Memory Analysis
Q-CERT Workshop

Matthew Geiger
mgeiger@cert.org
Outline

Why live system forensics?
  • Previous techniques
  • Drawbacks and new thinking

Approaches to memory acquisition

Evolution of memory analysis
  • Survey of tools & methodologies
  • The investigation gap & the future
Memory Analysis & Forensics

Increasing recognition in the forensics community that:

Advances in counter-forensic techniques
  • Metasploit Meterpreter
  • Malcode stealth strategies

Pervasive encryption

Can focus search for evidence ... in some cases
Evolution of Technique

Live forensics borrowed from incident response

• Scripted queries using response toolkits (MS COFEE)
• Often still used initial results to guide data collection
  – netstat → ps → lsof ... etc
• Tension between speed and thoroughness

Memory acquisition

• 800lb gorilla of forensics – what do you do with it?
• String dumps, virus scans, signature-based carving
Old School ↔ New School

Old School:
- Iterative & invasive
- Double the opportunity for subversion:
  - Data collection
  - Interpretation
- One-way, ephemeral information channel
- Memory collection requires privileged access

New School:
- Working groups developing accepted practices
- New analysis tools extract familiar information – and more – from memory
- Repeatable results
- Novel acquisition techniques & tools:
  - Increase assurance
  - Bypass access controls
Memory Acquisition – Software

Software mediated

• Crash / core dump
• WinHex, other applications – secondary functionality
• dd.exe
• Commercial enterprise forensics packages: EnCase, ProDiscover, etc

Access restrictions on `\\PhysicalMemory`

Kernel-mode window needed

• Commonly use driver installation routines
• George Garner’s KnTTools released in 2007
• AccessData, other vendors are working on it
Memory Acquisition – Hardware

Hardware-based

- Komoku CoPilot, BBN Tech, Tribble
- IEEE 1394

Can extend DMA access to PC Card and Express Card devices

Developing field-deployable memory acquisition unit
Live Forensics Evolutionary Tree

Memory Analysis Branch

- String searches, file carving
- PE “analysis” & carving
- Informed analysis: state & context reconstruction

Run-Time Analysis Branch

- IR-style running system investigation
- Controlled runtime analysis
Recent History

Tool development inspired by DFRWS 2005 challenge
  - Two entries shared prize: Garner/Mora & Betz
  - Tools released at subsequent conferences step up pace

Flurry of subsequent activity
  - Mariusz Burdach – WMFT (plus Linux tools)
  - Andreas Schuster – PTFinder, PoolFinder
  - Harlan Carvey – Focused Perl utilities
  - Garner – KnTTTools / KnTList
  - Jesse Kornblum – Buffalo tool
  - Walters/Petroni – Volatility
Two Paths to Memory Reconstruction

Tree & list traversal

- Memparser
- KnTList
- WMFT
- Volatility

Object “fingerprint” searches

- PTFinder / PoolFinder
- Volatility
List Traversal Basics

Find index into lists and tables of interesting structure

- Kernel image needed for offsets & symbols that help find a number of these

- Addresses can change from SP to SP
  - Copy of NT kernel part of KnTTools acquisition process
  - Other approach is to build hardcoded tool modules for each

EPROCESS linked list is a common example, with pointers to

- _ETHREAD structures
- SID of starting user
- Start time, PID, other metadata in PEB
- Process virtual memory pages

These structures allow reconstruction of some familiar IR-style data
Volatility Framework

- At present, most actively developed open-source tool in this space
  - Running processes, DLLs loaded for each
  - Open network sockets, network connections
  - Open files handles for each process
  - System modules
  - Mapping interesting strings to process (physical offset to virtual address translation)
  - Virtual Address Descriptor information

- Recently added pattern-scanning tools
  - processes & threads
  - sockets & connections

- Framework approach intentionally maintains IR feel
Fingerprint Searching Basics

Scan for sufficiently unique structure signatures
  • PTFinder works with EPROCESS, ETHREAD structs
  • PoolFinder parses kernel pool memory

Perform basic sanity checks on data to weed out corrupt records, duplicates

PTFinder doesn't perform further analysis but does provide optional graphical output
### Pros

<table>
<thead>
<tr>
<th>Pattern search</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Find unlinked, dead structures (warm reboot)</td>
</tr>
<tr>
<td>• Can work with imperfect dumps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List traversal</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can stitch together more related records from kernel perspective</td>
</tr>
</tbody>
</table>

### Cons

<table>
<thead>
<tr>
<th>Pattern search</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Less context without following related structures/objects</td>
</tr>
<tr>
<td>• Susceptible to chaff</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List traversal</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can miss unlinked, dead structures</td>
</tr>
<tr>
<td>• Targeted counter-measures</td>
</tr>
</tbody>
</table>
Enhanced Techniques

Pagefile incorporation

Combining “naive” pattern searches with list-traversal techniques

• Cross-view analysis
• Defense against chaff

Highlighting potentially interesting situations

• Orphaned threads still referenced in other structures
• Executable segments not mapped into shared sections
What's next

Specialized tools will bridge the investigative gap
- Focus now centers on malcode, execution state analysis – but the investigative mission is much broader
- Recovery of cryptographic material to defeat disk encryption

Forensic platform vendors making friendlier analysis tools
- Bring some analysis tasks into mainstream
- Provide momentum to adoption of memory analysis
- Automate extraction of typically interesting data
- Provide better anomaly detection or flags

Court cases and working groups will hammer out standards
Questions / Comments?
References

*PTFinder* - by Andreas Schuster
http://computer.forensikblog.de/en/2006/09/ptfinder_0_3_00.html

*Volatility* - by AArnon Walters and Nick Pedroni Jr.
http://www.volatile-systems.com/VolatileWeb/volatility.gsp

Brian Carrier and Joe Grand's work on hardware-based memory acquisition

George Garner's *KnTTools and KnTList* memory acquisition and analysis suite
http://www.gmgsystemsinc.com/knttools/

Mariusz Burdach – *Windows Memory Forensic Toolkit*
http://forensic.seccure.net/

Harlan Carvey's *memory tools*
http://sourceforge.net/project/showfiles.php?group_id=164158

Chris Betz's *Memparsear*
http://sourceforge.net/project/showfiles.php?group_id=167028