

Wireshark & Tshark

How to do network Analysis with Wireshark and Tshark

- [1.1 Importing Profiles](#)
- [1.2 Customizing Profiles](#)
- [1.3 Adding Filters / Columns](#)
- [1.4 Using Tshark for a Deeper Dive](#)

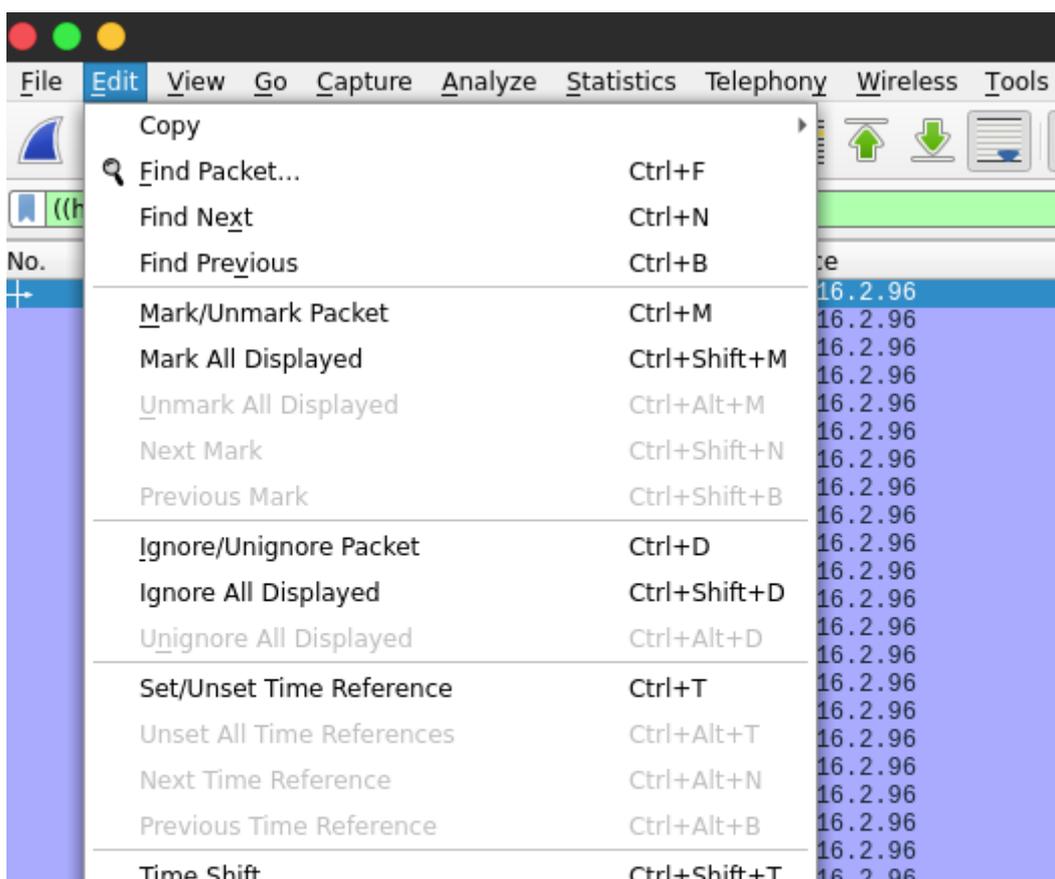
1.1 Importing Profiles

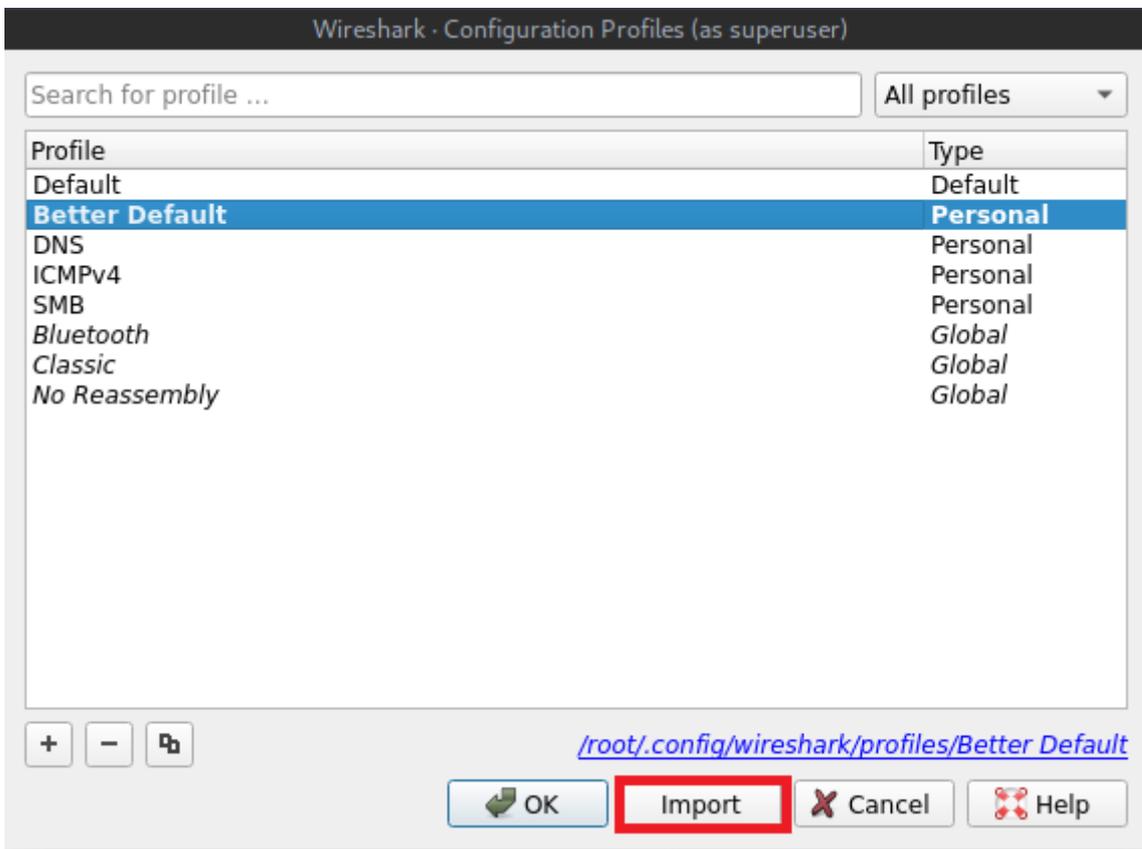
Importing a profile: There are many ways you can go about this, but for this guide, I will provide a link where I downloaded a profile called better default. You can change the layout once you import this profile.

Link to Better Default profile - <https://www.cellstream.com/resources/wireshark-profiles-repository/281-a-better-default-profile-for-wireshark/file>

Save the profile in any location you like, but record this location for reference so when you go to import, you know where it is

Step 1. Go to **Edit** -> **Configuration Profiles** or **Ctrl+Shift+A (Linux)**.
See screenshots





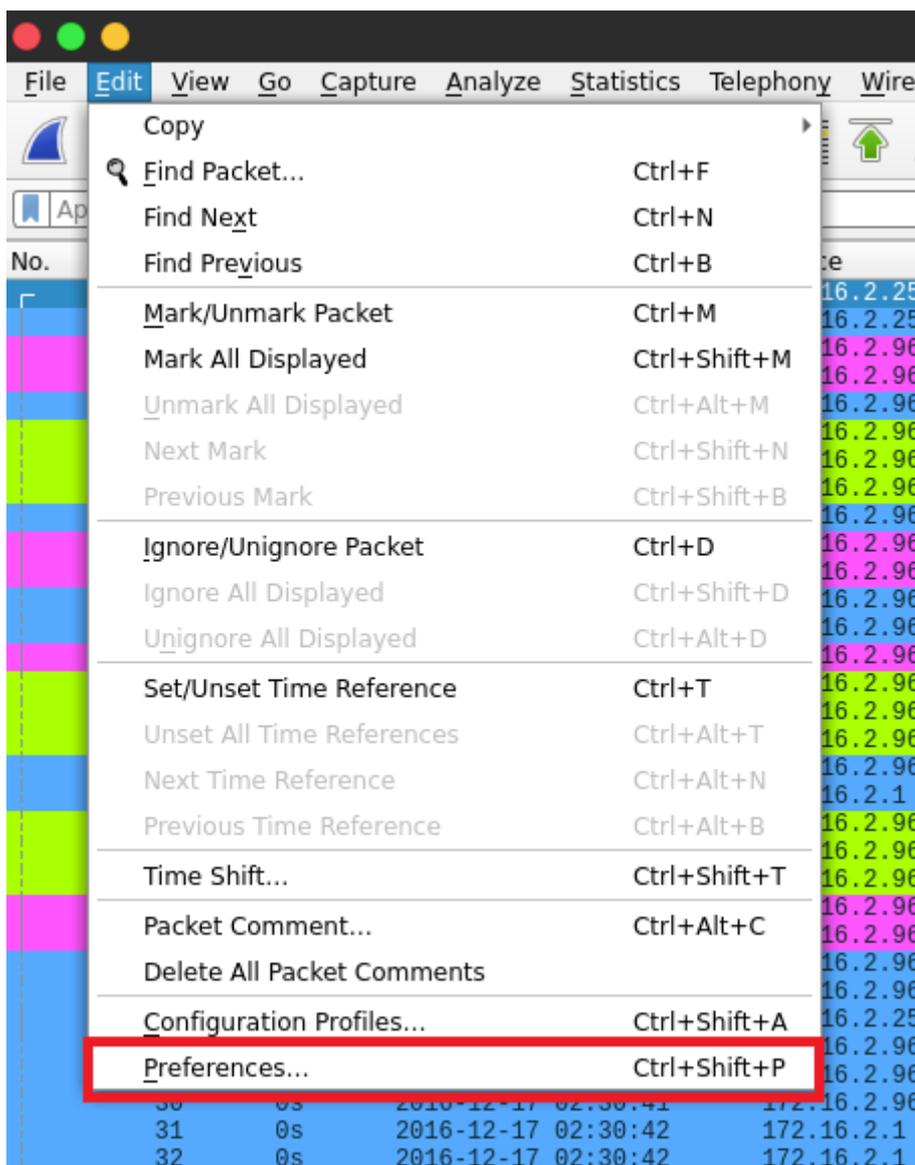
Ensure that the profile you just imported is selected and hit ok. You should see a noticeable change if you have a PCAP loaded. There are several other profiles you can download as well from the same site I provided early in this documentation. You will see some in the list such as the **DNS, ICMPv4, SMB**. I am using the Better Default profile from Section 1.a.i from the link in this document.

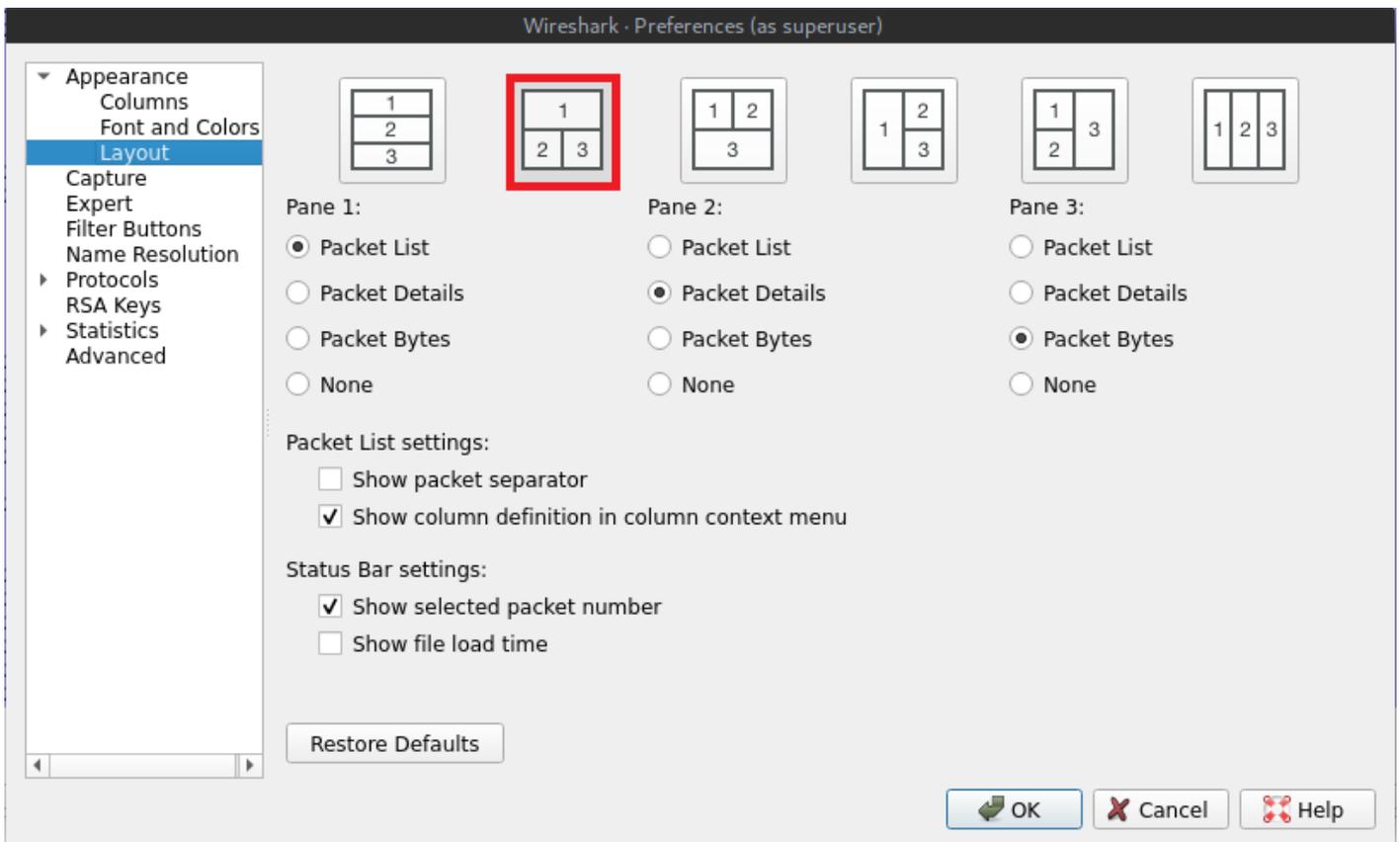
1.2 Customizing Profiles

This profile is going to be the default that is used, however, keep in mind, this will be dictated by your environment and the type of hunt you are doing. You can tweak them all to your liking, however, this will server as a baseline.

This is really more of a personal preference thing, but you can customize this profile and add / remove to your liking.

Step 1. Go to **Edit** -> **Preferences** -> **Layout** or **Ctrl+Shift+P** (Linux).





This will set up your profile to where there is the main pane on top is the packet list, the bottom left are the details of the packet, and the bottom right would be the pack bytes (It shows you the hex and ASCII values)

1.3 Adding Filters / Columns

Creating Search Filters / Columns: This is very useful and will save you time in the long run. What we want to accomplish here is creating your most common filters of data you think you will most likely be looking to dissect out of the pcap.

Where to Start

A good start might be to look at user agents, request URIs, HTTP request methods or Kerberos CNameString if in an enterprise environment where Kerberos is utilized pretty heavily. The situation and end state matter, but these are some examples

Search Filter examples

1. user_agent
2. request.method
3. request.uri
4. request.full_uri
5. CNameString

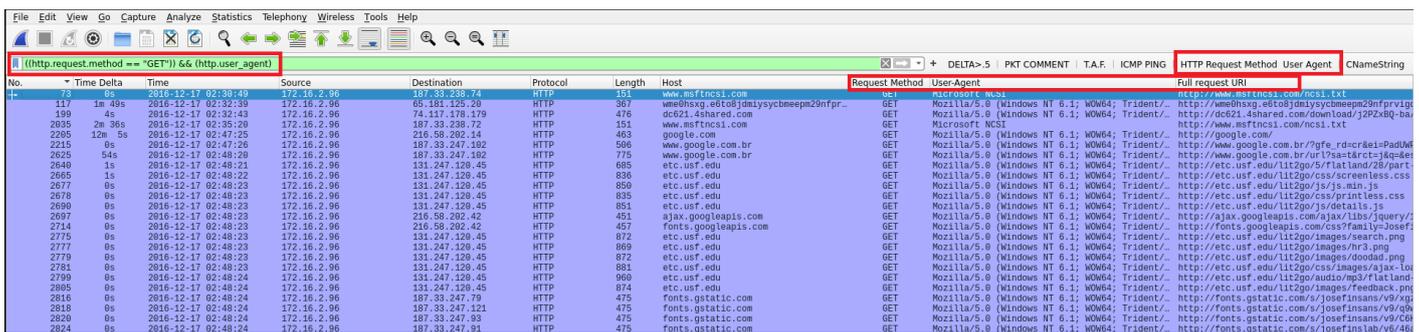
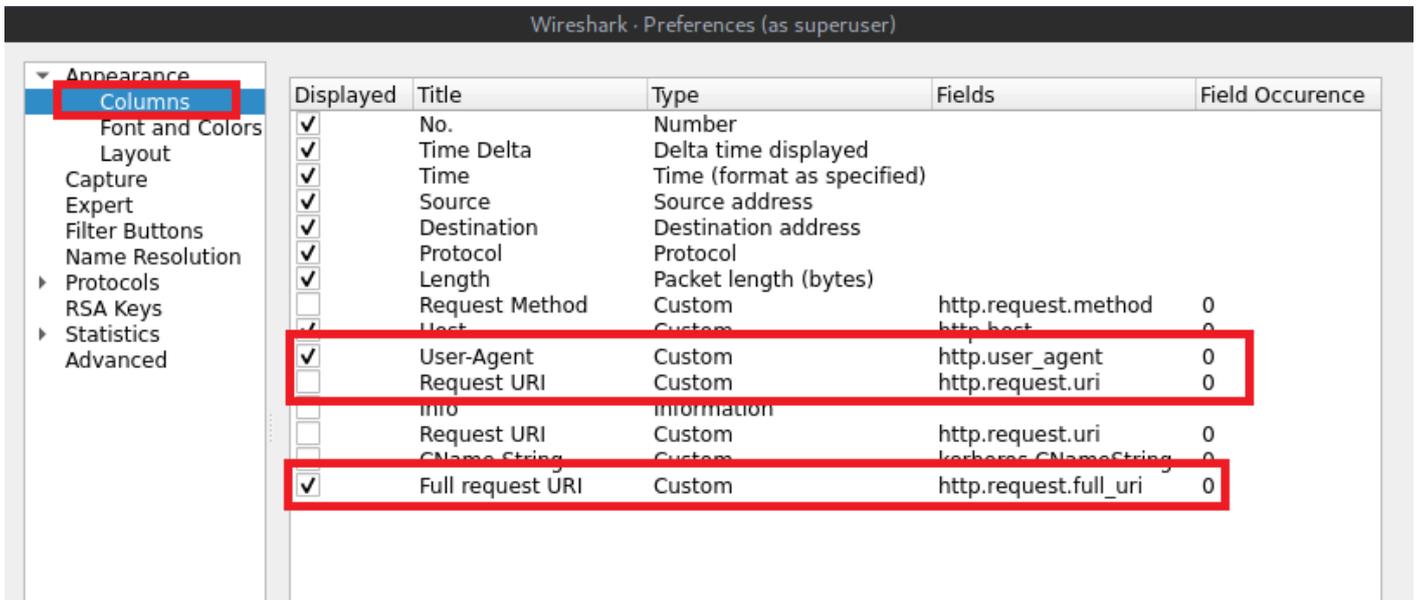
Note** If any data matches this filter, you will see the packet list on the top pane change to show only that data you requested to see. This is only part of the battle. Next, you too add the column to see this data a lot easier, then you can drill down into it. The goal is to find things

faster and more efficient, automate where you can.

Step 1. We are going to take what we did with add searching filters and create column's that will show the data we filtered on. This helps make analyzing the data easier and more efficient

Go to **Edit -> Preferences -> Columns** or **Ctrl+Shift+P (Linux)**.

See screenshots Below



I wish I could zoom in, but if you notice, I have custom filters added which we discussed early. Any search you do, you can save that as a named button like I have. I combined search criteria as you can see in the above. I used the && operator signifying that I wanted to see http request method and the user agent. I then created the columns alike and clicked my search filter button, and just like that, the data I am looking for is front and center.

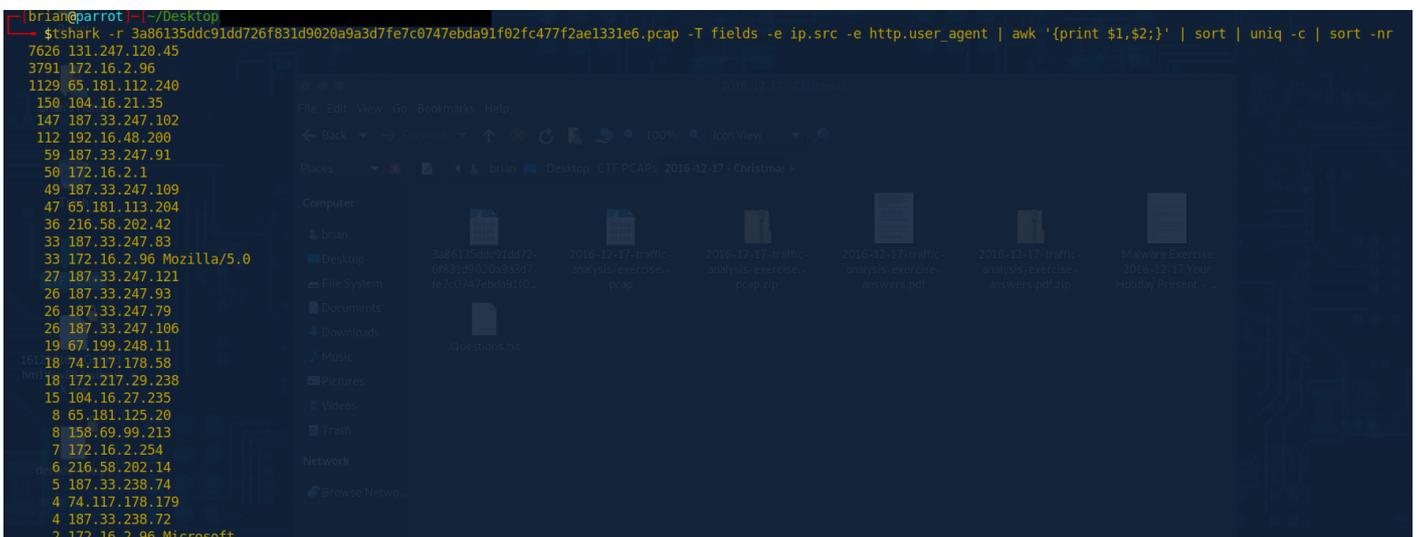
1.4 Using Tshark for a Deeper Dive

A Closer Look: This is not going to be a deep dive into T-Shark, however, what this will show is where you can start taking the same data that we were searching for in the above scenario of looking for user agents. Given this pcap doesn't have a large amount of user agent data, you can manually come to the same conclusion, but that is not an ideal way to do that. Let us get into it now.

Carving out the data: Just like above, not exact, because I am not focused on the request method, I am focused on the user agent. I will filter it using two different commands so you can see the difference in output.

Command 1

```
tshark -r 3a86135ddc91dd726f831d9020a9a3d7fe7c0747ebda91f02fc477f2ae1331e6.pcap -T fields -e ip.src -e http.user_agent | awk '{print $1,$2;}' | sort | uniq -c | sort -nr
```



```
brian@parrot: ~/Desktop
└─$ tshark -r 3a86135ddc91dd726f831d9020a9a3d7fe7c0747ebda91f02fc477f2ae1331e6.pcap -T fields -e ip.src -e http.user_agent | awk '{print $1,$2;}' | sort | uniq -c | sort -nr
7626 131.247.120.45
3791 172.16.2.96
1129 65.181.112.240
150 104.16.21.35
147 187.33.247.102
112 192.16.48.200
59 187.33.247.91
50 172.16.2.1
49 187.33.247.109
47 65.181.113.204
36 216.58.202.42
33 187.33.247.83
33 172.16.2.96 Mozilla/5.0
27 187.33.247.121
26 187.33.247.93
26 187.33.247.79
26 187.33.247.106
19 67.199.248.11
18 74.117.178.58
18 172.217.29.238
15 104.16.27.235
8 65.181.125.20
8 158.69.99.213
7 172.16.2.254
6 216.58.202.14
5 187.33.238.74
4 74.117.178.179
4 187.33.238.72
2 172.16.2.96 Microsoft
```

Command 2

```
tshark -r 3a86135ddc91dd726f831d9020a9a3d7fe7c0747ebda91f02fc477f2ae1331e6.pcap -T fields -e ip.src -e http.user_agent | awk '{if ($2) print $0;}' | sort | uniq -c | sort -nr
```

```
brian@parrot: ~/Desktop  
$ tshark -r 3a86135ddc91dd726f831d9020a9a3d7fe7c0747ebda91f02fc477f2ae1331e6.pcap -T fields -e ip.src -e http.user_agent | awk '{if ($2) print $0;}' | sort | uniq -c | sort -nr  
33 172.16.2.96 Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko  
2 172.16.2.96 Microsoft NCSI
```

Using Tshark for a Deeper Dive

As you can see from the output from command 1 and command 2, one is nowhere near as clean. Because this search is looking for source ip and matching that to a user agent, the first command will print out every source IP it finds and every user agent. We adjusted the command to then only show if there is data in field 2 by using a simple if statement inside of awk. Then we have a much cleaner output. We can see from the output above, that 33 times, we saw the user agent Mozilla and version, then outputs the system information of Windows NT 6.1 which is windows 7. This packet does at one point show the version of windows 7, however, that is for another day. I just wanted to show you how you can leverage Wireshark and Tshark together in a more automated fashion with presets to help speed up the hunt time and make it more efficient and less painful.