Härten von Webservern

Armin Hammer

v2.1.3 - 2. Februar 2014

Inhaltsverzeichnis

Kapitel 1

Einführung

Verloren ist der Glaube, dass Programme sicher und gemäss den Versprechungen der Hersteller out-of-the-box nutzbar sind. Man erinnere sich an das das SSL-Exploit[1], welches dem Angreifer innert Sekunden eine lauffähige Kommandozeile auf dem angegriffenen Rechner öffnete. Heute ist davon auszugehen, dass jeder Server sicherheitsrelevante Lücken im Betriebssystem, innerhalb des installierten Webservices und der darauf aufbauenden Skript- oder Programmiersprache aufweist. Im Internet gibt es genügend Foren und Blogs, welche veröffentlichte Lücken so aufbereiten und dokumentieren, dass selbst Unkundige diese für eigene Einbruchsversuche nutzen können.

Steve McConnell schrieb in seinem Buch «Code Complete»[2], dass bei einem durchschnitten Industrieprojekt

> «about 1 - 25 errors per 1000 lines of delivered code»

zu erwarten sind. Die Firma Coverty Inc.[3] hat sich mit ihrem Covery Scan Service auf das Aufspüren und dokumentieren von Softwarefehlern in Softwareprojekten spezialisiert und veröffentlicht jährlich einen Bericht mit Analysen und Statistiken.

Wie betreibt man nun trotzdem einen *sicheren Server*?

Die Anwort ist recht einfach. Der Administrator muss selber Hand anlegen. Man spricht dabei vom *Härten des Systems*. Anders ausgedrückt: Der Administrator erstellt eigene Sicherheitshürden und -barieren, die ein Angreifer zusätzlich überwinden muss, um auf dem geknackten System Schaden anzurichten. Dies ist vergleichbar mit einer Zwiebel - bei der man Schale für Schale entfernen muss, um zum Kern zu gelangen.

Ein Webserver lässt sich in drei Bereiche unterteilen.

- Betriebssystem
- Webservice
- Skript-/Programmiersprache

Die nachfolgenden Kapitel behandeln diese drei Bereiche Schritt für Schritt und zeigen an konkreten Beispielen, welche Möglichkeiten zum Härten des Systems bestehen. Zum Einsatz kommen Slackware Linux[4] als Betriebssystem inklusive OpenSSH[5] für die Remote Administration, Apache[6] als Webservice und PHP[7] als Scriptsprache.

[1] http://www.kb.cert.org/vuls/id/150236
[2] Publication Date: July 7, 2004 | ISBN-13: 978-0735619678 | Edition: 2nd, page 521
[3] https://scan.coverity.com
[4] http://www.slackware.com
[5] http://www.openssh.org
[6] http://httpd.apache.org
[7] http://www.php.net

«Härten von Systemen» ist nicht zu Verwechseln mit «Security through Obscurity»[8], dem Bestreben einiger Herstellern, möglichst wenig über ihre Produkte zu veröffentlichen. Dies in der Hoffnung, dass wenn mögliche Angreifer deren Arbeitsweisen nicht kennen, sie es auch nicht überlisten oder umgehen können. Alle hier beschrieben Massnahmen sind ausführlich im Internet dokumentiert.

Noch ein kleiner Punkt. Die vorliegende Dokumentation befasst sich ausschliesslich mit Einstellungen und Modulen, welche in aller Regel zur Verfügung stehen. Mehr Sicherheit lässt sich mit der Installation zusätzlicher Programme und Module erreichen, z.B. Tripwire, SeLinux und App Armor. Eine Beschreibung dieser Zusätzprogramme würde aber den Rahmen dieser Dokumentation sprengen.

[8] http://en.wikipedia.org/wiki/Security_through_obscurity

Kapitel 2

Betriebssystem Linux

2.1 Serviceaccounts

Dienste und Services des Betriebssystemes starten in der Regel automatisch beim Startvorgang des Servers. Als Benutzerkontext nutzen diese Dienste den Administratoraccount *root*. Knackt ein Angreifer einen dieser laufenden Dienste, hat er durch dessen Berechtigungen Zugang zum gesamten System.

Um dies zu verhindern nutzen Administratoren sogenannte Serviceaccounts. Diese Accounts sind fix einem Service zugeordnet und limitieren dessen Rechte auf dem lokalen System. Als Beispiel sei hier die Unreal Tournament 2003 Server-Engine aufgeführt. Ohne spezielle Vorkehrungen arbeitet dieser Service im Benutzerkontext des Administrators root. Es ist aber ein leichtes, einen neuen Benutzer einzurichten und beim Start des Dienstes diesen als Benutzerkontext zu übergeben.

```
# useradd -s /bin/false -d /opt/ut -m ut2003
# passwd -l ut2003
# chown -R ut2003:users /opt/ut
# chmod -R go-rwx /opt/ut
# su ut2003 -c "/opt/ut/DM-Asbestos > /opt/ut/logs/ut.log" &
```

Die erste Zeile eröffnet den neuen Benutzer *ut2003* und setzt mit -s /bin/false eine ungültige Shell, um ein interaktives Anmelden zu unterbinden. Weiter wird mit der Option -d das Homeverzeichnis /opt/ut fest- und mit -m angelegt. Anschliessend sperrt *passwd -l* das Kennwort von *ut2003*.

Die nächste Zeile erteilt dem frisch angelegten Benutzer die Berechtigungen, im Homeverzeichnis arbeiten zu dürfen. Zusätzlich wird das Homeverzeichnis gegenüber weiteren Gruppenmitgliedern und anderen Benutzern auf dem System gesperrt.

Zuletzt wird der Dienst gestartet wobei mit *su* der Benutzerkontext geändert. Die Kontextänderung lässt sich mit dem Befehl *top* kontrollieren.

```
# top
...
PID   USER   PRI  SIZE  RSS SHARE STAT %CPU %MEM TIME COMMAND
26358 root    11  1044 1044   816 R     0.3  0.2 0:01 top
 1068 ut2003   9   972  844   844 S     0.0  0.2 0:00 DM-Asbestos
 1075 ut2003   9 63420 4488  1768 S     0.0  1.1 0:17 ucc-bin
..
```

Als Kurzreferenz hier ein Auszug der Man-Page.

```
NAME
su -- substitute user identity

SYNOPSIS
su [-flm] [login] [-c shell arguments]]
```

Der «substitute user identity» Mechanismus stösst aufgrund der UNIX-Architektur aber an Grenzen. Die Architektur legt fest, dass Ports unter 1024 nur vom Benutzer root geöffnet, im Fachjargon gebunden, werden dürfen. Ein Webserver auf Port 80 oder ein Mailserver auf Port 25 lässt sich deshalb so nicht starten.

Hier bewährt sich eine andere Vorgehensart. Die Programmierer sind sich des oben beschriebenen Problemes bewusst und schalten, sobald der Benutzerkontext des Administrators nicht mehr benötigt wird - also nach dem Binden des Ports - automatisch auf einen Serviceaccount mit geringeren Berechtigungen um. Bei Apache ist das oft der Benutzer *nobody* in der Gruppe *nogroup*, wie sie von den Direktiven User und Group in der Konfigurationsdatei httpd.conf festgelegt sind.

```
# If you wish httpd to run as a different user or group,
# you must run httpd as root initially and it will switch.
#
# The name (or #number) of the user/group to run httpd as.
#
User nobody
Group #-1
```

RedHat und Novell (aka. Suse) versuchen diesem Problem zusätzlich mit einer weiteren Sicherheitsschicht zu begegenen. Diese zusätzliche Sicherheitsschicht regelt die Berechtigungen noch differenzierter und erlaubt fein granulierbare Zugriffsabstufungen.

SE Linux Security Enhanced Linux[1] erweitert den Linux Kernel um Mandatory Access Control. Der Administrator legt per Regelsatz fest, welcher User worauf zugreifen kann. Dieses ehemals von der amerikanischen NSA angeregte Projekt wird heute von RedHat Enterprise Linux verwendet und von einer Community weiterentwickelt. Die Nutzung in anderen Distributionen ist entsprechend auch möglich.

App Armor Application Armor[2] legt um vorgängig definierte Anwendungen eine Kontrollschicht und schützt so das umgebende System. Der Administrator regelt pro Anwendung deren Berechtigungen mit einem Profil. Das Ursprünglich von der Firma Immunix entwickelte kommerzielle Produkt Subdomain wurde nach dem Kauf von Novell in App Armor umgetauft und 2005 unter GPL gestellt.

Vor- und Nachteile dieser beiden Systemen würde den Rahmen dieses Dokumentation sprengen. In der Ausgabe 06/2006 des deutschen Linux Magazins[3]wird ausführlich darüber berichtet.

2.2 sudo Framework

Alle aktuellen Linux Distributionen sind mit dem sudo Framework[4] ausgestattet. Damit lassen sich Berechtigungen auf Programmebene an normale Benutzer weitergeben. Zusätzlich ermöglicht das Framework eine lückenlose Protokollierung der getätigten Aktionen in */var/log/secure*.

Statusabfrage, was ist für den aktuell angemeldeten Benutzer hinterlegt

```
$ sudo -l
```

Zentrale Konfigurationsdatei */etc/sudoers* für den Administrator

```
# visudo
```

[1] http://selinux.sourceforge.net
[2] http://www.opensuse.org/apparmor
[3] http://www.linux-magazin.de
[4] http://www.sudo.ws

Beispiele

```
# Erlaubt allen Mitgliedern der Gruppe uxadm mittels
# $ sudo su - root Superuserrechte zu erlangen.
%uxadmin ALL=NOPASSWD: /bin/su - root
# Erlaubt allen Mitgliedern der Gruppe uxadmin den Befehl
# rzarch auszuführen, welcher unter dem Benutzer sapadm läuft
%uxadmin ALL=(sapadm)NOPASSWD: /usr/sap/trans/bin/rzarch
```

Der klassische Weg hierfür war bisher das Setzen des Sticky-Bits[5]. Hier regeln das SUDO Framework den Zugriff auf Benutzerbasis eleganter und übersichtlicher.

Grosse Umgebungen mit vielen verschiedenen SUDO Regeln lassen sich sehr einfach verwalten, da nebem der zentralen Konfigurationsdatei auch noch das Verzeichnis /etc/sudoers.d bereit steht. Als Best-Practice hat sich erwiesen, pro Applikation oder grosser Benutzergruppe eine eigene Konfigurationsdatei zu erstellen und verwalten.

2.3 Zugriffsmaske umask

Legt man Dateien in Verzeichnissen mit laschen Zugriffsberechtigungen ab, z.b. /tmp, so muss man vor dem Erstellen der Dateien die Zugriffsrechte so setzen, dass nur die gewünschte Zielgruppe Zugriff erhält. Korrigiert man die Berechtigungen erst danach, kann aufgrund einer eventuellen Race-Condition ein anderer Prozess auf die Datei zugreifen, noch bevor die restriktiveren Berechtigungen aktiv sind.

Besondere Aufmerksamkeit sollte man in diesem Zusammenhang den Backup-Scripts widmen.

```
# Fix access rights
$ umask -S u=rw,g=,o= > /dev/null 2> /dev/null
```

oder kurz in der Kurzform

```
$ umask 0177
```

Setzt die Zugriffsfmaske für neue Dateien auf

```
rw- --- --- username groupname filename
```

anstelle der normalerweise konfigurierten.

```
rw- r-- r-- username groupname filename
```

Die umask Vorgabe des Systems liegt bei Slackwre Linux unter */etc/profile* und lautet

```
umask 022
```

Als Kurzreferenz hier ein Auszug der Man-Page für die Bash-Shell.

```
NAME
umask - get or set the file mode creation mask

SYNOPSIS
umask [-S] [mask]

DESCRIPTION
The user file-creation mode mask is set to mask.
The three octal digits refer to read/write/execute permissions
for owner, group, and other, respectively. The value of each
specified digit is subtracted from the corresponding «digit»
specified by the system for the creation of a file.
```

[5] http://en.wikipedia.org/wiki/Sticky_bit

```
For example, umask 022 removes write permission for group and
other (files normally created with mode 777 become mode 755.
Files created with mode 666 become mode 644).
...
```

Die mögliche Anwendung des Befehls demonstriert das Backup-Scripts im Anhang.

2.4 Kennwörter

Kennwörter dienen zusammen mit dem Benutzernamen zur Authentifikation gegenüber dem System. Der Benutzername ist oft relativ leicht ausfindig zu machen. Das Kennwort dagegen sollte möglichst stark - sprich schwer zu erraten sein.

Ein beliebtes Werkzeug zum Testen von Kennwörter ist *John*. Das nachfolgende Beispiel lädt die aktuelle (Stand 2004) John Version runter, kompiliert und aktiviert diese.

```
# cd /opt/src
# wget http://www.openwall.com/john/b/john-1.6.tar.gz
# tar xzf john-1.6.tar.gz
# cd john-1.6
# cd src
# make linux-x86-mmx-elf
# cd ..
# cd run
# nice -18 john
```

Achtung: Die Entwicklerversionen von John enthalten nicht alle für das Cracken der Accounts notwendigen Dateien. So fehlt zum Beispiel die Datei *all.chr*. Dies stellt aber kein Problem dar, da sie sich einfach von einer älteren Version kopieren lässt.

Der Aufruf mittels

```
$ nice -18 john
```

reduziert die CPU-Last und verhindert damit auffällige Lastspitzen auf dem Server.

Komplexe Passwörter kann man auch erzwingen, ebenso das Sperren und das Ablaufen von Kennwörtern. Linux regelt dies im PAM-Framework, die Passwörter spezifisch in /etc/pam.d/common-password. Hier eine ehemalige Debian Standardkonfiguration

```
#
# /etc/pam.d/common-password - password-related modules common to
# all services
#
# The "nullok" option allows users to change an empty password, else
# empty passwords are treated as locked accounts. #
# (Add 'md5' after the module name to enable MD5 passwords)
#
# The "obscure" option replaces the old 'OBSCURE_CHECKS_ENAB' option in
# login.defs. Also the "min" and "max" options enforce the length of the
# new password.
password required pam_unix.so nullok obscure min=4 max=8 md5
# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
# (Replaces the 'OBSCURE_CHECKS_ENAB', 'CRACKLIB_DICTPATH')
#
# password required pam_cracklib.so retry=3 minlen=6 difok=3
# password required pam_unix.so use_authtok nullok md5
```

PAM ist nicht auf allen Linux Distributionen anzutreffen - Slackware Linux ist so eine Distribution.

2.5 OpenSSH

Telnet und rlogin sind alte Bekannte und sollten heute eigentlich von den Servern verbannt sein, denn sie übertragen die Kennwörter im Klartext. Keine gute Sache.

Das Werkzeug der Stunde ist SSH - Secure Shell. Hierbei wird die Kommunikation verschlüsselt und erlaubt so eine abhörsichere Übertragung von Passwörtern und sonstigen Informationen.

2.5.1 Signatur

Beim Gebrauch von SSH - insbesondere der Open-Source-Variante OpenSSH - werden in der Regel die genauen Versionsinformationen kommuniziert. Dies erleichtert einem Angreifer die Ermittlung der eingesetzten Softwareversion und entsprechend die Suche noch möglichen Lücken.

Abbildung 2.1: Openssh Versionsangabe

Die OpenSSH Versionsinformation liegt in der Datei version.h.

```
#define SSH_VERSION «OpenSSH_4.2»
```

z.B.

```
#define SSH_VERSION «mySSH_X.Y»
```

Gespeichert, frisch kompiliert, installiert und verschwunden ist die Standardkennung.

2.5.2 Tarpitting

Unter Tarpitting versteht man einen Mechanismus, welcher eingehende Verbindungen künstlich verlangsamt, um deren Durchsatz zu verringern. Bekannter ist dieser Mechanismus bei Mail-Servern, um Spam-Attacken zu begegnen. Doch lässt sich dieses Verfahren auch auf Anwendungen anwenden, um Bruteforce-Attacken zu verlangsamen, hier am Beispiel von MD5 Passwörtern.

In der Datei *auth-passwd.c*

```
...
      }
#endif
      result = sys_auth_passwd(authctxt, password);
      if (authctxt->force_pwchange)
         disable_forwarding();

      // result = 0 : invalid user and/or password
      // result = 1 : valid user and password
      if ( result != 1) {
         logit(«tarpitting in action»);
         sleep(30);
      }
      return (result && ok);
...
```

fügt man obigen Code ein. Diese Befehlssequenz hält den Logonvorgang für 30 Sekunden an, falls die Logininformationen nicht korrekt sind. Anschliessend ist die Änderung mit

```
$ make
# make install
# rc.sshd restart
```

zu aktivieren.

So gesichert, kann ein Angreifer selbst bei schnellster Internet und Rechenleistung, genau maximal 2880 Kennwörter pro Tag testen. Diese Verzögerung sollte ausreichen, damit jedes Security Operating Center den Angriff bemerken und stoppen kann.

2.5.3 Parameter

Folgende Parameter in der Konfigurationsdatei *sshd_config* können die Sicherheit erhöhen:

PermitRootLogin no Deaktiviert die Möglichkeit, sich direkt als root anzumelden. Bei der Verwendung des SUDO Frameworks kann der Benutzer trotzem später Root permissions erlangen, falls notwendig

Protocol 2 Beschränkt das Verbindungsprotokoll auf die sichere Version 2.0. Vorherige Versionen besitzen Schwachstellen im Design und sind gezielt angreifbar

MaxAuthTries 6 Beschränkt die maximale Anzahl Versuche pro Verbindung auf 6

MaxStartups 10 Beschränkt die maximale Anzahl nicht authentifizierte Verbindungen auf 10. Weitere Verbindungen werden abgewiesen.

AllowUsers Listed die Benutzernamen explizit auf, welche diesen SSH-Service nutzen dürfen

2.5.4 Schlüssel anstelle von Kennwörter

OpenSSH bietet die Möglichkeit anstelle von Kennwörter mit Schlüsseln zu arbeiten. Die beiden Verschlüsselungsverfahren RSA und DSA stehen in verschiedenen Schlüsselstärken zur Verfügung.

```
user@server1 $ mkdir $HOME/.ssh
user@server1 $ chmod 700 $HOME/.ssh
user@server1 $ cd ~/.ssh
user@server1 $ ssh-keygen -t rsa -b 4096
```

Anschliessend, kopiere den Public-Key auf die zuzugreiffenden Servern.

```
user@server1 $ scp id_rsa.pub user@server2:/home/user/.ssh
user@server2 $ cat id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Nun kann der Benutzer user auf Server1 ungehindert und direkt Befehle auf Server2 ausführen.

```
user@server1 $ ssh user@server2 -c date
user@server1 $ ssh user@server2
```

2.5.5 SCP und SFTP

Mit der Nutzung von OpenSSH als sichere Fernadministration erhält man gratis gleich noch einen sicheren Dateitransfer mit hinzu.

scp secure copy[6]

sftp secure file transfer protocol[7] - do not mistake this with FTP-S, the SSL protected FTP transfer

2.6 Sym-link Attacke

Symbolische Verknüpfungen erleichtern die Arbeit im Dateisystem, indem Verzeichnisbäume oder Dateien an einer anderen Stelle im Dateisysteme wiederum zur Verfügung stehen. Dabei handelt es sich nicht um Kopien, sondern - wie der Name schon ausdrückt - um Verknüpfungen, so dass anstelle der Benutzer anstelle einer Kopie direkt mit dem Original arbeiten kann.

Linux unterstützt zwei Arten von symbolischen Verknüpfungen

- soft-link - Soft-Link ist die flexiblere Art der Verknüpfung. Sie lässt sich systemweit auf Dateien und Verzeichnisse anwenden. Ein Soft-Link ist in der Verzeichnisliste klar ersichtlich (*Typ l*)

  ```
  # ln -s /home opt
  # ls -l /
  ...
  lrwxrwxrwx 1 root root 5 Feb 5 2005 opt -> /home
  ...
  ```

- hard-link - Hard-Links lassen sich nur auf Dateien und auch nur innerhalb des selben Dateisystems anwenden. Einmal erstellt, ist für den Anwender kein Unterschied zwischen Original und Verknüpfung ersichtlich.

  ```
  $ touch test
  $ ln test test2
  $ ls -l
  ...
  -rw-r--r-- 2 user1  group1  0 Jun 11 11:44 test
  -rw-r--r-- 2 user1  group1  0 Jun 11 11:44 test2
  ...
  ```

Wie sich symbolische Links als Angriffswerkzeuge missbrauchen lassen, sei hier an zwei Beispielen erläutert.

[6] http://en.wikipedia.org/wiki/Secure_copy
[7] http://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol

2.6.1 Beispiele

2.6.1.1 Soft-link

Ein Hoster sichert allabendlich die Log-Daten seiner Kunden. Er verwendet dazu das gebräuchliche TAR in Kombination mit GZ. Die Backupdatei .tgz im Backupverzeichnis wird dem Kunden anschliessend zur Verfügung gestellt. Bei einer Beispielstruktur wie

```
/home
 |- kunde1
... |- logs
 |  \- backup
 \ kunde2
    ...
```

könnte es wie folgt aussehen

```
# tar czhf /home/kunde1/backup/kunde1-log.tar.gz /home/kunde1/log
```

Schafft es der nun der Kunde, das Verzeichnis *logs* durch einen Symlink auf /*root* zu ersetzen, hat er damit zwar noch keinen Zugriff auf die Dateien von root, aber vielleicht hat er Glück.

```
$ rm -rf /home/kunde1/logs
$ ln -s /root /home/kunde1/logs
```

Läuft nämlich das Backupscript mit Administratorenrechte, so hat das Backupscript die Berechtigung den Link aufzulösen und die Dateien von root zu lesen und je nach Einstellung (Parameter *-h*) auch zu sichern. Anschliessend erhält der Kunde die Dateien von *root* in seinem Backupordner.

Sym-Link Angriffen kann man auf 2 Arten begegnen, wobei erstere die bevorzugte Variante darstellt.

- Zugriffsberechtigungen
- Symlink-Detection

2.6.1.2 Hard-link

Das Beispiel für eine Hard-Link Attacke ist vergleichbar aufgebaut. Ein Hoster sichert allabendlich die Daten seiner Kunden. Bei einer Beispielstruktur wie

```
/home
 |- kunde1
... |
 |  \- backup
 \- kunde2
    | ...
    \- backup
```

könnte es wie folgt aussehen

```
# tar czf /home/kunde1/backup/backup.tar.gz /home/kunde1
```

Schafft es der nun der Kunde1, einen Hardlink auf das Backup von Kunden2 zu erstellen, kann dieses bei einem Backupscript mit Administratorenrechte mitgesichert werden.

```
$ ln /home/kunde2/backup/backup.tar.gz backup-kunde2.tar.gz
```

Der Kunde1 erhält somit die Backupdatei von Kunde2 in seinem Backupordner.

Sym-Link Angriffen kann man auf 2 Arten begegnen, wobei erstere die bevorzugte Variante darstellt.

- Zugriffsberechtigungen
- Trennung der Partitionen

2.6.2 Zugriffsberechtigungen

Oben genannte Fallbeispiele laufen ins Leere, wenn das Backupscript nicht als root, sondern jeweils mit den Berechtigungen des zu sichernden Kunden läuft.

Bei einem Bash Script lässt sich die mit dem Kommando su, als Abkürzung für substitute user identity, bewerkstelligen. Su ist der selbe Befehl, welcher bereits bei den Serviceaccounts Verwendung findet.

```
#!/bin/bash
su -c kunde1 customerbackup.sh /home/kunde1
su -c kunde2 customerbackup.sh /home/kunde2
...
```

Die Verzeichnisstruktur muss dafür aber bereits entsprechend angelegt und mit Berechtigungen getrennt vorliegen.

```
.../kunde1      drwxrwxrwx 2 kunde1 group1 0 Jun 11 11:44 kunde1
   |- www
   |- scripts
   |- logs
   |- mail
   |- backub
   |- temp
   \- upload
.../kunde2      drwxrwxrwx 2 kunde2 group2 0 Jun 11 11:44 kunde2
   |...
   \- upload
```

Um Bash-Skripte vor dem Ausführem als root zu schützen, lässt sich eine einfache Abfrage durchführen.

```
#!/bin/bash
if [ `id -u` = 0 ]; then
echo "root detected - abort process";
exit 1;
fi
# do the work
echo working
```

Testet die aktuelle Useridentifikation und bricht, falls es der User root ist, die weitere Ausführung ab.

2.6.3 Symlink-Detection

Folgendes Beispiel demonstriert wie in der Bash Symlinks erkannt werden. Zuerst wird eine Testumgebung aufgesetzt.

```
$ cd $HOME
$ mkdir original
$ ln -s $HOME/original symlink
$ vi original/testfile
test file in original directory
:wq
```

Ruft man nun das Testscript auf, ergibt sich nachfolgende Inhalt. Der Inhalt zeigt, dass der Inhalt des originalen Ordners via Originalpfad und dem Symlinks sich auslesen lassen.

```
/home/ahammer
/home/ahammer/symlink
test file in original directory
/home/ahammer
/home/ahammer/original
test file in original directory
/home/ahammer/symlink is a symlink
/home/ahammer/original is NOT a symlink
```

Bash Gurus werden sicher einwenden, weshalb die Kapselung in einer eigenen Funktion und nicht der direkte Aufruf.

```
if [ -L $testdir2 ] ; then
    echo "$testdir2 is a symlink"
  else
    echo "$testdir2 is NOT a symlink"
fi
```

Wer nicht täglich mit Bash Scripts erstellt, wird spätestens nach einem halben Jahr nicht mehr Wissen, dass der Schaltet *-L* auf ein Symlink testet.

2.6.4 Trennung der Partitionen

2.6.4.1 Partitionen

Gegen Hardlinks gibt es neben den klassischen Zugriffsberechtigungen nur eine Möglichkeit. Das System muss von den Daten bzw. die Kunden müssen untereinander getrennt werden und zwar durch verschiedene Partitionen. Hardlinks lassen sich bekanntlich nur innerhalb der selben Partition erstellen.

```
/dev/sda1      /
/dev/sda2      /var
/dev/sda5      /opt
/dev/sda6      /home/customer1
/dev/sda7      /home/customer2
/dev/sda8      /home/customer3
...
```

Leider ist dieses Verfahren mit althergebrachten Werkzeugen und Mechanismen nicht flexibel im Bezug auf die dynamische Anpassung des Speicherplatzes. Hier helfen aber dynamische Speichermanager wie zum Beispiel das Logical Volume Management[8] für Linux - kurz LVM2.

LVM vorbereiten bereitet die Partition sda6 (Typ 0x8e anstelle von 0x83) für die Verwendung mit LVMs vor

```
# pvscan
# vgscan
# pvcreate /dev/sda6
```

Volume Group erstellen erstellt eine Volume Group, welche in Zukunft alle Kundenpartitionen enthalten soll

```
# vgcreate customervg /dev/sda6
```

Logical Volumes erstellen erstellt die Kundenpartitionen mit jeweils 1500MByte Startgrösse

[8] http://tldp.org/HOWTO/LVM-HOWTO/index.html

```
# lvcreate -L 1500 -n customer1lv customervg
# lvcreate -L 1500 -n customer2lv customervg
...
```

Logical Volume vergrössern vergrössert eine Partition um 1 GByte Speicherplatz. Je nach verwendetem Dateisystem kann dies im Onlinebetrieb erfolgen.

```
# lvextend -L +1G /dev/customervg/customer1lv
```

Mit *resize2fs* passt man das aktive Dateisystem ext3/ext4 im laufenden Betrieb der vergrösserten Partition an.

Damit sieht die Partitionsliste wie folgt aus

```
..
/dev/customervg/customer1lv  /home/customer1
/dev/customervg/customer2lv  /home/customer2
...
```

Das ehemalige Kommando *ext2online* ist in heutigen Linux-Systemen nicht mehr zu finden.

2.6.4.2 Dateisysteme

Dateisysteme bzw. deren mögliche Konfigurationen helfen die Zugriffe bereits auf Systemlevel zu kontrollieren und konkret zu steuern. Unter Linux ist pro Mountpoint ein Dateisystem anzugeben, dies ist in der Datei */etc/fstab*[9] konfiguriert.

```
/dev/sda5  swap        swap    defaults         0 0
/dev/sda1  /           ext3    defaults         1 1
/dev/sda6  /var        ext3    defaults,noatime 1 2
/dev/sda7  /home       ext3    defaults,noatime 1 2
/dev/cdrom /mnt/cdrom  iso9660 noauto,owner,ro  0 0
/dev/fd0   /mnt/floppy auto    noauto,owner     0 0
devpts     /dev/pts    devpts  gid=5,mode=620   0 0
proc       /proc       proc    defaults         0 0
```

Die Notation folgt dabei der Konvention

```
Device Mountpoint Filesytem Options Dump-Freq Pass-Num
```

Die hier gezeigten Beispiel beziehen sich ausschliesslich auf ext3[10], dem Standarddateisystem von Linux. Je nach Distribution sind aber auch ext2 oder andere Dateisysteme im Einsatz.

ro Read-Only - Setzt das Dateisystem auf readonly.

noexec Prevents the execution of binaries - Auf Partitionen mit dieser Option lässt sich kein Programm starten. Interessant für Datensammlungen oder rein temporäre Daten (e.g. /tmp)

noatime No acces time - Aktualisiert bei einem Lesezugriff nicht den Last-Access-Timestamp wie es überlicherweise bei Schreibzugriffen geschieht. Dies ist in erste Linie eine Performanceverbesserung, kann aber auch das Aufspüren von unerwünschten Schreibzugriffen vereinfachen

Weitere Optionen lassen die Nutzung des gemounteten Dateisystems einschränken.

nosuid No suid - unterbindet die Nutzung der Sticky-Bits

nodev Unterbindet, dass Devicefiles auf diesem Dateisystem abgelegt werden können

Abhängig von den zusätzlich installierten Packages lassen sich diese Restriktionen aber umgehen, z.B. mit suidperl.

[9] http://en.wikipedia.org/wiki/Fstab
[10] http://en.wikipedia.org/wiki/Ext3

Beispiel Eine sehr gebräuchliche Anwendung zusätzlicher Mount Optionen ist das Härten des globalen, temporären Speicherbereiches in /tmp. Hier der Ausschnit aus /etc/fstab

tmpfs /tmp tmpfs defaults,size=2G,noatime,nodev,nosuid,noexec 0 0

Weiter ist hier das Verzeichnis /tmp in der Grösse von 2 Gigabyte in den Arbeitsspeicher gelegt. Damit ist sicher gestellt, dass bei einem Neustart das temporäre Verzeichnis auch wirklich leer ist.

Testprogramm

```
# vi /tmp/test.sh
#!/bin/sh
echo «Hello World»
# chmod a+x /tmp/test.sh
```

Testverfahren

```
# /tmp/test.sh
-bash: /tmp/test.sh: Permission denied
```

und nun die Verifizierung mit ausgeschalteter noexec Option

```
# mount -o remount,exec /tmp
# /tmp/test.sh
Hello World
```

2.7 Limiten

2.7.1 ulimit

Der intern BASH Befehl ulimit erlaubt das Setzen von Limiten, um den Speicherverbrauch und die Rechenzeit zu begrenzen. Die aktuellen Werte lassen sich wie folgt auslesen.

```
$ ulimit -a
core file size (blocks, -c) 0
data seg size (kbytes, -d) 6144
file size (blocks, -f) unlimited
max locked memory (kbytes, -l) unlimited
max memory size (kbytes, -m) unlimited
open files (-n) 256
pipe size (512 bytes, -p) 1
stack size (kbytes, -s) 8192
cpu time (seconds, -t) unlimited
max user processes (-u) 100
virtual memory (kbytes, -v) unlimited
```

Der Aufruf zum Setzen von neuen Limiten sollte direkt vor dem Start der zu kontrollierenden Anwendung erfolgen.

```
#!/bin/sh
ulimit -S -n 1024
exec httpd
```

Obiges Beispiel erhöht die Anzahl möglicher offener Files - sogenannte File Handles - für Apache auf 1024 und setzt diesen Wert als neue soft limit. Soft limit können - im Gegensatz zu hard limits - per Software bis zum hard limit erhöht werden.

Die genaue Limitenhierachie ist wie folgt definiert

```
open files <= soft limit <= hard limit <= kernel limit
```

Die systemweiten Einstellungen sind in /etc/sysctl.conf hinterlegt.

Aktuell aktive Werte

```
# sysctl -a | grep fs.file-nr
fs.file-nr = 1536 0 262144
```

Die aktuelle Kernellimite beträgt 262144 mögliche Handles wovon aktuell 1536 belegt sind.

2.7.2 nice

Das Kommando nice ändert das Scheduling des angegebenen Prozesses, sprich regelt die einem Programm zuweisene CPU-Zeit.

```
$ nice -n PROGRAM [PARAMETERS...]
```

Der Parameter -n kann Werte zwischen -20 und 20 annehmen. -20 entspricht dabei der höchsten Priotität, so dass ein damit gestartetes Programm maximale CPU-Zeit erhält. 20 dagegen setzt den Prozess auf die minimalste Priorität.

Mit nice ist der Befehl renice verwandt, welcher die Priorität eines bereits laufenden Prozesses ändert.

Kapitel 3

Apache

Als Basis für die nachfolgenden Ausführungen dient Apache in der Version 2.0 Series. Ausführungen, welche nur für die Version 2.2 Series oder neuer gelten, sind speziell ausgewiesen und separat aufgeführt.

3.1 Quellcode

3.1.1 Footprint

Der Webservice Apache generiert eine klare und eindeutige Signatur bei jeder HTTP-Response, z.B. Apache / 2.0.48 (Unix). Script-Kiddies haben so leichten Zugang zur Information über die verwendete Apache Version. Mit dieser Infomation kann man sich in einschlägigen Foren umsehen und Sicherheitslücken dieser spezifischen Version ausfindig machen. Ändert man diese Signatur, wird der Suchaufwand vergrösssert und automatisierte Suchwerkzeuge laufen ins leere.

Die Apache Signatur ist in der Datei *include/ap_release.h* als Konstante abgelegt.

```
#define AP_SERVER_BASEVENDOR "Apache Software Foundation"
#define AP_SERVER_BASEPRODUCT "Apache"
#define AP_SERVER_MAJORVERSION "2"
#define AP_SERVER_MINORVERSION "0"
#define AP_SERVER_PATCHLEVEL "48"
```

Die Signatur sollte möglichst verfremdet werden, jedoch für Menschen lesbar bleiben. Zum Beispiel Ap4che / 9.9.99 (Unix).

```
#define AP_SERVER_BASEVENDOR "Ap4che Software Foundation"
#define AP_SERVER_BASEPRODUCT "Ap4che"
#define AP_SERVER_MAJORVERSION "9"
#define AP_SERVER_MINORVERSION "9"
#define AP_SERVER_PATCHLEVEL "99"
```

Weiter einschränken lassen sich die automatisch migeschickte Serverinformation mit der Einstellung *ServerToken=PROD*.

Ein simples Telnet erlaubt den Blick in die HTTP-Response eines Webservers.

```
$ telnet localhost 80
Connected to localhost
Escape character is '^]'.
GET / HTTP/1.1
HOST: localhost
HTTP/1.1 200 OK
Server: Apache/2.0.48 (Unix) mod_perl/1.99 Perl/v5.8.0 \
mod_ssl/2.0.48 OpenSSL/0.9.7c PHP/4.3.3 DAV/2
X-Powered-By: PHP/4.3.3
<html>.....
```

Dies wäre ein Beispiel, wie man es nicht machen sollte. Neben der Apacheversion, kennt ein möglicher Angreifer nun auch die Perl, OpenSSL und PHP Version. Ein gehärteter Server liefert nur noch folgende Informationen.

```
$ telnet localhost 80
Connected to localhost
Escape character is '^]'.
GET / HTTP/1.1
HOST: localhost
HTTP/1.1 200 OK
Server: Ap4che
X-Powered-By: PI-IP/9.9.9
<html>.....
```

Die Zeile X-Powered-By: PI-IP/9.9.9 kann mit der PHP Direktive *EXPOSE_PHP* deaktiviert werden.

3.1.2 Modulselektion

Apache lässt sich sehr flexibel über Module beim Kompilierungsvorgang an die eigenen Wünsche und Sicherheitsanforderungen anpassen. Das passende Makefile für den Compiler wird durch das Werkzeug *./configure* genriert und steuert anhand von Parametern die Wahl der einzubindenden Apache-Module.

–enable-info Modul, welches über die URL /server-info (Standardeinstellung) den Inhalt der Konfigurationsdatei httpd.conf ausgibt. Der Zugang zu dieser URL sollte unbedingt geschützt werden.

–enable-rewrite Modul, welches ankommende URL-Anfragen gemäss dem vorgegebenen Regelsatz vor der Abarbeitung des Webserver bearbeitet. Damit lässt sich auf einfache Weise URL-Filter und Weiterleitungen realisieren.

–enable-ssl Aktiviert die Secure-Socket-Layer Unterstützung SSL von Apache. Befindet sich die Verschlüsselungsbibliothek nicht in den Standardverzeichnissen unter */usr/include* und */usr/lib*, ist mit der Direktive –with-ssl=/path der Pfad explizit anzugeben.

–with-ssl=/path Pfad zur OpenSSL Verschlüsselungsbibliothek

–disable-userdir Deaktiviert benutzerbasierte Verzeichnisse in Apache

–disable-negotiation Deaktiviert die dynamische Anpassung der HTTP-Responses anhand der Browserkennung

–disable-setenvif Deaktiviert die Umgebungsanpassungen, z.B. automatische Sprachwahl beim Manual.

–disable-imap Deaktiviert die Verarbeitung serverseitig vorbereiteter Imagemaps mit den Endung .map.

–disable-include Deaktiviert den Support von Serverside Includes Dateien .shtml.

–disable-autoindex Deaktiviert die automatische Generierung von Verzeichnislisten, wenn kein Indexfile gemäss der Direktive DirectoryIndex gefunden wurde.

Beispiel

```
$ ./configure --prefix=/opt/apache --enable-ssl \
  --with-ssl=/opt/openssl --enable-info --enable-rewrite \
  --disable-userdir --disable-imap --disable-autoindex \
  --disable-negotiation --disable-setenvif --disable-include
```

Deaktiviert man die oben genannten Module, sind folgende Direktiven aus den Konfigurationsdateien http.conf und ssl.conf zu entfernen.

- UserDir
- SetEnvIf
- IndexOptions
- AddIconByEncoding
- AddIconByType
- AddIcon
- DefaultIcon
- ReadmeName
- HeaderName
- IndexIgnore
- LanguagePriority
- ForceLanguagePriority
- BrowserMatch

Neben den manuell aktivierten Module SSL, Info und ReWrite sind nur noch folgende Module aktiv.

Core HTTP Grundfunktionen

SO Modul für die Nutzung externer Apachemodule, z.B. PHP

Alias Verbindet URL-Pfade mit lokalen Verzeichnissen, z.B. /cgi-bin

Actions Erlaubt MIME-basiertes Starten von CGI-Scripts

Dir Spezifiziert die DirectoryIndex Dateien

Cgi Erlaubt den Aufruf externer Programme via CGI-Interface

Asis Erlaubt das Senden von Antworten ohne die nomalen HTTP-Headers.

Status Erlaubt die Statusabfrage mittels http://localhost/server-status

Mime Modul zur Zuteiltung von MIME-Types zu Dateiendungen

Prefork Standard Multi-Processing Module MPM

Env Erlaubt die Uebergabe von Umgebungsvariablen an CGI-Programmen

Log_Config Modul für die Konfiguration erweiterter Logging-Funktionen

Auth Erlaubt die Nutzung von Allow/Deny, z.B. IP-Restriktionen

Access Erlaubt die Nuztung von Basic Authentication

3.1.3 Reverseanalyse bestehender Binaries

Bei bereits bestehenden Binaries, zum Beispiel aus einer Distribution, listen die Optionen *-V* und *-l* die Kompilationseinstellungen auf.

Version

```
/opt/apache/bin# ./apachectl -V
Server version: Ap4che/9.9.99
Server built: Feb 20 2007 18:46:48
Server's Module Magic Number: 20020903:12
Server loaded: APR 0.9.12, APR-UTIL 0.9.12
Compiled using: APR 0.9.12, APR-UTIL 0.9.12
Architecture: 32-bit
Server compiled with....
-D APACHE_MPM_DIR="server/mpm/prefork"
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D HTTPD_ROOT="/opt/apache-2.0.59"
-D SUEXEC_BIN="/opt/apache-2.0.59/bin/suexec"
-D DEFAULT_PIDLOG="logs/httpd.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_LOCKFILE="logs/accept.lock"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="conf/mime.types"
-D SERVER_CONFIG_FILE="conf/httpd.conf"
```

Module

```
/opt/apache/bin# ./apachectl -l
Compiled in modules:
core.c
mod_access.c
mod_auth.c
mod_log_config.c
mod_env.c
mod_expires.c
mod_ssl.c
prefork.c
http_core.c
mod_mime.c
mod_status.c
mod_asis.c
mod_info.c
mod_cgi.c
mod_dir.c
mod_actions.c
mod_alias.c
mod_rewrite.c
mod_so.c
```

3.1.4 Kompilation

Die Verarbeitungsgeschwindigkeit von Apache kann durch Compilereinstellungen - aka. CFLAGS - erhöht werden, dies in Abhängigkeit der verwendeten Systemarchitektur und Prozessor.

```
./configure .... CFLAGS="-march=athlon -O3 -pipe -fomit-frame-pointer"
```

-march= Systemarchitektur, hier AMD Athlon. Weitere mögliche Parameter sind in der GCC Dokumentation[1] aufgelistet.. Unter Linux kann mit # *cat /proc/cpuinfo* die aktuelle CPU angezeigt werden. GCC v4.4 vereinfacht die optimale Vergabe von Optimierungsangaben mit *-march=native*.

```
# cat /proc/cpuinfo
processor : 0
vendor_id : AuthenticAMD
cpu family : 6
model : 8
model name : AMD Athlon(tm) XP 2000+
stepping : 0
cpu MHz : 1670.797
cache size : 256 KB
...
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
        pat pse36 mmx fxsr sse syscall mmxext 3dnowext 3dnow
bogomips : 3335.78
```

-O3 Höchster Optimierungsgrade des Binärcodes

-pipe Optimiert den GCC Buildprozess - keine eigentliche Auswirkungen auf das Laufzeitverhalten von Apache

-fomit-frame-pointer Optimiert die Stackverwaltung

Weitere CFLAGS Einstellungen erhöhen die Sicherheit.

-fstack-protector-all Vermindert das Risiko von Buffer Overflows indem verschiedene, zusätzliche Prüfunktion in den Stack Code eingebaut werden[2]

```
./configure .... CFLAGS="-march=native -O3 -pipe -fomit-frame-pointer -fstack-p
```

3.2 Konfiguration

3.2.1 Serviceaccounts

Die Serviceaccount nouser und nogroup sind systemweit verfügbar. Nutzt man den Server neben Apache noch für weitere Services, kann es zu Mehrfachbelegung des Accounts führen.

Um diesem Umstand vorzubeugen, haben diverse Linux Distribution (z.B. CentOS) begonnen, für den Apache einen eigenen Serviceaccount und Gruppe zu definiere.

```
User apache
Group apache
```

Dies kann auf älteren System einfach nachgebaut werden.

```
# groupadd -g 48 apache
# useradd -u 48 -g 48 -c «Apache» -s /bin/nologin \
    -d /var/www -m apache
```

[1] http://gcc.gnu.org
[2] http://en.wikipedia.org/wiki/Buffer_overflow_protection

3.2.2 Rewrite

HTTP/1.1[3] bietet verschiedene Methoden für Requests an. TRACE zum Beispiel wird als kritisch bezüglich Sicherheit angesehen, da es URL übergreifende Angriffe ermöglicht. Als einfacher Filter dient das Rewrite Modul.

Folgender Code aktiviert die Rewrite Engine und weisst alle gemäss RFC als nicht «safe» beschriebenen Request Methoden als Fehler zurück.

```
RewriteEngine on
RewriteCond %{REQUEST_METHOD} !^(GET|POST|HEAD)
RewriteRule .* - [F]
```

Dieser Code muss für alle virtuell eingerichteten Webserver wiederholt werden.

3.2.2.1 OPTIONS Beispiel

OPTIONS erlaubt das Auslesen der für die Kommunikation möglichen Parameter und Optionen.

```
$ telnet localhost 80
Connected to localhost
Escape character is '^]'.
OPTIONS / HTTP/1.1
HOST: localdomain.tld
HTTP/1.1 200 OK
Date: Thu, 29 Dec 2005 08:48:01 GMT
Server: Ap4che
Allow: GET,HEAD,POST,OPTIONS,TRACE
Content-Length: 0
Content-Type: text/html
```

Ein Windows 2003 Server als Vergleich.

```
$ telnet win2003sample 80
Connected to win2003sample
Escape character is '^]'.
OPTIONS / HTTP/1.1
HOST: win2003sample
HTTP/1.1 200 OK
Allow: OPTIONS, TRACE, GET, HEAD
Content-Length: 0
Server: Microsoft-IIS/6.0
Public: OPTIONS, TRACE, GET, HEAD, POST
X-Powered-By: ASP.NET
Date: Thu, 29 Dec 2005 09:14:40 GMT
```

3.2.2.2 TRACE Beispiel

Baut man den Trace-Befehl in eine Website ein, so kann man die Informationen des Clients wie auch des Servers einsehen. Nutzbar zum Beispiel, um Informationen über normalerweise geschützte Clients oder Server herauszufinden.

```
$ telnet localhost 80
Connected to localhost
Escape character is '^]'.
TRACE / HTTP/1.1
HOST: localdomain.tld
HTTP/1.1 200 OK
Date: Thu, 29 Dec 2005 08:54:18 GMT
```

[3] http://www.w3.org/Protocols/rfc2616/rfc2616.html

```
Server: Ap4che
Transfer-Encoding: chunked
Content-Type: message/http
2b
TRACE / HTTP/1.1
HOST: locoldomain.tld
0
```

Eine ausführliche Beschreibung finden Sie unter CERT-867593[4].

3.2.3 /server-status

Produktive Server lassen sich auf einfache Weise mittels der URL *http://localhost/server-status* überwachen.

```
Apache Server Status for localhost
Server Version: Ap4che
Server Built: Nov 22 2005 18:37:54
Current Time: Wednesday, 30-Nov-2005 18:52:36 CET
Restart Time: Wednesday, 30-Nov-2005 18:43:21 CET
Parent Server Generation: 1Server uptime: 9 minutes 15 seconds
Total accesses: 10 - Total Traffic: 90 kB
CPU Usage: u.01 s0 cu0 cs0 - .0018% CPU load
.018 requests/sec - 166 B/second - 9.0 kB/request
1 requests currently being processed, 1 idle workers
W_.....................................................................
.....................................................................
.............................
Scoreboard Key:
"_"  Waiting for Connection,
"S"  Starting up,
"R"  Reading Request,
"W" Sending Reply,
"K"  Keepalive (read),
"D"  DNS Lookup,
"C"  Closing connection,
"L"  Logging,
"G"  Gracefully finishing,
"I"   Idle cleanup of worker,
"."  Open slot with no current process
Srv PID Acc   CPU  SS  Conn VHost      Request
0-1 203 4/8/8 0.01 0   30.1 localhost  GET /server-status HTTP/1.1
1-1 206 0/2/2 0.00 479 0.0  localhost  GET /favicon.ico HTTP/1.1
Srv    Child Server number - generation
PID    OS process ID
Acc   Number of accesses this connection / this child / this slot
CPU   CPU usage, number of seconds
SS     Seconds since beginning of most recent request
Conn Kilobytes transferred this connection
```

Diese Infos sind jedoch auch für Hacker sehr interessant. Also schränkt man den Zugriff möglichst ein. In diesem Beispiel auf den IP Adressrange 192.168.200.0/24 und die IP 127.0.0.1.

```
<Location /server-status>
  Order deny,allow
  Deny from all
  Allow from 192.168.200
```

[4] http://www.kb.cert.org/vuls/id/867593

```
  Allow from 127.0.0.1
</Location>
```

Noch mehr Informationen über den aktuellen Status des Apache Webservers erhält man durch setzen der Direktive *ExtendedStatus*.

Aktiviert man die Option *server-status* im globalen Bereich von Apache, so ist diese URL in allen definierten virtuellen Hosts erreichbar! Empfehlenswert wäre also, diese Option nur innerhalb eines dedizierten und speziell gesicherten virtuellen Hostes zu aktivieren.

3.2.4 /server-info

Server-Info liefert die aktuelle Apache Konfiguration als HTML-Seite an den Browser.

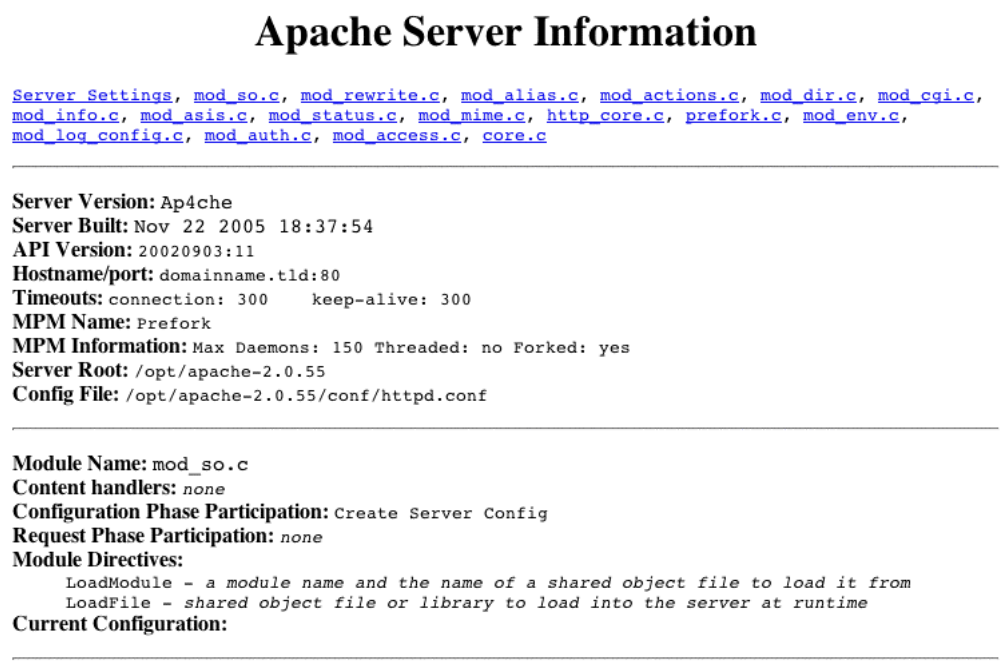


Abbildung 3.1: /server-status Auszug

Die Konfiguration ist vergleichbar mit /server-status.

```
<Location /server-info>
  Order deny,allow
  Deny from all
  Allow from 192.168.200
  Allow from 127.0.0.1
</Location>
```

Zuerst wird die Rangordnung von Deny und Allow festgelegt. Im Sinne von «Es ist alles verboten, es sei den, man erlaube es.» werden alle Zugriffe unterbunden und anschliessend jene Zugriffe aus den spezifizierten Adressen explizit erlaubt.

Weiter einschränken lässt sich der Zugriff mittels Basic Authentication - mehr dazu Siehe unten oder durch setzen eines nicht bekannten Namens.

```
<Location /admin/myprivateserverinfo>
  SetHandler server-info
</Location>
```

Aktiviert man diese Option im globalen Bereich von Apache, so ist diese URL in allen definierten virtuellen Hosts erreichbar! Empfehlenswert wäre also, diese Option nur innerhalb eines dedizierten und speziell gesicherten virtuellen Hostes zu aktivieren.

3.2.5 Basic Authentication

Basic Authentication dient zur Abfrage von Benutzername und Kennwort gemäss der Spezifikation im HTTP Standard. Bei einer Basic Authentication überträgt der Client das Kennwort im Klartext. Schutz vor Lauschangriffen bietet eine zusätzlich aktivierte SSL-Verschlüsselung.

Das Beispiel schützt den Pfad /path mit dem Realm Name REALNAME. Zugelassen werden nur die Benutzer USER1 und USER1 und des weiteren alle Benutzer der Gruppe GROUPNAME.

```
<Directory /opt/www/default/www>
  AuthType Basic
  AuthName "REALMNAME"
  AuthUserFile /opt/www/default/http_passwords
  AuthGroupFile /opt/www/default/http_groups
  Require group GROUPNAME
  Require user USER1, USER2
</Directory>
```

AuthUserFile

Die Passwortdatei verwaltet htpasswd.

```
/opt/apache/bin/htpasswd -c /opt/www/default/http_passwords webadmin
New password: ********
Re-type new password: *********
Adding password for user webadmin
```

Dieser Befehl erstellt die Passwortdatei http_passwords und fügt den Benutzer webadmin hinzu. Verwendet man später erneut die Option -c wird die bisherige Passwortdatei vollständig überschrieben.

```
/opt/apache/bin/htpasswd /opt/www/default/http_passwords webuser
New password: *******
Re-type new password: ******
Adding password for user webuser
```

Fügt einen weiteren Benutzer hinzu.

AuthGroupFile

Die Gruppen werden in einer simplen Textdatei verwaltet und folgen folgender Syntax:

```
gruppe1: user1 user2 user3
gruppe2: user1 user4 user5
```

3.2.6 Footerinformation

Apache fügt bei serverseitig generierten Seiten, z.B. Fehlerseiten oder Directory Listings, in der Standardkonfiguration eine Fusszeile mit Servername und EMail-Adresse des Administrators ein.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
 <head>
  <title>404 Not Found</title>
 </head>
 <body>
  <h1>Not Found</h1>
```

```
  <p>The requested URL /404 was not found on this server.</p>

  <hr>
   Ap4che Server at
   <a href="mailto:you@example.com">localhost</a> Port 80
 </body>
</html>
```

Die Direktive *Serversignatur* deaktiviert diese Funktion.

```
ServerSignature Off
```

3.2.7 HTTP Header Informationen

Seit Apache Version 2.0.44 erlaubt die Option ServerToken zu definieren, wie detailliert sich der Apache im HTTP header zu erkennen gibt.

```
ServerTokens Prod
```

Gemäss dem Apache Manual[5] sind aktuell 6 Stufen definiert.

ServerTokens Prod Server sends (e.g.): Server: Apache

ServerTokens Major Server sends (e.g.): Server: Apache/2

ServerTokens Minor Server sends (e.g.): Server: Apache/2.0

ServerTokens Min Server sends (e.g.): Server: Apache/2.0.44

ServerTokens OS Server sends (e.g.): Server: Apache/2.0.44 (Unix)

ServerTokens Full (or not specified) Server sends (e.g.): Server: Apache/2.0.44 (Unix) PHP/4.2.2 MyMod/1.2

3.2.8 PHP Variablen

Beim Einsatz von PHP als Scripting-Sprache stellt Apache mit den Direktiven *php_admin_flag* und *php_admin_value* Methoden zur Verfügung, um beim Betrieb von virtuellen Hosts eigene PHP-Settings pro virtuellem Host zu übergeben.

Setzen von PHP-Umgebungsvariablen innerhalb der Apache httpd.conf

```
php_admin_value open_basedir value
php_admin_value upload_tmp_dir value
php_admin_value session.save_path value
php_admin_flag register_globals off
```

Siehe PHP Betrieb für den Funktionsbeschrieb der obigen Direktiven.

Ist die Nutzung von .htaccess Dateien erlaubt, können PHP Einstellungen mit den Direktiven *php_flag* und *php_value* verändert werden.

```
php_value memory_limit 20M
php_flag register_globals off
php_flag display_errors off
```

[5] http://httpd.apache.org/docs/2.2/mod/core.html#servertokens

3.2.9 HTTP Expires Header

Der Einsatz der HTTP Expires Header gemäss RFC 2616[6] erhöht nicht die eigentliche Sicherheits eines Servers, sondern dient dazu, die Anzahl der Zugriff zu verringern. Dies reduziert das Datenvolumen eines Netzwerkscans markant und hilft so, sich bei Angriffen auf die eigentlichen Verbindungen zu konzentrieren.

Daneben verbesert das mit den Expires Header kontrollierte Caching auf Seiten der Clients deren Darstellungsgeschwindigkeit.

Aktiviert wird das Module expires mit der Kompliationsoption

```
--enable-expires
```

Anschliessend wird das Modul aktiviert und den Expire-Zeitpunkt pro MIME-Type festgelegt.

```
ExpiresActive On
ExpiresByType image/x-icon "access plus 20 minutes"
ExpiresByType image/gif "access plus 20 minutes"
ExpiresByType image/jpg "access plus 20 minutes"
ExpiresByType image/jpeg "access plus 20 minutes"
ExpiresByType image/png "access plus 20 minutes"
ExpiresByType text/css "access plus 10 minutes"
ExpiresByType text/javascript "access plus 20 minutes"
ExpiresByType text/x-javascript "access plus 20 minutes"
ExpiresByType application/x-javascript "access plus 20 minutes"
ExpiresDefault "now"
```

3.3 Betrieb

3.3.1 Options Indexes und FollowSymLinks

Das Prinzipt, aktiviere nur was wirklich benötigt wird, zieht sich bei den Verzeichnisoptionen weiter. Eine häufige Einstellung

```
<Directory "/opt/www/default/www">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
# Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
...
Options Indexes FollowSymLinks
...
</Directory>
```

aktiviert die Auslieferung von Directory Indexes und das Folgen von Symbolischen Links.

Sicherer ist die Deaktivierung - solange die Applikation diese Funktionen nicht benötigt.

```
Options None
```

Ist nur eine bestimmte Option zu deaktivieren, nutzt man das Minus als Vorzeichen.

```
Options -Indexes
```

Damit lassen sich vererbte Einstellungen überschreiben.

[6] http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

3.3.2 AllowOverride oder «Wie funktioniert .htaccess?»

Bei Shared Hosting Umgebungen ist es die Regel, dass der Kunde selber keine Zugriff auf die zentrale Apache Konfigurationsdatei erhält. Denoch möchte man dem Kunden die Möglichkeit geben, eigene Einstellungen vorzunehmen.
Als Lösung hierfür bietet sich die Nutzung der Konfigurationsdatei «.htaccess» an. Diese Datei - falls vorhanden - liegt in der Regel im Webroot. Hier als Beispiel das .htaccess von Owncloud[7]

```
$ cat .htaccess
ErrorDocument 403 /core/templates/403.php
ErrorDocument 404 /core/templates/404.php
<IfModule mod_php5.c>
 php_value allow_url_fopen On
 php_value upload_max_filesize 513M
 php_value post_max_size 513M
 php_value memory_limit 512M
 <IfModule env_module>
  SetEnv htaccessWorking true
 </IfModule>
</IfModule>
<IfModule mod_rewrite.c>
 RewriteEngine on
 RewriteRule .* - [env=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
 RewriteRule ^.well-known/host-meta /public.php?service=host-meta [QSA,L]
 RewriteRule ^.well-known/carddav /remote.php/carddav/ [R]
 RewriteRule ^.well-known/caldav /remote.php/caldav/ [R]
 RewriteRule ^apps/calendar/caldav.php remote.php/caldav/ [QSA,L]
 RewriteRule ^apps/contacts/carddav.php remote.php/carddav/ [QSA,L]
 RewriteRule ^apps/([^/]*)/(.*\.(css|php))$ index.php?app=$1&getfile=$2 [QSA,L]
 RewriteRule ^remote/(.*) remote.php [QSA,L]
</IfModule>
Options -Indexes
```

Gesteuert wird .htaccess durch die Direktive AllowOverride

```
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
# Options FileInfo AuthConfig Limit
AllowOverride All
```

Ist klar, dass die eigenen Internetanwendung selber keine .htaccess Konfigurationsdatei benötigt, ist es empfohlen, diese Funktion zu deaktivieren.

```
AllowOverride None
```

Das hat zusätzlich den positiven Nebeneffekt, dass die Internetanwendung performanter wird, da nicht bei jedem Seitenaufruf noch nach einer allfällig vorhandenen .htaccess Datei gesucht wird.

3.3.3 X-Frame-Options

Die X-Frame-Options ist ein HTTP Header Feld, welche festlegt, ob und wie eine Webseite in Frames integriert werden soll. Seit Oktober 2013 ist es Teil der offiziellen Standards - RFC 7034[8].
Der Standard definiert aktuell drei mögliche Werte

[7] https://owncloud.com/
[8] http://tools.ietf.org/html/rfc7034

DENY Untersagt jegliche Darstellung der Webseite in einem Frame Tag

SAMEORIGIN Die Seite darf in einem Frame dargestellt werden, falls diese auf der identischen Domain stattfindet

ALLOW-FROM uri Die Seite darf nur in einem Frame darstellt werden, dessen Quelle mit der URI übereinstimmt

Alle aktuellen Browser implementieren diesen HTTP Header und wenden Ihn auf folgende Tags an

1. IFRAME tag
2. Frame tag
3. Object tag (requires a redirect)
4. Applet tag
5. Embed tag

Die spezifische Unterstützung der X-Frame-Options kann direkt online überprüft werden.

```
http://erlend.oftedal.no/blog/tools/xframeoptions
```

Apache unerstützt dieses HTTP Header Feld. Sie kann in der zentralen Konfigurationsdatei wie auch in .htaccess definiert werden.

```
Header always append X-Frame-Options DENY
```

oder

```
Header always append X-Frame-Options SAMEORIGIN
```

Eine Kontrolle der HTTP Header kann direkt im Browser getätigt werden.

Abbildung 3.2: X-Frame-Options: DENY

Content Management Systeme - zum Beispiel Joomla - sind auf die Option SAMEORIGIN angewiesen, da diese den Inhalt einer Seite als iFrame zur Vorschau stellen. Bei DENY bleibt die Voransicht leer.

3.3.4 FileETag Directive

Ein File-Entity-Tag wird zur eindeutigen Kennung einer statischen Datei bei der Auslieferung von Apache automatisch generiert. Sie findet bei der Wiedererkennung der selten Datei für Caching und auf Proxy-Systemen verwendung.

Das FileETag an sich ist keine Security relevante Einstellung, jedoch wird für die Verwendung der Kennung in der Standardeinstellung von Apache den Datei-INode, das Änderungsdatum wie auch die Grösse verwendet.

```
FileETag INode MTime Size
```

Insbesondere der INode Wert lässt Rückschlüsse auf die Position der Datei im Dateisystem zu. Es wird deshalb empfohlen, die Standardeinstellung zu verändern in

```
FileETag MTime Size
```

oder gänzlich auf FileETag zu verzichten.

```
Header unset Etag
FileETag none
```

Verzichtet man auf das FileETag komplett, sind Cache und Ablaufristen mit dem Apache Expire Modul anzugeben.

3.3.5 Aliases /manuals und /icons

Im produktiven Betrieb ist das Apache Manual nicht mehr nötig, bzw. wird konsequenterweise direkt Online von der Apache Projektwebseite[9] gelesen. Das Alias /manual zur mitinstallierten Manual kann demnach entfallen. Ebenfalls sind nach der Deaktivierung der Directory Listings die dazu passenden Icons ebenfalls überflüssig. Das Alias /icons lässt sich bedenkenlos auskommentieren.

3.3.6 Benutzerspezifizierte Fehlerseiten

Benutzerspezifische Fehlerseiten sind ein weiteres Hilfemittel zum Härten von Systemen. Nicht selten stören Hacker mit automatisierten Scripts die Struktur und fehlerfreie Interpretation einer Seite. Anstelle der fehlerhaften Seite mit der Fehlermeldung, wird nun direkt eine vollständige, in der Regel nicht modifizierte Seite angezeigt.

```
ErrorDocument 404 /libs/error404.php
```

3.3.7 Logfile Rotation

Anstelle eines immer wachsenden Logfiles, lässt sich auch eine periodische Rotation einstellen. Insbesondere fürs Reporting sind täglich, rotierende Logfiles wünschenswert.

```
CustomLog "|/opt/apache/bin/rotatelogs access%Y%m%d.log 86400" combined
```

Diese Direktive splittet die anfallenden Logdaten so, dass pro Tag ein Logfile angelegt wird, dessen Namen das Datum der darin enthaltenen Daten darstellt.

```
access20060101.log
access20060102.log
access20060103.log
...
```

Die Notation folgt dabei der Konvention der ANSI-C Funktion strftime(). Konkret wären dies

```
%A full weekday name (localized)
%a 3-character weekday name (localized)
%B full month name (localized)
%b 3-character month name (localized)
%c date and time (localized)
%d 2-digit day of month
%H 2-digit hour (24 hour clock)
```

[9] http://httpd.apache.org/docs-2.0/

```
%I 2-digit hour (12 hour clock)
%j 3-digit day of year
%M 2-digit minute
%m 2-digit month
%p am/pm of 12 hour clock (localized)
%S 2-digit second
%U 2-digit week of year (Sunday first day of week)
%W 2-digit week of year (Monday first day of week)
%w 1-digit weekday (Sunday first day of week)
%X time (localized)%x date (localized)
%Y 4-digit year
%y 2-digit year
%Z time zone name
%% literal '%'
```

Beim Errorlog kann auf die Rotation verzichtet werden. Ein manuelles löschen beim periodischen Audit reicht hier aus.

```
ErrorLog logs/error.log
```

Möchte man überhaupt kein Logging, ist dieses mit

```
CustomLog logs/access.log combined env=DOES_NOT_EXIST
```

zu deaktivieren.

3.3.8 Secure Socket Layer

Apache nutzt die freie Bibliothek OpenSSL als Basis für verschlüsselte HTTP Übertragung - *https*. Die Verschlüsselung gilt dabei für eine URL, z.B. http://secure.vhost.tld.

Um SSL ranken sich zwei Missverständnisse, die hier gleich zu Beginn ausgeräumt werden sollen.

1. Ein Zertifikat ist an eine URL gebunden, z.B. secure.vhost.tld. Umzüge auf andere Server sind ohne Probleme möglich, solange die selbe URL verwendet wird. Eher selten und nur schwer zu bekommen sind so genannte Wildcard-Zertifikate, z.B. *.host.tld. Sie gelten für eine Gruppen von URLs. In diesem Fall zum Beispiel für secure.vhost.tld, www.vhost.tld und so weiter.

2. Für eine SSL-Verbindung müssen IP und Port eindeutig sein, z.B. 127.0.0.1:443. Eine weitere SSL-Verbindung auf dem selben physikalischen Rechner muss sich in Port oder IP unterscheiden, z.B. 127.0.0.1:444 oder 127.0.0.2:443. SSL-Verbindungen lassen sich nicht per Hostheader trennen, wie es bei HTTP möglich ist.

Halt STOP - um der Wahrheit die Ehre zu geben, ist der zweite Punkt so nicht mehr korrekt. Seit 2006 existiert die RFC4366[10] - Transport Layer Security (TLS) Extensions, welche genau dies - also die Nutzung von Hostheader bei SSL-Verbindungen spezifiert und zulässt. Die technische Implementation nennt sich «Server Name Identication» (SNI).

Leider lassen sich die Webserver- und Browserprogrammierer Zeit mit der Implementation von SNI. So ist selbst im Jahre 2010 die Verbreitung noch gering, z.B. Windows XP ist nicht SNI tauglich.

SSL basiert auf Zertifikaten, die der Browser anhand einer Referenzliste von sogenannten Certifizierungsinstanzen überprüfen kann.

[10] http://www.ietf.org/rfc/rfc4366.txt

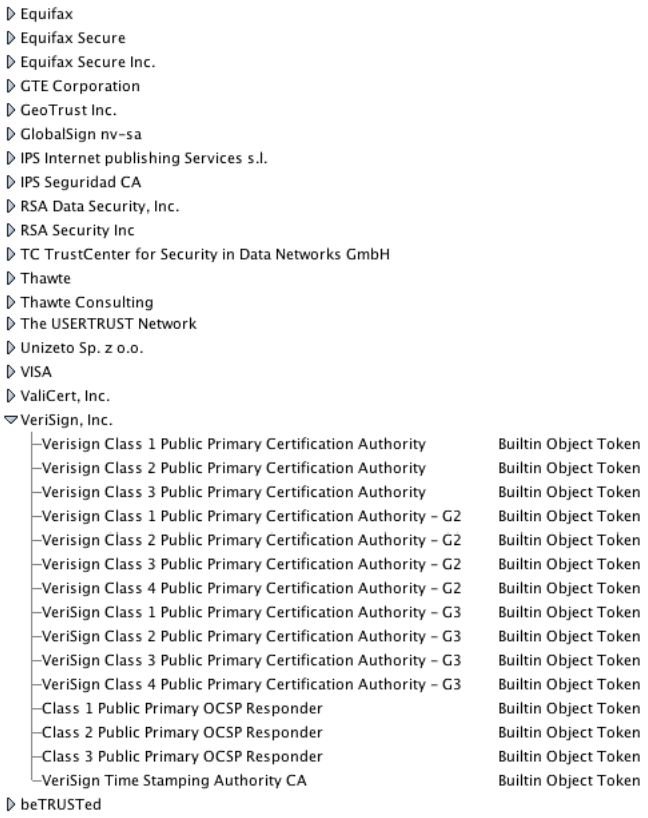


Abbildung 3.3: Zertifizierungsinstanzen Auszug

Die bekannteste Certifizierungsinstanz - certification authority CA - ist Verisign.

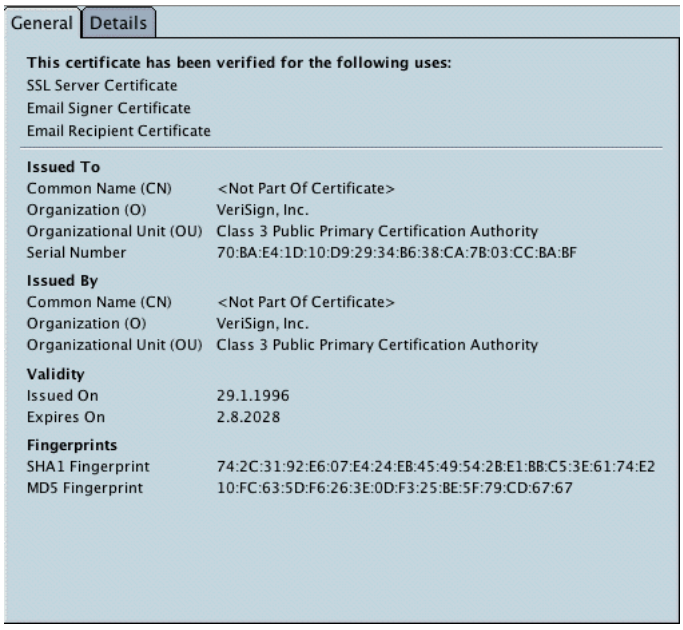


Abbildung 3.4: Zertifikatsdetails Versign

Mit ca. 400$ für das Erstellen und knapp 150$ pro Jahr für die Erneuerung ist Verisign[11] die teuerste Zertifizierungsinstanz. Einfacher und günstiger ist es in der Regel,

[11] http://www.verisign.com

kleine Zertifizierungsinstanzen - z.B. Comodo[12], RapidSSL 60.- Fr/Jahr oder GeoTrust - anzugehen bzw. selber eine Certifizierungsinstanz aufzubauen und zu betreiben.

Certification Authority

Eine Zertifizierungsinstanz besteht aus einem privaten Schlüssel und einem daraus generierten Zertifikat. Im folgenden wird der zugrunde liegende Schlüssel im tripple DES Verfahren und einer Schlüssellänge von 2048 Bits erstellt.

```
# ./openssl genrsa -des3 -out ca.key 2048
Generating RSA private key, 2048 bit long
modulus.........................................+++
..+++e is 65537 (0x10001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
```

Der private Schlüssel wird mit einer Passphrase gesichert.

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,B34B0D5C93F00FA6
7Ah5ejerE2ca/xQp1jreN+6rbpBeYhzHZdVdl91ia7KZSrtQTb7lhMrO3KMO4vwA
...
527hVnjnVO09knl3eiCc3aB52A8rjNp/ka5ALPDEz4lQepOu4OYZqrywFETb09Hx
-----END RSA PRIVATE KEY-----
```

Exkurs Schlüssellängen Als die erste Version dieses Dokumentes 2005 erschien, waren Schlüssellängen mit 2048 Bits und mehr als sicher eingestuft und empfohlen. Mit der rassanten Weiterentwicklung der Rechenleistung, steigen nun die Schlüssellängen. Damit werden ehemals sichere Schlüssel neu als schwache Schlüssel eingestuft und sind nicht mehr empfehlenswert.

Bei jeder Erneuerung von Schlüsseln sollten deshalb aktuelle Versionen von Standardwerken wie zum Beispiel FIPS-140[13] oder dem IT Grundschutzhandbuch der BSI[14] über die empfohlene Länge von Schlüssel konsultiert werden.

Basierend auf diesem Schlüssel erstellt man ein Stammzertifkat im X509 Format und in diesem Beispiel mit einer Gültigkeitsdauer von 3 Jahren.

```
# ./openssl req -new -x509 -days 1095 -key ca.key -out ca.crt
Enter pass phrase for ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank.
For some fields there will be a default value,If you enter '.', the field
will be left blank.
-----
Country Name (2 letter code) [AU]:CH
State or Province Name (full name) [Some-State]:Central Switzerland
Locality Name (eg, city) []:Rotkreuz
Organization Name (eg, company) [Internet Widgits]:Sample Company
Organizational Unit Name (eg, section) []:Sample Unit
Common Name (eg, YOUR name) []:Sample Name
Email Address []:postmaster@domain.tld
```

Dies generiert folgendes Zertifkat.

[12] http://www.comodo.com
[13] http://en.wikipedia.org/wiki/FIPS_140-2
[14] https://www.bsi.bund.de

```
-----BEGIN CERTIFICATE-----
MIIE6zCCA9OgAwIBAgIJALHR9TKvags5MA0GCSqGSIb3DQEBBAUAMIGpMQswCQYD
...
ZEjiI5waMWhQVkPYO1Sp
-----END CERTIFICATE-----
```

Serverschlüssel und CSR

Beim Server startet man ebenso mit einem privaten Schlüssel.

```
# ./openssl genrsa -des3 -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.............+++
....+++e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
```

Auch dieser ist mit einer Passphrase geschützt. Die Passphrase ist bei jeder Nutzung des Schlüssels notwendig - deshalb auch beim späteren Start von Apache. Um nicht bei jedem Start des Webservers die Passphrase neu einzugeben, kann man den Passphrasenschutz deaktivieren.

Aber Achtung! Der Schlüssel ist dann offen und ohne Schutz!

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,42AC8046F8BECF2D
PBc0K3EUUhtFwyqLwniIZlj2zIN00rMUWAwL7SV3i6i7Lxg32FwIBYcf4fFi1Hw9
...
ygLIwbOqG+t2sXfBs+35qxjLQrKrT/Im2IDcCqV0S0FSfFcKN4LNUQ==
-----END RSA PRIVATE KEY-----
```

Nach dem Schlüssel nun das CSR. Dieses Zertifikat lässt man anschliessend von einer CA zertifizieren. In unserem Beispiel erledigen wird das selber. Dazu später mehr.

```
#  ./openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.What you are about to enter is what is
called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blankFor some fields
there will be a default value,If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CH
State or Province Name (full name) [Some-State]:Central Switzerland
Locality Name (eg, city) []:Rotkreuz
Organization Name (eg, company) [Internet Widgits]:Sample Company
Organizational Unit Name (eg, section) []:Sample Unit
Common Name (eg, YOUR name) []:secure.vhost.tld
Email Address []:postmaster@vhost.tld
Please enter the following 'extra' attributesto be sent with your
certificate request
A challenge password []:
An optional company name []:
```

Das zu signierende Certifikat - server.csr.

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC8zCCAdsCAQAwga0xCzAJBgNVBAYTAkNIMRwwGgYDVQQIExNDZW50cmFsIFN3
...
zryWw6ldOXrautvwRfDt0bcmLX16r9sNGTQ2m9SRzSfCY+ylAh4M
-----END CERTIFICATE REQUEST-----
```

Selfsigned Certifikate

Um Certifikate selber zu zertifizieren existieren verschiedene fertige Scripts. Hier jenes von Ralf S. Engelschall[15].

```
# ./sign.sh server.csr
CA signing: server.csr -> server.crt:
Using configuration from ca.config
Enter pass phrase for ./ca.key:#
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CH'
stateOrProvinceName   :PRINTABLE:'Central Switzerland'
localityName          :PRINTABLE:'Rotkreuz'
organizationName      :PRINTABLE:'Sample Company'
organizationalUnitName:PRINTABLE:'Sample Unit'
commonName            :PRINTABLE:'secure.vhost.tld'
emailAddress          :IA5STRING:'postmaster@vhost.tld'
Certificate is to be certified until Feb 28 17:48:07 2008 GMT (1095 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: server.crt <-> CA cert
server.crt: OK
.
server.crt
Certificate:
  Data:
    Version: 1 (0x0)
    Serial Number: 1 (0x1)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=CH, ST=Central Switzerland, L=Rotkreuz, O=Sample Company,
    OU=Sample Unit, CN=Sample Name/emailAddress=postmaster@domain.tld
    Validity
      Not Before: Feb 28 17:48:07 2005 GMT
      Not After : Feb 28 17:48:07 2008 GMT
    Subject: C=CH, ST=Central Switzerland, L=Rotkreuz, O=Sample Company,
    OU=Sample Unit, CN=secure.vhost.tld/emailAddress=postmaster@vhost.tld
    Subject Public Key Info:
       Public Key Algorithm: rsaEncryption
       RSA Public Key: (2048 bit)
          Modulus (2048 bit):
            00:e5:7a:37:c6:0f:c1:37:7b:1d:45:dd:fc:d5:67:
            ...
            8b:b0:c1:5c:01:d2:cf:bf:f8:0c:8a:3c:04:05:90:
            a9:4b
            Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        60:91:3a:d5:d6:81:db:84:c7:d6:ef:2e:d2:9c:bb:e4:cb:c0:
       ...
       d9:c6:78:98
-----BEGIN CERTIFICATE-----
MIIDzDCCArQCAQEwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAkNIMRwwGgYD
...
7ZnAYl3liFOcujr+2cZ4mA==
-----END CERTIFICATE-----
```

[15] http://engelschall.com/

Das Zertifikat für den Server ist nun fertig. Nun entfernt man noch die Passphrase vom Serverschlüssel.

```
$ cp server.key server.key.org
$ ./openssl rsa -in server.key.org -out server.key
Enter pass phrase for server.key.org:
writing RSA key
```

Der Vollständigkeit halber hier noch der nun offene Serverschlüssel - server.key.

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA5Xo3xg/BN3sdRd381WctDTP/37MuxYHctM9qd5Vp2p9Z9TNn
...
ixglPDmPI69LEAyDrqi62wH5yIH/kP5OFE18V5Zbm4+MIRjpu3mA
-----END RSA PRIVATE KEY-----
```

MD5 Hash

Im Dezember 2008 veröffentlichte das CERT Programm den Vulnerability report «MD5 vulnerable to collision attacks»[16]. Dieser beschreibt, dass mit MD5 Hash Kollisionen, neue, eigene Security Tokens generiert werden können, welche authentisch erscheinen. Unter http://www.win.tue.nl/hashclash/rogue-ca/ ist der komplette Angriff auf diesen Vektor im Detail beschrieben.

Entsprechend muss das Sign.sh Script angepasst werden, um anstelle den als unsicher eingestuften MD5 Hashes durch einen SHA1 Hash zu ersetzen.

Ein einfaches Shell Script von Scott Yang[17]erlaubt eine Kontrolle.

```
#!/bin/sh
echo "HEAD / HTTP/1.0 Host: $1:443
EOT " \
| openssl s_client -connect $1:443 2>&1 \
| sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p' \
| openssl x509 -noout -text -certopt no_signame \
| grep 'Signature Algorithm:'
```

Aktuelle Zerfikate ergeben den SHA1 Algorithmus als Resultat.

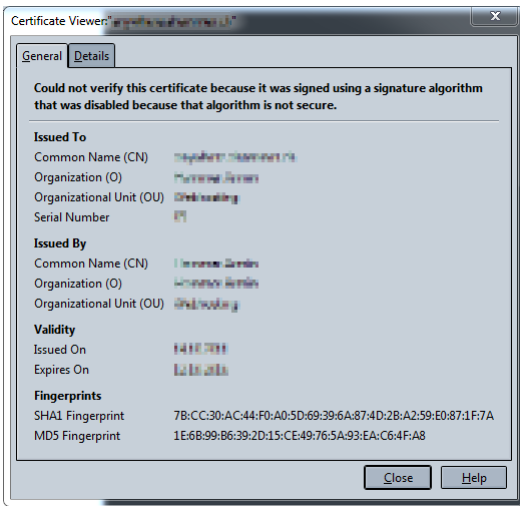
```
$./get_sig_algo.sh www.ahammer.ch
    Signature Algorithm: sha1WithRSAEncryption
```

Ältere dagegeben weisen den MD5 Hash aus.

```
Signature Algorithm: md5WithRSAEncryption
```

Aktuelle Browser erkennen und markieren SSL Zertifikate, welche mit einem MD5 Hash gesichert sind!

[16]http://www.kb.cert.org/vuls/id/836068
[17]http://hostingfu.com/article/your-ssl-certificates-signed-using-vulnerable-md5



3.3.9 SSL Verschlüsselungssträrke

Ältere Verschlüsselungsprotokolle stammen aus der Zeit der US-Exportbeschränkungen für starke Verschlüsselungen und sind heute überholt bzw. als unsicher einzustufen - vornehmlich die 56Bit Verschlüsselung. Um Apache die neuen Protokolle zu favorisieren dienen folgende Kommandos.

```
SSLProtocol -ALL +SSLv3 +TLSv1
SSLCipherSuite ALL:!aNULL:!ADH:!eNULL:!LOW:!EXP:RC4+RSA:+HIGH:+MEDIUM
```

Ein Beispiel

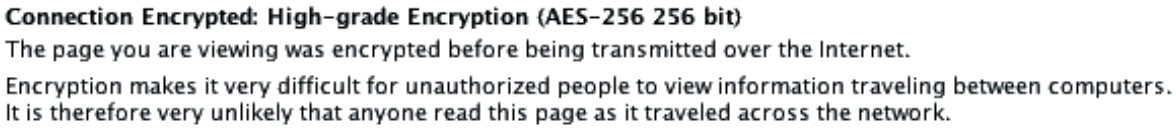


Abbildung 3.5: Bestätigung einer aktiven hohen Verschlüsselung

3.3.10 Include

Bei der Vielzahl von Einstellungen und verschiedenen vHosts verliert man schnell den Überblick. Eine sehr elegante Möglichkeit besteht nun darin, Einstellungen zu den einzelnen virtuellen Hosts auszuglieder. Frei nach dem Motto *"Teil und Herrsche"*.

Apache bietet dazu die Direktive Include.

```
<VirtualHost *:80>
DocumentRoot /opt/www/default/www
....
</VirtualHost>
<VirtualHost *:80>
DocumentRoot /opt/www/vhost.tld/www
...
</VirtualHost>
```

Wird vereinfacht, indem man beide VirtualHost Direktiven in je einen eigenen Datei auslagert.

```
Include /opt/www/default/httpd.conf
Include /opt/www/vhost.tld/httpd.conf
```

Auch die Fehlersuche wird dadurch vereinfacht.

```
# ./apachectl -S
VirtualHost configuration:
wildcard NameVirtualHosts and _default_ servers:
*:80 is a NameVirtualHost
   default server localhost.localdomain (/opt/www/default/httpd.conf:1)
   port 80 namevhost localhost.localdomain (/opt/www/default/httpd.conf:1)
   port 80 namevhost vhost.tld (/opt/www/vhost.tld/httpd.conf:1)
Syntax OK
```

Ein Test der virtuellen Hosts zeigt anschliessend den entsprechende Konfigurationsdatei an, welche die dazu nötigen Directiven enthält.

3.3.11 default website

Die Vorlage von Apache basiert auf einer einzigen Website, die auf alle IP-Adressen des laufenden Servers reagiert. Hackertools nutzen diese unvorsichtige Konfiguration aus, da der Webserver bereits auf die reine Angabe der IP-Adresse die dazu konfigurierte Website ausliefert. Automatisierte Werkzeuge können so ein leichtes alle benötigten Informationen eines Webservers abrufen.

Dem lässt sich abhelfen, wenn man mit sogenannten Hostheadern arbeitet. Dabei muss nicht nur die IP-Adresse stimmen, sondern auch noch ein vorgegebener Name, z.B. www.vhost.tld. Bei Apache nennt sich dies *virtual hosts*.

Beispiel

```
# aktiviert das virtuelle Hosting auf Port 80
NameVirtualHost *:80
<VirtualHost *:80>
  DocumentRoot /opt/www/default/www
  ...
</VirtualHost>
<VirtualHost *:80>
  DocumentRoot /opt/www/www.vhost.tld/www
  ServerName www.vhost.tld
  ServerAlias vhost.tld
  ...
</VirtualHost>
```

Der erste virtuelle Host übernimmt dabei die Rolle der ohne diese Funktion bestandenen Standardsite (default website), die bereits auf die IP-Adresse des Servers reagiert. In der Regel wird diese Website gesperrt, z.B. mit einer Basic Authentication oder einer Rewrite Rule. So dass diese Seite keine korrekte Antwort wiedergibt.

```
RewriteEngine on
RewriteRule .* - [F]
```

Der nächste virtuelle Host spezifiziert mit der Direktive ServerName seinen Hostheader. Also wenn eine Anfrage neben der IP-Adresse des Servers noch diesen Hostheader angibt, so antwortet Apache mit dieser Website. Mit ServerAlias lassen sich zusätzliche Hostheader angeben, die ebenfalls zu diesem virtuellen Host gehören.

Achtung

Besitzt ein VirtualHost den selben Servernamen wie die Default Website, wird anstelle des virtuellen Services der Default Website aufgerufen. Dies gibt auch dann, wenn im ersten VirtualHost Eintrag (Default Website) kein ServerName Eintrag steht. Es wird dann einfach jener in der Grundkonfigurationsdatei httpd.conf verwendet.

3.3.12 chroot Jail

Reichen alle bisher beschriebenen Massnahmen nicht für die gewünschte Sicherheit, so besteht auch die Möglichkeit des Betriebes von Apache in einem Jailroot Container. Die Grundidee dahinter besteht darin, dem laufenden Programm vorzugaukeln, es arbeite im produktiven Dateisystem. In Wahrheit arbeitet dieses aber in einer extrem abgespeckten Umgebung, spezfisch angepasst nur für dieses eine Programm.

```
/mnt
  \- chroot
      \- apache
         |- etc
         |- lib
         |- bin
         |- opt
         |   |- apache
         |   \- www
         \- var
```

Wie man sieht, ist im Verzeichnisbaum /opt/www/chroot/apache das bisher bekannte Dateisystem mit den benötigten Verzeichnissen /etc, /bin, /lib, /opt und /var abgebildet. Selbst wenn jetzt ein Cracker via Apache ins System einbricht und Daten aus /etc auslesen möchte, erhält er statt dessen nur die Dateien aus /opt/www/chroot/apache/etc.

Die chroot Umgebung platziert man bevorzugt auf einer eigenen Partition, um Hardlinks ins umgebende Dateisystem zu verhindern.

Programme in diesem Container haben nur Zugriff auf die abgespeckten Verzeichnisse. Die Schwierigkeit bei heutigen System besteht nun darin, dass die vielen Abhängigkeiten hier ebenfalls erfüllt sein müssen. Shared Libraries, Shells, Kommandozeilenwerkzeuge und Einstellungen muss man manuell kopieren und im Trial&Error Verfahren ausfindig machen.

Unterstützung bietet hier die Werkzeuge lsof und ldd.

lsof list open files - zeigt also die aktuell offenen Dateien des laufenden Systems an.

ldd print shared library dependencies - zeigt die für das Programm benötigten Bibliotheken an. Unter MacOsX steht ldd nicht zur Verfügung, hier hilft otool -L als Ersatz.

Hier ein paar Beispiele

```
$ lsof
COMMAND  PID  USER   FD  TYPE  SIZE/OFF  NAME
...
lsof     183   aha  cwd  VDIR       578  /private/opt/apache-2.0.55/bin
lsof     183   aha  txt  VREG    111380  /usr/sbin/lsof
lsof     183   aha  txt  VREG   1797788  /usr/lib/dyld
lsof     183   aha  txt  VREG   4379472  /usr/lib/libSystem.B.dylib
lsof     183   aha   0u  VCHR    0t24812 /dev/ttyp1
lsof     183   aha   1u  VCHR    0t24812 /dev/ttyp1
lsof     183   aha   2u  VCHR    0t24812 /dev/ttyp1
lsof     183   aha   3r  VCHR 0t58022148 /dev/kmem
lsof     183   aha   5w  PIPE
lsof     183   aha   6r  PIPE
```

```
Linux$ ldd httpd
  libaprutil-0.so.0 => /opt/apache-2.0.55/lib/libaprutil-0.so.0 (0x..)
  libdb3.so.3 => /usr/lib/libdb3.so.3 (0x4002c000)
  libexpat.so.0 => /opt/apache-2.0.55/lib/libexpat.so.0 (0x400d5000)
  libapr-0.so.0 => /opt/apache-2.0.55/lib/libapr-0.so.0 (0x400f9000)
  librt.so.1 => /lib/librt.so.1 (0x4011a000)
  libm.so.6 => /lib/libm.so.6 (0x4012b000)
```

```
    libcrypt.so.1 => /lib/libcrypt.so.1 (0x4014c000)
    libnsl.so.1 => /lib/libnsl.so.1 (0x40179000)
    libpthread.so.0 => /lib/libpthread.so.0 (0x4018d000)
    libdl.so.2 => /lib/libdl.so.2 (0x401a1000)
    libc.so.6 => /lib/libc.so.6 (0x401a5000)
    /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

MacOs$ otool -L httpd
httpd:
/usr/lib/libiconv.2.dylib (compatibility ver 5.0.0, current ver 5.0.0)
/usr/lib/libresolv.9.dylib (compatibility ver 1.0.0, current ver 369.1.5)
/usr/lib/libSystem.B.dylib (compatibility ver 1.0.0, current ver 88.1.2)
```

Eine ausführliche Anleitung zum Thema chroot jail findet man unter http://penguin.triumf.ca/chroot.html.

Ein weiteres Problem ist die Nutzung von Ports und IP-Adressen. Möchte man wirklich mehrere Apache-Instanzen nebeneinander betreiben, müssen sich diese in IP und/oder Port unterscheiden.

Virtualisierungstechniken wie XEN oder OpenVZ erlauben ein noch grössere Trennung als das chroot Jail alleine, da diese das komplette System mit Ausnahme des Kernels separieren.

3.4 Apache 2.2 Series

3.4.1 Quellcode

In der Apache 2.2 Series wurden verschiedene Module umbenannt, aufgesplittet und frisch hinzugfügt. Für eine vollständige Übersicht sei auf den Upgrade Guide der Apache Group verwiesen - Upgrading to 2.2 from 2.0 [18].

Die neue Kompile-Directive besteht entsprechend aus folgenden Teilen:

```
$ ./configure --prefix=/opt/apache --enable-ssl \
  --with-ssl=/opt/openssl --enable-info --enable-rewrite \
  --disable-userdir --disable-imagemap --disable-autoindex \
  --disable-negotiation --disable-setenvif --disable-include \
  --enable-expires
```

Beginnend mit Apache 2.2 ist das Module mod_expires optional und muss während der Konfiguration mit –enable-expires aktiviert werden.

3.4.2 Konfiguration

3.4.2.1 /server-info

Die Darstellung wurde stark überarbeitet und feiner gegliert. In der Kopfnavigation erlauben Tabs einzelne Bereiche direkt anzuspringen.

3.4.3 Betrieb

<work in progress>

[18] http://httpd.apache.org/docs/2.2/upgrading.html

Kapitel 4

PHP

Als Basis für die nachfolgenden Ausführungen dient PHP in der Version 4.x. Änderungen, welche für die Version 5.x gelten, sind speziell ausgewiesen und separat aufgeführt.

4.1 Quellcode

4.1.1 Signatur

Bei aktivierter Option *expose_php = On* in der Konfigurationsdatei, fügt die PHP-Engine jeder Seite seine eigene Signatur hinzu, z.B. X-Powered-By: PHP/4.3.8. Diese Signatur macht möglichen Eindringlingen auf versionsbedingte Sicherheitslücken aufmerksam.

Aktuellen Header-Informationen liefert folgende Befehlssequenz:

```
$ telnet localhost 80
Trying 127.0.0.1...
Connected to localhost
Escape character is '^]'.
GET / HTTP/1.1
HOST: localdomain
HTTP/1.1 200 OK
Server: Ap4che
X-Powered-By: PHP/4.3.9
Transfer-Encoding: chunked
Content-Type: text/html; charset=ISO-8859-1
```

Mit Verfremdung der Signatur lässt sich die Sicherheit erhöhen. Hier am Beispiel der Buchstabenkombination «I» «-» «I» anstellen des Grossbuchstabens H. Das Aussehen der Signature definiert die Datei *main/SAPI.h*

```
#define SAPI_PHP_VERSION_HEADER "X-Powered-By: PI-IP/" PHP_VERSION
```

Die Versionsnummer wird an sich definiert das im Script *configure*. Wie beim Apache wird hier anstelle der originalen Zahlen die Ziffern neun geschrieben.

```
MAJOR_VERSION=9
MINOR_VERSION=9
RELEASE_VERSION=9
EXTRA_VERSION=""
VERSION="$MAJOR_VERSION.$MINOR_VERSION.$RELEASE_VERSION$EXTRA_VERSION"
```

Nicht alle PHP-Applikationen verkraften diese radikale Versionsänderung. Falls Applikationen versionsspezifische Aufrufe tätigen, laufen diese ins leere bzw. generieren Fehlermeldungen. Eine moderatere Verfremdung wäre

```
MAJOR_VERSION=4
MINOR_VERSION=9
RELEASE_VERSION=9
```

Damit bleibt zumindest die Hauptversionsnummer bestehen.

4.1.2 Kompilation

Die dargestellte Optionen konfigurieren PHP als Apache Handler (Server API = Apache 2.0 Handler). Dergestalt installiert, arbeiten PHP-Scripts mit dem Serviceaccount des aktuell laufenden Apache Servers. Bei grösseren Hostinginstallation oder bei besonderen Ansprüchen kann PHP aber auch als CGI-Applikation (Server API = CGI) installiert werden. Letzeres erlaubt im Zusammenspiel mit anderen Modulen (z.B. mod_suphp), dass jede PHP Applikation unter einem eigenen Benutzeraccount läuft.

PHP in der CGI-Konfiguration wird in dieser Dokumentation nicht behandelt.

Die Standardeinstellungen erlauben der PHP-Engine beliebig viel Speicher zu reservieren. Dies kann im schlimmsten Fall zu einem Denial-Of-Service führen, wenn aller verfügbarer Speicher aufgebraucht ist. Es ist deshalb sinnvoll, den maximalen Speicherverbrauch zu limitieren. Damit ergebt sich folgende Liste wichtiger Parameter für die Kompilation:

```
--enable-memory-limit
--enable-safe-mode
--with-openssl=/opt/openssl
--with-zlib
--with-bz2=/path
--enable-calendar
--with-jpeg-dir=/path
--with-tiff-dir=/path
--with-gd=/path
--with-png-dir=/path
--with-freetype-dir=/path
```

Eine reale Konfiguration könnte somit wie folgt aussehen:

```
$ ./configure --prefix=/opt/php-4.3.9 --without-mysql --without-pgsql \
   --with-apxs2=/opt/apache/bin/apxs --enable-module=so \
   --with-openssl=/opt/openssl --enable-calendar --enable-memory-limit
```

Späte Versionen der 4.x Series haben die Option –enable-memory-limit aktiv. Erst mit Version 5.x Series wird die Option als entsprechend ungültig deklariert.

4.1.3 Optimized build

Spezielle Compilereinstellungen erhöhen die Verarbeitungsgeschwindigkeit in Abhängigkeit der verwendeten Systemarchitektur und Prozessors.

```
$ CFLAGS="-march=athlon -O3 -pipe -fomit-frame-pointer -prefer-non-pic"
$ export CFLAGS
$ ./configure ... --disable-ipv6 --enable-inline-optimization --disable-debug
```

-march= Systemarchitektur, hier AMD Athlon. Weitere Parameter sind in der GCC Dokumentation[1] aufgelistet.. Unter Linux zeigt # *cat /proc/cpuinfo* die aktuellen Prozessorinformationen.

[1] http://gcc.gnu.org

-O3 Hoher Optimierungsgrade des Binärcodes

-pipe Optimiert den GCC Build Prozess - keine eigentliche Auswirkungen auf das Laufzeitverhalten

-fomit-frame-pointer Optimiert die Stackverwaltung

-prefer-non-pic Optimiert das Zusammenspiel des PHP Modul mit Apache Webservice auf Linux x86 Systemen.

–disable-ipv6 Deaktiviert IP v6 Support

–enable-inline-optimization Aktiviert das Inlining von Funktionen

–disable-debug Deaktiviert die Debugmöglichkeit

4.2 Konfiguration

PHP bietet Einstellungen, die die Sicherheit einer Installation erhöhen können.

register_globals: Aktiviert die Funktionalität, welche per GET oder POST übergebene Parameter automatisch als Variablen deklariert

display_errors: Deaktiviert die Anzeige von Fehlermeldungen Browser. Empfehlenswert, da in Extremfällen der DB-Connectstring inklusive Benutzername und Kennwort angezeigt werden könnte.

log_errors: Im Zusammenspiel mit display_errors lassen sich Fehlermeldung in einer Datei erfassen und für die Entwickler aufbereiten.

safe_mode: Bei aktiviertem Safe_Mode testet PHP beim Öffnen von Dateien, ob die User-ID (UID) mit jener von Apache übereinstimmt. Möchte man nur ein Group-ID (GID) vergleich, ist zusätzlich die Option safe_mod_gid einzuschalten.

safe_mode_gid: Bei aktiviertem Safe_Mode_Gid testet PHP beim Öffnen von Dateien, ob die Group-ID (GID) mit jener von Apache übereinstimmt.

open_basedir: Open_basedir ist vergleichbar mit der CHROOT Umgebung in Linux. Der hier gesetzte Pfad ist bindend, sprich PHP kann nicht ausserhalb dieser Verzeichnisse zugreifen. Ideal für die Trennung virtueller Server.

upload_tmp_dir: Definiert das temporäre Verzeichnis, in welchem Files während des Uploadvorganges zwischengespeichert werden. Ebenfalls ideal für die Trennung virtueller Server, da ohne gesetztes Verzeichnis alle virtuellen Server gemeinsam das Standardverzeichnis /tmp nutzen.

session.save_path: Pfandangabe für die temporären Sessionsfiles. Ohne Angabe werden auch diese in /tmp abgelegt und für alle Rechnernutzern sichtbar.

allow_url_fopen: Aktiviert, erlaubt es die Angaben von http://... und ftp://.. anstelle von Pfadangaben.

disable_functions: Durch Komma getrennte Liste von Funktionsnamen, die die PHP-Engine für die Verwendung sperrt.

disable_classes: Pendant zu disable_functions aber für die bereitgestellten Klassen

expose_php: Deaktiviert die Headerzeile X-Powered-By: PHP/x.y.z

4.3 Betrieb

4.3.1 memory_limit

Memory_limit spezifiziert die maximale Speichermenge, die ein Script nutzen kann. Sie verhindert bei Programmierfehlern oder provozierter, massiver Speichernutzung den kompletten Verbrauch der Serverressouncen.

Den aktuellen Speicherverbrauch eines Scripts liest die Funktion memory_get_usage() aus.

```
<html>
 <body>
  Benutzer Speicher <?php echo memory_get_usage(); ?> bytes
 </body>
</html>
```

Bei Dateiuploads kann ein gesetztes Limit aber auch zu Problemen führen.

4.3.2 upload_tmp_dir

Die Standardeinstellung legt temporäre Dateien - zum Beispiel beim Dateiupload - ins globale, temporäre Verzeichnis des Systems ab. Dies ist nicht immer erwünscht. Die Einstellung upload_tmp_dir regelt den Ablageort.

Mit etwas PHP-Code aus dem Onlinemanual lässt sich diese Funktion überprüfen und kontrollieren.

```
<?php
  echo "received files ";
  print_r($_FILES);
  echo "<p>";

  $uploaddir = "/opt/www/domainname.tld/upload/";
  $uploadfile = $uploaddir . basename($_FILES['userfile']['name']);

  if ( move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile)) {
   echo "upload successfully<p>";
  } else {
   echo "upload failed<p>";
  }
?>
<form enctype="multipart/form-data" action="/php-upload.php" method="POST">
  <input type="hidden" name="MAX_FILE_SIZE" value="2097152" > upload file
  <input name="userfile" type="file" />
  <input type="submit"   value="upload">
</form>
```

Der Erstaufruf des Scripts verlangt noch einer Datei für den Upload.

Abbildung 4.1: Upload test sample - upload failed

Hier wird die Datei apache.png via upload_tmp_dir des Services in den Zielordner /opt/www/domainname.tld/upload geladen.

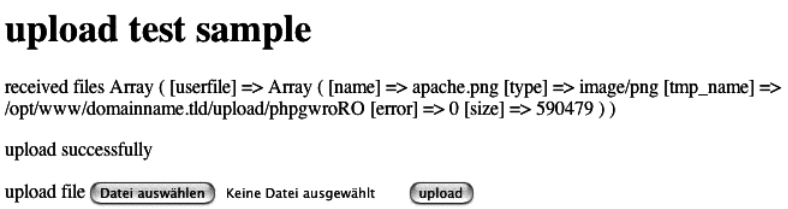


Abbildung 4.2: Upload test sample - upload successfully

Falls mal das Dateiupload partout nicht im spezifizierten Verzeichnis landen will, könnte unter Umständen fehlende Schreibrechte für den Serviceaccount nobody die Ursache sein.

4.3.3 session.save_path

Arbeitet ein Webdeveloppper mit Sessions, verwaltet die PHP-Engine pro Benutzer Verbindungsdaten. Diese Daten landen in der Regel als Datei im temporären Verzeichnis /tmp. Aus Gründen der Sicherheit, sollten diese Dateien nicht für alle einsehbar sein, denn wird eine Session nicht explizit per PHP zerstört, bleiben diese im Verzeichnis zurück.

Session.save_path spezifiziert den Pfad, in welche die Sessiondateien abgelegt werden, z.B. /opt/www/vhost/temp.

session_start()

```
<?php session_start();
      $sessionok = "yes";
      session_register("sessionok");
?>
```

session_unset() & session_destroy()

```
<?php session_unset();
      session_destroy();
?>
```

Sessiondatei

Erstellte temporäre Sessiondatei.

```
# ls -l
-rw------- 1 nobody nogroup sess_af56820722ea33194d7c9d21e3ac9f29
# cat sess_af56820722ea33194d7c9d21e3ac9f29
sessionok|s:3:"yes";
```

Beendet man die Session mit session_destroy(), wird diese temporäre Datei auch wieder gelöscht. Ohne dies, bleibt die Datei liegen und muss manuell mit einem Löschjob entfernt werden.

4.3.4 session.cookie_httponly

Diese Option markiert die Verwendung von Cookie Daten als «http only». Scripting Sprachen, zum Beispiel JavaScript, haben somit keine Zugriff auf die Daten mehr. Dies vermindert das Risiko von XSS Attacken aufgrund Cookie Diebstahl.

```
php_admin_value session.cookie_httponly true
```

4.3.5 open_basedir

Open_basedir spezifiziert eine Liste von Verzeichnissen, die die PHP-Engine und dessen Dateifunktionen beschränken. Greift man auf ein Verzeichnis ausserhalb dieser Liste zu, führt dies zu einem PHP-Fehler.

Im Beispiel ist das PHP-Script auf das Verzeichnis /opt/www/vhost/www und dessen Unterverzeichnisse eingeschränkt. Ein Zugriff auf /opt wird entsprechen mit folgender Meldung quitiert.

```
Warning: opendir(): open_basedir restriction in effect.
File(/opt) is not within the allowed path(s): (/opt/www/vhost/www)
in /opt/www/vhost/www/filebrowser.php on line 43
```

Ist diese Option nicht gesetzt, kann PHP auf alle Verzeichnisse und Dateien zugreifen, welche für den Apache Serviceaccount zugänglich sind.

Abbildung 4.3: Directory Listing sample

Das Testprogramm findet sich im Anhang.

4.3.6 disable_functions

Einige PHP-Funktionen erlauben sicherheitskritische Operationen. Die Liste disable_functions kann die Nutzung nun gezielt einschränken. Die per Komma getrennte Liste spezifiziert jene zu sperrenden Funktionen.

```
disable_functions exec,passthru,system,shell_exec,
                  escapeshellcmd,popen,proc_open,proc_nice,ini_restore,dl
```

Die Liste lässt sich bequem pro PHP-Applikation einstellen. Mehr hierzu findet man auf dem Heise Ticker[2] vom 2. Juni 2006.

exec Execute an external program, returns the last line of the execute program

passthru Execute an external program, returns alls data of the executed program

system Execute an external progam and display the output

shell_exec Execute command via shell and returns the complete output

escacpeshellcmd Escape shell metacharacters

popen Opens a pipe to a process executed by forking the command given (unidirectional)

[2]http://www.heise.de/newsticker/meldung/73837q

proc_open Opens a pipe to a process excuted by forking the command given (two-way / input-output)

proc_nice Change the priority of a current, running process

ini_restore Restore the value of a configuration option back to is original value

dl Loads a PHP extension at runtime

4.3.7 default_charset

Nicht unbedingt eine sicherheitsrelevante Einstellung, aber trotzdem von Zeit zu Zeit nützlich.

```
php_admin_value default_charset "utf-8"
```

Setzt den Standardzeichensatz von PHP auf UTF-8. Besonders hilfreich bei Applikationen, bei denen UTF-8 erst als Metatag im HTML-Code spezifizieren und dementsprechen Datenbankinhalte beim Rendern bereits falsch interpretieren.

4.3.8 Restriktive Beispielkonfiguration

Als Zusammenstellung der obigen Punkte man man folgende sehr restriktive Beispielkonfiguration von PHP.ini sehen. Mehr hierzu ist in der Zeitschrift c'T Ausgabe 18/2007 auf Seite 178 im Artikel «Serverfrieden - PHP-Anwendungen indivuduell absichern» zu finden.

```
[PHP]
register_globals = off
allow_url_fopen = off
safe_mode = on
open_basedir = <Pfadname des Weberzeichnisses>
disable_functions = exec,system,passthru,shell_exec,
        escapeshellcmd,proc_open,proc_nice,ini_restore,popen
display_errors = off
```

4.4 PHP Coding

4.4.1 register_globals=off

Register_globals regelt, ob in einem Script die per POST oder GET übergebenen Parameter automatisch als Variablen in PHP zur Verfügung stehen oder nicht. Seit PHP v4.1 steht diese Option Default auf off.

Der Programmierer ist somit angehalten, alle Variablen selber zu deklarieren und sauber zu initalisieren.

Hier ein Beispiel

```
<?php
  echo "register_globals=on Style variable1 - $frm_variable1 <br>";
  echo "register_globals=on Style variable2 - $frm_variable2 <br>";

  $variable1 = "(none)";
  if ( array_key_exists( 'frm_variable1', $_GET)) {
   $variable1 = trim( $_GET[ 'frm_variable1' ]);
  }
  echo "register_globals=off Style variable1 - $variable1<br>";
?>
<form action="/register-globals.php" method="get">
  variable1
  <input name="frm_variable1" type="text" size="20" maxlength="20">
```

```
  <input class="button" type="submit" value="submit">
</form>
```

Bei aktivierter Option register_globals werden alle übergebenen Parameter - egal ob per GET oder per POST - automatisch als Variablen in PHP eröffnet und mit den angegebenen Werten initialisiert.

Abbildung 4.4: register_globals=on

Deaktiviert man die Option und testet den Beispielcode erneunt, wird nur noch die sauber codierte Variable variable1 initalisiert und mit dem Parameterwert gefüllt.

Abbildung 4.5: register_globals=off

Bei älteren Applikationen wird diese Option oft benötigt. Best Practice hierbei ist, diese Applikation in einem eigenen virtuellen Webserver zu betreiben und nur für diese eine Applikation register_globals=on zu setzen.

4.4.2 sql injection

SQL injection beschreibt den Vorgang, bei dem anstelle des erwarteten Wertes oder Textes vollständiger und gültiger SQL-Code eingegeben wird. Dieser SQL-Code wird zusätzlich zum eigentlich programmierten Programmcode ausgeführt.

Hier ein Beispiel mit HTML- und entsprechendem SQL-Code.

```
<form action="/mylogon.php" method="post">
  Userid
  <input name="name" type="text" size="20" maxlength="20" />
  Password
  <input name="kennwort" type="password" size="20" maxlength="20" />
  <input class="button" type="submit" value="logon" />
</form>
```

Und der dazu gehörende PHP-Code.

```
if ( array_key_exists( 'name', $_POST)) {
  $benutzername = trim( $_POST[ 'name' ]);
}
if ( array_key_exists( 'kennwort', $_POST)) {
  $kennwort = trim( $_POST[ 'kennwort' ]);
}
$mysql = "select intuserid from tblusers \
   where vchrname like '$benutzername' and vchrpwd like '$kennwort';";
```

Ein findiger Benutzer könnte nun folgendes ausprobieren. Anstelle des Passwortes gibt er folgenden Text ein.

```
foobar'; delete * from tblusers; --
```

Foobar ist ein willkürlich gewählter Text und kann auch weggelassen werden, was einem leeren Passwort entspricht. Diese Eingabe wird direkt vom Angreifer mit '; abgeschlossen. Damit entsteht ein gültiger SQL -Code.

Der Weg ist nun frei, um einen oder mehrere eigene SQL-Befehle einzubringen - die eigentliche SQL-Injection beginnt. Hier im Beispiel wird versucht, alle Benutzer aus der Tabelle zu löschen. Falls das Delete Recht gesetzt wurde, könnte sich anschliessend niemand mehr anmelden.

Der Abschluss mit – zeigt an, dass der nachfolgende Text, den der Programmierer eigentlich ausführen wollte, als Kommentar zu behandeln ist. Damit wird keine Fehlermeldung aufgrund des unvollständigen SQL-Kommandos '; angezeigt.

Eine Quick&Dirty Abwehr gegen solche Angriffe ist einfach zu realisieren. Der Ansatz besteht darin, das Manipulieren des eigentlich gedachten SQL-Codes zu unterbinden. Aufbauend auf dem Beispiel müsste man also die Zeichen ' (Hochkomma), ; (Strichpunkt) und – (Kommentar) unterbinden.

```
$benutzername = str_replace( ";", "", $benutzername);
$benutzername = str_replace( "'", "", $benutzername);
$benutzername = str_replace( "--","", $benutzername);
$kennwort = str_replace( ";", "", $kennwort);
$kennwort = str_replace( "'", "", $kennwort);
$kennwort = str_replace( "--","", $kennwort);
```

Effizientere und sicherere Abwehrmechanismen sind in der Nutzung von Prepared SQL Statements im PHP oder die Auslagerung sensitiver Aktionen in SQL Stored procedures.

4.5 PHP 5.x Series

4.5.1 Quellcode

Bei der PHP v5 Series ist der Einsatz der Libxml2 Komponente empfohlen. Insbesondere bei älteren Grundsystemen fehlt diese und muss vorgängig installiert werden.

```
$ wget ftp://xmlsoft.org/libxml2/libxml2-sources-2.6.30.tar.gz
```

Vor der eigentlichen PHP Kompilation sind entsprechend die Libxml2 Pfade und Konfigurationen anzugeben.

```
export PKG_CONFIG_PATH=/opt/libxml2-2.6.30/lib/pkgconfig:$PKG_CONFIG_PATH
```

und

```
--with-libxml-dir=/opt/libxml2-2.6.30
```

Auf älteren System muss zudem damit gerechnet werden, dass die PThread Unterstützung nicht vollständig ist. Bei der Libxml2 Konfiguration ist die Threadunterstützung zu deaktivieren.

```
--without-threads
```

Mit PHP v5 sind die Optionen –enable-module=so und –enable-memory-limit nicht mehr notwendig.

4.5.2 Konfiguration

Beginnend mit PHP Version 5.2 steht zusätzlich die Option *allow_url_include* für die Steuerung von *include()* und *require()* Befehlen zur Verfügung.

Anhang A

GNU Free Documentation License

Version 1.2, November 2002

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed

as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least

five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already

includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Anhang B

Beispielumgebung

B.1 Programmliste

Zum Einsatz kommen UNIX (MacOsX und Linux[1]) als Betriebssysteme, Apache[2] als Webservice und PHP[3] als Scriptsprache.

B.2 mögliche Verzeichnisstruktur

```
/opt
  |- src
  |- apache-2.0.52
  |- php-4.3.9
  |- openssl-0.9.7e
  \- www
      |- default
      |   |- logs
      |   |- scripts
      |   |- www
      |   \- temp
      |- vhost.tld
      |   |- logs
      |   |- scripts
      |   |- www
      |   \- temp
      \- default
          |- logs
          |- scripts
          |- www
          \- temp
```

In den temp Verzeichnissen benötigt der Benutzer *nobody* Schreibrechte, um Uploads und/oder Sessiondateien abzulegen.

Des weiteren wurden folgende symbolische Links erstellt.

```
# ln -s /opt/apache-2.0.52 /opt/apache
# ln -s /opt/php-4.3.9 /opt/php
# ln -s /opt/openssl-0.9.7e /opt/openssl
```

[1] http://www.linuxhq.org
[2] http://httpd.apache.org
[3] http://php.net

Anhang C

Betriebssystem Linux

C.1 Bash Coding - Symlink Code

```
#!/bin/bash
#
# Code snippet which tests a given directory name
# if it is a symlink or not
#
# v1.1 aha isSymlink as function with return value
function isSymlink {
 if [ -L $1 ] ; then
    true
  else
    false
  fi
}
testdir1=$HOME/symlink
testdir2=$HOME/original
startpwd=`pwd`
echo `pwd`
cd $testdir1
echo `pwd`
cat testfile
cd $startpwd
echo `pwd`
cd $testdir2
echo `pwd`
cat testfile
# check directory entries
if { isSymlink $testdir1; } then
    echo "$testdir1 is a symlink"
  else
    echo "$testdir1 is NOT a symlink"
fi
if { isSymlink $testdir2; } then
    echo "$testdir2 is a symlink"
  else
    echo "$testdir2 is NOT a symlink"
fi
# Scriptende
```

C.2 Backupscript

```
#!/bin/bash
#
#####################################################################

mydir=/home/backup
mytardir=tar

myfile=`hostname`-system-`date+%Y%m%d`.tar
myalldb=`hostname`-db-all-`date +%Y%m%d`.sql
mytargz=`hostname`-full-`date+%Y%m%d`.tar.gz

#####################################################################

# Fix access rights
umask -S u=rw,g=,o= > /dev/null 2> /dev/null

cd $mydir

# Loesche die alten Backups
rm -f $mydir/*.tar.gz
rm -f $mydir/*.bz2
rm -rf $mydir/$mytardir

mkdir $mytardir

#####################################################################
echo Systembackup - starting - `date`

tar cf $mydir/$mytardir/$myfile --exclude=$mydir/$mytardir \
    --exclude=/dev --exclude=/sys --exclude=/proc --exclude=/mnt \
    --exclude=/tmp /*

echo Systembackup - finished - `date`

echo DB dump - starting -`date`

mydbrunning=`ps ax | grep "mysqld" | wc -l`
# z.T. wird eine Abfrage ebenfalls mitgezählt...
if [ "$mydbrunning" = "0" ] || [ "$mydbrunning" = "1" ] then
  echo "..no db running.."
else
  echo "..db dump in progress..."
  /opt/mysql/bin/mysqldump --user=USERNAME --password=PASSWORD \
  --all-databases --opt --result-file=$mydir/$mytardir/$myalldb
fi
echo DB dump - finished - `date`

echo GZIP - starting - `date`
tar czf $mydir/$mytargz $mydir/$mytardir
chown backup $mydir/$mytargz
echo GZIP - finished-`date`

#####################################################################

cd $mydir
```

```
# Loesche die alten Backups mit Ausnahme des tar.gz!
rm -rf $mydir/$mytardir

#####################################################################
```

Anhang D

Apache

D.1 sign.sh

```
#!/bin/sh
##
## sign.sh -- Sign a SSL Certificate Request (CSR)
## Copyright (c) 1998-2001 Ralf S. Engelschall, All Rights Reserved.
##
##  20130930 hammerar, replaced MD5 by SHA1 as digest algorithm
# argument line handling
CSR=$1
if [ $# -ne 1 ]; then
  echo "Usage: sign.sign <whatever>.csr"; exit 1
fi
if [ ! -f $CSR ]; then
  echo "CSR not found: $CSR"; exit 1
fi
case $CSR in
  *.csr ) CERT="`echo $CSR | sed -e 's/\.csr/.crt/'`" ;;
  * ) CERT="$CSR.crt" ;;
esac
# make sure environment exists
if [ ! -d ca.db.certs ]; then
  mkdir ca.db.certs
fi
if [ ! -f ca.db.serial ]; then
  echo '01' >ca.db.serial
fi
if [ ! -f ca.db.index ]; then
  cp /dev/null ca.db.index
fi
# create an own SSLeay config
cat >ca.config <<EOT
[ ca ]
default_ca = CA_own
[ CA_own ]
dir = .
certs = \$dir
new_certs_dir = \$dir/ca.db.certs
database = \$dir/ca.db.index
serial = \$dir/ca.db.serial
RANDFILE = \$dir/ca.db.rand
certificate = \$dir/ca.crt
private_key = \$dir/ca.key
```

```
default_days = 1095
default_crl_days = 30
#default_md = md5
default_md = sha1
preserve = no
policy = policy_anything
[ policy_anything ]
countryName = optional
stateOrProvinceName = optional
localityName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
EOT
# sign the certificate
echo "CA signing: $CSR -> $CERT:"
openssl ca -config ca.config -out $CERT -infiles $CSR
echo "CA verifying: $CERT <-> CA cert"
openssl verify -CAfile ca.crt $CERT
# cleanup after SSLeay
rm -f ca.config
rm -f ca.db.serial.old
rm -f ca.db.index.old
# die gracefully
exit 0
```

Anhang E

PHP

E.1 Filebrowser

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>

<head>
  <title>Filebrowser</title>
</head>

<body>
<?php
  if ( isset( $_GET['path'] )) {
   // Pfad einlesen
   $path = $_GET['path'];
   // Datei?
   if ( is_file($path) ) {
     echo "Oeffne " . $path . "<br>\n";
     $file = fopen( $path,"r");
     echo "<pre>\n";
     while (!feof($file)) {
       $zeile = fgets( $file, 4096);
       echo htmlentities( $zeile, ENT_QUOTES);
     } // End of While
     echo "</pre>\n";
     fclose( $file);
   } else {
     echo "<p>\n";
     echo "Inhalt von $path\n<br>\n";
     echo "<pre>\n";

     $dir = opendir( $path);
     while ($file = readdir($dir)) {
       $filepath = $path . "/" . $file;
       if ( is_dir( $filepath)) echo "[DIR ] ";
       if ( is_file($filepath)) echo "[FILE] ";
       //if ( is_link($filepath)) echo "[LINK] ";

       if ( $file == ".") {
         echo "<a href=\"filebrowser.php?path=$path\">.</a><br>";
         continue;
       }
       if ( $file == "..") {
```

```
          if ( substr( $path, 0, strrpos($path,"/")) == "") {
            echo "<a href=\"filebrowser.php?path=/\">..</a><br>";
          } else {
            echo "<a href=\"filebrowser.php?path=" .
            substr($path,0,strrpos($path,"/")) .
            "\">..</a><br>";
          }
          continue;
        } // End of $file ==..
        echo "<a href=\"filebrowser.php?path=" .
            (($path=="/") ? "" : $path) . "/" .
            rawurlencode($file) . "\">$file</a>";
        echo "\n";
      } // End of While
      closedir($dir);
      echo "</pre>\n";
    } // End of Datei?
    // End of isset....
  }
?>

<p>
<form action="filebrowser.php" method="get">
Pfadangabe -
<input type="text" name="path">
-
<input type="submit" value="anzeigen">
</form>

</body>
</html>
```

Anhang F

Abbildungsverzeichnis

Abbildungsverzeichnis

Anhang G

Quellenverzeichnis

Weitere, bis anhin nicht spezifisch angegebene Quellen sind...

http://www.us-cert.gov The *United States Computer Emergency Readiness Team* (US-CERT) is a partnership between the Department of Homeland Security and the public and private sectors. Established in 2003 to protect the nation's Internet infrastructure, US-CERT coordinates defense against and responses to cyber attacks across the nation.

http://www.sans.org *SANS* is the most trusted and by far the largest source for information security training and certification in the world. It also develops, maintains, and makes available at no cost, the largest collection of research documents about various aspects of information security, and it operates the Internet's early warning system - Internet Storm Center.

http://isc.sans.org The *Internet Storm Center* was created in 2001 following the successful detection, analysis, and widespread warning of the Li0n worm. Today, the ISC provides a free analysis and warning service to thousands of Internet users and organizations, and is actively working with Internet Service Providers to fight back against the most malicious attackers.

http://securityspace.com SecuritySpace.com is proudly brought to you by E-Soft Inc., a privately owned Canadian consulting firm, with proven expertise in internet security and on-line services. We specialize in the following areas: on-line network security auditing services, network monitoring services, Internet Research reporting, integrated web solutions and application development for secure data transactions.

http://www.heise.de/security Online Security Portal des Heise Zeitschriften Verlag GmbH & Co. KG in Deutschland.