Network Security Attacks

Network Mgmt/Sec.

Jim Binkley

Outline

- methodologies/motives
- the original worm and the lessons we learned (sic)
- DOS attacks of late
- some recent attacks of note
- network analysis and passwords
- ip address authentication
- a short word on viruses and “mobile code”
- tcp and sequence numbers

outline cont.

- sw engineering/fuzz revisited
- and patching
- sources of information on exploits/bugs/etc.
- lessons learned are what?

methods of attack

- scanning and exploits
  - scanning may include using search (google) as well as more traditional methods (nmap)
- social engineering
  - phishing/"trojans" (zlob/dnschanger)
- shooting yourself in the foot
  - you don't need to be social engineered
  - the school put your SSN up on the web
  - ok call it **information disclosure**
  - you downloaded marketscore or zlob/dnschanger

but first: what is the motive for the crime?

- traditional: “it’s fun”
- modern: it’s a business
  - selling viagra, used-cars, porn services, money laundering, drugs, stolen goods, blackmail, and who knows what else
  - spam, spam, spam
    - hacking web-sites to post web-spam
- point to ponder: just because you’ve been hacked doesn’t mean that 1: the hack works and 2. they have decided just what to do with you yet

one methodology of the attackers

- surveillance
  - find hosts (IP address search)
  - find type of host (os fingerprint), firewalls too
- find KNOWN bugs (known to them)
- exploitation post break-in
  - escalation of privilege, user attacks root
- hiding their tracks post Or pre break-in
  - root shells on UNIX

so scanning is one basic methodology

- finding ip dst addresses
  - single source
  - multiple sources
- scanning one ip dst
  - for tcp ports/udp ports open
  - single source
  - multiple sources
- then launch an exploit
  - launcher may be human or program

email: another methodology

- send program via email
- user naively executes attachment
  - or perhaps it is auto-launched in some cases
  - social engineering may be of use
    - “hi handsome ...”
- malware uses address book to launch itself at next targets
  - possibly with fake email sender

define some terms

- **exploit** - a piece of code that exploits a software bug leading to a security hole
- **virus** - a malware program that somehow rides on the back of another vehicle
  - but doesn't move itself
- **worm** - a malware program that provides its own transit
- **trojan-horse** - a malware program that somehow appears as something else entirely

more terms

- **footprint/signature**: some log entry or other trace left behind by an attack
- **signature**(in IDS sense): some way to identity a particular virus/worm/exploit attack
  - perhaps use pattern matching to id that a file/email/packet has a known attack in it
- **forensics**: the process of figuring out just how an attack occured after the attack succeeded
  - possibly may include collecting evidence for criminal case against criminal defendent

more terms

- **forensics again:**
  - **important idea: if we can’t figure out how they got in, how can we keep them out next time?**
- **counter-measures**: just what the white-hats do to keep the black-hats out
  - or what you do to WATCH for them
    - » on your network or hosts
  - what did you do to make your web-server safer?

one more ...

- an optimizer does not produce optimal code
- therefore define “secure”:
- maybe we should all say: “safer”
- or less-insecure
- there is no such thing as safe, or secure

more terms

- backdoor
- social engineering attack
- buffer overflow
- dictionary attack
- oh wait, we have the Morris worm for those terms

1988 - the Morris worm: problems included:

- fingerd gets does not check buffer-length on input - results in root shell for attacker
  - » **buffer-overflow** attack
- an idiot bug in sendmail that allowed attacker to fire up shell
  - – DEBUG opttion not turned off
- using rsh/rcp/rshd .rhost scheme (IP address authentication) to break into nearby sites (exploitation post break-in)

cont.

- password attacks
  - try built-in dictionary, try idiot guesses (no passwords), try /usr/dict/words
  - read /etc/passwd “result” and try to match
- fanout attacks included
  - looking at .forward since if we cracked this system, maybe user has same password on that system?
- worm tried to hide (fork and kill parent)

fingerd program BEFORE

- char line[512];   /* automatic storage */
- line[0] = ‘\0’
- gets(line); /* user to be fingered from stdin*/
- Morris fed it a carefully constructed program that caused a root shell to be executed

VAX buffer attack code

```
pushl $68732f ‘/sh\0’
pushl $6e69622f ‘/bin’
movl  sp, rl0
pushrl $0
pushrl $0
pushrl r10
pushrl $3
movl sp, ap
chmk $3b
```

result equivalent to:

- execve("/bin/sh", 0, 0);
- so root shell executed when main returned
- attacking system would have TCP connection to root shell and could proceed with other attacks
- lesson: **buffer overflows attempt to gain a privilege via an already privileged server**
  - UNIX attacks/exec or write (e.g., write passwd)

here’s the fix, but it doesn’t take

- **fgets(line,sizeof(line),stdin);** /* the fix */
- other fixes include:
  - no stack execution
  - Crispin Cowan’s stackguard
- defence mechanisms may include:
  - staying patched (hard especially if 1000 systems)
  - firewalls
- don’t use C, after all, java doesn’t have security problems (sic)

lessons learned

- explicit check for stack overflow!?
  - – yes, but the program was the 1st java applet?
- passwords should not be in the dictionary
  - – /etc/passwd should not be readable by the world (cut down on brute-force “crack” attempts)
- permissions of daemons should not be root
  - – limited to that daemon only (least privilege)

cont.

- weak authentication mechanisms based on ip address (.rhosts) should be viewed with circumspection
  - lots of cases of this though, NFS, DNS, rsh of course, ACL firewall mechanisms
- sendmail too (check out qmail)
- Ref: The Internet Worm Program: An Analysis, Eugene H. Spafford, Purdue, CSD-TR-823, November 1988.

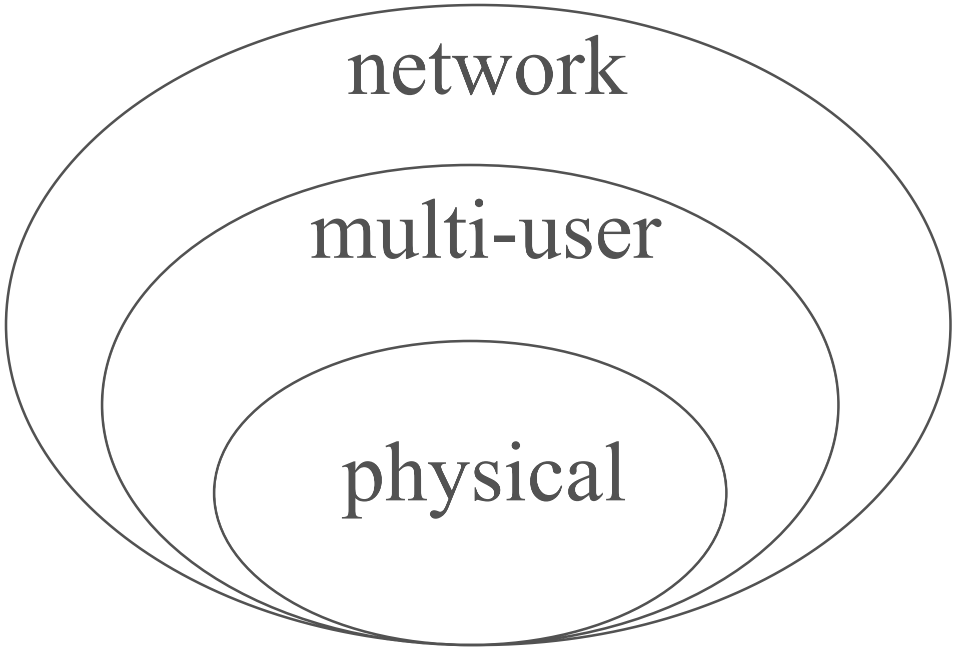
note three kinds of attacks in general from access POV

- attacks **over the network** (e.g., buffer overflow, or password guessing)
- may result NOT in root penetration, but in user account penetration
- which in turn may lead to attempt to take over system from “inside” - **multi-user user attacks**
  - lots of root-exploitation attacks on UNIX possible from access to local system

and physical attacks

- physical easier/quicker than multiuser
  - physical to root
- multiuser easier/quicker than network
  - multiuser to root
- network TOO easy
  - hacker on mars to root
- call this: the inverse-hack law

the closer you are, the easier they fall



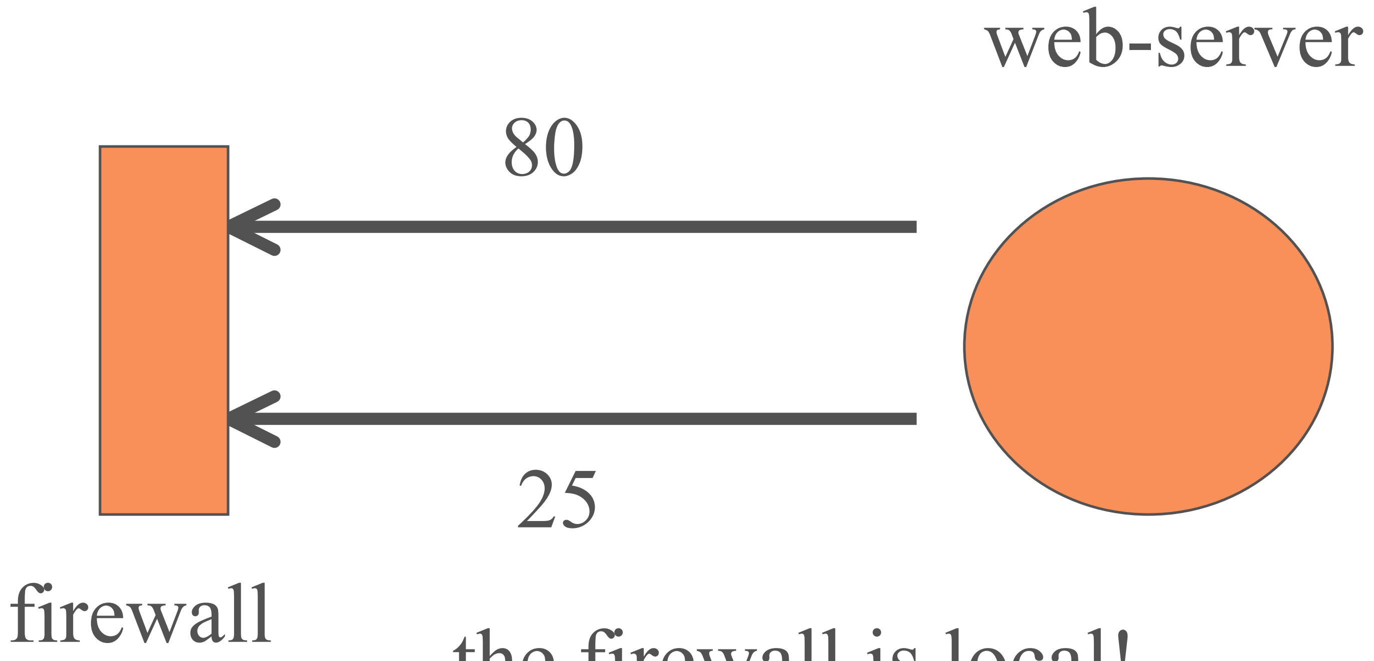
physical access is quicker than multi-user
which is quicker than network

web-server/cgi/classic - phf

- phone book cgi script came with early NCSA apache web servers
- due to bug, could be used to execute any command locally
- e.g., send /etc/passwd away for computational crack attack
- fire up xterm or telnet to get “back-channel” from server out
- solution: remove the script (or all cgi)

consider back-channel hack

telnet to /bin/sh to telnet



the firewall is local!

DOS attacks (denial of service)

- not Disk Operating System Stupid ...:->
- famous ping of death
- winnuke (handout)
- land (handout)
- smurf (handout)
- ip fragmentation attacks (teardrop, etc.)
- note: often one-way and may use ip spoof
- new DDOS attacks, combine some of these
- what next?

other notable attacks of late

- a sea of email or other worm/viruses
- directory traversal exploit
- code-red
- nimda
- blaster/welchia
- sql-slammer

directory traversal exploit - 2001

- signature (execute dir command)
  - http://you.org/scripts..%c1%1c../winnt/system32/cmd.exe?/c+dir
- unicode chars allow you to construct a web pathname to get arbitrary access to server AND EXECUTE COMMANDS
- Usoft security bulletin failed to point out:
  you could copy cmd.exe into remotely executable scripts directory
- Sadmin/IIS worm exploited this
- http://www.cert.org/advisories/**CA-2001-11**.html

code red worm - 2001

- CA-2001-13. buffer overflow in IIS indexing service DLL (ISAPI extensions/irq.dll)
- programmer failed to check input
- Code red exploit used it: CA-2001-19
- aimed at usoft IIS server, port 80, attacker can run arbitrary code on victim machine
- one goal: attack the white-house as DDOS attack
- failed due to mistake: use of hardwired IP as opposed to DNS name

nimda worm - 2001

- CA-2001-26 - usoft nightmare
- attack methods included:

1. client/client via email
2. client/client via usoft “file shares”
3. web-server to client via browsing of compromised files
4. client/web-server via directory traversal exploits
5. client/web-server via code-red and sadmin/IIS exploits

more details

- email - arrives via MIME attachment
  - attachment is auto-executed if clicked on
  - worm resends infected email every 10 days
  - email addresses taken from .htm files or email via MAPI service
- client machines scan for IIS bugs
  - code red and directory traversal
- writes mime-encoded version of self in attempt to infect .html files with javascript enabled code

more

- enabled sharing of c: drive
- creates a guest account
- adds account to administrator group
- creates various trojan binaries that 1st execute the worm
- so just what do you do if infected:
  - format and reinstall ...

counter-measures may include

- block port 69 since worm can use that
- virus scanning
- patching of Usoft systems
- disable javascript (won’t fly)
- do not open that there attachment
  - don’t send attachments ... (dream on)
  - chop off executables in email

W32/blaster/lovsan - 2003

- CA-2003-20
- exploits usoft RPC vulnerability, MS03-026
- DCOM RPC exploit, TCP port 135
  139, 445 also possible acc. to CERT
- post exploit, worm moves msblast.exe to system
- worm may launch TCP SYN denial-of-service attack against usoft site

CERT recommended blocking these ports/services:

- UDP/69 (tftp)
- TCP/UDP 135
- TCP/UDP 139
- TCP/UDP/445
- TCP/593
- TCP/444

W32/welchia/nachi worm - 2003

- targeted same systems vulnerable to W32/blaster (worm/virus)
- performed following actions:
  1. kill/remove blaster worm executable
  2. perform ICMP scanning to find more systems (92 byte ICMP echo)
  3. apply Usoft patch to fix blaster bug
  4. reboot system

so what does it do?

- uses icmp to find real IPs
  - classic incremental scanning
- tries its exploits
- if it succeeds, downloads more code to improve its capabilities
- if it succeeds, kills removes blaster and fixes system
- starts icmp again

welchia cont.

- exploited two vulnerabilities:
  1. RPC vulnerability
  2. IIS server vulnerability
- ports/services used:
  1. TCP/UDP port 135 (Usoft RPC)
  2. Usoft states ports 139/445/598 also possible for RPC
  3. uses TFTP (port 69) or port 707 to move files post break-in

sql-slammer worm - 2003

- exploited vulnerability in usoft SQL server, buffer stack overflow
- called W32/slammer or sapphire worm.
- primarily DOS attack aimed at UDP/1434
- in *one* packet
- sends max streams of UDP packets to semi-random IP destinations
- caused network monitoring/router CPU failures
- worm did not live on disk, resident in memory

counter-measures

- patch system to fix SQL worm
- block UDP port 1434
- ingress/egress filters MAY have some use
- classic case though of lack of IDS signature as worm spread over Internet in minutes
- flash worm - one name for this kind of thing
- non-trivial to eradicate
- TBD: How to Own the Internet paper

network analyzers and ASCII passwords

- assume box X is hacked and bad bart has root access, call this **fan-out attack**
- box X is on traditional network X
- black bart installs tcpdump (or whatever) and starts sniffing for telnet passwords/ftp/pop/http (non-anon) passwords
  - allows bart to attack other systems
  - hacker tools exist for password collection (dsniff)

network analyzers and ASCII passwords

- death of promiscuous mode is not good for bart ... (not guaranteed to be dead though)
  - switch forwarding table attacks
  - end user may have mere hub, so unicast segmentation not available
- network managers should protect their own sniffers and other probes!

ip address authentication

- arp spoofing (broadcast domain function)
  - bart is root on box X (129.1.2.1)
  - bart knows you are user U on box Z, same subnet (129.1.2.2)
  - # ifconfig eth0 129.1.2.2 (i.e., bart's machine claims to be you)
- bart can then setup an account as you on his box (same userid), and rlogin in sans password on some other box with .rhosts

viruses and worms via email or otherwise

- executable code written by others and accidentally executed by you
  - are not a good idea for you
- melissa - one of latest Usoft macro viruses
  - basic program
- java applet security problems
  - interpreted code
- so what would be safe actually?
- how do you know HTML cookies are safe?

all those nasty C programs ...

- “if we didn’t use C, we wouldn’t have those problems ...”
- this explains all the web-based hackery that involves php and javascript (not)
- some bottom lines:
  - don’t do interpreted code
    - you don’t execute “destroy.sh” found on a BBS, do you?
  - don’t mail .doc files ... (ASCII remains unexecutable)
  - for don’t read minimize. Oh yes. adobe and pdf.

tcp sequence # spoofing attacks

- Bellovin and others have pointed out that a TCP session may be hijacked
- e.g., see Phrack article Vol7/Issue48/File14
- **ip spoofing** - use of false ip src address
- **ip splicing** - injection of packets with guessed/correct tcp sequence # into stream in order to attack remote system

sequence-attack paradigm

- assume two hosts, target A, and trusted confederate B (e.g., assume B has ./rhosts access on A
- assume attacking host X
- the attacker on X has root access and wants root access on A
- X must query A to learn a reasonable sequence space
- X then must shut B up say with TCP syn attack or other means

sequence attack paradigm, cont

- X must then simultaneously attack A with a guessed sequence #, (send SYN, SYN-ACK, won't get initial ACK)
- and X must use B's ip address (ip spoof)
- if rlogin/rsh etc., open X may gain root access to A
  - plop down backdoor files
- what are defenses against this?

more attacks

- previous are representative ... but not all
- e.g., DNS attacks known
  - return false binding or overload (add false) binding to reply
  - cause cache to contain (victim DNS, man-in-the-middle IP)
  - **man-in-the-middle** hence possible
  - map site A to competitor ... or what?!

social engineering

- phishing: see:
- http://www.banksafeonline.org.uk/examples/phishing_halifax.html
- see your inbox:
  - hi I’m so/so and I work for OIT and I *need* your password. btw: email is from psu.edu
- http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-746.html
- - spear phishing - targeted phishing
- http://www.microsoft.com/protect/yourself/phishing/spear.mspx

social engineering #2

- zlob/dnschanger and the like
- falls under SE and dns attacks
- user surfs porn
  - told he/she needs new codec
  - gets DNS redirection software
  - visits sites in the Ukraine (or trojan.flush.m)
- motive includes: sell fake av software, collect visa card numbers

foot shooting and general stupidity

- spyware
  - you did install spybot S/D right on that windows box?
  - marketscore spyware (search on that and read)
- information disclosure
  - common for schools to use SSNs to post grades?! (name and SSN is velly bad)
  - visa card numbers
  - passwords
  - intellectual property

google searching

- http://johnny.ihackstuff.com/ghdb.php
  - google dorks
  - information disclosure AND exploited systems AND useful info for targeting AND what else?
  - printers to configure
  - Cisco appliances to configure
  - house control systems - turn the lights on/off at random times?
  - all kinds of sensitive info
- when you put something on the web - be careful

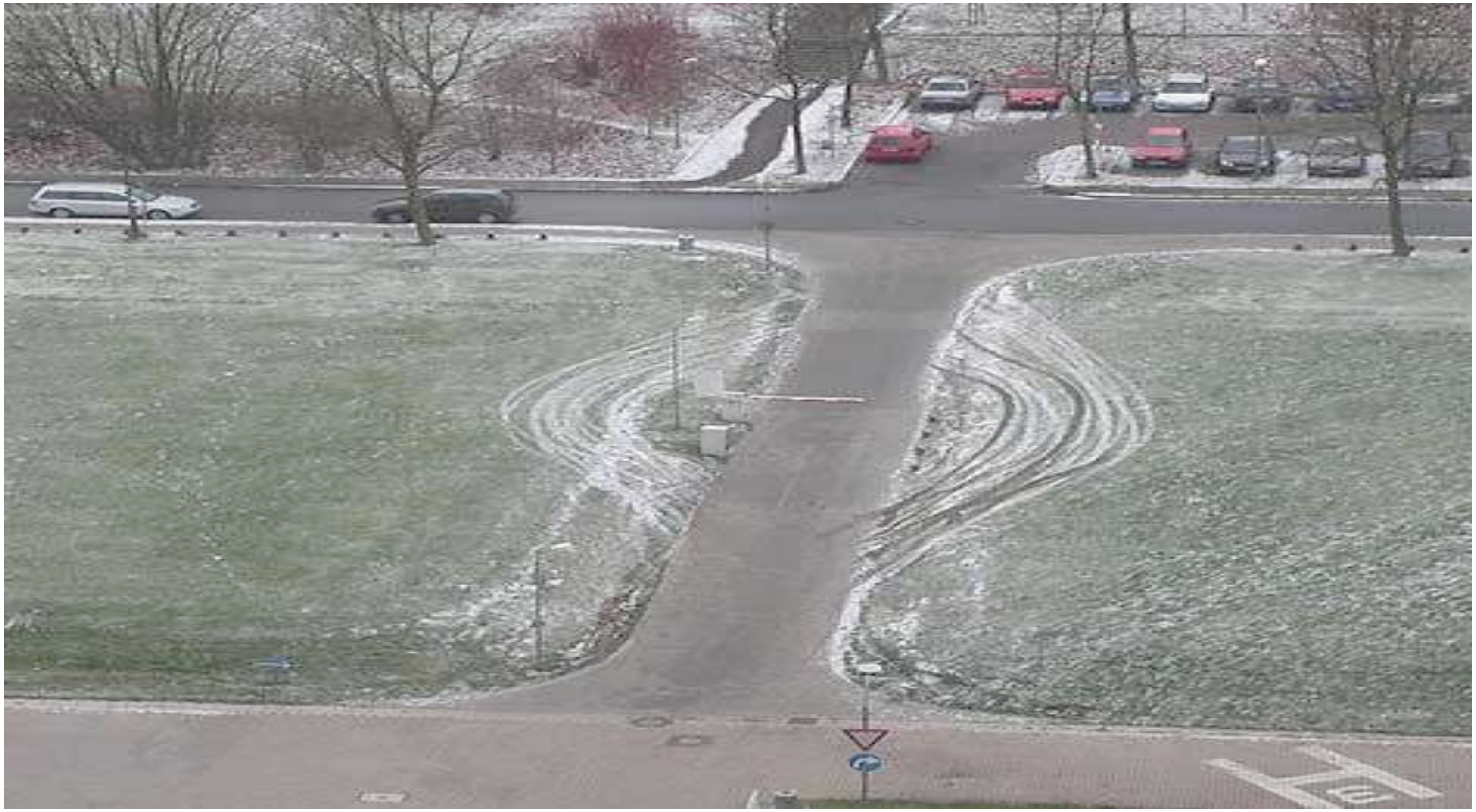
lessons

- hosts and routers exposed to the Internet have to be "patched" or have binaries kept current
- "the fundamental rule/s apply": (partly from Chapman/Zwicky":
  - **least privilege** - don't give it away unless you have to (e.g., daemons should have own UID)
  - **defense in depth** - router/host/protocol
    - » you have a great firewall ... use ssh too, ipfw on unix host

lessons, cont

- **choke point** - force the bad guys to come in the gate and sign in
- **weakest link** - security is as good as the worst security component
  - it’s not just bugs, bugs, bugs, but the worst password in your group, the hole in the firewall ...
  - the ones you don’t know about can kill you
  - every virtual machine MAY have a fatal security flaw you don’t know about (consider halting problem ...)
  - hacker rule: if manual says don’t do X, do X

weakest link - pictorial form



lessons, cont:

- **fail-safe**: if component fails (goes down), it should fail so that it denies access, not permits access (does not **fail-open**)
- **consider default deny vs. default permit** as a rather crucial decision (least privilege again)
- **no exceptions**
  - see weakest link
- **link security systems where possible**
  - show two picture IDs ..., permit IPSEC to bastion host X, or you only use ssh to get to bastion host X

principle of isolation

- put up a wall around it
  - a firewall is a wall
    - consider the “air gap” firewall
    - NAT applies to some extent
  - virtualization may apply
  - a jail
    - ftp readonly directory
    - good old user file permissions (ACLs)
  - tools like packetfence for walled gardens

and

- simplicity - KISS
  - security is often anti-user convenience though
- passwords are basic - improve authentication
- openness (and common sense)
  - it is widely believed that secure crypto algorithms are not secure unless open and publicly reviewed (and subject to test of time)

meta- lesson: fuzz revisited

- consider Microsoft or Sun or anyone who sells only binaries but not open src; i.e.,
- **OPEN CODE IS A MUST**
  - **code for tcp/ip stacks should be available ...**
  - **all security code should be available for review**
- ftp://grilled.cs.wisc.edu/technical_papers/fuzz-revisited.ps.Z
- “An Empirical Study of the Reliability of UNIX Utilities”, Miller, Fredrickson, and So, ACM Communications, Volume 33, issue 12, pp. 32-44

fuzz, and not just lint

- Barton Miller subjected UNIX utilities to random input (“line noise”) and learned:
- too many crashes due to random inputs
  - EOF in middle of input line
- vendor programs had failure rates at 15-43%
- bugs didn’t change from 1990 to 1995
  - although failure ids/test results made available

and here's the clincher

- two lowest failure rates:
- 1. Free Software Foundation's GNU utilities (7%)
  - FSF does not allow fixed-length buffers
- 2. Linux utilities (%9)
  - lots of GNU stuff in here as Stallman is quick to point out
- See Cathedral and Bazaar url (on class home page)

point to ponder

- software will have bugs ... and still be useable
- security takes one and only one (possibly unknown up until now) fatal flaw
  - said flaw needs to be fixed ASAP -
  - not stone-walled
  - not put off until the next release

consider the patch culture for a moment

- here’s a patch
  - 1 million patches and no time
  - what is the correct order of patches?
- you mean you didn’t do the patch?
  - it’s your fault! (feel guilty ...)
- does an auto-net-based update make sense?
  e.g., gentoo/debian getapt
  - src-based vs binary package over the net?
  - FreeBSD cvsup (date) system is interesting
- what are pros/cons of net update?

some apps (OS?) have a bad track record

- usoft IIS/IE
  - compared to apache/mozilla
- outlook
- pine/imapd
- DNS bind before version 9.0 (?)
- sendmail (some say use qmail)
- but is this coding practice/net exposure?
  - your judgement here

a security person needs to stay alert

- monitor web pages or get email alerts
  - one counter-measure is: staying informed
- however this is not easy ...
- how real is the threat?
- what is the precise technical information that I need?
- what does the world REALLY know about attack X?
- how should I change our local policy?

some of many places to note

- Bruce Schneier, counterpane: **www.counterpane.com**
- **www.dshield.org**
- **www.emergingthreats.net**
- **shadowserver.org**
- http://siblog.mcafee.com/portal/
  - trend-micro is good too, there are others
- blog blog blog

commercial AV pages also have info

- symantec: www.symantec.com/avcenter
- f-secure: www.f-secure.com/virus-info
  - and hoax page
- virustotal.com is interesting
  - upload the virus,
  - see if I of M AV products recognize it
  - be depressed at the result?

things to do to improve

- patch
  - turn usoft update on, patch 3rd party apps too
- minimize use of risky apps
  - name one …   gee that’s too easy
- turn it off if you don’t use it
- encrypt it (not always helpful)
- passwords - change them, make them stronger
- contain it (firewalls) - containment is a open problem

staying out of the tar pit #part 2

- virus check it (AV/spyware/HIDS)
- email and web browsing downloads are dangerous
  - be careful
  - think about phishing and targeted phishing
- think about info disclosure
  - search yourself
  - google is also a powerful security info tool

so what did we actually learn?

- it's hopeless?
- what patterns of attacks can one discern?
  - DOS and DDOS/exploits/viruses
  - meta-issues (fan-out/ip address authentication, arp spoofing/sniffing)
- what about counter-measures?

left unvisited

- security policies as a general notion
- host os security
  - prevention of holes in multi-user system
  - logging/auditing (we'll briefly touch on that later)
  - passwords all over again
  - physical host security
- pre and post attack measures
  - backups and forensics