XML기반 윈도우 보안패치 자동관리 시스템

박정진[O*] 박진섭[*] 신영선[*] 김봉회[**]
*대전대학교 컴퓨터공학과, **(주)유비엔씨
parkjj@ubnc.net[O*] jspark@dju.ac.kr[*] ysshin@zeus.dju.ac.kr[*] kbh@ubnc.net[**]

Windows Security Patch Auto-Management System Based on XML

Jungjin Park[O*] Jinsub Park[*], Youngsun Shin[*] Daejeon University. Bonghoi Kim[**] UBNC Corp.

요 약

최근 윈도우의 취약점을 이용하는 웜 및 바이러스로 인한 정보시스템과 네트워크에 대한 피해가 급속히 증가하고 있다. 윈도우의 취약점을 이용한 공격에 대한 해결방법은 해당 취약점은 적시에 신속하게 패치를 설치하는 것이다. 본 논문에서는 기존의 패치관리시스템이 관리자의 개입을 요구하기 때문에 패치를 신속하게 적용할 수 없는 단점을 보완하여, XML 기술을 기반으로 MS 다운로드 센터에서 패치를 자동으로 다운받아 클라이언트까지 자동으로 보안패치파일을 설치하는 보안패치 자동관리 시스템을 제안한다.

1. 서 론

윈도우의 보안 취약점을 이용한 웜 및 바이러스로 인하여 사용자 시스템에 장애를 일으키고, 네트워크의 트래픽을 기하급수적으로 발생시켜 네트워크의 마비를 시키는 경우도 발생하고 있어 그에 대한 인적, 경제적 피해가 급증하고 있다.

이러한 윈도우의 보안 취약점을 이용한 악성 프로그램으로부터 시스템 및 네트워크를 보호하기 위하여 많은 보안 시스템 및 백신소프트웨어를 도입하여 운영하고 있다. 이러한 네트워크 보안 시스템 및 백신은 윈도우의 취약점을 근본적으로 해결해 주지 못하기 때문에 변종에 의해 계속적으로 감염이 될 수 있다.

윈도우의 보안 취약점을 근본적으로 해결하기 위해서는 해당 보안 패치프로그램을 적시에 신속하게 설치하는 것이다. 기존에 많은 패치관리 시스템들이 개발되어 보급되고 있다. 그럼에도 불구하고 패치관리 시스템의 비효율적인 관리로 인하여 취약점을 이용한 피해는 여전히 증가하고 있는 추세이다.

본 논문에서는 기존의 패치관리시스템들이 관리자의 개입을 필요로 하는 반자동적인 단점을 개선하여, 관리자의 개입이 없이도 관리대상 클라이언트에 자동으로 보안패치를 설치해주는 XML기반의 윈도우 보안패치 자동관리 시스템을 제안한다.

2. 국내외 패치관리 시스템 분석

2.1 패치관리 시스템의 기술 동향

국내외의 기존 패치관리시스템들의 특징은 패치관리시스템을 공급하는 벤더 또는 도입하는 사이트에서 발표되는 패치 파일을 서버에 등록을 해야 하는 과정을 거치고 있다. 이는 신속함을 필요로 하는 패치관리시스템에 있어서 악성코드를 신속하게 방어하기에 취약한 부분이라 할 수 있다.

가. Inciter

국내에서 개발된 시스템으로, 소프트웨어 설치 유도 기능을 적용하여 도입하는 사이트의 클라이언트에 패치 에이전트를 효과적으로 설치할 수 있는 기능을 보유하고 있다.

패치 파일을 배포하는 과정은 벤더가 MS로부터 패치 파일을 공급받아 IDC센터에 벤더가 운영하는 서버로 해당 패치 파일을 등록하는 과정을 거친다. 이렇게 등록된 패치 파일은 패치관리시스템을 도입한 사이트의 서버가 벤더운영 서버에 접속하여 패치 파일을 다운받고 패치가 필요한 사용자 클라이언트에 패치 파일을 설치하게 된다.

본 패치관리시스템은 벤더가 패치 파일을 등록하는 시간이 조금이라도 지연된다면 그만큼 도입한 사이트에도 신속하게 적용할 수 없는 단점을 가지고 있다.

나. PatchLink Update

본 제품은 해외 개발 시스템으로 특징은 윈도우 운영체제 외에도 리눅스, 유닉스 계열의 패치도 지원한다. 128Bit SSL(Secure Socket Layer)를 사용하여 보안성을 강조하였다. 또한 사용자가 패치설치 시점과 재부팅 시점을 선택하게 함으로써 유연하게 패치를 관리할 수 있도록 개발되었다. 또한 Active Directory와 LDAP으로의 손쉬운 통합이 가능하다.

본 패치관리시스템은 패치를 설치하는 행위에 대해 해당 시스템 사용자가 판단을 해야 하기 때문에 패치의 설치가 신속하게 이루어 지지 않는 경우를 발생시킨다.

3. XML기반 보안패치 자동관리 시스템

본 논문에서 제안하는 XML기반의 보안패치 자동 관리 시스템은 MS에서 제공하는 SUS(Software Update Service)를 연동하여 MS 다운로드 서버에서 도입사이트 관리 시스템을 거쳐 최종 클라이언트 시스템에 패치가 적용되기까지 벤더나 관리자의 개입이 없이 모든 과정이 자동으로 이루어지기 때문에 신속하고 효과적으로 윈도우의 보안 패치관리를 수행할 수 있다. 본 논문에서 제안하는 패치관리시스템의 대상 운영체제는 MS의 윈도우 2000/XP/2003을 대상으로 하고 있다.

제안하고자 하는 패치관리시스템에서는 데이터베이스의 장점인 추가/삭제, 검색의 편리함을 수용하면서 파일 기반 환경을 이용하여 각각의 클라이언트에게 분배하고 필요한 패치 여부를 확인할 수 있도록 검색하는 작업의 수행을 위해 XML 기술을 이용한다.

3.1 XML 기반 보안패치 자동관리 시스템의 구성

보안패치 자동관리 시스템은 MS의 다운로드 센터로부터 최신 패치 파일을 동기화하는 SUS서비스와 SUS서비스의 관리와 패치 대상 클라이언트의 관리와 패치 파일을 관리하는 패치관리 서버, 그리고 패치관리 서버로부터 패치 정보 및 패치를 클라이언트에 실제로 설치하는 패치에이전트로 구성된다.

가. SUS서비스

SUS서비스는 MS에서 제공하는 공개 소프트웨어로 패치관리 서버와 동일한 시스템에서 운영되며 설치되면 시스템 서비스로 백그라운드에서 자동 스케줄링에 의해 동작한다.

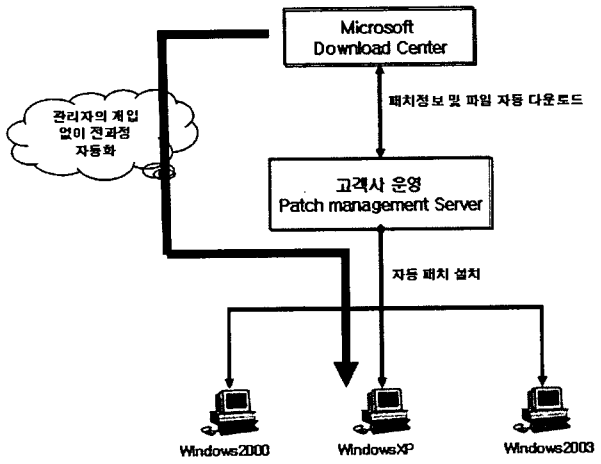
주요 기능은 MS 다운로드 센터로부터 운영체제 정보 및 패치 정보, 패치 파일을 스케줄링에 의해 주기적으로 다운받는다 [1].

나. 패치관리 서버

SUS가 다운받은 패치 정보 및 파일을 분석하여 XML 포맷의 패치 정보파일을 생성한다. XML형태의 패치정보 파일은 Internet Explorer에 해당하는 패치 정보 파일과 운영체제 그리고 그 외의 보안에 관련된 패치 정보 파일을 생성한다.

다. 패치 에이전트

패치 에이전트는 패치관리 서버에서 생성한 XML파일을 자신의 운영체제 종류 및 버전, internet Explorer 버전을 분석하여 클라이언트의 환경에 맞는 패치정보 XML파일을 전송받아 설치해야할 패치 정보를 분석하고, 패치관리 서버로부터 패치 파일을 다운로드받아 클라이언트에 설치하는 기능을 담당한다.



[그림 1] 시스템 구성도

3.2 기능 설계

가. 패치 정보 XML 생성

items.xml 파일은 크게 설치사실을 확인할 수 있는 정보 <installed> Tag와 설치배제대상 여부를 판단할 수 있는 정보 <excluded> Tag로 구성된다.

또한 <and>, <or>, <not> 연산자를 이용하여 2개씩 묶여지며 참, 거짓 여부를 종합적으로 판단하여 해당 패치가 설치할 필요가 있는지 여부를 결정할 수 있도록 지원한다.

detail.xml 파일은 패치 위험도 등급, 파일 크기 및 이름, 패치에 대한 제목 및 자세한 설명, 파일에 대한 CRC값, 설치 옵션 등의 정보를 가지고 있다.[2]

나. 윈도우 스캔 리스트 설계

윈도우 스캔리스트에서 item.xml은 [표 1]과 같이 크게 6가지의 확인방법을 제공한다.

[표 1] 스캔 리스트에서 지원하는 확인 정보

Tag	설 명
regKeyValue	레지스트리의 값의 존재 여부를 확인
regKeyExists	레지스트리 키의 존재 여부를 확인
regKeyVersion	레지스트리의 버전 값이 적당한 범위 내에 있는지 확인
regKeySubString	레지스트리의 값에 주어진 문자열이 존재하는지 확인
fileVersion	파일의 버전 값이 적당한 범위 내에 있는지 확인
fileExists	파일이 특정 위치에 존재하는지 확인

detail.xml은 패치에 대하여 [표 2]와 같이 자세한 정보를 담고 있다.

[표 2] detail.xml 파일의 필드 정보 종류

TAG	설 명
size	파일의 크기를 숫자로 표현
title	패치 번호 및 간략한 제목을 나타냄
descriptionText	패치에 대한 간략한 설명을 나타냄
priority	패치에 대한 등급을 나타냄
exclusive	단독 설치가 필요한지 여부를 나타냄
needsReboot	설치 후 시스템을 재시작할 필요가 있는지를 나타냄
CRC	Windows Update 사이트로부터 패치 파일을 다운로드받은 후, 정상적인 패치 파일인지 여부를 확인할 수 있는 crc값을 나타냄
name	다운로드받은 후 crc값이 일치하면 합당한 패치로 판단하고 name에 주어진 파일의 이름처럼 파일 이름을 간략하게 바꿈
switches	패치 파일을 설치하는 과정에서 필요한 옵션을 나타냄

다. SUS와의 연동 설계

MS의 다운로드 센터와의 동기화를 통하여 패치관리 서버가 XML형태의 패치정보를 생성할 수 있는 기초 데이터를 제공한

다.

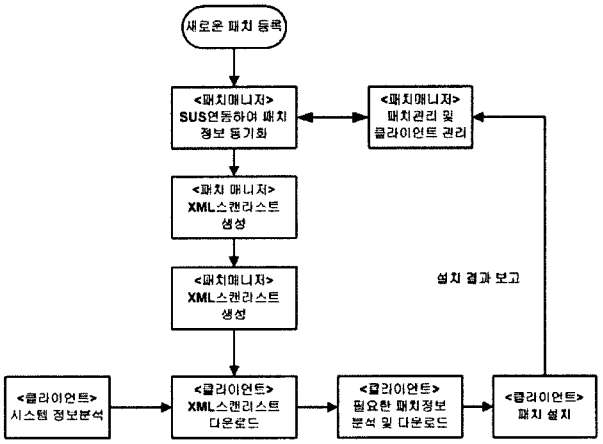
items.txt는 각각의 패치에 대한 GUID, name, publishername, <installed>, <excluded> 등의 정보를 포함하고 있다. itemsindex.txt는 각각의 패치에 대한 itemID 및 GUID 정보를 포함하고, itemstring.txt는 각각의 패치에 대한 GUID, 관련 설명, MS사의 관련 링크 주소 등의 정보를 포함하고 있다. itemstringindex.txt는 각각의 패치에 대한 GUID정보를 포함하고, product2items.txt는 각각의 운영체제에 대해서 필요한 보안패치 itemID 목록을 포함하고 있다.

그 밖에도 product2items.txt, productgroupstrings.txt, products.txt 등에 패치 정보를 분석하기 위해 운영체제 및 언어와 관련된 정보들을 포함하고 있다.

라. 패치 에이전트 설계

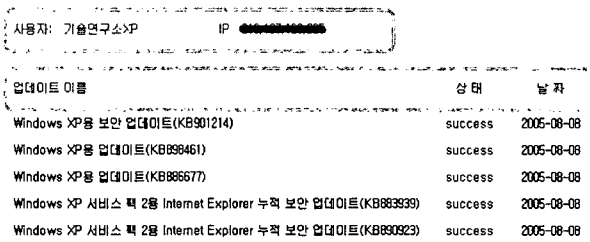
패치 에이전트는 스캔리스트를 이용하여 클라이언트 시스템에 설치된 보안패치와 설치해야할 보안패치를 검색하여 전자의 경우에 대한 내용을 담고 있는 정보파일과 후자의 경우에 대한 내용을 담고 있는 정보파일을 생성한다. 스캔리스트의 <installed> 필드 정보를 이용하여 보안패치가 클라이언트 시스템에 설치되어 있는지 여부를 확인한다. <excluded>필드 정보를 통하여 보안패치가 클라이언트 시스템에 설치될 수 없는 환경인지 여부를 확인한다.

MSXML의 XML 검색관련 API를 이용하여 전체 패치의 개수를 구한다. <detection>의 <installed>, <excluded> 밑에 있는 정보들을 검색하기 위해 설치 여부 점검 모듈 및 설치 배제 대항 여부 점검 모듈을 수행한다.[3] 설치배제 대상이 아님에도 불구하고 설치가 되어 있지 않은 경우 설치대상 정보파일에 해당 패치 파일 목록을 추가하고, 이와 반대로 설치되는 경우 설치된 정보파일에 해당 패치 파일 목록을 추가한다.



[그림 2] 새로운 패치의 분배과정

새로운 패치가 등록되었을 경우 [그림 2]의 과정을 통하여 패치가 최종의 클라이언트에 자동적으로 설치되고 해당 패치의 설치결과를 패치 매니저에 보고한다[그림 3].[4][5]


사용자: 기술연구소XP IP [redacted]

업데이트 이름	상 태	날 짜
Windows XP용 보안 업데이트(KB901214)	success	2005-08-08
Windows XP용 업데이트(KB898461)	success	2005-08-08
Windows XP용 업데이트(KB886677)	success	2005-08-08
Windows XP 서비스 팩 2용 Internet Explorer 누적 보안 업데이트(KB883939)	success	2005-08-08
Windows XP 서비스 팩 2용 Internet Explorer 누적 보안 업데이트(KB890923)	success	2005-08-08

[그림 3] 보안패치 설치결과 보고서

4. 결론 및 향후 연구 방향

본 논문에서는 윈도우의 보안패치를 신속하게 적용하기 위해 XML기반의 윈도우 보안패치 자동관리 시스템을 제안하였다. 제안하는 시스템에서는 기존의 관리자가 패치정보를 등록해야 하는 반자동적인 구조에서 관리자 및 벤더의 개입이 없이 운영체제를 제공하는 벤더의 패치다운로드 센터에서 패치관리시스템을 도입하는 최종 사이트의 클라이언트에 이르기까지 패치 설치의 전 과정을 자동으로 이루어지도록 설계하였다. 결과적으로 패치의 발표 시점에 맞춰 신속하게 적용함으로써 시스템의 취약점을 제거하고 네트워크의 안정화를 이룰 수 있다.

향후에 연구 및 개발되어야할 시스템에서는 본 논문에서 대상이 된 윈도우 시스템 외의 리눅스 및 유닉스 계열의 운영체제에도 제안한 시스템과 같이 패치의 전과정을 자동화할 수 있는 연구가 필요하다.

5. 참고문헌

[1] Microsoft, "Software Update Services Components and Features", September 17, 2003

[2] Danny Ayers 외, XML Application Development with MSXML 4.0, 2002

[3] MSXML Technology Preview SDK, http://www.xml.com/pub/r/650, 2000

[4] 서정택 외 "안전한 패치 자동분배 및 설치 기법", the 13th JCCI 2003

[5] 손태식 외 "안전한 패치 분배 구조 설계", 한국정보보호학회 추계발표대회, 2002. 10.