen. Um eine bessere Untersuchung der Pakete zu ermöglichen, sollten fragmentierte Pakete zur Filterung wieder zu ganzen Paketen zusammengesetzt werden.

Paketfilter sind die einfachste Möglichkeit, eine Firewall zu realisieren. Es gibt höhere Formen, die in der Regel mit Paketfiltern kombiniert werden. Das nächste Kapitel beschäftigt sich mit höheren Firewalls, bei denen die Filterung auf der Anwendungsschicht passiert.

3.2.2 APPLICATION-LEVEL-GATEWAY

Während die Paketfilter den Datenverkehr lediglich bis zur OSI-Transportschicht kontrollieren, ist es für bestimmte Anwendungsprotokolle wichtig, sie feiner zu filtern. Diese Möglichkeit bieten die Application-Level Gateways.

Was ist ein Application-Level Gateway?

Bei der Beschreibung der Filterung auf Anwendungsschicht, ist die Begriffsbildung in der Literatur nicht einheitlich. Manche Autoren sehen die Funktionalität der Application-Level Gateways in einem Spezialfall, den Proxies, enthalten. Dieses sind auf dem Gateway laufende Prozesse, die lediglich Vermittlungsfunktionalität bieten. Sie werden im folgenden Abschnitt beschrieben (s. 3.2.3).

Um ein Protokoll genauer zu betrachten, muss es an einem Kopplungspunkt zwischen den zu trennenden Netzen verarbeitet werden. Dieses ist die Aufgabe des Application-Level Gateways. Bei dieser Verarbeitung ist die Filterung der Daten möglich.

Ein reines Application-Level Gateway muss Serverfunktionalitäten bereitstellen. Dazu enthält es „auf die Applikation spezialisierten Code, der bereits eine Vorverarbeitung durchführt oder einen Teildienst der Anwendung erbringt“ [Mück00]. Hat das Gateway zum Beispiel die Aufgabe den E-Mail-Verkehr zu filtern, muss es die E-Mail von einem anderen Server entgegennehmen. Der Client aus dem internen Netz verbindet sich beim Einsatz eines Application-Level Gateway nie mit dem externen Server, sondern nutzt nur den Gateway-Rechner als Mailserver.

3.2.3 PROXIES

Was ist ein Proxy?

Im Gegensatz zum Application-Level Gateway ist ein Proxy, zu Deutsch Vermittler, kein vollständiger Server. Es ist meist nur ein Prozess auf der Anwendungsschicht, der Verbindungen zwischen Client und Server unter Betrachtung des Sicherheitskonzeptes weiterleitet. Somit unterliegt die Verbindung seiner Kontrolle.

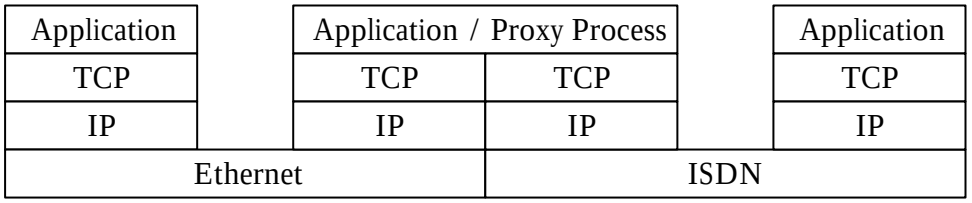


Abb. 4: Gateway und Proxy auf Anwendungsschicht (nach [LM 01])

Um das Prinzip des Single Point of Access zu wahren, muss der Proxy-Prozess auf einem Gateway-Rechner laufen. Da sowohl das Application-Level Gateway als auch der Proxy auf Anwendungsschicht arbeiten (siehe Abb. 4), kommt es meist zur Vermischung der beiden Begriffe.

Das eben beschrieben Prinzip des Proxy wird Application-Level Proxy bezeichnet. In der Literatur wird meist auch noch der Begriff Circuit-Level Proxy erwähnt. Dieser leitet ohne Betrachtung des Anwendungsprotokolls die Daten an den Server weiter. Da solch ein Proxy vom Protokoll unabhängig arbeitet, wird er meist als generischer Proxy implementiert. Seine Arbeitsweise ist äquivalent zu der des Paketfilters und soll hier nicht weiter betrachtet werden. Im Folgenden wird der Begriff Proxy synonym zum Application-Level Proxy verwendet.

Arbeitsweise eines Proxy

Der Client verbindet sich zuerst mit dem Proxy, der sich daraufhin an Stelle des Clients mit dem realen Server verbindet. Dazu muss der Client auf den Proxy abgestimmt sein. Die Software muss wissen, wie sie den Proxy anstatt des wirklichen Servers anspricht. Dafür gibt es modifizierte Versionen der Anwendungs- bzw. Betriebssystemsoftware. Zusätzlich gibt es auch andere Wege: Einige Programme wie zum Beispiel der Internet Explorer lassen sich von Haus aus an die Proxies anpassen. Eine andere Lösung sind generische Proxies wie zum Beispiel SOCKS. Dieses Programm fängt Verbindungsanfragen ab und leitet sie unter Einhaltung der Sicherheitspolitik an die Clients weiter.

Die Verbindung über einen Proxy zum realen Server nennt man transparent, wenn der Benutzer der Verbindung zwischen Client und Server nichts von dem Proxy bemerkt. Die Fire-

wall leitet hierbei die Verbindung über den Proxy um. Dabei fällt die Einschränkung, dass die Clientsoftware auf den Proxy abgestimmt sein muss.

Zusätzliche Funktionsweisen von Proxies

Mit der Unterbrechung der Verbindung durch den Proxy sind nun auch genauere Einschränkungen möglich. Der Paketfilter unterscheidet nur nach Absenderadresse und nach CEP. Zusätzlich bietet ein Proxy folgende Leistungen:

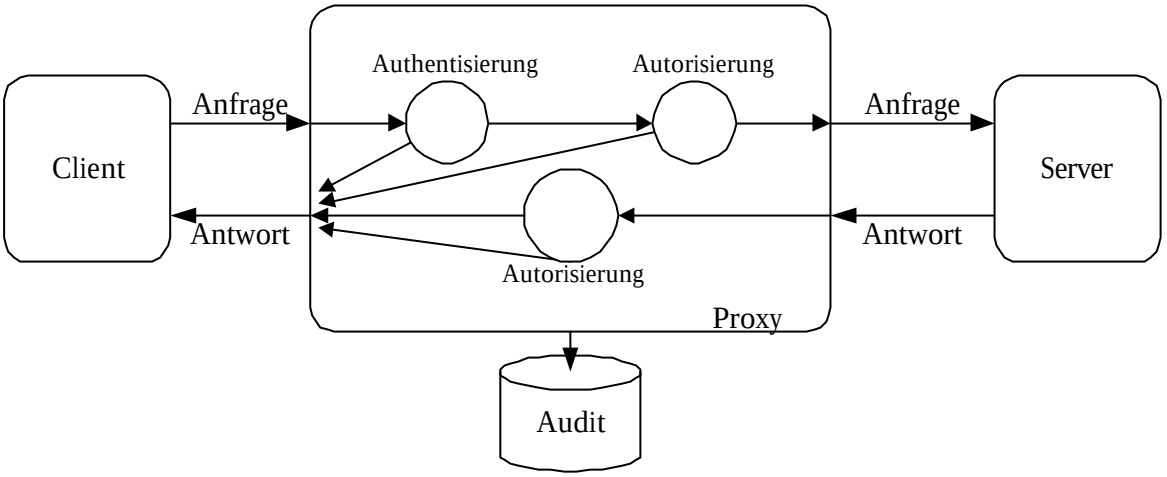


Abb. 5: Zugriffskontrolle durch einen Proxy in Anlehnung an [Mück 00]

- Authentiserung und Autorisierung

  Durch die Einschränkung eines kompletten Dienstes durch den Paketfilter sind meist alle Benutzer eines Hosts von einem Dienst ausgeschlossen. Es könnte aber nützlich sein, einigen von ihnen den Zugriff zu erlauben. Dieses lässt sich mit der Zugriffskontrolle, bestehend aus Authentisierung und Autorisierung, durchführen (Abb. 5). Durch die Authentisierung, also den Vorgang der Verifizierung der Identität eines Benutzers, lässt sich die Nutzung eines Dienstes von dem Benutzer abhängig machen. Ist ein Benutzer auch autorisiert, so darf er den Dienst nutzen.

- Content Filtering

  Da der Proxy auf Anwendungsschicht arbeitet (siehe Abb. 4), werden die Pakete zu Einheiten des Anwendungsschichtprotokolls zusammengesetzt. Dabei ist es nun möglich, sie nach Inhalt zu kontrollieren. So lässt sich der Zugriff auf bestimmte URLs sperren, die E-Mail nach bestimmten Schlagwörtern durchsuchen, der Java-Anteil eines HTTP-Datenstroms herausfiltern oder ein Virenscan durchführen. Eine nützliche Anwendung dieser Eigenschaft ist das Filtern nach personenbezogenen Daten, die eine Organisation unbedingt vor Diebstahl schützen will. Jeglicher Datenstrom nach draußen kann vom Proxy geblockt werden.

- Caching Funktionalität

  Neben den Schutzmaßnahmen kann der Proxy gleichzeitig zur Performanzsteigerung benutzt werden. Der Proxy speichert eine Kopie des Datenstroms. Werden die gespeicherten Daten häufig genug abgerufen, so steigt die Geschwindigkeit und die Netzauslastung sinkt erheblich. Diese Funktion macht vor allem bei HTTP-Proxies Sinn, da dieselbe Webseite mit höherer Wahrscheinlichkeit noch einmal angefordert wird, als zum Beispiel eine E-Mail.

  Allerdings steigt die Anforderung an die Hardwareausstattung des Rechners, auf dem der Proxy läuft. Die gepufferte Information benötigt Plattenplatz und die aufwendigere Funktionsweise des Proxy mehr Rechenleistung.

- Erweiterte Logginginformation

  Da ein Proxy die Protokolle, für die er zuständig ist, genauer kennt, kann er den Datenverkehr nicht nur genauer filtern, sondern ihn auch detaillierter in einer Audit-Aufzeichnung vermerken (so genanntes Loggen). So kann er statt jeder HTTP-Verbindung zum Beispiel nur HTTP-Verbindungen zu unbekannten URLs loggen. Das Log wird dadurch übersichtlicher und somit lesbarer.

- Fehlerhafte Pakete werden korrigiert

  Die ankommenden Pakete der niederen Schichten werden auf dem Proxy zu Einheiten der Anwendungsschicht übersetzt. Ist der Proxy bis hierhin immun gegen fehlerhafte Pakete, so sind auch alle Rechner hinter dem Proxy geschützt. Denn die für den weiteren Versand nötigen Pakete werden von dem Gateway-Rechner gebildet. Der Vorteil hierbei ist, dass die meist aufwendigere Verarbeitung und Korrektur von fehlerhaften Paketen nur vom Proxy durchgeführt werden muss. Die Clientrechner bekommen dann die Pakete von dem Proxy, dem sie vertrauen.

Betrachtung der Proxies in Hinblick auf die Topologie der Firewall

Der Einsatz eines Proxy ist nur dann sinnvoll, wenn man ihn in Kombination mit Maßnahmen, die den Verkehr beschränken, einsetzt. Installiert man zum Beispiel auf einem Router einen HTTP-Proxy, so kann der Angreifer immer noch das interne Netzwerk über andere Protokolle kompromittieren.

Eine Art des sinnvollen Einsatzes eines Proxy ist der dual-homed Host. Dies ist ein Rechner mit zwei Netzwerkkarten, der Pakete nicht von selbst weiterleitet. Diese Weiterleitung übernimmt der jeweilige Proxy, der auf dem Rechner installiert ist, sofern die Daten mit der Sicherheitspolitik vereinbar sind. Der Vorteil ist, dass hier bei Absturz der Proxy-Prozesse das

interne Netz nicht erreichbar ist. Die Firewall lässt sich also nicht durch deaktivieren der Proxies umgehen. Dazu ergibt sich bei dieser Konfiguration ein Nachteil: es können nur diese Dienste ins interne Netz gelangen, die auch durch einen Proxy betrachtet werden. Nicht vermittelbare Dienste wie ICMP können so nicht weitergeleitet werden.

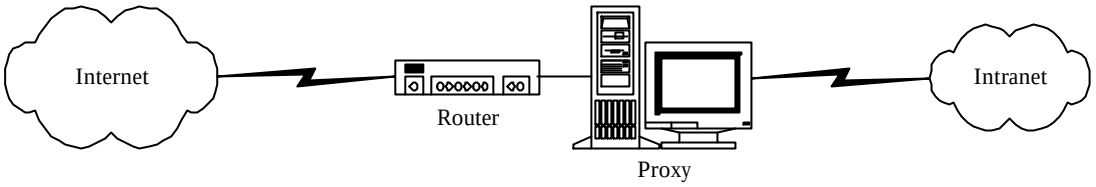


Abb. 6: Kombination von Proxy und Router

Dieser Nachteil kann überwunden werden, indem das Konzept der Proxies mit der Paketfilterung vereint wird (Abb. 6). Soweit ein Dienst gefiltert werden kann, wird dieser durch den Proxy betrachtet und weitergeleitet.

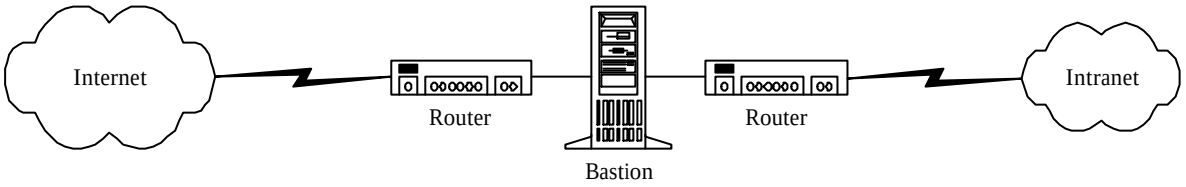


Abb. 7: Proxy als Bastion Host

Eine bessere Lösung ist, den Proxy auf einen Bastion Host einzusetzen. Ein Bastion Host ist ein Rechner, der überwunden werden muss, um in das interne Netz zu gelangen. Dazu befindet sich der Proxy-Rechner zwischen zwei Routern (Abb. 7). Dadurch ist immer noch ein Schutz vor dem internen Netz vorhanden, selbst wenn der Proxy bereits von Angreifern eingenommen werden konnte.

Die Einsatzmöglichkeiten des Proxy in Verbindung mit den Firewall-Topologien werden im nächsten Abschnitt genauer dargestellt.

3.3 EINSATZGEBIET

Nachdem wir in den vorangegangenen Kapiteln die verschiedenen Arbeitsweisen von Firewalls dargestellt haben, wollen wir nun auf die Einsatzgebiete eingehen. Zunächst einmal ist es wichtig zu wissen, warum ein Netz oder auch ein einzelner Rechner überhaupt durch eine Firewall (in den Abbildungen durch eine Mauer dargestellt) geschützt werden muss. Den Hauptgrund nennt die Anleitung zur Firewall Gauntlet der Firma Network Associates Technology, Inc.

[NAI 99]: Die Firewall gewährleistet den *sicheren* Zugang zu einem privaten Netz (oder einem privaten Rechner) und sichert gleichzeitig die Möglichkeit, dieses auf sicherem Weg mit anderen Netzen zu verbinden.

Die Firewall bietet dabei allen angeschlossenen Seiten oder auch für nur eine Seite Schutz. Es stellt sich natürlich die Frage, warum zur Gewährleistung dieses sicheren Zugangs eine gesonderte Software benötigt wird. Auf diese Frage gibt es zwei Antworten. Zum ersten sind die gängigen Rechner und Netzwerkprotokolle nach inhärent unsicheren Konzepten aufgebaut. Die heutigen Netzwerke haben eine unüberschaubare Menge von Sicherheitslücken. Da diese sich nicht pauschal und global beseitigen lassen, muss die aufgesetzte Firewall die Sicherheitslücken so gut wie möglich ausfüllen.

Der zweite Grund besteht darin, dass nicht jeder Betreiber eines privaten Netzes die gleichen Sicherheitsanforderungen hat. Jeder Betreiber hat im Regelfall seine eigene Sicherheitspolitik und somit sein eigenes Niveau an Sicherheitsanforderungen. Die Firewall ist aufgrund ihrer Konfigurierbarkeit bestens dazu geeignet, fast jede beliebige Sicherheitspolitik umzusetzen. Allerdings muss sich der Benutzer[2] der Firewall natürlich vorher darüber im Klaren sein, wie seine Sicherheitspolitik aussehen soll. Denn die Flexibilität birgt gleichzeitig die Gefahr, dass auch eine fehlerhaft erstellte Sicherheitspolitik wortgetreu umgesetzt wird. Selbst wenn die Sicherheitspolitik korrekt in eine Firewall-Konfiguration umgesetzt wurde, schützt die Firewall natürlich nicht vor einer anfänglichen Fehlspezifikation der Sicherheitspolitik. Vor dem Einsatz einer Firewall muss der Netzbetreiber daher eine genaue Anforderungsanalyse durchführen. Dabei sollte er nicht so vorgehen, dass er nach möglichen Gefahren sucht und gefährlichen Netzverkehr in seiner Sicherheitspolitik verbietet. Stattdessen sollte er analysieren, welchen Netzverkehr er für seine Zwecke benötigt und lediglich diesen ausgewählten Netzverkehr auf mögliche Schwachstellen untersuchen. Diese Vorgehensweise geht mit dem Konfigurationsprinzip des generellen Verbotes Hand in Hand, und eine so erstellte Sicherheitspolitik sollte sich vergleichsweise einfach in Firewallregeln umsetzen lassen. Während der Anforderungsanalyse muss genau darauf geachtet werden, dass die Sicherheitspolitik konsistent zum erwünschten Netzbetrieb bleibt. Wenn beispielsweise für eine Aufgabe ein bestimmter Typ von Netzverkehr benötig wird, der aber im Sinne der Sicherheitspolitik nicht zulässig ist, so entsteht eine Inkonsistenz, da die Aufgabe mit dieser Sicherheitspolitik nicht erledigt werden kann.

[2]Mit Benutzer sind im Folgenden Einzelanwender, Gruppen, Abteilungen und ganze Organisationen gemeint.

Ist die Sicherheitspolitik erarbeitet worden, muss festgelegt werden, in welcher Art und Weise die Firewall eingesetzt werden soll. Im Folgenden wird nun dargestellt, wie eine Firewall eingesetzt und in die Netzarchitektur eingebettet werden kann.

3.3.1 DIE PERSÖNLICHE FIREWALL

Im aller einfachsten Fall besteht das zu schützende „Netz“ lediglich aus einem einzelnen Rechner. Dieser Fall liegt vor allem dann vor, wenn ein Privatuser seinen Heimrechner mit Internetanschluss gegen mögliche Angreifer sichern will. Für diesen Zweck gibt es Firewalls, die direkt auf dem zu schützenden Rechner aufgesetzt werden. Diese Firewalls sind in der Regel einfacher aufgebaut als solche, die ein ganzes Netz schützen sollen. Das liegt daran, dass das interne „Netz“ ja nur aus einem, dem Firewall-Rechner selber, besteht und folglich keine inneren Interfaces verwaltet werden müssen. Viele Firewallhersteller bieten für ihre Firewall von vorn herein zwei getrennte Versionen an, von denen eine für den Schutz eines einzelnen Rechners (persönliche Firewall) und die andere zusätzlich für den Schutz eines Netzes gedacht ist.

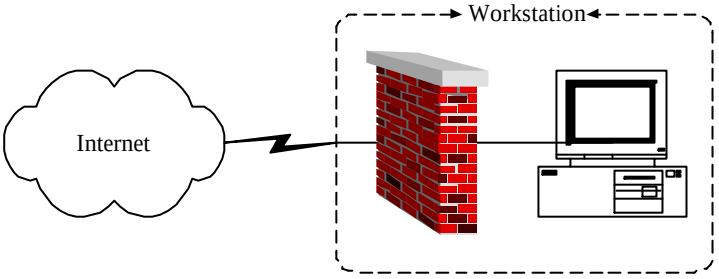


Abb. 8: Persönliche Firewall

Eine persönliche Firewall ist oft nicht so leistungsfähig wie ihre entsprechende Netzvariante. Das kommt daher, dass die persönliche Firewall sich ihren Rechner mit allen anderen Anwendungen teilen muss. Da der Betrieb des Rechners durch den Einsatz der Firewall nicht gestört werden soll, werden an eine persönliche Firewall hohe Performanzanforderungen gestellt. Sie darf den Rechner nicht unzumutbar verlangsamen und sollte einen relativ bescheidenen Speicherbedarf haben. Das macht zum Beispiel den Einsatz von dynamischen Paketfiltern als persönliche Firewall höchst schwierig.

3.3.2 SINGLE POINT OF ACCESS ZU EINEM NETZ, BASTIONEN

Der klassische Anwendungsfall für eine Firewall ist der Schutz eines Netzes vor den Gefahren aus anderen Netzen. Dabei wird die Firewall auf den Verbindungsrechner (Gateway) zwischen den beiden Netzen aufgesetzt. Der gesamte Verkehr, der zwischen den beiden Netzen stattfindet, wird dann über diesen Rechner geleitet. Die Firewall bildet den einzigen Zugangs-

punkt zum geschützten Netz. Damit besitzt diese Firewallkonfiguration einen Common Point of Trust. Bei dieser Konfiguration mit nur einem Zugangspunkt besteht ein Spezialfall dieses Prinzips, den Single Point of Access.

Die Durchsetzung dieses Prinzips ist von elementarer Wichtigkeit, denn wenn es noch einen zweiten Zugang zum geschützten Netz gibt, wird der Verkehr über diesen Zugang nicht gefiltert und der Schutz durch die Firewall wird hinfällig. Neben der Schutzaufgabe dient die Firewall häufig noch dem Logging des Netzverkehrs, also als Auditing System. Manchmal verbindet eine Firewall nicht nur zwei, sondern drei oder noch mehr Netze untereinander. Dies macht im Prinzip keinen Unterschied im Konzept der Einsatzweise, solange der Firewall-Rechner für jedes Netz der einzige Zugangspunkt zu den anderen Netzen ist. Das Konzept des einzigen Zugangspunktes macht die Firewall ihrerseits aber zu einem lohnenden Ziel für Angreifer, denn wenn der Firewall-Rechner ausfällt, ist das gesamte Netz von der Außenwelt abgeschnitten oder schlimmstenfalls dem externen Netz ohne Schutz ausgeliefert.

Sollte es aus Gründen der Lastverteilung und der Hochverfügbarkeit notwendig sein, zwischen zwei Netzen mehrere Zugänge zu haben, so muss nicht nur auf jedem dieser Zugänge die *gleiche* Firewall installiert sein, sondern sie muss auch überall die gleiche Konfiguration haben. Damit bleibt das Prinzip des Common Point of Trust bestehen. Andernfalls entstehen wiederum Inkonsistenzen in der Umsetzung der Sicherheitspolitik, wenn über einen Zugang Netzverkehr möglich ist, der über einen anderen Zugang nicht möglich wäre.

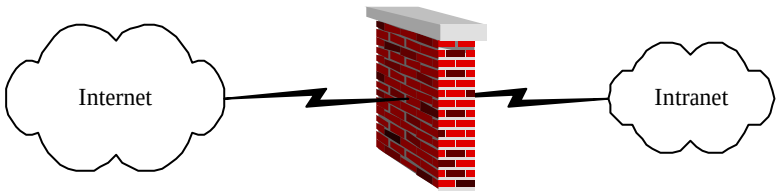


Abb. 9: Single Point of Access

Mitunter wird der Single Point of Access nicht durch eine einzelne Firewall realisiert, sondern durch mehrere gekoppelte Firewalls. Dies ist sinnvoll, wenn zwei verschiedene Produkte aufgrund ihres Regelangebots nicht ausreichen, um die Sicherheitspolitik umzusetzen oder wenn aus Sicherheitsgründen verschiedene Firewalls verwendet werden sollen. Damit ist dann auch nach Bekannt werden einer Sicherheitslücke der Schutz durch die zweite Firewall gewährleistet. Wenn ein Datenpaket mehrere Firewalls passieren muss, werden die Regeln der einzelnen Firewalls mit einem logischen Und verknüpft, wodurch der Regelraum der Firewall erweitert werden kann. Dieses entspricht dem Prinzip des Diversity of Defense.

Wenn auf diese Weise mehrere Firewall gestaffelt werden, um das Single Point of Access Prinzip umzusetzen, spricht man von einer Bastion. Der Unterschied zu einer Demilitarisierten Zone mit Kaskadentechnik (s.u.) besteht darin, dass bei der Bastion die Firewalls direkt hintereinander geschaltet sind. Es befindet sich kein anderen Netzteilnehmer dazwischen wie etwa ein Webserver.

3.3.3 DEMILITARISIERTE ZONEN

Nicht immer muss für ein gesamtes Netz eine einheitliche Sicherheitspolitik gelten. Manchmal sind einige Teile des Netzes sensibler als andere, oder einige Rechner erfordern Netzverkehr, den andere Rechner nicht nutzen dürfen sollen. Wenn eine Firma beispielsweise einen Webserver besitzt, der sowohl mit dem Firmennetz als auch mit dem Internet verbunden ist, so gelten für den Webserver vielleicht geringere Zugriffsbeschränkungen als für das sensible Netz. Um solche Situationen bedienen zu können, werden mit Hilfe von Firewalls Netzregionen mit verschiedenen Sicherheitsniveaus geschaffen. Meist handelt es sich dabei wie in unserem Beispiel um zwei Niveaus. Um diese Niveaus zu unterscheiden, werden im Normalfall zwei kaskadierte Firewalls eingesetzt. Zwischen der Außenwelt und den Bereich mit dem geringeren Niveau (in unserem Beispiel also zwischen Internet und Webserver) befindet sich eine Firewall, welche die Sicherheitspolitik für das geringere Niveau durchsetzt. Beim Übergang zum höheren Niveau befindet sich dann eine zweite Firewall, von der die strengere Sicherheitspolitik durchgesetzt wird. Der Bereich mit dem geringeren Niveau wird oft als demilitarisierte Zone bezeichnet.

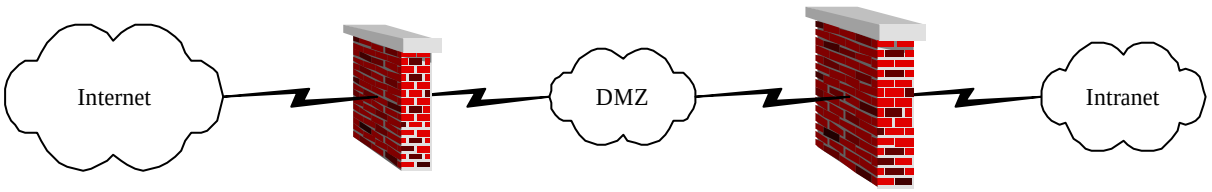


Abb. 10: Demilitarisierte Zone durch eine Kaskade

Die Kaskadierung ist eine einfache und leicht zu konfigurierende Architektur. Sie funktioniert aber nur, wenn der Verkehr, den die innere Firewall zulässt, eine Teilmenge des Verkehrs ist, den die äußere Firewall akzeptiert. Verkehr zwischen Außenwelt und dem Bereich auf dem hohen Niveau muss beide Firewalls passieren, und deshalb muss dieser Verkehr von beiden Firewalls durchgelassen werden.

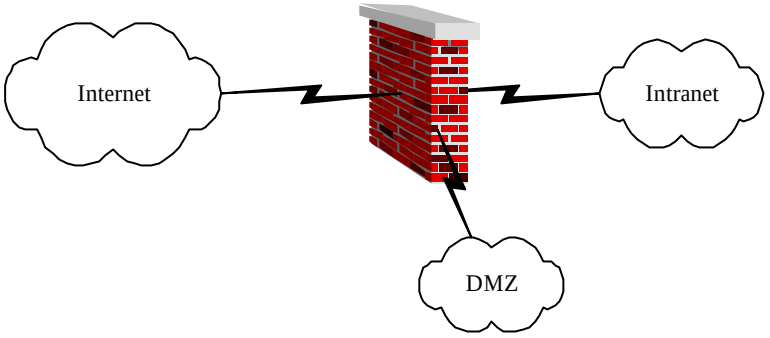


Abb. 11: Demilitarisierte Zone durch eine Weiche

Wenn die Sicherheitspolitiken solche Regelsätze nicht zulassen, kann keine Kaskadierung eingesetzt werden. In diesem Fall wird die demilitarisierte Zone über eine einzelne Firewall realisiert, die dann als Weiche zwischen der Außenwelt und beiden Niveaus fungiert. In dieser Firewall müssen dann drei getrennte Regelsätze vorhanden sein:

- ein Regelsatz für den Netzverkehr zwischen Außenwelt und dem Bereich auf dem geringen Sicherheitsniveau
- ein Regelsatz für den Netzverkehr zwischen Außenwelt und dem Bereich auf dem hohen Sicherheitsniveau
- ein Regelsatz für den Netzverkehr zwischen den beiden Niveaus.

Eine solche Weiche ist wesentlich schwieriger zu konfigurieren als die Kaskade, bietet dafür aber mehr Flexibilität.

3.3.4 FIREWALLS IM INTERNEN NETZ

Firewalls werden nicht für die Kontaktstellen zwischen zwei Netzen eingesetzt, sondern auch innerhalb eines Netzes zwischen verschiedenen Segmenten. Dabei handelt es sich um eine Erweiterung des Prinzips der demilitarisierten Zone, nur dass wesentlich mehr Sicherheitsniveaus vorhanden sind. Das Netz kann mit Firewalls in viele kleine Bereiche aufgeteilt werden, für die alle eigene Sicherheitspolitiken gelten. Der Hauptzweck einer solchen Unterteilung liegt aber nicht in der Erstellung vieler unterschiedlicher Niveaus, sondern auch der Eindämmung eines Angriffes. Wenn ein Angriff die äußere Firewall durchbrochen hat besteht die Gefahr, dass sich der Schädling im gesamten Netz ausbreitet. Angriffe dieser Art wären zum Beispiel Würmer, die sich an alle Rechner des Netzes versenden und dort Schaden anrichten. Durch weitere Firewalls im internen Netz kann dieser Befall eventuell auf einen kleinen Bereich des Netzes eingedämmt werden, wodurch sich der Schaden recht gut begrenzen lässt.

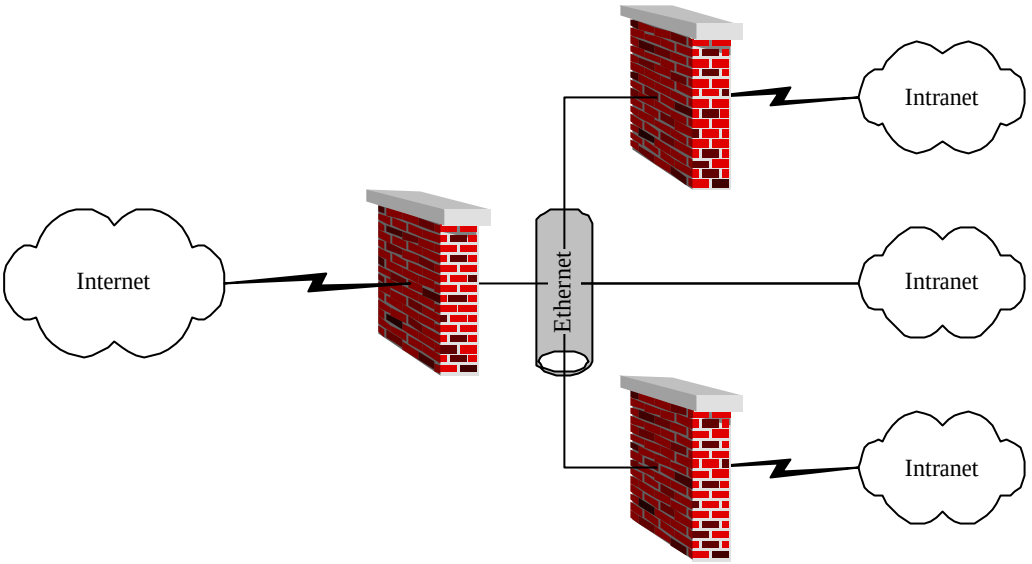


Abb. 12: Durch interne Firewalls aufgeteiltes Netz

Es ist für Netzbetreiber oft eine schwierige Entscheidung, ob sie Firewalls im internen Netz einsetzen wollen. Bei ihrem Einsatz muss jeder Netzverkehr (auch interner) mehrere Firewalls passieren, was zu großen Performanzeinbußen führen kann. Zudem ist der Konfigurationsaufwand sehr hoch, weil alle Firewalls mit untereinander konsistenten Regeln konfiguriert werden müssen.

3.3.5 FLUTTORE

Ein Problem bei der Bekämpfung der Gefahren aus öffentlichen Netzwerken besteht bei Angriffen, die auf Flooding-Techniken basieren. Sie können nicht nur die Leistung der Systeme im internen Netz beeinträchtigen, sondern auch die Performanz des gesamten Netzverkehrs. Dadurch kann bei solchen Angriffen die Leistung des Choke-Points Firewall so weit eingeschränkt werden, dass nicht mehr ausreichend Performance für den geschäftskritischen Verkehr zur Verfügung steht.

Eine Lösung für dieses Problem sind so genannte Fluttore. Dabei wird die Firewall um flexible Quality-of-Service-Kontrollen erweitert. So kann einer für den Geschäftsablauf wichtigen VPN-Verbindung zu einem Partner eine höhere Priorität gegenüber sonstiger Internetverbindungen zugewiesen werden. Reicht die Durchsatzleistung der Firewall für diese VPN-Verbindung nicht mehr aus, so kann der Durchsatz der öffentlichen Internetverbindung, die zum Beispiel wegen eines Flooding Angriffs viel Leistung in Anspruch nimmt, zu Gunsten der VPN-Verbindung eingeschränkt werden. Dies sind die wichtigsten Architekturen, in denen Firewalls zum Einsatz gebracht werden.

Dieser Abschnitt hat gezeigt, dass eine Firewall sehr vielseitig verwendet werden kann. Dennoch reicht eine Firewall alleine oft nicht aus, um genügend Schutz zu gewährleisten. Der folgende Abschnitt befasst sich mit den Problematiken, die beim Einsatz von Firewalls auftreten können, vor allem wenn Firewalls alleine eingesetzt werden.

3.4 PROBLEMATIKEN

Eine Firewall kann ein Netz gegen Gefahren sichern. Doch dies ist kein Allheilmittel, denn es gibt immer noch Möglichkeiten, das zu sichernde Netz zu schädigen. So erfolgen laut den Polizeibehörden von Großbritannien (Scotland Yard), USA (FBI) und der Deutschland (BKA) 75-80% der Hackerangriffe aus dem eigenen internen Netz, also durch Mitarbeiter.

„Eine Firewall kann zwar einen Netzübergang sichern, sie hat aber keinen Einfluss auf die Sicherheit der Kommunikation innerhalb der Netze.“ [BSI 99 M2.70] Eine Firewall sichert das zu schützende Netzsegment nur gegen Gefahren von außen. Dies ist auch leicht verständlich, da die Firewall als Single Point of Access als Übergang zwischen zwei Netzen mit unterschiedlichen Sicherheitsanforderungen dient. Gefahren von Innen müssen mit anderen Maßnahmen bekämpft werden und können zum Teil mit Intrusion Detection Systemen, wie sie im folgenden Kapitel beschrieben werden, analysiert werden.

Damit sei aber nicht gesagt, dass eine Firewall nicht gegen Angriffe aus dem internen, Organisationseigenen Netz schützt. Durch die weitere Unterteilung der Netzwetzwerkes in verschiedene Segmente mit unterschiedlichen Sicherheitsniveaus, können Angriffe im internen Netz eingedämmt werden. Innerhalb eines Solchen Segmentes kann aber durch eine Firewall nichts gegen die Angriffe getan werden. Hier benötigt man das Vertrauen in die anderen Teilnehmer.

Eine Firewall schützt auch nicht generell vor Denial-of-Service-Attacken. Ein Grund ist, dass die Anbindung an das zu sichernde Netz beim Provider verwundbar ist. Wird dort die Verbindung gekappt, so kann keine Konfiguration der Firewall des schutzbedürftigen Netzes den Ausfall der Anbindung an das Internet verhindern. Ein anderer Grund ist die Sicherung der Endsysteme. Eine geeignete Befehlsfolge eines per Paketfilter freigegebenen Dienstes kann bei einem Endsystem immer noch zum Ausfall führen.

Trotz Firewall lässt auch ein generelles Verbot eines Dienstes aufheben, sofern andere Dienste durchgeleitet werden. Über das Protokoll dieses freigegebenen Dienstes ist es möglich, ein anderes Protokoll zu tunneln. Dazu kann man sogar eine Firewall gegen sich selbst einsetzten.

Das Proxy-Paket SOCKS, das als Bestandteil einer Firewall eingesetzt werden kann, lässt sich dazu missbrauchen. So kann zum Beispiel ein Dienst auf Port 4000 benutzt werden, obwohl der Port durch die Firewall gesperrt ist, sofern zum Beispiel HTTP in der Firewall freigegeben ist. Mit SOCKS2HTTP kann dieser Dienst auf das HTTP-Protokoll umleiten. Dazu benötigt man im internen Netz einen Dekonverter, der den Port 4000 auf dem Endsystem freigibt. Um diesen zu installieren, benötigt man administrative Rechte, die ein normaler Benutzer eigenlicht nicht haben sollte. Doch bei der nachlässigen Wahl der Administrator-Passwörter ist dies nicht schwer. Somit ist die Firewall umgangen und damit zumindest teilweise unnütz.

Auch herrscht die Meinung, dass eine Firewall das Netz vollständig vor dem Befall von Malware schützen kann. Hierbei handelt es sich um einen erstzunehmenden Aberglauben. Denn auch eine Firewall muss sich selbst auf eine Anti-Virus-Software verlassen. Diese muss immer auf den neusten Stand gebracht werden.

Zudem bringt die Filterung von Viren, etc., einen enormen Aufwand mit sich. Denn die Bytefolge, anhand derer die Anti-Virus-Software die Malware erkennen kann, ist nicht immer in nur einem Paket enthalten. Die Firewall muss daher nicht nur die zu einer Applikation gehörenden Datenpakete komplett zwischenspeichern, sie muss auch noch entscheiden, welche Pakete zusammen gehören, um sie dann komplett scannen zu lassen. Dies geht natürlich nur auf Applikationsebene.

Ein externes Netz, wie zum Beispiel das Internet, bringt immer neue Bedrohungen für das interne Netz mit sich. Ein großer Teil der Nutzer verfügt über die ausreichende Kenntnis, um andauernd neue Sicherheitslücken zu finden, die zu neuen Attacken führen können. Eine Firewall kann nicht vollständig gegen alle Bedrohungen schützen. Zwar können durch ein generelles Verbot einige Gefahren ausgeschlossen werden. Doch finden sich in den erlaubten Diensten und in der Firewall selbst immer neue Lücken in Bezug auf die Sicherheit. Daher ist eine ständige Administration und Pflege der Firewall-Konfiguration nötig.

Damit stellt sich ein weiteres Problem: Wie soll die Administration vonstatten gehen? Dazu ist administrativer Zugang nötig. Die Administration darf nur von innen heraus erfolgen, um eine mögliche Schädigung der Firewall von außen zu vermeiden. Der Zugang kann entweder direkt erfolgen, indem sich die Hardware in einem separaten, physikalisch stark gesicherten Raum befindet, zu dem nur das Administrationspersonal Zugang hat. Oder es bietet sich die Möglichkeit eines Remote-Zugangs zu den Rechnern. Diese müssten über eine ebenfalls physikalisch stark geschützte separate Leitung mit den Kontrollrechnern verbunden sein, um so die Schädigung aus dem internen Netz mit anschließendem Ausfall der Firewall auszuschließen.

Der Ausfall der Firewall ist ein weiterer Problemfaktor. Für den Fall, dass diese ausfällt, darf das darunter liegende Betriebssystem keine Gateway-Funktionalitat bieten. In diesem Fall darf der Rechner also kein ankommendes Paket in das interne Netz leiten. Dies ist aber bei einigen Rechnerkonfigurationen der Fall gewesen, ein Beispiel ist der Microsoft Proxy Server. Als Abhilfe tauschen zumindest einige Windows NT Firewalls den kompletten Microsoft TCP/IP-Stack durch ihren eigenen aus.

Eine andere Art von Problematik ergibt sich aus der Tatsache, dass es verschiedene Arten von Firewalls gibt. Man muss feststellen, welches Konzept zum Einsatz in der entsprechenden Umgebung passt. Dazu müssen die Sicherheitsziele geklärt werden. Dies sind nach [BSI00] Punkt M2.70 u. a.:

- Schutz des internen Netzes gegen unbefugten Zugriff von außen
- Schutz der Firewall gegen Angriffe aus dem externen und Manipulation aus dem internen Netz
- Schutz der lokal übertragenen und gespeicherten Daten
- Schutz der lokalen Netzkomponenten
- Verfügbarkeit externer Information im internen Netz

Zudem wird als Grund gegen den Einsatz von Firewalls die Performanz genannt: Eine Firewall betrachtet den gesamten Netzverkehr nach außen, der durch den Single Point of Access fließt. Dies benötigt Zeit und verlangsamt somit den Verkehr zwischen den Netzen. Dies senkt somit die Netzperformanz.

Festzuhalten ist also: Eine Firewall löst nicht alle Sicherheitsprobleme. Es schützt nur in geringem Maße. Trotz Firewall können Einbrecher in das interne Netz gelangen und geheime Daten nach draußen schaffen, sei es durch Fehlkonfigurationen oder Lücken im Produkt selbst. Auch das größte Problem, die Schädigung von innen durch Mitarbeiter, wird von der Firewall nicht gelöst.

Dazu ist es nötig, weitere Schutzmaßnahmen einzuleiten, die Schädigungen des Netzes erkennen und verbannen sollen. Dies kann durch Analyse der Netz- und Hostdaten geschehen, was die Aufgabe einer weiteren Sicherheitsinstanz ist: Das Intrusion Detection System.

4 INTRUSION DETECTION SYSTEME

Wie im Kapitel 3 gezeigt wurde, gibt es einige Problematiken beim Schutz eines Systems nur durch Firewalls. Deshalb wurden ergänzend Intrusion Detection Systeme (IDS) entwickelt. Sie können so eingesetzt werden, dass sie zusammen mit einer Firewall ein System oder Netzwerk sicherer vor Angriffen schützen.

4.1 DIE EINSATZGEBIETE VON IDS

Ein IDS kann an verschiedenen Stellen eines Netzes eingesetzt werden, um den Datenverkehr auf Angriffe zu überprüfen. Es findet aber auch Verwendung zur Überwachung eines einzelnen Systems oder Hosts auf dem sich besonders sicherheitsbedürftige Daten befinden.

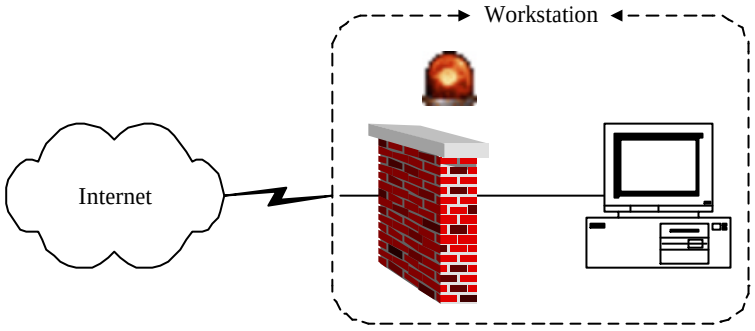


Abb. 13: Einzelner Host mit Firewall und IDS

Da die meisten größeren aber auch viele kleine Unternehmen und andere Organisationen Netzwerkarchitekturen für ihre Arbeit nutzen, ist vor allem der Einsatz von Intrusion Detection Systemen zum Schutz dieser Netzwerke sinnvoll. Sie sind das Hauptangriffsziel der meisten Hacker. Wo in dem Netz das IDS eingesetzt wird, ist abhängig von der Netzwerkarchitektur. Das IDS kann als eigenständige Komponente oder zusammen mit einer Firewall verwendet werden. Wird keine Firewall benutzt, kann das IDS zwar die Angriffe erkennen, aber nicht viel dagegen tun und so schnell selbst Opfer eines Angriffes werden. Es benachrichtigt bestenfalls noch den System Security Officer (SSO) und dieser kann je nach Sicherheitspolitik daraufhin Gegenmaßnahmen ergreifen, wie zum Beispiel für den Angriff wichtige Ports schließen, um so den Angreifer auszusperren. Dazu muss der Security Officer schnell reagieren. Da aber der Angreifer bereits eingedrungen ist, besteht nur noch die Möglichkeit, den Schaden zu begrenzen und gegebenenfalls für die spätere Auswertung zu dokumentieren.

Eine Möglichkeit für die Position eines IDS ist, dass das IDS direkt auf der Firewall aufsetzt und so alle ankommenden Pakete auf mögliche Angriffsmerkmale untersucht. Die Pakete haben dann bereits die Firewall passiert. Besteht ein Verdacht auf einen Angriff veranlasst das IDS die

Firewall den Port zu schließen, durch den die Pakete gekommen sind, und beendet so den Angriff, bevor weiterer Schaden entsteht.

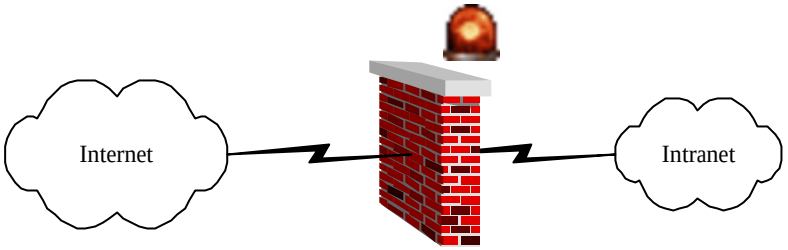


Abb. 14: Intranet geschützt durch Firewall mit aufgesetztem IDS

Eine andere Position für das IDS ist vor der Firewall, so dass Datenpakete aus fremden Netzen wie zum Beispiel dem gesamten Internet untersucht werden können, bevor sie die Firewall erreichen. Verdächtige Pakete können so gleich abgeblockt werden. Der SSO bekommt so auch einen Eindruck wie interessant „sein“ Netz für Hacker ist. Dieses Wissen ist wichtig für die Weiterentwicklung der Sicherheitspolitik durch die entsprechenden Mitarbeiter und Vorgesetzte.

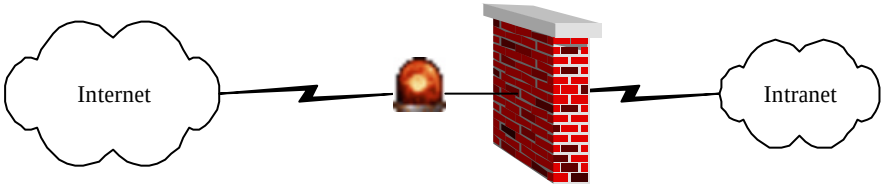


Abb. 15: Intranet geschützt durch Firewall und davor befindliches IDS

Auch ein IDS hinter der Firewall ist sinnvoll. Wenn ein Angreifer durch die Firewall gekommen ist, kann das IDS das Eindringen erkennen und dann Gegenmaßnahmen anregen. Bei größeren Netzen mit mehreren Subnetzen kann auch die Installation von mehreren IDS in verschiedenen Subnetzen zu einer Erhöhung der Sicherheit beitragen, denn in einem großen Netz leidet die Performanz, wenn nur ein einzelnes IDS den gesamten Datenverkehr überwacht. Ein besonders zu schützendes Subnetz kann durch ein IDS und zusätzlich durch eine Firewall geschützt werden. Was besonders wichtig ist, wenn der Angreifer in einem der anderen Subnetze sitzt.

In solchen Situationen, in denen Angreifer vom eigenen Netz aus aktiv werden, sind sie zwar „legitime“ Benutzer, die aber ihre Rechte im Netz überschreiten. Ursachen für dieses Verhalten gehen von Langeweile über Wut auf den Arbeitgeber bis zur gezielten Datenspionage. Laut Lutz Peichert von META Group Deutschland GmbH werden fast 80% der Angriffe von internen Netzbenutzern verübt.

In solchen großen Netzen ist auch die Überwachung von einzelnen Systemen durch ein IDS denkbar, wenn es besonders schützenswert ist. Dazu muss am einzigen Eingang zu dem gesamten Netzwerk eine Firewall stehen, die erkannte Angreifer von externen Netzen bei einem

Angriff auf diesen einzelnen Rechner ausschließt. Zur Überwachung des gesamten Netzwerkes werden die Alarmmeldungen der einzelnen IDS an eine zentralen Stelle gemeldet, von wo aus der SSO einen Gesamtüberblick hat und auf Angriffe entsprechend reagieren kann. Bei Angriffen von internen Benutzern kann der SSO, wenn er selbst keine Administratorrechte hat, den Systemadministrator benachrichtigen. Der Administrator wird daraufhin versuchen den Angreifer eindeutig zu identifizieren, um dessen Rechte soweit einzuschränken, so dass dieser keinen weiteren Schaden mehr anrichten kann. Natürlich wird der identifizierte interne Angreifer außerdem mit arbeitsrechtlichen und/oder strafrechtlichen Konsequenzen zu rechnen haben. Als Beweismittel hierzu dienen die Aufzeichnungen des IDS über den Angriff. Die Identifizierung von Angreifern von draußen ist weitaus schwerer und wird in Abschnitt 4.5. Intrusion Response behandelt.

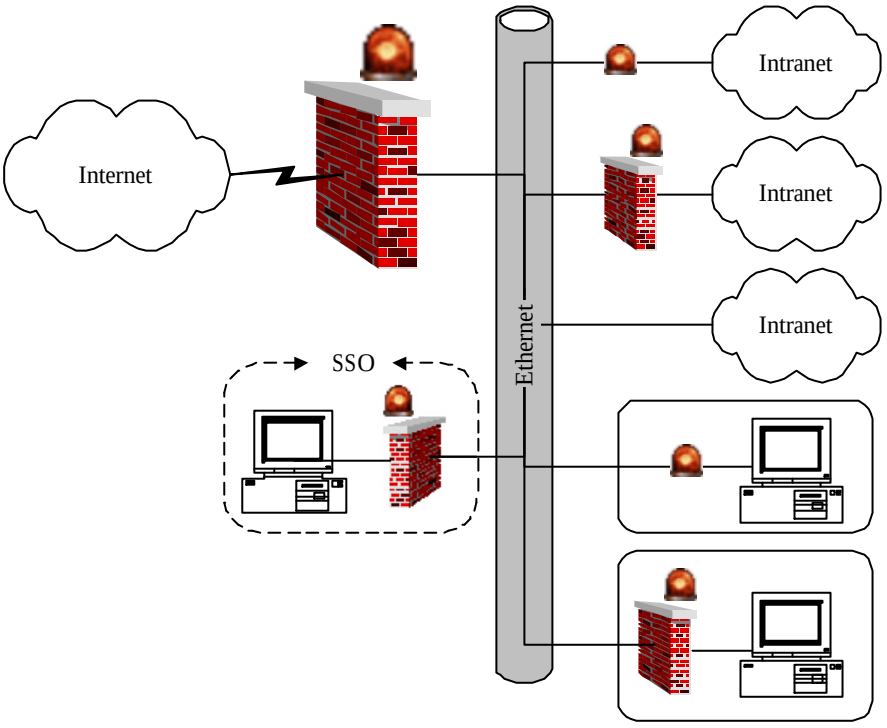


Abb. 16: Intranet aus Subnetzen und einzelnen Rechnern mit unterschiedlicher Sicherheitsanforderung

Zudem sind die meisten Mitarbeiter in einer Organisation ein Sicherheitsrisiko, weil sie für die Verbreitung der meisten Malware verantwortlich sind. Oft liegt dies einfach nur am mangelnden Sicherheitsbewusstsein der Mitarbeiter, weswegen sich die meisten Viren, Würmer und Trojaner in Organisationen ausbreiten. Es gibt auch Fälle, wo Malware gezielt von unzufriedenen Mitarbeitern eingesetzt wird. Deshalb werden IDS auch zur Überwachung der Integrität der Daten eingesetzt, um zum Beispiel mit Hilfe von Checksummenverfahren gerade eingebrachte Malware schnell zu erkennen und zu beseitigen. Natürlich werden hierfür auch Virenscanner eingesetzt. Diese stellen aber nicht fest, ob ein Angreifer Dateien manipuliert oder

gar gelöscht hat, wenn der Angreifer keine Malware einsetzt, sondern mit Hilfe seiner Benutzerrechte Schaden anrichtet.

Zum Schutz von den im Kapitel 3 eingeführten demilitarisierten Zonen (DMZ) können ebenfalls IDS eingesetzt werden. Wenn die Firewall als Weiche zwischen dem zu schützenden Netz und der dazugehörigen DMZ dient, kann das IDS auf der Firewall aufsetzen und von dort aus sowohl das Netz als auch die DMZ überwachen.

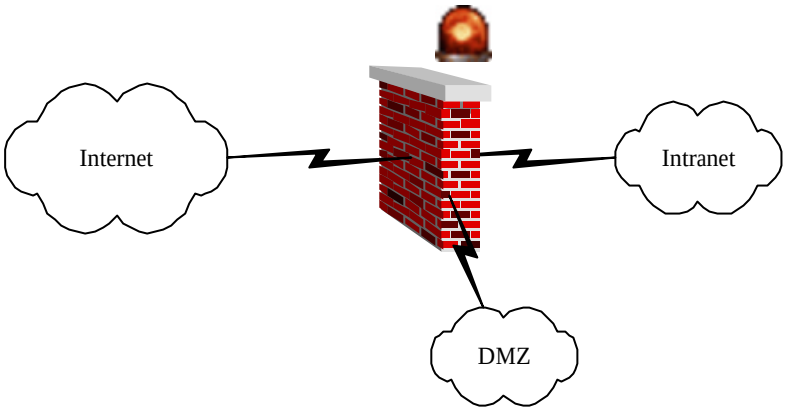


Abb. 17: Intranet und DMZ geschützt durch Firewall und IDS.

Wenn eine DMZ dadurch gebildet wird, dass sich zwei Firewalls hintereinander befinden (Kaskade), dazwischen die DMZ und hinter der zweiten Firewall das Intranet ist, dann kann das IDS sich auf dieser zweiten Firewall befinden. Von hieraus kann es Angriffe entdecken, die durch die erste Firewall gekommen sind und das Intranet zum Ziel haben. Gleichzeitig können Attacken aus dem Intranet heraus erkannt werden.

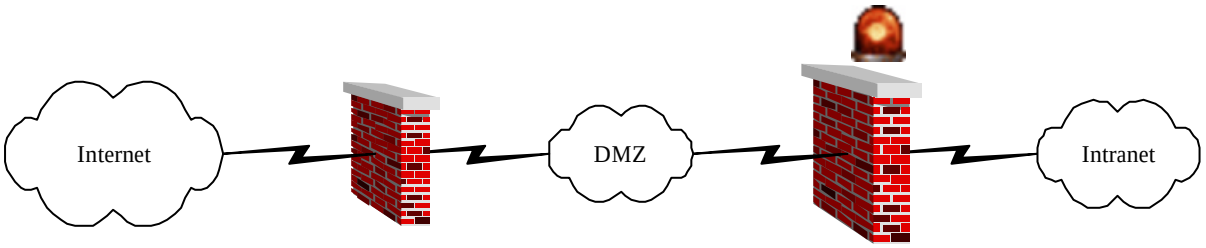


Abb. 18: Intranet geschützt durch Firewall-Kaskade und ein IDS

Wie gezeigt wurde, kann ein IDS an verschiedenen Positionen und zu unterschiedlichen Zwecken benutzt werden. Wo nun genau ein IDS eingesetzt wird, hängt von der Sicherheitspolitik der Organisation ab.

4.2 WAS SIND INTRUSION DETECTION SYSTEME?

Intrusion Detection Systeme (kurz: IDS, übersetzt: einbruchsentdeckende Systeme) dienen der Erkennung von Angriffen auf ein Computersystem oder ein Netzwerk. Sie beobachten den Datenverkehr und beurteilen anhand von Regeln, ob gerade ein Angriff stattfindet („real time IDS“ nach [Brunnstein 01]) oder ob ein Angriff stattgefunden hat. Im ersten Fall meldet ein IDS die Angriffe, damit schnell Gegenmaßnahmen getroffen werden können. Im zweiten Fall kann der System Security Officer (SSO) im Nachhinein den Angriff analysieren.

Das IDS kann als einzelne Komponente, im Verbund oder in Zusammenarbeit mit einer Firewall eingesetzt werden (nach [Brunnstein 01]). Das IDS dient also dem System oder Netzwerk als Alarmanlage. Es überwacht nicht nur einzelne Rechner und den Netzverkehr, sondern auch bestimmte sensible Teile eines Rechners, wie zum Beispiel dem Dateisystem, oder eines Netzwerkes, wie zum Beispiel den Rechner des SSO. Zusammen mit einer Firewall und/oder SSO bildet es ein Sicherheitssystem für einen Rechner oder ein Netz.

Für IDS gibt es zwei verschiedene Vorgehensweisen, um Angriffe zu erkennen. Sie können die Benutzeraktivitäten mit Benutzerprofilen (anomaly detection) oder die Benutzeraktivitäten mit bekannten Angriffsmethoden (misuse detection) vergleichen. Die meisten IDS machen zur höheren Sicherheit beides (nach [Mutlu 01]).

Des Weiteren werden zwei Typen von IDS unterschieden. Zum einem sind dies netzwerkbasierte IDS (NIDS), die den gesamten Datenverkehr eines Netzwerkes auf anormales Verhalten und bekannte Angriffsmuster hin überwachen. Das setzt voraus, dass der gesamte Datenverkehr aufgrund einer speziellen Netzarchitektur (zum Beispiel Ring, Bus) auch an das IDS geht. Zum anderen sind dies hostbasierte IDS (HIDS), die auf dem zu überwachenden System selbst laufen. Dabei wird nicht nur der Kommunikationsverkehr geprüft, sondern auch auf die Integrität der Systemdateien geachtet.

4.3 NETZWERK- UND HOSTBASIERTE INTRUSION DETECTION SYSTEME

Eine Möglichkeit, verschiedene IDS voneinander zu unterscheiden, ist, woher sie ihre Daten für die Angriffserkennung bekommen. Die IDS lassen sich danach in netzwerkbasierte IDS und hostbasierte IDS einteilen.

4.3.1 NETZWERKBASIERTE INTRUSION DETECTION SYSTEME (NIDS)

Netzwerkbasierte IDS überwachen, wie der Name schon andeutet, den Verkehr auf einem bestimmten Netzsegment.

Zu diesem Zweck untersuchen NIDS jedes Paket, das auf dem Netz unterwegs ist. Sie zerlegen die Pakete und überprüfen dann deren Inhalt. Dies geschieht automatisch, denn das manuelle Zerlegen und Begutachten der Pakete ist zu zeitraubend, um im Falle eines Angriffes von Nutzen zu sein.

Um alle Pakete einzufangen, wird die Netzwerk-Interface-Karte im wechselnden Modus betrieben. Netzwerk-Interface-Karten arbeiten in einem von zwei Modi:

- Im normalen Modus werden alle Pakete, die für einen Computer bestimmt sind, auch direkt zu ihm weitergeleitet.
- Im wechselnden Modus werden alle Pakete, die sich auf dem Netz befinden, erst einmal auch an das IDS weitergeleitet.

Nach [Mutlu 01] sind NIDS Weiterentwicklungen von Paket-Sniffern und Netzwerkmonitoren. Diese fangen die Pakete ein und haben dann mehrere Möglichkeiten der Paketnutzung: Sie zählen die Pakete über eine Zeitperiode, um die Netzwerkbelastung zu bestimmen. Eine andere Möglichkeit ist die Pakete detailliert zu überprüfen. Daraus wurden NIDS entwickelt. Ein Beispiel für ein NIDS ist die ISS RealSecure Engine. Diese überprüft die Pakete auf dem Netz, erlaubt den legitimen Paketen das Durchlaufen und unterbricht gegebenenfalls die Paketübertragung. Außerdem zeichnet es den Durchlauf von Paketen auf, um ihn später zu analysieren. Damit kann RealSecure eine zweite Verteidigungslinie hinter einer Firewall bilden.

Des Weiteren haben netzwerkbasierte IDS noch folgende Möglichkeiten, Angriffe zu entdecken:

- Sie sind in der Lage Portscans zu entdecken. Dies ist wichtig, weil den meisten Angriffsversuchen ein Portscan voraus geht, um Schwachstellen des Systems und somit Angriffsmöglichkeiten zu entdecken.
- NIDS werden, wie schon beschrieben, zur Entdeckung von bekannten Angriffsmustern benutzt.
- Netzwerkbasierte IDS identifizieren Spoofing Versuche, in dem sie sich die Quelladressen genau ansehen.

Wenn ein NIDS einen Angriff bemerkt, kann es Gegenmaßnahmen ergreifen. In Zusammenarbeit mit einer Firewall veranlasst das NIDS die Firewall, sämtlichen oder nur den für einen Angriff relevanten Verkehr, der von einem Rechner kommt, zu blockieren. Die Firewall schließt zum Beispiel über Packet-Screening sämtliche Ports, die für den Angriff benutzt werden.

4.3.2 HOSTBASIERTE INTRUSION DETECTION SYSTEME (HIDS)

HIDS laufen auf dem zu beobachtenden System selbst. Sie überprüfen den Kommunikationsverkehr innerhalb oder außerhalb des Systems, kontrollieren die Integrität der Systemdateien und versuchen, verdächtige Prozesse zu entdecken. Es gibt zwei Haupttypen von hostbasierten IDS:

- Als erstes zu nennen sind die Netzwerkmonitore, die den hereinkommenden Netzwerkverkehr überwachen. Im Unterschied zu den Netzwerkmonitoren der NIDS überprüfen sie nicht den gesamten Netzwerkverkehr, sondern nur den für den Host relevanten Teil. Deshalb läuft die Netzwerk-Interface-Karte hier im normalen Modus.
- Der zweite Typ sind Host-Monitore. Diese überwachen den Host selbst. Sie überprüfen zum Beispiel einzelne besonders zu schützende Dateien, Dateisysteme und Log-Files.

Die Aufgaben eines HIDS sind vielschichtig. Zum Beispiel werden hereinkommende Verbindungen daraufhin überprüft, ob es sich um einen Portscan handelt oder ob jemand versucht eine Verbindung zu einem Port herzustellen, hinter dem sich kein Dienst befindet. Dies wird gemacht, um verdächtige Pakete abzufangen, bevor sie Schaden verursachen. Als Gegenmaßnahme wird zum Beispiel die Firewall alarmiert oder die lokale Konfiguration modifiziert.

Eine weitere Aufgabe ist die Untersuchung von Login-Aktivitäten. Ein Hacker wird versuchen, die Kontrolle über ein System mittels Login-Informationen zu erlangen. Wenn ihm das Passwort nicht oder nur wage bekannt ist, werden Fehlversuche auftreten und ihn verdächtig machen.

Eine besonders wichtige Aufgabe für ein HIDS ist die Überwachung von Root-Aktivitäten, weil viele der Eindringlinge versuchen Root- oder Administrator-Zugriffsrechte auf dem System zu erlangen. Mit diesen Rechten kann der Angreifer beliebig großen Schaden im System anrichten. Deshalb sollten diese Rechte ständig durch Root-User oder Systemadministratoren überwacht werden. Wenn dies nicht immer möglich ist, übernimmt das IDS diese Aufgabe. Da meist die Wartung des Systems zu bestimmten Zeiten passiert, kann der Eindringling leichter erkannt werden, denn dem Angreifer sind diese Zeiten vermutlich nicht bekannt. Außerdem sind Cracker eher zu unüblichen Zeiten aktiv und werden deshalb mit höherer Wahrscheinlichkeit bemerkt.

Auch Dateisysteme sind ein beliebtes Ziel von Angreifern, weil hier die sensiblen Daten des Systems und eventuell des Benutzers liegen. Sie werden überwacht, um Veränderungen der

Dateien schnell zu entdecken. Eine Möglichkeit, sie zu schützen, sind kryptographische Prüfsummenverfahren, das heißt, wenn eine Datei verändert wird, dann ändert sich auch die Prüfsumme, was das HIDS sofort bemerkt. Die zweite Möglichkeit ist die Verwendung von Zeitstempeln bei der Anlegung und Veränderung von Dateien. Das HIDS sucht dabei nach verdächtigen Veränderungen in den Zeitstempeln. Die dritte Möglichkeit ist das Aufzeichnen von Root- oder Administrator-Aktivitäten. Wird diese Aufzeichnung verändert oder gelöscht ist dies ein Hinweis auf einen Angriff.

Nach [Mutlu 01] gibt es zwei Hauptklassen von hostbasierter IDS Software. Die eine Klasse sind Host Wrappers oder Personal Firewalls und die andere Klasse ist agentenbasierte Software. Beide Klassen sind effektiver im Vergleich zu NIDS bei der Entdeckung von Angriffen vom Inneren eines Netzes. Aber auch bei der Entdeckung von Angriffen von außen sind sie relativ effektiv.

Host Wrappers oder Personal Firewalls überwachen alle Pakete und Verbindungs- bzw. LogIn-Versuche bis hin zu nicht netzwerkbezogenen Kommunikationsports. Außerdem entdecken viele Personal Firewalls wie zum Beispiel ZoneAlarm (welches kein IDS ist) auch Programme, die versuchen vom Host aus Verbindungen in das Netzwerk aufzubauen.

Agentenbasierte Software überwacht die Zugriffe, Veränderungen in Systemdateien und Benutzerprivilegien.

NIDS und HIDS haben nach [Mutlu 01] bestimmte Vor- und Nachteile. Beide IDS belasten ihre zu überwachenden Komponente also das Netz oder den Host nur gering. NIDS erkennen IP-basierte Angriffe wie Denial-of-Service recht gut, während HIDS dabei Schwierigkeiten haben. HIDS können Angriffe in gekapselten Protokollen analysieren. NIDS erkennen diese Angriffe nur schlecht. Bei NIDS ist, wie in 4.1. erläutert, eines der Hauptprobleme, wo es im Netz am besten positioniert wird. Auch die für Hochgeschwindigkeitsnetze zu schlechte Performanz ist ein Problem. HIDS sind wegen ihrer Plattformabhängigkeit durch Betriebsystemschwächen angreifbar und verursachen hohen administrativen Aufwand.

Um eine möglichst hohe Erkennungsrate von Angriffen zu haben, werden beide Typen – netzwerkbasiertes IDS und hostbasiertes IDS – zusammen eingesetzt.

4.4 DIE ARBEITSWEISE VON INTRUSION DETECTION SYSTEMEN

IDS benutzen Audit-Trails um herauszufinden, ob ein Angriff stattfindet. Das Audit-System zeichnet die relevanten System- bzw. Benutzeraktivitäten auf, so dass das IDS anhand von Regeln, die auf die gewonnenen Daten angewendet werden, Alarm schlagen und der System Security Officer (SSO) oder eine Firewall selber Gegenmaßnahen einleiten kann.

Ein IDS wird deshalb auch in drei Komponenten geteilt:

1) Es enthält ein Auditing-System zur Datensammlung. Dabei werden verschiedene Systemkomponenten und/ oder der Netzwerkverkehr überwacht.

2) Des Weiteren wird eine Komponente zur Datenanalyse gebraucht.

3) Die dritte Komponente stellt die Analyseergebnisse benutzergerecht dar.

Auditing

Das Audit-System zeichnet bei hostbasierten IDS auf, welcher Benutzer wann auf welche Datei zugreift. Bei netzwerkbasierten IDS wird der vorüberlaufende Netzwerkverkehr protokolliert. Die Audit-Datensätze stammen aus verschiedenen Quellen (nach [Helden 98]):

- Auditdaten aus unterschiedlichen Systemeinheiten
  Meldungen finden auf einer hohen Abstraktionsebene statt. Z. B.: Wer hat sich angemeldet? Wann traten Schutzverletzungen auf?
  Insbesondere der Zugriff auf Dateisysteme wird überwacht, so werden auch die Zugriffe auf Dateien protokolliert. Das IDS meldet, wenn ein Benutzer seine Zugriffsrechte überschreitet oder wenn auf eine Datei wesentlich häufiger zugegriffen wird als üblich. Neben dem Dateisystem werden auch noch die Netzdienste überwacht, zum Beispiel wer sich von außen angemeldet hat. Das IDS schlägt Alarm, wenn Benutzer von außen nicht autorisiert sind, aber dennoch Zugriff auf das Innere des Netzes erlangt haben. Schließlich überwacht das IDS Systemkomponenten, die besonders schutzbedürftig sind und deshalb zum Beispiel schon durch eine Firewall geschützt werden. Zudem kann es zum Schutz der Firewall selbst verwendet werden.

- Betriebsmittelvergabe durch das Betriebssystem
  Parameter wie CPU-Auslastung und Anzahl der aktiven Netzverbindungen sind hier im besonderen Blickfeld. Zum Beispiel ist eine besonders hohe Anzahl der aktiven Netzverbindungen ein mögliches Zeichen für ein Denial-of-Service Angriff.

- Netzwerkdurchsatz
  Parameter wie Quell- und Zieladresse oder der Quell- und Zielports sind hier von Interesse. So sind zum Beispiel einige Ports wichtig für den Angriff durch bestimmte Toolz (=Angriffstools, siehe Kapitel 1). So verwendet Back Orifice 2000 meist den Port 54320.

Das Problem dabei ist, dass die Audit-Datensätze für die Analyse nicht zu viel aber auch nicht zu wenig Informationen enthalten dürfen und das große Audit-Datenmengen das Computersystem oder Netzwerk zu stark belasten Denn die Analyse sollte möglichst in „real time“ stattfinden, um schnell auf einen Angriff reagieren zu können. Wenn aber zu wenig Informationen im Audit-Datensatz enthalten sind, wird der Angriff eventuelle erst gar nicht erkannt. Sind die Datensätze zu groß, werden die für die Angriffserkennung wichtigen Informationen im Audit-Trail nicht schnell genug oder gar nicht gefunden. Bei zu großen Datenmengen leidet nicht nur die Performanz des Systems, sondern auch die Leistung des IDS, weil es diese nicht schnell genug verarbeiten kann, um noch in „real time“ zu reagieren.

Auf die so gewonnenen Audit-Datensätze wendet das IDS nun seine Regeln für die Angriffserkennung an.

Datenanalyse

Für die Datenanalyse gibt es zwei unterschiedliche Vorgehensweisen. Die erste Methode heißt „anomaly detection“. Sie untersucht, ob das Benutzerverhalten von den „normalen“ Benutzeraktivitäten abweicht. Die zweite Methode ist „misuse detection“. Hierbei werden die Benutzeraktivitäten auf bekannte Angriffsmuster untersucht.

Anomaly detection

Bei der Erkennung von Anomalien wird das IDS mehrere Wochen oder Monate in einem Lern- oder Trainingsmodus mitgefahren. Es zeichnet dabei auf, was der „normale“ Betrieb auf dem System oder in dem Netz ist. Das Benutzerverhalten wird in einer Profildatenbank gespeichert. In der Fachliteratur wird die „anomaly detection“ häufig auch als „paranoider Ansatz“ bezeichnet, weil alles was das System vorher noch nicht gesehen hat, als möglicher Angriff betrachtet wird. Nach [Helden 98] gibt es zwei Ansätze, um das „normale“ Verhalten der Benutzer abzuleiten:

- Die statischen Ansätze versuchen zunächst Normalwerte zum Beispiel für CPU-Auslastung, Dateizugriffe oder Anzahl der aktiven Netzwerkverbindungen zu bestimmen. Von diesen Normalwerten werden zustandsunabhängige Mittelwerte (zum Beispiel durchschnittliche Anzahl der aktiven Netzwerkverbindungen)

gebildet. Es wird aber auch mit bedingten Wahrscheinlichkeiten also zustandsabhängigen Mittelwerten gearbeitet. Weicht der Parameter von seinem Normalwert mit einer bestimmten Größe ab, löst das IDS einen Alarm aus. Schwierig ist es nun eine akzeptable Abweichung von dem Normalwert zu bestimmen. Denn wird diese Abweichung zu klein gewählt, gibt es Fehlalarme. Ist die Abweichung zu groß, besteht die Gefahr, dass Angriffe nicht erkannt werden.

- Bei der logischen Analyse steht die zeitliche Abfolge von Ereignissen in Mittelpunkt. Das Normalverhalten wird in Form von Regeln beschrieben. Erkennt das IDS den Anfang einer bestimmten Folge von Ereignissen anhand einer Regel, dann erwartet es, dass die Ereignisfolge wie in den Regeln beschrieben wurde, zu Ende geführt wird. Ist dies nicht der Fall wird ein Alarm ausgelöst.

Problematisch hierbei ist, die Parameter der Regeln richtig zu setzen, insbesondere da es in einem Netz häufig zu Veränderungen durch Neukonfigurationen von Netzwerkteilen aber auch durch neue Aufgabenstellungen für die Benutzer zu verändertem Benutzerverhalten kommt. Deshalb müssen die Regeln ständig angepasst werden, um Fehlalarme zu vermeiden. Das heißt auch, dass wie bei den statischen Ansätzen die Regeln so eingestellt sein müssen, dass leichte Abweichungen vom normalen Verhalten nicht sofort einen Alarm auslösen. Andererseits dürfen die Regeln nicht zu große Abweichungen tolerieren, weil sonst mögliche Angriffe nicht erkannt werden.

Eine andere Ursache für falsche Alarme ist, dass nicht das ganze Verhalten des Systems in der „Lernphase“ auftrat. Zudem ist die „Lernphase“ des IDS ein kritischer Moment, denn wenn in dieser Zeit das Netz oder ein System Opfer eines Hacker-Angriffes wird, dann kann das IDS diese Art von Hacker-Angriffen als normalen Betrieb auffassen.

Natürlich hat die anomaly detection auch einige Vorteile. Mit ihrer Hilfe können unbekannte Angriffsmuster entdeckt und schon beim ersten auftreten des Angriffes abgewehrt werden. Ein weiterer Vorteil ist, dass der Missbrauch von Benutzerprivilegien also ein Angriff aus dem Inneren des Netzes leichter erkannt wird, da anomaly detection gerade auf der Entdeckung von anormalen Benutzerverhalten basiert. Insgesamt muss aber gesagt werden, dass nach [Helden 98] anomaly detection noch nicht ausgereift genug ist, um in der Praxis alleine eingesetzt zu werden. Deshalb wird noch häufig die misuse detection eingesetzt.

Misuse detection

Bei der Erkennung von Missbrauch sucht das IDS nach bekannten Angriffsmustern. Dazu muss es wissen, worauf der Angriff basiert, zum Beispiel nutzen so genannte WinNuke-Programme eine Sicherheitslücke von Windows-Systemen aus, um über den Port 139 das System zum Absturz zu bringen. Hierzu gibt es bestimmte Signaturen oder Angriffsmuster, die in der Signaturdatenbank des IDS gespeichert sind. Die Signaturdatenbank ist die zentrale Komponente der Missbrauchserkennung. Durch eine detaillierte Analyse ist der Sicherheitsbeauftragte in der Lage auf Angriffe zu reagieren. Eine genaue Analyse durch das IDS ist erst einmal etwas zeitraubend, aber umso besser und umso schneller kann der SSO danach Gegenmaßnahmen ergreifen. Dadurch ist die misuse detection bei der richtigen Angriffserkennung relativ genau, woraus eine niedrige Falsch-Alarm-Rate resultiert. Der Nachteil ist, dass neue Angriffstechniken meist nicht erkannt werden. Deshalb bedarf die Signaturdatenbank, wie die eines Virenscanners, einer ständigen Wartung mit neuen Signaturen. Die Erstellung von Signaturen von neu entdeckten Angriffsmustern ist relativ schwierig, da sie ein umfangreiches Wissen über die genutzte Schwachstelle, deren Ausnutzung durch den Angreifer und die Möglichkeiten dem Angriff entgegenzuwirken erfordern. Dazu gehört, dass die Angriffe abhängig sind von Betriebssystem, Version und den Anwendungen die darauf laufen. Dies bindet das IDS an diese speziellen Umgebungen. Schließlich ist die Entdeckung von Angriffen aus dem Inneren des Netzes kaum möglich, weil die Angreifer keine Schwachstellen des Systems ausnutzen, sondern ihre Rechte missbrauchen.

Um die Schwächen beider Methoden zu kompensieren, werden häufig beide Vorgehensweisen verwendet.

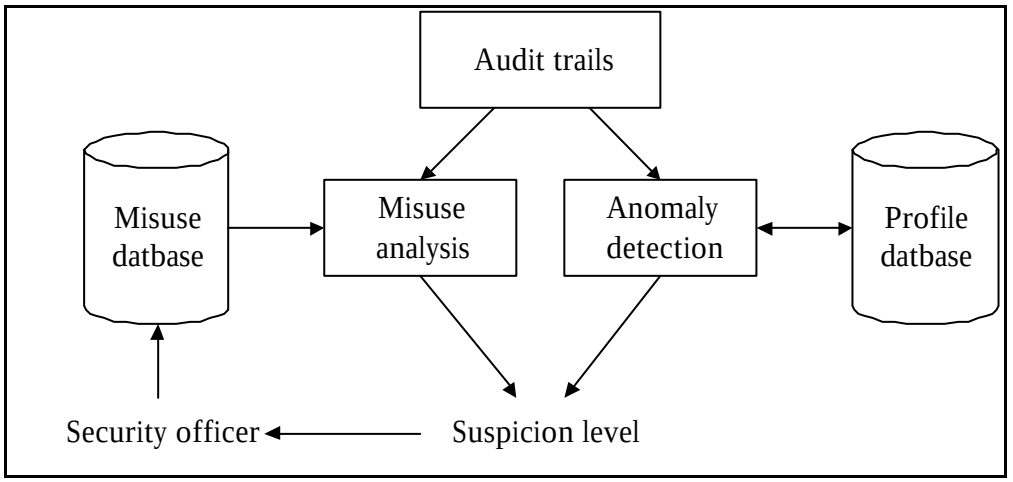


Abb. 19: Datenanalyse mittels Misuse und Anomaly Detection nach [Castano 94]

Zum Ende dieses Abschnitts wird noch kurz auf die so häufig erwähnten Regeln eingegangen.

Regelerstellung

Für die Regelerstellung werden die Mittel der Logik verwendet. Durch Anwendung von Inferenzregeln auf die Audit-Datenmengen werden Schlüsse über Angriffe gezogen. Dabei stellen die Audit-Daten die Faktenbasis dar. Das Wissen über das Erkennen von Angriffen wird in Form von Inferenzregeln gebracht (Wissensbasis/ Regelbasis). Diese Regeln bestehen aus einer Vorraussetzung und einer Folgerung. In der Vorraussetzung steht was erfüllt sein muss, damit die Regel zur Anwendung kommt. In der Folgerung steht was passiert, wenn die Vorraussetzung erfüllt ist. Die wichtigste Aufgabe ist es neue Fakten in die Faktenbasis einzufügen und nicht mehr gebrauchte aus ihr zu löschen. Die Inferenzregeln werden solange auf die Fakten angewandt bis ein Ergebnis herauskommt oder die Faktenbasis leer ist. Solche Ergebnisse führen dann zu einem Alarm, d. h. ein Angriff wurde erkannt. Die Inferenzregeln sind als Signaturen in das IDS eingebracht und stellen das Rückgrad der Missbrauserkennung dar. Die Regelerstellung für anormales Verhalten ist wesentlich schwieriger und es gibt heutzutage kein IDS, welches nur auf anomaly detection basiert. Genauere Angaben über die Regelerstellung sind in [Mutlu 01] zu finden.

Ergebnisdarstellung

Die Ergebnisse der Datenanalyse werden dem SSO, welcher ein Administrator sein kann, auf einer grafischen Benutzeroberfläche dargestellt. Je nach Erkennungsmethode werden unterschiedliche Präsentationen genutzt. Bei der Missbrauchserkennung gibt das IDS aus, ob ein Angriff entdeckt wurde. Es erkennt einen bestimmten Angriffstyp, so dass der Sicherheitsbeauftragte einschätzen kann, wie schwer der Angriff ist. Außerdem wird dieser bei entsprechenden Kenntnissen die erforderlichen Gegenmaßnahmen ergreifen. Die Anomalieerkennung liefert eine so genannte Verdachtbewertung, die angibt, wie stark ein Parameter vom Normalwert aktuell abweicht. Ist die Abweichung zu groß, wird der SSO benachrichtigt. Dafür gibt es nach [Helden 98] mehrere Wege:

- Das Sicherheitspersonal beobachtet ständig den Benachrichtigungsbildschirm und ergreift sofort Maßnahmen.
- Der Sicherheitsbeauftragte wird mittels Pager oder Mobiltelefon auch über größere Entfernungen hinweg benachrichtigt.
- Eine Alarmmeldung wird an ein entferntes System über ein gesichertes Netzsegment gesendet.
- Die Benachrichtigung wird über das interne Netz übermittelt, wobei die Verbindung kryptographisch gesichert sein muss und zudem ein Denial-of-Service Angriff das Netz nicht lahm legen darf.

Als nächstes wird darauf eingegangen was das IDS oder der SSO machen kann oder soll, wenn ein Angriff erkannt wurde.

4.5 INTRUSION RESPONSE

Unter Intrusion Response (IR) werden nach [Helden 98] automatische Gegenmaßnahmen bei einem Angriff verstanden. Die Intrusion Response muss, um erfolgreich zu sein, in der Sicherheitspolitik festgelegt sein. Zur Intrusion Response gehören nach [Helden 98] drei Ziele:

- die Identifizierung des Angreifers
- der Schutz vor weiteren Schäden
- die Behebung des Schadens

Diese Ziele sind zum Teil gegenläufig, weshalb es besonders wichtig ist, in der Sicherheitspolitik festzulegen, welche Ziele wie erreicht werden sollen.

Dabei ist es besonders hilfreich, wenn ein Maßnahmenkatalog festgelegt ist. Die Umsetzung der Maßnahmen kann automatisch, halbautomatisch oder manuell erfolgen. Eine automatische Umsetzung könnte ein Intrusion Response System (IRS) übernehmen. Heutzutage gibt es allerdings noch kein reines IRS, deshalb muss diese Aufgabe durch einen SSO und das IDS übernommen werden. Bei der halbautomatischen IR ist eine Interaktion zwischen SSO und IRS notwendig. Die manuelle IR wird alleine durch den SSO durchgeführt.

Um weitere Schäden zu verhindern, kann der Rechner abgeschottet werden oder ein Gegenangriff gestartet werden. Für eine Abschottung wird die Firewall rekonfiguriert, so dass sämtliche Pakete vom angreifenden Rechner abgeblockt werden. Des weiteren werden alle Ports geschlossen (die für den Angriff benutzt werden) und die entsprechenden Dienste und Programme beendet. Falls der Angriff von einem internen Benutzer kommt, wird dessen Account erst einmal gesperrt.

Ein Gegenangriff ist juristisch sehr problematisch und ethisch kaum vertretbar. Er hat zum Ziel zum Beispiel mittels einer Denial-of-Service-Attacke den Angriffsrechner außer Gefecht zu setzen. Dazu muss als erstes die Absenderadresse des Angreifers ausfindig gemacht werden. Dabei ist zu beachten, dass der Angreifer seine Adresse nicht gefälscht hat (Spoofing) und dass er keinen anderen Rechner zum Angriff „als Sprungbrett“ missbraucht hat.

Die Identifizierung des Angreifers erscheint zu erst dem Schutz vor weiterem Schaden gegenläufig zu sein. Aber zum Beispiel bei wiederholten, erfolgreichen Angriffen eines Hackers

ist es sinnvoller, um weiteren Schaden zu vermeiden, diesen zu identifizieren und dabei eine kurzfristig höhere Schädigung des Systems in Kauf zu nehmen. Um dies zu vermeiden kann der Angreifer aber auch in eine so genannte Gummizelle gelockt werden, wie in [Helden 98] erwähnt.

Für die Identifizierung des Angreifers müssen umfangreiche Protokollierungsmaßnahmen ergriffen werden. Diese sollen später auch noch als Beweismittel vor Gericht dienen können und müssen deshalb besonderen Anforderungen genügen. Insgesamt ist die Suche nach dem Angreifer äußerst schwierig. Es gibt keine klaren Regeln den Angreifer ausfindig zu machen. Der Aufwand ist sehr groß.

Oft ist es nicht möglich den entstandenen Schaden zu beheben, deshalb ist es wichtig immer gut gesicherte und aktuelle Sicherungskopien zu haben. Diese dürfen selbst nicht kompromitiert worden sein und sind deshalb besonders schutzbedürftig. Aber auch beim Einsatz von Sicherheitskopien oder anderen Recovery-Mechanismen sind häufig die letzten Änderungen von Dokumenten etc. nicht wieder herzustellen. Dies ist im besonderen Maße abhängig vom Intervall, in dem Sicherungskopien oder BackUps erstellt werden.

5 VERSUCHSREIHE

In den vorangegangenen Kapiteln wurden die theoretischen Grundlagen ausführlich geklärt. In diesem Kapitel beschreiben wir nun die Abläufe und Ergebnisse der Versuchsreihe zu dem Projekt „Intrusion Detection Systeme in Firewalls“. Zunächst stellen wir das Szenario vor, das uns als Grundlage diente. Darauf folgen dann der Versuchsaufbau und die einzelnen Versuchsphasen.

5.1 SZENARIO

In vielen kleinen Firmen gibt es aus Kostengründen keinen qualifizierten Netzwerkadministrator. Diese Tätigkeit wird von einem Mitarbeiter mit primär anderen Aufgaben als Nebenjob übernommen. Aus diesem Grund haben wir uns für folgendes Szenario entschlossen.

Ein Kleinunternehmer möchte für sein Detlef Abstauber Untrusted Schnellwaschzentrum die Finanzverwaltung automatisieren. Da er in einer Großstadt mit viel Konkurrenz zu tun hat, geht er eine Partnerschaft mit anderen Kleinunternehmern seiner Branche in einem Verbund ein. Hiervon verspricht er sich Vorteile in Bezug auf einen günstigeren Einkauf durch Großabnahme und gemeinsame Werbung. Für eine einfachere Abrechnung und um die Gebühren für den Mann in Gelb zu sparen, hat sich der Verbund dazu entschlossen, eine gemeinsame internetbasierte Finanzverwaltung für Windows 9x aus dem Jahre 1999 zu benutzen. Die Software nutzt ein eigenes auf TCP/IP basiertes Application Level Protokoll auf TCP-Port 54320. Dadurch soll ermöglicht werden, dass die einzelnen Waschzentren ihre Rechnungen jederzeit selber vom Server in der Verwaltung abrufen können. Zusätzlich sollen für die Bilanz des Verbundes am Ende eines jeden Monats die Umsätze der einzelnen Waschzentren eingesammelt werden. Da die Verwaltung keine guten Erfahrungen in Bezug auf die Zuverlässigkeit der Verbundteilnehmer gemacht hat, ruft die Verwaltung diese Daten von den einzelnen Rechnern der Verbundteilnehmer ab.

Da unserem DAU Schnellwaschzentrum finanzielle Mittel nur begrenzt zur Verfügung stehen, entscheidet sich der Kleinunternehmer, seinen pubertierenden sechszehnjährigen Sohn Hugo, dessen Spitzname Honk ist, mit dem Aufbau eines kleinen Netzwerkes mit Internetzugang zu beauftragen. Da der Vater Sohnemanns Computer finanziert hat, mit dem dieser seine gesamte Freizeit verbringt, soll er doch sein ganzes Wissen in das väterliche Unternehmen einbringen, damit er sich auch mal nützlich machen kann.

Da Honk sich aus seiner bisherigen Erfahrung bereits bestens mit Windows auszukennen glaubt, will er auf Betriebssystemebene bei den vertrauten Microsoftprodukten bleiben. So richtet

Hugo einen Anwendungsrechner und wegen der Erweiterbarkeit des Netzwerkes ein Gateway auf Basis Windows NT Server ein.

Diese Aufgabe muss er natürlich groß in der Schule verkünden. Davon hört nun auch sein Mitschüler Christian R. Acker, dessen Mutter regelmäßige Kundin in dem Waschzentrum von Herrn Abstauber ist. Christians neue Designerhose ist bei der letzten Wäsche mangels Qualität des Waschmittels eingelaufen. Da er keinen Ersatz bekommen hat, möchte C. R. Acker sich nun an Herrn Abstauber rächen. Er sucht im Internet nach passenden Anleitungen, um Herrn Abstauber eines besseren zu belehren. Dazu sucht er im Internet nach Maßnahmen für „educational purposes“ und stößt auf passende „Appz“ und „Toolz“ zu diesem Zweck.

Christian wendet nun diese Programme anhand der beiliegenden Anleitungen an. Herr Abstauber wundert sich, und da sein Sohn selbst ratlos ist, wenden sie sich an den Verbund. Dieser ist geschockt, da sich die Berichte über solche Vorfälle häufen. Aufgrund der Erfahrung entwickelt der Verbund im Namen seiner Partner für die Nutzung der Finanzverwaltung ein Sicherheitspolitik. Für die Umsetzung ist jeder Partner eigenverantwortlich.

Die Sicherheitspolitik sieht in erster Linie vor, dass die Finanzverwaltung funktionieren muss. Zudem ist die Kommunikation der Partner über E-Mail vorgesehen. Auf drängen einiger Waschzentren, soll auch der Einsatz von Webservern sowie das Surfen im WWW ermöglicht werden. Alle anderen Dienste sind standardmäßig nicht erlaubt.

Honk stöbert daraufhin in seiner verstaubten Sammlung von Computerzeitschriften, die er alle 14 Tage am Kiosk kauft, und befragt einige Internetsuchmaschinen. Nachdem er ausgiebig recherchiert hat, entscheidet er sich für das mächtige Firewall Toolkit Pgp Gauntlet 5.5 für Windows NT von Network Associates.

Trotz der Firewall fühlt sich Honk dennoch nicht sicher. Er beobachtet immer noch Unregelmäßigkeiten, die sich zudem noch im Log bemerkbar machen. Deshalb will er zusätzlich ein Intrusion Detection System einsetzen. Hierzu wählt er das zu Gauntlet passende Produkt Cybercop Monitor von NAI. Mit diesem möchte er nun die Sicherheitsauflagen des Verbunds erfolgreich umsetzten.

5.2 VERSUCHSAUFBAU

Das so eben beschriebene Szenario haben wir unter Laborbedingungen und unter entsprechendem Abstraktionsgrad simuliert. Es folgt nun eine Beschreibung der Testumgebung.

5.2.1 ALLGEMEINER TECHNISCHER AUFBAU

Es standen uns drei Standard-PCs zur Verfügung, die im AGN-Labor die Namen Felix, Ergo und Kathy hatten.

Zur Umsetzung des Szenarios, und um ausgiebige Tests vornehmen zu können, wurde festgelegt, dass die schützenswerten Netzteilnehmer des Intranets (blaues Testnetz) durch Kathy repräsentiert werden sollten. Alle Angriffe sollten aus dem Internet (gelbes Testnetz) und dessen Teilnehmer-Repräsentanten Felix erfolgen. Somit wurde der Rechner Ergo mittels zweiter Netzwerkkarten zum Vermittler (Gateway) beider Netze.

Um das Internet und ein Intranet im Labor nachzustellen, legten wir uns auf die Protokolle des TCP/IP-Stacks oberhalb von Ethernet (IEEE 802.3, s.a. [IEEE]) im gelben Testnetz (Internet) und oberhalb von Fast Ethernet (IEEE 802.12, s.a. [IEEE]) im blauen Testnetz (Intranet) fest. Zur besseren Auftrennung der Netze in simulierte Intra- bzw. Internets erhielten die Computer im Intranet Adressen aus dem Band 192.168.1.x und die Rechner des Internets aus dem Band 192.168.0.x. Die genaue Vergabe der IP-Adressen kann der folgenden Abbildung entnommen werden.

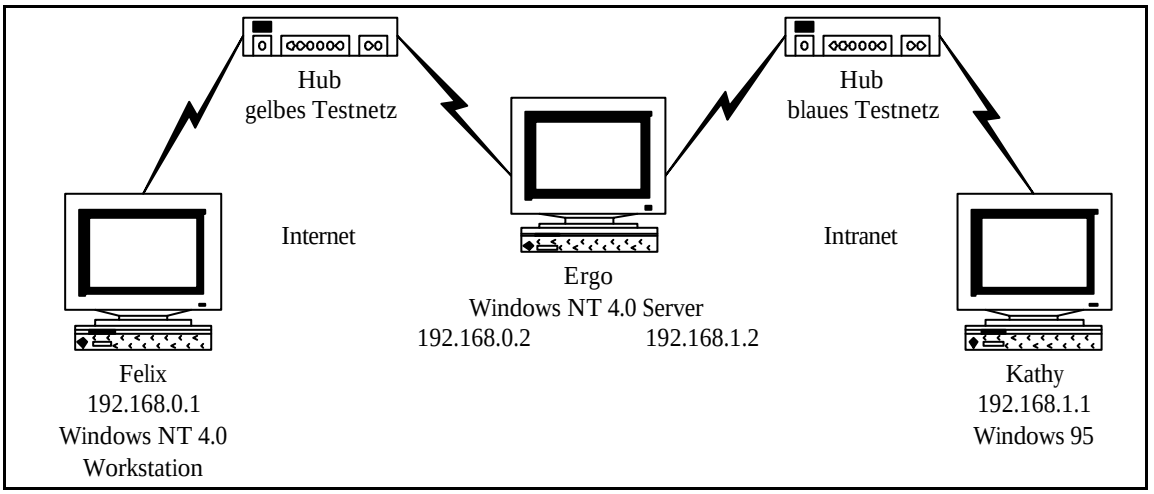


Abb. 20: Netzkonfiguration

Als Betriebssysteme wurden Microsoft Produkte verwendet. Bei dem Intranet-Teilnehmer Kathy wurde die erste Version von Windows 95 installiert, damit waren alle Sicherheitslücken, die dieses Produkt bietet, zur Nachahmung potenzieller Angriffe nutzbar. Das Gateway benötigte wegen nötiger Erweiterbarkeit zur Firewall mit aufgesetztem Intrusion Detection System Windows NT 4.0 Server mit Service Release 6a.

Zu guter Letzt wurde der Computer Felix, von dem die Hackerangriffe erfolgen sollten, mit Windows NT 4.0 Workstation versehen. Zur besseren Systemstabilität wurde auch hier das Service Release 6a eingespielt.

5.2.2 ANGRIFFSZIELE

Nachdem der technischer Aufbau festgelegt war, wurden drei Grobangriffsziele ins Auge gefasst: Auslesen von Benutzernamen und Passworten; Ausspionieren, Löschen und Ersetzen von Dateien; Reduktion der Systemverfügbarkeit. Für diese drei groben Ziele wurde jedes Mal der Computer „Kathy“ als Opfer ausgewählt.

Im Detail definierten wir für die Angriffe folgende Aufgaben:

1) Da zur Vorbereitung eines jeden Hackerangriffs fast immer ein IP- und Port-Scan gehört, war die erste Aufgabe das Ausspähen des Zielnetzes, in diesem Fall also des blauen Testnetzes, um festzustellen, welche IP-Adresse und welche Ports im Netz aktiv sind.

2) Die zweite Aufgabe bestand im Erlangen einer Benutzerkennung und des zugehörigen Passworts für den Zielrechner. Mithilfe der Benutzerkennung und des Passworts können Angriffe in dem betroffenen Netz unter falschem Namen durchgeführt werden. Die Ergebnisse dieser Aufgabe würden also einem echten Cracker zur Tarnung dienen.

3) Im Gegensatz zu den ersten beiden Aufgaben, die der Vorbereitung und Tarnung eines Angriffes dienen, verlangten wir nun, dass die Dateien to_be_changed.txt und to_be_changed2.txt gesucht werden sollten. Die Letztere sollte außerdem gelöscht werden. Die Dateien befanden sich dabei in allen Versuchen in dem Root-Verzeichnis des angegriffenen Rechners.

4) Die nächste Aufgabe bestand in dem Einsehen einer Datei bzw. in dem Kopieren dieser Datei auf das System des Angreifers. Die Datei auf dem angegriffenen Computer war als to_be_changed.txt gespeichert und hatte folgenden Inhalt:
„Dies ist ein neuer blütenweißer Text, der noch niemals Kontakt mit irgendwelchen fiesen Angreifern hatte. Und das soll (und wird) auch so bleiben!!!“

5) Die fünfte Aufgabe verlangte, dass die Datei to_be_changed.txt geändert werden sollte.

6) Da im Prinzip der angegriffene Rechner nicht nur eine Workstation für einen Endanwender sein konnte, sondern auch im (blauen Test-) Netz Dienste für andere Workstations anbieten hätte können, sollte auch die Verfügbarkeit des angegriffenen Rechners reduziert werden. Hierbei unterschieden wir dann verschiedene Grade von Verfügbarkeit (s. 5.3).

7) Da alle oben genannten Angriffe ab dem zweiten Versuch mit aktivierter Firewall bzw. mit zusätzlichem Intrusion Detections System (IDS) vorgenommen werden sollten, bestand die letzte Aufgabe darin, entweder die Firewall-Software ohne bzw. mit IDS zu deaktivieren oder alternativ den Rechner „Ergo“ so weit außer Gefecht zu setzen, dass lediglich noch die Gateway-Funktionalität aufrecht erhalten blieb. Gelang diese Aufgabe, so gingen wir davon aus, dass die gleichen Ergebnisse wie im Kapitel 5.3 erreicht werden können.

5.2.3 WAHL DER ANGRIFFSTECHNIKEN

Zur Erfüllung dieser Aufgaben stand uns im Prinzip alles zur Verfügung. Statt aber selber neue Angriffsprogramme zu schreiben, reduzierten wir die Auswahl auf im Internet erhältliche Werkzeuge, denn bei einem erfolgreichen Angriff wäre uns damit gleichzeitig noch der Nachweis gelungen, dass faktisch jeder ohne Vorkenntnisse mittels Internet-Suchmaschinen diese Angriffe hätte selber durchführen können. Für folgende frei verfügbaren Techniken und Software haben wir uns entschieden:

Portscan

Ein Portscan bezeichnet den Vorgang, der überprüft, welche Dienste, die über Ports auf einem Rechner adressiert werden, offen, dass heißt, verfügbar sind.

Im eigentlichen Sinne ist ein Portscan also keine Angriffstechnik. Lediglich in Hochgeschwindigkeitsnetzen und/oder bei schlechten Treiberimplementationen des TCP/IP-Stacks kann ein Portscan die Puffer der Ports zum Überlaufen bringen bzw. so viele Antworten anfordern, dass der gescannte Computer diesen nicht mehr nachkommen kann. Tritt eine solche Situation ein, dann wird ein Portscan zur Denial-of-Service-Attacke, da das angegriffene System auch auf Anfragen anderer Rechner nicht mehr reagieren kann.

Ein Portscan wird häufig zur Vorbereitung von Angriffen genutzt (zum Beispiel Sniffing). Hierzu wird überprüft, ob und welche Dienste ein Computer, identifiziert durch seine IP-Adresse, zur Verfügung stellt. Ein Auszug einer Zuordnungsliste von TCP-Ports zu den darüber üblicherweise adressierbaren Diensten befindet sich in Anlage A. Hierbei ist aber zu beachten,

dass zum Beispiel der TCP-Port 8080, der normalerweise den Dienst eines HTTP-Proxy bietet, auch anderweitig genutzt werden kann, zum Beispiel durch Back Orifice 2000 (siehe Seite 93). Hacker wie Cracker versuchen die offenen, also die aktiven Ports für ihre Zwecke zu missbrauchen und nutzen deshalb Portscanner.

Neben diversen Eigenprogrammierungen der Hacker steht im Internet auch eine große Anzahl von Portscannern als Freeware, Shareware und kommerzielle Software zur Verfügung. Portscanner werden nicht nur mit den oben beschriebenen böswilligen Absichten eingesetzt, sondern dienen Netzadministratoren auch, um die von ihnen betreuten Netze auf Sicherheitslücken bzw. Verfügbarkeit von Diensten zu überprüfen.

Für das Projekt „Intrusion Detection Systeme in Firewalls“ haben wir uns für die Freeware SuperScan 3.00 von Foundstone entschieden, da es zum einen sehr repräsentativ für die Klasse der Portscanner ist und des weiteren einen hohen Grad an Benutzerfreundlichkeit bietet. Im Vergleich mit unserer Definition von Portscannern bietet SuperScan 3.00, wie viele andere Programme dieser Art, noch weitere Funktionalitäten, die über den eigentlichen Funktionsumfang von Portscannern hinausgeht, wie zum Beispiel IP-Scanning.

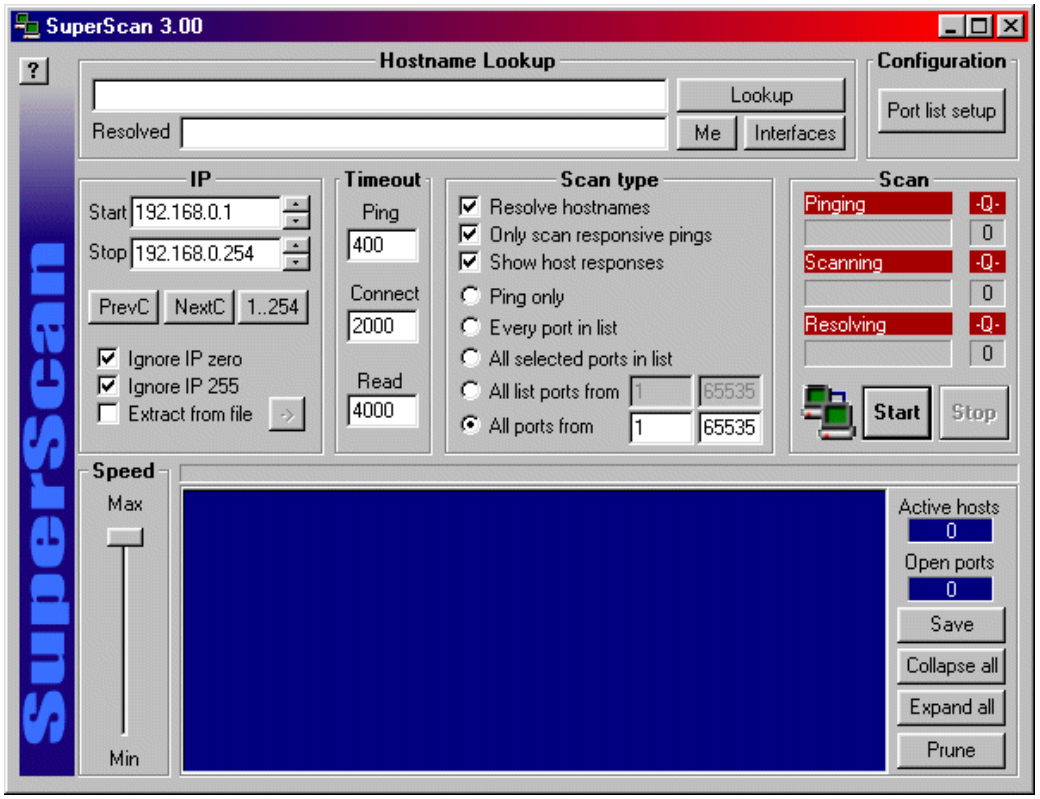


Abb. 21: SuperScan 3.00 Oberfläche

Beim IP-Scanning wird ein IP-Adressen-Band (zum Beispiel 192.168.0.1 bis 192.168.0.254) nach Computern durchsucht. Dazu werden an alle ausgewählten IP-Adressen Echo Requests geschickt. Der Empfänger schickt die empfangenen Daten als Echo Reply umgehend wieder

zurück, worüber dann erkannt werden kann, ob es diesen Rechner gibt bzw. ob dieser Computer gerade online ist und welche Laufzeit benötigt wurde.

SuperScan 3.00 verbindet IP-Scanning mit Port-Scanning, das heißt, dass die im durchsuchten Netz gefundenen aktiven IP-Adressen gleich einem Portscan unterzogen werden. Des Weiteren bietet SuperScan 3.00 die Möglichkeit, einen Rechnernamen über eine DNS-Anfrage in eine IP-Adresse auflösen zu lassen und umgekehrt. Durch alle diese Zusatzfunktionalitäten dieses Scanners werden die Anforderungen an die Vorkenntnisse des Angreifers über das Opfersystem reduziert. Außerdem lässt sich dieses Tool über Mausklicks bedienen und ist auch von einem DAU[3] ohne jegliches technisches Wissen einsetzbar. Damit ist SuperScan 3.00 ein fast schon optimales Werkzeug für sowohl den unerfahrenen als auch den erfahrenen Angreifer, um das Opfersystem erst einmal ins Visier zu nehmen. Genau für diese Funktionalität haben wir SuperScan 3.00 eingesetzt.

Ping of Death

Mit dem Ping-Befehl wird einem Anwender der ICMP-Dienst *Echo Request* bereitgestellt: Mit einem Ping sendet ein Computer ein Datenpaket, üblicherweise eine 32 Bytes lange alphabetische Zeichenfolge, an den adressierten Rechner, mit der Aufforderung, dieses Paket wieder zurückzusenden. Dieser Befehl wird gerne verwendet, um die Verfügbarkeit eines Rechners zu überprüfen oder die Laufzeiten im Netz zu analysieren. Ein Windows-Standard-Echo-Paket von 192.168.0.1 an 192.168.0.2 sieht hexkodiert wie folgt aus:

IP-Header	45	Version (4) & Header Länge (20B)
	00	Differentiated Services Field
	00 3c	Gesamtlänge (60)
	d8 00	Identifikationsnummer
	00 00	Flags & Fragmentversatz
	20	Restlaufzeit (32 Hops))
	01	Protokoll des IP-Daten (ICMP)
	41 6d	Header Checksumme
	c0 a8 00 01	Quelladress (192.168.0.1)
	c0 a8 00 02	Zieladresse (192.168.0.2)
ICMP-Header	08	Typ (8 entspricht Echo-Request)
	00	Code
	3e 5c	Checksumme
	01 00	Identifikationsnummer
	0e 00	Sequenznummer
Daten	61 62 63 64 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69	abcdefghijklmnopqrstuvwabcdefghi

Abb. 22: Analyse eines Echo Requests

[3] Dümmster Anzunehmender User

Das zurückgesendete Datenpaket (*Echo Reply*) entspricht den oberen Angaben, außer dass die Quell- und Zieladresse vertauscht und die IP-Identifikationsnummer, die Restlaufzeit, der ICMP-Typ von 8 (*Echo Request*) auf 0 (*Echo Reply*) sowie die Checksummen angepasst wurden.

Mit den vielen optionalen Parameter kann die Größe (Parameter -l [Paketlänge] in Microsoft Betriebssystemen) eines Ping-Paket geändert oder mit Parameter -t immer wieder versandt werden. Da durch den Ping-Befehl beim „angepingten“ Computer Ressourcen gebraucht werden (Speicher-/Pufferplatz und CPU-Zeit) erfüllen Pings bei entsprechender Parametrisierung den Tatbestand einer Denial-of-Service-Attacke. Eine spezielle Variante dieser Denial-of-Service-Attacken war der „Ping of Death“. Microsoft, wie viele andere Softwarehäuser, legte die Puffergröße für Ping-Pakete bei dem Betriebssystem Windows 95 relativ klein an. Wurde dann beim Verwenden des Ping-Pakets die Paketlänge auf 65510 Bytes (aufgrund der Fragmentierung ergab das eine Gesamtlänge des Paketes von 65535 Bytes) festgelegt, brachen die betroffenen Betriebssysteme auf der Empfängerseite zusammen, und im Falle von Microsoft zeigten sie die bekannten „Blue Screens of Death“ (BsoD).

WinNuke

1997 wurde entdeckt, dass Microsoft bei der Implementation des TCP-Protokolls in das Windows 95 Betriebssystem ein Fehler unterlaufen war. Sobald ein TCP-Paket mit gesetztem Urgent-Bit und -Pointer (auch Out-of-Band-Paket genannt) bei einem Windows95-Rechner eintrifft, erscheint eine als *Blue Screen* bekannte Ausnahmefehlermeldung. Normalerweise ist ein so „angegriffener“ PC danach noch betriebsbereit, lediglich die Kommunikation über ein Netz ist dann nur wieder nach einem Neustart des Systems möglich. Es gibt aber auch Computer mit Windows95-Installationen, die sich von den Out-of-Band-Paketen nicht mehr erholen und zur weiteren Nutzung wieder neugestartet werden müssen. In solchen Fällen sind dann auch alle ungespeicherten Informationen, wie neuerstellte Texte etc., verloren.

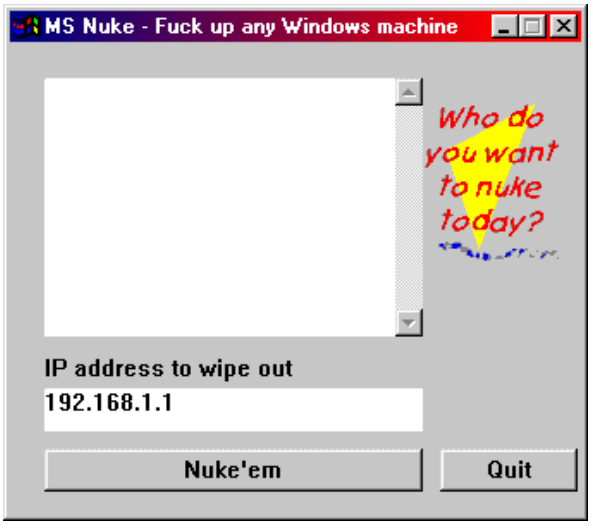


Abb. 23: Benutzeroberfläche von Liquid

Programme, die Out-of-Band-Pakete nur zum Zwecke eines Angriffes abschicken, werden Nuker oder WinNuke genannt. Das uns zur Verfügung stehende Werkzeug hatte den Programmnamen „Liquid“. Dieses baut zuerst eine NetBIOS Session Verbindung auf und sendet dann ein TCP-Out-of-Band-Paket.

Back Orifice 2000

Nach den Definitionen in [Fiolka 01] handelt es sich bei der Software Back Orifice 2000 (BO2k) vom Cult of the Dead Cow (CDC), wie auch bei Liquid, weder um ein Trojanisches Pferd noch um ein Back Door, da Back Orifice 2000 weder eine nützliche Funktionalität vortäuscht noch bei der Programmierung einer Software vergessen wurde, ein Schlupfloch zu schließen. Laut den eigenen Angaben des CDC auf der Website www.backorifice2000.com handelt es sich hierbei um ein Tool zur Netzwerkadministration. Auch wenn dieses das beabsichtigte Einsatzgebiet der Software sein sollte, so wird sie doch aufgrund ihrer vielseitigen Funktionalitäten sehr häufig von Crackern als Einbruchstool genutzt. Denn Back Orifice 2000 bietet hierfür so begehrte Funktionen, wie die Aktivierung einer Tarnkappe (die Serverinstallation auf dem Opfersystem kann als Dienst eingerichtet werden, so dass Back Orifice 2000 weder auf den ersten Blick, mittels eines Fensters zum Beispiel, sichtbar oder im Task Manager aufgelistet wäre). Des Weiteren kann bei der Erzeugung einer Serverdatei jeder beliebige Dateiname verwendet werden, was das Entdecken dieser Datei bzw. dieses Dienstes erschwert. Ferner wird die Datei automatisch in %systemroot%/system abgelegt, ein für die meisten Anwender wenig beachtetes Verzeichnis. Allerdings besitzt der Back Orifice 2000 Server eine solch markante Signatur, dass er von den meisten Malware-Scanner – zwar als Trojaner – aber immerhin erkannt wird.

Ein wesentlicher weiterer Vorteil von BO2k ist die Möglichkeit für Cracker, frei zwischen den verwendbaren Protokollen UDP oder TCP, sowie den Ports wählen zu können.

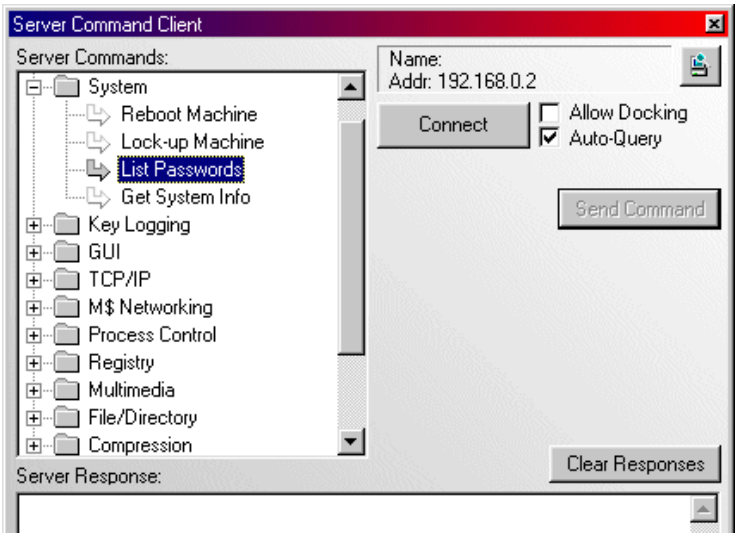


Abb. 24: Back Orifice 2000 Client Interface

Auf der Client-Seite von Back Orifice 2000, also auf dem Rechner des Angreifers, wurden die Funktionen auf der Benutzungsoberfläche (s. Abb. 24) gruppiert. Eine Auswahl davon wird im Folgenden beschrieben:

SYSTEM

In der Gruppierung „System“ befinden sich die ersten Funktionen, die bei Crackern besonderes Interesse hervorrufen. Mit den Funktion „Reboot Machine“ und „Lock-up Machine“ lässt sich die Verfügbarkeit des angegriffenen Rechners reduzieren. Die erste der beiden Funktionen führt dabei einen Neustart durch, ohne dass ein Anwender die Möglichkeit hat, ungespeicherte Daten auf dem angegriffenen Computer zu speichern. „Lock-up Machine“ führt zwar keinen Neustart durch, dafür werden aber auch keine Maus- und Tastatureingaben mehr von dem angegriffenen PC akzeptiert. Lediglich ein Neustart über den Reset-Schalter macht das System für einen Anwender wieder nutzbar. Dabei gehen allerdings wieder alle Informationen verloren.

Ebenfalls Teil der Gruppe „System“ ist der Befehl „List Passwords“. Damit wird versucht alle üblichen Speicherstellen für Passwörter auszulesen (zum Beispiel für Windows- bzw. Netzanmeldung und Bildschirmschoner). In Kombination mit der Funktion „Get System Info“ erlangt ein Angreifer auch noch weitere sensitive Informationen über das angegriffene System, wie zum Beispiel Rechnername, momentaner Anwender, Prozessorklasse, Betriebssystem, Arbeitsspeichergröße (und derzeitige prozentuale Belegung) mit Auslagerungsdateigröße und Ausnutzung sowie alle verfügbaren Laufwerke mit Kapazitäts- und Formatierungsangaben.

Allein mit den durch die Funktionsgruppe System erlangten Möglichkeiten und Informationen ist ein Hacker bestens vorbereitet um sich in einem Netzwerk als beides – den ausspionierten Computer und Nutzer – auszugeben.

KEY LOGGING

Die Funktionen in der Gruppe „Key Logging“ bieten einem Angreifer die Optionen:

- einen Prozess zu starten („Log Keystrokes“), der alle Tastatureingaben in einer Datei auf dem so angegriffenen Computer zu protokollieren,
- diesen Prozess später zu beenden („End Keystroke Log“),
- die gespeicherte Log-Datei einzusehen („View Keystroke Log“) und
- zur Verschleierung wieder zu löschen („Delete Keystroke Log“).

Wenn der Angriff längere Zeit unbeobachtet bleibt, kann ein Hacker über diese Gruppe von Funktionen auch Benutzernamen und Passwörter für Anwendungen auslesen, die er über „List Password“ nicht bekommen konnte. Des Weiteren können hierüber auch sicherheitsbedürftige

Informationen erhalten werden, bevor diese verschlüsselt werden, da die Eingabe dieser Informationen in der Regel im Klartext durchgeführt wird.

GUI

Die Funktion „System Message Box“ ist aus technischer Sicht kein wirklich gefährlicher Angriff, da lediglich eine Meldung mit beliebigem Text auf dem Server-PC erzeugt wird. Eine Message Box kann nicht nur unangenehm und verwirrend sein, sondern auch zu schweren Belästigungen genutzt werden. Ähnliches gilt für die in „Multimedia“ zusammengefassten Funktionen.

TCP/IP

Unter „TCP/IP“ werden Back Orifice 2000 Funktionen zusammengefasst, die sich hauptsächlich um die Zuordnung von Ports zu Diensten kümmern. Dadurch können Ports für Angriffe missbraucht werden, die eigentlich für einen anderen Dienst in der Firewall freigegeben worden sind.

M$ NETWORKING

Die „M$ Networking“-Gruppe stellt im Wesentlichen NetBIOS-Funktionalitäten zur Verfügung. So können Verzeichnisse als Shares eingerichtet und verbunden werden. Zusätzlich bietet die Funktion „List Shares on LAN“ die Möglichkeit, vom Back Orifice 2000-Server alle im LAN verfügbaren Shares aufgelistet zu bekommen.

PROCESS CONTROL

Die Funktionen „List Processes“, „Kill Process“ und „Start Process“ gehören zur Gruppe „Process Control“. Sie erlauben es, Prozesse zu starten und die Aktiven zu beenden. Damit kann ein Hacker ein System nicht nur gänzlich oder partiell außer Gefecht setzen, sondern auch andere für ihn nützliche Prozesse und Dienste starten.

REGISTRY

Laut Microsoft ist die Registry eine Datenbank, „die Informationen über die Konfiguration des Computers enthält“ (aus der Windows 2000 Hilfe). Alle in dieser Back Orifice 2000 Gruppe befindlichen Funktionen dienen somit auch der Manipulation dieser wichtigen Datenbank.

FILE/DIRECTORY

In der Gruppe „File/Directory“ findet der Anwender von BO2k alle für das Dateimanagement üblichen Befehle. Sowohl das Auflisten, Finden, Löschen, Anzeigen, Verschieben, Umbenennen und Kopieren von Dateien, als auch das Erstellen und Löschen von Verzeichnissen, sowie das Ändern von Dateiattributen und das Senden und Empfangen von Dateien über das Netz sind in dieser Gruppe als Funktionen enthalten.

Damit lassen sich die Dateien auf dem angegriffenen Computer manipulieren. Es können bei einem Angriff auf eine Firma zum Beispiel deren Assets gelöscht oder ausgelesen werden. Wenn zum Beispiel die ganze Kundendatenbank einer Abteilung oder nur eines einzigen Vertriebsbeauftragten verschwinden oder bekannt werden, ist die Sicherheitspolitik verletzt.

Zusammenfassend lässt sich sagen, dass Back Orifice 2000 ein für unerfahrene Hacker sehr mächtiges Tool ist, das aufgrund seiner Funktionen eher Malware als ein Netzwerkadministrationswerkzeug ist.

NetBus Pro 2.10

Die von uns eingesetzte Version 2.10 von NetBus Pro wird laut diversen Webseiten, die über die Möglichkeiten von Hacker-Toolz (zum Beispiel http://home.t-online.de/home/Gerhard.Glaser/) berichten, als genauso gefährlich eingestuft wie Back Orifice 2000.

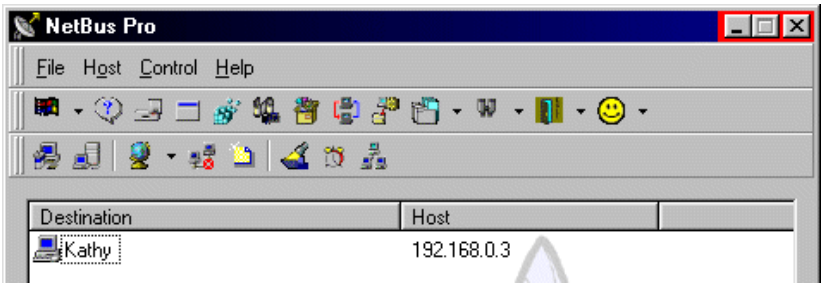


Abb. 25: NetBus Pro 2.10 Client Interface

Im direkten Vergleich mit BO2k hat NetBus Pro 2.10 eine anwenderfreundlichere Benutzungsoberfläche. Die Funktionen sind im wesentlichen die gleichen wie bei Back Orifice 2000, lediglich die Tarnkappe bietet bei BO2k mehr Möglichkeiten, dafür verfügt NetBus Pro 2.10 noch über die Funktionen zum Öffnen und Schließen des CD-ROM-Laufwerks und ähnlicher für Hacker-Angriffe wenig relevanter Zusätze.

5.3 VERSUCHSABLAUF OHNE SCHUTZMECHANISMEN

Bei den Vorbereitungen für diesen Versuch, bei dem weder der Computer Kathy noch das Netz über irgendwelche Schutzmaßnahmen verfügte, erzeugten wir eine Back Orifice Server-Datei. Dabei haben wir bewusst auf manche der Möglichkeiten verzichtet. Wir wählten eine Back Orifice 2000 Server Konfiguration, die nicht automatisch beim Systemboot gestartet wurde, sondern als nicht getarnter Dienst lief. Dies ermöglichte es uns zu überwachen, ob der Server erfolgreich gestartet wurde. Ähnlich verfuhren wir bei dem NetBus Pro-Server.

In diesem, wie in allen folgenden Versuchen, wurden die Server-Dienste auf Kathy bewusst und anwendergesteuert gespeichert, anstatt Dropper zur Platzierung der Serverdatei oder Autostartmechanismen zu entwickeln oder einzusetzen, zumal unser Fokus auf das Testen der Firewall und des ID-Systems lag.

5.3.1 VERSUCHSDURCHFÜHRUNG

Bei der Versuchsdurchführung hielten wir uns an die Reihenfolge der Aufgaben (s. 5.2.2). Also führten wir zunächst einen IP-Scan auf dem Adressen-Band von 192.168.1.1 bis 192.168.1.254 durch. SuperScan 3.00 meldete zwei Computer, die reagierten: 192.168.1.1 (Kathy) und 192.168.1.2 (Ergo). Da Kathy Opfer der Angriffe werden sollte, interessierte uns nur die Ergebnisse des Portscans von 1 bis 65535 auf Kathy's IP-Adresse. Es wurden die offenen Ports 139 (NetBios Session Service), 20034 (NetBus Pro) und 54320 (Back Orifice 2000) gefunden.

Der Versuch, den NetBus Pro 2.10 Client zu starten, schlug auch nach einigen Rekonfigurationen fehl. Unter der Prämisse, dass wir auch nur Angriffe von DAUs simulieren wollten, brachen wir die Angriffe über NetBus Pro 2.10 ab und wandten uns Back Orifice 2000 zu.

Mit Back Orifice nutzten wir als erstes die Funktion „List Passwords". Die verschiedenen vorgesehenen Felder waren teilweise leer oder mit Standards belegt. In einem dieser Felder fanden wir aber das fast richtige Passwort wieder. Wir stellten fest, dass trotz eines neunstelligen Passworts nur ein achtstelliges angezeigt wurde, bei dem die neunte Stelle fehlte. Da einem Hacker ein Passwort alleine nicht hilft, brauchten wir noch den Benutzernamen, den wir über die Funktion „Get System Info" neben einigen anderen Informationen auch erhielten.

Als nächstes versuchten wir, die Dateien to_be_changed.txt und to_be_changed2.txt ausfindig zu machen. Dazu nutzten wir weiterhin Back Orifice 2000. Mit dem Befehl „List Directory" und dem Parameter „c:\" wurde eine Auflistung aller Verzeichnisse und Dateien unterhalb von Root angezeigt. Da sich die genannten Dateien dort befanden, war ein weiteres Suchen nicht notwendig. Die gefundenen Dateien wurden sodann manipuliert, dass heißt die Zweite wurde über den Befehl „Delete File" ersatzlos gelöscht. Die Datei to_be_changed.txt wurde zuerst eingesehen, dann ebenfalls gelöscht und daraufhin durch eine Kopie der autoexec.bat mittels des Befehls „Copy file" ersetzt.

Um die Verfügbarkeit des Systems zu reduzieren, untersuchten wir mehrere Möglichkeiten. Als erstes nutzen wir die Back Orifice 2000 Funktion „Lock-up Machine". Es wurden die Eingabemöglichkeiten über Tastatur und Maus blockiert, so dass trotz dargestelltem Desktop keine Nutzung des Rechners durch einen Anwender am Gerät möglich war. Die Verfügbarkeit des

Systems war aber nicht gänzlich eingeschränkt, da der Computer weiterhin auf ICMP-Echo-Anfragen reagierte. Lediglich ein Neustart hob die Blockierung auf. In einem frühren Test des gleichen Befehls stellten wir bei unserem Gateway Ergo fest, dass der Computer trotz des Lock-Up noch seine Gateway-Funktion wahrnehmen konnte. Aufgrund dieser Erfahrung nahmen wir die Aufgabe 7) auf.

Die Reduktion der Verfügbarkeit auf Null erreichten wir kurzfristig mit dem Back Orifice 2000 Befehl „Reboot Machine“. Nach Übermittlung des Datenpakets, dass durch diesen Befehl versandt wurde, wurde Kathy neu gestartet, ohne dass eine Speicherung des Zustandes und der bearbeiteten Dokumente möglich war. Da das System aber wieder neu hochgefahren wurde, konnte mittels der „Reboot Machine“-Funktion keine permanente Unverfügbarkeit des Systems hergestellt werden. Um dieses Ziel zu erreichen, gaben wir den Ping of Death ein: ping -l 65510 192.168.1.1. Das Betriebssystem des Computers Felix, von dem aus wir alle Angriffe durchführten, verweigerte die Ausführung des Befehls mit dem Hinweis „Ungültiger Wert für die Option -l“.

```
   --> Version: Back Orifice 2000 (BO2K) v1.1

Passwords cached by system:
Cached Passwords:
Resource: ' ^•Ð•Må ñ%'  Password: 'pokemon9'
Resource: 'MAPI'  Password: 'MAPI'
End of cached passwords.
Unable to read value 'ScreenSave_Data'.
System info for machine 'KATHY'
Current user: 'Administrator'
Processor: I586
Win32 on Windows 95 v4.0 build 950
Memory: 127M in use: 11%  Page file: 384M free: 384M
C:\ - Fixed Sec/Clust: 16 Byts/Sec: 512,  Bytes free: 403496960/531324928
D:\ - CD-ROM
End of system info
Contents of directory 'c:\':
[…]
                  153 -A----- 05-18-2001 15:59 AUTOEXEC.BAT
[…]
TO_BE_~1.TXT       153 -A----- 05-18-2001 15:59 to_be_changed.txt
TO_BE_~2.TXT       106 -A----- 05-21-2001 14:05 to_be_changed2.txt
    BO2K.EXE    167936 -A----- 05-15-2001 17:23 bo2k.exe
861432 bytes in 24 files.
File deleted.
File copied.

Locking up machine
[Don't expect much to work after this!]
```

Abb. 26: gekürztes Back Orifice 2000 Protokoll

Als letzte Technik versuchten wir mit WinNuke bzw. Liquid das Opfersystem zu überfordern. Dies gelang mühelos. Nachdem die vier Datenpakete bei Kathy eingegangen waren und eine Verarbeitung versucht wurde, reagierte das Betriebssystem mit einem BSoD: „Das System ist ausgelastet oder instabil. Sie können warten, bis es verfügbar wird, oder den Computer neu

starten.“ Der angegriffene Computer erholte sich nicht von den Out-of-Band-Datenpaketen und reagierte auch nicht mehr auf Pings oder sonstige Netzanfragen. Er war somit außer Gefecht gesetzt. Nur ein Systemneustart stellte die Verfügbarkeit des Computers wieder her.

5.3.2 VERSUCHSAUSWERTUNG

Da in diesem Versuch weder eine Firewall noch ein sonstiger Sicherheitsmechanismus eingesetzt wurde, verwundert es wenig, dass alle Aufgaben (s. 5.2.2) zum Teil gleich mehrfach erfüllt werden konnten. Wegen mangelnder Sicherheitssysteme wurden nur solche Angriffe verhindert, bei denen Probleme anderer Art auftraten. Der NetBus Pro 2.10 Client ließ sich leider nicht starten, hätte aber ähnliche Funktionalitäten gehabt wie Back Orifice 2000.

Windows NT 4.0 Workstation mit Service Pack 6a verhinderte zwar das Absenden von langen, in der Definition von ICMP erlaubten Ping-Paketen und schützte so den Computer Kathy, der aufgrund einer mangelhaften ICMP-Implementation angreifbar gewesen wäre. Somit kam der Hersteller des eingesetzten Betriebssystems bereits auf dem Hackercomputer den Angriff zuvor. Im Vergleich dazu verbot Microsoft Windows NT 4.0 Workstation aber nicht das Senden von ebenfalls nach Definition korrekten TCP-Paketen mit gesetztem *Urgent*-Flag, obwohl hierbei ebenfalls eine Lücke der TCP-Implementation in Windows 95 ausgenutzt wurde. Beide Sicherheitslücken wurden 1997 – drei Jahre vor Erscheinen des Service Release 6a – bekannt.

Aufgrund der Erfahrungen aus diesem Versuch werden wir in den folgenden Versuchen auf den Ping of Death und auf NetBus Pro 2.10 verzichten.

5.4 VERSUCHSABLAUF MIT FIREWALL

Im Folgenden wird der Versuchsablauf der 2. Stufe „mit Firewall“ beschrieben. Hier soll gezeigt werden, welche Schutzmassnahmen die Firewall bietet.

5.4.1 KONFIGURATION DER FIREWALL

Wie in 5.2.1 stehen uns lediglich drei Rechner zur Verfügung. Der dual-homed Host ERGO hat hier die gesamte Firewallfunktionalität zu übernehmen. Auf ist die Gauntlet Firewall Version 5.5 for Windows NT von Network Associates, Inc. installiert. Dieses auf dem TIS Firewall Toolkit basierende Produkt beinhaltet verschiedenartige Proxyserver und einen integrierten Paketfilter. Daher können mit diesem Produkt die beiden in Kapitel 3.2 besprochenen Arbeitsweisen, die Paketfilter und die Proxy, getestet werden

Neben weiteren Eigenschaften, u. a. Werkzeuge zur Authentikation und zur Benutzung von Public Key Infratructures (PKI), unterstützt Gauntlet noch weitere Netzwerkkarten. Durch diese ist es möglich, die Firewalltopologie durch ein Perimeter-Netz zu erweitern.

Über die im Szenario beschrieben Beschränkungen wird im Folgenden ein Überblick gegeben. Dies gibt einen groben Überblick über die vom Verbund gegeben Sicherheitspolitik:

- E-Mail soll erlaubt werden,
- Die Finanzanwendung soll gestattet sein
- Surfen im WWW möglich
- Webserver sollen erlaubt werden
- Keine weiteren Dienste sind erlaubt.

Diese Beschränkungen sollen nun in Regeln umgesetzt werden.

Zunächst werden einige für die Netzwerksicherheit notwendige Regeln implementiert. Dies betrifft die allgemeine Gefahr des Spoofing und die Schwächen des ICMP-Protokols.

Spoofing

Damit ein Host aus dem externen Netz (gelbes Testnetz) unter der Vorgabe, ein Host des internen Netzes (blaues Testnetz) mit einer IP-Adresse 192.168.1.*/16 zu sein, sich nicht mit einem Host des internen Netzes verbinden kann, müssen Vorkehrung getroffen werden.

Diese Vorkehrung wird bei der Einstellung der in Ergo installierten Netzwerkadapter in Gauntlet getroffen. Hier wird die Netzwerkkarte des blauen Testnetzes der „trusted policy“ und die des externen Netzes der „untrusted policy“ zugewiesen. Gauntlet benennt diese Netzwerkkarten „trusted interface“ und „untrusted interface“.

Versucht ein Host, der einer IP-Adresse des internen Netzes hat, sich über das untrusted Interface mit einem Computer des internen Netzes (blaues Testnetz) zu verbinden, so erkennt Gauntlet dies als Spoofing.

Eine weitere Vorkehrung wurde zudem bei der Implementation der Paket-Filter-Regeln, hier Paket Screening Rules genannt, getroffen. Gauntlet verfolgt hier die Grundhaltung des generellen Verbots. In den Erlaubnis-Regeln sind die IP-Adressen der Kommunikationspartner explizit angegeben. Dies bedeutet, dass kein Host außer Felix sich mit Kathy verbinden darf, sofern überhaupt eine Verbindung von außen gestattet ist.

ICMP

Auch gegen die in Kapitel 2.4.2 beschriebenen Schwächen des ICMP-Protokolls müssen in der Firewall-Regeln implementiert werden. Dies geschieht über die Packet Screening Rules der Firewall. Die folgenden Regeln wurden in der hier beschriebenen Reihenfolge implementiert.

Im Folgenden gilt die Annahme, dass ein Nutzer des internen Netzes auf Rechner im externen Netz zugreifen möchte, um zum Beispiel eine Homepage zu betrachten. Dabei soll es möglich sein, sich die evtl. Präsenz eines Hosts im externen Netz mittels des Ping Befehls zu informieren.

Dazu ist es nötig, Kathy die Sendung des ICMP-Echo-Befehls an Felix zu gestatten. Nach dem Least-privelege Prinzip wird hier nicht mehr gestattet als nötig ist. Daher wird Kathy nur das senden des ICMP Message Type 8 (Echo request) gestattet. Dies ist mit folgender Filterregel implementiert:

Rule Name	PING to Felix
Protocol	ICMP
TCP/ICMP-Flags	ECHO
Interface	Trusted Interface
Screen Action	Forward Traffic with reply
Source Adress	192.168.1.1/16
Source Port range	-
Destination Adress	192.168.0.1/16
Destination Port range	-

Abb. 27: Filterregel „PING to Felix“

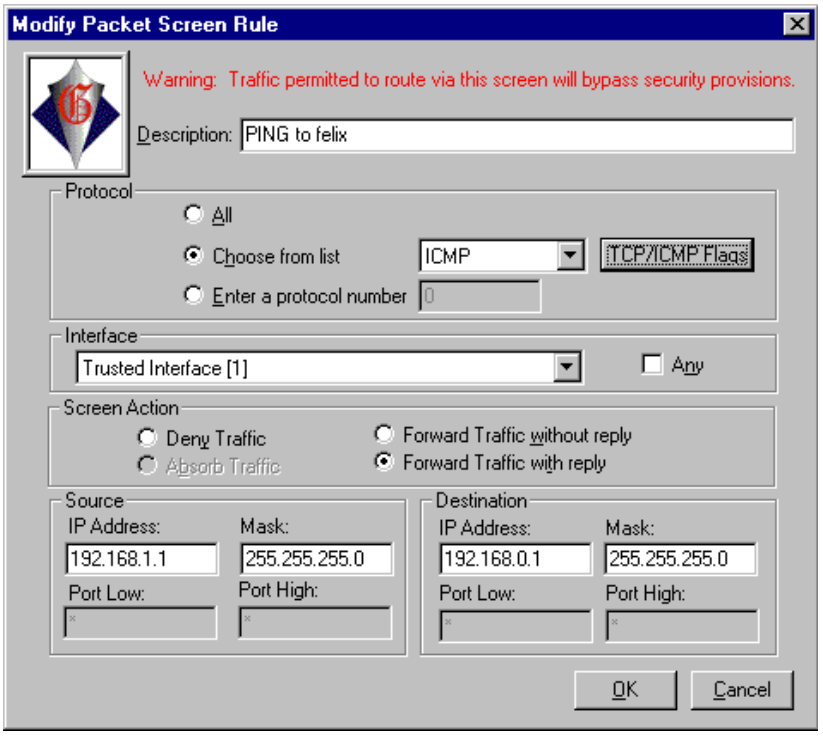


Abb. 28: Erstellung einer Filterregel mit Gauntlet

Um eine Aussage über die Verbindung zu Felix machen zu können, muß Kathy eine Antwort erhalten. Dafür sind im ICMP Protokoll die Message Types 0 (Echo reply), 3 (Destination unreachable) und 11 (Time to live exceeded) vorgesehen. Daher wird folgende Regel implementiert:

Rule Name	PING to Kathy
Protocol	ICMP
TCP/ICMP-Flags	ECHOREPLY or TIMXCEED or UNREACH
Interface	Untrusted Interface
Screen Action	Forward Traffic with reply
Source Adress	192.168.0.1/16
Source Port range	-
Destination Adress	192.168.1.1/16
Destination Port range	-

Abb. 29: Filterregel „PING to Kathy“

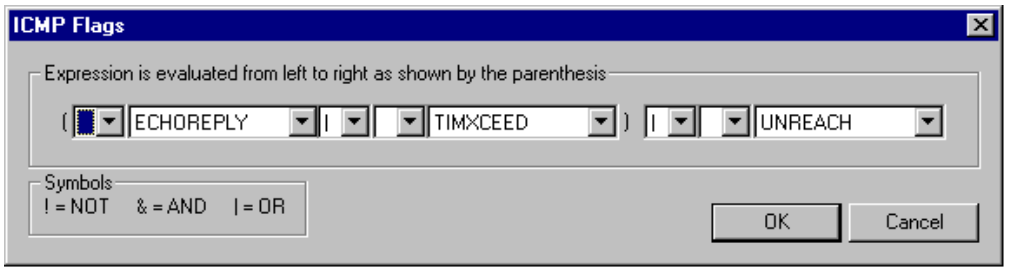


Abb. 30: Einstellung der ICMP Message Types (hier: ICMP Flags) mit Gauntlet

Die weiterhin erlaubten Dienste sollten nicht über eine Paketfilterregel. der Einsatz des Paketfilters wird in der Dokumentation nicht empfohlen. Alle Protokolle, die von einem Paketfilter betrachtet werden, werden nicht an die Proxies geleitet. Wie in Kapitel 3.2.3. besprochen, erlaubt ein Proxy die genauere Betrachtung eines Protokolls. So kann nach Inhalt gefiltert werden und die Benutzung des Applikationsprotokolls genauer ausgezeichnet werden.

Damit jedoch ein Protokoll von den Proxies, muss der Verkehr des darunter liegenden TCP-Protokolls absorbiert werden. Dies wird mit der letzten Paketfilterregel umgesetzt:

Rule Name	TCP absorb all
Protocol	TCP
TCP/ICMP-Flags	-
Interface	Any
Screen Action	Absorb traffic
Source Adress	*
Source Port range	*
Destination Adress	*
Destination Port range	*

Abb. 31: Filterregel „TCP absorb all“

Damit die Finanzanwendung benutzt werden kann, muss es einen Proxy geben, der die Verbindung auf diesen Port erlaubt. Daher wurde ein eigener Proxy definiert, der eine Verbindung auf diesen Port zulässt. Dies wurde mit folgender Einstellung umgesetzt:

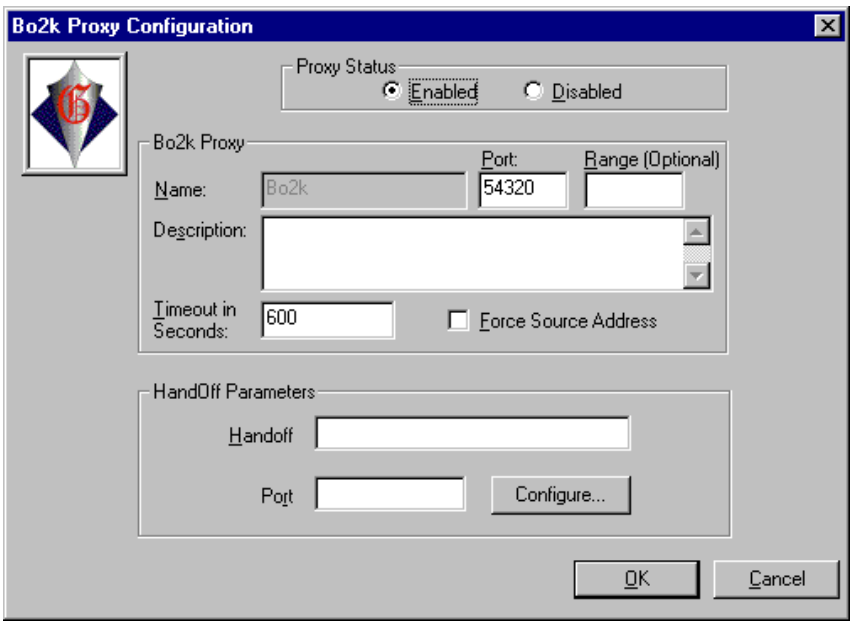


Abb. 32: Ein Proxy für die Finanzanwendung

Hierbei handelt es sich um einen benutzerdefinierten Proxy, der außer der Möglichkeit zur Verbindung auf einen bestimmten Port keine weiteren Applikationsfilter Eigenschaften aufweist. Die Eigenschaften eines Applikationsfilter sind lediglich bei den von Gauntlet mitgelieferten Proxies für die Standardprotokolle wie HTTP gegeben. Da wir im Szenario den Zugriff auf den Webserver beschrieben haben, wurde auch ein HTTP-Proxy eingerichtet. Dieser brauchte lediglich aktiviert werden. An den Standardeinstellungen wurde nichts geändert. Hierbei zeigen sich die Eigenschaften des Proxy auf der Applikationsebene. So konnte die in der Zugriffskontrolle enthaltene Authentikation eingestellt und Content wie Java, JavaScript und ActiveX aufgefiltert werden. Zudem konnten mehr als ein Port angegeben werden.

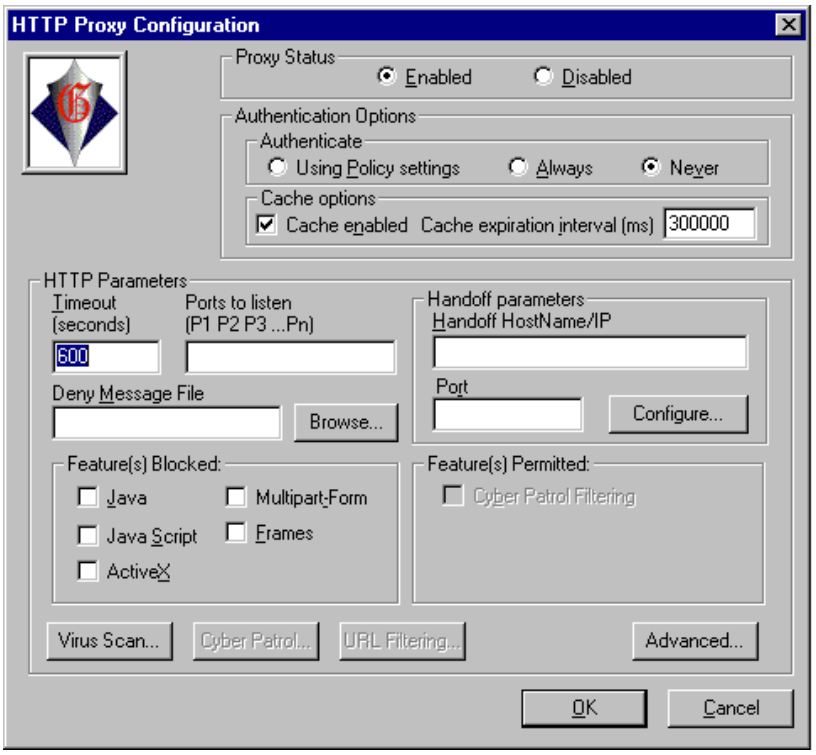


Abb. 33: Die Einstellungen des HTTP-Proxy

Zum Test dieses Proxies wurde der Web-server Mywebserver auf Kathy installiert und versucht, von einem auf Felix installierten Browser eine Verbindung herzustellen. Der Webserver bekam eine Anfrage und schickte die erste Datei als Antwort. Doch diese kam auf dem Rückweg nie durch die Firewall durch. Da das HTTP-Protokoll für keinen der Angriffe relevant ist, wurde nicht mehr getestet.

Auch der SMTP-Proxy wurde aktiviert. Dieser bietet ebenfalls die Möglichkeit des Content-filtering. Mangels eines E-Mail-Servers und der fehlenden Relevanz in den Versuchs-anforderungen wurde dieser nicht getestet.

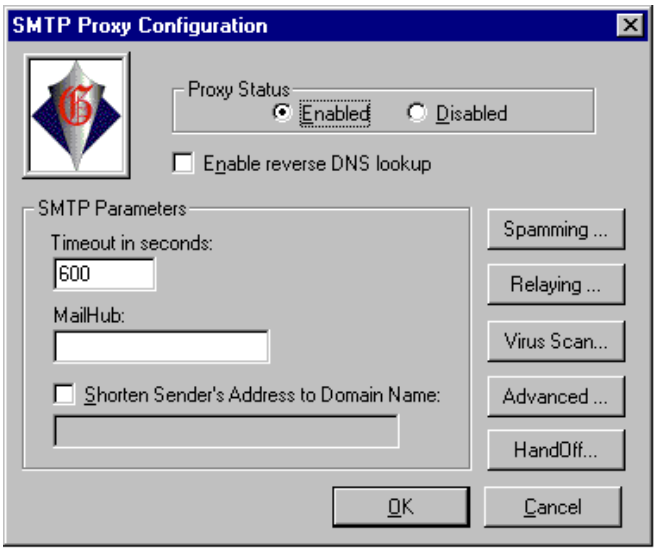


Abb. 34: Die Einstellungen des SMTP-Proxy

Gemeinsam ist allen Proxies, dass diese statisch sind. Es konnten keine Entscheidungen auf Grund vorheriger Ereignisse eingestellt werden. Des Weiteren war es nicht Möglich, ein Anwendungsprotokoll auf einen anderen Port zu leiten.

Die Proxies wurden einer Policy zugeordnet. Dabei steht die „untrusted policy“ für die Politik, die auf das „untrusted interface“ angewendet wird. Das gleiche gilt für die „trusted policy“.

Da sowohl im WWW gesurft werden soll, als auch ein Webserver hinter der Firewall laufen soll, müssen Verbindungen für das HTTP-Protokoll in beide Richtungen aufgebaut werden können. Gleiches gilt für den SMTP-Proxy, da E-Mails in beide Richtungen versandt werden sollen, und dem Proxy für die Finanzsoftware, da hier sowohl Rechnungen von den Wasch-zentren als auch Umsätze von der Verwaltung abgerufen werden können. Daher wurden diese Proxies beiden Policies zugefügt.

5.4.2 VERSUCHSPROTOKOLL DES ZWEITEN VERSUCHSABSCHNITTS

Bei der Durchführung der zweiten Versuchsphase führten wir die Angriffe nach Zwischenschaltung der Gauntlet-Firewall durch. Für den Angriff verwendeten wir dieselbe Angriffssoftware wie in der ersten Phase:

- Das Portscan-Programm „SuperScan“
- Back Orifice 2000
- Liquid, ein WinNuke-Programm

Die dabei getesteten Angriffsziele waren die gleichen, wie auch schon in der ersten Versuchsphase. Der Vollständigkeit halber wollen wir sie hier noch einmal aufführen:

- Die Durchführung eines Portscans auf dem gesamten Band
- Das Ausspionieren von Passwörtern auf dem angegriffenen Rechner.
- Das Auffinden und Löschen einer bestimmten Datei
- Das Stehlen einer Datei (Datenspionage)
- Das Ändern einer Datei auf dem angegriffenen Rechner
- Die Reduzierung der Verfügbarkeit der angegriffenen Maschine durch Reboot

Als zusätzliches Angriffsziel für die zweite Phase modellierten wir einen Angriff auf die Firewall selbst, da viele Angreifer in der Realität versuchen, eine Firewall vor einem Netz außer Gefecht zu setzen, um so Angriffe auf das eigentliche Netz leichter durchführen zu können. Es blieb nun noch die Aufgabe, einen Proxy für die Finanzbuchhaltung des Schnellwaschzentrums einzurichten. Wir verwendeten den Plug In Proxy der Gauntlet Firewall und konfigurierten ihn so, dass er Anfragen auf Port 54320 bedient. Dabei handelt es sich um den Port, den auch Back Orifice 2000 als Standardport verwendet. Wir wählten diese Einstellung, um eine gewisse Unsicherheit der Finanzanwendung zu modellieren.

Die Gauntlet Firewall bietet die Möglichkeit, für Proxyzugriffe aus bestimmten Policies eine Authentisierung mit Passwortabfrage zu verlangen. Dieses Passwort ist muss dann eingegeben werden, bevor die Firewall Zugang zu den Proxies gewährt. Um diese Einrichtung ebenfalls zu testen, führten wir die Angriffe zunächst mit deaktivierter und danach mit aktivierter Passwortabfrage durch.

Wir begannen die Versuche ohne Passwortabfrage mit dem Portscan. Der Portscanner SuperScan wurde so eingestellt, dass er das gesamte Adressband und sämtliche Ports abfragen sollte. Ein abgesetzter Ping-Befehl auf das Angriffsziel wurde von der Firewall gemäß ihrer Kon-

figuration und unserer Sicherheitspolitik abgeblockt und im Log aufgezeichnet (als von den Firewall-Regeln nicht zulässiges ICMP-Paket). Ein Ping-Befehl auf den Gateway-Rechner funktionierte nur dann, wenn er auf das äußere Interface der Firewall abgesetzt war. Ein Ping auf das innere Interface wurde ebenfalls geblockt und protokolliert (beides genau gemäß unserer Planung).

Der eigentliche Portscan verlief bei der Adresse des inneren Interfaces ergebnislos. Das Angriffsziel meldete folgende offene Ports:

- Port 13. Dieser ist für den Daytime Proxy zuständig, der Anfragen auf die Systemuhr ermöglicht. Die Abfrage auf diesen Port meldete die Firewall als zulässig, weil dieser Proxy für die Untrusted Policy freigegeben war.
- Port 21, zuständig für den FTP Proxy. Obwohl der Portscan diesen Port als offen meldete, wurde die Anfrage von der Firewall nicht zugelassen. Gemäß unserer Einstellung war der Filetransfer Proxy für die Untrusted Policy gesperrt.
- Port 23. Dieser Port wird vom Telnet Proxy verwendet. Die Anfrage wurde ebenfalls von der Firewall abgelehnt.
- Port 25, der vom SMTP Proxy verwendet wird. Dieser ist für den e-Mail Verkehr zuständig. Eine entsprechende Anfrage wurde von der Firewall akzeptiert, da der Mailverkehr in beide Richtungen möglich sein sollte.
- Port 54320. Er diente, wie gesagt, für unsere Finanzbuchhaltung. Da in unserem Szenario auch andere Firmen auf dieses Programm zugreifen können sollten, akzeptierte die Firewall eine Anfrage.

Nach dem Portscan verwendeten wir für die nächsten Angriffe Back Orifice 2000. Das Client Programm von Back Orifice 2000 konnte sich mit dem auf dem Angriffsziel installierten Serverprogramm verbinden. Die Firewall protokollierte dies als zulässigen Zugriff auf den Proxy für die Finanzbuchhaltung. Wir erhielten nach dem Verbindungsaufbau von Back Orifice die Meldung „Gathering Server information“, die nicht mehr vom Bildschirm verschwand. Während der gesamten Versuche mit Back Orifice 2000 protokollierte die Firewall immer nur den Verbindungsaufbau und den Verbindungsabbau, niemals jedoch die einzelnen Befehle, die wir mit dem Clientprogramm an den Server schickten.

Der Versuch, mit Back Orifice 2000 das Passwort des Angriffsziels auszuspionieren, scheiterte. Back Orifice 2000 schickte den Befehl zwar an das angegriffene System, erhielt jedoch keinerlei Antwort. Auch ein abgesetzter get system info Befehl, mit dem beispielsweise der Benutzername und das Betriebssystem auskundschaftet werden können, lieferte kein Ergebnis.

Als nächstes war das Auffinden und Löschen einer Datei an der Reihe. Zu diesem Zwecke erstellten wir auf dem Angriffsziel eine Textdatei mit dem Namen „to_be_changed.txt“, die einen kurzen ASCII-Text enthielt. Diese Datei wurde im Verzeichnis c:\ abgelegt. Der Versuch, mit Back Orifice 2000 auf dem Zielrechner nach dieser Datei suchen zu lassen, verlief erfolglos. Die Datei wurde dabei nicht nur einfach nicht gefunden. Vielmehr erhielt der Client von Back Orifice 2000 erneut überhaupt keine Antwort vom Server Programm. Also setzten wir für den nächsten Versuch voraus, dass der Angreifer sich auf dem Zielsystem gut auskenne und verwendeten den Löschbefehl direkt (Der Pfad mit der Datei wurde also als bekannt vorausgesetzt). Dieser Angriff verlief erfolgreich. Die Datei to_be_changed.txt verschwand vom Zielrechner. Allerdings blieb die übliche Rückmeldung des Servers „file deleted“, die wir in der ersten Angriffsphase erhalten hatten, diesmal aus.

Das nächste Ziel des Angreifers war die Datenspionage. Dazu wurde to_be_changed.txt wiederhergestellt, und der Angreifer sollte nun versuchen, die Datei mittels Back Orifice 2000 auf seinen eigenen Rechner zu kopieren. Wiederum wurde der Befehl abgesendet, aber der Server lieferte auch diesmal keine Antwort zurück. Ebenfalls fand die Übertragung der Datei auf den Angriffsrechner nicht statt.

Nach diesem Test sollte der Angreifer versuchen, eine Datei auf dem Zielsystem zu verändern. Wir wählten als Aufgabe, den Inhalt der autoexec.bat in die Datei to_be_Changed.txt zu kopieren. Obwohl der Server auch diesmal keine Rückmeldung lieferte, verlief der Angriff doch erfolgreich: Der alte text in to_be_changed.txt war mit dem Inhalt der autoexec.bat überschrieben worden.

Für den nächsten Angriff, die Reduzierung der Verfügbarkeit, verwendeten wir neben Back Orifice 2000 auch das Programm Liquid. Dieses Programm verwendet das TCP-Protokoll und nutzt eine Schwäche einiger Windows Systeme aus: Es setzt das Urgent Flag im TCP-Paket. Dieses Flag markiert ein Paket als dringende Sendung. Einige Windows Systeme haben einen Fehler, der dazu führt, dass sie bei der Verarbeitung eines solchen Paketes abstürzen. Zunächst aber zum Versuch, die Verfügbarkeit des Zielsystems mit Back Orifice 2000 zu reduzieren. Der Angreifer sollte das Zielsystem neu starten lassen. Der Befehl wurde abgesandt, und der Angriff verlief erfolgreich: Das Zielsystem bootete neu. Die Firewall protokollierte, dass die Verbindung über den Finanzanwendungsproxy über Port 54320 vom Partner (dem angegriffenen Rechner) abgebrochen worden sei. Es gibt noch eine zweite Möglichkeit, ein die Verfügbarkeit eines Systems mit Back Orifice 2000 herabzusetzen. Hierbei handelt es sich um den Lock up Befehl. Er führt dazu, dass das System keine Eingaben über Maus und Tastatur mehr akzeptiert. Allerdings stürzt der Rechner nicht ab, weshalb er immer noch Datenpakate verarbeiten und Routing-

Aufgaben übernehmen kann (Wir fanden die heraus, als wir einen Lock up Befehl auf den Gateway-Rechner absetzten, als auf diesem noch keine Firewall installiert war. Dieser nahm seine Routing-Aufgaben weiterhin war). Auch in der zweiten Phase war der Lock up Befehl auf das Zielsystem erfolgreich. Nach dem Lock up blieb die Back Orifice 2000 Verbindung bestehen, und der Angreifer konnte weitere Angriffe auf das System starten (allerdings alle mit den bisher in dieser Versuchsphase beschriebenen Ergebnissen). Der Angriff mit Liquid verlief dagegen erfolglos. Das TCP-Paket blieb in der Firewall hängen, da wir keine Paketfilterregeln für TCP-Pakete formuliert hatten und das Paket von Liquid auch keinem Format eines der Proxies entspricht. Liquid selber reagierte mit einer Fehlermeldung, das Ziel sei nicht erreichbar. Die Firewall protokollierte einfach ein nicht zulässiges TCP-Paket.

Kommen wir nun zum letzten Angriffspunkt, dem Angriff auf die Firewall selber. Wir begannen wieder mit einem Portscan, diesmal mit dem Ziel, freie Ports auf dem Firewall-Rechner herauszufinden. Der Portscan lieferte keine freien Ports. Die Firewall protokollierte nur Scan-Versuche auf das innere Interface, die Versuche, Ports des äußeren Interfaces zu scannen wurden hingegen nicht im Log vermerkt. Um die Firewall außer Gefecht zu setzen, versuchten wir, die Verfügbarkeit ihres Rechners zu reduzieren. Wir begannen wiederum mit Back Orifice 2000. Beide Angriffe, sowohl der Reboot als auch der Lock up, verliefen erfolglos, weil Back Orifice sich nicht mit dem Server verbinden konnte. Liquid meldete einen Erfolg an den Angreifer, aber der Firewall-Rechner arbeitete trotzdem tadellos weiter. Somit waren die Angriffe auf die Gauntlet Firewall gescheitert.

Als letzten Punkt testeten wir die angesprochene Authentisierung von Benutzern. Wir richteten einen Benutzer mit dem Namen „root“ unter Gauntlet ein und gaben im zunächst ein sehr einfaches Passwort, dass nicht den Sicherheitsnormen für Passwörter entspricht: „geheim“. Mit dieser Authentisierung schützten wir den Zugriff auf die Proxies über die Untrusted Policy. Da nur die Proxies geschützt wurden, verwendeten wir in dieser Testreihe nur Back Orifice 2000 als Angriffstool, weil es als einziges von den Proxies der Firewall Gebrauch macht. Back Orifice konnte sich trotz dieser Authentisierung problemlos mit dem Serverprogramm auf dem Zielsystem verbinden. Die Passwortabfrage tauchte überhaupt nicht auf. Um die Möglichkeit eines in Back Orifice 2000 eingebauten Password-Guessers (dabei handelt es sich um ein Programm, das automatisch bestimmte einfache Passwörter „ausprobiert“) auszuschließen, verwendeten wir nun als Passwort eine Folge von Buchstaben, Zahlen und Sonderzeichen größerer Länge. Jedoch auch in diesem Fall tauchte die Passwortabfrage beim Verbindungsaufbau nicht auf. Folglich lieferten alle Angriffsversuche mit Back Orifice 2000 trotz aktiviertem Passwortschutz mit den gleichen Ergebnissen wie ohne Passwortschutz. Gauntlet bietet noch eine sicherere Authentisierungs-

möglichkeit über PKI, also über digitale Benutzerzertifikate und digitale Signaturen, aber ein umfangreiches Testen dieser Einrichtung würde den Rahmen dieses Versuchs sprengen.

Ein weiteres erwähnenswertes Phänomen trat am Ende der Versuchsreihe auf, als wir den Firewall-Rechner herunterfuhren: Für einige Momente war die Gatewayfunktionalität des Rechners noch gegeben (er leitete also Pakete weiter), aber diese wurden nicht mehr von der Firewall gefiltert. Dieses Phänomen trat auch beim Hochfahren des Gateways auf.

5.4.3 AUSWERTUNG DES ZWEITEN VERSUCHSABSCHNITTS

Der zweite Versuchsabschnitt hat gezeigt, dass eine Firewall ein hilfreicher (und nötiger) Schutzmechanismus ist. Er hat aber auch ganz deutlich die Schwächen aufgezeigt, die eine Firewall hat, besonders, wenn man diese alleine zum Schutz eines Netzes einsetzt. Im Folgenden werden wir die aus der zweiten Versuchsstufe gewonnenen Erkenntnisse aufführen und Schlussfolgerungen daraus ziehen.

Unsere Firewall arbeitete mit dem Prinzip des generellen Verbots. Mit diesem Prinzip bestand direkt nach Installation der Firewall keinerlei Gefahr für das interne Netz, da die Firewall keinerlei Netzverkehr erlaubte und somit kein Angriff erfolgen konnte. Nun musste die Firewall so konfiguriert werden, dass sie den dem Szenario entsprechenden Netzverkehr zuließ. Da wir als Port für unsere Finanzanwendung denselben Port wie für Back Orifice 2000 vorsahen (Port 54320 des TCP Protokolls), stellten wir die Firewall und ihre Konfigurierbarkeit absichtlich vor ein Problem. Wenn man im Paketfilter alle TCP Pakete an Port 54320 durchließ oder einen Proxy für die Finanzanwendung vorsah, so konnten sowohl die Pakete für die Finanzanwendung wie auch die für Back Orifice 2000 passieren. Das lag daran, dass weder der Paketfilter noch der Plug-In-Proxy eine Möglichkeit hatten, Pakete der Finanzanwendung von denen für Back Orifice 2000 zu unterscheiden. Die zeigt ganz deutlich ein Problem der Konfigurierbarkeit unserer Firewall: Wenn die Entscheidung, ein Paket zu blocken oder durchzulassen, nur aufgrund von Adressen, Protokoll und Portnummer getroffen werden kann, dann ist die Konfigurierbarkeit nicht granular genug. Für unser Szenario wäre eine Inhaltliche Prüfung der Pakete nötig gewesen. Die Firewall protokollierte in ihrem Log alle Verbindungen von Back Orifice 2000 mit dem Angriffsziel. In unserem Szenario könnte nach Auftreten eines Schadens also immerhin das Log ausgewertet werden, um beispielsweise Pakete vom Angriffsrechner in Zukunft zu blocken. Diese Maßnahme wäre aber nur ein minimaler Schutz, da der Angreifer seine Adresse leicht ändern könnte.

Immerhin reduzierte die Firewall die Gefährlichkeit von Back Orifice 2000, indem sie alle Rückmeldungen an den Server blockte. Dennoch kann der durch die Firewall im Alleineinsatz gebotene Schutz gegen Back Orifice 2000 nur als unzureichend angesehen werden. Als Konsequenz resultiert daraus, dass die Firewall mit weiteren Schutzmechanismen verstärkt werden muss. Dabei könnte es sich beispielsweise um ein Intrusion Detection System handeln oder aber um eine geeignete zweite Firewall, die mit unserer zusammen als Bastion (siehe Abschnitt 3.3.) eingesetzt wird um den Satz der möglichen Regeln zu ergänzen.

Im Gegensatz zu den Back Orifice 2000 Attacken konnte die Firewall die Angriffe mit Liquid problemlos blocken. Das lag daran, dass Liquid einen TCP Port verwendet, der in keiner unserer Regeln freigegeben war. Die Firewall funktioniert also hervorragend, wenn die von der Angriffssoftware verwendeten Pakete den Paketen für den normalen Netzverkehr nicht zu ähnlich sehen. Dies ist vor allem dann problematisch, wenn der Angreifer die regelmäßig verwendeten Ports kennt und seine Angriffssoftware auf genau diese Ports (und eventuell sogar auf entsprechende Protokolle) umkonfigurieren kann. Diese Schwachstelle war bei unserem Versuch besonders gravierend, da der Angreifer erfolgreich einen Portscan auf das Angriffsziel ausführen konnte. Dadurch war es besonders leicht, den Port für die Finanzanwendung zu erfahren und die entsprechenden Einstellungen an der Angriffssoftware vorzunehmen.

Trotz der angesprochenen Probleme bot die Firewall einen gewissen Schutz. Sie verhinderte alle Rückmeldungen an den Angreifer, so dass dieser niemals in Erfahrung bringen konnte, ob seine Angriffe erfolgreich waren. Zudem wurden dadurch einige Angriffsziele komplett vereitelt, etwa die Datenspionage oder das Ausspähen von Passwörtern.

Bemerkenswert war die Fähigkeit der Firewall, sich selbst zu schützen. Kein einziger Angriff auf den Firewall-Rechner war erfolgreich. Dadurch konnte sich der Angreifer die Firewall nicht aus dem Weg räumen. Leider wurden die Angriffsversuche auf den Firewall-Rechner bis auf den Portscan des inneren Interfaces nicht von der Firewall im Log aufgezeichnet. Dies ist eine Schwachstelle, da der Angreifer so nicht bemerkt werden kann und eventuell später mit „geeigneterer“ Angriffssoftware einen neuen Angriff starten kann.

Welchen Schluss kann man nun aus diesen Erkenntnissen ziehen? Die Firewall ist ein nötiger Mechanismus, um Netze zu schützen. Dennoch gerät eine Firewall selbst in relativ einfachen Szenarien leicht an die Grenze ihrer Konfigurierbarkeit. Auch wenn eine drohende Gefahr dem Administrator bekannt ist, kann die gebotene Feinheit in den Regeln eventuell nicht ausreichen, um völligen Schutz zu gewähren. Deshalb muss die Firewall auf jeden Fall entweder

- mit einer größeren Feinheit der Regeln ausgestattet werden, was natürlich oft nicht möglich ist,
- mit einer anderen Firewall zusammen eingesetzt werden, um diese Feinheit zu erreichen,
- oder zusammen mit einem geeigneten IDS eingesetzt werden.

Unser Versuch verfolgt den dritten Ansatz. Die Ergebnisse dieses Versuchsabschnitts und die dabei immer noch auftretenden Probleme behandelt der nächste Abschnitt.

5.5 VERSUCHSABLAUF MIT FIREWALL UND IDS

Als nächster und letzter Versuch wird das Intrusion Detection System in der Firewall getestet. Hierfür wird die Firewall rekonfiguriert. Die Charakteristika des IDS werden kurz beschrieben, sowie die Einstellungen für den Test. Danach folgt das Protokoll der 3. Versuchsreihe und schließlich deren Auswertung.

5.5.1 KONFIGURATION DER FIREWALL

Für den Test der Firewall in Verbindung mit dem Intrusion Detection System wurde die Konfiguration der Firewall erweitert. Dies ist nötig, da sämtliche Angriffe von der Firewall geblockt wurden. Um zu sehen, ob das Intrusion Detection System einen wirklichen Einbruch erkennt, haben wir uns entschlossen, den NetBIOS-Dienst freizugeben. Damit ist es möglich, einen Angriff mit WinNuke durchzuführen, der auf Kathy einen Blue Screen hervorruft.

Um nun den NetBIOS dienst zu erlauben, muss eine TCP -Verbindung von einem beliebigen Port von Felix mit den Ports 135 bis 139 (nbsession) von Kathy freigegeben werden. Dies muss in auch in die andere Richtungen erlaubt sein. Da dieser Verkehr in diesem Versuch generell erlaub sein soll, reicht der Leistungsumfang des Paketfilters vollkommen aus. Somit implementieren wir zusätzlich folgende Filterregeln:

Rule Name	TCP WinNuke to Kathy
Protocol	TCP
TCP/ICMP-Flags	-
Interface	Untrusted Interface
Screen Action	Forward Traffic without reply
Source Adress	192.168.0.1/16
Source Port range	*
Destination Adress	192.168.1.1/16
Destination Port range	135 – 139 (nbsession)

Abb. 35: Filterregel „TCP WinNuke to Kathy“

Rule Name	TCP WinNuke to Felix
Protocol	TCP
TCP/ICMP-Flags	-
Interface	Trusted Interface
Screen Action	Forward Traffic without reply
Source Adress	192.168.1.1/16
Source Port range	135 – 139 (nbsession)
Destination Adress	192.168.0.1/16
Destination Port range	*

Abb. 36: Filterregel „TCP WinNuke to Felix“

Wichtig dabei ist die Beachtung der Reihenfolge der Regeln. Die in diesem Abschnitt erstellten Regeln müssen in der Abarbeitungsreihenfolge vor der Regel „TCP absorb all“ stehen. Ansonsten wird sämtlicher TCP-Datenverkehr, auch der für die NetBIOS Verbindungen, an die Proxies weitergeleitet. Daher werden alle Filterregeln in ihrer Reihenfolge noch einmal aufgeführt:

1) Ping to kathy
2) Ping to felix
3) Ping to untrusted interface
4) TCP WinNuke to kathy
5) TCP WinNuke to felix
6) TCP absorb all

5.5.2 WAHL UND KONFIGURATION DES INTRUSION DETECTION SYSTEMS

Die Firewall ist für den 3. Versuchsablauf umkonfiguriert worden. Die Beschreibung der Neukonfiguration steht im vorherigen Abschnitt 5.5.1.. Das IDS wurde auf die Firewall aufgesetzt. Als IDS wurde der CyberCop Monitor ebenfalls von Network Associates, Inc. benutzt.

Das Aufweichen der Firewall wurde nötig, weil die Pakete erst an die Firewall gehen und danach an das IDS, wenn die Firewall die Pakete durchgelassen hat. Im 2. Versuch hat die Firewall bereits alle Angriffe geblockt, so dass das IDS keine weiteren Angriffe mehr melden kann. Somit bestand nur noch die Möglichkeit die Firewall umzukonfigurieren, damit das IDS getestet werden konnte.

Das IDS befindet sich zusammen mit der Firewall auf dem Rechner Ergo. Der genaue Versuchsaufbau ist im Abschnitt 5.2.1. beschrieben.

Das IDS ist der hostbasierte Typ. Es hat sowohl Signaturen für die Hostüberwachung als auch für die Überwachung des Netzwerkverkehrs, der über diesen Host läuft. Das verwendete IDS kann auf mehreren Systemen gleichzeitig in einem Netz installiert werden. Die Meldungen

des IDS der einzelnen Rechner können an einen Server weitergeleitet und dort zentral von einem System Security Officer (SSO) überwacht werden. Es gibt nicht nur die einfache Möglichkeit den SSO durch ein aufpoppendes Fenster zu benachrichtigen. Er kann auch über Mobiltelefon, Pager oder E-Mail informiert werden.

Die Meldungen werden in einer Log-Datei gespeichert, um sie später auswerten zu können. Zu den einzelnen Signaturen gibt es jeweils kurze Beschreibungen durch welche Angriffsmuster sie ausgelöst, welche Ports angesprochen und welche Dienste hinter dem Port stehen. Im Log-Monitor gibt es zu jeder Signatur noch einen ausführlicheren Report.

92 NetBIOS OOB Data

A security hole in NetBIOS on port 139 allows malicious users to crash Windows NT machines. Malicious users can send OOB (Out Of Band) data to a user who is connected to a network, such as the Internet, and can crash Windows systems. WinNuke is a program that was written to send OOB data to an IP address of a Windows machine connected to the network. The most common port of attack is NetBIOS (port 139), but other ports are vulnerable if they are listening.

System At Risk: Windows 3.11, Windows 95, Windows NT 3.51, and Windows NT 4.0

Date/Time	Source IP	Destination IP	Repeat Count	Elapsed Time (Seconds)	Additional Message
25.06.01 14:40:18	N/A	N/A	1	39	
25.06.01 14:39:38	N/A	N/A	2	18	
25.06.01 14:39:18	N/A	N/A	1	5	
25.06.01 14:39:13	192.168.0.1	192.168.1.1	1	0	Source Port = 1085; Destination Port = 139; Source MAC Address = 00-E0-7D-02-80-B0; Destination MAC Address = 00-00-B4-B4-4F-15;
25.06.01 14:02:35	N/A	N/A	1	68	
25.06.01 14:01:24	N/A	N/A	1	39	
25.06.01 14:00:44	N/A	N/A	2	18	
25.06.01 14:00:24	N/A	N/A	1	5	
25.06.01 14:00:17	192.168.0.1	192.168.1.1	1	0	Source Port = 1082; Destination Port = 139; Source MAC Address = 00-E0-7D-02-80-B0; Destination MAC Address = 00-00-B4-B4-4F-15;

Abb. 37: Beschreibung der Signatur 92 zur Erkennung von WinNuke

Das IDS bietet an Sicherhiets-Policies zu erstellen. In diesen Sicherheits-Policies befinden sich die Signaturen für die Angriffserkennung. Zur Erstellung einer eigenen Policy werden aus den angebotenen Signaturen, die ausgewählt, die für die Sicherheitspolitik gebraucht werden. Die Erstellung von neuen Signaturen mit dem IDS ist nicht möglich. Dazu fehlt ein entsprechendes Tool. Außerdem liefert das IDS schon vorgefertigte Policies als Sample mit.

Für die 3. Versuchsreihe wurde die mitgelieferte „Empty Policy“ und „Sample – All Signature Rules Active“ verwendet. Ist die Empty Policy aktiviert, sind sämtliche Signaturen ausgeschaltet und das IDS erkennt keine Angriffe. Bei der ausgewählten Sample Policy sind alle verfügbaren Signaturen aktiviert, so dass getestet werden kann, welche Angriffe das IDS erkennt. Deshalb wurde in der Hauptsache für den Test die Sample Policy benutzt. Zur Überprüfung, ob die Toolz

funktionieren, wurde auch die empty policy verwendet. Die Benachrichtigung über die Auslösung einer Signatur erfolgt über ein aufpoppendes Fenster auf Ergo.

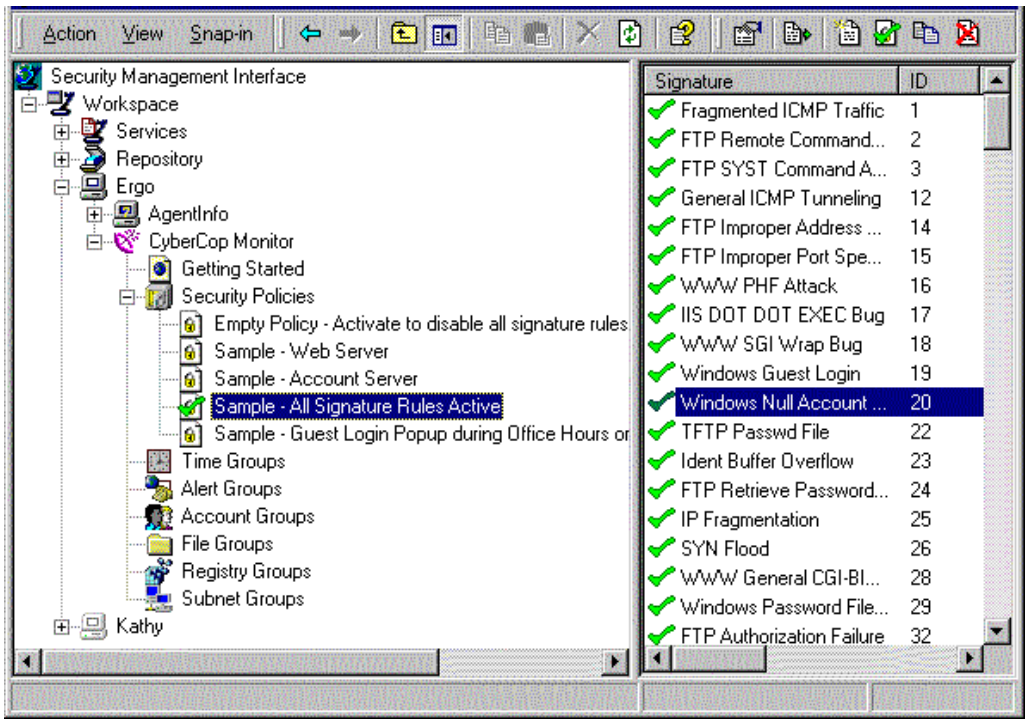


Abb. 38: Policy, in welcher alle Signaturen aktiviert sind.

5.5.3 VERSUCHSDURCHFÜHRUNG MIT FIREWALL UND IDS

Die Angriffsziele sind die gleichen wie im ersten und zweiten Versuch und sind in Kapitel 5.2.2. genau beschrieben. Zum Besseren testen des IDS wurden noch weitere Versuche durchgeführt. Die Angriffstools sind weiterhin Back Orifice 2000, Super Scan und Liquid.

1. Versuch: Es wurde ein Portscan über das gesamte Adressband mit dem Tool Super Scan durchgeführt. Der CyberCop Monitor meldet daraufhin, dass die Signaturen 42-46 und 48 ausgelöst wurden. Die Signaturen erkennen folgende Angriffsmuster:

Signatur	Benutzter Service	Angesprochener TCP Port
42	TCPMUX	1
43	Echo	7
44	Discard	9
45	Systat	11
46	Daytime	13
48	Chargen	19

Abb. 39: Tabelle der wiedererkannten Signaturen

Der Portscan liefert das Ergebnis, das bei dem Opferrechner „Kathy“ mit der IP-Adresse 192.168.1.1 die Ports 13, 21, 23, 25, 80, 135 und 54320 offen sind. Der Rechner Ergo (IP-Adresse: 192.168.1.2) gibt keine Antwort auf den Portscan.

2.-5. Versuch: Das Angriffstool für diese Versuche ist Back Orifice 2000, welches sich zu diesem Zweck mit Kathy verbinden muss. Die Arbeitsweise von Back Orifice 2000 ist in Kapitel 5.2.3. beschrieben. Der Verbindungsaufbau wird durch Gauntlet und den CyberCop Monitor geblockt. Eine Meldung an den System Security Officer erfolgt nicht, wenn alle Signaturen aktiviert sind. Wenn alle Signaturen deaktiviert sind, d. h. die empty policy ist eingestellt, gelingt der Verbindungsaufbau. Ein Nachversuch hat ergeben, dass durch die Signatur 20 des IDS der Verbindungsaufbau durch Back Orifice 2000 scheiterte. Die Signatur 20 erkennt „null sessions". Aber eine Meldung über die Auslösung der Signatur 20 wurde nicht angezeigt. Außerdem ist aufgefallen, dass beim Herunterfahren von Ergo ein Verbindungsaufbau durch Back Orifice 2000 auf Kathy möglich ist.

6. Versuch: Das Angriffstool ist Liquid ein WinNuke-Programm (ebenfalls in Kapitel 5.2.3. beschrieben), dass die Verfügbarkeit des angegriffenen Systems reduzieren soll. Der CyberCop Monitor meldet, dass die Signaturen 75 und 92 ausgelöst wurden und dass der Port 139 angesprochen wurde. Liquid meldet nach dem Versuchsstart: „Error connect () 10060 in failed to connect". Die Signatur 75 bemerkt den NetBIOS session service, welche den Port 139 nutzt. Ebenfalls wird der Port 139 durch die Signatur 92 überwacht. Diese erkennt die Ausnutzung einer Sicherheitslücke von NetBIOS durch Malware, um Windows-Systeme zum Absturz zu bringen.

7. Versuch: Der Gateway-Rechner selbst soll angegriffen werden, d. h. die Firewall und das IDS sind Ziel einer Hackerattacke. Back Orifice 2000 wird als Angriffstool verwendet. Es kommt zu keiner Verbindung, wenn alle Signaturen des IDS aktiviert sind.

Weitere Versuche: Die Versuche finden wie zuvor mit den gleichen Angriffstools und Angriffszielen statt, aber unter veränderten Bedingungen.

- Der Angriff erfolgt unter Verwendung einer internen IP-Adresse des trusted Netzes durch den Angreifer. Der Portscan liefert nur die offenen Ports des Angreiferrechners (Felix). Auch einen Verbindungsaufbau durch Back Orifice 2000 scheitert. Die Firewall Gauntlet meldet im Log-Monitor Spoofing, weil der Angreifer eine trustet Adresse am untrustet Interface verwendet. Weil die Firewall den Angriffsversuch bereits geblockt hat, meldet das IDS gar nichts.
- Der CyberCop Monitor wird daraufhin getestet, ob er ICMP, d. h. Pings, erkennt. Dazu wird Gauntlet umkonfiguriert, so dass es anders als bisher Pings von Kathy zu Felix durchlässt. Der Ping geht durch. Das IDS meldet Signatur 12 für „General ICMP tunneling", Signatur 159 für „ICMP Echo reply" und Signatur

161 für „ICMP Echo request“ wurden ausgelöst. Dabei wird der Log-Monitor von Gauntlet durch den Dauerping aufgefüllt, während der CyberCop Monitor wiederholt vorige Signaturen meldet. Das IDS kann die Firewall nicht konfigurieren, deshalb blockt die Firewall den Dauerping nicht.

- Der Port 139 wird auf Gauntlet freigeschaltet, damit Liquid nicht mehr durch die Firewall geblockt wird. Als erstes überwacht der CyberCop Monitor den Port 139 nicht. Der Angriff auf Kathy gelingt. Als zweites wird der Port 139 durch das IDS überwacht. Der Angriff auf Kathy gelingt auch diesmal. Nach 2 Minuten und 25 Sekunden meldet der CyberCop Monitor das die Signatur 92 für einen Angriff durch ein WinNuke-Programm ausgelöst wurde. Durch die späte Meldung des IDS können keine automatischen Reaktionen ausgelöst werden, die noch rechtzeitig etwas gegen den Angriff unternehmen können.

Eine besondere Auffälligkeit bestand darin, dass schon nach kurzer Zeit Keiner mehr die Meldungen des IDS durchlas, sondern diese sofort schloss. Die Ursache dafür war, dass das IDS ständig Meldungen ausgab, die sich auf den normalen Netzwerkverkehr zwischen Kathy und Ergo bezogen. Dies waren normale NetBIOS-Aktivitäten, welche Pakete alle paar Minuten austauschten. Bei den Meldungen handelt es sich um die Signaturen:

Signatur	Benutzter Service	Angesprochener Port
74	NetBIOS Datagram	TCP 138
78	XDMCP	TCP 177
194	NetBIOS Name	UDP 137
196	UDP Broadcast	

Abb. 40: Tabelle der wieder erkannten Signaturen

5.5.4 VERSUCHSAUSWERTUNG

Das IDS hat fast alle Angriffsversuche gemeldet.

Der Portscan wurde zwar nicht als solcher erkannt, dennoch schlug CyberCop Alarm, wobei mehrere Signaturen zutrafen. Eine Analyse der gleichzeitig auftretenden Meldungen hätte den Angriff als Portscan ausweisen können.

Auffällig war, dass das IDS den versuchten Verbindungsaufbau durch Back Orifice 2000 nicht gemeldet hat. Trotzdem wurde dieser sogar verhindert. Erst eine binäre Suche in den Signaturen ergab, dass eine Signatur des IDS dafür verantwortlich ist. Warum der Verbindungsaufbau geblockt wurde, konnte nicht herausgefunden werden. Laut der Beschreibung des IDS ist es möglich, dass das IDS die Firewall rekonfigurieren kann, um einen Angriff zu beenden. Diese Funktionalität wurde nicht gefunden. Jedenfalls kann nicht angegeben werden, dass bei

Erkennung eines Angriffs die entsprechenden Ports durch die Firewall geschlossen werden sollen.

Durch die Versuche mit WinNuke wurde klar, dass das IDS keinerlei Schutz bietet, wenn die Firewall NetBIOS durchlassen soll. Beim 6. Versuch hat die Firewall den Angriff durch WinNuke geblockt und das IDS diesen Angriff gemeldet. Doch einen weiteren Versuch wurde die Firewall soweit aufgeweicht, dass sie die WinNuke-Attacke durchließ. Das Ergebnis war, dass Kathy abstürzte und das IDS den Angriff erst nach fast 2,5 Minuten meldete. Also viel zu spät, um noch irgendetwas zu untemehmen. Ob diese starke zeitliche Verzögerung an einer Überlastung des Systems lag oder das IDS einfach zu langsam arbeitete, konnte nicht festgestellt werden.

In einem weiteren Versuch sollte auch überprüft werden, ob das IDS Pings erkennt. Dies klappte auch. Aber das Log der Firewall wurde stark aufgefüllt und das IDS meldete ständig ICMP-Pakete, denn es handelte sich um einen Dauerping. Wenn die Sicherheitspolitik verlangt, dass Pings erlaubt sein müssen, dann darf die Firewall diese nicht blocken. Trotzdem werden die Pings von der Firewall und dem IDS bemerkt und von ihnen geloggt bzw. gemeldet. Ein Dauerping verbraucht dadurch eine Menge Speicherplatz und die ständigen Meldungen (nicht nur von einem Dauerping, sondern auch von ganz normalen Pings) werden durch den System Security Officer (SSO) schnell weggedrückt. Dies führt zu einer verminderten Sicherheit, weil dabei auch Meldungen weggeklickt werden können, die für den SSO wirklich relevant sind. Dieses Phänomen tritt nicht nur bei Pings auf, sondern auch bei anderen regulären Diensten wie zum Beispiel den NetBIOS-Services, die gemeldet werden.

Alle anderen Attacken wurden erkannt und führten nicht zu einer Schädigung von Ergo und Kathy.

Als unmöglich erwies es sich, neue Signaturen zu erstellen. Zur normalen Wartung eines IDS sollte es gehören, dass neue Angriffsmuster in die Signaturdatenbank aufgenommen werden können. Auch eine Änderung der Signaturen sollte möglich sein, um diese auf spezielle Systeme anzupassen.

Insgesamt kann gesagt werden, dass ein IDS zur höheren Sicherheit beiträgt. Es kann den SSO bei seiner Arbeit unterstützen, weil es die meisten bekannten Angriffe erkennt und daraufhin Alarm schlägt. Trotzdem bietet eine Firewall mit aufgesetztem IDS keinen absoluten Schutz. Denn sowohl die Schwächen der Firewall (siehe 5.4.3) als auch die aufgezeigten Schwächen des IDS machen einen erfolgreichen Angriff möglich.

6 RESÜMEE

Unser Projekt hat aufgezeigt, dass das heutige Sicherheitsbedürfnis mit Firewalls und Intrusion Detection Systemen nicht in ausreichendem Maße befriedigt werden kann. Viele Sicherheitspolitiken lassen sich nicht vollständig in Regelsätzen für Firewalls oder Intrusion Detection Systemen umsetzen, weil diese nicht mit ausreichender Detailtiefe formuliert werden können. Ein Beispiel dafür ist in der zweiten Versuchsstufe aufgetreten: Das unzureichende Blocken von Back Orifice 2000 durch die Firewall, weil Back Orifice 2000 denselben Port wie die Finanzanwendung verwendete.

Wenn Firewalls und IDS nicht ausreichen, wie kann weitergehender Schutz gewährleistet werden? Heutzutage stehen noch einige weitere Sicherheitsmechanismen zur Verfügung, wie beispielsweise digitale Signaturen oder Einrichtungen wie Virtual Private Network (VPN). Bei digitalen Signaturen wird ein asymmetrisches Verschlüsselungsverfahren angewendet, um die Authentizität einer Nachricht zu gewährleisten. Die Nachricht wird mit dem privaten Schlüssel des Absenders „signiert“ und vom Empfänger mit dem öffentlichen Schlüssel des Absenders verifiziert. Mit einem solchen System der Authentisierung kann beispielsweise erreicht werden, dass nur Nachrichten ausgewählter Benutzer vom Netz beachtet werden. Wenn die Signierung durch die Kodierung mit dem privaten Schlüssel auf einen Hash-Wert der Nachricht angewendet wird, ist auch mit sehr hoher Wahrscheinlichkeit die Integrität der Nachricht anzunehmen.

Das Problem dabei ist, dass die nötige technische Infrastruktur – die so genannte Public Key Infrastructure (PKI) – heute in weiten Teilen noch nicht existiert. Außerdem gibt es auf dem Markt viele Unterschiedliche Instanzen, die mehr oder weniger vertrauenswürdige digitale Signaturmöglichkeiten anbieten. Bevor das Prinzip der digitalen Signatur großflächig genutzt werden kann, müssten also zunächst ein einheitliches Signatursystem und eine breite Akzeptanz für diese Infrastruktur geschaffen werden. Zudem setzen digitale Signaturen die Performanz bei der Übertragung über ein Netz herab, da alle Nachrichten einem weiteren Prüfungsschritt unterzogen werden müssen. Virtual Private Network (auch als Internet Tunneling bekannt) ist ein Verfahren, bei dem Internetverbindungen genutzt werden, um verschlüsselte Nachrichten zwischen zwei einander vertrauenden LANs zu versenden.

Selbst wenn digitale Signaturen und andere heute bekannte Maßnahmen in Zukunft verstärkt eingesetzt werden, dürften sich damit aber die Sicherheitsprobleme nicht vollständig lösen lassen. Schon längst treten neue Angriffsformen auf, gegen die es heute noch keine Abwehrmechanismen gibt. So können empfindliche Geräte Monitorstrahlung empfangen und analysieren. Dadurch kann das auf dem Monitor dargestellte Bild rekonstruiert werden, ohne dass es

eine physikalische Verbindung zum Rechner gibt. Die zunehmende Verbreitung drahtloser Netze (zum Beispiel wireless LANs) ermöglicht ebenfalls das Abhören des Netzverkehrs, ohne dass eine physikalische Verbindung nötig ist.

Solche Bedrohungen können heute nur vermindert werden, indem hochsensible Rechner vollständig von der Außenwelt abgeschirmt werden. Diese Abschirmung muss so vollständig sein, dass sie auch elektromagnetische Strahlungen blockt.

Man wird weitere Sicherheitsmaßnahmen erarbeiten müssen. Leider solche Maßnahmen nicht im Voraus entwickelt werden, sondern immer nur als Reaktion auf existierende Angriffe oder Bedrohungen, weshalb Unternehmen, die Sicherheitskonzepte vertreiben, in letzter Zeit verstärkt Cracker anheuern.

Das Voranschreiten der Technologie eröffnet immer neue Möglichkeiten, Netze und Rechner anzugreifen. Es ist daher ständige Wachsamkeit gefordert, um auf neue Bedrohungen möglichst schnell reagieren zu können. Beim Entwickeln einer neuen Technologie sollte diese immer auf eventuelle Sicherheitslücken untersucht werden, um die Gefahr von Anfang an zu mildern. Dennoch werden immer unentdeckte Lücken übrig bleiben, die im Nachhinein geschlossen werden müssen und bis zum Auftreten eines Angriffs unentdeckt bleiben.

Vollständige Sicherheit gibt es also leider nicht. Es kann lediglich das Vertrauen in die Sicherheit gestärkt werden, indem versucht wird, möglichst hohe Barrieren gegen Hacker aufzubauen.

QUELLENVERZEICHNIS

[Bonnard 97] Bonnard, A.; Wolff, C.: *Gesicherte Verbindung von Computernetzen mit Hilfe einer Firewall*. Studie im Auftrag des Bundesamtes für Sicherheit in der Informationstechnik, München: Siemens 1997.

[Brunnstein 01] Brunnstein, K.: *Gestaltbarkeit und Beherrschbarkeit von Informatiksystemen (GBI).* Vorlesungsskript, Universität Hamburg, 2001.

[Brunnstein 01a] Brunnstein, K.: *ITSec.4* Vorlesung am 31.5.2001, Fachbereich Informatik, Universität Hamburg.

[BMI 01] Bundesministerium des Innern: *Polizeiliche Kriminalstatistik 2000*. Internet http://www.bmi.bund.de/Anlage6166/Download.pdf, 22.05.2001.

[BSI 99] Bundesamt für Sicherheit in der Informationstechnik: *Grundschutzhandbuch.* Bonn, 1999, http://www.eddy.uni-duisburg.de/GSHB/index.html

[Chapman 00] Chapman, D. B.; Cooper, S.; Zwicky, E. D.: *Building Internet Firewalls.* 2. Aufl. Sebastapol, CA: O'Reilly & Associates Inc., 2000.

[Castano 94] Castano, S; Fugini, M.G.; Martella, G.; Samarati, P.: *Database Security*, Addison-Wesley Publishing Company, 1994.

[Engel 99] Engel, A.; Lessig, A. G.: *Internetgestützte Angriffe und ausgewählte Gegenmaßnahmen*. Diplomarbeit, Universität Hamburg 1999.

[FAZ 17.2.00] Frankfurter Allgemeine Zeitung: *Clinton will gegen Hacker kämpfen.* Ausgabe vom 17.02.2000.

[Fiolka 01] Fiolka, K.: *Auswirkungen des aVTC auf die Bedrohung der computerisierten Welt durch Malware und der Wintertest 2000/2001.* Baccalaureatsarbeit, Universität Hamburg 2001.

[Freiss 98] Freiss, M.: *Protecting Networks with SATAN*. O'Reilly & Associates Inc., Sebastopol, 1998.

[Garfinckel 96] Garfinkel, S.; Spafford, G.: *Practical UNIX & Internet Security*. 2. Aufl. Sebastapol, CA: O'Reilly & Associates Inc., 1996.

[Gasser 88] Gasser, M.: *Building a Secure Computer System.* Van Nostrand Reinhold, New York, 1988.

[Grill 92] Grill, G.; Zwahr, A.: *Meyers grosses Taschenlexikon.* 4. Aufl. Mannheim: B.I.-Taschenbuchverlag, 1992.

[Gröndahl 00] Gröndahl, B.: *Hacker.* Hamburg: Rotbuch 3000, 2000.

[Helden 98] Helden, J. von; Karsch, S.: *Grundlagen, Forderungen und Marktübersicht für Intrusion Detection Systeme (IDS) und Intrusion Response Systeme (IRS).* Debis IT-Security Services, Bonn, 1998.

[IEEE] Institute of Electrical and Electronics Engineers, Inc.: *IEEE802.* Internet http://standards.ieee.org/getieee802/, 2001.

[ISPTG] Guttman, B.; Bagwill, R.: *NIST Special Publication 800-XX, Internet Security Policy: A Technical Guide.* 1997, http://csrc.nist.gov/isptg/html/index.html

[Kerner 95] Kerner, H.: *Rechnernetze nach OSI.* 3. Aufl. Bonn: Addison-Wesley, 1995.

[LM 97] Leitner, A.; Schubert, J.: *Einführung in die Brandvorsorge.* Linux Magazin 06/2001, Linux New Media AG, München, 2001.

[Mück 00] Mück, H.-J.: *Leitfaden zur Absicherung von Rechnersystemen in Netzen.* DFN-CERT, Zentrum für sichere Netzdienste GmbH, Hamburg, 2000, ftp://ftp.cert.dfn.de/pub/docs/leitfaden/leitfaden.pdf

[Mutlu 01] Mutlu, S.; Schnell, A.; Yüksel, E.: *Intrusion Detection als ergänzender Sicherheitsmechanismus am Beispiel von Unix.* Diplomarbeit, Universität Hamburg, 2001.

[NAI 99] Network Associates Technology, Inc.: *Gauntlet for Windows NT, Administrator's Guide.* Benutzeranleitung, 1999.

[Nedon 00] Nedon, J.: *Ein IT-Sicherheitskonzept für eine wissenschaftliche Einrichtung.* Diplomarbeit, 2. Version, Universität Hamburg, 2000.

[Pfleeger 97] Pfleeger, C.: *Security in Computing.* Prentence Hall, New York, 1997.

[Raepple 01] Raepple, M.: *Security in Computing.* Dpunkt.verlag, Heidelberg, 2001.

[Raymond 00] Raymond, E. S.: *A Brief History of Hackerdom.* http://www.tuxedo.org/~esr/writings/hacker-history/hacker-history.txt, 2000.

[Raymond 01] Raymond, E. S. (Hrsg.): *Jargon File.* Version 4.3.0. http://www.tuxedo.org/~esr/jargon/, 2001.

[RedHat 01] Red Hat Inc.: *Red Hat Linux 7.1, Das Offizielle Red Hat Linux Handbuch Benutzerdefinierte Konfiguration.* Durham, 2001

[RFC] Internet Assigned Numbers Authority: *Request for Comments.* Internet http://www.rfc-editor.org, 2001.

[RFC782] Nabielsky, J.; Skelton, A. P.: *A Virtual Terminal Management Model.* Request for Comments 782, http://www.rfc-editor.org

[RFC791] Information Science Institute, University of Southern California: *Site Security Handbook.* Request for Comments 791, 1981, http://www.rfc-editor.org

[RFC792] Postel, J.: *Internet Control Message Protocol*. Request for Comments 792, 1981, http://www.rfc-editor.org

[RFC793] Information Science Institute, University of Southern California: *Transport Control Protocol.* Request for Comments 793 1981, http://www.rfc-editor.org

[RFC959] Postel, J.; Reynolds, J.: *FileTransport Protocol*. Request for Comments 959, 1985, http://www.rfc-editor.org

[RFC1001] Network Working Group.: *Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concept and Methods*. Request for Comments 1001, 1987, http://www.rfc-editor.org

[RFC1123] Braden, R.: R*equirements for Internet Hosts – Application and Support.* Request for Comments 1123, 1989, http://www.rfc-editor.org

[RFC2196] Fraser, B.: *Site Security Handbook.* Request for Comments 2196, 1997, http://www.rfc-editor.org

[RFC2616] Fielding R., et. al.: *Hypertext Transfer Protocol – HTTP/1.1* Request for Comments 2616, 1999, http://www.rfc-editor.org

[RFC2821] Klensin, J.: *Simple Mail Tranfer Protokoll*. Request for Comments 2821, 2001, http://www.rfc-editor.org

[Russell 91] Russell, D.; Gangemi, Sr. G. T.: *Computer Security Basics*. 1. Aufl. Sebastapol, CA: O'Reilly & Associates Inc., 1991.

[SANS 01] The SANS Security Policy Project, SANS Institute, 2001, http://www.sans.org/newlook/resources/policies/policies.htm

[Spitzner 00] Spitzner, L.: *Building Your Firewall Rulebase*. 2000, http://www.enteract.com/~lspitz/rules.html

[Sterling 93] Sterling, B.: *The Hacker Crackdown: Law and Disorder on the Electronic Frontier.* Spectra Books, 1993.

[Stoll 99] Stoll, C.: *Kuckucksei*. 3. Aufl. Frankfurt am Main: S. Fischer Verlag GmbH, 1999.

[Tanenbaum 98] Tanenbaum, A. S.: Computernetzwerke. 3. Aufl. München: Prentice Hall, 1998.

ABBILDUNGSVERZEICHNIS

ANLAGE A ZUORDNUNGSLISTE VON DIENSTEN ZU TCP-PORTS

Die nachfolgende Liste entstammt den Programmen SuperScan 3.00 von Foundstone, LANguard Network Scanner (v.1.1) von GFI FAX & VOICE, Netbrute Scanner Build Version 1.0.0.29 von Raw Logic Software und IP Ultra Scan 2000 von Ultrajones. Die Liste weist nur eine Auswahl von Ports auf, die entweder besonders wichtig oder von Hacker besonders gerne verwendet werden.

Port	Dienst
1	TCP Port Service Multiplexer
2	Management Utility
3	Compression Process
5	Remote Job Entry
7	Echo
9	Discard
11	Active Users
13	Daytime
17	Quote of the Day
18	Message Send Protocol
19	Character Generator
20	File Transfer [Default Data]
21	File Transfer Protocol [Control]
22	SSH Remote Login Protocol
23	Telnet
24	any private mail system
25	Simple Mail Transfer Protocol
27	NSW User System FE
29	MSG ICP
31	Master Paradise*
31	MSG Authentication
33	Display Support Protocol
35	any private printer server
37	Time
38	Route Access Protocol
39	Resource Location Protocol
41	Graphics
42	WINS Host Name Server
43	Who Is
44	MPM FLAGS Protocol
45	Message Processing Module [recv]
46	MPM [default send]
47	NI FTP
63	Whois
64	Communications Integrator (CI)
79	Finger
80	World Wide Web HTTP
81	World Wide Web HTTP
88	Kerberos
110	Post Office Protocol - Version 3
113	Authentication Service
115	Simple File Transfer Protocol
121	BO jammerkillahV*
121	Encore Expedited Remote Pro.Call
136	PROFILE Naming System
137	NETBIOS Name Service
138	NETBIOS Datagram Service
139	NETBIOS Session Service
143	Internet Message Access Protocol
194	Internet Relay Chat Protocol
414	InfoSeek
443	Hyper Text Transfer Protocl Secure
456	HackersParadise*
456	macon-tcp
512	remote process execution
513	Remote login
514	cmd
666	Attack FTP*
666	doom Id Software
1001	Silencer*
1001	WebEx*
1010	Doly 1.30 (Subm.Cronco)*
1011	Doly 1.1+1.2*
1015	Doly 1.5 (Subm.Cronco)*
1033	Netspy*
1042	Bla1.1*
1089	SocksServer*
1170	Streaming Audio*
1207	SoftWar*
1243	SubSeven*
1245	Vodoo*
1269	Maverick's Matrix*
1492	FTP99CMP*
1492	stone-design-1
1509	PsyberStreamingServer Nikhil G.*
1509	Robcad
1807	SpySender*
1807	Fujitsu Hot Standby Protocol
1981	ShockRave*
1999	Backdoor*
1999	Transcout 1.1 + 1.2*

* Dieser Dienst wird von maliziöser Software genutzt und/oder bereitgestellt.

Port	Dienst
1999	cisco identification port
2001	DerSpaeher 3*
2001	TrojanCow*
2023	Pass Ripper*
2140	The Invasor Nikhil G.*
2283	HVL Rat5*
2283	LNVSTATUS
2565	Striker*
2583	Wincrash V2.0*
2801	Phineas Nikhil G.*
3791	Total Eclypse (FTP)*
3883	Deep Throat 2*
4567	FileNail Danny*
4672	Remote file access server
4950	ICQTrojan*
5000	Socket23*
5011	OOTLT*
5011	TelepathAttack*
5031	NetMetro1.0*
5400	BladeRunner*
5400	BackConstruction1.2*
5521	IllusionMailer*
5550	XTCP 2.0x*
5569	RoboHack*
5742	Wincrash V1.03*
5800	Virtual Network Computing server
5882	Y3k*
5900	Virtual Network Computing server
6000	The tHing 1.6*
6400	The tHing*
6667	Internet Relay Chat server
6669	Vampire 1.0*
6670	Deep Throat*
6883	DeltaSource (DarkStar)*
6912	Shitheep*
6939	Indoctrination*
7306	NetMonitor*
7789	iCkiller*
8080	Standard HTTP Proxy
9400	InCommand*
9872	PortalOfDoom*
9875	Portal of Doom*
9989	iNi-Killer*
10607	Coma Danny*
11000	SennaSpyTrojans*
11223	ProgenicTrojan*
12076	Gjamer*
12223	Hack´99 KeyLogger*
12345	Windows Netbus backdoor*

Port	Dienst
12346	NetBus 1.x (avoiding Netbuster)*
12349	Bionet*
12701	Eclipse 2000*
16969	Priotrity*
17300	Kuang2*
17569	Infector*
20000	Millenium*
20001	Millenium trojan*
20024	Netbus 2.0 Pro*
20034	NetBus Pro*
20203	Logged!*
20203	Chupacabra*
20331	Bla*
21544	GirlFriend*
21554	GirlFriend*
22222	Prosiak 0.47*
23432	Asylum*
23456	EvilFtp*
27374	Sub-7 2.1*
29891	The Unexplained*
30029	AOLTrojan1.1*
30100	NetSphere*
30303	Socket25*
30999	Kuang*
31337	Back Oriffice*
31787	Hack'a'tack*
33911	Trojan Spirit 2001 a*
34324	Tiny Telnet Server*
34324	BigGluck TN*
40412	TheSpy*
40423	Master Paradise*
44444	Prosiak*
50766	Fore*
53001	RemoteWindowsShutdown*
54320	Back Orifice 2000 (default port)*
54321	Schoolbus 1.6+2.0*
61466	Telecommando*
65000	Devil 1.03*
65301	pcAnywhere