Hauptseminar ‚Malicious Code'

Virentechniken: Analyse und Metamorphismus

Holger Pawlita

Abstract. Vorliegende Ausarbeitung beginnt mit einer Einführung in das Thema und einer Definition des Begriffs „Virus". Im Hauptteil werden zunächst die diversen Virenarten erläutert und zu ihnen einige Beispiele genannt. Sodann wird auf die Techniken eingegangen, die moderne Viren benutzen, um sich vor Scannern zu schützen, sich zu replizieren oder einfach nur das System zu schädigen. Auch hier runden einige Beispiele die Darstellung ab. Abschließend wird ein Ausblick auf die voraussichtliche Entwicklung der Virentechniken in der PC- und Server-Welt für die nächsten Jahre gegeben.

1 Einleitung

Viren sind eine ernstzunehmende Gefahr. Sie können zwar die Hardware nicht beschädigen, führen aber zu Daten- und Zeitverlust, Zusammenbruch von Netzwerken, Unterbrechung kritischer Geschäftsprozesse etc. Für den privaten Anwender mag dies verschmerzbar sein, so sind sie für Unternehmen ein grosses Problem. Laut Trend-Micro, einem der weltweit grössten Hersteller von Antiviren-Software, belief sich der Schaden, der durch Virenattacken ausgelöst wurde, im Jahre 2003 auf 55 Milliarden US-Dollar. Das entspricht einer Verdopplung der Kosten zum Jahr 2002. Angesichts dieser enormen Zahlen ist das Gebiet Security immer mehr ins Sichtfeld grosser Unternehmen gestossen. Gerade für CIOs und das IT-Management ist sie ein grosses Thema. Die Kenntnis der verschiedenen Virenattacken und geeigneter Tools zu ihrer Abwehr ist für die betroffenen Unternehmen und für die Softwareindustrie von hohem Interesse.

2 Definition

Unter einem Virus versteht man ein Stück Programmcode - meistens in Assembler oder Macroscript geschrieben -, das selbstständig läuft, sich selbst immer wieder replizieren kann und sich an Dateien oder Laufwerke anhängt. Einige Viren sind so konstruiert, dass sie sich selbst über das Netzwerk verschicken und Sicherheitssysteme umgehen können. Viren sind Programme, die sich durch Befall und Änderung anderer (Programm)-Dateien verbreiten. Diese schädlichen Programme benötigen also einen Wirt. Manche Viren löschen oder verändern Daten, die sich auf dem PC befinden.

3 Taxonomie der verschiedenen Virengruppen

3.1 Bootviren

Bootviren sind Viren, die sich im Bootblock bzw. im MBR[1] oder DBR[2] von Festplatten oder im FBR[3] von Disketten einnisten. Jeder Datenträger besitzt so einen speziellen Sektor, der zuallererst geladen wird. Im Gegensatz zu nicht-bootfähigen Trägern, beinhaltet dieser bei bootfähigen Trägern wichtige Informationen wie Initialisierung der Hardware oder Routinen zum Testen. Wird ein System von einem infizierten Datenträger gebootet, wird zuerst der Virus ausgeführt, der sich dadurch auf Bootblocks anderer Träger verbreiten kann. Der MBR wird gelesen und analysiert, um die physikalische Adresse der Bootsektoren zu finden. Ein Virus dieser Sorte lagert den originalen Bootsektor an eine andere Stelle des Datenträgers aus und überschreibt zusätzlich den Bootsektor mit seinem eigenen Virencode. Dieser installiert den Virus im Arbeitsspeicher und startet letztendlich den originalen verschobenen Bootblock, der normalerweise das OS[4] lädt oder jetzt eigene Prozeduren ausführt.

Noch aggressivere Varianten wie der *Saddam-Hussein*-Virus verschlüsseln zusätzlich den verschobenen MBR und unterbinden jeglichen Zugriffsversuch darauf. Andere Beispiele: *Michelangelo, SubSeven.*

3.2 Dateiviren

Die Dateiviren zählen allgemein als die häufigste Virenart. Ihre Eigenart besteht darin, dass sie ausführbare Programme befallen. Die meisten Computerviren haben sich auf eine Dateiart spezialisiert, z.B. EXE, COM, SYS, DLL...

Wird ein infiziertes Programm ausgeführt, wird vor der eigentlichen Programmausführung der Virus gestartet, der sich daraufhin in andere Programme hineinkopiert und durch deren Weitergabe auf andere Systeme verbreiten kann.

3.2.1 Überschreibende Dateiviren

Diese Virenart zerstört bei ihrem Aufruf die Originaldatei entweder komplett oder teilweise bzw. ersetzt Programmcode und macht damit das Programm unwirksam. Ihre Größe ist sehr gering. Ihre Verbreitung ist nicht sehr groß, fallen sie doch durch die Zerstörung der Programme recht schnell auf. Beispiel: *Banana*

3.2.2 Verlängernde/Parasitäre Viren

Ein Vertreter dieser Gruppe lässt i.A. die Originaldatei bzw. das Wirtsprogramm unbeschädigt. Stattdessen schreibt sich der Virus entweder an das Ende des Programms oder setzt sich an den Anfang des Wirtes und schiebt ihn nach hinten.

Bei der zweiten Variante kopiert der Virus die ersten Bytes eines Programms ans Ende, hängt den eigentlichen Virencode an und überschreibt den gesicherten Anfang des Programms durch einen Sprungbefehl auf den Virencode. Beim Start des Programms erfolgt dann ein Sprung auf den schädlichen Code. Dieser installiert den Virus im System, kopiert den gesicherten Anfang des Programms wieder an den Anfang

---

[1] Master Boot Record
[2] DOS Boot Record
[3] Floppy Boot Record
[4] Operating System

zurück und startet das Programm regulär durch einen Sprung auf diesen Programmanfang.

3.2.3 Nicht-verlängernde Viren (Cave/Spacefiller/Cavity)

Manche Compiler erzeugen in ihren übersetzten Programmen an bestimmten Stellen Höhlen (Caves) mit langen Folgen von sich wiederholenden Zeichen, oftmals binäre Nullen. In diese Caves nistet sich der Virus ein und speichert zusätzlich die Anzahl der vorhandenen Stellen. Damit macht der Virus eine spätere Wiederherstellung des Originalcodes möglich. Eine Aufteilung auf mehrere Höhlen beherrschen manche Vertreter wie der *CIH*.

Beispiele: *Jerusalem*, *AntiEXE* oder *CIH*.

3.3 Makro-Viren

Dieser Virus bedient sich der mächtigen Makrosprache der Textverarbeitung. Makroviren unterscheiden sich von den anderen Viren darin, dass sie keine ausführbaren Programme befallen, sondern Datendateien, die beim Öffnen den Makro-Virus starten und im Weiteren andere Dateien dieses Typs schädigen.

Der erste bekannte Word-Makro-Virus war *DMV*. Der erste Virus itw[5] hiess *Concept*, der sich einer extremen „Beliebtheit“ erfreute, da es damals noch keine Sicherheitsvorkehrungen gab.

Word-Basic war eine vollständige Programmiersprache und erlaubte so die Erstellung von Viren, die auf allen Systemen mit Word- oder Office-Umgebung liefen. Die Makros wurden dann von den Programmen interpretiert und erlaubten die ersten system-unabhängigen „cross-platform“-Infektionen. Damit wurde ein Grundsatz, der bis dahin galt, verletzt und nichtig gemacht: nämlich, dass ein Virus nicht von einem zu einem anderen Typen von Computer wechseln kann.

Beispiele: *Nuclear*, *WM/ Concept*, *WM97/Melissa*

3.4 Skript-Viren

Ähnlich zu den Batch-Viren versteht man unter Skript-Viren ganz normale Programm-Viren, die kein ausführbares Programm darstellen, sondern ein interpretiertes Skript. Im Gegensatz zu den normalen Viren modifizieren diese aber keine Programme, sondern erstellen eigene Dateien mit dem Skript-Viren-Code. Das Programm verwendet dazu bestehende Datendateien, löscht diese und legt eine entsprechende Skript-Datei mit doppelter Dateiendung an. Beim Anklicken der Datei wird dann aber der Skript-Virus ausgeführt und nicht die vermeintliche Datendatei geöffnet. Beispiel: Anna Kournikova!.jpg.vbs.

3.5 Weitere Virenarten

- Source-Code-Viren
  Diese Viren können sich in Programm-Quellcode einkopieren und dann beim Kompilieren und nachfolgendem Ausführen des Programms aktiviert werden.

[5] itw = in the wild

- Companions
  Companion-Viren verändern nicht den Programmcode, sondern legen eigene Programmdateien mit einer anderen Endung an, die dann statt des eigentlichen Programms ausgeführt werden.
- Batch-Viren
  Sie sind nichts anderes als DOS-Batch-Dateien, die andere Batch-Dateien infizieren können, indem sie diese durch eigenen Code überschreiben.
- Multipartite Viren
  Unter multipartiten Viren, die auch als Hybridviren bekannt sind, versteht man Kombinationen aus Boot- und Dateiviren. Der Vorteil für die programmierenden Angreifer ist die Ausnutzung beider möglichen Infektionswege. Beispiel: *Tequila*

4 Virentechniken

Nach der einführenden Beschreibung der wichtigsten Viren und deren Derivaten, werden nun einige Konzepte bzw. Techniken vorgestellt, die moderne Viren nutzen, um vor den Anti-Viren-Scannern unbemerkt zu bleiben.

4.1 Stealth

Bestimmte Viren (auch unter dem Namen „Tarnkappenviren“ bekannt) bedienen sich sog. Stealth-Techniken, um ihre Anwesenheit zu verschleiern. Sie nutzen dabei die Möglichkeit, Systemkommandos zu modifizieren. Einige Viren prüfen z.B. beim Öffnen einer Datei, ob eben diese Datei schon viral ist und filtern dann beim Einlesen den Virus-Code heraus, der daraufhin ausgeführt wird. So werden die Virenscanner überlistet, die ja auf die Systemkommandos wie Öffnen und Lesen einer Datei angewiesen sind und so die Infektion nicht erkennen.

Andere nisten sich in Systemkommandos ein, um die veränderte Längenänderung einer Datei, die sich durch die Infektion zwangsläufig ergibt, vor den Virenscannern zu verschleiern. Dadurch dass sie jeglichen Zugriff auf die infizierten Wirte kontrollieren, sind diese Viren meist schwer feststellbar.

Der Virus *Brain*, der auch der erste DOS-Virus war, besaß eine derartige Stealth-Routine. Dabei lenkte er auf physikalischer I/O-Ebene Zugriffe auf *Brain*-infizierte Bootsektoren auf den originalen Bootsektor um. Nachteil dieser Technik besteht darin, dass der Virus resident im Speicher sein muss, um solche Modifikationen durchführen zu können. Virenscanner können dies aber beim Durchsuchen des Speichers feststellen.

4.2 Tunneln

Diese Virenart versucht hauptsächlich Antiviren-Software auszuhebeln. Sie suchen nach den originalen Interrupt-Handlern für DOS und BIOS, um diese dann direkt aufzurufen. Hierdurch werden eventuelle Wächterprogramme unter DOS umgangen, die sich gerade in diese Interrupts eingeklinkt haben, um Virenaktivitäten erkennen zu können. Zum Beispiel blockiert der Virus *W32/MTX* sowohl E-Mails an Antiviren-Hersteller als auch Aufrufe an deren Homepages. So ist es dem Opfer nicht möglich, ein Antiviren-Programm oder ein Update herunterzuladen.

4.3 Selbsterkennung

Um eine mehrfache Infektion zu vermeiden und dadurch eventuell aufzufallen, wie etwa die Urform des *Jerusalem*-Virus, muss sich der Virus irgendwie selbst identifizieren. Unter speicherresidenten DOS-Viren ist die Einrichtung eines "Are You There“-Calls beliebt. Dabei wird eine MS-DOS Systemfunktion verbogen, sodass der Aufruf mit dem vom Betriebssystem normalerweise ignorierten Parameter einen definierten Rückgabewert ergibt. Eine andere Möglichkeit, die nur sehr wenige Viren nutzen, ist, so wie Antiviren-Programme den Arbeitsspeicher nach sich selbst zu scannen. Entweder wird nach einer Kennmarke gesucht oder nach dem Viruscode selbst.

4.4 Speicherresidente Viren

Wenn sich nach Beenden eines infizierten Programms der Viruscode oder Teile im Arbeitsspeicher befinden, so spricht man von einem TSR[6]- oder residenten Virus. Diese Viren bleiben nach der ersten Ausführung im Arbeitsspeicher aktiv bis zum Ausschalten des PCs. Dabei klinken sie sich in der Regel in Systemfunktionen/Interrupts ein. Wenn eine dieser abgefangenen Systemfunktionen aufgerufen wird, erhält zuerst der Virencode die Kontrolle, und der Virus kann eine Datei oder einen Sektor infizieren, auf den das System zugreift. Danach wird die Kontrolle an die eigentlich aufgerufene Systemfunktion übergeben. Manche Viren wie der *DenZuk* schaffen es sogar, einen Softboot (Strg+Alt+Entf) zu überleben.

4.5 Selbstverschlüsselung

Bei dieser Technik versucht der Virus der Suchmethode nach Zeichenfolgen (sogenannte „Scanstrings“) durch Verschlüsselung zu entgehen. Die Viren verschlüsseln dabei den Hauptteil ihres Codes mit einem veränderlichen Schlüssel, um bei Scans nicht allzu anfällig zu sein, und lassen nur die Entschlüsselungsroutine unverschlüsselt im Viruscode. Verschlüsselte Viren bestehen somit aus mindestens einer Entschlüsselungsfunktion (Decryptor) und dem Hauptcode. Der Decryptor entschlüsselt den Hauptvirencode, damit dieser ausgeführt werden kann. Verschlüsselte Viren verwenden in der Regel feste oder variable Entschlüsselungsroutinen, während der Decryptor eines polymorphen Virus zufällig aus Prozessoranweisungen generiert wird und zahlreiche Befehle enthält, die für den Entschlüsselungsvorgang nicht benötigt werden. Am Beginn des Codes befindet sich eine Entschlüsselungsroutine, die dann den Virusbody direkt in den Arbeitsspeicher entpackt. Diese Viren haben eine Schwachstelle: nämlich die Entschlüsselungsroutine selbst, denn bei den einfachen Viren ändert sich diese nicht. Eine andere gebräuchliche Art der Verschlüsselung ist es, den Virus mit einem Laufzeitpacker wie PkLite, Diet oder UPX zu komprimieren.

4.6 Polymorphismus

Polymorphe Viren verschlüsseln sich ebenfalls. Aber im Gegensatz zu den eigentlichen Verschlüsselungsviren benutzen sie neben einer unbegrenzten Anzahl von Decryptoren zusätzlich mehrere verschiedene Verschlüsselungsmethoden. Das macht die

[6] Terminate and Stay Resident in memory

Arbeit für Virenscanner deutlich schwerer. Müssen sie nun nicht nur eine, sondern gleich mehrere verschiedene Verschlüsselungsroutinen erkennen.

Wie der Name schon sagt[7] können polymorphe Viren ihre Form ändern, d.h. sie enthalten keine festen Zeichenfolgen, und können sich somit vor Virenscannern als harmlos darstellen. Polymorphe Viren richten sich hauptsächlich gegen Signatur-Scanner, die das Dateisystem nach bestimmten Codesequenzen absuchen. Der Decryptor ändert sich komplett bei jeder neuen Infektion, so dass keine Kopie aussieht wie die andere, aber trotzdem seine Funktion gleich bleibt. Sie können ihren Code so variieren, dass unter mehreren verschiedenen Virenvarianten kaum gleiche Codestücke zur Verwendung in Virenscannern vorhanden sind. Jedes Mal, wenn der Virus eine Wirtsdatei befällt, verändert er die Entschlüsselungsroutine und/oder den Chiffrierschlüssel. Die Chiffrierung maskiert den Hauptteil des Virus, so dass eine Erkennung mittels einer Signatur nur schwer möglich ist.

Man unterscheidet bei den polymorphen Viren verschiedene Komplexitätsgrade. Z.B. könnten *mehrere Decryptoren* verwendet werden, von denen einer ausgesucht wird. Weiter lassen sich für jedes Teilstück *variable Instruktionen* verwenden, von denen jedesmal eine ausgewählt wird. *Garbage* bzw *Junk Code* sind Codestücke, die im Grunde völlig nutzlos sind, bzw. den Decryptor in seiner Funktionsweise nicht beeinträchtigen. Diese verhindern die Erzeugung von Scanstrings, da jedes Mal andere neutrale Befehle in den Decryptor eingefügt wird. Es gibt polymorphe Viren, die *Calls* und *Jumps* in den Decryptor einfügen, so dass der Code undurchsichtiger wird. Zusätzlich ist die *Vertauschung der Reihenfolge* der Instruktionen möglich. Polymorphe Engines, die alle vorangegangenen Fähigkeiten besitzen, gelten als recht effektiv. *Permutation* ist eine ganz andere Technik. Hierbei wird der Virus in verschiedene Teile zerstückelt (Suche nach Dateien, Infektion, Payload[8]) und die Reihenfolge dieser Teile vertauscht.

Ähnlich wie bei den Virus Construction Kits (Virenbaukästen) erschienen bald die ersten fertigen polymorphen Verschlüsselungsgeneratoren wie die MtE[9], die als OBJ-Datei wie ein Baustein nur noch in einen bereits bestehenden Virus eingebunden werden und so jeden möglichen Virus mit polymorphen Eigenschaften aufrüsten konnten. Die MtE ist der am weitesten verbreitete polymorphe Codegenerator und findet z.Zt. in ca. 36 verschiedenen Viren seine Anwendung.

4.7 Oligomorphismus

Eine kleine Abart zum Polymorphismus bildet der Oligomorphismus. Bei der Entschlüsselung entnehmen oligomorphe Viren die Schlüssel im Gegensatz zu polymorphen Viren, die ihre Schlüssel unbegrenzt neu erzeugen können, aus einer Liste.

4.8 Metamorphismus

Metamorphismus ähnelt sehr dem Polymorphismus. Aber während polymorphe Viren nur verschiedene Entschlüsselungsroutinen generieren können, können metamorphe Schädlinge ganzen variablen Code erzeugen und besitzen keinen Decryptor. Man muss sich die Meta-Engine wie eine Black Box vorstellen, in den man den gewünschten zu verändernden Code eingibt und diese dann daraus den veränderten Code ausgibt,

---

[7] *griech*. poly ~ viel und morphein ~ Gestalt

[8] erst durch die darin enthaltenen Schadensroutinen wird die Malware zum Schädling

[9] Mutation Engine

der sich syntaktisch vom Input unterscheidet, aber von der Funktionalität gleich ist. Sie besitzen i.G. zu den polymorphen Viren keinen konstanten Virus-Body. Das macht metamorphe Viren besonders knifflig für Virenscanner.

Wieder kann man die Level einer metamorphen Engine klassifizieren nach folgenden Features: *Neugenerierung des Virus-Code*, *Code Reduction*, *Code Expansion*, *Replication und Permutation*. Die Features einer metamorphen Engine laufen pseudozufällig ab. Daher treten einige von ihnen mit sehr geringer Wahrscheinlichkeit auf, d.h. die Länge des Codes kann bei jeder Ausführung des Virus variieren. Zusätzlich kann man die Komplexität auch noch hinsichtlich ihrer Schichten einteilen. *Win32.LTX* z.B. kann die Meta-Engine vier- bis siebenmal aufrufen und so über dem eigentlichen Virus-Code vier bis sieben Schichten aufbauen.

4.9 EPO[10]

Der erste parasitäre Virus *Impanate* verbarg sich bei einer Infektion, indem er sich in den Programmfluss eintrug ohne dabei den originalen Einsprungspunkt („Entry Point“) der Software zu ändern. Unter dem „Entry Point“ versteht man i.A. die Stelle, an der der eigentliche Programmcode in der Software beginnt. Er geht also bei der Datei-Infektion nicht den einfachen Weg und schreibt seinen Code an den Anfang oder Ende der zu infizierenden Datei, sondern schreibt sich in irgendeine Funktion in der Mitte. Auf diese Weise wird die Entdeckung durch Anti-Viren-Software sehr erschwert. Der Virus-Code wird dann aber nur ausgeführt, wenn die Routine, die den Virus enthält, aufgerufen wird. Wenn diese Routine nur selten ausgeführt wird, kann es sein, dass der Virus eine lange Zeit untätig ist. Beispiele: *Zmist*, *Simile.D*

4.10 Weitere Techniken

- **AntiBait**
  Dateien werden vor dem Infizieren auf AV-Köder-Eigenschaften überprüft.
- **Anti-Debugging**
  Der Virus enthält Teile, die es schwerer machen ihn zu debuggen/disassemblen.
- **Anti-Heuristic**
  Der Virus ist so programmiert, dass ein AV-Programm ihn nicht als Virus erkennt.
- **Anti-Removal**
  Der Virus enthält Code, der es erschwert, ihn zu entfernen.
- **Anti-AV/Retro-Virus**
  Der Virus greift aktiv Anti-Viren-Programme an. Weiter können derartige moderne Viren die Scanner erkennen und überlisten (Beispiel *CW-Shredder*).
- **Archiv-Infector**
  Der Virus kann ZIP oder andere Archive öffnen und enthaltene Dateien infizieren.
- **Replication**
  Der Virus sucht aktiv in der Verzeichnisstruktur nach anfälligen Dateien.

[10] Entry Point Obscuring

- **Dropper**
  Bei Befall mit einem Dropper wird der eigentliche Virus erst später installiert oder gestartet, um sich so vor dem Zugriff des Anti-Viren-Scanners zu schützen.

5 Ausblick

Wer gedacht hat, dass Malicious Code-Programmierer Computer-Fachleute sein müssen, der irrt. Selbst aus den Kinderzimmern von heute finden neue Kreationen ihren Weg ins Netz. Weiterhin verstand man früher Viren-Schreiber als Vandalen. Heute programmieren sie häufig Malware, um damit Geld zu verdienen. Schließlich gehören Virenschreiber zu den gesuchtesten Leuten bei Anti-Viren-Softwareherstellern. Angesichts immer neuer Ideen der Virenerfinder müssen sich die Virenscanner immer raffiniertere Routinen zum Echtzeitschutz ausdenken. Auf Jahre hinaus ist ein Wettlauf zwischen Viren und Anti-Viren-Software zu erwarten. Der Trend geht klar zu metamorphen Viren mit polymorphen Eigenschaften. Und da Microsoft selbst schon in den Virenscannermarkt investiert und durch die Übernahme des rumänischen Software-Herstellers GeCAD deren Virusscan-Technik eingekauft hat, zeigt doch wie lukrativ dieser Markt ist. Aber nicht nur Windows ist bei den Hackern beliebt, es kommen nun auch häufiger Schädlinge für UNIX oder Mac OS in Umlauf. Neuestes Ziel scheint die Desktopsuche zu werden. Wie ein gefundenes Fressen sammeln sie Daten und katalogisieren sie sogar und bieten so ein interessantes Ziel für Angreifer.

Die Bedrohung nimmt zu. Laut den Antiviren-Software-Herstellern haben sich die 20 bis 40 neu registrierten Viren pro Tag seit 2003 fast verdoppelt. Auch 38 % aller deutschen Unternehmen schätzen das „Virenrisiko“ momentan höher ein als noch vor einem Jahr, nur 20 % fühlen sich weniger bedroht.

6 References

- Krauß, Thomas: Viren, Würmer und Trojaner. Interest Verlag 2004
- http://www.computerbase.de/news/internet/hacker_sicherheit/2004/dezember/desktopsuche_ziel_virenschreibern/
- http://www.fuhs.de/de/fachartikel/buch/index.shtml
- http://vx.netlux.org/lib/static/vdat/epmetam2.htm
- http://www.hackernetwork.de/
- http://www.snake-basket.de/d/faq.txt
- http://www.irc-security.de/?go=viren
- http://www.f-secure.de/v-desk/virus.shtml
- http://www.ecs.syr.edu/Faculty/popyack/Lectures/Feb6/feb6.html
- http://www.securityfocus.com/virus
- http://securityresponse.symantec.com/avcenter/reference/hunting.for.metamorphic.pdf