macOS R users who tend to work on the bleeding edge likely noticed some downtime at this past weekend. Part of the issue was an SSL/TLS certificate expiration situation. Moving forward, we can monitor this with R using the super spiffy {openssl} and {pushoverr} packages whilst also generating a daily report with {rmarkdown} and {DT}.

**The Basic Process**

The {openssl} package has a handy function — download\_ssl\_cert() — which will, by default, hit a given host on the standard HTTPS port (443/TCP) and grab the site certificate and issuer. We’ll grab the “validity end” field and convert that to a date to use for comparison.

To get the target list of sites to check I used [Rapid7’s FDNS data set](https://opendata.rapid7.com/sonar.fdns_v2/) and a glance at a few certificate transparency logs to put together a current list of “r-project” domains that have been known to have SSL certs. This process could be made more dynamic, but things don’t change that quickly in r-project domain land.

Finally, we use the {DT} package to build a pretty HTML table and the {pushoverr} package to send notifications at normal priority for certs expiring within a week and critical priority for certs that have expired (the package has excellent documentation which will guide you through setting up a [Pushover account](https://pushover.net/)).

I put this all in a plain R script named r-project-ssl-notify.R that’s then called from a Linux CRON job which runs:

/usr/bin/Rscript -e 'rmarkdown::render(input="PATH\_TO/r-project-ssl-notify.R", output\_file="PATH\_TO/r-project-cert-status/index.html", quiet=TRUE)'

Here’s the contents of

#' ---

#' title: "r-project SSL/TLS Certificate Status"

#' date: "`r format(Sys.time(), '%Y-%m-%d')`"

#' output:

#' html\_document:

#' keep\_md: false

#' theme: simplex

#' highlight: monochrome

#' ---

#+ init, include=FALSE

knitr::opts\_chunk$set(

message = FALSE,

warning = FALSE,

echo = FALSE,

collapse=TRUE

)

#+ libs

library(DT)

library(openssl)

library(pushoverr)

library(tidyverse)

# Setup -----------------------------------------------------------------------------------------------------------

# This env config file contains two lines:

#

# PUSHOVER\_USER=YOUR\_PUSHOVER\_USER\_STRING

# PUSHOVER\_APP=YOUR\_PUSHOVER\_APP\_KEY

#

# See the {pushoverr} package for how to setup your Pushover account

readRenviron("~/jobs/conf/r-project-ssl-notify.env")

# Check certs -----------------------------------------------------------------------------------------------------

# [r-project.org](http://r-project.org) domains retrieved from Rapid7's FDNS data set

# (<https://opendata.rapid7.com/sonar.fdns_v2/>) and cert transparency logs

#+ work

c(

"[cloud.r-project.org](http://cloud.r-project.org)", "[cran.at.r-project.org](http://cran.at.r-project.org)", "[cran.ch.r-project.org](http://cran.ch.r-project.org)",

"[cran.es.r-project.org](http://cran.es.r-project.org)", "[cran.uk.r-project.org](http://cran.uk.r-project.org)", "[cran.us.r-project.org](http://cran.us.r-project.org)",

"[lists.r-forge.r-project.org](http://lists.r-forge.r-project.org)", "[mac.r-project.org](http://mac.r-project.org)", "[r-project.org](http://r-project.org)",

"[translation.r-project.org](http://translation.r-project.org)", "[user2014.r-project.org](http://user2014.r-project.org)", "[user2016.r-project.org](http://user2016.r-project.org)",

"[user2018.r-project.org](http://user2018.r-project.org)", "[user2019.r-project.org](http://user2019.r-project.org)", "[user2020.r-project.org](http://user2020.r-project.org)",

"[user2020muc.r-project.org](http://user2020muc.r-project.org)", "[www.user2019.fr](http://www.user2019.fr)"

) -> r\_doms

# grab each cert

r\_certs <- map(r\_doms, openssl::download\_ssl\_cert)

# make a nice table

tibble(

dom = r\_doms,

expires = map\_chr(r\_certs, ~.x[[1]][["validity"]][[2]]) %>% # this gets us the "validity end"

as.Date(format = "%b %d %H:%M:%S %Y", tz = "GMT"), # and converts it to a date object

delta = as.numeric(expires - Sys.Date(), "days") # this computes the delta from the day this script was called

) %>%

arrange(expires) -> r\_certs\_expir

# Status page generation ------------------------------------------------------------------------------------------

# output nice table

DT::datatable(r\_certs\_expir, list(pageLength = nrow(r\_certs\_expir))) # if the # of r-proj doms gets too large we'll cap this for pagination

# Notifications ---------------------------------------------------------------------------------------------------

# See if we need to notify abt things expiring within 1 week

# REMOVE THIS or edit the delta max if you want less noise

one\_week <- filter(r\_certs\_expir, between(delta, 1, 7))

if (nrow(one\_week) > 0) {

pushover\_normal(

title = "There are r-project SSL Certs Expiring Within 1 Week",

}

# See if we have expired certs

expired <- filter(r\_certs\_expir, delta <= 0)

if (nrow(expired) > 0) {

pushover\_critical(

title = "There are expired r-project SSL Certs!",

message = "Check which ones: [r-project-cert-status](https://rud.is/r-project-cert-status)"

)

}

**FIN**

With just a tiny bit of R code we have the ability to monitor expiring SSL certs via a diminutive status page and alerts to any/all devices at our disposal.