



Linux Security Basics

Linuxwochen 2003

<gleber@aon.at>

<christian.mock@vibe.at>

7.6.2003

<http://www.vibe.at/aktionen/200109/>

Über uns

- Unix/Linux Sysadmins @PING/EUnet, Adcon
- Security-Abteilung Xsoft
- Software-Entwicklung Adcon (Embedded Linux)
- Nunmehr CoreTEC IT Security Solutions
 - Security Checks
 - Firewalls & Zubehör
 - Consulting
- VIBE!AT, priv.at, LUGA, und auch sonst überall die finger drin...

Inhalt

- Was ist Security?
- Was ist eigentlich...?
 - Firewall, Intrusion Detection, Crypto, Unix-Permissions...
- Wie härte ich mein System?

Einführung...

- Was ist Security?
 - "CIA": Confidentiality, Integrity, Availability
 - Also nicht nur gegen böse Cracker, sondern auch gegen Schnüffler und Naturkatastrophen
- Warum brauch ich Security?
 - Scriptkiddies und Würmer suchen sich ihre Ziele nicht aus, sondern nehmen, was daherkommt
- Es gibt keine 100%ige Security!
 - Hauptaufgabe ist Risk Management

"security is a process" (Schneier?)

- risiko-analyse
- entscheidung über schutzmaßnahmen
- implementation derselben
- betrieB, up-to-date-halten
- überwachung
- back to square one

"principle of least privilege"

- wichtiges grundprinzip
- usern, prozessen, maschinen... nur soviel rechte geben wie notwendig
- z.B:
 - sparsam mit "other"-rechten im filesystem
 - firewalls als "default deny" (fail close) konfigurieren
 - special-use-usern keine shell geben
 - daemonen nicht als root und chroot(2)ed

Wer ist der Gegner?

- interne vs extern (60-80% intern?)
- cracker vs script kiddies
- würmer und andere automaten
- spammer
- beamtete und andere schnüffler
- der zufall

was ist eigentlich

- firewall
- intrusion detection
- https
- ssh
- VPN / IPSEC
- permissions, umask

firewalls

- übergangspunkt zwischen netzen verschiedenen trust-levels (nicht nur am access-router!)
- aller verkehr geht drüber -> einziger enforcement point, (relativ) gut unter kontrolle zu halten damit auch Single Point Of Failure
- **hat keine wunderkräfte!**
 - wenn schlecht konfiguriert, hilfts nix
 - was durchgelassen wird, ist i.A. nicht gesichert
 - WLANs: auch exponiert, aber hinter der firewall (WEP ist unsicher)

firewall-typen: application layer

- behandeln datenströme auf transport/application layer
- "plugs"
- können bei korrekter implementierung z.B. vor telnetd-buffer overflow schützen, aber kaum vor code red (gültige HTTP-requests)
- performen ned für high-volume-traffic (http) weil kernel->userspace->kernel
- Bsp: fwtk, plugger

firewall-typen: packet filter

- schauen IP-pakete an, mit seitenblick auf TCP/UDP header
- primitive PF sehen nur einzelne pakete
 - problematisch für UDP (NIS, DNS, ...)
 - TCP: checken ACK-bit, damit ACK-scans möglich
 - schnell wie hölle
- Bsp: Cisco ACL, ipchains/ipfwadm

firewall-typen: stateful packet filter

- versuchen, zusammenhang zwischen paketen zu sehen
- TCP: jenachdem, wie implementiert, besser oder schlechter (welcher SPF checkt sequence numbers?)
- UDP: heuristik "rückpaket muß innerhalb X sekunden kommen"
- recht schnell
- Bsp: iptables, ipfilter (*BSD), FireWall-1

firewall: auswahlkriterien

- auskennen
- preis
- GUI notwendig? ease of config
- logging-qualitäten
- nach bekannten security-holes recherchieren

intrusion detection

- wenn die firewall die verstärkte tür ist, ist das IDS die alarmanlage
- üblicherweise sniffer mit pattern-matching und mehr oder weniger buntem GUI
- sehr interessant, wenn man weiß, was das ding eigentlich meldet
- aufwand zum tunen nicht unerheblich
- automatische gegenschläge sind BÖSE[tm]
- Bsp: snort, RealSecure, CyberCop, ...

SSL/TLS

- crypto auf transport layer
- in applikation implementiert...
- ...oder gewrapped: sslwrap, stunnel
- verschlüsselung und authentisierung
- üblicherweise nur server gegenüber client authentisiert
- nur so gut wie die zertifikate (+ verifikation!)

SSL/TLS (2)

- für viele TCP-basierende protokolle heute unterstützt/definiert:
 - imap, pop3 (z.B. in netscape, fetchmail; server per wrapper)
 - nntp (netscape, inn)
- recht einfach zum aufsetzen, wenn man openssl begriffen hat
- CERTs generieren nicht so trivial (pyCA, openCA)

https: http+SSL

- verbreitetste SSL-anwendung
- in fast allen browsern vorhanden
- daher sehr gut für web-apps zu verwenden
- üblicherweise authentisiert sich der server gegenüber dem client, um spoofing zu verhindern
- client kann sich auch authentisieren, damit strong auth
- apache: mod_ssl, apache-ssl

ssh

- "rsh on crypto-steroids"
- starke verschlüsselung (verschiedene ciphers)
- starke authentisierung (public key crypto)
- X11-forwarding
- port forwarding vorwärts und zurück
- ssh-agent/ssh-add: authenticate once, use many
- scp: sicheres file-kopieren (rcp-ersatz)

VPN

- "virtual private network"
- ersatz für standleitungen und private dialups
- arbeiten üblicherweise am network layer
- daher für applikationen transparent
- diverse ansätze: proprietäre, hacks (PPP over ssh), IPSEC, CIPE, vtun, ...

IPSEC

- VPN-standard
- breite unterstützung
- meistens interoperabilität
- ziemlich komplex
- für linux: FreeS/WAN

permissions

- d**rwxrwxrwx**
- file type (regular, dir, socket, char/block special...)
- **user**
group
other
jeweils **read write execute**
- noch mehr: **chattr/lsattr**

permissions auf files

- `if(fileuser == user)`
 `return checkuserperms()`
`else if(filegroup == group)`
 `return checkgroupperms()`
`else`
 `return checkotherperms`
- d.h. user mock darf file
 `---rwxrwx mock mock`
 nicht lesen/schreiben/executen!

permissions auf dirs

- r: lesen, d.h. liste der files bekommen
- w: schreiben, d.h. files/subdirs anlegen/löschen
- x: chdir und generell befragen von subdirs
- d.h. permission zum löschen eines files hängt von der write-permission des directories, in dem es ist, ab!

suid-bits auf files

- `-r-sr-sr-x`
- owner-suid: bei execute läuft prozess mit uid des file-owners
- group-suid: prozess läuft mit gid des group owners
- other-suid: AFAIK nicht mehr in verwendung ("t", "sticky bit")
- suid-programme erlauben das ausführen unter anderen uids -> **gefährlich!**

suid-bits auf dirs

- owner-suid: noop, AFAIK
- group-suid:
bei file-creation, wenn user die entsprechende secondary group hat, bekommt file diese group
BSDgroups-verhalten, siehe mount(2)
- Other-suid ("t"):
directory-owner oder file-owner darf files löschen
oder umbenennen (/tmp: drwxrwxrwt)

umask

- welche bits der permission werden bei file-/dir-creation *nicht* gesetzt?
- "offener" standard: 022 (oktal), d.h. write für group/other wird nicht gesetzt
- "paranoid": 077, d.h. keine permissions für group/other
- "dumm": 0, d.h. `-rwxrwxrwx`

chattr/lsattr

- "file attributes": erweitere permissions
- security-relevant:
 - i: immutable
 - a: append-only
 - s: secure delete
 - u: undelete möglich
- chattr +-flags file ...
- lsattr dir/file

warum sollte ich nicht:

- dauernd als root arbeiten?
- telnet verwenden?
- rlogin/rsh verwenden?
- unverschlüsseltes POP3 verwenden?
- unverschlüsselte mails senden?
- mein OS in der "installiere alles"-Variante aufsetzen?

OS hardening

- aka "ich hab mein *nix aufgesetzt, kann ich jetzt ans netz?"
- warum?
 - netzwerk-daemons
 - standard-accounts
 - software-config
 - passwords

Wie finde ich raus, was überhaupt läuft?

- `netstat -aA inet`
zeigt alle in-use inet sockets an, TCP LISTENING und udp sind interessant
- `lsof -I protocol@host:port`
zeigt prozess dazu an
- portscan von anderem rechner aus
 - nmap
 - strobe
 - nessus

Wie entscheide ich, ob ich das brauche?

- wir erinnern uns ans "principle of least privilege": wenn ich nicht sicher bin, daß ich's brauch, weg damit!
- herausfinden, zu welchem paket der daemon gehört ("`dpkg -S /file/name`", "`rpm -qf /file/name`") und doku lesen
- trial and error
- vorsicht bei abhängigkeiten, z.b. gnome und KDE

wie werd ich's wieder los?

- am besten: paket deinstallieren, wenn nicht gebraucht
- herausfinden, woher gestartet:
 - /etc/rcX.d, /etc/rc.d/init.d
 - /etc/inittab
 - /etc/inetd.conf, /etc/xinetd.conf, /etc/xinetd.d
 - sonstwo
- ...und deaktivieren

wie überprüf ich's nachher?

- netstat
- portscan
- security scanner

nützliche Tools aufsetzen

- ssh (inkl key-mgmt)
- ipchains + port-forwarding
- apache + SSL
- logcheck
- snort
- tripwire

Betrieb (logs lesen, firewalls,...)

- Was gibt's zu tun?
 - logs lesen
 - Unregelmässigkeiten investigieren
 - tripwire laufen lassen
 - **software up-to-date halten:**
 - bugtraq@securityfocus.com,
 - vendor-security-mailinglists,
 - RedHat up2date, redcarpet
 - SuSE yast2 + finger brechen