CS5435: Security and Privacy concepts in the wild

Fall2014

Homework Assignment #4 Write-Up

Cracking Honey Words

Jai Chaudhary

jc2855@cornell.edu

Attack Strategies

After manual inspection of Different Honeyword sets of different teams, we have come up with the following attack strategies. The strategies are in increasing order of sophistication.

- **Repeat Honeywords –** We found many of the sets by different teams had repeat honeywords. These are highly likely candidates for passwords given they are password like.
- **General Pattern for User Generated Password–** Using the RockYou password aggregate statistics (41% passwords are lower case, 16% numbers only, 4% have special characters) as prior information.
    - We evaluate the posterior i.e. probability of a honeyword being a passwords with naïve bayes assumpitons .for eg:
        - Probability given its lower case = 0.41
        - Probability given it has special characters = 0.04
        - Probability of - *notacommonpassword!@#* Being a password = 0.41 * 0.04
    - Note: These probabilities are approximate and sort of represent a penalty for a hard to remember pattern
- **For first 100 test cases generated by algorithms trained without examples**:
    - The strategy is that the transformations to generate honey words are character level. That is,
        - Substitution of similar looking digits (like a and @, 0 and O)
        - Addition of punctuation and digits
        - Truncation of characters
    - So, we compute most common character at every position. The password is the honeyword with maximum number of such common characters.
- **Clustering**
    - The idea is to compute similarity distance (Hamming or Levenshtein) between pair of honeywords and
    - Then we perform clustering to do 2 things
        - Filter out Random sequence of characters
        - Cluster the Honeywords based on the sugarword they were generated from.
    - The clustering is done using scikit-learn's different clustering algorithms: Spectral Clustering, Agglomerative Clustering
    - The sugarwords are the center of clusters thus obtained. We choose the password as sugarword with largest cluster
- **Hybrid**
    - All above strategies are combined to come up with the most probable password that they have (approx) consensus on

- **Smart Distance:** One way to improve the accuracy is to incorporate in the distance measure, user tricks (like: a ←→ @)by assigning such substitutions with lower distance. Another strategy is to include wordnet dictionary to compute the synonymous distances.

Bonus: Honeyrides

The strategies for honeyrides abstractly follow from the same abstraction as above. Given a set of 100 suspected honeyrides , we compare honeyrides by the space-time distance between the points

$$s^2 = \Delta r^2 - c^2 \Delta t^2$$

where c unlike in relativistic physics is the average speed of bicycle i.e. 15kmph.

- The space-time intervals are then summed across all points in the ride.
- We then perform clustering same as above
- The cluster with lowest entropy is the one with most likely honeyrides.
- The honeyrides are the rides from that cluster.