System Design Document

Team: Lighthouse Solutions

Version: 1.0

Date: 10/31/2022

Client: Ben Drozdenko of the Naval Undersea Warfare Center Division Newport (NUWCDIVNPT)

Prepared by Johnny Driscoll, Sean Staton, Dylan Haughton, Brody Looney and Andy Howe

Abnormal Network Traffic Flow Dashboard Tool

System Design Document

Table of Contents

1 Introduction

This is a capstone project for Ben Drozdenko representing the Naval Undersea Warfare Center Division Newport (NUWCDIVNPT), in partial fulfillment of the Computer Science BS degree for the University of Maine. The purpose of this project is to give the client a viable means of analyzing traffic that moves around their network in order to protect the integrity of confidentiality of all data that flows through. The project serves to detect statistical anomalies and give minimalist, user-friendly reports.

1.1 Purpose of This Document

This document naturally serves to outline the mechanics of the coming project to be developed, in presenting architectural design and information regarding database and file usage. Below, one can find each of those evaluations, beginning with visual and theoretical analysis on the system's architecture, and ending with information regarding the proposed database, file usage, and requirements. This document is meant to be viewed and approved by the client, and thereafter, referred back to by the developers.

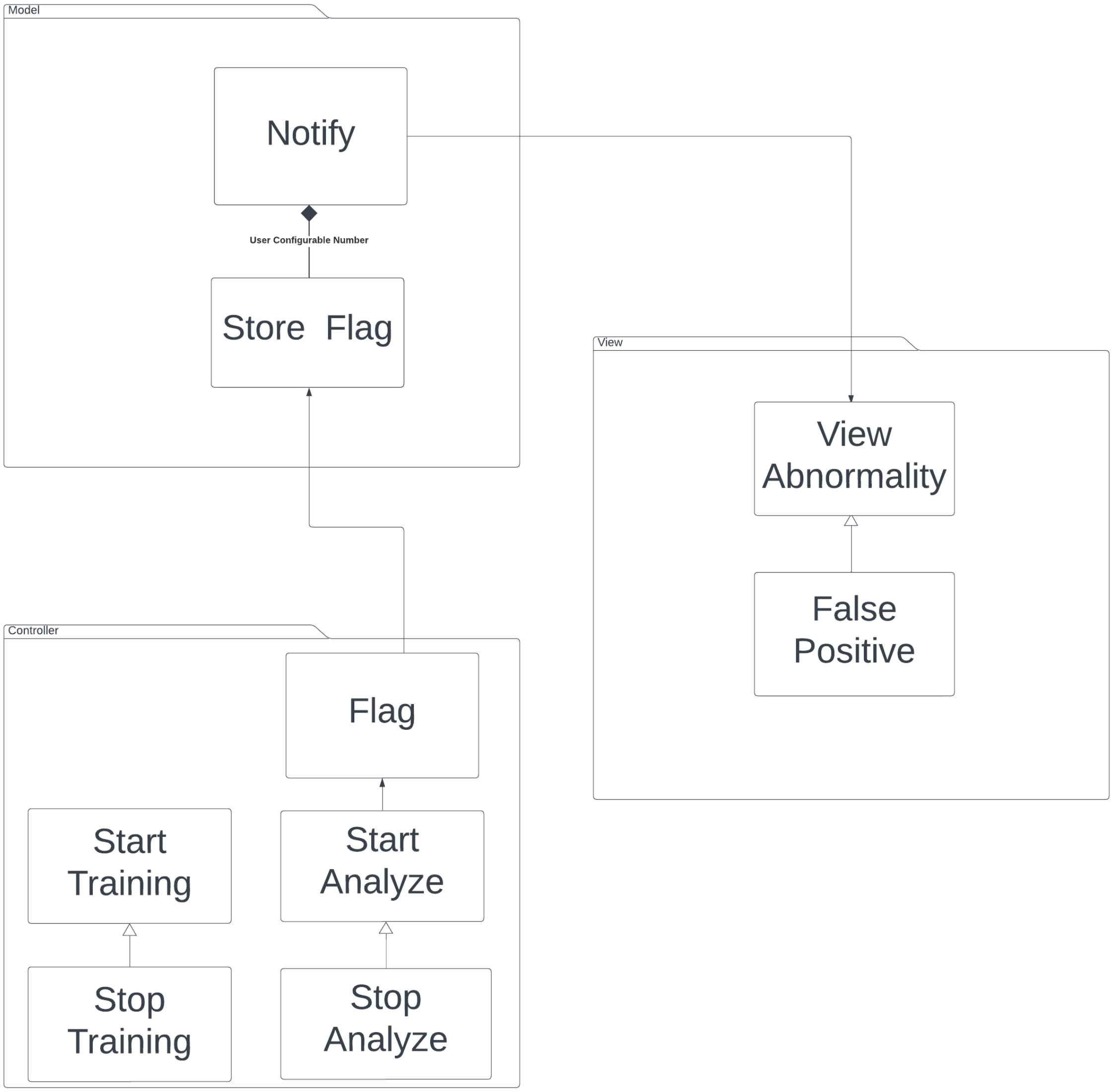
1.2 References

There are only three references to speak on currently. The first two of these are the original and revised SRS documents. These inform us on the necessary requirements of the project. The third reference document is a data set procured from the University of New South Wales, used as reference to the proposed file structure, type of information to be analyzed by the program, and the way in which data recording should be conducted.

2 System Architecture

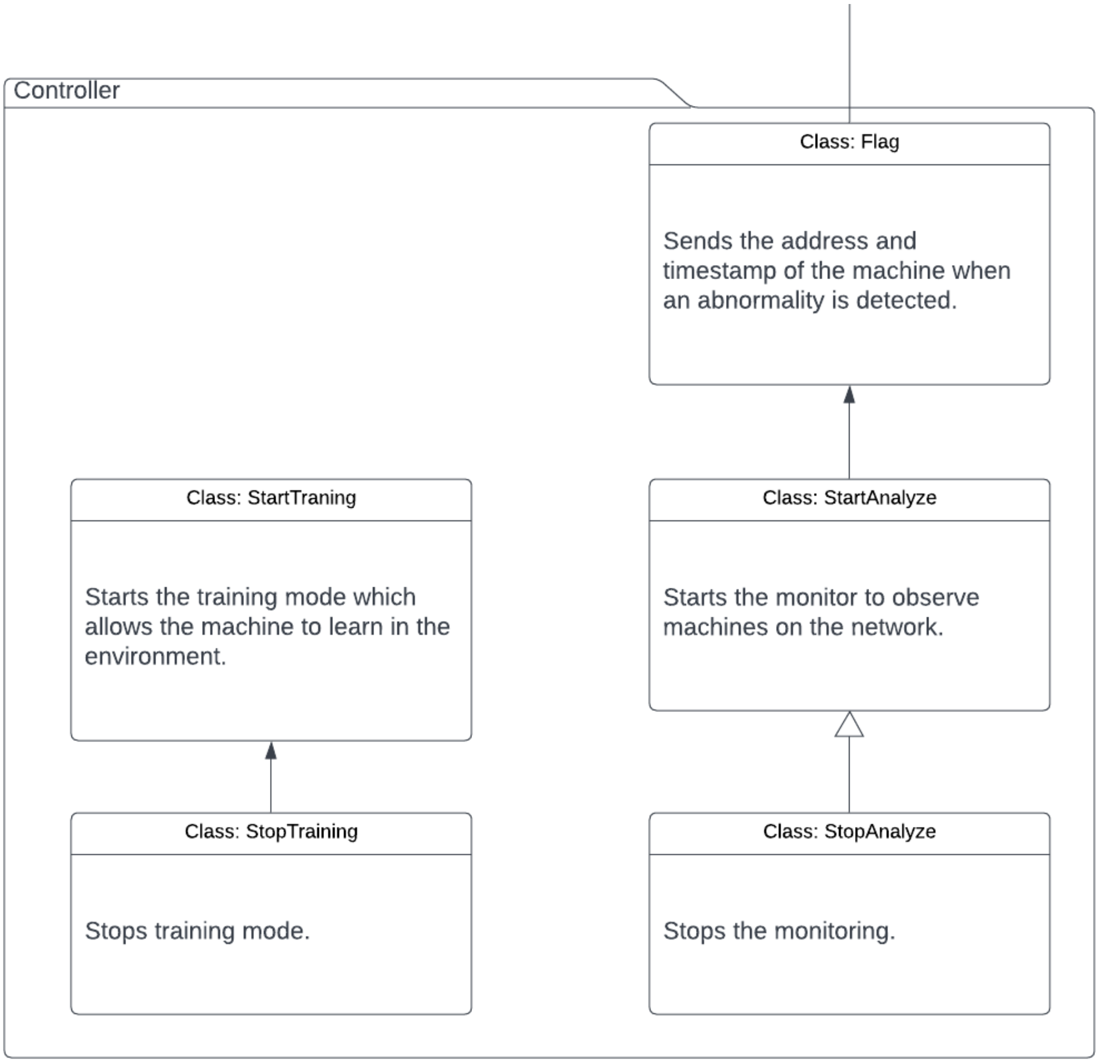
This section appropriately outlines, visually and systematically, the architectural design of the project. The first subsection will provide a detailed graphical design of the system architecture. The following section will deconstruct the different components of that graphic, and provide an explanation.

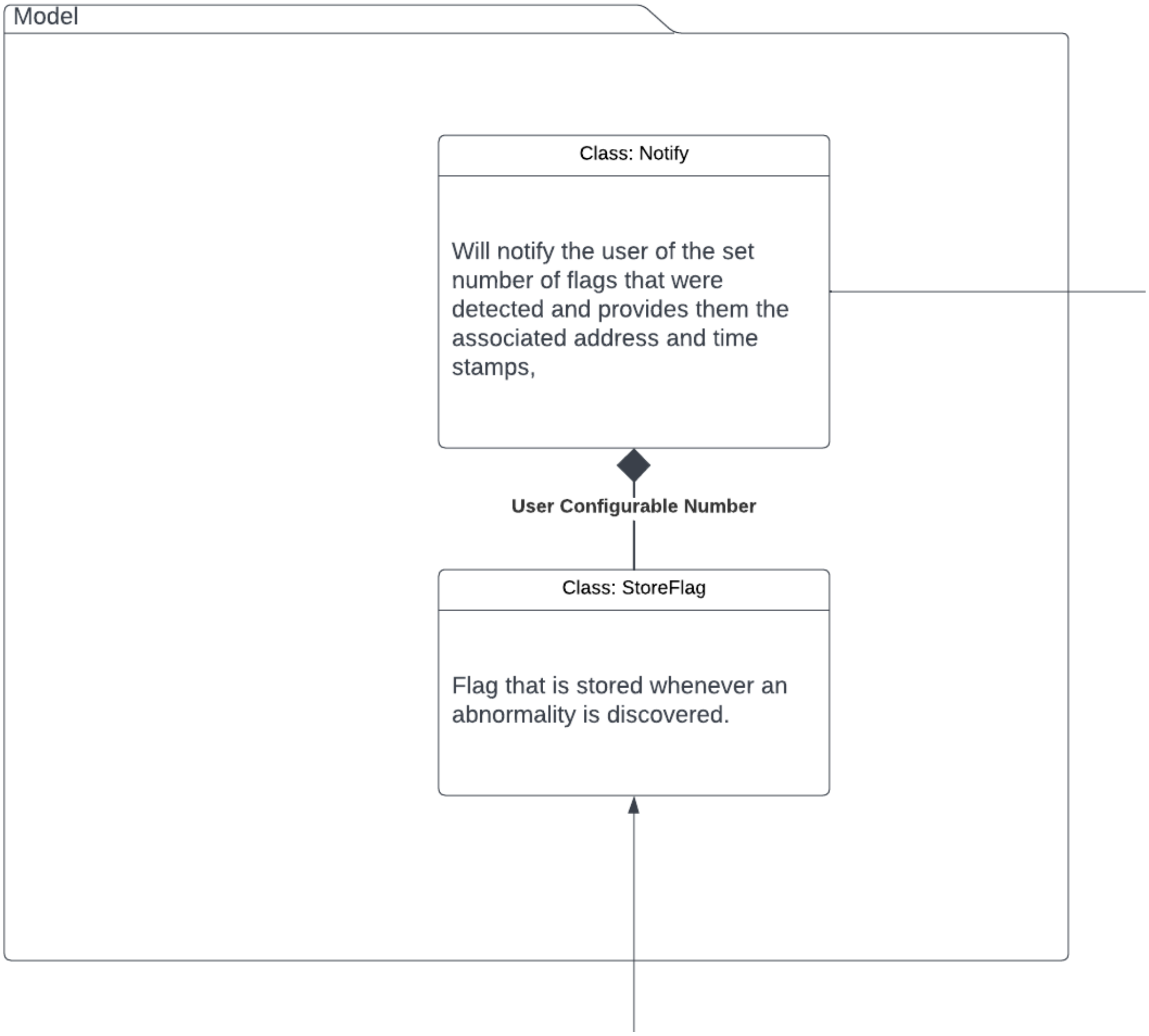
2.1 Architectural Design

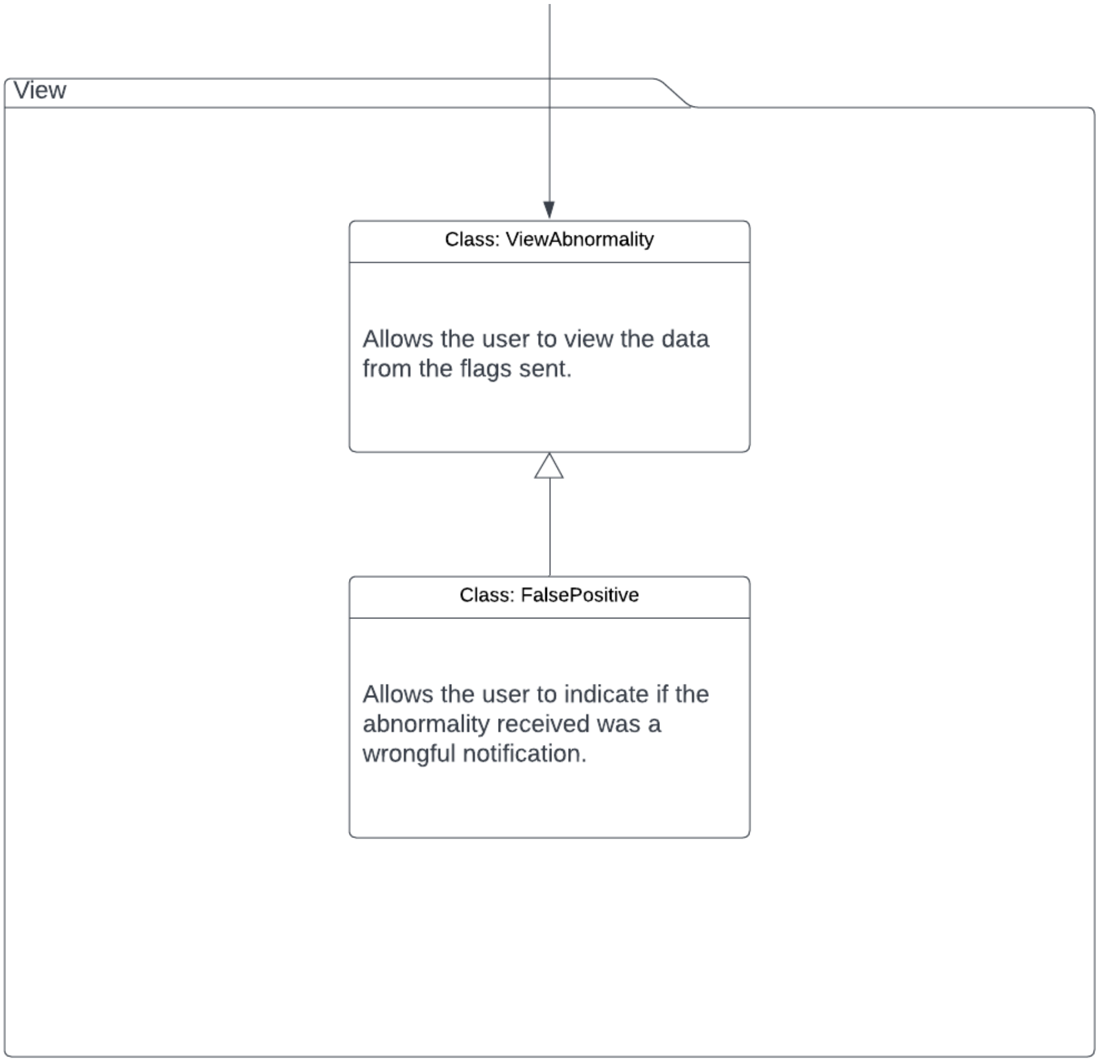


The system will be utilizing Zeek to analyze the traffic that goes through the network. Once the monitor detects an abnormality in the traffic. It will send a flag to the database, the flag will store the address of the machine that had this abnormality as well as a timestamp of when it was discovered. Once the database has three flags originating from the same address, it will send a message to the interface of the LAN Admin to inform them of the abnormalities. The message will provide the time stamps and the address of origination.

2.2 Decomposition Description


Controller
Class: Flag
Sends the address and timestamp of the machine when an abnormality is detected.
Class: StartTraning
Starts the training mode which allows the machine to learn in the environment.
Class: StartAnalyze
Starts the monitor to observe machines on the network.
Class: StopTraining
Stops training mode.
Class: StopAnalyze
Stops the monitoring.


Model
Class: Notify
Will notify the user of the set number of flags that were detected and provides them the associated address and time stamps,
User Configurable Number
Class: StoreFlag
Flag that is stored whenever an abnormality is discovered.

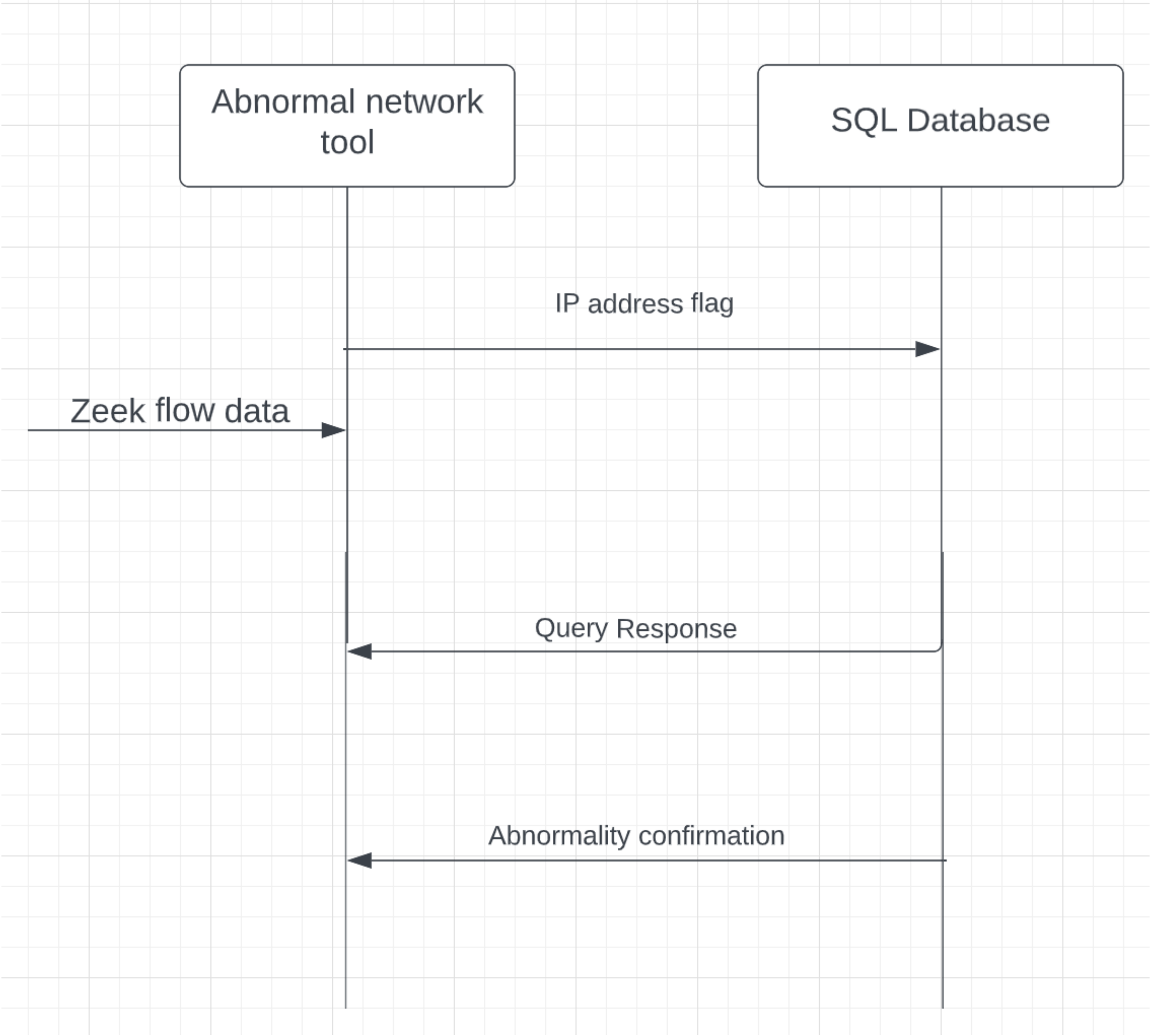

View
Class: ViewAbnormality
Allows the user to view the data from the flags sent.
Class: FalsePositive
Allows the user to indicate if the abnormality received was a wrongful notification.

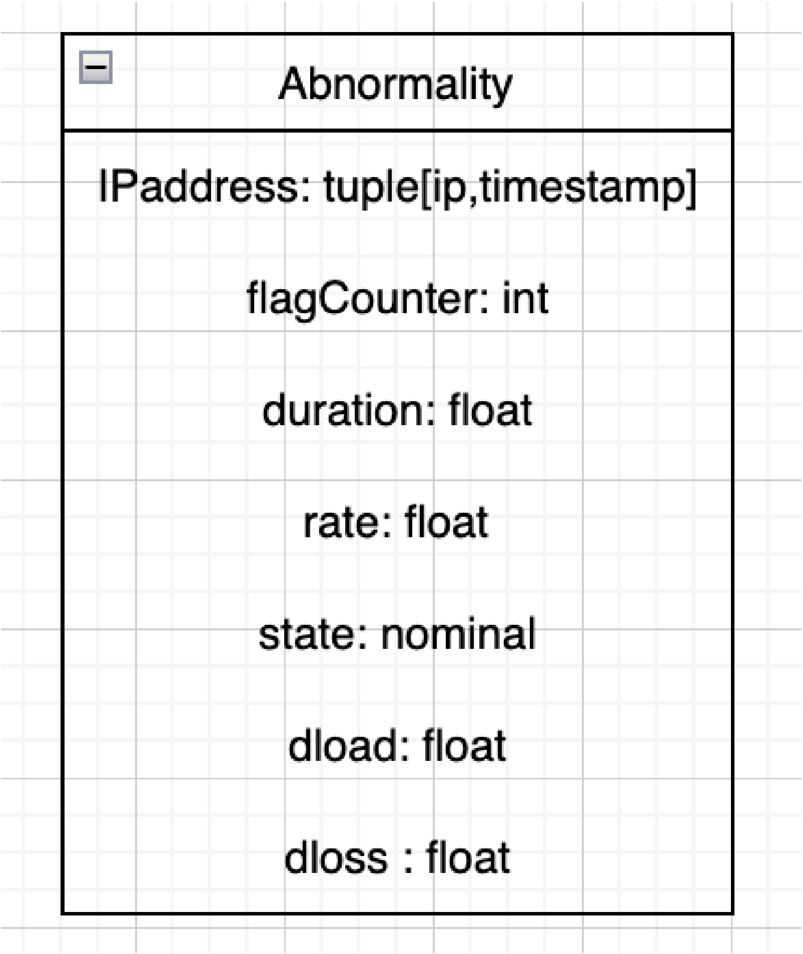
3 Persistent Data Design

This section describes, in detail, the nature of individual databases that are planned to be used, and the file interactions and structure to be implemented therein. The first of these subsections considers the use and evaluation of the database, while the second subsection describes the nature of various files used by the system, and the structure of that usage therein.

3.1 Database Descriptions (if you use a database)

Our application will communicate directly with the database without a middle layer involved. From the zeek flow network data, our tool will flag any abnormal data and send this directly to the database with the associated IP address. If the flagCounter rises to a certain number that our training algorithm will determine then an abnormality confirmation is sent back to the application. With this the LAN admin can view this information and take any necessary steps.





3.2 File Descriptions

The files that we are using in this system are .csv files that have 45 columns that each represent a different field within the file. The description of what each column represents is shown in the table below.

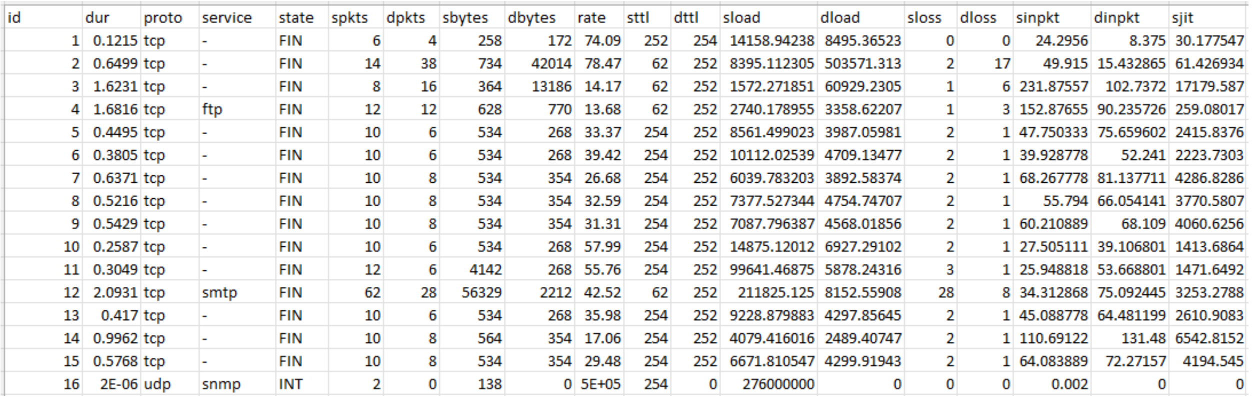
Name of Column	Data Type	Description
id	int	This serves as the identification number for the record.
dur	float	This is the duration of the record.
proto	nominal	This is the type of transaction protocol.
service	nominal	This is the service used to transfer the network traffic flow.
state	nominal	This indicates to the state and its dependent protocol.

spkts	int	Source to destination packet count.
dpkts	int	Destination to source packet count.
sbytes	int	Source to destination transaction bytes.
dbytes	int	Destination to source transaction bytes.
rate	float	Thsi is the rate at which data is being transmitted.
sttl	int	Source to destination time to live value.
dttl	int	Destination to source time to live value.
sload	float	Source bits per second.
dload	float	Destination bits per second.
sloss	int	Source packets retransmitted or dropped.
dloss	int	Destination packets retransmitted or dropped.
sinpkt	float	Source interpacket arrival time (mSec).
dinpkt	float	Destination interpacket arrival time (mSec).
sjit	float	Source jitter (mSec).
djit	float	Destination jitter (mSec).
swin	int	Source TCP window advertisement value.
stcpb	long	Source TCP base sequence number.
dtcpb	long	Destination TCP base sequence number.
dwin	int	Destination TCP window advertisement value.
tcprtt	float	TCP connection setup round-trip time, the sum of ’synack’ and ’ackdat’.
synack	float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
ackdat	float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
smean	int	Mean of the packet size transmitted by the src.

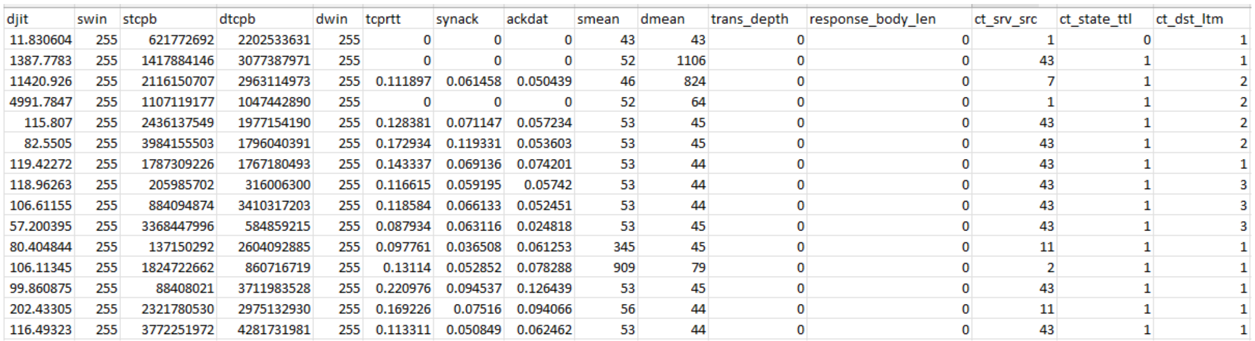
dmean	int	Mean of the packet size transmitted by the dst.
trans_depth	int	Represents the pipelined depth into the connection of http request/response transaction.
response_body_len	int	Actual uncompressed content size of the data transferred from the server’s http service.
ct_srv_src	int	Number of connections that contain the same service and source address in 100 connections according to the last time.
ct_state_ttl	int	Number for each state according to specific range of values for source/destination time to live.
ct_dst_ltm	int	Number of connections of the same destination address in 100 connections according to the last time.
ct_src_dport_ltm	int	Number of connections of the same source address and the destination port in 100 connections according to the last time.
ct_dst_sport_ltm	int	Number of connections of the same destination address and the source port in 100 connections according to the last time.
ct_dst_src_ltm	int	Number of connections of the same source and the destination address in in 100 connections according to the last time.
is_ftp_login	binary	If the ftp session is accessed by user and password then 1 else 0.
ct_ftp_cmd	int	Number of flows that has a command in ftp session.
ct_flw_http_mthd	int	Number. of flows that has methods such as Get and Post in http service.
ct_src_ltm	int	Number of connections of the same destination address in 100 connections according to the last time.
ct_srv_dst	int	Number of connections that contain the same service and destination address in 100 connections according to the last time.
is_sm_ips_ports	binary	If source and destination IP addresses equal and port numbers equal then, this variable takes value 1 else 0.

attack_cat	nominal	This shows if the piece of network traffic flow data is normal or if it is a type of attack. There are nine types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.
label	binary	This labels whether the record was normal or an attack. 0 for normal and 1 for attack records.

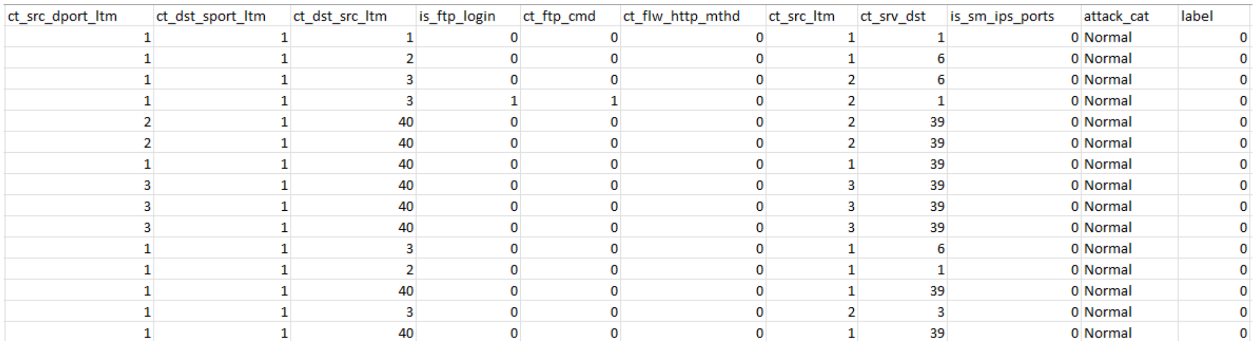
An example of this file is shown below. The file had to be broken into three screenshots since the file is very large. This file shows 16 rows within the file but in reality these files are around 100,000+ rows long.



id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	sinpkt	dinpkt	sjit
1	0.1215	tcp	-	FIN	6	4	258	172	74.09	252	254	14158.94238	8495.36523	0	0	24.2956	8.375	30.177547
2	0.6499	tcp	-	FIN	14	38	734	42014	78.47	62	252	8395.112305	503571.313	2	17	49.915	15.432865	61.426934
3	1.6231	tcp	-	FIN	8	16	364	13186	14.17	62	252	1572.271851	60929.2305	1	6	231.87557	102.7372	17179.587
4	1.6816	tcp	ftp	FIN	12	12	628	770	13.68	62	252	2740.178955	3358.62207	1	3	152.87655	90.235726	259.08017
5	0.4495	tcp	-	FIN	10	6	534	268	33.37	254	252	8561.499023	3987.05981	2	1	47.750333	75.659602	2415.8376
6	0.3805	tcp	-	FIN	10	6	534	268	39.42	254	252	10112.02539	4709.13477	2	1	39.928778	52.241	2223.7303
7	0.6371	tcp	-	FIN	10	8	534	354	26.68	254	252	6039.783203	3892.58374	2	1	68.267778	81.137711	4286.8286
8	0.5216	tcp	-	FIN	10	8	534	354	32.59	254	252	7377.527344	4754.74707	2	1	55.794	66.054141	3770.5807
9	0.5429	tcp	-	FIN	10	8	534	354	31.31	254	252	7087.796387	4568.01856	2	1	60.210889	68.109	4060.6256
10	0.2587	tcp	-	FIN	10	6	534	268	57.99	254	252	14875.12012	6927.29102	2	1	27.505111	39.106801	1413.6864
11	0.3049	tcp	-	FIN	12	6	4142	268	55.76	254	252	99641.46875	5878.24316	3	1	25.948818	53.668801	1471.6492
12	2.0931	tcp	smtp	FIN	62	28	56329	2212	42.52	62	252	211825.125	8152.55908	28	8	34.312868	75.092445	3253.2788
13	0.417	tcp	-	FIN	10	6	534	268	35.98	254	252	9228.879883	4297.85645	2	1	45.088778	64.481199	2610.9083
14	0.9962	tcp	-	FIN	10	8	564	354	17.06	254	252	4079.416016	2489.40747	2	1	110.69122	131.48	6542.8152
15	0.5768	tcp	-	FIN	10	8	534	354	29.48	254	252	6671.810547	4299.91943	2	1	64.083889	72.27157	4194.545
16	2E-06	udp	snmp	INT	2	0	138	0	5E+05	254	0	276000000	0	0	0	0.002	0	0



djit	swin	stcpb	dtcpb	dwin	tcprtt	synack	ackdat	smean	dmean	trans_depth	response_body_len	ct_srv_src	ct_state_ttl	ct_dst_ltm
11.830604	255	621772692	2202533631	255	0	0	0	43	43	0	0	1	0	1
1387.7783	255	1417884146	3077387971	255	0	0	0	52	1106	0	0	43	1	1
11420.926	255	2116150707	2963114973	255	0.111897	0.061458	0.050439	46	824	0	0	7	1	2
4991.7847	255	1107119177	1047442890	255	0	0	0	52	64	0	0	1	1	2
115.807	255	2436137549	1977154190	255	0.128381	0.071147	0.057234	53	45	0	0	43	1	2
82.5505	255	3984155503	1796040391	255	0.172934	0.119331	0.053603	53	45	0	0	43	1	2
119.42272	255	1787309226	1767180493	255	0.143337	0.069136	0.074201	53	44	0	0	43	1	1
118.96263	255	205985702	316006300	255	0.116615	0.059195	0.05742	53	44	0	0	43	1	3
106.61155	255	884094874	3410317203	255	0.118584	0.066133	0.052451	53	44	0	0	43	1	3
57.200395	255	3368447996	584859215	255	0.087934	0.063116	0.024818	53	45	0	0	43	1	3
80.404844	255	137150292	2604092885	255	0.097761	0.036508	0.061253	345	45	0	0	11	1	1
106.11345	255	1824722662	860716719	255	0.13114	0.052852	0.078288	909	79	0	0	2	1	1
99.860875	255	88408021	3711983528	255	0.220976	0.094537	0.126439	53	45	0	0	43	1	1
202.43305	255	2321780530	2975132930	255	0.169226	0.07516	0.094066	56	44	0	0	11	1	1
116.49323	255	3772251972	4281731981	255	0.113311	0.050849	0.062462	53	44	0	0	43	1	1



ct_src_dport_ltm	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd	ct_flw_http_mthd	ct_src_ltm	ct_srv_dst	is_sm_ips_ports	attack_cat	label
1	1	1	0	0	0	1	1	0	Normal	0
1	1	2	0	0	0	1	6	0	Normal	0
1	1	3	0	0	0	2	6	0	Normal	0
1	1	3	1	1	0	2	1	0	Normal	0
2	1	40	0	0	0	2	39	0	Normal	0
2	1	40	0	0	0	2	39	0	Normal	0
1	1	40	0	0	0	1	39	0	Normal	0
3	1	40	0	0	0	3	39	0	Normal	0
3	1	40	0	0	0	3	39	0	Normal	0
3	1	40	0	0	0	3	39	0	Normal	0
1	1	3	0	0	0	1	6	0	Normal	0
1	1	2	0	0	0	1	1	0	Normal	0
1	1	40	0	0	0	1	39	0	Normal	0
1	1	3	0	0	0	2	3	0	Normal	0
1	1	40	0	0	0	1	39	0	Normal	0

4 Requirements Matrix

This final section considers the requirements, as outlined in the SRS, and how they pertain to system components outlined in the first section of this paper. The table below shows this by requirement number, relating each to a specific component.

Requirement Number	**Name**	**System component used**
1	User interface	Database,DisplayData()
2	No bloat elements	GUI
3	Begin capture	Database, sFlow, BeginCapture()
4	End capture	Database, sFlow, EndCapture()
5	Collect network data	Database, CollectWhenDiscovered()
6	Display abnormal data	Database, DisplayData()
7	Allow user to identify abnormal data	Database, DisplayData(), AppendData(),DeleteData()
8	Alert LAN admin	Database, DisplayData(), SendMessage()
9	Individual abnormal data collection	Database, DisplayData(),SearchIP()
10	Training from a file	ReadData()
11	Training from a capture	ReadData()
12	Switch between ‘capture’ and ‘training’ mode	SwitchMode()

Appendix A – Agreement Between Customer and Contractor

This document states the system design for the *Abnormal Network Traffic Flow Dashboard Tool* contracted by Ben Drozdenko and the Naval Undersea Warfare Center Division Newport (NUWCDIVNPT). Both the customer and the development team agree that this document clearly states the system architecture, persistent data design, and requirements matrix for this project.

In the event that there are future changes made to this document the development team and customer will discuss the changes during our bi-weekly check-in meetings. If everyone agrees on the changes, we will document the change below and sign-off showing that the amendment has been approved.

Brody Looney	Brody Looney	11/8/2022
Typed Name	Signature	Date
Dylan Haughton	Dylan Haughton	11/8/2022
Typed Name	Signature	Date
Sean Staton	Sean Staton	11/8/2022
Typed Name	Signature	Date
Johnathan Driscoll	Johnathan Driscoll	11/8/2022
Typed Name	Signature	Date
Andrew Howe	Andrew Howe	11/8/2022
Typed Name	Signature	Date
Benjamin Drozdenko	DROZDENKO.BENJAMIN.MARTIN.1364363709 Digitally signed by DROZDENKO.BENJAMIN.MARTIN.1364363709 Date: 2022.11.10 10:51:27 -05'00'	11/10/2022
Typed Name	Signature	Date

Customer Comments:

Appendix B – Team Review Sign-off

Andy Howe, Dylan Haughton, Johnny Driscoll, Sean Staton, and Brody Looney have all reviewed this document. We all have agreed on the content that listed in this document is accurate and complete. We all agree that the format follows the correct structure. The signatures below prove the information stated previously is correct.

Brody Looney	Brody Looney	11/8/2022
Typed Name	Signature	Date
Dylan Haughton	Dylan Haughton	11/8/2022
Typed Name	Signature	Date
Sean Staton	Sean Staton	11/8/2022
Typed Name	Signature	Date
Johnathan Driscoll	Johnathan Driscoll	11/8/2022
Typed Name	Signature	Date
Andrew Howe	Andrew Howe	11/8/2022
Typed Name	Signature	Date

Comments:

Appendix C – Document Contributions

Group Member	Sections Worked On	Contribution Percentage
Sean Staton	Introduction, introductions for each section	20%
Dylan Haughton	System Architecture	20%
Andy Howe	Requirements Matrix	20%
Johnny Driscoll	Persistent Data Design	20%
Brody Looney	SRS Review, File Descriptions, Appendix A, Appendix B, and Appendix C	20%