A REVIEW PAPER ON SECURITY IN JAVA

Nikhil Mittal, Sneha Kumari, Pervinder Kaur
Information Technology
Dronacharya College Of Engineering,Gurgaon
M.D University,Rohtak

Abstract- **This paper seeks out an attempt to improve the security issues and associated bugs and vulnerabilities in java. The paper basically aimsat SECURITY MANAGER, a crucial feature of java. We tend to enhance the security manager and its features like the cryptographic signed byte code in particular. Further, the paper also opts to introduce an enhanced version of SECURITY API which will provide an upgraded cryptographic algorithm, advance authentication and secure communication protocol. JVM is a different type of security in java. The paper discusses how to protect in java its use a Security.**

I. INTRODUCTION

Java provides a number of features designed to improve the security of Java applications. Java security involved in JVM,Sandboxes,API and most important is security vulnerabilities.There are four models in java security.

First is involved in the need of java security,second is the sandbox,third is the concept of trusted code and finally fine grained access control .

- The need of Java Security is used to may reduce bandwidth requirements and improve functionality of web services.
- Sandboxes is used to establish a network connection only with its originating web server .
- Trusted code is very important used in unsigned applets and applets signed by an untrusted principal were restricted to the sandbox and local applications and applets signed by a trusted principal had unrestricted access to resources.
- Lastly Fine grained access control is used to the code source consists of a URL and an optional signature and permissions granted to a code source are specified in the policy file.

(1.1) **There are three types of java security**:-

1. **The Security manager** :-

Security manager implements a check permission method which takes a permission obect as parameter, and check permission method is called from trusted system classes that requires that all system resources are accessible only via trusted system classes.

b. **Class loaders**:-

classes loaded by a class loader instance belong to the same name space,a class in one name space cannot access a class in another name space à classes from different Web sites cannot access each other establish the protection domain (set of permissions) for a loaded class.

c. **The bytecode verifier** :-

Bytecode verifier implements a *static* analysis of the bytecode and syntactic analysis all arguments to flow control instructions must cause branches to the start of a valid instruction all references to local variables must be legal all references to the constant pool must be to an entry of appropriate type.

II. SECURITY FEATURES

Java security involved in JVM,Sandboxes,API and most important is security vulnerabilities.There are four models in java security.

First is involved in the need of java security,second is the sandbox,third is the concept of trusted code and finally fine grained access control .

(2.1) Java Virtual Machine:-

This is a first feature of Java Security and very important feature.

The JVM performs verification on this bytecode before running it to prevent the program from performing unsafe operations such as branching to incorrect locations, which may contain data rather than instructions. It also allows the JVM to enforce runtime constraints such as array bounds checking.JVM is an abstract computer that can load and excute java program.java bytecode language is the actual language that JVM can interpret and execute. Java bytecode provides instructions for different kind of register and stack operations.

(2.2) Security Manager:-

This is a second feature of Security Manager**,**the security manager also allows Java programs to be cryptographically signed; users can choose to allow code with a valid digital signature from a trusted entity to run with full privileges in circumstances where it would otherwise be untrusted.

III. SECURITY VULNERABILITIES IN JAVA APPLICATIONS

Different number of possible sources of security vulnerabilities in Java applications, some of which are common to non-Java applications and some of which are specific to the Java platform.

Vulnerabilities in the sandboxing mechanism which allow untrusted bytecode to circumvent the restrictions imposed by the security manager, Vulnerabilities in the Java class library which an application relies upon for its security. A vulnerability in the Java platform will not necessarily make all Java applications vulnerable.

IV. ADVANTAGES AND DISADVANTAGES

Advantages

a. Permits controlled execution of less trusted code (vs. ActiveX)

b.Permits fine-grained permission control

c. Attention paid to security

d. Portability

e. “Instant installation”

Disadvantages

a. No standardized auditing

b. Weak against denial-of-service & nuisances

c. Security policy management too complex for endusers and weak administrative support.

d. Flexible policies accepted by users may permit hidden breaching interactions.

V. CONCLUSIONS

Security is an important aspect of applications that transport sensitive data over the Internet.The Java Platform provides a robust basis for secure systems through features such as memory-safety. However, the platform alone cannot prevent flaws being introduced. This document details many of the common pitfalls .

REFERENCES

1.*Tim Lindholm, Frank Yellin, Gilad Bracha, and Alex Buckley*.

2. *Goodin, Dan (2012-10-18). "Apple removes Java from all OS X Web browsers". Ars Technica. Retrieved 2014-02-18.*

3. *Goodin, Dan (2013-03-01). "Another Java zero-day exploit in the wild actively attacking targets*".

4. *Goodin, Dan (2013-01-14). "Microsoft releases emergency update to patch Internet Explorer bug.*