범용적 패치분배 시스템을 위한 효율적인 DB설계

이인용, 이수영, 조재익, 문종섭
고려대학교 정보보호기술연구센터
{iylee, leesuyoung, chojaeik, jsmoon}@korea.ac.kr

Effective Patch Database Composing for Multi-OS and S/W

Inyong Lee, Suyoung Lee, Jaeik Cho, Jongsub Moon
Korea University Graduate School of Information Management & Security

요 약

소프트웨어가 증가하면서 소프트웨어에서 발생되는 프로그램의 오류도 증가하게 되었고, 이런 프로그램 오류를 해결하기 위한 패치도 증가하게 되었다. 하루가 다르게 급수적으로 증가하는 패치를 효율적으로 관리하기 위해서 관공서나 기업에서는 패치를 별도로 관리할 수 있는 패치관리시스템을 구성하여 운영하고 있으며, 학교 및 연구기관에서는 안전하게 패치가 분배될 수 있도록 연구를 진행하고 있다. 하지만, 기존의 연구는 안전한 패치분배에 집중되고 있으며, 급수적으로 늘어나는 패치를 효율적으로 관리할 수 있는 방안에 대해서는 연구가 부진한 것이 현실이다. 본 논문에서는 다양한 패치를 효율적으로 관리할 수 있는 범용적 패치관리 데이터베이스를 제안한다.

1. 서 론

소프트웨어가 증가하면서 프로그램상의 오류를 해결하기 위한 패치가 증가하게 되었고, 급수적으로 증가하는 패치를 안전하고 효율적으로 관리하기가 어려워 졌다. 이런 어려움을 해결하기 위해서 많은 패치관련 연구가 진행되어 왔으며, 기존의 연구는 안전한 패치시스템 구축과 패치분배에 집중적으로 연구되어 왔다[1,2,3,4,5,6]. 이런 연구결과로 패치분배는 패치 배포에서 발생될 수 있는 위협으로부터 보호가 가능해 졌다.

이와 같이 안전한 패치분배에 관한 연구는 진행되고 있지만, 기하 급수적으로 늘어나는 패치를 효율적으로 관리하는 연구는 미비한 실정이다. 안전한 패치 배포보다도 중요한 것이 패치관리이며, 본 논문은 패치관리시스템에서 다양한 패치를 범용적으로 관리할 수 있는 방안과 효과적으로 데이터데이스 구성할 수 방안에 대해서 한다.

2. 관련연구

본 장에서는 효율적이고 범용적인 패치관리 데이터베이스 설계를 위하여 국내외 패치관리시스템과 기존의 패치관련 연구에 대해서 분석한다.

2.1 국내외 패치관리시스템 분석

아래 <표1>과 같이 국내외 패치관리시스템은 대부분 특정 운영체제의 패치만을 관리하고 있다.

이런 패치관리시스템은 특정 운영체제의 패치관리에는 효율적이지만, 다양한 운영체제를 사용하고 있는 환경에서는 확장성과 경제성, 관리적인 면에서 매우 부적절하다. 다양한 운영체제를 지원하기 위해서는 각 운영체제마다 필요한 패치관리시스템을 확보해야 하며, 이는 추가적인 비용과 많은 관리시스템이 요구되기 때문에 적절하지 못하다. 또한, 패치관리시스템이 다양한 운영체제를 지원한다고 해도 운영체제 별로 패치 소프트웨어를 관리하기 때문에 추가적인 패치의 확정성과 범용성이 고려되지 않는 단점이 있다.

<표1> 국내외 패치관리시스템의 주요 기능

솔루션 / 기능	Patch Link	Ecora Software	ECM	ESM	Purdue Univ.	DOE SafePatch
다중 OS 지원	○	x	x	x	x	x
클라이언트 스캐닝	○	x	○	○	x	x
암호화전송	○	x	x	○	x	○
시스템 상태 알림	○	○	○	○	x	x
패치자동화	○	○	○	x		x
그룹화	○	○	○	x	○	x
스케줄	x	○	○	x	x	x
패치파일 암호화	○	x	x	○	○	○

"본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2006-(C1090-0603-0025))

2.2 패치관련 연구동향

기존의 패치관련 연구는 크게 보안패치 분배, 패치관리시스템 구성, 패치정보 수집에 관한 연구로 나눌 수 있으며 내용은 다음과 같다.

① 보안패치 분배에 관한 연구

네트워크상에서 암호화 통신 방법을 사용하여 안전하게 패치를 배포하는 방법에 관한 연구이다[3,5]. 데이터 전송의 기밀성, 무결성, 부인방지를 확보함으로써 패치분배가 기존의 비암호화 통신을 이용한 방법보다 안전하게 이루어 질 수 있도록 했다.

② 패치관리시스템 구성에 관한 연구

다양한 운영체제에서의 효율적인 패치시스템 구성에 관한 연구이다. 효율적인 패치관리를 위해서 관리의 대상이 되는 호스트에 에이전트 소프트웨어를 설치하여 관리하며, 분산환경을 고려한 효과적인 분배시스템을 설계하고 구성했다[1,4,6].

③ 패치정보 수집에 관한 연구

자동화된 패치정보 수집에 관한 연구로서, 소프트웨어 벤더에서 제공하는 패치정보의 데이터를 수집하고 분석하여 자동으로 데이터베이스를 구성하는 방안이다[2,7].

위와 같이 패치분배와 패치관리시스템 구성에 대해서는 연구되었지만, 범용적 운영체제 및 범용 소프트웨어에 적용 가능한 패치를 효율적으로 관리하는 방법에 관한 연구는 부족하다. 따라서, 본 논문은 확장성과 범용성을 고려한 패치관리 방법에 대해서 연구하였다.

3. 효율적이고 범용적인 데이터베이스 구성 방법

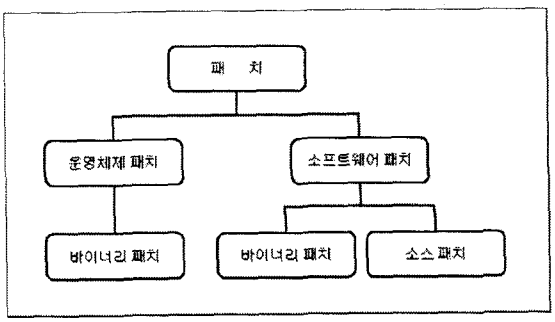
범용적이고 효율적인 패치관리 데이터베이스를 구성하기 위해서는 패치의 분류와 소프트웨어 정보의 구성요소 분석이 필수적이다. 본 논문은 운영체제 별로 운영체제 패치와 소프트웨어 패치를 나누어서 분석했으며, 대상 운영체제로서 MS의 운영체제 및 Linux, Unix 운영체제로 나누어 <표2>와 같이 분석했다.

<표2> 운영체제 분류와 파일 확장자

구 분	운영체제	파일 확장자
MS운영체제	Windows 2000, Windows XP, Windows 2003	exe, msi 파일
Unix	Solaris9, 10	deb, tar.gz 파일
Linux	Redhat Enterprise Linux 3,4	rpm, tar.gz 파일

3.1 패치분류

패치는 크게 운영체제 패치와 소프트웨어 패치로 구분하였으며, 패치배포 및 설치 방법에 따라 세부적으로 구분하였다. 각 분류 항목의 내용은 아래 [그림1]와 같다.



[그림1] 패치 분류

① 운영체제 패치

운영체제 패치는 바이너리 형태의 파일로 되어 있다. 운영체제의 패치 버전번호 정의는 소프트웨어 패치와 다르게 패치 벤더에서 자체적으로 정의하기 때문에 범용적 데이터베이스 설계 시 고려되어야 한다.

② 소프트웨어 패치

소프트웨어 패치는 다양한 형태 나눌 수 있으며, 크게 소스 패치, 바이너리 패치로 구분할 수 있다.

• 바이너리 패치

패치 파일이 바이너리 형태로 되어 있으며, 패치 대상 소프트웨어에서 패치가 필요한 바이너리 부분만 치환하는 방법으로 패치가 이루어 진다. 운영체제에 의존적이고 바이너리 형태로 되어 있는 소프트웨어만 가능하다는 단점이 있다. 하지만, 패치가 용이하다는 이점으로 현재 대부분의 시스템에서 사용되는 방법이다.

• 소스 패치

패치 파일이 소스코드 형태로 되어 있으며, 기존에 설치되어 있는 소프트웨어 중 패치 대상 파일을 컴파일 후 치환하는 방법으로 패치가 이루어 진다. 멀티 운영체제 지원이 가능하지만, 시스템의 하드웨어 및 운영체제에 종속적이기 때문에 현실적으로 설치가 어렵다.

3.2 소프트웨어 정보의 관리 및 구성요소 분석

소프트웨어 정보는 <표3>과 같이 다양한 구성요소들로 이루어져 있다. 이러한 요소들을 이용하여 패치 대상시스템의 소프트웨어를 구분할 수 있고 패치 적용여부도 확인할 수 있다. 소프트웨어 정보 구성요소 및 관리는 각 운영체제마다 다르며 정보수집 방법 또한 다르다. 따라서, 각 운영체제 별로 소프트웨어 정보와

관리 방법을 분석하여 범용적인 패치관리 데이터베이스 구성요소를 결정하고 구성하였다.

먼저, 운영체제 별로 소프트웨어 정보를 관리하는 방법에 대해서 설명하고 소프트웨어 구성요소에 대해 표현했다.

① 운영체제 별 소프트웨어 정보 관리 방법

• MS운영체제 계열

MS운영체제 계열의 소프트웨어 정보는 레지스트리 및 실행파일의 등록정보에서 정보를 수집할 수 있다. MS운영체제 계열은 운영체제 패치정보와 소프트웨어 정보를 별도로 관리하며, 제공되는 소프트웨어 정보는 다른 플랫폼에 비해 단순하다.

• Solaris의 계열

Solaris 계열의 소프트웨어 정보는 /var/sadm/pkg에 파일로 저장되어 있으며, 다양한 소프트웨어 정보와 패치 적용 여부를 확인할 수 있는 여러 요소를 나타낸다.

• Linux 계열

Linux계열의 소프트웨어 정보는 바이너리 형태로 /var/lib/rpm/Packages에 저장되어 있으며, 별도의 데이터베이스를 가지고 있다. 다른 운영체제에 비해 소프트웨어 별 버전관리가 가능하다.

② 소프트웨어 정보 구성요소

각 운영체제 별로 소프트웨어 정보의 구성요소들은 기본적인 요소를 포함하고 있다. 범용적인 패치관리 데이터베이스를 구성하기 위해서는 범용적 소프트웨어 정보의 구성요소를 정의해야 한다. 아래 <표3>와 같이 운영체제에서 분석이 가능한 소프트웨어의 구성요소들을 표현하고, 운영체제 별로 구성요소의 공통점을 확인하였다.

<표3> 소프트웨어 정보 구성요소

구분	구성요소	windows	Unix	Linux
소프트웨어 정보 구성요소	S/W 이름	○	○	○
	S/W 파일명	○	○	○
	S/W 버전	○	○	○
	S/W 패치버전	x	x	○
	S/W 추가설명	x	○	x
	S/W 제공 벤더명	○	x	○
부가적인 구성요소	S/W 무결성 정보	x	x	○
	S/W 파일경로	○	○	○
	S/W 설치시간	○	○	○

3.3 범용적인 패치관리 데이터베이스 설계

본 논문은 패치분류와 소프트웨어 구성요소를 이용하여 다양한 운영체제 및 소프트웨어를 위한 패치관리시스템에서 사용할 수 있는 효과적인 패치관리 데이터베이스를 설계하였다.

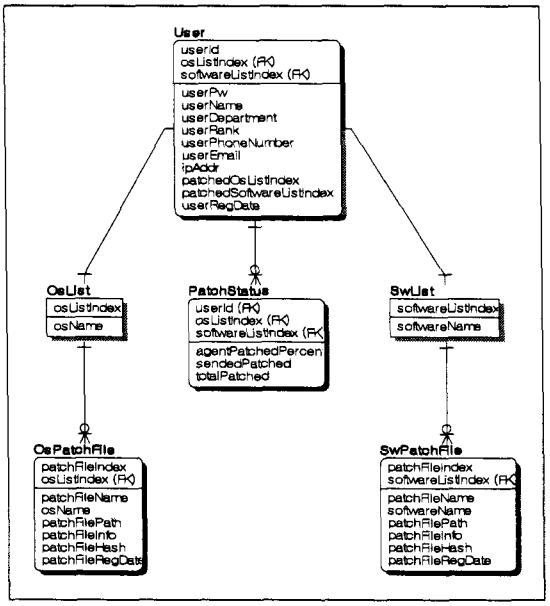
데이터베이스는 크게 운영체제 패치와 소프트웨어 패치로 구성하였다. 운영체제의 패치는 각 운영체제 벤더에서 제공되는 패치를 관리할 수 있도록 구성하였으며, 소프트웨어 패치는 운영체제 패치에 독립적으로 관리할 수 있도록 구성하였다. 위와 같은 구성으로 신규 운영체제나 소프트웨어가 등록이 되더라도 확장성을 고려한 범용적 패치관리가 이루어 질 수 있다. 다음 <표4>는 범용적 패치관리 데이터베이스의 테이블과 필드 항목을 나타낸 것이다. <표4>의 OS Patch File List 테이블과 S/W Patch File List 테이블을 보면, 각 소프트웨어마다 독립적으로 패치파일을 관리할 수 있는 것을 확인할 수 있다.

[표4] 범용적 패치관리 데이터베이스 테이블

테이블명	필드명	필드 설명
OS patch file	patchFileIndex	OS 패치 파일의 인덱스
	patchFileName	OS 패치 파일의 이름
	patchFilePath	OS 패치 파일 저장 경로
	patchFileInfo	OS 패치 내용 정보
	patchFileRegDate	OS 패치 파일의 등록날짜
	patchFileHash	OS 패치 파일의 해쉬값
	osName	OS 이름
S/W patch file	patchFileIndex	S/W 패치 파일의 이름
	patchFileName	S/W 패치 파일의 이름
	softwareName	S/W 이름
	patchFilePath	S/W 패치 파일 저장 경로
	patchFileInfo	S/W 패치 내용 정보
	patchFileRegDate	S/W 패치 파일의 등록날짜
	patchFileHash	S/W 패치 파일의 해쉬값
OS Patch File List	osListIndex	OS 패치 파일의 인덱스
	osName	OS 패치 파일의 이름
S/W Patch File List	softwareListIndex	S/W 패치 파일의 인덱스
	softwareName	S/W 패치 파일의 이름

[그림2]는 <표4>의 테이블간의 관계도를 나타낸 것으로 User테이블과 PatchStatus테이블은 범용적 패치관리시스템을 구성할 때 사용될 수 있는 테이블을

나타낸 것이다. 아래 [그림2]와 같이 데이터베이스 테이블을 이용하면 범용적 운영체제 및 소프트웨어 관리가 가능하다.



[그림2] 테이블 관계도(ER-Diagram)

4. 결론 및 향후 연구계획

본 논문에서는 패치를 보다 효율적으로 관리할 수 있도록 범용적 패치관리 데이터베이스 제안하고 구성하였다. 운영체제 패치와 소프트웨어 패치를 독립적으로 분리함으로써 범용적인 패치관리가 이루어질 수 있도록 하였으며, 소프트웨어 패치도 각 소프트웨어별로 패치를 관리할 수 있도록 데이터베이스를 구성하였다. 이로서 운영체제 및 소프트웨어에 대하여 독립적으로 패치를 관리 할 수 있다.

본 데이터베이스 구성을 이용하여 실제 패치관리시스템을 효과적으로 구성할 수 있으며, 향후 기존의 암호화 통신을 이용한 보안패치 구성에 대해 추가적으로 연구가 되어야 한다.

5. 참고문헌

[1] Taeshik Shon, Jongsub Moon, Cheolwon Lee, Eul-Gyu Im, Jung-Taek Seo, "Safe Patch Distribution Architecture in Intranet Environments", Security and Management 2003, pp. 455~460, 2003

[2] 이상원, 김윤주, 문종섭, 서정택, "멀티플랫폼 환경에서의 보안패치 분배를 위한 DB구축 및 검색 방법에 관한 연구", 한국정보과학회 학술발표논문집, 제31권, 제1호(A), pp. 337~339, 2004

[3] 이상원, 김윤주, 문종섭, 서정택, 최대식, 박응기, "RMI와 SSL를 이용한 멀티플랫폼 환경에서의 안전한 보안패치 분배 시스템 설계", 한국정보과학회 학술발표논문집, 제31권, 제1호(A), pp. 283~285, 2004

[4] Jung-Taek Seo, Dae-Sik Choi, Eung-Ki Park, Tae-Shik Shon, Jongsub Moon, "A Secure Patch Distribution Architecture", ISDA 2003, Lecture Notes in Computer Science, Springer-Verlag, pp. 654~661, 2003

[5] 손태식, 김진원, 박일곤, 문종섭, 서정택, 임을규, 이철원, "안전한 패치 분배 구조 설계", 한국정보과학회 학술발표논문집, 제29권, 제2호(I), pp. 559~561, 2002

[6] 이상원, 김윤주, 손태식, 문종섭, 서정택, 이은영, 이도훈, "일반화된 보안패치 분배 및 관리 시스템을 위한 프레임워크 설계", 한국정보과학회 학술발표논문집, 제31권, 제2호(I), pp. 502~504, 2004

[7] 민동옥, 손태식, 서정택, 구원본, 장정아, 문종섭 "보안패치 자동분배를 위한 패치 DB 자동구성 방안", 한국정보과학회 학술발표논문집, 제31권, 제1호(A), pp. 367~369, 2004