Hochschule für
Technik und Wirtschaft
des Saarlandes
University of Applied Sciences

Studiengang Kommunikationsinformatik
Prof. Dr.–Ing. Damian Weber

Systemmanagement und Sicherheit

7. Übung

Für die folgenden Aufgaben benötigen Sie den `play`–Rechner mit `root`–Zugang. Halten Sie hierfür ein Terminalfenster offen.

Aufgabe 1 (Kernel installieren)

Finden Sie mit dem `uname`–Kommando heraus, welche FreeBSD–Version auf dem `play`–Rechner aktiv ist.
Laden Sie Kernelsourcen dieser Serie in ein temporäres Verzeichnis. Die entsprechenden Dateien sind unter

```
http://www-crypto.htw-saarland.de/weber/teaching/07_ss_sysi/FreeBSD/sys/ssys.aa
http://www-crypto.htw-saarland.de/weber/teaching/07_ss_sysi/FreeBSD/sys/ssys.ab
...
http://www-crypto.htw-saarland.de/weber/teaching/07_ss_sysi/FreeBSD/sys/ssys.ap
```

zu finden.
Schreiben Sie ein Skript mit einer geeigneten `for`–Schleife, um diese Dateien zu übertragen. Sie können zum Herunterladen auf `isl-s-01` das Kommando `wget` oder `fetch` benutzen.
Das Aneinanderhängen all dieser Dateien ergibt eine Datei des Typs `.tar.gz`. Entpacken Sie den Kernel dann unter `/usr/src`.
Retten Sie den bisherigen compilierten Kernel mittels

```
cp /boot/kernel/kernel /boot/kernel/kernel.old
```

Kopieren Sie in `/usr/src/sys/i386/conf` die alte Kernelkonfiguration `GENERIC` als neue Konfiguration `PLAYnn`, wobei `nn` die Nummer Ihres `play`–Rechners ist. Dies gilt auch im folgenden.
Rufen Sie

```
/usr/sbin/config PLAYnn
```

auf, um die zur Kernelkonfiguration gehörigen Dateien zu erzeugen.
Wechseln Sie in das vom obigen Kommando ausgegebene `Kernel build directory` und erzeugen Sie mit

```
make depend
nohup make >& make.out &
```

den neuen Kernel.
Hinweis: das letzte `make` *dauert ca. 70 Minuten. Daher starten wir es mit Hilfe von* `nohup` *und im Hintergrund mit Umlenkung von* `stdout` *und* `stderr`*.*
Nach erfolgreichem `make` installieren Sie den neuen Kernel mittels

`make install`

Stellen Sie mit `uptime` fest, seit wann der Rechner nicht mehr gebootet wurde.

Booten Sie den Rechner und stellen Sie fest, ob der neue Kernel aktiv ist.

Aufgabe 2 (Filesysteme)

Untersuchen Sie die Partitionierung der Festplatte mit den Befehlen `fdisk` und `bsdlabel`. Die Festplatte soll hierbei nur gelesen und daher nicht verändert werden.

Stellen Sie außerdem fest, dass die Systemverzeichnisse von den vom Benutzer beeinflussbaren Verzeichnissen getrennt sind, d.h. auf verschiedenen Filesystemen liegen. Hierfür stehen drei Methoden zur Verfügung (`mount`, `df`, `/etc/fstab`).

Finden Sie im `dmesg` Output die realen Geräteeigenschaften zu `ad0`, `acd0`. Verfolgen Sie auch die Verkettung der Gerätetreiber von `ad0` bis hin zu `pci0`.

Mounten Sie das *proc*–Filesystem. Überprüfen Sie dort Einträge zu einem Ihnen bekannten Prozess. Siehe auch `man procfs`. Führen Sie danach ein erfolgreiches `umount` auf das *proc*–Filesystem aus.

Aufgabe 3 (Ramdisk)

Befolgen Sie die Anleitung der Vorlesungsfolien, um nacheinander eine Ramdisk mit den drei Implementierungstypen

- swap
- vnode
- malloc (nicht auf Folie)

der Größe 20 MB zu erzeugen. Mounten Sie die Ramdisk unter `/mnt`. Überprüfen Sie nach dem Mounten den freien Speicherplatz mit Hilfe von `df`.

Beim Typ *malloc* partitionieren Sie die Ramdisk zusätzlich mit `fdisk` und `bsdlabel`.

Erzeugen Sie bei mit Hilfe des Kommandos `dd` beim Platzverbrauch auf der Ramdisk einen Überlauf.

Aufgabe 4 (Service installieren (inetd/telnet))

Auf dem `play`–Rechner mit `root`–Zugang ist der `inetd` Server installiert. Dies ist der Internet–Superserver, der Ports für konfigurierte Dienste öffnet und beim Eintreffen einer Client–Nachricht für den Aufruf des entsprechenden Protokollservers sorgt.

Sorgen Sie dafür, daß er beim Booten aufgerufen wird. Hierzu setzen Sie die Umgebungsvariable in `/etc/rc.conf`.

Erzwingen Sie durch ein geeignetes `shutdown`–Kommando, daß die Maschine nach 1 Minute bootet. Überprüfen Sie danach, daß der `inetd` Server nach dem Booten läuft.

Editieren Sie die Datei `/etc/inetd.conf` derart, daß die Zeile für den Service `telnet` aktiv ist. Senden Sie dem `inetd` Server ein HUP–Signal, damit er seine Konfigurationsdatei erneut liest.

Überprüfen Sie mittels eines mittels `netstat`, daß der telnet–Port vorhanden ist.

Überprüfen Sie den `telnet`–Service mit einem Login–Vorgang mittels `telnet` von `isl-s-01` (beachten Sie, daß ein `root`–Login via `telnet` nicht möglich ist).

Aufgabe 5 (limit/ulimit)

Diese Aufgabe kann auf dem `play`–Rechner oder dem `isl`–Rechner gelöst werden.

Mit Hilfe der Shell–Kommandos `limit` (aus der C–Shell) bzw. `ulimit` (aus der Bourne–Shell) können Prozesslimits gesetzt werden. Sie finden eine Beschreibung zu den Kommandos innerhalb der Manualpages zu `csh` und `sh`.

Setzen Sie ein Filesize–Limit, das Sie mutwillig mit dem `yes`–Kommando überschreiten wollen.

Setzen Sie ein CPU-Time–Limit, das Sie mutwillig mit einem eigenen C–Programm überschreiten wollen.

Setzen Sie die Limits mit beiden Programmen (`limit` und `ulimit`).

Zählen Sie die Anzahl Ihrer aktiven Prozesse mit Hilfe von `ps` und `wc`. Setzen Sie ein knapp darüber liegendes Limit für die maximale Anzahl an Prozessen, das Sie dann durch im Hintergrund zu startende Prozesse zu überschreiten versuchen.

Beachten Sie hierbei, dass

- `root` die maximale Anzahl von Prozessen beliebig überschreiten darf
- die Überschreitung dieses Limits in der `/var/log/messages` Datei protokolliert wird