Sicherheit von Linux Systemen

Tino Reichardt
treichar@fh-lausitz.de

30. März 2001

Inhaltsverzeichnis

1 Überblick und Ziel des Vortrages

```
Eine kleine Geschichte aus de.comp.security.firewall (glaube ich):
Q: Wie mache ich meinen Rechner sicher ?
A: 1. ein Betriebssystem installieren
   2. alle Kabel, außer Strom und Grafik, vom Rechner entfernen
   3. den Rechner in einen Raum, in einem Bunker, welcher sich 10m
      under der Erde befindet, schaffen
   4. den Raum, nach sorgfältigem Anschluß der Kabel an die
      Notstromversorgung usw., verschließen
(so sollte man es log. weise nicht machen, jedoch ist der ansatz
richtig, je paranoider...)

Ziel dieses Vortrages:
- sicherheitsrelevante Begriffe erklären
- aufzeigen verschiedener Angriffe + Schutz vor diesen
- es wird nur auf lokale und IP-basierte Angriffe gegen Linux eingegangen
- es wird nicht auf Windows eingegangen
- Linux-Distributionen und deren Unterschiede sind auch nicht Thema
  (jedoch deren Standard-Software)
```

2 Begiffe

2.1 Denial of Service

Im computerbezogenen Sinne: das verhindern das ein Rechner/Server seinen Dienst anbieten kann.

2.2 Packetfilter

Oft auch als Firewall bezeichnet. Ist aber eigentlich nur ein Stück Hardware/Software welche nach bestimmten vordefinierten Regeln Packete wegschmeißt/ablehnt/durchläßt. Sie sollten sehr gut mit der OS-Ebene zusammenarbeiten.

2.3 Firewall

Als Firewall bezeichnet man ein organisatorisches und technisches Konzept zur Trennung von Netzbereichen, dessen korrekte Umsetzung und dauerhafte Pflege. Ein oft benutztes Instrument der Umsetzung ist ein Stück Hardware, das zwei physisch getrennte Netzbereiche genau so verbindet, wie es im Konzept zugelassen wird. Dieses Stück Hardware bezeichnet man als Firewall-Rechner/System oder verkürzt als Firewall.

2.4 Hacker $\neq$ Cracker

Hacker = Leute die ständig mehr Wissen wollen (oft auch SystemAdministratoren)
Cracker = Leute welche ihre sehr guten Kenntnisse negativ nutzen

2.5 Exploits

Sind vorgefertigte Sources/Programme, welche bekannte Sicherheitslücken ausnutzen. Werden oft als Referenzen und Beweismaterial benutzt, um zu zeigen, das in gewissen Programmen wirklich Fehler ausgenutzt werden können. Eine sehr bekannte Quelle ist www.securityfocus.com (BugtraqMailingliste).

2.6 Script Kiddy's

Personen, welche vorgefertigte Exploits, als auch andere (sogenannte) Hackingtools runterladen und ausführen, ohne zu wissen was sie bewirken bzw. wie sie funktionieren.

2.7 Auditing

Ist quasi eine systematische Suche nach Fehlern im Konzept (bzw. vom Aufbau) eines Systems. Wobei das System ein ganzes Unternehmen, ein einzelner Rechner, oder was man auch immer spezifiziert, sein kann. Man bedient sich meist Audit-Tools (LogAnalysierer) und einer vorher def. Audit-Policy (Regelwerk).

2.8 Schichtenmodelle

- OSI = Schichtenmodell, welches neben den Schichten des TCP/IP Modells noch die Schichten der Darstellung und Sitzung beinhaltet
- dieser Standard hat sich nie richtig durchgesetzt, wächst aber zum Teil mit

TCP/IP zusammen [1]

	OSI	TCP/IP	Bemerkung in Bezug zu TCP/IP
7	Anwendung	Verarbeitung	ftp, smtp, ssh, etc...
6	Darstellung	—	
5	Sitzung	—	
4	Transport	Transport	TCP und/oder UDP
3	Vermittlung	Internet	IP + ICMP
2	Sicherung	Host-an-Netz	
1	Bitübertragung	Host-an-Netz	

[1] siehe Computernetzwerke von Andrew S. Tanenbaum (ISBN 3-8272-9536-X)

3 Angriffe

3.1 Einteilung von Angriffen

- sehr schwierig, meistens führen Kombinationen von verschiedenen Angriffen erst zu einem Problem
- meist in folgende Einteilungen, die sich aber auch überschneiden können:

```
- lokal/remote Angriffe
- Art der Ausnutzung von Fehlern (Bufferoverflow, /tmp/races, Formatstrings)
- was man davon hat (root-account, oder nur uid des daemons)
```

- im Grunde genommen, ist die folgende Einteilung nich unbedingt empfehlendwert, soll aber vor diesen kleinen Vortrag reichen
- im allgemeinen gilt, wenn jemand in einen Rechner eingedrungen ist, wird er versuchen, diverse lokale exploits zu fahren, um einen Nutzen zu erlangen (und wenn er _nur_ eine Passwortliste eines anderen Rechners durch jack laufen läßt)

3.2 Angriffe (lokal)

3.2.1 Trojaner

- durchaus möglich, aber sehr selten (weil meist offene Sources)
- aber: sogenannte root-kits (z.B. angepaßtes ps+netstatprogramm)

3.2.2 Viren

- entfällt im allgemeinen bei Linux
- es gibt neuerdings aber erste Ansätze systemübergreifende Viren (Win32+LinuxELF) unter GPL zu stellen :)[2]

3.2.3 Fehlkonfigurationen

- das größte Problem für Linux
- Administratoren sind oft froh das etwas läuft, security steht viel zu (leider) erst an hinterer Stelle (wenn überhaupt)

3.2.4 buffer overflows

- das gößte Problem unter UNIX, und somit auch unter UNIX-artigen Systemen, wie Linux eines ist
- das Hauptproblem liegt an der Programmiersprache C, welche zwar schnell und recht hardwarenah ist, aber mit der manuellen Speicherverwaltung haben viele Programmierer ihre Probleme (nicht zu vergessen die Zeigers :))

3.2.5 suid Programme

- auch ein Problem, das nur unter UNIX zu finden ist
- unter UNIX gibt es eben die klassische Rechteverwaltung USER/GRUPPE/ANDERE
- wobei mache Programme besondere Privilegien benötigen um bestimmte Geräte ansprechen zu können, sind diese Programme fehlerhaft und stürzen ab, können Cracker (nachdem sie es vorher auf ihrem Rechner getestet haben, und die entsprechenden Speicherstellen kennen) einen Shellcode schreiben, welcher direkt nach dem Absturz ausgeführt wird (meist eine simple Routine, die eine root-shell zum vorschein bringt)

[2] siehe http://www.heise.de/newsticker/data/ps-28.03.01-000/

3.2.6 /tmp races

- machne Programme legen im /tmp-verzeichnis temporäre Dateien (mit spziellen Rechten) ab, ohne zu überprüfen, ob dort schon eine gleichnamige Datei existiert, ... insgesamt sind diese Angriffe sehr unterschiedlich, mal mit symlinks, mal eine ständige Überwachung, und dann ein zuschlagen im richtigen moment (Prog macht erst ein chown, dann ein chmod() - wenn überhaupt, etc...
- durch dieses ständige überwachen der file - /tmp/_races_

3.2.7 format strings

- seit 1999 bekannt, wieder eine Abart von C
- seit 2000 werden exploits (ohne Ende) dafür geschriebn, Programmierer sehen vor lauter Sanity-Chacks das eigentliche Programm nicht mehr, etc...
- char *lala;printf(%400s;lala); - stack, shellcode, ...

3.2.8 lilo

- immer wieder herrlich, wenn man einen Rechner vor sich stehen hat, der Betreuer daneben steht und sagt, nun zeige mir mal wie du hier root werden willst (PowerOff, lilo-prompt fragt nicht viel also TAB, aha vmlinuz heißt der kernel, also: boot=vmlinuz init=/bin/bash, aha: root#)

3.2.9 globbing

- lange file in einer user-shell (welche globbing unterstützt) starten, mit touch ne langelange file erstellen, mit ls *a*a*a*a*b die datei anzeigen lassen, ups (was macht der Rechner nur)

3.2.10 Passwortangriffe

- mehrere Arten der Speicherung von Passwörtern im herkömmlichen Sinne (kein npasswd, kein yppasswd): passwd/shadow group/gshadow des/md5...
- passwd: name:pw:uid:gid:zusatz,zusatz2:/homedir:/homeshell
- group: name:pw:gid:user1,user2

3.3 Angriffe (netz)

3.3.1 Sniffer

- zum abhören des lokalen Netzwerktraffiks, indem man seine Netzkarte in den promiscous Mode schaltet, macht sie eine Kopie des gesamten Ntztraffiks, diesem kann man log.weise speichern und filtern
- oft kommen dabei viele viele Passwörter zum vorschein

3.3.2 Traffik

- plumpe Methode, um z.B. w3 Seiten lahmzulegen
- man braucht aber eine gute Anbindung

3.3.3 Geschickte Packetwahl

- elegantere Methode, es gibt da mehrere Möglichkeiten
- nennen möchte ich mal folgende:

....

3.3.4 Scanner

- zum scannen verschiedener/aller Ports eines Rechners, um bestehende (und somit angreifbare) Dienste eines Rechners zu bestimmen
- besonders bekannt und gut ist nmap, ein tool, welches jeder Admin kennen sollte
- ein weiteres wäre netcat, das sogenannte Schweizer Taschenmesser eines Admins

3.3.5 Spoofing

- verfälschen des Absenders
- läuft auf mehreren Protokollebenen (IP/ARP/DSN/MAC/etc..)

3.3.6 Unverschlüsselte Verbindungen

- Standarddienste wie W3, POP, EMail, FTP, Telnet usw. sind alle unverschlüsselt, es sei denn man nutzt ihre (im laufe der Zeit hinzugekommenden) Erweiterungen hinsichtlich der Sicherheit
- also https, Apop, Imap, sftp, ssh etc...

3.3.7 X11

- neben vielen Bufferoverflows in diverses Standard-Apps der X-Distribution, sind oft die USER selbst Schuld
- die meisten nutzen ein simples ẍhost fremde-ipüm Programme anderer Rechner auf sein eigenen zu bringen - dies führt jedoch dazu, das jeder User dieses anderen Rechner zugriff auf den eigenen Bildschirm hat...
- Ausweg: xauth, basiert auf schlüssel, kann mit ssh automatisiert werden

4 Schutz

4.1 Uralt Services

- deaktivierung uralter Services wie Telnet, und völliger Umstieg auf ssh (tssh unter Windows ist frei verfügbar)
- allg. Abschaltung veralterter Dienste, und update zu den entsprechenden extensions
- ruhig mal eine Anleitung zum Programm lesen, und nicht nur herumprobieren
- services auf eine Anzahl beschränken, die man überwachen kann
- services auf mehrere Rechner verteilen, Redundanz wird in Sachen von aufkommenden QoS usw. immer wichtiger

5 Abkürzungen

IETF: Internet Engineering Task Force (RFC's + TCP/IP Modell)
IRTF: Internet Research Task Force (IETF ist Teil davon)
ISO: International Organization for Standardization (OSI-Modell)
ISOC: Internet Society (Sammelbegriff für Netzuser, seitens der IETF)
MIB: Management Information Base
OSI: Open Systems Interconnection
RFC: Request for Comments
TCP: Transmission Control Protocol
UDP: User Datagram Protocol
IP: Internet Protocol
ICMP: Internet Control Message Protocol
FTP: File Transfer Protocol
WWW: World Wide Web
SSH: Secure Shell (Telnet Ersatz)
SNMP: SimpleNetworkManagementProtokoll
DoS: Denial Of Service
ISP: Internet Service Provider