ARP SPOOF DETECTION

CS4115 FINAL PROJECT

Beiqi Li

BACKGROUND

- ARP is unauthenticated - Can be easily spoofed.
- S-ARP - which does have authentication - is not widely used.
- Write-once ARP table - Isn't quite standard compliant and can learn the spoofed packet.
- Static MAC entries - Hard to maintain for larger networks.

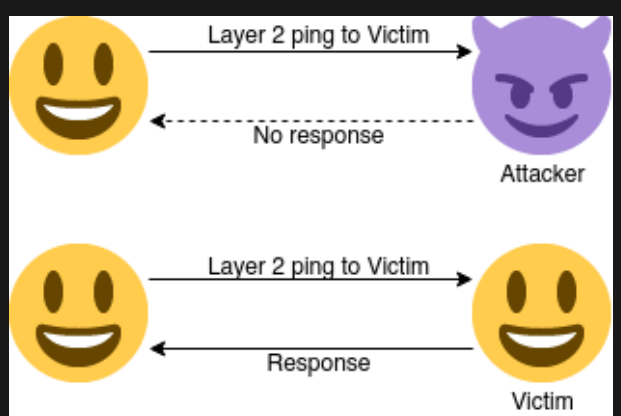
- Passive detection
  - Tracking ARP traffic and looking for inconsistencies.
  - Slow response. Performs bad on larger networks.
  - May also learn the wrong IP-MAC pair.
  - Does not detect ARP table overflow attack.

ACTIVE DETECTION

- Proposed by Ramachandran, et al. (2005).
- Use basic ping techniques and only check once - low overhead.
- Fast response.
- ARP responses are verified actively before learning.
- Possible to discover real IP-MAC pair under some conditions.

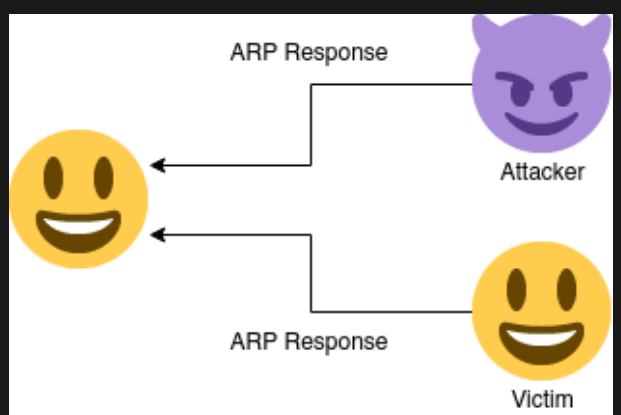
ASSUMPTIONS

- IP packets delivered to a machine that has the wrong destination IP will be dropped by the OS IP stack.
- We can use e.g. TCP pings sent on layer 2 to verify if a host is really using the IP.



ASSUMPTIONS

- Attacker will not stop victim from replying to ARP requests.
- We can send an ARP request to the potential victim and see if we got multiple replies.



DETECTION FLOW

- Sniff ARP traffics.
- Raise warning directly when:
  - The packet has different Ether MAC address and ARP MAC address.
  - The IP/MAC pair does not match our records.
- Otherwise, start active probing.

ACTIVE PROBING

- ARP session (flow): (`initiator mac`, `initiator ip`, `target ip`)
- 3 possible types:
  - Request half cycle - the flow passes a predefined time out and has 0 reply.
  - Response full cycle - unsolicited reply from target.
  - Full cycle - the flow passes a predefined time out and have 1+ replies.

- 2 main features:
  - # of responses of an ARP request (from us or sniffed)
  - Whether or not a host responds to TCP ping.

HANDLING DIFFERENT FLOW TYPES

- Simplified:
    1. Initiator: Always verify via layer 2 TCP pings.
    2. If it's a complete ARP session, verify all responders with layer 2 TCP pings.
    3. If there are only ARP responses, initiate an ARP session ourselves and treat all responders as with 2.

- Request half cycle
  - Check the initiator with TCP ping and raise warning if there's no response.
  - Otherwise record the IP/MAC pair.
- Response half cycle
  - Send an ARP request to the target and raise warning if there are 0 or more than 1 responses.
  - Otherwise, check the target with TCP ping and raise warning if there's no response.
  - Otherwise record the IP/MAC pair.

- Full cycle
  - Check the initiator with TCP ping and raise warning if there's no response, and
  - Raise warning if there are 0 or more than 1 captured ARP responses.
  - Otherwise, check the target with TCP ping and raise warning if there's no response.
  - Otherwise record the IP/MAC pair.

- Real host detection - in case of 1 or more available ARP responses
  - Check every target with TCP ping.
  - If there are exactly 1 reply, the replier is likely the real host (victim).
  - Otherwise, the real host cannot be detected.

- If there is no response, the attacker may be injecting invalid ARP records (e.g. table overflow).
- If there are more than 1 responses, the attacker is manually handling/forwarding packets.

IMPLEMENTATION

- Python 3.9 and Scapy.
- Implements all functionalities described in the paper.
- Simple MITM test with Ettercap shows promises.

DEMO

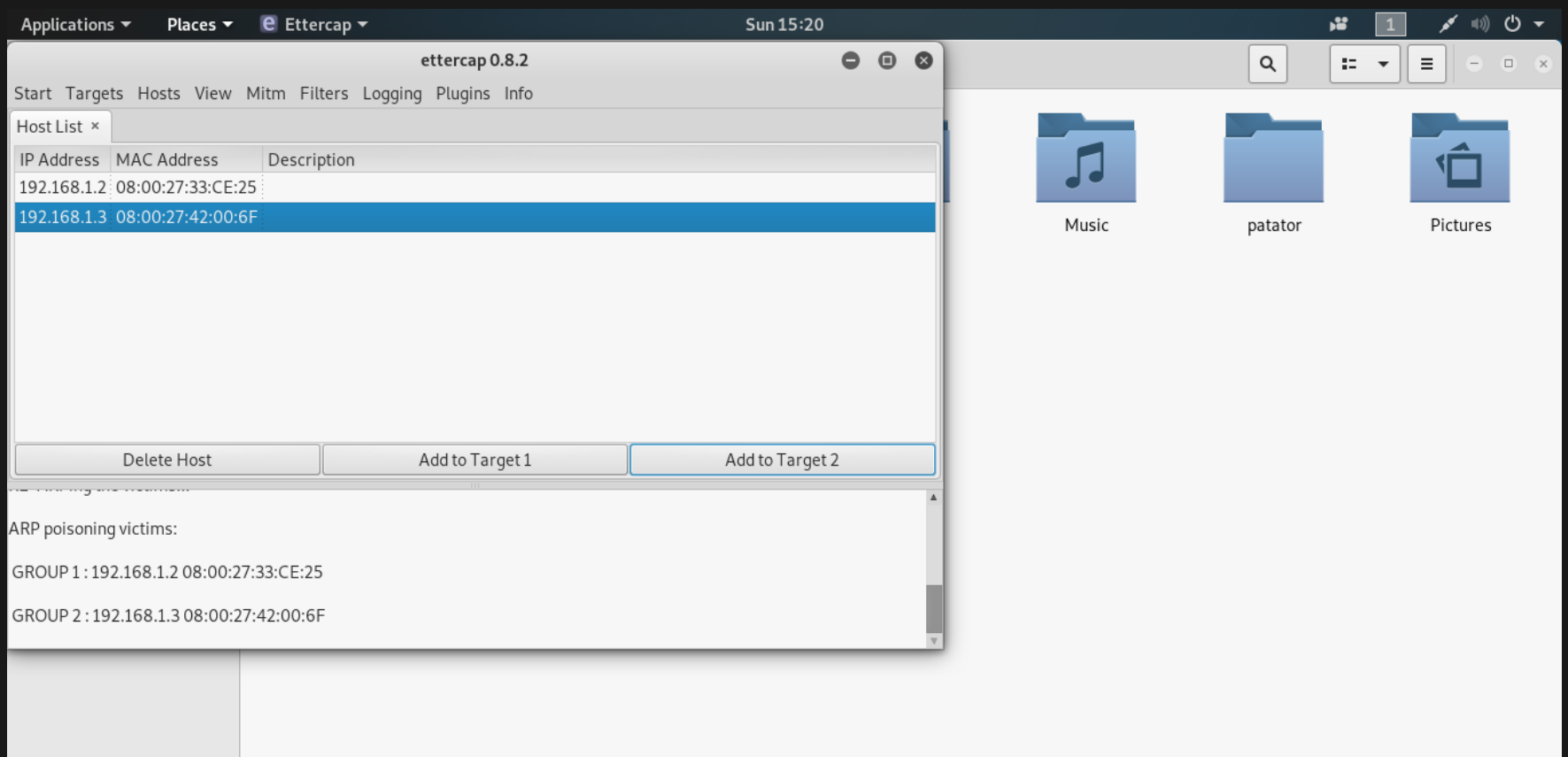
- Setup:
  - 2 victim VMs runs side-by-side.
  - Attacker Kali VM runs Ettercap and initiates ARP-spoofing-based MITM attack towards the 2 victim VMs.
  - The Ubuntu victim VM runs our software.

VICTIM

ubuntu@ubuntu-vic: ~

```
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
WARNING:arpspoof:Multiple hosts responded to the layer 2 ping. Unable to detect the real host.
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
INFO:arpspoof:Adding 192.168.1.1 (08:00:27:a1:b6:e6) to trusted ARP table.
Begin emission:
Finished sending 1 packets.
**
Received 2 packets, got 2 answers, remaining 0 packets
WARNING:arpspoof:Host 192.168.1.2 (08:00:27:a1:b6:e6) sent an unsolicited ARP response and multiple hosts
responded to the ARP ping. Possible spoofing event.
WARNING:arpspoof:Attempting to detect real host...
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
WARNING:arpspoof:Multiple hosts responded to the layer 2 ping. Unable to detect the real host.
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
INFO:arpspoof:Adding 192.168.1.2 (08:00:27:33:ce:25) to trusted ARP table.
WARNING:arpspoof:Contradicting ARP update found. Possible spoofing event.
```

ATTACKER


Applications
Places
Ettercap
Sun 15:20
ettercap 0.8.2
Start Targets Hosts View Mitm Filters Logging Plugins Info
Host List
IP Address MAC Address Description
192.168.1.2 08:00:27:33:CE:25
192.168.1.3 08:00:27:42:00:6F
Delete Host
Add to Target 1
Add to Target 2
ARP poisoning victims:
GROUP 1 : 192.168.1.2 08:00:27:33:CE:25
GROUP 2 : 192.168.1.3 08:00:27:42:00:6F
Music
patator
Pictures

CONTROL

```
File  Edit  View  Search  Terminal  Help                    En   3:22 PM
WARNING:arpspoof:Inconsistent Ether source (08:00:27:a1:b6:e6) and ARP source (08:00:27:33:ce:25). Possibl
e spoofing event.
^Cubuntu@ubuntu-vic:~$ clear

ubuntu@ubuntu-vic:~$ PYTHONUNBUFFERED=1 ./arpspoof -i eth0
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
INFO:arpspoof:Adding 192.168.1.1 (08:00:27:a1:b6:e6) to trusted ARP table.
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
INFO:arpspoof:Adding 192.168.1.2 (08:00:27:33:ce:25) to trusted ARP table.
```

FUTURE IMPROVEMENTS

- Test more scenarios.
- Fix edge cases.

REFERENCES

- Detecting ARP Spoofing: An Active Technique by Ramachandran, et al. (2005)
- Ettercap and middle-attacks tutorial (info on how to set up Ettercap for MITM)
- Scapy (the packet manipulation framework used in this project)

THANK YOU!