HOWTO zu "tripwire"

$Id: tripwire-HOWTO.txt,v 1.19 2019/11/26 19:37:07 tsbirn Exp $

Dieses Dokument beschreibt die Absicherung eines Linux-Systems Ã¼ber das
HIDS (Host Intrusion Detection System) "Tripwire" ("Stolperdraht").

INHALTSVERZEICHNIS

---

1) EinfÃ¼hrung

Tripwire (Stolperdraht) ist ein "Host Intrusion Detection System" (HIDS) oder
besser ein "System Integrity Verifier". Es erstellt eine Momentaufnahme Ã¼ber
den Zustand eines (Datei)Systems anhand von "Richtlinien" (Policies). Diese
wÃ¤hlen Verzeichnissen und Dateien des Dateisystems eines Rechners aus und
zeichnen bestimmte Eigenschaften von ihnen (GrÃ¶Ã¬_e, Besitzer, Zugriffrechte,
Alter, ...) in einer Spezialdatenbank auf.

Die "Richtlinien" (Policies) mÃ¼ssen sorgfÃ¤ltig definiert werden, da spÃ¤ter nur
genau diese Eigenschaften geprÃ¼ft werden. Zu strenge/laxe oder zu viele/wenige
Vorgaben fÃ¼hren zur Unbrauchbarkeit der generierten Datenbank.

Ein per Tripwire abzusicherndes System sollte sich im "sauberen Zustand"
befinden, damit die Datenbank nicht den Grundzustand eines bereits
kompromittierten Systems aufzeichnet (die Datenbank wÃ¤re dann wertlos):

  * Frisch installiert
  * Installations-Daten aus vertrauenswÃ¼rdiger Quelle (CD, DVD, Signatur)
  * Noch keine Netzwerkverbindung eingerichtet/verwendet

SpÃ¤ter kann Tripwire den AKTUELLEN Dateisystem-Zustand gegen den in der
Datenbank abgelegten Grundzustand prÃ¼fen. Bei Abweichungen generiert es einen
"Bericht" (Report), Ã¼ber den der Administrator entscheiden kann, von welchem
Typ die Abweichung ist:

  * Gewollt (z.B. Update, Konfiguration, SW-Inst.) --> Neu in Datenbank aufnehmen
  * Unbefugt (Fehler, Angriff) --> PrÃ¼fen ob gefÃ¤hrlich

D.h. es verhindert keinen Angriff, sondern erkennt einen Angriff "indirekt",
nÃ¤mlich NICHT online, sondern erst zu einem spÃ¤teren Zeitpunkt (offline) an
seinen Auswirkungen auf das Dateisystem.

Werden Abweichungen festgestellt, die auf Grund von Umkonfiguration,
Aktualisierung, Aufspielen oder Entfernen von Software-Paketen entstanden sind,
kÃ¶nnen diese Ã¬-^Dnderungen als neuer Ist-Zustand zur vorhandenen Datenbank
hinzugefÃ¼gt werden, damit sie nicht stÃ¤ndig "reported" werden (spart Zeit
gegenÃ¼ber Neu-Generierung und erlaubt eine feingranulare Kontrolle).

Die fÃ¼r Tripwire notwendigen ASCII-Konfigurationsdateien werden nach ihrer
Definition in ein binÃ¤res Format umgewandelt und mit einem "Site Key" signiert
(kryptografisch gesichert), das bringt Sicherheit und Performanz.

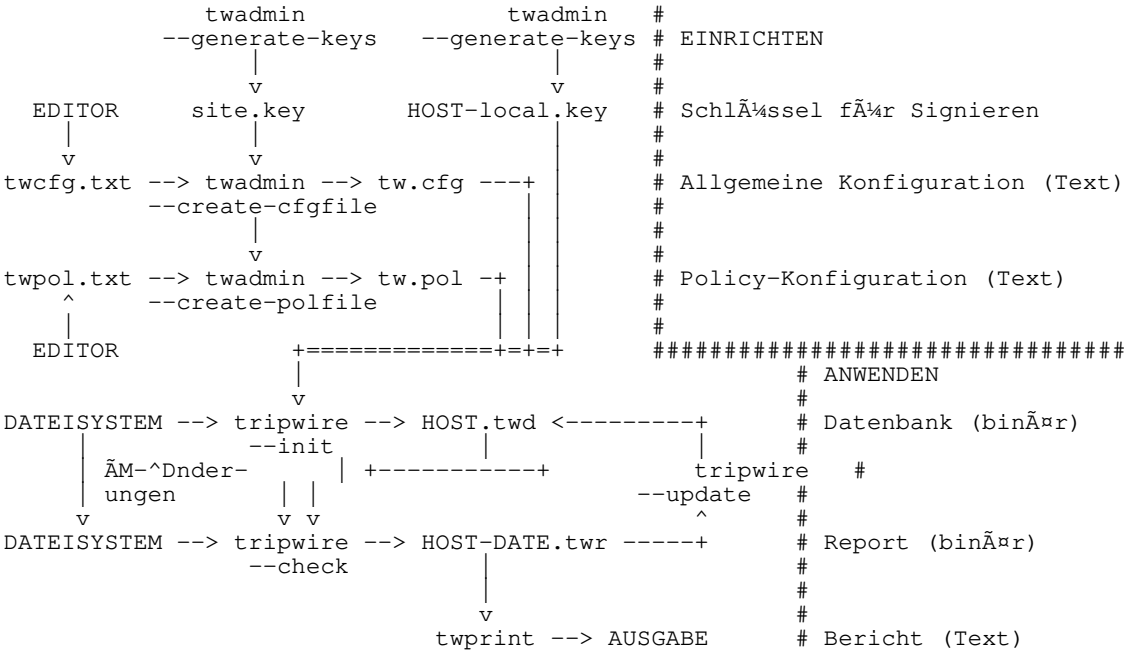
Die von Tripwire generierte Datenbank hat ebenfalls ein binÃ¤res Format und wird
durch einen "HOST Local Key" signiert (kryptografisch gesichert), das bringt
Sicherheit und Performanz.

Die beiden Keys (SchlÃ¼ssel) mÃ¼ssen 1x ganz am Anfang wÃ¤hrend der Einrichtung
von Tripwire mit "twadmin" erzeugt werden. Sie sollten mit je einer
"Passphrase" gesichert werden, damit sie nicht im Klartext im Dateisystem
liegen. Die Trennung der beiden SchlÃ¼ssel erlaubt eine Trennung der Aufgaben:

  * Eine Person definiert die zu prÃ¼fenden Richtlinien
  * Jemand anderes fÃ¼hrt die eigentliche PrÃ¼fung des Dateisystems durch

2) Ã‚M-^\bersicht zum Ablauf einer Tripwire-Konfiguration und -Anwendung
---

Folgende ZusammenhÃ¤nge zwischen den Kommandos (inkl. Optionen) und den Dateien gibt es:



Der Ablauf der Absicherung eines Systems durch Tripwire lÃ¤sst sich folgendermaÃM-^_en Ã¼bersichtlich darstellen. In Abschnitt 3) werden zu jedem Schritt 0-11 die dabei verwendeten Befehle beschrieben.

E	0	Rechnernamen ermitteln ("hostname")
I	1	Host Local Key "HOST-local.key" generieren (mit Local Passphrase)
N	2	Site Key "site.key" generieren (mit Site Passphrase)
I	3	Konfigurations-Datei "twcfg.txt" erstellen (ASCII, Editor)
C	4	Konfigurations-Datei "tw.cfg" generieren (binÃ¤r, Site Passphrase)
T	5	Richtlinien-Datei "twpol.txt" erstellen (ASCII, Editor)
E	6	Richtlinien-Datei "twl.pol" generieren (binÃ¤r, Site Passphrase)
N	7	BinÃ¤re Datenbank "HOST.twd" generieren (binÃ¤r, Local Passphrase)
A	8	Durch Vergleich System <-> Datenbank einen PrÃ¼fbericht generieren
N	9	ÃM-^Dnderung werden am (Datei)System vorgenommen PrÃ¼fbericht generieren --> Unterschiede Akzeptierte ÃM-^Dnderungen in DB nachtragen (Editor, Local Passphrase) PrÃ¼fbericht generieren --> Unterschiede (sollten keine mehr sein)
E	10	TÃ¤glich per Cronjob PrÃ¼fbericht generieren lassen
N	11	Neue/geÃ¤nderte Richtlinien in DB einpflegen (Site Passphrase)

2.1) Sicherheit

Die Klartext-Versionen der beiden Konfigurations-Dateien "twcfg.txt" und "twpol.txt" sollten nach der Generierung der binÃ¤ren Variante an einen sicheren Ort verschoben werden, damit Angreifer aus den darin stehenden Informationen keinen Nutzen ziehen kÃ¶nnen (z.B. nicht Ã¼berwachtes Verzeichnis erkennen).

Weiterhin sollte eine Kopie der generierten Datenbank "HOST.twd" auf ein Read-Only-Medium gemacht werden, damit der Angreifer sie nicht lÃ¶schen kann.

Tripwire kann sogenannte "Rootkits" (z.B. lrk3, lrk4, lrk5) entdecken, die einige wichtige Systembefehle gegen eigene Versionen austauschen. Bei "Kernelbasierten Rootkits" (z.B. Knark), die Systemaufrufe austauschen, hilft Tripwire allerdings nicht mehr, da der Kern eine andere Struktur und einen anderen Inhalt des Dateisystems vorspiegeln kann, als in Wirklichkeit vorliegt.

Hier helfen dann andere Tools wie "kstat" zur Analyse des Arbeitsspeichers.

Sollte auch der Kern durch "Kernel-Memory-Patching" unterwandert sein
(z.B. KIS), dann hilft ein System wie "LIDS" (Linux Intrusion Detection System)
weiter.

2.2) Eigenschaften

Von Tripwire kann pro Datei/Verzeichnis eine beliebige Auswahl folgender
EIGENSCHAFTEN ("Properties") Ã¼berprÃ¼ft werden (darf z.B. eine Datei wachsen,
sollte die GrÃ¶ÃM-^_e nicht geprÃ¼ft werden):

Code	Begriff	Eigenschaft
a	access	Zugriffsdatum
b	blocks	Anzahl DatenblÃ¶cke
c	change	Inode-ÃM-^Dnderungsdatum
d	device	GerÃ¤tenummer auf der Datei-Inode liegt
g	group	Besitzer-Gruppe (GID)
i	inode	Inode-Nummer
l	length	Datei darf wachsen (Verzeichnis?)
m	modify	Inhalt-ÃM-^Dnderungsdatum
n	number	Anzahl Hard-Links
p	permission	Zugriffsrechte und Dateimodus
r	major/minor	Device Nummer (nur fÃ¼r GerÃ¤te)
s	size	DateigrÃ¶ÃM-^_e
t	type	Dateityp (File, Verz., Symlink,...)
u	user	Besitzer (UID)

Durch Auflisten der einbuchstabigen Codes und Voranstellen von "+" und "-"
werden sie folgendermaÃM-^_en zu MASKEN (Kombination von zu prÃ¼fenden und zu
ignorierenden Eigenschaften) kombiniert:

Code	Begriff	Bedeutung
-	remove	Folgende Eigenschaften ignorieren
+	add	Folgende Eigenschaften prÃ¼fen

FÃ¤ngt eine Maske nicht mit "+" oder "-" an, dann gilt "+". Fehlt eine
Eigenschaft in einer Maske, dann wird sie nicht geprÃ¼ft (Ã¤quivalent "-" vor der
Eigenschaft). D.h alle folgenden Masken sind identisch:

```
+p+n                    # permission + number
pn                      # (analog)
+pn-g                   # (analog)
+pn-iugtsdrlbamcCMSH    # (analog)
```

Weiterhin kann in der Maske festgelegt werden, welche der mÃ¶glichen PrÃ¼fsummen
generiert werden sollen (sortiert nach steigendem Aufwand und zunehmender
Sicherheit, mind. zwei PrÃ¼fsummen sollten generiert werden):

Code	Bedeutung
C	CRC-32 Hashwert
M	MD5 Hashwert
S	SHA Hashwert
H	Haval Signatur

HÃ¤ufig benÃ¶tigte Masken ("Kombinationen" von Eigenschaften) sind Ã¼ber
vordefinierte Variable verfÃ¼gbar:

Name	Definition	Bedeutung
ReadOnly	+pinugtsdbmCM-rlacSH	Nur lesbar
Dynamic	+pinugtd-srlbamcCMSH	Dynamik erlaubt (Wachsen+Schrumpfen)
Growing	+pinugtdl-srbamcCMSH	Darf nur wachsen
Device	+pugsdr-intlbamcCMSH	GerÃ¤t (von Tripwire nicht zu Ã¶ffnen)
IgnoreAll	-pinugtsdrlbamcCMSH	Nur Vorhandensein/Abwesenheit prÃ¼fen
IgnoreNone	+pinugtsdrbamcCMSH-l	Startpunkt fÃ¼r eigene Regeln

Diese Kombinationen dienen als Ausgangspunkt fÃ¼r abweichende Masken durch
AnhÃ¤ngen von Eigenschaften mit "+" und "-":

```
$(IgnoreNone) -ar   # --> Alles auÃM-^_er Zugriffsdatum + Device-Nummer prÃ¼fen
```

```
In der Datei "twpol.txt" werden Ã¼ber "Richtlinien" (Regeln, Policies) Dateien
und Verzeichnisse mit den zu prÃ¼fenden Eigenschaften verknÃ¼pft. FÃ¼r jede
Datei/jedes Verz. kann nur EINE Richtlinie vorgegeben werden. Bei
Verzeichnissen kÃ¶nnen fÃ¼r Teile ihres Inneren abweichende Richtlinien
(AUSNAHME) vorgegeben werden.

  /bin        -> $(ReadOnly);    # Verzeichnis
  /etc        -> $(IgnoreAll);   # Verzeichnis
  /etc/passwd -> $(Growing);     # Datei in vorigem Verzeichnis

Soll eine Datei oder ein Verzeichnis NICHT geprÃ¼ft werden, so ist sie mit einem
"!" zu versehen.

  !/etc/init.d;
  !/etc/rc.d;
  !/etc/mnttab;

Eine Richtlinie kann mit einem "Namen" und einer "PrioritÃ¤t" (Wichtigkeitsgrad)
versehen werden. PrÃ¼fungen kÃ¶nnen auf Richtlinien mit einem bestimmten Namen
oder mit MindestprioritÃ¤ten beschrÃ¤nkt werden.

  /bin -> $(ReadOnly) (rulename = "Executables", severity = 0);

Jede Richtlinie ist mit Strichpunkt ";" abzuschlieÃM-^_en


2.3) Installation und Grund-Konfiguration
-----------------------------------------
Tripwire wird bei den meisten Distributionen mitgeliefert. Installieren von
Tripwire durch (i=install):

  rpm -i tripwire                # SuSE
  apt-get install tripwire       # Debian

PrÃ¼fung, ob Tripwire installiert ist durch (q=query, a=all):

  rpm -qa | grep tripwire        # SuSE   --> tripwire-2.3.1-186
  dpkg --list | grep tripwire    # Debian --> tripwire 2.3.1.2.0-11

Alle von Tripwire installierten Dateien auflisten (q=query):

  rpm -q --list tripwire
  dpkg --listfiles tripwire

Ergibt folgende Verzeichnisse bzw. Dateien:

  +-----------------------------------+-------------------------+
  | Pfad                              | Bedeutung               |
  +-----------------------------------+-------------------------+
  | /etc/tripwire/*                   | Konfiguration           |
  | /usr/sbin/*                       | Programme               |
  | /usr/lib/tripwire/databases/*     | Datenbank               |
  | /usr/share/doc/tripwire/*         | Dokumentation           |
  | /usr/share/doc/tripwire/examples/* | Beispiel-Konfiguration |
  | /usr/share/man/man8/*             | Manualpages             |
  | /var/lib/tripwire/*               | Datenbank               |
  | /var/lib/tripwire/report/*        | Berichte                |
  +-----------------------------------+-------------------------+

FÃ¼r Tripwire mÃ¼ssen folgende Dateien verwaltet werden:

  +----------------+-----------------------------------------------------+
  | Typ + man-Page | Zweck + Dateien                                     |
  +----------------+-----------------------------------------------------+
  | Key            | SchlÃ¼ssel zur Absicherung der Datenbank             |
  | (man twadmin)  |   /etc/tripwire/site.key                            |
  |                |   /etc/tripwire/HOST-local.key                      |
  +----------------+-----------------------------------------------------+
  | Configuration  | System-spezifische Information                      |
  | (man twconfig) |   /etc/tripwire/twcfg.txt --> /etc/tripwire/tw.cfg  |
  +----------------+-----------------------------------------------------+
  | Policy         | Richtlinien                                         |
  | (man twpolicy) |   /etc/tripwire/twpol.txt --> /etc/tripwire/tw.pol  |
  +----------------+-----------------------------------------------------+
  | Database       | Generierte Datenbank                                |
  | (man tripwire) |   /var/lib/tripwire/HOST.twd                        |
  +----------------+-----------------------------------------------------+
  | Report         | Generierte Berichte                                 |
  | (man twprint)  |   /var/lib/tripwire/report/HOST-DATE.twr            |
  +----------------+-----------------------------------------------------+
  | Backup         | Backup neu generierter Dateien                      |
  | (man twfiles)  |   Endung ".bak"                                     |
```

```
  +---------------+-----------------------------------------------------+

Datei-Extensions:

  +------+-------------------+
  | Ext  | Bedeutung         |
  +------+-------------------+
  | .bak | Backup            |
  | .cfg | Configuration     |
  | .key | SchlÃ¼ssel         |
  | .pol | Policy            |
  | .twd | Tripwire Database |
  | .twr | Tripwire Report   |
  +------+-------------------+

WÃ¤hrend der Installation von Tripwire entsteht ein Konfigurations-Verzeichnis

  /etc/tripwire

mit den beiden Konfigurationsdateien

  twcfg.txt   # Grundkonfiguration (Ort von Verz. + Dateien)
  twpol.txt   # Richtlinien (Systemdaten + Eigenschaften)

Die Datei "twcfg.txt" muss normalerweise nicht geÃ¤ndert werden, sie enthÃ¤lt die
Pfade fÃ¼r die restlichen von Tripwire benÃ¶tigten oder angelegten Dateien. Die
folgenden fÃ¼nf Angaben sind notwendig:

  +-------------------------------------------------------------------+
  | POLFILE      = /etc/tripwire/tw.pol                               |
  | DBFILE       = /var/lib/tripwire/$(HOSTNAME).twd                  |
  | REPORTFILE   = /var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr   |
  | SITEKEYFILE  = /etc/tripwire/site.key                             |
  | LOCALKEYFILE = /etc/tripwire/$(HOSTNAME)-local.key                |
  +-------------------------------------------------------------------+

Die folgenden Angaben dÃ¼rfen fehlen (Standardwerte angegeben):

  +-------------------------------------------------------------------+
  | ROOT                    = /usr/tripwire                           |
  | EDITOR                  = /bin/vi                                 | # /usr/bin/vim
  | TEMPDIRECTORY           = /tmp                                    |
  +-------------------------------------------------------------------+
  | LATEPROMPTING           = false                                   |
  | LOOSEDIRECTORYCHECKING  = false                                   |
  | SYSLOGREPORTING         = true                                    |
  | REPORTLEVEL             = 3                                       | # 0..4
  +-------------------------------------------------------------------+
  | GLOBALEMAIL             = none                                    | # tb@ostc.de
  | MAILMETHOD              = SENDMAIL                                | # SMTP
  | SMTPHOST                = mail.domain.com                         | # localhost
  | SMTPPORT                = 25                                      |
  | MAILPROGRAM             = /usr/lib/sendmail -oi -t                |
  | EMAILREPORTLEVEL        = 3                                       | # 0..4
  | MAILNOVIOLATIONS        = true                                    |
  +-------------------------------------------------------------------+


2.4) Richtlinien (Policies) definieren
--------------------------------------
Kommentare werden durch "#" eingeleitet und gelten bis zum Zeilenende.
Sie kÃ¶nnen auf einer Zeile fÃ¼r sich stehen oder nach einer Richtlinie.

  # Kommentar
  ... # Kommentar

Richtlinien haben folgende Form (ihre Reihenfolge ist nicht relevant):

  OBJEKT -> MASKE;   # Strichpunkt NICHT vergessen!

Soll eine Datei oder ein Verzeichnis NICHT geprÃ¼ft werden, ist eine
"Stop-Regel" anzugeben:

  !OBJEKT;   # Strichpunkt NICHT vergessen!

Als OBJEKT wird eine zu prÃ¼fende Datei oder ein Verzeichnis aufgelistet, pro
Objekt darf NUR EINE Richtlinie vorgegeben werden (wird geprÃ¼ft). Wird ein
Verzeichnis angegeben, dann werden alle Objekte ab diesem Startpunkt geprÃ¼ft.
Objekte mÃ¼ssen ABSOLUTE PFADE zu Dateien und Verzeichnissen sein, z.B.:

  /etc
  /usr/bin
```

```
  /sbin/init

In MASKE stehen die weiter oben aufgelisteten Kennbuchstaben (Codes) fÃ¼r
Eigenschaften und PrÃ¼fsummen-Typ. HÃ¤ufig benÃ¶tigte Kombinationen sind Ã¼ber
vordefinierte Variable verfÃ¼gbar:

  +------------+---------------------+--------------------------------------+
  | Name       | Definition          | Bedeutung                            |
  +------------+---------------------+--------------------------------------+
  | ReadOnly   | +pinugtsdbmCM-rlacSH | Nur lesbar                          |
  | Dynamic    | +pinugtd-srlbamcCMSH | Dynamik erlaubt (Wachsen+Schrumpfen) |
  | Growing    | +pinugtdl-srbamcCMSH | Darf nur wachsen                    |
  | Device     | +pugsdr-intlbamcCMSH | GerÃ¤t (von Tripwire nicht zu Ã¶ffnen) |
  | IgnoreAll  | -pinugtsdrlbamcCMSH  | Nur Vorhandensein/Abwesenheit prÃ¼fen |
  | IgnoreNone | +pinugtsdrbamcCMSH-l | Startpunkt fÃ¼r eigene Regeln        |
  +------------+---------------------+--------------------------------------+

Eigene Variablen fÃ¼r Objekte und Eigenschaften kÃ¶nnen jederzeit definiert und
anschlieÃM-^_end benutzt werden (GROSS/kleinschreibung der Variablennamen zÃ¤hlt,
der Name darf Buchstabe, Ziffern, Unterstrich und "+-@:." enthalten):

  VAR = WERT;         # Definition (Strichpunkt NICHT vergessen!)
  ... $(VAR) ...      # Verwendung ("$" und "(...)" NICHT vergessen!)

Vordefiniert sind bereits folgende Variablen:

  HOSTNAME   # Rechnername
  DATE       # Datum + Uhrzeit in der Form YYYYmmdd-HHMMSS

Beispiele fÃ¼r Variablendefinition:

  param1 = +CMSH;           # Variable "param1" definieren (Eigenschaft)
  dir1   = /etc/inet;       # 1. Variable "dir1" definieren (Objekt)
  DIR1   = /etc/init.d;     # 2. Variable "DIR1" definieren (Objekt)

Beispiele fÃ¼r Variablenverwendung:

  $(dir1)   -> +tbamc;        # Links Variable (Objekt)
  /etc/inet -> $(param1);     # Rechts Variable (Eigenschaft)
  $(DIR1)   -> $(param1);     # Links und rechts Variable (Objekt + Eigenschaft)

Beispiele fÃ¼r Richtlinien (Strichpunkt nicht vergessen!):

  +-------------------------------------------------------------------+
  | # Ganzen Dateibaum "/bin" prÃ¼fen (Vorhandensein + Readonly + ...) |
  | /bin                -> $(ReadOnly);                               |
  |                                                                   |
  | # Ganzen Dateibaum "/etc" prÃ¼fen (nur Vorhandensein!)             |
  | # AUSNAHMEN folgen noch                                           |
  | /etc                -> $(IgnoreAll);                              |
  |                                                                   |
  | # AUSNAHME: Datei "/etc/hostname.hme0" vollstÃ¤ndig prÃ¼fen        |
  | # (ausser access date und major/minor-device number)              |
  | /etc/hostname.hme0 -> $(IgnoreNone) -ar;                          |
  |                                                                   |
  | # AUSNAHME: Datei "/etc/passwd" genauer prÃ¼fen                   |
  | /etc/passwd         -> $(Growing);                                |
  |                                                                   |
  | # AUSNAHME: Dateibaum "/etc/init.d" + "/etc/rc.d" und             |
  | #           Datei "/etc/mnttab" NICHT prÃ¼fen                      |
  | !/etc/init.d;                                                     |
  | !/etc/rc.d;                                                       |
  | !/etc/mnttab;                                                     |
  +-------------------------------------------------------------------+

Richtlinien kÃ¶nnen auf 2 Arten zusÃ¤tzlich mit "Attributen" versehen werden, die
Name, Wichtigkeit, Verzeichnistiefe und MailempfÃ¤nger festlegen.

a) FÃ¼r einzelne Richtlinie (Strichpunkt NICHT vergessen!):

    OBJEKT -> MASKE (ATTRIBUT = WERT, ...);

b) FÃ¼r Gruppe von Richtlinien (Strichpunkt NICHT vergessen!):

    (ATTRIBUT = WERT, ...) {
        OBJEKT -> MASKE;
        OBJEKT -> MASKE;
        ...
    }

Folgende Attributangaben zu einer Richtinie sind mÃ¶glich:
```

Attribut	Bedeutung
emailto	eMail-EmpfÃ¤nger bei verletzter Regel
rulename	Zur Bericht-Auswahl + im Bericht-Output (Benennung von Regeln)
severity	Zur Bericht-Auswahl 0..1000000 (Default 0) (mind. notwendige Wichtigkeit)
recurse	Bis zu welcher Verzeichnistiefe untersuchen true (-1) = Komplett (Default) false (0) = Nur Verzeichniseintrag selbst 1..1000000 = Rekursionstiefe

Beispiele fÃ¼r Richtlinien mit Attributen:

```
  /usr/lib -> $(ReadOnly) (emailto = admin@foo.com, severity = 80);

  (emailto = tripwire@foo.com, severity = 100)
  {
      /bin     ->  $(ReadOnly);
      /sbin    ->  $(ReadOnly);
      /lib     ->  $(ReadOnly);
      /usr/bin ->  $(ReadOnly);
      /usr/sbin->  $(ReadOnly);
  }
```

Musterbeispiele fÃ¼r Richtlinien-Dateien findet man hier:

```
  /usr/share/doc/tripwire/examples/policyguide.txt.gz
  /usr/share/doc/tripwire/examples/twpol.txt.gz
```

2.5) Direktiven

Direktiven erlauben eine bedingte Interpretation der Policy-Datei und einige Diagnose- und Debugging-Operationen, sie haben folgende allgemeine Form

```
  @@DIRNAME [ARGS]    # Allgemeine Syntax
```

Folgende Directiven sind erlaubt:

Direktive	Bedeutung
@@ifhost HOST...	Bedingte Interpretation abhÃ¤ngig vom Host (|| = oder)
@@else	"
@@endif	"
@@section NAME	Abschnitt der Policy-Datei (FS, GLOBAL, NTFS, NTREG)
@@print "STRING"	Text auf Standard-Ausgabe ausgeben
@@error "STRING"	Text auf Standard-Ausgabe ausgeben und Abbruch
@@end	Logisches Dateiende (Rest der Datei wird ignoriert)

Hinweise: In einer @@ifhost-Direktive darf eine Liste von Hostnamen getrennt durch "||" angegeben werden. @@ifhost-Direktiven dÃ¼rfen verschachtelt werden. "FS" ist Standard bei "@@section". "NTFS" und "NTREG" haben nur Bedeutung auf Windows NT Systemen, auf UNIX-Systemen werden bei diesen beiden Angabeng alle folgenden Zeilen bis zur nÃ¤chsten @@section-Direktive Ã¼bersprungen.

Beispiele:

```
  @@section GLOBAL               # Abschnitt
                                 #
  @@ifhost plymouth              #
      /bin -> $(ReadOnly);       # Regel nur fÃ¼r Host "plymouth" geprÃ¼ft
  @@endif                        #
                                 #
  @@section FS                   # Abschnitt
                                 #
  @@ifhost exeter || charlton    #
      /usr/bin -> +pinug;        # Regel nur fÃ¼r Hosts exeter + charlton gÃ¼ltig
  @@else                         #
      /usr/bin -> +pinugsmC;     # Regel fÃ¼r alle anderen Hosts gÃ¼ltig
  @@endif                        #
                                 #
  @@print "Ok"                   # Ausgabe
  @@error "Fehler"               # Ausgabe + Abbruch
                                 #
  @@end                          # Rest der Datei wird ignoriert
```

```
3) DurchfÃ¼hrung einer Konfiguration und mehrerer PrÃ¼fungen
-----------------------------------------------------------
Hier nun der konkrete Ablauf zur Absicherung und PrÃ¼fung eines Systems gemÃ¤ÃM-^_
den Schritten 0-11 in der Tabelle von Abschnitt 2) (zur Steigerung der
Geschwindigkeit und Vermeidung von Tippfehlern die Befehle mit der Maus
kopieren!):

0. Rechnernamen ermitteln:

     echo $HOSTNAME           # --> z.B. "HOST"
     hostname                 # --> z.B. "HOST"

1. Lokalen Key erzeugen (Local Passphrase eingeben und merken!) und anzeigen
   (BinÃ¤rformat):

     twadmin --generate-keys -L /etc/tripwire/HOST-local.key
     ls -l /etc/tripwire/HOST-local.key

2. Site Key erzeugen (Site Passphrase eingeben und merken!) und anzeigen
   (BinÃ¤rformat):

     twadmin --generate-keys -S /etc/tripwire/site.key
     ls -l /etc/tripwire/site.key

3. Konfigurations-Datei anpassen:

     vi /etc/tripwire/twcfg.txt   # Normalerweise NICHT notwendig!

4. Konfigurations-Datei in binÃ¤re signierte Form umwandeln (Passphrase des
   Site Keys eingeben!) und anzeigen (signiertes BinÃ¤rformat):

     twadmin --create-cfgfile -S /etc/tripwire/site.key /etc/tripwire/twcfg.txt
     ls -l /etc/tripwire/tw.cfg

5. Richtliniendatei erstellen (vorher aus der mitgelieferten Musterdatei
   kopieren):

     cp /usr/share/doc/packages/tripwire/twpol.txt /etc/tripwire/twpol.txt
     vi /etc/tripwire/twpol.txt

   Einfaches Beispiel fÃ¼r "twpol.txt" (eigenes Heimatverzeichnis absichern):

     +----------------------------------------+
     | /home/tom -> $(IgnoreNone);            | # Strichpunkt NICHT vergessen!
     |                                        |
     | (rulename = FirstRule, severity = 0) { |
     |    /home/tom/cat -> $(IgnoreNone);     | # Strichpunkt NICHT vergessen!
     | }                                      |
     |                                        |
     | !/home/tom/bin;                        | # Strichpunkt NICHT vergessen!
     +----------------------------------------+

6. Richtlinien-Datei in binÃ¤re signierte Form umwandeln (Passphrase des Site
   Key eingeben!) und anzeigen (signiertes BinÃ¤rformat):

     twadmin --create-polfile -S /etc/tripwire/site.key /etc/tripwire/twpol.txt
     ls -l /etc/tripwire/tw.pol

7. GemÃ¤ÃM-^_ Richtlinien-Datei aus dem Dateisystem eine binÃ¤re, signierte
   Tripwire-Datenbank mit den ausgewÃ¤hlten Eigenschaften der ausgewÃ¤hlten
   Verzeichnisse und Dateien generieren (Passphrase des Local Key eingeben)
   und anzeigen:

     tripwire --init
     ls -l /var/lib/tripwire/HOST.twd

8. Aktuellen Systemzustand gegen Tripwire-Datenbank prÃ¼fen (gibt ASCII-Report
   auf Bildschirm aus, erzeugt binÃ¤ren Report in "/var/lib/tripwire/report"):

     tripwire --check
     ls -l /var/lib/tripwire/report    # --> HOST-20050218-103957.twr

   Erzeugte binÃ¤re Reportdatei nochmal in ausfÃ¼hrlicher ASCII-Form ausgeben:

     twprint --print-report -r /var/lib/tripwire/report/HOST-20050218-103957.twr

   EinschrÃ¤nkungen der zu prÃ¼fenden Regeln anhand Regelname oder Severity:

     tripwire --check --rule-name "FirstRule"   # Nur mit diesem Namen
     tripwire --check --severity "100"          # Mind. diese Wichtigkeit
```

9a. Mehrere Ã‚M-^Dnderungen am Heimatverzeichnis vornehmen:

```
cd /home/tom
mkdir VERZ
touch VERZ/DATEI
touch .profile
echo "echo HiHi..." >> .bashrc
```

9b. Aktuellen Systemzustand nochmal gegen Tripwire-Datenbank prÃ¼fen (gibt ASCII-Report auf dem Bildschirm aus und erzeugt binÃ¤ren Report in "/var/lib/tripwire/report"):

```
tripwire --check
ls -l /var/lib/tripwire/report    # --> HOST-20050218-143957.twr
```

Erzeugte binÃ¤re Reportdatei nochmal in ausfÃ¼hrlicher ASCII-Form ausgeben:

```
twprint --print-report -r /var/lib/tripwire/report/HOST-20050218-143957.twr
```

9c. Anhand eines erzeugten Berichts (als Parameter anzugeben) die Tripwire-Datenbank updaten. Dazu wird der Bericht mit "[x]" vor jeder Ã‚M-^Dnderung im Editor "vim" angezeigt (Pfad zum Editor vorher in Variable "EDITOR/VISUAL" bekannt geben). Durch LÃ¶schen des "x" wird die Ã‚M-^Dnderung nicht in der Datenbank eingetragen ("[X/x]" => "[]/[ ]"):

```
export VISUAL=/bin/vim
tripwire --update -r /var/lib/tripwire/report/HOST-20050203-143957.twr
```

9d. Aktuellen Systemzustand nochmal gegen Tripwire-Datenbank prÃ¼fen (gibt ASCII-Report auf dem Bildschirm aus und erzeugt binÃ¤ren Report in "/var/lib/tripwire/report"):

```
tripwire --check
ls -l /var/lib/tripwire/report    # --> HOST-20050218-143957.twr
```

Die akzeptierten Ã‚M-^Dnderungen sollten nun nicht mehr berichtet werden.

10. TÃ¤gliche LÃ¤ufe um 02:22 des Tripwire-Reporting als "cronjob" einrichten und den Bericht per Mail an USER@HOST versenden:

```
crontab -e
+-------------------------------------------------------------------------+
| # MIN STD TAG MON WOTAG KMDO                                            |
| 22 2 * * * /usr/sbin/tripwire --check 2>&1 | mail -s "Tripwire" USER@HOST |
+-------------------------------------------------------------------------+
```

11. Modifikationen der Richtlinien mÃ¼ssen in der ASCII-Richtliniendatei vorgenommen werden. Dann ist die Tripwire-Datenbank anzupassen und die binÃ¤re Richtlinien-Datei neu zu erzeugen ("low/high"):

```
vi /etc/tripwire/twpol.txt
tripwire --update-policy --secure-mode low /etc/tripwire/twpol.txt
```

Dabei werden Verletzungen der Richtlinien berichtet und AM SCHLUSS wird automatisch die BinÃ¤rform der Richtlinien-Datei erzeugt!
(letzteres wird beim Standardwert "--secure-mode high" nicht gemacht!).

4) man-Pages

man-Page	Inhalt
tripwire(8)	Beschreibung tripwire-Kommando
twadmin(8)	Beschreibung twadmin-Kommando
twprint(8)	Beschreibung twprint-Kommando
siggen(8)	Beschreibung siggen-Kommando
twintro(5)	EinfÃ¼hrung in Tripwire
twfiles(5)	Beschreibung der Tripwire-Dateien
twconfig(5)	Beschreibung /etc/tripwire/twcfg.txt
twpolicy(5)	Beschreibung /etc/tripwire/twpol.txt

* Das Kommando "siggen" zeigt die tripwire-Hashwerte (CRC-32, MD5, SHA, Haval) fÃ¼r eine oder mehrere Dateien an.

5) Literaturverweise

http://sourceforge.net/projects/tripwire/	Tripwire-Projekt
http://www.tripwire.org/	Tripwire-Projekt
http://www.tripwire.com/	Tripwire (kommerzielle Version)