Mini-project 2: Network Auditing

Team Members: Hrishikesh Patil and Yash Gangwar

Ans 1) We went to censys.io and entered ncsu.edu in the search bar. We came across a number of hosts with different Ip addresses and cidrs. We noticed that most of the ips had an ip address in the range of 152.1.x.x, 152.7.x.x and 152.14.x.x. We then went to ipinfo.io to check the cidrs of the ips and we selected the cidrs 152.1.0.0/16 and 152.7.0.0/16 for our experiment. Fig 6 shows all the IPv4 blocks owned by NCSU.

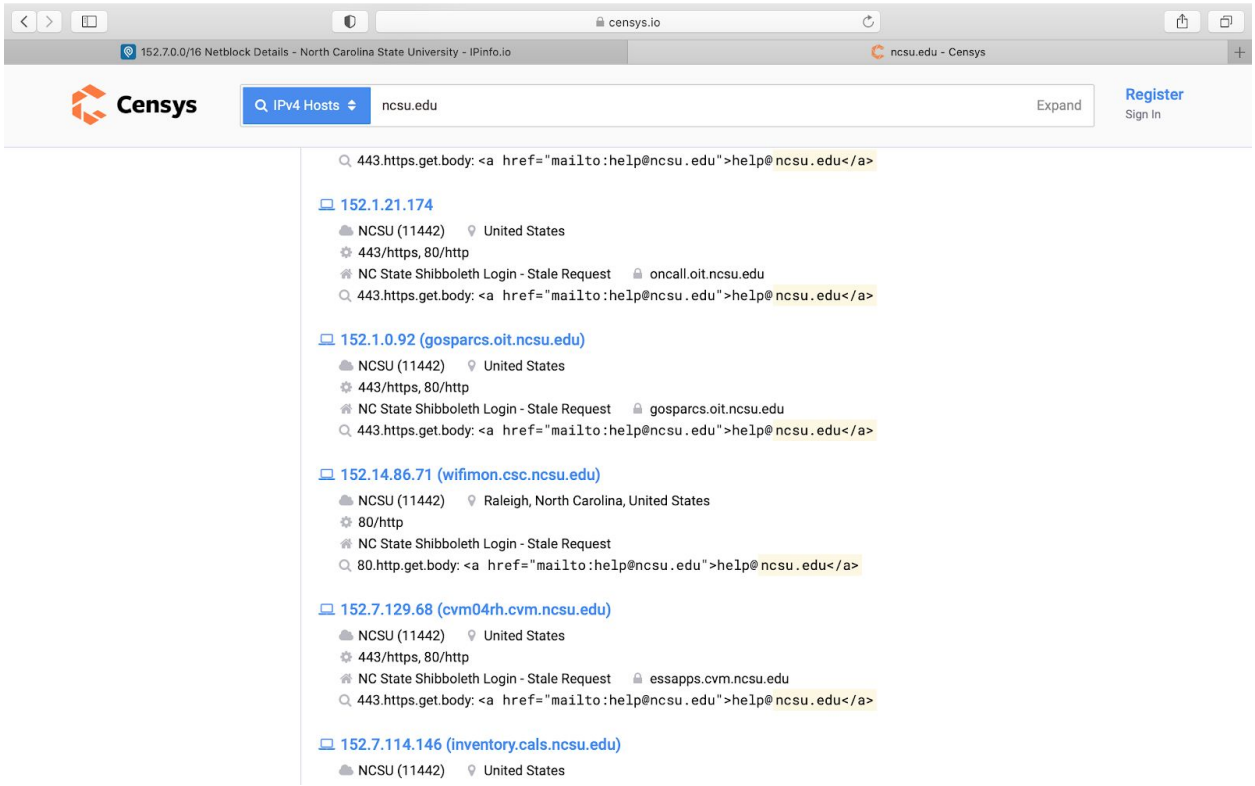

Fig 1- Screenshot depicting the ip address obtained from the search of ncsu.edu in censys.io

The IPv4 blocks we considered are

1) CIDR- 152.1.0.0/16 - https://ipinfo.io/AS11442/152.1.0.0/16
   Network Name- NCSU
   Autonomous System Number - AS11442
   Autonomous System Name - NCSU

   From Fig 2 we can see that the cidr range is 152.1.0.0/16, the network name is NCSU, the autonomous system number is AS11442.

On clicking the AS11442 link we were redirected to a page which shows the AS Name as seen in Fig3.

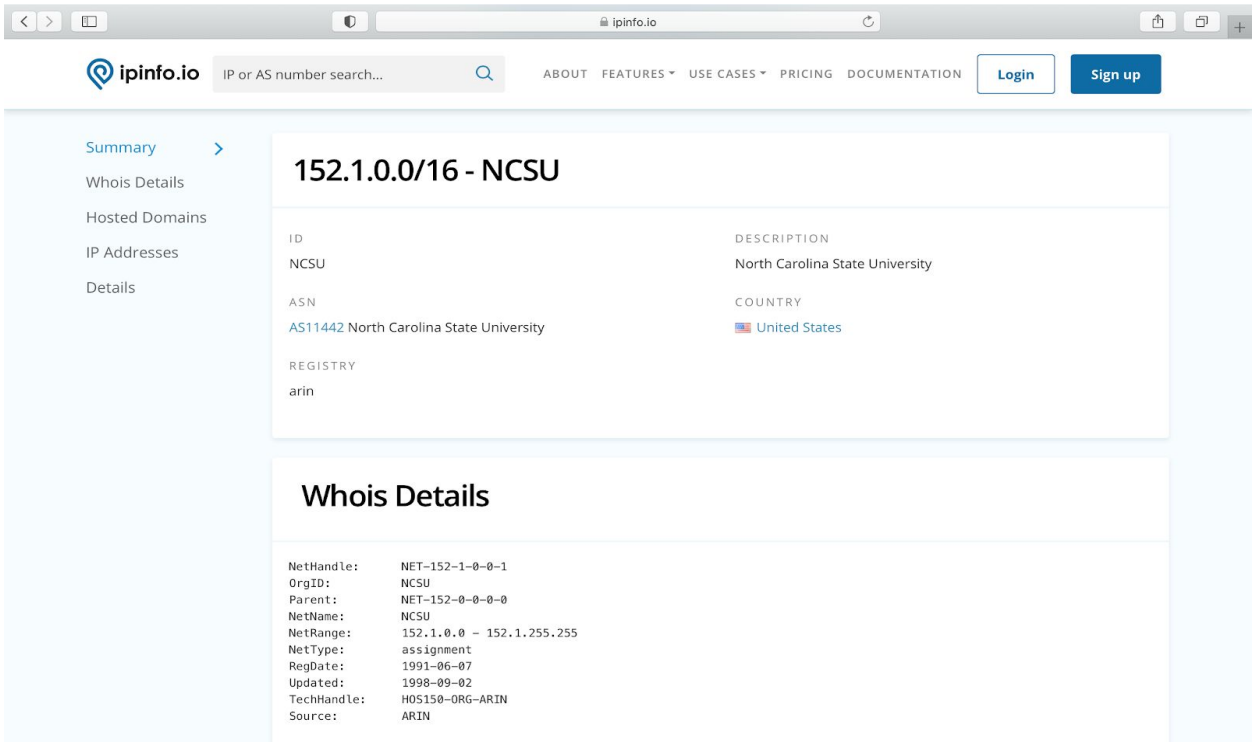

Figure 2: Screenshot showing the CIDR, Network Name, AS Number

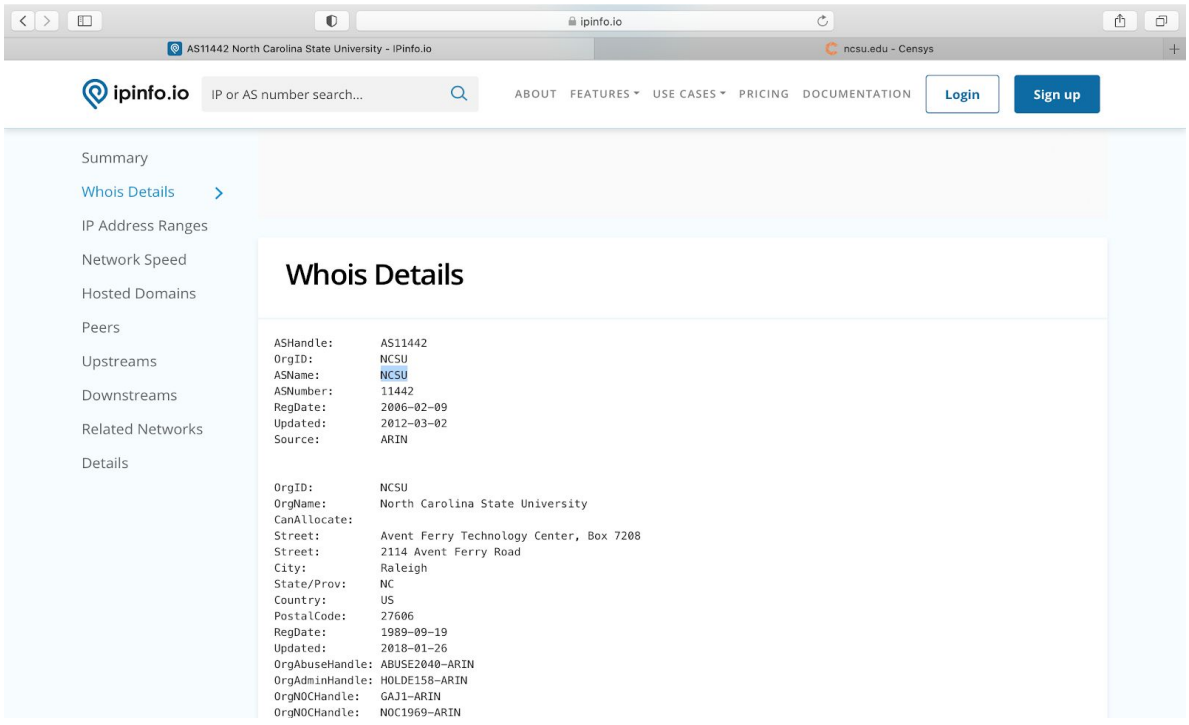

Figure 3: Screenshot showing the AS Name

2) CIDR- 152.7.0.0/16. https://ipinfo.io/AS11442/152.7.0.0/16
   Network Name- NCSU2
   Autonomous System Number - AS11442
   Autonomous System Name - NCSU

From Fig 4 we can see that the cidr range is 152.7.0.0/16, the network name is NCSU2, the autonomous system number is AS11442.

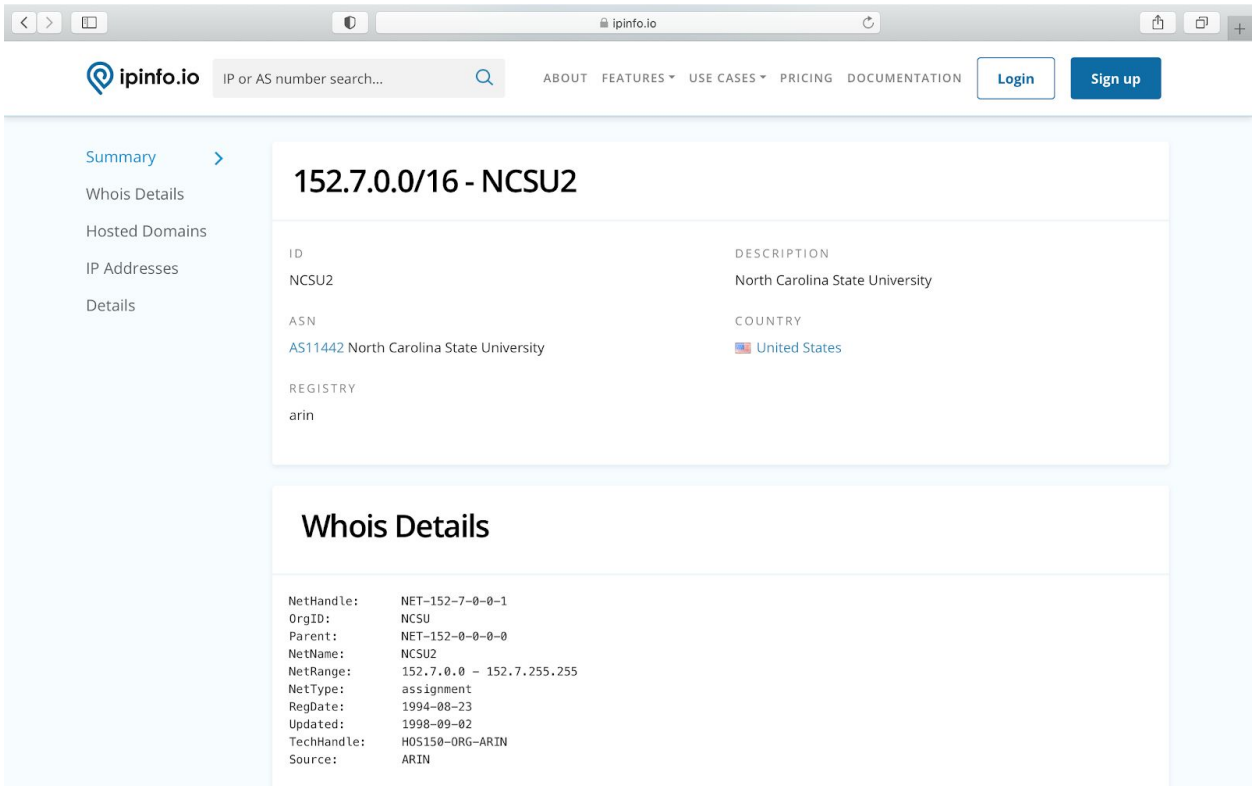

Figure 4: Screenshot showing the CIDR, Network Name, AS Number, AS Name

On clicking the AS11442 link we were redirected to a page which shows the AS Name as seen in Fig 5.

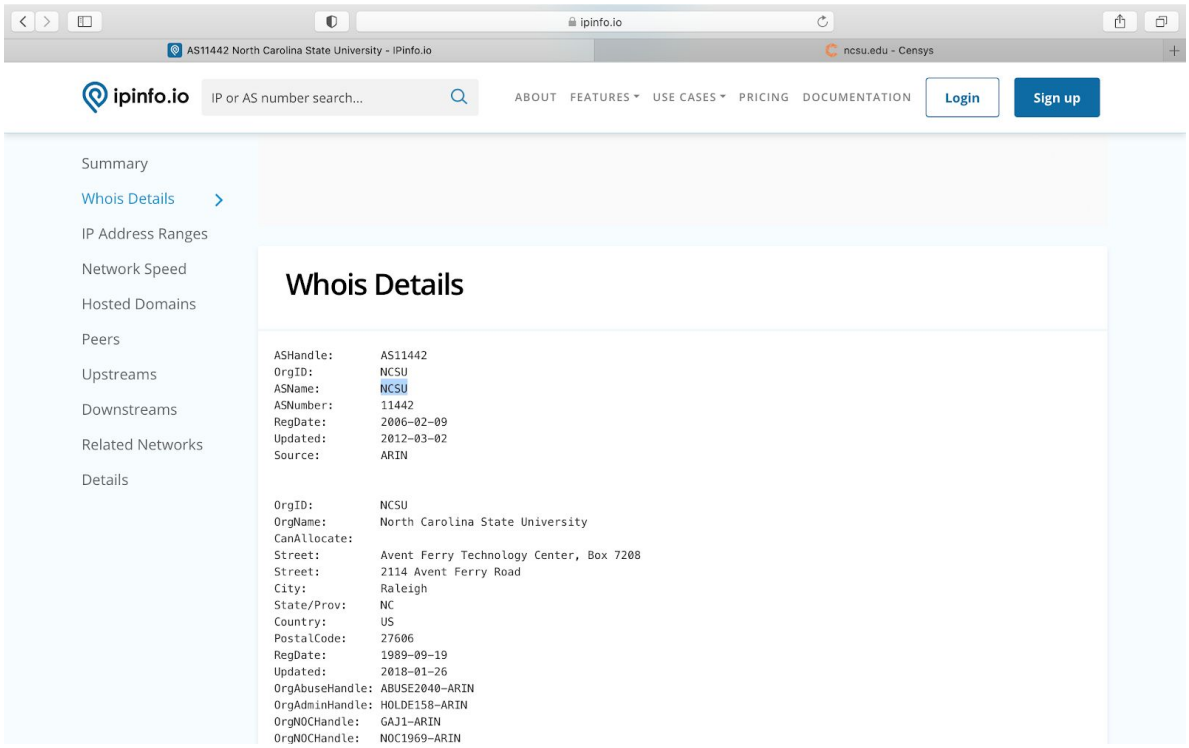

Figure 5: Screenshot showing the AS Name

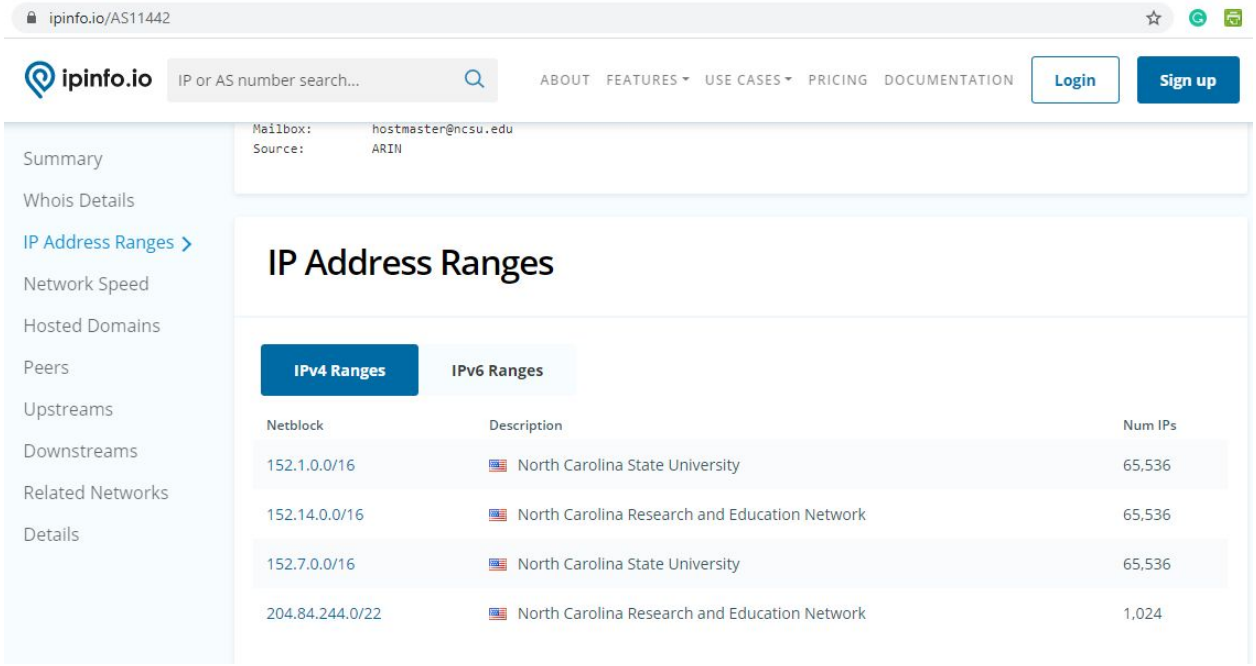

Figure 6: All NCSU IPv4 block: 152.1.0.0/16, 152.7.0.0/16, 152.14.0.0/16 and 204.84.244.0/22

Ans 2)

a) **Host by operating system:**

	Operating System	Host
0	Windows	142
1	Ubuntu	69
2	Unix	12
3	Raspbian	3
4	Fedora	4
5	CentOS	8
6	Debian	4
7	Win64	1

Table 1: CIDR 152.1.0.0/16

	Operating System	Host
0	Windows	60
1	CentOS	51
2	Ubuntu	8
3	RedHat	1
4	Unix	1

Table 2:CIDR 152.7.0.0/16

From Table 1 we see that CIDR 152.1.0.0/16 uses Windows OS for most of the hosts followed by Ubuntu and Unix and from Table 2 we see that CIDR 152.7.0.0/16 uses Windows OS for most of the hosts followed by CentOS and Ubuntu. These results were obtained from using Python API by querying the respective CIDR for the operating system('metadata.os'). These results are useful because they give an insight of the most used and all used operating systems on the CIDRs.

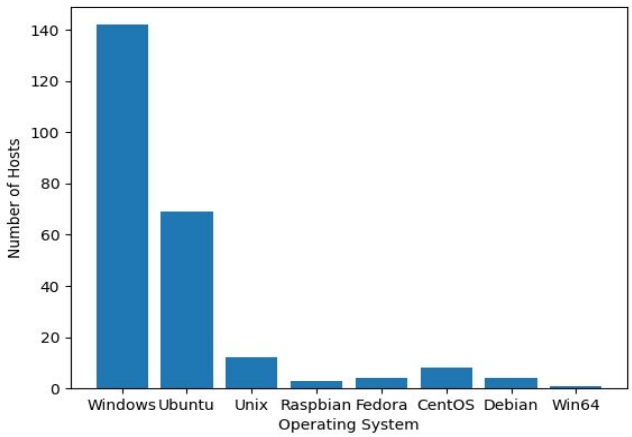

Figure7 : Bar graph plot of CIDR 152.1.0.0/16

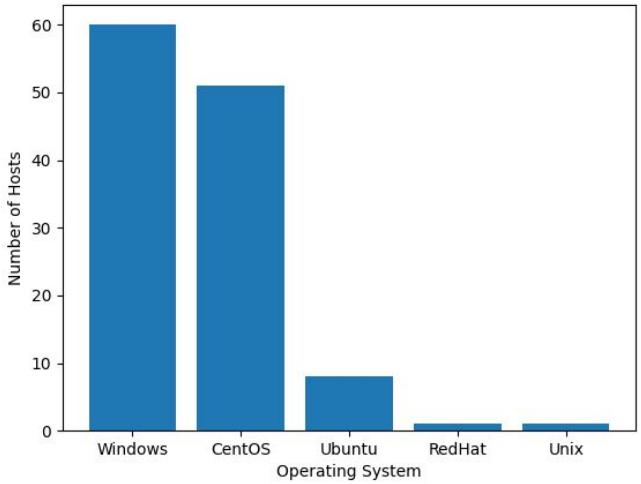

Figure8 : Bar graph plot of CIDR 152.7.0.0/16

b) Hosts by Webservers:

	Webserver	Host
0	Microsoft-IIS/10.0	119
1	Virata-EmWeb/R6_2_1	152
2	Web-Server/3.0	14
3	Apache	230
4	HP_Compact_Server	30
..	...	...
107	Apache/2.4.33	1
108	nginx/1.19.0	2
109	Boa/0.94.14rc19	1
110	Allegro-Software-RomPager/4.34	2
111	MEDIACTR-NAE-2	1

Table 3: Webserver in CIDR 152.1.0.0/16

From table 3 we see that hosts in CIDR 152.1.0.0/16 use many Web Servers and most of the hosts are using the Apache and Microsoft web servers. There are others like Virate-EmWeb, Hp_Compact_Server too. These results are useful because they give an insight of the most used and all used web servers used by the hosts on the CIDR. These results were obtained from using Python API by querying the respective CIDR for the Webserver (“80.http.get.headers.server”, “443.https.get.headers.server”).

	Webserver	Host
0	Microsoft-IIS/10.0	51
1	Apache	60
2	EPSON_Linux UPnP/1.0 Epson UPnP SDK/1.0	10
3	Microsoft-IIS/8.0	4
4	Apache/2.2.15 (Red Hat)	3
5	nginx/1.12.2	5
6	nginx/1.15.8	2
7	Apache/2.4.6 (CentOS)	2
8	Apache/2.2.15 (CentOS)	1
9	Microsoft-IIS/8.5	6
10	Microsoft-WinCE/7.00	1
11	nginx/1.18.0	1
12	uhttpd/1.0.0	3
13	nginx/1.7.10	2
14	nginx/1.16.1	1
15	lighttpd/1.4.18	2
16	GoAhead-Webs	1
17	lighttpd/1.4.32	2
18	Apache/2.4.34 (Unix)	1
19	Apache/2.4.29 (Ubuntu)	1
20	Apache/2.4.18 (Ubuntu)	1
21	Apache-Coyote/1.1	4
22	nginx/1.17.9	1
23	EPSON-HTTP/1.0	2
24	Microsoft-IIS/7.5	3
25	nginx/1.15.9	2
26	Virata-EmWeb/R6_2_1	4
27	lighttpd/1.4.23	1
28	Apache/2.4.41 (Ubuntu)	1
29	Jetty(9.4.31.v20200723)	2

Table 4: Webserver in CIDR 152.7.0.0/16

From table 4 we see that hosts in CIDR 152.7.0.0/16 use many Web Servers and most of the hosts are using the Apache and Microsoft web servers. There are others like Virate-EmWeb, nginx too. These results are useful because they give an insight of the most used and all used web servers used by the hosts on the CIDR. These results were obtained from using Python API by querying the respective CIDR for the Webserver (“80.http.get.headers.server”, “443.https.get.headers.server”).

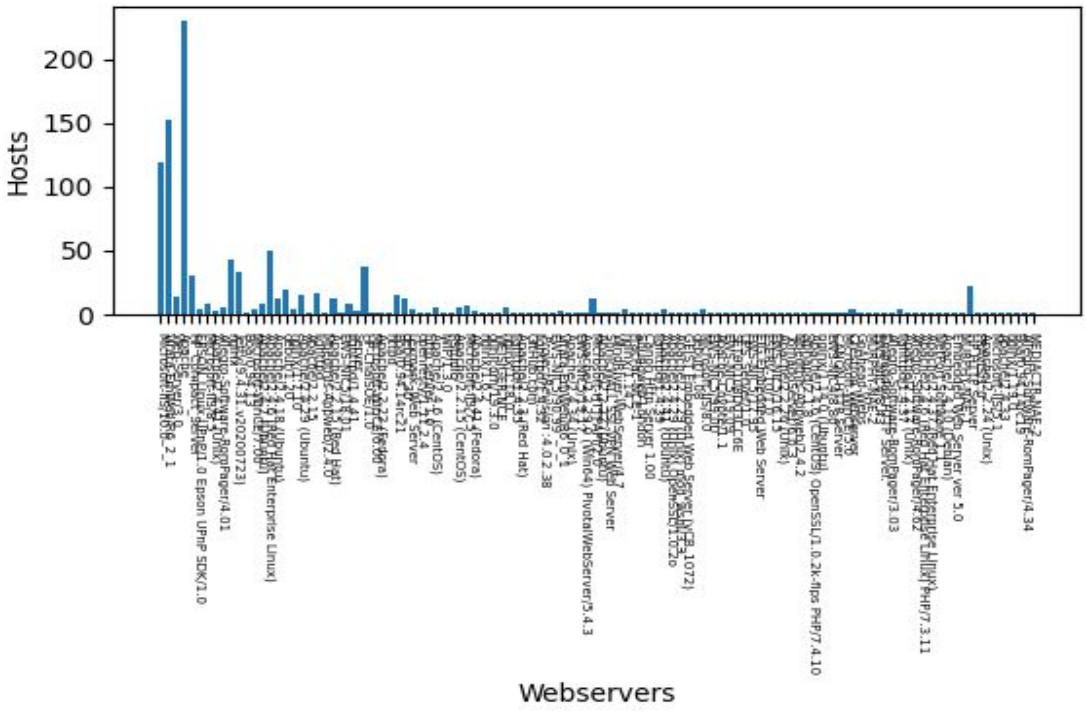

Figure9 : Web Servers for CIDR 152.1.0.0/16

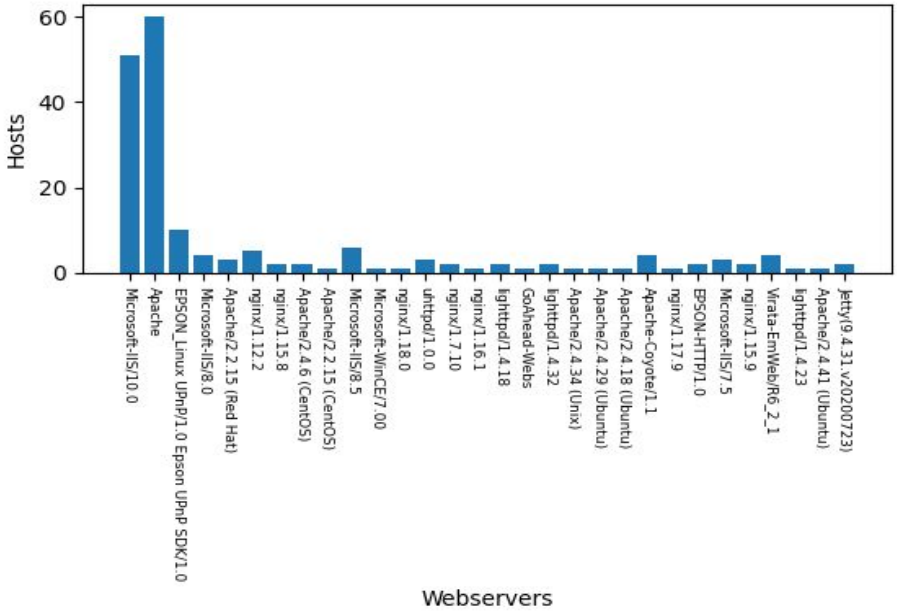

Figure10 : Web Servers for CIDR 152.7.0.0/16

c) Hosts by Protocols:

	Protocols	Host
0	80/http	651
1	443/https	544
2	8080/http	105
3	22/ssh	204
4	25/smtp	32
5	21/ftp	67
6	3306/mysql	29
7	11211/memcached	1
8	53/dns	13
9	16992/http	1
10	47808/bacnet	3
11	5672/amqp	1
12	502/modbus	1
13	20000/dnp3	1
14	1521/oracle	1
15	5432/postgres	3
16	623/ipmi	1
17	995/pop3s	1
18	110/pop3	1
19	587/smtp	1
20	143/imap	1
21	3389/rdp	1

Table 5: Protocols in CIDR 152.1.0.0/16

	Protocols	Host
0	443/https	172
1	80/http	202
2	22/ssh	50
3	5432/postgres	14
4	8080/http	20
5	3389/rdp	2
6	3306/mysql	13
7	21/ftp	13
8	993/imaps	3
9	465/smtp	2
10	995/pop3s	2
11	110/pop3	2
12	143/imap	3
13	53/dns	8
14	587/smtp	3
15	8888/http	2
16	502/modbus	1
17	25/smtp	3
18	47808/bacnet	2
19	1883/mqtt	1

Table 6: Protocols in CIDR 152.7.0.0/16

From table 5 we see that hosts in CIDR 152.1.0.0/16 use many protocols and most of the hosts are using the http, https and ssh protocols. There are others like smtp, ftp too. These results are useful because they give an insight of the most used and all used protocols used by the hosts on the CIDR. These results were obtained from using Python API by querying the respective CIDR for the protocols field (‘protocols’)

From table 6 we see that hosts in CIDR 152.7.0.0/16 use many protocols and most of the hosts are using the http, https and ssh protocols. There are others like smtp, ftp too. These results are useful because they give an insight of the most used and all used protocols used by the hosts on the CIDR. These results were obtained from using Python API by querying the respective CIDR for the protocols field (‘protocols’)

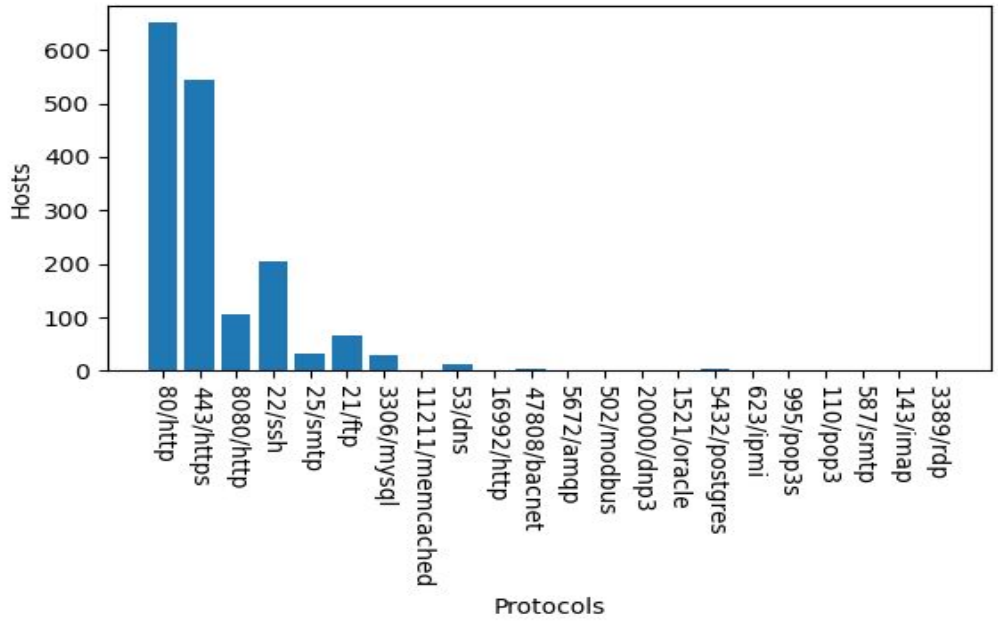

Figure11 : Protocols in CIDR 152.1.0.0/16

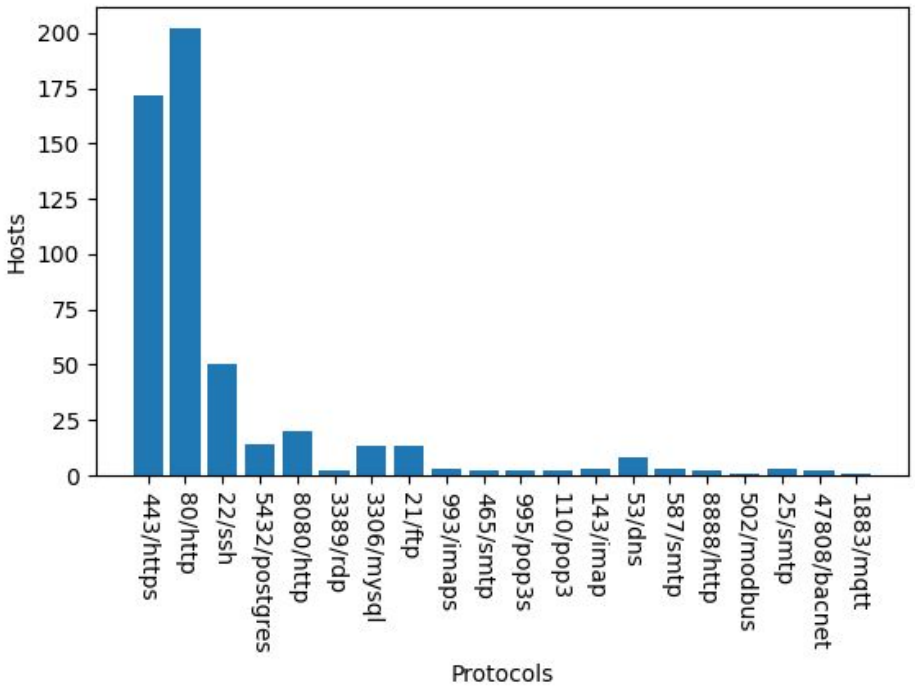

Figure12 : Protocols in CIDR 152.7.0.0/16

d) TLS Version:

	TLS version	Host
0	TLSv1.0	83
1	TLSv1.2	442

Table7: CIDR 152.1.0.0/16

	TLS version	Host
0	TLSv1.2	160
1	TLSv1.0	11

Table8 : CIDR 152.7.0.0/16

From table 7 we see that hosts in CIDR 152.1.0.0/16 use 2 TLS versions and they are TLSv1.0 and TLSv1.2. These results are useful because they give an insight of the TLS versions used by the hosts on the CIDR. TLSv1.0 has man in the middle attack vulnerability. So it is important to upgrade the device to be more secure.
These results were obtained from using Python API by querying the respective CIDR for the tls version field (‘443.https.tls.version’).

From table 8 we see that hosts in CIDR 152.7.0.0/16 use 2 TLS versions and they are TLSv1.0 and TLSv1.2. These results are useful because they give an insight of the TLS versions used by the hosts on the CIDR. TLSv1.0 has man in the middle attack vulnerability. So it is important to upgrade the device to be more secure.
These results were obtained from using Python API by querying the respective CIDR for the tls version field (‘443.https.tls.version’).

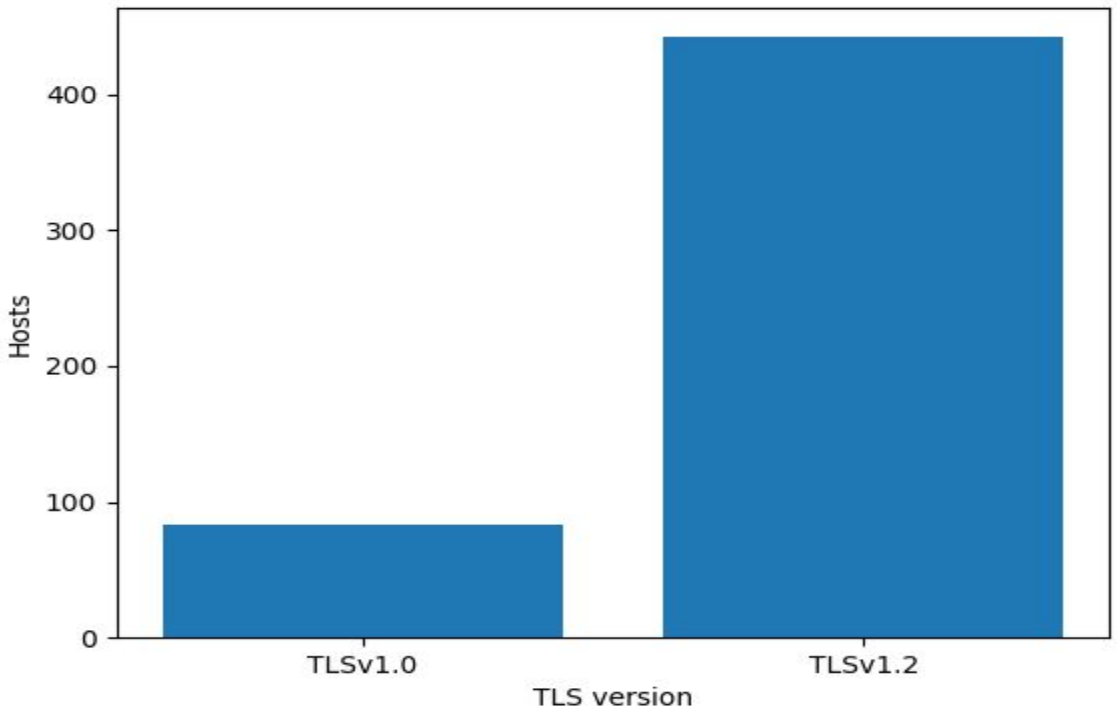

Figure 13 : CIDR 152.1.0.0/16 TLS version

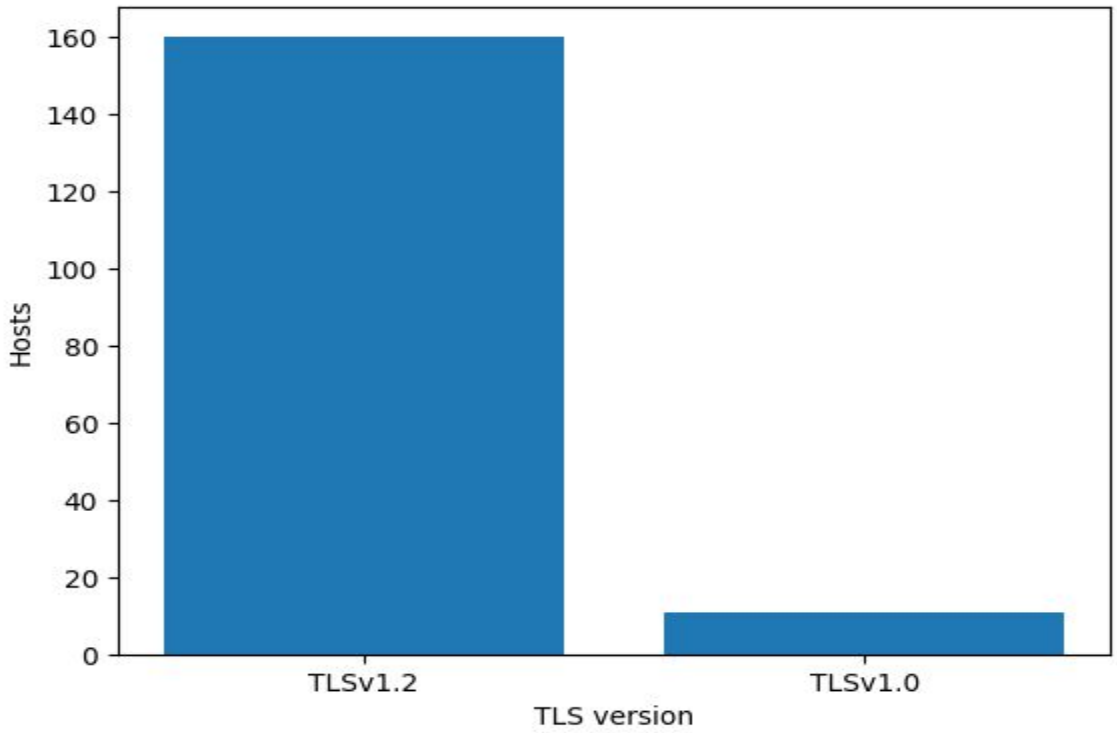

Figure 14: CIDR block: 152.7.0.0/16 TLS Version

Ans 3)

Poodle attacks:

CIDR 152.1.0.0/16 has 96 hosts and CIDR 152.7.0.0/16 has 12 hosts who are vulnerable to poodle attacks because they have support for SSL_3. Hosts and servers that support SSLv3 and CBC-mode are potentially to an active man in the middle attack even if the site supports TLS. For description of all CVE refer (https://cve.mitre.org/cve/search_cve_list.html)

List of hosts in CIDR 152.1.0.0/16:

['152.1.94.226', '152.1.105.197', '152.1.46.95', '152.1.227.104', '152.1.238.91', '152.1.166.106', '152.1.84.15', '152.1.56.161', '152.1.53.171', '152.1.152.212', '152.1.142.80', '152.1.105.159', '152.1.175.140', '152.1.109.5', '152.1.109.184', '152.1.168.242', '152.1.197.86', '152.1.145.200', '152.1.226.118', '152.1.152.159', '152.1.215.61', '152.1.56.22', '152.1.102.57', '152.1.166.48', '152.1.230.39', '152.1.100.213', '152.1.127.82', '152.1.206.107', '152.1.62.44', '152.1.46.106', '152.1.168.248', '152.1.226.128', '152.1.47.73', '152.1.168.60', '152.1.105.237', '152.1.208.16', '152.1.45.118', '152.1.105.158', '152.1.215.54', '152.1.207.70', '152.1.40.88', '152.1.91.249', '152.1.171.100', '152.1.213.241', '152.1.105.131', '152.1.111.190', '152.1.26.145', '152.1.0.92', '152.1.102.25', '152.1.227.102', '152.1.246.12', '152.1.72.117', '152.1.166.19', '152.1.102.179', '152.1.53.245', '152.1.227.140', '152.1.168.243', '152.1.31.142', '152.1.95.140', '152.1.160.232', '152.1.166.18', '152.1.38.179', '152.1.118.165', '152.1.192.228', '152.1.81.118', '152.1.198.105', '152.1.168.247', '152.1.152.176', '152.1.35.17', '152.1.159.136', '152.1.227.141', '152.1.32.10', '152.1.52.127', '152.1.168.245', '152.1.105.233', '152.1.56.143', '152.1.172.58', '152.1.39.238', '152.1.56.62', '152.1.160.229', '152.1.168.244', '152.1.102.82', '152.1.212.198', '152.1.37.239',

'152.1.45.120', '152.1.38.155', '152.1.178.169', '152.1.44.14', '152.1.213.160', '152.1.31.103', '152.1.166.196', '152.1.168.246', '152.1.152.122', '152.1.56.98', '152.1.152.198', '152.1.105.234']

List of hosts in CIDR 152.7.0.0/16:

['152.7.99.1', '152.7.95.10', '152.7.200.12', '152.7.99.62', '152.7.196.6', '152.7.155.253', '152.7.200.11', '152.7.204.9', '152.7.200.9', '152.7.192.65', '152.7.192.84', '152.7.129.68']

CVE for a 152.7.99.62 who is vulnerable to Poodle attacks:

['CVE-2019-1559', 'CVE-2014-0117', 'CVE-2017-7679', 'CVE-2016-0736', 'CVE-2015-3185', 'CVE-2015-3184', 'CVE-2018-1312', 'CVE-2016-4975', 'CVE-2018-5407', 'CVE-2017-3736', 'CVE-2017-3737', 'CVE-2014-0226', 'CVE-2017-3735', 'CVE-2017-3738', 'CVE-2014-3523', 'CVE-2017-15710', 'CVE-2017-15715', 'CVE-2013-6438', 'CVE-2014-0118', 'CVE-2018-0737', 'CVE-2018-0734', 'CVE-2018-0732', 'CVE-2018-17199', 'CVE-2017-9788', 'CVE-2018-0739', 'CVE-2014-8109', 'CVE-2017-9798', 'CVE-2016-2161', 'CVE-2016-8612', 'CVE-2019-1552', 'CVE-2014-0231', 'CVE-2013-4352', 'CVE-2019-0220', 'CVE-2014-0098', 'CVE-2018-1283', 'CVE-2016-8743']

Censys.io: query all the hosts who have support for SSLv3 because these are vulnerable to the poodle attacks and stored the IP addresses for the vulnerable hosts. Using these IP addresses we can get all information about CVE by querying the Shodan API.

Freak attack:
CIDR block 152.1.0.0/16 has 3 hosts vulnerable to Freak attack. CIDR block 152.1.0.0/16 has no host vulnerable to Freak attack. Servers that accept RSA_EXPORT cipher suites put their users at risk from the FREAK attack

Censys query to check for RSA_Export flag in all hosts, then stored the IP addresses of vulnerable hosts. Querying all the hosts using the Shodan API to get CVE for vulnerable hosts. For description of all CVE refer (https://cve.mitre.org/cve/search_cve_list.html)

Lists of hosts vulnerable to Freak attack:

['152.1.109.5', '152.1.220.152', '152.1.52.127']

For host **152.1.109.5:**

*['CVE-2011-5000', 'CVE-2017-15906', 'CVE-2014-1692', 'CVE-2010-5107', 'CVE-2012-0814', 'CVE-2016-10708', 'CVE-2010-4478', 'CVE-2016-0777', 'CVE-2011-4327', 'CVE-2010-4755', '**CVE-2015-0204**', 'CVE-2015-4000']*

For host **152.1.220.152:**

*['**CVE-2015-0204**', 'CVE-2015-4000']*

For host **152.1.52.127:**

['CVE-2011-5000', 'CVE-2017-15906', 'CVE-2014-1692', 'CVE-2010-5107', 'CVE-2016-10708', 'CVE-2010-4478', 'CVE-2016-0777', 'CVE-2011-4327', 'CVE-2010-4755', 'CVE-2012-0814']

CVE-2015-0204: 'The ssl3_get_key_exchange function in s3_clnt.c in OpenSSL before 0.9.8zd, 1.0.0 before 1.0.0p, and 1.0.1 before 1.0.1k allows remote SSL servers to conduct RSA-to-EXPORT_RSA downgrade attacks and facilitate brute-force decryption by offering a weak ephemeral RSA key in a noncompliant role, related to the "FREAK" issue. NOTE: the scope of this CVE is only client code based on OpenSSL, not EXPORT_RSA issues associated with servers or other TLS implementations.

Logjam attack:
CIDR block 152.1.0.0/16 has 3 hosts vulnerable to Logjam attack. CIDR block 152.1.0.0/16 has no host vulnerable to Logjam attack.
Censys query to check for Export_DHE flag in all hosts, then stored the IP addresses of vulnerable hosts. Querying all the hosts using the Shodan API to get CVE for vulnerable hosts. For description of all CVE refer (https://cve.mitre.org/cve/search_cve_list.html)

Lists of hosts vulnerable to Logjam attack: *['152.1.109.5', '152.1.220.152', '152.1.52.127']*

For host **152.1.109.5:**

['CVE-2011-5000', 'CVE-2017-15906', 'CVE-2014-1692', 'CVE-2010-5107', 'CVE-2012-0814', 'CVE-2016-10708', 'CVE-2010-4478', 'CVE-2016-0777', 'CVE-2011-4327', 'CVE-2010-4755', 'CVE-2015-0204', 'CVE-2015-4000']

For host **152.1.220.152:**

['CVE-2015-0204', 'CVE-2015-4000']

For host **152.1.52.127:**

['CVE-2011-5000', 'CVE-2017-15906', 'CVE-2014-1692', 'CVE-2010-5107', 'CVE-2016-10708', 'CVE-2010-4478', 'CVE-2016-0777', 'CVE-2011-4327', 'CVE-2010-4755', 'CVE-2012-0814']

CVE-2015-4000: 'The TLS protocol 1.2 and earlier, when a DHE_EXPORT ciphersuite is enabled on a server but not on a client, does not properly convey a DHE_EXPORT choice, which allows man-in-the-middle attackers to conduct cipher-downgrade attacks by rewriting a ClientHello with DHE replaced by DHE_EXPORT and then rewriting a ServerHello with DHE_EXPORT replaced by DHE, aka the "Logjam" issue.

152.1.109.5: host is vulnerable to Poodle, Freak, Logjam attacks. (https://censys.io/ipv4/152.1.109.5)

152.1.220.152: Host is vulnerable to Freak and Logjam attacks (https://censys.io/ipv4/152.1.220.152)

152.1.52.127: host is vulnerable to Poodle, Freak, Logjam attacks (https://censys.io/ipv4/152.1.52.127)

Ans 4) Since IPv4 is a 32 bit address, tools such as Censys and Shodan are able to perform wide scans easily but with IPv6 being 128 bit addresses these tools are not able to perform a wide scan due to the extremely high number of ips present in IPv6. As a result tools such as Censys and Shodan work by creating a hitlist of ips rather than performing a wide scan on the internet. Censys and Shodan repeatedly fully probe a densely populated IPv4 space and record time-stable hosts for efficient hitlists but these tools only probe the internet for IPv6 addresses in a sparsely populated way and hence they have to do an informed repeated probing to generate a core set of stable and active addresses. Tools such as ping or nmap are capable of probing IPv6 hosts, but they lack performance which is not the case when these tools probe IPv4 addresses. Tools like masscan & zmap can scan the entire IPv4 Internet in less than five minutes. However these tools lack support for IPv6 network measurements. As a result you need to extend zmap in order to make it IPv6-capable. Therefore, if these tools are not extended to support IPv6 addresses, their effectiveness in probing IPv6 addresses is zero.**[1]**

The speed at which the vulnerabilities are scanned is related to the node population density. The population density is measured as the number of active hosts divided by the total number of possible addresses. In an IPv4 network, there could be 100 active hosts on a /24 subnet with a total of 254 possible addresses for a density of .393. On an IPv6 /64 network with 100 active hosts the density would be .0000000000000000542. Because of this extremely low population density the effectiveness of tools such as Censys and Shodan take a hit as it is equivalent to finding a specific single grain of sand in the Sahara desert! Another place where the effectiveness of the scanning tools for IPv6 addresses lacks is the cost. If we purchase a license for 256 IP addresses, then we are limited to scanning a /24 IPv4 subnet or a specific list of 256 hosts. However, with IPv6, the subnet ranges are so large and extremely sparsely populated that this software licensing model breaks down. Another place where the effectiveness of tools like censys and shodan lacks is for systems that use both IPv4 and IPv6. Attackers can use multiple address families to attack the hosts and evade detection. For eg. a malware that infects one host over an IPv4 web vulnerability, and then uses IPv6 to spread to other nodes on the local LAN, and those newly infected nodes use either IPv4 or IPv6 to communicate to a botnet command and control network. Finding a connection between these attack trajectories is daunting. This same concept applies to security vulnerability scanning. How does the scanner tools like Censys and Shodan know that the node with the IPv4 address of 10.23.81.102 is the same node with the IPv6 address 5353:ac2:678:21:7a42:c18a:8ade:8a13? A scanner may not be able to correlate information across address families and as a result the effectiveness of the scanning tool takes a hit. **[2]**

References

1. Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, Georg Carle “Scanning the IPv6 Internet: Towards a Comprehensive Hitlist” 2016

2. Scott Hogg, “IPv6 Security Vulnerability Scanning” September 14 2016