보안패치 관리도구의 설계 및 구현

김윤주*, 문종섭**
*국가보안기술연구소
**고려대학교 정보보호대학원
e-mail:kyj@etri.re.kr, jsmoon@korea.ac.kr

Design and Implementation of Tools for Security Patch Management

Yun-Ju Kim*, Jong-Sub Moon**
*National Security Research Institute
**Center for Information Security Technologies, Korea University

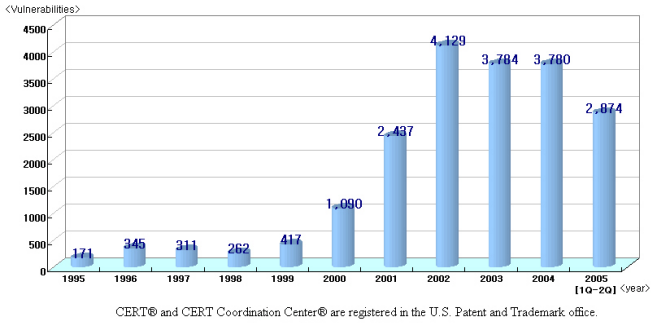
요 약

1.25 대란을 일으켰던 SQL Slammer 웜과 최근 IRCBot웜을 비롯한 다양한 악성코드들은 보안 취약점을 이용하여 전파되고 있다. 이러한 공격의 대부분은 사전에 보안패치를 적용하는 것만으로 막을 수 있기 때문에, 네트워크의 각 시스템들이 최신 패치 버전으로 업데이트 되었는지 점검하고 필요한 보안패치를 분배하는 자동화된 도구의 필요성은 강조되어 왔다. 본 논문에서는 보안패치 관리도구가 관리 대상 컴퓨터의 취약점을 분석하는 방안을 제시하고, 제시한 방안을 적용한 보안패치 관리도구를 설계 및 구현하였다.

1. 서론

카네기 멜론 대학 CyLab의 Sustainable Computing Consortium(SCC)은 상업용 소프트웨어가 일반적으로 1000라인의 소스 코드 당 20에서 30개의 결함을 가지고 있다고 보고하였다[1]. 모든 소프트웨어는 취약점을 포함하고 있고, 결함의 수는 소프트웨어의 복잡도가 증가함에 따라 증가한다. Microsoft사의 Windows는 2000 버전이 약13.5만 라인인 반면에 XP 버전은 약 40만 라인으로 구성되어 있어 새로운 버전이 출시되면서 점점 복잡해지는 것을 알 수 있다[2]. CERT Coordination Center (CERT/CC)에서 제공하는 1995년부터 2005년 상반기까지의 소프트웨어 결함에 의한 보안상의 취약점 보고 현황인 (그림 1)과 같이 실제 보안 취약점이 2000년 이후로 급격이 증가하고 있음을 볼 수 있다[3].

이러한 보안 취약점은 원격 코드를 실행, 권한 상승, 서비스 거부 등의 공격을 가능하게 하고, 최근에는 취약점이 발견된 후 극히 짧은 시간 내에 이를 이용한 사이버 공격이 발생하는 '제로데이 공격(zero-day attack)'이 현실로 다가오고 있어 최신 보안 패치를 신속하고 효율적으로 제공하기 위한 자동화된 보안패치 관리도구는 절실히 필요하다.

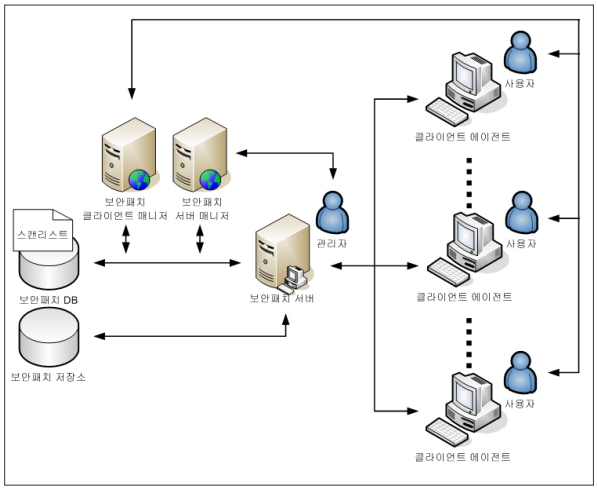


(그림 1) 보안취약점 보고 현황 (1995-2005상반기)

본 논문에서는 클라이언트 시스템의 취약점을 분석하고 XML을 기반으로 설치가 필요한 보안패치를 검색하는 방안을 제안하며, 제시한 방안을 적용한 보안패치 관리도구를 설계 및 구현한다. 2장에서는 보안패치 관리도구의 전체 구성을 살펴보고, 3장에서는 보안패치 분배 시나리오, 4장에서는 취약점 분석 방법을 서술한다. 5장에서 구현내용을 다루고, 마지막으로 6장에서 결론을 짓도록 한다.

2. 보안패치 관리도구의 전체 구성

본 논문에서 제안하는 도구는 다양한 운영체제를 대상으로 각 클라이언트에게 필요한 보안패치를 분배, 설치, 관리하는 도구이다. 전체 구성은 (그림 2)와 같고, 각각의 구성요소들의 기능을 설명한다.



(그림 2) 보안패치 관리도구 전체 구성도

- **보안패치 서버**: 클라이언트 에이전트와의 상호 통신작업을 지원하는 구성 요소이다. 에이전트로부터 프로파일 형식의 시스템 정보를 받고, 클라이언트에게 적합한 스캔리스트를 검색하고 전송하며, 설치 현황 정보를 관리하는 역할을 한다.
- **클라이언트 에이전트**: 사용자의 컴퓨터에 설치되어 시스템 정보 스캐닝, 패치 검색 및 설치, 스케줄 주기에 따른 동작 등의 일련의 작업을 수행한다.
- **보안패치 서버 매니저**: 관리자가 패치 관리 시스템을 관리할 수 있는 웹 기반 인터페이스를 제공하는 구성 요소이다. 새로운 패치가 나왔을 때 해당하는 패치 설치에 필요한 각종 정보를 추가, 수정, 자동동기화 하는 기능과 사용자 및 호스트의 정보 조회 기능을 제공한다.
- **보안패치 클라이언트 매니저**: 사용자가 자신의 시스템에 대한 정보를 살펴볼 수 있게 사용자에게 웹 기반 인터페이스를 제공하는 구성 요소로써 사용자 정보 조회, 시스템 정보 조회, 설치된 패치 조회/요청, 환경 설정 기능을 제공한다.
- **보안패치 DB**: 관리 대상 사용자 정보와 호스트 정보, 로그 정보 등을 저장한다.
- **스캔리스트**: 보안패치에 대한 정보를 포함하고 있는 XML 형태의 파일들이며, 각각의 운영체제 버전에 하나씩 존재한다. 스캔리스트는 클라이언트 에이전트에 설치해야할 보안패치가 존재하는지 여부를 판단하는 기준이 되며, 보안패치 설치에 필요한 정보를 제공한다.
- **보안패치 저장소**: 실제 패치 파일들이 저장되어 있는 구성요소이다.

3. 보안패치 분배 시나리오

본 도구는 클라이언트 에이전트가 주기적으로 패치 서버에 패치 정보를 요청하여 보안패치가 분배되는 경우, 사용자 요청에 의해 분배되는 경우와 관리자가 서버에게 분배요청을 하는 경우로 나뉘어 진다.

첫 번째 경우의 분배 과정은 다음과 같다.

> [C] : 클라이언트 에이전트가 하는 일
> [S] : 보안패치 서버가 하는 일
> A -> B: A에서 B로 전송

(1) [C] : 시스템 정보 스캐닝 및 프로파일 생성
프로파일은 운영체제 버전, 빌드번호, 서비스팩, 인터넷 익스플로러 버전 등 필요한 보안패치를 선택하기위한 정보를 포함한다.
(2) [C]->[S] : 프로파일 전송
(3) [S] : DB정보 업데이트 및 스캔리스트 선택
프로파일 정보를 DB에 업데이트하고, 그 정보를 이용하여 해당하는 스캔리스트를 선택한다.
(4) [S]->[C] : 스캔리스트 전송
(5) [C] : 설치해야할 보안패치 검색
스캔리스트의 정보를 이용하여 각 패치들의 설치여부를 확인하고 설치되지 않은 패치를 검색한다.
(설치되지 않은 패치가 있다고 가정)
(6) [C]->[S] : 설치되지 않은 패치 파일 요청
(7) [S]->[C] : 패치파일 다운로드
(8) [C] : 패치 설치
다운로드한 패치 파일을 스캔리스트의 설치 정보를 이용하여 설치한다.
(9) [C] : 로그 작성
(10) [C]->[S] : 로그 전송

사용자 요청과 관리자 요청에 의한 분배과정은 클라이언트 매니저와 서버 매니저를 통해 이루어지며 서버와 클라이언트간의 과정은 첫 번째 시나리오와 동일하다.

4. 클라이언트 시스템의 취약점 분석 방안

보안패치 관리도구는 대상 시스템의 취약점을 분석하여 적절한 보안패치를 분배해야 한다. 이를 위

해 기존에 제안한 자동화된 패치 관리 시스템은 서버가 대상 컴퓨터의 취약점을 찾아서 적절한 보안패치를 분배하였다[4][5]. 이 방안은 보안패치와 관련된 정보들을 서버의 데이터베이스에 저장하고, 클라이언트로부터 시스템 정보를 얻어와 해당 시스템의 취약점을 분석하기 때문에 다음과 같은 문제점이 발생하였다.

- 서버 과부하 - 서버에서 다수의 클라이언트를 대상으로 취약점을 분석해야 하므로 관리 도메인이 커질수록 서버에 과부하가 발생할 수 있다.
- 프라이버시 문제 - 패치 검색에 필요한 클라이언트 시스템에 대한 정보를 스캐닝하여 서버에게 전송해야 한다. 이 과정에서 중요 시스템 파일에 대한 정보나 특정 환경 설정값, 레지스트리 정보 등 많은 양의 정보가 전달되는 것을 사용자가 원하지 않을 수 있다.

이러한 문제점을 해결하기 위하여 본 논문에서는 서버가 클라이언트 시스템의 정보를 최소한으로(운영체제의 종류와 버전정보) 획득하며, 해당 시스템에 필요한 보안패치 정보만을 가지고 있는 스캔리스트를 구성하여 클라이언트 에이전트에게 전송해 주도록 설계하였다. 이렇게 전달받은 스캔리스트를 바탕으로 클라이언트 에이전트 스스로 자신의 취약점을 분석하며 만약 설치가 필요한 패치파일이 존재할 경우, 해당 패치파일을 서버에게 요청하도록 함으로써 앞에서 제시된 문제점들을 모두 해결하였다. 또한 데이터베이스를 이용하지 않기 때문에 별도의 설정 등이 필요하지 않아서 서버 구축이 매우 용이하다.

4.1. 스캔리스트의 구성

스캔리스트는 XML 기술을 기반으로 구성하여 DB의 장점인 추가·삭제의 편리함, 검색의 편리성 등을 모두 지원한다. 뿐만 아니라 XML은 정보를 파일로 저장하고 있기 때문에 클라이언트 에이전트가 스캔리스트를 요청했을 때 별도의 작업 없이 이미 파일로 저장된 XML형태의 스캔리스트를 전송할 수 있다는 장점이 있다.

스캔리스트는 <표 1>과 같은 태그로 구성되며, depth는 depth가 낮은 태그가 높은 태그에 포함됨을 의미한다. 즉 os 태그안에는 대상 시스템에 설치되어야 할 패치 개수만큼의 patch 태그를 포함하고, 각 patch 태그들은 detection 태그와 install 태그를 포함한다. os 태그의 운영체제 종류와 버전 정보를 가지고 대상 시스템의 관련 정보를 비교하여 적절한 스캔리스트를 받았는지 검사함으로써 정확성을 높인다. detection 태그의 installed 필드는 패치가 설치되었는지의 여부를 판단하기 위한 정보를, exception 태그는 해당 OS에 설치되어야 하는 패치이지만 특수한 경우 필요하지 않거나, 설치해서는 안 되는 패치를 구분하는 정보를 포함하고 있다. 이러한 정보를 바탕으로 해당 패치를 시스템에 설치해야 하는지의 여부를 판단할 수 있다. 그리고 install 태그는 description 태그와 installation 태그를 포함하는데, description 태그에 포함된 패치에 대한 설명은 사용자에게 어떠한 패치가 설치되는지 관련 정보를 제공하고, installation 태그는 실제 패치를 설치하는 과정에서 필요한 정보를 포함한다.

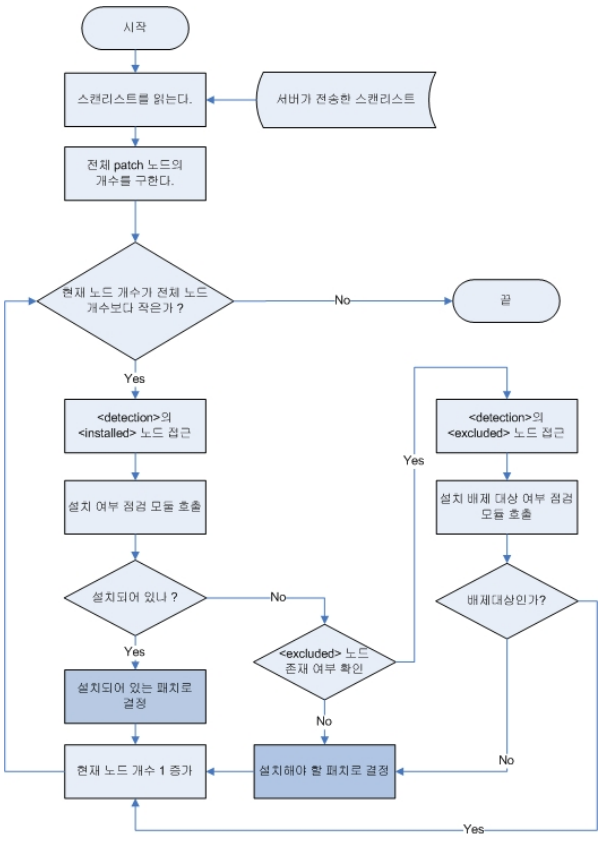
<표 1> 스캔리스트의 태그와 설명

태그	depth	설명
os	1	운영체제의 종류와 버전 - Windows의 경우, 버전, 제품 형태, 빌드 번호, 서비스팩 정보, 아키텍처 등을 포함 - Linux와 UNIX의 경우, 버전, 아키텍처, 커널 버전 등을 포함
patch	2	패치를 구분하는 단일의 정보
detection	3	설치된 패치와 설치해야 할 패치를 검색하는 기준
installed	4	설치 확인 정보 - 시스템 파일의 버전이나 레지스트리 값 등을 비교
exception	4	설치대상에서 제외되어야 하는지의 여부를 판단하기 위한 기준 - 의존성 문제를 해결하기 위한 태그
install	3	패치 설치 정보
description	4	사용자가 식별할 수 있는 패치에 관한 설명
installation	4	설치 과정에서 필요한 정보 - 파일이름, 파일 크기, 설치 형태, 설치 명령어, 재부팅 여부, 개별 설치 여부 등을 포함

스캔리스트의 정보는 충분한 테스트를 통해서 생성되어야 할 것이며, 스캔리스트의 정보를 악용한 공격이나 스캔리스트를 변조하는 공격의 가능성이 있으므로 암호화, 해쉬값 비교 및 인증 등의 방법을 이용하여 스캔리스트의 보안성을 확보할 수 있어야만 한다.

4.2. 취약점 분석 절차

클라이언트 에이전트가 보안패치 서버로부터 적합한 스캔리스트를 받은 후 (그림 3)과 같이 취약점 분석 과정을 통해서 설치되어 있는 패치와 설치해야 할 패치를 결정하여 서버에게 전송하면, 서버는 설치되어 있는 패치 정보로 패치 DB를 업데이트하고, 설치해야 할 패치의 설치파일을 클라이언트 에이전트에게 전송한다.



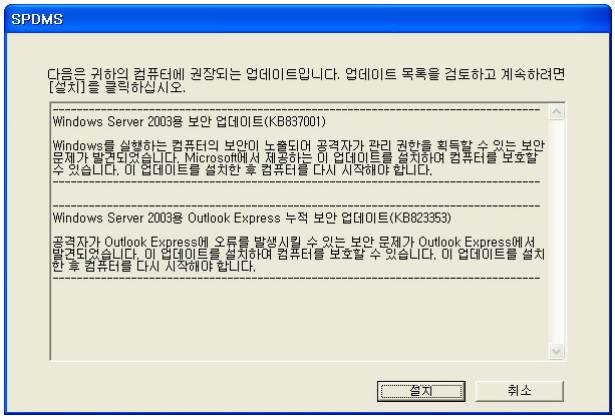
(그림 3) 취약점 분석 절차

5.구현

취약점 점검 모듈은 VC++ 6.0에서 구현하였고, MSXML.h에 포함된 IXMLDOMDocumentPtr, IXMLDOMNamedNodeMapPtr, IXMLDOMNodeListPtr, IXMLDOMNodePtr 클래스를 이용하여 XML 데이터를 활용하였다. 그리고 Windows 2000/XP/2003 시스템을 대상으로 테스트하였다. (그림 4)는 설치해야 할 패치를 검색하고 다운로드한 후, 사용자에게 설치여부를 확인하는 화면이다.

6.결론

본 논문에서 제안한 보안패치 관리도구는 보안패치 분배 및 설치를 자동화하여 대상 네트워크의 보안 취약점을 제거함으로써 악의적인 공격을 사전에 예방



(그림 4) 설치 승인 요청

할 수 있다. 특히 스캔리스트를 이용한 취약점 분석 방안은 기존에 제안된 보안패치 관리 시스템의 서버 과부하, 프라이버시 침해 문제점들을 보완하였다.

앞으로 멀티 플랫폼 환경에서 패치 관리가 효율적으로 이루어지기 위해서는 스캔리스트의 구성정보와 벤더의 패치 정보 제공 방식에 대한 표준화 연구가 필요하며, 패치 미적용시 국내의 경우 5분 국외의 경우 20분 내에 악의적인 공격으로부터 피해를 입는다는 통계를 비추어 볼 때, 공격보다 패치분배가 먼저 이루어져야하는 것에 대한 연구가 진행되어야 할 것이다.

참고문헌

[1] Michelle Delio, "Linux: Fewer Bugs Than Rivals" http://www.wired.com/news/linux/0,1411,66022,00.html, 2004

[2] Jeordan Legon, "Profanity, partner's name hidden in leaked Microsoft code", http://www.cnn.com/2004/TECH/biztech/02/13/microsoft.source/, 2004

[3] Computer Emergency Response Team(CERT)/Coordination Center(CC) : CERT Statistics, "Vulnerabilities reported 1995-2005", http://www.cert.org/stats/cert_stats.html

[4] Sohn T.S., Moon J.S., Seo J.T., Im E.K., Lee C.W., "Safe Patch Distribution Architecture in Intranet Environments", SAM, 2003

[5] Seo J.T., Yoon J.B., Choi D.S., Park E.K., Sohn T.S., Moon J.S., "A Study on the Patch Management System In Intranet", PACO, 2004