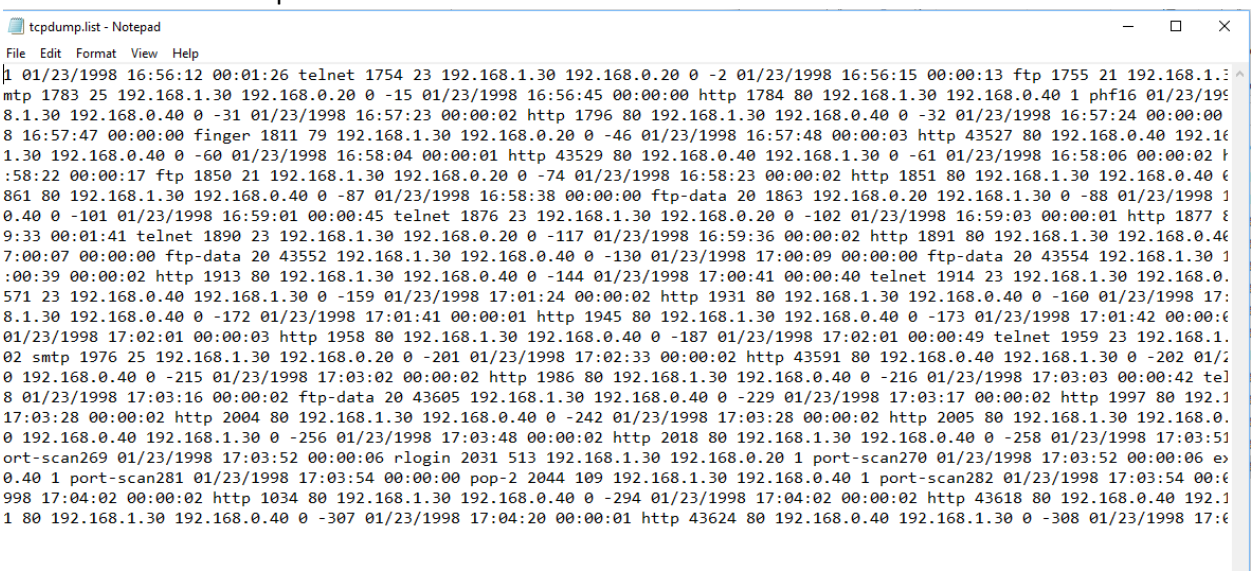
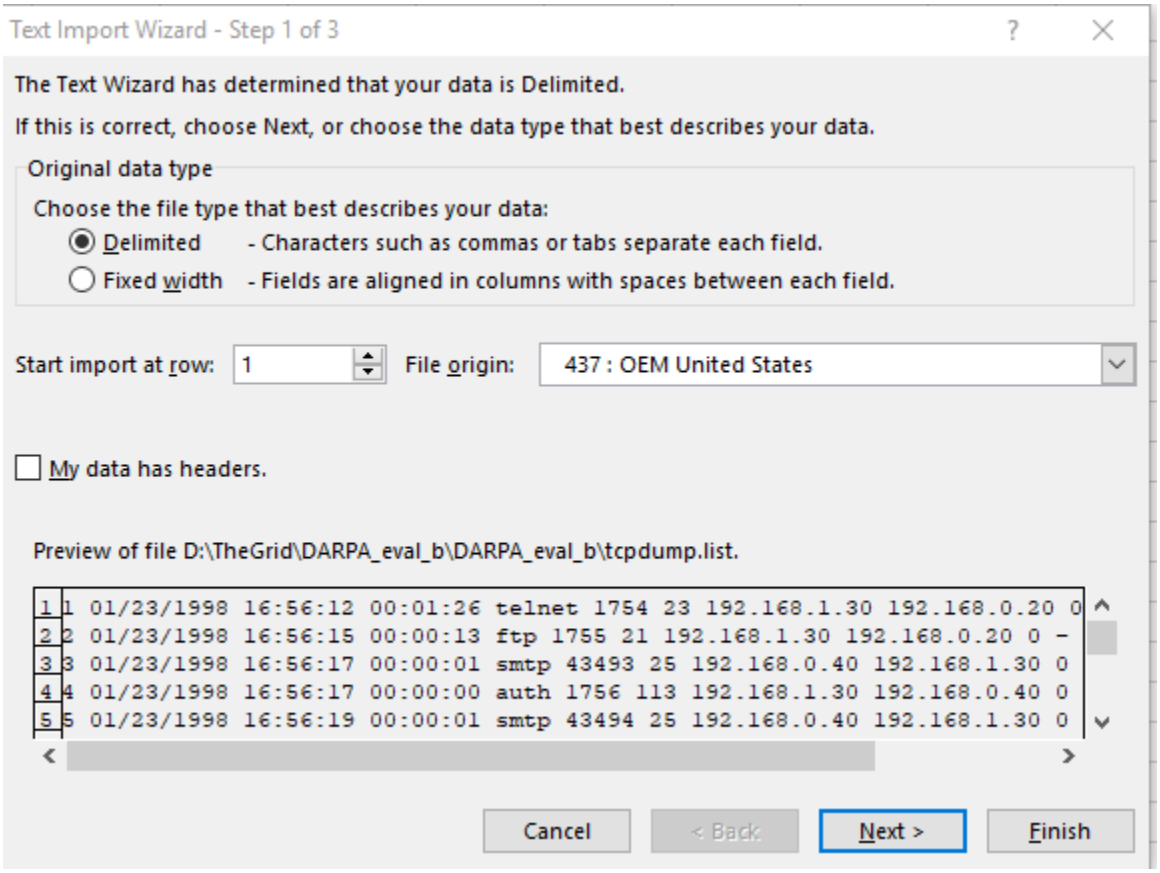
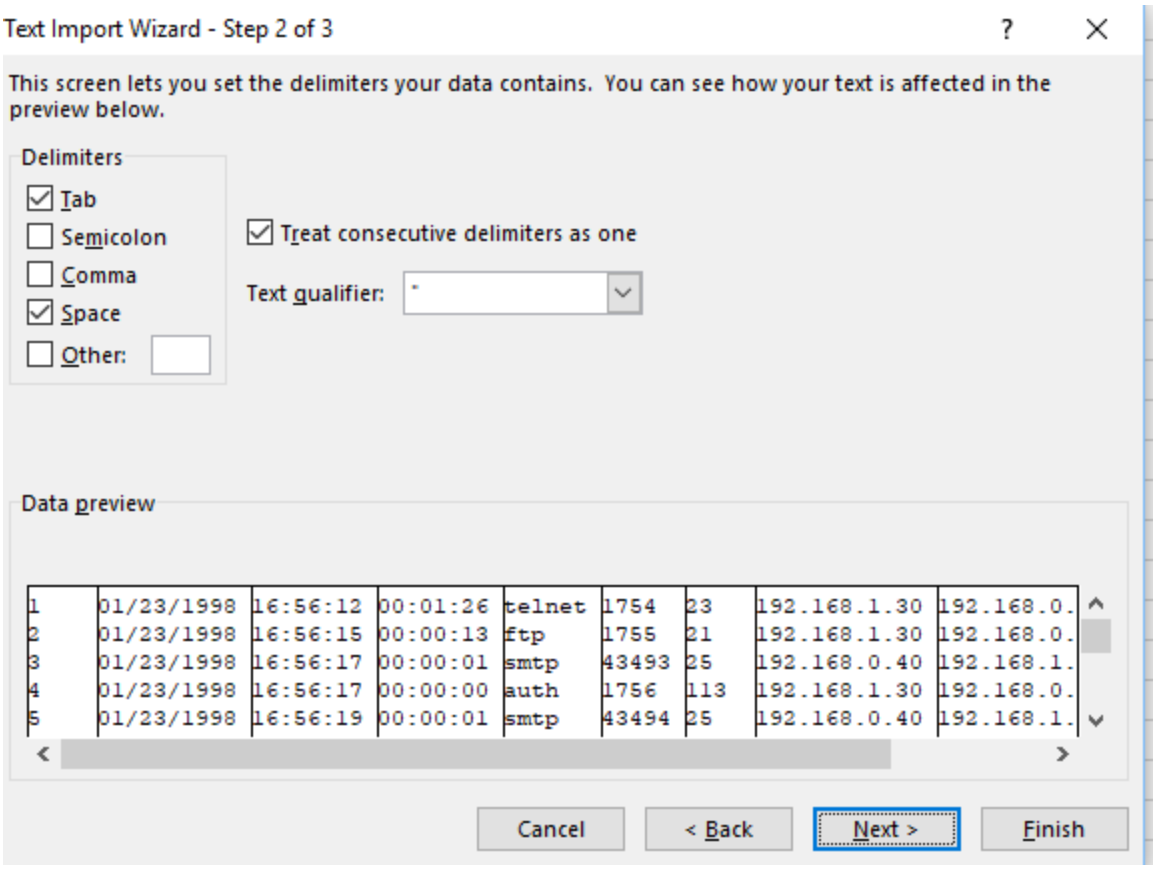
1. Downloaded data dump



2. Open with excel
3. Noting that the data can use spaces between each field so I'm choosing Fixed width



4. Then choosing the Space delimiter to break the data up



This creates a spreadsheet I can now cleanup and organize.

	A	B	C	D	E	F	G	H	I	J	K
1	1	1/23/1998	16:56:12	0:01:26	telnet	1754	23	192.168.1.30	192.168.0.20	0	-
2	2	1/23/1998	16:56:15	0:00:13	ftp	1755	21	192.168.1.30	192.168.0.20	0	-
3	3	1/23/1998	16:56:17	0:00:01	smtp	43493	25	192.168.0.40	192.168.1.30	0	-
4	4	1/23/1998	16:56:17	0:00:00	auth	1756	113	192.168.1.30	192.168.0.40	0	-
5	5	1/23/1998	16:56:19	0:00:01	smtp	43494	25	192.168.0.40	192.168.1.30	0	-
6	6	1/23/1998	16:56:19	0:00:00	auth	1761	113	192.168.1.30	192.168.0.40	0	-
7	7	1/23/1998	16:56:19	0:00:01	ftp-data	20	1762	192.168.0.20	192.168.1.30	0	-
8	8	1/23/1998	16:56:22	0:00:00	ftp-data	20	1767	192.168.0.20	192.168.1.30	0	-
9	9	1/23/1998	16:56:24	0:00:02	ftp-data	20	1768	192.168.0.20	192.168.1.30	0	-
10	10	1/23/1998	16:56:25	0:01:01	telnet	1769	23	192.168.1.30	192.168.0.20	0	-
11	11	1/23/1998	16:56:27	0:00:00	ftp-data	20	1770	192.168.0.20	192.168.1.30	0	-
12	12	1/23/1998	16:56:36	0:00:03	finger	1772	79	192.168.1.30	192.168.0.20	0	-
13	13	1/23/1998	16:56:42	0:00:03	smtp	1778	25	192.168.1.30	192.168.0.20	0	-
14	14	1/23/1998	16:56:43	0:00:03	smtp	1783	25	192.168.1.30	192.168.0.20	0	-
15	15	1/23/1998	16:56:45	0:00:00	http	1784	80	192.168.1.30	192.168.0.40	1	phf

I don’t need column A so I’ll delete that and label the rest of the colums.

	A	B	C	D	E	F	G	H	I	J
1	Start Date	Start Time	Duration	Server	Src Port	Dest Port	Src IP	Destination IP	Attack Score	Name
2	1/23/1998	16:56:12	0:01:26	telnet	1754	23	192.168.1.30	192.168.0.20	0	-
3	1/23/1998	16:56:15	0:00:13	ftp	1755	21	192.168.1.30	192.168.0.20	0	-
4	1/23/1998	16:56:17	0:00:01	smtp	43493	25	192.168.0.40	192.168.1.30	0	-
5	1/23/1998	16:56:17	0:00:00	auth	1756	113	192.168.1.30	192.168.0.40	0	-
6	1/23/1998	16:56:19	0:00:01	smtp	43494	25	192.168.0.40	192.168.1.30	0	-
7	1/23/1998	16:56:19	0:00:00	auth	1761	113	192.168.1.30	192.168.0.40	0	-
8	1/23/1998	16:56:19	0:00:01	ftp-data	20	1762	192.168.0.20	192.168.1.30	0	-
9	1/23/1998	16:56:22	0:00:00	ftp-data	20	1767	192.168.0.20	192.168.1.30	0	-
10	1/23/1998	16:56:24	0:00:02	ftp-data	20	1768	192.168.0.20	192.168.1.30	0	-

I'll save this as a csv so I can import it into R or Python.

setwd("D:/TheGrid/DARPA_eval_b/DARPA_eval_b")
data<-read.csv("sample.csv", header=TRUE)

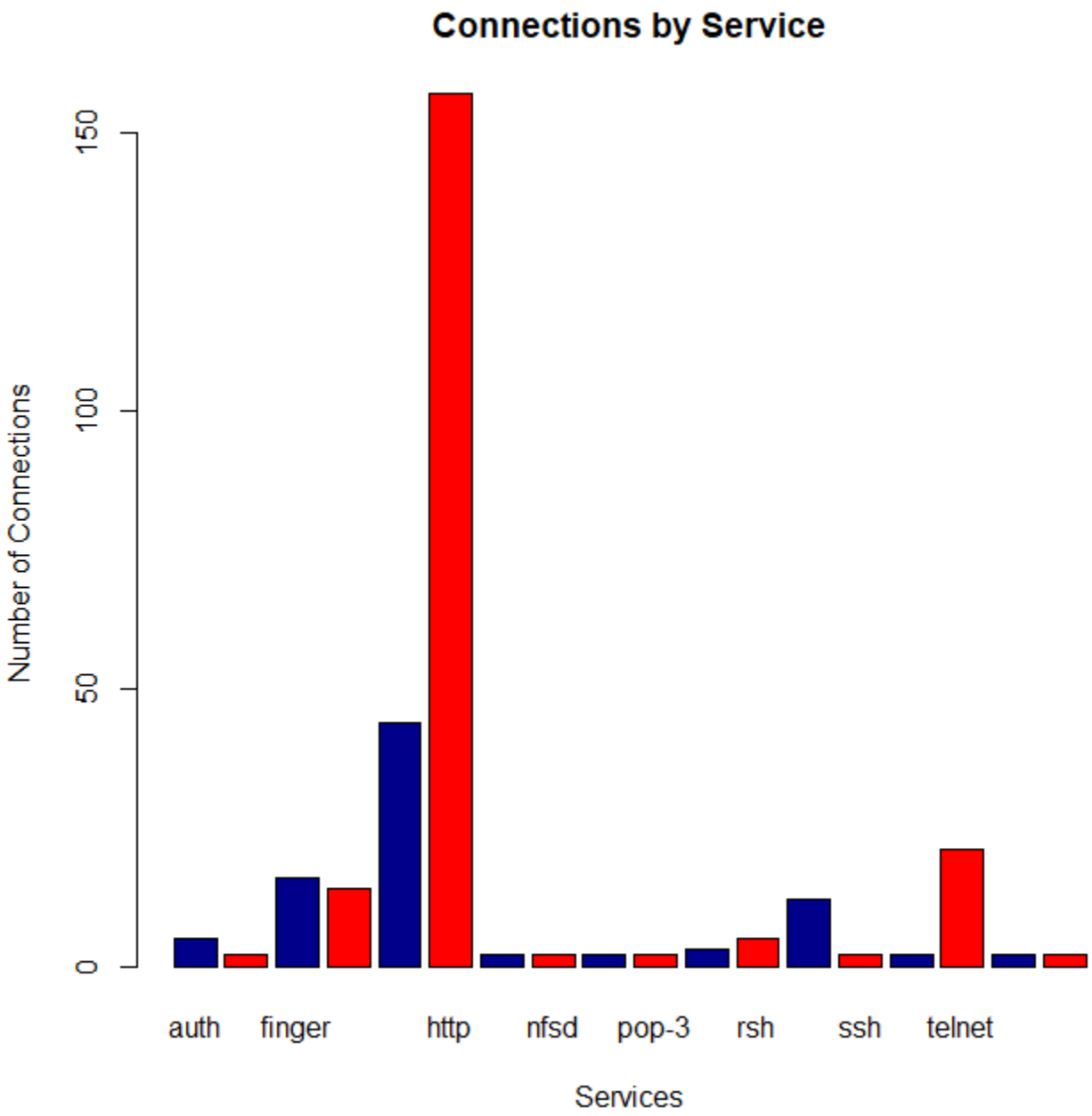
```
> data
    Start.Date Start.Time Duration    Server Src.Port Dest.Port       Src.IP
1    1/23/1998   16:56:12  0:01:26    telnet     1754        23 192.168.1.30
2    1/23/1998   16:56:15  0:00:13       ftp     1755        21 192.168.1.30
3    1/23/1998   16:56:17  0:00:01      smtp    43493        25 192.168.0.40
4    1/23/1998   16:56:17  0:00:00      auth     1756       113 192.168.1.30
5    1/23/1998   16:56:19  0:00:01      smtp    43494        25 192.168.0.40
6    1/23/1998   16:56:19  0:00:00      auth     1761       113 192.168.1.30
7    1/23/1998   16:56:19  0:00:01 ftp-data       20      1762 192.168.0.20
8    1/23/1998   16:56:22  0:00:00 ftp-data       20      1767 192.168.0.20
9    1/23/1998   16:56:24  0:00:02 ftp-data       20      1768 192.168.0.20
10   1/23/1998   16:56:25  0:01:01    telnet     1769        23 192.168.1.30
11   1/23/1998   16:56:27  0:00:00 ftp-data       20      1770 192.168.0.20
12   1/23/1998   16:56:36  0:00:03    finger     1772        79 192.168.1.30
13   1/23/1998   16:56:42  0:00:03      smtp     1778        25 192.168.1.30
14   1/23/1998   16:56:43  0:00:03      smtp     1783        25 192.168.1.30
15   1/23/1998   16:56:45  0:00:00      http     1784        80 192.168.1.30
16   1/23/1998   16:56:49  0:00:14       ftp    43504        21 192.168.0.40
17   1/23/1998   16:56:56  0:00:00 ftp-data       20     43505 192.168.1.30
18   1/23/1998   16:56:57  0:00:00 ftp-data       20     43506 192.168.1.30
19   1/23/1998   16:56:59  0:00:00 ftp-data       20     43508 192.168.1.30
20   1/23/1998   16:57:00  0:00:00 ftp-data       20     43509 192.168.1.30
21   1/23/1998   16:57:02  0:00:00 ftp-data       20     43510 192.168.1.30
```

Create table from "Server" column

tblServer <- table(data$Server)
tblServer

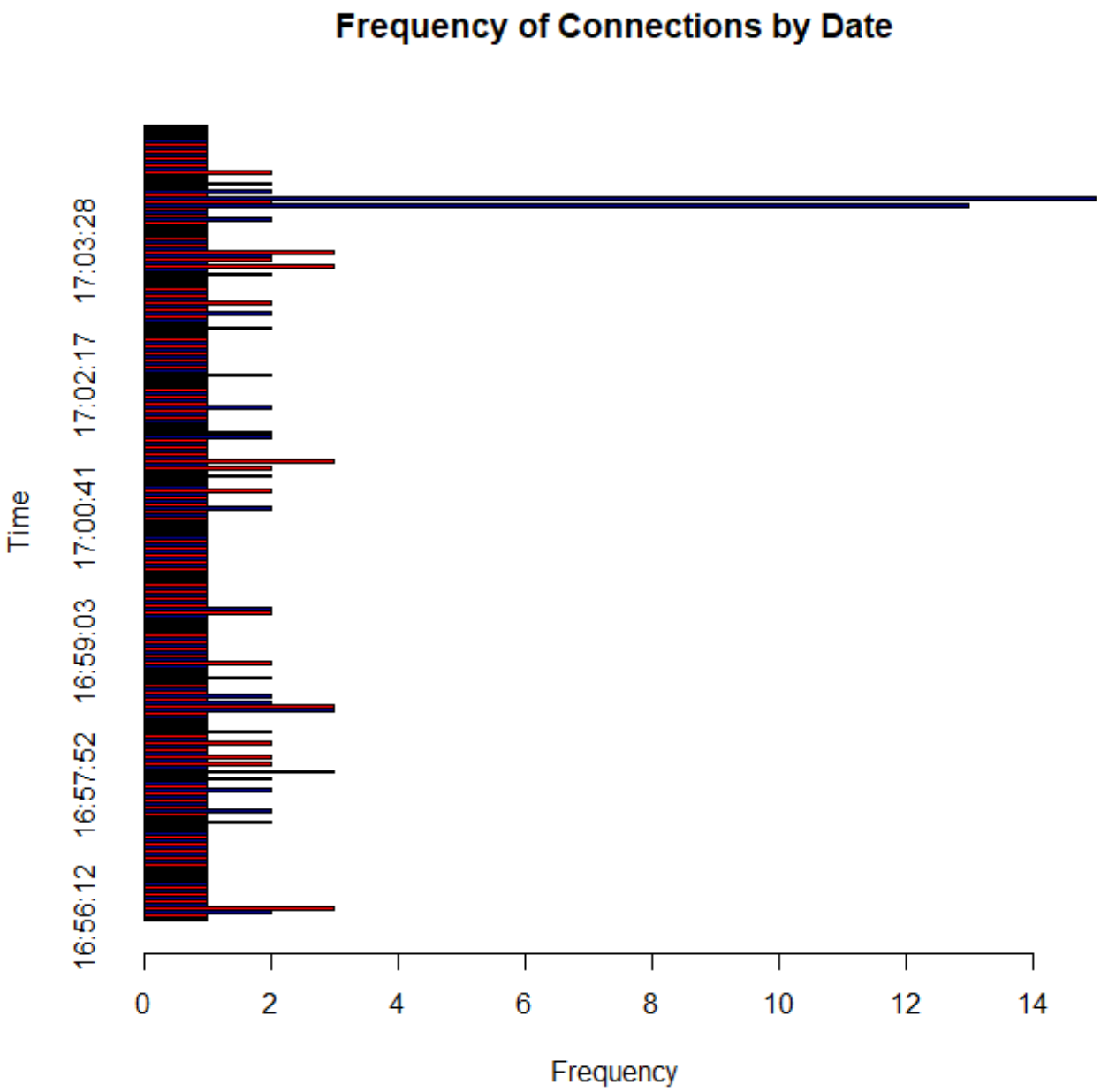
```
> tblServer <- table(data$Server)
> tblServer

    auth     exec   finger      ftp ftp-data     http
       5        2       16       14       44      157
 unknown x-server
       2        2
```

barplot(tblServer, main="Connections by Service", xlab="Services", ylab="Number of Connections", col=c("darkblue","red"))



barplot(startTime, main="Frequency of Connections by Date", xlab="Frequency", ylab="Date", col=c("darkblue","red"), hori=TRUE)

Nothing here would stand out, but this would be meaningful because we can establish baselines with this kind of data. If one of the services spiked beyond this baseline it would be a red flag for us.



This chart proves to be interesting because of the obvious spike that hits at 17:03. Lets focus on that time and see what happened.

I'm going to check out my startTime variable created earlier and focus on 17:03.

```
> startTime

16:56:12 16:56:15 16:56:17 16:56:19 16:56:22 16:56:24 16:56:25 16:56:27 16:56:36 16:56:42 16:56:43 16:56:45 16:56:49 16:56:56 16:56:57 16:56:59 16:57:00 16:57:02
       1        1        2        3        1        1        1        1        1        1        1        1        1        1        1        1        1        1
16:57:13 16:57:15 16:57:16 16:57:19 16:57:20 16:57:22 16:57:23 16:57:24 16:57:26 16:57:27 16:57:31 16:57:34 16:57:37 16:57:40 16:57:41 16:57:44 16:57:45 16:57:47
       1        1        1        1        1        1        1        1        1        2        1        1        2        1        1        1        1        1
16:57:48 16:57:52 16:57:53 16:57:55 16:57:57 16:57:59 16:58:02 16:58:03 16:58:04 16:58:06 16:58:07 16:58:10 16:58:11 16:58:13 16:58:16 16:58:18 16:58:20 16:58:21
       2        1        1        2        1        3        1        2        1        2        1        1        1        2        1        1        2        1
16:58:22 16:58:23 16:58:24 16:58:27 16:58:28 16:58:31 16:58:34 16:58:36 16:58:38 16:58:41 16:58:44 16:58:45 16:58:47 16:58:48 16:58:51 16:58:52 16:58:54 16:58:57
       1        1        1        1        3        3        2        1        2        1        1        1        1        2        1        1        1        2
16:59:00 16:59:01 16:59:03 16:59:06 16:59:09 16:59:12 16:59:15 16:59:18 16:59:21 16:59:23 16:59:24 16:59:26 16:59:29 16:59:33 16:59:36 16:59:42 16:59:45 16:59:47
       1        1        1        1        1        1        1        1        1        1        1        1        1        2        2        1        1        1
16:59:53 16:59:57 17:00:01 17:00:02 17:00:03 17:00:04 17:00:05 17:00:07 17:00:09 17:00:10 17:00:12 17:00:16 17:00:18 17:00:20 17:00:21 17:00:22 17:00:25 17:00:28
       1        1        1        1        1        1        1        1        1        1        1        1        1        1        1        1        1        1
17:00:32 17:00:35 17:00:39 17:00:41 17:00:43 17:00:45 17:00:46 17:00:49 17:00:52 17:00:56 17:01:00 17:01:03 17:01:06 17:01:14 17:01:24 17:01:25 17:01:28 17:01:31
       1        1        1        1        1        1        2        1        1        1        1        2        1        1        1        2        1        2
17:01:33 17:01:34 17:01:35 17:01:36 17:01:37 17:01:41 17:01:42 17:01:44 17:01:47 17:01:50 17:01:51 17:01:52 17:01:54 17:01:55 17:01:56 17:01:58 17:02:01 17:02:02
       1        3        1        1        1        1        1        1        2        2        1        1        1        1        1        1        2        1
```

I need to narrow this down by slicing it. Some quick math and I can guess I can simply slice it around record 175. Some quick tinkering and grabbing 166 through 204 grabs all of the data for 17:03. This is done with startTime[166:204] in R. A quick glance at this shows we can narrow the data down further.

```
17:03:20 17:03:21 17:03:23
       1        3        1
17:03:52 17:03:53 17:03:54
      13        2       15
```

17:03:52-54 contains 30 “connections”. We can go back to the original table we created (data) and look at those records.

243	1/23/1998	17:03:52	0:00:05	telnet	2020	23	192.168.1.30	192.168.0.20	1 port-scan
244	1/23/1998	17:03:52	0:00:00	ssh	2021	22	192.168.1.30	192.168.0.20	1 port-scan
245	1/23/1998	17:03:52	0:00:05	ftp	2022	21	192.168.1.30	192.168.0.20	1 port-scan
246	1/23/1998	17:03:52	0:00:04	finger	2023	79	192.168.1.30	192.168.0.20	1 port-scan
247	1/23/1998	17:03:52	0:00:00	http	2024	80	192.168.1.30	192.168.0.20	1 port-scan
248	1/23/1998	17:03:52	0:00:00	sunrpc	2025	111	192.168.1.30	192.168.0.20	1 port-scan
249	1/23/1998	17:03:52	0:00:00	pop-3	2026	110	192.168.1.30	192.168.0.20	1 port-scan
250	1/23/1998	17:03:52	0:00:00	pop-2	2028	109	192.168.1.30	192.168.0.20	1 port-scan
251	1/23/1998	17:03:52	0:00:00	lpr	2029	515	192.168.1.30	192.168.0.20	1 port-scan
252	1/23/1998	17:03:52	0:00:06	rsh	2030	514	192.168.1.30	192.168.0.20	1 port-scan
253	1/23/1998	17:03:52	0:00:06	rlogin	2031	513	192.168.1.30	192.168.0.20	1 port-scan
254	1/23/1998	17:03:52	0:00:06	exec	2032	512	192.168.1.30	192.168.0.20	1 port-scan
255	1/23/1998	17:03:52	0:00:00	nfsd	2033	2049	192.168.1.30	192.168.0.20	1 port-scan
256	1/23/1998	17:03:53	0:00:00	unknown	2034	3000	192.168.1.30	192.168.0.20	1 port-scan
257	1/23/1998	17:03:53	0:00:00	x-server	2035	6000	192.168.1.30	192.168.0.20	1 port-scan
258	1/23/1998	17:03:54	0:00:00	telnet	2037	23	192.168.1.30	192.168.0.40	1 port-scan
259	1/23/1998	17:03:54	0:00:00	ssh	2038	22	192.168.1.30	192.168.0.40	1 port-scan
260	1/23/1998	17:03:54	0:00:00	ftp	2039	21	192.168.1.30	192.168.0.40	1 port-scan
261	1/23/1998	17:03:54	0:00:00	finger	2040	79	192.168.1.30	192.168.0.40	1 port-scan
262	1/23/1998	17:03:54	0:00:00	http	2041	80	192.168.1.30	192.168.0.40	1 port-scan
263	1/23/1998	17:03:54	0:00:00	sunrpc	2042	111	192.168.1.30	192.168.0.40	1 port-scan
264	1/23/1998	17:03:54	0:00:00	pop-3	2043	110	192.168.1.30	192.168.0.40	1 port-scan
265	1/23/1998	17:03:54	0:00:00	pop-2	2044	109	192.168.1.30	192.168.0.40	1 port-scan
266	1/23/1998	17:03:54	0:00:00	lpr	2045	515	192.168.1.30	192.168.0.40	1 port-scan
267	1/23/1998	17:03:54	0:00:00	rsh	2046	514	192.168.1.30	192.168.0.40	1 port-scan
268	1/23/1998	17:03:54	0:00:00	rlogin	2047	513	192.168.1.30	192.168.0.40	1 port-scan
269	1/23/1998	17:03:54	0:00:00	exec	2048	512	192.168.1.30	192.168.0.40	1 port-scan
270	1/23/1998	17:03:54	0:00:00	nfsd	2052	2049	192.168.1.30	192.168.0.40	1 port-scan
271	1/23/1998	17:03:54	0:00:00	unknown	1029	3000	192.168.1.30	192.168.0.40	1 port-scan
272	1/23/1998	17:03:54	0:00:00	x-server	1030	6000	192.168.1.30	192.168.0.40	1 port-scan

As you can see IP address 192.168.1.30 was doing port scans on multiple targets with various services. Typically port scans are used to determine which services on target machines are active. This is a cause for alarm in this scenario and should be investigated further.

I would like to single out the 192.168.1.30 source IP. I'm going to create a data frame with the source IP column and attack name to see what what else this IP has been up to. This requires me to create variables (name and source) from their respective columns in the table. This is done with:

name <- data$Name
source <-data$source.IP

I then create the data frame using: dataDF <- data.Frame(name, source). I can do a summary of this data frame.

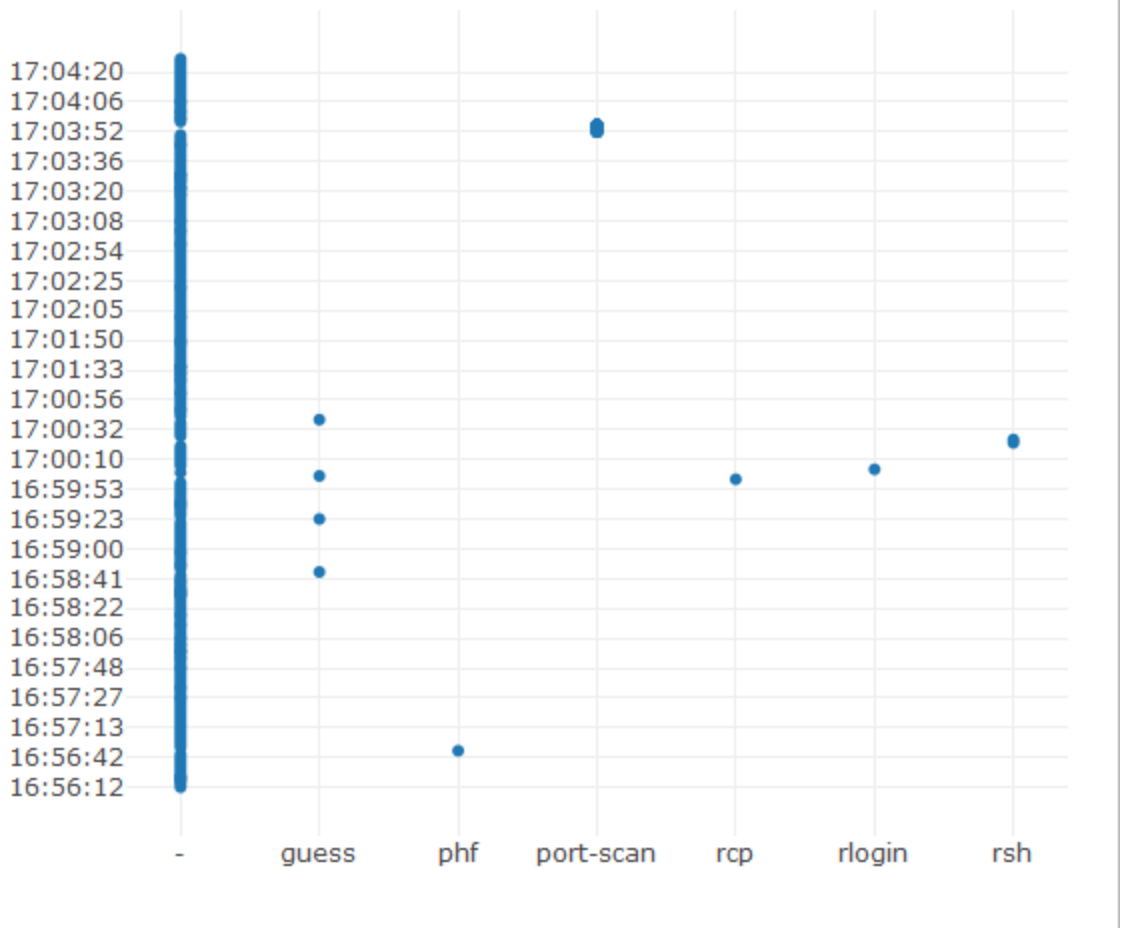
```
> summary(dataDF)
        name                source
 -         :256   192.168.0.20: 24
 guess     :  4   192.168.0.40: 51
 phf       :  1   192.168.1.30:220
 port-scan: 30
```

I can see the source IP Is listed 220 times and there's been other attacks. Some quick filtering gives met the following:

phf	**192.168.1.30**
guess	**192.168.1.30**
guess	**192.168.1.30**
rcp	**192.168.1.30**
guess	**192.168.1.30**
rlogin	**192.168.1.30**
rsh	**192.168.1.30**
guess	**192.168.1.30**
port-scan	**192.168.1.30**

*Note that there were 29 other port-scans I removed to save space.

Screen shot of HTML file:



Code:

```
> name <- data$Name
> start_time <- data$StartTime
> plot_ly(x=name,y=starttime, type='scatter',mode='markers')
Error in plot_ly(x = name, y = starttime, type = "scatter", mode = "markers")
:
  object 'starttime' not found
> plot_ly(x=name,y=start_time, type='scatter',mode='markers')
```