

telnet – windows

telnet (**te**letype **net**work) is a network protocol for two-way text-based communication through a CLI, allowing remote access. Telnet is vulnerable to cybersecurity attacks because it lacks encryption methods compared to the more modern SSH. However, it is still helpful for tasks that do not involve transmitting sensitive information.

Prerequisites

- Windows OS with administrator privileges
- Access to the command prompt
- An address and port to test

What is Telnet?

Telnet is a client-server protocol predating the TCP protocol. The network protocol allows a user to log into another computer within the same network through a TCP/IP connection.

A client machine running the Telnet client connects to a CLI on a remote device, most commonly a dedicated platform. Telnet is lightweight and fast, making it the preferred option in some use cases:

- Initial network hardware configuration.
- Remote access to trusted internal networks.
- Testing for open or used ports.
- Troubleshooting mail and web servers.

How Does Telnet Work?

The Telnet protocol creates a communication path through a virtual terminal connection. The data distributes in-band with Telnet control information over the transmission control protocol (TCP).

Unlike other TCP/IP protocols, Telnet provides a log-in screen and allows logging in as the remote device's actual user when establishing a connection on port 23. This type of access grants direct control with all the same privileges as the owner of the credentials.

Telnet comes with a command accessible from the command line in Windows.

The `telnet` command also exists for macOS and Linux operating systems.

How to Enable Telnet on Windows 10?

In Windows systems, Telnet is disabled by default. To check if Telnet is already activated, open your command line, and run `telnet`:

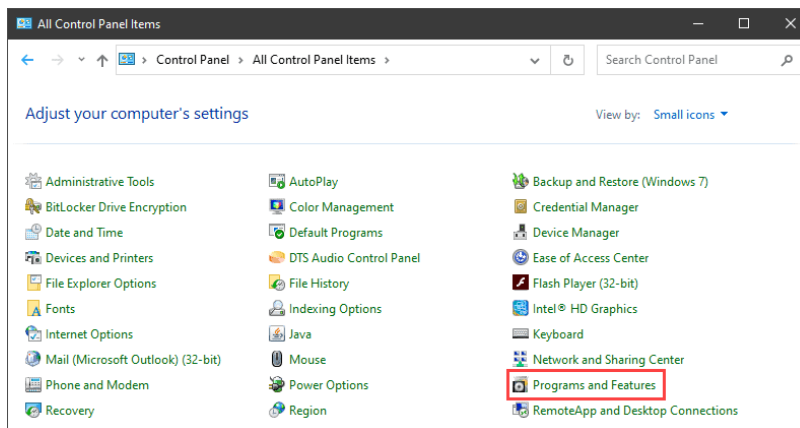
```
C:\Users\User>telnet
'telnet' is not recognized as an internal or external command,
operable program or batch file.
```

If the command prompt does not recognize the command, there are two possible ways to enable the Telnet client in Windows.

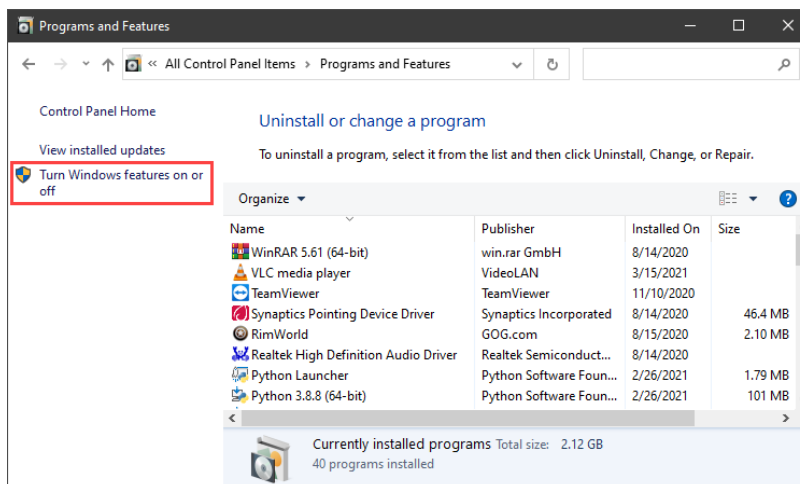
Option 1: Enable Telnet using GUI

To activate the Telnet command using the GUI:

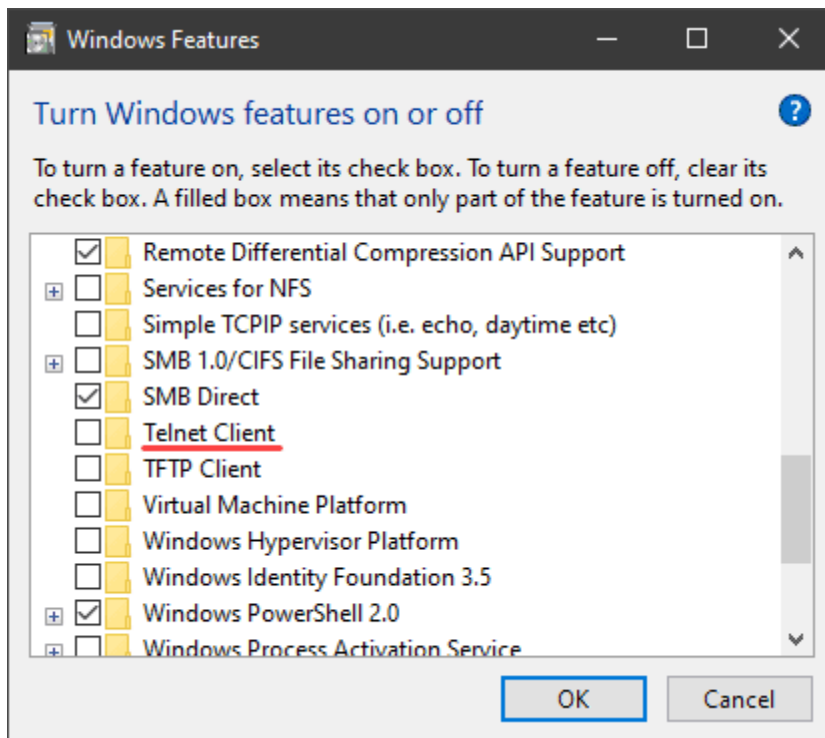
1. Open the **Programs and Features** options in Control Panel:



2. Click the **Turn Windows features on or off** setting:



3. Locate the **Telnet Client** option on the list, select it and click **OK** to install the feature:



4. When Windows completes the requested change, click **Close**.
5. Open the command prompt and run `telnet` to open the Microsoft Telnet Client:

```
Welcome to Microsoft Telnet Client  
Escape Character is 'CTRL+]'  
Microsoft Telnet>
```

6. Run `quit` to exit the Telnet client.

Option 2: Enable Telnet Using Command Prompt

To activate the Telnet client from the command prompt:

1. In the command prompt, run:

```
pkgmgr /iu:"TelnetClient"
```

2. Restart the command prompt and run `telnet` to open the Microsoft Telnet Client.

3. Run `quit` to exit the client:

```
Welcome to Microsoft Telnet Client  
Escape Character is 'CTRL+]'  
Microsoft Telnet> quit  
C:\Users\User>_
```

How to Use Telnet in Windows to Test Open Ports

The Telnet syntax for testing open ports is:

```
telnet <address> <port number>
```

The command accepts both symbolic and numeric addresses. For example:

```
telnet towel.blinkenlights.nl 23
```

Or alternatively:

```
telnet 127.0.0.1 80
```

After running the command, one of the following three options happen:

1. The command throws an error, indicating the port is not available for connection:

```
C:\Users\User>telnet 127.0.0.1 80  
Connecting To 127.0.0.1...Could not open connection to the host, on port 80: Connect failed
```

2. The command goes to a blank screen, indicating the port is available.

3. Running the command on an open port 23 displays the screen of the telnet host, confirming an established Telnet connection:

```
Welcome Visiting Huawei Home Gateway
Copyright by Huawei Technologies Co., Ltd.

Login:
```

For checking HTTP connection on port 80

```
C:\Users\ljmca>
C:\Users\ljmca>telnet 192.168.19.3 80
```

telnet www.javatpoint.com 80

If it displays blank screen that means that port is open.

Conclusion

The Telnet communication protocol provides a way to establish a direct connection with a remote host. Although not a secure option for most tasks, there are use cases where Telnet is a viable option.

Running the telnet client does not require a telnet server on the client end. If you aren't connecting to port 23 (telnet), a telnet server is not required on the remote end either.

People often use a telnet client to test connectivity to services on remote servers to verify they are not blocked by firewalls or not responding at all. All you are doing is making a TCP connection from a client to a server... it will work on any open TCP port which is not blocked.

Note: Verify you have the permissions to telnet from your computer over the network and to another computer. Verify no firewalls between you and the computer or device you tried connecting to are being blocked. Also, verify telnet service is enabled on both computers.

Secure Shell - linux

There are many ways to establish a connection with a remote machine depending on the operating system you are running, but the two most used protocols are:

- Secure Shell (SSH) for Linux-based machines
- Remote Desktop Protocol (RDP) for Windows-based machines

These tools allow you to gain access and remotely manage other computers, transfer files, and do virtually anything you can do while physically sitting in front of the machine.

Before you can **establish a secure remote desktop protocol** with a remote machine, there are a few basic requirements to meet:

- The remote computer must be turned on at all times and have a network connection.

- The client and server applications need to be installed and enabled.
- You need the IP address or the name of the remote machine you want to connect to.
- You need to have the necessary permissions to access the remote computer.
- Firewall settings need to allow the remote connection.

What is SSH?

Secure Shell, sometimes referred to as **Secure Socket Shell**, is a protocol which allows you to connect securely to a remote computer or a server by using a text-based interface.

When a secure **SSH** connection is established, a shell session will be started, and you will be able to manipulate the server by typing commands within the client on your local computer.

System and network administrators use this protocol the most, as well as anyone who needs to manage a computer remotely in a highly secure manner.

How Does SSH Work?

In order to establish an SSH connection, you need two components: a client and the corresponding server-side component. An SSH client is an application you install on the computer which you will use to connect to another computer or a server. The client uses the provided remote host information to initiate the connection and if the credentials are verified, establishes the encrypted connection.

On the server's side, there is a component called an SSH daemon that is constantly listening to a specific TCP/IP port for possible client connection requests. Once a client initiates a connection, the SSH daemon will respond with the software and the protocol versions it supports and the two will exchange their identification data. If the provided credentials are correct, SSH creates a new session for the appropriate environment.

The default SSH protocol version for SSH server and SSH client communication is version 2.

How to Enable an SSH Connection

Since creating an SSH connection requires both a client and a server component, you need to make sure they are installed on the local and the remote machine, respectively. An open source SSH tool—widely used for Linux distributions—is OpenSSH. Installing OpenSSH is relatively easy. It requires access to the terminal on the server and the computer that you use for connecting. Note that Ubuntu does not have SSH server installed by default.

How to Install an OpenSSH Client

Before you proceed with installing an SSH client, make sure it is not already installed. Many Linux distributions already have an SSH client. For Windows machines, you can install PuTTY or any other client of your choice to gain access to a server.

To check if the client is available on your Linux-based system, you will need to:

1. Load an SSH terminal. You can either search for “terminal” or press **CTRL + ALT + T** on your keyboard.
2. Type in **ssh** and press **Enter** in the terminal.
3. If the client is installed, you will receive a response that looks like this:

```
username@host:~$ ssh

usage: ssh [-1246AaCfGgKkMnNqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
[-D [bind_address:]port] [-E log_file] [-e escape_char]
[-F configfile] [-I pkcs11] [-i identity_file]
[-J [user@]host[:port]] [-L address] [-l login_name] [-m mac_spec]
[-O ctl_cmd] [-o option] [-p port] [-Q query_option] [-R address]
[-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
[user@]hostname [command]

username@host:~$
```

This means that you are ready to remotely connect to a physical or virtual machine. Otherwise, you will have to install the OpenSSH client:

1. Run the following command to install the OpenSSH client on your computer:
`sudo apt-get install openssh-client`
2. Type in your superuser password when asked.
3. Hit Enter to complete the installation.

You are now able to SSH into any machine with the server-side application on it, provided that you have the necessary privileges to gain access, as well as the hostname or IP address.

How to Install an OpenSSH Server

In order to accept SSH connections, a machine needs to have the server-side part of the SSH software toolkit.

If you first want to check if OpenSSH server is available on the Ubuntu system of the remote computer that needs to accept SSH connections, you can try to connect to the local host:

1. Open the terminal on the server machine. You can either search for “terminal” or press **CTRL + ALT + T** on your keyboard.
2. Type in **ssh localhost** and hit enter.
3. For the systems **without** the SSH server installed the response will look similar to this:

```
username@host:~$ ssh localhost
ssh: connect to host localhost port 22: Connection refused usern
ame@host:~$
```

If the above is the case, you will need to install the OpenSSH server. Leave the terminal open and:

1. Run the following command to install the SSH server:

```
sudo apt-get install openssh-server ii.
```

2. Type in your **superuser password** when asked.
3. **Enter** and **Y** to allow the installation to continue after the disk space prompt.

The required support files will be installed, and then you can check if the SSH server is running on the machine by typing this command:

```
sudo service ssh status
```

The response in the terminal should look similar to this if the SSH service is now running properly:

```
username@host:~$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor
preset: enab
Active: active (running) since Fr 2018-03-12 10:53:44 CET; 1min
22s ago Process: 1174 ExecReload=/bin/kill -HUP $MAINPID (code=e
xited, status=0/SUCCESS

Main PID: 3165 (sshd)
```

Another way to test if the OpenSSH server is installed properly and will accept connections is to try running the **ssh localhost** command again in your terminal prompt. The response will look similar to this screen when you run the command for the first time:


```
username@host:~$ ssh localhost
```

```
The authenticity of host 'localhost (127.0.0.1)' can't be established.  
ECDSA key fingerprint is SHA256:9jqmhko9Yo1EQAS1QeNy9xKceHFG5F8W6kp7EX9U3Rs.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
```

```
username@host:~$
```

Enter **yes** or **y** to continue.

Congratulations! You have set up your server to accept SSH connection requests from a different computer using an SSH client.

TIP

You can now edit the SSH daemon configuration file, for example, you can change the default port for SSH connections. In the terminal prompt, run this command:

```
sudo nano /etc/ssh/sshd_config
```

The configuration file will open in the editor of your choice. In this case, we used Nano.

If you need to install Nano, run this command:

```
sudo apt-get install nano
```

Please note that you need to restart SSH service every time you make any changes to the `sshd_config` file by running this command:

```
sudo service ssh restart
```

How to Connect via SSH

Now that you have the OpenSSH client and server installed on every machine you need, you can establish a secure remote connection with your servers. To do so:

1. Open the SSH terminal on your machine and run the following command: `ssh your_username@host_ip_address`

If the username on your local machine matches the one on the server you are trying to connect to, you can just type: `ssh host_ip_address` And hit **Enter**.

2. Type in your password and hit **Enter**. Note that you will not get any feedback on the screen while typing. If you are pasting your password, make sure it is stored safely and not in a text file.
3. When you are connecting to a server for the very first time, it will ask you if you want to continue connecting. Just type yes and hit **Enter**. This message appears only this time since the remote server is not identified on your local machine.
4. An ECDSA key fingerprint is now added and you are connected to the remote server.

If the computer you are trying to remotely connect to is on the same network, then it is best to use the private IP address instead of the public IP address. Otherwise, you will have to use the public IP address only. Additionally, make sure that you know the correct TCP port OpenSSH is listening to for connection requests and that the port forwarding settings are correct. The default port is 22 if nobody changed configuration in the `sshd_config` file. You may also just append the port number after the host IP address.

Here is the example of a connection request using the OpenSSH client. We will specify the port number as well:

```
username@machine:~$ ssh phoenixnap@185.52.53.222 -p7654 phoenixnap@185.52.53.222's password:
```

```
The authenticity of host '185.52.53.222 (185.52.53.222)' can't be established. ECDSA key fingerprint is SHA256:9lyrpzo5Yo1EQAS2QeHy9xKceHFH8F8W6kp7EX2O3Ps. Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added '185.52.53.222' (ECDSA) to the list of known hosts.
```

```
username@host:~$
```

You are now able to manage and control a remote machine using your terminal. If you have trouble connecting to a remote server, make sure that:

- The IP address of the remote machine is correct.
- The port SSH daemon is listening to is not blocked by a firewall or forwarded incorrectly.

- Your username and password are correct.
- The SSH software is installed properly.

For Windows – installation

Install openssh client

Download open ssh from <https://github.com/PowerShell/Win32-OpenSSH/releases> for windows.

Extract that folder and copy it to c:\

Now right click on ssh.exe and copy location and click on edit system variables

Now edit path variable copy that path and put semicolon at end.

Now go to ssh folder, press shift and right click and open command prompt.

Now type ssh then it will run.

Install openssh server

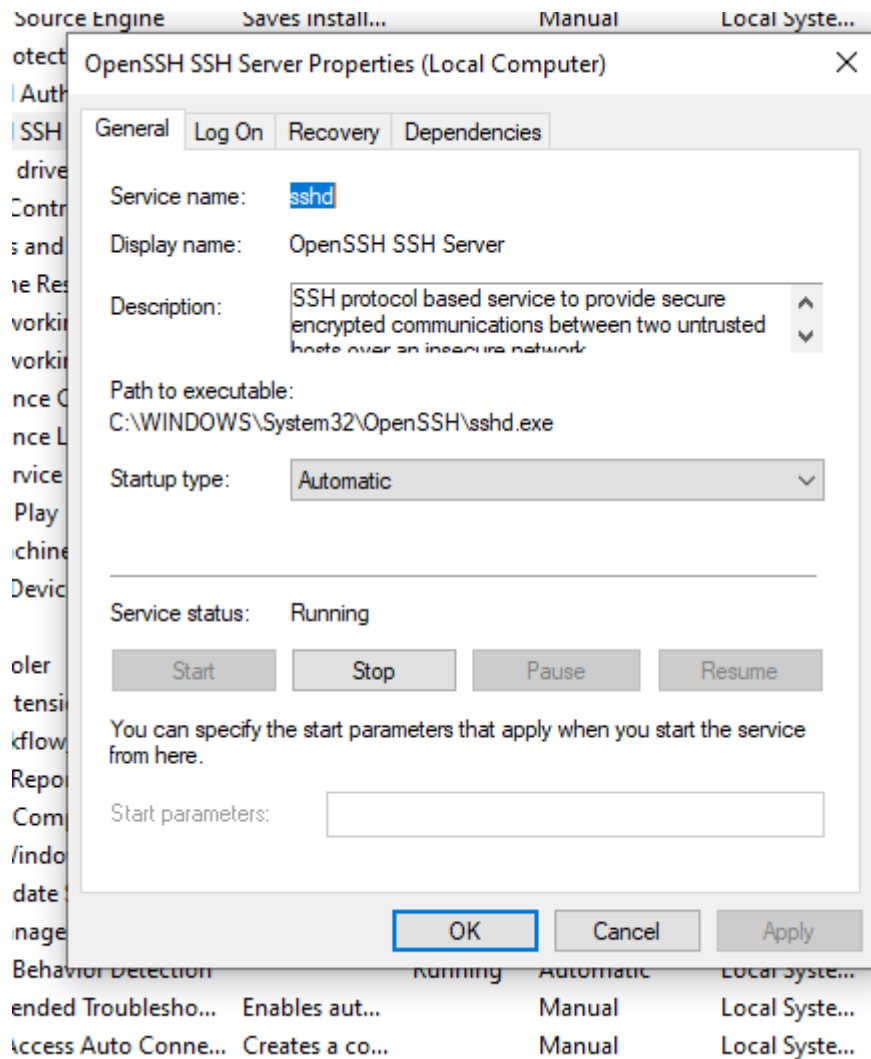
Type optional features in search bar

Then install openssh server.

Type services in run and then select openssh server and right click on it and select properties.

The screenshot shows the Windows Services console. The 'OpenSSH SSH Server' service is highlighted in blue. It is currently running with an automatic startup type and is configured to log on as the local system. The description of the service is: 'SSH protocol based service to provide secure encrypted communications between two untrusted hosts over an insecure network.'

Name	Description	Status	Startup Type	Log On As
Office 64 Source Engine	Saves install...		Manual	Local Syste...
Online Protection System		Running	Automatic	Local Syste...
OpenSSH Authentication A...	Agent to ho...		Disabled	Local Syste...
OpenSSH SSH Server	SSH protoc...	Running	Automatic	Local Syste...
Optimize drives	Helps the c...		Manual	Local Syste...
Parental Controls	Enforces pa...		Manual	Local Syste...
Payments and NFC/SE Man...	Manages pa...	Running	Manual (Trig...	Local Service
Peer Name Resolution Prot...	Enables serv...		Manual	Local Service
Peer Networking Grouping	Enables mul...		Manual	Local Service
Peer Networking Identity M...	Provides ide...		Manual	Local Service
Performance Counter DLL ...	Enables rem...		Manual	Local Service
Performance Counter DLL ...	Enables rem...		Manual	Local Service



Select status type to automatic and click on start then click on apply.

Now open command prompt and run following command. Check also if firewall settings allowed ssh request or not

```
C:\OpenSSH-Win64\OpenSSH-Win64>ssh hp@192.168.242.131
The authenticity of host '192.168.242.131 (192.168.242.131)' can't be established.
ED25519 key fingerprint is SHA256:jZlRPQ3kEjC6hav2xQh1BT5BJyfzaliI1iqcEpDI4IYQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.242.131' (ED25519) to the list of known hosts.
hp@192.168.242.131's password:
```

Introduction

SSH is a network tool used for remote, command-line login to systems that have the server enabled. It has sibling applications named SFTP and SCP that can be used to copy files.

Windows 10 systems with build 1803 or newer and Windows Server 2019 come with an implementation of OpenSSH that's enabled by default. Some older versions may have this as an optional component that needs to be installed before it can be used.

Checking for Installation

The easiest way to determine if OpenSSH is installed is to open a command window (go to the Start Menu, look under "Windows System", and pick "Command Prompt"). Type "ssh" and you should see something like this:

```
usage: ssh [-46AaCfGgKkMNNqsTtVvXxYy] [-B bind_interface]
          [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
          [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
          [-i identity_file] [-J [user@]host[:port]] [-L address]
          [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
          [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] destination [command]
```

If it returns:

'ssh' is not recognized as an internal or external command,
operable program or batch file.

Then it may need to be installed (see if the folder "C:\Windows\System32\OpenSSH" exists and has files such as "ssh.exe" in it). If it is installed, then that directory needs to be added to the search path (type "echo %PATH%" in the command prompt window to see if it's in there). If not, changing the system path requires administrative privileges so contact SENS for assistance.

Installation

Click on the "Settings" gear in the left pane of the Start Menu.

Click on "Apps".

Click on "Optional features".

Look in the list. If you do not see "OpenSSH Client", click "Add a feature".

Select "OpenSSH Client" and click "Install".

Usage

Once it's installed, using it is as simple as typing something such as:

```
> ssh user@linux.sens.buffalo.edu
```

To copy files using Secure FTP:

```
> sftp *.doc user@linux.sens.buffalo.edu
```

To copy files using Secure Copy, preserving original timestamps:

```
> scp -p *.doc user@linux.sens.buffalo.edu
```

Example

```
Command Prompt - ssh savvynik@192.168.1.117
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\savvy>ssh savvynik@192.168.1.117
savvynik@192.168.1.117's password: 
```

```
Last login: Fri Jun 19 00:45:16 2020 from 192.168.1.145
savvynik@savvyserver:~$ ls
savvynik@savvyserver:~$ cd /home/
savvynik@savvyserver:/home$ ls
savvynik
savvynik@savvyserver:/home$ cd savvynik/
savvynik@savvyserver:~$ ls
savvynik@savvyserver:~$ cd ..
savvynik@savvyserver:/home$ ls
savvynik
savvynik@savvyserver:/home$ cd ..
savvynik@savvyserver:/$ ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  swap.img  tmp  var
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr
savvynik@savvyserver:/$ cd /var/l
lib/  local/  lock/  log/
savvynik@savvyserver:/$ cd /var/
backups/  crash/  local/  log/  opt/  snap/  tmp/
cache/  lib/  lock/  mail/  run/  spool/  www/
savvynik@savvyserver:/$ cd /var/www/html/
savvynik@savvyserver:/var/www/html$ ls
index.html
```

Snort – A Network Intrusion Detection System for Ubuntu

Snort is a well-known open-source network intrusion detection and prevention system (IDS). Snort is very useful to monitor the package sent and received through a network interface. You can specify the network interface to monitor the traffic flow. Snort works on the basis of signature-based detection. Snort uses different types of rulesets to detect network intrusions such as community, Registered and subscription rules. Correctly installed and configured Snort can be very useful in detecting different kinds of attacks and threats like SMB probes, malware infections, compromised systems, etc. In this article, we will learn how to install and configure Snort on an Ubuntu 20.04 system.

Snort Rules

Snort uses rulesets to detect network intrusions which are as follows. There are three types of rule sets available:

Community rules

These are the rules created by the snort user community and available free of cost.

Registered rules

These are the rules provided by Talos and are only available for registered users. Registration takes only a moment and free of cost. After the registration, you will get a code that is needed to submit while sending the download request

Subscription rules

These rules are also the same as registered rules but are provided to registered users before the release. These rulesets are paid and costing is based on personal user or business user.

Snort Installation

Installation of snort in the Linux system would be a manual and lengthy process. Nowadays the installation is very simple and easier since most of the Linux distributions have made the Snort package available in the repositories. The package can be installed from the source as well as from the software repositories.

During the installation, you will be asked to provide some details regarding the network interface. Run the following command, and note the details for future use.

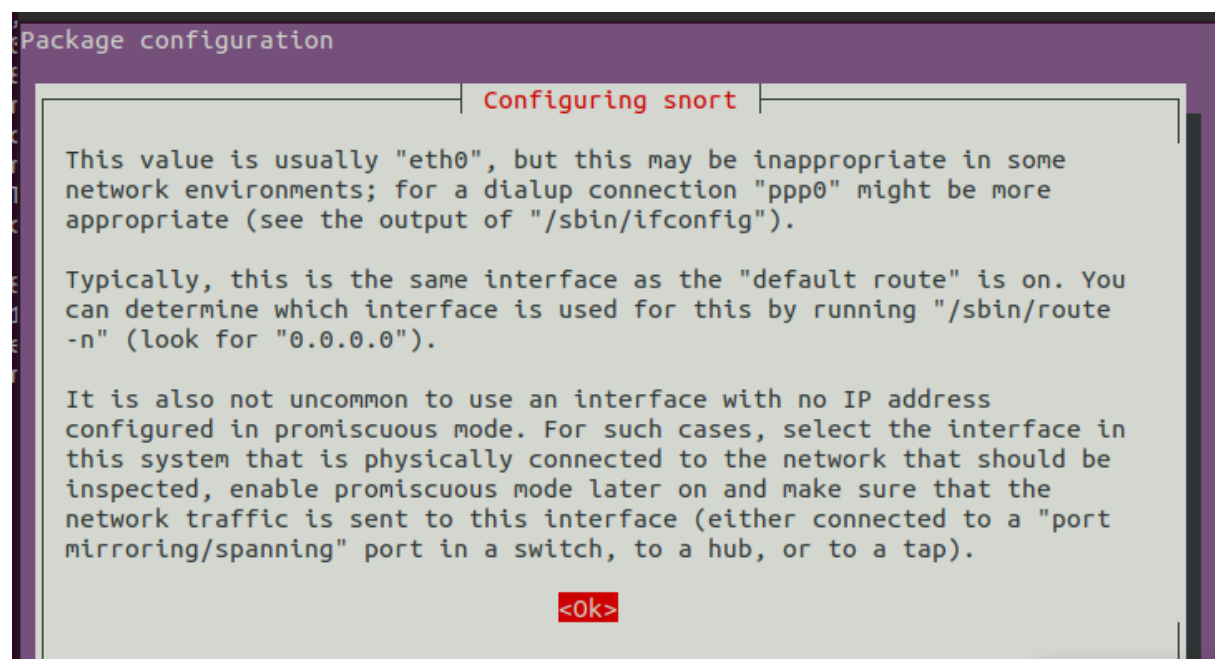
```
$ ip a
```

```
aayush@Linuxways:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:fa:af:9b brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.218.128/24 brd 192.168.218.255 scope global dynamic noprefixroute ens33
        valid_lft 1313sec preferred_lft 1313sec
    inet6 fe80::552e:4082:91e7:436e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
aayush@Linuxways:~$
```

To install the Snort tool in Ubuntu, use the following command.

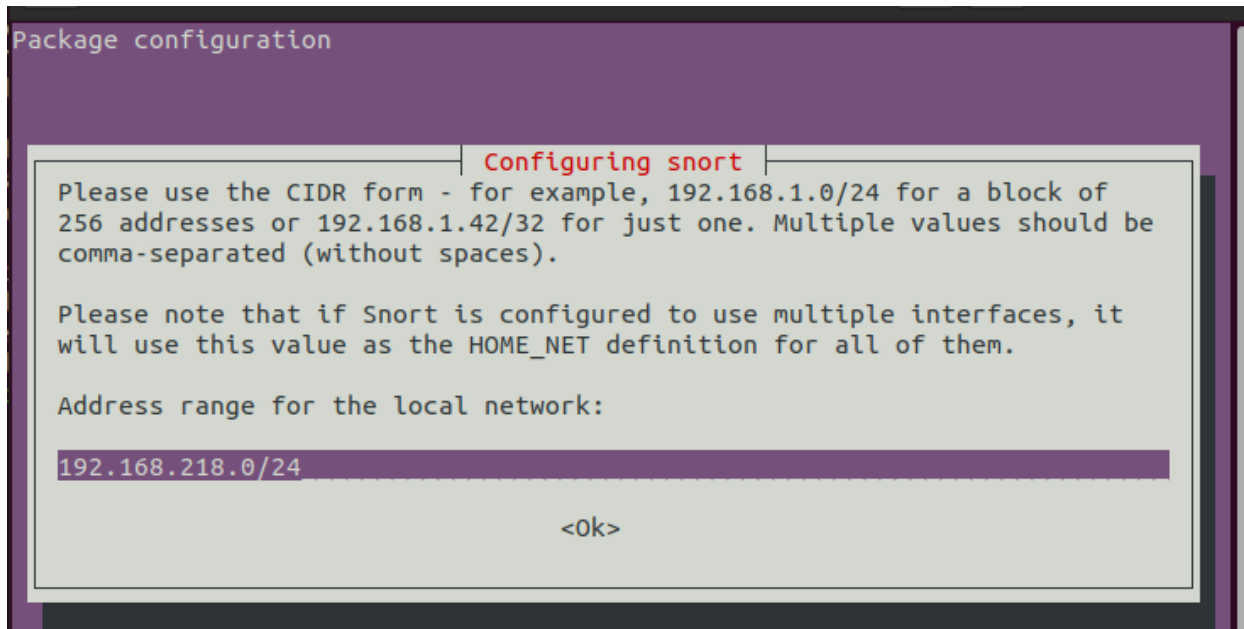
```
$ sudo apt install snort
```

In the above example, **ens33** is the name of the network interface and **192.168.218.128** is the ip address. The **/24** shows that the network is of subnet mask 255.255.255.0 . Take note of these things since we need to provide these details during the installation. Now, press tab to navigate to the ok option and press enter.



Now provide the name of the network interface, navigate to the ok option using the tab key, and press enter.

Provide the network address with the subnet mask. Navigate to the ok option using the tab key and press enter.



Once the installation is completed, run the command below to verify.

```
$ snort --version
```

```
aayush@Linuxways:~$ snort --version

,,_      -*> 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 1.9.1 (with TPACKET_V3)
         Using PCRE version: 8.39 2016-06-14
         Using ZLIB version: 1.2.11
```

Configuring snort

Before using the Snort, there are some things to be made in the configuration file. Snort stores the configuration files under the directory `/etc/snort/` as the file name `snort.conf`.

Edit the configuration file with any text editor and make the following changes.

```
$ sudo vi /etc/snort/snort.conf
```

Find the line **ipvar HOME_NET any** in the configuration file and replace any with your network address.

```
# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overridden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 192.168.218.0/24
# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
```

In the above example, a network address **192.168.218.0** with subnet mask **prefix 24** is used. Replace it with your network address and provide the prefix.

Save the file and exit

Download and Update Snort rules

Snort uses rulesets for intrusion detection. There are three types of rulesets that we have previously described at the start of the article. In this article, we will download and update community rules.

To install and update the rules, create a directory for the rules.

```
$ mkdir /usr/local/etc/rules
```

Download the community rules using the following command.

```
$ wget https://www.snort.org/downloads/community/snort3-community-
rules.tar.gz
```

Or else, you can browse the link below and download the rules.

<https://www.snort.org/downloads/#snort-3.0>

Extract the downloaded files in the previously created directory.

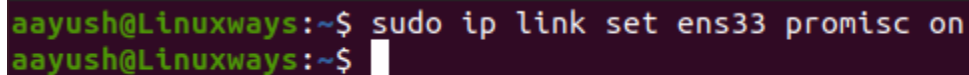
```
$ tar xzf snort3-community-rules.tar.gz -C /usr/local/etc/rules/
```

Enable Promiscuous Mode

We need to make Snort computer's network interface listen to all traffic. To make this happen, enable promiscuous mode . Run the following command with the interface name.

```
$ sudo ip link set ens33 promisc on
```

Where ens33 is the interface name



```
aayush@Linuxways:~$ sudo ip link set ens33 promisc on
aayush@Linuxways:~$
```

Running snort

Now we are good to start the Snort. Follow the syntax below and substitute the parameters accordingly.

```
$ sudo snort -d -l /var/log/snort/ -h 192.168.218.0/24 -A console -c /etc/snort/snort.conf
```

Where,

-d is used to filter application layer packets

-l is used to setup the logging directory

-h is used to specify the home network

-A is used to send the alert to the console windows

-c is used to specify the snort configuration

Once the Snort is started, you will get the following output in the terminal.

```
aayush@Linuxways:~$ sudo snort -d -l /var/log/snort/ -h 192.168.218.0/24 -A console -c /etc/snort/snort.conf
Running in IDS mode

--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848
5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 830
0 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3
702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181
8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined : [ 2123 2152 3386 ]
Detection:
  Search-Method = AC-Full-Q
  Split Any/Any group = enabled
  Search-Method Optimization = Disabled
```

You can check the log files to get information about intrusion detection.

Snort works on the basis of rulesets. So, always keep the rulesets up to date. You can set up a cronjob to download the rules and update them periodically

Snort - windows

Introduction To Snort:

In this tutorial we will look at installing and configuration of snort on Windows 10. Snort is an open source and popular Intrusion Detection System (IDS). It works by actively monitoring of network traffic parsing each packet and alerting system administrator of any anomalous behavior that goes against the snort rules configured by the administrator according to the security policies of an organization.

Installing Snort 2.9.17 on Windows 10 A Step By Step Guide:

1. For Windows 10 64 bit supported SNORT's executable file can be downloaded from **here**.
2. Open the downloaded snort executable file.
3. Click On 'I Agree' on the license agreement.

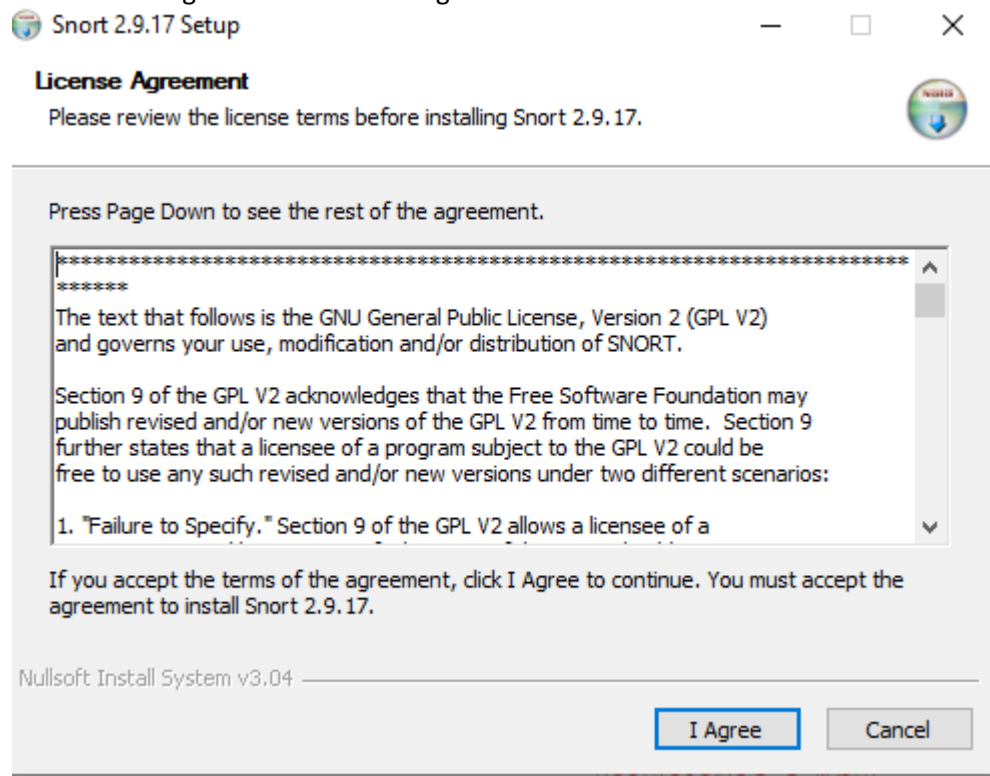


Figure 01: License agreement for Snort 2.9.17

4. Choose components of Snort to be installed.

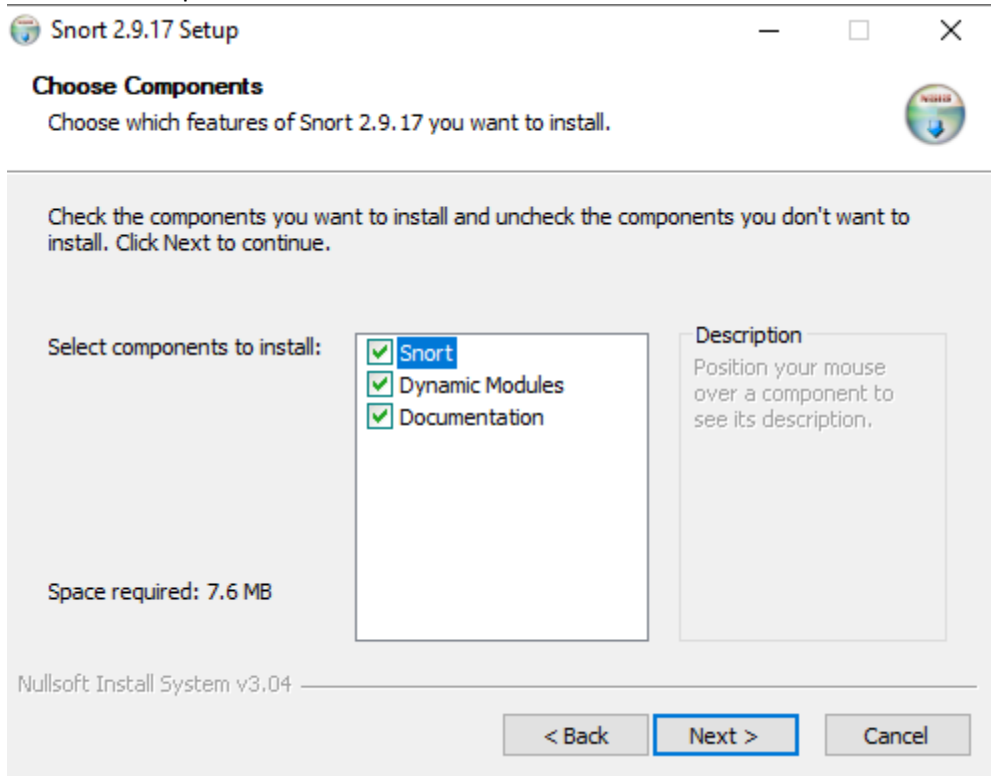


Figure 02: Choosing Components for Snort 2.9.17

5. Click "Next" and then choose install location for snort preferably a separate folder in Windows C Drive.

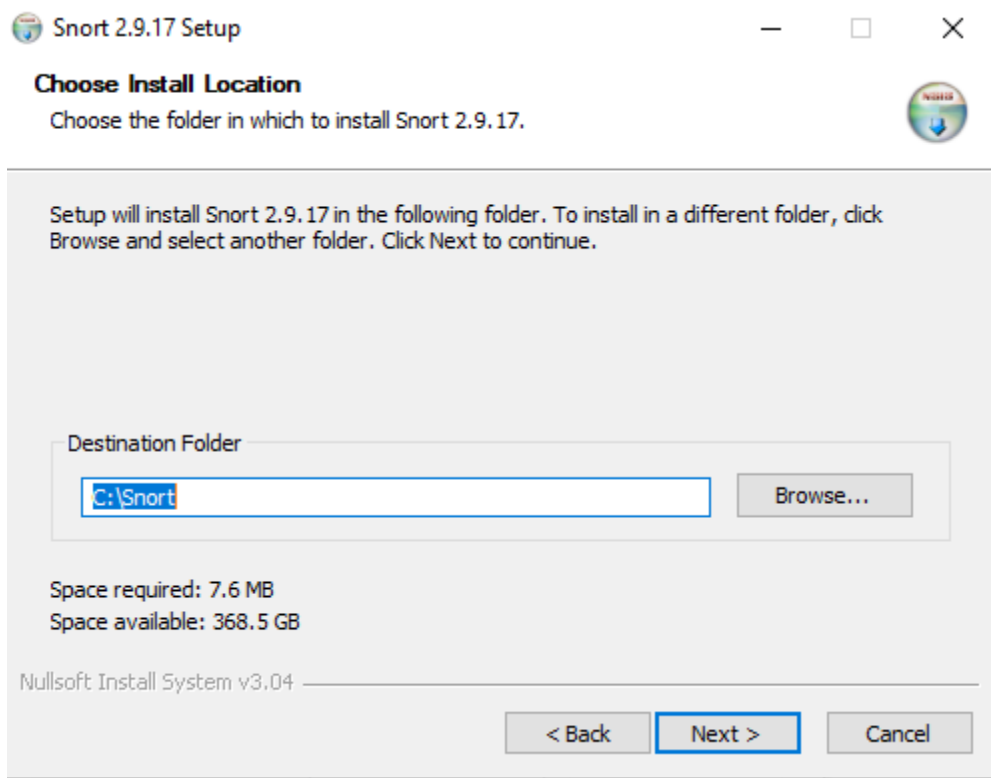


Figure 03: Choose Install location for Snort 2.9.17

6. Click “Next” Installation process starts and then it completes as shown in figure 04:

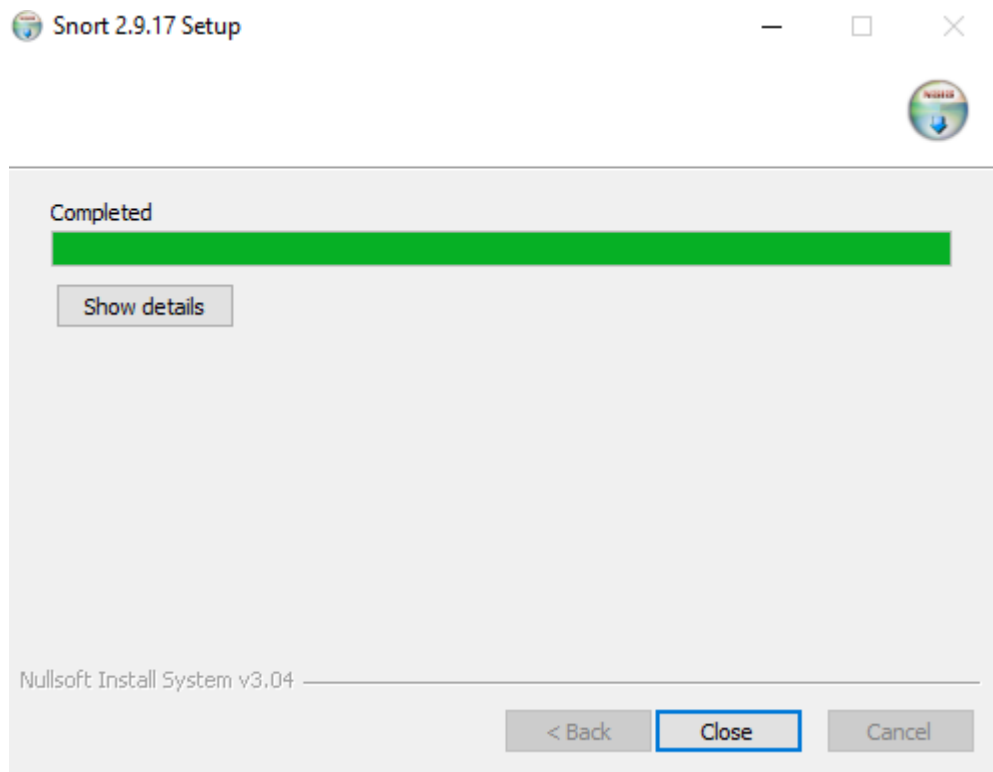


Figure 04: Setup Complete for Snort 2.9.17

7. When you click "Close" you are prompted with this dialogue box:

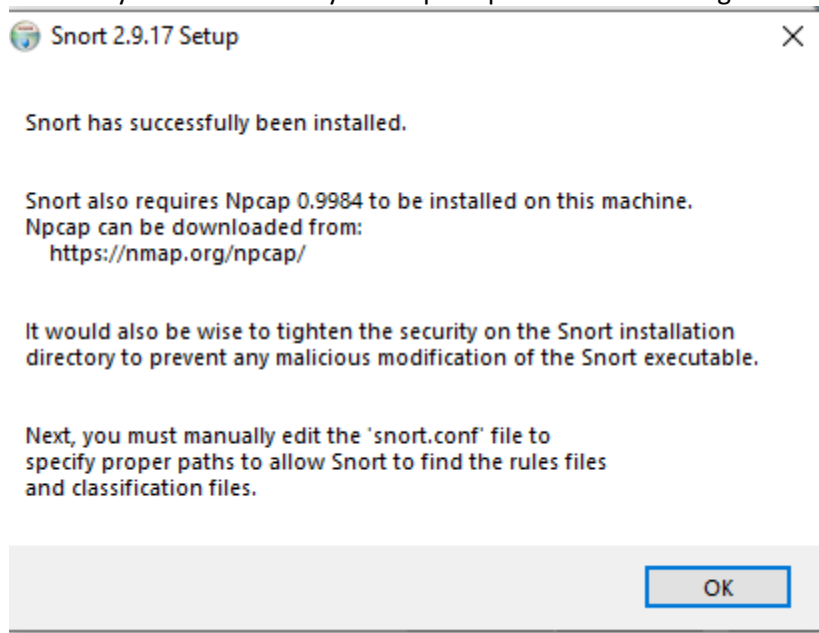


Figure 05: Window showing details of software needed to run Snort successfully

8. Installing Npcap or winpcap is required by snort for proper functioning.

Winpcap - <https://www.winpcap.org/install/>

9. Npcap for Windows 10 can be downloaded from **here**.

10. Opening Npcap setup file, Click on 'I Agree' To license agreement.

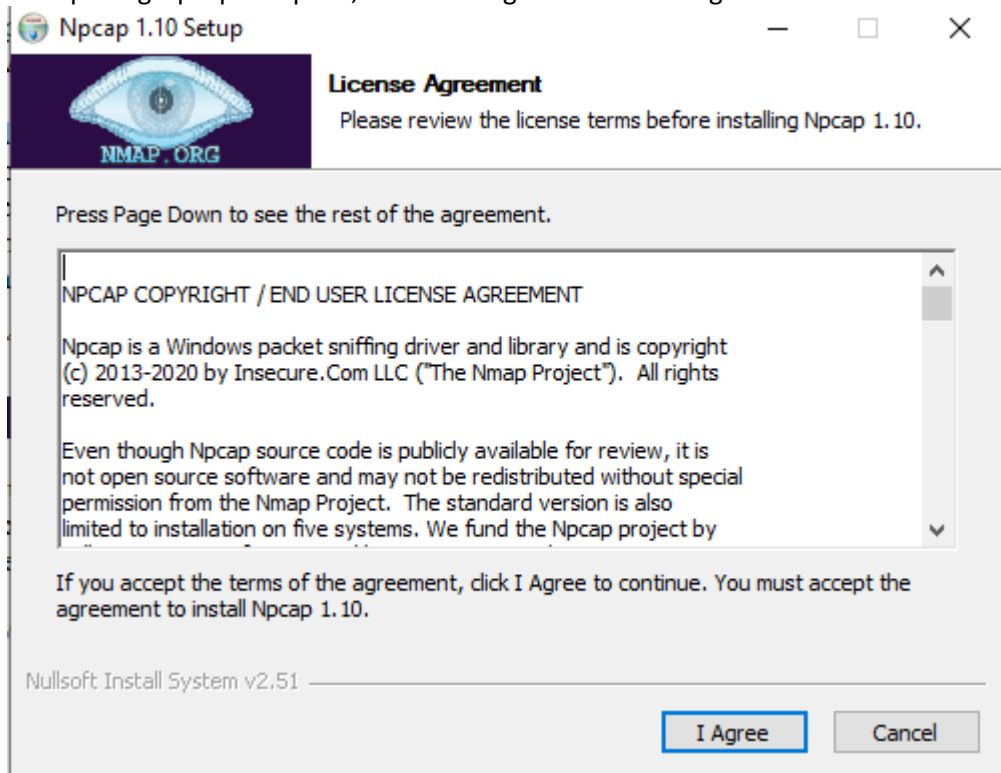


Figure 06: License agreement for Npcap 1.10

11. Now we proceed to choose which components of Npcap are to be installed and then clicking on "Install".

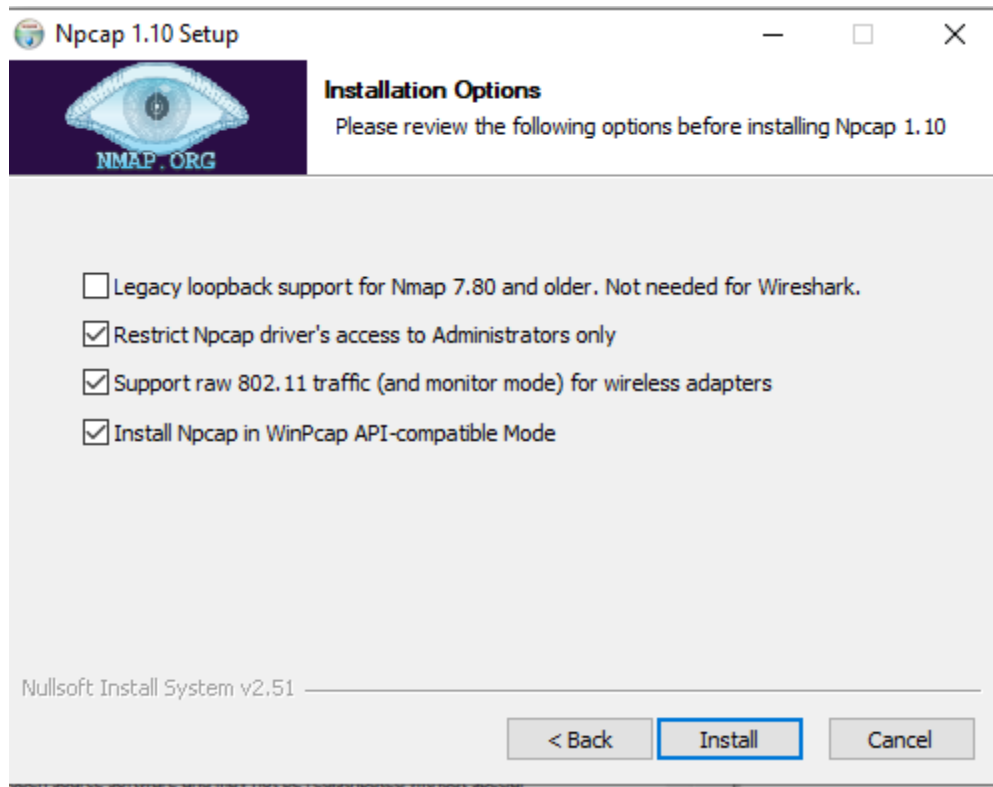


Figure 07: Choose Components to install for Npcap 1.10

12. Installation process starts and completes. Clicking on “Next” we have:

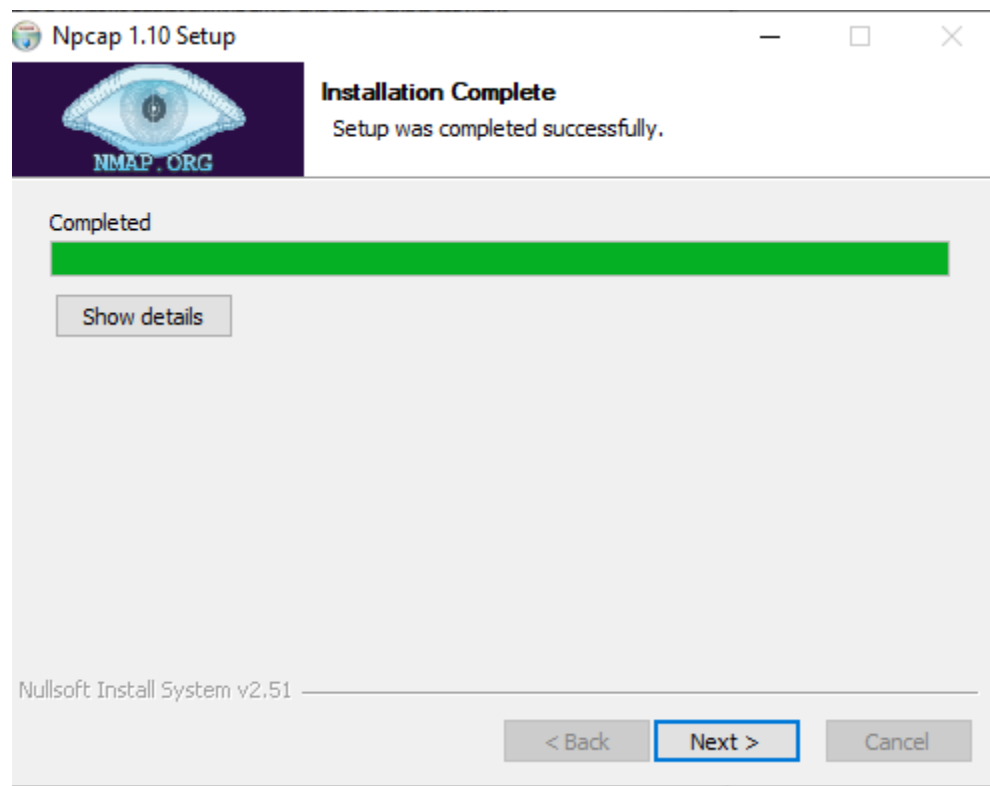


Figure 08: Setup completed for Npcap 1.10

13. Now the window for installation of Npcap shows it has been installed. Clicking “Finish”.

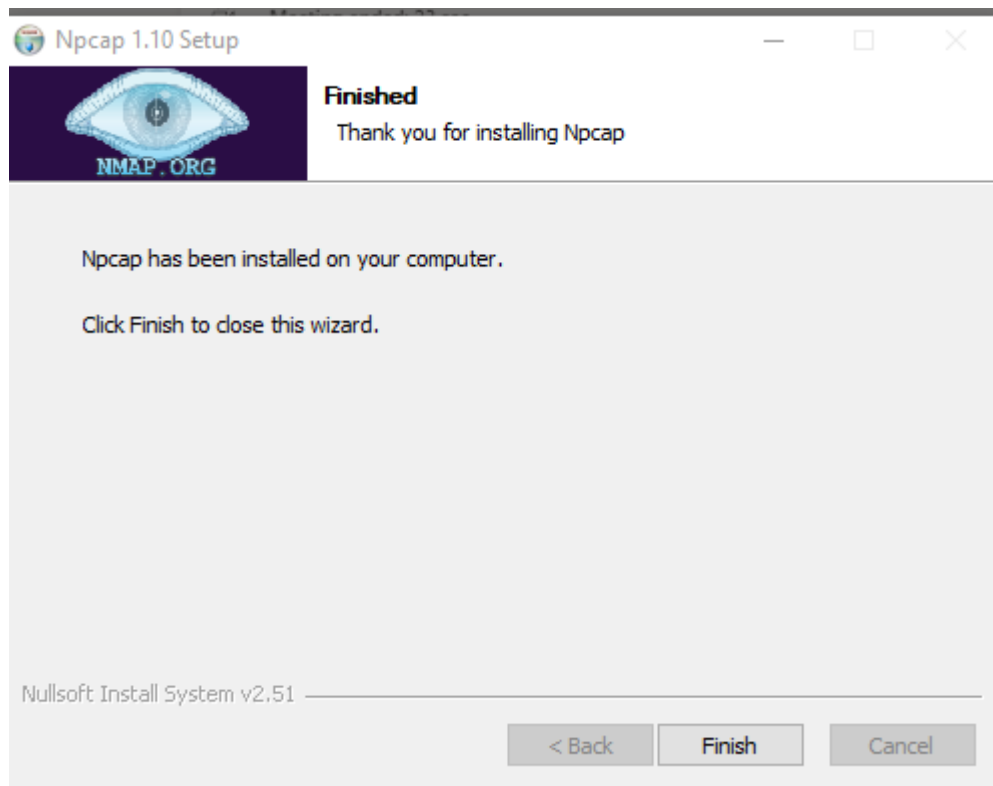


Figure 09: Successful installation for Npcap 1.10 completed

14. After installing Snort and Npcap enter these commands in windows 10 Command prompt to check snorts working

```
C:\Users\Zaeem786>cd--  
'cd--' is not recognized as an internal or external command,  
operable program or batch file.  
  
C:\Users\Zaeem786>cd..  
  
C:\Users>cd..  
  
C:\>cd snort  
  
C:\Snort>cd bin  
  
C:\Snort\bin>snort -V  
  
  _ _ _ _ _  
  o" )~  
  ' ' ' ' '  -*> Snort! <*-  
Version 2.9.17-WIN64 GRE (Build 199)  
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014-2020 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using PCRE version: 8.10 2010-06-25  
Using ZLIB version: 1.2.11  
  
C:\Snort\bin>
```

Figure 10: Successfully running Snort on Windows 10 through command prompt

15. As you can see in the above figure that snort runs successfully.

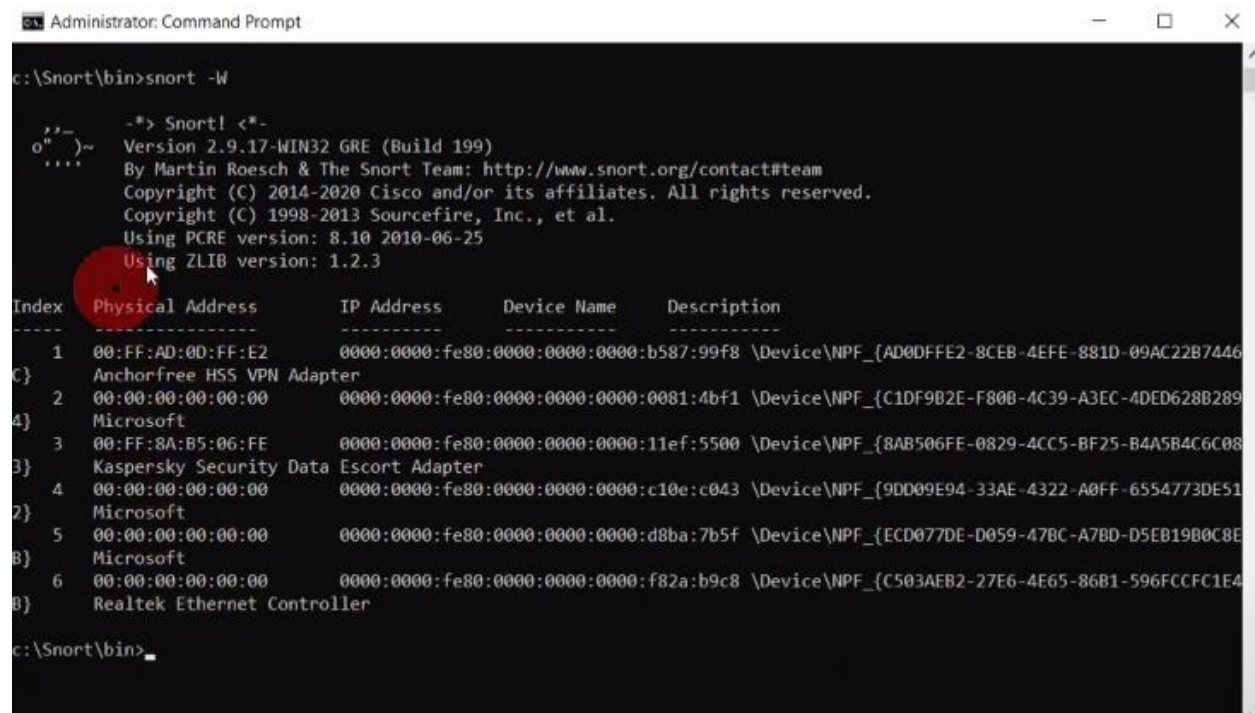
This is how you can download and install Snort along with its dependency i.e. Npcap.

Now we can run command like `c:\snort\bin>snort.exe`

After installation download rules from <https://www.snort.org/downloads>

Extract folders and copy rules folder into snort folder.

Then open snort/etc folder and open snort.conf file. Change Ip address and other settings into conf file.



```
Administrator: Command Prompt
c:\Snort\bin>snort -W

-*> Snort! <*-
o" )~ Version 2.9.17-WIN32 GRE (Build 199)
**** By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
      Copyright (C) 2014-2020 Cisco and/or its affiliates. All rights reserved.
      Copyright (C) 1998-2013 Sourcefire, Inc., et al.
      Using PCRE version: 8.10 2010-06-25
      Using ZLIB version: 1.2.3

Index  Physical Address      IP Address      Device Name      Description
-----
1      00:FF:AD:0D:FF:E2      0000:0000:fe80:0000:0000:0000:b587:99f8 \Device\NPF_{AD0DFFE2-8CEB-4EFE-881D-09AC22B7446}
C} Anchorfree HSS VPN Adapter
2      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:0081:4bf1 \Device\NPF_{C1DF9B2E-F800-4C39-A3EC-4DED628B289}
4} Microsoft
3      00:FF:8A:B5:06:FE      0000:0000:fe80:0000:0000:0000:11ef:5500 \Device\NPF_{8AB506FE-0829-4CC5-BF25-B4A5B4C6C08}
3} Kaspersky Security Data Escort Adapter
4      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:c10e:c043 \Device\NPF_{9DD09E94-33AE-4322-A0FF-6554773DE51}
2} Microsoft
5      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:d8ba:7b5f \Device\NPF_{ECD077DE-D059-47BC-A7BD-D5EB19B0C8E}
8} Microsoft
6      00:00:00:00:00:00      0000:0000:fe80:0000:0000:0000:f82a:b9c8 \Device\NPF_{C503AEB2-27E6-4E65-86B1-596FCCFC1E4}
8} Realtek Ethernet Controller

c:\Snort\bin>
```

Configuration commands:

```
snort -i -1 -c c:\Snort\etc\snort.conf -T
snort -i 2 -c c:\snort\etc\snort.conf -A console
```

Windump – windows /tcpdump – linux

Step 1 – Download and install Windump

<http://www.winpcap.org/windump/>



You will need to place your network card into promiscuous mode – for this, install WinPcap.

Step 2 – Download and install WinPcap

<http://www.winpcap.org/install/default.htm>



Step 3 – Open a Command Prompt with Administrator Rights

Start > Accessories > Command Prompt

Right Click > Run As Administrator

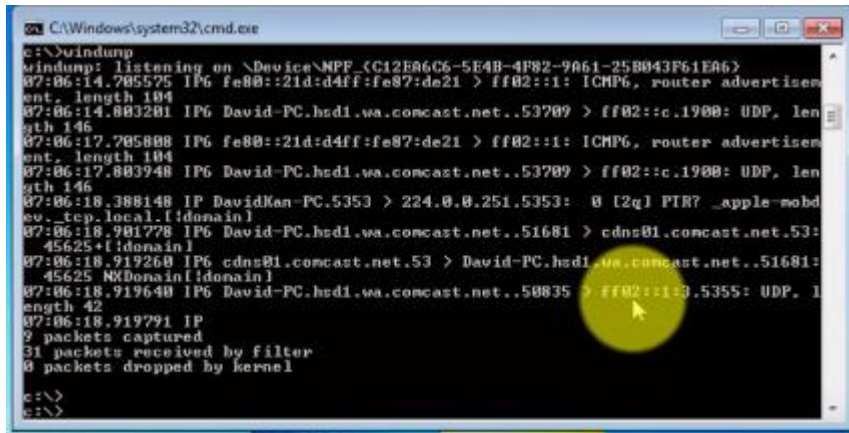
Change the directory to your download directory – normally in windows this is:

```
cd c:\Users\Smile\Downloads
```

Smile will be replaced with your username eg `cd c:\Users\your username\Downloads`

Copy windump.exe into for example c:\ then run command from that location.

Type following commands:



```
c:\>windump
windump: listening on \Device\NPF{C12E06C6-5E4B-4F82-9A61-25B043F61E06}
07:06:14.705575 IP6 fe80::21d:d4ff:fe87:de21 > ff02::1: ICMP6, router advertisement, length 104
07:06:14.803201 IP6 David-PC.hsd1.wa.comcast.net..53789 > ff02::c:1900: UDP, length 146
07:06:17.705808 IP6 fe80::21d:d4ff:fe87:de21 > ff02::1: ICMP6, router advertisement, length 104
07:06:17.803948 IP6 David-PC.hsd1.wa.comcast.net..53789 > ff02::c:1900: UDP, length 146
07:06:18.388148 IP David-Kan-PC.5353 > 224.0.0.251.5353: 0 [2q] PIR? _apple-mobd
eu._tcp.local.(domain)
07:06:18.701778 IP6 David-PC.hsd1.wa.comcast.net..51681 > cdns81.comcast.net.53:
45625*(domain)
07:06:18.919268 IP6 cdns81.comcast.net.53 > David-PC.hsd1.wa.comcast.net..51681:
45625*(domain)
07:06:18.919640 IP6 David-PC.hsd1.wa.comcast.net..50835 > ff02::1:3.5355: UDP, length 42
07:06:18.919791 IP
9 packets captured
31 packets received by filter
0 packets dropped by kernel
c:\>
c:\>
```

Its start capturing packets, press ctrl+c to stop capture.

Windump -d , list all the available interfaces on system.

Windump -l 1 , to listen to particular interface.

Windump -l 1 -w c:\test\mycap.pcap, it will write every packet captured into mycap.pcap file we can also analysis this file into wireshark.