Technische Hochschule Augsburg

Analyse von Linux Malware

Advanced Security Testing 25

Rajbir Singh · Niklas Sirch – 20.04.2025

Agenda

1. Motivation
2. Überblick
3. Statische Analyse
4. Dynamische Analyse
5. Malware Erkennung
6. Fazit

Motivation

The Hacker News

Chinese Hackers Target Linux Systems Using SNOWLIGHT Malware and VShell Tool

The China-linked threat actor known as UNC5174 has been attributed to a new campaign that leverages a variant of a known malware dubbed SNOWLIGHT and a new...

vor 1 Tag

GBHackers News

Chinese Hackers Unleash New BRICKSTORM Malware to Target Windows and Linux Systems

A sophisticated cyber espionage campaign leveraging the newly identified BRICKSTORM malware variants has targeted European strategic.

vor 1 Tag

CybersecurityNews

HelloKitty Ransomware Resurafced Targeting Windows, Linux, & ESXi Environments

Revival of the HelloKitty ransomware, with new variants actively targeting Windows, Linux, and ESXi environments simultaneously.

vor 3 Tagen

Trend Micro

BPFDoor's Hidden Controller Used Against Asia, Middle East Targets

A controller linked to BPF backdoor can open a reverse shell, enabling deeper infiltration into compromised networks. Recent attacks have been observed...

vor 3 Tagen

Security Boulevard

ELFDICOM: PoC Malware Polyglot Exploiting Linux-Based Medical Devices

A high severity vulnerability in DICOM, the healthcare industry's standard file protocol for medical imaging, has remained exploitable years after its...

vor 5 Tagen

Security Affairs

New malware 'ResolverRAT' targets healthcare, pharmaceutical firms

New malware 'ResolverRAT' is targeting healthcare and pharmaceutical firms, using advanced capabilities to steal sensitive data.

vor 2 Tagen

The Hacker News

New BPFDoor Controller Enables Stealthy Lateral Movement in Linux Server Attacks

BPFDoor malware's new controller enables firewall-bypassing shell access + lateral movement in 2024 atta

vor 1 Tag

The Hacker News

Pakistan-Linked Hackers Expand Targets in India with CurlBack RAT and Spark RAT

SideCopy hackers adopt MSI staging and launch CurlBack RAT attacks on Indian ministries, oil, and rail sectors.

vor 3 Tagen

SC Media

Chinese hackers set sights on Linux systems, Ivanti appliances

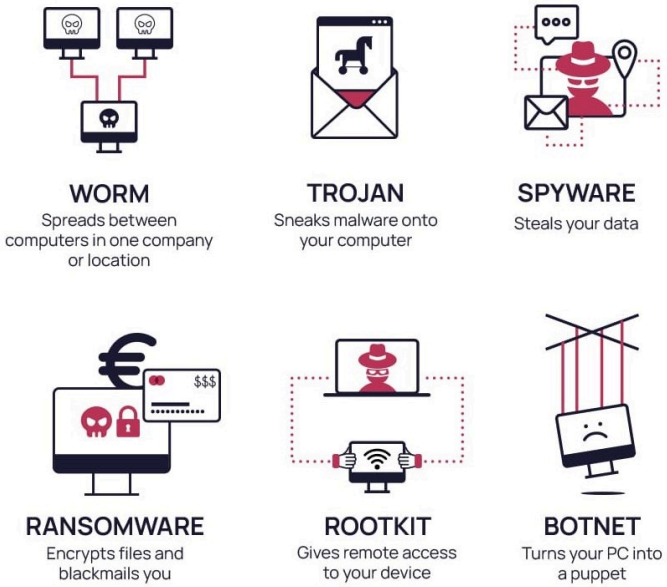
Vulnerable Linux and Ivanti Connect Secure VPN devices have been targeted in separate Chinese malware attack campaigns, reports The Hacker News.

vor 22 Stunden

Überblick

Typen von Linux-Malware

Überblick



https://sosafe-awareness.com/glossary/malware/

Fallbeispiele: Prometei, Shikitega

Überblick

- Prometei
  - Modular aufgebautes Botnetz mit Fokus auf Kryptomining
  - Verbreitung über schwache Zugangsdaten und bekannte Schwachstellen
  - Persistenz, laterale Bewegung, Root-Zugriff
- Shikitega
  - Nutzt Schwachstellen für Privilege Escalation
  - Polymorph, nutzt legitime Cloud-Dienste für Command-and-Control
  - Führt Metasploit-Meterpreter aus
- Mirai
  - IoT-Botnet
  - DDoS-Angriffe
  - Verbreitung über SSH oder Telnet

Statische Analyse

Tools

Statische Analyse

Einfache Tools

- strings – liest lesbare Zeichen aus Binärdateien
- file – erkennt Dateityp und Architektur
- binwalk – extrahiert eingebettete Dateien und Header

Reverse Engineering Tools

- objdump – disassembliert Maschinencode
- Rizin / Cutter – Reverse Engineering mit GUI
- Ghidra – High-Level Analyse, Quellcode-Rekonstruktion

Verschleierungs- & Verschlüsselungstechniken

Statische Analyse

- Packing – Code ist gepackt, z. B. via UPX
- Polymorphismus – Code verändert sich bei jeder Infektion
- Ziel: Erkennung durch Virenscanner erschweren
- Herausforderung: Erfordert Unpacking oder Laufzeit-Analyse

Beispiel 1: Statische Analyse von Prometei

Statische Analyse

- 428 KB große statisch gelinkte ELF-Datei
- Tools: `file`, `strings`, `binwalk`, `Cutter`
- Hinweise auf:
  - `systemctl enable` → Persistenz
  - UPX-Komprimierung erkannt

Prometei: Entpacken im Debugger

Statische Analyse

- UPX verhindert reguläres Entpacken
- Vorgehen:
  - Ausführung in Cutter + Breakpoints auf `mprotect`
  - Extraktion des entpackten Codes via `gdb`:

    ```
    dump memory prometei-unpacked.bin <start> <end>
    ```

  - Disassemblierung mit `objdump`

Prometei: Erkenntnisse

Statische Analyse

- Malware entpackt sich schrittweise zur Laufzeit
- Nur zweiter entpackter Speicherbereich enthält aktiven Code
- Entpackter Code = ca. 20.000 Zeilen Assembler
- Analyse zu aufwendig für vollständige manuelle Auswertung

Beispiel 2: Shikitega

Statische Analyse

- Sehr kleine Datei: 4.0 KB, ELF, keine Strings
- `binwalk` : keine Auffälligkeiten
- Keine sichtbaren `syscalls` in Cutter

Shikitega: Polymorph verschlüsselt

Statische Analyse

- Nutzt Encoder „Shikata Ga Nai“
  - Polymorph, schwer entpackbar
  - Teil von Metasploit
- Ergebnis:
  - Statische Analyse kaum möglich
  - Dynamische Analyse notwendig

Dynamische Analyse

Sichere Testumgebung

Dynamische Analyse

- Sandboxing z.B. Any.run oder Cuckoo
- Virtuelle Maschinen (VMs)
- Einschränkung des Zugriffs auf Netzwerk

Sichere Testumgebung

Sichere Testumgebung

Dynamische Analyse

- Sandboxing z.B. Any.run oder Cuckoo
- Virtuelle Maschinen (VMs)
- Einschränkung des Zugriffs auf Netzwerk

Tools für die dynamische Analyse

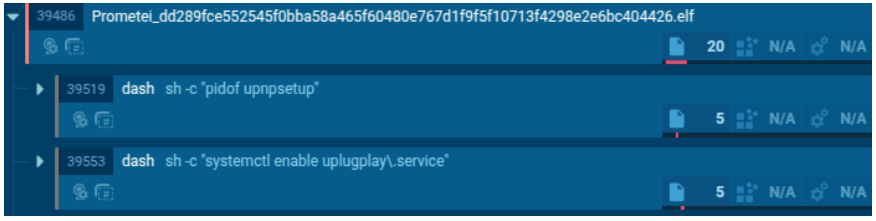
Dynamische Analyse

- GDB: Breakpoints, Variablen, Codefluss
- Strace: Verfolgt Systemcalls und Signale
- Wireshark: Netzwerkaktivität überwachen

Beispiel: Prometei Malware

Dynamische Analyse

- Analyse zuerst in Any.run



- Persistenz durch systemd-Service
- HTTP-Verbindung zu Command-and-Control-Server (USA)

```
info {
  v4.02V_Unix64
  ubuntu22
  4x Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz
}
```

Beispiel: Mirai Malware

Dynamische Analyse

- Strace zeigt Name des Hackers

```
write(1, "kovey/cursinq was here, go away!", 32kovey/cursinq was here, go away!) = 32
unlink("/sbin/reboot")                  = 0
...
fork()                                  = 5079
```

- Um fork zu folgen extra Option -ff benötigt

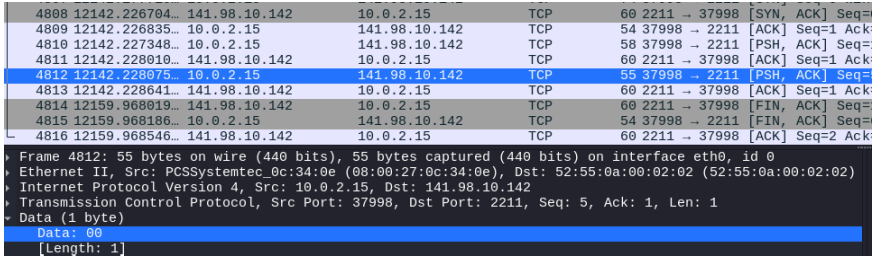
  => VM schaltet sich aus

- In gdb Prozess anhalten der überprüft, ob gerade gedebuggt wird.

Beispiel: Mirai Malware

Dynamische Analyse

- Kommunikation mit Command-and-Control-Server



- Server gerade inaktiv

Malware Erkennung

Erkennungsmethoden

Malware Erkennung

Signaturbasierte Erkennung

- Signaturen in Malware-Code
- Tools: YARA, ClamAV
- Schnelle Erkennung bekannter Malware
- Unzureichend bei unbekannten Varianten

Verhaltensbasierte Erkennung

- Überwacht System- und Audit-Logs
- Netzwerk: zeek, System: auditd
- Zero-Day-Angriffe
- Hohe Fehlalarme möglich

Beispiel: Prometei

Malware Erkennung

- Befehl: `clamscan -v prometei_sample.elf`
- Erkennung durch ClamAV:
  - 3 von 5 Varianten nicht identifiziert
  - Varianten zeigen einfache Änderungen im Code

Beispiel: Prometei

Malware Erkennung

- Beispiel für YARA-Regel, die „uplugplay“ als Merkmal nutzt:

```
rule Prometei
{
    strings:
        $binary = "uplugplay"
        $alt_bin = "Bon=UPlug"
    condition:
        $binary or $alt_bin
}
```

- Erkennung aller 5 Sample

Beispiel: Shikitega

Malware Erkennung

- Shikitega nutzt Verschleierungstechniken Shikata Ga Nai
- Problem: String-basierte YARA-Regeln schlagen fehl, da der Code verschlüsselt ist
- Lösung: Erkennung durch musterbasierte YARA-Regeln, die typische XOR-Schleifen erkennen
- Erkennung aller Shikitega-Varianten

Fazit

Fazit

- Zunehmende Bedrohung durch Linux-Malware
- Statische Analyse:
  - Liefert gefahrlose Einblicke in Struktur
  - Herausforderungen durch Packmethoden
- Dynamische Analyse:
  - Sandboxing und Debugging
  - Debuggingsperren können Analyse erschweren
- Malware-Erkennung:
  - Neuere Samples können durch Signaturen nicht immer erkannt werden
- Problem bei Analyse:
  - Samples nicht immer verfügbar
  - Command-and-Control-Server sind oft offline

Fragen?