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.

Installation steps

Run the windows installer script.

```
> win_installer.bat
```

You will get a prompt to install wireshark. Install using default settings provided and install in C:\Program Files\Wireshark. After wireshark is installed, the script will do the following:

- Install Python's Virtual Environment Builder
- Create and activate virtual environment
- Install required dependencies

After the installation is done, you'll see the following message:

```
**********************
All dependencies where installed successfully
run 'main.py --no-sandbox' to start DVS
```

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

```
(dvs-venv)> python main.py --no-sandbox
```

Note: If you are unable to install eceld-wireshark through the command above mentioned, please go to this link: DVS-wireshark installer, download the eceld-wireshark installer and install using default configurations. Alternatively, you can download eceld-wireshark, from their GitHub page, modify the files specified in the Wireshark View of this document, compile and build using the instructions found at the Wireshark Win32/64: Step-by-Step Guide.

Linux Installation

Requirements

In order for DVS to run properly, Eceld-Wireshark must be installed.

The following are ways to install it:

- Install Eceld-Netsys: Refer to the Git-Hub page

  https://github.com/ARL-UTEP-OC/eceld-netsys.git

- Install Eceld-Wireshark: Refer to the Git-Hub page

  https://github.com/ARL-UTEP-OC/eceld-wireshark

- Install with the DVS Installer

DVS installation steps

The first step of installation is to clone the repository provided in the link above and navigate to the project folder.

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

You will be prompted if you would like to install Eceld-Wireshark:

```
kali@kali:~/DVS$ sudo ./installDeb.sh Running apt-get updateHit:1
https://packages.microsoft.com/repos/vscode stable InReleaseHit:2
http://kali.download/kali kali-rolling InReleaseReading package
lists... DoneDVS depends on : eceld-wireshark would you like to
install it [Y/n]
```

Input “Y” to install Eceld-Wireshark, if already installed input “n” to skip.

Activate Environment:

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

Run DVS:

```
(venv) kali@kali:~/DVS$ sudo python3 main.py --no-sandbox
```

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. The following describes the components delivered in each iteration:

Iteration 1:

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.

Iteration 2:

For the second iteration of the DVS system, the team focused on getting the Packet View set up. The team updated front end and backend for this iteration.

Iteration 3:

For the third iteration of the DVS system, the team focused on Synchronization features, color assignment, Suricata alerts, and more frontend features. The team also tested the system using large data and updated the installer.

Iteration 4:

For the fourth iteration of the DVS system, the team focused on Synchronization features, color assignment, Suricata alerts, and more frontend features. The team also tested the system using large data and updated the installer. This iteration was expanding on features implemented in iteration 3.

Iteration 5:

For the fifth iteration of the DVS system, the team focused on Synchronization features, color assignment for Wireshark, Suricata alerts, packet comments, and more frontend features. The team also tested the system using large data and updated the installer. This iteration was expanding on features implemented in iteration 4.

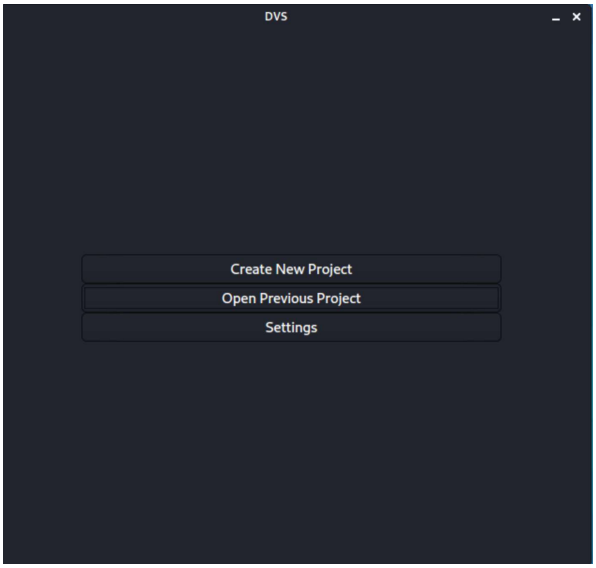
Iteration 6:

For the last iteration, the team focused on working on the feedback given by the client on Demo 5. The components included in this iteration include the margin for the sync, opening a zip project folder, editing text from the datalines, adding a tag, timestamp widget, adding and deleting a row, and the saving functionality.

The following sections will review what are the current available features.

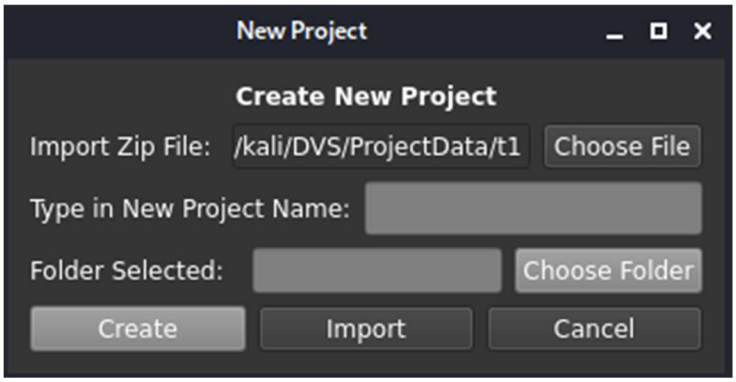
Start Window

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

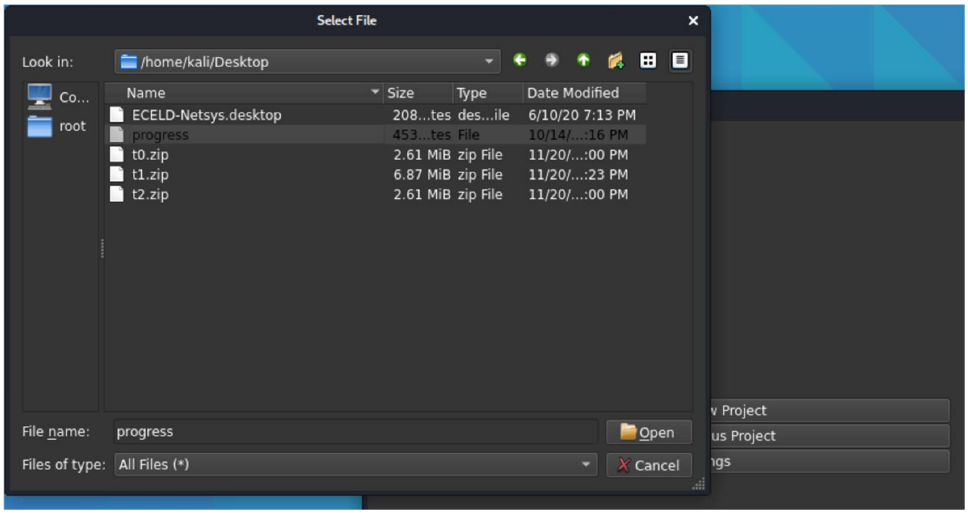


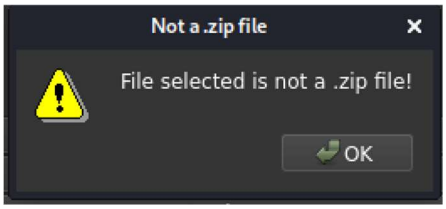
Import Project

The first option “Create New Project” has the option to import a zip file or to create a new project.



The import option has a button for the user to choose the zip file. Click on the option Choose File to select the zip folder, the path of this folder will be displayed and the create new project options will be disabled. If a zip file is not selected, the system will display a warning window. Select the Import button to unzip the folder, which will be stored in DVS/Project Data.

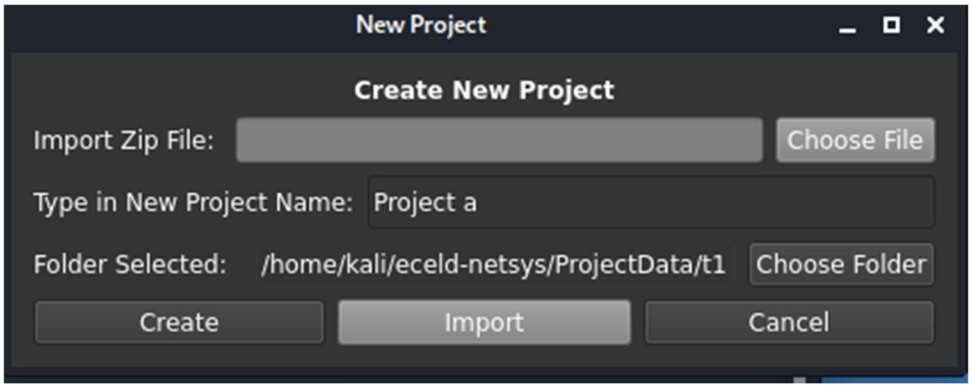




When selecting Import, two views will be displayed, the Timeline View and the Packet View.

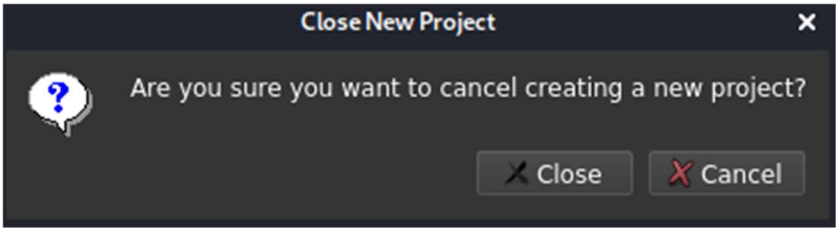
Create New Project

To create a new project from an existing folder, write the name of the new project and select the folder where the data is stored. The create option will create a copy of the folder selected, which will be stored in DVS/Project Data. The import options will be disabled.



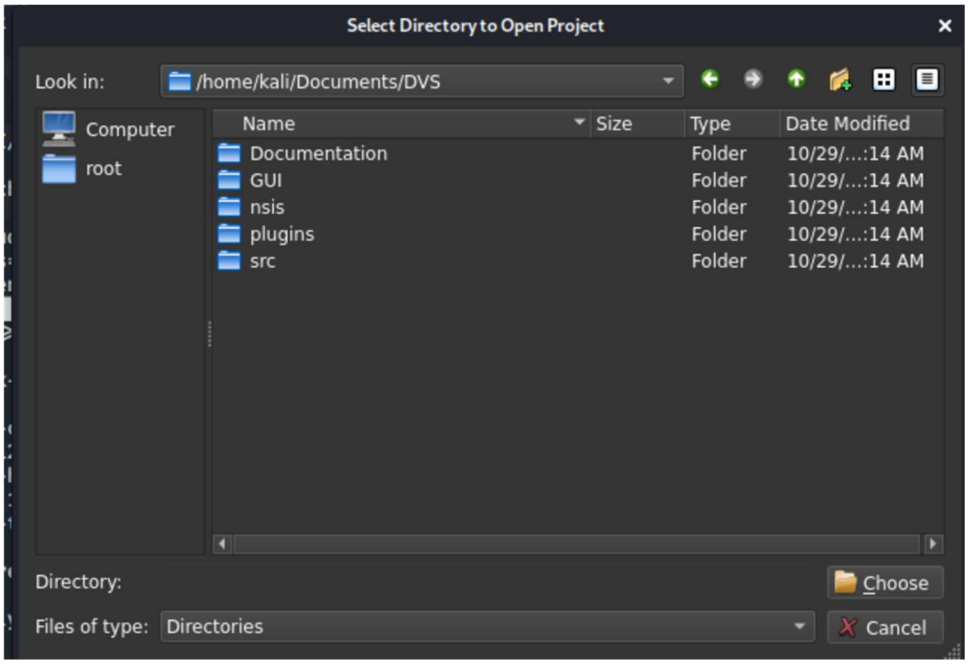
When selecting Create, two views will be displayed, the Timeline View and the Packet View.

At any time, cancel the selection by clicking on the Cancel button.



Open Previous Project

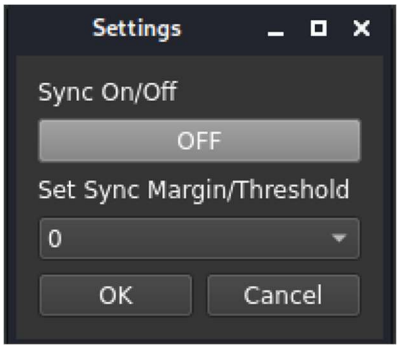
The “Open Previous Project” button shows a file browser to allow you to select a previous project. The following will appear when you select “Open Previous Project”:



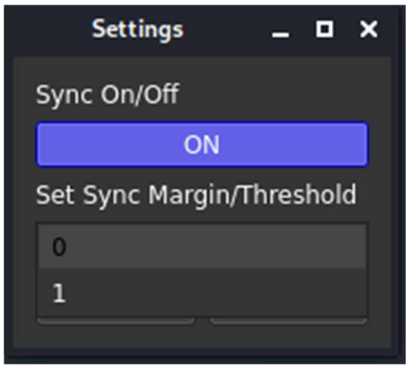
Once you select a project, the Timeline and the Packet view will appear.

Sync Settings

The settings button will display the following:



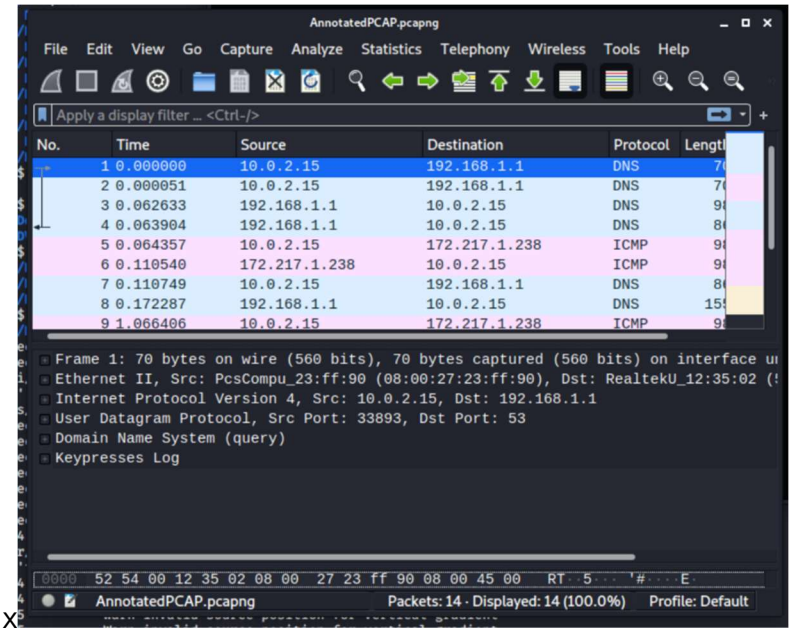
This is for sync configuration, select the margin as shown in the next figure. The default margin is 0 seconds but can be changed to 1 second.



Select OFF button to turn the sync ON, with this functionality the Wireshark Sync and Timestamp Sync will be ON when the project is opened.

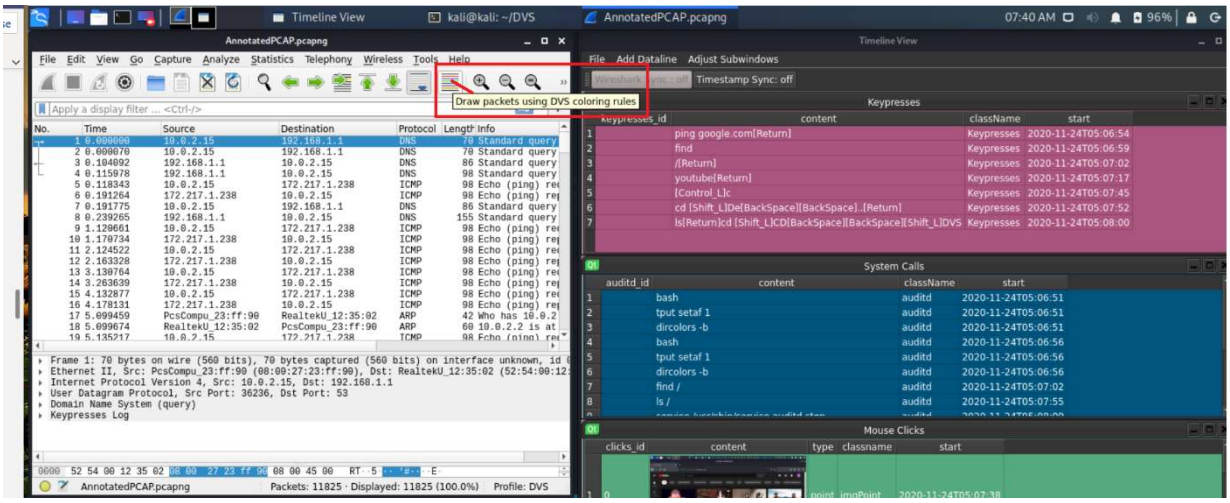
Packet View

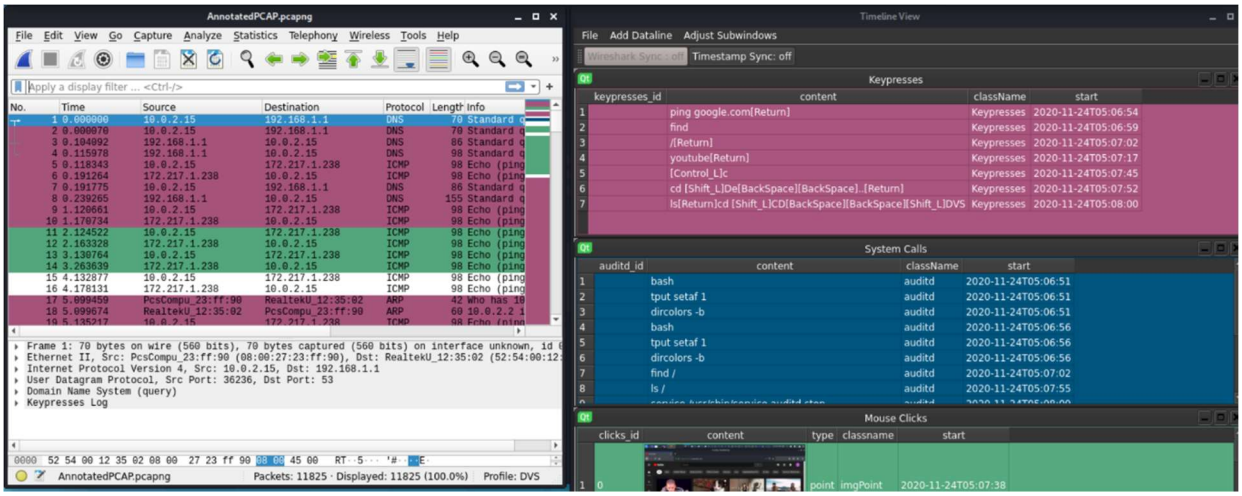
The packet view consists of the ECELd version of Wireshark that is available at https://github.com/ARL-UTEP-OC/eceld-wireshark.



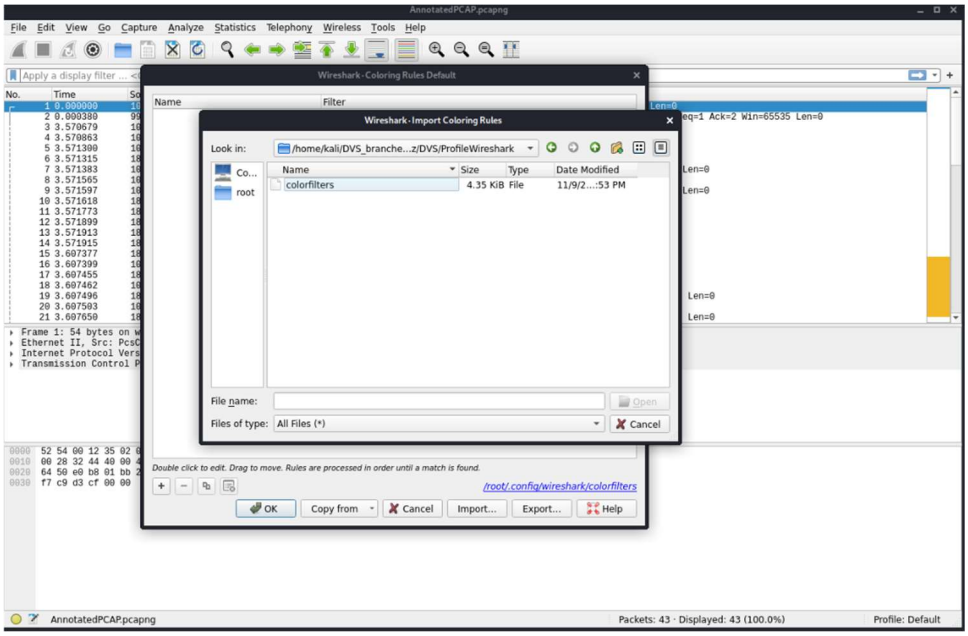
Wireshark Coloring Rules

The system will generate a color rule file that can be read by Wireshark. In order to colorize the packets in the Wireshark View to correspond to active data lines on the Timeline View, the user can click on the  toolbar button and refresh the page by pressing “CTRL+R”. Enabling/Disabling this button colors and de-colors packets accordingly.

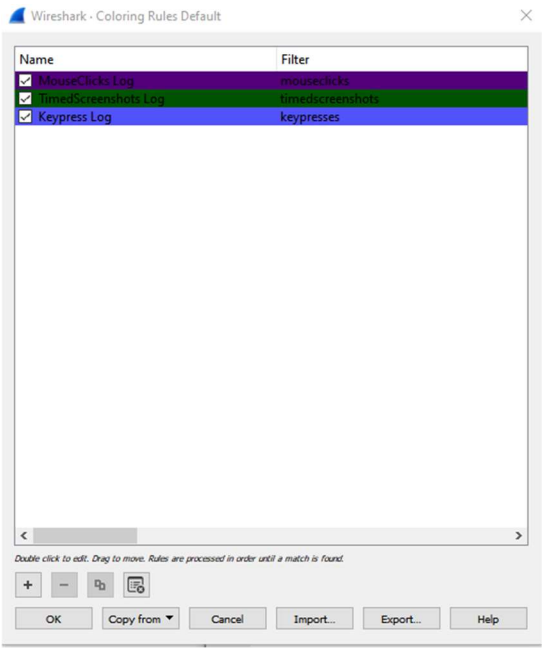




Alternatively, the user can import this file manually into Wireshark. This file is located in the profile directory within Wireshark (.. /eceld-wireshark/wireshark-3.2.0/profiles/DVS/colorfilters) **OR** in the DVS directory (../DVS/GUI/PacketView/colorfilters)



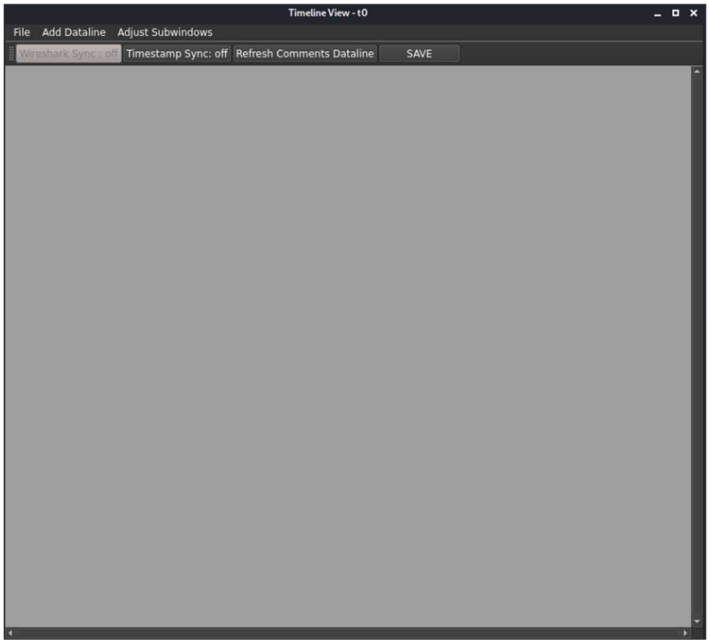
If there are any current rules on the dialog, clear them using this button  Then the desired colorfilter file can be selected and applied:



Currently, the DVS on a Windows OS supports only the manual import of the color profile.

Timeline View

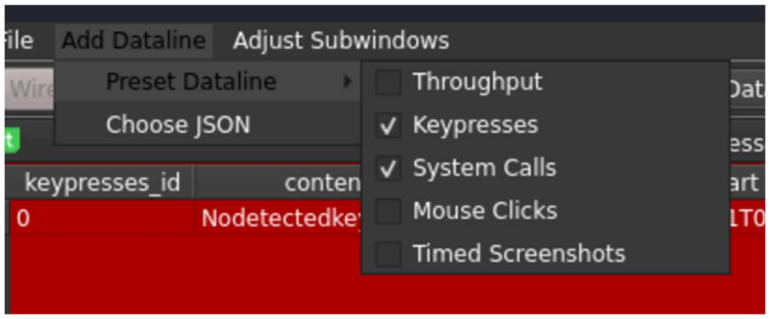
The Timeline View looks like the following:



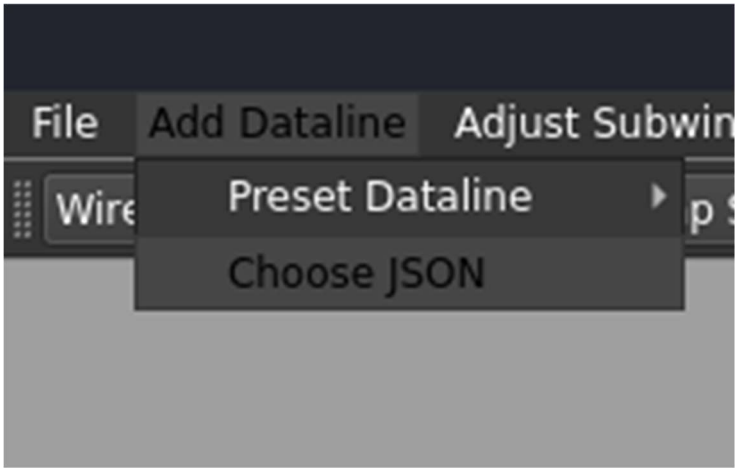
Adding Datalines

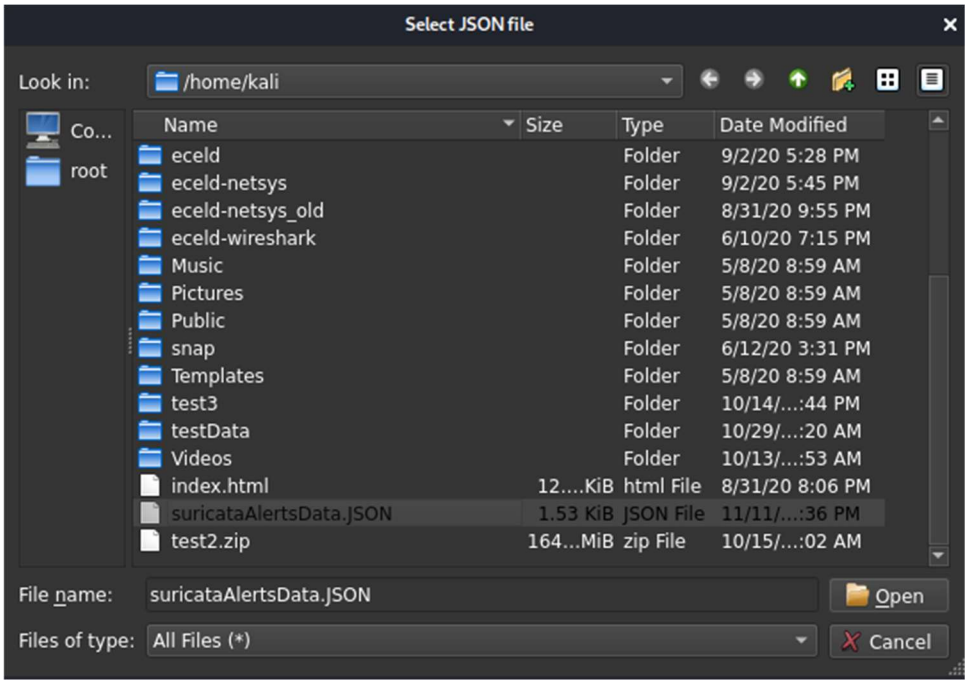
The “Add Dataline” menu option has two options: Preset Dataline and Choose JSON .

When selecting the Preset Dataline option, a list of the predefined datalines will become visible. These datalines will use the data from the project created in the start window. When a dataline is open, the checkbox will be checked. When the dataline is closed, the checkbox is unchecked. This feature allows for one view of each dataline to be available at once.

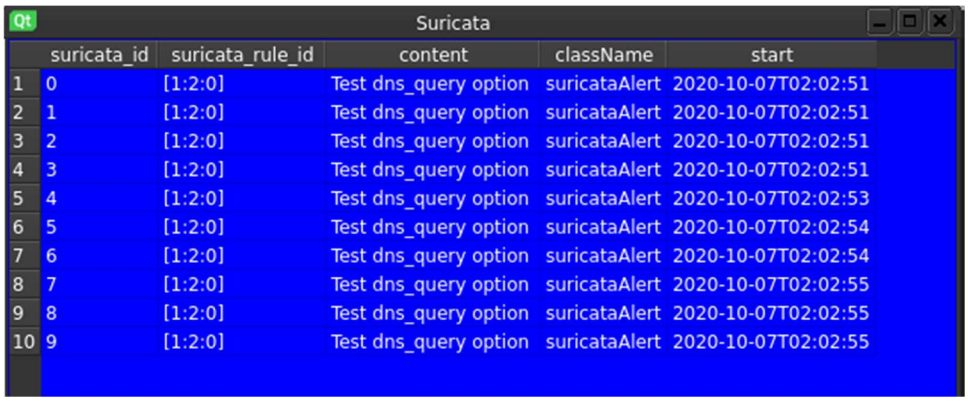


When selecting the Choose JSON option, select a JSON file to open it on the timeline view. According to the JSON data, the correct render will be used to display the data.





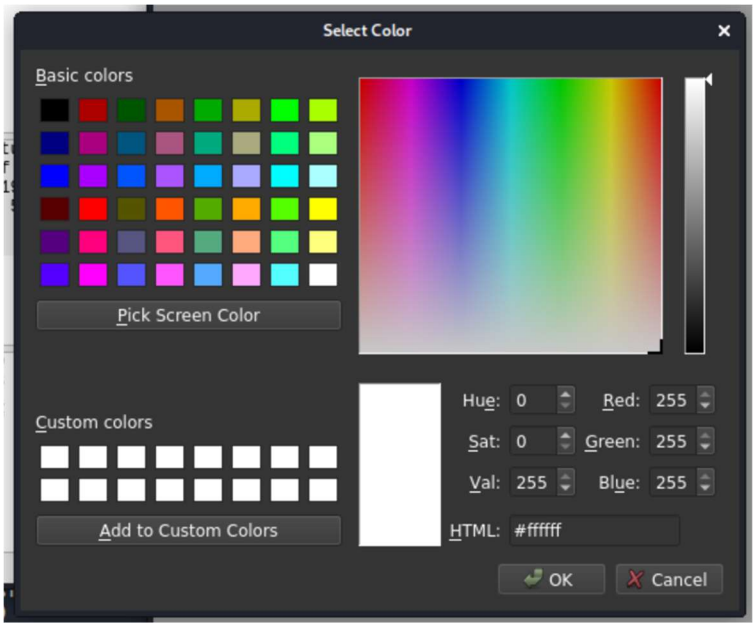
As an example, if selecting a new JSON file like the Suricata Alerts Data, it will be rendered as a text table display.



	suricata_id	suricata_rule_id	content	className	start
1	0	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:51
2	1	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:51
3	2	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:51
4	3	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:51
5	4	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:53
6	5	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:54
7	6	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:54
8	7	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:55
9	8	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:55
10	9	[1:2:0]	Test dns_query option	suricataAlert	2020-10-07T02:02:55

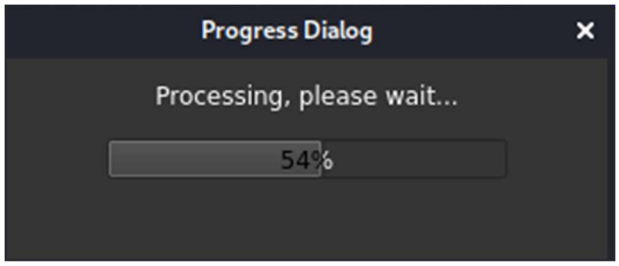
Color Selection

Once each dataline is selected, the user will be asked to select a color to assign to the dataline. The color selector looks like the following:

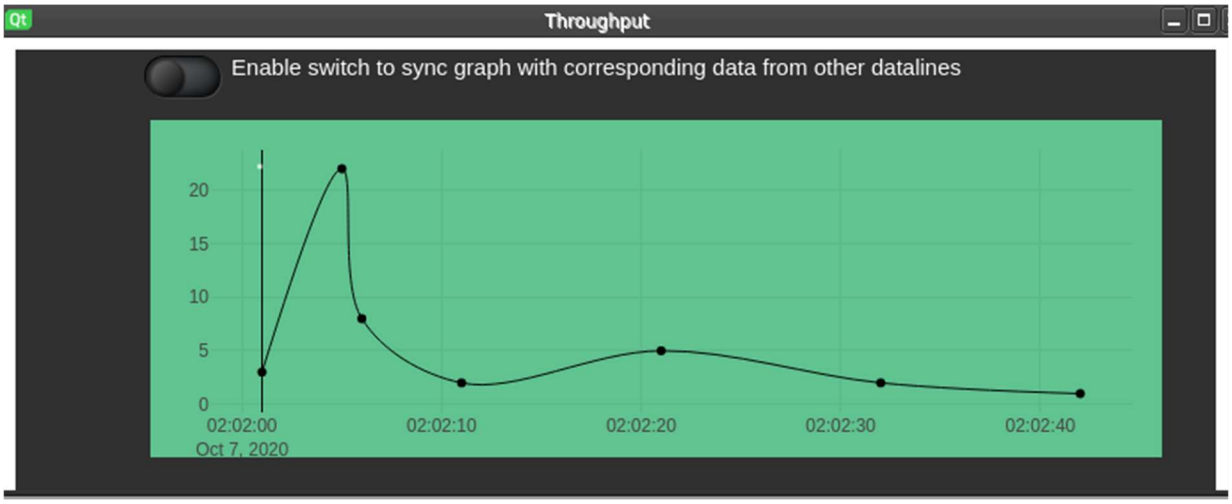


Throughput

The first dataline option is throughput. When selecting this option, a progress dialog will be visible indicating that the dataline is loading.

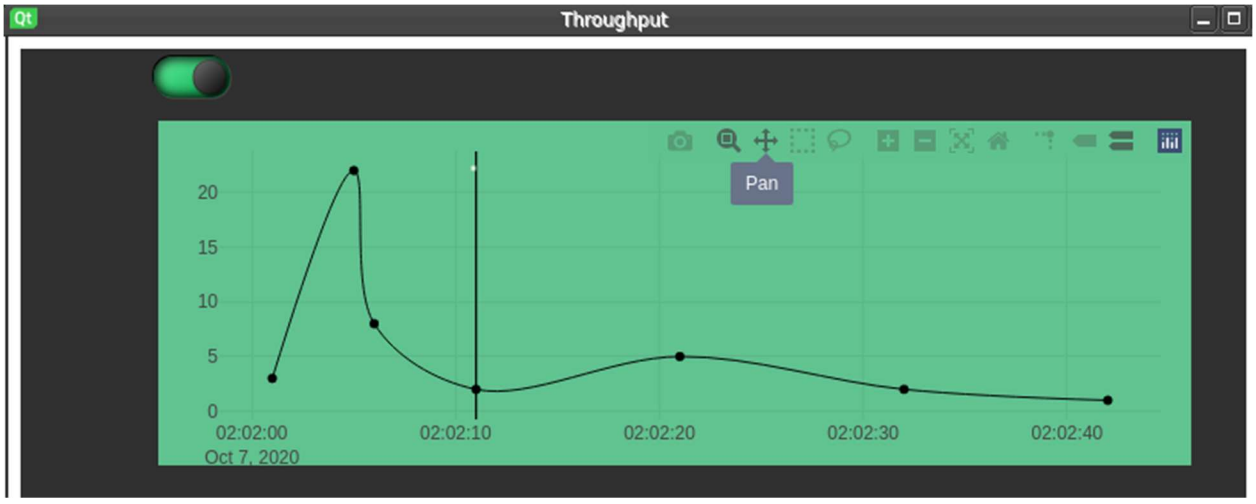


The following dataline will be shown in the timeline view.



The Throuput has the following features:

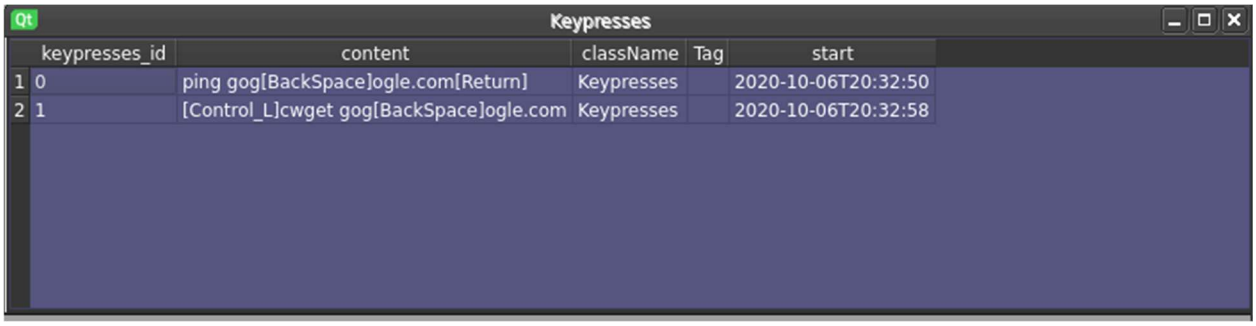
- Pan
- Zoom in and out
- Sync toggle button



This dataline is a dash graph render displaying the tshark network throughput data in time intervals.

Keypresses

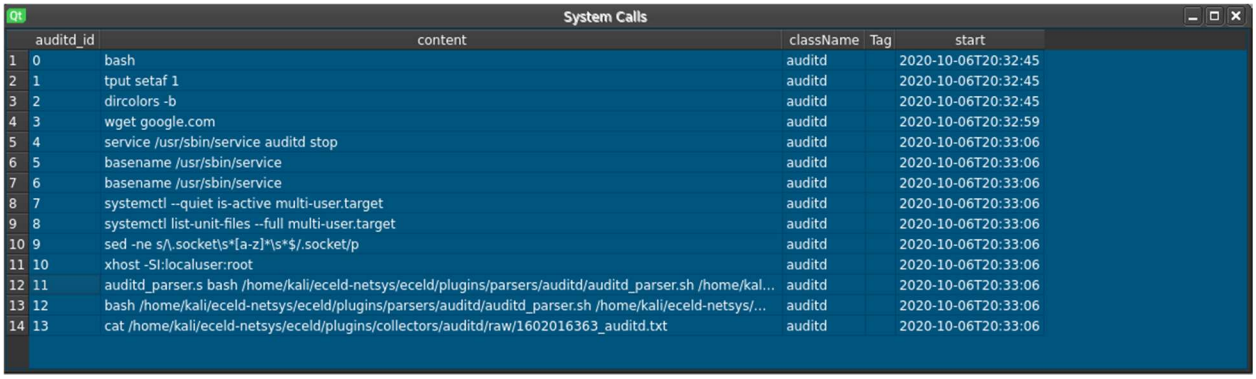
The second dataline option is keypresses. When selecting this option, the following dataline will be shown in the timeline view:



This dataline is a text render in a table display containing the parsed data from the keypresses JSON file. This table includes the id of the keypress (keypresses_id), the content, className, tag, and the timestamp.

System Calls

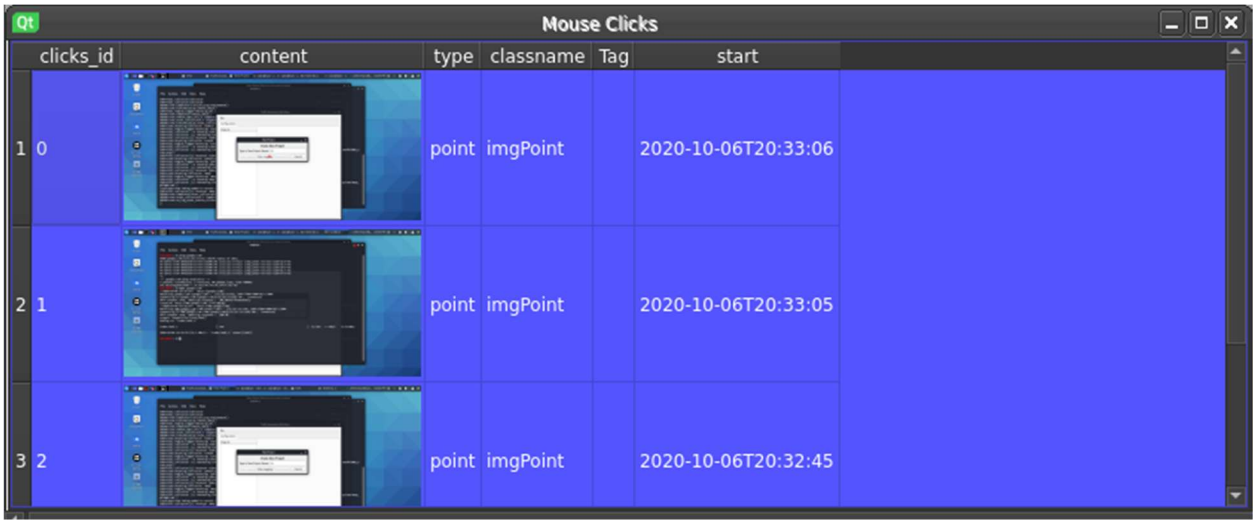
The third dataline option is the system calls. When selecting this option, the following dataline will be shown in the timeline view:



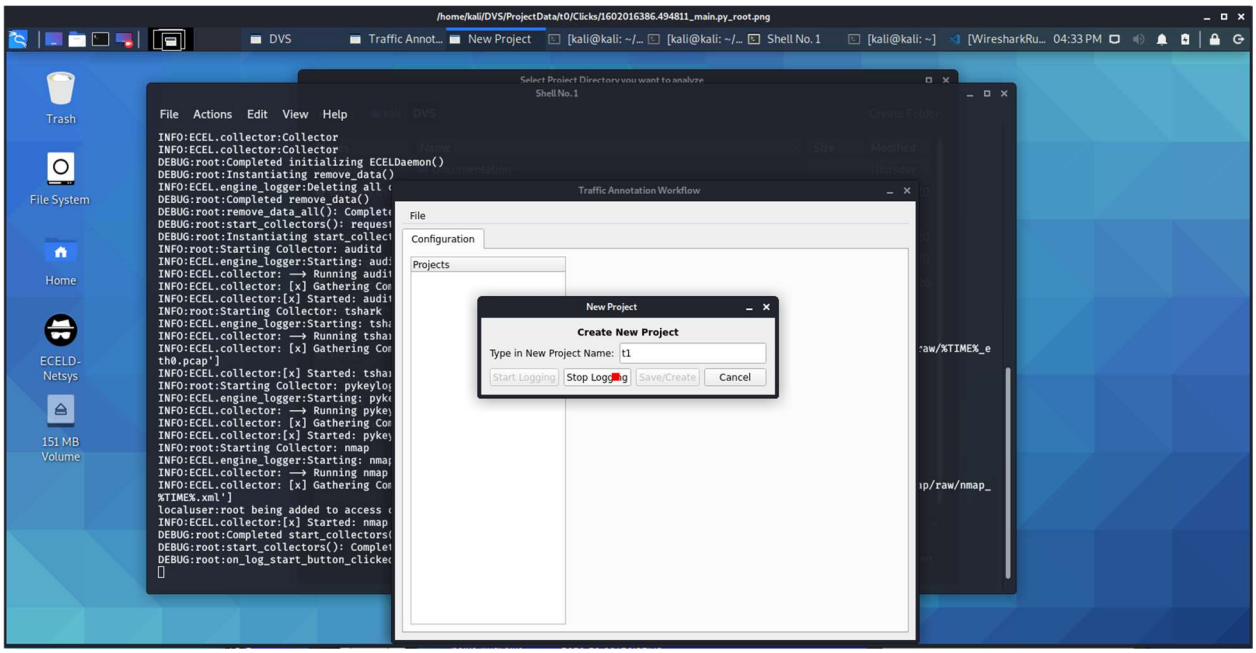
This dataline is a text render in a table display containing the parsed data from the auditdData JSON file. This table includes the auditd id, the system call, className, tag, and timestamp in which the system call was used.

Mouse Clicks

The fourth dataline option is the mouseclicks. When selecting this option, the following dataline will be shown in the main window:

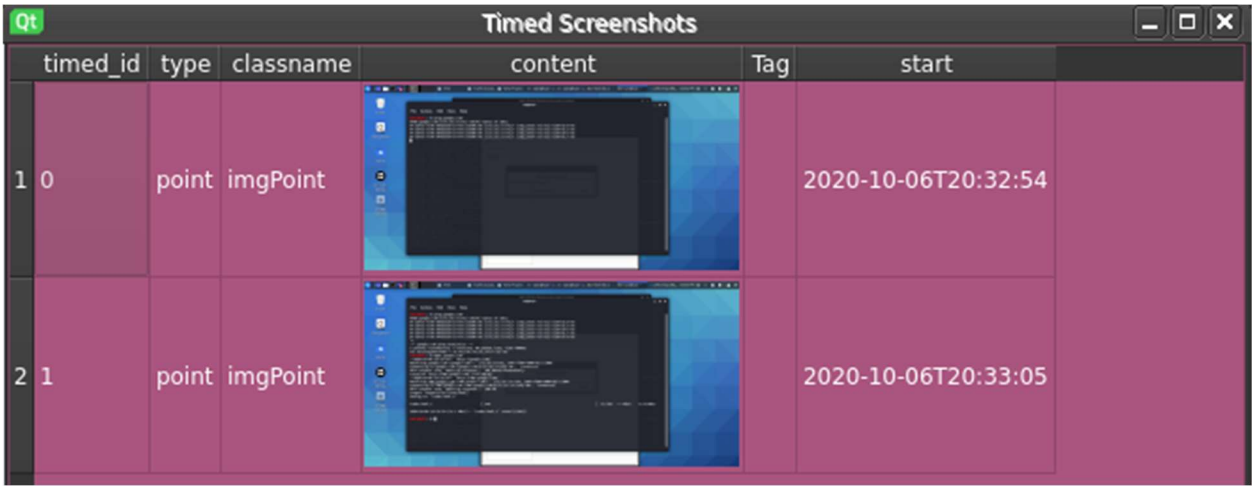


This table includes the click id, the screenshot, the type, className, tag, and the timestamp of the click. Click on any of the images to view it in a new window. The new window contains as a title the path of the screenshot, it is possible to open multiple screenshots at a time.



Timed Screenshots

The last option is the Timed screenshots. When selecting this option, the following dataline will be shown in the main window:



This table includes all the screenshots that were timed, the timed_id, the type, className, tag, and the timestamp of the screenshot. Click on any of the images to view it in a new window.

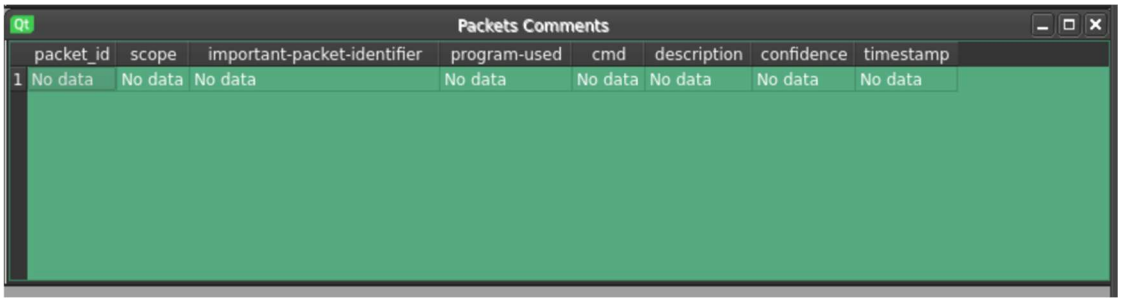
Packets Comments

A packets comments JSON file is created when selecting the import or create a new project option available in the start window of the DVS system.

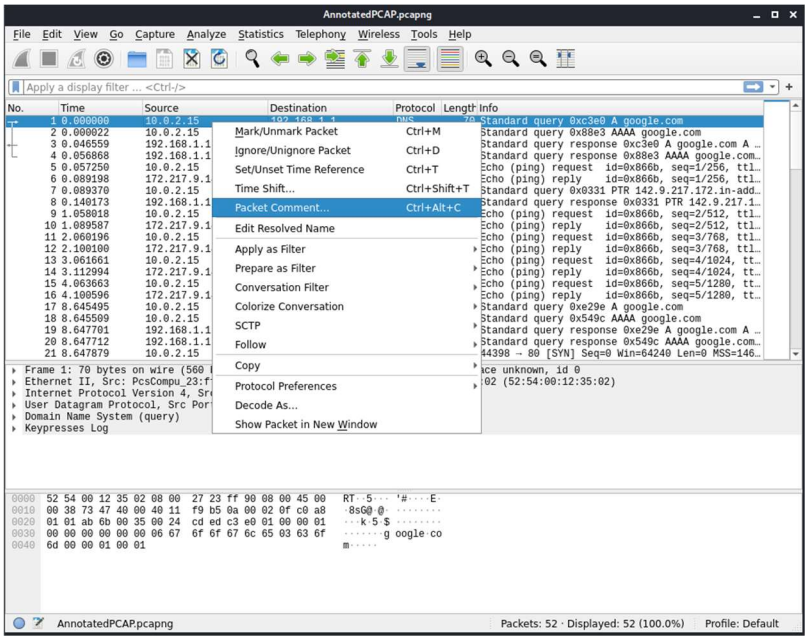
To view the Packets Comments dataline on the Timeline View, select the chooseJSON option to open the file located in DVS/ProjectData/NameofProject/ ParsedLogs /pcomments.json

This file is associated with the PCAP contained in the folder DVS/ProjectData/PCAP

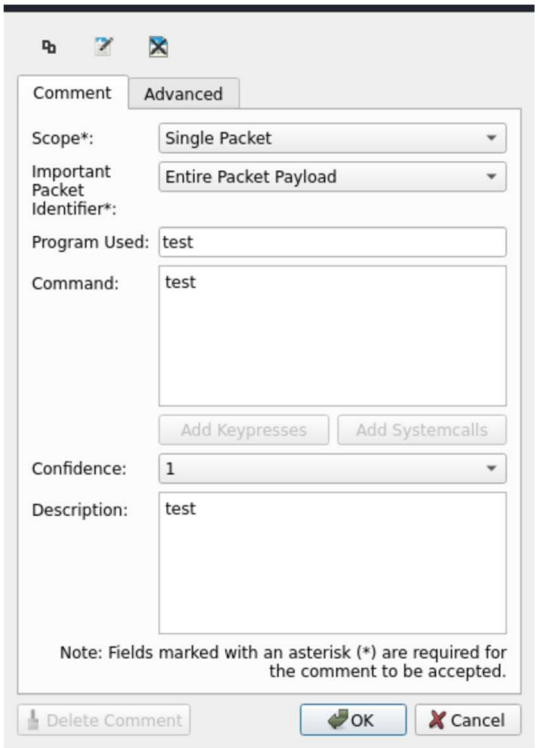
If the PCAP contains no packets comments, the following will be displayed:



Add packet comments by going to the packet view, then selecting a packet, right click > Packet Comment ...

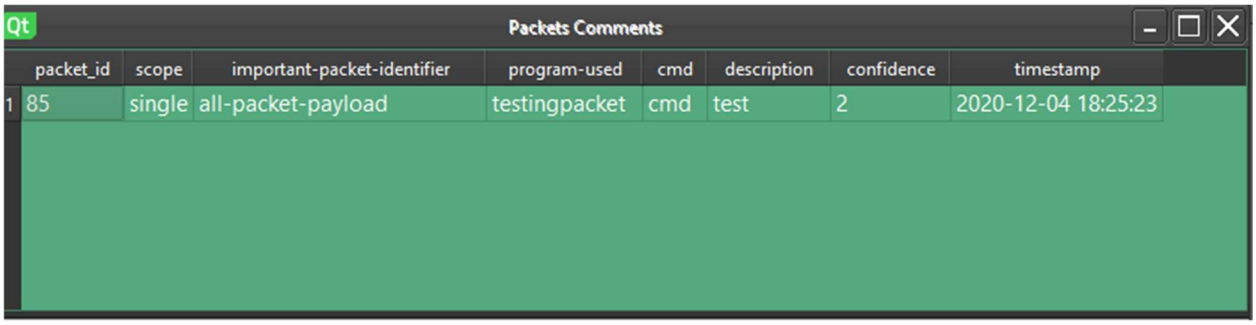


The following dialog will be displayed:

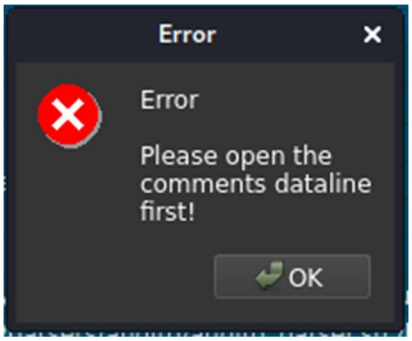


Once the comment is added, save the PCAP with CTRL + S , then click on the Refresh Comments Dataline option, this will trigger a tshark command.

The pcomments.json file will be updated and the new information will be displayed in the dataline.



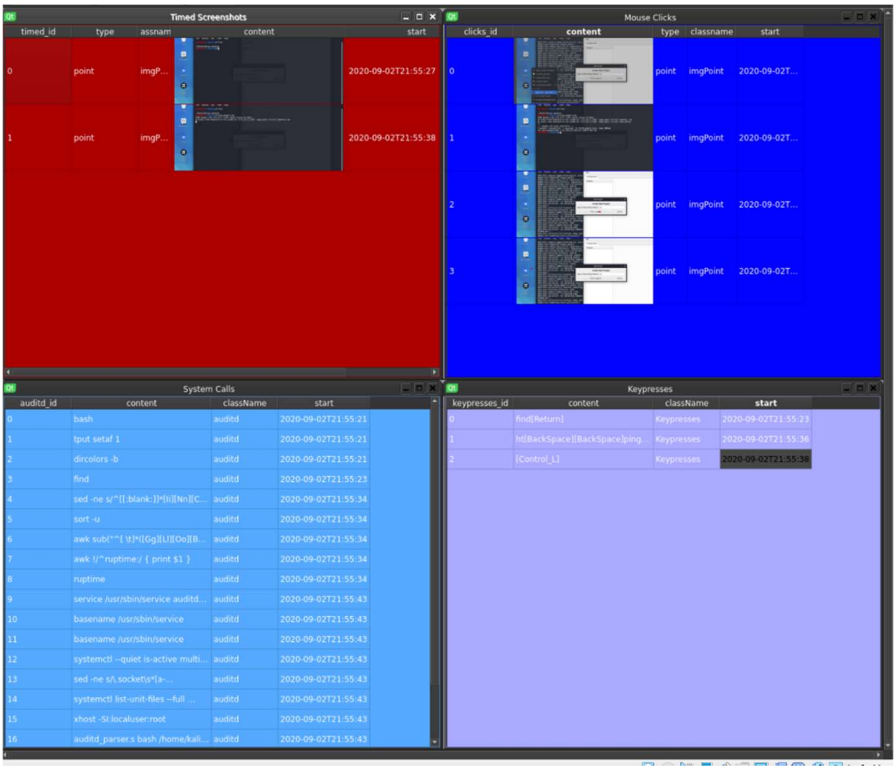
If the Packets Comments dataline is not opened and the Refresh Comments Dataline option was selected, the following warning will appear:



Adjust Subwindows

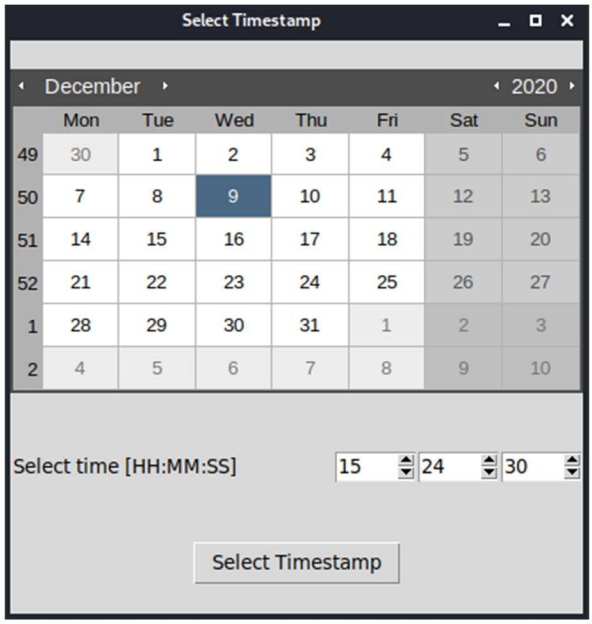
The system allows to open, close and minimize each dataline window individually as well as displaying a combination of the datalines available at the same time. The option Adjust

subwindows readjusts the windows in a tile formatting. The following is an example of what happens when there are 4 windows open and this option is selected:



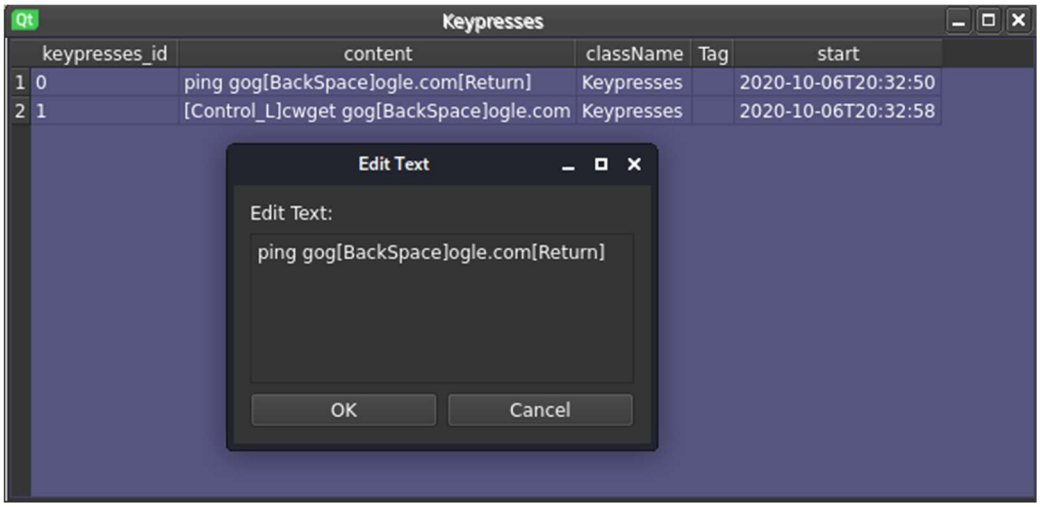
Timestamp Widget

To edit the timestamp of any row in the dataline, double click the timestamp cell in the table to open the Timestamp widget. Select the new time and click on Select Timestamp, the new timestamp will be visible in the column start or timestamp.



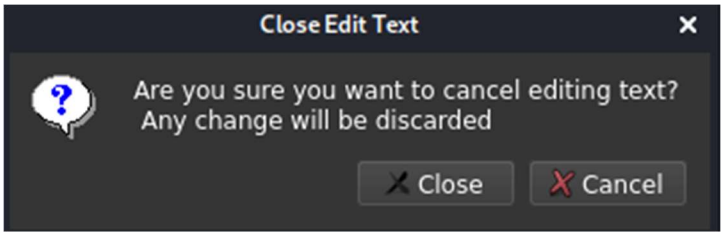
Edit Text

The datalines have the feature to edit the text on the content column, double click on a row to open the Edit Text widget. In the following example, the first row was double clicked. The widget will display the current information in the content column.



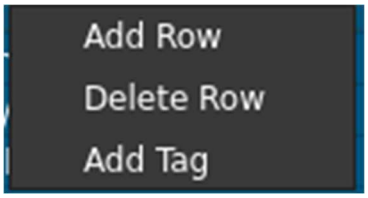
Write the new string and select OK to view the changes on the table.

To cancel the edit, select Cancel.



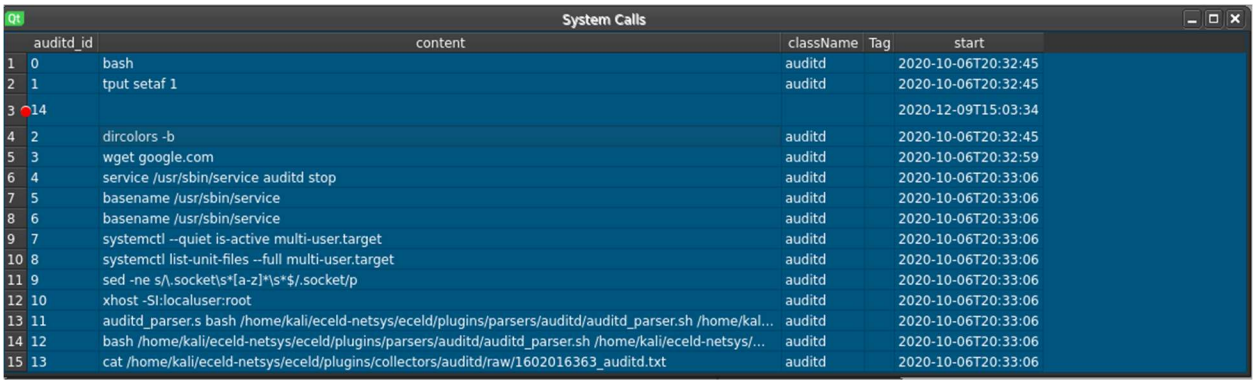
Context Menu

Right click on any row of the dataline to make visible the context menu. This menu has three options: add row, delete row, and add/edit tag.



The “Add row” option adds a row to the table selected, if selecting this option, the timestamp widget will appear to allow for the selection of the new timestamp for the new row.

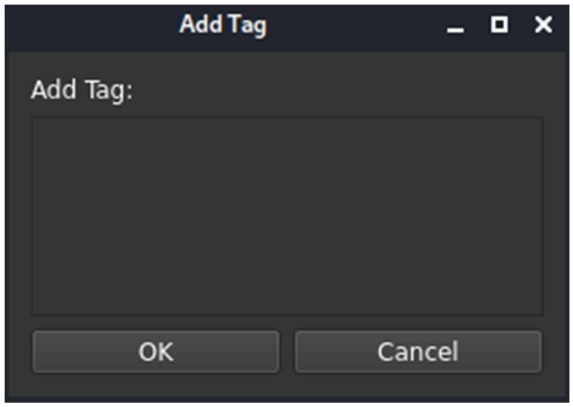
The new row will be visible where the context menu was triggered, the system will generate the id based on the last id of the JSON file selected. In the next figure, it shows that row 14 is the new row added.

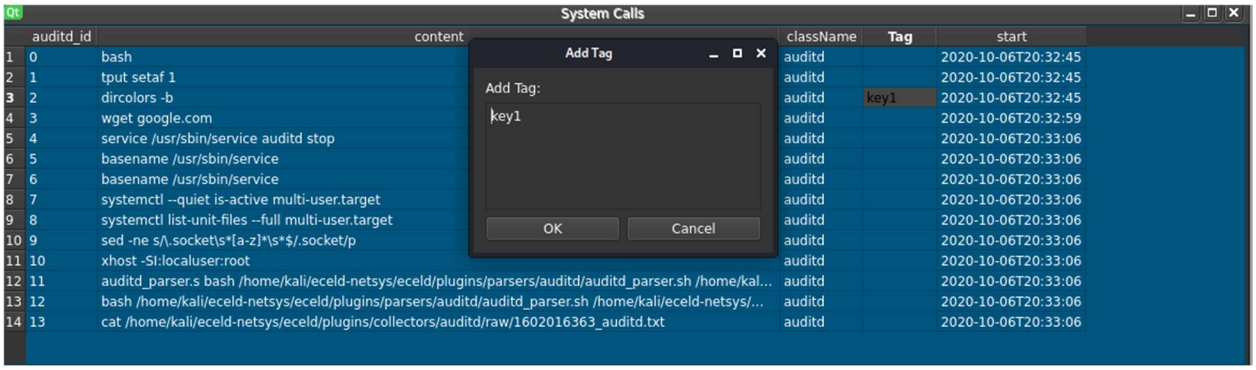


	auditd_id	content	className	Tag	start
1	0	bash	auditd		2020-10-06T20:32:45
2	1	tput setaf 1	auditd		2020-10-06T20:32:45
3	14				2020-12-09T15:03:34
4	2	dircolors -b	auditd		2020-10-06T20:32:45
5	3	wget google.com	auditd		2020-10-06T20:32:59
6	4	service /usr/sbin/service auditd stop	auditd		2020-10-06T20:33:06
7	5	basename /usr/sbin/service	auditd		2020-10-06T20:33:06
8	6	basename /usr/sbin/service	auditd		2020-10-06T20:33:06
9	7	systemctl --quiet is-active multi-user.target	auditd		2020-10-06T20:33:06
10	8	systemctl list-unit-files --full multi-user.target	auditd		2020-10-06T20:33:06
11	9	sed -ne s/\.socket\s*[a-z]*\s*$/.socket/p	auditd		2020-10-06T20:33:06
12	10	xhost -SI:localuser:root	auditd		2020-10-06T20:33:06
13	11	auditd_parser.s bash /home/kali/eceld-netsys/eceld/plugins/parsers/auditd/auditd_parser.sh /home/kal...	auditd		2020-10-06T20:33:06
14	12	bash /home/kali/eceld-netsys/eceld/plugins/parsers/auditd/auditd_parser.sh /home/kali/eceld-netsys/...	auditd		2020-10-06T20:33:06
15	13	cat /home/kali/eceld-netsys/eceld/plugins/collectors/auditd/raw/1602016363_auditd.txt	auditd		2020-10-06T20:33:06

The “Delete row” option deletes the row selected.

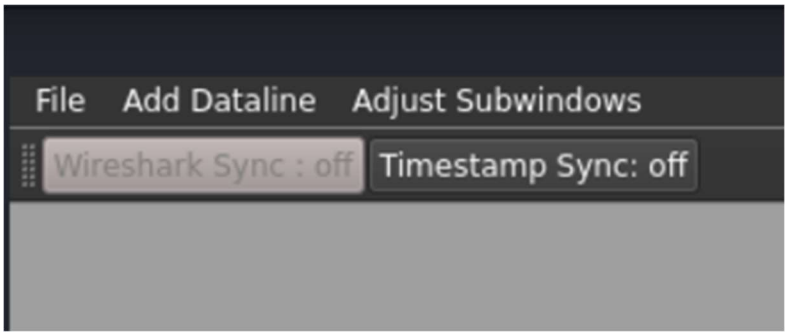
The “Add/Edit Tag” option will open the Add Tag widget. If the row selected does not have tags, the tag widget will be empty. Add text to add a tag and select the OK button to see it on the Tag column of the dataline.





Buttons to enable/disable synchronization

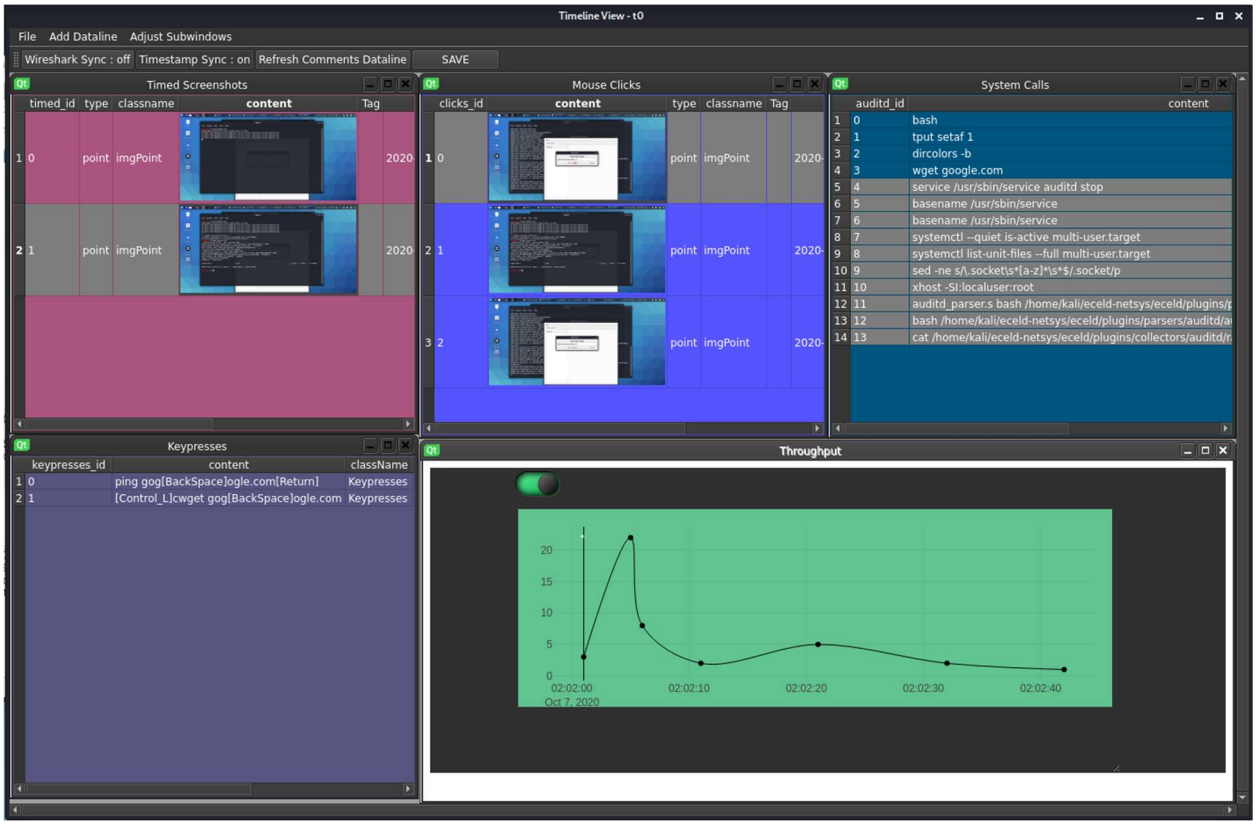
This feature will allow the user to synchronize views. The following are the sync buttons:



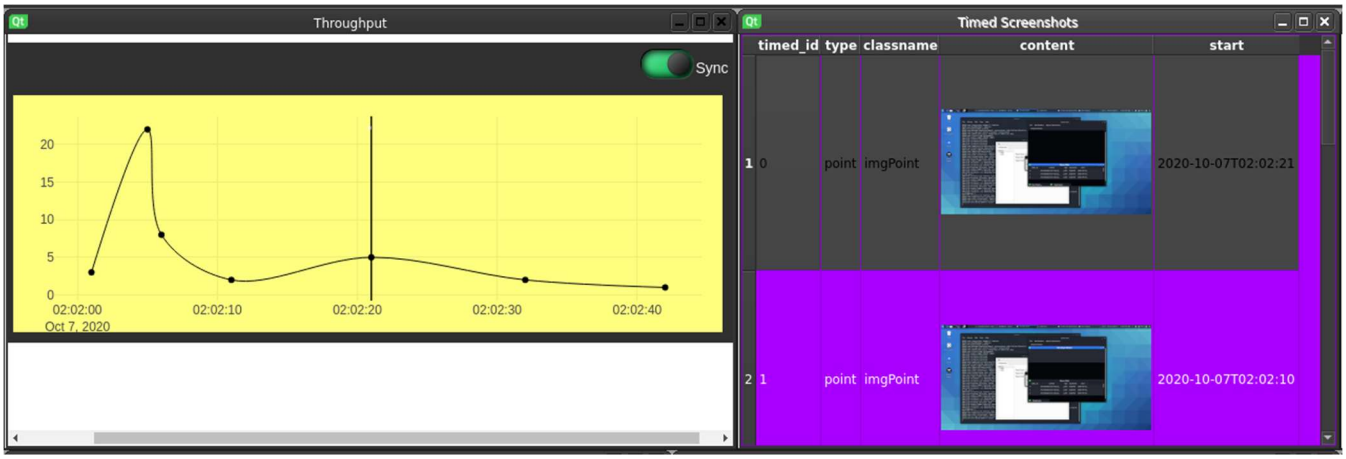
Timestamp Sync

When selecting “Timestamp Sync: off” it will update to “Timestamp Sync: on.” All the datalines in the timeline view will be synchronized according to the timestamp, highlighting the rows in the tables were the system finds a matching timestamp.

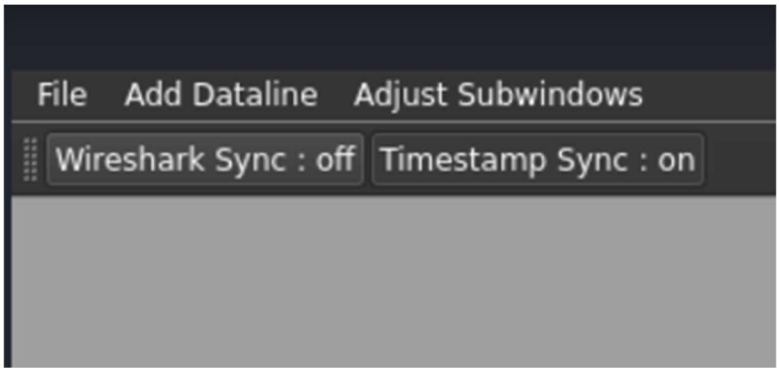
The user can select any row of any dataline to sync the datalines according to the timestamp selected. The following shows the sync behavior:



The Throughput has a sync toggle button, that when activated will send a signal to the system, this will sync the throughput when the user selects a timestamp from the other datalines.



Initially, the Wireshark button is disabled and can only become clickable when Timestamp sync has been turned on:

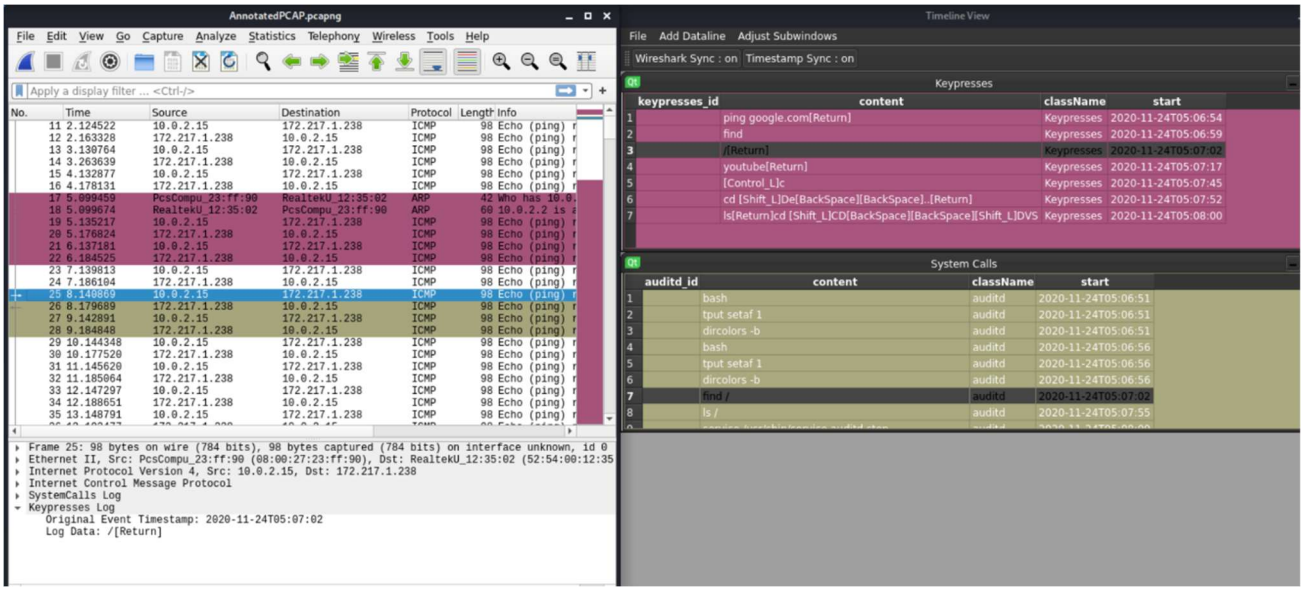


It will become disabled whenever Timestamp sync is turned off again.

To enable the sync buttons from the beginning, go to the settings section for more information.

Wireshark Sync

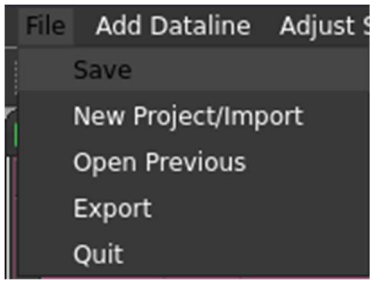
As mentioned previously, Wireshark Sync can only be turned on if Timestamp sync is on. When the user clicks on “Wireshark Sync: off” it will be updated to “Wireshark Sync: on.” Now, clicking a packet on the Wireshark view will highlight the corresponding data in the Timeline View and vice versa:



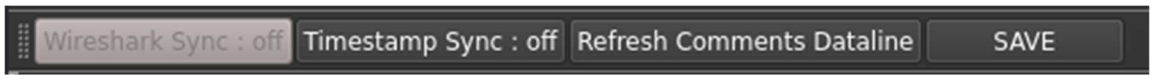
Save

The DVS system has three options to save the changes:

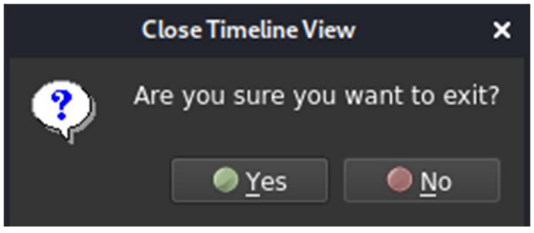
- File Drop-Down Menu > Save



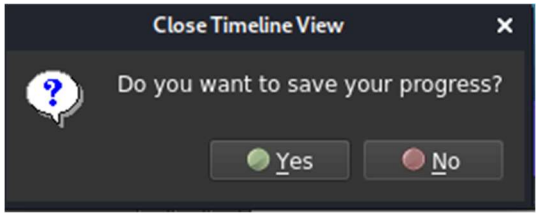
- Main menu > Save



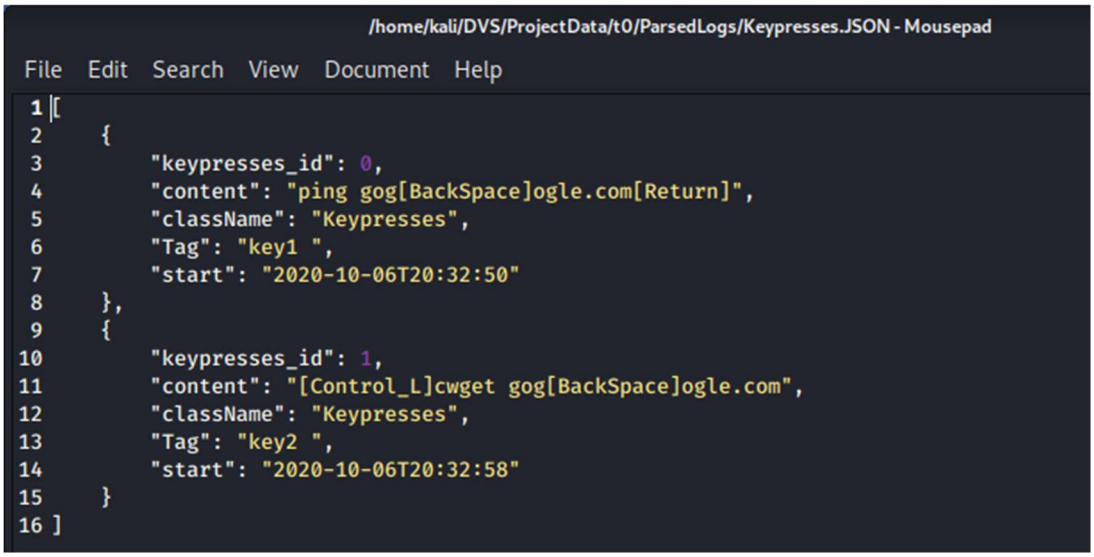
- Closing the Timeline View



Selecting Yes, to close the Timeline View.



The Save functionality will save/update the files located in the folder DVS/ProjectData/NameofProject/ParsedLogs

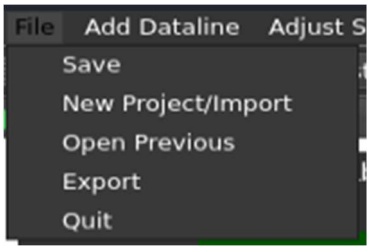


File Drop-Down Menu

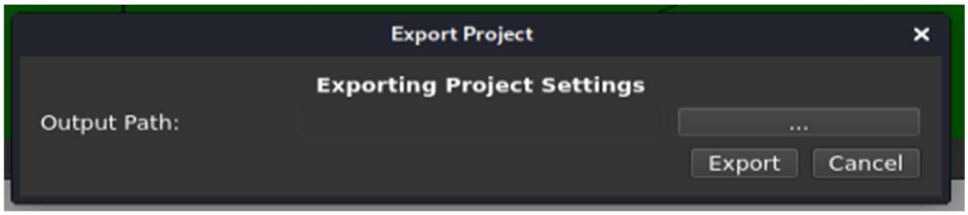
The DVS system now has more options for the file drop-down menu, however for iteration 5 only 2 functionalities work: Import, Export. If the user clicks on any other option the system will simply ignore the request. These include: Save, Open Previous.

File Drop-Down Menu (Export)

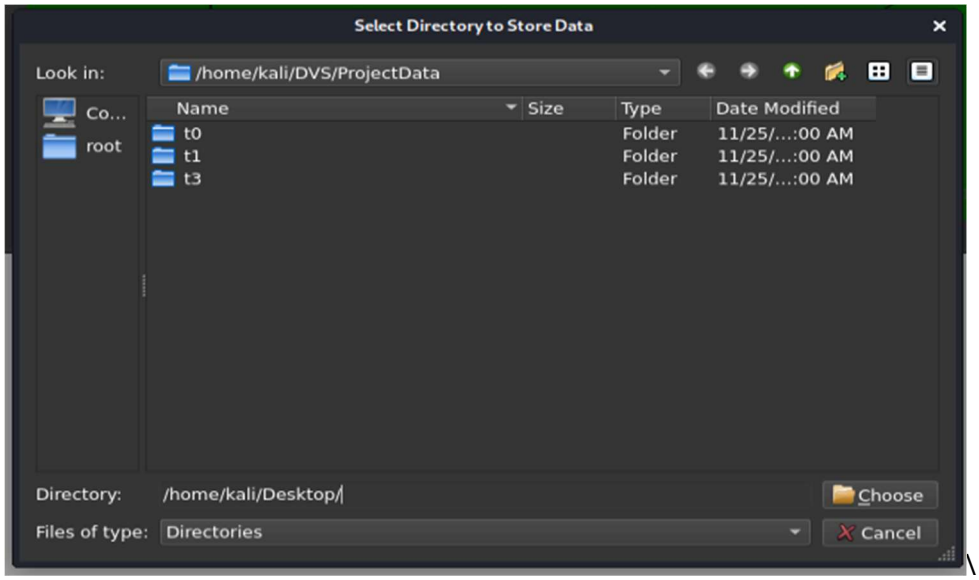
The DVS system is now able to export a working project to a directory that you choose, you will need to navigate to the tool bar on top and select the “File” tab.



A drop down will show with different types of functionalities, to export click the “Export” selection.

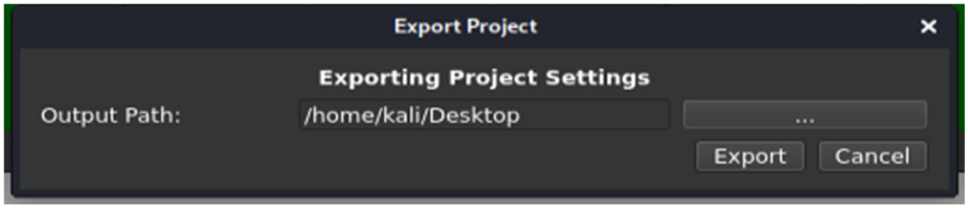


The DVS system will then display a pop-up window “Export Project” that will prompt the user to provide information about the location to export to. For this iteration, the user would be able to click on the image below to choose the directory.

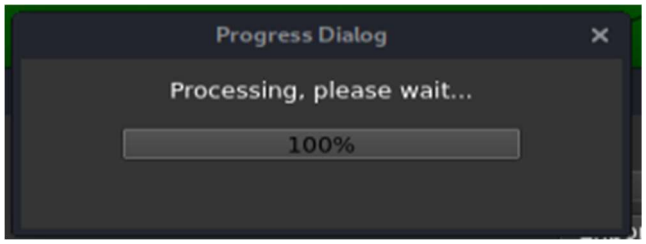


Once you manually enter the directory or navigate to the directory, click on the “choose” button showed in the image below.

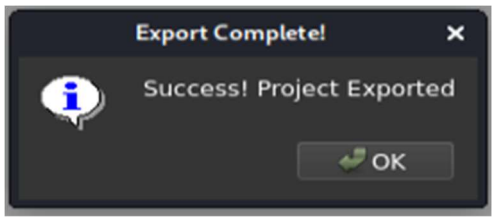
This will then populate the Output Path to the directories path you selected, now click on the export button.



While the DVS is exporting the project, it will provide the user with a progress dialog.



When finished, the DVS system will display a pop-up window notifying the user that the project has been exported.



Wireshark View

The DVS system uses a modified version of eceld-Wireshark in order to properly apply the sync and colorization functionalities. These are the files that were modified in the following directories:

- **/eceld-wireshark/wireshark-3.2.0/ui/qt/:**
    - packet_list.cpp
    - coloring_rules_dialog.cpp
    - coloring_rules_dialog.h
    - main_window_slots.cpp
    - main_window.ui
- **/eceld-wireshark/wireshark-3.2.0/:**
    - CMakeOptions.txt
        - Include build option “Build qtshark” ON

In order to display Suricata Log, eceld-wireshark had to be modified as well. The following files were modified

- **/eceld-wireshark/wireshark-3.2.0/ui/qt/:**
    - proto_tree.cpp
    - proto_tree.h

All of the modified files can be found in the folder: Wireshark_mod_files located in the DVS GitHub repository.

Appendix

Keywords

DVS - Data Visualization System

ECELd – Evaluator-Centric and Extensible Logger

ECELd-netsys