패치관리시스템의 효율적인 구성요소에 관한 연구

*이인용 *이수영 *문종섭 *임종인

고려대학교 정보경영공학전문대학원

*{iylee, leesuyoung, jsmoon, jilim}@korea.ac.kr

A Study on Efficient Component In Patch Manangement System

*Lee, In-Yong *Lee, Su-Young *Moon, Jong-Sub *Lim, Jong-In

Korea University Graduate School of Information Management & Security

요약

컴퓨터가 대중화 되면서 다양한 소프트웨어에 대한 수요가 증가하게 되었고, 많은 소프트웨어들이 단시간에 개발되어지고 있다. 이런 이유로 많은 소프트웨어들에 대한 취약점들이 생겨나게 되었고, 이를 해결하기 위해서 소프트웨어 벤더들은 패치를 만들고 배포를 하고 있다. 하지만, 다양한 시스템과 소프트웨어를 관리하는 곳에서 일일이 패치를 벤더로부터 받아 대상시스템에 설치하고 관리하기에는 어려움이 많으며, 일괄적이고 통합적인 방법이 필요하다. 이런 문제와 요구를 해결하기 위해 패치관리시스템에 관한 많은 연구들이 진행되어 왔으며 상용제품들도 하나둘씩 개발되고 있다. 하지만, 안타깝게도 많은 패치관련 연구들이 안전한 패치관리시스템 설계나 구성에 관해서만 연구되고 진행되어 왔다. 안전한 패치관리시스템을 설계하거나 구성하기 전에 무엇보다도 우선시 되어야 하는 것이 필수적인 패치관리시스템 구성요소들을 정의하는 것이며, 이와 관련된 표준이나 연구가 많이 부족하다. 따라서 본 논문은 패치관리시스템을 구성하기 위한 기본적이면서 필수적인 구성요소들을 고려하고 정의했으며, 이를 바탕으로 기본 패치관리시스템 프레임워크를 설계했다.

1. 서론

소프트웨어 기술과 인터넷의 발전으로 인하여 다양한 응용 프로그램들이 생겨났고, 이를 노리는 악의적인 공격 또한 많이 증가하였다. 현재의 공격들은 과거 90년대와 다르게 더욱 정교해졌으며, 누구나 쉽게 사용할 수 있도록 자동화되고 지능화되었다. 뿐만 아니라 Zero-Day 공격과 같은 신속하고 위협적인 공격으로 인하여 이를 대처하기는 더욱 어려워졌으며, 이로 인한 피해는 과거에 비해 꾸준히 증가하는 추세이다.

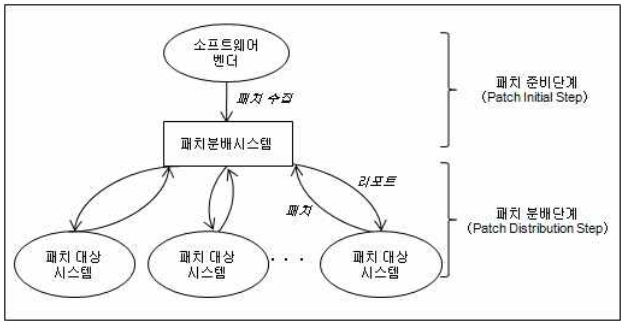
현재 다양한 공격으로부터의 피해를 사전에 예방하기 위한 연구들이 여러 분야에서 진행되고 있으며, 최근 패치에 대한 중요성이 강조되면서 패치관리시스템에 대한 연구들이 많은 진행되고 있다. 패치관리시스템에 관한 연구는 크게 보안패치에 관한 연구[1,2]와 패치정보관리에 관한 연구[2,3], 패치분배 프레임워크에 관한 연구[3,4,5,6,7]로 나눌 수 있으며, 이런 다양한 연구들로 인하여 현재 많은 곳에서 패치관리시스템들을 개발하고 적용시키고 있다.

위와 같은 연구들로 인하여 패치관리시스템에 대한 기반 연구들은 많은 진척을 보였지만, 아쉽게도 패치관리시스템에 꼭 필요한 구성요소들은 무엇이며, 패치관리시스템 구성 시 고려되어야 할 사항들에 대해서는 분석과 연구가 부족한 실정이다. 따라서 본 논문에서는 기존의 패치관련 연구들에 대해서 분석하고, 패치관리시스템 구성에 있어서 기본적이면서 필수적인 요소들과 고려되어야 할 사항들에 대해서 정리하였다. 본 논문의 2장에서는 패치관리시스템 구조에 대해서 분석하고, 패치관리시스템에 필수적인 구성요소들이 무엇인지 정리하였다. 3장에서는 패치관리시스템 구성요소를 구성할 시 고려되어야 할 사항들에 대해서 언급하고 이를 바탕으로 기본 패치관리시스템 프레임워크를 설계하였다.

2. 패치관리시스템

2.1 패치관리시스템 구조

패치관리시스템은 논리적 또는 물리적인 특정 그룹으로 나눈 호스트들의 모든 소프트웨어 버그를 자동으로 검색하고 패치해 주는 시스템을 말한다. 관리하는 시스템들의 모든 소프트웨어를 검색하고 패치하다는 점에서 단일 소프트웨어 벤더에서 제공하는 자동패치시스템하고는 개념적으로 다르다.



[그림1] 패치관리시스템 구조

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음.

(IITA-2006-(C1090-0603-0025))

패치관리시스템은 패치대상시스템들의 다양한 운영체제와 소프트웨어 패치들을 관리한다. 일반적인 패치관리시스템 구조는 [그림1]과 같다.[3,4,5,6,7,8]

[그림1]과 같이 패치관리시스템을 패치 준비단계와 패치 분배단계로 나누며 각 단계별 세부 구성단계는 다음과 같다.

2.1.1 패치 준비단계(Patch Initial Step)

① 패치수집 단계

패치수집 단계는 패치관리시스템이 관리하는 소프트웨어들의 신규 패치들을 벤더로부터 받는 단계이다. 벤더로부터 받는 패치는 보안채널이 형성된 통신환경에서 받거나, 직접 물리적 저장매체(디스켓, CD등)를 통해 받는다.[9,10]

② 패치확인 및 검증 단계

패치확인 및 검증 단계는 벤더로부터 받은 패치들의 안전성 및 무결성을 검증하고 유의사항을 확인하는 단계이다.[2,3,9,10] 이 단계는 정확하고 조심스럽게 진행되어야 하며, 이 단계가 중요한 이유는 3장에서 언급하겠다.

③ 패치대상시스템 확인 단계

패치대상시스템 확인 단계는 신규 패치가 적용될 대상시스템들을 확인하여 패치대상 목록을 만드는 단계이다. 패치관리시스템은 패치대상시스템의 Footprint(시스템 정보, 소프트웨어 정보, 레지스트리 정보 등)을 확인하여 패치대상 목록을 만든다.[9,10,11] 이 목록을 통하여 패치대상시스템과 패치를 선정하여 분배를 한다.

2.1.2 패치 분배단계(Patch Distribution Step)

① 패치스케줄링 단계

패치 스케줄링 단계는 특정 시간(월, 일, 시간)이나 특정 단계(긴급, 중요, 보통 등)에 따라 패치를 분배하는 단계이다.[9,10] 패치 스케줄링은 여러 가지 조건에 따라 자동으로 패치가 분배될 수 있도록 구성할 수 있기 때문에 없어서는 안되는 단계이다.

② 패치설치 단계

패치설치 단계는 패치관리시스템에서 분배한 패치를 패치대상시스템에서 설치하는 단계이다.[11,12,13] 자동설치 프로그램으로 설치하며, 설치 오류 시 이전상태로 복구될 수 있도록 roll-back 기능을 가지고 있다.

③ 패치리포트 단계

패치대상시스템에 설치한 패치 정보를 수집하여 새로운 Footprint를 만들고 이를 패치관리시스템에 전송한다.[11,13] 패치관리시스템은 전송받은 Footprint정보를 분석하여 관리자에게 최종적으로 패치현황을 제공한다.

2.2 패치관리시스템 구성 요소

여러 패치관리시스템 관련 연구에서 요구하고 제안하는 구성 요소들은 매우 다양하지만, 기본적으로 요구되는 구성 요소에 대해서는 모두 공통적이다.[3,4,5,9,11,12,13] 기본적인 공통 사항들을 분석하고 일반화하여 구성한 것이 [표1]과 같으며, 구성요소 별 기능은 다음과 같다.

<표1> 패치관리시스템 구성 요소

구성요소	세 부 기 능
정보수집	패치/정보수집, 패치 목록화
검증	패치 안정성 검증, 패치 무결성 검증
스케줄링	정책관리, 예약관리
인증	사용자 인증, 시스템 인증
분배	암호화, 그룹분배
설치	Footprint, 리포트, Roll-back

① 정보수집

벤더로부터 패치파일 및 관련 정보들을 얻는 기능이다. 관련 정보에는 취약점 정보, 패치파일 CVE코드, 유의사항 등의 정보가 있다.

② 검증

벤더로부터 받은 패치파일을 검증하는 기능으로 패치파일 무결성 검증과 안정성 검증으로 나눈다. 무결성 검증은 패치파일 자체에 대한 검증이며, 안전성 검증은 패치파일을 대상시스템에 설치하기 전에 안전상에 문제가 없는지에 대한 검증이다. 안정성 검증은 패치를 분배하기 전에 대상시스템과 유사한 환경의 가상시스템에서 검증을 한다.

③ 스케줄링

스케줄링은 패치관리시스템에 설정된 정책을 기반으로 특정 조건이나 시간에 따라 이루어진다. 어떻게 스케줄러를 구현하는냐에 따라 패치관리시스템의 성능이 좌우된다.

④ 인증

인증은 시스템 인증과 사용자 인증으로 나누어진다. 시스템 인증은 패치대상시스템이 패치관리시스템에서 관리하는 시스템인지 인증하는 것이며, 사용자 인증은 권한에 따라 패치파일을 다르게 분배하기 위해서 사용되는 인증이다.

⑤ 분배

패치를 암호화해서 분배하는 기능으로 패치대상시스템을 그룹화하여 분배한다. 그룹은 논리적 그룹(Logical Group)과 물리적 그룹(Physical Group)으로 나누거나 서로 병합하여 사용할 수 있다. 물리적 그룹은 패치대상시스템을 지리적 또는 물리적 특성에 따라 그룹을 나눈 것이며, 논리적 그룹은 S/W별 또는 부서별과 같은 일정한 규칙에 의해 논리적으로 그룹을 나눈 것이다. 그룹화는 패치 분배와 관리의 효율성을 위해서 필요하다.

암호화는 비암호화 분배상황에서 발생될 수 있는 취약점 노출을 예방하기 위해서 필요한 기능이다.

⑥ 설치

Footprint기능은 패치를 설치하기 전에 패치대상시스템에 대한 정확한 정보(레지스터 정보, 시스템 정보, 패치정보 등)를 수집하여 목록화하는 기능이다. Footprint의 정확한 정보를 이용하여 패

치를 설치하거나 설치시 문제가 있는 패치에 대해서는 Roll-back을 한다. 설치가 완료되면 Footprint을 갱신하고 설치에 대한 리포트를 작성하여 패치관리시스템에 전송한다.

3. 패치관리시스템 프레임워크

본 3장에서 2장에서 분류한 구성요소들 중에서 반듯이 고려되어야 할 사항들에 대해서 언급하고, 분류한 구성요소들을 이용하여 패치관리시스템 프레임워크를 설계한다. 본 논문에서 제안하는 패치관리시스템 프레임워크는 패치관리시스템을 구성하고 구현할 때 으로 기본적으로 필요한 기능에 대해서 일반화하여 구성한 프레임워크이다.

3.1 패치관리시스템 구성요소 고려사항

① 정보수집

패치정보 수집은 벤더와 패치관리시스템사이에 기밀성, 무결성, 보안성이 확보된 환경에서 이루어져야 한다.[1,2] 위 사항들이 확보되지 않는 상황에서 패치 정보를 수집하여 분배하는 것은 패치관리시스템뿐만 아니라 패치대상시스템에게도 커다란 위험을 초래할 수가 있다. 한 예로 안전하지 않는 환경에서 벤더로부터 악성코드에 감염된 패치를 받아 분배할 경우 그 피해는 매우 크다. 따라서 패치관리시스템과 벤더사이에는 항상 안전성이 확보된 환경이 조성되어 있어야 한다.

② 패치 검증

패치 검증에서 조심스럽게 검증되어야 할 것이 패치 안전성 및 호환성 검증이다. 안전성 및 호환성 검증은 벤더에서 이루어지는 것이 일반적이지만, 패치대상시스템들의 시스템 환경들은 매우 다양하기 때문에 금융, 국방, 교통망 시스템과 같은 민감한 데이터를 취급하는 시스템들에 대한 검증은 패치관리시스템에서 별도로 직접 안전성 및 호환성을 검증해야 한다.[10,11]

운영체제의 커널이나 민감한 시스템 설정파일을 수정하는 패치인 경우, 안전성 및 호환성 검증없이 패치를 설치하면 안전한 시스템 설정이 변경되거나 시스템이 멈출 수 있는 요지가 생긴다.

패치 안전성 및 호환성 검증은 민감한 데이터를 취급하는 곳에서는 반드시 필요하며, 테스트 비용을 절감하기 위해서 가상머신을 이용한 테스트가 많이 이루어지고 있다.

③ 스케줄링

신규패치를 벤더로부터 받아 바로 분배하여 적용하는 것보다는 스케줄링을 통하여 적절한 시기에 패치를 분배하고 적용해야 한다.[8,9,11] 패치적용 후 시스템 재시작이 요구되는 패치를 중요한 작업을 하고 있는 시스템에 바로 분배하여 적용시키는 것은 좋지 않다. 패치대상시스템의 자원을 사용하지 않을 때나 중요한 작업이 없는 적절한 시간 때에 스케줄링을 통하여 패치를 적용하는 것이 패치관리시스템에서는 중요하며 반드시 고려되어야 할 사항이다.

④ 인증

패치관리시스템으로부터 분배되는 패치 중 시스템에 민감한 패치가 있기 때문에 패치 종류와 등급에 따라 적절한 사용자 권한이 가진 사람에게 패치가 이루어지도록 해야 한다.[1,2,11] 사용자별 권한에 대한 적절한 인증없이 패치가 이루어진다면, 시스템에 치명적인 위험을 줄 수 있다. 한 예로 공동으로 작업하는 시스템에서 시스템 인증으로만 인증하여 패치가 이루어진다면 시스템에서 다른 작업을 하고 있는 다른 사용자에게 피해를 줄 수가 있다. 이런 문제를 예방하기 위해서는 사용잘 별 인증과 패치 적용에 대한 적절한 권한이 반드시 필요하다.

또한, 패치관리시스템은 Laptop이나 Wireless를 사용하는 사용자를 위한 인증 메커니즘도 제공하여 한다.

⑤ 분배

신속한 패치분배를 위해서는 패치관리시스템과 패치대상시스템 사이에 분배서버를 두는 것이 효율적이다.[3,4,5,6,7] 분배서버를 통하여 확장성을 확보할 수 있을 뿐만 아니라 시스템 및 네트워크 트래픽 부하분산을 위해서도 효율적이다. 분배서버는 물리적 그룹과 논리적 그룹에 따라 적절하게 배치하고 구성하는 것이 좋다.

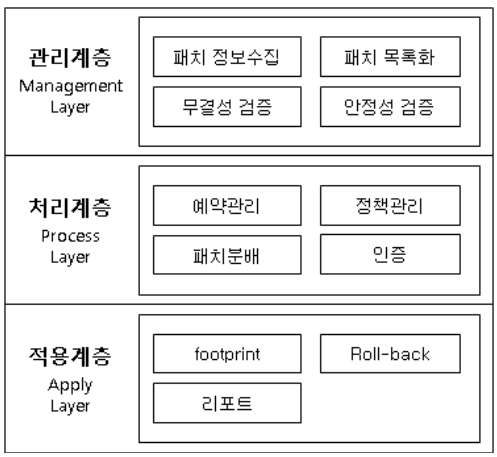
⑥ 설치

패치 설치에 있어서 고려되어야 할 사항은 패치 설치 중에 다른 긴급한 작업이나 다른 프로세스의 인터럽트로 인하여 도중에 패치 적용이 중단되어도 이를 체크하고 이전 환경으로 roll-back될 수 있어야 한다.[7,9,11] 위와 같은 상황에서 roll-back기능이 없다면 패치대상시스템에 치명적 위험을 줄 수 있으며, 이는 패치 설치 프로그램을 구현할 때 반드시 고려되어 구현되어야 한다.

뿐만 아니라 패치 적용 후 시스템 재시작이 요구되는 경우 사용자의 동의 없이 어떠한 경우에도 재시작이 이루어지면 안된다. 장시간 작업이 요구되는 프로세스가 작동하고 있는 경우 사용자 동의 없이 시스템 재시작이 이루어진다면 사용자에게는 매우 치명적이다. MS windows의 SUS의 경우 주요 패치를 설치한 후 지속적으로 특정 시간 내에 사용자의 재부팅을 요구하는데, 이는 사용자가 시스템을 사용하지 않고 자리에 없을 경우 재시작이 될 요지가 있으며, 실제로 이로 인한 피해가 종종 발생되고 있다.

3.2 패치관리시스템 프레임워크

2장에서 정의하고 분류한 구성요소와 고려사항들을 기반으로 기본 패치관리시스템 프레임워크를 [그림2]와 같이 설계했다.



[그림2] 패치관리시스템 프레임워크

[그림2]의 패치관리시스템 프레임워크는 패치관리시스템에서 기본적이면서 필수적으로 필요한 구성요소와 기능들을 표현하고 있다.

기본 패치관리시스템 프레임워크에 있는 구성요소들을 기반으로 확장된 패치관리시스템 프레임워크를 설계하고 패치관리시스템을 구성할 수 있다.

4. 결론

본 논문에서는 기존에 연구된 여러 패치관련 논문들을 분석하고 이를 바탕으로 패치관리시스템에서 기본적이면서 필수적인 구성요소들에 대해서 정의하였다. 또한, 패치관리시스템에 구성요소들을 구성할 때 반드시 고려되어야 할 사항들에 대해서 언급했으며, 이를 바탕으로 기본 패치관리시스템 프레임워크를 설계하였다.

본 논문에서 설계한 기본 패치관리시스템 프레임워크에는 패치관리시스템에 기본적이면서 필수적인 요소들로 구성되어 있으며, 구성요소들을 구성할 때 고려되어야 할 사항들이 포함되어 있다. 비록, 본 논문에서 제안한 구성요소들이 표준은 아니지만 이런 구성요소들을 바탕으로 다른 구성요소들을 추가하여 새로운 패치관리시스템 프레임워크를 설계한다면, 보다 안전하고 효율적인 패치관리시스템을 구성할 수 있다.

5. 참고문헌

[1] 이상원(Sangwon Lee), 김윤주(Yun-Ju Kim), 손태식(Tae-Shik Sohn), 문종섭(Jong-Sub Moon), 서정택(Jung-Taek Seo), 이은영(Eun-Yong Lee), 이도훈(Do-Hoon Lee), “일반화된 보안패치 분배 및 관리 시스템을 위한 프레임워크 설계”, 한국정보과학회 2004년도 가을 학술발표논문집 제31권 제2호(Ⅰ), pp. 502~504, 2004. 10

[2] 김윤주(YunJu Kim), 이상원(SangWon Lee), 손태식(Tae-Shik Sohn), 문종섭(Jong-Sub Moon), 서정택(JungTaek Seo), 이은영(Eun-Yong Lee), 박응기(EungKi Park), “XML을 이용한 보안패치 중앙 관리 시스템 설계”, 한국정보과학회 2004년도 가을 학술발표논문집 제31권 제2호(Ⅰ), pp. 505~507 , 2004. 10

[3] 민동옥, 손태식, 서정택, 구원본, 장정아, 문종섭, “보안패치 자동분배를 위한 패치 DB 자동구성 방안”, 한국정보과학회 학술발표논문집, 제 31권, 제1호(A), pp367~369, 2004

[4] 손태식, 김진원, 박일곤, 문종섭, 서정택, 임을규, 이철원, “안전한 패치분배시스템 구조 설계”, 한국정보과학회 학술발표논문집, 제 29권, 제2호(I), pp. 559~561, 2002

[5] 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

[6] 김윤주(YunJu Kim), 이상원(SangWon Lee), 손태식(Tae-Shik Sohn), 문종섭(Jong-Sub Moon), 서정택(JungTaek Seo), 윤주범(JuBum Yun), 박응기(EungKi Park), “확장성을 고려한 계층적 패치 분배 시스템 프레임워크 설계”, 한국정보과학회 2004년도 봄 학술발표논문집 제31권 제1호(A),pp. 199~201, 2004. 4

[7] Chuan-Wen Chaang, Dwen-Ren Tsai, Jui-Mi Tsai, "A cross-site patch management model and architecture design for large scale heterogeneous environment" Security Technology 2005. CCST '05 39th, IEEE, pp.41~46, 2005

[8] Taeshik Shon, Jongsub Moon, Cheolwon Lee, EulGyu Im, JungTaek Seo, “Safe Patch Distribution Architecture in Intranet Environments", Security and Management 2003, pp. 455~460, 2003

[9] Brykczynski B, Small. R. A, "Reducing Internet-based intrusions: Effective security patch management" Software, IEEE, pp. 50-57, 2003

[10] Brandman, G., "Patching the enterprise", Queue ACM, PP32-39, Mar, 2005

[11] Dadzie, Understanding software patching, Queue ACM, pp 24-30, Mar. 2005

[12] Greg Hoglund, Gray McGraw, “EXPLOITING SOFTWARE: How to break code”, Addision Wesley Professional, pp. 44~53, 2004

[13] Chen, Haibo Yu, Jie Chen, Rong Zang, Binyu Yew, Pen-Chung, "POLUS: A POwerful Live Updating System" Software Engineering, 2007. ICSE 2007. 29th International Conference, IEEE, pp 271-281, May 2007