



# MEMORY ANALYSIS AND FORENSICS



# Who am I?

- Peter Silberman
- Researcher/Engineer at MANDIANT
- Author of Audit Viewer
- One of the authors of Memoryze

# Who Are *YOU*?

- *You say ... No Comment...*
- You are familiar with acronyms like APT/CDT
  - You've seen these furry creatures in action?
- You've caught APT using disk forensics?
- You've done some memory analysis/acquisition?
- Didn't raise their hand but did catch APT?

# What's this talk about?

- Brief intro to memory analysis
- Why we need memory analysis
  - Why we need it for APT
- Introducing Malware Rating Index (MRI)
- DEMO can you say A P T?
  - Yea... I bet you can...

5

# Memory Analysis

Disk forensics is soooo 2000 and late



## F.A.Q. about Memory Analysis

- Is memory analysis and acquisition possible on:
  - Windows Vista,
  - 2003 Server, and
  - 2008 Server?
- Can I analyze or acquire more than 4 GB of RAM with a 32-bit application?
- Is live memory analysis easier to subvert than acquisition?

## F.A.Q. about Memory Analysis

- What is the difference in the memory footprint between live memory analysis and acquisition?
- I've heard that memory analysis/acquisition destroys data
  - This is why I don't do it
  - Is this true?

# Why Memory Analysis?

- There is more data to look through each day
  - Hard drives are larger
  - More files exist
  - More places for an attacker to hide
- Memory analysis can facilitate quickly triaging hosts
  - If the host is still ill (has malicious software running), memory is the best place to look
  - Memory analysis can powerfully augment disk analysis

# Chasing APT

- Why memory should be used in APT investigations?
  - APT tries very hard to camouflage itself
    - Use Windows file names
    - Usually not packed
    - Replicate system DLL resource sections
    - Small file sizes
  - It's harder to hide in memory


# APT Camouflage

- APT isn't like mass malware
- Only 10% of APT backdoors were packed
- File names and sizes:
  - MM most common file name: setup.exe
  - MM average file size: 675.975 KB
  - APT most common file name: svchost.exe
  - APT average file size: 121.85 KB

APT malware filename	Mass malware filename
svchost.exe	setup.exe
iexplore.exe	server.exe
iprinp.dll	keygen.exe
winzf32.dll	1.exe

# Case Study

- From an actual incident
- Can you spot the APT?
  - You have a 1 in 34 chance!
  - C'mon!
  - You win a backpack!



- glmf32.dll
- glu32.dll
- gpedit.dll
- gpkcsp.dll
- gpksrc.dll
- gpprefcl.dll
- gptext.dll
- grxccq.dll
- GSWAG32.DLL
- GSWDLL32.DLL
- h323msp.dll
- HAL.DLL
- hbaapi.dll
- hccoin.dll
- hdaprop.dll
- hdaudres.dll
- HHActiveX.dll
- hhsetup.dll
- hid.dll
- hlink.dll
- hnetcfg.dll
- hnetmon.dll
- hostmib.dll
- hotplug.dll
- HP1006LM.DLL
- hpbicoin.dll
- hpbmiapi.dll
- HPBMINI.DLL
- HPBNRAC2.DLL
- hpboid.dll
- hpboidps.dll
- hpbpro.dll
- hpbprops.dll
- hpcbrand.dll

# Case Study

- At first neither could we....
- Memory forensics solved this incident
  - Least Frequency of Occurrence
    - grxccq.dll

# Malware Behaviors

Behaving badly was never so unsexy



# APT Behaviors (in Memory)

- Double infection prevention/Suspicious Handles
  - Create unique mutants to identify compromised machines
  - Not as common in APT, but has been seen
- Process path execution
  - Copies itself to folders in `\windows\system32\somesubfolder\*`
  - Executes as `svchost.exe`
- User Execution
  - Processes such as `svchost` executing as a normal user
  - Tied to image unmapping or process path

# Can you spot the bad mutants?

## Mutex

985635577-7  
985635577-99  
nfafzvelhgzxh

op1mutx9

Hacker.com.cn\_MUTEX

'D'r'o'p'p'e'd'S'k'y'N'e't'

\_-o0]xX|-S-k-y-N-e-t-|Xx[0o-\_-

YY99knPY

\_\_\_\_->>>>U<<<<--\_\_\_\_

u\_joker\_v3.06

\_!SHMSFTHISTORY!\_

# Bad Mutants

Virus Name	Mutex
Kido.ih aka Conficker	.*-7 and .*-99 and [randomascii]
Sality.AA	op1mutx9
Flystud.??	Hacker.com.cn_MUTEX
NetSky	'D'r'o'p'p'e'd'S'k'y'N'e't' _-o0]xX -S-k-y-N-e-t- Xx[0o-_ YY99knPY ____- - - >>>U<<<<- - ____
Sality.W	u_joker_v3.06
Explorer.exe (not a virus)	_!SHMSFTHISTORY!_

# APT Behaviors (in Memory)

- Service Injection
  - Overwrite standard services
    - 6to4
  - Create new services
    - Windns (made up service name)
  - Very Common
    - Only evidence is a DLL
      - Unsigned

# Malware Rating Index (MRI)

Ratings never meant so much

# MANDIANT MRI: Background

- The positive:
  - Helps pinpoint what should be analyzed
  - Should help identify malware
  - Helps apply scarce resources efficiently
  - Highly configurable to environments
- Debbie Downer
  - *Will not work in all cases*
  - *Not a silver/gold/platinum/magic bullet*
  - *Will not replace analysis*

# MANDIANT MRI: Background

- Uses output from Memoryze
- MRI logic built into Audit Viewer
- Two components:
  - Builds on APT/Mass Malware known behaviors
  - Unique process/module scoring system

# MRI

- ***Definable*** behaviors:
  - Process Path Verification
  - Process User Verification
  - Process Handle Verification
- Defining malware behavior is as simple as an English sentence.
  - *With little punctuation* 😊

# Process Path Verification

- If *process* is executed from any other directory than `\windows\system32` display this message to the user

<ProcessPath>*svchost.exe*:\windows\system32:process was launched from a non default location, this is very suspicious</ProcessPath>

# Process Path Verification

- If process is executed from any other directory than `\windows\system32` display this message to the user

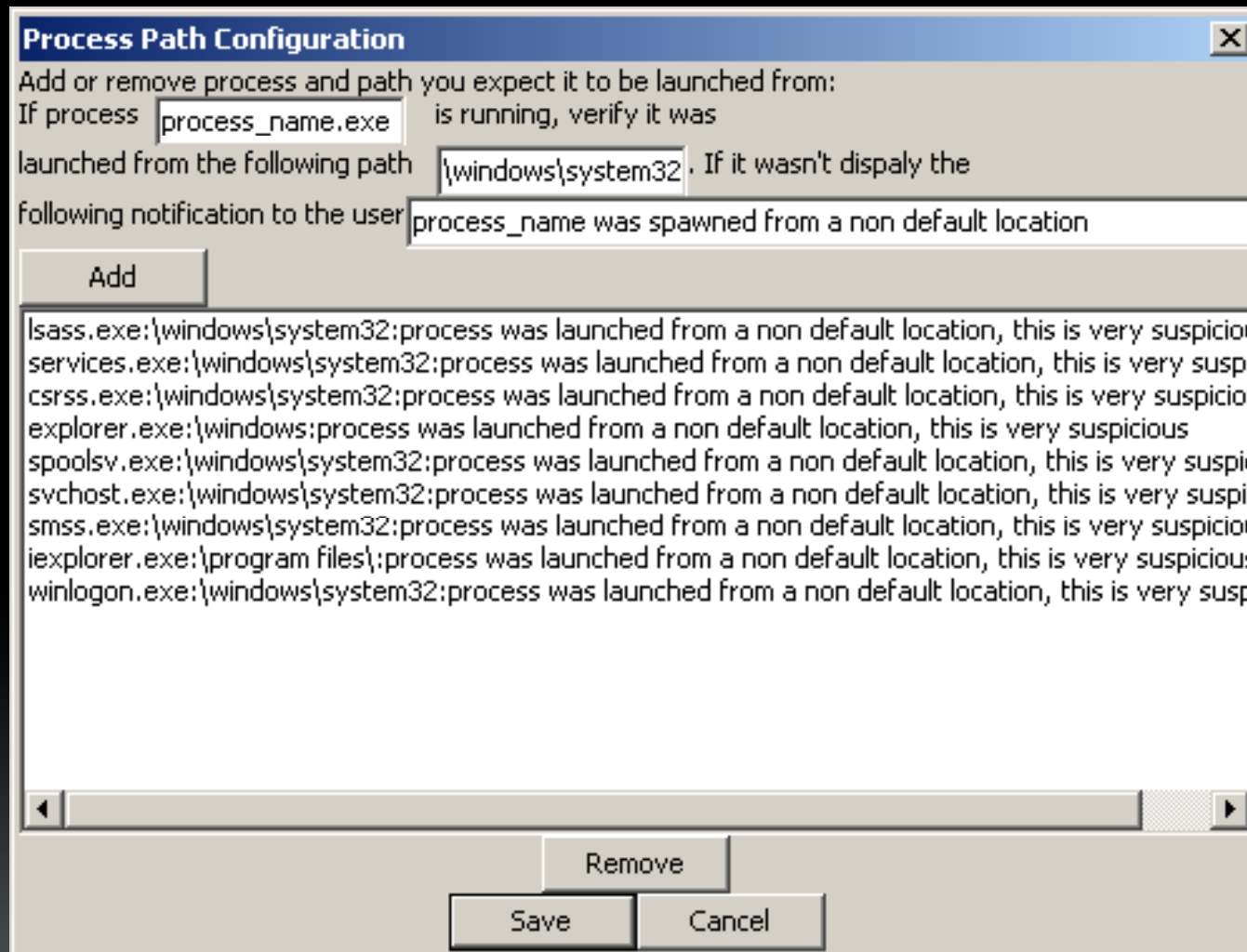
```
<ProcessPath>svchost.exe: windows\system32:process was launched from a non default location, this is very suspicious</ProcessPath>
```

# Process Path Verification

- If process is executed from any other directory than `\windows\system32` display *this message to the user*

<ProcessPath>svchost.exe:\windows\system32:*process was launched from a non default location, this is very suspicious*</ProcessPath>

# Process Path Verification



# Process User Verification

- If *process* is being executed by any other user, other than *Local service* then display *this message to the user*.
- `<ProcessUser>svchost.exe:local service:the processes owning user is unexpected, and may indicate the process is malware</ProcessUser>`

# Process User Verification

- If *process* is being executed by any other user, other than **Local service** then display *this message to the user*.
- `<ProcessUser>svchost.exe:local service:the processes owning user is unexpected, and may indicate the process is malware</ProcessUser>`

# Process User Verification

- If *process* is being executed by any other user, other than *Local service* then display ***this message to the user.***
- `<ProcessUser>svchost.exe:local service:the processes owning user is unexpected, and may indicate the process is malware</ProcessUser>`

# Process User Verification

**Configure Process Username Verification** [X]

Add or remove process username specifications:  
If process  is not launched by username   
then alert the user with the following message

spoolsv.exe:system:the processes owning user is unexpected, and may indicate the process is malware  
smss.exe:system:the processes owning user is unexpected, and may indicate the process is malware  
svchost.exe:local service:the processes owning user is unexpected, and may indicate the process is malware  
svchost.exe:system:the processes owning user is unexpected, and may indicate the process is malware  
svchost.exe:network service:the processes owning user is unexpected, and may indicate the process is malware  
services.exe:system:the processes owning user is unexpected, and may indicate the process is malware  
csrss.exe:system:the processes owning user is unexpected, and may indicate the process is malware

# Process Handle Verification

- If *process* has a handle to *c:\windows\system32\password\_Log.txt* of type *file* then display *this message to the user*.
- `<Handle>svchost.exe:cmd.exe:process:service potentially has a spawned command shell</Handle>`
- `<HandleRegex>*.*-7$:mutant:known conficker mutant</HandleRegex>`

# Process Handle Verification

- If *process* has a handle to *c:\windows\system32\password\_Log.txt* of type *file* then display *this message to the user*.
- `<Handle>svchost.exe:cmd.exe:process:service potentially has a spawned command shell</Handle>`
- `<HandleRegex>*:.*-7$:mutant:known conficker mutant</HandleRegex>`

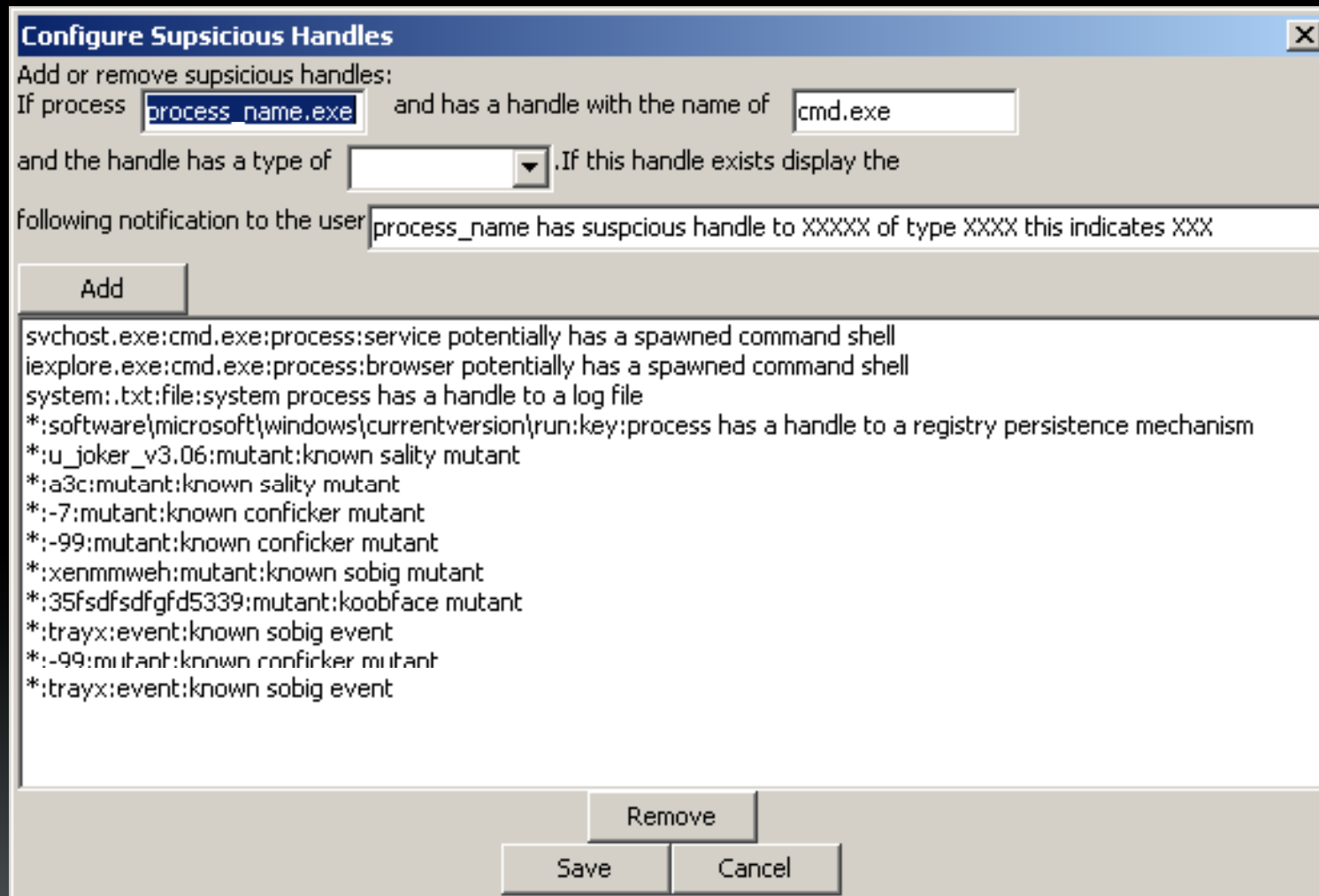
# Process Handle Verification

- If *process* has a handle to *c:\windows\system32\password\_Log.txt* of type *file* then display *this message to the user*.
- `<Handle>svchost.exe:cmd.exe:process:service potentially has a spawned command shell</Handle>`
- `<HandleRegex>*:*-7$:mutant:known conficker mutant</HandleRegex>`

# Process Handle Verification

- If *process* has a handle to *c:\windows\system32\password\_Log.txt* of type *file* then display *this message to the user*.
- `<Handle>svchost.exe:cmd.exe:process:service potentially has a spawned command shell</Handle>`
- `<HandleRegex>*:*-7$:mutant:known conficker mutant</HandleRegex>`

# Process Handle Verification



# MRI

- Scoring
  - Based on digital signatures on disk
    - **NOTE:** Does not indicate if modifications in memory were made.
    - Only checks if it was signed on disk!
  - Any module that occurs in over  $X\%$  of processes is implicitly trusted
    - $X$  is user defined number
    - Default is 75%
  - Any module has a signature, that is verified and the issuer is trusted, is trusted.

# MANDIANT Memoryze

## ENUMERATION

- All running processes
  - Handle table
  - Memory sections
  - Ports
  - Strings
  - Digital Signatures
- Drivers
  - Including layered ones
- Certain kernel hooks

## ACQUISITION

- Physical memory image
- Running processes memory space
  - Binary
  - Loaded DLL's
  - Stacks
  - Heaps
  - Data sections
- Drivers

# MANDIANT Memoryze

- Can analyze memory live, or from image
  - Live analysis can use paging file for a more complete picture of memory
- Supported platforms
  - 32-bit Windows 2000, XP, 2003 Server, Vista
- Download at
  - <http://www.mandiant.com/>

# MANDIANT Audit Viewer

- Visualizes Memoryze Output
  - Add intelligence, virtualization, searching, grouping...
  - Apply snort rules to strings in memory
  - Acquire processes/drivers from live views
- Seeing is believing

# DEMOS

Those that can, do  
And those that can't, teach  
I can, and I DEMO



# DEMO 1

- Actual APT sample from an IR
- Was active when we responded
  
- Who thinks we can do an APT investigation in < 5 minutes?

# DEMO 2

- Actual APT sample from an IR
- Who thinks we can do this in under 1 minute?



# Q&A

- Thank you!
- [peter.silberman@mandiant.com](mailto:peter.silberman@mandiant.com)
  - <http://www.mandiant.com>
  - <http://blog.mandiant.com>
  - <http://www.twitter.com/petersilberman>
- [http://www.mandiant.com/products/free\\_software/memoryze/](http://www.mandiant.com/products/free_software/memoryze/)
- [http://www.mandiant.com/products/research/mandiant\\_audit\\_viewer/](http://www.mandiant.com/products/research/mandiant_audit_viewer/)