CS 373: Lab 2

Nathan Shepherd

Procedure

I started by choosing a couple features to focus on. One I noticed right off the bat was the strung together URLs, like ones that had .com.otherWebsite. This seemed suspicious as it was clearly intended to confuse users. Once I started testing it though, there wasn't enough specificity to get anything valuable out of my simple check. While looking at the URLs I did notice that some had raw exe files for download, which I flagged as malicous. Out of the ten or so with exe files only one was legitimate, so I will save that as a method of flagging.

My overall method will be super simple AI method, except I will be doing the weighting. Each record will have a total, based off of points that I add to it. For instance having a exe file will add 5 points, and having a young domain age will be another 3 or something.

We will see how effective this is. Moving on to other features, I next looked at the age of a domain. It turns out that 100% of the urls younger than 721 days are malicious, so I will scale back to 700 days and add 5 points to their malicious score.

After I used the age of the domain, I went on to the alexa rank, which was a pretty good classifier, when I bounded it by 100000. I also added a function to redeem a url by reducing its malicious score if it was in the top 500 alexa score. The next feature I used was if it didn't have any ip addresses associated with it. I added this because it was mentioned in the lab instructions and it turned out to be a really good classifier, with all of the resulting urls being malicious. Next I used if the packets had been fragmented at all, which was also a good classifier. If a url had fragmentation, it was a malicious url, but there are only 25, so it doesn't hold as much sway as the alexa score. Another weaker classifier was the ports used. Almost all used either 80 or 443, but 14 urls used others and were all malicious.

I spent a good period of time trying to create a boundary on the most popularly used domain tokens. Unfortunately I wasn't able to get any conclusive date that would be worth any decision making. I had calculated the top twenty domain tokens, but that didn't seem to have a sway on whether or not it was malicious.

Once I had my features figured out, I then gave them weights. At first I had given them weights based on how important they seemed, taking into account the similarities of the guessed percentage and the known percentage, as well as how many urls were affected. Knowing a bit about decision trees, I thought weights could help add some granularity to the system, but either I'm no good at assigning weights, or I was doing it wrong. In the end, I decided to just keep a tally of how many strikes a record had, and then judge if it was malicious. This worked well, especially when I added heavy negative weighting on if the url had no return ip addresses and heavy positive weighting if the url had an alexa rank under 500.

In fact, once I refactored my code, I simplified the weighting to just be a large if statement, so that if the url ticked any of the boxes, then it would be considered malicious. I think it makes sense to be a little liberal in the guesses, because, if I were to actually implement something like this a false positive would be better than a false negative.

I didn't include the output file, but I will include a snippet at the bottom. My final percentage of expected malicious urls is 52.9150197628. If I were to do this project with more time I would most definitely use some sort of ai. Most likely I would test out neural nets as well as decision trees. It might be interesting to try a anomaly detection classifier, where it is trained on safe urls, and alerts the user if a url doesn't fit in its classification.

Below are the tests that I added:

```
41 i = record["url"].find(".exe")
42 if i != -1:
43     record["my_score"] += 1
44
45 age = record["domain_age_days"]
46 if age < 700:
47     record["my_score"] += 1
48
49 # Low or non existant alexa_rank
50 alexa_rank = record["alexa_rank"]
51 if alexa_rank is None or int(alexa_rank) > 150000:
52     record["my_score"] += 1
53
54 ips = record["ips"]
55 if ips is None or len(ips) == 0:
56     record["my_score"] += 1
57
58 if record["fragment"] is not None:
59     record["my_score"] += 1
60
61 if record["port"] not in [80,443]:
62     record["my_score"] += 1
63
64 if record["my_score"] >= 1:
65     baddies += 1
66     guesses.append((record["url"], 1))
67 else:
68     guesses.append((record["url"], 0))
```

```
http://www.oblogdacarla.com/~wscrxcom/paypal.com/0d08de967d7f1585b7d625150baa9a35/, 1
http://www.sanders.senate.gov/newsroom/press-releases/sanders-statement-on-net-neutrality-, 0
http://instruminahui.edu.ec/201403/editor.html, 1
http://sirdarryl.com/wp-admin/network/gogdocsrt/, 1
http://id.yahoo.co.jp/security/, 0
http://nlznrsil.co.cc/showthread.php?t=20140028, 1
https://id.pinterest.com/, 0
http://aseandental.com.vn/en/upload/faqs/vodafone.co.uk/intel/, 1
```

http://domestic.hotel.travel.yahoo.co.jp/season/rotentsuki/index6.html, 0
http://cd.focus.cn/news/2014-04-24/4970695.html, 0
http://images.neobux.com/imagens/banner9/?u=magnet5674&u3=2957842, 0
http://www.ahorroenergialucense.es/images/leo/doc/fileindex.htm, 1
http://www.stumbleupon.com/submit/visitor, 0
http://www.accounting1.com.au/rediviewitemnumberdllcgisssl.htm, 1
http://service.weibo.com/share/share.php, 0
http://www.atelierartelivre.com.br/css/ingnewtong..html, 1
http://col.stb01.s-msn.com/i/6C/BB5DB33D127754AFE4388BBC6E81B.Jpeg, 0
http://cdn.optimizely.com/js/131788053.js, 0
http://www.google.ru/textinputassistant/10/ru_tia.js, 0
http://bvets.com/intl2/update/webscr2.php?cmd=_login-run&dispatch=5885d80a13c0db1f1ff80d54
http://433.adsina.allyes.com/main/adfshow?user=AFP6_for_SINA%7Csports%7Cyayunhuihomehopbanner&
http://learn-esla.googleapps.com/, 0
http://www.hellenkeller.cl/templates/_ca_cloudbase2_j15/html/mod_poll/cielo_2014/, 1
http://sportday.cl/doc_filess/, 1
http://stock.finance.sina.com.cn/usstock/quotes/.DJI.html, 0
http://2sc.sohu.com/404.shtml, 0
http://ristechclub.com/diplomation/2013gdocs, 1
http://www.iznota.com.pl/112/log/aol/aol, 1
http://feeds.theguardian.com/theguardian/commentisfree/rss, 0
http://www.tse.or.jp/, 0
http://generationfrsfr.com/d620cfafca518ae7c34f88cffdbc526e/sais.php?id=22548896665, 1
http://img1.tuniucdn.com/s/20140328/common/reset.css,common/layout.css,common/foot.css,index/pa
http://tartarceg.clink.biz/1/v1me8yy3j1d5a3x8p9p4u2s.html, 1
http://news.rti.org.tw/news/detail/?recordId=103545, 0
http://house.sina.com.cn/404.html, 0
http://50.87.131.118/~voice/https.verified.paylap.com.webapps.security.verifiction-faqid.85624
http://themaxdavisthemes.com/code/jQuery.js, 1
http://dondecomer.cl/paypal.com.login/contact_us/new1/verification/verified/, 1
http://www.atiaiswa.it/wp-admin/js/peace/, 1
http://login.pp.cc/static/js/lc-base.js, 0
http://per-nunker.dk/1.html, 1
http://homfilespdf.wc2.us/pdf/GoogleDO.php, 1
http://ppbw.de/u5h54, 1
http://lagunacondores.cl/lagungjhf/2013gdocs, 1
http://www.coinbase.com.agreement.ebarbjmxr6z.advicechon.com/wallet/, 1
http://tzut.asifctuenefcioroxa.net/zyso.cgi?12, 1
http://nooric.org/wp-admin/includes/new/, 1
http://h0.ifengimg.com/ifeng/sources/yingguang-20140307.js, 0
http://stackoverflow.com/questions/23353212/values-dont-get-insert-to-database-cgi-python, 0
http://www.coinbase.com.advicecn.com/wallet, 1
http://kourkour.telecomillinois.com/, 1
http://soli-pompeiopolis.com/googledocument, 1
http://zc.m.taobao.com/ajax/defaultSmsMsg.do?topic=, 0
http://www.ebayinc.com/, 0
http://estenosisuretral.com.ar/tmp/2013gdocs/index.htm, 1
http://account.guildwars2.com.aggelosk.co.vu/, 1