XML을 이용한 보안패치 중앙 관리 시스템 설계

김윤주° ,이상원*, 손태식*, 문종섭*, 서정택**, 이은영**, 박응기**
고려대학교 정보보호대학원*, 국가보안기술연구소**
{zzuya99°, a770720, 743zh2k, jsmoon}*@korea.ac.kr, {seojt, eylee, ekpark}**@etri.re.kr

Design the Security Patch Central Management System Using XML

YunJu Kim° ,SangWon Lee*, Tae-Shik Sohn*, Jong-Sub Moon*, JungTaek Seo**, Eun-Yong Lee**, EungKi Park**
Center for Information Security Technologies (CIST), Korea University*,
National Security Research Institute**

요 약

최근 급증하고 있는 시스템에 존재하는 취약성을 이용한 공격들의 경우, 관련된 보안패치를 설치함으로써 대처할 수 있다. 그러나, 기업이나 공공기관과 같이 대규모 네트워크를 형성하고 있는 곳에서 패치 설치 작업을 각각의 개인에게 믿고 맡기는 것은 매우 수동적인 자세로서, 지금까지의 보안침해 사고 현황에서도 알 수 있듯이 매우 위험한 발상이다. 따라서, 조직의 중앙에서 능동적으로 각각의 시스템의 보안패치 설치 현황을 파악하고, 필요한 경우 적절한 조치를 취하는 것이 보다 바람직한 대처방안이라고 할 수 있다. 본 논문에서는 XML을 이용하여 중앙에서 클라이언트의 보안패치 설치 현황을 일괄적으로 관리할 수 있는 '보안패치 중앙 관리 시스템'을 제안한다.

1. 서 론

최근 다양한 웜들로 인한 피해 사례가 속출하면서 보안에 대한 관심이 높아지고 있다. 특히 MS SQL Slammer worm에 의해 발생한 1.25 대란은 전 세계적인 네트워크 마비로 이어져 시스템 관리자가 보안의 중요성을 느낄 수 있는 계기가 되었다. 그러나 표1과 같이 웜의 초기 발견일보다 약 7개월 전에 이미 취약성과 그에 대한 패치가 발표되었기 때문에, 보안패치를 미리 적용하였더라면 그와 같은 대란은 겪지 않을 수 있었다. 따라서, 보안패치의 적용은 매우 중요한 작업이라고 할 수 있다.

Worm	웜 초기 발견일	취약성 발견일
Nimda	2001년9월18일	2000년10월17일
CodeRed	2001년7월16일	2001년6월18일
SQL Slammer	2003년1월25일	2002년7월24일
Blaster	2003년8월11일	2003년7월16일

표 1 웜에 대한 초기 발견일 및 관련 취약성 발견일[1]

하지만 새로운 취약점과 그에 대한 패치가 빠른 속도로 등장하고 있기 때문에 패치를 단순히 수동적으로 관리하는 것은 각종 패치 수집, 테스트, 배포, 적용 및 확인의 어려움이 있다. 따라서 이러한 일련의 패치 관리 작업을 자동화하기 위한 솔루션을 도입할 필요가 있다. 이때 자동화된 보안패치 관리 시스템에서 가장 중요한 것은 자신이 관리하는 네트워크 내부에 취약성을 가진 시스템이 존재하는지 여부를 파악하고, 만약 존재할 경우 해당 시스템에 대해서 중앙에서 필요한 조치를 취할 수 있도록 하는 것이다.

우선 취약성을 가진 시스템의 존재 여부를 파악하기 위해서는 각각의 시스템에 대하여 취약점을 검색할 수 있는 기준(*스캔 리스트*)이 필요하며, 다음으로 중앙에서 필요한 조치를 취하기 위해서는 해당하는 판단의 근거가 될 수 있는 정보가 필요하다.

본 논문에서는 지금까지 일반적으로 DB를 통한 관리 시스템에서 벗어나, 보다 많은 장점을 확보할 수 있는 XML을 이용하여 스캔 리스트를 구성하고, 각각의 사용자별 보안패치 설치 현황(*패치 리스트*) 및 각각의 보안패치별 사용자 설치 현황(*클라이언트 리스트*) 정보를 확인할 수 있는 목록을 생성하여 보안패치 중앙 관리 시스템에서 이들을 활용하는 방안에 대해서 제안하고자 한다.

2. 보안패치 관리 시스템의 구성 및 시나리오

2.1 보안패치 관리 시스템 구성

패치 관리 시스템은 상이한 시스템들로 구성되어 있는 대규모 네트워크 환경에 적합한 보안패치를 자동 분배, 설치하고 관리하는 시스템이다[2, 3]. 그림1은 패치 관리 프레임워크의 전체 구성도이며, 보안패치 서버, 보안패치 매니저, 보안패치 클라이언트, 보안패치 에이전트, 보안패치 저장소 및 보안패치 DB를 대체하는 XML형태의 데이터로 구성된다.

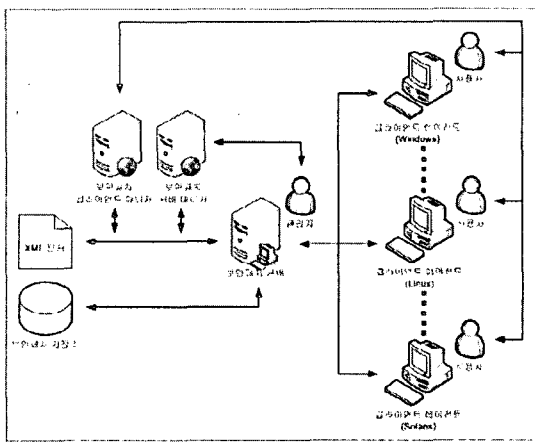

그림 1 보안패치 관리 시스템 전체 구성

- **XML형태의 데이터** : 스캔 리스트, 패치 리스트, 클라이언트 리스트를 포함
- **보안패치 서버** : 각각의 클라이언트에게 필요한 스캔 리스트를 제공하고, 중앙 관리를 위한 정보를 수집
- **보안패치 서버 매니저** : XML 구성정보와 서버를 관리하며, 중앙 관리 기능을 직접 수행
- **보안패치 클라이언트 에이전트** : 스캔 리스트를 바탕으로 클라이언트 시스템을 검색하고 패치 자동 설치를 수행
- **보안패치 클라이언트 매니저** : 클라이언트와 관련된 정보를 관리

- **보안패치 저장소** : 실제 보안패치 파일을 저장하는 공간

2.2 보안패치 분배 및 설치 시나리오

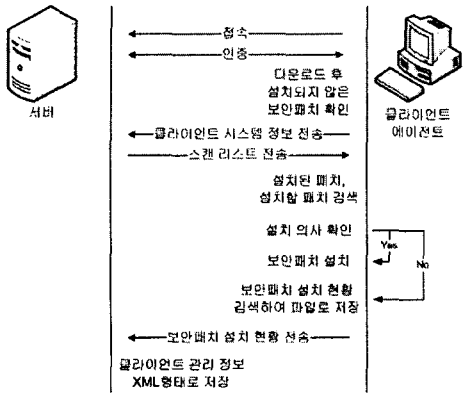

그림 2 보안패치 관리 시나리오

그림 2와 같이 클라이언트 에이전트는 서버에 접속하여 상호 인증과정을 거친 후, 사전에 설치되지 않은 보안패치가 존재할 경우 관련 설치작업을 진행한다. 그리고 서버에게 클라이언트 시스템 정보를 전송하면, 서버는 시스템에 대응하는 스캔 리스트를 선택하여 전송한다. 스캔 리스트는 시스템에 설치되어 있어야 할 보안패치들의 설치 관련 정보들로 구성되어 있다. 우선, 스캔 리스트의 설치 확인 정보를 이용하여 설치되어 있는 보안패치와 설치해야할 보안패치를 검색한다. 설치해야할 패치가 존재하는 경우, 사용자에게 설치 의사를 확인하여 보안패치를 설치하고 최종 설치 현황인 패치 리스트를 서버에 전송한다. 이것은 서버가 항상 클라이언트에 대한 최신 정보를 유지하기 위해서이다. 서버는 클라이언트의 보안패치 설치 현황을 이용하여 각각의 패치에 대한 클라이언트 현황인 클라이언트 리스트를 별도로 저장한다. 이와 같은 과정으로 보안패치 분배 및 설치 작업이 이루어진다.

3. XML을 이용한 보안패치 관리의 장점

XML은 eXtensible Markup Language의 약자로 W3C에서 1998년에 제정한 표준으로, 보안패치 관리 시스템에서 이용함으로서 다음과 같은 이점을 얻을 수 있다[4][5][6].

표 2 XML을 이용한 보안패치 관리의 장점

서버 구축의 용이성	데이터베이스를 이용하지 않기 때문에 별도의 설정등이 필요하지 않아 서버 구축이 용이하다.
서버의 과부하 방 지	데이터베이스를 이용하는 경우 서버가 정보를 가지고 있기 때문에 설치되어야 할 보안패치 검색과정을 서버가 처리해야한다. 대규모 네트워크를 관리하는 경우 수많은 클라이언트들에 대한 처리를 해야 하므로 서버의 과부하를 가져 올 수 있고 클라이언트의 정보를 얻어 오는 것에 한계가 있으며 클라이언트의 정보를 서버가 추출해 가는 것은 프라이버시 관점에서도 문제가 된다. 그러나 XML을 이용하는 경우 정보를 클라이언트에게 주어서 클라이언트 시스템이 직접 설치된 패치와 설치해야할 패치 목록을 작성하여 보다 정확한 정보를 얻을 수 있고, 위의 문제들도 해결할 수 있다.
스캔 리스트 업데이트 및 확장의 용이	새로운 패치가 벤더로부터 배포되어 스캔 리스트를 업데이트하거나 관리를 위한 목적으로 스캔 리스트를 확장시 매우 용이하다.
클라이언트 관리 정보 업데이트	각 패치에 대한 클라이언트들의 설치 여부를 저장할 때 XML을 이용하면 클라이언트가 패치를 설치하였을 때 업데이트가 용이하고, 서버 매니저에서 정보를 출력할 때 검색이 용이하다.
설치 정보 검색	설치가 필요할 경우 스캔리스트에서 설치 정보의 검색이 용이하고 설치 정보를 클라이언트가 가지고 있으므로 서버와의 통신횟수를 줄일 수 있다.

4. XML을 이용한 보안패치 관리 방안

4.1 스캔 리스트

스캔 리스트는 각각의 운영체제 벤더 및 버전에 따라서 별도로 구성되며, 시스템에 설치되어 있어야 할 패치들의 설치 사실을 확인할 수 있는 정보와 설치 작업과 관련하여 필요한 명령어 등의 정보를 포함하고 있다.

```
- <os type="Windows">
  - <patch id="24595123" name="security patch">
    - <scan>
      + <installed>
      + <exclusion>
      + <necessary>
      </scan>
    - <install>
      + <description>
      + <installation>
      </install>
    </patch>
  + <patch id="24595150" name="security patch2">
  + <patch id="24595151" name="security patch3">
  </os>
```

그림 3 스캔 리스트의 구성

표 3 스캔 리스트의 태그와 설명

Tag		Description
os		운영체제의 종류와 버전
patch		각각의 패치를 구분하는 태그
scan		시스템에 설치된 패치와 설치할 필요가 있는 패치의 존재 여부를 확인할 수 있는 정보를 가지고 있는 태그
installed		설치 여부 확인
선택	exclusion	installed에서 설치되지 않은 것으로 판단한 경우, 설치 제외대상에 속하는지 확인 제외대상이 아니면 설치 필요
	necessary	installed에서 설치되지 않은 것으로 판단한 경우, 설치에 적합한 환경인지 확인 적합하면 설치 필요
install		설치하는 과정에서 필요한 정보를 가지고 있는 태그
description		사용자에게 알리는 패치에 관한 설명
installation		설치에 필요한 정보 (예를 들면, 파일이름, 파일크기, 설치형태, 설치명령어, 재부팅여부, 개별설치여부 등)

스캔 리스트를 구성하는 정보인 설치 확인 정보, 설치 제외 대상 또는 설치 환경, 설치 정보는 보안패치의 특성상 각각의 운영체제 벤더에 대해서 의존적일 수밖에 없다. 현재는 직접 벤더로부터 기술 정보를 제공받거나, 벤더에서 공개적으로 제공하는 패치에 대한 정보를 통해서 임의로 구성할 수 있다. 전자의 경우 각 벤더별 특징들 때문에 일관된 스캔 리스트를 생성할 수 없다는 문제점이 존재하며, 후자의 경우 보통 수동적으로 관련 정보들을 취합하여 재구성해야 하기 때문에 이 과정에서 스캔 리스트의 정확성이 떨어질 수 있으며 이것은 시스템에 맞지 않는 잘못된 패치를 분배하여 설치하는 등의 매우 심각한 문제를 발생시킬 수 있다.

따라서, 어떠한 플랫폼에도 쉽게 적용된다는 XML의 장점을 최대한 살려서 각 벤더에서 공통된 형식으로 보안패치 정보를 제공할 수 있도록 그 표준을 정하는 것이 가장 바람직한 방법이다.

또한 스캔 리스트의 정보는 악의적인 사용자에게 공개될 경우, 실제로는 설치하지 않은 시스템에 대해서 마치 설치한 것처럼 속이는 등의 공격을 할 수 있다. XML의 보안체계가 아직까지는 미비하므로 암호화, 해쉬값 비교 및 인증 등의 방법을 이용하여 스캔 리스트의 안정성을 확보할 수 있어야만 한다.

4.2 보안패치 설치 현황 - 패치 리스트

클라이언트 시스템에 설치되어 있는 패치 목록과 설치해야할 목록을 정리한 정보이다.

```
- <user id="kdhong">
  - <installed value="1">
    + <identity id="24595123" name="security patch">
    + <description>
    </installed>
  + <installed value="1">
  - <installed value="0">
    + <identity id="24595150" name="security patch3">
    + <description>
    </installed>
  + <installed value="1">
  </user>
```

그림 4 보안패치 설치 현황의 구성

표 4 보안패치 설치 현황의 태그와 설명

Tag	Description
user	클라이언트 ID에 따른 구분 태그
installed	value의 값이 1이면 설치된 패치 0이면 설치해야할 패치
identity	패치의 구분
description	사용자가 식별할 수 있는 패치에 관한 설명

각 클라이언트의 보안패치 설치 현황을 저장하여 중앙에서 관리의 목적으로 사용할 수 있을 뿐만 아니라 사용자 자신이 직접 자신의 시스템의 취약점을 확인할 수 있도록 할 수 있다. 이 정보는 클라이언트 에이전트에서 구성하여 서버에게 전송해서 항상 최신 정보를 유지할 수 있도록 운영한다. 또한 관리자가 클라이언트 시스템에 보안 취약성이 존재한다고 판단하였을 때 보안패치 설치를 권고하는 등의 관리를 할 수 있는 기반정보를 제공한다.

4.3 클라이언트 관리 정보 - 클라이언트 리스트

클라이언트 관리 정보는 각 패치에 대해서 설치한 사용자와 설치해야할 사용자를 저장한다.

```
- <patch id="24595123" name="security patch">
  - <installed value="1">
    + <user id="kdhong">
    </installed>
  - <installed value="0">
    + <user id="zzuya">
    </installed>
  - <installed value="1">
    + <user id="comanz">
    </installed>
  + <installed value="1">
  </patch>
```

그림 5 클라이언트 관리 정보의 구성

표 5 클라이언트 관리 정보의 태그와 설명

Tag	Description
patch	패치의 구분
installed	value의 값이 1이면 설치한 사용자 0이면 설치해야3할 사용자를 의미
user	클라이언트의 구분

이 정보는 각각의 패치에 대해서 얼마나 많은 클라이언트들이 설치하고 있는지 확인할 수 있는 정보들로 구성된다. 중요한 패치인 경우 이 정보를 이용하여 설치하지 않은 시스템들을 모두 확인할 수 있고, 필요한 경우 일괄적으로 설치를 권고하는 기능을 제공한다.

5. 결 론

최근 자동화된 패치 관리 시스템에 대한 연구가 많이 이루어지고 있으며, 그에 따른 상용화 제품도 속속들이 등장하고 있다. 이것은 패치 관리 시스템의 필요성에 대해서 이미 많은 사람들이 인식하고 있다는 의미일 것이다. 그러나 보안패치 관리 시스템이 운영체제 벤더에 대해서 지나치게 의존적으로 구성되어 있으며, 보다 효율적으로 중앙에서 관리할 수 있도록 하는 방안에 대한 연구가 활발히 진행되고 있지 않다.

그러므로 본 논문에서 제안하는 XML을 이용한 보안패치 중앙관리 시스템은 벤더의 영향을 최소로 하는 코드를 작성하여 시스템의 유지 보수를 용이하게 하고, 사용자뿐만 아니라 관리자가 각 사용자의 시스템에 적용된 보안패치 현황을 손쉽게 파악할 수 있으며, 각 패치에 대해 어떠한 사용자가 설치하였고 설치하지 않았는지의 정보를 정확하게 파악할 수 있다. 이것은 전체 네트워크의 취약성 제거를 각각의 개인이 담당하는 것이 아니라 중앙에서 직접 보안패치 관리를 할 수 있도록 제공함으로써 보다 안전한 네트워크를 형성할 수 있도록 할 것이다. 마지막으로 각 운영체제 벤더들이 스캔 리스트의 표준화 작업을 시작할 것을 다시 한번 제안한다.

6. 참고 문헌

[1] KISIA, http://www.kisia.or.kr/

[2] Sohn Tae-Shik, "Safe Patch Distribution Architecture in Intranet Environments", SAM, 2003

[3] Cheol-Won Lee, "A Secure Patch Distribution Architecture", ISDA 2003, Lecture Notes in Computer Science, Springer-Verlag, 2003

[4] Vidgen, R., Goodwin, S., "XML: what is it good for?", Computing & Control Engineering Journal, Volume: 11 Issue: 3, June 2000, Page(s): 119 -124

[5] 이상원, "멀티플랫폼 환경에서의 보안패치 분배를 위한 DB구축 및 검색 방법에 관한 연구", 한국정보과학회, 4. 2004

[6] 민동옥, "보안패치 자동분배를 위한 패치 DB 자동구성 방안", 한국정보과학회, 4. 2004