Bundesamt
für Sicherheit in der
Informationstechnik

Sicherheit von Java

Grundlagen und Sicherheitsempfehlung

Java ist eine weitverbreitete Programmiersprache und Laufzeitumgebung, was oft zu gewissen Missverständnissen führt. Aus diesem Grund werden in der vorliegenden Veröffentlichung wichtige Begriffe zu Java kurz dargestellt und grundlegende Informationen zu Java und zum Java-Sicherheitskonzept zusammengefasst. Diese richten sich sowohl an Bürger im Allgemeinen als auch an fortgeschrittene Benutzer.

Java genießt in der IT-Welt eine hohe Verbreitung, was sie jedoch gleichzeitig zu einem beliebten Ziel für Angriffe und Missbrauch macht. Aufgrund der in letzter Zeit gemeldeten Sicherheitslücken, die im Browser ausgeführte Java-Inhalte betreffen, empfiehlt das BSI zurzeit grundsätzlich die Deaktivierung von Java im Browser.

Was ist Java?

Hinter dem Begriff Java verbergen sich unterschiedliche Technologien, was häufig zu Missverständnissen führt. Das folgende Kapitel soll einen kurzen Überblick verschaffen.

Java als Programmiersprache

Java ist eine objektorientierte Programmiersprache, die aufgrund ihrer Plattformunabhängigkeit und Einfachheit gegenüber Maschinensprachen oder prozeduralen Sprachen wie beispielsweise C aber auch aufgrund der Möglichkeit, Programme direkt im Browser einzubinden und auszuführen, eine weite Verbreitung gefunden hat.

Java als Laufzeitumgebung

Neben den Entwicklungskomponenten wird unter Java meist die Laufzeitumgebung (Java Runtime Environment, JRE) verstanden. Die Laufzeitumgebung ist die Voraussetzung, dass Java-Applikationen wie Open-/LibreOffice oder die AusweisApp ausgeführt werden können.

Java ist *nicht* JavaScript

An dieser Stelle soll auf einen weitverbreiteten Irrtum hingewiesen werden: Java ist *nicht* JavaScript! Hierbei handelt es sich lediglich um eine unglückliche Namensähnlichkeit. Im Gegenzug zu Java ist JavaScript eine sog. Skriptsprache, die als Teil einer Webseite im Browser des Benutzers ausgeführt wird.

OpenJDK und IcedTea Java Plug-in

OpenJDK ist die Open Source Alternative zu Java von Oracle und wird für verschiedene Linux-Distributionen als Standardinstallation angeboten.

Bei IcedTea handelt es sich um ein Open Source Java Web Browser Plug-in sowie eine Open Source Implementierung der Web-Start-Technologie[1].

Varianten von Java

Java ist in verschiedenen Ausprägungen und für unterschiedliche Betriebssysteme verfügbar, von denen die wichtigsten hier kurz dargestellt werden.

Clientseitig

Java-Applikationen

Java-Applikationen sind eigenständige Programme, die lokal auf dem Computer ausgeführt werden. Hierzu benötigen Java-Applikationen die bereits erwähnte Laufzeitumgebung. Eine solche Java-Applikation ist mit einer ausführbaren .exe-Datei aus der Windows-Welt vergleichbar, die allerdings nicht unmittelbar auf dem Betriebssystem zur Ausführung kommt, sondern in der Java-Laufzeitumgebung, die zur Erreichung einer Plattformunabhängigkeit des Programmcodes eine zusätzliche Abstraktionsschicht zwischen Betriebssystem und Anwendung bereitstellt.

Java Web Start

Eine weitere Art zur Verwendung einer Java-Anwendung ist die Web-Start-Technologie. Auf Grundlage des Java Network Launching Protocol (JNLP) werden Applikationen über den Browser aus dem Internet geladen und in einem Anwendungscache abgelegt. Die Applikationen bzw. deren einzelne Teile liegen dabei in Form von .jar-Dateien (Java Archive) vor, welche auch eine Signatur-Möglichkeit beinhalten. Anschließend kommen sie jedoch nicht im Browser, sondern in der lokalen Java-Laufzeitumgebung zum Einsatz, wie dies auch bei lokalen Java-Anwendungen der Fall ist. Somit wird nicht die eigentliche Java-Anwendung im Browser ausgeführt, sondern es wird lediglich die initial erforderliche JNLP-Datei – ein XML-Format – über den Browser heruntergeladen und an die Java-Laufzeitumgebung übergeben. Schwachstellen im Browser-Plug-in sind daher für Web Start Anwendungen nicht relevant.

Java Applets

Java Applets werden in HTML-Seiten eingebettet und kommen im Gegensatz zu den zuvor genannten Ansätzen direkt innerhalb des Browsers oder lokal über den Appletviewer zur Ausführung. Um Applets im Browser ausführen zu können, wird das sog. Java Plug-in benötigt. Weiterführende Informationen zu diesem Thema finden Sie auf der BSI-Webseite für Bürger (www.bsi-fuer-buerger.de).

Serverseitig

Im Bereich der Webanwendungen ist Java neben PHP und .NET die wichtigste Softwareplattform. Hier sind insbesondere Java Servlets und Java Server Pages (JSP) zu nennen. Servlets sind Java-Klassen, die innerhalb eines Web-Servers ausgeführt werden und eine Ausgabe in Form von HTML generieren. JSP hingegen ähneln den klassischen HTML-Dateien, in denen Java-Codefragmente eingefügt wurden. Diese werden vor der Auslieferung an den Client zu Servlets kompiliert und in der JVM des Web-Servers ausgeführt.

1 http://de.wikipedia.org/wiki/Java_Web_Start

In beiden Fällen wird der Java-Programmcode auf dem Web-Server statt auf dem Client ausgeführt. Der Client selbst benötigt in der Regel keine Java-Laufzeitumgebung, um diese Webanwendungen zu benutzen. Schwachstellen von Java betreffen hier also zunächst den Server.

Weitere Java-Konstrukte

Neben den zuvor genannten Varianten von Java existieren weitere Java-Konstrukte für andere Gerätearten, wie Java Micro Edition für mobile Geräte oder Java für Embedded Systems.

Android-Apps werden ebenfalls in Java – und ggf. ergänzendem nativen Code – geschrieben. Die Java-Laufzeitumgebung in Android ist bezüglich der Architektur und der integrierten Sicherheitsmechanismen durchaus mit dem ursprünglichen Java vergleichbar. Allerdings wurde die Laufzeitumgebung von Google neu entwickelt, sodass Sicherheitslücken in der von Oracle gepflegten Laufzeitumgebung nicht automatisch die Android-Variante betreffen.

Auf die speziellen Varianten von Java sowie auf Android-Apps wird nicht weiter eingegangen.

Wo finden Sie Java nicht

Die Java-Laufzeitumgebung ist auf iOS-Geräten, wie zum Beispiel dem iPhone oder iPad, nicht verfügbar.

Das Java-Sicherheitskonzept

Java bietet eine Vielzahl interessanter und durchdachter Sicherheitskonzepte sowie architektonischer Eigenschaften, die in anderen Softwareplattformen nicht oder nur teilweise verfügbar sind. Ein wichtiger Aspekt ist dabei die strenge Typprüfung sowohl während des Kompilierens als auch zur Laufzeit. Java bietet auch die Möglichkeit an, die Zugriffe auf Klassen (public, protected, final) oder Variablen gezielt zu regeln. Das sichere Management des Hauptspeichers, die mehrstufige Verifikation des Bytecodes sowie gängige Sicherheitsmechanismen, wie Verschlüsselung, Signieren des Codes und gesicherte Netzwerkkommunikation, gehören ebenfalls zum Sicherheitskonzept von Java.

Wie bereits beschrieben, werden Applets in HTML-Seiten eingebettet und innerhalb des Browsers ausgeführt. Besucht der Anwender eine solche Seite, wird der Code des Applets automatisch heruntergeladen und, ggf. nach Bestätigung eines Warnhinweises, zur Ausführung vom Java-Plugin im Browser an die Laufzeitumgebung geleitet. Handelt es sich dabei um Schadcode, der eine Schwachstelle in der Laufzeitumgebung ausnutzt, kann – je nach Art dieser Schwachstelle – das jeweilige System teilweise oder vollständig kompromittiert werden und für die Zwecke des Angreifers missbraucht werden. Um dies zu vermeiden, unterliegen Applets gewissen Sicherheitseinschränkungen.

Applets sowie Web Start Anwendungen werden nur innerhalb einer *Sandbox* (engl. übersetzt: „Sandkasten“) ausgeführt. Die Sandbox hat die Aufgabe, den Code vom Rest des Systems abzukapseln und somit potenziell gefährliche Operationen, z. B. das Schreiben auf die Festplatte oder den Verbindungsaufbau zu fremden Servern, zu unterbinden. Diese Aufgabe erfüllt die Sandbox, indem sie beim Start des Browsers oder des Appletviewers stets den Sicherheitsmanager (Security Manager) aufruft, der wiederum mithilfe von Sicherheitsrichtlinien (Java Security Policy) die Zugriffsrechte auf kritische Funktionen, z. B. Netzwerkaktivitäten, und Informationen sehr detailliert festlegt und verwaltet.

Die oben beschriebenen Sicherheitseinschränkungen gelten nicht für Java-Applikationen, signierte Applets oder Applets, die sich lokal auf dem Rechner befinden und über den Appletviewer aus-

geführt werden, da diese als vertrauenswürdig erachtet werden.

Sicherheitsempfehlung

Grundsätzlich verfügt Java über eine Reihe sinnvoller und durchdachter Konzepte und Mechanismen, um Anwendungen sicher betreiben zu können. Jedoch ist die Anzahl von bekannt gewordenen Schwachstellen in der Java-Laufzeitumgebung, welchen der Hersteller Oracle in der jüngeren Vergangenheit immer wieder durch Sicherheitsaktualisierungen begegnen musste, signifikant hoch. Viele dieser Java-Schwachstellen werden bereits vor der Bereitstellung von Sicherheitsaktualisierungen aktiv für Angriffe ausgenutzt, u. a. in weit verbreiteten sog. Exploit-Kits, die großflächige Angriffe mit geringem Aufwand für den Angreifer ermöglichen. Gleichzeitig dauerte es oftmals unverhältnismäßig lange bis Oracle Sicherheitsaktaktualisierungen bereitstellt, um Systeme gegen laufende Angriffsversuche zu schützen.

Die häufigste Ursache für diese Schwachstellen stellen fehlerhafte Implementierungen in der Laufzeitumgebung dar. Durch das Aushebeln der Sicherheitsmechanismen der Sandbox gelang es Angreifern immer wieder, Schadcode direkt auf dem Client auszuführen. Da der Schadcode in der Regel in Form eines Java-Applets verbreitet wird, sind Clients mit dem Java Browser-Plugin in besonderem Maße betroffen.

Das BSI empfiehlt, die Ausführung von Java-Inhalten im Browser zu deaktivieren.

Wie die Sicherheitsempfehlungen umgesetzt werden, erfahren Sie in den Java-Empfehlungen auf www.bsi-fuer-buerger.de.