INTRUSION DETECTION USING SNORT

Gaurav CG *R Siva Girish*

INTRUSION DETECTION SYSTEMS

What is Intrusion?

Intrusion is unauthorised system or network activity performed on one or more computers in a network. E.g. Trying to escalate one's privileges in a system (accessing the root).

What is Intrusion Detection System?

The art of detecting the intrusions attempted on the systems in a network in order to prevent or alert the required administrator is done by an Intrusion Detection System.

Types of IDS

1. Network Based Intrusion Detection System
2. Host based Intrusion Detection System
3. Distributed Intrusion Detection System

Network based IDS

NIDS is used to monitor an entire network segment or subnet. This is done by setting your Network Interface Card to promiscuous mode which let NIDS read all the packets that is sent and received from the network. The advantage of NIDS is that it does not load the host machine and an attacker might not even know of its presence.

Host based IDS

HIDS differs from NIDS in two ways. First, an installed HIDS protects only the system on which it is residing not the entire network. Second the Network Interface Card runs on non-promiscuous mode.

Distributed IDS

DIDS is a combination of both NIDS and HIDS distribution across the network all reporting to a central system.

How IDS watch the network?

There are many ways IDS watches the data and collects it for further analysis they are:

Packet Sniffing

IDS capture each packet that crosses the wire on the local subnet. This is achieved by setting promiscuous mode. Packet sniffing is a classic way of doing intrusion detection, and there are equally classic techniques of IDS evasion that can be used against packet sniffing IDS.

Log Parsing

Another source of security data is from system logs. IDS look over all system logs and alerts if there is any suspicious change in the log files.

System Call monitoring

A system call is a request that a program makes of the operating system kernel. If an IDS (specifically HIDS) detects malicious system call like changing user ID to root, it creates an alert or disallow such system call.

Filesystem Monitoring

IDS (specifically HIDS) monitors the size and attributes of crucial files in the system. It alerts when there is a sudden change in the size of such files and if the system administrator is unaware of it.

SNORT

What is Snort?

Snort is an open source network intrusion detection and prevention system. An IDS is a tool that monitors network traffic as well as analyses network packets from the log files of routers, firewalls and servers.

An IDS has the capability of storing the signatures of known attackers in a database and compare patterns of activity, traffic or behaviour it observes in the data it's monitoring. By doing this sort of a comparison an IDS can monitor potential threats as well as isolate attackers from normal users.

Snort can operate in 3 modes~:

- ❖ Packet Sniffer
  - ♦ Packet sniffing is the act of capturing packets of data flowing across a computer network.
  - ♦ snort –v
- ❖ Packet Logger
  - ♦ Packet logger makes a copy of the packets transmitted in a network
  - ♦ Snort –l /usr/local/log/snort
- ❖ Network Intrusion detection System (NIDS)

INSTALLING SNORT

It is preferable to install snort in Linux operating system as it is available in the default packages and makes configuration easier. For windows operating system skip to the other section.

Steps to install snort in Linux

Step 1: It is recommended to update all packages and dependencies

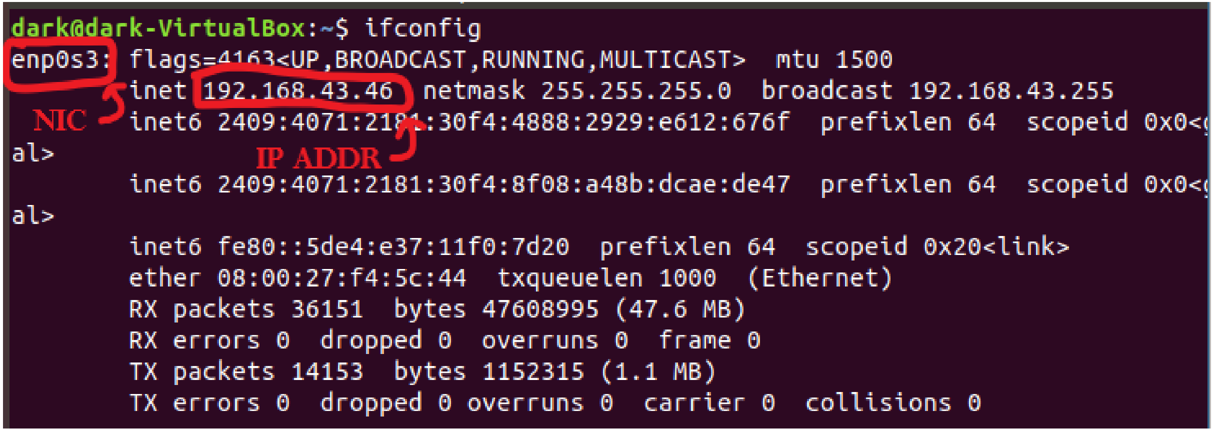
```
sudo apt-get update
```

Step 2: Installing snort using the command

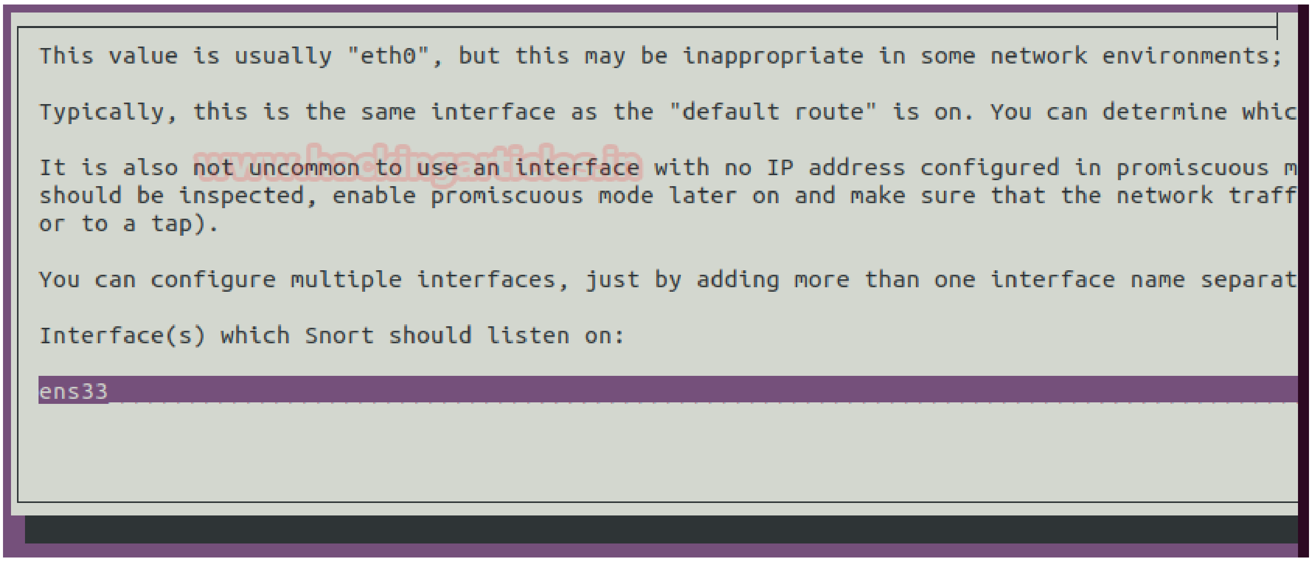
```
sudo apt-get install snort
```

This step will install snort along with all its dependencies.

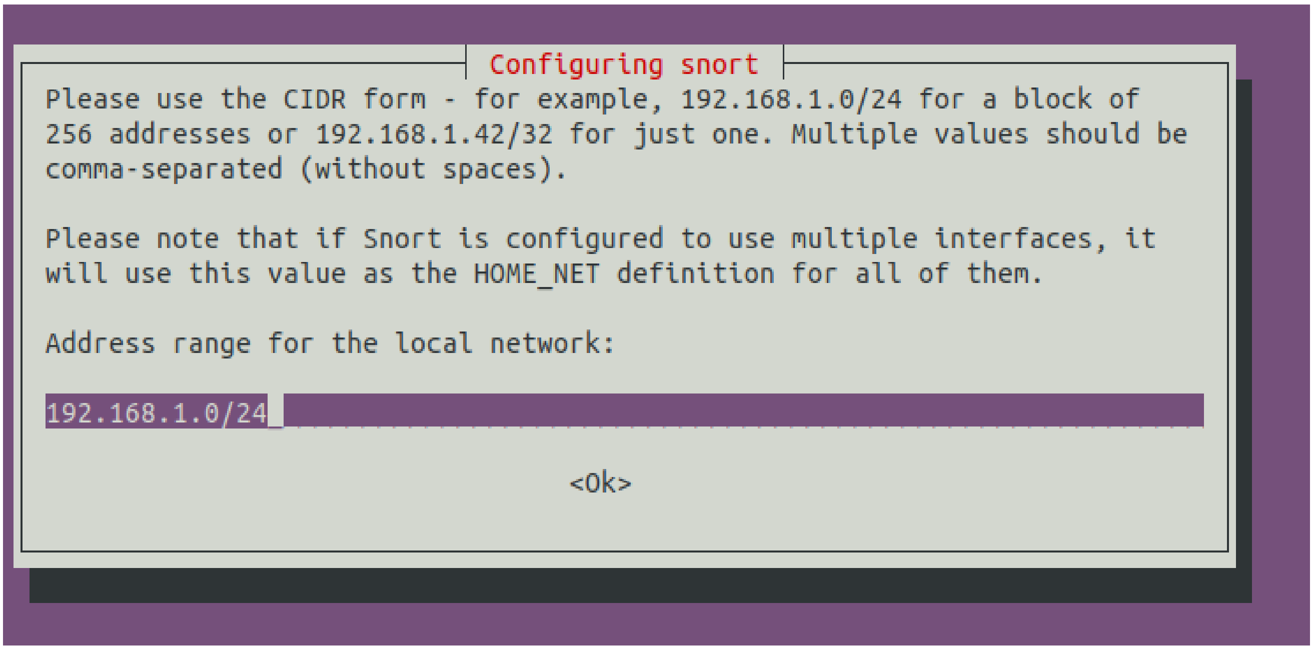
During installation the Network Interface Card and home IP address need to be provided. This can be found by the command 'ifconfig'.



```
dark@dark-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.46  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 2409:4071:2181:30f4:4888:2929:e612:676f  prefixlen 64  scopeid 0x0<g
al>
        inet6 2409:4071:2181:30f4:8f08:a48b:dcae:de47  prefixlen 64  scopeid 0x0<g
al>
        inet6 fe80::5de4:e37:11f0:7d20  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f4:5c:44  txqueuelen 1000  (Ethernet)
        RX packets 36151  bytes 47608995 (47.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14153  bytes 1152315 (1.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



```
This value is usually "eth0", but this may be inappropriate in some network environments;

Typically, this is the same interface as the "default route" is on. You can determine whic

It is also not uncommon to use an interface with no IP address configured in promiscuous m
should be inspected, enable promiscuous mode later on and make sure that the network traff
or to a tap).

You can configure multiple interfaces, just by adding more than one interface name separat

Interface(s) which Snort should listen on:

ens33
```



```
Configuring snort
Please use the CIDR form - for example, 192.168.1.0/24 for a block of
256 addresses or 192.168.1.42/32 for just one. Multiple values should be
comma-separated (without spaces).

Please note that if Snort is configured to use multiple interfaces, it
will use this value as the HOME_NET definition for all of them.

Address range for the local network:

192.168.1.0/24

                              <Ok>
```

Step 3: By default, snort will require superuser privileges to run. It is advised not to run snort as superuser.

A) Create a new user and group to run snort

```
    sudo adduser snort –system –group
```

This creates a system user with no login and with the group name same as the username.

*CIDR Notation: 192.168.1.0/24 = 192.162.1.0 255.255.255.0 (24 is the IP network Prefix)

24 indicates the netmask. This number is the number of bits that are set in a 32bit Ip address.

B) Setting permissions for that user to access snort

There are multiple restricted locations snort needs access to those are
/etc/snort - location where snort rules and files are stored.
/var/log/snort - location where snort writes its logs.
/usr/local/lib/snort_dynamicrules – location where additional dynamic rules of snort are stored.
The permissions are set using chmod(change mode)

```
    sudo chmod -R 5775 /etc/snort
    sudo chmod -R 5775 /var/log/snort
    sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules
    sudo chown snort:snort /etc/snort
    sudo chown snort:snort /var/log/snort
    sudo chown snort:snort /usr/local/lib/snort_dynamicrules
```

chown is used to change the ownership from root to snort user.

Step 4: Running snort

Snort can be run using the command snort and additional parameters should also be provided.

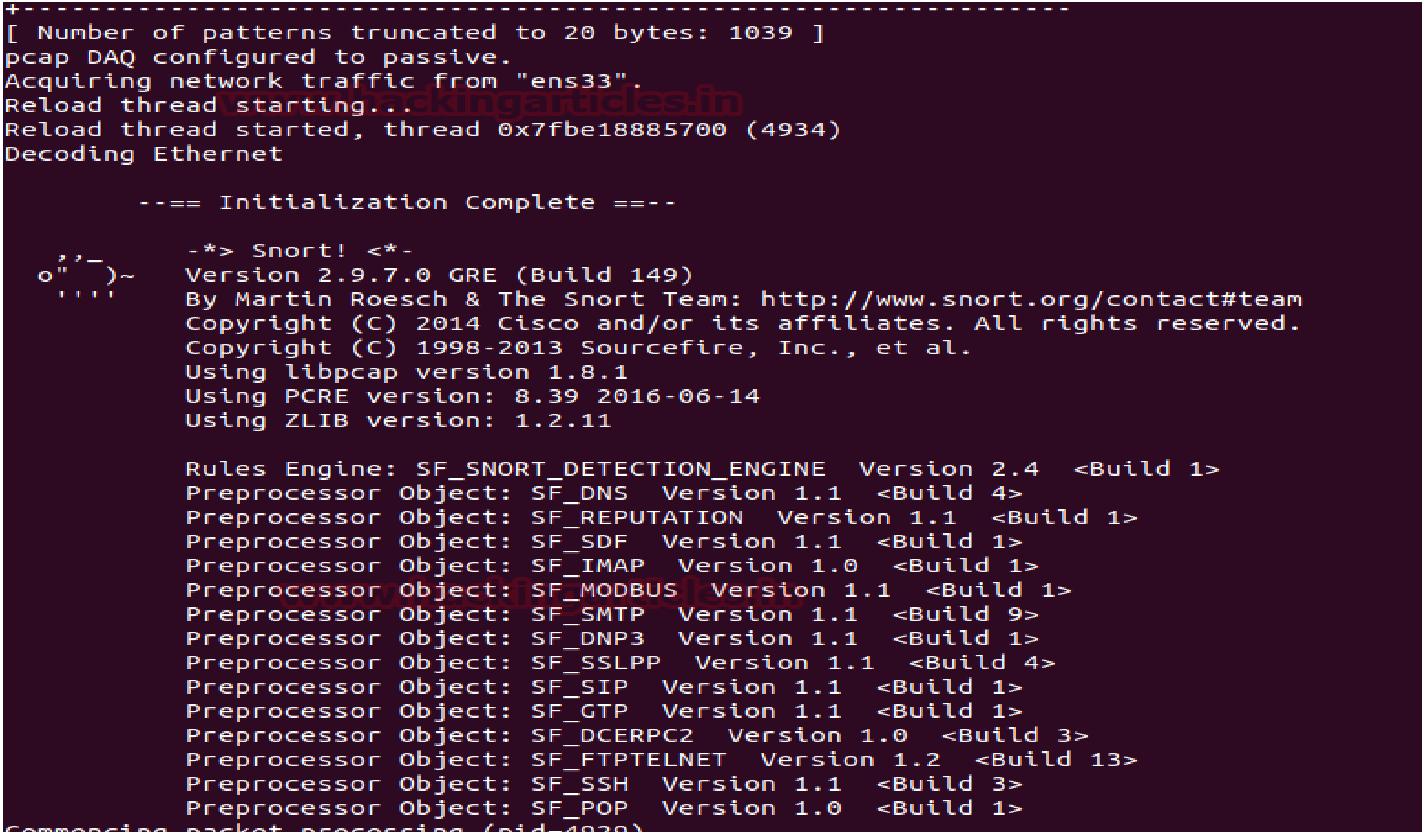
```
snort -i eth0 –u snort –g snort –c /etc/snort/snort.conf -l /var/log/snort
```

~i is used to indicate the network interface

~u & ~g is used to indicate the username and group (both are snort in our case)

~c is used to indicate the conf file directory

~l is used to indicate the location to write the logs



```
+--------------------------------------------------------------
[ Number of patterns truncated to 20 bytes: 1039 ]
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Reload thread starting...
Reload thread started, thread 0x7fbe18885700 (4934)
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.8.1
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.4  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
Commencing packet processing (pid=4939)
```

Running a simple snort rule and testing it

First open the snort conf file present in /etc/snort.
In the conf file locate the section where rules are included, comment all the rule include statements except for local.rules.
Now move to rules folder and open local.rules and start writing your rules.

Snort rule is made of two parts 'the header' and 'the options'.
The header contains information such as action, protocol, source ip and port, network packet direction operator and destination ip and port followed by the options part.
Syntax: `Action Protocol Source IP Source port -> Destination IP Destination port (options)`
Header Fields: -

Action: It informs Snort what kind of action to be performed when it discovers a packet that matches the rule description. There are five existing default job actions in Snort: alert, log, pass, activate, and dynamic are keyword use to define the action of rules. You can also go with additional options which include drop, reject, and sdrop.

Protocol: After deciding the option for action in the rule, you need to describe specific Protocol (IP, TCP, UDP, ICMP, any) on which this rule will be applicable.

Source IP: This part of header describes the sender network interface from which traffic is coming.

Source Port: This part of header describes the source Port from which traffic is coming.

Direction operator ("->", "<>"): It denotes the direction of traffic flow between sender and receiver networks.

Destination IP: This part of header describes the destination network interface in which traffic is coming for establishing the connection.

Destination Port: This part of header describes the destination Port on which traffic is coming for establishing the connection.

Option Fields:

The body for rule option is usually written between circular brackets "()" that contains keywords with their argument and separated by semicolon ";" from another keyword.

There are four major categories of rule options.

General: These options contain metadata that offers information with reference to them.

Payload: These options all come across for data contained by the packet payload and can be interconnected.

Non-payload: These options come across for non-payload data.

Post-detection: These options are rule specific triggers that happen after a rule has fired."

A simple rule is to alert the user if external system tries to ping our computer. This can be written as follows

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"we are
being pinged"; icode:0; itype:8; sid:10001;)
```

The action given here is 'alert', the protocol here is 'icmp' as all ping follow this protocol and `$EXTERNAL_NET` and `$HOME_NET` are the source and destination IPs respectively. Here the value of `$HOME_NET` is the IP address given during installation. The `$EXTERNAL_NET` in our case is anything but our IP address. The port values are given as 'any' because it is usually random. The '->' indicates that the direction is from left to right meaning from source to destination.

In the option field the msg keyword is used to alert the sentence given.

For detecting ping, the icmp values are 0 icode and 8 itype. Every rule should have its own unique id called sid. These rules run as processes and the value should always be greater than 10,000 because system processes have IDs below 10,000.

Testing the rule:

Now run snort with the command,

```
snort -i eth0 –u snort –g snort –c /etc/snort/snort.conf -l /var/log/snort
```

This will start the snort IDS and snort stores its alerts in /var/log/alert file.

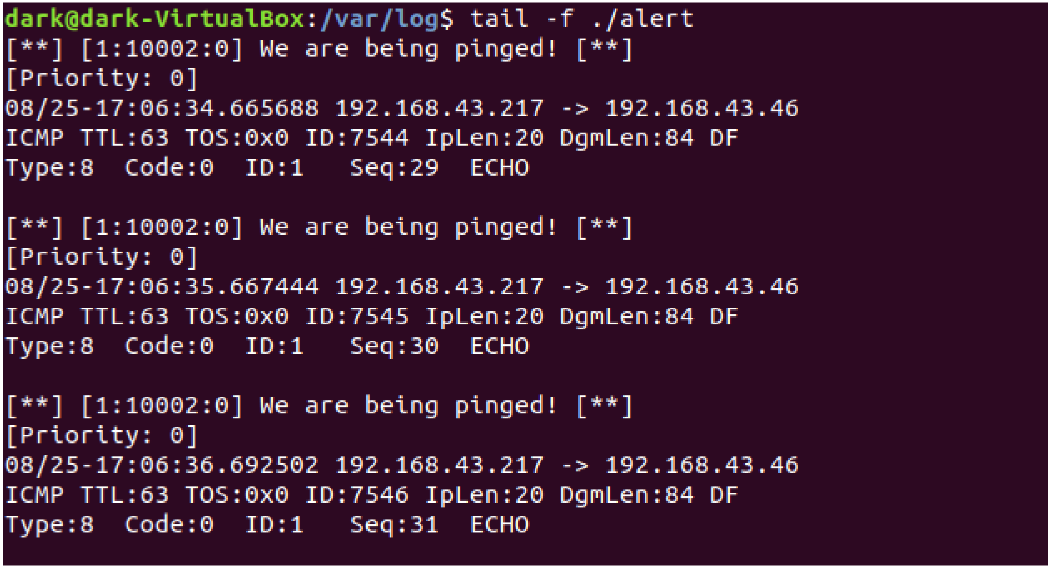
In another terminal locate to /var/log and tail the alert file (tail is used to display the last few lines of the file).

To test your rule, try pinging to your IP address from another machine or VM.

In another VM connected to the same network as your system type

ping your_ip_addr

If your rule is written properly you can see the below output.



```
dark@dark-VirtualBox:/var/log$ tail -f ./alert
[**] [1:10002:0] We are being pinged! [**]
[Priority: 0]
08/25-17:06:34.665688 192.168.43.217 -> 192.168.43.46
ICMP TTL:63 TOS:0x0 ID:7544 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:1   Seq:29  ECHO

[**] [1:10002:0] We are being pinged! [**]
[Priority: 0]
08/25-17:06:35.667444 192.168.43.217 -> 192.168.43.46
ICMP TTL:63 TOS:0x0 ID:7545 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:1   Seq:30  ECHO

[**] [1:10002:0] We are being pinged! [**]
[Priority: 0]
08/25-17:06:36.692502 192.168.43.217 -> 192.168.43.46
ICMP TTL:63 TOS:0x0 ID:7546 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:1   Seq:31  ECHO
```