Malware Analysis Start

이강석 Certlab%Gmail.com

CodeEngn

ASM Love
어셈블리어 개발자그룹
[어셈러브] asmlove.co.kr

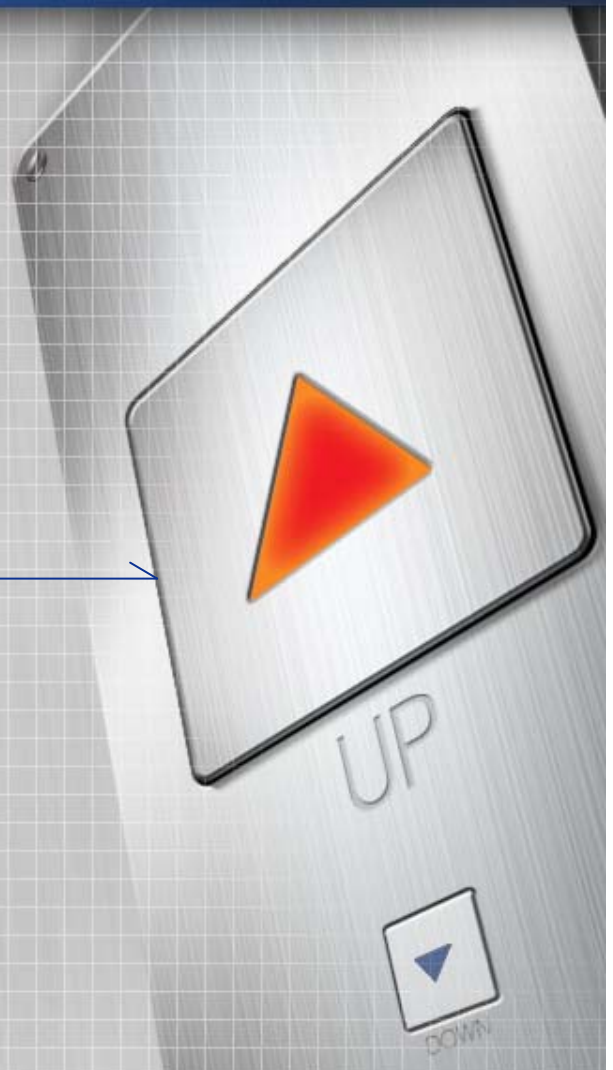
1st CodeEngn Seminar
07.07.21
http://www.CodeEngn.com

Agenda

1. 악성코드의 정의
2. 공부해야 할 것들
3. 악성코드의 제작
4. 발전하는 악성코드 제작툴
5. 악성코드 분석 랩 구축
6. 악성코드 분석
7. AntiVirus Program
8. 악성코드 탐지기법

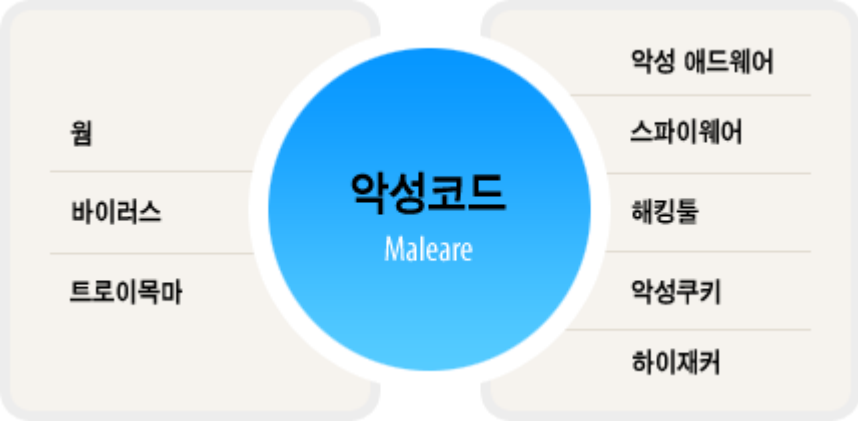
#1. MalwareCode

악성코드의 소개



#1. MalwareCode

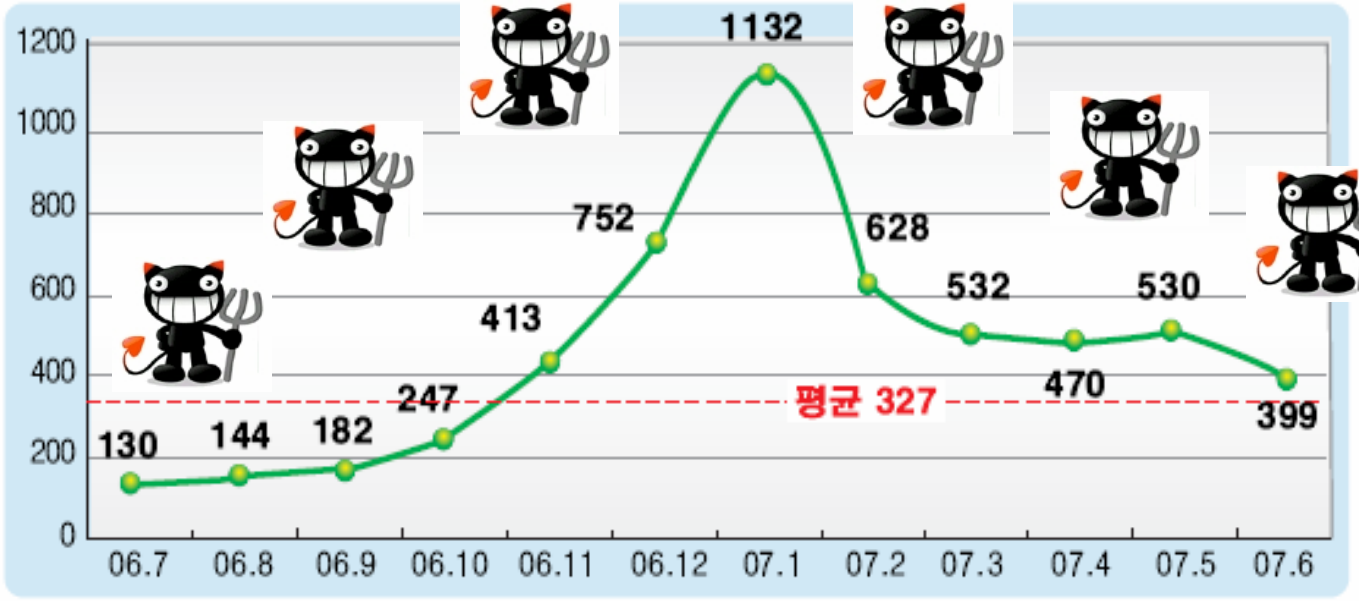
■ Malicious Code : Malware
(Virus + Worm + Backdoor,
Trojan, Spyware, Etc..)



■ 정의
사용자의 의사와는 관계없이 시스템을 파괴하거나 정보를 유출하는 등 악의적 활동을 수행하도록 의도적으로 제작된 소프트웨어를 말합니다.

#1. MalwareCode

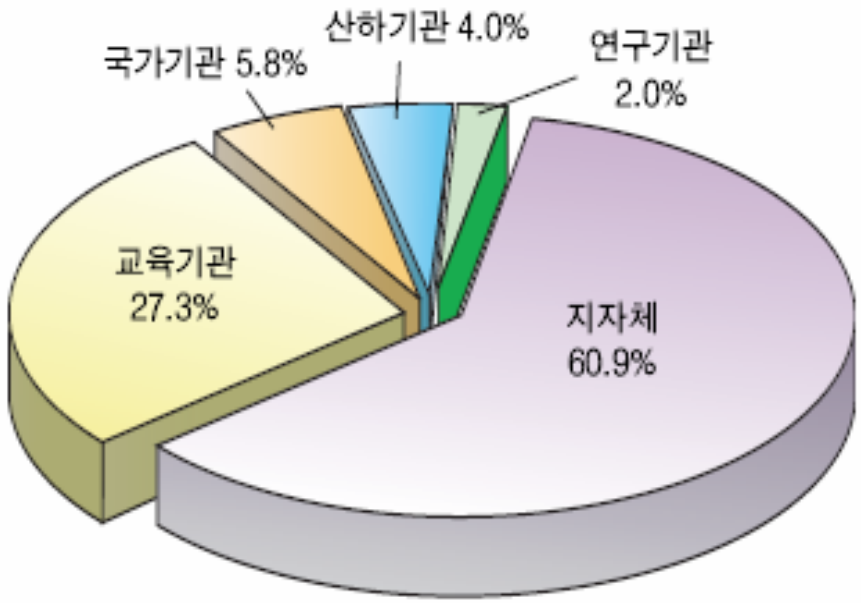
■ 웜, 바이러스 감염사고 월별 발생 추이



[그림 4] 2006년 7월 이후 월별 악성코드 감염사고 발생 추이

출처 : 국가정보원 – Monthly 사이버 시큐리티 2007년 7월호

#1. MalwareCode

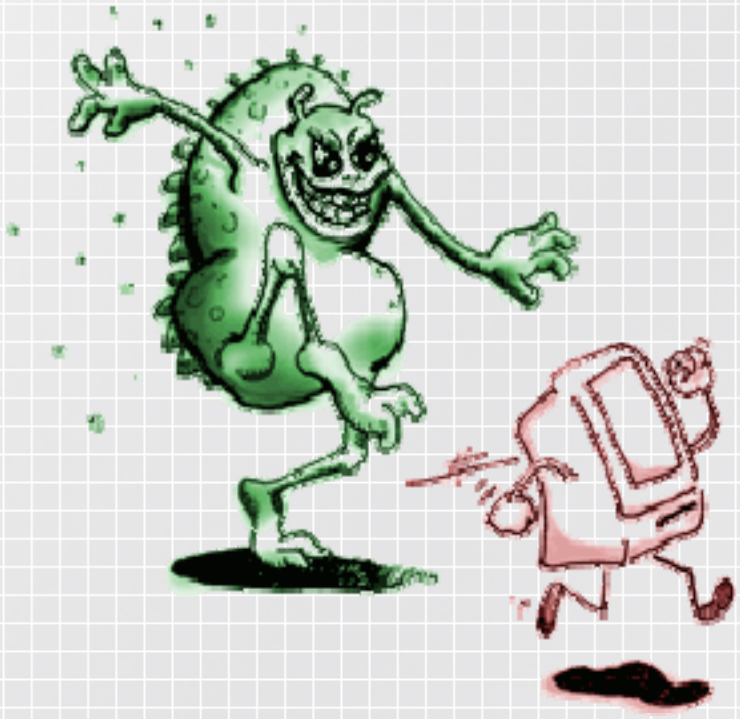


[그림 5] 2007년 6월 악성코드 감염기관 분포

출처 : 국가정보원 –
Monthly 사이버 시큐리티 2007년 7월호

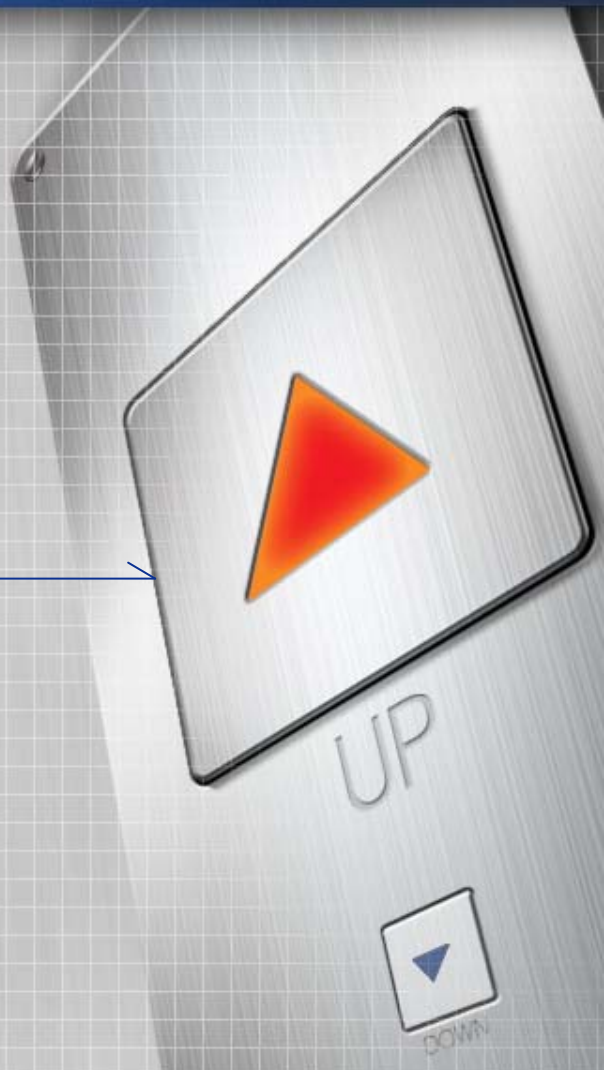
 # 문제점

- 최신 윈도우 보안패치 미흡
- 백신프로그램의 업데이트 미흡



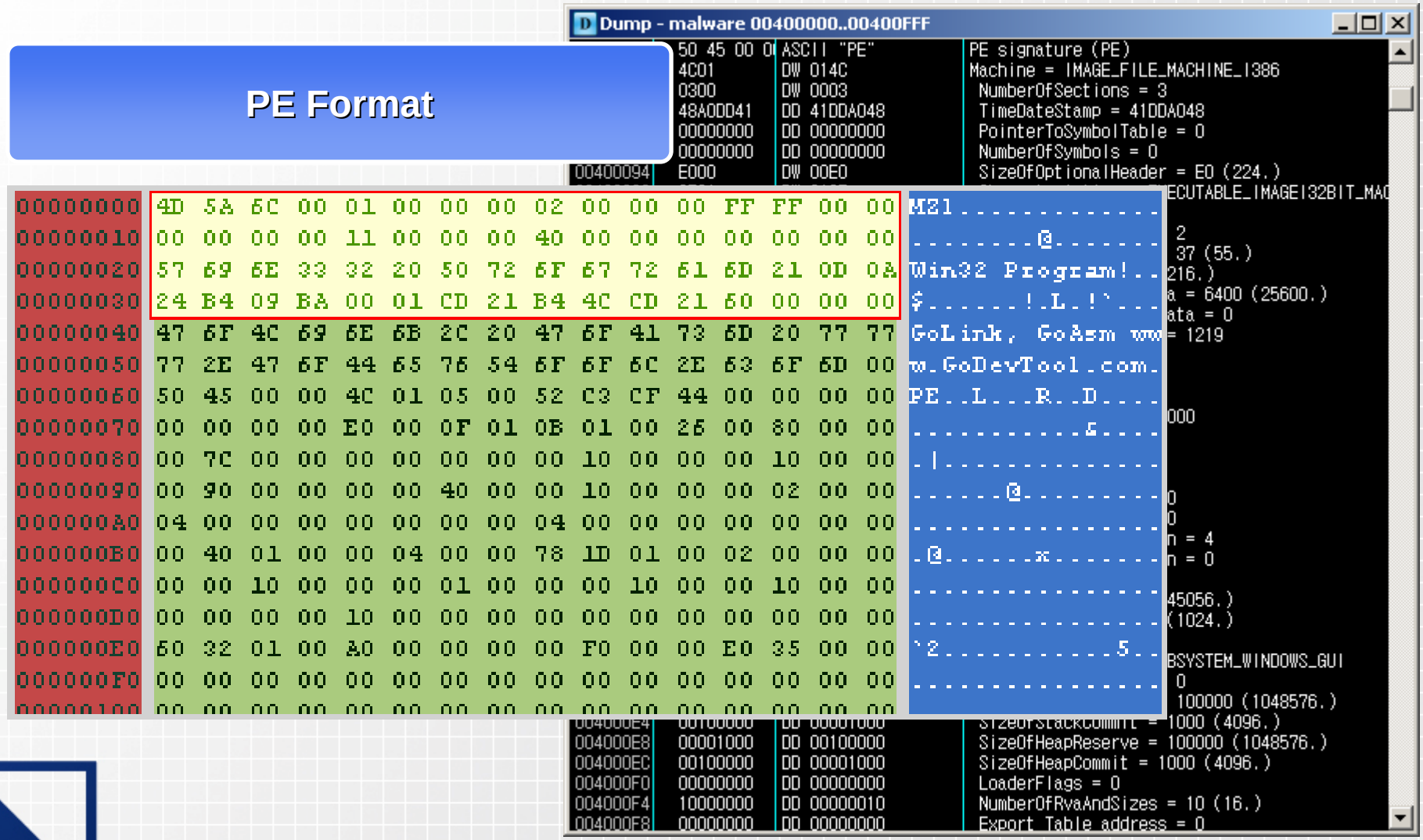
#2. 공부해야할것들

악성코드 분석을 위한 공부과제들



#2. 공부과제

PE Format



#2. 공부과제

Assembly

```
.text:00408870
.text:00408870 __alloca_probe:
.text:00408870                 push    ecx
.text:00408871                 cmp     eax, 1000h
.text:00408876                 lea     ecx, [esp+8]
.text:0040887A                 jb      short loc_408890
.text:0040887C
.text:0040887C loc_40887C:                             ; CODE XREF: .text:0040888E↓j
.text:0040887C                 sub     ecx, 1000h
.text:00408882                 sub     eax, 1000h
.text:00408887                 test    [ecx], eax
.text:00408889                 cmp     eax, 1000h
.text:0040888E                 jnb     short loc_40887C
.text:00408890
.text:00408890 loc_408890:                             ; CODE XREF: .text:0040887A↑j
.text:00408890                 sub     ecx, eax
.text:00408892                 mov     eax, esp
.text:00408894                 test    [ecx], eax
.text:00408896                 mov     esp, ecx
.text:00408898                 mov     ecx, [eax]
.text:0040889A                 mov     eax, [eax+4]
.text:0040889D                 push    eax
.text:0040889E                 retn
```

#2. 공부과제

API

Address	Disassembly	Destination
004088AC	CALL DWORD PTR DS:[<&MSVCRT._onexit>]	MSVCRT._onexit
00407879	CALL DWORD PTR DS:[<&KERNEL32.OutputDebugStringA>]	kernel32.OutputDebugStringA
00407C75	CALL DWORD PTR DS:[<&USER32.PtInRect>]	USER32.PtInRect
00407D15	CALL DWORD PTR DS:[<&USER32.PtInRect>]	USER32.PtInRect
00408934	CALL DWORD PTR DS:[<&MSVCRT.__p__commode>]	MSVCRT.__p__commode
00408926	CALL DWORD PTR DS:[<&MSVCRT.__p__fmode>]	MSVCRT.__p__fmode
004012E0	CALL EBP	MSVCRT.rand
00401B4D	CALL DWORD PTR DS:[<&GDI32.RealizePalette>]	GDI32.RealizePalette
00401C5E	CALL DWORD PTR DS:[<&USER32.ReleaseDC>]	USER32.ReleaseDC
00401CF2	CALL DWORD PTR DS:[<&USER32.ReleaseDC>]	USER32.ReleaseDC
00401DAD	CALL DWORD PTR DS:[<&USER32.ReleaseDC>]	USER32.ReleaseDC
00401E16	CALL DWORD PTR DS:[<&USER32.ReleaseDC>]	USER32.ReleaseDC
00401E34	CALL DWORD PTR DS:[<&USER32.ReleaseDC>]	USER32.ReleaseDC
00401E43	CALL DWORD PTR DS:[<&GDI32.SelectObject>]	GDI32.SelectObject
00401E8A	CALL DWORD PTR DS:[<&GDI32.SelectObject>]	GDI32.SelectObject
00407802	CALL EDI	USER32.SendMessageA
00407816	CALL EDI	USER32.SendMessageA
004079E3	CALL DWORD PTR DS:[<&USER32.SendMessageA>]	USER32.SendMessageA
00401B88	CALL DWORD PTR DS:[<&GDI32.SetDIBitsToDevice>]	GDI32.SetDIBitsToDevice
00408D56	CALL DWORD PTR DS:[<&MSVCRT._setmbcp>]	MSVCRT._setmbcp
00402050	CALL DWORD PTR DS:[<&USER32.SetRect>]	USER32.SetRect
00402829	CALL DWORD PTR DS:[<&KERNEL32.SetThreadPriority>]	kernel32.SetThreadPriority
00407903	CALL DWORD PTR DS:[<&USER32.SetTimer>]	USER32.SetTimer
00408960	CALL DWORD PTR DS:[<&MSVCRT.__setusermatherr>]	MSVCRT.__setusermatherr

```
EL32.GetModuleHandleA>
TR SS:[EBP-300]
EL32.GetModuleFileNameA>
TR SS:[EBP-400]
EL32.GetSystemDirectoryA>
DS:[4040A4]
TR SS:[EBP-400]
405ACA
00402708  .  50            PUSH EAX
00402709  .  E8 FE0A0000   CALL <JMP.&USER32.wsprintfA>
0040270E  .  83C4 10       ADD ESP,10
00402711  .  6A 00         PUSH 0
00402713  .  8D85 00FEFFF  LEA EAX,DWORD PTR SS:[EBP-200]
00402719  .  50            PUSH EAX
0040271A  .  E8 2D0A0000   CALL <JMP.&KERNEL32.CreateDirectoryA>
```

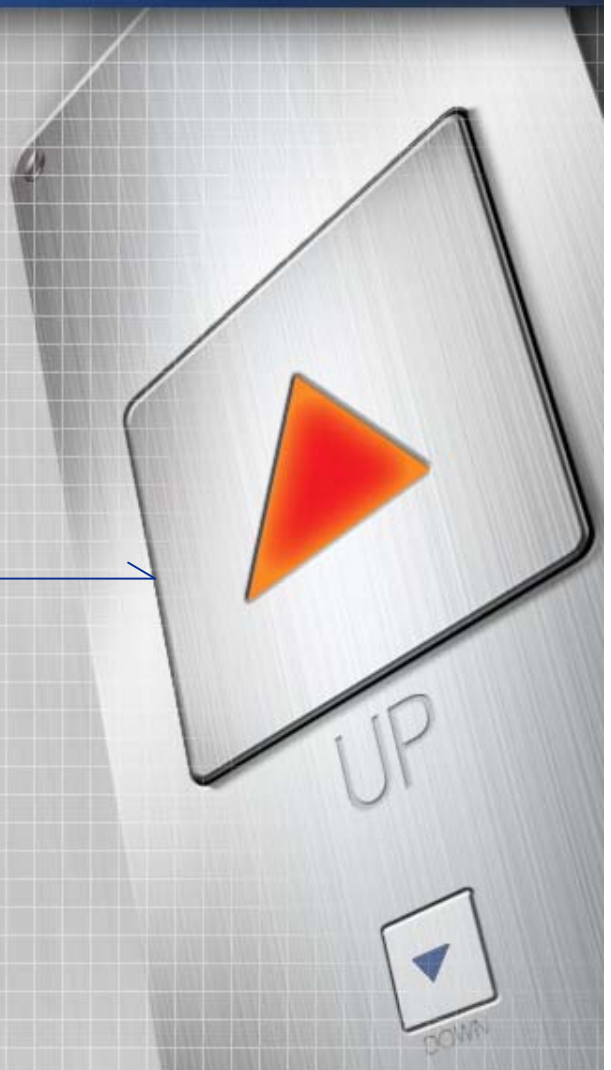
```
pModule = NULL
GetModuleHandleA
BufSize = 100 (256.)

PathBuffer
hModule
GetModuleFileNameA
BufSize = 100 (256.)

Buffer
GetSystemDirectoryA
<%s> = "msview"

<%s>
Format = "%s\%s"

s
wsprintfA

pSecurity = NULL

Path
CreateDirectoryA
```

#2. 공부과제

Programming Language	Manual Unpacking
File Format	Windows Kernel
Memory / Process	SEH
Debugging	...
Calling Conventions / Stack	...

#3. 악성코드의 제작

일반화 되어있는 악성코드들의 기본뼈대



#3. MalwareCode Coding

기본적인 MalwareCode 제작
Sample01.c

```
1   void MalwareCode()
2   {
3       status();
4       target_search();
5       injection();
6       work();
7       jmpEP();
8   }
```

status();

Target_search();

Injection();

Work();

jmpEP();

#3. MalwareCode Coding

기본적인 p2p MalwareCode 제작
Sample02.c

```
1   void MalwareCode02( )
2   {
3       Copy_file( );
4       Run_add( );
5       Search_p2p( );
6       copy_mfile( );
7   }
```

Copy_file();

Run_add();

Search_p2p();

Copy_mfile();

#3. MalwareCode Coding

기본적인 IRC MalwareCode 제작
Sample03.c

```
1   void MalwareCode03( )
2   {
3       Copy_file( );
4       Run_add( );
5       Run_IRC( );
6   }
```

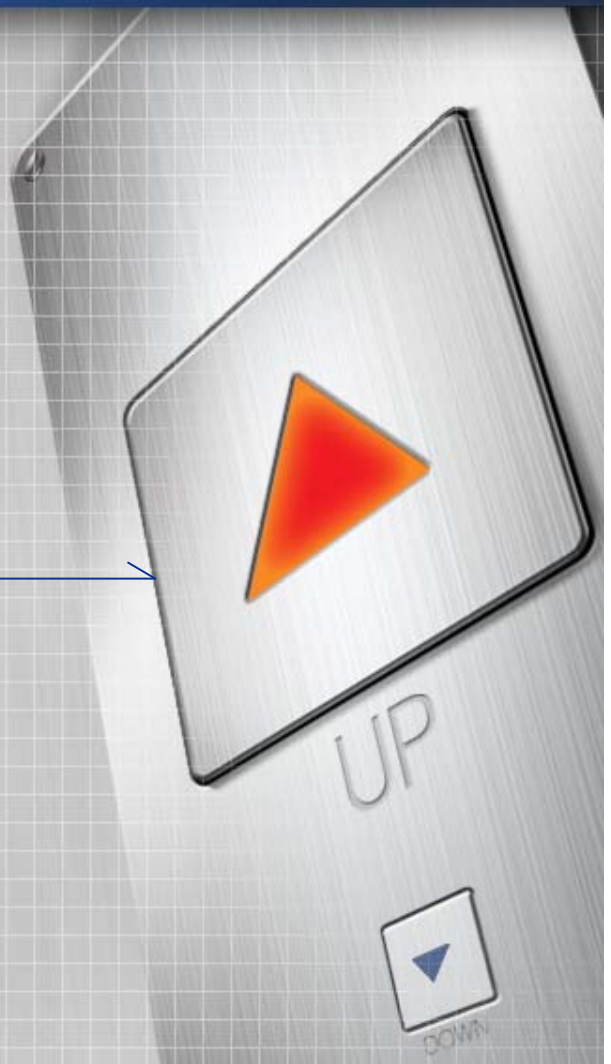
Copy_file();

Run_add();

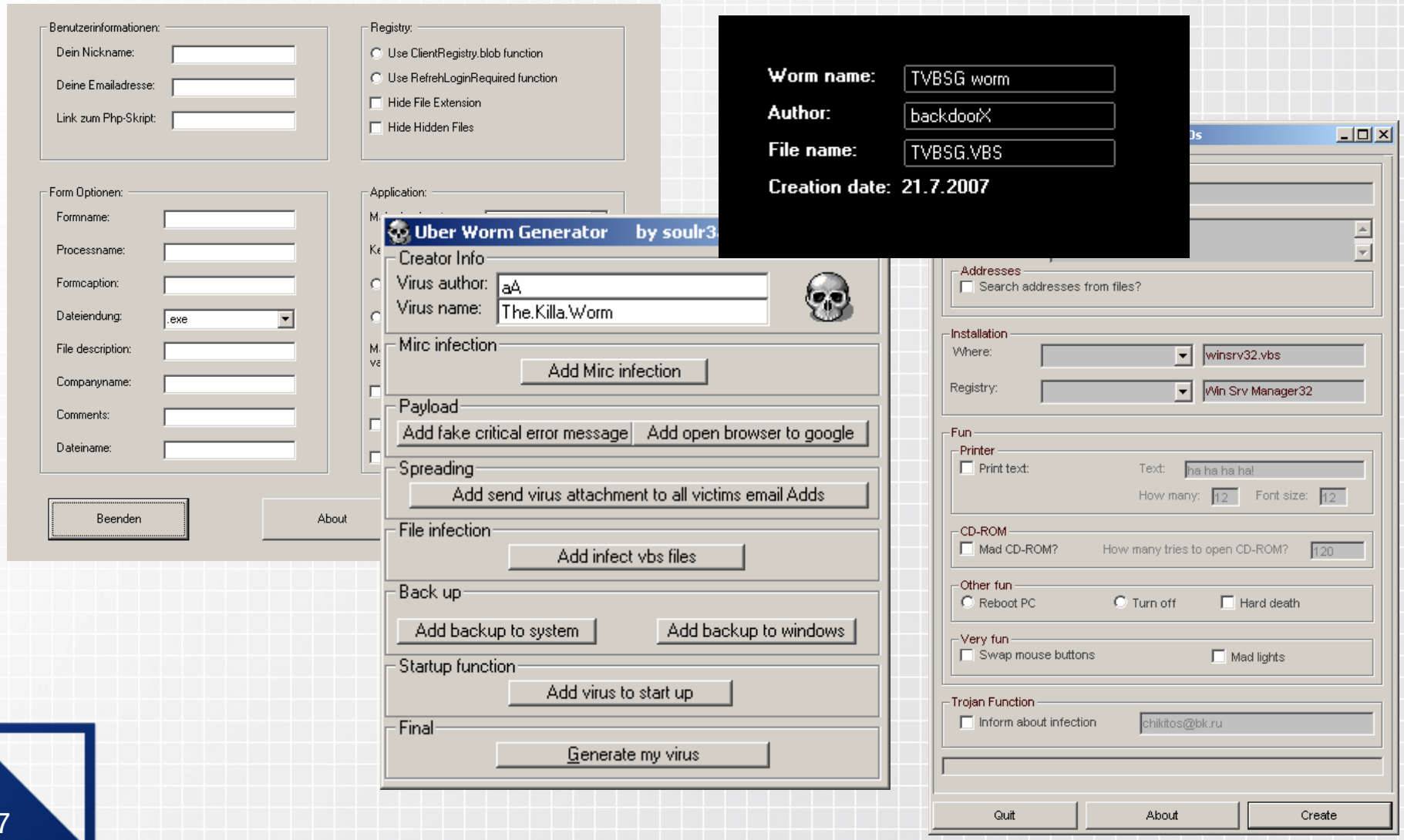
Run_IRC();

#4. 악성코드 제작 툴

악성코드 제작툴도 발전하고 있다.



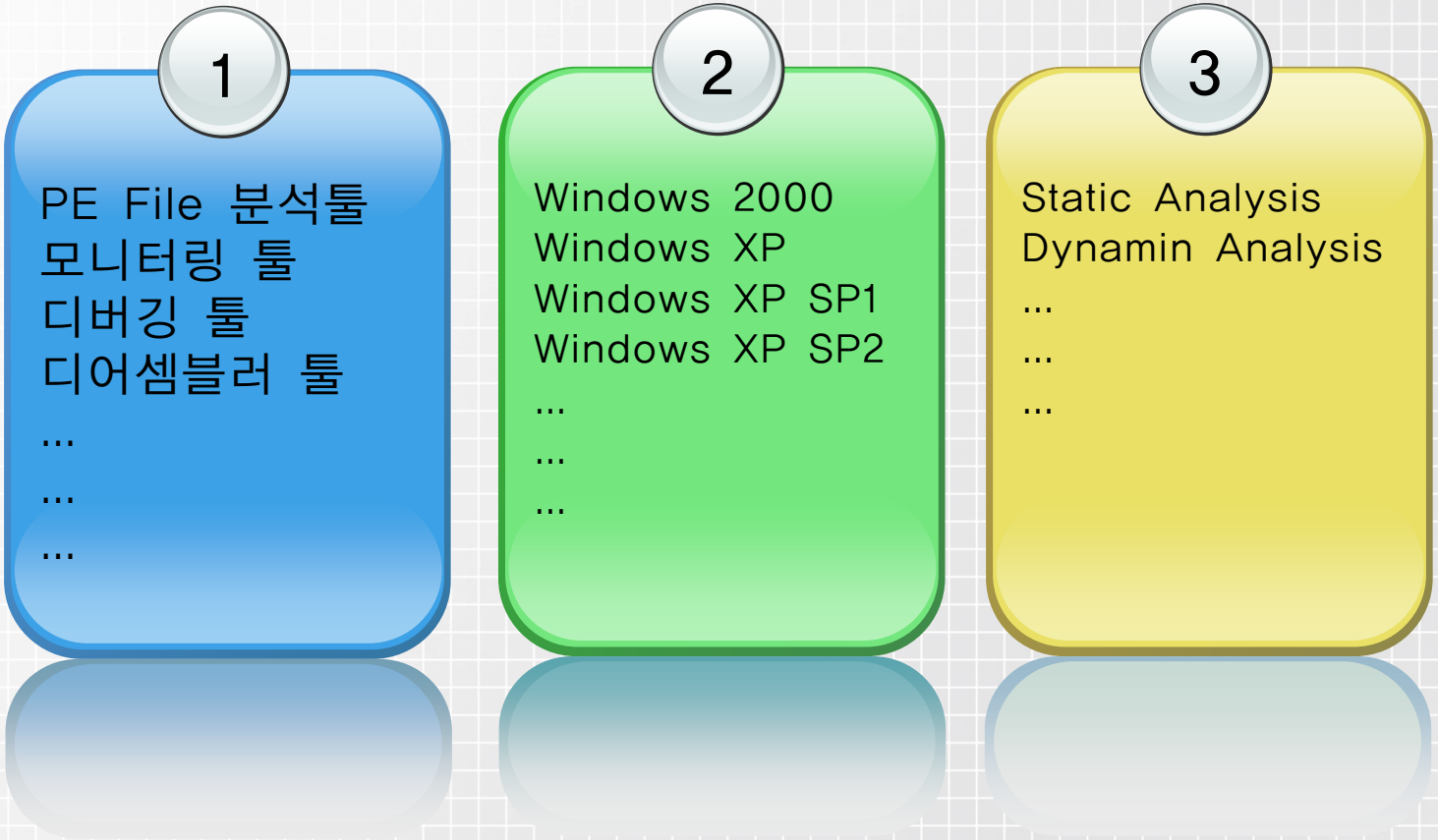
#4. 악성코드 제작툴


Benutzerinformationen:
Dein Nickname:
Deine Emailadresse:
Link zum Php-Skript:
Registry:
Use ClientRegistry.blob function
Use RefrehLoginRequired function
Hide File Extension
Hide Hidden Files
Form Optionen:
Formname:
Processname:
Formcaption:
Dateiendung:
.exe
File description:
Companyname:
Comments:
Dateiname:
Application:
Beenden
About
Worm name: TVBSG worm
Author: backdoorX
File name: TVBSG.VBS
Creation date: 21.7.2007
Uber Worm Generator by soulr3
Creator Info
Virus author: aA
Virus name: The.Killa.Worm
Mirc infection
Add Mirc infection
Payload
Add fake critical error message
Add open browser to google
Spreading
Add send virus attachment to all victims email Adds
File infection
Add infect vbs files
Back up
Add backup to system
Add backup to windows
Startup function
Add virus to start up
Final
Generate my virus
Addresses
Search addresses from files?
Installation
Where:
winsrv32.vbs
Registry:
Win Srv Manager32
Fun
Printer
Print text:
Text:
ha ha ha ha!
How many:
12
Font size:
12
CD-ROM
Mad CD-ROM?
How many tries to open CD-ROM?
120
Other fun
Reboot PC
Turn off
Hard death
Very fun
Swap mouse buttons
Mad lights
Trojan Function
Inform about infection
chikitos@bk.ru
Quit
About
Create

#5. 악성코드 분석 랩 구축

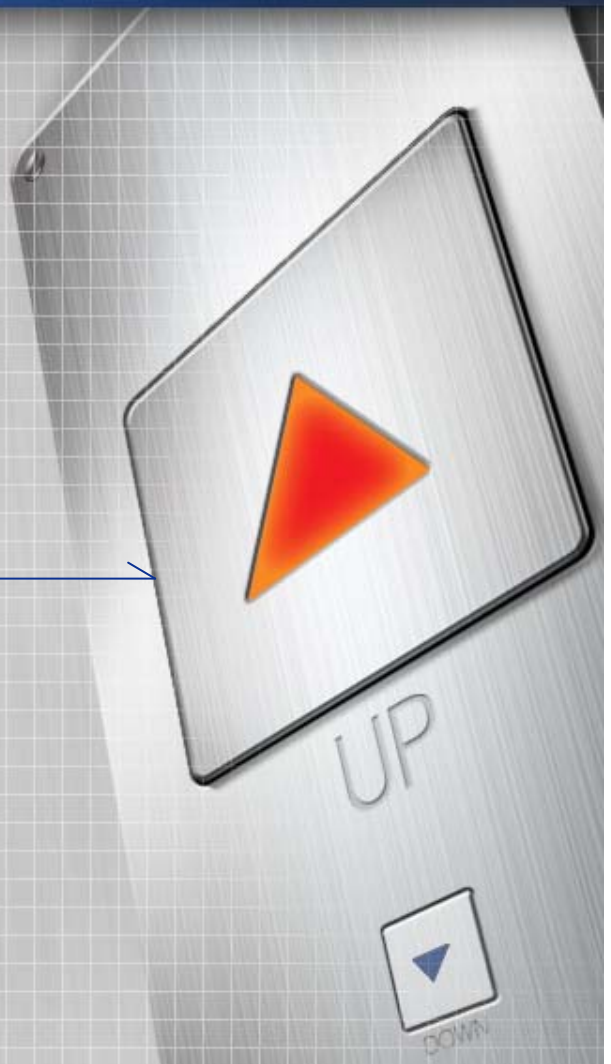
악성코드 분석을 위한 다양한 환경 구축

#5. 악성코드 분석 랩 구축

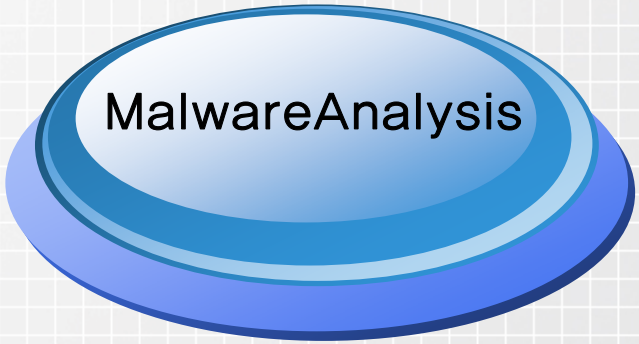

1
PE File 분석툴
모니터링 툴
디버깅 툴
디어셈블러 툴
...
...
...
2
Windows 2000
Windows XP
Windows XP SP1
Windows XP SP2
...
...
...
3
Static Analysis
Dynamin Analysis
...
...
...

#6. 악성코드 분석

MalwareCode Sample Analysis



#6. 악성코드 분석



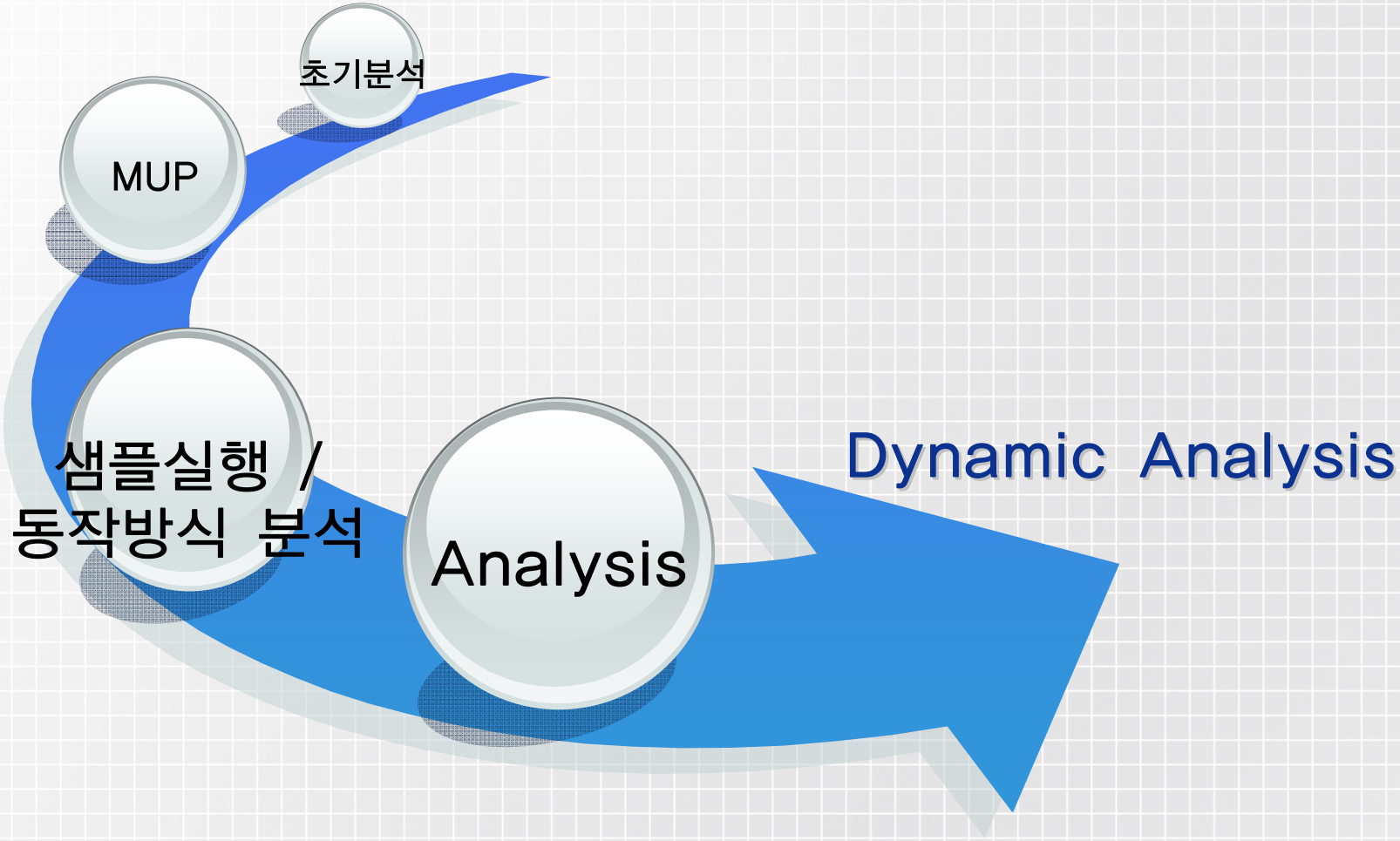
Static Analysis

-정밀조사가 필요한 핵심부분 세밀한 코드분석
-디버깅&코드 분석

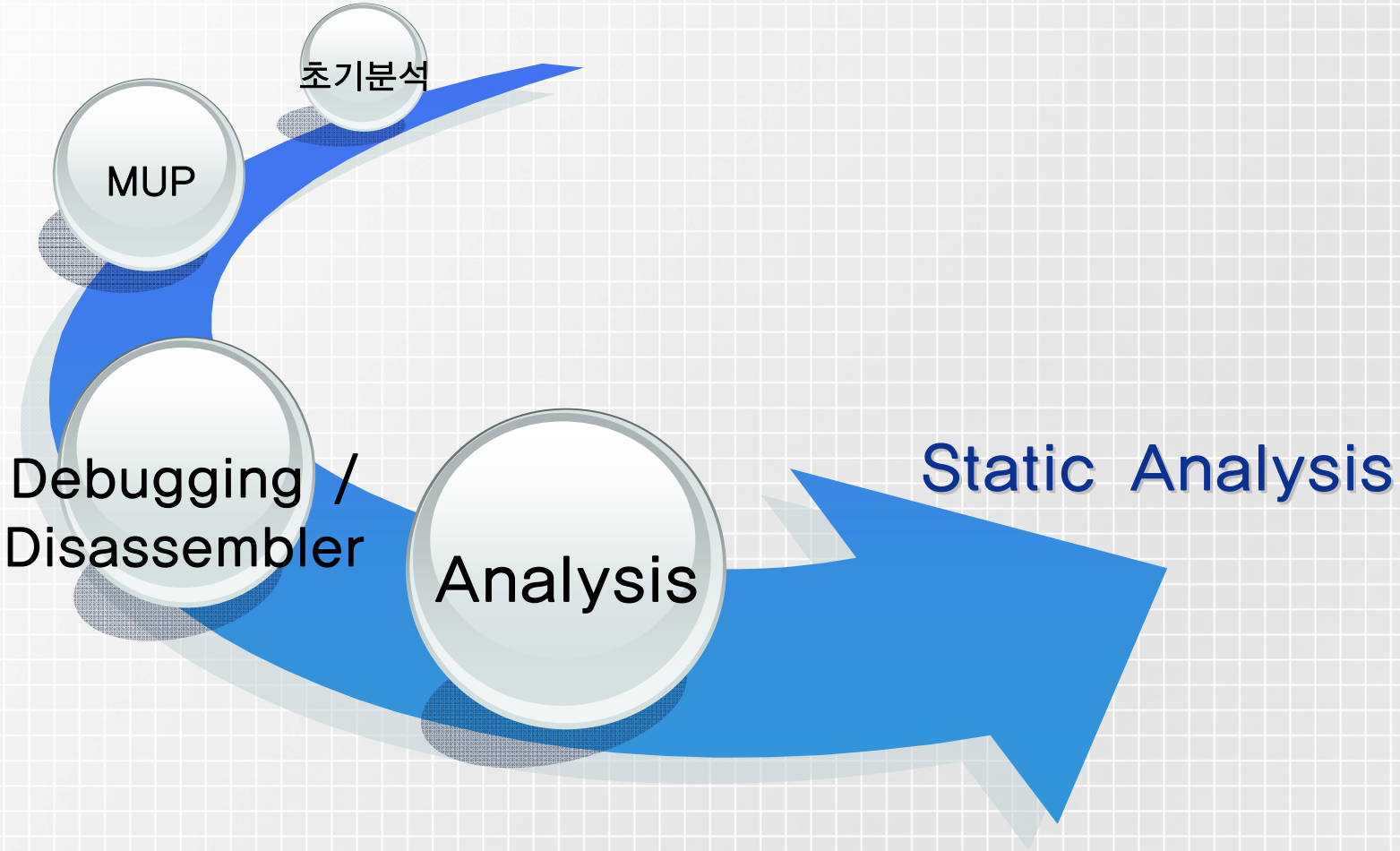
Dynamic Analysis

-실행을 통한 분석
-그에따른 환경구축
-시나리오 파악
-디버깅&코드 분석

#6. 악성코드 분석

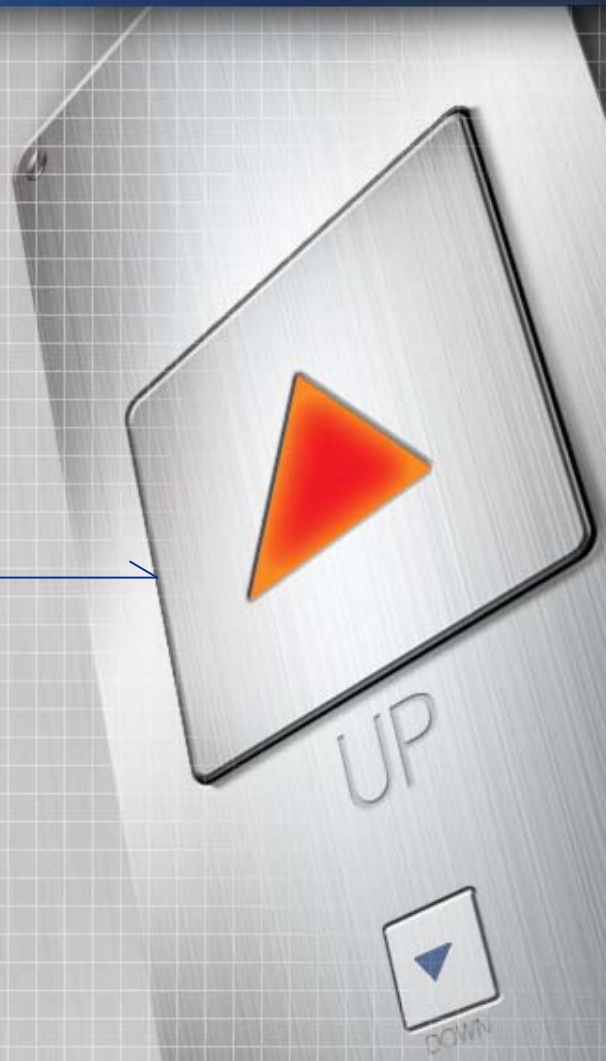


#6. 악성코드 분석


초기분석
MUP
Debugging / Disassembler
Analysis
Static Analysis

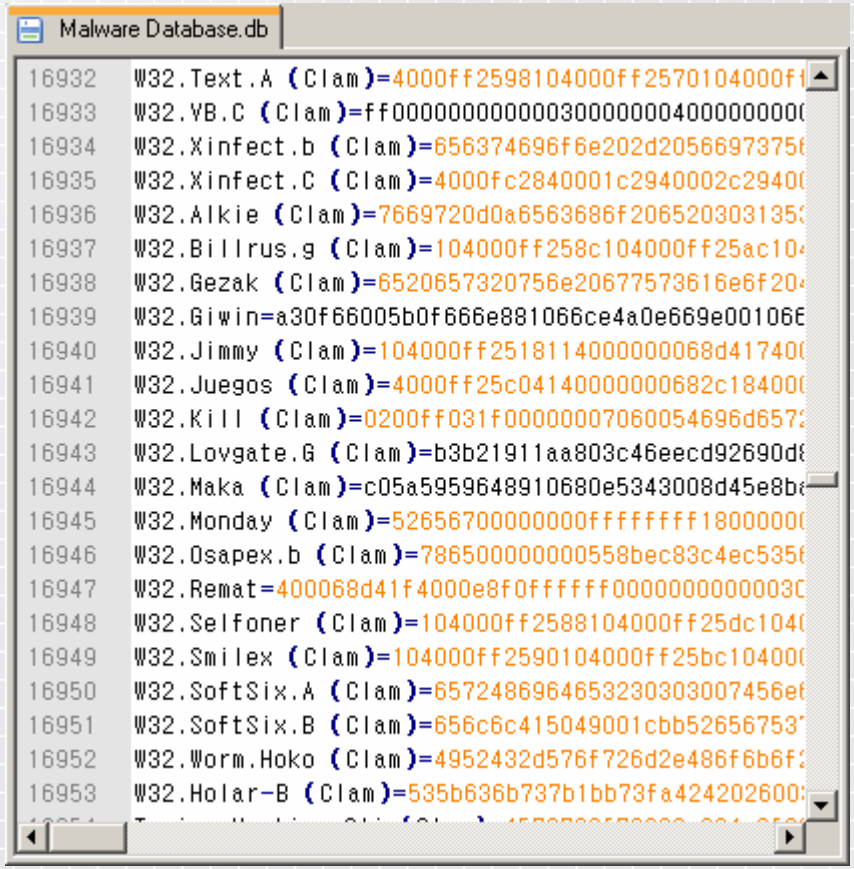
#7. AntiVirus Program

AntiVirus Program



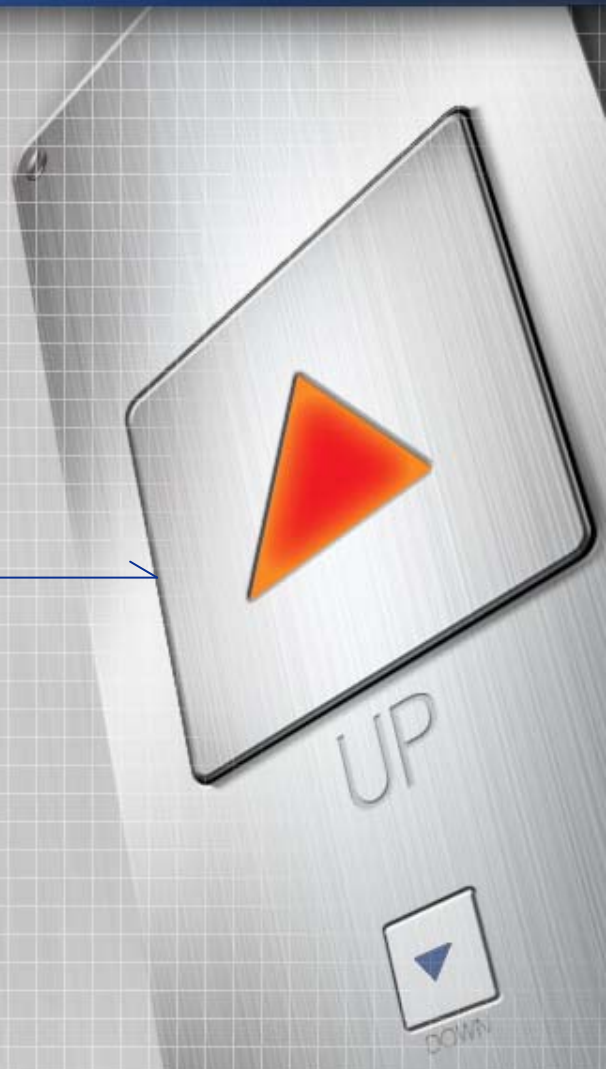
#7. AntiVirus Program

어셈러브 AntiVirus 2007 by Certlab



#8. 악성코드 탐지기법

악성코드를 탐지하는 다양한 기법들



#8. 악성코드 탐지기법

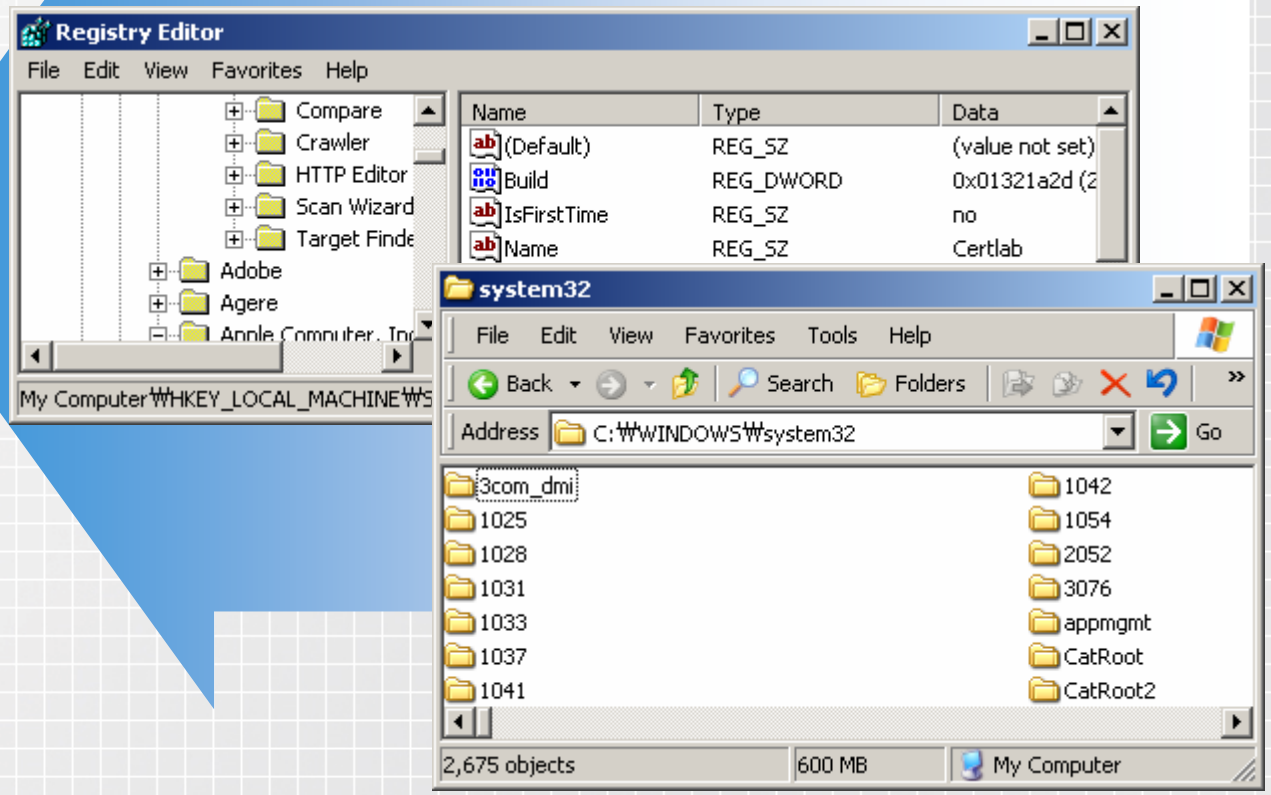
File base Detection Tech

Malware Database.db

```
17031  Trojan.Millenium.B.Server (Clam)=f000111000ff035500000003060043686563 6b310005012900
17032  W32.MircNew (Clam)=6d030000000400 6d495243000d0016006d4952432076362e3020333262697420
17033  W32.MircNew-1 (Clam)=ff020000110100ff03540000000203006c6162000105000078 00af147701ff
17034  W32.Mix.1852 (Clam)=e8000000005d83ed08db452ddb4531dee9db55358b4535b9c00100005531853
17035  W32.Mockoder.1120 (Clam)=c560040000bb3480112e8db5c4fbffff8bfeb90f010000adf7d0d3c033
17036  W32.Mogul.6800 (Clam)=60e8000000005d81ed061040008d952f1040008dbd97274000813a5053515
17037  W32.Morgoth.2560 (Clam)=fe4f66b85c7766ab66b8696e66ab66b82e6566ab66b8786566ab5e6a006
17038  W32.MTX (Clam)=536f6674776172652070726f7669646520627920 5b4d41545269585d205658207465
17039  W32.Munfor.D (Clam)=ab08ff0360d75533e9a46a7c03e02548d3740d5f87859293ae7ea9034d75cb6
17040  W32.Myparty.B (Clam)=537ff3ffff756e4d6f6e5475655765645468754672695361744a616e466562
17041  W32.Mystery.2560 (Clam)=397ab23e45274c274cb2d88154 0d4d1d274d1db2d8a9540d4db2d8c9540
17042  W32.Navidad.B=b28443685213667fe6a199306dd7d12af649015755e48b3659f50a0c325726656453e
17043  W32.Navidad.e (Clam)=08b56dea4682326762422b165997cbdb401c02d24340a0996520992a9da121
17044  W32.NgVck.D1=770d5753ff951515400083f80075c053ff951915400081c40004000061c36080bd8c17
```

#8. 악성코드 탐지기법

Heuristic Detection Tech



#8. 악성코드 탐지기법

Generic Detection Tech #1

```
Address   Hex dump          Disassembly                              Comment
004089E4  .  8D45 A4        LEA EAX,DWORD PTR SS:[EBP-5C]
004089E7  .  50             PUSH EAX                                 pStartupinfo
004089E8  .  FF15 3CA0400(  CALL DWORD PTR DS:[<&KERNEL32.GetStartu  GetStartupInfoA
004089EE  .  F645 D0 01     TEST BYTE PTR SS:[EBP-30],1
004089F2  .v 74 11          JE SHORT CodeEngn.00408A05
004089F4  .  0FB745 D4      MOVZX EAX,WORD PTR SS:[EBP-2C]
004089F8  .v EB 0E          JMP SHORT CodeEngn.00408A08
004089FA  >  803E 20        CMP BYTE PTR DS:[ESI],20
004089FD  .^ 76 D8          JBE SHORT CodeEngn.004089D7
004089FF  .  46             INC ESI
00408A00  .  8975 8C        MOV DWORD PTR SS:[EBP-74],ESI
00408A03  .^ EB F5          JMP SHORT CodeEngn.004089FA
00408A05  >  6A 0A          PUSH 0A
00408A07  .  58             POP EAX
00408A08  >  50             PUSH EAX
00408A09  .  56             PUSH ESI
00408A0A  .  53             PUSH EBX
00408A0B  .  53             PUSH EBX                                 pModule
00408A0C  .  FF15 40A0400(  CALL DWORD PTR DS:[<&KERNEL32.GetModule  GetModuleHandleA
00408A12  .  50             PUSH EAX
00408A13  .  E8 0A030000    CALL CodeEngn.00408D22
```

#8. 악성코드 탐지기법

Generic Detection Tech #2

```
Address   Hex dump      Disassembly
004089E4  .  8D45 A4    LEA EAX,DWORD PT
004089E7  .  50         PUSH EAX
004089E8  .  FF15 3CA0400( CALL DWORD PTR D
004089EE  .  F645 D0 01 TEST BYTE PTR SS
004089F2  .v 74 11      JE SHORT CodeEng
004089F4  .  0FB745 D4  MOVZX EAX,WORD P
004089F8  .v EB 0E      JMP SHORT CodeEn
004089FA  >  803E 20    CMP BYTE PTR DS
004089FD  .^ 76 D8      JBE SHORT CodeE
004089FF  .  46         INC ESI
00408A00  .  8975 8C    MOV DWORD PTR S
00408A03  .^ EB F5      JMP SHORT CodeE
00408A05  >  6A 0A      PUSH 0A
00408A07  .  58         POP EAX
00408A08  >  50         PUSH EAX
00408A09  .  56         PUSH ESI
00408A0A  .  53         PUSH EBX
00408A0B  .  53         PUSH EBX
00408A0C  .  FF15 40A0400( CALL DWORD PTR D
00408A12  .  50         PUSH EAX
00408A13  .  E8 0A030000 CALL CodeEngn.00
```

```
Address   Hex dump      Disassembly
004089E4  .  8D45 A4    LEA EAX,DWORD PTR SS:[EBP-5C]
004089E7     53         PUSH EBX
004089E8  .  FF15 3CA0400( CALL DWORD PTR DS:[<&KERNEL32.GetStartu
004089EE  .  F645 D0 01 TEST BYTE PTR SS:[EBP-30],1
004089F2  v  75 11      JNZ SHORT CodeEngn.00408A05
004089F4  .  0FB745 D4  MOVZX EAX,WORD PTR SS:[EBP-2C]
004089F8  .v EB 0E      JMP SHORT CodeEngn.00408A08
004089FA  >  803E 20    CMP BYTE PTR DS:[ESI],20
004089FD  .^ 76 D8      JBE SHORT CodeEngn.004089D7
004089FF     47         INC EDI
00408A00  .  8975 8C    MOV DWORD PTR SS:[EBP-74],ESI
00408A03  .^ EB F5      JMP SHORT CodeEngn.004089FA
00408A05     6A 0B      PUSH 0B
00408A07  .  58         POP EAX
00408A08     52         PUSH EDX
00408A09  .  56         PUSH ESI
00408A0A  .  53         PUSH EBX
00408A0B  .  53         PUSH EBX
00408A0C  .  FF15 40A0400( CALL DWORD PTR DS:[<&KERNEL32.GetModule
00408A12  .  50         PUSH EAX
00408A13  .  E8 0A030000 CALL CodeEngn.00408D22
```

Reference

Reference

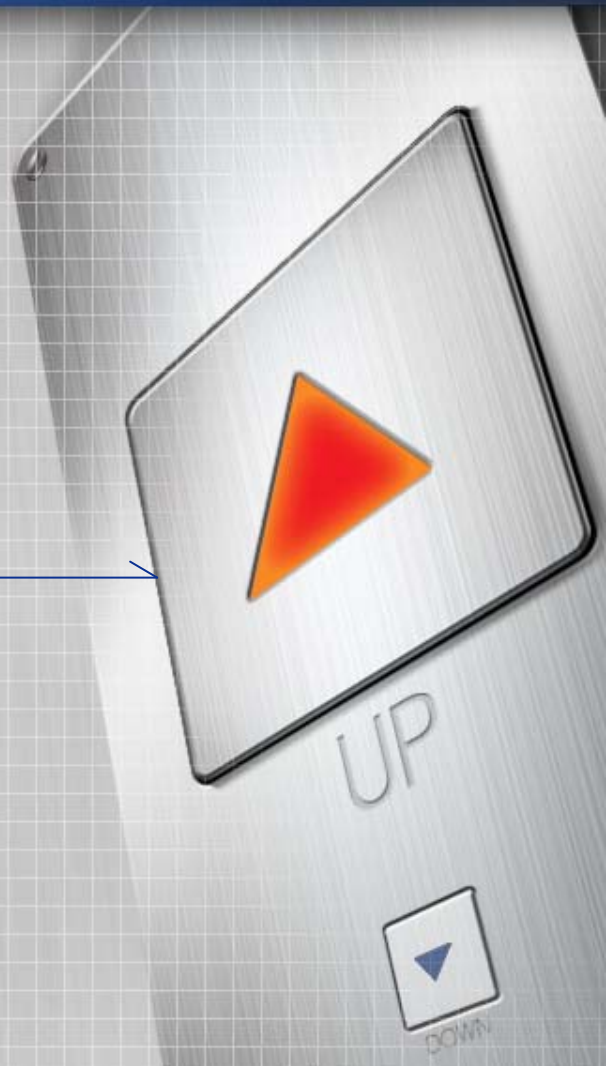
어셈블리어 개발자그룹
www.asmlove.co.kr

다양한 악성코드 탐지기법
www.ahnlab.co.kr

국가정보원 - Monthly 사이버 시큐리티 2007년 7월호
www.ncsc.go.kr

Manual Unpacking
www.icrack.co.kr

감사합니다.



www.certlab.org
www.asmlove.co.kr

이강석 / **certlab@gmail.com**