National College of Ireland

Threat Hunting Using a Machine Learning Approach

MSc Internship
Cyber Security

Yash Shukla
Student ID: x18175104

School of Computing
National College of Ireland

Supervisor: Mr.Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Yash Shukla
Student ID:	x18175104
Programme:	Cyber Security
Year:	2020
Module:	MSc Internship
Supervisor:	Mr.Vikas Sahni
Submission Due Date:	17/08/2020
Project Title:	Threat Hunting Using a Machine Learning Approach
Word Count:	6293
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:	Yash Shukla
Date:	14th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

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

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

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

Threat Hunting Using a Machine Learning Approach

Yash Shukla
x18175104

Abstract

The past few years have witnessed an increase in data breaches and attacks that leverage infrastructure misconfigurations. As a result, companies have to bear huge amounts of financial loss. In some cases, business information and personally identifiable information is also compromised. These threats can be detected at an early stage using proactive defence approaches with the help of experienced security practitioners. Traditionally, this involves manual analysis of logs and pcap files. Threat hunting reveals the adversaries as soon as they initiate an attack on an infrastructure. But network devices generate a vast number of logs, thus to analyse them it becomes a tedious task. To tackle this problem, this process can be automated by incorporating machine learning algorithms. This research has been conducted by applying machine learning algorithms such as Naive Bayes, SVM,Logistic Regression along with ensemble techniques like voting classifier, which can identify threats from log files based on conditions related to attacks and business logic. The accuracy rate of the experiments conducted is higher compared to traditional approaches. It was observed that machine learning can achieve higher accuracy in a limited time frame.

1 Introduction

The past several years have indicated a rise in cyber attacks on numerous industries. These attacks have had a long-lasting effect on consumers and their infrastructure. Organizations use an array of network devices and gateways like Switches, Demilitarized Zones, Intrusion Detection/Prevention Systems and Advanced Firewalls. Intruders can still pose a significant challenge for the organization by using advanced methods like encoding packets and leveraging the use of APT's along with learning about the internal workings of the organization. Several organizations that deploy customer information/finance-based systems pose the risk of exposure in the form of credentials and personally identifiable information. The systems deployed are not able to cope up with all of the latest intrusion techniques that are used, this is evident by discovering the following attacks.

- The Naikon APT was active in Asia Pacific till 2015, but it was recently discovered that the APT has been active since then using a different backdoor called "Aaria Body". This allowed the APT to run for 5 years till May 2020 without being detected [1].
- There has been a report of web skimming using an e-commerce platform on June 25 2020 ,where hackers have embedded JavaScript in exif data using stenography [2].

- On July 24th and 25th 2020 the smartwatch network of Garmin was shut down after it was attacked by a ransomware [3].
- On July 27th 2020 it was found that several Network Attached Storage(NAS) devices were inserted with malware which has previously stolen data from over 62,000 devices [4].
- The Taj Mahal APT remained undetected till 10^{th} April 2019, it was found when a diplomatic organizations system was infected with the APT. It is a highly modular malware that can have plugins attached to it for more destruction [5].
- In a recent security misconfiguration event on June 1st 2020 the Joomla Resource Directory registered users details were leaked as an unencrypted copy of the backup was kept on their private amazon S3 bucket. Most of the information leak was public data as the website is a directory for JOOMLA developers , but passwords and Ip addresses were private [6].

There has been a drastic change on the part of cyber defenders, in some cases they perform red team and blue team exercises to check whether their network is secure. Often a Security Operations Centre is used as a security barrier to several organizations. SOC teams have experts in particular fields of detection and incident response that make them efficient in nature. In some cases where the organization has no security team, they outsource the security to a third-party vendor. In order to make all of the infrastructures secure, the security teams work proactively on the network infrastructure to hunt for threats.

1.1 Reasons Threat Hunting is essential

- Threat hunting reduces the time required to perform forensics investigations; as log analysis is generally performed after the event is mitigated.
- Threat hunting helps organizations establish a better cyber threat detection and mitigation cycle, as it proactively works against threats and saves time.
- It has the capability to ensure that the organization have proper security enforced and SOC is always on the lookout for newer advanced threats.
- Threat hunting reduces the rate of false positives, which is one of the major issues that can be seen by various network and security devices.
- It makes organization develop a threat maturity model, which encompasses their threat levels to various attacks.
- Threat hunting ensures that the security team's matrix for threat detection is adapting to the ever-changing infrastructure of the company.

1.2 Notification of a personal data breach in the EU

In accordance with GDPR which has come into effect since May 25 2018, there is a need to report the breach of personal data within 72 hours. There are significant penalties imposed by GDPR, in case of serious infringements there is a penalty of 4 % of global turnover or 20million whichever is higher [7].Routine threat hunts can significantly reduce the costs of data breach and help an organization identify attacks rapidly.

1.3 Objective

Even though threat hunting is crucial, it is a tedious task that needs people with experience to identify threats from millions of logs. At an enterprise level, threat hunting teams need to go through each every log sequentially. They have to investigate on each log which further needs to be classified into a malicious or normal log. This task can be made simpler by incorporating machine learning techniques to speedup the process. This study focuses on applying machine learning algorithms to detect threats in logs based on analysis of logs.

1.4 Research Question

Will supervised machine learning improve the accuracy of threat hunting in comparison with the traditional approaches of threat detection?

2 Related Work

This section primarily reviews previous research that is based on machine learning techniques, followed by approaches in threat hunting and studies that have used pcap files for threat hunting.

2.1 Use of Machine Learning in Cyber Security

The fusion of machine learning and cyber security has been pivotal in detecting obscure threats. Machine Learning for Survival Analysis is a research based on the notion that analysis of data is to be done till an event that is of interest occurs. The authors describe the statistical methods like Non-parametric, Parametric and Semi Parametric and discusses the machine learning models like Survival Trees, Bayesian Methods, Support Vector Machines, and advanced machine learning models like Ensemble, Active learning [8]. This is important in our research as we have used this as a comparison matrix for the models to be applied. The next step was selecting supervised machine learning as a way for implementing prediction. The paper also offers insight on boosting and bagging which are types of ensemble methods. Following is a research on anomaly detection that shows how a certain technique is efficient for detection of threats. They have used Pcap files as their baseline along with Bro log reader [9]for extraction of the features. This eliminates need of a manual condition; the program is dependent upon bro log reader for tracing the appropriate features. It converts the text data to numeric data to use in the iForest algorithm for classification purposes. The hunt would be successful but there no chance for customizing the hunt according to the change of needs. The entire dataset needs to retrain for every new hunt that needs to be done for an organization. The proposed algorithms performance is measured using the ROC curve. Using this research, we were able to determine what to extract from a PCAP file [10].

2.2 Models for Threat hunting

Several threat hunting models have been researched over the years, one such model is the diamond model. It is a novel intrusion model proposed in 2013 which states that, for every intrusion there is an adversary that inches on towards the target set by him using

his own ability over any infrastructure. The diamond model is explained in four steps: Adversary, Infrastructure , Victim and Capability [11]. This model provides a different insight into threat intelligence and TTP for threat hunting. It is also used to detect data sources that might help in the forensics investigation

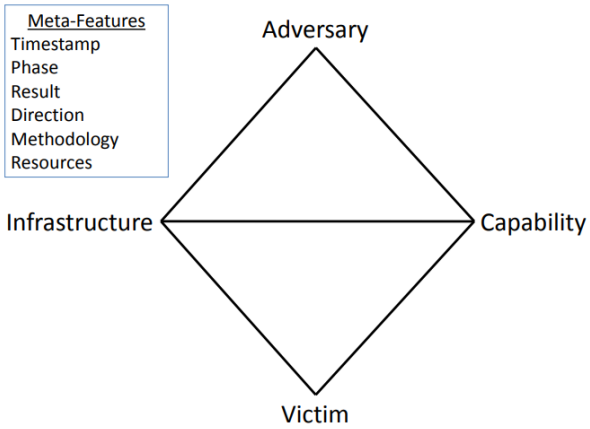

Figure 1: The Diamond Model for Threat Hunting

While researching it was found that SANS proposed a secure threat hunt model in six stages that outlines the following steps [12]:

1. Purpose: This focuses on where the hunt occurs, what to look for in the hunt and the outcome.

2. Scope: The scope stage specifies the plan for the hunt and what information will be needed. The scope model is divided into two parts, in the first part it identifies with system under test selection and the second part develops a hypothesis.

3. Equip: This stage discusses what tools are to be used for the hunt.

4. Plan Review: In this stage the actual results are compared to the expected results. If the purpose of the hunt is not fulfilled gaps are identified for the next process of the hunt.

5. Execute: In this stage the actual hunt for the threats is carried out on the data analyzed. The findings are reported in the hunt summary report.

6. Feedback: In this stage what needs to be improvised in the entire process model is discussed.

The SANS model provided a baseline for development of hypothesis based on attack. This research encompasses information about how the GRR framework can be applied to a certain test environment for threat hunting. The GRR is an open source live forensics framework capable to perform a wide array of tasks right from collection of logs to checking the registry of various clients. There are three types of experiments that were hypothesized in this paper. There were three hunts performed using the same setup the GRR Memory Scan, the GRR Registry Finder Hunt and the Network Status Hunt [13]. All of these were successful and were able to detect the threats. All the threats detected

were configured to check for the Indicators of Compromise after a certain duration, but this will increase the amount of data that is collected per node. All this collected data has to be manually checked using the GRR framework, automating the process of parsing through the data can provide better results. Also, the GRR platform does not allow to export large scale data which makes it difficult to scrutinize a certain event further on.

While reading about the CARVE approach, it was observed that the authors present a five-step approach for threat hunting. This model uses hypothesis testing and find a relationship between all the components present during the threat hunt. The CARVE method is a five-stage model which consist of Collect, Analyse, Relate, Validate, Establish [14]. This method is solely dependent on the hypothesis that is created, it verifies the model by performing a certain set of experiments. Also, from a business perspective the hypothesis established by the method would not be tangible to the threats to the business. This model uses Indicators of Compromise and Indicators of Attack as metrics.

Another research proposed by Ibrahim Ghafir and Vaclav Prenosil illustrates the usage of an experimental setup using the BRO intrusion detection(now ZEEk) [9]. This research describes a method that uses blacklisting as a mechanism. The method takes into consideration malicious domain name service traffic discovered by the system along with previous intelligence gathered about malicious domains [15]. Using a combination of these methods it curates a blacklist that can be used to thwart the efforts of the intruders.

The Design of Cyber Threat Hunting Games is a research that highlighted the difference between threat hunting, threat detection, cyber Defense, and forensics [16]. The research uses scenario-based experimentation to perform periodic threat hunts. At first baselines of network flow and consumers is taken, which is then used to write conditions for the threat hunting. Further Activity logs are compared to the baselines that were taken in the previous step; this mechanism allows the threat detection based on the systems modifications

In this paper the authors have shown that how anomaly detection systems used for network intrusion detection are not accurate, they have also highlighted that the semantic gap and different types of network make the process of detecting every single anomaly. This semantic gap can be mitigated by using machine learning. [17].

2.3 Use of PCAP Files for Threat hunting

While reading about pcap files a research exhibited the variants of the pcap file. In this research the pcap files are classified according to their extension type. It is explained that how the merging of pcap influences the output. The research compares LibPcap, PCAPpng and MS NetMon based on the structure and working. The experiment in this paper shows that during the merging of pcap files a time order issue is observed [18]. This disrupts the timeline that is used by threat analysts, also this alters the results that are computed from the pcap file. The authors have proposed a system that resolves this issue and performance testing is done on this new algorithm. The implementation seen in this experiment can be pivotal in our experiment, as there would be a situation where pcap files must be merged.

The authors Robert M. Lee and Rob T. Lee have summarized several threat hunting related statistics such as how frequently the organization carries out a threat hunt, what would be the trigger for starting a threat hunt in an organization. The paper also demonstrates that what backgrounds does a threat hunting team need along with their percentages. It also has several surveys about Critical baselines and DFIR skills required

in threat hunting [19] [20].

While researching on attacks a research showcased how frequent Pattern Mining can be used Ransomware based Threat Hunting. It also talks about a mehtod where threat Intelligence along with repeated pattern mining can be used to find out ransomware threats in a hunting scenario. It has a decisive say on a certain program on whether it is a ransomware by means of running it for the first ten seconds. It uses this data it can find out whether the given program is malicious or not [21]. To write a hypothesis based for attacks this approach can be used, we can identify whether any event names or previously used names are present in the packet. This will predict the packet as a malicious or normal.

An interesting research by Anthony Paris, Jr explains how threat hunting is being used by all defense departments around the globe [22]. It explains that how threat hunting can be formulated to form an active defense, the best way to form a defense from threats that might bypass the security devices is to actively find them via different methods. The research describes that the clients must atleast have endpoint detection along with firewalls and other security measures. The authors describe a situation where the hackers have already made their way into the network, if there are no set of checks beyond a certain zone the infiltrators can have access to the network unrestricted. To avoid such incidents, it is advised to carry out threat hunts.

3 Research Methodology

This experiment has been developed after consideration of various SDLC models [23] . The secure spiral model was selected as a base model for development of this research as it takes into consideration risk before development at every stage. It works in an iterative manner where security risk is identified before the prototype is developed. This is essential for the research as it will be developed to hunt threats in critical environments. The task of threat hunting is essential, business risk and security should be at the utmost priority.

For the implementation of the risk structure in our research, at first columns which have no Personally Identifiable Data are taken into consideration for finding out the threats. This is possible due the nature of mixed traffic data present in pcap files. The secure spiral model [24] is implemented as follows.

1. Requirement Gathering - The programming language needs to decide upon taking various factors into consideration, python is selected as the language as it supports machine learning and is secure. The second requirement would be a dataset source which was taken from a publicly available resource known as Netresec [25].The dataset used in this project is 4cis_pcap downloaded from Netresec and contains 2,00,000 rows of data. Due to the hardware restrictions the maximum number of rows used is 40000.

2. Security Risk Analysis - Previous research revealed that there is always a possibility that a threat may always be in action. Thus, security risk analysis gives us insight into how the application after development may be vulnerable to various bugs. Abuse cases are used in some scenarios to identify the security risks. To help mitigate the attacks we have used python. The python file will be tokenized to a

.pyc file and then further encrypted for deployment purposes [1].

3. Prototyping- In this step there are several aspects to be taken into consideration:

   (a) Hypothesis- Here several hypotheses are made based on the previous intrusions that might have occurred in the organization or based on the business logic developed. The above options can be used in combination to get a better hunting matrix, which will make for a better training dataset. All these hypotheses over the years can make up an efficient training dataset for next data to be processed.

   (b) Model Selection- In this step various statistical models were taken into consideration like non-parametric methods, semi-parametric methods, parametric for using the basic regression approach. But the results were not relevant to the mechanism needed for prediction. The next step was to use machine learning techniques for either classification or prediction mechanism. The project demands that the threats should be predicted based on previous hunting hypothesis relevant to the organization [26]. Using both mechanisms a code was to be written using appropriate coding standards. For this purpose, the various models as proposed by Intel, Microsoft and McAfee were studied and then the application was developed using a combination of these models to minimize threats [27].

4. Benchmarking- For finding out how the model performed against previous models, it was evaluated against other methods. The performance metrics of machine learning are based majorly on Classification Accuracy, Confusion Matrix, Area Under Curve, F1 Score, and various kinds of errors [28] [29]. There are several measures like Sensitivity, Specificity, Precision, Type 1, or Type 2 error which need to be calculated for benchmarking the algorithm which is implemented. The algorithm will be compared to historic methods which have used in the past for threat hunting. The manual methods cannot be bench-marked using the same approximation methods, we need to use manual testing methods for checking how accurately threats are found [30] [31].

5. Verification Validation- To validate whether a threat has been found according to said hypothesis, we need to check whether the label given by the prediction mechanism is correct according to the said hypothesis. The verification can be done by manually checking against the line against the condition. We can use a regex code for checking whether the condition worked, this will be manually edited whenever a check has to be done.

6. Integration with current infrastructure- Integration of the code is easily possible in any infrastructure; it can be deployed either after a checkpoint or firewall which have a secure mechanism for packet checking. After this the devices logs will be generated. These logs can be read or exported in the form of a pcap file. In a single day, all the security devices in an enterprise will create several million lines of log files. To check all these files manually and find out threats is called threat hunting. This approach can be automated by using as hypothesis-based approach. If there are dedicated threat hunters across the infrastructure who are aware of the

[1] PCAP Source :https://www.netresec.com/?page=PcapFiles

positioning of the infrastructure and business logic, it would be easy to manipulate the hypothesis for threat search every time a hunt would be conducted [32] [33].

3.1 Using Pcap files :

A pcap file is used to capture network data on a packet level which helps in the analysis of the data. There are several kinds of pcap versions like Libpcap, WinPcap, PCAPng, Npcap which provide real time information and point towards the root cause of the problems. Pcap files have compatibility with almost all kinds of operating systems. The Pcap file has several columns which makes multiple options available for feature selection. The features make analysis easier for security teams. They provide resolution of complex security problems that might arise from the network.

- The PCAP files help eliminate false positives that might arise from the Intrusion Detection and Prevention systems.
- They help see through attacks that are encoded well to bypass the firewalls or other intrusion detection systems.
- It provides an insight into new attack vectors that might not be coded into previous IDS rules, this gives us an opportunity for threat hunting as well.

3.2 Indicators of Compromise

There a specific set of skillsets used to infiltrate into enterprise level networks or applications. It is important to know how a person can infiltrate a network, as the method that is used by the person is effectively a vulnerability in the ecosystem. We will explain with an example why a hypothesis can be effectively written on each of these tactics demonstrated . Some of these infiltration and attack tactics are described by Mitre as follows [34].

- TA0001 Initial Access: One of the vectors to gain initial foothold in the enterprise network is phishing. It is one of the most common attacks and can be detected using URL detection. We can implement a condition like
  (if len(link) $<$ 20 ; malicious else normal)
- TA0002 Execution : Malicious files executions are executed on all major platforms like Windows MAC and Linux. The detection technique we can implement to find such files is check the extension of the file and provide execution prevention on secure computers [35]. Condition :
  if $(extra == df.str.match(pat = `(.docx)\|(.nameofextensions)')setmalicious)$
- TA0003 Persistence : The adversary now tries to maintain foothold in the enterprise, a very simple method is to change the cron utility scheduling so that it executes malicious code. Check if cron file is accessed [36]. Monitor task creation for communication outside the network from an unknown file using condition:
  $if(`link' == df.str.match(pat = (Skidmap)\|(attackname\|cron)))$
- TA0005 Defense Evasion: These are techniques used by adversaries to avoid detection , using obfuscated scripts for masquerading their trail. The set GUID and UID can be checked from the network when the remote network asks for files that are

capable to set these ID's [37]. Logic dependent on filename
$(if`extra == df.str.match(pat == `(filename1)|(filename2))'$

- TA0006 Credential Access: Credential access is gaining access of critical user related information, this information once obtained can be used to get legitimate access. Automated brute force tools have a low timeframe and try multiple credentials [38]. We can write a time-based threat hunt like $(if`time' == (11:12:00)|(11:18:00)); check for further conditions.$

- TA0007 Discovery : The adversary tries to find out the connections that are accessible to the breached machine. Will check for cloud-based services like Azure and look for the Active Directory [39]. Threat Hunts can be conducted using inward traffic to the AD using business logic conditions.

- TA0008 Lateral Movement: Lateral movement can be traced in the network environment when an untimely request is sent from a legitimate node to access a certain other network node [40]. Thus, a time based, and node access can be checked for threat hunting lateral movement.
if 'time' == '(11:12:00)—(11:18:00)' && ip == range('192.168.0.224-234')

- TA0009 Collection : This stage the attacker will gather the critical information, this information is then exfiltrated outside of the network [41]. This is our threat hunt vector or condition, simply check for data packets that are large and have an untimely entry which means after enterprise work hours.
if ('time' == '(11:12:00)—(06:18:00)' && ip == range('192.168.0.100-148') && size > 4000)

- TA0011 Command and Control: The attackers try to maintain control via the command and control ,they also go to extremes such that they monitor the network for normal traffic and mimic the same for keeping the network and avoiding being detected [42]. Check for DNS protocol
if (ip == range(192.168.0.224-234) && Protocol_Used == 'DNS')

- TA0010 Exfiltration : This is the process of stealing data from the network. Monitor process file access patterns and network behaviour [43]. Unrecognized processes or scripts that appear to be traversing file systems and sending network traffic may be suspicious. We can monitor network behaviour set protocols that send files out of the network like SFTP, FTP. if (ip == range('192.168.0.224-234') && Protocol_Used == 'FTP')

4 Design Specification

The traditional approach for threat hunting involves the following steps [44] [45]:

- Select whether threat hunting is to be done, make a team of threat hunters.

- Give architectural knowledge to the threat hunters.

- Construct a hypothesis based on related threats.

- Perform experiments in series to confirm or deny threats that might be present.

- Select a matrix for evaluation of the results and analyse the results. Reporting of the threats making threat hunts more efficient.

After studying this basic model it was seen that the time needed for finding the conditions that satisfy the hypothesis in a certain time frame would be difficult, as there are several millions of line of pcap files to be analyzed. To overcome this issue machine learning can be used, once a certain model is trained to check for specific condition, the lines of threat conditions will be easily identified and can be analyzed further by threat hunters. The analysis of code to verify against a certain hypothesis would be time consuming and might provide several false positive cases. It is important that the number of false positive cases is reduced so as the threat hunters can focus on finding relevant threats which may cause damage to the infrastructure or business.

4.1 Design of pcap conversion

In this research, at first the network devices were analyzed and all the devices were checked whether their logs can be used a mechanism for threat hunting [46] [47]. Also, it was checked whether the devices were capable of traffic capture. When the devices are capable of traffic capture, the pcap files are exported from their interface. All these network devices are critical for security infrastructure as they keep a log of every event that happens. Conversion of these logs would be possible using Tshark command Line Interface utility [49].

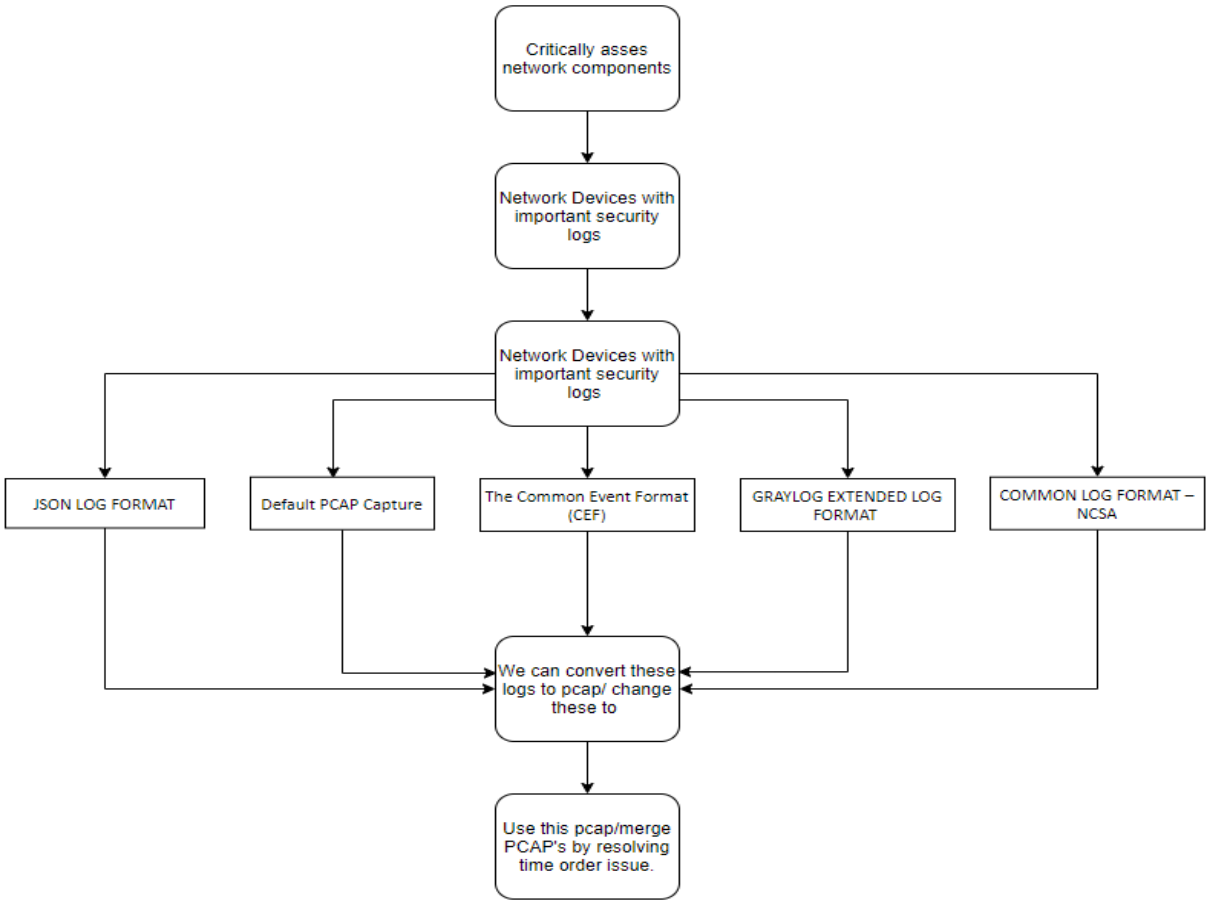

Figure 2: Selection of Pcap files

The second option is the usage of exported pcap files. There is another option to merge the pcap files, which would be necessary in situations where information from multiple network devices have to analyzed. We need to consider the time order issue, as when multiple pcap files are merged there might be problems due to similar time records in the two files. The log that has to be chronologically in order should be recorded first in the new file. After the resolution for time-based problem is done it can give a more appropriate relation of the threats present and the hunt hypothesis will be written further. Several event/log formats were taken into consideration and researched upon to reach upon the conclusion of using the PCAP files. The PCAP files contain packet level information which pinpoint the exact source from where the threat might have originated.

4.2 Design of the system

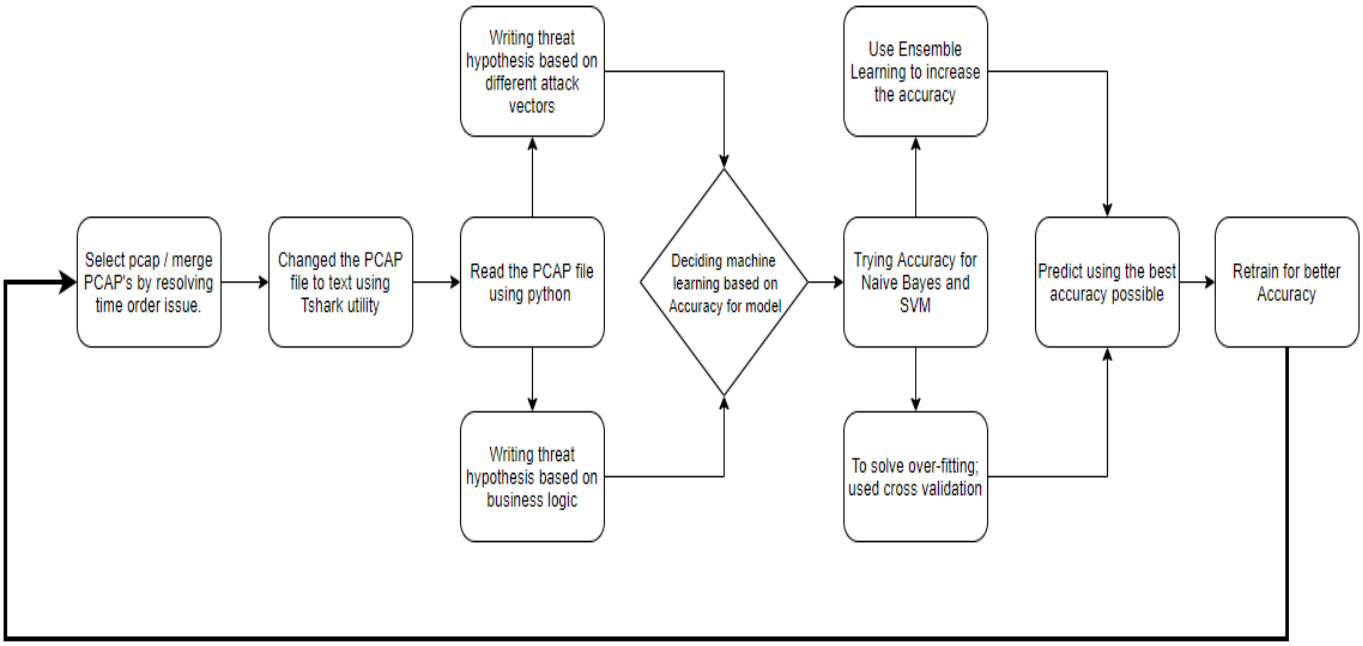

Figure 3: Process for Threat hunting using Machine Learning

The above image represents how the pcap file is transformed and used to find threats based on various factors like attacks or business logic. At first the Pcap file from the previous stage is taken and pre-processed using the tshark utility. This gets the file readily available in python code. Further the algorithms are selected using the accuracy method. Hypothesis building is one of the most important aspects that needs to be considered in this model as, the conditions decided during the hypothesis will decide what the hunt will be based on. The data is trained and accuracy for each model is checked. This points us towards using a better approach like ensemble learning or cross validation[50]. Keep on retraining the dataset on different sets of data for better accuracy. In case of a simple or small implementation of data there might be an issue of overfitting which means that the data sample is small in nature.

5 Implementation

Step 1: Data Acquisition: In this project, an existing PCAP file was selected which has been taken from a third party open source website[51]. This gave an insight on how actual files might be in a black box environment.

Step 2: : Pre-processing of PCAP: The first step was to extract the contents of the pcap file into a text or CSV format which can be used for processing and analysis. A Command Line Utility based on the Wireshark model called TSHARK was used for this purpose. The command for converting a pcap file to text file is :

```
tshark -r /PATH/Filename. Pcap -t ad > /PATH/ Filename.txt
```

Step 3: : PCAP Analysis: The next step was to label the data based on the hypothesis that is mentioned in phase one. The data from the text file was read into a Python dataframe. For this purpose, the columns that represent each value in the dataframe were required to be identified. The data frame was divided into the following columns:

1. Sr.No: It is the number of rows present in a certain file, can be used as an index to keep track of the threats line numbers.
2. Date: The date is a critical column from which many threats which have infiltrated the system can be traced back to the specific network source.
3. Time: The timestamp on a certain network packet is stored here.
4. Source_IP: This column stores the Ip address from where the packet has been sent.
5. Arrow: this is a placeholder column
6. Dest_Ip: This column stores the Ip address from where the packet has been received.
7. Protocol Used: This is the type of protocol used by the packet.
8. Length: The length of the specific packet can be identified using this column
9. Method : It can display a certain method used by the protocol
10. Link : If any link was accessed it will be displayed here
11. Version: It can be an associate version of the protocol in use
12. Extra: it is a combination of information that might be critical but is not relevant.

Some of the above columns had a combination of NAAN values and null values which had to be removed. All of the rows were deleted as the machine would not be able to predict correct values based on Naan values. Data cleaning was performed to create an appropriate dataset. The data was further split into training and testing data in the ratio of 70:30 as advised by many machine learning algorithms and papers.

Step 4: : Machine Learning Algorithm Selection: Several techniques that could be used for statistical analysis like parametric testing or machine learning were studied.

But to perform prediction, machine learning models were taken into consideration. Further there were two classes of machine learning models to select from supervised and unsupervised. But to have control over what hypothesis is to be selected thus supervised machine learning was selected. Several models were implemented, and their accuracy was calculated.

Step 5: : The overfitting Paradigm: At first the dataset was predicting a value of 100% accuracy, which is considered very high. But it was found out that data overfitting was being done on the said pcap file as the size of the file for training was very small. To ensure that overfitting does not occur cross validation of data was done. This solved the problem, but this needs to be verified against a larger dataset.

Step 6: Ensemble Learning: To check whether all of these algorithms work better individually or in combination,an ensemble class was implemented for getting more accuracy. Thus, Naïve Bayes and Support Vector machine have been combined to form an ensemble model for predicting threats. The ensemble approach will give an overall better result and will also solve the overfitting problem to some extent. The final predicted values of the hypothesis can be seen as follows:

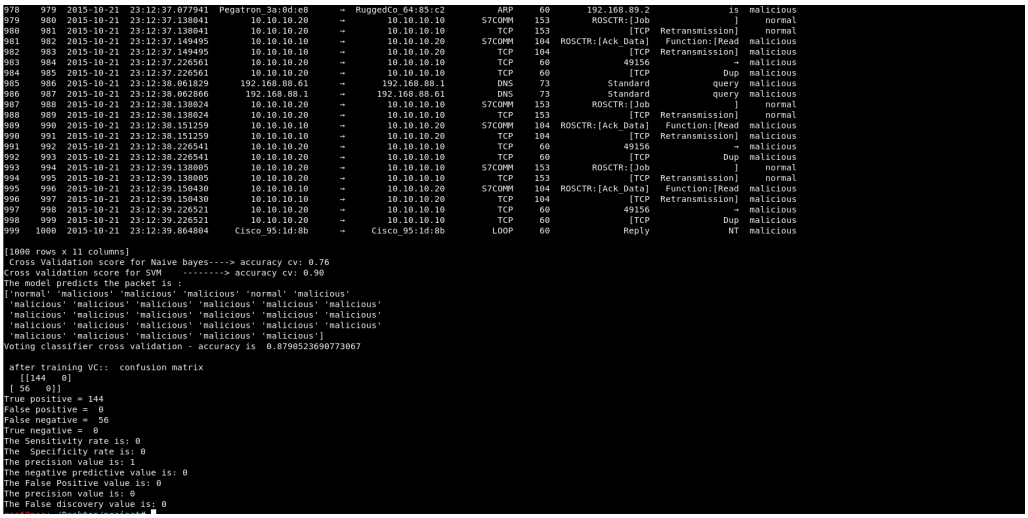

Figure 4: Ensemble Output

Step 7: : Retrain using a large dataset for better results: The training data can increase in size over the years.

- Selection of what threats are to be detected (based on business logic)
- When the hypothesis is finalized we can label the dataset and then implement our machine learning algorithm. This will fetch better results from the new log data files.
- If a comparison is to be done to recheck whether the machine is giving a proper output it would be necessary to distinguish normal and abnormal behaviour from captured pcap files.
- Retrain the model with a larger training data for greater accuracy.

6 Evaluation

The evaluation of this project is based on statistical analysis provided by python models. The evaluation of machine learning models is described by various authors [48] [49] [50] [51] [52] [53]. Using all of these findings as a baseline, the evaluation of this prediction model will be based on the metrics mentioned below. But to calculate and understand what all of these metrics are based on , we will need to define the values they are derived from. The following are the basis for defining every metric that is needed for machine learning algorithms

1. Accuracy: This is the ratio of the number of predictions made that are correct to the total number of input samples

2. The following table presents the accuracy scores of all classifiers. It can be observed that values in are in the same range for all models. The multinomial logistic regression, Support Vector machine and Voting Classifier combining all three have very close accuracy results. Voting classifier outperforms all other models.As the dataset used for this study is very large, a sample of 20000 rows is considered for implementation and efficient performance.

Algorithm Used	Number of Rows	Accuracy Obtained
Naive Bayes	20000	48.5%
SVM	20000	98.5%
Multinomial LR	20000	98.5%
Voting Classifier	20000	98.8%

Table 1: Representation of Accuracy for ML models

3. Confusion matrix: It is a kind of table that shows us the performance of a supervised machine learning algorithm. The rows represent the predicted class and the columns represent the actual values of the class. It actually shows whether two classes are being mislabelled(confusion).

	P	N
P	97	0
N	46	57

Table 2: Confusion Matrix

   All correct predictions are located in the diagonal of the table (97,57), so it is easy to visually inspect the table for prediction errors, as they will be represented by values outside the diagonal.

4. Recall: It is the number correct positive results upon the total number of samples.

5. Precision: It is the number of correct positive results upon the number of positive classes predicted by the classifier

6. Specificity : It is the measure of the actual negative values that have been predicted as negative

```
[1000 rows x 11 columns]
('naivebayes accuracy :', 0.485)
('SVM ACCU:', 0.985)
('LR ACCU:', 0.985)
The model predicts the packet is :
['normal' 'normal' 'malicious' 'malicious' 'normal' 'normal' 'malicious'
 'malicious' 'normal' 'malicious' 'malicious' 'normal' 'malicious'
 'malicious' 'normal' 'malicious' 'malicious' 'normal' 'malicious'
 'malicious' 'normal' 'malicious' 'malicious' 'normal' 'malicious'
 'malicious' 'normal' 'malicious' 'malicious']
Voting classifier cross validation - accuracy is  0.9887405546284664

 after training VC::  confusion matrix
   [[97  0]
 [46 57]]
True positive = 97
False positive =  0
False negative =  46
True negative =  57
The Sensitivity rate is: 0
The  Specificity rate is: 1
The precision value is: 1
The negative predictive value is: 0
The False Positive value is: 0
The precision value is: 0
The False discovery value is: 0
```

Figure 5: Result Computation

6.1 Experiment 1

Here a hypothesis based on business logic is taken into consideration. Consider a scenario where a web development company has to find a threat, that logins to the system in non office hours and their only access to the server is in the form of credentials. Also, the relevant server has several ports active, one of the ports is HTTP port active to communicate to the rest of the office. The system is hardened and performs a packet capture on the active ports. Malicious packets are identified based on protocol and time bound activity.

```
df['state'] = 'normal'
df.loc[(df['Length'] < 65)
& (df['ProtocolUsed'] == 'HTTP')
& (df['Time'] > '23:10:34.995270' )
& (df['Time'] < '23:40:00.995270' ),'state']  = 'malicious'
```

Figure 6: Experiment 1

6.2 Experiment 2

Here it is taken into consideration a hypothesis for a type of attack. There was a delay in serving customers for a certain banking website, which had updated information about

how to deal with credit card loss. That webserver was down for a few minutes after a certain interval. This kept happening for multiple days. To find out packets that might have affected the server logs were checked for http packets, which were clear. Next our machine learning algorithm can check for a condition of DOS using ICMP requests. IT can correctly identify malicious packets using the following condition:

```
df['state'] = 'normal'
df.loc[(df['Length'] < 100)
& (df['ProtocolUsed'] == 'ICMP')
& (df['link'].str.contains('unreachable')),'state']  = 'malicious'
```

Figure 7: Experiment 2

6.3 Discussion

In the above experiment threat conditions have been successfully hypothesized for detection of malicious of packets in a limited time-frame. While verification of experiments based on attack vectors and business logic it was proved that detection of malicious packets from PCAP files could be more efficiently done via machine learning. Once the hypothesis is converted to actual logical code, that same threat hunting tool can be re-used. Ensemble learning need several iterations to learn from the training data. The following are limitations of my experiment

1. The above machine learning model will require a huge dataset of training data.
2. In the above experiment the labelling of data is done based on the conditions, this might be a tedious task as it has to be repeated for every set of training data.
3. There is a chance of the data being biased towards a certain result, there may be a need for noise reduction.
4. Any intelligence that is achieved by the model for a certain condition or instance would be bound to to that certain hypothesized condition.

7 Conclusion and Future Work

In this research pcap files have been used to hunt network level threats based on features like IP, time, protocol and length of the header. Finding threats in limited time and using hypothesis based conditions to customize threat matrix have been key features of the research. Traditional algorithms (Naıve Bayes, SVM and Logistic Regression) were implemented and ensemble techniques combining these algorithms were applied to improve the performance of individual classifiers.The ensemble techniques implemented present a better and efficient path for threat hunting.
The future work can be presented in the form of automating the hypothesis that is currently based on manual conditions. If this can be implemented the entire threat hunting process will be done on the basis of previously gained intelligence about attacks, each hypothesis would be training set for the machine. The use of algorithms like KNN along with Keras can be efficient ways of hypothesizing conditions.

References

[1] "This Asia-Pacific Cyber Espionage Campaign Went Undetected for 5 Years," library Catalog: thehackernews.com Section: Article. [Online]. Available: https://thehackernews.com/2020/05/asia-pacific-cyber-espionage.html

[2] J. Segura, "Web skimmer hides within EXIF metadata, exfiltrates credit cards via image files," Jun. 2020, library Catalog: blog.malwarebytes.com Section: Threat analysis. [Online]. Available: https://blog.malwarebytes.com/threat-analysis/2020/06/web-skimmer-hides-within-exif-metadata-exfiltrates-credit-cards-via-image-files/

[3] "Smartwatch Maker Garmin Shuts Down Services After Ransomware Attack," library Catalog: thehackernews.com Section: Article. [Online]. Available: https://thehackernews.com/2020/07/garmin-ransomware-attack.html

[4] "QSnatch Data-Stealing Malware Infected Over 62,000 QNAP NAS Devices," library Catalog: thehackernews.com Section: Article. [Online]. Available: https://thehackernews.com/2020/07/qnap-nas-malware-attack.html

[5] "Sophisticated 'TajMahal APT Framework' Remained Undetected for 5 Years," library Catalog: thehackernews.com Section: Article. [Online]. Available: https://thehackernews.com/2019/04/apt-malware-framework.html

[6] C. Cimpanu, "Joomla team discloses data breach," library Catalog: www.zdnet.com. [Online]. Available: https://www.zdnet.com/article/joomla-team-discloses-data-breach/

[7] Citizensinformation.ie, "Overview of the General Data Protection Regulation (GDPR)," archive Location: Ireland Library Catalog: www.citizensinformation.ie Publisher: Citizensinformation.ie. [Online]. Available: https://www.citizensinformation.ie/en/government_in_ireland/data_protection/overview_of_general_data_protection_regulation.html

[8] P. Wang, Y. Li, and C. K. Reddy, "Machine Learning for Survival Analysis: A Survey," *ACM Computing Surveys*, vol. 51, no. 6, pp. 110:1–110:36, Feb. 2019. [Online]. Available: https://doi.org/10.1145/3214306

[9] "The Zeek Network Security Monitor," library Catalog: zeek.org. [Online]. Available: https://zeek.org/

[10] "Cyber Threat Hunting 7hrough the 8se of an Isolation Fores." [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3134302.3134319

[11] "The Diamond Model of Intrusion Analysis." [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA586960.pdf

[12] "A Practical Model for Conducting Cyber Threat Hunting." [Online]. Available: https://www.sans.org/reading-room/whitepapers/threathunting/practical-model-conducting-cyber-threat-hunting-38710

[13] H. Rasheed, A. Hadi, and M. Khader, "Threat Hunting Using GRR Rapid Response," in *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Oct. 2017, pp. 155–160.

[14] K. Wafula and Y. Wang, "CARVE: A Scientific Method-Based Threat Hunting Hypothesis Development Model," in *2019 IEEE International Conference on Electro Information Technology (EIT)*, May 2019, pp. 1–6, iSSN: 2154-0373.

[15] I. Ghafir and V. Prenosil, "DNS traffic analysis for malicious domains detection," in *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb. 2015, pp. 613–918.

[16] M. N. S. Miazi, M. M. A. Pritom, M. Shehab, B. Chu, and J. Wei, "The Design of Cyber Threat Hunting Games: A Case Study," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, Jul. 2017, pp. 1–6.

[17] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *2010 IEEE Symposium on Security and Privacy*, May 2010, pp. 305–316, iSSN: 2375-1207.

[18] "SANS 2018 Threat Hunting Survey Results." [Online]. Available: http://staging-resources.malwarebytes.com/files/2018/09/Survey_ThreatHunting-2018_Malwarebytes.pdf

[19] "Threat Hunting: Open Season on the Adversary." [Online]. Available: https://www.malwarebytes.com/pdf/white-papers/Survey_Threat-Hunting-2016_Malwarebytes.pdf

[20] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, "Know Abnormal, Find Evil: Frequent Pattern Mining for Ransomware Threat Hunting and Intelligence," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 2, pp. 341–351, Apr. 2020, conference Name: IEEE Transactions on Emerging Topics in Computing.

[21] "Threat Hunting and Active Cyber Defense - ProQuest," library Catalog: search.proquest.com. [Online]. Available: https://search.proquest.com/openview/1c55e99cbbe6158228a236259e46a024/1?pq-origsite=gscholar&cbl=18750&diss=y

[22] O. Benediktsson, D. Dalcher, and H. Thorbergsson, "Comparison of software development life cycles: a multiproject experiment," *IEE Proceedings - Software*, vol. 153, no. 3, pp. 87–101, Jun. 2006, conference Name: IEE Proceedings - Software.

[23] L. L. Ray, "Security considerations for the spiral development model," *International Journal of Information Management*, vol. 33, no. 4, pp. 684–686, Aug. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0268401213000418

[24] P. Wang, Y. Li, and C. K. Reddy, "Machine Learning for Survival Analysis: A Survey," *ACM Computing Surveys*, vol. 51, no. 6, pp. 110:1–110:36, Feb. 2019. [Online]. Available: https://doi.org/10.1145/3214306

[25] *Public PCAP files for download*, publication Title: Netresec. [Online]. Available: https://www.netresec.com/?page=PcapFiles

[26] "Secure Software Development Life Cycle Processes | CISA." [Online]. Available: https://us-cert.cisa.gov/bsi/articles/knowledge/sdlc-process/secure-software-development-life-cycle-processes

[27] "software-security-practices." [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/misc/ms-product-software-security-practices.pdf

[28] "Metrics to Evaluate your Machine Learning Algorithm | by Aditya Mishra | Towards Data Science." [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

[29] "API Reference — scikit-learn 0.23.1 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics

[30] "Simple guide to confusion matrix terminology," Mar. 2014, library Catalog: www.dataschool.io. [Online]. Available: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

[31] "classification - What does AUC stand for and what is it?" library Catalog: stats.stackexchange.com. [Online]. Available: https://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it

[32] "The Fractured Statue Campaign: U.S. Government Agency Targeted in Spear-Phishing Attacks," Jan. 2020, library Catalog: unit42.paloaltonetworks.com Section: Unit 42. [Online]. Available: https://unit42.paloaltonetworks.com/the-fractured-statue-campaign-u-s-government-targeted-in-spear-phishing-attacks/

[33] "Cardinal RAT Active for Over Two Years," Apr. 2017, library Catalog: unit42.paloaltonetworks.com Section: Unit 42. [Online]. Available: https://unit42.paloaltonetworks.com/unit42-cardinal-rat-active-two-years/

[34] "MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/

[35] "Tactics - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/enterprise/

[36] "Persistence, Tactic TA0003 - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/TA0003/

[37] "Defense Evasion, Tactic TA0005 - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/TA0005/

[38] "Credential Access, Tactic TA0006 - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/TA0006/

[39] "Discovery, Tactic TA0007 - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/TA0007/

[40] "Lateral Movement, Tactic TA0008 - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/TA0008/

[41] "Collection, Tactic TA0009 - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/TA0009/

[42] "Command and Control, Tactic TA0011 - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/TA0011/

[43] "Exfiltration, Tactic TA0010 - Enterprise | MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/tactics/TA0010/

[44] "Applying the Scientific Method to Threat Hunting." [Online]. Available: https://www.sans.org/reading-room/whitepapers/threathunting/applying-scientific-method-threat-hunting-39610

[45] L. Sun, J. Wu, Y. Zhang, and H. Yin, "Comparison between physical devices and simulator software for Cisco network technology teaching," in *2013 8th International Conference on Computer Science Education*, Apr. 2013, pp. 1357–1360.

[46] G. Stone, B. Lundy, and G. Xie, "Network policy languages: a survey and a new approach," *IEEE Network*, vol. 15, no. 1, pp. 10–21, Jan. 2001, conference Name: IEEE Network.

[47] whoopsie, "m6c7l/pcapify," Sep. 2019, original-date: 2018-02-03T16:04:51Z. [Online]. Available: https://github.com/m6c7l/pcapify

[48] "Public PCAP files for download," library Catalog: www.netresec.com. [Online]. Available: https://www.netresec.com/?page=PcapFiles

[49] "The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics." [Online]. Available: https://www.aaai.org/Papers/ICML/2003/ICML03-028.pdf

[50] G. S. Handelman, H. K. Kok, R. V. Chandra, A. H. Razavi, S. Huang, M. Brooks, M. J. Lee, and H. Asadi, "Peering Into the Black Box of Artificial Intelligence: Evaluation Metrics of Machine Learning Methods," *American Journal of Roentgenology*, vol. 212, no. 1, pp. 38–43, Oct. 2018, publisher: American Roentgen Ray Society. [Online]. Available: https://www.ajronline.org/doi/full/10.2214/AJR.18.20224

[51] "An Analysis of Rule Evaluation Metrics." [Online]. Available: https://www.aaai.org/Papers/ICML/2003/ICML03-029.pdf

[52] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi, "An Evaluation of Machine Learning-Based Methods for Detection of Phishing Sites," in *Advances in Neuro-Information Processing*, ser. Lecture Notes in Computer Science, M. Köppen, N. Kasabov, and G. Coghill, Eds. Berlin, Heidelberg: Springer, 2009, pp. 539–546.

[53] G. V. Cormack and M. R. Grossman, "Evaluation of machine-learning protocols for technology-assisted review in electronic discovery," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, ser. SIGIR '14. Gold Coast, Queensland, Australia: Association for Computing Machinery, Jul. 2014, pp. 153–162. [Online]. Available: https://doi.org/10.1145/2600428.2609601