DuD
Datenschutz und Datensicherheit

Sandbox-Modell

Dirk Fox

Gateway (Tor, Durchgang) ist ein Vermittlungscomputer, der zwei unterschiedliche, aber gleichartige Kommunikationssysteme verbindet. Dazu setzt er die Kommunikationsregeln (Protokolle) des einen in die des anderen um. So ermöglicht er den Teilnehmern beider Systeme, sich zu verständigen. In unserem „Gateway" werden Juristen technische und Technikern juristische Begriffe erläutert.

Mit der zunehmenden Vernetzung von Rechnern gewinnen neben einheitlichen Übertragungsprotokollen und kompatiblen Datenformaten Plattform-unabhängige Programme an Bedeutung („write once, run anywhere"). Ein erster Schritt in diese Richtung war die Entwicklung von „Makro-Sprachen" für auf unterschiedlichen Betriebssystemen verbreitete Anwendungen wie Textverarbeitung oder Tabellenkalkulation. Makros sind eine Folge von Anweisungen, die von einer Anwendung interpretiert und ausgeführt werden können. Der automatische Aufruf beschleunigt dabei die Bearbeitung.

Da Makros auch in Dateien gespeichert werden können, erlauben sie die Integration einfacher „Programme" in Dokumente und heben damit die Trennung von „Datei" und „Programm" auf. Mächtige Makro-Sprachen machen daher auch Dateien anfällig für Viren – sogenannte „Makro-Viren".

Java und ActiveX

Einen Schritt weiter gingen (neben anderen ähnlichen Entwicklungen) im Mai 1995 Sun Microsystems mit der Entwicklung von „Java" und Microsoft mit „ActiveX":

- Java ist eine objektorientierte Progammier*sprache*, die die Entwicklung von sogenannten „Applets" erlaubt.

Deren Programmcode ist von Betriebssystem und Rechnerhardware unabhängig – er benötigt zur Ausführung lediglich einen „virtuellen Java-Computer" (*Java Virtual Machine*, JVM), d.h. eine Software, die den Java-Bytecode in die Maschinensprache des jeweiligen Rechners umsetzt.

- ActiveX ist eine Programmier*schnittstelle* für die Microsoft-Betriebssysteme Windows 9x und Windows NT.

Sie gibt sogenannten „ActiveX-Controls", in beliebiger Programmiersprache entwickelt, den Zugriff auf Betriebssystemfunktionen frei.

Java-Applets und ActiveX-Controls können in WWW-Seiten, also Dateien, integriert werden. Lädt ein Internet-Nutzer eine solche Seite und enthält sein Internet-Browser eine JVM (z.B. HotJava oder Netscape-Navigator ab Version 2.0) bzw. erlaubt dieser den Zugriff auf Betriebssystemfunktionen (Microsofts Internet-Explorer), dann wird das Applet bzw. Control beim Laden ausgeführt, sofern diese Funktion nicht deaktiviert wurde. Damit können WWW-Seiten prinzipiell auch Schadensprogramme (wie trojanische Pferde oder Viren) enthalten.

Sicherheitskonzept

Anders als Microsofts Spezifikation, die einem ActiveX-Control den vollen Zugriff auf das Betriebssystem des lokalen Rechners freigibt,[1] hat Sun für Java ein eigenes Sicherheitskonzept entwickelt. Im Kern beruht es auf dem sogenannten „Sandbox-Modell": Innerhalb einer von der JVM kontrollierten Umgebung darf ein Applet tun, was es möchte – auf die Umgebung außerhalb dieses „Sandkastens" (lokale Dateien, Netz) hat es jedoch keinen oder beschränkten Zugriff.

Das Sandbox-Modell umfaßt vier Sicherheitskomponenten [FrMu_96]:

- die *Java-Sprachspezifikation*, die sicherheitskritische Elemente (wie z.B. direkte Speicherzugriffe, die Verwendung von Zeigern und Typänderungen von Objekten) verbietet,
- einen *Bytecode-Verifizierer*, der als Teil der JVM den Code eines Applets vor Ausführung auf Einhaltung der Java-Sprachregeln und die Abwesenheit von Programmfehlern (wie z.B. negative Feld-Indizes, Stack-Überläufe etc.) überprüft,
- einen *Security-Manager*, der während der Codeausführung eine Überschreitung der in der Implementierung konfigurierten „Ränder" des Sandkastens (Zugriffsrechte auf lokale Dateien, erlaubte Internet-Verbindungen; meist: kein Dateizugriff, Netzverbindung nur mit Herkunftsserver) und den Zugriff auf andere Applets verhindert, sowie
- einen *Class Loader*, der nur die Objekt-Klassen lädt, auf die das Applet nach den Vorgaben des Security-Managers zugreifen darf.

Die Umsetzung dieses Sicherheitskonzepts hängt dabei allein von der Implementierung des „virtuellen Java-Computers" (JVM) ab. Die meisten dokumentierten Java-Sicherheitslücken betreffen solche Implementierungsfehler [DeFW_96]. Sun entwickelte daher zusammen mit Blackwatch Inc. ein formales Sicherheits-Referenzmodell sowie Testprogramme, mit denen JVM-Implementierungen auf Einhaltung des Referenzmodells überprüft werden sollen.

Seit dem Java-Release 1.1 können Authentizität und Integrität von Java-Applets sowie anderer über das Netz geladener Dateien mit digitalen Signaturen sichergestellt werden. Damit lassen sich z.B. vertrauenswürdige Applets von anderen unterscheiden.[2]

Erweiterungen des Sicherheitsmodells erlauben nun auch die Vorgabe einer Sicherheits-Policy, die z.B. ausgewählten, vertrauenswürdigen Java-Applets (bspw. von Servern eines unternehmensinternen Intranets) erweiterte Freiheiten, d.h. einen „größeren Sandkasten" einräumt.

Literatur

[DeFW_96] Dean, Drew; Felten, Edward W.; Wallach, Dan S.: *Java Security: From HotJava to Netscape and Beyond*. In: Proceedings of IEEE Symposium on Security and Privacy, 1996.

[FrMu_96] Fritzinger, J. Steven; Mueller, Marianne: *Java Security*. White paper, Sun Microsystems, 1996.

[Sun_96] Sun Microsystems: *Frequently Asked Questions: Java Security* (FAQ), http://java.sun.com/sfaq.

[1] Die einzige Schutzmöglichkeit vor Angriffen (wie dem vom Chaos-Computer-Club demonstrierten auf eine lokale Home-Banking-Anwendung) ist, ActiveX im Browser zu deaktivieren.

[2] Einen ähnlichen Mechanismus verwendet Microsofts „AuthentiCode" zum Schutz von ActiveX-Controls.