Understanding Linux Malware

Emanuele Cozzi\textsuperscript{1}, Mariano Graziano\textsuperscript{2}, Yanick Fratantonio\textsuperscript{1}, Davide Balzarotti\textsuperscript{1}

\textsuperscript{1}EURECOM

\textsuperscript{2}Cisco Systems, Inc.

IEEE Symposium on Security & Privacy, May 2018
Malware and operating systems
Malware and operating systems
Linux malware on the rise

Hackers Used New Weapons to Disrupt Major Websites Across U.S.

By Nicole Perlroth

Oct. 21, 2016
Linux malware on the rise

The New York Times

Erebus

Web host agrees to pay $1m after it’s hit by Linux-targeting ransomware

Windfall payment by poorly secured host is likely to inspire new ransomware attacks.

DAN GOODIN - 6/20/2017, 12:52 AM
Linux malware: Leak exposes CIA's OutlawCountry hacking toolkit

OutlawCountry malware sends traffic from Linux machines to the CIA's servers.

By Liam Tung | July 4, 2017 -- 11:50 GMT (2:50 BST) | Topic: Security

BIZ & IT —

Web host agrees to pay $1m after it’s hit by Linux-targeting ransomware

Windfall payment by poorly secured host is likely to inspire new ransomware attacks.

DAN GOODIN - 6/20/2017, 12:52 AM
Linux malware: Leak exposes CIA's OutlawCountry malware sends

By Liam Tung | July 4, 2017 -- 11:50 GMT (19:50 PDT)

OutlawCountry malware sends

Web host by Linux-the-aki-windfall payment by

DAN GOODIN - 6/20/2017, 12:52
Objectives

- Develop a dynamic analysis sandbox for Linux binaries (and IoT devices)
Objectives

- Develop a dynamic analysis sandbox for Linux binaries (and IoT devices)
  - Previous studies only looked at the network behavior \(^1\) \(^2\)

---


\(^2\) Yin Minn Pa et al. "IoTPOT: analysing the rise of IoT compromises," USENIX Workshop on Offensive Technologies 2015.
Objectives

- Develop a dynamic analysis sandbox for Linux binaries (and IoT devices)
  - Previous studies only looked at the network behavior

- Identify challenges and limitations of porting traditional techniques to the new environment

---

2 Yin Minn Pa et al. "IoTPOT: analysing the rise of IoT compromises," USENIX Workshop on Offensive Technologies 2015.
Objectives

• Develop a dynamic analysis sandbox for Linux binaries (and IoT devices)
  ▶ Previous studies only looked at the network behavior

• Identify challenges and limitations of porting traditional techniques to the new environment

• Understand differences in the malware characteristics (packing, obfuscation, VM detection, privilege escalation, persistence...) wrt Windows malware

---

2 Yin Minn Pa et al. "IoTPOT: analysing the rise of IoT compromises," USENIX Workshop on Offensive Technologies 2015.
Diversity

CPU: Intel
Diversity

CPU: Intel, ARM, MIPS, Motorola, Sparc
Diversity

CPU: Intel, ARM, MIPS, Motorola, Sparc

OS: Linux
Diversity

CPU: Intel, ARM, MIPS, Motorola, Sparc

OS: Linux, BSD, Android
Diversity

CPU: Intel, ARM, MIPS, Motorola, Sparc

OS: Linux, BSD, Android

Libraries: glibc
Diversity

CPU: Intel, ARM, MIPS, Motorola, Sparc

OS: Linux, BSD, Android

Libraries: glibc, uclibc, libpcap, libopencl
Diversity

OS: Linux, BSD, Android

Libraries: glibc, uclibc, libpcap, libopencl

Statically-linked ELF unportable
Diversity

Statically-linked ELF unportable

Unknown device
Analysis infrastructure

Data collection

File & metadata analysis
- AVClass
- ELF anomaly
- File recognition

Static analysis
- Code analysis
- Packing identification

Dynamic analysis
- Packer analysis
- Sandbox preparation
- Emulation
- Trace analysis
Analysis infrastructure

Data collection

File & metadata analysis
- File recognition
- ELF anomaly
- AVClass

Static analysis
- Code analysis
- Packing identification

Dynamic analysis
- Packer analysis
- Sandbox preparation
- Emulation
- Trace analysis

Sandbox preparation

Trace analysis
Data collection

From November 2016 to November 2017

200 candidate samples per day

Dataset of 10,548 Linux malware
## Dataset

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Samples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>X86-64</td>
<td>3018</td>
<td>28.61%</td>
</tr>
<tr>
<td>MIPS I</td>
<td>2120</td>
<td>20.10%</td>
</tr>
<tr>
<td>PowerPC</td>
<td>1569</td>
<td>14.87%</td>
</tr>
<tr>
<td>Motorola 68000</td>
<td>1216</td>
<td>11.53%</td>
</tr>
<tr>
<td>Sparc</td>
<td>1170</td>
<td>11.09%</td>
</tr>
<tr>
<td>Intel 80386</td>
<td>720</td>
<td>6.83%</td>
</tr>
<tr>
<td>ARM 32-bit</td>
<td>555</td>
<td>5.26%</td>
</tr>
<tr>
<td>Hitachi SH</td>
<td>130</td>
<td>1.23%</td>
</tr>
<tr>
<td>AArch64 (ARM 64-bit)</td>
<td>47</td>
<td>0.45%</td>
</tr>
<tr>
<td>others</td>
<td>3</td>
<td>0.03%</td>
</tr>
</tbody>
</table>

Distribution of the 10,548 downloaded samples across architectures.
### Dataset

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Samples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>X86-64</td>
<td>3018</td>
<td>28.61%</td>
</tr>
<tr>
<td>MIPS I</td>
<td>2120</td>
<td>20.10%</td>
</tr>
<tr>
<td>PowerPC</td>
<td>1569</td>
<td>14.87%</td>
</tr>
<tr>
<td>Motorola 68000</td>
<td>1216</td>
<td>11.53%</td>
</tr>
<tr>
<td>Sparc</td>
<td>1170</td>
<td>11.09%</td>
</tr>
<tr>
<td>Intel 80386</td>
<td>720</td>
<td>6.83%</td>
</tr>
<tr>
<td>ARM 32-bit</td>
<td>555</td>
<td>5.26%</td>
</tr>
<tr>
<td>Hitachi SH</td>
<td>130</td>
<td>1.23%</td>
</tr>
<tr>
<td>AArch64 (ARM 64-bit)</td>
<td>47</td>
<td>0.45%</td>
</tr>
<tr>
<td>others</td>
<td>3</td>
<td>0.03%</td>
</tr>
</tbody>
</table>

Distribution of the 10,548 downloaded samples across architectures
## Dataset

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Samples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>X86-64</td>
<td>3018</td>
<td>28.61%</td>
</tr>
<tr>
<td>MIPS I</td>
<td>2120</td>
<td>20.10%</td>
</tr>
<tr>
<td>PowerPC</td>
<td>1569</td>
<td>14.87%</td>
</tr>
<tr>
<td>Motorola 68000</td>
<td>1216</td>
<td>11.53%</td>
</tr>
<tr>
<td>Sparc</td>
<td>1170</td>
<td>11.09%</td>
</tr>
<tr>
<td>Intel 80386</td>
<td>720</td>
<td>6.83%</td>
</tr>
<tr>
<td>ARM 32-bit</td>
<td>555</td>
<td>5.26%</td>
</tr>
<tr>
<td>Hitachi SH</td>
<td>130</td>
<td>1.23%</td>
</tr>
<tr>
<td>AArch64 (ARM 64-bit)</td>
<td>47</td>
<td>0.45%</td>
</tr>
<tr>
<td>others</td>
<td>3</td>
<td>0.03%</td>
</tr>
</tbody>
</table>

Distribution of the 10,548 downloaded samples across architectures
ELF manipulation

- ELF header
- Program header table
- .text
- ... (omitted)
- .data
- Section header table
ELF manipulation

- Anomalous ELF
  - Sections table removed

Diagram:

- ELF header
- Program header table
- .text
- ...
- .data
- Section header table
ELF manipulation

- Anomalous ELF
  - Sections table removed

- Invalid ELF
  - Segments table points beyond file
  - Overlapping header/segment
  - Sections table points beyond file
ELF manipulation

- Anomalous ELF
  - Sections table removed
- Invalid ELF
  - Segments table points beyond file
  - Overlapping header/segment
  - Sections table points beyond file
- Problems with common analysis tools
  - readelf 2.26.1
  - GDB 7.11.1
  - pyelftools 0.24
  - IDA Pro 7
<table>
<thead>
<tr>
<th>AVClass</th>
<th>Pymadro</th>
<th>Miner</th>
<th>Ebolachan</th>
<th>Golad</th>
<th>Lady</th>
<th>Connectback</th>
<th>Mirai</th>
</tr>
</thead>
</table>
Static analysis

Data collection

File & metadata analysis
- File recognition
- ELF anomaly
- AVClass

Static analysis
- Code analysis
- Packing identification

Dynamic analysis
- Packer analysis
- Sandbox preparation
- Emulation
- Trace analysis

Sandbox preparation

Emulation

Trace analysis
Vanilla UPX and custom variants are the prevalent packers (almost 4% of the dataset)
Vanilla UPX and custom variants are the prevalent packers (almost 4% of the dataset)
Packing

- Vanilla UPX and custom variants are the prevalent packers (almost 4% of the dataset)
- Vanilla UPX and custom variants are the prevalent packers (almost 4% of the dataset)
  - modified magic bytes
  - modified strings
  - junk bytes
Vanilla UPX and custom variants are the prevalent packers (almost 4% of the dataset)
  - modified magic bytes
  - modified strings
  - junk bytes

At least one malware family is using a custom packer
Dynamic analysis

Data collection

File & metadata analysis
- AVClass
- File recognition
- ELF anomaly

Static analysis
- Code analysis
- Packing identification

Dynamic analysis
- Packer analysis
- Sandbox preparation
- Emulation
- Trace analysis
Behaviors

- Process interaction
- Deception
- Anti-debugging
- Anti-execution
- Persistence
- Shell commands
- Sandbox detection
- Information gathering
- Processes enumeration
- Privileges escalation
- Process injection
- Required privileges
Malicious processes assume new names to trick process listing tools

52% of the samples renamed the process
• Malicious processes assume new names to trick process listing tools
• 52% of the samples renamed the process
Malicious processes assume new names to trick process listing tools

52% of the samples renamed the process
Malicious processes assume new names to trick process listing tools
52% of the samples renamed the process
Evasion

• Detect VMware, VirtualBox, QEMU, KVM but also OpenVZ, XEN or chroot jails

```
if (!sandbox) {
    // do evil
} else {
    print("https://lmgtfy.com/q=how+to+@@@@@@@@@@@@@@@@@@@")
Enter:
```

```
rm -r /
```
Evasion

• Detect VMware, VirtualBox, QEMU, KVM but also OpenVZ, XEN or chroot jails
• Malware may also check their file name before real execution
Evasion

• Detect VMware, VirtualBox, QEMU, KVM but also OpenVZ, XEN or chroot jails
• Malware may also check their file name before real execution

```
if (!sandbox) {
    // do evil
}
else {
    print("https://lmgtfy.com/q=how+to+@@@@@@@@@@@@@@@@@@")
    rm -rf /
}
```
• OS/ABI field in ELF header is not used
• Malware executed by root or user
• Processes enumeration
• Unstripped symbols (?)
- OS/ABI field in ELF header is not used
- OS/ABI field in ELF header is not used
- Malware executed by root or user
- OS/ABI field in ELF header is not used
- Malware executed by root or user
- Processes enumeration
• OS/ABI field in ELF header is not used
• Malware executed by root or user
• Processes enumeration
• Unstripped symbols (?)
Conclusions

- Linux malware still in its infancy
- Already a broad range of behaviors and tricks
- ELF binaries *could* run anywhere from a thermostat to a large server
- New research needed to overcome the lack of information about the execution environment
Thank you

https://padawan.s3.eurecom.fr/

Emanuele Cozzi
cozzi@eurecom.fr
@invano