# Data Goals:

See what content packs have been downloaded and by who, how many times.

See what customers are at what server level.

# Data Source:

Logs from CDN (may want to automate the pull on a monthly (or other periodic) basis.

# 20150818

Plan for today is to redo in R, the preprocessing I was doing in Bash via Awk, et al.

Instead of doing everything at once like I’ve been doing, there may be more value in maintaining a separate complete list of solutions and also one for server versions. They can always be merged into a third data set in a separate file.

I may set up a VM with RStudio (or just R) to do the processing for KCL

# Solution Data Harvest:

166.78.9.169

/var/log/nginx . I copy the desired access.log\*.gz files to my chughes/data directory, then use SCP from my local system to retrieve them.

This is the Bash code I’ve been using to preprocess the solutions relevant data.

|  |
| --- |
|  |
| grep "GET /files/published/" access.log | awk '{print $7, $11, $NF}'| grep -v "\"\-\"" | sed 's/\"//g' | sed 's/https:\/\// /g' | sed 's/\// /g' | awk '{print $3","$6","$NF}'| grep -v errata | sort -u > `date +"%Y%m%d\_%H%M%S"`\_solutionmap.csv |
|  |

1. Retrieve lines with “GET /files/published/”
2. Select appropriate columns
3. Discard rows without a requestor
4. Pull out timestamp
5. Pull out Solution name

This is the Bash code used to get the Server Version info.

|  |
| --- |
|  |

# To cleanse CDN Server logs

#

#remove the next line's comment for a fresh read

data0 <- readLines("data/access.log")

#######################################################################

# Begin Solution Section

#######################################################################

#######################################################################

# # Remove common, noisy entries of Log Batch

# data1 <- grepl("[00:000000:]", data0, fixed = TRUE)

# data2 <- data0[!data1]

# Keep only the lines of interest

sol1 <- grepl("GET /files/published/", data0, fixed = T)

sol2 <- data0[sol1]

#######################################################################

#######################################################################

#split these results into fields based on " . Setting fixed=F allows REGEX.

soldf1 <- unlist(strsplit(sol2, split = "\"", fixed = F))

#######################################################################

#######################################################################

# Convert to data frame

sol3 <- matrix(unlist(soldf1), ncol = 8, byrow = TRUE)

colnames(sol3) <- c("TimeStamp", "Solution", "HTML Code", "Four", "Five", "Six", "Seven", "Requester")

sol4 <- as.data.frame(sol3, stringsAfFactors = FALSE)

# Remove rows that lack requestor (code 404 168 is an indicator too)

sol5 <- subset(sol4, !(Requester=="-"))

#######################################################################

#######################################################################

#Remove needless columns

sol6 <- sol5[c(1,2,8)]

colnames(sol6) <- c("TimeStamp", "Solution", "Requester")

#Trim out CDN Address from TimeStamp field

tstp1 <- as.character(sol6$TimeStamp)

tstp2 <- unlist(strsplit(tstp1, split = "[", fixed = T))

tstp3 <- strsplit(tstp2, split = "]", fixed = T)

tstp4 <- unlist(tstp3[grep("/",tstp3)])

tstp5 <- strsplit(tstp4, split = " +0000", fixed = T)

tstp6 <- tstp5[grep("/",tstp5)]

sol6$TimeStamp <- tstp6

#Trim out Solution name for Solution field

sln1 <- as.character(sol6$Solution)

sln2 <- unlist(strsplit(sln1, split = "GET /files/published/", fixed = T))

sln3 <- unlist(strsplit(sln2, split = " HTTP/[0-9].[0-9]"))

#sln4 <- as.data.frame(sln3)

sol6$Solution <- sln3

#######################################################################

#Begin Server Version Section

#######################################################################

# Keep only the lines of interest, the v= is only present on lines that

# reference the Tanium Server Version

con1 <- grepl("v=", data0, fixed = T)

con2 <- data0[con1]

# First we have to turn the line of text into a dataframe with discernible fields that can be manipulated

con3 <- as.data.frame(matrix(unlist(strsplit(con2, split = "\"", fixed = T)), ncol=8, byrow = T),stringsAfFactors = FALSE)

colnames(con3) <- c("TimeStamp","TaniumServerVersion","Code", "Referer","Junk1", "Junk2", "Junk3" ,"Requester")

# Keep only the useful columns

con3 <- con3[c("TimeStamp","TaniumServerVersion","Referer","Requester")]

# # Remove rows that lack requestor (code 404 168 is an indicator too)

con4 <- subset(con3, !(con3$Requester=="-"))

# Get the timestamp.

#Trim out CDN Address from TimeStamp field

cntstp1 <- as.character(con4$TimeStamp)

cntstp2 <- unlist(strsplit(cntstp1, split = "[", fixed = T))

cntstp3 <- strsplit(cntstp2, split = "]", fixed = T)

cntstp4 <- unlist(cntstp3[grep("/",cntstp3)])

cntstp5 <- strsplit(cntstp4, split = " +0000", fixed = T)

cntstp6 <- cntstp5[grep("/",cntstp5)]

con4$TimeStamp <- cntstp6

# Get the Tanium Server Version

cntsv1 <- as.character(con4$TaniumServerVersion)

# 5957

cntsv2 <- unlist(strsplit(cntsv1, split = "?v=", fixed = T))

# 11905

cntsv3 <- cntsv2[grepl("&s=", cntsv2, fixed = T)]

# 5948

cntsv4 <- unlist(strsplit(cntsv3, split = "&s=", fixed = T))

cntsv5 <- cntsv4[grepl("[0-9][.][0-9][.][0-9]{3}[.][0-9]{4}", cntsv4, fixed = F) ]

con4$TaniumServerVersion <- cntsv5

#######################################################################

#End Server Version Section - Complete

#######################################################################

#######################################################################

#Begin Referer Section

#######################################################################

# Will strip down the data to just what is in the domain portion

cntrf1 <- as.character(con4$Referer)

cntrf2 <- as.character(strsplit(cntrf1, split = "http[s]\*://", fixed = F))

cntrf3 <- strsplit(cntrf2, split = "/", fixed = T)

t1 <- as.character(cntrf3)

t2 <- gsub('c[(]["]c[(][\\]["][\\\\\\]["][,] [\\\\\\]["]',"",t1, fixed = F)

t3 <- gsub('["][,] ["][\\\\\\]["][)]["][)]',"",t2, fixed = F)

t4 <- gsub('[\\]["][)]["]\*[)]\*',"", t3, fixed = F)

t5 <- gsub('["][,].\*',"", t4, fixed = F)

con4$Referer <- t5

#######################################################################

#Begin Requester Section

#######################################################################

r1 <- as.character(con4$Requester)

r2 <- gsub('(.\*[,])\*',"",r1, fixed = F)

con4$Requester <- r2