Grundlagen der Unix-Security

Peter Dieterich

dieterich@rz.uni-wuerzburg.de

Rechenzentrum der Universität Würzburg
Am Hubland, D-97074 Würzburg

RZUW

Inhaltsübersicht

- Einführung und Zielsetzung
- Grundlegende lokale Sicherheitsaspekte
- Gefahren aus dem Netz
- Logdateien erzeugen und auswerten
- Analyse der Rechnerkonfiguration
- Informationsdienste
- Einsatz der Kryptographie
- Diskussion

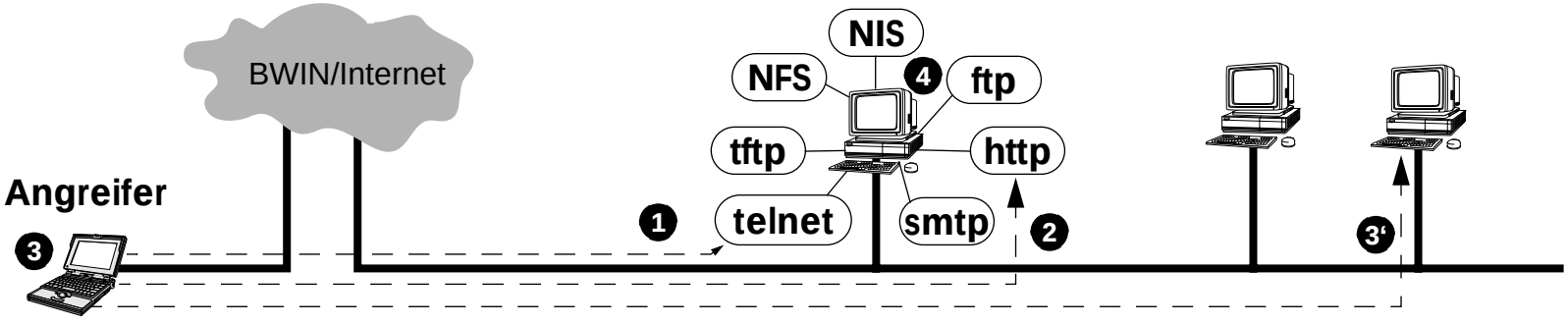
1 Einführung

Übersicht

- Szenario eines Hackerzugriffs
- Ziel der Veranstaltung
- Sicherheit (warum und wieviel)
- Denkanstöße
- Definition der persönlichen Sicherheit
- Welche Gefahren gibt es und woher kommen diese?
- Hochschulnetz: Charakter, Status und Anforderungen
- Konzept für Gegenmaßnahmen
- Problematik der rechtlichen Situation
- Wo gibt es Informationen?
- Aufgaben und Leistungen von DFN-CERT
- Literaturhinweise

Ein typischer Hackerangriff

1. Suche nach offenem Account (z. B. guest bei SGI)
2. Versuch, das Paßwortfile zu erhalten
   (tftp, WWW, NFS, NIS, ...)
3. Cracklauf, um neue Accounts zu bekommen
   (weiter an andere Rechner im Cluster)
4. root-Rechte über exploit-Skripten
   (sendmail, lpr, perl, ...)



Was kann gegen derartige Angriffe unternommen werden?

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?
- Entwurf und Diskussion eines mehrstufigen Konzeptes, das helfen soll, die Sicherheit in einem Universitätsnetz zu verbessern.
- Austausch von Erfahrungen.

Was wird nicht erreicht?

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

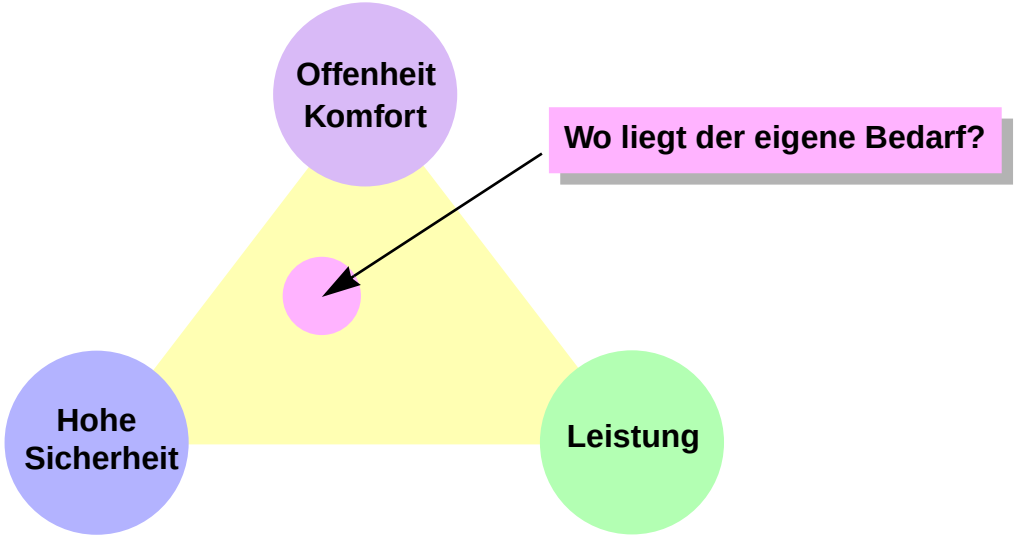
Sicherheit - warum und wieviel?

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
- Leistungsverluste
- Widerspricht a priori dem offenen Charakter eines Hochschulnetzes

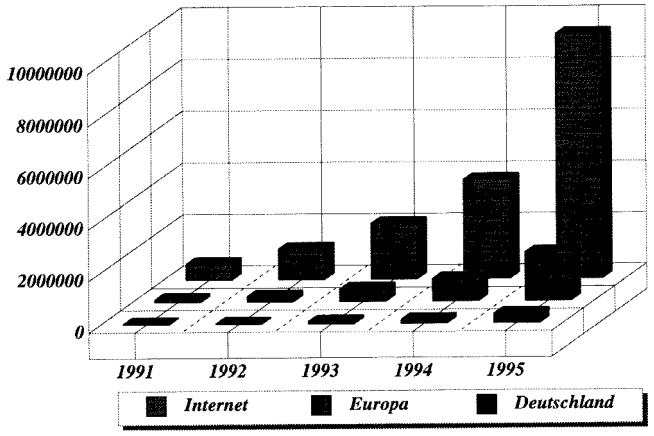


Probleme:

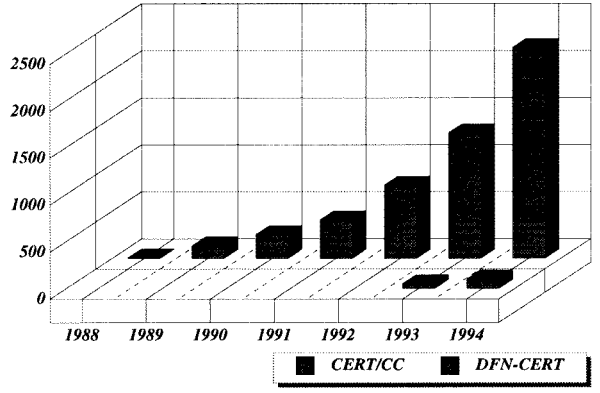
- Entscheidungsfindung
- Konzept
- Realisierung
- Maintenance

Denkanstöße

Wachstum des Internets



Anzahl der Vorfälle



Überlegungen

- ❒ Woher kommen die Angriffe? Außen/Innen
- ❒ Was gibt es für Angriffe? Toolkits zum Einbruch
- ❒ Wie wichtig sind die Daten meiner Rechner?
- ❒ Was kann ich tun und woher bekomme ich Know-how und Informationen?
- ❒ Was ist bisher passiert?

❒ 95 % der bei CERT registrierten Vorfälle erfolgten auf Unix-Plattformen

Definition der persönlichen Sicherheit

Sicherheit als Ziel

- ❒ Risiken erkennen
- ❒ Risiken abwägen
- ❒ Konsequenzen ziehen

Hilfen zur Orientierung

- ❒ Institutionen CERT und DFN-CERT
  `ftp://info.cert.org/pub`
  `ftp://ftp.cert.dfn.de/`
  `http://www.cert.dfn.de/`
  Diese Institutionen erstellen Advisories und Bulletins und geben diverse Sicherheitsinformationen weiter. DFN-CERT veranstaltet Tutorien und Workshops, informiert und berät.
- ❒ Regelmäßiges Lesen von News-Gruppen, z. B.
  `alt.security`
  `comp.security.misc`
  `comp.security.unix`
  Weitere Listen finden sich unter
  `http://www.cert.dfn.de/resource/maillist/`
- ❒ Literaturhinweise:
  Practical UNIX Security, Simson Garfinkel und Gene Spafford, 1996 (2. Auflage), O'Reilly & Asc.
  Firewalls and Internet Security, Repelling the Wily Hacker, William R. Cheswick und Steven M. Bellovin, 1994, Addison Wesley
  Diverse Papers auf dem ftp-Server des DFN-CERT.

Konsequenzen

- ❒ Eigene Risikoanalyse (sehr schwierig, vor allem in einer offenen Hochschulumgebung).
- ❒ „Baseline Protection"
- ❒ Schon einfache Maßnahmen können die Sicherheit stark erhöhen.

Woher kommen die Gefahren?

Gefahren?

- Raubkopien von Software
- Nutzung fremder Ressourcen
- Zugriff auf Kreditkartennummern
- (Kinder-)Pornographie
- Belästigung
- Gefälschte Mails
- Denial of Service

Wer übt Angriffe aus?

- Kriminelle?
- Hacker?
- Spieler?
- Personen des Instituts?

Art der Angriffe

- Nutzung einfacher Löcher („well known bugs“, offene Accounts, „schlecht“ verwaltete Rechner ...)
- Ausnutzung konzeptioneller Schwächen (NFS, NIS ...).
- Ausgereifte Toolkits (im Netz verfügbar) für Sniffing, IP-Spoofing, Hijacking, Trojaner ...
- Zukunft: Noch besser automatisierte Tools, Trojaner via Multimedia-Anwendungen.

Universitätsbereich

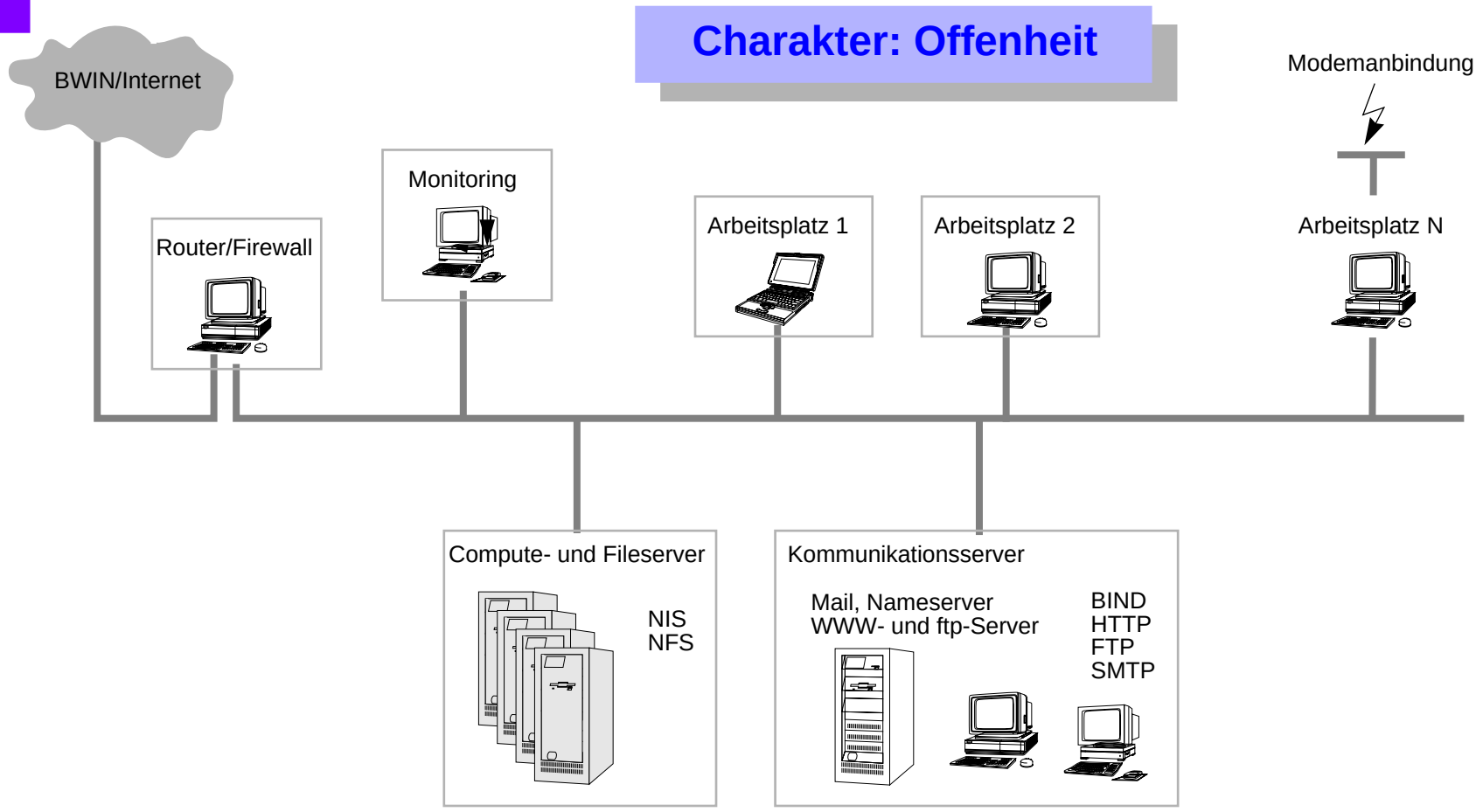
Gefahr von Innen

- Student will die Klausuraufgaben
- Frustrierter Postdoc will das root-Paßwort
- Studenten in den CIP-Pools spielen gerne

Gefahr von Außen

- Daten meist nicht sehr geheim
- Interessant als Sprungbrett

Das Hochschulnetz



RZUW

Suche nach einem Konzept ...

Erhöhung der lokalen Rechnersicherheit

- Physikalische Sicherheit des Rechners und Netzzuganges
- regelmäßige Backups
- Weitergabe von Sicherheitsinformationen an die Systemmanager
- Schließen der wichtigsten Löcher (Paßwortsicherheit, well know bugs, OS-Updates)
- Aufsetzen von C2-Security (falls möglich)
- Einsatz von Public Domain Tools (Tiger, Cops, TCP-Wrapper, PD-Portmapper)

Modifizierung der Routerkonfigurationen

- Verhinderung von IP-Spoofing und Source Routing
- Filtern diverser gefährlicher Ports (problematisch)

Protokollierung

- Aufzeichnung des Netzverkehrs (MoniBox)
- Auditing auf jedem Rechner
- Auswertung mit diversen Tools (Swatch, Logsurfer)

Einsatz kryptographischer Elemente

- PGP zur Absicherung des Mail-Verkehrs
- Secure Shell zur verschlüsselten Kommunikation über IP

Aktivitäten nach einem Einbruch?

Auf dem betroffenen Rechner

- ❒ alle Spuren sichern
- ❒ Veränderungen rückgängig machen oder Neuinstallation

Im betroffenen Netz

- ❒ Suche nach weiteren betroffenen Rechnern (z. B. anhand von Logdateien)

Meldung des Vorfalls

- ❒ an das Rechenzentrum
- ❒ an den Provider, über den der Vorgang ablief (whois-Service)

```
# whois -h whois.ripe.net 130.236.48
```

- ❒ DFN-CERT (jeweils zur weiteren Verfolgung oder Koordination von Vorfällen)
- ❒ an die Rechtsabteilung zur Anzeige

Rechtliche Situation?

- ❒ meist unklar.
  Hinweise finden sich in
  `http://www.rz.uni-wuerzburg.de/netzbericht/`
- ❒ Rolle der Rechenzentren?
- ❒ Rolle der Rechtsabteilung?
- ❒ Macht Verfolgung und Anklage Sinn?

2 Lokale Rechnersicherheit

Übersicht

- Backups:
  Die wichtigsten Kommandos
- Paßwortsicherheit
  Funktionalität, Probleme, gute Paßwörter ...
- Zugriffsrechte
  Permissions, SUID-Programme, Device-Files, Pfade, Integrität der Zugriffsrechte
- Systemintegrität

Backups: Übersicht der wichtigsten Kommandos

! Regelmäßige Backups sind essentiell (auch im Hinblick auf Sicherheitsprobleme) !

Übersicht der wichtigsten Kommandos

Kommando	Erklärung
tar	Archivieren und Installieren von Files.
cpio	Kopieren von Files auf den Standard-Output.
dd	Physikalische Kopie (byte per byte).
dump	Erzeugt Backup von Partitionen.
restore	Einspielen von Backups, die mit dump ausgeführt wurden.

Tip: Vollständige Backups sollten aufgehoben werden, um gegebenenfalls frühere Systemmanipulationen nachvollziehen zu können (z. B. monatliche Backups).

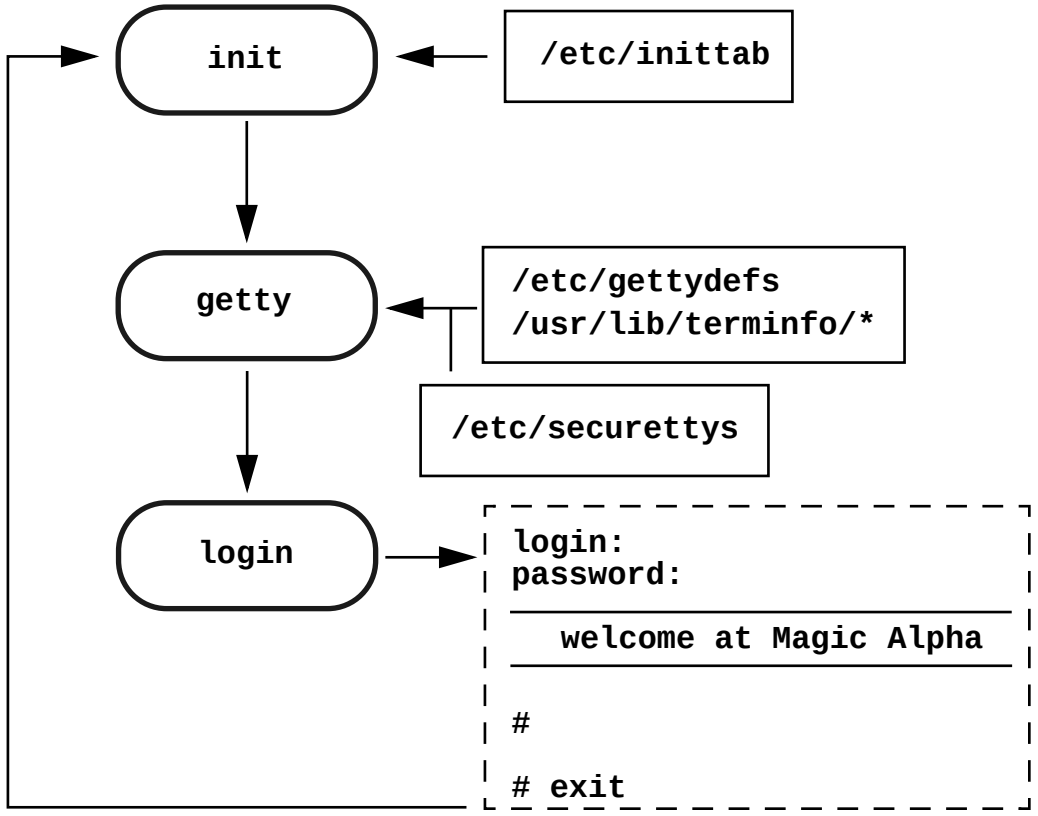
Backups: Übersicht der wichtigsten Kommandos

Übersicht der Herstellerangebote

	AIX	Digital UNIX	HP-UX	IRIX	Linux	Solaris	SunOS
Filesystem (lokal)	backup	dump vdump	dump fbackup	dump Backup	-	ufsdump	dump
(remote)	rdump	rdump	rdump fbackup	rdump Backup	-	ufsdump	rdump
Restauration (local)	restore	restore	restore frecover	restore Restore	-	ufsrestore	restore
(remote)	rrestore	rrestore	rrestore frecover	rrestore Restore	-	ufsrestore	rrestore
Backup Tools	smit (backup)	-	sam fbackup	ctape	-	-	-
Files (user-Kommandos)	tar cpio	tar cpio	tar cpio fbackup	tar cpio bru	tar cpio	tar cpio	tar cpio
Physical	dd	dd	dd	dd	dd	dd	dd

- Standard-Unix-Backups sind „gewöhnungsbedürftig“
- Kommerzielle Lösungen sind meist teuer (Legato Networker)

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, da es in der Regel in Klartext übertragen wird.
- Ist jemand im Besitz des Paßwortfiles, so kann er versuchen, einige Paßwörter mittels Crack zu erraten.

Elemente des Paßwort-Files

Feld	Funktion	Beschreibung
1	Login name	Login-Name des Benutzers, maximal 8 Zeichen lang.
2	Password	Verschlüsseltes Paßwort für den Benutzer. Kann mit dem Befehl **`passwd`** verändert werden.
3	User ID	Benutzernummer wird vom System zur Identifikation benutzt (UID). **`root`** hat die Nummer 0. Anzeige mit **`id`**.
4	Group ID	Gruppennummer des Benutzers (GID) gestattet Gruppenzugriff auf Files.
5	Personal	Einträge zum Benutzer: Name, Anschrift, Telefon. Die Einträge können vom Benutzer mit **`chfn`** verändert werden.
6	Login directory	Name des Login-Verzeichnisses des Benutzers.
7	Login shell	Login-Shell des Benutzers, kann mit **`chsh`** vom Benutzer verändert werden (z. B. `/bin/sh, /bin/csh,...`).

Wichtig:

- ❏ keine offenen Accounts
- ❏ UID und GID eindeutig vergeben
- ❏ UID=0 gibt Systemprivilegien

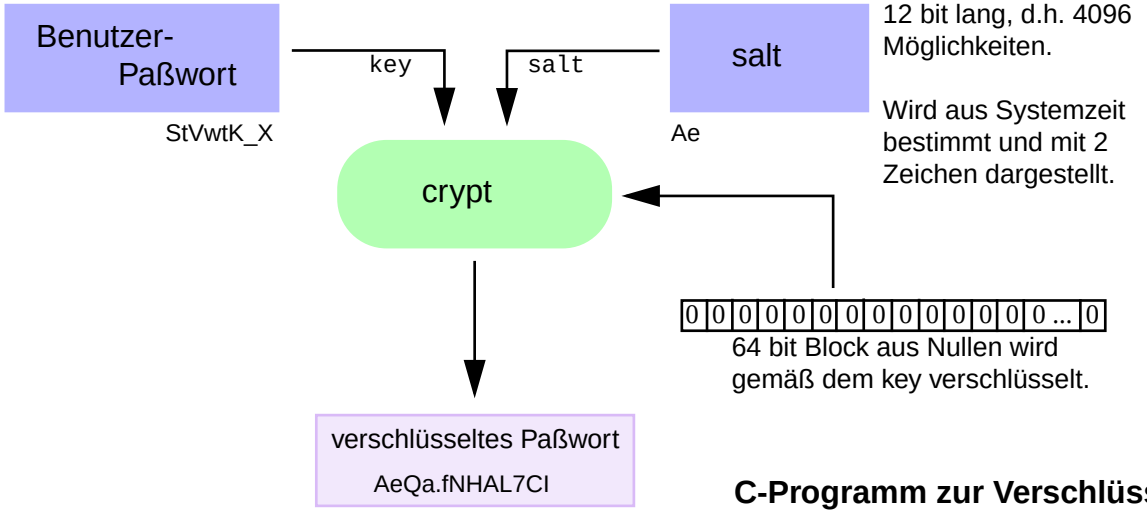
Empfehlung:

- ❏ regelmäßiger Konsistenzcheck des Paßwortfiles z. B. mit Tiger oder Cops

```
# cat /etc/passwd
root:GQpf4ZO726g9i:0:1:system PRIVILEGED account:/:/bin/csh
nobody:*Nologin:65534:65534:anonymous NFS user:/:
nobodyV:*Nologin:60001:60001:anonymous SystemV.4 NFS user:/:
daemon:*:1:1:system background account:/:
bin:*:3:4:system librarian account:/bin:
cron:*:7:14:Cron Subsystem:/usr/adm/cron:
lp:*:8:12:Line Printer Subsystem:/users/lp:
adm:*:10:19:Administration Subsystem:/usr/adm:
rzuw001:GtO9D9e8Uasb.:101:300:Werner Lanke,RZ,5199,:/users/rzuw001:/bin/ksh
```

Wie entsteht das verschlüsselte Paßwort?

Mechanismus



C-Programm zur Verschlüsselung: `# a.out key salt`

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
        /*                              key      salt     */
        printf("\n %s \n\n", crypt(argv[1], argv[2]) );
        return 0;
}
```

Gefahren durch Crack

Das Programm `Crack`

- ❏ verschlüsselt Worte und Kombinationen von Worten und vergleicht diese mit den kodierten Einträgen in `/etc/passwd`.
- ❏ Crack kennt Namen und Worte aus Lexikas.
- ❏ Crack erkennt Umbenennungen:
  - i, l, t nach 1
  - E nach 3
  - o nach 0 (NULL)
  - A nach 4
  - g nach 9
  - Z nach 7
- ❏ Weiterhin wird das Anhängen von Ziffern und Rückwärtsschreiben erkannt.

Konsequenzen

- ❏ Wahl eines möglichst guten Paßwortes; Test z.B. mit **checkpw**.
- ❏ Schutz des Paßwortfiles.

Tips zur Paßwortwahl

- ❏ Auf keinen Fall den Login-Namen verwenden!
- ❏ Keine Namen, Telefonnummern oder existierenden Worte!
- ❏ Verwenden von Sonderzeichen oder beispielsweise den Anfangsbuchstaben, Silben oder Worten eines Textes.

Beispiele für Crack-Läufe

Beispiele für den Rechenzeitbedarf von Crack-Läufen auf einem DEC-Alpha-Compute-Server

Eine optimierte crypt-Routine bearbeitet etwa 20000 Versuche pro Sekunde. Je nach Anzahl der Zeichen, die für das Klartextpaßwort verwendet werden können, ergeben sich die folgenden Zeiten für das Durchlaufen aller Möglichkeiten.

26 Zeichen = Alphabet in Groß- oder Kleinbuchstaben
52 Zeichen = Alphabet in Groß- und Kleinbuchstaben
62 Zeichen = Alphabet und 10 Ziffern

Anzahl der Zeichen	Paßwort länge	Anzahl der Möglichkeiten	Maximale Zeit
128	8	128^8 = 72057594037927936	114246 Jahre
	7	128^7 = 562949953421312	892 Jahre
	6	128^6 = 4398046511104	7 Jahre
95	8	95^8 = 6634204312890625	10518 Jahre
	7	95^7 = 69833729609375	110 Jahre
	6	95^6 = 735091890625	1 Jahr
62	8	62^8 = 218340105584896	346 Jahre
	7	62^7 = 3521614606208	6 Jahre
	6	62^6 = 56800235584	33 Tage
52	8	52^8 = 53459728531456	85 Jahre
	7	52^7 = 1028071702528	2 Jahre
	6	52^6 = 19770609664	11 Tage
26	8	26^8 = 208827064576	121 Tage
	7	26^7 = 8031810176	5 Tage
	6	26^6 = 308915776	4 Stunden

Schutz des Paßwortfiles

Tips zum Paßwort-File

- Unsichere Paßwörter z. B. mit Crack ausfindig machen und den Benutzer darüber informieren.
- Manche Systeme bieten die Möglichkeit, den Benutzer dazu aufzufordern, spezielle Sonderzeichen etc. bei der Paßwort Wahl zu verwenden. Auch das Aufsetzen von password-aging kann von Vorteil sein.
- Sicherstellen, daß auf das File /etc/passwd nicht über tftp zugegriffen werden kann. Hierzu ist die tftp Konfiguration in /etc/inetd.conf entsprechend zu modifizieren. Auch sollte abgewogen werden, ob finger nicht zu viel Informationen liefert.
- Suche nach offenen Accounts, beispielsweise:

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

- Manche Hersteller, z. B. Silicon Graphics, liefern firmenseitig installierte Maschinen mit einer Reihe offener Accounts aus.
- Suche nach inkonsistenten Einträgen, z. B. Benutzern mit UID=0:

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

Welche Probleme bleiben bestehen und welche Ansätze zur Lösung gibt es?

- Lokale User können den Inhalt von /etc/passwd und die verschlüsselten Paßwörter lesen
  - ➜ Aufsetzen eines *Shadow Password Files* (problematisch zusammen mit NIS)
- Abhören der Paßwörter im Netz
  - ➜ Verwendung von .rhosts File (bedingt wiederum weitere Probleme)
  - ➜ Einsatz der *Secure Shell* (ssh) - erlaubt verschlüsselte Kommunikation

Zugriffsrechte

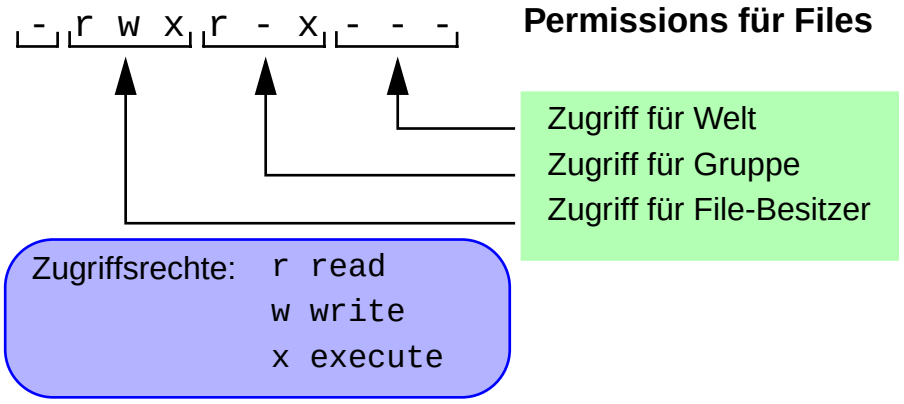
Permission bits für chmod

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

Zuordnung für umask

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

Permissions für Files



Permissions für Verzeichnisse

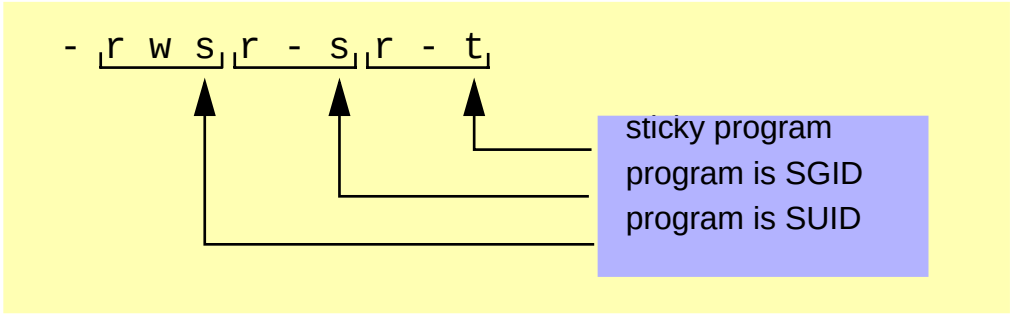
--x = search bit: Erlaubt das Betreten, aber kein ls

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: Set-UID-root-Programme



Das SUID Bit

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

Funktionalität und Gefährlichkeit

- Programme mit SUID-root laufen beim Aufruf mit den Privilegien von root, im Gegensatz zu „normalen“ Programmen, die unter der UID dessen laufen, der das Programm aufruft.
- Notwendig ist der SUID-Mechanismus für viele Programme, beispielsweise das Programm passwd zum Ändern des Paßwortes.
- Die Gefährlichkeit besteht in der „unsauberen“ Ausführung der Programme, die dann den root-Zugang eröffnen (Bsp. 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!).

Welche SUID-Programme sind gefährlich?

- im Prinzip alle
- insbesondere Programme wie senmail, wo es immer wieder neue Lücken gibt.

Beispiel eines einfachen Mißbrauchs

- Der eigeloggte 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.

RZUW

SUID- und SGID-Programme unter Digital UNIX

/sbin/df
/sbin/killall
/sbin/ps
/sbin/ping
/sbin/arp
/usr/sbin/acct/accton
/usr/sbin/wall
/usr/sbin/smtpd
/usr/sbin/sendmail
/usr/sbin/srconfig
/usr/sbin/quot
/usr/sbin/mailq
/usr/sbin/killall
/usr/sbin/fstat
/usr/sbin/netstat
/usr/sbin/trpt
/usr/sbin/traceroute

/usr/sbin/timedc
/usr/sbin/ping
/usr/sbin/arp
/usr/sbin/lpc
/usr/sbin/newaliases
/usr/lib/mh/spop
/usr/opt/s5/sbin/killall
/usr/opt/s5/bin/df
/usr/lbin/slvmod
/usr/lbin/exrecover
/usr/lbin/chgpt
/usr/lbin/expreserve
/usr/lbin/ex3.7preserve
/usr/lbin/lpd
/usr/lbin/ex3.7recover
/usr/bin/X11/dxpause
/usr/bin/X11/xterm

/usr/bin/X11/dxterm
/usr/bin/X11/xload
/usr/bin/X11/xconsole
/usr/bin/X11/dxconsole
/usr/bin/mh/msgchk
/usr/bin/mh/inc
/usr/bin/write
/usr/bin/w
/usr/bin/vmstat
/usr/bin/binmail
/u/usr/bin/chfn
/usr/bin/ps
/usr/bin/pfstat
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/mail

/usr/bin/login
/usr/bin/ipcs
/usr/bin/crontab
/usr/bin/rsh
/usr/bin/tip
/usr/bin/rlogin
/usr/bin/rcp
/usr/bin/ct
/usr/bin/cu
/usr/bin/rdist
/usr/bin/nfsstat
/usr/bin/lprm
/usr/bin/lpr
/usr/bin/lpq
/usr/bin/at
/usr/bin/uptime
/usr/bin/nrdist

Suche nach Files und Verzeichnissen

Welt-schreibbare Verzeichnisse und Files

- Für /tmp, /usr/tmp etc. sinnvoll und notwendig, für andere Verzeichnisse jedoch problematisch, da hier alle Benutzer Files erzeugen und ablegen können.
- Suche beispielsweise mit

```
# find / -perm -2 \! -type l -print      für Welt schreibbar
# find / -perm -20 \! -type l -print     für Gruppen schreibbar
```

Suche nach SUID/SGID-Programmen

- Es sollen nur wohldefinierte Programme mit diesen Permissions auf dem Rechner sein.
- Die Suche erfolgt mit

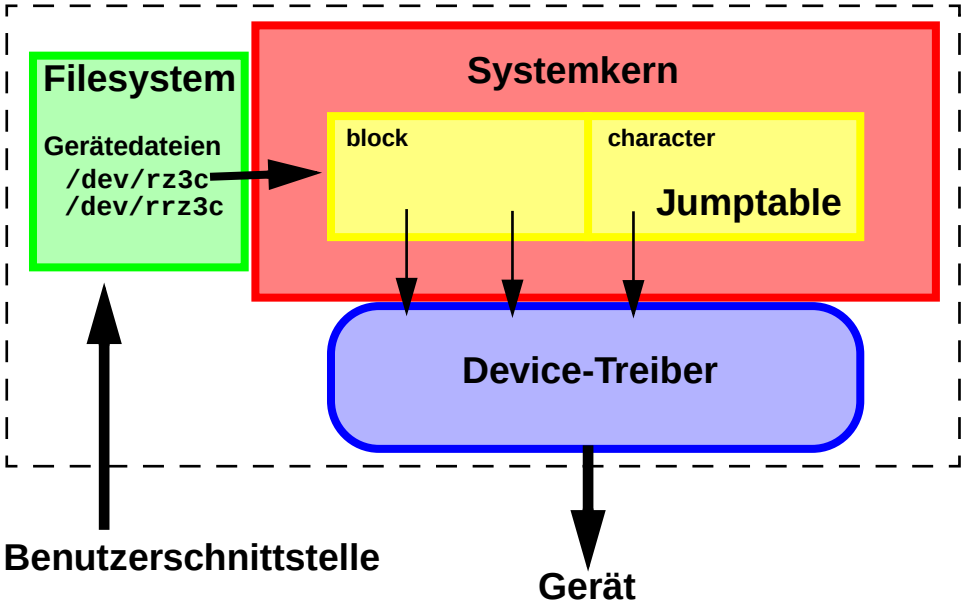
```
# find / \( -perm -4000 -o perm -2000 \) \! -type f -print
```

- Sinnvoll ist es, Listen bei jedem Suchlauf zu generieren und diese mit diff zu vergleichen.
- Filesysteme sollten soweit wie möglich mit der Option -nosuid gemountet werden. Diese verbietet die Ausführung von SUID-Programmen, die in diesem Bereichen abgelegt sind.

Permissions von Special Files

Funktionalität und Problematik

- Device-Files bilden Schnittstellen des Kerns zu den einzelnen Hardware-Devices.
- Lese- und Schreibrecht auf diese Special Files bedeutet auch das Zugriffsrecht auf das Device.
- Beispielsweise könnten (bei falschen Permissions) von jedem Benutzer aus /dev/kmem Systemaktivitäten abgehört und modifiziert werden.



Analyse der Permissions

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

brw-------   1 root     system     8,4098 Jan 23  1994 /dev/rz4c
crw-------   1 root     system     8,4098 Jan 24  1995 /dev/rrz4c
crw-rw-rw-   1 root     system     9,5120 Jan 24  1995 /dev/rmt0h
```

Wichtigkeit und Gefährlichkeit der Pfadwahl

Hinweise zur Pfadwahl

- Es sollten nur eigene und System-Programme ausgeführt werden.
- Der Punkt im Pfad ist gefährlich. Wenn er unbedingt sein soll, dann am Ende des Pfades.
- Der eigene Pfad sollte nicht in ein welt-schreibbares Verzeichnis zeigen.
- Pfade von root sollten nur auf Programme von root zeigen. Der Punkt im Pfad sollte bei root ganz besonders vermieden werden.

Beispiel für die Gefährlichkeit des Punktes im Pfad

- Das folgende Shell-Skript mir dem Namen ls wird z. B. in /tmp abgelegt.

```
#!/bin/sh
(/bin/cp /bin/sh /tmp/.my-secret-shell
 /bin/chown root /tmp/.my-secret-shell
 /bin/chmod 4755 /tmp/.my-secret-shell) 2> /dev/null
rm -f $0
exec /bin/ls "$@"
```

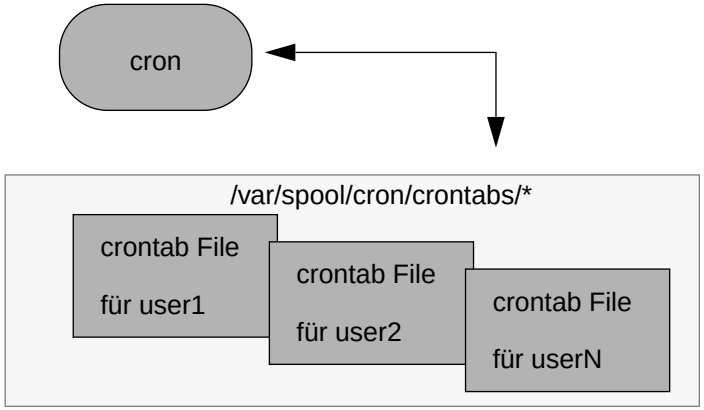
- Falls der Pfad des Superusers den Punkt vor dem Verzeichnis /bin enthält, wird bei den Kommandos

```
# cd /tmp; ls -al
```

das obige Skript abgearbeitet und anschließend vernichtet. Dem Angreifer steht nun die root-Shell /tmp/.my-secret-shell zur Verfügung.

Der crontab-Mechanismus

crontab: Ermöglicht das periodische Ausführen von Kommandos

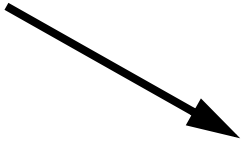


Überprüfen!

- ❏ Welche crontab-Einträge sind vorhanden (v. a. für die privilegierten Accounts)?
- ❏ Sollte eine Absicherung über crontab.allow bzw. crontab.deny erfolgen?

Integrität des Systems

Problem: Unbemerkte Modifikationen des Systems



Vorsichtsmaßnahmen und Möglichkeiten einer Analyse

- Partitionen sollten möglichst nur read only gemountet werden.
- Regelmäßige Backups sollten ausgeführt werden.
- Gleich nach der Installation kann eine Liste aller Files und Verzeichnisse erstellt werden, die dann mittels diff auf Veränderungen im aktuellen System überprüft werden kann.
- Installation von Nicht-Systemsoftware sollte möglichst ohne root-Privilegien erfolgen, um unbeabsichtigte Veränderungen zu vermeiden.
- Bei größeren OS-Release-Wechseln trägt eine Neuinstallation zur Sauberkeit des Systems sicher bei.
- Veränderungen des Systems können beispielsweise mit dem PD-Tool tripwire erkannt werden.

tripwire

siehe auch http://www.cert.dfn.de/infoserv/dib/dib-9304.html

Zielsetzung: Festhalten des Systemstatus (Daten und Permissions) und Erkennung von Veränderungen

Funktionsweise

- tripwire generiert eine Datenbank von einem Filesystembaum. Neben Informationen über Zugriffszeiten, Inode und Größe, werden zwei kryptographische Prüfsummen zu jedem File erstellt.
- In einer Konfigurationsdatei können Verzeichnisse definiert werden, die in die Datenbank aufgenommen werden sollen. Auch die Art der Überprüfung kann festgelegt werden (z. B. read-Zugriff o. ä.).
- Verzeichnisse bzw. deren Inhalte (z. B. von /tmp) können ausgeschlossen werden.
- Die Datenbank sollte auf einem „sicheren“ Medium gespeichert werden.
- Nach dem Aufbau der Datenbank kann tripwire die Differenzen zum aktuellen Status im Vergleich ausgeben.
- Bis zur endgültigen Konfiguration von tripwire vergehen i. d. R. einige Tage, da sich viele Dateien auch im normalen Betrieb verändern, ohne daß dies sicherheitsrelevant wäre.

Sonstiges zur lokalen Sicherheit

Stand des Systems

- Betriebssystemversion sollte möglichst aktuell sein.
- relevante Sicherheitspatches sollten eingespielt sein.

Gefährliche Informationsdienste

- ftp-Server
- http-Server/Client (Java-Skripten)
- mail-Mechanismus
- ...

X-Windows-Sicherheit

- Über Anfragen an den X-Server können bei unzureichendem Schutz alle Aktionen am Bildschirm, der Tastatur und Maus mitgehört und modifiziert werden.
- Der Zugriff kann über xhost eingeschränkt werden (Maschinen-bezogen), besser ist die Verwendung von Magic Cookies (Personen-gebunden). Beide Möglichkeiten haben jedoch Schwächen.

Services für den Angreifer

- finger
- showmount
- rpcinfo
- ...

hosts.equiv und .rhosts

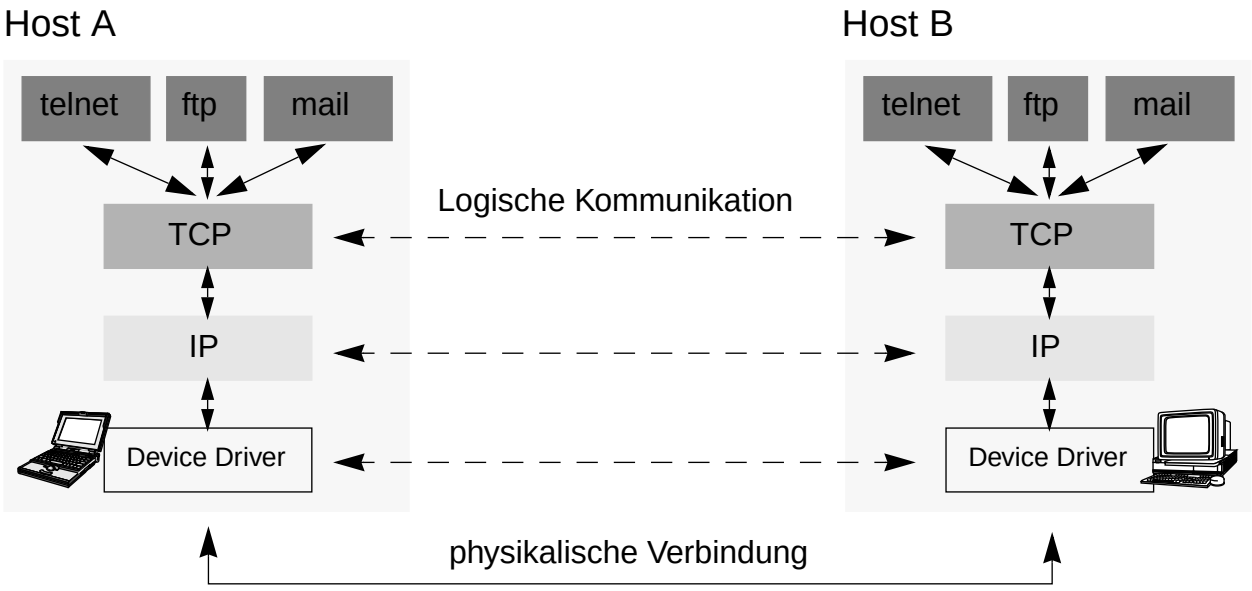
- Über diesen Mechanismus kann das Einloggen auf dem System ohne Paßwort erfolgen.
- Die Verwendung ist zumindest nicht unproblematisch.

3 Gefahren aus dem Netz

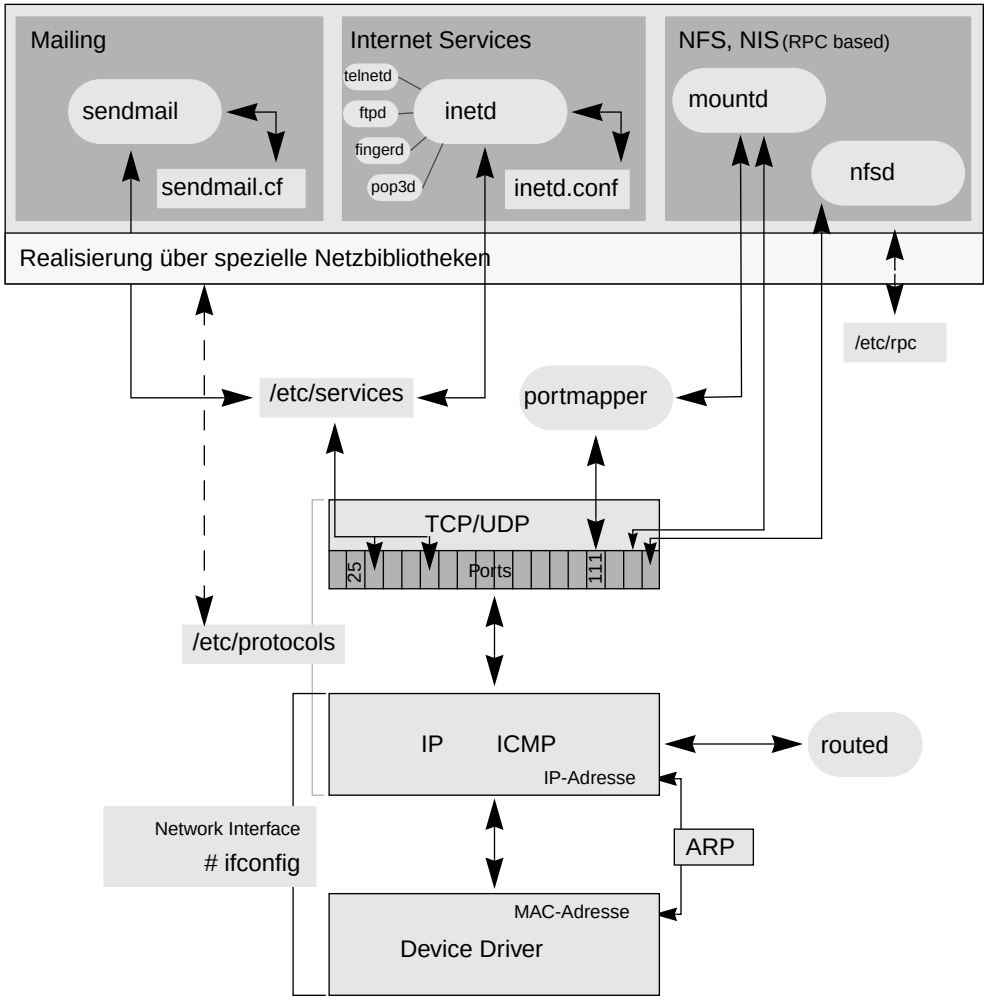
Übersicht

- Aufbau der Unix-Netzfunktionalität
- Gefahren aus dem Netz
  Denial of Service, Sniffer, IP-Spoofing
- RPC-basierende Dienste
- Tools zur Absicherung
  TCP-Wrapper und PD-Portmapper
- Routing und Paketfilterung
- Protokollierung des Netzverkehrs

TCP/IP-Protokoll



Unix-Implementierung der Netzwerkdienste



Wichtige Netz-Konfigurationsdateien

/etc/services - Zuordnung von Ports und Diensten

- Syntax

```
official service name  port number/protocol   alias service name
```

- Beispiel

```
netstat         15/tcp
quote           17/udp          text
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp          mail
name            42/tcp          nameserver
tftp            69/udp
finger          79/tcp
pop             109/tcp
nntp            119/tcp         readnews
ntp             123/udp         ntp
exec            512/tcp
login           513/tcp
syslog          514/udp
talk            517/udp
route           520/udp         router routed
timed           525/udp         timeserver
```

Einfaches Ansprechen von Ports

```
# telnet hrz13 25

Trying 132.187.111.13...
Connected to 132.187.111.13.
Escape character is ‚^]‘.
220 hrz13.uni-wuerzburg.de ESMTP Sendmail 8.8.8/8.8.8;
Tue, 18 Mar 1997 17:51:12)
help
214-This is Sendmail version 8.8.5
214-Topics:
214-    HELO    EHLO    MAIL    RCPT    DATA
214-    RSET    NOOP    QUIT    HELP    VRFY
214-    EXPN    VERB    ETRN    DSN
214-For more info use “HELP <topic>“.
214-To report bugs in the implementation send email to
214-    sendmail-bugs@sendmail.org.
214-For local information send email to Postmaster at your site.
```

Wichtige Netz-Konfigurationsdateien

inetd-Dämon - Verwaltung von Internet-Anwendungen

- Konfiguration in `/etc/inetd.conf`

```
#servive socket protocol [no]wait user serverpath serverargs
#
ftp     stream  tcp     nowait  root    /usr/sbin/ftpd          ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/telnetd       telnetd
shell   stream  tcp     nowait  root    /usr/sbin/rshd          rshd
login   stream  tcp     nowait  root    /usr/sbin/rlogind       rlogind
exec    stream  tcp     nowait  root    /usr/sbin/rexecd        rexecd
finger  stream  tcp     nowait  root    /usr/sbin/fingerd       fingerd
#tftp   dgram   udp     wait    root    /usr/sbin/tftpd         tftpd /tmp
#talk    dgram   udp     wait    root    /usr/sbin/talkd         talkd
time    dgram   udp     wait    root    internal                time
.......
```

Feld	Erklärung
Service-Name	Der Service-Name entspricht dem Eintrag im File / etc/services. Über diesen Namen findet inetd den Port, den er abhören soll.
Socket Type	Legt fest, ob die Kommunikation über *streams* oder *datagrams* ablaufen soll.
Protocol-Name	Hiermit wird das Protokoll des Dienstes festgelegt. TCP arbeitet mit *streams*, wohingegen UDP auf *datagrams* zurückgreift.

Feld	Erklärung
Wait/NoWait	Im Falle Wait wird die gesamte Kommunikation über den definierten Port abgewickelt. Hingegen erfolgt bei NoWait die Erzeugung eines neuen Serverprozesses über einen anderen Port via fork und exec.
UserName	Spezifiziert die UID, unter der der Server-Prozeß läuft.
ServerPath	Dies sind die Argumente, die inetd mit argv[0] zur Ausführung übergeben werden. Manche Zeilen haben hier den Eintrag `internal`, der auf ein internes Programm von inetd verweist.
ServerArgs	

Übersicht der gefährlichen Ports ...

Ports und Services: Empfehlungen diverser Institutionen

Port	TCP	UDP	Name	Erklärung	Cheswick/Bellovin		CERT	
1	✔		tcpmux	TCP port multiplexer (RFC 1078). Anfragen werden über diesen Port direkt an eine Anwendung weitergeleitet. Im Gegensatz zum portmapper arbeitet die Anwendung auf diesem Port, wohingegen der portmapper einen neuen Port zuweist. Entsprechend unterbindet Filtering dieses Ports alle Anwendungen, wohingegen Filtern von Port 111 nur den leichten Zugriff auf die Anwendungen verhindert.	keine Empfehlung, Port wird selten verwendet.			
7	✔	✔	echo	Der echo-Server ist ein Highlevel-Pendant zum einer ICMP-basierenden ping-Anfrage. Über diesen Mechanismus kann der Status einer Maschine (alive) abgefragt werden.	-			
9	✔	✔	discard	Dabei handelt sich um /dev/null des Internets.	Harmlos.			
11	✔		systat	Wird gelegentlich von Programmen netstat, w und ps verwendet.	Sollt unbedingt blockiert werden.	**block**		
13	✔	✔	daytime	Tageszeit in human-readable Form.	Harmlos			
15	✔		netstat	Erklärung analog zu systat (port 13).		**block**		
19	✔	✔	chargen	Charakter Stream Generator.	Sinnlos.			
20	✔		ftp-data	Datenkanal für ftp.	Schwer filterbar.			
21	✔		ftp	FTP Kontrollkanal.	Sollte nur (wenn überhaupt) am ftp-Server offen sein.			
23	✔		telnet	Telnet-Kanal.	Offen nur über ein Login-Gateway.			
25	✔		smtp	Simple Mail Transfer Protokoll. Über diesen Port wird die Mail abgewickelt.	Nur offen für incoming Mialgateways. Diese sollten ohne sendmail laufen.			

Übersicht der gefährlichen Ports ...

Ports und Services: Empfehlungen diverser Institutionen

Port	TCP	UDP	Name	Erklärung	Cheswick/Bellovin		CERT	
37	✔	✔	time	Zeit in Maschinen-lesbarer Zeit. Über ICMP ist die gleiche Information verfügbar.	Kein Grund zu blockieren.			
43	✔		whois	Für (gefährlichen) whois-Server	Blockieren, falls nicht benötigt.	**block**		
53	✔	✔	domain	Über diesen Port werden Nameserver-Informationen weitergegeben. Ein Blockieren schützt vor der Weitergabe dieser Informationen.			block	
67		✔	bootp	Zum Booten von Terminals und diskless Clients.	Sollte blockiert werden, da zu viele Informationen weitergegeben werden.	**block**		
69		✔	tftp	Trivial File Transfer Protocol. Wird meist zum Booten von X-Terminals verwendet. Bei falscher Konfiguration können beliebige Files ohne Paßwort via tftp vom Rechner geholt werden.	Sehr gefährlich.	**block**	block	
70	✔		gopher	Gopher Internet Service.	Nützlich, aber gefährlich.			
79	✔		finger	Finger Service.	Sollte nur zu einem finger Server erlaubt, sonst unterbunden werden.	**block**		
83	✔		http	WWW-Server.	Gefährlich, aber nützlich.			
87	✔		link	Wird selten genutzt, außer von Hackern (eignet sich als Alarm-Port)	Alarm-Port.		block	
88		✔	kerberos	Der offizielle Kerberos Port.	Blockieren, falls ungenutzt, zudem die Ports 749-751.	**block**		
95	✔		supdup	Ein weiterer nur von Hackern genutzter Alarm-Port.				

Übersicht der gefährlichen Ports ...

Ports und Services: Empfehlungen diverser Institutionen

Port	TCP	UDP	Name	Erklärung	Cheswick/Bellovin		CERT	
109	✔		pop-2	Post Office Protocol.	Blockieren, falls Mail nicht von außerhalb gelesen werden soll.	**block**		
110	✔		pop-3	analog	analog.	**block**		
111	✔	✔	sunrpc	Portmapper-Port. Zu beachten ist, daß der Port-Bereich auch ohne den Portmapper gescannt werden kann.	Blockieren!	**block**	block	
113	✔		auth	Falls dieser blockiert wird, sollten keine ICMP-Rejections gesendet werden.	Im allgemeinen sicher.			
119	✔		nntp		Bei Verwendung sollten Source und Destination Filter verwendet werden.			
123		✔	ntp		Sicher, falls ntp-Access Control Listen verwendet werden.			
144	✔		NeWS	Ein Window-System.	Ähnlich X11.	**block**		
161		✔	snmp	Simple Network Management Protocol		**blovk**		
162		✔	snmp-trap		Blockieren, falls nicht Router außerhalb des Netzes aufgezeichnet werden sollen.	**block**		
177		✔	xdmcp	Regelt X11 Login.		**block**		
512	✔		exec	Kann in Zusammenhang mit rcp Verwendung finden. Der Interworm machte Gebrauch davon. Zudem erfolgt keinerlei Logging.		**block**	block	
512		✔	biff		Gefährlich und fehlerhaft.	**block**		

Übersicht der gefährlichen Ports ...

Ports und Services: Empfehlungen diverser Institutionen

Port	TCP	UDP	Name	Erklärung	Cheswick/Bellovin		CERT	
513	✔		login	Port für rlogin, rsh ...	„Shudder“	**block**	block	
513		✔	who	An diesem Port sollten keine legitimierten Anfragen erfolgen.		**block**		
514	✔		shell	Kein Logging.	„Double shudder“	**block**	block	
514		✔	syslog	Besitzt diverse Sicherheitslücken, zudem können eigene Logs attackiert werden.		**block**		
515	✔		printer	Abwicklung für remote-Printer	Da selten Bedarf an remote Druckern sollte dies blockiert werden.	**block**	block	
517		✔	talk	Die aktuelle Implementierung beinhaltet eine Konversation zwischen zufällig ausgewählten Ports.		**block**		
518		✔	ntalk	analog		**block**		
520		✔	route	Erlaubt das Spielen mit Routing Tabellen.		**block**		
540	✔		uucp	Historisch gefährlich und eigentlich überflüssig.		**block**	block	
1025	✔		listener	???		**block**		
2000	✔		openwin	Funktionalität wie X11		**block**	block	
2049		✔	nfs			**block**	block	
2766	✔		listen	Der SystemV Listener, ähnlich zu tcpmux, aber mit mehr Services.		**block**		
600-600X	✔		X11		Der gesamte Bereich sollte blockiert werden.	**block**	block	

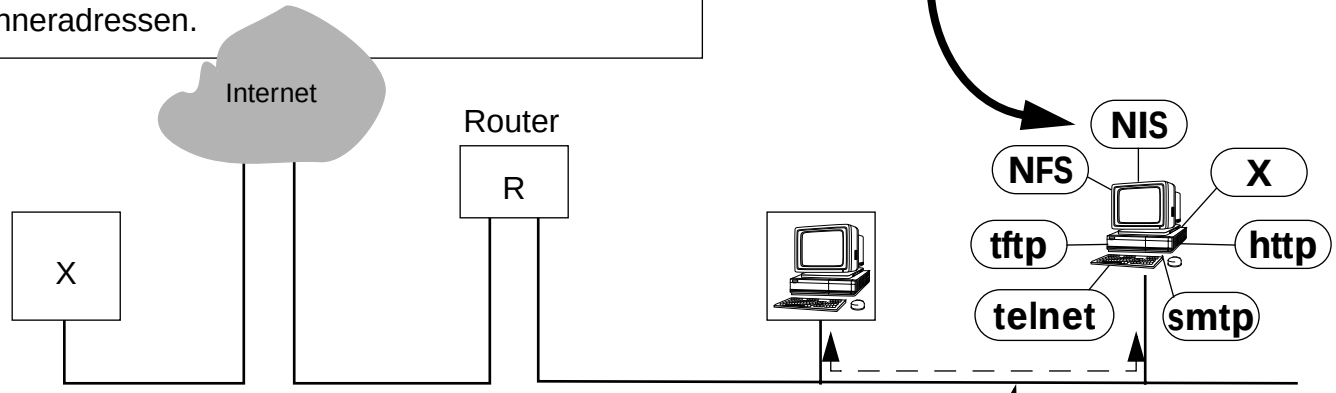
Übersicht: Gefahren aus dem Netz

Mögliche Fälschungen

- Rechneradressen
- Rechnernamen
- Benutzerkennung
- Hauptproblem: Viele Authentisierungen und Zugriffs-kontrollen erfolgen anhand der Rechnernamen und Rechneradressen.

Diverse gefährliche Dienste

- Network File System (NFS)
- Network Information System (NIS)



Abhörgefahr

- Nahezu alle Informationen gehen unverschlüsselt über das Netz und können abgehört werden.
- Paßwörter können einfach im Klartext ermittelt werden.

Denial of Service

Was passiert?

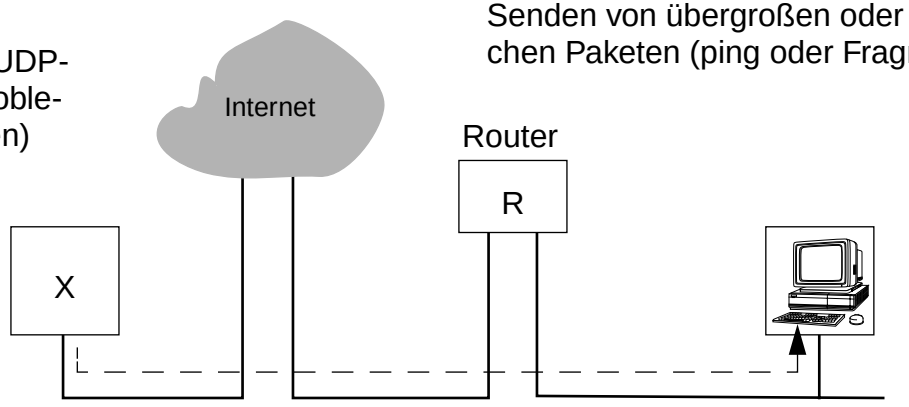
- der Rechner wird in seiner Funktionsfähigkeit vorübergehend starkt beeinträchtigt
- es kann zu einem Absturz des Rechners kommen
- es werden (i. d. R) keine Rechte auf dem betroffenen Rechner erlangt

Typische Mechanismen

- Spam mit SYNs oder UDP-Paketen (Überlastung des Rechners)
- Ausnutzung von Protokollschwächen, z. B. Senden von übergroßen oder ungewöhnlichen Paketen (ping oder Fragmentierung)

Gegenmaßnahmen

- Patches einspielen (z. B. bei ping-Problem)
- Stoppen von externen UDP-Paketen am Router (problematisch an Universitäten)



Sniffer

Was passiert?

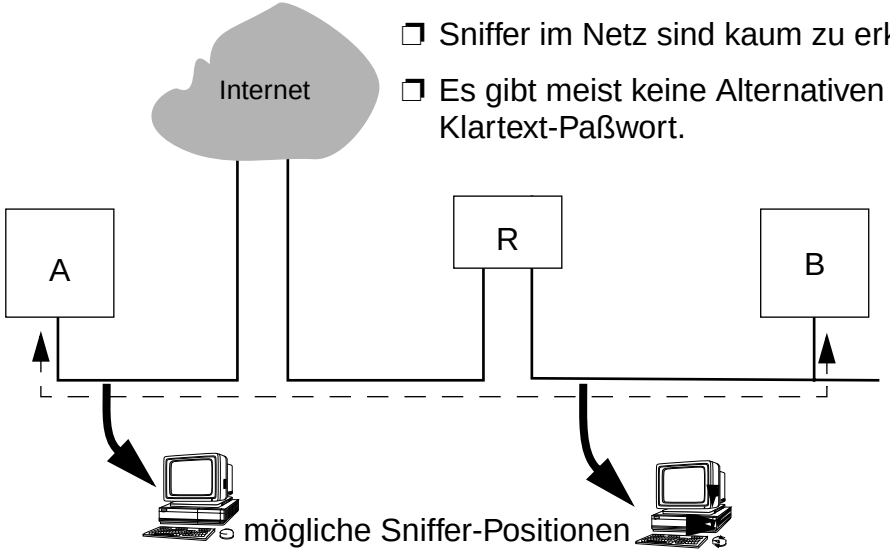
- Die meisten Informationen gehen unverschlüsselt über das Netz und können leicht abgehört werden.
- Insbesondere gilt dies meist auch für Username und Paßwort.
- Das Sniffen eines Paßwort ermöglicht den direkten Zugriff auf einen Rechner.

Probleme

- Sniffer im Netz sind kaum zu erkennen.
- Es gibt meist keine Alternativen für das Klartext-Paßwort.

Gegenmaßnahmen

- Paßwörter nur über „sichere“ Netze verwenden.
- Versuchen, das lokale Netz gegen Sniffer zu sichern (z. B. Trennung von öffentlich und privat).
- Verwendung von .rhost-Mechanismus (bedingt jedoch andere Probleme).
- Einsatz der Secure Shell oder von Einmal-Paßwörtern.



IP-Spoofing

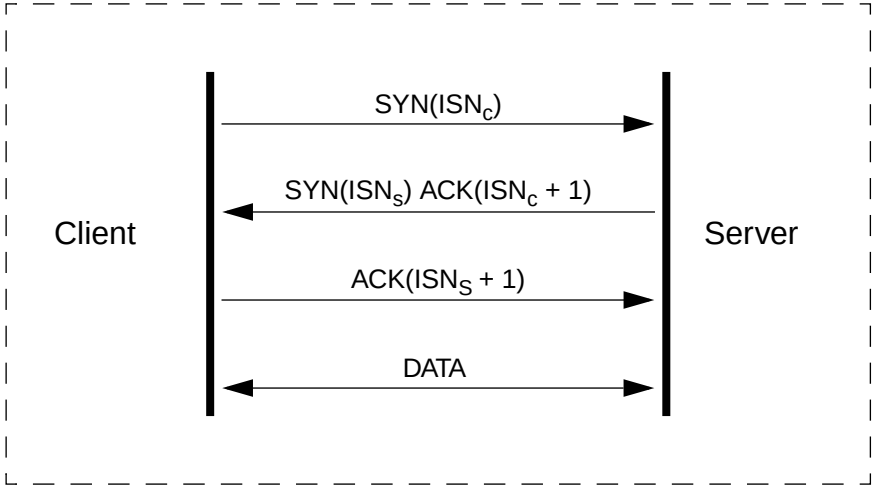
Möglichkeiten und Probleme

- Fälschung von IP-Adressen über Router-Grenzen hinweg
- Ausnutzung von Diensten über eine gefälschte IP-Authentisierung
- IP-Spoofing aus den Internet heraus möglich

Mechanismus

- Es werden Schwächen des TCP-Protokolls ausgenutzt.
- Es muß eine „trusted relation“ zwischen zwei Rechnern bestehen.
- Richtiges IP-Spoofing ist nicht-trivial (s. u.)

Grundlage: three way hand handshake

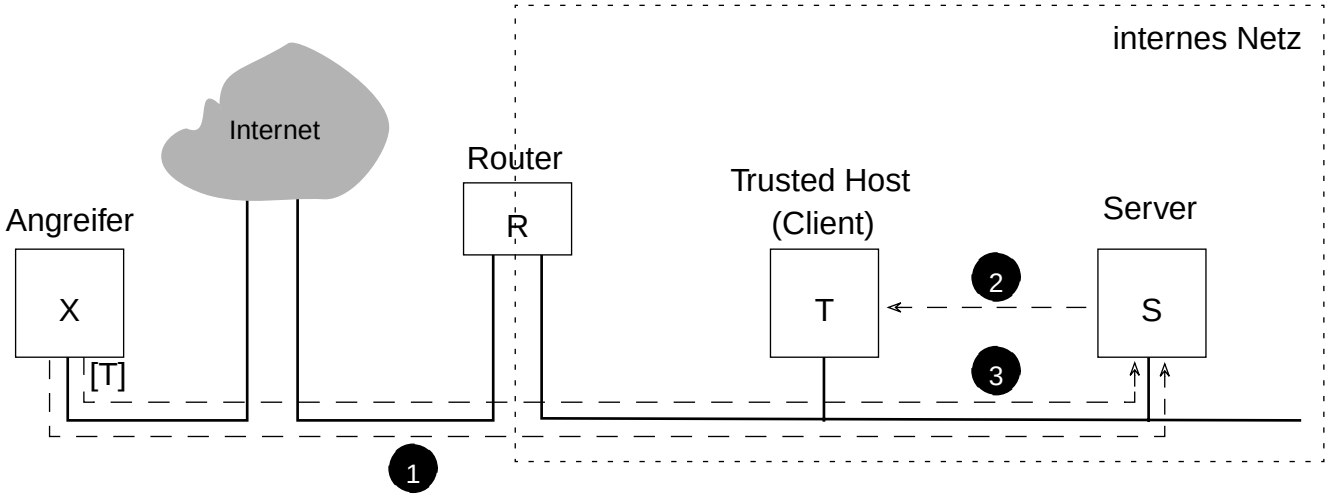


SYN	open request
ACK	acknodlegement
ISN	initial sequence number

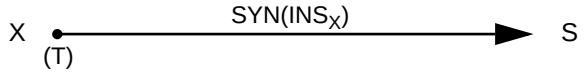
Rolle der ISN

- 32-Bit-Counter
- wird über Zeitmechanismus erhöht (+1 pro 4ms, +1 pro Request)
- Vermeidung von ISN-Überschneidungen
- Problem: ISN in etwa vorhersagbar

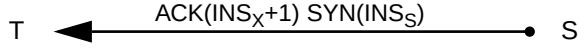
IP-Spoofing-Mechanismus



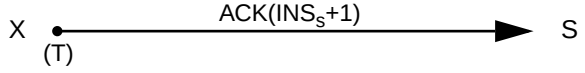
1. X gibt vor T zu sein und schickt SYN(INS_X) an den Server S



2. S schickt ACK von INS_X an T und ein neues SYN(INS_S)



3. X, das vorgibt T zu sein, bestätigt INS_S



Somit können Daten fließen ...

Probleme

- INS_S kommt nie bei X an
  Wie erfolgt Bestätigung?
  ☛ Sequence Number Prediction durch Scannen vor dem Angriff
- T wird versuchen, die Verbindung zurückzusetzen, da kein SYN von T ausging
  ☛ Denial of Service Attacke auf T (da T sonst schneller)
- Um Zugang zu T zu erreichen, muß eine *trusted dependency* zwischen T und S bestehen (z. B. .rhosts auf S für T)

IP-Spoofing

Verwundbare Dienste
(alle IP-basierenden Dienste)

- r-Dienste (.rhosts)
- NFS
- TCP-Wrapper
- X-Window-Server (xhosts)

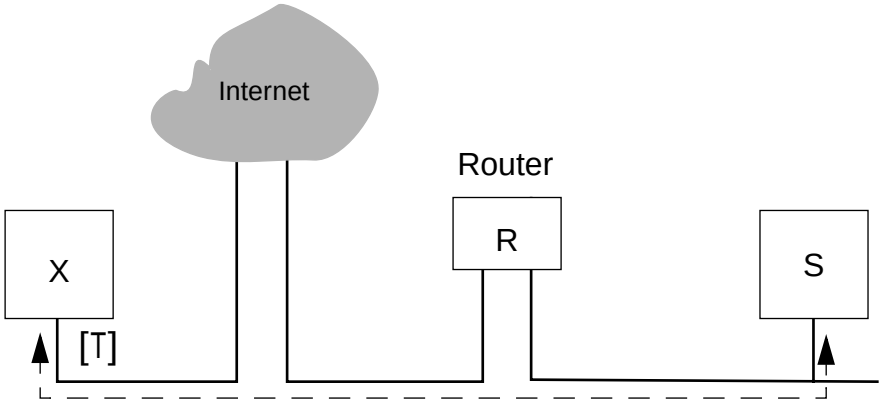
Gegenmaßnahmen

- Auffinden von Versuchen
  Suche nach Paketen in Logfiles (z. B. MoniBox), die von außen kommen und vorgeben, Adressen von innerhalb zu besitzen.
- Routerkonfiguration
  Kann bewirken, daß Pakete, die von außen kommen und IP-Adressen vorgeben, die nur innerhalb des Netzes sein dürfen, zurückgewiesen werden.

IP-Source-Routing

Problemdefinition

- Laut RFC 1122 kommen Pakete eines angesprochenen Zielrechners auf dem inversen Weg zurück.
- Einem IP-Paket kann ein Routing-Weg mitgegeben werden (sogenanntes *loose source routing*).
- Ein Angreifer kann diesen Mechanismus mißbrauchen.

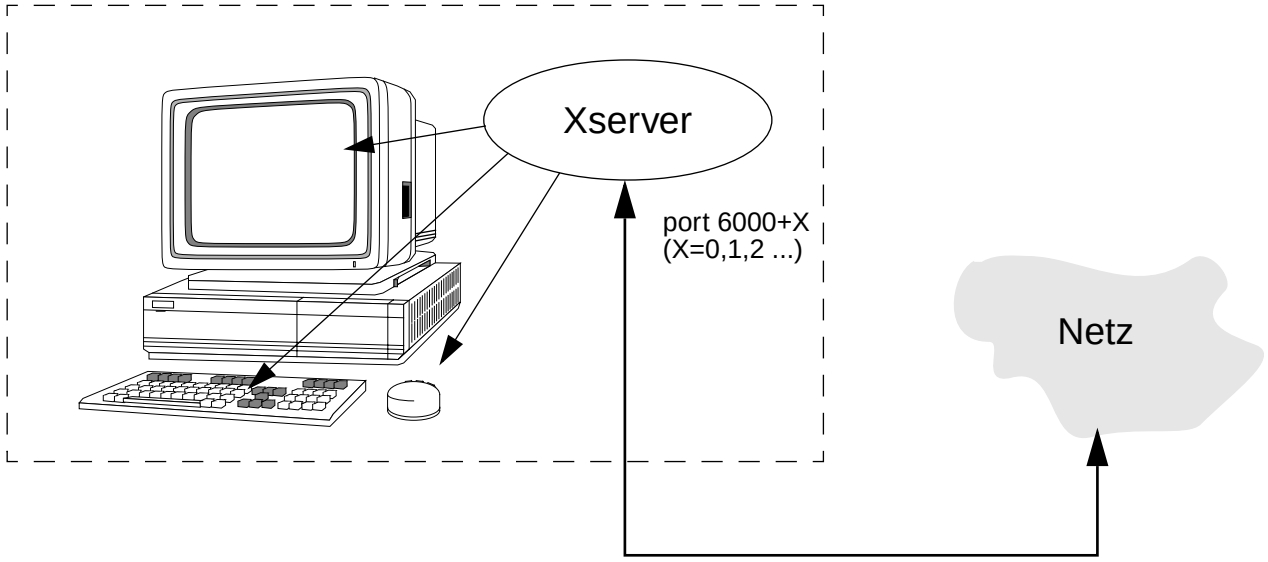


1. X gibt vor T zu sein, und definiert eine Route von X nach S.
2. Die Pakete von S gehen an X zurück (inverser Weg).
3. Der Mechanismus kann ausgenutzt werden, wenn S dem Rechner T vertraut (d. h. ein .rhosts-File auf S für T).

Konsequenz

- *loose source routing* sollte am Router unterbunden werden

Das X-Window-System



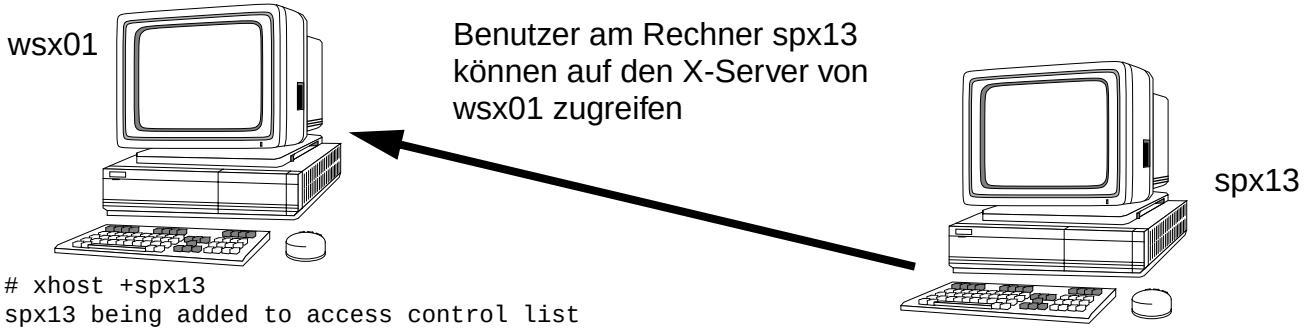
Möglichkeiten und Probleme

- Der X-Server kontrolliert Bildschirm, Maus und Tastatur.
- Der Xserver kann über das Netz angesprochen werden.
- Ist der Xserver nicht „abgesichert“, können alle Vorgänge auf dem Bildschirm oder Tastatureingaben mitprotokolliert werden. Analog sind Eingaben von extern möglich.

Absicherung des X-Servers

Rechnerbezogen - xhost-Mechanismus

- Syntax
  # xhost [+|-] [hostname]
- Vorteil: einfach
- Nachteil: nicht sehr sicher, da nur rechnerbezogen



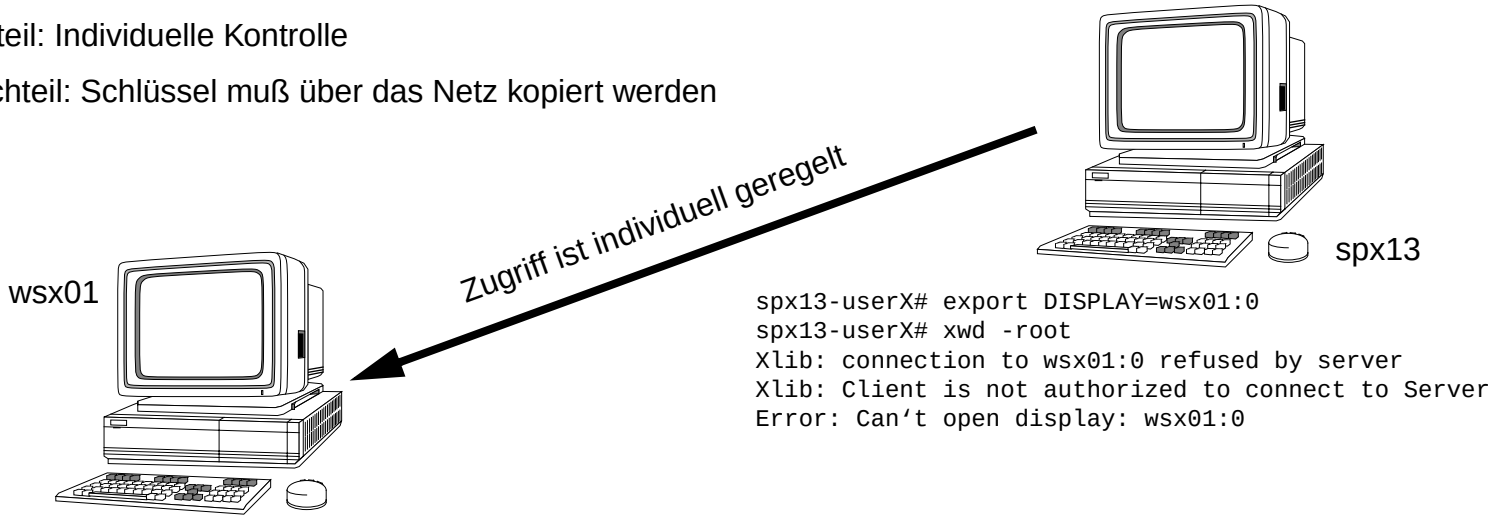

```
# xhost +spx13
spx13 being added to access control list
# rlogin spx13
...
spx13# export DISPLAY=wsx01:0
spx13# xterm
```

```
spx13-userX# export DISPLAY=wsx01:0
spx13-userX# xwd -root
```

Absicherung des X-Servers

Individuell - xauth magic cookies

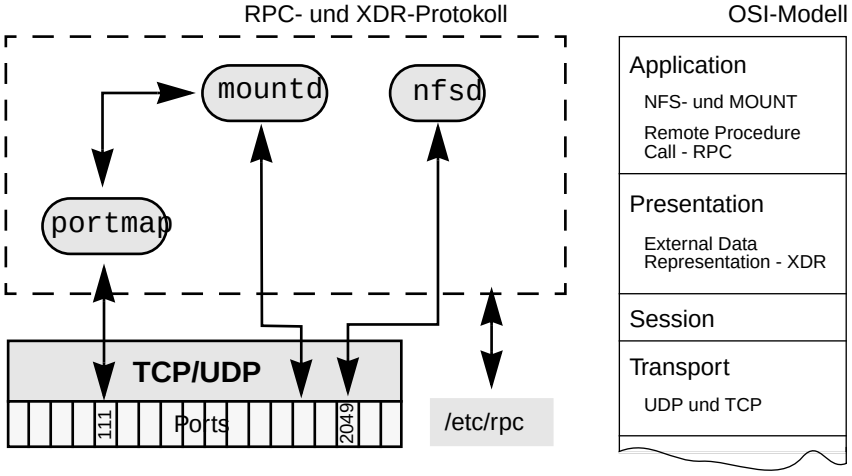
- xdm erzeugt beim Start einen Schlüssel im File $HOME/.Xauthority
- Zugriffe auf den X-Server können mit diesem Schlüssel geregelt werden.
- Vorteil: Individuelle Kontrolle
- Nachteil: Schlüssel muß über das Netz kopiert werden




```
spx13-userX# export DISPLAY=wsx01:0
spx13-userX# xwd -root
Xlib: connection to wsx01:0 refused by server
Xlib: Client is not authorized to connect to Server
Error: Can‘t open display: wsx01:0
```

```
# xauth list
wsx01:0  MIT-MAGIC-COOKIE-1 3bbef30e160...
# rlogin spx13
...
spx13# xauth add wsx01:0  MIT-MAGIC-COOKIE-1 3bbef30e160..
spx13# xterm -display wsx01:0
```

RPC-basierende Dienste und portmapper



portmapper Funktionen

1. **SET**: Hiermit kann ein Dienst registriert werden.
2. **UNSET**: Der gewünschte Dienst wird aus der Tabelle entfernt.
3. **GETPORT**: Die Portnummer eines gewünschten Dienst kann so abgefragt werden.
4. **DUMP**: Dies erlaubt den Zugriff auf den gesamten Inhalt der Portmapper-Tabelle.

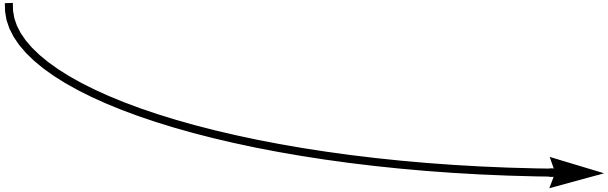
Abfragen der portmapper-Tabelle (mittels DUMP bzw. rcpinfo)

```
# rpcinfo -p [targethostname]
program vers proto port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp   1029  mountd
    100005    3   udp   1029  mountd
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100005    1   tcp    977  mountd
    100005    3   tcp    977  mountd
    100004    2   udp    974  ypserv
    100004    2   tcp    975  ypserv
    100007    2   tcp   2978  ypbind
    100007    2   udp   4718  ypbind
    100009    1   udp   1023  yppasswdd
    100069    1   udp    789  ypxfrd
    100069    1   tcp    791  ypxfrd
```

Sicherheitsprobleme des portmappers

Probleme

- Mittels rpcinfo können die RPC-Portnummern (weltweit) abgefragt werden. Das Risiko ist offensichtlich.
- Mit UNSET kann versucht werden, Dienste aus der Registrierungstabelle zu entfernen. Dies entspricht einem Denial of Service Angriff.
- Die meisten Portmapper ermöglichen (um Overhead zu reduzieren) die lokale Weiterleitung einer Anfrage. Diese werden dann wie lokale Anfragen behandelt. Sicherheitsmechanismen können so beliebig unterlaufen werden.

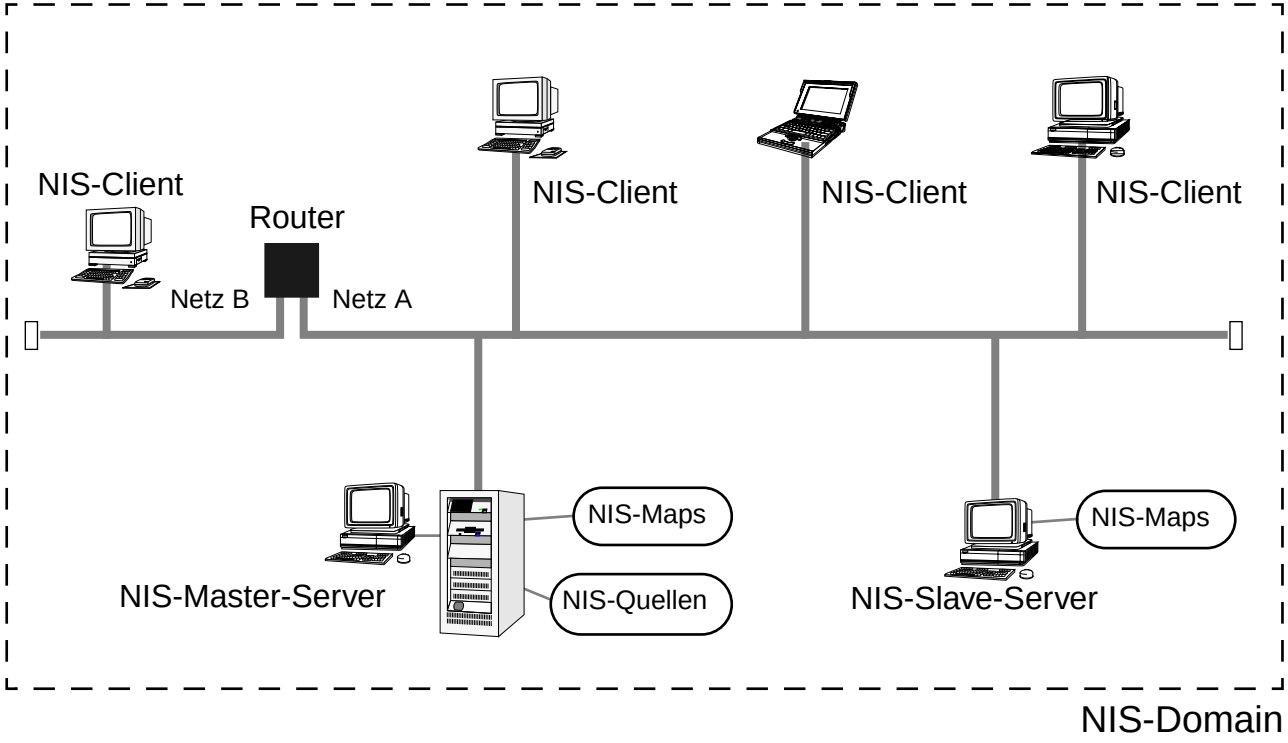


Abhilfe

- Verzicht auf portmapper (NFS und NIS) kaum möglich
- Einsatz des PD-portmappers von Wietse Venema (s. u.)

Arbeitsweise von NIS

Funktionalität: Zentrale Verwaltung eines Clusters



NIS-Elemente

Daemon	Funktionalität
portmap	Dieser Dämon verwaltet die Registrierungstabelle aller RPC-Dienste. Er muß als erster vor allen anderen RPC-basierenden Services gestartet werden.
ypserv	Hiermit werden alle Client-Anfragen behandelt.
ypbind	Dieser Dämon bindet den Client an einen Server.
ypxfrd	Dieser Dämon des Master-Servers beschleunigt den Transfer der Maps vom Master-Server zum Slave-Server. NIS hat auch ohne `ypxfrd` die volle Funktionalität.
yppasswdd	Mit Hilfe dieses Dämons am Master-Server ist es den Benutzern der Clients möglich, ihr Paßwort zu modifizieren.

Sicherheitsprobleme von NIS

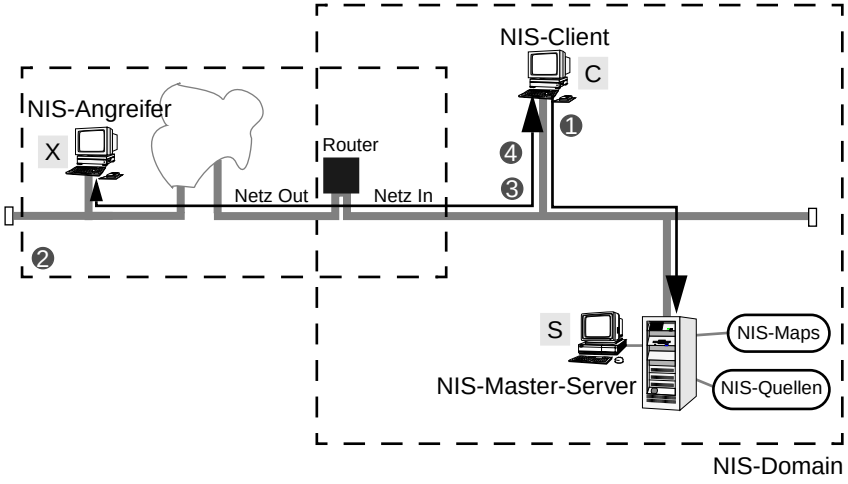
NIS = konzeptionell unsicher

- ❒ NIS arbeitet auf dynamischen Ports und ist deshalb schwer zu filtern.
- ❒ Die Zugriffskontrolle erfolgt nur über den Domainnamen.
- ❒ Andere Rechner können sich in den NIS-Verbund einhängen, Abfragen ausführen und Anworten des Servers vortäuschen (Einschränkungen bei einigen Unix-Derivaten in der Datei securenets möglich).
- ❒ Bei NIS-Anfragen geht das Paßwort über das Netz.
- ❒ Der ypbind-Dämon sollte auf den Master-Server gerichtet sein.
- ❒ Hosteinschränkungen können über den PD-Portmapper vorgenommen werden.

! Besondere Gefahr: domainname !

- ❒ Die Kenntnis des NIS-domainname erlaubt die Zugriffe auf alle NIS-Daten, d. h. insbesondere das Paßwortfile.
- ❒ Der NIS-domainname sollte deshalb keinesfalls mit dem Netzdomainnamen oder ähnlichen Kombinationen identisch sein!

Sicherheitsprobleme von NIS

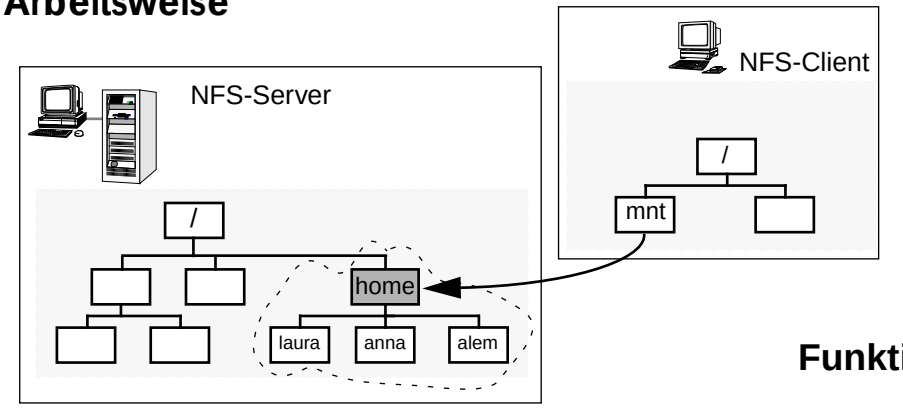


ypbind-Optionen

Option	AIX	Digital UNIX	HP-UX	IRIX	SunOS/Solaris	Erklärung
-s	✔	✔	✔	✔		ypbind läuft im secure mode, d. h. ypbind nimmt nur Informationen von einem Server, der einen privilegierten Port (< 1024) verwendet.
-S [server] [server]		✔		✔		Hierdurch wird eine feste Anbindung an einen NIS-Server definiert.
-ypset	✔	✔	✔	✔	✔	Läuft ypbind mit dieser Option, so kann jeder Rechner der Domain die Anbindung an einen anderen Server via ypset veranlassen. (Sicherheitsrisiko!!!)
-ypsetme	✔	✔	✔	✔	✔	Mit dieser Option kann das Binding an einen Server nur lokal modifiziert werden.

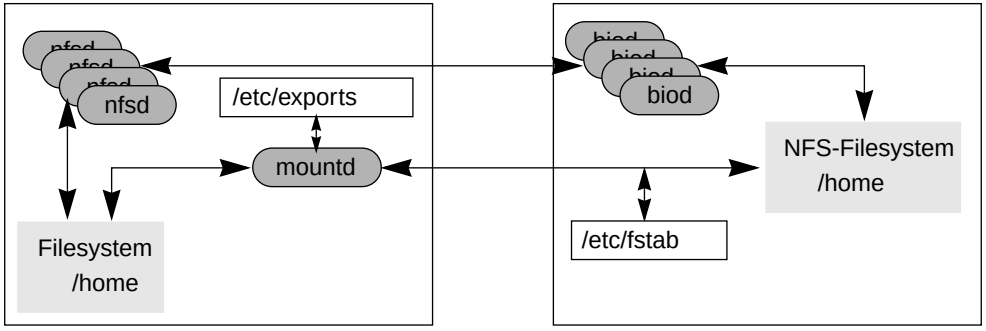
Arbeitsweise von NFS

Arbeitsweise



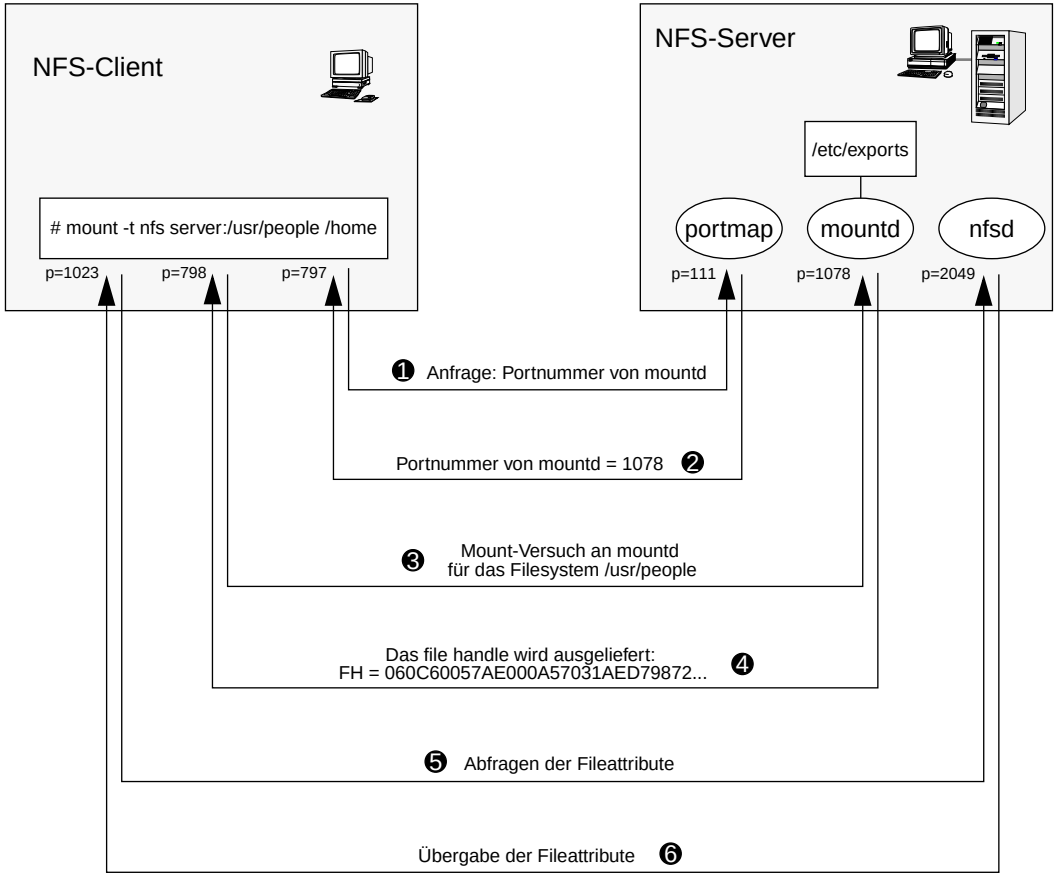
Funktionalität: Zentrale Datenhaltung

Beteiligte Dämonen



Vorgänge beim Mounten

```
client# mount -t nfs servername:/usr/people /home
```



Zentraler Schlüssel

= Filehandle

- die Kenntnis des Filehandles gestattet den direkten Zugriff auf die NFS-Daten (die /etc/exports-Einträge sind wirkungslos)
- Das Filehandle kann teilweise sehr leicht erraten werden (mit fsirand können „bessere“ Handles generiert werden)

Schutz von NFS

NFS = konzeptionell unsicher!

Vorsichtsmaßnahmen

- bessere Filehandles mit fsirand generieren
- nur wenige Filesysteme exportieren (wenn möglich read only)
- am Server sollte kein root-Zugriff für den Client erlaubt werden (mapping auf nobody)
- als Client sollten (soweit möglich) die Filesysteme mit -o nosuid gemountet werden, was das Ausführen von SUID-root-Programmen verhindert
- Einschränkung des Portmapper-Zugriffs auf wenige Rechner (der mountd kann jedoch noch direkt angesprochen werden, falls der Port erraten wird)

Zentrale Schutzmaßnahme

- am Router die Ports 111 (portmapper) und 2049 (nfsd) sperren

Informationen „für“ das Netz

Problem

- diverse Dienste liefern Informationen ins Netz, die für einen Angriff verwendet werden können
- diese Informationen sind intern häufig wichtig, sollten jedoch nicht nach außen gelangen (z. B. finger)

Dienst/Abfrage	Funktion	Kontrolle
finger @host	Zeigt die eingeloggten Benutzer an (insbesondere deren Usernamen)	in /etc/inetd.conf. Über den TCP-Wrapper können diese Informationen nur an Rechner oder spezielle Domains weitergegeben werden.
showmount -e host	Gibt Information über die exportierten Filesysteme.	Hier wird der moutd-Dämon angesprochen.
rpcinfo -p host	Liefert die Liste der angebotenen RPC-Dienste	Hier wird der Portmapper angesprochen. Eine Einschränkung der Informationen kann bei Verwendung des PD-Portmapper vorgenommen werden.

TCP-Wrapper und Portmapper

Problem

- Die standard-Unix Konfiguration des inetd-Dämon erlaubt keine Einschränkung der Dienste auf spezielle Bereiche (z. B. für finger, telnet ...)
- Analog besteht dieses Problem für die RPC-basierenden Dienste, insbesondere den Portmapper.

TCP-Wrapper

- Dieser wird dem inetd-Dämon „vorgeschaltet“ (in /etc/inetd.conf).
- Der TCP-Wrapper bietet eine erweiterte Logging-Funktionalität.
- Mit Hilfe der Files
  `/etc/hosts.allow`
  `/etc/hosts.deny`
  können Services auf einzelne Rechner oder Domänen eingeschränkt werden.

PD-Portmapper

- Dieser beseitigt die typischen Schwächen der konventionellen Portmapper, insbesondere die Weiterleitung von Anfragen.
- Mit Hilfe der Files
  `/etc/hosts.allow`
  `/etc/hosts.deny`
  kann der Zugriff auf einzelne Rechner oder Domänen eingeschränkt werden.

Packetfiltering

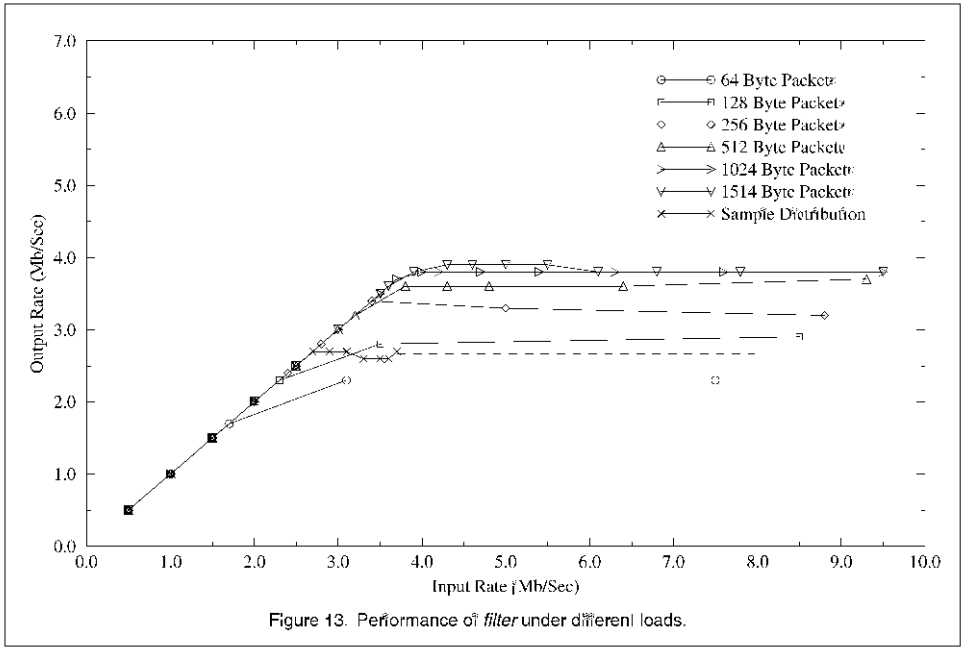


Figure 13. Performance of *filter* under different loads.

Hauptprobleme

- Aufstellen geeigneter Filterregeln
- Überprüfung der Regeln
- Einbruch der Performance

Möglichkeiten

- Verhinderung von IP-Spoofing
- Blockieren der gefährlichsten Ports

→ **Firewalltechnologie?**
siehe z. B. http://www.cert.dfn.de/fwl/

Protokollierung des Netzverkehrs

Was ist möglich?

- Mithören im Netz ist im *promiscuous mode* der Netzwerkkarte leicht möglich.
- Alle Daten mitzuprotokollieren ist kaum möglich.
- Wie können Daten geeignet ausgewertet werden?

snoop und tcpdump

- Alle Pakete werden protokolliert. Quelle und Ziel können definiert werden.
- Die beiden Tools können sehr gut eingesetzt werden, um Netzproblemen (z. B. bei NFS) nachzugehen.
- Zur Protokollierungen aller Verbindungen in ein Subnetz sind diese jedoch nicht brauchbar, da zu viele Daten anfallen.

Argus

- Online Datenreduktion und Auswertung.
- Geeignet für einen Ethernetstrang.
- s. http://www.cert.dfn.de/infoserv/dib/dib-9602.html

Erlanger MoniBox

- Verbindungen werden in Kontexte eingeordnet, was die Datenmenge deutlich reduziert.
- Protokolliert werden Quelle, Ziel, Port und Zeiten.
- Die MoniBox eignet sich sehr gut für den Verkehr in ein Subnetz, da bei Vorfällen, aus den Logfile viele Erkenntnisse gewonnen werden können. Auch können die Logfiles präventiv ausgewertet werden (z. B. Portscans etc.).

4 Logdateien

Übersicht

- syslog
  Funktionalität und Konfiguration
- Auswerte-Tools
  grep, Swatch und Logsurfer

Analyse von Logdateien

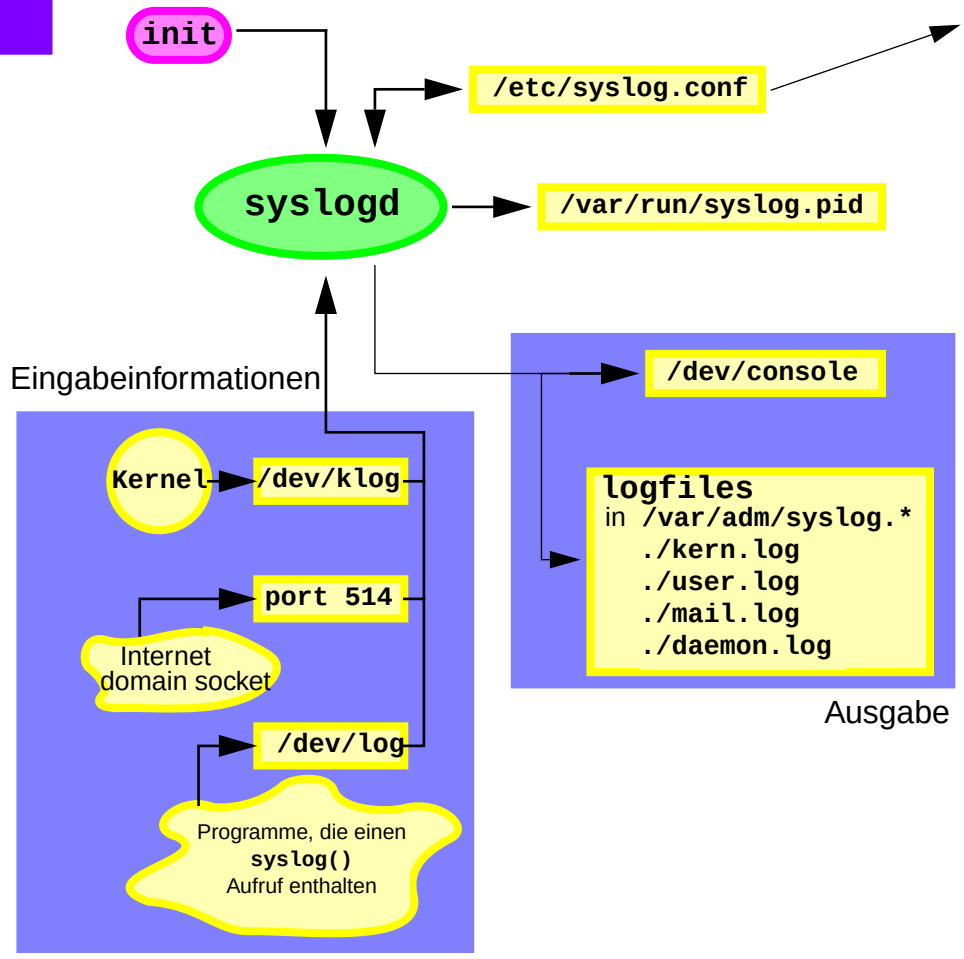
Diverse Logdateien

- Festhalten von Login/Logout der User (Dateien: lastlog, utmp, wtmp). Die Auswertung kann mit dem Kommando last erfolgen.
- Die Aktivitäten der Benutzer können mit lastcomm aus der Datei acct gelesen werden. Hierzu muß das Accounting mit `accton` aktiviert sein. Besser ist das Aufsetzen des Auditings unter der C2-Security.
- Das Wechseln auf root-Privilegien kann im File sulog analysiert werden.
- Über den syslog-Dämon können diverse Systemvorgänge protokolliert werden.

Problem: Geeignete Auswerung der Log-Informationen

- Von Hand: Mühsam und zeitaufwendig.
- Wonach soll überhaupt gesucht werden?
- PD-Tools Swatch und Logsurfer können die Funktionen übernehmen

syslog-Mechanismus



Aufbau des Konfigurationsfiles

- Allgemeine Syntax des Konfigurationsfiles:

  facility.level <tabulator> action

- Beispiel:

  mail.info <tabulator> /var/log/maillog

syslog-Konfiguration

facility	Erklärung
kern	Messages generated by the kernel. These cannot be generated by any user processes.
user	Messages generated by user processes. This is the default facility when none is specified.
mail	Messages generated by the mail system.
daemon	Messages generated by system daemons.
auth	Messages generated by the authorization system: login, su, and so on.
lpr	Messages generated by the line printer spooling system.
local0 ... local7	Reserved for local use.

priority	Erklärung
LOG_EMERG	A panic condition was reported to all users.
LOG_ALERT	Specifies a condition to be corrected immediately; for example, a corrupted database.
LOG_CRIT	Specifies a critical conditions; for example, hard device errors.
LOG_ERR	Specifies errors.
LOG_WARNING	Specifies warning messages.
LOG_NOTICE	Specifies that it is not an error condition, but a condition requiring special handling.
LOG_INFO	Specifies general information messages.
LOG_DEBUG	Specifies messages containing information useful in debugging a program.

Syntax	Erklärung
/filename	Der Slash (/) zeigt an, daß die von syslogd erzeugten Daten in das angegebene File geschrieben werden.
@hostname @ip-adr	Das Zeichen @ bewirkt, daß der syslog-Dämon die Informationen an die angegebenen Rechner weiterleitet. Es kann sowohl der Hostname als auch die IP-Adresse verwendet werden.
user1,user2	Die Mitteilungen erreichen die angegebenen Benutzer, sofern diese eingeloggt sind. Mehrere User können durch Kommas getrennt angegeben werden.
*	Mit dieser Wildcard werden die Meldungen an alle eingeloggten Benutzer weitergegeben.

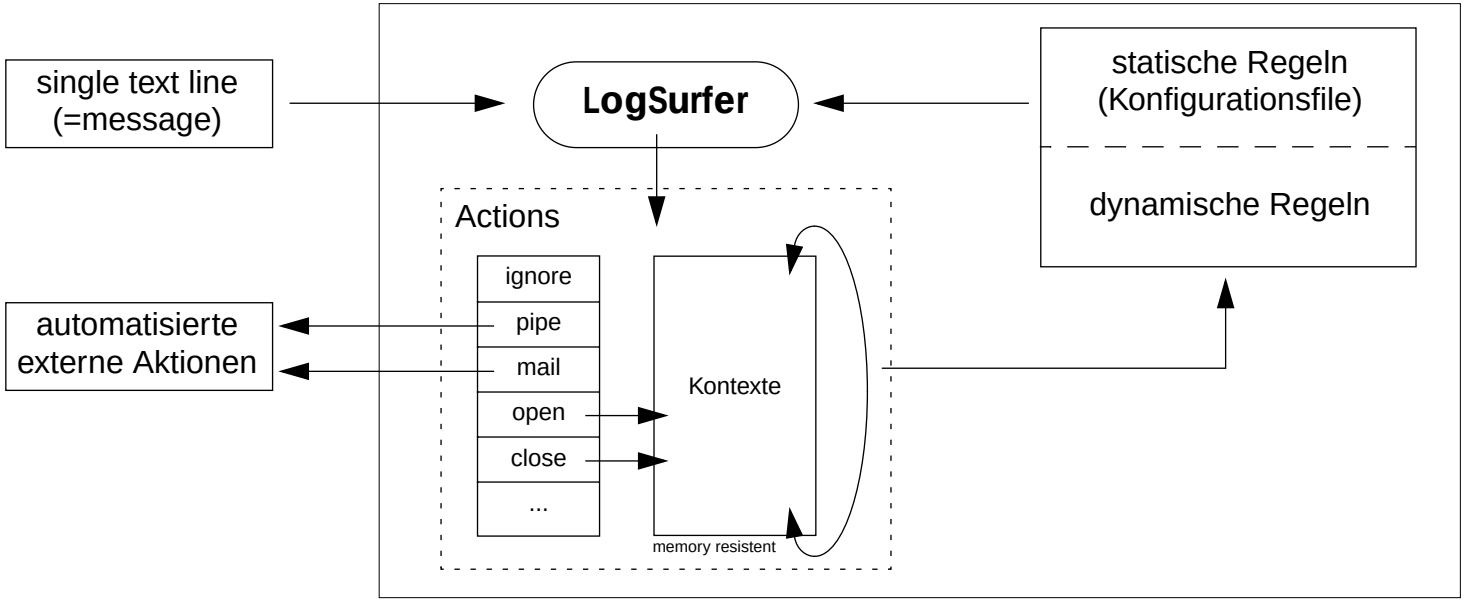
Zentrale Logdateien

- Vorteile
  zentrale Datenhaltung und Auswertung
- Nachteile
  alle Informationen gehen über das Netz
  der Logrechner sollte gesichert werden

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 Gegebnsatz zu Swatch (oder grep) arbeitet Logsurfer kontextabhängig.



RZUW

5 Analyse der Rechnerkonfiguration

Übersicht

- Was sollte überprüft werden?
- Tools für den lokalen Check
  Crack, Cops und Tiger
- Tools für den Test über das Netz
  ISS und Satan

Was sollte analysiert werden?

Probleme

- Rechnerkonfiguration ist komplex.
- Es können Fehler unterlaufen.
- Wie sieht eine „korrekte“ Konfiguration aus?
- Wie können Fehler gefunden werden?
- Wie kann ein Einbruch erkannt werden?

Korrekte Konfiguration?

- Herstellerhinweise beachten.
- Möglichst sauber installieren.
- Möglichst neue Systemversion nutzen.

Tools

- Status des Systems festhalten
  tripwire
- Lokale Analyse
  Crack, Tiger, Cops ...
- Analyse über das Netz
  ISS und SATAN
- Auswertung der Logdateien
  grep, Swatch und LogSurfer

Unix-Checklist

AUSCERT Computer Security Checklist

ftp://ftp.dfn.cert.de/pub/csir/auscert/papers/unix_security_checklist

1. Patches
   - Installed latest patches?
2. Network security
   - Filtering
   - „r“ commands
   - /etc/hosts.equiv
   - /etc/netgroup
   - $HOME/.rhosts
   - NFS
   - /etc/hosts.lpd
   - Secure terminals
   - Network services
   - Trivial ftp (tftp)
   - /etc/services
   - tcp_wrapper (also known as log_tcp)
   - /etc/aliases
   - Sendmail
   - majordomo
   - fingerd
   - UUCP
   - REXD
   - World Wide Web (WWW) - httpd
3. ftpd and anonymous ftp
   - Versions
   - Configuration
   - Anonymous ftp only
   - Configuration of your ftp server
   - Permissions
   - Writable directories
   - Disk mounting
4. Password and account security
   - Policy
   - Proactive Checking
   - NIS, NIS+ and /etc/passwd entries
   - Password shadowing
   - Administration
   - Special accounts
   - Root account
   - .netrc files
   - GCOS field
5. File system security
   - General
   - Startup and shutdown scripts
   - /usr/lib/expreserve
   - External file systems/devices
   - File Permissions
   - Files run by root
   - Bin ownership
   - Tiger/COPS
   - Tripwire
6. Vendor operating system specific security
   - SunOS 4.1.x
   - - Patches
   - - IP forwarding and source routing
   - - Framebuffers /dev/fb
   - - /usr/kvm/sys/*
   - - /usr/kvm/crash
   - - /dev/nit (Network Interface Tap)
   - - Loadable drivers option
   - Solaris 2.x
   - - Patches
   - - IP forwarding and source routing
   - - Framebuffers /dev/fbs
   - IRIX
   - - Patches
   - AIX
   - - Patches
   - HPUX
   - - Patches
   - OSF
   - - Patches
   - ULTRIX
   - - Patches
7. Security and the X Window System
   - Problems with xdm
   - X security - General

Das Tiger-Programm-Paket (lokaler Check)

Entwicklung

- An der Universität von Texas, nachdem dort eine hohe Anzahl von Einbrüchen stattgefunden hatte.

Eigenschaften

- Shellskript zur Sicherheitsanalyse eines Rechners.
- Ist leicht zu konfigurieren und läuft auf sehr vielen Plattformen.
- Es gibt keine Dokumentation, dafür aber zu jedem gefundenen Problem eine ausführlichere Erklärung und einen Vorschlag für das weitere Verhalten.

Liste der Tiger-Checks

- Login Accounts, d.h. Paßwort-File
- Anonymous ftp-Account
- NFS-exportierte Filesysteme
- Konfiguration von inetd
- .rhosts-Files
- .netrc-Files
- cron Einträge
- Paßwort- und Gruppenfiles
- das printcap-File
- Festlegung von Pfaden
- Embedded PATHs
- File Permissions
- Filesystem-Scan
- e-mail Aliases
- Bekannte Einbruchssignale

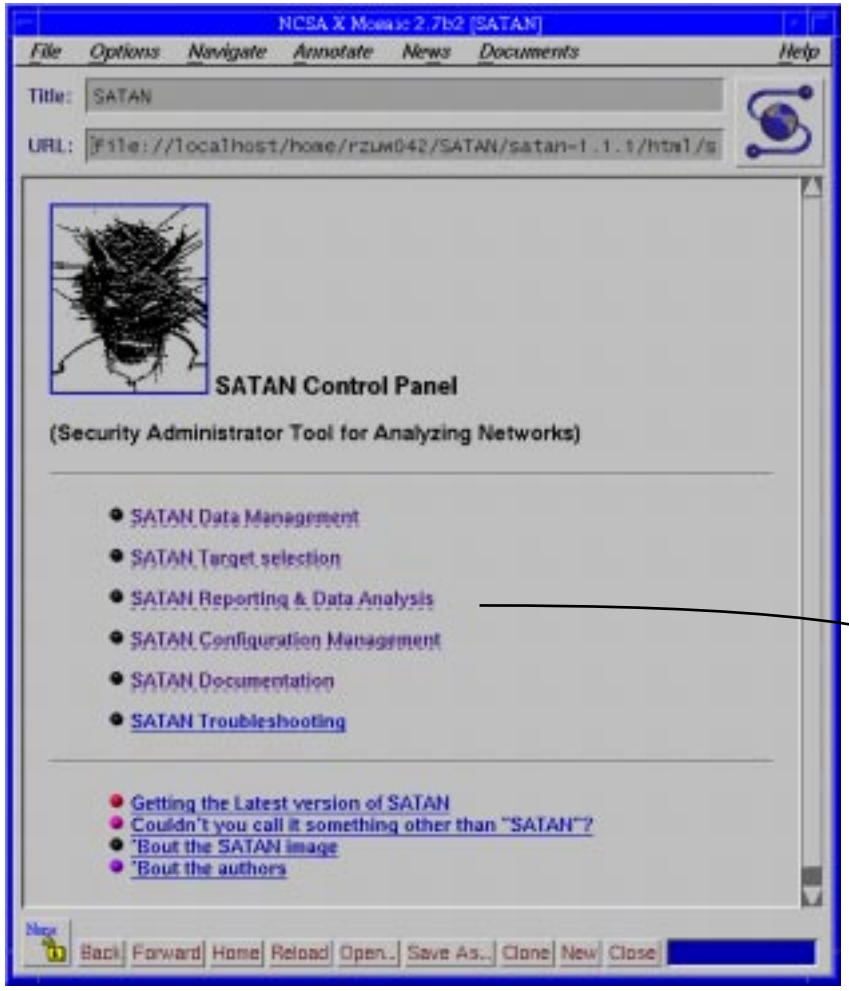
RZUW

SATAN und SAINT - Check über das Netz

Was wird überprüft? http://www.wwdsi.com/saint/

- TFTP
  Nach Möglichkeit nicht benutzen. Falls doch notwendig, entsprechend absichern (Pfad in inetd.conf oder besser noch via TCP-Wrapper incl. chroot()).
- rsh/rlogin
  Wildcards nicht zulassen und Zugriff auf definierte Rechner einschränken (TCP-Wrapper)
- X-Server-Zugriff
  X-Server so konfigurieren, daß eine Authentisierung vorgenommen wird (xauth)
- Portmapper
  Einsatz eines sicheren Portmappers/Rpcbind Programmes.
- NFS - Network File System
  Platten nur an bestimmte Rechner, wenn möglich read-only exportieren. Nur Portnummer unterhalb 1024 für NFS-Anfragen zulassen.
- NIS - Network Information System
  Zugriff auf die NIS-Informationen (wenn möglich) am Server begrenzen.
- Sendmail
  Einsatz des neuesten Sendmails (8.8.5)
- rexd
  Abschalten des rexd.
- (anonymous)-FTP-Konfiguration
  Keine Dateien sollten dem Benutzer ftp gehören oder anderweitig schreibbar sein. chroot() sicherstellen sowie „dummy“ Paßwortfile.

SATAN in Action



Vorteile von SATAN

- ❒ leicht zu installieren

```
./reconfig
make
./satan
```

und zu bedienen

- ❒ sehr gute Dokumentation

6 Informationsdienste

Übersicht

- WWW
  Server- und Client-Sicherheit
- Mailing
  sendmail-Problematik, Spam-Relaying
- Diverse Kommunikationsserver
  ftp-Server, Nameserver (BIND), Timeserver (NTP)

World Wide Web

- http-Verbindungen stellen einen großen Anteil des Internetverkehrs dar.
- Vieles ist automatisiert und einfach möglich.

Probleme

- Jeder WWW-Server stellt ein potentielles Risiko dar (überlegen, ob Server notwendig).
- Eventuell ist über den httpd-Dämon der Zugriff auf alle Server-Dateien (z. B. /etc/passwd) möglich. Wenn möglich sollte ein dedizierter Rechner verwendet werden.
- Client-Daten sind durch Java(script)-Programme o. ä. gefährdet.

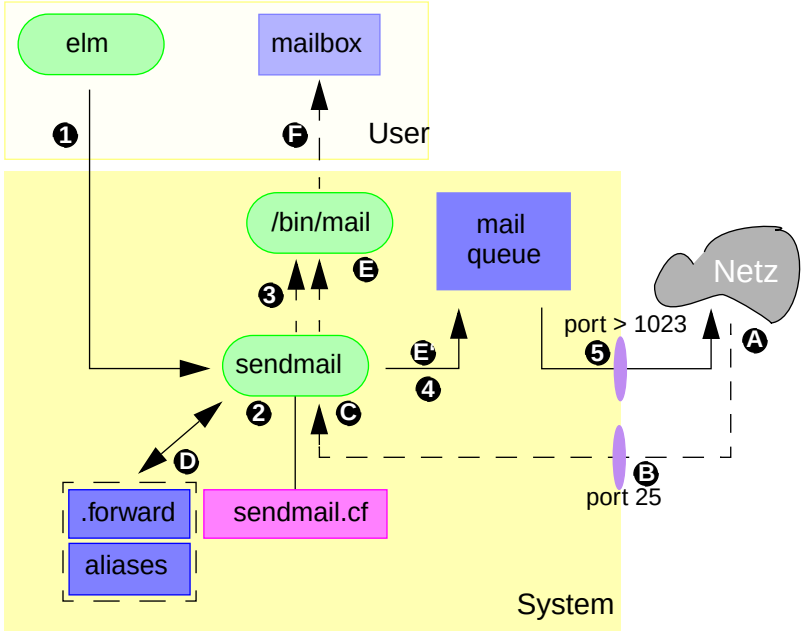
Konfigurationshinweise

- Möglichst aktuelle Server-Version einsetzen und sauber konfigurieren (ohne zu viele Features).
- Maximal den httpd-Server-Dämon unter UID=root laufen lassen. Die childs nicht als root. Falls möglich, kann eine chroot()-Umgebung Sinn machen.
- CGI-bins sollten auf ein Minumum reduziert werden (eventuell mit CGI- Wrapper).

Mailing

Grundlegende Probleme mit sendmail

- Sehr komplex und deshalb Ausgangspunkt sehr vieler Sicherheitsprobleme (Dämon läuft zudem unter UID=root).
- Sowohl problematisch nach außen, als auch lokal (root-Access).
- Stets versuchen, die neueste Version (z. Zt. 8.9) verwenden.
- Nach schlechten Paßwörtern liefert sendmail sicher die zweitwichtigste Sicherheitslücke für Angreifer (zumindest ältere Versionen davon)



Weitere Probleme

- Ausspähen von Daten über VRFY und EXPN
  ☛kann deaktiviert werden
- Mailbomben
  ☛ Größe kann limitiert werden
- Mailspam
  ☛ einzelne Hosts oder Domains können abgeklemmt werden
- Spam-Mailrelaying
  ☛ kann deaktiviert werden
- leicht fälschbare Mailadressen
  ☛ Einsatz von PGP

Weitere Informationsserver

anonymous FTP-Server

- Aktuelle Server-Version einsetzen.
- chroot()-Umgebung realisieren und ‚dummy'-Paßwort-file einsetzen.
- Korrekte Permissions der Files und Verzeichnisse beachten.
- Protokollierung aktivieren.

Nameserver (BIND)

- Viele Dienste vertrauen bei Authentisierungen auf Nameserver-Antworten (z. B. NFS, NIS, X11 ...)
- Es können falsche Namen generiert und mißbraucht werden.
- Es existieren diverse Protokollschwächen. Bind 4.9.5 bietet einige Verbesserungen (Herkunftsqualität, Minimaler Cache, Logging...).

Timeserver (NTP)

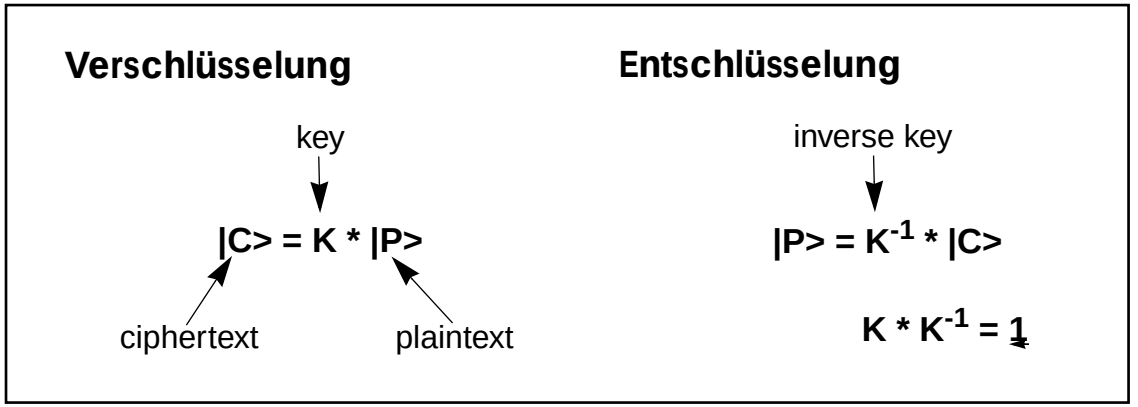
- Alle Rechner eines Clusters sollten möglichst genau zeitsynchronisiert sein.
- Dies ist sehr wichtig für die Zeiten in den Logdateien.

7 Kryptographie

Übersicht

- Grundlagen
- Secure Shell (SSH)
- Pretty Good Privacy (PGP)

Grundlagen der Kryptographie



Private Key Verfahren
(= symmetrisches Verfahren)

$$K \equiv K^{-1}$$

- Typische Verfahren DES und IDEA
  (typ. Schlüssellänge 64 oder 128 Bit)
- Nachteil: Schlüssel muß über das Netz übertragen werden

Public Key Verfahren
(= asymmetrisches Verfahren)

$$K \neq K^{-1}$$

- Es kommen zwei Schlüssel zum Einsatz
  encryption key (public) E
  decryption key (private) D

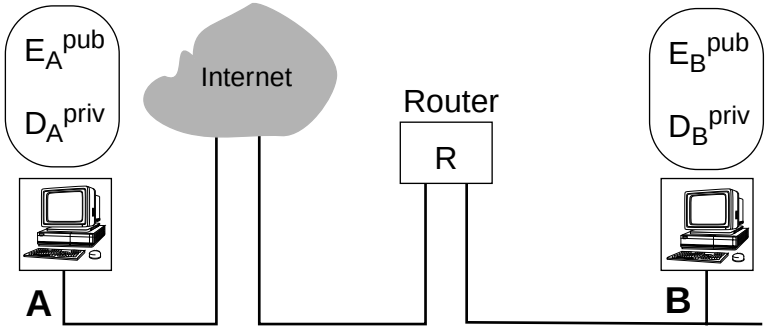
  |C> = E * |P>
  |P> = D * |C>
  E * D = 1

- Generierung z. B. via RSA (typ. Länge > 512 Bit)

Typischer Einsatz der Kryptographie

Problemstellung

- A möchte Daten an B senden, ohne daß diese im Netz abgehört werden können.
- A möchte sicherstellen, daß nur B die Daten lesen kann.
- B soll sicher sein, daß die Daten auch tatsächlich von A kommen.



Versuch mit symmetrischem Verfahren

- A verschlüsselt mit K die Daten und sendet diese an B.
- B kann nur dann die Daten lesen, falls B auch den Schlüssel K bekommt.
- Das Problem kann nicht vernünftig mit diesem Verfahren gelöst werden.

Lösung mit dem asymmetrischen Verfahren

- A verschlüsselt die Nachricht mit dem öffentlichen Schlüssel E_B^{pub} von B:

  $|C_1> = E_B^{pub} * |P>$

  Damit ist sichergestellt, daß nur B die Nachricht mit Hilfe des privaten Keys entschlüsseln kann:

  $|P'> = D_B^{priv} * |C_1> = D_B^{priv} * E_B^{pub} * |P> = |P>$

  B kann jedoch nicht sicher sein, daß die Nachricht von A kommt.

- Um dies sicherzustellen muß A die Nachricht zusätzlich mit D_A^{priv} verschlüsseln.

  $|C_2> = E_B^{pub} * D_A^{priv} |P>$

  B kann dann mit dem öffentlichen Schlüssel von A und dem eigenen privaten Key die Nachricht entschlüsseln:

  $|P''> = E_A^{pub} * D_B^{priv} * |C_2> = E_A^{pub} * D_B^{priv} * E_B^{pub} * D_A^{priv} |P> = |P>$

Übersicht - Secure Shell (SSH)

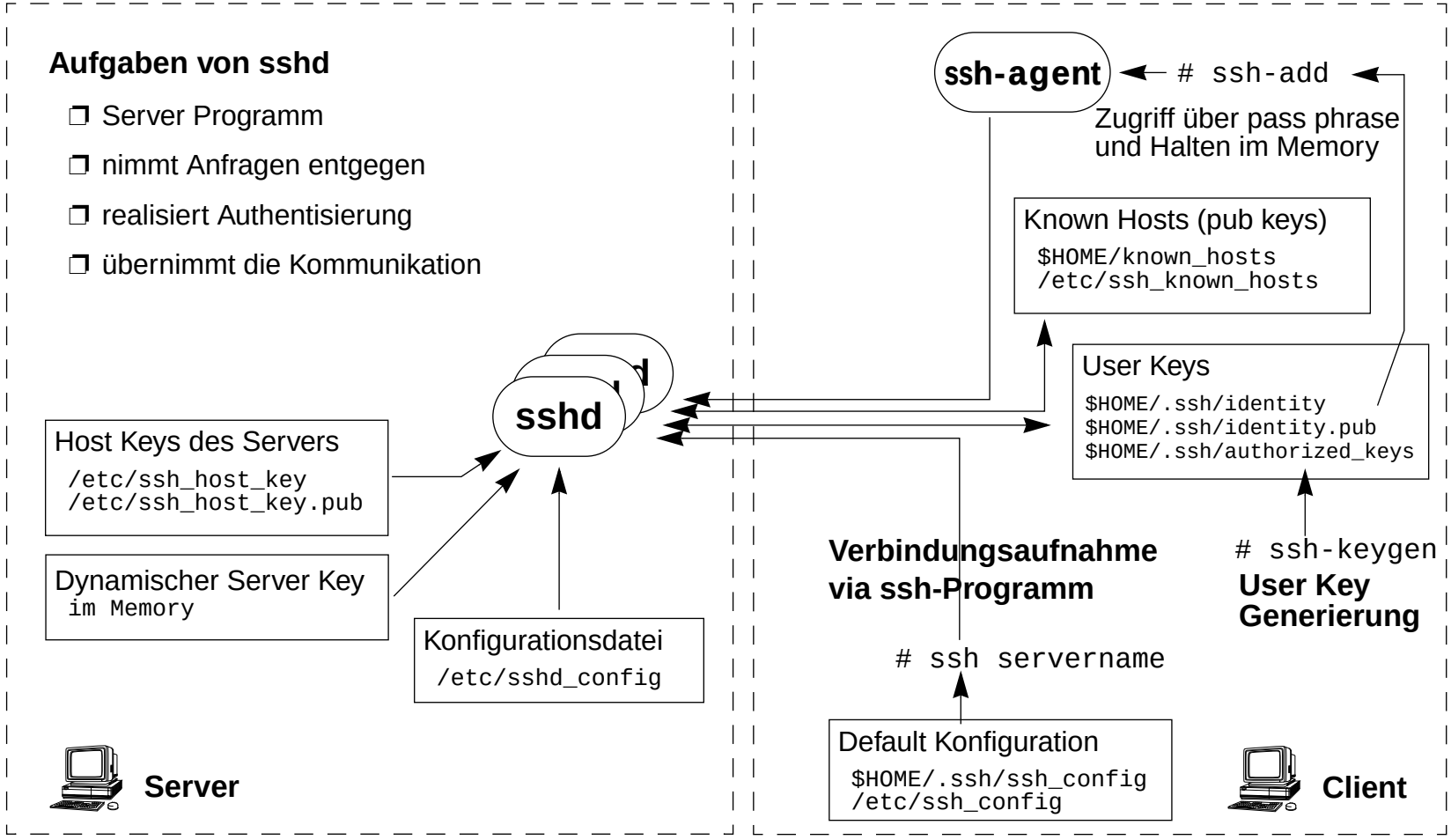
Eigenschaften der Secure Shell

verschlüsselte Kommunikation

Sicherung der Authentisierung

- Starke Authentisierung (host oder user) durch ein Public Key Verfahren (verhindert IP-, Routing- und DNS-Spoofing).
- Die Kommunikation wird transparent verschlüsselt.
- Sichere X11-Kanäle.
- Beliebige verschlüsselte Kanäle (ports) können aufgesetzt werden (z. B. auch für pop).
- Für Benutzer nur wenige Veränderungen.
- SSH ersetzt die Unix-Kommandos rlogin, rcp und rsh.
- SSH existiert auf (allen) Unix-Plattformen. Es gibt auch (teilweise kommerzielle) Versionen für Windows, OS/2 und MacIntosh.

Elemente der Secure Shell



Übersicht der diversen Keys

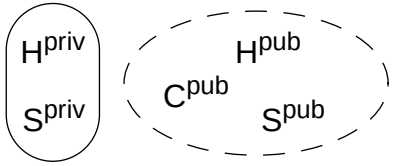
Schlüsselverteilung

am Client

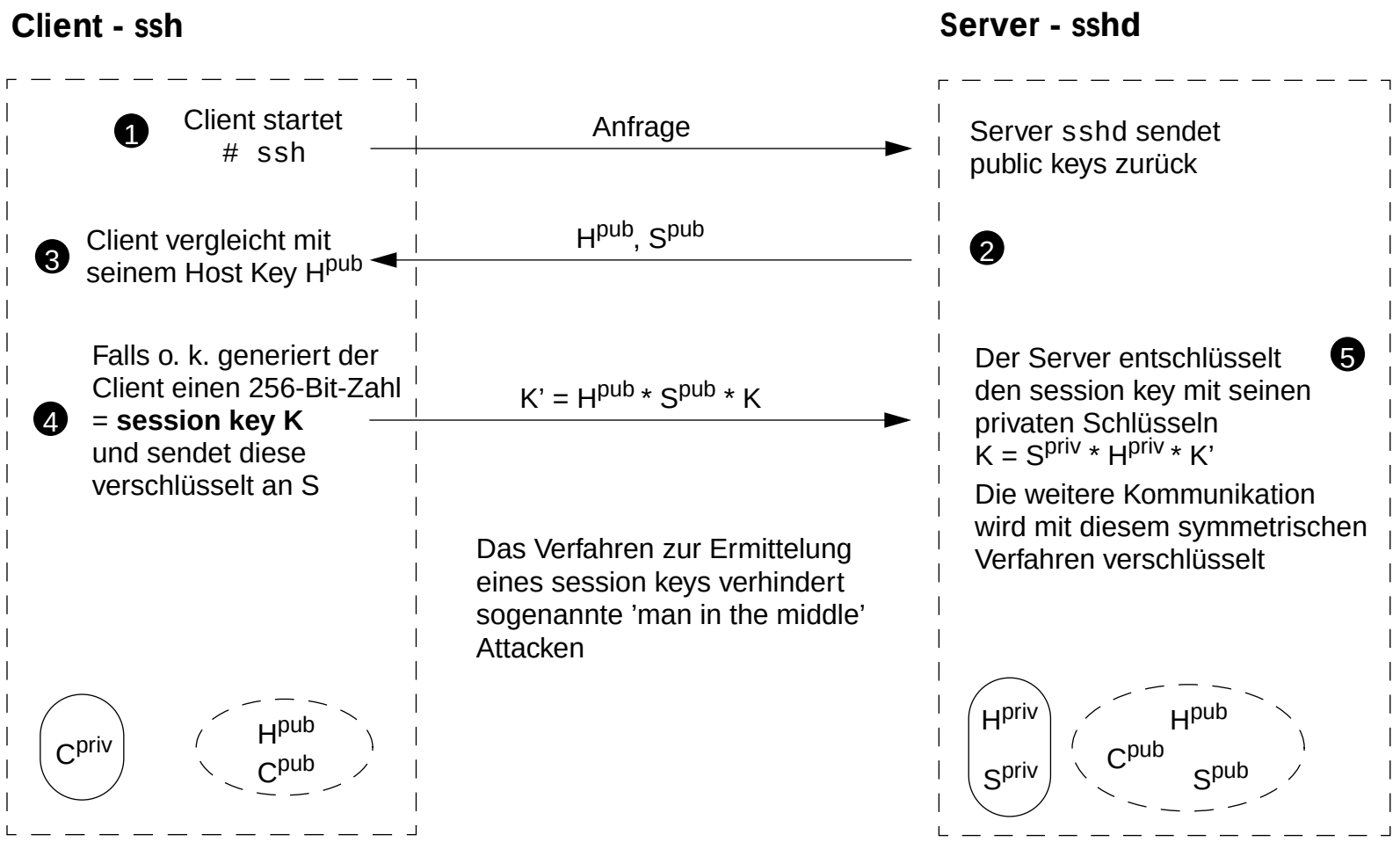
- Client Keys, die mit ssh-keygen generiert wurden:
  C^{pub} in `$HOME/.ssh/identity.pub`
  C^{priv} in `$HOME/.ssh/identity`
  Der Zugriff auf den private key ist i.d.R. noch durch eine pass phrase geschützt (d. h. das Besitzen dieses Keys nützt nichts)
- Die öffentlichen Schlüssel von Servern (d. h. Host Keys H^{pub}) in
  `$HOME/.ssh/known_hosts`
  `/etc/ssh_known_hosts`

am Server

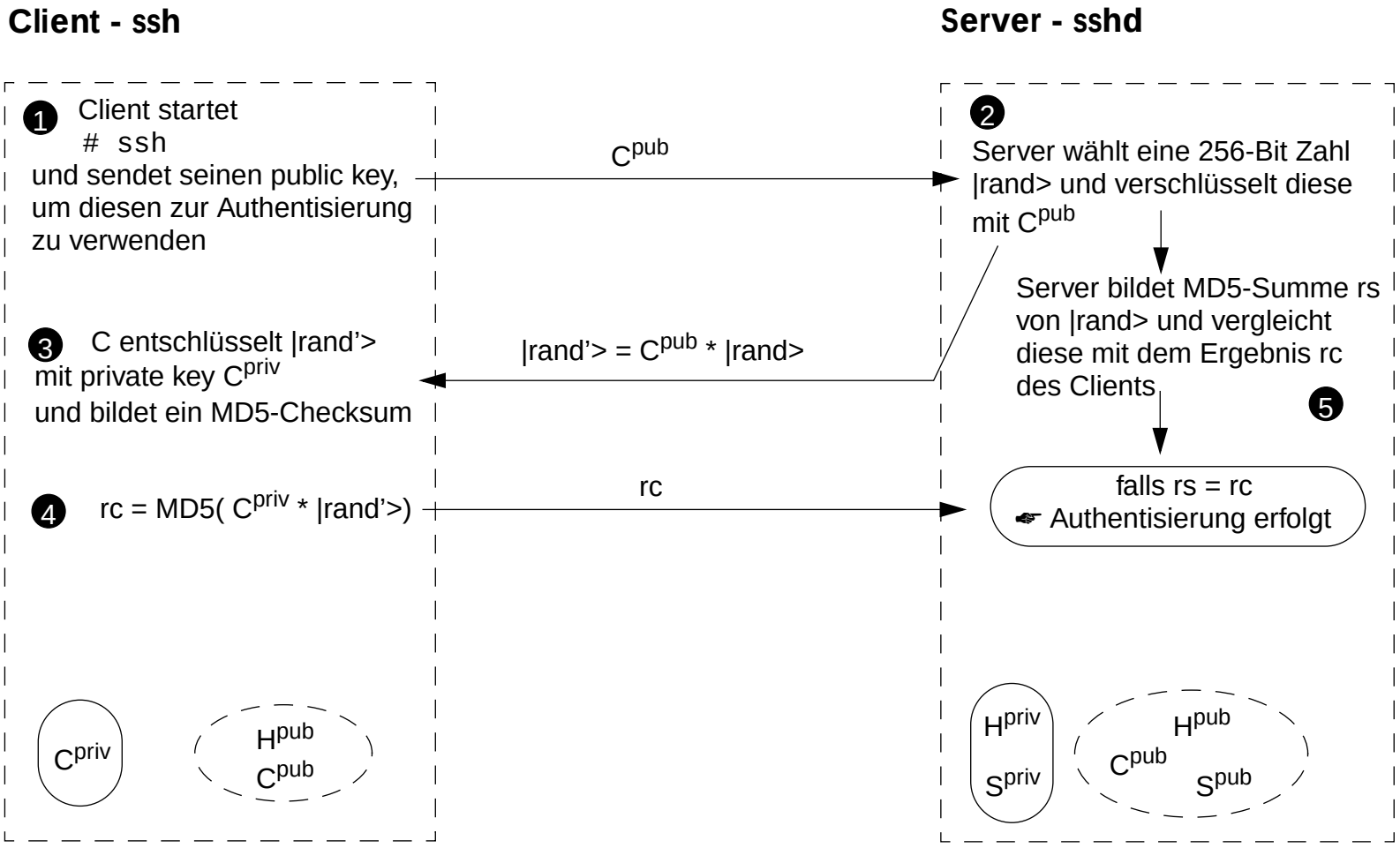
- Host Keys (1024 Bit), die bei der Installation generiert wurden:
  H^{pub} in `/etc/ssh_host_key.pub`
  H^{priv} in `/etc/ssh_host_key`
- Server Keys (768 Bits) im Memory (dynamisch erneuert)
  S^{pub} und S^{priv}
- Public Key des Clients in
  `$HOME/.ssh/authorized_keys`



Verbindungsaufnahme - session key



Authentisierungsphase



Installation und Konfiguration der SSH

Server-Installation

- Auspacken und Compilieren

```
# tar xvf ssh-1.2.26.tar
# cd ssh-1.2.26
# ./configure; make
# make install
```

- Bei make install wird ein Host-Key automatisch erzeugt
- Start des sshd-Dämons (entsprechende Einträge in einem Startup File oder von Hand)

Server-Installation mit NFS-Zugriff auf Server-Quellen

- Quellen bereits compiliert

```
# cd ssh-1.2.26
# make hostinstall
```

- Bei make hostinstall werden nur die Konfigurationsdateien und der Host-Key erzeugt.
- Start des sshd-Dämons

Client-User-Setup

- Erzeugen der Keys

```
OSF1:hrz40$ ssh-keygen

Initializing random number generator...
Generating p:  ..........++ (distance 170)
Generating q:  ..++ (distance 30)
Computing the keys...
Testing the keys...
Key generation complete.
Enter file in which to save the key ($HOME/.ssh/identity):
Enter passphrase: XXX...
Enter the same passphrase again: XXX...
Your identification has been saved in $HOME/.ssh/identity.
Your public key is:
1024 35 148557381691254631130616865765 .....
Your public key has been saved in $HOME/.ssh/identity.pub
```

- Kopie die public keys des Benutzers nach `$HOME/.ssh/authorized_keys`
- Aufruf der Secure Shell

```
OSF1:hrz40$ ssh hrz16
....
```

Arbeiten mit der SSH - Server

```
hrz16# sshd -d -p 2345 -f ../etc/sshd_config -h ../etc/ssh_host_key

debug: sshd version 1.2.26 [alpha-dec-osf4.0]
debug: Initializing random number generator;
seed file /etc/ssh_random_seed
log: Server listening on port 2345.
log: Generating 768 bit RSA key.
Generating p:  ..............++ (distance 236)
Generating q:  ..............++ (distance 170)
Computing the keys...
debug: Server will not fork when running in debugging mode.
log: Connection from 132.187.3.40 port 3828
debug: Client protocol version 1.5; client software version 1.2.21
debug: Sent 768 bit public key and 1024 bit host key.
debug: Encryption type: idea
debug: Received session key; encryption turned on.
debug: Attempting authentication for peter.
log: RSA authentication for peter accepted.
debug: Allocating pty.
debug: Received request for X11 forwarding with auth spoofing.
debug: Allocated channel 0 of type 1.
debug: Forking shell.
debug: Setting controlling tty using TIOCSCTTY.
debug: Entering interactive session.

debug: X11 connection requested.
debug: Allocated channel 1 of type 3.
debug: Received channel open confirmation.
debug: Channel now open, status bits 0
```

Arbeiten mit der SSH - Client

```
hrz40# ./ssh -v -p 2345 hrz16

SSH Version 1.2.21 [alpha-dec-osf4.0], protocol version 1.5.
Standard version.  Does not use RSAREF.
hrz40.rz.uni-wuerzburg.de: ssh_connect: getuid 2377 geteuid 2377 anon 1
hrz40.rz.uni-wuerzburg.de: Connecting to hrz16 [132.187.3.16] port 2345.
hrz40.rz.uni-wuerzburg.de: Connection established.
hrz40.rz.uni-wuerzburg.de: Remote protocol version 1.5, remote software version 1.2.21
hrz40.rz.uni-wuerzburg.de: Waiting for server public key.
hrz40.rz.uni-wuerzburg.de: Received server public key (768 bits) and host key (1024 bits).
hrz40.rz.uni-wuerzburg.de: Host ,hrz16‘ is known and matches the host key.
hrz40.rz.uni-wuerzburg.de: Initializing random
hrz40.rz.uni-wuerzburg.de: Encryption type: idea
hrz40.rz.uni-wuerzburg.de: Sent encrypted session key.
hrz40.rz.uni-wuerzburg.de: Received encrypted confirmation.
hrz40.rz.uni-wuerzburg.de: No agent.
hrz40.rz.uni-wuerzburg.de: Trying RSA authentication with key ,peter@hrz40.rz.uni-wuerzburg.de‘
hrz40.rz.uni-wuerzburg.de: Received RSA challenge from server.
Enter passphrase for RSA key ,peter@hrz40.rz.uni-wuerzburg.de‘: XXXXXXX.........
hrz40.rz.uni-wuerzburg.de: Sending response to host key RSA challenge.
hrz40.rz.uni-wuerzburg.de: Remote: RSA authentication accepted.
hrz40.rz.uni-wuerzburg.de: RSA authentication accepted by server.
hrz40.rz.uni-wuerzburg.de: Requesting pty.
hrz40.rz.uni-wuerzburg.de: Requesting X11 forwarding with authentication spoofing.
hrz40.rz.uni-wuerzburg.de: Requesting shell.
hrz40.rz.uni-wuerzburg.de: Entering interactive session.
Last login: Wed Dec  3 15:43:47 1997 from hrz40.rz.uni-wu
Digital UNIX V4.0B  (Rev. 564); Wed Sep 17 15:46:34 MET DST 1997

!!!!!   Achtung! Auf dieser Maschine laeuft Digital UNIX 4.0.      !!!!!

...
```

Arbeiten mit der SSH - Client

```
hrz40# ./ssh -v -p 2345 hrz16

...
hrz40.rz.uni-wuerzburg.de: Entering interactive session.
Last login: Wed Dec  3 15:43:47 1997 from hrz40.rz.uni-wu
Digital UNIX V4.0B  (Rev. 564); Wed Sep 17 15:46:34 MET DST 1997

!!!!!   Achtung! Auf dieser Maschine laeuft Digital UNIX 4.0.      !!!!!

Environment:
  HOME=/rzuw/peter
  USER=peter
  LOGNAME=peter
  PATH=/bin:/usr/bin:/usr/ucb:/usr/bin/X11:/usr/local/bin:/opt/Tools/SSH/install/bin
  MAIL=/var/spool/mail/peter
  SHELL=/bin/ksh
  SSH_CLIENT=132.187.3.40 3828 2345
  SSH_TTY=/dev/ttyp8
  TERM=xterm
  DISPLAY=hrzx16.rz.uni-wuerzburg.de:10.0

Running /usr/bin/X11/xauth add hrz16.rz.uni-wuerzburg.de:10.0 MIT-MAGIC-COOKIE-1 1a5a1240fbbe06...
Running /usr/bin/X11/xauth add 132.187.3.16:10.0 MIT-MAGIC-COOKIE-1 1a5a1240fbbe06d615d13dac2f5...

OSF1:hrz16$
OSF1:hrz16$ xclock &
hrz40.rz.uni-wuerzburg.de: Received X11 open request.
hrz40.rz.uni-wuerzburg.de: Allocated channel 0 of type 9.
hrz40.rz.uni-wuerzburg.de: Sending open confirmation to the remote host.
OSF1:hrz16$
```

Pretty Good Privacy (PGP)

Möglichkeiten

- Einsatz von asymmetrischen Verschlüsselungsverfahren beim Mailen
- Die Mail kann nur vom gewünschten Adressaten gelesen werden.
- Der Adressat kann sicher sein, daß die Nachricht vom richtigen Absender stammt.

Einsatzmöglichkeiten und Probleme

- Beim Mailen werden meist Signaturen an das Dokument angehängt. Mit Hilfe des Public Keys des Senders kann der Empfänger überprüfen, ob die Mail tatsächlich von diesem Sender stammt und ob diese verändert wurde.
- Der Public Key kann z. B. über die Homepage des Senders o. ä. bereit gestellt werden. Besser jedoch sind sogennante Public Key Server.
- Mailprogramme wie elm bieten die Möglichkeit, PGP angenehm einzusetzen.