mix_protocols traffic volume

Rafilx

2022-07-11

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

## Loading required package: viridisLite

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

##
## Attaching package: 'scales'

## The following object is masked from 'package:viridis':
##
##     viridis_pal
```

R Markdown

Analisar a porcentagem de equisições por protocolo, dividindo por períodos. Essa análise não envolve os payloads, apenas os quantitativos de requisições.

Resultados esperados:

- gráficos de linhas e de barras mostrando a evolução

Dados analisados

- Todos os protocolos foram agrupados em um único database para realizar a análise de dados

```
db <- dbConnect(RSQLite::SQLite(), dbname="../db/database-2022-05-11/mix_protocol.sqlite")

data_unfetch <-dbSendQuery(db, "
  SELECT *, CAST(CAST(year AS text) || CAST(period AS text) as integer) as year_period
    FROM (
      SELECT *,  strftime(\"%Y\", tempo_inicio) as year, ((strftime(\"%m\", tempo_final) - 1) / 3) + 1 
        FROM MIX_PROTOCOL
    )
")
data <- fetch(data_unfetch)

dbDisconnect(db)
```

```
## Warning in connection_release(conn@ptr): There are 1 result in use. The
## connection will be released when they are closed
```

```
data['tempo_final_cast'] = as.POSIXct(data[['tempo_final']], format = "%Y-%m-%d %H:%M:%S")
data['tempo_inicio_cast'] = as.POSIXct(data[['tempo_inicio']], format = "%Y-%m-%d %H:%M:%S")

minimum_percentage_as_others = 5
decimals_digits = 2
```

- Agrupamento por trimestre

```
data_grouped_period = data %>%
  mutate(year_period_int = year_period,
         vitima_ip = as.factor(vitima_ip),
         year_period = as.factor(year_period)) %>%
  group_by(year_period) %>%
  summarise(sum_requests_per_attack = sum(requests_per_attack),
            number_of_attacks = n(),
            count_victim = n_distinct(vitima_ip))

data_grouped_period_percentage = data_grouped_period %>%
  ungroup() %>%
  group_by() %>%
  summarise(year_period = year_period,
            number_of_attacks = number_of_attacks,
            sum_requests_per_attack = sum_requests_per_attack,
```

```
            count_victim = count_victim,
            sum_count_victim = sum(count_victim),
            sum_all_number_of_attacks = sum(number_of_attacks),
            sum_all_requests_per_attack = sum(sum_requests_per_attack)) %>%
  mutate(number_of_attacks_percentage = (number_of_attacks / sum_all_number_of_attacks) * 100,
         number_of_requests_percentage = (sum_requests_per_attack / sum_all_requests_per_attack) * 100,
         number_of_victim_percentage = (count_victim / sum_count_victim) * 100)

data_grouped_period_percentage_selected = data_grouped_period_percentage %>%
  select('year_period', 'number_of_attacks_percentage', 'number_of_requests_percentage', 'number_of_vic

# data_grouped_period_percentage = data_grouped_period_protocol_percentage %>%
#   mutate(
#     attack_protocol = case_when(
#       number_of_requests_percentage < minimum_percentage_as_others ~ "OUTROS",
#       TRUE ~ as.character(attack_protocol)
#     )
#   ) %>%
#   group_by(year_period, attack_protocol) %>%
#   summarise(number_of_requests_percentage = sum(number_of_requests_percentage))
```

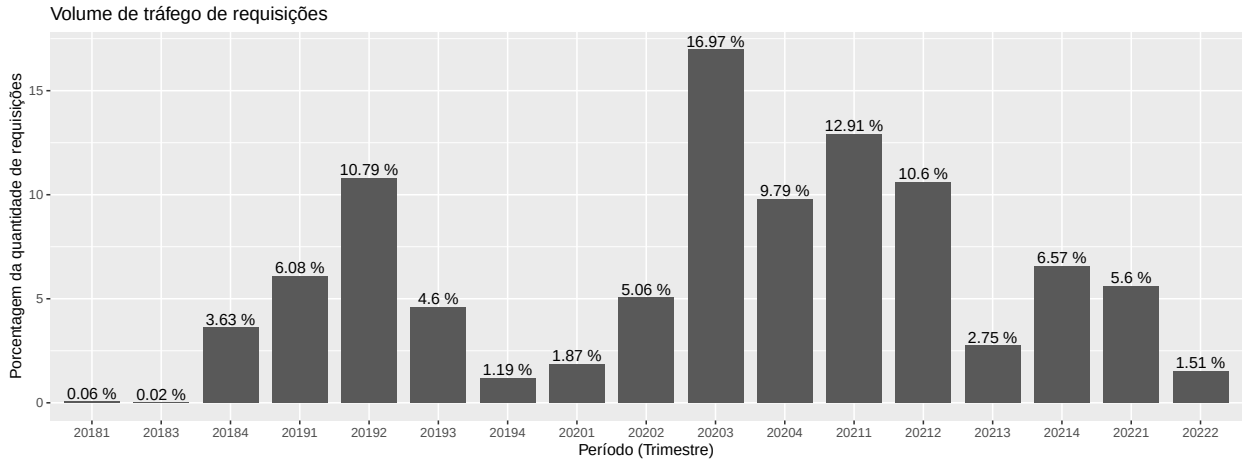
- Todos os trimestres em porcentagem

```
data_grouped_period_percentage_selected %>%
  mutate(number_of_attacks_percentage = round(number_of_attacks_percentage, 2),
         number_of_requests_percentage = round(number_of_requests_percentage, 2),
         number_of_victim_percentage = round(number_of_victim_percentage, 2)) %>%
  rename(attacks = number_of_attacks_percentage,
         requests = number_of_requests_percentage,
         victim = number_of_victim_percentage) %>%
  print(n=17)
```

```
## # A tibble: 17 x 4
##    year_period attacks requests victim
##    <fct>         <dbl>    <dbl>  <dbl>
##  1 20181          0        0.06   0
##  2 20183          0.02     0.02   0.05
##  3 20184          2.01     3.63   2.68
##  4 20191          2.2      6.08   4.8
##  5 20192          3.89    10.8    8.18
##  6 20193          4.69     4.6   12.1
##  7 20194          0.45     1.19   1.15
##  8 20201          2.69     1.87  11.7
##  9 20202          1.68     5.06   4.8
## 10 20203          3.98    17.0    9.23
## 11 20204          7.23     9.79  14.5
## 12 20211          4.11    12.9    8.04
## 13 20212         12.0     10.6    6.91
## 14 20213         12.9      2.75   3.13
## 15 20214         13.4      6.57   5.4
## 16 20221         20.7      5.6    3.47
## 17 20222          8.11     1.51   3.95
```

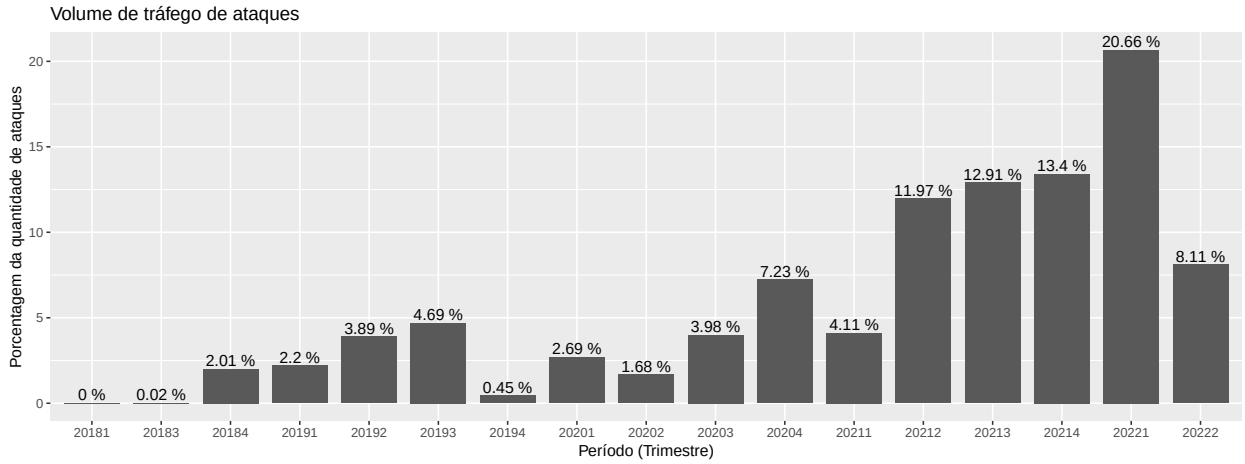
- Total de requisições por trimestre

```
data_grouped_period_percentage_selected %>%
  ggplot( aes(x=year_period, y=number_of_requests_percentage)) +
  geom_bar(stat="identity", width = 0.8, position="dodge") +
  geom_text(aes(label = paste(round(number_of_requests_percentage, decimals_digits), "%"),  vjust = -0.1
  scale_fill_viridis(discrete=TRUE) +
  #theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  ylab("Porcentagem da quantidade de requisições") +
  xlab("Período (Trimestre)") +
  ggtitle("Volume de tráfego de requisições")
```



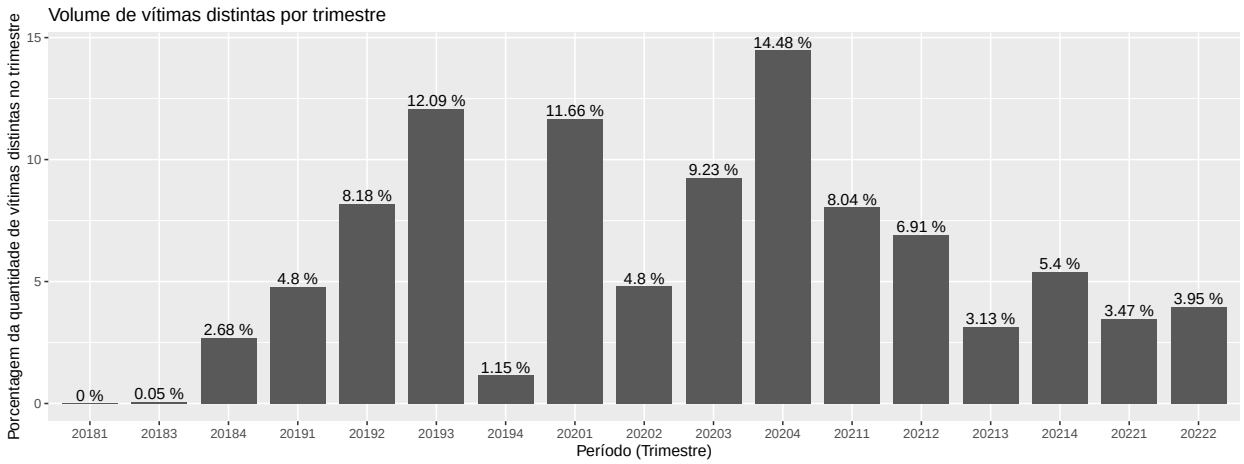
- Total de ataques por trimestre

```
data_grouped_period_percentage_selected %>%
  ggplot( aes(x=year_period, y=number_of_attacks_percentage)) +
  geom_bar(stat="identity", width = 0.8, position="dodge") +
  geom_text(aes(label = paste(round(number_of_attacks_percentage, decimals_digits), "%"),  vjust = -0.25
  scale_fill_viridis(discrete=TRUE) +
  #theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  ylab("Porcentagem da quantidade de ataques") +
  xlab("Período (Trimestre)") +
  ggtitle("Volume de tráfego de ataques")
```



- Total de vítimas distintas por trimestre

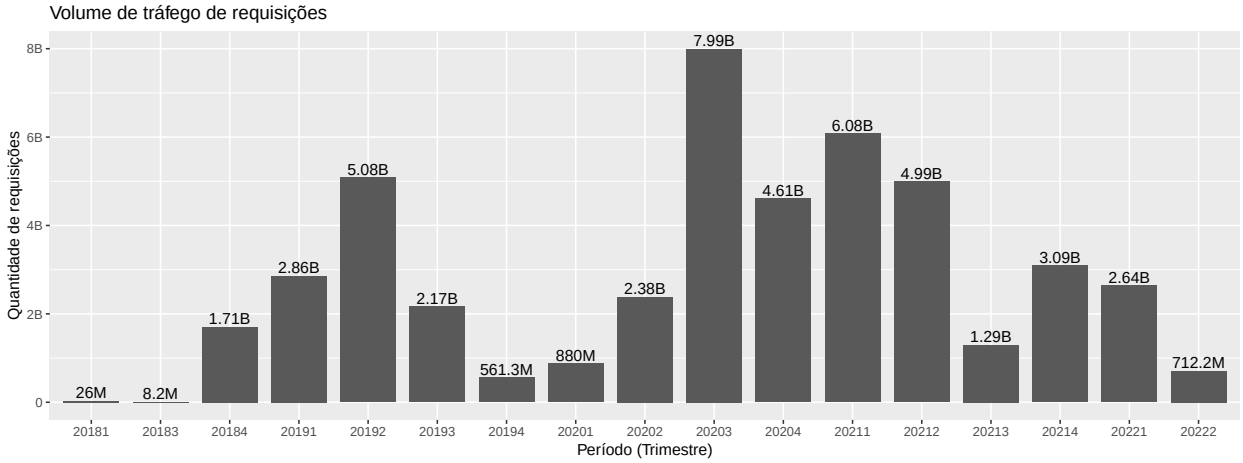
```
data_grouped_period_percentage_selected %>%
  ggplot( aes(x=year_period, y=number_of_victim_percentage)) +
  geom_bar(stat="identity", width = 0.8, position="dodge") +
  geom_text(aes(label = paste(round(number_of_victim_percentage, decimals_digits), "%"),  vjust = -0.25
  scale_fill_viridis(discrete=TRUE) +
  #theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  ylab("Porcentagem da quantidade de vítimas distintas no trimestre") +
  xlab("Período (Trimestre)") +
  ggtitle("Volume de vítimas distintas por trimestre")
```



Apresentar os resultados com números ao invés de porcentagens

- Total de requisições por trimestre

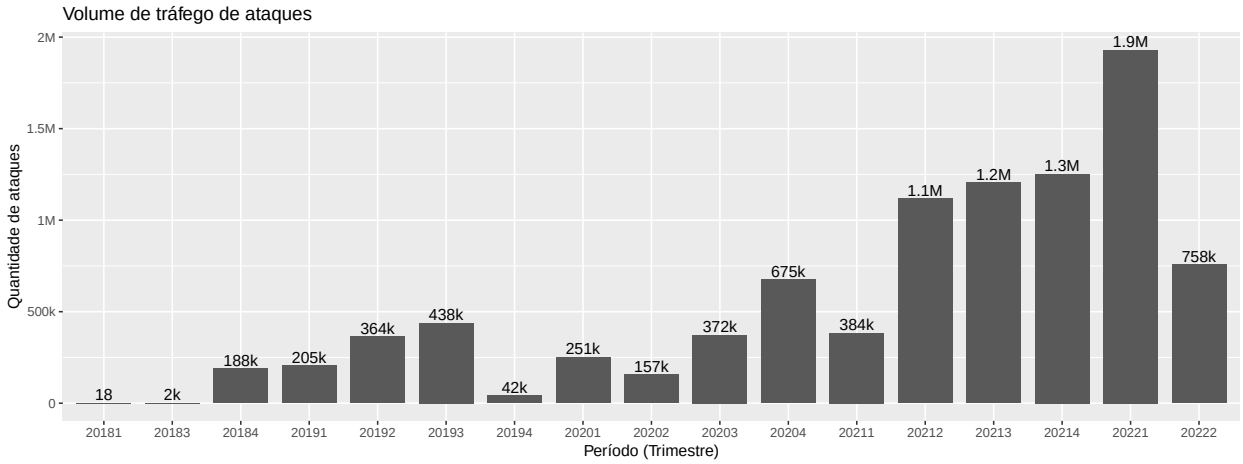
```
data_grouped_period %>%
  ggplot( aes(x=year_period, y=sum_requests_per_attack)) +
  geom_bar(stat="identity", width = 0.8, position="dodge") +
  geom_text(aes(label = addUnits(sum_requests_per_attack),  vjust = -0.25)) +
  scale_fill_viridis(discrete=TRUE) +
  scale_y_continuous(labels = addUnits) +
  ylab("Quantidade de requisições") +
  xlab("Período (Trimestre)") +
  ggtitle("Volume de tráfego de requisições")
```



- O terceiro trimestre de 2020 foi o que teve o maior volume de requisições, alcançando a marca de 7.99 Bilhões de requisições realizadas

- Total de ataques por trimestre

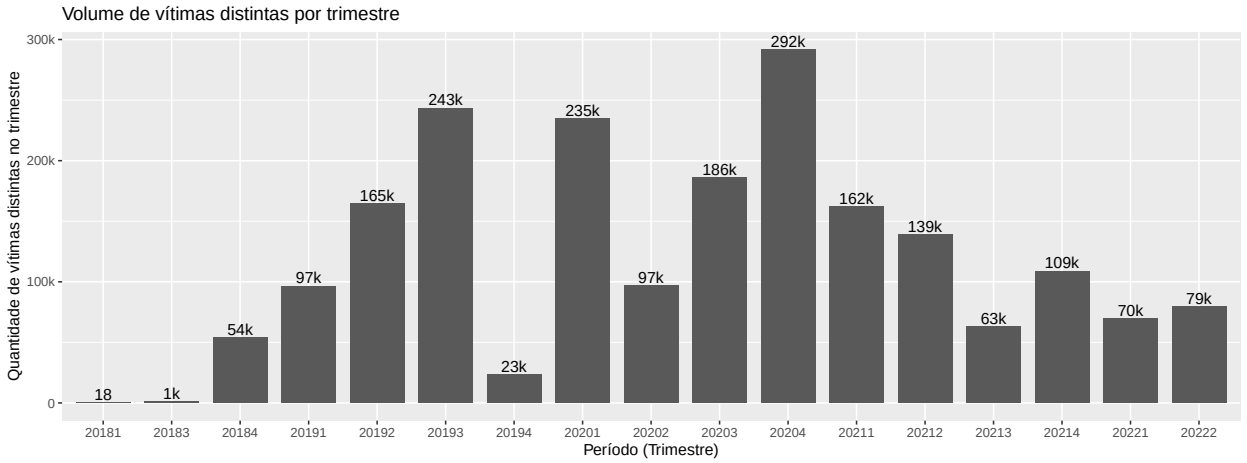
```
data_grouped_period %>%
  ggplot( aes(x=year_period, y=number_of_attacks)) +
  geom_bar(stat="identity", width = 0.8, position="dodge") +
  geom_text(aes(label = addUnits(number_of_attacks),  vjust = -0.25)) +
  scale_fill_viridis(discrete=TRUE) +
  scale_y_continuous(labels = addUnits) +
  ylab("Quantidade de ataques") +
  xlab("Período (Trimestre)") +
  ggtitle("Volume de tráfego de ataques")
```



- O trimestre com maior número de requisições foi o 2020.3 com 7.99 Bilhões de requisições em apenas 372 mil ataques, representando que uma maior quantidade de requisições não significam maior número de ataques, mas sim que tiveram ataques com muitas requisições O trimestre com maior número de ataques foi o de 2022.1 com 1.9 Milhões de ataques registrados para apenas 2.64 Bilhões de requisições.

- Total de vítimas distintas por trimestre
  - é contado as vítimas distintas por trimestre, então a vítima de ip "52.233.175.59" que aparece no trimestre 20204 e 20211 será contado como um nos dois ou mais trimestres em que esse ip é atacado

```
data_grouped_period %>%
  ggplot( aes(x=year_period, y=count_victim)) +
  geom_bar(stat="identity", width = 0.8, position="dodge") +
  geom_text(aes(label = addUnits(count_victim),  vjust = -0.25)) +
  scale_fill_viridis(discrete=TRUE) +
  scale_y_continuous(labels = addUnits) +
  #theme(axis.text.x=element_text(angle=60, hjust=1)) +
  # scale_fill_manual(values=safe_colorblind_palette) +
  # scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x), labels = trans_format("log10", math
  ylab("Quantidade de vítimas distintas no trimestre") +
  xlab("Período (Trimestre)") +
  ggtitle("Volume de vítimas distintas por trimestre")
```



Vítimas novas

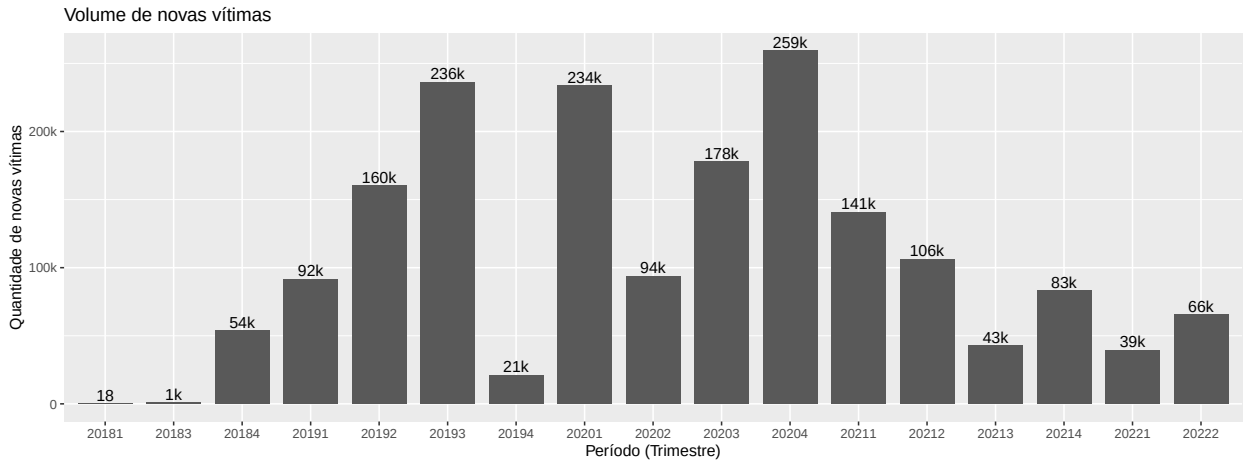
- Será verificado o aparecimento e desaparecimento de ip de vítimas durante os períodos

```
# data %>%
#   filter(year_period == 20191) %>%
#   count(vitima_ip)
#
# data %>%
#   ungroup() %>%
#   filter(year_period == 20191) %>%
#   group_by(vitima_ip) %>%
#   summarise(year_period = year_period)


data_new_victim_period = data %>%
  ungroup() %>%
  group_by(vitima_ip) %>%
  summarise(year_period = min(year_period)) %>%
  # filter(year_period == 20181) %>%
  ungroup() %>%
  group_by(year_period) %>%
  summarise(new_victims = n_distinct(vitima_ip)) %>%
  mutate(year_period=as.factor(year_period))
```

- Total de novas vítimas que surgiram em cada trimestre
  - é contado as vítimas distintas, então a vítima de ip "52.233.175.59" que aparece no trimestre 20204 e 20211 será contado como um somente no primeiro momento em que apareceu (20204)

```
data_new_victim_period %>%
  ggplot( aes(x=year_period, y=new_victims)) +
  geom_bar(stat="identity", width = 0.8, position="dodge") +
  geom_text(aes(label = addUnits(new_victims),  vjust = -0.25)) +
  scale_fill_viridis(discrete=TRUE) +
  scale_y_continuous(labels = addUnits) +
  # theme(axis.text.x=element_text(angle=60, hjust=1)) +
  # scale_fill_manual(values=safe_colorblind_palette) +
  # scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x), labels = trans_format("log10", math
  ylab("Quantidade de novas vítimas") +
  xlab("Período (Trimestre)") +
  ggtitle("Volume de novas vítimas")
```



- Existe um número significativo de novas vítimas em todos os trimestres, com aumentos expressivos no terceiro trimestre de 2020 em que o protocolo CoAP foi adicionado o que pode apresentar que no terceiro trimestre de 2020 o honeypot foi encontrado por scans e entre o terceiro e quarto trimestre ele iniciou sendo incorporado a listas de refletores disponíveis a ser usado por ferramentas de ataque ou booters