홈네트워크를 위한 Jini 2.0 Security 분석

이윤경, 한종욱, 김도우, 주홍일
한국전자통신연구원

Analysis of Jini 2.0 Security for Home Network

Yun-kyung Lee, Jong-wook Han, Do-woo Kim, Hong-il Ju
KoreaElectronics and Telecommunications Research Institute
E-mail : neohappy@etri.re.kr

요 약

Jini는 선 마이크로시스템즈에서 제안한 미들웨어 이다. Jini는 각종 디바이스나 소프트웨어들이 하나의 큰 시스템에 연결된 것처럼 서로 정보를 주고 받고, 공유하고, 제어할 수 있는 동적인 분산 연합 시스템의 구축을 목표로 한다. Jini는 UPnP, HAVi와 함께 홈 네트워크의 주요 미들웨어로 꼽힌다.

홈 네트워크에서 프라이버시 보호, 방범, 등의 보안은 아주 중요하며 이를 위하여 Jini에서도 시큐리티를 보강하여 2.0 버전을 2003년 11월에 발표하였다. 따라서 본 논문에서는 Jini 2.0의 시큐리티 분석 및 이를 홈 네트워크에서의 보안과 연관지어 기술하고자 한다.

ABSTRACT

Jini is a middle ware supposed in Sun Microsystems. The goal of Jini is the establishment of dynamic distributed system, which can share information and control of other Jini technology-enabled services or devices in the same Jini system. Jini is one of the best middleware together UPnP and HAVi.

Home network security, such as privacy protection, crime prevention, and etc, is very important. So Jini 2.0 adds security mechanism in 11. 2003. This paper describes the analysis of Jini 2.0 security mechanism, and home network security.

키워드
Jini, Jini security, Home network security

Ⅰ. 서 론

Jini는 1999년 선 마이크로시스템 사가 제안한 차세대 접속기술이다. Jini는 어떤 인위적인 조작이나 설치 없이 각종 디바이스들과 서비스들이 마치 하나의 시스템 내에 있는 것처럼 서로 통신하고, 리소스를 공유할 수 있다. Jini 시스템은 자바 프로그래밍 언어를 기반으로 생겨난 기술이기 때문에 자바 가상머신(Java Virtual Machine)이 있는 디바이스 또는 자바 컴파일러가 읽을 수 있는 바이트코드 형태로 된 어플리케이션들이 Jini 시스템에 참여할 수 있다. 그리고 이러한 디바이스 또는 서비스들은 사용자가 보기에 Jini 시스템에 스스로 설치되고, 연합체를 조직하며, 스스로 설정하고, 스스로 진단하는 것처럼 느껴진다. 이러한 Jini 기술은 홈 네트워크에서 미들웨어로 사용될 수 있다. 그래서 홈 내의 각종 디바이스들을 하나의 네트워크로 묶어서 서로 상호 작용하도록 할 수 있다.

Jini는 마이크로소프트사의 UPnP(Universal Plug and Play)와 소니를 중심으로 한 가전 업체들이 모여서 만든 HAVi와 함께 홈 네트워크에서 중요한 미들웨어로 손꼽히고 있다. UPnP는 IP기반의 네트워크가 지원되는 모든 기기들을 대상으로 하고 있고, HAVi는 IEEE 1394를 이용하여 AV기기들을 대상으로 하고 있다. 반면, Jini는 네트워크 통신이 가능하고 Java VM을 내장하고 있거나 Java 바이트코드 형태로 지원되는 디바이스 혹은 서비스를 대상으로 하고 있다.

본 논문에서는 Jini 시스템과 Jini 서비스의 중심이 되는 "discovery", "join", "lookup", "leasing" 프로토콜 및 Jini의 시큐리티에 관하여 기술할 것이다. 또한 Jini 2.0에서 추가된 시큐리티도 함께 기술

할 것이다.

II. Jini 시스템 개요

Jini 시스템의 목적은 디바이스들과 소프트웨어 컴퍼넌트들의 그룹들을 한 개의, 동적 분산 시스템 연합체로 구성하는 것이다. 그리고 Jini 시스템에서 가장 중요한 개념은 서비스이다. 서비스는 사람, 프로그램 또는 또 다른 서비스에 의해서 사용될 수 있는 엔터티(entity)를 뜻하는데, 그 예로는 저장공간, 또 다른 사용자와의 통신 채널, 소프트웨어 필터, 하드웨어 디바이스, 또는 다른 사용자가 있다.

Jini 시스템의 멤버들은 서비스에 대한 억세스를 공유하기 위해서 연합한다. Jini 시스템은 클라이언트와 서버들, 사용자와 프로그램들, 또는 프로그램들과 파일들의 집합이 아니라 특정 업무를 수행하기 위해서 함께 소집될 수 있는 서비스들로 구성되어 있다. 서비스들은 또 다른 서비스를 사용할 수 있고, 한 서비스의 클라이언트는 그 자체가 자신의 클라이언트를 가진 서비스가 될 수 있다. Jini 시스템은 서비스 구성, 룩업, 통신, 그리고 분산 시스템의 사용 등에 관한 메커니즘을 제공한다.

Jini 시스템에서 서비스들은 룩업 서비스에 의해서 발견되고 분해되고, 룩업 서비스는 시스템과 시스템 사용자 사이의 접촉의 중요 포인트를 제공한다.

서비스는 "discovery/join" 이라는 프로토콜의 쌍에 의해서 룩업 서비스에 추가된다. 먼저 서비스가 "discovey"프로토콜을 사용하여 적절한 룩업 서비스의 위치를 찾으면, 그 서비스는 "join" 프로토콜을 이용하여 룩업 서비스에 등록된다.

"discovery" 프로토콜을 좀 더 살펴보자. 서비스 제공자가 자신을 알리기 위해서 로컬 네트워크내의 임의의 룩업 서비스에게 리퀘스트를 멀티캐스트 한다. 리퀘스트 패킷을 받은 룩업 서비스들은 그 리퀘스트에 응답할지 여부를 결정하고, 유니캐스트로 응답 메시지를 보내거나 보내지 않는다. 룩업 서비스가 리퀘스트에 응답할지 여부를 결정하는 기준이 되는 항목에는 다음과 같다.

- 리퀘스트를 요청한 서비스가 원하는 그룹에 룩업서비스가 속해 있을것.
- 리퀘스트를 요청한 서비스가 이미 알고있는 룩업 서비스가 아닐것.

위 두 가지 조건을 모두 만족할 때 룩업 서비스는 리퀘스트에 응답을 한다. 룩업 서비스는 리퀘스트 메시지를 통해서 리퀘스트 메시지를 보낸 서비스의 위치를 알 수 있기 때문에 멀티캐스트가 아닌 유니캐스트의 방법으로 자신의 위치 정보를 알리는 응답(unicast discovery packet)을 전송한다. 룩업 서비스로부터 응답을 받아 룩업서비스의 위치를 알게 된 서비스는 자신을 룩업 서비스에 등록한다. 이 과정에서 서비스의 proxy가 룩업 서비스에 전송된다. 이로써 "join"과정이 끝나게 된다.

Jini 시스템 내에서 서비스를 사용하고자 하는 클라이언트는 원하는 서비스의 위치를 알기 위해서 시스템의 룩업 서비스에 서비스를 요청한다 ("lookup"). 룩업 서비스에 원하는 서비스 목록이 있다면 룩업 서비스에 저장된 서비스 proxy를 카피해온다. 원하는 서비스의 proxy의 카피를 가진 서비스는 원하는 서비스와 직접 RMI 통신을 함으로써 원하는 서비스를 받을 수 있게 된다.

Jini 시스템에서 서비스들에 대한 억세스는 임대(lease)를 기반으로 이루어진다. 즉, 특정 시간동안 특정 서비스를 이용할 수 있다. 그리고 그 서비스를 이용하는 시간은 서비스 제공자와 서비스 요청자 사이의 협상으로 결정이 된다. 서비스 이용을 요구한 기간이 끝나기 전에 서비스 사용자가 "leasing"의 갱신을 요구하지 않거나, 서비스 제공자가 갱신을 거절하면 "lease"는 끝나게 되고, 더 이상 서비스를 사용할 수 없게된다.

III. Jini security와 홈네트워크의 보안

Jini 시스템의 보안은 기본적으로 자바 언어 자체의 보안에 근거하고 있다. 자바에서 제공하는 모든 보안 오브젝트를 이용하여 Jini 시스템의 안전을 보호한다. 그러나 자바의 보안에 근거한 Jini의 보안에는 한계가 있기 때문에 이를 보완하기 위해서 시큐리티가 새롭게 추가되어 Jini 2.0 버전이 2003년 11월에 발표되었다. 이 장에서는 Jini 2.0에서 추가된 Jini 보안을 살펴보고, 홈네트워크에 Jini 시스템을 적용하기 위해서, 홈네트워크에서 요구하는 보안 사항과 Jini 보안을 비교하고자 한다.

1. Jini 2.0 security

Jini 2.0에서는 서비스 제공자와 클라이언트 사이의 상호 인증과, 접근권한제어, 동적인 권한 부여 및 코드의 무결성 체크 등의 시큐리티 기능이 추가되었다. 이들 중 상호인증, 접근권한제어, 동적인 권한부여 기능은 ProxyPreparer라는 클래스가 제공한다. ProxyPreparer클래스는 원하는 서비스의 proxy를 다운로드 하고자 할 때 시큐리티를 지원하기 위한 초기화 과정에 해당한다. ProxyPreparer에서 제공하는 세 가지 기능에는 다음과 같다.

- verify trust in the proxy : proxy가 믿을 만한 소스로부터 온 것인지를 클라이언트가 확인하는 과정
- grant permission to the proxy : 일단 클라이언트가 proxy를 믿을때, 그 proxy에게 일정 권한을 허용하는 과정, 클라이언트가 proxy를 믿는 정도에 따라 허가하는 권한에 차이가 있음
- set constraints on the proxy : 클라이언트가 그 proxy를 다음에 사용 할 때 proxy에

게 요구하는 사항을 미리 설정해 두는 것

위에서 기술하였듯이, Jini 2.0에서 추가된 시큐리티는 클라이언트가 요청한 서비스를 이용하기 전에 그 서비스의 proxy를 다운로드 받을 때, 반드시 거쳐야 할 시큐리티 요구사항을 지정한 것이다. 즉, proxy를 믿을 수 있는지 여부를 판단한 후, proxy에 대한 신뢰의 정도에 따라 권한을 적절하게 허용하고, 그 proxy를 다시 사용할 때를 대비해서 proxy 이용 전에 proxy가 제공해야 할 요소들을 지정해 둔다.

또한 Jini 2.0에서 추가된 부분이 코드의 기밀성 보호이다. 네트워크를 이용한 통신의 기본은 데이터의 무결성을 믿는 것이다. 그런데 Jini를 이용하고, 동적으로 코드를 다운로드 할 때에는 데이터를 고려하는 것이 아니라 object를 고려하는 것이다. 그리고 object는 '코드'와 '데이터'로 이루어져 있다. Jini에서 코드는 out-of-band로 보내지거나, 데이터와 분리된다. 그리고 클라이언트와 서비스들은 object를 포함하는 메시지를 서로 교환한다. object를 여기저기 보낼 때, object에 대한 데이터는 메시지의 한 부분으로써 in-band로 보내지지만, 코드는 그렇지 않다. 메시지의 부분으로써 클래스 파일을 보내는 대신 코드를 어디에서 얻을 수 있는지를 나타내는 URL들을 보내게 된다. URL은 in-band 데이터의 부분이고, 비동기적으로 다운로드 하게 될 코드는 실제 out-of-band로부터 온다. 이를 해결하기 위하여 Jini에서는 httpmdIntegrityVerifier, httpsIntegrity Verifier, FileIntegrityVerifier 클래스를 제공한다. httpmdIntegrityVerifier의 경우 http URL에서 복구한 데이터의 메시지 다이제스트가 그 URL에 명시된 값과 같은지를 체크함으로써 코드의 무결성을 체크할 수 있다.

Jini 2.0에서는 "discovery" 프로토콜을 수행하기위한 요구/응답 패킷들을 보호하기 위한 메커니즘도 제공한다. 즉, "discovery" 프로토콜의 패킷 포맷이 정해져 있다. 그래서 통신하는 상대에게 어떠한 포맷의 "discovery"패킷을 전송하는 지를 알리기 위해서 "selected packet format ID value"를 패킷에 추가한다. "discovery" 프로토콜에서 사용되는 패킷 포맷에는 다음의 다섯 가지가 있다.

- net.jini.discovery.plaintect format : 이 포맷은 discovery 패킷 데이터를 암호화 하거나 무결성 보호 등을 지원하지 않는다.
- net.jini.discovery.x500.SHA1DSA format : plaintext 포맷을 확장한 형태이다. 이는 패킷에 있는 데이터 중 송신측의 DSA 서명과 관련된 인증 블록을 포함한다. 따라서 이 포맷은 패킷 내용의 무결성 보호와 송신측의 인증을 지원한다. 그러나 이 포맷은 패킷 내용의 암호화는 지원하지 않는다.
- net.jini.discovery.x500.SHA1withRSA format : 데이터의 서명을 위해서 DSA 대신 RSA를 사용한다는 점을 제외하면 DSA 포맷과 동일하다.
- net.jini.discovery.ssl format : 패킷이 TLS/SSL 연결을 따라 전송됨을 의미한다. 데이터의 암호화, 인증, 무결성이 보호된다.
- net.jini.discovery.kerberos format : 패킷 데이터가 kerberos v5 GSS-API 메커니즘을 사용하여 안전하게 연결된 통로로 전송된다. Kerberos는 인증을 지원하고, 전송된 데이터의 암호화와 무결성이 GSS-API에 따라서 제공될 수도 있다.

2. Jini 시큐리티 와 홈 네트워크의 시큐리티

이제 Jini에서 제공하는 시큐리티를 홈 네트워크에서의 인증 및 접근권한과 연관지어 기술할 것이다. 먼저, 홈 네트워크에서 필요로하는 인증 기술에는 사용자 인증, 기기간 인증, 메시지 인증이 있고, 접근권한 제어 기술도 필요로 한다.

가. 사용자 인증

홈 네트워크에서의 사용자 인증은 단어 그대로 홈 내의 기기를 사용하는 사람에 대한 인증을 말한다. 즉, 외부인이 홈 내의 기기를 마음대로 조정하도록 두어서는 안되기 때문에 필요한 기술이라 하겠다. Jini에서는 *사용자 인증 기능이 제공*된다고 할 수 있다. Jini 자체에서 제공하는 사용자 인증 메커니즘은 없지만, Jini의 시큐리티는 자바의 시큐리티에 상당부분 의존하기 때문에 이 부분은 자바의 사용자 인증 기능을 이용하면 해결된다. 즉, 자바의 JAAS(Java Authentication and Authorization System)을 이용한다. JAAS를 이용하면 어플리케이션을 사용하기 전에 컴퓨터에 아이디, 패스워드를 입력해서 로그인 하듯이, 아이디와 패스워드를 입력하여 로그인 하여야 한다. Jini에서는 JAAS를 이용하여 클라이언트가 서비스를 이용하기 전에(proxy를 사용하기 전에) 로그인 아이디와 패스워드를 입력 한 후, Proxy-Preparer를 이용하여 proxy 인증을 수행할 수 있다.

사용자 인증을 위한 메커니즘으로 SSL이나 Kerberos를 이용할 수 있는데, SSL을 이용하여 사용자 인증을 하는 경우, 사용자가 로그인 ID를 입력하면 SSL 클라이언트에 관한 파일로부터 사용자가 입력한 ID에 해당하는 패스워드를 읽는다. 사용자가 팝업 다이얼로그 윈도우에 패스워드를 입력하면 비교하여 사용자 인증을 할 수 있다.

Kerberos를 이용할 경우 사용자가 로그인 ID를 입력하면 Kerberos 서버들이 keytab 파일로부터 로그인 ID에 해당하는 key를 읽는다. 사용자가 팝업 다이얼로그 윈도우에 패스워드를 입력하면 사용자 인증을 하고, 인증이 끝나면, 사용자가 다음에 접속했을 때 사용할 수 있는 TGT(Ticket Granting Ticket)

나. 기기간 인증

홈 네트워크에서 기기간 인증은 말 그대로 기기들 사이의 인증이다. 즉, 홈 네트워크에 참여한 기기를 식별하고, 기기가 검증받은 기기 인지를 구분하며, 인증된 기기간의 신뢰 관계를 구축하고 유지하기 위한 것이다. Jini에서는 ProxyPreparer 클래스가 제공하는 세 가지 기능 중 첫 번째 단계인 proxy에 대한 신뢰를 체크하는 기능에서 기기간 인증을 제공한다 할 수 있다. 클라이언트가 proxy를 다운로드 할 때 proxy가 믿을 만한 소스에서 온 것인지를 확인 하고, proxy를 믿을 수 있다는 판단이 섰을 때 proxy를 다운로드 하고, proxy에 적절한 권한을 부여하게 된다. 그리고 서비스를 제공하는 서버의 측면에서 살펴보면, 서버는 자신의 proxy를 다운로드 하여 사용하고자 하는 클라이언트에게 클라이언트 인증을 요구할 수 있고, 서버의 판단에 따라 클라이언트를 믿을 수 없다면 proxy를 다운로드 할 수 없도록 할 수 있다.

Jini에서 이러한 서비스 제공자와 서비스 요구자(이용자) 사이의 상호 인증은 *net.jini.security.security.VerifyObjectTrust* 클래스에서 제공하고, 이 클래스는 자바의 isTrustedObject 클래스를 이용하여 이루어 진다. 이는 인터넷 익스플로러에서처럼 신뢰된 서버 또는 클라이언트들을 미리 정해두고, 이 리스트에 속한 서버 또는 클라이언트는 믿을 수 있다고 판단하는 것으로 보인다.

다. 메시지 인증

홈 네트워크에서 메시지 인증은 홈 외부의 단말로부터 무선으로 받은 메시지가 홈 거주자가 보낸 메시지 인지를 확인하는 과정이다. 이는 자바에서 제공하는 기밀성과 데이터 무결성 체크 기능을 사용하면 메시지 인증이 가능하다. 자바는 자바 보안 패키지에서 전자 서명과 메시지 다이제스트 생성 기능을 제공하며, 자바 보안 패키지가 아닌 JCE (Java Cryptography Extention), JSSE (Java Secure Socket Extention)에서 데이터의 암호/복호화 기능을 제공한다.

라. 접근권한 제어

홈 네트워크에서 접근 권한 제어는 정당한 사용자가 홈 내의 기기에 접근하도록 하거나, 사용자마다 적절한 접근 권한을 두어 홈을 보호하는데 목적이 있다. 예를 들어, 홈을 방문한 손님에게 부여 할 수 있는 접근 권한과 홈의 주인에게 부여 하는 접근 권한에는 차이가 있어야 하고, 같은 홈의 주인이라 하더라도 부모의 접근 권한과 어린 자식의 접근 권한은 달라야 할 것이다. Jini에서 ProxyPreparer가 제공하는 권한 허가 기능을 이용하여 접근권한 제어가 가능하다. ProxyPreparer는 net.jini.security.grant 클래스를 이용하는데, 이 클래스는 자바의 grant 클래스와 관련있다. 동적 권한 허용을 지원하는 proxy의 경우, 실시간으로 object에 권한 부여가 가능하며, 이 object는 홈 내에서 기기를 사용하는 사용자가 될 수 있다. 자바에서 제공하는 접근 권한제어는 각 object와 관련된 ACL을 정리한 파일을 참조하여 objcect에 부여할 권한을 결정하는 방법을 이용하는데, 각 object마다 하나의 ACL을 갖는다.

IV. 결 론

본 논문에서는 Jini에서 사용하는 기본 메커니즘인 discovery protocol, join protocol, lookup service, leasing mechanism과 Jini 2.0에서 추가된 보안 기능에 관하여 간략히 기술하였다. 또한 Jini와 홈 네트워크를 연결하여, 홈 네트워크에서 필요한 인증 기능 및 접근권한 제어 기술이 Jini에서 제공되는지, 그리고 제공 된다면 어떤 방법으로 이들 기능을 제공하는지에 관하여 기술하였다.

Jini의 discovery 프로토콜은 서비스를 제공하기 위해서 룩업 서비스를 찾는 과정이고, join 프로토콜은 찾은 룩업 서비스에 서비스를 등록하는 과정이다. 서비스를 이용하고자 하는 클라이언트는 룩업 서비스에 요청하여 서비스 proxy를 다운로드 받아 서비스를 이용하며, 서비스 제공자의 리소스는 leasing 시간 동안만 이용 가능하다.

Jini와 자바에서는 (Jini의 시큐리티는 자바의 시큐리티에 의존한다.) 홈 네트워크에서 필요한 접근권한 제어 기술을 제공한다. 또한 사용자 인증, 기기간 인증, 메시지 인증 기능을 제공한다. 홈 네트워크에서 필요한 보안 기능을 위해서 Jini에서 제공하는 이들 시큐리티 기술들을 적절히 이용할 필요가 있다.

참고문헌

[1] www.jini.org
[2] Jini specification at www.jini.org
[3] 스콧 오크스 저, 이재광, 홍상욱 역, "Java Security," O'Reilly, 2001.
[4] Ken Arnold, " The Jini Specifications Second Edition," Addison Wesley, 2001.