# dns_query_id_qname

2022-03-31

## R Markdown

- Executar as seguintes availiações por quarter
  - verificar se o mesmo queryid existe por vários periodos
  - verificar se o mesmo qname aparece por vários periodos
  - olhar as tabelas antes de plotar
  - Olhar o período passado para verificar o surgimento de novos casos de qname / query_id
  - verificar se tirar ou não o " HAVING qnt_repeat_query_id > 1"
  - quero gráficos de barra
  - group por qname, qtype e queryid para verificar a ocorrência de grupo de ataques

verificar rfc1034,1035 para queryid repetidos

```
library('RSQLite')
library('ggplot2')
library(DBI)
options("scipen"=100, "digits"=4)
library(dplyr)
```

```
##
## 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
```

```
library(tibble)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(lubridate)
```

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

```
db <- dbConnect(RSQLite::SQLite(), dbname="./dnstor_statistics_dns.sqlite")

data_unfetch <-dbSendQuery(db, "
  SELECT *, CAST(CAST(year AS text) || CAST(period AS text) as integer) as year_period
    FROM DNS_ANALYSIS
        JOIN DNS_ANALYSIS_QUESTION
          ON DNS_ANALYSIS.id = DNS_ANALYSIS_QUESTION.dns_analysis_id
     WHERE QTYPE != 0
")
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
```

- Contar a frequência que query_id aparece em cada year_period

```
data_split_year_period = data %>%
  group_split(year_period)

N=10

period_query_id = data.frame()
for (i in c(1:length(data_split_year_period))) {
  query_id_frequency = data_split_year_period[[i]] %>%
    count(query_id)

  query_id_frequency['year_period'] = data_split_year_period[[i]]$year_period[1]

  period_query_id = rbind(period_query_id, head(query_id_frequency[order(-query_id_frequency$n),], N) )
}
```

### Os 10 query_id mais utilizados divididos por período e ordenados pela frequência em que apareceram no período

```
period_query_id %>%
  group_split(year_period)
```

```
## <list_of<
##   tbl_df<
##     query_id   : integer
##     n          : integer
##     year_period: integer
##   >
## >[6]>
## [[1]]
## # A tibble: 10 x 3
##    query_id     n year_period
##       <int> <int>       <int>
##  1    17767  1917       20204
##  2    16049  1060       20204
##  3    56064  1049       20204
##  4    63710   782       20204
##  5    59797   741       20204
##  6    63374   729       20204
##  7    59378   723       20204
##  8    31694   718       20204
##  9    13304   715       20204
## 10    46512   707       20204
##
## [[2]]
## # A tibble: 10 x 3
##    query_id     n year_period
##       <int> <int>       <int>
##  1    17767 67047       20211
##  2    28940   418       20211
##  3    13551   318       20211
##  4    50265   305       20211
##  5    19592   277       20211
##  6    45810   214       20211
##  7    57166   197       20211
##  8    43855   168       20211
##  9    56643   125       20211
## 10    56686   124       20211
##
## [[3]]
## # A tibble: 10 x 3
##    query_id     n year_period
##       <int> <int>       <int>
##  1    26566  5090       20212
##  2    17767  3748       20212
##  3    13551   348       20212
##  4    50265   118       20212
##  5    37845    81       20212
##  6        1    65       20212
##  7    36379    60       20212
##  8    45810    59       20212
##  9     1525    47       20212
## 10    40074    38       20212
##
## [[4]]
## # A tibble: 10 x 3
##    query_id     n year_period
##       <int> <int>       <int>
##  1        1  6859       20213
##  2    17767  5838       20213
##  3    13551   783       20213
##  4       27   403       20213
##  5    59252   253       20213
##  6    60765   220       20213
##  7    13143   212       20213
##  8    53342   157       20213
##  9    65372   102       20213
## 10    14262   100       20213
##
## [[5]]
## # A tibble: 10 x 3
##    query_id     n year_period
##       <int> <int>       <int>
##  1    26566 29963       20214
##  2    17767  8479       20214
##  3        1   677       20214
##  4    13551   566       20214
##  5       27   102       20214
##  6    28826    70       20214
##  7    36609    69       20214
##  8     3803    60       20214
##  9    47132    58       20214
## 10    50265    51       20214
##
## [[6]]
## # A tibble: 10 x 3
##    query_id     n year_period
##       <int> <int>       <int>
##  1    26566  5876       20221
##  2    17767   895       20221
##  3    13551   124       20221
##  4        1   120       20221
##  5    64206    63       20221
##  6    28826    41       20221
##  7       27    29       20221
##  8    14602    19       20221
##  9    50293    15       20221
## 10        6    12       20221
```

### Os query_id que apareceram com maior frequência entre os top 10 em todos os períodos

- Caso o query_id 13213 fosse top 1 em 20204 e top 3 em 20211 e não aparecer em mais nenhum outro período seu "n" seria 2

```
period_query_id %>%
  count(query_id) %>%
  arrange(desc(n)) %>%
  filter(n > 1)
```

```
## # A tibble: 8 x 2
##   query_id     n
##      <int> <int>
## 1    17767     6
## 2    13551     5
## 3        1     4
## 4       27     3
## 5    26566     3
## 6    50265     3
## 7    28826     2
## 8    45810     2
```

**Group by qname, query_id**

```
period_query_id_qname = data.frame()
for (i in c(1:length(data_split_year_period))) {
  query_id_qname_frequency = data_split_year_period[[i]] %>%
    count(qname, qtype, query_id, year_period, sort = TRUE) %>%
    filter(n > 1)

  period_query_id_qname = rbind(period_query_id_qname, head(query_id_qname_frequency, N) )
}
```

**Os 10 query_id, qname, qtype mais utilizados divididos por período e ordenados pela frequência em que apareceram no período**

```
period_query_id_qname %>%
  group_split(year_period)
```

```
## <list_of<
##   tbl_df<
##     qname      : character
##     qtype      : character
##     query_id   : integer
##     year_period: integer
##     n          : integer
##   >
## >[6]>
## [[1]]
## # A tibble: 10 x 5
##    qname    qtype query_id year_period     n
##    <chr>    <chr>    <int>       <int> <int>
##  1 isc.org. ANY      17767       20204  1141
##  2 irs.gov. ANY      16049       20204  1060
##  3 irs.gov. ANY      56064       20204  1049
##  4 irs.gov. ANY      63710       20204   782
##  5 sl.      ANY      17767       20204   764
##  6 irs.gov. ANY      59797       20204   741
##  7 irs.gov. ANY      63374       20204   728
##  8 irs.gov. ANY      59378       20204   721
##  9 irs.gov. ANY      31694       20204   718
## 10 irs.gov. ANY      13304       20204   713
##
## [[2]]
## # A tibble: 10 x 5
##    qname         qtype query_id year_period     n
##    <chr>         <chr>    <int>       <int> <int>
##  1 isc.org.      ANY      17767       20211 56753
##  2 sl.           ANY      17767       20211 10260
##  3 .             ANY      28940       20211   417
##  4 VERSION.BIND. TXT      13551       20211   314
##  5 .             ANY      19592       20211   273
##  6 .             ANY      57166       20211   196
##  7 .             ANY      43855       20211   164
##  8 fe18.ru.      ANY      56643       20211   124
##  9 fe18.ru.      ANY      56686       20211   122
## 10 .             ANY      10000       20211   117
##
## [[3]]
## # A tibble: 10 x 5
##    qname                                     qtype query_id year_period     n
##    <chr>                                     <chr>    <int>       <int> <int>
##  1 peacecorps.gov.                           ANY      26566       20212  5090
##  2 sl.                                       ANY      17767       20212  3739
##  3 VERSION.BIND.                             TXT      13551       20212   346
##  4 213.1.168.192.in-addr.arpa.               PTR      37845       20212    80
##  5 com.                                      ANY       1525       20212    46
##  6 67b.org.                                  AAAA     40074       20212    38
##  7 hcc.nl.                                   ANY          3       20212    33
##  8 version.bind.                             TXT          6       20212    30
##  9 pizzaseo.com.                             RRSIG        1       20212    29
## 10 200-19-107-238.measurebr.xiaofengtest.com. A       50265       20212    24
##
## [[4]]
## # A tibble: 10 x 5
##    qname         qtype query_id year_period     n
##    <chr>         <chr>    <int>       <int> <int>
##  1 pizzaseo.com. RRSIG        1       20213  6236
##  2 sl.           ANY      17767       20213  5764
##  3 VERSION.BIND. TXT      13551       20213   783
##  4 pizzaseo.com. ANY         27       20213   403
##  5 .             ANY      59252       20213   252
##  6 .             ANY      60765       20213   219
##  7 .             ANY      13143       20213   212
##  8 .             ANY      53342       20213   155
##  9 .             ANY      65372       20213   102
## 10 .             ANY      14262       20213   100
##
## [[5]]
## # A tibble: 10 x 5
##    qname             qtype query_id year_period     n
##    <chr>             <chr>    <int>       <int> <int>
##  1 peacecorps.gov.   ANY      26566       20214 29963
##  2 sl.               ANY      17767       20214  8433
##  3 VERSION.BIND.     TXT      13551       20214   564
##  4 pizzaseo.com.     ANY         27       20214   101
##  5 pizzaseo.com.     RRSIG        1       20214    97
##  6 .                 ANY      36609       20214    69
##  7 ip.parrotdns.com. A        28826       20214    68
##  8 .                 ANY      47132       20214    58
##  9 .                 ANY       3803       20214    57
## 10 .                  ANY      20986       20214    47
##
## [[6]]
## # A tibble: 10 x 5
##    qname                     qtype query_id year_period     n
##    <chr>                     <chr>    <int>       <int> <int>
##  1 peacecorps.gov.           ANY      26566       20221  5876
##  2 sl.                       ANY      17767       20221   841
##  3 VERSION.BIND.             TXT      13551       20221   122
##  4 ip.parrotdns.com.         A        28826       20221    41
##  5 dnsscan.shadowserver.org. A        64206       20221    33
##  6 version.bind.             TXT      64206       20221    30
##  7 pizzaseo.com.             ANY         27       20221    29
##  8 isc.org.                  ANY      17767       20221    24
##  9 pizzaseo.com.             RRSIG        1       20221    21
## 10 clients1.google.com.      A        14602       20221    19
```

### Os query_id que apareceram com maior frequência entre os top 10 em todos os períodos, agrupados por qname e qtype

- Caso o query_id 13213 de qname = "isc.org." e qtype = "ANY" fosse top 1 em 20204 e top 3 em 20211 e não aparecer em mais nenhum outro período seu "n" seria 2

```r
top_queryid_qname = period_query_id_qname %>%
  count(query_id, qtype, qname) %>%
  arrange(desc(n)) %>%
  filter(n > 1)

top_queryid_qname
```

```
## # A tibble: 7 x 4
##   query_id qtype qname                 n
##      <int> <chr> <chr>             <int>
## 1    17767 ANY   sl.                   6
## 2    13551 TXT   VERSION.BIND.         5
## 3        1 RRSIG pizzaseo.com.         4
## 4       27 ANY   pizzaseo.com.         3
## 5    17767 ANY   isc.org.              3
## 6    26566 ANY   peacecorps.gov.       3
## 7    28826 A     ip.parrotdns.com.     2
```

### Checar se houve sobreposição temporal entre ataques com o mesmo query_id, qtype e diferentes qnames entre os top em cada período

```r
#data['tempo_final']
data['tempo_final_cast'] = as.POSIXct(data[['tempo_final']], format = "%Y-%m-%d %H:%M:%S")
data['tempo_final_date'] = date(data[['tempo_final_cast']])
data['tempo_final_hour'] = hour(data[['tempo_final_cast']])


data_overlap = select(data, 'query_id', 'qtype', 'qname', 'year_period', 'requests_per_attack', 'tempo_
top_repeated_query_id = top_queryid_qname %>%
  count(query_id) %>%
  filter(n > 1) %>%
  select('query_id')


top_repeated_qname = top_queryid_qname %>%
  filter(query_id %in% top_repeated_query_id[['query_id']]) %>%
  select('qname')


data_overlap_split_qname = data_overlap %>%
  filter(query_id %in% top_repeated_query_id[['query_id']]) %>%
  filter(qname %in% top_repeated_qname[['qname']]) %>%
  arrange(tempo_final_cast) %>%
  count(query_id, qname, tempo_final_date, tempo_final_hour) %>%
  group_split(qname)

top_overlap = data_overlap %>%
  filter(query_id %in% top_repeated_query_id[['query_id']]) %>%
  filter(qname %in% top_repeated_qname[['qname']]) %>%
  arrange(tempo_final_cast) %>%
  select('query_id', 'qtype', 'qname', 'tempo_final_date', 'tempo_final_hour')

head(top_overlap, N)
```

```
##    query_id qtype    qname tempo_final_date tempo_final_hour
## 1     17767   ANY isc.org.       2020-11-01               22
## 2     17767   ANY isc.org.       2020-11-01               23
## 3     17767   ANY isc.org.       2020-11-02               10
## 4     17767   ANY      sl.       2020-11-13                0
## 5     17767   ANY      sl.       2020-11-18                8
## 6     17767   ANY      sl.       2020-11-18                8
## 7     17767   ANY      sl.       2020-11-20                4
## 8     17767   ANY      sl.       2020-11-20                4
## 9     17767   ANY      sl.       2020-11-20                4
## 10    17767   ANY      sl.       2020-11-20                4
```

```
#all_equal(data_overlap_split_qname[[1]], data_overlap_split_qname[[2]], )

#duplicated(top_overlap, incomparables = TRUE)

#top_overlap[duplicated(top_overlap, incomparables = TRUE),]

#all.equal(select(data_overlap_split_qname[[1]], 'query_id', 'tempo_final_date', 'tempo_final_hour', 'q


#merge(select(data_overlap_split_qname[[1]], 'query_id', 'tempo_final_date', 'tempo_final_hour', 'qname


#data_overlap_split_qname[[1]] %>%
 # anti_join(data_overlap_split_qname[[2]], by="qname")
```

```
#data_overlap_split_qname[[2]] %>%
 # distinct(tempo_final_date)

#distinct(data_overlap_split_qname[[1]][['tempo_final_date']])

#semi_join(data_overlap_split_qname[[1]], data_overlap_split_qname[[2]], by=c("tempo_final_date", "temp

#top_queryid_qname %>%
  #filter(query_id )
 # top_queryid_qname[['query_id']]
```