Recovering Executables with Windows Memory Analysis
Overview

- Why Memory Analysis
- Windows without Windows
- Gathering Information
- Parsing the Processes
- The Rootkit Paradox
- Recovering Executables
- Fuzzy Hashing
- Getting memory images
- Birds of a Feather
Introduction
Encryption Keys
- BitLocker, PGP Whole Disk Encryption, etc.

What was happening on the system
- Running programs, open (unsaved) documents
- Unpacked contents of packed programs
- Network connections

What was really happening on the system
- Not the sanitized (lying) version from the OS
- Hidden programs, rootkits, injected code
- Destroy the hacker defense

What was really happening on the system
- What was running ten minutes ago
Windows without Windows

- We must do everything the OS would do
- Windows is complex
  - Minor understatement
- Thousands of structures
  - Many change between versions
  - Mostly undocumented
- Hacks on top of short cuts on top of optimizations on top of…
Windows without Windows
Windows without Windows

- **Microsoft Windows Internals**
  - Russinovich and Solomon
  - Yes, the SysInternals guy
Other Resources

- Intel
- Microsoft Documentation
- Malware authors
Getting Started

- Brute force search through image
  - Processes
  - Network connections
  - Application specific information
High/Low Comparison

- Compare Windows structures to brute force search results
  - Similar to rootkit detectors
Active Process Links
Suspicious Processes

- Linked Process is in memory image but not OS list
  - Hidden process
  - API Hooking
  - Hacker Defender rookit, et al.

- Linked Process in OS list but not memory
  - Phantom process
  - Never seen, but could happen
  - “Yes, your anti-virus program is running! Really!”
Suspicious Processes

- Find both running and terminated processes
- Terminated means it ended
  - Yes, Windows really records this
  - Including the time it ended
- Running process means either
  1. Unlinked process
  2. Leftover from previous boot
- Compare start time versus boot time
Sure!

RAM is not wiped at boot

It used to be wiped
  - RAM test at startup
  - Took too much time, so now disabled by default
Suspicious Processes

- Unlinked Active Processes
  - Found during brute search and not in OS list
  - The result of Direct Kernel Object Manipulation
  - Almost certainly the work of malicious code
Process Information

- Full name and path
- Command line arguments
- Process ID number (PID)
- Parent PID
- Current directory
- Window Title
- Handles
  - Files, devices, drivers
- List of loaded modules
  - DLLs
Process Information

- Suspicious program names
  - UMGR32.EXE

- Suspicious command lines
  - C:\TEMP\backdoor.exe -r 63.161.169.137 –stealth
  - X:\h4x0r-Toolkit\notepad.exe
  - c:\windows\system32\cmd.exe
Process Information

- Most system processes have well defined parents
  - cmd.exe should not be the parent of lsass.exe

- Most user processes are started by Explorer.exe

- It’s suspicious when they’re not
  - Maybe started from a command prompt
  - Orphaned processes

- Some system processes should never start programs
  - lsass.exe should not start cmd.exe
Process Information

- List of DLLs for each process
  - We get the name, path, and size of each
- What is notepad.exe doing with wsock32.dll?
- What is iexplore doing with c:\temp\wsock32.dll?
- What if there is no path information?
  - Injected code!
  - First part of the rootkit paradox
  - If they supply fake data, where is it on the disk?
The more a rootkit tries to hide, the easier it is to see

All rootkits obey two rules
1. They don’t want to be seen
2. They want to execute

We will see you in the operating system or in memory analysis, your choice
The Rootkit Paradox

- Anything can be subverted
  - Especially memory acquisition
- Normally, errors don’t happen
- If rootkit creates error, it’s just indicated its presence!
  - The lack of data is in the indicator
Recovering Executables

- We have pointers to recover each module
- Recreate files (mostly) as they existed on the disk
- First page of module has PE header
  - Names, sizes, and locations of sections
  - Locations both in memory and on disk
- Can read sections from memory and write back to disk
Recovering Executables

On Disk

In Memory

Recovered
Recovering Executables

- Look at section names
  - Normal values are things like .code, .data, .text
- Packed programs
  - Self decompressing or decrypting
- Well known packing program UPX
- Also check for sections that are zero bytes on disk
  - Data is decompressed on load into these sections
  - Indicate packed program
- Some packed programs are “ok”
  - Skype
Recovering Executables

On Disk

In Memory

Recovered

PE

PE

PE

PE

PE

Packed Section
Recovering Executables

- Recovered programs not identical to versions on disk
  - Resource mappings change
  - Program variables
- If only we had a tool to match slightly different versions of a file back to the originals...
Disclaimer

- I didn’t invent this math
- Originally Dr. Andrew Tridgell
  - Samba
  - rsync was part of his thesis
  - Modified slightly for spamsum
  - Spam detector in his “junk code” folder
- User report that rsync confuses similar Word documents
MD5 Explained

How MD5 (roughly) works:

1. Start with an initial state
2. Look at fixed size block of input
   - Do mathy stuff with current state and block
   - Get new state
3. Advance to next block of input
4. Repeat steps 2 and 3 until out of input blocks
5. Ending state is the hash
MD5 Explained

- If you change one bit in the middle, you change the next state
- Which ends up changing the end result

- Is this a good thing or a bad thing?
**Piecewise Hashing**

- Developed for integrity during imaging
- Divide input into fixed sized sections and hash separately
- Insert or delete changes all subsequent hashes

```
3b152e0baa367a8038373f6df
40c39f174a8756a2c266849b
fdb05977978a8bc69ecc46ec
```
It would be nice to set boundaries such that
- insertions and deletions are contained within a block
A different kind of hash function

- Produces a pseudorandom output for every position in a file
  - Depends only on last few bytes
  - Lots of academic work on these
  - Just mathy tricks

Four score -> 83,742,221
Four score -> 5
Four score -> 90,281
Rolling Hash

To update state \((c,x,y,z,\text{window})\) for a byte \(d\):

\[
\begin{align*}
y &= y - x \\
y &= y + \text{size} \times d \\
x &= x + d \\
x &= x - \text{window}[c \mod \text{size}] \\
\text{window}[c \mod \text{size}] &= d \\
c &= c + 1 \\
z &= z \ll 5 \\
z &= z \text{ XOR } d \\
\text{return } (x + y + z)
\end{align*}
\]
Rolling Hash

- We use the rolling hash to generate block boundaries
- Select some values as trigger points
- When we hit a trigger point, end the block

Example
- Excerpt from "The Raven" by Edgar Allan Poe
- Triggers on ood and ore
Deep into the darkness peering, long I stood there, wondering, fearing
Doubting, dreaming dreams no mortals ever dared to dream before;
But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word,
Lenore?, This I whispered, and an echo murmured back the word,
"Lenore!" Merely this, and nothing more
Deep into the darkness peering, long I stood there, wondering, fearing

Doubting, dreaming dreams no mortals ever dared to dream before;
But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word,
Lenore?, This I whispered, and an echo murmured back the word,
"Lenore!" Merely this, and nothing more
Deep into the darkness peering, long I stood

there, wondering, fearing Doubting, dreaming dreams no mortals ever dared to dream before

; But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word,Lenore

?, This I whispered, and an echo murmured back the word,"Lenore

!" Merely this, and nothing more
Rolling Hash

- How do we choose the triggers?
  - Chosen randomly, before reading the file
  - Based on the size of the input file
  - Really just a set of numbers
  - Has nothing to do with type of input data
Fuzzy Hashing

- Combine Rolling Hash with a Traditional Hash
- Use Fowler/Noll/Vo (FNV) hash
  - That’s what Tridgell did
  - Faster and less complex than MD5
  - We’re only using a small part of the result

- Start reading file, compute Rolling and Traditional Hashes
- When Rolling Hash triggers
  - Record LSB of Traditional Hash value
- When finished, combine LSBs to make signature
Deep into the darkness peering, long I stood

there, wondering, fearing Doubting, dreaming dreams no mortals
ever dared to dream before

; But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word,Lenore

?, This I whispered, and an echo murmured back the word,"Lenore

!" Merely this, and nothing more
Deep into the darkness peering, long I stood
there, wondering, fearing Doubting, dreaming dreams no mortals
ever dared to dream before
But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word, Lenore
This I whispered, and an echo murmured back the word, "Lenore"
Merely this, and nothing more
Deep into the darkness peering, long I stood
there, wondering, fearing Doubting, dreaming dreams no mortals
ever dared to dream before

; But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word,Lenore

?, This I whispered, and an echo murmured back the word,"Lenore"
!
" Merely this, and nothing more
Deep into the darkness peering, long I stood

there, wondering, fearing Doubting, dreaming dreams no mortals ever dared to dream before

; But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word,Lenore

?, This I whispered, and an echo murmured back the word,"Lenore

!" Merely this, and nothing more
Deep into the darkness peering, long I stood
there, wondering, fearing **I AM THE LIZARD KING!** Doubting, dreaming dreams no mortals ever dared to dream before

; But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word,**Lenore**

?, This I whispered, and an echo murmured back the word,**"Lenore**

!" Merely this, and nothing more
Deep into the darkness peering, long I stood there, wondering, fearing I AM THE LIZARD KING! Doubting,
dreaming dreams no mortals ever dared to dream before

; But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word,Lenore

?, This I whispered, and an echo murmured back the word,"Lenore"

!" Merely this, and nothing more
Deep into the darkness peering, long I stood there, wondering, fearing I AM THE LIZARD KING! Doubting, dreaming dreams no mortals ever dared to dream before; But the silence was unbroken, and the stillness gave no token, And the only word there spoken was the whispered word, Lenore, This I whispered, and an echo murmured back the word, "Lenore!" Merely this, and nothing more
Matching

Signature 1:  3 2 7 3 0
Signature 2:  3 0 7 3 0

- **Edit Distance**
  - Number of insertions, modifications and deletions to turn Signature 1 into Signature 2.
  - For the example above, the edit distance is one.

- **Signatures (and thus files) match when the ratio of the edit distance to the length is small**
LAW ENFORCEMENT SENSITIVE
DO NOT DUPLICATE

WARNING:
EXPLICIT IMAGERY
Corrupted File

Known kitty porn

MATCH
Demonstration

Different File

Known kitty porn

No match
Demonstration

File Header

Known kitty porn MATCH
Demonstration

Known kitty porn MATCH
Demonstration

File Footer (attached to header)

Known kitty porn MATCH
Issues

- Does not work for similar looking graphics
- Unable to handle cropping, resizing, and other edits
- Confused by many small changes throughout input
- Computationally intensive
  - 7-10 times slower than MD5
- No way to sort signatures
  - Must compare each input to all known signatures
Fuzzy Hashing

- Matches similar but not identical bitstreams
  - Great for corrupted or partial documents
  - Also great for source code reuse
- Freely available
  - http://ssdeep.sf.net/
  - Windows, Windows GUI, *nix, and OS X
- Academic paper
Recovering Modules

- Recovered programs not identical to versions on disk
  - Resource mappings change
  - Program variables
- If only we had a tool to match slightly different versions of a file back to the originals...

```
C:\> ssdeep -rlm known-good-winxpssp2.txt recovered\*
recovered\pid-018c\c__\windows\notepad.exe matches winxpssp2\c__\windows\notepad.exe (94)
```
Address Translation

- Until now, ignored address translation
- Programs use virtual addresses
  - Keeps things simple
  - Avoids hardware dependence
- Must be translated to physical address
  - In memory image
  - Pagefile
- Using Every Part of the Buffalo in Windows Memory Analysis
  - Next issue of Digital Investigation
Naïve Translation

Virtual Memory

Memory Image
Robust Translation

Virtual Memory

Memory Image

Prototype

Page File
Getting Memory Images

- Hardware debugger
  - Best, but hard to do
  - Especially for incident response
- Suspend virtual machine
- dcfldd, dd and the \Device\PhysicalMemory object
  - Not like a disk image
  - Data changed during acquisition and by acquisition
  - Requires device driver for Windows 2003 SP 1 and above
- Microsoft COFEE tool
  - Law enforcement only
  - Does not require a driver (unverified)
  - mem_p and mem_d
Getting Memory Images

- DFRWS Memory Images
  - Two Windows 2000 Service Pack 1 images
  - Malware on both

- Digital Forensics Tool Testing
  - http://dftt.sourceforge.net/test13/
  - Boomer is multiprocessor
Analysis Tools

- No major commercial support
Analysis Tools

- **FATKit** – http://www.4tphi.net/
  - Best set of links
  - Volatools coming soon…

- **Chris Betz - Memparser** -
  http://sourceforge.net/projects/memparser

- **The rest of the field:**
  - Harlan Carvey, AAaron Walters, Nick Petroni, Tom Goldsmith and Tim Vidas are here today
  - Andreas Schuster - ptfinder
  - Mariusz Burdach - WMFT
  - Nicholas Maclean
  - George Garner - KnTTools
  - Joe Stewart - pmodump (TRUMAN)
Birds of a Feather

Right after this talk in Landmark 1
Conclusion

- Why Memory Analysis
- Windows without Windows
- Gathering Information
- Parsing the Processes
- The Rootkit Paradox
- Recovering Executables
- Fuzzy Hashing
- Getting memory images
- Birds of a Feather
Questions

Jesse Kornblum
jesse.kornblum@mantech.com