

TCP/IP

Security Attacks





1. TCP Segment Format, Connection Setup, Disconnect
2. IP: Address Spoofing, Covert Channel, Fragment Attacks, ARP, DNS
3. TCP Flags: Syn Flood, Ping of Death, Smurf, Fin
4. UDP Flood Attack
5. Connection Hijacking
6. Application: E-Mail, Web spoofing

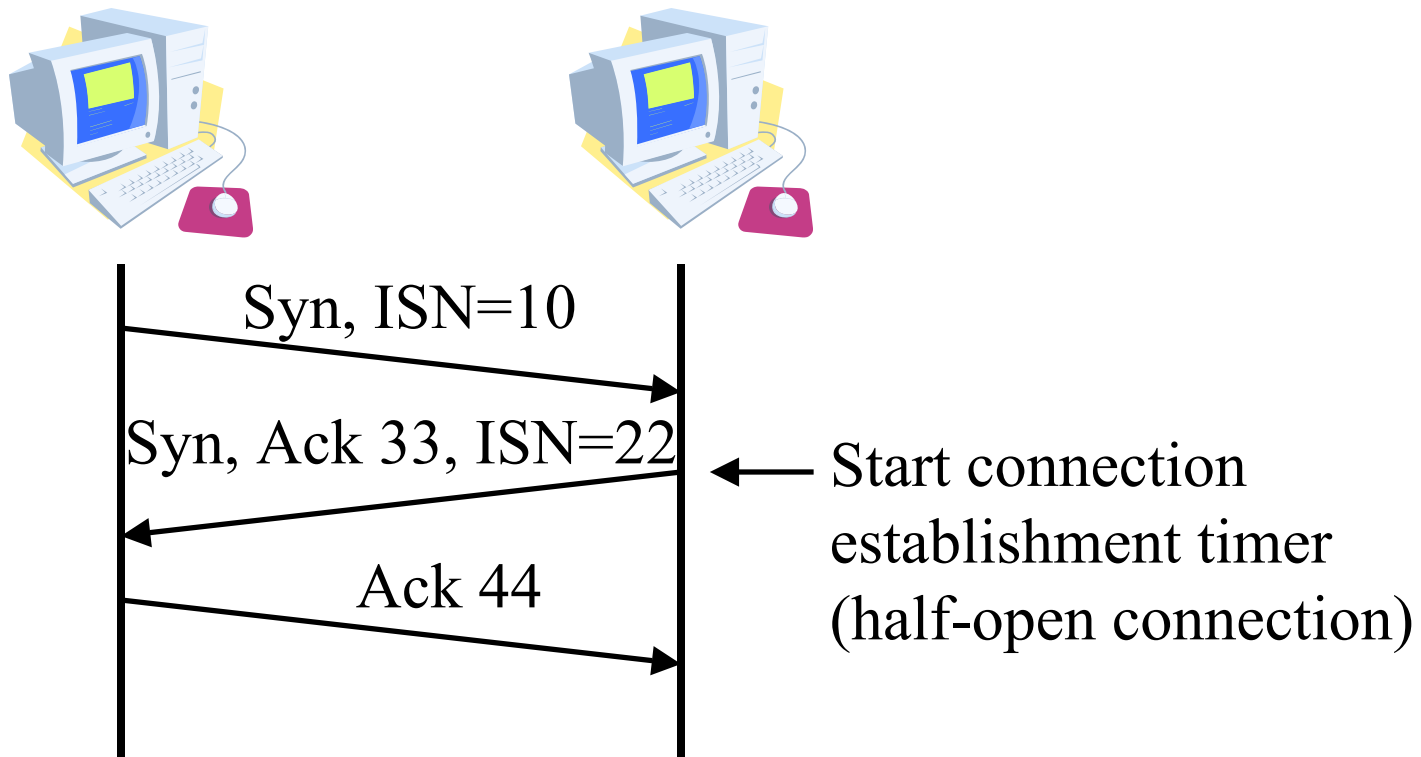
TCP Segment Format

Source Port				Destination Port				
Sequence Number								
Ack Number								
Data Offset	Res	Urg	Ack	Push	Reset	Syn	Fin	Window
Checksum				Urgent Pointer				
Options							Padding	
Data								

- ❑ Urgent: Deliver immediately at destination
- ❑ Push: Leave source immediately
- ❑ First data byte is ISN+1. Ack is next byte expected.
Expecting Ack to Ack+window-1 next.

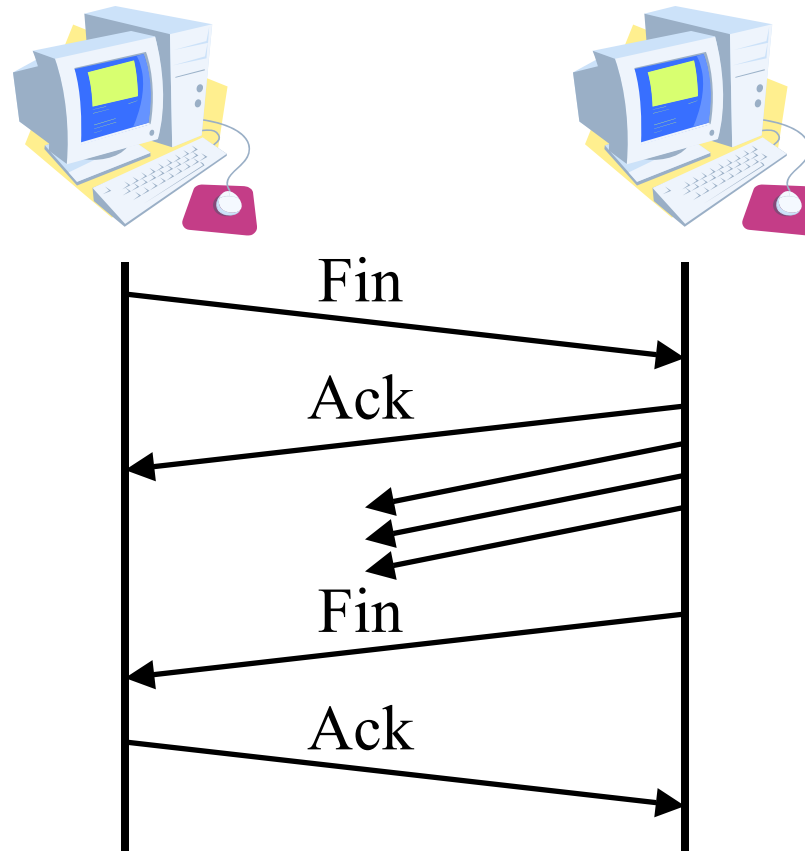
TCP Connection Setup

❑ Three way handshake



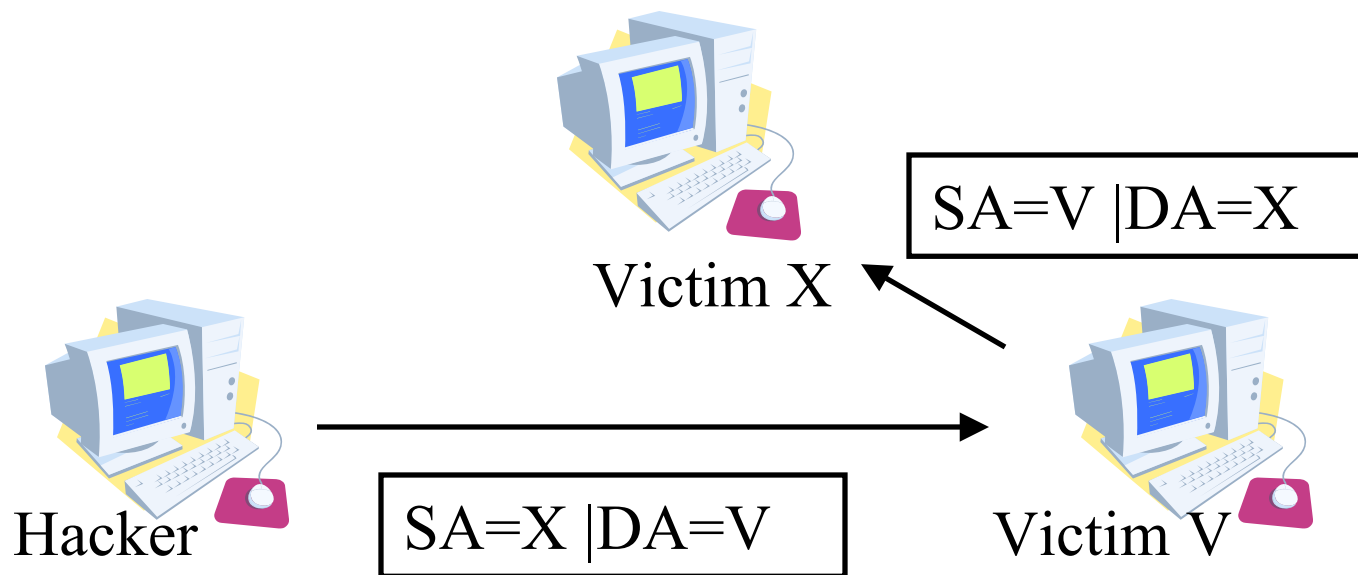
TCP Disconnection

- ❑ Fin \Rightarrow No more data. Connection can be closed.
- ❑ Four-way handshake



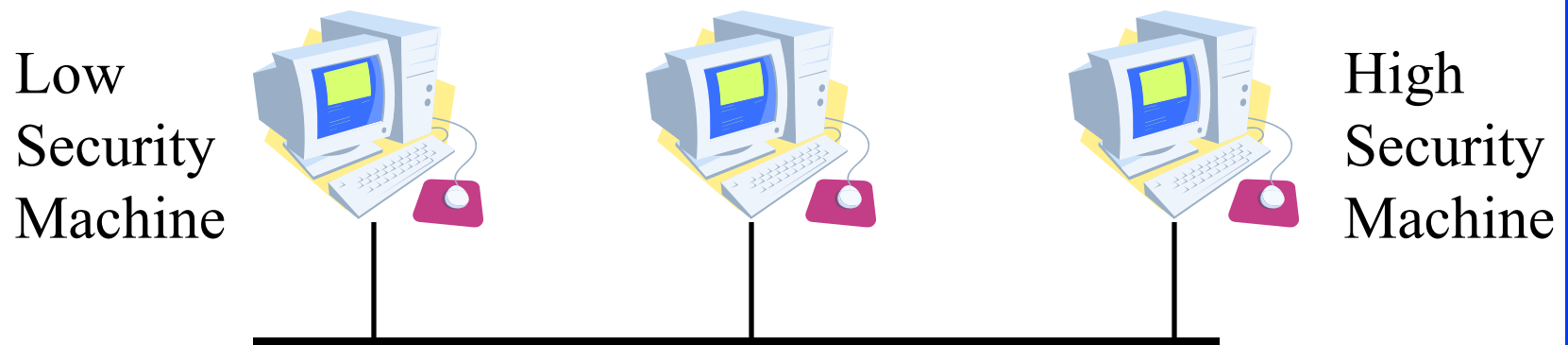
IP Address Spoofing

- Send requests to server with someone X's IP address. The response is received at X and discarded. Both X and server can be kept busy \Rightarrow DoS attack



Covert Channel

- ❑ **Loki** - a client server application,
 - Uses ICMP echo to send covert commands
 - <http://xforce.iss.net/> <https://exchange.xforce.ibmcloud.com/>
- ❑ **Timing Channel** - CPU load indicates a 0 or 1
(Two processes on the same machine)
- ❑ **Storage Channel** - Print queue length large = 1, small=0



IP Fragment Attacks

- ❑ Fragments can overlap
- ❑ Final packets can be too large

These attacks are carried out in two ways,

first, the attacker sends out fraudulent packets larger than the MTU can handle; these packets are forgeries and in some cases cannot be reassembled by the receiving network leading to network overload and a denial of service condition.

The second way data fragmentation attacks are carried out is by the attacker targeting the IP assembly systems, preventing the network from putting the packets back together by sending duplicate fragments, unsequenced fragments, and fragments with reassembly instructions forged erroneously; eventually receiving server is again overloaded and a denial of service condition follows.

TCP Flags

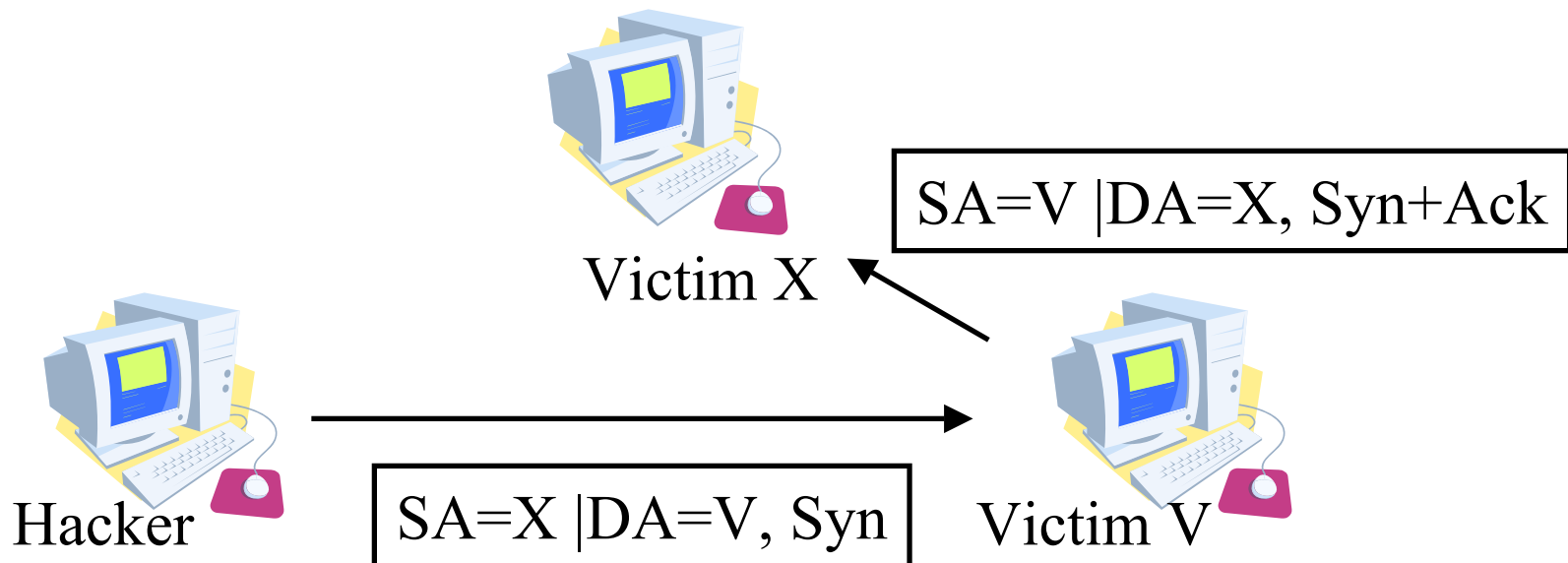
❑ Invalid Combinations

Syn	Fin	Psh	Rst
1	1	0	0
1	1	1	0
1	1	0	1
1	1	1	1

❑ May cause recipient to crash or hang

Syn Flood

- ❑ A sends Syn request with IP address of X to Server V.
- ❑ V sends a syn+ack to X
- ❑ X discards syn+ack leaving an half open connection at V.
- ❑ Many open connections exhausts resources at V \Rightarrow DoS



Ping of Death

- ❑ Send a ping with more than 64kB in the data field.
- ❑ Most systems would crash, hang or reboot.

Ping of Death (a.k.a. PoD) is a type of Denial of Service (DoS) attack in which an attacker attempts to crash, destabilize, or freeze the targeted computer or service by sending malformed or oversized packets using a simple ping command.

Smurf

- ❑ Send a broadcast echo request with the V's source address.
- ❑ All the echo replies will make V very busy.

The steps in a Smurf attack are as follows:

First, the malware creates a network packet attached to a false IP address — a technique known as "spoofing."

Inside the packet is an ICMP ping message, asking network nodes that receive the packet to send back a reply

These replies, or "echoes," are then sent back to network IP addresses again, setting up an infinite loop.

When combined with IP broadcasting — which sends the malicious packet to every IP address in a network — the Smurf attack can quickly cause a complete denial of service.

How to Protect Yourself from Smurf Attack

The Smurf Attack sounds cute but poses real risks if servers are overwhelmed. Disabled IP broadcasting and reliable detection tools help limit the chance and impact of this attack. Here are a couple of steps to for Smurf attack mitigation:

- make sure to block directed broadcast traffic coming into the network
- configure hosts and routers not to respond to ICMP echo requests.

A variation to the Smurf attack is the **Fraggle attack**.

→ The attack is essentially the same as the Smurf attack but instead of sending an ICMP echo request to the direct broadcast address, it sends UDP packets. For the Fraggle attack, it is the same mitigation process.

Fin

- ❑ In the middle of conversation between X and V.
- ❑ H sends a packet with Fin flag to V.
- ❑ V closes the connection and disregards all further packets from X.
- ❑ RST flag can be used similarly

An adversary uses a TCP FIN scan to determine if ports are closed on the target machine. This scan type is accomplished by sending TCP segments with the FIN bit set in the packet header. The RFC 793 expected behavior is that any TCP segment with an out-of-state Flag sent to an open port is discarded, whereas segments with out-of-state flags sent to closed ports should be handled with a RST in response. This behavior should allow the adversary to scan for closed ports by sending certain types of rule-breaking packets (out of sync or disallowed by the TCB) and detect closed ports via RST packets

Fin

In addition to its relative speed in comparison with other types of scans, the major advantage a TCP FIN Scan is its ability to scan through stateless firewall or ACL filters.

Such filters are configured to block access to ports usually by preventing SYN packets, thus stopping any attempt to 'build' a connection.

FIN packets, like out-of-state ACK packets, tend to pass through such devices undetected.

FIN scanning is still relatively stealthy as the packets tend to blend in with the background noise on a network link.

UDP Flood Attack

- ☐ An adversary may execute a flooding attack using the UDP protocol with the intent to deny legitimate users access to a service by consuming the available network bandwidth.
- ☐ Additionally, firewalls often open a port for each UDP connection destined for a service with an open UDP port, meaning the firewalls in essence save the connection state thus the high packet nature of a UDP flood can also overwhelm resources allocated to the firewall.
- ☐ UDP attacks can also target services like DNS or VoIP which utilize these protocols. Additionally, due to the session-less nature of the UDP protocol, the source of a packet is easily spoofed making it difficult to find the source of the attack.

UDP Flood Attack

On most Unix-like operating systems, a CHARGEN server is built into the inetd or xinetd daemon. The CHARGEN service is usually not enabled by default. It may be enabled by adding the following lines to the file /etc/inetd.conf and telling inetd to reload its configuration:

```
chargen stream tcp  nowait root  internal
chargen dgram  udp   wait  root  internal
```

The CHARGEN service may be used as a source of a byte-stream for debugging TCP network code for proper bounds checking and buffer management. It may also be a source of generic payload for bandwidth measurement and/or QoS fine-tuning

UDP Flood Attack

- ❑ Character Generator (Chargen) request results in a response with random characters being returned.
- ❑ Used to diagnose lost packets on the path between two hosts.
- ❑ Uses TCP/UDP port 19.
- ❑ H can send a chargen request from X to V.
- ❑ V can respond to X wasting their bandwidth.

chargen example

```
$ telnet localhost chargen
```

```
Trying 127.0.0.1...
```

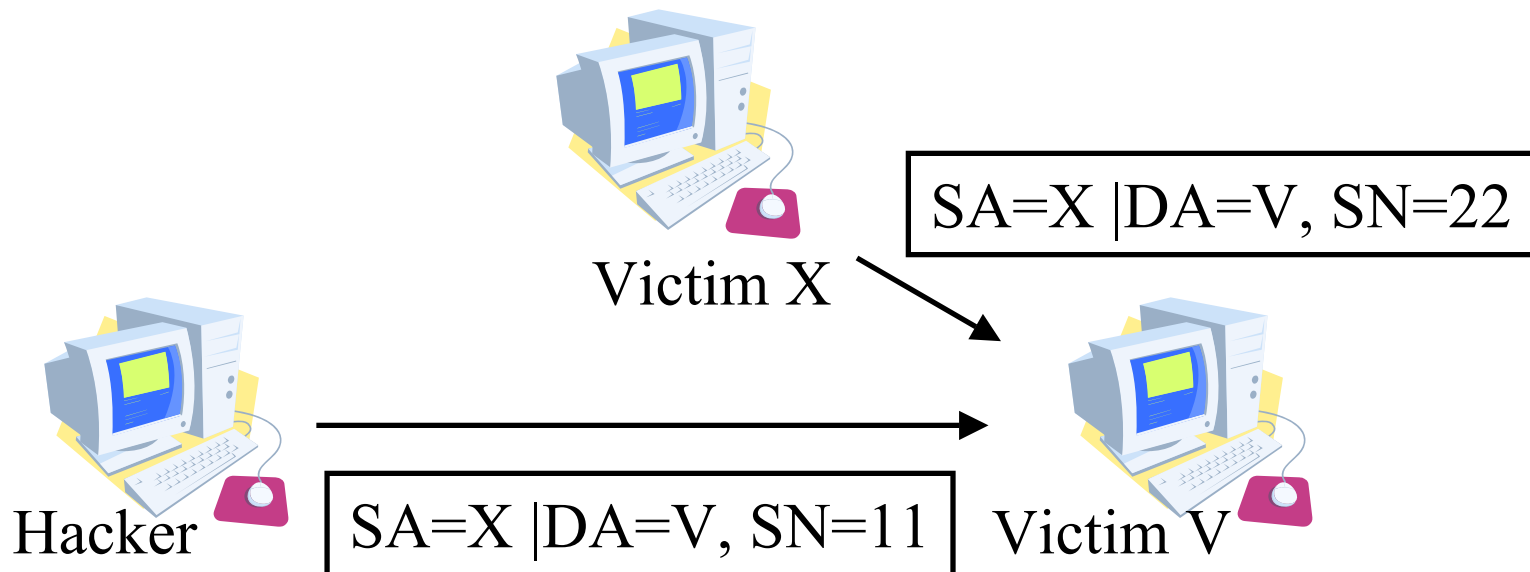
```
Connected to localhost.
```

```
Escape character is '^['.
```

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg  
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefgh  
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghi  
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghij  
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijk  
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijkl  
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklm  
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmn  
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmno
```

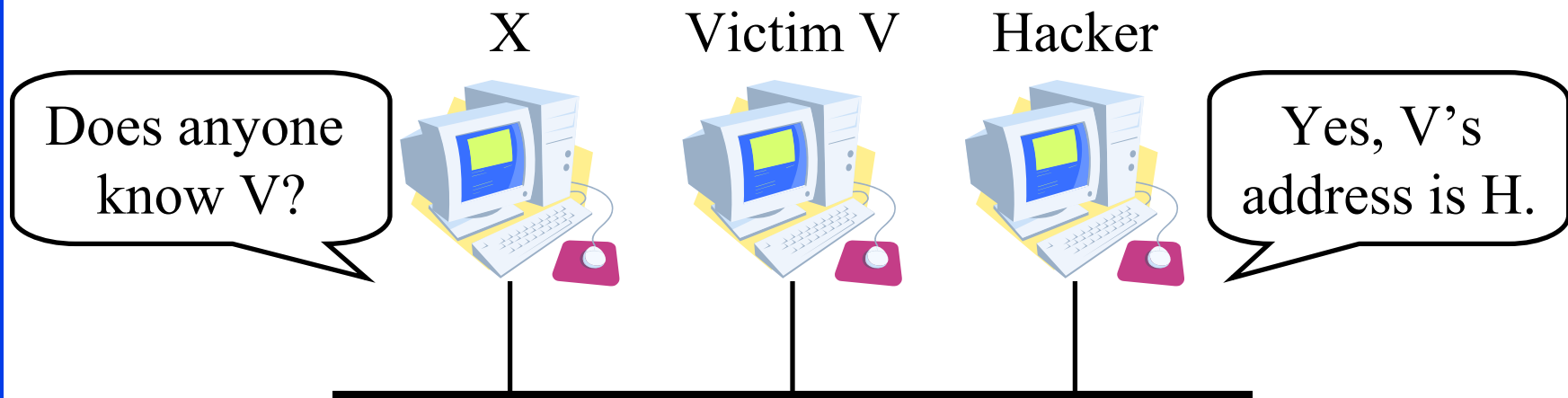
Connection Hijacking

- ❑ H sends packets to server X which increments the sequence number at X.
- ❑ All further packets from V are discarded at X.
- ❑ Responses for packets from H are sent to V - confusing him.

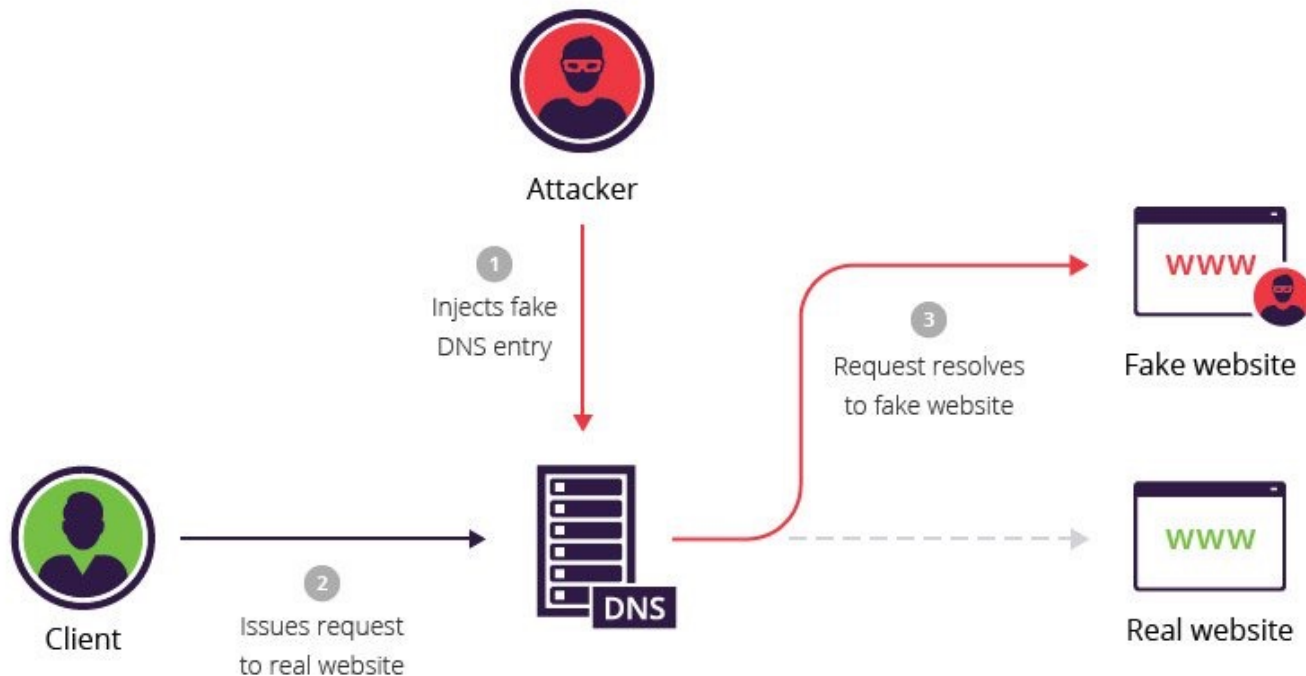


ARP Spoofing

- ❑ X tries to find the MAC address of Victim V
- ❑ Hacker H responds to ARP request pretending to be V.
- ❑ All communication for V is captured by H.
- ❑ Countermeasure: Use static ARP



DNS cache poisoning example



- The following example illustrates a DNS cache poisoning attack, in which an [attacker](#) (IP 192.168.3.300) intercepts a communication channel between a client (IP 192.168.1.100) and a server computer belonging to the website `www.estores.com` (IP 192.168.2.200).
- In this scenario, a tool (e.g., `arp spoof`) is used to dupe the client into thinking that the server IP is 192.168.3.300. At the same time, the server is made to think that the client's IP is also 192.168.3.300.

DNS cache poisoning example

- 1- The attacker uses arpspoof to issue the command: arpspoof 192.168.1.100 192.168.2.200. This modifies the MAC addresses in the server's ARP table, causing it to think that the attacker's computer belongs to the client.**
- 2- The attacker once again uses arpspoof to issue the command: arpspoof 192.168.2.200 192.168.1.100, which tells the client that the perpetrator's computer is the server.**
- 3- The attacker issues the Linux command: echo 1 > /proc/sys/net/ipv4/ip_forward. As a result, IP packets sent between the client and server are forwarded to the perpetrator's computer.**
- 4- The host file, 192.168.3.300 estores.com is created on the attacker's local computer, which maps the website www.estores.com to their local IP.**
- 5- The perpetrator sets up a web server on the local computer's IP and creates a fake website made to resemble www.estores.com.**
- 6- Finally, a tool (e.g., dnsspoof) is used to direct all DNS requests to the perpetrator's local host file. The fake website is displayed to users as a result and, only by interacting with the site, malware is installed on their computers.**

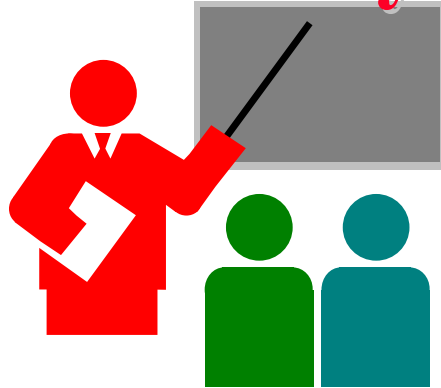
E-Mail Spoofing

- ❑ From address is spoofed.
- ❑ Malware attachment comes from a friendly address.
- ❑ From: God@heavens.com

Web Spoofing

- ❑ The web site looks like another
- ❑ Southwest Airline,
<http://airlines.ws/southwest-airline.htm>
- ❑ For every .gov site there is a .com, .net giving similar information
- ❑ For misspellings of popular businesses, there are web sites.

Summary



1. TCP port numbers, Sequence numbers, ack, flags
2. IP addresses are easy to spoof.
ARP and DNS are not secure.
3. Flags: Syn Flood, Ping of Death, Smurf, Fin,
Connection Hijacking
4. UDP Flood Attack
5. Application addresses are not secure

References

1. Gert De Laet and Gert Schauwers, “Network Security Fundamentals,” Cisco Press, 2005, ISBN:1587051672

Lab Homework 3

- ❑ This lab consists of using the following tools:
- ❑ XP Keylogger,
<http://www.bestvistadownloads.com/download/t-free-xp-keylogger-download-zhtdqdn.html>
- ❑ Snort, vulnerability scanner, <http://www.codecraft-canada.com/Snort/>
- ❑ Password dump, Pwdump3,
<http://www.openwall.com/passwords/microsoft-windows-nt-2000-xp-2003>
- ❑ John the ripper, Brute force password attack,
<http://www.openwall.com/john/>

Lab Homework 3 (Cont)

- ❑ If you have two computers, you can install these programs on one computer and conduct these exercises.
- ❑ Alternately, you can remote desktop to CSE571XPC and conduct exercises 1-4 and then remote desktop to CSE571XPS and conduct exercise 5.
- ❑ Use your last name (with spaces removed) as your user name.

1. Keylogger

- ❑ Delete all previous log files, if any,
e:\program files\xp keylogger\logs*.*
- ❑ Dstart xp keylogger
- ❑ Browse to www.google.com and search for your name
- ❑ Dtop keylogger
- ❑ CD to e:\program files\xp keylogger\logs\
- ❑ Open the htm file in the browser
- ❑ Notedown the texts shows there on a paper and submit.
- ❑ Delete the log e:\program files\xp keylogger\logs*.*

2. Snort

- ☐ Delete all the previous logs, if any, `e:\snort\log\new*.*`
- ☐ Start snort
- ☐ Go back to your machine
- ☐ Run `smbdie` to attack CSE571XPC
- ☐ When the program stops, connect back to CSE571XPC
- ☐ Use control-C to stop snort
- ☐ Type the log file `e:\snort\log\new\alert.ids`
- ☐ Count how many time `smbdie` is mentioned.
- ☐ Delete the logs, `e:\snort\log\new*.*`

3. PWDump3

- ❑ Goal: Get the password hash from the server CSE571XPS
- ❑ On CSE571XPC, open a dos box
- ❑ CD to e:\pwdump3
- ❑ Run pwdump3 without parameters for help
- ❑ Run pwdump3 with parameters to get the hash file from server CSE571XPS
- ❑ You will need the administrator account and password supplied in the class.
- ❑ Open the hashfile obtained in notepad. Delete all lines except the one with your last name.
- ❑ Save the file as e:\johntheripper\<your_last_name>.txt

4. Find your password

- ☐ On CSE571XPC, use the command box
- ☐ CD to e:\johntheripper
- ☐ Delete john.pot
- ☐ Run johntheripper without parameters to get help
- ☐ Run johntheripper with the file you created in step 3
- ☐ This will tell you your password, write it down on a paper to submit with the homework.
- ☐ Close your remote desktop session.

5. Change your password

- ❑ Now remote desktop to CSE571XPS
- ❑ Use your last name as username and the password you obtained in step 4.
- ❑ Go to start → Control panel → Computer management → Local users and groups → Users
- ❑ Select your name, right click and change your password to a strong password.
- ❑ Note the time and date you change the password. Submit the time as homework answer.
- ❑ Logout

Computer Sharing Rules

- ❑ One client and one server to be shared among all the students of the class.
- ❑ Time slotted system with each slot of 1 hour starting at 00:00AM.
- ❑ You can use one slot or part of one slot and must disconnect at the end of the slot time.
- ❑ You can come back after 15 minutes, if no one has connected, you can use the remainder of the next slot.
- ❑ You can repeat this 15 minute break + 45 minute work cycle as long as needed.
- ❑ Do your exercise early, do not wait till the last day.