MULTI-LAYER PROTECTION OF MOBILE CODE[†]

Chenghui Luo, Ph.D.
Fraunhofer Center for Research in Computer Graphics, Inc.
321 South Main St.
Providence, RI 02903
Email: cluo@crcg.edu

ABSTRACT

Mobile code is computer code that roams and executes remotely on a computer network. While it has advantages such as autonomous computing, mobile code is very hard to protect against malicious host attacks. In this paper, a multi-layer protection framework of mobile code, including complete obfuscation, encrypted execution and code watermarking, is presented. With security strengthened, mobile code is ready for large-scale deployment on open distributed computer networks.

1. INTRODUCTION

Mobile code is computer code that roams and executes remotely on a computer network. Supporting exchange of executable computer code, the mobile code paradigm differs from the classic client-server model where a client and server exchange messages. Autonomous mobile code, called mobile agent, is a computing object that knows what and how to perform a predefined task independently. This autonomous and remote execution nature makes mobile code (agent) an efficient means to deliver dynamic actions instead of only static data. In a military computing environment, mobile code enables more agile and flexible deliver of digital actions from a command center to field troops and it also greatly simplifies a military unit's information management and increases its survivability.

Mobile code is an increasingly prevalent trend for distributed computing in a large computer network environment. However, mobile code is vulnerable to security problems [Gong, 1997], including malicious agents and malicious hosts. A malicious agent is an agent that performs harmful actions on a remote host like a computer virus. Possible damaging actions include unauthorized access, modification and overuse of local resources, such as sensitive data, system calls, and CPU time. A malicious host is a host computer that performs harmful actions on a mobile agent, including spying out and manipulating agent code, data, and control flow; listening to and tampering with data exchange between an agent and agent owner; executing code incorrectly; and denying execution and masquerading as another host.

In addition to malicious host attacks, agent reverse engineering [Cifuentes, 1994], the process of decompilation of mobile agent binary code to obtain its source code, is also a great threat for mobile agents. Currently most mobile agent applications are developed with the Java programming language because of Java's strong security and mobility features, but it is rather easy to reverse engineer Java bytecode and get back the original, human-readable source code, then the source code can be illicitly modified, secret data can be revealed, and intellectual property can be stolen. All of these attacks on mobile code become much easier based on the analysis of a mobile agent's source code.

To deploy mobile code's advantages in a military application, mobile code security problems must be solved. At national security level, long before the September 11th terrorist attacks, network attacks such as distributed denial of service (DDoS) and e-mail viruses exploited the vulnerabilities of the nation's IT infrastructure. Clearly, mobile code, as part of the nation's information infrastructure, must be well protected to secure the nation's digital homeland.

At Fraunhofer Center for Research in Computer Graphics (CRCG), we have developed a multi-layer mobile code protection framework called *Information Armor*™ (to solve the malicious host problem, which is much harder than the malicious agent problem. The latter can be basically solved by various access control techniques). Including complete obfuscation, encrypted execution and code watermarking technologies, Information Armor™ (Figure 1) provides complementary mutual supporting protections of mobile code. This technology is the result of research projects funded by the Air Force Research Laboratory (AFRL), as well as internally-funded research at Fraunhofer CRCG. We have focused initially on the protection of Java-based agents because Java-based mobile code is so widespread in wired and wireless networks. What we have learned about protecting Java code at Fraunhofer CRCG, however, can be applied to other programming languages and platforms. In the rest of this paper, we summarize the three protection layers and finally present our conclusion and future work.

[†] This project is funded by the Air Force Research Laboratory (AFRL) under contract F30602-01-2-0505.

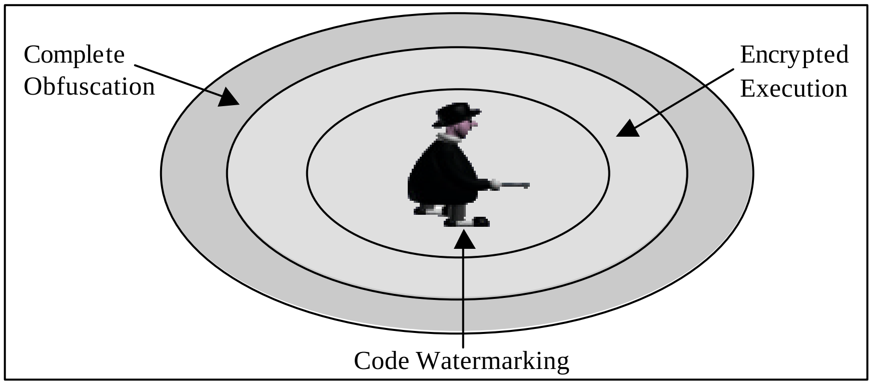

Figure 1. Information Armor™: a multi-layer mobile code protection framework.

2. PROTECTION FRAMEWORK

Complete Obfuscation

Code obfuscation is a technology to hide or remove symbolic information in mobile code to protect mobile code's source code even if a program is decompiled. Because of the vulnerability of Java bytecode under the reverse-engineering attack, we protect mobile code up to the *Java system class level*. There is a number of off-the-shelf Java obfuscators and decompilers that work on the application portion of the code. However, a malicious host can learn much about agent source code by observing calls to the Java system classes. Only the Fraunhofer CRCG technology is capable of obfuscating Java system classes. This technology provides mobile code *complete obfuscation* from application classes to system classes, so it offers much stronger protection than traditional obfuscation approaches. Mobile code so protected is very difficult to reverse-engineer and manipulate. This technique can also be used to protect intellectual property rights embedded in Java-based mobile agents.

Encrypted Execution

In our mobile code protection framework, Java bytecode is not only completely obfuscated but also *encrypted* as well, so that a malicious host can't understand the agent bytecode. With CRCG Information Armor™, mobile agents *never* execute in-the-clear on a remote host, and moreover, we have developed a technique called *class evolution*, that enables the encryption to change throughout the execution cycle. With our customized mobile code class loader, Java system classes remain obfuscated and encrypted at run time, and our encrypted execution mechanism also supports secure execution of watermarked mobile code.

Code Watermarking

Code watermarking is a technology used to embed a secure and invisible label in mobile code. Watermarks embedded in mobile code can help authenticate a host *environment* in which a mobile code executes, as well as authenticating the mobile code itself. The robustness of a watermark measures the difficulty of removing the watermark from mobile code. A non-robust watermark is called a "fragile watermark", and it is this fragility that helps to authenticate mobile code. Mobile agents can be watermarked before they are dispatched to a network. Each time an agent executes on a remote host, it checks to see if its watermarks have been removed or altered. If so, the agent knows it has been tampered with and can report back to the host that dispatched it. If it never reports back, that also tells the dispatcher there has been an attack.

CONCLUSION AND FUTURE WORK

No cyber-protection mechanism can be 100 percent secure. But just as many layers of armor reinforce each other, Fraunhofer CRCG Information Armor™ provides multiple layers of protection that work together to make mobile code as secure as possible. Moreover, the Fraunhofer CRCG Information Armor™ is *unique to each agent*, which means that even if a malicious host can crack one mobile agent, that information is completely useless for cracking any other agent! In addition, Fraunhofer CRCG Information Armor™ enables *intrusion detection* for mobile agents. The armor not only protects an agent, it can provide a warning to the agent that it has been tampered with.

Equipped with our complete obfuscation, encrypted execution and code watermarking armors, mobile code, once risky and dangerous in an insecure malicious host environment, is now much more secure. Working from this foundation, we will continue our efforts to secure the IT infrastructure, and we will begin to develop large-scale mobile agent-based applications, such as digital weapon maneuver, network management systems and web services.

REFRENCES

Gong, L., "Survivable Mobile Code is Hard to Build", *Proc. of the DARPA Workshop on Foundations for Secure Mobile Code*, pp. 26 – 28, March 1997.

Cifuentes, C., "Reverse Compilation Techniques", *PhD Dissertation*, Queensland University of Technology, Department of Computing Science, 1994.