Malware Classification and Triage Problem Set

Description

The goal of this problem set is to develop a malware clustering system suitable for sample triage. In particular, you will implement a version of the BitShred feature hashing system.

To complete the problem set, you will need to ssh to your container at `$user@amplifier.ccs.neu.edu:$port` , where `$user` is your gitlab username and `$port` is your assigned ssh port (https://seclab-devel.ccs.neu.edu/snippets/6). Authentication is performed using any of your uploaded ssh public keys in gitlab.

Important Information	
Available	Fri 17 Apr 20:00 EST
Submission Deadline	Sat 25 Apr 18:00 EST

Sample Execution

The data set you will use to evaluate your clustering system is located on your container at `/usr/local/share/samples` . A JSON document at `/usr/local/share/samples.json` indicates the arguments you should use to execute each sample, should you choose to do so.

Note

These samples are not actual malware. It should be safe to execute them on your container using the provided arguments.

Feature Extraction

For each sample, you will need to extract a feature vector. The feature vector you use is up to you. For instance, one approach you can use is to extract system call sequences and arguments using `strace` and the provided sample arguments.

Feature Hashing

Next, you will need to implement feature hashing. For each sample's features, create a fingerprint using the hashing function of your choice. For further details, refer to the lecture notes and the original paper (/assets/refs/jang2011bitshred.pdf).

Using the sample fingerprints, compute a distance matrix that represents the pairwise Jaccard distance for all samples.

Sample Clustering

Using the machine learning library of your choice (or, alternatively, your own implementation), perform agglomerative hierarchical clustering on the fingerprint distance matrix. The result should be a dendrogram that indicates the sample clustering hierarchy.

Use a threshold to identify a cut in the dendrogram that represents the most likely set of sample clusters.

Answer Submission

Create a repository in gitlab at `git@seclab-devel.ccs.neu.edu:$user/prset07.git` . Commit your clustering system to `clustering/` , and include an executable script at `clustering/cluster` that runs your system with the following command-line interface:

```
$ ./cluster $path_to_configuration
```

The configuration file should contain a set of pre-computed feature vectors for each sample on your container in the file format of your choice. These should be the original vectors, *not* fingerprints.

The output of your tool should be the most likely set of clusters in JSON format:

```
{
    "clusters": [
        [<sample_c1_1>, <sample_c1_2>, ...],
        [<sample_c2_1>, <sample_c2_2>, ...],
        // ...
    ]
}
```

Problem Set

Description

Sample Execution

Feature Extraction

Feature Hashing

Sample Clustering

Answer Submission

Extra Credit

Links

Course Overview (/course/2015/spring/cs5770)

For example:

```
{
    "clusters": [
        ["0000", "0001", "0002", "0003"],
        ["0004", "0005", "0006", "0007"]
    ]
}
```

NOTE: Your tool *must* be executable using the above interface from a fresh git checkout of your repository to receive full credit.

Also, commit a `README.md` that describes in as much detail as possible the following:

- The features that you extract from the sample set
- The feature hashing strategy you use
- The criterion you use to choose a cluster set

Extra Credit

For extra credit, implement co-clustering. Modify your tool's output to the following:

```
{
    "clusters": [
        {
            "samples": [<sample_c1_1>, <sample_c1_2>, ...],
            "features": [
                <shared_feature>,
                // ...
            ]
        }
    ]
}
```

Add to your `README.md` a description of your co-clustering implementation.

Updated Fri 07 Aug 2015 10:38 EDT
Revision master/52d826a

bootstrap (http://getbootstrap.com/) — ember (http://embe