You may work in groups of 2 on this homework.
Due date: 4:00pm on Tuesday 10/21

Question 1. Botnets (10 points)

Find a botnet from the past 15 years and research its

a. General stats: how many machines, purpose, total network traffic
b. Method of C&C
c. Method of replication (pseudocode level)
d. Types of attacks launched and the attacks' victims. What would you do to defend against this botnet if your personal computer or personal subnet became victim?
e. How would you preliminarily fix the OS or network protocol exploit that made the botnet possible (from either replication or C&C perspective)?
f. How, if at all, was the botnet mitigated or stopped?

Question 2. Anomaly Detection via "Eigenface" of Command History (30 points)

In this problem, we will distinguish malicious users from benign users using the history of bash command using an algorithm inspired by the eigenface algorithm for face recognition. The details of the algorithm can be found at
http://web.stanford.edu/class/cs259d/hw/AnomalyDetection.pdf and
http://web.stanford.edu/class/cs259d/hw/AnomalyDetection2.pdf

The idea behind the algorithm is to construct connected graphs of bash command flows for each user. If the flow becomes unusual, then we detect this as a malicious user. The algorithm works in several stages. Over all sequences in the dataset, we perform Principal Component Analysis (PCA) to come up with a new coordinate system represented by a set of eigen co-occurrence matrices. Co-occurrence matrix is analogous to the face in image analysis while the eigen co-occurrence matrix is analogous to the eigenface. To find the features associated with a sequence, the algorithm projects the co-occurrence matrix onto the space defined by the Eigen co-occurrence matrices. In addition, the paper treats each principal component feature in order, calling each principal component feature "a layer".
Dataset. Available at http://web.stanford.edu/class/cs259d/hw/AnomalyDataset.zip. Dataset contains UserXX files that contain 15,000 bash commands for User XX. The first 5000 entries in each file are training entries; they are guaranteed to have been entered by User XX. The rest of the file is test data and is divided into sections of 100 entries that either belong to user XX or to someone else.
The file ref.txt provides the key for the rest of the file. It is a 100 x 50 matrix. Each column represents a User index and each row represents whether the segment of 100 commands has been entered by the user (labeled by 0) or by a masquerader (1). For example, with our counting starting at 1, rather than 0, [row 2, col 3] = 0 means that the commands 5100 - 5200 in file User3 have actually been entered by User 3.

a. Divide each input into a series of sequences of L = 100. Construct the co-occurrence matrix for each user (for each sequence). Choose a window size *w* for your experiments. What are the values for *m* (total # of commands) and *n* in the provided dataset? For users 1 and 2 what is the co-occurrence of “rm” with “ls” (ls following rm) for 5 of the 50 sequences (choose the sequences)?
b. Normalize the matrices to be centered at 0, so that co-occurrences with less than average frequency would have negative values. What type of mean calculation did you choose? What are the values for user 1 and 2 for relationship between “rm” and “ls” for those sequences you chose earlier?
c. Calculate Covariance Matrix by first rearranging the matrices from (b) into a vector.
d. Find the eigenvectors of the covariance matrix and sort them by the eigenvalues. Plot the contribution rate (equation 8, AnomalyDetection.pdf) to find a good dimension of the feature space (N). Note that each eigenvector is semantically a matrix converted into a vector -- so it can be transferred back into a *m* x *m* matrix. These matrices is what we call “Eigen co-occurrence matrices.” [See first revision at the end of the doc]
e. Find the feature vectors for users 1-5 and report the feature vectors for users 1 and 2 for their first 5 sequences. Report these in user1FV.csv and user2FV.csv
f. Construct the network layers described in the paper for users 1-5 (for each sequence) -- either Layered Network Model or Combined Network Model. Specify which you choose.
g. For each test sequence for users 1-5, evaluate the algorithm by constructing co-occurrence matrix and mapping it into Eigen matrix space. Then compare the network similarity to the networks you have associated with that user. Classify each sequence as anomalous or friendly. Report back the false positive and false negative rates (you can include more users for extra credit). This section is open to experimentation and you are free to deviate from the authors’ testing strategies.
h. Classify the test sequences of user 21 as anomalous (1) or benign (0) and submit it as a user21.csv file with 100 entries (1 classification for each test sequence).

README section. Since this problem is based on a research paper, there are ambiguities both in algorithm and parameter selection. Please note the assumptions that you made, which parameters you chose, and how you tweaked them to come up with the best result. In the same README section, specify what you thought was the hardest part of this problem. Also, if you are working in a group, specify how you divided up the work. It’s ok to not reach the accuracy results in the paper.

Question 3. Continuous Authentication via Biometric Behavior (50 points)

This is an open ended question, intended for you to explore and implement algorithms in biometric behavior classification.
Provided is a dataset of users typing in a password:
http://web.stanford.edu/class/cs259d/hw/KeyboardData.csv

The data set was modified from the standard dataset available on the CMU website: http://www.cs.cmu.edu/~keystroke/
The test dataset is found at http://web.stanford.edu/class/cs259d/hw/KeyboardTestData.csv
The fields are the same, but session and rep number have been filled with dummy zeros.

The goal is to correctly label whether [userXX, keystroke data] actually belongs to userXX based on historical evidence. We will provide a test data set for you to label.

1. What is your general approach to this problem?
2. What features did you select and why?
3. Analyze the successes and failures of your algorithm. Provide an ROC curve showing the tradeoff between detecting true positives vs false positives.
4. Submit answer.csv in response to the test dataset, labeling *each row* as coming from that user or not (Notation: 0 for real user, 1 for masquerader). The test dataset contains 3 independent rows per user, so you should provide a label for each of the rows.
5. Extra credit: for those that you selected as masquerader, provide top 5 users that are closest.

Rubric:
40/50 points based on the report, covering the algorithms you've tried and analysis
10/50 points based on the results that you got

Submission

Email cs259d-aut1415-staff@lists.stanford.edu with an archive (tar or zip) that has a structure resembling the following tree. Include a subject line "Submission: SUNET, SUNET".

README.txt -- your SUNET(s) + amount of time you spent + thoughts on the homework.
p1/answers.pdf -- contains answers to problem 1 questions. Any common format (.pdf, .txt, .doc) is fine

p2/answers.pdf -- contains answers to problems 2a, 2b, 2d, 2e, 2f, 2g*, and an addendum** (again, any format for the file is fine)
/user1FV.csv -- comma separated feature vector for user 1 (problem 2e)
/user2FV.csv -- comma separated feature vector for user 2 (problem 2e)
/user21.csv -- should contain 100 comma separated entries with anomalous (1) or benign (0) labels for user 21
/bin -- directory that contains any runnable code you wrote.
/bin/README.txt -- few sentences describing how to run your code, including any library dependencies
*2g is a whole analysis section for comparing networking similarity for users 1-5 and any results you got
**Addendum describes any assumptions you made, any ambiguities that you resolved, and any difficulties you encountered. Feel free to vent any frustrations about this problem (or life). Then think some happy thoughts because you are done! What's the first thing that came to your mind?

p3/README.pdf -- analysis for problem 3
/answer.csv -- 153 entries separated by comma; 0 for real user, 1 for masquerader
/ec.txt -- (extra credit) 153 lines with top comma-separated 5 users for each row that you labeled as masquerader
/bin -- directory containing any runnable code you wrote

/bin/README.txt -- how to run your code

Good luck!

Revisions

(for keeping a log of revisions to this homework description)

1. *Question two requires a lot of memory. Please feel free to approximate the data by sampling to reduce the number of commands in the dataset. The important part is to implement the algorithm and have it work on data. If the provided dataset takes too long, please describe how you reduced it.*
2. *Changed “cd” to “rm” when looking for co-occurrences in question 2.*
3. *Test Data Posted*
4. *Submission Instructions Posted*