Network Security
389.159 - SS 2018
Lab Exercise 3 & Lab Exercise 4

TEAM 02
Corentin Bergès (11741629) (066 506)
Christoph Echtinger-Sieghart (00304130) (066 938)

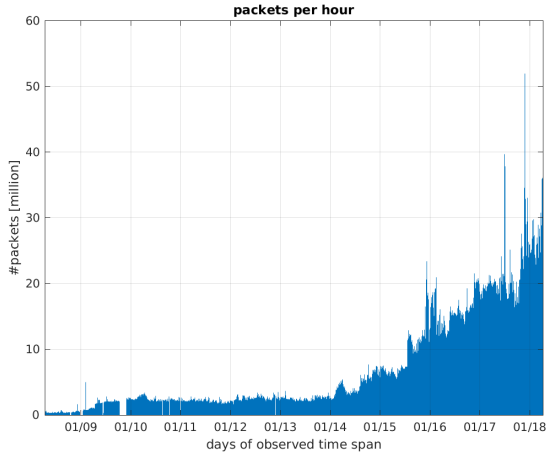
June 11, 2018

List of Corrections

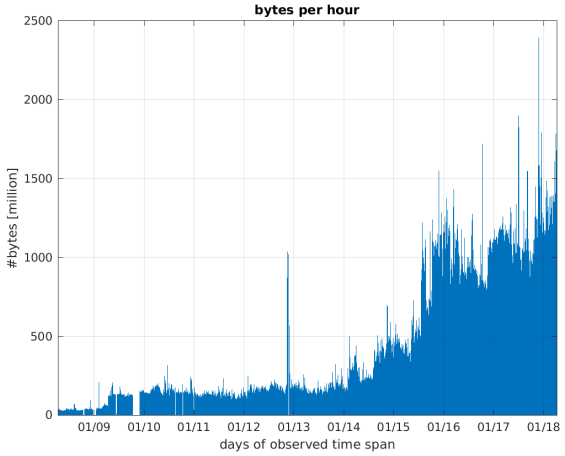
1 Lab Exercise 3

1.1 rep-10

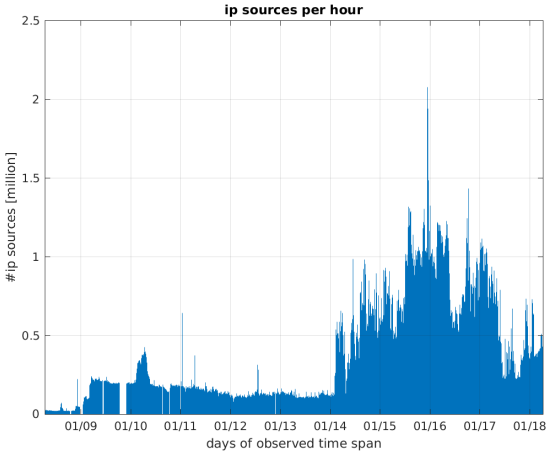
Matlab Code (Listing 2)
Figure 1 shows the stem plots for packets, bytes, unique IP sources and unique IP destinations per hour.



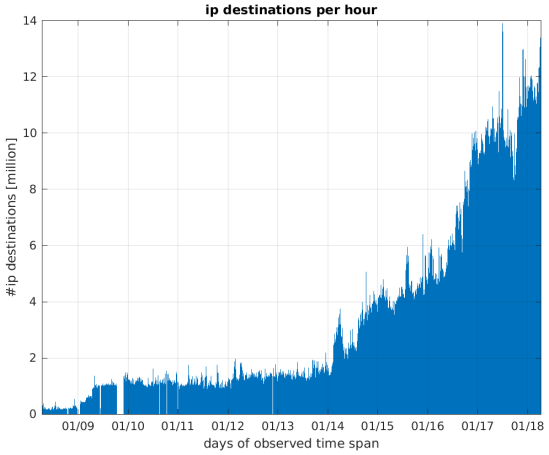
(a) Packets per hour



(b) Bytes per hour



(c) IP sources per hour



(d) IP destinations per hour

Figure 1

Optional Figure 2 shows all signals from Figure 1 combined, normalized and smoothed with a moving average filter.

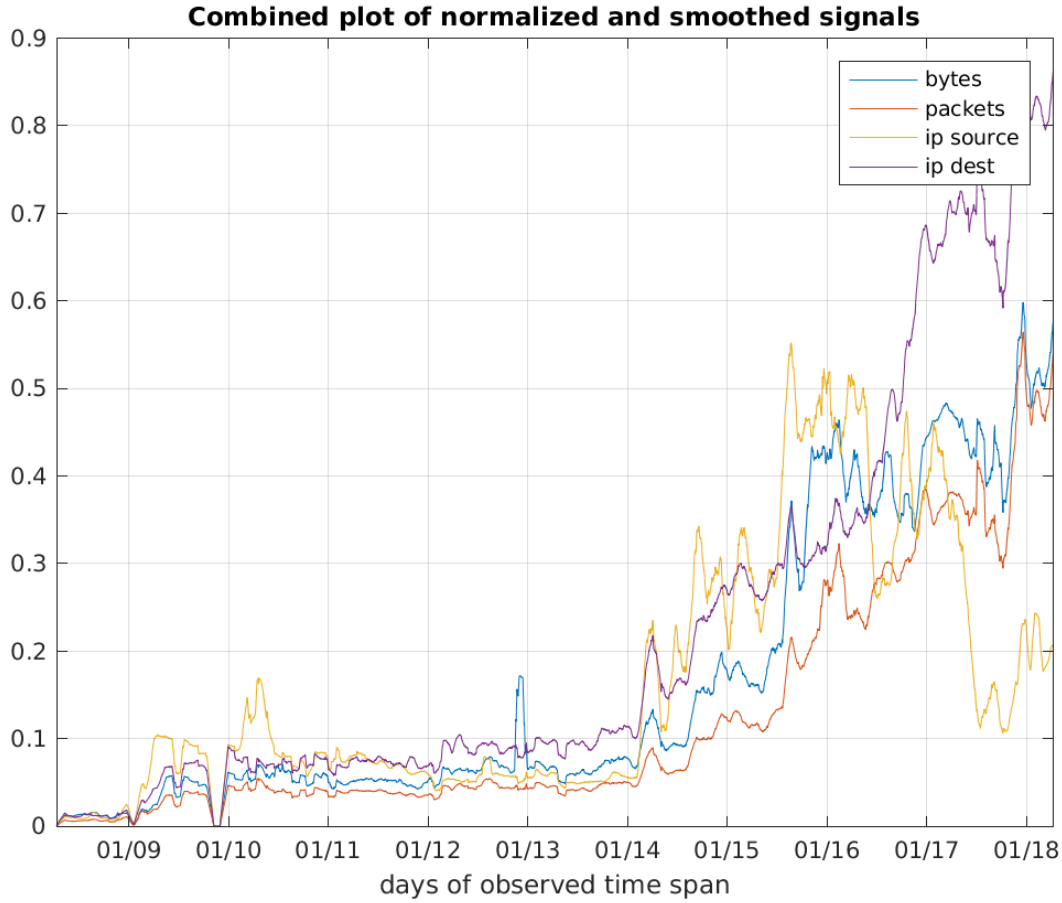


Figure 2: Combined, normalized and smoothed signals

1.2 rep-11

Matlab Code (Listing 3)

The signal that shows the lower correlation to the other signals is **IP sources**. The minimum linear correlation coefficient is **0.588568** between the signals **IP sources** and **IP destinations**. See Table 1 for the raw data.

	Bytes	Packets	IP src	IP dst
Bytes	1	0.9655	0.7203	0.9340
Packets	0.9655	1	0.6105	0.9732
IP src	0.7203	0.6105	1	0.5886
IP dst	0.9340	0.9732	0.5886	1

Table 1: Correlation coefficients between signals

The reason for why the drop in unique IP sources does not cause a proportional drop in the other signals, could be that many small attackers (botnets), that did not contribute a lot to the other signals somehow stopped sending traffic.

1.3 rep-12

Matlab Code (Listing 4)

The number of IP sources is bigger in average than the number of darkspace addresses receiving packets. There are on average around ten times more IP sources than IP destinations. This makes sense, because the darkspace

is only a small part of the internet address-space, but the IP sources are taken from the whole address-space.

1.4 rep-13

Matlab Code (Listing 5)
The main peak in IP sources starts on 14-Dec-2015 and lasts until 16-Dec-2015. See Table 2 for the detailed data.

Date	# IP sources
14-Dec-2015	2075358.074306
15-Dec-2015	1704892.012500
16-Dec-2015	1942072.404167

Table 2: Detailed data for peak in IP sources

Optional Matlab Code (Listing 6) The main peak in Bytes starts on 14-Nov-2012 and lasts until 22-Nov-2012. Note that on 19-Nov-2012 no data was available. See Table 3 for the detailed data.

Date	# Bytes
14-Nov-2012	870858582.136110
15-Nov-2012	1009586335.331900
16-Nov-2012	1038654926.456100
17-Nov-2012	1021464983.022200
18-Nov-2012	954193481.914190
20-Nov-2012	1005163238.508500
21-Nov-2012	1020526661.658000
22-Nov-2012	989613880.615110

Table 3: Detailed data for peak in Bytes

1.5 rep-14

Matlab Code (Listing 7)
Table 4 gives statistics for the data from `global_last10years.csv`. Table 5 gives statistics for the data from `Feb2017_gen.csv`.

1.6 rep-15

The values do not coincide. February 2017 seems to be a month that is not really representative for the data collected over a span of 10 years.

Optional Matlab Code (Listing 8)
A difference between the box plots for the hourly and daily data are the positions of the medians in the plots. The medians in the daily averaged data show a clear tendency to be in the vicinity of the first quartile, whereas the medians in the hourly averaged data show no clear tendency. A second noticeable difference are the outliers. All outliers in the daily averaged data are above the whiskers, whereas the boxplots for the hourly averaged data show outliers below and above the whiskers.

	Sum	Mean	Median	StdDev
# Packets [millions]	146373.391	41.845	17.699	40.916
# Bytes [millions]	2381.003	0.681	0.263	0.735
# IP src [millions]	123.613	0.035	0.020	0.031
# IP dst [millions]	1150.796	0.329	0.142	0.330

Table 4: Statistics for daily data

	Sum	Mean	Median	StdDev
# Packets [millions]	76871.319	114.392	113.464	7.033
# Bytes [millions]	1272.998	1.894	1.890	0.097
# IP src [millions]	59.651	0.089	0.091	0.018
# IP dst [millions]	619.875	0.922	0.931	0.070

Table 5: Statistics for hourly data

1.7 rep-16

We used https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml to look up the protocol numbers.

Protocol 6 (TCP) The Transmission Control Protocol is a connection oriented, reliable protocol that is widely used. A TCP connection is created by performing a three-way handshake (SYN, SYN/ACK, ACK). TCP uses port numbers to address applications.

Protocol 1 (ICMP) The Internet Control Message Protocol is a protocol used to exchange status and error messages concerning the IP protocol. ICMP is transported via IP. The popular `ping` and `traceroute` programs are applications that make use of ICMP.

Protocol 17 (UDP) The User Datagram Protocol is a connectionless, unreliable protocol. UDP does not perform a three-way handshake, but also uses port numbers to address applications.

1.8 rep-17

Matlab Code (Listing 9)
Table 6 shows statistical information for Packets/hour grouped by protocol. Table 7 shows statistical information for IP sources/hour grouped by protocol. Table 8 shows statistical information for IP destinations/hour grouped by protocol.

	Mean	Median	StdDev
TCP	0.840	0.833	0.048
UDP	0.114	0.113	0.032
ICMP	0.043	0.054	0.026
Others	0.004	0.004	0.001

Table 6: Statistical information for Packets/hour (percentages)

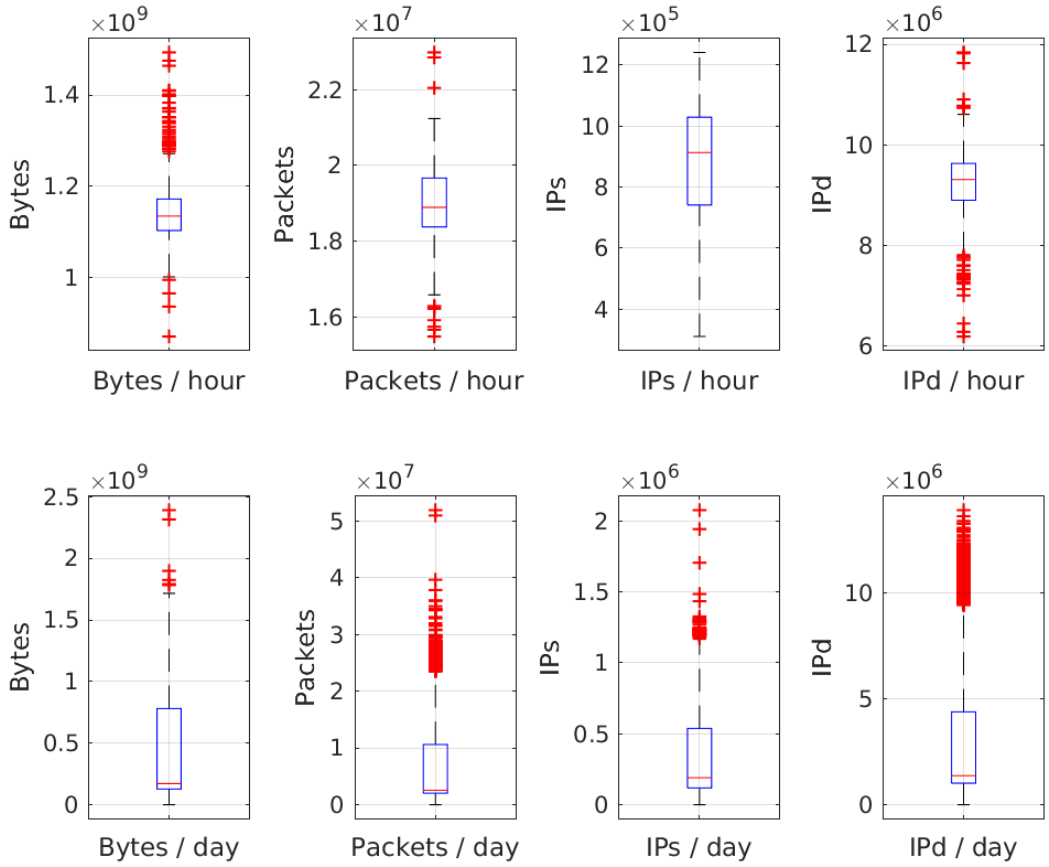


Figure 3: Boxplots for hourly and daily averaged data

	Mean	Median	StdDev
TCP	0.592	0.519	0.143
UDP	0.336	0.343	0.034
ICMP	0.175	0.225	0.106
Others	-0.103	-0.103	0.044

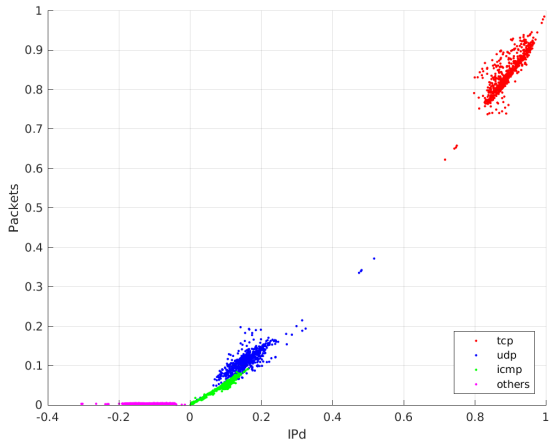
Table 7: Statistical information for IP sources/hour (percentages)

	Mean	Median	StdDev
TCP	0.890	0.883	0.039
UDP	0.158	0.155	0.045
ICMP	0.080	0.104	0.048
Others	-0.127	-0.142	0.042

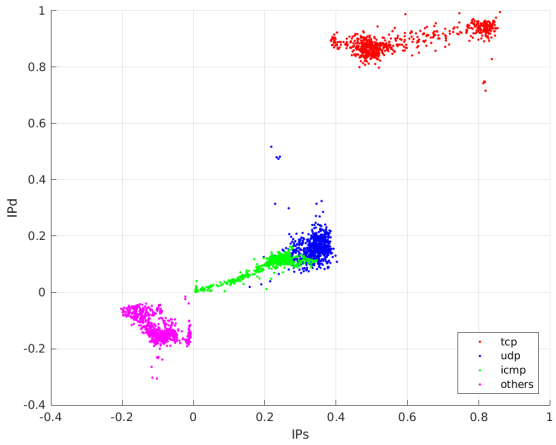
Table 8: Statistical information for IP destinations/hour (percentages)

FiXme Error boxplot per signal

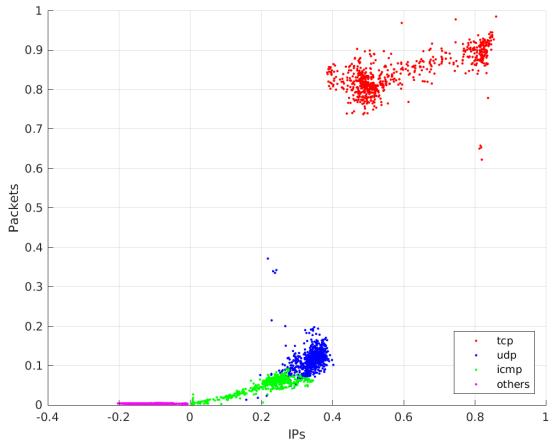
Optional Figure 4 shows the various scatter plots.



(a) Packets vs IP destinations



(b) IP destinations vs IP sources



(c) Packets vs IP sources

Figure 4: Scatter plots

1.9 rep-18

We obtained negative values for "Others" because, in the combined data addresses are collapsed. The same address might get a TCP, a UDP and an ICMP packet, but will only be counted once.

1.10 rep-19

Matlab Code (Listing 10)

1.11 rep-20

1.12 rep-21

1.13 rep-22

1.14 rep-23

Listing 1: Command used to obtain IP address

```
team02@pc01:~$ ip address show dev em1
```

Port 113

```
IP 192.168.83.20.1073 > 192.168.83.33.113: Flags [S], seq 0, win 8192, length 0
IP 192.168.83.33.113 > 192.168.83.20.1073: Flags [R.], seq 0, ack 1, win 0, length 0
```

2 Lab Exercise 4

2.1 rep-24

2.2 rep-25

2.3 rep-26

2.4 rep-27

2.5 rep-28

2.6 rep-29

2.7 rep-30

A Matlab Code

Listing 2: Matlab code to solve rep-10

```
function team02_rep10
% rep-10
    [timestamps, bytes, packets, ip_s, ip_d] = read_custom_csv('~/workfiles/global_last10years.csv')

    function save_stem_plot(data, my_title, y_label, filename)
    % Do a stem plot of data in millions and write it to filename.png
        set (gca, 'fontname', 'Helvetica', 'fontsize', 20)
        figure
        stem(timestamps, data/10^6, 'marker', 'none')
        datetick('x', 'mm/yy');
        xlabel('days␣of␣observed␣time␣span');
        ylabel(y_label);
        title(my_title);
        grid on
        set(gca, 'layer', 'top');
        xlim([min(timestamps) max(timestamps)]);
        saveas(gcf, filename, 'png')
    end

    save_stem_plot(bytes, 'bytes␣per␣hour', '#bytes␣[million]', 'plots/rep_10_2');
    save_stem_plot(packets, 'packets␣per␣hour', '#packets␣[million]', 'plots/rep_10_1');
    save_stem_plot(ip_s, 'ip␣sources␣per␣hour', '#ip␣sources␣[million]', 'plots/rep_10_3');
    save_stem_plot(ip_d, 'ip␣destinations␣per␣hour', '#ip␣destinations␣[million]', 'plots/rep_10_4')

    % optional part

    function result = smooth_filter(data)
    % Moving averages filter for data
        window_size = 30;
        b = (1 / window_size) * ones(1, window_size);
        a = 1;
        % 1-D digital filter
        result = filter(b, a, data);
    end

    smooth_bytes = smooth_filter(bytes / unique(max(bytes)));
    smooth_packets = smooth_filter(packets / unique(max(packets)));
    smooth_ip_s = smooth_filter(ip_s / unique(max(ip_s)));
    smooth_ip_d = smooth_filter(ip_d / unique(max(ip_d)));

    figure
    plot(...
        timestamps, smooth_bytes, '-', ...
        timestamps, smooth_packets, '-', ...
        timestamps, smooth_ip_s, '-', ...
        timestamps, smooth_ip_d, '-' ...
    );
    legend('bytes', 'packets', 'ip␣source', 'ip␣dest');
    datetick('x', 'mm/yy');
    xlabel('days␣of␣observed␣time␣span');
    title('Combined␣plot␣of␣normalized␣and␣smoothed␣signals');
    grid on
    set(gca, 'layer', 'top');
    xlim([min(timestamps) max(timestamps)]);
    saveas(gcf, 'plots/rep_10_optional', 'png')
end
```

Listing 3: Matlab code to solve rep-11

```
function team02_rep11
% rep-11
    [~, bytes, packets, ip_s, ip_d] = read_custom_csv('~/workfiles/global_last10years.csv');

    function result = correlation(a, b)
        result = unique(min(corrcoef(a, b)));
    end

    names = { ...
        'Bytes <-> Packets', 'Bytes <-> IPs', 'Bytes <-> IPd', ...
        'Packets <-> IPs', 'Packets <-> IPd', 'IPs <-> IPd' ...
    };
    correlations = [ ...
        correlation(bytes, packets), correlation(bytes, ip_s), ...
        correlation(bytes, ip_d), correlation(packets, ip_s), ...
        correlation(packets, ip_d), correlation(ip_s, ip_d) ...
    ];
    [minimum_coeff, idx] = min(correlations);
    fprintf('Minimum linear correlation coeff: %f (%s)\n', minimum_coeff, names{idx});

    names_signal = {'Bytes', 'Packets', 'IPs', 'IPd'};

    means = [ ...
        % Bytes         | Packets         | IPs           | IPd
        correlations(1), correlations(1), correlations(2), correlations(3); ...
        correlations(2), correlations(4), correlations(4), correlations(5); ...
        correlations(3), correlations(5), correlations(6), correlations(6)
    ];
    disp(names_signal);
    disp(means);
end
```

Listing 4: Matlab code to solve rep-12

```
function team02_rep12
    [~, ~, ~, ip_s, ip_d] = read_custom_csv('~/workfiles/global_last10years.csv');
    ip_s(ip_s==0) = NaN;
    ip_d(ip_d==0) = NaN;

    fprintf('Ratio IPs to IPd: %f\n', nanmean(ip_s) / nanmean(ip_d));
end
```

Listing 5: Matlab code to solve rep-13

```
function team02_rep13
    [timestamps, ~, ~, ip_s, ~] = read_custom_csv('~/workfiles/global_last10years.csv');
    % from visual inspection
    cutoff = 1.5*10^6;
    peak_locations = ip_s>cutoff;

    peak_timestamps = timestamps(peak_locations);
    peaks = ip_s(peak_locations);

    dates = arrayfun(@datestr, peak_timestamps, 'UniformOutput', false);
    result = dates';
    result(2,:) = num2cell(peaks);
    fprintf('%s: %f IPs\n', result{:});
end
```

Listing 6: Matlab code to solve rep-13 optional

```
function team02_rep13_optional
    [timestamps, bytes, ~, ~, ~] = read_custom_csv('~/workfiles/global_last10years.csv');
    % From visual inspection
    cutoff = 8*10^8;
    timestamps = timestamps(timestamps<=datenum('2014-01-01'));
    bytes = bytes(timestamps>0);

    peak_locations = bytes>cutoff;
    peak_timestamps = timestamps(peak_locations);
    peaks = bytes(peak_locations);

    dates = arrayfun(@datestr, peak_timestamps, 'UniformOutput', false);
    result = dates';
    result(2,:) = num2cell(peaks);
    fprintf('%s: %f Bytes\n', result{:});
    % NOTE: There is a gap because on 19-nov-2012 there was no data
end
```

Listing 7: Matlab code to solve rep-14

```
function team02_rep14

    function result = stats(data)
        data(data==0) = NaN;
        result = round([nansum(data), nanmean(data), nanmedian(data), nanstd(data)] ./ 10e6, 3);
    end

    disp('---- Daily avg ---');
    [~, bytes, packets, ip_s, ip_d] = read_custom_csv('~/workfiles/global_last10years.csv');
    for col = horzcat(bytes, packets, ip_s, ip_d)
        fprintf('%.3f %.3f %.3f %.3f\n', stats(col));
    end

    disp('----- Hourly avg ---');

    % WARNING: order is different
    [~, packets, bytes, ip_s, ip_d] = read_custom_csv('~/workfiles/Feb2017_gen.csv');
    for col = horzcat(bytes, packets, ip_s, ip_d)
        fprintf('%.3f %.3f %.3f %.3f\n', stats(col));
    end
end
```

Listing 8: Matlab code to solve rep-15 optional

```
function team02_rep15_optional
    [~, bytes_daily, packets_daily, ip_s_daily, ip_d_daily] = read_custom_csv('~/workfiles/global_la
    % WARNING order is different
    [~, packets_hourly, bytes_hourly, ip_s_hourly, ip_d_hourly] = read_custom_csv('~/workfiles/Feb20

    set (gca, 'fontname', 'Helvetica', 'fontsize', 20)

    ax1 = subplot(2,4,1);
    boxplot(ax1, bytes_hourly, 'Labels', {''})
    ylabel(ax1, 'Bytes');
    xlabel(ax1, 'Bytes / hour');
    grid on
    set(gca, 'layer', 'top');

    ax2 = subplot(2,4,2);
    boxplot(ax2, packets_hourly, 'Labels', {''})
    ylabel(ax2, 'Packets');
    xlabel(ax2, 'Packets / hour');
    grid on
    set(gca, 'layer', 'top');

    ax3 = subplot(2,4,3);
    boxplot(ax3, ip_s_hourly, 'Labels', {''})
    ylabel(ax3, 'IPs');
    xlabel(ax3, 'IPs / hour');
    grid on
    set(gca, 'layer', 'top');

    ax4 = subplot(2,4,4);
    boxplot(ax4, ip_d_hourly, 'Labels', {''})
    ylabel(ax4, 'IPd');
    xlabel(ax4, 'IPd / hour');
    grid on
    set(gca, 'layer', 'top');

    ax5 = subplot(2,4,5);
    boxplot(ax5, bytes_daily, 'Labels', {''})
    ylabel(ax5, 'Bytes');
    xlabel(ax5, 'Bytes / day');
    grid on
    set(gca, 'layer', 'top');

    ax6 = subplot(2,4,6);
    boxplot(ax6, packets_daily, 'Labels', {''})
    ylabel(ax6, 'Packets');
    xlabel(ax6, 'Packets / day');
    grid on
    set(gca, 'layer', 'top');

    ax7 = subplot(2,4,7);
    boxplot(ax7, ip_s_daily, 'Labels', {''})
    ylabel(ax7, 'IPs');
    xlabel(ax7, 'IPs / day');
    grid on
    set(gca, 'layer', 'top');

    ax8 = subplot(2,4,8);
    boxplot(ax8, ip_d_daily, 'Labels', {''})
    ylabel(ax8, 'IPd');
    xlabel(ax8, 'IPd / day');
    grid on
    set(gca, 'layer', 'top');

    saveas(gcf, 'plots/rep_15_optional.png', 'png')
end
```

Listing 9: Matlab code to solve rep-17

```
function team02_rep17
    % WARNING: order is switched
    [~, combined_packets, ~, combined_ip_s, combined_ip_d] = read_custom_csv('~/workfiles/Feb2017_ge
    [~, tcp, udp, icmp] = read_custom_protocol_csv('~/workfiles/Feb2017_proto.csv');
    % packets, ip_s, ip_d

   function result = packets(data)
        result = data(:,1);
    end

    function result = ip_s(data)
        result = data(:,2);
    end

    function result = ip_d(data)
        result = data(:,3);
    end

    others_packets = combined_packets - packets(tcp) - packets(udp) - packets(icmp);
    others_ip_s = combined_ip_s - ip_s(tcp) - ip_s(udp) - ip_s(icmp);
    others_ip_d = combined_ip_d - ip_d(tcp) - ip_d(udp) - ip_d(icmp);
    others = horzcat(others_packets, others_ip_s, others_ip_d);

    function result = percentages(data)
        p = packets(data) ./ combined_packets;
        s = ip_s(data) ./ combined_ip_s;
        d = ip_d(data) ./ combined_ip_d;
        result = horzcat(p, s, d);
    end

    tcp_percentages = percentages(tcp);
    udp_percentages = percentages(udp);
    icmp_percentages = percentages(icmp);
    others_percentages = percentages(others);

    function table(t, u, i, o)
        fprintf('%f␣%f␣%f\n', mean(t), median(t), std(t));
        fprintf('%f␣%f␣%f\n', mean(u), median(u), std(u));
        fprintf('%f␣%f␣%f\n', mean(i), median(i), std(i));
        fprintf('%f␣%f␣%f\n', mean(o), median(o), std(o));
    end

    table(packets(tcp_percentages), packets(udp_percentages), packets(icmp_percentages), packets(oth
    disp('--');
    table(ip_s(tcp_percentages), ip_s(udp_percentages), ip_s(icmp_percentages), ip_s(others_percenta
    disp('--');
    table(ip_d(tcp_percentages), ip_d(udp_percentages), ip_d(icmp_percentages), ip_d(others_percenta

    function stat_plot(data, my_title)
        figure
        ax1 = subplot(1, 3, 1);
        boxplot(ax1, packets(data), 'Labels', {''})
        ylabel(ax1, 'Packets');
        xlabel(ax1, 'Packets␣/␣hour');
        grid on
        set(gca, 'layer', 'top');

        ax2 = subplot(1, 3, 2);
        boxplot(ax2, ip_s(data), 'Labels', {''})
        ylabel(ax2, 'IPs');
        xlabel(ax2, 'IPs␣/␣hour');
        title(my_title)
        grid on
        set(gca, 'layer', 'top');

        ax3 = subplot(1, 3, 3);
        boxplot(ax3, ip_d(data), 'Labels', {''})
        ylabel(ax3, 'IPd');
        xlabel(ax3, 'IPd␣/␣hour');
        grid on
```

Listing 10: Matlab code to solve rep-19

```
function team02_rep19()
    [port_data, column_names] = read_tcp_ports_csv('~/workfiles/Feb2017_TCPdstport.csv');
    [~, tcp, ~, ~] = read_custom_protocol_csv('~/workfiles/Feb2017_proto.csv');

    tcp_packets = tcp(:,1);
    % for element-wise division later on
    tcp_packets = horzcat(tcp_packets, tcp_packets, tcp_packets, tcp_packets);

    % We don't want the timestamp
    port_data = port_data(:,2:end);
    column_names = column_names(:,2:end);

    % Sum over the columns and sort descending
    [~, idx] = sort(sum(port_data), 'descend');

    port_data = port_data(:,idx);
    port_data = port_data(:,1:4);

    column_names = column_names(idx);
    fprintf('Most␣used␣ports:␣%s␣%s␣%s␣%s\n', column_names{:,1:4});

    % Absolute values
    disp('Absolute');
    fprintf('mean␣%.3f␣%.3f␣%.3f␣%.3f\n', (mean(port_data) ./ 10e6));
    fprintf('std␣␣%.3f␣%.3f␣%.3f␣%.3f\n', (std(port_data) ./ 10e6));

    disp('Perc');
    fprintf('mean␣%.1f␣%.1f␣%.1f␣%.1f\n', mean(port_data ./ tcp_packets) * 100);
    fprintf('std␣␣%.1f␣%.1f␣%.1f␣%.1f\n', std(port_data ./ tcp_packets) * 100);
end
```