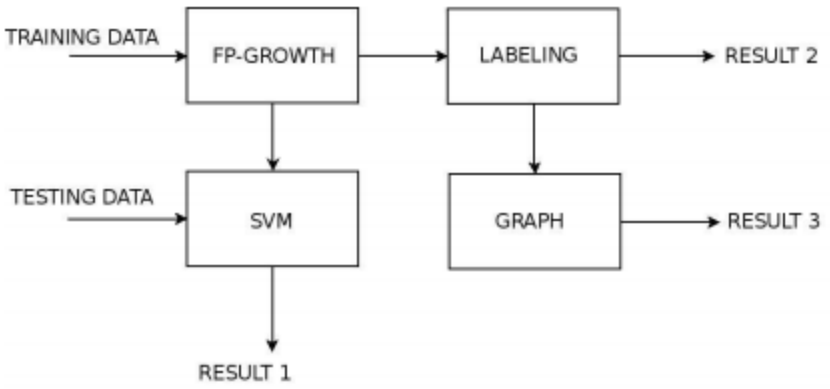
Senior C++ Developer - Test Code Audition

Test prerequisites

To complete this test you need possess good knowledge in Machine Learning, C++ programming.

Test requirements

Your test program will perform the task of identifying malware campaign.



1. FP-GROWTH

TRAINING DATA – attributes: "time" , "id" , "source" , "category" , "name" , "md5" , "ip" ,

"url" , "domain" , "asn" , "cc" , "details"

TEST DATA – attributes: "time" , "id" , "source" , "category" , "name" , "md5" , "ip" , "url" ,

"domain" , "asn" , "cc" , "details"

As a result, we get a list of tokens that goes on FP-Growth algorithm. Based on fp-growth algorithm is performed grouping tokens and their transactions in descending order.

In the FP-Growth algorithm we have the ability to add "suport" value which allows to set in count % (or what is the absolute value) minimum number of values will be discarded.

Finally results from the algorithm you should get the FP-Tree.

2. SVN

The next step is to reconstruct the URL from which it was built the resulting tree FP-Tree with some confidence interval. For this purpose consider URLs where the token was repeated more then x3 is classified as a campaign (assigning a class to "-1") while the rest has a class "+1". Then return to the input data and add these URLs all other attributes in the form of "time", "id", "source", "category", "name", "md5", "ip", "url", "domain", "asn", " cc ", "details". The resulting collection will be our test set of SVM machine learning. The kernel used to learn will be RBF. After learning classifier is built on the basis of which it is possible to test a set of test data and the estimation accuracy of the classification (RESULT 1).

3. LABELING

Based on the results of FP-Growth we give class to each URL associated with tokens from the tree and mark the labels of each campaign. Ie. coming out of the root nodes are the centers of the campaign tree and denote them as: K1, K2, K3, ..., Kn, and others - for not qualified under the tree. RESULT 2 is a table containing a list of URLs in the various campaigns and simple statistics relating to: the amount of tokens occurring in the campaign, the depth of occurrence of tokens in relation to the center of the campaign, etc.

4. GRAPH CORRELATION

Already last stage will see the creation of correlation between campaigns tree K1, K2, K3, etc. With other sets from the database. It should check all URLs associated with K1, K2, ... etc. if there are still in other database collections. To this end, we connect to the database and query the IP addresses of the data related to specific URLs. As a result you will get some data, then we distinguish them domain names and query the database again after these names -> build a graph of correlation that is the example.: IP_1 is correlated with the 3 domains. Then query the database and obtain the domains that associated with a given domain is 5 IP addresses. Then we build a tree correlation in which we count the frequency of individual addresses. Then further query the database of the domain associated with the new IP address, etc ... RESULT 3 is a correlation graph and table showing information on the campaign and correlated with their IP addresses and domains.

To complete your test assignment you have 24 hrs.

Things that are valued

Conventions, structure, how make the foundation of the app and organize files and folders through most important coding guidelines and quality.