Transparent Computing (TC) Data Annotation Stack (DAS)

The purpose of this deliverable is to enable researchers and interested members of the scientific community to visualize the Transparent Computing (TC) Engagement 5 attack data in meaningful ways. By coupling real system event data with the attack write-up from the organization administering these demonstrative attackks, the data can be used to paint a picture of what an attack looks like with respect to a systems' vital signs. Through this lens, patterns in the data can be identified and used to further the ability of software to mitigate attacks on computer systems.

Background

The TC DAS is derived from the "ELG" stack (based on the ELK stack) - an implementation of the open-source tools Elasticsearch, Logstsash, and Grafana. These tools coupled together provide a wholly integrated platform for consuming high volumes of data, adding context to this data, and providing a way to view the data in the context of its provenance.

Getting Started

System Requirements

- Docker Engine version 17.05+
- Docker Compose version 1.12.0+

Directory Structure and Contents

- `docker-compose.yml`: A file containing the *docker-compose* configuration used to start, stop, and manage the multiple containers used for the DAS.
- `grafana_var_data`: A directory that exists as a Docker volume for the Grafana container. This stores all data used and stored by the Grafana interface.
- `elasticsearch`: A directory that contains configuration for Elasticsearch and all data stored in the Elasticsearch index as a Docker volume.
- `logstash`: A directory that contains the configuration for the Logstash service as well as settings for data ingestion into the Elasticsearch instance.
  - `logstash/config/logstash.yml`: The file containing configuration options for the logstash service
  - `logstash/pipeline/logstash.conf`: The configuration file for the logstash input settings which allow for CDM records to be consumed into the DAS.
- `env.grafana`: Environment variables used by Grafana at runtime

Running the Data Annotation Stack

In this directory, run the following: `$ docker-compose up -d`

Once the stack is up and running, you can navigate to `[host]:3000` in your browser. To check the startup status of any of the three services, you can run the `docker logs` command with the name of the service, either `logstash`, `elasticsearch` or `grafana`. Using the flags `-ft` will allow you to follow and tail the logs of these services.

Once you reach the Grafana login screen, the username and password are both `darpa`.

Adding Data for Annotation

Out of the box, this tool comes pre-loaded with data from three attacks on hosts. This annotated data is intended to demonstrate the capabilities of this platform before you add your own annotated data, or import that of others.

Loading New Data for Annotation

Importing new data is done using Logstash's input plugin functionality. The import functionality provided by Logstash allows you to consume event data in nearly any format and translate it into that which is stored in Elasticsearch. We have pre-configured Logstash to take in event data as output from `log4j`, but also support reading in from a file.

Loading data from compressed files (preferred)

This guide assumes that you have compressed `.bin.gz` files containing serialized CDM data from previous engagements. An example name for a file like this might be `ta1-cadets-1-e5-official-2.bin.1.gz`.

1. Prepare a directory with a collection of `.bin.gz` files
2. Run `import.sh` from `/imports/from_file/` with the following parameters: `$ ./import.sh [directory] [avro file] [host] [port] -v` The paramaters break down to the following:

   - `[directory]`: The directory where you have multiple `.bin.1.gz` files
   - `[avro file]`: the AVRO schema that will be used to deserialize the serialized data into the appropriate JSON format. This is provided as `TCCDMDatum.avsc` in the same directory as the script.
   - `[host]`: the host address of the system hosting the docker containers for the DAS
   - `[port]`: the port for Logstash, where the default is `4712`

   Example: `bash $ ./import.sh /.../tc_data_delivery/ /.../TCCDMDatum.avsc 0.0.0.0 4712`

Loading Existing Annotations for Data

Annotations are stored separately from the data being annotated. This allows for data to be loaded on demand and have additional annotations piped in as needed. If one user creates annotations in the TC DAS and wants to provide these annotations to another user, the two users must coordinate to initialize two instances of the TC DAS. Once this has happened, the users run a script which copies annotations from instance A to instance B.

Prerequisites: In order to run this script, you should have Python 3 installed on your system.

1. Create two running instances at arbitrary host `host-1` and arbitrary host `host-2` (see 'Getting Started' above).
2. Use the pre-configured API keys for the TC DAS to run the annotation importer script from `./importers/annotations/` as below:
   `bash $ ./import_annotations.py -ga [host-1]:3000 -gb [host-2]:3000 -bk eyJrIjoiU0Z4bWdTNjBTaDFHY3oxU21CZFVoaHRWSmF6RThQTG4iLCJuIjoiS0VZIiwiaWQiOjF9 -ak eyJrIjoiU0Z4bWdTNjBTaDFHY3oxU21CZFVoaHRWSmF6RThQTG4iLCJuIjoiS0VZIiwiaWQiOjF9 --verbose`

Replace the host names above with the appropriate IP address for your Grafana instances.

Navigating Grafana

All views in Grafana are referred to as "Dashboards" and are composed of a variety of interactive panels. The TC DAS uses multiple dashboards with unique panel layouts recommended for completing various tasks in the DAS. To view the Dashboards described below, navigate to the Grafana Home screen and select any of the Dashboards.

Annotation Creation

The "Add Annotations" dashboard is designed to facilitate adding annotations to the data based on the Ground Truth documentation inputs. By navigating the graph above, one can add annotations to the entries on screen.

Annotation Review

The "Review Annotations" dashboard is designed to allow a user to review existing annotations and attempt to draw meaningful conclusions from the data presented.

Detailed Event Viewer

The "Timeslice Viewer" dashboard displays detailed JSON data for all events occuring at a specific timestamp. This dashboard is linked from the tables in the Annotation Creation and Annotation Review screens.

Version 0.1.2 - November 1, 2019