Virtual SCADA Network: Exploratory Data Analysis

Lisa MALIPHOL

Introduction

As a first step in the SCAD@COPS project presented in its introduction [1], the initial phase of exploratory data analysis is conducted in order to be able to better understand the data. In addition to the traditional methods of using descriptive statistics to explain the data, the various graphical and visual manners of representing the data are presented.

The paper is an analysis and statistical study of network traffic captured over a virtual SCADA network with simulated attacks. The network traffic was captured using Wireshark, and R was the language used to carry out the statistical analysis. The organisation of this study is presented in the following sections-

The paper is organized as follows:

- Tools used during this process
- Data source
- Exploratory Data Analysis
  - Statistical definitions
  - Visual representations defined
  - Analysis

Tools

A great deal of work is typically involved in preparing the raw data for analysis. Depending on the initial state of the data, various pre-processing and transformations may be required. The following tools were used in the exploratory phase of data analysis in order to capture, transform and analyze the data. The commands and scripts used in this process are found in the Appendix.

Wireshark[1] - Network Traffic Analysis Tool

Developed in 1997 by Gerald Combs originally named Ethereal, Wireshark is now an Open Source GNU project. It is a network packet analyzer, or “packet sniffer”, that captures and displays network packets.

Captured network packets are saved in the pcap file format and can be dissected and parsed by Wireshark in order to analyze its contents. An important aspect of Wireshark is that of its passive/monitoring nature and so does not send, manipulate, or modify the data passing over the network.

An initial packet capture file was created over simulated network traffic using Wireshark. Using its export facilities, various files were created for further analysis, with information such as TCP endpoints, conversations, etc.

[1] https://www.wireshark.org/docs/wsug_html_chunked

TShark[2]

Another tool from the Wireshark suite is the command-line tool similar to tcpdump is tshark, a network protocol analyzer. In addition to capturing packet data over a live network, it is also capable of analyzing packets from an existing capture file. TShark was used to parse out various pertinent variables pertaining to the Modbus/TCP application protocol enclosed in the packet data.

sed

In order to further parse and transform the data, the UNIX utility tool sed, which supports the use of regular expressions, was also used.

R - Statistical Tool[3]

R is an Open Source programming language and environment used for statistical computing and graphics. Initially developed by John Chambers at Bell Labs as the S language in 1993, R was created as a freely available version under the GNU project by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand.

Maintained by the R Development Core Team and with an active and growing community, it provides various statistical and graphical creation capabilities available under most operating systems, and is extensible with numerous packages available.

Data Source

PCap[4] File

A packet capture file was created via Wireshark, which captured the network traffic simulated over a virtual SCADA network. This file also included injected random attacks over the network.

SCADA_20150429_.pcap	
File	
Length:	271279028 bytes
Format:	Wireshark/tcpdump/... - libcap
Encapsulation:	Ethernet
Packet size limit:	65536
Time	
First packet:	2015-04-29 12:51:40
Last packet:	2015-04-29 17:28:37
Elapsed:	04:36:56
Traffic	Captured
Packets	3566852
B/t first and last pkt	16616,418 sec
Avg. packets/sec	214,661
Avg. packet size	60,055 bytes
Bytes	214208732
Avg. bytes/sec	12891,390
Avg. Mit/sec	0,103

[2] https://www.wireshark.org/docs/man-pages/tshark.html

[3] http://www.r-project.org/

[4] http://www.winpcap.org/ntar/draft/PCAP-DumpFileFormat.html

Once the network traffic was captured and saved in a pcap file, Wireshark provides the capability to export the raw data into various comma delimited files in order to do further analysis. Exported files were created with TCP endpoints, TCP conversations, as well as the entire pcap file, each as a CSV file. (Appendix A)

Exploratory Data Analysis

Originally championed by John Tukey [2], Exploratory Data Analysis (EDA) is an initial approach to understanding a data set in order to get a “feel” for the data, to summarizing its essential characteristics and to studying patterns in the data. In addition to using quantitative techniques, it is supported predominantly by means of graphical representations.

Conducting EDA possibly gives further insight into the form and structure of the data set, in addition to extracting value from it, visualizing it, and just as importantly, in communicating it.

Following are some brief explanations of descriptive statistical terms, as well as the graphical representations used.

Statistical Definitions

Mean

The (arithmetic) mean is a measure of central tendency, which is a single value which represents an average of the sample or population. It is calculated by dividing all the observations by the number of observations.

Median

Another measure of central tendency is the median, however, in this case, the median is determined by first ordering the observations by magnitude. Then the median is taken as the value which falls in the middle, or the average of the two middle values in the case of an even number of observations. The median is better suited when there are observations, or outliers, that fall way outside the norm. These are extreme values that differ greatly from other values in the data set.

Variance

The variance is the expected value of the squared differences between the random variables and its mean that is always positive. It gives an indication of how far apart the values are from the mean and each other.

$$var[X] = E[(X - E[X])^2]$$

Standard Deviation

The standard deviation is a measure of dispersion, or how spread out a random variable is around its mean. It is calculated as the square root of the variance and is, unlike the variance, expressed in the same terms as the data.

$$std[X] = \sqrt(var[X])$$

Covariance

A measure of how closely two variables change, or vary together is the covariance. Random random variables whose covariance is 0 is said to be uncorrelated.

$$cov[X, Y] = E[(X - E[X])(Y - E[Y])]$$

Correlation

Correlation is the strength between the relationship of, or dependence between, two variables whose value is typically bounded between the values of -1 and 1, that is to say, that the value has been normalized. It describes the magnitude and the direction of the relationship. If the correlation is positive, their values increase together, and if it is negative, one value decreases as the other value increases.

$$corr[X, Y] = cov[X, Y]/(std[X]std[Y])$$

Visual Representations

Pie chart

A pie chart is a circular diagram representing numerical proportions as slices of the pie. Scatter plot A diagram showing a collection of points as depicted by the coordinates between (typically) two variables on a plane. One axis represents the independent variable, whereas the other represents the dependent variable.

Histogram

A graphical representation which shows the distribution of continuous numerical values is a histogram and can be representative of a probability distribution. A frequency histogram is a univariate graphical way to show frequency counts of a value depicted with bars of different heights.

Bar chart

Similar to a histogram, a bar chart shows the distribution of values of a given variable,however, the data is in categorized.

Boxplot

An effective and graphical method for visualizing outliers is the boxplot. It displays the data in terms of interquartiles, where outliers are depicted as individual points. (Boxplot image source)

TODO insert image

Heat Map

A heat map displays data in a matrix where the values are represented by a range of colors. Typically displayed in 2D, larger values are usually shown in darker colors and smaller values in lighter colors on a heat map. They can also be accompanied by a dendrogram, a tree diagram used to illustrate clusters.

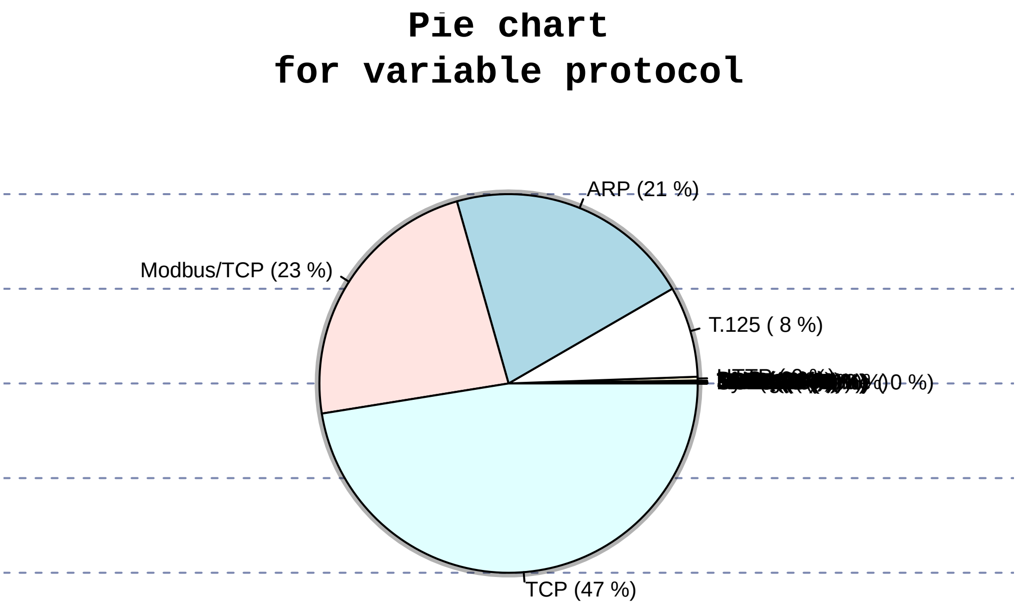
Network Graph

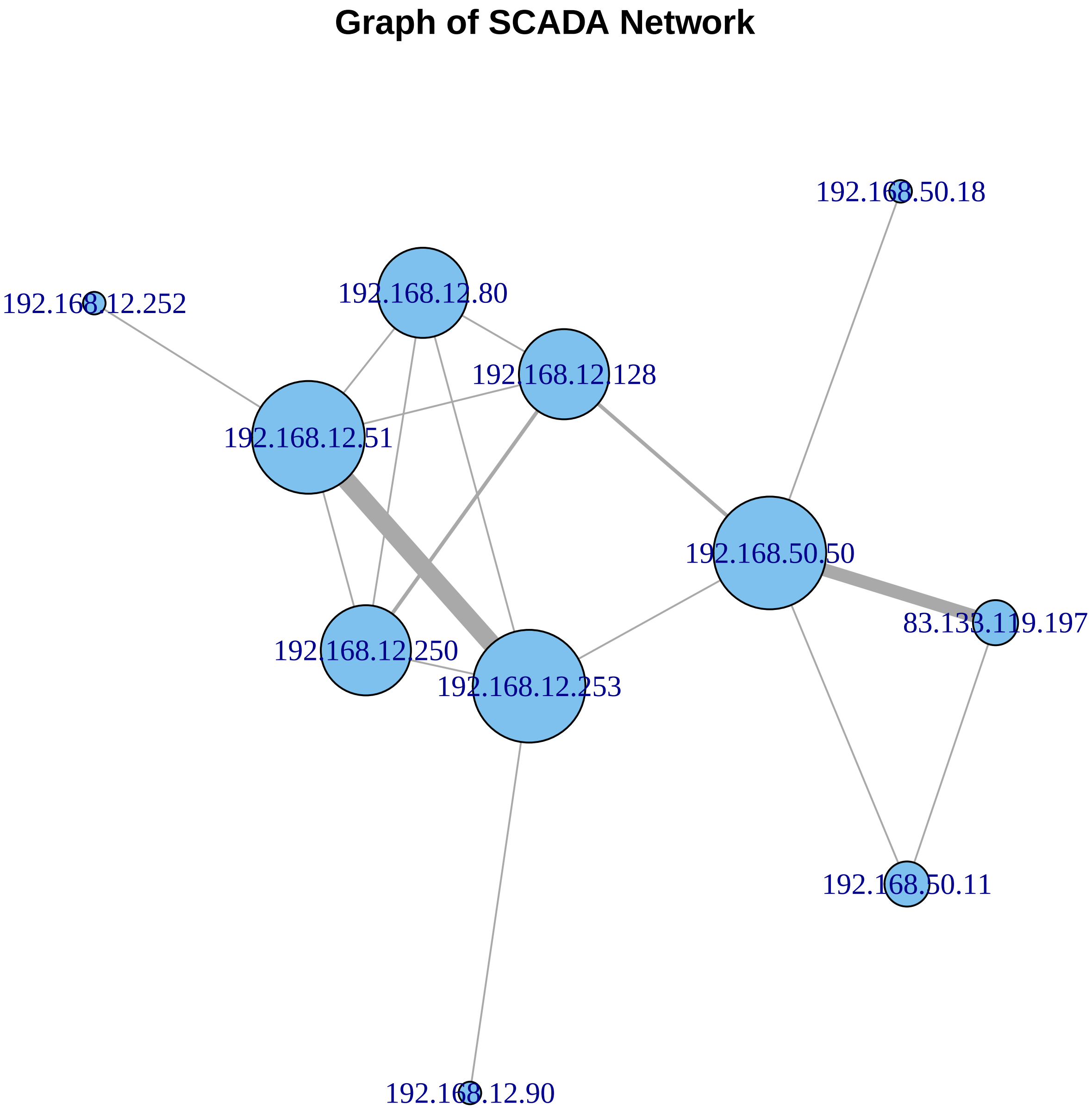
Used to model relations between objects, another mathematical structure is the graph, comprised of nodes, or vertices, and edges. Depending on the nature of the relationship, a graph may be either cyclic or acyclic, directed or undirected. Attributes of a node or edge may be reflected in the graph as well.

Analysis

Protocols

```
##          Protocol   Count
##  1:          TCP 1692588
##  2:   Modbus/TCP  825521
##  3:          ARP  751226
##  4:        T.125  277283
##  5:         HTTP   11275
##  6:          DNS    2525
##  7:          SMB    1007
##  8:          UDP     861
##  9:         IMAP     849
## 10:        TLSv1     575
## 11:         SMTP     533
## 12:         ICMP     526
## 13:         NBNS     491
## 14:       PN-DCP     364
## 15:       DHCPv6     273
## 16:       Syslog     246
## 17:      BROWSER     181
## 18:         SSDP     168
## 19:        LLMNR     128
## 20:       LANMAN     108
## 21:         NBSS      80
## 22:         MDNS      28
## 23:       DCERPC      21
## 24: RELOAD Frame      14
## 25:       REMACT       6
## 26:       SRVSVC       6
## 27:          IMF       5
## 28:         TPKT       4
##          Protocol   Count
```


Pie chart
for variable protocol
ARP (21 %)
Modbus/TCP (23 %)
T.125 (8 %)
TCP (47 %)



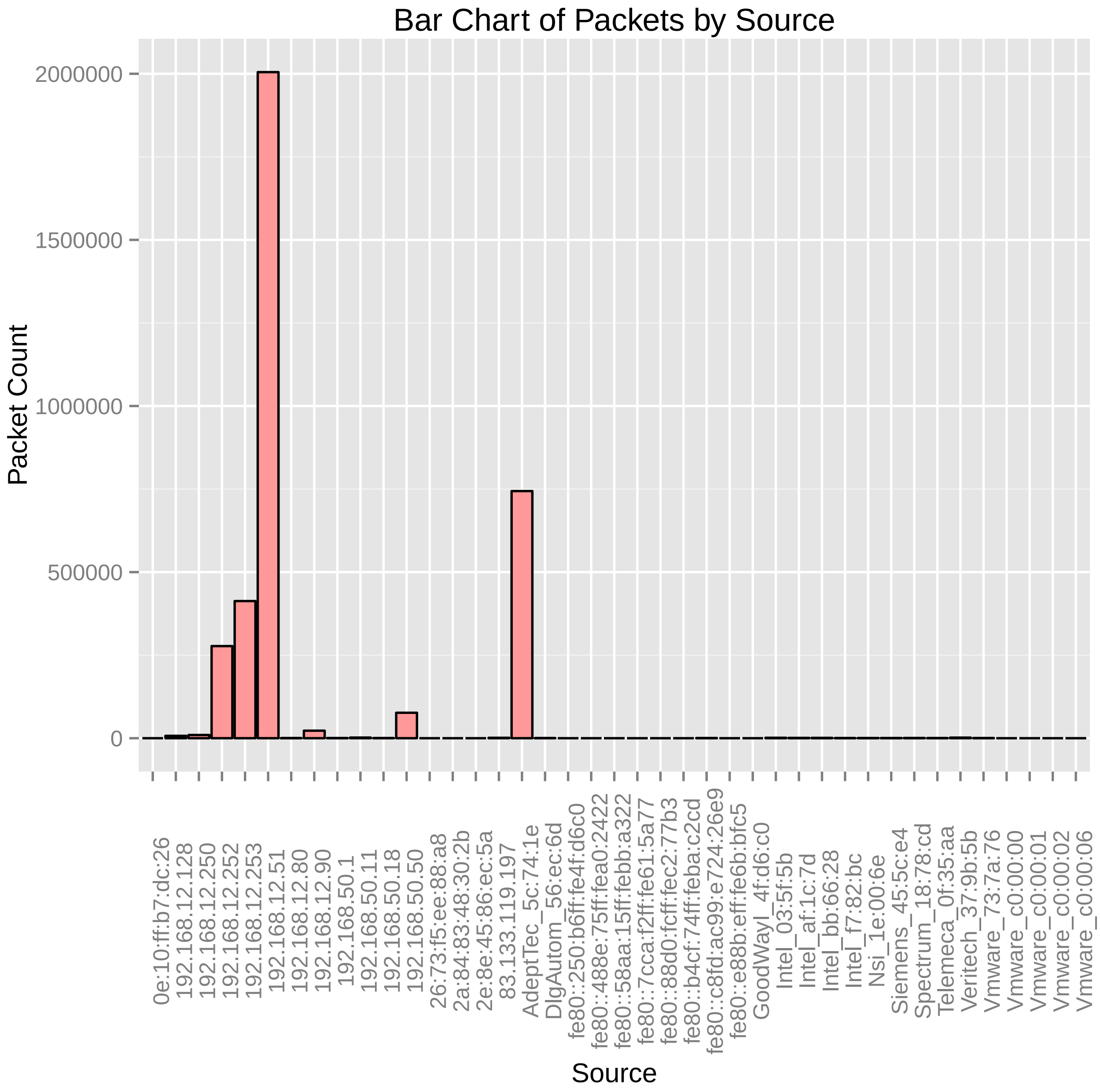
In the network graph shown above, the size of the node is according to its degree of centrality, that is, the number of adjacent vertices. The thicker edges indicate a higher number of interactions between two nodes.

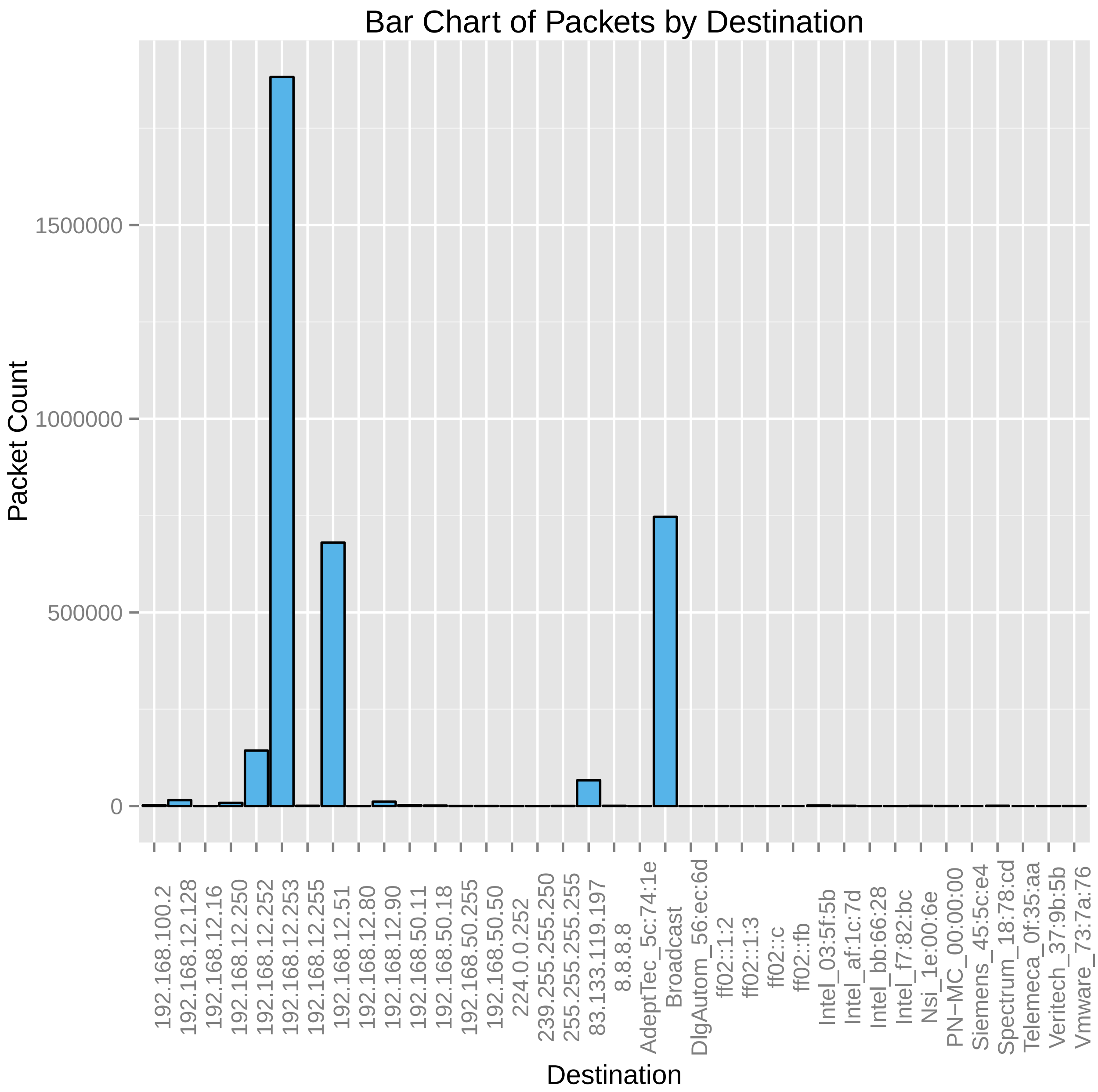
Node IP Addresses	
192.168.12.253	Schneider
192.168.12.51	
192.168.50.50	
83.133.119.197	
192.168.12.80	
192.168.12.250	
192.168.12.128	
192.168.50.11	
192.168.50.18	
192.168.12.90	
192.168.12.252	

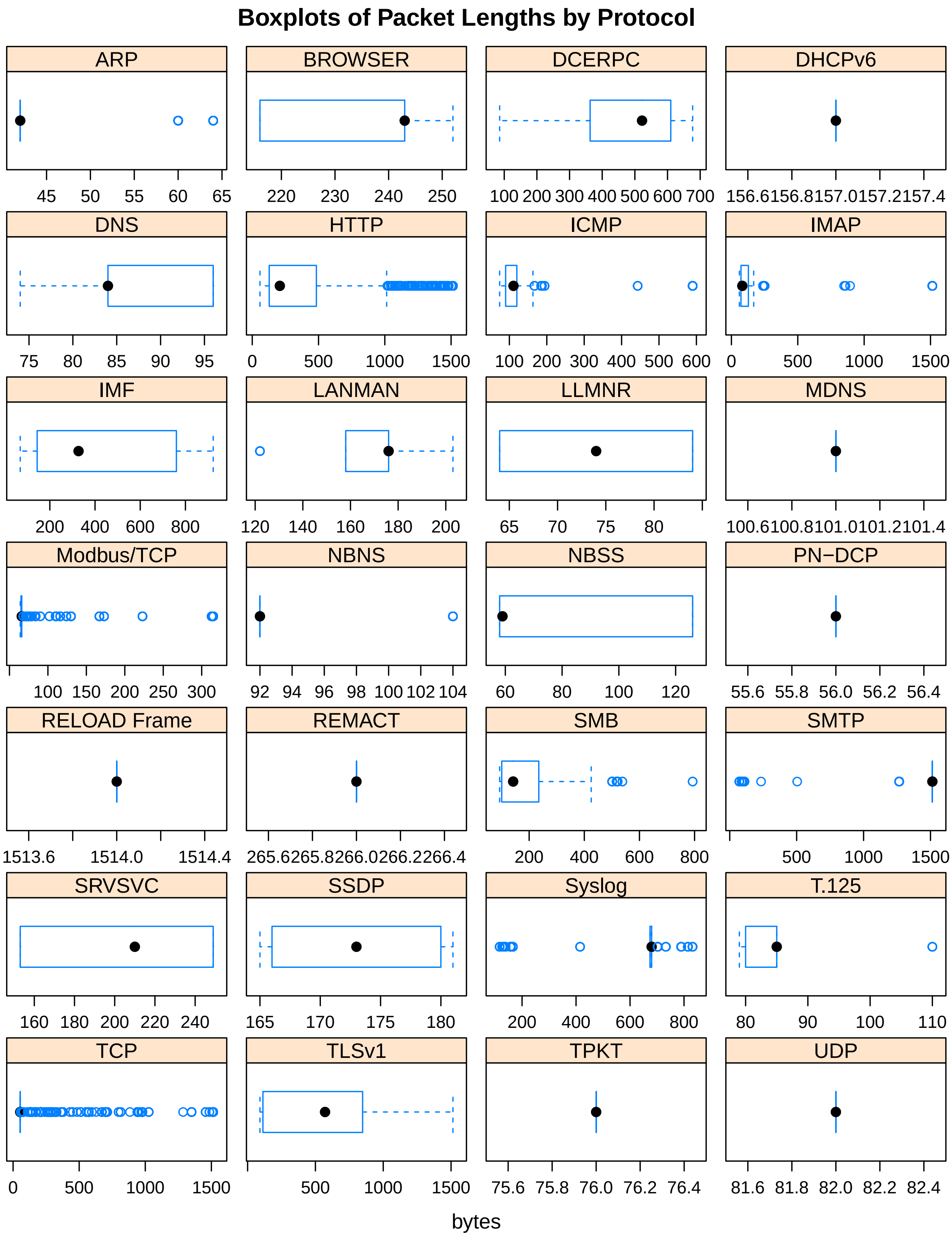
Packet Length Statistics

```
summary(scadaDT[.(Protocol="TCP"),.(Length)])
```

```
##       Length
##  Min.   :  54.00
##  1st Qu.:  54.00
##  Median :  54.00
##  Mean   :  58.09
##  3rd Qu.:  54.00
##  Max.   :1514.00
```



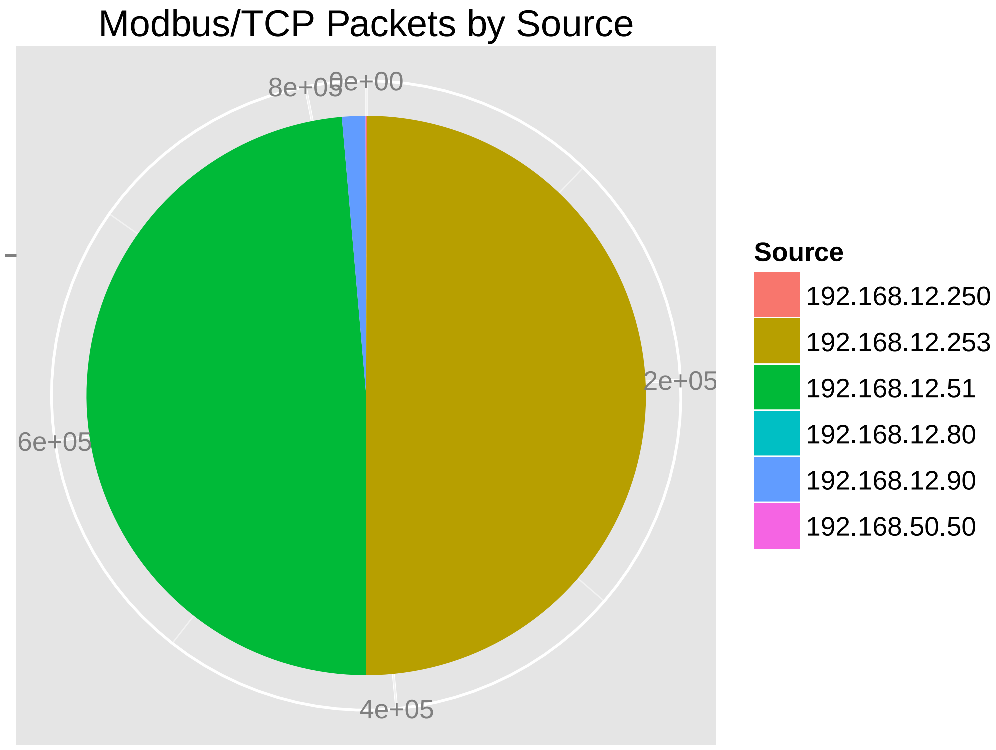

Bar Chart of Packets by Destination
Packet Count
1500000
1000000
500000
0
192.168.100.2
192.168.12.128
192.168.12.16
192.168.12.250
192.168.12.252
192.168.12.253
192.168.12.255
192.168.12.51
192.168.12.80
192.168.12.90
192.168.50.11
192.168.50.18
192.168.50.255
192.168.50.50
224.0.0.252
239.255.255.250
255.255.255.255
83.133.119.197
8.8.8.8
AdeptTec_5c:74:1e
Broadcast
DlgAutom_56:ec:6d
ff02::1:2
ff02::1:3
ff02::c
ff02::fb
Intel_03:5f:5b
Intel_af:1c:7d
Intel_bb:66:28
Intel_f7:82:bc
Nsi_1e:00:6e
PN−MC_00:00:00
Siemens_45:5c:e4
Spectrum_18:78:cd
Telemeca_0f:35:aa
Veritech_37:9b:5b
Vmware_73:7a:76
Destination

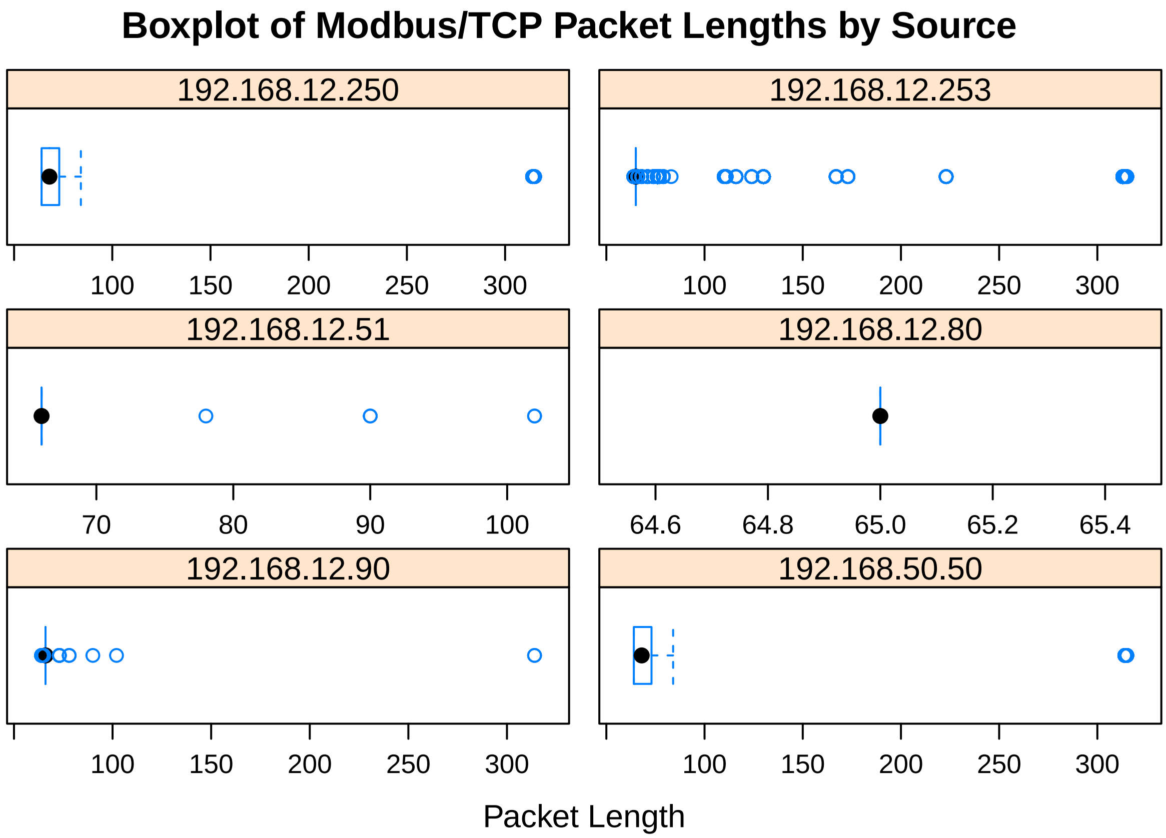

Boxplots of Packet Lengths by Protocol
ARP
45 50 55 60 65
BROWSER
220 230 240 250
DCERPC
100 200 300 400 500 600 700
DHCPv6
156.6156.8157.0157.2157.4
DNS
75 80 85 90 95
HTTP
0 500 1000 1500
ICMP
100 200 300 400 500 600
IMAP
0 500 1000 1500
IMF
200 400 600 800
LANMAN
120 140 160 180 200
LLMNR
65 70 75 80
MDNS
100.6100.8101.0101.2101.4
Modbus/TCP
100 150 200 250 300
NBNS
92 94 96 98 100 102 104
NBSS
60 80 100 120
PN−DCP
55.6 55.8 56.0 56.2 56.4
RELOAD Frame
1513.6 1514.0 1514.4
REMACT
265.6265.8266.0266.2266.4
SMB
200 400 600 800
SMTP
500 1000 1500
SRVSVC
160 180 200 220 240
SSDP
165 170 175 180
Syslog
200 400 600 800
T.125
80 90 100 110
TCP
0 500 1000 1500
TLSv1
500 1000 1500
TPKT
75.6 75.8 76.0 76.2 76.4
UDP
81.6 81.8 82.0 82.2 82.4
bytes

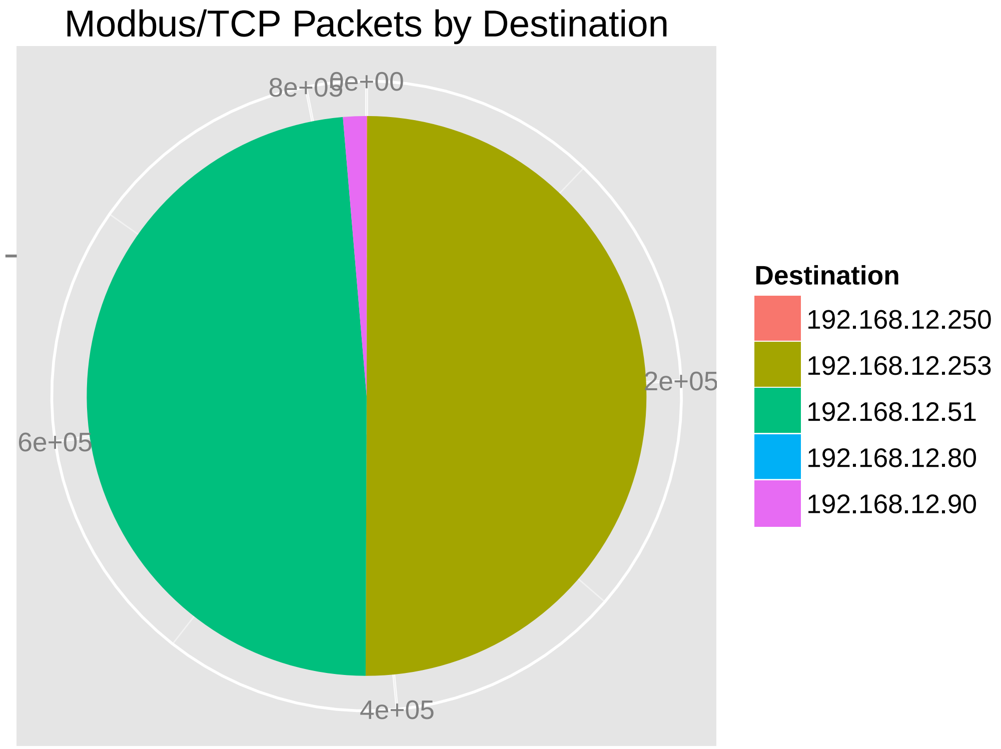
Modbus/TCP Statistics

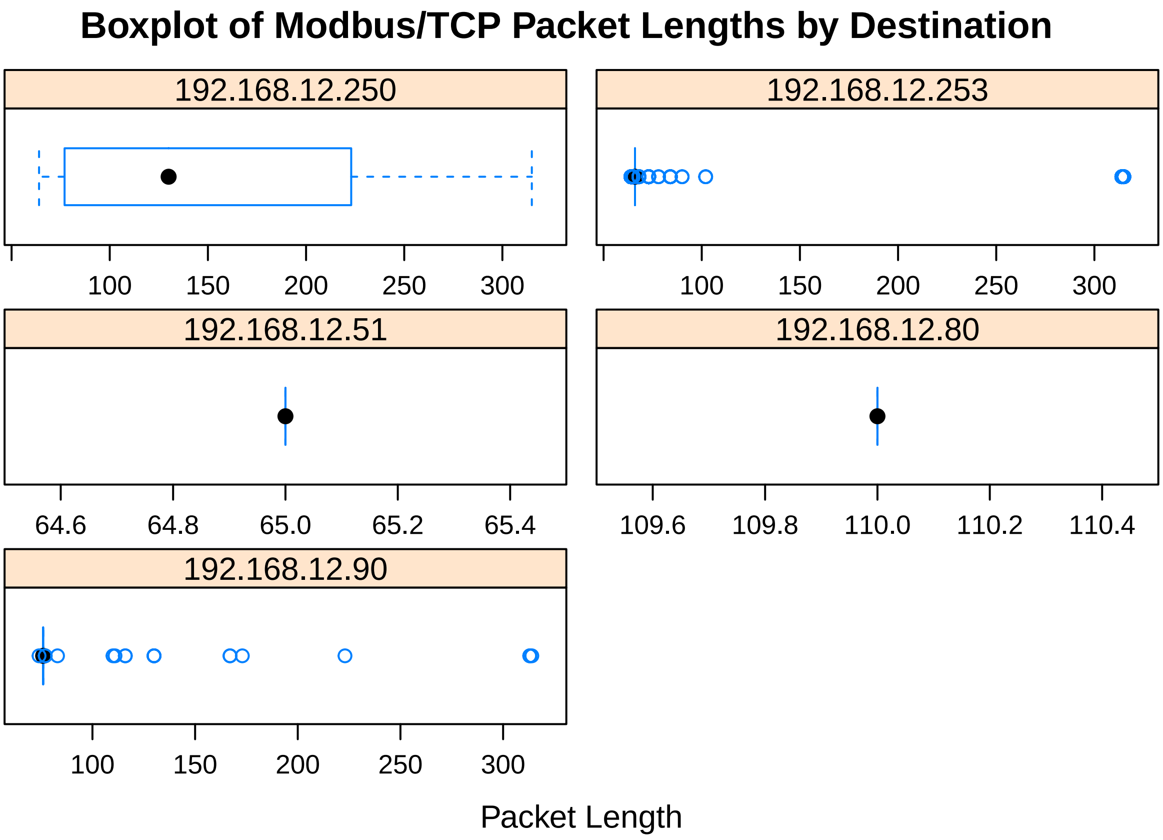
```
summary(scadaDT[.(Protocol="Modbus/TCP"),.(Length)])
```

```
##      Length
##  Min.   : 64.0
##  1st Qu.: 65.0
##  Median : 66.0
##  Mean   : 65.7
##  3rd Qu.: 66.0
##  Max.   :315.0
```



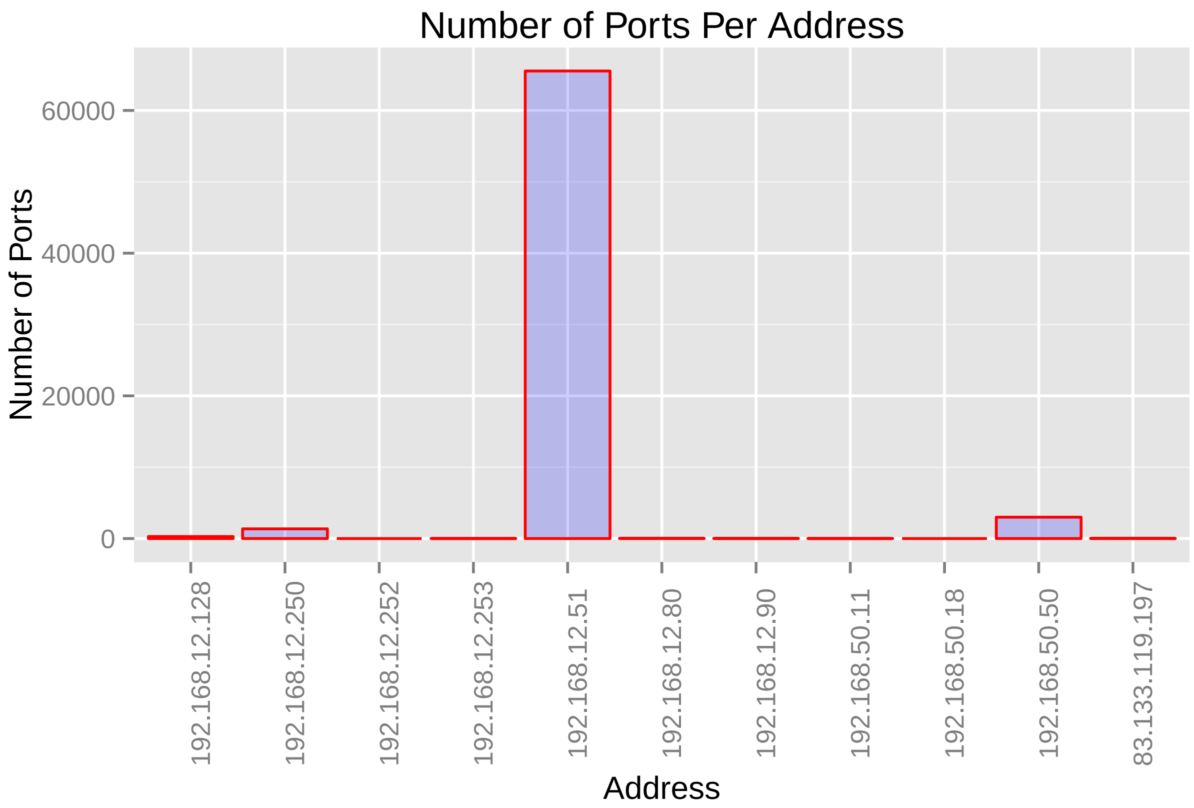

Boxplot of Modbus/TCP Packet Lengths by Source


Modbus/TCP Packets by Destination

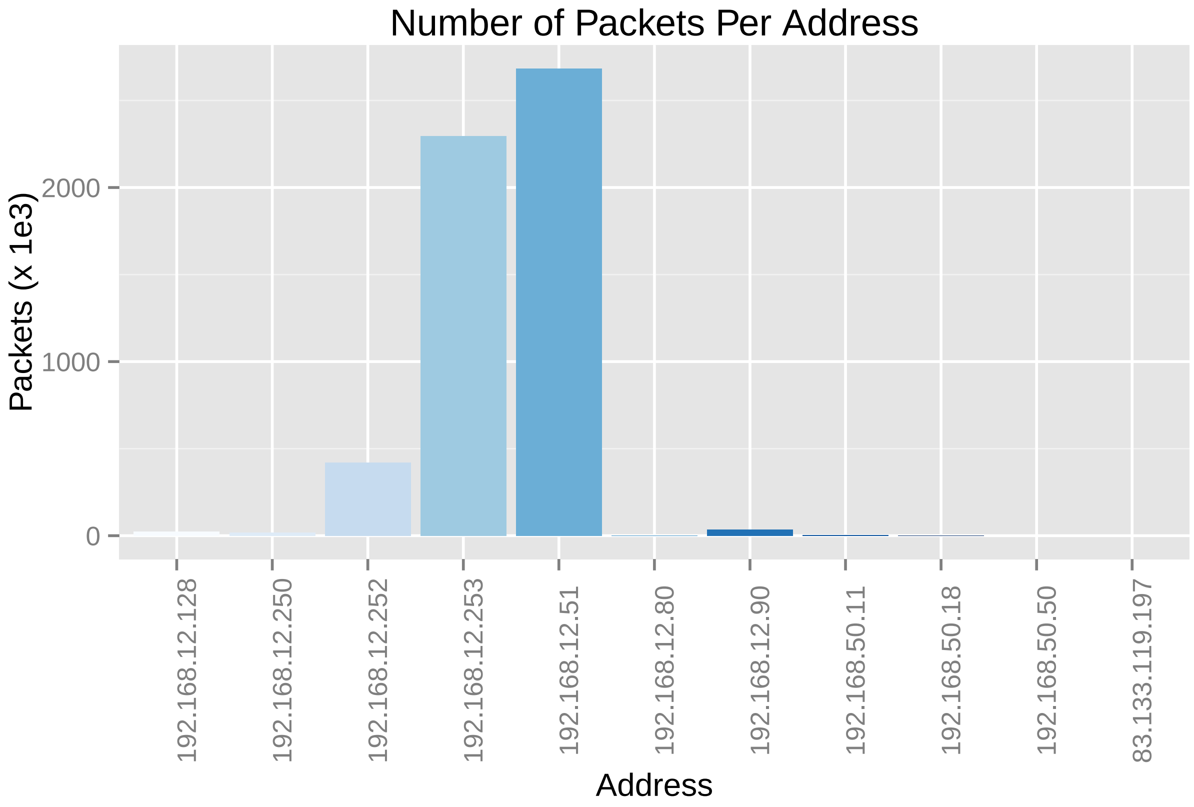


Endpoints

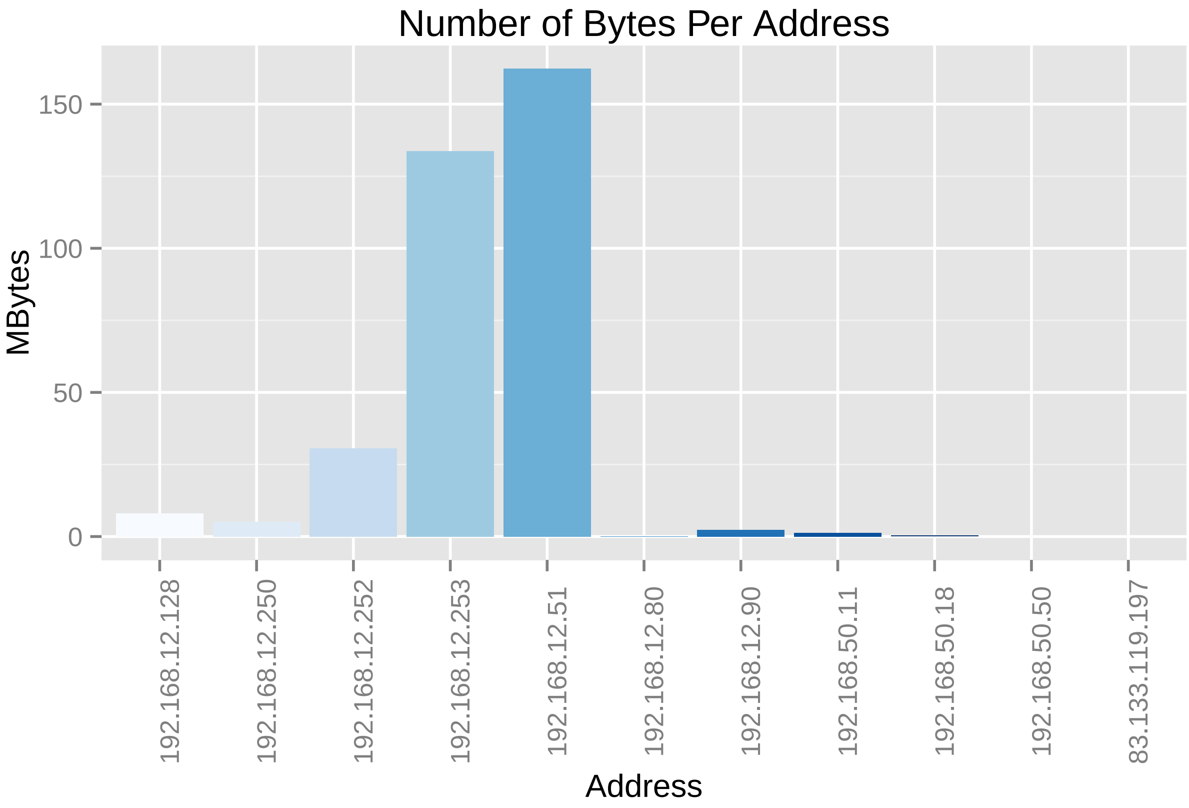
SCADA_Security_042915_TCP_Endpoints.csv



```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Blues is 9
## Returning the palette you asked for with that many colors
```



```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Blues is 9
## Returning the palette you asked for with that many colors
```



```r
cor(obs, use="complete.obs",method="spearman")
```

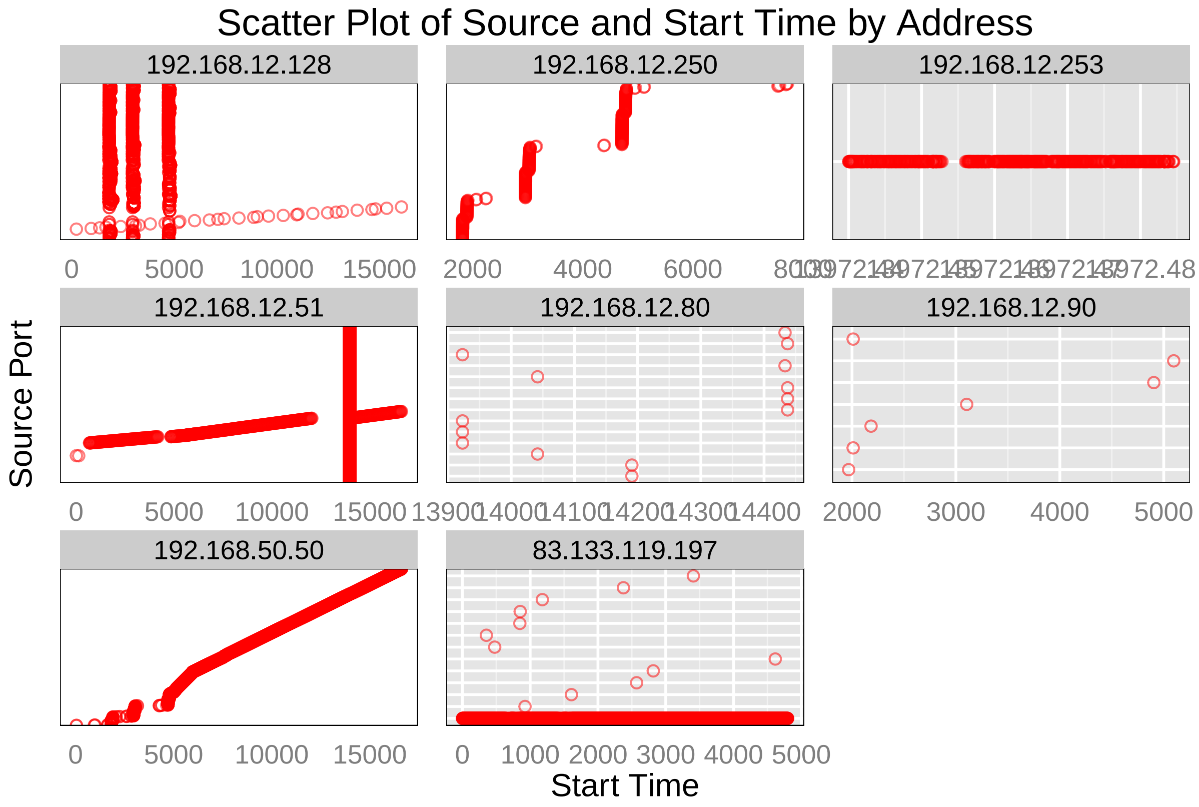
```
##                     Packets       Bytes Packets.A.B  Bytes.A.B Packets.A.B.1
## Packets          1.00000000  0.97894124   0.9788233  0.5040848    0.89660460
## Bytes            0.97894124  1.00000000   0.9165046  0.4720734    0.92313903
## Packets.A.B      0.97882329  0.91650460   1.0000000  0.5149616    0.83109826
## Bytes.A.B        0.50408481  0.47207338   0.5149616  1.0000000    0.42692965
## Packets.A.B.1    0.89660460  0.92313903   0.8310983  0.4269296    1.00000000
## Bytes.A.B.1      0.40776399  0.42066174   0.3770389 -0.5593192    0.47568586
## Duration        -0.04869179 -0.02865136  -0.0669847 -0.2816442    0.02599502
## bps.A.B          0.27557708  0.25036473   0.2899822  0.7759541    0.19682668
## bps.A.B.1        0.25341226  0.22904557   0.2673998 -0.3608199    0.19145965
##                 Bytes.A.B.1    Duration     bps.A.B  bps.A.B.1
## Packets           0.4077640 -0.04869179  0.2755771  0.2534123
## Bytes             0.4206617 -0.02865136  0.2503647  0.2290456
## Packets.A.B       0.3770389 -0.06698470  0.2899822  0.2673998
## Bytes.A.B        -0.5593192 -0.28164421  0.7759541 -0.3608199
## Packets.A.B.1     0.4756859  0.02599502  0.1968267  0.1914597
## Bytes.A.B.1       1.0000000  0.27201179 -0.5562958  0.5958265
## Duration          0.2720118  1.00000000 -0.7473304 -0.4840310
## bps.A.B          -0.5562958 -0.74733042  1.0000000  0.1014912
## bps.A.B.1         0.5958265 -0.48403095  0.1014912  1.0000000
```

```
cov(obs,method="spearman",use="complete.obs")
```

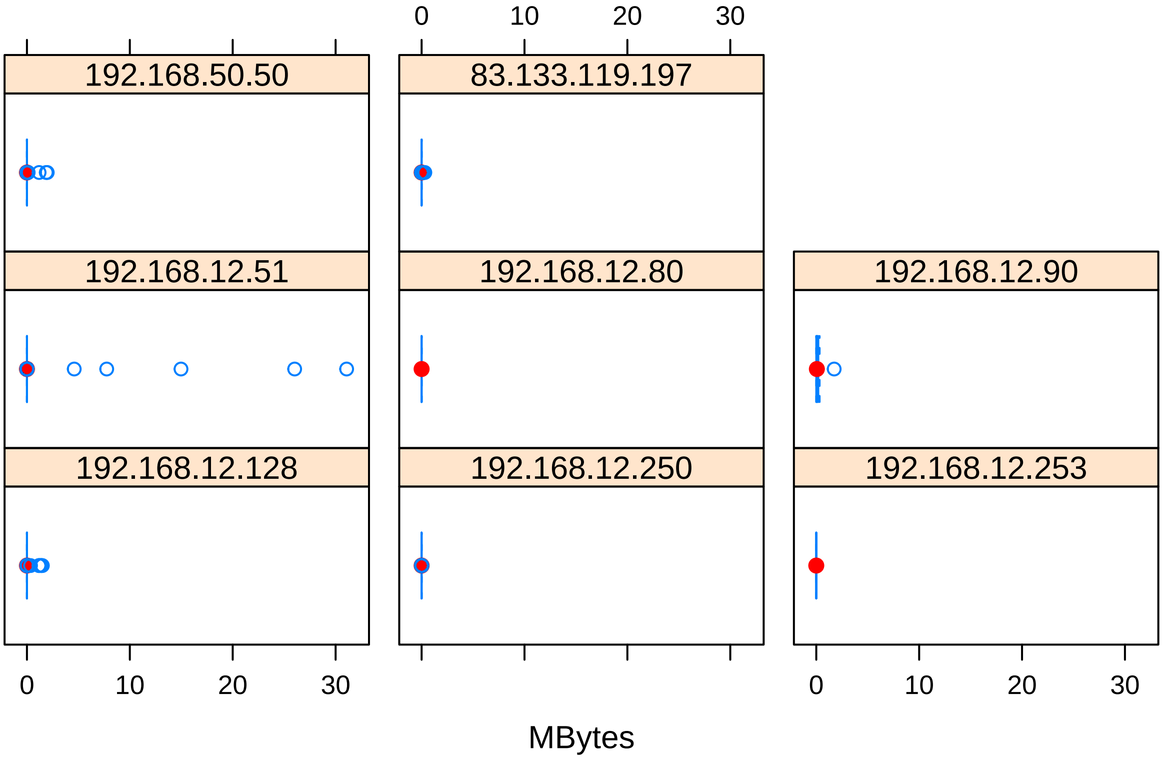
```
##                  Packets     Bytes Packets.A.B  Bytes.A.B Packets.A.B.1
## Packets        54907.736 54905.176   54898.479   54897.22     46449.061
## Bytes          54905.176 57290.127   52506.587   52514.52     48850.185
## Packets.A.B    54898.479 52506.587   57289.957   57285.41     43979.555
## Bytes.A.B      54897.218 52514.516   57285.412  216003.07     43867.763
## Packets.A.B.1  46449.061 48850.185   43979.555   43867.76     48878.562
## Bytes.A.B.1    44407.418 46795.361   41942.599 -120814.64     48877.541
## Duration       -6062.009 -3643.588   -8518.417  -69546.43      3053.468
## bps.A.B        34308.759 31838.911   36877.019  191606.90     23120.026
## bps.A.B.1      31549.286 29127.750   34005.206  -89097.51     22489.594
##                Bytes.A.B.1    Duration     bps.A.B   bps.A.B.1
## Packets           44407.42   -6062.009    34308.76    31549.29
## Bytes             46795.36   -3643.588    31838.91    29127.75
## Packets.A.B       41942.60   -8518.417    36877.02    34005.21
## Bytes.A.B       -120814.64  -69546.427   191606.90   -89097.51
## Packets.A.B.1     48877.54    3053.468    23120.03    22489.59
## Bytes.A.B.1      216003.00   67167.881 -137366.51   147127.84
## Duration          67167.88  282285.180 -210960.85 -136635.11
## bps.A.B         -137366.51 -210960.846   282286.63    28649.61
## bps.A.B.1        147127.84 -136635.114    28649.61   282286.62
```

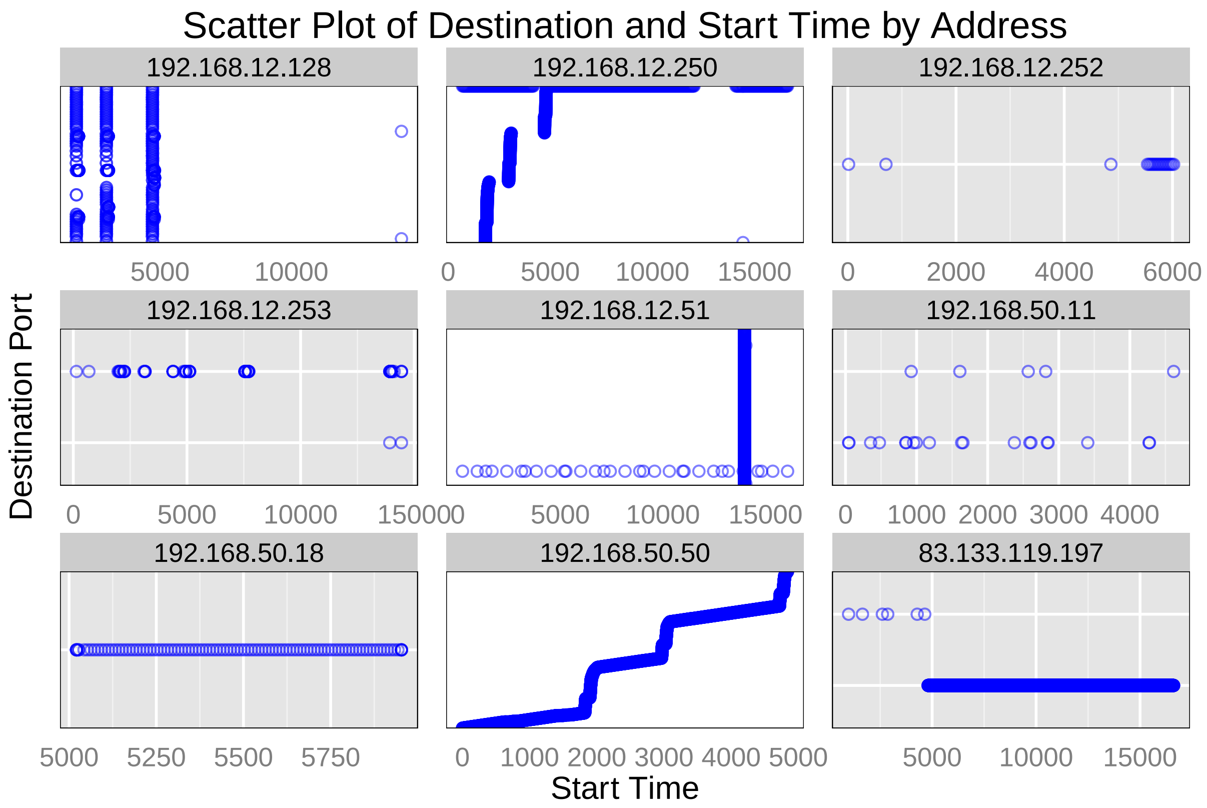
Conversations

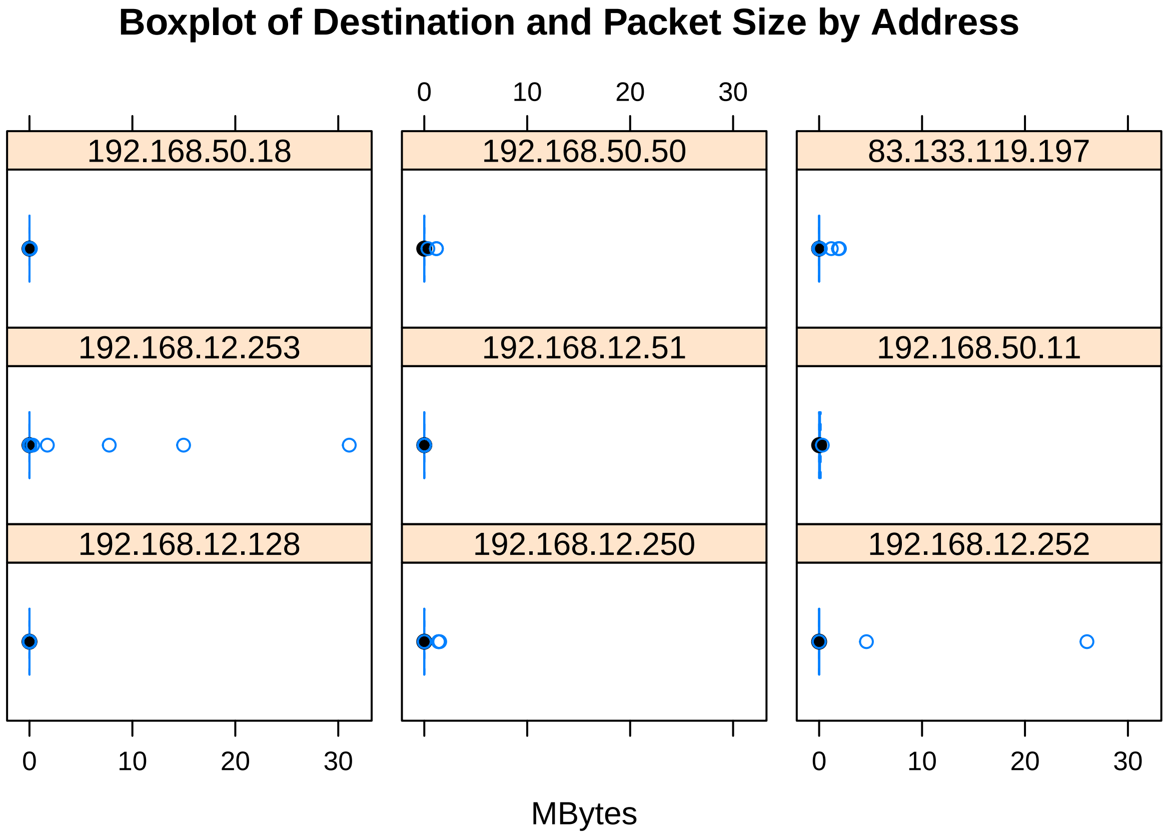
SCADA_Security_042915_TCP_Conversations.csv

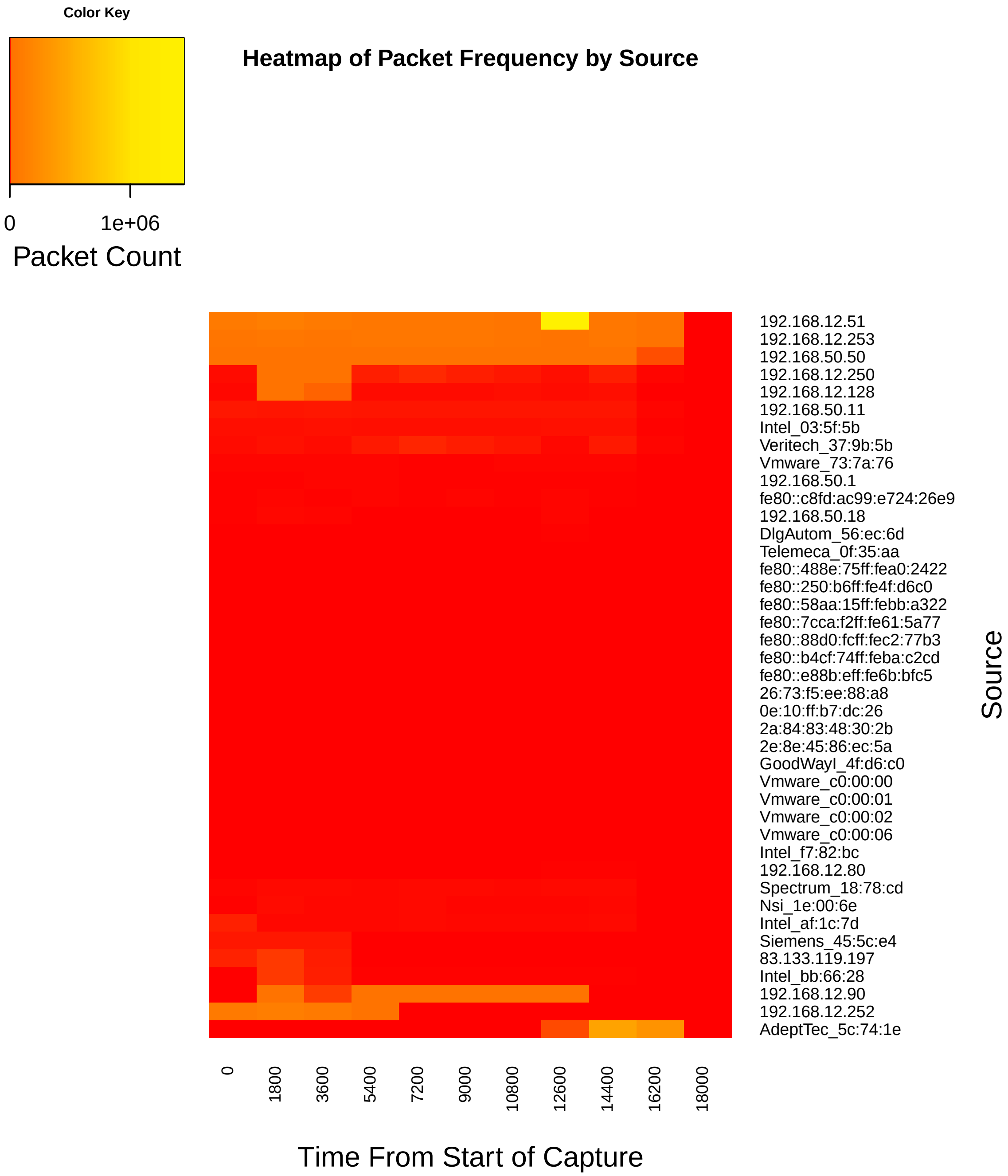


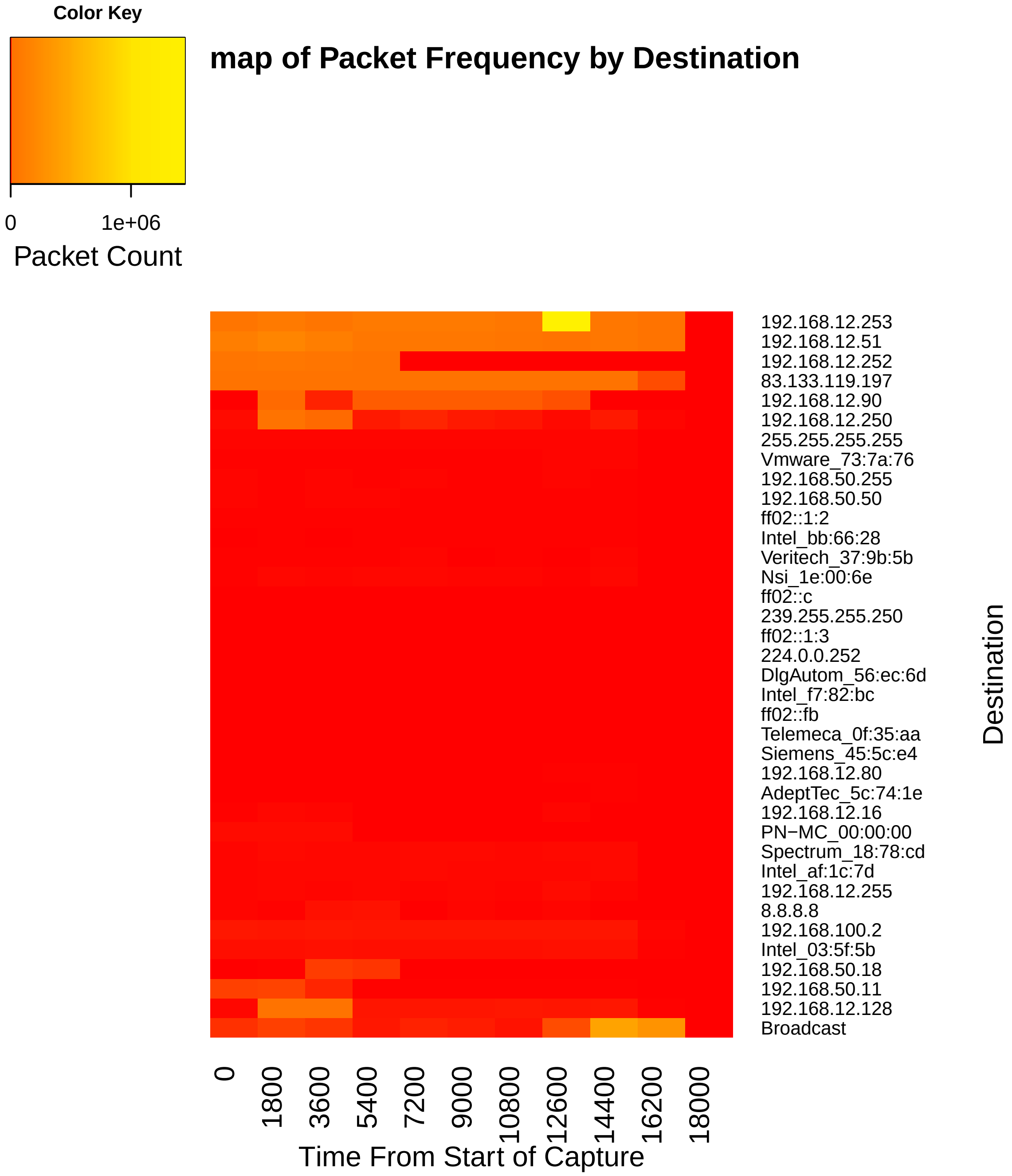
Boxplot of Source and Packet Size by Address

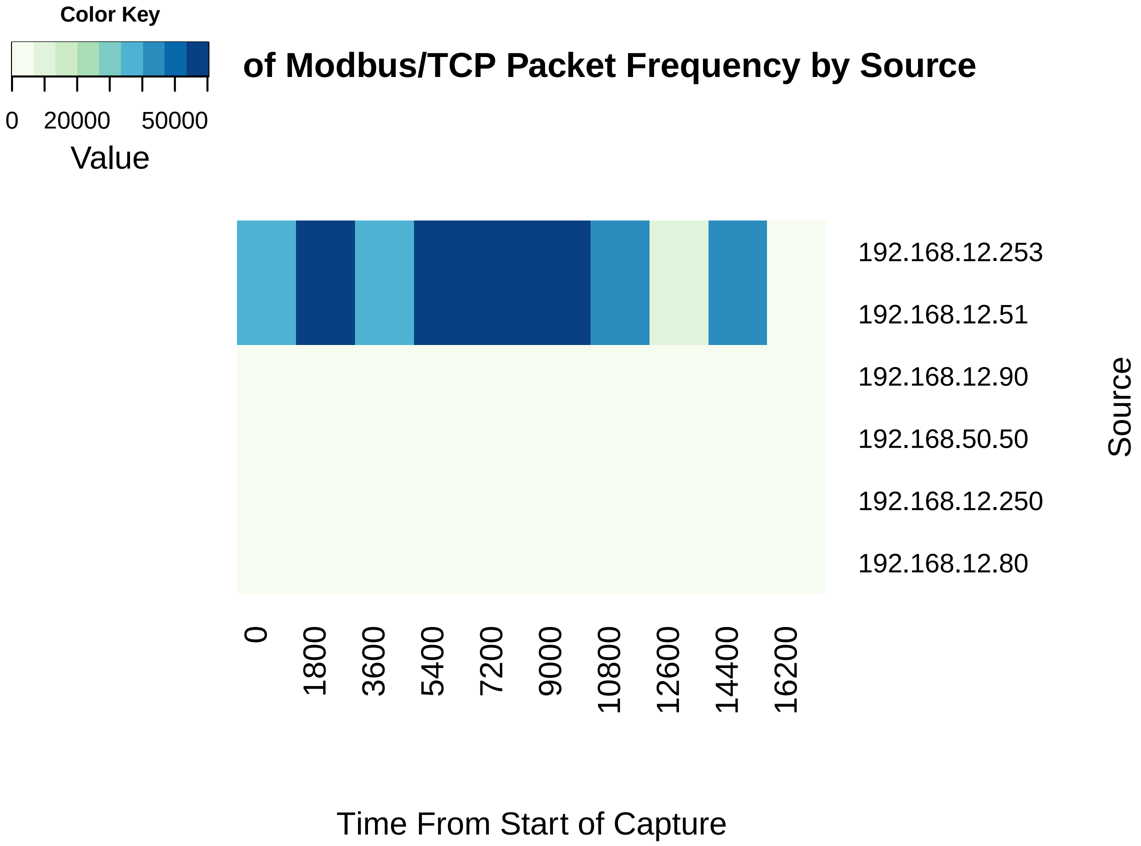


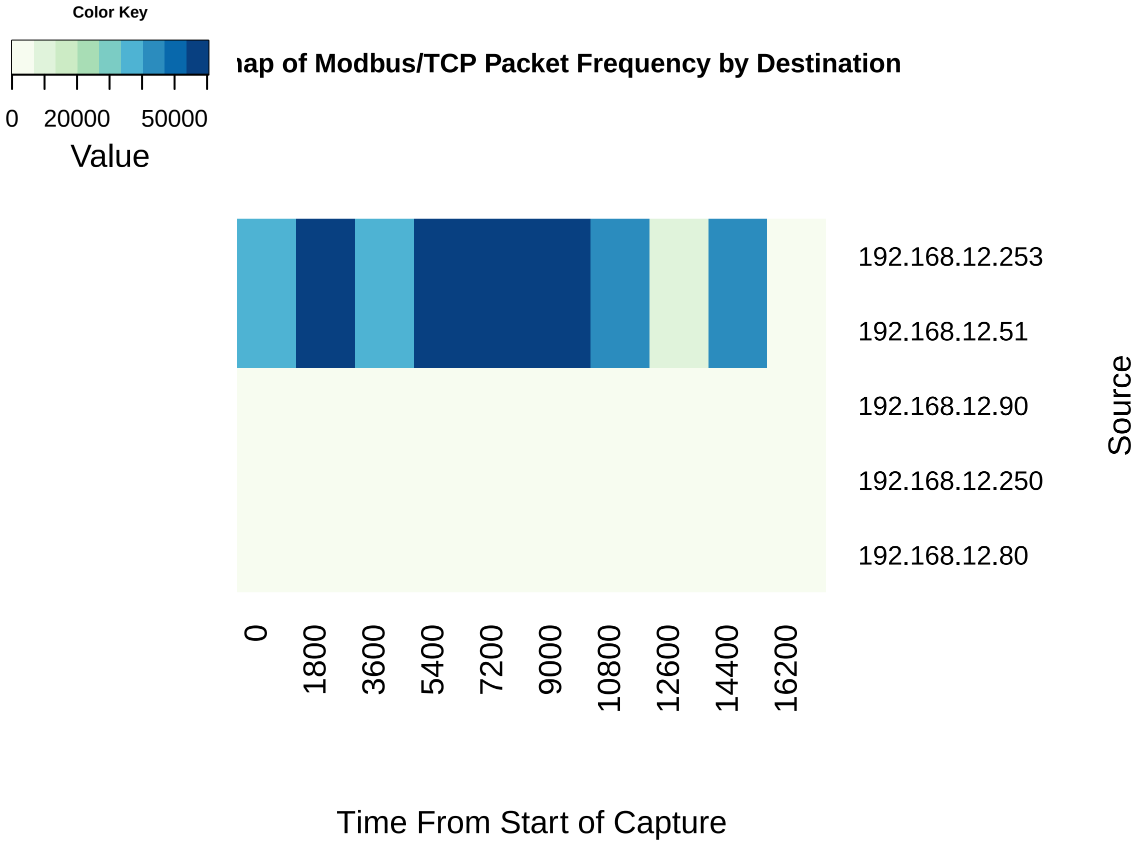

Scatter Plot of Destination and Start Time by Address
192.168.12.128
192.168.12.250
192.168.12.252
5000
10000
0
5000
10000
15000
0
2000
4000
6000
192.168.12.253
192.168.12.51
192.168.50.11
0
5000
10000
150000
5000
10000
15000
0
1000
2000
3000
4000
192.168.50.18
192.168.50.50
83.133.119.197
5000
5250
5500
5750
0
1000
2000
3000
4000
5000
5000
10000
15000
Destination Port
Start Time


Boxplot of Destination and Packet Size by Address
0
10
20
30
192.168.50.18
192.168.50.50
83.133.119.197
192.168.12.253
192.168.12.51
192.168.50.11
192.168.12.128
192.168.12.250
192.168.12.252
0
10
20
30
0
10
20
30
MBytes


Color Key
0
1e+06
Packet Count
Heatmap of Packet Frequency by Source
192.168.12.51
192.168.12.253
192.168.50.50
192.168.12.250
192.168.12.128
192.168.50.11
Intel_03:5f:5b
Veritech_37:9b:5b
Vmware_73:7a:76
192.168.50.1
fe80::c8fd:ac99:e724:26e9
192.168.50.18
DlgAutom_56:ec:6d
Telemeca_0f:35:aa
fe80::488e:75ff:fea0:2422
fe80::250:b6ff:fe4f:d6c0
fe80::58aa:15ff:febb:a322
fe80::7cca:f2ff:fe61:5a77
fe80::88d0:fcff:fec2:77b3
fe80::b4cf:74ff:feba:c2cd
fe80::e88b:eff:fe6b:bfc5
26:73:f5:ee:88:a8
0e:10:ff:b7:dc:26
2a:84:83:48:30:2b
2e:8e:45:86:ec:5a
GoodWayI_4f:d6:c0
Vmware_c0:00:00
Vmware_c0:00:01
Vmware_c0:00:02
Vmware_c0:00:06
Intel_f7:82:bc
192.168.12.80
Spectrum_18:78:cd
Nsi_1e:00:6e
Intel_af:1c:7d
Siemens_45:5c:e4
83.133.119.197
Intel_bb:66:28
192.168.12.90
192.168.12.252
AdeptTec_5c:74:1e
Source
0
1800
3600
5400
7200
9000
10800
12600
14400
16200
18000
Time From Start of Capture


Color Key
0
1e+06
Packet Count
map of Packet Frequency by Destination
192.168.12.253
192.168.12.51
192.168.12.252
83.133.119.197
192.168.12.90
192.168.12.250
255.255.255.255
Vmware_73:7a:76
192.168.50.255
192.168.50.50
ff02::1:2
Intel_bb:66:28
Veritech_37:9b:5b
Nsi_1e:00:6e
ff02::c
239.255.255.250
ff02::1:3
224.0.0.252
DlgAutom_56:ec:6d
Intel_f7:82:bc
ff02::fb
Telemeca_0f:35:aa
Siemens_45:5c:e4
192.168.12.80
AdeptTec_5c:74:1e
192.168.12.16
PN−MC_00:00:00
Spectrum_18:78:cd
Intel_af:1c:7d
192.168.12.255
8.8.8.8
192.168.100.2
Intel_03:5f:5b
192.168.50.18
192.168.50.11
192.168.12.128
Broadcast
Destination
0
1800
3600
5400
7200
9000
10800
12600
14400
16200
18000
Time From Start of Capture


Color Key
0
20000
50000
Value
of Modbus/TCP Packet Frequency by Source
192.168.12.253
192.168.12.51
192.168.12.90
192.168.50.50
192.168.12.250
192.168.12.80
Source
0
1800
3600
5400
7200
9000
10800
12600
14400
16200
Time From Start of Capture


Color Key
0
20000
50000
Value
ıap of Modbus/TCP Packet Frequency by Destination
192.168.12.253
192.168.12.51
192.168.12.90
192.168.12.250
192.168.12.80
Source
0
1800
3600
5400
7200
9000
10800
12600
14400
16200
Time From Start of Capture

MODBUS/TCP Data

MODBUS/TCP requests are identified by packets having port number 502

```r
summary(requests)
```

```
##  frame.time_relative frame.time_delta_displayed   frame.len     ip.proto
##  Min.   :   0.0031   Min.   : 0.00001           Min.   : 54.0   6:51554
##  1st Qu.: 948.5749   1st Qu.: 0.00026           1st Qu.: 66.0
##  Median :1325.3119   Median : 0.00032           Median : 66.0
##  Mean   :1324.5160   Mean   : 0.02243           Mean   : 65.3
##  3rd Qu.:1703.6292   3rd Qu.: 0.00047           3rd Qu.: 66.0
##  Max.   :2063.4165   Max.   :77.63863           Max.   :315.0
##
##  ip.version            ip.src                 ip.dst        ip.hdr_len
##  4:51554    192.168.12.250:   45   192.168.12.250:  150   Min.   :20
##             192.168.12.253:    0   192.168.12.253:51404   1st Qu.:20
##             192.168.12.51 :50948   192.168.12.51 :    0   Median :20
##             192.168.12.90 :  518   192.168.12.90 :    0   Mean   :20
##             192.168.50.50 :   43                          3rd Qu.:20
##                                                           Max.   :20
##
##   tcp.srcport     tcp.dstport    mbtcp.prot_id mbtcp.trans_id
##  2499   :50792   502    :51554    : 3247       Min.   :    0.0
##  1032   :  463   1032   :    0   0:48307       1st Qu.:   63.0
##  1742   :   77   1033   :    0                 Median :  128.0
##  1034   :   30   1034   :    0                 Mean   :  268.1
##  1033   :   23   1742   :    0                 3rd Qu.:  192.0
##  1744   :   11   1744   :    0                 Max.   :58880.0
##  (Other):  158   (Other):    0                 NA's   :3247
##    mbtcp.len       mbtcp.modbus.func_code mbtcp.modbus.reference_num
##  Min.   :  4.000     : 3247               Min.   :0.000
##  1st Qu.:  6.000   1 :  245               1st Qu.:0.000
##  Median :  6.000   4 :47969               Median :1.000
##  Mean   :  6.035   43:    1               Mean   :0.761
##  3rd Qu.:  6.000   90:   92               3rd Qu.:1.000
##  Max.   :255.000                          Max.   :3.000
##  NA's   :3247                             NA's   :3340
##  mbtcp.modbus.word_cnt mbtcp.modbus.data
##  Min.   :1                     :51505
##  1st Qu.:1             00:04   :   10
##  Median :1             01:04   :   10
##  Mean   :1             00:01:00:    6
##  3rd Qu.:1             00:02   :    6
##  Max.   :1             01:12   :    4
##  NA's   :3585          (Other) :   13
```

```r
table(moddataDT[,mbtcp.modbus.func_code])
```

```
##
##           1     4    43    90
##  3425   489 95937     2   147
```

MODBUS/TCP data

value vs time

References

[1] L. Maliphol, SCAD@COPS: A Hybrid Network Intrusion Detection System

[2] J.W. Tukey, (1977). Exploratory Data Analysis. Addison-Wesley. ISBN 0-201-07616-0 P. Lafaye de Micheaux et al., The R Software: Fundamentals of Programming and Statistical Analysis, Statistics and Computing

Appendix A

Using the export facility in Wireshark, the following are a description of the exported files:

SCADA_20150429_csv - entire pcap file exported in CSV format Fields: Time, Source, Destination, Protocol, Length, Info

SCADA_Security_042915_TCP_Endpoints.csv - list of endpoints, the traffic to and from an IP address Fields: Address, Port, Packets, Bytes, Tx.Packets, Tx.Bytes, Rx.Packets, Rx.Bytes, Latitude, Longitude

SCADA_Security_042915_TCP_Conversations.csv - list of conversations, the traffic between two endpoints Fields: Address.A, Port.A, Address.B, Port.B, Packets, Bytes, Packets.A.B, Bytes.A.B, Packets.A.B.1, Bytes.A.B.1, Rel.Start, Duration, bps.A.B, bps.A.B.1

Appendix B

Commands and Scripts

TShark

Command used to extract various fields from the pcap file used for analysis.

```
tshark -r modbus_100k -T fields -E separator=, -t r -E header=y -e frame.time_relative -e frame.time_delta_displayed -e frame.len -e ip.proto -e ip.version -e ip.src -e ip.dst -e tcp.srcport -e tcp.dstport -e mbtcp.prot_id -e mbtcp.trans_id -e mbtcp.len -e mbtcp.modbus.func_code -e mbtcp.modbus.reference_num -e mbtcp.modbus.word_cnt -e mbtcp.modbus.data > data.txt
```

sed

Command used to remove empty lines from the pcap data.

```
sed ’/ˆ,.*$/d’ modbus.data > modbus_transform.data
```

R

scada.R - Script in the language R containing for conducting statistical analysis and creating graphic visualisations.