

POC2008 Reverse Engineering Report

Reverse Engineering 문제
분석 결과 보고서

2008.11.14

목 차 (총 51페이지)

- I. Reversing 문제 분석
 - ➔ Reversing 문제 요약
 - ➔ 상세 분석
- II. Script 문제 분석
 - ➔ Script 문제 요약
 - ➔ 상세 분석
- III. Network 문제 분석
 - ➔ 요약 분석
 - ➔ 상세분석
- IV. Spyware 문제 분석
 - I. 요약 분석
 - II. 상세 분석

□ 문제 목표

- 코드 보호를 위하여 실행 압축된 파일을 분석하여 원하는 Key 값 확인
- Process 점검을 이용한 Anti-debugging 기법을 우회하여 원하는 Key 값 확인

□ 풀이 환경

- 분석도구 : Olly-debugger, IDA Pro

□ 풀이 방법

- 실행압축 알고리즘을 감추기 위해 Signature를 변조하더라도 프로그램, 실행 시 메모리에는 압축이 풀린 상태로 적재된다는 점에 착안하여 Debugging 수행.
- 프로그램 실행 후 Debugging을 수행하는 방법을 이용하여, 실행 초기에 수행되는 Anti-debugging 기법을 우회.

□ 풀이 결과

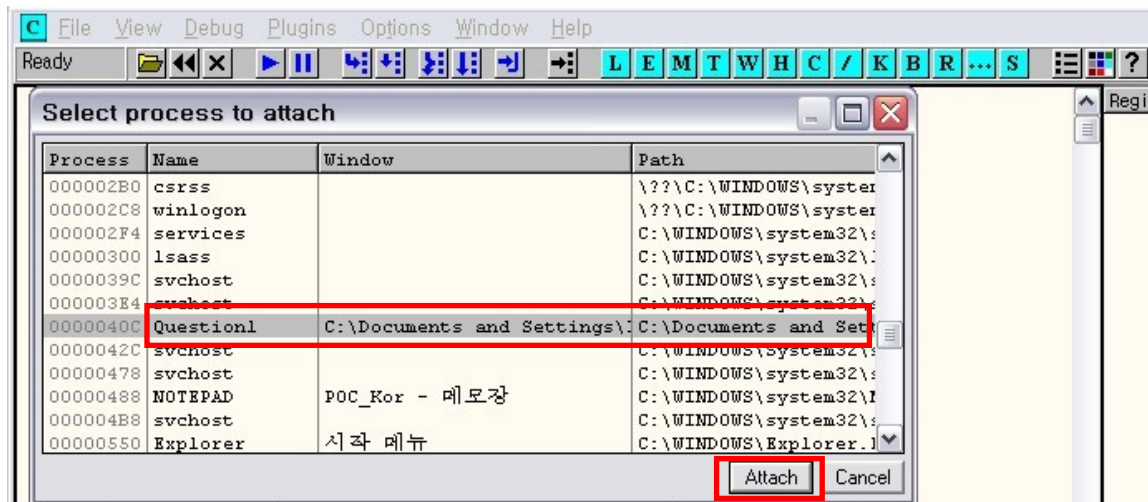
- 1번 문제의 답 : Beginner
- 2번 문제의 답 : POCACHIP

❑ Question1.exe 문제 상황

- Question1.exe를 Disassemble하기 위하여 Ollydbg나 IDA pro를 이용하여 분석을 시도할 경우 실행파일 압축으로 인하여 분석 진행 불가
- Unpack을 위한 Peid 등을 이용한 압축알고리즘을 확인 불가 (실행 압축 후 Signature에 해당 되는 부분이 변조 됨) – (MUP 지양)

❑ 상세 풀이 방법

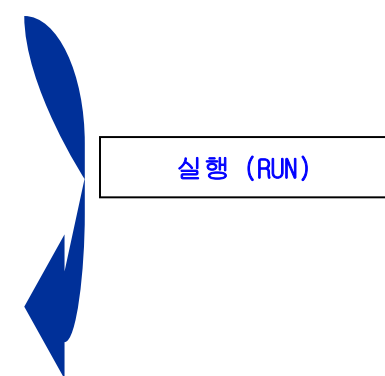
- Question1.exe 실행 후 Ollydbg의 Attach 기능을 이용하여 압축이 풀린 상태로 메모리에 적대된 실행코드에 접근.



□ Question1.exe 상세분석 (계속)

- Attach 후 Process를 Run시킨 후 Memory map을 통하여 Question1.exe의 .code 섹션에 Breakepoint 설정.
- 임의의 입력 값과 숨겨진 Serial 번호를 비교하기 위해 Question1.exe의 .code 섹션에서 비교 Code가 실행되는 순간에 프로그램이 Break 되도록 설정할 수 있음.

Address	Size	Owner	Section	Contains	Type	Access	Initial	Mapped as
00380000	00003000	Teerayoo		PE header	Map	R	R	\\Device\\HarddiskVolu
003A0000	00001000	Teerayoo			Image	R	RWE	
003A1000	0003E000	Teerayoo	.text	code	Image	R	RWE	
003DF000	0000A000	Teerayoo	.data	data	Image	R	RWE	
003E9000	00001000	Teerayoo	.tls		Image	R	RWE	
003EA000	00002000	Teerayoo	.idata	imports	Image	R	RWE	
003EC000	00004000	Teerayoo	.edata	exports	Image	R	RWE	
003F0000	00002000	Teerayoo	.rsrc	resources	Image	R	RWE	
003F2000	00004000	Teerayoo	.reloc	relocations	Image	R	RWE	
00400000	00001000	Question		PE header	Image	R	RWE	
00401000	00006000	Question	.text	code	Image	R	RWE	
00407000	00001000	Question	.rdata	imports	Image	R	RWE	
00408000	00004000	Question	.data	data	Image	R	RWE	



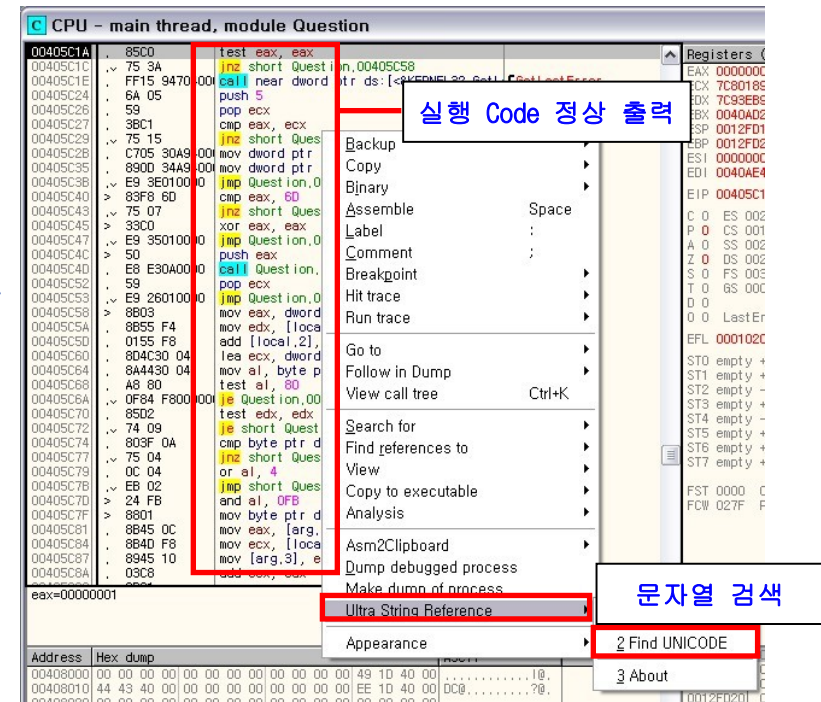
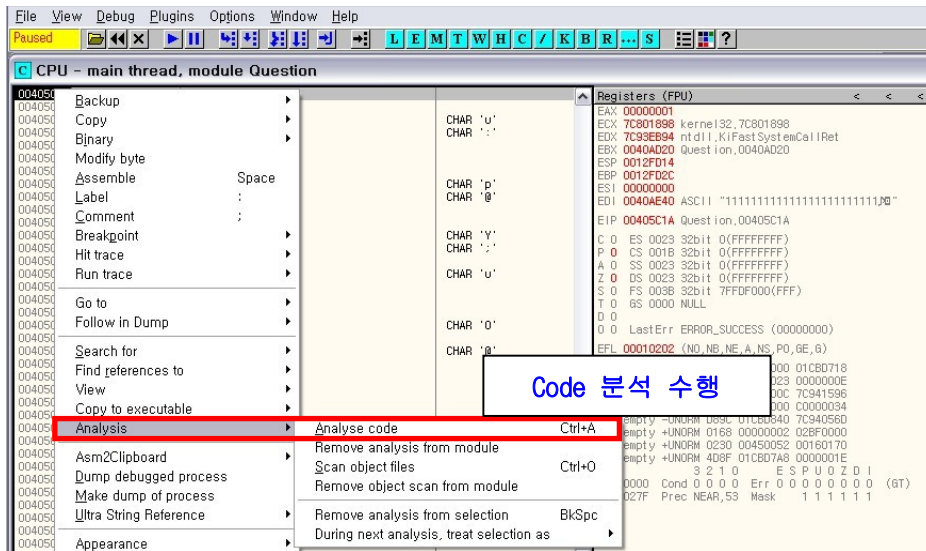
Break 걸림

임의의 값 입력

❑ Question1.exe 상세분석 (계속)

— “Analyse code”를 이용하여 .code 섹션 부분의 실행코드 분석 수행.

— 분석 된 실행 코드 내의 문자열 검색을 이용하여 정답 및 오답에 대한 문자열이 위치하는 부분으로 이동.



1. Reversing 문제 분석

II. 상세 분석

❑ Question1.exe 상세분석 (계속)

- 해당 위치로 이동하면, 임의로 입력한 값과 실제 Key값을 비교하는 부분을 확인할 수 있음
- 최종 Key 값을 비교하는 함수에 Breakpoint를 걸고 프로그램을 실행시키면 Register 내에 입력한 값과 실제 Key 값을 비교할 수 있음.

```
Ultra String Reference
Address Disassembly Text String
00401078 push Question,00408054 What is the KEY? :
00401089 push Question,00408050 %s
004010AC push Question,00408040 Good Job !!
004010C6 push Question,00408030 You'r Wrong !!
004016EF mov edi, Question,00408080
004017FF mov edi, Question,00408078
~::~~::~~::~
~::~~::~~::~
```



```
CPU - main thread, module Question
00401069 . 8A4415 E8 mov al, byte ptr ss:[ebp+edx-
0040106D . 04 01 add al, 1
0040106F . 8B4D E4 mov ecx, [local,7]
00401072 . 88440D E8 mov byte ptr ss:[ebp+ecx-18],
00401076 . EB DF jmp short Question,00401057
00401078 > 68 54804000 push Question,00408054 ASCII "What i
0040107D . E8 BD000000 call Question,0040113F
00401082 . 83C4 04 add esp, 4
00401085 . 8D55 F4 lea edx, [local,3]
00401088 . 52 push edx
00401089 . 68 50804000 push Question,00408050 ASCII "%s"
0040108E . E8 95000000 call Question,00401128
00401093 . 83C4 08 add esp, 8
00401096 . 6A 08 push 8
00401098 . 8D45 E8 lea eax, [local,6]
0040109B . 50 push eax
0040109C . 8D4D F4 lea ecx, [local,3]
0040109F . 51 push ecx
004010A0 . E8 4B000000 call Question,004010F0
004010A5 . 83C4 0C add esp, 0C
~::~~::~~::~
```

입력결과 비교

```
CPU - main thread, module Question
00401069 . 8A4415 E8 mov al, byte ptr ss:[ebp+edx-
0040106D . 04 01 add al, 1
0040106F . 8B4D E4 mov ecx, [local,7]
00401072 . 88440D E8 mov byte ptr ss:[ebp+ecx-18],
00401076 . EB DF jmp short Question,00401057
00401078 > 68 54804000 push Question,00408054 ASCII "What is the KEY? :
0040107D . E8 BD000000 call Question,0040113F
00401082 . 83C4 04 add esp, 4
00401085 . 8D55 F4 lea edx, [local,3]
00401088 . 52 push edx
00401089 . 68 50804000 push Question,00408050 ASCII "%s"
0040108E . E8 95000000 call Question,00401128
00401093 . 83C4 08 add esp, 8
00401096 . 6A 08 push 8
00401098 . 8D45 E8 lea eax, [local,6]
0040109B . 50 push eax
0040109C . 8D4D F4 lea ecx, [local,3]
0040109F . 51 push ecx
004010A0 . E8 4B000000 call Question,004010F0
004010A5 . 83C4 0C add esp, 0C
004010A8 . 85C0 test eax, eax
004010AA . 75 1A jnz short Question,004010C6
004010AC . 68 40804000 push Question,00408040 ASCII "Good Job !! "
004010B1 . E8 89000000 call Question,0040113F
004010B6 . 83C4 04 add esp, 4
004010B9 . 68 10270000 push 2710
004010BE . FF15 00704000 call near dword ptr ds:[<&KERN
004010C6 . EB 18 jmp short Question,004010DE
004010C6 > 68 30804000 push Question,00408030 ASCII "You'r Wrong !!"
004010CB . E8 6F000000 call Question,0040113F
004010D0 . 83C4 04 add esp, 4
004010D3 . 68 10270000 push 2710
004010D8 . FF15 00704000 call near dword ptr ds:[<&KERN
~::~~::~~::~
```

입력결과 비교

비교결과 확인

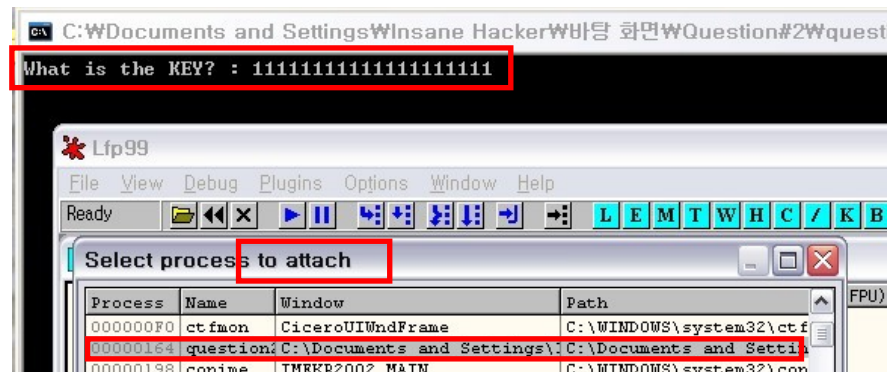


□ Question2.exe 문제 상황

- Question2.exe를 Disassemble하기 위하여 Ollydbg나 IDA pro를 이용하여 분석을 시도할 경우 실행파일 압축으로 인하여 분석 진행 불가
- Unpack을 위한 Peid 등을 이용한 압축알고리즘을 확인 불가 (실행 압축 후 Signature에 해당 되는 부분이 변조 됨)
- Ollydbg가 실행된 상태에서 Question2.exe 실행 시 Ollydbg 실행 종료 (몇 개의 특정 Process 확인 후 실행중단 시키는 Anti-Debugging 기법 적용 됨)

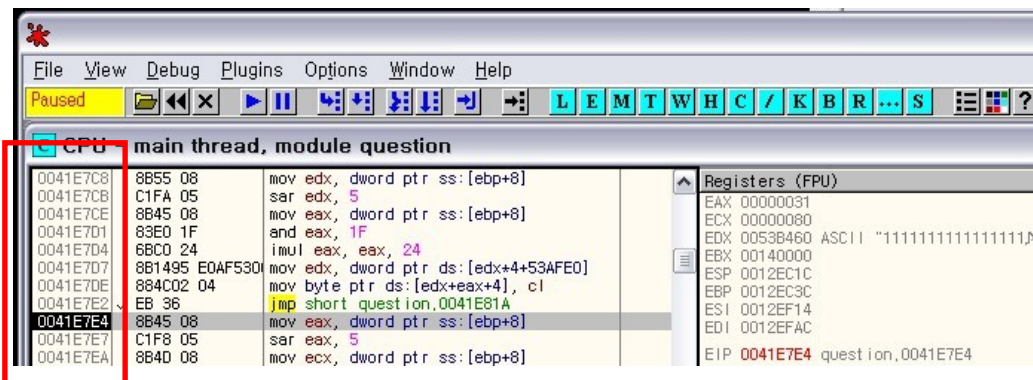
□ Question2.exe 상세 분석

- Question2.exe 실행 후 Ollydbg의 Attach 기능을 이용하여 압축이 풀린 상태로 메모리에 적대된 실행코드에 접근.

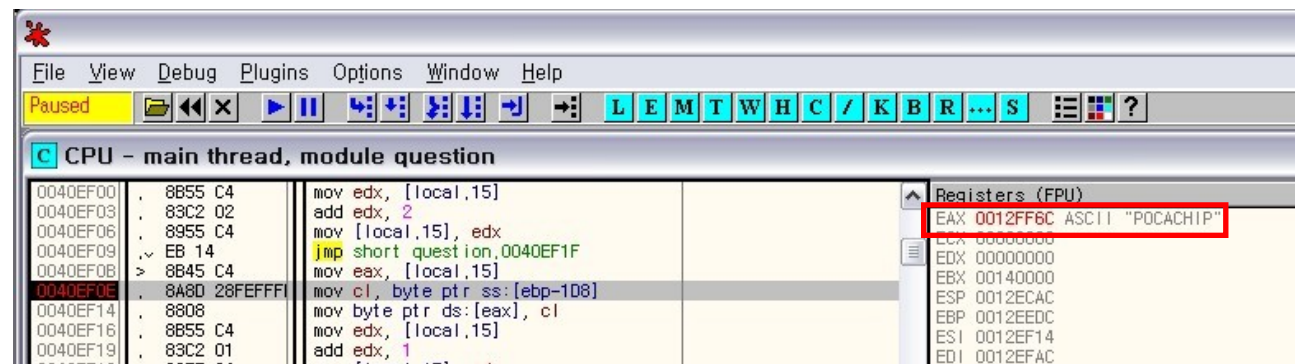


□ Question2.exe 상세분석 (계속)

- 현재 코드의 실행이 Kernel 부분에 멈춰 있으나, 실제 정답과 사용자 입력 값을 비교하는 부분은 .code 섹션 내 존재하므로 'F7' 혹은 'F8'과 같은 단계별 실행 키를 이용하여 Question2.exe의 .code 섹션으로 이동.
- Question2.exe의 .code 섹션으로 이동 후 지속적으로 단계별 수행을 하며 Register 값의 변화를 살펴다 보면, 원하는 문자열을 찾을 수 있음.

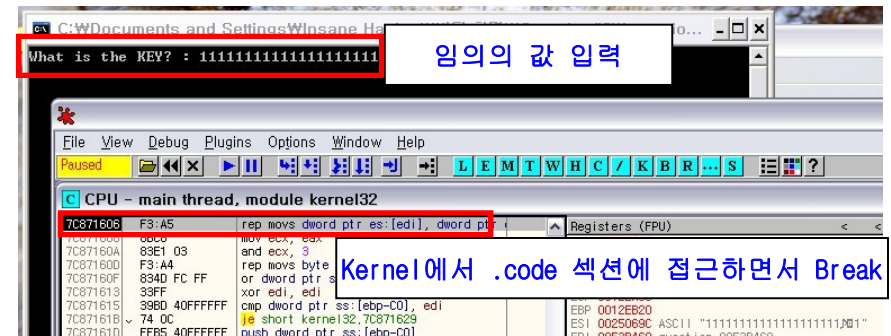
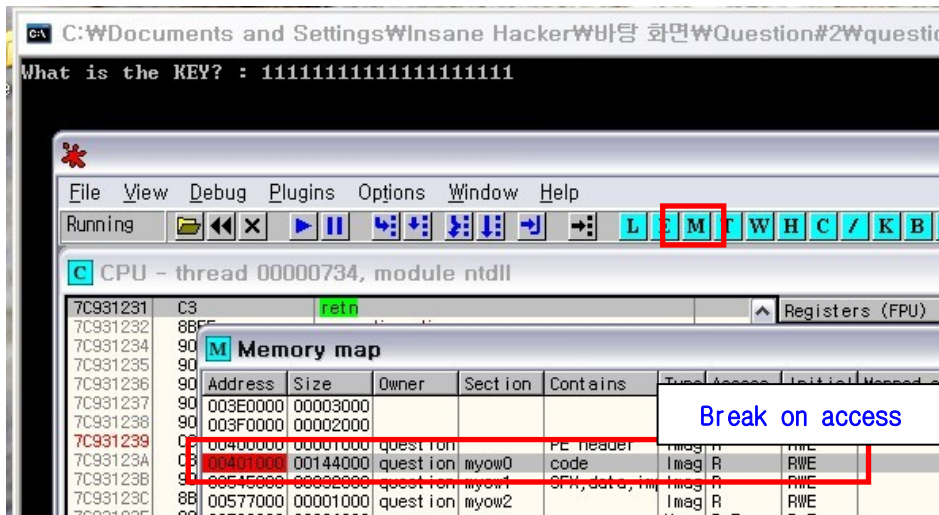


.code 섹션으로 이동



❑ Question2.exe 상세분석 (계속)

- Attach 후 Process를 Run시킨 후 Memory map을 통하여 Question1.exe의 .code 섹션에 Breakpoint 설정.
- 임의의 입력 값 입력 후 숨겨진 Serial 번호와 비교하기 위해 Kernel 에서 Question2.exe의 .code 섹션에 접근하는 순간 Break가 됨.



❑ Question2.exe 상세분석 (계속)

- 또한, Question2.exe의 .code 섹션 상단으로 이동하여 보면, 실행압축 해제 후 새로 만들어진 코드의 OEP를 찾을 수 있으며, 그 밑으로 실행 흐름을 따라가 보면, 아래와 같이 Question2.exe의 Anti-debugging 기법에 의하여 종료되는 대상 Process의 목록도 확인 가능함.

0040160A	8945 C1	mov dword ptr ss:[ebp-1F], eax	
0040160D	8945 E5	mov dword ptr ss:[ebp-1B], eax	
00401610	C745 CC ACB0	mov dword ptr ss:[ebp-34], question,005	ASCII "OLLYDBG,EXE"
00401617	C745 D0 A0B0	mov dword ptr ss:[ebp-30], question,005	ASCII "idag.exe"
0040161E	C745 D4 90B0	mov dword ptr ss:[ebp-2C], question,005	ASCII "WINDBG,EXE"
00401625	C745 D8 80B0	mov dword ptr ss:[ebp-28], question,005	ASCII "proccxp.exe"
0040162C	C745 DC 70B0	mov dword ptr ss:[ebp-24], question,005	ASCII "taskmgr.exe"
00401633	C785 2CF0FFF	mov dword ptr ss:[ebp-FD4], 0	
0040163D	R9 F7030000	mov ecx, 3F7	

- 상세 Anti-Debugging 기법이 적용된 부분들...

```

call ds:BeModuleHandler
cmp esi, esp
call ds:Sleep
mov esi, esp
push 0h
call ds:Sleep
cmp esi, esp
pop eax
mov esi, esp
push 0
call ds:BeModuleHandler
cmp esi, esp
call ds:Sleep
mov esi, esp
push 0
call ds:BeModuleHandler
cmp esi, esp
call ds:Sleep
mov eax, large fs:18h
mov eax, [eax+30h]
movzx eax, byte ptr [eax+2]
test eax, eax
jnz short loc_4022D1
    
```

BeingDebugged Anti-Debugging

```

mov edx, edx
nop
nop
push eax
nop
push eax
nop
rdtsc
cmp edx, ebx
ja Return3
    
```

RDTSC Anti-Debugging

```

add ax, 40h
add ax, 0
pop ax
push ax
sub ax, 17h
add ax, 27h
add ax, 57h
add ax, 50h
pop ax
xor ebx, ebx
mov ebx, large fs:10h
cmp byte ptr [ebx+0Ch], 0
jnz Return3
    
```

NtGlobalFlag, HeapFlags Anti-Debugging

```

push 0
call ds:Sleep
cmp esi, esp
call ds:Sleep
mov esi, esp
push 0
call ds:BeModuleHandler
cmp esi, esp
call BeingDebugged_Check
add eax, 0C6h
test eax, eax
jz short loc_4022D8
    
```

BeingDebugged Anti-Debugging

```

call ds:Sleep
cmp esi, esp
call chksesp
push offset main
call CheckBreakPoint
add esp, 4
and eax, 0FFh
test eax, eax
jz short loc_4023C7
    
```

Break point check Anti-Debugging

- 문제 목표
 - Encoding된 HTML 파일을 Decoding하여 얻은 Javascript 코드를 분석하여 원하는 Password 획득
 - Adobe Reader의 Buffer overflow 취약점을 이용한 공격을 위하여 임의로 삽입된 Java script를 추출, 분석하여 원하는 Password 획득

- 풀이 환경
 - 분석도구 : Internet Explorer, notepad, Ultraedit, pdftk(Linux)

- 풀이 방법
 - US-ASCII 및 escape로 encoding된 source code를 decoding.
 - Pdftk를 이용하여 pdf 內 은닉/압축된 Stream을 추출 및 decoding 수행.

- 풀이결과
 - 1번 문제의 답 : ASEC is AhnLab Security E-response Center
 - 2번 문제의 답 : Do U Know ASEC?

□ Script 1번 문제 상세 분석

- Notepad.exe를 이용하여 문제의 파일을 열어보면 US-ASCII 형식으로 Encoding 된 HTML파일이라는 것을 알 수 있음.
- Internet Explorer에서 실행되는 Minesweeper를 통해 문제는 US-ASCII로 Encoding된 Java script라는 사실을 유추할 수 있음.



※ 부연설명: US-ASCII 인코딩 방식은 구형 컴퓨터에서 문자를 표현할 때 쓰이던 방식이며, 1개의 문자는 8비트(1바이트)로 표현되는데, US-ASCII 방식은 이중 7비트를 취하고 나머지 1비트를 버린다. 예를 들어, 화면상에 문자를 표현하고자 때 10110101과 00110101은 같은 결과를 보여준다.

□ Script 1번 문제 상세 분석 (계속)

— Ultra Editor를 이용하여 해당 문자열을 수작업을 통해 일부 검증해 보면 Plain text를 볼 수

있음. (1000000은 Hexcode로 0x80이므로, 각 문자의 Hex값에서 0x80만큼 빼주면 원하는 문자를 얻을 수 있음)

— 간단한 C code를 이용하여 전체 Code를 US-ASCII 방식으로 Decoding하여 Source code 추출

```
41 53 43 49 49 22 20 2F 3E 0D 0A 0C C2 C1 D3 C5 ; ASCII" />...<BASE
A0 C8 D2 C5 C6 BD A2 E8 F4 F4 F0 BA AF AF E1 E8 ; 쥔璿력?諱齊?訟
EE EC E1 E2 AD F3 E5 E3 F5 F2 E9 F4 F9 AE E3 EF ; 佢濃?憶瑋郁陂薪
ED AF E7 E1 ED E5 AF A2 BE 8A 8A BC D3 C3 D2 C9 ; 攷詣張?壽壽痰膚
D0 D4 A0 CC C1 CE C7 D5 C1 C7 C5 BD A2 CA E1 F6 ; 吟觀壼합장탐☐囚
E1 D3 E3 F2 E9 F0 F4 A2 A0 E9 E4 BD A2 ED E9 EE ; 驪訊昆淒香鶻?或
E5 C7 E1 ED E5 B1 A2 BE 8A AF AF A0 AA AA AA AA ; 洋殺捺♥뽕꺼오오
AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA ; 오오오오오오오
AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA ; 오오오오오오오
```

```
41 53 43 49 49 22 20 2F 3E 0D 0A 3C 42 41 53 45 ; ASCII" />...<BASE
20 48 52 45 46 3D 22 68 74 74 70 3A 2F 2F 61 68 ; HREF="http://ah
6E 6C 61 62 2D 73 65 63 75 72 69 74 79 2E 63 6F ; nlab-security.co
6D 2F 67 61 6D 65 2F 22 3E 0A 0A 3C 53 43 52 49 ; m/game/">...<SCRI
50 54 20 4C 41 4E 47 55 41 47 45 3D 22 4A 61 76 ; PT LANGUAGE="Jav
61 53 63 72 69 70 74 22 20 69 64 3D 22 6D 69 6E ; aScript" id="min
65 47 61 6D 65 31 22 3E 0A 2F 2F 20 2A 2A 2A 2A ; eGame1">...// ****
2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A ; *****
2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A ; *****
2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A ; *****
```



```
#include <iostream>
#include <fstream>

using namespace std;
int main() {
    unsigned char c;
    fstream in("index.html",ios::in); // 괴상한 파일 읽기
    fstream out("modified.html",ios::out); // 결과를.
    while(1) {
        in >> c; // 한글자씩 읽기.
        if (in.eof())
            break;
        else if (c > 0x80)
            c -= 0x80;
        out << c; // 파일에 출력
    }
    in.close();
    out.close();
    return 0;
}
```

수작업 검증

추출 코드

□ Script 1번 문제 상세 분석 (계속)

— 분석된 Encoding 문자열에서는 미정의된 함수 및 변수에 대한 정보가 없으므로, 문제의 전체 실행코드를 분석하여 나머지 미정의 함수 및 변수에 대한 선언 부분을 검색해 보면, “mineGame2” 부분의 mineCount2 변수가 이름과 다르게 특정 문자열을 저장하는 변수로 사용되고 있음을 알 수 있음.

— 이 변수의 내용을 출력하도록 Java script 작성 후 웹 브라우저를 통해 결과를 확인해보면, 숨겨진 변수 및 함수를 정의하는 source code를 얻을 수 있음.

```
index_modified - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
mineName2=mineList2[c];
for(f=0; f < d; f++)
{
    y = ((mineName2.length - (8*d)) + (f*8));
    v = 0;
    for(x = 0; x < 8; x++)
    {
        if(mineName2.charCodeAt(x+y) > 9)
        {
            v++;
        }
        if(x != 7)
        {
            v = v << 1;
        }
    }
    mineCount2 += String.fromCharCode(v);
}
document.write("<xmp>" + mineCount2 + "</xmp>");
//Mouse Over New Buttons
```

문자열 생성 부분

생성된 문자열 확인

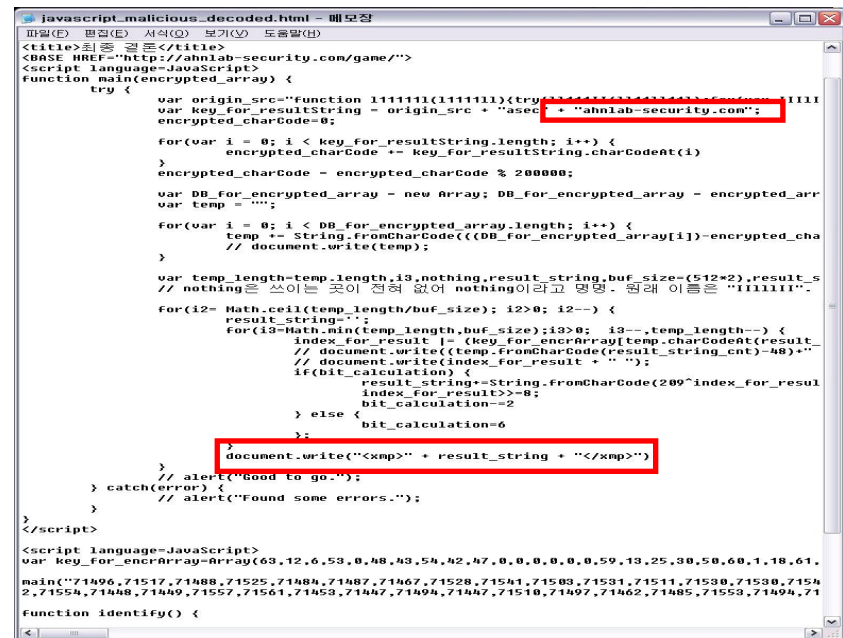
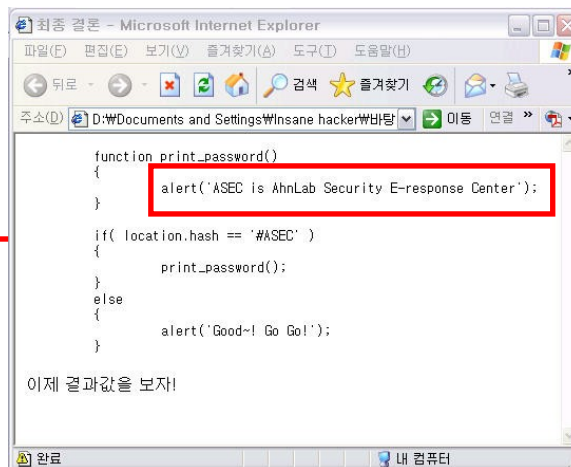


숨겨진 정의 부분

□ Script 1번 문제 상세 분석 (계속)

- 얻어진 source code를 모두 조합하여 정리해 보면 미정의 함수 및 변수의 문제는 해결이 되지만, 테스트를 수행하는 환경이 사용자의PC이므로 location.hostname을 읽는 부분에서 문제가 발생할 것이라는 것을 짐작할 수 있음.
- Index.html을 decoding하여 얻은 source code의 최상단에서 정의된 <base href=http://ahnlab-security.com/game/>에 착안하여 location.hostname을 “ahnlab-security.com”으로 치환한 후 웹 브라우저를 통해 최종 결과를 확인하면 원하는 Password를 얻을 수 있음.

```
<script>
var 11111111 = "var 11111111 = arguments.callee.toString();
var 11111111 = 11111111 + '#' + asec + location.hostname;
var 11111111 = 0;";
11111111=eval;
</script>
```



□ Script 2번 문제 상세 분석 (계속)

- 문제의 Stream을 decoding하기 위하여 Linux 기반의 pdftk라는 유틸리티를 이용하면, 평문화된 문자열을 얻을 수 있음.

```
xterm
Security:/home/linuxpro# ls ./pdf*
./poc2008.pdf_
Security:/home/linuxpro# pdftk poc2008.pdf_ output ./ext2_pdf.txt uncompress
Security:/home/linuxpro#
Security:/home/linuxpro#
```

Stream 추출 및 압축해제

- Plain-Text로 변환된 문서를 Wordpad를 이용하여 확인하면 아래와 같이 두개의 decoding된 code를 확인할 수 있음.

```
stream
0 0 595.28000 841.89000 re W n
endstream
```

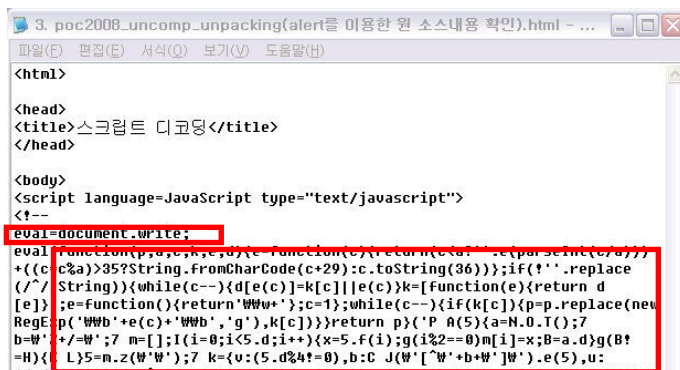
첫번째 decoding된 stream

```
stream
eval(function(p,a,c,k,e,d){e=function(c){return(c<a?'':e(parseInt(c/a)))+(c=c%a)>35?String.fromCharCode(c+29):c.toString(36)};if(!''.replace(/^/,String)){while(c--){d[e(c)]=k[c]||e(c)}k=[function(e){return d[e]}];e=function(){return 'w+w+'};c=1};while(c--){if(k[c]){p=p.replace(new RegExp('w+w'+e(c)+'w+w','g'),k[c])}return p}('P A(5){a=N.O.T);7 b=w+Z+/=w+;7 m=[];!(i=0;i<5.d;i++){x=5.f(i);g(!%2=0)m[i]=x;B=a.d}g(B)=H)}k L)5=m.z(w+w');7 k={v:(5.d%4=0),b:C J(w+['w'+b+w']w+w').e(5),u:(/=/.e(5)&&/=[^=]/.e(5)||/=3/.e(5))};g(k.v|k.b|k.u)G C S(w+U w+w');7 9= [];7 c=0;x(c<5.d){7 w=b.h(5.f(c++));7 t=b.h(5.f(c++));7 l=b.h(5.f(c++));7 n=b.h(5.f(c++));7 j=(w<<Y)+(t<<M)+((l&F)<<6)+(n&F);7 D=(j&(p<<E))>>E;7 s=(l=y)?-1:(j&(p<<8))>>8;7 r=(n=y)?-1:(j&p);9[9.d]=q.o(D);g(s)=0;9[9.d]=q.o(s);g(r)=0;9[9.d]=q.o(r)}e=9.z(w+w');Q(e)}A(w+R=v+v w+w');',62,62,'| | | | |str| |var| |decoded| |chars| |length| |test| |charAt| |if| |indexOf| |buff| |invalid| |i2| |ee| |i3| |fromCharCode| |255| |String| |b2| |b1| |it |equals| |strlen| |id| |key| |64| |join| |AhnLab_ASec| |ss| |new| |b0| |16| |63| |throw| |1017| |for| |RegExp| |return| |false| |12 |arguments| |callee| |function| |eval| |dVmVFvYVIVHVAVgVPVSVAVnVUVGVFVzVcV3VdVwVcVmVQVgVaVXVMVgVIVkVRVwVIVFVUVgVSV2V5VvVdVyVVBVBV UVDVVDVVPyVIVnVOVwV|Error|toString|Invalid| |data| |while| |18| |ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'.split ('').0,{}])
endstream
```

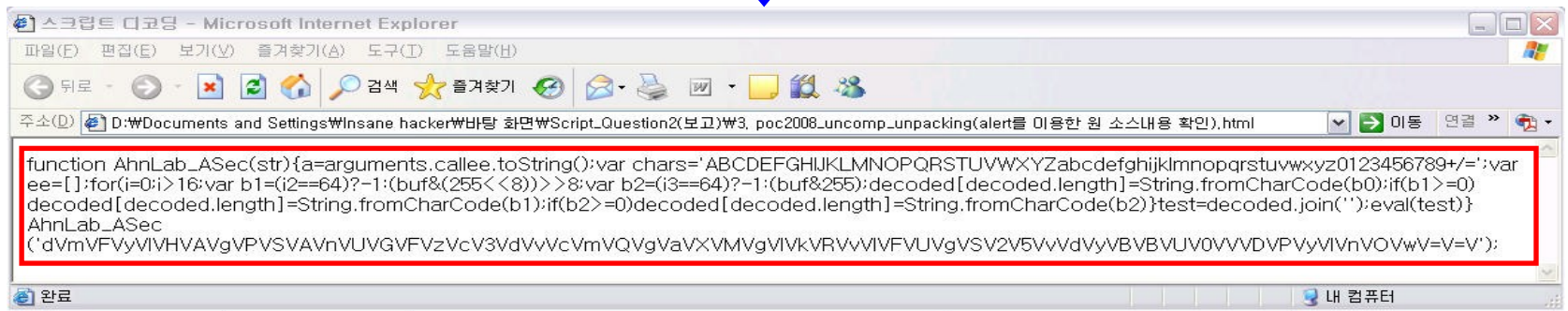
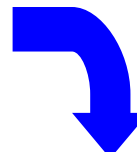
두번째 decoding된 stream

□ Script 2번 문제 상세 분석 (계속)

- 얻어진 Script는 “eval(function(p,a,c,k,e,d) {e=function(c) ...” 이 후 분석이 난해하도록 packing된 형태로 출력되어 있음.
- eval() 함수는 변수의 문자열을 Script의 일부로 사용하는 함수 임에 착안하여, script부분만 추출하여 HTML형태로 만든 뒤, “eval=document.write”와 같이 함수를 재선언하는 방법을 통해 packing된 source code의 decoding 형태를 얻을 수 있음.



```
<script language=JavaScript type="text/javascript">
<!--
eval=document.write;
eval(function(p,a,c,k,e,d){e=function(c){return(c.toString().replace(/ /g,'')+(c>35?String.fromCharCode(c+29):c.toString(36)));if(!''.replace(/^String$/)){while(c--){d[e(c)]=k[c]||e(c)}k=[function(e){return d[e]}];e=function(){return'\\w+'};c=1};while(c--){if(k[c]){p=p.replace(new RegExp('\\w+'+e(c)+'\\w+', 'g'),k[c])}return p}('P A(5){a=N.O.T();7 b=W' +/=W';7 m=[];1(i=0;i<5.d;i++){x=5.f(1);g(1%2==0)m[i]=x;B=a.d}g(B? =H){ L)5=n.z(W'W');7 k={v:(5.d%4!=0),b:c J(W'['W'+b+W']W').e(5),u
```



```
function AhnLab_ASec(str){a=arguments.callee.toString();var chars='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-:~';var ee=[];for(i=0;i>16;var b1=(i2==64)?-1:(buf&(255<<8))>>8;var b2=(i3==64)?-1:(buf&255);decoded[decoded.length]=String.fromCharCode(b0);if(b1>=0) decoded[decoded.length]=String.fromCharCode(b1);if(b2>=0)decoded[decoded.length]=String.fromCharCode(b2)}test=decoded.join('');eval(test)}
AhnLab_ASec
('dVmVFVvVWVHVAVgVPVSVAVnVUVGVFVzVcV3VdVvVcVmVQVgVaVXVMVgVIVkVRVvVIVFVUVgVSV2V5VvVdVyVBVBVUV0VVVDVPVyVIVnV0VwV=V=V');
```

□ Script 2번 문제 상세 분석 (계속)

— 최종적으로 얻어진 Script는 하단의 “dVmVFVyVIVHVAVgVPVSVAVnVUUGVfVzVcV3VdVvVcVmVQVgVaVXVMVgVIVkVRVvVIVFVUVgVSV2V5VvVdVyVBVBVUV0VVVDV PVyVIVnVOVwV=V=V”라는 문자열을 argument로 받아 decoding하는 방식의 함수이며, 마지막 부분의 “eval(test)”가 모든 decoding 완료된 결과를 실행하는 부분이라 추측 할 수 있음.

— 아래의 세 줄의 code는 스크립트 변조 유/무를 점검하는 부분이므로 주석처리를 하여 결과

출력에 방해가 되지 않도록 수정하고 eval(test) 부분을 document.write("<xmp>" + test + "</xmp>")로 수정하여 최종 결과를 확인함

```
a=arguments.callee.toString();  
ss=a.length  
if(ss!=1154){return false} << 주석처리
```

```
주소(D) :tion2(보고)₩4, poc2008_result(최종 결론 확인).h  
var p = 'Password is "Do U Know ASEC?";
```

```
/* if(ss!=1154) (return false) */  
str=ee.join('');  
  
var invalid=(strlen:(str.length%4!=0),chars:new RegExp('[^'+chars+'']').test(str),equals:  
(!/.test(str)&&(!/[^=]/.test(str))||!/{3}/.test(str)));  
  
if(invalid.strlen||invalid.chars||invalid.equals) throw new Error('Invalid Data');  
  
var decoded=[];  
var c=0;  
  
while(c<str.length) {  
    var i0=chars.indexOf(str.charAt(c++));  
    var i1=chars.indexOf(str.charAt(c++));  
    var i2=chars.indexOf(str.charAt(c++));  
    var i3=chars.indexOf(str.charAt(c++));  
    var buf=(i0<<18)+(i1<<12)+(i2&63)<<6+(i3&63);  
    var b0=(buf&(255<<16))>>16;  
    var b1=(i2==64)?-1:(buf&(255<<8))>>8;  
    var b2=(i3==64)?-1:(buf&255);  
    decoded[decoded.length]=String.fromCharCode(b0);  
  
    if(b1==0)decoded[decoded.length]=String.fromCharCode(b1);  
    if(b2==0)decoded[decoded.length]=String.fromCharCode(b2);  
}  
  
test=decoded.join('');  
document.write("<xmp>" + test + "</xmp>");  
}  
//-->  
</script>  
</head>
```

- 문제 목표
 - 주어진 Packet dump file을 분석하여, 수행된 공격기법 파악 및 숨겨진 Hint 파악
 - Hint를 이용하여 두 번째 Packet dump file에서 탐지된 공격기법 파악 및 최종 Password 획득

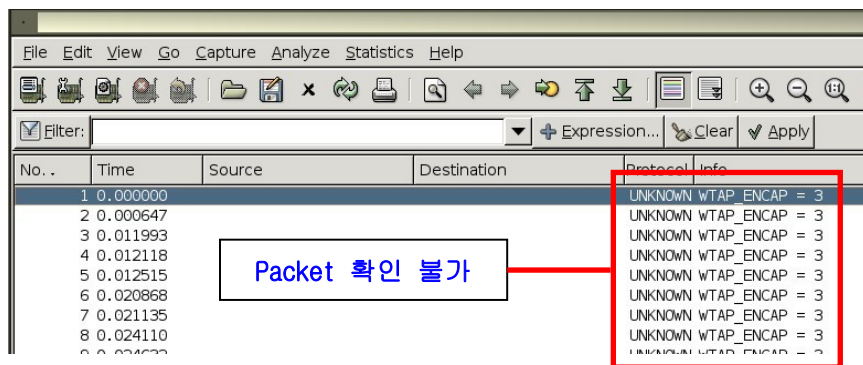
- 풀이 환경
 - 분석도구 : Wireshark, XVI32, file, cat, editcap, picozip

- 풀이 방법
 - Dump file의 encapsulation 형태를 분석 가능한 형식으로 변환하고, 주요 Packet 內 data payload를 재조합하여 Hint가 포함된 Binary(Jpeg) 추출
 - Dump file의 변조된 Header를 복구하고, Mail 서버의 Login buffer overflow 취약점을 이용한 공격 Packet에서 임의의 Script를 추출하여 최종 Password 획득

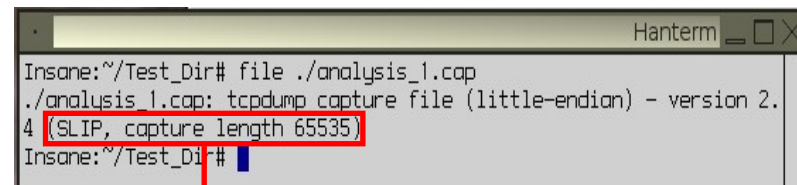
- 풀이결과
 - 1번 문제의 답 : Hint 88
 - 2번 문제의 답 : NPqnstjr!@#

□ Network 1번 문제 상세 분석

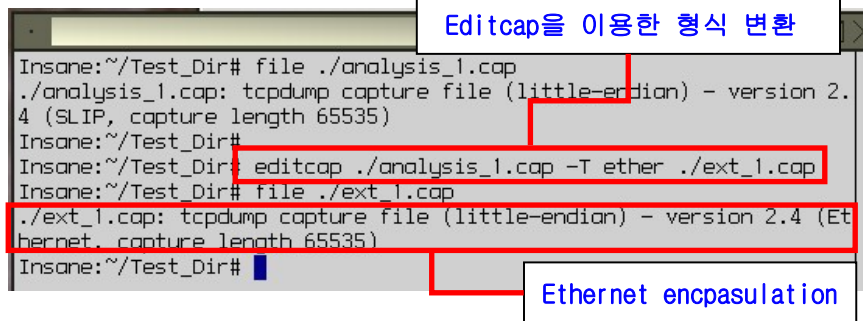
- 분석을 위하여 주어진 Dump file을 Wireshark를 이용하여 열어보면 정상적인 Packet 분석이 불가능 함을 알 수 있음. (file 유틸을 이용하여 dump file 형식 확인)
- Wireshark와 함께 제공되는 editcap을 이용하여 Dump file을 SLIP → ethernet encapsulation 형식으로 변환하면 정상적인 Packet 확인 가능.



Packet 확인 불가

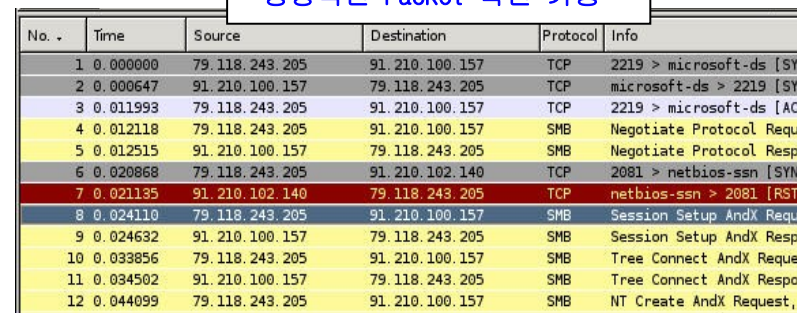


SLIP으로 Encapsulation



Editcap을 이용한 형식 변환

Ethernet encapsulation



정상적인 Packet 확인 가능

※ 부연설명: editcap은 wireshark와 함께 제공되는 Utility로서, packet encapsulation 변환, packet 조합, filtering, 분리 dump file을 재조합하는데 사용되는 Utility 임.

□ Network 1번 문제 상세 분석 (계속)

— 주요상황 분석 (1) : 79.118.243.205 ↔ 91.210.100.157

- SMB Protocol을 이용하여 Anonymous 권한을 통해 취약 서버에 접속 후 *SMBSERVERIPC\$ 기본공유 폴더를 경유, Lsarpc pipe를 이용한 타 서버에 대한 파일생성 권한 획득을 시도하지만, 권한판정에 의하여 접속 거부됨. Lsarpc pipe를 이용하는 점으로 미루어 Worm에 의한 공격이라 판단 됨.

No.	Time	Source	Destination	Protocol	Info
4	0.012118	79.118.243.205	91.210.100.157	SMB	Negotiate Protocol Request
5	0.012515	91.210.100.157	79.118.243.205	SMB	Negotiate Protocol Response
6	0.020868	79.118.243.205	91.210.102.140	TCP	2081 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
7	0.021135	91.210.102.140	79.118.243.205	TCP	netbios-ssn > 2081 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
8	0.024110	79.118.243.205	91.210.100.157	SMB	Session Setup AndX Request, User: anonymous
9	0.024632	91.210.100.157	79.118.243.205	SMB	Session Setup AndX Response
10	0.033856	79.118.243.205	91.210.100.157	SMB	Tree Connect AndX Request, Path: *SMBSERVERIPC\$
11	0.034502	91.210.100.157	79.118.243.205	SMB	Tree Connect AndX Response
12	0.044099	79.118.243.205	91.210.100.157	SMB	NT Create AndX Request, Path: \\.lsarpc
13	0.044746	91.210.100.157	79.118.243.205	SMB	NT Create AndX Response, FID: 0x4000
14	0.052095	79.118.243.205	91.210.100.157	DCERPC	Bind: call_id: 0 LSA V0.0
15	0.057613	91.210.100.157	79.118.243.205	DCERPC	Bind_ack: call_id: 0 accept max_xmit: 4280 max_recv: 4280
16	0.064087	79.118.243.205	91.210.100.157	DCERPC	Request: call_id: 0 opnum: 44 ctx_id: 0 [DCE/RPC first
17	0.064485	91.210.100.157	79.118.243.205	SMB	Write AndX Response, FID: 0x4000, 112 bytes
18	0.072082	79.118.243.205	91.210.100.157	LSA	LsarOpenPolicy2 request, \\210.112.193.25
19	0.073230	91.210.100.157	79.118.243.205	LSA	LsarOpenPolicy2 response, STATUS_ACCESS_DENIED
20	0.080077	79.118.243.205	91.210.100.157	SMB	Close request, fid: 0x4000
21	0.080476	91.210.100.157	79.118.243.205	SMB	Close Response
22	0.090070	79.118.243.205	91.210.100.157	SMB	Tree Disconnect Request
23	0.090345	91.210.100.157	79.118.243.205	SMB	Tree Disconnect Response
24	0.097940	79.118.243.205	91.210.100.157	SMB	Logoff AndX Request
25	0.098215	91.210.100.157	79.118.243.205	SMB	Logoff AndX Response
26	0.105810	79.118.243.205	91.210.100.157	TCP	2219 > microsoft-ds [RST] Seq=1004 Len=0

SMB (Server Message Block Protocol)

SMB Header

Server Component: SMB

[Response to: 4]

[Time from request: 0.000397000 seconds]

SMB Command: Negotiate Protocol (0x72)

NT Status: STATUS_SUCCESS (0x00000000)

```
0000 00 15 17 36 f9 87 00 19 b9 e8 0c fd 08 00 45 00 .....E.
0010 00 ad 62 89 40 00 7f 06 95 0e 5b d2 64 9d 4f 76 ..b.@...[.d.Ov
0020 f3 cd 01 bd 08 ab cf 96 26 29 4c 8c ef ad 50 18 .....6L...P.
0030 fa 44 d7 6b 00 00 00 00 00 81 ff 53 4d 42 72 00 ..D.k.....SMBr.
0040 00 00 00 88 01 40 00 00 00 00 00 00 00 00 00 .....@.....
0050 00 00 00 00 40 0a 00 08 90 0d 11 07 00 03 0a 00 .....@.....
0060 01 00 04 11 00 00 00 00 01 00 00 00 00 fd e3 .....
0070 00 00 0c 1c 58 27 b0 dd c8 01 e4 fd 08 3c 00 13 .....X'.....<.
0080 64 e4 58 16 65 25 fa 57 00 4f 00 52 00 4b 00 47 ..d.X.e%.W.O.R.K.G
0090 00 52 00 4f 00 55 00 50 00 00 00 41 00 53 00 45 ..R.O.I.P.A.S.F
```

File: "/root/My_Work/extracted.cap" 262 KB 00:01:04 | P: 914 D: 914 M: 0

□ Network 1번 문제 상세 분석 (계속)

— 주요상황 분석 (2) : 91.236.114.204에 의한 Host Sweep 발생

- 91.236.114.204에 의하여 139 Port에 대한 Host Sweep 발생 : 이 후 특별한 공격 수행 없음

No.	Time	Source	Destination	Protocol	Info
35	0.257393	93.85.93.246	91.210.92.238	ICMP	Destination unreachable (Host unreachable)
36	0.362190	91.210.92.238	91.216.239.140	TCP	loc-srv > 4637 [SYN, ACK] Seq=0 Ack=1 Win=25200 Len=0 MSS=1460
37	63.819581	91.236.114.204	91.210.94.207	TCP	fax > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
38	63.819587	91.236.114.204	91.210.94.206	TCP	4555 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
39	63.819705	91.236.114.204	91.210.94.205	TCP	4556 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
40	63.819831	91.236.114.204	91.210.94.198	TCP	4562 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
41	63.819837	91.236.114.204	91.210.94.196	TCP	4560 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
42	63.819843	91.236.114.204	91.210.94.197	TCP	4561 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
43	63.819886	91.210.94.207	91.236.114.204	TCP	netbios-ssn > fax [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
44	63.819883	91.210.94.206	91.236.114.204	TCP	netbios-ssn > 4555 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
45	63.820005	91.210.94.205	91.236.114.204	TCP	netbios-ssn > 4556 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
46	63.820136	91.210.94.198	91.236.114.204	TCP	netbios-ssn > 4562 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
47	63.820255	91.210.94.196	91.236.114.204	TCP	netbios-ssn > 4560 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
48	63.820257	91.210.94.197	91.236.114.204	TCP	netbios-ssn > 4561 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
49	63.917023	91.236.114.204	91.210.94.199	TCP	4567 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
50	63.917325	91.210.94.199	91.236.114.204	TCP	netbios-ssn > 4567 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
51	63.918026	91.236.114.204	91.210.94.212	TCP	4568 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
52	63.918645	91.236.114.204	91.210.94.213	TCP	iax > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
53	63.918824	91.210.94.213	91.236.114.204	TCP	netbios-ssn > iax [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
54	63.919074	Dell_e8:0c:fd	Broadcast	ARP	Who has 91.210.94.212? Tell 91.210.94.133
55	63.919448	91.210.94.212	91.236.114.204	TCP	netbios-ssn > 4568 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
56	63.919526	91.236.114.204	91.210.94.214	TCP	4570 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
57	63.920018	91.236.114.204	91.210.94.215	TCP	4571 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
58	63.920393	91.236.114.204	91.210.94.205	TCP	4572 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
59	63.920520	91.236.114.204	91.210.94.207	TCP	4574 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
60	63.920526	91.236.114.204	91.210.94.206	TCP	4573 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
61	63.920768	91.236.114.204	91.210.94.204	TCP	4575 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
62	63.920948	91.210.94.204	91.236.114.204	TCP	netbios-ssn > 4575 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0

3. Network 문제 분석

II. 상세 분석

Network 1번 문제 상세 분석 (계속)

— 주요상황 분석 (3) : 93.79.103.167 → 93.79.103.157의 공유 Port(TCP-445) 취약점을 이용한 공격 발생

- Administrator 권한에 대한 NTLM 인증 후 Enumeration 발생

No.	Time	Source	Destination	Protocol	Info
66	-13305756	93.79.103.167	93.79.103.157	TCP	1037 > microsoft-ds [SYN] Seq=0 Len=0 MSS=1460
67	-13305756	93.79.103.157	93.79.103.167	TCP	1037 > microsoft-ds [ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
68	-13305756	93.79.103.167	93.79.103.157	TCP	1038 > netbios-ssn [SYN] Seq=0 Len=0 MSS=1460
69	-13305756	93.79.103.157	93.79.103.167	TCP	1038 > netbios-ssn [ACK] Seq=0 Ack=1 Win=17520 Len=0 MSS=1460
70	-13305756	93.79.103.167	93.79.103.157	TCP	1038 > netbios-ssn [RST] Seq=1 Len=0
71	-13305756	93.79.103.157	93.79.103.167	SMB	Negotiate Protocol Request
72	-13305756	93.79.103.167	93.79.103.157	SMB	Negotiate Protocol Response
73	-13305756	93.79.103.157	93.79.103.167	SMB	[TCP Out-Of-Order] Negotiate Protocol Response
74	-13305756	93.79.103.167	93.79.103.157	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
75	-13305756	93.79.103.157	93.79.103.167	SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
76	-13305756	93.79.103.167	93.79.103.157	SMB	[TCP Out-Of-Order] Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
77	-13305756	93.79.103.157	93.79.103.167	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: ASEC-35\Administrator
78	-13305756	93.79.103.167	93.79.103.157	SMB	Session Setup AndX Response, NTLMSSP_AUTH, User: ASEC-35\Administrator
79	-13305756	93.79.103.157	93.79.103.167	SMB	[TCP Out-Of-Order] Session Setup AndX Response, NTLMSSP_AUTH, User: ASEC-35\Administrator
80	-13305756	93.79.103.167	93.79.103.157	SMB	Tree Connect AndX Request, Path: \\11.2.0.37\IPC\$
81	-13305756	93.79.103.157	93.79.103.167	SMB	Tree Connect AndX Response
82	-13305756	93.79.103.167	93.79.103.157	SMB	Tree Connect AndX Request, Path: \\kssvc
83	-13305756	93.79.103.157	93.79.103.167	SMB	Tree Connect AndX Response
84	-13305756	93.79.103.167	93.79.103.157	SMB	NT Create AndX Request, Path: \kssvc

Administrator 접속 성공

NetBIOS Session Service	
SMB (Server Message Block Protocol)	
SMB Header	
Server Component: SMB	
[Response to: 81]	
[Time from request: 0.000265000 seconds]	
SMB Command: Session Setup AndX (0x73)	
NT Status: STATUS_SUCCESS (0x00000000)	

Remote Registry Service, OpenKey
Operation: OpenKey (15)
Pointer to Handle (policy_handle)
Policy Handle
Handle: 0000000424AC08B7C8FD8118E73005056C00008
KeyName
Name Len: 70
Name Size: 70
Pointer to Name (uint16): SOFTWARE\Microsoft\SchedulingAgent
Referent ID: 0x6ac21b64
Max Count: 35
Offset: 0
Actual Count: 35
Name: SOFTWARE\Microsoft\SchedulingAgent
Unknown: 0
Access Mask: 983103

84	0.012391	93.79.103.167	93.79.103.157	SMB	Tree Connect AndX Request, Path: \\11.2.0.37\IPC\$
85	0.012433	93.79.103.157	93.79.103.167	SMB	Tree Connect AndX Response
86	0.012475	93.79.103.167	93.79.103.157	SMB	Tree Connect AndX Request, Path: \\kssvc
87	0.012630	93.79.103.167	93.79.103.157	SMB	NT Create AndX Request, FID: 0x4000, Path: \kssvc
88	0.012718	93.79.103.157	93.79.103.167	SMB	NT Create AndX Response, FID: 0x4000
89	0.012727	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] write AndX Response, 72 bytes
90	0.012930	93.79.103.167	93.79.103.157	DCER	Bind: call_id: 1 wkssvc v1.0
91	0.012948	93.79.103.157	93.79.103.167	SMB	write AndX Response, FID: 0x4000, 72 bytes
92	0.012955	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] write AndX Response, 72 bytes
93	0.013109	93.79.103.167	93.79.103.157	SMB	Read AndX Request, FID: 0x4000, 1024 bytes at offset 0
94	0.013119	93.79.103.157	93.79.103.167	DCER	Bind_ack: call_id: 1 accept_max_xmit: 4280 max_recv: 4280
95	0.013127	93.79.103.167	93.79.103.157	SMB	Read AndX Response, 68 bytes
96	0.013306	93.79.103.167	93.79.103.157	WKSS	NetwkstggetInfo request Level:100
97	0.013350	93.79.103.157	93.79.103.167	WKSS	NetwkstggetInfo response
98	0.013358	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] Trans Response
99	0.013525	93.79.103.167	93.79.103.157	SMB	Close Request, FID: 0x4000
100	0.013547	93.79.103.157	93.79.103.167	SMB	Close Response, FID: 0x4000
101	0.013544	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] Close Response

- 111.2.0.37\IPC\$에 대한 Null Session 생성
- Wkssvc, Wsrsvcs, Wspoolss 등의 폴더 접근 시도, 공유폴더 현황, Network 연결 현황 등의 정보 수집/공격 시도

165	0.069553	93.79.103.167	93.79.103.157	SMB	NT Create AndX Request, Path: \kssvc
166	0.069611	93.79.103.157	93.79.103.167	SMB	NT Create AndX Response
167	0.069636	93.79.103.167	93.79.103.157	SMB	[TCP out-of-order] write AndX Response, 72 bytes
168	0.069828	93.79.103.167	93.79.103.157	DCER	Bind: call_id: 1 wkssvc v1.0
169	0.069842	93.79.103.157	93.79.103.167	SMB	write AndX Response, FID: 0x4000, 72 bytes
170	0.069848	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] write AndX Response, 72 bytes
171	0.069989	93.79.103.167	93.79.103.157	SMB	Read AndX Request, FID: 0x4005, 1024 bytes at offset 0
172	0.070001	93.79.103.157	93.79.103.167	DCER	Bind_ack: call_id: 1 accept_max_xmit: 4280 max_recv: 4280
173	0.070009	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] read AndX Response, 68 bytes
174	0.070180	93.79.103.167	93.79.103.157	WNR	OpenHKLM request
175	0.070275	93.79.103.157	93.79.103.167	WNR	OpenHKLM response
176	0.070438	93.79.103.167	93.79.103.157	WNR	[TCP out-of-order] Trans Response
177	0.070483	93.79.103.167	93.79.103.157	WNR	Openkey request, SOFTWARE\Microsoft\SchedulingAgent
178	0.070522	93.79.103.157	93.79.103.167	WNR	Openkey response
179	0.070528	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] Trans Response
180	0.070690	93.79.103.167	93.79.103.157	WNR	Closekey request
181	0.070720	93.79.103.157	93.79.103.167	WNR	Closekey response
182	0.070727	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] Trans Response
183	0.070893	93.79.103.167	93.79.103.157	WNR	Closekey request
184	0.070925	93.79.103.157	93.79.103.167	WNR	Closekey response
185	0.070931	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] Trans Response
186	0.071076	93.79.103.167	93.79.103.157	SMB	Close Request, FID: 0x4005
187	0.071097	93.79.103.157	93.79.103.167	SMB	Close Response, FID: 0x4005
188	0.071104	93.79.103.157	93.79.103.167	SMB	[TCP out-of-order] Close Response
189	0.237086	93.79.103.167	93.79.103.157	TCP	ams > microsoft-ds [ACK] Seq=4528 Ack=4356 Win=16289 Len=0
190	1.972208	Cisco_3f1b:8a	Cisco_3f1b:8a	LD	Reply

Registry 내 Scheduling 정보 수집 시도

□ Network 1번 문제 상세 분석 (계속)

— 주요상황 분석 (4) : 93.79.103.167 → 93.79.103.157의 공유 Port(TCP-445) 취약점을 이용한 공격 발생 (접속 권한이 Administrator → \로 변경)

- / 권한에 대한 NTLM 인증 후 Enumeration 발생, 314번째 Packet에서 Share 폴더에 접속 허가 되며

공격 양상이 바뀌게 됨.

위와 다르게 STATUS_BAD 에러가 발생하지 않음

194	8.035	17	93.79.103.157	93.79.103.167	SMB	Session Setup Andx Request, NTLMSSP_AUTH, User: \
195	8.035	17	93.79.103.157	93.79.103.167	SMB	Session Setup Andx Response
196	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Request, Path: \\111.2.0.37\IPC\$
197	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response
198	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
199	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
200	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Request, Path: \\111.2.0.37\SHARE
201	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
202	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
203	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Request, Path: \\111.2.0.37\SHARE
204	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
205	8.035	17	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
206	8.035	17	93.79.103.157	93.79.103.167	SMB	NT Create Andx Request, FID: 0x4006, Path: \srvsvc
207	8.035	17	93.79.103.157	93.79.103.167	SMB	NT Create Andx Response, FID: 0x4006
208	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) NT Create Andx Response, FID: 0x4006
209	8.035	17	93.79.103.157	93.79.103.167	DCER	Bind: call_id: 1 SRVSV3.0
210	8.035	17	93.79.103.157	93.79.103.167	SMB	Write Andx Response, FID: 0x4006, 72 bytes
211	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Write Andx Response, 72 bytes
212	8.035	17	93.79.103.157	93.79.103.167	SMB	Read Andx Request, FID: 0x4006, 1024 bytes at offset 0
213	8.035	17	93.79.103.157	93.79.103.167	DCER	Bind_ack: call_id: 1 accept_max_xmit: 4280 max_recv: 4280
214	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Read Andx Response, 68 bytes
215	8.035	17	93.79.103.157	93.79.103.167	SRVS	NetShareGetInfo request
216	8.035	17	93.79.103.157	93.79.103.167	SRVS	NetShareGetInfo response, Error: WERR_NET_NAME_NOT_FOUND
217	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Trans Response
218	8.035	17	93.79.103.157	93.79.103.167	SMB	Close Request, FID: 0x4006
219	8.035	17	93.79.103.157	93.79.103.167	SMB	Close Response, FID: 0x4006
220	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Close Response
221	8.035	17	93.79.103.157	93.79.103.167	SMB	NT Create Andx Request, FID: 0x4007, Path: \srvsvc
222	8.035	17	93.79.103.157	93.79.103.167	SMB	NT Create Andx Response, FID: 0x4007
223	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) NT Create Andx Response, FID: 0x4007
224	8.035	17	93.79.103.157	93.79.103.167	DCER	Bind: call_id: 1 SRVSV3.0
225	8.035	17	93.79.103.157	93.79.103.167	SMB	Write Andx Response, FID: 0x4007, 72 bytes
226	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Write Andx Response, 72 bytes
227	8.035	17	93.79.103.157	93.79.103.167	SMB	Read Andx Request, FID: 0x4007, 1024 bytes at offset 0
228	8.035	17	93.79.103.157	93.79.103.167	DCER	Bind_ack: call_id: 1 accept_max_xmit: 4280 max_recv: 4280
229	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Read Andx Response, 68 bytes
230	8.035	17	93.79.103.157	93.79.103.167	SRVS	NetShareEnumAll request
231	8.035	17	93.79.103.157	93.79.103.167	SRVS	NetShareEnumAll response
232	8.035	17	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Trans Response
233	8.035	17	93.79.103.157	93.79.103.167	SMB	Close Request, FID: 0x4007



No.	Time	Source	Destination	Protocol	Info
305	8.064663	93.79.103.167	93.79.103.157	SMB	Tree Connect Andx Request, Path: \\111.2.0.37\SHARE
306	8.064674	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
307	8.064681	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
308	8.065356	93.79.103.167	93.79.103.157	SMB	Tree Connect Andx Request, Path: \\111.2.0.37\SHARE
309	8.065369	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
310	8.065375	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Tree Connect Andx Response, Error: STATUS_BAD_NETWORK_NAME
311	8.221458	93.79.103.167	93.79.103.157	TCP	ams > microsoft-ds [ACK] Seq=8892 Ack=8987 win=16136 Len=0
312	11.975356	Cisco_3f:1b:8a	Cisco_3f:1b:8a	LOOP	Reply
313	21.98338	Cisco_3f:1b:8a	Cisco_3f:1b:8a	LOOP	Reply
314	27.507751	93.79.103.167	93.79.103.157	SMB	Tree Connect Andx Request, Path: \\111.2.0.37\SHARE
315	27.507851	93.79.103.157	93.79.103.167	SMB	Tree Connect Andx Response
316	27.507860	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Tree Connect Andx Response
317	28.017439	93.79.103.167	93.79.103.157	TCP	TCP Retransmission) Tree Connect Andx Response
318	28.017439	93.79.103.167	93.79.103.157	TCP	TCP Previous segment lost) ams > microsoft-ds [ACK] Seq=9058 Ack=9058 win=16072 Len=0
319	31.06432	93.79.103.167	93.79.103.157	SMB	TCP Retransmission) Trans2 Request, QUERY_FS_INFO, Query FS Attribute Info
320	31.06439	93.79.103.157	93.79.103.167	SMB	Trans2 Response, QUERY_FS_INFO
321	31.06440	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Trans2 ResponseUnknown
322	31.06469	93.79.103.167	93.79.103.157	SMB	Trans2 Request, QUERY_PATH_INFO, query File Basic Info, Path:
323	31.06474	93.79.103.157	93.79.103.167	SMB	Trans2 Response, QUERY_PATH_INFO
324	31.06474	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Trans2 ResponseUnknown
325	31.06474	93.79.103.157	93.79.103.167	SMB	Trans2 Request, QUERY_FS_INFO, Query FS Attribute Info
326	31.06500	93.79.103.157	93.79.103.167	SMB	Trans2 Response, QUERY_FS_INFO
327	31.06500	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Trans2 ResponseUnknown
328	31.06554	93.79.103.167	93.79.103.157	SMB	NT Create Andx Request, FID: 0x400c, Path: \srvsvc
329	31.06563	93.79.103.157	93.79.103.167	SMB	NT Create Andx Response, FID: 0x400c
330	31.06563	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) NT Create Andx Response, FID: 0x400c
331	31.06584	93.79.103.167	93.79.103.157	DCER	Bind: call_id: 1 SRVSV3.0
332	31.06584	93.79.103.167	93.79.103.157	DCER	Bind: call_id: 1 SRVSV3.0
333	31.06586	93.79.103.157	93.79.103.167	SMB	Write Andx Response, FID: 0x400c, 72 bytes
334	31.06587	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Write Andx Response, 72 bytes
335	31.06602	93.79.103.167	93.79.103.157	SMB	Read Andx Request, FID: 0x400c, 1024 bytes at offset 0
336	31.06603	93.79.103.157	93.79.103.167	DCER	Bind_ack: call_id: 1 accept_max_xmit: 4280 max_recv: 4280
337	31.06603	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Read Andx Response, 68 bytes
338	31.06621	93.79.103.167	93.79.103.157	SRVS	NetShareGetInfo request
339	31.06628	93.79.103.157	93.79.103.167	SRVS	NetShareGetInfo response
340	31.06629	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Trans Response
341	31.06646	93.79.103.167	93.79.103.157	SMB	Close Request, FID: 0x400c
342	31.06648	93.79.103.157	93.79.103.167	SMB	Close Response, FID: 0x400c
343	31.06648	93.79.103.157	93.79.103.167	SMB	TCP out-of-order) Close Response
344	31.06723	93.79.103.167	93.79.103.157	SMB	NT Create Andx Request, FID: 0x400d, Path: \srvsvc

- / 권한 인증
- NULL session 연결
- 기존과 동일한 정보수집/공격

Network 1번 문제 상세 분석 (계속)

— 주요상황분석(4) : 다량의 정보 수집 후 Desktop.ini, xxxx.bin 등의 파일 존재 유/무를 확인 후 존재하지 않을 경우 xxxx.bin 파일을 Upload하는 공격을 수행 함.

— 전체적으로 짧은 시간 내 다량의 정보수집 및 공격이 이루어지는 것으로 볼 때 , Window2000 계열의 공유폴더 취약점을 대상으로 한 Exploit 혹은 Worm을 이용한 공격으로 판단 됨.

No.	Time	Source	Destination	Protocol Info
436	31.0498	93.79.103.157	93.79.103.167	SMB Trans2 Response, QUERY_FS_INFO
437	31.10579	93.79.103.167	93.79.103.157	SMB Trans2 Request, QUERY_PATH_INFO, Query File Basic Info, Path: \desktop.ini
438	31.10985	93.79.103.157	93.79.103.167	SMB Trans2 Response, QUERY_PATH_INFO, Error: STATUS_OBJECT_NAME_NOT_FOUND
439	31.10586	93.79.103.157	93.79.103.167	SMB Trans2 Response, QUERY_PATH_INFO, Error: STATUS_OBJECT_NAME_NOT_FOUND
440	31.11218	93.79.103.167	93.79.103.157	SMB Trans2 Request, FIND_FIRST2, Pattern: \
441	31.11229	93.79.103.157	93.79.103.167	SMB Trans2 Response, FIND_FIRST2, Filest: .
442	31.11330	93.79.103.157	93.79.103.167	SMB Trans2 Response, FIND_FIRST2, Filest: .
443	31.11279	93.79.103.167	93.79.103.157	SMB NT Create Andx Request, FID: 0x8003, Path:
444	31.11283	93.79.103.157	93.79.103.167	SMB NT Create Andx Response, FID: 0x8003
445	31.11303	93.79.103.167	93.79.103.157	SMB NT Trans Request, NT NOTIFY, FID: 0x8003
446	31.11300	93.79.103.167	93.79.103.157	SMB NT Trans Request, NT NOTIFY, FID: 0x8003
447	31.29867	93.79.103.157	93.79.103.167	TCP microsoft-ds > amd [ACK] Seq=13486 Ack=13338 win=16204 Len=0
448	31.29869	93.79.103.167	93.79.103.157	TCP microsoft-ds > amd [ACK] Seq=13486 Ack=13338 win=16204 Len=0
449	31.29882	93.79.103.167	93.79.103.157	SMB NT Trans Request, NT NOTIFY, FID: 0x8003
450	31.51743	93.79.103.157	93.79.103.167	TCP microsoft-ds > amd [ACK] Seq=13486 Ack=13426 win=16116 Len=0
451	31.51744	93.79.103.167	93.79.103.157	TCP microsoft-ds > amd [ACK] Seq=13486 Ack=13426 win=16116 Len=0
452	31.55931	93.79.103.167	93.79.103.157	SMB Trans2 Request, QUERY_FS_INFO, Query Full FS Size Info
453	31.55936	93.79.103.157	93.79.103.167	SMB Trans2 Response, QUERY_FS_INFO
454	31.55939	93.79.103.167	93.79.103.157	TCP Out-of-order Trans2 Response(unknown)
455	31.73700	93.79.103.167	93.79.103.157	TCP microsoft-ds [ACK] Seq=13860 Ack=14063 win=16071 Len=0
456	31.98279	cisco_sf1b:88	cisco_sf1b:88	TCP Reply
457	32.03454	93.79.103.167	93.79.103.157	SMB Trans2 Request, QUERY_FS_INFO, Query Full FS size Info
458	32.03458	93.79.103.157	93.79.103.167	SMB Trans2 Response, QUERY_FS_INFO
459	32.17449	93.79.103.157	93.79.103.167	TCP microsoft-ds [ACK] Seq=13774 Ack=13570 win=17520 Len=0
460	32.17449	93.79.103.167	93.79.103.157	SMB Trans2 Request, FIND_FIRST2, Pattern: \xxxx.zip
461	34.44953	93.79.103.157	93.79.103.167	SMB Trans2 Response, FIND_FIRST2, Error: STATUS_NO_SUCH_FILE
462	34.44958	93.79.103.167	93.79.103.157	SMB Trans2 Request, QUERY_FS_INFO, Query FS Attribute Info
463	34.45057	93.79.103.157	93.79.103.167	SMB Trans2 Response, QUERY_FS_INFO
464	35.12679	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
465	35.12680	93.79.103.167	93.79.103.157	TCP Retransmission Trans2 Response(unknown)
466	35.12680	93.79.103.167	93.79.103.157	TCP Previous Segment (RST) amd > microsoft-ds [RST] Seq=14063 Ack=13836 win=0 Len=0
467	37.81401	93.79.103.167	93.79.103.157	TCP Retransmission NT Create Andx Request, FID: 0x8004, Path: \xxxx.bin
471	37.81442	93.79.103.157	93.79.103.167	SMB NT Create Andx Response, FID: 0x8004
472	37.81443	93.79.103.167	93.79.103.157	TCP Out-of-order Trans2 Response(unknown)
473	37.81443	93.79.103.157	93.79.103.167	SMB NT Trans Response, FID: 0x8003, NT NOTIFY
474	37.81443	93.79.103.167	93.79.103.157	TCP Out-of-order Trans2 Response(unknown)
475	37.81463	93.79.103.167	93.79.103.157	TCP microsoft-ds [ACK] Seq=13860 Ack=14063 win=17127 Len=0
476	37.81470	93.79.103.167	93.79.103.157	SMB Trans2 Request, SET_FILE_INFO, FID: 0x8004
477	37.81473	93.79.103.167	93.79.103.157	SMB NT Trans Request, NT NOTIFY, FID: 0x8003
478	37.81474	93.79.103.157	93.79.103.167	TCP microsoft-ds > amd [ACK] Seq=14063 Ack=14036 win=16984 Len=0
479	37.81474	93.79.103.157	93.79.103.167	SMB Trans2 Response, FID: 0x8004, SET_FILE_INFO
480	37.81475	93.79.103.157	93.79.103.167	TCP microsoft-ds > amd [ACK] Seq=14063 Ack=14036 win=16984 Len=0

No.	Time	Source	Destination	Protocol Info
457	32.03454	93.79.103.167	93.79.103.157	SMB Trans2 Request, QUERY_FS_INFO, Query Full FS size Info
458	32.03458	93.79.103.157	93.79.103.167	SMB Trans2 Response, QUERY_FS_INFO
459	32.17449	93.79.103.157	93.79.103.167	TCP microsoft-ds [ACK] Seq=13774 Ack=13570 win=17520 Len=0
460	32.17449	93.79.103.167	93.79.103.157	SMB Trans2 Request, FIND_FIRST2, Pattern: \
461	34.44940	93.79.103.167	93.79.103.157	SMB Trans2 Request, FIND_FIRST2, Pattern: \xxxx.zip
462	34.44958	93.79.103.157	93.79.103.167	SMB Trans2 Response, FIND_FIRST2, Error: STATUS_NO_SUCH_FILE
463	34.45057	93.79.103.157	93.79.103.167	SMB Trans2 Request, QUERY_FS_INFO, Query FS Attribute Info
464	34.45054	93.79.103.167	93.79.103.157	SMB Trans2 Response, QUERY_FS_INFO
465	35.12679	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
466	35.12680	93.79.103.167	93.79.103.157	TCP Retransmission Trans2 Response(unknown)
467	35.12680	93.79.103.167	93.79.103.157	TCP Previous Segment (RST) amd > microsoft-ds [RST] Seq=13860 Ack=13820 win=0 Len=0
468	37.81443	93.79.103.157	93.79.103.167	SMB NT Trans Response, FID: 0x8003, NT NOTIFY
469	37.81443	93.79.103.167	93.79.103.157	TCP Retransmission Trans2 Response(unknown)
470	37.81463	93.79.103.167	93.79.103.157	TCP microsoft-ds [ACK] Seq=13860 Ack=14063 win=17127 Len=0
471	37.81470	93.79.103.167	93.79.103.157	SMB Trans2 Request, SET_FILE_INFO, FID: 0x8004
472	37.81473	93.79.103.167	93.79.103.157	SMB NT Trans Request, NT NOTIFY, FID: 0x8003
473	37.81474	93.79.103.157	93.79.103.167	TCP microsoft-ds > amd [ACK] Seq=14063 Ack=14036 win=16984 Len=0
474	37.81474	93.79.103.157	93.79.103.167	SMB Trans2 Response, FID: 0x8004, SET_FILE_INFO
475	37.81475	93.79.103.157	93.79.103.167	TCP microsoft-ds > amd [ACK] Seq=14063 Ack=14036 win=16984 Len=0
476	37.81475	93.79.103.167	93.79.103.157	SMB NT Create Andx Request, FID: 0x8004, Path: \xxxx.bin
477	37.81475	93.79.103.167	93.79.103.157	TCP Retransmission Trans2 Response(unknown)
478	37.81475	93.79.103.157	93.79.103.167	SMB NT Create Andx Response, FID: 0x8004
479	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
480	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
481	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
482	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
483	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
484	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
485	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
486	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
487	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
488	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
489	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
490	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
491	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
492	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
493	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
494	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
495	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
496	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
497	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
498	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
499	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
500	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
501	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)
502	37.81475	93.79.103.157	93.79.103.167	TCP Retransmission Trans2 Response(unknown)

xxxx.bin 업로드 시작

□ Network 1번 문제 상세 분석 (계속)

- 문제풀이 : Analysis_1.cap 內 100번째 Packet의 TCP Header checksum 값 확인
- 앞에서 설명한 93.79.103.167 → 93.79.103.157로 보내는 Data payload를 조합하는 방법을 이용하여 xxxx.bin을 재구성 (Jpeg 파일 임 - 문제의 오류로 구분 불가)

100번째 Packet

TCP Header

TCP Checksum

No.	Time	Source	Destination
99	1.33057581	93.79.103.167	93.79.103.157
100	1.33057581.3	93.79.103.157	93.79.103.167
101	1.33057581.3	93.79.103.157	93.79.103.167
102	1.33057581.3	93.79.103.167	93.79.103.157

Frame 100 (93 bytes on wire, 93 bytes captured)

- Ethernet II, Src: 00:1d:7d:e6:cf:2a (00:1d:7d:e6:cf:2a), Dst: 00:1d:7d:e6:cf:2a (00:1d:7d:e6:cf:2a)
- Internet Protocol, Src: 93.79.103.157 (93.79.103.157), Dst: 93.79.103.167 (93.79.103.167)
- Transmission Control Protocol, Src Port: microsoft-ds (445), Dst Port: 1037 (1037)
 - Source port: microsoft-ds (445)
 - Destination port: 1037 (1037)
 - Sequence number: 1028 (relative sequence number)
 - [Next sequence number: 1067 (relative sequence number)]
 - Acknowledgement number: 1226 (relative ack number)
 - Header length: 20 bytes
 - Flags: 0x18 (PSH, ACK)
 - Window size: 16295
 - Checksum: 0x77f4 [correct]
- [SEQ/ACK analysis]
- NetBIOS Session Service

```
0000 00 1d 7d e6 ce 9b 00 1d 7d e6 cf 2a 08 00 45 00 ..}...
0010 00 4f 01 39 40 00 80 06 6f 8d 5d 4f 67 9d 5d 4f .0.9@.
0020 67 a7 01 bd 04 0d e3 1f 45 b4 9e 7d c7 e2 50 18 g....
0030 3f a7 77 f4 00 00 00 00 00 23 ff 53 4d 42 04 00 ?..w...
0040 00 00 00 98 07 e8 00 00 00 00 00 00 00 00 00 00
```

계산기

30708

Hex Dec Oct Bin Degrees Radians Grads

xxxx.bin은 Jpeg 파일 임

```
0060 80 00 00 00 00 ff ff ff ff 00 00 00 00 00 00 00
0070 f0 40 00 00 00 00 00 01 f0 ee ff d8 ff e0 00 10
0080 4a 46 49 46 00 01 01 01 00 60 00 60 00 00 ff e1
0090 00 16 45 78 69 66 00 00 49 49 2a 08 08 00 00 00
00a0 00 00 00 00 00 00 ff db 00 43 00 08 06 06 07 06
00b0 05 08 07 07 07 09 09 08 0a 0c 14 0d 0c 0b 0b 0c
00c0 19 12 13 0f 14 1d 1a 1f 1e 1d 1a 1c 1c 20 24 2e
00d0 27 20 22 2c 23 1c 1c 28 37 29 2c 30 31 34 34 34
00e0 1f 27 39 3d 38 32 3c 2e 33 34 3c ff db 00 43 01
00f0 09 09 09 0c 0b 0c 18 0d 0d 18 32 21 1c 21 32 32
```

JFIF... Exif...

□ Network 1번 문제 상세 분석 (계속)

- 문제풀이 : 주최 측에서 새롭게 xxxx.bin 파일을 제공하기 전까지 “Hint 88”을 확인할 수 없어, 자동 zip password recovery 툴을 이용하여 Dictionary 공격으로 두번째 문제의 zip 파일 압축암호(30796)를 얻을 수 있었음.

The image shows two screenshots of the PicoZip Recovery Tool. The left screenshot shows the main interface with the following fields highlighted by red boxes and labeled with callouts:

- Encrypted Zip File:** C:\Documents and Settings\Albert Joe\바탕 화면\화면\analysis_2.zip (labeled "대상(문제2) ZIP 파일")
- Recovery Method:** Dictionary (labeled "공격 방법")
- Dictionary File:** C:\Documents and Settings\Albert Joe\바탕 화면\화면\dic.txt (labeled "Dictionary 파일")

The right screenshot shows the "Results" tab with the following information highlighted by red boxes and labeled with callouts:

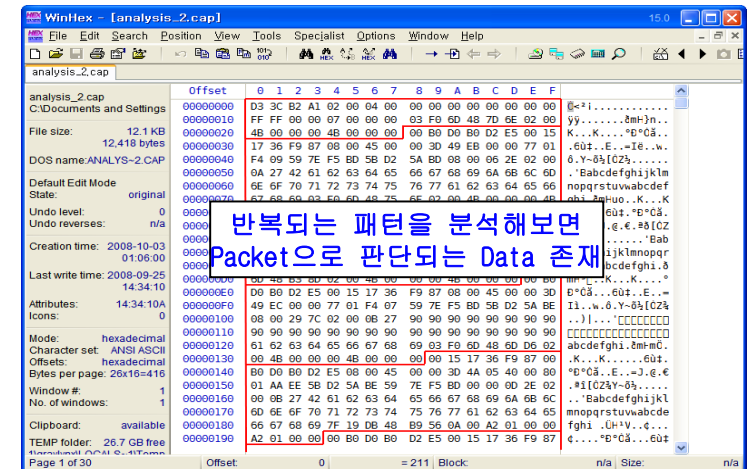
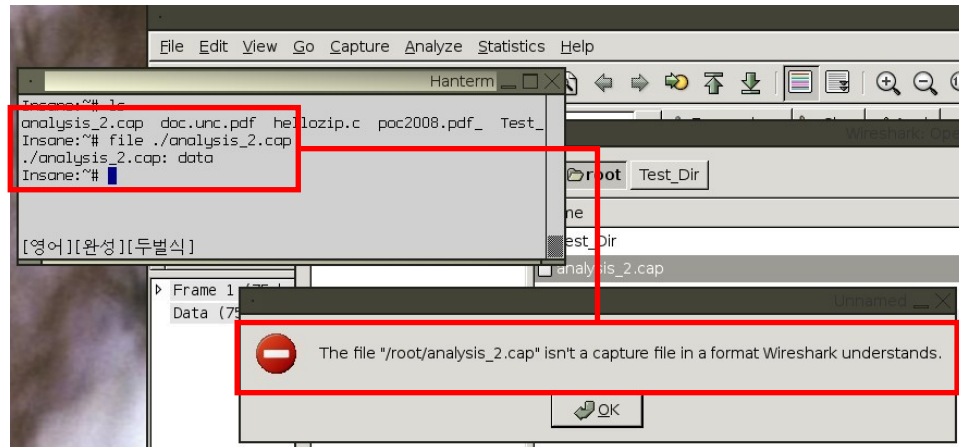
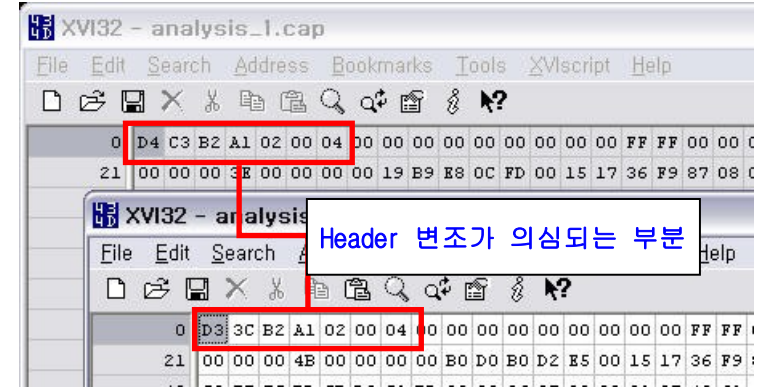
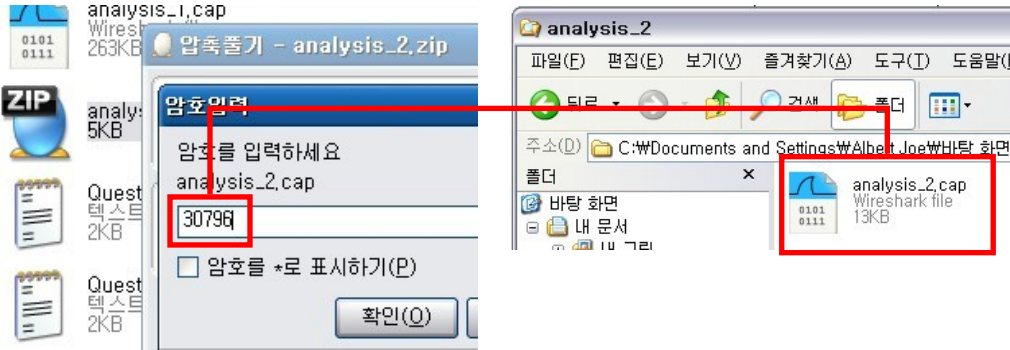
- Filename:** analysis_2.zip (labeled "대상(문제2) ZIP 파일")
- Password:** 30796 (labeled "Password 추출 성공, Hint 88 획득")
- Total Passwords Tested:** 88 (labeled "공격 방법")

3. Network 문제 분석

II. 상세 분석

□ Network 2번 문제 상세 분석

- '30796'을 이용하여 압축 해제 후 file이나 Wireshark를 이용한 분석 시도 실패.
- 문제의 file을 winhex나 Xvi32를 이용하여 확인하면 Dump 파일임을 확인할 수 있음.



3. Network 문제 분석

II. 상세 분석

□ Network 2번 문제 상세 분석

- 변조된 Header 복구 후 file utility를 이용하여 encapsulation type을 확인.
- 1번 문제와 동일하게 SLIP → Ethernet으로 Encapsulation 변환하면 정상 Dump 확인 가능

```
Hanterm _ □  
Insane:~/Test_Dir# file ./analysis_2.cap  
./analysis_2.cap: tcpdump capture file (little-endian) - version  
2.4 (SLIP, capture length 65535)  
Insane:~/Test_Dir#
```



```
Hanterm _ □  
Insane:~/Test_Dir# editcap ./analysis_2.cap -T ether ./ext2.cap  
Insane:~/Test_Dir# file ./ext2.cap  
./ext2.cap: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture length 65535)  
Insane:~/Test_Dir#
```



No.	Time	Source	Destination	Protocol	Info
1	0.000000	89.126.245.189	91.210.90.189	ICMP	Echo (ping)
2	0.000248	91.210.90.189	89.126.245.189	ICMP	Echo (ping)
3	0.007990	89.126.245.189	91.210.90.190	ICMP	Echo (ping)
4	0.026608	91.210.90.190	89.126.245.189	ICMP	Echo (ping)
5	7154044.518	89.86.39.135	91.210.90.189	UDP	Source port: 89.86.39.135
6	7154080.478	89.86.39.135	91.210.94.190	UDP	Source port: 89.86.39.135
7	2.361574	75.206.93.156	91.210.102.132	TCP	x11 > loc-sr
8	2.361942	75.206.93.156	91.210.102.135	TCP	x11 > loc-sr

IMAP 관련 취약점을 이용한 공격 추정

1번 문제에서 얻은 88번째 패킷

Protocol	Info
TCP	1273 > imap [SYN] Seq=0 Len=0 MSS=1460
TCP	1273 > imap [SYN, ACK] Seq=0 Len=0 MSS=1460
TCP	1273 > imap [ACK] Seq=143

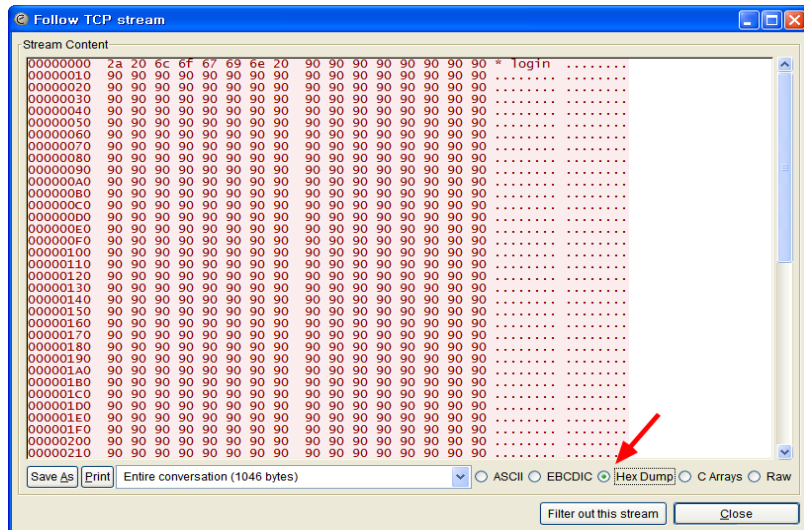
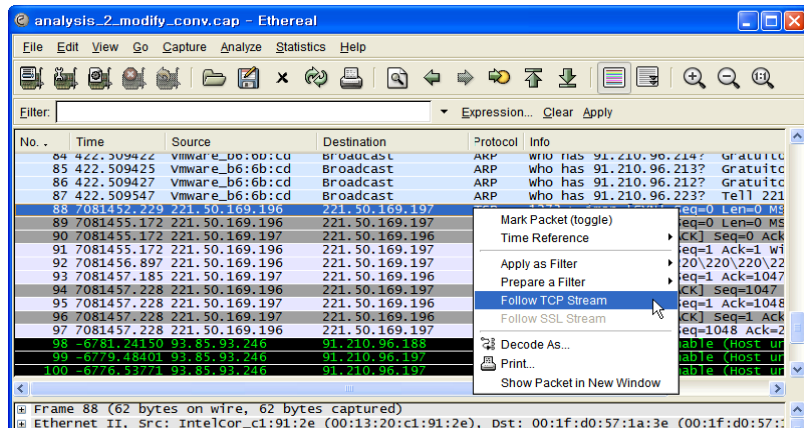
Frame 88 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: IntelCor-cl:91:2e (00:13:20:c1:91:2e), Dst: 00:1f:d0:57:1a:3e (00:1f:d0:57:1a:3e)
Internet Protocol, Src: 221.50.169.196 (221.50.169.196), Dst: 221.50.169.197 (221.50.169.197)
Transmission Control Protocol, Src Port: 1273 (1273), Dst Port: imap (143), Seq: 0, Len: 0

3. Network 문제 분석

II. 상세 분석

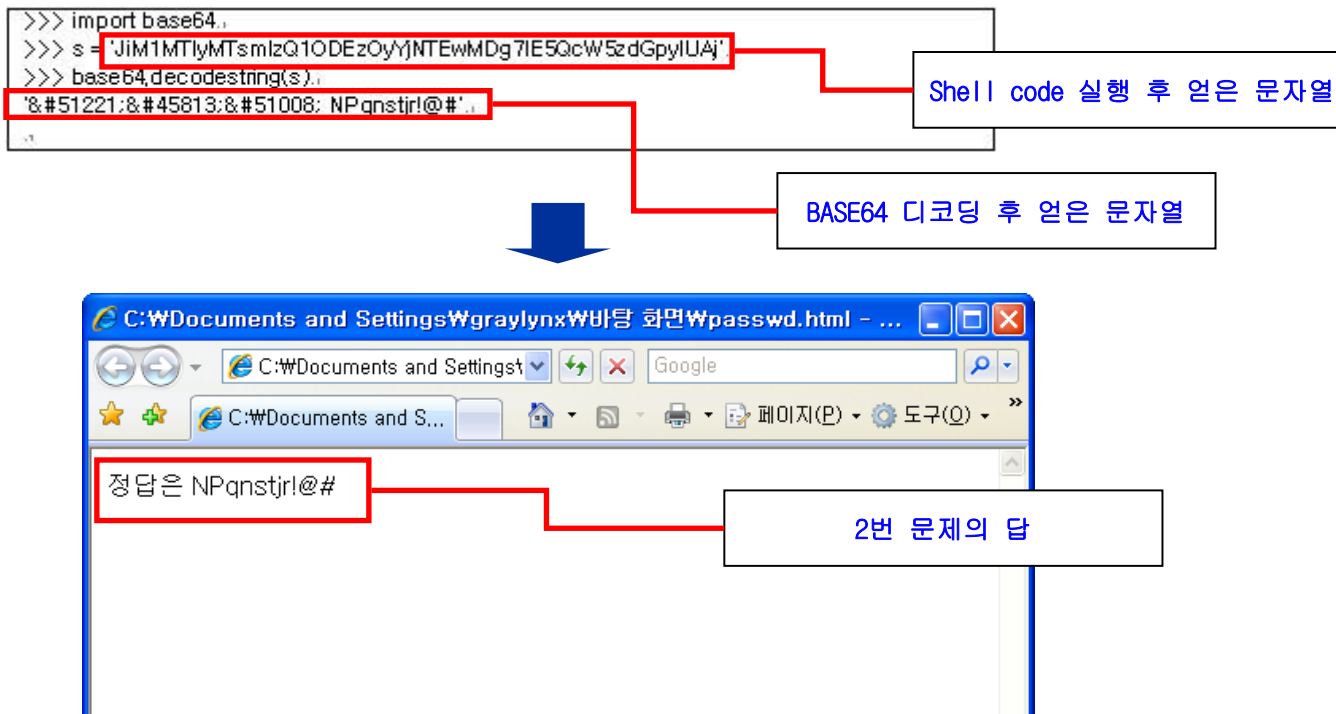
□ Network 2번 문제 상세 분석

— IMAP관련 접속에 대한 자세한 분석을 위해 Follow TCP Stream을 이용하여 IMAP Session에 대한 분석 시도



Network 2번 문제 상세 분석

- 추출된 Shell code를 실행시킬 경우 BASE64로 인코딩 된 것으로 판단되는 문자열을 얻을 수 있으며, Python을 이용한 간단한 Script로 해당 문자열을 디코딩할 수 있음.
- 디코딩된 문자열을 HTML 파일로 저장한 뒤 웹 브라우저로 열면 정답을 확인할 수 있음.



□ 문제 목표

- Ugo Boy BHO에 대한 진단/분석/조치/예방 방법 기술
- Spyware 'sapkin.exe'에 대한 진단/분석/조치/예방 방법 기술

□ 풀이 환경

- 분석도구 : Wireshark, winanalysis, Ollydbg, IDA Pro, TCPView, Process explorer 등

□ 풀이 방법

- IDA Pro 및 Ollydbg를 이용한 Sample.dll 분석 및 Password 출력 조건 분석
- Winanalysis 및 Sysinternal Suite를 이용하여 sapkin.exe의 실행 전/후 시스템 변경 사항 점검, IDA Pro를 이용한 sapkin.exe의 행동방식 분석

□ 풀이 결과

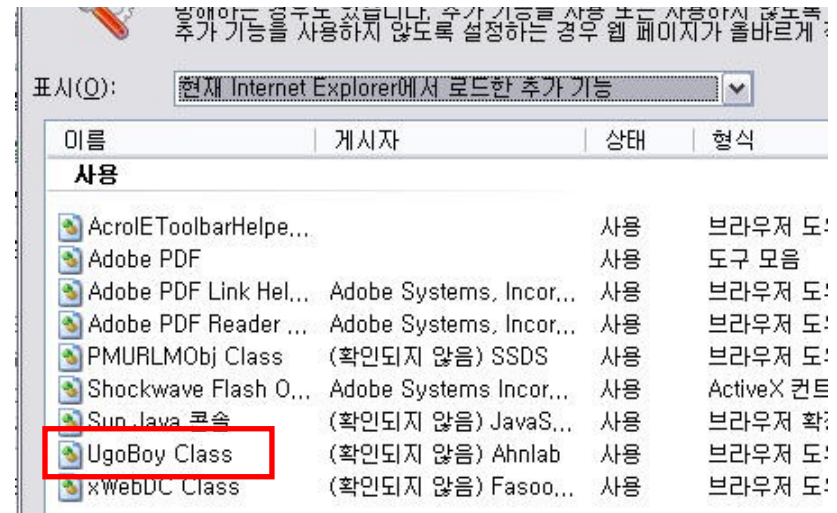
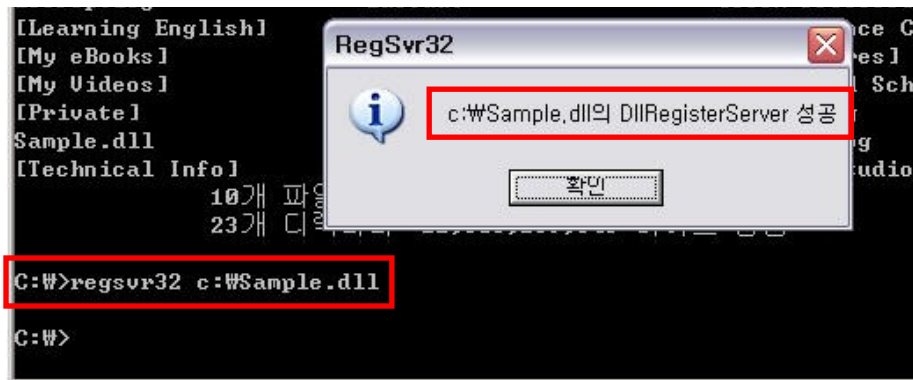
- 1번 문제의 답 : AS long as you love me
- 2번 문제의 답 : 복사 작업 중 Clipboard에 저장되는 정보를 sapkin.log 파일로 저장하는 일종의 Key logger.

□ Spyware 1번 - “진단”

— BHO의 개념

- BHO(Browser Helper Object)는 Internet Explorer Browser에서 기본적으로 지원하는 기능 외에 별도의 기능을 지원하기 위하여 Plug-in 형태로 Internet Explorer에 추가되는 DLL 모듈.
- BHO가 제대로 동작하기 위해서는 Registry 内の,
 - (1) \HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser helper Object 부분에 CLSID가 등록되어야 하며,
 - (2) \HKCR\CLSID에 Class ID가 등록되어 있어야 한다.

- 문제의 DLL을 Regsvr32를 이용하여 등록하면 Internet Explorer 실행 시 UgoBoy BHO가 Load되는 것을 확인할 수 있음. 그러나 System 內 특이한 변동사항은 없음



□ Spyware 1번 - “분석”

- Code 속에 다량의 데이터가 있는 부분을 검색하여 해당 부분이 임의의 Code가 아닌지 확인 수행 → Data부분을 Code로 변환해 보면, 대략 5개의 함수가 존재함을 알 수 있음.

```
.text:100064BB ;
.text:100064BE
.text:100064C0
.text:100064C1
.text:100064C2
.text:100064C3
.text:100064C4
.text:100064C5
.text:100064C6
.text:100064C7
.text:100064C8
.text:100064C9
.text:100064CA
.text:100064CB
.text:100064CC
.text:100064CD
.text:100064CE
.text:100064CF
.text:100064D0
.text:100064D1
.text:100064D2
.text:100064D3
.text:100064D4
.text:100064D5

align 10h
db 81h ;
db 0ECh ;
db 2Ch ;
db 1 ;
db 0 ;
db 001h ;
db 0F8h ;
db 20h ;
db 3 ;
db 10h ;
db 33h ;
db 0C4h ;
db 89h ;
db 84h ;
db 24h ;
db 28h ;
db 1 ;
db 0 ;
db 56h ;
db 88h ;
```

Data를 포함하고 있는 부분

```
.text:100064BB ;
.text:100064BE
.text:100064C0
.text:100064C6
.text:100064CB
.text:100064CD
.text:100064D5
.text:100064DC
.text:100064DD
.text:100064DE
.text:100064DF
.text:100064E0
.text:100064E1
.text:100064E2
.text:100064E3
.text:100064E4
.text:100064E5
.text:100064E6
.text:100064E7
.text:100064E8
.text:100064E9
.text:100064EA
.text:100064EB
.text:100064EC
.text:100064ED
.text:100064EE
.text:100064EF
.text:100064F0
.text:100064F1
.text:100064F2
.text:100064F3
.text:100064F4
.text:100064F5
.text:100064F6
.text:100064F7
.text:100064F8
.text:100064F9
.text:100064FA
.text:100064FB
.text:100064FC
.text:100064FD
.text:100064FE
.text:100064FF
.text:10006500
.text:10006501
.text:10006502
.text:10006503
.text:10006504
.text:10006505
.text:10006506
.text:10006507
.text:10006508
.text:10006509

align 10h
sub esp, 12ch
mov eax, dword_100320F8
xor eax, esp
mov [esp+128h], eax
push esi
mov esi, [esp+130h]
push edi
mov edi, [esp+130h]
mov dword ptr [esp+8], 0
test esi, esi
jnz short loc_100064F3
push esi
jmp short loc_1000652D

loc_100064F3:
push ecx
lea eax, [esp+130h]
push eax
call sub_100064F9
add esp, 8
test eax, eax
jnz short loc_10006501
lea ecx, [esp+130h]
push ecx
```

Data에서 Code로 변환된 부분

```
sub_100063C0 .text
sub_10006400 .text
sub_100064C0 .text
sub_10006550 .text
SuspiciousFunction .text
SuspiciousFunction2 .text
UgoBoy__Invoke .text
ARGUMENT_MODULE_STATE(void) .text
```

수상한 Stack Operation을 하는 두개의 함수

IDA의 Com Plugin을 사용하여 발견한 함수

새로 발견한 5개의 함수들

4. Spyware 문제 분석

II. 상세 분석

□ Spyware 1번 - “분석”

- IUgoBoy_Invoke 함수를 분석하여, 웹페이지가 로딩되면 SuspiciousFunction2 함수가 호출됨을 알 수 있으며, Suspicious Function2에는 Encrypt된 code가 저장된 것으로 의심되는 부분을 찾을 수 있음. 이 후에 나오는 부분에서 Decrypt 함수로 의심되는 함수 발견 가

```
puArgErr= dword ptr 24h
mov     eax, [esp+dispidMember]
sub     eax, 0FDh
jz      short loc_1000742B
```

dispidMember = 259 즉,
DISPID_DOCUMENTCOMPLETE일때,
(웹페이지 로딩 시)

```
sub     eax, 6
jnz     short loc_1000743B
```

```
mov     eax, [esp+dispparans]
mov     ecx, [eax+DISPPARANS.rguarg]
mov     edx, [ecx+18h]
mov     ecx, [esp+pthis]
push   edx
add     esp, 4
call   SuspiciousFunction2
xor     eax, con
retn   24h
```

```
mov     [esp+98h+var_58], 4Eh
mov     [esp+98h+var_59], 48h
mov     [esp+98h+var_5A], C1
mov     [esp+98h+var_5B], B1
mov     [esp+98h+var_5C], 8Ch
mov     [esp+98h+var_5D], D1
mov     [esp+98h+var_5E], 1Eh
mov     [esp+98h+var_5F], 46h
mov     [esp+98h+var_60], B1
mov     [esp+98h+var_61], 3
mov     [esp+98h+var_62], 1Fh
mov     [esp+98h+var_63], B1
mov     [esp+98h+var_64], 2
mov     [esp+98h+var_65], 5Dh
mov     [esp+98h+var_66], 11h
mov     [esp+98h+var_67], 7
mov     [esp+98h+var_68], 0Ch
mov     [esp+98h+var_69], 48h
mov     [esp+98h+var_6A], C1
mov     [esp+98h+var_6B], B1
mov     [esp+98h+var_6C], 8Ch
mov     [esp+98h+var_6D], D1
mov     [esp+98h+var_6E], 1Eh
mov     [esp+98h+var_6F], 47h
mov     [esp+98h+var_70], 2
mov     [esp+98h+var_71], 1Fh
mov     [esp+98h+var_72], 3
mov     [esp+98h+var_73], 19h
mov     [esp+98h+var_74], 10h
mov     byte ptr [esp+98h+var_3C+1], 9
mov     byte ptr [esp+98h+var_3C+2], C1
mov     byte ptr [esp+98h+var_38+1], 4
mov     byte ptr [esp+98h+var_38+2], 19h
mov     byte ptr [esp+98h+var_38+3], 2Eh
mov     [esp+98h+var_34], al
mov     [esp+98h+var_33], 0Ah
mov     [esp+98h+var_32], 0Fh
```

Encrypt된 코드가 저장된 부분

```
EAX 01A1F618 ASCII "cjsrhakq1"
ECX 00000040
EDX 00000032
EBX 00000000
ESP 01A1F5D8
EBP 02815E00
ESI 02818C98 ASCII "http://global.ahnlab.com/global/viruscenter_view.ESD?virus_seq=16376&seType=2"
EDI 02815FE0
```

이 ydbg를 이용하여 해당 함수
부분에 Breakpoint 걸고 Run
수행 후 단계별 실행하면 아래와
같은 URL 정보 획득

```
lea     eax, [esp+0A0h+var_6C]
push   eax
lea     ecx, [esp+0A4h+var_60]
push   ecx
push   ecx
mov     [esp+0ACh+var_6C], 63h
mov     [esp+0ACh+var_6B], 60h
mov     [esp+0ACh+var_6A], 73h
mov     [esp+0ACh+var_69], 72h
mov     [esp+0ACh+var_68], 68h
mov     [esp+0ACh+var_67], 61h
mov     [esp+0ACh+var_66], 68h
mov     [esp+0ACh+var_65], 71h
mov     [esp+0ACh+var_64], 6Ch
call   sub_10004450
push   esi
call   sub_10004500
push   eax
lea     ecx, [esp+98h+var_...
```

위에서 Encrypt가 저장된 배열과
배열의 크기(77), ASCII 배열 넘김

넘겨진 인수를 이용한 함수 호출
(Decrypt하는 함수로 의심 됨)

위와 같은 사실로 상기의 URL이 Encrypt된 문자열이었으며, 복호화 후 상기의 문자열로 바뀔 것을 알 수 있음

□ Spyware 1번 - “분석”

- Suspicious Function2 함수의 분석을 계속하다 보면 밑부분에서 Suspicious Function1의 호출부분을 찾을 수 있음 → Decrypt의 결과로 미루어 Suspicious Function1이 최종 Password를 Decrypt하는 함수임을 짐작할 수 있음.

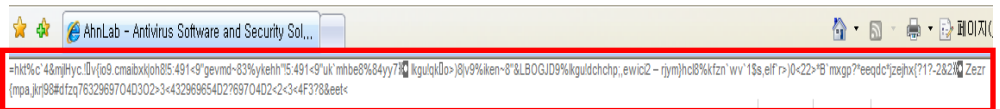
```
mov [esp+164h+var_E4], 1
mov [esp+164h+var_E3], 4
mov [esp+164h+var_E2], a1
mov [esp+164h+var_E1], 13h
mov [esp+164h+var_E0], 45h
mov [esp+164h+var_D8], d1
mov [esp+164h+var_D6], 51h
mov [esp+164h+var_D5], 50h
mov [esp+164h+var_D4], d1
mov [esp+164h+var_D3], c1
mov [esp+164h+var_D2], 55h
mov [esp+164h+var_D1], a1
mov [esp+164h+var_D0], 15h
mov [esp+164h+var_C0], 0Bh
mov [esp+164h+var_B8], 5
mov [esp+164h+var_B7], 49h
mov [esp+164h+var_B6], 49h
mov [esp+164h+var_B5], 0Fh
mov [esp+164h+var_B4], 021, 3Fh
mov [esp+164h+var_B3], 011, 39h
mov [esp+164h+var_B2], 001, 3Ch
mov [esp+164h+var_B1], 081, 5Fh
mov [esp+164h+var_B0], c1
mov [esp+164h+var_A0], 57h
mov [esp+164h+var_90], 45h
mov [esp+164h+var_88], 19h
mov [esp+164h+var_87], 41h
mov [esp+164h+var_86], c1
mov [esp+164h+var_85], 4Fh
mov [esp+164h+var_84], 5Ch
mov [esp+164h+var_83], 0Fh
mov [esp+164h+var_82], 19h
mov [esp+164h+var_81], 8
mov [esp+164h+var_80], 44h
mov [esp+164h+var_7F], 46h
mov [esp+164h+var_7E], 8
mov [esp+164h+var_7D], 15h
```

Suspicious function2에서와 같이 Encrypt된 것으로 의심되는 문자열 발견



이 lydbg를 이용하여 Encrypt된 부분이 복호화되는 과정을 Trace해 보면 오른쪽과 같은 문자열이 생성되고 이 문자열이 웹브라우저에서 출력되는 것을 확인할 수 있다.

	ASCII
02818AC0 63 00 60 00 34 00	e.h.k.t.%c..4.
02818AD0 79 00 63 00 2E 00	&.m.j.l.H.y.c...
02818AE0 6F 00 39 00 2E 00	!.0.v.{.i.o.9...
02818AF0 78 00 68 00 7C 00	c.m.a.i.b.x.k. .
02818B00 3A 00 34 00 39 00	o.h.8.!..5.:4.9.
02818C00 65 00 76 00 6D 00	l.<.9."g.e.v.m.
02818D00 68 00 68 00 22 00 21 00 35 00 3A 00 34 00 39 00	d.~.8.3.%y.k.e.
02818E00 31 00 3C 00 39 00 22 00 25 00 68 00 60 00 6D 00	l.<.9."u.k.~m.



1차 Decrypt된 URL로 접속 시 상기의 이상한 문자열을 얻을 수 있다. 아마도 Decrypt하는 데 쓰인 Key가 틀렸기 때문 일 것이라고 의심할 수 있음.

□ Spyware 1번 - “분석”

- Suspicious Function 1을 분석해 보면, 앞서 표시한 배열을 이용한 Data 초기화 후, ‘%d%H%M%s’ 형식의 날짜 문자열을 입력 받아 Decrypt 함을 알 수 있고, 최종적으로 XOR 연산을 수행하여 Password를 출력함을 알 수 있음.
- 이 의미는 제대로 된 날짜가 입력되지 않으면 Decrypt가 제대로 수행되지

```

0200F4C0 0D 50 5B 45 15 50 51 04 16 55 5A 5D 78 4A 52 1E
0200F4D0 11 47 46 4A 59 5C 08 1E 53 55 51 58 52 4B 5A 4C
0200F4E0 5F 50 08 10 05 09 05 09 01 04 09 13 57 56 47 5D
0200F4F0 54 46 08 02 15 4A 5A 55 58 50 12 10 05 09 05 09
0200F500 01 04 09 13 45 58 51 5D 58 5A 55 09 15 0B 05 49
0200F510 49 0F 3F 39 3C 5F 5A 57 45 19 41 5A 4F 5C 0F 19
0200F520 08 44 46 08 15 5A 5A 55 5E 46 08 13 16 7F 73 7F
0200F530 77 72 74 08 15 5F 5A 57 45 19 54 52 58 50 59 4D
0200F540 06 14 55 46 59 50 58 02 3C 3E 38 43 5A 4A 5C 4D
0200F550 58 5B 5C 09 15 58 57 4A 5E 58 47 47 50 02 15 43
0200F560 1C 5D 5C 57 50 41 0F 19 00 04 02 03 0E 19 73 50
0200F570 5D 40 57 41 0F 19 54 55 41 5C 53 1B 5A 49 54 5A
0200F580 58 40 4B 0E 01 0C 1C 02 16 0A 3F 39 3C 69 54 4A
0200F590 42 43 5D 41 51 1F 5B 56 42 44 09 09 13 57 57 4A
0200F5A0 41 0F 06 02 02 0A 07 09 07 77 04 75 03 7C 03 0E
0200F5B0 03 04 04 02 02 0A 07 09 06 0D 04 75 02 0C 07 09
0200F5C0 07 77 04 75 02 0F 03 0C 03 04 04 77 03 0C 09 16
0200F5D0 55 5D 44 0D
    
```

< Stack에 저장된 복호화 대상 >

```

.text:10006C92      push    114h
.text:10006DDA      call   _malloc
.text:10006DE4      mov     esi, eax
.text:10006DE6      push   0
.text:10006DE8      push   esi
.text:10006DE9      call   _memset
    
```

< 복호화 대상용 메모리 할당 >

```

.text:10006DEE      push   0
.text:10006DF0      call   _time64
.text:10006DF5      mov     [esp+178h+var_138], eax
.text:10006DF9      mov     al, 25h
.text:10006DFB      add     esp, 14h
.text:10006DFE      mov     [esp+164h+var_130], al
.text:10006E02      mov     [esp+164h+var_12E], al
.text:10006E06      mov     [esp+164h+var_12C], al
.text:10006E0A      mov     [esp+164h+var_12A], al
.text:10006E0E      lea    eax, [esp+164h+var_130]
.text:10006E12      push   eax
.text:10006E13      lea    ecx, [esp+168h+var_148]
.text:10006E17      push   ecx
.text:10006E18      lea    ecx, [esp+16Ch+var_138]
.text:10006E1C      mov     [esp+16Ch+var_134], ecx
.text:10006E20      mov     [esp+16Ch+var_12F], 64h
.text:10006E25      mov     [esp+16Ch+var_12D], 48h
.text:10006E2A      mov     [esp+16Ch+var_12B], 4Dh
.text:10006E2F      mov     [esp+16Ch+var_129], 53h
.text:10006E34      mov     [esp+16Ch+var_128], 0
.text:10006E39      call   sub_10006550
    
```

< 날짜 시간형식의 문자열 변환 >

```

.text:10006E58      add     esp, 4
.text:10006E5B      push   eax
.text:10006E5C      push   114h
.text:10006E61      lea    edx, [esp+174h+var_124]
.text:10006E65      push   edx
.text:10006E66      call   sub_10004450
    
```

< 복호화 함수 호출 >

```

.text:10006E3E      mov     eax, [esp+164h+var_148]
.text:10006E42      mov     ecx, [eax-0Ch]
.text:10006E45      push   esi
.text:10006E46      push   ecx
.text:10006E47      push   eax
.text:10006E48      mov     [esp+170h+var_4], 0
.text:10006E53      call   sub_100044A0
    
```

< 변환된 문자열을 ASC코드로 재변환 >

□ Spyware 1번 - “조치”

- 1. 모든 Internet Explorer를 종료
- 2. HKEY_CLASSES_ROOT\CLSID\{202D7790-9289-483E-8E4A-16BD52ADFC2 A} \InprocServer32 키의 값에 나와있는 경로의 Sample.dll을 삭제.
- 3. HKEY_CLASSES_ROOT\CLSID\{202D7790-9289-483E-8E4A-16BD52ADFC2A} 키 값 삭제
- 4. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion \Explorer\Browser Helper Objects\{202D7790-9289-483E-8E4A-16BD52ADFC2A} 키 값 삭제
- 혹은, Internet Explorer → 도구 → ‘추가기능관리’로 이동하여 의심스러운 BHO는 “사용 안 함”에 Check.

□ Spyware 1번 - “예방”

- 1. 검증되지 않은 Site에서 다운로드 받은 파일을 무턱대고 실행하지 않음.
- 2. 검증되지 않은 Site에서 제공하는 ActiveX 등을 무심코 설치하지 않는다.
- 3. 백신 S/W는 항상 최신버전으로 유지하고, 전용 Checker 프로그램이나 Process Explorer 혹은 Internet Explorer의 ‘추가기능관리’를 통하여 의심스러운 BHO가 설치되지 않았는지 수시로 확인한다.

4. Spyware 문제 분석

II. 상세 분석

□ Spyware 2번 - “진단”

- 악성코드를 초기 설치하는 install.bat 분석
- Winanalysis를 이용하여 System 內 변경 점 분석 (클립보드의 내용을 로깅하는 Key logger)

install MS-DOS 일괄 파일 1KB
Question_Eng 텍스트 문서
Question_Kor 텍스트 문서 1KB
sapkin

기존에 설치된 것이 있으면 -k 옵션을 이용하여 실행 중지

install - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
@echo off
start %SystemRoot%\system32\sapkin.exe -k
copy /y .\sapkin.exe %SystemRoot%\system32\sapkin.exe
cd /d %SystemRoot%\system32
sapkin.exe -i
sapkin.exe -s
pause

Sapkin 에러로그 생성
System32 폴더로 Sapkin.exe 복사
Sapkin 서비스 시작
System32 폴더로 이동하여 Sapkin.exe 실행

WWINSANE-COMPUTER Changes

	Critical	Warning	Info	Ignored	Tested
Files	0	0	6	0	2008-10-20 오후 10...
Groups	0	0	0	0	2008-10-20 오후 10...
Registry	0	3	54	7	2008-10-20 오후 10...
HKLM\	0	0	0	0	2008-10-20 오후 10...
Rights	0	0	0	0	2008-10-20 오후 10...
Scheduler	1	0	0	0	2008-10-20 오후 10...
Services	0	0	0	0	2008-10-20 오후 10...
Shares	0	0	0	0	2008-10-20 오후 10...
System	0	0	0	0	2008-10-20 오후 10...
Users	0	0	0	0	2008-10-20 오후 10...

새로 생성된 서비스

Description	Name	New Value
New Service	sapkin	

새로 생성된 파일

- New File C:\sapkin.log
- New File C:\WINDOWS\System32\sapkin.exe
- New File C:\WINDOWS\System32\sapkin.log

사용자의 Clip board에 저장된 내용을 sapkin.log에 저장

□ Spyware 2번 - “분석”

— IDA Pro를 이용하여 주요 동작 방식, Anti-debugging 기법 분석

- Regmon (18467-41), Process explorer (PROCEXPL), Process Monitor (PROCMON_WINDOW_CLASS), ?? (hzzfqgqzbt)

이 실행은 가치 있고 실행 가치 시 가제 조르

```
.text:004029A0 MainThread proc near.  
...  
.text:004029A8      mov     [esp+14h+var_14], offset ClassName ; "18467-41"  
.text:004029B0      cmp     eax, 1  
.text:004029B3      mov     [esp+14h+var_10], offset aProcepl ; "PROCEXPL"  
.text:004029BB      mov     [esp+14h+var_C], offset aProcmo ;  
"PROCMON_WINDOW_CLASS"  
.text:004029C3      mov     [esp+14h+var_8], offset aHzzfqgqzbt ; "hzzfqgqzbt"  
...  
.text:004029EC      push   0 ; lpWindowNam  
.text:004029EE      push   eax ; lpClassName  
.text:004029EF      call  Anti_Monitors_GetMonitorProcessID ;  
.text:004029F4      add     esp, 8  
.text:004029F7      test   eax, eax  
.text:004029F9      jz     short loc_402A04  
.text:004029FB      push   eax ; Monitor_Process_ID  
.text:004029FC      call  KillMonitorProcess ;  
...
```

실행 여부 점검대상 클래스 ID

주어진 클래스 ID와 동일한 ID를 갖고 있는 프로세스의 ID 반환

반환된 ID를 갖고 있는 프로세스를 Terminate

- Dispatcher_Thread proc near.
...
ext:00402772 push offset aC ;
"C:\windows\system32\drivers\sapkin.sys"
.text:00402777 push 68h ; Const_68h
.text:00402779 mov dword_404C18, eax
.text:0040277E call Load_sapkin_Driver ;
.text:00402783 mov ecx, 40h ;

“진단” 단계에서 감지되지 않은 sapkin.sys의 생성을 위한 정보

Sapkin.sys의 생성 정보를 인수로 한 Load_sapkin_Driver 함수 호출

□ Spyware 2번 - “분석”

— Load_sapkin_Driver 함수를 통한 Sapkin.sys 실행(1)

- Load_sapkin_Driver 함수의 시작 부분
- Driver Load를 위한 권한 설정
- Beep Service Stop
- c:\windows\system32\drivers\beep.sys 파일을 c:\windows\system32\beep.sys로 복사
- 68h와 'C:\windows\system32\drivers\sapkin.sys' 문자열을 인자로 넘겨주면서 Copy_File_From_Resource 함수를 호출

```

.text:00401650 ; int __cdecl Load_sapkin_Driver(LPCSTR Const_68h, int Driver_Name)..
.text:00401650 Load_sapkin_Driver proc near
...
.text:00401656     push  offset ProcName           ; "RtlAdjustPrivilege"..
.text:00401658     push  offset ModuleName        ; "ntdll.dll"..
.text:00401660     mov   [esp+44h+var_2C], 0..
.text:00401668     mov   [esp+44h+var_28], 0..
.text:00401670     mov   [esp+44h+var_24], 0..
.text:00401678     call  ds:GetModuleHandleA..
.text:0040167E     push  eax                       ; hModule..
.text:0040167F     call  ds:GetProcAddress..
.text:00401685     mov   esi, eax
...
.text:00401687     lea  eax, [esp+3Ch+var_2C]..
.text:00401688     push  eax..
.text:0040168C     push  0..
.text:0040168E     push  1..
.text:00401690     push  10                         ; SeLoadDriverPrivilege..
.text:00401692     call  esi..
.text:00401694     lea  ecx, [esp+3Ch+var_28]..
.text:00401698     push  ecx..
.text:00401699     push  0..
.text:0040169B     push  1..
.text:0040169D     push  20                         ; SeDebugPrivilege..
.text:0040169F     call  esi..
.text:004016A1     lea  edx, [esp+3Ch+var_24]..
.text:004016A5     push  edx..
.text:004016A6     push  0..
.text:004016A8     push  1..
.text:004016AA     push  9                         ; SeTakeOwnershipPrivilege..
.text:004016AC     call  esi..
...
.text:004016C8 loc_004016C8:                                ; CODE XREF:
Load_sapkin_Driver+6Dj..
.text:004016C8     push  0F01FFh                   ; dwDesiredAccess..
.text:004016CD     push  offset ServiceName        ; "Beep"..
.text:004016D2     push  eax                       ; hSCMManager..
.text:004016D3     call  ds:OpenServiceA..
...
.text:004016FA     push  ecx                       ; lpServiceStatus..
.text:004016FB     push  SERVICE_STOPPED          ; dwControl..
.text:004016FD     push  esi                       ; hService..
.text:004016FE     call  ds:ControlService..
...
.text:00401766     mov   esi, ds:CopyFileA..
.text:0040176E     push  0                         ; hFailIfExists..
.text:0040176E     push  offset NewFileName        ; "c:\windows\system32\beep.sys"..
.text:00401773     push  offset ExistingFileName   ; "c:\windows\system32\drivers\beep.sys"..
.text:00401778     call  esi : CopyFileA..
...
.text:00401785     mov   ecx, [esp+3Ch+Driver_Name]..
.text:00401789     mov   edx, [esp+3Ch+Const_68h]..
.text:0040178E     push  ecx                       ; Driver_Name_Path..
.text:0040178E     push  edx                       ; Resource_Identifier..
.text:0040178F     call  Copy_File_From_Resource..
...
    
```

□ Spyware 2번 - “분석”

— Load_sapkin_Driver 함수를 통한 Sapkin.sys 실행(2)

- Copy_File_From_Resource 함수의 시작 부분
- 지정된 ID의 Resource를 검색
- 검색된 Resource를 Load
- c:\Windows\system32\drivers\sapkin.sys 파일로,
Load된 Resource를 Dump하여 sapkin.sys 파일 생성

```

.text:00401540 Copy_File_From_Resource proc
...
.text:00401558      push 0                      ; lpModuleName
.text:0040155D      call ds:GetModuleHandleA
.text:00401563      mov edi, eax
.text:00401565      mov eax, [esp+28h+Resource_Identifier]
.text:00401569      push offset Type           ; "BIN"
.text:0040156E      push eax                   ; lpName
.text:0040156F      push edi                   ; hModule
.text:00401570      call ds:FindResourceA
.text:00401576      mov esi, eax
...
.text:0040158E      call ?AfxGetModuleState@@@YGPVAFX_MODULE_STATE@@@XZ
AfxGetModuleState(void)
.text:00401593      mov eax, [eax+0Ch]
.text:00401596      push esi                   ; hResInfo
.text:00401597      push eax                   ; hModule
.text:00401598      call ds:LoadResource
.text:0040159E      test ecx, ecx
...
.text:004015D3      push esi                   ; hResInfo
.text:004015D4      push edi                   ; hModule
...
.text:004015D8      call ds:SizeofResource
.text:004015DB      lea ecx, [esp+28h+var_1C]
.text:004015DF      mov esi, eax
.text:004015E1      call ??0CFile@@@QAE@XZ    ; CFile::CFile(void)
.text:004015E6      mov ecx, [esp+28h+Driver_Name_Path]
.text:004015EA      push 0
.text:004015EC      push 9001h
.text:004015F1      push ecx
.text:004015F2      lea ecx, [esp+34h+var_1C]
.text:004015F6      mov [esp+34h+var_4], 0
.text:004015FE      call ?Open@CFile@@@UAHPBDIPAVCFileException@@@Z
CFile::Open(char const *, uint, CFileException *)
...
    
```

```

.text:0040178D      push ecx                   ; Driver_Name_Path
...
.text:004017A2      push 0                      ; bFailIfExists
.text:004017A4      push offset ExistingFileName
...
.text:004017A9      push offset aCWindowsSystem
...
.text:004017B5      esi; CopyFileA
...
.text:004017C0      call ds>DeleteFileA
...
.text:004017C8      push 0                      ; dwNumServiceArgs
...
ds:StartServiceA
...
MOVEFILE_REPLACE_EXISTING; dwFlags
...
.text:004017EA      push offset ExistingFileName
...
.text:004017EA      push offset NewFileName
...
.text:004017EF      call ds:MoveFileExA
    
```

생성된 sapkin.sys를 beep.sys로 덮어 씌워 복사 함 (beep.sys변조)

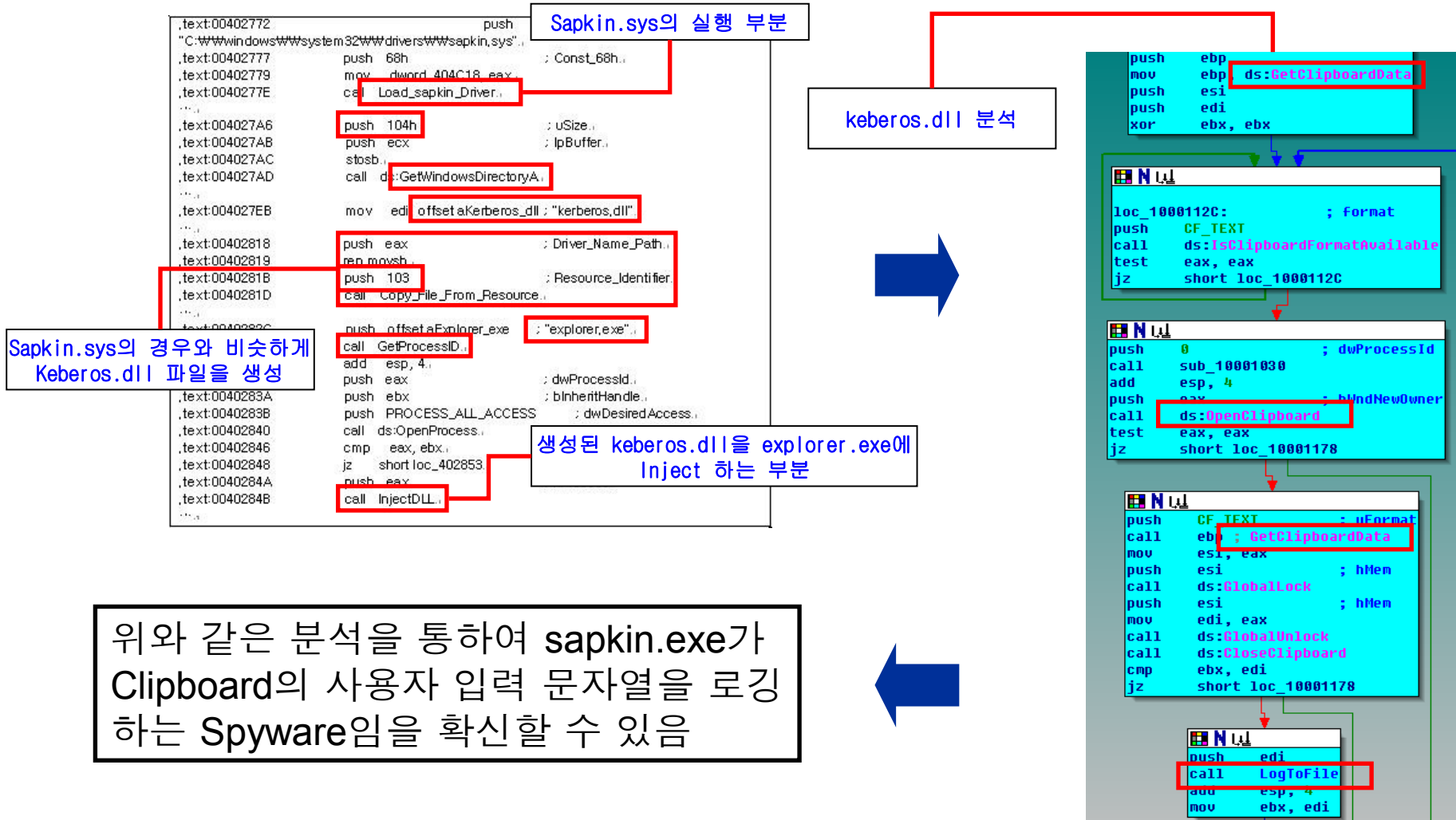
변조가 완료되었으므로 흔적을 남기지 않기 위해 sapkin.sys 삭제

변조된 beep.sys로 beep서비스를 재시작하고 백업했던 beep.sys도 변조된 beep.sys로 교체

이 후부터 beep.sys 서비스를 이용하여 sapkin.sys가 항상 동작하게 됨

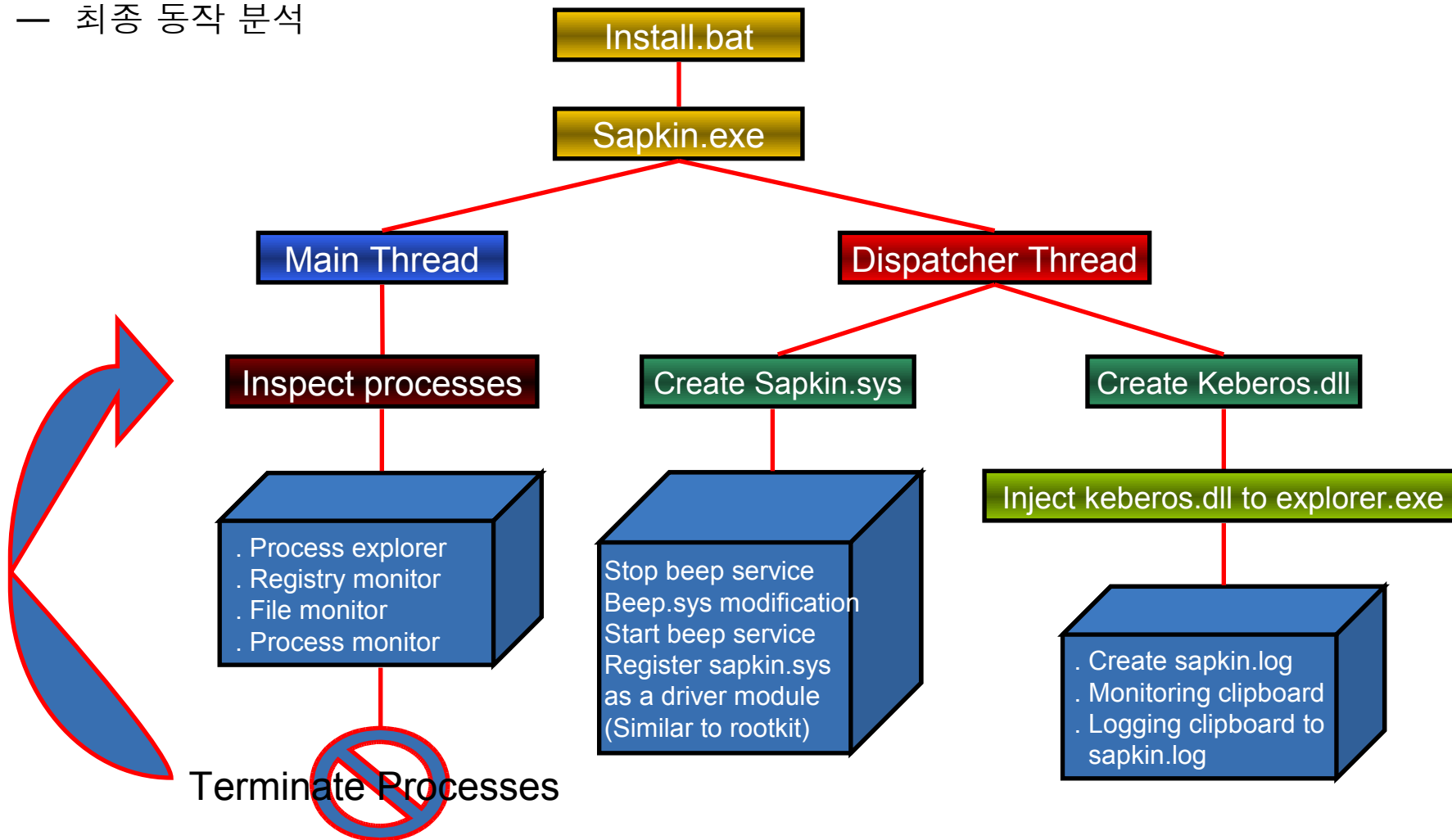
□ Spyware 2번 - “분석”

— Sapkin.sys 실행 후의 keberos.dll 생성 및 explorer.exe에 Injection



□ Spyware 2번 - “분석”

— 최종 동작 분석



□ Spyware 2번 - “조치”

- 1. Stop sapkin service
- 2. Delete c:\windows\system32\sapkin.exe
- 3. Stop beep service (To unload sapkin.sys from memory)
- 4. Delete c:\windows\sapkin.log
- 5. Delete c:\windows\keberos.dll
- 6. Reboot system

□ Spyware 2번 - “예방”

- 1. 검증되지 않은 Site에서 다운로드 받은 파일을 무턱대고 실행하지 않는다.
- 2. PC방과 같이 안정성이 검증되지 않은 장소에서는 민감한 정보를 다루는 업무를 수행하지 않는다.
- 3. 백신 S/W는 항상 최신버전으로 유지한다.

감사합니다.

기타문의 사항은 20silvers@naver.com으로...