Toward Reproducible Malware Forensics

Brendan Dolan-Gavitt,
Columbia University
ACSAC MMF 2014
Reproducibility

• Basic ingredients:

  1. Describe your methods well

  2. (Optional, but highly recommended) Release your code

  3. Release your data

• (1), (2) examined in previous work (Collberg et al., 2014)

• But in the context of malware analysis, what does (3) mean?
Reproducibility Problems

- Software execution is ephemeral
  - Environment may change
  - Timings may change
  - Library versions, time of day, etc.
- Thus, *dynamic* analyses are hard to reproduce
Reproducible Malware Analyses

• Malware has short “shelf life”

• C&C servers are quickly taken down

• Lack of access to reliable data sets makes research harder

• Barriers to entry: need your own malware feed

• Can’t tell if previous results are correct
Previous Efforts

- Generally *artifact*-based or assume static analysis
- Malware sample repositories
  - VXShare, OpenMalware (née Offensive Computing), Contagio, …
- Malware artifact repositories
  - DHS Predict (PCAPs), Malwr (behavioral reports)
Idea: Shareable Record/Replay

- Full execution traces would solve this, but are enormous (GB/s)
- Instead, use record/replay: classic (30+ years old) technique for recording program executions
- Lots of academic literature on it: ReVirt, TTVM
- Main idea: record the non-deterministic inputs
- Until recently, no open source whole-system implementations
Record/Replay

CPU

Get Current Date

Outside World

Fri May 23 11:33:27

==
==
==
?= 0x45?
>= 0x80?
Record/Replay

CPU

== Friday?

== 0x45?

>= 0x80?

Outside World

Fri May 23 11:33:27
Record/Replay

CPU

== Friday?

== 0x45?

>= 0x80?

Get Current Date

Outside World

Fri May 23 11:33:27

Recv Packet

0x0000: 4500 002c 0000 4000
0x0008: 4006 6b48 127e 0021
0x0010: 5dae 5f37 01bb bed4
0x0018: fccd 820f d690 0847
0x0020: 6012 3908 cfa2 0000
0x0028: 0204 05b4
Record/Replay

CPU

== Friday?

== 0x45?

>= 0x80?

Outside World

Fri May 23 11:33:27

Recv Packet

0x0000: 4500 002c 0000 4000
0x0008: 4006 6b48 127e 0021
0x0010: 5dae 5f37 01bb bed4
0x0018: fccd 820f d690 0847
0x0020: 6012 3908 cfa2 0000
0x0028: 0204 05b4
Record/Replay

CPU

Get Current Date

>= 0x80?

Recev Packet

Outside World

Fri May 23 11:33:27

0x0000:  4500 002c 0000 4000
0x0008:  4006 6b48 127e 0021
0x0010:  5dae 5f37 01bb bed4
0x0018:  fccd 820f d690 0847
0x0020:  6012 3908 cfa2 0000
0x0028:  0204 05b4

== Friday?
Record/Replay

CPU

== Friday?

== 0x45?

>= 0x80?

Get Current Date

Recv Packet

Outside World

Fri May 23 11:33:27

Record Log

0x0000: 4500 002c 0000 4000
0x0008: 4006 6b48 127e 0021
0x0010: 5dae 5f37 01bb bed4
0x0018: fccd 820f d690 0847
0x0020: 6012 3908 cfa2 0000
0x0028: 0204 05b4
Record / Replay

Time

rdtsc  interrupt  DMA
PANDA

- Based on QEMU 1.0.1
- **Deterministic record/replay**
- Translation to LLVM for all QEMU architectures (extended from S2E code)
- Android (ARM) emulation support
- Plugin architecture – easy to extend to new analyses
## Log Size

<table>
<thead>
<tr>
<th>Replay</th>
<th>Instructions</th>
<th>Log Size</th>
<th>Instr/Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>freebsdboot</td>
<td>9.3 billion</td>
<td>533 MB</td>
<td>17</td>
</tr>
<tr>
<td>spotify</td>
<td>12 billion</td>
<td>229 MB</td>
<td>52</td>
</tr>
<tr>
<td>haikuurl</td>
<td>8.6 billion</td>
<td>119 MB</td>
<td>72</td>
</tr>
<tr>
<td>carberp1</td>
<td>9.1 billion</td>
<td>43 MB</td>
<td>212</td>
</tr>
<tr>
<td>win7iessl</td>
<td>8.6 billion</td>
<td>9.4 MB</td>
<td>915</td>
</tr>
<tr>
<td>Starcraft</td>
<td>60 million</td>
<td>1.8 MB</td>
<td>33</td>
</tr>
</tbody>
</table>
This site stores recordings made with the PANDA dynamic analysis platform. To find out more about PANDA's record/replay features, you can peruse the documentation. After downloading, the .rr files can be extracted using `scripts/rrunpack.py` in the PANDA distribution.

### Upload a new record/replay log

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
<th>Download</th>
<th>Size</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>cve-2012-4792-exploit</td>
<td>Exploitation of cve-2012-4792</td>
<td>rrlogs/cve-2012-4792-exploit.rr</td>
<td>130.1 MB</td>
<td>968.8 million</td>
</tr>
<tr>
<td>cve-2012-4792-crash</td>
<td>Crashing instance of cve-2012-4792</td>
<td>rrlogs/cve-2012-4792-crash.rr</td>
<td>129.9 MB</td>
<td>608.8 million</td>
</tr>
<tr>
<td>cve-2011-1255-exploit</td>
<td>Exploitation of cve-2011-1255</td>
<td>rrlogs/cve-2011-1255-exploit.rr</td>
<td>126.6 MB</td>
<td>2.1 billion</td>
</tr>
<tr>
<td>cve-2014-1776-crash</td>
<td>Crashing instance of cve-2014-1776</td>
<td>rrlogs/cve-2014-1776-crash.rr</td>
<td>155.9 MB</td>
<td>1.2 billion</td>
</tr>
<tr>
<td>dia2dump</td>
<td>Parsing a PDB with dia2dump</td>
<td>rrlogs/dia2dump.rr</td>
<td>190.8 MB</td>
<td>5.4 billion</td>
</tr>
<tr>
<td>line2</td>
<td>Sending an IM using LINE for Android</td>
<td>rrlogs/line2.rr</td>
<td>64.6 MB</td>
<td>10.4 million</td>
</tr>
<tr>
<td>win7_64bit_install_STOP_D1</td>
<td>Failure during boot to install CD of Win7 64bit. DRIVER_IRQL_NOT_LESS_OR_EQUAL</td>
<td>rrlogs/win7_64_install_fail.rr</td>
<td>203.3 MB</td>
<td>5.3 billion</td>
</tr>
<tr>
<td>carberp2</td>
<td>Running custom RU_Az build of the Carberp malware</td>
<td>rrlogs/carberp2.rr</td>
<td>91.9 MB</td>
<td>2.9 billion</td>
</tr>
<tr>
<td></td>
<td>Running custom Full build of the Carberp</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Replay Subsumes Other Artifacts

Replay

PCAPs
Reports
Memory Dumps
Screenshots
Movies
MalRec: A Malware Recording Platform

- Based on PANDA dynamic analysis platform

- Simple agentless setup:
  - Malware loaded via CD image
  - Started by sending keystrokes to VM
  - No in-guest monitoring utilities (reports can be generated from replays)
Malware Pipeline

Ingest Malware Feed → Malware Recorders → Log Compressors
Implementation Details

• Samples fetched once per day at 22:30 UTC, random subset of 100 chosen

• `inotifywait` monitors incoming directory and passes off samples to GNU parallel & PANDA

• PANDA runs using `-record_from` and base Win7 32- or 64-bit base QCOW2

• Resulting logs are compressed with xz through another inotify/parallel queue
Limitations

• Analysis time is fixed & no interaction is done
  • In particular, only one path through malware

• PANDA is based on QEMU 1.0.1 & non-virtualized – very detectable

• Lock-in: replay logs can currently only be processed by PANDA, so initial analyses must be done in PANDA as well
Future Work

• Scaling up
  • Currently, limited by disk space (~20GB/day)
  • We get ~2000 samples/day, only record 100

• Add automated reports
  • Currently some basic support in PANDA, e.g. http://laredo-13.mit.edu/~brendan/opcleaver/reports/0c3e4035-c9ab-47bc-b245-35c80ceafe5e_proclog.txt

• Movies of executions
Future Work

• Malware “mind reading”

• Record all memory reads/writes and look for printable strings

• Save all printable strings

• Index and use information retrieval (e.g., Lucene or Terrier) to build search engine for malware memory accesses
Obtaining the Recordings

- [http://panda.gtisc.gatech.edu/malrec/](http://panda.gtisc.gatech.edu/malrec/)

- If you want easier bulk access, contact me at brendan@cs.columbia.edu and we can arrange something like an rsync transfer.
Questions
Discussion