AuthLog

Robert Baranic

2023-05-05

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(stringr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
getwd()
```

```
## [1] "/Users/bobbybaranic/Documents/UCD 22-23/SQ 2023/STA 141B/Project 2"
```

```
list.files()
```

```
## [1] "AuthLog.pdf"    "AuthLog.Rmd"    "getCaptures.R"  "logs.pdf"
## [5] "MergedAuth.log"
```

```
source_url("https://raw.githubusercontent.com/duncantl/ST141B_S23/main/Data/Weblogs/getCaptures.R")
```

```
## i SHA-1 hash of file is "48568419b845166b3f4eefbbe762faeb66e362c9"
```

```
lines = readLines("MergedAuth.log")
```

```
## Warning in readLines("MergedAuth.log"): incomplete final line found on
## 'MergedAuth.log'
```

```
head(lines, n = 20)
```

```
##  [1] ""
##  [2] "# auth.log"
##  [3] "Nov 30 06:39:00 ip-172-31-27-153 CRON[21882]: pam_unix(cron:session): session closed for user r
##  [4] "Nov 30 06:47:01 ip-172-31-27-153 CRON[22087]: pam_unix(cron:session): session opened for user r
##  [5] "Nov 30 06:47:03 ip-172-31-27-153 CRON[22087]: pam_unix(cron:session): session closed for user r
##  [6] "Nov 30 07:07:14 ip-172-31-27-153 sshd[22116]: Connection closed by 122.225.103.87 [preauth]"
##  [7] "Nov 30 07:07:35 ip-172-31-27-153 sshd[22118]: Connection closed by 122.225.103.87 [preauth]"
##  [8] "Nov 30 07:08:13 ip-172-31-27-153 sshd[22120]: Connection closed by 122.225.103.87 [preauth]"
##  [9] "Nov 30 07:17:01 ip-172-31-27-153 CRON[22125]: pam_unix(cron:session): session opened for user r
## [10] "Nov 30 07:17:01 ip-172-31-27-153 CRON[22125]: pam_unix(cron:session): session closed for user r
## [11] "Nov 30 08:17:01 ip-172-31-27-153 CRON[22172]: pam_unix(cron:session): session opened for user r
## [12] "Nov 30 08:17:01 ip-172-31-27-153 CRON[22172]: pam_unix(cron:session): session closed for user r
## [13] "Nov 30 08:42:04 ip-172-31-27-153 sshd[22182]: Invalid user admin from 187.12.249.74"
## [14] "Nov 30 08:42:04 ip-172-31-27-153 sshd[22182]: input_userauth_request: invalid user admin [preau
## [15] "Nov 30 08:42:04 ip-172-31-27-153 sshd[22182]: Received disconnect from 187.12.249.74: 11: Bye B
## [16] "Nov 30 08:42:14 ip-172-31-27-153 sshd[22184]: Did not receive identification string from 187.12
## [17] "Nov 30 09:17:01 ip-172-31-27-153 CRON[22214]: pam_unix(cron:session): session opened for user r
## [18] "Nov 30 09:17:01 ip-172-31-27-153 CRON[22214]: pam_unix(cron:session): session closed for user r
## [19] "Nov 30 09:22:03 ip-172-31-27-153 sshd[22218]: Did not receive identification string from 196.20
## [20] "Nov 30 10:17:01 ip-172-31-27-153 CRON[22251]: pam_unix(cron:session): session opened for user r
```

First step is to read in the data as a table. It appears to be a unique form, so the best way to pull data out would be to use regex and capture groups to read it into a dataframe. These were the iterative processes used to obtain the regex below. Each part was added one at a time through gregexpr. Same regex is fed to grepl so we can table the results to see how many lines it matched. I then explore the lines not matched so I can update the expression until everything is matched.

```
rx = "^(?P<time>[A-Za-z]{3}[ ]+[0-9]+ [0-9:]+)
↪  (?P<ip>[A-Za-z0-9]+-[0-9\\-]+|combo|LabSZ|authorMacBook-Pro)
↪  (?P<app>[A-Za-z0-9\\-\\(\\)\\_\\. ]+(?=\\[|:))(?P<pid>\\[[0-9]+\\]:|\\[[0-9]+\\]|
↪  |:)(?P<message>.*$)"

x = gregexpr(rx, lines, perl = T)

y = grepl(rx, lines, perl = T)

table(y)
```

```
## y
## FALSE  TRUE
##     8 99960
```

```
head(lines[!y], n = 20)
```

```
## [1] ""                              "# auth.log"
## [3] ""                              "# auth2.log"
## [5] ""                              "# loghub/Linux/Linux_2k.log"
## [7] "# loghub/Mac/Mac_2k.log"       "# loghub/OpenSSH/SSH_2k.log"
```

```
tail(lines[!y], n = 20)
```

```
## [1] ""                              "# auth.log"
## [3] ""                              "# auth2.log"
## [5] ""                              "# loghub/Linux/Linux_2k.log"
## [7] "# loghub/Mac/Mac_2k.log"       "# loghub/OpenSSH/SSH_2k.log"
```

Now, we must clean up blank lines. Thus, only lines that don't match any of the regex are the start of the log files.

```
lines = lines[!lines == ""]
head(lines)
```

```
## [1] "# auth.log"
## [2] "Nov 30 06:39:00 ip-172-31-27-153 CRON[21882]: pam_unix(cron:session): session closed for user r
## [3] "Nov 30 06:47:01 ip-172-31-27-153 CRON[22087]: pam_unix(cron:session): session opened for user r
## [4] "Nov 30 06:47:03 ip-172-31-27-153 CRON[22087]: pam_unix(cron:session): session closed for user r
## [5] "Nov 30 07:07:14 ip-172-31-27-153 sshd[22116]: Connection closed by 122.225.103.87 [preauth]"
## [6] "Nov 30 07:07:35 ip-172-31-27-153 sshd[22118]: Connection closed by 122.225.103.87 [preauth]"
```

Next, we split the lines by the log file. All five log files are headed by "#" at the beginning of the line so they are easy to split on.

```
grep("^#", lines)
```

```
## [1]     1 86841 93963 95964 97965
```

```
logStart = grepl("^#", lines)
table(logStart)
```

```
## logStart
## FALSE  TRUE
## 99960     5
```

```
splitter = cumsum(logStart)

lines2 = split(lines, splitter)
```

Next, we must use the regex to put this file into a dataframe. I was running into some errors with GetCapture(), so here is the alternative I found:

```
head(regmatches(lines[!logStart], regexec(rx, lines[!logStart], perl = T)))
```

```
## [[1]]
## [1] "Nov 30 06:39:00 ip-172-31-27-153 CRON[21882]: pam_unix(cron:session): session closed for user r
## [2] "Nov 30 06:39:00"
## [3] "ip-172-31-27-153"
## [4] "CRON"
## [5] "[21882]:"
## [6] " pam_unix(cron:session): session closed for user root"
##
```

```
## [[2]]
## [1] "Nov 30 06:47:01 ip-172-31-27-153 CRON[22087]: pam_unix(cron:session): session opened for user r
## [2] "Nov 30 06:47:01"
## [3] "ip-172-31-27-153"
## [4] "CRON"
## [5] "[22087]:"
## [6] " pam_unix(cron:session): session opened for user root by (uid=0)"
##
## [[3]]
## [1] "Nov 30 06:47:03 ip-172-31-27-153 CRON[22087]: pam_unix(cron:session): session closed for user r
## [2] "Nov 30 06:47:03"
## [3] "ip-172-31-27-153"
## [4] "CRON"
## [5] "[22087]:"
## [6] " pam_unix(cron:session): session closed for user root"
##
## [[4]]
## [1] "Nov 30 07:07:14 ip-172-31-27-153 sshd[22116]: Connection closed by 122.225.103.87 [preauth]"
## [2] "Nov 30 07:07:14"
## [3] "ip-172-31-27-153"
## [4] "sshd"
## [5] "[22116]:"
## [6] " Connection closed by 122.225.103.87 [preauth]"
##
## [[5]]
## [1] "Nov 30 07:07:35 ip-172-31-27-153 sshd[22118]: Connection closed by 122.225.103.87 [preauth]"
## [2] "Nov 30 07:07:35"
## [3] "ip-172-31-27-153"
## [4] "sshd"
## [5] "[22118]:"
## [6] " Connection closed by 122.225.103.87 [preauth]"
##
## [[6]]
## [1] "Nov 30 07:08:13 ip-172-31-27-153 sshd[22120]: Connection closed by 122.225.103.87 [preauth]"
## [2] "Nov 30 07:08:13"
## [3] "ip-172-31-27-153"
## [4] "sshd"
## [5] "[22120]:"
## [6] " Connection closed by 122.225.103.87 [preauth]"
```

```
#https://stackoverflow.com/questions/952275/regex-group-capture-in-r-with-multiple-capture-groups
```

Using do.call(rbind), we coerce the list into a dataframe, so now we must clean it.

```
regexList = regmatches(lines[!logStart], regexec(rx, lines[!logStart], perl = T))
df = as.data.frame(do.call(rbind, regexList))
```

The first thing I want to do is change the first column to the log name. The first column is currently the whole line that is matched with regmatches, so it will not be used. The splitter used originally has 99965 length since it includes the lines with the log names. The dataframe has 99960 lines since it does not include these lines. In order to get a length of 99960 I did some unusual manipulation so that we get a vector of 99960 for the splitter such that we can subset the lines correctly with the corresponding log file.

```
sum(sapply(lines2, length))
```

```
## [1] 99965
```

```
logname = splitter - 5 * logStart
logname = logname[logname >= 1]

df$V1 = lines[logStart][logname]
```

Here is validation that all valid PIDs are numbers. Below cleans up the PIDs such that there are no [] or : in the entry. When I table the entries that are not [0-9]+ we get 946 empty strings showing that the rest are numbers. Then we use as.numeric to convert them to numbers and the empty strings are NA.

```
colnames(df) = c("logFile", "date-time", "loggingHost", "app", "PID", "message")

df$PID = gsub("\\[|\\]|\\:", "", df$PID)
table(df$PID[!grepl("[0-9]+", df$PID)])
```

```
##
##
## 946
```

```
df$PID = as.numeric(df$PID)

df$message = trimws(df$message, "left")
```

Here is validation for the number of lines in each log file.

```
table(df$logFile)
```

```
##
##                   # auth.log                  # auth2.log
##                       86839                       7121
## # loghub/Linux/Linux_2k.log     # loghub/Mac/Mac_2k.log
##                        2000                       2000
## # loghub/OpenSSH/SSH_2k.log
##                        2000
```

To make finding the range of dates easier, we will convert the date and time to POSIXct first, then explore. POSIXct defaults to putting the year as 2023 even though there is not a date in the log files. I do not believe this will impact the exploration of dates. Below is the min and max dates for the total log file and verification that there are no NA values.

```
df$`date-time` = as.POSIXct(strptime(df$`date-time`, "%b %d %H:%M:%S"))

sum(is.na(df$`date-time`))
```

```
## [1] 0
```

```
min(df$`date-time`)
```

```
## [1] "2023-03-27 13:06:56 PDT"
```

```
max(df$`date-time`)
```

```
## [1] "2023-12-31 22:27:48 PST"
```

Below is date range for auth.log

```
min(df$`date-time`[df$logFile == "# auth.log"])
```

```
## [1] "2023-11-30 06:39:00 PST"
```

```
max(df$`date-time`[df$logFile == "# auth.log"])
```

```
## [1] "2023-12-31 22:27:48 PST"
```

```
max(df$`date-time`[df$logFile == "# auth.log"]) - min(df$`date-time`[df$logFile == "#
↪  auth.log"])
```

```
## Time difference of 31.65889 days
```

Below is date range for auth2.log

```
min(df$`date-time`[df$logFile == "# auth2.log"])
```

```
## [1] "2023-03-27 13:06:56 PDT"
```

```
max(df$`date-time`[df$logFile == "# auth2.log"])
```

```
## [1] "2023-04-20 14:14:29 PDT"
```

```
max(df$`date-time`[df$logFile == "# auth2.log"]) - min(df$`date-time`[df$logFile == "#
↪  auth2.log"])
```

```
## Time difference of 24.04691 days
```

Below is date range for loghub/Linux/Linux_2k.log

```
min(df$`date-time`[df$logFile == "# loghub/Linux/Linux_2k.log"])
```

```
## [1] "2023-06-14 15:16:01 PDT"
```

```
max(df$`date-time`[df$logFile == "# loghub/Linux/Linux_2k.log"])
```

```
## [1] "2023-07-27 14:42:00 PDT"
```

```
max(df$`date-time`[df$logFile == "# loghub/Linux/Linux_2k.log"]) -
↪  min(df$`date-time`[df$logFile == "# loghub/Linux/Linux_2k.log"])
```

```
## Time difference of 42.97638 days
```

Below is date range for loghub/Mac/Mac_2k.log

```
min(df$`date-time`[df$logFile == "# loghub/Mac/Mac_2k.log"])
```

```
## [1] "2023-07-01 09:00:55 PDT"
```

```
max(df$`date-time`[df$logFile == "# loghub/Mac/Mac_2k.log"])
```

```
## [1] "2023-07-08 08:10:46 PDT"
```

```
max(df$`date-time`[df$logFile == "# loghub/Mac/Mac_2k.log"]) -
↪  min(df$`date-time`[df$logFile == "# loghub/Mac/Mac_2k.log"])
```

```
## Time difference of 6.965174 days
```

Below is date range for loghub/OpenSSH/SSH_2k.log

```
min(df$`date-time`[df$logFile == "# loghub/OpenSSH/SSH_2k.log"])
```

```
## [1] "2023-12-10 06:55:46 PST"
```

```
max(df$`date-time`[df$logFile == "# loghub/OpenSSH/SSH_2k.log"])
```

```
## [1] "2023-12-10 11:04:45 PST"
```

```
max(df$`date-time`[df$logFile == "# loghub/OpenSSH/SSH_2k.log"]) -
↪  min(df$`date-time`[df$logFile == "# loghub/OpenSSH/SSH_2k.log"])
```

```
## Time difference of 4.149722 hours
```

Now we will explore the application names. To check if the applications have number we will use grepl. It appears that numbers only appear as version numbers.

```
df$app[grepl("[0-9]", df$app)]
```

```
##  [1] "syslogd 1.4.1"        "syslogd 1.4.1"        "syslogd 1.4.1"
##  [4] "syslogd 1.4.1"        "syslogd 1.4.1"        "syslogd 1.4.1"
##  [7] "syslogd 1.4.1"        "BezelServices 255.10" "BezelServices 255.10"
## [10] "BezelServices 255.10" "BezelServices 255.10"
```

Next, we will explore the logging host. All have the same logging host except for loghub/Mac/Mac_2k.log which has many different logging hosts.

```
table(df$loggingHost[df$logFile == "# auth.log"])
```

```
##
## ip-172-31-27-153
##            86839
```

```
table(df$loggingHost[df$logFile == "# auth2.log"])
```

```
##
## ip-10-77-20-248
##            7121
```

```
table(df$loggingHost[df$logFile == "# loghub/Linux/Linux_2k.log"])
```

```
##
## combo
##  2000
```

```
table(df$loggingHost[df$logFile == "# loghub/Mac/Mac_2k.log"])
```

```
##
##   airbears2-10-142-108-38  airbears2-10-142-110-255         authorMacBook-Pro
##                        15                        79                       554
## calvisitor-10-105-160-179 calvisitor-10-105-160-181 calvisitor-10-105-160-184
##                        19                         6                        39
## calvisitor-10-105-160-205 calvisitor-10-105-160-210  calvisitor-10-105-160-22
##                        30                         9                         7
## calvisitor-10-105-160-226 calvisitor-10-105-160-237  calvisitor-10-105-160-37
##                        17                        53                        12
##  calvisitor-10-105-160-47  calvisitor-10-105-160-85  calvisitor-10-105-160-95
##                         6                        83                       140
## calvisitor-10-105-161-176 calvisitor-10-105-161-225 calvisitor-10-105-161-231
##                         6                        16                         2
##  calvisitor-10-105-161-77 calvisitor-10-105-162-105 calvisitor-10-105-162-107
##                         2                       338                        13
## calvisitor-10-105-162-108 calvisitor-10-105-162-124 calvisitor-10-105-162-138
##                         4                        29                         3
## calvisitor-10-105-162-175 calvisitor-10-105-162-178 calvisitor-10-105-162-211
##                         4                       256                         3
## calvisitor-10-105-162-228  calvisitor-10-105-162-32  calvisitor-10-105-162-81
##                         5                        27                         3
##  calvisitor-10-105-162-98  calvisitor-10-105-163-10 calvisitor-10-105-163-147
```

```
##                         8                        34                         5
## calvisitor-10-105-163-168 calvisitor-10-105-163-202 calvisitor-10-105-163-253
##                         4                       137                        26
##  calvisitor-10-105-163-28   calvisitor-10-105-163-9
##                         2                         4
```

```
table(df$loggingHost[df$logFile == "# loghub/OpenSSH/SSH_2k.log"])
```

```
##
## LabSZ
##  2000
```

Lastly, we will explore the frequency of apps used by the logging hosts.

```
table(df$app[df$loggingHost == "ip-172-31-27-153"])
```

```
##
##  CRON  sshd
##  1593 85246
```

```
table(df$app[df$loggingHost == unique(df$loggingHost[df$logFile == "# auth2.log"])])
```

```
##
##       chpasswd           CRON       groupadd           sshd             su
##            417           1264              3           4095             45
##           sudo        systemd systemd-logind        useradd
##            557            238            452             50
```

```
table(df$app[df$loggingHost == unique(df$loggingHost[df$logFile == "#
↪  loghub/Linux/Linux_2k.log"])])
```

```
##
##         -- root       bluetooth            cups            ftpd      gdm-binary
##               1               2              12             916               1
##   gdm(pam_unix)             gpm            hcid      irqbalance          kernel
##               2               2               1               1              76
##         klogind login(pam_unix)       logrotate           named         network
##              46               2              43              16               2
##         nfslock         portmap          random              rc       rpc.statd
##               1               1               1               1               1
##       rpcidmapd            sdpd           snmpd  sshd(pam_unix)    su(pam_unix)
##               1               1               1             677             172
##          sysctl          syslog   syslogd 1.4.1            udev          xinetd
##               1               2               7               8               2
```

```
table(df$app[df$logFile == "# loghub/Mac/Mac_2k.log"])
```

```
##
##                         AddressBookSourceSync
```

```
##                                         14
##                                AirPlayUIAgent
##                                          1
##                           BezelServices 255.10
##                                          4
##                                      blued
##                                         10
##                                CalendarAgent
##                                         10
##                                       cdpd
##                                          4
##                                ChromeExistion
##                                          6
##                                     cloudd
##                                          5
## com.apple.AddressBook.ContactsAccountsService
##                                          2
##  com.apple.AddressBook.InternetAccountsBridge
##                                         59
##                           com.apple.CDScheduler
##                                         39
##                                com.apple.cts
##                                        166
##                                com.apple.geod
##                                          3
##                     com.apple.ncplugin.weather
##                                          2
##                  com.apple.ncplugin.WorldClock
##                                          1
##                       com.apple.SecurityServer
##                                          3
##                    com.apple.WebKit.Networking
##                                          2
##                    com.apple.WebKit.WebContent
##                                         60
##                         com.apple.xpc.launchd
##                                         19
##                                   CommCenter
##                                          1
##                                     configd
##                                         24
##                                  corecaptured
##                                        158
##                     CrashReporterSupportHelper
##                                          1
##                                        Dock
##                                         14
##                                     Dropbox
##                                          4
##                                      garcon
##                                          2
##                                Google Chrome
##                                          2
##                       GoogleSoftwareUpdateAgent
```

```
##                                     21
##                           GPUToolsAgent
##                                      1
##                                   hidd
##                                      1
##                         iconservicesagent
##                                     20
##                         identityservicesd
##                                      7
##                                 imagent
##                                      4
##                                 kernel
##                                    775
##                                 ksfetch
##                                      8
##                                locationd
##                                     60
##                               loginwindow
##                                      2
##                                   Mail
##                                      4
##                           mDNSResponder
##                                      9
##                                    mds
##                                      1
##                                mdworker
##                                      1
##                           Microsoft Word
##                                     72
##                                netbiosd
##                                      3
##                             NeteaseMusic
##                                      1
##                                networkd
##                                     43
##                                   ntpd
##                                      8
##                                    pkd
##                                      1
##                                 Preview
##                                      5
##                                    QQ
##                                     75
##                               quicklookd
##                                     14
##                         QuickLookSatellite
##                                      3
##                                   Safari
##                                     31
##                                sandboxd
##                                     35
##                                   SCIM
##                                      1
##                                   secd
```

```
##                                         16
##                                   sharingd
##                                         32
##                             SpotlightNetHelper
##                                          3
##                                   symptomsd
##                                         33
##                                    syslogd
##                                         12
##                                   taskgated
##                                          1
##                                      TCIM
##                                          1
##                                 UserEventAgent
##                                         23
##                                  VDCAssistant
##                                          1
##                                       WeChat
##                                         13
##                                  WindowServer
##                                         38
##                                 wirelessproxd
##                                          5
```

```
table(df$app[df$loggingHost == "LabSZ"])
```

```
##
## sshd
## 2000
```

Logins: valid logins from hosts:

```
validLogins = df$message[grepl("Connection from|Accepted|New session", df$message)]

table(do.call(rbind, regmatches(validLogins, regexec("(?<=for |user )[A-Za-z0-9\\_]+",
↪  validLogins, perl = T)))) #usernames
```

```
##
## elastic_user_0 elastic_user_1 elastic_user_2 elastic_user_3 elastic_user_4
##             58             50             24             22             34
## elastic_user_5 elastic_user_6 elastic_user_7 elastic_user_8 elastic_user_9
##             28             48             36             40             40
##           fztu         ubuntu
##              1             72
```

```
table(do.call(rbind, regmatches(validLogins, regexec("(?<=from )[0-9.]+", validLogins,
↪  perl = T)))) #ip
```

```
##
## 119.137.62.142      127.0.0.1  182.32.215.94 186.219.213.14  24.151.103.17
##              1              3              1              1             48
##  85.245.107.41   95.93.96.191
##            186              4
```

```
valid.ip = do.call(rbind, regmatches(validLogins, regexec("(?<=from )[0-9.]+",
↪  validLogins, perl = T)))
```

invalid logins: Since there are a lot, I will put them into a dataframe so we can keep track of the usernames and associated IPs

```
invalidLogins = df$message[grepl("^Invalid\\b |^error", df$message)]

invalid.user = as.data.frame(do.call(rbind, regmatches(invalidLogins, regexec("(?<=user
↪  |for )[A-Za-z0-9\\.\\_\\\\\\-]+", invalidLogins, perl = T))))#usernames

invalid.ip = do.call(rbind, regmatches(invalidLogins, regexec("(?<=from )[0-9\\.]+",
↪  invalidLogins, perl = T)))

invalid.user$ip = invalid.ip[grepl("(?<=user |for )[A-Za-z0-9\\.\\_\\\\\\-]+",
↪  invalidLogins, perl = T)]

head(invalid.user)
```

```
##          V1              ip
## 1     admin   187.12.249.74
## 2     admin 122.225.109.208
## 3     admin  124.205.250.51
## 4     guest  124.205.250.51
## 5   support  124.205.250.51
## 6 avconroot   218.26.11.118
```

```
length(table(invalid.user$ip)[table(invalid.user$ip) > 1]) #all ips with multiple logins
```

```
## [1] 1015
```

```
unique(invalid.user$ip[invalid.ip %in% valid.ip]) #invalid ips that were valid at some
↪  point
```

```
## [1] "85.245.107.41" "182.32.215.94"
```

```
x = sapply(unique(invalid.user$V1[table(invalid.user$V1) > 1]), function(x)
↪  length(unique(invalid.user$ip[invalid.user$V1 == x])))
x[x > 1] #all invalid users that used more than 1 ip
```

```
##           admin           guest         support       avconroot       webmaster
##             826             230             175               2               4
##        postgres          oracle            test             git          zabbix
##              20             179             189              16               9
##          apache            Test             ftp          system           jboss
##               7               6             159               7               7
##         webmail          nagios          apache2            boot        weblogic
##               7              17               6               8               9
##       guestuser            ftp1        sysadmin       cactiuser           squid
```

```
##              6              6              7              7              7
##             ip          cacti         tomcat            web            dff
##              2              8              9              7             12
##            123          nginx         zxin10           java         zhaowei
##              9              6              7              6              6
##          \\026            adm           ubnt         ftpuser             pi
##              2              2            186            323            219
##        PlcmSpIp           user        jenkins         hadoop       deployer
##            252            154              3              7              2
##         deploy       testuser        redmine      minecraft            www
##              3              2              5              4              6
##            vnc        biadmin          ankit           mike              b
##              3              2              2            144              3
##           user1           ajay       zhangyan         123456           bash
##              2              2              9              8              8
##           r00t          resin         apache1          httpd     nagiosuser
##              5              4              4              4              4
##         nologin           ftpd         wangyi       webadmin         guestx
##              4              5              4              7              4
##          httpd2       httpdocs    nagiosadmin         upload        ibmuser
##              4              4              6              4              3
##          hduser         vyatta            nan         sm0k3y          rsync
##              3            223              5              2              3
##           xbian            dev              a           xbmc          plesk
##             207              3              6            184              2
##         tomcat7           alex       db2inst1       homepage          mysql
##              2              5              2              2              4
##         javaprg       username        gyaseen       nfsnobod           dede
##              4              4              3              4              2
##            User     Administra         D-Link          sales            log
##              4              4            280              7            260
##           debug oooooooooooooo          karaf          arbab        dreamer
##             267              2            225            200            205
##         default   administrator           Conf          Admin         Menara
##             215              3              4              8              4
##            gusr          lihan         syncro            app         Sorin
##              5              2              2              3              2
##            adam        ubuntu1          cisco        bwadmin           info
##              2              2              4              3            150
##        uploader          agata      marketing           bill           xymon
##             128            160            142            154              2
##           kevin           temp         portal        manager       operator
##              2              3              2              3              4
##         postfix           root        invalid              0           1234
##              2             82             64              4              3
##         monitor            api          ghost         ubuntu          test2
##              3              2              3              2              2
```

```
max.ip.lines = df$message[grepl("\\bmaximum\\b", df$message)]
max.ip = do.call(rbind, regmatches(max.ip.lines, regexec("(?<=from )[0-9\\.]+",
↪  max.ip.lines, perl = T)))#ips with maximum authentication attempts
head(max.ip)
```

```
##      [,1]
```

```
## [1,] "122.176.37.221"
## [2,] "95.152.57.58"
## [3,] "90.144.183.19"
## [4,] "186.128.152.44"
## [5,] "201.177.23.130"
## [6,] "190.178.62.6"
```

Sudo commands:

```
sudoLines = lines[grepl("sudo", lines)]
tail(sudoLines, n = 20)
```

```
##  [1] "Apr 10 11:55:03 ip-10-77-20-248 sudo:   ubuntu : TTY=pts/0 ; PWD=/opt/filebeat/filebeat-6.0.0-a
##  [2] "Apr 10 11:55:03 ip-10-77-20-248 sudo: pam_unix(sudo:session): session opened for user root by u
##  [3] "Apr 10 11:55:24 ip-10-77-20-248 sudo: pam_unix(sudo:session): session closed for user root"
##  [4] "Apr 10 11:55:25 ip-10-77-20-248 sudo:   ubuntu : TTY=pts/0 ; PWD=/opt/filebeat/filebeat-6.0.0-a
##  [5] "Apr 10 11:55:25 ip-10-77-20-248 sudo: pam_unix(sudo:session): session opened for user root by u
##  [6] "Apr 10 11:55:26 ip-10-77-20-248 sudo: pam_unix(sudo:session): session closed for user root"
##  [7] "Apr 10 11:55:27 ip-10-77-20-248 sudo:   ubuntu : TTY=pts/0 ; PWD=/opt/filebeat/filebeat-6.0.0-a
##  [8] "Apr 10 11:55:27 ip-10-77-20-248 sudo: pam_unix(sudo:session): session opened for user root by u
##  [9] "Apr 10 11:56:23 ip-10-77-20-248 sudo: pam_unix(sudo:session): session closed for user root"
## [10] "Apr 10 11:56:25 ip-10-77-20-248 sudo:   ubuntu : TTY=pts/0 ; PWD=/opt/filebeat/filebeat-6.0.0-a
## [11] "Apr 10 11:56:25 ip-10-77-20-248 sudo: pam_unix(sudo:session): session opened for user root by u
## [12] "Apr 10 11:56:31 ip-10-77-20-248 sudo: pam_unix(sudo:session): session closed for user root"
## [13] "Apr 10 11:56:37 ip-10-77-20-248 sudo:   ubuntu : TTY=pts/0 ; PWD=/opt/filebeat/filebeat-6.0.0-a
## [14] "Apr 10 11:56:37 ip-10-77-20-248 sudo: pam_unix(sudo:session): session opened for user root by u
## [15] "Apr 10 12:37:47 ip-10-77-20-248 sudo:   ubuntu : TTY=pts/1 ; PWD=/home/ubuntu/misc_scripts ; US
## [16] "Apr 10 12:37:47 ip-10-77-20-248 sudo: pam_unix(sudo:session): session opened for user root by u
## [17] "Apr 10 12:59:47 ip-10-77-20-248 sudo: pam_unix(sudo:session): session closed for user root"
## [18] "Apr 10 14:11:51 ip-10-77-20-248 sudo: pam_unix(sudo:session): session closed for user root"
## [19] "Apr 10 15:32:59 ip-10-77-20-248 sudo:   ubuntu : TTY=pts/0 ; PWD=/opt/filebeat/filebeat-6.0.0-a
## [20] "Apr 10 15:32:59 ip-10-77-20-248 sudo: pam_unix(sudo:session): session opened for user root by u
```

```
sudoLines2 = sudoLines[grepl("(?<=COMMAND\\=)[A-Za-z0-9\\.\\-\\=\\,\\_\\:/ ]+$",
↪  sudoLines, perl = T)]

sudoCommands = as.data.frame(do.call(rbind, regmatches(sudoLines,
↪  regexec("(?<=COMMAND\\=)[A-Za-z0-9\\.\\-\\=\\,\\_\\:/ ]+$", sudoLines, perl = T))))
sudoCommands$user = do.call(rbind, regmatches(sudoLines,
↪  regexec("(?<=USER\\=)[A-Za-z0-9\\.\\-\\=/ ]+(?=;)", sudoLines, perl = T)))

sudoCommands$ip = do.call(rbind, regmatches(sudoLines2, regexec("ip[0-9\\-]+\\b",
↪  sudoLines2, perl = T)))
colnames(sudoCommands) = c("executable", "user", "ip")
sudoCommands$executable = trimws(sudoCommands$executable, "left")
head(sudoCommands$executable)
```

```
## [1] "/usr/bin/curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-5.2.2-amd64.d
## [2] "/usr/bin/apt-key add -"
## [3] "/usr/bin/apt-get install apt-transport-https"
## [4] "/usr/bin/tee -a /etc/apt/sources.list.d/elastic-5.x.list"
## [5] "/usr/bin/apt-get update"
## [6] "/usr/bin/apt-get install filebeat"
```

```
head(sudoCommands)
```

```
##                                                                                                                executable
## 1 /usr/bin/curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-5.2.2-amd64.deb
## 2                                                                                                    /usr/bin/apt-key add -
## 3                                                                                /usr/bin/apt-get install apt-transport-https
## 4                                                                  /usr/bin/tee -a /etc/apt/sources.list.d/elastic-5.x.list
## 5                                                                                                    /usr/bin/apt-get update
## 6                                                                                /usr/bin/apt-get install filebeat
##    user              ip
## 1 root  ip-10-77-20-248
## 2 root  ip-10-77-20-248
## 3 root  ip-10-77-20-248
## 4 root  ip-10-77-20-248
## 5 root  ip-10-77-20-248
## 6 root  ip-10-77-20-248
```