IIS 웹서버에서 WebDAV의 버퍼 오버플로우 취약점 분석

장혜영, 민욱기, 조성제
단국대 정보컴퓨터학부 컴퓨터과학및통계학과
chenill@dankook.ac.kr, wearemin@nownuri.net, sjcho@dankook.ac.kr

Analysis of a Buffer Overflow Vulnerability of WebDAV on IIS Web Server

Hye-Young Chang[0], Uk-ki Min, Seong-Je Cho
Division of Information and Computer Science, Dankook University

요 약

최근 소프트웨어 보안취약성 분석의 일환으로 프로그램 소스가 없고 기계어 프로그램만 제공되는 소프트웨어의 보안 취약점을 분석하는 연구가 많이 수행되고 있다. 본 논문에서는 MS 윈도우 2000 운영체제의 IIS 웹서버를 대상으로, WebDAV의 한 버퍼 오버플로우 취약점을 공격하여 취약성을 재현한 후, 디버거 및 역공학 도구를 사용하여 해당 보안 취약점을 가진 코드를 분석하는 방법을 제시하였다. 본 연구 결과는 취약성분석 절차 방법 및 신뢰성 있는 소프트웨어 개발에 기여할 수 있을 것으로 기대된다.

1. 서론

소프트웨어 코드에 대한 취약성은 오라클과 같이 수년간에 걸쳐 테스트를 거친 상용 소프트웨어 제품에서도 계속 발견되고 있다. 여러 연구 결과에 의하면 개발자들이 프로그램을 제작할 때 동일한 실수를 반복하며, 그 중 상당 부분은 개발 시의 부주의로 인해 오류가 발생한다. 대부분의 공격 또한 공격자가 새로운 오류를 찾아 공격을 성공하기보다는 이미 알려져 있는 취약점을 다시 악용하는 경우가 많다고 알려져 있다. 따라서 프로그래머들이 주의를 기울여서 소프트웨어를 개발하기만을 바라는 것은 한계가 있으며, 완성된 소프트웨어에 대한 보안취약성을 분석하여 보안 결함을 줄이는 것이 필요하다.

보안취약성은 공격의 원천이며, 공격자가 이용할 수 있는 상용제품의 보안취약성의 양은 매우 많다. 정보보호는, 각 소프트웨어에 대한 보안취약성의 분석을 통하여 보안취약점을 탐지해 내며, 탐지된 보안취약점을 점검하여 이를 악용하는 보안사고를 방지하는 것이라고 볼 수 있다. 즉, 소프트웨어의 보안취약성을 분석하여 프로그램에 존재하는 보안취약점을 재연하고 발견·분석하는 체계적인 방법이 요구되어진다.

소프트웨어 보안 취약성 분석기술은 소프트웨어에 존재하는 보안취약성을 분석하여 보안취약점을 발견하는 기술로 소프트웨어 신뢰도를 향상시켜 주는 주요 기반 기술이다. 일부 취약성 분석 기술은 개발이 완료된 소프트웨어를 대상으로 하기 때문에 역공학 기술과도 관련 있으며, 소프트웨어 결함이 보안사고와 연관될 수 있는지를 실제의 보안사고에 앞서 판단하는 심도 있는 예측을 하는 보안사고 예방분야이기도 하다.

본 논문은, MS 윈도우 서버 운영체제 상에서의 기계어 프로그램인 IIS 웹서버를 대상으로 보안 취약점 자료를 수집하여 보안 취약성을 재연하는 방법을 제시하였으며, 다음으로 디버깅, 역공학 등의 기술을 이용하여 재연된 보안 취약점 발생 과정을 추적하고 취약점 관련 부분을 체계적으로 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 개략적으로 살펴보고, 3장에서 IIS 웹서버의 구조 및 WebDAV 취약점에 대해 기술한다. 4장에서는 그 취약성을 재연시켜보고 IIS의 취약점을 분석한다. 5장에서는 대응 방법을 간단하게 기술하고, 6장에서 결론을 내린다.

2. 관련 연구

소프트웨어의 취약점을 분석하는 방법으로는 정적 분석(static analysis), 동적 분석(dynamic analysis), 소프트웨어 결함주입(fault injection), 역공학(reverse engineering) 등의 기법이 사용된다.

정적 분석은 프로그램 소스코드를 자동화된 도구를 이용하여 분석하는 것을 의미한다[1]. 정적 분석 자동화 도구는 컴파일러와 비슷하게 동작하며, 코드 사이의 불일치, 타입 검사, 모델 검사, 흐름제어, 주석, 프로그램 값 분석 등을 수행하여 육안으로 판독하기 어려운 취약점과 취약성을 일으킬 수 있는 프로그램 부분들을 탐지·파악한다[2]. 이 방법은 프로그램 전체를 총괄적으로 분석하므로 실행 가능한 일반적인 경우에 대해서 적용 가능하다는 데에 장점이 있다.

정적 분석이 중요한 접근 방식이긴 하지만 모든 것을 해결해 줄 수 있는 것은 아니다. 프로그램을 직접 수행하면서 체계적으로 테스트하는 방법과 자세하게 평가하는 방법을 완전히 대체할 수는 없다. 정적 분석은 단지 프로그램 작성 시에 부주의로 발생할 수 있는 문제 부분들을 미리 발견하게 도와주는 것이다[2]. 또한 정적분석 도구를 사용할 때 많은 파싱 오류 및 false positive를 접하게 되며, 그러한 문제를 해결하기 위해 추가적인 분석시간이 많이 필요하게 된다[3].

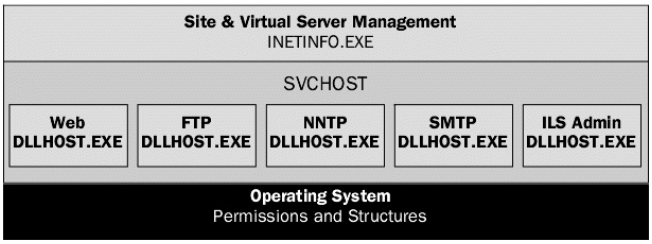
정적분석이 소스코드를 보고 분석하는 것이라면 동적 분석은 프로그램을 실제로 실행 시켜가면서 분석하는 방법이다[4]. 동적 분석에서는 프로그램에 입력값을 주면서 실행시키거나 프로그램의 기능을 차례차례 혹은 임의적으로 실행시켜가면서 분석한다. 최근에 많이 연구되는 있는 소프트웨어 결함주입 기법과 동일하다고 볼 수 있다. Purify, Fuzz, Property-based Test 등이 동적 분석 방법의 예이다[5][6]. Purify의 경우 메모리 접근 과정에 발생한 에러를 분석하는데, 이를 위해 컴파일된 코드에 메모리 참조 내용을 추적하는 코드를 추가함으로써 예외분석을 한다. Fuzz는 다양한 입력 조합을 사용하여 프로그램 수행 결과를 분석하는 방법이며, Property-based Test는 수행 중에 지켜져야 할 조건들을 상세히 기술(specification)하고 이를 어긴 경우를 발견하여 분석하는 방법이다.

3. IIS 웹서버 구조 및 WebDAV 취약점

한국정보보호 진흥원(www.kisa.or.kr)이 발표한 해킹 피해 결과를 보면 MS 윈도우 운영체제에 대한 해킹 피해 건수가 많은 부분을 차지하고 있다. 따라서 본 논문에서도 MS 윈도우 운영체제의 IIS 웹서버에 대한 보안취약성을 분석하였다.

3.1 IIS 웹서버 구조

IIS의 구조가 (그림 3-1)에 나타나있다. 사용자 요청에 대해, INETINFO.EXE 프로세스가 적절한 서비스(HTTP, FTP, SMTP 등)를 제공하기 위해 중간에서 각각의 프로세스에게 연결시켜 주는 역할을 하며, SVCHOST의 DLLHOST.EXE 프로세스에서 사용자에게 직접적인 서비스를 제공하게 된다[7].



(그림 3-1) IIS 구조

IIS와 응용 프로그램을 보호하기 위해서 관리자는 응용프로그램 보호 등급을 낮음(IIS 프로세스), 보통(풀링됨), 높음(격리됨) 중의 하나로 설정할 수 있다. '낮음(IIS 프로세스)'으로 설정될 경우, 서비스(취약한 함수)의 실행 주소공간이 IIS의 주소공간과 같기 때문에 잘못된 기능의 작동으로 IIS 전체가 영향을 받게 된다. 반면, '높음(격리됨)'으로 설정될 경우 IIS의 실행 주소공간과 서비스의 주소공간이 분리되어 작동하기 때문에 서비스가 문제를 유발시킨다 하더라도 IIS가 영향을 받지 않게 된다[8].

3.2 취약점 개요

본 논문에서는 IIS의 한 취약점 정보를 CVE(Common Vulnerabilities and Exposure) 중심으로 수집하였다. CVE란 컴퓨터 취약성에 대해서 표준화된 이름을 제공하기 위한 취약성 명명법을 말한다. 대상 취약점 정보가 아래 [표 3-1] 요약되어 있다.

취약점 이름	취약점 유형	공격 위험도
CAN-2003-0226	Buffer Overflow	높음

[표 3-2] 취약점 명명법

IIS 5.0이나 5.1에서, 웹 컨텐츠의 원격지 저술 및 관리를 허용하는 HTTP 프로토콜의 확장 기능인 WebDAV의 긴 요청과 HTTP 명령어를 나타내는 메소드중에서 'PROPFIND'이나 'SEARCH'와 함께 요청을 받게되면, ntdll.dll 파일의 불충분한 범위 체크(Bounds Checking)을 수행하는 함수를 호출하게 된

다. 이때, WebDAV 요청 패킷에 대해 적절한 범위 체크가 이루어지지 않게 되어 버퍼 오버플로우를 일으키게 된다[9].

위의 취약점은 CVE-2001-0508와 비슷하기는 하지만 'PROPFIND'이나 'SEARCH'를 이용하지 않는다는 점에서 다르게 분류가 된다[10].

4. 취약성 재연 및 취약점 분석

4.1 취약점 공격

WebDAV의 긴 요청패킷과 함께 'SEARCH' 메소드를 요청함으로써 버퍼 오버플로우를 유발, IIS가 중지되어 정상적인 서비스가 제공되지 않게 된다. 실제 취약점을 재연하기 위해, 서비스 팩 및 패치가 전혀 이루어지지 않은 Windows 2000 Server IIS 5.0에서 실행시켰다. 공격코드의 핵심 부분은 (그림 4-1)과 같다.

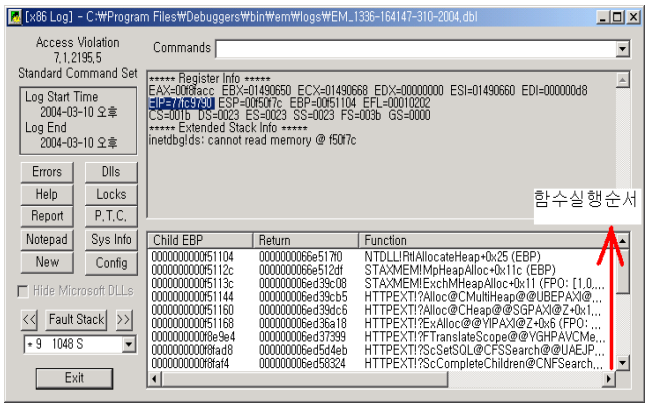
```
$over=$ch x $ll; #string to overflow
$xml='<?xml version="1.0"?> <D:searchrequest xmlns:D="DAV:"><D:sql>SELECT DAV:displayname from SCOPE('".$over."')</D:sql></D:searchrequest>'."\n";
$l=length($xml);
 $req="SEARCH / HTTP/1.1\nContent-type:
 text/xml\nHost: $host\nContent-length: $l\n\n$xml\n\n";
syswrite($socket,$req,length($req));
            ... ... ... ... ( 생      략 ) ... ... ... ...
do vv(126000,"V");
sleep(1);
do vv(126000,"V");
```

(그림 4-1) 해당 취약점 공격 코드

4.2 취약점 분석

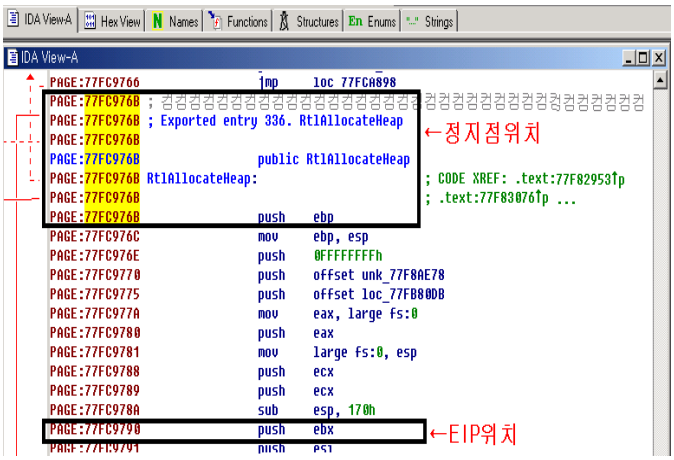
공격 코드를 실제 실행하여 예외상황을 유발시킨 취약점을 분석하기 위해서 여러 디버거 도구가 필요하게 된다. SoftICE와 같은 동적 분석 도구, IDA Pro와 같은 정적 분석 도구, 윈도우 디버거(WinDBG), 비주얼 스튜디오 디버거등을 사용하게 된다. 이들 디버깅 도구 및 역공학 도구를 사용하는 이유는 소프트웨어 취약성 분석 대상 응용들에 대한 소스 코드는 없고 기계어 코드가 제공되기 때문이다.

이 취약점으로 인해 IIS가 공격을 받아 문제가 생기면 관리자의 설정에 따라 중지되거나 서비스가 재시동 되게 된다. 기본적으로 IIS 서비스가 재 시작하기 때문에 사용자의 입장에서는 잠깐 서비스가 중단된 정도로만 느끼게 된다. 취약점을 분석하기 위한 정보를 획득하기 위해서 WinDBG의 Exception Monitor로 IIS 프로세스를 Attach하여 공격한 후 기록된 로그정보를 조사한다. (그림 4-2)의 로그파일은 오류가 발생된 함수뿐만 아니라 EIP값 77fc9790까지 보여준다.



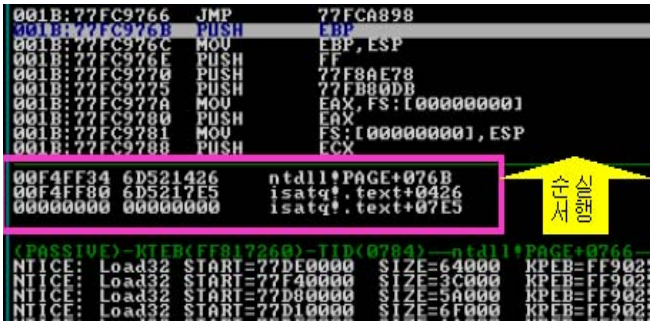
(그림 4-2) WinDBG을 이용한 로그파일분석

(그림 4-2)를 보면 HTTPEXT의 여러함수를 거친 후 STAXMEM의 ExchMHeapAlloc함수 그리고 STAXMEM의 MpHeapAlloc함수 호출 후에 NTDLL의 RtlAllocateHeap함수 부분에서 오류가 났음을 알 수 있다. 그러나 설치된 디버깅 정보와 윈도우와의 심벌정보가 맞지 않을 경우가 있기 때문에 정확하게 어떠한 함수에서 오류가 발생했는지를 확인하기 위해 IDA Pro를 사용하여 디버거에서 사용할 정지점(Break Point) 위치를 확인한다. 정지점은 예외상황이 발생한 EIP보다 앞쪽에서 설정되어야 하기 때문에 (그림 4-3)에서 획득한 77fc976b주소에 정지점을 설정한다.



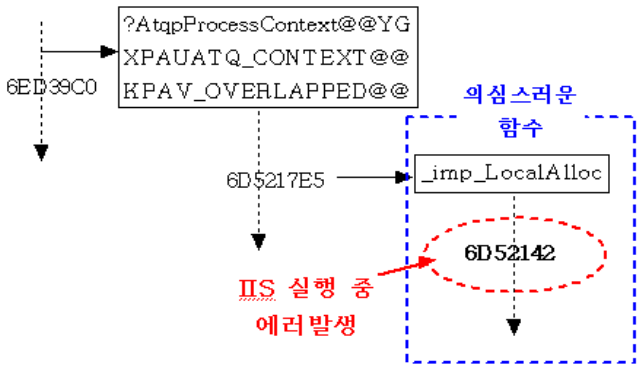
(그림 4-3) IDA Pro를 이용한 정지점 추측

정지점을 설정하고 공격을 다시 하면 SoftICE를 통하여 (그림 4-4)와 같은 스택 구조를 확인할수 있다.



(그림 4-4) SoftICE를 이용하여 스택보기

위 (그림 4-4)의 스택 정보를 분석한 후, IDA Pro로 실제 함수호출의 흐름을 추적하였다. ((그림 4-5)참조)



(그림 4-5) IIS 스택정보를 이용한 역추적

5. 대응 방법

관련 취약점을 예방하기 위해 레지스트리 편집기를 사용하여 WebDAV를 비 활성화 할 수도 있지만 그 경우, 웹 컨텐츠의 원격지 저술 및 관리 등의 기능을 이용할 수 없다. 그러므로, IIS의 사용 시 마이크로소프트사로부터 주기적으로 패치하여 해결 할 수 있다. 혹은 프로그램 작성 시 프로그래머가 주의를 기울이거나 자동으로 범위를 체크하는 도구의 적용이 필요로 한다.

6. 결론

취약성을 분석하기 위해 사용되는 주요 방법들은 정적 분석 및 동적 분석, 결함주입 등이다. 이들 각 방법들은 주로 소스가 있는 프로그램이나 규모가 크지 않은 프로그램, 복잡도가 낮은 특정한 취약성을 가진 프로그램들을 대상으로 하고 있어 대규모 기계어 프로그램 분석에는 적합하지 않다.

본 논문에서는 MS 윈도우 2000 상에서 기계어 코드만 주어진 IIS 웹서버 프로그램을 대상으로, WebDAV의 버퍼 오버플로우 취약점을 분석하는 절차에 대한 연구를 수행하였다. 대부분의 상용 소프트웨어가 실행 코드만 주어진다는 점과 관련 연구가 많이 이루어지지 않았다는 점에서 의의가 있다. 결과적으로, 보안 전문가가 이를 활용할 경우, 보안 사고의 발생 가능성을 미연에 방지할 수 있는 기반 자료로 활용 가능하다고 보여진다.

[참고 문헌]

[1] D. Evans and D. Larochelle. "Improving Security Using Extensible Lightweight Static Analysis.", IEEE Software, Jan/Feb 2002.

[2] E. Haugh and M. Bishop, "Testing C Programs for Buffer Overflow Vulnerabilities", Proceedings of the 2003 Symposium on Networked and Distributed System Security (Feb. 2003)

[3] Pete Broadwell and Emil Ong, "A Comparison of Static Analysis and Fault Injection Techniques for Developing Robust System Services.", class project paper, May 2002

[4] Lo R., Kerchen P., Crawford R., Ho W., Crossley J., Fink G., Levitt K., Olsson R., Archer M.: Towards a Testbed for Malicious code detection. *COMPCON Spring '91. Digest of Papers.* San Francisco, CA, pp. 160-166, Feb.-Mar. 1991.

[5] David Wagner, Jeffrey S. Foster, Eric A. Brewer, and Alexander Aiken, "A First Step Towards Automated Detection of Buffer Overrun Vulnerabilities", Networking and Distributed System Security Symposium 2000, San Diego, California. February 2000.

[6] G. Fink and M. Bishop, "Property Based Testing: A New Approach to Testing for Assurance", White paper, Apr. 2000. (WindowsSecurity.com)

[7] Microsoft, "Core IIS Administraion", http://www.microsoft.com

[8] 김소영 역, "웹 서버 운영자를 위한 IIS 5.0", 사이버 출판사, 2001.7

[9] CAN-2003-0226, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0226

[10] CVE-2001-0508, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0508