韓國 컴퓨터情報學會 2006동계학술발표논문집&학회지
第14卷 第2號, 2006. 12.

OSGi 컴퓨팅 환경에서 접근 제어를 이용한 원격 관리 서비스 구현

최 규 상*, 정 헌 만*, 이 세 훈*, 백 영 태**

Implementation of Remote Management Service using Access Control on OSGi Computing Environment

Choi Kyu Sang, Jung Heoun Mam, Lee Se Hoon, Beak Yong Tae

요 약

이 논문에서는 OSGi 서비스 프레임워크 환경에서 발생할 수 있는 보안 요구 사항 중에서 서로 다른 번들의 서비스 간의 상호 작용에서 발생할 수 있는 접근 제어(access control)문제를 설명하고 이를 만족 시킬 수 있는 구체적인 서비스 보안 모델을 제시하였다. 특히 번들이나 서비스에 대한 접근 통제를 유연성과 확장성이 보장되는 방식으로 처리 할 수 있도록 번들에 포함된 파일 형식으로 명시될 보안 정책(security policy)의 구성요소와 의미론을 정의 한다. 또한, 원격 관리 서비스를 위한 효율적인 서비스 구조를 제안하였다.

Abstract

In this paper, we proposed service security model and remote management service on OSGi computing environment. The model is used to make access control decisions like permission-based security inside the java2 platform. In model, policies are defined using a flat text file and include bundles. It method granted flexibility and extendability of access control of bundles and services. Also, we proposed service architecture efficiently for remote management service.

▸ Keyword : OSGi, 접근제어(Access Control), OSGi 보안 모델(Security Model), 원격 관리 서비스(Remote Management Service)

• 제1저자 : 최규상
* 인하공업전문대학 컴퓨터시스템과 ** 김포대학 멀티미디어과

Ⅰ. 서 론

OSGi(Open Services Gateway Initiative)는 개방형 서비스 게이트웨이의 표준을 지향하는 업체들이 모여 만든 표준화 단체로, 전 세계적으로 퍼져있는 컴퓨터 위주의 인터넷을 가정의 가전제품과 점검기기들에 연결하여 다양한 서비스를 제공 받을 수 있게 하고 이를 확산시키기 위한 목적으로 조직된 그룹이다[1,2].

이러한 OSGi 플랫폼 환경의 서비스는 기존의 네트워크 서비스와는 달리 번들이라는 자체 설치가 가능한 컴포넌트 형태로 제공되어 동적으로 배치되며, 다른 서비스와의 상호 작용도 자주 일어난다. 이러한 특성들로 인하여 네트워크상에서 인증되지 않은 오퍼레이터에 의해 악의적인 서비스가 배치되거나 서비스가 변질 될 위험이 있다. OSGi는 기본적으로 자바 보안 모델[3]을 따르고 있으며, 서비스 플랫폼 릴리즈 3에서는 PKI(Public Key Infrastructure) 기반 서비스 번들 인증 메커니즘과 RSH(Remote Communication in a Secure way based on HTTP) 프로토콜을 사용할 것을 권고하는 기본적인 보안 모델 방향을 제시하고 있으나, 저장 공간이나 연산이 제한된 자원을 가지고 있는 OSGi 서비스 플랫폼에서 작동하는 데에는 성능의 저하가 예상된다[4,5,6,7,8]

본 논문에서는 OSGi 서비스 프레임워크 환경에서 발생할 수 있는 보안 요구 사항 중에서 서로 다른 번들의 서비스 간의 상호 작용에서 발생할 수 있는 접근 제어(access control)문제를 설명하고 이를 만족 시킬 수 있는 구체적인 서비스 보안 모델을 제시하고자 한다. 특히 번들이나 서비스에 대한 접근 통제를 유연성과 확장성이 보장되는 방식으로 처리 할 수 있도록 번들에 포함된 파일 형식으로 명시될 보안 정책(security policy)의 구성요소와 의미론을 정의한다.

구현과 실험을 위해 공개 OSGi 프레임워크인 Knopflerfish[9]를 기반으로 하여 홈 서비스를 구현한다.

2. 관련 연구

프레임 워크 자체는 JVM(Java Virtual Machine) 상에서 작동하는 어플리 케이션으로 간주할 수 있으므로 OSGI에 설치되는 서비스에 대한 접근 제어는 호스트의 운영체제 상에 설치된 Java 언어는 네트워크 상에서 이동하며 실행되는 모바일 코드의 보안성을 염두에 두고 개발되었으며, 특히 Java2의 보안 아키텍처에서는 보호 도메인(protection domain)의 개념과 스택 조사(stack inspection)알고리즘을 기반으로 안전한 보안 관리기(security manager)와 확장성 있는 클래스 로더(class loader)를 제공하고 있다.

Java 2 보안 아키텍처[3]의 중앙 집중형 보안 정책은 유연성 있는 고수준의 보안 관리(예 : 주체 그룹화, 권한 위임, 예외 설정 등)가 어렵고, 서로 다른 위치에서 전송된 모바일 코드 간의 상호 작용에 필요한 보안 관계설정이 불가능한 단점을 가지고 있다. 이를 극복하기 위하여 시스템 권한에 거부, 예외를 결합한 복합형 정책 설정, 키 지향 인증서(key-oriented certificate)를 이용한 권한 위임, 모바일 에이전트(mobile agent)에 대한 확장된 보안 정책 설정 등의 연구가 기존의 Java 2 보안 아키텍처 상에서 이루어졌다.

사용자로부터 서비스와 OSGi를 보호하려면 주체를 표현하는 서비스 제공자 명칭과 번들의 URL 위치 대신에 사용자 명칭만 갖는 엔트리를 정의 할 수 있도록 해준다. 서비스 제공자 명칭과 사용자 명칭은 실제로는 해당하는 실체를 증명할 수 있는 인증서(certificate)의 키 저장소 내부에 등록되는 별명(alias)에 해당한다. 번들에 보안 정책을 포함시켜서 다운로드하는 방식은 JAAS의 principal 기반 정책을 이용하는 방식과 유사하므로 사용자에게 전달되는 접속용 프로그램(예: login applet)에 동적으로 해당 사용자의 권한 엔트리를 추가하여 전송할 수 있다. 사용자는 인증서에 대응하는 개인키(private key)를 소유하고 있어야 한다[4,5,6].

3. OSGi 퍼미션과 관리 연산

OSGi 스펙에서 정의하고 있는 퍼미션은 다음과 같다[2].

- AdminPermission : 번들에 대한 라이프 사이클 연산 수행을 할 수 있도록 해 주는 Permission이다. 인터페이스로는 BundleContext와 Bundle이며, API로는 installBundle, start, update, stop, uninstall, getHeaders, getLocation 등이다.
- ServicePermission : 번들 서비스의 등록이나 접근을 통제할 수 있도록 해주는 Permission이다. API로는 registerService, getServiceReference, getService 등이 있다.

• PackagePermission : 번들 import 나 export를 할 수 있도록 해주는 Permission이다. 인터페이스나 API가 not applicable 하다.

다음은 PackagePermission의 코드이다.

```
PackagePermission pp1 = new
 PackagePermission("com.acme.service.print", "export");
```

pp1은 com.acme.service.print 패키지를 export할 수 있는 권한을 가지고 있음.

번들을 얻어오기 위해 getBundle 메소드를 호출할 때 접근 권한을 검사하는 코드이다.

```
public void set(String loc){
        SecurityManager sm =
      System.getSecurityManager();
          if (sm != null){
   sm.checkPermission(adminPermission);
              }
          }
```

OSGi 보안 정책 파일

OSGi에 따른 보안 정책을 명시하기 위하여 flat text file(policy)을 작성한다.

다음은 Policy 파일에 작성한 보안 정책 파일의 예를 보여준다.

```
grant codeBase
"file:/home/Param/out/HomeService.jar"{
permission org.osgi.framework.Servicepermission
"com.acme.service.param.*", "register";
permission org.osge.framework.PackagePermission
"com.acme.service.param", "import,export";
permission org.osgi.framework.AdminPermission;
}

grant "file:/home/Admin/out/Admin.jar"{
permission org.osgi.framework.ServicePermission
"com.acme.service.param.ParameterAdmin", "get";
permission org.osgi.framework.PackagePermission
"com.acme.service.param", "import";
permission org.osgi.framework.AdminPermission;
}
```

4. 접근제어 시스템 구성

HomeService Bundle안에서 HomeClient Bundle과 Admin Bundle을 이용 하여 권한을 어떻게 사용하는 지 알 수 있다. HomeClient Bundle은 자신의 방의 가전기기를 제어 할 수 있다. HomeAdmin Bundle은 Client 번들의 상태를 확인하고 수정 할 수 있다.

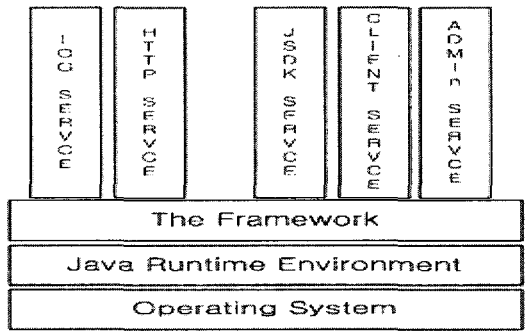

그림 1. 시스템의 계층 구조
Fig 1. hierarchical Architecture of the System

실행 환경 : JDK 1.4, Knopflerfish 2.0

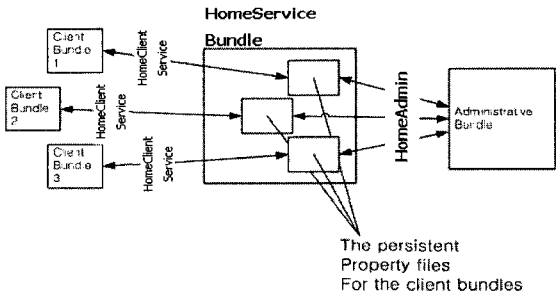

그림 2. 홈서비스 및 관리번들
Fig 2. Home Service and Management Bundles

4.1 번들

• HomeService Bundle

HomeClient Interface : Client Bundle을 이용하여 방의 가전기기를 제어 할 수 있다. HomeClientImpl Class에서 구현.

- HomeAdmin Interface : Admin Bundle을 이용하여 각 층의 상태를 볼 수 있고, 또한 제어가 가능하다. HomeAdminImpl Class에서 구현.
- ServiceFacetory Interface : Bundle간의 동시 접근을 피하기 위하여 OSGI ServiceFactory API를 이용하여 HomeServiceFactory에서 구현.
- HomeStore Class : 데이터를 save, load, clear 할 수 있다. 데이터들은 번들 ID로 디렉터리 생성 후 Property file에 저장 된다.
- Activator Class : HomeService Bundle을 시작하고 종료 할 수 있다.

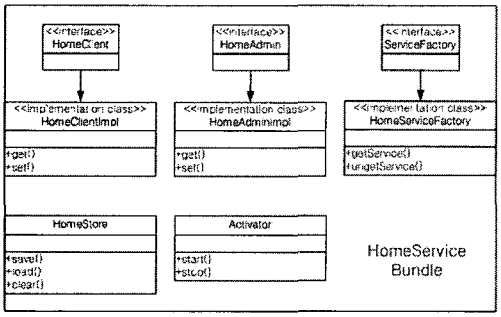

그림 3. HomeService 번들 클래스 구조도
Fig 3. Class Diagram of HomeService Bundle

• HomeClient Bundle

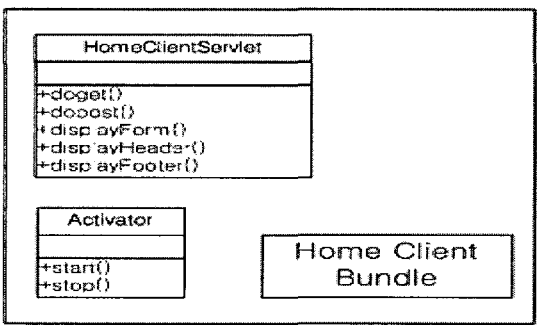

그림 4. HomeClient Bundle Class 구조도
Fig 4. Class Diagram of HomeClient Bundle

HomeClinetServlet Class : OSGI HTTP Service를 이용하여 웹에서 자신의 방에 접속이 가능하다. 방의 상태를 확인 할 수 있으며 값을 입력받아 HomeService Bundle로 전달하여 준다.

• HomeAdmin Bundle

- HomeAdminServlet Class : OSGI HTTP Service를 이용하여 웹에서 Client Bundle의 접속 상태와 각방의 상태를 확인할 수 있으며 수정을 할 수 있다 .

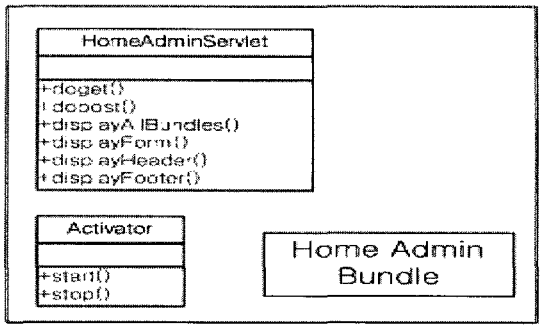

그림 5. HomeAdmin 번들 클래스 구조도
Fig 5. Class Diagram of HomeAdmin Bundle

4.2 핵심 번들 소스

아래는 HomeAdmin Bundle에 권한을 검사하는 소스코드 이다.

```
SecurityManager sm = System.getSecurityManager();
if (sm != null) {
      sm.checkPermission(adminPermission);
}
```

아래는 Client Bundle ID로 저장 하는 소스 코드 이다.

```
AccessController.doPrivileged(
new PrivilegedExceptionAction(){
public Object run() throws IOException {
      String id = Long.toString(b.getBundleId());
      File paramDir = new File(dataRoot, id);

if(!paramDir.exists()) {
        if(!paramDir.mkdir())
        throw new IOException(
        "Could't create dir:" + paramDir);
      }
File paramFile = new File(paramDir, FILENAME);
FileOutputStream out =
        new FileOutputStream(paramFile);
rops.save(out, "Parameters for" + b.getLocation());
```

4.3 실행 결과

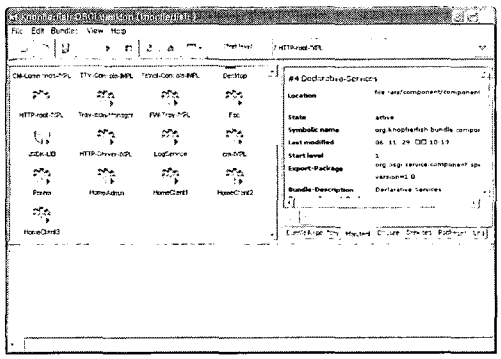

그림 6. Knopflerfish 프레임워크 데스크탑 화면
Fig 6. Knopflerfish Framework Desktop Screen

그림6는 각각의 Bundle을 OSGI Framework에 Install 시킨 화면이다.

• Home Client Bundle

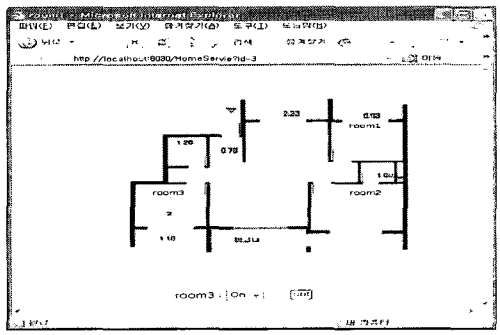

그림 7. HomeClient Bundle 접속 화면
Fig 7. Web Screen of HomeClient Bundle

그림7은 Client Bundle이 접속하였을 경우 방의 상태를 보여주며 상태를 변경할 수 있다.

• Home Admin Bundle

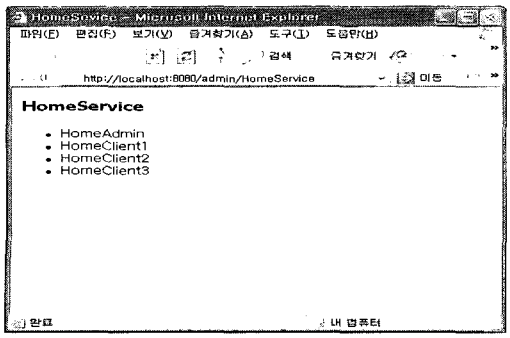

그림 8. HomeAdmin Bundle 접속 화면
Fig 8. Web Screen of HomeAdmin Bundle

그림8은 Admin Servlet에 접속하여 현재 접속 되어 있는 Home Client Bundle을 볼 수 있다.

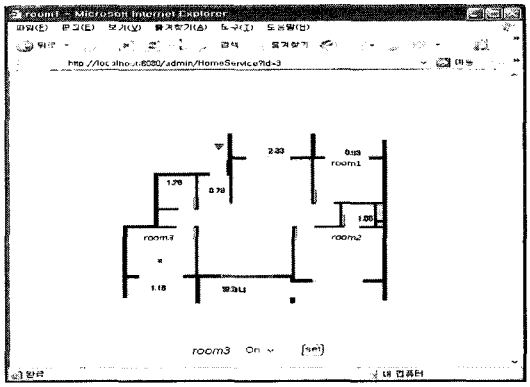

그림 9. Admin 권한으로 접속한 클라이언트 번들
Fig 9. Client Bundle access with Admin Account

그림9와 같이 Client Bundle의 상태를 확인 할 수 있다. 또한 상태를 변경할 수도 있다.

5. 원격 관리 서비스 구성

관리자들은 Bundle의 Life Cycle을 관리 할 수 있는 권한이 있다. 예를 들어 Bundle의 install, uninstall등의 관리를 할 수 있다.

OSGI API는 원격(telnet)에서 접속하여 Bundle을 관리할 수 있는 기능을 제공 하고 있다. Facilitator Bundle을 이용하여 원격에서 번들을 install 하는 방법을 알 수 있다. Map 파일에 Bundle을 기술하여 Install 할 수 있다.

• 번들

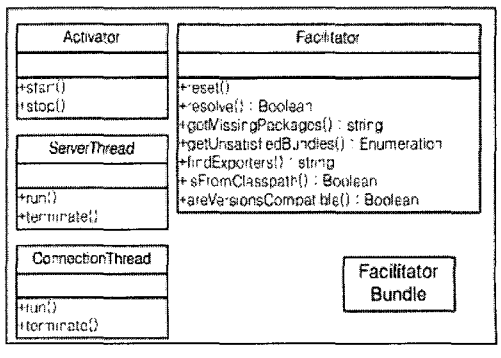

그림 10. Facilitator Bundle 구조도
Fig 10. Class Diagram of Facilitator Bundle

- ServerThread Class : Thread를 상속받아 계속해서 돌고 있으며 외부 클라이언트가 8088 포트로 접속하기를 기다린다.
- ConnectionThread Class : Thread를 상속받아 계속 해서 실행 중이며 ServerThreadClass에서 접속한 외부 클라이언트가 프레임워크에 등록하길바라는 번들의 경로를 받는다.
- Facilitator Class : ConnectionThread Class에서 입력 받은 경로의번들을 그 번들이 의존하는 다른 번들들과 함께Framework에 등록한다.

• 핵심 번들 소스

아래 소스는 소켓을 이용하여 Facilitator Bundle에 접속하여 url경로에 따라 Bundle을 install 한다.

```
Bundle b = context.installBundle(urlstr);
visited.put(urlstr, Boolean.TRUE);
Dictionary headers = b.getHeaders();
```

• 실행 결과

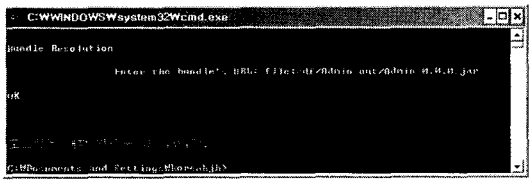

그림 11. Telnet 접속후 url 경로 입력 화면
Fig 11. url input screen after Telnet Contact

그림11과 같이 호스트에 접속하여 url 경로를 입력하면 OK메시지와 함께 번들이 Install 된다. 그 후 접속을 종료한다.

그림12는 Telnet 접속 종료 후 Install 한 Bundle의 상태를 확인할 수 있다.

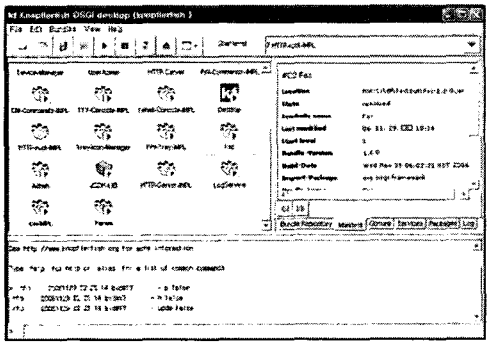

그림 12. Bundle Install후 프레임워크 화면
Fig 12. Framework Screen after Bundle Install

6. 결론

OSGi 프레임워크는 자바2의 보안 모델을 상속 받아 세가지의 확장을 하였다. 이 논문에서는 OSGi 프레임워크 환경에서 발생할 수 있는 보안 요구 사항 중에서 permission의 접근 제어 문제를 만족 시킬 수 있는 유연성과 확장성이 보장되는 서비스 보안 모델을 제시하였으며, 관리자에 의해 프레임워크와 번들을 원격(telnet)에서 접속하여 Bundle을 관리할 수 있는 Facilitator 중심 구조를 구현하고 실험하였다. 제안 시스템을 이용하여 공개 OSGi 프레임워크인 Knopflerfish를 기반으로 하여 홈 서비스를 구현하고 동작의 가능함을 보였다.

참고문헌

[1] OSGi, http://www.osgi.org
[2] Kirk Chen and Li Gong, Programming Open Service Gateways with Java Enbedded Server Technology, Addison-Wesley, 2002
[3] Scott Oaks, Java Security, 2nd Ed., Oreilly, May 2001.
[4] 남기만, 안전한 홈네트워크를 위한 홈게이트웨이 관점의 보안 요구사항 연구, 석사학위논문, 단국대 정보통신대학원, 2006
[5] 임희영, OSGi 환경에서 XML 보안을 이용한 번들의 인증 및 권한부여 메커니즘, 석사학위논문, 고려대학교 대학원, 2005
[6] 김수정, OSGi 서비스 플렛폼에서 사용자 접근제어를 위한 프레임워크와 사용자 관리 서비스, 석사학위논문, 고려대 컴퓨터과학기술대학원, 2005
[7] 황철준, OSGi 서비스 플랫폼 환경에서의 안전한 사용자 인증 매커니즘 연구, 석사학위논문, 경남대학교 대학원, 2005
[8] 김영갑외, "OSGi 서비스 플랫폼 환경에서 서비스 번들 인증 메커니즘의 검증 및 구현", 정보과학회논문지:시스템 및 이론 제31권 제1·2호, 2004. 2
[9] knopflerfish, http://www.knopflerfish.org

저 자 소 개

이 세 훈

1985년 2월 : 인하대학교 전자계산학과 졸업
1987년 2월 : 인하대학교 대학원 전자계산학과 졸업
1996년 2월 : 인하대학교 대학원 전자계산공학과 졸업(공학박사)
1993년 ~ 현재 : 인하공업전문대학 교수
관심분야 : 유비쿼터스 컴퓨팅, 모바일컴퓨팅, 상황인식서비스, 웹서비스

정 헌 만

1996년 2월 서울산업대학교 전자계산공학과
2001년 2월 인하대학교 전자계산공학과 공학석사
2004년 2월 인하대학교 컴퓨터정보공학과 박사과정 수료
관심분야 상황인식, 시맨틱웹, 웹서비스, 유비쿼터스 센서네트워크

백 영 태

1989년 2월 : 인하대학교 전자계산학과 졸업
1993년 2월 : 인하대학교 대학원 전자계산공학과 졸업
2002년 2월 : 인하대학교 대학원 전자계산공학과 졸업(공학박사)
1998년 ~ 현재 : 김포대학 교수
관심분야 : 하이퍼미디어 시스템, 멀티미디어, 지능형시스템, e-Learning