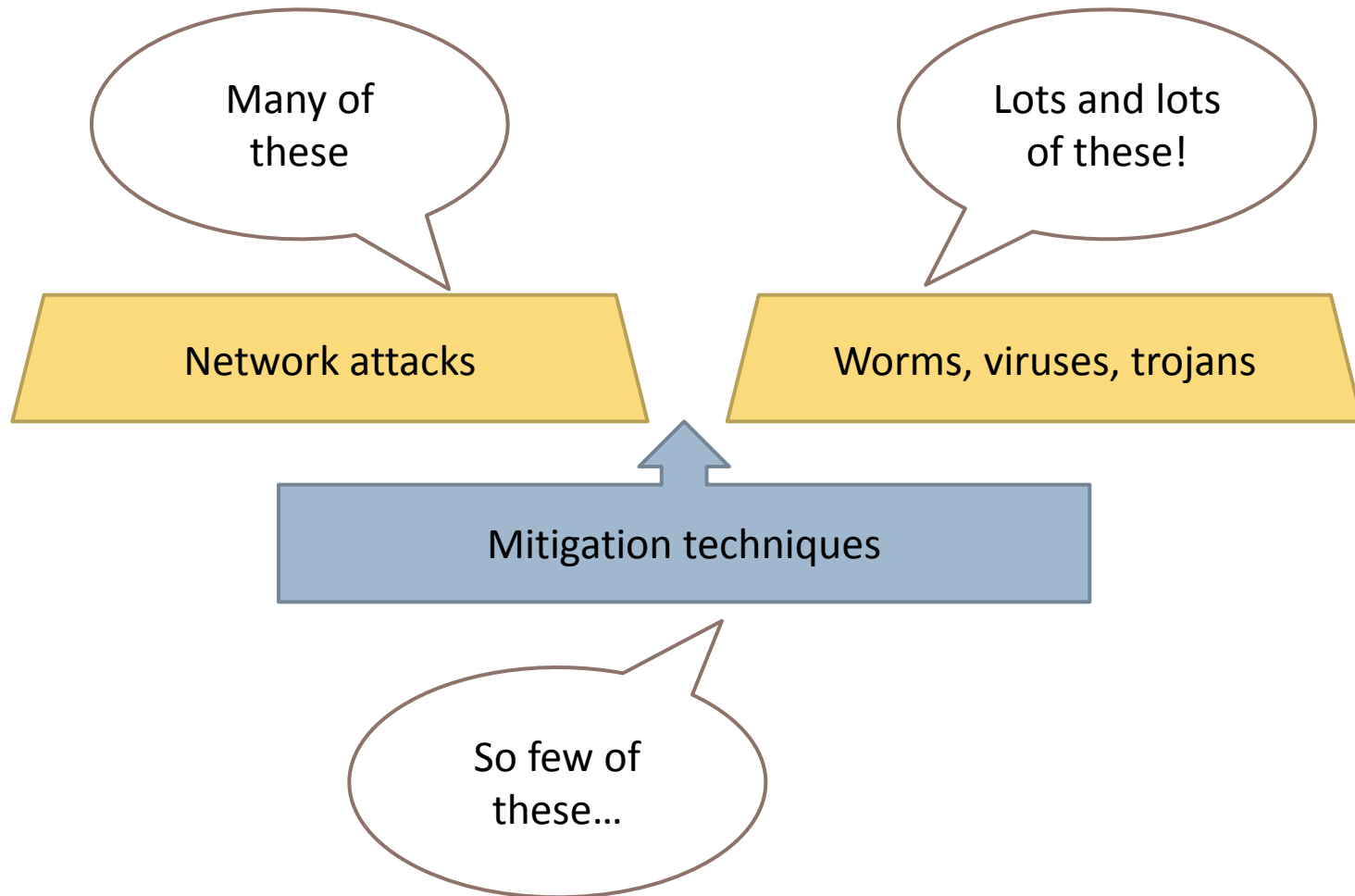


Network Attacks

October 14, 2014

What's threatening us?



What type of network attacks are there?

- ▶ Reconnaissance attacks



- ▶ Access attacks



- ▶ Denial-of-service attacks



Reconnaissance attacks

- ▶ First of all: find what to attack
- ▶ Get as much info as possible on your target
- ▶ Even public information can be useful.
- ▶ Purpose: identifying possible vulnerabilities
- ▶ Similar to a thief surveying a neighborhood for vulnerable homes to break into or cars to steal.



Reconnaissance attacks – hosts and ports

- ▶ First step – identify the vulnerable services
- ▶ How?
 - ▶ Perform a ping sweep to determine active hosts in a network
 - ▶ Obtain information about the operating system running on the active hosts
 - ▶ Scan active hosts for open ports to determine what services are running
 - ▶ Open ports often provide information about the service's version
- ▶ Vulnerable services are identified and can be exploited
- ▶ Port scanners: nmap, nessus

Nmap example #1

Ping sweep

```
linux$ sudo nmap -sP 141.85.37.0/24
Starting Nmap 5.00 ( http://nmap.org ) at 2009-10-09 18:12 EEST
Host csr.cs.pub.ro (141.85.37.1) is up (0.00040s latency).
  MAC Address: 00:09:6B:89:06:67 (IBM)
Host ns.catc.ro (141.85.37.2) is up (0.00097s latency).
  MAC Address: 00:17:31:49:3A:E4 (Asustek Computer)
Host prof.cs.pub.ro (141.85.37.3) is up (0.00043s latency).
  MAC Address: 00:09:6B:89:05:24 (IBM)
Host turing.cs.pub.ro (141.85.37.7) is up (0.00089s latency).
  MAC Address: 00:50:56:9A:33:46 (VMware)
Host ns.cs.pub.ro (141.85.37.8) is up (0.00028s latency).
  MAC Address: 00:09:6B:89:06:67 (IBM)
Host ef001.cs.pub.ro (141.85.37.9) is up (0.00088s latency).
  MAC Address: 00:15:5D:25:14:00 (Microsoft)
Host dnscache.cs.pub.ro (141.85.37.11) is up (0.00047s latency).
  MAC Address: 00:09:6B:89:06:67 (IBM)
Host xeno.cs.pub.ro (141.85.37.12) is up (0.00088s latency).
  MAC Address: 00:50:56:9A:51:6D (VMware)
Host nix.cs.pub.ro (141.85.37.13) is up (0.00088s latency).
  MAC Address: 00:EE:B1:03:0A:DE (Unknown)
```

Nmap example #2

OS identification
Open ports listing

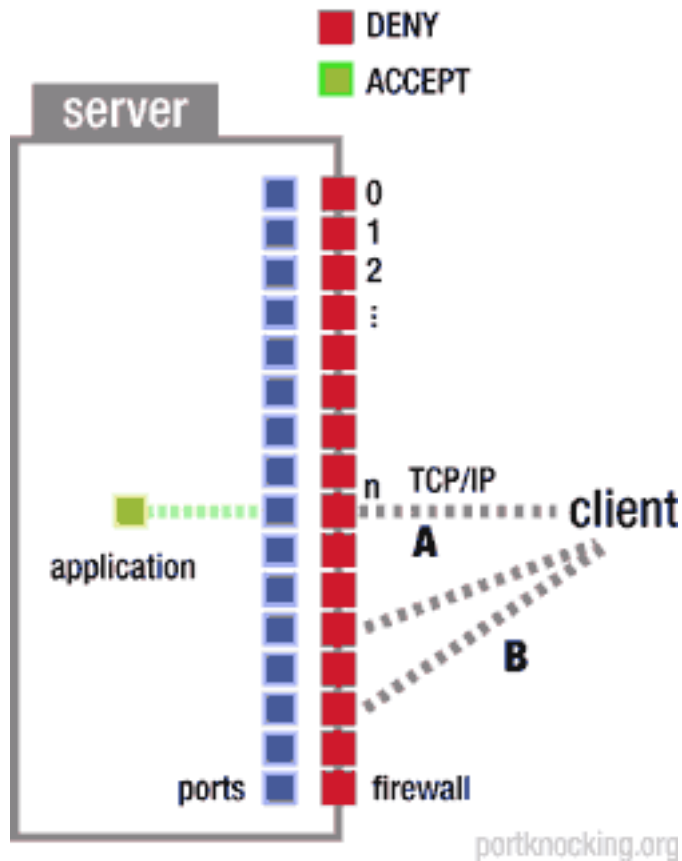
```
linux$ sudo nmap -sS -O 141.85.37.132
Starting Nmap 5.00 ( http://nmap.org ) at 2009-10-09 18:21 EEST
Interesting ports on dhcp-132.cs.pub.ro (141.85.37.132):
Not shown: 996 closed ports
PORT      STATE SERVICE
88/tcp    open  kerberos-sec
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3323/tcp  open  unknown
Device type: general purpose
Running: Apple Mac OS X 10.5.X
OS details: Apple Mac OS X 10.5 - 10.5.6 (Leopard) (Darwin 9.0.0 - 9.6.0)

OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.16 seconds
```

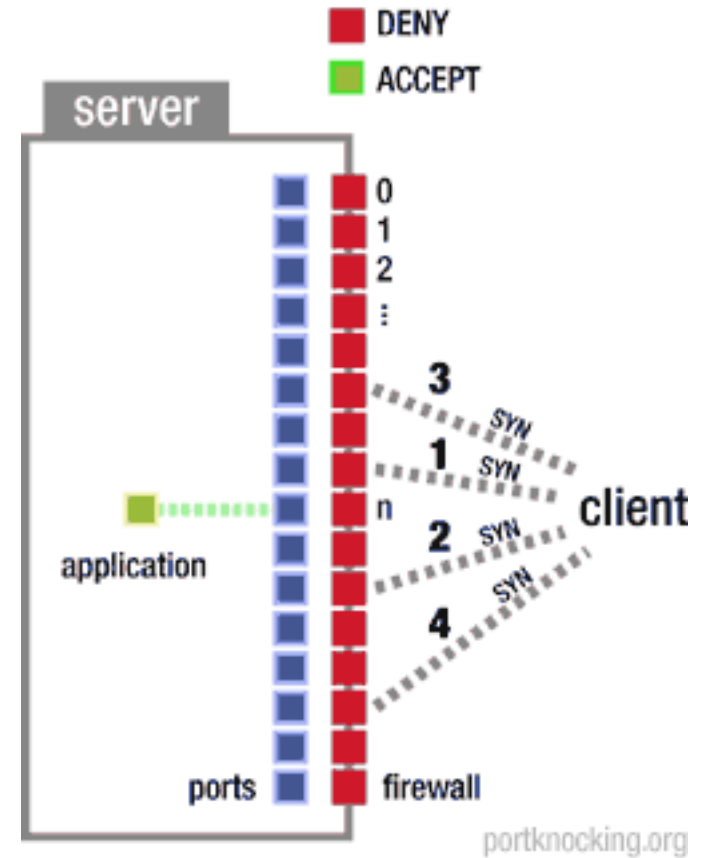
How to avoid port scanning?

- ▶ Theoretically, you cannot
- ▶ All open ports **will be detected**
- ▶ How can you hide it then?
- ▶ Answer: **Port Knocking**
 - ▶ By default, the desired port is closed
 - ▶ The client sends a set of SYN packets in a certain order
 - ▶ A daemon listens for a specific sequence of SYN packets sent to closed ports
 - ▶ If the sequence is correct, the desired port will be open and the “knocker” will be allowed to send data
- ▶ Of course, the client has to know the “knock” sequence

Port knocking phases

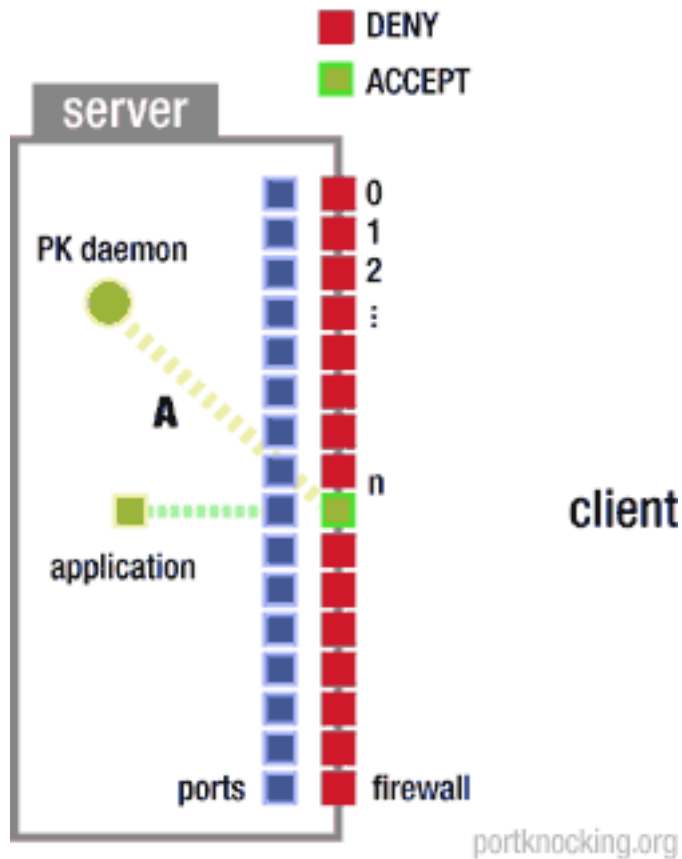


A) The client cannot connect to the application. The client cannot establish a connection to any port.

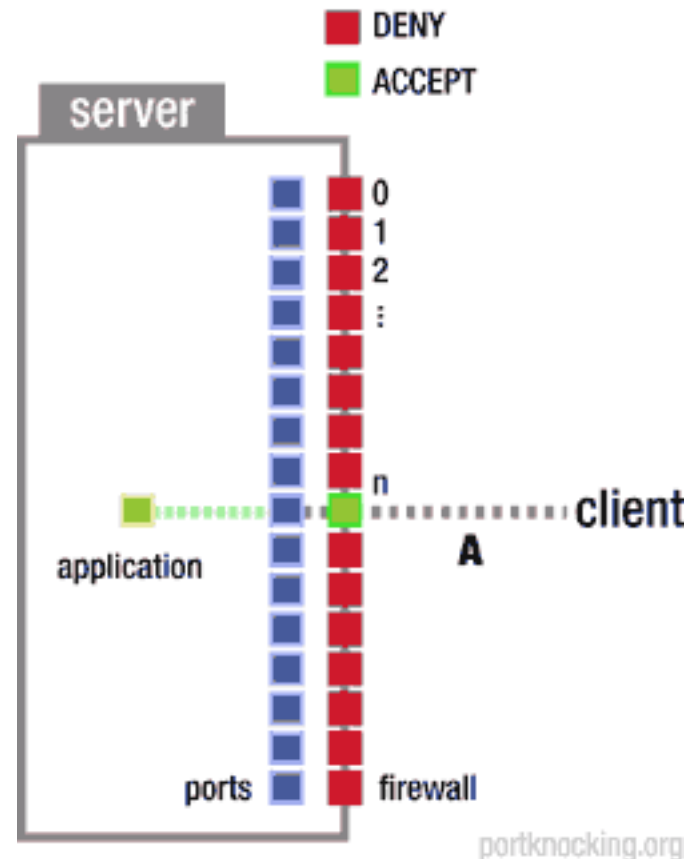


B) The client attempts connection to a number of ports in a predefined sequence. Client receives no ACKs.

Port knocking phases



C) The PK daemon interprets the attempts and carries out a task. For example, it opens a specific port (n).

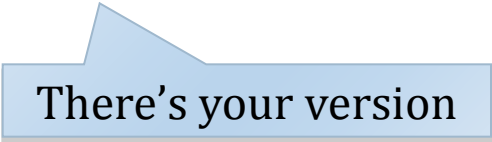


D) The client can now connect to port n.

Reconnaissance attacks – who is running what?

- ▶ To sum up: Who is providing the information?
 - ▶ Ping sweeps determine which hosts are “alive”
 - ▶ Port scanning determines which services are running
 - ▶ Well-known services run on well-known ports (TCP and UDP)
 - ▶ Telnetting to an open port will most likely return a banner informing you of the service running on that port.

```
linux$ telnet cs.pub.ro 22
Trying 141.85.37.5...
Connected to cs.pub.ro.
Escape character is '^]'.
SSH-2.0-OpenSSH_5.1p1 Debian-5
```



There's your version

The version issue

- ▶ Hiding the service's version is NOT REALLY helpful...
 - ▶ Hackers usually try all the exploits they have
 - ▶ If your version has a vulnerability, it's still there
- ▶ Not all services allow you to modify it
 - ▶ Open SSH doesn't allow it, by default
 - ▶ You need to edit and recompile the sources or...
... use a commercial version
- ▶ Some services allow it and it's quite simple
 - ▶ For example, vsftpd's configuration file:
`ftpd_banner=...`

Reconnaissance attacks – packet sniffing

- ▶ Sniffing random traffic can also provide useful information about the network and its services
- ▶ **Promiscuous mode** sniffing
 - ▶ The network card will process traffic that is normally dropped
 - ▶ The OS has to “agree” with this – not all OS'es support it
- ▶ **Listening:**

Shared network (no switches)	Switched network
Traffic between any two hosts is seen by all (shared segment, hubs).	Traffic is isolated at switchport level.

- ▶ Packet sniffers: Wireshark, tcpdump

Wireshark

- ▶ **General-purpose protocol analyzer**
 - ▶ Displays the entire content of packets passing through the network adapter.
 - ▶ Identifies a great range of protocols: from data link layer to application layer.
 - ▶ Can follow streams of data based on TCP sequence numbers.
 - ▶ Can define filters, save results.
 - ▶ Can perform VoIP analysis.
 - ▶ Supports 802.11, PPP, ATM, Bluetooth, etc.
 - ▶ Displays IPsec, WEP, WPA(2) as decrypted.
 - ▶ Multi-platform



Wireshark interface

The screenshot shows the Wireshark interface with a list of captured packets and a detailed view of a selected packet. The packet list shows the following data:

No.	Time	Source	Destination	Protocol	Info
29	1.270306	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [ACK] Seq=190 Ack=190
30	1.259654	192.168.0.1	192.168.0.2	TCP	[TCP Window Update] http > 3197
31	1.266628	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [PSH, ACK] Seq=1 Ack=
32	1.266819	192.168.0.2	192.168.0.1	TCP	5000 > 1025 [PSH, ACK] Seq=1 Ack=
33	1.267850	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [ACK] Seq=510 Ack=20
34	1.274361	192.168.0.1	192.168.0.2	TCP	http > 3197 [PSH, ACK] Seq=1 Ack=
35	1.274447	192.168.0.2	192.168.0.1	TCP	3197 > http [FIN, ACK] Seq=190 Ac
36	1.274987	192.168.0.1	192.168.0.2	TCP	http > 3197 [FIN, ACK] Seq=20 Ack
37	1.275018	192.168.0.2	192.168.0.1	TCP	3197 > http [ACK] Seq=191 Ack=21
38	1.276019	192.168.0.1	192.168.0.2	TCP	http > 3197 [FIN, ACK] Seq=26645
39	1.281649	192.168.0.1	192.168.0.2	TCP	[TCP Window Update] 1025 > 5000
40	1.282181	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [FIN, ACK] Seq=510 Ac

The detailed view of the selected packet (Frame 36) shows the following structure:

- Frame 36 (60 bytes on wire, 60 bytes captured)
- Ethernet II, Src: Netgear_2d:75:9a (00:09:5b:2d:75:9a), Dst: 192.168.0.2 (00:0b:5d:20:cd:02)
- Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.2 (192.168.0.2)
- Transmission Control Protocol, Src Port: http (80), Dst Port: 3197 (3197), Seq: 20, Ack: 190, Len: 0
 - Source port: http (80)
 - Destination port: 3197 (3197)
 - Sequence number: 20 (relative sequence number)
 - Acknowledgement number: 190 (relative ack number)
 - Header length: 20 bytes

The hex/ASCII view of the packet shows the following data:

```
0000 00 0b 5d 20 cd 02 00 09 5b 2d 75 9a 08 00 45 00  ...] ... [-u...E.  
0010 00 28 00 84 00 00 04 06 f8 f8 c0 a8 00 01 c0 a8  .(....@.....  
0020 00 02 00 50 0c 7d 00 00 68 14 3c 38 dd 9b 50 11  .P.)...h.<8..P.  
0030 0c 00 93 ca 00 00 00 00 00 00 00 00 00 00 00 00  .....
```

Summary of captured packets

Detailed tree-view of encapsulated protocols

Hex/ASCII view of packets

Wireshark – DNS query example (Layer 2)

Packet summary

557 34.742747 192.168.2.100 141.85.37.11 DNS Standard query A images.google.ro

Frame 557 (76 bytes on wire, 76 bytes captured)

Arrival Time: Oct 9, 2009 20:31:58.500353000
[Time delta from previous captured frame: 0.096506000 seconds]
[Time delta from previous displayed frame: 0.096506000 seconds]
[Time since reference or first frame: 34.742747000 seconds]
Frame Number: 557
Frame Length: 76 bytes
Capture Length: 76 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:dns]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]

Encapsulated protocols

Ethernet II, Src: Vmware_7b:55:dd (00:0c:29:7b:55:dd), Dst: Tp-LinkT_84:dd:4a (00:19:e0:84:dd:4a)
Destination: Tp-LinkT_84:dd:4a (00:19:e0:84:dd:4a)
Source: Vmware_7b:55:dd (00:0c:29:7b:55:dd)
Type: IP (0x0800)

Layer 2 source and destination addresses

Upper-protocol code (IP)

Wireshark – DNS query example (Layers 3 and 4)

IP header; source and destination addresses

- [-] Internet Protocol, Src: 192.168.2.100 (192.168.2.100), Dst: 141.85.37.11 (141.85.37.11)
 - Version: 4
 - Header length: 20 bytes
 - [+] Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
 - Total Length: 62
 - Identification: 0x0152 (338)
 - [+] Flags: 0x00
 - Fragment offset: 0
 - Time to live: 128
 - Protocol: UDP (0x11)
 - [+] Header checksum: 0xc3f0 [correct]
 - Source: 192.168.2.100 (192.168.2.100)
 - Destination: 141.85.37.11 (141.85.37.11)
- [-] User Datagram Protocol, Src Port: iad2 (1031), Dst Port: domain (53)
 - Source port: iad2 (1031)
 - Destination port: domain (53)
 - Length: 42
 - [+] Checksum: 0xf63e [validation disabled]

UDP header; source and destination ports

Wireshark – DNS query example (Application)

Flags

```
[-] Domain Name System (query)
  [Response In: 560]
  Transaction ID: 0x8bca
  [-] Flags: 0x0100 (Standard query)
    0... .. = Response: Message is a query
    .000 0... .. = Opcode: Standard query (0)
    .... ..0. .... = Truncated: Message is not truncated
    .... ..1 .... = Recursion desired: Do query recursively
    .... ..0. .... = Z: reserved (0)
    .... ..0 .... = Non-authenticated data OK: Non-authenticated data is unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  [-] Queries
    [-] images.google.ro: type A, class IN
      Name: images.google.ro
      Type: A (Host address)
      Class: IN (0x0001)
```

One query

Tcpdump short quiz

▶ Enter the command for capturing 10 packets:

```
linux$ sudo tcpdump -i en1 -c 10
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on en1, link-type EN10MB (Ethernet), capture size 96 bytes
20:58:04.947521 IP 192.168.2.102.65193 > 64.236.76.160.http: FP 3351223874:3351224692(818) ack
  3791731684 win 65535
20:58:05.048363 IP 192.168.2.102.64948 > cs111.msg.sp1.yahoo.com.mmcc: P 3808254532:3808254567(35)
  ack 1468375673 win 65535 <nop,nop,timestamp 1041121821 2502504253>
20:58:05.154875 IP 192.168.2.102.64397 > dnscache.cs.pub.ro.domain: 23404+ PTR? 160.76.236.64.in-
  addr.arpa. (44)
20:58:05.928980 IP dnscache.cs.pub.ro.domain > 192.168.2.102.64397: 23404 NXDomain 0/1/0 (110)
20:58:05.931073 IP 192.168.2.102.60327 > dnscache.cs.pub.ro.domain: 4591+ PTR? 16.217.180.68.in-
  addr.arpa. (44)
20:58:06.236795 IP dnscache.cs.pub.ro.domain > 192.168.2.102.60327: 4591 1/5/5 (251)
20:58:06.648490 arp who-has 192.168.2.112 tell 192.168.2.103
20:58:06.649205 arp who-has 192.168.2.113 tell 192.168.2.103
20:58:07.239861 IP 192.168.2.102.55585 > dnscache.cs.pub.ro.domain: 9323+ PTR? 112.2.168.192.in-
  addr.arpa. (44)
20:58:09.053072 IP 192.168.2.102.64948 > cs111.msg.sp1.yahoo.com.mmcc: P 0:35(35) ack 1 win 65535
  <nop,nop,timestamp 1041121861 2502504253>
10 packets captured
20 packets received by filter
0 packets dropped by kernel
```

Tcpdump short quiz

▶ Enter the command for capturing 10 http requests:

```
linux$ sudo tcpdump -i en1 -c 10 dst port 80
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
listening on en1, link-type EN10MB (Ethernet), capture size 96 bytes
```

```
21:04:27.331834 IP 192.168.2.102.65285 > f36.ymdb.vip.sp2.yahoo.com.http: S 3835386219:3835386219(0) win 65535 <mss 1460,nop,wscale 3,nop,nop,timestamp 1041125640 0,sackOK,eol>
```

```
21:04:27.541264 IP 192.168.2.102.65285 > f36.ymdb.vip.sp2.yahoo.com.http: . ack 346088808 win 65535 <nop,nop,timestamp 1041125642 3613110350>
```

```
21:04:27.541458 IP 192.168.2.102.65285 > f36.ymdb.vip.sp2.yahoo.com.http: P 0:184(184) ack 1 win 65535 <nop,nop,timestamp 1041125642 3613110350>
```

```
21:04:27.796773 IP 192.168.2.102.65250 > 65.55.12.249.http: P 4197506267:4197507391(1124) ack 211762492 win 65535
```

```
21:04:27.860367 IP 192.168.2.102.65285 > f36.ymdb.vip.sp2.yahoo.com.http: . ack 2897 win 65535 <nop,nop,timestamp 1041125645 3613110562>
```

```
21:04:28.076775 IP 192.168.2.102.65285 > f36.ymdb.vip.sp2.yahoo.com.http: . ack 5793 win 65522 <nop,nop,timestamp 1041125648 3613110879>
```

```
21:04:28.232615 IP 192.168.2.102.65250 > 65.55.12.249.http: . ack 4381 win 65535
```

```
21:04:28.236517 IP 192.168.2.102.65250 > 65.55.12.249.http: . ack 7301 win 65535
```

```
21:04:28.244273 IP 192.168.2.102.65250 > 65.55.12.249.http: . ack 10221 win 65535
```

```
21:04:28.260835 IP 192.168.2.102.65285 > f36.ymdb.vip.sp2.yahoo.com.http: . ack 7241 win 65535 <nop,nop,timestamp 1041125649 3613110879>
```

Tcpdump short quiz – Boss 😊

- ▶ Enter the command for capturing and saving to a file all the packets that are not intended for web servers and with numerical address format:

```
$ tcpdump -ni eth0 -w file.cap not port 80
```

- ▶ Enter the command for displaying the capture file:

```
$ tcpdump -r file.cap
```

Reconnaissance attacks – “whois” information

▶ Internet information queries: **whois cisco.com**

Domain Name..... cisco.com	Admin Phone..... +1.4085273842
Creation Date..... 1987-05-14	Admin Fax..... +1.4085264575
Registration Date.... 2011-04-06	Tech Name..... Network Services
Expiry Date..... 2013-05-16	Tech Address..... 170 W. Tasman Drive
Organisation Name.... Cisco Technology, Inc.	Tech Address.....
Organisation Address. 170 W. Tasman Drive	Tech Address.....
Organisation Address.	Tech Address..... San Jose
Organisation Address.	Tech Address..... 95134
Organisation Address. San Jose	Tech Address..... CA
Organisation Address. 95134	Tech Address..... UNITED STATES
Organisation Address. CA	Tech Email..... dns-info@cisco.com
Organisation Address. UNITED STATES	Tech Phone..... +1.4085279223
Admin Name..... Info Sec	Tech Fax..... +1.4085267373
Admin Address..... 170 West Tasman Drive	Name Server..... NS1.CISCO.COM
Admin Address.....	Name Server..... NS2.CISCO.COM
Admin Address.....	
Admin Address. San Jose	
Admin Address..... 95134	
Admin Address..... CA	
Admin Address..... UNITED STATES	
Admin Email..... infosec@cisco.com	



Reconnaissance attacks – DNS information

▶ Listing mail servers

```
linux$ host -t MX cisco.com
cisco.com mail is handled by 25 syd-inbound-a.cisco.com.
cisco.com mail is handled by 10 sj-inbound-a.cisco.com.
cisco.com mail is handled by 10 sj-inbound-b.cisco.com.
cisco.com mail is handled by 10 sj-inbound-c.cisco.com.
cisco.com mail is handled by 10 sj-inbound-d.cisco.com.
cisco.com mail is handled by 10 sj-inbound-e.cisco.com.
cisco.com mail is handled by 10 sj-inbound-f.cisco.com.
cisco.com mail is handled by 15 rtp-mx-01.cisco.com.
cisco.com mail is handled by 20 ams-inbound-a.cisco.com.
```

▶ Listing name servers

```
linux$ host -t NS cs.pub.ro
cs.pub.ro name server ns.cs.pub.ro.
cs.pub.ro name server pub.pub.ro.
```

Access attacks

- ▶ Exploit known vulnerabilities
- ▶ Target services that (normally) do not offer access to everyone
- ▶ This is where password breaking comes into play
- ▶ Purpose: to gain access to servers, accounts and confidential data
 - ▶ basically: to steal or destroy stuff
- ▶ What do you think is the motivation behind:
 - ▶ Information theft
 - ▶ Destruction of information



Types of access attacks

- ▶ Password attack – dictionary or brute-force
- ▶ Trust exploitation – unauthorized use of privileges
- ▶ Port redirection – compromised system used to attacks other targets
 - ▶ Must have an intrusion tool installed on the system
- ▶ Man-in-the-middle attack
 - ▶ The attacker intercepts all communications between peers
 - ▶ Purpose: to read traffic and/or to alter it
- ▶ Buffer overflow
 - ▶ Sending data to a program beyond its allocated buffer
 - ▶ Valid data gets overwritten – enables other functions

Detecting access attacks

- ▶ Logs – look for failed and repeated logins attempts
 - ▶ Do not allow unlimited failed login attempts => brute-force
- ▶ Unusually high network traffic: Possible MiTM attack
 - ▶ MiTM attacks replicate data
- ▶ High CPU load, program crashes
 - ▶ Possible buffer overflow



Mitigating access attacks

- ▶ The basics: sTr0ng! P4\$\$w0rdz!
- ▶ Strong authentication and encryption make sniffing very little effective
 - ▶ Example: one-time-password (banking)
 - ▶ Vital **business traffic** should be encrypted
 - ▶ Network **management traffic** should be encrypted
- ▶ Switched networks isolate traffic
- ▶ Port scanning can be detected and stopped by IPS
- ▶ Deactivating ICMP prevents ping sweeps
 - ▶ But makes network troubleshooting more difficult



Denial-of-service attacks

- ▶ Send many requests in a short timespan
- ▶ Purpose: to overwhelm the target application or computer and to prevent it from processing normal requests.
- ▶ DoS attacks can crash and slow down processes
- ▶ DDoS = Distributed Denial of Service
 - ▶ Sends many requests from several sources at a time

DoS attacks

- ▶ DoS attacks rely on the fact that servers must maintain state information
 - ▶ That is, servers use memory for each request, until it is completed
- ▶ Servers might not be able to differentiate between legitimate requests and flooded requests.
 - ▶ Hard to avoid
- ▶ Many tools available
 - ▶ Very simple to conduct

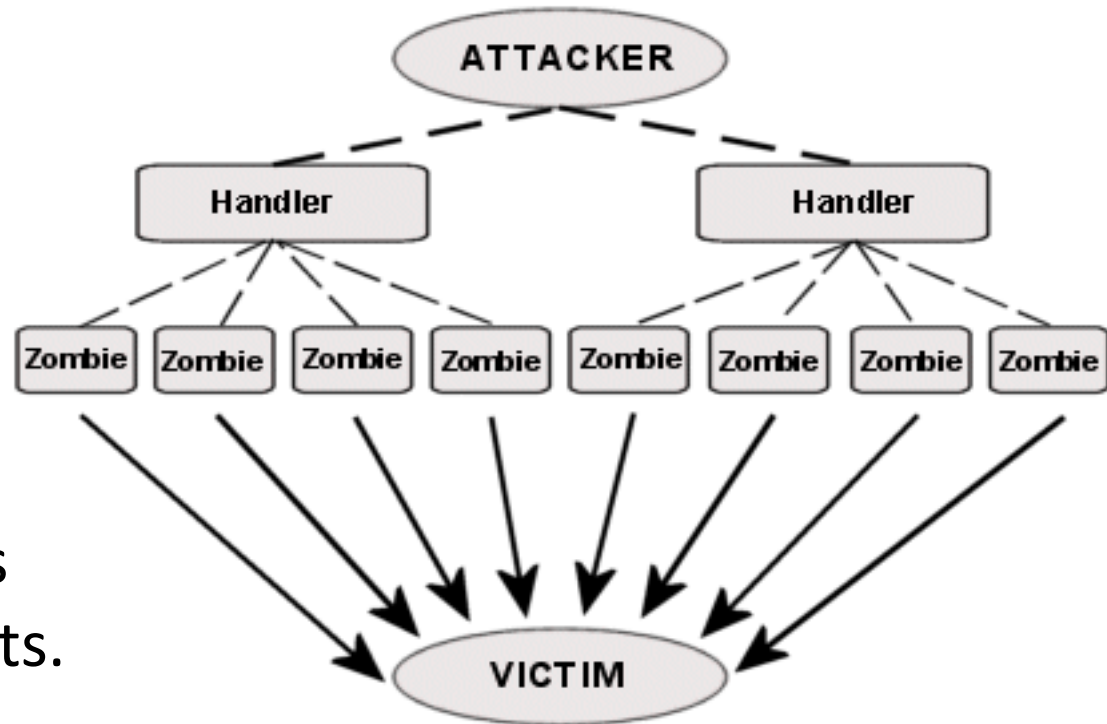
Identifying DoS (and other attacks)

- ▶ Each network **MUST** have a benchmark of:
 - ▶ Total bandwidth utilization
 - ▶ Bandwidth usage per protocol
 - ▶ Protocols active in the network
 - ▶ Hardware load
 - ▶ For hosts
 - ▶ For network devices
 - ▶ Servers
- ▶ All the above measured for different times of the day
- ▶ These statistics can be used to detect **anomalies**
 - ▶ Anomalies can represent attacks



DDoS

Architecture of a DDoS Attack

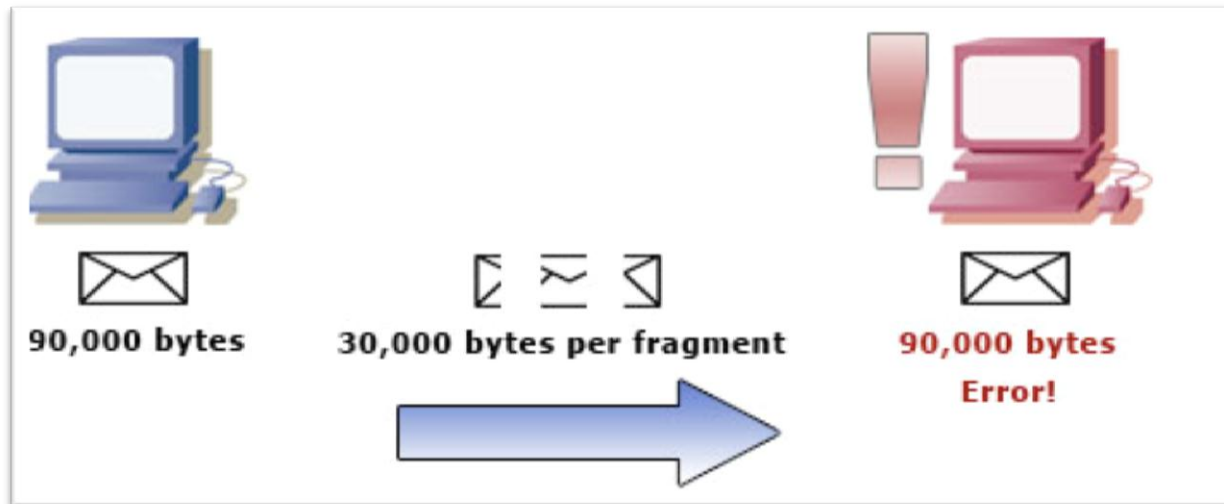


Handlers = “masters”
Zombies = “slaves”

- ▶ Handlers and zombies are compromised hosts.
- ▶ Once started, much harder to stop than a DoS.
 - ▶ Why is it harder?

Types of DoS attacks

- ▶ Ping of death (POD)
 - ▶ 10 years ago
 - ▶ IP packet with an echo request larger than 65535 bytes
 - ▶ It used to crash basically everything: Unix, Linux, Windows, Mac, routers and printers!
 - ▶ They've all been patched up until today.



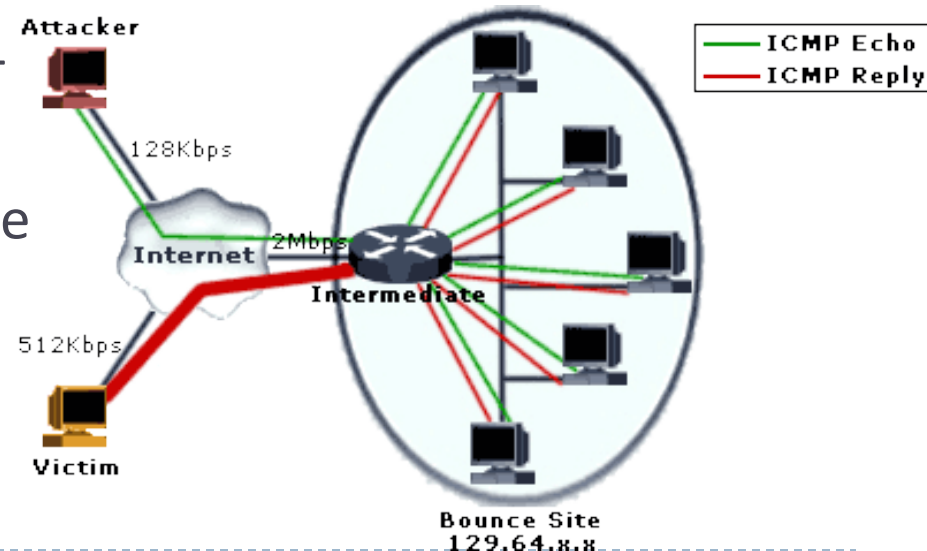
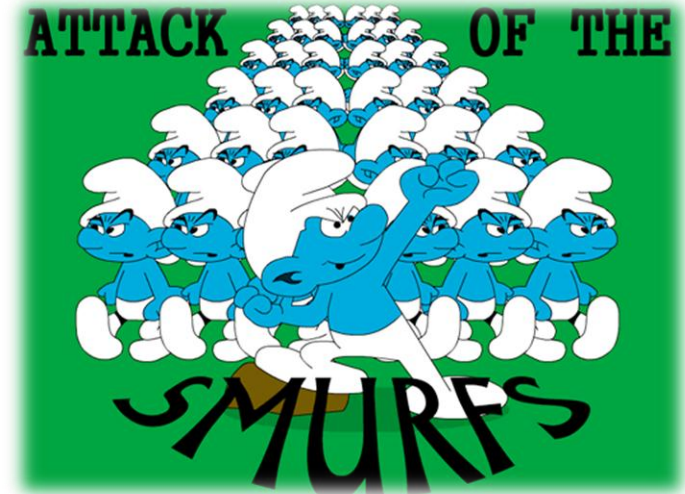
Types of DoS attacks

▶ Smurf attack

- ▶ Large number of ICMP requests (“smurfs” 😊) to a network’s broadcast address.
- ▶ The source of the ICMP packets is spoofed

▶ Result:

- ▶ All hosts reply with ICMP echo-reply packets.
- ▶ The victim – the host having the address that was spoofed
- ▶ Large networks could cause hundreds of hosts to generate traffic.



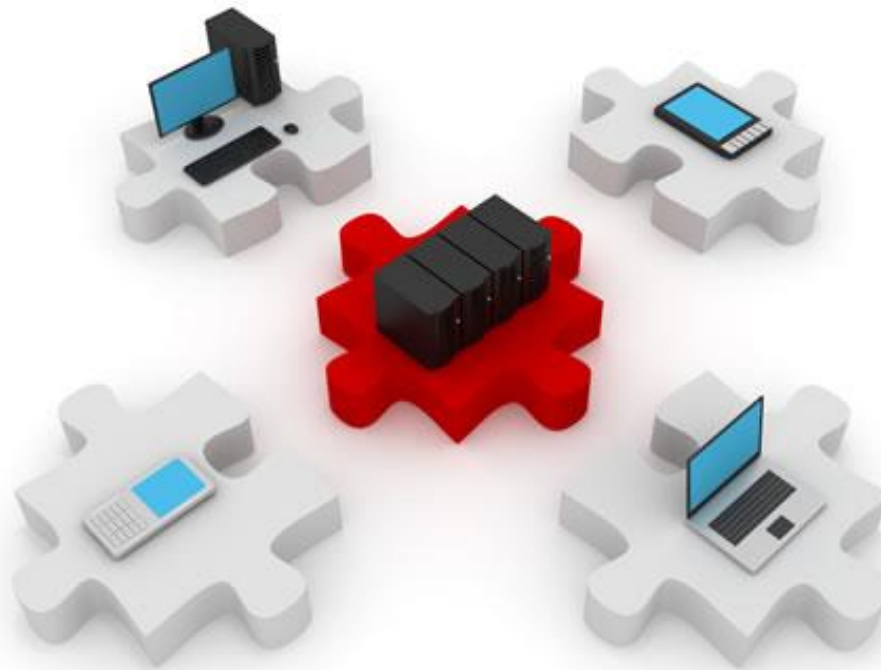
Types of DoS attacks

- ▶ How to avoid smurf attacks?
 - ▶ **Install a trap for the smurfs !!!**
 - ▶ No, in fact it is much simpler than that
 - ▶ Routers must not allow **directed broadcasts**
- ▶ Just to get a hint:
 - `Router(config-if)# no ip directed-broadcast`
 - ▶ And you're done
- ▶ How simple is that? 😊



TCP SYN Flood

- ▶ Sending a large number of TCP SYN packets
- ▶ Each packet is handled like a connection request
- ▶ The server sends back TCP SYN-ACK packets but does not receive responses to complete the three-way handshake
 - ▶ Result: Many half-open TCP connections
 - ▶ The server's connections become saturated
 - ▶ The server cannot respond to legitimate requests
- ▶ Solution: limit the number of half-open connections

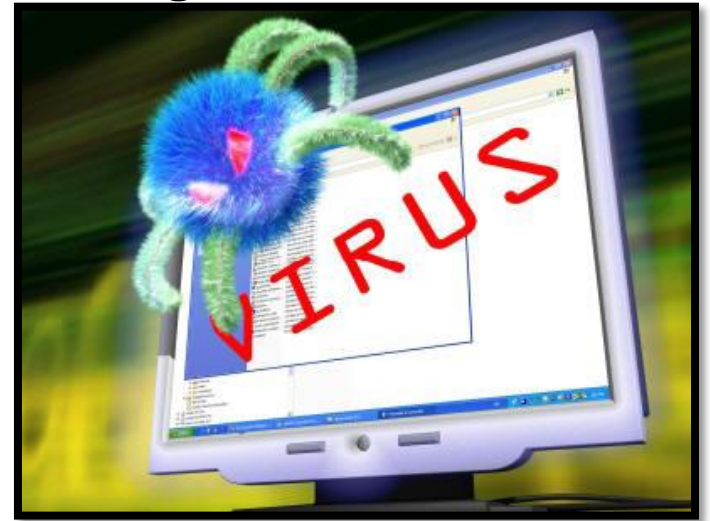


Malicious Software

Viruses, worms, trojans and other species.

Viruses

- ▶ Most harmful type of malware
- ▶ Code attached to legitimate programs
- ▶ Require user interaction with the infected file
- ▶ When activated, can spread to other files
- ▶ Infecting the operating system allows the virus to execute any code, with full administrative privileges
- ▶ Viruses spread by:
 - ▶ USB sticks
 - ▶ Network share
 - ▶ E-mail attachments
 - ▶ Downloaded files



Virus mitigation techniques

- ▶ Updated antivirus software
- ▶ NAC implementation
 - ▶ NAC = Network Access Control
 - ▶ NAC: consider **endpoint security** prior to offering access
 - ▶ When a computer connects, it is completely isolated until it complies with a set of standards:
 - ▶ Valid identity
 - ▶ Anti-virus system
 - ▶ Firewall
 - ▶ System update
 - ▶ Other policies



Worms

- ▶ User interaction not required, unlike viruses
- ▶ Not need to attach to other programs
- ▶ Worms have the ability to run and replicate by themselves on other hosts.
- ▶ Programmed to search for known vulnerabilities.
 - ▶ When found, they are exploited to allow the worm to propagate.



Worm mitigation procedure

▶ Containment

- ▶ Isolate infected parts of the network
- ▶ Contain the worm's spread

▶ Inoculation

- ▶ Patch all uninfected systems
- ▶ Run a deep scan on uninfected systems

▶ Quarantine

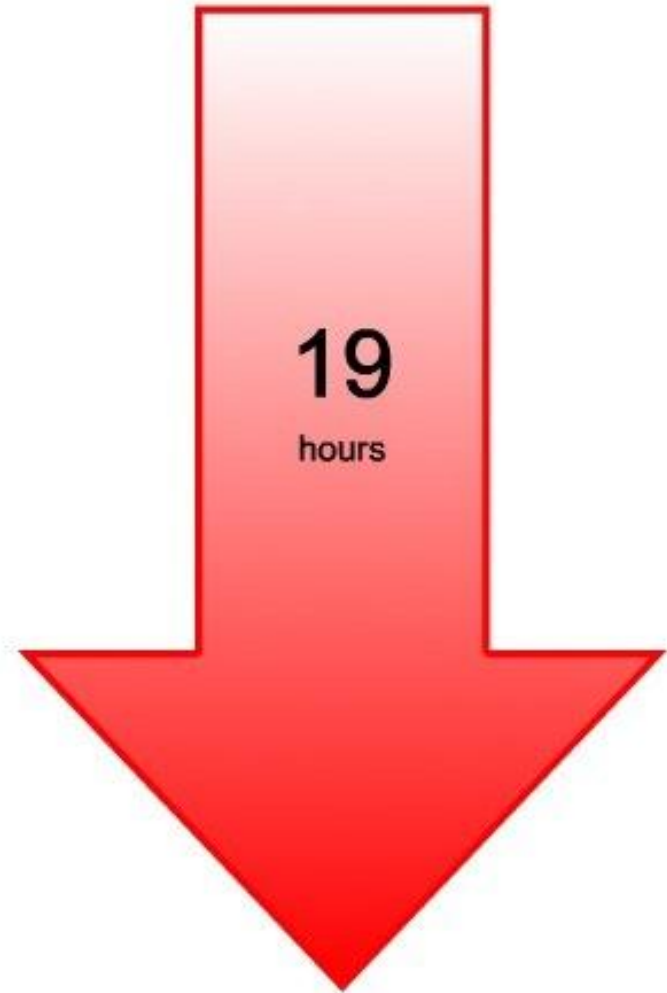
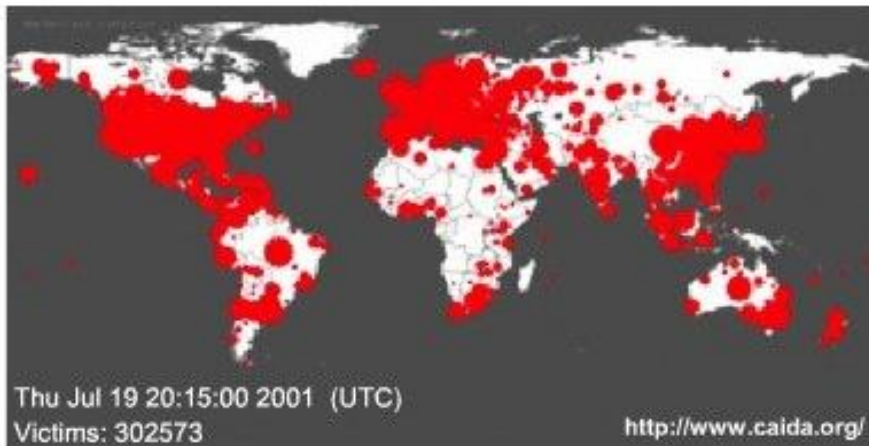
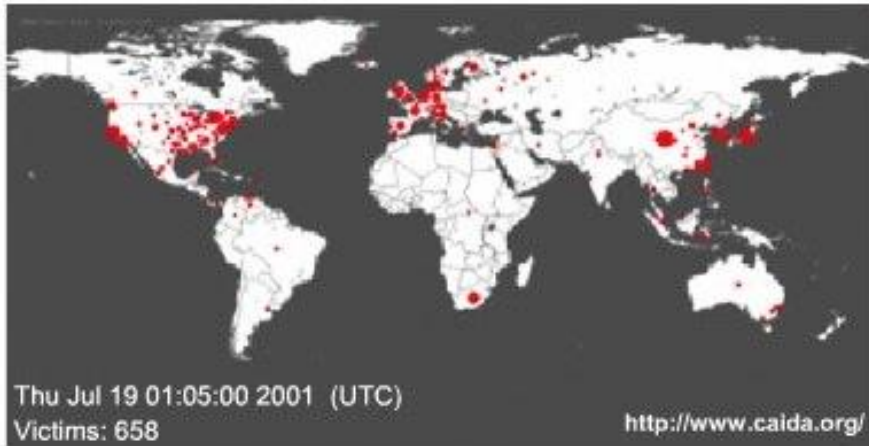
- ▶ Isolate each infected host from the network

▶ Treatment

- ▶ Patch infected systems, if possible
- ▶ Reinstall completely otherwise



Example: SQL Slammer Worm (2001-2003)



Trojans

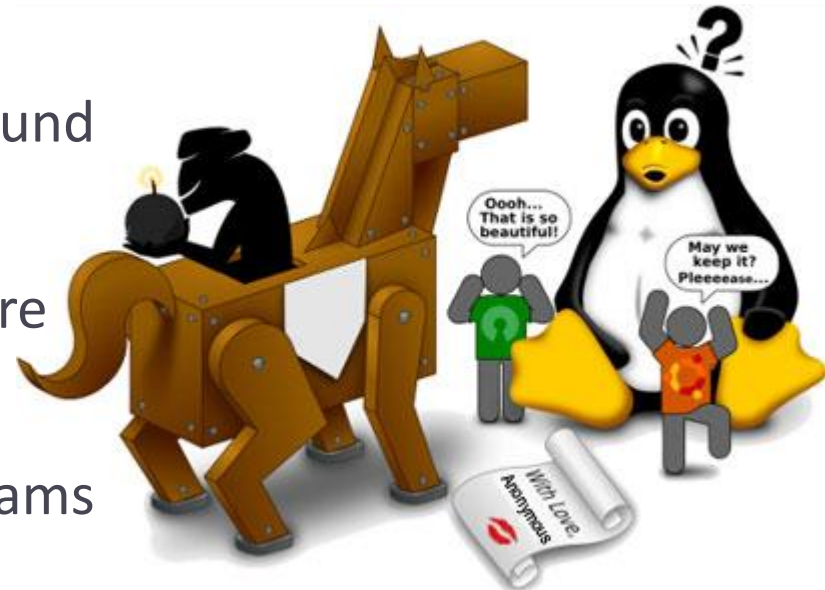
- ▶ Malicious code hidden behind a legitimate function or application.
- ▶ The program executes normally
- ▶ The trojan code runs in the background

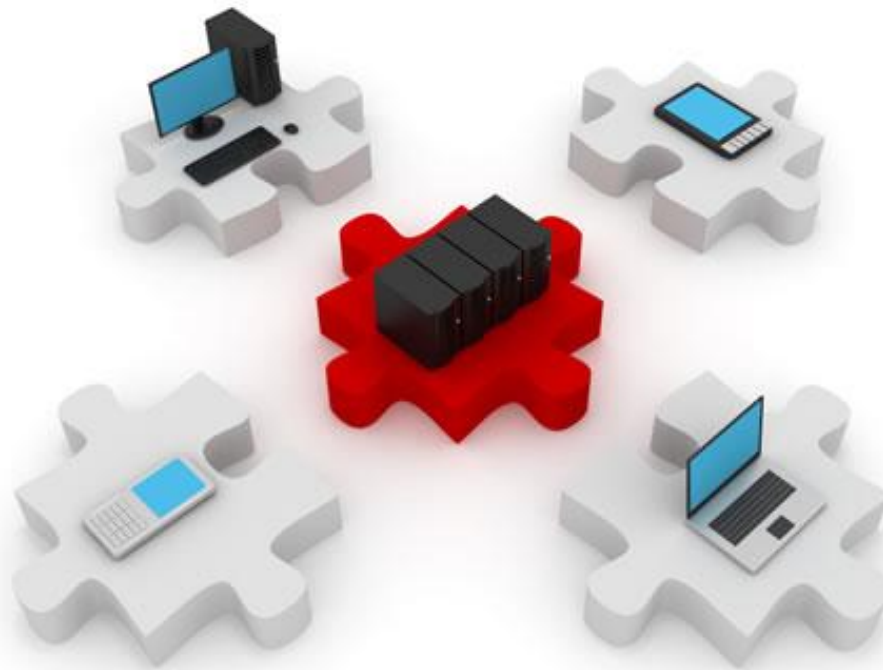
- ▶ Most do not have immediate effect, but open backdoors.
- ▶ Can be designed for specific targets
 - ▶ extremely hard to detect



Types of trojans

- ▶ Remote access trojan
 - ▶ Opens certain ports that provide remote access
- ▶ Data-sending trojan
 - ▶ Gathers information from the computer and sends them to a specific address
- ▶ Proxy trojan
 - ▶ Runs a proxy server in the background
- ▶ Security trojan
 - ▶ Stops antivirus and firewall software
- ▶ Destructive trojan (rare)
 - ▶ Deletes or corrupts files and programs





Hackers

Beginnings

▶ “Phreakers”

- ▶ Started in 1960
- ▶ Exploited switches from telephone companies using tone generators (“blue boxes”), to make long-distance calls
- ▶ Later on, they managed to make their own phone numbers free to call

▶ “Wardialers”

- ▶ Started in 1980, when dial-up modems were introduced
- ▶ Dialed random numbers in search of modems then attempted to break the computer’s password
- ▶ The “ancestor” of today’s ping sweep

History fact:

- ▶ 1972: John Draper, soon to be known as "Captain Crunch," discovers that the plastic whistle in a box of breakfast cereal reproduces a 2600-hertz tone. With a blue box, the whistle unlocks AT&T's phone network, allowing free calls and manipulation of the network.

A "blue box" tone generator



History fact:

- ▶ The first worm was created by... Xerox, in 1979
- ▶ Engineers created a short program that scanned the network for idle processors intending to provide more efficient computer use
- ▶ The scanning and replication mechanism is now used by modern destructive worms

The meaning of “hacker”

Positive	Negative
<ul style="list-style-type: none">• Network professional• User of sophisticated tools• Internet programming skills• Security tester	<ul style="list-style-type: none">• Gains unauthorized access• Targets sensitive data• Attempts to destroy data• Restricts network access• Slows or shuts down services

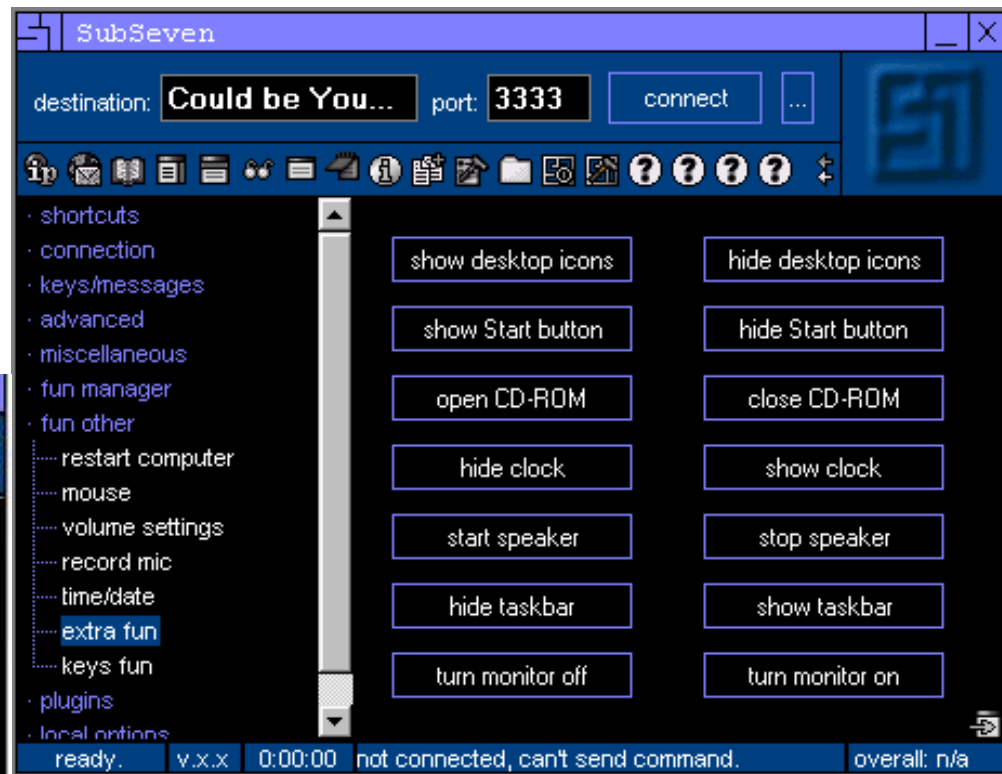
Hacker “flavors”

- ▶ **White hat**
 - ▶ Also known as “ethical hacker”
 - ▶ Breaks for non-malicious reasons, but for testing
 - ▶ Term for “security consultant”
- ▶ **Black hat**
 - ▶ Or “cracker”, illegally breaks computer security
 - ▶ Steals or compromises data
- ▶ **Grey hat**
 - ▶ Middle-ground between the above two
- ▶ **Script kiddie**
 - ▶ Has little understanding of security
 - ▶ Simply uses tools developed by other hackers
- ▶ **Hacktivist**
 - ▶ Hacks only to promote a message: ideological, political, etc.



Tools: Sub7

- ▶ The “classic” script-kiddie tool for many years
- ▶ Client-server application
- ▶ Installs on victim computer and provides access to:
 - ▶ File system
 - ▶ Hardware devices
 - ▶ Operating system
 - ▶ Keylogger
 - ▶ Screen capture



Tools: Project Metasploit

- ▶ Project for exploiting security vulnerabilities
- ▶ Sub-project: Metasploit Framework
 - ▶ Contains a database of several hundreds of known exploits for all operating systems
 - ▶ Tool for developing and executing exploit code on target machines
- ▶ Useful for:
 - ▶ Penetration testing
 - ▶ IDS signature development
 - ▶ Exploit research

METASPLOIT



Tools: Metasploit

msf > show exploits

```
windows/misc/hp_ovtrace HP OpenView Operations OVTrace Buffer Overflow
windows/misc/ib_isc_attach_database Borland InterBase isc_attach_database() Buffer Overflow
windows/misc/ib_isc_create_database Borland InterBase isc_create_database() Buffer Overflow
windows/misc/ib_svc_attach Borland InterBase SVC_attach() Buffer Overflow
windows/misc/landesk_aolnsrvr LANDesk Management Suite 8.7 Alert Service Buffer Overflow
windows/misc/mercury_phonebook Mercury/32 <= v4.01b PH Server Module Buffer Overflow
windows/misc/ms07_064_sami Microsoft DirectX DirectShow SAMI Buffer Overflow
windows/misc/netcat110_nt Netcat v1.10 NT Stack Overflow
windows/misc/shixxnote_font ShixxNOTE 6.net Font Field Overflow
windows/misc/tiny_identd_overflow TinyIdentD 2.2 Stack Overflow
windows/misc/windows_rsh Windows RSH daemon Buffer Overflow
windows/mssql/ms02_039_slammer Microsoft SQL Server Resolution Overflow
windows/mssql/ms02_056_hello Microsoft SQL Server Hello Overflow
windows/mysql/mysql_yassl MySQL yaSSL SSL Hello Message Buffer Overflow
windows/nntp/ms05_030_nntp Microsoft Outlook Express NNTP Response Parsing Buffer Overflow
windows/novell/groupwisemessenger_client Novell GroupWise Messenger Client Buffer Overflow
windows/novell/nmap_stor Novell NetMail <= 3.52d NMAP STOR Buffer Overflow
windows/novell/zenworks_desktop_agent Novell ZENworks 6.5 Desktop/Server Management Overflow
windows/pop3/seattlelab_pass Seattle Lab Mail 5.5 POP3 Buffer Overflow
windows/proxy/bluecoat_winproxy_host Blue Coat WinProxy Host Header Overflow
windows/proxy/ccproxy_telnet_ping CCProxy <= v6.2 Telnet Proxy Ping Overflow
windows/proxy/proxypro_http_get Proxy-Pro Professional GateKeeper 4.7 GET Request Overflow
windows/scada/realwin DATAC RealWin SCADA Server Buffer Overflow
windows/sip/aim_triton_cseq AIM Triton 1.0.4 CSeq Buffer Overflow
```

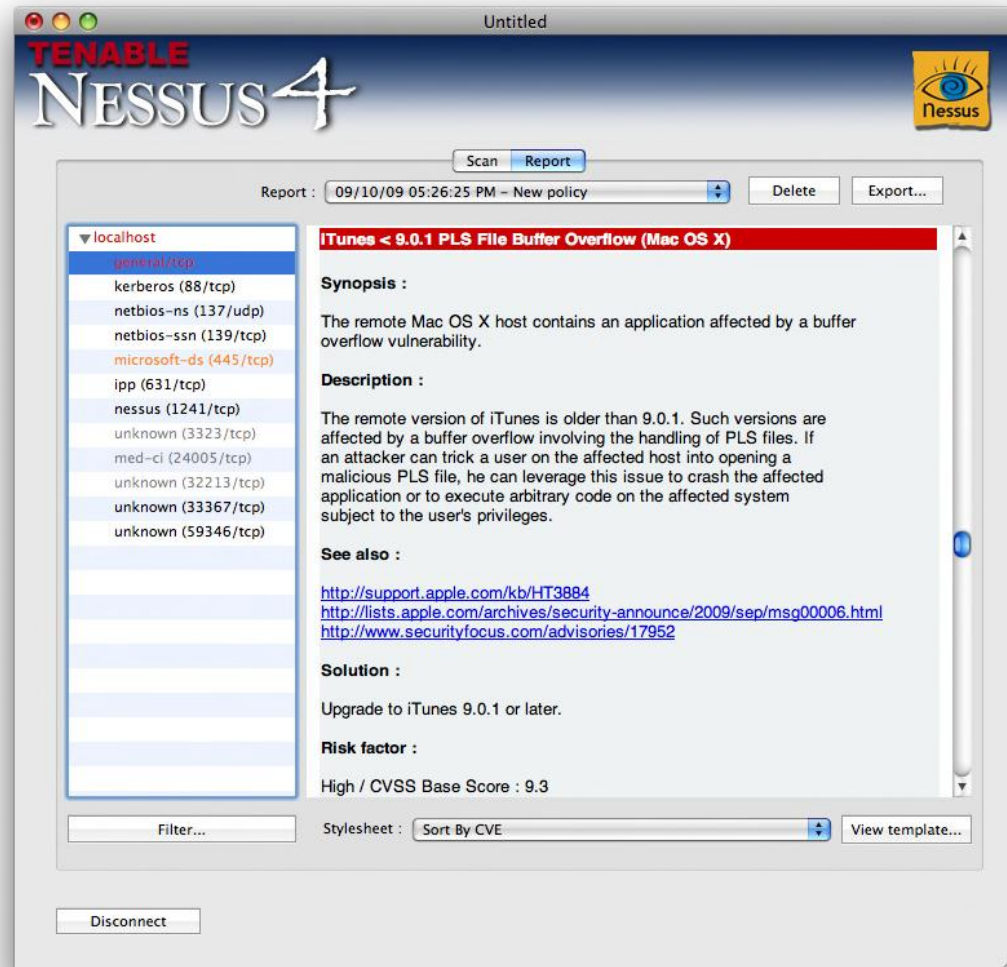
Just a short list of available exploits for Windows systems.

Tools: Nessus

- ▶ Vulnerability scanning tool
- ▶ Client-server application
- ▶ Ability to scan remote hosts
- ▶ Periodic plugin updates



Buffer overflow vulnerability found (iTunes)



The screenshot displays the Nessus 4 web interface. The main window is titled "Untitled" and features the Nessus logo in the top right corner. The interface is divided into several sections:

- Navigation:** "Scan" and "Report" buttons are located at the top center.
- Report Information:** A dropdown menu shows the report name "09/10/09 05:26:25 PM - New policy", with "Delete" and "Export..." buttons to its right.
- Hosts List:** A sidebar on the left lists hosts under "localhost", including "general/tcp", "kerberos (88/tcp)", "netbios-ns (137/udp)", "netbios-ssn (139/tcp)", "microsoft-ds (445/tcp)", "ipp (631/tcp)", "nessus (1241/tcp)", and several "unknown" entries with various ports.
- Vulnerability Details:** The main content area displays a vulnerability titled "iTunes < 9.0.1 PLS File Buffer Overflow (Mac OS X)".
 - Synopsis:** "The remote Mac OS X host contains an application affected by a buffer overflow vulnerability."
 - Description:** "The remote version of iTunes is older than 9.0.1. Such versions are affected by a buffer overflow involving the handling of PLS files. If an attacker can trick a user on the affected host into opening a malicious PLS file, he can leverage this issue to crash the affected application or to execute arbitrary code on the affected system subject to the user's privileges."
 - See also:** Three links are provided: <http://support.apple.com/kb/HT3884>, <http://lists.apple.com/archives/security-announce/2009/sep/msg00006.html>, and <http://www.securityfocus.com/advisories/17952>.
 - Solution:** "Upgrade to iTunes 9.0.1 or later."
 - Risk factor:** "High / CVSS Base Score : 9.3"
- Footer:** A "Filter..." input field, a "Stylesheet:" dropdown set to "Sort By CVE", and a "View template..." button are located at the bottom of the main content area. A "Disconnect" button is positioned at the bottom left of the window.

Something to laugh about on your way home...

“Nowadays, security guys break the Mac every single day. Every single day, they come out with a total exploit, your machine can be taken over totally. I dare anybody to do that once a month on the Windows machine.”

Bill Gates (2007)

