Memory Forensics

An introduction
DISCLAIMER

- Speak only for myself
- These are opinions, not facts
- I could be wrong about anything
- Use at your own risk
About Me

• On corporate security team
• Analyze malware as a hobby
• Not an expert by any stretch
• Goal for talk:
  • Introduce concepts, show fun demos
Agenda

• Introduction
• Concepts
• Acquisition methods (demo!)
• Analysis (demo!)
• Wrap-up
• Links, links, links
Agenda

• Introduction
• Concepts
• Acquisition methods (demo!)
• Analysis (demo!)
• Wrap-up
• Links, links, links
Types of Forensics

- Disk/filesystem
- Network/signals
- Memory/volatile
Why Memory?

- Unpacked binary
- Observe behavior
- Encryption keys
- Memory-only malware
- Memory-only artifacts
Agenda

• Introduction
• Concepts
• Acquisition methods (demo!)
• Analysis (demo!)
• Wrap-up
• Links, links, links
What Does Memory Look Like?

- Objects: Linked lists, structs, mapped files
- Process lists, sockets, file handles, jump tables, registry hives
- Memory pages—different access privileges
- Process space, global & local variables
Memory Layout (Virtual address space of a C process)

**STACK**

- System
  - env
  - argv
  - argc
  - auto variables for main()
  - auto variables for func()
  - available for stack growth

- heap
  - malloc.o (lib*.so)
  - printf.o (lib*.so)
  - available for heap growth

- Heap (malloc arena)

- global variables

- "...%d...

- malloc.o (lib*.a)

- printf.o (lib*.a)

- file.o

- main.o (compiled code)

- creset.o (startup routine)

**DATA**

- uninitialized data (tex)

- initialized data

- library functions if dynamically linked
  - (usual case)

- library functions if statically linked
  - (not usual case)

- ra (return address)

**Low memory**

Stack illustrated after the call
func(72,73) called from main(),
assuming func defined by:

```
func(int a, int y)
{
    int a;
    int b[3];
    /* no other auto variables */

    Assumes int = long = char type of
    size 4 and assumes stack at high
    address and descending down.
```

**Expanded view of the stack**

- mfp - frame pointer (for main)

- stack pointer
  - (grows downward if func() calls another function)

- bkp point

- stack pointer
  - (top of stack)
  - points here

- frame pointer
  - points here

- offset from current frame pointer (for func)
  - main()
  - auto variables

- contents
  - y
  - x
  - return address
caller's frame pointer
  - a
  - b[2]
  - b[1]
  - b[0]

All auto variables and parameters are referenced via offsets from the frame pointer.

The frame pointer and stack pointer are in registers (for fast access).

When func returns, the return value is stored in a register. The stack pointer is
move to the y location, the code is jumped to the return address (ra),
and the frame pointer is set to mfp
(the stored value of the caller's frame pointer). The caller moves the return
value to the right place.

http://www.cs.uleth.ca/~holzmann/C/system/memorylayout.gif
Sidebar...

- Security pros need deeper knowledge than other tech pros
- Ex: Developer, how inputs are handled
- Ex: Sysadmin, how kernel & filesystem work
Agenda

• Introduction
• Concepts
• **Acquisition methods (demo!)**
• Analysis (demo!)
• Wrap-up
• Links, links, links
Software

- Access raw device
- Install custom driver/kernel module
- Swap file on disk
- Hibernation image on disk
  - hiberfil.sys (Win)
  - sleepimage (OSX)
Examples

- Memoryze & Memoryze for the Mac
- LiME
- F-Response
- FTK Imager
- DumpIt
- FastDump Pro

http://www.forensicswiki.org/wiki/Tools:Memory_Imaging
Direct Memory Access

“Systems may be vulnerable to a DMA attack by an external device if they have a FireWire, ExpressCard, Thunderbolt, or other expansion port that, like PCI and PCI-Express in general, hooks up attached devices directly to the physical address space.”

http://en.wikipedia.org/wiki/DMA_attack
CaptureGUARD Physical Memory Acquisition Hardware - ExpressCard

This is an ExpressCard device capable of imaging the physical memory of the computer it's connected to. Creates dump files in the standard WinDD format that can be used with WindowsSCOPE Cyber Forensics Ultimate or with other WinDD compatible dump analysis tools. Connects directly to the physical memory to read contents. Requires a small CaptureGUARD driver for the device to be recognized and to store memory contents to file.

Specifications

http://www.windowsscope.com

http://www.breaknenter.org/projects/inception/

http://digitalfire.ucd.ie/?page_id=430

http://macfwdump.sourceforge.net/
Cold-boot

“The attack relies on the **data remanence** property of **DRAM** and **SRAM** to retrieve memory contents which remain readable in the seconds to minutes after power has been removed.”

http://en.wikipedia.org/wiki/Cold_boot_attack
https://citp.princeton.edu/research/memory/
Figure 6: Before powering off the computer, we spray an upside-down canister of multipurpose duster directly onto the memory chips, cooling them to $-50^\circ{}C$. At this temperature, the data will persist for several minutes after power loss with minimal error, even if we remove the DIMM from the computer.
DEMO (click me!)

Build LiME
Create Volatility profile
Dump memory over TCP
Find bash history
Important Notes!

• Don’t build LiME or mem profile on victim!
• Use virtual machine with same OS/kernel
• Build module & profile ahead of time
  • if you can (speed up response)
• Requires gcc, gdb, make, etc
Agenda

- Introduction
- Concepts
- Acquisition methods (demo!)
- Analysis (demo!)
- Wrap-up
- Links, links, links
Suspicious Signs

- Handles to other processes
- Missing from one or more process list
- Has injected sections
- Holds suspicious mutex
• Direct Kernel Object Manipulation
  • Unlink process from _EPROCESS list
  • CSRSS process also has handles
  • and internal list
Process Injection

• Process Environment Block
• Command line & arguments
• Three lists of the loaded DLLs
  • Could unlink list, but VAD has map
  • Tampering w/VAD requires rootkit

• Process hollowing (similar to injection)
• Start legit binary in suspended thread
  • Replace the image, resume thread
• Mutex
• Ensure only one copy of malware runs
  • or avoid concurrency w/specific prog
Failed to import volatility.plugins.registry.lsadump

failed to find module named Crypto.Hash

administrator!cookies!

ZonesCacheCounterMutex
ZonesCounterMutex
\^??
??
WPA PR_MUTEX
RasPbFile
SHIMLIB_LOG_MUTEX
ShimCacheMutex

!VoqA.I4

c!documents and settings!a
local service!local settings!temporary internet files!content.ie5!
RAS_MO_01
SingleSesMutex
c!documents and settings!a
local service!cookies!

C:\>EnumerateMutex.exe


http://pmelson.blogspot.com/2012/10/grrcon-2012-forensics-challenge.html
DEMO (click me!)

Collect artifacts to net share
Import artifacts to Redline
Discover injected memory
Locate events in timeline

(Not shown: Creating the collector)
Agenda

• Introduction
• Concepts
• Acquisition methods (demo!)
• Analysis (demo!)
• Wrap-up
• Links, links, links
Wrap-up

- Memory forensics offer unique advantages
- Concealment techniques leave a trail
- Tools can help, but knowledge is required
  - Study system internals
- Many free tools & guides exist
  - Barrier to entry is low!
Pop Quiz
Pop Quiz

• Name one interface for DMA attack
Pop Quiz

- Name one interface for DMA attack
- What does DKOM stand for?
Pop Quiz

- Name one interface for DMA attack
- What does DKOM stand for?
- Name a software memory acquisition tool
Agenda

• Introduction
• Concepts
• Acquisition methods (demo!)
• Analysis (demo!)
• Wrap-up
• Links, links, links
ERMAHGERD

BERKS

http://www.quickmeme.com/meme/3otxsn/

N BLERGS
Malware Analyst’s Cookbook and DVD
http://www.malwarecookbook.com/

SecurityXploded
http://securityxploded.com/malware-memory-forensics.php

Forensics Wiki
http://www.forensicswiki.org/

Gustavo Duarte
http://duartes.org/gustavo/blog/

Windows Incident Response
http://windowsir.blogspot.com/

Journey Into Incident Response
http://journeyintoir.blogspot.com/

contagio malware dump
http://contagiodump.blogspot.com/

Practical Malware Analysis
http://practicalmalwareanalysis.com/

DigitalFIRE
http://digitalfire.ucd.ie/

Memory Forensics
http://memoryforensics.blogspot.com/

APTish Attack via Metasploit
http://www.sysforensics.org/

Linux Sleuthing
http://linuxsleuthing.blogspot.com/

DeepEnd Research
http://www.deependresearch.org/

SEMPERSECURUS
http://sempersecurus.blogspot.com/
Memoryze
http://www.mandiant.com/resources/download/memoryze

Memoryze for the Mac
http://www.mandiant.com/resources/download/mac-memoryze

LiME
https://code.google.com/p/lime-forensics/

Inception
http://www.breaknenter.org/projects/inception/

Volatility
https://www.volatilesystems.com/default/volatility

Redline
http://www.mandiant.com/resources/download/redline

Yara
http://code.google.com/p/yara-project/

Cuckoo Sandbox
http://www.cuckoosandbox.org/
Thanks!
Brian Keefer
http://rants.effu.se
https://twitter.com/chort0
https://alpha.app.net/chort
http://www.SMTPS.net
chort0 on Freenode