Security introduction

Brian Candler
Presented by
Hervey Allen
Main Security Concerns

Confidentiality
  Keeping our data safe from prying eyes

Integrity
  Protecting our data from loss or unauthorised alteration

Authentication and Authorisation
  Is this person who they claim to be?
  Is this person allowed to do this?

Availability
  Are our systems working when we need them? (Denial of Service, backups, proper configs)
Security Implications of connecting to the Internet

The Internet lets you connect to millions of hosts
  but they can also connect to you!

Many points of access (e.g. telephone, cybercafes, wireless nets, university, work...)
  Even if you can trace an attack to a point on the Internet, the real source may be untraceable
  Many "0wned" machines or "bots" from which further attacks are launched

Your host runs many Internet services
  Many potential points of vulnerability
  Many servers run as "root"! (buffer overflows)
Network-based attacks

Passive attacks
  e.g. packet sniffers, traffic analysis

Active attacks
  e.g. connection hijacking, IP source spoofing, exploitation of weaknesses in IP stack or applications (e.g. Internet Explorer)

Denial of Service attacks
  e.g. synflood

Attacks against the network itself
  e.g. smurf
Other common attacks

Brute-force and Dictionary attacks (password guessing, password too complex)

Viruses

Spyware

Trojan horses

Humans are often the weakest link

"Hi, this is Bob, what's the root password?"

Opening infected E-mails
Authentication: Passwords

Can be guessed

If too complex, users tend to write them down

If sent unencrypted, can be "sniffed" from the network and re-used (pop, imap, telnet, webmail)
Choosing good passwords

Combinations of upper and lower-case letters, numbers and symbols
'brute force' attacker has to try many more combinations

Not in any dictionary, including hackers dictionaries

$40&yc4f
"Money for nothing and your chicks for free"

wsR!vst?
"workshop students aRe not very sleepy today ?"
Authentication: Source IP address

Not verified by the network (since not used in datagram delivery)

Datagrams are easily forged

TCP 3-way handshake gives some degree of protection, as long as you can't guess TCP sequence numbers
- Legitimate example: controlling SMTP relaying by source IP address

Any UDP protocol is completely vulnerable
- e.g. NFS
Authentication: Host name

Very weak

DNS is easily attacked (e.g. by loading false information into cache)

Slight protection by ensuring that reverse and forward DNS matches
  e.g. Connection received from 80.248.72.254
  Lookup 80.248.72.254 -> noc.ws.afnog.org
  Lookup noc.ws.afnog.org -> 80.248.72.254

This is why many sites won't let you connect unless your forward and reverse matches
Cryptographic methods

Can provide REALLY SECURE solutions to authentication, privacy and integrity

Some are hard to implement, many different tools, usually requires special clients

Export and usage restrictions (less of a problem these days)

Take care to understand where the weaknesses lie
Simple combinations

The lock on your front door can be picked

Two locks are better than one

The thief is more likely to try somewhere else
IP source address AND password authentication

Most applications have password authentication, but some also include their own IP-based access controls.

Some applications link to "libwrap" (also known as "tcp wrappers")

/etc/hosts.allow
All services which are started by inetd are covered.

For info and examples: man 5 hosts_access
Most essential steps

Disable all services which are not needed

Apply security patches promptly; join the announcement mailing lists

Good password management

Take special care with 'root' access

Combine passwords with IP access controls where appropriate

Use cryptographic tools where possible
And don't forget these...

Make sure you have current backups!
How else will you recover from a break-in?

Make sure your machine is physically secure!
If someone can walk off with the machine, they can walk off with your data

Log files are valuable!
May want to consider software which watches them,
e.g. swatch, logwatch, logsurfer:
tail -f /var/log/messages
http://www.nsrc.org/security/#logging
More advanced steps

Scan your machines from outside
  nmap, nessus

Firewalls
  apply policy at the network edge
  assert control at a small number of places
  very difficult to build a really GOOD firewall of your own
  not effective if your own users violate security (by downloading viruses, for example)

Intrusion Detection Systems (IDS)

Token-based authentication
UNDERSTAND what you're doing

A bad security solution is worse than no security at all

Know what you're doing
  Read all the documentation
  Read sample configurations
  Build test machines
  Ask questions
  Join the announcements mailing list for your O/S and applications

Test what you've done
  Try connecting from outside your network
  Try circumventing your own rules
Some helpful guides

The FreeBSD handbook at www.freebsd.org
  Chapter 14 on security

"Practical Unix & Internet Security" (O'Reilly)

http://nsrc.org/security/

Security alert mailing lists, including:
  http://www.securityfocus.com/ ("Bugtraq")
  http://www.cert.org/
  http://www.rootshell.com/