Australian Government
Department of Defence
Intelligence and Security

CYBER SECURITY OPERATIONS CENTRE

PROTECT

JUNE 2014

LEGAL NOTICE: THIS PUBLICATION HAS BEEN PRODUCED BY THE DEFENCE SIGNALS DIRECTORATE (DSD), ALSO KNOWN AS THE AUSTRALIAN SIGNALS DIRECTORATE (ASD). ALL REFERENCES TO ASD SHOULD BE TAKEN TO BE REFERENCES TO DSD.

Minimising the threat from Java-based intrusions

Introduction

1. Java applications are used widely to perform necessary business functions. Like most available software, there have been security vulnerabilities found in Java. Combined with its popularity, Java is particularly attractive to cyber adversaries seeking unauthorised access into corporate networks. This document is intended to assist Information Technology Security Advisers to secure Java without impeding important business functions.

What is Java?

2. Java is a software platform that is owned and supported by Oracle. The Java platform consists of the Java Virtual Machine (JVM), which is installed on the host computer, and user applications that are written using the Java programming language.

3. The JVM is powerful, flexible and easily deployed to a wide range of devices. The JVM software runs like any other program on the host. The JVM provides a bridge between Java applications and the host computer.

   a. Unofficial implementations of the JVM are available. Due to lack of support, these versions are not recommended for enterprise use.

4. Java applications give a consistent user experience independent of the underlying system. This makes Java a sound option for many software solutions used in enterprise. Unlike most user programs, Java applications require the JVM to be used, and will not run natively on the host computer. Java applications can be presented in a web browser as an applet, or launched outside of the browser as a Java Web Start application. Java applications may run in either privileged or sandbox mode.

5. Software patches from the vendor will be applied to the JVM. Updates typically address security issues or add new features. Periodically, Oracle will release a new version of the JVM and phase out support of older versions. This sort of feature-rich update to the JVM may cause compatibility issues with existing Java applications, which may stop working correctly. Agencies often avoid updating Java because of continued reliance on legacy applications.

What are the security issues?

6. Java is heavily scrutinised by the IT community for new security flaws because it is widely used and has a history of exploitable security vulnerabilities. The two categories of Java intrusion are:

   a. Exploits that target security vulnerabilities in the JVM, via drive-by browser exploitation.

   b. Malicious Java applications that run outside the sandbox as privileged applications, which may be found on websites or as email attachments.

7. Once an adversary executes malicious code using either method, the compromised system could be used to conduct activities such as stealing valuable information or gaining access to other computers on the network.

   a. The potential for harm done by an adversary in this situation can be reduced by methods covered in the Australian Signals Directorate's (ASD) *Strategies to Mitigate Targeted Cyber Intrusions*. ASD advises a multi-layered strategy to provide defence-in-depth. A multi-layered defence can protect the system from further compromise.

Exploitation of security vulnerabilities in the JVM

8. Exploitation of the JVM is mostly associated with browsing to a malicious or compromised website, but can also occur when opening an email or attachment. This type of exploitation allows an adversary to run malicious non-Java code outside of the JVM that compromises the native system. The adversary will gain the same level of access as the user, or possibly even higher.

   a. Java exploits are valued because they can grant access to a system without the knowledge or authorisation of the user.

9. The discovery of a new vulnerability by cyber adversaries could lead to exploitation before a security patch is available.

   a. Exploits targeting those vulnerabilities that have not been publically disclosed are known as zero-day attacks. Once the vulnerability is publically known, it is no longer considered to be a zero-day.

10. Security vulnerabilities are exploitable up until the time that the patch has been applied. This is a highly attractive window of opportunity for adversaries. In the time between a patch becoming available and being applied, both the number and quality of exploits will increase. For those agencies that are unable to patch quickly (usually for support of legacy applications), known vulnerabilities remain exploitable.

11. Patching is recommended as part of a defence-in-depth approach to Java security. A patch is only effective once it has been applied. Security controls can be used to minimise harm and protect IT systems until security patches can be safely deployed.

Malicious Java applications

12. Malicious Java applets are presented as trustworthy or legitimate. Adversaries may try to use targeted emails, known as spear phishing, to deliver a malicious website link or email attachment that is relevant or interesting to the user. This type of social engineering can entice a user to unknowingly permit the malicious application to run with high privileges.

13. Malicious Java applets will request permission from the user with a pop up dialogue box. If the user trusts that the applet is safe and accepts the certificate, the malicious applet can run in privileged mode. Once running with privilege, an adversary can access parts of the system that were previously protected by the sandbox, such as files and network connections.

a. By default, Java applets running in the browser that request privileged mode will request permission with a pop up dialogue box. This setting can be changed in the Java security control panel. Changing this setting to be more permissive will leave computers at greater risk, and should not be done.

b. If the user declines the certificate, a Java applet can still run in sandbox mode. A sandboxed application can still gather information that may be useful to the adversary, but will not have as much freedom to cause harm as a privileged application.

14. Java applications are run inside the JVM, and not the native operating system. Most application whitelisting implementations cannot control Java applications. Generally, the JVM is allowed to run, but there is no discrete control of Java applications that are run by the JVM, even those that are known to be good.

How can Java be used securely?

15. Agencies should gather requirements and use cases for Java. Use cases need to address which applications need to be run and the degree of trust associated with each.

a. For example, a user interface for an internal database that is developed in-house would be necessary for database access and have a high level of trust, while an application from the Internet to view video files may have both a low business use and a low trust level.

16. Business requirements can be used to determine which of the mitigation strategies below are most suitable. These mitigations are not mutually exclusive, and should be combined to implement a stronger defence-in-depth.

a. If there is no driving business need to run Java on the desktop then Java does not need to be installed. A system that cannot run Java cannot be compromised by Java. This is an absolute measure, and requires no further mitigations.

17. Recommended mitigations against Java intrusions include, but are not limited to the following:

a. using Deployment Rule Sets to whitelist Java applications

b. applying security patches

c. permitting use of Java applications from trusted sources only

d. content filtering at the gateway

e. configuring separate browsers for internal and external use

f. host based detection using Enhanced Mitigation Experience Toolkit (EMET).

Each of the above strategies is discussed below.

Using Deployment Rule Sets to whitelist Java applications

18. Application whitelisting allows system administrators to control applications and the context in which they are run. It is the number 1 strategy of the Top 4 *Strategies to Mitigate Targeted Cyber Intrusions*. Until now, it has not been possible to control Java applications without the use of a 3rd party application whitelisting product.

19. A security feature added in Oracle Java 7 Update 40 is Deployment Rule Sets, which allows administrators to whitelist Java applications based on attributes such as location, file hash or signature hash. Oracle provides comprehensive documentation on configuring Deployment Rule Sets[1].

a. Deployment rules can be used to only permit Java applications from trusted sources, with greater control than other methods discussed below. Signature hash is a powerful way to identify and verify applications from trusted sources.

20. Deployment rules allow administrators to specify the version of Java that is required to run any particular application. By default, Java applications will attempt to run in the newest and most secure version of the JVM available. Legacy applications that do not work correctly under newer versions of Java can be run in a specified older version.

a. Importantly, this maintains compatibility of vital applications without compromising security. Deployment Rule Sets are highly recommended for organisations that rely on legacy applications.

b. For example, an agency needs archived records from an old database. The database can only be accessed using a Java application that runs reliably in version 5 of Java, but not in any newer version. A deployment rule is created that runs the legacy application in the version 5 JVM, but forces all other applications to run in the latest secure version.

[1] Available at: http://docs.oracle.com/javase/7/docs/technotes/guides/jweb/security/deployment_rules.html.

Applying security patches

21. Install vendor supplied Java security patches to protect systems against exploitation of known vulnerabilities. Applying security patches is number 2 of the Top 4 *Strategies to Mitigate Targeted Cyber Intrusions.*

22. Security patches will protect systems against exploitation of known JVM vulnerabilities. The occurrence of exploitation attempts increases after a patch is released as adversaries become aware of the vulnerability and produce working exploits. Many adversaries rely on unpatched software, and can be stopped by installing security updates.

a. Java applications running with privilege can still compromise a patched system, because they rely on social engineering and do not need to exploit an unpatched vulnerability.

Permitting use of Java applications from trusted sources only

23. Allow Java applications to run only from trusted sources, such as the corporate intranet or Australian Government (gov.au) Internet domains.

24. Signed Java applications can run with a high privilege level by asking for user consent. Only trusted signing certificates should be entitled to this level of privilege.

a. Users can be enticed to run a malicious Java application if it appears harmless. An application that has been signed by an untrusted source may still appear trustworthy to the user.

25. Trusted domains can be configured in the web browser, at the gateway or using Deployment Rule Sets (mentioned above).

a. Web browser based controls work effectively when used with separate browsers (see below).

b. To control trusted domains at the gateway, consult your gateway vendor documentation. This can be implemented in conjunction with content filtering at the gateway level (see below).

c. Deployment Rule Sets are the most flexible way of configuring both trusted sources and trusted code signing certificates.

Configuring separate browsers for internal and external use

26. Different browsers can be configured for use with internal (intranet) and external (Internet) sites. The use of separate browsers is simple, but can effectively control the use of Java applications.

27. Configure the external browser to block or heavily restrict Java. This may inhibit legitimate use of Java applications from the Internet. If Java from external sites is required, then this browser should be configured to only allow Java from trusted sources (mentioned above). Untrusted websites should still be controlled or blocked.

28. The internal browser may be more permissive, as the intranet traffic is inherently more trustworthy. The browser used for intranet traffic needs to be blocked at the gateway to prevent it from being used for uncontrolled Internet websites. However, it can be freely allowed to run any Java applications that are found on the internal network.

29. Multiple browsers can each be used with a different Java version if necessary. For example, an internal use browser may need to run Java 6 for compatibility with a legacy business application, but the external browser could be updated to the latest version for security reasons.

Content filtering at the gateway

30. Delivery of Java content to users can be controlled using a proxy or web content filtering device. Such devices can be configured to refuse outgoing requests for Java content, based on URL file extension or MIME type.

a. Java file extensions include `.class`, `.jar` and `.jnlp` files.

b. Java MIME types include application/.*-java-.* and application/java-archive.

31. Restriction of trusted sources can be implemented by creating exceptions for trusted domains. For example, a proxy is configured to drop all requests for Java file extensions, unless the request is for the local intranet address range or all `.gov.au` domains, in which case the request will be allowed.

Host based detection using Enhanced Mitigation Experience Toolkit (EMET)

32. Microsoft’s Enhanced Mitigation Experience Toolkit (EMET) can help prevent intrusions by enforcing additional security measures around native programs. If EMET detects suspicious code execution, such as techniques often used in exploits, then the program will be interrupted and reported.

a. EMET is not limited to working only with Java, and may provide wider benefit by mitigating other threats, such as PDF or web browser exploits.

b. EMET adds a layer of security that will complement any of the previously mentioned mitigations.

33. In the case of Java, the security surrounding the JVM itself will be bolstered. EMET has recommended settings specific to the JVM. The aim is to detect and prevent unwanted behaviour, while allowing legitimate Java applications to work without interruption.

a. EMET settings will need to be tested with all Java applications that support necessary business functions to ensure that the user experience is not disrupted.

34. EMET 5.0 can be used in conjunction with Internet Explorer to control Java applications based on source. EMET can block the use of Java applications originating from Internet sites, while allowing Java applications from the local intranet to be used. This setting can be pushed out through group policy.

a. If the corporate browser used is Internet Explorer, this technique can achieve a similar effect to the use of separate web browsers discussed above.

Further information

35. Complementary guidance, including *Application Whitelisting Explained*, *Strategies to Mitigate Targeted Cyber Intrusions – Mitigation Details* and *Top 4 Mitigation Strategies to Protect Your ICT System*, is available on ASD's website at www.asd.gov.au.

Contact details

Australian Government customers with questions regarding this advice should contact ASD Advice and Assistance at asd.assist@defence.gov.au or by calling 1300 CYBER1 (1300 292 371).

Australian businesses or other private sector organisations seeking further information should contact CERT Australia at info@cert.gov.au or by calling 1300 172 499.