1. Einleitung

Der Benutzer eines Computersystems sieht die Daten auf seinem Rechner heutzutage einer Vielzahl von Bedrohungen ausgesetzt. Besonders, wenn das Computersystem an ein Netzwerk (z.B. das Internet) angeschlossen ist, besteht die Gefahr von Angriffen auf den Rechner aus dem Netz. Zusätzlich zu direkten Angriffen besteht auch eine Bedrohung durch bösartige Software, die über Dienste wie e-mail und ftp-Download oder über Disketten auf den Rechner des Benutzers gelangen kann und dort lokal ausgeführt wird. Bösartige Software - im folgenden auch als Malware bezeichnet - verbreitet sich dabei teilweise selbständig in Dateisystemen oder Netzwerken (zum Beispiel Würmer, die sich selbst an sämtliche Adressen des e-mail Adressbuchs des befallenen Rechners weiterversenden)[1]. Teilweise gelangt sie nur mit Hilfe des Benutzers auf den Rechner, etwa beim Download von Dateien aus dem Internet oder beim Herunterladen von e-mail-Anhängen. Solche - vom Benutzer bewußt ausgeführte (oder zumindest bewußt geöffnete) Software - kann bösartige Funktionalitäten besitzen, die dem Benutzer nicht bekannt sind (zum Beispiel "Trojanische Pferde", die nach außen eine gewisse Funktion zu erfüllen scheinen, aber im Hintergrund andere - vor dem Benutzer verborgene - Funktionen ausführen, wie etwa das Ausspähen des Rechners, Verschicken von Dateien oder Löschen von sensitiven Daten)[2]. Bei falsch gewählten Einstellungen der Browsersoftware (welche teilweise die Standardeinstellungen sind) kann selbst der Besuch von Webseiten gefährlich sein und die Ausführung von bösartiger Software, die Schaden auf dem Computersystem anrichtet, zur Folge haben[3].

Im einzelnen bestehen - unter anderem - folgende Bedrohungen für Benutzer von (vernetzten) Computersystemen[4]:

- Verlust von Daten
- Unbefugter Zugriff auf lokale Daten (aus dem Netzwerk)
- Unbefugter (eventuell unbemerkter) Versand von sensitiven Daten
- Unbefugte (eventuell unbemerkte) Veränderung von lokalen Daten (aus dem Netzwerk)
- Datenspionage bei Versand von Daten über ein Netzwerk

---

[1] Ein Beispiel für einen solchen Internet-Wurm ist der sogenannte "*I love you*"-Wurm (auch *loveletter* genannt), der als e-mail in der Betreffzeile *I love you* stehen hat und sich bei Öffnen dieser e-mail unter Microsoft Outlook an alle Adressen des Adreßbuches versendet. Genaue Angaben zum Loveletter-Wurm finden sich unter [Loveletter 2002].

[2] Ein Beispiel für ein trojanisches Pferd ist "*Liberty.A*", ein Programm für das Betriebssystem *Palm OS*, das sich als Crack für einen Gameboy-Emulator ausgibt, tatsächlich aber bei Aktivierung sämtliche Programmdateien auf dem befallenen Palm-Computer löscht. Genaue Angaben zum trojanischen Pferd *Liberty* finden sich unter [Liberty 2002].

[3] Der Wurm "*Nimda*" verbreitet sich auf verschiedene Arten, unter anderem kann er Rechner beim Besuch infizierter Webserver befallen. Genaue Angaben zum Wurm *Nimda* finden sich unter [Nimda 2002].

[4] vgl. [Pfleeger 1997], S. 390ff

- Adreßfälschung: eine andere Person gibt sich im Netz als der Benutzer aus und verschickt unter dessen Identität Nachrichten oder führt andere Aktionen aus
- Verhinderung des Zugriffs auf Dienste im Netzwerk (*denial of service*)

Da die Benutzer von Computersystemen in der Regel weder auf ausführbare Programme noch auf den Austausch von Daten mit anderen Computerbenutzern verzichten wollen und da es keinen generellen Schutz vor den beschriebenen Bedrohungen gibt, müssen sich die Benutzer auf Sicherheitssoftware verlassen, die ihre Systeme schützt. Dies gilt um so mehr, je weniger Wissen der Benutzer hinsichtlich der Sicherheit von Computersystemen besitzt und je wichtiger und sensitiver die Daten auf dem entsprechendem Computersystem sind. Verfügbare Schutzmechanismen für die Sicherheit von Computersystemen sind[5]:

- Verschlüsselung
- Digitale Signatur
- Checksummen
- Anti-Malware-Software (Virenscanner)
- Firewalls
- Intrusion Detection Systeme (IDS)

Jeder dieser Sicherheitsmechanismen schützt den Benutzer vor einer bestimmten Bedrohung. So kann beispielsweise Verschlüsselung die Vertraulichkeit von Daten schützen. Checksummen können die Integrität von Daten gewährleisten; Firewalls schützen ein Computersystem vor Bedrohungen durch Angriffe von außen. Keine der aufgelisteten Schutzmechanismen bietet dem Benutzer jedoch vollständigen Schutz vor jeglicher Art von Bedrohung.

1.1 Einführung in die Thematik

Wenige Benutzer verfügen über Fachwissen und sind in der Lage, den durch Sicherheitsmechanismen in bestimmten Softwareprodukten gewährleisteten Schutz einzuschätzen. Hinzu kommt, daß selbst diese Benutzer kaum den Aufwand betreiben können, um den Schutz ihrer Software zu prüfen. Da sich die Benutzer aber auf die entsprechenden Sicherheitsprogramme und deren Funktionen zum Schutz ihrer Systeme verlassen müssen, ist eine Qualitätsanalyse der verfügbaren Sicherheitsprogramme nötig.

Computerviren und andere Formen von Malware (wie zum Beispiel trojanische Pferde) sind in den letzten Jahren eine immer größere Bedrohung für Benutzer von Computersystemen geworden. Es werden weltweit nicht nur immer mehr Viren entdeckt; hinzu kommt, daß ab und an auch neue Arten von Computerviren auftauchen, wie zum Beispiel Skriptviren oder

[5] vgl. [Brunnstein 2001], Kapitel 3

sogenannte "0190-Dialer"[6] in den letzen Jahren. Abbildung 1.A verdeutlicht die wachsende Bedrohung für Benutzer durch eine immer größer werdende Zahl an Computerviren[7]. Um sich vor solch einer Vielzahl an bösartiger Software zu schützen, sind Benutzer auf den Schutz durch Anti-Malware-Software angewiesen, die bösartige Software entdeckt und deren Ausführung verhindert.

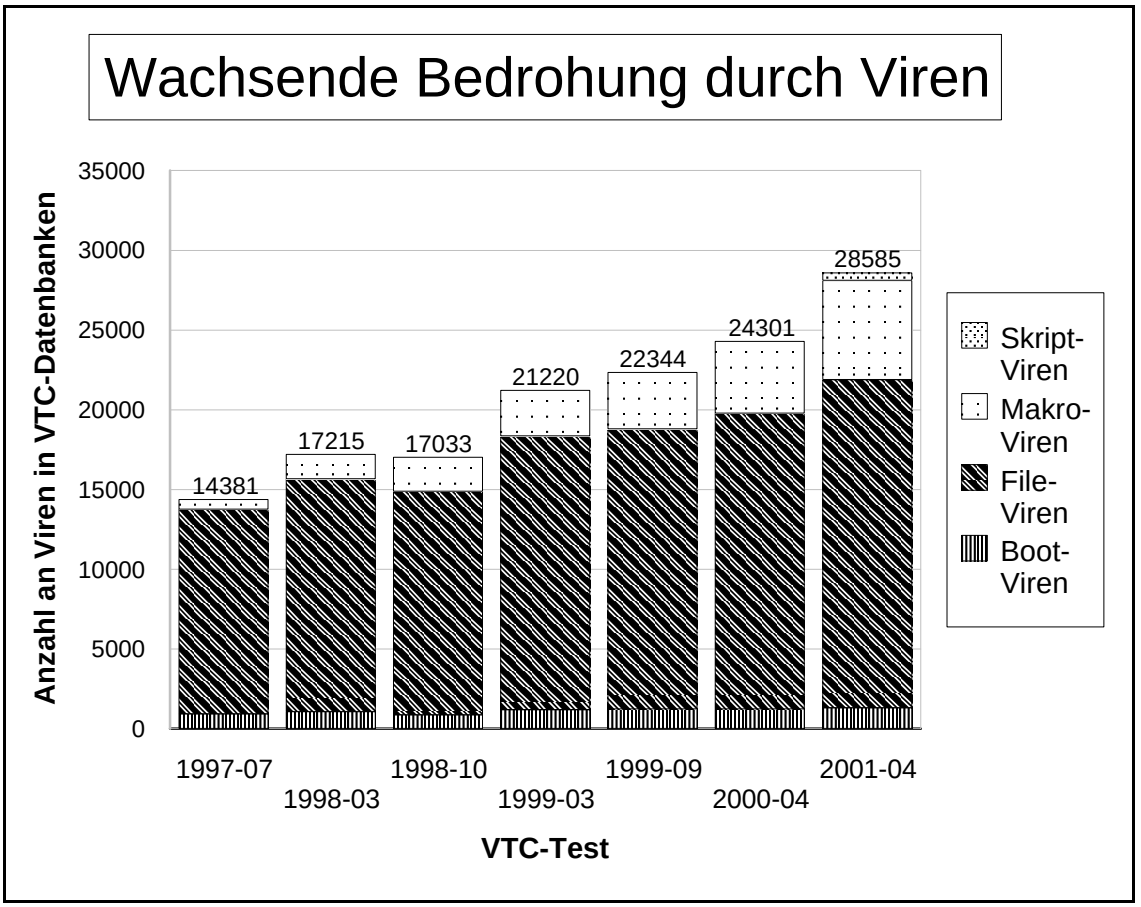

Abbildung 1.A: Wachsende Bedrohung durch Computerviren

[6] Als *0190-Dialer* bezeichnet man Programme, die sich in das Telefonnetz einwählen können und dort kostenpflichtige Nummern mit der Vorwahl *0190* wählen, durch die teilweise erhebliche Kosten für den Benutzer entstehen. Solche Programme sind maliziös, da die Einwahl entweder von alleine - ohne Zutun des Benutzers - geschieht oder aber auf die entstehenden, extrem hohem Kosten nicht hingewiesen wird.

[7] Die Grafik zeigt die Zahl an Viren - unterteilt nach Plattform - die in den Datenbanken des Virus Test Centers der Universität Hamburg (VTC) enthalten sind. In den Tests 2000-08 und 2001-10 wurden keine File- und Boot-Datenbanken getestet, deshalb ist die Anzahl an Viren dieser Tests in der Abbildung nicht mit aufgeführt.

Die vorliegende Arbeit befasst sich mit der Qualitätsbestimmung von Anti-Malware-Software. Anti-Malware-Programme (oder Virenscanner) schützen den Benutzer vor bösartiger Software durch verschiedene Mechanismen und Funktionen. Wie man die Qualität des gewährten Schutzes einer solchen Software bestimmen kann, ist Thema dieser Arbeit. Dabei orientiert sich die Arbeit am Virus Test Center der Universität Hamburg, wo regelmäßig Produkttests von Anti-Malware-Programmen durchgeführt werden.

Um Anti-Malware-Produkte testen zu können, muß zunächst festgelegt werden, was ein gutes Anti-Malware-Produkt ausmacht, welche Kriterien also zu testen sind. Kapitel 2 beschäftigt sich mit der Frage, welche Anforderungen an ein Anti-Malware-Produkt zu stellen sind und was ein gutes Anti-Malware-Produkt für Eigenschaften besitzen sollte. Kapitel 3 untersucht das Testen von Software und insbesondere Anti-Malware-Software. Es werden Qualitätsrichtlinien für Tests von Anti-Malware-Produkten aufgestellt. Außerdem wird erläutert, wie Anti-Malware-Programme getestet werden können. In Kapitel 4 wird ausführlich das Verfahren des Virus Test Centers der Universität Hamburg (VTC) zum Testen von Anti-Malware-Software dargestellt. Dabei wird auf die in Kapitel 2 beschriebenen Qualitätskriterien eingegangen und die Methodik des VTC im Detail geschildert. Anhand der in Kapitel 3 erläuterten Qualitätsrichtlinien für Tests von Anti-Malware-Software wird das in Kapitel 4 dargestellte Verfahren des Virus Test Centers in Kapitel 5 mit anderen Testinstitutionen, die Anti-Malware-Software testen, verglichen. Kapitel 6 beschreibt die im Rahmen der vorliegenden Arbeit entwickelten Verbesserungen und Erweiterungen am Verfahren des Virus Test Centers zur Qualitätsbestimmung der Erkennung von bösartiger Software, die unter anderem die Auswertung der Testergebnisse automatisieren. In Kapitel 7 werden einige durch die in Kapitel 6 beschriebenen Verbesserungen aufbereitete Testergebnisse aktueller VTC-Tests dargestellt.

1.2 Terminologie

Zum besseren Verständnis dieser Arbeit und zur eindeutigen Bestimmung der verwendeten Fachausdrücke sollen die Begriffe aus dem Bereich der IT-Sicherheit, die zur Analyse bösartiger Software relevant sind, erläutert und definiert werden. Dies ist auch deshalb von besonderer Bedeutung, da die in Kapitel 2 aufgestellten Kriterien für die Qualität von Produkten zur Erkennung bösartiger Software Fachbegriffe beinhalten, die in der Literatur unterschiedlich definiert sind.

1.2.1 Definitionen

Die folgenden Definitionen sind die vom Autor verwendeten Begriffsbestimmungen, auf die sich die weitere Verwendung in der Arbeit bezieht. Weitere Begriffsbestimmungen und Einführungen zu Computerviren und Malware finden sich in [Pfleeger 1997], [Gasser 1988] und [KittelTicak 2002].

Malware / bösartige Software

Definition[8]:

> *"A software or module is called "malicious" ("malware") if it is intentionally dysfunctional, and there is sufficient evidence (e.g. by observation of behaviour at execution time) that dysfunctions may adversely influence the usage or the behaviour of the original software."*

Malware faßt als Oberbegriff alle Arten von ***bösartiger Software*** zusammen, die für Benutzer eine Gefahr darstellt. Mit ***Malware*** wird jegliche Art von Software bezeichnet, die *intentional dysfunktional* ist. *Dysfunktional* bedeutet, daß die Software von der (formalen oder informalen) Spezifikation abweicht[9] und somit für den Anwender "unerwünschte und unerwartete Funktionen besitzt"[10]. *Intentional* bedeutet, daß diese zusätzlichen, verborgenen Funktionalitäten vom Programmierer der Software beabsichtigt sind. Intentional dysfunktionale Software (Malware) beinhaltet als Begriff also nicht sogenannte "Bugs" oder sonstiges, unbeabsichtigtes und im Rahmen von Softwareentwicklung oft versehentlich entstandenes Fehlverhalten.

Als Synonyme werden in dieser Arbeit für ***Malware*** auch die Begriffe ***bösartige Software*** und ***maliziöse Software*** verwendet.

Virus

Definition[11]:

> *"Any software that reproduces (or "self-replicates"), possibly depending on specific conditions, at least in 2 consecutive steps upon at least one module each (called "host") on a single system on a given plattform, is called a "virus" (for that plattform). A virus may be compiled (e.g. boot and file virus) or interpreted (e.g. script virus)."*

Ein ***Virus*** ist eine bestimmte Art von Malware, nämlich bösartige Software, die durch Infektion andere Dateien befällt und sich so reproduziert. Die befallene Datei heißt ***Wirt***. Ein Virus muß sich mindestens zweimal (durch Befall anderer Dateien) reproduzieren können. Das heißt, auch von dem Virus befallene Dateien können wieder andere Dateien infizieren. Ein ***Virus***

[8] siehe [Brunnstein 1999]
[9] siehe [Brunnstein 1999]
[10] siehe [RetschTode 2000], S.12
[11] siehe [Brunnstein 1999]

verbreitet sich auf lokalen Systemen; verbreitet sich maliziöse Software selbstständig über Netzwerke, so wird sie als ***Wurm*** bezeichnet.

Anti-Malware-Software

Anti-Malware-Software sind Programme, die den Benutzer vor Malware schützen. ***Antivirenprogramme*** schützen den Benutzer vor ***Viren*** (und auch anderer Malware), indem sie mit Hilfe von ***Signaturen*** (s.u.) bösartige Software erkennen, deren Ausführung und Verbreitung verhindern und befallene Dateien säubern.

In dieser Arbeit werden die Begriffe *Anti-Malware-Software* und *Anti-Virus-Software* als Synonyme verwendet, obwohl eigentlich *Anti-Malware-Software* ein Oberbegriff von *Anti-Virus-Software* ist. Die Begriffe werden gleichbedeutend benutzt, da viele als Anti-Virus bezeichnete Programme den Benutzer auch vor anderer Malware als Viren (wie zum Beispiel trojanischen Pferden) schützen. Deshalb ist mit der Bezeichnung *Anti-Virus-Software* eigentlich *Anti-Malware-Software* gemeint. Die Begriffe ***Antivirenprogramm, Anti-Malware-Programm*** und ***Anti-Virus-Scanner*** werden ebenfalls als Synonyme für ***Anti-Malware-Software*** verwendet.

Scannen / Scanner

Mit ***Scannen***[12] bezeichnet man das Überprüfen von Dateien oder Verzeichnissen auf bösartige Objekte durch ein Anti-Malware-Programm. Deswegen werden Antivirenprogramme auch ***Anti-Virus-Scanner*** genannt. Meldet ein Anti-Malware-Programm fälschlicherweise eine saubere Datei als maliziös, so bezeichnet man das als ***false positive***[13].

Beim ***Scannen*** können die meisten Antivirenprogramme auf Anweisung eine sogenannte ***Protokolldatei*** (auch ***Reportdatei*** genannt) erzeugen, in der die Ereignisse und Entdeckungen von bösartiger Software beim ***Scannen*** von Verzeichnisstrukturen protokolliert werden. Dadurch kann ein Scanvorgang im nachhinein betrachtet und ausgewertet werden.

On-demand Modus / On-access Modus

Anti-Virus-Programme können bösartige Software (Malware) grundsätzlich in zwei verschiedenen Betriebsarten erkennen:

[12] engl. "to scan": genau oder kritisch prüfen

[13] wobei mit *positive* das positive Testergebnis bei der Untersuchung einer Datei auf maliziösen Befall bzw. Inhalt analog zu medizinischen Untersuchungen gemeint ist. *False positive* bedeutet ein falsches positives Testergebnis, also eine "Fehldiagnose" des Scanners.

Im ***On-demand Modus*** wird die Software bei Bedarf (engl. "demand") aktiviert und zur Überprüfung (***Scannen***) von Dateien oder Verzeichnissen eingesetzt. Mit ***On-demand Testen*** wird das Testen von Software im ***On-demand Modus*** bezeichnet.

Im ***On-access Modus*** ist die Software im Hintergrund aktiv und überwacht die Aktivitäten auf dem Rechner. Sie überprüft bei jedem Dateizugriff (engl. "access") automatisch im Hintergrund die entsprechende Datei auf bösartige Software. Mit ***On-access Testen*** wird das Testen von Software im ***On-access Modus*** bezeichnet.

Signatur / Engine / Update

Signaturen sind bestimmte Muster, durch die Anti-Malware-Software maliziösen Code erkennen kann. Eine regelmäßige Aktualisierung (***Update***) der Signaturen eines Scanners ermöglicht die Erkennung von Malware, die seit der letzten Aktualisierung von ***Signaturen*** neu entdeckt wurde. Durch Signaturen wird also der Kenntnisstand des Anti-Malware-Programmes auf den neuesten Stand gebracht.

Als ***Engine*** eines Anti-Malware-Programmes bezeichnet man den Teil der Software, der die Erkennung von Malware durch Signaturen zur Ausführungszeit steuert. Bei Auftauchen einer neuen Art von Malware oder Viren muß auch die ***Engine*** von Anti-Malware-Programmen aktualisiert werden, damit gänzlich neue Signaturtypen verarbeitet und entsprechende Erkennungsmechanismen ausgeführt werden können.

Datenbank / Testmenge

Der Begriff ***Datenbank*** wird in der vorliegenden Arbeit mit zwei verschiedenen Bedeutungen verwendet: Erstens als nach bestimmten Kriterien eingeteilte ***Testmenge*** von maliziösen Objekten und zweitens als eine Datenbank nach dem ***relationalen Datenmodell*** zur redundanzfreien Speicherung von großen Datenmengen.

Anti-Malware-Software wird auf einer ***Testmenge*** (engl. ***testbed***) von ***Musterdateien*** (engl. ***samples***) getestet. Im Hauptteil dieser Arbeit bezeichnet der Begriff ***Datenbank*** eine bestimmte ***Testmenge*** von maliziösen Objekten. Jede Datenbank in diesem Sinne enthält Malware oder Viren einer bestimmten Kategorie (vgl. 1.2.2 und 1.2.3). Die Festsetzung des Inhaltes einer Datenbank mit Malwareobjekten zu einem bestimmten Zeitpunkt bezeichnet man als das ***Einfrieren*** der Datenbank; neu erscheinende Objekte werden nach dem ***Einfrieren*** nicht mehr in die ***Datenbank*** eingefügt.

In Kapitel 6 und 7 dieser Arbeit wird die ***relationale Datenbank*** "VTED" vorgestellt. Hierbei handelt es sich um eine Datenbank im eigentlichen Sinne, in der Daten strukturiert gespeichert und analysiert werden können. In den beiden genannten Kapiteln sollte aus dem Zusammenhang klar werden, ob mit Datenbank die Datenbank "VTED" oder eine Malware-Datenbank (also eine Testmenge) gemeint ist.[14]

1.2.2 Klassifikation von Malware

Die Gesamtmenge aller bösartigen Software kann man grundlegend danach unterscheiden, ob die Verbreitung auf einem Einzelsystem oder über ein vernetztes System stattfindet. Zusätzlich lässt sich Malware danach differenzieren, ob die bösartige Software selbstreplizierend ist oder nicht. Selbstreplizierende Software verbreitet sich bei Ausführung selbst, nicht-selbstreplizierende Software verbreitet sich nicht bei der Ausführung.

Nach diesen zwei Unterscheidungsmerkmalen lassen sich vier Gruppierungen von Malware vornehmen (in Klammern jeweils der Oberbegriff für diese Art von Malware)[15]:

- selbstreplizierende Software auf einem Einzelsystem (Viren)
- selbstreplizierende Software auf vernetzten Systemen (Würmer)
- nicht-selbstreplizierende Software auf einem Einzelsystem (Trojanische Pferde)
- nicht-selbstreplizierende Software auf vernetzten Systemen (Hostile Applets)

Viele maliziöse Softwareobjekte sind nicht eindeutig einer der oben genannten Kategorien zuzuordnen, da sie Elemente mehrere Kategorien in sich vereinen. So gibt es zum Beispiel viele Würmer, die auf befallenen Systemen trojanische Pferde oder Viren installieren[16].

1.2.3 Klassifikationen von Viren

Da es sich bei einem Großteil der verbreiteten bösartigen Software um Viren handelt, haben sich verschiedene, allgemein akzeptierte Klassifikationen für diese spezielle Art von Malware durchgesetzt. Zum einen kann man Viren nach der Plattform, für den der Virus programmiert wurde und auf dem er läuffähig ist (das heißt, auf dem er sich durch Replikation verbreiten kann), einteilen. Zum anderen werden Viren häufig danach eingeteilt, ob sie verbreitet sind oder ob es sich um seltene, nicht häufig gefundene Viren handelt.

Eine Einteilung in verschiedene Klassen ist sinnvoll, um einen Überblick über die große Masse an verschiedenen Viren zu erlangen und um neu auftauchende Exemplare einordnen

[14] In nicht eindeutigen Fällen wird letztere Art von Datenbank explizit als Malware-Datenbank bezeichnet, um zu verdeutlichen, daß eine Testmenge von Malware gemeint ist
[15] nach [KittelTicak 2002], S.4
[16] Ein Beispiel hierfür ist der Wurm *RemoteExplorer*, der sich unter *Windows NT*-Betriebssystemen selbst verbreitet und auf befallenen Systemen einen Virus installiert, der lokale Dateien befällt. Weitere Informationen zu diesem Wurm finden sich unter [RemoteExplorer 2002].

zu können. Allerdings lassen sich einige Viren nicht eindeutig zu den beschriebenen Klassen (s.u.) zuordnen. Da anhand dieser Klassifikationen auch die Malware-Datenbanken des Virus-Test-Centers strukturiert sind, werden in den folgenden beiden Unterabschnitten die Klassifikation nach Plattform und nach Verbreitung kurz erläutert.

1.2.3.1 Klassifikation nach Plattform

Jede Art von programmiertem Computercode ist jeweils nur auf einer bestimmten Plattform lauffähig. Da ein Computervirus jeweils nur auf dieser Plattform ausgeführt werden kann[17], kann man Viren nach der Plattform, für die der Code geschrieben wurde, unterscheiden:

- Boot-Viren
- Datei-Viren (engl. *file viruses*)
- Makro-Viren (engl. *macro viruses*)
- Skript-Viren (engl. *script viruses*)

Boot-Viren

Bootviren benutzen als Wirt keine Anwendungsprogramme. Stattdessen infizieren sie das System selbst, weshalb sie auch als Systemviren bezeichnet werden. Kittel und Ticak beschreiben die Funktionsweise von Bootviren folgendermaßen ([KittelTicak 2002], S.6): "Disketten und Festplatten, von denen ein Computer gebootet werden kann, enthalten in Systembereichen ausführbaren Code, der beim Systemstart ausgeführt wird. Diese Tatsache nutzen diese Viren aus, indem sie ihren eigenen Code dorthin schreiben und somit zuverlässig zur Ausführung gelangen." Bootviren befallen Boot-Sektoren und Master-Boot-Sektoren von Festplatten und Disketten und werden beim Start über die befallene Festplatte oder Diskette ausgeführt.

Datei-Viren (File-Viren)

Dateiviren, im weiteren Verlauf der Arbeit auch mit dem international üblichen, englischen Ausdruck "File-Viren" bezeichnet, befallen ausführbare Dateien. Dabei infiziert der Virus die Wirtsdatei so, daß bei einem Aufruf der Wirtsdatei der Virus ausgeführt wird. Je nach Virus kann dabei der Code der Wirtsdatei gar nicht, vor oder nach Ausführung des Viruscodes ausgeführt werden. Dateiviren kann man noch detaillierter anhand der Art des Befalls von Dateien unterteilen. Man unterscheidet folgende Typen[18]:

- *prepending viruses*:
  Diese Viren schreiben ihren Code vor den Code des Wirtsprogrammes, so daß der Viruscode zuerst ausgeführt wird und danach der ursprüngliche Code des Wirtsprogrammes.

[17] Ausnahme: sogenannte Multi-Partite Viren, die auf mehr als einer Plattform lauffähig sind
[18] vgl. [Ferbrache 1992], S. 31-35, [Pfleeger 1997], S. 180ff und [KittelTicak 2002], S. 6f

- *appending viruses*:
  Solche Viren hängen ihren Code hinter den Code des Wirtsprogrammes. Damit der Viruscode trotzdem zuerst ausgeführt wird, fügt der Virus beim Befall der Wirtsdatei zusätzlich einen Sprungbefehl an den Anfang der Datei ein, der auf den ersten Befehl des Viruscodes zeigt. Nach Ausführung des Viruscodes kann ein weiterer Sprungbefehl auf den ursprünglichen Code des Wirtsprogrammes zeigen.
- *cavity viruses*:
  Cavity-Viren[19] schreiben ihren Code in Freiräume der befallenen Dateien. Diese Art des Befalls ist nicht einfach zu realisieren, denn es müssen vom Virus beim Befall einer Datei ungenutzte Code-Segmente gefunden werden, in die der Virus seinen Code schreiben kann.
- *shell viruses*:
  Der Code des Wirtsprogramms wird komplett in den Viruscode integriert und kann als Subroutine vom Virus aufgerufen und gesteuert werden.
- *overwriting/replacing viruses*:
  Das Wirtsprogramm wird vom Virus überschrieben und ist danach nicht mehr ausführbar. Diese Art von Virus wird schnell entdeckt, da der ursprüngliche Programmcode der Wirtsdatei zerstört ist und somit bei Aufruf der Datei nicht mehr ausgeführt wird.

Makro-Viren

Heutzutage bieten viele Anwendungsprogramme (wie zum Beispiel Microsoft Word, Excel, Powerpoint oder Access) die Möglichkeit, in Dokumenten der jeweiligen Anwendung auch ausführbaren Code - etwa zur automatisierten Ausführung von Operationen auf den Dokumenten - zu entwerfen und in Dokumenten zu speichern. Solche automatisierten Operationen bezeichnet man als Makros.

Da es sich bei Makros um ausführbaren Code handelt, der insbesondere die in den Dokumenten enthaltenen Daten verändern, aber auch Betriebssystemfunktionen nutzen kann, gibt es eine Vielzahl von sogenannten *Makroviren*, die entsprechende Dokumente von Anwendungsprogrammen befallen. Die Viren sind dabei in der jeweiligen Makro-Programmiersprache der entsprechenden Anwendung geschrieben (z.B. *Visual Basic for Applications* für Microsoft-Office Dokumente oder *Lotus-Script* für Lotus Smartsuite Dokumente). Zur Ausführung gelangen die Viren meist schon beim Öffnen eines befallenen Dokumentes der entsprechenden Anwendung.

[19] engl. cavity: Aushöhlung, Hohlraum

Skript-Viren

Skriptviren befallen Skripte von sogenannten Skriptprogrammiersprachen wie *Java-Skript* oder *Visual-Basic-Skript*. Skripte dieser Sprachen sind im Internet und auf Webseiten weit verbreitet. Bei Skripten handelt es sich nicht um direkt ausführbaren Maschinencode, wie etwa bei Dateiviren (die ausführbare Dateien befallen), sondern um Programmcode, der über einen Interpreter (wie zum Beispiel den *Windows Scripting Host* unter Microsoft *Windows* Betriebssystemen) zur Laufzeit eingelesen und ausgeführt wird. Je nach Programmiersprache und Laufzeitumgebung können Skripte unterschiedlich mächtig sein, und dementsprechend großer Schaden kann durch bösartige Skripte entstehen.

1.2.3.2 Klassifikation nach Verbreitung

Eine Aussage, wie hoch die Gefahr ist, sich mit einem bestimmten Virus zu infizieren, kann über die Analyse der Verbreitung von Viren getroffenen werden. Viele Hersteller von Anti-Viren-Software stellen auf ihren Webseiten Übersichten bereit, auf denen sie angeben, welche Viren am meisten verbreitet sind ("Top 10" und ähnliches). Diese Übersichten basieren auf den Daten des jeweiligen Herstellers. Obwohl einige Hersteller von Anti-Malware-Software über eine große Anzahl an Kunden verfügen, sind solche Auflistungen nicht notwendigerweise repräsentativ für die tatsächliche Verbreitung bestimmter Viren.

Eine allgemein akzeptierte Einteilung von Viren nach ihrer Verbreitung ist die Einteilung der *WildList Organisation* ([Wildlist 2002]). Die WildList Organisation teilt Viren in zwei Klassen ein:

- Viren, die als *in-the-wild* betrachtet werden
- Viren, die nicht als *in-the-wild* betrachtet werden

Mit *in-the-wild* sind Viren gemeint, die im Umlauf sind und ahnungslose Benutzer befallen. Die Entscheidung, ob ein Virus *in-the-wild* ist oder nicht, wird anhand von Meldungen durch "Reporter" (Experten, die Virenvorfälle an die WildList Organisation melden) vorgenommen. Wird ein Virus innerhalb eines Monats von mindestens zwei "Reportern" gemeldet, so gilt er als *in-the-wild*. In der sogenannten *WildList* wird einmal pro Monat eine Auflistung aller als *in-the-wild* eingestuften Viren veröffentlicht.

Beim Testen von Anti-Malware-Software werden die Viren, die auf der aktuellen WildList stehen, als *in-the-wild* Viren bezeichnet (abgekürzt ITW). Die restlichen Viren einer Testmenge bezeichnet man als Zoo-Viren, da sie kaum verbreitet sind und sich deshalb - die Analogie zu Tieren aufrechterhaltend - nicht in "freier Wildbahn" sondern in einem "Zoo" befinden.

1.3 Das Virus Test Center der Universität Hamburg

Den Hintergrund der vorliegenden Arbeit und die Basis für die realisierten Erweiterungen liefert das Virus Test Center der Universität Hamburg (siehe [VTC 2002]). Als Hauptstudiumsprojekt im Vertiefungsgebiet IT-Sicherheit bzw. im Studienprofil ISO werden im Virus Test Center unter Leitung von Prof. Dr. Klaus Brunnstein regelmäßig Tests von Anti-Malware-Software durchgeführt. Studenten arbeiten in verschiedenen Bereichen an dem Projekt mit (vgl. Kapitel 4) und können so den wissenschaftlichen Umgang mit bösartiger Software und das wissenschaftliche Testen von Anti-Malware-Software erlernen.

Für das Testen von Anti-Malware-Software hat das VTC durch jahrelange Erfahrung eine Methodik entwickelt (eine genaue Beschreibung der VTC-Testmethodik findet sich in Kapitel 4), mit der sich Testprodukte teilweise automatisiert auf großen Datenbanken von bösartiger Software testen lassen. Zum Abschluß eines jeden Tests wird ein ausführlicher Testbericht mit detaillierten Testergebnissen und Zusammenfassungen veröffentlicht.

Das VTC wurde von Vesselin Bontchev, einem ehemaligen Doktoranden von Prof. Brunnstein am Arbeitsbereich AGN des Fachbereichs Informatik, gegründet. Ursprünglich wurden auch Viren analysiert und wissenschaftlich untersucht, daher der Name Virus Test Center. Inzwischen besteht allerdings der Großteil der Arbeit im VTC-Projekt im Testen von Antivirenprogrammen. Deshalb heißt das Virus Test Center (zum Beispiel auf der Webseite) auch aVTC (für Antiviren Test Center), da ja Antivirenprogramme und nicht Viren getestet werden. In dieser Arbeit wird jedoch die international bekannte Bezeichnung VTC verwendet.