아파치 웹서버의 URL 공격에 대한 보안취약성 분석

최종천* 최진우** 조성제***
* 단국대학교 컴퓨터과학 및 통계학과 컴퓨터과학전공
** 단국대학교 컴퓨터공학과
*** 단국대학교 정보·컴퓨터학부 컴퓨터과학전공
e-mail : godofslp@dankook.ac.kr

Analysis of a Security Vulnerability of URL Attack on Apache Web Server

Jong-Cheon Choi*, Jin-Woo Choi**, Seong-Je Cho***
*Dept of Computer Science and Statistics, Dankook University
**Dept of Computer Engineering, Dankook University
***Division of Information and Computer Science, Dankook University

요 약

본 논문에서는 기계어 프로그램만 주어진 환경에서 디버깅 및 역공학 도구를 사용하여 소프트웨어 보안취약성을 분석하는 방법에 대한 연구를 수행하였다. 즉, MS사의 윈도우즈 2000 서버 운영체제 상에서 아파치(Apache) 웹서버를 대상으로, URL 공격에 대한 한 취약성을 재연하고, 취약점이 있는 코드 부분을 추출하였다. 이는 기계어 프로그램을 실행하면서 보안 취약성 분석 절차를 이해하고 보안 결함을 발견해 내는 기반 자료로 활용될 수 있을 것이다.

1. 서론

취약성(vulnerability)이란 '보안 정책을 위반하는 시스템 고장'이라고 볼 수 있으며, 소프트웨어는 명세, 개발, 또는 구성 등에서 존재하는 오류로 인해 취약성이 존재할 수 있다. 따라서 공격이란 취약성의 악용이며, 보안취약성은 보안사고를 유발하는 공격의 근원이라 할 수 있으며, 해커가 보안취약성을 악용하면 그것이 보안취약점으로 구체화되어 보안사고의 원인이 된다.

소프트웨어에 존재하는 보안취약점은 개발자의 실수에 의해 발생되기도 한다. 예를 들어, 버퍼 오버플로우, 서비스 거부, 명령 수행 실패, 파일 접근 실패 등이 보안 취약성이며, 대부분의 소프트웨어에서 발생할 수 있는 하나의 특징이며, 이러한 취약성이 특정 소프트웨어에서 발견되어 보안 문제의 소지가 되면, 이를 보안취약점이라 한다.

정보보호의 한 분야는, 소프트웨어의 보안취약성을 분석하여 보안취약점을 탐지해 내며, 탐지된 보안취약점을 보완하여 이를 악용하려는 공격을 방지하는 것이라고 볼 수 있다. 공격자가 이용할 수 있는 상용제품의 보안취약성의 양은 매우 많다고 알려져 있다. 따라서, 소프트웨어의 보안취약성을 분석하여 개개의 소프트웨어에 존재하는 보안취약점을 재연하고 발견·분석하는 체계적인 방법이 요구되어진다.

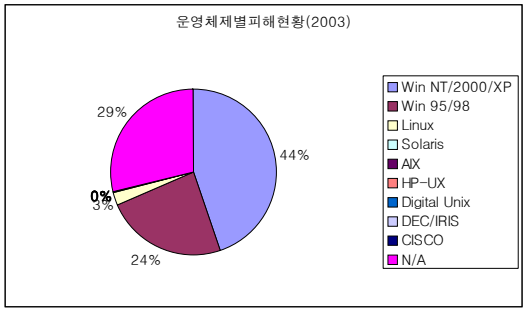
취약성 분석은 소프트웨어에 존재하는 결함이 보안사고와 연관될 수 있는지를 앞서 판단하는 심도 있는 예측을 하는 보안사고 예방분야이기도 하며, 개발이 완료된 기계어 프로그램 분석의 경우 소프트웨어 역공학과도 관련 있는 중요한 기술이다.

본 논문에서는, MS 윈도우즈 운영체제 상의 아파치(Apache) 웹서버를 대상으로, URL 공격 취약점 자료를 수집하여 해당 취약성을 재연한 다음, 정적 분석, 동적 분석, 역공학 등의 기술을 이용하여 재연된 보안 취약성을 중심으로 취약점 발생 과정을 추적하고 관련 코드 부분을 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 보안취약성 현황 및 분석 방법을 설명하고, 3장에서는 아파치 웹서버 구조 및 분석할 취약점 정보에 대해 기술한다. 4장에서는 취약성을 재연하고 분석하였으며, 5장에서 결론 및 향후 연구에 대해 기술한다.

2. 보안취약성 현황 및 분석 방법

많이 사용되고 있는 MS 윈도우즈의 경우 소스가 제공되지 않는 상용프로그램이 주류를 이루고 있으며, 이에 따라 알려진 취약성에 대한 개발사의 대응에 사용자는 전적으로 의존하게 된다. 한편, (그림 2-1)에 나타나 있는 것처럼, MS 윈도우즈 계열 운영체제가 최근 해커들의 주요 공격대상이 되고 있음을 알 수 있다. 이는 MS사가 제공하는 프로그램뿐만이 아닌 윈도우즈 상에서 동작되는 다양한 네트워크 소프트웨어에 대한 보안취약성의 분석과 그에 대한 면역강화가 필요함을 보여주고 있다[1].



(그림 2-1) 2003 운영체제 피해현황

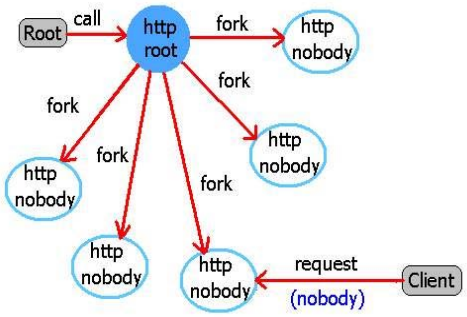
소프트웨어의 취약점에 기인한 막대한 손실과 보안 관련 오류의 심각성이 커짐에 따라 소프트웨어 결함을 줄이려는 다양한 노력이 시도되고 있으며, 최근 소프트웨어의 규모증가와 그에 따른 분석의 복잡도 증가로 취약성 탐지를 자동화하는 도구를 개발하는 연구가 진행되고 있다. 이러한 연구노력으로는 프로그래머에 대한 체계적인 교육, 자동화된 프로그래밍 도구 개발, 품질보증 테스트, 그리고 소프트웨어의 취약성을 자동으로 분석하는 도구 개발 등이 있다.

취약성 자동 분석 도구는 접근 방법에 따라 정적 분석, 동적 분석, 결함주입 등으로 구분할 수 있으며, 이들 개별 방법에 의한 보안취약성의 분석은 소프트웨어의 크기가 커짐에 따라 한계상황에 달하고 있다. 따라서, 대규모 기계어 프로그램에 대한 보안취약성을 효과적으로 분석하는 방법과 도구를 개발할 필요가 있다.

3. 아파치 웹서버

3.1 아파치 웹서버 구조

아파치(Apache) 웹서버의 구조가 (그림 3-1)에 나타나 있다. 아파치 데몬이 있어 클라이언트로부터 요청이 들어오면 새로운 프로세스를 생성하고 해당 클라이언트의 요청을 처리하게된다. 윈도우즈와 유닉스 상에서 광범위하게 사용되는 보편적인 네트워크 서버인 점과 실행코드에 대한 접근이 용이한 공개소프트웨어인 점을 고려하여 본 논문에서는 아파치를 분석 대상으로 삼았다[2].



(그림 3-1) 아파치 프로세스 구조

3.2 분석 대상 취약점

분석 대상에 대한 취약점 정보는 CVE (Common Vulnerabilities and Exposure)를 중심으로 수집하였다. CVE란 보안 취약성에 대해서 표준화된 이름을 제공하기 위한 취약성 명명법을 말한다[3].

분석할 내용은 Apache HTTP Server 1.3.x에 대한 URL 공격에 대한 것으로, 관련정보가 [표3-1]에 나타나 있으며, 공격에 필요한 코드(exploit code)는 SecuriTeam.com에서 획득하였다[4, 5].

CVE-2000-0505
The Apache 1.3.x HTTP server for Windows platforms allows remote attackers to list directory contents by requesting a URL containing a large number of / characters.

[표 3-1] CVE-2000-0505 내용

이는 URL 요청에서 다량의 '/' 문자에 의하여 발생하는 취약점으로, 취약점 공격이 성공되면 루트 디렉토리의 리스트가 표시된다. 이 취약점에 노출된 버전은 아파치 1.3.12 및 이전 버전이다.

4. 취약성 재연 및 취약점 분석

4.1 취약점 공격

공격은 [표 4-1]과 같이 지정한 호스트에 대하여 취약점에 의한 응답이 있을 때까지 다수의 '/'를 전송하는 것으로 종료되며, 이에 대한 결과는 현재 아파치

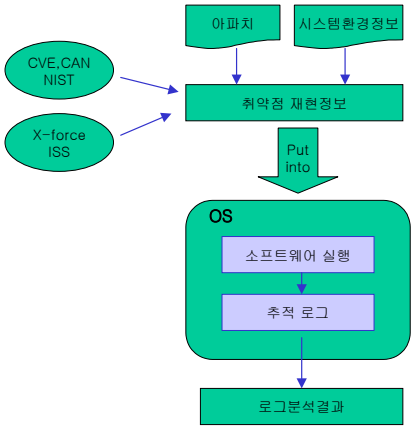
웹서버의 루트디렉토리에 대한 정보가 HTML 문서화 되어 보여진다. 이것은 해당 디렉토리가 가지는 index.html의 존재와 무관하게 발생하는 오류로, 접근 권한이 없는 주요 정보가 공격자에게 제공된다.

```
$odata = get("http://$host/");
if ($odata eq "")
{
    die "no response from server:  $host\n";
}
for ($i = 2; $i < 4096; $i++)
{
    $data = get("http://$host" . ("/" x $i));
    if ($data ne $odata)
    { exit; }
}
```

[표4-1] 아파치 공격코드 요약

4.2 취약점 분석

본 논문에서는 취약성을 재연하기 위해 (그림 4-1)와 같은 과정을 수행하였다. 즉, 취약점 정보 및 공격 코드를 수집·분석하여 4.1절에서 보안취약성을 재연하고 추적하여, 취약점 관련 코드를 분석하였다.

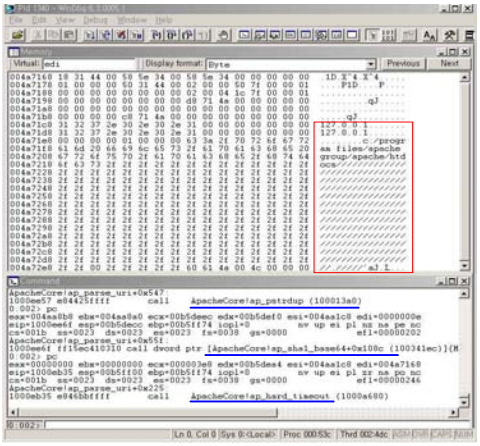


(그림 4-1) 소프트웨어 보안취약점의 재연과정

공격절차는 먼저 4.1절의 공격코드를 사용하여 아파치 서버를 공격하고, MS에서 제공하는 WinDBG 디버거를 사용(attach)하여 (그림 4-2)와 같이 그림에서 EDI값을 참조하면 공격에 사용한 다수의 '/'를 확인할 수 있으며, 오류와 관계된 함수정보를 확인 할 수 있었다[6].

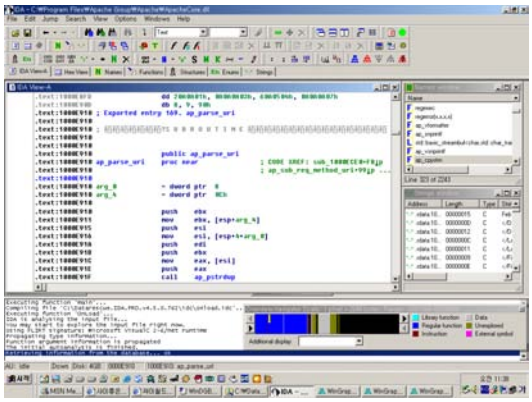
아파치는 apache.exe를 실행하면 호출에 의해 apachecore.dll에 내장된 함수가 작동하는 구조로 되어있어 실제 핵심적인 함수와 내용은 apachecore.dll에 있으며, 이러한 구조는 MS의 비주얼스튜디오에서 제공하는 Depends 유틸리티를 이용하여 실행 및 함수에 대한 관계를 파악할 수 있다. 분석결과 apachecore.dll내에서 오류가 발생함을 알 수 있었다.

분석된 내용을 기반으로 역공학 도구인 IDApro를 통하여 아파치 상에서 사용되는 실제 함수와 그에 대한 연관 구조를 파악할 수 있었으며, 이러한 과정을 거쳐 취약점이 있는 지점을 유추하였으며, WinDBG를 사용하여 아파치를 부착(attach)한 후 유추된 함수들 중 가능성이 높은 함수집단에 대해 브레이크 포인트를 통하여 추적을 진행함으로써 (그림 4-2)과 같이 ap_parse_uri라는 URL분석 함수에서 오류가 유발됨을 확인 할 수 있었다.



(그림 4-2) WinDBG를 통한 정지점 및 EDI 내용

이러한 분석된 내용을 기반으로 IDApro를 다시 사용하여 역공학을 통한 내용분석을 진행하였다[7].



(그림 4-3) IDApro를 통한 역공학 결과

이때 (그림 4-3)와 같이 ap_parse_uri 함수에 대한 분석된 내용에서 실제 취약점 코드를 확인할 수 있으며, 실제 코드 상에서의 취약점 위치를 추적할 수 있었다.

4.3 대응방법

본 논문에서 다루어진 취약성은 아파치 웹서버의 버전 향상만을 통해서 쉽게 패치가 가능하다. 하지만,

문제의 원인이 URL 분석에 따른 오류에서 발생하는 문제였던 만큼 프로그램 작성 시 오류가능성에 대한 철저한 검증이 따르지 않는다면 또다시 발생할 가능성을 가지고 있다. 때문에 소프트웨어 개발자는 다양한 상황에 대하여 예외처리 기법을 적용해야 하며, 추가로 개발 프로그램의 신뢰성을 검증할 수 있는 자동화된 도구의 개발이 필요하다.

5. 결론 및 향후연구

연구 결과에 의하면 개발자들이 프로그램 작성 시 동일한 실수를 반복하며, 그 중 상당 부분은 개발 시의 부주의로 인해 오류가 발생한다. 따라서, 개발된 소프트웨어에 대해 알려진 취약성을 분석하여 소프트웨어 결함을 줄이고 소프트웨어의 신뢰도를 향상시키는 연구가 필요하며, 이는 보안사고를 미연에 예방하는 효과를 가져다 주기도 한다.

본 논문에서는 MS사의 윈도우즈 서버 운영체제 상에서 기계어 코드만 주어진 아파치 웹서버 프로그램을 대상으로, 디버깅 및 역공학 도구를 사용하여 URL 관련취약성을 재연하고 관련 코드를 분석하는 연구를 수행하였다.

향후, 제시된 소프트웨어 보안취약성 분석 과정을 체계화하는 심층 연구를 계속 수행하여, 소스가 없고 기계어 프로그램만 주어지는 시스템 환경에서 취약점을 분석하여 보안 패치를 개발할 계획이다. 이처럼, 기계어 프로그램에 대한 보안 패치에 대한 연구가 이루어진다면, 해당 프로그램 개발 회사에 의존하지 않고 좀 더 보안이 강화된 환경을 자체적으로 구축할 수 있게 될 것이다.

참 고 문 헌

[1] 2003년 12월 해킹바이러스 통계 및 분석 월보, KISA, Dec, 2003.
[2] http://www.apache.kr.net/#intro
[3] http://www.cve.mitre.org/cve
[4] http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0505
[5] http://www.securiteam.com/bid/1284
[6] John Robbins , *"Debugging Applications"*, Microsoft Press , 2000.
[7] IDA Pro 웹 사이트, http://www.datarescue.com/idabase