Secure Computing: Unsichere und weniger Unsichere Systeme
Proseminar im Wintersemester 07/08

Intrusion Detection

Virenscanner

Philip Daubmeier
Technische Universität München

31.01.2008

Zusammenfassung

Diese Arbeit beschäftigt sich mit der allgemeinen Funktionsweise von Antivirus Programmen und geht dabei auch auf spezielle Virentypen, wie polymorphe und metamorphe Viren, ein. Diese haben es notwendig gemacht, neben dem klassischen Signaturscannen, bessere Methoden zur Entdeckung zu entwickeln, wie etwa heuristische Verfahren oder Software Transformatoren. Danach wird vorgestellt wie ein befallenes Programm von Viren befreit wird. Abschließend wird erörtert, wie ein bestmöglicher Schutz erzielt werden kann, und welche Gefahren uns in Form von Schadsoftware uns in Zukunft erwarten.

Inhaltsverzeichnis

1 Einführung

In momentan erhältlichen Antivirus Paketen diverser Hersteller sind eine Vielzahl verschiedener Hilfsmittel zur Abwehr von schadhafter Software, wie Viren, Würmer und Trojaner sowie eventuell auch Dialer, Spy- und Adware enthalten. Immer öfter sind auch Spamfilter für den Emailverkehr, sowie eine Firewall gegen Angreifer von aussen enthalten, um eine bestmögliche Abdeckung gegen Angreifer zu erreichen. Der potenzielle Schaden für den Nutzer kann von vielfältiger Natur sein: Im einfachsten Fall zeit- und geduldraubende Werbung in Form von Spam oder Adware bis hin zur Zerstörung wichtiger Daten oder Ausspähung privater und finanziell sensibler Daten. Diese Ausarbeitung befasst sich mit Virenscannern und erklärt insbesondere, welche Arten von Viren derzeit bekannt sind und welche Möglichkeiten der Erkennung sich anbieten.
Das Grundprinzip eines Virenscanners besteht darin, Dateien, speziell ausführbare Dateien, zu durchsuchen und auf Virenbefall hin zu Untersuchen. Dazu wird meißt der Maschinencode des Programms mit Signaturen aus einer mitgelieferten Datenbank verglichen. Diese Methode wird später näher erläutert.

2 Arten von Virenscannern

Zunächst muss zwischen drei Klassen von Virenscannern unterschieden werden, die allerdings alle die gleichen Methoden zur Erkennung und Behandlung implementieren, sich jedoch in ihrer Ausführung unterscheiden.

2.1 Echtzeit Scanner (Realtime Scanner)

Diese Art von Scanner beobachtet Lese- und Schreibvorgänge auf allen Laufwerken des PCs. Bei einigen Antivirus Programmen lässt sich einstellen, ob nur auf Lese- oder nur auf Schreibzugriffe oder beides reagiert werden soll. Bei anderen Programmen ist dies fest voreingestellt. Der Nachteil beim Überwachen jedes Lesezugriffs sind teilweise spürbare Geschwindigkeitseinbussen des Computers, da Leseoperationen viel häufiger durchgeführt werden als Schreibvorgänge. Jedoch bietet diese Form der Überwachung grösseren Schutz, da bei reiner Überwachung der Schreiboperationen zum Beispiel ein auf der Festplatte eingenisteter, aber noch nicht aktiver Virus bei dessen Ausführung nicht erkannt werden kann. Zum Zeitpunkt der Infektion war der Virenscanner vielleicht noch nicht installiert oder hatte noch keine Signatur des Virus zur Verfügung, um ihn zu erkennen. Grundsätzlich fängt

der Echtzeit Scanner also Dateien vor dem tatsächlichen Lesen oder Schreiben ab und scannt sie auf Befall von schadhaftem Code. So kann bei vorhandener Infektion der Datei der Vorgang abgebrochen werden, bevor der Virus auf dem Rechner aktiv werden kann. Oft wird der Virus dann aus der Datei entfernt - soweit dies möglich ist - oder beim Benutzer nachgefragt, ob der Virus gelöscht oder in Quarantäne verschoben werden soll, bis eine Möglichkeit gefunden wird, den Virus endgültig aus der Datei zu entfernen. Ob die Datei nach dem Entfernen des Virus noch funktionstüchtig und voll vorhanden ist, hängt von der Art des Virus ab.

2.2 Festplatten Scanner (On-Demand Scanner)

Diese Art von Scanner kommt immer dann zum Einsatz, wenn der Benutzer den Scanner explizit startet oder das Programm so eingestellt ist, dass es den Scanvorgang nach einem bestimmten Zeitplan selbst startet, etwa jede Woche am Sonntag. Dabei werden die komplette Festplatte und meist auch andere an den Computer angeschlossene Medien, wie etwa externe Festplatten oder USB Sticks, komplett nach Virenbefall hin untersucht. Eine Möglichkeit ist auch, dass der Benutzer den Scanvorgang startet und nur eine bestimmte Datei oder ein Verzeichnis durchsuchen lässt, etwa wenn ein Programm aus dem Internet geladen oder aus einer anderen nicht vertrauenswürdigen Quelle bezogen wurde und der Benutzer sichergehen möchte, dass dieses frei von Schadsoftware ist.

2.3 Online Scanner

Der Online Scanner wird aus dem Internetbrowser heraus gestartet, muss also nicht auf dem System installiert werden. Oft ist dazu allerdings der Internet Explorer notwendig, da die Online Scanner auf ActiveX aufbauen. Dieses ActiveX Steuerelement erlangt nach der Bestätigung durch den Benutzer Zugriff auf die komplette Festplatte des Systems. Deshalb ist es anzuraten, nur vertrauenswürdige Webseiten der Antivirus Hersteller selbst dafür zu benutzen. Von einigen Herstellern wird allerdings auch angeboten, Programme mit Verdacht auf Befall einzeln hochzuladen und überprüfen zu lassen. Wird dann ein Virus gefunden, kann dieser sofort entfernt werden und der Benutzer kann das desinfizierte Programm wieder herunterladen.

Der Online Scanner kommt entweder zum Einsatz nachdem der PC bereits infiziert wurde und zu befürchten ist, dass das Antivirus Programm selbst bereits befallen ist oder ganz einfach um den Rat eines zusätzlichen

Virenscanners, meist von einem anderen Hersteller einzuholen. Inzwischen wird dieser oft kostenlose Service von vielen grossen Antivirus Herstellern, wie Kaspersky, Symantec, Sophos, McAfee und einigen weiteren, angeboten. [14]

3 Viren, Erkennung und Entfernung

3.1 Virentypen

Viren und Würmer sind seit ihrem ersten Auftreten immer komplexer und schwieriger in der Erkennung und Entfernung geworden. Waren Viren früher noch kleine Programme, die sich lediglich selbst verbreitet und eventuell auch Schaden angerichtet haben, so bringen sie heute oftmals Techniken mit, um sich selbst ünsichtbarßu machen. Als kleiner Einblick in die Historie der schadhaften Software werden deshalb im Folgenden kurz ein paar ausgewählte Formen von Viren vorgestellt. Es sei erwähnt, dass dies nur ein Auszug der möglichen Tricks ist und sich heutzutage teilweise sehr komplexe Viren im Umlauf befinden, die eine Kombination mehrerer der im Folgenden vorgestellten Techniken nutzen, um sich vor Antivirus Programmen zu verbergen. Viele Viren können sich darüber hinaus auch auf vielfältige Weise und über verschiedenen Wegen fortpflanzen. [7]

3.1.1 Verschlüsselte Viren

Diese Art von Viren verschlüsselt sich selbst. Dabei hat das Virus vor seinem eigentlichen Code einen Entschlüssler, der den Code in den Hauptspeicher dechiffriert und dann ausführt. Ein Antivirus Programm kann Viren solcher Art meist am Decryptor, der in unverschlüsselter Form vorliegt, erkennen.

Oligomorphe Viren Oligomorphe Viren sind eine Weiterentwicklung einfach verschlüsselter Viren. Sie ändern jedes mal ihren Schlüssel, indem sie ihn aus einer vorgegebenen Liste aussuchen. Anstatt sich bei einer Vervielfältigung einfach zu klonen, verschlüsseln sie ihren Code, der zur Ausführungszeit im Hauptspeicher liegt, erneut mit einem anderen Schlüssel, um dann mit dieser angepassten Version Programme zu infizieren.

Polymorphe Viren Polymorphe Viren erzeugen eine endlose Variation von verschiedenen Ver- und Entschlüsselungsmechanismen, mit sich ändernden Schlüsseln oder sogar mehrstufiger Verschlüsselung. Sie ändern mit jeder Generation ihren (unverschlüsselten) Entschlüsselungsmechanismus. Der

Algorithmus, der die neuen Decryptoren erzeugt, ist seinerseits ebenfalls im entschlüsselten Teil des Virus untergebracht. Damit wird die Erkennung für das Antivirus Programm anhand einer Signatur erschwert, die den Decryptor des Virus und somit den Virus selbst entdecken könnte. [10]

3.1.2 Metamorphe Viren

Metamorphe Viren verändern ihren Code bei jeder Reproduktion selbst und damit ihre Signatur, ohne dabei ihre Funktionalität zu verändern. Dies kann bereits durch einfaches Einfügen von NOPs (NOP: No OPeration, Befehl zum "Verschwenden" eines CPU Taktes) geschehen oder durch Einfügen von Code, der keinerlei Nutzen bringt und die Programmlogik nicht verändert (so genannte Junk Code Insertion). Durch diesen "Müll", der zwischen den eigentlichen Virusbefehlen liegt, wird erreicht, dass Signaturen nicht mehr auf den Virus passen und der Virenscanner diesen nicht mehr findet.

Maschinenregister umbelegen Ein einfaches Beispiel für solche selbstverändernden Viren ist die Methode der Registerumbelegung des "Win95/Regswap" Virus. Es ersetzt in seinem Code bei jeder Vervielfältigung Registernamen durch andere, was die Funktionsweise des Virus nicht beeinträchtigt, aber seine Signatur ändert.

Sprünge setzen Ein weiteres Beispiel für diesen Virus Typ ist der "Zperm.A" Virus, der Code durch äquivalenten anderen, also "mov eax, 0" durch "sub eax, eax" oder "xor eax, eax" ersetzen kann, die jeweils genau das selbe Ergebnis produzieren, nämlich die Zahl 0 in das Register EAX zu schreiben. Die eigentliche Metamorphie wird jedoch durch Setzen von Sprüngen erzielt.
Wie in Abbildung 1 zu sehen, wird praktisch nach jedem Befehl ein Sprung ausgeführt, wobei die sequentielle Abfolge in zufälliger Reihenfolge auseinandergezogen wird, um die Signatur unkenntlich zu machen. Außerdem wird der zweite Teil des Codes in den Speicher geschrieben und zu diesem gesprungen, eine weitere Maßnahme, um unerkannt zu bleiben. Der entscheidende Faktor dabei ist jedoch, dass bei jeder Reproduktion des Virus eine andere Variante entsteht und somit eine andere Signatur.

Komplexere Umformungen Der "Win32/Evol" Virus geht noch weiter und erzeugt ganz neuen Code von sich selbst, in dem er Befehle in mehrere einzelne aufspaltet. Anstatt zum Beispiel den Wert "5151EC8Bh" in die Speicherzelle mit der Adresse aus dem Register ESI zu schreiben, schreibt

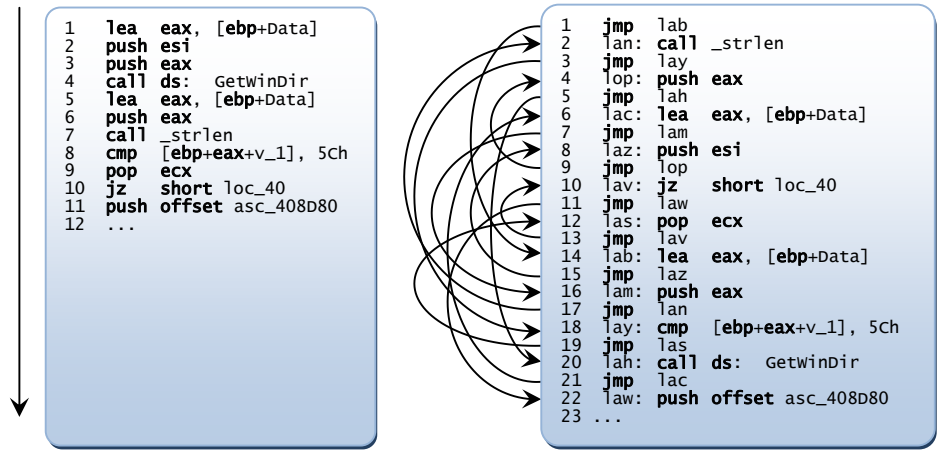

Abbildung 1: Sequentieller und veränderter Programmfluss

er den Wert in seiner zweiten Generation erst in ein Zwischenregister um es dann in den Speicher zu schreiben. In der dritten Generation teilt er dann sogar den konstanten Wert in einen zufällig gewählten und einen Rest auf, um sie dann zur Laufzeit wieder zusammen zu addieren und das Ergebnis wiederum in den Speicher zu schreiben, wie in Abbildung 2 zu sehen. Somit kann der Virenscanner nicht einmal mehr den Konstanten Wert "5151EC8Bh" in der Signatur finden. Mit dieser Metamorphie Engine verändert er sogar die Engine selbst mit jeder Generation. Um sich jedesmal stark von seiner Vorgängergeneration zu unterscheiden, lässt er viele kleine Zufallswerte entscheiden, wie genau sein Code jedesmal geändert wird. Damit erreicht er nahezu unendlich viele verschiedene Signaturen von sich aufzuweisen. [2, 8]

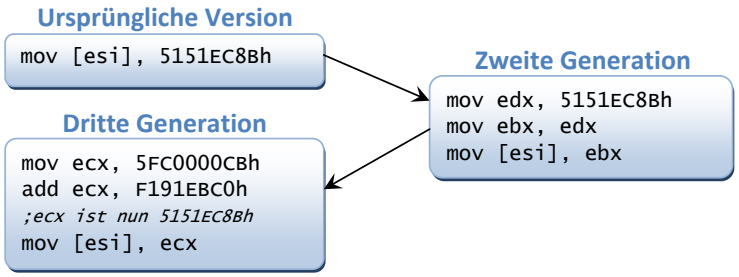

Abbildung 2: "Win32/Evol" schematisch in 3 Generationen

3.1.3 Einstiegspunkt verbergende Viren

Diese Viren Variante setzt sich nicht an den Beginn oder das Ende einer Datei, wird also nicht beim Start des Programms ausgeführt, sondern setzt sich irgendwo in die Mitte in eine Unterroutine. Falls diese Unterroutine des Programms nur selten oder gar nicht aufgerufen wird, kommt es auch zu keiner Ausführung des Virus. Der Virus verbreitete sich also relativ langsam, aber stetig, da er sich zu Zeiten, als Virenscanner diese Art Viren nicht fanden, ungehindert ausbreiten konnte.

3.2 Erkennung

Wie Eingangs erwähnt, darf man diese Klassifizierung der Viren nicht als scharf gezogene Grenze sehen. Viele Viren benutzen eine Vielzahl an Techniken gleichzeitig, und auch Abwandlungen davon, die dazu dienen, sich vor Antivirus Software zu verstecken. Deshalb werden im Folgenden einige Techniken vorstellt, die Virenscanner benutzen, um ein infiziertes Programm aufzuspüren. Beispiele erläutern jeweils, auf welche Arten von Viren diese abzielen. Nicht jede auf dem Markt befindliche Antivirus Software hat diese Techniken gleichermassen implementiert, und die im folgenden vorgestellten Techniken sind auch nicht zwingend unabhängig voneinander. Die Kombination der verschiedenen Methoden erzeugt den Schutz, den heutige Virenscanner bieten.

3.2.1 Signaturscanning

Die grundlegendste und auch Heute noch am meisten eingesetzte Technik ist das Suchen nach einer Signatur in einem ausführbaren Programm. Dabei werden neu entdeckte Viren von Spezialisten bei Antivirus Herstellern analysiert und daraufhin eine möglichst passende Signatur des Virus erzeugt. Diese ist im Optimalfall in allen Viren dieses Typs enthalten und passt zudem möglichst nicht auf harmlose Programme. Je nach Implementierung der Hersteller werden meist Reguläre Ausdrücke oder andere effiziente String Matching Algorithmen, z.B. der Boyer Moore Algorithmus [6], verwendet. Mit Hilfe dieser Algorithmen wird dann bei einem Scanvorgang das Programm nach Übereinstimmungen mit allen gespeicherten Signaturen gesucht. Dabei können auch Wildcards, also Platzhalter die für eine beliebige Anzahl an Bytes im Code stehen, behilflich sein. Reguläre Ausdrücke mit Wildcards stellen also in diesem Fall eine Art Schablone dar, die mit dem Maschinencode abgeglichen wird. Damit können auch Viren mit kleineren Modifikationen, etwa Junk Code Insertion (siehe 3.1.2), gefunden werden.

Auch einfache verschlüsselte Viren können mit dieser Methode noch gefunden werden, denn sie brauchen, um ihren Körper zu entschlüsseln, einen Decryptor, der in ausführbarem Maschinencode vorliegen muss. Anhand dieses Decryptors kann der Virenscanner erkennen, um welchen Virus es sich handelt.
Wichtig ist bei dieser Technik, dass die Signatur Datenbanken der Virenscanner über das Internet immer möglichst aktuell gehalten werden müssen, denn ein Signaturscanner kann neue Viren nur dann identifizieren, wenn er mit den neuesten Signaturen versorgt ist. [12, 9]

3.2.2 Transformation

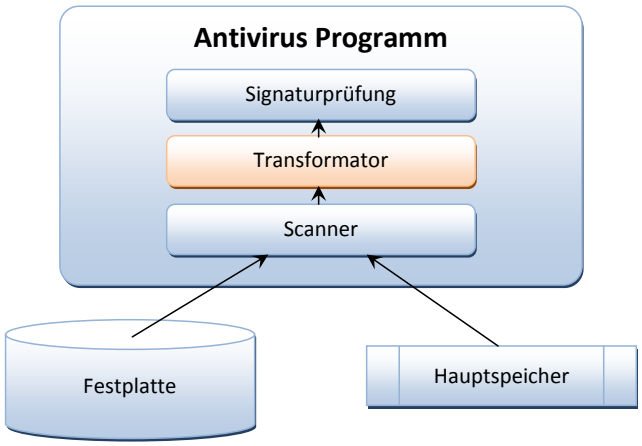

Abbildung 3: Transformator vor der eigentlichen Signaturprüfung

Eine Möglichkeit metamorphe Viren (siehe 3.1.2) aufzuspüren, sind Software-Transformationen, die man vor den eigentlichen Signaturscanner schaltet, vgl. Abbildung 3. Dabei wird der durch Metamorphie unkenntlich gewordene Maschinencode durch einen mehrstufigen Algorithmus auf eine möglichst einfache und eindeutige Form zurück transformiert. Bei dieser Transformation werden zuerst, wenn möglich, Maschinenbefehle in einfachere überführt und zusammengefasst. Junk Code, also für die Ausführung irrelevanter Code, wird entfernt. Danach werden nichtbedingte Sprünge eliminiert und der Code wieder in die richtige Reihenfolge gebracht und zuletzt sequentiell Register und Labels benannt. Dies führt in den meisten Fällen dazu, dass ein Virus, der sich selbst beliebig oft metamorph verwandelt hat, immer wieder auf ein und dieselbe Signatur zurückgeführt werden kann. Die-

ser fertig transformierte Code kann dann in einem letzten Schritt in einem Signaturscanner mit einer Signatur gematcht werden. Für ausführlichere Informationen zu dieser Vorgehensweise sei auf [4] verwiesen.

3.2.3 Prävention

Vorbeugung von Virenbefall dient zwar nicht in direktem Sinne der Erkennung, kann jedoch hilfreich sein, Viren im Nachhinein, also beim Scanvorgang, besser und auch schneller zu finden.

Prüfsummen Oft wird zu diesem Zweck eine Tabelle erstellt, in der von jedem ausführbaren Programm des Rechners Prüfsummen bzw. Hashwerte, etwa MD5, SHA-1 oder ähnliche, gespeichert werden. Weicht bei einem späteren Scanvorgang dieser gespeicherte Wert vom dem neu gebildeten ab, kann dies ein Indiz für einen Virenbefall des Programms sein und der Virenscanner kann mit den vorgestellten Techniken beginnen, in dem Programm nach dem Virus zu suchen.

Impfung Es wurde auch bereits eine auf den ersten Blick interessante Technik vorgeschlagen, die die zu schützenden Programme "impft". Dabei wird ausgenutzt, dass einige Viren in jedem Programm, das sie befallen, einen Wert einfügen, um später leichter verifizieren zu können, ob dieses Programm bereits befallen wurde. Wenn dieser Wert im Zuge einer Impfung schon vor einem Virenangriff gesetzt wird, wird der Virus das Programm als bereits infiziert ansehen und das Programm nicht befallen.
Der entscheidende Nachteil war einerseits, dass viele Viren andere Methoden benutzen, um zu markieren, ob sie den Rechner bereits befallen haben. So legen manche Viren etwa einen Schlüssel in der Windows Registry an. Zum anderen müsste man jedes Programm gegen sämtliche neu auftretende Virenarten neu impfen. Das Problem dabei ist, dass diese verschiedenen Impfungen sich zum Teil gegenseitig ausschliessen. Zum Beispiel erwartet Virus X den Hexwert "FF" als erstes Byte der Datei, Virus Y dagegen sucht nach dem Wert "C3" an erster Byteposition, um die Datei nicht erneut zu infizieren. Aus diesen Gründen wurde diese Methode der Prävention nie in komerziellen Antivirus Programmen implementiert. [13]

3.2.4 Heuristische Erkennung

Unter Heuristik (altgriechisch heuriskein, "(auf-)finden", "entdecken") versteht man in der Informatik Algorithmen, die, um Rechenzeit einzusparen,

nicht die optimale, sondern nur eine hinreichend genaue Lösung zu einem Problem finden. Unter diesem Gesichtspunkt fällt also auch das einfache Signaturscanning unter die Kategorie Heuristik [11, vgl. s 1-2]. Im Bezug auf Virenscanner jedoch, wird dem Begriff Heuristik meist eine spezielle Methode zugeordnet, die Programme auf Struktur und Verhalten hin untersucht, statt auf eine speziell auf jeden Virus zugeschnittene Signatur hin zu überprüfen. Dafür werden in einem ersten Schritt Verhaltensmuster in einem Programm aufgedeckt. In einem weiteren Schritt wird dann versucht, Anhand der gefundenen Eigenschaften zu bewerten, ob das Programm von einem Virus infiziert wurde. Prinzipiell existieren zwei verschiedene Vorgehensweisen für den ersten Schritt der Analyse: das statische und das dynamische heuristische Scanverfahren.

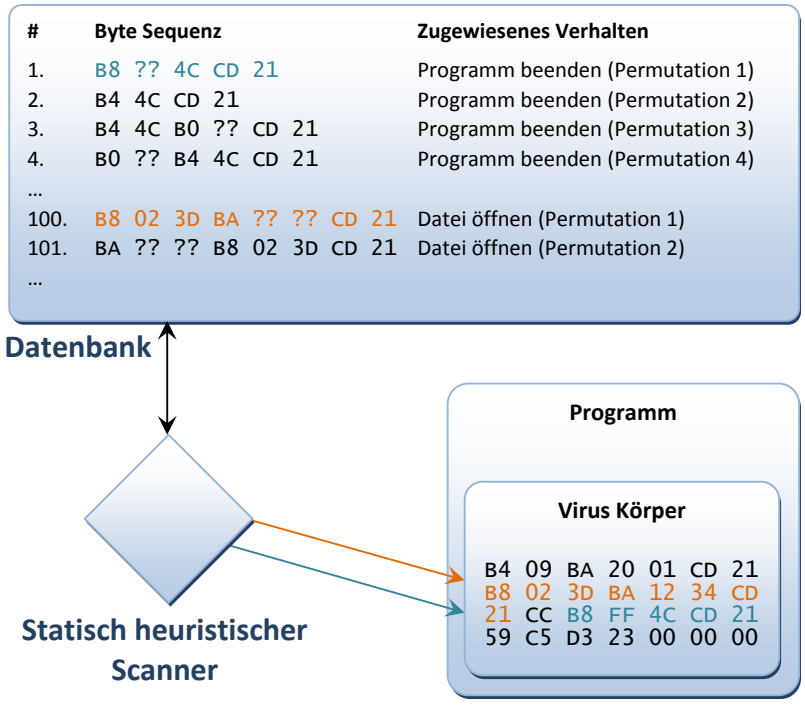

Abbildung 4: Der Scanner sucht nach Verhaltensmustern, die in seiner Datenbank enthalten sind.

Statischer heuristischer Scanner Bei diesem Verfahren wird nach Bytefolgen gesucht, denen ein bestimmtes Verhalten zugewiesen ist. Dazu hat

der statische heuristische Scanner eine Datenbank zur Verfügung, in der festgelegt ist, welchem Verhalten ein Bytemuster entspricht. Dabei können auch mehrere, auf den ersten Blick verschieden aussehende Bytefolgen dem selben Verhalten entsprechen, da es in Maschinencode viele Wege gibt, ein und den selben Befehl zu formulieren, wie wir bereits vorher gesehen haben (siehe 3.1.2).
Die Algorithmik ist hierbei ganz ähnlich wie beim Signaturscanning, jedoch wird ein anderes Ziel verfolgt. Im Gegensatz zum Signaturscanning, das für jeden Virus eine eigene, auf den Virus maßgeschneiderte Signatur benötigt, wird bei diesem Scanvorgang eine Vielzahl von typischerweise von Viren ausgeführten Aktionen gesucht und kategorisiert. In Abbildung 4 wird illustriert, wie ein solcher Scanner herausfindet, dass das Programm eine Datei öffnet und danach terminiert.

Dynamischer heuristischer Scanner Dynamische heuristische Scanner erzeugen einen virtuellen PC, in dem das Programm gestartet wird. Dieser virtuelle PC ist komplett vom Rest des Systems abgeschottet und entzieht dem Virus die Möglichkeit, Schaden anzurichten. Führt das Programm Betriebssystem Aufrufe aus, kann der Scanner dies mitprotokollieren. Der Scanner sieht damit, was das Programm bezwecken will, noch bevor es dies auf einem tatsächlichen System umsetzen kann.

Unterschiede zwischen den Scannern Der Erfolg von statischen Scannern ist sehr davon abhängig, wie genau der Virus seine Programmlogik implementiert hat. Ist ein Befehl nicht in dessen Datenbank, wird er ihn "übersehen". Dynamische Scanner hingegen suchen nicht nach dem Weg, auf dem ein Virus sein Ziel erreichen will, sondern beobachtet ihn dabei, wenn er sein Ziel erreicht. Sein beabsichtigtes Vorhaben wird entdeckt. Damit hat dieser Scanner auch keine Probleme mit metamorphen und polymorphen Viren, da diese sich im virtuellen PC selbst entpacken und ihre Verbreitungs- und Schadensroutinen ausführen. Diese benötigen wiederum Aufrufe, die vom Scanner protokolliert werden.
Ein entscheidender Nachteil des dynamischen Scanners ist einerseits die schlechte Performanz, denn die Emulation eines PCs ist ein sehr langsamer Prozess. Die virtuelle Maschine muss zuerst gestartet werden, und dem Virus dann die Möglichkeit gegeben werden, sich eventuell zu entpacken oder zu entschlüsseln und sich danach auszuführen. Dies kostet alles Zeit, in der ein statischer Scanner bereits eine Vielzahl von Programmen scannen kann. Ausserdem kann ein dynamischer Scanner relativ leicht umgangen werden.

Der Pseudocode “1. Wenn die Systemzeit gleich 12:00 Uhr ist springe zu 3; 2. Springe zu 1; 3. Führe Viruscode aus“ zeigt anschaulich, dass dieser Virus nur ausgeführt wird, wenn die Systemzeit des virtuellen Rechners gerade 12 Uhr ist. Da die Wahrscheinlichkeit gross ist, dass dies nicht zutrifft, wird ein dynamischer Scanner diesen Virus nicht erkennen können, da er erst gar nicht zur Ausführung seines eigentlichen Codes kommt.

Analyse der gefundenen Strukturen Wie in Abbildung 4 zu sehen war, hat das Beispielprogramm eine Datei geöffnet und sich danach beendet. Dies kann der Scanner entweder durch seinen statischen oder den dynamischen heuristischen Scanner entdeckt haben. Das Entscheidende ist jedoch, dass diese Erkenntnis weder auf einen Virus schliessen lässt, noch einen solchen ausschliesst. Deshalb folgt nun die Bewertung. Dieser Schritt geschieht in den meisten erhältlichen Virenscannern mit Hilfe eines Expertensystems. Ein Experten System ist allgemein eine Klasse von Software Systemen, die auf der Basis von Expertenwissen, in Form von definierten Regeln, zur Lösung oder Bewertung bestimmter Problemstellungen dient. In diesem Fall besitzt dieses Expertensystem ein Regelwerk, das jedes gefundene Verhalten mit Gewichtungen versieht. Das heißt, einzelne Verhaltensmuster können das Programm als Virus verdächtigen, andere können das Programm “entlasten“. Sobald die Summe der Gewichtungen eine gewisse Grenze überschreitet, kann davon ausgegangen werden, dass es sich um einen Virus handelt. Erweitert wird solch ein Regelwerk oft noch durch Algorithmen, die etwa einer bestimmten Kombinationen von Einzeloperationen ein Verhalten zuordnen können. So können zum Beispiel die Mehrheit aller Entschlüsselungsroutinen als solche entlarvt werden, sie bestehen nämlich semantisch immer aus der Folge “1. Lese Byte; 2. Entschlüssle Byte; 3. Speichere Byte; 4. Springe zu 1.“. Selbst komplexe Polymorphe Viren (siehe 3.1.1) weisen immer dieses Verhalten auf und können so sicher gefunden werden.
Die genaue Implementierung ist bei einem realen Antivirus System sicher weitaus komplexer, und bei jedem Antivirus Software Hersteller anders gestaltet. Wichtig ist jedoch, dass mit dieser Methode versucht wird, Viren wegen ihrem andersartigen Verhalten von normalen Programmen zu unterscheiden, also zum Beispiel wenn ein Programm plötzlich versucht auf sämtliche ausführbare Dateien mit einem Schreibzugriff zuzugreifen. [5]

Vor- und Nachteile des Verfahrens Heuristische Verfahren sind hilfreich im Kampf gegen polymorphe und metamorphe Viren, also dort wo ein

Signaturscanner nur wenige Chancen auf eine Erkennung hat. Der Vorteil ist, dass Viren schon entdeckt werden können, bevor eine Signatur für alle verbreiteten Antivirus Programme vorliegt. Die Verbreitung kann also schon im Keim erstickt werden. Eine kleine Einschränkung ist die eventuell leicht höhere falsche Erkennungsrate (false positive: ein Programm wird als Virus klassifiziert, ist jedoch nicht infiziert). Diese wird allerdings durch ständige Weiterentwicklung der Technologie sehr niedrig gehalten. Ein grosses Problem ist jedoch, dass auch Virenschreiber Zugriff auf diese Scanner haben und ihren Virus so lange optimieren können, bis dieser von keinem oder nur noch von wenigen Virenscannern erkannt wird (Anti-Heuristic Virus).

3.2.5 System Überwachung

Es kann hilfreich sein, außer dem Dateisystem auch andere Bereiche des Systems, wie die Windows Registry, zu überwachen. Wenn dort ein bestimmter Schlüssel auftaucht, der eindeutig einem Virus zugewiesen werden kann, kann Alarm ausgelöst, das System weitestgehend angehalten, und mit der Suche nach dem Virus begonnen werden [3]. Weitere Bereiche können auch die globalen Systemvariablen sein, in der manche Viren zum Beispiel einen Wert schreiben, um sich selbst zu signalisieren, dass sie das System bereits infiziert haben.

3.3 Entfernung

Vor 1990 wurde ein Programm mit einem erkannten Virus meist nur als infiziert markiert und so belassen. Die einzige Möglichkeit war dann die ausführbare Datei wieder mit einer originalen, nicht infizierten Version zu überschreiben. Bei damaligen Systemen war dies meist auch kein grösseres Problem, da oft nur eine überschaubare Anzahl von Programmen auf dem Rechner installiert waren.
Inzwischen ist es jedoch essentiell für ein Antivirus Programm, eventuelle Viren nicht nur zu erkennen, sondern auch zu Entfernen. Besonders in Anbetracht der Datenmengen die bereits auf Heim PCs zu finden sind. Ohne Entfernungsmöglichkeiten wäre der einzige Weg dann nur noch ein komplettes Neuaufsetzen des Systems.
Deshalb bringen Antivirus Programme mit einer Aktualisierung der Signaturdatenbank auch gleichzeitig Regeln mit, die das Entfernen der jeweiligen Viren beschreiben. Dabei wird dem Virenscanner mitgeteilt was genau er aus dem infizierten Programm entfernen muss, wenn er einen Virus eindeutig identifizieren konnte. Mit diesen Regeln wird sichergestellt, dass das desin-

fizierte Programm keinen Schaden davonträgt und funktionstüchtig bleibt. Des weiteren ist es manchmal auch nötig, Schlüssel aus der Registry zu entfernen oder Dateien vom System zu löschen. So legen manche Viren eine Kopie von sich im Windows Verzeichnis ab. Um solch eine Entfernung durchzuführen, muss der Virenscanner allerdings die nötigen Informationen besitzen. Diese müssen zuvor von Antivirus Experten erarbeitet werden. Ist solch eine Information (noch) nicht verfügbar, wird das infizierte Programm in Quarantäne verschoben. Damit kann es vorerst nicht mehr ausgeführt werden, um Schaden am System zu vermeiden. In seltenen Fällen ist eine Entfernung auch gar nicht mehr möglich, da manche Viren ihre Wirtprogramme entweder beschädigen, oder ganz überschreiben. [1]

4 Ausblick

Mit Weiterentwicklung und Verbreitung heuristischer und semantischer Erkennungsmethoden bei Virenscannern wird es die herkömmliche Form eines Virus in Zukunft schwerer haben, unentdeckt zu bleiben.

Neue anti-heuristik Viren Deswegen ist meine persönliche Prognose, dass dadurch neue Formen entstehen, die sich in ihrem Verhalten von heutigen Viren unterscheiden werden. Denkbar wäre dann zum Beispiel Schadsoftware die sich, um unentdeckt zu bleiben, nicht mehr in alle ausführbaren Dateien einnistet, sondern sich gezielt in Programme schreibt, die auf vielen Rechnern installiert sind (Beispielsweise das Messaging Programm ICQ), um dann deren Internetfunktionalität nutzt, um Daten zum Ersteller des Virus zu senden. So wird es dann für ein Antivirus Programm immer schwerer, zu entscheiden, ob es sich bei einer Operation um eine rechtmässige Aktivität des Programms oder die eines darin platzierten Schadprogramms handelt. Damit könnten auch die Erkennungsmethoden vorsichtiger werden, um nicht zu viele Fehlalarme (false positives) auszulösen.

Hochsprachen Viren Eine weitere Herausforderung für Antivirus Hersteller werden Hochsprachen Viren sein. Etwa in Java oder in .net Sprachen geschriebene Viren, die nicht in Maschinencode Form vorliegen, sondern in speziellem Binärcode, der dann erst zur Laufzeit durch die Java Virtual Machine bzw. den .net eigenen Just-in-time Compiler übersetzt werden. Denkbar wäre sogar ein sozusagen mehrfach hybrider Hochsprachen Virus, der zum Teil in Java und teilweise in .net Sprache geschrieben wurde. Zusätzlich könnte er beispielsweise seinen eigenen Binärcode in ein textbasiertes Script packen um sich zusätzlich auf diesem dritten Weg zu verbreiten. Eine solche Art von Viren könnte hochgradig selbstverändernd gestaltet werden, sich eventuell sogar selbst neu kompilieren und damit ganz neue Versionen von sich erzeugen. Wenn man bedenkt, dass sich dieser Virus zusätzlich mit Hilfe von Rootkits verstecken, wird klar was rein technisch machbar ist und wie schwierig sich die Aufspürung solcher Viren gestalten könnte.

Bestmöglicher Schutz und Vorsorge Die Bedrohung durch Viren und andere Schadsoftware wird sich meiner Meinung nach wegen den eben genannten technischen Möglichkeiten, auch in mittelfristiger Zukunft nicht einfach abwehren lassen. Es ist vielmehr ein ständiger Prozess, bei dem sowohl Virenprogrammierer als auch Antivirus Hersteller fortwährend neue

Technologien entwickeln um Virenscanner zu umgehen, bzw. neue Viren aufzuspüren. Gerade deshalb ist es so wichtig, sich aufgrund eines installierten Virenscanners nicht unbekümmert in Sicherheit zu wägen, sondern selbst zu beurteilen, ob einer Quelle vertraut werden kann aus der man seine Software bezieht. Die grösste Sicherheitslücke eines PCs ist und bleibt der Benutzer selbst.
Es ist zu betonen, dass ein Virus seine Schadfunktion nur in den allerwenigsten Fällen voll entfalten kann, wenn der Benutzer achtsam, und eine Antivirus Software aktiv war. Falls es dann doch zu einer Infektion kommt, denn 100%igen Schutz gibt es nicht, ist es wichtig, Daten schon vor dem Befall auf externe und vom PC und Netz abgekoppelte Medien gesichert zu haben. Eine Desinfektion sollte dann nur von ausserhalb, also nicht aus dem laufenden befallenen System heraus, durchgeführt werden. Etwa mit einer bootfähigen Live CD, die einen Online Scan der Festplatte durchführt.

Neuronale Netze Als letztes, und als Ausblick, ist eine Technologie zu nennen, die langfristig als Gegenmassnahme denkbar wäre. Es ist fraglich wann und ob diese Technologie zum Einsatz kommt. Denn es wird schon seit Jahrzehnten an lernfähigen Programmen geforscht, bisher mit eher mässigem Erfolg. Sogenannte Neuronale Netze, die dem menschlichen Gehirn nachempfunden sind, und selbst "lernen" können, könnten als Bewertungseinheit für Schadsoftware dienen. Der Virusscanner würde in solch einem fiktiven Computer die laufenden Programme analysieren und erkennen, welche Funktionen sie bieten. Würde nach einer Virusinfektion nun ein Programm eigenständig zusätzlich andere Aktionen ausführen, also solche, die nicht vom Benutzer ausgelöst wurden und auch nicht zum Funktionsumfang des Programms gehören, würde der Scanner dies sofort bemerken und Alarm auslösen. Fraglich ist jedoch, ob diese Technologie jemals die gewünschte Erfolgsquote erreichen kann, also nicht auch Programme als infiziert klassifiziert, die es gar nicht sind oder sogar Viren "übersieht". Denn die Entscheidungsgrenze, ob Virus oder nicht, lässt sich nicht scharf ziehen, es ist eher ein fliessender Übergang. So kann ein Virus aus Funktionen zusammengebaut werden, die einzeln auch in anderen Programmen zu finden sind, zusammen jedoch Schaden anrichten. Ob so etwas dann trotzdem funktionieren kann, darüber lassen sich im Moment nur Vermutungen anstellen, denn eine solche Technologie existiert bisher nur in einem Ansatz, der es noch nicht annähernd erlaubt in einer Antivirus Software Einsatz zu finden. Zusätzlich ist dazu eine viel grössere Rechenkraft nötig als Rechner sie heute bieten. Der Computer müsste demnach selbst über sich reflektieren, und intelligent

entscheiden, ob eine Aktion, die er ausführen soll, vom Benutzer gewollt ist oder nicht, anstatt einfach loszurechnen.

Literatur

[1] Jason Bruce. *The challenge of detecting and removing installed threats.* Sophos, 2006.

[2] Peter Ferrie and Peter Ször. *Hunting For Metamorphic White Paper.* Symantec Press, 2006.

[3] Shlomo Hershkop, Eleazar Eskin, Sal Stolfo, Frank Apap, and Andrew Honig. *Detecting Malicious Software by Monitoring Anomalous Windows Registry Accesses.* 2001.

[4] Johannes Kinder, Stefan Katzenbeisser, Helmut Veith, Mihai Christodorescu, and Somesh Jha. Software Transformations to Improve Malware Detection. *Journal in Computer Virology*, pages 253–265, 2007.

[5] Matthias Langhammer. *Viren, Würmer und Co. Statische Programmanalyse.* TU München, 2005.

[6] J. Strother Moore and Robert S. Boyer. *A Fast String Searching Algorithm.* Communications of the ACM, 1977.

[7] Holger Pawlita. *Virentechniken: Analyse und Metamorphismus.* TU Müenchen, 2005.

[8] Peter Ször. *Advanced Code Evolution Techniques and Computer Virus Generator Kits.* Symantec Press, 2005.

[9] Valentin Pletzer. *Computervirenbekämpfung.* TU München, 2003.

[10] Symantec. *Understanding and Managing Polymorphic Viruses White Paper.* Symantec Press, 1996.

[11] Symantec. *Understanding Heuristics: Symantec's Bloodhound Technology White Paper.* Symantec Press, 1997.

[12] `http://www.hackingspirits.com/eth-hac/papers/whitepapers.asp` and Debasis Mohanty. *Anti-Virus Evasion Techniques and Countermeasures.* 2004.

[13] Wolfgang Zejda. *Antivirenprogramme.* TU Müenchen, 2005.

[14] Martin Zohlhuber. *Destruktive Programme und Gegenmaßnahmen am Bespiel Viren und Virenscanner.* TU Wien, 2002.