이동코드 보안 모델에 관한 연구

신정화[†] , 신원[‡] , 이경현[‡‡]
† 부경대학교 전산정보학과
‡ 부경대학교 전자계산학과
‡‡ 부경대학교 컴퓨터멀티미디어공학전공

A Study on a Security Model for Mobile Code

Jung-Hwa Shin[†] , Weon Shin[‡] , Kyung-Hyune Rhee[‡‡]
† Department of Computer and Information Science, PKNU
‡ Department of Computer Science, PKNU
‡‡ Department of Computer and Multimedia Engineering, PKNU

요 약

본 논문에서는 웹의 발달로 인해 정적이고 이미지 위주의 정보를 제공해주던 기존의 단순한 웹 환경에서 벗어나 동적이고 멀티미디어 형태의 정보 제공이 가능한 Java와 ActiveX 기술을 이용한 이동코드에 대하여 살펴보고, 이동코드 사용시 발생 가능한 취약성과 그에 따른 보안 모델을 분석한다. 또한 기존의 보안 모델을 보완한 새로운 보안 모델을 제안한다.

1. 서론

현재의 인터넷 환경을 대중화하는데 결정적인 역할을 한 월드와이드웹(World Wide Web)은 텍스트로 제한된 인터넷 환경에 하이퍼미디어(HyperMedia)가 지원되는 통합형 인터넷 서비스를 제공하여 사용자들에게 편리한 인터넷 작업을 할 수 있는 환경을 만들어 주었다[8]. 인터넷 사용 인구 증가와 사용 용도의 다양화로 인해 정적이고 수동적인 정보를 제공하는 기존 웹 환경의 단순함에서 벗어나 동적이고 멀티미디어 형태의 정보 제공이 가능한 웹 환경을 요구하게 되면서 Java와 ActiveX 기술을 이용한 이동코드(Mobile Code)가 등장하게 되었다. 이식성, 보안성, 분산 환경, 객체지향성 등을 제공하는 Java는 인터넷 쇼핑몰, 실시간 인터넷 게임, 인터넷 주식, 애니메이션 구현에 사용 가능[9]하고 동적인 컨텐트를 만들 수 있도록 통합된 플랫폼을 제공하는 ActiveX는 기존의 소프트웨어, 애플리케이션을 이용하여 애니메이션과 3차원 가상 현실 및 동영상을 실시간으로 보여줄 수 있다[8]. 이러한 이동코드의 등장으로 정적인 문서와 이미지 위주의 웹 페이지는 더욱 세련된 멀티미디어 환경으로 확장될 수 있고 웹 서버와 클라이언트간의 상호 작용성도 높아졌다. 이러한 편리함을 가지는 이동코드가 악의적인 목적으로 작성된다면 사용자 시스템을 파괴하거나 정보 유출 등의 피해를 줄 수 있다[10]. 따라서, 본 논문에서는 이동코드 사용에 따른 취약성 및 보안 모델 분석을 통해 새로운 보안 모델을 제안함을 목표로 한다. 본 논문의 구성은 다음과 같다. 2장에서 이동코드와 동작방식에 대해서 살펴보고, 3장에서 이동코드 사용시 발생 가능한 취약성을 분석해 본다. 그리고, 4장에서는 취약성 해결을 위한 보안 모델에 대하여 살펴보고, 5장에서는 기존의 보안 모델을 보완한 새로운 보안 모델을 제안한다.

2. 이동코드의 정의와 동작방식

2.1 이동코드의 정의

이동코드(Mobile Code)는 네트워크를 통해 원격지로 전송되어 실행되는 프로그램으로 웹 브라우저를 통하여 쉽게 수행 가능하고 누구나 작성할 수 있고,

브라우저를 수행할 수 있는 어떤 컴퓨터에서도 동작 가능하다[1][2]. 즉, 운영체제나 하드웨어에 관계없이 어떤 플랫폼에서도 동일 코드가 수행될 수 있다. 또, 이동코드는 서버에서 다운로드된 후에는 서버의 자원을 소모하지 않고 클라이언트의 브라우저 환경 내에서 실행 가능하다. 현재 이동코드로 Java 애플릿, ActiveX 컨트롤, JavaScript, VBScript가 대표적으로 사용되고 있지만 본 논문에서는 script로 해석되는 형태가 아닌 독립적인 실행환경을 가지는 Java 애플릿과 ActiveX 컨트롤로 이동코드의 범위를 제한한다.

2.2 이동코드 동작방식

웹 페이지에서 다운로드 받을 수 있는 이동코드의 동작방식은 다음과 같다[8].

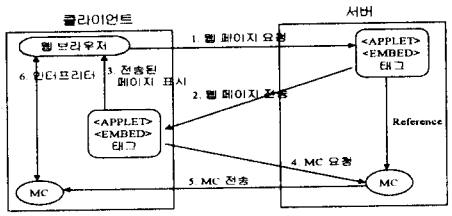

그림 1 이동코드 동작방식

(1) 클라이언트는 서버 쪽으로 정보를 가지고 있는 웹 페이지를 요청한다
(2) 서버는 클라이언트가 요청한 웹 페이지를 전송시켜 준다.
(3) 전송된 웹 페이지는 클라이언트의 브라우저를 사용하여 내용을 표시한다.
(4) 클라이언트는 전송된 웹 페이지에서 Java 애플릿인 경우 <APPLET> 태그를 통하고, AcitveX인 경우 <EMBED> 태그를 통하여 서버로 이동코드를 요청한다.
(5) 요청된 이동코드가 클라이언트로 전송된다.
(6) 전송된 이동코드는 클라이언트의 해석기에 의해 해석되어 특정한 동작을 수행한다.

3. 이동코드 사용에 대한 취약성 분석

이동코드는 프로그램을 인터넷상에서 동적으로 다운로드 받아 로컬 프로그램의 형태로 다양한 플랫폼에서 실행이 가능하므로 악의적인 목적으로 작성된 이동코드를 다운로드 받거나 실행하는 경우 사용자 시스템에 다양한 공격이 가능하다[11]. 이러한 공격은 사용자 시스템의 정보 유출에 대한 "비밀성 침해 공격", 사용자 시스템의 정보 변경에 대한 "무결성 침해 공격", 정상적인 작업을 방해하는 "가용성 침해 공격"으로 나눌 수 있다[10].

3.1 Java 사용시 취약성

네트워크 상에서 다운로드 되어 자동적으로 실행되고 다양한 플랫폼에서 동작 가능한 Java 애플릿을 다운로드 받아 실행하는 경우 클라이언트의 하드디스크를 삭제한다거나 클라이언트에 연결된 내부 네트워크를 위험에 빠뜨릴 수 있다[1][3][5][9].

(1) 비밀성 침해 공격은 불법으로 작성된 애플릿을 다운 로드 받는 경우 사용자 시스템 자원을 불법적으로 획득하고 사용자 개인 정보나 시스템 정보를 유출하는 공격이다.
(2) 무결성 침해 공격은 침입코드를 애플릿에 삽입하여 사용자가 애플릿을 다운로드 하는 경우 제3자의 시스템을 공격하여 사용자 시스템 자원을 불법적으로 수정, 변경, 삭제하는 공격이다.
(3) 가용성 침해 공격은 애플릿이 동작될 때 시스템의 프로세서 시간을 소모하도록 기다리게 하거나, 시스템이 정지할 때까지 메모리를 할당하여 사용자의 정상적인 작업을 방해하는 공격이다.

3.2 ActiveX 사용시 취약성

Java와 마찬가지로 자동적으로 다운로드 하여 설치나 실행이 가능하므로 컨트롤을 악의적으로 작성해서 배포할 경우 사용자 시스템에 여러 가지 문제를 일으킬 수 있다[1][4][5]. ActiveX의 경우 컨트롤에 대한 실행 권한이 한번 설정되면 시스템에 대한 모든 권한을 가지므로 자바에서 보다 더 큰 시스템 피해가 발생할 수 있다.

(1) 비밀성 침해 공격은 다운로드 받은 ActiveX 컨트롤에 서명이 되어 있다면 어디서나 사용이 가능하므로, 공격자가 서명된 컨트롤의 CLSID(식별자)와 파라메터를 알아내어 사용자 정보를 유출하는 공격이다.
(2) 무결성 침해 공격은 컨트롤에 대하여 서명 여부만 검사하고 설치일은 검사하지 않기 때문에 공격자가 만든 컨트롤의 서명일을 위조하여 사용자 시스템에 계속 남아 있으면서 부당한 동작을 수행할 수 있다.
(3) 가용성 침해 공격은 사용자가 이미 서명을 받아서 동작하고 있는 컨트롤을 다운로드 받아 사용

하는 도중 문제가 발생한 경우 서명을 취소할 수 있는 방법이 제공되지 않기 때문에 해당 컨트롤을 실행하는 경우 동일한 문제가 계속 발생할 수 있다.

3.3 공통으로 적용되는 취약성

Java와 ActiveX에 공통으로 적용되는 취약성에 대한 것으로 주로 서비스 거부 공격이 이에 해당된다[11].

(1) 무결성 침해 공격은 안전하지 않은 인터넷 환경에서 Java, ActiveX 컨트롤 동작에 대한 안전성 문제가 있다.

(2) 서비스 거부 공격은 특정 사이트 방문시 동작하는 이동코드에 트로이 목마를 삽입하거나 윈도우를 계속 생성하여 이동코드 수행을 방해하거나 불쾌감을 유발하는 화면 및 사운드 생성 기능을 추가하여 사용자를 번거롭게 한다.

4. 이동코드 보안 모델

Java와 ActiveX 기술을 이용한 이동코드 사용시 발생할 수 있는 취약성을 막기 위한 기존의 보안 모델에 대해서 고찰한다.

4.1 Java 보안 모델

Java에서 보안은 애플릿을 로컬 시스템으로 들여오는 Java 브라우저 응용 프로그램으로부터 Java 언어와 Java 가상 기계에 이르기까지 내재된 여러 가지 보안 메커니즘의 제공으로 이루어지고 있다. Java에 제공하는 보안 메커니즘으로 3가지 요소가 있다[3][11].

(1) 클래스로더(Class Loader) : 외부에서 다운로드된 코드와 로컬 코드에 서로 다른 권한을 부여하여 실행될 수 있도록 이름 영역을 할당하는 작업을 수행한다.

(2) 바이트코드 검사기(ByteCode Verifier) : Java 가상 머신에 전달된 바이트코드의 정당성, 스택 오버플로우, 데이터 타입의 무단 변경 등을 검사한다.

(3) 보안 관리기(Security Manager) : 다운로드 받은 코드가 주어진 이름 영역 내에서만 활동할 수 있도록 감시하며 중요한 시스템에 대한 접근 권한을 통제하는 역할을 수행한다.

이러한 세 가지 메커니즘을 적용한 Java 보안 구조가 JDK(Java Development Kit) 버전별로 발전되어 오고 있다[3][7][9][12].

1. JDK 1.0 - 샌드박스(sandbox) 모델
   시스템 자원에 대한 접근을 제한하기 위해 JVM(Java Virtual Machine)내에 샌드박스를 만들어 Java 실행 시에 원격지에서 다운로드된 애플릿은 모두 신뢰성이 없는 애플릿으로 간주하여 샌드박스 내에 있는 보호 정책에 따라 제한된 권한으로 동작하도록 하고 로컬 시스템에 저장된 애플릿은 모두 신뢰성이 있는 것으로 간주하여 모든 권한을 가지고 동작한다.
2. JDK 1.1 - 전자 서명(signed applet) 모델
   애플릿의 서명 여부에 따라 신뢰성을 부여하여 서명된 애플릿은 로컬 코드와 동일하게 동작하며 아무 제한 없이 사용자 시스템의 여러 가지 자원에 접근 가능하고 그렇지 않은 경우 JDK 1.0과 같이 샌드박스 내에서만 동작한다.
3. JDK 1.2 - fine-grained 접근 통제 모델
   현재 설정된 보안 정책을 기초로 애플릿이 로드될 때 각 애플릿에 허가권을 할당하고 애플릿은 할당된 허가권에 따라 사용자 자원에 접근이 가능한 모델이다.

위와 같이 발전되어온 보안 모델은 네트워크 기반의 어플리케이션 제작에 활기를 불어넣었고 응용 프로그램 개발자가 접근 권한의 미세한 부분까지 제어할 수 있도록 해준다.

4.2 ActiveX 보안 모델

Java와 같이 샌드박스를 가지지 않기 때문에 인증서를 기반으로 동작을 하고 실제 실행은 사용자 판단에 의존한다. 현재까지 알려진 관련 보안 모델은 다음과 같다[2][5][6].

1. 인터넷을 통한 소프트웨어 배포와 관련하여 신뢰성 있고 안전한 방법의 하나로 인증에 기반한 방법을 이용한다. 다운로드 받은 컨트롤에 대하여 인증서가 유효한지 여부와 컨트롤 작성자의 ID와 인증서가 일치하는지, 유효기간이 끝나지 않았는지를 확인한 후 동작하도록 한다.
2. Internet Explorer에서 다운로드 중인 프로그램과 파일 관리 방법을 4가지 영역(인터넷, 인트라넷, 신뢰된 사이트, 제한된 사이트)으로 나누어 영역별로 보안 수준 설정을 다르게 할 수 있다.

5. 이동코드를 위한 새로운 보안 모델 제안

본 장에서는 3장과 4장의 이동코드 보안 모델에 대한 취약성 분석을 통해 보다 안전한 보안 모델을 제안한다. 제안된 기법은 신뢰 기반으로 이동코드를 동작시키는 "수동적인 이동코드 보안 모델"과 신뢰되지 않은 이동코드를 동작시킬 경우 사용자의 클라이언트를 보호하기 위한 "능동적인 이동코드 보안 모델"이다.

5.1 수동적인 이동코드 보안 모델

클라이언트와 서버 사이에 FMC(Filter for Mobile Code)를 두고 MFC를 이용하여 신뢰되지 않은 애플릿의 생성을 막고 사전에 수행 여부를 결정할 수 있도록 하는 모델로 동작 방식은 다음과 같다

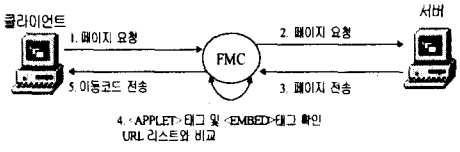

그림 2 FMC(Filter for Mobile Code) 동작방식

① 이동코드를 다운로드 받기 위해 클라이언트는 FMC로 이동코드를 가지고 있는 웹 페이지를 요청한다.
② FMC는 클라이언트의 요청을 서버로 보낸다.
③ 서버는 FMC가 요청한 웹 페이지를 FMC로 전송해 준다.
④ FMC는 전송 받은 웹 페이지에 Java 애플릿인 경우 <APPLET> 태그, ActiveX인 경우 <EMBED> 태그가 있는 지 확인한 후 태그가 포함되어 있다면 제거하고 전송 받은 웹 페이지의 URL이 미리 작성해둔 신뢰 리스트에 있는 URL인지 비교한다.
⑤ 태그 제거 후 나머지 부분을 클라이언트로 전송하여 실행한다.

5.2 능동적인 이동코드 보안 모델

클라이언트와 서버 사이에 PMC(Proxy for Mobile Code)를 두고 다운로드 받은 이동코드의 수행 여부는 PMC 내에서 처리하고 I/O 결과만 사용자에게 전송해 주는 모델이다. 동작방식은 다음과 같다.

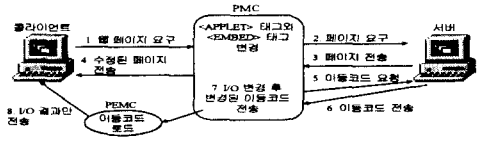

그림 3 PMC(Proxy fo Mobile Code)의 동작방식

① 클라이언트는 PMC에게 이동코드가 있는 웹 페이지를 요청한다.
② PMC는 클라이언트의 요청을 서버로 보낸다.
③ 서버는 PMC에게 클라이언트가 요청한 웹 페이지를 전송한다.
④ PMC는 전송 받은 웹 페이지에서 Java 애플릿인 경우 <APPLET> 태그, ActiveX인 경우 <EMBED> 태그를 확인하여 클라이언트에 저장된 신뢰된 이동코드 환경을 이용하여 전송 받은 웹 페이지에 있는 이동코드를 수정한 후 이동코드를 포함한 웹 페이지를 클라이언트로 전송한다.
⑤ PMC는 서버에게 다운로드 할 이동코드를 요청한다.
⑥ PMC는 이동코드를 전송받은 PMC내에서 I/O를 위해 이동코드를 변경한다.
⑦ PMC는 변경된 이동코드를 PEMC(Play Envrionment for Mobile Code)로 전송한다.
⑧ 전송 받은 이동코드의 실행은 PEMC에서 이루어지고 I/O 결과만 클라이언트로 전송된다.

6 결론

본 논문에서는 웹 환경의 발전과 더불어 사용 빈도가 증가하고 있는 이동코드의 개념과 동작방식, 사용시 발생 가능한 취약성, 보안 모델에 대하여 살펴보았다. 이동코드를 이용한 기술은 전자 상거래와 전자출판에서 실시간 정보 전송과 멀티미디어 환경 제공, 데이터 베이스 검색 서비스, 그래픽 애니메이션 서비스 제공 등 다양한 분야에서 활용될 수 있다. 따라서, 본 논문에서는 안전한 이동코드 환경을 위하여 새로운 보안 모델을 제안하였고 본 논문에서 제안한 모델을 적용할 경우 기존의 침해 방식에 따르는 취약성을 대부분 해결할 수 있지만 추가적으로 발생하는 부분에 대해서는 본 분야와 관련하여 지속적인 연구가 필요하다.

[참고문헌]

[1] Sergio Loureiro, Refik Molva, Yves Roudier,

"Mobile Code Seucurity"
[2] Camille LeBlanc, Hialing Jiang, Aylin Kuntay, Kevin Gerson, "Mobile Code: Java v. ActiveX", UC Berkeley, 1997
[3] Gary McGraw, Edware W.Felter, "Securing Java: Getting Down to Business with Mobile Code", 1999
[4] David Hopwood, "A Comparison between Java and ActiveX Security", zetnet, 1997
[5] "An objective survey of the security risks associated with ActiveX and Downloadable and Executable Content", 1997
[6] Jalal Foghhi, Jalil Feghni, Peter Williansm, "Digital Certificates:Applied Internet Security", 1999
[7] Marco Pistoia, Duane F.Reller, Deepak Gupta, Milind Nagnur, Ashok K.Ramani, "Java 2 Network Security", 1999
[8] 이정훈, "웹 페이지 Executable Contents 현황", 한국정보보호센터, 1999
[9] 신정화, 이경현, "전자상거래 환경에서의 Java 모델에 대한 고찰", 한국멀티미디어학회, Vol.2, No.2, pp.64-69, 1999
[10] 이정효, "Java 애플릿 및 Java 스크립트를 이용한 공격", 한국정보보호센터
[11] 한국정보보호센터, 자바보안기술분석서, 1997
[12] 이완석, 김홍근, "Java 보안 모델", 정보과학회지, Vol.6, No.4, pp.29-35, 1998
[13] Dirk Balfanz, Edward W.Felten, "A Java Filter", Princeton University
[14] David M.Martin Jr, Sivaramakrishnan Rajagopalan, "Blocking Java Applets at the Firewall", Boston University, Bellcore, 1997
[15] Li Gong, "New Security Architectures Directions for Java", JavaSoft, Cupertino, Californiam, 1996