Minding the Gap(s) in Memory Forensics

Golden G. Richard III

Professor of Computer Science and Engineering
Associate Director for Cybersecurity
Center for Computation and Technology (CCT)
Louisiana State University

golden@cct.lsu.edu / @nolaforensix
Condensed "Me" Timeline

- Created cybersecurity program at UNO
- Founding Director, GNOCIA
- NSA CAE, CAE-R, CAE Cyber Ops Designations
- Fellow of the American Academy of Forensic Sciences

- NSA Employee, TS/SCI Clearance
- CAE-CO Internship Program
Research and Teaching

- Reverse Engineering
- Memory Forensics
- Malware Analysis
- Digital Forensics
- Exploit Development
- Penetration Testing
- Secure Programming
- Network Security
- Darknet Technologies
- Blockchains / Smart Contract Security

Disclaimer!
Collaborators, Students, and Support

Andrew Case     Nathan Lewis     Joe Sylve     Lodovico Marziale

Aisha Ali-Gombe     Brendan Saltaformaggio

Nathan Lewis     Arian Shahmirza     Raphaela Mettig     Sneha Kalliat

Austin Orgah     Modhu Manna     Ryan Maggio

Irfan Ahmed     Shravya Pararuchuri     Devi Sowjanya     M. Jalalzai

Md Firoz-Ul-Amin     Mingxuan Sun

and others!

• Some of the research in this talk supported by NSF:
  – SaTC: CORE: Medium: Robust Memory Forensics Techniques for Userland Malware Analysis, Award # 1703683, PI: Golden G. Richard III, $1,113,426
Digital Forensics

“Tools and techniques to recover, preserve, and examine digital evidence stored on or transmitted by digital devices.”

Computers, PDAs, cellular phones, videogame consoles, digital cameras, copy machines, printers, digital voice recorders...
Where’s the Evidence?

Non-volatile Evidence

- Files and Deleted Files
- Filesystem metadata
- Application metadata
- Windows registry
- Print spool files
- Hibernation files
- Temp files
- Log files
- Slack space
- Swap files
- Browser caches
- Network traces

Memory forensics

RAM: OS and app data structures

Volatile Evidence
Memory Forensics

Important Kernel and Application Data Structures

Passwords and Encryption Keys

Previously Executed Commands

Filesystem Cache

Volatile Registry Branches

Executing Malware!

Some data (and malware) is only in RAM!
Goal in memory forensics is to build an accurate, deep model of the entire kernel state as well as application states.

- Kernel data structures
  - Process descriptors
  - Thread descriptors
  - Memory allocation
  - Networking structures
  - Timers
  - Hooks
  - ...lots more

- Virtual address spaces for processes
  - Access to all code, data, heap, stack, et al
  - Snapshot of language runtime
Memory Analysis: 2004

$ grep -i murder /dev/mem

I loved Sally, but I murdered her in the park on...
Murder
murder
Murderous
You murdered my hamster!
Murdered
Are kickturns on vertical walls possible?
Initial Challenges

Acquisition

Understanding
Memory Analysis Timeline

• strings / grep / etc.
  – essentially no tools circa 2004

• DFRWS memory forensics challenge (2005)

• pt_findew (~2006)
  – Windows process enumeration

• FATKIT (~2006)
  – Predecessor of Volatility

• FACE (~2007-8)
  – Memory forensics framework for Linux created at the University of New Orleans

• Community supported frameworks, e.g., Volatility
Schuster’s PTfinder

Introduced pool tag scanning technique, widely used in Windows memory forensics
Memory Analysis: Now

Capture RAM from live system

- physical memory dumping tool
- VM memory snapshot
- VM introspection

Analyze Memory Dump

- strings
- carving
- Volatility / Rekall

Expose OS and Application Structures

- to yield useful evidence
Volatility Framework

• Most popular memory analysis framework
• Completely open source
• Portable, written in Python 😊
• Supports analysis of Windows, Linux, Mac, Android memory dumps
• Plugins add new functionality, build on vast amount of research that’s already been done
• Lots of existing plugins, fairly straightforward to add new ones
Memory Analysis: Now

Use plugins to analyze:

- Running processes
- Hidden processes
- Hooks that hide malware
- Network connections
- Encryption keys
- Private browsing data
- Clipboard data
- Volatile registry branches
- Command history
- Window hierarchy
+ "easily" develop plugins
## Process Lists

```bash
$ python vol.py -f lab.mem --profile=WinXPSP3x86 pslst
```

<table>
<thead>
<tr>
<th>Offset(V)</th>
<th>Name</th>
<th>PID</th>
<th>PPID</th>
<th>Thds</th>
<th>Hnds</th>
<th>Sess</th>
<th>Start</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x823c8830</td>
<td>System</td>
<td>4</td>
<td>0</td>
<td>56</td>
<td>537</td>
<td></td>
<td>2013-03-14 03:02:22</td>
</tr>
<tr>
<td>0x81e7e180</td>
<td>smss.exe</td>
<td>580</td>
<td>4</td>
<td>3</td>
<td>19</td>
<td></td>
<td>2013-03-14 03:02:25</td>
</tr>
<tr>
<td>0x82315da0</td>
<td>csrss.exe</td>
<td>644</td>
<td>580</td>
<td>10</td>
<td>449</td>
<td>0</td>
<td>2013-03-14 03:02:26</td>
</tr>
<tr>
<td>0x81f37948</td>
<td>winlogon.exe</td>
<td>668</td>
<td>580</td>
<td>18</td>
<td>515</td>
<td>0</td>
<td>2013-03-14 03:02:27</td>
</tr>
<tr>
<td>0x81fec128</td>
<td>services.exe</td>
<td>712</td>
<td>668</td>
<td>15</td>
<td>281</td>
<td>0</td>
<td>2013-03-14 03:02:27</td>
</tr>
<tr>
<td>0x81eb4300</td>
<td>vmtoolsd.exe</td>
<td>1684</td>
<td>1300</td>
<td>6</td>
<td>213</td>
<td>0</td>
<td>2013-03-14 03:02:45</td>
</tr>
<tr>
<td>0x8210b9c8</td>
<td>IEXPLORE.EXE</td>
<td>1764</td>
<td>1300</td>
<td>16</td>
<td>642</td>
<td>0</td>
<td>2013-03-14 03:03:04</td>
</tr>
<tr>
<td>0x81e79020</td>
<td>firefox.exe</td>
<td>180</td>
<td>1300</td>
<td>27</td>
<td>447</td>
<td>0</td>
<td>2013-03-14 03:03:05</td>
</tr>
<tr>
<td>0x81cb63d0</td>
<td>wuaclt.exe</td>
<td>1576</td>
<td>1072</td>
<td>3</td>
<td>104</td>
<td>0</td>
<td>2013-03-14 03:03:40</td>
</tr>
<tr>
<td>0x81e86bf8</td>
<td>alg.exe</td>
<td>1836</td>
<td>712</td>
<td>5</td>
<td>102</td>
<td>0</td>
<td>2013-03-14 03:04:00</td>
</tr>
<tr>
<td>0x8209eda0</td>
<td>wsctfy.exe</td>
<td>2672</td>
<td>1072</td>
<td>1</td>
<td>28</td>
<td>0</td>
<td>2013-03-14 03:04:01</td>
</tr>
<tr>
<td>0x82013340</td>
<td>jucheck.exe</td>
<td>2388</td>
<td>1656</td>
<td>2</td>
<td>104</td>
<td>0</td>
<td>2013-03-14 03:07:45</td>
</tr>
<tr>
<td>0x81e79418</td>
<td>thunderbird.exe</td>
<td>3832</td>
<td>1300</td>
<td>30</td>
<td>339</td>
<td>0</td>
<td>2013-03-14 03:12:54</td>
</tr>
<tr>
<td>0x8202b398</td>
<td>AcroRd32.exe</td>
<td>3684</td>
<td>180</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2013-03-14 14:19:16</td>
</tr>
<tr>
<td>0x81ecd3c0</td>
<td>cmd.exe</td>
<td>3812</td>
<td>3684</td>
<td>1</td>
<td>33</td>
<td>0</td>
<td>2013-03-14 14:19:29</td>
</tr>
<tr>
<td>0x81f55bd0</td>
<td>a[1].php</td>
<td>2280</td>
<td>3812</td>
<td>1</td>
<td>139</td>
<td>0</td>
<td>2013-03-14 14:19:30</td>
</tr>
<tr>
<td>0x8223b738</td>
<td>IEXPLORE.EXE</td>
<td>2276</td>
<td>2280</td>
<td>7</td>
<td>280</td>
<td>0</td>
<td>2013-03-14 14:19:32</td>
</tr>
<tr>
<td>0x822c8a58</td>
<td>AcroRd32.exe</td>
<td>2644</td>
<td>180</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2013-03-14 14:40:16</td>
</tr>
</tbody>
</table>
Parent / Child Relationships

```
$ python vol.py -f lab.mem --profile=WinXPSP3x86 pstree
Volatility Foundation Volatility Framework 2.4

[snip]

0x82263378:explorer.exe  1300  1188   11  363 2013-03-14 03:02:42
. 0x81e85da0:TSVNCache.exe  1556  1300    7  53 2013-03-14 03:02:43
. 0x81e79020:firefox.exe    180  1300   27 447 2013-03-14 03:03:05
.. 0x8202b398:AcroRd32.exe  3684   180    0 ------ 2013-03-14 14:19:16
... 0x81ecd3c0:cmd.exe      3812  3684    1  33 2013-03-14 14:19:29
.... 0x81f55bd0:a[1].php    2280  3812    1 139 2013-03-14 14:19:30
..... 0x8223b738:IEXPLORE.EXE  2276  2280    7 280 2013-03-14 14:19:32
```
Cross-referencing OS Kernel Structures

- Malware can easily hide from analysis tools that rely on one information source
  - e.g., the kernel's process list

- Expand scope by cross-referencing kernel structures that provide information about executing processes

```
pid 4 <-> pid 37 <-> pid 2260 <-> pid 578 <-> pid 3288
```
DKOM under Windows

Example of DKOM: Direct Kernel Object Manipulation
C:\> fu –ph 2260

Processes continue to run because Windows scheduler handles threads, not processes

Doubly-linked process list in Windows kernel
<table>
<thead>
<tr>
<th>Image Name</th>
<th>PID</th>
<th>User Name</th>
<th>CPU</th>
<th>Mem Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>alg.exe</td>
<td>2608</td>
<td>LOCAL SERVICE</td>
<td>00</td>
<td>1,600 K</td>
</tr>
<tr>
<td>TPAutoConnSvc.exe</td>
<td>2528</td>
<td>SYSTEM</td>
<td>00</td>
<td>2,188 K</td>
</tr>
<tr>
<td>btwdns.exe</td>
<td>2220</td>
<td>SYSTEM</td>
<td>00</td>
<td>1,364 K</td>
</tr>
<tr>
<td>wimprov.exe</td>
<td>2208</td>
<td>SYSTEM</td>
<td>00</td>
<td>6,700 K</td>
</tr>
<tr>
<td>cvpnd.exe</td>
<td>2020</td>
<td>SYSTEM</td>
<td>00</td>
<td>2,712 K</td>
</tr>
<tr>
<td>CodeMeter.exe</td>
<td>1972</td>
<td>SYSTEM</td>
<td>00</td>
<td>2,596 K</td>
</tr>
<tr>
<td>ADEngineW.exe</td>
<td>1912</td>
<td>SYSTEM</td>
<td>00</td>
<td>34,720 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>1892</td>
<td>LOCAL SERVICE</td>
<td>00</td>
<td>1,692 K</td>
</tr>
<tr>
<td>ADEngineS.exe</td>
<td>1864</td>
<td>SYSTEM</td>
<td>00</td>
<td>134,972 K</td>
</tr>
<tr>
<td>prtk_worker_serviz.exe</td>
<td>1856</td>
<td>SYSTEM</td>
<td>00</td>
<td>652 K</td>
</tr>
<tr>
<td>prtk_supervisor_siz.exe</td>
<td>1844</td>
<td>SYSTEM</td>
<td>00</td>
<td>652 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>1808</td>
<td>LOCAL SERVICE</td>
<td>00</td>
<td>1,784 K</td>
</tr>
<tr>
<td>spoolsv.exe</td>
<td>1732</td>
<td>SYSTEM</td>
<td>00</td>
<td>5,872 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>1528</td>
<td>LOCAL SERVICE</td>
<td>00</td>
<td>2,084 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>1504</td>
<td>NETWORK SERVICE</td>
<td>00</td>
<td>1,148 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>1452</td>
<td>SYSTEM</td>
<td>00</td>
<td>32,288 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>1416</td>
<td>NETWORK</td>
<td>00</td>
<td>1,912 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>1312</td>
<td>SYSTEM</td>
<td>00</td>
<td>2,416 K</td>
</tr>
<tr>
<td>vmaeip.exe</td>
<td>1292</td>
<td>SYSTEM</td>
<td>00</td>
<td>864 K</td>
</tr>
<tr>
<td>lsass.exe</td>
<td>1100</td>
<td>SYSTEM</td>
<td>00</td>
<td>1,600 K</td>
</tr>
<tr>
<td>services.exe</td>
<td>1088</td>
<td>SYSTEM</td>
<td>00</td>
<td>2,564 K</td>
</tr>
<tr>
<td>winlogon.exe</td>
<td>1044</td>
<td>SYSTEM</td>
<td>00</td>
<td>1,380 K</td>
</tr>
<tr>
<td>vmtosd.exe</td>
<td>952</td>
<td>SYSTEM</td>
<td>00</td>
<td>13,120 K</td>
</tr>
<tr>
<td>taskmgr.exe</td>
<td>948</td>
<td>Golden</td>
<td>00</td>
<td>4,820 K</td>
</tr>
<tr>
<td>wdfmgr.exe</td>
<td>940</td>
<td>LOCAL SERVICE</td>
<td>00</td>
<td>677 K</td>
</tr>
</tbody>
</table>

Processes: 60    CPU Usage: 0%    Commit Charge: 1630M / 500SM
FU on PID 2260

<table>
<thead>
<tr>
<th>Offset(P)</th>
<th>Name</th>
<th>PID</th>
<th>pslist</th>
<th>psscan</th>
<th>thrdproc</th>
<th>pspcid</th>
<th>csrss</th>
<th>session</th>
<th>deskthrd</th>
<th>ExitTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00319180</td>
<td>priv_kworker_ser</td>
<td>360</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>0x3960c58</td>
<td>vmtoolsd.exe</td>
<td>3932</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x004ed000</td>
<td>InstallShield L</td>
<td>1292</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x0aad8888</td>
<td>services.exe</td>
<td>1152</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09852990</td>
<td>wcescomm.exe</td>
<td>4052</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09eb52d0</td>
<td>wmpmvse.exe</td>
<td>2984</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f15810</td>
<td>wdfmgr.exe</td>
<td>2316</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f2560</td>
<td>sshd.exe</td>
<td>2880</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f3820</td>
<td>CodeMeterCC.exe</td>
<td>1972</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f4528</td>
<td>cmd.exe</td>
<td>2164</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f513</td>
<td>oracle.exe</td>
<td>2996</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f588</td>
<td>rundll32.exe</td>
<td>3640</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f5d0</td>
<td>wmschlp.exe</td>
<td>332</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f6b20</td>
<td>svchost.exe</td>
<td>1228</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>winlogon.exe</td>
<td>1756</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>googleupdate.exe</td>
<td>184</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>cfmon.exe</td>
<td>362</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>svchost.exe</td>
<td>168</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>cygrunsrv.exe</td>
<td>155</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>CodeMeter.exe</td>
<td>243</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>extjob.exe</td>
<td>297</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>vmtoolsd.exe</td>
<td>2476</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>svchost.exe</td>
<td>1744</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>svchost.exe</td>
<td>1632</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>svchost.exe</td>
<td>2240</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>svchost.exe</td>
<td>396</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>svchost.exe</td>
<td>564</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>MagicDisc.exe</td>
<td>2260</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>wuautil.exe</td>
<td>2776</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>AdobeAM.exe</td>
<td>3004</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>tsas3.exe</td>
<td>1164</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>UITroy.exe</td>
<td>2080</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>GoogleUpdate.exe</td>
<td>4828</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>juscched.exe</td>
<td>3880</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>alg.exe</td>
<td>3244</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>0x09f3f75d0</td>
<td>hosplms.exe</td>
<td>880</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>
## Clipboard Contents

```
$ python vol.py --profile=Win7SP1x64 -f win7.vmem clipboard

Volatility Foundation Volatility Framework 2.4

<table>
<thead>
<tr>
<th>Session</th>
<th>WindowStation</th>
<th>Format</th>
<th>...</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WinSta0</td>
<td>CF_UNICODETEXT</td>
<td>I didn't kill Josephine!!!</td>
<td></td>
</tr>
</tbody>
</table>
```

...
Windows Command History

Source: AMF
consoles

```
bigjoe:volatility golden$ python vol.py --profile=Win7SP1x64 -f ~/Documents/Virtual/Machines.localized/Windows\7\x64/Windows\7\x64-1e433ca5.vmem consoles
Volatility Foundation Volatility Framework 2.4
******************************************************************************
ConsoleProcess: conhost.exe Pid: 5480
Console: 0xffffffff CommandHistorySize: 50
HistoryBufferCount: 2 HistoryBufferMax: 4
OriginalTitle: %SystemRoot%\system32\cmd.exe
Title: Administrator: C:\Windows\system32\cmd.exe
AttachedProcess: cmd.exe Pid: 5472 Handle: 0x60

----
CommandHistory: 0x2fb00e0 Application: PsList.exe Flags:
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0

----
CommandHistory: 0x159cc0 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 3 LastAdded: 2 LastDisplayed: 2
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
Cmd #0 at 0x158f50: pslist
Cmd #1 at 0x158f90: cd\util
Cmd #2 at 0x158fb0: dir *fu*
----
```
## consoles

```
Screen 0x83cc0 X:80 Y:4000
Dump:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Golden>pslist

pslist v1.29 - Sysinternals PsList
Copyright (C) 2000-2009 Mark Russinovich
Sysinternals

Process information for ETOUFFEE-VM:

<table>
<thead>
<tr>
<th>Name</th>
<th>Pid</th>
<th>Pri</th>
<th>Thd</th>
<th>Hnd</th>
<th>Priv</th>
<th>CPU Time</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0:19:35.171</td>
<td>0:00:00.000</td>
</tr>
<tr>
<td>System</td>
<td>4</td>
<td>8</td>
<td>102</td>
<td>575</td>
<td>132</td>
<td>0:00:18.735</td>
<td>1:58:24.698</td>
</tr>
<tr>
<td>smss</td>
<td>216</td>
<td>11</td>
<td>2</td>
<td>29</td>
<td>380</td>
<td>0:00:00.078</td>
<td>1:58:24.698</td>
</tr>
<tr>
<td>csrss</td>
<td>296</td>
<td>13</td>
<td>9</td>
<td>912</td>
<td>1964</td>
<td>0:00:01.014</td>
<td>1:58:21.890</td>
</tr>
<tr>
<td>csrss</td>
<td>336</td>
<td>13</td>
<td>12</td>
<td>506</td>
<td>16664</td>
<td>0:00:00.904</td>
<td>1:58:21.656</td>
</tr>
<tr>
<td>wininit</td>
<td>344</td>
<td>13</td>
<td>3</td>
<td>74</td>
<td>1308</td>
<td>0:00:00.280</td>
<td>1:58:21.625</td>
</tr>
<tr>
<td>winlogon</td>
<td>376</td>
<td>13</td>
<td>5</td>
<td>129</td>
<td>2844</td>
<td>0:00:00.436</td>
<td>1:58:21.547</td>
</tr>
<tr>
<td>services</td>
<td>452</td>
<td>9</td>
<td>17</td>
<td>293</td>
<td>5868</td>
<td>0:00:01.341</td>
<td>1:58:21.094</td>
</tr>
<tr>
<td>lsass</td>
<td>460</td>
<td>9</td>
<td>11</td>
<td>669</td>
<td>4124</td>
<td>0:00:01.653</td>
<td>1:58:21.048</td>
</tr>
<tr>
<td>lsml</td>
<td>468</td>
<td>8</td>
<td>11</td>
<td>168</td>
<td>2464</td>
<td>0:00:00.031</td>
<td>1:58:21.048</td>
</tr>
<tr>
<td>svchost</td>
<td>556</td>
<td>8</td>
<td>11</td>
<td>375</td>
<td>4080</td>
<td>0:00:04.820</td>
<td>1:58:20.330</td>
</tr>
<tr>
<td>svchost</td>
<td>632</td>
<td>8</td>
<td>11</td>
<td>385</td>
<td>5668</td>
<td>0:00:01.887</td>
<td>1:58:20.174</td>
</tr>
<tr>
<td>MsMpEng</td>
<td>680</td>
<td>8</td>
<td>18</td>
<td>371</td>
<td>105612</td>
<td>0:07:54.695</td>
<td>1:58:20.127</td>
</tr>
<tr>
<td>svchost</td>
<td>820</td>
<td>8</td>
<td>33</td>
<td>606</td>
<td>19044</td>
<td>0:00:00.904</td>
<td>1:58:19.862</td>
</tr>
<tr>
<td>svchost</td>
<td>868</td>
<td>8</td>
<td>24</td>
<td>432</td>
<td>6912</td>
<td>0:00:00.468</td>
<td>1:58:19.784</td>
</tr>
<tr>
<td>svchost</td>
<td>896</td>
<td>8</td>
<td>31</td>
<td>545</td>
<td>10004</td>
<td>0:00:01.419</td>
<td>1:58:19.722</td>
</tr>
<tr>
<td>svchost</td>
<td>952</td>
<td>8</td>
<td>75</td>
<td>1628</td>
<td>58120</td>
<td>0:04:47.057</td>
<td>1:58:19.207</td>
</tr>
<tr>
<td>svchost</td>
<td>1100</td>
<td>8</td>
<td>29</td>
<td>785</td>
<td>40132</td>
<td>0:00:02.308</td>
<td>1:58:04.839</td>
</tr>
<tr>
<td>spoolsv</td>
<td>1224</td>
<td>8</td>
<td>27</td>
<td>517</td>
<td>14760</td>
<td>0:00:00.421</td>
<td>1:58:03.154</td>
</tr>
<tr>
<td>svchost</td>
<td>1252</td>
<td>8</td>
<td>31</td>
<td>363</td>
<td>9348</td>
<td>0:00:00.670</td>
<td>1:58:03.030</td>
</tr>
<tr>
<td>eEBSvc</td>
<td>1356</td>
<td>8</td>
<td>33</td>
<td>173</td>
<td>4360</td>
<td>0:00:00.062</td>
<td>1:58:02.530</td>
</tr>
<tr>
<td>armvsv</td>
<td>1540</td>
<td>8</td>
<td>5</td>
<td>69</td>
<td>1192</td>
<td>0:00:00.031</td>
<td>1:58:01.906</td>
</tr>
<tr>
<td>AppleMobileDeviceService</td>
<td>1576</td>
<td>8</td>
<td>11</td>
<td>136</td>
<td>3252</td>
<td>0:00:00.062</td>
<td>1:58:00.700</td>
</tr>
<tr>
<td>mDNSResponder</td>
<td>1608</td>
<td>8</td>
<td>6</td>
<td>118</td>
<td>1940</td>
<td>0:00:00.530</td>
<td>1:58:00.440</td>
</tr>
</tbody>
</table>
```
consoles

Screen 0x186fd0 X:80 Y:300
Dump:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>f:

F:\>dir
Volume in drive F is PATRIOT
Volume Serial Number is 52E1-6FB3

Directory of F:\
04/10/2014 00:26 PM  268,435,456 ext3-256MB.img
04/10/2014 00:26 PM  5,162 vizdiff.pl
04/01/2014 08:19 AM  <DIR> IMAGES
04/10/2014 01:26 PM  268,435,456 msdos-256MB.img
04/10/2014 01:26 PM  268,435,456 ntfs-256MB.img
04/10/2014 01:26 PM  268,435,456 zeros-256MB.img
08/24/2014 09:38 AM  893,413 Xubuntu1404.zip
04/17/2014 02:11 PM  997 otr.private_key
07/22/2014 07:16 PM  957,349,888 xubuntu-14.04-desktop-amd64.iso
09/06/2014 05:21 PM  2,678,614,016 Windows Vista RTM x86.iso
03/03/2013 05:34 AM  <DIR> winpmem-1.4
06/25/2014 08:46 AM  3,181,248,512 WIN7.ISO
10 File(s) 7,891,853,812 bytes
2 Dir(s) 90,739,441,664 bytes free

F:\>cd winpmem-1.4
F:\winpmem-1.4>dir
Volume in drive F is PATRIOT
Volume Serial Number is 52E1-6FB3

Directory of F:\winpmem-1.4
$ python vol.py -f carberp.mem --profile=WinXPSP3x86 malfind
Volatility Foundation Volatility Framework 2.4
[snip]

Process: svchost.exe Pid: 992 Address: 0x9d0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

```
0x009d0000 b8 35 00 00 00 e9 8b d1 f3 7b 68 6c 02 00 00 e9 .5........{hl.....
0x009d0010 94 63 f4 7b 8b ff 55 8b ec e9 6c 11 e4 7b 8b ff .c.{...U...1...{
0x009d0020 55 8b ec e9 99 2e 84 76 8b ff 55 8b ec e9 74 60 U......v..U...t`
0x009d0030 7f 76 8b ff 55 8b ec e9 8a e9 7f 76 8b ff 55 8b .v..U........v..U.

0x9d0000 b835000000 MOV EAX, 0x35
0x9d0005 e98bd1f37b JMP 0x7c90d195
0x9d000a 686c020000 PUSH DWORD 0x26c
0x9d000f e99463f47b JMP 0x7c9163a8
0x9d0014 8bff MOV EDI, EDI
0x9d0016 55 PUSH EBP
```

Legitimate instructions
And Solve Real Forensics Cases
Credit Card Statement

<table>
<thead>
<tr>
<th>Reference</th>
<th>Sold</th>
<th>Posted</th>
<th>Activity Since Last Statement</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>89XB773</td>
<td>1/9</td>
<td>1/10</td>
<td>Pizza Palace</td>
<td>$24.53</td>
</tr>
<tr>
<td>78XY667</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34XR889</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23XY001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76X0E11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Previous Balance | (+) | 189.43 | Current Amount Due | 1000.00
Purchases        | (+) | 820.57 | Amount Past Due    |        
Cash Advances     | (+) |        | Amount Over Credit Line |        
Payments          | (-) | 10.00  | Minimum Payment Due | 30.00  
Credits           | (-) |        |                  |        
Finance Charges   | (+) |        |                  |        
Late Charges      | (+) |        |                  |        
NEW BALANCE       | (-) | 1000.00|                  |        

FINANCE CHARGE SUMMARY

<table>
<thead>
<tr>
<th>Periodic Rate</th>
<th>Annual Percentage Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 %</td>
<td>36 %</td>
</tr>
<tr>
<td>36 %</td>
<td></td>
</tr>
</tbody>
</table>

For Customer Service Call: 1-800-555-5555
For Lost or Stolen Cards, Call: 1-888-555-5555
Hotkey sequence brings up bare authentication dialog: 
\(<CTRL><SHIFT><ALT>-T.\)

This sequence can be changed. Can’t access configuration screen without the password.

Process is spawned to handle this dialog.

Process is terminated when dialog is dismissed.
Computer powered off, but eBlaster is persistent.

Clone and virtualize system to investigate.

- dd image
- raw2vmdk
- VMWare virtual machine
- Live forensics analysis
DumpIt window
(dumps physical memory)
The password that you have entered is invalid!
467105092:VERYUNIQUE

1bd77544 [3772:0012f544] VERYUNIQUE

PID of process whose address space includes string

# python vol.py --profile=WinXPSP3x86 -f YOUR-US67PI6LUV-20121017-214253.raw strings -s strings.txt --output-file=stringslocation.txt -S

467105092:VERYUNIQUE

Search memory dump for invalid password VERYUNIQUE

Map location of invalid pw to owning process

Verify that it’s the process that gets spawned when dialog opens

Dump memory of that process and correlate location of incorrect and correct passwords
Activity Reports, Forwarding Services and Alerts are sent via email using the following settings:

**Report Delivery Summary**

Based on your current settings, eBlaster will send an Activity Report via email **every 60 minutes** to 'larry004@yahoo.com'.

eBlaster will automatically forward Emails (including attachments), Chat / Instant Messages and Keyword Alerts to 'larry004@yahoo.com'.

**Send via Email to:**

- larry004@yahoo.com

**Activity Reports**

- Activity Report Delivery: **On**
- Send a Report:
  - Every 60 Minutes of Activity
- Format Report as: **HTML**

**Email**

- Forward All Emails: **On**
- Include Attachments: **On**

**Chat / Instant Message**

- Forward All Chat/IMs: **On**

**Alerts**

- Forward Keyword Alerts: **On**
Memory Forensics: Some Gaps

- New OS Features
- Userspace Apps and Malware
- Performance, Reliability, Automation
- GPUs, Other Hardware
New OS Features

Windows 10
Virtualization Based Security

OpenBSD
KARL: A new kernel on every boot...

Compressed RAM

WSL

Linux (Ubuntu)
Some New OS Features vs. Memory Forensics
Compressed RAM
“Traditional” forensics: capture contents of storage devices, including the swap file.

Swap file contains RAM “overflow”—without this data, don't have a complete view of system memory.

Memory forensics: capture RAM.

Acquiring both RAM and disk contents results in memory *smearing*, where the swap file may be (very) out of sync with the memory dump.
Forensics Folks Hate Swap Files Anyway

Data trapped in unsanitized blocks allocated to swap file

void swap_crypt_ctx_initialize ( void ) {
    unsigned int i;
    if ( swap_crypt_ctx_initialized == FALSE ) {
        for (i = 0; i < (sizeof ( swap_crypt_key ) / sizeof ( swap_crypt_key [0])); i++) {
            swap_crypt_key [i] = random();
        }
        aes_encrypt_key ( (const unsigned char *) swap_crypt_key ,
            SWAP_CRYPT_AES_KEY_SIZE ,
            &swap_crypt_ctx.encrypt);
        ...
        swap_crypt_ctx_initialized = TRUE ;
    }
}

554-88-2345
kool@gmail.com
murder
struct { int ip;
... } netstat;
All Evidence Sources: Compressed RAM

Mac OS X + Linux + Win10 now set aside RAM and compress pages that would otherwise have been swapped due to memory pressure.
Compressed RAM: Visual

Physical memory dump:

Have:

Want:

Process P1

Process P2

Have

Want

Have

Want
Compressed RAM: The Scoop

put the rose in my hair like the Andalusian girls used or shall I wear a red yes and how he kissed me under the Moorish wall and I thought well as well him as another and then I asked him with my eyes to ask again yes and then he asked me would I yes to say yes my mountain flower
Benchmarks (Sorry, Python)

WKdm compression / decompression ops/sec

This has implications beyond compressed RAM analysis, of course.

All on 3.4GHz i7
iMac w/ 32GB RAM

391 per sec
There Can (Could!) Be Only One (OS)

- Linux
- Windows 10
- Mac OS X

Memory Forensics Tool

Actionable Evidence
Windows 10 Subsystem for Linux (WSL)

Useful, but a nightmare for malware detection and memory forensics!
There Can Be Only One

Required an extensive reverse engineering effort of WSL—handled largely by a grad student who took the RE course

New tools for interrogating "hosted" operating systems with memory forensics

Win10 Linux Subsystem Analysis

```
E:\>python volatility\vol.py -f samples\WSL.vmem --profile=Win10x64_15063 picobash
Volatility Foundation Volatility Framework 2.6

<table>
<thead>
<tr>
<th>Win PID</th>
<th>WSL PID</th>
<th>Command Time</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>5472</td>
<td>2</td>
<td>2018-01-05 22:40:00 UTC+0000</td>
<td>uname -a</td>
</tr>
<tr>
<td>5472</td>
<td>2</td>
<td>2018-01-05 22:40:02 UTC+0000</td>
<td>ls -ltr</td>
</tr>
<tr>
<td>5472</td>
<td>2</td>
<td>2018-01-05 22:40:11 UTC+0000</td>
<td>ifconfig</td>
</tr>
<tr>
<td>5472</td>
<td>2</td>
<td>2018-01-05 22:40:21 UTC+0000</td>
<td>echo $PATH</td>
</tr>
<tr>
<td>5472</td>
<td>2</td>
<td>2018-01-05 22:40:27 UTC+0000</td>
<td>which iperf</td>
</tr>
<tr>
<td>5472</td>
<td>2</td>
<td>2018-01-05 22:40:32 UTC+0000</td>
<td>top</td>
</tr>
<tr>
<td>716</td>
<td>17</td>
<td>2018-01-05 22:40:56 UTC+0000</td>
<td>iperf -s</td>
</tr>
<tr>
<td>3160</td>
<td>27</td>
<td>2018-01-05 22:41:01 UTC+0000</td>
<td>iperf -c 127.0.0.1</td>
</tr>
<tr>
<td>3160</td>
<td>27</td>
<td>2018-01-05 22:41:27 UTC+0000</td>
<td>sudo -H python get-pip.py</td>
</tr>
</tbody>
</table>
```
Win10 Linux Subsystem Analysis

E:\python vol.py -f WSL.mem --profile=Win10x64_15063 handles -p 3232
Volatility Foundation Volatility Framework 2.6
Offset(V)  Pid  Handle  Access Type  Details
-------------------  ------  -----------------  ------------------  ------

[no output]

E:\python vol.py -f WSL.mem --profile=Win10x64_15063 picolfsof -p 3232
Volatility Foundation Volatility Framework 2.6
Pid: 3232 Name: /tmp/bin
  0 -> /dev/tty1
  1 -> /dev/tty1
  2 -> /dev/tty1
  3 -> /tmp/logfile.txt

Busting a keystroke logger running inside of WSL
Userland Malware Detection
Crisis: Notorious Mac OS X Malware

• Kernel and user-level components
• Patches Activity Monitor to hide
• Code injection and Objective C pointer swizzling
• Takes screenshots, captures audio, video
• Snoops web browsing
• Intercepts and logs messaging
  — ...even if you use encryption protocols like OTR (Adium)
• Basically, all the stuff you're scared malware will do to you

• Source: https://github.com/hackedteam/core-macos & https://github.com/hackedteam/driver-macos
Pointer Swizzling Detection on Mac OS X

- Better kernel-level defenses = renewed interest in userland malware
- Focus on Objective-C malware (e.g., Crisis)
- Complex runtime provides lots of opportunities
  - Add methods
  - Switch implementations
  - ...

```swift
swizzleByAddingIMP(@selector(webFrameLoadCommitted:), @selector(webFrameLoadCommittedHook:));
...
swizzleByAddingIMP(@selector(closeCurrentTab:), @selector(closeCurrentTabHook:));
...
swizzleByAddingIMP(@selector(setTitle:), @selector(setTitleHook:));
...
Swizzling Detection

Warning signs:

- Implementation in different library than most other methods
- Address in non-file-backed memory region
Automatically Analyzing Hooks in RAM

Physical Memory Dump

HookTracer

Filesystem
Registry
Network
Syscall Emulation

unicorn emulator

IOCsm

Behavioral Profiles
- Keystroke data is logged to a file
- Keystroke data is sent over the network
- Keystroke data is written to the registry
- Contents of the clipboard are accessed
- Screenshots are generated
- Debugging APIs are used (code injection)
- The hook immediately calls CallNextHookEx

...to appear in DFRWS 2019.
Reliability and Automation
Gaslight: Fuzzing Memory Forensics Frameworks

Gaslight v2

HPX

Fuzzing Harness

Custom FS

Custom FS

Custom FS

Custom FS

MFF

MFF

MFF

MFF

mutated memory images

Mutation Set

Memory Image
## Volatility: Crashes

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Programming Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux_library_list</td>
<td>List enumeration code didn’t properly validate pointer to data structures before processing it</td>
</tr>
<tr>
<td>linux_dmesg</td>
<td>Did not validate that log structures referenced a valid page before attempting to process them</td>
</tr>
<tr>
<td>linux_arp</td>
<td>Integer overflow in bit shifting operation</td>
</tr>
<tr>
<td>mac_check_syscall</td>
<td>Crashed when system call table entries were not on a mapped page</td>
</tr>
</tbody>
</table>
## Volatility: Infinite Loops

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Programming Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux_bash</td>
<td>List enumeration code infinite looped when a mutation caused two list members to point to each other</td>
</tr>
<tr>
<td>linux_arp</td>
<td>List enumeration code infinite looped when a mutated list entry pointed to a previous entry</td>
</tr>
<tr>
<td>mac_lsmop</td>
<td>List enumeration code infinite looped when a mutated list entry pointed to a previous entry</td>
</tr>
<tr>
<td>mac_lsof</td>
<td>Nearly infinite loop when the variable specifying how many handles a process had opened was mutated to ~3 billion</td>
</tr>
</tbody>
</table>
# Volatility: Unusable Output

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Programming Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux_psaux</td>
<td>The <em>mm_struct</em> members that specify the start and end of the command line arguments were mutated to specify a size in the gigabytes.</td>
</tr>
<tr>
<td>linux_psenv</td>
<td>Same as issue as <em>linux_psaux</em>, but for the members that specify the size of the process' environment variables.</td>
</tr>
<tr>
<td>mac_dyld_maps</td>
<td>List enumeration code infinite looped and the rendering code did not validate structure properly before reporting (printing).</td>
</tr>
<tr>
<td>mac_psaux</td>
<td>Same base issue as <em>linux_psaux</em></td>
</tr>
</tbody>
</table>
# Volatility: Resource Exhaustion

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Programming Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux_procdump</td>
<td>Gaslight forced the plugin to attempt to create a 20GB+ file as the function used for extraction to disk, <code>write_elf_file</code>, did not properly validate file size metadata</td>
</tr>
<tr>
<td>linux_librarydump</td>
<td>This plugin relied on the same <code>write_elf_file</code> function when extracting shared libraries to disk</td>
</tr>
</tbody>
</table>
Lots to Do (Want Some?)

• Automation
• User-level application memory forensics
• Machine learning for malware discovery and reducing tedium for investigators
• Support for other operating systems
  – e.g., BSD, other Unix flavors
  – One of my students created preliminary support for Volatility, but has "moved on" after M.S.
• Windows 10 hypervisor issues
  – Very preliminary work: "Live forensics on the Windows 10 secure kernel" by Hans Kristian Brendmo
    https://brage.bibsys.no/xmlui/bitstream/handle/11250/2448948/18109_FULLTEXT.pdf?sequence=1
• Better performance
• GPU malware and malware infecting other subsystems
• Lots more
Some Papers if You're Interested

golden@cct.lsu.edu

cct.lsu.edu/~golden

Talk: https://cct.lsu.edu/~golden/Materials/minding-the-gaps.pdf