 Exploratory Data Analysis  
Virtual SCADA Network:

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 Appendix B.

## Wireshark[[1]](#footnote-23) - 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.

## TShark[[2]](#footnote-26)

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.

## UNIX Utilities

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]](#footnote-30)

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]](#footnote-34) 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\_Security\_042915.pc | ap |
| 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 |

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.

### 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.

### 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.

### 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.

## 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[3])

### 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.

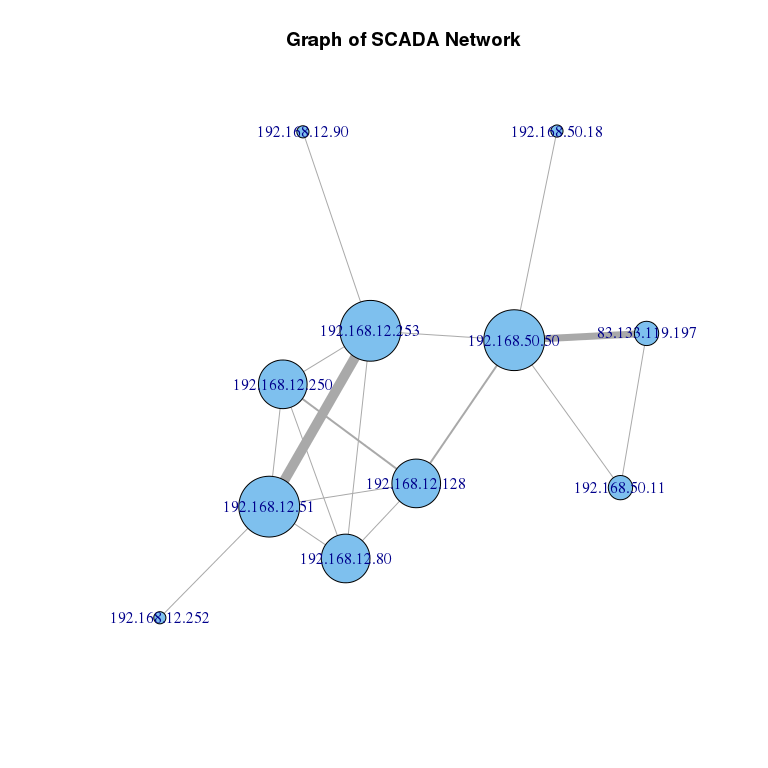
### Network Graph

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



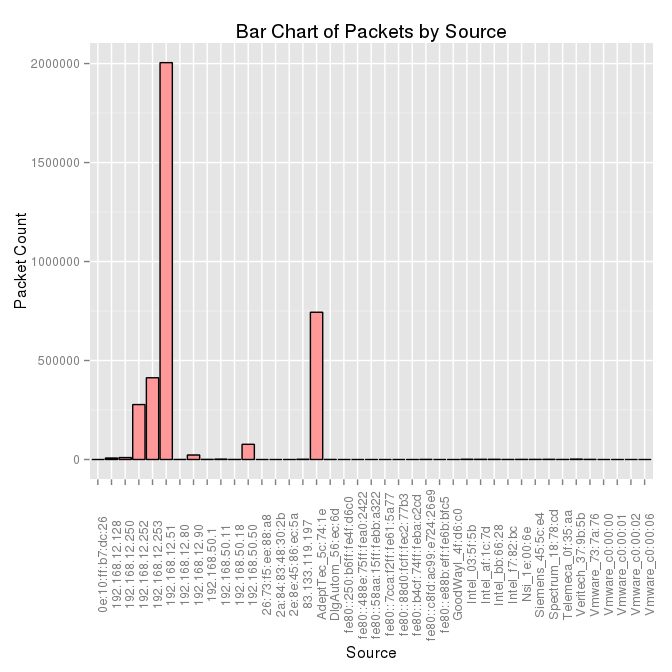
In the network graph shown above, the size of the node is according to its degree of centrality, that is, the number of adjacent vertices. The thicker edges indicate a higher number of interactions between two nodes.

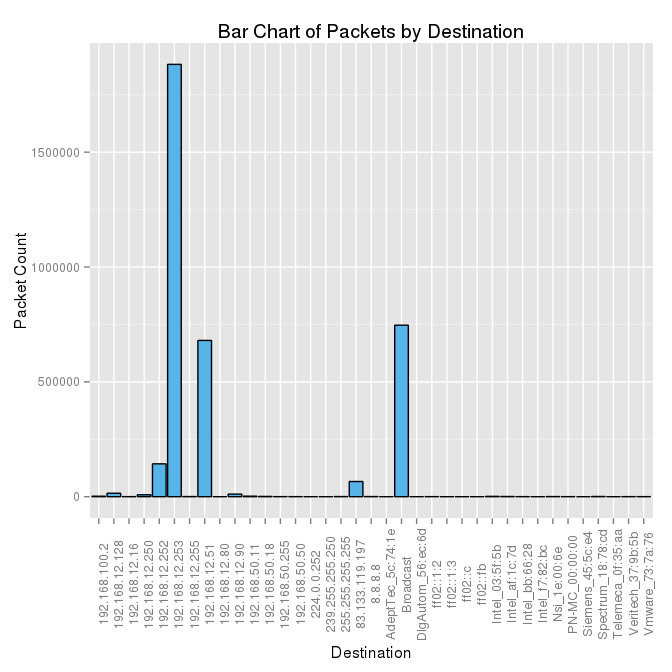
|  |  |
| --- | --- |
| Node IP Addresses |  |
| 192.168.12.253 | Schneider |
| 192.168.12.51 | HMI |
| 192.168.50.50 |  |
| 83.133.119.197 |  |
| 192.168.12.80 |  |
| 192.168.12.250 |  |
| 192.168.12.128 |  |
| 192.168.50.11 |  |
| 192.168.50.18 |  |
| 192.168.12.90 |  |
| 192.168.12.252 |  |

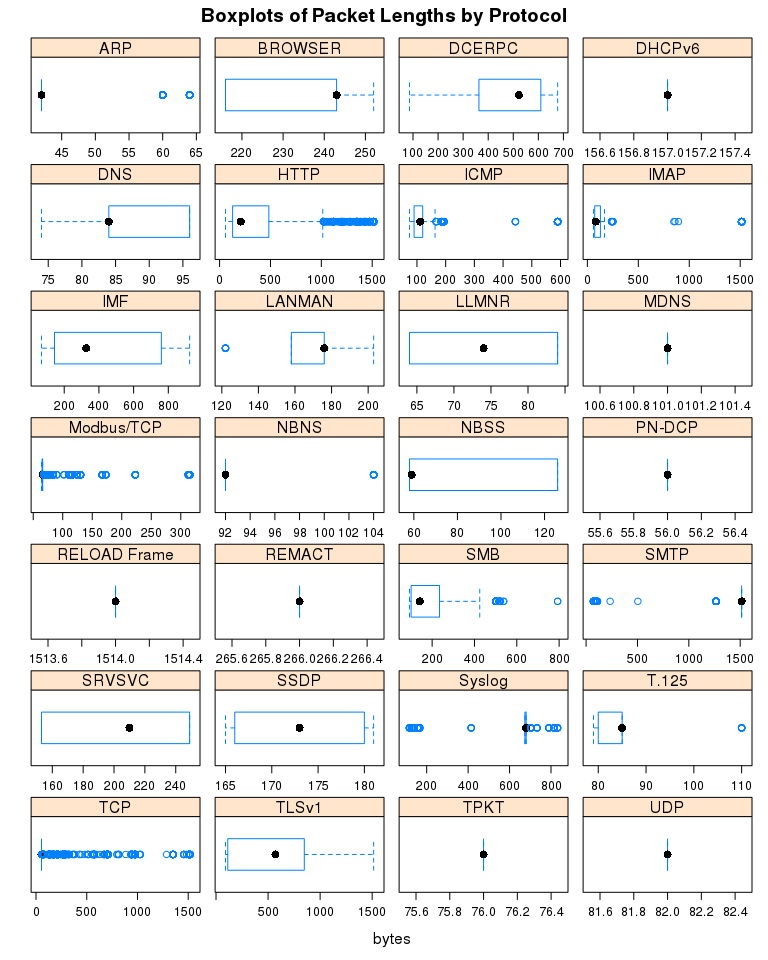
### Packet Length Statistics

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



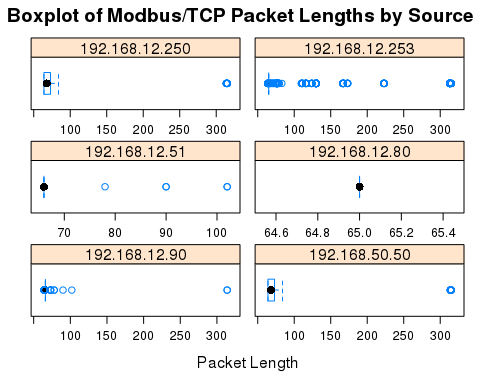


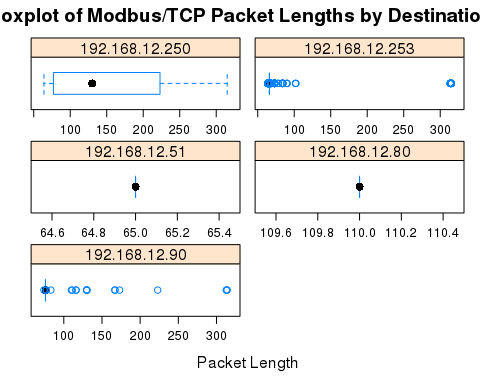


### Modbus/TCP Statistics

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

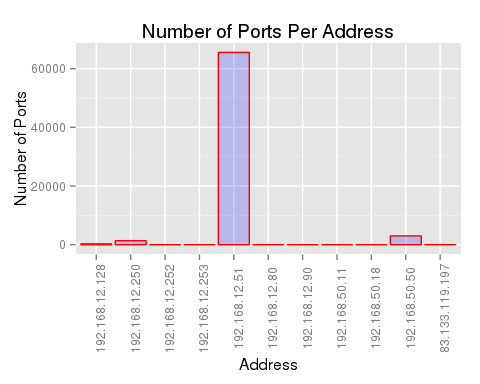


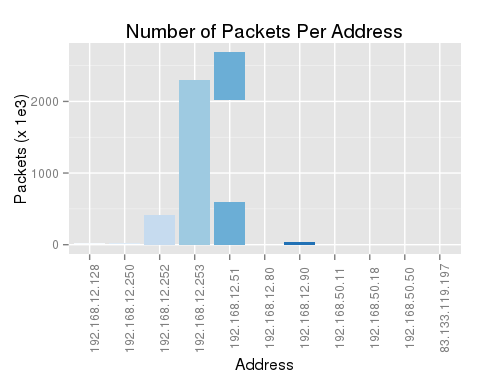


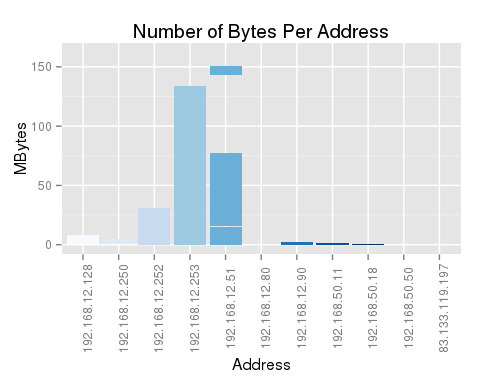
TODO describe figs

### Endpoints

SCADA\_Security\_042915\_TCP\_Endpoints.csv







### Correlation and Covariance

# TODO spearman?  
cor(packets, 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(packets,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

### MODBUS/TCP Data[[5]](#footnote-68)

TODO change to full data

MODBUS/TCP responses are identified by packets having source port number 502

summary(responses)

## frame.time\_relative frame.time\_delta\_displayed frame.len ip.proto   
## Min. : 0.0 Min. : 0.00000 Min. : 54.00 6:48446   
## 1st Qu.: 946.6 1st Qu.: 0.00953 1st Qu.: 65.00   
## Median :1321.4 Median : 0.00969 Median : 65.00   
## Mean :1321.2 Mean : 0.01872 Mean : 65.12   
## 3rd Qu.:1698.1 3rd Qu.: 0.00980 3rd Qu.: 65.00   
## Max. :2063.4 Max. :150.30267 Max. :315.00   
##   
## ip.version ip.src ip.dst ip.hdr\_len  
## 4:48446 192.168.12.250: 150 192.168.12.250: 47 Min. :20   
## 192.168.12.253:48296 192.168.12.253: 0 1st Qu.:20   
## 192.168.12.51 : 0 192.168.12.51 :48127 Median :20   
## 192.168.12.90 : 0 192.168.12.90 : 272 Mean :20   
## 192.168.50.50 : 0 3rd Qu.:20   
## Max. :20   
##   
## tcp.srcport tcp.dstport mbtcp.prot\_id mbtcp.trans\_id   
## 502 :48446 2499 :47969 : 178 Min. : 0   
## 1032 : 0 1032 : 232 0:48268 1st Qu.: 63   
## 1033 : 0 1742 : 40 Median : 128   
## 1034 : 0 1033 : 24 Mean : 268   
## 1742 : 0 1034 : 15 3rd Qu.: 192   
## 1744 : 0 2017 : 8 Max. :58880   
## (Other): 0 (Other): 158 NA's :178   
## mbtcp.len mbtcp.modbus.func\_code mbtcp.modbus.reference\_num  
## Min. : 4.000 : 178 Min. : NA   
## 1st Qu.: 5.000 1 : 244 1st Qu.: NA   
## Median : 5.000 4 :47968 Median : NA   
## Mean : 5.164 43: 1 Mean :NaN   
## 3rd Qu.: 5.000 90: 55 3rd Qu.: NA   
## Max. :255.000 Max. : NA   
## NA's :178 NA's :48446   
## mbtcp.modbus.word\_cnt mbtcp.modbus.data  
## Min. : NA 00:75 :19754   
## 1st Qu.: NA 00:50 :18396   
## Median : NA 00:54 : 4174   
## Mean :NaN 12:14 : 1618   
## 3rd Qu.: NA 0a:b8 : 1281   
## Max. : NA 0a:b6 : 986   
## NA's :48446 (Other): 2237

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

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

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](mailto:SCAD@COPS): A Hybrid Network Intrusion Detection System

[2] J.W. Tukey, (1977). Exploratory Data Analysis. Addison-Wesley. ISBN 0-201-07616-0

[3] P. Lafaye de Micheaux et al., (2013). The R Software: Fundamentals of Programming and Statistical Analysis, Statistics and Computing. Springer New York. ISBN 978-1-4614-9019-7

# Appendix A

TODO change table titles

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

Entire pcap file exported in CSV format:

## SCADA\_20150429\_042915.csv

Time  
Source  
Destination  
Protocol  
Length  
Info

List of endpoints, the traffic to and from an IP address:

## SCADA\_Security\_042915\_TCP\_Endpoints.csv

Address  
Port  
Packets  
Bytes  
Tx.Packets  
Tx.Bytes  
Rx.Packets  
Rx.Bytes  
Latitude  
Longitude

List of conversations, the traffic between two endpoints :

## SCADA\_Security\_042915\_TCP\_Conversations.csv

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 SCADA\_Security\_042915\_modbus.pcap -T fields -E separator=, -t r -E header=y -e frame.number -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 > modbus.data

### sed

Command used to remove empty lines from the pcap data.

TODO changed modbus dataset

sed '/^,.\*$/d' modbus.data > modbus\_transform.data

### R

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

1. <https://www.wireshark.org/docs/wsug_html_chunked> [↑](#footnote-ref-23)
2. <https://www.wireshark.org/docs/man-pages/tshark.html> [↑](#footnote-ref-26)
3. <http://www.r-project.org/> [↑](#footnote-ref-30)
4. <http://www.winpcap.org/ntar/draft/PCAP-DumpFileFormat.html> [↑](#footnote-ref-34)
5. <https://www.wireshark.org/docs/dfref/m/mbtcp.html> [↑](#footnote-ref-68)