Research paper on cyber security mobile code

Law, Security

Abstract

Following the tremendous development of distributed computing and telecommunication technology, demand has risen for support of the mobile code, obtained remotely, possibly from untrusted systems for local execution. Encouragement of mobile code results in a significant count of dangerous security violations and safety issues that need to be countered. This document will discuss the concept of mobile code, the types of mobile code, security considerations and finally presents comments on mobile code.

Definition of Mobile Code

Mobile code is a kind of technology whereby a code is downloaded from remote computing device, possibly not trusted systems, but execution takes place on local machine. Mobile code can also be defined as a program that is produced by one entity called originator, and is subsequently transferred to a second entity, the host, immediately before it is executed by the host. The same concept can also be called mobile agents, downloadable code, active capsules, executable content or remote code.

Types of Mobile Code

i. Web Applets - These are java mini programs that are loaded automatically and executed on html file. A file can include many applets which can be sourced from many different servers, and executed virtually without the knowledge of the user.

ii. Dynamic Email - Dynamic emails use MIME technology and so inclusion of Safe-TCL scripts are vital components of the MIME email. It is simply email that is delivered with accompanying types of mobile code. They often assume the forms of executable files that are activated as soon as the email is opened. These scripts can either run on delivery or when the email is read by the recipient.

iii. TINA Building Blocks. This is a kind of evolving technology that includes external service providers to act as providers of Telecommunications Information Networking Architecture, which can influence network resources so as to provide enhanced services to the clients.

iv. ActiveX control - These are mobile codes designed to work as Component Object Models, which are a type of computer architectures employed by Microsoft applications. It is normally used with applications that run on browser.

v. Embedded script -These are mobile codes that are embedded within web pages. Jscript and VB Script are the most common sub-types of embedded script. They enable internet users to load video player’s off-the-net and control the general programming flow.

Security considerations

The utilization of mobile code has raised a lot of security concerns: access control, user authentication, data integrity, non-repudiation, data confidentiality and auditing.

Two security problems arise in the area of mobile code: protecting the host from malicious code and protecting the code from malicious hosts. The problem of malicious code has caught a reasonable attention because of the looming threat of computer viruses and Trojan horses.

Mobile code is prone to various security intimidations; a malevolent host may scrutinize the code, try to study the secrets carried by an agent, and take advantage of this knowledge in its interaction with the agent to gain an unfair advantage. A host may as well try to influence the outcome of the computation.

Protecting the mobile code is possible with application of encryption technology. The mobile code is executed in an encrypted form on an untrusted host environment.

Resource access and safety

For mobile code to execute and carry out its predefined task, it has to access system resources. Mobile Code has to be granted limited access to the system resources for safety reasons. The varieties of resources that the mobile code requires for execution are: file system, random memory, network, output devices, input devices, process control, system calls and user environment.

Language support for safety

When considering means of providing safe execution of the mobile code, if the mechanism of protecting enlarged portion of address space is not used, then there is going to be much dependence on the type-safe languages. These guarantees inbound stay of arrays, valid pointing and that code cannot violate variable typing.

Granting access

The key issue in provision of safe environment for execution of the mobile code is establishing precisely which resource a specific unit of code needs to be given rights of access. There is need for a security policy which determines the type of access for every mobile code.

Protecting the host from malicious mobile code

1. Sandboxing – this is the idea whereby the host confines the visiting mobile code to certain execution environment. The host permits the mobile code access to specific resources and prevents others. The sandbox can be tailored to different sizes to suit the requirements of different programs.

2. Digital shrink-wrap – this is whereby the mobile code is authenticated before it is permitted to execute. Here the producer of the code is required to sign it. The consumer on the other end has to validate the signature before using it. It is however hard to determine whether a given mobile code contains a malicious code but one is at least capable of knowing whether it is authentically coming from its claimed source.

3. Proof-carrying code – this technique enables the host to determine, automatically and with certainty that a program code provided by another system is can install and execute safely. Basically, the code producer is required to provide an encoding of a proof that his/her code conforms to the security policy set by the code consumer. The proof is encoded for digital transmission.

4. Firewalling – this approach involves securing mobile code by selectively choosing to run a program at the very point it enters the client domain. If an organization is running a firewall or web proxy, it may be handy to try to identify java applets, observe them, and decide whether or not to serve them to the client.

Protecting the mobile code from malicious host

1. Limited Blackbox Security - The major idea in blackbox security is the generation of an executable code from a given agent specification. This kind of generated code is executed as a blackbox by the host. This means that the host is not capable of modifying or reading it but it only executes the code the way it is. It is however important to note that blackbox security does not protect against every possible attack. It is still possible for the host to refute the execution or to echo erroneous system call results. It is also probable for the attacker to read and influence the code content but since role of the elements of the application cannot be determined, the attack outcomes are random.

2. Computing with encrypted functions – this is whereby a program is executed in a cipher program form instead of plaintext. It involves use of polynomial functions as a general solution for the security requirements of mobile code.

3. Cryptographic Traces – the mechanism is based on post-mortem analysis of data (traces) that are collected during the execution of the mobile code. The traces are then used as a basis for execution verification. However, there are limitations of the same. First it allows detection of an attack after

the execution therefore needs a different mechanism for timely detection. Secondly, it is a detection technique and so a mechanism of punishing cheating sites is principals must be devised.

Comments: Mobile Code friend or foe?

Mobile code is experiencing rising demand due to growth of distributed systems. Safe execution of mobile code implies a need for controlled access to resources, access which should ideally be negotiated for each mobile code unit.

References

Howard, R., Graham, J., & Olson, R. (2010). Cyber security Essentials. United Kingdom: Edward Edgar Publishing Limited.

Probst, C. W., Hunker, J., & Gollmann, D. (2010). Insider Threats in Cyber Security. Australia: John Wiley & Sons.

Singhal, A. (2007). Data warehousing and data mining techniques for cyber security. Australia: Macmillan.