

Forensic Investigation & Malware Analysis against Targeted Attack using Free Tools

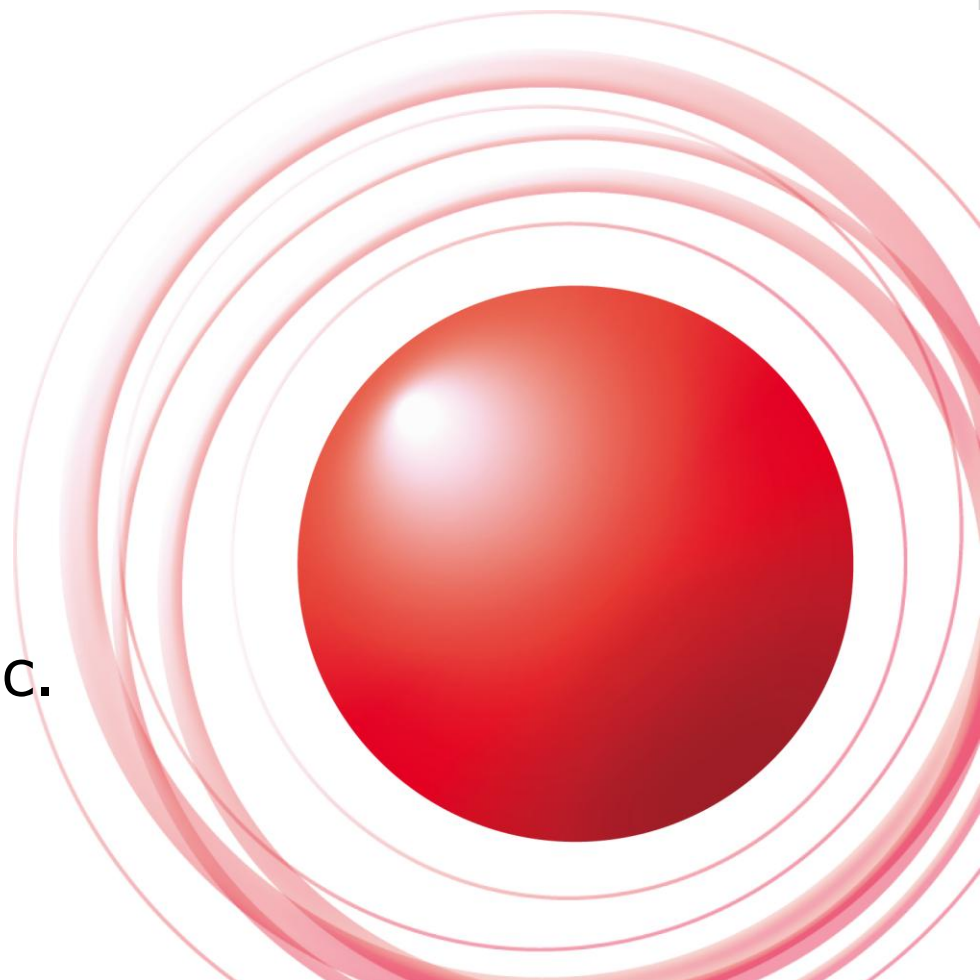


2013/1/30

IIJ-SECT

Internet Initiative Japan Inc.

Ongoing Innovation

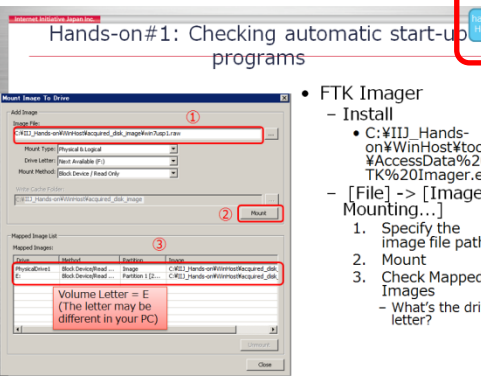


Setup Instructions

- Copy the files in USB flash memory
 - Copy “IIJ_Hands-on” to “C:\” of your laptop (Host OS)
 - leaked_file
 - 7z file including documents leaked during this incident
 - WinHost
 - Data and tools used on host OS
 - WinVM
 - Data and tools used on Windows VM
 - Documents
 - Hands-on PDF and its answer PDFs (password protected)
 - references for forensic investigation
 - NOTICE: “\” stands for backslash in Japanese OS
- Extract the disk image
 - C:\IIJ_Hands-on\WinHost\acquired_disk_image\win7usp1.zip
 - Vista and 7 users: Use “Extract all files” of OS function
 - XP users: Install 7-Zip and use it
 - C:\IIJ_Hands-on\WinHost\tools\7z920.exe
 - **DELETE** the image after your hands-on!!

IMPORTANT: Hands-on Mark

Hands-on#1: Checking automatic start-up programs

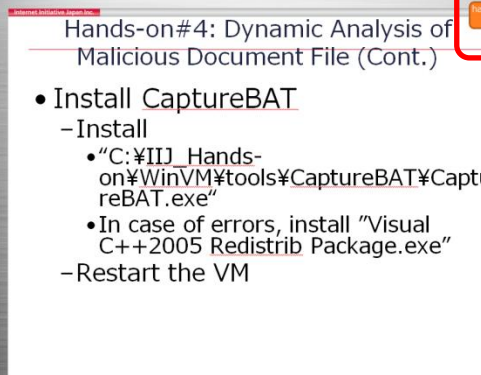


- FTK Imager
 - Install
 - C:\I1J_Hands-on\WinHost\tools\AccessData%20FTK%20Imager.exe
 - [File] -> [Image Mounting...]
 1. Specify the image file path
 2. Mount
 3. Check Mapped Images
 - What's the drive letter?

hands-on Host OS

Work on something in your host OS

Hands-on#4: Dynamic Analysis of Malicious Document File (Cont.)

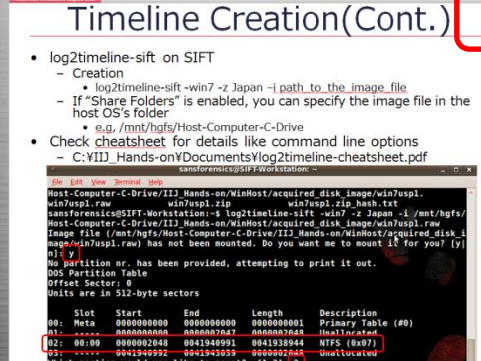


- Install CaptureBAT
 - Install
 - "C:\I1J_Hands-on\WinVM\tools\CaptureBAT\CaptureBAT.exe"
 - In case of errors, install "Visual C++2005 Redistrib Package.exe"
 - Restart the VM

hands-on VM

Work on something in your guest OS

Timeline Creation(Cont.)



- log2timeline-sift on SIFT
 - Creation
 - log2timeline-sift -win7 -z Japan -i path_to_the_image_file
 - If "Share Folders" is enabled, you can specify the image file in the host OS's folder
 - e.g. /mnt/hgfs/Host-Computer-C-Drive
 - Check cheatsheet for details like command line options
 - C:\I1J_Hands-on\Documents\log2timeline-cheatsheet.pdf

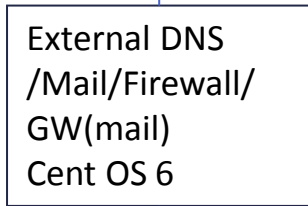
Just look

Scenario

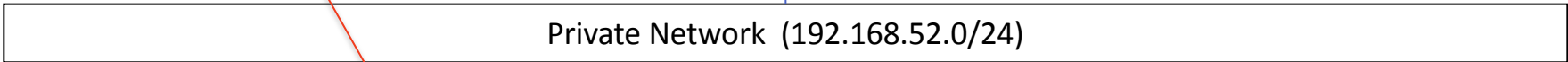
- You are a member of CSIRT at a certain company
- You were externally-pointed out information of your company leaked
 - The leaked private documents were uploaded on the Internet
 - The file name is “a.7z”
- You identified the suspicious PC from the following evidences
 - file sharing server’s event logs
 - interview outcome of clients
- That’s why you decided to examine the PC

Network Configuration

The file server was accessed from Client A using toshi (executive) account. Okita never knows the password.

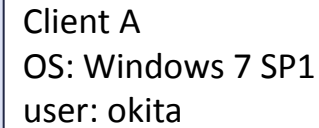


.32

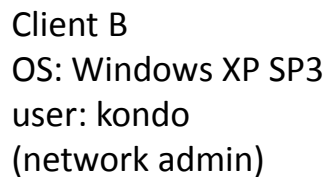


Domain : shinsen-group

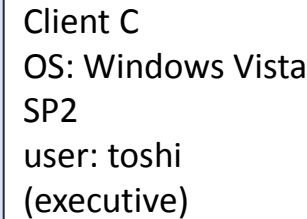
.50



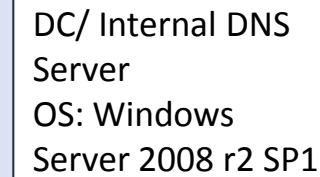
.52



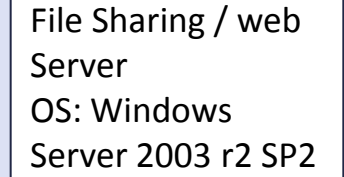
.51



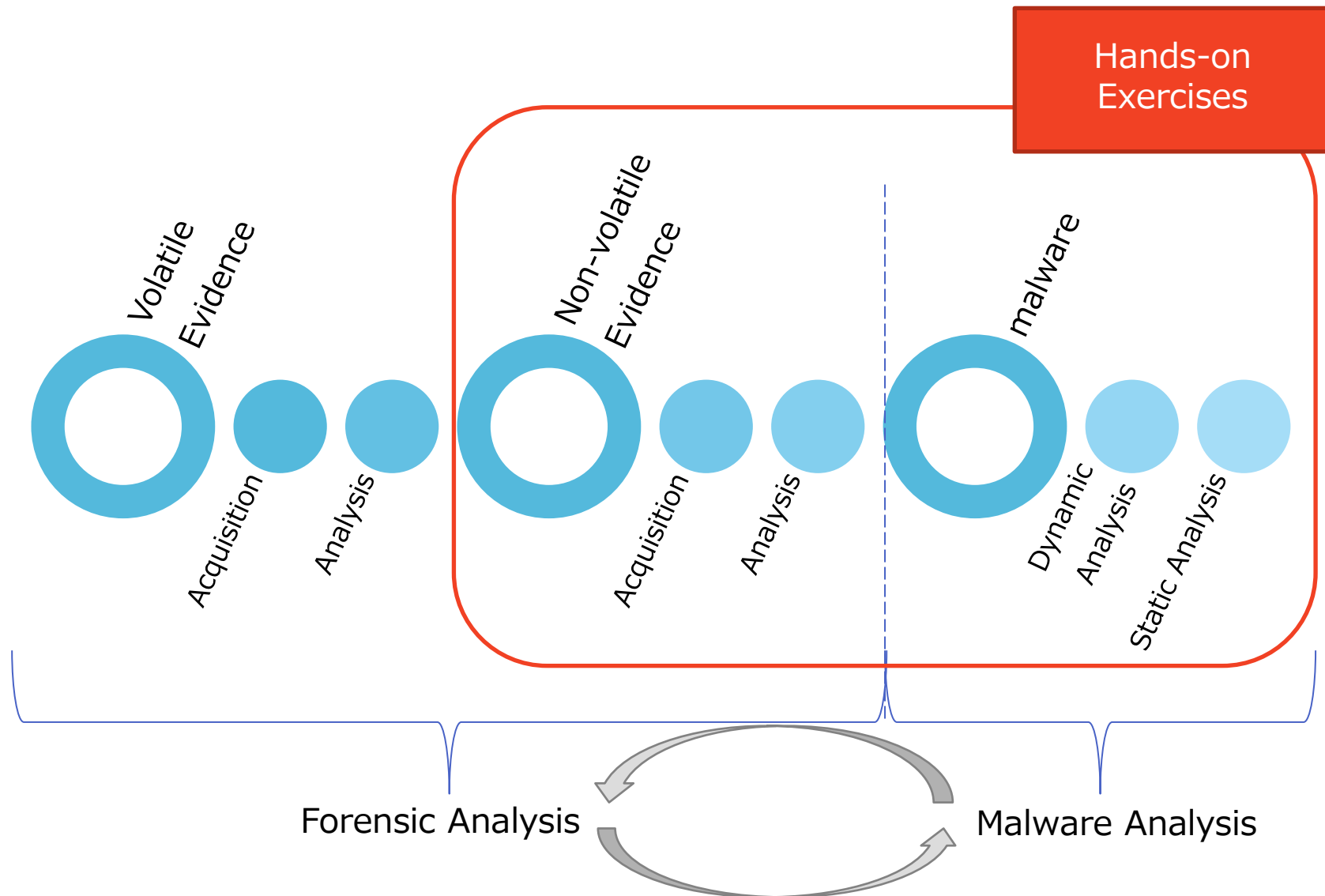
.33



.34



Flow of Incident Response



Analysis in the Case

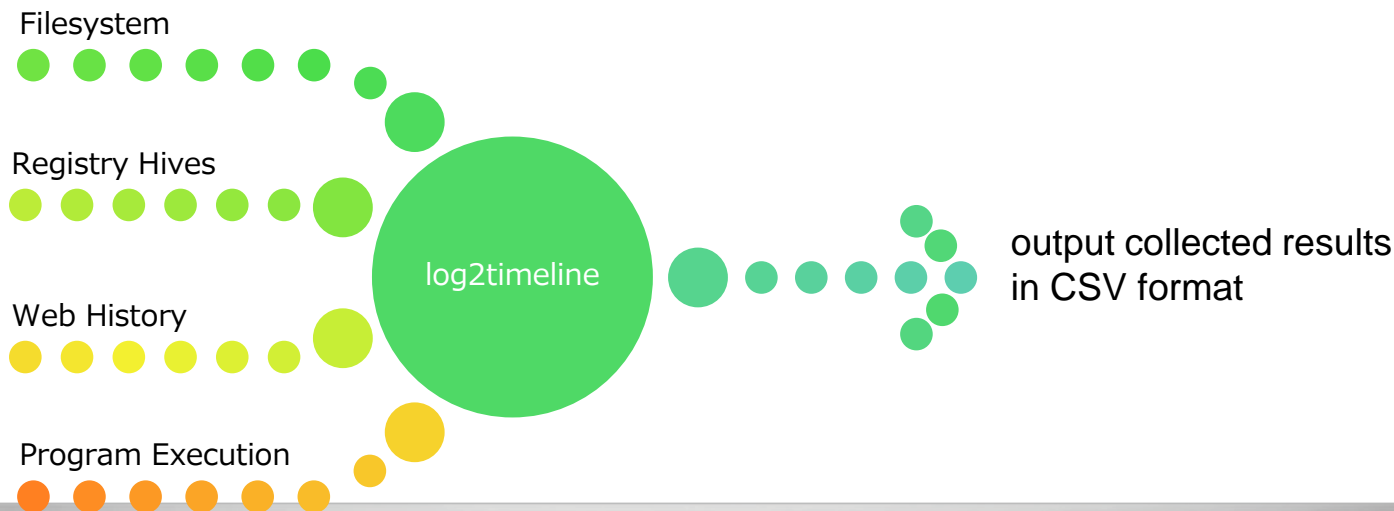
- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Timeline Creation

- Create Timeline using log2timeline on SANS SIFT Workstation
 - Put together various timestamps (e.g., filesystem, registry) into one output form
- Narrow down time period of malware infection by using some information
 - Find malware infection signs (e.g., start-up locations, execution history caches)
 - Use external information (e.g., malicious URLs, IPS logs)
- Check the time period
 - Trace back from the period for infection root cause
 - Follow malicious activities after the period



Timeline Creation(Cont.)

- log2timeline-sift on SANS SIFT Workstation
 - Creation
 - log2timeline-sift -win7 -z Japan -i path_to_the_image_file
 - If "Share Folders" is enabled, you can specify the image file in the host OS's folder
 - e.g, /mnt/hgfs/Host-Computer-C-Drive
- Check cheatsheet for details like command line options
 - C:¥IIJ_Hands-on¥Documents¥log2timeline-cheatsheet.pdf

```
sansforensics@SIFT-Workstation: ~
File Edit View Terminal Help
Host-Computer-C-Drive/IIJ_Hands-on/WinHost/acquired_disk_image/win7uspl.
win7uspl.raw          win7uspl.zip          win7uspl.zip_hash.txt
sansforensics@SIFT-Workstation:~$ log2timeline-sift -win7 -z Japan -i /mnt/hgfs/
Host-Computer-C-Drive/IIJ_Hands-on/WinHost/acquired_disk_image/win7uspl.raw
Image file (/mnt/hgfs/Host-Computer-C-Drive/IIJ_Hands-on/WinHost/acquired_disk_i
mage/win7uspl.raw) has not been mounted. Do you want me to mount it for you? [y]
n]: y
No partition nr. has been provided, attempting to print it out.
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors
```

	Slot	Start	End	Length	Description
00:	Meta	0000000000	0000000000	0000000001	Primary Table (#0)
01:	-----	0000000000	0000002047	0000002048	Unallocated
02:	00:00	0000002048	0041940991	0041938944	NTFS (0x07)
03:	-----	0041940992	0041943039	0000002048	Unallocated

```
Which partion would you like to mount?: [1-3]: 2
sudo /bin/mount -o ro,loop,show sys files,streams interface=windows,offset=10485
```

Timeline Creation(Cont.)

- log2timeline-sift on SANS SIFT Workstation
 - filter by date range
 - l2t_process -b /cases/timeline-output-folder/ImageFileName_bodyfile.txt
StartDate (..EndDate) > path_to_output_CSV

```
sansforensics@SIFT-Workstation:~$ l2t_process -b /cases/timeline-output-folder/w
in7usp1_bodyfile.txt 09-01-2012 > /cases/timeline-output-folder/20120901win7usp1
_bodyfile.csv
There are 58 that fall outside the scope of the date range, yet show sign of pos
sible timestomping.
Would you like to include them in the output? [Y/n] y

Total number of events that fit into the filter (got printed) = 150381
Total number of duplicate entries removed = 30743
Total number of events skipped due to whitelisting = 0
Total number of events skipped due to keyword filtering = 0
Total number of processed entries = 514036
Run time of the tool: 15 sec
```

Timeline Creation(Cont.)

- log2timeline-sift on SANS SIFT Workstation
 - Check source types of entries extracted from CSV
 - `awk -F, '{print $6;}' path_to_the_csv_file | grep -v sourcetype | sort | uniq`
 - v2.13 drops event log entries!
 - due to Japanese OS image?

```
sansforensics@SIFT-Workstation:~$ awk -F, '{ print $6 }' /cases/timeline-output-  
folder/20120901win7usp1_bodyfile.csv | grep -v sourcetype | sort | uniq  
Application  
Chrome History  
Deleted Registry  
EXIF metadata  
FileExts key  
Firefox Cache  
Flash Cookie  
Internet Explorer  
Map Network Drive MRU key  
Microsoft-Windows-Application-Experience/Program-Inventory
```

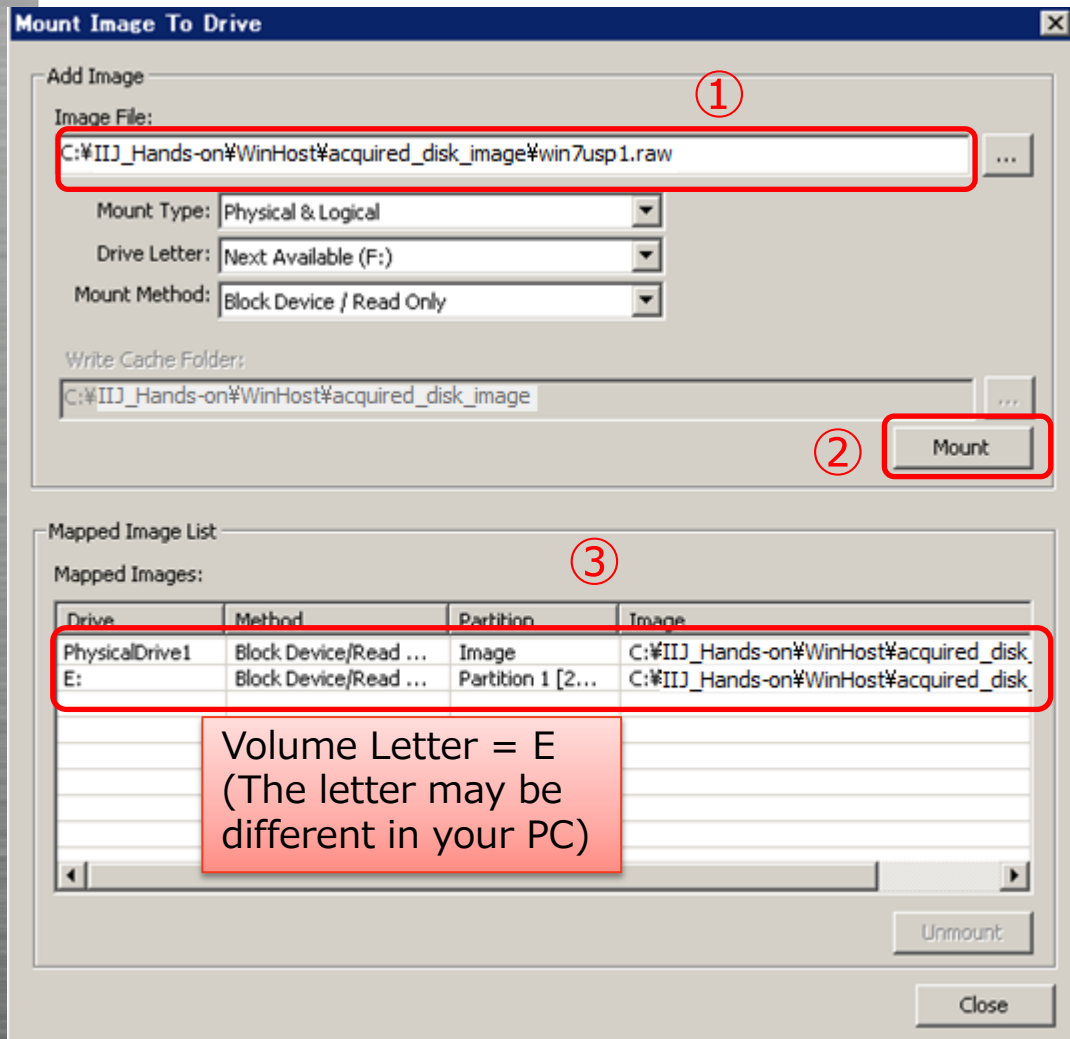
Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Checking automatic start-up programs

- malware adds its automatic start-up setting in order to run after reboot or logon
 - Checking the configurations is one of the most effective methods to detect malware
- AutoRuns
 - Display all-in output of startup settings
 - e.g., registry Run keys, services, BHOs, etc..
 - Not only live systems, but offline system volumes can be examined
 - Use “Analyze Offline System” function
- FTK Imager
 - Mount disk images with read-only

Hands-on#1: Checking automatic start-up programs

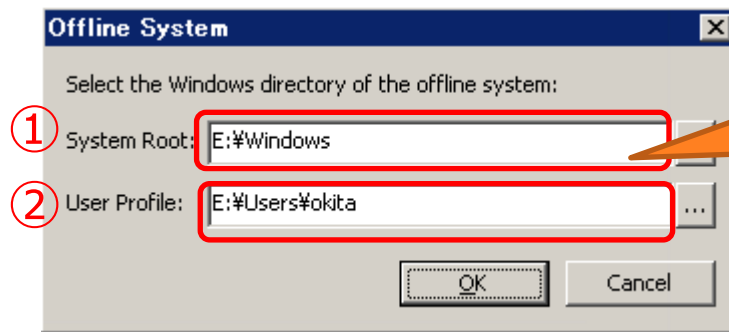
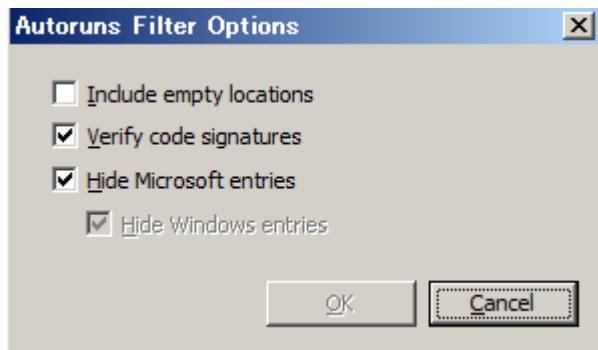


- FTK Imager
 - Install
 - C:\¥IJJ_Hands-on¥WinHost¥tools¥AccessData%20FTK%20Imager.exe
 - [File] -> [Image Mounting...]
 1. Specify the image file path
 2. Mount
 3. Check Mapped Images
 - What's the drive letter?

Hands-on#1: Checking automatic start-up programs (Cont.)

- AutoRuns

- Extract "C:¥IIJ_Hands-on¥WinHost¥tools¥Autoruns.zip"
- Run autoruns.exe **as administrator**
 - Check the window name (admin user name is displayed?)
- Select [Options] -> [Filter Options] to reduce some noise
- [File] -> [Analyze Offline System...]
 1. System Root = Mounted_Image_Volume_Letter:¥Windows
 2. User Profile = Mounted_Image_Volume_Letter:¥Users¥okita



Check your volume letter in FTK Imager

Hands-on#1: Checking automatic start-up programs (Cont.)

- Question

- Can you find the entry of a suspicious executable file in the result of AutoRuns?
 - the registry path and file path
 - why suspicious?

- Hints

- The system is Windows 7 SP1, UAC enabled
 - Focus on user settings (e.g., HKCU) first
- Most Microsoft binaries are not signature-verified unless the offline OS version is identical with your live OS version
 - Skip the Microsoft entries for now

Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

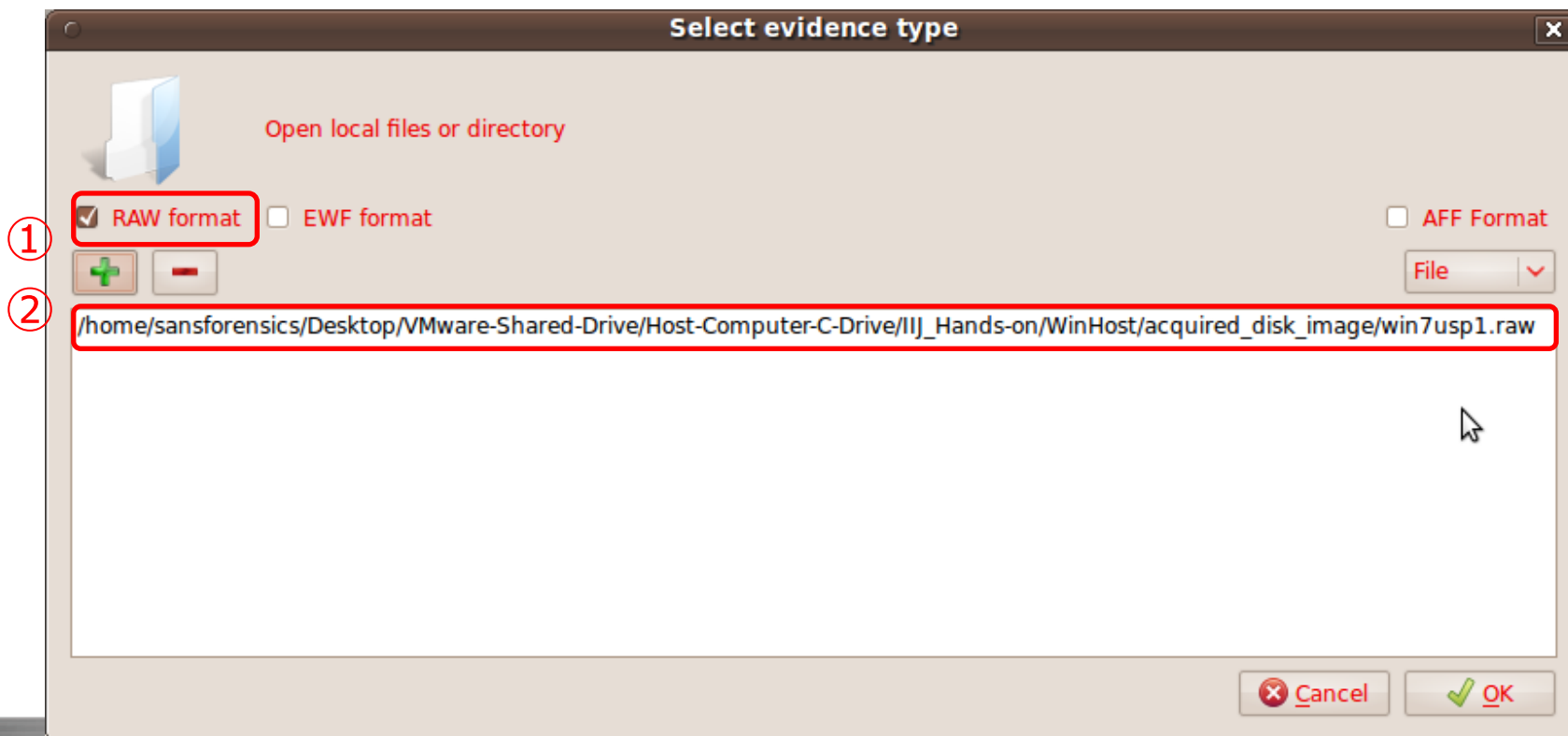
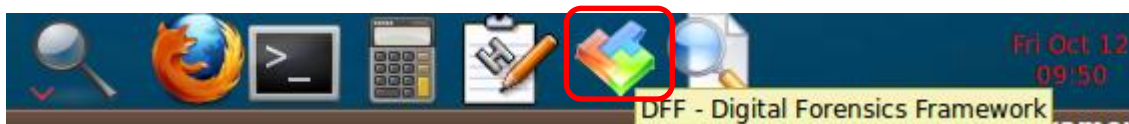
Identifying Malware Installation Time

- We found suspicious registry entries
- Registry keys include last written timestamps
 - For root cause of malware infection, we can trace back timeline based on the timestamps
- Registry File Extraction
 - Digital Forensic Framework
 - Parse disk images, then browse/display file content including deleted/unallocated space
- Registry Analysis
 - Registry Decoder
 - Parse registry files, then browse/search the keys/values/data

Hands-on 2

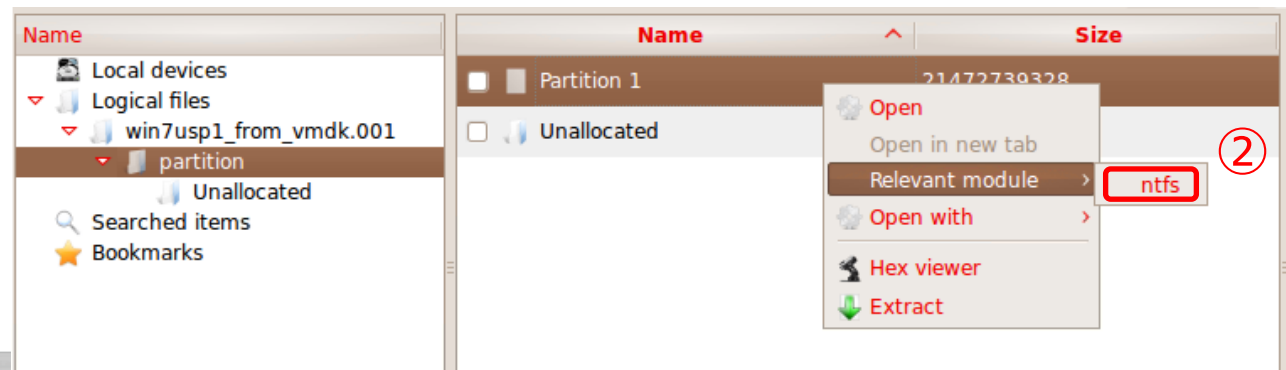
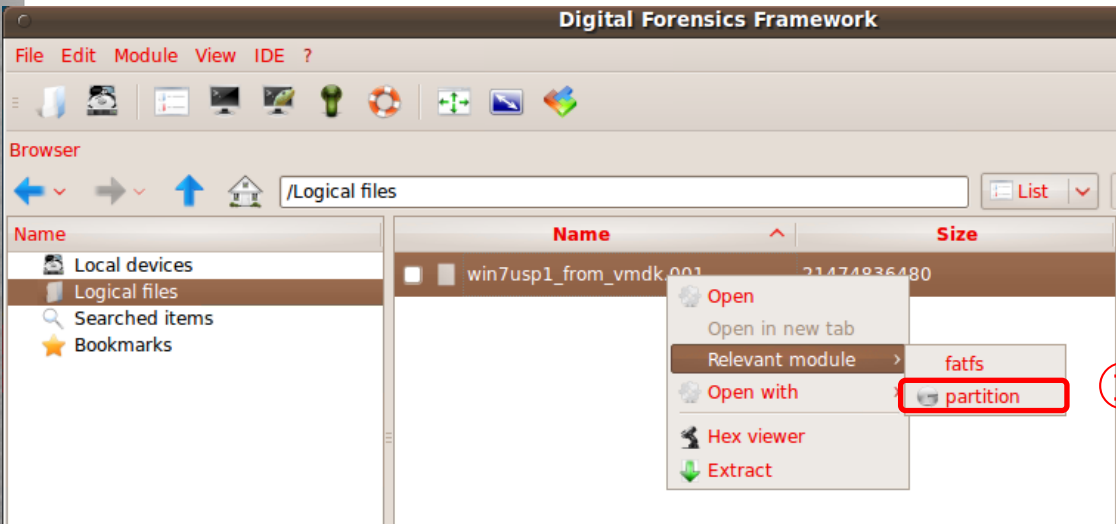
Registry File Extraction

- Digital Forensic Framework on SIFT
 - Click DFF icon on SIFT menu bar
 - [File] -> [Open evidence file(s)]
 - Specify ①RAW format, ②image file path



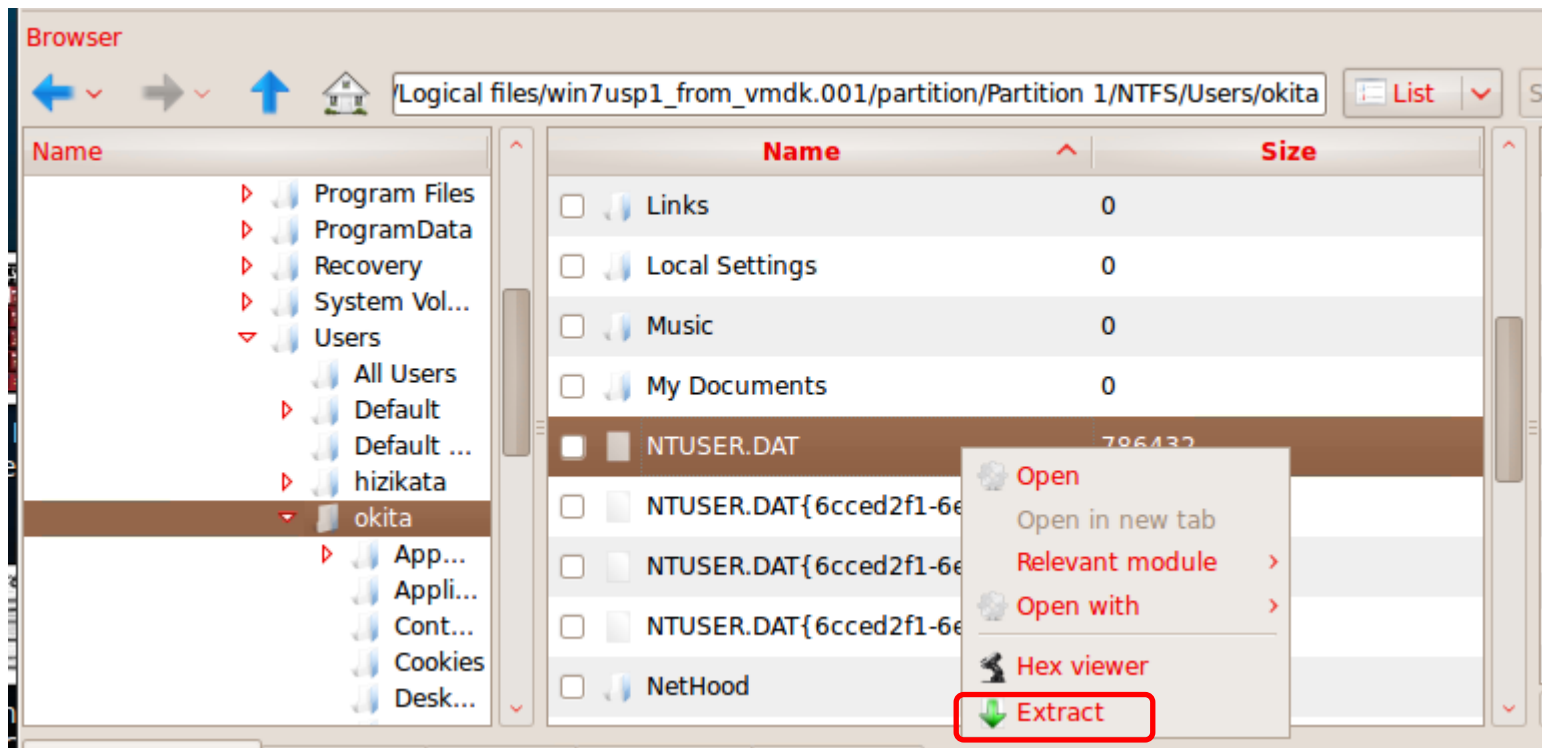
Registry File Extraction (Cont.)

- Digital Forensic Framework on SIFT
 - Parse NTFS filesystem using “Relevant module”
 - [Relevant module] -> ①partition, ②ntfs



Registry File Extraction (Cont.)

- Digital Forensic Framework on SIFT
 - Extract the registry file
 - Save it to Host OS's folder



Hands-on#2: Registry Analysis

- Registry Decoder
 - Extract "C:\IIJ_Hands-on\WinHost\tools\regedcoderR103.zip"
 - Run regdecoderR103.exe
 - Select [Start a new case] and Next
 - Create Case
 - Case Directory="C:\IIJ_Hands-on\WinHost\tools\regedcoderR103\test"
should be newly-created!!
 - Add Evidence
 - C:\IIJ_Hands-on\WinHost\exported_registry_files\okita\NTUSER.DAT

Create Case

Case Name: IIJ_Hands-on

Case Number:

Investigator Name:

Comments:

Case Directory:

The folder should be empty or crash!!

Add Evidence

	File Path	Alias (Optional)
1	C:\IIJ_Hands-on\WinHost\exported_registry_files\okita\NTUSER.DAT	

Registry Types (Disk Images Only): Current Backups (System Restore)

Hands-on#2: Registry Analysis

- Question
 - Check the last written time of the registry key including the suspicious registry values
 - How?

Hands-on#2: Registry Analysis

- Hints
 - Use Registry Decoder's Browse function
 1. Select [File View] tab, then click [View]
 2. Select opened [Browse] tab, then check the Run key
 - Use Registry Decoder's Search Function
 1. Select the registry file in [Search] tab
 2. Input search keyword in [Search Term] text area
 - You should extract the keyword from exe file path
 3. Select [Partial Search] if needed
 4. Check all kinds of search targets
 - Keys, Names, Data

Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - **Timeline Analysis (Hands-on#3)**
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Timeline Analysis

- Approach to root cause of malware infection
 - Check various timestamps of the suspicious binary for validation

	Registry key	File System	Prefetch	ShimCache
Description	last written time	MACB times	first & last run time	file modification time
Tool	log2timeline, Registry Decoder	log2timeline	Windows Prefetch Parser	ShimCache Parser
Risk	overwritten by another values	modified by malware	SSD image	? (shutdown needed)
Result YYYY/MM/DD HH:MM:SS	2012/10/5 18:48:30	2012/10/5 17:05:56	not found	2012/10/5 17:05:56

Hands-on#3: Timeline Analysis

- Check timeline generated by log2timeline-sift
 - Extract “C:\¥IIJ_Hands-on¥WinHost¥timeline¥win7usp1-current¥20120901-win7usp1-bodyfile.zip”
 - Open the CSV file with Excel or OpenOffice

Focus on
A, B, D, E, F, K columns

modification/last access/entry modified/creation

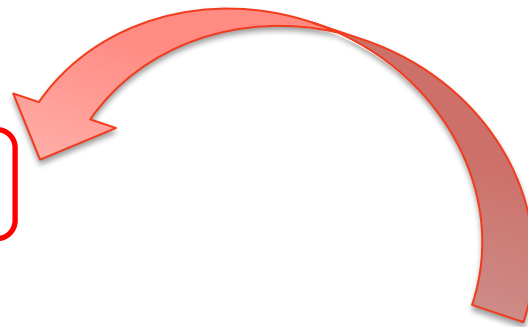
date	time	timezone	MACB	source	sourcetype	type	user
10/27/2006	9:49:52	Japan	M...	FILE	NTFS \$MFT	\$SI [M...] time	-

host	short	desc	VE
WIN7USP1	C:/Users/okita/AppData	C:/Users/okita/AppData/Roaming/Micr	

If deleted, “(deleted)” is added

Hands-on#3: Timeline Analysis (Cont.)

- Question
 - Are there any activities before the malware creation timestamp?
 - Related to the infection, what files were accessed/opened/created?
 - What was the user doing at that time?
- Hints
 - We have two timestamps
 - File System/ShimCache
 - 2012/10/5 17:05:56
 - Registry Key
 - 2012/10/5 18:48:30
 - In this hands-on, trace back timeline from **the earlier timestamp only**
 - In real case, we should check both of them
 - Check the activities **for several minutes** from the timestamp



Other Evidences of User Activities

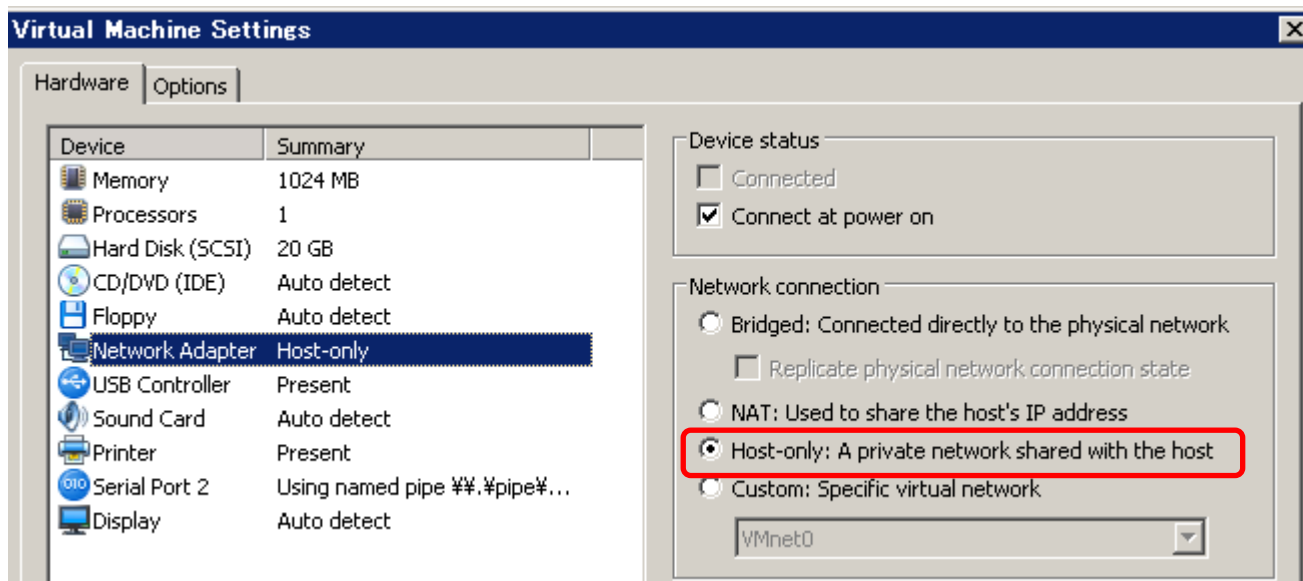
- GUI programs executed by the user
 - UserAssist key in NTUSER.DAT
 - number of runs, last run timestamp
 - Registry Decoder's User Assist plugin
- Opened files
 - Recently used Office files
 - C:\Users\\AppData\Roaming\Microsoft\Office\Recent
 - JumpList
 - C:\Users\\AppData\Roaming\Microsoft\Windows\Recent
 - JumpLister
 - NTUSER.DAT
 - Shell Bag, RecentDocs, etc..
 - Registry Decoder plugins (Search is also effective)

Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Setting up Windows Guest VM

- Install VMWare Tools
- Settings for running malware
 - Change the network connection to “Host-only”
 - If you use VMWare Workstation, take a snapshot for restoration
 - [VM] -> [Snapshot] -> [Take Snapshot]
 - If you use VMWare Player, edit the .vmx file to clear changes after power off (See below)
 - C:\IJJ_Hands-on\WinHost\conf\VMWare\Player_Win_setting_En.txt
- Power-on & logon
- Create “C:\MalwareAnalysis” folder on Windows VM, and drag and drop “C:\IJJ_Hands-on\WinVM” on host OS into that folder

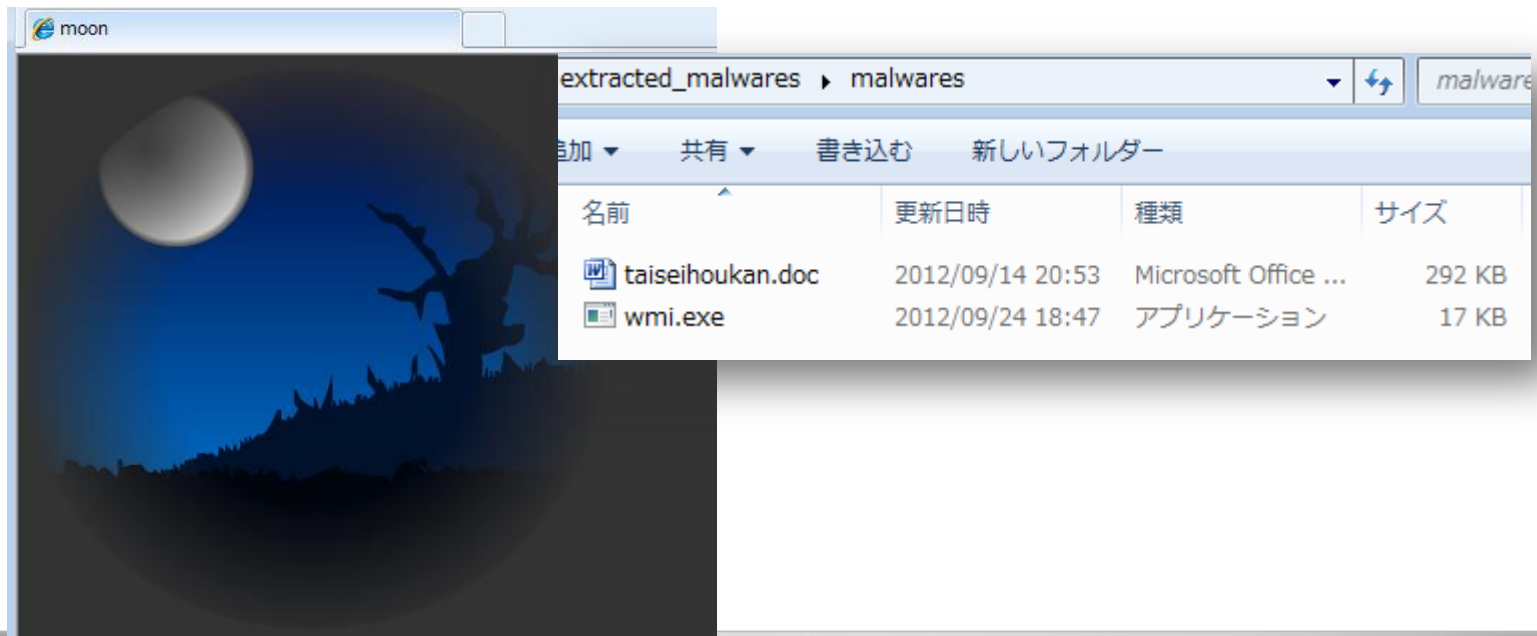


Dynamic Analysis of Malicious Document File

- Dynamic Analysis
 - Monitor RAM/disk/network activities after opening the doc file “taiseihoukan.doc” on Windows VM
 - Monitor process/filesystem/registry/network
 - Process Hacker/Process Explorer
 - CaptureBAT
 - Emulate fake server
 - FakeNet

Hands-on#4: Dynamic Analysis of Malicious Document File

- Set up for dynamic analysis
 - Install Adobe Flash Player ActiveX
 - "C:¥MalwareAnalysis¥WinVM¥tools¥flashplayer11_2r202_233_winax_32bit.exe"
 - Access to a Flash test page using Internet Explorer
 - "C:¥MalwareAnalysis¥WinVM¥tools¥flash_IE_test_page¥moon.html"
 - Extract the malware from zip file (Password: "infected")
 - C:¥MalwareAnalysis¥WinVM¥extracted_malwares¥malwares.zip".

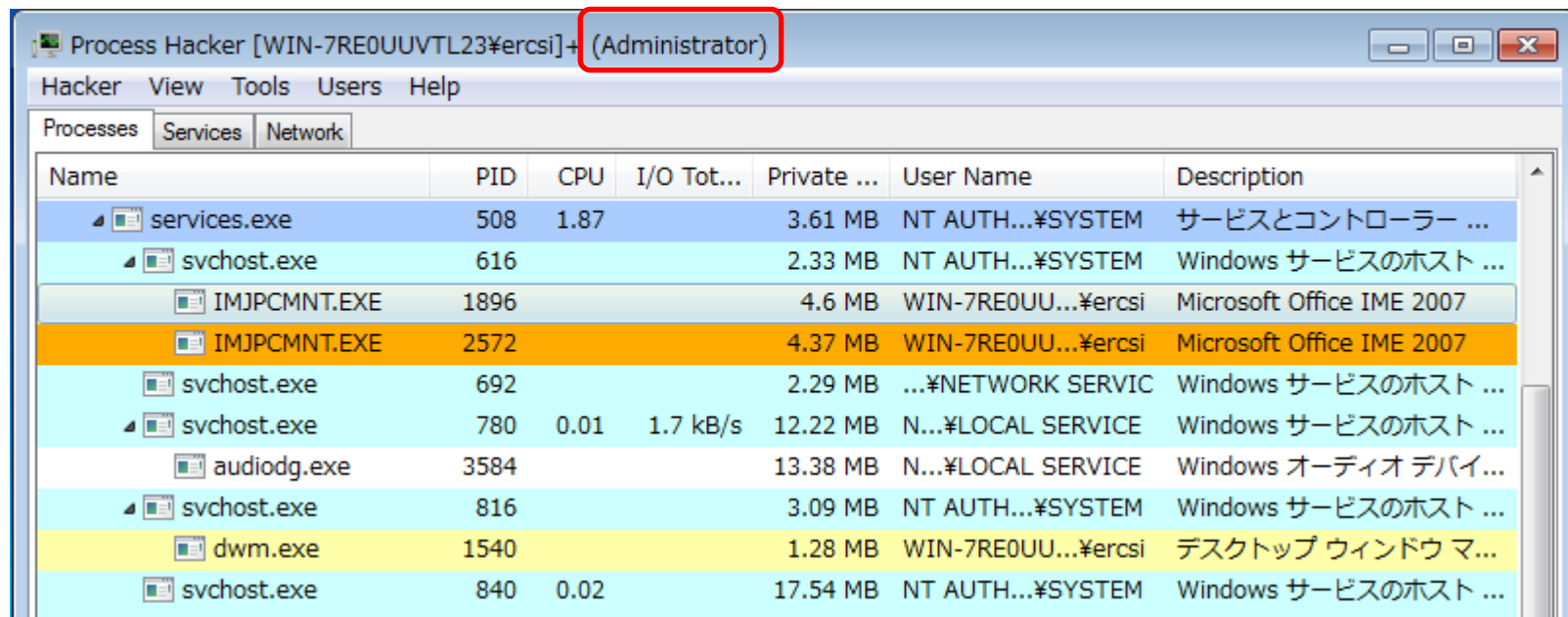


Hands-on#4: Dynamic Analysis of Malicious Document File (Cont.)

- Install CaptureBAT
 - Install
 - “C:\MalwareAnalysis\WinVM\tools\CaptureBAT\CaptureBAT.exe”
 - Restart the VM

Hands-on#4: Dynamic Analysis of Malicious Document File (Cont.)

- Process Hacker
 - Extract
 - "C:¥MalwareAnalysis¥WinVM¥tools¥processhacker-2.28-bin.zip"
 - Run **as administrator**
 - ArchName¥ProcessHacker.exe
 - Check process trees, installed services, network socket status



Hands-on#4: Dynamic Analysis of Malicious Document File (Cont.)

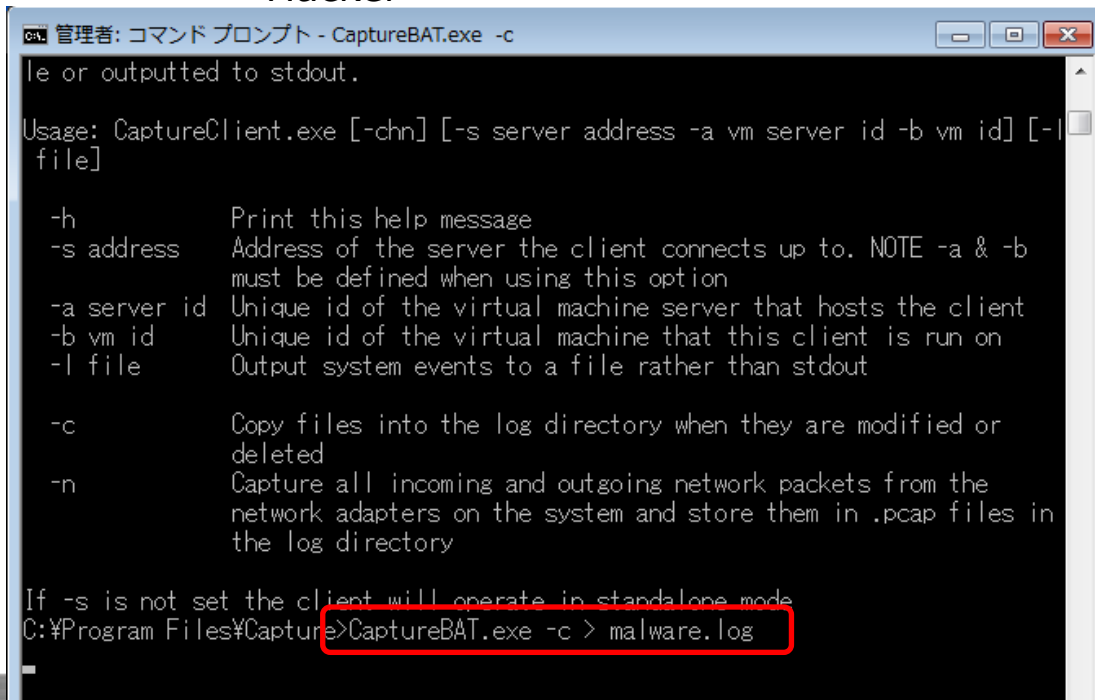
- FakeNet
 - Disable Windows Firewall
 - Extract
 - "C:¥MalwareAnalysis¥WinVM¥tools¥Fakenet1.0c.zip"
 - Run **as administrator** on cmd.exe
 - Check the configuration using nslookup command or web access



```
管理者: コマンド プロンプト - fakenet
C:¥mms¥tools¥windows_VM¥Fakenet1.0c¥Fakenet1.0b>fakenet
FakeNet Version 1.0
[Starting program, for help open a web browser and surf to any URL.]
[Press CTRL-C to exit.]
[Modifying local DNS Settings.]
Scanning Installed Providers
Installing Layered Providers
Preparing To Reorder Installed Chains
Reordering Installed Chains
Saving New Protocol Order
[Listening for DNS traffic on port: 53.]
[Listening for traffic on port 80.]
[Listening for SSL traffic on port 443.]
[Listening for SSL traffic on port 8443.]
[Listening for traffic on port 8080.]
[Listening for traffic on port 8000.]
[Listening for traffic on port 1337.]
[Listening for SSL traffic on port 31337.]
[Listening for ICMP traffic.]
[Listening for traffic on port 25.]
[Listening for SSL traffic on port 465.]
```

Hands-on#4: Dynamic Analysis of Malicious Document File (Cont.)

- Run CaptureBAT
 - After installation, the binary is located at **C:¥Program Files¥Capture**
 - Run **as administrator** on cmd.exe
 - Redirect the output to log file
 - -c: Capture modified and deleted files
 - After running, Check whether Process Hacker reports CaptureBAT services are created
 - If you cannot find the message, please check Services tab in Process Hacker



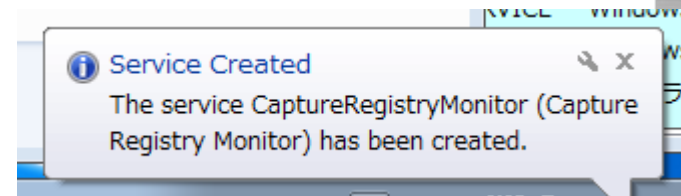
```
管理者: コマンド プロンプト - CaptureBAT.exe -c
le or outputted to stdout.

Usage: CaptureClient.exe [-chn] [-s server address -a vm server id -b vm id] [-l
file]

-h          Print this help message
-s address  Address of the server the client connects up to. NOTE -a & -b
            must be defined when using this option
-a server id Unique id of the virtual machine server that hosts the client
-b vm id    Unique id of the virtual machine that this client is run on
-l file     Output system events to a file rather than stdout

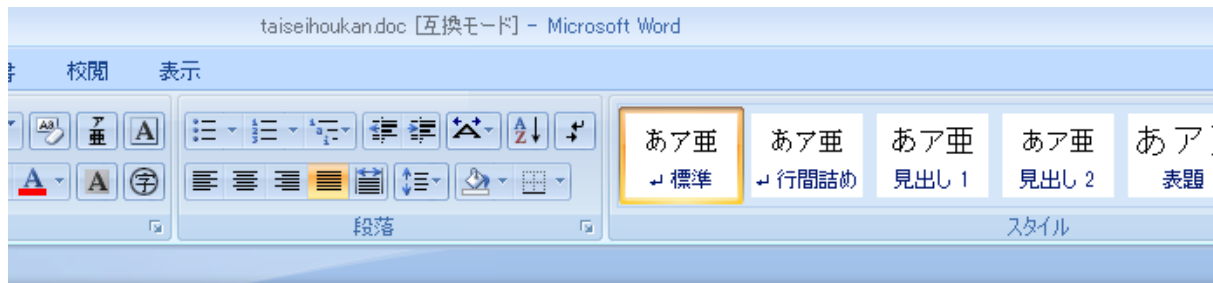
-c          Copy files into the log directory when they are modified or
            deleted
-n          Capture all incoming and outgoing network packets from the
            network adapters on the system and store them in .pcap files in
            the log directory

If -s is not set the client will operate in standalone mode
C:¥Program Files¥Capture>CaptureBAT.exe -c > malware.log
```



Hands-on#4: Dynamic Analysis of Malicious Document File (Cont.)

- Open the doc file
 - taiseihoukan.doc in "C:¥MalwareAnalysis¥WinVM¥extracted_malwares¥malwares.zip"
 - Run wmi.exe if Office 2007 is not installed in your VM
 - does NOT work on Office 2003 and 2010
- If successful, a dummy document will be opened



江戸時代、徳川将軍は日本の統治者として君臨していたが、形式的には朝廷より将軍宣下があり、幕府が政治の大権を天皇から預かっているという大政委任論も広く受け入れられていた。幕末、朝廷が自立的な政治勢力として急浮上し、主に対外問題における幕府との不一致により幕府権力の正統性が脅かされる中で、幕府は朝廷に対し大政委任の再確認を

Hands-on#4: Dynamic Analysis of Malicious Document File (Cont.)

- Questions
 - What's the malicious **hostname and port number** where the malware tries to connect?
 - Which **process** adds auto-start settings for the malware?
- Hint
 - Check the results
 - CaptureBAT
 - Press any key to exit
 - Search doc/exe name in the log
 - FakeNet
 - Press Ctrl-C
 - Check the console output

Analyzing Malicious Office Documents

- Checking embedded code/file
 - String search
 - Flash file signatures (“FWS”, “CWS”)
 - JavaScript (“ScriptBridge”), etc..
 - Parse OLE structure
 - FileInsight
 - Pyew/hachoir-subfile
- Scanning malicious payloads
 - OfficeMalScanner
 - Detect & extract PE/shellcode/swf

Hands-on#5: Analyzing Malicious Office Documents

- You **should work in VM**, not host OS (See hands-on mark)
- Question
 - Do you think what vulnerability was used for the exploitation of the PC?
 - **Guess CVE number** of this exploit.
- Hints
 - **Notice: The document seemed to include a Flash object**
 - Check & extract an embedded object in the Office document
 - FileInsight
 - OfficeMalScanner
 - Decompile the object
 - AS3 Sorcerer
 - Read the decompiled code and guess the vulnerability
 - Find characteristic strings and use search engine (e.g. Google) ;-)

Hands-on#5: Analyzing Malicious Office Documents (Cont.)

- How to use & install tools
 - FileInsight
 - Install
 - "C:\MalwareAnalysis\WinVM\tools\fileinsight.exe" in VM
 - Run
 - Drag and Drop "taiseihoukan.doc" into FileInsight

The screenshot displays the FileInsight interface. On the left, a 'Navigation' pane shows a tree view of the document's structure. The 'ObjectPool' folder is expanded, revealing a sub-folder named '_1406132688'. Within this folder, several objects are listed, including 'OLE' (20), 'rCompObj' (145), 'ObjInfo' (6), 'LOCXNAME' (34), and 'Contents' (202740). A red circle highlights this entire navigation pane. A red arrow points from the 'OLE' object in the tree to a red callout box on the right. The callout box contains the text: 'Browse OLE structure of the document'. The main window shows a hex dump of the document's content, with columns of hexadecimal values and their corresponding ASCII representations. The hex dump starts with '000034F0' and continues down to '00003630'. The ASCII column shows various characters, including 'BD 01 00 00 BE 01 00 00 BF 01 00 00 C0 01 00 00' and so on. At the bottom of the hex dump, there is a line of text: 'FUFU....FWS....' followed by 'x.....p.....D..' and '.....<rdf:RD' and 'F xmlns:rdf='htt'.

Hands-on#5: Analyzing Malicious Office Documents (Cont.)

- How to use & install tools
 - OfficeMalScanner
 - Extract
 - "C:¥MalwareAnalysis¥WinVM¥tools¥OfficeMalScanner.zip"
 - Run "OfficeMalScanner.exe path_to_doc scan"
 - Search PE/shellcode patterns and extract them
 - Extract SWF file

```
C:¥work¥tools¥OfficeMalScanner>OfficeMalScanner C:¥work¥malwares¥cve-2012-1535_m
odified_20120914¥cve-2012-1535_modified¥mws¥final¥taiseihoukan.doc scan

+-----+
|               OfficeMalScanner v0.55               |
|   Frank Boldwin / www.reconstructor.org             |
+-----+

[*] SCAN mode selected
[*] Opening file C:¥work¥malwares¥cve-2012-1535_modified_20120914¥cve-2012-1535_
modified¥mws¥final¥taiseihoukan.doc
[*] Filesize is 298496 (0x48e00) Bytes
[*] Ms Office OLE2 Compound Format document detected
[*] Format type Winword
[*] Scanning now...
```

Hands-on #5: Analyzing Malicious Office Documents (Cont.)

- How to use & install tools
 - AS3 Sorcerer
 - Install
 - “C:\MalwareAnalysis\WinVM\tools\as3sorcerer_setup.exe” in VM
 - Run and drag-and-drop the swf file into AS3 Sorcerer
 - Find characteristic strings and guess the vulnerability
 - Use search engine (e.g. google)

```
public var allocs:Array;

public function Main():void{
    this.FontClass = Main_FontClass;
    super();
    this.heapSpray();
    this.TextBlock_createTextLineExample();
}

public function TextBlock_createTextLineExample():void{
    var _local1 = "Edit the world in hex.";
    var _local2:FontDescription = new FontDescription
```

```
public function heapSpray():void{
    var _local1:uint;
    _local1 = 0;
    this.kbArray = new ByteArray();
    this.kbArray.endian = Endian.LITTLE_ENDIAN;
    var _local2:* = "0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c909090";
    var _local3:* = (_local2 + "9090909090E947010000C28F36D8A0DF");
    var _local4:String = _local3;
    var _local5:ByteArray = this.hexToBin(_local4);
    var _local6:uint = (_local4.length / 2);
    _local1 = 0;
    while (_local1 < 0x0400) {
```

Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - **Analysis of Shellcode and Malware**
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Shellcode Analysis

- Identification by reading decompiled code or p-code
- extraction from swf file
 - Use hex editor (e.g., FileInsight)
- emulation (checking APIs)
 - e.g., libemu
 - But, emulation doesn't work for this shellcode...
- Debugging
 - binary paste to debuggers or use launcher program
 - <http://practicalmalwareanalysis.com/labs/>
- Static Analysis
 - IDA Pro

```
push    ecx
push    [ebp+sc.field_113_hFile_exp_doc]
call    [ebp+sc.field_8_kernel32_GetFileSize]
cmp     eax, [ebp+sc.field_12F_word_doc_size]
jnz     short loc_1E2
push    ebp
push    0
push    80h ; 'I'
push    2
push    0
push    1
push    GENERIC_WRITE
lea     eax, [ebp+sc.field_34_aWordl_tmp]
push    eax
add     [ebp+sc.field_4_kernel32_CreateFileA], 5
jmp     short loc_224 ; opening C:¥WINDOWS¥ WORDL.tmp
```

Identifying the Malware

- Open the pcap captured by fakenet using Wireshark
 - The malware initiated communication by **sending random 256 bytes** on TCP port 80 of the server
 - PoisonIvy?
 - Camellia Encryption's challenge-response negotiation
 - <https://media.blackhat.com/bh-eu-10/presentations/Dereszowski/BlackHat-EU-2010-Dereszowski-Targeted-Attacks-slides.pdf>
 - <http://labs.alienvault.com/labs/index.php/category/blog/page/3/>

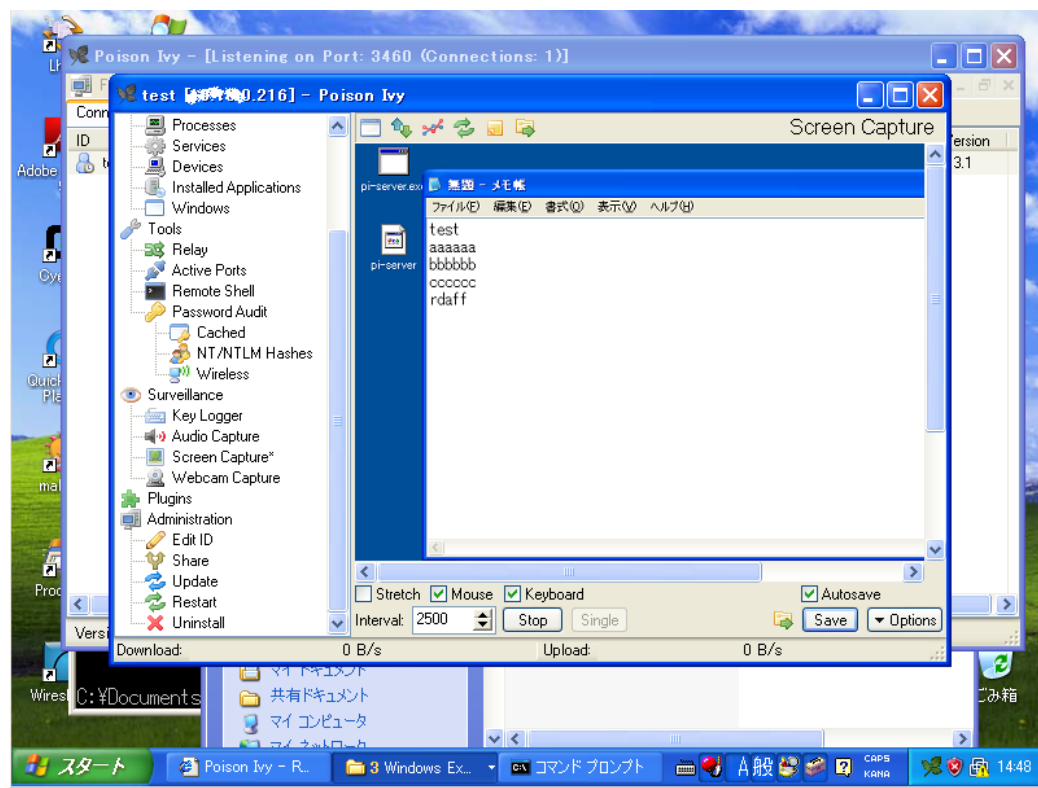
```

Data (256 bytes)
Data: 85d9136fb07238f8a41a9a0daf41e5fa79a957c0bffa5f7f...
[Length: 256]

0000 45 00 01 28 00 00 00 00 50 06 6b cd 7f 00 00 02 E..(....P.k.....
0010 7f 00 00 01 51 05 00 50 00 00 00 02 00 00 00 02 ....Q..P.....
0020 50 10 04 00 00 00 00 00 85 d9 13 6f b0 72 38 f8 P.....o.r8.
0030 a4 1a 9a 0d af 41 e5 fa 79 a9 57 c0 bf fa 5f 7f ....A..y.W...
0040 75 93 06 37 fb ac 4c 62 44 16 b8 be 26 12 4f d5 u..7..Lb D..&.O.
0050 73 01 a9 18 7e f9 1f 17 9b 9a 0d 15 7a 31 63 d2 s...~...z1c.
0060 0f d5 e8 3c e5 76 9f 22 05 17 87 03 ab 0e 12 cc ...<.v.".....
0070 4f e5 8d ea 89 86 d0 55 ef 62 5a f2 5b 56 6d 0d o.....U .bz.[vm.
0080 90 66 ac 4d 39 5b 1e f9 46 95 78 1f 63 4e 74 c9 .f.M9[. F.x.cNT.
0090 2b e9 1f 31 8f 0b a3 fb f4 34 73 59 04 13 ed 89 +..1...4sY...
00a0 7f 83 cc 02 08 3e 48 bb 93 6b f9 e9 1c b7 88 67 .....>H. .k....g
00b0 7a 3a 61 aa ad 4d 14 09 b8 38 e9 4b d5 83 a7 d8 z:a..M. .8.K....
00c0 5e 86 cb 51 0c e8 5b 36 c2 bb 7f e8 23 1b 04 8e ^..Q..[6 ....#.
00d0 ca d3 c8 2f 50 5c d2 ff 2e 4e 2d ba 8e 5a 11 2b .../P\..N-.Z.+
00e0 1a 25 36 d2 97 91 f8 05 bb 0e 02 b3 3a 1c ed 01 .%6.....:...
00f0 7d ce a8 19 b7 9f f4 ba 50 3b 37 b2 02 c0 78 14 }.....P;7...x.
0100 5f b8 7f e1 4d cd e0 c1 ae 76 70 a8 1b a6 6b 80 ...M...vp...k.
0110 7f 8a a7 54 22 82 fc fb 7f 33 e1 0d c1 44 d9 31 ...T"....3...D.1
0120 e8 c4 21 24 63 9f 6b d7 ..!$c.k.
  
```

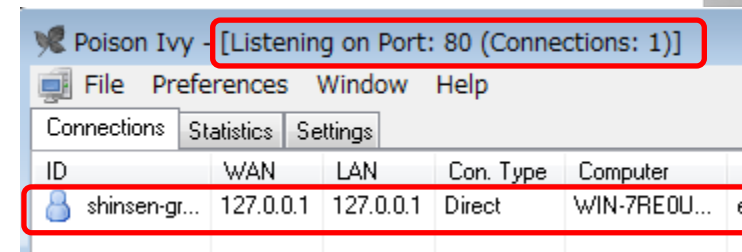
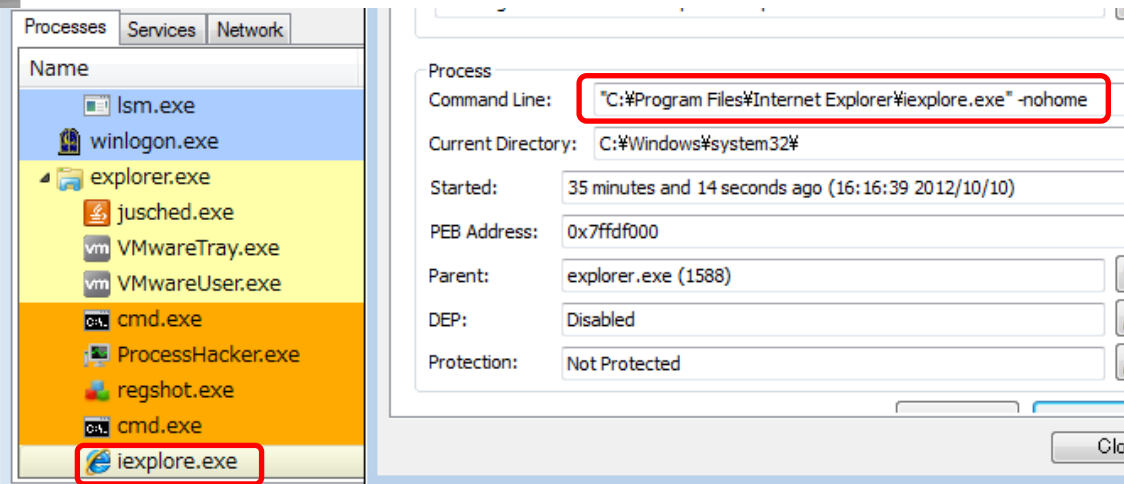

What's Poison Ivy?

- Poison Ivy is an infamous RAT (Remote Administration Tool)
 - Everyone can download the latest version at a certain web site
- execute arbitrary code
 - keylogging
 - hijacking mouse/keyboard
 - stealing data MIC/WebCam
 - file download/upload
- and so on ...



Other Traits of Poison Ivy

- Hidden iexplore.exe
- PoisonIvy GUI client in VM can be connected from the malware
 - Because Fakenet redirect the connection to localhost
 - The password is default ;-)
- Quick Analysis using Memory Forensics
 - Redline's Malware Risk Index (handle name: !VoqA.I4)
 - Code injection activities



Malware Risk Index Hits



This process has a module which imports a suspicious Handle (Mutant) !VoqA.I4. "Process has a known Poison Ivy mutant".

Analyzing Poison Ivy

- Unpacking
 - Break VirtualAllocEx/VirtualProtectEx and extract the unpacked PE
- Debugging
 - Fragmented code injections
 - wmi.exe
 - inject code to explorer.exe
 - explorer.exe
 - install wmi.exe, create iexplore.exe process and inject code to it
 - iexplore.exe
 - connect to Poison Ivy GUI client
- Static Analysis
 - shellcode-like API resolution
 - position-independent code (e.g., call [esi + *])

```
push 40h ; fIProtect
push 3000h ; fIAllocationType
push [ebp+dwSize] ; dwSize
push 0 ; lpAddress
push [ebp+hProcess] ; hProcess
call [esi+pi_struct.field_b1_kernel32_VirtualAllocEx]
push eax
lea edi, [ebp+var_4]
push edi ; *lpNumberOfBytesWritten
push [ebp+dwSize] ; nSize
push [ebp+arg_C] ; lpBuffer
push eax ; lpBaseAddress
push [ebp+hProcess] ; hProcess
call [esi+pi_struct.field_b5_kernel32_WriteProcessMemory]
```

Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - **Result**
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Result about Root Cause Analysis of Malware Infection

See the answer slide

Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Bonus Hands-on: Tracking Attacker's Activities

- Question 1
 - Examine post-infection activities
 - Is there any tool or exploit used by the attacker?
 - When was the tool downloaded?

Bonus Hands-on: Tracking Attacker's Activities (Cont.)

- Hints for Question 1
 - Imagine Attacker's Activities from evidences that have been achieved thus far
 - a.7z
 - Domain Controller password hash database (ntds.dit) was included
 - » It means DC was compromised ☹
 - Event logs
 - Different person account was authenticated on Client A
 - » The acquired password hash may be used
 - What kind of tools did he use for these operations?

Bonus Hands-on: Tracking Attacker's Activities (Cont.)

- Hints for Question1
 - Strategies checking timeline
 - check the period after malware installation
 - check external information to narrow down the time period
 - in this case, "a.7z"
 - check result*.txt
 - » suspicious path
 - » "C:¥Users¥okita¥AppData¥Local¥Temp¥t"
 - » sign of "psexec" execution
 - » "¥PIPE¥psexecsvc" found in "net file" command
 - search "psexec" on timeline

Bonus Hands-on: Tracking Attacker's Activities (Cont.)

- Hints for Question1
 - timestamps changed by the attacker
 - Two kinds of timestamps in NTFS file system
 - Standard Information (SI) Attribute
 - File Name (FN) Attribute
 - If you want to make timeline with FN attribute timestamps for yourself, you should change log2timeline-sift code
 - <http://list-archives.org/2012/07/10/dfir-lists-sans-org/log2timeline-vs-log2timeline-sift/f/4359338113>

SI Attribute includes timestamps generally referred to by OS. They can be modified by APIs (e.g., SetFileTime).

FN Attribute also has timestamps but it cannot be modified by APIs.

MFT record
of the file

MFT Header

Standard Information
(SI) Attribute

Filename
(FN) Attribute

Remaining Attributes...
(e.g., Data Attribute)

Bonus Hands-on: Tracking Attacker's Activities (Cont.)

- Hints for Question1
 - Extract and check the timeline with FN timestamps
 - "C:\¥IJJ_Hands-on¥WinHost¥timeline¥win7usp1-current-with-fn¥20120901-win7usp1_bodyfile_with-fn.csv.zip"
 - Search one of the tool names (e.g., "psexec")
 - check the FN attribute timestamp
 - You can differentiate kinds of file system timestamp by means of type(G) column

A	B	C	D	E	F	G	H	I	J	
date	time	timezone	MACB	source	sourcetype	type	user	host	short	desc
10/27/2006	9:49:52	Japan	M...	FILE	NTFS \$MFT	\$SI [M...] time	-	WIN7USP: C:/Users/c	C:/Users/okita/App	
7/1/2007	1:35:21	Japan	M...	FILE	NTFS \$MFT	\$FN [M...] time	-	WIN7USP: C:/Users/c	C:/Users/okita/App	
2/5/2008	8:00:00	Japan	M...	FILE	NTFS \$MFT	\$FN [M...] time	-	WIN7USP: C:/Users/c	C:/Users/okita/App	
2/10/2008	14:30:46	Japan	M...	FILE	NTFS \$MFT	\$FN [M...] time	-	WIN7USP: C:/Users/c	C:/Users/okita/App	
6/11/2009	6:16:34	Japan	.A.B	FILE	NTFS \$MFT	\$FN [.A.B] time	-	WIN7USP: C:/Windo	C:/Windows/Syster	
6/11/2009	6:16:34	Japan	.A.B	FILE	NTFS \$MFT	\$FN [.A.B] time	-	WIN7USP: C:/Windo	C:/Windows/winsx:	
6/11/2009	6:16:34	Japan	.A.B	FILE	NTFS \$MFT	\$FN [.A.B] time	-	WIN7USP: C:/Windo	C:/Windows/winsx:	

Bonus Hands-on: Tracking Attacker's Activities (Cont.)

- Question2
 - Examine post-infection activities
 - Can you find “a.7z”?
 - Any other leaked files?

Bonus Hands-on: Tracking Attacker's Activities (Cont.)

- Hints for Question2

- overwritten file meta data or securely deleted files

- Restore files from Volume Shadow Copy

- Windows Approach (Windows 7/Server 2008 required)

- » Convert the dd image to vhd format (image backup recommended)

- » `vhdtool /convert <filename>`

- » `C:¥IJJ_Hands-on¥WinHost¥tools¥vhdtools`

- » Mount the vhd image

- » "Attach VHD" in Disk Management

- » Check VSCs and export files

- » ShadowKit

- » `C:¥IJJ_Hands-on¥WinHost¥tools¥ShadowKit_Portable_v1.5`

- SANS SIFT Workstation's Approach

- » Calculate the disk offset to mount

- » `fdisk -lu <filename>`

- » Extract VSCs

- » `vshadowmount -o <disk_offset_value>`

- » Check VSCs and export files

- » `log2timeline-sift` and `TSK`

- » The generated VSC timeline is located in "`C:¥IJJ_Hands-on¥WinHost¥timeline¥win7usp1-vss3¥20120901-vss3-bodyfile.zip`"

The image will be overwritten without confirmation!

Don't run twice!

Analysis in the Case

- Timeline Creation
- Root Cause Analysis of Malware Infection
 - Checking automatic start-up programs (Hands-on#1)
 - Identifying Malware Installation Time (Hands-on#2)
 - Timeline Analysis (Hands-on#3)
 - Analysis of Malicious Document File (Hands-on#4, Hands-on#5)
 - Analysis of Shellcode and Malware
 - Result
- Analysis of Post-infection Activities (Bonus Hands-on)
 - Investigating Attacker's Activity
 - Analyzing Unknown Binary
- Wrap-up

Timeline of the Incident

See the answer slide

Wrap-up

- Forensic investigation and malware analysis combination can clear
 - root cause of malware infection
 - malware type/functions
 - post-infection activities
- Practical disk image is more chaotic
 - high-capacity disk, many unknown binaries
 - data loss over long term
 - evidence contamination by first responders
- Free tools have reasonable functions, but commercial tools often work effectively
 - IDA Pro
 - EnCase/X-Ways Forensics
 - etc..
- **IMPORTANT: delete the disk image after hands-on**

Contact



E-mail:

t-haruyama@iij.ad.jp

hiroshi-suzuki@iij.ad.jp

Twitter:

@cci_forensics

@herosi_t

Ongoing Innovation

本書には、株式会社インターネットイニシアティブに権利の帰属する秘密情報が含まれています。本書の著作権は、当社に帰属し、日本の著作権法及び国際条約により保護されており、著作権者の事前の書面による許諾がなければ、複製・翻案・公衆送信等できません。IIJ、Internet Initiative Japanは、株式会社インターネットイニシアティブの商標または登録商標です。その他、本書に掲載されている商品名、会社名等は各会社の商号、商標または登録商標です。本文中では™、®マークは表示していません。

©2010 Internet Initiative Japan Inc. All rights reserved. 本サービスの仕様、及び本書に記載されている事柄は、将来予告なしに変更することがあります。

URL Reference

- Forensic Analysis
 - OS
 - SANS SIFT Forensic Workstation
 - <http://computer-forensics.sans.org/community/downloads>
 - Timeline Creation
 - log2timeline-sift
 - blogs.sans.org/computer-forensics/files/2012/06/SANS-Digital-Forensics-and-Incident-Response-Poster-2012.pdf
 - <http://computer-forensics.sans.org/blog/2011/12/16/digital-forensics-sifting-cheating-timelines-with-log2timeline>
 - <http://computer-forensics.sans.org/blog/2011/12/07/digital-forensic-sifting-super-timeline-analysis-and-creation>
 - <http://computer-forensics.sans.org/blog/2011/11/30/log2timeline-plugin-creation>
 - log2timeline
 - <http://code.google.com/p/log2timeline/>
 - Adding \$fn (\$filename) attribute timestamps
 - <http://list-archives.org/2012/07/10/dfir-lists-sans-org/log2timeline-vs-log2timeline-sift/f/4359338113>
 - File System Analysis
 - Digital Forensic Framework
 - <http://www.digital-forensic.org/>
 - TSK
 - <http://www.sleuthkit.org/>
 - Checking Program Execution Cache
 - Prefetch Parser
 - <http://computer-forensics.sans.org/blog/2010/02/12/prefetch-parser-v1-4/>
 - ShimCacheParser
 - <https://github.com/mandiant/ShimCacheParser>
 - Volume Shadow Copy Analysis
 - Accessing Volume Shadow Copies
 - <http://windowsir.blogspot.jp/2011/01/accessing-volume-shadow-copies.html>
 - Shadow Timelines And Other VolumeShadowCopy Digital Forensics Techniques with the Sleuthkit on Windows
 - <http://computer-forensics.sans.org/blog/2011/09/16/shadow-timelines-and-other-shadowvolumecopy-digital-forensics-techniques-with-the-sleuthkit-on-windows>
 - ShadowKit
 - <http://redrocktx.blogspot.jp/p/shadowkit.html>

URL Reference (Cont.)

- Forensic Analysis
 - Registry Analysis
 - Registry Decoder
 - <http://www.digitalforensicssolutions.com/registrydecoder/>
 - Web History Analysis
 - IECacheView
 - http://www.nirsoft.net/utills/ie_cache_viewer.html
 - IEHistoryView
 - <http://www.nirsoft.net/utills/iehv.html>
 - Web historian
 - <http://www.mandiant.com/resources/download/web-historian>
 - Detecting Suspicious Auto-start Programs
 - Autoruns
 - <http://technet.microsoft.com/ja-jp/sysinternals/bb963902.aspx>
 - Disk Image Mounting
 - FTK Imager
 - <http://accessdata.com/support/product-downloads>
 - OSFMount
 - <http://www.osforensics.com/tools/mount-disk-images.html>
 - Event Log Analysis
 - Event Viewer
 - Event Log Explorer
 - <http://www.eventlogxp.com/>
 - Image Format Conversion
 - qemu-img
 - https://access.redhat.com/knowledge/docs/ja-JP/Red_Hat_Enterprise_Linux/5/html/Virtualization/sect-Virtualization-Tips_and_tricks-Using_qemu_img.html
 - FTK Imager
 - <http://accessdata.com/support/product-downloads>
 - vhdtool
 - <http://archive.msdn.microsoft.com/vhdtool>

URL Reference (Cont.)

- Malware Analysis
 - Observing process/file system/network
 - process monitor
 - <http://technet.microsoft.com/ja-jp/sysinternals/bb896645.aspx>
 - process explorer
 - <http://technet.microsoft.com/ja-jp/sysinternals/bb896653.aspx>
 - process hacker
 - <http://processhacker.sourceforge.net/>
 - captureBAT
 - <http://www.honeynet.org/node/315>
 - API Monitor
 - <http://www.rohitab.com/apimonitor>
 - Checking difference after malware execution
 - regshot
 - <http://sourceforge.net/projects/regshot/>
 - Network Analysis during malware execution
 - Wireshark
 - <http://www.wireshark.org/>
 - Simulating internet servers
 - InetSim
 - <http://www.inetsim.org/>
 - FakeNet
 - <http://practicalmalwareanalysis.com/fakenet/>

URL Reference (Cont.)

- Malware Analysis
 - Code Analysis
 - CFF Explorer
 - <http://www.ntcore.com/exsuite.php>
 - IDA Pro 5.0 Free
 - http://www.hex-rays.com/products/ida/support/download_freeware.shtml
 - OllyDbg
 - <http://www.ollydbg.de/>
 - Immunity Debugger
 - <http://debugger.immunityinc.com/>
 - libemu
 - <http://libemu.carnivore.it/>
 - Malzilla
 - <http://malzilla.sourceforge.net/>
 - Binary Editor
 - FileInsight
 - <http://www.mcafee.com/us/downloads/free-tools/fileinsight.aspx>
 - Javascript Analysis
 - jsunpack-n
 - <https://code.google.com/p/jsunpack-n/>
 - Revelo
 - <http://www.kahusecurity.com/2012/revelo-javascript-deobfuscator/>
 - Malzilla
 - <http://malzilla.sourceforge.net/>
 - PDF Analysis
 - <http://computer-forensics.sans.org/blog/2011/05/04/extract-flash-from-malicious-pdf-files/>
 - PDF Stream Dumper
 - <http://sandsprite.com/blogs/index.php?uid=7&pid=57>
 - <http://blog.zeltser.com/post/3235995383/pdf-stream-dumper-malicious-file-analysis>
 - <http://www.kahusecurity.com/2011/pdf-analysis-using-pdfstreamdumper/>
 - peepdf
 - <http://eternal-todo.com/tools/peepdf-pdf-analysis-tool>

URL Reference (Cont.)

- Malware Analysis
 - Analyzing MS Office documents
 - OfficeMalScanner
 - <http://www.reconstructor.org/code.html>
 - offvis
 - <http://www.microsoft.com/en-us/download/details.aspx?id=2096>
 - Flash Analysis
 - AS3 Sorcerer
 - <http://www.as3sorcerer.com/>
 - SWFTOOLS
 - <http://www.swftools.org/>
 - <http://securitylabs.websense.com/content/Blogs/3165.aspx>
 - SWFREtools
 - <https://github.com/sporst/SWFREtools/>
 - <http://www.google.co.jp/search?q=malicious+swf+analysis&ie=utf-8&oe=utf-8&hl=ja&client=ubuntu&channel=fs>
 - SWFInvestigator
 - <http://labs.adobe.com/technologies/swfinvestigator/>
 - JAVA Analysis
 - jad
 - <http://www.varaneckas.com/jad/>
 - jd
 - <http://java.decompiler.free.fr/?q=jdgui>

URL Reference (Cont.)

- Exploit
 - CVE-2012-1535
 - <http://contagiodump.blogspot.jp/2012/08/cve-2012-1535-samples-and-info.html>
 - <http://contagio.deependresearch.org/docs/CVE-2012-1535-Adobe-Flash-Player-Integer-Overflow-Vulnerability-Analysis.pdf>
 - <http://labs.alienvault.com/labs/index.php/2012/cve-2012-1535-adobe-flash-being-exploited-in-the-wild/>
 - CVE-2011-1249 (MS11-046)
 - <http://www.exploit-db.com/wp-content/themes/exploit/docs/18712.pdf>

Book Reference

- Forensic Analysis
 - File System Forensic Analysis
 - Windows Forensic Analysis DVD Toolkit
 - Mastering Windows Network Forensics and Investigation
- Malware Analysis, Reverse-Engineering
 - Rootkits: Subverting the Windows Kernel
 - The Rootkit Arsenal
 - Malware Analyst's Cookbook and DVD
 - Practical Malware Analysis
 - IDA Pro Book
 - Reversing: Secrets of Reverse Engineering
 - Windows Internals