National College of Ireland

Configuration Manual

MSc Research Project
Cybersecurity

Keith Cooney
Student ID: 18201270

School of Computing
National College of Ireland

Supervisor: Dr Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Keith Cooney
Student ID:	18201270
Programme:	MSc in Cybersecurity **Year:** 2021
Module:	Research Project
Lecturer:	Dr Imran Khan
Submission Due Date:	20th September 2021
Project Title:	Operational Technology Intrusion Detection Application for Power Grid Security Operation Centre
Word Count:	**2244** **Page Count: 26**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Keith Cooney

Date: 20th September 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

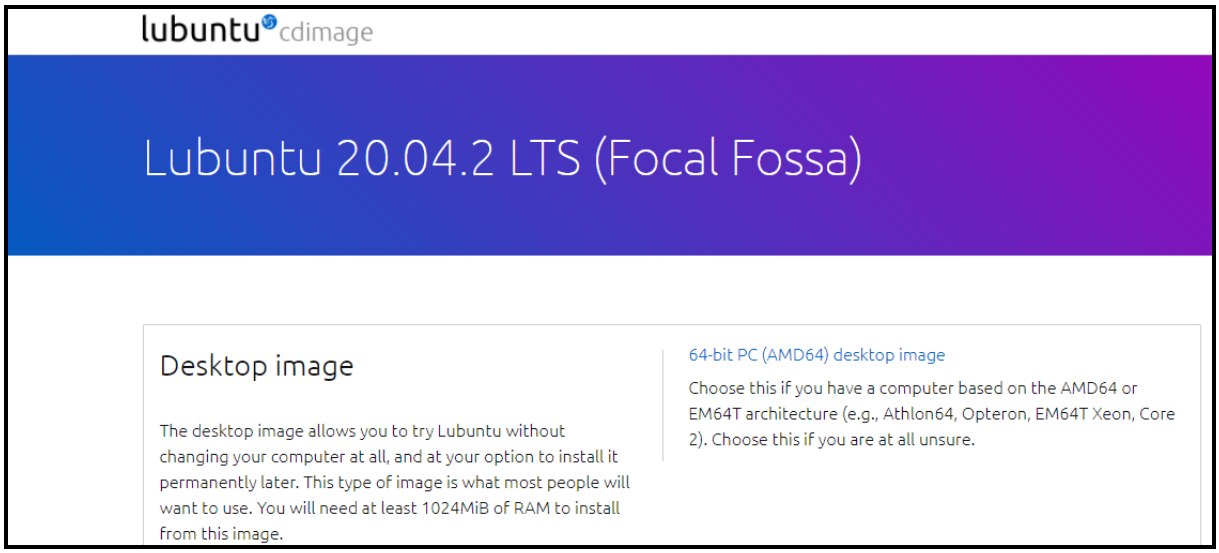
Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

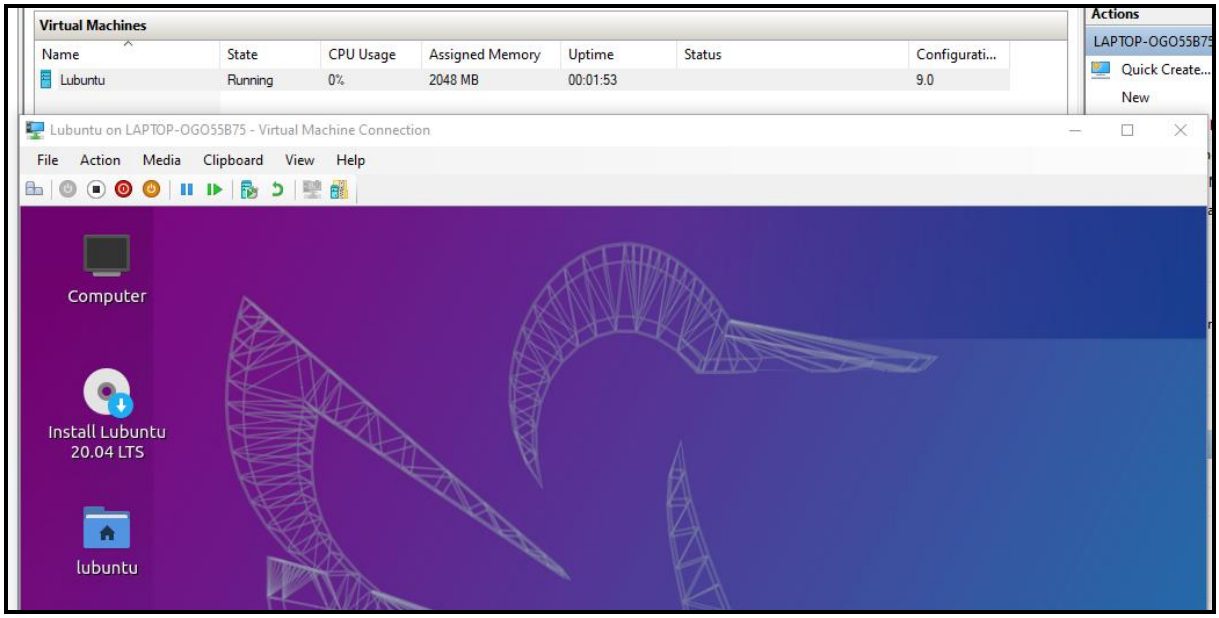
Keith Cooney
Student ID: 18201270

1 Setting up Virtual Machine

Download the Lubuntu ISO for AMD processors. Lubuntu was chosen as it is a lightweight operating system. Refer to (Lubuntu, 2021)



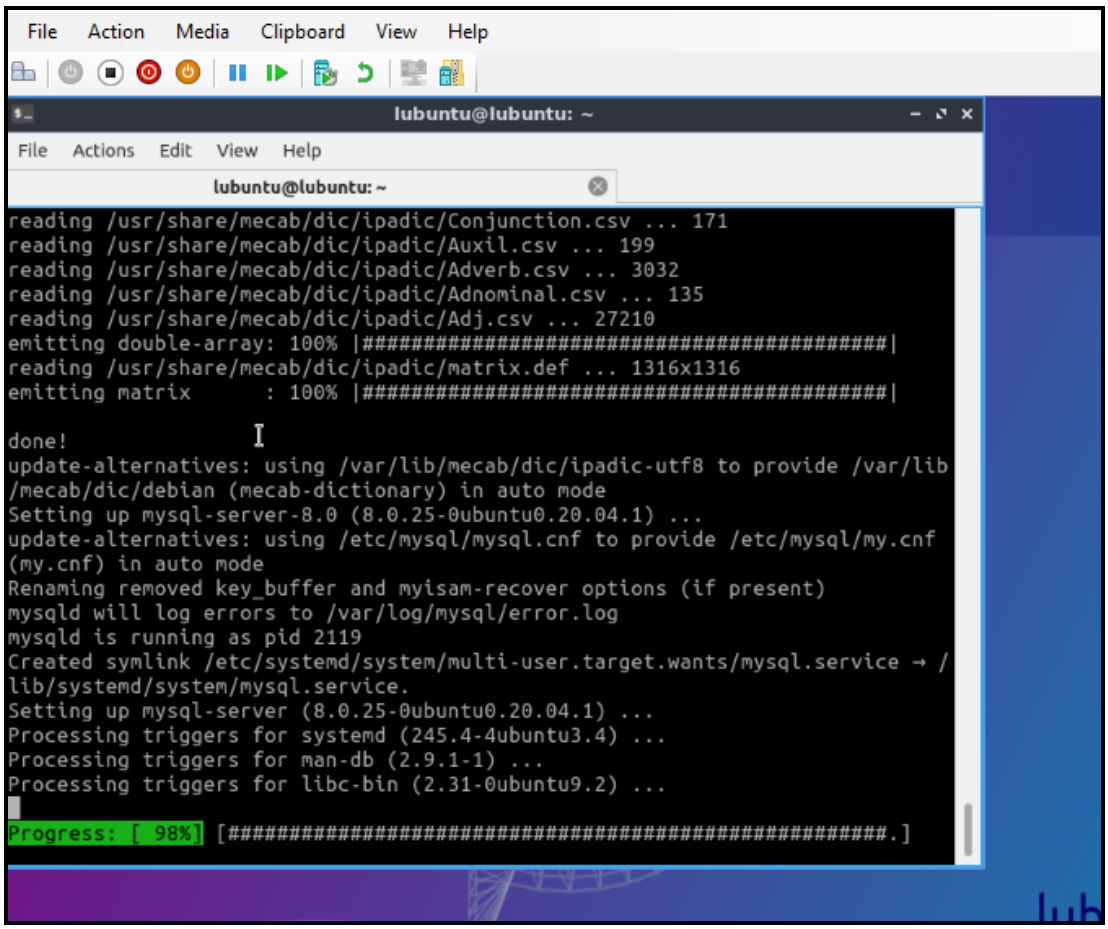
The Lubuntu ISO file was run using Hyper-V Hypervisor that can be run in Windows 10 Professional.



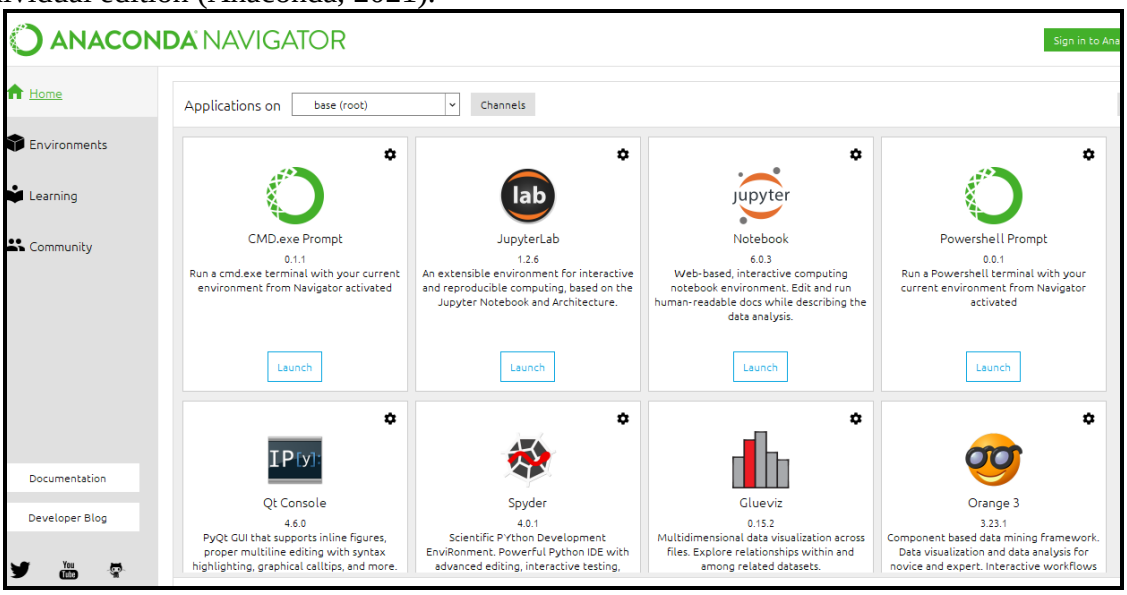
Using the terminal in Lubuntu – download and install the MYSQL Server application.

Instructions to install the MySQL Server were referenced in (Digital Ocean, 2021)

```
sudo apt install mysql-server
```



Downloading Python Environment on Windows 10 Host – *Anaconda*. Use the open-source individual edition (Anaconda, 2021).



Using the Spyder application, the Python scripts were developed.

2 Datasets

Raw Datasets (Samples):

Due to the size of the dataset only a sample is shown here. The SCADA dataset has many thousands of rows. Important to note the 3rd column. This represents time.

Problem:
DateTime type in MYSQL does not correspond to the SCADA dataset time format which is [12-03-2021 13:18:16] However, MySQL uses [2021-03-12 13:18:16].

Solution:
Pre-process the excel events list with a method of conversion via =TEXT(A4,"YYYY-MM-DD HH:MM:SS"). This can be then imported into the MYSQL database.

SCADA Power Grid Events Dataset:

46	2021-03-12 10:56:00	"2021-03-12 10:56:00	FELLDOWNEY	38	IKERRELLOW	AMPS	PASS LEV2HI SECT
47	2021-03-12 10:56:00	"2021-03-12 10:56:00	LISCOVE_NE	38	BOOSTER	CB	OPEN
48	2021-03-12 10:56:03	"2021-03-12 10:56:03	MONEYPOBAY	17	IND T4002	KV	OUT END MEASURMENT LEVEL
49	2021-03-12 10:56:04	"2021-03-12 10:56:04	CARMACROON	T68	TAP		OUT END MEASURMENT LEVEL
50	2021-03-12 10:56:05	"2021-03-12 10:56:05	GARBORROWE	20	CAHERDOWNEY	AMPS	PASS LEV2HI SECT
51	2021-03-12 10:56:05	"2021-03-12 10:56:05	GARBORROWE	20	CAHERDOWNEY	MW	PASS LEV2LO SECT
52	2021-03-12 10:56:07	"2021-03-12 10:56:07	MAREYCOVET	38	RIBBON VALLEY	MW	PASS LEV2HI SECT
53	2021-03-12 10:56:08	"2021-03-12 10:56:08	DODGEFIELD	220	GREAT ISLAND	AMPS	PASS LEV2HI SECT
54	2021-03-12 10:56:08	"2021-03-12 10:56:08	DRORYLAKES	20	KNOCNALOU/BALYNBRAOR	AMPS	NORMAL
55	2021-03-12 10:56:10	"2021-03-12 10:56:10	CARRIGTHOM	38	BAWNCOVE GEN	MW	PASS LEV4HI SECT
56	2021-03-12 10:56:10	"2021-03-12 10:56:10	TULLELGARR	110	HALDSOORO	KV	114
57	2021-03-12 10:56:11	"2021-03-12 10:56:11	GREETBAYGE	20	GREETBAY (IND END)	KV	PASS LEV2LO SECT
58	2021-03-12 10:56:11	"2021-03-12 10:56:11	FELLDOWNEY	38	IKERRELLOW	AMPS	PASS LEV4HI SECT
59	2021-03-12 10:56:18	"2021-03-12 10:56:18	SWALLEFORD		DISTRIBUTION SYST ALM		RESET
60	2021-03-12 10:56:20	"2021-03-12 10:56:20	CARRIGTHOM	38	BAWNCOVE GEN	MW	PASS LEV2HI SECT
61	2021-03-12 10:56:20	"2021-03-12 10:56:20	MONEYVALES	38	GORTAHILE GEN	MW	PASS LEV2HI SECT
62	2021-03-12 10:56:22	"2021-03-12 10:56:22	DOLLERTONE	38	PRAGUEE	KV	PASS LEV2HI SECT
63	2021-03-12 10:56:26	"2021-03-12 10:56:26	SWALLEFORD	10	T68	CB	CLOSED
64	2021-03-12 10:56:28	"2021-03-12 10:56:28	DRORYLAKES	20	KNOCNALOU/BALYNBRAOR	AMPS	PASS LEV2HI SECT

Site Login Application Dataset:
The Site Login Application Database was constructed as below. This was a hypothetical dataset showing the login and logout from certain Hypothetical High Voltage Sites.

datetime	log	plant	name
2021-03-12 10:54:11	In	MONEYVALES	John Mooney
2021-03-12 10:55:07	In	PIPERMILLY	Joe Bloggs
2021-03-12 10:56:04	In	SUNNTOYLAN	Linux Torvald
2021-03-12 10:56:11	In	LOUTHCARIG	Bill Gates
2021-03-12 10:57:05	In	BISHOPBRAC	Chris Krebs
2021-03-12 10:57:42	In	RECLORP789	Steve Jobs
2021-03-12 10:57:56	In	RECLORR034	Clint Eastwood
2021-03-12 10:58:00	In	CASTLEVIEW	Ada Cabrera
2021-03-12 10:58:13	In	DOCKERMEWS	Sidney Day
2021-03-12 10:58:19	In	DEPT_PLEXE	Marcia Howe
2021-03-12 10:58:19	In	EAST_MARSH	Alfreda Kennedy
2021-03-12 10:58:30	In	FAST__WALL	Stacey Michael
2021-03-12 11:00:07	In	SCION_CAPT	Prince Watkins
2021-03-12 17:29:20	Out	MONEYVALES	John Mooney
2021-03-12 17:29:23	Out	PIPERMILLY	Joe Bloggs
2021-03-12 17:29:24	Out	SUNNTOYLAN	Linux Torvald
2021-03-12 17:29:31	Out	LOUTHCARIG	Bill Gates
2021-03-12 17:29:31	Out	BISHOPBRAC	Chris Krebs
2021-03-12 17:29:32	Out	RECLORP789	Steve Jobs
2021-03-12 17:29:34	Out	RECLORR034	Clint Eastwood
2021-03-12 17:29:40	Out	CASTLEVIEW	Ada Cabrera
2021-03-12 17:30:01	Out	DOCKERMEWS	Sidney Day
2021-03-12 17:30:10	Out	DEPT_PLEXE	Marcia Howe
2021-03-12 17:30:11	Out	EAST_MARSH	Alfreda Kennedy
2021-03-12 17:30:17	Out	FAST__WALL	Stacey Michael
2021-03-12 17:30:17	Out	SCION_CAPT	Prince Watkins

Syslog/SNMP Dataset (Sample):
A sample of the Hypothetical Syslog Dataset is shown below. Here the communication and authentication messages are stored in the dataset upon receipt of associated syslog's into the centralised syslog server. SNMP messages from OT network infrastructure can also be stored here.

19400	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: COMMUNICATION FAILURE
19401	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: COMMUNICATION LINK DETECT
19402	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: AUTHENTICATION FAILURE
19403	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: COMMUNICATION LINK DETECT
19404	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: COMMUNICATION LINK DETECT
19405	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: AUTHENTICATION FAILURE
19406	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: COMMUNICATION FAILURE
19407	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: COMMUNICATION LINK DETECT
19408	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: AUTHENTICATION FAILURE
19409	2021-03-12 20:28:45	RECLOR9945	DISTRIBUTION AUTOMATION: COMMUNICATION FAILURE

Understanding the Datasets:
Important to understand how the datasets format the raw data. Need to know where we are likely to find certain data fields when it is delimited. This is the basis for using the Python Pandas data frames. See below for example of how the data is formatted with respect to expected field and character lengths.

11Characters8Characters	11Characters4Characters21Characters26Characters63Characters			
12/03/2021 10:54	SALLSFORD_	11	T67	OVD ALARM ON
12-03-2021 10:54:47	SALLSFORD_	11	T67	OVD PART 1 TRIP ALARM

11Characters8Characters	15Characters47Characters5Characters	
12-03-2021 11:31:50	MONEYTAHUL	DISTRIBUTION SYS ALERT RESET
12-03-2021 11:31:56	MONEYTAHUL	SIREN SWITCH OFF

11Characters8Characters	11Characters4Characters47Characters5Characters	
12-03-2021 11:32:38	PORTVIN	11KV E/F ALARM

11Characters8Characters	11Characters4Characters21Characters26Characters3Characters		
12-03-2021 11:27:06	KONTALLY	220 MONEYTAHUL RCL	OFF

11Characters8Characters	15Characters47Characters63Characters		
12-03-2021 11:34:07	KONTALLY	ACTIVE POWER ENABLE	ALARM
12-03-2021 11:34:17	KONTALLY	ACTIVE POWER ENABLE	NORMAL

11Characters8Characters	10Characters52Characters6Characters	
12-03-2021 11:35:47	FORTH	DISTIBUTION SYS ALERT ( ) NORMAL

3 Inputting the Databases to MySQL

Note the difficulty with importing dates of a different format into MYSQL. Use the STR_TO_DATE function for loading difficult datetime formats in csv file (StackOverFlow, 2021).

```
mysql> LOAD DATA LOCAL INFILE "/home/scada/Desktop/idsanonSYSlogs.csv" INTO TABLE idsanonsyslog.
syslogs FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES (Id, @var, Event) SET d
atetime=STR_TO_DATE(@var, '%Y-%m-%d %k:%i:%s');
```

Inputting a Dataset into MySQL (for example simulated 'Scada' system).

To enter MySQL use

- sudo -i

Then type:

- mysql -u root -p

```
scada@scada-virtualmachine:~$ sudo -i
root@scada-virtualmachine:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 8.0.25-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Show the Existing Databases

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| Login              |
| PowerEventList     |
| PowerEventList2    |
| PowerEventList3    |
| PowerEventList4    |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
9 rows in set (0.00 sec)

mysql>
```

Prepare to Create a new database. Exit out of MySQL and login again with the local-infile parameter.

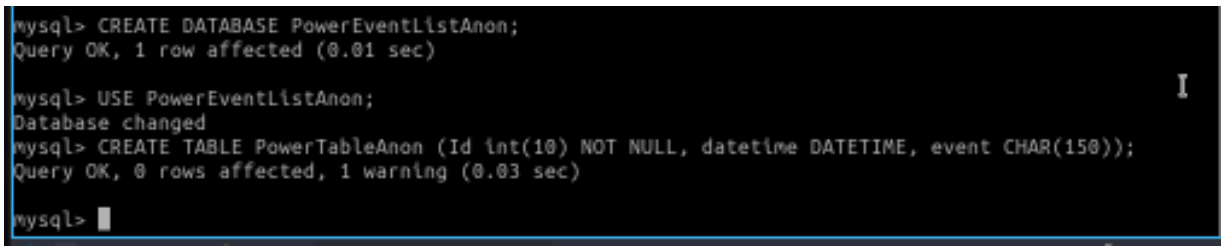
After login enter the command: SHOW GLOBAL VARIABLES ‘local_infile’;.

```
root@scada-virtualmachine:~# mysql -u root -p --local-infile
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 34
Server version: 8.0.25-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW GLABAL VARIABLES LIKE 'local_infile';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
 MySQL server version for the right syntax to use near 'GLABAL VARIABLES LIKE 'local_infile'' at li
ne 1
mysql> SHOW GLOBAL VARIABLES LIKE 'local_infile';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| local_infile  | OFF   |
+---------------+-------+
1 row in set (0.00 sec)

mysql>
```

Set the Local_infile Parameter to 1 (this prepares the database for input of formatted data)

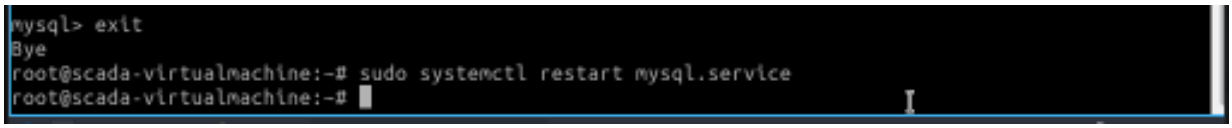
Create new Database and associated Table: New Database is called 'PowerEventListAnon'

Select the Database and create the Table 'PowerTableAnon'



Exit out of MySQL and restart the service from Linux command line:

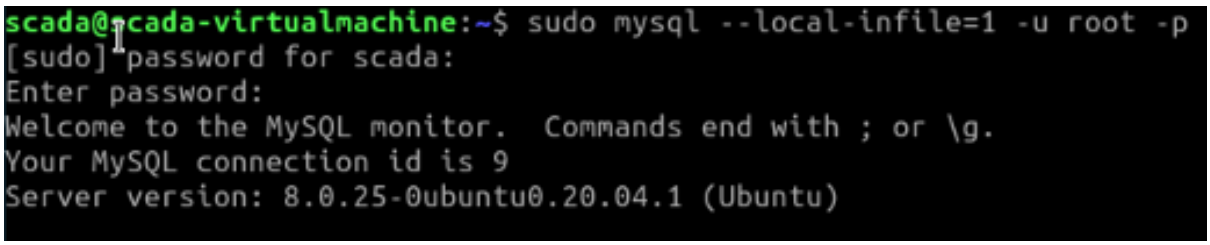
- Sudo systemctl restart mysql.service



Load the file to the database:

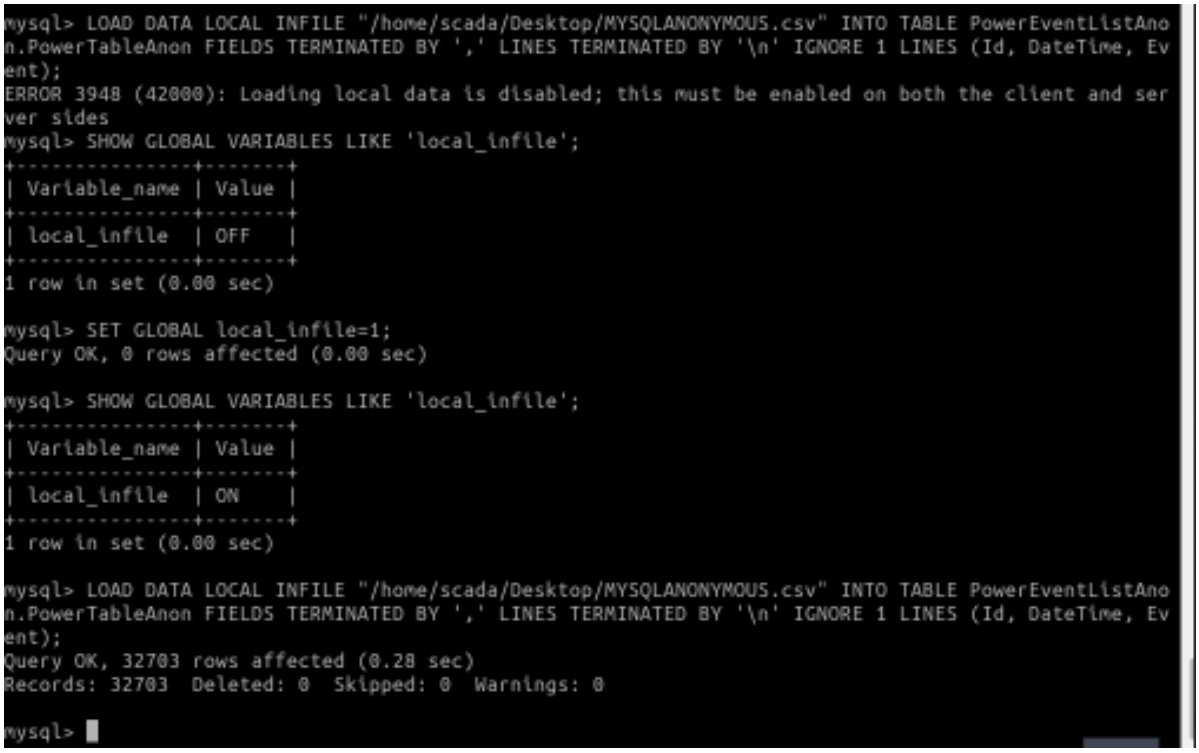
Enter MySQL. You will need to set the local_infile parameter back to 1 again. Load the data to the Database.Table in with the commands below.

There may be a permission issue. Enter the following for local infile.



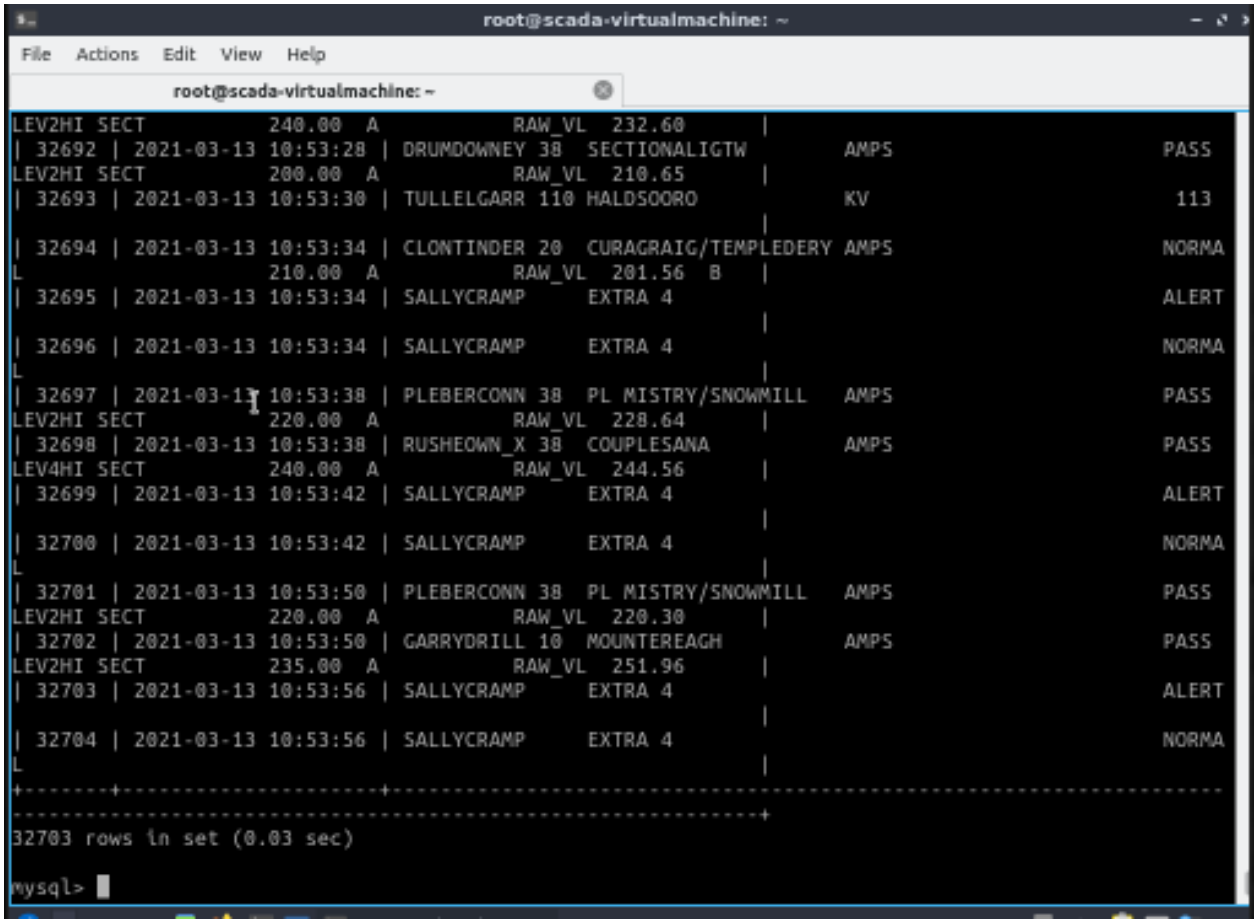
The command to enter the data into MySQL is:

LOAD DATA LOCAL INFILE "/home/scada/Desktop/MYSQLANONYMOUS.csv" INTO TABLE PowerEventListAnon.PowerTableAnon FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES (Id, DateTime, Event);

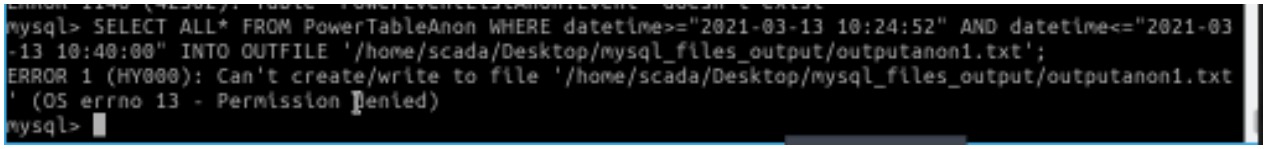


Check the data is in the database:

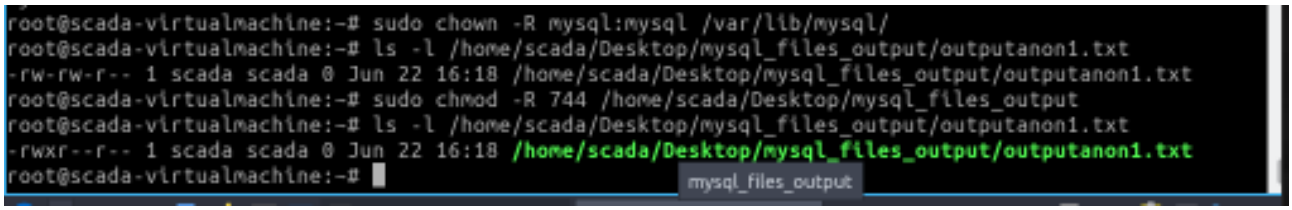
- Mysql> USE DATABASE PowerEventListAnon;
- Mysql> SELECT ALL* FROM Events.



Output the results of the query to a file such as CSV. There may be a permission issue.



The variable secure_file_priv is used to limit data import and export operations. This requires adjustment as follows:



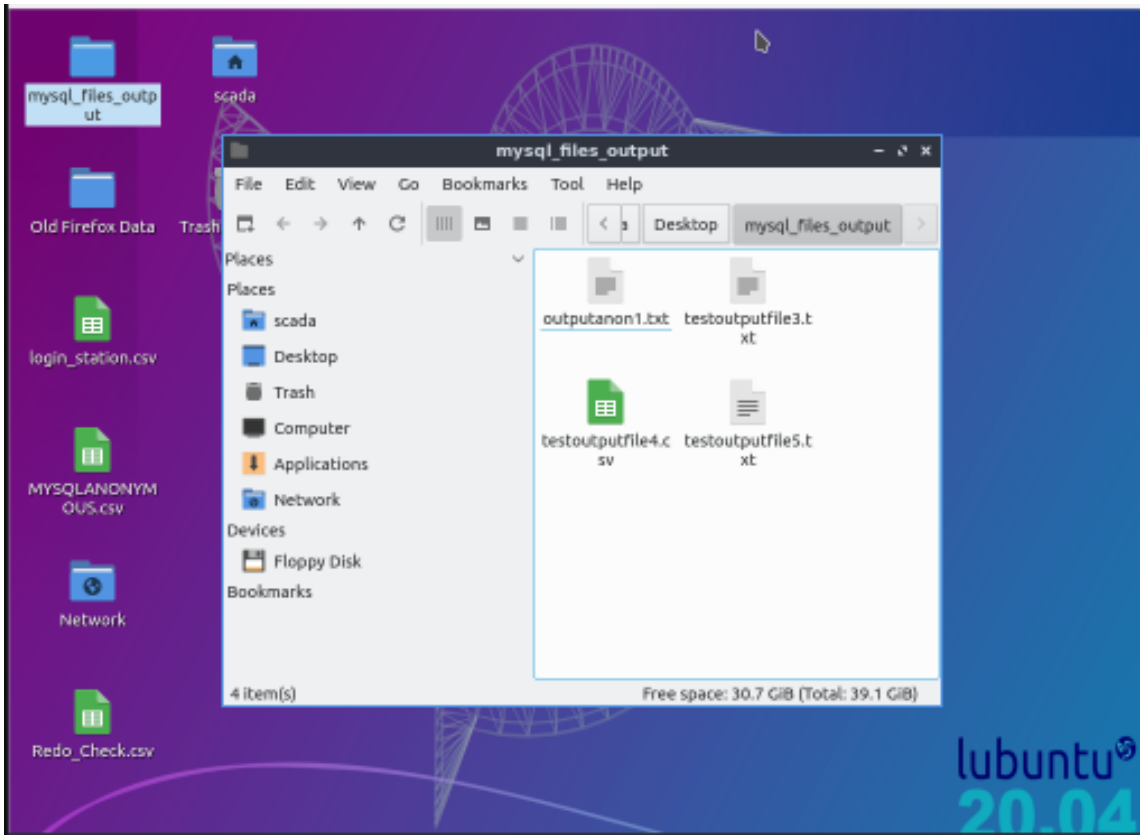
The -rwxr—r— indicates that generated files from the database can be read and written to and can be executable.

For example, we can create the test file (outputanon1.txt) using an appropriate SQL Select Query.

Copy it to the desktop folder for convenient access.

```
root@scada-virtualmachine:~# sudo cp /var/lib/mysql-files/outputanon1.txt /home/scada/Desktop/mysql
_files_output/outputanon1.txt
root@scada-virtualmachine:~#
```

Then it is in desktop folder mysql_files_output.



Set permissions for other Users to query the database. For example, we can set permission for Users ‘scada’ and ‘ids’. We can also associate the user with an IP address. If User ‘ids’ is a remote user on another Host (with IP Address 10.10.10.20) then permissions can be setup accordingly to access the database from the remote host. The ‘scada’ is associated with IP address 10.10.10.10.

GRANT ALL PRIVILEGES ON PowerEventListAnon TO ‘ids’@’10.10.10.20’;

GRANT ALL PRIVILEGES ON PowerEventListAnon TO ‘scada’@’10.10.10.10’;

```
mysql> USE PowerEventListAnon;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> GRANT ALL PRIVILEGES ON PowerEventListAnon TO 'ids'@'10.10.10.20';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON PowerEventListAnon TO 'scada'@'10.10.10.10';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON PowerEventListAnon TO 'user'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILIGES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
 MySQL server version for the right syntax to use near 'PRIVILIGES' at line 1
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Also, grant permission to the table PowerTableAnon. Example for user ‘ids is below.’

```
mysql> GRANT ALL PRIVILEGES ON PowerTableAnon TO 'ids'@'10.10.10.20';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Exit MySQL and restart services:

- Sudo systemctl restart mysql.service

Enter the ‘Site Login Dataset’ to correlate against the SCADA Power Events.

Like the way the PowerEventListAnon Database, create Database ‘Login’ using appropriate fields. Select ‘Use Login’ Database.

```
mysql> USE Login
```

Create the Table ‘Stationlog_anon’ with fields as follows:

```
mysql> CREATE TABLE Stationlog_anon (datetime DATETIME, log CHAR(5), plant CHAR(11), name VARCHAR(5
0));
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Grant ALL Privileges for user ids for both database Login and the new table ‘Stationlog_anon’.

```
mysql> GRANT ALL PRIVILEGES ON Login TO 'ids'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON Login TO 'ids'@'10.10.10.20';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON Stationlog_anon TO 'ids'@'10.10.10.20';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Also, perform this for ‘Scada’ user.

```
mysql> GRANT ALL PRIVILEGES ON Stationlog_anon TO 'scada'@'10.10.10.10';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON Login TO 'scada'@'10.10.10.10';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

Also remember to ‘FLUSH PRIVILEGES’.

Exit MySQL and restart services:

- Sudo systemctl restart mysql.service

Load the Infile stored on the desktop (called anon_login1.csv)

```
mysql> LOAD DATA LOCAL INFILE "/home/scada/Desktop/anon_login1.csv" INTO TABLE Login.Stationlog_ano
n FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES (datetime, log, plant, name);
Query OK, 26 rows affected (0.01 sec)
Records: 26  Deleted: 0  Skipped: 0  Warnings: 0

mysql>
```

Enter the ‘Syslog Dataset’ to correlate against the SCADA Power Events.

Create the new Table for the syslog dataset.

```
mysql> CREATE TABLE syslogs (id int(5), datetime DATETIME, event CHAR(150));
```

Like the method for the previous database tables. Select the Database and load the raw data from the dataset ‘idsanonSYSlog.csv’ to the new table ‘Syslogs’ as follows:

```
mysql> LOAD DATA LOCAL INFILE "/home/scada/Desktop/idsanonSYSlogs.csv" INTO TABLE idsanonsyslog.
syslogs FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES (Id, @var, Event) SET d
atetime=STR_TO_DATE(@var, '%Y-%m-%d %k:%i:%s');
```

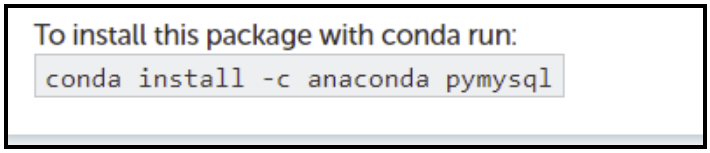
Also, grant permission to the table idsanonsyslog.syslogs. Then Flush Privileges.

4 Mechanism to Query the Databases from Application

The Python IDS application will be running on the Windows 10 Machine. The Lubuntu Operating System that is hosting the MySQL Database is a guest OS virtualised via the Hyper-V hypervisor supplied under Windows 10 professional. For the Python IDS application to access the data held within the MySQL database tables, the application will use an SSH tunnel module to connect securely to the remote database and query the databases as needed. To import the required packages to implement the SSH tunneller module the following steps were performed using the Anaconda Environment.

SSH Tunnelling with Python to Remote Computer: Download SSH Tunnel module in the Anaconda environment (pypi.org, 2021).

Also install module for pymsql (Anaconda.org, 2021).



For different versions of the SSH Tunneler we also need to import (Geeksforgeeks.org, 2021):

```
conda install -c anaconda mysql-python
conda install -c anaconda mysql-connector-python
```

Unfortunately mysql-python may not install due to incompatibity with Python 3.7. In order to get the necessary package to work (i.e. Mysqldb) we need to do a pip install on the anaconda base command line i.e.

```
(base) PS C:\Users\sando> pip install mysqlclient
Collecting mysqlclient
  Downloading mysqlclient-2.0.3-cp37-cp37m-win_amd64.whl (178 kB)
     |████████████████████████████████| 178 kB 726 kB/s
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.0.3
(base) PS C:\Users\sando>
```

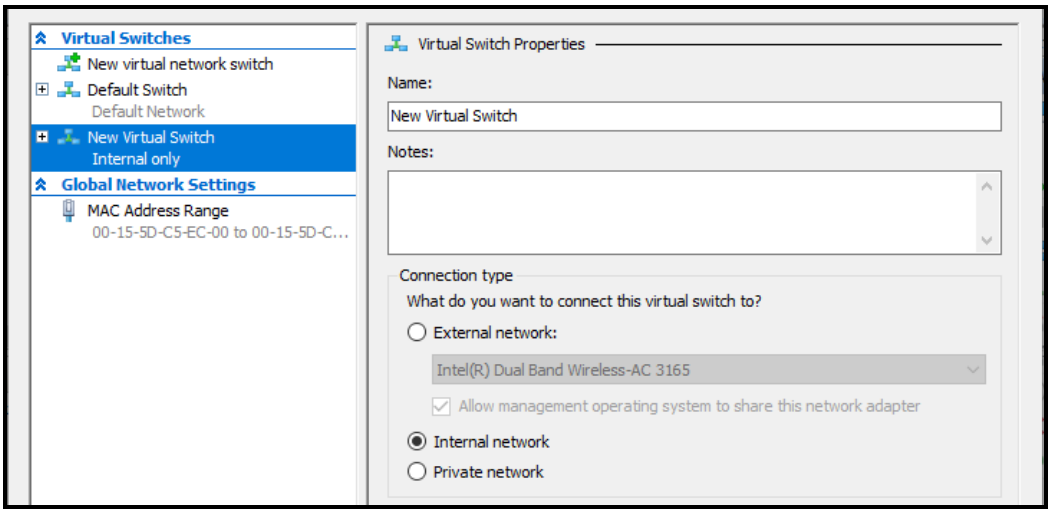
This will enable the SSH Tunneler program to work and query the remote database via select and store to the Pandas dataframes.

For the code to open the SSH tunnel refer to the StackOverflow forum (Stackoverflow, 2021).

The SSH Tunneller module can be referred to in the Python Implementation Section.

5 Windows 10 Host connection to Lubuntu Guest VM

To allow the communication (SSH protocol) between the Windows Host and the Hyper-V Guest (Lubuntu) Operating System, an internal network must be setup between the Host and the Guest via Hyper-V.



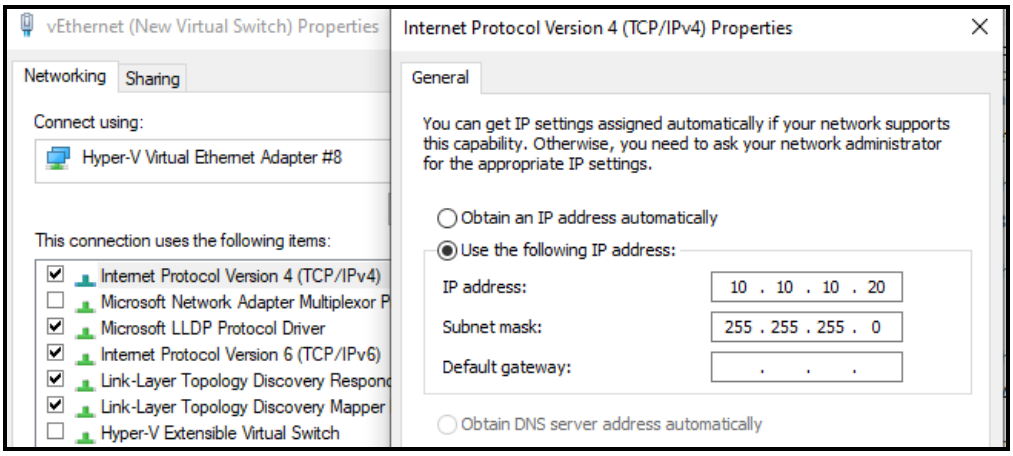
To allow the communication (SSH protocol) the Windows Firewall must be configured to allow python application (anaconda3) for the Guest VM.

The Internal Network for the respective Operating Systems is set to the same subnet:

Lubuntu Guest: inet **10.10.10.10**

```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.10  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::2fc:e8c6:f753:f7a  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:c5:ec:07  txqueuelen 1000  (Ethernet)
        RX packets 151623  bytes 6927663 (6.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 96  bytes 10082 (10.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Windows 10 Hyper-V Network Adapter: IP Address set to **10.10.10.20**



6 Python Implementation of OT IDS Application

There are several components that enable the implementation of IDS application. The important features are as follows:

- SSH Tunneller Module (query the MySQL Database from Remote Computer running the IDS application)
- Ruleset Modules (there are 8 power grid operational technology 'Use Cases' discussed in the report, therefore 8 Rules are coded in the IDS application)
- Alert Generation No. 1 – Custom email alert (via a dedicated Gmail account)
- Alert Generation No. 2 – Custom Microsoft Event (can be viewed in the OS Event Viewer. These events could be interrogated by a Security Information Event Management System e.g., Q-Radar SIEM)

SSH Tunneller Module:
The tunneller connects to the localhost via a referred port behind the SSH server i.e., it uses the remote operating systems localhost and MySQL port to connect to the database.

```
import MySQLdb as db
import pandas as pd
import csv

def ssh_import():

    def query(q):
        with SSHTunnelForwarder(
            ('10.10.10.10', 22),
            ssh_username='scada',
            ssh_password=[redacted],
            remote_bind_address=('127.0.0.1', 3306)
        ) as server:
            conn = db.connect(host='127.0.0.1',
            port=server.local_bind_port,
            user='scada',
            passwd[redacted],
            db='idsanonsyslog')
            return pd.read_sql_query(q, conn)

    # Put teh the SQL Query here to a pandas dataframe df
    select_idsanonsyslog = query('SELECT datetime, event FROM idsanonsyslog.syslogs')
```

The result of the query is then stored into a Python Pandas dataframe, for example 'select_idsanonsyslog'. The SSH Tunneller has been wrapped in function call ssh_import(). In this way it is possible to call the function from the 'main' Python Program.

Refer to stackoverflow site (StackOverFlow, 2021):

Ruleset Modules:

An example code for Rule 8 – Distribution Automation: Layer 2 Authentication 802.1x, is shown below. Here the rule checks the appropriate dataset for 802.1x Supplicant authentication Failures. If the event is present in the database, the result is stored in the Pandas dataframe ***keyword_SAS_802.1X_rule_8***

```
import numpy as np
import smtplib, ssl
import win32evtlogutil
import win32evtlog
import sys
import time
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from IPython.display import HTML

def ids_rule_8():


    # ------------------------------------ Beginning of Rule 8: DISTRIBUTION AUTOMATION: Layer 2 Authentication 802.1x ------------------------------------------------

    # to handle the embedded quotes in the imported file, we must use quoting=csv.Quote_None. This requires import of csv type above. The engine python (instead of C) is more complete.
    SAS_802_1X_rule_8 = pd.read_csv(r'C:\Users\sando\Desktop\[redacted]python snips\import_idsanonsysLog_
    print(SAS_802_1X_rule_8)
    # (Optional) Further Clean the Data by Deleting the first Column which may be superflous. Variations can be used for other columns or rows [startrow:endrow, startcolumn:endcolumn].
    SAS_802_1X_rule_8 = SAS_802_1X_rule_8.iloc[: , 1:]
    # Demonstrate the file can be stored to a .txt file with appropriate column names above (i.e. header ='infer').
    get_csv_file = SAS_802_1X_rule_8.to_csv(r'C:\Users\sando\Desktop\[redacted]ts\python snips\SASLoginRule4
    # Search the DataFrame 'data_selected_1' for values containing a given string (e.g. 'TRIP') and output the associated rows variable 'keyword_dataframe' to another file
    # In the below command the search is conducted via the 'event' column and it searches for 'TRIP' string in that column
    keyword_SAS_802_1X_rule_8 = SAS_802_1X_rule_8[SAS_802_1X_rule_8['event'].str.contains('DISTRIBUTION AUTOMATION: Layer 2 Authentication')]
    print(keyword_SAS_802_1X_rule_8)

    # Demonstrate the file generation with the keyword search
    get_keyword_csv = keyword_SAS_802_1X_rule_8.to_csv(r'C:\Users\sando\Desktop[redacted]python snips\SA
```

The actual path used for the Operating System that runs the IDS has been blocked out as it is specific to the PC user (sando). The rule has wrapped in a function called **ids_rule_8()** so it can be called by the 'Main' Python Program.

The other 7 rules are coded in a similar manner. The relevant database tables are queried, and the results are stored to Pandas dataframes. These are then checked based on the logic described in the OT IDS application Use Cases.

Alert Generation No. 1 – Custom Email Alerts:
To demonstrate the alerting functionality via email, a test Gmail account was setup to act as the 'sender email'. The code below accesses the sender Gmail account and provides the extracted data in the form of a HTML format. The information in the alert is converted to HTML format using the method to_html().

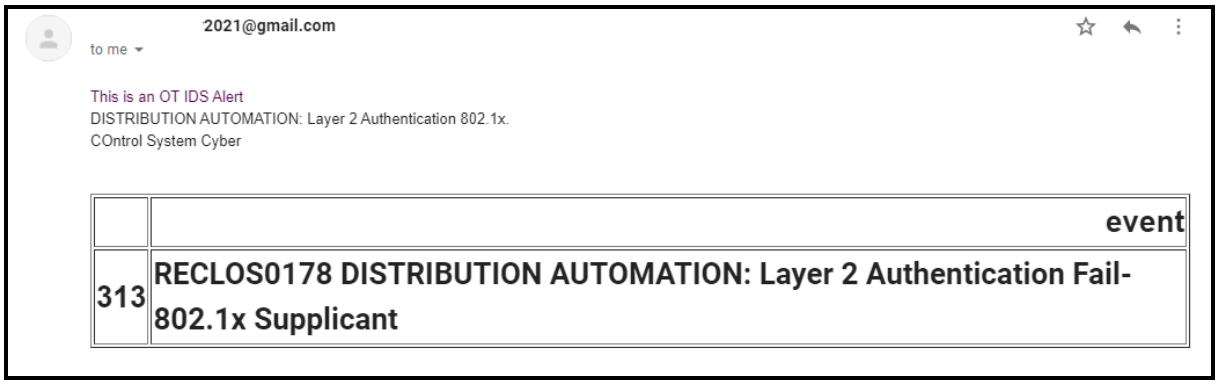
some_result = keyword_SAS_802.1X_rule_8.to_html()

The example of Rule 8 – Distribution Automation: Layer 2 Authentication 802.1x is provided below. As well as the specific information detected by the IDS with respect to the datasets, the alert also provides more generic information into the subject line to enable the analyst in the SOC to quickly determine the nature of the alert i.e. 'An OT System Alert'.

The email alerting code is also contained within the respective rule module i.e., for Rule 8 the code is wrapped in the Rule 8 function **ids_rule_8().**

```
# ------------------------------------------------------------ Send an the Alert Email
some_result = keyword_SAS_802_1X_rule_8.to_html()
print(some_result)

sender_email = "[redacted]2021@gmail.com"
receiver_email = "[redacted]@gmail.com"
password = "[redacted]"
[redacted]

message = MIMEMultipart("alternative")
message["Subject"] = "OT System Alert"
message["From"] = sender_email
message["To"] = receiver_email

# Create the plain-text and HTML version of your message
text = """\
This is an OT IDS Alert
DISTRIBUTION AUTOMATION: Layer 2 Authentication 802.1x

"""

# Print out the alerted variable from the IDS in HTML i.e. {some_result}
html = """\
<html>
  <body>
    <p>This is an OT IDS Alert<br>
       DISTRIBUTION AUTOMATION: Layer 2 Authentication 802.1x.<br>
       COntrol System Cyber<br>
       <br><h1>{some_result}</h1><br>
    </p>
  </body>
</html>
""".format(some_result=some_result)


# Turn these into plain/html MIMEText objects
part1 = MIMEText(text, "plain")
print(part1)
part2 = MIMEText(html, "html")
# Add HTML/plain-text parts to MIMEMultipart message
# The email client will try to render the last part first
message.attach(part1)
message.attach(part2)

# Create secure connection with server and send email
context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
    server.login(sender_email, password)
    server.sendmail(
        sender_email, receiver_email, message.as_string()
    )
```

Refer to (StackOverFlow, 2021b). The resulting email alert for a positive detection of Rule 8 can be seen below.

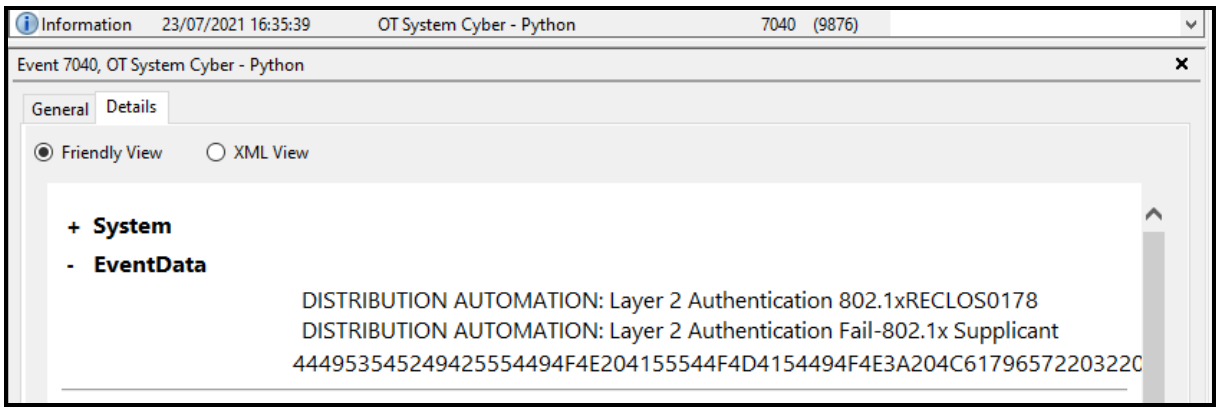


Alert Generation No. 2 – Custom MS Event:
To demonstrate the alerting functionality via the MS Event generation the code below takes the data that has been positively detected by the Pandas dataframe search and populates it into the appropriate format used by the system event logs.

```
# Put in the MS Event log code here for rule 8.
        No documentation available
App_Name = "OT System Cyber - Python"
App_Event_ID = 7040
App_Event_Category = 9876
#App_Event_Type = win32evtlog.EVENTLOG_WARNING_TYPE
App_Event_Str = ["DISTRIBUTION AUTOMATION: Layer 2 Authentication 802.1x{}".format(var) for var in keyword_SAS_802_1X_rule_8['event']]
App_Event_Data= b"DISTRIBUTION AUTOMATION: Layer 2 Authentication 802.1x"
'''

win32evtlogutil.ReportEvent(ApplicationName, EventID, EventCategory,EventType,Inserts, Data, SID)

'''
win32evtlogutil.ReportEvent(App_Name,App_Event_ID, eventCategory= App_Event_Category, eventType=win32evtlog.EVENTLOG_INFORMATION_TYPE, strings=App_Event_Str,data=App_Event_Data)
```

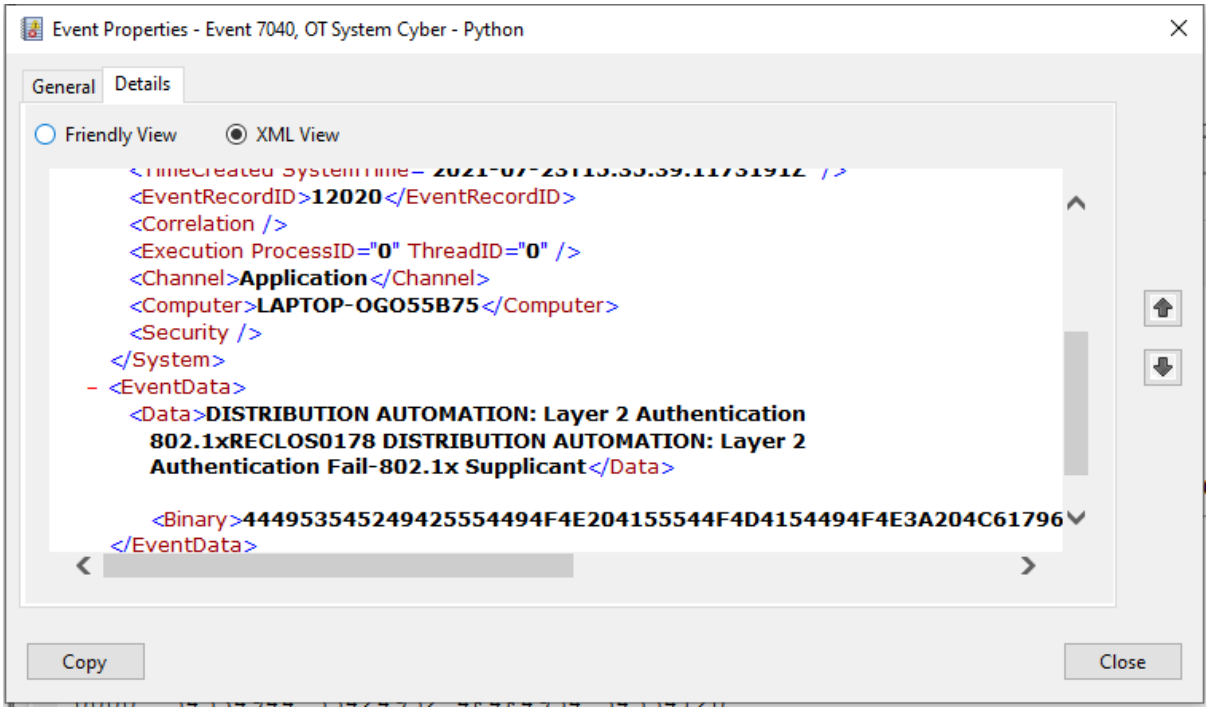
The above code is provided from the example Rule 8 function. Like the email alert, the above code is wrapped within the overall rule function i.e., Rule 8 function **ids_rule_8().** When the rule 8 module is called by the main program it executes the email alert and system log event each time it is called. An example of the system log event generated from a positive detection of rule 8 is show below.

Refer to (StackOverFlow, 2021).



Other Data view is also available from the event viewer to show the detected alert information can be put into the XML format that can be read by other systems.

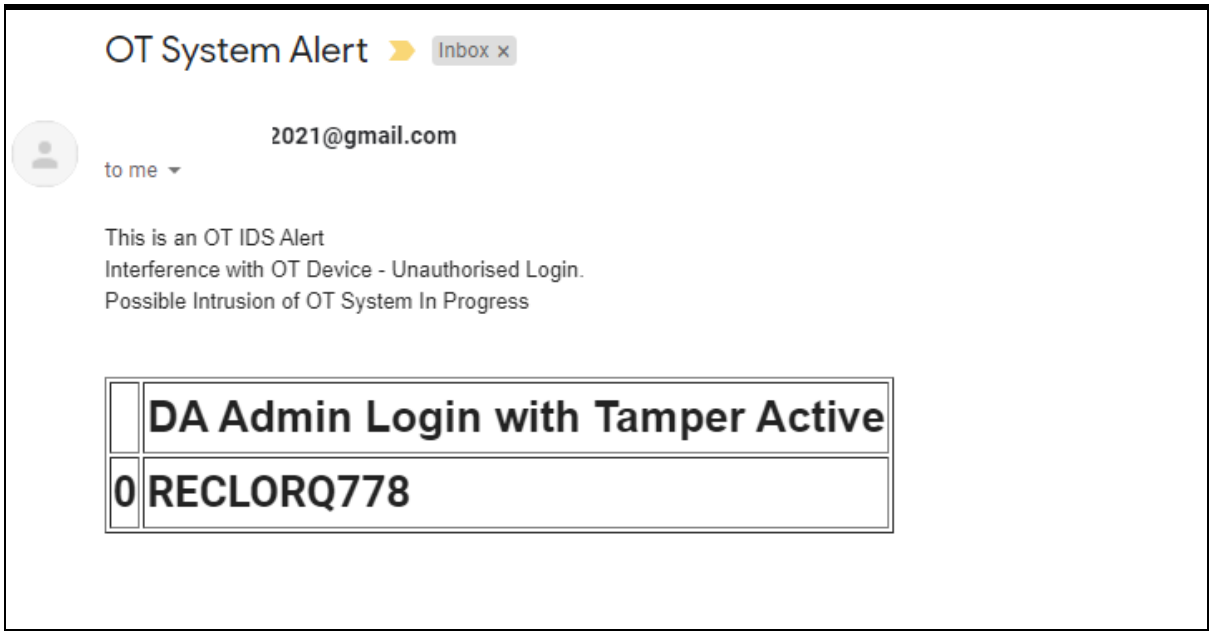
Note that the system log is also timestamped by the Host System i.e. event logged on **23/07/2021** at **16:35:39**.



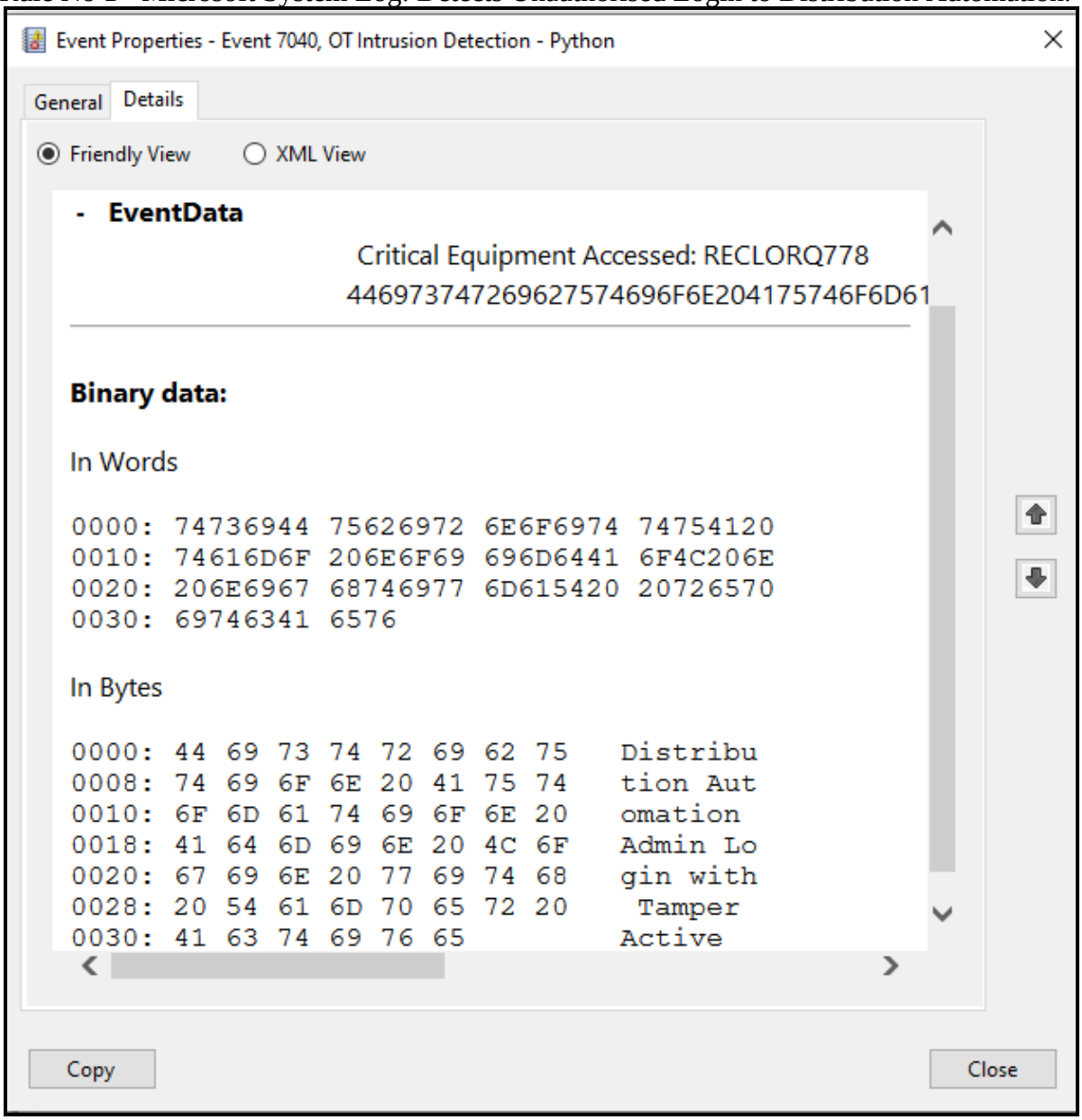
Summary of OT IDS Alerting for remaining Rules 1 to 7:

Rule No 1. Call's function ids_rule_1() in Main Program.

Rule No 1 - Email Alert: The Device RECLORQ778 is Subject to Possible Tampering.

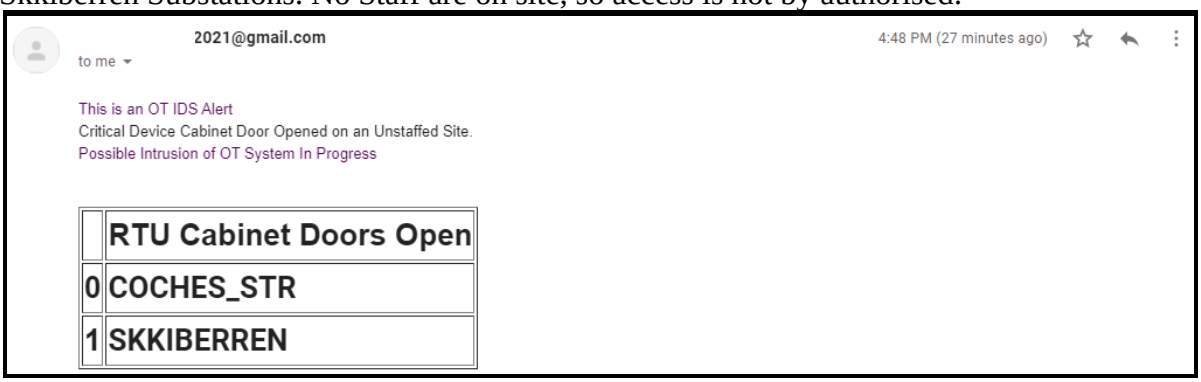


Rule No 1 - Microsoft System Log: Detects Unauthorised Login to Distribution Automation.

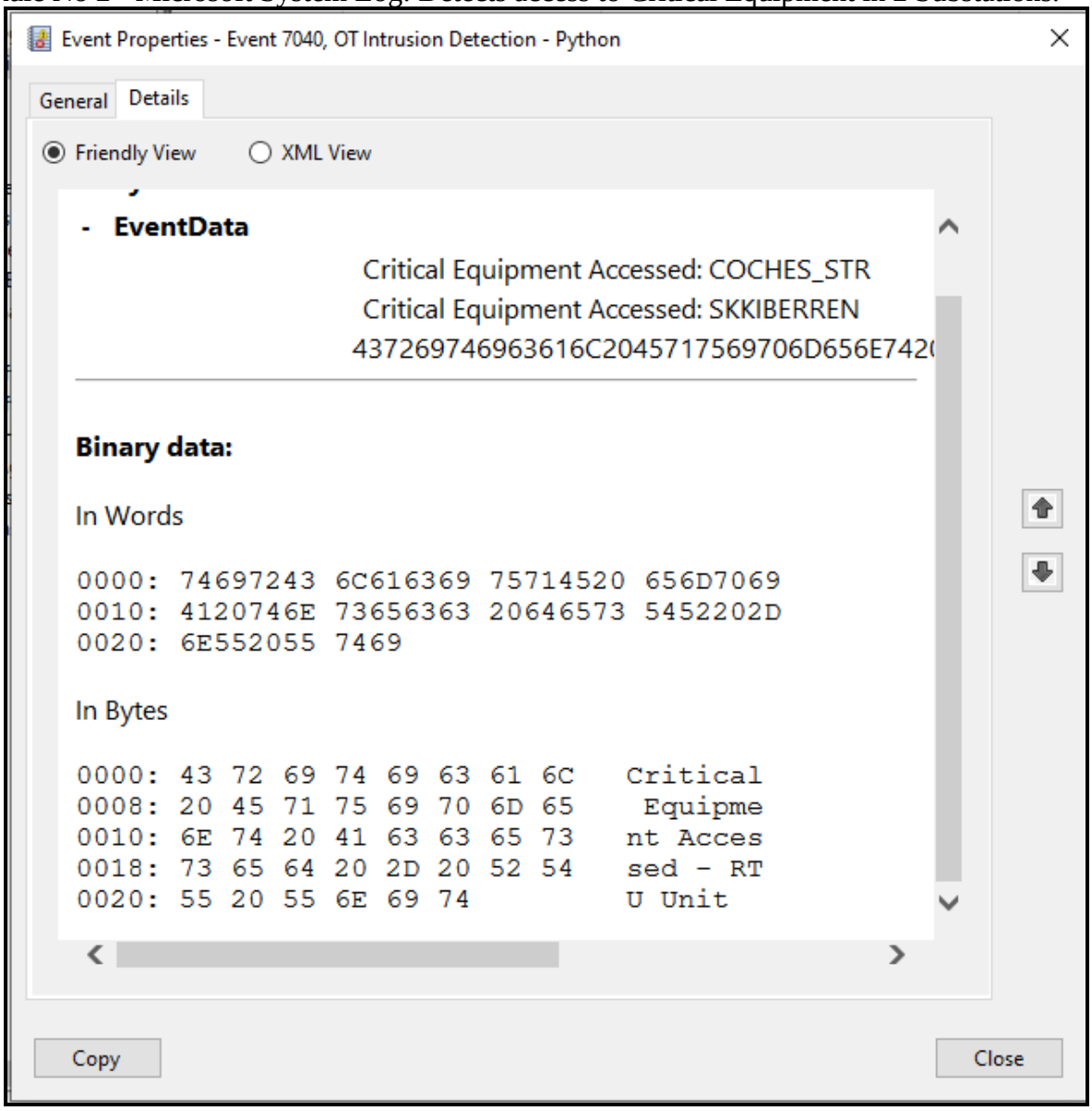


Rule No 2. Call's function ids_rule_2() in Main Program.

Rule No 2 - Email Alert: Unauthorised Access to Critical Equipment in Coches_Str and Skkiberren Substations. No Staff are on site, so access is not by authorised.

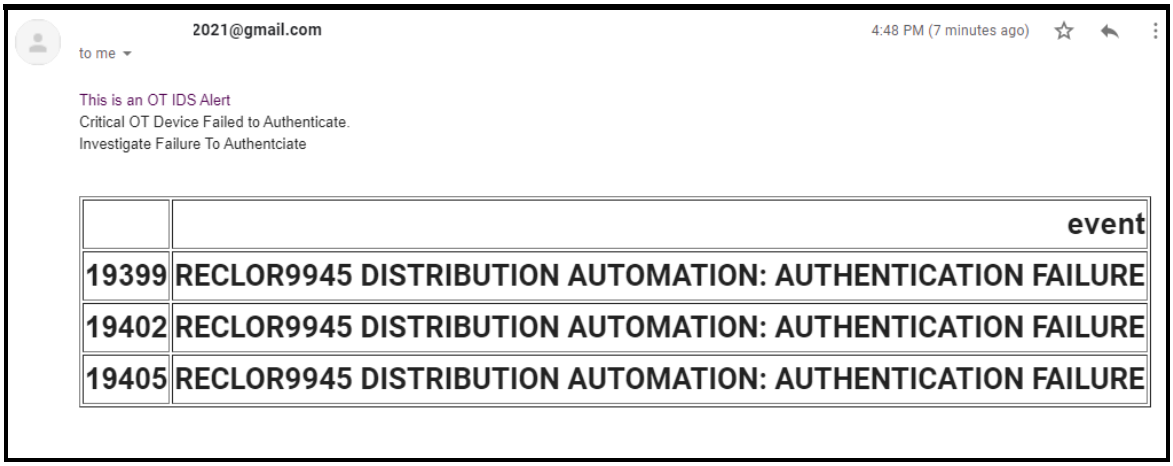


Rule No 2 - Microsoft System Log: Detects access to Critical Equipment in 2 Substations.



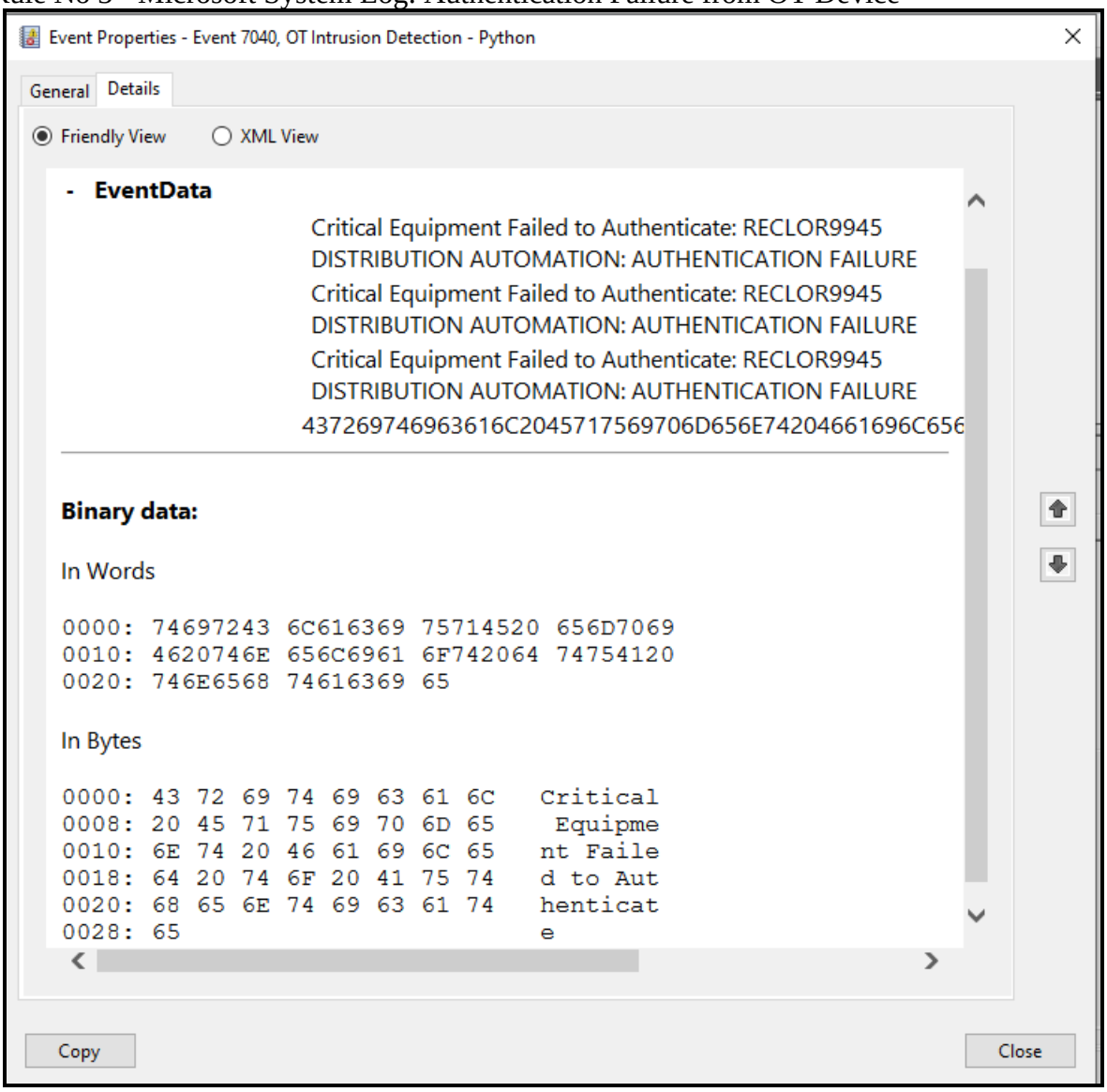
Rule No 3. Call's function ids_rule_3() in Main Program.

Rule No 3 - Email Alert: An OT Device has failed to Authenticate to the Network



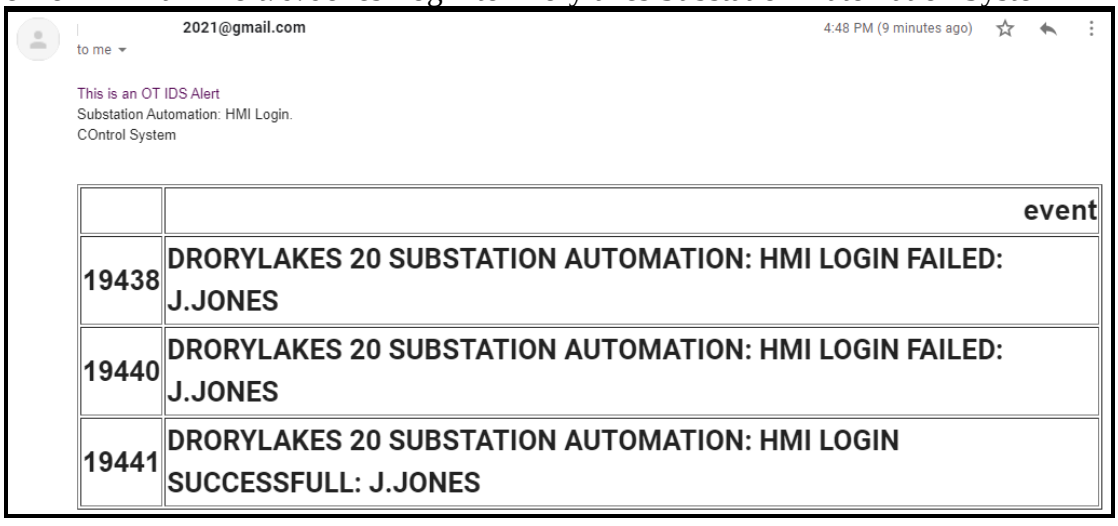
	event
19399	RECLOR9945 DISTRIBUTION AUTOMATION: AUTHENTICATION FAILURE
19402	RECLOR9945 DISTRIBUTION AUTOMATION: AUTHENTICATION FAILURE
19405	RECLOR9945 DISTRIBUTION AUTOMATION: AUTHENTICATION FAILURE

Rule No 3 - Microsoft System Log: Authentication Failure from OT Device



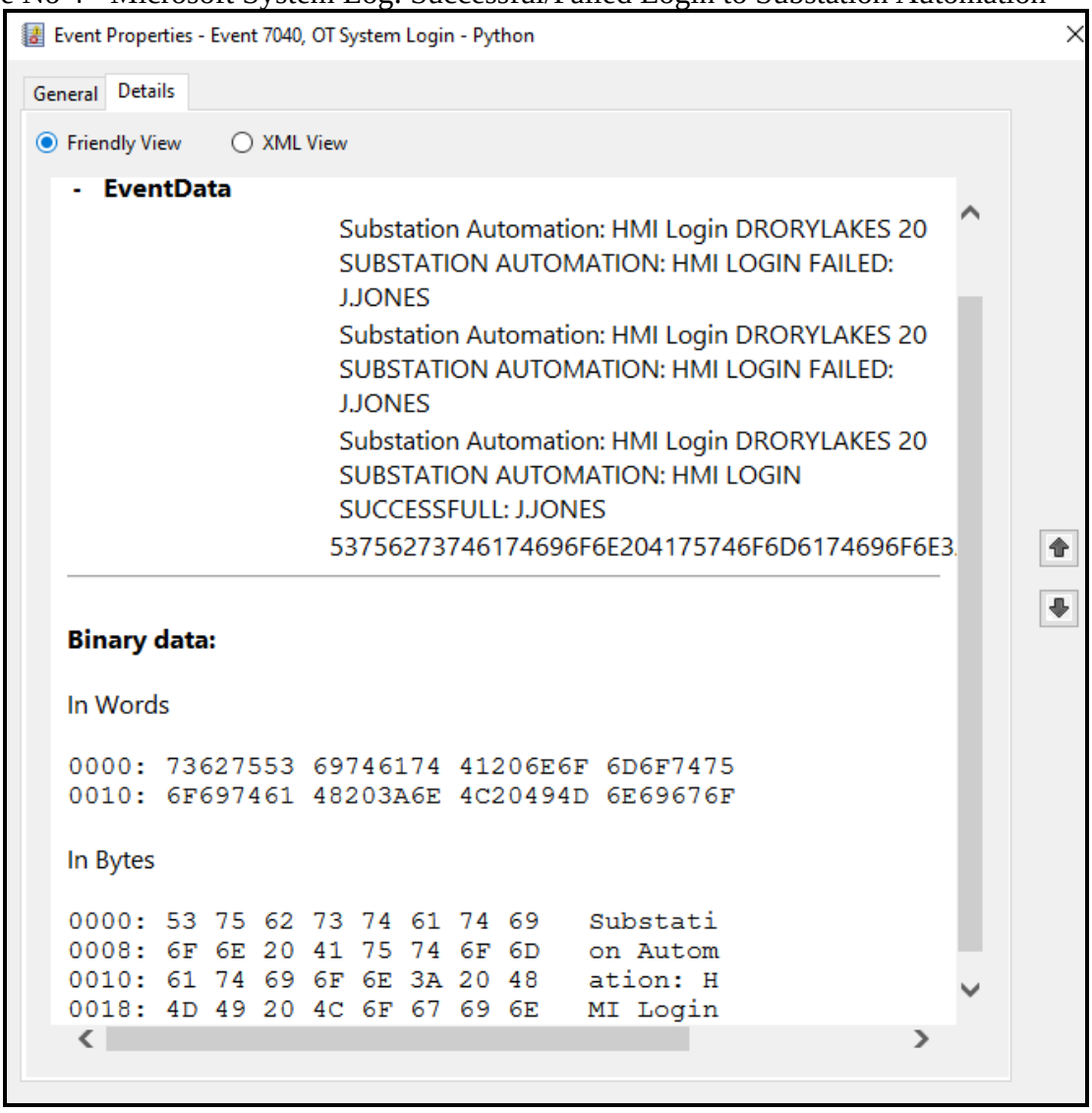
Rule No 4. Call's function ids_rule_4() in Main Program.

Rule No 4 - Email Alert: J. Jones Login to Drorylakes Substation Automation System



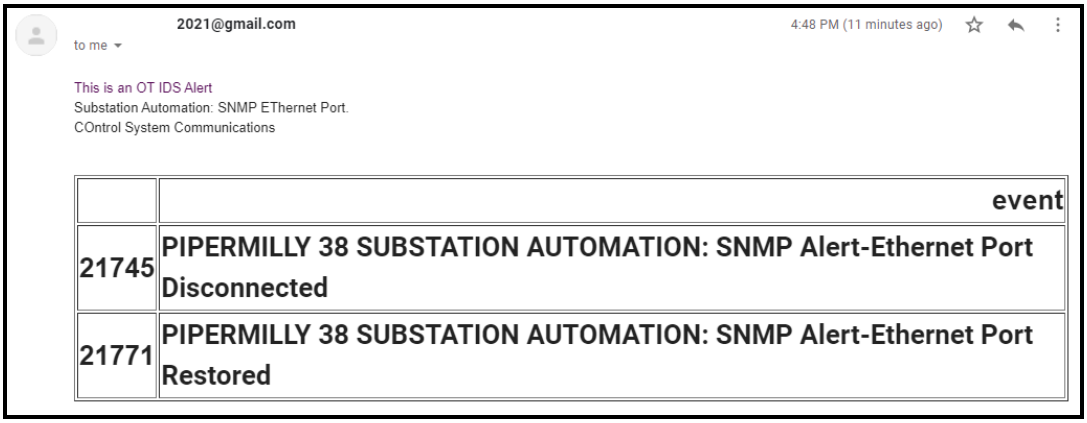
	event
19438	DRORYLAKES 20 SUBSTATION AUTOMATION: HMI LOGIN FAILED: J.JONES
19440	DRORYLAKES 20 SUBSTATION AUTOMATION: HMI LOGIN FAILED: J.JONES
19441	DRORYLAKES 20 SUBSTATION AUTOMATION: HMI LOGIN SUCCESSFULL: J.JONES

Rule No 4 - Microsoft System Log: Successful/Failed Login to Substation Automation



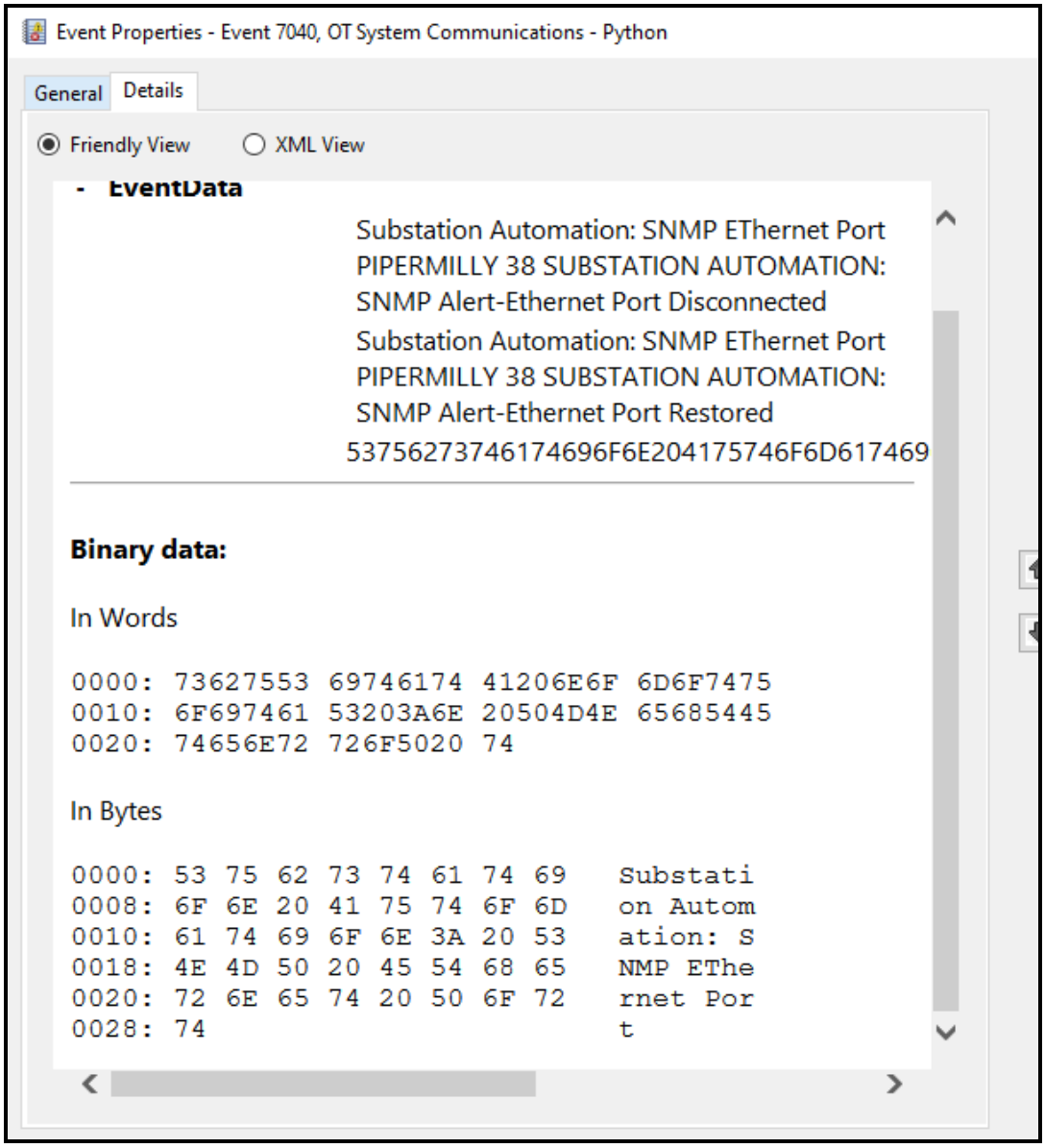
Rule No 5. Call's function ids_rule_5() in Main Program.

Rule No 5 - Email Alert: Pipermilly Substation Automation Port Disconnection Detected



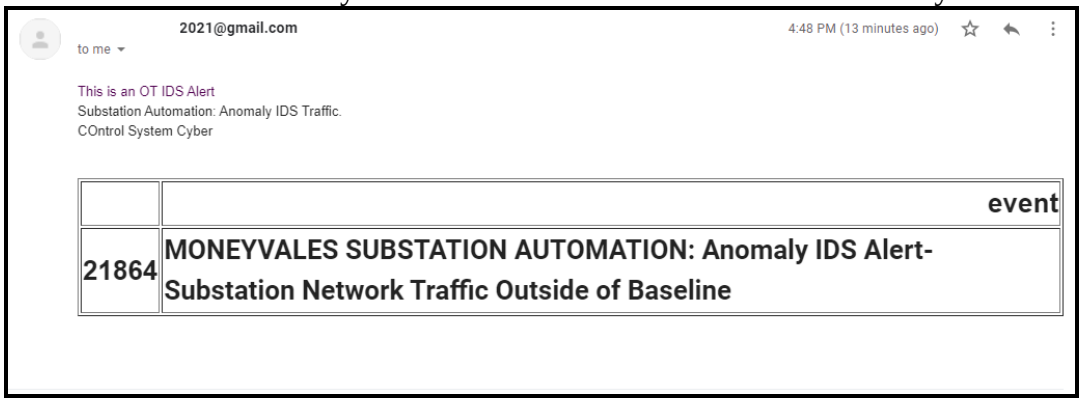
	event
21745	PIPERMILLY 38 SUBSTATION AUTOMATION: SNMP Alert-Ethernet Port Disconnected
21771	PIPERMILLY 38 SUBSTATION AUTOMATION: SNMP Alert-Ethernet Port Restored

Rule No 5 - Microsoft System Log: SNMP Port Failure Detection in OT Network

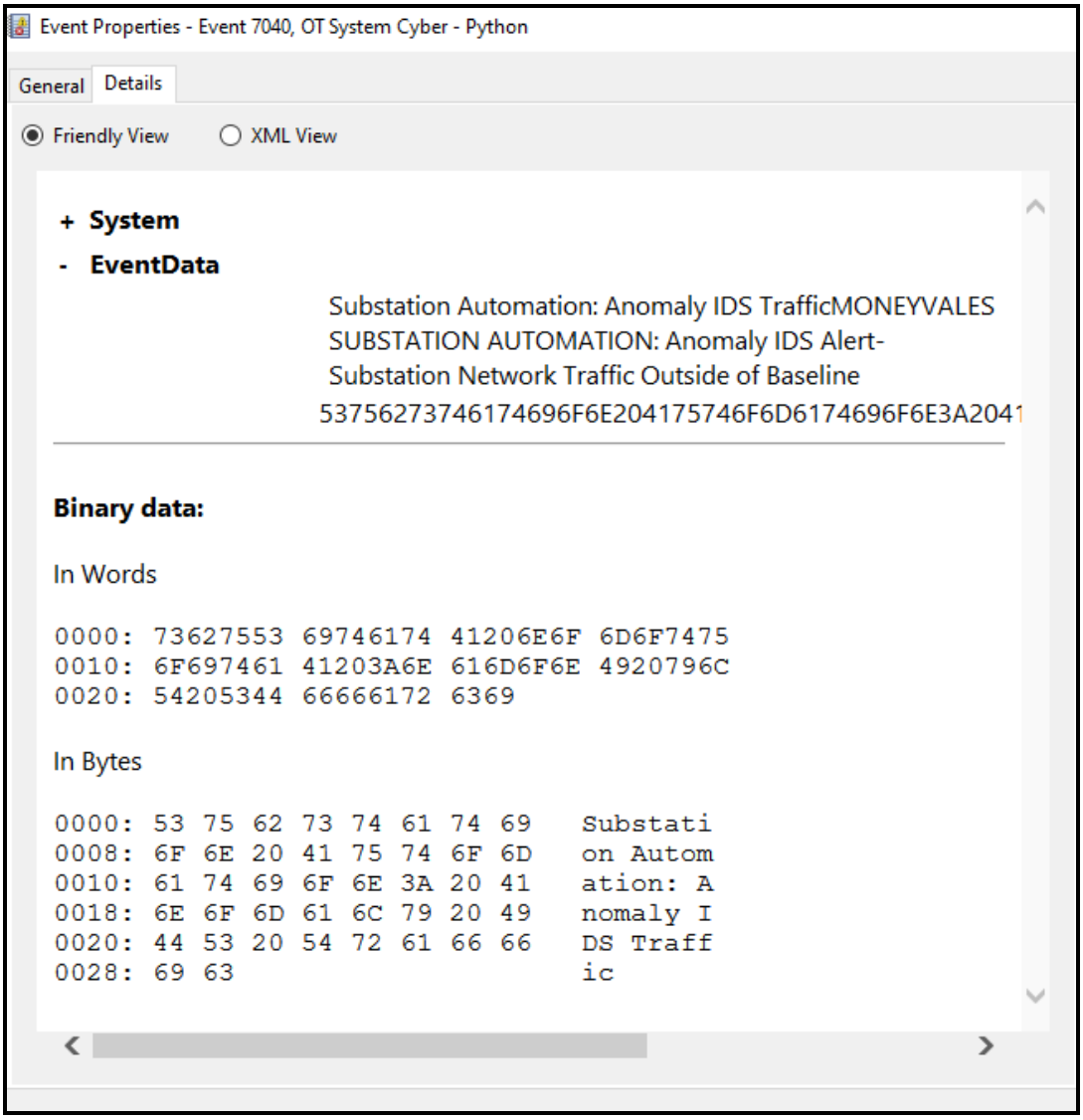


Rule No 6. Call's function ids_rule_6() in Main Program.

Rule No 6 - Email Alert: Moneyvales Substation Automation Traffic Anomaly Detected.

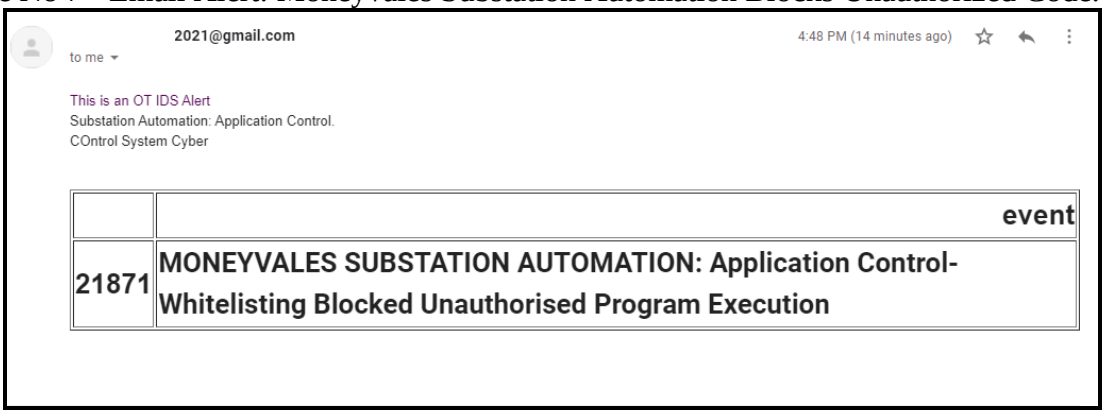


Rule No 6 - Microsoft System Log: OT Anomaly Based IDS Traffic Abnormal.

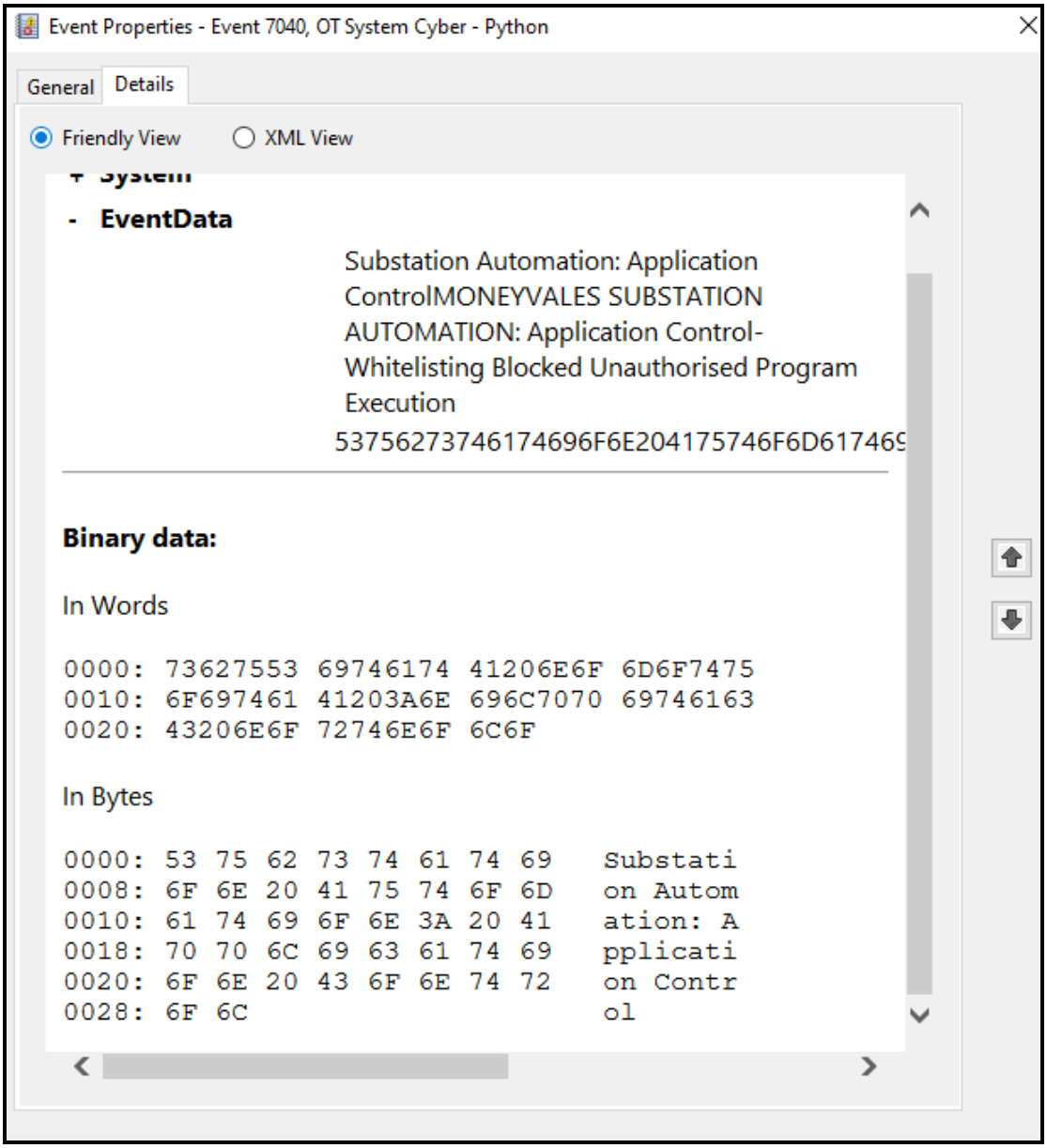


Rule No 7. Call's function ids_rule_7() in Main Program.

Rule No 7 - Email Alert: Moneyvales Substation Automation Blocks Unauthorized Code.



Rule No 7 - Microsoft System Log: Substation Automation System Application Control.



7 Running Program – ‘IDS Program Main’

The OT IDS application has been designed in a modular fashion. A main program calls the relevant program modules in a recursive manner. The Main Python Program is shown below.

```
import IDS_Program_Rule_1
import IDS_Program_Rule_2
import IDS_Program_Rule_3
import IDS_Program_Rule_4
import IDS_Program_Rule_5
import IDS_Program_Rule_6
import IDS_Program_Rule_7
import IDS_Program_Rule_8
import ssh_tunneller_ver_2
import IDS_Program_Clean_Up
import time


while True:

    ssh_tunneller_ver_2.ssh_import()
    time.sleep(5)
    IDS_Program_Rule_1.ids_rule_1()
    time.sleep(5)
    IDS_Program_Rule_2.ids_rule_2()
    time.sleep(5)
    IDS_Program_Rule_3.ids_rule_3()
    time.sleep(5)
    IDS_Program_Rule_4.ids_rule_4()
    time.sleep(5)
    IDS_Program_Rule_5.ids_rule_5()
    time.sleep(5)
    IDS_Program_Rule_6.ids_rule_6()
    time.sleep(5)
    IDS_Program_Rule_7.ids_rule_7()
    time.sleep(5)
    IDS_Program_Rule_8.ids_rule_8()
    time.sleep(5)
    IDS_Program_Clean_Up.ids_cleanup()
    time.sleep(5)
```

The ‘while True:’ function ensures the program runs continuously. It continuously selects OT database tables information and parses them against the 8 rules. At the beginning of each loop the data is acquired by the SSH Tunneller module and stored to a location where each rule is applied. For demonstration purposes each module is delayed by 5 seconds. The final module performs a tidy up of the data that has been generated by the preceding loop to ensure each loop executes from clean file data and no variables reside in memory from a previous loop execution.

References

Anaconda.org (2021) *pymysql: Anaconda.org*. Available at: https://anaconda.org/anaconda/pymysql (Accessed: 30 July 2021).

Anaconda (2021) ‘Anaconda | The World’s Most Popular Data Science Platform’. Available at: https://www.anaconda.com/ (Accessed: 30 July 2021).

Digital Ocean (2021) *How To Install MySQL on Ubuntu 20.04 | DigitalOcean*. Available at: https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04 (Accessed: 30 July 2021).

Geeksforgeeks.org (2021) *MYSQLdb Connection in Python - GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/mysqldb-connection-python/ (Accessed: 30 July 2021).

Lubuntu (2021) *Lubuntu 20.04.2 LTS (Focal Fossa)*. Available at: https://cdimage.ubuntu.com/lubuntu/releases/20.04/release/ (Accessed: 30 July 2021).
pypi.org (2021) *sshtunnel · PyPI*. Available at: https://pypi.org/project/sshtunnel/ (Accessed: 30 July 2021).

Stackoverflow (2021) *Conecting to MySQL in a remote server from python - Stack Overflow*. Available at: https://stackoverflow.com/questions/42726681/conecting-to-mysql-in-a-remote-server-from-python (Accessed: 30 July 2021).

StackOverFlow (2021a) *How to create Windows event log with Python - Stack Overflow*. Available at: https://stackoverflow.com/questions/64424417/how-to-create-windows-event-log-with-python (Accessed: 30 July 2021).

StackOverFlow (2021b) *oauth 2.0 - Sending email via gmail & python - Stack Overflow*. Available at: https://stackoverflow.com/questions/37201250/sending-email-via-gmail-python (Accessed: 30 July 2021).

StackOverFLow (2021a) *Access remote DB via ssh tunnel (Python 3) - Stack Overflow*. Available at: https://stackoverflow.com/questions/45213676/access-remote-db-via-ssh-tunnel-python-3 (Accessed: 30 July 2021).

StackOverFLow (2021b) *Import null and improperly formatted datetime values into datetime column MySQL - Stack Overflow*. Available at: https://stackoverflow.com/questions/24915498/import-null-and-improperly-formatted-datetime-values-into-datetime-column-mysql (Accessed: 30 July 2021).