Final Year Project Report
on

Digital Forensics on SCADA System

Submitted
in partial fulfillment of
the requirement of the degree for the Degree of Bachelor Of Technology

by

Ayush Shah (171090010)
Gurkirat Nagpal (181091902)
Atharva Marathe (171090053)
Sarvesh Yenarkar (171090063)

Under the Guidance of

Dr. Faruk S. Kazi

DEPARTMENT OF ELECTRICAL ENGINEERING
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
(An Autonomous Institute Affiliated to Mumbai University)
(Central Technological Institute, Maharashtra State)
Matunga, MUMBAI - 400019

A.Y. 2020-2021

STATEMENT OF CANDIDATE

I state that work embodied in Stage-I of this Project entitled **"Digital Forensics on SCADA System"** form my own contribution of work under the guidance of Dr. Faruk S. Kazi at the Department of Electrical Engineering, Veermata Jijabai Technological Institute, Mumbai. The report reflects the work done during the period of Stage-I.

Ayush Shah
Roll No.- 171090010
Date:
Place: VJTI, Mumbai

Gurkirat Nagpal
Roll No.- 181091902
Date:
Place: VJTI, Mumbai

Atharva Marathe
Roll No.- 171090053
Date:
Place: VJTI, Mumbai

Sarvesh Yenarkar
Roll No.- 171090063
Date:
Place: VJTI, Mumbai

APPROVAL SHEET

The credit report ***"Digital Forensics on SCADA System"*** by Ayush Shah, Gurkirat Nagpal, Atharva Marathe and Sarvesh Yenarkar is found to be satisfactory and is approved for the Degree of Bachelor of Technology.

Dr. F.S. Kazi
Ph.D. Supervisor

Examiner

Examiner

Examiner

Place: *Veermata Jijabai Technological Institute, Mumbai.*
Date: / /2021

CERTIFICATE

This is to certify that Ayush Shah (171090010), Gurkirat Nagpal(181091902), Atharva Marathe(171090053) and Sarvesh Yenarkar(171090063) a student of B. Tech (Electronics and Telecommuncation), Veermata Jijabai Technological Institute, Mumbai has successfully completed the Stage-I of project titled ***"Digital Forensics on SCADA System"*** under the guidance of Dr. Faruk Kazi.

Dr. F. S. Kazi

Supervisor

Dr. F. S. Kazi

Head of Electrical Engineering Department

Place: *Veermata Jijabai Technological Institute, Mumbai.*
Date: / /2021

Acknowledgement

We would like to express our gratitude to everyone whose support, guidance and cooperation has been a source pf unparalleled support during the project. We would like to thank our mentor **Dr.Faruk Kazi** for guiding us throughout the project. We are also thankful to **COE CNDS, VJTI** for providing us with such a wonderful platform to explore new avenues and practically implement theoretical concepts into real life problems.

We would also like to thank PhD mentor Mrs. Pranita Binnar, Shah Rukh Khan and all other members of the COE lab for directly or indirectly helping me for the completion of the project and the resources provided.

We would once again like to show our deepest gratitude to each and every person who supported us, encouraged us or contributed to helping us in the project in any way.

Ayush Shah
Roll No.- 171090010

Gurkirat Nagpal
Roll No.- 181091902

Atharva Marathe
Roll No.- 171090053

Sarvesh Yenarkar
Roll No.- 171090063

Contents

List of Figures vii

List of Tables viii

List of Abbreviations ix

1 SYNOPSIS 1
- 1.1 Abstract . . . 1
- 1.2 Technical Keywords . . . 1
- 1.3 Project Timeline . . . 1

2 INTRODUCTION 2
- 2.1 SCADA components . . . 2
- 2.2 Logs in cyber security . . . 2
- 2.3 Steps for Memory Forensics . . . 3
- 2.4 Machine Learning Components . . . 3
  - 2.4.1 Correlation . . . 3
  - 2.4.2 Normalized and Standardized Scalar . . . 3
  - 2.4.3 Naive Bayes Algorithm . . . 4
  - 2.4.4 Decision Tree . . . 4
  - 2.4.5 Random Forest Classifier . . . 4
  - 2.4.6 K-Means Algorithm . . . 4
  - 2.4.7 Linear Discriminator . . . 4
  - 2.4.8 Linear SVC Mode . . . 4
- 2.5 Memory forensics and Memory Dump . . . 5
- 2.6 Volatility in Memory Forensics . . . 5
- 2.7 Memory Artifacts . . . 5
- 2.8 Normal and Abnormal Traffic . . . 5
- 2.9 Intrusion Prevention systems in IT and OT networks . . . 5
- 2.10 Motivation . . . 7

3 LITERATURE SURVEY 8
- 3.1 Gap of Literature Survey . . . 11
- 3.2 Summary of Literature Survey . . . 11

4 Proposed Methodology 13
- 4.1 Problem Statement . . . 13
- 4.2 Objective and Goals . . . 13
- 4.3 Proposed System . . . 14
  - 4.3.1 Network Forensics in ICS Environment . . . 14
  - 4.3.2 Analysing PCAP files: . . . 17
  - 4.3.3 Forensic Analysis of Memory Dump . . . 21
  - 4.3.4 Machine Learning Algorithms . . . 31

4.4 Implementation 38
4.4.1 Hardware and Software Requirements 38

5 CONCLUSION AND FUTURE WORK **40**
5.1 CONCLUSION 40
5.2 FUTURE WORK 41

Bibliography **42**

List of Figures

2.1 Purdue Model for Forensics over SCADA architecture - Source:[26] . . . 6
2.2 Deploying an intrusion prevention system - Source:[26] . . . 7

4.1 Ethernet Endpoints obtained . . . 17
4.2 Protocol Hierarchy . . . 18
4.3 IP endpoints . . . 18
4.4 scanning attacks . . . 19
4.5 opcodes of packets . . . 20
4.6 Testbed Architecture - Source:[4] . . . 21
4.7 Flow Diagram of forensic analysis of the Memory Dump . . . 22
4.8 Hash Value for evidence memory dump . . . 23
4.9 Extraction of memory dump using FTK Imager . . . 24
4.10 Memory dump viewed in FTK imager . . . 24
4.11 Acquired memory dump . . . 25
4.12 Verified Hash Value Integrity . . . 25
4.13 Memory Dump Image Profile . . . 26
4.14 Running Process . . . 26
4.15 Hidden Processes . . . 27
4.16 Current and Parent Processes . . . 27
4.17 Active IP Addresses and Ports . . . 28
4.18 mDNS Responser Memdump . . . 28
4.19 Malfind and VirusTotal . . . 30
4.20 Testbed Architecture - Source:[17] . . . 31
4.21 Instance Frequency for Source IP's . . . 32
4.22 Instance Frequency for Destination IP's . . . 32
4.23 Protocol frequency in the dataset . . . 33
4.24 Steps in Pre-processing - Source:[12] . . . 34
4.25 Correlation generalised equation - Source:Science Direct . . . 34
4.26 Accuracy Comparison for algorithms . . . 37

List of Tables

3.1 Popular Techniques of Anomaly Detection 12

4.1 Commands used for Analysis 29
4.2 Binary Classification Outputs 35
4.3 Multi class Classification Outputs 36
4.4 Output of the algorithms 36
4.5 Hardware Requirements 39

List of Abbreviations

ML	Machine Learning
OT	Operational Technology
SVC	Support Vector Classifier
IPS	Intrusion Prevention System
IDS	Intrusion Detection System
SF	SCAFA Forensics
NB	Naive Bayes
KMM	K-Means Algorithm
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
RF	Random Forest Classifier
DT	Decision Tree
WUSTL	Washington University in St.Louis
ML	Maximum-Likelihood
MD	Memory Dump
EM	Expectation-Maximization
SCADA	Supervisory control and data acquisition

Chapter 1

SYNOPSIS

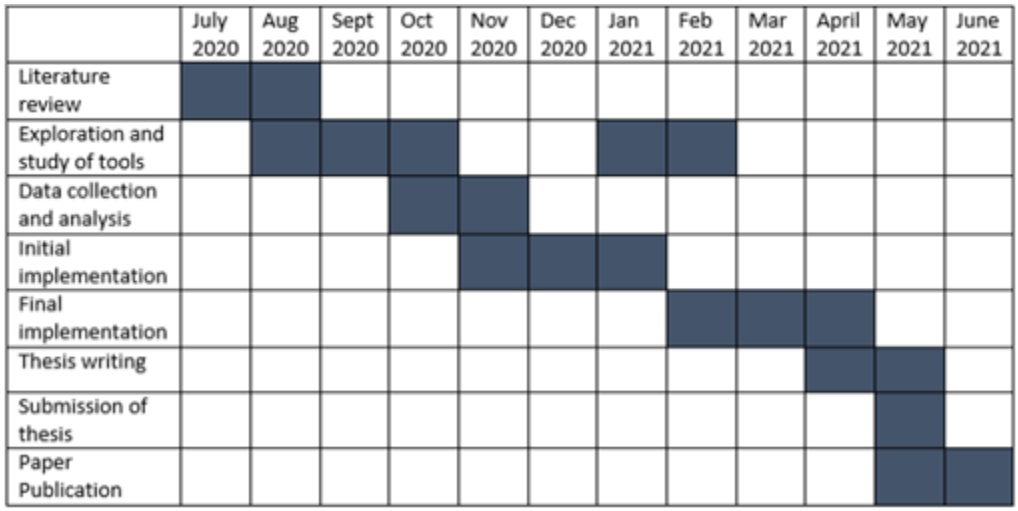
1.1 Abstract

Kudankulam power plant: In recent years India's largest Kudankulam nuclear power plant has suffered from Cyber-attack. It may cause cyber warfare. Energy sectors are rely upon SCADA for its safe operation through monitoring, automation and control. Cyber-attack within critical infrastructure can have devastating consequences affecting human life, the environment and the economy. Therefore, it is vital that a forensic investigation takes place to provide remediation, understanding and to help in the design of more secure system. SCADA system forensic analysis and investigation is an essential process the cyber security life cycle that not only helps to identify the cause of the incident and those responsible but to help develop and design more secure systems of the future. In this report we have pursued the forensic analysis of the case study helping us lead into the network forensic analysis for SCADA systems and finally K-Nearest Neighbours algorithm have been used over the dataset to predict malware with 99.98% accuracy.

1.2 Technical Keywords

Memory Forensics,SCADA Forensics, Network Forensic, Naive Bayes, Decision Trees, Machine Learning, K-Means Clustering,Malware Analysis

1.3 Project Timeline



Chapter 2

INTRODUCTION

Cybersecurity is collections of propositions and attempts to secure Network oriented systems containing i/o devices, software, and data from anomalous behavioural attacks. Because of the dependency on computers that store and output a heavy capacity of private and important data about people. Cyber anomalous behavioural is now an important issue which needs to be resolved as security attacks could be hazardous to the global economy. Industries/Companies transmit critical information across networks and to native i/o components in managing business, even a casual user/person inputs and outputs a lot of confidential data, thus cybersecurity describes to secure that data and the system used to input/output. Cybersecurity has three goals namely, confidentiality, integrity, availability of information and to preserve it.

2.1 SCADA components

Rather then manually going for data collection, we deploy SCADA to get that information ans sent it to your station's main control station. They send digital data instantaneously and also gets the logs of received information for analysing later. The most important components of SCADA when data acquisition is necessary are:

- Instrments of field: This refers to the range of monitoring and transmission devices needed on the premises of SCADA usage. They will detect any changes that happen in the different components and then send it to the Control Station in the center.
- Controller(RTU or PLC) : they will assemble and simplify the data received from the instruments to send it for analysis by the human or machine. They depend on mincroprocessors for converting the data into information. They work on local subsystems, giving the user a chance to manipulate the instrumentation directly.
- Human/Machine work interface: They are the main systems which allow the user to look over the acquitted information by SCADA. They serve as the main processing unit, allowing the regulation of the n/w as required. Another usage is to get the knowledge of data received by GUI.
- Network Connection: Main reliance of the system is on integration of the network connect within the operating field. This is completed using wired or non-wired components. They help us connect scattered components to the central point.

2.2 Logs in cyber security

Logging in cybersecurity is like bookkeeping in accounts, without which no further processing can take place. It's the interpretation of host or network generated events that are reviewed for the purpose of cyber security in industry. When the log collector is analysed it displays a chronological descriptive sequence of events that are necessary to gain insights of the communication traffic of the network. Logs could mend themselves according to the business requirements because they can be of various types fitting to each need. Some of

them are – system logs, technical logs, cybersecurity logs, networking logs. For our field of research the gem of processing are the cyber security based logs which are further classified as Data loss protection (DLP) logs, Malware protection software logs, Network intrusion detection system (NIDS) logs, Network intrusion prevention system (NIPS) logs. The important events (event logging) are captured and the cyber security analyst tend to extract information only on the basis of these automated software-based monitoring feature. SCADA(Supervisory Control and Data Acquisition System) Systems which are cornerstone of industrial plant like power plant,water plant or even nuclear plant.The SCADA System is responsible for controlling and mentoring the communications between network structures like PLCs.To prevent any anomalous behavior or attack on on SCADA System,Forensics investigation is in need and these investigation of unkown attack is stated as SCADA Forensics. It is forensic analysis of volatile memory or data in a memory dump and log analysis.Log analysis is important for acquiring information on the specifications and whereabouts of the attack. Logs for Windows Operating System are divided into Application logs,security logs and system logs.In case of Linux Operating System following commands states their respective logs type. /var/log/message: used to retrieve general message, /var/log/auth.log: Authentication logs, /var/log/kern.log: Kernel logs, /var/log/boot.log : System boot log, /var/log/utmp or /var/log/wtmp : Login records file A memory dump is the captured data of a computer at a particular instant.This captured data may contain useful information of forensics data like system's condition before an attack or a crash.This dump compromises of RAM information for locating the cause of the particular attack. Volatile memory is very important to understand the system's state and gives insights on the cause of the attack.

2.3 Steps for Memory Forensics

Memory Forensics comprises mainly of 5 steps stated as follows.

- Policy and Procedure Development : The information in memory dump is very delicate and must be protected and respected or else the memory dump may be compromised. Law Agencies are responsible for designing appropriate protocols for such specific scenarios in need of forensics.
- Evidence Assessment : This step involves Investigator Processing the evidence and then analyzing the whereabouts of the incident which leads to concrete results about the source of anomalous attack.
- Evidence Acquisition : It is a tool used for collecting the data/memory like DumpIt for Windows or AVML used for Linux.Volatility is the best example for Evidence analysis.
- Evidence Examination :For Efficient investigation, investigator examines the valuable data by organizing and storing the data to the specific database.
- Documenting and Reporting : After analysis, the information of hardware/software specification of SCADA system with all the methodologies used in the investigation is Documented.

2.4 Machine Learning Components

2.4.1 Correlation

It is a parameter or coefficient that is responsible of reading degree of difference or strengths of variables to know the relation they posses. The correlation factor ranges from -1 to 1. When the value of correlation is 1 its called perfect positive correlation. It is independent of what parameter or a variable caused those variables. The Formula is given as :

2.4.2 Normalized and Standardized Scalar

These are the scaling measures responsible for Data transformation. Normalization is used for re scaling the respective variable range into [0,1]. While Standardization is used for scaling the range by using the mean as 0 and the standard deviation to be 1. They are usually used for scaling the features of classifiers in Machine learning programming like One class SVM Model.Standardization is used for less standard deviation in features while Normalization is used for obtaining bi linear value.

2.4.3 Naive Bayes Algorithm

It is a classifier based on Bayes' Theorem by considering features of network to be independent with respect to each other. It assumes that the features of a class is not related to any other. For example, a vegetable may be a cabbage when its characteristic features are set to be green, round, and about 40-70 cms in radius. It seems these features to depend on the other still they independently relate to the feature of this vegetable to be a cabbage and hence it is termed by Naive word. The model recommends huge data-sets.

2.4.4 Decision Tree

It is a structure similar to hierarchical tree where every node contains an attribute or related feature/outcome while every leaf node has a class labels. It is such that its learning is done by dividing the source set into subsets based on the attributes of test. They are termed as 'non-parametric supervised learning'. This learning is done in a recursive manner and is done when subset at a node has the same value of the target variables.

2.4.5 Random Forest Classifier

As the name suggests, Random Forest contains many decision trees that act as a combined group. Here decision trees is used as a very popular method for various machine learning operations and they are seldom accurate. Each separate tree in the group gets further divided out into a predictions of a class and the class consisting maximum votes. Here it can be used to generate hundreds of such trees. As we have seen in a decision tree, when we have to divide a node further, each feature is taken in consideration and picks the one that produces the maximum difference between left nodes with respect to the right node. This leads to variation in the trees and less correlated outcomes.

2.4.6 K-Means Algorithm

k-means clustering is a classification method whose goal is to partition 'n' observations into 'k' clusters. Here we define k centroids for every clusters. These centres are present such that different position triggers unique outcomes. This algorithm occurs in a loop which makes the location of the centres to differ slightly on each iterations until it gets saturated to a certain position.

2.4.7 Linear Discriminator

Linear Discriminator is used or responsible for data visualization and reduction in dimension. It is a statistical function used in analyzing machine learning model classes for separation of class objects which gives relatively better value. It is best suited for binary class classification mainly used in linear regression techniques. To classify two features or class object in one single dimension the variables or features are plotted on an axis by minimizing the variance between them and maximizing the distance of average between their respective data points.Thus dimension reduction and good separable class is achieved in order to avoid computational costs and over fitting of data.

2.4.8 Linear SVC Mode

Linear SVC (Support Vector Classifier) Model is a Machine learning Classifier. This is used to design a line or plane to separate or split the features of one class into two classes and categorize them for best fit outcomes. his Classification method is best suited for large number of sample as the accuracy increases with the number of sample a class or a node posses.After the model fits it can be used to predict new outcomes.

2.5 Memory forensics and Memory Dump

It is forensic analysis of volatile memory or data in a memory dump. A memory dump is the captured data of a computer at a particular instant. This captured data may contain useful information of forensics data like system's condition before an attack or a crash. This dump compromises of RAM information for locating the cause of the particular attack [22].

Volatile memory is very important to understand the system's state and gives insights on the cause of the attack. Memory Forensics is divided into Evidence acquisition and Evidence analysis. Evidence acquisition is a tool used for collecting the data/memory like DumpIt for Windows or AVML used for Linux.Volatility is the best example for Evidence analysis.

2.6 Volatility in Memory Forensics

For Evidence Analysis[20], the most highly powerful open source tool used is Volatility. It supports multiple platforms like Linux, Windows, Mac etc.The Volatility Framework can be used to analyze crash dumps or VMware(Virtual Machine software) dumps. It is also used as malware analysis. Steps involved in Evidence Analysis are Selecting Profile, Viewing Running Processes, Using File Scan for example Kpcrscan which is used to scan KPCR (Kernel Processor Control Region) and also Psscan for malware analysis. Memory Artifacts and Volatility tool works hand to hand and these steps are further explained in Memory Artifacts.

2.7 Memory Artifacts

Memory Artifacts mainly compromises of memory profile, process list, command line history, IP address , hidden processes and file scan.Memory profile is selected for analysis for memory dump while command line history helps to identify the information of computer the memory dump file belongs to. For Example the command line "Type ./vol.py imageino –f <Destination of the memory Dump>" suggests profile of operating system the dump file belongs to. This Plug-in profiles also helps to provide the background of running process at the time the snapshot was taken. This running processes are called hidden processes. File Scan operations such as Kdbgscan, Kpcrscan, Dlllist, Dlldump, Pstree can be used to perform further investigation on the incident about the memory dump.

2.8 Normal and Abnormal Traffic

The growth in Traffic leads to more consumption in bandwidth and the study of traffic might lead to the cause which is crucial and may help in cybersecurity Forensics. Abnormal Traffic[21] is caused due to incident or anomalous behavior. To Classify the Normal and abnormal or anomalous traffic understanding the flow of traffic is important. Flow analysis is a method used to understand the patterns of the packets traveling in the traffic. By Analysing Packet Distribution and Packet size in attack and normal data set has shown that Abnormal Traffic contains packet with smaller size. The Observation has also shown that mainly TCP packet flows leads to dominate normal traffic.

2.9 Intrusion Prevention systems in IT and OT networks

In today's world, and even our project nothing works around without the use of machines, computers and digitized systems. That's how we know Information Technology, also called the IT industry has come to existence. Anything and Everything related to computers comes in the same Industry. It might be seen as a backcloth of the industrial setting since it's presence is usually covered over by the details of the respective industry. Such devices are regularly refurbished and reprogrammed to keep them up to date. It supervises the data and content of the entire industry. But here, are we really working on IT? To be unbiased we first need to catch up on the idea of OT, specifically, Industrial control systems. Now IT systems are the frameworks for all industry related process, but what actually controls the physical procedures of the industry? These are the Industrial control systems. These not only supervise the physical processes but also run them to reap

all our desired outputs. This could even be as small as a system of running a switch to control a process as large as a cooler in radioactive industry. Do we really need to go over the consequences if this system fails? If a cooler is turned off in radioactive industry for more than threshold time the whole unit could blow up and the mishap could cause side effects for generations to come. Even though the ICS is just administering over a switch its job has out-turn bigger than we can imagine. For actual supervision, the ICS may provide like a graphical control or some easy way to control over the physical processes to make it simple to be used. Adjustments could be made in a click and alarms could be issued right away! Now what's the difference in these two networks? One must think why our project focuses on ICS and what is the relation between IT and OT?

Before computers were invented, the supervision of physical process that we have talked about relied entirely on humans through push button switches or maybe on-off adjusters, modulators, etc. Now for humans to run liberally over a large scale plant to control these processes was difficult. Also maintaining faithful communication between different supervisors was necessary to keep the functionality constant. That's where IT steps up, because computers make it much simpler. As seen in our daily lives, everything is one click away! so like all industries, IT is the framework of OT industry also, but differently. Usually IT computers are connected on a network, hence the vulnerability quotient is low, but OT networks are stand-alone systems, making them highly vulnerable to attacks. Also, the consequences as discussed for OT networks are more pronounced, so every hacker tries to make their mark by injecting malware into such systems. A higher level management is provided to interface peripherals such as Programmable Logic controllers, Remote terminal units, human machine interfaces and others, and become a familia called SCADA.

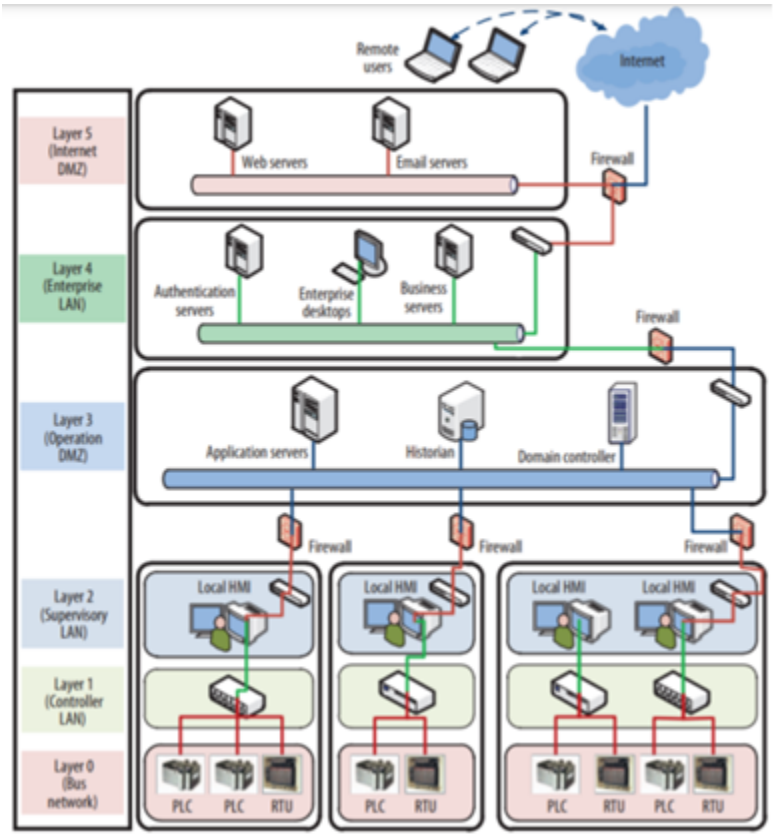


Figure 2.1: Purdue Model for Forensics over SCADA architecture - Source:[26]

The challenges posed by SCADA network and their hazardous ramifications instills the need of Intrusion prevention system. These systems provide prompt responses on averting any hazardous activity from happening and detects abnormality in the system. Intent of such systems is to rightfully identify the malicious activities and bring forth congruous responses effect immediately.It needs live capturing of network traffic information and monitor it in real-time before forwarding packets, and making it lightweight enough for lower utilization of resources. Today's SCADA network have more varieties in the form of information because packets come from traditional informational and communication technology and improvised sophisticated SCADA technology. Now we have employed various advanced ML techniques to function even in a mixed traffic profile. In a large scaled real time operated SCADA systems more precision has to be incurred. Situational awareness of malicious activities can be provided by tools like Wireshark, Tcpdump and IDS applications. Optimal attribute sets can be obtained to increase reliability of these techniques by data mining tools such as Weka ML toolkit. Filter and wrapper functions are studied as methodology for attribute reduction functions. Any anonymous user could pose serious damage in SCADA networks, hence must be completely avoided. Network rules must be defined and white-listing policies to be established for correct implementation of such systems.

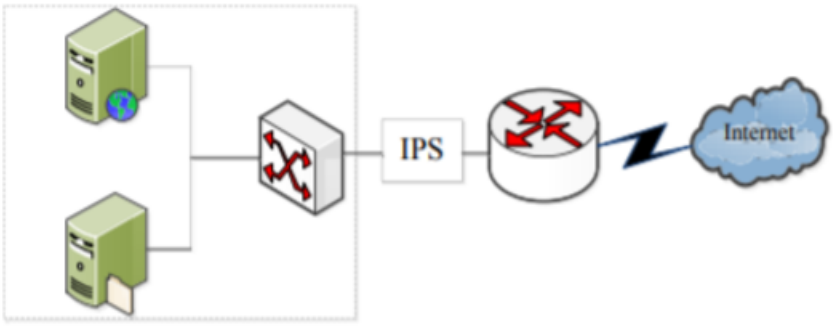


Figure 2.2: Deploying an intrusion prevention system - Source:[26]

2.10 Motivation

With time the architecture of the SCADA system has evolved such that with time we have moved from Network oriented systems to now Web-oriented applications are also available. All the traffic logged into the system can be saved by the Historian and sent over online gateways to secure alienated servers for analysis under the Industrial Internet of Things or fourth reinvention of the Industries. The involvement of internet increases the chances of damage to the availability and integrity of the system via various cyber-attacks. The main structures that are under prime threat by attackers include water purification plants, power generation plants, storage plants and other essential service providing factories. These attackers attack them by injecting malware into the systems by spoofing, scanning and malicious command and injections of data. When any intruder attacks the communication link between the PLC and SCADA system and sniffs out the information related to system control then the system loses its integrity to Man in the Middle attack. The Integrity of the system issue comes up when wrong commands and data are injected by attacker into the desired workstation, resulting into wrong decisions by the system. These attacks like Denial of Service or flooding attack can cause a havoc in the different plants. Current algorithms to encrypt and detect threat and firewalls are not suffice enough to take care of the advanced attack vectors planted sent by the intruders. This leads to find out methods to detect and prevent network intrusion by monitoring network activities not only on source-sink based but must check the various fields of communication parameters.

Chapter 3

LITERATURE SURVEY

One of the biggest rising concerns, cybersecurity in SCADA systems is righteously addressed in the paper published by T.Cruz in December 2016, where a Distributed Intrusion Detection System (DIDS) is presented [1]. A Dynamic Perimeter Intrusion Detection System is modelled on a CockPitCI project for ceaseless valuation of the security perimeter. To reap from both the anomaly detection and the signature-based techniques it uses two important components namely, OCSVM modules and correlators. It enlists this IDS's functional validation of various attacks even though the designed structure is domain specific with details about validation evaluation considering the speed and overheads involved.

Cybersecurity addresses the need of safeguard hosts, networks and their corresponding data from unapproved access, modifications and deletions. In May 2018 Y. Xin et al., published a paper to condense numerous traditional methods used in Deep Learning and Machine Learning to reinforce network and cybersecurity [2]. It lists down a few datasets easily accessible through net to test all these algorithms. Methods discussed for network security are ML based namely Support Vector Machine, K-nearest neighbours, whereas cybersecurity employed Decision tree and Dep Belief. Future direction towards newer improved developments and algorithms for further boost in the accuracy and detection speed have been cited.

Our primary focus, and in fact the infant procedure of this entire project stems its root from the ideas proposed in H.Hilal's paper of Network security analysis published in 2017 under IEEE[29]. Process automation in SCADA delineates from the real infrastructure but also accentuates the vulnerabilities of attacks. So penetration testing done best by Wireshark and Kali linux have been exploited to apprehend these attacks. Analysis implemented with abnormal and normal data traffic circumstances. The conclusive research is penetration of SCADA technologies using Kali Linux, which is used to perform the vulnerability attack and make data traffic between Human Machine Interface (HMI) with Programable Logic Controller (PLC) becomes cogent, and consequentially cause SCADA system to knock over with high data traffic on the network, thereby emphasizing that SCADA networks are unguarded against malware threats and attacks. The network analysis establishes that there are incoming and hiden users, (identified with User Anonymous with Wireshark). It recommends to maintain SCADA security from sundry attacks and threats by separating control networks with other communication networks, confining access, disabling ports on redundant endpoints.

To dive deeper into the details of forensics in SCADA we tried to manifest energy into the research already done on the tools of SCADA Digital forensics. Since this is a relatively new field of interest we found a paper released only three years back by Rima Asmar Awad[25] based on the ideologies and methodologies of the same. It begins with the reasons of sabotages in SCADA sector and highlights the importance of these growing technologies. It also provides a revised literature survey of the research challenges, technical challenges, the methodologies like Data Retrieval and Incident Response, Applying Existing IT Tools to SCADA, Case Study on USB-based Attacks, Live Data Acquisition and so on so forth. It divides its literature survey into network and end point forensics because ideally these follow different paths and as mentioned rightly, must be worked upon differently. They then approach into discussion of tools and recommend that cybersecurity in SCADA can be dealt in 3 steps: expanding the applicable frameworks, building end point precise tools to perform forensic analysis, and also to make them generalised in order to allow scalability of these procedures.

Even though Forensics is a well-developed field of cybersecurity, still it is scrutinized and dreaded upon when the application is SCADA industry. This is well highlighted by the article presented by Ahmed[26] in 14th IEEE International Symposium on Multimedia. This article is rather the easiest to comprehend. It starts with the basics of SCADA system and architecture, defines its components and protocols and leads on to the real time acquisition of data. For SCADA forensics it layers down the architecture and focuses on first 3 namely Layer 0 (Bus network), Layer 1 (Controller LAN) and Layer 2 (Supervisory LAN). The importance of live forensics is explained and so are its challenges, like Digital evidence validity and early data acquisition after malware injection. Furthermore, it emphasises that till date, the methodologies of live capture affects the system's performance. Extensive lower-layer data, inadequate logging, Resource-constrained devices and other problems are further discussed. The performance can be increased by preparedness of appropriate forensic tools. To acquire the most timed relevant data, preparedness must be made with a Data acquisition plan, and any serious damage to the system must be avoided by a monitoring plan. In order to inhibit exhaustion of resources on the network and endpoint the plan must be for lightweight data. All the forensic analysis can be further augmented by the plug-ins available.

In May 2019, Simon Duque Anton and his colleagues published a paper implementing all traditional ML algorithms over a fictitious industrial communication using Modbus/TCP protocol [3]. Out of the four algorithms, two of them namely, Support Vector Machine and Random Forest perform efficiently whereas the other two, k-nearest neighbour and k-means clustering performed significantly bad. A comprehensive description of the protocol used and the datasets with their respective features extracted are provided by the authors. Two metrics are listed to compare the results – accuracy and f-measure. Naïve approach is discussed to distinguish between derived and basic features and provides group of features distinguishable between normal and anomalous traffic. The first technique discussed is the SVM where the data is trained and further attributed by signum function. A linear kernel divides the normal and anomalous traffic. The next technique is the Random Forest explained with a brief overview of Decision trees. The last commands of this algorithm are made through consensus and they converge faster hence getting an upper hand as opposed to other ML techniques. The k-nearest algorithm is a distance dependent (Euclidean distance) determinant methodology and thus provides inadequate results to the given datasets with a significant increase in the number of false positives. The last technique described is the k clustering algorithm which is an extension of the latter except for the fact that the distance dimensions are measured from the centre and is the only unsupervised technique used in this paper. The results are easily comparable and techniques to improve them are mentioned such as determining the silhouettes coefficients. This paper emphasizes on network related features with their timing pattern.

Jiong Zhang [3] focuses on the anomaly detection by Random Forest Algorithm(RFA) where novel intrusions can be detected by outlier detection mechanism of RFA. Formation of attack library for IDS can be time consuming hence author proposes on building the patterns of dataset by RFA. The results obtained by evaluating the model on KDD'99 dataset were found efficient with higher detection rate and lowest false positive rate. However feature selection and imbalanced intrusions were highlighted as the challenges in proposed IDS.

Our project focuses on classifying SCADA network traffic, and the thesis written by Werling[27] is a pool of knowledge to work in this area. Today's SCADA network have more varieties in the form of information because packets come from traditional informational and communication technology and improvised sophisticated SCADA technology. The thesis proposes modern ML techniques to identify the abnormal traffic even in mixed profile. . Four supervised ML algorithms: Bayesian Network (BayesNet), Naıve Bayes, Naıve Bayes Tree (NBTree), and J4.8 Decision Tree are studied to provide FPR and TPR reports. Major attacks like watering hole attacks and phishing are elaborately discussed and the vulnerability of SCADA networks to be likely affected is reported. Situational awareness of malicious activities can be provided by tools like Wireshark, Tcpdump and IDS applications. Optimal attribute sets can be obtained to increase reliability of these techniques by data mining tools such as Weka ML toolkit. Filter and wrapper functions are studied as methodology for attribute reduction functions. An algorithm is considered efficacious if it procures a an FPR of $< .05$, the misclassification rate of ICT dataflows as SCADA and TPR of at least .99 for SCADA network traffic.Network based behavioural profiling is done by packet timing, data throughput, and packet size. Twenty four flow based features are used for total behavioural profiling. Again k-fold validation technique is used for cross validation due to its nature of full exploitation of information covered in training and testing datasets. The results clearly show increase in the efficacy of these algorithms with use of attribute

reduction functions. Reinforcement of ML algorithms over these functions even reduced build time and increased accuracy to greater than 98

Using empirical and analytical approaches optimum techniques of tackling cyber threats [4] have been proposed in the paper published by H.M Farooq and N.M Otaibi in 2018. Clustering algorithms are discussed foremost due to their ease and speed. K-means clustering, Density-Based Spatial Clustering (DBSCAN) and Balanced Iterative Reducing Clustering using Hierarchical (BIRCH) are drawn in a way to being out their advantages over each other. One Class SVM (OCSVM) Classifier are used for anomaly detections where baselining of traffic is more evolved than outliers. Predetermine of behaviour is done by linear regression studied in the paper with Decision tree (DT) regressor and Random Forest (RF) regressors. Message classification is done by RF classifier with pre-processing through Term Frequency Inverse Document Frequency. The step to shift from supervised to semi or unsupervised learning is emphasised.

ML based techniques compared on the basis of a thorough set of attacks provide a robust output in any general SCADA cyber threat [5]. To test the robustness of algorithms on attacks like Structured Query Language (SQL) injection attacks, backdoor and command injection, M. Zolanvari and M. A. Teixeira using seven different SVM, KNN, naïve Bayes (NB), RF, DT, logistic regression (LR), and ANN. The comparative analysis depends upon the true/false positives and negatives. A thorough evaluation of performance is made through other metrics like undetected rate, accuracy, Matthews Correlation Coefficient, false alarm rate and sensitivity.

For lowly detected anomalies like Remote-to-Local (R2L) and User-to-Root (U2R), Yongsu Kim implemented a four stage IDS system starting from pre-processing to feature extraction using the decision tree model [6]. The performance of the system is measured through metrics like accuracy, confusion matrix and receiver operation characteristic.

To broaden the horizon of attacks, we take the example of the largest data and code repository GitHub which with its incalculable expectancy to expand brings about vulnerabilities like malicious code repositories and dangers for users forking the data [7]. Their own defence mechanism is a robust inexhaustible resource called GitCyber which extracts features of content posted on GitHub and provides it to a Deep Neural network (DNN), where cybersecurity learning is consolidated through heterogeneous information network (HIN). This will engulf the social coding mechanisms into the DNN and designs neighbourhood relations based meta-paths. The network schema of HIN is developed through observational knowledge parameters like user-fork-repository, repository-have-file, user-comment-repository, user-star-repository and user-contribute-repository. The meta-path definition depends on neighbourhood through social norms like existence of two different source code from the same src library. A cyber-guided loss function parameterize the disagreement of cyber loss function for baselining the repositories. Various methods like BoW-DNN, BoW-SVN, Git-DNN, Git-SVN, M2V-DNN and M2V-SVN are used for comparison with the GitCyber with Stability, Scalability and Sensitivity.

Volatility became our official weapon of forensic analysis after we dived into the case studies in the paper published by Srivastava in 2017 in the IEEE ICOS[28]. This is a rather fermented paper because it has been trying and testing on code injected malwares which are even harder to detect and primarily the function of our project. It emphasis that the processing memory has the bulk of information with the executable files, in simpler words, the whole paper focuses on extracting execution stack forensically. To detect anomalies of code injection we use the same and take help of the VAD (virtual address descriptor) tool. Consistency verification or in paper's terms determining integrity is reduction of volumes of data for cross validation and exclusion from further analysis. When code injection takes place it occupies space of a legitimate executing process with correct protection required for the job. The injected code is so modified that commands on volatility like malfind would fail. The efficacy seems to be considerable over various cases and its future work defines investigation of APIs and create signature of API functions.

Our analytics in memory forensics roots from the research done in paper presented by M. Murthaja, B. Sahayanathan and their team in the International Conference on Advancements in Computing (ICAC), 2019[24]. This paper explores Volatility as a tool to automate the feature selection, extraction, and the model trained on it when a Malware is injected onto a system with memory dump. The features extracted were from domains like DLLs, APIs, Registry, and network. Furthermore, by studying these features over 4 types of malwares it was also concluded that the features obtained through DLL like ldr modules, DLL load time, ldr mapped path, DLL load time, ldr base, envars process, DLL path, envars base, DLL load time, etc. were most effective in detecting the malicious activities by behavioural markings. The extracted feature sets were trained with the support vector clustering (SVC), Gaussian naive Bayes (GNB), Logistic Regression (LR), Decision Tree Classifier (DTC), linear discriminant analysis (LDA), random forest classifier (RFC),and K-nearest neighbor classifier (KNN), using the python Sci-kit libraries with K-fold cross validation method to analyse their reliabilities. Their future aim suggested classifying malwares further into variety of families through thorough understanding of their impact, spread and patterns. They also aim to study more features because DLLs might fail to understand the intricacies of a few malware classes.

3.1 Gap of Literature Survey

The entire survey discusses what cybersecurity is, what types of attacks are performed and how they are tackled by various ML based techniques. All of these datasets seem domain specific and significantly different from the real world data generated in industry. Testing algorithms in these environments might generate an ideal world output which can't be employed unless tested with stricter parametric tools in the actual SCADA system. Also these provide relatively good values for attacks that we've known till now. Cyber criminals have newer advanced techniques of penetrating the system which might go unnoticed by these techniques if further learning isn't provided.

A better way of going forth is studying the intruder's behaviour and generating more detailed logs for such circumstances and developing a dynamically adaptable technique.

The troubles in live capture are not precisely defined in the literature survey. Malicious activities that are code injected might not be classifiable by ML algorithms and might need specialised Memory dump analysis. Our gathered information of Volatility as a tool has always been general memory forensics and to adapt it to the intricacies of SCADA forensics is a tiresome journey because its adaptation to protocols and SCADA devices is limited, which will also be shown in the examples that we will discuss in this report.

3.2 Summary of Literature Survey

Conclusively, SCADA systems have in numerous variety of attacks each targeting vulnerable data with creative spoofing, man in the middle, malware injection for economical gain. To prevent these attacks we need a robust IDS system which will have its performance metrics aligned significantly high for all domains. Cross validation or deterministic integrity defines the performance of various algorithms used. The TPR and FPR values of these ML processes defines their use in this application.

The development of such algorithms require testing over heterogeneous datasets. Five techniques out of these have been studied further in our project namely, SVM, K-means, Decision tree, linear separators.

Most of the information is situation in the executing stack of the endpoint device and that is what is extracted by the volatility tool. It has been used to capture malicious activities when trying and testing on code injected malwares which are even harder to detect and primarily the function of our project.

Forensics in SCADA is not the easiest field in fact it has limitations from research and technology point of view also. The methodologies of live capture affects the system's performance. Extensive lower-layer data, inadequate logging, Resource-constrained devices and other problems are further discussed. The performance can be increased by preparedness of appropriate forensic tools. To acquire the most timed relevant data, preparedness must be made with a Data acquisition plan, and any serious damage to the system must be avoided by a monitoring plan. In order to inhibit exhaustion of resources on the network and endpoint the plan must be for lightweight data. All the forensic analysis can be further augmented by the plug-ins available.

Table 3.1: Popular Techniques of Anomaly Detection

Serial No.	**Techniques**	**Popular Algorithms**
1	Density Based Techniques	K Nearest Neighbor, Local Outlier Factor, Isolation Forests
2	Decision Trees	ID3, Logistic Model Tree
3	Bayesian Classifier	Multinomial Naive Bayes Classifier
4	Support Vector Machines	Linear, Non-Linear, Multi Class
5	Clustering And Outlier Detection	K-Means, Hierarchical
6	Discriminant Analysis	Linear, Quadratic, Normal and Fischer

Chapter 4

Proposed Methodology

4.1 Problem Statement

Develop a forensic procedure to scrutinize network traffic generated in SCADA environment and carry out memory dump analysis over it in order to identify malicious files. Further, to employ empirical machine learning techniques such as Decision Tree, Random Forest, K-Means Clustering and Linear Discriminant Analysis to classify the obtained data accurately.

4.2 Objective and Goals

SCADA systems are control systems that are mainly used for monitoring and controlling process throughout the industry for various domains. Intrusion detection in these SCADA systems is a very important technique that helps in analysis of data and systematically identify potential threats on the process in SCADA system. These threats are seen when any hacker or ill-legitimate user gains rights to perform actions on the network and make it look legit, which may cause a disruption in our system. To detect this we propose a semi-automated method of log processing and analysis. In this report we have seen the efficiency of Naïve bayes, Classifiers, K-means clustering and decision tree when we deployed attacks on the network. Initial study suggested that the approach is quite effective to detect the fishy events or anomalies.

1. To understand and respond to data breaches and other security incidents and forensic analysis for our case study.
2. To select suitable machine learning algorithm to model the behaviour of an attacker.
3. Network forensic analysis of the SCADA system.
4. To analyze the memory dump using forensic procedure to identify the suspicious malware files in the Water treatment plant.

4.3 Proposed System

4.3.1 Network Forensics in ICS Environment

As we've discussed thoroughly in our litersture survey that there are two types of forensics procedures that could be followed, endpoint forensics and network forensics. While, we'll be going through the nitigrities of endpoint forensics in the name of memory dump analysis later in the report, lets first dive into network forensics, encompassing protocol structures, IP and port information, packet analysis,Data collection, base lining and classifying the network behaviour as normal and abnormal. The input to our forensics system is a PCAP file. These are essential carriers of full content data of network. Every segment of packet transfers are fully captured by these files. Now these could be acquired using tools like TCPDUMP and can be visualised graphically using Tshark/Wireshark. These softwares help to accentuate high-level data streams that the analyst might actually miss, like IP protocols not being exhibited. Alert messages can also be customised into such system by defining "rules" or "policies" over these devices. These are typically produced by Network Intrusion Detection Systems (NIDS). Lastly, probably the most important information gathered from PCAP files, even for which we can use Wireshark is apprehending statistical information like number of packet transferred between two End point devices, the opcodes majorly sent in the network, the number of protocols and services being used at a time, average packet speed, etc, which could also be used as pointers in anomaly detection.

There is existent difference between Host based and Network based IPS system. A NIPS has the potentiality of dropping or changing detrimental detected packets which have been detected by analysis at strategic points within a network for segments or devices. While this is a guard for all systems on the network, a host IDS only focuses on a single endpoint. What we are trying to carry out is an forensic investigation along with intrusion detection which has additional elements involved like inclusion of head to toe concept, i.e, we try to dig to the roots of attack and climb to the consequences of its effect. After evidence is gathered it is thoroughly analysed, which will also be performed in our memory dump analysis. Investigation of networks for security doesn't fundamentally branch out into prosecution, but can motion a legal action like terminating job of the hacker or employee that helped in seeping into the system.

1. **Identification:**

   We have already identified why OT networks, SCADA specifically are the most vulnerable networks. How do we identify what is abnormal and normal behaviour? For this, we set up a monitoring policies. This could be of various types, like, Blacklist monitoring, also referred as the enumerate evil. In this method, signs are distinctly specified and abnormality is written clearly in the form of rules. It has the best accuracy when compared with other policies, but when we try to implement it in real life, is it really practical? There are new attacks coming up daily and hackers are becoming smarter than ever in this era of digital evolution, it won't be days before a malware could be injected inside the system that has not been blacklisted and could hamper the entire ICS environment. The Anomaly or whitelisting monitoring defines the normal rules explicitly and anything beyond or a step away from these rules gets listed as abnormal behaviour. It must be kept upto date if any new endpoints or services are to be provided. The rate of false positives are lower than that of black listing policies. An advanced form of whitelisting monitoring is the policy monitoring which uses staistical means to define the normal behaviour of the network.

2. **Data acquisition:**

   Based on the resources available, and the technical nitigrities, keeping a comprehensive tab on everything becomes difficult due to processing and storage issues. Data acquisition could be live or dead, or more commonly known as active or passive. This process is called sniffing and may even incorporate scrutinizing the network ports of systems to determine their contemporary state. till date, the methodologies of live capture affects the system's performance. Extensive lower-layer data, inadequate logging, Resource-constrained devices and other problems are further discussed. The performance can

be increased by preparedness of appropriate forensic tools. To acquire the most timed relevant data, preparedness must be made with a Data acquisition plan, and any serious damage to the system must be avoided by a monitoring plan. In order to inhibit exhaustion of resources on the network and endpoint the plan must be for lightweight data. All the forensic analysis can be further augmented by the plug-ins available. Now for network analysis we acquire traffic on cables using network taps. The cables could be twisted pair or coaxial cables, fiber optic cables and the taps could be Inline network taps, vampire taps, and fiber optic cable taps. Another way of capturing data traffic is by connecting a system to the hub itself which relays the traffic from any ports to all ports, implying that our endpoint on the network will have access to each packet going and coming through the hub.

3. **Data collection:**

   This stage only begins if we can accurately distinguish between the normalcy and abnormally of traffic in the network. For this we need a pre-process, i.e, base-lining of network traffic, filtering of the data collected and then building signatures or features upon them. The baselining we perform is called network baselining where trends and patterns on the network over period are abstracted to distinguish traffic. Traffic volume, TCP conversations, session data, etc can be used. Some baselines like casting method - unicast, broadcast, and multicast where identification of new hosts, rogue DHCP-servers, rogue routers and VPN-tunnels can be done by baselines. It should include source and destination IP addresses, MAC addresses, port numbers, typical types of packets, application used, etc. If a host is suspected to corrupt the traffic, the monitoring policies discussed can be adapted as baselines and can be compared to abstract necessary information. Attacks like man in the middle and denial of services try to re-route packets to sniff traffic, so baselines over routing protocols also can be established. Login/Logout sequences, Operating system installation, backups and upgrades, browsing sessions, Application launch sequences and key tasks, are other features over which baselines are developed. Constricting down a colossal pond of prospective corroboration to one or more subsets of scrutinizing data. Forensic Analyst deal with more traffic than what can be analysed without pressurizing the resources, this step reduces data to a reasonable amount. Features or signatures are developed over these "narrowed down" data to get a meaningful list of sceptical values, like exploit code patterns, hash-sums of known-bad files, filenames, etc. which are part of the anomalous behaviour.

4. **Analysis:**

   What do we do with the data that we've captured? The approach we employed is analysis through Tshark/Wireshark. Detecting different attacks require different approaches. Like for detecting an enumeration, which is like the progenitor of attacks where maximum information about network traffic is extracted, we need to specifically look out for unseen hosts that have rising interests in the network information and relay all information to themselves, or activities like network scan or port scan to extract the information on active IPs and Ports of the network.This means a repeating (for each target IP-address) pattern of packets. It can be effortlessly implemented as an ICMP echo (or 'ping') request, or packets destined to specific TCP or UDP ports. Usually this attack is to set a target and their is repeated pinging or trend of connection attempts. More difficult to track are the distributed scans when many endpoints begin port scanning to identify targeting host. Scan detection can be done best by whitelisting or policy monitoring. To step up the complexity of scanning attack, hackers perform probe scanning, which is higher up in the protocol stack. After these types of attacks we have the lateral movement detection, because scanning attack often fail to be recognised by the organisations. To obtain managerial or administrative access over the network, attackers manipulate the initial system as a prerogative, for ambushing other systems in the organisations network, til they reach their targeted endpoint. In one of the case studies in our project we tried to analyse the same, a lateral movement in an ICS organisation, to turn on or off the PLC and disrupting in the processes of waste water treatment plant. The few star points while detecting lateral movements are firstly, detecting unknown or unverified hosts in the network that are also not mentioned in the whitelisting policies of the organisation's network. Secondly, is the presence of sessions of network traffic that are also not white

listed. The attacking endpoint will try to make conversations with Engineering workstations in order to get the intricacies of the ICS networks. This should send alert data and must be detected by the NIDS. The suspected system usually has an unusual behaviour like numerous attempts to crack the password using VPN, or port scanning, even probe scanning. After getting the access of passwords these devices send packets to the targeted endpoints whose nature is suspicious. In our case study the PLC variables were trying to be reprogrammed, and their subscriptions were being changed. This clearly accounts for the malicious activities that the hacker wants to perform.

5. **Reporting:**

   After the analysis, the information extracted in this process is sent to the concerned authorities. As mentioned, Investigation of networks for security doesn't fundamentally branch out into prosecution, but can motion a legal action like terminating job of the hacker or employee that helped in seeping into the system. We then employ the use of threat intelligence to actually form up a decision. Now Cyber Threat Intelligence (CTI), is of various types like: tactical, strategic, operational and technical. Depending on the organisation and the level of damage done by attacks, decisions are made. Information about impending attacks could be given to board managers, high level staff, or can be consumed by defenders and incident responders in order to rebuild the wall of security. If a faster approach is to be employed without human interaction then technical cyber threat intelligence is used. The artifacts obtained also decides the measures taken ahead. Lateral movement for example exhibits the vulnerabilities of the entire network hence redesigning might also need to take place.

4.3.2 Analysing PCAP files:

We carried out analysis of some PCAP files, and some of the results that we obtained are shown below. The following PCAP file is obtained from setup of ICS environment. This approach can be used generally regardless of the type of attack:

1. Ethernet endpoints obtained using tshark. Siemens PLC and other workstations in the networks were discovered.

```
C:\Program Files\Wireshark>tshark -q -z endpoints,eth -r D:\ATHARVA\coe.pcapng
================================================================================
Ethernet Endpoints
Filter:<No Filter>
                      |  Packets  | |  Bytes  | | Tx Packets | | Tx Bytes | | Rx Packets | | Rx Bytes |
Elitegro_b0:c1:c3         536408      51809884     357068        33226724      179340        18583160
SiemensN_1c:69:e4         526584      49791460     179340        18583160      347244        31208300
Broadcast                 256093      16236953          0               0      256093        16236953
Dell_87:d4:96             255332      17814332     255332        17814332           0               0
Spanning-tree-(for-bridges)_00      128360       7701600          0               0      128360         7701600
IPv4mcast_7f:ff:fa         40313       8659831          0               0       40313         8659831
Hirschma_ea:39:15          34218       2510600      34218         2510600           0               0
Hirschma_ea:39:13          34218       2510600      34218         2510600           0               0
Hirschma_ea:38:de          34218       2491448      34218         2491448           0               0
Hirschma_ea:39:37          34218       2472296      34218         2472296           0               0
LLDP_Multicast             25944       4687316          0               0       25944         4687316
Tp-LinkT_6c:83:e6          25023       3395925      25023         3395925           0               0
Tp-LinkT_6c:8c:97          18148       2890430      18148         2890430           0               0
Tp-LinkT_6c:85:91          15463       1224160      15463         1224160           0               0
SiemensN_1c:69:e5          12836       1270764      12836         1270764           0               0
IPv4mcast_fb               10767        955213          0               0       10767          955213
IPv6mcast_fb                9655        995178          0               0        9655          995178
IPv6mcast_01:00:02          8735       1328667          0               0        8735         1328667
Tp-LinkT_6c:8f:b0           7468        671225       7468          671225           0               0
IPv6mcast_16                7357        662430          0               0        7357          662430
IPv4mcast_16                7340        440460          0               0        7340          440460
IPv4mcast_fc                7030        469576          0               0        7030          469576
IPv6mcast_01:00:03          7030        610176          0               0        7030          610176
VMware_bb:33:ef             6355        591735       6355          591735           0               0
VMware_d4:e8:fd             6090        585485       6090          585485           0               0
VMware_64:e3:88             5820        544395       5820          544395           0               0
Rockwell_a1:89:a4           2852        171120       2852          171120           0               0
HewlettP_28:87:81           2374        479587       2374          479587           0               0
Hirschma_ea:39:0b           2128        546896       2128          546896           0               0
Hirschma_ea:39:0c           2128        566048       2128          566048           0               0
Elitegro_b0:c1:a5             16           960         16             960           0               0
IPv6mcast_ff:03:89:e4         15          1290          0               0          15            1290
IPv6mcast_ff:70:ef:a8         15          1290          0               0          15            1290
IPv6mcast_ff:52:35:dc         15          1290          0               0          15            1290
IPv6mcast_ff:74:36:59         15          1290          0               0          15            1290
IPv6mcast_ff:ea:8c:81         15          1290          0               0          15            1290
IPv6mcast_ff:57:24:ed         15          1290          0               0          15            1290
IPv6mcast_ff:26:51:a6         15          1290          0               0          15            1290
================================================================================
```

Figure 4.1: Ethernet Endpoints obtained

2. Protocol Hierarchy and number of packets transferred in each protocol type can be extracted using Wireshark or tshark.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	1035313	100.0	92547890	11 k	0	0	0
Ethernet	100.0	1035313	15.7	14494382	1806	0	0	0
Logical-Link Control	12.4	128360	5.4	5006040	623	0	0	0
Spanning Tree Protocol	12.4	128360	5.0	4620960	575	128360	4620960	575
Link Layer Discovery Protocol	2.5	25944	4.7	4324100	538	25944	4324100	538
Internet Protocol Version 6	3.2	32882	1.4	1315280	163	0	0	0
User Datagram Protocol	2.5	25420	0.2	203360	25	0	0	0
Multicast Domain Name System	0.9	9655	0.4	396568	49	9655	396568	49
Link-local Multicast Name Resolution	0.7	7030	0.2	174316	21	7030	174316	21
DHCPv6	0.8	8735	0.9	787097	98	8735	787097	98
Internet Control Message Protocol v6	0.7	7462	0.2	209656	26	7462	209656	26
Internet Protocol Version 4	58.6	606616	13.1	12161680	1515	0	0	0
User Datagram Protocol	8.4	86956	0.8	695648	86	0	0	0
Simple Service Discovery Protocol	3.9	40313	7.5	6966685	868	40313	6966685	868
NetBIOS Name Service	1.0	9975	0.5	498750	62	9975	498750	62
NetBIOS Datagram Service	0.1	851	0.2	171051	21	0	0	0
SMB (Server Message Block Protocol)	0.1	851	0.1	101269	12	0	0	0
SMB MailSlot Protocol	0.1	851	0.0	21275	2	0	0	0
Microsoft Windows Browser Protocol	0.1	851	0.0	28083	3	851	28083	3
Multicast Domain Name System	1.0	10767	0.5	483109	60	10767	483109	60
Link-local Multicast Name Resolution	0.7	7030	0.2	174316	21	7030	174316	21
Data	1.7	18020	0.8	720800	89	18020	720800	89
Transmission Control Protocol	49.5	512320	32.9	30483540	3799	342260	16963710	2114
TPKT - ISO on TCP - RFC1006	16.4	170060	0.7	680240	84	0	0	0
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol	16.4	170060	0.6	510180	63	0	0	0
S7 Communication	16.4	170060	9.6	8928210	1112	170060	8928210	1112
Internet Group Management Protocol	0.7	7340	0.1	117560	14	7340	117560	14
Address Resolution Protocol	23.3	241511	12.0	11109506	1384	241511	11109506	1384

Figure 4.2: Protocol Hierarchy

3. The IP endpoints discovered to get a set of whitelisted IP values.

```
C:\Program Files\Wireshark>tshark -n -r D:\ATHARVA\coe.pcapng -q -z endpoints,ip
================================================================================
IPv4 Endpoints
Filter:<No Filter>
                  |  Packets  |  |  Bytes  |  | Tx Packets |  | Tx Bytes |  | Rx Packets |  | Rx Bytes |
172.16.0.71          521792      50918940      349584         32763700      172208         18155240
172.16.0.2           512320      48935620      172208         18155240      340112         30780380
239.255.255.250       40313       8659831           0                0       40313          8659831
172.16.0.241          30516       3740224       30516          3740224           0                0
172.16.0.255          16824       1614719           0                0       16824          1614719
172.16.0.164          14835       2621370       14835          2621370           0                0
172.16.0.163          14205       2490615       14205          2490615           0                0
224.0.0.251           10767        955213           0                0       10767           955213
224.0.0.22             7340        440460           0                0        7340           440460
224.0.0.252            7030        469576           0                0        7030           469576
172.16.1.21            6008        492656        6008           492656           0                0
172.16.1.255           6008        492656           0                0        6008           492656
255.255.255.255        6004        492328           0                0        6004           492328
172.16.0.161           5035        427350        5035           427350           0                0
172.16.0.162           3690        287685        3690           287685           0                0
172.16.0.183           2915        210135        2915           210135           0                0
172.16.0.185           2825        221710        2825           221710           0                0
172.16.0.181           2515        176565        2515           176565           0                0
172.16.0.210           2235        469268        2235           469268           0                0
192.168.0.181            45          6315          45             6315           0                0
192.168.0.255            10          2430           0                0          10             2430
================================================================================
```

Figure 4.3: IP endpoints

4. TCP conversations of some suspected IP addresses.

5. ARP request, response conversations, also marking out scanning attacks.

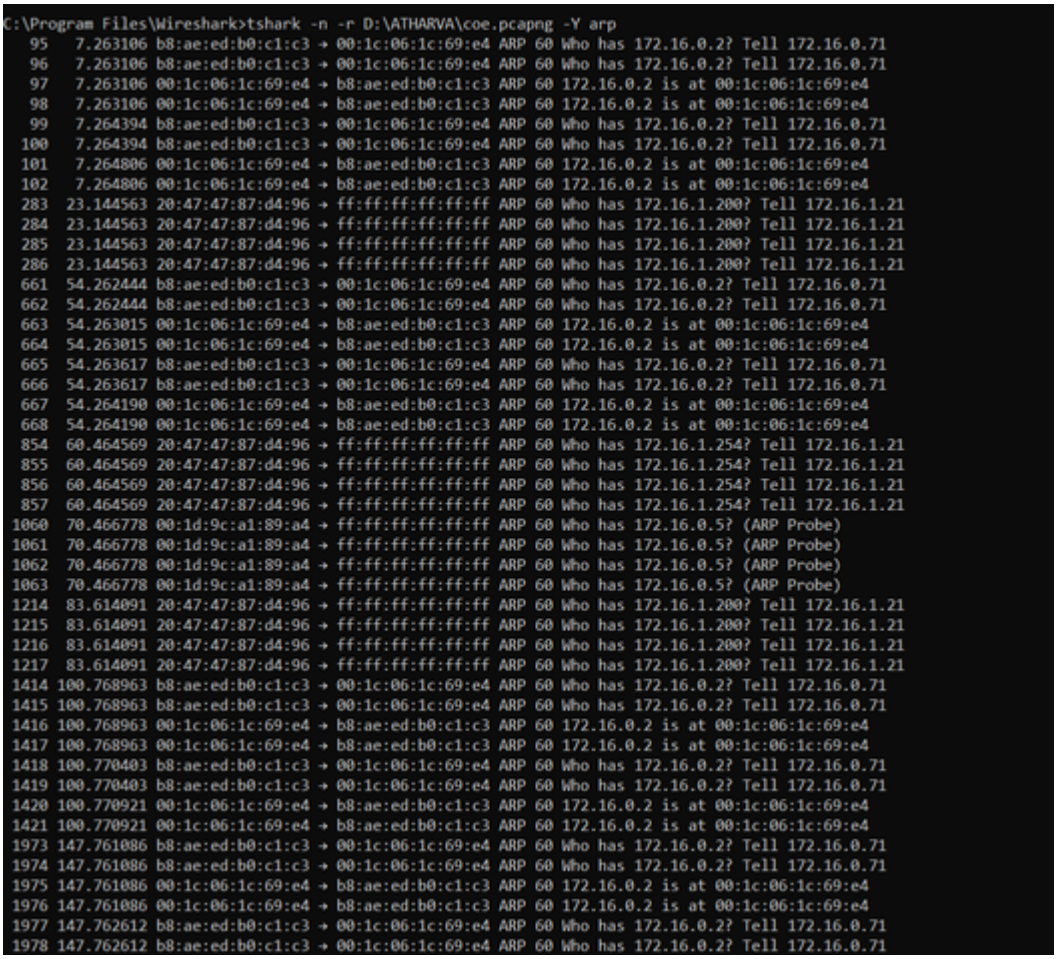


Figure 4.4: scanning attacks

6. Obtaining opcodes of packets sent between suspicious endpointsusing Wireshark or tshark.

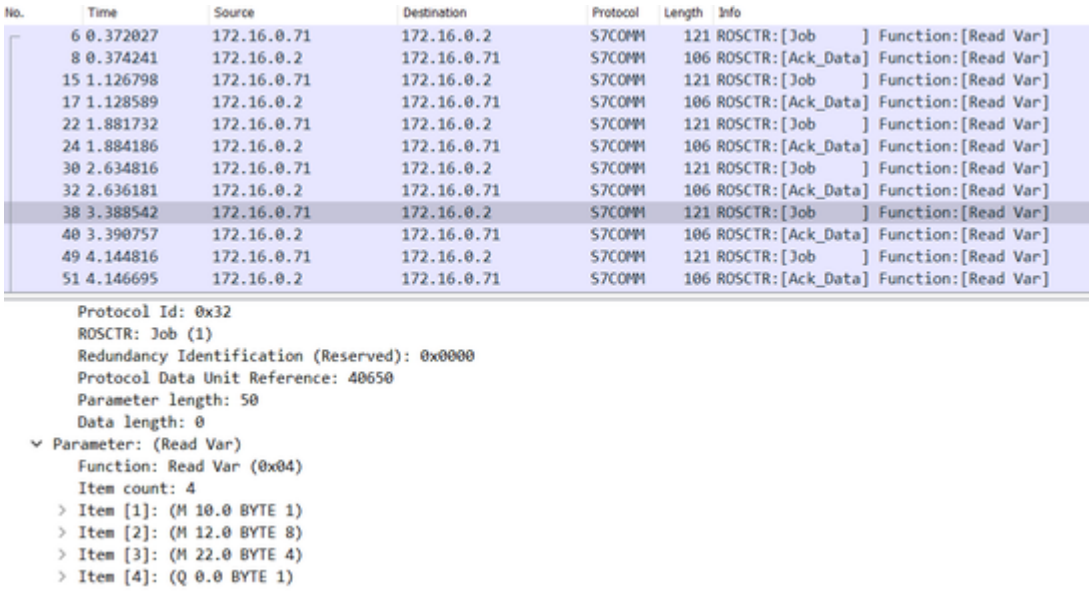


Figure 4.5: opcodes of packets

7. Checking out the variables of the packets sent by the suspected end point.
8. Checking for multiple password unlocking trials, maybe even the use of VPN for the same.
9. Checking out change in the behaviour of objects like PLC after a new endpoint has been detected in the network. Use of protocols not mentioned in the whitelisting policies. This could even be a change as small as use of S7 protocol instead of S7 plus or vice versa.

4.3.3 Forensic Analysis of Memory Dump

Till now we discussed network forensics and its implications over our Intrusion prevention system. But this is only half the possible ways of going ahead with forensics investigation of given malicious activities. We also need to consider single hosts to extract out exactly what happened. This is called computer forensics, subset of digital forensics like the network framework we studied. It also employs the same steps for investigation that we will be discussing. For including our project along with the investigation each point is explained in corollary to the forensic procedure.

Memory forensics is a growing field in digital forensics today which includes recovering, extracting and analysing evidence such documents, images and chat histories from volatile memory devices to non-volatile memory devices. This is the sole means of investigation that remains functioning, when attackers do not manipulate data to the endpoint's non-volatile storage throughout carrying out their malicious activities. Furthermore, this technique permits the scrutiny of more escaping evidence, i.e. RAM content that would vanish subsequently if a system reboots or is powered-off, in corollary to the present state of the kernel(operating system) or the executing processes running on the system. Memory forensics hence complements network forensics as well as computer- based forensics.

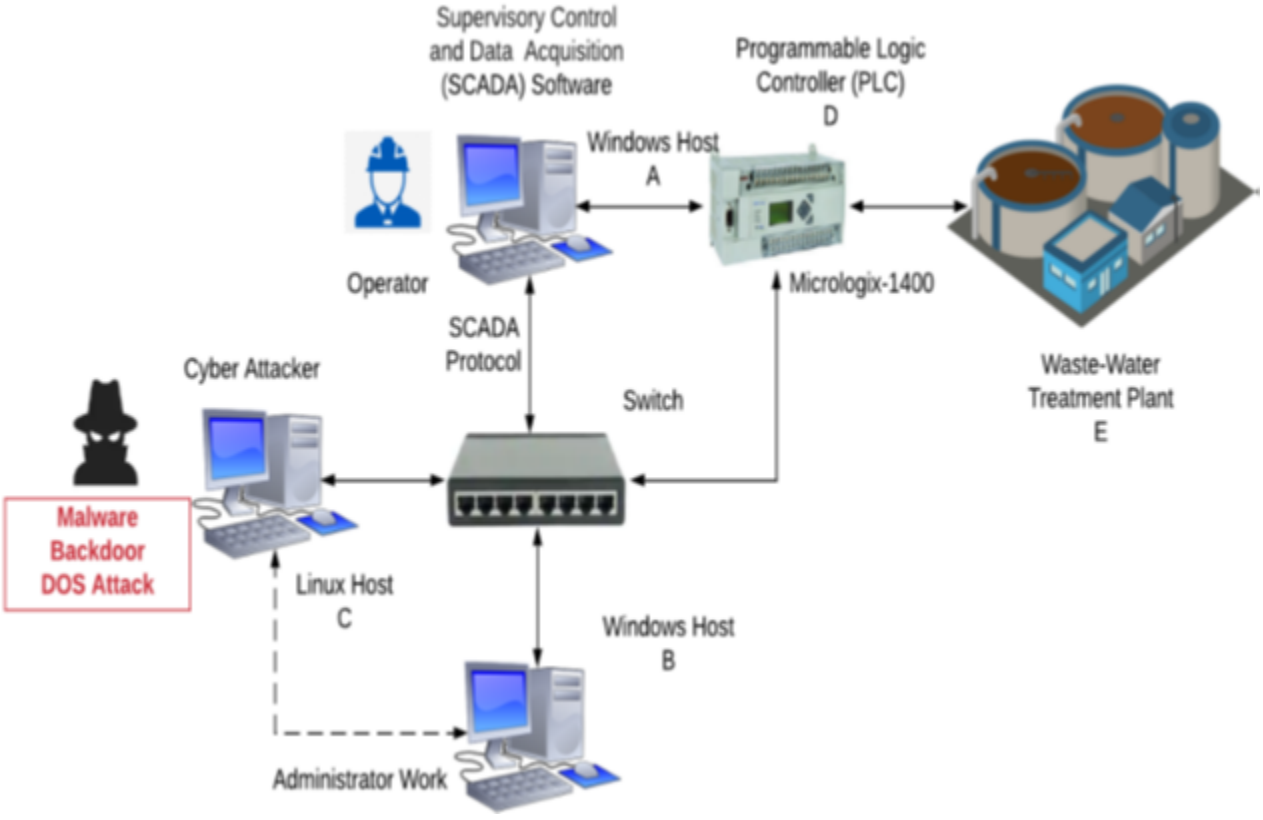


Figure 4.6: Testbed Architecture - Source:[4]

This is the test bed architecture on which the attack was performed. The engineering workstation was identified as the targeted endpoint. Data was collected using FTK Imager forensic tool in the duration of attack. The attack performed was denial of service where the network was flooded with packets so as to be overwhelmed and finally cease to service properly. Artifacts were collected using the volatility tool, malware view in order to carry out realistic analysis over it.

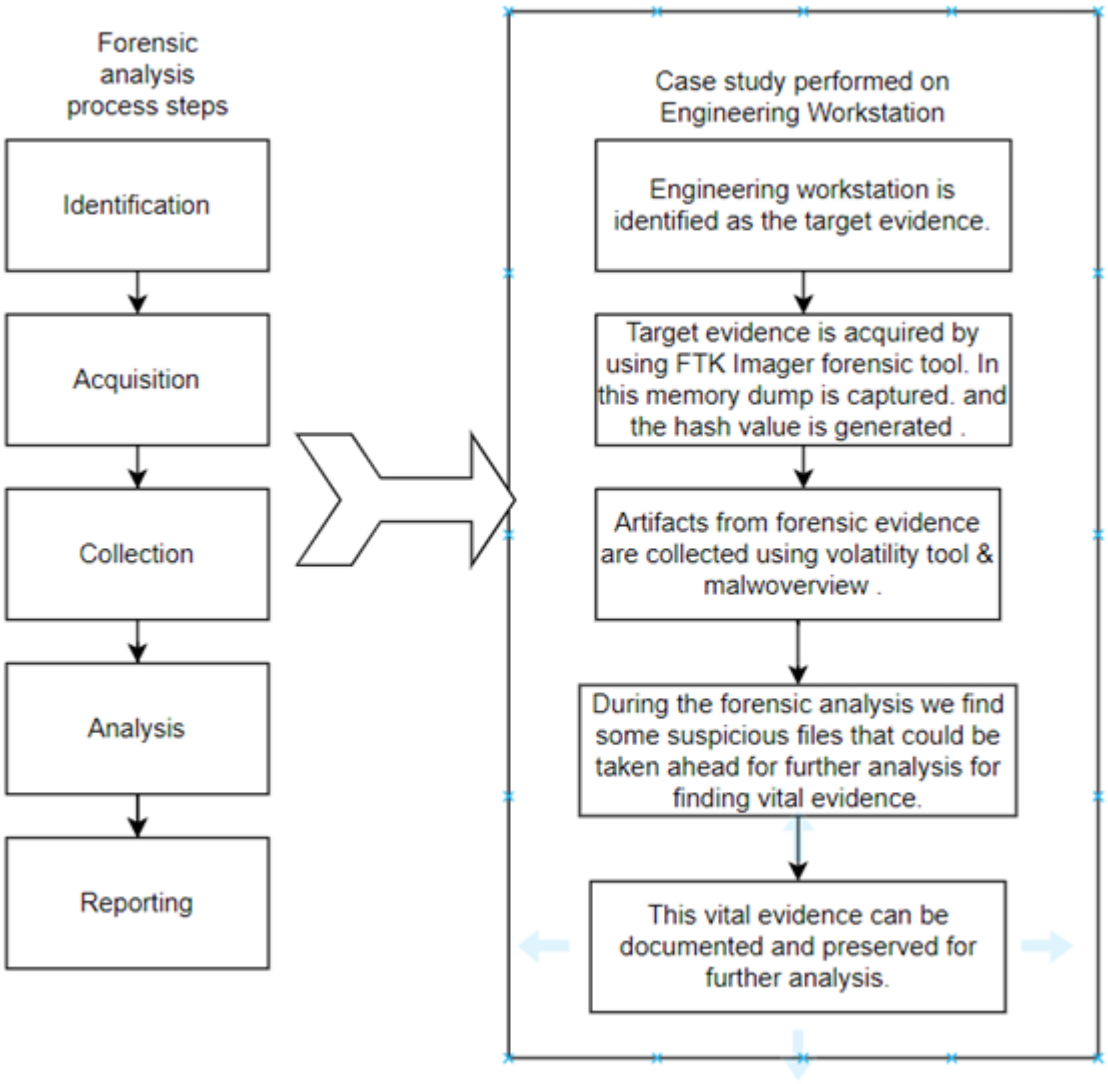


Figure 4.7: Flow Diagram of forensic analysis of the Memory Dump

Identification:

Now this attack essentially involved engineering workstation being code injected using a malware corrupted file using email or manually inserting the file through pen drive. But this causes backdoor and enables the attacker to control the PLC. Infact, here DOS attack is being performed on the system. This can be easily identified by the change in network traffic employing a statistical approach as discussed above because the protocol and the number of packets delivered in that duration increases drastically.

Acquisition:

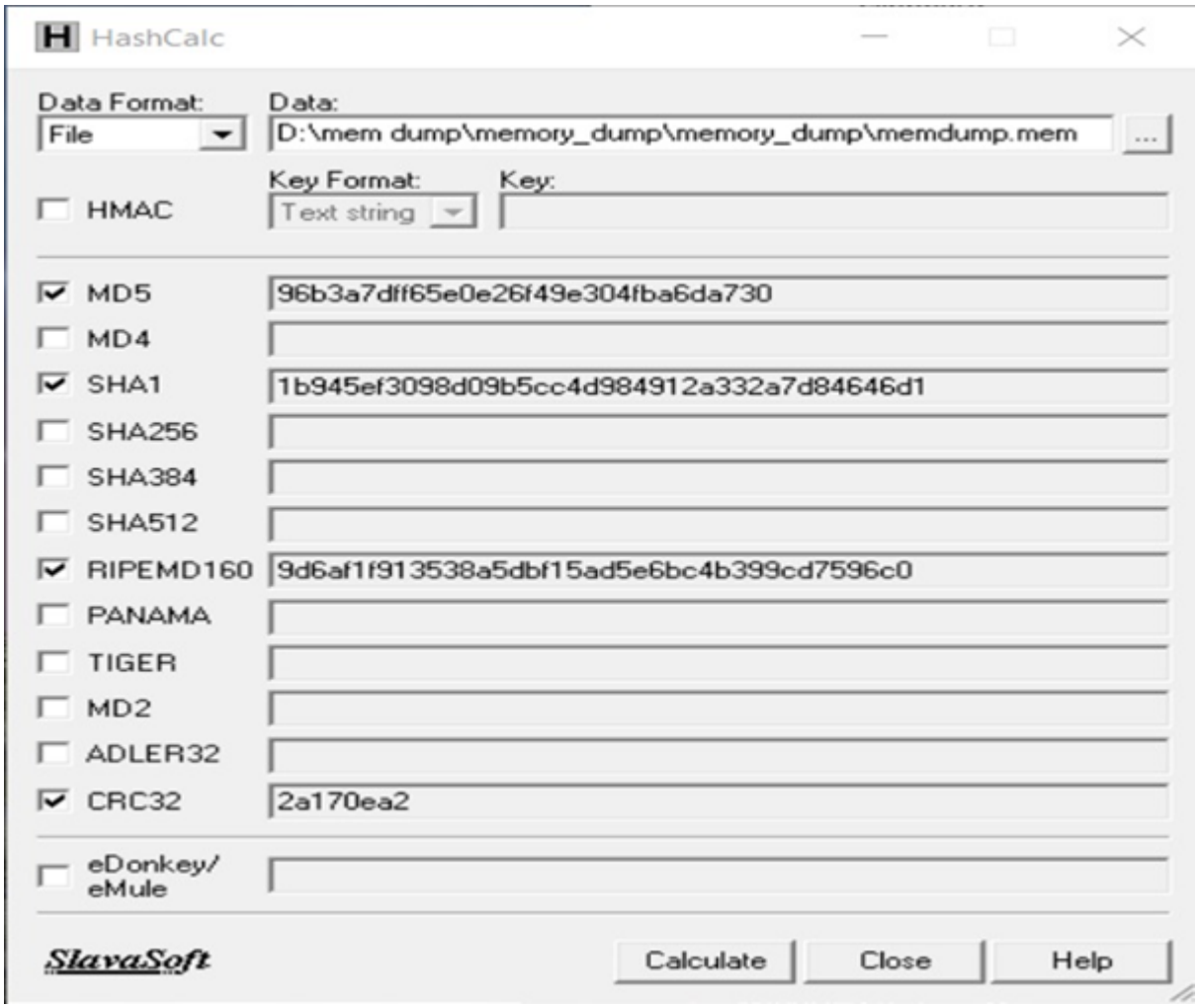


Figure 4.8: Hash Value for evidence memory dump

The volatility framework was used to process memdump.mem obtained using FTK Imager. Memory dump is acquired using FTK Imager tool.

FTK Imager is an open-source software by AccessData that is used for developing perfect copies of the original evidence without actually making any changes to it. The Image of the original evidence is remaining the same and allows us to copy data at a much faster rate, which can be soon be preserved and can be analyzed further. Inbuilt integrity checking function is provided which generates a hash report which helps in matching the hash of the evidence before and after creating the image of the original evidence.

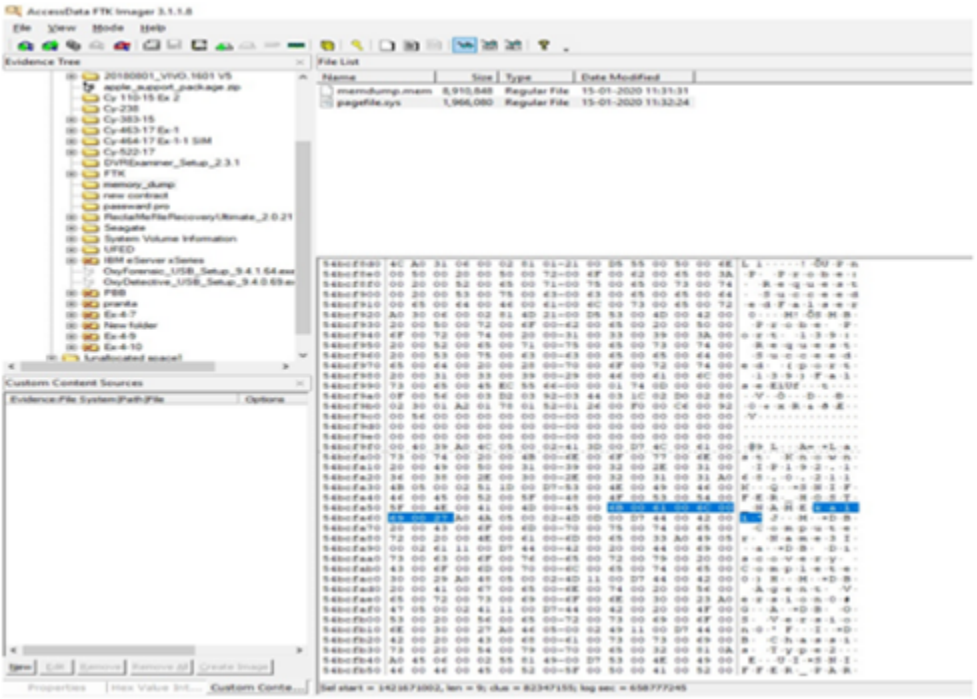


Figure 4.9: Extraction of memory dump using FTK Imager

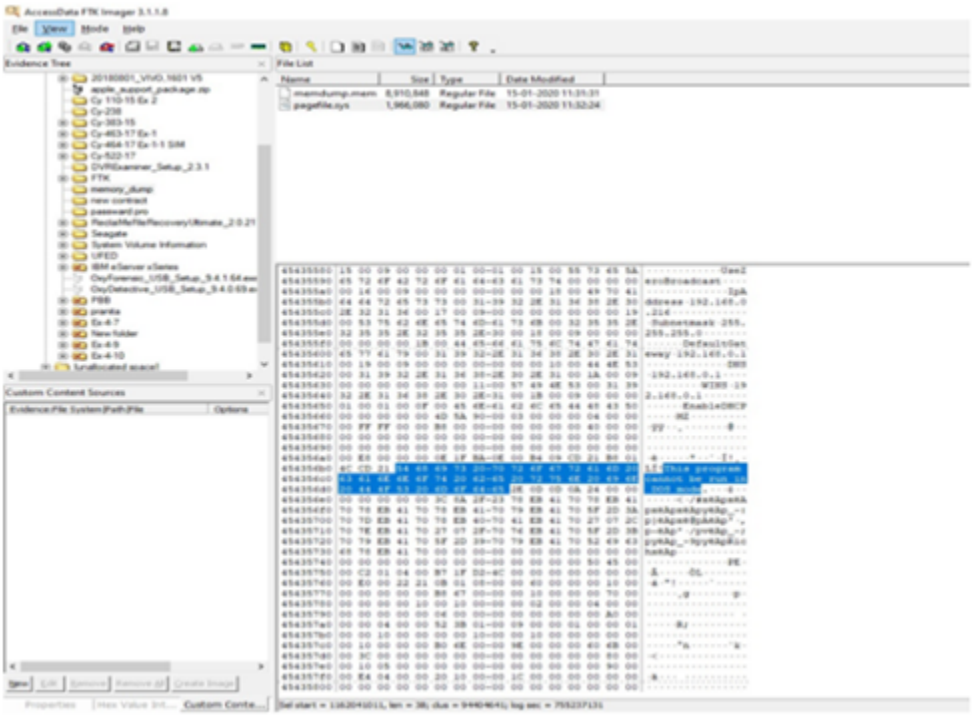


Figure 4.10: Memory dump viewed in FTK imager

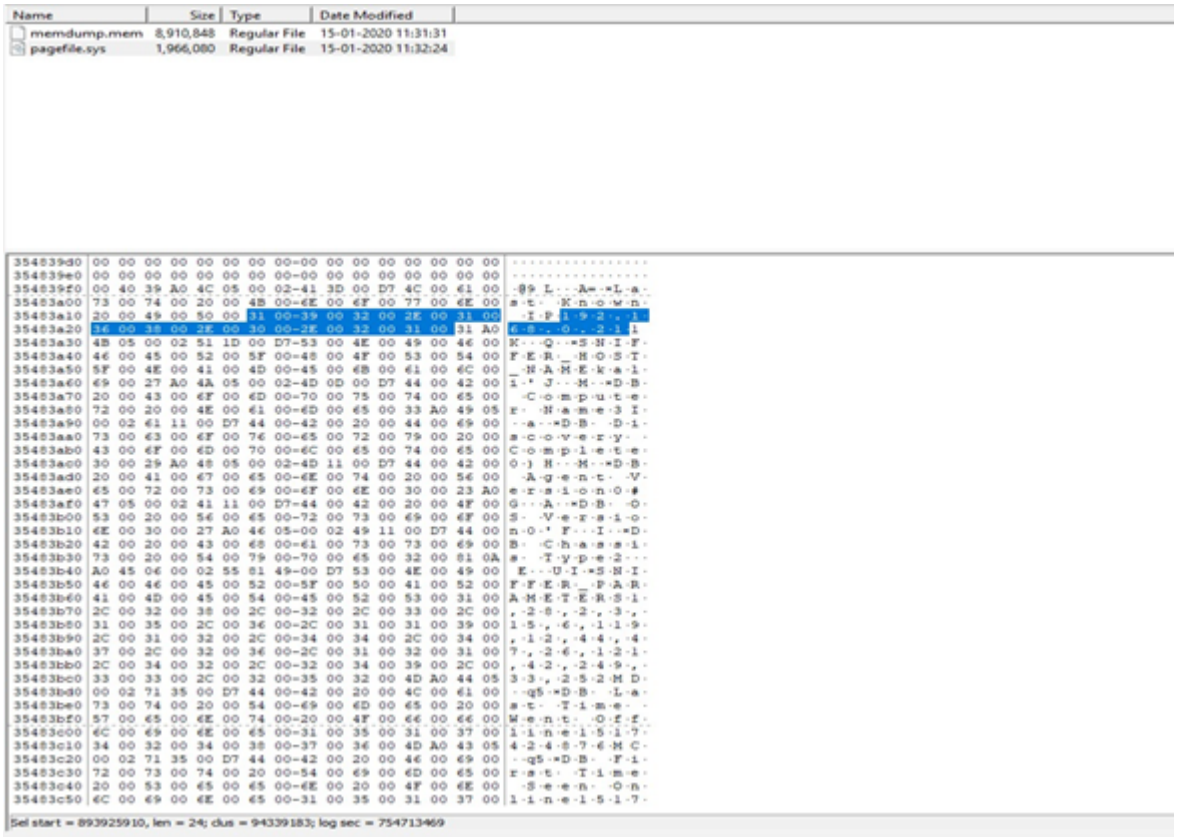


Figure 4.11: Acquired memory dump

Hash Calc is a tool for checking hash result of the collected evidence in various forms such as MD5 AND SHA256.

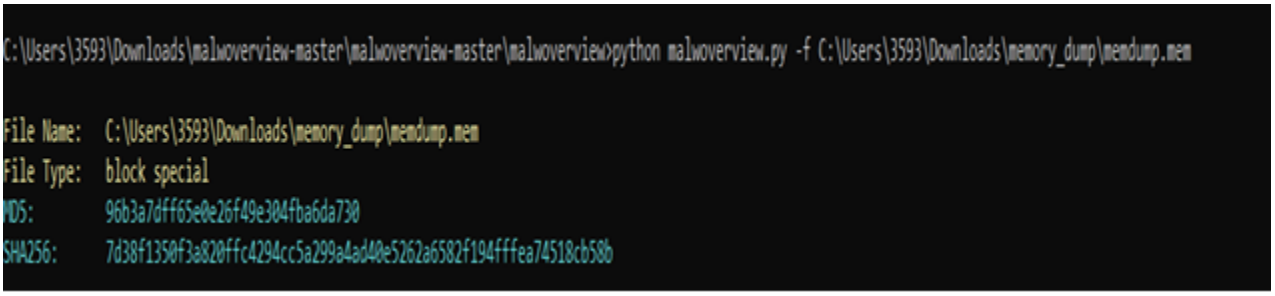


Figure 4.12: Verified Hash Value Integrity

It is clear that the memory sample is not tampered by verifying the MD5 hash value.

Volatility Framework processes RAM dumps in various formats which can be used to process crash dumps, hibernation files and, page files that may be found on dumps of storage drives. A huge amount of data can be extracted on analysing volatile memory. It includes data like p information on open files, registry handles, processes, network information and open ports, rootkits, cryptographic keys and passwords.

The files recovered from the FTK Imager capture are memdump.mem and pagefile.sys. The file sizes are 8.49 GB and 1.87GB respectively.

Analysis Results

The hash value has to be re-verified before analysis to check if the memory sample isn't tampered.

It is clear that the memory sample is not tampered by verifying the MD5 hash value. Firstly the image profile of the memory dump needs to be discovered. Image profiles are available for various operating systems such as Windows, Linux and Mac. Imageinfo command is used for the purpose.

```
┌──(root㉿kali)-[/home/kali/Desktop/Memory dump/volatility-kali-master]
└─# vol.py -f memdump.mem --profile=Win10x64_18362 imageinfo
\Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win10x64_18362
                     AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/kali/Desktop/Memory dump/volatility-kali-master/memdump.mem)
                      PAE type : No PAE
                           DTB : 0x1ad002L
                          KDBG : 0xf80540c215e0L
          Number of Processors : 4
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xfffff8053f6a5000L
                KPCR for CPU 1 : 0xffffb60160b40000L
                KPCR for CPU 2 : 0xffffb60160bdd000L
                KPCR for CPU 3 : 0xffffb60160e80000L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2020-01-15 11:28:06 UTC+0000
     Image local date and time : 2020-01-15 16:58:06 +0530
```

Figure 4.13: Memory Dump Image Profile

pslist: A command for viewing high level running processes .When executed a list of processes along with the id , threads and handles is obtained.

psscan: List of running processes can be obtained using psscan command. This plugin along with the names of processes also provides information about threads, sessions and handles. The timestamp with respect to the start of process is also displayed.

```
┌──(root㉿kali)-[/home/kali/Desktop/Memory dump/volatility-kali-master]
└─# vol.py -f memdump.mem --profile=Win10x64_18362 psscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Name                PID   PPID PDB              Time created
------------------ ---------------- ------ ------ ---------------- ------------------------------
0x0000c883b8be5080 Registry            104      4 0x00000000039b002 2020-01-15 11:24:26 UTC+0000
0x0000c883b8d4c440 cmd.exe            7144   3688 0x00000001d2d1f002 2020-01-15 11:25:01 UTC+0000
0x0000c883ba32f040 smss.exe            396      4 0x000000002036a2002 2020-01-15 11:24:34 UTC+0000
0x0000c883bae7c0c0 svchost.exe        5792    728 0x0000000007bc002 2020-01-15 11:29:01 UTC+0000
0x0000c883bdea44c0 svchost.exe         692    728 0x00000001db43f002 2020-01-15 11:27:03 UTC+0000
0x0000c883be792080 wininit.exe         656    552 0x0000000001648002 2020-01-15 11:24:56 UTC+0000
0x0000c883be799080 csrss.exe           672    648 0x0000000001a12002 2020-01-15 11:24:56 UTC+0000
0x0000c883be7de200 services.exe        728    656 0x0000001999b1c002 2020-01-15 11:24:56 UTC+0000
0x0000c883bee2a380 mySCADAservice     3168    728 0x00000001aaedd002 2020-01-15 11:24:58 UTC+0000
0x0000c883bee860c0 fontdrvhost.ex      948    656 0x0000001994e9002 2020-01-15 11:24:56 UTC+0000
0x0000c883bee883c0 svchost.exe         924    728 0x00000019b5d8002 2020-01-15 11:24:56 UTC+0000
0x0000c883bee8a2c0 fontdrvhost.ex      956    812 0x0000001994b8002 2020-01-15 11:24:56 UTC+0000
0x0000c883bef5b440 svchost.exe         488    728 0x0000001999993002 2020-01-15 11:24:57 UTC+0000
0x0000c883bef673c0 svchost.exe         652    728 0x0000001981a002 2020-01-15 11:24:57 UTC+0000
0x0000c883beff9080 svchost.exe        1112    728 0x0000001981c002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf0273c0 svchost.exe        1216    728 0x00000019a3a2002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf04f400 svchost.exe        1256    728 0x00000019e879002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf0513c0 svchost.exe        1264    728 0x00000019a3d0002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf0c7240 svchost.exe        1456    728 0x0000001f850002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf0c9480 svchost.exe        1440    728 0x0000001f966002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf0cb480 svchost.exe        1448    728 0x0000001f708002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf1e02c0 svchost.exe        1824    728 0x00000001a1726002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf1ea440 svchost.exe        1684    728 0x00000001a30fb002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf1eb080 svchost.exe        1716    728 0x00000001a3129002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf24f3c0 svchost.exe        1840    728 0x00000001a2159002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf291480 svchost.exe        1972    728 0x00000001a5075002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf2b93c0 igfxCUIService     2024    728 0x00000019e647002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf387440 svchost.exe        2100    728 0x00000001a37d1002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf3e3440 svchost.exe        2140    728 0x00000019eecf002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf43b080 WUDFHost.exe       2232    728 0x00000001a357a002 2020-01-15 11:24:57 UTC+0000
0x0000c883bf47c440 svchost.exe        2304    728 0x00000001a48c7002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf4b4440 svchost.exe        2388    728 0x00000001a6abd002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf547440 svchost.exe        2528    728 0x00000001a5746002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf55d3c0 svchost.exe        2584    728 0x00000001a7fe1002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf64c380 spoolsv.exe        2788    728 0x00000001a3f09002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf70f480 svchost.exe        3016    728 0x00000001a8d8c002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf8aa440 armsvc.exe         2148    728 0x00000001a3ff9002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf8ac400 mDNSResponder.     2476    728 0x0000001998f21002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf8ae480 svchost.exe        2724    728 0x00000001ab60c002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf8c8400 svchost.exe        3036    728 0x00000001a9e46002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf8ca100 FTActivationBo     2748    728 0x00000001ab410002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf8e1440 svchost.exe        2856    728 0x00000001aa38e002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf907480 FTSysDiagSvcHo     3088    728 0x00000001801cd002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf9544c0 mySCADAservice     3240    728 0x00000001afa05002 2020-01-15 11:24:58 UTC+0000
0x0000c883bf9cc4c0 mySCADAservice     3284    728 0x00000001adcbb002 2020-01-15 11:24:58 UTC+0000
0x0000c883bfa0a440 NmspHost.exe       3364    728 0x00000001ae753002 2020-01-15 11:24:58 UTC+0000
```

Figure 4.14: Running Process

Psxview command aids in discovering hidden processes in the dump.

```
┌──(root㉿kali)-[/home/kali/Desktop/Memory dump/volatility-kali-master]
└─# vol.py -f memdump.mem --profile=Win10x64_18362 psxview
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------- --------
0x00000001abaac400 mDNSResponder.        2476 True   True   False    True   True   True    False
0x00000001983de200 services.exe           728 True   True   False    True   True   True    False
0x00000001ae195080 conhost.exe           3812 True   True   True     True   True   True    False
0x00000001ae0ca080 myscadacom.exe        3584 True   False  True     True   True   True    False
0x00000001d5cd7080 WMIC.exe              2616 False  False  False    True   True   False   False
0x00000001ae1b8080 conhost.exe           3964 True   False  False    True   True   True    False
0x0000000217799080 svchost.exe           4844 True   False  True     True   True   True    False
0x0000000214e7a080 browser_broker        3164 True   False  False    True   True   True    False
0x00000000199a2a380 mySCADAservice       3168 True   True   True     True   True   True    False
0x0000000001996773c0 svchost.exe         1376 True   False  False    True   True   True    False
0x00000001abb3c3c0 svchost.exe           3192 True   False  True     True   True   True    False
0x00000001ae008400 openvpnserv.ex        3356 True   False  False    True   True   True    False
0x00000001ae0c9440 RdcyHost.exe          3540 True   False  True     True   True   True    False
0x00000001d181d3c0 igfxTray.exe          7600 True   False  False    True   True   True    False
0x00000001abaae480 svchost.exe           2724 True   True   True     True   True   True    False
0x00000001fe5624c0 svchost.exe           7276 True   True   False    True   True   True    False
0x00000001ae0d74c0 MsMpEng.exe           3576 True   True   True     True   True   True    False
0x00000001ae1b7480 node.exe              3924 True   False  True     True   True   True    False
0x0000000217789080 RNADirMultiple        7532 True   False  True     True   True   True    False
0x00000001d1821080 MicrosoftEdgeC        6984 True   False  True     True   True   True    False
0x00000001a4a7c440 svchost.exe           2304 True   True   False    True   True   True    False
0x00000001c9a11480 dasHost.exe           5392 True   False  False    True   True   True    False
0x00000001a4a98440 RtkAudioServic        2448 True   False  False    True   True   True    False
0x00000001d1810080 ctfmon.exe            2892 True   False  True     True   True   True    False
0x00000001d1844080 WmiPrvSE.exe          6832 True   False  True     True   True   True    False
0x00000001c9b9c480 svchost.exe           5620 True   True   True     True   True   True    False
0x000000018b55b4c0 SgrmBroker.exe        9412 True   False  False    True   True   True    False
0x00000001abb06080 svchost.exe           3108 True   False  False    True   True   True    False
0x00000001d19863c0 svchost.exe           6176 True   False  True     True   True   True    False
0x00000001abae1440 svchost.exe           2856 True   True   False    True   True   True    False
0x0000000199b673c0 svchost.exe            652 True   True   True     True   True   True    False
0x00000001ae0c8080 myalerting.exe        3608 True   False  True     True   True   True    False
0x00000001d19a6480 PresentationFo        3232 True   False  True     True   True   True    False
0x00000001fa6a60c0 SecurityHealth        7760 True   False  False    True   True   True    False
0x00000001fa7c8400 svchost.exe           9276 True   False  False    True   True   True    False
0x00000001ae1ac380 conhost.exe           3884 True   False  True     True   True   True    False
0x00000001abb404c0 mySCADAservice        3208 True   False  False    True   True   True    False
0x00000001997eb080 svchost.exe           1716 True   True   False    True   True   True    False
0x00000001c9a26480 EventClientMul        5400 True   True   False    True   True   True    False
0x0000000019839f862                       520 False  False  False    False  True   False   False
0x00000001a4105480 svchost.exe           2872 True   False  True     True   True   True    False
0x00000001d5d6c0c0 igfxEM.exe            7560 True   False  False    True   True   True    False
0x00000001d180e080 svchost.exe           6360 True   False  True     True   True   True    False
0x00000001ae10e3c0 svchost.exe           3664 True   True   False    True   True   True    False
0x00000001d5d6f080 svchost.exe           7636 True   True   True     True   True   True    False
0x00000001ae0d0080 myscadadataman        3568 True   True   False    True   True   True    False
```

Figure 4.15: Hidden Processes

Pstree: Pstree command to display the running processes along with child parent relationship. It aids in discovering some abnormal process.

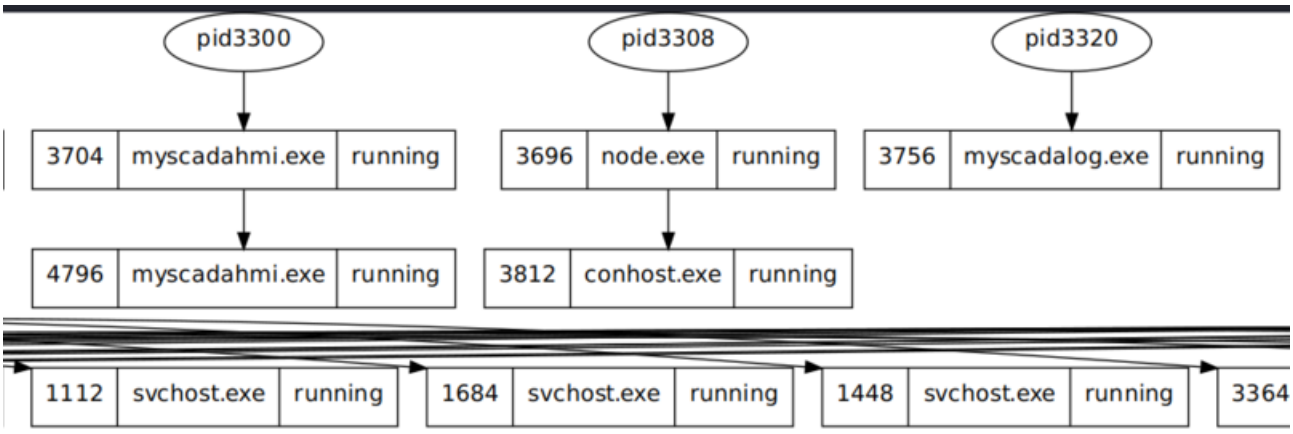


Figure 4.16: Current and Parent Processes

Netscan: Netscan discovers active IP addresses and the ports. It recovers network related artifacts from pool tag scanning. It extracts all TCP listeners, UDP listeners and endpoints. It also provides remote and local IP. It is an important command to search for an unauthorized connection.

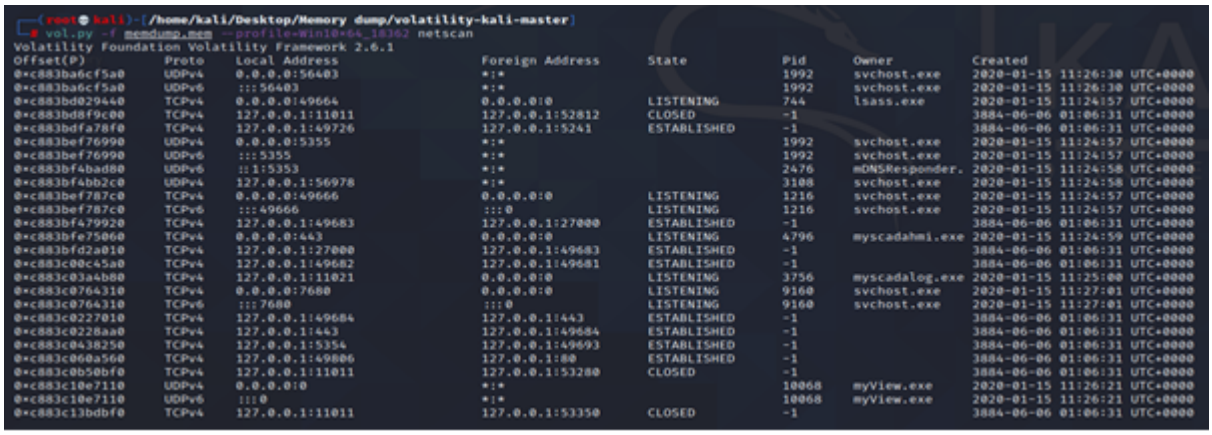
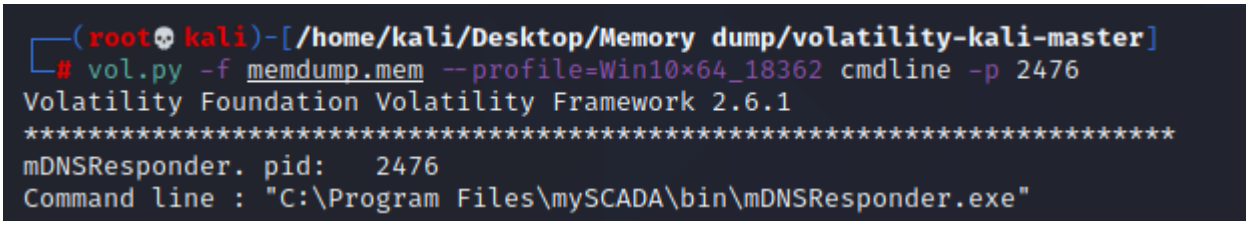


Figure 4.17: Active IP Addresses and Ports



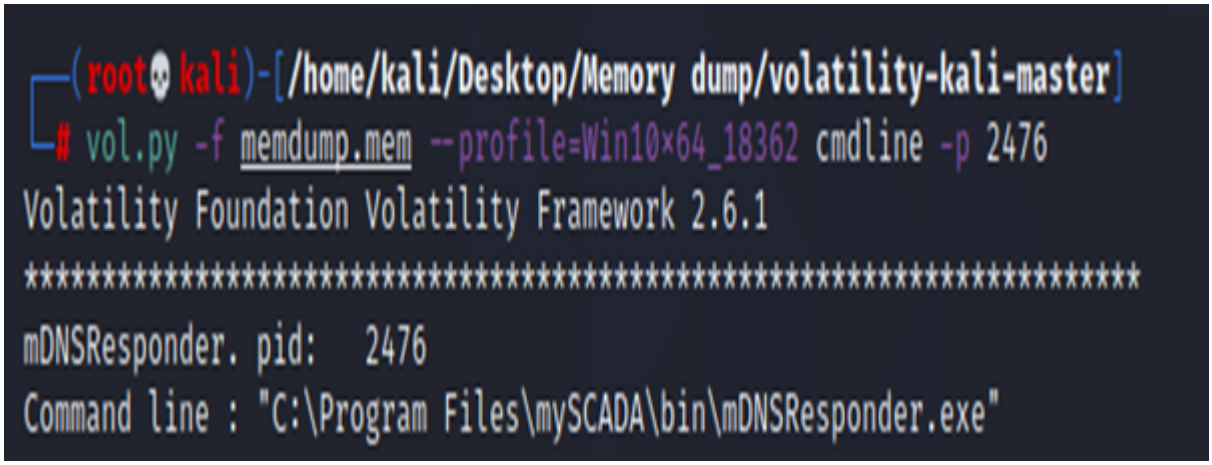


Figure 4.18: mDNS Responser Memdump

This suspicious file in above image is moved to another folder. This exe file is uploaded on Virus Total which will investigate the file using more than 70 antivirus scanners which detects if any hidden malware is present. Virus total scans the exe or dll file and checks for any suspected malware by utilizing the available plugins. Virus Total offers a number of file submission methods, including the desktop uploaders, browser extensions and a programmatic API and public web interface. The image below shows the malware present in the exe file. Even though our required files were undetected it by virus total, memory forensics using volatility could alone help in carrying out the forensic investigation. These types of suspicious exe and dll files are sent for further forensic analysis.

Table 4.1: Commands used for Analysis

Command	**Description**
Dlllist	It helps in extracting the dll files within the processes. Malware will try to modify the dll list to hide itself.
Dll dump	Used to dump dll from memory space of process to other location for analysis.
Timeliner	Presents the timeline of processes found in memory dump.
Modscan	It can pickup hidden and unloaded drivers which are hidden by the rootkits.
Filescan	Recovers file objects from the memory and can open files which contain rootkit.
Cmdscan	Looks for commands attacker might have entered through command prompt.
Connscan	It is a plugin for obtaining TCP connections of the processes.
Malfind	Malfind command gives hidden and injected code. One of the executable files that is found suspicious and is hidden.
Cmdline	This outputs the process command line arguments.

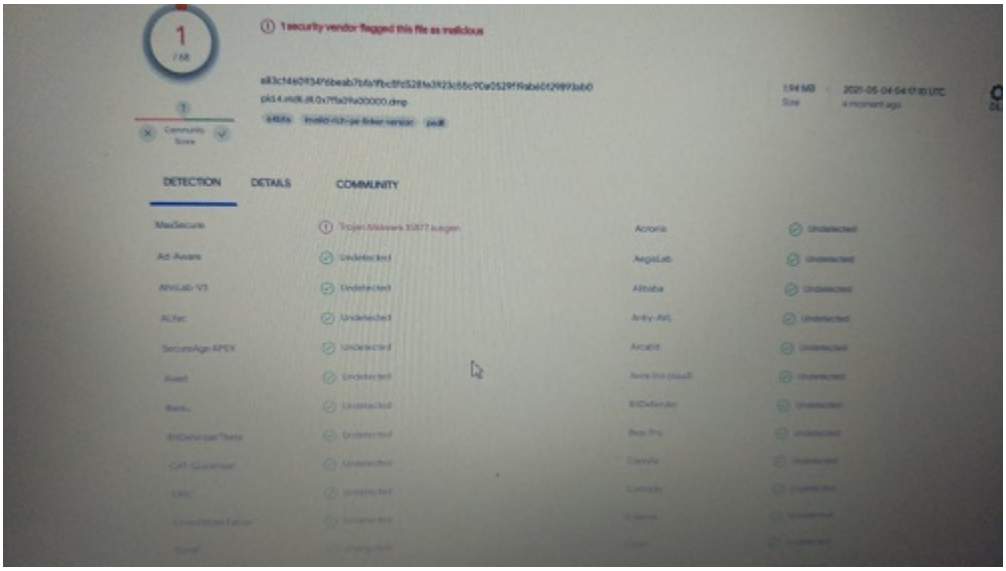


Figure 4.19: Malfind and VirusTotal

Reporting

The suspicious exe or dll files obtained using volatility will be sent for further forensic analysis.After the analysis, the information extracted in this process is sent to the concerned authorities. As mentioned, Investigation of networks for security doesn't fundamentally branch out into prosecution, but can motion a legal action like terminating job of the hacker or employee that helped in seeping into the system. The Hash value is verified after receiving using the hash calculators.

4.3.4 Machine Learning Algorithms

In this report the forensic analysis of the case study, the network forensic analysis for SCADA systems and K-Nearest Neighbours algorithm have been used over the dataset. When we see the results analytically we get greater than 99.9 percent in Binary Classifier and 95.6 percent accuracy in multi-class classifier on various attack vectors performed on the testbed with very low false positives in the confusion matrix.

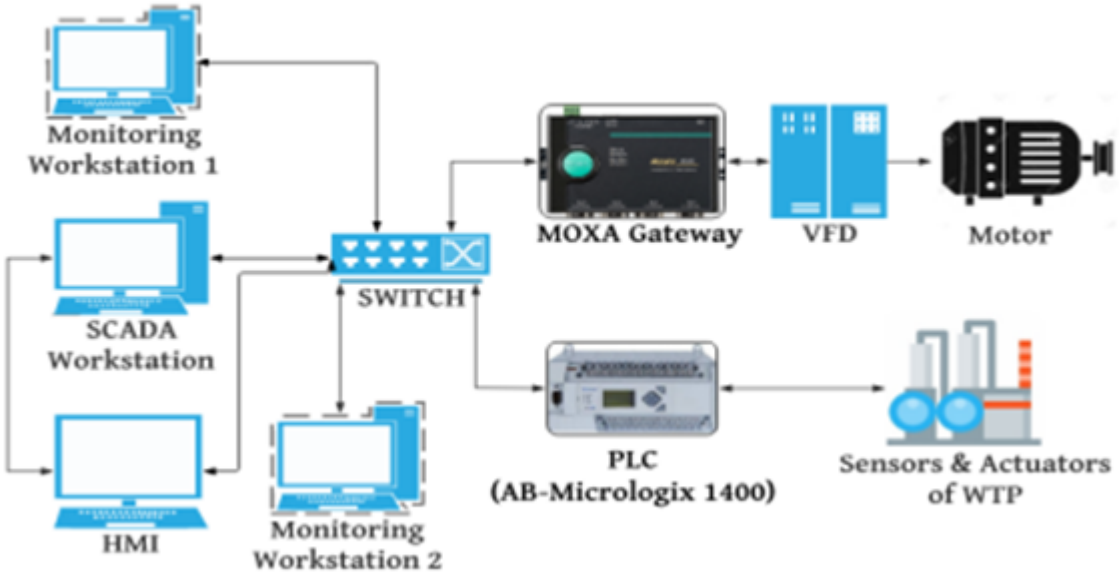


Figure 4.20: Testbed Architecture - Source:[17]

Datasets

Data is collected from the Test-bed architecture for over a few hours and then it is used as raw datasets for this project. We use the testbed in order to re-create the systems used in the real world industries. This will help us have nearly realistic cyber-attacks on the testbed. It allows to carry out cyber-attacks realistically.

COE TestBed Dataset

The IPS Dataset from [17] is used for test bed architecture of Anomaly Detection. The Data set comprises of packet attributes categorized in normal or anomalous data with type of protocol used. Total No.of records in dataset: 105150

Total No.of fields in dataset: 11

Total No.of Destination IPs in dataset: 13

Total No.of Protocols in dataset: 5

Total No.of Source IPs in dataset: 13

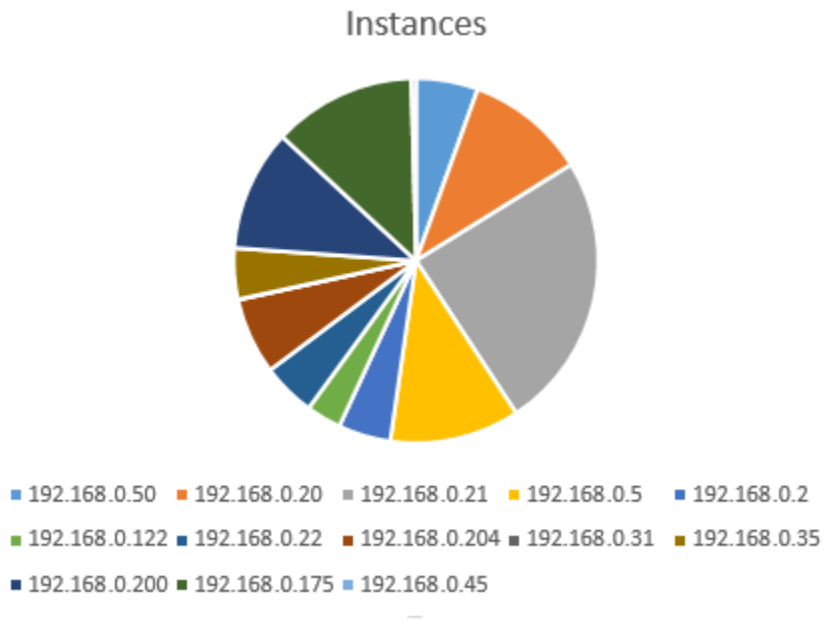


Figure 4.21: Instance Frequency for Source IP's

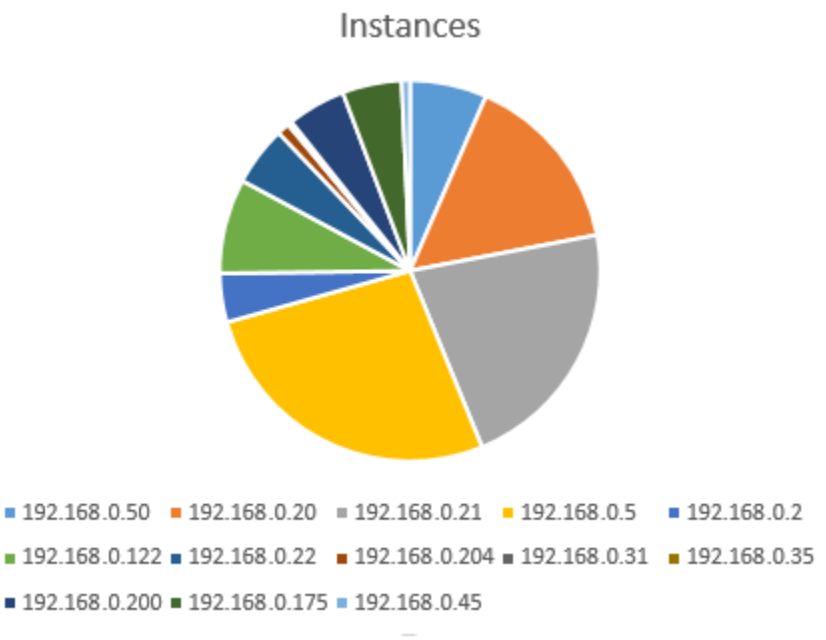


Figure 4.22: Instance Frequency for Destination IP's

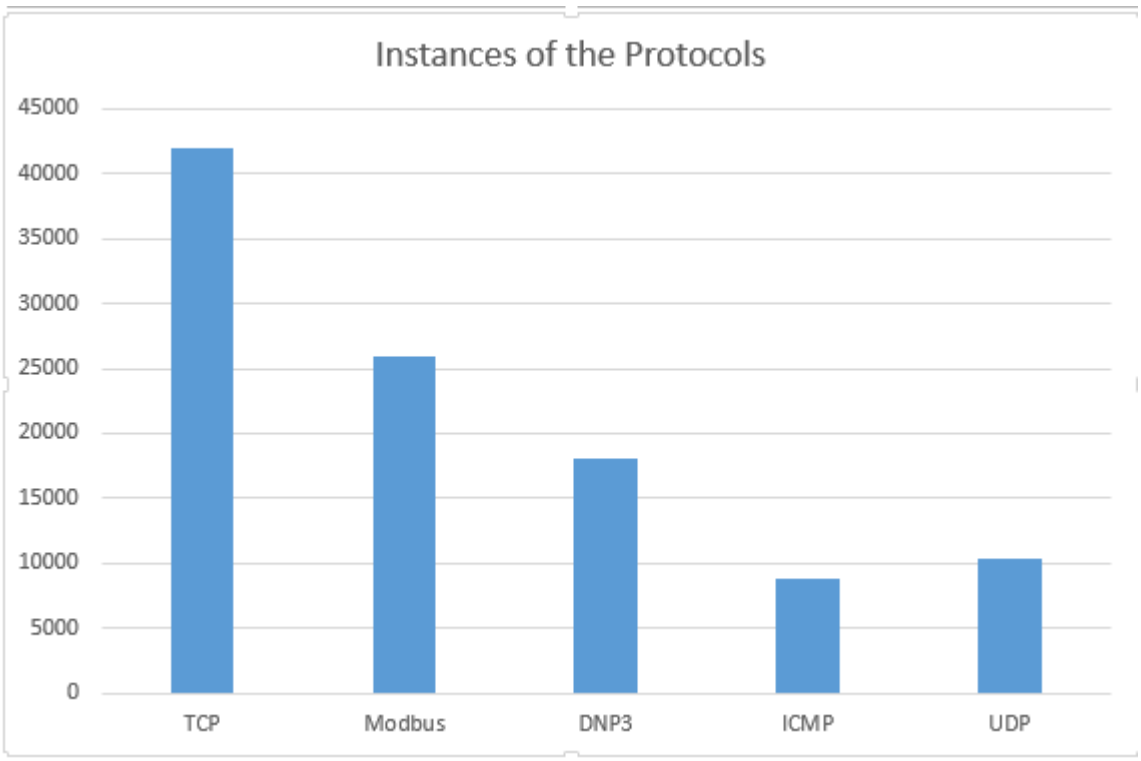


Figure 4.23: Protocol frequency in the dataset

Along with this we had seen another dataset Wustl-scada-2018. This dataset had larger set of values observed over larger time period. This was collected the Washington University in St.Louis for research purposes.

Here we focus on reconnaissance attacks from the network so that possible vulnerable areas can be used for attacks in the future.

Pre-Processing

Figure 4.24: Steps in Pre-processing - Source:[12]

Steps in Pre-Processing:

- Clean the dataset by removing the columns with object datatype(as they can't be used for modelling out algorithms) and also removing the Null values – 7 columns were removed from the dataset.
- Append the data into a single dataset such that the logs are in order.
- Select the attributes that we need – Source port, Destination Port, Length of Packet and Type of Packet
- Remove all the NaN values and replace them with suitable values (according to the column)
- Find the Unique Values possible of all the selected attributes and plot graphs(to understand distribution of data in the dataset) wherever possible.

Modelling Algorithms:

Decision Tree, Random Forest, K-Means and Linear Discriminant Analysis all are available in keras library of tensor flow and they will be used for training and testing on the IPS-dataset that was got from our lab last year.

Inspection and processing was done the dataset using the above mentioned algorithms and reasonable accuracy with high precision parameters have been seen in both binary as well as multi class classification.

The accuracy was increased by trying to factor in various ways in which parameters and the input can be varied and compare the results from different models.

- Correlation: It is a parameter or coefficient that is responsible of reading degree of difference or strengths of variables to know the relation they possess. The correlation factor ranges from -1 to 1. When the value of correlation is 1 its called perfect positive correlation. It is independent of what parameter or a variable caused those variables.

$$r = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2 (Y - \bar{Y})^2}}$$

Figure 4.25: Correlation generalised equation - Source:Science Direct

Here r is correlation coefficient,X and Y are variables while X bar and Y bar are the means of respective variables.

- Normalized and Standardized Scaler: These are the scaling measures responsible for Data transformation. Normalization is used for re scaling the respective variable range into [0,1]. While Standardization is used for scaling the range by using the mean as 0 and the standard deviation to be 1. They are usually used for scaling the features of classifiers in Machine learning programming like One class SVM Model.
- L1 Normalization: It is a method of normalization that will change the values of the dataset such that the absolute values for every row adds up to 1. Thus it's also called as Least Absolute Deviations.
- Memory required: The memory requirements of different models needs to be calculated and compared. Memory profiler is a module in python that can be used for this.
- Time for training and testing: The time required for training and testing can be measured for different to evaluate their performance.

Output

Comparative Analysis of different algorithms is very important when finally we have to select one algorithm for our dataset. The current parameters under which be are testing it are:

- Accuracy: As the name suggests it is the ratio of correct classifications upon the total number of examples present in the dataset. This is one of the main ways to compare different models.
- Precision: This gives the value of correct positives predicted to the amount of positives predicted. If this ratio is high then it states that the if 0 is predicted then it is actually 0.
- Recall: This gives us the ratio with value of correct positives to the amount of positives present in the dataset. High recall states that the class is recognized.
- F1 score: This parameter was formulated to have a quantifiable measure of precision and recall values. This is the harmonic mean of precision and recall.

Table 4.2: Binary Classification Outputs

Model	**Fitting Time**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Decision Tree	0.146918	0.999856	0.999845	0.999865	0.999856
Random Forest	7.330480	0.999688	0.999845	0.999865	0.999856
K-Nearest Neighbors	3.343110	0.999856	0.999845	0.996875	0.999856
Linear Discriminant Analysis	0.451408	0.990058	0.990867	0.990328	0.990050

The algorithm that has been proposed classifies between abnormal and normal traffic with a 99.9856 percent with very little R2 confusion(false positivity rate) indicating infinitesimally small chance of misjudging abnormal traffic as normal.

When we see the table 4.3 we can we have a recall value of 99.68 percent showing that almost all the traffic indicated as malicious by the model, is actually malicious in nature. This feature holds importance

when the size of the dataset becomes very large. Similarly the precision of 99.9845 percent dictates that the number of positives declared by the model is equal to the number of true positives in the traffic. This parameter along with the FPS helps us determine the correctness of the model.

Finally the F1 score harmonically averages the precision and recall parameter to indicate the performance of the model. The proposed model's high F1 score of 0.999856 shows that the model has very high true positivity rate and the tuning of the parameters of the model can be stopped. In a model we plot AOC curves in order to check the characteristics of the receiving operators. The proposed model has achieved an AUC-ROC of 1.0000

Once we get the results for this the same model is trained and tested for several different types of attacks giving an accuracy of 95.6 percent and all the evaluation metrics for them are listed in the table below

Table 4.3: Multi class Classification Outputs

Classes	**Precision**	**Recall**	**F1 Score**
Normal	0.94	0.94	0.93
DOS Modbus Flooding	0.91	0.98	0.94
DOS UDP Scan	0.91	0.94	0.93
MITM	0.93	1.00	0.96
Malware Stealth Scan	0.97	0.95	0.93
Ping Scan	0.97	0.98	0.97
UDP port and service scan	0.97	0.93	0.95

The first thing that strikes is the high precision with the ping scan and the UDP and service scan. This can be explained by the ping can containing only IMCP from not known sources and UDP differ from the normal traffic's behavior in the protocol data.

Second conclusion to note the relatively lower recall and precision values for DOS UDP and Modbus flooding. This occurs cause their traffic can overlap UDP port scan and regular data thus they can be mis-classified easily as different types of attacks. Finally the low precision of MITM can be contributed to the lower instances in the dataset.

Table 4.4: Output of the algorithms

Algorithms	**Accuracy**
Decision Tree	99.9856
Random Forest	99.9688
K-Means Neighbours	99.9868
Linear Discriminant Analysis	99.0058

Now we see that the maximum accuracy is achieved by K-Nearest neighbours and decision trees when it was compared to other algorithms. These models can be selected depending on the applications, however the evaluation parameters of other algorithms can be increased with changes in features, tuning hyper parameters and data set. The other parameter in selection of classification of algorithm is process time required by each algorithm to estimate current packet is normal or malicious. When the work was compared with [17] on these algorithms, the evaluation metrics have improved considerably. This was achieved by increasing the preprocessing of the data, increase the cleaning of the data, changing the encoding method from category encoder to a combination of category, binary and one hot encoder. New methods to increase accuracy such as adding a correlator and standardized and normalized scaler were added to the model.

When you increase the preprocessing the data is cleaned more than usual. This will have lower NaN values, higher connection between the possible in the dataset. The techniques like correlation and scaler helps us get a better knowledge of the dataset.Possible model-able relations like frequency of the recorded instances, relations between sending IP and the type of protocol used for transmission etc can be found to increase the efficiency. Finally changing the encoding method helps you disintegrate the outputs into further more category thus helping in increasing the evaluation metrics.

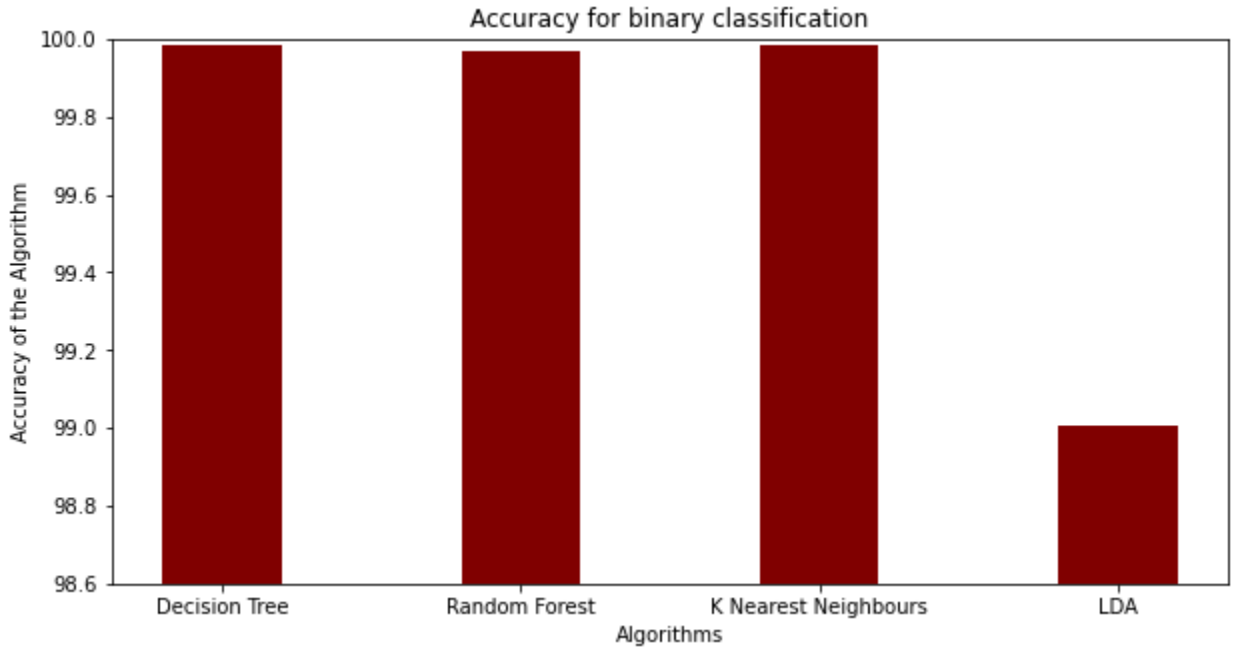


Figure 4.26: Accuracy Comparison for algorithms

4.4 Implementation

4.4.1 Hardware and Software Requirements

Software Requirements

PYTHON 3.5.2: Open source programming language use to code Behaviour based intrusion prevention system. It is the platform used for testing all our ML algorithms.

SCAPY 2.4.3 Scapy is a Python program that qualifies the user to sniff, send, dissect and forge network packets.This competency permits construction of tools that can probe, scan or attack networks.

MATPLOTLIB Matplotlib is a comprehensive library for creating animated, static and interactive visualizations in Python.

NMAP 7.5 Nmap ("Network Mapper") is a free and open source utility for network discovery and security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those endpoints offer, what type of packet filters/firewalls are in use, what operating systems (and OS versions) are being run, and other characteristics.

SEABORN Seaborn is a data visualization library of Python data visualization library contingent on matplotlib. It provides a high-level interface for drawing pleasing and didactic statistical graphics. It helps you defineunderstand your data. It's plotting functions work upon data arrays and frames constituting whole datasets and internally bringing out the requisite statistical aggregation and semantic mapping to produce instructional plots.

SKLEARN Scikit-learn provides a collection of unsupervised and supervised learning algorithms via a accordant interface in Python. It is authorized under a permissive simplified BSD license and is assigned to many Linux distributions, to stimulate academic use. It has been used to implement our IPS system.

KERAS Keras is an API delineated for humans, not computers. Keras follows practices for bringing down cognitive load: it offers congruous yet simple APIs, reducing the number of user actions requisited for common applications to a minimum, and it lays out coherent and pragmatic feedback upon user error.

CATEGORY ENCODERS A set of scikit-learn-style transformers for encoding definite variables into numeric with various procedures.

VOLATILITY v2.6 Volatility introduced people to the power of scrutinizing the runtime state of a machine wielding the data found in volatile storage (RAM). It also lays out a cross-platform, flexible, and protractile platform to stir up supplemental work into this exciting area of research.

FTK Imager v3.1.1.8 It's typically used in extracting the memory dump from the system. It is used alongside Volatility to obtain faithful memory dump analysis.

TSHARK/WIRESHARK TShark is a network protocol analyser that allows reading packets from a stored capture file or capturing packet data from a live network, or, either reproducing a decoded form of those packets to the standardized output or storing the packets to the endpoint. For our project it is used to analyse PCAP artefacts thoroughly.

Hardware Requirements

Table 4.5: Hardware Requirements

Algorithms	**Accuracy**	**Accuracy**
LINUX with Remnux-64 bit	8 Gb	64Gb
Kali Linux	8 Gb	80Gb
Kali Linux-64 bit	8 Gb	128 Gb

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

The cybersecurity is the need of the hour for SCADA systems to protect itself from intrusion and damage. Today's SCADA network have more varieties in the form of information because packets come from traditional informational and communication technology and improvised sophisticated SCADA technology. Our research begins with the reasons of sabotages in SCADA sector and highlights the importance of these growing technologies. It also provides a revised literature survey of the research challenges, technical challenges, the methodologies like Data Retrieval and Incident Response,Memory Forensics, advanced ML techniques used in prevention of attacks, Applying Existing IT Tools to SCADA, Live Data Acquisition and so on so forth.

So, the penetration testing is done best by Wireshark and Kali linux thus they have been exploited to apprehend these attacks. Analysis procedure is implemented with abnormal and normal data traffic circumstances taken into consideration.

The report studies the analysis of using Decision Trees, Random Forest, KMeans and linear separators on the dataset. The accuracy value for the Decision tree and Random forest comes out to be 99.9%, for K Nearest-Neighbourhood it comes out to be 99.6% and the linear discriminant function has the least yet adequate efficacy of 99%. But as seen during the Implementation and output of our methodology there are many different things possible to improve the efficiency and the scores for our algorithms.

Along with this we performed memory dump forensics for DLL extraction and performing memory dump analysis. It emphasis that the processing memory has the bulk of information with the executable files. When code injection takes place it occupies space of a legitimate executing process with correct protection required for the job. The injected code is so modified that commands on volatility like malfind would fail. The efficacy seems to be considerable over various cases and its future work defines investigation of APIs and create signature of API functions.

5.2 FUTURE WORK

The future work of our project has its core in the development of a streamlined system to get all our objectives done automatically, independent of the data or mode of malware injection, like memory or code. Our major setback has been in the memory forensics because the engineering workstation software seemed to be incompatible with Volatility tool and hence full exploitation of the tools were not possible, So compatibility enlargement with SCADA devices, or applying different tools so as to reap out most benefit from memory analytics forensics can be one of the major upcoming goals.

A better way of going forth is studying the intruder's behaviour and generating more detailed logs for such circumstances and developing a dynamically adaptable technique.

Also, development of more flexible and versatile ML algorithms for SCADA forensics is always appreciated which not only classifies the traffic but also forms classes of the malware injected on the system so that definite path of forensics can be laid out for that particular attack.

Bibliography

[1] T. Cruz et al., "A Cybersecurity Detection Framework for Supervisory Control and Data Acquisition Systems," in IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2236-2246, Dec. 2016, doi: 10.1109/TII.2016.2599841.

[2] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in IEEE Access, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[3] Simon Duque Anton, Suneetha Kanoor, Daniel Fraunholz, and Hans Dieter Schotten. 2018, "Evaluation of Machine Learning-based Anomaly Detection Algorithms on an Industrial Modbus/TCP Data Set," ARES 2018 International Conference on Availability, Reliability and Security, August 27– 30, 2018, Hamburg, Germany. ACM, New York, NY, USA, 9 pages. https: //doi.org/10.1145/3230833.3232818

[4] H. M. Farooq and N. M. Otaibi, "Optimal Machine Learning Algorithms for Cyber Threat Detection," 2018 UKSim-AMSS 20th International Conference on Computer Modelling and Simulation (UKSim), Cambridge, 2018, pp. 32-37, doi: 10.1109/UKSim.2018.00018.

[5] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan and R. Jain, "Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things," in IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6822-6834, Aug. 2019, doi: 10.1109/JIOT.2019.2912022.

[6] Le, Thi-Thu-Huong; Kim, Yongsu; Kim, Howon. 2019. “Network Intrusion Detection Based on Novel Feature Selection Model and Various Recurrent Neural Networks,” Applied Sciences, vol. 9, no. 7, p. 1392, Apr. 2019 [Online]. Available: http://dx.doi.org/10.3390/app9071392

[7] Y. Zhang et al., "Cyber-guided Deep Neural Network for Malicious Repository Detection in GitHub," 2020 IEEE International Conference on Knowledge Graph (ICKG), Nanjing, China, 2020, pp. 458-465, doi: 10.1109/ICBK50248.2020.00071.

[8] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain and A. Nawaz, "HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems," in IEEE Access, vol. 7, pp. 89507-89521, 2019.

[9] X. L-Novo, M. V-Barbas, V. A. Villagrá and M. Sanz Rodrigo, "Evaluation of Cybersecurity Data Set Characteristics for Their Applicability to Neural Networks Algorithms Detecting Cybersecurity Anomalies," in IEEE Access, vol. 8, pp. 9005-9014, 2020

[10] J. E. Dickerson and J. A. Dickerson, "Fuzzy network profiling for intrusion detection," PeachFuzz 2000. 19th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.00TH8500), Atlanta, GA, USA, 2000, pp. 301-306, doi: 10.1109/NAFIPS.2000.877441

[11] R. Perdisci, G. Gu and W. Lee, "Using an Ensemble of One-Class SVM Classifiers to Harden Payload-based Anomaly Detection Systems," Sixth International Conference on Data Mining (ICDM'06), Hong Kong, 2006, pp. 488-498, doi: 10.1109/ICDM.2006.165.

[12] Gu, Xiaoyi Akoglu, Leman Rinaldo, Alessandro. (2019). Statistical Analysis of Nearest Neighbor Methods for Anomaly Detection.

[13] Amer, Mennatallah Goldstein, Markus. (2012). Nearest-Neighbor and Clustering based Anomaly Detection Algorithms for RapidMiner. Proc. of the 3rd RapidMiner Community Meeting and Conferernce. 10.5455/ijavms.141.

[14] Goldstein, Markus Dengel, Andreas. (2012). Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm.

[15] Goldstein M, Uchida S (2016) A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. PLoS ONE 11(4): e0152173. https://doi.org/10.1371/journal.pone.0152173

[16] https://github.com/Amogh1809/IntrusionPreventionSystem - Intrusion Prevention System (Github)

[17] https://github.com/Akshay-Rajapkar/OT-IPS-Deep-Neural-Net - Akshay(Senior) Intrusion Detection and Prevention System (Github)

[18] Introduction to Network Forensics- Enisa Europe- Version 1.1- August 2019 – Ebook for Network Security

[19] Raj Jain, *WUSTL-IIOT-2018 Dataset for ICS (SCADA) Cybersecurity Research*, vol.1, Washington:WUSTL, 2018. [Dataset]. Available: https://www.cse.wustl.edu/-jain/iiot/index.html [Accessed: December 8, 2020].

[20] F. 1, J. 29, and J. 26, "Memory forensics and analysis USING VOLATILITY," 30-Oct-2020. [Online]. Available: https://resources.infosecinstitute.com/topic/memory-forensics-and-analysis-using-volatility/: :text=Volatility

[21] A. Salamanis, G. Margaritis, D. D. Kehagias, G. Matzoulas, and D. Tzovaras, "Identifying patterns under both normal and abnormal traffic conditions for short-term traffic prediction," Transportation Research Procedia, vol. 22, pp. 665–674, 2017.

[22] "Memory Forensics," Memory Forensics - an overview | ScienceDirect Topics. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/memory-forensics. [Accessed: 08-May-2021].

[23] S. Rahmat, Q. Niyaz, A. Y. Javaid, and W. Sun, "Normal and anomalous traffic flow pattern analysis for organizational networks," 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 2017.

[24] M. Murthaja, B. Sahayanathan, A. N. T. S. Munasinghe, D. Uthayakumar, L. Rupasinghe and A. Senarathne, "An Automated Tool for Memory Forensics," 2019 International Conference on Advancements in Computing (ICAC), 2019, pp. 1-6, doi: 10.1109/ICAC49085.2019.9103416.

[25] Rima Asmar Awad, Saeed Beztchi, Jared M. Smith, Bryan Lyles, and Stacy Prowell. 2018. Tools, Techniques, and Methodologies: A Survey of Digital Forensics for SCADA Systems. In Proceedings of ICSS 2018 at ACSAC 2018 (ICSS '18). ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.475/1234

[26] Ahmed, S. Obermeier, M. Naedele and G. G. Richard III, "SCADA Systems: Challenges for Forensic Investigators," in Computer, vol. 45, no. 12, pp. 44-51, Dec. 2012, doi: 10.1109/MC.2012.325.

[27] Werling, Jessica R., "Behavioral Profiling of SCADA Network Traffic using Machine Learning Algorithms" (2014). Theses and Dissertations. 634. https://scholar.afit.edu/etd/634

[28] A. Srivastava and J. H. Jones, "Detecting code injection by cross-validating stack and VAD information in windows physical memory," 2017 IEEE Conference on Open Systems (ICOS), 2017, pp. 83-89, doi: 10.1109/ICOS.2017.8280279.

[29] H. Hilal and A. Nangim, "Network security analysis SCADA system automation on industrial process," 2017 International Conference on Broadband Communication, Wireless Sensors and Powering (BCWSP), 2017, pp. 1-6, doi: 10.1109/BCWSP.2017.8272569.

[30] S. Krishnan and M. Wei, "SCADA Testbed for Vulnerability Assessments, Penetration Testing and Incident Forensics," 2019 7th International Symposium on Digital Forensics and Security (ISDFS), 2019, pp. 1-6, doi: 10.1109/ISDFS.2019.8757543.

[31] M. S. Zareen, A. Waqar and B. Aslam, "Digital forensics: Latest challenges and response," 2013 2nd National Conference on Information Assurance (NCIA), 2013, pp. 21-29, doi: 10.1109/NCIA.2013.6725320.

[32] A. Almalawi, A. Fahad, Z. Tari, A. Alamri, R. AlGhamdi and A. Y. Zomaya, "An Efficient Data-Driven Clustering Technique to Detect Attacks in SCADA Systems," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 5, pp. 893-906, May 2016, doi: 10.1109/TIFS.2015.2512522.