OREGON STATE UNIVERSITY

CS 373

WINTER 2019

Week 7 Lab 2

Author:
Thomas Noelcke

Instructor:
D. Kevin McGrath

I. Google

I started by looking for web sites that contain the word google or sites that might look like google. I assigned a penalty to any sites that contain google in them. This is because it is one of the most visited sites on the internet. As such it is a likely target for spoofing and fakes of many kinds. To counteract this for actual google sites I made the penalty for having google in the name less than the threshold as just having google in the name doesn't mean that it is bad.

II. Alexa rank

Initially, I tried to assign a penalty for not having an Alexa rank. However, I found that this adversely impacted my error rate for detecting sites that weren't malicious but were labeled as such by my program. I ended up assigning no penalty for not having an Alexa rank but did assign a bonus for sites that were in the top million. This would also help counteract the google penalty I mentioned earlier.

III. Site Age

Initially, I set a penalty on any site that was younger that 180 days. However, I fond that I was still having errors on some sites that should have been detected but were not. I ended up setting a penalty on any site younger than one year old. This is because sites that are younger are generally more malicious than sites that have been around for a while.

IV. IP

If the site didn't return an IP address were also given a penalty. I gave these sites a penalty that would cause them to fail if they could not also earn a bonus some where else either via Alexa ranking or via scheme. This is because urls that don't return ip addresses tend to be malicious and I would like to assume that these are guilty until proven innocent.

V. scheme

One of the first things that I decided to do was to give a bonus to sites that use the https scheme. Using https won't keep a domain from being detected but rather gives them a little credit for at least trying. I did this because https does ensure that there is some security built in to the site. I didn't just pass these urls along because though this does add some security it also possible to buy a certificate and serve up malicious content that is signed.

VI. file type

I also assigned some penalties based on file type. I originally tried to asses a penalty for using JS files but I found this lead to a lot of false positives. For detecting JS code that is bad it may be better to use some other means of detection. I did however assign a heavy penalty to exe files as these are likely to contain malicious code. I also assigned a slight penalty to aspx, asp, xml and de files as these are another way that malicious content may be delivered. However these file types were not assesed a penalty large enough to disqualify them outright unlike exe files.

VII. The Code

```
import json, sys, getopt, os, re

def usage():
  print("Usage: %s --file=[filename]" % sys.argv[0])
  sys.exit()

def main(argv):

  file=''

  myopts, args = getopt.getopt(sys.argv[1:], "", ["file="])

  for o, a in myopts:
    if o in ('-f, --file'):
      file=a
    else:
      usage()

  if len(file) == 0:
    usage()

  corpus = open(file)
  urldata = json.load(corpus, encoding="latin1")
  numUrls = 0
  maliciousCount = 0
  actualMalicious = 0
  thresholdTotal = 400
  error = 0

  for record in urldata:
    threshold = 0
    numUrls = numUrls + 1
    malicious = 0
    regexGoogle = re.search("[^www\.)]google(docs|doc|drive|mail|plus|calendar)*", reco
    regexIp = re.match("^(\.[0-9][0-9]?[0-9])+$", record["host"])

    if regexIp:
      threshold = threshold + 500

    if record["scheme"] == "https":
      threshold = threshold - 300

    domainAge = int(record['domain_age_days'])
    if domainAge < 360:
      threshold += 600

    ext = record["file_extension"]
```

```
    if(ext in ["zip", "php"]):
      threshold = threshold + 400
    elif ext == "exe":
      threshold = threshold + 300
    elif ext in ["aspx", "asp" "xml", "de"]:
      threshold = threshold + 300

    if record["alexa_rank"] == None:
      threshold = threshold + 0
    elif record["alexa_rank"] < 100000:
      threshold = threshold - 300

    if regexGoogle:
      threshold += 700

    if record["malicious_url"]:
      actualMalicious = actualMalicious + 1

    if threshold > thresholdTotal:
      maliciousCount = maliciousCount + 1
      malicious = 1
    if malicious != record["malicious_url"]:
      print record["url"], malicious, record["malicious_url"]
      error = error + 1

  print "actual malicious:", actualMalicious
  print "identified malicious:", maliciousCount
  print "errors:", error

  corpus.close()

if __name__ == "__main__":
  main(sys.argv[1:])
```