

# Hacking Articles

Raj Chandel's Blog

[Author](#)[Web Penetration Testing](#)[Penetration Testing](#)[Courses We Offer](#)[My Books](#)[Donate us](#)

## ICMP Penetration Testing

posted in **PENETRATION TESTING** on **JANUARY 3, 2018** by **RAJ CHANDEL**  **SHARE**

In our previous article we had discussed “**ICMP protocol with Wireshark**” where we had seen how an ICMP protocol work at layer 3 according to OSI model and study its result using wireshark. Today we are going discuss to ICMP penetration testing by crafting ICMP packet to test our IDS “Snort” against all ICMP message Types using Cat Karat tool, you can download it from <http://packetbuilder.net> link.

For configuring Snort as IDS read our previous article “**Configure snort in Ubuntu**” it will automatically install snort in your system with predefine set of rules that will help in packet capturing of your network.

**Let's start!!**

Search

Subscribe to Blog via Email

**SUBSCRIBE**

Basically we will perform this practical in three phases as describe below:

**Packet crafting:** In this phase we will craft each ICMP packet with different type ICMP message using Cat Karat. For more detail about Packet crafting process read our previous [article](#).

**Packet Capturing:** In this phase we will capture the ICMP packet and receive an alert when it will enters into target's network using snort as IDS.

**Packet Analysis:** In this phase we will investigate captured packet using wireshark.

### Brief Introduction on ICMP protocol

ICMP message contains two types of codes i.e. query and error.

**Query:** The query messages are the information we get from a router or another destination host.

For example given below message types are some ICMP query codes:

- Type 0 = Echo Reply
- Type 8 = Echo Request
- Type 9 = Router Advertisement
- Type 10 = Router Solicitation
- Type 13 = Timestamp Request
- Type 14 = Timestamp Reply

**Error:** The error statement messages reports problem which a router or a destination host may generate.

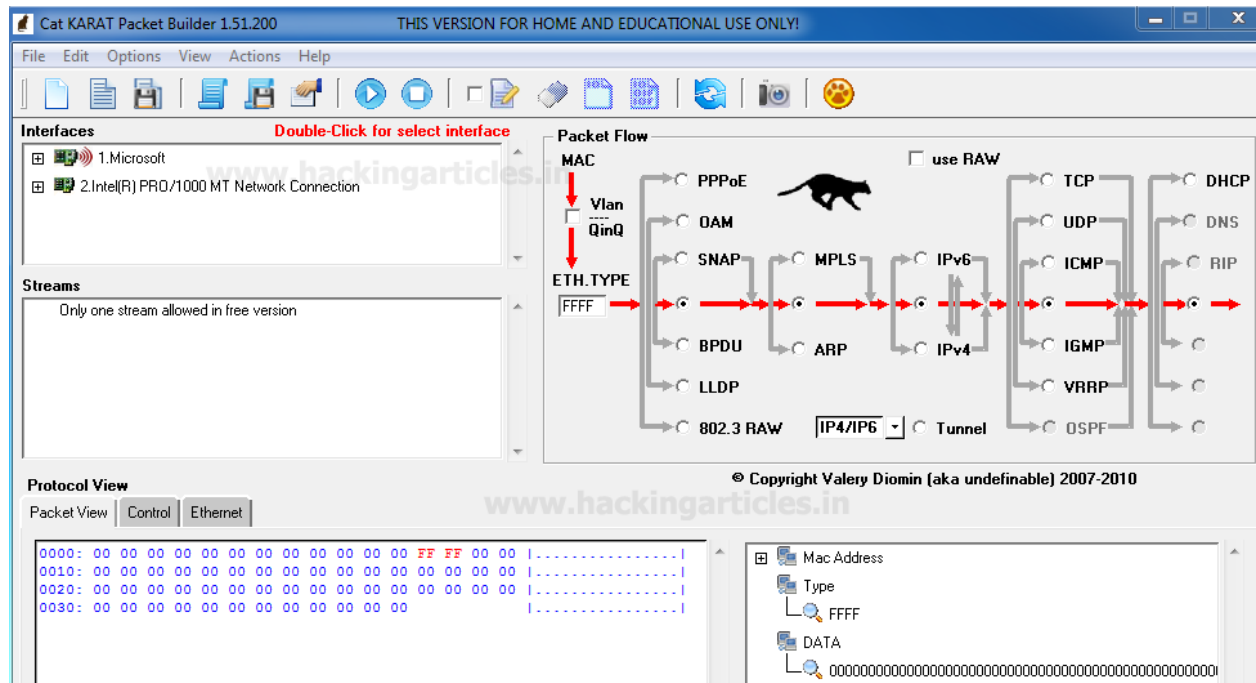
For example: given below message types are some of the ICMP error codes:

- Type 3 = Destination Unreachable
- Type 4 = Source Quench



- Type 5 = Redirect
- Type 11 = Time Exceeded
- Type 12 = Parameter Problems

Now when you will run the installed application “Cat KARAT” you will observe three important sections “Interfaces”, “Packet flow” and Packet view which in their default state as shown in given below image.



## Message TYPE 0 ICMP Packet Crafting

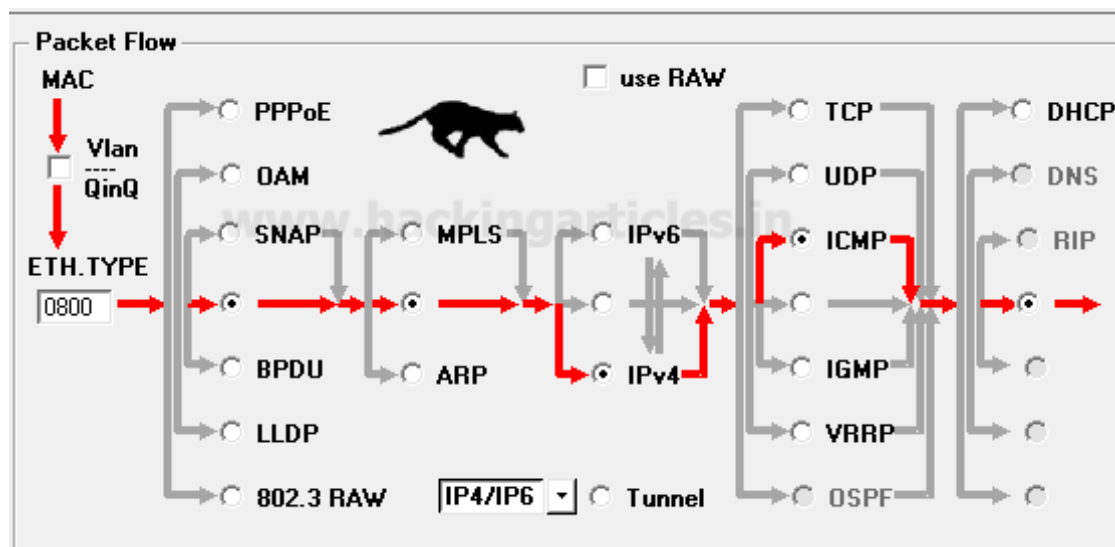
So as we know in **Packet Crafting** Operation “**Packet Assembly**” is 1st phase where we need to decide protocol for crafting any packet, which is quite easy to select with this tool. Only enable the radio button for selecting protocol and direction flow of packet. Here I had

## Categories

- 🔖 BackTrack 5 Tutorials
- 🔖 Best of Hacking
- 🔖 Browser Hacking
- 🔖 Cryptography & Steganography
- 🔖 CTF Challenges
- 🔖 Cyber Forensics
- 🔖 Database Hacking
- 🔖 Domain Hacking
- 🔖 Email Hacking
- 🔖 Footprinting
- 🔖 Hacking Tools
- 🔖 Kali Linux
- 🔖 Nmap
- 🔖 Others
- 🔖 Penetration Testing
- 🔖 Social Engineering Toolkit
- 🔖 Trojans & Backdoors
- 🔖 Website Hacking
- 🔖 Window Password Hacking
- 🔖 Windows Hacking Tricks
- 🔖 Wireless Hacking
- 🔖 Youtube Hacking

enable radio buttons for “IPv4” and “ICMP” without disturbing remaining default packet flow as shown given below image.

Next we need to select the “interface” which you can select from the Interfaces by double-Click on it.



Now next is “**packet Editing**” phase where you need to specify source IP address such as: **192.168.1.2** from which packet will be sent and Destination IP address such as: **192.168.1.107** on which packet is received. Moreover you can also make some changes in your packet such as Time to live (TTL), Data length and also can go with packet fragmentation.

From given below image you can observe I had added source and destination IP in packet under the third section **protocol view -> Ipv4**

## Articles

Select Month



## Facebook Page



Protocol View © Copyright Valery Diomin (aka undefin)

Packet View Control Ethernet IPv4 ICMP

4 Version 20 Header Length 76 ☐ Total Length Override 192.168.1.107 Fixed Dest. Address

5 x4 ☐ Header Length Override 0000 Fixed Identifier (0-65535)

QoS

☒ TOS ☐ DSCP

TOS Bits 0-2 (Precedence) 000

TOS Bit 3 (Delay) 0

TOS Bit 4 (Throughput) 0

TOS Bit 5 (Reliability) 0

TOS Bit 6 (Cost) 0

TOS Bit 7 (Reserved) 0

0-May Fragment 0-Last Fragment Fragmentation

0000 Fragm.Offset (x8) (0-8191) 255 Time to Live

1-ICMP Protocol

37f3 ☐ Checksum Override (hex)

192.168.1.2 Fixed Source Address

Options

Copy Source IP from NIC IP Broadcast

Under 3<sup>rd</sup> section protocol view in cat Karat explore **ICMP tab** and select “0-Echo Response” option which is generate type 0 ICMP message. Once everything is edited then your packet is ready to send on target network. **Click on play** button given in menu bar for sending packet on target’s network which known as “packet playing” phase of packet Crafting operation. This ICMP message type also uses to test the strength of IDS and Firewall against ICMP smurf Dos Attack.

Protocol View

Packet View Control Ethernet IPv4 ICMP

0-Echo Response

0 Code (0-255)

ffff ☐ Checksum Override (hex)

00000 ID (0-65535)

00000 Sequence (0-65535)

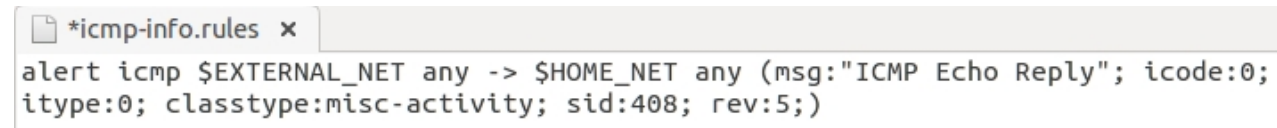
## Capturing ICMP-Type0 packet through IDS

Advantage of install snort through apt respiratory is that, it is quick and easy to install in your system as well as it contains predefined set of rule files related to every type of network traffic either TCP/UDP or ICMP.

From given below image you can observe that inside the file “icmp-info.rules” an alert rule is already implemented for capturing the traffic of **ICMP echo Reply** packet is found in network. This rule also works against Smurf Dos attack in which ICMP echo reply/response traffic is received on target’s network without sending genuine ICMP request packet from target’s network to other network.

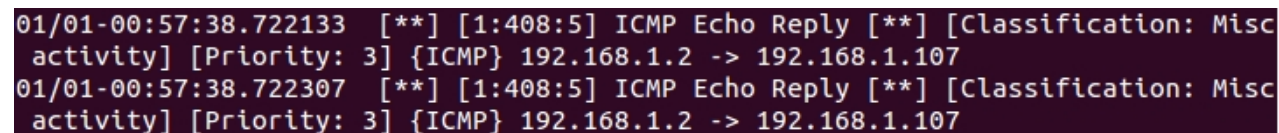
Now turn on IDS mode of snort by executing given below command in terminal:

**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0**



```
*icmp-info.rules x
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:'ICMP Echo Reply'; icode:0;
itype:0; classtype:misc-activity; sid:408; rev:5;)
```

So when IDS received any matching packets defined in file of rules then generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for “ICMP Echo Reply” packets from source address 192.168.1.1.2 to destination 192.168.1.107.

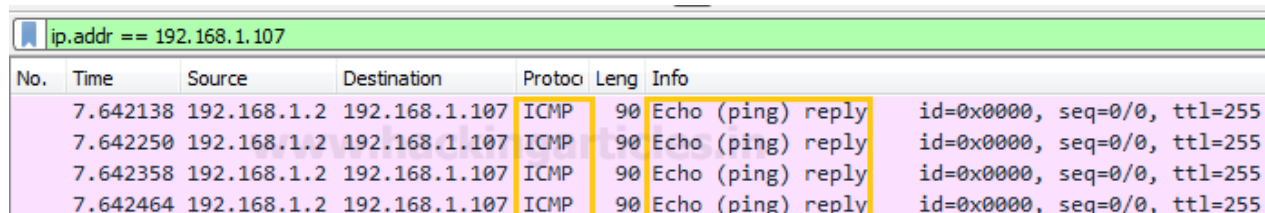


```
01/01-00:57:38.722133  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-00:57:38.722307  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc
activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
```

## Analysis ICMP-Type0 packet through Wireshark

Now Last phase is **Packet Analysis** which is also last mode of operation of packet crafting process where received packet is analysis using packet analysis tool. Here we had use

wireshark for capturing incoming traffic. From given below image you can observe that wireshark has captured exactly same information which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP message type packet and other information.

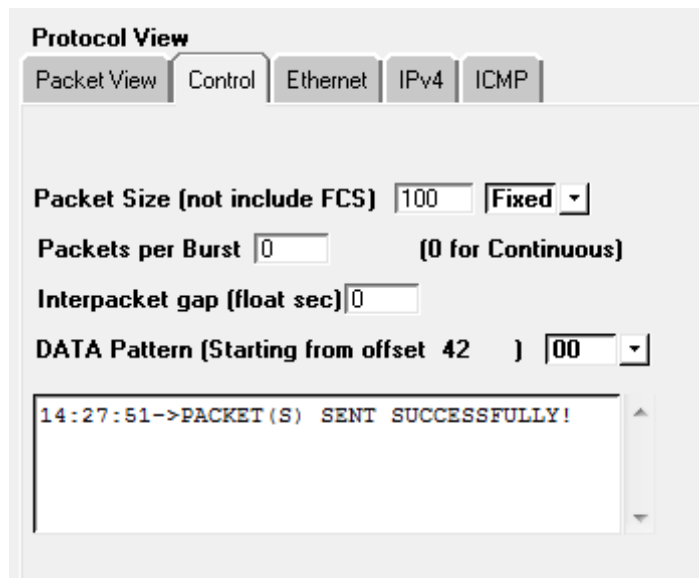


The image shows a Wireshark packet capture window with a filter set to 'ip.addr == 192.168.1.107'. It displays four packets, all of which are ICMP Echo (ping) replies. The 'Protocol' column is highlighted in yellow for all four packets.

No.	Time	Source	Destination	Protocol	Length	Info
7.642138	192.168.1.2	192.168.1.107	ICMP	90	Echo (ping) reply	id=0x0000, seq=0/0, ttl=255
7.642250	192.168.1.2	192.168.1.107	ICMP	90	Echo (ping) reply	id=0x0000, seq=0/0, ttl=255
7.642358	192.168.1.2	192.168.1.107	ICMP	90	Echo (ping) reply	id=0x0000, seq=0/0, ttl=255
7.642464	192.168.1.2	192.168.1.107	ICMP	90	Echo (ping) reply	id=0x0000, seq=0/0, ttl=255

When the tester will click on **Stop button** given in menu bar of Cat Karat tool he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 0 is successfully sent on target machine.



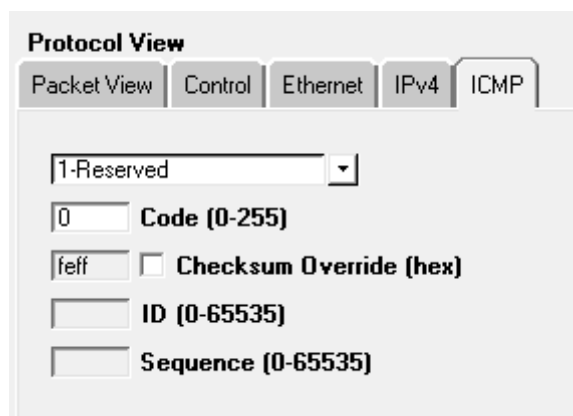
## Message TYPE 1 ICMP Packet Crafting

So the Packet Assembly phase and Packet Editing phase for ICMP packet crafting is almost same as above only the difference is make in change ICMP message through which connection will be established with target network.

Since we want to send traffic through message type 1 packets for establishing connection with target network therefore select **Type -1 Reserved** from given list.

Once everything is edited then your packet is ready to send on target network.

**Click on play** button given in menu bar for sending packet on target's network which known as "packet playing" phase of packet Crafting operation.



The screenshot shows a 'Protocol View' window with tabs for 'Packet View', 'Control', 'Ethernet', 'IPv4', and 'ICMP'. The 'ICMP' tab is selected. Below the tabs, there is a dropdown menu showing '1-Reserved'. Below this, there is a text input field with '0' and the label 'Code (0-255)'. Below that, there is a text input field with 'feff' and a checkbox labeled 'Checksum Override (hex)'. Below that, there is a text input field with the label 'ID (0-65535)'. Below that, there is a text input field with the label 'Sequence (0-65535)'.

## Capturing ICMP-Type1 packet through IDS

From given below image you can observe that inside the file "icmp-info.rules" an alert rule is already implemented for capturing the traffic of **ICMP unassigned type 1** packet is found in network.

Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



```
*icmp-info.rules x
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP unassigned type 1";
icode:0; itype:1; classtype:misc-activity; sid:458; rev:7;)
|
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP unassigned type 1
undefined code"; itype:1; classtype:misc-activity; sid:459; rev:7;)
```

So when IDS received any matching packets defined in file of rules then generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for "ICMP unassigned type 1" packets from source address 192.168.1.1.2 to destination 192.168.1.107.

```
01/01-01:05:10.447347  [**] [1:458:7] ICMP unassigned type 1 [**] [Classificatio
n: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-01:05:10.447448  [**] [1:459:7] ICMP unassigned type 1 undefined code [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.1
07
```

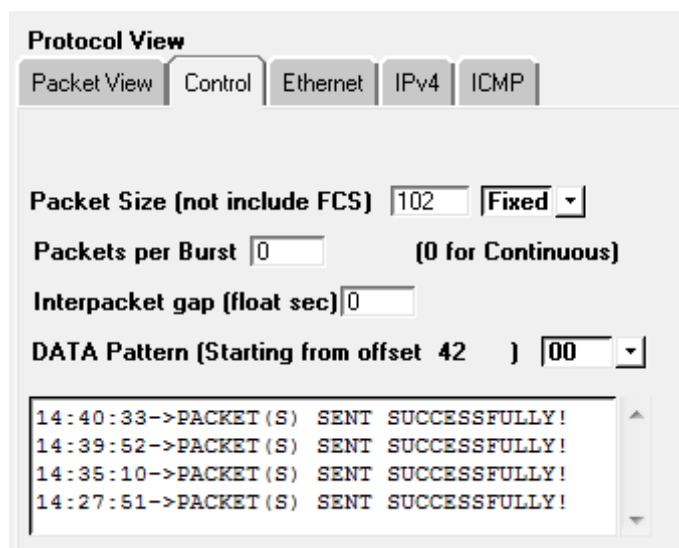
## Analysis ICMP-Type1 packet through Wireshark

From given below image you can observe that wireshark has captured exactly same information which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP message type "Reserved" packets and other information.

ip.addr == 192.168.1.107						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.2	192.168.1.107	ICMP	100	Reserved
2	0.000161	192.168.1.2	192.168.1.107	ICMP	100	Reserved
3	0.000269	192.168.1.2	192.168.1.107	ICMP	100	Reserved
4	0.000372	192.168.1.2	192.168.1.107	ICMP	100	Reserved
5	0.000704	192.168.1.2	192.168.1.107	ICMP	100	Reserved

When the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 1 is successfully sent on target machine.



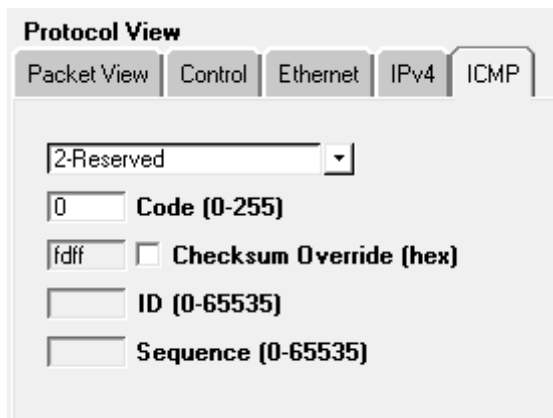
## Message TYPE 2 ICMP Packet Crafting

Again the Packet Assembly phase and Packet Editing phase for ICMP packet crafting is almost same as above only the difference is make in change ICMP message through which connection will be established with target network.

Since we want to send traffic through only message type 2 packets for establishing connection with target network therefore select **Type 2 Reserved** from given list.

Once everything is edited then your packet is ready to send on target network.

Click on **play** button for sending packet on target's network.



## Capturing ICMP-Type2 packet through IDS

From given below image you can observe that inside the file “icmp-info.rules” an alert rule is already implemented for capturing the traffic of **ICMP unassigned type 2** packet is found in network.

Now turn on IDS mode of snort by executing given below command in terminal:

**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0**

```
*icmp-info.rules x
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP unassigned type 2";
icode:0; itype:2; classtype:misc-activity; sid:460; rev:7;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP unassigned type 2
undefined code"; itype:2; classtype:misc-activity; sid:461; rev:7;)|
```

So when again our IDS received any matching packets defined in its file of rules then generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for “ICMP unassigned type 2” packets from source address 192.168.1.1.2 to destination 192.168.1.107.

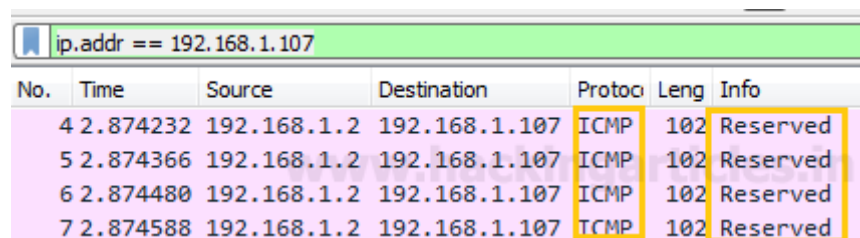
```

01/01-01:12:17.687896  [**] [1:460:7] ICMP unassigned type 2 [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-01:12:17.688933  [**] [1:461:7] ICMP unassigned type 2 undefined code [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107

```

## Analysis ICMP-Type2 packet through Wireshark

Here also the wireshark has captured exactly same information as per our prediction and fetch same details which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP message “Reserved” packet and other information.



No.	Time	Source	Destination	Protocol	Length	Info
4	2.874232	192.168.1.2	192.168.1.107	ICMP	102	Reserved
5	2.874366	192.168.1.2	192.168.1.107	ICMP	102	Reserved
6	2.874480	192.168.1.2	192.168.1.107	ICMP	102	Reserved
7	2.874588	192.168.1.2	192.168.1.107	ICMP	102	Reserved

Again when the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 2 is successfully sent on target machine.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

Packet Size (not include FCS) 120 Fixed

Packets per Burst 0 (0 for Continuous)

Interpacket gap (float sec) 0

DATA Pattern (Starting from offset 42 ) 00

```

14:42:17->PACKET(S) SENT SUCCESSFULLY!
14:42:5->PACKET(S) SENT SUCCESSFULLY!
14:40:33->PACKET(S) SENT SUCCESSFULLY!
14:39:52->PACKET(S) SENT SUCCESSFULLY!
14:38:10->PACKET(S) SENT SUCCESSFULLY!

```

## Message TYPE 3 ICMP Packet Crafting

Now we want to send traffic through message type 3 packets for establishing connection with target network therefore select **Type 3 Destination Unreachable** from given list.

Once everything is edited then your packet is ready to send on target network.

Click on **play** button given in menu bar for sending packet on target's network.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

3-Destination Unreachable

0 Code (0-255)

f0ff ☐ Checksum Override (hex)

ID (0-65535)

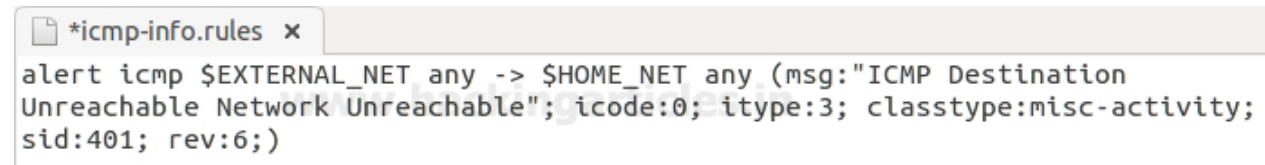
Sequence (0-65535)

## Capturing ICMP-Type3 packet through IDS

From given below image you can observe that inside the file “icmp-info.rules” an alert rule is already implemented for capturing the traffic of **ICMP Destination Unreachable Network Unreachable** packet when found in network.

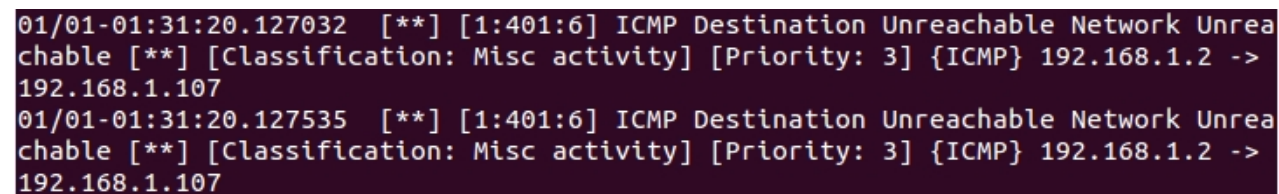
Now turn on IDS mode of snort by executing given below command in terminal:

**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0**



```
*icmp-info.rules x
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:'ICMP Destination
Unreachable Network Unreachable'; icode:0; itype:3; classtype:misc-activity;
sid:401; rev:6;)
```

As said above so when IDS received any matching packets defined in file of rules then generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for “ICMP Destination Unreachable Network Unreachable” packets from source address 192.168.1.1.2 to destination 192.168.1.107.



```
01/01-01:31:20.127032  [**] [1:401:6] ICMP Destination Unreachable Network Unrea
chable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 ->
192.168.1.107
01/01-01:31:20.127535  [**] [1:401:6] ICMP Destination Unreachable Network Unrea
chable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 ->
192.168.1.107
```

## Analysis ICMP-Type3 packet through Wireshark

From given below image you can observe that wireshark has captured exactly same information which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP message type “Destination Unreachable” (Network Unreachable) packet and other information.

ip.addr == 192.168.1.107						
No.	Time	Source	Destination	Protoc	Leng	Info
1	0.000000	192.168.1.2	192.168.1.107	ICMP	120	Destination unreachable (Network unreachable)
2	0.001012	192.168.1.2	192.168.1.107	ICMP	120	Destination unreachable (Network unreachable)
3	0.002216	192.168.1.2	192.168.1.107	ICMP	120	Destination unreachable (Network unreachable)
4	0.003443	192.168.1.2	192.168.1.107	ICMP	120	Destination unreachable (Network unreachable)
5	0.004673	192.168.1.2	192.168.1.107	ICMP	120	Destination unreachable (Network unreachable)

Again when the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 3 is successfully sent on target machine.

**Protocol View**

Packet View
 Control
 Ethernet
 IPv4
 ICMP

Packet Size (not include FCS)
 70
 Fixed

Packets per Burst
 0
 (0 for Continuous)

Interpacket gap (float sec)
 0

DATA Pattern (Starting from offset 42 )
 00

15:1:20->PACKET(S) SENT SUCCESSFULLY!  
 14:42:17->PACKET(S) SENT SUCCESSFULLY!  
 14:42:5->PACKET(S) SENT SUCCESSFULLY!  
 14:40:33->PACKET(S) SENT SUCCESSFULLY!  
 14:38:53->PACKET(S) SENT SUCCESSFULLY!

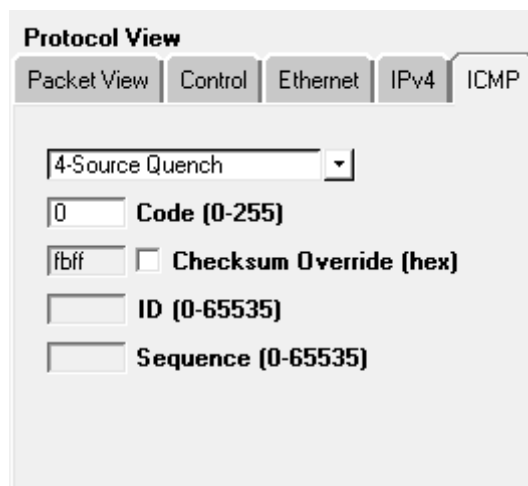
## Message TYPE 4 ICMP Packet Crafting

So the Packet Assembly phase and Packet Editing phase for ICMP packet crafting is almost same as above only the difference is make in change ICMP message through which connection will be established with target network.

Since we want to send traffic through message type 4 packets for establishing connection with target network therefore select **Type 4 Source Quench** from given list.

Once everything is edited then your packet is ready to send on target network.

**Click** on **play** button given in menu bar for sending packet on target's network which known as "packet playing" phase of packet Crafting operation.



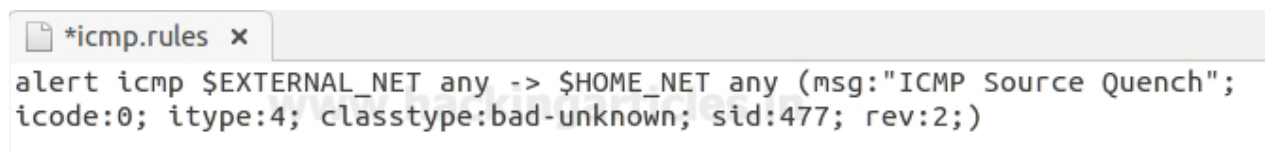
The image shows a 'Protocol View' window with tabs for Packet View, Control, Ethernet, IPv4, and ICMP. The ICMP tab is selected. Below the tabs, there is a dropdown menu showing '4-Source Quench'. Below the dropdown, there is a text box for 'Code (0-255)' with the value '0'. Below that, there is a text box for 'Checksum Override (hex)' with the value 'fbff' and an unchecked checkbox. Below that, there is a text box for 'ID (0-65535)' and a text box for 'Sequence (0-65535)'.

## Capturing ICMP-Type4 packet through IDS

From given below image you can observe that inside the file "icmp-info.rules" an alert rule is already implemented for capturing the traffic of **ICMP Source Quench** packet when found in network.

Now turn on IDS mode of snort by executing given below command in terminal:

**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0**



The image shows a terminal window with a tab labeled '\*icmp.rules x'. The terminal displays the following rule configuration: 

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Source Quench";  
  icode:0; itype:4; classtype:bad-unknown; sid:477; rev:2;)
```

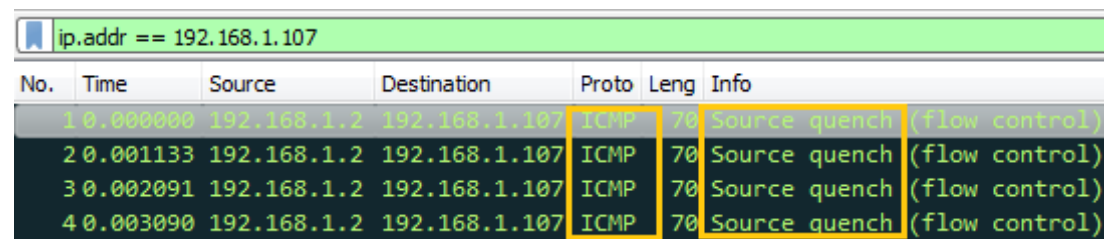


So when IDS received any matching packets defined in file of rules then generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for "ICMP Source Quench" packets from source address 192.168.1.1.2 to destination 192.168.1.107.

```
01/01-01:36:14.741051  [**] [1:477:2] ICMP Source Quench [**] [Classification: P
otentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-01:36:14.741142  [**] [1:477:2] ICMP Source Quench [**] [Classification: P
otentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.2 -> 192.168.1.107
```

## Analysis ICMP-Type4 packet through Wireshark

Here also the wireshark has captured exactly same information as per our prediction and fetch same details which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP message type "Source quench" packet and other information.



No.	Time	Source	Destination	Proto	Leng	Info
1	0.000000	192.168.1.2	192.168.1.107	ICMP	70	Source quench (flow control)
2	0.001133	192.168.1.2	192.168.1.107	ICMP	70	Source quench (flow control)
3	0.002091	192.168.1.2	192.168.1.107	ICMP	70	Source quench (flow control)
4	0.003090	192.168.1.2	192.168.1.107	ICMP	70	Source quench (flow control)

Again when the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 4 is successfully sent on target machine.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

www.hackingarticles.in

Packet Size (not include FCS) 80 Fixed

Packets per Burst 0 (0 for Continuous)

Interpacket gap (float sec) 0

DATA Pattern (Starting from offset 42 ) 00

```

15:6:14->PACKET(S) SENT SUCCESSFULLY!
15:1:20->PACKET(S) SENT SUCCESSFULLY!
14:42:17->PACKET(S) SENT SUCCESSFULLY!
14:42:5->PACKET(S) SENT SUCCESSFULLY!
14:40:33->PACKET(S) SENT SUCCESSFULLY!
  
```

## Message TYPE 5 ICMP Packet Crafting

We want to send traffic through message type 5 packets for establishing connection with target network therefore select **Type 5 Redirect** from given list.

Once everything is edited then your packet is ready to send on target network.

Click on **play** button given in menu bar for sending packet on target's network.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

5-Redirect

0 Code (0-255)

faff ☐ Checksum Override (hex)

ID (0-65535)

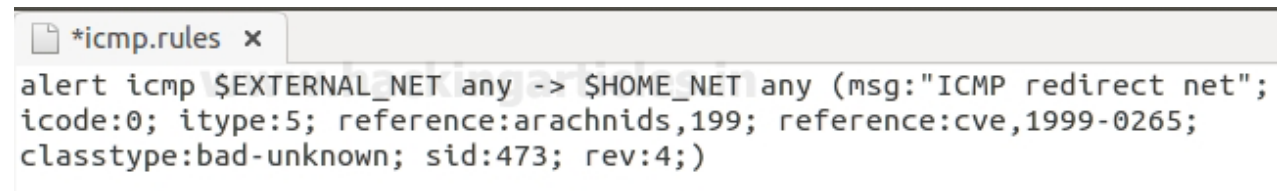
Sequence (0-65535)

## Capturing ICMP-Type5 packet through IDS

As given in below image you can observe that inside the file “icmp-info.rules” an alert rule is already implemented for capturing the traffic of **ICMP redirect net** packet when found in network.

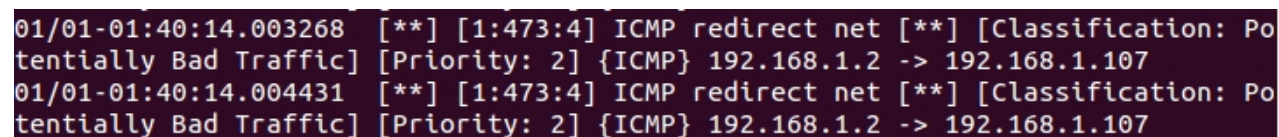
Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



```
*icmp.rules x
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP redirect net";
icode:0; itype:5; reference:arachnids,199; reference:cve,1999-0265;
classtype:bad-unknown; sid:473; rev:4;)
```

So when IDS received any matching packets defined in file of rules then generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for “ICMP Redirect net” packets from source address 192.168.1.1.2 to destination 192.168.1.107.



```
01/01-01:40:14.003268  [**] [1:473:4] ICMP redirect net [**] [Classification: Po
tentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-01:40:14.004431  [**] [1:473:4] ICMP redirect net [**] [Classification: Po
tentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.2 -> 192.168.1.107
```

## Analysis ICMP-Type5 packet through Wireshark

Again as per our prediction Wireshark has captured exactly same information which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP message type “redirect” packet and other information.

ip.addr == 192.168.1.107						
No.	Time	Source	Destination	Proto	Leng	Info
7	2.870592	192.168.1.2	192.168.1.107	ICMP	80	Redirect (Redirect for network)
8	2.870733	192.168.1.2	192.168.1.107	ICMP	80	Redirect (Redirect for network)
9	2.870866	192.168.1.2	192.168.1.107	ICMP	80	Redirect (Redirect for network)

Again when the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 5 is successfully sent on target machine.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

Packet Size (not include FCS)  Fixed

Packets per Burst  (0 for Continuous)

Interpacket gap (float sec)

DATA Pattern (Starting from offset 42 )

```

15:12:50->PACKET(S) SENT SUCCESSFULLY!
15:10:14->PACKET(S) SENT SUCCESSFULLY!
15:6:14->PACKET(S) SENT SUCCESSFULLY!
15:1:20->PACKET(S) SENT SUCCESSFULLY!
14:42:17->PACKET(S) SENT SUCCESSFULLY!
  
```

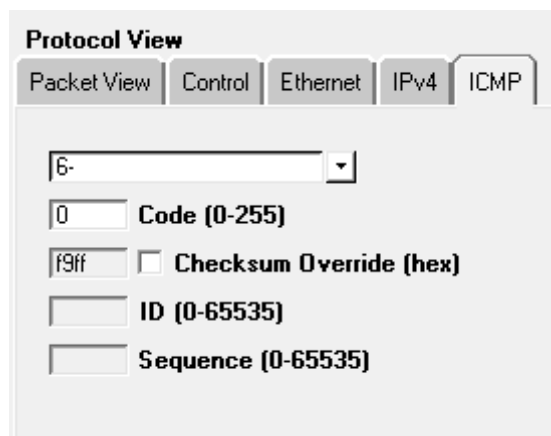
## Message TYPE 6 ICMP Packet Crafting

So the Packet Assembly phase and Packet Editing phase for ICMP packet crafting is almost same as above only the difference is make in change ICMP message through which connection will be established with target network.

Here now next we want to send traffic through message type 6 packets for establishing connection with target network therefore select **Type 6** for Alternate Host Address from given list.

Once everything is edited then your packet is ready to send on target network.

**Click** on **play** button given in menu bar for sending packet on target's network.



The image shows a 'Protocol View' window with tabs for Packet View, Control, Ethernet, IPv4, and ICMP. The ICMP tab is selected. It contains a dropdown menu showing '6-', a 'Code (0-255)' field with '0', a 'Checksum Override (hex)' checkbox with 'f9ff' and an unchecked box, and 'ID (0-65535)' and 'Sequence (0-65535)' fields.

## Capturing ICMP-Type6 packet through IDS

From given below image you can observe that inside the file "icmp-info.rules" an alert rule is already implemented for capturing the traffic of **ICMP Alternate Host Address** packet is found in network.

Now turn on IDS mode of snort by executing given below command in terminal:

**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0**



The image shows a terminal window with a file named '\*icmp-info.rules' open. The rule configuration is as follows:

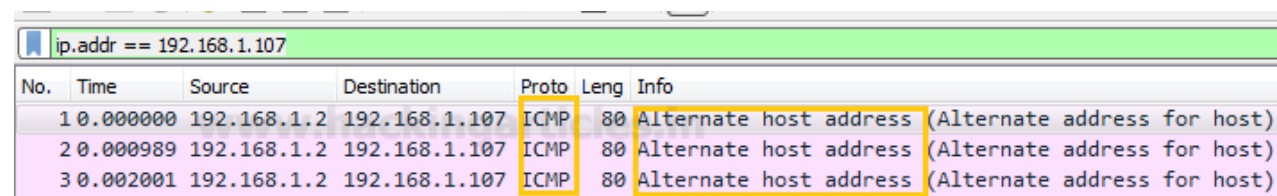
```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Alternate Host Address"; icode:0; itype:6; classtype:misc-activity; sid:390; rev:5;)
```

So when IDS received any matching packets defined in file of rules then generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for “**ICMP Alternate Host Address**” packets from source address 192.168.1.1.2 to destination 192.168.1.107.

```
01/01-01:42:50.823642  [**] [1:390:5] ICMP Alternate Host Address [**] [Classifi
cation: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-01:42:50.823710  [**] [1:390:5] ICMP Alternate Host Address [**] [Classifi
cation: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
```

## Analysis ICMP-Type6 packet through Wireshark

From given below image you can observe that wireshark has captured exactly same information which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP message type “Alternate Host Address” packet and other information.



No.	Time	Source	Destination	Proto	Leng	Info
1	0.000000	192.168.1.2	192.168.1.107	ICMP	80	Alternate host address (Alternate address for host)
2	0.000989	192.168.1.2	192.168.1.107	ICMP	80	Alternate host address (Alternate address for host)
3	0.002001	192.168.1.2	192.168.1.107	ICMP	80	Alternate host address (Alternate address for host)

Again when the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 6 is successfully sent on target machine.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

Packet Size (not include FCS)  Fixed

Packets per Burst  (0 for Continuous)

Interpacket gap (float sec)

DATA Pattern (Starting from offset 42 )

```

15:16:43->PACKET(S) SENT SUCCESSFULLY!
15:12:50->PACKET(S) SENT SUCCESSFULLY!
15:10:14->PACKET(S) SENT SUCCESSFULLY!
15:6:14->PACKET(S) SENT SUCCESSFULLY!
15:1:20->PACKET(S) SENT SUCCESSFULLY!
  
```

## Message TYPE 7 ICMP Packet Crafting

Again Repeat the same and send traffic through message type 7 packets for establishing connection with target network therefore select **Type 7** for **Unassigned** from given list.

Once everything is edited then your packet is ready to send on target network.

Click on **play** button given in menu bar for sending packet on target's network.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

7-

Code (0-255)

☐ Checksum Override (hex)

ID (0-65535)

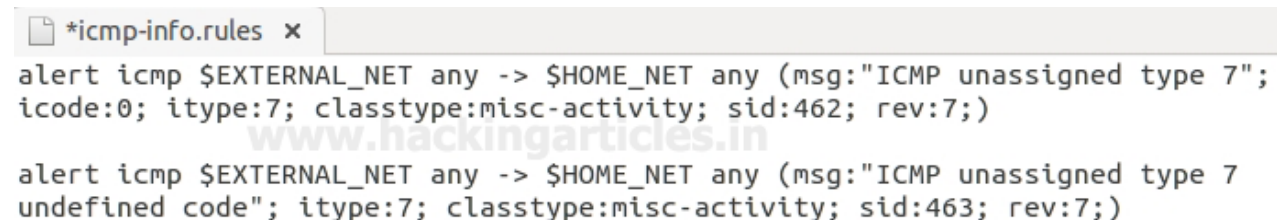
Sequence (0-65535)

## Capturing ICMP-Type7 packet through IDS

From given below image you can observe that inside the file “icmp-info.rules” an alert rule is already implemented for capturing the traffic of **ICMP Alternate Host Address** packet is found in network.

Now turn on IDS mode of snort by executing given below command in terminal:

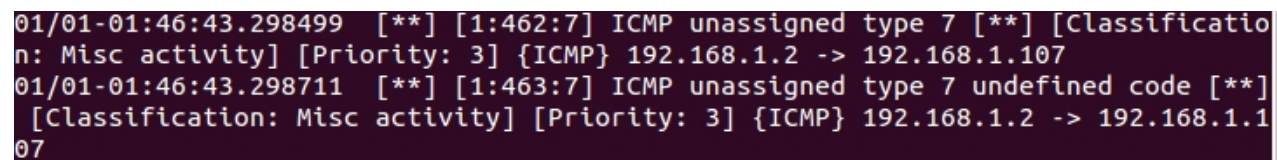
**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0**



```
*icmp-info.rules x
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP unassigned type 7";
icode:0; itype:7; classtype:misc-activity; sid:462; rev:7;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP unassigned type 7
undefined code"; itype:7; classtype:misc-activity; sid:463; rev:7;)
```

Therefore when IDS received any matching packets described in file of rules then it will generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for “ICMP unassigned type 7” packets from source address 192.168.1.1.2 to destination 192.168.1.107.



```
01/01-01:46:43.298499  [**] [1:462:7] ICMP unassigned type 7 [**] [Classificatio
n: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-01:46:43.298711  [**] [1:463:7] ICMP unassigned type 7 undefined code [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.1
07
```

## Analysis ICMP-Type7 packet through Wireshark

Wireshark has captured exactly same information which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP unknown message type “obsolete or malformed” packet and other information.



ip.addr == 192.168.1.107						
No.	Time	Source	Destination	Proto	Leng	Info
1	0.000000	192.168.1.2	192.168.1.107	ICMP	95	Unknown ICMP (obsolete or malformed?)
2	0.001031	192.168.1.2	192.168.1.107	ICMP	95	Unknown ICMP (obsolete or malformed?)

Again when the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 7 is successfully sent on target machine.

**Protocol View**

Packet View
 Control
 Ethernet
 IPv4
 ICMP

Packet Size (not include FCS)
 100
 Fixed

Packets per Burst
 0
 (0 for Continuous)

Interpacket gap (float sec)
 0

DATA Pattern (Starting from offset 42 )
 00

```

15:16:43->PACKET(S) SENT SUCCESSFULLY!
15:12:50->PACKET(S) SENT SUCCESSFULLY!
15:10:14->PACKET(S) SENT SUCCESSFULLY!
15:6:14->PACKET(S) SENT SUCCESSFULLY!
15:1:30->PACKET(S) SENT SUCCESSFULLY!

```

## Message TYPE 8 ICMP Packet Crafting

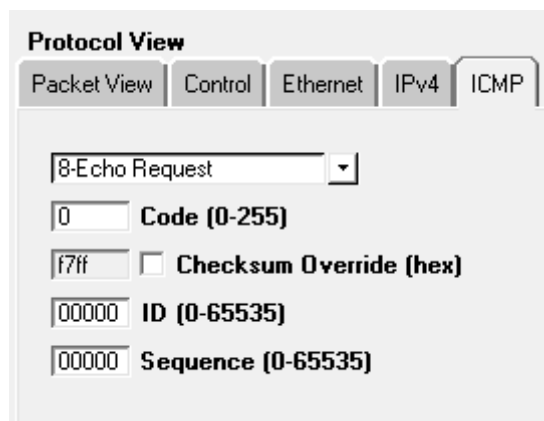
Since we want to send traffic through message type 8 packets for establishing connection with target network therefore select **Type 8** for **ICMP echo Request** from given list.

This step is very useful because it will craft a packet will send ICMP Request packet on target's network to test the strength of IDS and Firewall.

Infinite packet ICMP Request packet is consider as ICMP Flood or Ping of Death Attack when sent only network therefore we can check our IDS and Firewall Strength against such DOS attack through this packet crafting.

Once everything is edited then your packet is ready to send on target network.

**Click on play** button given in menu bar for sending packet on target's network.



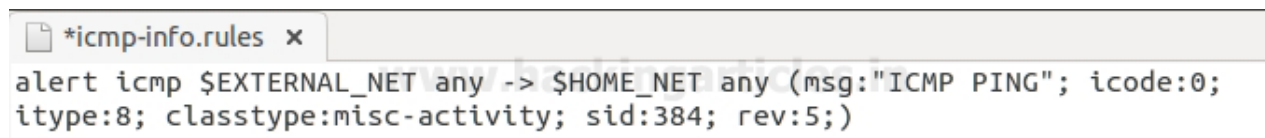
The image shows a 'Protocol View' window with tabs for Packet View, Control, Ethernet, IPv4, and ICMP. The ICMP tab is selected. It contains a dropdown menu set to '8-Echo Request', a 'Code (0-255)' field with '0', a 'Checksum Override (hex)' checkbox which is unchecked and has 'f7ff' in the adjacent field, an 'ID (0-65535)' field with '00000', and a 'Sequence (0-65535)' field with '00000'.

## Capturing ICMP-Type8 packet through IDS

From given below image you can observer that inside the file “icmp-info.rules” an alert rule is already implemented for capturing the traffic of **ICMP Ping** packet is found in network. As we know ICMP echo Request packet is consider as Ping request packet which sends request to a network IP for establishing connection with it.

Now turn on IDS mode of snort by executing given below command in terminal:

**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0**



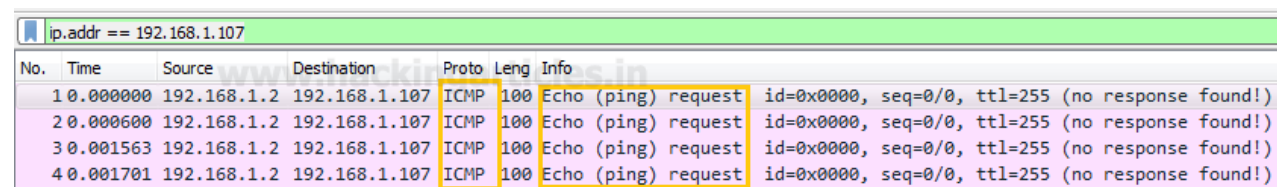
The image shows a terminal window with a tab titled '\*icmp-info.rules x'. The terminal displays the following alert rule: `alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING"; icode:0; itype:8; classtype:misc-activity; sid:384; rev:5;)`

So when IDS received any matching packets defined in file of rules then generates an alert for captured packet. From given below image you can observe that an alert is generated by snort for "ICMP Ping" packets from source address 192.168.1.1.2 to destination 192.168.1.107.

```
01/01-01:52:18.894554  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-01:52:18.894739  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
```

### Analysis ICMP-Type8 packet through Wireshark

From given below image you can observe that wireshark has captured Ping packet for ICMP Echo request as described above, exactly same information which we had bind in packet such as ICMP protocol, ICMP Ping request message packet and other information.



No.	Time	Source	Destination	Proto	Leng	Info
1	0.000000	192.168.1.2	192.168.1.107	ICMP	100	Echo (ping) request id=0x0000, seq=0/0, ttl=255 (no response found!)
2	0.000600	192.168.1.2	192.168.1.107	ICMP	100	Echo (ping) request id=0x0000, seq=0/0, ttl=255 (no response found!)
3	0.001563	192.168.1.2	192.168.1.107	ICMP	100	Echo (ping) request id=0x0000, seq=0/0, ttl=255 (no response found!)
4	0.001701	192.168.1.2	192.168.1.107	ICMP	100	Echo (ping) request id=0x0000, seq=0/0, ttl=255 (no response found!)

Again when the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 8 is successfully sent on target machine.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

Packet Size (not include FCS)

Packets per Burst  (0 for Continuous)

Interpacket gap (float sec)

DATA Pattern (Starting from offset 42 )

```

15:22:19->PACKET(S) SENT SUCCESSFULLY!
15:16:43->PACKET(S) SENT SUCCESSFULLY!
15:12:50->PACKET(S) SENT SUCCESSFULLY!
15:10:14->PACKET(S) SENT SUCCESSFULLY!
15:6:14->PACKET(S) SENT SUCCESSFULLY!
  
```

## Message TYPE 9 ICMP Packet Crafting

Now at last we want to send traffic through message type 9 packets for establishing connection with target network therefore select **Type 9** for **router Advertisement** from given list.

Once everything is edited then your packet is ready to send on target network.

**Click** on **play** button given in menu bar for sending packet on target's network.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

9-Router Advertisement

0 Code (0-255)

f6ff ☐ Checksum Override (hex)

ID (0-65535)

Sequence (0-65535)

## Capturing ICMP-Type9 packet through IDS

From given below image you can observe that inside the file “icmp-info.rules” an alert rule is already implemented for capturing the traffic of **ICMP router Advertisement** packet is found in network.

Now turn on IDS mode of snort by executing given below command in terminal:

**sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0**

```
*icmp-info.rules x
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP IRDP router
advertisement"; itype:9; reference:arachnids,173; reference:bugtraq,578;
reference:cve,1999-0875; classtype:misc-activity; sid:363; rev:7;)

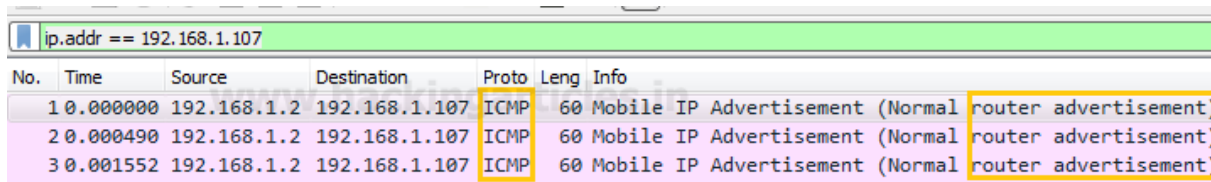
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Router
Advertisement"; icode:0; itype:9; reference:arachnids,173; classtype:misc-
activity; sid:441; rev:6;)
```

So when IDS received any matching packets defined in file of rules then generate an alert for captured packet. From given below image you can observe that an alert is generated by snort for “ICMP router Advertisement” packets from source address 192.168.1.1.2 to destination 192.168.1.107.

```
01/01-01:55:38.895682  [**] [1:363:7] ICMP IRDP router advertisement [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
01/01-01:55:38.896833  [**] [1:441:6] ICMP Router Advertisement [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.107
```

## Analysis ICMP-Type9 packet through Wireshark

From given below image you can observe that wireshark has captured exactly same information which we had bind in packet during packet Assembly and packet Editing mode such as ICMP protocol, ICMP P ICMP router Advertisement message packet and other information.



The image shows a Wireshark packet capture window with a filter set to 'ip.addr == 192.168.1.107'. The packet list shows three packets, all of which are ICMP Router Advertisements. The 'Info' column for each packet is highlighted with a yellow box, showing 'router advertisement'.

No.	Time	Source	Destination	Proto	Leng	Info
10.000000	192.168.1.2	192.168.1.107	ICMP	60	Mobile IP Advertisement (Normal	router advertisement
20.000490	192.168.1.2	192.168.1.107	ICMP	60	Mobile IP Advertisement (Normal	router advertisement
30.001552	192.168.1.2	192.168.1.107	ICMP	60	Mobile IP Advertisement (Normal	router advertisement

Again when the tester will click on **Stop button**, he will receive the status of sent packet either as successful or as failed.

From given below image you can perceive that our ICMP Type 9 is successfully sent on target machine.

**Protocol View**

Packet View Control Ethernet IPv4 ICMP

Packet Size (not include FCS)

Packets per Burst  (0 for Continuous)

Interpacket gap (float sec)

DATA Pattern (Starting from offset 42 )

```
15:22:19->PACKET(S) SENT SUCCESSFULLY!
15:16:43->PACKET(S) SENT SUCCESSFULLY!
15:12:50->PACKET(S) SENT SUCCESSFULLY!
15:10:14->PACKET(S) SENT SUCCESSFULLY!
15:6:14->PACKET(S) SENT SUCCESSFULLY!
```

**Author: Rahul Virmani** is a Certified Ethical Hacker and the researcher in the field of network Penetration Testing (CYBER SECURITY). Contact [Here](#)

Share this:



Like this:

Loading...

## ABOUT THE AUTHOR

---



### RAJ CHANDEL

Raj Chandel is a Skilled and Passionate IT Professional especially in IT-Hacking Industry. At present other than his name he can also be called as An Ethical Hacker, A Cyber Security Expert, A Penetration Tester. With years of quality Experience in IT and software industry

---

#### PREVIOUS POST

← TCP & UDP PACKET CRAFTING  
WITH CATKARAT

#### NEXT POST

HACK THE BASIC PENETRATION VM  
(BOOT2ROOT CHALLENGE) →

## Leave a Reply

Your email address will not be published. Required fields are marked \*

Comment



Name \*

Email \*

Website

☐

Save my name, email, and website in this browser for the next time I comment.

**POST COMMENT**

☐

Notify me of follow-up comments by email.

☐

Notify me of new posts by email.

