Lab 3: Recon and Firewalls
Before we can create firewall rules, we have to know the basics of network protocols. Here's a quick review...

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>The underlying packet delivery protocol</td>
</tr>
<tr>
<td>ICMP</td>
<td>Error messages, network book keeping and more</td>
</tr>
<tr>
<td>UDP</td>
<td>Connectionless and unreliable</td>
</tr>
<tr>
<td>TCP</td>
<td>Connections, reliable and sequenced</td>
</tr>
</tbody>
</table>
TCP 3-Way Handshake

- TCP requires that both computers track the state of all connections. Here's how TCP connections are established. This is important to know when writing firewall rules.
Our Playground Network

Internet

Attacker 172.16.20.20

Firewall 172.16.20.30
  172.16.10.254
  172.16.10.10

Victim 172.16.10.10
NMap

Nmap is one of the most commonly used network scanners used today.

It's commonly used for:
- Mapping hosts on networks
- Identifying a host's OS
- Locating and identifying network services
Here's a common way that nmap is used to probe a network:

```
nmap -A -sS -sR -p1- 192.168.10.0/24
```

- **-A** same as -O -sV which give OS detection and service version probing
- **-sS** TCP SYN-only scan
- **-sR** identify any RPC services
- **-p1-** probe all ports (TCP only in this case)

There are a **lot** more [command line switches](#) for nmap.
Exercise #1: Mapping Networks

Use Nmap to map out the hosts on the victim network:

1. Map out the hosts on the network:
   ```bash
   nmap -sP 172.16.10.0/24
   ```

2. Scan the hosts in more detail. Do this for each host that answered:
   ```bash
   nmap 172.16.10.X
   ```

3. Scan the hosts in even more detail. Do this for at least one host:
   ```bash
   nmap -A -sV -p1- 172.16.10.X
   ```

4. Examine the logs on the scanned hosts:
   ```bash
   less /var/log/messages
   ```
Why Do This?

A lot of information can be collected by just scanning:

- Open ports
- Operating system identification
- Service versions
- Even the existence of hosts

Firewalls are about the easiest and most commonly used methods for stopping or reducing this type of information gathering.
Firewalls

Firewalls can be classified into three types:

- Stateless
- Stateful
- Proxy

There are other features that are orthogonal to these classifications:

- NAT/PAT
- Port Redirection (RDR)
- Bridging
Based on FreeBSD with PF, pfSense is a very easy to use but still feature-rich firewalling systems available today. pfSense is also capable of:

- Captive Portal
- Wireless authentication (WPA)
- Terminating VPN connections
- Load Balancing
- Priority Queuing (through ALTQ)
- Failover (through CARP)
- Some proxy support
- Supports external packages (e.g. snort, pfflowd, squid, etc)

On reasonable hardware, it will easily perform as well as mid-sized commercial products.
Exercise #2: Blocking Packets

1. Open a browser on the victim and point it at http://172.16.10.254
2. Select the **Firewall** tab and then **Rules**
3. Disable the one rule that passes all traffic
4. Apply the changes
5. Now go back and repeat the same scans from Exercise #1
Exercise #3: Allow Some Access

This type of "one-way gate" is useful, but is not a perfect solution since we probably want to allow some inbound connections.

Let's punch some holes in this firewall to allow needed services:

1. Go back to Firewall -> Rules
2. Add a new rule allowing the following:
   - protocol: TCP
   - source: any
   - destination IP: 172.16.10.10 (this is Victim)
   - destination port: 80
3. Repeat the last scan from Exercise #1 against 172.16.10.10
Firewall Rule Design Philosophies

There are several major philosophies regarding the design of firewall rules:

- **Organic**
  - Grow as you go

- **Mostly Open**
  - "Allow All" rules and block traffic as necessary

- **Mostly Closed**
  - "Block All" rules and allow traffic as necessary

- **Paranoid**
  - "Mostly Closed" and find ways to *not* punch holes

- **Really Paranoid**
  - Don't connect the computers to a network
Nessus

Another common network reconnaissance tool is **Nessus**. Whereas Nmap scans for open ports and maybe does a small amount of service probing, Nessus runs "plugins" that collect information and can try to identify and even exploit real vulnerabilities.
Exercise #4: Nessus

1. Start the nessus daemon on Attacker:
   
   ```sh
   sudo /bin/sh /usr/local/etc/rc.d/nessus start
   ```

2. Start the nessus client: `nessus`

3. Connect to the local nessus daemon:
   
   | Nessusd Host | localhost |
   | Port         | 1241      |
   | Login        | attacker  |
   | Password     | attacker  |

4. Enable all plugins

5. Target 172.16.10.0/24

6. Start the scan and wait for the results
Questions?
Lab 3: Recon and Firewalls
### IP, UDP, TCP and ICMP

Before we can create firewall rules, we have to know the basics of network protocols. Here's a quick review...

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>The underlying packet delivery protocol</td>
</tr>
<tr>
<td>ICMP</td>
<td>Error messages, network book keeping and more</td>
</tr>
<tr>
<td>UDP</td>
<td>Connectionless and unreliable</td>
</tr>
<tr>
<td>TCP</td>
<td>Connections, reliable and sequenced</td>
</tr>
</tbody>
</table>
TCP 3-Way Handshake •
Map is one of the most commonly used network scanners today. It is commonly used for:

- Mapping hosts on networks
- Identifying a host’s OS
- Locating and identifying network services
Here's a common way that nmap is used to probe a network:

- `nmap -A -sS -sR -p1 192.168.10.0/24`

  - `-A` as same as `-O -sV` which give OS detection and service version probing.
  - `-sS` TCP SYN-only scan.
  - `-sR` identify any RPC services.
  - `-p1` probe all ports (TCP only in this case).

There are a lot more command line switches for nmap.
Exercise #1: Mapping Networks

Use Nmap to map out the hosts on the victim network:

1. Map out the hosts on the network:
   ```
   nmap -sP 172.16.10.0/24
   ```

2. Scan the hosts in more detail. Do this for each host that answered:
   ```
   nmap 172.16.10.X
   ```

3. Scan the hosts in even more detail. Do this for at least one host:
   ```
   nmap -A -sV -p1 - 172.16.10.X
   ```

4. Examine the logs on the scanned hosts:
   ```
   less /var/log/messages
   ```
W h y  D o  T h i s ?

A  l o t  o f  i n f o r m a t i o n  c a n  b e  c o l l e c t e d  b y  j u s t  s c a n n i n g :

- O p e n  p o r t s
- O p e r a t i n g  s y s t e m  i d e n t i f i c a t i o n
- S e r v i c e  v e r s i o n s
- E v e n  t h e  e x i s t e n c e  o f  h o s t s

F i r e w a l l s  a r e  a b o u t  t h e  e a s i e s t  a n d  m o s t  c o m m o n l y  u s e d  m e t h o d s  f o r
s t o p p i n g  o r  r e d u c i n g  t h i s  t y p e  o f  i n f o r m a t i o n  g a t h e r i n g .
Firewalls can be classified into three types:

- Stateless
- Stateful
- Proxy

There are other features that are orthogonal to these classifications:

- NAT / PAT
- Port Redirection (RDR)
- Bridging
Based on FreeBSD, pfSense is a very easy to use but still feature-rich firewalling system available today. pfSense is also capable of:

- Captive Portal
- Wireless Authentication (WPA)
- Terminating VPN connections
- Load Balancing
- Priority Queuing (through ALTQ)
- Failover (through CARP)
- Some proxy support
- Supports external packages (e.g., snort, pf sense, squid, etc.)

On reasonable hardware, it will easily perform as well as mid-sized commercial products.
Exercise #2: Blocking Packets

1. Open a browser on the victim and point it at http://172.16.10.254
2. Select the Firewall tab and then Rules
3. Disable the one rule that passes all traffic
4. Apply the changes
5. Now go back and repeat the same scans from Exercise #1
Exercise #3: Allow Some Access

This type of "one-way gate" is useful, but is not a perfect solution since we probably want to allow some inbound connections.

Let's punch some holes in this firewall to allow needed services:

1. Go back to Firewall -> Rules
2. Add a new rule allowing the following:
   - protocol: TCP
   - source: any
   - destination IP: 172.16.10.10 (this is Victim)
   - destination port: 80
3. Repeat the last scan against 172.16.10.10
There are several major philosophies regarding the design of firewall rules:

- **Organic**
  - *Grow as you go*

- **Mostly Open**
  - "Allow All" rules and block traffic as necessary

- **Mostly Closed**
  - "Block All" rules and allow traffic as necessary

- **Paranoid**
  - "Mostly Closed" and find ways to not punch holes

- **Really Paranoid**
  - Don't connect the computers to a network
Another common network reconnaissance tool is Nessus. Whereas Nmap scans for open ports and may do a small amount of service probing, Nessus runs "plugins" that collect information and can try to identify and even exploit real vulnerabilities.
Exercise #4: Nessus

1. Start the nessus daemon on Attacker:
   ```
   sudo /bin/sh /usr/local/etc/rc.d/nessus start
   ```

2. Start the nessus client:
   ```
   nessus
   ```

3. Connect to the local nessus daemon:
   ```
   Nessusd Host
   local host
   Port
   ```
   Log in at attacker
   Password  atacker

4. Enable all plugins

5. Target 172.16.10.0/24

6. Start the scan and wait for the results.