RMI와 SSL를 이용한 멀티플랫폼 환경에서의 안전한 보안패치 분배 시스템 설계

이상원[O] , 김윤주*, 문종섭*, 서정택**, 최대식**, 박웅기**
고려대학교 정보보호대학원*, 국가보안기술연구소**
{a770720[O], zzuya99, jsmoon}*@korea.ac.kr, {seojt, dschoi, ekpark}** @etri.re.kr

Design the Multi-Platform Based Automatic Distribution Method of Security Patches with RMI and SSL

Sangwon Lee[O], Yun-Ju Kim*, Jong-Sub Moon*, Jung-Taek Seo**, Dae-Shik Choi**, Eung-Ki Park**
Center for Information Security Technologies (CIST), Korea University*,
National Security Research Institute**

요 약

개발과정에서의 오류로 인한 취약점을 이용한 공격들이 늘어나면서 이에 대한 보안패치를 체계적인 방법으로 신속하게 자동 분배가 가능한 보안패치 분배 시스템에 대한 필요성이 증대되고 있다. 그러나, 이러한 시스템을 설계 및 구현하는 과정에서 많은 문제점이 대두되었다. 본 논문에서는 이러한 문제점들을 해결하고, 분산 시스템 환경 및 안전한 통신을 기반으로 보안패치를 분배할 수 있도록 하기 위해서 Java RMI와 SSL를 함께 사용한 보안패치 분배 시스템 프레임워크를 설계, 제시하고자 한다.

1. 서 론

최근 새로운 취약점들을 보완하기 위한 패치들이 많이 나오고 있기 때문에 단순히 수동적인 방법으로는 대규모 네트워크를 구축하고 있는 기업체를 비롯한 관공서, 대학 등에서는 시스템의 결함들을 제대로 제거할 수 없다. 따라서, 이러한 과정들을 보다 체계적인 방법으로 자동화 시켜놓은 보안패치 분배 시스템이 필요하다.[7,8]

이러한 필요성을 기반으로 안전한 보안패치 분배 구조의 설계 및 구현[9]을 시작으로 대규모 네트워크 환경에서의 패치 분배 시스템 개발[1]이 진행되어 왔지만, 그 과정속에서 많은 문제점이 표출되어 왔다.

본 논문에서는 위 논문들에서 설계 및 구현한 보안패치 분배 시스템의 프레임워크에서 발생한 문제점들을 해결하기 위해서 JAVA 개발 환경내에서 분산 시스템 프레임워크를 지원할 수 있는 RMI[2,3]와 최근 안전한 인터넷 통신을 위해서 많은 분야에 도입되고 있는 SSL[4]를 이용하여 멀티플랫폼 환경에서의 안전한 보안패치 분배 시스템을 설계하는 방안을 제안하고자 한다.

2. 기본 보안패치 분배 시스템의 문제점

① 보안패치 분배 서비스를 제공해야하는 도메인내에 방화벽이 존재하거나 사설IP를 사용할 경우에 정상적인 서비스를 제공할 수 없다.

② 고정된 동일한 그룹키를 사용하여 보안패치 분배 서비스와 관련한 각종 통신 작업이 이루어지기 때문에 상호 통신 과정속에서의 안정성을 확보할 수 없다.

③ 제대로 된 객체지향 개발 개념을 통한 개발과정이 이루어지지 않았기 때문에, 중요 연구 과제에 대한 개발보다 단순 통신 작업을 위한 라인 프로토콜의 자체 설계 및 개발 과정속에서 많은 부하가 발생하게 된다. 또한 해당 시스템의 재활용성이 매우 떨어지며, 프레임워크 및 기능 확장성이 거의 불가능하다는 단점이 있다.

3. RMI와 SSL

3.1 분산 시스템

분산시스템이란 다수의 컴퓨터가 하나의 시스템처럼 작동하여 사용자에게 그 차이를 느끼지 못하게 하는 시스템을 말한다. 현재 UNIX에서 가장 많이 사용되고 있는 통신 프로토콜이 바로 RPC (Remote Procedure Call)이다.

그러나, RPC는 객체지향 개발 개념을 가지고 구현하고 있지 않기 때문에 분산되어 존재하는 객체간의 메세지 전송을 가능케 해주는 RMI 프로토콜을 사용하게 된다. 근래엔 객체지향 개발 개념을 이용한 시스템 개발이 주류를 이루므로 데이터베이스, 애플리케이션, 그리고 클라이언트가 모두 객체를 통하여 구현되고 따로 저장되는 일이 많아졌기 때문이다.[2,3]

3.2 RMI를 이용한 설계시 얻을 수 있는 장점 [2,3]

① RMI에서는 TCP/IP로 통신이 안되면 스스로 HTTP를 사용하여 통신을 시도하므로, 방화벽이 있는 경우에도 통신이 가능하게 할 수 있다.

② 분산 객체간의 상호작용 패턴을 캡슐화한 객체들을 조립식으로 연결만 하면 시스템이 완성되므로, 분산 응용 소프트웨어의 생산성 및 확장성을 향상시킬 수 있다.

③ 대규모 네트워크 서비스를 지원하는 프레임워크 설계시 항상 객체의 전달에 관한 문제에 직면하게 되는데 그때마다 발생하는 라인 프로토콜 자체 개발로 인한 많은 시간과 비용이 소요되는 부담을 없앨 수 있다.

④ RMI 자체의 안정적인 구조하에서 응용 프로그램이 동작하기 때문에 라인 프로토콜의 자체 설계 및 개발 과정속에서 발생할 수 있는 에러들을 사전에 방지할 수 있으며, 원격 호스트에서 발생한 예외를 알 수 있기 때문에 보안패치 분배 시스템이 갖추어야하는 필수조건인 안정성을 확보할 수 있다.

3. SSL (Secure Sockete Layer)

3.1 SSL [4]

SSL은 인터넷 프로토콜이 보안면에서 기밀성을 유지하지 못한다는 문제를 극복하기 위하여 Netscape Communications가 개발한 것으로서, 웹브라우저와 웹서버간에 데이터를 안전하게 주고받기 위한 업계 표준 프로토콜이다. 최근 보안을 요구하고 있는 많은 분야에서 사용하고 있다.

3.2 SSL를 이용한 설계시 얻을 수 있는 장점[4]

① 메시지에 대한 기밀성을 제공한다.
② 메시지에 대한 무결성을 제공한다.
③ 자신의 행위에 대한 부인방지 기능을 제공한다.
④ 통신하고 있는 상대방을 인증할 수 있다.

4. RMI와 SSL를 이용한 프레임워크 디자인

4.1 보안패치 분배 시스템 전체 구조[1, 10, 11]

① 보안패치 매니저 : 보안패치 분배서버, 보안패치 DB 그리고 보안패치 프로파일에 대한 관리 기능을 수행하며 웹 기반의 사용자 인터페이스를 제공
② 보안패치 분배 서버 : 클라이언트와 보안패치 프로파일에 기반한 분배 메커니즘을 사용하여 실제 패치 분배 과정을 수행
③ 보안패치 DB : 보안 도메인 내에 구성되어 있는 클라이언트에 필요한 보안패치 파일 및 관련 정보를 저장하여 보안패치 분배 과정에서 보안패치 분배 서버의 요청에 의해서 보안패치를 제공
④ 보안패치 에이전트 : 보안패치 클라이언트와 보안패치 프로파일 정보에 대한 관리 기능을 수행하며 보안패치 에이전트는 웹 기반의 사용자 인터페이스를 제공
⑤ 보안패치 클라이언트 : 패치 분배 서버와 보안패치 프로파일에 기반한 분배 메커니즘을 사용하여 실제 보안패치 분배 과정을 수행

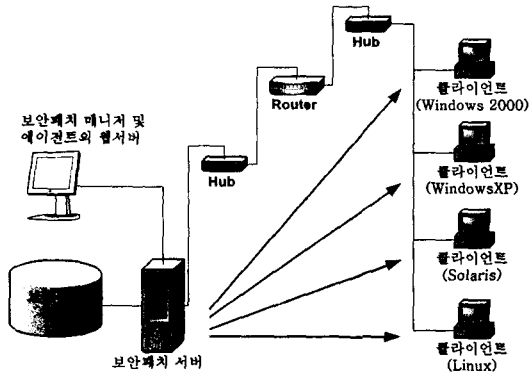

그림 1 보안패치 분배 시스템 전체 구조

4.2 분산 객체간 통신 기반 구조[그림2]

① 보안패치 서버 객체와 보안패치 클라이언트 객체는 상호 통신 작업이 필요하기 때문에 각각의 원격 객체가 호출할 수 있는 인터페이스를 정의한다.
② 보안패치 에이전트 객체와 보안패치 매니저 객체는 보안패치 서버 객체에 필요한 정보를 단순히 전달해주는 역할만을 수행하기 때문에 별도의 Interface를 정의하지 않으며, 단순히 보안패치 서버의 인터페이스만을 호출할 수 있다.

4.3 보안패치 서버 객체의 구성

4.3.1 Remote Interface

① void SendProfile (String strUserID)
- 클라이언트로부터 프로파일을 다운로드

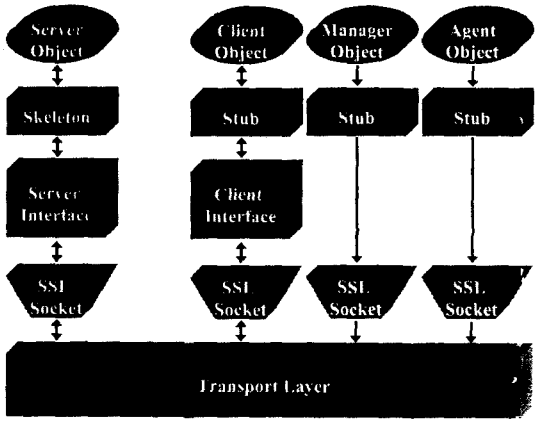

그림 2 분산 객체간 통신 기반 구조

② String[] GetSupportProgramList
(String strUserID, String strOSType)
- 분배 서비스 지원 대상 프로그램 목록 가져오기
③ byte[] DownloadPatchfile
(String strFilePathName, Long lFilePointer)
- 패치파일을 전송
④ void InsertNewPatchfile (String strFilePathName)
- 새로운 패치를 등록
⑤ structure NEW_PATCHFILE_INFO GetNewPatchfileInfo
(String strFilePathName)
- 새롭게 등록된 패치파일에 관한 정보 가져오기
⑥ void SendSplitedClientList
(String[] strSplitedClientList, String[] strPatchList)
- 그룹화 기능시 필요한 클라이언트 목록 전송
⑦ void ClientRequestPatchfile
(String strUserID, int[] nRequestFileList)
- 클라이언트가 패치파일 전송 요청

4.3.2. Local Function

① structure SaveClientSystemInfo (String strUserID)
- 프로파일로부터 정보를 추출하여 DB에 등록
② String SearchPatchfile structure
(structure CLIENT_SYSTEM_INFO)
- 설치가 필요한 패치가 있는지 검색
③ String[] SearchClientForNewPatchfile
(structure NEW_PATCHFILE_INFO)
- 새로운 패치 파일 설치가 필요한 클라이언트 목록 가져오기

④ String[] HaveDistributionServer ()
- 새로운 패치 분배용 서버가 존재하는지 검사

⑤ String[] SplitClientList
(String[] strClientList, String[] strDistributionServerList, String[] strPatchList)
- 클라이언트 목록을 나눈다.

⑥ void DistributeToClients
(String[] strClient, String[] strPatchList)
- 패치파일을 클라이언트에게 분배

⑦ void CreatePatchListWhenRequested
(String strUserID, int[] nRequestFileList)
- 클라이언트 요청시 패치 리스트 생성

4.4 보안패치 클라이언트 객체의 구성

4.4.1 Remote Interface

① Byte[] RequestProfile
(String strUserID, Long lFilePointer)
- 프로파일 전송

② void SendPatchfileList
(String[] strPatchList)
- 설치해야할 패치파일 목록 전송

4.4.2 Local Function

① String ReadMyID ()
- 자신의 ID 가져오기

② void CreateProfile_Windows (String strUserID)
- Windows 경우의 프로파일 생성

③ void CreateProfile_Linux (String strUserID)
- Linux 경우의 프로파일 생성

④ void CreateProfile_Solaris (String strUserID)
- Solaris 경우의 프로파일 생성

⑤ String CreateHash (String strHashValue)
- 해쉬값을 생성

⑥ boolean CheckHash (String strHashValue)
- 새 해쉬값과 기존 해쉬값을 비교 및 분석

⑦ String GetOSType ()
- 클라이언트 운영체제 종류 분석

⑧ CreateBatchfile (String[] strPatchList)
- 자동설치를 위한 배치파일을 생성

⑨ InstallPatchfile ()
- 패치파일 설치

4.5 보안패치 분배 시스템에 대한 SSL의 적용[6]

- 자바 RMI에서는 기본적으로 소켓통신 구현 부분을 외부에 노출시키지 않고 내부적으로 처리하도록 되어 있다. 그러나, RMI를 통한 통신 과정에서 인의적인 소켓을 생성할 필요가 생길 경우를 대비해서 자바 내부에는 Custom RMI Socket Factory라는 기능을 제공한다.
- 이 기능을 사용하여 SSL socket를 인의적으로 생성함으로써 RMI의 Custom RMI Socket Factory 부분에 SSL를 접목시킬 수 있다.

5. 결론

보안패치 분배 시스템은 특정 규모의 네트워크에 대해서 취약점을 가지고 있는 클라이언트가 존재하지 않도록 관리해주는 시스템이다. 이때 네트워크를 이루고 있는 클라이언트들의 플랫폼은 매우 다양할 수 있다. 이러한 환경을 지원하기 위해서 각각의 플랫폼별로 보안패치 분배 시스템을 설계 및 개발하는 것은 많은 부하가 걸리는 작업이며, 이렇게 개발한 각각의 모듈들을 하나로 통합하여 커다란 보안패치 분배 시스템을 운영하는 것은 자칫 안정적이지 못한 시스템을 양산할 수 있다.

더군다나 보안패치 분배 시스템이 정상적으로 작동하지 않을 경우 해당 네트워크의 안정성에 큰 위협을 가할 수 있기 때문에, 보안패치 분배 시스템은 반드시 그 운영에 있어서 안정성을 보장할 수 있어야만 한다.

이러한 조건들을 모두 충족시키며, 원격 객체 사이의 메쏘드 호출을 통한 분산 컴퓨팅 기능을 제공할 수 있도록 하기 위해서 본 논문에서는 Java RMI 기술과 안전한 통신을 보장하기 위해서 SSL를 함께 도입한 보안패치 분배 시스템을 제안하였다. 향후 제안한 스킴을 가지고 실제 구현을 함으로써 보다 실질적이고 효과적인 보안패치 분배 시스템을 구축하는데 필요한 연구를 진행하고자 한다.

6. 참고문헌

[1] 손태식, 서정우, 구원본, 민동욱, 이상원, 김대공, 김윤주, 문종섭, "대규모 네트워크 환경에서의 패치 분배 시스템 개발", KoreaCrypt'03, CIST, 2003

[2] Java RemoteMethodInvocation Specification, ftp://ftp.java.sun.com/docs/j2se1.4/rmi-spec-1.4.pdf, Sun Microsystems, Inc.

[3] Wutka, M., "Hacking Java: The Java Professional's Resource Kit", pp. 281~408, QUE Corporation, 1997

[4] Frier A., Karlton P. & Kocher P., The SSL Protocol Version 3.0, http://wp.netscape.com/eng/ssl3/draft302.txt

[5] Sridharan, P., "Advanced Java Networking", pp. 95~231, Prentice Hall PTR, 1997

[6] Using RMI with SSL, http://java.sun.com/j2se/1.4.2/docs/guide/rmi/socketfactory/SSLInfo.html, Sun Microsystems, Inc.

[7] Sohn Tae-Shik, "Safe Patch Distribution Architecture in Intranet Environments", SAM, 2003

[8] Cheol-Won Lee, "A Secure Patch Distribution Architecture", ISDA 2003, Lecture Notes in Computer Science, Springer-Verlag, 2003

[9] Tae-Shik Sohn, Jung-Woo Seo, Jong-Sub Moon, Jung-Taek Seo, Eul-Gyo Im, Cheol-won Lee, "Design and Implementation of a Secure Software Architecture for Security Patch Distribution", 한국정보보호학회, 2003

[10] Sohn Tae-Shik, "Safe Patch Distribution Architecture in Intranet Environments", SAM, 2003

[11] Cheol-Won Lee, "A Secure Patch Distribution Architecture", ISDA 2003, Lecture Notes in Computer Science, Springer-Verlag, 2003