## Installation of Wireshark

Wireshark will run on a variety of operating systems and is not difficult to get up and running. We will touch on Ubuntu Linux, Centos and Windows.

#### Install on Ubuntu or Debian

#apt-get update

#apt-get install wireshark tshark

#### Install on Fedora or CentOS

#yum install wireshark-gnome

#### Install on Windows

Head over to the [Wireshark Download](https://www.wireshark.org/download.html) page, grab the installation executable and run it to install. Pretty straight forward, you will also be installing a packet capture driver. This allows the network card to enter promiscuous mode.

## Getting Started with Filters

After running an initial capture you will see the standard layout and the packet details that can be viewed through the interface.

Once you have captured a HTTP session, stop the capture and try playing with some basic filters and the Analyze | Follow | HTTP Stream options.

The filters are easy to read and self explanatory. You simply enter these expressions into the filter bar (or on the command line if using tshark). A primary benefit of the filters is to remove the noise (traffic you don't want to see). As can be seen here you can filter on MAC address, IP address, Subnet or protocol. The easiest filter is to simply type http into the filter bar, only HTTP (tcp port 80) traffic will now be shown.

#### IP Address Filter Examples

ip.addr == 192.168.0.5

!(ip.addr == 192.168.0.0/24)

#### Protocol Filter Examples

tcp

udp

tcp.port == 80 || udp.port == 80

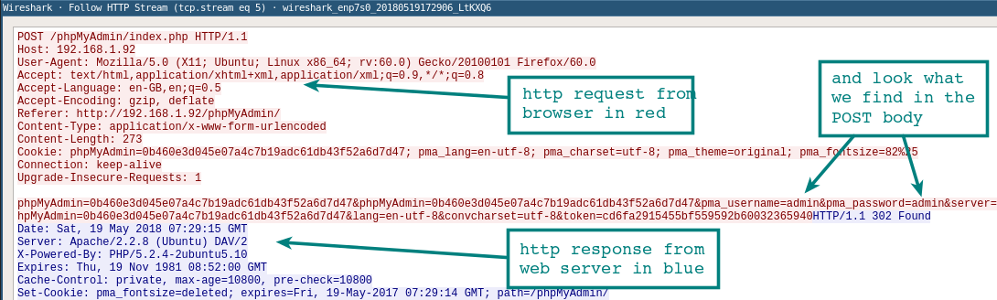
http

not arp and not (udp.port == 53)

Try generating a filter combination that shows all non HTTP and HTTPS traffic leaving your local system that is not destined for the local network. This is a good way to find software (malware even) that is communicating with the Internet using unusual protocols.

## Follow the Stream

Once you have a number of packets showing HTTP you can select one and then Analyze | Follow | HTTP Stream from the drop down menu. This will show you an assembled HTTP session. In this new window you can see the HTTP request from the browser and HTTP response from the web server. Goal! You are now winning at Wireshark. Continue reading our Wireshark Tutorial for more advanced tips.



## Resolve DNS in Wireshark

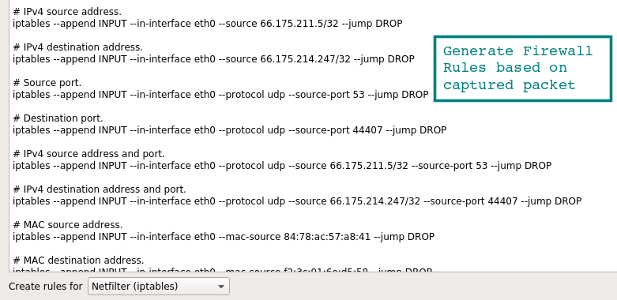
By default Wireshark won't resolve the network address that it is displaying in the console. Only showing IP addresses, by changing an option in the preferences you can enable the resolution of IP addresses to network names. This will just as it does when using tcpdump slow down the display of packets as the resolution has to take place. It is also important to understand that if you are doing a live capture the DNS requests from your Wireshark host will be additional traffic that you then might be capturing.

## Tshark for the Command Line

If you haven't had a play with tshark, take a look at our [tshark tutorial and filter examples](https://hackertarget.com/tshark-tutorial-and-filter-examples/). This program is often overlooked but is a great way to capture application layer sessions on a remote system. The advantage over tcpdump is the fact that you can capture and view application layer sessions on the fly, as the protocol decoders included in Wireshark are also available to tshark.

## Build Firewall Rules

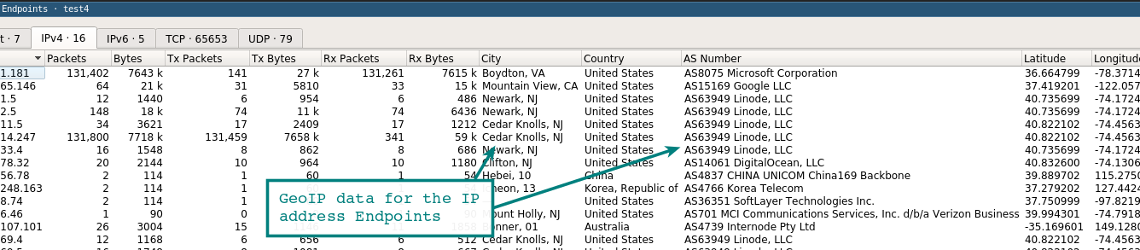
A quick way to generate command line firewall rules, this can save a few minutes Googling for different firewall syntax. Select a rule, and then head up to the Tools | Firewall ACL Rules. Different firewall products such as Cisco IOS (standard and extended), ipfilter, ipfw, iptables, pf and even Windows firewall using netsh.



## Wireshark GeoIP Mapping

As long as Wireshark has been compiled with GeoIP support and you have the Free Maxmind databases available you are able to resolve IP addresses to locations. Take a look at About | Wireshark to see what has been compiled with the version you are using. If you see GeoIP listed, make sure you have the GeoLite City, Country and ASNum databases in a directory on your system running Wireshark. Point to the location of the databases in Edit | Preferences | Name Resolution.

Test it by loading a capture and selecting Statsitcs | Endpoints | IPv4. The columns on the right should show the location and ASN information for the IP address.



Another function of the GeoIP feature is to filter traffic based on location using the ip.geoip display filter.

For example to exclude traffic from an ASN you could use this filter. ASN 63949 is the Linode block so the filter now displays only IP traffic not coming from this netblock.

ip and not ip.geoip.asnum == 63949

Of course you can apply the [same filter to city and country based queries](https://www.wireshark.org/docs/dfref/i/ip.html). Removing noise from your capture display and allowing you to focus in on the packets you care about.