

# Security Tools 5: tripwire

*Bernd Lehle / Oliver Reutter*

**"Er ... ging in die Kü che und nä herte sich vorsichtig dem Kü hlschrank. Dirk kauerte sich vor den Kü hlschrank und inspizierte eingehend den Rand der Tü r. Er fand wonach er suchte. Besser gesagt, er fand mehr als wonach er suchte. Nahe der Unterseite der Tü r, auf dem schmalen Spalt, der die Tü r von dem eigentlichen Kü hlschrank trennte und der den grauen Dichtungsgummistreifen enthielt lag ein einzelnes menschliches Haar. Es war dort mit getrockneter Spucke befestigt. Das hatte er erwartet. Er hatte es selbst vor drei Tagen dort hingeklebt und seitdem mehrere Male nachgesehen. ..."**

(Auszug aus *Der lange dunkle Fünfuhrtee der Seele* von Douglas Adams)

Bisher haben wir hauptsächlich Tools vorgestellt, die helfen das eigene System gegen Angriffe sicherer zu machen. Etwa einmal im Monat stehen wir allerdings als Arbeitsgruppe Systemsicherheit vor dem Problem, daß es irgendwo an der Universität Stuttgart jemandem gelungen ist, vorhandene/nicht vorhandene Sperren eines Rechners zu überwinden und dort Systemprivilegien zu erhalten. Die betroffenen Systembetreuer stehen dann meist vor dem Problem, ihre Maschinen so schnell wie möglich wieder sicher und benutzbar zu machen.

Das kann sehr schnell in eine Sisyphos-Arbeit ausarten, da gewitzte Angreifer sofort nach dem Eindringen Hintertüren installieren, die ihnen in Zukunft den Zugang zu dem betroffenen Rechner leichter machen, selbst wenn ihre erste Einstiegs Luke entdeckt und gestopft wurde. Es gibt bereits fertige Programmpakete (root kits) die veränderte Versionen einiger wichtiger Systemdateien (`ifconfig`, `telnetd`, etc.) enthalten, mit denen man in Minutenschnelle ein einmal geknacktes System mit zahlreichen weiteren Löchern versehen kann. Die veränderten Versionen sind oft auf den ersten Blick nicht von den Originalen zu unterscheiden.

Was kann nun der geplagte Systembetreuer tun, wenn er unter diesem DamoklesSchwert sein System so schnell wie möglich wieder flott bekommen will ?

Die sicherste Methode ist natürlich die Neuinstallation von einem sauberen Boot-Medium. Das einzige Problem liegt dann darin, sicherzustellen, daß sich unter den Benutzerdaten kein trojanisches Pferd in Gestalt eines `.rhosts`-Files oder eines SetUID-Programmes befindet.

Eine elegantere Methode ist es, die Dateigrößen und Zugriffsdaten mit dem letzten Backup zu vergleichen, bzw. diesen einfach wieder einzuspielen. Problematisch ist hierbei, daß man nicht sicher sein kann, was der letzte saubere Backup ist, wenn der Angriff erst Wochen oder Monate später entdeckt wurde. Außerdem ist es für qualifizierte Angreifer kein Problem, Dateigröße, Zugriffszeiten und sogar einfache Prüfsummen der veränderten Programme gezielt zu fälschen.

Wenn man sich allerdings schon vor einem solchen Angriff mit dem Tag X auseinandersetzt, gibt es ein Programm, daß einem im Ernstfall viel Arbeit sparen kann und die An-griffswege und Hintertüren aufdecken hilft. In Analogie zu unserer einleitenden Story wurde es von seinen Autoren `tripwire` (Stolperdraht) genannt.

`tripwire` durchsucht einen bestimmten Teil der Dateisysteme nach allen Dateien und speichert wichtige Informationen wie Zugriffszeiten, Inodes und Längen ab. Zusätzlich werden von jeder Datei zwei unabhängige kryptographische Prüfsummen (snefru und MD-5) berechnet. Im Gegensatz zu der Standard-Unix-Prüfsumme `sum` sind diese kryptographischen Summen nicht gezielt fälschbar. Sollte es doch mit einer der beiden gelingen, was einen mehrmonatigen Rechenaufwand bedeuten würde, steht die andere als Backup zur Stelle.

So kann nach einmaligem Aufbau einer Datenbank immer wieder genau nachvollzogen werden, welche Dateien verändert wurden.

## Installation von tripwire

Am besten besorgt man sich die aktuelle Version:

```
ftp://ftp.uni-stuttgart.de/pub/unix/security/Tripwire-1.2.tar.gz
```

```
ftp://ftp.cert.dfn.de/pub/tools/admin/Tripwire/Tripwire-1.2.tar.gz
```

Nach dem Auspacken mit `gunzip` und `tar` sollte im `Makefile` sichergestellt werden, daß die Definitionen für C-Compiler und Optionen richtig gesetzt sind. Tips für die richtigen Einstellungen befinden sich in der `Ported-Datei`.

Dann sollte im Verzeichnis `./configs` nach einem passenden `conf-<os>.h` gesucht werden, das am besten zum lokal verwendeten Betriebssystem paßt. Dort wird hauptsächlich festgelegt, welche Teile des Dateisystems mit einbezogen werden. Teile, die sich im Normalbetrieb oft ändern wie Spoolbereich, Logfiles oder Benutzerdaten müssen natürlich ausgeklammert werden.

Nun wird `./include/config.h` den lokalen Gegebenheiten entsprechend angepaßt. Hier einige Auszüge aus einer Beispielinstallation:

```
#include "../configs/conf-svr4.h"
#define CONFIG_PATH "/opt/sonst/tripwire-1.2/bin/databases"
#define DATABASE_PATH "/opt/sonst/tripwire-1.2/bin/databases"
```

In dieser Datei wird auch festgelegt, wo die Datenbank für `tripwire` später angelegt wird. Von den Autoren des Programms wird empfohlen die Pfade, die in `CONFIG_PATH` und `DATABASE_PATH` spezifiziert sind auf eine Partion zu

legen, die nur lesbar ist, um Modifikationen entgegenzuwirken. Am sichersten ist natürlich die Lagerung der Daten auf Disketten, zumal die Datenbanken meist klein genug sind, um auf eine Diskette zu passen.

Die eben erwähnte Konfigurationsdatei muß dann unter dem Namen `tw.config` im Verzeichnis `CONFIG_PATH` stehen.

Ein Auszug aus einem `tw.config` file:

```
# Unix itself
/kernel/unix                                R
# Now, some critical directories and files
# Some exceptions are noted further down
/dev                                         L
/devices                                    L
=/devices/pseudo                            L
/etc                                         +pnugsm12-iac
```

Hierbei besteht ein Eintrag pro Zeile aus zwei Feldern, die durch ein TAB getrennt sind. Das erste Feld bezeichnet die Datei oder das Verzeichnis das von `tripwire` untersucht wird, wobei man auch mit Hilfe der Operatoren `!` und `=` Dateien und Verzeichnisse ausschließen kann. Im zweiten Feld gibt man an, welche Dateiattribute in die Datenbank aufgenommen werden. Dies kann detailliert erfolgen, wie es bei `etc` de-monstriert ist oder man kann die vordefinierten Makros `R`, `L`, `N` und `E` benutzen.

Die genaue Bedeutung der Operatoren, Modifikatoren und Makros ist im Vorspann der Beispiel-config-Dateien erläutert.

Zum Schluß ist eigentlich nur noch `make` auf der obersten Ebene des `tripwire`-Dateibaums aufzurufen.

## Benutzung von `tripwire`

Bevor überhaupt ein Vergleich mit den gespeicherten Daten möglich ist, muß mit dem Kommando eine Referenzdatenbank erzeugt werden:

```
# ./tripwire -initialize
### Phase 1: Reading configuration file
### Phase 2: Generating file list
### Phase 3: Creating file information database
```

Jetzt wird eine Datei `tw.db_[hostname]` generiert. Sie wird an der in der Variable `DATABASE_PATH` angegebenen Stelle abgespeichert.

Bei den regelmäßigen Kontrollen (am besten wöchentlich) startet man `tripwire` einfach mit `# ./tripwire` oder `# ./tripwire -interactive`

Im ersten Fall wird nur über gefundene Unterschiede berichtet, was dann auch in regelmäßigen cron-Jobs gemacht werden kann. Im zweiten Fall besteht auch die Möglichkeit die Unterschiede interaktiv als legitim zu kennzeichnen und damit die Datenbank gleich auf den neusten Stand zu bringen. Es gibt noch einige weitere Optionen, die man mit `./tripwire -help` erhält, die aber im Routinebetrieb kaum von Bedeutung sind.

**Tip:** Einige der Dateien, die `tripwire` lesen muß, um Prüfsummen zu berechnen, sind nur für `root` lesbar. Daher sollte es immer nur als `root` benutzt werden, um unnötige Fehlermeldungen und falsche Ergebnisse zu verhindern.

## Interpretation der Ergebnisse

Man wird sich wundern, wieviele unerwartete Unterschiede beim ersten Durchlauf auftauchen. Nach und nach wird man dann allerdings feststellen, daß beispielsweise das Betriebssystem jedes Wochenende seine Platten neu strukturiert oder seine logfiles archiviert. Es erinnert an Programme, die abends noch mal schnell installiert und dann vergessen wurden. Ein zuverlässiges Bild des Dateisystems, das im Falle eines zu reproduzierenden Angriffs wertvolle Informationen liefert, entsteht erst nach einigen Durchläufen und Veränderungen der `tw.config`-Datei.

## Literaturverzeichnis

- Kim, G. H., Spafford, E. H. : Experience with Tripwire: Using Integrity Checkers for Intrusion Detection, Purdue Technical Report CSD-TR-93-071, 21 February 1994
- Kim, G. H., Spafford, E. H. : Writing, Supporting, and Evaluating Tripwire: A Publically Available Security Tool, Purdue Technical Report CSD-TR-94-019, 12 March 1994
- Kim, G. H., Spafford, E. H. : The Design and Implementation of Tripwire: A File System Integrity Checker, Purdue Technical Report CSD-TR-93-071, 19 November 1993

## Warnung!

Die hier gegebenen Hinweise reichen nicht zum vollständigen Gebrauch des Programms aus. Jeder, der die Verwendung des Programms erwägt, sollte gründlichst die beigefügte Dokumentation durchlesen und verstehen was er tut.

Für Risiken und Nebenwirkungen lesen Sie die Programmdokumentation oder fragen Sie Ihren zuständigen Security-Administrator.

## Ansprechpartner in sicherheitsrelevanten Fragen

An wen kann ich mich in so einem Fall wenden? Hier zwei wichtige Adressen und Ansprechpartner:

[sneakers@rus.uni-stuttgart.de](mailto:sneakers@rus.uni-stuttgart.de)  
[dfncert-request@cert.dfn.de](mailto:dfncert-request@cert.dfn.de)

Bernd Lehle, NA-5531  
E-Mail: [lehle@rus.uni-stuttgart.de](mailto:lehle@rus.uni-stuttgart.de)

Oliver Reutter, NA-4513  
E-Mail: [oliver.reutter@rus.uni-stuttgart.de](mailto:oliver.reutter@rus.uni-stuttgart.de)