

# Böswillige Software, mobiler Code (Java, JavaScript, ActiveX), WEB-Sicherheit

Heiko Leseberg

19. März 2003

## Zusammenfassung

Der vorliegende Text soll zunächst einen Einblick in die verschiedenen Arten der böswilligen Software geben und dann die Technologien aufzeigen, die man benutzt hat, um sie zu implementieren. Zum Schluß soll ein Überblick gegeben werden, wie man sich versucht, dagegen zu schützen.

## 1 Einleitung

Das Internet, welches sich von einem nur textbasierte Medium hin zu einem immer stärker aktiven und interaktiven gewandelt hat, ist heutzutage nicht nur für Wissenschaftler und Experten zugänglich, sondern für jedermann. Die Entwicklung der letzten Jahre hat es sogar notwendig gemacht, dass sich jeder damit beschäftigt. Einige Vorfälle der letzte Jahre, die erheblich wirtschaftliche Schäden nach sich gezogen haben, haben gezeigt, dass es wichtiger wird, das anfangs euphorisch genutzte Internet auch unter dem Licht des Sicherheitsaspektes zu betrachten.

## 2 Böswillige Software

Böswillige Software wird auch *Malware* und *maliziöse Software (malicious software)* genannt und bezeichnet im wesentlichen jene Software, die aus Sicht des Benutzers Schaden anrichtet. Die Frage, was bössartige Software ist, läßt sich vielleicht nicht immer leicht klären, es hängt u. a. damit zusammen, welche Erwartungen der Benutzer an die Software hat. Bei Programmen, die völlig unerwartet und eigenständig etwas wichtiges löschen, ist wohl die schlechte Absicht eindeutig. Bei Programmen, die man jedoch selbst startet und bei denen einem vielleicht nicht ganz klar ist, was sie tun, kann dies jedoch zum Grenzfall werden.

Man kann zunächst die böswillige Software in zwei Klassen unterteilen, jene, die als eine wesentliche Eigenschaft hat, dass sie sich in vernetzten Systemen verbreitet, durch ihre Programmierung und jene, die keine Netzwerke braucht, um sich zu verbreiten, sie nutzt dazu den Datenaustausch z.B. über Disketten. Bei letzteren ist jedoch auch eine Nutzung vernetzter Systeme möglich, um Schaden anzurichten. Beide Klassen werden wiederum in sich selbständig und sich nicht selbständig vervielfältigende (replizierende) Software unterteilt [VTC 2000].

Der Schaden, der entstehen kann, reicht erfahrungsgemäß von einfachen Bildschirmmeldungen, ohne wichtige Bedeutung bis zum Löschen aller Daten auf Festplatten und anderen Datenträgern oder Attacken auf das BIOS. Es gibt Viren, die das BIOS überschreiben oder die BIOS-Passwörter aktivieren, falls jenes nicht mit einem dagegen wirksamen Schreibschutz versehen ist. Im Bereich der vernetzten Systeme ist es sogar möglich, PCs durch bestimmte Malware, von der sie befallen sind, fernzusteuern. Man muss damit rechnen, dass irgendwann alles ausgenutzt wird, was möglich ist, um Schaden anzurichten.

## 2.1 Malware auf Einzelrechnern

### Malware mit eigenständiger Replikation

Der *Virus* ist die klassische Art der böswilligen Software, er kopiert sich in andere Software auf einem Rechner hinein, die dann das Wirtsprogramm genannt wird, und wird entweder durch das Starten dieses Wirtsprogrammes oder durch einen anderen Prozess wie z.B. die Zeit oder das Datum gestartet. Der Begriff Virus geht also analog zu dem Begriff aus der Biologie, dem er entliehen ist. Er enthält ein Programmabschnitt, der dafür verantwortlich ist, dass er weiterkopiert wird (Reproduktionsteil) und einen Teil (Payload), der den schädigenden Code enthält. Der eigentliche Schaden kann vielfältig sein (siehe oben). Es können aber schon durch Belegung von Speicher und dem Ausführen unerwünschter Funktionen, die keine schädigende Veränderung am System vornehmen, Ressourcen verbraucht werden.

Man kann Viren unterscheiden in:

- **Systemviren/Bootviren:** Diese Viren kopieren sich in den Bootsektor von Festplatten, Disketten und anderen Datenträgern und werden dann aktiv, wenn der Rechner hochfährt oder eine Diskette oder ein anderer Datenträger gelesen wird. Der Programmierer des Virus nutzt hier die Tatsache, dass der Bootsektor, zumindest der Festplatte eines Rechners ziemlich häufig, nämlich immer beim Hochfahren des Computers gelesen bzw. ausgeführt wird.
- **Dateiviren:** Sie schreiben sich zunächst in ausführbare Dateien, wobei sie entweder die Wirtsprogramme so verändern, dass diese noch ausführbar bleiben, diese werden *Anhängende Viren* genannt, oder sie überschreiben den eigentlichen Programmcode des Wirtes und machen so die Ausführung des Wirtes unmöglich

(*Überschreibende Viren*). Sogenannte *Cavity-Viren* sind sogar in der Lage, eventuelle Freiräume im Wirtsprogramm zu nutzen und sich dort hinein zu kopieren, auf diese Weise verändert sich die Größe der Wirtsdatei nicht, was bei den anderen beiden Dateivirentypen der Fall ist. An einer Größenveränderung einer Datei läßt sich also eventuell auf Viren schließen.

- **Companion-Viren:** Dieser Virustyp verändert seine Opferdatei nicht, er erzeugt eine neue Datei mit gleichem Namen, die beim Aufrufen der „infizierten“ Datei zuerst gestartet wird, erst danach wird der „Wirt“ gestartet. Unter DOS kann man dazu eine bestimmte Schwäche in der Ausführungsreihenfolge von gleichnamigen Programme unterschiedlichen Typs ausnutzen.
- **Dateisystemviren:** Der Virus manipuliert das Dateisystem derart, dass ein Verzeichniseintrag auf ihn, den Virus, und nicht auf die ursprünglich hinter dem Eintrag stehende Datei verweist.
- **Multi-Partite-Viren:** Hier handelt es sich um eine Mischform aus Bootsektor und Dateiviren.
- **Makroviren:** Dies sind Viren, die in einer Makroprogrammiersprache geschrieben sind. Sie befinden sich in den Dateien, in denen Makros vorkommen, nämlich Textdokumenten von Schreibprogrammen wie Word oder anderen Dokumenten von Büroanwendungen. Sie können durch die Mächtigkeit der Makrosprache die Funktionen des Betriebssystems benutzen und haben so weitreichende Möglichkeiten. Die Makroviren sind in ihrer Ausführung nur von der Anwendung abhängig und nicht vom Betriebssystem. Was möglich macht, dass sie sich über verschiedene Betriebssysteme ausbreiten.
- **Skript-Viren** nicht zu verwechseln mit Skript-Würmern: Sie befallen Programmcode wie VisualBasicScript und JavaScript.
- **Stealth-Viren:** Dies sind Viren, die sich „tarnen“. Die Viren können sich vor dem Zugriff eines Virenschanners aus der befallenen Datei entfernen und nach dem Scannen wieder hineinschreiben, dies erfordert einiges an Aufwand und an Eingriffen in das Betriebs- bzw. Dateisystem.
- **polymorphe Viren:** Ein Virus, der sich verschlüsselt oder verändert durch Einfügen von nutzlosen Befehlen oder Verändern des Programmtextes in gleichbedeutenden anderen Programmtext.

### Malware ohne eigenständige Replikation

- **Trojanische Pferde/Trojaner:** Diese Programme führen einerseits eine vom Benutzer gewünschte Funktion aus, dazu aber noch eine, die vom Programmierer in anderer, schlechter, Absicht geschrieben wurde. Die zusätzliche vom Nutzer

nicht gewünschte Funktion kann eine schädliche für den Rechner sein oder aber mit dem Ausspionieren des Rechners zu tun haben, so könnten sensitive Daten des Benutzers ausgespäht und weitergeschickt werden. Die schlechten Funktionen werden meist im Verborgenen ausgeführt, damit der Benutzer das Programm weiter benutzt, ohne zu denken, es sei schädlich. Die Trojaner gewinnen durch das Internet an Bedeutung, da durch ActiveX und Java-Applets eine gute Möglichkeit geschaffen wurde, sie zu verbreiten. Auch per Email kann man Trojanische Pferde, einfach als Attachment, verschicken.

- **Logische Bomben:** Sie sind oft Teil eines größeren Programmes. Es sind Funktionen, die nach einer gewissen Dauer oder bei einem gewissen Ereignis auftreten. Das Ereignis z.B. eine gewisse Zeit stellen einen *Trigger* dar, der die schädliche Funktion aktiviert. Man kann diese logische Bombe als eine gewisse Art des Trojanisches Pferdes sehen.
- **Dropper:** Ein Dropper ist ein Programm, welches Viren aussetzt oder Trojaner installiert, es ist selbst nicht als Wirtsprogramm anzusehen. Er ist auch kein Virus, da er nicht über die Möglichkeit der Selbstreplikation verfügt.

## 2.2 Malware mit Befall von gesamten Netzen

### Malware mit eigenständiger Replikation

- **Würmer:** Dies sind Programme, die sich mit Hilfe von Rechnernetzen verbreiten und vervielfältigen. Im Unterschied zu Viren befallen sie ganze Netze und nicht Einzelrechner, die an kein Netzwerk angeschlossen sind. Sie sind meistens nicht an ein Wirtsprogramm gebunden, sondern residieren im Speicher, wo sie neue Netzwerkadressen ermitteln und sich nach dort versenden. Die Gefahr wird auch in der hohen Ausbreitungsgeschwindigkeit gesehen, erkennt man die Würmer nicht rechtzeitig, so haben sie schon den nächsten Rechner befallen.
- **MIRC-Würmer:** Sie verbreiten sich durch einen Chat Client mit Hilfe dessen Skriptsprache. Dateien mit Skriptinhalten können von Benutzer zu Benutzer verschickt werden.
- **Mailer/Mass Mailer:** Diese Wurmform wurde durch den LoveLetter Wurm bekannt, es handelt sich dabei um ein Email bzw. dessen Attachment. Der aktive Inhalt befindet sich beim Mailer als eingebettetes Skript oder in einem Attachment im Email. So kann er aktiviert werden beim Öffnen der Email oder beim Öffnen des Attachements. Der Mailer verschickt sich selbst und richtet auf dem jeweiligen System Schaden an. Die Technologien, die benutzt werden, sind u.a. VBScript und JavaScript.

**Beispiel:** LoveLetter-Virus Dieser Wurm verschickt sich in einer Email, in der Betreffzeile steht ILOVEYOU, der enthaltene Text ist: kindly check the attached

LOVELETTER coming from me. Die Email hat ein Attachment mit dem Namen LOVE-LETTER-FOR-YOU.TXT.vbs, also VisualBasicScript. Beim Starten des Skriptes wird die Win-Bugsfix.exe ausgeführt. Der Virus kopiert sich in das Systemverzeichnis und das Windowsverzeichnis und überschreibt u.a. mp3, jpeg und vbs Dateien.

### Malware ohne eigenständige Replikation

- **Hoax:** Dies sind Warnungen über Viren oder Trojaner, die nicht existieren. Die Weitergabe erfolgt durch Benutzer, die diese falsche Warnmeldung nicht als falsch identifizieren.
- **Hostile Applets/Agents:** Diese Malware kam mit der größer werdenden Menge an Multimedia-Webseiten auf. Durch Übertragen von Applets und deren Ausführung auf dem System des Benutzers konnten hier aktuelle Daten übertragen werden oder Laufschriften erzeugt werden. Da die Applets aktive Inhalte sind, können sie auf dem Rechner des Benutzers Schaden anrichten. Es ist mit dieser Technologie u.a. möglich, Daten vom Rechner des Benutzers wegzuschicken, wie z.B. PINs Passwörter oder Seriennummern (siehe auch Java).

Es gibt dann noch Malware wie zum Beispiel Viren, die beabsichtigt waren, die aber nicht richtig funktionieren. Sie infizieren eventuell ihre Opfer nicht wie vom Programmierer beabsichtigt, oder ihre zweite Generation kann sich nicht mehr richtig replizieren. Keime sind Viren, die in ursprünglicher Form sind, wie sie vom Programmierer kommen. Scherzprogramme sind eine weitere Kategorie von Malware, die hauptsächlich auf das Erschrecken der Benutzer abzielt, dadurch, dass sie vorgeben, ein Virus oder Ähnliches zu sein. Fehler in der Software, die bei der Implementation oder beim Entwurf entstanden sind, sind als Bugs (Wanzen) bekannt, sie stellen aber keine böswillige Attacke dar und grenzen sich so von anderer Malware ab.

## 2.3 Statistisches zu Malware

Virenmeldungen 1999

Makroviren	64%
Bootviren	14%
Trojaner	12%
Würmer	5%
Dateiviren	3%
sonstige	2%

- Für 1999 berichtet man einen Virenschaden von ca. 12 Mrd. Dollar

- Täglich kommen 10-15 neue hinzu. Es sind ungefähr 200, die für ernsthafte Unruhe sorgen.
- Die 10 meistverbreitetsten sind für ein Drittel der Schäden verantwortlich.
- Man rechnet im Schnitt für eine Virusattacke 7500 Euro Schaden.
- Die Behebung eines Schadens dauert 44 Stunden. 3,3 % der Computer werden im Monat infiziert [Tietz99]

## 2.4 Virens Scanner

Virens Scanner suchen nach Viren, entweder auf Anfrage (on demand), wobei hier meistens größere Teile des Systems überprüft werden, oder beim Zugriff auf eine Datei (on access). Es gibt unterschiedliche Strategien, die bei der Suche angewendet werden.

### Integrity-Check

Der Virens Scanner bestimmt für ein sauberes System die Checksumme, speichert diese und vergleicht sie mit jeweils denen, die er bei den folgenden Scans erhält. Erhält er bei einem der folgenden Scans eine andere Checksumme, so weiß der Benutzer, dass eine der Dateien verändert wurde. Diese kann gezielt durch den Benutzer verändert worden sein oder durch einen Virus verändert worden sein. Man kann bei diesem Check aber weder genau sagen, welche Datei infiziert wurde, noch welcher Virus das System befallen hat. Außerdem kann man nur die Dateien mit dem Integrity-Check schützen, die sich standardmäßig nicht verändern, wie z.B. command.com.

Dokumente, die man ständig verändert, sind für diesen Check nicht geeignet, da mit ihrer Veränderung sich auch die Checksumme verändert. Der Integrity-Check ist sehr schnell, jedoch sehr ungenau bzw. er gibt wenig Informationen über die Verseuchung.

### Signatur Scanner

Der Virens Scanner überprüft bei diesem Verfahren die zu prüfenden Daten auf eine gewisse Signatur, welche er in seiner Datenbank hat. Eine Signatur ist hier ein Auszug aus einem Virus. An dieser Signatur, eine Sequenz von Bytes, soll der Scanner einen Virus erkennen. Der Scanner hat in seiner Datenbank jeweils nur eine Byte-Sequenz des Virus, weil es zu aufwendig wäre, den ganzen Virus abzuspeichern, bzw. zu viel Platz wegnehmen würde in der Datenbank. Bei Updates wäre gerade in Zeiten, in denen man ein Update über das Internet durchführen kann, das Update ziemlich aufwendig bzw. zeit- und kostenintensiv. Da nur eine gewisse Byte-Sequenz und nicht der ganze Virus in der Datenbank des Scanners gespeichert ist, bleibt ein Restrisiko, ob in einer für infiziert befundenen Datei eventuell doch nur ein sauberes Programm ist, was zufällig dem Programmcode eines Virus ähnlich ist.

Die Schwäche dieses Verfahrens zu scannen liegt darin, dass der Virus, der entdeckt werden soll, bekannt sein muss, bzw als Signatur in der Datenbank des Virusscanners vorhanden sein muss. Updates über Virendefinition erhält man heute übers Internet. Das Internet macht es jedoch auch möglich, dass sich die Viren immer schneller verbreiten und man mit den neuen Updates für Virens Scanner nicht so schnell nachkommt.

### **Heuristische Scanner**

Das heuristische Verfahren geht vor, in dem es unter anderem Programmstruktur, Programmierlogik und die Befehle an das Computersystem von dem zu scannenden Programm überprüft. Es wird in zwei Phasen vorgegangen, in der ersten Phase katalogisiert der Scanner das, was das Programm macht, wenn es aktiviert wird. Der Scanner versucht, die Stellen herauszufinden, an denen ein Virusaufenthalt am wahrscheinlichsten ist. Diese Stellen werden, in der zweiten Phase, nach ihrem Inhalt überprüft, wenn ungewöhnliche oder das Betriebssystem schädigende Aktionen hier durchgeführt werden, ist das ein Indiz für einen Virus. Der heuristische Scanner kann also auch Viren erkennen, die eigentlich noch unbekannt sind.

## **3 Mobiler Code**

Mobiler Code sind z.B. Programme, die im Internet mit Hilfe einer Browserverbindung übertragen werden können, und die dann auf dem Zielrechner bestimmte Tätigkeiten ausführen können. Es kann so eine erweiterte Interaktion zwischen dem Benutzer und dem Internetserver erreicht werden. Meistens sind diese Programme in Html-Seiten eingebunden oder zumindest ein Verweis, der sie von der Html-Seite heraus aufruft. Es gibt aber auch die Möglichkeit, dass sie per Email verschickt werden. Im wesentlichen lassen sich folgende Technologien bzw. Programmiersprachen, mit clientseitiger Abarbeitung, als *active Content* also aktive Inhalte, einer Webseite sehen:

- **Java**
- **JavaScript**
- **ActiveX**

Es gibt u.a. noch VisualBasicScript, PlugIns und eventuell andere Skript Sprachen, die aufgelisteten sind aber die am meist verbreitetsten.

PlugIns werden so installiert, dass sie ausgeführt werden können als seien sie Bestandteil des Browsers. Sie erhalten ein MIME-Type (Multimedia Internet Mail Extension) und eine Dateieindung zugewiesen. Anhand des MIME-Type erkennt der Browser, welches PlugIn gestartet werden muss, wenn bestimmte Daten empfangen werden, die natürlich auch diesen MIME-Type haben.

### 3.1 Java

Java gibt es seit 1991, zunächst als Forschungsprojekt von Sun Microsystems, es sollten u.a. Geräte wie Fernseher und Kühlschränke damit programmiert werden. Ein paar Jahre später zeigte man, dass es möglich war, auch größere Anwendungen mit ihr zu programmieren. Mit der Möglichkeit, Applets zu programmieren, die eben jener mobile Code sind, konnte man jetzt Webseiten dynamischer gestalten, was zum Aufstieg von Java beitrug. Applets sind eine Art Programme, die aus dem Internet geladen werden können und im Browser auf Benutzerseite ausgeführt werden können. Man legt die Einführung von Java ungefähr auf 1995.

Zu Java gibt es drei Begriffe, die Programmiersprache Java, die Java Virtual Machine (JVM) und die Plattform Java. Die Programmiersprache Java wird zunächst benutzt, um Java Programme zu schreiben, diese können dann kompiliert werden und sind nun als Bytecode vorhanden. Der Bytecode kann von einer Java Virtual Machine, die auf einer realen Plattform läuft, z.B. Windows, Unix oder Mac, ausgeführt werden. Dabei ist es wichtig zu sehen, dass ein und derselbe Bytecode auf verschiedenen Rechnern, mit unterschiedlichen Betriebssystemen, zum Laufen gebracht werden kann. Die Java Plattform oder auch Laufzeitumgebung ist eine Menge von Klassen, die für ein kompiliertes Javaprogramm da ist, und die von Betriebssystem zu Betriebssystem verschieden ist. Diese *Portabilität* von Java ist eines der Hauptmerkmale dieser Sprache im Gegensatz zu anderen Programmiersprachen wie C oder C++.

Java ist eine objektorientierte Sprache, der Quelltext steht in einzelnen Klassen, diese Art der Modularisierung macht ein Nachladen einzelner Bausteine, der Klassen, während der Laufzeit möglich. Die Sprache Java zeichnet sich unter anderem dadurch aus, dass sie nicht zu mächtig ist, im Gegensatz zu C. So gibt es in Java z.B. keine Zeigerarithmetik, also auch kein Zugriff auf Speicheradressen des Systems, dies schränkt die Fähigkeiten von Java vielleicht ein, aber sorgt auch für Sicherheit.

Die sogenannte Bytecodeprüfung stellt sicher, dass die Bytecodes einer Klasse gültig sind, und dass z.B. der Stack der VM nicht zum Überlauf gebracht wird. Die Bytecodeprüfung ist hauptsächlich dazu da, um zu verhindern, dass Bytecode ausgeführt wird, der die VM zum Absturz bringt, oder der sie in einen nicht vorgesehenen Zustand bringt, in dem Angriffe erleichtert werden. Sie verhindert, dass manipulierter oder von einem nicht vertrauenswürdigen Java Compiler kompilierter Bytecode zur Ausführung kommt.

Schon am Anfang der Entwicklung von Java hat man das Thema der Sicherheit mit berücksichtigt. Applets, die aus dem Internet geladen werden, sind nicht notwendigerweise vertrauenswürdig, man kennt nicht unbedingt den Programmierer von jedem Applet, das man ausführt. Man muss also bei der Ausführung darauf achten, dass die Operationen des Applets keinen Schaden anrichten können bzw. keine Operationen ausgeführt werden, die zur Schädigung missbraucht werden können. Java löst dieses Problem mit einer Zugriffskontrolle, welche dem Code eines Applets gewisse Teile der

API<sup>1</sup> vorenthält und so dafür sorgt, dass z.B. nicht auf den Benutzerrechner geschrieben werden darf. Man kann nie gänzliche Sicherheit garantieren. So gab es z.B. einen Fehler (aufgespürt August 1999), der in der Java Virtual Machine von Microsoft entdeckt wurde, welche im Internet Explorer 4.0 und 5.0 verwendet wird. Dieser Fehler war gefährlich, da er zuließ, dass ein Applet unbeschränkten Zugriff auf das System erhielt.

Die Zugriffskontrolle wird durch den *Sandkasten*<sup>2</sup> geregelt, bei ihm oder vielmehr beim Security-Manager-Objekt wird erfragt, ob eine Operation ausgeführt werden darf, bzw. wird festgestellt, ob der Code als vertrauenswürdig erachtet wird und Zugriff erhält. Der Zugriff soll generell so geregelt werden, dass ein Applet unter anderem zunächst

- keine Dateien schreiben, löschen, lesen oder umbenennen kann, und auch keine Verzeichnisse. Ebenfalls soll auch das Änderungsdatum oder die Länge einer Datei nicht feststellbar sein.
- keine Verbindung zu einem Computer aufbauen kann, der nicht der ist, von dem es heruntergeladen wurde, und auch keine derartigen Verbindungen annehmen kann.
- keine Systemfunktionen, wie z.B. das Starten eines neuen Prozesses oder das Stoppen der VM, ausführen können soll. Es sollen auch keine Threads beeinflusst werden können, die nicht vom Applet stammen.
- keine mit bestimmten Eigenschaften versehenen Graphik oder GUI Funktionen benutzt, keinen Druckauftrag an den Drucker gibt und keinen Zugriff auf die Zwischenablage oder Event-Warteschlange hat.
- keine Systemeigenschaften einsehen kann.

Es soll schließlich nicht möglich sein, dass ein Applet ein neues Security-Manager-Objekt registriert, um die oben genannten Punkte zu umgehen.

Es ist möglich, Klassen digital zu signieren, eine digital signierte Klasse kann so bei einem Webbrowser als vertrauenswürdig eingestuft werden, je nach Signatur und Absender. In Java 1.1 hat man unterschieden zwischen völligem Zugriffsrecht auf die API oder gar keinem Zugriffsrecht, je nachdem, ob die Vertrauenswürdigkeit hergestellt werden konnte oder nicht.

Mit Java 1.2 hat man Policies eingeführt, die es dem Webbrowser erlauben, je nachdem, ob eine Klasse als vertrauenswürdig oder nicht eingestuft wurde, verschiedene Zugriffsrechte zu geben. Es ist dann möglich, Code unterschiedlicher Herkunft verschiedene Zugriffsrechte zu zuweisen.

---

<sup>1</sup>Application Programming Interface, Klassen, die ein Java-Programm benutzt.

<sup>2</sup>Umgebung in der ein Java-Programm nur eingeschränkt läuft.

Es gibt Browser, die es zulassen, dass ein Grant (ein Grant ist eine Art Bewilligung) angefordert wird, dies heißt, dass der Benutzer bei einem Zugriffsversuch des Applets auf eine geschützte Funktion nochmals gefragt wird, ob sie nicht doch freigeben will.

Die auftretenden Risiken bei Java sind zunächst einmal eine fehlerhafte Implementation des Securitymanagers, wodurch es möglich werden kann, dass dieser umgangen werden kann.

Wenn eine Warnmeldung auftritt, kann es sein, dass ein Benutzer diese nicht richtig interpretiert (siehe auch bei JavaScript) und, z.B. im Falle einer Grant Anforderung, eine Freigabe auf Ressourcen macht, die er eigentlich gar nicht will.

Es ist möglich, einen überflüssigen Ressourcenverbrauch zu verursachen, indem man ein Applet mit sinnlosen Befehlen zur Ausführung bringt, wobei es hiergegen zwar Schutzmechanismen gibt, die aber leicht umgangen werden können.

Es gibt eine Methode, die es ermöglicht, andere Webseiten zu laden, die eventuell kostenpflichtig sind. Diese Methoden sind nicht vom Securitymanager erfasst.

Es gibt auch die Möglichkeit Funktionen aufzurufen, die Informationen mit Hilfe von URLs nach außen bringen.

## 3.2 JavaScript

JavaScript ist hauptsächlich vom Namen her ähnlich mit Java, hat aber sonst wenig mit Java zu tun. Es handelt sich um eine einfache objektorientierte Skriptsprache, die von Netscape entwickelt wurde und jetzt von vielen Browsern unterstützt wird, wobei es allerdings keine hundertprozentige Kompatibilität gibt. Es kann folgende, reguläre, Arten des Auftretens von JavaScript geben:

- JavaScript kann innerhalb von zwei SCRIPT-Tags stehen, innerhalb einer Html-Seite. Es ist eine der häufigsten Erscheinungsformen. Beispiel:

```
<Script language="JavaScript">
document.write("Schreiben mit JavaScript");
</Script>
```

- Es kann in einer JavaScript-Datei, auf die durch ein SRC-Attribut im SCRIPT-Tag verwiesen wird, stehen. Es kann aber auch in einer JAR-Datei<sup>3</sup> stehen, auf die ein ARCHIV-Attribut verweisen kann. Beispiel für SRC-Attribut:

```
<Script SRC=Datei.js"JavaScript">
```

- Es kann mit Eventhandlern auftreten, welche optionale Html-Attribute sind. Der Eventhandler kann auf grund eines Ereignisses, wie z.B. das Ziehen des Mauspeils über ein Element, einen aktiven Inhalt ausführen. Dem Eventhandler wird dann ein Funktionsaufruf zugewiesen, der eine JavaScript-Funktion aufruft, die

---

<sup>3</sup>Java Archiv, das Javaklassen, aber, in diesen Fall wichtig, auch JavaScript enthalten kann.

meistens eingebettet, in eine Html-Seite, wie unter Punkt eins beschrieben ist, steht.

- Es kann in einer URL stehen. Es steht hier nach „Javascript:“
- Es kann zur Generierung von Sonderzeichen benutzt werden und steht hinter „&{“ . Die geschweifte Klammer wird wieder geschlossen.
- Es kann bei STYLE-Tags stehen, die mit Cascading Stylesheets<sup>4</sup> verwendet werden. Der STYLE-Tag wird verwendet, um auf einen Bereich zu verweisen, in dem sich Definitionen für ein Textformat oder einen Style finden. In diesem Fall kann der Tag ein Attribut „TYPE=text/javascript“ enthalten. JavaScript kann in einem darauffolgenden Block stehen, der von einem STYLE-Tag beendet wird.

JavaScript kennt keine Typen und hat auch kein ausgefeiltes Sicherheitssystem wie Java. Es fehlen Operationen zur Manipulation von Dateien, was aber nicht ausschließt, dass sich Wege finden lassen, auf Dateien zuzugreifen, die auf dem Benutzerrechner liegen. Mit JavaScript kann man Browserfenster öffnen oder das Erscheinungsbild des Browsers verändern, hauptsächlich wird es jedoch gebraucht, um Webseiten dynamisch zu machen. Beim Netscape Navigator werden die Benutzereinstellungen in einer JavaScript-Datei gespeichert, die beim Browserstart ausgeführt wird. Die Risiken bei JavaScript liegen nun darin, dass z.B.

- die oben genannte Möglichkeit zum Browser Öffnen missbraucht werden kann, um Webseiten mit eventuell kostenpflichtigem Inhalt unerwünscht zu öffnen.
- es sogar möglich ist, unter Umständen Applikationen dazu zu veranlassen, Dokumente downzuloaden.
- der unnütze Ressourcenverbrauch durch Ausführen sinnloser Befehle möglich ist. Es gibt zwar eine Limitierung seitens des Browsers, was die Anzahl der hintereinander auszuführenden Kommandos betrifft, die aber umgangen werden kann.
- ein Applet-Tag generiert werden kann mit Hilfe von JavaScript, um Filter zu umgehen, dabei schleust man zunächst einzelne Tag-Teile wie <AP durch den Filter und setzt diese nachher wieder zusammen.
- die Statuszeile des Browsers mit Hilfe von JavaScript verändert werden kann, so dass Zieladressen eines Links nicht oder falsch angezeigt werden.

Beim Benutzer kann ein Grant erfragt werden, wobei aktive Inhalte dann bei Erteilen des Grants

- Emails versenden können.

---

<sup>4</sup>abk. CCS, wird verwendet, um Texte zu formatieren.

- die Browsereinstellungen verändern können.
- die File-Upload-Fields verändern können und Dateien dadurch uploaden können.
- auf die History-Einstellungen des Browsers zugreifen können und auch auf die Inhalte anderer Browserfenster, welche unter Umständen vertrauliche Informationen enthalten.

Die letzteren Aktionen, die nach Erteilung eines Grant erst ermöglicht werden, haben ihre Tücke darin, dass ein Dialogfenster, was bei einer Anfrage über einen Grant auf geht, zuerst vielleicht nicht in dem Grad über ein eventuelles Risiko informiert, dass jeder Benutzer dies erkennt.

### 3.3 ActiveX

ActiveX, welches man in Konkurrenz zu Java sehen könnte, basiert auf dem Component Object Modul (COM). Es werden spezifische Eigenschaften des Microsoft Betriebssystems benutzt, wobei man für MacOS und einige Unix-Systeme auch Möglichkeiten geschaffen hat, ActiveX zu benutzen. Es gibt für den Browser von Netscape PlugIns, um ihn activeX-fähig zu machen. ActiveX-Controls sind Module, die in verschiedenen Sprachen geschrieben werden können, sie sind die Einheiten eines ActiveX Programmes. Der ausführbare Code kann z.B. als JavaByteCode, JavaScript, VBScript übergeben werden und je nachdem auf einer VM oder Scriptengine der jeweiligen Sprache zur Ausführung gebracht werden.

Generell gilt bei ActiveX, dass die einzelnen Aktionen, die ein Control ausführt, nicht so differenziert geblockt oder freigeschaltet werden können wie bei Java. Es gilt ausführen oder nicht, was ein sehr hohes Sicherheitsrisiko darstellt.

Als Sicherung für den Benutzer gibt es die Möglichkeit, dass ein ActiveX-Control digital signiert sein kann, der Benutzer kann entscheiden, ob er zertifizierte ActiveX-Steuerelemente zur Ausführung zulässt oder nicht.

Wenn man ein Java-ActiveX-Control zur Ausführung bringen will, kann man gleiche Bedenken in Bezug auf die Sicherheit haben wie bei Java, wobei man hier aber ein ganzes Archiv oder Programm freischalten muss. Ein Binärprogramm hat keine Einschränkungen mehr und kann Dateien auf dem Rechner lesen und verändern. Es ist auch möglich, auf ein laufendes Programm zuzugreifen. Es können Rechner im lokalen Netzwerk angegriffen werden. Es können bei manchen Betriebssystemen auch privilegierte Aktionen ausgeführt werden, u.a. können so Daten in einem Broadcast-Netzwerk aufgenommen werden, und diese dann auch über Email oder ftp versandt werden. Diese Möglichkeiten erlauben es, auch Viren oder Trojaner zu installieren.

### 3.4 Angriffe unter Nutzung mehrerer Sprachen

Es gibt die Möglichkeit, einem Angriff unter Verwendung mehrerer Sprachen ausgesetzt zu sein.

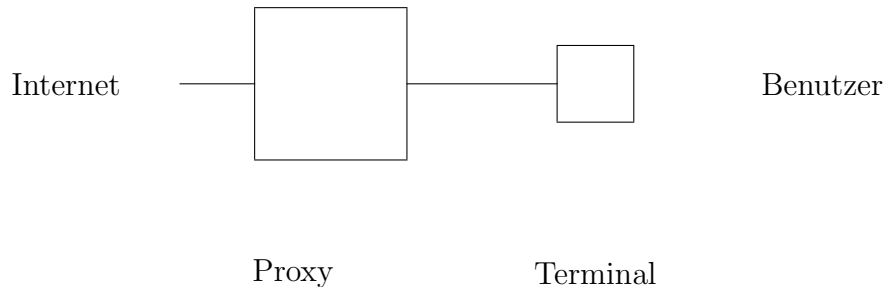
Es ist z.B. möglich, dass man JavaScript mittels Java an einem Filter, der JavaScript herausfiltern soll, vorbeischmuggelt. So ist es nicht sicher, wenn man JavaScript in einem Filter sperrt, aber Java nicht. Bei den Browsereinstellungen ist dies in diesem Fall aber sinnvoll. Die Möglichkeit, einen APPLET-Tag dynamisch zu generieren (siehe JavaScript) mit Hilfe von JavaScript, macht es sinnvoll, ByteCode-Filter einzusetzen.

ActiveX macht es möglich, dass man nicht nur andere Sprachen an einem Filter vorbeischickt, sondern auch, dass man eigene Skriptengines verschickt und so beliebigen, für sie geeigneten Code von ihnen abarbeiten läßt.

## 4 WEB-Sicherheit

Wenn man sich mit hundertprozentiger Sicherheit vor böswilligem Code schützen will, ist es notwendig, alle Applikationen, die mit dem Internet verbunden sind, abzuschalten, oder gleich die physikalische Verbindung zum Internet zu kappen. Eine weitere Möglichkeit ist es, keine aktiven Inhalte zu zulassen, was in der Praxis nicht immer funktioniert, man könnte auch alle Maschinen, die aktive Inhalte abarbeiten, von seinem Computer deinstallieren. Da diese Engines aber meistens nicht nur für den über das Netz geladenen Code, sondern auch für lokal vorhandenen gebraucht werden, ist dies nicht ratsam. Da man heutzutage wohl nicht mehr um aktive Inhalte und deren Nutzung herumkommt, will man idealer Weise alles das, was man braucht und was gut ist, bei sich zur Ausführung bringen und alles das, was schlecht ist, nicht ausführen. Dies endet in der Praxis darin, dass man einzelne Technologien (Java, JavaScript..) jeweils abschalten kann, oder dass diese Technologien Sperrmechanismen haben für gewisse Funktionen, die schädigend genutzt werden könnten, allem vorausgesetzt, es funktioniert und ist implementiert wie vorgesehen. Man kann auch versuchen, vertrauenswürdigen Code zu erkennen und nicht vertrauenswürdigen zu sperren. Für die Entscheidung kann man sowohl Zertifikate als auch die Herkunft von aktiven Inhalten zu Rate ziehen.

## 4.1 Filter und Proxies



Filternde Proxies sind der eigentlichen Anwendung vorgeschaltet und können Webinhalte anhand einer auf ihnen installierten Firewall u.a. nach Herkunft herausfiltern. Es lassen sich mit ihnen auch, wenn sie diese Erweiterung haben, Java-Applets, ActiveX-Controls oder JavaScript herausfiltern.

Es ist sinnvoll, bei JavaScript einen Filter einzusetzen, da es in Html häufig eingebettet ist, und es nicht nur an einem bestimmten Tag auffindbar ist. Es gibt die Möglichkeit, dass es von SCRIPT-Tags umgeben ist, oder aber dass es mit einem Attribut zusammen steht. Die meisten Filter bieten die Möglichkeit, JavaScript herauszufiltern, wenn es in SCRIPT-Tags vorkommt oder wenn es ein Eventhandler ist.

Bei Java ist es auch möglich, es zunächst am APPLET-Tag aufzufinden. Dies ist aber nicht immer möglich, wenn z.B. mit zwei Sprachen, also vielleicht noch JavaScript, ein Angriff gestartet wird, und der Tag zerlegt wird. Es wäre also gut, wenn man die Möglichkeit hätte, den ByteCode zu erkennen.

Die übertragenen Dateien kann man zunächst an der Dateiendung .class erkennen. Der übertragene MIME-Type kann für Javaklassen „application/octet-stream“ sein, wie er auch für andere Binärdateien verwandt wird. Es gibt aber auch „application/java“ und „application/java-vm“. Java kann also nicht immer am MIME-Type erkannt werden. Wenn jemand den Datenstrom abfängt, kann er auch den MIME-Type verändern. Wenn keine JAR-Dateien sondern die Klassen einzeln geladen werden, kann man auch die Dateiendung .class zum Filtern benutzen. Eine Javaklasse kann aber auch eindeutig an den ersten vier Bytes ihres Inhaltes als solche identifiziert werden, welcher lautet 0xCAFEBAFE. Die Version des Compilers, der diese Klasse erzeugt hat, ist in den nächsten vier Byte zu finden. Die Analyse von Klassen im Filter könnte auch dazu benutzt werden, um herauszufinden, ob die Klasse native Klassen verwendet, die auf Dateien oder Netzwerke zugreifen.

Werden JAR-Dateien benutzt, die mehrere Klassen zusammenfassen, kann eine eventuell eingetragene digitale Signatur Vertrauen schaffen. Dieses Zertifikat wird auch angezeigt bei der Erfragung eines Grants. Ein Problem neben der Frage nach der Sicherheit von Zertifizierungen ist hier, dass in JAR-Dateien auch JavaScript enthalten

sein kann. Der Nachteil einer JAR-Datei ist für den Filter auch, dass er diese erst entpacken muss und so ein Performanceverlust bei einer Übertragung entstehen kann.

Die Filterung von ActiveX-Controls ist sehr viel schwerer. Zunächst kann man den OBJECT-Tag in Html-Seiten zur Erkennung benutzen, hier ergibt sich aber das Problem der dynamische Html-Generierung wie beim APPLET-Tag. Es ist also wünschenswert, dass man auch hier den eigentlichen aktiven Inhalt, wie bei Java den ByteCode filtert. Die Uneinheitlichkeit von „ActiveX-Code“ macht es jedoch schwer oder praktisch unmöglich, ihn zu filtern. Die Dateieindungen sind genauso uneinheitlich wie die Art des Codes, so bleibt die digitale Signatur, welche die bekannten Sicherheitsmerkmale hat.

Das Problem an Proxies ist, dass sie nicht die Möglichkeit haben, eine verschlüsselten Datenstrom zu analysieren, wenn sie den Schlüssel nicht haben. So müsste man entweder eine verschlüsselte Verbindung zum Proxy hin und eine von ihm weg aufbauen, oder sich sicher sein, dass der verschlüsselte Strom sicher ist. Ein weiteres Problem ist die rasant weiterschreitende Technik, Browser lassen vielleicht immer weitere Tags zum Aufrufen von aktiven Inhalten zu, desweiteren sind die Html-Standards der unterschiedlichen Browser nicht einheitlich.

Ein Nachteil der Filterung ist, dass ein Performanceverlust in Bezug auf den Datendurchsatz auftreten kann, wenn man nicht genug Ressourcen zur Verfügung stellt.

## 4.2 Zertifikate

Es gibt die Möglichkeit, dass aktive Inhalte ein Zertifikat haben, Browser wie z.B. der Internet Explorer oder Netscape Navigator können so automatisch diesen Inhalten mehr Zugriffsrechte geben, als wenn diese Inhalte nicht digital signiert wurden. Die Zertifikate werden von einer Zertifizierungsstelle vergeben (CA, Certification Authority), sie enthalten Informationen, von wem sie ausgestellt worden sind, für wen, von wann sie gültig sind und bis wann sie gültig sind. Man kann im Browser generell nachgucken, für welche Zertifikate er den aktiven Inhalten mehr Rechte einräumt, und die oben genannten Informationen sind ebenfalls abrufbar. Es ist bei Bedarf möglich, weitere Zertifikate, die akzeptiert werden sollen, hinzuzufügen. Beim Internet Explorer gibt es dann die Möglichkeit, darüber zu entscheiden, ob man unsignierte oder nur signierte ActiveX-Steuerelemente herunterladen will, eine ähnliche Möglichkeit gibt es für Java, hier kann man signiertem und unsigniertem Inhalt jeweils verschiedene Zugriffsrechte einräumen. Es wird angenommen, dass eine Signatur gültig ist, wenn

- man das Zertifikat mit dem gleichen Schlüssel signiert hat wie der, der im Root-Zertifikat eingetragen ist.
- der Schlüssel, der verwendet wurde, um zu signieren, mit dem identisch ist, der im Zertifikat der Person oder der Identität angegeben ist.
- ein Root-Zertifikat dem Browser bekannt ist, dass von der CA stammt, die das Zertifikat des Signierenden ausgestellt hat.

Bei der Prüfung der Gültigkeit muss man bedenken, dass Zertifikate nicht notwendigerweise auf noch Bestehen der Gültigkeit geprüft werden. Auch kann es sein, dass Zertifikate zurückgezogen wurden. Mit der Authenticode<sup>5</sup> Technik von Microsoft ist es möglich, diese Fragen zu klären.

Will man näher auf die eigentliche Sicherheit von Zertifikaten bzw. die Sicherheit, die man sich durch sie erhofft, eingehen, muss man sich zunächst die Frage stellen, ob eine Datei sicher ist, allein dadurch, dass sie signiert wurde. Die Antwort ist kein uneingeschränktes Ja, so ist lediglich gewährleistet, dass eine Person, die von einer CA dazu autorisiert wurde, eine Signatur vergeben hat. Es könnten Inhalte signiert worden sein, die der Signierende gar nicht selbst programmiert hat. Ebenfalls müßte man immer darauf achten, dass Leute, die sich ein Zertifikat haben ausstellen lassen, dies auch nur verwenden, wenn sie hundertprozent sicher sind, dass ihr Programm nicht schädlich ist. Die Aussage, die über ein mit Zertifikat ausgestattetes Objekt gemacht werden kann, ist, wer es signiert hat, wobei dies voraussetzt, dass die Identität des Signierenden bei der Ausstellung die richtige war bzw. dass die richtige angegeben wurde. Nach eventueller Schadennahme durch ein zertifiziertes Objekt ist zur Identitätsbestimmung natürlich auch notwendig, dass diese Informationen noch verfügbar sind. Bei Löschung der Festplatte durch ein schadhaftes aber zertifiziertes Applet kann so der Urheber meistens nicht mehr ausfindig gemacht werden.

### 4.3 Sicherheit bei Internet Explorer

Der Internet Explorer hat verschiedene Einstellungsmöglichkeiten in Bezug auf aktive Inhalte, die hier kurz vorgestellt werden sollen. Zunächst gibt es, wenn man im Menü Extras die Internetoptionen aufruft und die Karteikarte Sicherheit anklickt, einige Zonen, die letztlich verschiedene Menge von Netzwerkverbindungen bilden. So bildet die

- **Zone für Eingeschränkte Sites** die Menge aller Netzwerkverbindungen, die der Nutzer als nicht sicher erachtet hat und hier anhand ihres Domain Names oder der IP-Adresse **selbstständig** eingetragen hat.
- **Zone für vertrauenswürdige Sites:** die Menge aller Netzwerkverbindungen, die der Nutzer als sicher erachtet hat und hier anhand ihres Domain Names oder der IP-Adresse **selbstständig** eingetragen hat.
- **lokale Intranetzone:** die Menge aller Netzwerkverbindungen, deren URL keine Punkte enthält und die in keiner anderen Zone eingetragen sind; auf die der Zugriff über CIFS<sup>6</sup> also NetBIOS erfolgt; auf die nicht mittels Proxy zugegriffen wird.

---

<sup>5</sup>Authenticode prüft, ob das Zertifikat noch gültig ist, und die Identität des Softwareherausgebers mit der des Zertifikats stimmig ist.

<sup>6</sup>Common Internet Filesystem, ein Internet Dateisystem

- **Internetzone:** die Menge aller Netzwerkverbindungen, die zu keiner anderen Zone gehören. Bei einem Einzelrechner ohne LAN und durchschnittlichem Internetgebrauch dürften die meisten Netzwerkverbindungen in diese Zone fallen.
- Zone mit Namen **Arbeitsplatz:** nur der lokale Rechner ist in dieser Zone.

Beim Internetverkehr wird normalerweise die Internetzone benutzt, darum soll an ihr hier erklärt werden, welche weiteren Einstellungen man vornehmen kann. Die Internetzone verfügt über gewisse Standard-Einstellungen, die man einstellt, dadurch, dass man den Button „Standardstufe“ anklickt. Es gibt vier verschiedene Stufen, die man, will man nur eine grobe Einstellung vornehmen, wählen kann (Hoch, Mittel, Niedrig und sehr Niedrig). Diese vier groben Stufen stellen automatisch eine Feinabstimmung ein, die man über den Button „Stufe anpassen“ erreicht. Man sollte eigentlich diese Feinabstimmung selbst vornehmen und nicht eine der vier groben Stufen wählen, wenn man nämlich die Einstellung Hoch hier wählt, sind trotzdem die Ausführung von Skripten und Java-Applets aktiviert. Man kann bei diesen Feinabstimmungen u.a. folgende Punkte wählen:

- ActiveX-Steuerelemente ausführen, die für das Scripting sicher sind: Hierbei handelt es sich um die Frage, will man ActiveX zulassen, wenn es vom Hersteller für sicher erachtet wurde, und wenn es durch JScript oder VBScript gesteuert oder zur Ausführung gebracht wird. Es soll Beispiele geben, wo diese Einschätzung nicht richtig war. Also sollte man diese Funktion deaktivieren.
- ActiveX-Steuerelemente initialisieren und ausführen, die nicht sicher sind: Diesen Punkt sollte man auch deaktivieren.
- Download signierter ActiveX-Steuerelemente: Der Hersteller kann seine ActiveX-Controls mit einer digitale Signatur versehen. Vertraut man diesen Elementen, kann man hier aktivieren einstellen, jedoch hat sich bei dem Thema Zertifikate gezeigt, dass man auch bei diesen Elementen nicht sicher sein kann, dass sie keinen Schaden anrichten, also Empfehlung deaktivieren.
- unsignierte ActiveX-Steuerelemente downloaden: Wenn man signierten Elementen nicht traut, sollte man diese wohl auch nicht benutzen.

Es zeigt sich, dass ActiveX hier fast keine Chance kriegt, benutzt zu werden, wer aber nicht darauf verzichten will, sollte sich die „Eingabe“ Option aber immer noch offen halten. Ein weiterer aktiver Inhalt wird zum Beispiel von Java dargestellt, wenn man hier „Benutzerdefiniert“ einstellt, kann man in gleichem Fenster die Java-Einstellungen aufrufen. Es ist möglich, sich die Zugriffsrechte, die Java hat, anzeigen zu lassen, oder sie zu verstellen oder sich auch die Zugriffsrechte anzugucken, wie sie eingestellt werden, wenn man die etwas gröberen Einstellungen wie mittlere, hohe Sicherheit usw. vornimmt. Für JavaScript und VBScript gibt es die Einstellungen

- Aktiviertes Scripting, welches sich auf die Ausführung von JavaScript und VBScript bezieht, sollte man deaktivieren, was allerdings dazu führt, dass man einiges an Funktionen bei vielen Websites verliert. Vielleicht ist es sinnvoll, das Scripting für nur einige Sites freizuschalten.
- Einfügeoperationen über Scripts: Diese Option sollte man dann auch nur sinnvollerweise für einige Websites aktivieren.
- Scripting von Java-Applets: siehe vorheriger Punkt.

Weitere für die Sicherheit wichtige Einstellungen findet man aber auch noch auf der Registerkarte „Erweitert“, wenn man sich in den Internetoptionen befindet. Punkte wie „Zählen der übertragenen Seiten aktivieren“, „Auf zurückgezogene Serverzertifikate prüfen“, „Auf zurückgezogene Zertifikate von Herausgebern prüfen“ und „Bei Site-Zertifikaten warnen“ sind auch für die Sicherheit bei aktiven Inhalten relevant.

#### 4.4 Netscape Browser

Die Einstellungen in Bezug auf Java und JavaScript können in „Erweitert“ in den Einstellungen vorgenommen werden. Diese Einstellungen können hier nur allgemein vorgenommen werden, es gibt keine Möglichkeiten, nach Webseiten zu differenzieren. Es kann aber ein Aus- und Anschalten ziemlich spontan erfolgen, was wichtig ist, da u.a. die Hilfe des Browsers auf JavaScript angewiesen ist. Java Applets und JavaScripts können, wenn die Funktionen angeschaltet sind, ausgeführt werden. Sie erhalten allerdings die weiteren Rechte nur, wenn sie eine digitale Signatur tragen, die der Browser akzeptiert. Wird ein Applet oder JavaScript freigeschaltet, so hat jedes Applet und jedes JavaScript, welche das gleiche Zertifikat haben, automatisch diese Rechte, ohne dass es zu einer Rückfrage kommt. Die erweiterten Zugriffsrechte umfassen u.a. Sachen wie Freigabe von Lese- und Schreibrecht auf lokale Dateien.

ActiveX wird nicht standardmäßig vom Netscape Browser ausgeführt, es gibt aber ein PlugIn, wobei man die Sicherheitsrisiken bei ActiveX bedenken sollte.

#### 4.5 Email Programme

Bei Emailprogrammen ist es wichtig, dass ein sofortiges Ausführen des Attachements verhindert wird; hier sei gesagt, dass auch in Html-Emails eingebettete Skripte und die Html-Seite als solches gesehen werden müssten. Es empfiehlt sich vor dem Öffnen der Emails, den Sender zu überprüfen. Es kann sein, dass auch eine Email gesandt wurde, die eine Absendeadresse trägt, dessen Inhaber gar nicht die Email gewollt abgeschickt hat.

## 4.6 Sicherheit durch das Betriebssystem

Das Betriebssystem als Grundlage der Anwendungen könnte auch zusätzlich benutzt werden, um Angriffe einzuschränken oder abzuwehren. Bei Betriebssystemen, die nicht nur einen Benutzer zulassen, sondern mehrere Benutzer haben, die in ihren Zugriffsrechten gegenseitig von einander getrennt sind, könnte man einen Benutzer einrichten, der dann wenige Zugriffsrechte auf Anwendungen und Dokumente, die auf dem System sind, hat, aber der Internetzugriff hat. Dieser Benutzer kann dann relativ sicher sein, dass auf die Dokumente, auf die er schon als Benutzer kein Zugriffsrecht hat, auch aus dem Internet durch aktive Inhalte nicht zugegriffen wird.

Der unnütze Ressourcenverbrauch durch Malware ist eine weitere Problematik, die man lösen kann, indem man, z.B. dem Benutzer nur einen gewissen Platz auf der Festplatte einräumt. Auch einzelne Prozesse können verschieden priorisiert werden, um so Ressourcen an der richtige Stelle zu sparen, hierbei kann der Administrator die Priorität erhöhen, aber die Benutzer haben nur die Möglichkeit, die Priorität ihrer Prozesse zu verringern. Von Unix-Systemen kommt ein Konzept, welches sich „ROOT-Reserve“ nennt, hier werden den Benutzern nicht die gesamten Ressourcen bereitgestellt, um im Notfall dem Administrator genügend Möglichkeiten offen zu lassen, um Probleme zu beheben. Die meisten Dateisysteme bei Unix besitzen deswegen Teile, auf die nur der Administrator schreiben darf.

Beim Windows Betriebssystem gibt es die Möglichkeit, Dateien zu öffnen, indem man sie anklickt, es wird dann das eigentliche Programm gestartet, und die Datei etwa ein Textdokument eingelesen, es ist auch möglich, dass eine Datei, die als Email verschickt wird, bei Öffnen der Email auch geöffnet wird. Man kann beim Betriebssystem die Dateibindung an das Programm aufheben oder das Programm selbst deinstallieren, um zu verhindern, dass eventuelle aktive Inhalte in den einzulesenden Dateien zur Ausführung kommen. Dies geht natürlich nur, wenn das Programm nicht benötigt wird.

## 4.7 Fazit

Man sieht, dass es möglich ist, Schutzmechanismen zu umgehen, und man sieht auch, dass es möglich ist, dies gerade durch Anwendung mehrerer Programmiersprachen zu tun, wenn man sie zusammen benutzt. Es ist trotzdem immer ratsam, die Schutzmöglichkeiten, die man hat, zu benutzen. Es ist bereits möglich, mit Einstellungen am Browser anzufangen, wer hier alle aktiven Inhalte verbietet, hat hier Vorteile, doch nur so lange er nicht auf Seiten trifft, auf denen aktive Inhalte zur Ausführung gebracht werden müssen, um ein gewisses Ziel zu erreichen. Nun hat sich das Internet in letzter Zeit zu einem stärker interaktiven und multimedialen Netz entwickelt, was die stärkere Nutzung aktiver Inhalte mit sich bringt. Auch die Nutzung des Internets für Einkauf und Bankgeschäfte bringt in der Regel stärker die Nutzung von activ content mit sich.

Wer einige Seiten hat, die er häufiger besucht, kann das Zonen Konzept anwenden, um diesen Seiten in Bezug auf die Ausführung aktiver Inhalten mehr Rechte ein-

zuräumen. Die Grenzen dieser Handhabung tun sich auf, wenn man sieht, dass Angriffe möglich sind, die das Zonenkonzept aushebeln, und wenn man bedenkt, dass nicht nur von außen nach innen eine Schädigung erfolgen kann, sondern dass auch ein Trojaner von innen z.B. zu schützende Informationen herausenden kann. Ein Trojaner könnte vielleicht auch ganze Aufträge manipulieren und Transaktionen verfälschen.

Gerade der Umstand, dass man sich einmal sehr frei und ein anderes mal sehr sicher bewegen möchte im Internet, legt die Überlegung nahe, sich zwei Computer anzuschaffen, oder zumindest zwei von einander getrennte Systeme oder zumindest Festspeicher zu nutzen. Jeder dieser Festspeicher enthält ein komplettes Betriebssystem. Wenn ein System aktiv ist, ist es notwendig, dass es keine Möglichkeit gibt, dass ein Datenaustausch zwischen beiden Systemen zu stande kommt, man schließt also am besten immer nur eine der beiden Festplatten oder andere Speicher an den Rechner an. Ein System könnte man so konfigurieren, dass nur das minimale Risiko eingegangen wird, auf ihm wären sensitive Daten, und es würde für die Bankgeschäfte genutzt. Das andere System konfiguriert man von der Sicherheit her freizügiger, so dass man hier die Möglichkeit hat, aktive Inhalte intensiver zu nutzen. An dieser Stelle sei gesagt, dass es grundsätzlich richtig ist, sich von wichtigen Daten eine Sicherungskopie zu erstellen, so dass man im Ernstfall nicht alle Daten verliert.

Die Zertifizierung von aktiven Inhalten ist ein weiterer Beitrag zur Sicherheit, er sollte aber nicht überschätzt werden, da nicht eindeutig sichergestellt ist, dass ein zertifizierter Inhalt heißt, dass es sich um unschädlichen Code handelt. Ein Zertifikat hat in der Praxis nur einen minimalen Anteil an der Sicherheit.

Unbedingt sinnvoll und erforderlich sind Virens Scanner, sie sollten auch auf Computern installiert sein, die nicht über einen Netzwerkanschluß verfügen, aber trotzdem neue Daten, wie neue Programme oder Dokumente erhalten. In diesem Zusammenhang sei noch einmal auf die Problematik der Makroviren, die sich meist in Dokumenten befinden, wie sie im Büroalltag vorkommen (z.B. Textdokumenten, Tabellenkalkulationsdokumenten) hingewiesen. Die Signaturscanner haben den Nachteil, dass sie der Entwicklung hinterher laufen, weil sie immer nur das finden, was schon jemand entdeckt haben muss. Man kann aber auch zusätzlich einen heuristischen Scanner verwenden, der wie oben beschrieben arbeitet, und so auch noch nicht registrierte Viren mit hoher Wahrscheinlichkeit entdeckt. Beide Verfahren bieten natürlich zur Virenerkennung keinen absoluten Schutz. Moderne Virens Scanner bieten beide Scanverfahren, sowie eine im hintergrundlaufende On-Access Suche und die obligatorische On-Demand Suche. Der Schutz vor aktiven Inhalten ist allerdings durch Virens Scanner nicht gewährleistet.

Die Möglichkeit, filternde Proxies einzusetzen, sollte man, wenn es geht, nutzen. Sie können eine Menge aktiver Inhalte herausfiltern, bevor diese den Browser erreichen, sie haben jedoch den Nachteil, dass sie bei komprimierten und verschlüsselten Datenströmen nicht wirken. Man sieht, dass es viele Maßnahmen gibt, die man einsetzen kann. Es ist auch gut, wenn man sie zu mehreren einsetzt. Darüber hinaus ist es aber wichtig, dass der Benutzer selbst Grundlegendes weiß, z.B., dass man wissen sollte, wo ein Programm herkommt, welches man ausführen will, oder wo es überhaupt zur Ausführung aktiver Inhalte kommen kann.

## Literatur

- [VTC2000] Virus Test Center. „Viren und Malware“, Universität Hamburg, Fachbereich Informatik, 2000
- [NTC2000] Netzwerk Test Center, „Sicherer ins Internet unter Windows 98 und Windows 98 SE“, 2000
- [Tietz99] W. Tietz, „Computer-Viren“, Online unter <http://www.home.t-online.de/home/walter.tietz>
- [Fla 2000] D. Flanagan, „Java in a Nutshell“, O'Reilly, 2000
- [MüBeKe99] H.-J. Mück, C. Benecke, Stefan Kelm, „Sicherheit in vernetzten Systemen“, 1999
- [DFN99] Deutsches Forschungsnetz, 6. Workshop „Sicherheit in vernetzten Systemen“, 1999