Indian Institute of Technology Guwahati

System Programming Lab Report

Topic : ADFA-LD Dataset Pre-processing

Under the guidance of : Prof. Santosh Biswas

Submitted by :

- Mukul Verma , 150101038
- Piyush Jain , 150101046
- Shubhanshu Verma , 150101073

Contents

1 Introduction

Description of Australian Defence Force Academy-Linux Dataset (ADFA-LD) :

1. The dataset was generated on Linux local server running on Ubuntu 11.04, offering a variety of functions such as file sharing, database, remote access and web server.

2. Six types of attacks occur in ADFA-LD including two brute force password guessing attempts on the open ports enabled by FTP and SSH respectively, an unauthorised attempt to create a new user with root privileges through encoding a malicious payload into a normal executable, the uploads of Java and Linux executable Meterpreter payloads for the remote compromise of a target host, and the compromise and privilege escalation using C100 webshell. These types are termed as Hydra-FTP, Hydra-SSH, Adduser, Java-Meterpreter, Meterpreter and Webshell respectively.

3. 833 and 4373 normal traces are generated for training and validation respectively, over a period during which no attacks occur against the host and legitimate application activities ranging from web browsing to document writing are operated as usual.

2 Task Steps

Following steps have been performed to do the task :

1. We split the **A**ttack data for each category into 70% training data and 30% test data. For that, first 7 folders have been used for training data and last 3 for test data. For **N**ormal data , files in "Training_Data_Master" folder have been used for training data and files in "Validation_Data_Master" have been used as test data.

2. For each type of attack. the python script first combines all the training data files into one list, then the frequency all the unique n-grams have been found for them as well as for Normal training data files. Concatenating entire training file for a particular class of attack and then running the script on the concatenated file instead of running it individually on each file saves time.

3. For finding n-grams , a list of all possible values of n is maintained. For general case, we have included only one element i.e. n = 3, the list can be extending just by adding other elements like 5 or 7. And the same program will work because the whole process iterates over all elements of n.

4. For calculating the frequencies of all n-grams, the **n**grams_freq function first takes three consecutive elements of the list (of concatenated training data files) and makes it a string. In this was, the whole list is iterated and such strings are compared to get the frequency of n-grams.The function returns a dictionary of n-grams frequencies. This also saves our time comparing 3 elements individually.Similarly, the script finds the unique n-grams frequency for Normal files.

5. For all the category of attacks and the Normal, top 30% most frequent n-grams are selected. For that, the whole n-gram frequency dictionary is sorted and then top 30% have been picked. These n-grams are used as the features to create the dataset.

6. Now, the training dataset is created. File **t**rain.csv is created. The frequencies of n-grams which are in features list are found from each and every files which are in training data individually. A row is added in the train.csv file corresponding to each file. Corresponding label has been added for each data.

7. Now, for creating the test dataset, first **t**est.csv file is created.And same thing is done on each of the files individually. And one row is added for each file processed.

3 Statistics

For 3-grams :

1. Total number of features : 4142

2. Total size of training dataset : 1338

   Distributed as :

   - AddUser : 56
   - Hydra_FTP : 115
   - Hydra_SSH : 117
   - Java_Meterpreter : 84
   - Meterpreter : 55
   - Web_Shell : 78
   - Normal : 833

3. Total size of test dataset : 4613

   Number of test data derived from each class :

   - AddUser : 35
   - Hydra_FTP : 47
   - Hydra_SSH : 59
   - Java_Meterpreter : 40
   - Meterpreter : 20
   - Web_Shell : 40
   - Normal : 4372