Contenido

[1 Goals 2](#_Toc362620781)

[1.1 Analysis Goals 2](#_Toc362620782)

[1.2 Exploration Goals 2](#_Toc362620783)

[2 Data selection 2](#_Toc362620784)

[2.1 General dataset 2](#_Toc362620785)

[2.2 Single logs 2](#_Toc362620786)

[2.2.1 Web Server 3](#_Toc362620787)

[2.2.1.1 SANPARTICULARES 09 3](#_Toc362620788)

[2.2.1.2 SANPARTICULARES 10 9](#_Toc362620789)

[2.2.1.3 Cleaning up 10](#_Toc362620790)

[2.2.1.4 Combining logs 10](#_Toc362620791)

[2.2.1.5 Request Rate 11](#_Toc362620792)

[2.2.2 Firewall 14](#_Toc362620793)

[2.2.2.1 XML Schema 15](#_Toc362620794)

[2.2.2.2 Mulan/traffic-drop.log 17](#_Toc362620795)

[2.2.2.3 Hopper/traffic-drop.log 17](#_Toc362620796)

[2.2.2.4 Hopper/traffic-accept.log 25](#_Toc362620797)

[2.2.2.5 Mulan/traffic-accept.log 25](#_Toc362620798)

[2.2.2.6 Tmp 25](#_Toc362620799)

# Goals

## Analysis Goals

* Predict future failure events from logs

## Exploration Goals

* Identify primary prediction logs
  + With significant failures
  + With event patterns
* Identify secondary prediction logs
  + Describe systems associated with those of primary logs
  + Over same period of time of primary logs
  + Can be analyzed together with primary logs to identify patterns

# Data selection

## General dataset

Described in [Filename]

Data set is too large, broad (200+ unrelated logs) to analyze as a single entity.

This is not the data set where our final analysis and algorithms will be drawn from

* But it has elements in common
* Explore log types that will be in future datasets

## Single logs

Firewall and webserver logs will be present in future dataset

### Web Server

Available WebServer logs:

* WEB
* production\IHS
  + AGENDACOMERCIAL
    - spscapbks05-adm.san.corp
    - spscapbks06-adm.san.corp
    - spscapbks07-adm.san.corp
    - spscapbks08-adm.san.corp
  + ESTRUCTPART
    - bvsalstinp01-adm.wt.santander.corp
    - bvsalstinp03-adm.wt.santander.corp
  + ESTRUCTPARTOCU
    - bvsalstinp01-adm.wt.santander.corp
    - bvsalstinp03-adm.wt.santander.corp
  + SANPARDNI
    - bvsalwparp03-adm.wt.santander.corp
  + SANPARDNIOCU
    - bvsalwparp03-adm.wt.santander.corp
    - bvsalwparp04-adm.wt.santander.corp
    - bvsalwparp05-adm.wt.santander.corp
    - bvsalwparp06-adm.wt.santander.corp
    - bvsalwparp07-adm.wt.santander.corp
    - bvsalwparp08-adm.wt.santander.corp
  + SANPARTICULARES
    - bvsalwparp09-adm.wt.santander.corp
    - bvsalwparp10-adm.wt.santander.corp
    - bvsalwparp11-adm.wt.santander.corp
    - bvsalwparp12-adm.wt.santander.corp
    - bvsalwparp13-adm.wt.santander.corp
    - bvsalwparp14-adm.wt.santander.corp
  + SANPARTICULARESOCU
    - bvsalwparp03-adm.wt.santander.corp
    - bvsalwparp08-adm.wt.santander.corp
  + SAPORTALBROKER
    - bosaswebp02-adm.santander.corp
    - bvsalportp01-adm.wt.santander.corp
    - bvsalportp02-adm.wt.santander.corp
  + SANPORTALINFOPRO
    - bosaswebp02-adm.santander.corp
    - bvsalportp01-adm.wt.santander.corp
    - bvsalportp02-adm.wt.santander.corp
  + SANSUPERN
    - bosaswebp02-adm.santander.corp

#### SANPARTICULARES 09

We will start with SANPARTICULARES/ bvsalwparp09-adm.wt.santander.corp

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\LogicalEntity\WEB-IHS\SANPARTICULARES\bvsalwparp09-adm.wt.santander.corp-02\access\_1.log

Route (formatted for R):

C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\LogicalEntity\\WEB-IHS\\SANPARTICULARES\\bvsalwparp09-adm.wt.santander.corp-02\\access\_1.log

File details

* 166MB
* 270k Lines
* 1 day (Jan-01-13)
* Sample line:

Jan 01 00:00:45 bvsalwparp09-adm.wt.santander.corp Santander.Santander.LogicalEntity.WEB.production.IHS.access.SANPARTICULARES: 41.108.71.217 - - [30/Dec/2012:17:13:54 +0100] "GET /Estatico/Globales/V166/Bhtcs/Internet/AT/DatosContexto\_mM.bjs HTTP/1.1" 200 2596 "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)" "OPERPAR1\_JSESSIONID=00018udFL9uOUJgvB5-Drw5erSB:16urjsn5s;PARTICULARES\_BKS\_JSESSIONID=-;OPERPAR2\_JSESSIONID=-;VENOPERPAR\_JSESSIONID=0001IfcjhtAnfYqj2oFsaTUYERY:16v0956k6;SUPFPA\_ENS\_JSESSIONID=00001KQs-l5gYjJ\_5CTn5HNlseK:16vniv9ee" 756

##### XML Schema

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\0-log-tables\tables\table-ihs-santander-sanparticulares-access.xml

Formatted Route:

C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\0-log-tables\\tables\\table-ihs-santander-sanparticulares-access.xml

Content:

<?xml version="1.0" encoding="ISO-8859-1"?>

<table name="IHS-Santander-SANPARTICULARES-access">

<source>IHS-Santander-SANPARTICULARES-access</source>

<path type='offline'>\\\\vmwtbitafiler01.ad.produban.corp\\cuaderno2\$\\bitacora\\cuaderno\\{YEAR}\\{MONTH}\\{DAY}\\(Santander)\\(Santander)\\(LogicalEntity)\\(WEB)\\(\w+)\\IHS\\(SANPARTICULARES)\\(.\*?)\\(.\*?)\\access.log(\.gz|)</path>

<log>(.{15})\s+(\S+)\s+([^:]+):\s+(\S+)\s+(\S+)\s+(\S+)\s\[([^\]]\*)\]\s&quot;(\S\*)\s(\S\*)\s(\S\*)&quot; (\d+) (\d+) &quot;(.+?)&quot;.\*?</log>

<categories>

<category schema="LOG\_MANAGEMENT" name="HTTP\_SERVICE"/>

<category name="IHS-Santander"/>

</categories>

<columns>

<column name="raw" from="log" group="0" parser="STRING"/>

<column name="event\_date" from="log" group="1" parser="syslog-date" eventdate="true"/>

<column name="http\_request\_url" from="log" group="9" parser="STRING"/>

<column name="http\_protocol" from="log" group="10" parser="STRING"/>

<column name="http\_response\_code" from="log" group="11" parser="INTEGER"/>

<column name="http\_response\_size" from="log" group="12" parser="INTEGER"/>

<column name="ip" from="log" group="2" parser="STRING"/>

<column name="source\_ip" from="log" group="4" parser="STRING"/>

<column name="http\_request\_domain" from="log" group="9" parser="STRING"/>

<column name="generated\_date" from="log" group="7" parser="apache-date"/>

<column name="http\_user\_agent" from="log" group="13" parser="STRING"/>

<column name="http\_method" from="log" group="8" parser="STRING"/>

<column name="program" from="log" group="3" parser="STRING"/>

<column name="hostname" from="path" group="10" parser="STRING"/>

<column name="company" from="path" group="4" parser="STRING"/>

<column name="vdc" from="path" group="5" parser="STRING"/>

<column name="category" from="path" group="6" parser="STRING"/>

<column name="type" from="path" group="7" parser="STRING"/>

<column name="functionalenvironment" from="path" group="8" parser="STRING"/>

<column name="instance" from="path" group="9" parser="STRING"/>

<column name="product\_vendor" from="expression" type="STRING">&apos;IBM&apos;</column>

<column name="product\_model" from="expression" type="STRING">&apos;IHS&apos;</column>

<column name="product\_version" from="expression" type="STRING">&apos;-&apos;</column>

<column name="asset" from="expression" type="STRING">&apos;-&apos;</column>

<column name="item" from="expression" type="STRING">&apos;-&apos;</column>

<column name="http\_port" from="expression" type="STRING">&apos;&apos;</column>

</columns>

</table>

##### Columns

The log has the following columns:

* raw
* event\_date
* http\_request\_url
* http\_protocol
* http\_response\_code
* http\_response\_size
* ip
* source\_ip
* http\_request\_domain
* generated\_date
* http\_user\_agent
* http\_method
* program
* hostname
* company
* vdc
* category
* type
* functionalenvironment
* instance
* product\_vendor
* product\_model
* product\_version
* asset
* item
* http\_port

Not a lot of error prediction material. http\_response\_code could be useful, though, if we look for 4xx or 5xx responses.

##### R code

###### Preprocessing

Working directory

C:\Users\capelastegui\workspace\OFP\Santander-1\2-R\preprocess

C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\2-R\\preprocess

setwd('C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\2-R\\preprocess')

01-01-preprocess-web-sanpart.R

#Replace file, xmlFile as required

file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\LogicalEntity\\WEB-IHS\\SANPARTICULARES\\bvsalwparp09-adm.wt.santander.corp-02\\access\_1.log"

indic

xmlFile <- " C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\0-log-tables\\tables\\table-ihs-santander-sanparticulares-access.xml"

library(XML)

library(gsubfn)

x <- xmlTreeParse(xmlFile)

r <- xmlRoot(x)

l<- xmlSApply(r[["columns"]], xmlAttrs)

regex <- xmlValue(r[["log"]])

#Switch lines to read full log or summary

lines <- readLines(file, 10)

#lines <- readLines(file)

log<- data.frame(strapplyc(lines, regex, simplify = "rbind"))

myF <- function(x, name="name")

{x[[name]]}

vapply(l,myF, "", USE.NAMES=FALSE)

myF <- function(x, name="name") {if(name%in%names(x)) {x[[name]]} else "NA"}

#get names, groups, froms, types

froms <- vapply(l,myF, "", "from", USE.NAMES=FALSE)

groups <- vapply(l,myF, "", "group", USE.NAMES=FALSE)

names <- vapply(l,myF, "", "name", USE.NAMES=FALSE)

types <- vapply(l,myF, "", "parser", USE.NAMES=FALSE)

indices <- as.numeric(groups[froms=="log"])

indicesB <- indices[indices>0]

line1 <- log[1,]

line1a <- line1[indicesB]

names(line1a) <- names[froms=="log"][indices>0]

logA <- log[indicesB]

names(logA) <- names[froms=="log"][indices>0]

#Column procesing

logA$http\_response\_size <- as.numeric(logA$http\_response\_size)

#Date

Sys.setlocale("LC\_TIME", "english")

event\_date2 <- strptime(logA$event\_date, "%b %d %H:%M:%S")

generated\_date2 <- strptime(logA$generated\_date, "%d/%b/%Y:%H:%M:%S")

logA$event\_date <- event\_date2

logA$generated\_date <- generated\_date2

###### tmp: Playing with R, preprocessing

rm(list = ls())

setwd('C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\2-R\\preprocess')

source("01-01-preprocess-web-sanpart.R")

#runs slowly, almost crashes

###### Note: on the processing of "expression" columns

Log table XMLs include fixed value columns, identified by "from=expression". For these, the value is a fixed value indicated in the XML. These columns can be safely ignored while analyzing individual logs, but will need to be processed when aggregating files.

###### Error code analysis

summary(logA$http\_response\_code)

200 206 302 403 404 416

234779 22 30 18 4473 1

No server error -> not very useful!

###### Date analysis

event\_date

Jan 01 00:00:45

%m %d %H:%M:%S

generated\_date

30/Dec/2012:17:13:54

Sys.setlocale("LC\_TIME", "english")

event\_date2 <- strptime(logA$event\_date, "%b %d %H:%M:%S")

generated\_date2 <- strptime(logA$generated\_date, "%d/%b/%Y:%H:%M:%S")

logA$event\_date <- event\_date2

logA$generated\_date <- generated\_date2

par(mfrow=c(2,1))

par(pch=1)

par(cex=0.5)

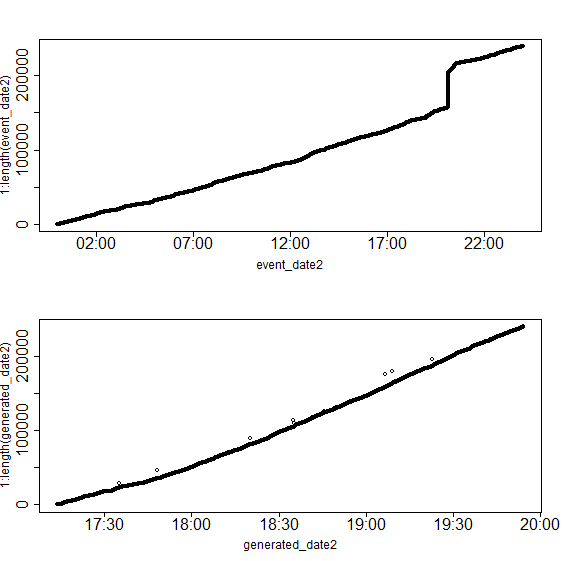
par(cex.axis=2)

par(cex.lab=1.5)

#par(col=rgb(0,0,0,0.005))

plot (event\_date2, 1:length(event\_date2))

plot (generated\_date2, 1:length(generated\_date2))



event\_date2[1]

[1] "2013-01-01 00:00:45"

event\_date2[length(event\_date2)]

[1] "2013-01-02 00:00:16"

generated\_date2[1]

[1] "2012-12-30 17:13:54"

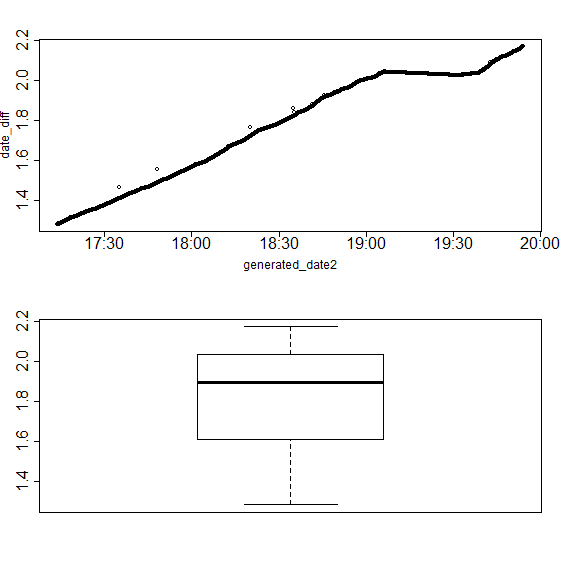
generated\_date2[length(generated\_date2)]

[1] "2012-12-30 19:54:00"

date\_diff <- event\_date2 - generated\_date2

plot(generated\_date2, date\_diff)

boxplot(as.numeric(date\_diff))



date\_diff[1]

Time difference of 1.282535 days

log$dateNum <- date3

log <- log[c("dateNum", "code", "source", "sevNum", "message")]

#plot(log2dateNum,log$sevNum)

###### Other

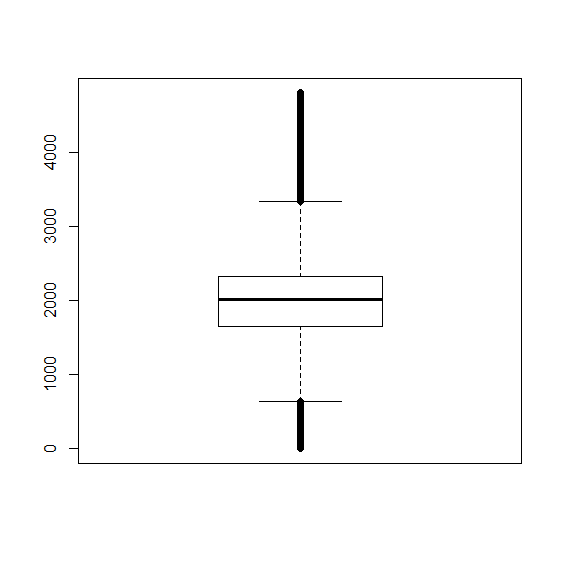
logA$http\_response\_size <- as.numeric(logA$http\_response\_size)

summary((logA$http\_response\_size))

Min. 1st Qu. Median Mean 3rd Qu. Max.

1 1653 2014 2100 2327 4797

boxplot((logA$http\_response\_size))



#### SANPARTICULARES 10

Another Web Server, just like the previous one

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\LogicalEntity\WEB-IHS\SANPARTICULARES\bvsalwparp10-adm.wt.santander.corp-02\access\_1.log

Route (formatted for R):

C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\LogicalEntity\\WEB-IHS\\SANPARTICULARES\\bvsalwparp09-adm.wt.santander.corp-02\\access\_1.log

##### R

#Replace file, xmlFile as required

file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\LogicalEntity\\WEB-IHS\\SANPARTICULARES\\bvsalwparp10-adm.wt.santander.corp-02\\access\_1.log"

01-01-preprocess-web-sanpart-10.R

#Replace file, xmlFile as required

#file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\LogicalEntity\\WEB-IHS\\SANPARTICULARES\\bvsalwparp10-adm.wt.santander.corp-02\\access\_1.log"

xmlFile <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\0-log-tables\\tables\\table-ihs-santander-sanparticulares-access.xml"

library(XML)

library(gsubfn)

x <- xmlTreeParse(xmlFile)

r <- xmlRoot(x)

l<- xmlSApply(r[["columns"]], xmlAttrs)

regex <- xmlValue(r[["log"]])

#Switch lines to read full log or summary

#lines <- readLines(file, 10)

lines <- readLines(file)

log<- data.frame(strapplyc(lines, regex, simplify = "rbind"))

myF <- function(x, name="name")

{x[[name]]}

vapply(l,myF, "", USE.NAMES=FALSE)

myF <- function(x, name="name") {if(name%in%names(x)) {x[[name]]} else "NA"}

#get names, groups, froms, types

froms <- vapply(l,myF, "", "from", USE.NAMES=FALSE)

groups <- vapply(l,myF, "", "group", USE.NAMES=FALSE)

names <- vapply(l,myF, "", "name", USE.NAMES=FALSE)

types <- vapply(l,myF, "", "parser", USE.NAMES=FALSE)

indices <- as.numeric(groups[froms=="log"])

indicesB <- indices[indices>0]

line1 <- log[1,]

line1a <- line1[indicesB]

names(line1a) <- names[froms=="log"][indices>0]

logA <- log[indicesB]

names(logA) <- names[froms=="log"][indices>0]

#Column procesing

logA$http\_response\_size <- as.numeric(logA$http\_response\_size)

#Date

Sys.setlocale("LC\_TIME", "english")

event\_date2 <- strptime(logA$event\_date, "%b %d %H:%M:%S")

generated\_date2 <- strptime(logA$generated\_date, "%d/%b/%Y:%H:%M:%S")

logA$event\_date <- event\_date2

logA$generated\_date <- generated\_date2

We load the script, and explore data

Same time pattern

#### Cleaning up

Remove redundant column

log9\_1 <- log9\_10 [!(colnames(log9\_10) %in% c("http\_request\_domain"))]

Change log names

library(plyr)

minilog09$program <- revalue(minilog09$program, c("Santander.Santander.LogicalEntity.WEB.production.IHS.access.SANPARTICULARES"="WEB.IHS.SANPART"))

minilog10 <- minilog10[c("event\_date", "generated\_date", "http\_response\_code", "source\_ip", "ip", "http\_protocol", "http\_response\_size", "http\_method", "program", "http\_user\_agent", "http\_request\_url" )]

#Change log names

library(plyr)

minilog10$program <- revalue(minilog10$program, c("Santander.Santander.LogicalEntity.WEB.production.IHS.access.SANPARTICULARES"="WEB.IHS.SANPART"))

log9\_10$ip<- revalue(log9\_10$ip,c( "bvsalwparp09-adm.wt.santander.corp"="web-sanpar-09","bvsalwparp10-adm.wt.santander.corp"="web-sanpar-10" ))

#### Combining logs

We are going to try to combine log09, log10

* First log: log-web-09
* Second log: log-web-10

rm(list = ls())

setwd('C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\2-R\\preprocess')

file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\LogicalEntity\\WEB-IHS\\SANPARTICULARES\\bvsalwparp09-adm.wt.santander.corp-02\\access\_1.log"

source("01-01-a-preprocess-web-generic.R")

log\_web\_09 <- logA

file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\LogicalEntity\\WEB-IHS\\SANPARTICULARES\\bvsalwparp10-adm.wt.santander.corp-02\\access\_1.log"

source("01-01-a-preprocess-web-generic.R")

log\_web\_10 <- logA

minilogA <- log\_web\_09 [1:10,]

minilogB <- log\_web\_10 [1:10,]

minilogAB <- rbind (minilogA, minilogB)

minilogAB <- minilogAB[order(minilogAB$event\_date),]

library(plyr)

minilogAB$program<- revalue(minilogAB$program,c("Santander.Santander.LogicalEntity.WEB.production.IHS.access.SANPARTICULARES"="WEB.IHS.SANPAR"))

minilogAB$ip<- revalue(minilogAB$ip,c( "bvsalwparp09-adm.wt.santander.corp"="web-sanpar-09","bvsalwparp10-adm.wt.santander.corp"="web-sanpar-10" ))

minilogAB[!(colnames(minilogAB) %in% c("http\_request\_domain"))]

log9\_10 <- rbind (log\_web\_09, log\_web\_10)

log9\_10 <- log9\_10[order(log9\_10$event\_date),]

##### Cleaning up combined logs

#cleanup

library(plyr)

log9\_10$program<- revalue(log9\_10$program,c("Santander.Santander.LogicalEntity.WEB.production.IHS.access.SANPARTICULARES"="WEB.IHS.SANPAR"))

log9\_10$ip<- revalue(log9\_10$ip,c( "bvsalwparp09-adm.wt.santander.corp"="web-sanpar-09","bvsalwparp10-adm.wt.santander.corp"="web-sanpar-10" ))

log9\_1 <- log9\_10 [!(colnames(log9\_10) %in% c("http\_request\_domain"))]

summary(log9\_10$http\_response\_code[log9\_10$ip=="web-sanpar-09"])

200 206 302 403 404 416

234779 22 30 18 4473 1

summary(log9\_10$http\_response\_code[log9\_10$ip=="web-sanpar-10"])

200 206 302 403 404 416

222608 21 24 18 4253 0

#### Request Rate

a<- rle(log9\_1$event\_date)

reqsPerSec <- data.frame(date=a$values, n=a$lengths)

eventDateMins <- round.POSIXt(log9\_1$event\_date, "mins")

b<- rle(eventDateMins)

reqsPerMin <- data.frame(date=b$values, n=b$lengths)

##### Plotting req rate

par(pch=1)

par(cex=0.25)

par(cex.axis=1)

par(cex.lab=1)

par(mfrow=c(2,1))

plot(reqsPerSec $date, reqsPerSec $n)

plot(reqsPerMin$date, reqsPerMin$n)

plot(reqsPerSec $date, log(reqsPerSec$n +1))

plot(reqsPerMin$date, log(reqsPerMin$n +1))

plot(reqsPerMin$date, log(reqsPerMin$n +1), col="red")

plot(reqsPerMin$date, log(reqsPerMin$n +1), type="l")

plot(reqsPerMin$date, reqsPerMin$n, type="l")

#partial requestrates

eventDateMins9 <- round.POSIXt(log9\_1$event\_date[log9\_1$ip=="web-sanpar-09"], "mins")

eventDateMins10 <- round.POSIXt(log9\_1$event\_date[log9\_1$ip=="web-sanpar-10"], "mins")

a9<- rle(eventDateMins9)

a10<- rle(eventDateMins10)

plot(reqsPerMin$date, log(reqsPerMin$n +1), type="l")

lines(a9$values, log(a9$lengths+1), col="blue")

lines(a10$values, log(a10$lengths+1), col="red")

par(mfrow=c(1,1))

par(pch=1)

par(cex=0.8)

par(cex.axis=1)

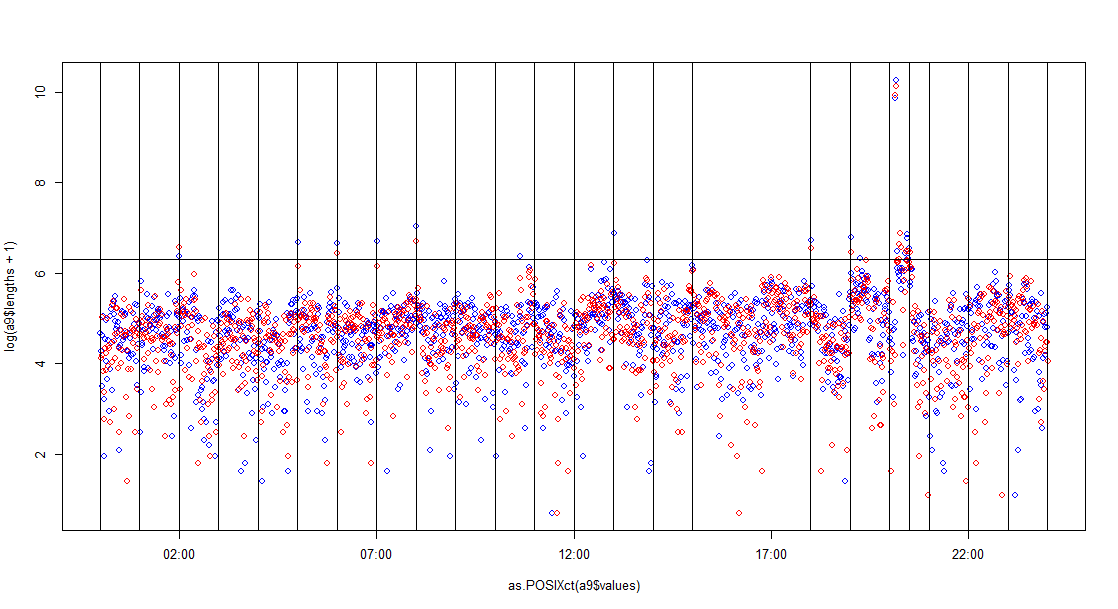
par(cex.lab=1)

plot(as.POSIXct(a9$values), log(a9$lengths+1), col="blue")

points(as.POSIXct(a10$values), log(a10$lengths+1), col="red")

abline(v= as.POSIXct (strptime("2013-01-01 01:00:00 CET", format= "%Y-%m-%d %H:%M:%S")))

abline(v= as.POSIXct (strptime("2013-01-01 02:00:00 CET", format= "%Y-%m-%d %H:%M:%S")))



summary(a9$lengths)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.0 71.0 119.0 173.2 176.0 28810.0

> summary(a10$lengths)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.0 68.0 111.0 164.9 169.0 25340.0

### Firewall

Available Firewall logs:

* Firewall\production\CheckpointFW1
* Conectividad
  + audit
    - CMA\_Conectividad
    - CMA\_Conectividad\_HA
  + traffic
    - banzai
    - criasor
    - gerion
    - habis
    - jasper
    - norax
* Chile
  + audit\CMA\_Chile
  + traffic
    - FWCHL-FRON-BEP1
    - FWCHL-FRON-BOP1
    - FWDRPQBO1
    - FWDRPSOV
* SAN\_Internet\_MS
  + audit
    - SAN\_INTERNET\_HA\_MS
    - SAN\_INTERNET\_MS
  + traffic
    - 0.0.0.0
    - hopper
    - vmwtbitarecol03
    - mulan
    - pinocho
    - SAN\_INTERNET\_HA\_MS
    - SAN\_INTERNET\_ML
    - SAN\_INTERNET\_MS
    - wendy
* SPB\_MS
  + audit
    - SPB\_HA\_MS
    - SPB\_MS
  + traffic
    - fuxi
    - nuwa
    - pangu
    - SPB\_ML
    - SPB\_MS
* Universia
  + audit
    - CMA\_Universia
    - CMA\_Universia\_HA
  + traffic
    - Malaka
    - Onuba

We will start with SAN\_Internet\_MS / mulan (and then hopper)

2 files: Accept, drop

#### XML Schema

##### Schema: fw-sanpart-accept

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\0-log-tables\tables\table-ihs-santander-sanparticulares-access.xml

Formatted Route:

C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\0-log-tables\\tables\\table-ihs-santander-sanparticulares-access.xml

<?xml version="1.0" encoding="ISO-8859-1"?>

<table name="CheckpointFW1-SAN\_Internet\_MS-traffic-accept">

<source>Local-CommunicationSystems-Firewall-CheckpointFW1-SAN\_Internet\_MS-Traffic-Accept</source>

<path type='offline'>\\\\vmwtbitafiler01.ad.produban.corp\\cuaderno2\$\\bitacora\\cuaderno\\{YEAR}\\{MONTH}\\{DAY}\\(\w+)\\(\w+)\\(SecuritySystem)\\(Firewall)\\(\w+)\\CheckpointFW1\\(SAN\_Internet\_MS)\\traffic\\(.\*?)\\(.\*?)\\traffic-accept.log(\.gz|)</path>

<log>(.{15}) (\S+) (\S+) ([^:]+):\s+(time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:product=(.{18}).\*?|rule=(\S+)\s|src=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|s\_port=(\S+)\s|dst=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|service=(\S+)\s|proto=(\S+)\s|.+?)+).\*$</log>

<categories>

<category name="FW-TRAFFIC-ACCEPT"/>

<category name="CHECKPOINT-TRAFFIC-SAN\_Internet\_MS"/>

</categories>

<columns>

<column name="raw" from="log" group="0" parser="STRING"/>

<column name="generated-date" from="log" group="6" parser="eventlog-date"/>

<column name="message" from="log" group="5" parser="STRING"/>

<column name="event\_date" from="log" group="1" parser="syslog-date" eventdate="true"/>

<column name="source\_ip" from="log" group="14" parser="STRING"/>

<column name="src-port" from="log" group="15" parser="STRING"/>

<column name="destination\_ip" from="log" group="16" parser="STRING"/>

<column name="dst-port" from="log" group="17" parser="STRING"/>

<column name="rule" from="log" group="13" parser="STRING"/>

<column name="if\_dir" from="log" group="9" parser="STRING"/>

<column name="service" from="log" group="17" parser="STRING"/>

<column name="program" from="log" group="4" parser="STRING"/>

<column name="log\_action" from="log" group="7" parser="STRING"/>

<column name="orig" from="log" group="8" parser="STRING"/>

<column name="if\_name" from="log" group="10" parser="STRING"/>

<column name="has\_accounting" from="log" group="11" parser="STRING"/>

<column name="product" from="log" group="12" parser="STRING"/>

<column name="company" from="path" group="4" parser="STRING"/>

<column name="vdc" from="path" group="5" parser="STRING"/>

<column name="category" from="path" group="6" parser="STRING"/>

<column name="type" from="path" group="7" parser="STRING"/>

<column name="functionalenvironment" from="path" group="8" parser="STRING"/>

<column name="hostname" from="path" group="10" parser="STRING"/>

<column name="console" from="path" group="9" parser="STRING"/>

<column name="product-vendor" from="expression" type="STRING">&apos;Checkpoint&apos;</column>

<column name="product-model" from="expression" type="STRING">&apos;Firewall One&apos;</column>

<column name="product\_version" from="expression" type="STRING">&apos;-&apos;</column>

<column name="item" from="expression" type="STRING">&apos;-&apos;</column>

<column name="action" from="expression" type="STRING">&apos;ACCEPT&apos;</column>

</columns>

</table>

##### Schema: fw-sanpart-drop

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\0-log-tables\tables\table-ihs-santander-sanparticulares-drop.xml

Formatted Route:

C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\SecuritySystem\\Firewall\\SAN\_Internet\_MS\\traffic\\mulan\\03\\traffic-drop.log

Content:

<?xml version="1.0" encoding="ISO-8859-1"?>

<table name="CheckpointFW1-SAN\_Internet\_MS-traffic-drop">

<source>Local-CommunicationSystems-Firewall-CheckpointFW1-SAN\_Internet\_MS-Traffic-Drop</source>

<path type='offline'>\\\\vmwtbitafiler01.ad.produban.corp\\cuaderno2\$\\bitacora\\cuaderno\\{YEAR}\\{MONTH}\\{DAY}\\(\w+)\\(\w+)\\(SecuritySystem)\\(Firewall)\\(\w+)\\CheckpointFW1\\(SAN\_Internet\_MS)\\traffic\\(.\*?)\\(.\*?)\\traffic-drop.log(\.gz|)</path>

<log>(.{15}) (\S+) (\S+) ([^:]+):\s+(time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:product=(.{18}).\*?|rule=(\S+)\s|src=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|s\_port=(\S+)\s|dst=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|service=(\S+)\s|proto=(\S+)\s|.+?)+).\*$</log>

<categories>

<category name="FW-TRAFFIC-DROP"/>

<category name="CHECKPOINT-TRAFFIC-SAN\_Internet\_MS"/>

</categories>

<columns>

<column name="raw" from="log" group="0" parser="STRING"/>

<column name="generated-date" from="log" group="6" parser="eventlog-date"/>

<column name="message" from="log" group="5" parser="STRING"/>

<column name="event\_date" from="log" group="1" parser="syslog-date" eventdate="true"/>

<column name="src-port" from="log" group="15" parser="STRING"/>

<column name="dst-port" from="log" group="17" parser="STRING"/>

<column name="rule" from="log" group="13" parser="STRING"/>

<column name="source\_ip" from="log" group="14" parser="STRING"/>

<column name="destination\_ip" from="log" group="16" parser="STRING"/>

<column name="if\_dir" from="log" group="9" parser="STRING"/>

<column name="service" from="log" group="17" parser="STRING"/>

<column name="program" from="log" group="4" parser="STRING"/>

<column name="log\_action" from="log" group="7" parser="STRING"/>

<column name="orig" from="log" group="8" parser="STRING"/>

<column name="if\_name" from="log" group="10" parser="STRING"/>

<column name="has\_accounting" from="log" group="11" parser="STRING"/>

<column name="product" from="log" group="12" parser="STRING"/>

<column name="company" from="path" group="4" parser="STRING"/>

<column name="vdc" from="path" group="5" parser="STRING"/>

<column name="category" from="path" group="6" parser="STRING"/>

<column name="type" from="path" group="7" parser="STRING"/>

<column name="functionalenvironment" from="path" group="8" parser="STRING"/>

<column name="hostname" from="path" group="10" parser="STRING"/>

<column name="console" from="path" group="9" parser="STRING"/>

<column name="product-vendor" from="expression" type="STRING">&apos;Checkpoint&apos;</column>

<column name="product-model" from="expression" type="STRING">&apos;FirewallOne&apos;</column>

<column name="product\_version" from="expression" type="STRING">&apos;-&apos;</column>

<column name="item" from="expression" type="STRING">&apos;-&apos;</column>

<column name="action" from="expression" type="STRING">&apos;DROP&apos;</column>

</columns>

</table>

##### Schema Summary

Same fields for accept, drop

* generated-date
* message
* event\_date
* src-port
* dst-port
* rule
* source\_ip
* destination\_ip
* if\_dir
* service
* program
* log\_action
* orig
* if\_name
* has\_accounting
* product

#### Mulan/traffic-drop.log

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\SecuritySystem\Firewall\SAN\_Internet\_MS\traffic\mulan\03\traffic-drop.log

Route (formatted for R):

File details

* 225 MB
* 420k lines
* 1 day (Jan-01-13)
* Sample line:

Jan 01 00:00:00 SAN\_Internet\_MS mulan Santander.Santander.SecuritySystem.Firewall.production.CheckpointFW1.SAN\_Internet\_MS.traffic: time=Mon Dec 31 23:59:59 2012 action=drop orig=mulan i/f\_dir=inbound i/f\_name=eth-s1p1c0 has\_accounting=0 product=VPN-1 & FireWall-1 \_\_policy\_id\_tag=product=VPN-1 & FireWall-1[db\_tag={1029F2AE-5032-11E2-8D8C-160FE8A9F5F5};mgmt=SAN\_INTERNET\_MS;date=1356618672;policy\_name=FW\_Internos] rule=755 rule\_uid={631031AE-CE77-11D8-B260-AC166B75E9E9} src=S\_WVA\_180.132.26.17 s\_port=42871 dst=S\_boafaoraci01.afb.corp\_172.22.7.67 service=tcp-5527 proto=6

#### Hopper/traffic-drop.log

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\SecuritySystem\Firewall\SAN\_Internet\_MS\traffic\mulan\03\traffic-drop.log

Route (formatted for R):

File details

File details

* 1.6 MB
* 3k lines
* 1 day (Jan-01-13)
* Sample line:

Jan 01 00:00:00 SAN\_Internet\_MS mulan Santander.Santander.SecuritySystem.Firewall.production.CheckpointFW1.SAN\_Internet\_MS.traffic: time=Mon Dec 31 23:59:59 2012 action=drop orig=mulan i/f\_dir=inbound i/f\_name=eth-s1p1c0 has\_accounting=0 product=VPN-1 & FireWall-1 \_\_policy\_id\_tag=product=VPN-1 & FireWall-1[db\_tag={1029F2AE-5032-11E2-8D8C-160FE8A9F5F5};mgmt=SAN\_INTERNET\_MS;date=1356618672;policy\_name=FW\_Internos] rule=755 rule\_uid={631031AE-CE77-11D8-B260-AC166B75E9E9} src=S\_WVA\_180.132.26.17 s\_port=42871 dst=S\_boafaoraci01.afb.corp\_172.22.7.67 service=tcp-5527 proto=6

##### R code

rm(list = ls())

setwd('C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\2-R\\preprocess')

source("01-01-preprocess-web-sanpart-10.R")

#runs slowly, almost crashes

###### Preprocessing

Working directory

C:\Users\capelastegui\workspace\OFP\Santander-1\2-R\preprocess

C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\2-R\\preprocess

setwd('C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\2-R\\preprocess')

file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\SecuritySystem\\Firewall\\SAN\_Internet\_MS\\traffic\\mulan\\03\\traffic-drop.log"

file <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\SecuritySystem\\Firewall\\SAN\_Internet\_MS\\traffic\\hopper\\03\\traffic-drop.log"

source("01-01-a-preprocess-fw-generic.r")

csvFile <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Processed\\CSV\\log.csv"

write.table(log, file=csvFile, sep="\t", row.names =FALSE)

01-01-a-preprocess-fw-generic.R

#Replace file, xmlFile as required

#file <- "..."

xmlFile <- "C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\1-Data\\Logs\\0-log-tables\\tables\\table-checkpointfw1-san\_internet\_ms-traffic-drop.xml"

library(XML)

library(gsubfn)

x <- xmlTreeParse(xmlFile)

r <- xmlRoot(x)

l<- xmlSApply(r[["columns"]], xmlAttrs)

regex <- "(.{15}) (\\S+) (\\S+) ([^:]+):\\s+()time=(.{24})\\s+action=(\\w\*) orig=(\\S\*) i/f\_dir=(\\S\*) i/f\_name=(\\S\*) has\_accounting=(\\S+) (?:product=(.{18})(?:.{152}))rule=(\\S+)\\s\\S\*\\ssrc=(?:\\S+\_)?(\\d+\\.\\d+\\.\\d+\\.\\d+)\\s(?:s\_port=(\\S+)\\s)?dst=(?:\\S+\_)?(\\d+\\.\\d+\\.\\d+\\.\\d+)\\s(?:service=(\\S+)\\s)?proto=(\\S+)\\s?"#Switch lines to read full log or summary

lines <- readLines(file, 10)

#lines <- readLines(file)

log<- data.frame(strapplyc(lines, regex, simplify = "rbind"))

myF <- function(x, name="name")

{x[[name]]}

vapply(l,myF, "", USE.NAMES=FALSE)

myF <- function(x, name="name") {if(name%in%names(x)) {x[[name]]} else "NA"}

#get names, groups, froms, types

froms <- vapply(l,myF, "", "from", USE.NAMES=FALSE)

groups <- vapply(l,myF, "", "group", USE.NAMES=FALSE)

names <- vapply(l,myF, "", "name", USE.NAMES=FALSE)

types <- vapply(l,myF, "", "parser", USE.NAMES=FALSE)

indices <- as.numeric(groups[froms=="log"])

indicesB <- indices[indices>0]

line1 <- log[1,]

line1a <- line1[indicesB]

names(line1a) <- names[froms=="log"][indices>0]

logA <- log[indicesB]

names(logA) <- names[froms=="log"][indices>0]

#Column procesing

logA$http\_response\_size <- as.numeric(logA$http\_response\_size)

#Date

Sys.setlocale("LC\_TIME", "english")

event\_date2 <- strptime(logA$event\_date, "%b %d %H:%M:%S")

generated\_date2 <- strptime(logA$generated\_date, "%d/%b/%Y:%H:%M:%S")

logA$event\_date <- event\_date2

logA$generated\_date <- generated\_date2

###### Preprocessing problem:

REGEX isn't well processed in R

dst=(?:\S+\_.\*?)?(\d+\.\d+\.\d+\.\d+)\s

Jan 01 00:00:00 SAN\_Internet\_MS mulan Santander.Santander.SecuritySystem.Firewall.production.CheckpointFW1.SAN\_Internet\_MS.traffic: time=Mon Dec 31 23:59:59 2012 action=drop orig=mulan i/f\_dir=inbound i/f\_name=eth-s1p1c0 has\_accounting=0 product=VPN-1 & FireWall-1 \_\_policy\_id\_tag=product=VPN-1 & FireWall-1[db\_tag={1029F2AE-5032-11E2-8D8C-160FE8A9F5F5};mgmt=SAN\_INTERNET\_MS;date=1356618672;policy\_name=FW\_Internos] rule=755 rule\_uid={631031AE-CE77-11D8-B260-AC166B75E9E9} src=S\_WVA\_180.132.26.17 s\_port=42871 dst=S\_boafaoraci01.afb.corp\_172.22.7.67 service=tcp-5527 proto=6

strapplyc(readLines(fileTmpLines, 1), readLines(fileTmpRegex)[1], simplify = "rbind", backref = NULL)

###### Explanation for Preprocessing problem:

<http://www.regular-expressions.info/engine.html>

There are two kinds of regular expression engines: text-directed engines, and regex-directed engines. Jeffrey Friedl calls them DFA and NFA engines, respectively. All the regex flavors treated in this tutorial are based on regex-directed engines

This is because certain very useful features, such as lazy quantifiers and backreferences, can only be implemented in regex-directed engines

You can easily find out whether the regex flavor you intend to use has a text-directed or regex-directed engine. If backreferences and/or lazy quantifiers are available, you can be certain the engine is regex-directed. You can do the test by applying the regex regex|regex not to the string regex not. If the resulting match is only regex, the engine is regex-directed. If the result is regex not, then it is text-directed. The reason behind this is that the regex-directed engine is "eager".

strapplyc("regex|regex not", "regex not")

[[1]]

[1] "regex not"

R is text-directed

Is this based on grep? or on a specific regex library? Explore

<http://www.regular-expressions.info/repeat.html>

this tutorial only talks about regex-directed engines. Text-directed engines do not backtrack. They do not get the speed penalty, but they also **do not support lazy repetition operators**.

Alternative to laziness:

To identify a <tag>

Instead of <.+?>

We can use a greedy plus and a negated character class: <[^>]+>

<http://www.regular-expressions.info/lookaround.html>

Negative lookahead is indispensable if you want to match something not followed by something else. When explaining character classes, I already explained why you cannot use a negated character class to match a "q" not followed by a "u". Negative lookahead provides the solution: q(?!u).

Positive lookahead works just the same. q(?=u) matches a q that is followed by a u

###### Solutions for preprocessing problem

Revise usage of '?' at end of groups. Try adding/removing some

###### RegEx Analysis

#Original RegEx

(.{15}) (\S+) (\S+) ([^:]+):\s+(time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:product=(.{18}).\*?|rule=(\S+)\s|src=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|s\_port=(\S+)\s|dst=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|service=(\S+)\s|proto=(\S+)\s|.+?)+).\*$

#Log Line

Jan 01 00:00:00 SAN\_Internet\_MS mulan Santander.Santander.SecuritySystem.Firewall.production.CheckpointFW1.SAN\_Internet\_MS.traffic: time=Mon Dec 31 23:59:59 2012 action=drop orig=mulan i/f\_dir=inbound i/f\_name=eth-s1p1c0 has\_accounting=0 product=VPN-1 & FireWall-1 \_\_policy\_id\_tag=product=VPN-1 & FireWall-1[db\_tag={1029F2AE-5032-11E2-8D8C-160FE8A9F5F5};mgmt=SAN\_INTERNET\_MS;date=1356618672;policy\_name=FW\_Internos] rule=755 rule\_uid={631031AE-CE77-11D8-B260-AC166B75E9E9} src=S\_WVA\_180.132.26.17 s\_port=42871 dst=S\_boafaoraci01.afb.corp\_172.22.7.67 service=tcp-5527 proto=6

#revised dst portion

dst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\s

#Removed group 5, useless

(.{15}) (\S+) (\S+) ([^:]+):\s+(?:time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:product=(.{18}).\*?|rule=(\S+)\s|src=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|s\_port=(\S+)\s|dst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\s|service=(\S+)\s|proto=(\S+)\s|.+?)+).\*$

Problem with original:

In R:

[,1] [,2] [,3]

[1,] "Jan 01 00:00:00" "SAN\_Internet\_MS" "mulan"

[,4]

[1,] "Santander.Santander.SecuritySystem.Firewall.production.CheckpointFW1.SAN\_Internet\_MS.traffic"

[,5]

[1,] "time=Mon Dec 31 23:59:59 2012 action=drop orig=mulan i/f\_dir=inbound i/f\_name=eth-s1p1c0 has\_accounting=0 product=VPN-1 & FireWall-1 \_\_policy\_id\_tag=product=VPN-1 & FireWall-1[db\_tag={1029F2AE-5032-11E2-8D8C-160FE8A9F5F5};mgmt=SAN\_INTERNET\_MS;date=1356618672;policy\_name=FW\_Internos] rule=755 rule\_uid={631031AE-CE77-11D8-B260-AC166B75E9E9} src=S\_WVA\_180.132.26.17 s\_port=42871 dst=S\_boafaoraci01.afb.corp\_172.22.7.67 service=tcp-5527 proto=6"

[,6] [,7] [,8] [,9] [,10] [,11] [,12]

[1,] "Mon Dec 31 23:59:59 2012" "drop" "mulan" "inbound" "eth-s1p1c0" "0" ""

[,13] [,14] [,15] [,16] [,17] [,18]

[1,] "" "" "" "" "" ""

group 12 misses! Should be VPN-1 & FireWall-1

RegexStructure

(.{15})

(\S+)

(\S+)

([^:]+):\s+

(time=

(.{24})

\s+action=(\w\*)

orig=(\S\*)

i/f\_dir=(\S\*)

i/f\_name=(\S\*)

has\_accounting=(\S+)

(?:

product=(.{18}).\*?|

rule=(\S+)\s|

src=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|

s\_port=(\S+)\s|

dst=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|

service=(\S+)\s|

proto=(\S+)\s|

.+?

)

+).\*$

Alt Structure

(.{15})

(\S+)

(\S+)

([^:]+):\s+

(time=

(.{24})

\s+action=(\w\*)

orig=(\S\*)

i/f\_dir=(\S\*)

i/f\_name=(\S\*)

has\_accounting=(\S+)

(?:

product=(.{18}).\*?|

rule=(\S+)\s|

src=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|

s\_port=(\S+)\s|

dst=(?:\S+\_.\*?)?([A-Za-z0-9-]+|\d+\.\d+\.\d+\.\d+)\s|

service=(\S+)\s|

proto=(\S+)\s|

.+? //Lazy Match: problem!

)

+).\*$

Rewriting group 12

Regex:

product=(.{18}).\*?|

#lookahead implementation

product=(.{18}).\*(?=rule=|src=|s\_port=|dst=|service=)|

product=(.{18}).\*(?=rule= )|(?=src= )|(?=s\_port= )|(?=dst= )|(?=service=)

product=(.{18}).\*(?=rule=)|(?=src=)|(?=s\_port=)|(?=dst=)|(?=service=)

product=(.{18})|(?:.\*(?=rule=)|(?=src=)|(?=s\_port=)|(?=dst=)|(?=service=))

Log samples:

...

421K samples with that product id (all of them!)

product=VPN-1 & FireWall-1 \_\_policy\_id\_tag=product=VPN-1 & FireWall-1[db\_tag={1029F2AE-5032-11E2-8D8C-160FE8A9F5F5};mgmt=SAN\_INTERNET\_MS;date=1356618672;policy\_name=FW\_Internos]

product=VPN-1 & FireWall-1 \_\_policy\_id\_tag=product=VPN-1 & FireWall-1[db\_tag={EA3D9A48-4F6C-11E2-8D8C-160FE8A9F5F5};mgmt=SAN\_INTERNET\_MS;date=1356533997;policy\_name=FW\_Perimetrales]

This works on R

product=(.{18}).\*(?=rule=|src=|s\_port=|dst=|service=)

Change list

* product=

product=(.{18}).\*(?=rule=)|(?=src=)|(?=s\_port=)|(?=dst=)|(?=service=)

* src=
* src=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\s
* dst=
* dst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\s

TODO: Last term doesn't work! It's a lazy match.

.+?

How to integrate?

(.\*(?=rule=)|(?=src=)|(?=s\_port=)|(?=dst=)|(?=service=))\*

Last reference (does not work)

(.{15}) (\S+) (\S+) ([^:]+):\s+(time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:(?:product=(.{18})(.{152}))|rule=(\S+)\s|src=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\s|s\_port=(\S+)\s|dst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\s|service=(\S+)\s|proto=(\S+)\s|.+?)+).\*$

Good regex (a bit hacky)

(.{15}) (\S+) (\S+) ([^:]+):\s+(time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:(?:product=(.{18})(?:.{152}))rule=(\S+)\s\S\*\ssrc=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\ss\_port=(\S+)\sdst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\sservice=(\S+)\sproto=(\S+)\s?))

(.{15}) (\S+) (\S+) ([^:]+):\s+(?:time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:(?:product=(.{18})(?:.{152}))rule=(\S+)\s\S\*\ssrc=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\ss\_port=(\S+)\sdst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\sservice=(\S+)\sproto=(\S+)\s?))

(.{15}) (\S+) (\S+) ([^:]+):\s+(?:()time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:(?:product=(.{18})(?:.{152}))rule=(\S+)\s\S\*\ssrc=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\ss\_port=(\S+)\sdst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\sservice=(\S+)\sproto=(\S+)\s?))

(.{15}) (\S+) (\S+) ([^:]+):\s+()time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:(?:product=(.{18})(?:.{152}))rule=(\S+)\s\S\*\ssrc=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\ss\_port=(\S+)\sdst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\sservice=(\S+)\sproto=(\S+)\s?)

Most simple, works

(.{15}) (\S+) (\S+) ([^:]+):\s+()time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:product=(.{18})(?:.{152}))rule=(\S+)\s\S\*\ssrc=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\ss\_port=(\S+)\sdst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\sservice=(\S+)\sproto=(\S+)\s?

Need to make fields optional

(.{15}) (\S+) (\S+) ([^:]+):\s+()time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:product=(.{18})(?:.{152}))rule=(\S+)\s\S\*\ssrc=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\s(?:s\_port=(\S+)\s)?dst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\s(?:service=(\S+)\s)?proto=(\S+)\s?

Last attempt:

(.{15}) (\S+) (\S+) ([^:]+):\s+(time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (.{15}) (\S+) (\S+) ([^:]+):\s+(time=(.{24})\s+action=(\w\*) orig=(\S\*) i/f\_dir=(\S\*) i/f\_name=(\S\*) has\_accounting=(\S+) (?:(?:product=(.{18})(?:.{152}))rule=(\S+)\s\S\*\ssrc=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\ss\_port=(\S+)\sdst=(?:\S+\_)?(\d+\.\d+\.\d+\.\d+)\sservice=(\S+)\sproto=(\S+)\s?))

###### Further preprocessing problem!

Regex: ^(?:(a)|(b)|(c))+

text: abcdece

[,1] [,2] [,3]

[1,] "" "" "c"

You cannot alternate capturing groups!

#### Hopper/traffic-accept.log

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\SecuritySystem\Firewall\SAN\_Internet\_MS\traffic\mulan\03\traffic-drop.log

Route (formatted for R):

File details

#### Mulan/traffic-accept.log

Route:

C:\Users\capelastegui\workspace\OFP\Santander-1\1-Data\Logs\SecuritySystem\Firewall\SAN\_Internet\_MS\traffic\mulan\03\traffic-drop.log

Route (formatted for R):

File details

#### Tmp

rm(list = ls())

setwd('C:\\Users\\capelastegui\\workspace\\OFP\\Santander-1\\2-R\\preprocess')

source("01-01-preprocess-web-sanpart-10.R")

#runs slowly, almost crashes