EECS450 Internet Security

Project Progress Report #1

Yu Zhou, Shangping Liu

Project status:

At the beginning of the project, we chose to use a dummy database as our analyzing object. We are in the progress of designing a prototype algorithms that can find common strings within two files.

Initial results:

In order to design an algorithms to find candidate identifiers, we need to use the dummy database first, which need to exact the some sample data to analyze. In the beginning, we chose to only look at cookies.

the following steps extract cookies that embedded in a request which behaves as third-party request (send from one domain to a different domain).As the dummy database provide four data tables, we exported the four tables to the following four txt files:

pages.txt

http\_requests.txt

redirects.txt

request.headers.txt

1) Extract id\_rootid.txt, which contain the page id and corresponding root page id.

*note*: we write a script “rootpage.py” to track each page to its root page and return the root id.

2) Import id\_rootid.txt to database, as table rootid2

3) Query the database and export url\_location\_id.txt, which contain the url a request is sending to, and the location from where the request initiated.

*note: the sql query is as following:*

*“select url, location, test.http\_requests.id from test.rootid2 cross join test.pages cross join test.http\_requests where test.rootid2.id = test.http\_requests.page\_id and test.rootid2.root\_id = test.pages.id;”*

4) Compare the domain name of the url and location in the url\_location\_id.txt file, find the url and request is which its domain name does not match to the domain name of the page who send the request.

*note: we write a script “third\_party.py” to compare the domain name and output the file “third\_party. txt”*

5) Import third\_party.txt to database, as table thirdParty2 (third\_domain, id)

6) Query the database and get the cookies embedded in the request which sent to third party, grouped by third party domains. The output file is “cookie\_domain.txt”

*note: The sql query is following:*

*“select value, third\_domain from test.thirdParty2 cross join test.http\_request\_headers where test.thirdParty2.requestID = test.http\_request\_headers.http\_request\_id and test.http\_request\_headers.name = “cookie" order by third\_domain;”*

After we got the cookie\_domain.txt file, we then can proceed to attacking the task of finding candidate identifiers. We tentatively formulate our problem to be a Longest Common Substring Problem (LCSP), which is defined as to find the longest string (or strings) that is a [substring](http://en.wikipedia.org/wiki/Substring) (or are substrings) of two or more strings.

**Problem definition:**

Given two strings, S of length m and T of length n, find the longest strings which are substrings of both S and T.

For our case, since we observed multiple requests from same domain, we decided to generalize our problem to the classic **k-common substring problem**, which defined as : given the set of strings S = {S1, S2, …., Sk}, where length of Si = ni, and . Find for each 2<=*k*<=K, the longest strings which occur as substrings of at least *k* strings.

Problems encountered:

1. The comparison of the domain name. The domain name is consisted of