안전한 패치 분배 구조 설계

손태식[0], 김진원, 박일곤, 문종섭*, 서정택, 임을규, 이철원**
*고려대학교 정보보호기술연구센터, **국가보안기술연구소
{743zh2k[0], mirujohn, cist_gon, jsmoon}@korea.ac.kr, {seojt, imeg, cheolee}etri.re.kr

The Design of a Secure Patch Distribution Architecture

Tae-Shik Sohn[0], Jin-Won Kim, Il-Gon Park, Jong-Sub Moon, Jung-Taek Seo, Eul-Kyu Im, Cheol-Won Lee
*CIST, Korea Univ, **NSRI

요 약

시스템이나 네트워크 보안에 있어 관련된 패치의 업데이트는 보안의 최우선적 사항이다. 그러므로 패치 분배과정에 있어 패치 정보가 누출된다는 것은 시스템이나 네트워크의 취약성이 그대로 노출되는 것과 같은 문제를 발생시킨다. 본 논문에서는 일반적인 인터넷 환경에서의 패치 분배가 아닌 특정 조직이나 기관의 도메인 내부에서 안전하게 패치를 분배할 수 있는 구조에 대하여 제안한다. 본 제안 구조에서는 서버 인증서 기반의 사용자 인증, SKIP 모듈러스를 사용하는 DH 키분배, 패치 체크섬 암호화를 통한 기밀성·무결성 보장, 원격지에서의 패치 자동 설치 기능을 제공한다.

1. 서론

운영체제 및 여러 응용 프로그램들은 프로그램 개발 과정의 특성상 보안 취약성을 가지고 있기 마련이다. 또한 현재 인터넷이 폭넓게 보급되어 네트워크를 통한 시스템 침해 사례가 증가되고 있다. 이러한 현실에서 패치에 대한 안전하고 신속한 분배 및 원격지에서의 자동 설치 같은 서비스 제공은 해당 시스템의 보안을 위한 가장 기본적이고 필수적인 요소이다. 패치는 대상 시스템에서 있어 취약성을 보완해주는 일종의 보조 프로그램이라고 말 할 수 있으며, 이러한 패치는 시스템에 주기적으로 계속적인 업데이트가 수반되어야 하는 필수적인 요소이다. 하지만 시스템 역공학적인 관점에서 볼 때 해당 시스템에 필요한 패치의 종류, 설치된 프로그램에 대한 부가 사항 등의 패치 정보가 누출된다면, 결국 대상 시스템의 취약한 부분을 그대로 보여주는 것과 같은 결과를 초래할 수 있다. 또한 악의의 제3자에 의해 트로이안 목마와 같은 백도어 기능이 삽입된 패치가 자신의 네트워크 내부의 중요 시스템에 적용된다면, 시스템 침해 및 기밀 정보 누출이라는 보안상 치명적인 문제점을 가져 올 수 있다. 즉, 공개 환경에서의 개인 사용자의 경우와 달리 특정 조직이나 기관의 시스템에 대한 패치 분배는 민감한 문제가 아닐 수 없다.

그러므로 본 논문에서는 여러 운영체제 벤더들의 패치 자동분배 방안에 대한 관련 기술 분석을 바탕으로 특정 조직이나 기관의 단일 도메인 내의 안전한 패치 분배 구조에 대하여 제안한다.

2. 기존의 관련 연구 동향

2.1 벤더별 패치 분배 동향

본 논문에서는 23개의 운영체제 벤더들이 보안 패치를 분배하는 과정에 있어서의 패치 분배 및 인증, 무결성, 기밀성 제공 방안에 대하여 분석하였다. 패치의 분배 과정에 있어서 전자서명과 같은 암호학적 기법을 사용하여 안전하게 패치를 분배할 수 있는 기반을 제공하는 것은 중요하며 분배 과정에 있어서는 적극적인 분배와 수동적인 개념의 분배로 나누어서 고려할 수 있다. 다음에서는 분석에 사용된 운영체제별 패치 분배 과정의 인증 및 무결성, 기밀성 등 보안성을 보장 방안에 대해서 열거한다.

- PGP(Pretty Good Privacy)를 이용하여 여러 가지 형태로 사용 가능 : 이 방법은 첨부되는 전자 서명이 전체 패치 파일에 대한 서명으로서 사용 가능하며 또한 일부 패치 문서에 대한 서명에 대해서도 사용 가능. 서명된 패치에 대한 인증 및 무결성 제공 가능
- HTTPs 연결 설정을 통한 패치 분배 : SSL(Secure Socket Layer) 기반의 패치 분배 기밀성 보장 기능 제공
- SSH(Secure SHell) 연결 설정을 통한 패치 분배 - 패치 분배 과정에 있어서의 기밀성 보장 가능

[표 1] 벤더별 분배 기술 분석

적용기법 / 운영체제	PGP	HTTPs	SSH	Package	기타
Caldera/Open Linux	O			O	
Cobalt				O	일부 사용
Compaq/Tru64 unix		O			일부 사용
Conectiva Linux	O			O	
Corel	O				
Debian Linux	O			O	
FreeBSD	O				SFSRO
HP-UX	O	O			일부사용
IBM-AIX	O	O			
WireX/Immunix	O				자가서명
Mandrake Linux	O			O	
MicroSoft Windows	O				
NetBSD	O		O		ssh 일부사용
Novell/Netware					Equifax
OpenBSD			O		CD권고
Redhat Linux	O			O	
SCO					PGP 권고
SGI/IRX	IO				
Slackware Linux					CD
Sun Solaris	IO	O			일부사용
Suse Linux	O			O	일부사용
Trustix	O			O	자가서명
Turbo Linux	O	O		O	

위와 같은 패치 분배 과정의 여러 보안 요구사항을 충족시키기 위한 방안 중 처음에 언급된 PGP 서명은 보편적으로 가장 많이 사용되는 방안이지만 PGP 서명만을 통해서는 충분한 보안성을 얻을 수 없다. 패치 분배 과정에 있어서 요구되는 보안 사항으로는 기본적으로 사용자 인증과 패치 무결성 보장이 필요하며 또한 전송과정에 있어서의 기밀

성 또한 필수적으로 제공 되야 할 보안 요구 사항이다.

또한 패치 분배 과정 자체는 적극적인 개념의 패치 분배와 수동적인 개념의 패치 분배로 나뉠 수 있다. 적극적인 개념의 패치 분배는 새로운 패치가 나왔을 때 패치 제공 밴더가 고객에게 패치의 존재에 대하여 전자메일과 같은 수단을 사용하여 공지를 보내 알리고 이러한 통보를 받은 사용자는 해당 패치를 얻을 수 있는 웹사이트나 FTP에 접근하여 다운로드받는 것이다. 그리고 수동적인 개념의 패치 분배는 밴더들이 제공하는 웹사이트나 FTP사이트에 사용자가 접근하여 패치를 업데이트하고 관련 정보를 검색하고 해당 패치를 다운로드 하는 것이다.

2.2 패치 분배 과정에서의 취약점

패치 분배 과정에 있어서 사용자들은 일반적으로 패치가 정당한 사이트로부터 인증되어 자신에게 올바르게 배포되었다고 여기게 된다. 이러한 이유는 대부분의 패치 분배는 밴더가 제공하는 자체 사이트를 통해서 분배되기 때문이다. 따라서 이러한 사용자의 잘못된 믿음과 패치 분배 과정 자체에서 기인하는 여러 보안 취약성으로 인해서 많은 문제점을 내포하고 있다. 아래의 사항들은 패치 분배 과정에서 발생할 수 있는 취약성들을 나열한 것이다.

- 밴더의 패치 서명과 사용자의 패치 검증에 사용되는 키가 취소되는 경우에 그 즉시 사용자가 이러한 키 폐기 사실을 알기 어려움
- 패치 인증에 사용된 서명키가 실제로 밴더의 서명키인지 명확하게 검증하는 것이 어려움
- 전자 서명이 패치 파일 자체에 대한 인증이 아닌 단지 패치 정보의 인증에만 사용되는 경우
- 올바른 서명키로 악의의 목적을 가지고 제작된 패치가 서명되어 분배되는 경우
- 디지털 서명 소프트웨어(예 PGP)나 이 소프트웨어를 신뢰하는 OS의 구성요소에 트로이 목마 프로그램의 배포

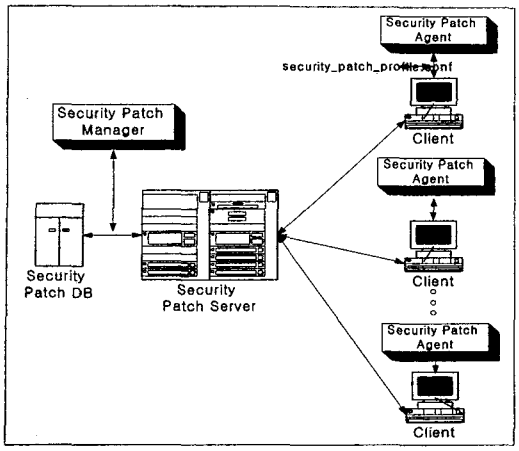
3. Our Approach

본 논문에서 제안하는 안전한 패치 분배 구조는 운영체제 밴더들이 공개 도메인(Public Domain)상의 인터넷 웹 서비스, FTP 서비스를 통한 분배 환경과는 다른 특정 조직이나 기관의 제한된 보안 도메인(Security Domain) 안에서 수행되는 것을 가정한다. 제안하는 안전한 패치 분배 구조에서는 보안 도메인 안의 사용자 인증 방안, 패치 전송 과정에서의 기밀성 보장 방안, 분배된 패치 파일의 무결성 보장 방안 그리고 원격 시스템에서의 패치 자동 설치 기능을 제안한다.

3.1 안전한 패치 분배 전체 구성도

안전한 패치 분배 전체 구성은 패치 DB, 패치 분배 서버, 패치 매니저, 패치 클라이언트, 패치 에이전트로 구성된다.

- 패치 DB : 보안 도메인 내에 구성되어 있는 시스템에 해당하는 패치 파일 및 관련 정보와 패치 클라이언트 시스템의 패치 프로파일 및 사용자 정보를 저장(이때 패치 DB에 저장되는 패치 파일은 패치 프레임워크에 알맞은 포맷으로 변환되어 저장)
- 패치 서버 : 패치 클라이언트와의 분배 프로토콜을 통하여 패치 클라이언트에게 필요한 패치를 패치 DB로부터 가져와 분배 과정 수행
- 패치 클라이언트 : 패치 서버에게 필요한 패치를 요구하는 등의 실제 패치 분배 과정 수행
- 패치 매니저 : 패치 DB의 구성 정보에 대한 관리 및 패치 서버 관리, 패치 매니저는 웹기반의 UI를 이용하여 관리자에게 편의성을 제공
- 패치 에이전트 : 패치 클라이언트 측에 설치되어 대상 클라이언트의 패치 정보 관리 또한 웹기반의 UI를 통해 사용자에게 편의성을 제공



[그림 1] 패치 자동분배 프레임워크 전체 구성도

안전한 패치 분배 구조의 패치 자동분배 과정은 새로운 패치가 제공되는 경우 서버 측에서 해당 클라이언트에게 패치를 제공하는 과정과 클라이언트 시스템 환경 변화에 의해 클라이언트의 패치 요청에 의한 패치 분배과정으로 나뉜다.

가. 서버의 패치 제공 과정

1. 운영체제의 밴더에 의한 새로운 패치 발표
2. 패치 DB에 새로운 패치 파일 및 관련 체크섬 정보 등 저장(패치의 저장과정에서 패치 DB 포맷으로 변환)
3. 새로운 패치 분배를 필요로 하는 클라이언트의 선정
4. 대상 클라이언트와의 상호 인증 과정 및 패치 분배
5. 클라이언트의 패치 자동설치
6. 클라이언트는 새롭게 변경된 패치 프로파일을 패치 서버에게 전송
7. 패치 서버는 클라이언트의 패치 프로파일 업데이트

나. 클라이언트의 패치 요청 과정

1. 클라이언트 시스템의 구성 환경 변화 발생
2. 클라이언트의 패치 프로파일 설정 변경
3. 클라이언트의 패치 프로파일 변경 사항을 패치 서버에게 통보
4. 패치 서버는 갱신된 패치 프로파일에 따른 해당 패치 검색
5. 패치 서버는 클라이언트와 상호 인증 및 패치 전송
6. 클라이언트의 패치 자동설치
7. 클라이언트는 새롭게 변경된 패치 프로파일을 패치 서버에게 전송
8. 패치 서버는 클라이언트의 패치 프로파일 업데이트

3.2 Patch Distribution Mechanism

가. 패치 인증, 기밀성, 무결성 보장 프로토콜

패치 분배 인증 프로토콜에서는 인증서 기반의 사용자 인증, SKIP 모듈러스를 사용하는 Diffie-Hellman 키 분배를 통한 기밀 통신 그리고 패치에 대한 MD5 체크섬 확인 기능을 제공한다.

- 패치 서버 인증서 기반의 사용자 인증

Assume : 인증에 사용되는 서버의 인증서는 사전 사용자 등록 과정에서 사용자에게 제공

서버는 우선 클라이언트에게 패치 분배 메시지를 타임스탬프 값과

함께 전송한다. 이 메시지를 수신한 클라이언트는 랜덤 넘버를 생성하여 미리 제공받은 서버의 공개키로 암호화하여 전송하고 이 랜덤값에 대한 검증이 수행되면 서버에 대한 인증을 확인하게 된다. 그 후 클라이언트는 자신의 패스워드를 다시 서버의 공개키로 암호화 전송하여 클라이언트에 대한 인증을 수행한다. 이렇게 본 논문의 제안 구조에서는 서버 인증서를 기반으로 서버/클라이언트 상호 인증이 가능하다.

SP_Server -> SP_Client	: Send Msg_Patch_Distribution | T
SP_Client	: Generating R
SP_Client -> SP_Server	: Epub[R|T]
SP_Server	: Verify Dpri[Epub[R|T]]
SP_Server -> SP_Client	: ER+1[T]
SP_Client	: Verify DR+1[ER+1[T]]
SP_Client -> SP_Server	: ID | Epub[PW]
SP_Server	: Verify Dpri[Epub[PW]]
SP_Server -> SP_Client	: Send Msg_auth_success

- SKIP 모듈러스를 사용하는 DH 기반 패치의 기밀성 및 무결성

SKIP 모듈러스를 사용한 DH 키분배를 통하여 서버와 클라이언트간 암호화된 통신 채널을 확립하며 이 기밀 통신 채널로는 패치의 체크섬 값만을 암호화하여 전송한다. 즉, 부가적인 패치 파일에 대한 암호화 없이 체크섬 값을 암호화하여 통신 효율을 증가시킨다. 이렇게 암호화된 체크섬은 패치 체크섬 자체의 기밀성 보장은 물론 패치 체크섬 값의 안전한 분배를 통해 패치 자체에 대한 기밀성을 또한 제공한다. 패치와 암호화된 체크섬을 전송 받은 클라이언트는 암호화 된 체크섬 값만을 복호화하여 전송된 패치의 체크섬을 계산한 값과 비교한다. 만약 비교된 값이 일치하면 패치의 무결성이 보장된 것이며 그렇지 않은 경우에는 패치 파일을 폐기한다.

SP_Server -> SP_Client	: P | Ek[C(P)]
SP_Client	: Dk[Ek[C(P)]], C(P)
SP_Client -> SP_Server	: if Patch is verified, Send Msg_Patch_verified

나. 패치 분배 프로토콜

패치를 분배하는 과정은 서버 측에서 패치를 제공하는 경우와 클라이언트 시스템에서 필요한 패치를 요청하는 두 가지 경우로 나눌 수 있고 다음에 각각의 경우에 대한 패치 제공 프로토콜을 제안한다.

1) 패치 제공 프로토콜(패치 서버)

- 새로운 패치가 패치 DB에 업데이트 되는 경우 해당 패치가 필요한 클라이언트에게 패치 제공

SP -> SP_DB	: 새로운 패치가 패치 DB에 업데이트 됨
SP_Server	: 해당 패치가 필요한 클라이언트 선별
SP_Server -> SP_Client	: 클라이언트에게 패치 제공 메시지 전송
SP_Client -> SP_Server	: 패치 서버에게 승낙/거부 메시지 전송
SP_Server <-> SP_Client	: 사용자 인증
SP_Server -> SP_Client	: 패치 전송
SP_Client	: 패치 설치
SP_Client -> SP_Server	: 설치 후 클라이언트 프로파일 정보 전송
SP_Server	: 클라이언트의 패치 프로파일 업데이트

2) 보안패치 요청 프로토콜(클라이언트)

- 클라이언트 시스템의 구성 환경 변화나 그 외 기타 상황으로 인한 보안패치 요청

SP_Client_Agen	: 시스템 변경 후 필요한 보안패치 선별
SP_Client	: security_patch_profile 설정 파일을 참조
SP_Client -> SP_Server	: 필요한 보안패치 요청 메시지 전송
SP_Server -> SP_Client	: 클라이언트에게 승낙/거부 메시지 전송
SP_Client <-> SP_Server	: 상호간 인증서 or ID/PW 기반 인증
SP_Server -> SP_Client	: 보안 패치 전송
SP_Client	: 패치 설치
SP_Client -> SP_Server	: 설치 후 클라이언트 패치 프로파일 전송
SP_Server	: 클라이언트의 보안패치 프로파일 업데이트

3.3 Patch Auto Installation & Management Mechanism

가. 패치 자동 설치 프로토콜

원격지에서 분배받은 패치를 자동으로 설치하기 위해서 먼저 각 클라이언트에 대한 패치 프로파일을 중앙의 패치 서버에서 관리하는 것이 필요하다. 즉, 클라이언트에서는 현재 시스템에 구성된 패치 정보를 하나의 패치 프로파일로 구성하고 구성된 정보를 패치 서버에게 전송한다. 패치 서버는 이러한 클라이언트들에 대한 패치 프로파일 정보를 관리하며 특정 시스템에 대한 패치가 발표되었을 때 해당 클라이언트를 선별하여 패치를 분배한다. 또한 이렇게 분배된 패치는 해당 클라이언트에서 자신의 패치 프로파일 정보를 확인함으로써 관리자의 부가적인 관여 없이 자동으로 설치 될 수 있다. 다만 시스템 의존적인 특성으로 발생하는 패치 설치 과정의 에러는 각 에러 경우에 대한 이밴트 매니저를 사용하여 시스템 관리자에게 즉각 통보 될 수 있게 한다.

SP_Client -> SP_Server	: Send Patch_Profile
SP -> SP_Server	: Searching
SP_Server -> SP_Client	: Distribute Patch
SP_Client	: Refer Patch_Profile, : Auto Installation

4. 결론 및 향후 연구 방향

논문에서는 패치 분배 과정에 있어서 발생할 수 취약성을 보완하기 위한 안전한 패치 분배 구조를 제안하였다. 제안된 패치 분배 구조에서는 서버 인증서 기반의 사용자 인증 기능 제공, SKIP 모듈러스를 사용한 DH 기반의 키교환, 암호화된 패치 체크섬을 이용한 패치 기밀성 및 무결성 보장 그리고 패치 프로파일 관리를 통한 원격지에서의 패치 설치 기능을 제공한다. 이러한 패치 분배 프레임워크를 통하여 보안 도메인 내의 패치 분배 과정에서의 패치 정보 누출로 인한 시스템 침해 사례의 방지는 물론 보다 안전한 보안성을 제공할 수 있다.

향후 연구 방향으로는 패치DB에 저장되는 패치 파일에 대한 정보 및 패치 프로파일 정보 구성을 위한 포맷의 정의가 필요하며 제안된 스킴의 세부 구현 및 분배 시나리오 구성에 대한 검증이 요구된다.

5. 참고문헌

[1] "Vulnerabilities in Operating-System Patch Distribution", http://razor.bindview.com/publish/papers/os-patch.html

[2] M.A. Bashar,"Low-threat security patches and tools ", ICSM '97, October, 1997 , Bari, ITALY

[3] Tim Polk. Automated tools for testing computer system vulnerability. Technical Report NIST SP800-6, National Institute of Standards and Technology,1993

[4] Comerford White. ABYSS: A trusted architecture for software protection. In Proc. 1987 IEEE Symposium on Security and Privacy, Oakland, California, pages 38-51. IEEE Computer Society Press, April 27-29, 1987.