Sicherer Umgang mit Ubuntu im Netz

Dr. Erwin Hoffmann

20. September 2010

Ubuntu Linux ist nicht nur bei den Professoren am FB 2 beliebt, auch viele Studenten sehen das offene Unix-System mit seiner funktionalen und graphischen Benutzerführung als willkommene und preisgünstige Alternative zu Microsoft *Windows* und Apple *MacOS X*.

Im Gegensatz zu den häufig in der Öffentlichkeit diskutierten Sicherheitsmängeln – besonders bei Windows – wird Linux nicht unbedingt ein höheres Mass an Fehlerfreiheit, aber doch weitgehende Immunität gegenüber Viren attestiert.

Die Hauptgefahr der Infektion von Windows-Systemen geht heute nicht mehr von 'vervirten' Anhängen in Emails aus, sondern beim Web-Surfen per 'drive-by'. Unter Verwendung des *Firefox*-Browsers und mittels geeigneter Ergänzungen, wie *NoScript*, durch das speziell die potentielle IFRAME-Verarbeitung ausgeschlossen wird, lassen sich die Risiken weitgehend minimieren. Die Anfälligkeit der nachgelagerten Software-Produkte, wie dem *Adobe PDF-Reader* und *Quicktime*, gegenüber speziellen sog. Zero-Day Attacken bleibt hingegen bestehen und können daher trotz Verwendung eines aktuellen Virenschutzes das gesamte System beeinflussen.

Einsatz von eGroupWare

Die Hauptgefahr beim Einsatz von Linux geht hingegen von sog. *Rootkits* aus. Im Rahmen mehrerer Diplomarbeiten wurde am Fachbereich ein Linux-Server mit Ubuntu 9.10 und der Software *eGroupWare* aufgesetzt, die als Projekt-Management-Software und zur Code-Verwaltung eingesetzt werden sollte. Hierbei war auch vorgesehen, dass der Rechner über das Internet erreichbar sein sollte: Projektdokumentation, Aufgabenverwaltung und das Einstellen von Fehler-Tickets sollte über eine **HTTP**-Verbindung remote möglich sein; sowie speziell Source Code mittels **SCP** hochgeladen werden können.
Hierzu wurde der Rechner bidirektional für die Ports:

- HTTP (80)

- HTTPS (443) und
- SSH (22)

geöffnet.

eGroupWare ist eine **PHP**-Applikation und gilt daher als nicht unbedingt sicher. Zudem werden potentiell sensitive Daten übertragen, was den Einsatz einer Transport-Verschlüsselung mittels **HTTPS** notwendig macht. Zur Vermeidung einer unbefugten Nutzung wurde des weiteren der Benutzerkreis mittels der **Apache**-Option *.htaccess* eingeschränkt, indem vor dem eigentlichen Zugang zu *eGroupWare* ein zusätzliches Passwort mandatorisch abgefragt wird.

Der Rechner mit der *eGroupWare* Software blieb als Server beständig online und mit dem Internet verbunden; solange zumindest bis die ausstehenden Diplomarbeiten abgeschlossen sein sollten. Mit den vorgestellten Massnahmen schien der Sicherheit zunächst genüge getan zu sein; ein fataler Irrtum, wie sich herausstellen sollte.

DFN-CERT#24618

Am 27. Juli gab das DFN-CERT das Ticket #2618 aus:

> "... soeben erreicht uns ein Hinweis vom DFN-Cert, dass es von einem Rechner der FH (VLAN 1246 - fb2_inf_6 - 194.94.82.250 - Gb.1 Raum 149.2.1.4) aus zu massiven SSH-Loginversuchen sowie scanns nach Schwachstellen gekommen ist."

Der Rechner mit der IP-Adresse 194.94.82.250 war unser Projekt-Rechner!

Nach dem sofortigen Abklemmen und späteren eingehenden 'forensischen' Untersuchungen konnten auf dem Rechner eine Vielzahl von Trojanern und infizierten Dateien identifiziert werden. Einige Beispiele:

```
/bin/egrep: Linux.RST.B-1 FOUND
/bin/umount: Linux.RST.B-1 FOUND
/bin/nano: Linux.RST.B-1 FOUND
/bin/nc.traditional: Linux.RST.B-1 FOUND

/usr/lib/libsh/.sniff/shsniff: Trojan.Linux.Sysniff FOUND
/usr/lib/libsh/shsb: Linux.LionCleaner FOUND

/sbin/ttymon: Trojan.Linux.Rootkit.A FOUND
/usr/bin/pstree: Trojan.Rootkit-118 FOUND
```

Wie konnte das geschehen ?

Die Verbreitung von Trojaner und Rootkits geschieht unter Unix typischerweise über schlecht gesicherte Benutzer-Accounts. Zugriff auf das System erhält ein Angreifer über eine Brute-Fore Attacke, in dem per sog. *Dictionary*-Scan einfach alle Möglichkeiten von Benutzername und Passwort ausprobiert werden.

Die Wahrscheinlichkeit eines erfolgreichen Versuchs hängt – sofern keine weiteren Massnahmen wie *TCP-Wrapper* oder *Port-Knocking* benutzt werden – ausschliesslich von der Gesamt-Entropie des Gespanns 'Benutzername+Passwort' ab. Ist diese entsprechend hoch, laufen Angriffe ins Leere.

Auf meinem Internet-Server finden sich häufig Angriffsversuche mit folgenden typischen Einträge in der Logdatei '/var/log/auth.log':

```
reverse mapping checking getaddrinfo for bd20d434.virtua.com.br failed -
  POSSIBLE BREAKIN ATTEMPT!
Illegal user fluffy from ::ffff:189.32.212.52
reverse mapping checking getaddrinfo for bd20d434.virtua.com.br failed -
  POSSIBLE BREAKIN ATTEMPT!
Illegal user admin from ::ffff:189.32.212.52
reverse mapping checking getaddrinfo for bd20d434.virtua.com.br failed -
  POSSIBLE BREAKIN ATTEMPT!
Illegal user test from ::ffff:189.32.212.52
```

Wir sehen, dass hier der Angreifer Annahmen über den Namen des Benutzeraccounts vornehmen muss; wie oben gezeigt z.B. 'fluffy', 'test' oder 'admin'.

Da unter Unix *immer* der Account 'root' existieren muss, ist dieser in der Regel nur lokal, d.h. auf dem Rechner selbst nutzbar. Bei Ubuntu lautet hierzu das Kommando **sudo su**, mit dem ein Benutzer (temporäre) *root-Rechte* erhält.

SSHD unter Ubuntu

Damit der Rechner per **SSH** (*Secure Shell*) bzw. **SCP** (*Secure Copy*) über das Netz erreichbar ist, muss zunächst ein Dienst eingerichtet werden: der SSH-Daemon (**SSHD**). Die Konfiguration des SSH-Dienstes wird über die Kontrolldatei '/etc/ssh/sshd_config' vorgenommen und braucht in der Regel nicht angepasst werden.

Bei aller Sorgfalt, mit der der Hersteller von Ubuntu (*Canonical*) das System sicher gemacht haben – bei der Konfigurationsdatei 'sshd_config' liegt ein klassischer Lapsus vor:

```
# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
```

In der Default-Einstellung wird somit der Root-Zugriff erlaubt ('PermitRootLogin yes')!

Solange kein Root-Passwort gesetzt ist, wirkt sich dies nicht aus, da nachfolgend keine Benutzer-Shell gestartet wird. Auf dem Projekt-Rechner musste aber für andere Dienste ein Root-Passwort gesetzt werden, da diese ansonsten nicht laufen. Als fatale Folge hieraus ist die Sicherheit des Systems für den *remoten Zugriff* deutlich geringer als beim 'switch user' über das **sudo** Werkzeug!

Die effektive Entropie des Root-Accounts reduziert sich auf die Stärke des Root-Passworts selbst; und diese war einfach zu gering. Hierdurch stand dem Angreifer der Zugang zum Projektrechner praktisch Scheunentor-weit auf – ohne, dass dies dem Anwender bewusst wurde. Im Gegenteil, die vorgebliche Sicherheit unter Ubuntu verschleierte das eigentliche Problem.

Fazit

Wird bei Ubuntu der Dienst **SSHD** benutzt, sollte ausdrücklich beachtet werden, den 'PermitRootLogin no' zu setzen, andernfalls besteht die begründete Möglichkeit, das System über Dictionary-Angriffe zu kompromittieren.

Ferner sollten selbstverständlich immer 'starke' Passwörter zur Sicherung der Unix-Accounts gewählt werden. Unix Account-Namen mit gemischter Gross/Kleinschreibung sind ebenfalls ein probates Mittel, die Sicherheit des Gesamtsystems zu erhöhen.

Bleibt zu erwähnen, dass in Bezug auf den Projekt-Rechner, die Aktionen des Angreifers ermittelt werden konnten; einschliesslich dessen Email-Adresse!