

Antes de explicarmos os procedimentos para a realização do ataque em questão, iremos abordar alguns detalhes sobre o protocolo *ARP* ([RFC 6747](#)) que serão essenciais para perceber o ataque.

Quando foi criado, não era focado na segurança. Sabendo isto, é possível encontrar fragilidades que nos permitirá realizar o ataque.

Funciona com endereços IPv4 de 32 bits (mais antigo). O IPv6 utiliza um protocolo diferente NDP (*Neighbor Discovery Protocol* - [RFC 4861](#)) que garante uma maior segurança usando chaves criptográficas para verificar as identidades dos *host*.

No entanto, a maioria da internet ainda utiliza o protocolo *IPv4*, logo o *ARP* continua a ser muito utilizado.

Como o protocolo *ARP* não verifica a identidade dos *hosts*, isto permite com que o “*hacker*” faça *spoofing* do *IP* do *router*, fingindo ser o mesmo. De seguida a vítima quando envia algo para o *IP* do *router*, na verdade está a enviar para a máquina do “*hacker*” e este reencaminha de seguida para o *router* fazendo *spoofing*, desta vez, com o *IP* da vítima (e assim vice-versa), interceptando assim todo o tráfego enviado. A isto chamamos *MitM* (*Man-in-the-middle* – [RFC 5713](#)).

Neste ataque começamos por verificar os endereços *MAC* das máquinas existentes na rede com o comando “*arp -a*” :

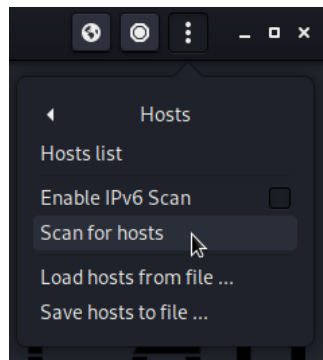
```
(kali㉿kali)-[~]  
$ arp -a  
? (192.168.100.240) at 06:23:69:29:aa:f1 [ether] on eth0  
  
(kali㉿kali)-[~]  
$
```

```
root@ipterm-1:~# arp -a  
? (192.168.100.240) at 06:23:69:29:aa:f1 [ether] on eth0  
root@ipterm-1:~# ping 192.168.100.240
```

Este comando é relevante para perceber que o endereço MAC do Router já se encontra nas tabelas ARP tanto do ipterm-1 como do kali-linux.

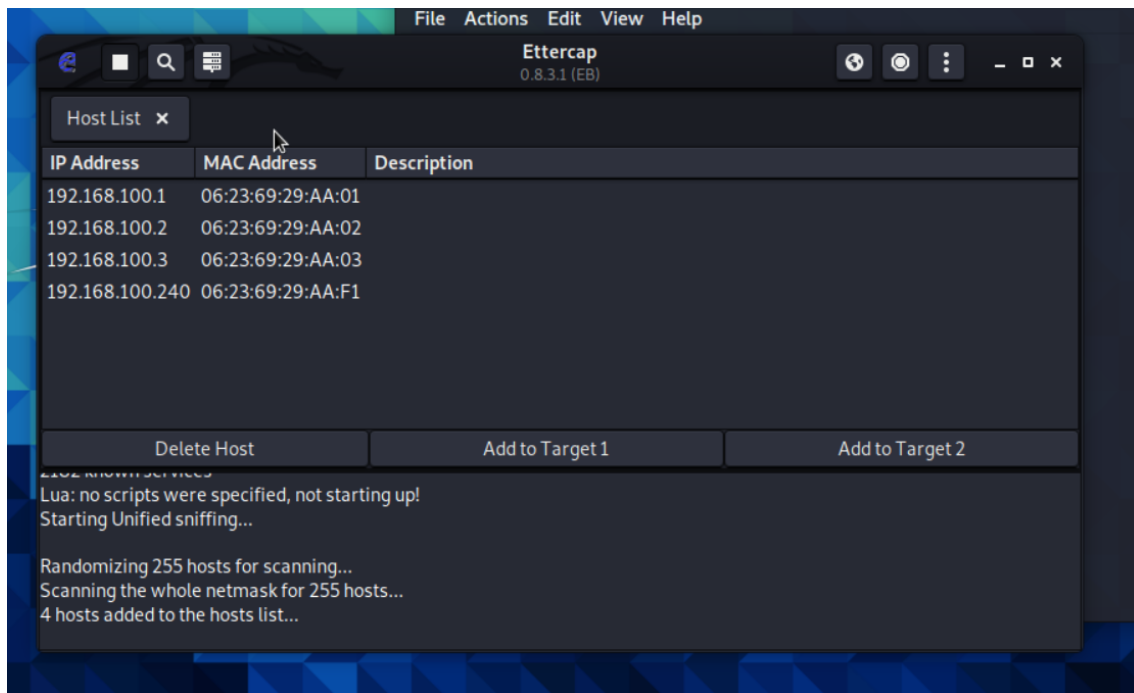
De seguida, podemos começar com o ataque propriamente dito. Para isto utilizamos um programa existente na máquina KaliLinux denominado de Ettercap

Já dentro do programa, após ter seleccionado o visto, (canto superior direito), para entrar, damos *scan* aos *hosts* existentes na rede. Após esta ação são adicionados à “*Hosts List*” presente no mesmo menu.



Quando o Ettercap realiza a função Scan for hosts, procura ativamente por todos dispositivos ativos na rede. Esta função permite que o Ettercap identifique quais dispositivos estão presentes na rede local e quais endereços IP estão a ser usados.

Ao executar o Scan for Hosts, ele envia pacotes ARP (Address Resolution Protocol) para toda a rede, solicitando que os dispositivos respondam com seus endereços IP e MAC correspondentes. O Ettercap analisa as respostas recebidas e constrói uma lista dos dispositivos descobertos na rede. Na captura do wireshark à saída do wireshark pode ser verificado que do pacote 18 ao 282 o kali-linux está apenas a enviar apenas pacotes ARP



Seleciona-se os targets, neste caso o 192.168.100.1 (ipterm-1) e o 192.168.100.240 (Router)

Após isto, o ettercap envia para rede pacotes ARP falsos para a rede local. Estes pacotes contêm informações falsas do endereço MAC, associadas aos IP's do ipterm-1 e do router. Pacotes 304 e 305.

304	63.267098	-	0c:c7:9a:9a:00:00	06:23:69:29:aa:01	ARP	60	192.168.100.240	is at	0c:c7:9a:9a:00:00
305	63.267896	-	0c:c7:9a:9a:00:00	06:23:69:29:aa:f1	ARP	60	192.168.100.1	is at	0c:c7:9a:9a:00:00

Quando voltamos a ver de novo a tabela arp do ipterm-1 percebemos que este já associa o ip da default gateway ao MAC address do kali-linux

```

root@ipterm-1:~# arp -a
? (192.168.100.5) at 0c:c7:9a:9a:00:00 [ether] on eth0
? (192.168.100.240) at 0c:c7:9a:9a:00:00 [ether] on eth0
root@ipterm-1:~#

```

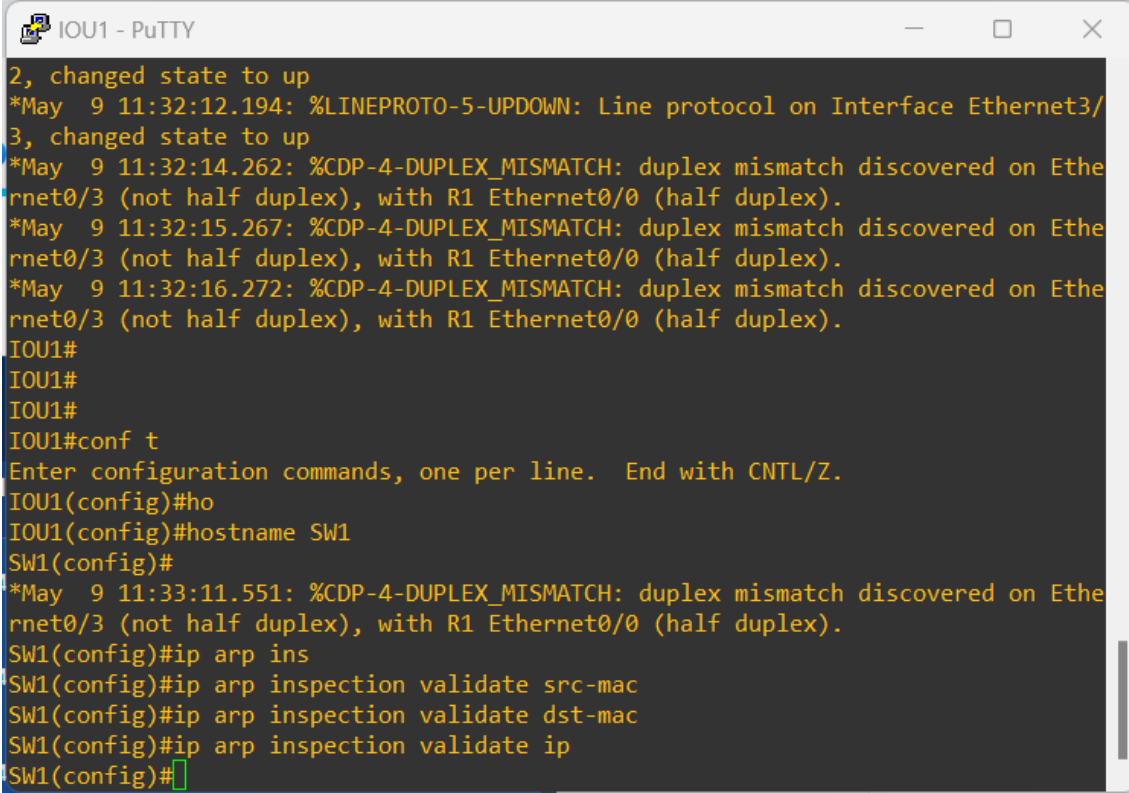
Percebemos então que já está a haver um redireccionamento do tráfego, pois ao analisar o wireshark fazem já 2 ICMP Request e 2 ICMP Reply. (Pacotes 321 a 324)

321	64.565108	-	192.168.100.1	192.168.100.240	ICMP	98	Echo (ping) request	id=0x0041, seq=14/3504, ttl=64 (no response found!)
322	64.566511	-	192.168.100.1	192.168.100.240	ICMP	98	Echo (ping) request	id=0x0041, seq=14/3504, ttl=64 (reply in 323)
323	64.567055	-	192.168.100.240	192.168.100.1	ICMP	98	Echo (ping) reply	id=0x0041, seq=14/3504, ttl=255 (request in 322)
324	64.574543	-	192.168.100.240	192.168.100.1	ICMP	98	Echo (ping) reply	id=0x0041, seq=14/3504, ttl=255

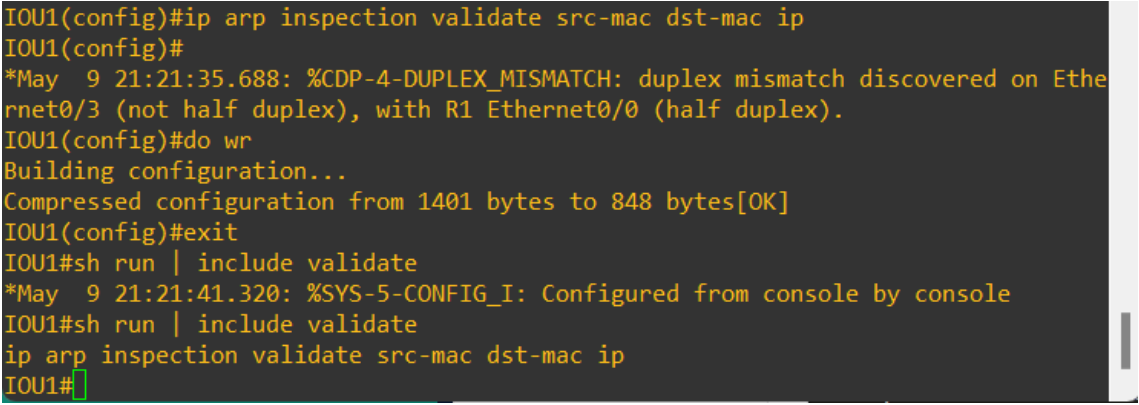
Um para o IP .240 com o MAC address do kali-linux e outra para o IP .240 com o MAC address do router e vice-versa.

Na PDU 319 percebemos que há um conflito de endereço IP duplicado para o gateway padrão (IP-default gateway). O Wireshark identificou que o endereço IP do gateway padrão está sendo usado por dois dispositivos com endereços MAC diferentes: o endereço MAC do Kali Linux e o endereço MAC do roteador. Isto porque existem dois MAC address associados a apenas um IP.

## Medidas de mitigação.



```
IOU1 - PuTTY
2, changed state to up
*May  9 11:32:12.194: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet3/
3, changed state to up
*May  9 11:32:14.262: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on Ethe
rnet0/3 (not half duplex), with R1 Ethernet0/0 (half duplex).
*May  9 11:32:15.267: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on Ethe
rnet0/3 (not half duplex), with R1 Ethernet0/0 (half duplex).
*May  9 11:32:16.272: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on Ethe
rnet0/3 (not half duplex), with R1 Ethernet0/0 (half duplex).
IOU1#
IOU1#
IOU1#
IOU1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
IOU1(config)#ho
IOU1(config)#hostname SW1
SW1(config)#
*May  9 11:33:11.551: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on Ethe
rnet0/3 (not half duplex), with R1 Ethernet0/0 (half duplex).
SW1(config)#ip arp ins
SW1(config)#ip arp inspection validate src-mac
SW1(config)#ip arp inspection validate dst-mac
SW1(config)#ip arp inspection validate ip
SW1(config)#
```



```
IOU1(config)#ip arp inspection validate src-mac dst-mac ip
IOU1(config)#
*May  9 21:21:35.688: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on Ethe
rnet0/3 (not half duplex), with R1 Ethernet0/0 (half duplex).
IOU1(config)#do wr
Building configuration...
Compressed configuration from 1401 bytes to 848 bytes[OK]
IOU1(config)#exit
IOU1#sh run | include validate
*May  9 21:21:41.320: %SYS-5-CONFIG_I: Configured from console by console
IOU1#sh run | include validate
ip arp inspection validate src-mac dst-mac ip
IOU1#
```

```
S1(config)# do show run | include validate
ip arp inspection validate ip
S1(config)# ip arp inspection validate src-mac dst-mac ip
S1(config)# do show run | include validate
ip arp inspection validate src-mac dst-mac ip
S1(config)#
```

S1(config)# ip arp inspection validate src-mac:

Este comando configura o switch para validar o endereço MAC de origem nos pacotes ARP recebidos. O switch verifica se o endereço MAC de origem é válido e corresponde ao dispositivo que está enviando o pacote ARP. Se o endereço MAC de origem não for válido, o pacote será descartado.

S1(config)# ip arp inspection validate dst-mac:

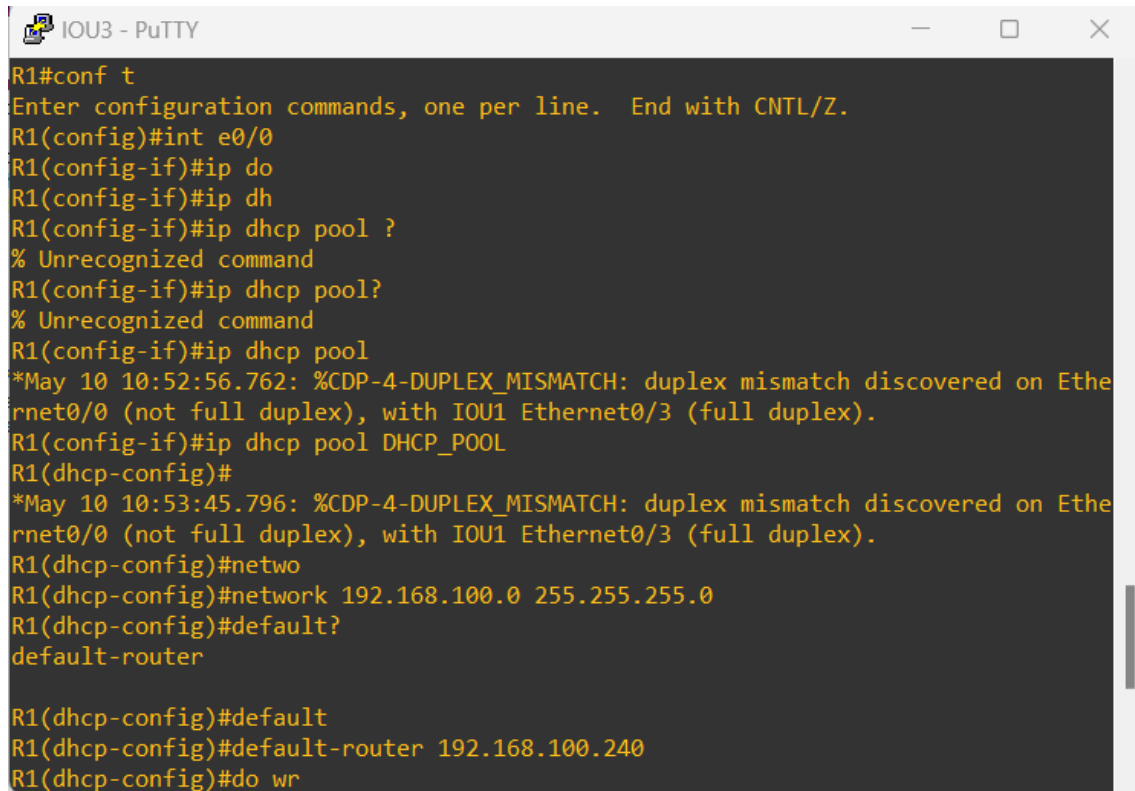
Este comando configura o switch para validar o endereço MAC de destino nos pacotes ARP recebidos. O switch verifica se o endereço MAC de destino é válido e corresponde ao dispositivo para o qual o pacote ARP é destinado. Se o endereço MAC de destino não for válido, o pacote será descartado.

S1(config)# ip arp inspection validate ip:

Este comando configura o switch para validar o endereço IP nos pacotes ARP recebidos. O switch verifica se o endereço IP é válido e corresponde ao dispositivo que está enviando o pacote ARP. Se o endereço IP não for válido, o pacote será descartado.

## DHCP Starvation/Exhaustion Attack

Primeiro configurar o router como dhcp server



```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int e0/0
R1(config-if)#ip do
R1(config-if)#ip dh
R1(config-if)#ip dhcp pool ?
% Unrecognized command
R1(config-if)#ip dhcp pool?
% Unrecognized command
R1(config-if)#ip dhcp pool
*May 10 10:52:56.762: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on Ethe
rnet0/0 (not full duplex), with IOU1 Ethernet0/3 (full duplex).
R1(config-if)#ip dhcp pool DHCP_POOL
R1(dhcp-config)#
*May 10 10:53:45.796: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on Ethe
rnet0/0 (not full duplex), with IOU1 Ethernet0/3 (full duplex).
R1(dhcp-config)#netwo
R1(dhcp-config)#network 192.168.100.0 255.255.255.0
R1(dhcp-config)#default?
default-router

R1(dhcp-config)#default
R1(dhcp-config)#default-router 192.168.100.240
R1(dhcp-config)#do wr
```

Para este ataque importei uma maquina do kali-linux nova pois estava com dificuldades em configurar a appliance.

Para o DHCP Starvation/Exhaustion attack foi usada a ferramenta chamada yersinia. Este permite realizar uma variedade de ataques de protocolo de rede.

DHCP (Dynamic Host Configuration Protocol) é um protocolo de rede que permite que dispositivos numa rede obtenham automaticamente endereços IP e outras informações de configuração de rede. Um ataque de esgotamento de DHCP, também conhecido como ataque de inanição de DHCP, é um tipo de ataque DoS que envolve o esgotamento da pool de endereços IP disponíveis no servidor DHCP, impedindo assim que outros dispositivos obtenham endereços IP e se conectem à rede.

O ataque de esgotamento de DHCP é realizado por um invasor que envia uma grande quantidade de solicitações DHCP para o servidor DHCP, solicitando endereços IP em rápida sucessão. Isso pode ser feito por meio de várias técnicas, como o uso de múltiplos dispositivos para enviar solicitações simultâneas, o uso de endereços MAC falsificados para simular uma grande quantidade de dispositivos ou o uso de ferramentas automatizadas que geram solicitações de DHCP em massa.

o Yersinia realiza o ataque DHCP Starvation/Exhaustion:

Inundação de solicitações DHCP: O Yersinia gera um grande número de solicitações DHCP falsas(DHCP Discovery). Essas solicitações são enviadas ao servidor DHCP com o objetivo de solicitar um endereço IP.

**Nota: Pacotes DHCP Discovery-** Quando um dispositivo se conecta a uma rede e está configurado para obter seu endereço IP automaticamente, ele envia um pacote DHCP Discover para descobrir servidores DHCP disponíveis na rede.O pacote é enviado como uma transmissão de broadcast, ou seja, é enviado para todos os dispositivos na rede local, incluindo o servidor DHCP.

Falsa alocação de endereços IP: O Yersinia continua a gerar e enviar solicitações DHCP falsas em um ritmo rápido e constante, de forma a esgotar a pool de endereços IP disponíveis no servidor DHCP. Ao receber estes pacotes o servidor DHCP ele verifica a pool de endereços disponíveis e envia um pacote DHCP Offer como resposta. Esse pacote é enviado como uma transmissão de Broadcast.

**Nota: Pacotes DHCP Offer-** Contém informações sobre o endereço IP que o servidor DHCP está a passar ao cliente. Além disso, inclui outras configurações de rede, como a máscara de sub-rede, o gateway, servidores DNS e outras opções de configuração definidas pelo administrador da rede.

Após iniciar o ataque irei fazer algumas verificações para perceber como consiste e perceber realmente o ataque levado a cabo pelo yersinia

Conseguimos perceber que o servidor dhcp associou ip's a endereços MAC falsos.

```
R1#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/
                    Hardware address/
                    User name
192.168.100.4        0108.0027.c7e1.36    May 12 2023 01:37 PM    Automatic
192.168.100.5        0100.5079.6668.00    May 12 2023 01:39 PM    Automatic
192.168.100.6        f0b0.fb2f.47fc       May 11 2023 01:49 PM    Automatic
192.168.100.7        4353.034a.3188       May 11 2023 01:49 PM    Automatic
192.168.100.8        ce75.9105.bffe       May 11 2023 01:49 PM    Automatic
192.168.100.9        1b87.3f33.31dd       May 11 2023 01:49 PM    Automatic
192.168.100.10       c35e.966a.cc5f       May 11 2023 01:49 PM    Automatic
192.168.100.11       aa78.2f3a.dfeb       May 11 2023 01:49 PM    Automatic
192.168.100.12       ac7c.430f.0b98       May 11 2023 01:49 PM    Automatic
192.168.100.13       3422.9f1a.42cf       May 11 2023 01:49 PM    Automatic
192.168.100.14       6923.a245.59a0       May 11 2023 01:49 PM    Automatic
192.168.100.15       7b59.f77e.36c1       May 11 2023 01:49 PM    Automatic
192.168.100.16       7317.cc1e.da04       May 11 2023 01:49 PM    Automatic
192.168.100.17       261a.4473.54c4       May 11 2023 01:49 PM    Automatic
192.168.100.18       a290.b221.32a3       May 11 2023 01:49 PM    Automatic
192.168.100.19       357a.6520.964d       May 11 2023 01:49 PM    Automatic
192.168.100.20       2c9d.720a.2231       May 11 2023 01:49 PM    Automatic
192.168.100.21       cb52.6231.8b88       May 11 2023 01:49 PM    Automatic
192.168.100.22       d651.935b.7421       May 11 2023 01:49 PM    Automatic
192.168.100.23       2d5b.8f35.ca28       May 11 2023 01:49 PM    Automatic
192.168.100.24       e63a.4d65.174b       May 11 2023 01:49 PM    Automatic
192.168.100.25       637d.9808.8de9       May 11 2023 01:49 PM    Automatic
192.168.100.26       2926.bb04.d538       May 11 2023 01:49 PM    Automatic
192.168.100.27       1518.583d.d3b6       May 11 2023 01:49 PM    Automatic
192.168.100.28       0729.d927.4192       May 11 2023 01:49 PM    Automatic
192.168.100.29       e72e.2f57.ebb9       May 11 2023 01:49 PM    Automatic
192.168.100.30       1762.8350.88de       May 11 2023 01:49 PM    Automatic
192.168.100.31       3805.ea75.d9e8       May 11 2023 01:49 PM    Automatic
192.168.100.32       641b.fe47.8c88       May 11 2023 01:49 PM    Automatic
192.168.100.33       3ce3.f149.b815       May 11 2023 01:49 PM    Automatic
192.168.100.34       f789.c721.c8c7       May 11 2023 01:49 PM    Automatic
192.168.100.35       76a4.d73e.5e53       May 11 2023 01:49 PM    Automatic
192.168.100.36       c429.274f.7b67       May 11 2023 01:49 PM    Automatic
192.168.100.37       4175.a866.54a6       May 11 2023 01:49 PM    Automatic
192.168.100.38       4f9d.9f6b.a5ec       May 11 2023 01:49 PM    Automatic
192.168.100.39       3f43.df0b.2329       May 11 2023 01:49 PM    Automatic
192.168.100.40       b6b2.0605.19d5       May 11 2023 01:49 PM    Automatic
192.168.100.41       e071.d020.a0e2       May 11 2023 01:49 PM    Automatic
192.168.100.42       b0b6.3f57.1de8       May 11 2023 01:49 PM    Automatic
192.168.100.43       b57b.c01d.d05e       May 11 2023 01:49 PM    Automatic
192.168.100.44       3fc4.1f36.8c1f       May 11 2023 01:49 PM    Automatic
192.168.100.45       282c.946c.f92c       May 11 2023 01:49 PM    Automatic
192.168.100.46       ff32.164b.de93       May 11 2023 01:49 PM    Automatic
192.168.100.47       e09f.f275.7761       May 11 2023 01:49 PM    Automatic
```

Neste caso percebi que o servidor DHCP ficou completamente cheio.



Também corremos o comando `sh ip dhcp server statistics` para que nos fornecesse um resumo das estatísticas do servidor DHCP.

```
R1#sh ip dhcp server statistics
```

```
Memory usage      2138464
Address pools     1
Database agents   0
Automatic bindings 250
Manual bindings   0
Expired bindings  0
Malformed messages 0
Secure arp entries 0
```

```
Message           Received
BOOTREQUEST       0
DHCPDISCOVER      25765
DHCPREQUEST        3
DHCPDECLINE        0
DHCPRELEASE        0
DHCPINFORM         0
```

```
Message           Sent
BOOTREPLY          0
DHCPOFFER          251
DHCPACK             2
DHCPNAK             0
```

Após a execução deste comando interrompemos o ataque e fazemos o reset do servidor DHCP para perceber as diferenças.

**Após executar o `sh ip dhcp binding`:**

```
R1#sh ip dhcp binding
```

```
Bindings from all pools not associated with VRF:
```

IP address	Client-ID/ Hardware address/ User name	Lease expiration	Type
192.168.100.4	0108.0027.c7e1.36	May 12 2023 02:04 PM	Automatic
192.168.100.5	0100.5079.6668.00	May 12 2023 02:05 PM	Automatic

```
R1#
```

**Após executar o `sh ip dhcp statistics`:**

```
IOU3 - PuTTY
R1#sh ip dhcp server statistics
Memory usage      40880
Address pools     1
Database agents   0
Automatic bindings 2
Manual bindings   0
Expired bindings  0
Malformed messages 0
Secure arp entries 0

Message           Received
BOOTREQUEST       0
DHCPDISCOVER      5
DHCPREQUEST       3
DHCPDECLINE       0
DHCPRELEASE       0
DHCPINFORM        0

Message           Sent
BOOTREPLY         0
DHCPOFFER        3
DHCPACK          2
DHCPNAK          0
R1#
```

Após as estáticas do servidor de DHCP se terem normalizado, é possível tirar conclusões.

Após o ataque:

**Memória usada pelo servidor:** é possível observar um aumento significativo no uso de memória. Isto ocorre porque o servidor DHCP precisa de armazenar informações adicionais sobre as solicitações de DHCP recebidas durante o ataque, como os endereços IP atribuídos temporariamente e os bindings de DHCP correspondentes. O aumento na demanda por endereços IP leva a um uso mais intensivo da memória pelo servidor DHCP.

**Número de pacotes DHCP Discover :** Após o ataque de esgotamento DHCP, houve um aumento significativo de pacotes DHCP Discover devido à inundação da rede com solicitações falsas.

**Aumento taxa de pacotes DHCP Offer:** o servidor DHCP ao tentar responder a uma alta demanda por endereços IP, resulta num maior número de pacotes DHCP Offer enviados pelo servidor para atender às solicitações dos “dispositivos na rede”.

### Analisar a captura:

Ao analisar a captura percebemos uma “inundação de pacotes” DHCP Discover, que o router fica quase sem capacidade de resposta. Entre os milhares de pacotes DHCP Discover vemos alguns pacotes DHCP Offer vindos do router.

Também é possível o router a enviar pacotes ARP (Address Resolution Protocol) broadcast para toda a rede. Esses pacotes ARP broadcast são enviados para descobrir quais dispositivos na rede possuem os endereços IP mencionados nos pacotes DHCP Discover e DHCP Offer.

## DHCP SPOOFING ATTACK

Um ataque de spoofing de DHCP ocorre quando um servidor DHCP invasor está conectado à rede e fornece falsos parâmetros de configuração IP aos clientes legítimos. Um servidor não autorizado pode fornecer uma variedade de informações enganosas:

**Gateway errado-** O servidor não autorizado fornece um gateway inválido ou o endereço IP de seu host para criar um ataque man-in-the-middle. Isso pode passar totalmente despercebido, pois o invasor intercepta o fluxo de dados pela rede e o encaminha para o gateway padrão real.

**Servidor DNS errado-** O servidor não autorizado fornece um endereço de servidor DNS incorreto, apontando o usuário para um site nefasto.

**Endereço IP errado-** O servidor não autorizado fornece um endereço IP inválido, criando efetivamente um ataque de negação de serviço no cliente DHCP.

Para este ataque voltamos a usar a ferramenta Ettercap.

Ao analisar a captura vemos o PC a enviar um pacote DHCP Discover para toda a rede e de seguida a maquina kali-linux a enviar um DHCP Offer, o pc envia um dhcp request(confirmar a aceitação do DHCP Offer), o servidor que neste caso é o kali Linux, envia o dhcp ACK para atribuir o ip à maquina, que neste caso foi um definido por mim.

Formas de mitigação para o DHCP Starvation.

```
S1 (config)#interface range e0/0 - 3
```

```
S1 (config-if-range)#ip dhcp snooping limit rate 1
```

```
S1 (config-if-range)#exit
```

Com este comando definimos o limite de taxas de mensagens DHCP que as interfaces podem receber.O limite de taxa é definido como 1 pacotes por segundo. Isso impõe uma restrição na velocidade em que as mensagens DHCP podem ser recebidas em cada interface seleccionada.

