Data Visualization System

Introduction

The purpose of the Data Visualization System is to a tool that provides the user with an integrated view of the data generated by the Evaluator Centric and Extensible Logger daemon (ECELd) and the ability to tag and modify the data associated with a capture taken by ECELd.

Problem statement:
Data is a critical asset in cybersecurity; it helps researchers and practitioners develop novel technologies based on real past incidents. However, this asset is lacking, and even more, tools that specifically focus on the analysis of data are far and few between. The Evaluator Centric and Extensible Logger daemon (ECELd) is a tool that attempts to reduce this gap. The tool allows users to collect data for the purpose of analysis. Collected data include network data, keystrokes, system calls, and screenshots, among others. Additionally, the ECELd includes a wireshark component which allows users to annotate the network data using the popular packet sniffer application. Both the ECELd tool and the ECELd-wireshark component are integrated into the ECELd-netsys.

Team Members:
- Briana Sanchez
- Stephanie Medina
- Luisana Clarke
- Rocio Cardona
- Bianca Alvarez
- Dima Abdel Jaber

Client:
- Dr. Jaime Acosta

Guidance Team:
- Dr. Salamah Salamah

Description

The system consists of two primary components: the packet view shows ECELd-Wireshark and the timeline view shows the parsed ECELd-collected data using a horizontal layout.

These two windows combined will allow for the user to have a better visualization of what is being analyzed. Many features are integrated in this system in order to bring better visualization and each feature will be specific to the type of data that is going to be displayed. In this system, each specific set of data will be considered a dataline. For example, a mouseclicks dataline will contain different information than what a keypresses dataline contains, which will result in specific features for each. Moreover, with different datasets, there must be a way to be able to distinguish between each of them, so each dataline will have an assignable color in order to make it easier to tell them apart. This color assignment shall be displayed in both the timeline view and the packet view. Finally, the most important feature of this system will be synchronization. The synchronization feature shall be enabled and disabled depending on the user's preference. The main purpose for this feature is to allow the user to use both timeline view and packet view at the same time and be able to see the data that is correlated to where the user is currently analyzing.

Installing the Data Visualization System (DVS)

The Data Visualization System's source code can be found in the following link: https://github.com/smedina7/DVS. The system works in both Linux and Windows. Follow the instructions below for installation.

Windows Installation

The first step of installation is to clone the repository provided in the link above and navigate to the project folder. Users have two options, install and run the program in a virtual environment or in their local environment.

Virtual Environment

If you wish to run the program within a virtual environment install Python's Virtual Environment Builder by typing the following command (if it's not already installed):

```
C:\Users\DVS > pip install virtualenv
```

After installation is done, you will create a virtual environment and activate it:

```
C:\Users\DVS > python -m venv dvs-venv

C:\Users\DVS > dvs-venv\Scripts\activate
```

Install required dependencies:

```
(dvs-venv) C:\Users\DVS > pip install -r requirements.txt
```

Lastly, run main.py to start the DVS GUI.

```
(dvs-venv) C:\Users\DVS > main.py
```

Local Environment

If you wish to install the program in your local environment, run the following:

```
C:\Users\DVS > pip install -r requirements.txt
```

Once installation is done, run main.py to start the program.

```
C:\Users\DVS > main.py
```

Linux Installation

The first step of installation is to clone the repository provided in the link above and navigate to the project folder. Users have two options, install and run the program in a virtual environment or in their local environment.

Virtual Environment

If you wish to run the program within a virtual environment install Python's Virtual Environment Builder by typing the following command (if it's not already installed):

```
kali@kali:~/DVS$ sudo pip install virtualenv
```

After installation is done, you will create a virtual environment and activate it:

```
kali@kali:~/DVS$ python3 –m venv dvs-venv

kali@kali:~/DVS$ source dvs-venv/bin/activate
```

Install required dependencies:

```
(dvs-venv) kali@kali:~/DVS$ pip install -r requirements.txt
```

Lastly, run main.py to start the DVS GUI.

```
(dvs-venv) kali@kali:~/DVS$ python3 main.py
```

Local Environment

If you wish to install the program in your local environment, run the following:

```
kali@kali:~/DVS$ sudo ./installDeb.sh
```

Once installation is done, run main.py to start the program.

```
kali@kali:~/DVS$ python3 main.py
```

Using the Data Visualization System (DVS)

The Data Visualization System has two main components: Packet View and Timeline View. This section will provide a tutorial on basic usage of each component. Furthermore, users will be able to use the DVS as a standalone program or integrated as part of ECELd.

Standalone Version

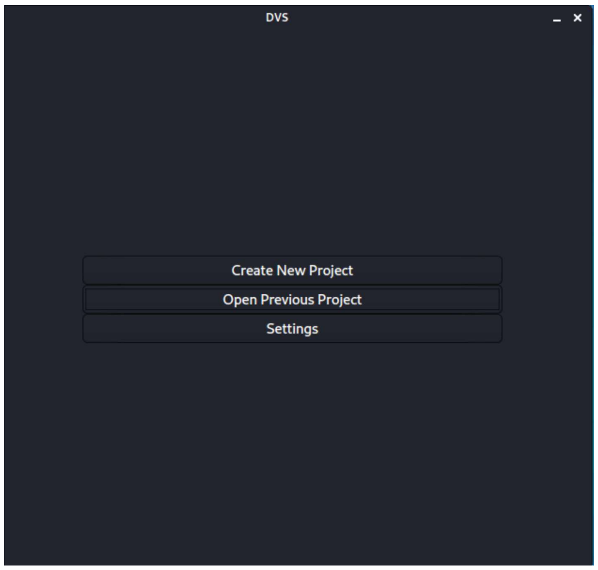
In the standalone version, users will be able to choose from what folder they'll be analyzing their data. In the event that the user already has the data that they need, this gives the flexibility to just do the data visualization and analyzation without having to open ECELd-NetSys.

Iteration 1 Components:

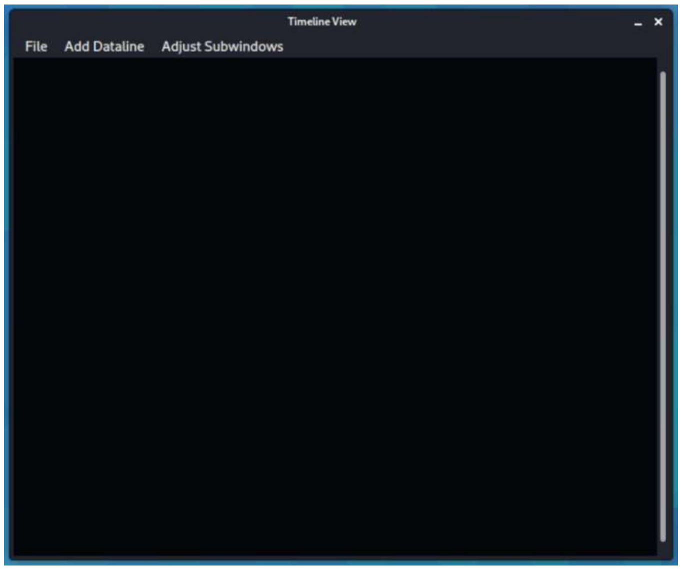
For the first iteration of the DVS system, the team focused on getting the Timeline View set up. This means that the packet view will not appear yet for this iteration. The reason for that is because the packet view is already implemented, and we will just need to combine the two windows once the timeline view is ready. The following sections will review what are the available features for iteration 1.

Start Window

Once the DVS is started, the following window will appear:

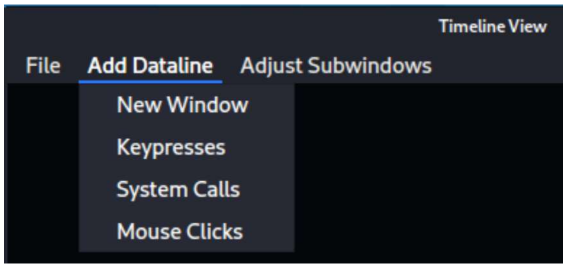


Only 1 out of the 3 buttons will take you to the timeline view. The settings button will only display a message indicating that it hasn't been implemented yet. The "Open Previous Project" a file browser should appear, however, for now no file or directory could be selected. Finally, the "Create New Project" is where it'll take you to the features that were added. This button will change the window to look like the following:



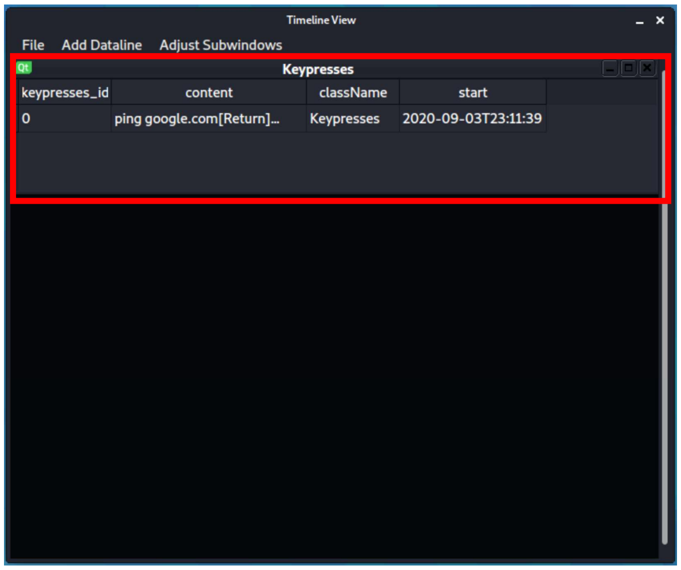
Adding Datalines

The "Add Dataline" menu option is what will be the focus for this window. When selecting that menu option, the following selections will be available:



Keypresses

The first dataline option is keypresses. When the user selects this option, the following dataline will be shown in the main window:

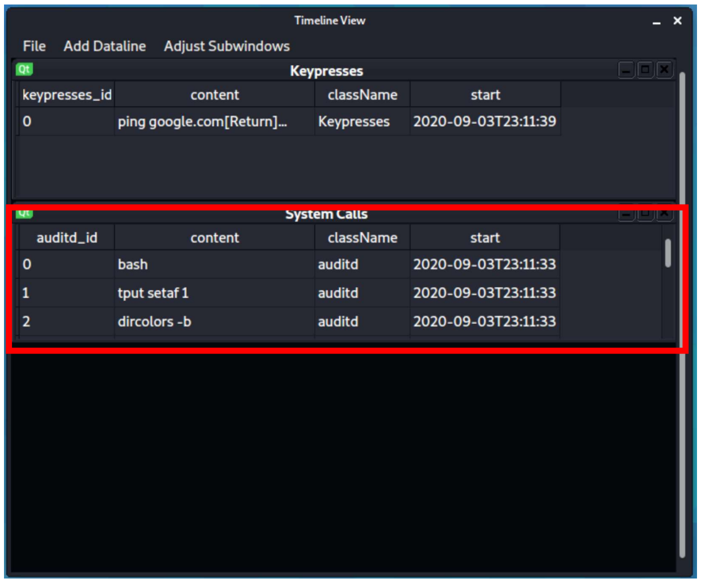


keypresses_id	content	className	start
0	ping google.com[Return]...	Keypresses	2020-09-03T23:11:39

Here, a table containing the parsed data from <json file> will be shown. This table includes the id of the keypress (keypress_id), the content, className and the timestamp.

System Calls

The second dataline option is the system calls. When the user selects this option, the following dataline will be shown in the main window:



Keypresses

keypresses_id	content	className	start
0	ping google.com[Return]...	Keypresses	2020-09-03T23:11:39

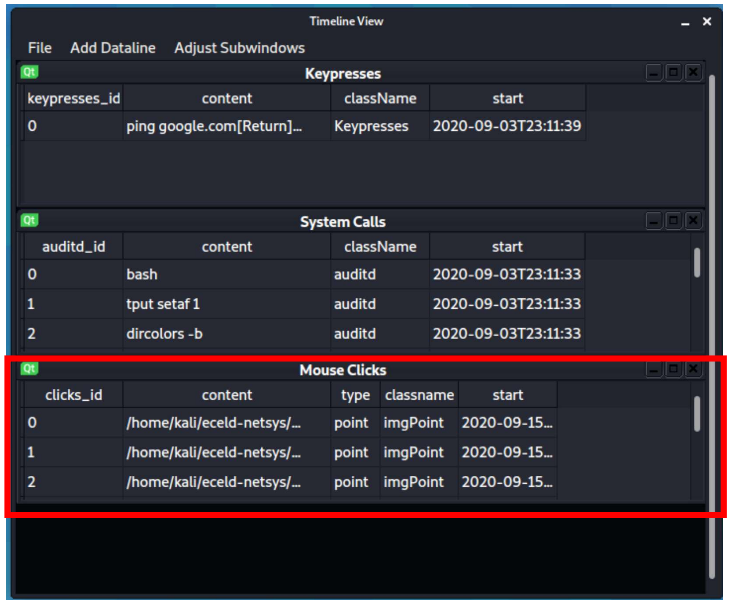
System Calls

auditd_id	content	className	start
0	bash	auditd	2020-09-03T23:11:33
1	tput setaf 1	auditd	2020-09-03T23:11:33
2	dircolors -b	auditd	2020-09-03T23:11:33

Here, a table containing the parsed data from <json file> will be shown. This table includes the auditd id, the system call, className and timestamp in which the system call was used.

Mouseclicks

The third dataline option is the mouseclicks. When the user selects this option, the following dataline will be shown in the main window:



Keypresses

keypresses_id	content	className	start
0	ping google.com[Return]...	Keypresses	2020-09-03T23:11:39

System Calls

auditd_id	content	className	start
0	bash	auditd	2020-09-03T23:11:33
1	tput setaf 1	auditd	2020-09-03T23:11:33
2	dircolors -b	auditd	2020-09-03T23:11:33

Mouse Clicks

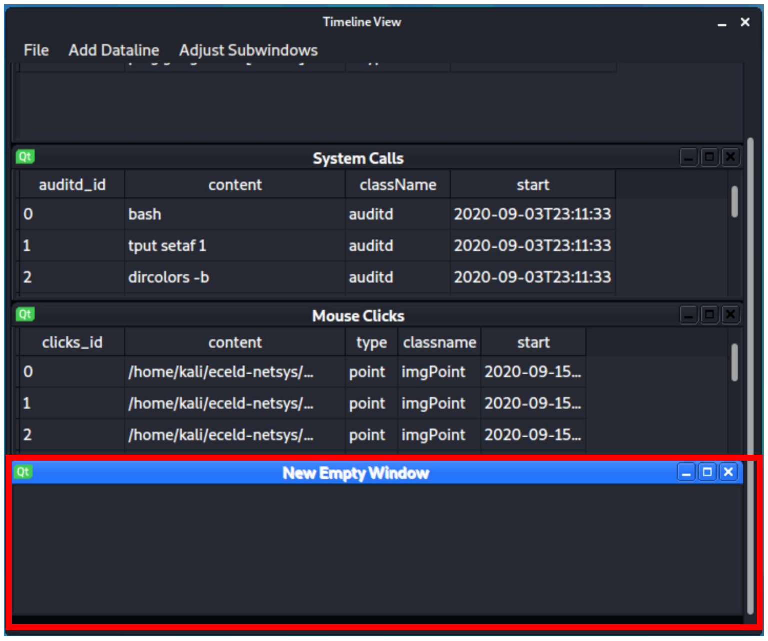
clicks_id	content	type	classname	start
0	/home/kali/eceld-netsys/...	point	imgPoint	2020-09-15...
1	/home/kali/eceld-netsys/...	point	imgPoint	2020-09-15...
2	/home/kali/eceld-netsys/...	point	imgPoint	2020-09-15...

This table includes the click id, the path to where the screenshot is saved, the type, className and the timestamp of the click. Enlarging each image individually is not implemented in this iteration.

The user is able to open, close and minimize each dataline window individually as well as displaying a combination of the datalines available at the same time.

New Window

This option is just to display a blank window and show what happens once the main window is full. Once there is more than 3 windows open in the main window, the scroll bar will allow for one to scroll between the datalines.

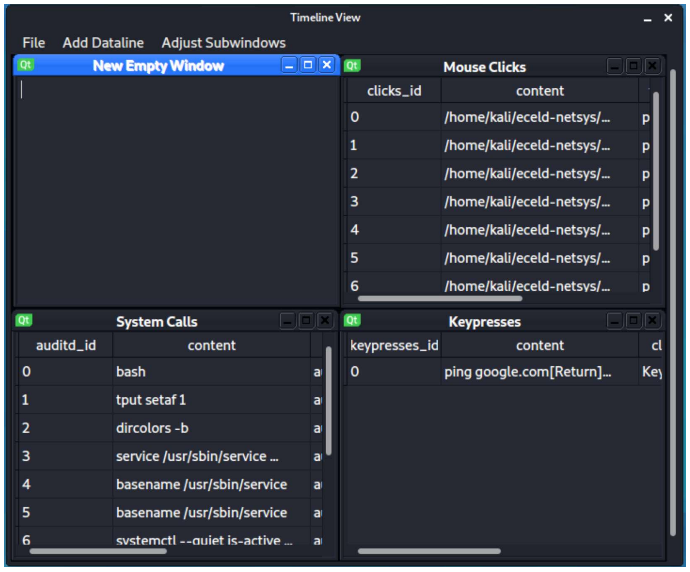


auditd_id	content	className	start
0	bash	auditd	2020-09-03T23:11:33
1	tput setaf 1	auditd	2020-09-03T23:11:33
2	dircolors -b	auditd	2020-09-03T23:11:33

clicks_id	content	type	classname	start
0	/home/kali/eceld-netsys/...	point	imgPoint	2020-09-15...
1	/home/kali/eceld-netsys/...	point	imgPoint	2020-09-15...
2	/home/kali/eceld-netsys/...	point	imgPoint	2020-09-15...

Adjust Subwindows

This option allows the user to readjust the windows in a tile formatting. The following is a sample of what happens when there are 4 windows open and this option is selected:



Mouse Clicks

clicks_id	content	
0	/home/kali/eceld-netsys/...	p
1	/home/kali/eceld-netsys/...	p
2	/home/kali/eceld-netsys/...	p
3	/home/kali/eceld-netsys/...	p
4	/home/kali/eceld-netsys/...	p
5	/home/kali/eceld-netsys/...	p
6	/home/kali/eceld-netsys/...	p

System Calls

auditd_id	content	
0	bash	a
1	tput setaf 1	a
2	dircolors -b	a
3	service /usr/sbin/service ...	a
4	basename /usr/sbin/service	a
5	basename /usr/sbin/service	a
6	systemctl --quiet is-active ...	a

Keypresses

keypresses_id	content	cl
0	ping google.com[Return]...	Key

Integrated Version

In the integrated version, users will first gather data using ECELd-NetSys, which will then be analyzed using the DVS system t. This version is not available at this point.

Keywords

Data Visualization System – DVS

ECELd – Evaluator-Centric and Extensible Logger

ECELd-NetSys