

# SelfLinux-0.13.1



## Samhain



Autor: Rainer Wichman ([rwichmann@hs.uni-hamburg.de](mailto:rwichmann@hs.uni-hamburg.de))  
Formatierung: Florian Frank ([florian@pingos.org](mailto:florian@pingos.org))  
Formatierung: Alexander Fischer ([alexander.fischer@selflinux.org](mailto:alexander.fischer@selflinux.org))  
Lizenz: GFDL

Samhain ist ein Werkzeug zur Überprüfung der Integrität eines Rechners bzw. des Betriebssystems des Rechners und/oder der darauf gespeicherten Daten.

# Inhaltsverzeichnis

## 1 Einleitung

- 1.1 Unterschied zu anderen Einbruchserkennungssystemen (IDS)

## 2 Installation, Konfiguration und Initialisierung

- 2.1 Installation
- 2.2 Vertrauenswürdige Benutzer
- 2.3 Konfiguration
- 2.4 Initialisierung

## 3 Allgemeines zur Benutzung

- 3.1 Die Datenbank
- 3.2 Verifikation der System-Integrität
- 3.3 Pflege der Datenbank
- 3.4 Inhalt der Datenbank ansehen
- 3.5 Verifikation der Email-Nachrichten von Samhain
- 3.6 Verifikation der lokalen Log-Datei
- 3.7 Verifikation des Samhain-Programmes

## 4 Konfiguration der Logging-Eigenschaften

- 4.1 Unterstützte Log-Möglichkeiten
- 4.2 Filtern von Log-Berichten
  - 4.2.1 Levels (Dringlichkeitsstufen)
  - 4.2.2 Konfigurierbare Levels (Dringlichkeitsstufen)
  - 4.2.3 Ereignisklassen
- 4.3 Ausschalten von Log-Möglichkeiten
- 4.4 Spezielle Optionen
  - 4.4.1 Email
  - 4.4.2 Datenbank

## 5 Konfiguration der Integritätseigenschaften innerhalb des Dateisystems

- 5.1 Vordefinierte Policys
- 5.2 Wahl einer Policy
- 5.3 Rekursionstiefe
- 5.4 Unterverzeichnisse ignorieren
- 5.5 Keine Benachrichtigung über neue/gelöschte Dateien
- 5.6 Hardlink Test
- 5.7 Test auf seltsame Dateinamen
- 5.8 Änderung einer Policy
- 5.9 Zeiten der Integritätsprüfung

## 6 Logging zu einem Log-Server (Yule)

- 6.1 Übersetzen des Log-Servers (Yule)
- 6.2 Übersetzen von Samhain
- 6.3 Authentifizierung gegenüber Yule
- 6.4 Datenbank- und Konfigurationsdatei auf dem Log-Server
- 6.5 Besonderheiten beim Logging

## **7 Weitere Eigenschaften von Samhain**

- 7.1 Signierte Datenbank- und Konfigurationsdateien
- 7.2 Stealth (versteckter Modus)
- 7.3 Überwachung des Kernels
- 7.4 Überwachung von Login/Logout-Vorgängen
- 7.5 Suche nach SUID/SGID-Dateien

## **8 Optionen in der Konfigurationsdatei**

- 8.1 Bedingte Anweisungen
- 8.2 Überwachte Dateien
- 8.3 Dringlichkeitsstufe von Ereignissen
- 8.4 Filter für Log-Möglichkeiten
- 8.5 Überwachung von Login/Logout-Ereignissen
- 8.6 Überprüfung des Kernels
- 8.7 Suchen nach SUID/SGID-Dateien
- 8.8 Logging zu einer relationalen Datenbank
- 8.9 Verschiedenes
- 8.10 Externe Skripte
- 8.11 Clients

# 1 Einleitung

Ist ein Computer an das Internet angeschlossen, so besteht zwangsläufig die Gefahr, dass Fremde versuchen, über bekannte Schwachstellen in den Rechner einzubrechen, um ihn für eigene, meist illegale Zwecke zu missbrauchen. In der Regel wird ein solcher Einbrecher versuchen, ein so genanntes Rootkit zu installieren, um die dauerhafte Kontrolle der Computers zu sichern und seine Anwesenheit zu verbergen. Andererseits sind die Veränderungen, die ein Einbrecher verursacht, eine hervorragende Möglichkeit, ihn schnellstmöglichst zu entdecken.

**Samhain** ist ein Werkzeug zur Überprüfung der Integrität eines Rechners bzw. des Betriebssystems des Rechners und/oder der darauf gespeicherten Daten. Die Grundfunktion von Samhain besteht darin, eine Datenbank mit Prüfsummen und anderen Eigenschaften aller zu überwachenden Dateien zu erstellen, um anschließend regelmäßig zu überprüfen, ob eine dieser Dateien manipuliert wurde.

Zusätzlich besteht die Möglichkeit, die Integrität des Kernels zu überwachen, d. h. zu testen, ob der Kernel zur Laufzeit manipuliert wurde, Login/Logout-Vorgänge zu melden, und/oder das gesamte Dateisystem regelmäßig nach SUID-Programmen zu durchsuchen.

## 1.1 Unterschied zu anderen Einbruchserkennungssystemen (IDS)

Bei Systemen zur Entdeckung von Einbrechern oder Einbruchversuchen ([Intrusion Detection Systems](#)) unterscheidet man generell Netzwerk-basierte *IDS* (*NIDS*), die an einer zentralen Stelle den Netzwerkverkehr überwachen, und Host-basierte *IDS*, die den lokalen Computer überwachen.

Der Vorteil eines *NIDS* besteht offensichtlich darin, dass man nur ein Überwachungssystem an einer zentralen Stelle installieren muss. Andererseits kann ein solches System nur Angriffsversuche erkennen, deren Muster bereits bekannt sind; und nach der Entdeckung eines Angriffsversuches ist nicht unbedingt klar, ob er nun erfolgreich war oder nicht. Ein praktisches Problem besteht auch darin, Fehlalarme zu reduzieren, ohne dabei echte Angriffsversuche zu übersehen.

Eine Host-basierte Anwendung zur Verifikation der Systemintegrität wie **Samhain** ist natürlich aufwendiger zu installieren, wenn man mehrere Computer überwachen möchte. Andererseits kann ein solches System auch Angriffe mit neuartigen, bisher unbekannt Methoden erkennen. Die Problematik von Fehlalarmen ist deutlich reduziert, da das System nur Alarm schlägt, wenn tatsächlich Veränderungen an einem Computer stattgefunden haben.

Besonders in großen Firmennetzen besteht ein großes Problem auch in Angriffen von innerhalb; z. B. von mitgebrachten Laptops aus, oder mit Hilfe von Passwörtern, die sich ein Angreifer z. B. durch geschickt geführte Telefongespräche verschafft hat. Auch bei solchen Angriffen kann ein Host-basiertes System wie z. B. **Samhain** eine Manipulation des Systems erkennen.

## 2 Installation, Konfiguration und Initialisierung

Samhain wird als Quellcode unter der GNU Public License frei angeboten, und die jeweils aktuellste Version ist erhältlich unter der URL  <http://www.la-samhna.de/samhain/samhain-current.tar.gz>

Zum Übersetzen der Quellen ist ein *ANSI C Compiler* wie z. B. *GNU gcc* erforderlich. Einige optionale Funktionen erfordern weitere Software (siehe Abschnitte [▶ Unterstützte Log-Möglichkeiten](#) und [▶ Weitere Eigenschaften von Samhain](#)). Insbesondere setzt die Verwendung von signierten Konfigurations- und Datenbank-Dateien die Installation von *GnuPG* (*GNU Privacy Guard*) voraus (siehe Abschnitt [▶ Signierte Datenbank- und Konfigurationsdateien](#)).

### 2.1 Installation

Nach dem Herunterladen kann man das tar-Paket auspacken, worauf man zwei Dateien erhält: zum einen ein weiteres tar-Paket mit dem eigentlichen Quellcode, und zum anderen dessen PGP-Signatur.

```
user@linux ~/ $ tar xzvf samhain-current.tar.gz
samhain-X.Y.Z.tar.gz
samhain-X.Y.Z.tar.gz.asc
```

Nun kann man (optional) die Signatur überprüfen - alle Versionen von Samhain sind mit dem Schlüssel *0x0F571F6C* (*Rainer Wichmann*) signiert:

```
user@linux ~/ $ gpg --verify samhain-X.Y.Z.tar.gz.asc
samhain-X.Y.Z.tar.gz

gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
gpg: Signature made Mon Nov  8 07:38:21 2004 CET using DSA key ID
0F571F6C
gpg: Good signature from "Rainer Wichmann <rwichmann@la-samhna.de>"
gpg:                aka "Rainer Wichmann <rwichmann@hs.uni-hamburg.de>"
```

Die Übersetzung der Quellen erfolgt dann nach dem *Standardverfahren*:

```
user@linux ~/ $ tar xzvf samhain-X.Y.Z.tar.gz
user@linux ~/ $ cd samhain-X.Y.Z/
user@linux ~/samhain-X.Y.Z/ $ ./configure && make
```

... ebenso wie die Installation:

```
user@linux ~/samhain-X.Y.Z/ $ su
Password:
geheimes_root_passwort
```

```
root@linux /home/user/samhain-X.Y.Z/ # make install
root@linux /home/user/samhain-X.Y.Z/ # make install-boot
```

## 2.2 Vertrauenswürdige Benutzer

Bei der Installation von Samhain ist zu berücksichtigen, dass Samhain voraussetzt, dass nur vertrauenswürdige Benutzer Schreibrechte auf den Pfaden zu Konfigurations-, Log-, und Datenbank-Dateien haben. Dies gilt auch für Mitglieder von Gruppen, soweit Gruppen-Schreibrechte bestehen.

Standardmäßig sind nur `root` und der Benutzer von `Samhain` vertrauenswürdige Benutzer lassen sich beim Übersetzen von `Samhain` definieren über die Option `--with-trusted=0,uid1,uid2,...`

## 2.3 Konfiguration

Die Konfigurationsdatei heißt standardmäßig `/etc/samhainrc`. Wer sie lieber unter `/usr/local/etc/` haben möchte, kann dies bei der Übersetzung der Quellen mit `./configure --prefix=/usr/local` erzwingen.

Bei der Installation wird eine Standard-Konfigurationsdatei installiert, die man in der Regel zunächst an die eigenen Bedürfnisse anpassen möchte.

Die Syntax der Konfigurationsdatei lehnt sich an an INI-Dateien: es gibt einzelne Sektionen, die jeweils mit einer Überschrift in eckigen Klammern beginnen, und die "Parameter=Wert"-Zeilen enthalten. Leere Zeilen sowie Zeilen, die mit `"#"` beginnen, werden ignoriert; in "Parameter=Wert"-Zuweisungen werden Leerzeichen vor "Parameter", vor und nach dem Gleichheitszeichen, sowie am Zeilenende ignoriert.

Die wesentlichen Eigenschaften, die konfiguriert werden sollten, sind:

1. das Logging (siehe Abschnitt [► Konfiguration der Logging-Eigenschaften](#)), d. h. wie ausführlich soll `Samhain` seine Aktivität protokollieren, und welche Methoden soll es benutzen ?, und
2. die Integritätseigenschaften innerhalb des Dateisystems (siehe Abschnitt [► Konfiguration der Integritätseigenschaften innerhalb des Dateisystems](#)), d. h. welche Dateien sollen überwacht werden, und welche Eigenschaften dieser Dateien sollen überprüft werden?

## 2.4 Initialisierung

Nach der Überprüfung und Anpassung der Konfiguration kann nun durch den Benutzer `root` die Datenbank initialisiert werden:

```
root@linux ~/ # samhain -t init
```

Je nach Umfang der zu überwachenden Dateien kann die Initialisierung mehrere Minuten dauern. Die erzeugte Datenbank wird standardmäßig nach `/var/lib/samhain/samhain_file` geschrieben.

Zu beachten ist dabei, dass grundsätzlich an das Dateiende angehängt wird, falls die Datei bereits vorhanden ist. Dieser Befehl ist also nicht dazu geeignet, eine bereits vorhandene Datenbank zu aktualisieren. Für letzteres sollte entweder der Befehl `samhain -t update` verwendet werden, oder die Datenbank zuvor verschoben werden.

## 3 Allgemeines zur Benutzung

### 3.1 Die Datenbank

Standardmäßig wird die Datenbank nach `/var/lib/samhain/samhain_file` geschrieben. Nach der Initialisierung der Datenbank ist in dieser der aktuelle Zustand der überwachten Dateien gespeichert: Prüfsummen des Dateiinhaltes, Dateigröße, Zugriffsrechte, Eigentümer und Gruppe, Zeiten, Anzahl von Hardlinks, die Nummer der Inode, und (bei Geräten) die Gerätenummer.

*Samhain* benutzt standardmäßig den TIGER192-Prüfsummenalgorithmus. Optional kann auch SHA-1 oder MD5 benutzt werden (MD5 wird wegen möglicher Schwächen nicht mehr empfohlen).

Diese Datenbank sollte natürlich vor unabsichtlichem oder böswilligem Überschreiben gesichert werden. Dazu bieten sich mehrere Möglichkeiten an:

1. die Datenbank kann auf einem nicht beschreibbaren Gerät gespeichert werden, z. B. ein USB-Stick, der auf **nicht beschreibbar** geschaltet ist,
2. die Datenbank kann mit **GnuPG** signiert werden, wie [weiter unten](#) beschrieben wird - dies schützt zwar nicht davor, dass sie überschrieben wird, ermöglicht aber, eine solche Manipulation zu erkennen -, oder
3. die Datenbank kann auf einem anderen Rechner gespeichert werden, und *Samhain* kann so konfiguriert werden, dass die Datenbank beim Start von diesem Rechner heruntergeladen wird (siehe Abschnitt [Logging zu einem Log-Server \(Yule\)](#)).

### 3.2 Verifikation der System-Integrität

Die Verifikation der System-Integrität erfolgt durch Vergleich der überwachten Dateien mit den Angaben, die in der Datenbank gespeichert sind. Manuell (oder per **cron**) kann dies erfolgen mit dem Kommando:

```
root@linux ~/ # samhain -t check
```

Eine wesentliche Eigenschaft von *Samhain* besteht jedoch darin, dass es zum Einsatz als Dämon entwickelt wurde. D. h. *Samhain* startet automatisch beim Booten des Rechners, überwacht das Dateisystem, und meldet Veränderungen entsprechen den eingestellten Logging-Eigenschaften (siehe Abschnitt [Konfiguration der Logging-Eigenschaften](#)). Bei der Installation wird mit dem Kommando:

```
root@linux ~/ # make install-boot
```

dafür gesorgt, dass *Samhain* beim nächsten Booten gestartet wird. Will man nach der Initialisierung der Datenbank den *Samhain-Daemon* gleich starten, so kann man dies mit folgendem Befehl tun:

```
root@linux ~/ # samhain start
```

### 3.3 Pflege der Datenbank

Natürlich gibt es in einem Dateisystem durchaus auch gewollte Veränderungen, z. B. infolge von mehr oder weniger regelmäßigen Sicherheits-Aktualisierungen. Nach solchen Veränderungen muss die Datenbank von *Samhain* jeweils auf aktuellen Stand gebracht werden. Dies erfolgt mit dem Kommando:

```
root@linux ~/ # samhain -t update [--interactive] [-l none] [-e none]
```

Die in Klammern gesetzten Optionen sind nicht notwendig, aber gegebenenfalls hilfreich:

1. Mit `--interactive` fragt Samhain bei jeder entdeckten Veränderung, ob der entsprechende Eintrag in der Datenbank geändert werden soll.
2. Falls das Update der Datenbank stattfindet, während der *Samhain-Daemon* läuft, sollte man darauf achten, Konflikte beim Logging zu vermeiden, indem man bestimmte Log-Optionen nicht benutzt. Dies gilt insbesondere für die lokale Log-Datei (ausschalten mit `-l none`) sowie für Logging zu einem Log-Server (siehe Abschnitt [Logging zu einem Log-Server \(Yule\)](#); ausschalten mit `-e none`).

### 3.4 Inhalt der Datenbank ansehen

Wenn man den Inhalt der Datenbank ansehen möchte, kann man folgenden Befehl benutzen:

```
root@linux ~/ # samhain [-a] -d /pfad/zur/datenbank
```

Dies liefert eine Ausgabe, die ähnlich ist wie `ls -l`. Mit der Option `-a` werden weitere Details angezeigt, die man bei `ls -l` nicht erhält (z. B. die Prüfsumme einer Datei).

### 3.5 Verifikation der Email-Nachrichten von Samhain

Wird Email als Log-Option benutzt, so ist es möglich, zu überprüfen, ob die erhaltenen Emails tatsächlich von Samhain stammen. Hierzu gibt es den das Kommando:

```
root@linux ~/ # samhain -M /pfad/zur/mailbox
```

Dabei gibt es folgende Einschränkungen:

1. alle Email sollten in einer Datei sein (z.B eine Mailbox im mbox-Format),
2. die jeweils erste Email nach dem Start von Samhain kann nicht verifiziert werden, und
3. das zur Verifikation benutzte Samhain-Programm muss den selben Schlüssel enthalten wie das sendende Samhain-Programm (siehe Abschnitt [Verifikation des Samhain-Programmes](#)).

### 3.6 Verifikation der lokalen Log-Datei

Wird die lokale Log-Datei benutzt, so ist es möglich, zu überprüfen, ob die Einträge tatsächlich von Samhain stammen, und nicht verändert wurden. Hierzu gibt es den das Kommando:

```
root@linux ~/ # samhain -M /pfad/zur/logdatei
```

Dabei gibt es folgende Einschränkungen:

1. es ist ein Schlüssel notwendig, der nach Eröffnung der Log-Datei per Email geschickt wird, falls (und nur falls) auch Email als Log-Option benutzt wird, und
2. das zur Verifikation benutzte Samhain-Programm muss den selben Schlüssel enthalten wie das sendende Samhain-Programm (siehe Abschnitt [Verifikation des Samhain-Programmes](#)).

### 3.7 Verifikation des Samhain-Programmes

Beim Übersetzen der Quellen wird standardmäßig ein zufällig erzeugter Schlüssel eingebettet. Der Wert dieses Schlüssels wird von `./configure` am Ende angezeigt (die Zahlen sind nur Beispiele):

```
Base key: 1076404394,1083932597
```

Ein Samhain-Programm mit identischem Schlüssel lässt sich erzeugen, indem man als Option für `./configure` angibt: `--enable-base=1076404394,1083932597`.

Da sich Emails (siehe Abschnitt ► [Verifikation der Email-Nachrichten von Samhain](#) - und lokale Log-Dateien (siehe Abschnitt ► [Verifikation der lokalen Log-Datei](#)) - nur verifizieren lassen, wenn verifizierendes und erzeugendes Programm den selben eingebetteten Schlüssel haben, ist es somit prinzipiell möglich, die Integrität eines Samhain-Programmes zu überprüfen, dass auf einem entfernten Rechner läuft, und per Email Log-Berichte schickt.

## 4 Konfiguration der Logging-Eigenschaften

Samhain verfügt über viele verschiedene Log-Möglichkeiten, die teilweise erst durch entsprechende Optionen beim Übersetzen der Quellen verfügbar werden.

Jede unterstützte Log-Option lässt sich über die Konfigurationsdatei gesondert konfigurieren und ein- oder ausschalten.

### 4.1 Unterstützte Log-Möglichkeiten

Samhain unterstützt standardmäßig *Email*, *Syslog*, eine Logdatei (`/var/log/samhain_log`), und die Konsole (bzw. `stderr`, wenn Samhain nicht als Daemon läuft). Weiterhin ist es möglich, Log-Nachrichten an ein externes Skript oder ein Programm zu übergeben, und auf diese Weise beliebige weitere Log-Möglichkeiten zu realisieren.

Weitere Möglichkeiten sind:

1. Logging zu einem Log-Server. Dies wird weiter unten in einem [▶ eigenen Abschnitt](#) näher beschrieben.
2. Logging zu einem Prelude IDS-System. Hierzu muss die `libprelude`-Bibliothek installiert sein, und die Option `--with-prelude` muss bei `./configure` angegeben werden, wenn die Quellen übersetzt werden. Der Sensor-Name, der beim Ausführen des Prelude-Programmes `sensor-adduser` (zum Anmelden eines Sensors) angegeben werden muss, lautet **Samhain**.
3. Logging zu einer relationalen Datenbank (MySQL, PostgreSQL, Oracle). Hierzu muss die entsprechende Client-Bibliothek der Datenbank installiert sein, und die Option `--with-database=mysql` (oder `postgresql`, `oracle`) muss bei `./configure` angegeben werden, wenn die Quellen übersetzt werden.

### 4.2 Filtern von Log-Berichten

Jede der unterstützten Log-Möglichkeiten lässt sich individuell einstellen. Einerseits ist es möglich, eine Untergrenze anzugeben für die Dringlichkeitsstufe / den Level der Nachrichten, die man erhalten möchte. Je niedriger man diese Untergrenze angibt, desto ausführlicher ist das Logging. Andererseits ist es möglich, das Logging auf bestimmte Klassen von Ereignissen zu beschränken.

#### 4.2.1 Levels (Dringlichkeitsstufen)

Die Reihenfolge der Levels ist:

<code>debug</code>	nur zur Fehlersuche
<code>info</code>	detaillierte Informationen
<code>notice</code>	Informationen
<code>warn</code>	Warnungen
<code>mark</code>	Zeitmarken
<code>err</code>	Fehler
<code>crit</code>	kritische Probleme
<code>alert</code>	Start/Stop des Programmes; Fehler, die zum Programmabbruch führen

Der gewünschte Level wird eingestellt in der **[Log]**-Sektion der Konfigurationsdatei mit den Optionen der Form `xxxSeverity = level`:

/etc/samhainrc

```
[Log]
# E-Mail
MailSeverity = crit
# Konsole
PrintSeverity = info
# Syslog
SyslogSeverity = err
# Logdatei
LogSeverity = err
#
# Log Server (optional)
# ExportSeverity = err
#
# Datenbank
# DatabaseSeverity = err
#
# Prelude
# PreludeSeverity = crit
```

#### 4.2.2 Konfigurierbare Levels (Dringlichkeitsstufen)

Der Level einer Nachricht ist im Allgemeinen festgelegt. Die folgenden Nachrichten haben konfigurierbare Levels: Änderungen bei überwachten Dateien, Zugriffsfehler bei Dateien und Directories, Dateien mit ungültigen UIDs/GIDs oder seltsamen Dateinamen (nicht-druckbare Zeichen). Diese Levels werden konfiguriert in der [EventSeverity]-Sektion der Konfigurationsdatei:

/etc/samhainrc

```
[EventSeverity]
#
# Überwachte Dateien (für die Bedeutung von 'ReadOnly',
# 'LogFiles', usw. siehe Abschnitt Konfiguration der Integritätseigenschaften
# innerhalb des Dateisystems).
#
SeverityReadOnly=crit
SeverityLogFiles=crit
SeverityGrowingLogs=warn
SeverityIgnoreNone=crit
SeverityIgnoreAll=info
#
# Zugriffsfehler für Dateien (Files) und Directories (Dirs)
#
SeverityFiles=err
SeverityDirs=err
#
# Dateinamen, ungültige UIDs/GIDs
#
SeverityNames=info
```

#### 4.2.3 Ereignisklassen

Die folgenden Ereignisklassen sind definiert:

EVENT	Überwachte Ereignisse (z.B. Änderungen einer Datei)
START	Start/Stop-Meldungen

---

STAMP	Zeitmarken
LOGKEY	Der Schlüssel zur Verifikation der Log-Datei (siehe Abschnitt <a href="#">▶ Verifikation der lokalen Log-Datei</a> )
ERROR	Fehlermeldungen
OTHER	Alles andere (z.B. detaillierte Informationen)
AUD	System-Aufrufe (für Fehlersuche)

Standardmäßig sind alle Ereignisklassen für alle Log-Möglichkeiten zugelassen. Will man dies ändern, so geschieht dies in der **[Log]**-Sektion der Konfigurationsdatei mit Optionen der Form `xxxClass = Liste` der zugelassen Ereignisklassen, wobei in der Liste Leerzeichen zur Separation benutzt werden sollten, z. B.:

```
/etc/samhainrc  
  
[Log]  
# E-Mail  
MailClass = EVENTS ERROR LOGKEY  
# Konsole  
PrintClass = EVENTS ERROR STAMP  
# Syslog  
SyslogClass = START  
# Logdatei  
LogClass = EVENTS START STAMP ERROR OTHER
```

### 4.3 Ausschalten von Log-Möglichkeiten

Möchte man eine unterstützte Log-Möglichkeit nicht benutzen, so kann man sie ausschalten, indem man den Level auf den speziellen Wert `none` (nichts) setzt, z. B. `MailSeverity = none`.

### 4.4 Spezielle Optionen

Einige Log-Möglichkeiten erfordern spezielle Optionen, die im Folgenden beschrieben werden.

#### 4.4.1 Email

Spezielle Optionen für Email werden in der **[Misc]**-Sektion der Konfigurationsdatei gesetzt:

---

/etc/samhainrc

```
[Misc]
#
# Für Email ist die Angabe der Empfängeradresse notwendig mit:
SetMailAddress = xyz@example.com
#
# Ferner kann es notwendig sein, ein Relay anzugeben:
SetMailRelay = relay.example.com
#
# Nachrichten werden i.A. zunächst in eine Warteschlange gestellt, und
# dann in einer Email versendet. Die maximale Anzahl von Nachrichten
# in der Warteschlange ist konfigurierbar (Obergrenze: 128):
SetMailNum = 10
#
# Auch die maximale Zeitspanne (in Sekunden) bis zum Versenden der
# Nachrichten in der Warteschlange ist konfigurierbar:
SetMailTime = 86400
#
# Schließlich ist es möglich, die Subjekt-Zeile der Email zu bestimmen:
MailSubject = xyz
```

#### 4.4.2 Datenbank

Für den Zugriff auf eine Datenbank ist es notwendig, die relevanten Informationen anzugeben. Dies geschieht in der **[Database]**-Sektion der Konfigurationsdatei:

---

/etc/samhainrc

```
[Database]
#
# Zunächst wird der Name der Datenbank benötigt (standardmäßig 'samhain')
SetDBName = samhain
#
# Außerdem ist der Name der Log-Tabelle anzugeben:
SetDBTable = log
#
# Auf welchem Rechner läuft der Datenbank-Server ?
SetDBHost = localhost
#
# Für den Zugriff auf die Datenbank sind Benutzername und Passwort notwendig:
#
# Benutzername
SetDBUser = samhain
#
# Passwort (kein voreingestellter Standardwert)
SetDBPassword = ...
#
# Soll der Log-Server eigene Zeitstempel für erhaltene Nachrichten loggen ?
SetDBServerTstamp = True
#
# Soll eine ständige Verbindung zur Datenbank aufrecht erhalten werden ?
UsePersistent = True
```

---

## 5 Konfiguration der Integritätseigenschaften innerhalb des Dateisystems

In Abschnitt 4 wurde besprochen, wie Samhain berichten soll. Natürlich muss man auch festlegen, worüber Samhain berichten soll. D. h., es muss angegeben werden, über welche Dateien-Änderungen man informiert werden möchte.

Samhain speichert die Prüfsumme des Dateiinhaltes (bei regulären Dateien), die Dateigröße, Zugriffsrechte, Besitzer und Gruppe, Zeiten, Anzahl von Hardlinks, die Nummer der Inode, und (bei Geräten) die Gerätenummer.

Nicht immer ist es sinnvoll, alle diese Eigenschaften zu prüfen. Z.B. ändern sich Größe, Prüfsumme und Zeitstempel von Logdateien fortwährend. Auch Größe und Zeitstempel einer Verzeichnis-Inode ändern sich, wenn Dateien in diesem Verzeichnis erzeugt und/oder gelöscht werden. Man beachte dabei den Unterschied zwischen den Dateien in einem Verzeichnis, und der Verzeichnis-Inode selbst, d. h. der speziellen Datei, die eine Liste der ersten Dateien enthält. In der Regel ist es sinnvoll, manche Dateien nach anderen Policies (Grundsätzen) zu behandeln als andere.

Samhain bietet mehrere vordefinierte Policies, sowie die Möglichkeit, diese Policies nach eigenem Ermessen zu ändern. Eine Policy ist dabei ein Grundsatz, der besagt, welche Dateieigenschaften sich ändern dürfen, und welche nicht.

### 5.1 Vordefinierte Policies

Die folgenden Policies sind vordefiniert:

ReadOnly	Nur die Zeit des letzten Zugriffs (access time) darf sich ändern.
Prelink	Wie ReadOnly, aber für Programme und Bibliotheken, die mit Hilfe von <code>prelink</code> verändert wurden, um einen schnelleren Programmstart zu ermöglichen.
LogFiles	Alle Zeitstempel, sowie Prüfsumme und Dateigröße dürfen sich ändern.
GrowingLogFiles	Sowohl Zeitstempel als auch Prüfsumme dürfen sich ändern. Die Dateigröße darf sich nicht verringern, aber erhöhen.
Attributes	Nur Veränderungen von Eigentümer/Gruppe oder Zugriffsrechten werden berichtet.
IgnoreAll	Alle Veränderungen werden ignoriert, aber die Existenz einer Datei wird geprüft.
IgnoreNone	Alle Veränderungen einschließlich der Zugriffszeit werden berichtet. Praktisch bedeutet dies allerdings, dass nun die Zeit der letzten Inode-Änderung anstelle der Zeit des letzten Zugriffs ignoriert wird.
User0, User1	Diese Policies sind standardmäßig auf <b>nichts darf sich ändern</b> gesetzt und für benutzerdefinierte Policies gedacht.

### 5.2 Wahl einer Policy

Jede Policy entspricht einer Sektion in der Konfigurationsdatei, deren Titel der Name der Policy ist. Ein Verzeichnis oder eine einzelne Datei wird unter diese Policy gestellt, indem man in der entsprechenden Sektion einen Eintrag der Form `dir=/directory` bzw. `file=/file` macht. Dabei sind immer absolute Pfade anzugeben; **Jokerzeichen** im Shell-Stil werden unterstützt:

/etc/samhainrc

```
[ReadOnly]
#
dir=/bin
dir=/usr/bin
dir=/sbin
dir=/usr/sbin
#
file=/etc/passwd*
file=/etc/shadow*
```

### 5.3 Rekursionstiefe

Standardmäßig werden Verzeichnisse nicht rekursiv geprüft. Dies kann global geändert werden mit dem Eintrag `SetRecursionLevel = zahl` in der `[Misc]`-Sektion, wobei **zahl** die Anzahl der Rekursionsebenen angibt (maximal 99).

Weiterhin ist es möglich, die Rekursionstiefe für ein Verzeichnis individuell zu konfigurieren, indem man die gewünschte Anzahl Rekursionsebenen dem Verzeichnis voranstellt:

/etc/samhainrc

```
[ReadOnly]
#
dir=5/var
```

### 5.4 Unterverzeichnisse ignorieren

Will man einen Verzeichnisbaum rekursiv prüfen, aber bestimmte Unterverzeichnisse (Teilbäume) komplett auslassen, so müssen diese Unterverzeichnisse mit einer Rekursionstiefe von **-1** unter die Policy `[IgnoreAll]` gestellt werden:

/etc/samhainrc

```
[IgnoreAll]
#
dir=-1/var/nicht/zuhause
```

### 5.5 Keine Benachrichtigung über neue/gelöschte Dateien

Gelegentlich möchte man es vermeiden, über bestimmte Dateien informiert zu werden, die ständig neu erzeugt und/oder gelöscht werden. Hierzu dienen die Optionen `IgnoreAdded = regexp` (neu erzeugt ignorieren) und/oder `IgnoreMissing = regexp` (gelöscht ignorieren) in der `[Misc]`-Sektion der Konfigurationsdatei. Das Argument ist jeweils ein [regulärer Ausdruck](#) für den absoluten Pfad.

### 5.6 Hardlink Test

Unter UNIX oder Linux hat eine Directory üblicherweise ebensoviele Hardlinks wie sie Unterverzeichnisse hat (einschließlich . und ..). Samhain testet dies - die Idee dabei ist, dass ein Rootkit ein Unterverzeichnis verbergen kann, aber in der Regel nicht die Zahl der Hardlinks des darüberliegenden Verzeichnisses korrigiert. Dieser Test lässt sich ausschalten mit der Option `UseHardlinkCheck = no` in der **[Misc]**-Sektion.

Das Wurzelverzeichnis einer ReiserFS-Partition hat zwei zusätzliche Hardlinks. Für solche Fälle ist es möglich, mit der Option `HardlinkOffset = N:/directory` in der **[Misc]**-Sektion eine Korrektur anzugeben. Dabei ist N gleich (tatsächliche - erwartete) Hardlinks.

## 5.7 Test auf seltsame Dateinamen

Samhain prüft Dateinamen auf unübliche Zeichen, z. B. Zeilenvorschub, Tabulator o. ä. Mit der Option `AddOKChars = N1, N2, ..` in der **[Misc]**-Sektion lässt sich die Menge der **guten** Zeichen erweitern; dabei ist N1 ... der vorzeichenlose Byte-Wert des Zeichen in hexadezimaler (0xNN), oktaler (0NNN) oder dezimaler Notation.

Mit `AddOKChars = all` lässt sich dieser Test komplett ausschalten.

## 5.8 Änderung einer Policy

Möchte man eine Policy ändern, so geschieht dies in der **[Misc]**-Sektion mit Anweisungen der Form: `RedefPOLICYNAME = +XXX -YYY ...`, wobei die Liste der +/-XXX die Tests sind, die man hinzufügen oder streichen möchte. Mögliche Tests sind: CHK (Prüfsumme), LNK (Name eines Links), HLN (Anzahl Hardlinks), INO (Inode-Nummer), USR (Besitzer), GRP (Gruppe), MTM (Zeit der letzten Änderung), ATM (Zeit des letzten Zugriffs), CTM (Zeit der letzten Inode-Änderung), SIZ (Dateigröße), RDEV (Gerätenummer) und/oder MOD (Zugriffsrechte und Dateityp).

Dabei ist zu beachten, dass eine Policy in der Konfigurationsdatei geändert werden muss, bevor sie benutzt wird, d.h. im nachfolgenden Beispiel wäre es falsch, die **[Misc]**-Sektion hinten anzustellen:

```
/etc/samhainrc

[Misc]
#
# ReadOnly Policy ändern: Inode-Nummer, Besitzer und Gruppe nicht prüfen
#
RedefReadOnly = -INO -USR -GRP
#
# ... und geänderte Policy benutzen:
#
[ReadOnly]
#
# nicht ganz sinnvoll, aber ist ja nur ein Beispiel
#
dir = /usr/bin
```

## 5.9 Zeiten der Integritätsprüfung

Wenn Samhain als Dämon läuft, so gibt es zwei verschiedene Möglichkeiten, in der **[Misc]**-Sektion die Zeiten der Integritätsprüfung festzulegen:

- \* erste Möglichkeit: mit `SetFilecheckTime = Sekunden` ein Intervall festlegen
- \* zweite Möglichkeit: mit `FileCheckScheduleOne = Zeitplan` einen Zeitplan im `crontab`-Stil

festlegen. Dabei sind folgende Abweichungen zu crontab zu beachten: (a) Listen sind nicht erlaubt, und (b) Bereiche mit Namen (z. B. Mon-Fri) sind erlaubt. Es ist möglich, die Option `FileCheckScheduleOne = Zeitplan` mehrmals zu verwenden, um dadurch eine Liste zu konstruieren.

Wenn die zweite Möglichkeit genutzt wird, so gibt es zusätzlich die Möglichkeit, mit `FileCheckScheduleTwo = Zeitplan` einen weiteren Zeitplan zu definieren für Verzeichnisse (keine einzelnen Dateien), die seltener geprüft werden sollen. Die entsprechenden Verzeichnisse müssen mit `%SCHEDULE_TWO ... !%SCHEDULE_TWO` geklammert werden:

/etc/samhainrc

```
[Misc]
#
# Alle 5 Minuten
#
FileCheckScheduleOne = */5 * * * *
#
# Einmal täglich
#
FileCheckScheduleTwo = 37 0 * * *
#
[ReadOnly]
#
dir = /Pfad/oft/prüfen
#
%SCHEDULE_TWO
dir = /Pfad/seltener/prüfen
!%SCHEDULE_TWO
#
```

## 6 Logging zu einem Log-Server (Yule)

Die meisten Programme zur Überwachung der Systemintegrität sind nur für einen einzelnen Rechner gedacht, und bieten von sich aus keine Unterstützung für die zentralisierte Überwachung mehrerer Rechner.

Samhain hingegen unterstützt (optional) sowohl zentralisiertes Logging als auch die zentrale Verwaltung. Konkret bedeutet dies:

- \* Samhain kann so konfiguriert werden, dass Log-Nachrichten an einen Log-Server geschickt werden. Der Log-Server wird *Yule* genannt. Dabei werden Log-Nachrichten in verschlüsselter und signierter Form über [TCP](#) übertragen. Alle Log-Möglichkeiten von Samhain (mit Ausnahme des Weiterschickens an einen weiteren Log-Server) stehen auch für den Log-Server zur Verfügung.
- \* Die Konfigurationsdatei von Samhain kann auf dem Log-Rechner liegen, und Samhain kann diese beim Start herunterladen.
- \* Die Datenbank mit den gespeicherten Eigenschaften von Dateien kann auf dem Log-Rechner liegen, und Samhain kann diese beim Start herunterladen.

Da die Log-Nachrichten von Samhain alle nötigen Informationen enthalten, um die Samhain-Datenbank zu aktualisieren, ist es möglich, die Pflege der Datenbank auf dem Server durchzuführen. Hierzu gibt es eine separate Anwendung namens [Beltane](#).

### 6.1 Übersetzen des Log-Servers (Yule)

Yule wird aus den gleichen Quellen übersetzt wie Samhain. Das `configure`-Skript muss hierbei jedoch mit der Option `--enable-network=server` aufgerufen werden; ansonsten erfolgt das Übersetzen und Installieren analog zu Samhain. Als Name des installierten Programmes wird standardmäßig *Yule* gesetzt.

Um Missverständnissen vorzubeugen: auch wenn Yule aus den gleichen Quellen übersetzt wird, ist es dennoch ein von Samhain verschiedenes Programm. Genausowenig wie Samhain als Log-Server dienen kann, kann Yule die Integrität des Systems überprüfen.

Alle Konfigurations-Optionen, die die Überprüfung des Systems betreffen, sind also irrelevant für Yule; Konfigurations-Optionen bezüglich des Loggings sind dagegen auch für Yule gültig (siehe aber auch den Abschnitt [► Besonderheiten beim Logging](#)).

### 6.2 Übersetzen von Samhain

Samhain kann nur Log-Nachrichten an Yule schicken, wenn es mit der hierfür nötigen Unterstützung übersetzt wurde. Dazu ist es notwendig, das `configure`-Skript mit der Option `--enable-network=client` aufzurufen.

Möchte man außerdem auch die Konfigurationsdatei und die Samhain-Datenbank vom Log-Server herunterladen, so sind die folgenden weiteren Optionen für `configure` notwendig:

- \* `--with-logserver=a.b.c.d` (wobei a.b.c.d die Adresse oder der Name des Log-Rechners ist),
- \* `--with-config-file=REQ_FROM_SERVER/etc/samhainrc` (das `REQ_FROM_SERVER` sagt Samhain, dass die Datei vom Server geholt werden soll), und
- \* `--with-data-file=REQ_FROM_SERVER/var/lib/samhain/samhain_file`.

Dabei ist folgendes zu beachten:

- \* Wird ein Pfad nach `REQ_FROM_SERVER` angegeben, so spielt dieser nur lokal eine Rolle. Samhain kann Yule nicht sagen, unter welchem Pfad die betreffende Datei auf dem Log-Server liegt.
- \* Bei `--with-config-file` ist der lokale Pfad optional. Er wird nur beim Initialisieren der Samhain-Datenbank genutzt, und auch dann nur, wenn das Herunterladen vom Server fehlschlägt (d. h. man kann die Datenbank initialisieren, auch wenn Yule noch nicht bereit ist).
- \* Bei `--with-data-file` ist der lokale Pfad zwingend notwendig, da beim Initialisieren der Datenbank diese in eine lokale Datei geschrieben wird. Es obliegt dem Benutzer, sie dann auf den Log-Rechner zu kopieren.

### 6.3 Authentifizierung gegenüber Yule

Yule nimmt keine beliebigen Anfragen zum Herunterladen von Konfigurationsdateien oder Datenbanken entgegen. Ebenso nimmt Yule keine beliebigen Log-Nachrichten entgegen. Samhain muss sich zunächst authentifizieren, woraufhin Yule und Samhain einen Schlüssel vereinbaren, der für die weitere Kommunikation benutzt wird. Die Authentifizierung erfolgt über das  [Secure Remote Password \(SRP\)](#) Protokoll.

Hierzu sind zwei Zutaten notwendig: Samhain muss ein gültiges Passwort kennen, und Yule muss das Passwort (das für jeden Rechner verschieden sein kann) verifizieren können.

Ein zufälliges Passwort lässt sich erzeugen mit:

```
root@linux / # yule -G
5B5CDF18CE8D66A3
```

Dieses Passwort kann man nun in Samhain einbetten mit:

```
root@linux / # ./samhain_setpwd samhain new 5B5CDF18CE8D66A3
INFO    old password found
INFO    replaced:  f7c312aaaa12c3f7  by:  5b5cdf18ce8d66a3
INFO    finished
```

Das Hilfsprogramm `samhain_setpwd` wird erzeugt, wenn man Yule übersetzt. Es liest ein ausführbares Samhain-Programm (erstes Argument, also `samhain`), und schreibt eine Kopie mit dem gewünschten Suffix (zweites Argument, also `new`), in die das Passwort (drittes Argument) eingebettet ist.

Um das Passwort verifizieren zu können, braucht Yule in seiner Konfigurationsdatei (standardmäßig `/etc/yulerc`) einen entsprechenden Eintrag in der **[Clients]**-Sektion. Falls diese Sektion die letzte in der Konfigurationsdatei ist, kann dies einfach folgendermaßen geschehen:

```
root@linux / # yule -P 5B5CDF18CE8D66A3 | sed
s%HOSTNAME%client.example.com% >> /etc/yulerc tail -2 /etc/yulerc

[Clients]
Client=client.example.com@8A542F99C3514499@744C3A3EE8323470D9DAD42E2485BD0
B138F6B4116E964A9991A0B0D221E1AADE5800968804B99B494C39E7B9DD5710D18F1E6703
D1DB6D6393295E05DF6A6AA8D10BB4A21D7D9DC4901D444500D4EA358C1B44A3E3D44ACEC6
45F938F790A11AB0D03586143977E2BCE3A2D689445AC89134B409E68F34B0DE8BD8242ADD
7C0
```

Der Befehl `yule -P password` erzeugt eine einzige, sehr lange Zeile mit der notwendigen Verifikations-Information. Diese Zeile enthält die Zeichenkette **HOSTNAME**, die ersetzt werden muss durch den Namen des Rechners, auf dem Samhain läuft (dies geschieht hier mit dem `sed`-Befehl). Schließlich muss diese Zeile in die **[Clients]**-Sektion der Konfigurationsdatei von Yule eingetragen werden.

Für jeden Rechner, auf dem Samhain laufen soll, muss eine solcher Eintrag gemacht werden. Das jeweilige Passwort kann, muss aber nicht verschieden sein.

Ist Yule bereits gestartet, so muss mit `kill` das Signal `SIGHUP` an Yule geschickt werden, damit Änderungen der Konfigurationsdatei wirksam werden.

## 6.4 Datenbank- und Konfigurationsdatei auf dem Log-Server

Wenn Samhain beim Start eine Anfrage an Yule macht, um die Datenbank- oder Konfigurationsdatei herunterzuladen, so sucht Yule die entsprechende Datei in seinem Datenverzeichnis (standardmäßig `/var/lib/yule/`). Wenn `client.example.com` der Name des Rechners ist, auf dem Samhain läuft, so muss

- \* die Konfigurationsdatei `rc.client.example.com` heißen, und
- \* die Datenbank `file.client.example.com`.

## 6.5 Besonderheiten beim Logging

Standardmäßig loggt Yule alle Nachrichten von Samhain nur zu Log-Möglichkeiten, die für große Mengen von Daten geeignet sind (Konsole, Logdatei, aber nicht Email oder Syslog), wobei die gewählten [Filter-Einstellungen](#) ignoriert werden (d. h. wenn Samhain konfiguriert ist, eine bestimmte Nachricht an Yule zu schicken, wird Yule diese Nachricht auch tatsächlich loggen).

Um Nachrichten von Samhain durch Yule (nochmals) zu filtern und/oder Log-Möglichkeiten zu benutzen, die für große Datenmengen eher wenig geeignet sind, muss Yule so eingestellt werden, dass auch Nachrichten von Samhain den Filter-Einstellungen von Yule unterliegen.

Hierzu muss man in der **[Misc]**-Sektion die Optionen `UseClientSeverity = yes` und `UseClientClass = yes` setzen.

## 7 Weitere Eigenschaften von Samhain

### 7.1 Signierte Datenbank- und Konfigurationsdateien

Samhain unterstützt **GnuPG**-signierte Datenbank- und Konfigurationsdateien. Dabei ist folgendes zu beachten:

- \* Separate Signaturen werden nicht unterstützt.
- \* Die Signatur muss so erfolgen, dass die Datei nicht verändert wird. Das Ende der Konfigurationsdatei sollte mit **[EOF]** markiert werden, damit Samhain nicht versucht, die Signatur als Option zu interpretieren. Die empfohlenen GnuPG-Optionen zum Signieren sind: `gpg -a --clearsign --not-dash-escaped DATEI`
- \* Der öffentliche Schlüssel des Schlüsselpaares muss im Schlüsselring (`HOME/.gnupg/pubkey.gpg`) des Benutzers sein, mit dessen Rechten Samhain läuft - in der Regel also `root`.
- \* Zur Überprüfung kann man als Benutzer `root` mit dem Befehl `/pfad/zu/gpg --status-fd 1 --verify DATEI` die Signatur überprüfen. Wenn das nicht klappt, wird Samhain auch keinen Erfolg haben ...

Zur Verifikation der Signatur muss Samhain mit der entsprechenden Option übersetzt sein: `./configure --with-gpg=/PFAD_ZU_GPG`.

Es empfiehlt sich, den Fingerabdruck des verwendeten GnuPG-Schlüssels mit anzugeben, damit Samhain auch überprüfen kann, ob der korrekte Schlüssel zum Signieren verwendet wurde.

Weiterhin ist es möglich, die TIGER192-Prüfsumme des GnuPG-Programmes mit anzugeben. Dies schließt die Lücke, dass ein Einbrecher die Samhain-Datenbank modifizieren könnte, und außerdem das GnuPG-Programmes austauschen könnte, um zu verbergen, dass die Signatur nicht mehr gültig ist. Der Nachteil dabei ist, dass Samhain neu übersetzt werden muss, wenn GnuPG aktualisiert wird. Die TIGER192-Prüfsumme lässt sich berechnen mit `gpg --load-extension tiger --print-md TIGER192 /usr/bin/gpg` oder mit `samhain -H /usr/bin/gpg`; die komplette Ausgabe sollte angegeben werden:

```
root@linux / # ./configure --with-gpg=/usr/bin/gpg \  
--with-checksum="/usr/bin/gpg: 1C739B6A F768C949 FABEF313 5F0B37F5 \  
22ED4A27 60D59664" \  
\ --with-fp="EF6C EF54 701A 0AFD B86A F4C3 1AAD 26C8 \  
0F57 1F6C"
```

### 7.2 Stealth (versteckter Modus)

Samhain unterstützt die Möglichkeit, auf versteckte Weise zu arbeiten. Dabei gibt es beim Übersetzen der Quellen die folgenden Optionen:

- \* Mit `--enable-install-name=NAME` wird nicht nur der Name des Programmes geändert; auch alle Verzeichnisse und Dateien, die diesen Namen enthalten (`/var/lib/samhain/`, `/etc/samhainrc`, ...) werden bei der Installation unter geändertem Namen angelegt.
- \* Mit `--enable-nocl[=XYZ]` wird die Interpretation von Optionen auf der Kommandozeile verhindert. Das optionale Argument XYZ ist ein **magisches Wort**, das es ermöglicht, Kommandozeilen-Argumente von stdin zu lesen, z. B.: `echo -- --help | samhain XYZ`
- \* Mit `--enable-micro-stealth=ZAHL` (ZAHL muss ganzzahlig und im Bereich 127 bis 255 sein) werden alle Zeichenketten in den Quellen so verändert, dass sie nicht mehr mit dem Befehl `strings` sichtbar werden. Dies soll verhindern, dass ein Einbrecher nach Samhain suchen kann mit z. B. dem Befehl: `strings /usr/local/sbin/* | grep samhain`.

- \* Mit `--enable-stealth=ZAHL` (ZAHL muss ganzzahlig und im Bereich 127 bis 255 sein) anstelle von `--enable-micro-stealth=ZAHL` werden auch Zeichenketten in der Logdatei und in der Samhain-Datenbank entsprechend verändert. Weiterhin erwartet Samhain nun, dass die Konfigurationsdatei steganographisch in einem Bild versteckt ist. Das Bild muss in unkomprimiertem Postskript-Format vorliegen; dieses Format kann z. B. mit `convert +compress bild.jpg bild.ps` erzeugt werden. Ein Hilfsprogramm `samhain_stealth` steht zur Verfügung, um die Konfigurationsdatei zu verstecken: `samhain_stealth -s bild.ps konfigurationsdatei` zum Verstecken, `samhain_stealth -g bild.ps`, um die versteckte Datei wieder zu sehen.
- \* Mit `--enable-khide=/boot/System.map-$(uname -r)` (das Argument ist also die System.map-Datei, die zum Kernel gehört) wird ein Kernel-Modul erzeugt, das beim Laden alle Prozesse und Dateien versteckt, die die Zeichenkette `samhain` enthalten, bzw. die Zeichenkette `NAME`, wenn die Option `--enable-install-name=NAME` benutzt wurde, um Samhain unter einem anderen Namen zu installieren.

Offensichtlich ist es nicht möglich, die Logdatei einfach mit `more` oder `less` anzusehen, wenn die Option `--enable-stealth=ZAHL` benutzt wird. Stattdessen kann man die Datei mit `samhain -jL /pfad/zur/logdatei | less` ansehen.

### 7.3 Überwachung des Kernels

Samhain kann die Integrität des Kernels überwachen, also prüfen, ob der Kernel zur Laufzeit z. B. durch Laden eines Rootkit-Moduls verändert wurde. Hierzu ist es notwendig, beim Übersetzen der Quellen die Option `--with-kcheck=/boot/System.map-$(uname -r)` anzugeben (d. h. das Argument ist die zum Kernel passende System.map-Datei).

Mit dem Standard-Kernel neuerer Fedora Core-Systeme ist diese Option nicht verfügbar, da durch einen Kernel-Patch die Datei `/dev/kmem` nicht mehr lesbar (oder beschreibbar) ist.

### 7.4 Überwachung von Login/Logout-Vorgängen

Wird Samhain mit der Option `--enable-login-watch` übersetzt, so ist es möglich, Login/Logout-Vorgänge zu überwachen. D. h. es wird eine Nachricht geschickt, wenn ein solcher Vorgang stattfindet.

Diese Funktion beruht auf der Überwachung der `wtmp`-Datei in regelmäßigen Abständen (standardmäßig 300 Sekunden).

### 7.5 Suche nach SUID/SGID-Dateien

Wenn Samhain mit der Option `--enable-suidcheck` übersetzt ist, so ist es möglich, das gesamte Dateisystem regelmäßig nach Dateien zu durchsuchen, bei denen das SUID- oder SGID-Bit gesetzt ist, und neu entdeckte Dateien dieser Art zu melden.

## 8 Optionen in der Konfigurationsdatei

Standardmäßig heißt die Konfigurationsdatei `/etc/samhainrc` (Samhain) bzw. `/etc/yulerc` (Yule).

Die Konfigurationsdatei besteht aus einzelnen Sektionen, die jeweils mit einer Überschrift in der Form `[Sektionsname]` eingeleitet werden. Zeilen, die mit `#` beginnen, sind Kommentare und werden ignoriert.

Optionen haben die Form `Optionsname = Wert`; dabei sind Leerzeichen vor und nach dem Gleichheitszeichen optional.

Alles, was vor der ersten Sektion und nach einem `[EOF]` steht, wird ignoriert. Es ist nicht generell notwendig, die Datei mit `[EOF]` abzuschließen, aber empfehlenswert, wenn am Ende der Datei Dinge stehen, die nicht als Optionen interpretiert werden sollen (z. B. eine GnuPG-Signatur, siehe Abschnitt [Signierte Datenbank- und Konfigurationsdateien](#)).

Manche Optionen sind nur relevant, wenn Samhain bzw. Yule mit Unterstützung für entsprechende Möglichkeiten übersetzt wurde. Falls eine solche Option dennoch benutzt wird, erfolgt eine Warnung beim Lesen der Konfigurationsdatei.

### 8.1 Bedingte Anweisungen

Bedingte Anweisungen ermöglichen es, in die Konfigurationsdatei Optionen zu schreiben, die nur auf bestimmten Rechnern interpretiert (und sonst ignoriert) werden.

Die Bedingung kann entweder der Name des Rechners sein, oder das Tripel **Betriebssystem:Version:Hardware**, das man mit `uname -srm` erhält. Durch Voranstellen eines Ausrufezeichens kann die Bedingung invertiert werden.

```
                                /etc/samhainrc

@abc.example.com
# Wird nur gelesen wenn 'abc.example.com' der Name dieses Rechners ist
@end

!@abc.example.com
# Wird nicht gelesen wenn 'abc.example.com' der Name dieses Rechners ist
@end

$Linux:2.4.26:i686
# Wird nur auf Rechnern gelesen, auf denen
# 'uname -srm' "Linux 2.4.26 i686" ausgibt
$end

!$Linux:2.4.26:i686
# Wird nicht auf Rechnern gelesen, auf denen
# 'uname -srm' "Linux 2.4.26 i686" ausgibt
$end
```

### 8.2 Überwachte Dateien

Die möglichen Sektionsnamen sind hier: `[Attributes]`, `[LogFiles]`, `[GrowingLogFiles]`, `[IgnoreAll]`, `[IgnoreNone]`, `[ReadOnly]`, `[User0]`, `[User1]`, und `[Prelink]`.

Jede dieser Sektionen entspricht einer gleichnamigen [Policy](#) zur Überwachung der Dateien in der betreffenden Sektion.

Jede dieser Sektionen kann beliebig viele Einträge der Form `file = /absoluter/pfad/zur/datei` und `dir = optionale_rekursionstiefe/absoluter/pfad/zum/verzeichnis` haben; dabei ist `optionale_rekursionstiefe` die [Rekursionstiefe](#) für das betreffende Verzeichnis (maximal 99).

```
/etc/samhainrc

[ReadOnly]
dir = /bin
dir = /sbin
dir = /usr/bin
dir = /usr/sbin
dir = 2/boot

file = /etc/motd

[GrowingLogFiles]
file = /var/log/messages
file = /var/log/mail
```

### 8.3 Dringlichkeitsstufe von Ereignissen

In einer Sektion, die mit `[EventSeverity]` eingeleitet wird, ist es möglich, die Dringlichkeitsstufe mancher Ereignisse anzupassen. Mögliche Dringlichkeitsstufen sind: none (nichts loggen), debug (niedrigste Stufe), info, notice, warn, mark, err, crit, alert (höchste Stufe).

Mögliche Ereignisse sind hier:

- \* Veränderungen von überwachten Dateien unter den verschiedenen Policies (Optionen SeverityReadOnly, SeverityLogFiles, SeverityGrowingLogs, SeverityIgnoreNone, SeverityIgnoreAll, SeverityUser0, SeverityUser1, SeverityPrelink)
- \* Zugriffsfehler bei Dateien (Option SeverityFiles)
- \* Zugriffsfehler bei Verzeichnissen (Option SeverityDirs)
- \* Seltsame Dateinamen, verwaiste Dateien (Option SeverityNames)

```
/etc/samhainrc>
```

```
[EventSeverity]
#
# Überwachte Dateien (für die Bedeutung von 'ReadOnly',
# 'LogFiles', usw. siehe Abschnitt Konfiguration der Integritätseigenschaften innerhalb des
# Dateisystems).
#
SeverityReadOnly=crit
SeverityLogFiles=crit
SeverityGrowingLogs=warn
SeverityIgnoreNone=crit
SeverityIgnoreAll=info
#
# Zugriffsfehler für Dateien (Files) und Directories (Dirs)
#
SeverityFiles=err
SeverityDirs=err
#
# Dateinamen, ungültige UIDS/GIDS
#
SeverityNames=info
```

## 8.4 Filter für Log-Möglichkeiten

In einer Sektion, die mit **[Log]** eingeleitet wird, wird definiert, was zu welcher Log-Möglichkeit geloggt werden soll. D. h., es werden die gewünschten [Dringlichkeitsstufen](#) und eventuell [Ereignisklassen](#) eingestellt.

Mögliche Optionen für Dringlichkeitsstufen sind: MailSeverity (Email), PrintSeverity (Konsole), LogSeverity (Logdatei), SyslogSeverity (Syslog), PreludeSeverity (Prelude), ExportSeverity (logging zu Yule), ExternalSeverity (externe Skripte), DatabaseSeverity (logging zu relationaler Datenbank).

Die zuweisbaren Dringlichkeitsstufen sind: none (kein Logging), debug (niedrigste Stufe), info, notice, warn, mark, err, crit, alert (höchste Stufe).

Mögliche Optionen für Ereignisklassen (Standard ist alle) sind: MailClass (Email), PrintClass (Konsole), LogClass (Logdatei), SyslogClass (Syslog), PreludeClass (Prelude), ExportClass (logging zu Yule), ExternalClass (externe Skripte), DatabaseClass (logging zu relationaler Datenbank).

Das Argument ist eine Liste (mit Komma oder Leerzeichen getrennt) von zugelassenen Ereignisklassen:

EVENT	Überwachte Ereignisse (z.B. Änderungen einer Datei)
START	Start/Stop-Meldungen
STAMP	Zeitmarken
LOGKEY	Der Schlüssel zur Verifikation der Log-Datei (siehe Abschnitt <a href="#">Verifikation der lokalen Log-Datei</a> )
ERROR	Fehlermeldungen
OTHER	Alles andere (z.B. detaillierte Informationen)
AUD	System-Aufrufe (für Fehlersuche)

```
/etc/samhainrc
```

```
[Log]
MailSeverity = alert
PrintSeverity = mark
LogSeverity = notice
LogClass = EVENT, START, STAMP, ERROR
SyslogSeverity = err
```

## 8.5 Überwachung von Login/Logout-Ereignissen

In der Sektion **[Utmp]** werden Optionen für die Überwachung von Login/Logout-Ereignissen gesetzt.

LoginCheckActive = yes no	<b>yes</b> zum Einschalten, <b>no</b> zum Ausschalten
LoginCheckInterval = Sekunden	Intervall zwischen Überprüfungen
SeverityLogin = Level	Level/Dringlichkeitsstufe für Login-Ereignis
SeverityLoginMulti = Level	Level/Dringlichkeitsstufe für mehrfaches Login
SeverityLogout = Level	Level/Dringlichkeitsstufe für Logout-Ereignis

```
/etc/samhainrc
```

```
[Utmp]
#
LoginCheckActive = yes
LoginCheckInterval = 10
SeverityLogin = crit
SeverityLoginMulti = crit
SeverityLogout = notice
```

## 8.6 Überprüfung des Kernels

In der Sektion **[Kernel]** werden Optionen zur Überprüfung des Kernels gesetzt. Die folgenden Optionen stehen zur Verfügung:

KernelCheckActive = yes no	<b>yes</b> zum Einschalten, <b>no</b> zum Ausschalten
KernelCheckInterval = Sekunden	Intervall zwischen Überprüfungen
KernelCheckIDT = yes no	Auch Kernel Interrupt Descriptor Table prüfen ? Standard ist ja (yes).
SeverityKernel = Level	Level/Dringlichkeitsstufe für Ereignis

Die folgenden Optionen sind nur notwendig, wenn der Kernel neu übersetzt wurde (auch wenn die Version gleich geblieben ist !). Das Argument **Adresse** ist eine hexadezimale Zahl, mit der die Zeile beginnt, die der jeweils angegebene `grep`-Befehl ausgibt. Die Zahl sollte mit dem Präfix **0x** versehen werden.

KernelSystemCall = Adresse	Adresse der system_call Funktion ( <code>grep system_call System.map</code> )
KernelSyscallTable = Adresse	Adresse der sys_call_table Tabelle ( <code>grep sys_call_table System.map</code> )

KernelProcRoot = Adresse	Adresse der proc_root Funktion ( <code>grep proc_root\$ System.map</code> )
KernelProcRootIops = Adresse	Adresse der proc_root_inode_operations Funktion ( <code>grep proc_root_inode_operations System.map</code> )
KernelProcRootLookup = Adresse	Adresse der proc_root_lookup Funktion ( <code>grep proc_root_lookup System.map</code> )

```
/etc/samhainrc

[Kernel]
KernelCheckActive=yes
KernelCheckInterval=20
KernelCheckIDT=yes
SeverityKernel=crit
#
KernelSystemCall = 0xc0106cf8
KernelSyscallTable = 0xc01efb98
KernelProcRoot = 0xc01efb98
KernelProcRootIops = 0xc01efb98
KernelProcRootLookup = 0xc01efb98
```

## 8.7 Suchen nach SUID/SGID-Dateien

Die Suche nach SUID/SGID-Dateien wird konfiguriert in einer Sektion, die mit **[SuidCheck]** eingeleitet wird. Die Optionen sind:

SuidCheckActive = yes/no	<b>yes</b> zum Einschalten, <b>no</b> zum Ausschalten
SeveritySuidCheck= Level	Level/Dringlichkeitsstufe für Ereignis
SuidCheckInterval = Sekunden	Intervall zwischen Überprüfungen
SuidCheckSchedule = Zeitplan	Zeitplan im <b>crontab</b> -Stil
SuidCheckExclude = /path	Optional ein Verzeichnis, das ausgelassen werden soll
SuidCheckFps = Anzahl	Optional die maximale Anzahl Dateien/Sekunde, die überprüft werden sollen

Es kann entweder SuidCheckInterval oder SuidCheckSchedule angegeben werden, aber beides gleichzeitig ist sinnlos.

```
/etc/samhainrc

[SuidCheck]
SuidCheckActive = yes
SuidCheckSchedule=0 * * * *
SeveritySuidCheck=crit
```

## 8.8 Logging zu einer relationalen Datenbank

Sofern man zu einer Datenbank loggen möchte, z.B. zu einer MySQL- oder [PostgreSQL](#)- Datenbank, muss der Zugriff auf die Datenbank konfiguriert werden. Dies erfolgt in der **[Database]**-Sektion. Die Optionen sind:

SetDBHost = Rechner-Name	Name des Datenbank-Servers. Für PostgreSQL muss dies die numerische IP-Adresse sein.
SetDBName = Datenbank	Name der Datenbank (Standard: samhain)
SetDBTable = Tabelle	Name der Tabelle (Standard: log)
SetDBUser = Benutzer	Als <b>Benutzer</b> anmelden (Standard: samhain)
SetDBPassword = Passwort	Beim Anmelden <b>Passwort</b> benutzen
UsePersistent = yes no	<b>yes</b> für dauerhafte Verbindung zur Datenbank
SetDBServerTstamp = yes no	(nur Yule) Diese Option sagt dem Server, ob er eigene Zeitstempel für Nachrichten von Samhain loggen soll (yes) oder nicht (no)

```

/etc/samhainrc

[Database]
SetDBHost = mysql.example.com
SetDBName = samhain
SetDBTable = log
SetDBUser = samhain
SetDBPassword = jsabfkej
UsePersistent = no

```

## 8.9 Verschiedenes

In der Sektion **[Misc]** finden sich verschiedene weitere Optionen:

Daemon = yes no	Als Dämon laufen (yes = ja)
ChecksumTest = none init update check	Die Standard-Handlung, wenn nicht auf der Kommandozeile angegeben (keine Datenbank initialisieren Datenbank aktualisieren mit Datenbank vergleichen). Standard ist <b>none</b>
VersionString = Zeichenkette	In der Datenbank <b>Zeichenkette</b> zum Markieren der Version einsetzen (zusammen mit Rechner-Name und Zeitstempel)
SetNiceLevel = -19..19	Priorität des Prozesses für Dateiprüfung setzen (-19 = höchste, 19 = niedrigste)
SetIOLimit = kbps	Datenrate (Kilobytes pro Sekunde) begrenzen.
SetLoopTime = Sekunden	Intervall zwischen geloggtten Zeitstempeln
SetFilecheckTime = Sekunden	Intervall zwischen Dateiprüfungen (Standard: 600)
FileCheckScheduleOne = Plan	Zeitplan im crontab-Stil für Dateiprüfungen (alternativ zu SetFilecheckTime)
UseHardlinkCheck = yes no	Anzahl Hardlinks für Verzeichnisse prüfen (yes = ja, no = nein)
HardlinkOffset=N:/Pfad	Ausnahme für Hardlink-Prüfung. N ist der Unterschied (tatsächliche - erwartet) für /Pfad.
AddOKChars = N1, N2, ..	Liste der akzeptablen Zeichen (Byte-Werte) für den Test auf seltsame Dateinamen. Nn muß entweder hexadezimal (führendes <b>0x</b> : 0xNN), oktal (führende Null: 0NNN), oder dezimal sein. Mit <b>all</b> als Argument wird der Test ausgeschaltet.
IgnoreAdded = /regex	Ignorieren, wenn die Datei <b>/regex</b> neu erzeugt wird. <b>/regex</b> kann ein regulärer Ausdruck sein, und muss einen absoluten Pfad bezeichnen.
IgnoreMissing = /regex	Ignorieren, wenn die Datei <b>/regex</b> gelöscht wird. <b>/regex</b> kann

---

ReportOnlyOnce = yes no	ein regulärer Ausdruck sein, und muss einen absoluten Pfad bezeichnen.
ReportFullDetail = yes no	Jede Dateiveränderung nur einmal berichten (Standard: yes = ja)
UseLocalTime = yes no	Alle Details zu einer veränderten Datei berichten (Standard: no = nein)
SetConsole = Gerät	Zeitstempel für Dateien in lokaler Zeit statt GMT berichten.
MessageQueueActive = yes no	Diese Option sollte nicht benutzt werden, wenn Beltane eingesetzt wird.
SetMailTime = Sekunden	Das Konsolen-Gerät bestimmen (Standard: /dev/console).
SetMailNum = 0..127	Die System V IPC message queue benutzen (Standard: no = nein).
SetMailAddress = Empfänger	Maximale Zeitspanne zwischen Email-Nachrichten (Standard: 86400).
SetMailRelay = IP-Adresse	Maximale Anzahl wartender Emails in interner Warteschlange.
MailSubject = Zeichenkette	Einen Email-Empfänger hinzufügen (maximal 8).
SamhainPath = /Pfad	Einen Relay-Rechner für Email festlegen.
SetBindAddress = IP-Adresse	Eigenes Format für die Betreff-Zeile definieren.
SetTimeServer = IP-Adresse	Pfad des Programmes. Falls diese Option benutzt wird, wird die Prüfsumme beim Programmende mit derjenigen zum Programmstart verglichen.
MessageHeader="%S \%T \%F \%L \%C"	IP-Adresse (Netzwerk-Karte) für ausgehende Verbindungen (auf Maschinen mit mehreren IP-Adressen).
SyslogFacility = LOG_XXX	Optionaler Zeitdienst. Es wird das einfache <b>time</b> -Protokoll (37/tcp) benutzt.
HideSetup = yes no	Benutzerdefiniertes Format für den Mail-Header
TrustedUser = user1, ...	Welche syslog-Facility soll benutzt werden (Standard: LOG_AUTHPRIV)
SetDatabasePath = AUTO Pfad	Beim Start keine Namen von Datenbank und Konfigurations-Dateien loggen
SetLogfilePath = AUTO Pfad	Liste zusätzlicher vertrauenswürdiger Benutzer
SetLockfilePath = AUTO Pfad	Pfad zur Datenbank (AUTO um Rechner-Namen an Standard-Pfad anzuhängen)
DigestAlgo = SHA1 MD5	Pfad zur Log-Datei (AUTO um Rechner-Namen an Standard-Pfad anzuhängen)
RedefReadOnly = +XXX -XXX	Pfad zur Lock-Datei, die den Zugriff zur Log-Datei blockiert (AUTO um Rechner-Namen an Standard-Pfad anzuhängen)
RedefAttributes= +XXX -XXX	SHA1 oder MD5 anstelle von TIGER192 als Prüfsummenalgorithmus benutzen (Standard: TIGER192)
RedefLogFiles= +XXX -XXX	Den Test XXX zur Policy <b>ReadOnly</b> hinzufügen bzw. von dieser wegnehmen (Liste möglich).
RedefGrowingLogFiles = +XXX -XXX	Den Test XXX zur Policy <b>Attributes</b> hinzufügen bzw. von dieser wegnehmen (Liste möglich).
RedefIgnoreAll = +XXX -XXX	Den Test XXX zur Policy <b>LogFiles</b> hinzufügen bzw. von dieser wegnehmen (Liste möglich).
RedefIgnoreNone= +XXX -XXX	Den Test XXX zur Policy <b>GrowingLogFiles</b> hinzufügen bzw. von dieser wegnehmen (Liste möglich).
RedefUser0 = +XXX -XXX	Den Test XXX zur Policy <b>IgnoreAll</b> hinzufügen bzw. von dieser wegnehmen (Liste möglich).
	Den Test XXX zur Policy <b>IgnoreNone</b> hinzufügen bzw. von dieser wegnehmen (Liste möglich).
	Den Test XXX zur Policy <b>User0</b> hinzufügen bzw. von dieser wegnehmen (Liste möglich).

---

RedefUser1 = +XXX -XXX	wegnehmen (Liste möglich). Den Test XXX zur Policy <b>User1</b> hinzufügen bzw. von dieser wegnehmen (Liste möglich).
SetLogServer = IP-Adresse MACType=HASH-TIGER HMAC-TIGER	Adresse des Log-Servers Die Art der Authentifizierungs-Codes wählen. Falls diese Option benutzt wird, muss für Yule und Samhain derselbe Wert gesetzt sein. Standard ist HMAC-TIGER.
SetReverseLookup = yes no	Wenn nein (no), dann keinen reverse DNS-Lookup durchführen, wenn Verbindung zu einem Rechner hergestellt wird, der mit Namen statt IP-Adresse angegeben ist.

Die folgenden Optionen sind nur relevant für den Log-Server (Yule).

SetClientFromAccept = yes no	Wenn ja (yes), dann die Adresse des Clients benutzen, wie sie dem Kernel bekannt ist (kann falsch sein, wenn z. B. NAT-Router en route). Sonst (Standard) den Namen benutzen, den der Client bekanntgibt, gegen die IP-Adresse prüfen, und immer akzeptieren (mit einer Warnung, wenn die Prüfung fehlschlägt).
SeverityLookup = Level	Level/Dringlichkeitsstufe für Meldung, wenn die oben erwähnte Prüfung fehlschlägt.
UseClientSeverity = yes no	Wenn ja (yes), Nachrichten von Samhain nicht mit spezieller Dringlichkeitsstufe loggen.
UseClientClass = yes no	Wenn ja (yes), Nachrichten von Samhain nicht mit spezieller Ereignisklasse loggen.
UseSeparateLogs = yes no	Wenn ja (yes), Nachrichten von unterschiedlichen Clients nach unterschiedlichen Log-Dateien loggen (Name des Clients wird an Standard-Pfad angehängt).
SetClientTimeLimit = Sekunden	Maximale Zeit bis zur nächsten Meldung eines Clients (bei Überschreitung Alarm).
SetUseSocket = yes no	Wenn ja (yes), ein Unix Domain Socket öffnen, um Yule Kommandos zu übergeben, die an Clients weitergeleitet werden sollen (wenn diese Kontakt zu Yule aufnehmen).
SetSocketAllowUid = UID	Der Benutzer, der Kommandos über o. g. Socket absetzen kann (Standard: 0 = root)
SetChrootDir = /Pfad	Falls diese Option gesetzt ist, macht der Server nach dem Start ein <code>chroot /Pfad</code>
SetStripDomain = yes no	Wenn ja (yes), wird der Name des Clients ohne die Domäne geloggt (Standard: ja)
SetUDPActive = yes no	Auch auf 514/udp (syslog port) auf Nachrichten von syslog-Dämonen warten

## 8.10 Externe Skripte

In der Sektion **[External]** ist es möglich, externe Skripte oder Programme zu definieren, die zum Loggen benutzt werden können. Das Skript/Programm erhält jeweils eine Zeile mit Informationen, gefolgt von einer weiteren Zeile mit **[EOF]**. Die Optionen sind:

OpenCommand=/Pfad/zum/Skript	Beginnt einen Definitions-Block
SetType = log srv	Mit <b>log</b> wird das Programm als Log-Möglichkeit definiert; mit

SetCommandline = Liste  
SetEnviron = Variable=Wert  
SetChecksum = TIGER Prüfsumme  
SetCredentials = Benutzer  
SetFilterNot = Liste  
SetFilterAnd = Liste  
SetFilterOr = Liste  
  
SetDeadtime = Sekunden  
SetDefault = yes|no

**srv** (nur Yule) wird es aufgerufen, wenn sich der Status eines Clients ändert.  
Die Kommandozeile (also Argumente für das Skript)  
Umgebungsvariable (kann mehrfach benutzt werden)  
Tiger/192-Prüfsumme des Skripts (optional)  
Der Benutzer, als der das Skript ausgeführt werden soll.  
Worte, die in der gelogten Nachricht nicht vorkommen sollen  
Worte, die alle in der gelogten Nachricht vorkommen müssen  
Worte, von denen mindestens eines in der gelogten Nachricht vorkommen muss  
Minimale Zeit zwischen zwei Aufrufen des Skripts  
Standard-Umgebung setzen (yes = ja) Die  
Standard-Umgebung ist: HOME aus `/etc/passwd`,  
SHELL=`/bin/sh`, PATH=`/sbin:/usr/sbin:/bin:/usr/bin`

## 8.11 Clients

In der **[Clients]**-Sektion, die nur für den Server (Yule) relevant ist, hat jeder Samhain-Client einen Eintrag der Form:

```
Client = Rechnername@abc@xyz
```

**Rechnername** ist dabei der volle Name (üblicherweise keine IP-Adresse), unter dem Yule den Client kennt. Falls **Rechnername** falsch ist, wird Yule folgende Fehlermeldung ausgeben: **Invalid connection attempt: Not in client list**, und wird in dieser Fehlermeldung auch sagen, welchen Namen es für den Client ermittelt hat: **client="client.example.com"**.