Firewalk: Determining Firewall Rules
Find out firewall rules for new connections
We don’t care about target machine, just about packet types that can get through the firewall
Find out distance to firewall using traceroute
Ping arbitrary destination setting TTL=distance+1
If you receive ICMP_Time_Exceeded packet went through

Defenses Against Firewalking
Filter out outgoing ICMP traffic
Use firewall proxies

Vulnerability Scanning
The attacker knows OS and applications installed on live hosts
He can now find for each combination
Vulnerability exploits
Common configuration errors
Default configuration
Vulnerability scanning tool uses a database of known vulnerabilities to formulate packets and send them to hosts
Vulnerability scanning is also used for sysadmin

Vulnerability Scanning Tools
SARA
http://www-arc.com/sara
SAINT
http://www.wwdsi.com/saint
VLAD
http://razor.bindview.com/tools
Nessus
http://www.nessus.org

Defenses Against Vulnerability Scanning
Close your ports and keep systems patched
Find your vulnerabilities before the attackers do

Phase 3: Gaining Access
Exploit vulnerabilities
Exploits for a specific vulnerability can be downloaded from hacker sites
Skilled hackers write new exploits
Stack-based Overflow Attacks

Stack stores important data on procedure call

```
void sample_function(char* s)
{
    char buffer[10];
    strcpy(buffer, string);
    return;
}
```

And a main program
```
void main()
{
    int i;
    char buffer[200];
    for(i=0; i<200;i++) buffer[i]='A';
sample_function(buffer);
    return;
}
```

Argument is larger than we expected

Large input will be stored on the stack, overwriting information
```
void sample_function(char* s)
{
    char buffer[10];
    strcpy(buffer, string);
    return;
}
```

And a main program
```
void main()
{
    int i;
    char buffer[200];
    for(i=0; i<200;i++) buffer[i]='A';
sample_function(buffer);
    return;
}
```

Attacker overwrites return address to point somewhere else
‘Local variables’ portion of the stack
Places attack code in machine language at that portion
Since it is difficult to know exact address of the portion, pads attack code with NOPs before and after

IDS could look for sequence of NOPs to spot buffer overflows
Attacker uses polymorphism: he transforms the code so that NOP is changed into some other command that does the same thing, e.g. MV R1, R1
Attacker XORs important commands with a key
Attacker places XOR command and the key just before the encrypted attack code, for decryption XOR command is also obscured

What type of commands does the attacker execute?
Commands that help him gain access to the machine
Writes a string into inetd.conf file to start shell application listening on a port, then uses Netcat to make raw interactive connection to the port
Starts TFTP to transfer Netcat onto the victim, then accesses it
Starts Xterm
Stack-based Overflow Attacks

How does an attacker discover stack-based overflow?

- Looks at the source code
- Runs application on his machine, tries to supply long inputs and looks at system registers

Read more at

http://packetstormsecurity.nl/docs/hack/smashstack.txt

Defenses Against Stack-based Overflow

For system administrators:

- Apply patches, keep systems up-to-date
- Disable execution from the stack
- Monitor writes on the stack
- Store return address somewhere else
- Monitor outgoing traffic

For software designers:

- Apply checks for buffer overflows
- Use safe functions

Password Attacks

Attacker attempts to login with some known username, and to guess a password

- Trying dictionary words
- Trying combinations of dictionary words
- Performing brute-force search

Attacker steals encrypted or hashed password file and tries to decrypt it

Defenses Against Password Attacks

Make strong passwords

- Think of a phrase, take first letters, mix big caps and special characters
- Use password filtering software
- Use strong encryption/hash techniques

Web Application Attacks

Account harvesting

- Gather usernames by observing error messages, then try to guess passwords
- Defense: use same error messages for everything

Hijack a session ID

- Observe session ID and how it changes between sessions
- Change your session ID to another one
- Defense: digitally sign or hash session ID, make them long enough and apply timestamps

SQL Piggybacking

Malformed input into Web form may trigger informative message from an SQL server

Input: 111111111'
Error in SQL syntax near `111111111' at line 1
SELECT * FROM account WHERE (userid='10001'
and number='111111111''

Attacker then adds SQL commands into input

Input: 111111111'+or+userid%3d'10002
SELECT * FROM account WHERE (userid='10001'
and number='111111111' or userid='10002'

Defense: filter user input
Gaining Access Using Network Attacks

- Sniffing for passwords and usernames
- Spoofing addresses
- Hijacking a session

Sniffing

- Looking at raw packet information on the wire
- Some media is more prone to sniffing – Ethernet
- Some network topologies are more prone to sniffing – hub vs. switch

Sniffing on a Hub

- Ethernet is a broadcast media – every machine connected to it can hear all the information
- Passive sniffing

Sniffing on a Switch

- Switch is connected by a separate physical line to every machine and it chooses only one line to send the message

Sniffing on a Switch – Take 1

- Attacker sends a lot of ARP messages for fake addresses to R
- Some switches send on all interfaces when their table overloads
Address Resolution Protocol (ARP) maps IP addresses with MAC addresses.

1. Who has X?
2. I do
3. I have R, tell Y
4. For X, MAC (R)
5. For X, MAC (A)
6. For X, MAC (X)

Attacker uses ARP poisoning to map his MAC address to IP address X.

1. Remember MAC(X)
2. Set up IP forwarding for X to R

Active Sniffing Tools
- Dsniff
  - http://www.monkey.org/~dugsong/dsniff
  - Also parses application packets for a lot of applications
  - Sniffs and spoofs DNS

Spoofing DNS
- Attacker sniffs DNS requests, replies with his own address faster than real server
- When real reply arrives client ignores
- This can be coupled with man-in-the-middle attack on HTTPS and SSH

Sniffing Defenses
- Use end-to-end encryption
- Use switches
  - Statically configure MAC and IP bindings with ports
- Don’t accept suspicious certificates

IP Address Spoofing
- Attacker cannot see reply packets
- He sees the request, but not the reply.
Guessing a Sequence Number

It used to be ISN=f(Time), still is in Windows.

On Linux ISN=f(time)+rand

On BSD ISN=rand

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

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`

### 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 backdoor 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


### Trojan Defenses

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

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/](http://www.tripwire.org/)

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

Phase 5: Covering Tracks

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

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
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

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

Covert Channels

Transfer data across the network in unsuspicious 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)

Defenses Against Covert Channels

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