
# Mac OS X Consoliero

Weiterführende Dokumentationen für Administratoren.

# OS X Absichern

Christoph Müller, PTS

# Mac OS X Consoliero:
# Der vi-Editor

## Inhaltsverzeichnis


1 Mac OS X Client absichern ....3
1.1 Mac OS X Standard Sicherheits-Features ....4
1.1.1 Secure Network Service....4
1.1.2 Der Schlüsselbund ....5
2 Die Echtheit von System- und Sicherheits-Updates prüfen ....6
3 File Vault ....7
3.1 File Vault konfigurieren....8
4 Das Verschlüsseln von Dateien und Ordner ....10
5 Anpassen von umask....12
6 Systemkonten absichern....14
6.1 Den Benutzerordner des Administrators abschließen ....14
6.2 Administrative Benutzerkonten ausblenden ....15
6.3 Den root-Benutzer sichern ....16
6.4 sudo benutzen....17
6.5 Den „single user boot" absichern ....19
6.6 Sicherheitshinweise beim Anmelden ....24
6.7 Unsichere Hardware Komponenten entfernen ....26

**Mac OS X Consoliero:**
# Mac OS X Client absichern

## Einleitung

Trotz der allgegenwärtigen Versprechen von Apple, das Mac OS X das sicherste Betriebssystem der Welt sei, lässt sich dieses sehr leicht aushebeln. Diese nicht so sehr in der Anwendung selber, sondern in der Art und Weise wie Mac OS X aufgebaut und konzipiert ist. Aus den Augen eines Windowsbenutzers ist Mac OS X ein offenes Buch und benötigt dringend ein Siegel. Wie dieses Siegel aussehen muss wird in diesem Consoliero erarbeitet.

Die Nummerierung der Bilder scheint ein bisschen komisch. Dies rührt daher, dass dieser Text ein Teil eines Kapitels aus meinem neuen Buch ist. Welches allerdings noch nicht erschienen ist.

Christoph Müller, www.pts.ch

## Konventionen

Wenn im Text ein **^X** angezeigt wird, bedeutet das einen so genannten „controll character". Eingegeben wird dieser mit „ctrl" + „X" Taste. Befehle sind in Courier und **Fett** gehalten. Also in etwa:

```
vi testfile.txt
```

Ausgaben des Terminal werden in Courier gehalten, werden aber nicht fett gedruckt.

```
tcsh: was: Command not found
```

Pfade `/Library/Preferneces` innerhalb des Fliesstextes werden ebenfalls in Courier gehalten.

## 1 Mac OS X Client absichern

Bekanntlich ist der wichtigste Teil von Mac OS X auf BSD Unix aufgebaut. Der Kern des Betriebssystems (Darwin) besteht aus zwei Teilen: dem Mach Kernel und dem BSD Subsystem.

Der Mach Kernel und das BSD Subsystem bieten Sicherheitsfunktionen, welche unter Mac OS 9 nicht möglich waren. Der Mach Kernel steuert alle Interaktionen des Systems mit der Hardware. Ebenso stellt der Mach Kernel die Kontrolle und die Sicherheit über die auf dem System laufenden Programme sicher. Zusätzlich bietet der Mach Kernel eine moderne virtuelle Speicherverwaltung, die jedem Prozess seinen eigenen Speicherbereich sichert. Programme können nun nicht mehr in den reservierten Speicherbereich des Systems zugreifen und so Abstürze des Rechners provozieren.

Das BSD Subsystem bietet eine Mehrbenutzer Umgebung, in der jeder Benutzer seine eigene Benutzer Identität hat (ID). Benutzer können zudem Mitglieder in Gruppen sein und so Zugriff zu Dateien bekommen. Mit diesem Modell steuert das System Zugriff auf

systemrelevante Dateien, so dass nur autorisierte Benutzer Zugriff auf die Konfiguration des Systems, oder auf Prozesse haben.

Wie die meisten Unix basierte Betriebssysteme hat auch Mac OS X einen Super-User, oder root-Benutzer. Dieser hat auf dem System alle Rechte. Im Unterschied zu vielen Unix Derivaten wird dieser root-Benutzer bei Mac OS X standardmäßig deaktiviert. Für jede Aktion die mit root-Rechten durchgeführt werden sollte, muss der Administrator sein Login und das dazugehörende Passwort eingeben. Natürlich kann der Administrator den root-Benutzer auch aktivieren. Dies empfiehlt sich aber aus diversen Gründen nicht. Einer der wichtigsten Gründe ist sicher die fehlende Möglichkeit sich direkt als root-Benutzer am System anzumelden. Dadurch muss der Administrator von seinem Konto aus sich temporär als root-Benutzer ausgeben (sudo). Wenn der sudo Befehl benutzt wird, erzeugt das System einen Eintrag im system.log unter /var/log/system.log. Hier ein Beispiel:

```
Jan  7 23:17:35 localhost sudo:      pts : TTY=ttyp1 ; PWD=/Users/pts ;
USER=root ; COMMAND=/usr/sbin/diskutil list
```

Dadurch ist ersichtlich ob ein Systemfehler aus einer Fehlmanipulation oder einer böswilligen Aktion entstanden ist. Lassen Sie also auf Systemen welche sensiblen Daten enthalten und im Netz exponiert sind, den root-Benutzer deaktiviert.

### 1.1 Mac OS X Standard Sicherheits-Features

Die Standardkonfiguration ist eines der wichtigsten Sicherheits-Features von Mac OS X. Wie oben erwähnt, ist der root-Benutzer deaktiviert. Zweitens, alle Netzwerkdienste sind deaktiviert. Drittens ist das Setup der Berechtigungen der Benutzerkonten für die meisten Anwendungen anwendbar.

#### 1.1.1 Secure Network Service

Ein wichtiges Feature in Mac OS X ist das SSH-Protokoll, welches die älteren Programme wie `etwa telnet, rlogin, rcp und ftp` ersetzt. Diese Programme verschlüsselten weder den Benutzernamen, das Passwort und die Daten welche über das Netz geschickt wurden. SSH bietet hier die nötige Sicherheit in der Netzwerk-Kommunikation in dem es eine verschlüsselte Verbindung zwischen den Host aufbaut bevor Daten übertragen werden. Diese erste verschlüsselte Verbindung ist für den Benutzer weitgehend transparent.

| Unverschlüsselte Programme | SSH-Ersatz |
|---|---|
| `telnet` | `ssh` |
| `rlogin` | `slogin` |
| `rcp` | `scp` |
| `ftp` | `sftp` |

Ein Mac OS X System kann Client wie auch Server sein. Wenn nur der Client gebraucht wird, sind keine Modifikationen nötig. Wird ein SSH-Server gebraucht, kann dieser über das Sharing-Kontrollfeld in der Systemsteuerung durch aktivieren von „Entfernte Anmeldung" aktiviert werden (Abbildung 6.16). Der Gebrauch von unverschlüsselten Programmen ist für sensible Anwendungen eigentlich nicht mehr akzeptabel. Wenn keine sensiblen Daten verschoben werden, wie etwa ein FTP-Download mit anonymen FTP, spielt es natürlich keine Rolle.

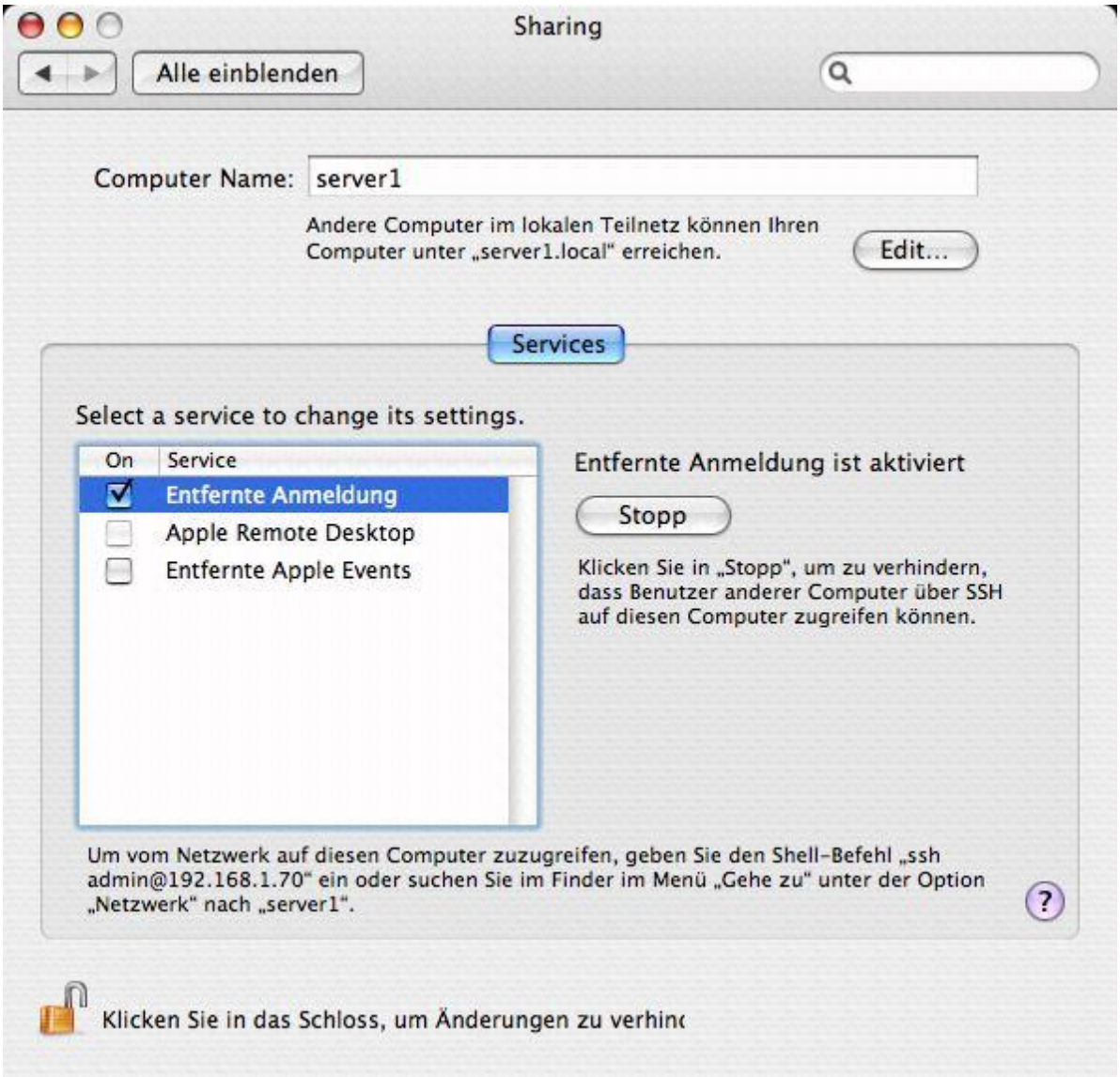

Abbildung 6.16 – Sharing Kontrollfeld

### 1.1.2 Der Schlüsselbund

Der Schlüsselbund (Abbildung 6.17) war schon in älteren Versionen von Mac OS verfügbar. In der Zwischenzeit ist dieses Feature verbessert und auch wichtiger für Mac OS X geworden. Der Schlüsselbund kann benutzt werden um diverse Passwörter und Zertifikate zu verwalten. Alle diese Schlüsselbunde werden zusätzlich zum Benutzerkennwort mit einem weiteren Kennwort versehen und verschlüsselt. Sie können erst wieder ausgelesen und eingesehen werden, nachdem die Erlaubnis und die nötigen Passwörter geliefert wurden. Jeder Eintrag im Schlüsselbund enthält eine einmalige „Access Control List“ (ACL) die spezifiziert, welche Programme autorisiert sind diese Schlüssel auszulesen. Zertifikate können ebenso im Schlüsselbund verwaltet werden und können so einfach benutzt werden um E-Mails digital zu signieren oder zu verschlüsseln. Wenn der Schlüsselbund korrekt konfiguriert ist, ist dieses Programm ein gutes Werkzeug um Benutzer Passwörter und Zertifikate zu handhaben. Zudem ist der

Schlüsselbund eine einfach zu handhabende Software, welche Passwörter und Zertifikate für Safari, Mail und andere Programme wie etwa Microsoft Entourage zur Verfügung stellt. Natürlich können auch Zertifikate von öffentlichen Zertifikatsstellen wie etwa VeriSign eingelesen und benutzt werden (Abbildung 6.18).

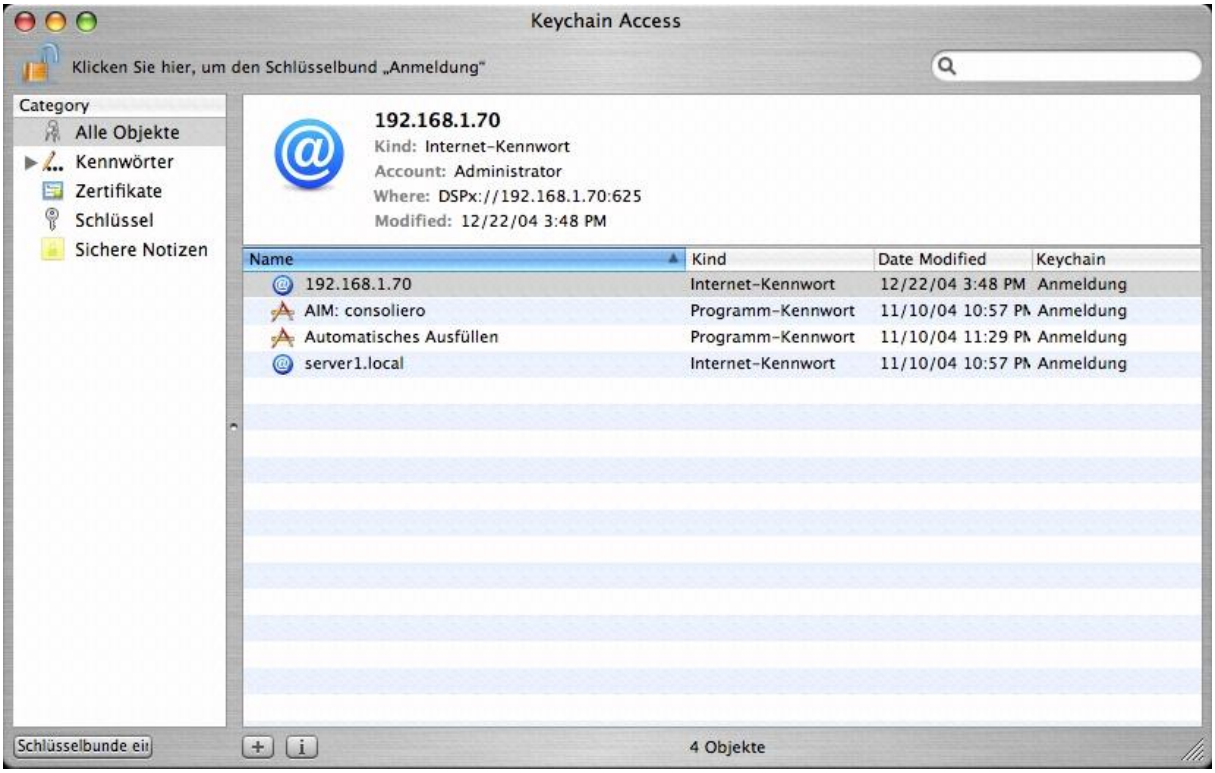

Abbildung 6.17 – Schlüsselbund

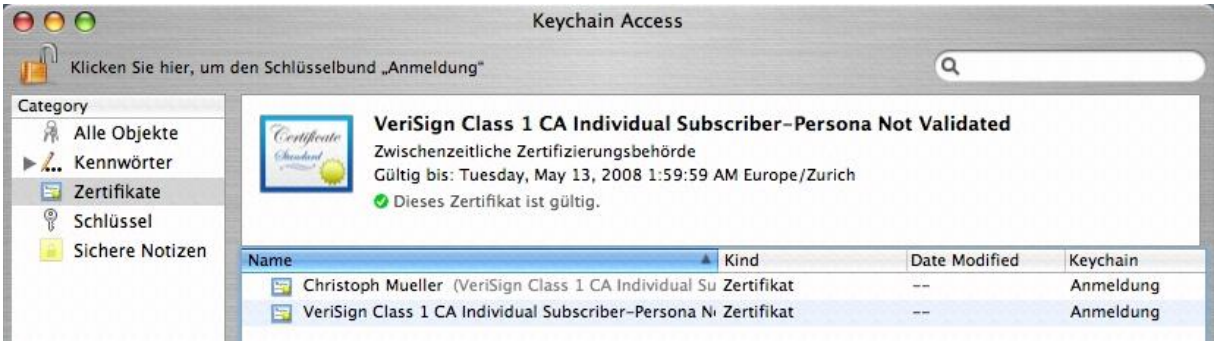

Abbildung 6.18 – VeriSign Zertifikat

## 2 Die Echtheit von System- und Sicherheits-Updates prüfen

Apple bietet über die Softwareupdate-Funktion des Betriebssystems eine automatische Benachrichtigung und Installation der Software an. Die meisten Benutzer von Mac OS X benutzen diese Funktion auch so. Allerdings ist dies nicht die sicherste Art sein System zu aktualisieren, da der Update-Server dem System vorgespiegelt sein kann, und so modifizierte Updates eingespielt werden können. Apple ist sich dem bewusst und bietet eine Überprüfung der Updates mittels einem SHA-1 Wert an. Um an sichere Updates zu kommen, geht man am besten wie folgt vor. Wenn Mac OS X einem ein neues

Systemupdate oder Sicherheitsupdate ankündigt, sucht man sich dieses auf der Apple Seite unter: http://www.apple.com/support/downloads/. Bei den Details zu dem gewünschten Update wird der entsprechende SHA-1 Wert angezeigt (Abbildung 6.19). Diesen notiert man sich, oder lässt die Webseite offen. Danach lädt man das Update von dieser Seite, oder via dem Befehl `sudo softwareupdate –d Updatename` auf den Computer herunter.

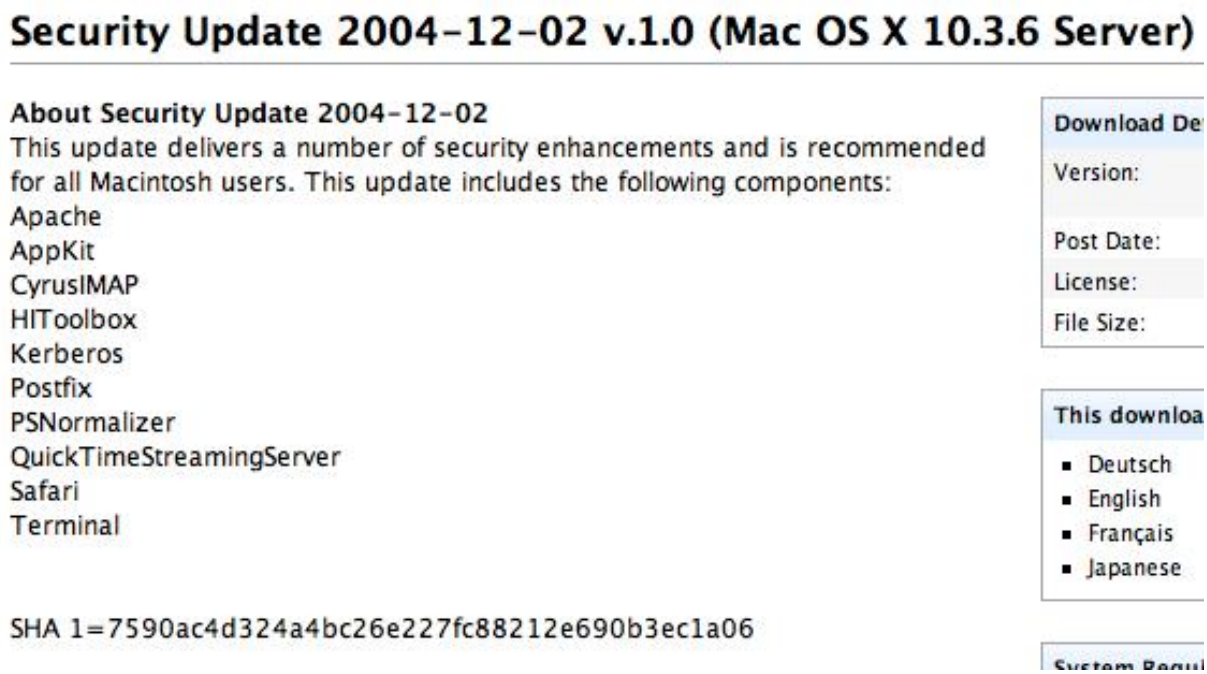


Abbildung 6.19 – SHA-1 Wert Zertifikat

Wenn der Download beendet ist, erhält man in der Regel ein Diskimage-File (.dmg). Dieses Image-File lässt sich nun anhand dem SHA-1 Wert auf der Apple Website auf seine Integrität überprüfen. Dazu öffnet man das Terminal und nutzt den Befehl openssl sha1. Mit einem Download auf den Desktop sieht das so aus:

```
/usr/bin/openssl sha1 /Users/pts/Desktop/SecUpdSrvr2004-12-02Pan.dmg

SHA1(/Users/pts/Desktop/SecUpdSrvr2004-12-02Pan.dmg)=
7590ac4d324a4bc26e227fc88212e690b3ec1a06
```

Nun muss man die Ausgabe des SHA-1 Wertes im Terminal mit dem Wert auf der Apple Webseite vergleichen. Sind die Werte identisch, kann man davon ausgehen, dass die herunter geladene Datei unverändert auf den Rechner gekommen ist. Nun kann man das Update mit „gutem Gewissen" installieren.

## 3 File Vault

In den Systemeinstellungen findet sich in der Reihe „persönlich" ein Kontrollfeld mit dem Titel „Sicherheit". Dieses interessiert uns in diesem Abschnitt natürlich besonders. In diesem Kontrollfeld wird die Funktion File Vault aktiviert. File Vault aktiviert eine Verschlüsselung des Benutzer Verzeichnisses. Im Gegensatz zu vielen Kontrollfeldern

steuert dieses Kontrollfeld die Einstellungen global über den Computer und die Einstellung für den im Moment angemeldeten Benutzer.

Die File Vault Funktion empfiehlt sich überall dort wo sensible Daten auf dem Rechner gespeichert sind, und die physische Sicherheit des Gerätes nicht garantiert werden kann. Wie etwa bei PowerBooks und iBooks. In einem solchen Fall sollte File Vault für alle Benutzerkonten aktiviert werden. Durch die Verschlüsselung werden alle Dateien innerhalb des Benutzer-Ordners verschlüsselt und können nicht mehr eingesehen werden. Der Nachteil dabei ist, dass der Schreib- und Lesezugriff verlangsamt werden. Harddisk intensive Aufgaben wie etwa Videobearbeitung sind mit aktiviertem File Vault nicht möglich. Dieses Problem hängt damit zusammen wie File Vault arbeitet. Wenn File Vault aktiviert wird, wird der gesamte Inhalt des Benutzer Ordners in ein verschlüsseltes Disk Image konvertiert. Die Daten werden danach in das Image geschrieben oder daraus gelesen. Dieser Vorgang kostet Rechenzeit welcher sich in der Performance bemerkbar macht. Zudem hat das Konvertieren einen weiteren Nachteil. Eine Datei die vor dem Konvertieren schon auf dem Rechner war, kann unter Umständen trotzdem noch ausgelesen werden. Gelöschte Daten existieren weiterhin auf der Harddisk, außer sie werden mit der Finderfunktion „Papierkorb sicher entleeren" gelöscht. Was allerdings nicht die Regel ist (Abbildung 6.20). Ansonsten können diese Daten von diversen Datenrettungs Tools wie Norton Disk Doktor noch ausgelesen werden. Unter Mac OS X 10.4 kann man jedoch beim Konvertieren wählen ob man die Daten welche konvertiert werden „sicher löschen" möchte.

Abbildung 6.20 – Sicheres Papierkorb entleeren

Aktivieren Sie also bei solchen sensiblen Geräten File Vault sofort und vor dem Einsatz, oder speichern Sie sensible Daten erst darauf, nachdem File Vault aktiviert wurde.

Zu beachten ist außerdem, dass File Vault die Daten nicht schützt wenn sie aus dem Benutzer Ordner über das Netzwerk verschoben werden oder auf einen Datenträger wie CD-ROM, oder einen USB-Stick kopiert werden. Mehr dazu später.

### 3.1 File Vault konfigurieren

Um File Vault zu aktivieren, muss zuerst ein Hauptkennwort festgelegt werden. Dieses Hauptkennwort kann alle verschlüsselten Benutzer Ordner entschlüsseln. Was nötig werden kann, wenn ein mobiler Benutzer zum Beispiel sein Passwort vergisst. Dieses Passwort schreibt man sich am besten nieder und legt es in einem Umschlag zu den

anderen essentiellen Passwörtern in den Save. Das Hauptkennwort sollte auf jeden Fall nicht identisch mit dem Administrator Passwort sein. Trotzdem sollte es den Richtlinien für ein Administratorkennwort genügen. Ein Administrator Passwort sollte mindesten 12 Zeichen beinhalten und vor allem Groß- und Kleinschreibung sowie Zeichen enthalten.

Ein guter Ort um die Stärke von Passwörtern zu ermitteln ist: https://passwortcheck.datenschutz.ch/. Achten Sie aber darauf keine echten Passwörter zu testen, sondern andere in etwa mit der gleichen Logik angefertigten Passwörter. Sie können auch beim Eingeben des Hauptkennwortes auf das Fragezeichen neben dem Passwortfeld klicken und Mac OS X bietet Ihnen ein Passwort an, oder bewertet Ihr Passwort (Abbildung 6.21). Die Bewertung ist allerdings etwas einfach. Basierend auf der Länge des Kennwortes verschiebt sich der Balken bereits in Richtung guter Qualität. Die Bewertung der Passwörter auf der Seite des Datenschützers des Kantons Zürich in der Schweiz geht hier wesentlich wissenschaftlicher vor.

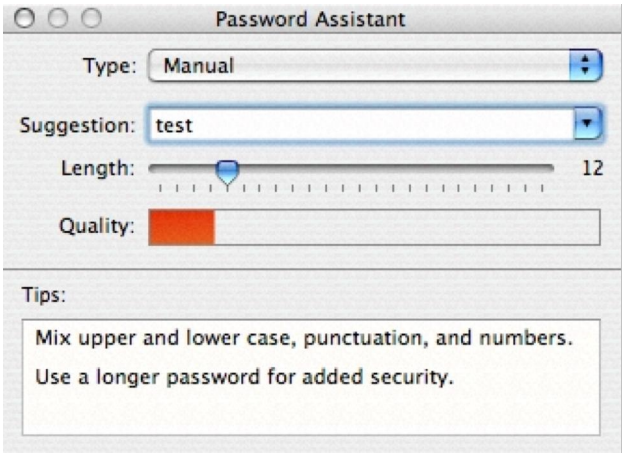


Abbildung 6.21 – Passwort Assistent (Bilder/6_5/picture6.jpg)

Wenn das Hauptkennwort gesetzt ist, kann man nun für den aktuell angemeldeten Benutzer über das gleiche Kontrollfeld mit der Taste „File Vault aktivieren" den Benutzer Ordner verschlüsseln. Ein Benutzer welcher keine lokalen Administrationsrechte hat, kann File Vault jedoch nicht selber aktivieren. Trotz dem Vorhandensein eines Hauptkennwortes muss zusätzlich ein Kennwort eines Administrators angegeben werden. Sie müssen also beim Bereitstellen des Computers bereits daran denken, File Vault zu aktivieren. Wenn File Vault für den Benutzer aktiv ist, lässt sich das an einem neuen Benutzer Ordner Symbol erkennen. Statt des Hauses, wird nun ein Haus mit einem Tresorschloss angezeigt.

Wenn nun ein Benutzer sein Passwort vergessen hat, kann man nun das Masterpasswort dazu benutzen, das vergessene Benutzerpasswort zurückzusetzen. Um das zu tun, klickt man das Schloss-Symbol mit dem Titel „Passwort vergessen" beim Anmeldefenster an. Danach kann man das Hauptkennwort eingeben (Abbildung 6.22). Dabei wird nicht nur das Benutzerpasswort zurückgesetzt, sondern auch der File Vault geschützte Benutzer Ordner entschlüsselt.

Abbildung 6.22 – Passwort Assistent

## 4 Das Verschlüsseln von Dateien und Ordner

Wie vorhin beschrieben, kann man mit File Vault den gesamten Benutzer Ordner verschlüsseln. Wenn man nur einen einzelnen Ordner verschlüsselt haben möchte, kann das Mac OS X nicht transparent durchführen, wie es etwa Windows XP kann, sondern man muss sich mit ein paar Tricks zu helfen wissen. Mit dem Festplatten-Dienstprogramm kann man Disk Images erzeugen, welche alle Arten von Dateien und Ordner enthalten können. Wie File Vault benutzt auch das Festplatten-Dienstprogramm den Advanced Encryption Standard (AES) mit einem 128-Bit-Schlüssel. Wer also File Vault nicht benutzen will und trotzdem den Zugriff unberechtigter Benutzer auf einzelne Dateien verhindern will, kann das mit dem Festplatten-Dienstprogramm tun.

Dazu wählt man im Festplatten-Dienstprogramm unter Mac OS X 10.3 im Menü „Images" den Menüpunkt „Neu". Unter Mac OS X 10.4 erstellt man ein leeres Disk Image über das Menü „Ablage". Dort wählt man den Menüpunkt „Neu" und „Leeres Disk Image" (Abbildung 6.23).

Abbildung 6.23 – Festplatten-Dienstprogramm

Nun muss man entscheiden, wie groß das Image werden soll. Die Größe sollte adäquat zur Benutzung sein, da Disk Images nicht direkt vergrößert werden können. Bei der Verschlüsselung wählt man AES-128. Denn ohne diese Option wird das Image nicht verschlüsselt. Beim Format wählt man „Mitwachsendes Image" (sparseimage). Mit dieser Option wird ein Image mit der angegebenen maximal Größe erzeugt. Die Datei selber ist aber nicht wie in Abbildung 6.24 100MB groß, sondern nur soviel wie aktuell benutzt

wird. Das bedeutet: das Image wächst mit der Anzahl der Dateien welche in ihm gespeichert sind. Dies kann es bis zu der angegebenen maximalen Größe tun. Im Beispiel von Abbildung 6.24, also 100 MB.

Abbildung 6.24 – Festplatten-Dienstprogramm

Wenn die Einstellungen stimmen und man die Taste „Erstellen" gedrückt hat, fragt einem das Programm noch nach dem gewünschten Passwort für das Image. Unter Mac OS 10.4 steht einem auch wieder der Passwort Assistent zur Verfügung. Das Image wird nach dem Erstellen auch gleich gemountet. Nun kann man die Dateien welche gesichert werden sollen hinein kopieren und das Image unmounten. Danach bleibt einem nur noch die Image Datei. Diese kann kopiert oder auf einen Datei-Server gelegt werden. Wenn sie jemand öffnet wird er nach einem Passwort gefragt (Abbildung 6.25). Wichtig ist, dass diese Image Datei nun nicht bloß Passwort geschützt ist, sondern zusammen mit unserem Passwort eine Verschlüsselung angewendet wird, jedes Mal wenn wir etwas Neues hineinkopieren.

Abbildung 6.25 – Verschlüsseltes Diskimage

Wenn man zum Beispiel sichere Office Dokumente in einem verschlüsselten Image hat und diese bearbeiten möchte, öffnet man sie am besten ab diesem Image. Alle Office Applikationen speichern nämlich ihre Zwischenspeicherungen und Cache-Dateien unter dem Pfad an dem sich das zu bearbeitende File befindet. So ist sichergestellt, dass keine Daten trotzdem auf den Rechner gelangen und ausgelesen werden können.

## 5 Anpassen von umask

Die umask Einstellungen bestimmen die Berechtigungen einer Datei oder Ordners wenn er gespeichert wird. Unter Mac OS 10.3 wurde ein neues Gruppenmodell eingeführt. Ab 10.3 erhielt jeder Benutzer auch gleichzeitig eine eigene Gruppe. Somit ist die lokale Benutzerverwaltung einfacher geworden. Die Verwaltung der Berechtigungen im Netzwerk dafür schwieriger. Jeder Mac OS X Administrator kennt das Problem, wenn die Benutzer im Transfer Ordner die Dokumente nicht öffnen können.

Diese Gruppe mit nur einem Mitglied ist aber nicht unser Sicherheitsproblem, sondern die Zugriffeinstellungen für die Gruppe „Andere". Wenn man unter Mac OS X eine Datei speichert oder einen Ordner erzeugt, wird diese Gruppe immer mit Leserechten ausgestattet (Abbildung 6.26), was unter Umständen nicht erwünscht ist. Gesteuert und ausgelesen wird diese Einstellung unter Unix Systemen mit `umask`. Um die aktuelle Einstellung des Benutzers auszulesen, gibt man lediglich den Befehl `umask` im Terminal ein:

```
pts$ umask
0022
```

22 setzt also den Besitzer und die Gruppe auf volle Berechtigung. Die Gruppe „Andere" auf die Berechtigung „Lesen". Um das ein bisschen lesbarer zu haben, kann man `umask` mit der Option –S anweisen, das Ganze auszuschreiben:

```
pts$ umask -S
u=rwx,g=rx,o=rx
```

Nun sehen wir, das der Besitzer volle Berechtigung hat, die Gruppe und die Anderen nur zum lesen und ausführen. Dies können wir nun ändern. Dem Befehl `umask` kann man die neue Berechtigung angeben. Eine gute und sichere Berechtigung ist:

```
pts$ umask 037
```

Damit sieht die Berechtigung so aus:

```
pts$ umask -S
u=rwx,g=r,o=
```

Damit hat der Ersteller der Datei den vollen Zugriff. Die Gruppe in der er sich befindet nur Berechtigung zum lesen. Alle anderen haben keine Berechtigung und können die Datei nicht mal lesen. Wie diese Werte zustande kommen lässt sich nicht in ein paar Worten erklären. Das macht aber nichts, da man im Terminal in aller Ruhe die Werte austesten kann. Schließt man das Terminal Fenster haben die Angaben sowieso keine Gültigkeit mehr. Um für das System gültig zu sein müssen diese erst noch eingetragen werden.

Eintragen kann man sie an zwei Orten. Zum einen im Systemordner. An diesem Ort haben diese Werte für alle Benutzer ihre Gültigkeit. Die Datei in der sie eingetragen werden müssen, heißt „global.defaults" und liegt unter:

```
/System/Library/Frameworks/PreferencePanes.framework/Versions/A/Resources/global.defaults
```

In der Datei „global.defaults" findet man den Eintrag:

```
NSUmask = "18";
```

Warum haben wir hier nun den Wert 18, obwohl wir oben gesehen haben, dass uns umask den Wert 22 angegeben hat? Das rührt daher, dass zwar NSUmask und umask dasselbe bewirken, jedoch andere Eingaben erwarten. 22 ist ein oktaler Wert und konvertiert in das dezimale System, ist das 18. Wenn man sich mit der Umrechung nicht auskennt, benutzt man am besten eine Umrechnungstabelle etwa unter: http://www.pts.ch/consoliero/Conversion_table.html. Es hat sich aber gezeigt, dass sich in diesen globalen Einstellungen allzu restriktive Einstellungen als nicht sehr gut erwiesen haben. Der Installer von Mac OS X nimmt einem das sehr übel. Auch unter Mac OS X 10.4 funktionieren gewisse Installationsroutinen nicht mehr.

Einfacher und sicherer geht es wenn man diese Einstellungen nicht global macht, sondern beim Benutzer direkt. Da jeder Benutzer diese Einstellungen aus den „global.defaults" übernimmt, hat Mac OS X keinen Wert in den Benutzereinstellungen bereitgestellt. Diesen müssen wir erst erstellen. Wenn er mal generiert und definiert ist, überschreibt er die Angaben der „global.defaults"-Datei. Beim Benutzer wird diese in der unsichtbaren Datei „.GlobalPreferences.plist" eingetragen. Diese liegt unter:

```
~/Library/Preferences/.GlobalPreferences.plist
```

Wie gesagt fehlt in dieser Datei der NSUmask-Schlüssel, er muss zuerst eingetragen werden. Aus obigen sicheren Beispielen wollen wir nun den Wert „37" eintragen. Aus der Umrechnungstabelle können wir entnehmen, dass der oktale Wert 37 dezimal 31 ist. Nun müssen wir in der Parameterliste eine Wert für NSUmask mit dem dezimalen Wert 31 eintragen. Beginnen Sie gleich unter <dict>:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>NSUmask</key>
<integer>031</integer>
                        <key>AppleCollationOrder</key>
                              <string>de</string>
```

Nachdem man gespeichert und sich neu angemeldet hat, wird jede Datei und jeder Ordner der gespeichert wird wie folgt erstellt:

```
u=rwx,g=r,o=
```

Abbildung 6.26 – Standard Berechtigungen

Trotzdem hat der Benutzer die Möglichkeit über den Finder oder im Terminal mit chmod die Berechtigung von Dateien und Ordner zu modifizieren. Ihm dieses Recht wegzunehmen ist nicht das Ziel. Das Ziel ist, ihn nicht nur mit Worten auf die Gefahr aufmerksam zu machen, sondern jede Datei automatisch schon mal richtig abgespeichert zu haben.

Wenn Sie nun zu den Administratoren gehören die immer wieder Probleme mit Zugriffsberechtigungen haben wo eigentlich keine sein sollten, weil Sie zum Beispiel gar keine Berechtigungen wollen, sondern alle Benutzer Vollzugriff haben sollten, probieren Sie einmal den Wert 0. Dieser ist Vollzugriff für alle, inklusive „Jeder".

## 6 Systemkonten absichern

Zwei Benutzerkonten brauchen im Laufe der Grundkonfiguration unsere Aufmerksamkeit. Zum einen müssen wir Anpassungen an den Berichtigungen des ersten Administrators des Systems vornehmen. Zweitens müssen wir den root-Benutzer ein bisschen modifizieren.

### 6.1 Den Benutzerordner des Administrators abschließen

Wenn File Vault nicht eingeschaltet ist, erlauben es die Standard Berechtigungen von Mac OS X den Inhalt des Benutzerordners anzuschauen (Abbildung 6.27). Dies hat damit zu tun, dass Apple für jeden Benutzer eine Art Briefkasten vorgesehen hat, den Ordner „Öffentlich". Um diesen über den Finder zu erreichen, muss der Benutzerordner für alle Benutzer lesbar sein.

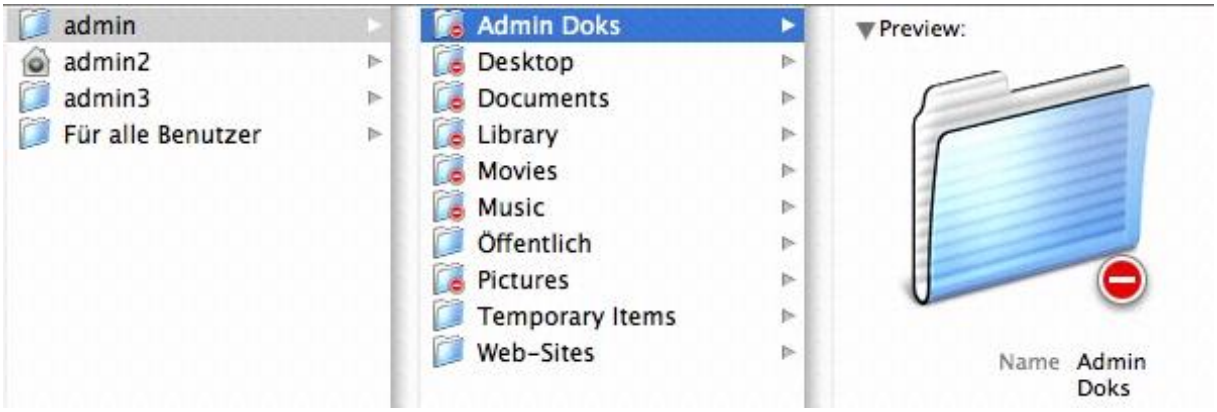


Abbildung 6.27 – Berechtigungen des Benutzerordners

Da aber nicht alle Benutzer genau die Ordnerstruktur benutzen die Apple vorgibt, kann mit dieser Berechtigungs-Einstellung im schlechtesten Fall, Daten aus dem Benutzerordner ausgelesen werden. Und auch wenn mit den Berechtigungen alles geklappt hat, sieht jeder der an diesem Computer arbeitet, zumindest die Ordnerstruktur die der Administrator hat. Um dieses zu verhindern, sperrt man im Terminal mit chmod einfach den Benutzerordner für alle anderen. Für einen Benutzerordner namens „admin" sieht das in etwa so aus.

```
chmod 700 /Users/admin
```

Somit ist es außer dem Besitzer niemandem erlaubt auf den Ordner zuzugreifen. Dadurch kann der Inhalt auch nicht mehr ausgelesen werden (Abbildung 6.28).

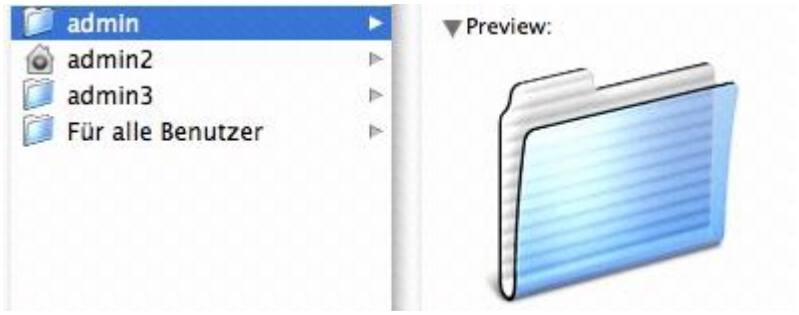


Abbildung 6.28 – Modifizierte Berechtigungen des Benutzerordners

## 6.2 Administrative Benutzerkonten ausblenden

Beim Erstellen eines ASR Images welches auf alle Mac's verteilt wird, erstelle ich jeweils einen lokalen Administrator. Dieser hilft mir administrative Aufgaben, usw. wahrzunehmen. Wenn bei den Anmeldeoptionen die Auswahl „Liste der Benutzer anzeigen" gewählt ist, wird natürlich auch der lokale Administrator angezeigt. Allerdings ist es nicht wichtig, dass der Benutzer diesen beim Anmelden immer zu Gesicht bekommt. Zudem ist es auch ein Sicherheitsrisiko, da die Hälfte der Sicherheit schon vergeben wurde. Einem potentiellen Angreifer sind durch die Listenansicht schon einmal alle Benutzernamen bekannt. Wenn man nun schon den Ordner verschlossen hat, kann man den lokalen Administrator auch gleich ganz ausblenden. Mit einem kleinen Eintrag in die Einstellungsdatei des Login-Fensters kann man das aber schnell anpassen. Die entsprechende Datei liegt unter:

```
cd /Library/Preferences/com.apple.loginwindow.plist
```

Der Eintrag in diese Datei sieht wie folgt aus:

```
sudo defaults write com.apple.loginwindow.plist HideAdminUsers true
```

Danach werden alle Benutzer mit administrativem Status ausgeblendet.

Übrigens, wenn Benutzer eingerichtet werden, welche eine UID unter 500 haben, werden diese automatisch ausgeblendet.

### 6.3 Den root-Benutzer sichern

Wie andere Unix basierte Betriebssysteme auch, enthält Mac OS X ein Benutzerkonto für den root-Benutzer. Administrative Systemaufgaben werden mit diesem Benutzerkonto durchgeführt. Wenn Mac OS X installiert wird, ist der root-Benutzer deaktiviert. Es empfiehlt sich aus vielerlei Hinsicht, dass der root-Benutzer auch deaktiviert bleibt. Benutzerkonten mit Administrationsrechten bieten einem Sicherheit durch erneutes identifizieren bei systemkritischen Vorgängen. Der root-Benutzer kennt das nicht mehr. Vorgänge welche durch das root-Konto ausgelöst werden, werden sofort ausgeführt und können die Stabilität des Systems gefährden.
Zudem hat der root-Benutzer den kompletten Zugriff auf alle Daten auf dem Rechner, was wiederum ein Sicherheitsloch darstellen könnte.

Wenn der root-Benutzer aktiviert wurde, sollte er nun deaktiviert werden. Mit der folgenden Prozedur wird das grafische Anmelden am Finder durch den root-Benutzer verhindert.

Alle administrativen Aufgaben können, wenn nötig, von einem Administratorenkonto aus, mit dem Befehl sudo, ausgeführt werden. Mehr dazu in 6.5.6.4. Keine der administrativen Aufgaben in der grafischen Benutzeroberfläche benötigt root-Rechte. Jedoch im Terminal werden für gewisse Aufgaben diese Berechtigungen benötigt.

1. Man öffnet das NetInfo-Dienstprogramm im Dienstprogramm Ordner (Abbildung 6.28)

2. Man identifiziert sich als autorisierter Benutzer über das Schloss-Symbol unten links.

3. Über die Spalte „User" findet man den „root"-Benutzer. Im unteren Teil des Fensters werden nun die Details angezeigt (Abbildung 6.28).

4. Nun löscht man den Wert und den Schlüssel von „authentication_authority", indem man das „Löschen"-Symbol in der Symbolleiste drückt.

5. Beim Wert „passwd" doppelklickt man in das Wertefeld das „********" anzeigt. Hier trägt man einen einzelnen Stern „*" ein.

6. Nun schließt man das Schloss am linken unteren Rand wieder. NetInfo fragt einem nun ob man diese Kopie von NetInfo aktualisieren möchte. Dies bestätigt man mit „Ja".

Mit dieser Prozedur ist es nun nicht mehr möglich sich als root-Benutzer am Finder anzumelden.

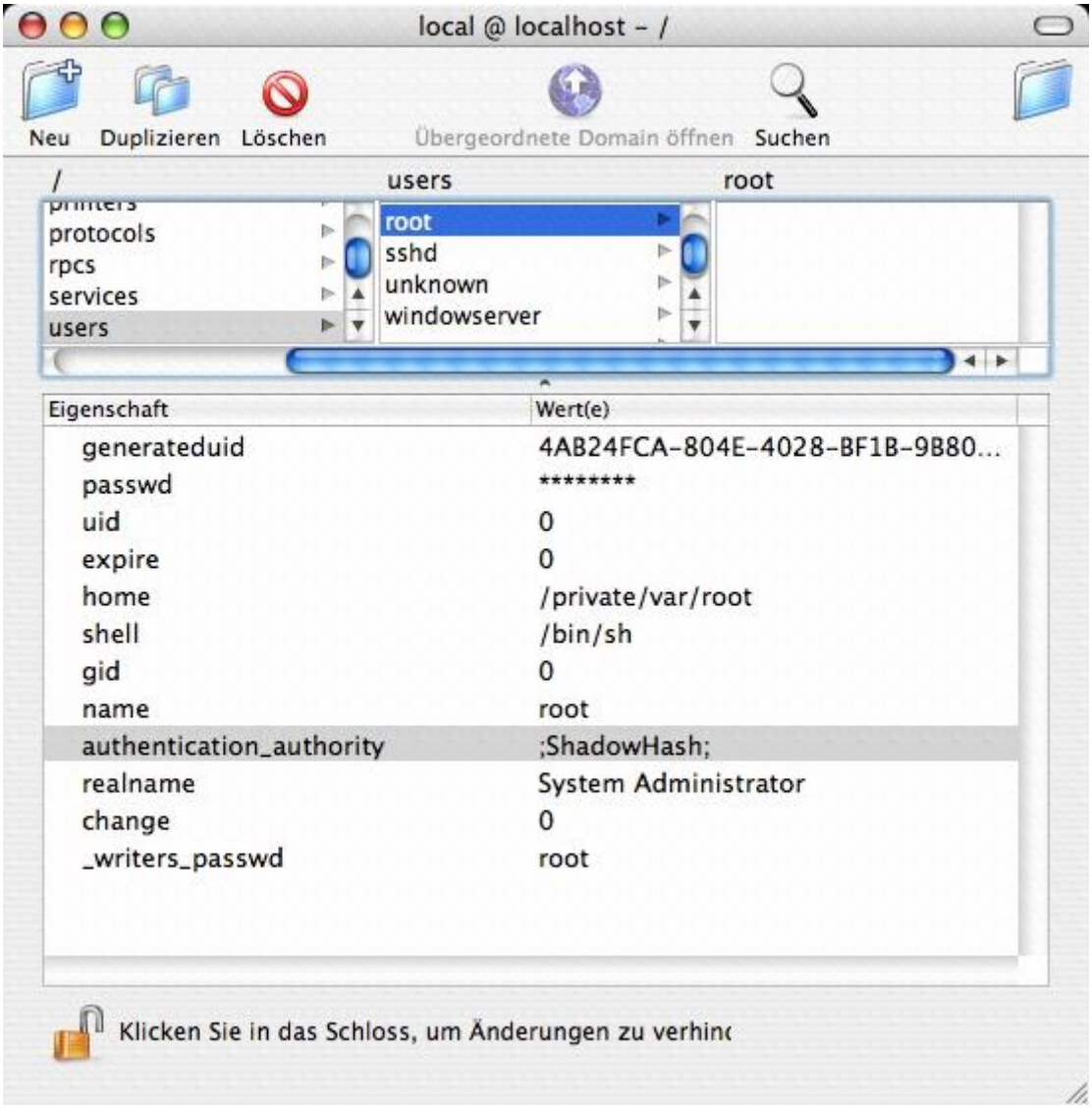

Abbildung 6.29 – NetInfo Manager

## 6.4 sudo benutzen

Das sudo Programm lässt Benutzer welche sich in der Administratoren Gruppe befinden, Terminal Befehle mit root-Rechten ausführen. Um dies zu tun, gibt man einfach vor dem Befehl den man mit root-Rechten ausführen möchte, `sudo` ein:

```
pts$ sudo /usr/libexec/locate.updatedb
Password:
```

Das System fragt einem danach nach dem Passwort des aktuellen Benutzers. Standardmäßig können alle Benutzer, welche sich in der Administratorengruppe befinden, Befehle mit sudo ausführen. Diese Einstellungen sind in einer Datei namens `/etc/sudoers`, gespeichert. Diese Datei kann nur mit root-Rechten angezeigt und bearbeitet werden:

```
pts$ sudo less /etc/sudoers
Password:

# User privilege specification
root    ALL=(ALL) ALL
%admin  ALL=(ALL) ALL
```

Um festzustellen ob sich ein Benutzer in der Administratorgruppe befindet, schaut man am sichersten im NetInfo Manager nach. Dazu sucht man in „Groups" nach der „admin"-Gruppe. Dort kann man dann im Wert im Schlüssel „users" ablesen, welche Benutzer eingetragen sind (Abbildung 6.30).

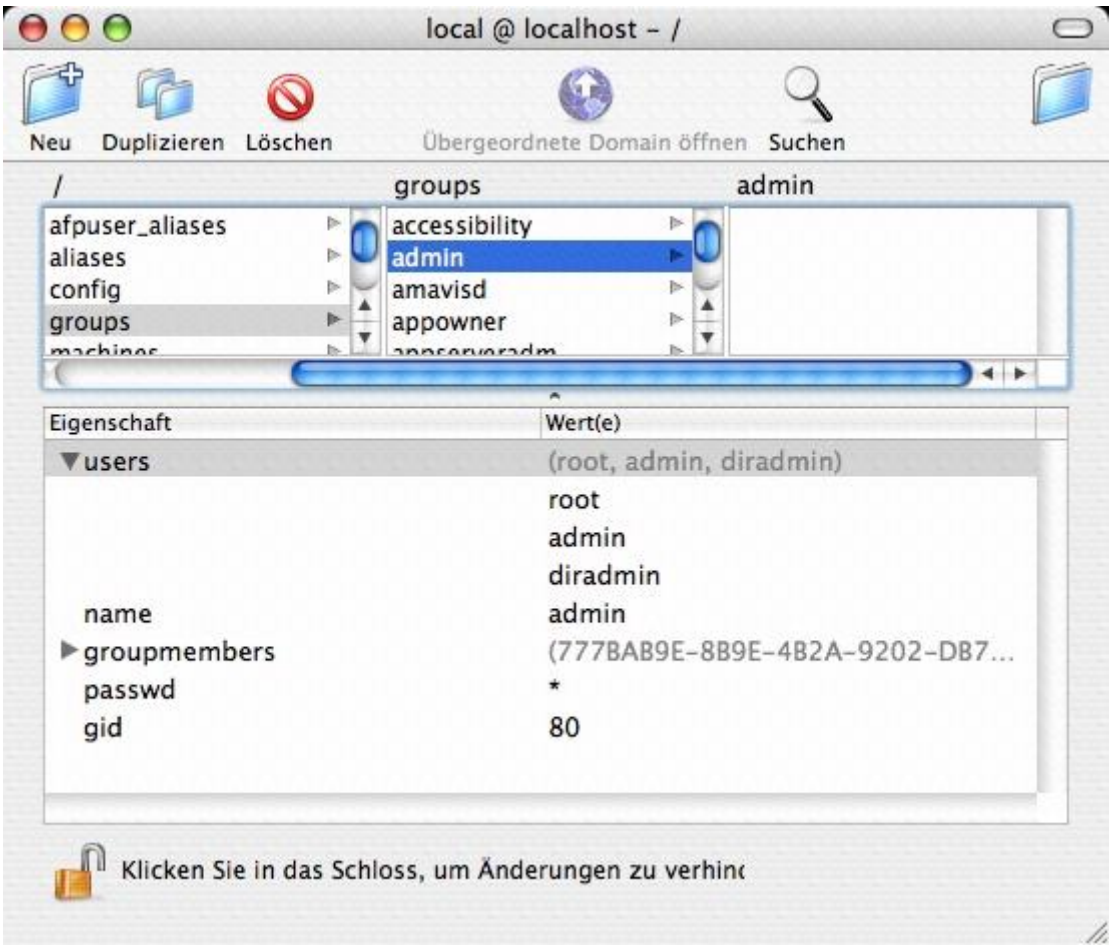


Abbildung 6.30 – NetInfo Manager (Bilder/6_5/picture15.jpg)

Das manuelle Anpassen von `/etc/sudoers` ist eher kompliziert. Wenn man einzelne Benutzer mit erweiterten Berechtigungen über `/etc/sudoers` ausstatten möchte, liest man am besten das Manual:

```
pts$ man sudoers
```

## 6.5 Den „single user boot" absichern

Wenn man beim Start des Rechners die Tastenkombination „Befehl"+"S drückt, bootet der Rechner in den so genannten „single user mode". Dieser heisst so, weil nur ein Benutzer verfügbar ist. Der root-Benutzer. Alle anderen Benutzer werden erst durch die NetInfo-Datenbank dem System zur Verfügung gestellt. Diese ist aber in diesem Modus noch nicht verfügbar. Dass man mit zwei Tasten root-Rechte bekomme und so auch alle Dateien auf dem Rechner auslesen kann, ist natürlich ein Problem. Dies können wir aber mit zwei Maßnahmen verhindern.

Auf jedem Mac wird, sofort nachdem der Computer angeschaltet wird, die Open Firmware ausgeführt. Diese Boot-Firmware ist vergleichbar mit einem BIOS unter x86 basierenden PC's. Um zu verhindern, dass Benutzer root-Rechte durch den „single user mode" bekommen, oder den Rechner von einer anderen Harddisk starten können, müssen wir die Sicherheitseinstellungen (Security Mode) der Open Firmware anpassen. Dazu booten wir den Computer und wechseln in die Open Firmware. Dies geschieht durch drücken von „Befehl"+"alt"+"O"+"F" während dem Start.

Beim nun erscheinenden Prompt, gibt man folgendes ein:

```
password
```

Den Befehl bestätigt man mit „Enter". Danach erscheint ein Prompt welcher das Passwort und die Passwort Verifizierung abfragt. Nun müssen wir uns über die Stufe der gewünschten Sicherheit entscheiden. Konkret werden drei Sicherheits-Modi unterstützt: none, command, oder full.

| | |
|---|---|
| none: | Dieser Modus schaltet die anderen zwei Modi aus |
| command: | Dieser Modus ist für Desktop- und Laptop-Geräte der wohl beste Modus. Im command-Modus wird kein Passwort abgefragt. Es ist dem Benutzer jedoch nicht möglich im single user mode zu starten, im verbose mode zu booten, das boot-volumen zu ändern, den Laptop ab einer externen HD oder einer CD zu starten. Ebenso ist es nicht möglich den Computer in den Target-Mode zu versetzen, so dass er an einem anderen Gerät via FireWire gemountet werden kann. Auch das Parameter RAM kann nicht gelöscht werden. |
| full | In diesem Modus wirkt kummulativ mit dem command-Modus. Zusätzlich muss man bei jedem Systemstart das Open Firmware Passwort eingeben. |

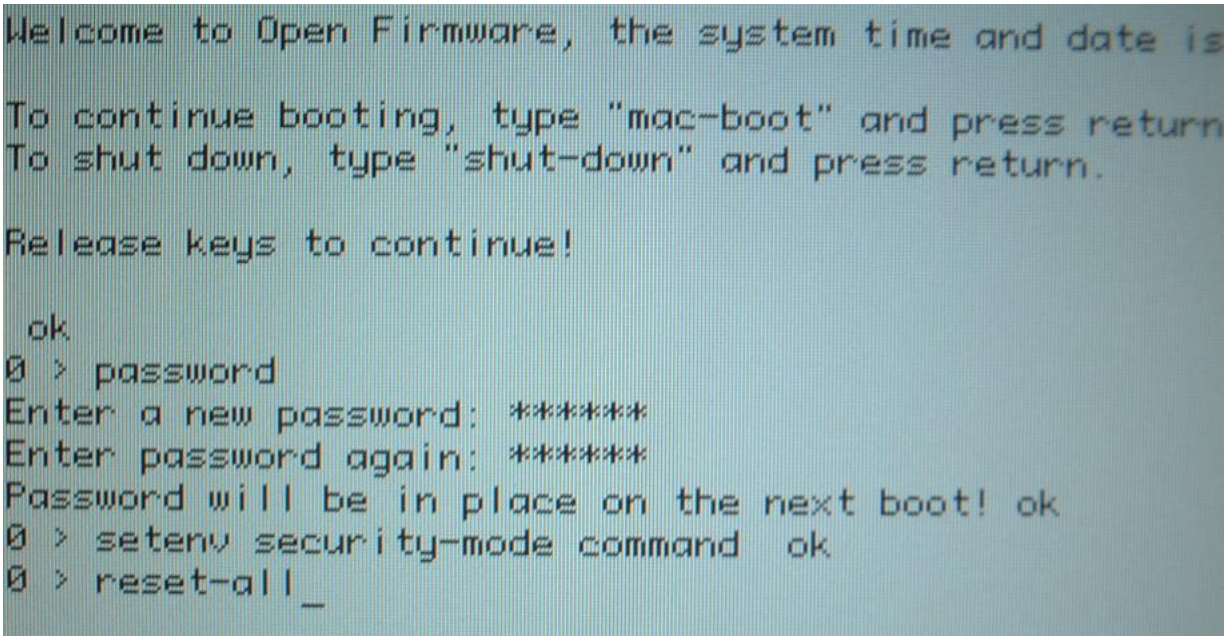

Abbildung 6.31 – OpenFirmware

Wie gesagt ist der command-modus für Workstations die optimale Wahl. Um diesen Modus auszuwählen geben wir beim Prompt folgendes ein:

```
setenv security-mode command
```

Auch hier wird mit „Enter" bestätigt. Um den Computer neu zu starten und die Einstellungen zu aktivieren, tippt man nun:

```
reset-all
```

Der ganze Ablauf der Aktivierung wird in Abbildung 6.31 dargestellt.

Um das den Sicherheits-Modus wieder zu entfernen, begibt man sich wieder in die Open Firmware und setzt den Modus auf „none" (Abbildung 6.32):

```
setenv security-mode none
reset-all
```

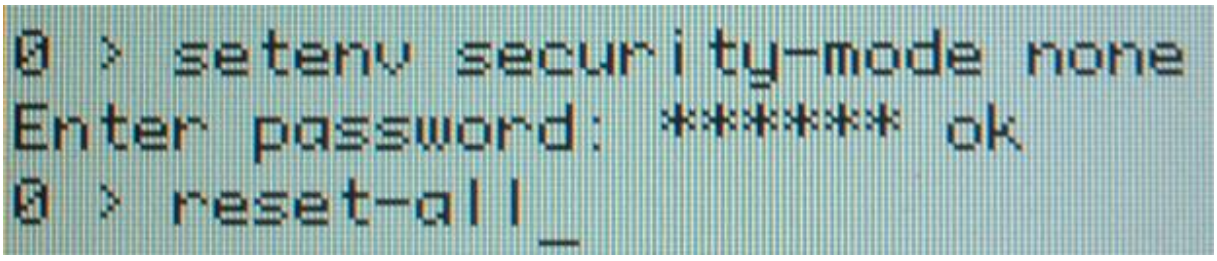

Abbildung 6.32 – OpenFirmware

**Open Firmware Sicherheit aushebeln:** Die Sicherheit welche Open Firmware bietet, kann ausgehebelt werden wenn der Benutzer physischen Zugang zum Rechner hat. Wenn man die Anzahl der RAM verändert, also RAM entfernt oder hinzufügt kann man durch dreimaliges löschen des Parameter RAM's „Befehl"+"alt"+"P"+"R" den Sicherheits-Modus auf „none" zurücksetzen.

Sie sehen, dass die Open Firmware Sicherheit sehr leicht ausgetrickst werden kann. Vor allem an einem Laptop der gestohlen wurde, ist es ein Leichtes, root-Rechte zu bekommen. Ideal wäre es wenn man im single user mode nicht automatisch root-Rechte hätte, sondern sich diese Rechte zuerst mit dem root-Passwort freischalten müsste. Das lässt sich einrichten.

Um das System dazu zu bringen, dass es im single user mode nach dem root-Passwort fragt, müssen wir die Konsole als „unsicher" klassifizieren. Dies geschieht in der Datei `/etc/ttys`. Wenn wir das gemacht haben, fragt das System im single user mode nach dem root-Passwort. Dieses wird in der Datei `/etc/master.passwd` abgefragt. Normalerweise ist in dieser Datei nichts gespeichert. Also kein Passwort von irgendeinem Benutzer. Das bedeutet für uns, dass sich so kein Benutzer anmelden kann. Nicht einmal der root-Benutzer. Damit haben wir die totale Sicherheit, solange dem Rechner nicht die Harddisk ausgebaut wird.

Um also die Frage nach einem Passwort im single user mode zu aktivieren gehen wir wie folgt vor:

1. Melden Sie sich als Administrator an

2. Starten Sie das Terminal in den Dienstprogrammen

3. Wechseln Sie in das Verzeichnis /etc.

```
pts$ cd /etc
```

4. Erstellen Sie zur Sicherheit eine Kopie von /etc/ttys.

```
pts$ cp ttys ttys.old
```

5. Öffnen Sie ttys nun in einem Editor

```
pts$ sudo pico ttys
```

6. Ändern Sie in der Datei ttys alle Vorkommnisse von „secure" in Konfigurationszeilen in „insecure" um. Konfigurationszeilen sind solche welche nicht mit „#" zu Zeilenbeginn auskommentiert sind.

```
#                               @(#)ttys                    5.2         (Berkeley)
6/10/93
#
# name                            getty
                                                          type          status
                                                          comments
#
# If the console is marked insecure, single-user requires
# the root password.
##
```

```
# Since DirectoryServices is not running by the time we enter
# single-user mode, init will ask for the non-shadow crypt
# password stored for root in /etc/master.passwd.  If no such
# password exists, it will not be possible to enter single-user
# mode from a console marked insecure.
##
#console          "/usr/libexec/getty std.57600"       vt100on       secure
console
"/System/Library/CoreServices/loginwindow.app/Contents/MacOS/loginwindow"
vt100 on insecure onoption="/usr/libexec/getty std.9600"
#tahoe's only
#remote            "/usr/libexec/getty std.1200"    pt      on          #
diagnostics

# The tty.serial entry initializes the serial port (if any) for use as a
# terminal (enabling logons over serial). If marked secure, the serial
# port will allow root logons.
# To make the serial port available for outbound
# communications, the tty.serial entry should be turned off
# (set the 4th field to off).
tty.serial      "/usr/libexec/getty serial.57600"   vt100   off insecure

# Fax reception is off by default, use the
# System Preferences panel to enable it.
fax                   "/usr/bin/fax answer"            unknown         off

# Hardwired lines are marked off, by default, so getty(8)
# is quiet when they don't exist.
tty00             "/usr/libexec/getty std.9600"        unknownoff insecure
tty01             "/usr/libexec/getty std.9600"        unknownoff insecure
tty02             "/usr/libexec/getty std.9600"        unknownoff insecure
tty03             "/usr/libexec/getty std.9600"        unknownoff insecure
tty04             "/usr/libexec/getty std.9600"        unknownoff insecure
tty05             "/usr/libexec/getty std.9600"        unknownoff insecure
tty06             "/usr/libexec/getty std.9600"        unknownoff insecure
tty07             "/usr/libexec/getty std.9600"        unknownoff insecure

ttyp0                          none                                  network
ttyp1                          none                                  network
```

7. Verlassen Sie den Editor und speichern Sie die Änderungen. Testen Sie die Änderungen in dem Sie den Rechner in den single user mode starten. Wenn alles korrekt läuft, müssten Sie beim Login nun eine Fehlermeldung bekommen (Abbildung 6.33).

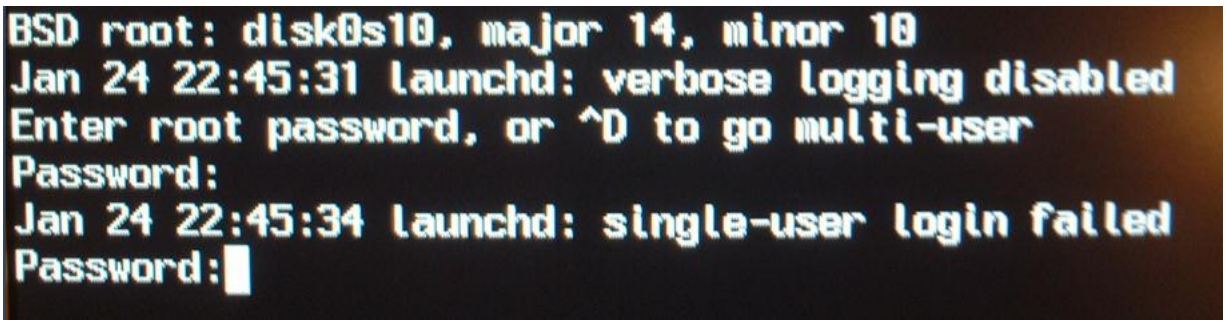


Abbildung 6.33 – Insecure Konsole

Wenn man nun aber trotzdem ab und zu im single user mode booten und gewisse Anpassungen vornehmen muss, hat man mit dieser Konfiguration eventuell ein Problem. Für diesen Fall müssen wir in der Datei `/etc/master.passwd` ein Passwort für den root-Benutzer hinterlegten. Logischerweise steht das Passwort in `/etc/master.passwd` nicht im Klartext, sondern wird mit einem Passwort Hash verschlüsselt. Das bedeutet, das wir unser root-Benutzer Passwort zuerst verschlüsseln und danach in `/etc/master.passwd` eintragen müssen:

1. Melde Sie sich als Administrator an

2. Starten Sie das Terminal in den Dienstprogrammen

3. Wechseln Sie in das Verzeichnis /etc.

```
pts$ cd /etc
```

4. Nun editieren wir master.passwd

```
pts$ sudo pico master.passwd
```

5. Innerhalb des Editors löschen Sie das „*"-Zeichen nach dem Wort „root" und dem „:".

```
nobody:*:-2:-2::0:0:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0::0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
unknown:*:99:99::0:0:Unknown User:/var/empty:/usr/bin/false
smmsp:*:25:25::0:0:Sendmail User:/private/etc/mail:/usr/bin/false
lp:*:26:26::0:0:Printing Services:/var/spool/cups:/usr/bin/false
postfix:*:27:27::0:0:Postfix User:/var/spool/postfix:/usr/bin/false
www:*:70:70::0:0:World Wide Web Server:/Library/WebServer:/usr/bin/false
eppc:*:71:71::0:0:Apple Events User:/var/empty:/usr/bin/false
mysql:*:74:74::0:0:MySQL Server:/var/empty:/usr/bin/false
sshd:*:75:75::0:0:sshd Privilege separation:/var/empty:/usr/bin/false
```

6. Öffnen Sie nun ein neues Fenster innerhalb der Applikation „Terminal". Nun müssen wir das verschlüsselte Passwort generieren. Dies geschieht mit dem Befehl `openssl passwd` . Der Syntax des Befehls sieht so aus: `openssl passws –salt <xx> <passwort>`. Wobei <xx> irgend zwei Buchstaben sind, während <passwort> ein 8-stelliges Passwort ist. Im Terminal sieht das dann so aus:

```
pts$ openssl passwd -salt ab pass123
abBxjdJQWn8xw
```

Die Ausgabe von openssl ist der Passwort Hash. Diese muss nun dem root-Benutzer in der Datei `/etc/master.passwd` zugewiesen werden. Der Hash-Wert wird dort eingesetzt wo wir in Punkt 5 das „*"-Zeichen gelöscht haben:

```
##
nobody:*:-2:-2::0:0:Unprivileged User:/var/empty:/usr/bin/false
root:abBxjdJQWn8xw:0:0::0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
unknown:*:99:99::0:0:Unknown User:/var/empty:/usr/bin/false
```

7. Speichern Sie nun die Datei und testen Sie die Konfigurationsänderung. Wenn alles korrekt gelaufen ist, müssen Sie sich als root-Benutzer im single user mode identifizieren (Abbildung 34).

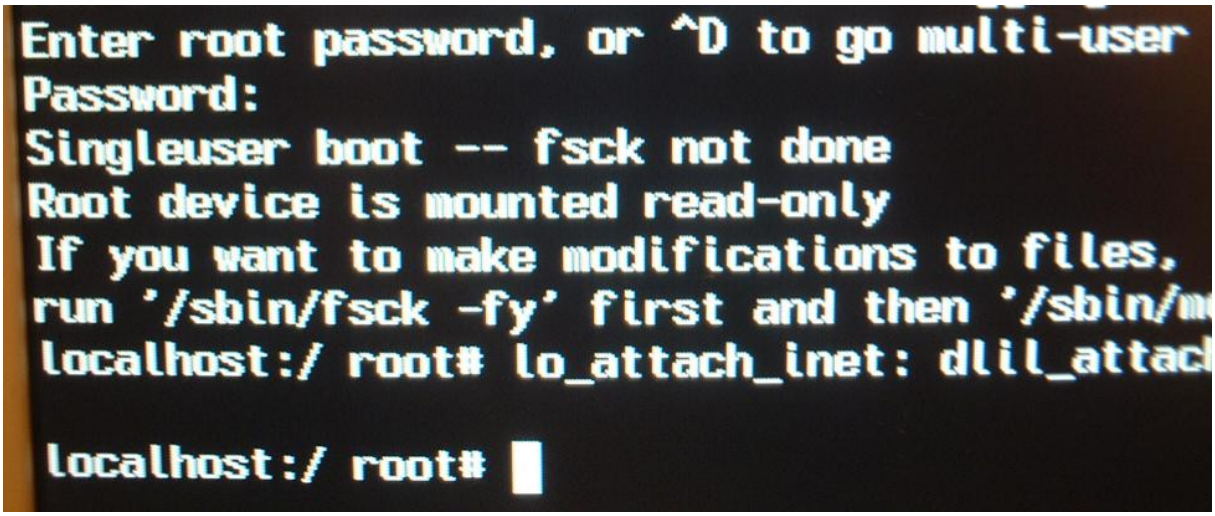

Abbildung 6.34 – Root login im single user mode

### 6.6 Sicherheitshinweise beim Anmelden

Das Anmeldefenster kann gebraucht werden, um Notizen und Hinweise auf die Rechtslage in Ihrem Betrieb hinzuweisen. Zum Beispiel, dass der Netzwerkverkehr gescannt wird, oder dass andere Überwachungstools laufen. Zum einen können solche Hinweise potentielle Angreifer mindestens einschüchtern, zum anderen den Benutzer erinnern, dass sein System zumindest von einem Systemadministrator eingesehen werden kann.

Ein Benutzer kann sich auf verschiedene Arten an ein System anmelden. Zum einen interaktiv am Computer selber, oder von einem anderen Terminal aus. Für beide sollte natürlich die gleiche Meldung erscheinen. Um für die grafische Anmeldung eine Meldung im Anmeldefenster anzuzeigen, müssen wir die Präferenzdatei `/Library/Preferences/com.apple.loginwindow.plist` anpassen.

```
pts$ sudo pico /Library/Preferences/com.apple.loginwindow.plist
```

Fügen Sie nun Ihren gewünschten Text ein. Fügen Sie alles ein, was im unteren Beispiel fett gedruckt ist.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
      <key>LoginwindowText</key>
      <string>Dieses System gehoert der Firma PTS. Mit der Benutzung des
Systems akzeptieren Sie die strikte Einhaltung der IT-Richtlinien. Bei
Fragen wenden Sie sich bitte an das HelpDesk. Intern 222</string>
      <key>MCXPeriodicLaunch</key>
      <integer>600</integer>
```

Speichern Sie nun die Datei. Der nächste Benutzer der sich interaktiv am System anmeldet wird Ihre Meldung am Anmeldefenster angezeigt bekommen (Abbildung 6.35).

Abbildung 6.35 – Anmeldefenster mit Hinweisen

Natürlich müssen wir auch die Benutzer auf dieselben Dinge hinweisen, die sich remote anmelden. Also über das Terminal. Dies geschieht in der Datei `/etc/motd`. Ändern Sie diese ebenfalls ab. Im Unterschied zu einer plist wie beim Anmeldefenster ist die Datei `/etc/motd` keine XML-Datei und kann deshalb einfach angepasst werden:

```
pts$ sudo pico /etc/motd
```

Damit sieht das für einen Benutzer, welcher sich über das Terminal anmeldet, so aus:

```
pts$ ssh admin@server1
admin@server1's password:
Last login: Tue Jan 25 21:29:52 2005 from pts0mac2

Welcome to Server1
_____________________________________________________________________
Dieses System gehoert der Firma PTS.
Mit der Benutzung des Systems akzeptieren Sie die strikte Einhaltung
der IT-Richtlinien.
Bei Fragen wenden Sie sich bitte an das HelpDesk.
```

```
Intern 222
_____________________________________________________________________
server1:~ admin$
```

### 6.7 Unsichere Hardware Komponenten entfernen

Bluetooth und Wireless-Lan können mögliche Sicherheitslöcher sein. Bei Anwendungen bei denen Sicherheit ein Thema ist, sollten diese Hardwarekomponenten wenn möglich physisch entfernt werden. Dies ist allerdings nicht immer möglich, da an gewissen Komponenten nur von Apple zertifizierte Techniker ran sollten. Nicht, dass Sie das nicht auch könnten, aber der Garantie wegen. Eine alternative Variante ist die entsprechenden Kernel Extensions zu entfernen. Dies entfernt zwar keine Hardware, dennoch sind die Geräte auf keinen Fall brauchbar. Dies ist nicht so sicher wie sie physisch zu entfernen, jedoch sicherer als sie in der Systemsteuerung zu deaktivieren. Wenn diese Komponenten später wieder gebraucht werden, können wir diese wieder zurück kopieren. Für das Entfernen wie auch das Zurücksetzen braucht es aber auf jeden Fall die Berechtigung eines Administrators.

Im Ordner `/System/Library/Extensions/` finden wir diese Kernel Erweiterungen. Sie haben die Endung „.kext" (Abbildung 6.36).

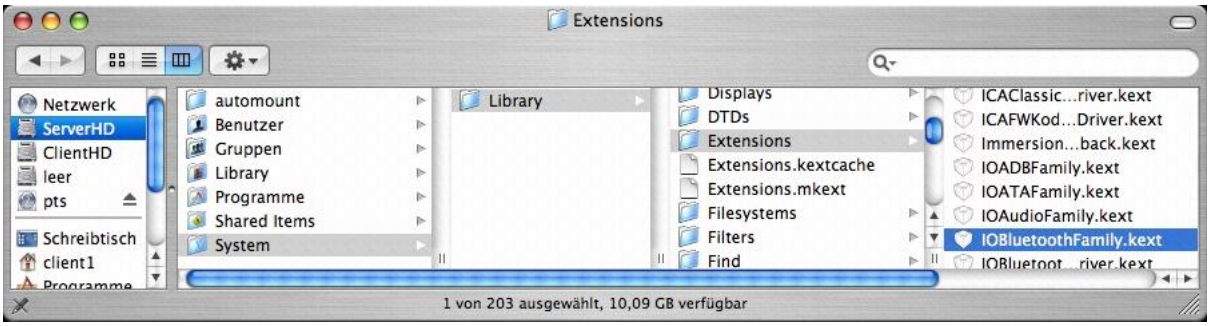

Abbildung 6.36 – Kernel Extensions

Um die AirPort Unterstützung zu deaktivieren, zieht man folgende Komponenten in den Papierkorb:

```
AppleAirPort.kext
AppleAirPort2.kext
AppleAirPortFW.kext
```

Um die Bluetooth Unterstützung zu deaktivieren, zieht man folgende Komponenten in den Papierkorb:

```
IOBluetoothFamily.kext
IOBluetoothHIDDriver.kext
```

Das Betriebssystem hält sich selber eine Liste mit den Kernel Erweiterungen. Diese stimmt nun mit unseren manuellen Änderungen natürlich nicht mehr überein. Deshalb müssen wir diese Dateien ebenfall löschen, so dass sie beim nächsten Neustart neu aufgebaut werden. Dazu löscht man diese beiden Dateien:

```
/System/Library/Extensions.kextcache
/System/Library/Extensions.mkext
```

Löschen Sie den Papierkorb „sicher" (Abbildung 6.20) und starten Sie den Mac neu.

Christoph Müller - www.pts.ch

Publishing Tools Support
Rüschlikon, 9.2.2005

Bei Fragen oder Anmerkungen, kontaktieren Sie mich bitte unter chm@pts.ch

Weitere detaillierte Informationen erhalten Sie aus meinem Buch: **„Mac OS X „Consoliero-Client" Praxis Handbuch"**: ISBN-Nr. 3-905647-17-6.