Disclosure to or reproduction
for others without the specific
written authorization of AhnLab
is prohibited.

Copyright (c) AhnLab, Inc.
All rights reserved.

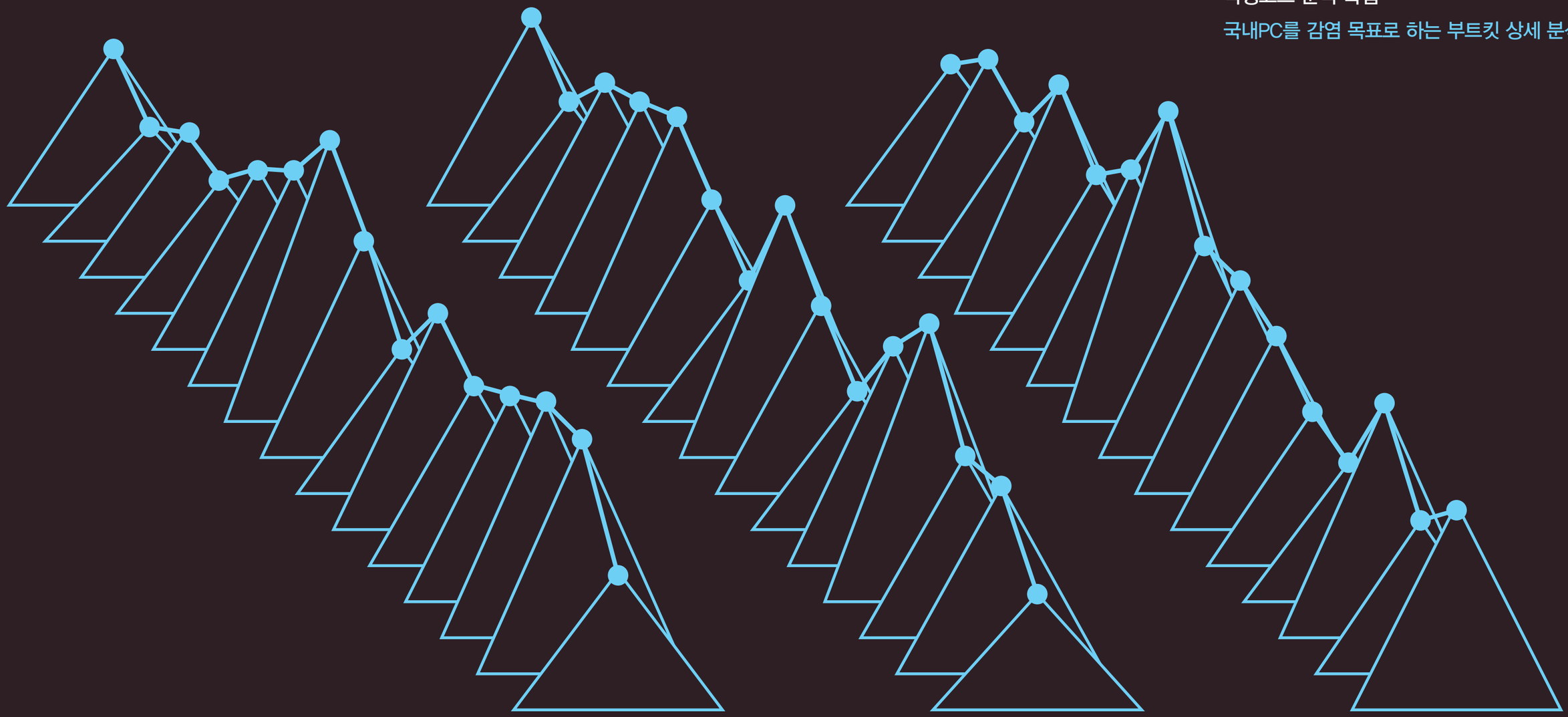
ASEC REPORT

VOL.22 | 2011.11

안철수연구소 월간 보안 보고서

악성코드 분석 특집

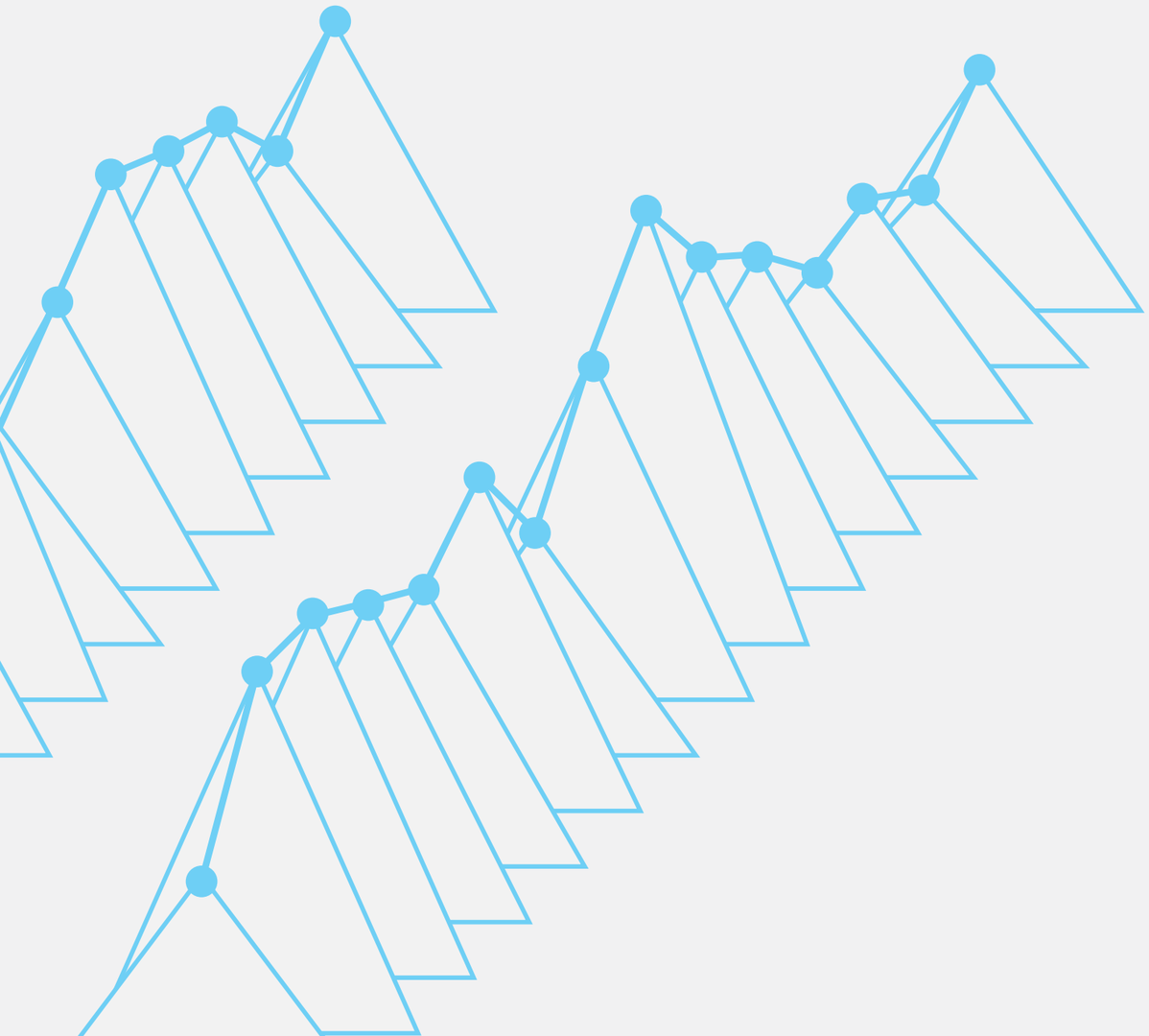
국내PC를 감염 목표로 하는 부트킷 상세 분석



Ahn 안철수연구소

AhnLab Security Emergency response Center

ASEC (AhnLab Security Emergency response Center)은 악성코드 및 보안 위협으로부터 고객을 안전하게 지키기 위하여 보안 전문가로 구성된 글로벌 보안 조직입니다. 이 리포트는 ㈜안철수연구소의 ASEC에서 작성하며, 매월 발생한 주요 보안 위협과 이슈에 대응하는 최신 보안 기술에 대한 요약 정보를 담고 있습니다. 자세한 내용은 안랩닷컴(www.ahnlab.com)에서 확인하실 수 있습니다.



CONTENTS

01. 악성코드 동향
a. 악성코드 통계

악성코드 감염보고 Top 20

2011년 10월 악성코드 통계 현황은 다음과 같다. 2011년 10월의 악성코드 감염 보고에서는 JS/Agent가 가장 많았으며, Textimage/Autorun과 JS/Iframe이 그 뒤를 이었다. 신규로 Top20에 진입한 악성코드는 Swf/Uqust, Exploit/Cve-2011-2140, JS/Mult 등 총 8건이다.

순위	등락	악성코드명	건수	비율
1	▲4	JS/Agent	737,610	20.0 %
2	▼1	Textimage/Autorun	542,815	14.7 %
3	▲1	JS/Iframe	522,958	14.2 %
4	NEW	Swf/Uqust	222,551	6.0 %
5	▼3	JS/Redirector	172,029	4.7 %
6	NEW	Exploit/Cve-2011-2140	153,102	4.1 %
7	▼1	Dropper/Malware.495616.HT	121,027	3.3 %
8	▲3	Swf/Agent	117,025	3.2 %
9	—	Win-Trojan/Downloader.217088.AE	113,868	3.1 %
10	▲3	Als/Bursted	108,345	2.9 %
11	▼3	Win32/Induc	107,377	2.9 %
12	NEW	JS/Mult	102,328	2.8 %
13	▼1	Win32/Palevo1.worm.Gen	99,913	2.7 %
14	NEW	Win-Trojan/Hupigon.425984.BU	99,120	2.7 %
15	NEW	Html/Flasher	95,032	2.6 %
16	NEW	Swf/Cve-2010-2884	86,617	2.3 %
17	▲2	Swf/Exploit	84,813	2.3 %
18	▼1	Win32/Olala.worm	72,667	2.0 %
19	NEW	Html/Popupper	68,539	1.9 %
20	NEW	RIPPER	62,817	1.6 %
			3,690,553	100.0 %

[표 1-1] 악성코드 감염보고 Top 20

악성코드 대표진단명 감염보고 Top 20

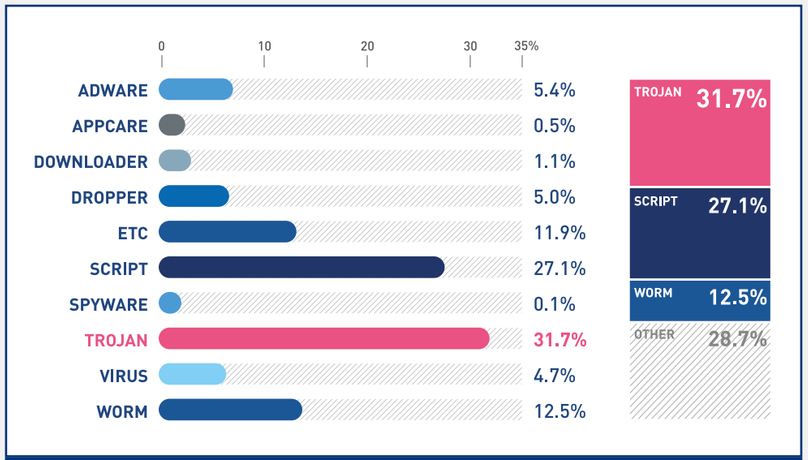
[표 1-2]는 악성코드별 변종 종합 감염 보고 순위를 악성코드 대표 진단명에 따라 정리한 것이다. 이를 통해 악성코드의 동향을 파악할 수 있다. 2011년 10월의 감염 보고 건수는 JS/Agent가 총 737,610건이며 Top20 중 13.6%의 비율을 점유해 1위인 것으로 나타났다. Win-Trojan/Agent가 578,728건/10.6%로 2위, Textimage/Autorun이 542,921건/10.0%로 3위를 기록했다.

순위	등락	악성코드명	건수	비율
1	▲13	JS/Agent	737,610	13.6 %
2	▼1	Win-Trojan/Agent	578,728	10.6 %
3	—	Textimage/Autorun	542,921	10.0 %
4	▲8	JS/Iframe	522,958	9.6 %
5	▼3	Win-Trojan/Downloader	360,016	6.6 %
6	—	Win-Trojan/Onlinegamehack	266,382	4.9 %
7	▼2	Win-Adware/Korad	237,270	4.4 %
8	▲1	Win32/Conficker	235,793	4.3 %
9	▼1	Win32/Virut	230,076	4.2 %
10	NEW	Swf/Uqust	222,551	4.1 %
11	▼1	Win32/Autorun.worm	220,533	4.1 %
12	▼5	Dropper/Malware	212,104	3.9 %
13	▲2	Win32/Kido	181,298	3.3 %
14	▼10	JS/Redirector	172,029	3.2 %
15	NEW	Exploit/Cve-2011-2140	153,102	2.8 %
16	NEW	Win-Trojan/Hupigon	119,262	2.2 %
17	NEW	Swf/Agent	117,025	2.2 %
18	NEW	Dropper/Agent	111,989	2.1 %
19	NEW	Als/Bursted	108,345	2.0 %
20	▼2	Win32/Induc	107,445	1.9 %
			5,437,437	100.0 %

[표 1-2] 악성코드 대표진단명 감염보고 Top 20

악성코드 유형별 감염보고 비율

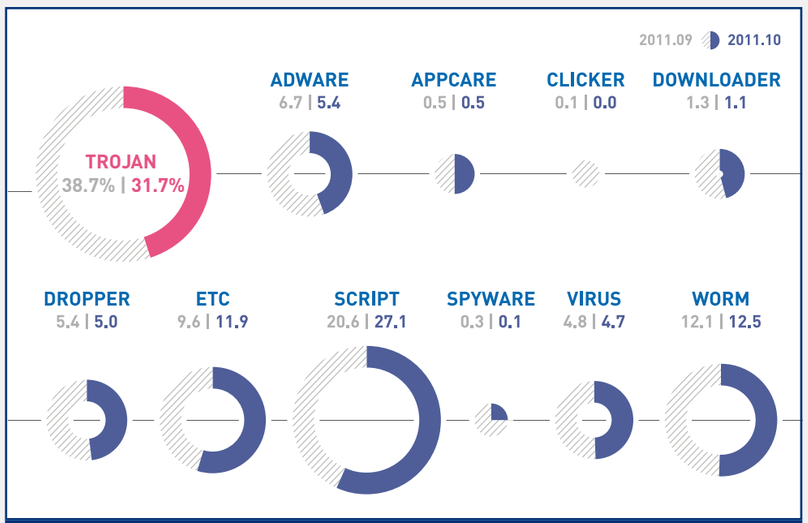
[그림 1-1]은 고객으로부터 감염이 보고된 악성코드 유형별 비율이다. 이것은 2011년 10월 한 달 동안 안철수연구소가 집계한 악성코드의 유형별 감염 비율을 분석한 결과다.2011년 10 월의 악성코드를 유형별로 살펴보면, 감염 보고 건수 비율은 트로이목마(TROJAN)가 31.7%로 가장 많았으며, 스크립트(SCRIPT)가 27.1%, 웜(WORM)이 12.5%로 그 뒤를 잇고 있다.



[그림 1-1] 악성코드 유형별 감염보고 비율

악성코드 유형별 감염보고 전월 비교

악성코드 유형별 감염 보고 비율을 전월과 비교하면, 스크립트와 웜은 전월에 비해 증가세를 보이고 있는 반면, 트로이목마와 애드웨어(ADWARE), 드롭퍼(DROPPER), 바이러스(VIRUS), 다운로더(DOWNLOADER), 스파이웨어(SPYWARE), 클리커(CLICKER)는 전월에 비해 감소했다. 애프케어(APPCARE)는 전월 수준을 유지하고 있다.



[그림 1-2] 악성코드 유형별 감염보고 전월 비교

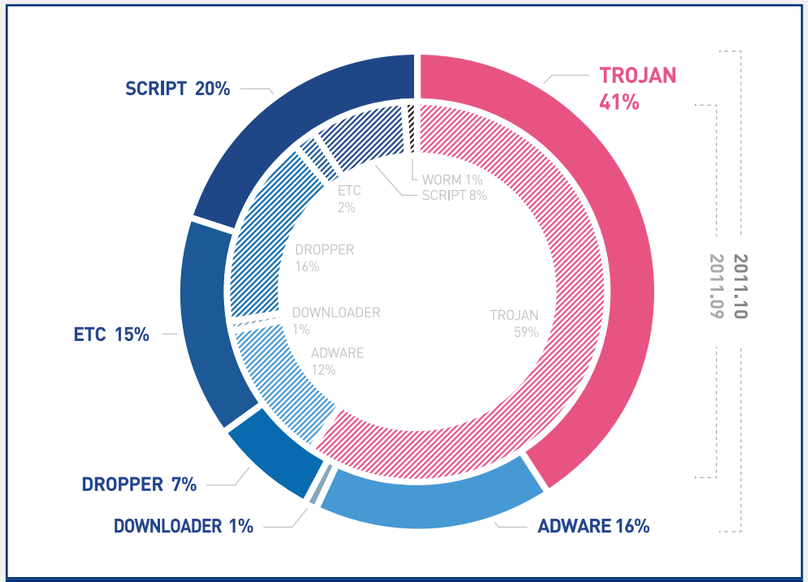
악성코드 월별 감염보고 건수

10월의 악성코드 월별 감염 보고 건수는 10,498,643건으로 9월의 악성코드 월별 감염 보고 건수 11,061,009건에 비해 562,366건이 감소하였다.

[그림 1-3] 악성코드 월별 감염보고 건수

신종 악성코드 유형별 분포

10월의 신종 악성코드 유형을 보면 트로이목마가 41%로 가장 많았고, 그 뒤를 이어 스크립트가 20%, 애드웨어가 16%를 점유하였다.



[그림 1-4] 신종 악성코드 유형별 분포

신종 악성코드 감염보고 Top 20

[표 1-3]은 10월에 신규로 접수된 악성코드 중 고객으로부터 감염이 보고된 악성코드 Top20이다. 10월의 신종 악성코드 감염 보고 Top 20은 SWF/Uqust가 222,551건으로 전체 24%로 1위였으며, Exploit/Cve-2011-21400이 153,102건으로 그 다음으로 많았다.

순위	악성코드명	건수	비율
1	SWF/Uqust	222,551	24.0 %
2	Exploit/Cve-2011-2140	153,102	16.5 %
3	Win-Trojan/Hupigon.425984.BU	99,120	10.7 %
4	Dropper/Agent.747008.E	55,724	6.0 %
5	Win-Adware/BHO.UBar.1339904	49,949	5.4 %
6	Win-Trojan/Agent.630272.O	49,524	5.3 %
7	Win-Trojan/Infostealer.434688	47,867	5.2 %
8	Win-Trojan/Agent.487677	42,989	4.6 %
9	Win-Adware/LineAd.266240	25,271	2.7 %
10	Win-Trojan/Agent.450560.CM	24,442	2.6 %
11	Win-Trojan/Korad.434176	19,425	2.1 %
12	Win-Dropper/LineAd.757734	17,392	1.9 %
13	Win-Trojan/Onlinegamehack.65541	17,390	1.9 %
14	Win-Trojan/Adload.883712	16,746	1.8 %
15	Win-Adware/BHO.WebSide.1841664	16,270	1.8 %
16	Dropper/Pasta.103500	15,929	1.7 %
17	Win-Adware/KorAd.446464.C	14,597	1.6 %
18	Win-Adware/KorAd.458752	13,566	1.5 %
19	Win-Trojan/Adload.179200.C	13,235	1.4 %
20	Win-Trojan/Agent.61440.BAZ	12,294	1.3 %
		927,383	100.0 %

[표 1-3] 신종 악성코드 감염보고 Top 20

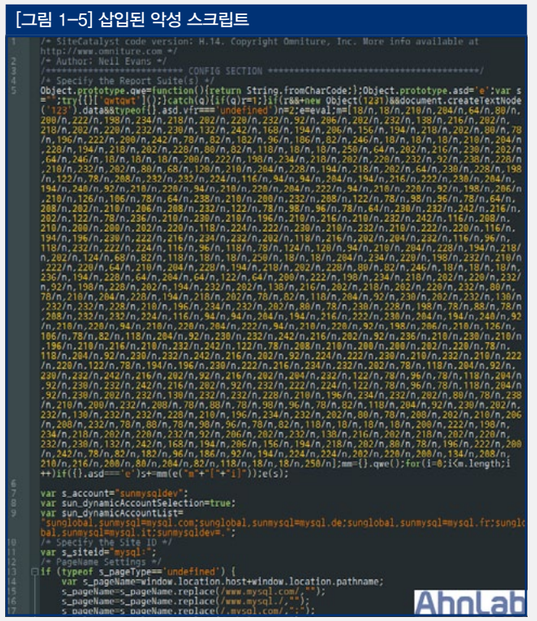
01. 악성코드 동향
b. 악성코드 이슈

MySQL 사이트에 삽입된 악성 스크립트

지난 9월 26일 발생했던 MySQL 웹 사이트 해킹 시 삽입되었던 악성 스크립트에 대하여 알아보자.

사이트(www.mysql.com)에 접속 시 연결되는 파일 중 한 군데에 [그림 1-5]와 같은 내용의 악성 스크립트가 삽입되어 있다.

[그림 1-5] 삽입된 악성 스크립트



삽입된 스크립트는 난독화되어 있으며 해제하면 [그림 1-6]과 같은 코드가 확인된다.

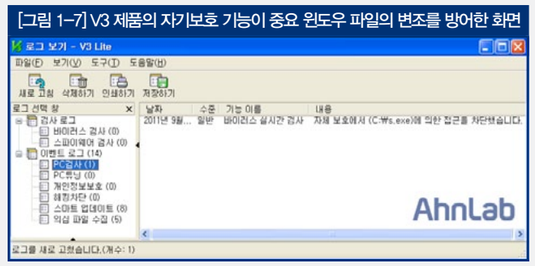
[그림 1-6] 난독화 스크립트를 풀었을 때 나오는 코드

위 페이지에 접속하면 어도비 플래시, 어도비 리더, 자바 등의 취약점을 이용한 악성코드에 감염될 수 있다.

해당 악성코드에 감염되면 아래와 같은 파일이 생성된다.

- C:₩WINDOWS₩system32₩5rc.dll
- C:₩WINDOWS₩system32₩imm32.dll
- C:₩WINDOWS₩system32₩version.dll
- C:₩Documents and Settings₩[사용자이름]₩Local Settings₩Temp₩xth1A.tmp.exe

[그림 1-7] V3 제품의 자기보호 기능이 중요 윈도우 파일의 변조를 방어한 화면



해당 사이트의 악성코드는 V3 제품군에서 다음과 같이 진단한다.

- Dropper/Win32.Mudrop
- Trojan/Win32.Patched
- Trojan/Win32.Agent

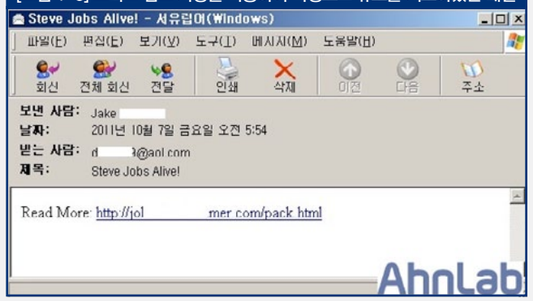
스티브 잡스 사망 관련 메일로 위장한 악성코드

연말연시 또는 유명 인사와 관련한 사건들이 발생할 때마다 악성코드 제작자들은 이러한 이슈들을 이용하여 악성코드 유포를 시도한다. IT 업계의 혁신을 주도한 인물로 평가되는 애플(Apple)사의 창업자인 스티브 잡스(Steve Jobs)의 사망을 악용해 악성코드 유포를 시도한 사례가 10월 7일 발생하였다. 이번에 발견된 악성코드는 [그림 1-8]과 같은 형태의 메일을 통해 유포되었으며, 해당 메일들은 다음의 제목 중 하나를 사용하였다.

- Steve Jobs Alive!
- Steve Jobs Not Dead!
- Steve Jobs: Not Dead Yet!
- Is Steve Jobs Really Dead?

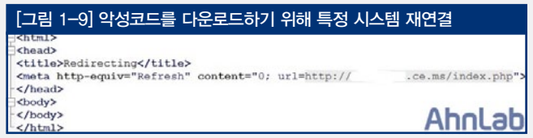
유포된 메일은 [그림 1-8]과 같이, 악성코드가 직접 첨부되어 있지 않고 메일 본문에 악성코드를 다운로드하기 위한 웹 사이트 링크가 존재하였다.

[그림 1-8] 스티브 잡스 사망을 악용하여 악성코드 유포를 시도하였던 메일



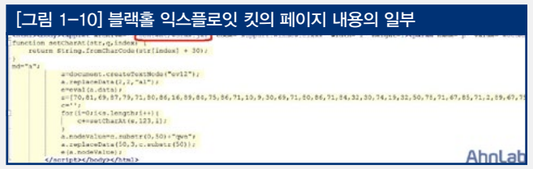
해당 메일의 본문에 존재하는 웹 사이트 링크를 자세히 분석해보면 [그림 1-9]와 같이 특정 시스템으로 재연결을 하도록 구성된 것을 볼 수 있다.

[그림 1-9] 악성코드를 다운로드하기 위해 특정 시스템 재연결



재연결되는 특정 시스템에는 웹 익스플로잇 툴 킷(Web Exploit Toolkit)으로 알려진 블랙홀 익스플로잇 킷(Blackhole Exploit Kit)이 설치되어 있으며, 사용자는 해당 툴킷으로 연결되어 [그림 1-10]과 같은 웹 페이지로 접속하게 된다. 이윽고 접속을 시도한 사용자의 시스템에는, 존재하는 다양한 취약점들이 악용되어 악성코드가 자동으로 다운로드되고 실행된다.

[그림 1-10] 블랙홀 익스플로잇 킷의 페이지 내용의 일부



해당 악성코드에 감염되면, 시스템에 존재하는 다양한 FTP 클라이언트 프로그램의 설정 파일에 기록된 FTP 서버 주소, 계정과 암호를 수집하여 외부로 전송한다. 또한 감염된 시스템에 USB와 같은 이동형 저장 장치가 존재 하면, 'MS10-046 Windows 셸의 취약점으로 인한 원격 코드 실행 문제점(2286198)' 취약점을 악용하는 바로 가기 파일(*.lnk)과 자신의 복사본을 이동형 저장 장치에 생성한다. 해당 취약점이 존재하는 윈도우를 사용 중인 환경에 감염된 USB를 연결할 경우, 윈도우 탐색기로 USB를 열어보는 것만으로도 악성코드에 감염된다. 이번에 발견된, 스티브 잡스 사망을 악용하여 유포된 악성코드들은 V3 제품군에서 다음과 같이 진단한다.

- Win-Trojan/Bredolab.44032.U
- Win-Trojan/Xema.85504.D
- Win-Trojan/Bredolab.884736.B

Smiscer Rootkit

최근에는 부트킷(Bootkit)처럼 감염된 PC의 MBR 영역을 조작하여 백신이 악성코드를 치료하더라도 부팅 시마다 계속 악성코드를 새로 만들어내는 방식이 자주 이용되고 있다. 초기의 Smiscer는 로드된 운영체제의 드라이버 중 하나를 감염시키고, 감염된 드라이버의 원본은 파일 볼륨을 하나 만들어 거기에 백업을 해두고 윈도우 하위 system₩config 폴더에 루트킷을 생성하였다. 이 루트킷이 파일 시스템을 가로채고 있기 때문에 감염 여부를 쉽게 알 수 없었다.

ZeroAccess나 Zaccess, 혹은 Max++로 불리기도 하는 Smiscer 악성코드는 2010년 1월경부터 제작/배포된 것으로 보이지만, 이 시기에 국내에서 보고된 별다른 피해 사례는 없었다. 하지만 감염 기법과 은닉 상태로 감안한다면, 다만 그 존재를 감지하지 못한 경우도 있었을 것으로 추정된다. 이후 Smiscer.C 변형에 대한 감염 사례는 국내에서 보고된 바 있다.

참고:

- http://asec.ahnlab.com/328
- http://download.ahnlab.com/asecReport/ASEC_Report_Vol.16_Kor.pdf (MBR Infector)
- http://download.ahnlab.com/kr/site/magazineAhn/ahn_201110.pdf (TDL4 Bootkit)
- http://download.ahnlab.com/asecReport/ASEC_Report_Vol.14_Kor.pdf (Smiscer Rootkit)

이번에는 자체 보호 기능이 있는 Smiscer.C 악성코드에 대해 살펴보자.

A. 유포지 : 'http://ya**********/install_flash_player.exe'

유포 파일명을 보면, 플래시 플레이어 설치 파일처럼 사용자를 속이고 있는 것을 볼 수 있다.

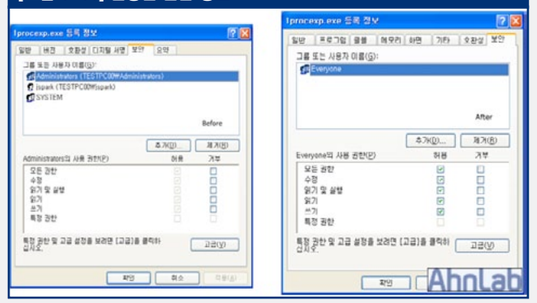
B. 감염 시 다음과 같은 파일이 생성된다.

C:₩WINDOWS₩3842759408:254145562.exe(ADS 동작)
C:₩WINDOWS₩assembly₩gac_msil₩desktop.ini
C:₩WINDOWS₩system32₩drivers₩***.sys(시스템 정상 파일을 패치)
C:₩WINDOWS₩$NtUninstallKB1216$(시스템별 다름)
C:₩documents and settings₩[사용자계정]₩local settings₩application data₩[시스템별 다름]₩X

참고: ADS(Alternate Data Stream) http://core.ahnlab.com/7

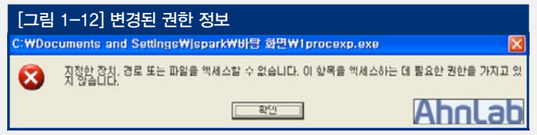
C. 감염 후 3842759408:254145562.exe의 특정 영역에 접근하면, 접근한 프로세스를 강제로 종료하고 권한을 변경한다.

[그림 1-11] 변경된 권한 정보



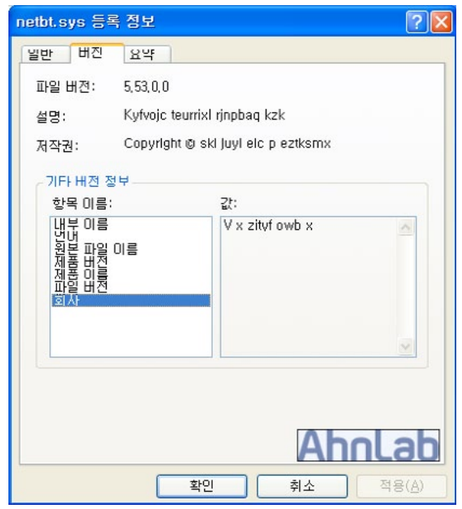
추후 실행이 불가능하며, [그림 1-12]와 같은 오류 메시지가 뜬다.

[그림 1-12] 변경된 권한 정보



패치된 시스템 파일 정보가 변경된 것을 볼 수 있다.

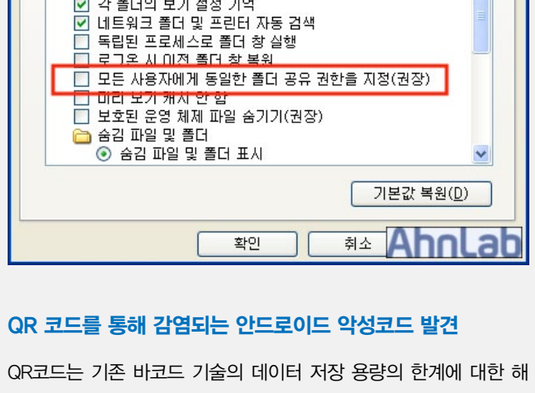
[그림 1-13] 변경된 버전 정보



D. 조치 방법

1. 안철수연구소 홈페이지에서 제공하는 전용 백신으로 검사 및 치료한다.
2. 전용 백신 다운로드 http://www.ahnlab.com/kr/site/download/vacc/vaccView.do?seq=103
3. 치료 후 재부팅해야 한다.
4. 재부팅 후에는 V3 정밀 검사를 실시한다.

[그림 1-14] 전용 백신 치료 후 재부팅



E. 권한 복원 방법

재부팅 후에 변경된 권한을 복원할 때, '보안' 탭이 보이지 않으면 [그림 1-15]와 같이 폴더 옵션을 수정하여 권한을 부여한다.

[그림 1-15] 폴더 옵션 변경

QR 코드를 통해 감염되는 안드로이드 악성코드 발견

QR코드는 기존 바코드 기술의 데이터 저장 용량의 한계에 대한 해결책으로 새롭게 각광받고 있다. QR 코드는 스마트폰의 카메라(스캔) 기능을 활용하여 그림과 같은 물리적 자료에서 디지털 정보, 즉 인터넷 주소 등을 변환해준다. 특히 이벤트나 광고에서 많이 사용되고 있다.

참고: http://ko.wikipedia.org/wiki/QR_코드

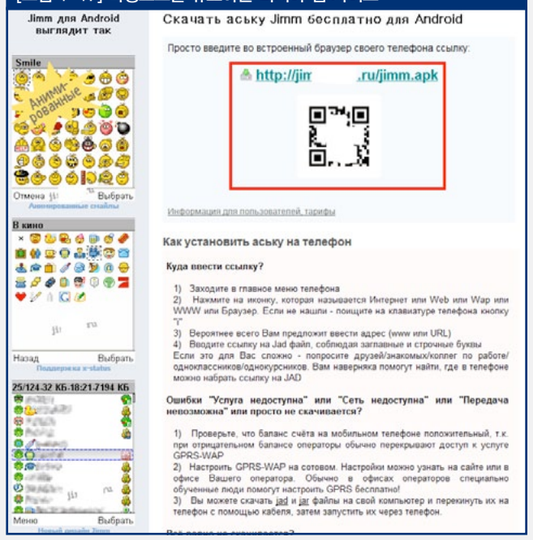
이번에도 가능한 한 많은 PC가 악성코드에 감염되도록 다양한 취약점을 사용하였으며 전체적인 구조를 요약해 보면 아래와 같다.

해당 악성코드에 감염되면 윈도우 정상 파일(imm32.dll)을 교체하는 증상이 발생한다. 악성 imm32.dll이 실행되면서 백업 된 정상 imm32.dll인 Shell64.dll을 로딩하도록 해 두었는데 이는 악성 imm32.dll의 코드를 통해서 확인할 수 있다.

[그림 1-16] QR 코드가 광고에 사용된 예

최근 러시아의 특정 웹 사이트에서 QR 코드가 스마트폰 악성코드를 유포하는 데에 이용되었다. 이 웹 페이지에서 애플리케이션 홍보를 위해 QR 코드와 주소 링크를 게시하여 사용자로 하여금 해당 애플리케이션을 설치하도록 유도한 것이다. 해당 웹 페이지 화면 중앙 상단에는 안드로이드 애플리케이션을 바로 내려받을 수 있는 URL과 QR 코드를 스캔할 수 있게 되어 있고, 좌측에는 애플리케이션 실행 화면이 나와 있다.

[그림 1-17] 악성코드를 유포하는 러시아 웹 사이트

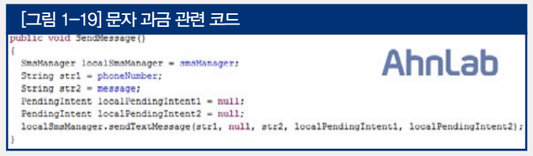


해당 악성코드는 [그림 1-18]과 같은 권한을 사용하는데, SEND_SMS 권한은 문자를 송신할 때 반드시 요구되는 권한으로 악성 애플리케이션을 구분하는 중요한 단서가 된다. 만일 설치하려는 애플리케이션이 문자를 송신할 필요가 전혀 없는 애플리케이션인데 SEND_SMS 권한을 사용하고 있다면, 한번쯤 악성 애플리케이션인지 의심해 볼 만하다.

[그림 1-18] 악성 애플리케이션 정보

[그림 1-19]와 같은 코드를 이용하여 '2476'의 premium rate number로 문자를 송신하는 기능이 있다. (러시아에선 약 6달러가 결제된다고 알려졌다).

[그림 1-19] 문자 과금 관련 코드



이 악성코드는 V3 모바일 제품군에서 다음과 같이 진단한다.

– Android-Trojan/SmsSend.K

참고로 이 악성코드는 Disassemble 시 코드를 분석하기 어렵게 약간 난독화되어 있다. 모바일 악성코드도 항상 최신 동향을 반영하여 좀더 많은 기기를 감염시키기 위해 발전된 기법을 연구하는 등, 사용자들을 유혹하고 있다.

다음과 같은 수칙을 반드시 지켜 악성코드로부터 스마트폰을 안전하게 보호할 것을 권장한다.

1. 애플리케이션을 설치하거나 이상한 파일을 다운로드한 경우에는 반드시 악성코드 검사를 실시한다.
2. 게임 같은 애플리케이션을 다운로드할 때는 먼저 다른 사용자가 올린 평판 정보를 면밀히 확인한다.
3. 브라우저나 애플리케이션으로 인터넷에 연결 시 이메일이나 문자 메시지에 있는 URL은 신중하게 클릭한다.
4. PC로부터 파일을 전송받을 경우 악성코드 여부를 꼭 확인한다.
5. 백신의 패치 여부를 확인해서 최신 백신 엔진을 유지한다.
6. 스마트폰의 잠금 기능(암호 설정)을 이용해서 다른 사용자의 접근을 막는다. 잠금 기능에 사용한 비밀번호는 수시로 변경한다.
7. 블루투스 기능 같은 무선 기능은 필요할 때만 켜놓는다.
8. ID, 비밀번호 등을 스마트폰에 저장하지 않는다.
9. 주기적으로 백업해서 분실 시 정보의 공백이 생기지 않도록 한다.
10. 임의로 개조하지 않고 복사 방지 같은 기능을 해제하지 않는다.

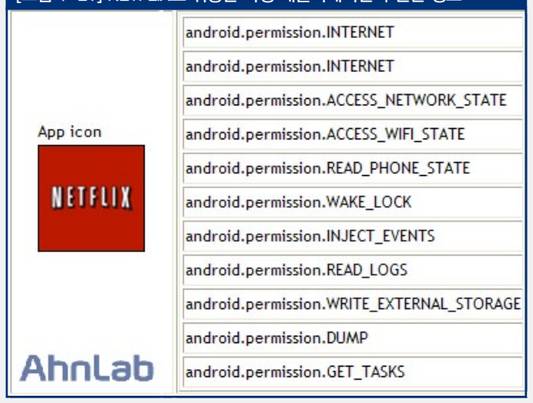
NETFLIX 위장 안드로이드 악성 애플리케이션

Google+를 위장한 악성 애플리케이션 Google++(8월 발견), Google Search를 위장한 Google SSearch(6월 발견)에 이어, 이번에는 미국에서 인기 있는 온라인 스트리밍 비디오 서비스 애플리케이션으로 위장한 악성 애플리케이션 Netflix가 발견되었다.

[그림 1-20] Google 위장 안드로이드 악성 애플리케이션

NETFLIX로 위장한 악성 애플리케이션의 특징을 살펴보자.

[그림 1-21] NETFLIX로 위장한 악성 애플리케이션의 권한 정보



1. 안드로이드 OS 1.6 이상에서 설치 가능하도록 제작되었다.

[그림 1-22] Manifest 정보

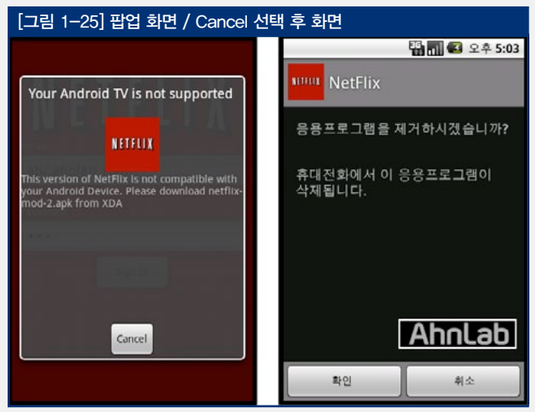
[그림 1-23] 악성 애플리케이션의 응용 프로그램 정보

2. 악성 애플리케이션을 실행하면 [그림 1-24]와 같이 E-mail과 Password를 입력하라는 메시지가 나온다.

[그림 1-24] 악성 애플리케이션 실행 화면

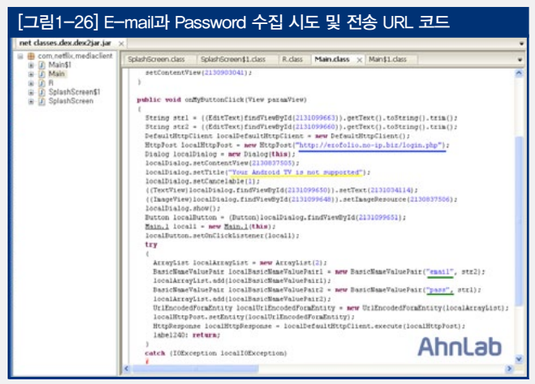
3. E-mail과 Password를 입력하면 [그림 1-25]와 같은 화면이 팝업되고, Cancel을 선택하면 제거 화면으로 넘어간다. 단, cancel 선택은 취소할 수 없고, 반드시 삭제하게 되어 있다.

[그림 1-25] 팝업 화면 / Cancel 선택 후 화면



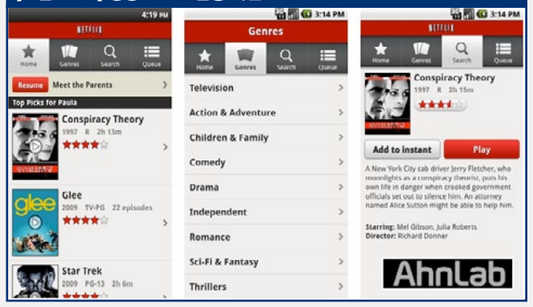
4. 수집된 E-mail과 Password를 외부 서버로 전송한다.

[그림1-26] E-mail과 Password 수집 시도 및 전송 URL 코드



5. 정상 Netflix 실행 시 [그림 1-27]과 같이 스마트폰 상단의 상태바가 표시되므로 악성과 구분할 수 있다.

[그림 1-27] 정상 Netflix 실행 화면

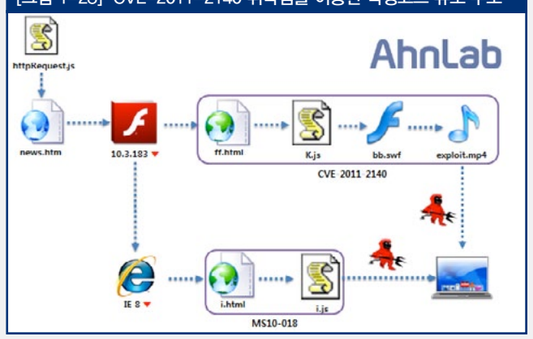


이처럼 인기 있는 애플리케이션으로 위장한 경우가 지속적으로 발견되고 있으므로, 애플리케이션을 설치할 때 불필요한 권한을 요구하지는 않는지 스마트폰 안전 수칙을 항상 염두에 둬야 한다.

CVE 2011-2140 취약점을 이용한 악성코드 유포

10월에 CVE 2011-2140 취약점을 이용한 악성코드 유포 사례가 다수 발견되었다. 당시 악성코드의 유포 방식을 살펴보면 [그림 1-28]과 같다.

[그림 1-28] CVE-2011-2140 취약점을 이용한 악성코드 유포 구조



[그림 1-28]을 보면 CVE-2011-2140 취약점을 이용하는 악성코드는 총 4개의 파일로 구성되어 있다. IE 버전에 상관없이 플래시 플레이어 버전이 10.3.183보다 낮으면 해당 취약점이 동작하여 온라인 게임 사용자의 계정 정보를 탈취하는 악성코드에 감염된다.

참고: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2140(CVE-2011-2140)

악성코드 제작자들은 금전적인 이득을 취하기 위해 여러 PC에 악성코드를 퍼뜨리려 하고 이 과정에서 가능한 한 많은 취약점을 사용하려고 한다. 따라서 자신의 PC를 악성코드 감염으로부터 보호하기 위해서는 PC에 설치된 프로그램은 무엇인지, 해당 프로그램들의 보안 업데이트는 잘하고 있는지 살펴보는 것이 중요하다.

플래시가 당신 컴퓨터의 웹 카메라와 마이크를 조종한다

스탠퍼드대학교에 재학중인 한 학생이 어도비 플래시를 통하여 컴퓨터의 웹 카메라와 마이크를 원격에서도 조종할 수 있다는 사실을 증명해 보였다.

[그림 1-29] 웹 카메라

이 취약점을 발견한 Feross Aboukhadijeh는 어도비사에 관련 취약점을 알렸지만, 몇 주 동안 아무런 답변도 전달받지 못하자 이 위험성에 대해서 공유하기 위해 블로그에 게시하였다. 그 후, 여러 온라인 매체에 이 내용이 소개되었고 어도비는 블로그에 게시된지 이틀만에 취약점 문제를 해결하였다. 이 기법은 오래 전부터 알려진 클릭재킹이라는 기술이다. 예전에도 이러한 방법으로 카메라의 기능을 조정할 수 있었고 어도비는 framebusting이라는 코드를 삽입하여 동작하지 못하도록 하였다. 하지만 Feross는 이 클릭재킹이 다른 방법을 통해서 여전히 동작한다는 것을 발견했다.

[그림 1-30] 클릭재킹을 보여준 데모 화면

Feross는 [그림 1-30]과 같이 컴퓨터에 장착된 웹 카메라를 작동하여 아직도 취약점이 유효하다는 사실을 데모로 보여주었다. 그는 문제가 되는 플래시 플레이어 세팅 매니저를, iframe을 이용하여 전체 페이지를 불러오는 것이 아니라 SWF 파일만을 로드하도록 하여 기존의 framebusting 자바스크립트 코드를 우회하도록 하였다.

이번 취약점은 사용자의 플래시 업데이트 없이 이미 문제가 해결되었는데, 세팅 매니저가 어도비의 시스템에서 관리되기 때문에 빠르게 반영할 수 있었다. 만약 Feross가 이 사실을 공개하지 않았다면 누군가 이 취약점을 악용할 수도 있었을 것이다.

이와 같은 이유로, 취약점은 공개되어야 한다는 쪽과 그렇지 않다는 쪽으로 의견이 나뉘고 있다. 취약점이 공개되면 오히려 악용될 소지가 있다는 점은 분명해 보인다. 하지만 알려지지 않은 취약점이 방치된 소프트웨어를 과연 사용해야 할 것인가? 아마 이에 대한 대답은 여전히 사용자의 몫일 것이다.

리눅스 Tsunami DDoS 공격 툴의 맥 OS X 포팅

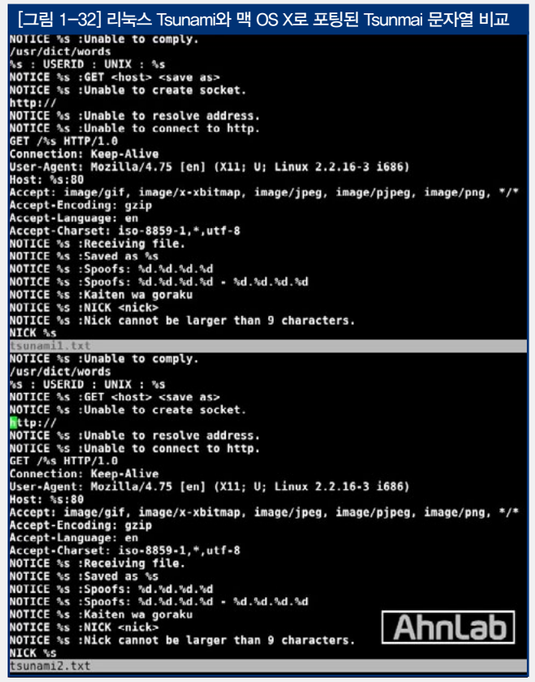
오래 전에 리눅스에서 이용되었던 백도어 프로그램이 최근 맥OS X로 포팅되었다는 소식이 전해졌다. 해당 공격 툴은 Tsunami이며, 2002년 당시 진단명은 Tsunami 혹은 Kaiten이었다. Tsunami는 IRC 기반의 분산 서비스 공격 클라이언트 프로그램으로 볼 수 있으며, 지정된 IRC 서버에 접속해서 공격 명령을 전달받는 구조로 다음과 같은 기능이 있다.

- PUSH+ACK Flooding
- SYN Flooding
- UDP Flooding
- 웹으로부터 파일 다운로드
- 명령어 실행

[그림 1-31] 맥 OS X로 포팅된 Tsunami의 오리지널 소스코드 화면

이번에 발견된 맥 OS X로 포팅된 코드는 2000년 초반에 알려진 Kaiten 코드를 포팅한 것이다. 차이점은 맥에서 돌아간다는 것과 IRC 접속 서버 주소, 채널, 패스워드 정도다. [그림 1-32]는 리눅스에서 발견된 것과 최근 맥 OS X로 포팅된 바이너리에서 추출한 문자열을 비교한 것이다.

[그림 1-32] 리눅스 Tsunami와 맥 OS X로 포팅된 Tsunmai 문자열 비교



문자열이 같은 것을 확인할 수 있으며, 문자열의 User-Agent 필드를 보면 리눅스 2.2.16-3 커널 정보가 보인다. 2.2.16이 나왔던 시점이 2000년이었음을 고려하면 해당 코드가 작성된 시점하고도 비슷하다. 이처럼 공개된 소스코드가 있다면, 이러한 형태의 백도어가 맥 OS X뿐만 아니라 다양한 운영체제를 대상으로 포팅될 수 있다. 물론, 그만큼의 사용자층이 존재하는 운영체제라면 말이다.

01. 악성코드 동향
c. 악성코드 분석 특집

국내PC를 감염 목표로 하는 부트킷 상세 분석

추석을 기점으로 MBR을 감염시키는 악성코드가 등장하였다. 기존에도 많은 부트킷(Bootkit)이 나타났지만, 국내를 대상으로 한 첫 시도가 아닐까 생각된다. 또한 국내에서 사용하는 백신 프로그램을 무력화하는 코드도 존재한다. MBR을 감염시키는 방식은 DOS 시절부터 자주 사용되었으며, 윈도우에서 Protect 모드가 실행되기 전 ntoskrnl 등을 메모리 패치하여 부팅 시점에서부터 감염이 이루어지게 한다. 해당 악성코드의 주요 특징은 다음과 같다.

- MBR을 감염시켜 악성 루트킷(Rootkit)을 부팅 시점에 로드
- API 후킹을 통해 국산 백신 및 백업 프로그램 무력화
- 정상 윈도우 패치를 통한 자기 보호 기능

해당 악성코드는ASD(Ahnlab Smart Defense)에 의해 9월 13일 최초 탐지되었다. 4320여 대가 감염된 것으로 파악되었으며, 모두 국내로 추정된다. 해당 악성코드는 'http://www.hasshudo.jp/img/ta.jpg'를 통해 유포되는 것으로 분석 되었다. 어도비 플래시(Adobe Flash)의 취약점을 이용하며 드라이버를 이용한 자체 보호 기능, MBR을 감염시키는 부트킷 기능이 있다.

1. 감염 동작의 특징

가. 동작 순서

A. MBR이 감염되면 Ntldr을 패치하고, Ntoskrnl이 로드되면 Call IoitSystem() Instruction을 찾아 Call 패치를 하여 악성 드라이버를 로드 한다.

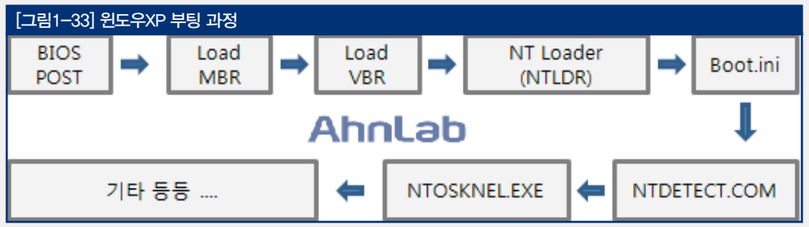
B. 악성 드라이버는 Atapi.sys의 Major Function INTERNAL_DEVICE_CONTROL을 후킹 한다.

C. NtCreateProcessEx() SSDT를 후킹하여, 자기 보호 및 AV 프로그램을 무력화한다.

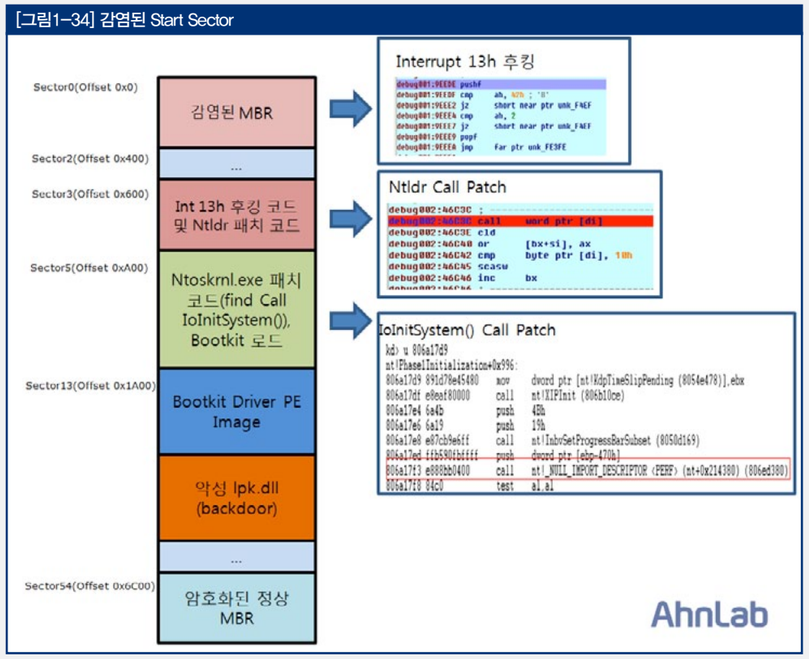
나. MBR의 변화

정상적인 PC 시스템의 부팅 과정은 [그림1-33]과 같다. 악성코드에 감염되면, MBR 이 변조되어 [그림 1-34]와 같이 Start Sector부터 감염 코드 및 악성 루트킷, DLL이 저장된다. 악성코드는 EXE 형태의 드롭퍼로, 루트킷을 드롭하고 서로 통신하면서 MBR 감염 및 자기 보호 기능을 수행한다.

[그림1-33] 윈도우XP 부팅 과정



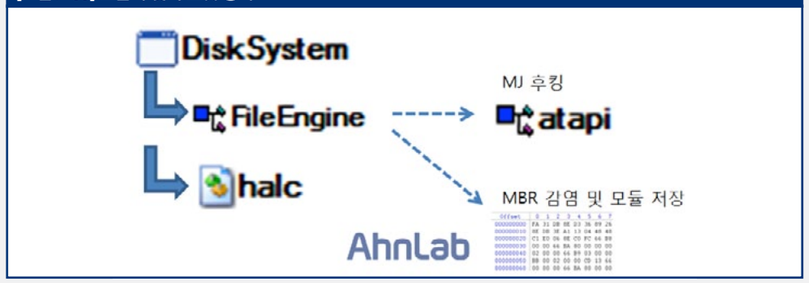
[그림1-34] 감염된 Start Sector



2. 드롭퍼 및 루트킷 상세 분석

악성코드는 웹을 통해 드롭퍼를 내려받아 실행한다. 해당 드롭퍼는 UPX로 실행 압축되어 있으며, FileEngine.sys 라는 악성 루트킷을 드롭하고 서비스에 등록하여 해당 드라이버를 실행한다. [그림 1-35]는 해당 드롭퍼에 대한 간단한 설명이다.

[그림1-35] 드롭퍼 및 루트킷 동작



1. 드롭퍼와 루트킷은 서로 통신을 하므로 루트킷은 Symbolic Link와 같은 초기화 작업을 수행한다. 이후 nt!IoCreateFile()을 [그림 1-36]과 같이 메모리 패치 하여 후킹한다.

[그림1-36] nt!IoCreateFile 후킹을 위해 메모리 패치 된 부분

```
kd> u 80572bf3
nt!IoCreateFile:
80572bf3 e9595a1176      jmp     FileEngine+0x651 (f6688651)
80572bf8 83ec0c          sub     esp,0Ch
80572bfb 53              push    ebx
80572bfc 56              push    esi
80572bfd 33f6            xor     esi,esi
80572bff 8975fc          mov     dword ptr [ebp-4],esi
80572c02 8b1d38805580    mov     ebx,dword ptr [nt!ExHotpSyncRenameSequence (80558038)]
80572c08 f6c301          test    bl,1
```

AhnLab

2. 후킹에 성공했을 때, 특정 보안제품의 nsvmXX.npc의 프로세스가 존재 할 경우 STATUS_OBJECT_NAME_NOT_FOUND(0xC0000034)를 리턴하여 해당 제품을 무력화한다. 또한 루트킷의 경우 XXXEngine.sys, XXXEn~1.sys, XXXSystem.exe, XXXSY~1.EXE, halc.dll라는 이름으로 IoCreateFile()을 실행하면, STATUS_UNSUCCESSFUL(0xC0000001)을 리턴하여 액세스를 방해한다. 다시 정리하면, 아래와 같은 문자열로 CreateFile()을 수행할 경우 에러를 리턴하여 실행을 방해한다.

– XXXEngine.sys , XXXEn~1.sys XXXSystem.exe, XXXSY~1.EXE, halc.dll

[그림1-37]은 위의 문자열을 비교하여 에러를 리턴하는 코드다.

[그림1-37] nt!IoCreateFile이 후킹되었을 때 실행되는 코드

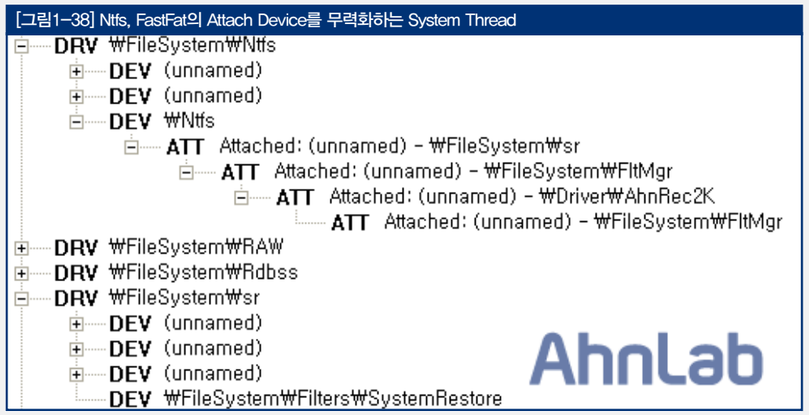
```
v23 = (char *)GetCurrentProcess() + 372;
if ( stricmp(v23, "nvmon.npc") )
{
  if ( !VirtualAddress
    || !MmIsAddressValid(VirtualAddress)
    || !*((_DWORD *)VirtualAddress + 2)
    || !MmIsAddressValid(*((PVOID *)VirtualAddress + 2))
    || (v15 = *((_DWORD *)VirtualAddress + 2), !*(_DWORD *)(v15 + 4))
    || !*(_WORD *)v15
    || !MmIsAddressValid(*(PVOID *)(v15 + 4)) )
    goto LABEL_24;
  memset(&v21, 0, 0x400u);
  v16 = *((_DWORD *)VirtualAddress + 2);
  v17 = *(_WORD *)v16;
  v18 = *(const void **)(v16 + 4);
  LOBYTE(v16) = *(_WORD *)v16;
  v17 >>= 2;
  memcpy(&v21, v18, 4 * v17);
  memcpy(&v21 + 2 * v17, (char *)v18 + 4 * v17, v16 & 3);
  v19 = wcslen(&v21);
  for ( i = v19 - 1; i > 0 && *(&v21 + i) != 92; --i )
    ;
  if ( (signed int)v19 > 1
    && i != v19 - 1
    && (sub_1057C(v23) && sub_105D6((const wchar_t *)&v22[i]) || sub_10300((const wchar_t *)&v22[i])
                                                              && !sub_1043E(v23)) )
    result = 0xC0000001u;
  else
LABEL_24:
    result = sub_10630();
}
else
{
  result = 0xC0000034u;
}
return result;
```

AhnLab

다른 파일로 접근하기 위해서는 CreateFile()을 이용하여 파일의 핸들을 얻어야 하므로, 후킹된 코드는 위에 설명한 파일들의 핸들을 얻는 것을 방해한다.

3. 이후 Atapi.sys 의 DeviceType 중 FILE_DEVICE_DISK Type의 DeviceOb/ect를 찾고 Disk.sys를 확인한다. DriverEntry의 행위는 위에서 설명한 것과 같으며, DeviceIoControl()를 이용하여 드롭퍼와 통신하면서 MBR 감염 작업을 진행한다. 또한 ₩₩FileSystem₩₩Ntfs , ₩₩FileSystem₩₩Fastfat의 Attach Device를 모두 0으로 변경하는 1개의 System Thread를 실행한다. 원래는 ₩₩FileSystem₩₩Sr 이 Attach Device로 존재하는데, 해당 Device는 FltMgr device와 관련이 있으며 AV 프로그램 및 SystemRestore를 무력화 한다.

[그림1-38] Ntfs, FastFat의 Attach Device를 무력화하는 System Thread



해당 코드는 [그림 1-39]와 같다.

[그림1-39] System Thread 코드

```
char __stdcall sub_1A94F(PCUSTR lpObject)
{
  char result; // al@2
  int i; // eax@3
  UNICODE_STRING DestinationString; // [sp+4h] [bp-8h]@1

  RtlInitUnicodeString(&DestinationString, lpObject);
  if ( ObReferenceObjectByName(&DestinationString, 64, 0, 0, IoDriverObjectType, 0, 0, &Object) >= 0 )
  {
    for ( i = *((_DWORD *)Object + 1); i; i = *(_DWORD *)(i + 0xC) )
      *(_DWORD *)(i + 0x18) = 0;                // Attach Device = 0x0
    ObfDereferenceObject((PVOID)Object);
    result = 1;
  }
  else
  {
    result = 0;
  }
  return result;
}
```

AhnLab

코드에 대해 간단히 설명하면, ObReferenceObjectByName()을 이용하여 Driver 이름으로 Driver Object를 얻을 수 있다. 그 후 Driver Object에서 Device Object를 얻어와 Attach Device를 찾는다. Attach Device의 값을 0x0으로 변경하면 해당 드라이버의 Attach Device는 없어진다. 이러한 동작은AV프로그램과 System Restore를 무력화하는데 이용된다.

4. 드롭퍼와 DeviceIoControl()을 이용하여 통신하며 MBR 감염을 시작한다.
먼저 정상 MBR을 0x9C Key로 XOR 1byte 암호화한 뒤 백업하는 작업을 수행하며, 암호화를 진행하는 코드는 [그림 1-40]과 같다.

[그림1-40] 0x9C 1byte XOR 암호화 루틴

```
00401753  8DB3 BE010000   LEA   ESI, DWORD PTR DS:[EBX+1BE]
00401759  B9 10000000     MOV   ECX, 10
0040175E  BF EE714000     MOV   EDI, DiskSyst.004071EE
00401763  33C0            XOR   EAX, EAX
00401765  F3:A5           REP   MOVS DWORD PTR ES:[EDI], DWORD PTR DS:[ESI]
00401767  8A0C18          MOV   CL, BYTE PTR DS:[EAX+EBX]
0040176A  80F1 9C         XOR   CL, 9C
0040176D  880C18          MOV   BYTE PTR DS:[EAX+EBX], CL
00401770  40              INC   EAX
00401771  3D 00020000     CMP   EAX, 200
```

A. 드롭퍼는 설치된 루트킷과 통신하며, 자신의 코드를 루트킷을 이용하여 Start Sector에 저장한다. Start Sector에 저장된 모듈은 감염된 MBR에 의해 로드되는 것이다.

[그림1-41] Start Sector에 쓰여진 데이터

```
003F0000 02 00 00 00|01 00 00 00|9C 60 E8 00|00 00 00 58
003F0010 2D 07 00 00|00 89 C5 FC|8B 7C 24 24|81 E7 00 00
003F0020 F0 FF B0 C7|AE 75 FD BB|46 34 00 40|39 1F 75 F4
003F0030 B0 A1 AE 75|FD 8B 37 8B|36 89 F3 8B|36 39 DE 75
003F0040 16 61 9D 89|C6 85 C0 75|0D 9C 50 B8|21 00 00 00
003F0050 01 44 24 08|58 9D C3 8B|5E 18 8B 46|20 2D 04 00
003F0060 00 00 31 C9|BE 6A 4B 6A|19 39 C1 77|D4 39 34 0B
003F0070 74 06 41 E9|F1 FF FF FF|01 D9 8B 41|04 25 FF 00
003F0080 00 00 3D E8|00 00 00 75|B8 B0 E8 8D|79 09 B9 0F
003F0090 00 00 00 AE|75 FD 8B 47|04 25 FF FF|00 00 3D 84
003F00A0 C0 00 00 75|9C 57 8B 73|3C 8B 74 33|50 56 89 E8
003F00B0 05 00 02 00|00 01 DE 81|EE 00 04 00|00 89 F7 89
003F00C0 C6 B9 00 01|00 00 F3 A5|5E 5F 8B 07|01 F
003F00D0 00 00 00 01|DE 89 9E 0D|FC FF FF 81|EE F
003F00E0 00 29 F0 2D|05 00 00 00|89 46 01 81|EE 0
003F00F0 00 29 FE 81|EE 04 00 00|00 89 37 E9|41 FF FF FF
```

드롭퍼는 루트킷과 통신하기 위해 DeviceIoControl() API를 사용하는데, 이때 인자 중 IOCTL_CODE 0x222014 는 악성코드가 필요한 데이터를 Start Sector에 쓰는 기능을 한다. [그림 1-41]을 보면, 첫 번째 4byte는 Sector의 위치, 두 번째 4byte는 쓰여질 크기이며 나머지는 코드다.

B. IOCTL_CODE 0x222020은 Atapi.sys의 Major Function INTERNAL_DEVICE_CONTROL을 후킹하여 Start Sector에 대한 Write 기능을 무력화한다.

[그림1-42] 후킹 된 Atapi.sys INTERNEL_DEVICE_CONTROL Major Function

```
kd> !drvobj \driver\atapi 2
Driver object (81f87510) is for:
 \Driver\atapi
DriverEntry:    f84a85f7
DriverStartIo:  f849a7c6
DriverUnload:   f84a4204
AddDevice:      f84a2300

Dispatch routines:
[00] IRP_MJ_CREATE                      f849d572    +0xf849d572
[01] IRP_MJ_CREATE_NAMED_PIPE           805051be    nt!IopInvalidDeviceRequest
[02] IRP_MJ_CLOSE                       f849d572    +0xf849d572
[03] IRP_MJ_READ                        805051be    nt!IopInvalidDeviceRequest
[04] IRP_MJ_WRITE                       805051be    nt!IopInvalidDeviceRequest
[05] IRP_MJ_QUERY_INFORMATION           805051be    nt!IopInvalidDeviceRequest
[06] IRP_MJ_SET_INFORMATION             805051be    nt!IopInvalidDeviceRequest
[07] IRP_MJ_QUERY_EA                    805051be    nt!IopInvalidDeviceRequest
[08] IRP_MJ_SET_EA                      805051be    nt!IopInvalidDeviceRequest
[09] IRP_MJ_FLUSH_BUFFERS               805051be    nt!IopInvalidDeviceRequest
[0a] IRP_MJ_QUERY_VOLUME_INFORMATION    805051be    nt!IopInvalidDeviceRequest
[0b] IRP_MJ_SET_VOLUME_INFORMATION      805051be    nt!IopInvalidDeviceRequest
[0c] IRP_MJ_DIRECTORY_CONTROL           805051be    nt!IopInvalidDeviceRequest
[0d] IRP_MJ_FILE_SYSTEM_CONTROL         805051be    nt!IopInvalidDeviceRequest
[0e] IRP_MJ_DEVICE_CONTROL              f849d592    +0xf849d592
[0f] IRP_MJ_INTERNAL_DEVICE_CONTROL     f6689a30    +0xf6689a30
[10] IRP_MJ_SHUTDOWN                    805051be    nt!IopInvalidDeviceRequest
[11] IRP_MJ_LOCK_CONTROL                805051be    nt!IopInvalidDeviceRequest
[12] IRP_MJ_CLEANUP                     805051be    nt!IopInvalidDeviceRequest
[13] IRP_MJ_CREATE_MAILSLOT             805051be    nt!IopInvalidDeviceRequest
[14] IRP_MJ_QUERY_SECURITY              805051be    nt!IopInvalidDeviceRequest
[15] IRP_MJ_SET_SECURITY                805051be    nt!IopInvalidDeviceRequest
[16] IRP_MJ_POWER                       f849d5bc    +0xf849d5bc
[17] IRP_MJ_SYSTEM_CONTROL              f84a4164    +0xf84a4164
[18] IRP_MJ_DEVICE_CHANGE               805051be    nt!IopInvalidDeviceRequest
[19] IRP_MJ_QUERY_QUOTA                 805051be    nt!IopInvalidDeviceRequest
[1a] IRP_MJ_SET_QUOTA                   805051be    nt!IopInvalidDeviceRequest
[1b] IRP_MJ_PNP                         f84a4130    +0xf84a4130
```


C. 패치 된 Atapi.sys Major Function은 아래와 같이 STATUS_ACCESS_DENIED (0xC0000022) 코드를 리턴하여 감염 이후의 MBR 변조를 막는다.

[그림1-43] 패치 된 코드

```
if ( Transfer_length && LBA <= 0_12DA8 + (unsigned int) 3b__12DB0 && LBA + Transfer_length >= 0_12DA8 )
{
  Irp->IoStatus.Information = 0;
  Irp->IoStatus.Status = 0xC0000022u;
  IofCompleteRequest(Irp, 0);
  return 0xC0000022u;
}
return OrgAtapi_12DA4(a1, Irp);
```

E. SCSI_REQUEST_BLOCK 구조체의 CDB 구조체를 검사하여 MBR에 대한 접근을 STATUS_ACCESS_DENIED (0xC0000022) 리턴하여 방해한다.

[그림1-44] Write Cdb 구조체

bit→ ↓byte	7	6	5	4	3	2	1	0
0	Operation code = 2Ah							
1	WRPROTECT			DPO	FUA	Reserved	FUA_NV	Obsolete
2-5	LBA							
6	Reserved			Group Number				
7-8	Transfer length							
9	Control							

Operation code와 LBA를 확인한 후 Sector 부분을 직접 수정하는 것을 방해한다.

3. 감염된 MBR 및 부트킷 분석

가. 감염된 MBR

감염된 MBR은 [그림 1-34]와 같이 Interrupt 13h을 후킹하여 저장된 코드를 읽어오고 실행한다. [그림 1-45]는 감염된 MBR 코드다.

[그림1-45] 감염된 MBR 코드

```
BOOT_SECTOR:7C00 public start
BOOT_SECTOR:7C00 start:
BOOT_SECTOR:7C00 cli
BOOT_SECTOR:7C01 xor     bx, bx
BOOT_SECTOR:7C03 mov     ss, bx
BOOT_SECTOR:7C05 assume ss:BOOT_SECTOR
BOOT_SECTOR:7C05 mov     ss:word_7BFE, sp
BOOT_SECTOR:7C09 mov     sp, 7BFEh
BOOT_SECTOR:7C0D push    ds
BOOT_SECTOR:7C0E pushad
BOOT_SECTOR:7C10 mov     ds, bx
BOOT_SECTOR:7C12 assume ds:BOOT_SECTOR
BOOT_SECTOR:7C12 db      3Eh
BOOT_SECTOR:7C13 mov     ax, ds:413h
BOOT_SECTOR:7C16 dec     ax
BOOT_SECTOR:7C17 dec     ax
BOOT_SECTOR:7C18 db      3Eh
BOOT_SECTOR:7C18 mov     ds:413h, ax
BOOT_SECTOR:7C1C db      3Eh
BOOT_SECTOR:7C1C mov     ax, ds:413h
BOOT_SECTOR:7C20 shl     ax, 6
BOOT_SECTOR:7C23 mov     es, ax
BOOT_SECTOR:7C25 cld
BOOT_SECTOR:7C26 mov     eax, 201h
BOOT_SECTOR:7C2C mov     ecx, 1
BOOT_SECTOR:7C32 mov     edx, 80h ; '€'
BOOT_SECTOR:7C38 xor     ebx, ebx
BOOT_SECTOR:7C3A int     13h                    ; DISK - READ SECTORS INTO MEMORY
BOOT_SECTOR:7C3A                                ; AL = number of sectors to read, CH = track, CL = sector
BOOT_SECTOR:7C3A                                ; DH = head, DL = drive, ES:BX -> buffer to fill
BOOT_SECTOR:7C3A                                ; Return: CF set on error, AH = status, AL = number of sectors read
BOOT_SECTOR:7C3D mov     eax, 201h
BOOT_SECTOR:7C43 mov     ecx, 3
BOOT_SECTOR:7C49 mov     edx, 80h ; '€'
BOOT_SECTOR:7C4F mov     ebx, 200h
BOOT_SECTOR:7C55 int     13h                    ; DISK - READ SECTORS INTO MEMORY
BOOT_SECTOR:7C55                                ; AL = number of sectors to read, CH = track, CL = sector
BOOT_SECTOR:7C55                                ; DH = head, DL = drive, ES:BX -> buffer to fill
BOOT_SECTOR:7C55                                ; Return: CF set on error, AH = status, AL = number of sectors read
BOOT_SECTOR:7C57 mov     eax, 201h
BOOT_SECTOR:7C5D mov     ecx, 4
BOOT_SECTOR:7C63 mov     edx, 80h ; '€'
BOOT_SECTOR:7C69 mov     ebx, 400h
BOOT_SECTOR:7C6F int     13h                    ; DISK - READ SECTORS INTO MEMORY
BOOT_SECTOR:7C6F                                ; AL = number of sectors to read, CH = track, CL = sector
BOOT_SECTOR:7C6F                                ; DH = head, DL = drive, ES:BX -> buffer to fill
```

1. Int 13h를 후킹 후, 원본 MBR 코드를 복호화하여 MBR 영역에 덮어 쓴 후 실행한다. 후킹된 Int 13h 코드는 Ah가 0x42, 0x2일 때 후킹된 코드로 넘어가는데, 0x2는 Read Setors From Drive이며 0x42는 Extended Read Sectors From Drive로 모두 Read와 관련 있다.

2. 이후 Ntldr(osloader.exe) 에서 0x15FF를 찾아 Call을 패치 한다. Ntldr(osloader.exe) 에 의해 패치 된 코드는 ntoskrnl.exe 로드 시점에서 Call InitSystem() 명령어를 찾아 해당 Call을 다시 한번 패치 한다.

[그림1-46] Ntldr(osloader.exe) Call 패치

```
debug002:46C3C ; ---------------------------
debug002:46C3C call    word ptr [di]
debug002:46C3E cld
debug002:46C40 or      [bx+si], ax
debug002:46C42 cmp     byte ptr [di], 10h
debug002:46C45 scasw
debug002:46C46 inc     bx
```

3. 악성코드는 ntoskrnl.exe 마지막 섹션에서 실행할 코드를 저장한다. 해당 코드는 Start Sector에 저장된 드라이버를 읽어와 메모리 할당 후 실행한다. TDL4와 마찬가지 방법으로 64bit 코드로 컴파일되었다면 Sign 체크를 우회할 수 있을 것이다. ntoskrnl.exe 마지막 섹션에서 실행할 코드는Nt!IoltSystem()의 호출 코드를 패치하였기 때문에, 먼저 loinitSystem()를 실행한다.

[그림1-47] ntoskrnl.exe 마지막 세션에 저장된 악성코드

```
nt!_NULL_IMPORT_DESCRIPTOR <PERF> (nt+0x214382):
806ed382 8b442428        mov     eax,dword ptr [esp+28h]
nt!_NULL_IMPORT_DESCRIPTOR <PERF> (nt+0x214386):
806ed386 50              push    eax
nt!_NULL_IMPORT_DESCRIPTOR <PERF> (nt+0x214387):
806ed387 e8af2afbff      call    nt!IoInitSystem (8069fe3b)
```

이후 진행되는 코드는 드라이버를 메모리에 로드 후 해당 드라이버로 접근하는 코드다.

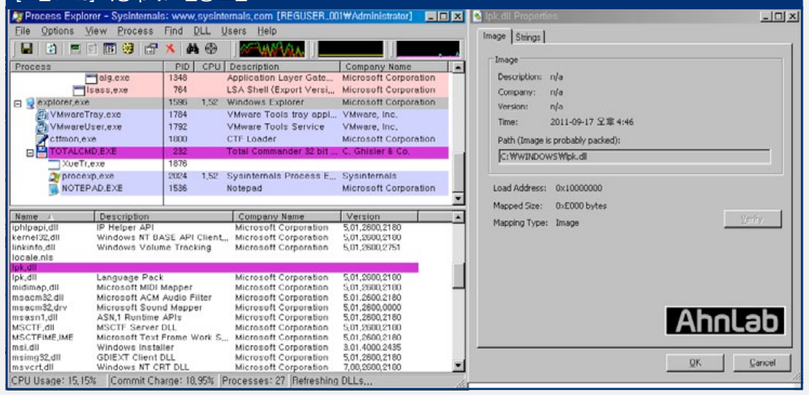
나. 부트킷

해당 부트킷의 역할은 다음과 같다.

1. C:₩Windows 디렉터리에 lpk.dll 명으로 파일을 생성한다.(DLL Preloading)
2. CreateProcessEx() SSDT 후킹 하여 AV 프로그램 및 복원 프로그램을 무력화한다.
3. Atapi.sys의 Major Function을 후킹한다.

1. Start Sector Offset 0x1A00 에서 악성 DLL을 읽어 온다. 해당 악성코드는 이전의 드롭퍼에 의해 생성된 halc.dll과 같은 코드로 읽어들인 악성 DLL을 lpk.dll로 Windows 디렉터리에 저장한다. 이를 통하여 Explorer.exe 나 Notepad.exe와 같이 Windows 디렉터리에 같이 있으면 악성 lpk.dll이 먼저 실행된다.

[그림1-48] 악성 lpk.dll 실행 모습



2. 이후, AV 프로그램과 복원 프로그램을 무력화하기 위해 CreateProcessEx()를 SSDT 후킹한다. 해당 루트킷에 저장된 문자열과 비교하여 같으면, 해당 프로그램의 상태 정보를 0xC0000022로 0xC0000022 STATUS_ACCESS_DENIED를 리턴하여 해당 프로세스를 실행하지 못하게 한다.

[그림1-49] CreateProcessEx() SSDT 후킹

```
804e4750  805d7222 nt!NtCreateMailslotFile      804e4744  805a9166 nt!NtCreateJobObject
804e4754  8057bf88 nt!NtCreateMutant            804e4748  8063027f nt!NtCreateJobSet
804e4758  80581059 nt!NtCreateNamedPipeFile     804e474c  80570761 nt!NtCreateKey
804e475c  805b97b8 nt!NtCreatePagingFile        804e4750  805d7222 nt!NtCreateMailslotFile
804e4760  805983a9 nt!NtCreatePort              804e4754  8057bf88 nt!NtCreateMutant
804e4764  805af314 nt!NtCreateProcess           804e4758  80581059 nt!NtCreateNamedPipeFile
804e4768  8058241a nt!NtCreateProcessEx         804e475c  805b97b8 nt!NtCreatePagingFile
804e476c  80649b59 nt!NtCreateProfile           804e4760  805983a9 nt!NtCreatePort
804e4770  8056641b nt!NtCreateSection           804e4764  805af314 nt!NtCreateProcess
804e4774  805767cf nt!NtCreateSemaphore         804e4768  81db0644
804e4778  8059f4db nt!NtCreateSymbolicLinkObject 804e476c  80649b59 nt!NtCreateProfile
004e477c  0057d1c5 nt!NtCreateThread            804e4770  8056641b nt!NtCreateSection
804e4780  8059c849 nt!NtCreateTimer             804e4774  805767cf nt!NtCreateSemaphore
804e4784  805a6acf nt!NtCreateToken             804e4778  8059f4db nt!NtCreateSymbolicLinkObject
804e4788  805a0d5c nt!NtCreateWaitablePort      804e477c  8057d1c5 nt!NtCreateThread
804e478c  8065b019 nt!NtDebugActiveProcess      804e4780  8059c849 nt!NtCreateTimer
                                                804e4784  805a6acf nt!NtCreateToken
```

[그림 1-50]은 후킹된 코드로 무력화할 대상 보안 프로그램의 프로세스 이름을 확인, 비교하여 STATUS_ACCESS_DENIED를 리턴한다.

[그림1-50] AV 무력화 코드

```
v13 = 0;
memset(v14, 0, 0x204u);
*(_WORD *)&v14[516] = 0;
if ( GetCurrentProcessName_1050A(a6, (int)&v13, (int)&v15) )
{
  v9 = v15;
  while ( 1 )
  {
    --v9;
    if ( v9 <= 0 )
      break;
    if ( *(&v13 + v9) == 0x5C )
    {
      CurrentProcessName = (int)&v14[2 * v9];
      if ( !CurrentProcessName )
        return OrgNtCreateProcessEx_1ED8(a1, a2, a3, a4, a5, a6, a7, a8, a9);
      TargetName = (int)L"Pro.exe";          // Start.exe,Agent.aye,ServiceMT.aye,Vac.aye,Scan.aye,Svc.exe
                                             // SP.exe,Main.exe,PScan.exe,Medic.exe,LTray.exe,LSvc.exe,
                                             // tight.exe,Svc.exe,CAgent.npc,C.npc,vman.npc,avsvc.npc,verAgent.exe
      while ( !CmpProcessName_10280(CurrentProcessName, TargetName) )
      {
        TargetName += 0x22u;
        if ( TargetName >= (signed int)&atapi_INTERNAL_DEVICE_CONTROL_10E98 )
          return OrgNtCreateProcessEx_1ED8(a1, a2, a3, a4, a5, a6, a7, a8, a9);
      }
      return 0xC0000022u;                    // STATUS_ACCESS_DENIED
    }
  }
}
return OrgNtCreateProcessEx_1ED8(a1, a2, a3, a4, a5, a6, a7, a8, a9);
```

AhnLab

악성 lpk.dll은 UPX로 실행 압축 되어 있으며 [그림 1-51]의 문자열에서 보는 바와 같이 DNS 변조 및 host 파일 변조, 다운로드 등 봇(bot)의 기능이 있다.

[그림1-51] AV 무력화 코드

A	00008030	10008030	0	8.8.8.8
A	00008094	10008094	0	208.67.222.222
A	000080F8	100080F8	0	165.87.201.244
A	0000815C	1000815C	0	209.166.160.36
A	000081C0	100081C0	0	168.95.192.1
A	00008338	10008338	0	InternetReadFile
A	0000834C	1000834C	0	InternetCloseHandle
A	00008360	10008360	0	InternetOpenUrlA
A	00008374	10008374	0	InternetOpenA
A	00008384	10008384	0	wininet.dll
A	00008390	10008390	0	127.0.0.1
A	0000839C	1000839C	0	Content-Length
A	000083AE	100083AE	0	%s %s
A	000083B8	100083B8	0	\drivers\etc\hosts
A	000083CC	100083CC	0	%s %s
A	000083D4	100083D4	0	program files\Internet Explorer\IEXPLORE.EXE
A	00008404	10008404	0	SOFTWARE\Microsoft\Internet Explorer\Main
A	00008430	10008430	0	TabProcGrowth
A	00008440	10008440	0	rundll32.exe
A	00008450	10008450	0	IpCount
A	00008460	10008460	0	softurl%d
A	0000846C	1000846C	0	explorer.exe
A	0000847C	1000847C	0	FileCount
A	00008488	10008488	0	Config
A	00008490	10008490	0	TimeKey
A	00008498	10008498	0	error
A	000084A0	100084A0	0	\VersionKey.ini
A	000084B0	100084B0	0	CsExit
A	000084C8	100084C8	0	Range: bytes=
A	000084D8	100084D8	0	Set Cookie:0
A	000084E8	100084E8	0	Connection:Keep-Alive
A	00008500	10008500	0	User-Agent:Mozilla/4.0 (compatible; MSIE 5.00; Windows 98)
A	0000853C	1000853C	0	Accept:*/*
A	00008548	10008548	0	Referer:
A	00008554	10008554	0	Host:
A	00008560	10008560	0	HTTP/1.1
A	0000857C	1000857C	0	ftsWordBreak
A	0000858C	1000858C	0	LpkUseGDIWidthCache
A	000085A0	100085A0	0	LpkPSMTextOut
A	000085B0	100085B0	0	LpkGetTextExtentExPoint
A	000085C8	100085C8	0	LpkEditControl
A	000085D8	100085D8	0	LpkGetCharacterPlacement
A	000085F4	100085F4	0	LpkExtTextOut
A	00008604	10008604	0	LpkDrawTextEx
A	00008614	10008614	0	LpkDllInitialize
A	00008628	10008628	0	LpkTabbedTextOut
A	0000863C	1000863C	0	LpkInitialize
A	0000864C	1000864C	0	\lpk.dll
A	00008658	10008658	0	Global\BktDownUpdate
A	00008670	10008670	0	Global\BktDownVersion
A	00008688	10008688	0	FkDownLoad
A	00008694	10008694	0	\\.\FileEngine
A	000086A4	100086A4	0	Control
A	000086AC	100086AC	0	\\.\Audio

AhnLab

3. 마지막으로 Atapi.sys의 Major Function을 후킹하여 자신을 보호한다.
이후 진행되는 코드는 드라이버를 메모리에 로드 후 해당 드라이버로 접근하는 코드다.

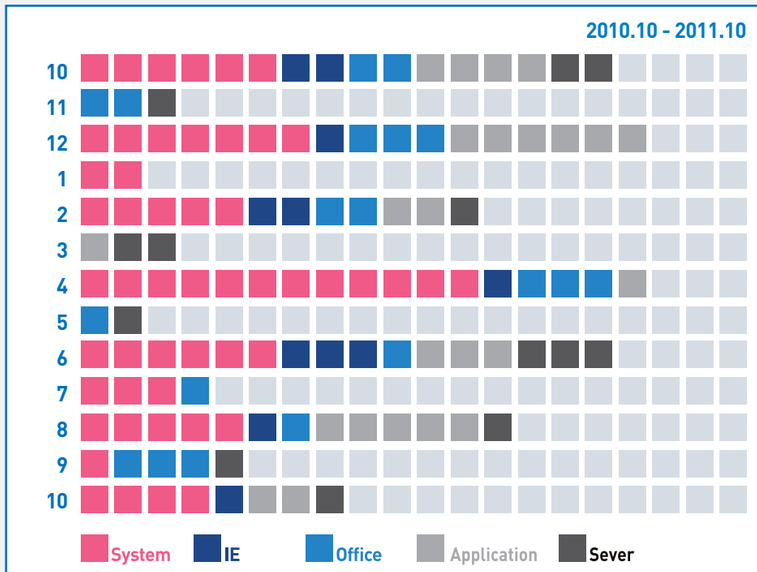
해당 부트킷은 TDL4 와 같이 64bit 시스템에서도 동작하며 감염된 MBR에 의해 백신 프로그램보다 먼저 실행되기 때문에 백신이 무력화 된다. 해당 악성코드는 취약점에 의해 전파되므로 최신 패치의 적용이 필요하며, 앞으로도 루트킷과 MBR를 감염시키는 악성코드가 많이 발생할 수 있으므로 주의가 요구된다.

TDL4 참고: 월간 安 Vol 2011.10 (http://download.ahnlab.com/kr/site/magazineAhn/ahn_201110.pdf)

02. 시큐리티 동향
a. 시큐리티 통계

10월 MS보안 업데이트 현황

마이크로소프트사가 제공하는 이달의 보안 업데이트는 8건이며, 총 8건의 패치가 발표되었다. 긴급을 요하는 것은 2건으로 비교적 낮은 편이었다. 긴급을 요하는 패치 중 인터넷 익스플로러 누적 보안 업데이트에는 개별적으로 보고된 총 8건의 원격 코드 실행 취약점 문제가 해결되었다.



[그림 2-1] 공격 대상 기준별 MS 보안 업데이트

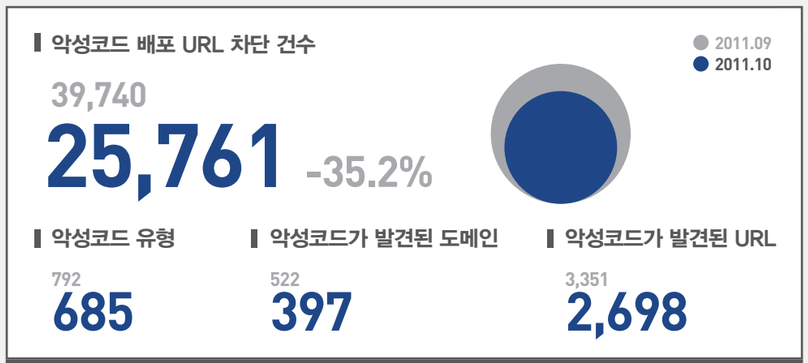
위험도	취약점
긴급	.NET 프레임워크 및 실버라이트의 취약점으로 인한 원격코드 실행(MS11-078)
긴급	인터넷 익스플로러 누적 보안 업데이트(MS11-081)
중요	마이크로소프트 Active Accessibility의 취약점으로 인한 원격코드 실행(MS11-075)
중요	Windows Media Center의 취약점으로 인한 원격코드 실행 문제점(MS11-076)
중요	Windows 커널 모드 드라이버의 취약점으로 인한 원격코드 실행(MS11-077)
중요	Forefront Unified Access Gateway의 취약점으로 인한 원격코드 실행(MS11-079)
중요	Ancillary Function 드라이버의 취약점으로 인한 권한 상승(MS11-080)
중요	Host Integration 서버 취약점으로 인한 서비스 거부(MS11-082)

[표 2-1] 2011년 10월 주요 MS 보안 업데이트

03. 웹 보안 동향
a. 웹 보안 통계

웹사이트 보안 요약

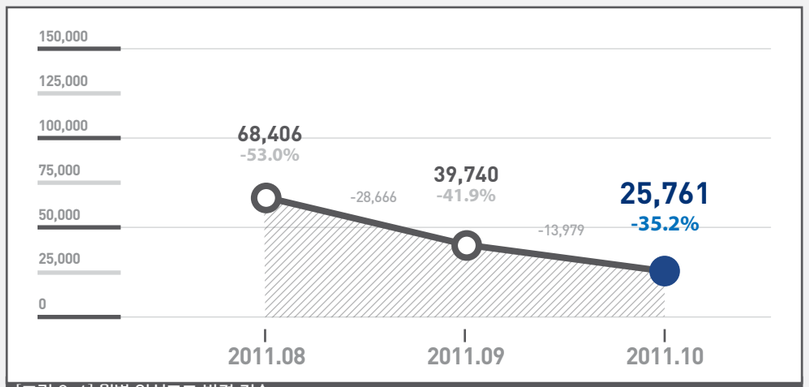
안철수연구소의 웹 브라우저 보안 서비스 사이트가드(SiteGuard)를 통해 웹 사이트 보안 통계 자료에 따르면, 산출한 2011년 10월 악성코드를 배포하는 웹 사이트의 차단 건수는 25,761건이다. 또한 악성코드 유형은 685건이며, 악성코드가 발견된 도메인은 397건, 악성코드가 발견된 URL은 2,698건이다. 2011년 10월에는 2011년 9월보다 악성코드 발견 건수, 악성코드 유형, 악성코드가 발견된 도메인, 악성코드가 발견된 URL이 전반적으로 감소하였다.



[표 3-1] 웹 사이트 보안 요약

월별 악성코드 배포 URL 차단 건수

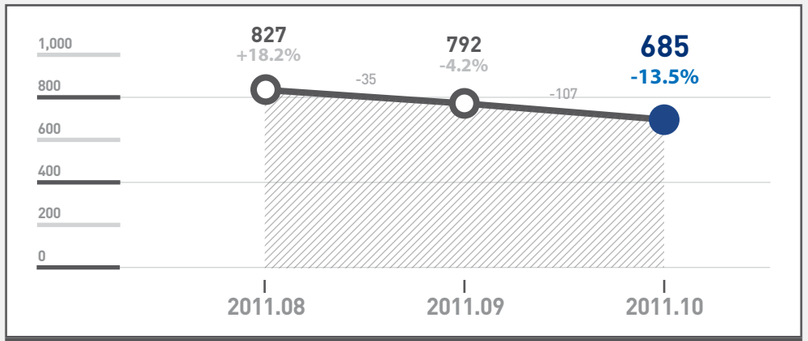
2011년 10월 악성코드 배포 웹 사이트 URL 접근에 따른 차단 건수는 지난달 39,740건에 비해 65% 수준인 25,761건에 그쳤다.



[그림 3-1] 월별 악성코드 발견 건수

월별 악성코드 유형

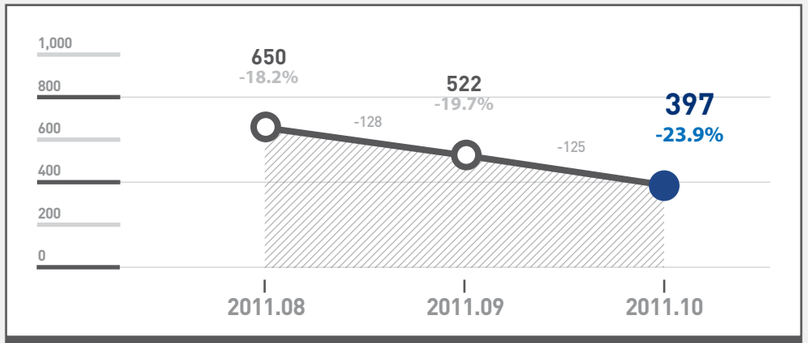
2011년 10월 악성코드 유형은 전달의 792건의 86% 수준인 685건이다.



[그림 3-2] 월별 악성코드 유형

월별 악성코드가 발견된 도메인

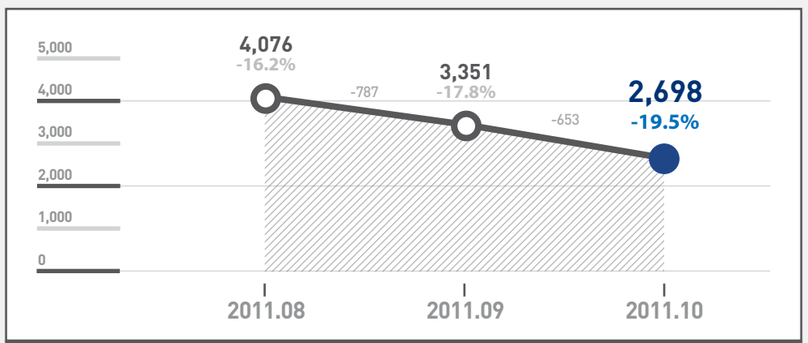
2011년 10월 악성코드가 발견된 도메인은 전달의 522건에 비해 76% 수준인 397건이다.



[그림 3-3] 월별 악성코드가 발견된 도메인

월별 악성코드가 발견된 URL

2011년 10월 악성코드가 발견된 URL은 전달의 3,351건에 비해 81% 수준인 2,698건이다.



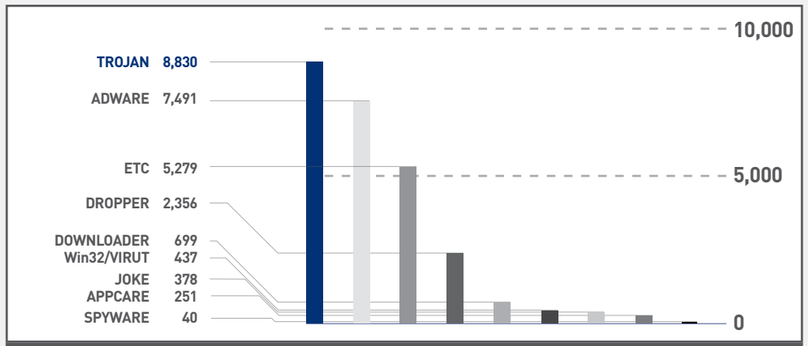
[그림 3-4] 월별 악성코드가 발견된 URL

악성코드 유형별 배포 수

악성코드 유형별 배포 수에서 트로이목마가 8,830건으로 전체의 34.3%로 가장 많았고, 애드웨어가 7,491건/29.1%로 2위였다.

유형	건수	비율
TROJAN	**8,830**	**34.3 %**
ADWARE	7,491	29.1 %
DROPPER	2,356	9.1 %
DOWNLOADER	699	2.7 %
Win32/VIRUT	437	1.7 %
JOKE	378	1.5 %
APPCARE	251	1.0 %
SPYWARE	40	0.1 %
ETC	5,279	20.5 %
	25,761	**100 %**

[표 3-2] 악성코드 유형별 배포 수



[그림 3-5] 악성코드 유형별 배포 수

악성코드 배포 순위

악성코드 배포 순위는 Win-Adware/ToolBar.Cashon.308224가 2,938건으로 가장 높았으며 Packed/Upack, Dropper/Small.Gen 등 4건이 Top10에 새로 등장하였다.

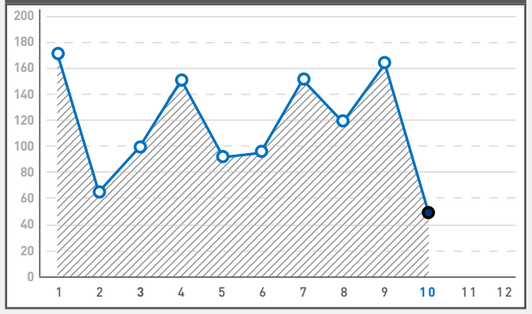
순위	등락	악성코드명	건수	비율
1	**—**	**Win-Adware/ToolBar.Cashon.308224**	**2,938**	**28.7 %**
2	▲2	Win-Adware/FunWeb.210992.D	1,228	12.0 %
3	▲4	Win-Trojan/Buzus.430080.J	1,154	11.3 %
4	▲5	Win-Trojan/StartPage.40960.AH	857	8.4 %
5	▼2	Dropper/Kgen.225280.M	765	7.5 %
6	NEW	Packed/Upack	739	7.2 %
7	NEW	Dropper/Small.Gen	705	6.9 %
8	▼2	Win32/Induc	698	6.8 %
9	NEW	Win-Trojan/Peed.44416.B	588	5.7 %
10	NEW	ALS/Bursted	555	5.5 %
		10,227	**100.0 %**	

[표 3-3] 악성코드 배포 Top 10

03. 웹 보안 동향
b. 웹 보안 이슈

2011년 10월 침해 사이트 현황

[그림 3-6] 2011년 10월 악성코드 유포 목적의 침해 사이트 현황



[그림 3-6]은 악성코드 유포를 목적으로 하는 침해 사고가 발생했던 사이트의 현황이다. 2011년 10월의 경우, 9월보다 그 수가 상당히 감소하였으나 원인은 불분명하다.

[표 3-4] 침해 사이트를 통해서 유포된 악성코드 Top 10

순위	악성코드명	건수
1	Dropper/Win32.OnlineGameHack	32
2	Win-Trojan/Onlinegamehack55.Gen	29
2	Win-Trojan/Onlinegamehack56.Gen	29
4	Win-Trojan/Onlinegamehack69.Gen	20
5	Win-Trojan/PatchedImm5.Gen	15
6	Dropper/Win32.OnlineGameHack	11
7	Win-Trojan/Onlinegamehack.84992.CC	11
8	Dropper/Win32.OnlineGameHack	10
9	Win-Trojan/PatchedImm7.Gen	9
10	Dropper/Win32.Rootkit	9

[표 3-4]는 한 달간 침해 사이트를 통해서 가장 많이 유포된 악성코드 Top 100이다. 2011년 10월의 경우 Dropper/Win32.OnlineGameHack이 32개의 사이트에서 유포됐다. 1, 6, 8위의 경우 진단명은 같지만 온라인 게임 사용자의 계정 정보를 탈취하기 위해 각기 다른 사이트에서 유포되었다.

VOL. 22
ASEC REPORT Contributors

집필진
책임 연구원 정 관 진
선임 연구원 안 창 용
선임 연구원 장 영 준
주임 연구원 이 도 현
연구원 이 도 한

참여연구원 ASEC 연구원
SiteGuard 연구원

편집장
선임 연구원 안 형 봉

편집인 안철수연구소
마케팅실

디자인 안철수연구소
UX디자인팀

감 수
상 무 조 시 행

발행처 (주)안철수연구소
경기도 성남시 분당구
삼평동 673
(경기도 성남시 분당구
판교역로 220)
T. 031-722-8000
F. 031-722-8901

Disclosure to or reproduction
for others without the specific
written authorization of AhnLab is
prohibited.

Copyright (c) AhnLab, Inc.
All rights reserved.

AhnLab