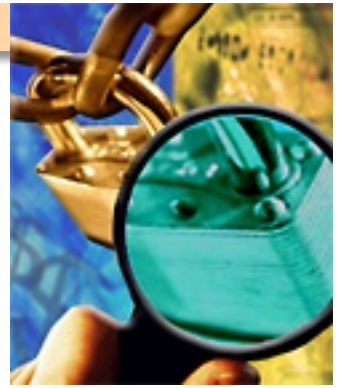
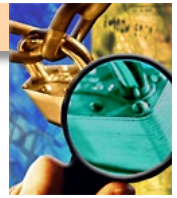


Vulnerability Assessment Tools

Vulnerability Assessment Course

All materials are licensed under a Creative Commons “Share Alike” license.



- **http://creativecommons.org/licenses/by-sa/3.0/**

You are free:

to Share — to copy, distribute and transmit the work

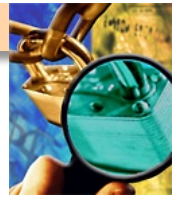
to Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

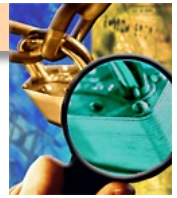
Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

Agenda

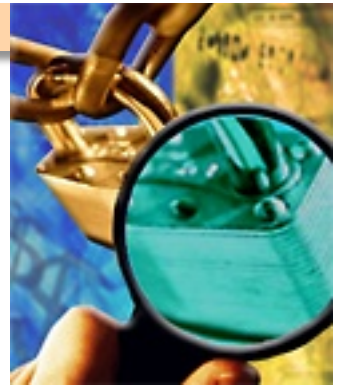


- Introduction to Backtrack Linux
  - Lab
- Monitoring Network Traffic
  - Tcpdump: lab
  - Wireshark: lab
- Host and service enumeration
  - Nmap: lab
- Vulnerability scanning
  - Nessus: lab

Disclaimer

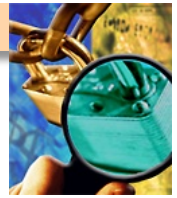


- The tools herein should be “safe” when used properly
  - But unsafe under certain conditions
  - Even a “safe mode” vulnerability scanner can crash a host or firewall
- Do not try on any system or network without approval!
- If you want to practice any techniques
  - Do it at home at your own risk!
  - Using resources that don’t belong to someone else
  - YOU HAVE BEEN WARNED!



Vuln-Assessment Tools: Getting Started

Initiating Lab Setup



- Your lab workstation is running VMware on Windows
- This is so you can run and use multiple different "machines"
- One of your lab workstation VM images is BackTrack Linux
- One of your lab workstation VM images is Windows XP
- One of your lab workstation VM images is Solaris (UNIX)
- Crank up your lab workstation, start the BackTrack image

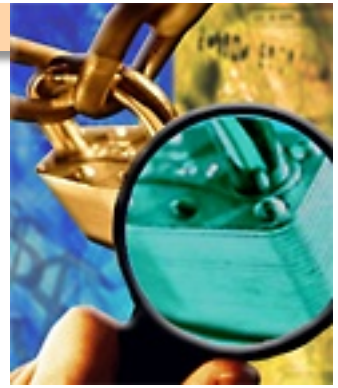
Testing Backtrack

- Did BackTrack Linux launch okay?
- Do you have a UNIX command prompt?
- Do you see an XWindows graphic (windows) interface?
- When you type “ifconfig –a”, do you have an IP address?
- Try pinging your own IP address (“ping x.x.x.x”)
- Then try pinging a neighbor’s IP address “ping y.y.y.y”)

DO *NOT* SCAN OR ATTACK OTHER LAB MACHINES UNLESS TOLD TO DO SO

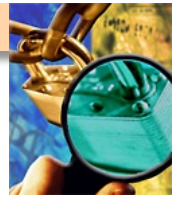
Backtrack Lab

- Play around with the BackTrack Linux interface + windows
- Try clicking bottom left corner (like Windows “Start” Menu)
- Navigate through the various menus and utilities
- Pay particular attention to the “BackTrack Utilities” menu
- If you want to experiment with a utility, now is a good time
- BREAK TIME – keep playing around or take a breather



Sniffing Traffic with tcpdump

Introduction



- Tcpdump is the de facto tool for recording network traffic
- Available on Linux, BSD, other *nixes
  - Solaris (Sun UNIX) uses both ‘tcpdump’ and ‘snoop’ (older)
- A “packet sniffer” collects/analyze packets from a network
- Generally works by putting NIC in “promiscuous mode”
  - Typically my network-card ignores traffic not meant for me
  - But in ‘promiscuous mode’ whatever I see I collect/record
  - IMPORTANT DISTINCTION HERE: “hub” versus “switch”

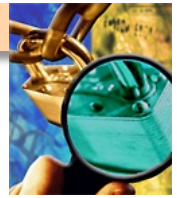
    [will impact your sniffing/traffic-collection results]
  - Several different formats for recorded traffic data

    [.pcap (“packet-capture”) format seems to be universal]

Modes Of Operation

- Packet Logging Mode
  - Writing packets to a file on disk (or a Unix pipe)
  - Good for record-keeping, good for later analysis
- Traffic Analysis Mode
  - Header analysis
    - Displays many details of each packet's header, layers 1-4
  - Full packet capture analysis
    - Also records packet contents, payload, application-type, flags, etc.
  - Generally used for diagnosing network issues
- Recording your OWN assessment traffic = Good Idea
  - Sort of a chain-of-custody issue…could keep you out of jail
  - Write all network traffic to a file (see following slides on how)

Tcpdump -- Options



- By default tcpdump captures the first 68 bytes of a packet
  - This can be adjusted…"tcpdump –s 25" "tcpdump –s 128"
  - "tcpdump –s 0" will capture the *entire* packet (all bytes)
- Reverse DNS-resolution (IP → name) can be slow/stressful
  - "tcpdump –n" will list IP addresses only, not resolve names (faster)
- You can view tcpdump packet data in hexadecimal or ASCII
  - "tcpdump –X" will let you see the data both ways
- If you have multiple NICs, tell tcpdump which to sniff on
  - "tcpdump –i eth0" is one example [will listen to eth0]

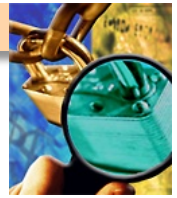
Capture Filters

- This is similar to a search filter… “only what I want to sniff”

- Can filter various packet flags, packet fields
- Can filter “only show traffic from specific IP addresses”
- Can filter “only show traffic going to specific IP addresses”
- Can filter “only show traffic from specific MAC addresses”
- Can filter “only show traffic which contains the number 15”
- Can filter “only show port 80 traffic”
- Can filter “only show HTTP protocol traffic”

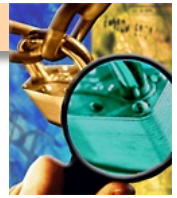
- Can combine these with AND, OR
- These filters almost get to be like programming languages

Capture Filters – Examples



- Addresses
  - host 10.10.10.1
  - src host 192.168.1.1
  - dst host zeus
  - ether src AA:BB:CC:DD:EE:FF
  - dst net 192.168.0.0
- Port Numbers
  - port 22
  - tcp dst port 8080

Capture Filters – More Examples



- Operators
  - dst host 10.10.10.1 and not tcp port 22
  - host bilbo and (cheiron or nettos)
- Protocol keywords
  - TCP flags: tcp-syn, tcp-ack, tcp-fin, etc.
  - ICMP: icmp-echoreply, icmp-unreach
  - Used as an offset
    - tcp[tcpflags] & (tcp-syn|tcp-fin) != 0
    - icmp[icmptype] = icmp-echoreply

Interpreting Tcpdump Output

```
15:39:05.435985 < nettos.1264 > zeus.ftp: S 2138865536:2138865536(0) win
65535 <mss 1460,nop,nop,sackOK> (DF)
15:39:05.511620 < zeus.ftp > nettos.1264: S 4198232748:4198232748(0) ack
2138865537 win 5840 <mss 1460,nop,nop,sackOK> (DF)
15:39:05.511632 < nettos.1264 > zeus.ftp: . 1:1(0) ack 1 win 65535 (DF)
15:39:05.588085 < zeus.ftp > nettos.1264: P 1:62(61) ack 1 win 5840 (DF)
15:39:05.728369 < nettos.1264 > zeus.ftp: . 1:1(0) ack 62 win 65474 (DF)
```

- Connection from nettos to zeus
- Three way handshake during first three lines
  - Note that tcpdump displays *relative* sequence and ack numbers
- Followed by some data transfer
- Note the Don’t Fragment (DF) bit is set
- Full packet capture similar, just provides application data

tcpdump Usage

- Typical uses
  - Diagnosing problems on a network
  - Capturing packets for later analysis
  - Keeping a record of network activity during assessments

    [your own activity *and* other stuff you see on the network]

- Caveats
  - Don’t over-filter at first…you might miss something

    [start broad, then filter in narrow, don’t exclude stuff like ICMP/ping]
  - Remember your own SSH connection may pollute the tcpdump

    [ex: you ssh in, then tcpdump, it shows thousands of port 22]

    [prevent this by using ‘not (tcp port 22 and host <my_own_ip>)’]

Lab

- Create an account on the demo phpbb server and post the decoded tcdpump output showing your password.

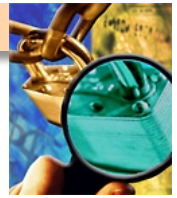
- Hints
  - Use filters like port (possibly 80?), protocol (http?)
  - -X
  - Oh you'll need more then the default number of bytes

- Bonus point
  - Capture your neighbors password :-)

Pointers

- http://www.tcpdump.org/
- http://netgroup-serv.polito.it/winpcap/
- http://www.robertgraham.com/pubs/sniffing-faq.html

Questions

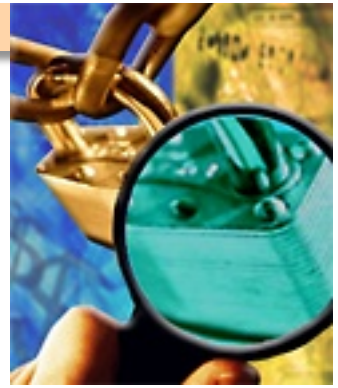


- Common tcpdump usage during a assessment:
  - tcpdump –w outfile.cap –s 1580
  - Consider running your own tcpdump in a background window

    [might show interesting things in later analysis]

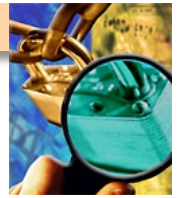
    [might also serve as valuable evidence if you are accused later]

- ???

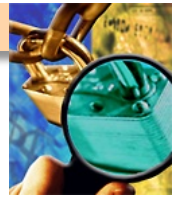


Analyzing Traffic with Wireshark

Welcome to WireShark



- Used to be called Ethereal (two products have diverged)
- WireShark is a sniffer *and* network protocol analyzer
  - Handles just about every application you've heard of
  - List at http://www.wireshark.org/faq.html#q1.10
- Essentially "the Windows tcpdump" with a useful GUI
  - WireShark runs on Windows, Linux, BSD, all sorts of platforms
  - You may find valuable vulnerabilities w. WireShark traffic alone
- Allows for capture-filtering, using text expressions or GUI
- Like tcpdump, can sniff data "live" or using a capture file
- May have trouble handling very large packet-capture files



WireShark – Sample Demo

sendmail-exim.cap - Ethereal

File Edit Capture Display Tools Help

No.	Time	Source	Destination	Length	Protocol	Info
1	0.000000	nirvana.ramirez	10.255.255.255	174	WHO	nirvana: 0.04 0.28 0.33
2	180.000020	nirvana.ramirez	10.255.255.255	174	WHO	nirvana: 0.28 0.26 0.29
3	228.166341	00:40:33:d9:7c:fd	Broadcast	42	ARP	Who has 10.0.0.254? Tell 10.0.0.6
4	228.168293	00:00:39:cf:d9:cd	00:40:33:d9:7c:fd	60	ARP	10.0.0.254 is at 00:00:39:cf:d9:cd
5	228.168383	nirvana.ramirez	dns1-rcs.rcsntx.swbel	75	DNS	Standard query AAAA mail.swbell.net
6	233.169573	nirvana.ramirez	dns1-rcs.rcsntx.swbel	75	DNS	Standard query AAAA mail.swbell.net
7	243.179411	nirvana.ramirez	dns1-rcs.rcsntx.swbel	75	DNS	Standard query AAAA mail.swbell.net
8	243.265951	dns1-rcs.rcsntx.swbel	nirvana.ramirez	136	DNS	Standard query response
9	243.266924	nirvana.ramirez	dns1-rcs.rcsntx.swbel	75	DNS	Standard query A mail.swbell.net
10	243.363886	dns1-rcs.rcsntx.swbel	nirvana.ramirez	217	DNS	Standard query response A 151.164.30.28 A 151.1
11	243.372890	nirvana.ramirez	mta4.rcsntx.swbell.ne	74	TCP	1363 > smtp [SYN] Seq=4217165042 Ack=0 Win=3185
12	243.453093	mta4.rcsntx.swbell.ne	nirvana.ramirez	74	TCP	smtp > 1363 [SYN, ACK] Seq=1963522946 Ack=421716
13	243.453255	nirvana.ramirez	mta4.rcsntx.swbell.ne	66	TCP	1363 > smtp [ACK] Seq=4217165043 Ack=1963522947
14	243.544546	mta4.rcsntx.swbell.ne	nirvana.ramirez	166	TCP	smtp > 1363 [PSH, ACK] Seq=1963522947 Ack=421716
15	243.544671	nirvana.ramirez	mta4.rcsntx.swbell.ne	66	TCP	1363 > smtp [ACK] Seq=4217165043 Ack=1963523047
16	243.545488	nirvana.ramirez	mta4.rcsntx.swbell.ne	88	TCP	1363 > smtp [PSH, ACK] Seq=4217165043 Ack=19635
17	243.643053	mta4.rcsntx.swbell.ne	nirvana.ramirez	66	TCP	smtp > 1363 [ACK] Seq=1963523047 Ack=4217165065
18	243.705096	mta4.rcsntx.				

- Frame 1 (174 on wire, 174 ca
- Ethernet II
- Internet Protocol
- User Datagram Protocol
  - Source port: who (513)
  - Destination port: who (51
  - Length: 140
  - Checksum: 0xa301
- Who
  - Version: 1
  - Type: 1
  - Send Time: Dec 5, 1999 0
  - Receive Time: Dec 31, 196
  - Hostname: nirvana
  - Load Average Over Past 5
  - Load Average Over Past 10
  - Load Average Over Past 15
  - Boot Time: Dec 4, 1999 1
  - Who utmp Entry
  - Who utmp Entry
  - Who utmp Entry

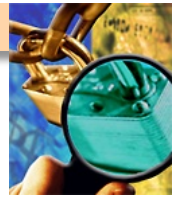
```
0000  ff ff ff ff ff ff 00 40
0010  00 a0 28 44 00 00 40 11
0020  ff ff 02 01 02 01 00 84
0030  03 c6 00 00 00 00 6e 65
0040  00 00 00 00 00 00 00 00
0050  00 00 00 00 00 00 00 00
0060  00 21 38 49 47 d9 3a 30
0070  61 6d 00 00 00 00 38 45
0080  73 2f 30 00 00 00 67 73
```

Filter:

gram@nirvana:~/prj/ethereal

```
$ ./tethereal -r ~/prj/sniff/sendmail-exim.cap |more
  1   0.000000 nirvana.ramirez -> 10.255.255.255 WHO nirvana: 0.04 0.28 0.33
  2 180.000020 nirvana.ramirez -> 10.255.255.255 WHO nirvana: 0.28 0.26 0.29
  3 228.166341 00:40:33:d9:7c:fd -> Broadcast    ARP Who has 10.0.0.254?  Tell 10.0.0.6
  4 228.168293 00:00:39:cf:d9:cd -> 00:40:33:d9:7c:fd ARP 10.0.0.254 is at 00:00:39:cf:d9:cd
  5 228.168383 nirvana.ramirez -> dns1-rcs.rcsntx.swbell.net DNS Standard query AAAA mail.swbell.net
  6 233.169573 nirvana.ramirez -> dns1-rcs.rcsntx.swbell.net DNS Standard query AAAA mail.swbell.net
  7 243.179411 nirvana.ramirez -> dns1-rcs.rcsntx.swbell.net DNS Standard query AAAA mail.swbell.net
  8 243.265951 dns1-rcs.rcsntx.swbell.net -> nirvana.ramirez DNS Standard query response
  9 243.266924 nirvana.ramirez -> dns1-rcs.rcsntx.swbell.net DNS Standard query A mail.swbell.net
 10 243.363886 dns1-rcs.rcsntx.swbell.net -> nirvana.ramirez DNS Standard query response A 151.164.30.28 A 151.164
.30.25 A 151.164.30.26 A 151.164.30.27
 11 243.372890 nirvana.ramirez -> mta4.rcsntx.swbell.net TCP 1363 > smtp [SYN] Seq=4217165042 Ack=0 Win=31856
 12 243.453093 mta4.rcsntx.swbell.net -> nirvana.ramirez TCP smtp > 1363 [SYN, ACK] Seq=1963522946 Ack=4217165043
Win=10136
 13 243.453255 nirvana.ramirez -> mta4.rcsntx.swbell.net TCP 1363 > smtp [ACK] Seq=4217165043 Ack=1963522947 Win=3
2160
 14 243.544546 mta4.rcsntx.swbell.net -> nirvana.ramirez TCP smtp > 1363 [PSH, ACK] Seq=1963522947 Ack=4217165043
Win=10136
 15 243.544671 nirvana.ramirez -> mta4.rcsntx.swbell.net TCP 1363 > smtp [ACK] Seq=4217165043 Ack=1963523047 Win=3
2160
 16 243.545488 nirvana.ramirez -> mta4.rcsntx.swbell.net TCP 1363 > smtp [PSH, ACK] Seq=4217165043 Ack=1963523047
Win=32696
 17 243.643053 mta4.rcsntx.swbell.net -> nirvana.ramirez TCP smtp > 1363 [ACK] Seq=1963523047 Ack=4217165065 Win=1
0136
 18 243.705096 mta4.rcsntx.swbell.net -> nirvana.ramirez TCP smtp > 1363 [PSH, ACK] Seq=1963523047 Ack=4217165065
Win=10136
 19 243.705595 nirvana.ramirez -> mta4.rcsntx.swbell.net TCP 1363 > smtp [PSH, ACK] Seq=4217165065 Ack=1963523260
Win=32696
 20 243.800146 mta4.rcsntx.swbell.net -> nirvana.ramirez TCP smtp > 1363 [PSH, ACK] Seq=1963523260 Ack=4217165104
Win=10136
 21 243.800552 nirvana.ramirez -> mta4.rcsntx.swbell.net TCP 1363 > smtp [PSH, ACK] Seq=4217165104 Ack=1963523295
Win=32696
 22 243.925938 mta4.rcsntx.swbell.net -> nirvana.ramirez TCP smtp > 1363 [ACK] Seq=1963523295 Ack=4217165140 Win=1
0136
 23 247.285127 mta4.rcsntx.swbell.net -> nirvana.ramirez TCP smtp > 1363 [PSH, ACK] Seq=1963523295 Ack=4217165140
Win=10136
 24 247.285572 nirvana.ramirez -> mta4.rcsntx.swbell.net TCP 1363 > smtp [PSH, ACK] Seq=4217165140 Ack=1963523335
Win=32696
--More--
```

Wireshark Demo



- Start up your lab-workstation BackTrack Linux image
- Find WireShark in your application menus, launch it

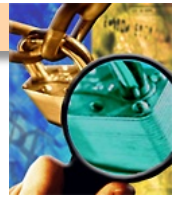
- Click “Start Capture” or “Capture on Interface <whatever>”

- Generate your own traffic by pinging other workstations
- Generate your own traffic by web-surfing
- Generate your own traffic by getting files from FTP server

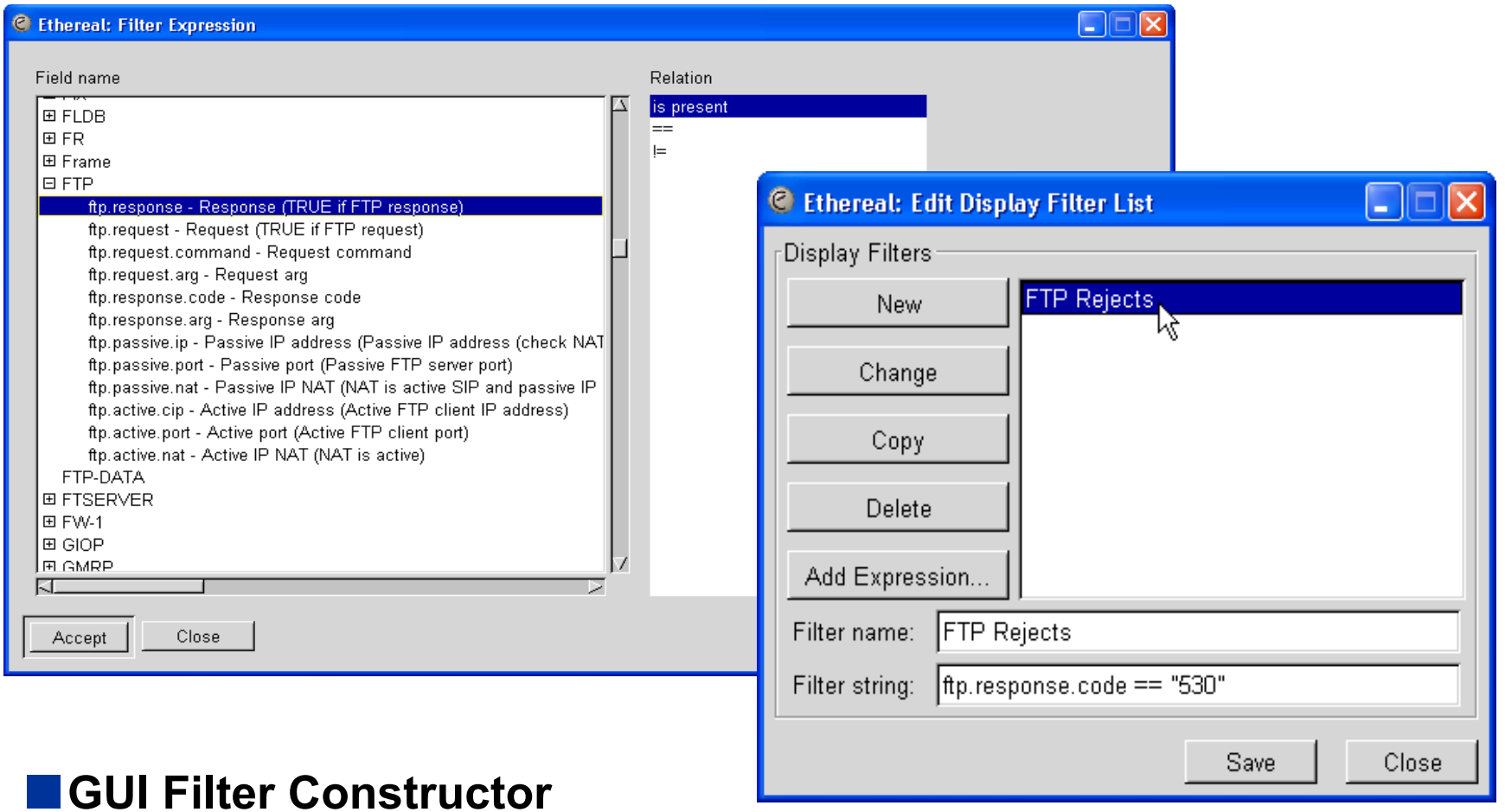
- Stop the capture and see if/how WireShark saw your traffic

WireShark – Display Filters

- Boolean syntax
- Supported data types
  - Numeric (integer, float)
  - String
  - Boolean
  - Various addresses (Ethernet, IP, IPX, etc.)
- Nearly every tree view is filterable
- Similar to the ‘capture filters’ from earlier, but can do more

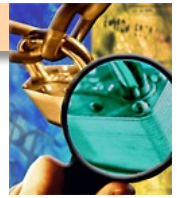


WireShark – Display Filters (cont.)

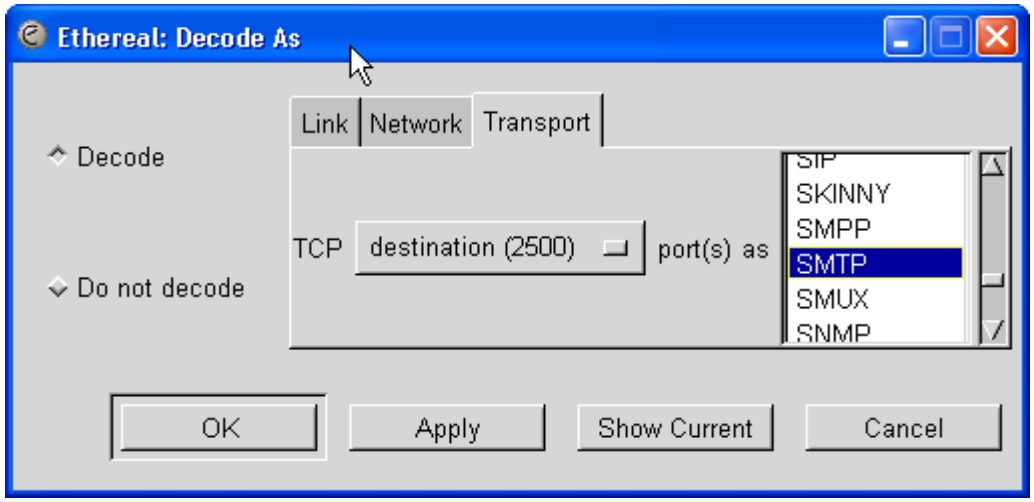


- GUI Filter Constructor
- Filter string can also be crafted by hand
- See how WireShark knows the FTP request structure (above)?

WireShark – Decoding Protocols

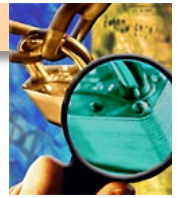


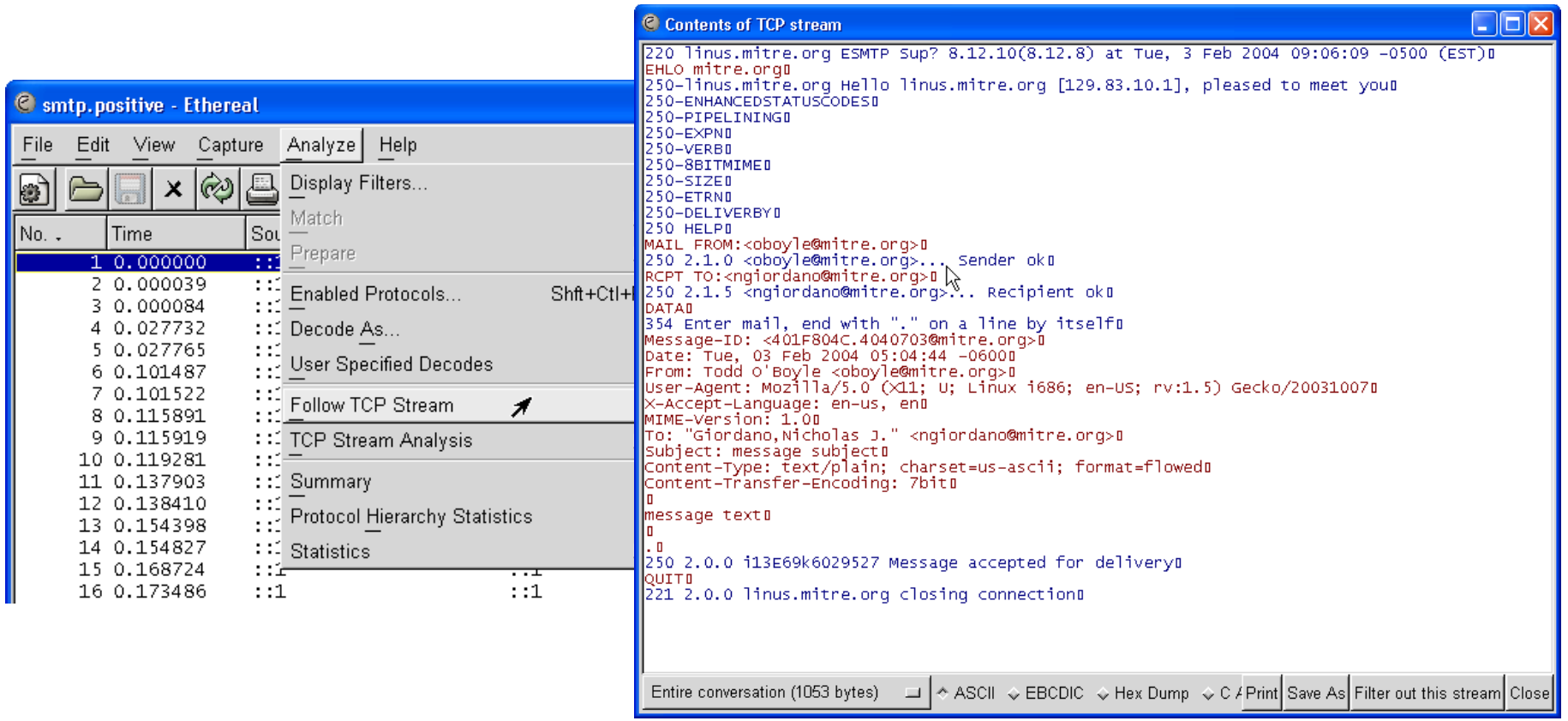
```
 3 0.000084    ::1                ::1                TCP      32793 > 2500 [ACK] Seq=1 Ack=1 Win=32752 Len=0 TSV=3340693 TSER=3340692
 4 0.027732    ::1                ::1                TCP      2500 > 32793 [PSH, ACK] Seq=1 Ack=1 Win=32728 Len=88 TSV=3340720 TSER=3340693
 5 0.027765    ::1                ::1                TCP      32793 > 2500 [ACK] Seq=1 Ack=89 Win=32752 Len=0 TSV=3340720 TSER=3340720
 6 0.101487    ::1                ::1                TCP      32793 > 2500 [PSH, ACK] Seq=1 Ack=89 Win=32752 Len=16 TSV=3340794 TSER=3340720
 7 0.101522    ::1                ::1                TCP      2500 > 32793 [ACK] Seq=89 Ack=17 Win=32728 Len=0 TSV=3340794 TSER=3340794
 8 0.115891    ::1                ::1                TCP      2500 > 32793 [PSH, ACK] Seq=89 Ack=17 Win=32728 Len=198 TSV=3340808 TSER=3340794
 9 0.115919    ::1                ::1                TCP      32793 > 2500 [ACK] Seq=17 Ack=287 Win=32752 Len=0 TSV=3340808 TSER=3340808
10 0.119281    ::1                ::1                TCP      32793 > 2500 [PSH, ACK] Seq=17 Ack=287 Win=32752 Len=30 TSV=3340812 TSER=3340808
11 0.137903    ::1                ::1                TCP      2500 > 32793 [PSH, ACK] Seq=287 Ack=47 Win=32728 Len=43 TSV=3340830 TSER=3340812
```




```
 4 0.027732    ::1                ::1                SMTP     Response: 220 linus.mitre.org ESMTP Sup? 8.12.10(8.12.8) at Tue, 3 Feb 2004 09:06:09 -0500 (EST)
 5 0.027765    ::1                ::1                TCP      32793 > 2500 [ACK] Seq=1 Ack=89 Win=32752 Len=0 TSV=3340720 TSER=3340720
 6 0.101487    ::1                ::1                SMTP     Command: EHLO mitre.org
 7 0.101522    ::1                ::1                TCP      2500 > 32793 [ACK] Seq=89 Ack=17 Win=32728 Len=0 TSV=3340794 TSER=3340794
 8 0.115891    ::1                ::1                SMTP     Response: 250-linus.mitre.org Hello linus.mitre.org [129.83.10.1], pleased to meet you
 9 0.115919    ::1                ::1                TCP      32793 > 2500 [ACK] Seq=17 Ack=287 Win=32752 Len=0 TSV=3340808 TSER=3340808
10 0.119281    ::1                ::1                SMTP     Command: MAIL FROM:<oboyle@mitre.org>
11 0.137903    ::1                ::1                SMTP     Response: 250 2.1.0 <oboyle@mitre.org>... Sender ok
```

WireShark – Stream Re-Assembly Plugin





- Can rebuild host-to-host conversation from multiple packets
- Can be viewed in both ASCII and Hex
- Good for analyzing FTP, IM chat, SMTP mail, TELNET, any cleartext

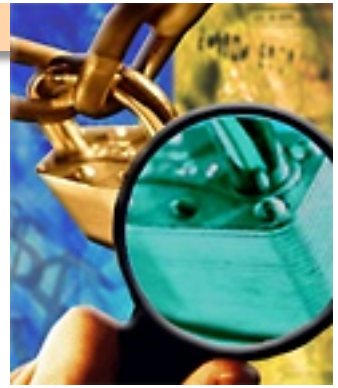
Pointers

- http://www.wireshark.org/
- http://netgroup-serv.polito.it/winpcap/
- http://www.robertgraham.com/pubs/sniffing-faq.html

Break!

- Keep playing with WireShark if you like
- Ask me whatever questions you like

- Go take a breather when you're done, or to clear your head



Vuln-Assessment: Enumeration, Reconnaissance and Scanning

Overview

- Purpose
  - Why to scan? What to look for? What to ‘enumerate’ ?
  - What mindset to use while scanning, probing, investigating?
  - What the does he mean by ‘enumerate’, anyway?
  - Gain some [brief] exposure to scanning tools, such as NMAP
- Format
  - Discussion of network mapping and surrounding issues
  - Lecture and demonstration of Nmap, w. specific techniques
  - Hands-on lab with instructor supervision

Network Mapping

- Definition
  - Collect information on a target network-address range
  - Learn about (document) visible hosts, devices, protocols
  - Accurately represent, understand the target for future reference

- When we're done, we should have:
  - Host IP addresses and MACs of all targets within scope
  - Operating system versions
  - Ports, protocols, and (usually) service version information

- DO **NOT** BLINDLY TRUST YOUR SCAN RESULTS
  - These tools are notorious for false positives
  - What you think you're seeing may not be reality
  - Try to corroborate with network maps, host-internal outputs
    [ifconfig –a, netstat –a, netstat –an, netstat –rn, other…]

Methodologies

- Rumor (unreliable)
- Informed Estimate, Interviews (nice start, but incomplete)
- Physical Inventory (thorough, but time consuming)

- Automated Discovery Tools (hopefully reliable, fast(er))
  - Use a network-mapping tool from one point on the network
  - Repeat from another point on the network if you need 2+ views
  - Consolidate that data for future reference, attack-planning
  - A ‘realistic’ map may require different maps from diff. points

- DO **NOT** BLINDLY TRUST DESIGN DOCS OR MAPS
  - They age (become out-of-date), may even omit intentionally
  - Build, plan and report from your own scans + enumeration

Scanning – a Typical Approach

Inform appropriate authorities "I am about to scan XYZ"

[Scanning can be seen as an 'act of war', may sound alerts]

From **OUTSIDE the firewall ("outsider's attack view")**

- **Scan the target area for running hosts with ping and TCP scans**
- **Once you have a list of "live" hosts, port-scan all those hosts**
  a) **Document hosts that 'should be there' but didn't show up as live**
  b) **Consider doing TCP scans, UDP scans *and* RPC scans on the hosts**
  c) **Re-scan any expected hosts or services that didn't show up as 'live'**

Repeat this scan-process from **INSIDE the firewall**

[form a composite picture, inside-view *and* outside-view]

Unexpected hosts/services are most interesting – pursue them

Forming a Picture – Scanning + Analysis

- What parts of the network are visible to an outsider?
  - Which of these were expected to be externally-visible?
- What parts of the network are visible to an insider?
  - Which of these were unexpected, not seen in any docs/maps?

- What parts of the network are "highest-value" targets?
- Which of the targets most need protection?
- Which ports/protocols/services are now exposed/visible?

- Compare what-should-be against what-you-actually-see
- Very common to find great discrepancies here
- Very common to find more ports open than "should be"

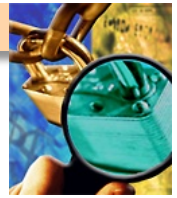
- Sometimes you will even discover information-leakage

Common Network/Port Scanners

- Cheops-NG
- IP Sonar
- Scanrand
  - Very fast, hard to get working
- SolarWinds
- Visio Enterprise Edition
  - Expensive, not a good fit (for inventory, not vuln-assessment)

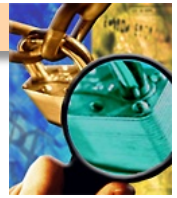
- ISS (Internet Security Scanner)
  - Now IBM-owned, also called “IS”, “ES”, “Proventia”
- eEye Retina
  - Now McAfee-owned, McAfee also owns Foundstone utilities

- NMAP



SolarWinds

- Commercial set of utilities for network discovery
- Runs on Windows
- Point & click interface
- Easy to use
- Powerful discovery tools (IP Network Browser, SNMP Sweep, MAC Address Discovery, Ping Sweep)
- Website: http://www.solarwinds.net
- Cost: $695 Professional Edition (30 Day FREE Evaluation)

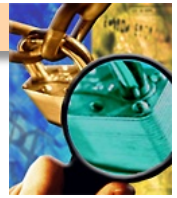


NMAP

- The “classic Mustang muscle-car” of port-scanners
- Runs on just about everything

- Purists use a command-line (text) interface
- Various GUI point-and-click front ends are also available

- MANY scan options (TCP, UDP, SYN-scan, version-probe)
- Widespread use + support throughout the security world
- Website: http://www.insecure.org/nmap

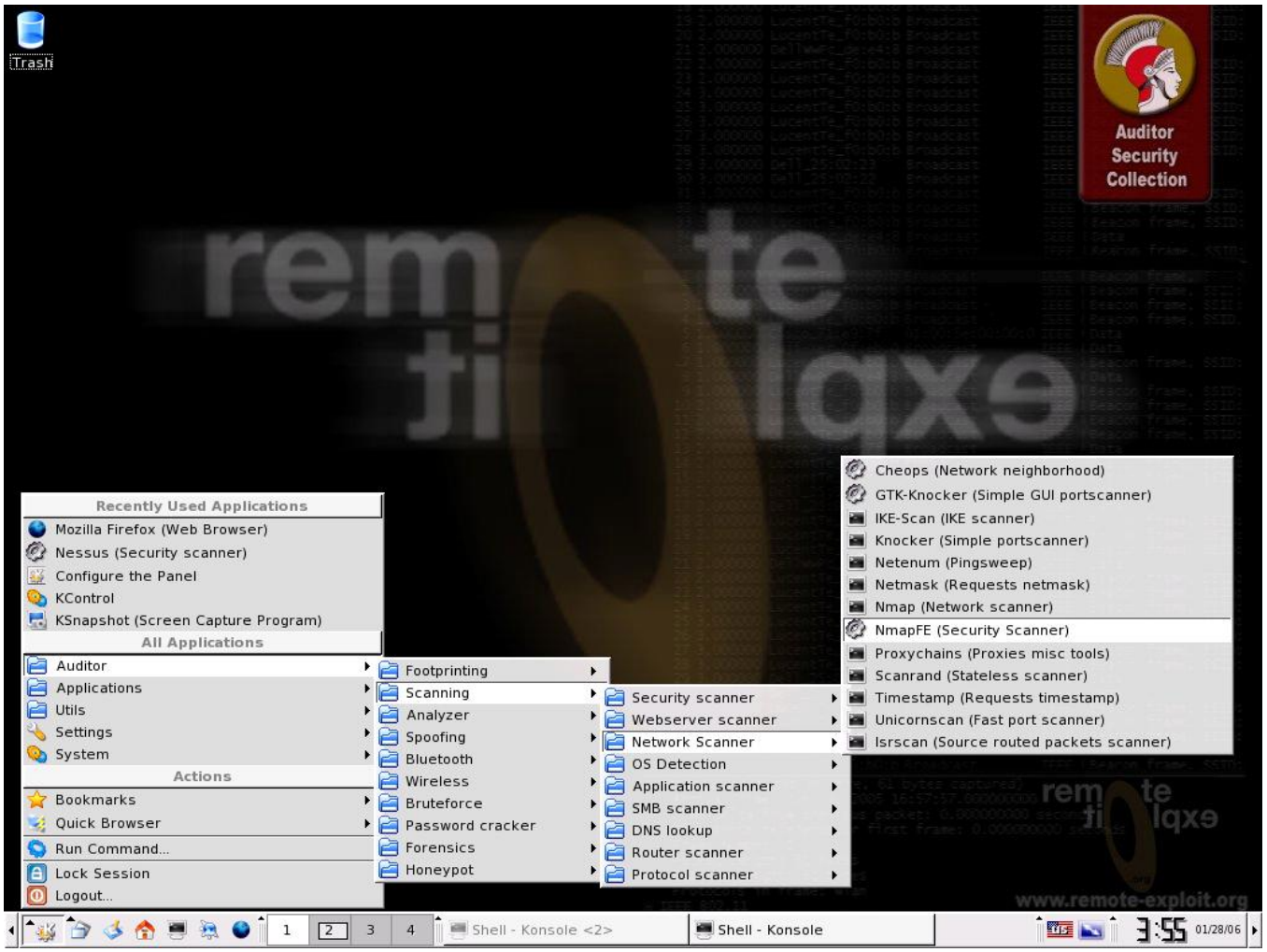


NMAP – Just a Few Sample Port-Scan Options

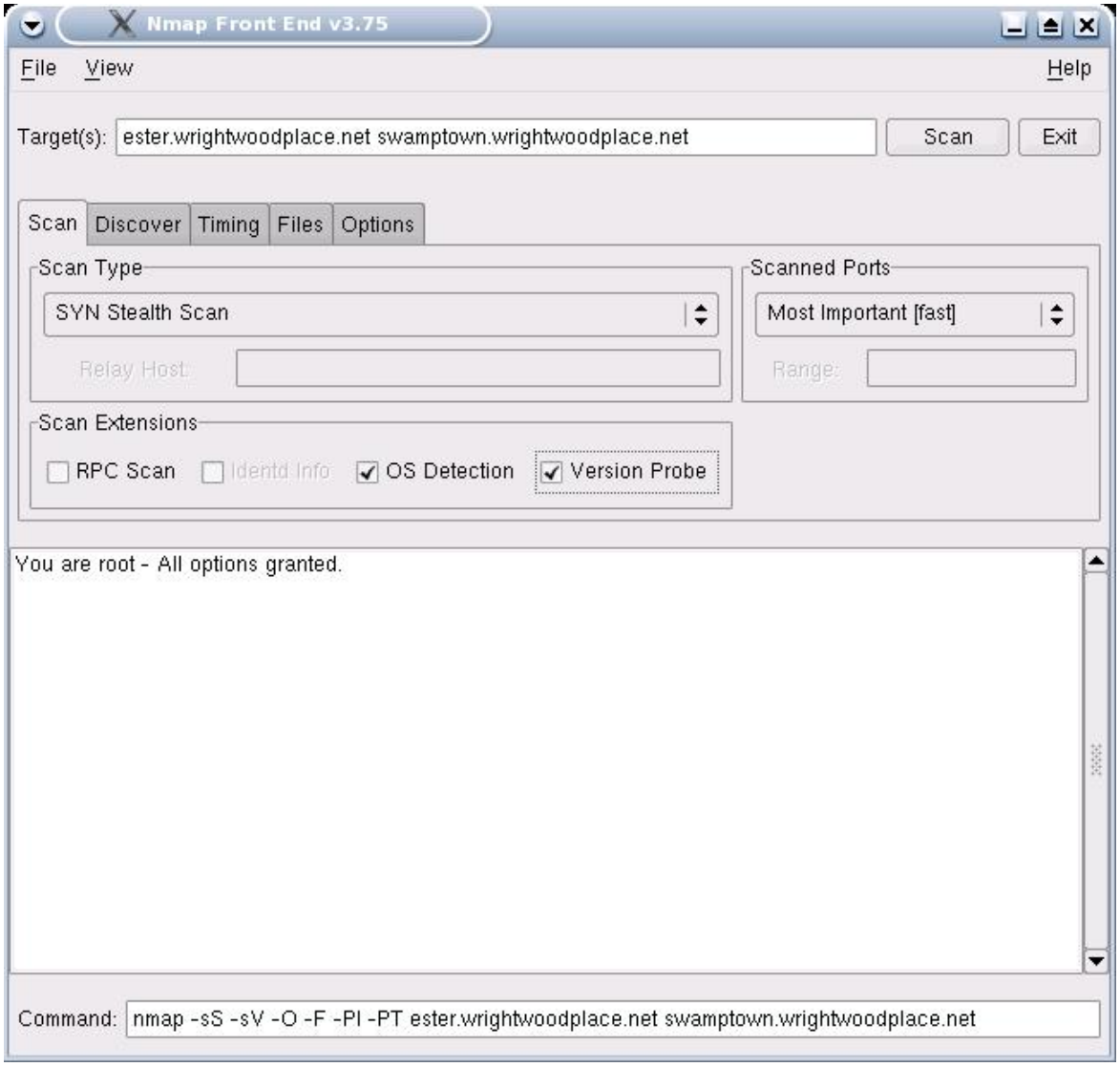
Vanilla TCP connect() scanning **	**[–sT]**
TCP SYN (half open) scanning **	**[–sS]**
TCP FIN (stealth) scanning	**[– sF]**
TCP ftp proxy (bounce attack) scan	**[-b <ftp relay>]**
SYN/FIN scan using IP fragments	**[–f]**
UDP scanning **	**[–sU]**
UDP raw ICMP port unreachable scanning	**[–sO]**
ICMP scanning (ping-sweep) **	**[–sP]**
Try-to-figure-out-version scanning **	**[–sV]**
Do not try to resolve DNS names (this saves time) ******	**[–n]**
Verbose mode (lots of extra debug-like output) ******	**[–v] [–vv]**
Reverse-ident scanning	**[– I]**

EXAMPLE: **nmap –O –sS –sV 192.168.1.***

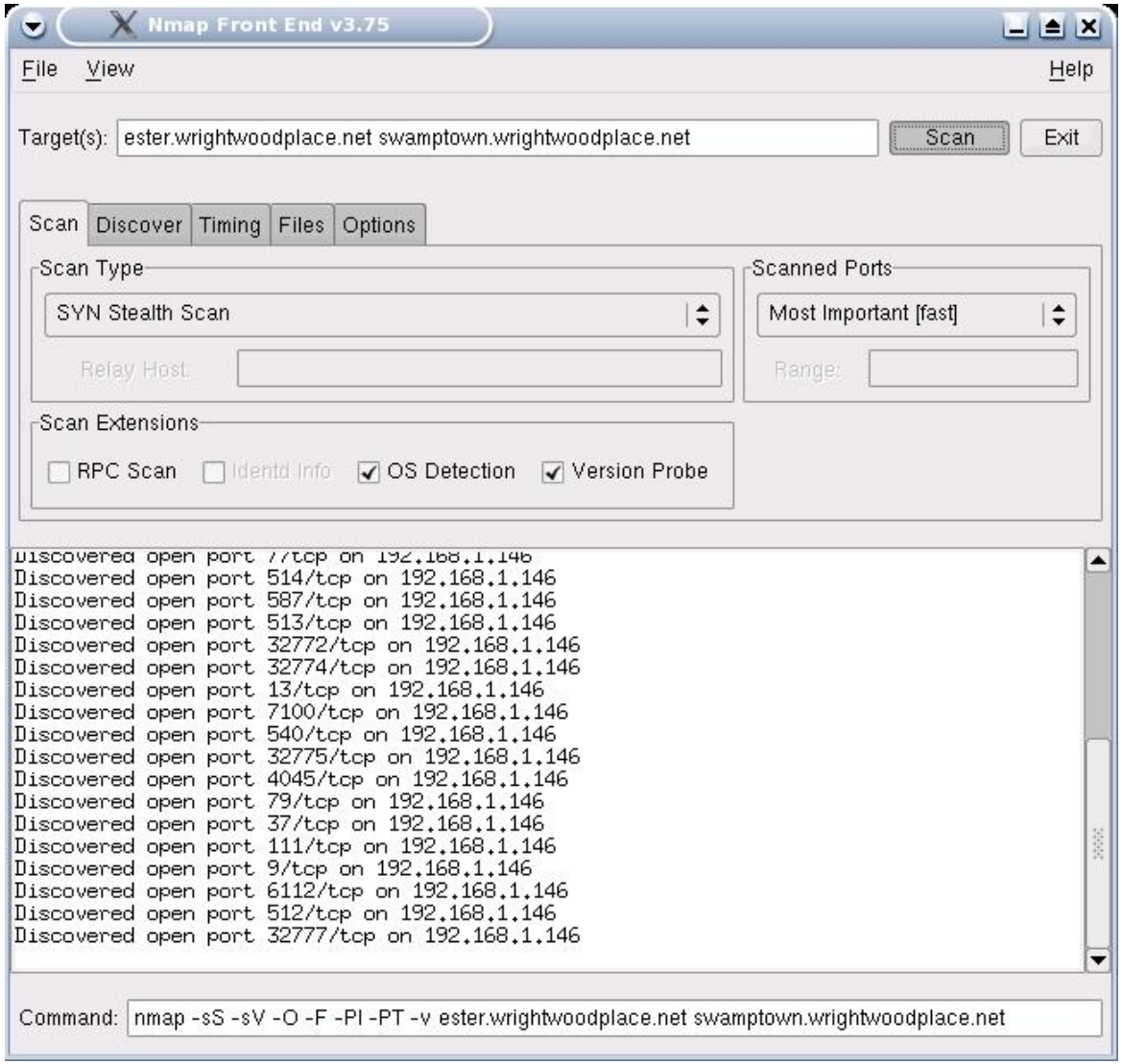
Lab: Running NMAP



Configuring NMAP

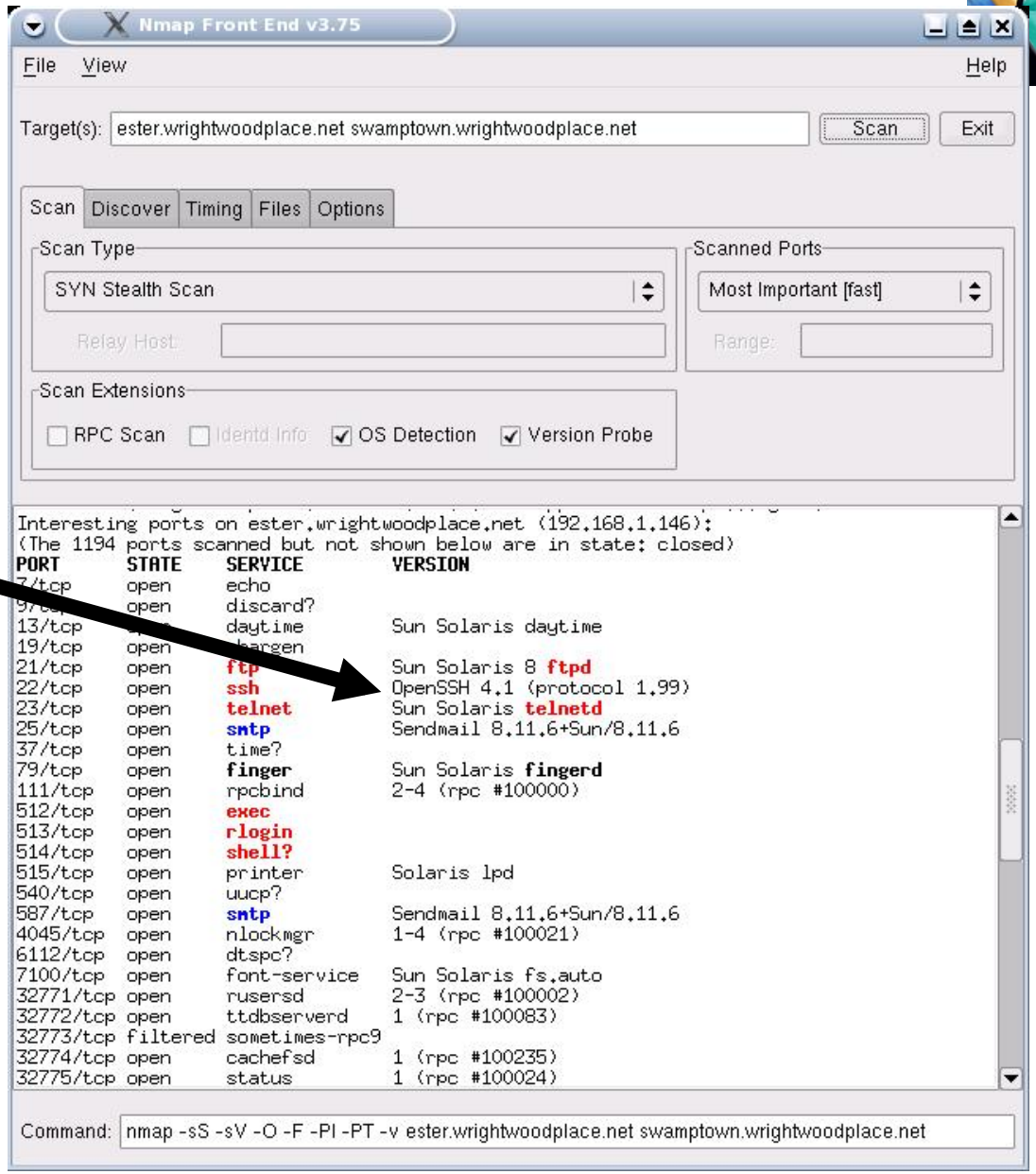


Running NMAP



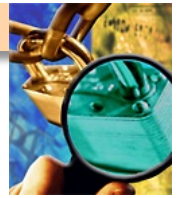
NMAP Output

Service banner grabbing with version numbers?! Outstanding!



NMAP – Common Pitfalls

- Choose ports-to-scan carefully… speed vs. missing stuff
  - Often critical services are listening on a non-standard port #
  - Web (HTTP) servers don't have to listen on port 80, or 443…
- NMAP version-probe functionality… exceptionally useful
  - Will try to figure out not only the port/service, but its version
  - Fantastic feature, but DO *NOT* BLINDLY TRUST THIS OUTPUT
  - Sometimes advertising a precise service version is bad/unsafe
  - Some specialists advise falsifying or not showing versions
- Scanning a host that doesn't respond to PING is painful
  - You have to use the –P0 option, scan may take 10x as long
- Consider scan-timing carefully, might overwhelm a target
  - Use the nmap –T flag (–T 0 is polite/slow, –T 5 is max/insane)



Nmap Screenshot

Nmap Front End v3.49

File View Help

Target(s): www.insecure.org Scan Exit

Scan | Discover | Timing | Files | Options

Scan Type: SYN Stealth Scan

Relay Host:

Scanned Ports: Most Important [fast]

Range:

Scan Extensions: RPC Scan | Identd Info | OS Detection | Version Probe

```
Starting nmap 3.49 ( http://www.insecure.org/nmap/ ) at 2003-12-19 14:28 PST
Interesting ports on www.insecure.org (205.217.153.53):
(The 1212 ports scanned but not shown below are in state: filtered)
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 3.1p1 (protocol 1.99)
25/tcp  open   smtp    qmail smtpd
53/tcp  open   domain  ISC Bind 9.2.1
80/tcp  open   http    Apache httpd 2.0.39 ((Unix) mod_perl/1.99_07-dev Perl/v5.6.1)
113/tcp closed auth
Device type: general purpose
Running: Linux 2.4.X|2.5.X
OS details: Linux Kernel 2.4.0 - 2.5.20
Uptime 212.119 days (since Wed May 21 12:38:26 2003)

Nmap run completed -- 1 IP address (1 host up) scanned in 33.792 seconds
```

Command: nmap -sS -sV -O -F -PI -T4 www.insecure.org

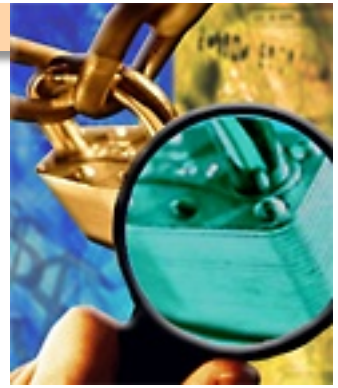
Lab

- Perform a TCP nmap scan of the solaris and windows server IP addresses
  - Write the output in all file formats (-oA)
  - Save the results for tomorrow, you will need them

Break!

Take a breather, ask the instructor questions, experiment…

(But don’t work through all the breaks, you’ll lose focus)

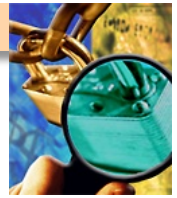


Vulnerability Scanning

Overview

- Purpose
  - Basic instruction in the use of Nessus
  - Familiarity such that students can use tools on their own
  - Maybe a sprinkling of associated vuln-scan techniques
- Format
  - Discussion of vuln-scanning and surrounding issues
  - Lecture and demonstration of Nessus (possibly others)
  - Hands-on lab with instructor supervision

Vulnerability Scanning



- Definition
  - Probing specific services/protocols for weaknesses
  - Not just generic IP addresses anymore
  - Most useful when working from pre-gathered info

    [such as a network-wide NMAP scan you ran earlier]

- Methodology
  - Manual Attempts and Permutation (this will take a long time…)
  - Manual Version Probe (slightly better, but still very slooooow…)
  - Custom Protocol-Specific Attacks (requires special knowledge)
  - Automated Vuln-Scanner (simple, fairly reliable, fast, thorough)

    DO *NOT* TRUST VULN-SCANNER OUTPUTS BLINDLY

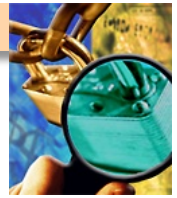
    IF SCANNER FINDS A “VULNERABILITY”, VERIFY IT

Vulnerability Scanning

- **Vuln-Assessment Analysis**
  - **What vulnerabilities are visible from an outside-eye view?**
  - **What vulnerabilities are visible from an inside-firewall view?**
  - **What is the severity of the vulnerabilities discovered?**
  - **Are the vulnerabilities false-positives? Could you verify them?**
  - **There will likely be too many vulnerabilities to handle in a visit**
  - **Ranking by “severity” and “outside exposure” helps prioritize**
  - **It also helps the system owner understand, you can tell a better story**

Common Network Vuln-Scanners

- Nessus
- Tenable Scanner
- ISS Vulnerability Scanner
- eEye Retina
- Microsoft Baseline Security Analyzer (MBSA)
- SAINT

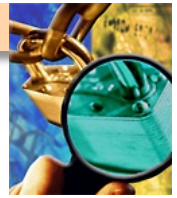


eEye Retina (now McAfee-owned)

- Commercial utility for automated vulnerability scanning
- Widely used in DoD
- Runs on Windows
- Point and click interface
- Generates professional looking reports
- Website: http://www.eeye.com
- Cost: $995 - $2995 or more.

eEye Digital Security

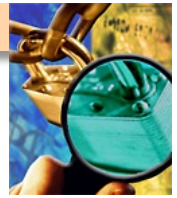
Retina
NETWORK SECURITY SCANNER



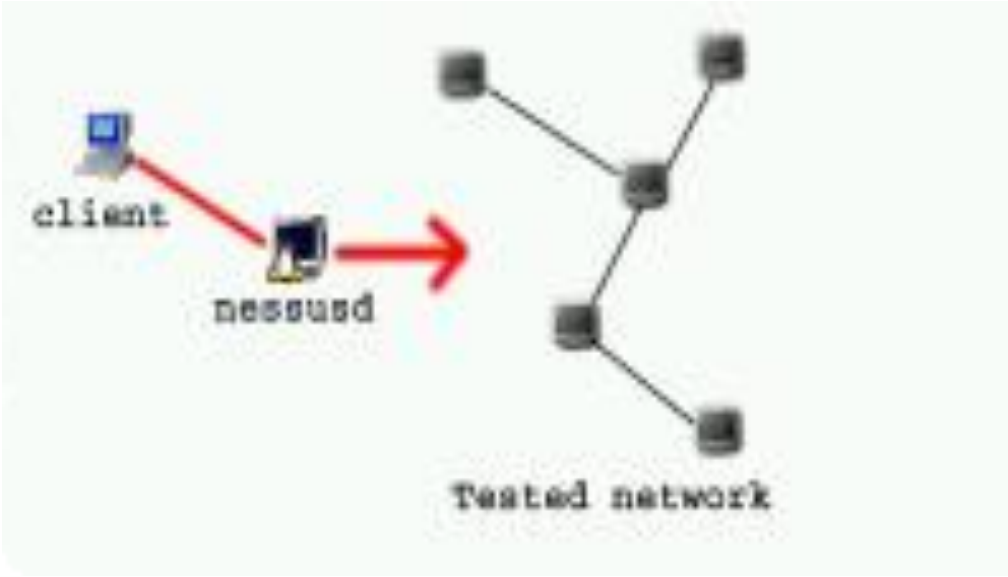
Nessus

- Nessus is a fast and modular vulnerability scanner
  - Widely used and accepted by the security community
- Runs as a client/server or browser/web server installation
  - Your Nessus 'client' connects to a 'server', gets policy, plugins
- Highly configurable and intelligent
  - Thousands of attack/vulnerability plugins available
  - Users can even write their own plugins with NASL scripting
- Runs on Linux, BSD, and even Windows
- Now does local host scanning
- Website: http://www.nessus.org
- Cost: FREE (kind of, be careful here)
  - 2.0 is still open source
  - 3.0 and beyond are closed-source

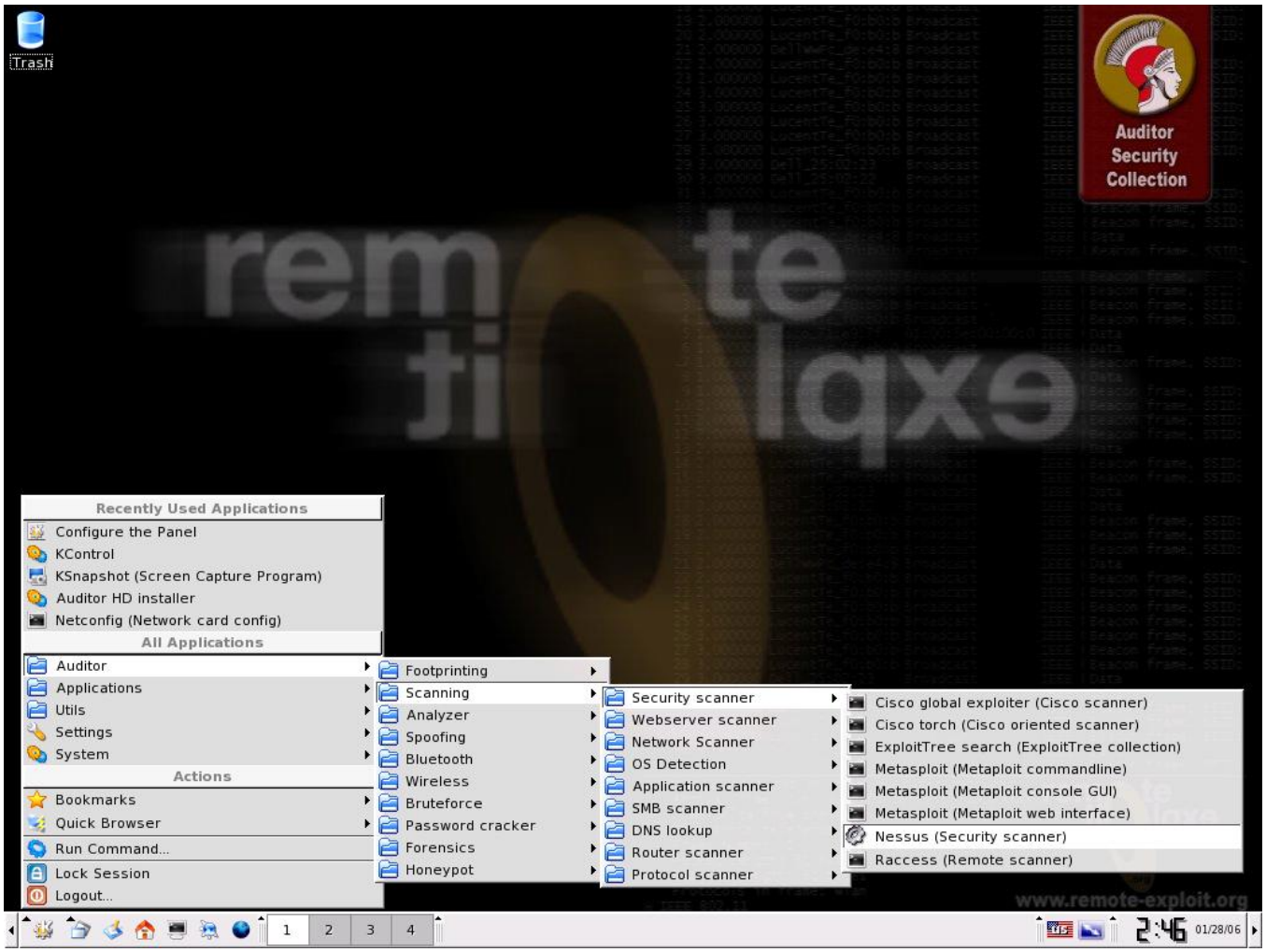
Nessus Architecture



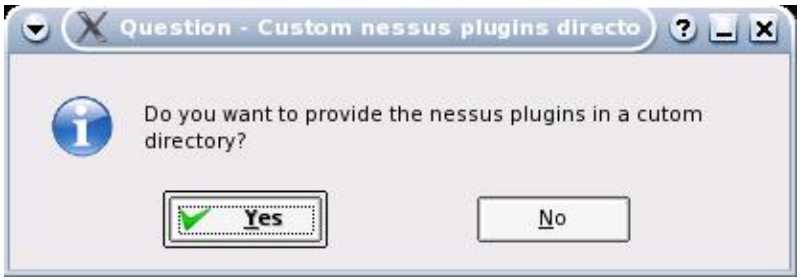
- nessusd (server) controls the attacks
- nessus (client) front end to configure the server
- Multi user with ACLs for each user
- Secure communications between server and client

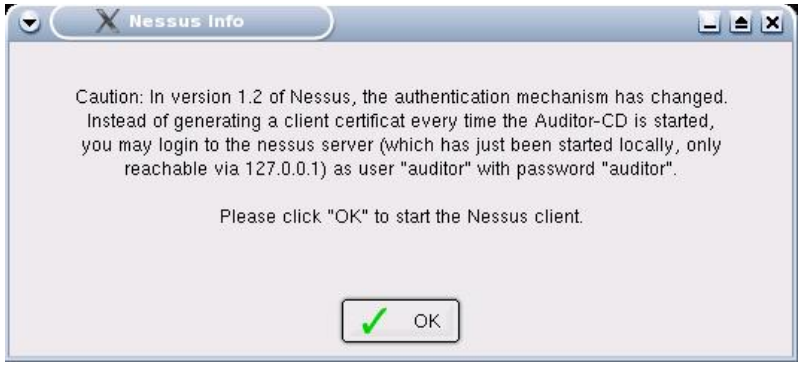


Starting Nessus in BackTrack (or similar)

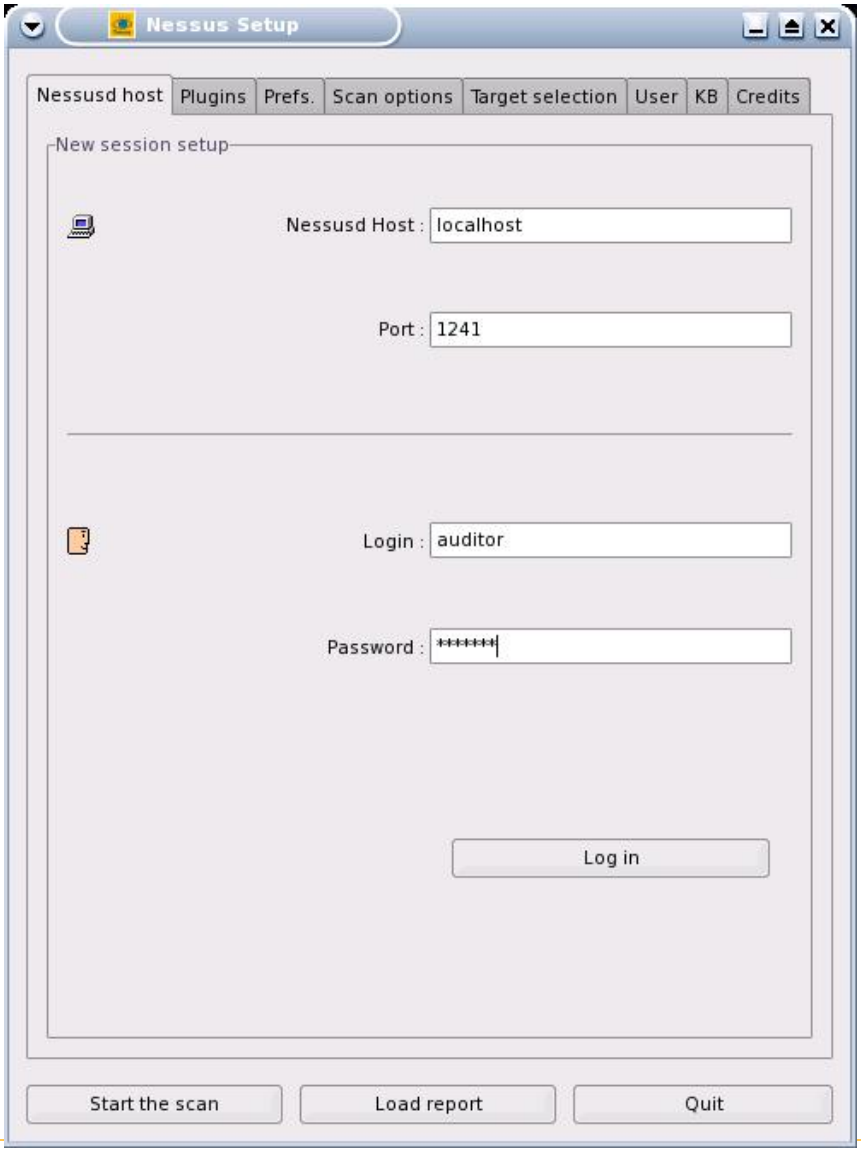


Starting Nessus…

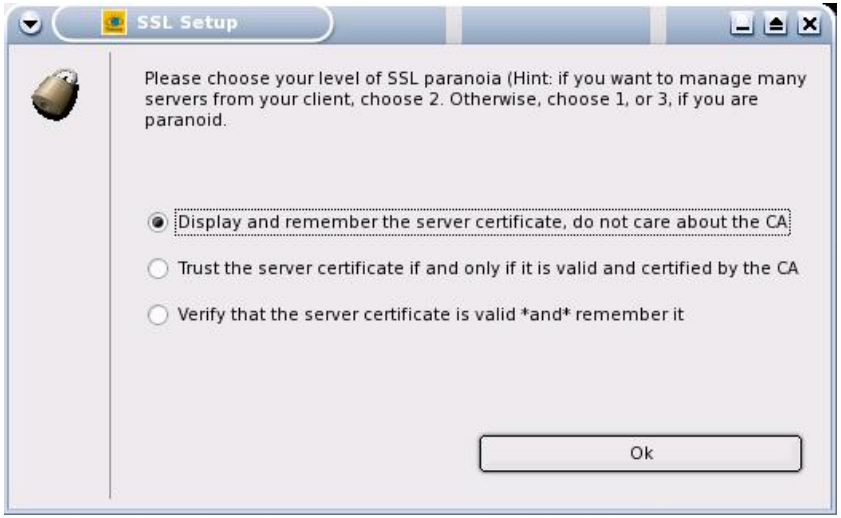


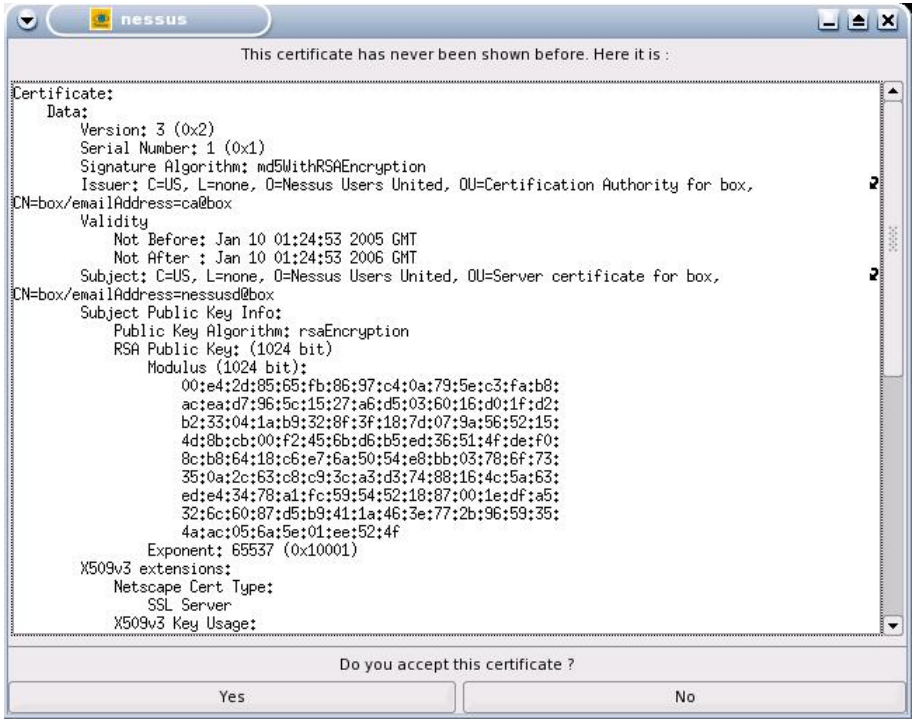


Logging Into Nessus



More Logging In…



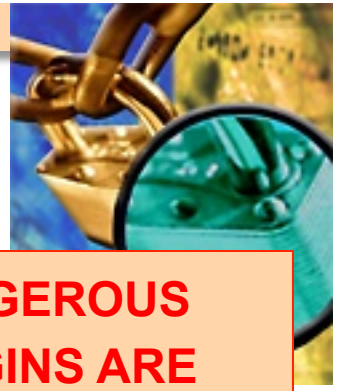


Nessus Plugin Intelligence

- All plug-ins have the ability to share their information
  - (they work together, not just run the same checks repeatedly)
- Example
  - A first plug-in determines port UDP/137 and TCP/139
  - A second plug-in retrieves the remote host netbios name
  - A third attempts to login with the null session
  - A fourth retrieves the remote host SID
  - A fifth enumerates the users/shares on the host
- This method provides for a more comprehensive audit
- Don’t be too optimistic… Nessus scans are still pretty slow

Nessus – Plugin Updates

- Nessus parent company (Tenable) releases these regularly
  - Some delay between “paying” customers and “free” users
  - Update by clicking “Update” or running “nessus-fetch”
  - YOU MUST GET INTO THE HABIT OF DOING THIS REGULARLY
- False positives are preferred over false negatives
  - I’d rather alert on something that’s not there than miss stuff
  - The Nessus developers (who write plugins) feel this way too
- Plugins are duly tested and reviewed before publishing
- Published in CVS version-sourcing and on the web
- Can be customized via NASL scripting language



DANGEROUS PLUGINS ARE SECURITY TESTS THAT CAN CRASH OR DAMAGE A TARGET SYSTEM

Nessus Setup

Nessusd host | Plugins | Prefs. | Scan options | Target selection | User | KB | Credits

Plugin selection

- CGI abuses
- FTP
- Windows
- Gain a shell remotely
- Denial of Service
- Backdoors
- Windows : User management
- Remote file access
- SNMP
- Default Unix Accounts
- RPC

Enable all | Enable all but dangerous plugins | Disable all | Upload plug

Enable dependencies at runtime

Filter...

- Unchecked Buffer in XP Redirector (Q810577)
- DCE Services Enumeration
- Unchecked Buffer in PPTP Implementation Could Enable DOS Attacks
- The messenger service is running
- Microsoft's SQL TCP/IP listener is running
- SMB Registry : is the remote host a PDC/BDC
- SMB Registry : value of SFCDisable
- Unchecked Buffer in Decompression Functions(Q329048)
- WM_TIMER Message Handler Privilege Elevation (Q328310)
- Telnet Client NTLM Authentication Vulnerability
- Opening Group Policy Files (Q318089)

Start the scan | Load report | Quit

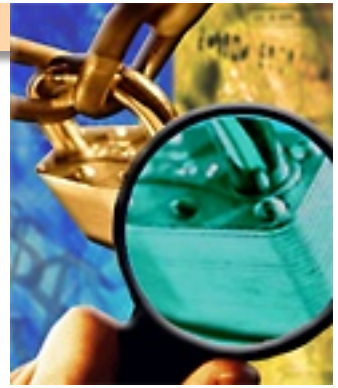
Nessus Setup

Nessusd host | Plugins | Prefs. | Scan options | Target selection | User | KB | Credits

Plugins preferences

NNTP account :	
NNTP password (sent in clear) :	
FTP account :	anonymous
FTP password (sent in clear) :	**********
FTP writeable directory :	/incoming
POP2 account :	
POP2 password (sent in clear) :	
POP3 account :	
POP3 password (sent in clear) :	
IMAP account :	
IMAP password (sent in clear) :	
SMB account :	Administrator
SMB password (sent in clear) :	*********
SMB domain (optional) :	
SNMP community (sent in clear) :	

Start the scan | Load report | Quit



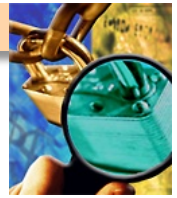
Nessus Scan Method

- Banner grabbing (method 1)
  - Pros
    - Tests are easy to write
    - Scanner is not intrusive
    - It will not harm the remote host
  - Cons
    - What if there is no banner (RPC)?
    - False negatives
    - Non standard application/custom banners

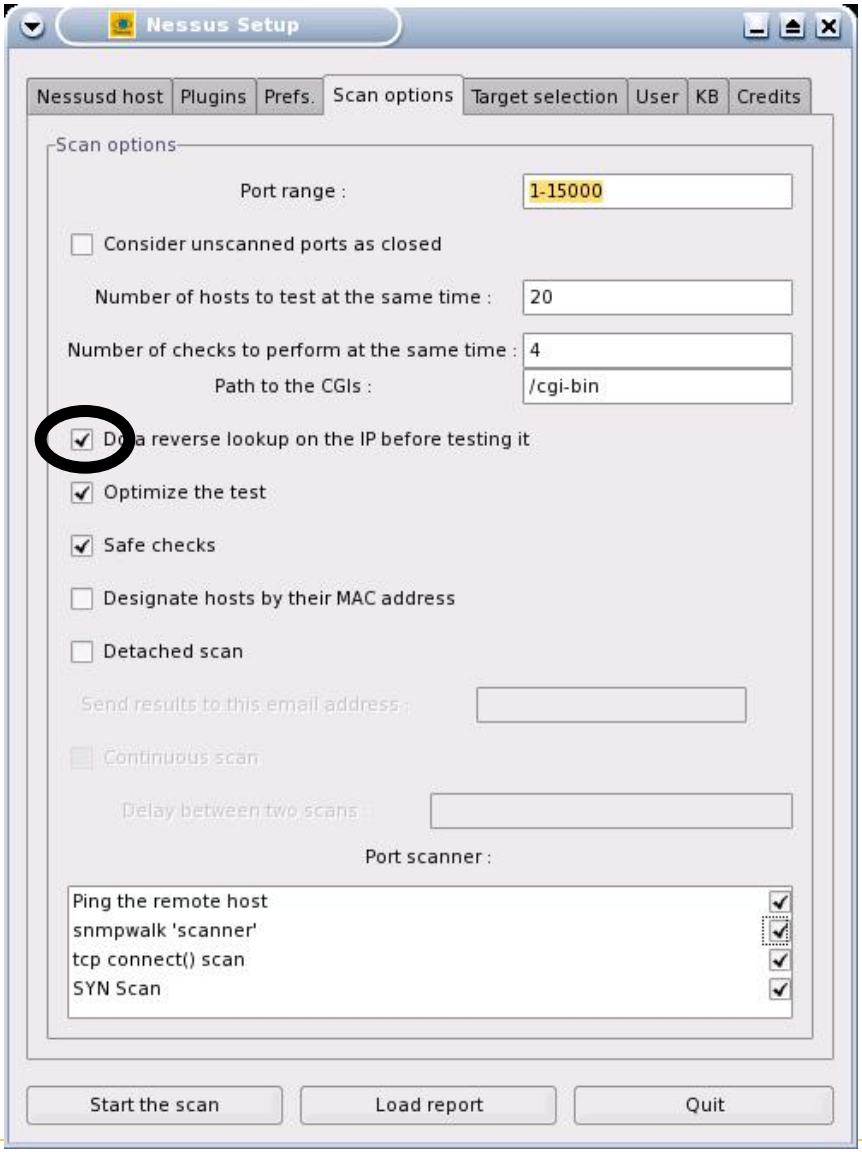
Nessus Scan Method

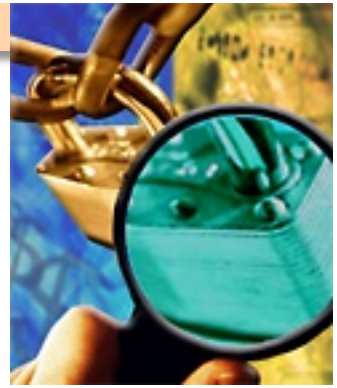
- Actually testing for the vulnerability (method 2)
  - Pros
    - Reliable against unknown servers
    - Results valid at a later time
    - Find new bugs
  - Cons
    - May harm the remote host (crash the service)
    - Tests are more difficult to write
    - May produce false positives

Configuring Nessus: Scan Options



Be careful of anything that requires DNS while in the lab!





Nessus Setup

Nessusd host | Plugins | Prefs. | Scan options | Target selection | User | KB | Credits

Target selection

Target(s) : 10.163.156.0-10.163.156.254 Read file...

Perform a DNS zone transfer

Save this session

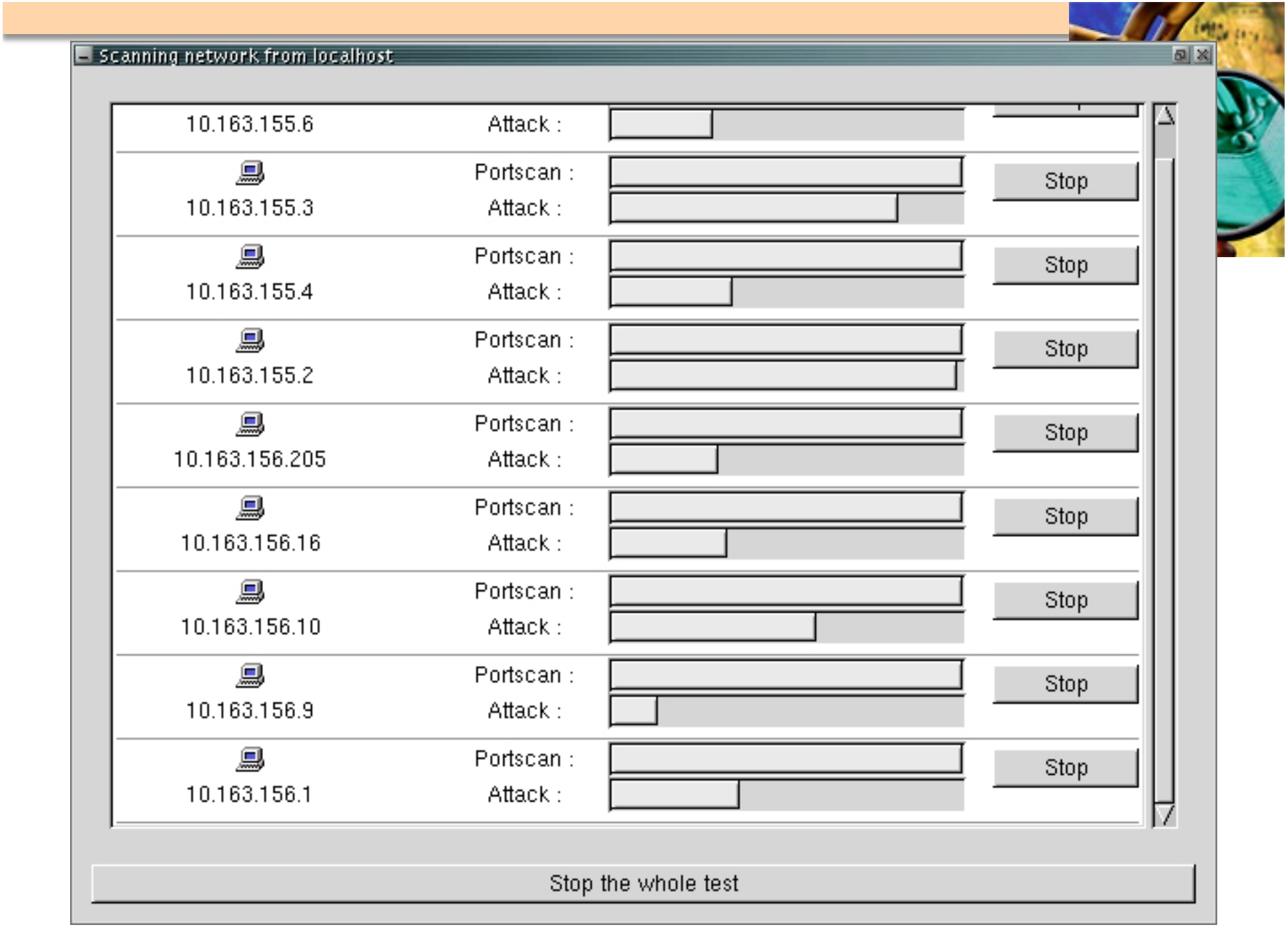
Save empty sessions

Previous sessions :

Session	Targets

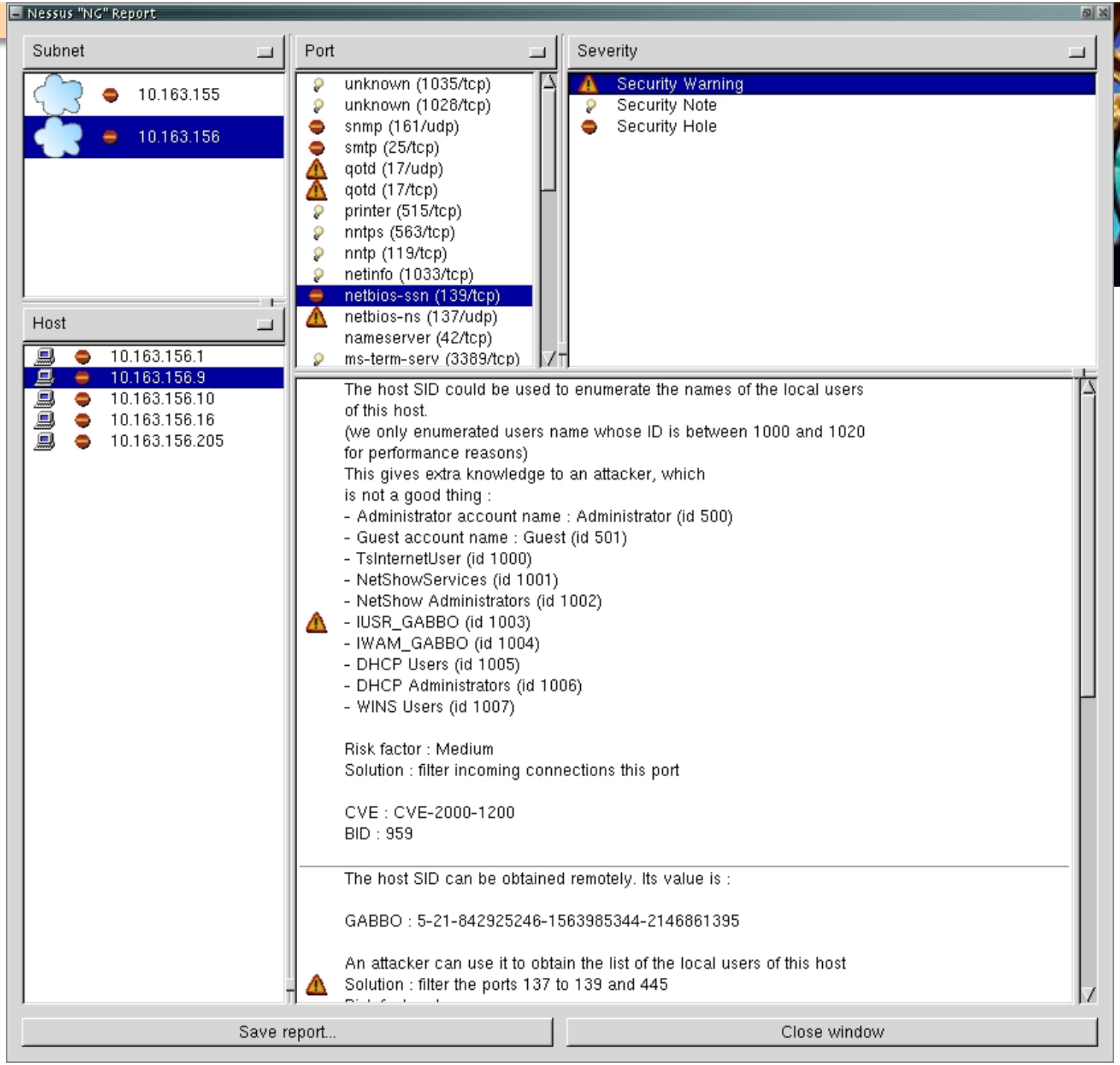
Restore session | Delete session

Start the scan | Load report | Quit

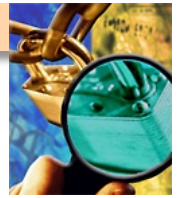

Scanning network from localhost
10.163.155.6
Attack :
Portscan :
10.163.155.3
Attack :
Stop
Portscan :
10.163.155.4
Attack :
Stop
Portscan :
10.163.155.2
Attack :
Stop
Portscan :
10.163.156.205
Attack :
Stop
Portscan :
10.163.156.16
Attack :
Stop
Portscan :
10.163.156.10
Attack :
Stop
Portscan :
10.163.156.9
Attack :
Stop
Portscan :
10.163.156.1
Attack :
Stop
Stop the whole test

Reports

- Multiple formats
  - HTML
  - HMTL with charts/graphics
  - Text
  - NBE (proprietary Nessus report format)
    (can be re-loaded, re-read, re-used by a Nessus scanner)
- Detailed results of scans
- We (your instructors) typically save .HTM and .NBE copies
  - The .HTML reports are easy to read, include in system owner reports
  - The .NBE dumps can be easily picked up and re-run on followup visits

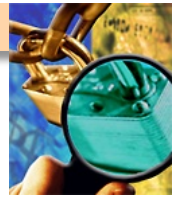

Nessus "NG" Report
Subnet
10.163.155
10.163.156
Port
unknown (1035/tcp)
unknown (1026/tcp)
snmp (161/udp)
smtp (25/tcp)
qotd (17/udp)
qotd (17/tcp)
printer (515/tcp)
nntps (563/tcp)
nntp (119/tcp)
netinfo (1033/tcp)
netbios-ssn (139/tcp)
netbios-ns (137/udp)
nameserver (42/tcp)
ms-term-serv (3389/tcp)
Severity
Security Warning
Security Note
Security Hole
Host
10.163.156.1
10.163.156.9
10.163.156.10
10.163.156.16
10.163.156.205
The host SID could be used to enumerate the names of the local users
of this host.
(we only enumerated users name whose ID is between 1000 and 1020
for performance reasons)
This gives extra knowledge to an attacker, which
is not a good thing :
- Administrator account name : Administrator (id 500)
- Guest account name : Guest (id 501)
- TsInternetUser (id 1000)
- NetShowServices (id 1001)
- NetShow Administrators (id 1002)
- IUSR_GABBO (id 1003)
- IWAM_GABBO (id 1004)
- DHCP Users (id 1005)
- DHCP Administrators (id 1006)
- WINS Users (id 1007)
Risk factor : Medium
Solution : filter incoming connections this port
CVE : CVE-2000-1200
BID : 959
The host SID can be obtained remotely. Its value is :
GABBO : 5-21-842925246-1563985344-2146861395
An attacker can use it to obtain the list of the local users of this host
Solution : filter the ports 137 to 139 and 445
Save report...
Close window

Common Nessus Pitfalls



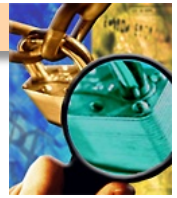
- DO *NOT* RUN NESSUS, SAY “DONE” AND WALK AWAY
  - Nessus will almost certainly generate false positives
  - Treat as a preliminary indicator (“something might be wrong”)
  - Must follow-up/confirm/verify (look at target’s config files, etc.)
- Proceed carefully with Nessus config/settings
  - I love to run “dangerous” plugins, but that’s really very risky
  - Should *NOT* run these without explicit permission
- Even “non-dangerous” Nessus plugins can crash a host
  - It’s regrettable (and a bit embarrassing) when this happens
  - *BUT* write it up as a finding, the system owner needs to know
  - If you can do it by accident, an outsider could as well…

Nessus Lab



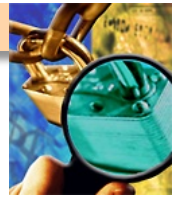
- Perform a scan of the solaris and windows server IP addresses
  - Write the output in all file formats (-oA)
  - Save the results for tomorrow, you will need them
- Bonus points if you manage to crash something in the process

Questions?



- Break?
- Stay tuned for the bonus slides…

A Few Places to Get Tools



- Much of this stuff (tools) comes pre-canned in Linux CDs
- If using BSD, “portupgrade –rR security/???” syntax

- Freshmeat (http://www.freshmeat.net)
- Fyodor (http://insecure.org) maintains the NMAP scanner
- PacketStorm (http://www.packetstormsecurity.org)
- Tenable (http://www.nessus.org) has the Nessus scanner

  [pls check the commercial license on this tool, it’s not “free”]

- Subsequent instructors will make other recommendations
- Various conventions will have stuff as well, but be careful!

  (BlackHat, CanSecWest, DefCon, ShmooCon, ToorCon)

Questions