FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Live Response USB Stick für Linux Systeme

Mark Zumbruch (vertreten durch Stefan Nagel)

Inhaltsverzeichnis

Ziele

- Entwicklung eines USB Tools zur Datenaufnahme von Linux Systemen
- GUI
  - zur einfachen Verwaltung verschiedener Konfigurationen
  - sowie dem Export von Berichten

Anwendungen

- Untersuchung eines Systems auf Virenbefall
- Sichern von flüchtigen Daten die nach ausschalten des Systems verloren gehen.
- Beweissicherung (Inhalt von verschlüsselten Laufwerken)

Motivation

- Es existiert noch keine umfassende Live Response Implementierung für Linux.
- Entwicklung einer Software die auch von Personen mit minimalen Linux Kenntnissen eingesetzt werden kann.

Unterschiede: Live Response – Post Mortem Analyse

- Post Mortem
  - Dateisystem
  - Logs
  - Analyse Arbeitsspeicher-Images
  - Browserverlauf
  - ...
- Live Response
  - Laufende Prozesse
  - Arbeitsspeicher
  - Offene Ports
  - Inhalt verschlüsselter Laufwerke

Anforderungen Live-Response

- Verändert das Zielsystem in möglichst kleinem Umfang (Nachvollziehbar)
- Nutzt keine Programme und Funktionen des Zielsystems
- Überprüfung der Ergebnisse auf Änderungen/Validität
- Kann von externem Datenträger ausgeführt werden

Existierende Live Response Tools

- Google Rapid Response (GRR)[1]
  - Client/Server Modell
  - Live Remote Forensics
- Linux Evidence Collection Toolkit (LECT)[2]
  - Erfüllt alle Anforderungen
  - Nicht verfügbar
- FastIR Collector Linux[3]
  - Live Response Skript
  - Keine Verwaltungsoberfläche
  - Erfordert zusätzlichen Interpreter auf zu untersuchendem System.

Datenaufnahme

- Allgemeine Systeminformationen
  - Reflektieren den Zustand des Systems (Uhrzeit / Art des Betriebssystems)
- Benutzer und Gruppeninformationen
  - Rechte von Benutzern / Eingeloggte Benutzer
- Prozesse und Dienste
  - Laufende Prozesse/Dienste / von diesen genutzte Dateien
- Netzwerkinformationen
  - Netzwerkadapter / Offene Ports
- Arbeitsspeicher
  - Anlegen eines Speicherabbildes
- Log Dateien
  - Sichern von System/Programm Logs
- Backup des Inhalts verschlüsselter Laufwerke

Umsetzung – Datenaufnahme Tool

- Bash Skript
- Modularer Aufbau
  - Netzwerk/Sysinfo/User/RAM/Log/Cpfolder
- Statisch Kompilierte Software
  - Busybox
- Von USB Stick ausführbar
- Leicht erweiterbar

Technische Limitationen – Datenaufnahme Tool

- Root-Rechte benötigt
- USB-Stick muss mit Ausführbarkeits-Rechten eingehängt werden

Umsetzung - GUI

- In Ruby als Serveranwendung
- Verwaltung verschiedener Konfigurationen
- HTML-Export für durchgeführte Datenaufnahmen
- Zusatzinformationen für den Export via Konfigurationsdatei

Ausführung / Ablauf

- Datenaufnahme Tool auf USB Stick Kopieren
- Anlegen eines Profils via GUI
- USB Stick in Zielsystem einstecken
- USB Stick in das Dateisystem einhängen
- Per Konsole in das USB Verzeichnis wechseln
- Starten der Datenaufnahme mit Profilnamen
- Entfernen des USB Sticks
- Export des Berichtes über die GUI

HTML Bericht - Aufbau

Bericht: BASIC

- Start
- Log

Sysinfo

- Passwort Datei
- Speicherplatz
- Group
- Shadow
- Laufende Prozesse
- RAM Info
- Datum
- Laufwerke
- USB Geräte
- Kernel Version
- Kernel Module
- Geöffnete Dateien
- Hosts
- CPU Info

Network

- IP Addresse
- lsof
- arp
- netstat

Log

Datei: sysinfo.mount.html

Zeigt eingehängte Laufwerke und Mountpunkte

Checksumme verifiziert

```
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
dev on /dev type devtmpfs (rw,nosuid,relatime,size=4051276k,nr_inodes=1012819,mode=755)
run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)
/dev/sdc2 on / type ext4 (rw,relatime,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/syst
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/net_cls type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=33,pgrp=1,timeout=0,minproto=5,maxproto=5,dir
mqueue on /dev/mqueue type mqueue (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,relatime)
configfs on /sys/kernel/config type configfs (rw,relatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev)
```

Versuche

- Testen der Implementierung
  - Szenario 1 - Netcat/Reverse Shell
  - Szenario 2 - Hausdurchsuchung

Versuche – Szenario 1

- Netcat/Reverse Shell
- Aufbau
  - „Infiziertes System“ baut Verbindung zu einem entfernten Rechner auf der Shellzugriff erhält.
  - Autostart Verbindungsaufbau beim einschalten des Systems
  - Durch die remote Shell wird der Befehl „ping google.com“ ausgeführt.
- Verwendete Software
  - Netcat

Szenario 1 - Artefakte

- „ps aux“ - Laufende Prozesse:

  PID USER   TIME   COMMAND

  2246 vagrant 0:00   {backdoor.sh} /bin/bash /home/vagrant/backdoor.sh

- „pstree“ - Prozesse als Baumstruktur

  systemd-+-VBoxClient---VBoxClient---{SHCLIP}

  ...

  |-login---bash-+-backdoor.sh---sh---ping

  | ‘-startx---xinit-+-Xorg

Szenario 1 - Artefakte

- „ps aux | grep ping“:

  PID USER TIME COMMAND

  2673 vagrant 0:00 ping google.com

- „netstat -atn“ - Aufgebaute Verbindungen

  Active Internet connections (servers and established)

  Proto Recv-Q Send-Q Local Address Foreign Address State

  ...

  tcp 0 0 192.168.50.4:39720 192.168.50.1:9099 ESTABLISHED

Versuche – Szenario 2

- Hausdurchsuchung
- Aufbau
  - Aufsetzen eines entfernten Laufwerks (Netzwerk)
  - Anlegen eines Verschlüsselten Laufwerks
  - Installation und Download mit einem Bittorrent Clienten.
- Verwendete Software
  - Vera Crypt
  - rsync
  - Deluge

Szenario 2 – Artefakte Fileserver

- „mount“:

vagrant@192.168.50.6:/home/vagrant/data/ on \

/home/vagrant/fileserver type fuse.sshfs \

(rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)

- „Netstat -atn“

Active Internet connections (servers and established)

Proto ... Local Address Foreign Address State

tcp 192.168.50.5:46900 192.168.50.6:22 ESTABLISHED

Szenario 2 – Artefakte Fileserver

- „ps aux“:

```
PID USER TIME COMMAND
3842  root     0:13 ssh -x -a -oClearAllForwardings=yes -2\
      vagrant@192.168.50.6 -s sftp
3843  root     0:06 sshfs -o allow_other\
      vagrant@192.168.50.6:/home/vagrant/data/\
      /home/vagrant/fileserver/
```

Szenario 2 – Artefakte Verschlüsseltes Laufwerk

- „mount“:

veracrypt on /tmp/.veracrypt_aux_mnt1 type fuse.veracrypt

(rw,nosuid,nodev,...)

/dev/mapper/veracrypt1 on /media/veracrypt1 type vfat (rw,relatime,...)

- „lsof“ - Ort des eingehangenen Containers

...

3872 /usr/bin/veracrypt /home/vagrant/encrypted-container

3872 /usr/bin/veracrypt /dev/null

3872 /usr/bin/veracrypt /dev/fuse

Szenario 2 – Artefakte Verschlüsseltes Laufwerk

- Cpfolder Modul – Inhalt des Laufwerks:

report_03_09_2017__11-13/cp/media/veracrypt1/images/

cat-treats.jpg

dbe5f0727b694870
16ffd67a6689e75a.j
peg

photo.jpg

report_03_09_2017__11-13/cp/media/veracrypt1/

images

illegal-file1.txt

Szenario 2 – Artefakte Verschlüsseltes Deluge

- „ps aux“:

PID USER TIME COMMAND

...

4479 vagrant 0:43 {deluge-gtk} /usr/bin/python /usr/bin/deluge-gtk

- „lsof“:

4479 /usr/bin/python2.7 socket:[36040]

4479 /usr/bin/python2.7 /home/vagrant/fileserver\

/pclinuxos64-mate-2017.02/pclinuxos64-MATE-2017.02.iso

...

Zukünftige Arbeit

- Unterstützung weiterer Architekturen
- Format der Ergebnisse
- Entwicklung weiterer Module
  - Sichern von Konfigurationsdateien/SSH Keys/...
- Umgehen von Sicherheitsmaßnahmen

Vielen Dank für die Aufmerksamkeit

Referenzen

[1] https://github.com/google/grr

[2] J. Choi, A. Savoldi, P. Gubian, S. Lee and S. Lee, "Live Forensic Analysis of a Compromised Linux System Using LECT (Linux Evidence Collection Tool)," 2008 International Conference on Information Security and Assurance (isa 2008), Busan, 2008, pp. 231-236.

[3] https://github.com/SekoiaLab/Fastir_Collector