

Sicherheit in Unix-Systemen

Markus Krieger
Rechenzentrum Uni Würzburg

Inhaltsübersicht

- Einführung und Zielsetzung
- Präventive Maßnahmen
 - ◆ Lokale Rechtersicherheit
 - ◆ Netzwerksicherheit
 - ◆ Einschränkung des Root-Accounts
- Erkennung von Vorfällen
- Bearbeiten eines Einbruchs
- Informationsquellen

Einführung

- Ziel der Veranstaltung
- Definition von Sicherheit und Gefahren
- Denkanstöße
- Angreifer, Angriffsmöglichkeiten & Motivationen
- Schritte eines erfolgreichen Angriffs
- Das Hochschulnetz
- Suche nach einer Strategie
- Sicherheit - warum und wieviel

Ziel der Veranstaltung

Was soll erreicht werden?

- ◆ Definition und Diskussion der wichtigsten sicherheitsrelevanten Problemstellungen.
- ◆ Was verbirgt sich hinter diversen Schlagwörtern?
- ◆ Wo sind die wesentlichen Sicherheitslöcher?
- ◆ Wie sehen erste Schritte aus, um die Sicherheit zu verbessern?
- ◆ Vorstellung von Utilities zur Verbesserung der Systemsicherheit.

Ziel der Veranstaltung

Was wird nicht erreicht?

- ◆ Ein ausgereiftes Sicherheitskonzept.
- ◆ Konkrete und genau definierte Schritte was zu tun ist.
- ◆ Sichere Unix-Rechner.

Praktische Definition von Systemsicherheit

- „Ein System ist sicher, wenn es sich so verhält wie man es von ihm erwartet“
- Sicherheitsansprüche von Benutzern / Administratoren:
 - ◆ Integrität von Daten und Programmen
 - ◆ Verfügbarkeit
 - ◆ Konsistenz
 - ◆ Zugriffskontrolle
 - ◆ Auditing

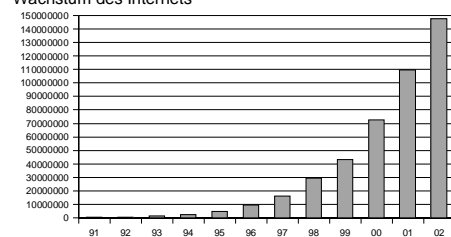
Gefahren für die Sicherheit

- Hardwarefehler
- Softwarefehler
- Fehler von Benutzern / Administratoren
- Äußere Einflüsse (Feuer, Diebstahl, ...)
- Angriffe von Innen und von Außen

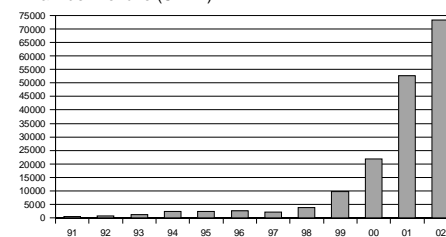
Dies führt zum Verlust von Integrität, Verfügbarkeit und Vertraulichkeit.

Denkanstöße

Wachstum des Internets



Anzahl der Vorfälle (CERT)



Überlegungen

- ◆ Was gibt es für Angriffe?
- ◆ Woher kommen die Angriffe?
- ◆ Wer greift mich an und warum?
- ◆ Wie wichtig sind die Daten meiner Rechner?
- ◆ Was kann ich tun und woher bekomme ich Know-how und Informationen?

Angriffsmöglichkeiten

- Lokale / Remote Angriffe
- Trojaner / Viren / Würmer / Backdoors
- Buffer Overflow
- Falsche Berechtigungen für Dateien / Verzeichnisse
- Fehlerhafte Umgebungsvariablen
- Trust Exploitation (ip-spoofing, hijacking, ...)
- Condition Race
- (D)DOS

Angreifer

- lokale Angreifer:
 - ◆ Studenten
 - ◆ Mitarbeiter
 - ◆ Erfolgreiche remote Angreifer ohne Root-Privilegien
- remote Angreifer:
 - ◆ Kriminelle
 - ◆ Hacker
 - ◆ Script Kiddies
 - ◆ Würmer

Motivation der Angreifer

- Spieltrieb / Neugier
- Zugriff auf Daten / Dienste
- Bandbreite
- Plattenplatz
- Sprungbrett für weitere Angriffe
- Demonstration von Macht und Können
- Das Hochschulnetz ist offen und die Rechner sind schlecht gesichert

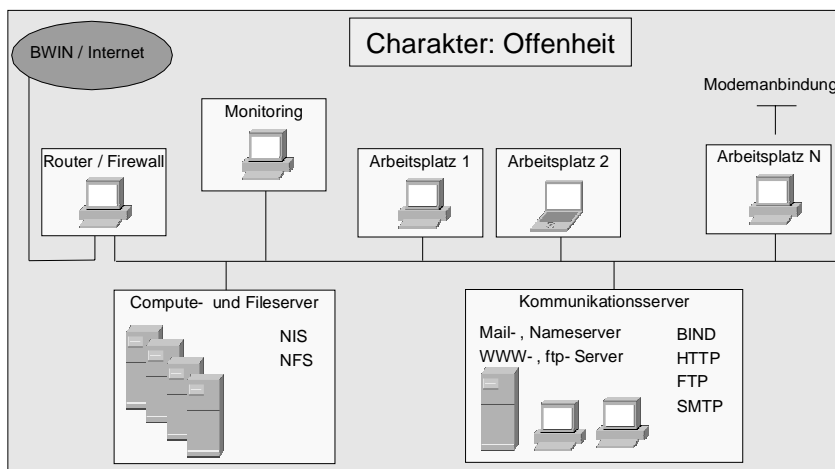
Schritte eines erfolgreichen Angriffs

- 1) Sammeln von Informationen (Scannen nach OS / Diensten, DNS, ...)
- 2) Informationen nach Verwundbarkeiten durchsuchen
- 3) Zugang zum System verschaffen
- 4) Einsatz von exploit-Scripten für root-Rechte
- 5) Installieren von Backdoors, Sniffen und Rootkits und Verwischen von Spuren

Unix-Rechner als beliebte Ziele

- Viele Dienste
- Viele Varianten
- Mächtige Bordmittel
- Bieten die optimale Einbindung ins Internet
- Bieten viele Möglichkeiten der Fehlkonfiguration durch den Administrator
- Laufen oft in einer Defaultinstallation und sind somit angreifbar

Das Hochschulnetz



Suche nach einer Strategie

- Erhöhung der lokalen Sicherheit durch Vorsorgemaßnahmen
 - ◆ Physikalische Sicherheit des Rechners und Netzzuganges
 - ◆ Regelmäßige Backups
 - ◆ Einspielen aktueller Patches und deaktivieren nichtbenötigter Accounts / Dienste
 - ◆ Aktivierung von effektivem Logging
 - ◆ Weitergabe von Sicherheitsinformationen
 - ◆ Einsatz von Tools um die Rechnersicherheit zu überprüfen und um die Grundlage zur Erkennung von Sicherheitsproblemen zu schaffen
 - ◆ Einsatz von Kryptographie
 - ◆ Führen eines Logbuchs

Suche nach einer Strategie

- Erkennen sicherheitsrelevanter Vorfälle
 - ◆ Auswerten von Logdateien
 - ◆ Regelmäßiges Überprüfen der Rechnerkonfiguration (Dienste und Dateisystem)
 - ◆ abnormales Verhalten (Abstürze, hohe Last, ...)
 - ◆ externe Informationsquellen (IDS, Meldungen)
- Problematisch dabei:
 - ◆ Ist der Rechner gehackt, läßt sich der Zustand mit Bordmitteln nicht mehr verlässlich überprüfen

Suche nach einer Strategie

■ Reagieren auf einen Vorfall

- ◆ Ruhe bewahren! Durch voreiliges Handeln kann sonst mehr Schaden als durch den eigentlichen Vorfall angerichtet werden.
- ◆ Befolgen von vorhandenen Sicherheitsrichtlinien
- ◆ Sicherung von Spuren
- ◆ Wiederherstellung des Sollzustands
- ◆ Analyse und Suche nach weiteren betroffenen Rechner im lokalen Netz
- ◆ Meldung an Rechenzentrum / Rechtsabteilung / Provider / CERT (Koordination / Strafverfolgung)

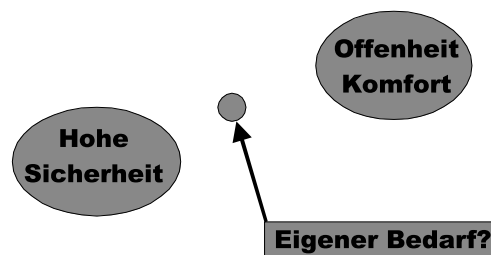
Sicherheit - warum und wieviel?

■ PRO hohe Sicherheit

- ◆ Schutz der eigenen Ressourcen
- ◆ Erhaltung der Verfügbarkeit
- ◆ Schutz der Vertraulichkeit
- ◆ Verhinderung von Mißbrauch
- ◆ Rechtliche Situation z. Zt. Unklar

■ CONTRA hohe Sicherheit

- ◆ Fragwürdigkeit des Nutzens
- ◆ Einschränkung des bequemen Arbeitens
- ◆ Implementierung zeitaufwendig und schwierig
- ◆ Widerspricht dem offenen Charakter eines Hochschulnetzes



Probleme:

- ◆ Entscheidungsfindung
- ◆ Konzept
- ◆ Realisierung
- ◆ Maintenance

Lokale Rechnersicherheit

- Physikalische Sicherheit
- Grundinstallation
- Backups: Die wichtigsten Kommandos
- Paßwortsicherheit
- Zugriffsrechte

Physikalische Sicherheit

- Zugang zum Rechner bzw. der Rechnerhardware sichern
- BIOS-Paßwort setzen
- Bootreihenfolge fest auf Festplatte einstellen
- Paßwort für Bootloader setzen (z.B. LILO)
- Rechner vor unbeabsichtigtem Herunterfahren sichern

„Bootaccess = Rootaccess“

Grundinstallation

VOR einer Installation

- Rechner bei Grundinstallation nicht am Netz betreiben
- Patches und Updates nur aus vertrauenswürdigen Quellen beziehen
- Checksummen der Updates mit den Herstellerangaben vergleichen (z.B. md5 Summen, Signaturen)
- Security Mailinglisten des Herstellers abonnieren
- Lokale Security Mailinglisten abonnieren
- Geeignete Partitionierung mit getrenntem /, /var/log und /tmp überlegen
- Geeignete Filesysteme für die Partitionen auswählen

Grundinstallation

- Updates einspielen.
- Nach Einspielen von Updates Konfigurationsdateien überprüfen (neue Defaults möglich)
- Nichtbenötigte Dienste deaktivieren (Voranstellen eines # vor die entsprechende Zeile in /etc/inetd.conf)
- Nicht benötigte Start- / Stopscripten deaktivieren (/etc/rc.d/, /etc/init.d/)
- Tcpwrapper in /etc/inetd.conf für Zugangskontrolle und Logging konfigurieren
- /etc/hosts.equiv löschen, falls die r* Utilities nicht verwendet werden (ssh bietet eine sichere Alternative)

Grundinstallation

- Nicht benötigte Systemaccounts / -gruppen deaktivieren (aus `/etc/passwd` entfernen oder beim Paßword * voranstellen)
- Hardening Scripten ablaufen lassen
- Interaktive Remotelogins für Systemaccounts abschalten (`/etc/login.access` oder `/etc/ftputils`)
- Verbiete `lpd` und `syslogd` Verbindungen aus dem Netz anzunehmen
- `/etc/hosts.allow` und `/etc/hosts.deny` restriktiv konfigurieren

- Suchpfad von `root` darf das aktuelle Verzeichnis NICHT enthalten

Grundinstallation

- Nach einem Reboot die laufenden Dienste überprüfen (`netstat -a` und `ps aux`)
- Remote logging in `/etc/syslog.conf` für wichtige Logeinträge
- Sicherstellen daß alle von `root` bzw. von den Start-/Stopsripten verwendeten Dateien und ALLER ihrer übergeordneten Verzeichnisse als Besitzer `root` haben und nur `root` Schreibrechte besitzt
- Verwendung absoluter Dateinamen in Scripten
- Backup bzw. Image des kompletten Systems anlegen (wichtig für Wiederherstellung bzw. Überprüfen des Rechners)

Grundinstallation

- Nach erfolgreicher Grundinstallation unbedingt mit tripwire oder aide Datenbank mit Informationen über den Zustand des Dateisystems erstellen
- Erstellen einer Liste aller Dateien und Verzeichnisse mit „ls -laR /“ (Zugriffszeiten, Größe und Mode) und md5-Summen (Integrität) mit „find / -type f -exec md5sum {} \;“
- Die Datenbank bzw. die Liste unbedingt auf einem schreibgeschützten Datenträger aufbewahren.
- Mit Hilfe der so gewonnenen Daten lassen sich Veränderungen am Dateisystem feststellen (evtl. nur mit Hilfe eines Rescue-Systems)

Backup

- Übersicht über die wichtigsten Kommandos:

Kommando	Erklärung
tar	Archivieren und Installieren von Dateien
cpio	Kopieren von Dateien auf die Standardausgabe
dd	Low-level Kopie (Byte für Byte)
dump	Erzeugt ein Backup einer Partition
restore	Einspielen von Backups, die mit dump erzeugt wurden

- regelmäßige Backups sind essentiell (z.B. bei Sicherheitsproblemen)
- Day 0, komplett und inkrementell (z.B. mit tar)
- Vollständige Backups sollten aufgehoben werden, um frühere Systemmanipulationen nachvollziehen zu können (z.B. monatliche Backups)
- Auf wechselnde Medien schreiben!
- Wiederherstellung testen

Backup

Was man bei Backups unter Unix bedenken sollte:

- ◆ Bei einem Backup werden i.d.R. die Zeitstempel in den Inodes geändert. Nur beim Low-level Zugriff bzw. bei einem readonly gemounteten Dateisystem bleiben sie unverändert.
- ◆ Beim dateiweisen Kopieren benötigt man durch „sparse files“ auf dem Zielmedium mehr Platz (ls -l und ls -s).
- ◆ Das Verzeichnis /etc sollte immer komplett gebackuped werden.
- ◆ Die Backupmedien müssen immer sicher verwahrt bzw. vernichtet werden (Medien verschlüsseln?).

Zugangskontrolle über das Paßwort

Probleme:

- Kenntnis eines Paßworts verschafft den Zugang zum System und definiert dort die Rechte
- Offene Accounts erlauben den Zugang ohne Paßwort (bei SGI üblich)
- Sniffer im Netz können das Paßwort mithören, falls es im Klartext übertragen wird (telnet, ftp, pop, ...)
- Ist jemand im Besitz des Paßwortfiles, so kann er versuchen mittels Crack weitere Paßworte zu ermitteln

Elemente des Paßwortfiles

Feld	Funktion	Beschreibung
1	Login Name	Login-Name des Benutzer, maximal 8 Zeichen
2	Password	Verschlüsseltes Paßwort.
3	User ID	Benutzernummer zur Identifikation im System (UID). Anzeige mit id .
4	Group ID	Gruppennummer des Benutzers (GID) gestattet Gruppenzugriff auf Files.
5	Personal	Freitext Beschreibung des Benutzers.
6	Login directory	Pfad des Login-Verzeichnisses des Benutzers
7	Login shell	Login-Shell des Benutzers, kann mit chsh vom Benutzer geändert werden.

Wichtig:

- ◆ keine offenen Accounts
- ◆ UID eindeutig vergeben
- ◆ UID=0 gibt Rootrechte

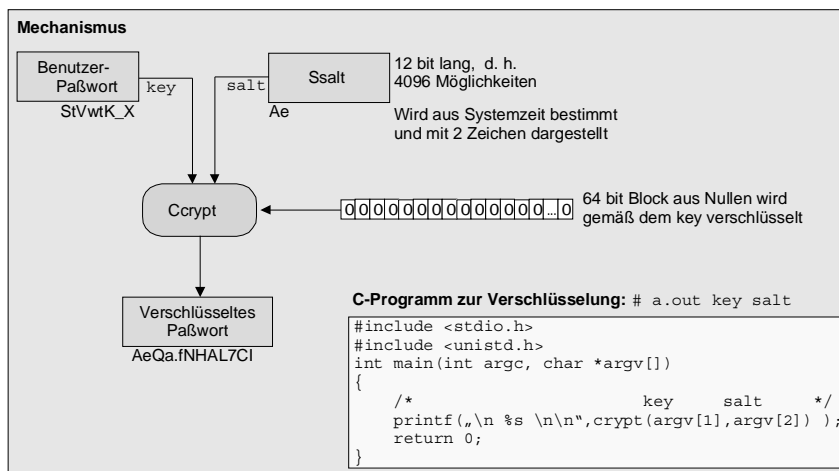
Empfehlung:

- ◆ regelmäßiger Konsistenz-check mit z.B. TARA

cat /etc/passwd

```
root:gw1oNN0pIlsFE:0:0:Superuser Account:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:*:2:2:Daemon:/sbin:/bin/bash
```

Wie entsteht das verschlüsselte Paßwort



Gefahren durch Crack

- Verschlüsselt Worte und Kombinationen von Worten und vergleicht die Ergebnisse mit den Einträgen in /etc/passwd.
- Crack verwendet Wörterbücher und einfache, aber bei Benutzern übliche Variationen wie:
 - ◆ i,l,t nach 1
 - ◆ E nach 3
 - ◆ o nach 0 (NULL)
- Weiterhin wird das Anhängen von Ziffern und Spiegeln erkannt.
- Für ein Paßwort mit 6 Großbuchstaben benötigt Crack weniger als 4 Stunden bei einem Brute-Force Angriff

Schutz des Paßwortfiles

Tips zum Paßwortfile:

- Regelmäßig Paßworte mit Crack testen
- Password-aging einsetzen?
- Benutzer zwingen Sonderzeichen zu verwenden
- Zugriff auf /etc/passwd für tftp und ftp verbieten
- Einsatz von Shadow Password Files da lokale Benutzer das Paßwortfile lesen können
- suchen nach inkonsistenten (UID=0) oder offenen Accounts

```
# awk -F: '{ if ($2=="") print $1 }' /etc/passwd
```

```
# awk -F: '{ if ($3=="0") print $1 }' /etc/passwd
```

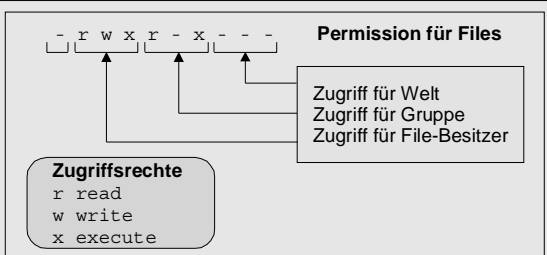

Zugriffsrechte

Permission bits für chmod

octal	binary	perms
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

Zuordnung für umask

octal	binary	perms
0	000	rwx
1	001	rw-
2	010	r-x
3	011	r--
4	100	-wx
5	101	-w-
6	110	--x
7	111	---



Permissions für Verzeichnisse

--x = search bit: Erlaubt das Betreten, aber kein ls
r-x = erlaubt das Betreten und Anzeigen

Veränderung der Zugriffsrechte mit chmod

```
# chmod [-fR] absolute_mode file
# chmod [-fR] [who]+permission... file...
# chmod [-fR] [who]-permission... file...
# chmod [-fR] [who]=[permission...] file...
```

Spezielle Permissions: SUID root

`- r w s r - s r - t`

sticky program
program is a SGID
program is a SUID

Das SUID Bit

s = SUID (octal = 4000)
s = SGID (octal = 2000)
t = sticky bit (octal = 1000)

Welche SUID-Programme sind gefährlich?

- > im Prinzip alle
- > insbesondere Programme wie sendmail, wo es immer wieder neue Lücken gibt

Beispiel eines einfachen Mißbrauchs

- > Eingeloggter Superuser läßt sein Terminal kurz unbeaufsichtigt:


```
#root> cp /bin/ksh /tmp/.my -su -sh
#root> chmod 4755 /tmp/.my -su -sh
```
- > d. h. 2 Befehle genügen, um sich längerfristig root-Privilegien zu verschaffen

Spezielle Permissions: SUID root

Funktionalität und Gefährlichkeit

- Programme mit SUID-root laufen beim Ablauf mit den Privilegien von root, anders als „normale“ Programme, die unter der UID dessen laufen, der das Programm aufruft.
- Notwendig ist der SUID-Mechanismus für viele Programme, z. B. das Programm passwd (Ändern des Paßworts).
- Die Gefahr besteht in der „unsauberen“ Ausführung der Programme, die dann den root-Zugang eröffnen (z. B. Buffer overflow, IFS-Bug).
- Selbst nach vielen Jahren werden oft noch Schwachstellen in SUID-Programmen gefunden.
- Shell-Skripten dürfen auf keinen Fall mit SUID-root versehen werden. (Shell-Escape!)

SUID-/GUID-Programme unter SuSE Linux 7.3

/bin/su	/usr/bin/rsh	/usr/bin/newgrp	/usr/X11R6/bin/xlock-mesa
/bin/ping	/usr/bin/ssh	/usr/bin/passwd	/usr/X11R6/bin/xlock
/bin/eject	/usr/bin/bing	/usr/bin/gpasswd	/usr/X11R6/bin/xterm
/bin/mount	/usr/bin/chfn	/usr/bin/rlogin	/usr/X11R6/bin/v4l-conf
/bin/ping6	/usr/bin/chsh	/usr/bin/screen	/usr/X11R6/bin/XFree86
/bin/umount	/usr/bin/sudo	/usr/lib/mc/bin/cons.saver	/sbin/unix_chkpwd
/opt/kde2/bin/konsole	/usr/bin/wall	/usr/lib/pt_chown	
/opt/kde2/bin/artswrapper	/usr/bin/lppasswd	/usr/sbin/pppd	
/opt/kde2/bin/kdesud	/usr/bin/crontab	/usr/sbin/traceroute6	
/opt/kde2/bin/kreatecd	/usr/bin/chage	/usr/sbin/traceroute	
/opt/kde2/bin/kcheckpass	/usr/bin/mandb	/usr/sbin/sendmail	
/opt/kde2/bin/konsole_grantpty	/usr/bin/write	/usr/sbin/suexec	
/usr/bin/at	/usr/bin/ziptool	/usr/X11R6/bin/dga	
/usr/bin/rcp	/usr/bin/expiry	/usr/X11R6/bin/Xwrapper	

Suche nach Files und Verzeichnissen

- Welt-/ Gruppenschreibbare Verzeichnisse sind nur für /tmp, /usr/tmp etc. sinnvoll und notwendig.

```
# find / -perm -2 \! -type d -print      für Welt schreibbar  
# find / -perm -20 \! -type d -print    für Gruppe schreibbar
```
- SUID / SGID-Berechtigungen sollten nur bei wohldefinierten Programmen vorhanden sein.

```
# find / \( -perm -4000 -o -perm -2000 \) -type f -print
```
- Sinnvollerweise generiert man bei jedem Suchlauf Listen und vergleicht diese mit diff.
- externe Filesysteme sollten mit -nosuid oder gleich mit -noexec gemountet werden.

Permissions von Special Files

- Device-Files bilden die Schnittstelle des Kernels zu den einzelnen Hardware-Devices.
- Lese- und Schreibrechte auf ein Special File bedeuten entsprechende Rechte auf das Device
- Beispielsweise können über /dev/kmem Systemaktivitäten abgehört und modifiziert werden oder über /dev/hda low-level auf die Festplatte zugegriffen werden.
- Special Files sind auch außerhalb von /dev gültig!

```
# find / \( -type c -o -type b \) -exec ls -l {} \;
```

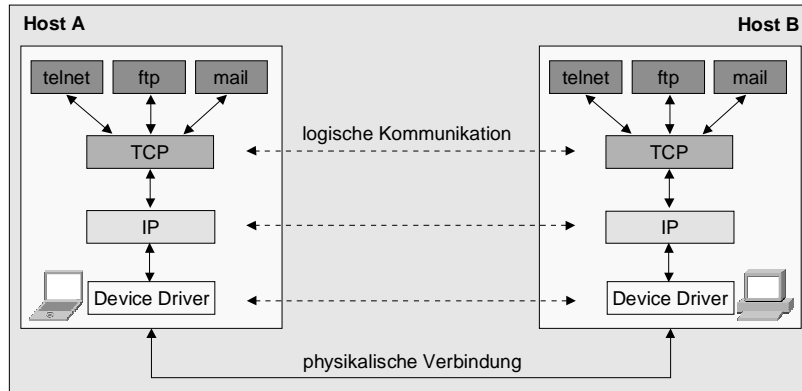
crontab und at

- crontab und at ermöglichen die periodische bzw. einmalige Ausführung von Kommandos zu bestimmten Zeiten
- Bei falschen Berechtigungen auf `/var/spool/cron/tabs` oder `/var/spool/atjobs` können lokale Benutzer dort root ein Script unterschieben
- Evtl. at deaktivieren und crontab über `crontab.allow` bzw. `crontab.deny` absichern

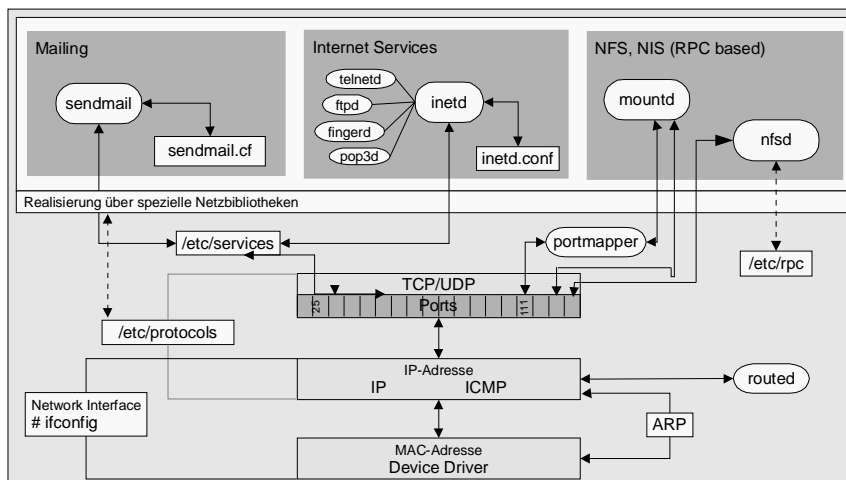
Netzwerksicherheit eines Rechners

- TCP-IP
- Gefahren aus dem Netz
- Anfällige Dienste
- geschwätzige Dienste
- Tcp-Wrapper und PD-Portmapper
- Personal Firewall

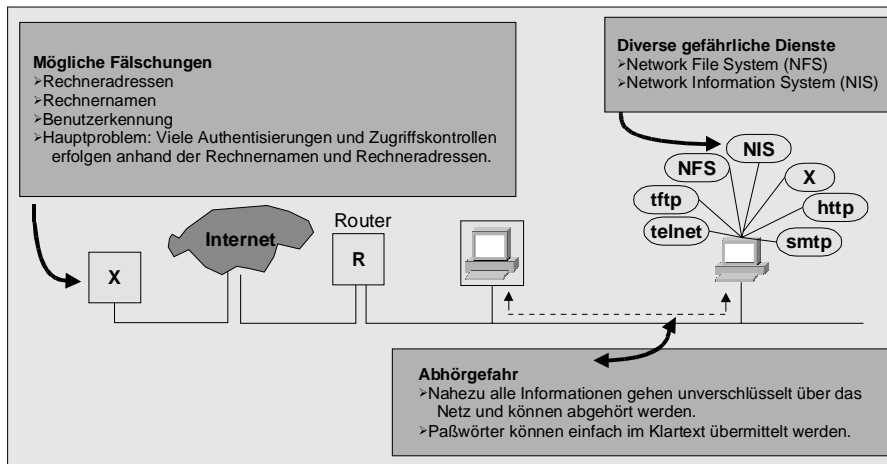
TCP-IP Protokoll



Unix-Implementierung der Netzwerkdienste



Übersicht: Gefahren aus dem Netz



Denial of Service

- der Rechner wird vorübergehend in seiner Funktionstüchtigkeit stark beeinträchtigt
- es kann zu einem Absturz kommen
- es werden (i.d.R.) keine Rechte auf dem Rechner erlangt
- Gegenmaßnahmen
 - ◆ Patches einspielen
 - ◆ Pakete am Router stoppen
- Typische Mechanismen
 - ◆ Überlastung des Rechners durch Spam mit SYNs oder UDP-Paketen
 - ◆ Ausnutzung von Implementierungsfehlern (speziellen Paketen, Fragmentierung)

Sniffer

- Viele Informationen gehen unverschlüsselt übers Netz und können leicht abgehört werden (z.B. Benutzername und Paßwort)
- Durch die abgehörten Login-Informationen hat man Zugriff auf die entsprechenden Ressourcen
- Gegenmaßnahmen:
 - ◆ Paßwörter nur über „sichere“ Netze verwenden
 - ◆ Versuchen das lokale Netz gegen Sniffer zu sichern
 - ◆ Einsatz z.B. von Secure Shell, https, imaps etc.
- Sniffer sind im Netz kaum zu erkennen (da passiver Angriff)
- Sniffer können auch über einen Switch hinweg mithören
- Es gibt oft keine Alternativen zu Klartext-Paßwort

IP-Spoofing

- Fälschung von IP-Adressen über Router-Grenzen hinweg
- Ausnutzung von Diensten über eine gefälschte IP-Authentisierung
- Oft DOS-Attacke auf vorgegebenen Host nötig
- Source-Routed Frames ermöglichen Angriffe aus dem Internet
- Verwundbar: alle Dienste mit Authentifizierung über IP
(r-Dienste, NFS, TCP-Wrapper, X, ...)
- Ermöglicht Connection Hijacking und Command Insertion
- Gegenmaßnahme: Routerkonfiguration um Source-Routed-Frames und gespoofte IP-Adressen abzuweisen

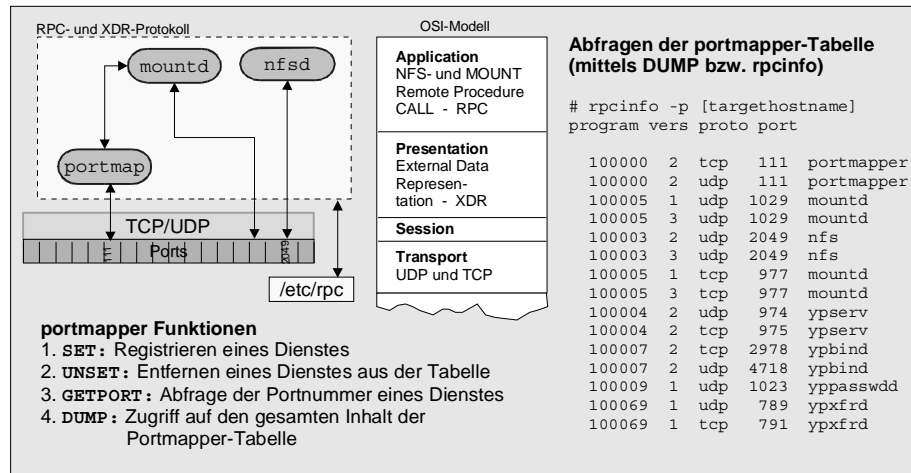
Angriff über Bande

- DNS-Cache Poisoning:
 - ◆ Füttern von Nameservern mit falschen Informationen
- direkte Manipulation von Nameservern
- Man-in-the-Middle Attacken
 - ◆ bei Serverdiensten
 - ◆ Routing
- Gegenmaßnahmen:
 - ◆ Verwendung von Kryptographisch abgesicherten Protokollen mit bekannten Zertifikaten der Gegenstelle
 - ◆ Eintragen wichtiger IP-Adressen in eigene /etc/hosts Datei

Absicherung des X-Servers

- Möglichkeiten und Probleme
 - ◆ Der X-Server kontrolliert Bildschirm, Maus und Tastatur
 - ◆ Ist der X-Server ungesichert übers Netz erreichbar, können alle Events mitprotokolliert werden bzw. externe Eingaben eingeschleußt werden
- Absicherung:
 - ◆ Hostbezogen: xhost [+|-] [hostname] Anfällig für IP-Spoofing
 - ◆ Benutzerbezogen: xauth magic cookies Cookie kann abgehört werden
 - ◆ Einsatz von SSH-Tunneln

RPC-basierende Dienste und portmapper



Sicherheitsprobleme des portmappers

- Mittels rpcinfo können die RPC-Portnummern (weltweit) abgefragt werden
- Mit UNSET kann versucht werden, Dienste aus der Registrierungstabelle zu entfernen (d.h. DOS Angriff)
- Die meisten Portmapper ermöglichen die lokale Weiterleitung einer Anfrage. Diese werden dann wie lokale Anfragen (mit entsprechenden Rechten) behandelt
- Verzicht auf portmapper wegen z.B. NFS und NIS kaum möglich
- Abhilfe durch Einsatz eines möglichst sicheren Portmappers
- Am Router den Port 111 sperren

Sicherheitsprobleme von NIS

- Ermöglicht zentrale Verwaltung eines Clusters
- Ist konzeptionell unsicher:
 - ◆ arbeitet auf dynamischen Ports und ist deshalb schwer zu filtern
 - ◆ Zugriffskontrolle nur über NIS-Domainname
 - ◆ die Kenntnis des NIS-Domainnamens erlaubt Zugriff auf alle NIS-Daten, d.h. insbesondere auf das Paßwortfile
 - ◆ die Kenntnis des NIS-Domainnamens erlaubt fremden Rechnern sich in den Verbund einzuhängen
- Wichtig: NIS-Domainname sollte nichts mit Netzdomainnamen gemeinsam haben

Schutz von NFS

- Ermöglicht zentrale Datenhaltung
- /etc/exports verwendet IP/Hostname basierte Authentifizierung
- Zugriff auf Mountpoint über leicht zu erratendes statisches Filehandle (/etc/exports ist bei Kenntnis des Filehandles wirkungslos)
- Vorsichtsmaßnahmen:
 - ◆ mit fsrand bessere Filehandles generieren
 - ◆ nur wenige Filesysteme (möglichst) readonly exportieren
 - ◆ Server darf keinen Rootzugriff durch den Client erlauben (root_sqash)
 - ◆ der Client sollte das Filesystem mit -o nosuid,nodev mounten
 - ◆ Sperren des Ports 2049 am Router
 - ◆ Einschränkung des Portmapper-Zugriffs

Geschwätzige Dienste

- diverse Dienste liefern Informationen über System, Benutzer, Prozesse und Netzverbindungen
- Diese für den lokalen Gebrauch wichtigen Informationen erleichtern einen Angriff ungemein.
- Dienste die über den inetd gestartet werden:
netstat, finger, systat (telnet <host> <dienst>)
- Weitere mögliche „Lecks“:
 - ◆ showmount -e host Gibt Informationen über exportierte Filesysteme
 - ◆ rpcinfo - host Liefert Liste der angebotenen RPC-Dienste

TCP-Wrapper und PD-Portmapper

- TCP-Wrapper
 - ◆ wird in der /etc/inetd.conf dem inetd-Daemon vorgeschaltet
 - ◆ bietet eine erweiterte Logging Funktionalität
 - ◆ Bietet eine IP-basierte Zugriffskontrolle
- PD-Portmapper
 - ◆ beseitigt insbesondere die Weiterleitung von Anfragen
- Beide verwenden /etc/hosts.allow und /etc/hosts.deny zur Zugriffskontrolle

Personal Firewall

- Kann einen vorgeschalteten Firewall ergänzen
- Kann dienstunabhängig lokal die gefährlichsten Ports sperren
- Kann Zugriffe auf Ports über syslog protokollieren
- für Linux, BSD und Solaris 8 verfügbar
- Hauptprobleme:
 - ◆ Aufstellen geeigneter Regeln
 - ◆ Überprüfen der Regeln
 - ◆ hilft wenig gegen Spoofing

Einschränkungen für ROOT

- Problem:
 - ◆ root-account unter Unix hat uneingeschränkten Zugriff auf das System
 - ◆ ein Angreifer mit Rootberechtigung kann das System verändern (Log-/Konfigurationsdateien ändern/löschen, Prozesse starten, Kernel Module laden, ...)
 - ◆ Wie kann man ein System vor noch unbekanntem Angriffen schützen?
- Lösung:
 - ◆ Verwenden von „chattr“ im ext2 und ext3 Dateisystemen
 - ◆ Linux Intrusion Detection System (LIDS) Kernel Patch
 - ◆ Saint Jude / Saint Michael

Einschränkungen für ROOT

- Linux Intrusion Detection System (LIDS) Kernel Patch
- Ermöglicht:
 - ◆ Versiegeln des Kernels
 - ◆ Kernelbasierte ACLs für Dateien und Prozesse
 - ◆ Kernelbasiertes Erkennen von Portscans
 - ◆ Schutzmechanismen werden beim Booten aktiviert und gelten auch für root
- Problem:
 - ◆ Schwierig einzustellen

Einschränkungen für ROOT

- Saint Jude / Saint Michael sollen ein System gegen unbekannte Gefahren schützen
- Saint Jude
 - ◆ Kernel Modul für Linux (2.2, 2.4) und Solaris 8 Sparc (32/64 Bit)
 - ◆ Erkennt und verhindert Root-Exploits nach Trainingsphase durch das Beobachten von Privilegänderungen
- Saint Michael
 - ◆ Kernel Modul für Linux 2.4
 - ◆ soll Kernel Module Rootkits erkennen
 - ◆ soll Kernelstrukturen wiederherstellen

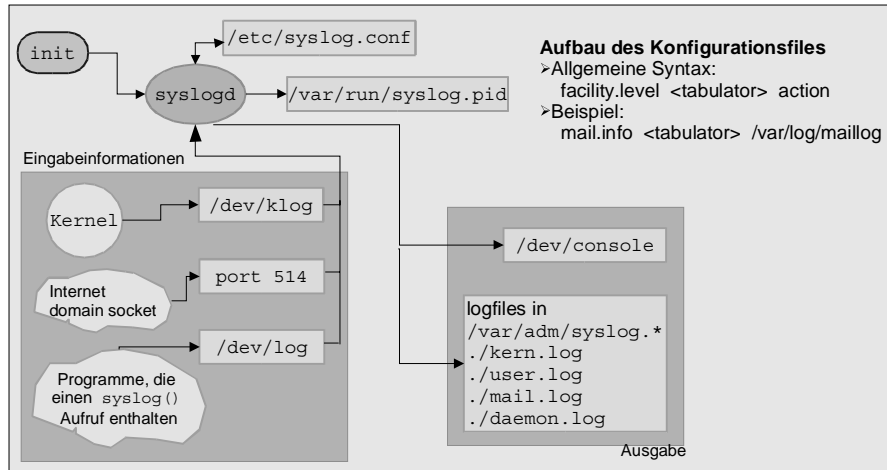
Erkennen von Vorfällen

- Größtes Problem: Die unbemerkte Modifikation des Systems
- Möglichkeiten Angriffe (und Modifikationen) zu erkennen:
 - ◆ lokales und zentrales Logging mit automatischer Auswertung und Benachrichtigung
 - ◆ regelmäßige Überprüfung der Filesystemintegrität (rpm -V, md5sum, tripwire)
 - ◆ lokale und remote Überprüfung der Rechnerkonfiguration (netstat, Portscans)
 - ◆ Analyse des Netzwerkdatenverkehrs
 - ◆ Host- und netzwerkbasierende Intrusion Detection Systeme
 - ◆ Beobachten von Ports auf Portscans
 - ◆ Überprüfen von Programmen mit „strings“ oder „strace“
 - ◆ chkrootkit

Analyse der Logdateien

- Diverse Logdateien
 - ◆ utmp
 - ◆ wtmp
 - ◆ lastlog
 - ◆ acct
 - ◆ sulog
 - Protokollierung über syslog
- Problem:
- Von Hand zu mühsam
 - Wonach soll gesucht werden?
- Lösung: Swatch oder Logsurfer

Syslog-Mechanismus



Aufbau des Konfigurationsfiles
 >Allgemeine Syntax:
 facility.level <tabulator> action
 >Beispiel:
 mail.info <tabulator> /var/log/maillog

Syslog-Konfiguration

Facility	Erklärung	Syntax	Erklärung
kern	generated by the kernel	/filename	Die von Syslog erzeugten Einträge werden in die angegebene Datei geschrieben (-/ unterläßt sync)
user	generated by user processes (default facility)	-filename	
mail	generated by the mail system	@hostname	syslogd leitet die Einträge an die angegebenen Rechner weiter
daemon	generated by system daemons	@ip-adresse	
auth	generated by the authentication system: login, su, etc.	user1,user2	Die Meldungen werden an die angegebenen Benutzer ausgegeben, falls diese eingeloggt sind
lpr	generated by the printer spooling system	*	Die Meldung geht an alle eingeloggten Benutzer
local0 ... local7	reserved for local use	named pipe	Die Meldung wird an die named Pipe (mkfifo) gegeben

priority	Erklärung
LOG_EMERG	a panic condition reported to all users
LOG_ALERT	a condition to be corrected immediately
LOG_CRIT	critical condition
LOG_ERR	errors
LOG_WARNING	warning messages
LOG_NOTICE	condition requiring special handling
LOG_INFO	general information messages
LOG_DEBUG	containing information useful for dbgging

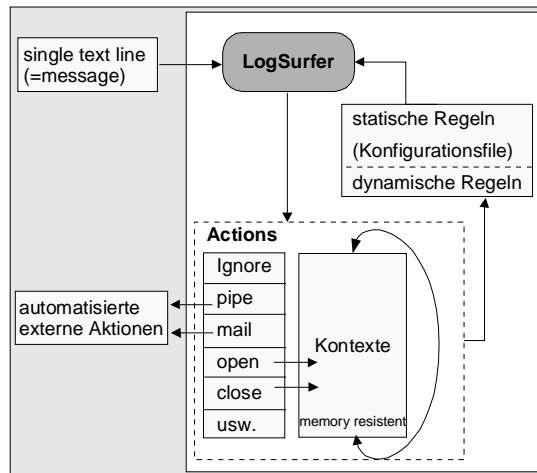
Zentrale Logdateien

- >Vorteile
zentrale Datenhaltung und Auswertung
- >Nachteile
alle Informationen gehen über das Netz.
Der Logrechner sollte gesichert werden.
Anzahl der Logeinträge.

Auswertung mit LogSurfer

■ Grundprinzip

- ◆ LogSurfer vergleicht Textzeilen (aus den logfiles) mit Regeln.
- ◆ Die Regeln sind aus „regular expressions“ aufgebaut.
- ◆ Das System veranlaßt anhand des Regelsatzes diverse Aktivitäten.
- ◆ Im Gegensatz zu Swatch (oder grep) arbeitet LogSurfer kontextabhängig.



tripwire

Ziel: Festhalten aller Informationen über das Filesystem und Erkennung von Veränderungen

Funktionsweise:

- tripwire generiert nach den Vorgaben einer Konfigurationsdatei eine Datenbank mit Zugriffszeiten, Inode, Größe, Besitzer und mehreren kryptographischen Checksummen einer jeden Datei
- Über die Konfigurationsdatei können die zu sammelnden Informationen für jede Datei bzw. Verzeichnis einzeln festgelegt werden.
- Nach Erstellung der Datenbank kann tripwire die Differenzen zum aktuellen Status ausgeben

tripwire

- Zur Sicherheit muß tripwire, die Datenbank und die zugrunde liegenden Konfigurationsdatei auf einem „sicheren“ Medium gespeichert werden.
- Tripwire sollte statisch gelinkt sein
- Nachteil: Konfiguration ist aufwendig da sich viele Dateien im normalen Betrieb ändern
- Open Source Alternative: AIDE

lokaler Konfigurationscheck

- Rechnerkonfiguration ist komplex und zeitaufwändig
- Es können Flüchtigkeitsfehler unterlaufen
- „korrekte“ Konfiguration ist nicht trivial erreichbar
- Abhilfe durch Tools wie TARA, die scriptgesteuert das System auf bekannte Schwachstellen untersuchen und einen Report abliefern
- TARA ist einfach zu Bedienen und liefert gute Erklärungen und Vorschläge zu gefundenen Problemen

remote Überprüfung

- Gründe für eine Remote Überprüfung:
 - ◆ Gezielte scriptgesteuerte Suche nach Schwächen von Netzwerkdiensten
 - ◆ System kann manipuliert sein und zeigt mit Bordmitteln nicht mehr alle offenen Ports an
- Tools:
 - ◆ SAINT
 - ◆ SARA
 - ◆ Nessus
 - ◆ nmap

Analyse des Netzverkehrs

- Bei Netzwerkproblemen oder der Analyse von verdächtigem Datenverkehr
- tcpdump
- ethereal

Bearbeiten eines Einbruchs

- Systemintegrität unklar
- Rootkit verhindert Analyse mit Bordmitteln (versteckte Verzeichnisse, Dateien, Prozesse oder Ports)
- Auffälliger sind meist Symptome wie hohe Last, volles Dateisystem oder eine hohe Netzlast)
- Es gibt kein „Rezept“ für weitere Schritte

Bearbeiten eines Einbruchs

- Allgemeine Hinweise:
 - ◆ Logbuch zur späteren Nachvollziehbarkeit von Aktionen führen. Bei Arbeiten am Rechner den Befehl **script** verwenden.
 - ◆ Schritte welche Daten verändern nur nach guter Überlegung angehen (der erste Bearbeiter hat die größten Chancen Beweise zu sichern – oder zu vernichten).
 - ◆ Ein dediziertes Analysesystem (evtl. Laptop) ist hilfreich.
- Strategische Ziele:
 - ◆ Alle betroffenen Rechner ausfindig machen
 - ◆ Den Angriff nachvollziehbar machen
 - ◆ Benachrichtigung an weitere Betroffene
 - ◆ falls rechtliche Schritte angedacht sind eine korrekte und komplette Datensicherung

Arbeiten vor einem Shutdown

- Potentieller Selbstzerstörungsmechanismus in Shutdown integriert.
- VOR Shutdown möglichst viele Informationen aus dem laufendem System sammeln:
 - ◆ last / w
 - ◆ ps auxwww bzw. ps elfwww
 - ◆ lsof
 - ◆ ltrace, strace auf verdächtige Programme
 - ◆ find und ls -lat (ändern Timestamps!)
 - ◆ netstat -a
- Potentiell können die genannten Tools und auch Shared Libraries oder Kernelmodule durch einen Rootkit ersetzt sein

Analyse des gehackten Systems

- wichtig: lowlevel Kopie mit z.B. dd anfertigen und nur mit der Kopie arbeiten. Original sicherstellen.
- Medium immer readonly und -o nosuid,nodev mounten
- Logbuch führen
- Verdächtige Programme mit „strings“ bearbeiten
- Alles weitere hängt vom speziellen Fall ab

Informationsquellen

- **www.cert.org:**
 - ◆ Statistiken:
http://www.cert.org/stats/cert_stats.html
 - ◆ Unix Security Checklist:
http://www.cert.org/tech_tips/AUSCERT_checklist2.0.html
 - ◆ Steps for Recovering from an Unix or NT Compromise:
http://www.cert.org/tech_tips/win-UNIX-system_compromise.html
 - ◆ Unix Cnfiguration Guidelines:
http://www.cert.org/tech_tips/unix_configuration_guidelines.html
 - ◆ Aktuelle Angriffe auf:
 - ◆ SSH1 CRC-32 compensation attack detector
 - ◆ Apache/mod_ssl („slapper“ Wurm)

Informationsquellen

- **www.sans.org:**
 - ◆ Information Security Reading Room
<http://rr.sans.org>
 - ◆ virtually free tools (analyzer, ethereal, nmap)
http://rr.sans.org/tools/ns_software.php
http://rr.sans.org/tools/free_tools.php
 - ◆ Using Snort as an IDS and Network Monitor in Linux
<http://rr.sans.org/intrusion/monitor.php>
 - ◆ Building Network Intrusion Detection Systems Using Open Source Software
http://rr.sans.org/intrusion/net_id2.php
 - ◆ Nailing the Intruder
<http://rr.sans.org/incident/nailing.php>

Informationsquellen

- ◆ The Twenty Most Critical Internet Security Vulnerabilities
<http://www.sans.org/top20.htm>
- Lance' Security Papers (u.a. armoring Linux/Solaris/NT)
<http://www.enteract.com/~lspitz/>
- Basic Steps in Forensic Analysis of Unix Systems:
<http://staff.washington.edu/dittrich/misc/forensics/>
- Das Honeynet Projekt
<http://www.honeynet.org>
- Security Focus (Bugtraq)
<http://www.securityfocus.com>

Informationsquellen

- Practical UNIX Security, Simson Garfinkel and Gene Spafford, 1996 (2. Auflage), O'Reilly & Asc.
- Firewalls and Internet Security, Repelling the Wily Hacker, William R. Cheswick and Steven M. Bellovin, 1994, Addison Wesley
- Hacking Exposed (3rd. Ed.), McClure, Scambray and Kurtz, 2001, McGraw-Hill (www.hackingexposed.com)
- Hacking Linux Exposed, Hatch, Lee and Kurtz, 2001, McGraw-Hill (www.hackinglinuxexposed.com)
- Software: www.freshmeat.net