상용 exploit탐지 제품의 오탐율 실험결과

홍 진 아*, 김 광 조*,**, 김 용 대*,***
한국과학기술대학원 *정보보호대학원, ** 전산학부, *** 전자공학과

An experimental result of false negative rate in commercial exploit-detection products

Jina Hong*, Kwangjo Kim*,**, Yongdae Kim*,***
*Graduate School of Information Security, **School of Computing, ***Electrical Engineering, KAIST

요 약

제로데이 취약점을 사용하는 exploit은 APT, exploitkit 등 많은 공격들에서 사용되고 있다. 이러한 exploit들은 signature가 존재하지 않기 때문에 국내외 많은 상용 exploit탐지 제품들이 실행 파일, 혹은 첨부 파일 등의 행위에 대한 분석을 통해서 false positive가 높더라도 탐지하려는 노력을 하고 있다. 본 연구에서는 이러한 exploit을 탐지하고자하는 국내외 4종의 제품을 동적 분석을 수행하여 상용 exploit탐지 제품이 알려져 있는 공격방법들을 얼마나 잘 처리할 수 있는지 확인하였다.

Ⅰ. 서 론

보안을 위협하는 악성 코드가 갈수록 지능화, 다양화되며, 이러한 경향의 하나로 공격자들은 은밀하게 수행되는 exploit이 성행하고 있다. 이러한 exploit은 정부와 일반 기업과 같은 특정 대상을 목표로 하는 Advanced Persistent Threat(APT)공격부터 drive-by-download 방식으로 대량 배포되는 exploitkit 까지 다양한 목적을 위한 공격방식으로 악용된다. 본 논문에서 대상으로 하는 exploit은 Adobe Reader, Adobe Flash, Microsoft Word 와 같은 프로그램의 취약점을 사용하여, 악성코드를 실행하는 공격방식으로 한정한다. exploit은 과거 exe확장자를 가지는 악성코드와는 공격방식이 다르기 때문에, 기존의 시그니쳐 기반의 악성코드 탐지 방식은 exploit을 탐지하는데 한계가 있다.

최근 국내외 많은 백신 업체에서는 제로데이 취약점을 사용한 악성코드도 탐지 할 수 있는 제품을 개발하고 있다. Microsoft사의 EMET, Malwarebytes의 ant-exploit, Palo Alto Networks 의 Traps, SurfRight의 HitmanPro.Alrt 등과 같은 exploit 탐지 제품이 있고, 이에 발맞추어 국내의 백신업체도 탐지 프로그램을 출시하였다.

본 연구에서는 4개의 상용 exploit탐지 제품의 보안성, 특히 오탐율(false negative rate)에 대하여 평가하였다. 탐지기법의 탐지율을 알아보기 위하여 공개

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보 통신·방송 연구개발사업의 일환으로 수행하였음. [B0101-16-1270 , 생체모방 알고리즘(Bio-inspired Algorithm)을 활용한 통신기술 연구]
* 한국과학기술대학원 정보보호대학원 (jina3453@kaist.ac.kr)
** 한국과학기술대학원 전산학부 (kkj@kaist.ac.kr)
*** 한국과학기술대학원 전자공학과(yongdaek@kaist.ac.kr)

된 모의해킹 툴을 사용하여 상용 exploit탐지 제품을 블랙박스 테스팅을 수행하였다. 이를 통해 각 제품의 오탐율을 측정하였고, 탐지 결과를 상세히 분석하였다.

Ⅱ. Exploit과 탐지 방법

2.1 Exploit

이 논문에서는 설명의 편의를 위해 exploit을 다음과 같은 3 단계로 구분하였다. 첫 번째 단계로 취약점 존재는 프로그램 구현, 설계상의 실수로 나타나는 소프트웨어적인 버그이다. C 관련 언어를 사용한 프로그램에서는 메모리 에러와 관련된 버그가 있다. [1].

두 번째 단계로 페이로드 전달 기법은 취약점으로부터 페이로드를 실행하기 위한 기술이다. 공격자는 페이로드 전달기법을 활용할 수 있는 입력 값을 프로그램에 주어 control flow 혹은 데이터를 변조한다. 대표적으로 code-reuse attack이 있다.

세 번째 단계인 페이로드는 공격자가 최종적으로 수행하고자 하는 코드 혹은 프로그램이다. 대표적으로 쉘을 실행시키거나 권한상승, 서비스 거부 등의 행위가 있다.

2.2. Control Flow Integrity

Control Flow Integrity(CFI)[5]는 프로그램의 실행 흐름이 미리 지정된 Control Flow Graph(CFG)만 따르도록 규정하는 기술이다. CFI를 사용하여 최근 많이 사용되는 페이로드 전달기법인 return-oriented programming(ROP)[6]을 막을 수 있다. ROP는 일반적인 프로그램의 실행과는 다른 특유의 실행 흐름을 가지고, 사전에 정의된 실행 흐름이 아니기 때문이다.

Ⅲ. 분석 대상 및 분석 방법

본 논문에서 호스트 레벨에서 일어나는 시스템의 행동을 실시간으로 모니터링하고 제로데이 공격 탐지를 목표로 하는 국내 외 4개의 제품에 대해서 동적 분석을 수행하였다. 4개의 제품은 편의상 G1,G2,D1,D2으로 구분하였다.

본 연구에서는 공개된 모의해킹 툴[7,8]을 사용하여 선정한 상용 exploit탐지 제품의 탐지율을 측정하였다. 실험의 입력 값은 모의해킹 툴에서 제공하는 exploit이며, 오탐율을 측정하기 위하여 공격행위만 선정하였고 상용 exploit탐지 제품의 커버리지를 높이기 위하여 다양한 종류의 공격 행위를 선정하였다. 상용 exploit제품의 로그 정보를 사용하여 exploit을 공격으로 진단한 이유, 탐지 시기, 미탐 이유를 분석하였다.

IV. APT공격 탐지 제품에 대한 동적 분석

4.1. testset : 메타스플로잇

상용 exploit탐지 제품의 보안성을 다방면으로 측정하기 위하여 서로 다른 취약점, 페이로드 전달기법과 페이로드를 선정하였다. 이 논문에서는 메타스플로잇에서 제공하는 모듈을 사용하였다. 메타스플로잇[7]은 모의해킹을 쉽게 할 수 있도록 여러 프로그램을 모아놓은 툴이며, 취약점을 공격할 수 있는 페이로드 전달 기법을 모듈로 제공한다. 공격 모듈은 drive-by-download 방식으로 배포되며, 실제 공격 상황과 유사하다. 메타스플로잇 공격모듈의 페이로드는 사용자가 지정할 수 있기 때문에 각 어플리케이션에서 실행 가능한 4개의 페이로드를 선정하였다. payload1과 3은 윈도우즈의 기본 커멘드 쉘을 실행하고, payload2와 4는 메타스플로잇에서 제공하는 메타프리터를 실행한다. 그 외 테스팅시 사용한 모듈의 분석과 exploit 탐지 제품의 동적 분석 결과은 [표 1]에 요약하였다.

4.2. testset : Hitman Exploit Test tool

탐지율 측정을 위해 사용한 다른 공개된 모의해킹 툴은 Hitman 의 Exploit Test Tool[8]이다. 해당 툴을 Internet Explorer(IE)의 취약점 17개의 페이로드 전달 기법 중 사용자가 선택한 페이로드 전달 기법을 사용하여 계산기 수행으로 공격을 제공한다. [표 2]에서 Exploit Test Tool[8]을 사용하여 페이로드 전달기법과 이에

〔표 1〕 메타스플로잇을 사용한 exploit 탐지 제품의 동적 분석 결과 (O : 공격 탐지, X : 공격 미탐)

Module	취약점	페이로드 전달 기법	페이로드	G1	G2	D1	D2
Module1	Information leak (CVE-2013-0074)	ROP, Reflective dll injection	payload 1	O	O	X	X
			payload 2	O	O	X	X
Module2	Heap corruption (CVE-2013-0634)	Heapspray, ROP, Reflective dll injection	payload 1	O	O	X	X
			payload 2	O	O	X	X
Module3	Integer Overflow (CVE-2013-2551)	Heapspray, ROP, Reflective dll injection	payload 1	O	O	X	X
			payload 2	O	O	X	X
Module4	Use After Free (CVE-2012-4969)	Heapspray, ROP, Reflective dll injection	payload 1	O	O	X	X
			payload 2	O	O	X	X
Module5	Integer Overflow (CVE-2014-0569)	ROP	payload 1	O	O	X	X
			payload 2	O	O	X	X
Module6	Use After Free (CVE-2015-0313)	ROP	payload 1	O	O	X	X
			payload 2	O	O	X	X
Module7	findClass method (CVE-2013-0422)	Java Reflection	payload 3	O	O	O	O
			payload 4	O	O	O	O
Module8	Array Indexing (CVE-2013-2465)	Memory Corrupting	payload 3	O	O	O	O
			payload 4	O	O	O	O

따라 상용 exploit탐지 제품의 탐지율 변화를 요약하였다.

3.3 G1,G2 프로그램의 결과 분석

G1과 G2는 모든 메타스플로잇으로 수행한 모의 해킹 공격을 탐지하여 막았다. 상용 exploit 제품의 로그값 분석 결과, G1과 G2는 exploit의 페이로드 전달 기법을 대부분 탐지하였고, 일부 공격은 페이로드 수행을 탐지하였으며, 어떠한 공격에 대해서도 취약점이 무엇인지는 검사하지 않았다.

〔표 1〕의 Module1부터 6은 다른 취약점과 유사한 페이로드 전달 기법을 가지지만 G1과 G2는 해당 모듈의 페이로드 전달 기법을 탐지하였다. 이를 통해 페이로드 전달 기법에서 공격으로 판단되면 이후의 페이로드는 검사하지 않는 것을 알 수 있다.

메타스플로잇에 제공하는 페이로드 전달 기법은 고전적인 기술이라 휴리스틱 탐지방법으로 쉽게 탐지가 가능하였고, 이는 〔표 1〕을 통해 알 수 있다. 추가적인 실험으로 발전된 페이로드 전달 기법을 제공하는 공개된 모의해킹 툴인 Hitman 의 Exploit Test Tool〔8〕을 사용하여 상용 exploit제품의 탐지율을 측정하였다. 실험 결과는 〔표 2〕과 같다. G1은 17개의 모든 페이로드 전달 기법을 EAF 탐지 기법으로 방어 하였다. G2는 17개 중 9개를 탐지하지 못하였다. 그 이유는 Exploit Test Tool〔8〕에서 제공하는 진화된 ROP는 방어기법을 우회할 수 있도록 ROP에 call명령어를 추가하고, stack pivoting을 하지 않는 페이로드 전달 기법 을 수행하였기 때문이다.

G2는 Module7과 8에서 payload3과 4으로 exploit을 수행하였을 때 서로 다른 페이로드 탐지 결과를 보였다. 이전의 Module은 모두 PE파일로써 ROP, Heapspray와 같은 페이로드 전달 기법을 가지는 반면 Module 7과 8은 JVM의 ecurity manager를 우회하는 페이로드 전달기법을 가진다. G1와 G2는 security manager를 우회하는 페이로드 전달기법을 잡지 못하였지만, JVM밖에서 수행되는 payload3과 4는 악성행위로 탐지하였다.

〔표 2〕 페이로드 전달 기법 탐지율 측정 (O : 공격 탐지, X : 공격 미탐)

페이로드 전달 기법	G1	G2	D1	D2
Stack pivot	O	O	X	X
Stack Exec	O	O	X	X
DEP	O	O	X	X
Unpivot Stack	O	X	X	X
ROP - WinExec()	O	O	X	X
ROP - VirtualProtect()	O	O	X	X
ROP-CALL preceded VirtualProtect()	O	X	X	X
ROP-NtProtectVirtualMemory()	O	O	X	X
ROP - system() in msvcrt	O	X	X	X
ROP-VirtualProtect() via CALL gadget	O	X	X	X
ROP - WinExec() via anti-detour	O	X	X	X
IAT Filtering	O	X	X	X
NULL Page	O	O	X	X
SEHOP	O	O	X	X
Heap Spray - Single byte NOP-sled	O	X	X	X
Heap Spray - Polymorphic NOP-sled	O	X	X	X
Heap Spray - JavaScript	O	X	X	X

4.4. D1, D2 프로그램의 결과 분석

메타스플로잇으로 모의해킹을 수행한 결과 D1과 D2는 일부 모의 해킹 공격을 막아내었으며 일부 공격은 페이로드를 탐지하였다. 그러나 수행한 모의 해킹 testset으로는 취약점과 페이로드 전달 기법을 탐지한 결과를 찾지 못하였다.

D1은 페이로드 실행 시 발생하는 파일과 관련된 행위를 접근 제어한다. Module7은 java.exe 프로그램이 cmd.exe 에 접근에 악성으로 판단되었고, Module8은 java.exe프로그램이 java.exe 프로그램에 접근하여 악성으로 판단되었다.

D1과 D2가 탐지하지 못한 Module1부터 Module7은 IE에 active-X로 동작하는 Flash, sliverlight 등의 취약점을 공격한다. 공격 모듈은 IE의 Active-X에 접근하지만, D1의 행위가 IE에서 IE로 접근하는 것으로 식별하여 악성 행위로 판단하지 않았다. 만약 접근 제어 이후에 CFI를 수행하였다면 다른 결과를 보였을 수 도 있지만, 파일 접근 제어에서 정상으로 판단한 이후에는 CFI를 수행하지 않는다.

Hitman 의 Exploit Test Tool〔8〕을 사용한 공격은 아무것도 탐지하지 못하였다. D1과 D2는 계산기를 악성 페이로드로 판단하지 않기 때문에, 파일 접근 제어 결과 정상 행위로 진단되며, 어떠한 CFI기술도 호출되지 않았다.

V. 결 론

기존의 시그니쳐 기반 백신으로는 진화하는 공격을 막기에 한계가 있기 때문에, 행위기반탐지 방식과 CFI가 제안되었다. 상용 exploit탐지 제품의 오탐율을 알아보기 위하여 공개되어 있는 모의 해킹 툴을 사용하여 오탐율을 측정하였다. 그 결과 몇 개의 제품은 파일을 중심으로 하는 탐지방법에서 크게 벗어나지 못하였고, 탐지 방법을 알면 쉽게 우회가 가능하다는 한계점을 파악하였다.

참 고 문 헌

〔1〕Debbabi, Mourad, et al. "Dynamic monitoring of malicious activity in software systems." Proceedings of the Symposium on Requirements Engineering for Information Security 2001.

〔2〕Veldman, Frans. "Heuristic anti-virus technology." International Virus Bulletin Conference. 1994.

〔3〕박남열; 김용민; 노봉남. 우회기법을 이용하는 악성코드 행위기반 탐지 방법. 정보보호학회논문지, 2006.

[4]Szekeres, Laszlo, et al. "Sok: Eternal war in memory." Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013.

[5]Abadi, Martín, et al. "Control-flow integrity." Proceedings of the 12th ACM conference on Computer and communications security. ACM, 2005.

[6]H. Shacham. The geometry of innocent flesh on the bone:return-into-libc without function calls (on the x86). In the 14th ACM conference on Computer and communications security (CCS), 2007.

〔7〕 metasploit,https://www.metasploit.com

〔8〕surfright, http://www.surfright.nl/en