University of Nevada, Reno

Analysis of Failed SSH Attempts for Intrusion Detection

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Computer Science and Engineering

by

AKIN ALKAN

Dr. Engin Arslan - Thesis Advisor
May 2024

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

Akin Alkan

entitled

Analysis of Failed SSH Attempts for Intrusion Detection

be accepted in partial fulfillment of the
requirements for the degree of

Master of Science

Engin Arslan, Ph.D.
Advisor

Shamik Sengupta, Ph.D.
C*ommittee Member*

Gi W. Yun, Ph.D.
Graduate School Representative

Markus Kemmelmeier, Ph.D., Dean
Graduate School

May, 2024

Abstract

SSH brute force attacks remain among the most common attack types in computer systems. Recent threat analysis reports consistently highlight their prevalence as a top web security vulnerability. Various passive and active methodologies have been developed to deal with this problem, each with its unique set of advantages and disadvantages. Amongst all, analysis of monitoring logs is important to understand the root cause of the problem and implementing necessary countermeasures. Hence, this thesis focus on implementing automated intrusion detection solutions by analyzing historical failed SSH attempts.

The data is captured between March 2022 and June 2022 from a server located at University of Nevada, Reno (UNR) campus. We first present thorough analysis of the dataset understand common patterns in the dataset such as origin of IP addresses, usernames, and time of SSH attempts. We identified various types of attack patterns, including slow, steady, and stealthy ones. Since the logs contain both benign and malicious attempts, we utilized external databases (e.g., IPWHOIS and ABUSEIPDB) to classify them as malicious or not, which served as a training data for machine learning models.

We developed several machine learning models to categorize SSH attempts as malicious or benign. The models relied on several features including username, time difference of the attacks, the number of previous attempts, and similarity of IP addresses. We trained Random Forest, Decision Tree, XGBoost, SVM, and Logistic Regression models to evaluate their performance. The Decision Tree model exhibited the best performance, achieving 100% precision and a recall rate of 97.9%. In comparison, the best performed rule-based formulation failed to identify 1.5% malicious

IPs whereas the Decision Tree model only missed 0.01%. We also validated the results against the public datasets. We noticed that the proposed model detected five malicious IP addresses before they appear in public databases such as ABUSEIPDB, which is a promising result for the proposed model.

Dedication

To my parents, Yusuf and Mesude Alkan, my wife Ikbal Alkan, my children Murat and Nihal Alkan, and to my friends, with a special mention to those in Turkey whose freedom is unjustly restrained.

Acknowledgments

I would like to extend my sincere gratitude to my advisor, Dr. Engin Arslan, for being a constant source of inspiration and guidance throughout all phases of my research. Additionally, I want to express my heartfelt appreciation to my committee members, **Dr. Shamik Sengupta and Dr. Gi W. Yun** for their dedicated time in reviewing this thesis and providing invaluable insights for my research.

I am deeply grateful to my wife and kids for their unwavering support throughout my journey. I also would like to express my thanks to my friends Osama, Nagmat, Arif, Ehsan, Roya, Jahanzeb, and Mukul for supporting me during this journey.

Table of Contents

List of Figures

Chapter 1

Introduction

Internet is impacting our life than ever before not only the size of data we use, but also the devices we interact, in our productivity, in our social life, healthcare, finance, education and more. With the developments in artificial intelligence (AI), internet of things (IoT), robotics, cloud systems, and wearable devices interaction through wires goes far beyond access to the information. Parallel to improvement in internet, the threat landscape has also increased. There is no one or no organization out of this threat as well as they have an access to the digital world.

Brute force SSH attacks against remote access to servers is one of the most common attack types in security. Even if it is perceived as old-fashioned attack, according to the threat analysis reports it is still at the top for web security vulnerabilities. In the first quarter of 2022, brute force attacks accounted for 51 percent of the most common attack vectors across cloud providers [4]. Briefly, a SSH brute force attack is a network level attack that the attacker tests the username and password matches to gain access to the system by several attempts. Once the attacker successfully login to the system, according to the attackers' intention they can ex-filtrate the data, consume the resources such as CPU time and bandwidth, harm the reputation of a

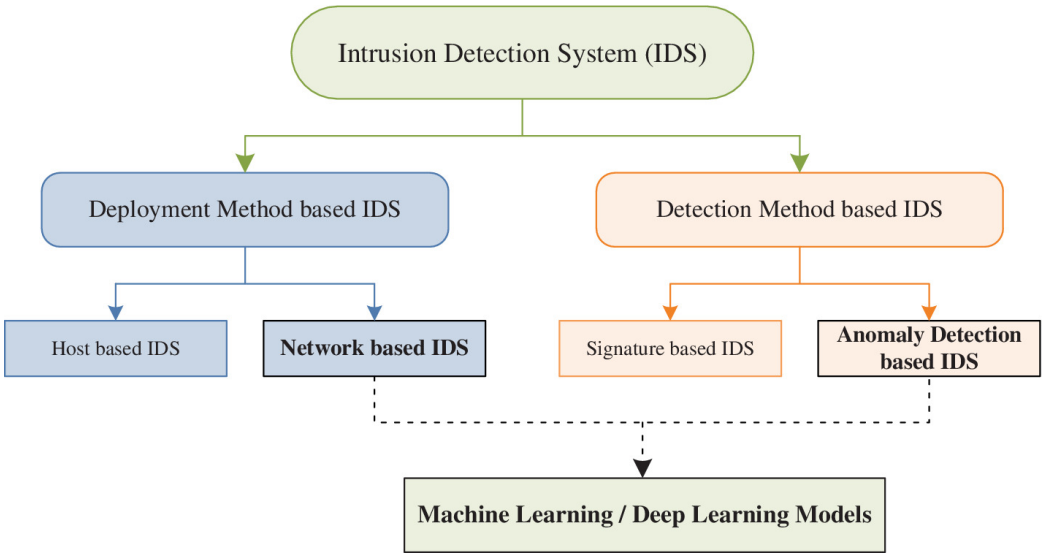

Figure 1.1: Intrusion Detection System Classification [1]

company resulting in revenue loss and diminished customer satisfaction, deny access to the system. There are different passive and active methodologies that have been developed to overcome this problem; forcing use strong passwords, disabling root SSH logins, employing two factor authentication, using intrusion detection systems and more. All of these measures have pros and cons in terms of time, cost, and effectiveness. However, they complement each other in regards to detection and prevention.

An in-depth solution is required to overcome this problem. Monitoring the logs is an important stage to understand the root cause of the problem and for taking necessary precautions. An intrusion detection system (IDS) is a system that detects an unauthorized access with monitoring the logs. These logs can be monitored in host level (HIDS-Host-based intrusion detection system) or in network level (NIDS-Network-based intrusion detection system). One drawback of detection in host-level is its scalability. HIDS needs to be deployed each device and this results in extra processing time and cost. NIDS are deployed on the network, they continuously monitor the network traffic and scans for any potential threat to protect every device

connected to that network.

Independent of deployment method there are two main detection approaches against intrusions: signature-based intrusion detection and anomaly detection-based intrusion detection. Signature-based intrusion detection are effective for known threats since the pattern is formulated and stored in a database, when a data pattern matches with one of the stored patterns, an alarm is initiated. On the other side, if there is a new attack pattern that has not been experienced before, signature-based intrusion detection systems fail. Anomaly based intrusion detection systems are based on the behaviour of the benign activity. If there is a deviation from this profile, then data pattern is considered abnormal. Anomaly detection fills the gap that HIDS does not cover, the novel attacks. However, they produce much more false alarm rate than HIDS that takes extra time and cost and it is hard to tune the boundary between the normal and abnormal data patterns. In summary, signature based IDS use rules and thresholds to detect the malicious activities, while anomaly detection based IDS use machine learning and deep learning models, and each of them has its own advantages and disadvantages [1] [7].

1.1 Motivation

Recent studies on intrusion detection focus on the efficiency of machine learning and deep learning models in detecting malicious behaviors. Most of the researchers used IDS 2017 and IDS 2018 datasets in their studies, where as some others created their own datasets with honeypots. Labeling datasets is a crucial yet time-consuming and knowledge-intensive task. Without precise labeling, models may not perform as intended, and the resulting accuracy and precision may not reflect the actual outcomes. This study has a new contribution in this field with a new dataset and the

explanation how it is labelled.

One of the findings in these studies is there is no commonly accepted evaluation metric. This situation makes it challenging to compare results especially where unbalanced datasets are used. We advocate for the use of the confusion matrix as the most informative evaluation metric, allowing the observation of false positives and false negatives.

Additionally, every organization has its unique network traffic —whether it's a university network, a financial institution network, or a government department network— resulting in different traffic patterns based on the organization's web activities. Therefore, it is crucial to fine-tune the models according to each network's characteristics.

Our study focuses on detecting malicious activities with limited information, and the primary source for the logs is the Linux btmp file under the 'var/log/' directory. The btmp file records all unsuccessful login attempts and comprises eight columns. In our study, we have specifically utilized three key features: username, hostname, and datetime information. While username information is not a part of netflow data, both hostname and datetime information can be found.

It has been mentioned earlier that one drawback of host-based systems is scalability. However, in today's world, crucial logs such as security logs (including login attempts), system logs, and application logs are commonly sent to SIEM solutions. This practice significantly contributes to making host-based systems more scalable. While previous studies focused on network-based solutions, our study is a machine learning approach from host-base perspective with a new accurately labelled dataset, fewer but important features, and a more reliable evaluation metrics.

Chapter 2

Background and Literature Review

2.1 Secure Shell Connection and Linux Logs

SSH, a cryptographic network protocol, ensures secure operation of network services across unsecured networks [8]. Its primary applications include secure remote login and command-line execution. Telnet, developed in 1969, initially facilitated remote host communication but posed a security risk as it transmitted data in plain text. As the internet rapidly expanded, a significant security gap emerged.

In response, Finnish computer scientist Tatu Ylonen designed SSH in 1995 at Helsinki University of Technology. Ylonen's creation addressed the security flaw in Telnet, gaining widespread acceptance globally. SSH's clear design, simplicity of implementation, and independence from centralized infrastructure contributed to its global adoption as a secure communication solution [9]. It is one of the most popular communication protocols on the internet used by developers, webmasters, and system administrators. It permits the user to gain remote access to a service in seconds using an encrypted communication channel [10].

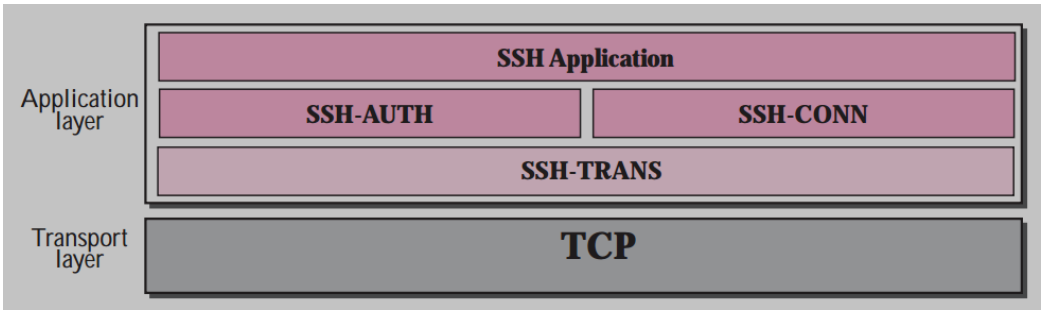

Figure 2.1: SSH Architecture [2]

SSH is a secure application layer protocol that operates over the transport layer (TCP), facilitating remote secure communication. Initially, an insecure TCP connection is established, but it is followed by a secure channel referred to as SSH-TRANS. Subsequently, authentication between the server and client, known as SSH-AUTH, is validated. Finally, SSH-CONN combines multiple data streams and transmits them efficiently over a single medium [2].

Linux records security-related information under the /var/log directory. There are four key files related to SSH activity.

- var/log/auth: contains system authorization information.
- var/log/utmp: Records currently logged in users.
- var/log/wtmp: Logs all current and past logins.
- var/log/btmp: Documents all unsuccessful (bad) login attempts.

When an attacker or an unauthorized user attempts to gain access to the system, they try to connect to the remote server using several username and passwords. To overcome this challenge, attackers often resort to a SSH brute force attack, relentlessly attempting various username-password combinations to breach the system's security. These logs provide essential information about who accessed the system, when, from where, and how many times.

2.2 SSH Brute Force Attacks

A SSH brute force attack is a methodical approach employed to discover an unknown value by exhaustively testing every possible key combination, with the aim of gaining access to a specific target resource. Besides its use in targeting login credentials like SSH, a SSH brute force attack can also be employed to guess hidden pages or concealed content, decipher session ID values, crack one-time pass-codes, unveil credit card numbers, reverse cryptographic hash functions, and to exhaust server resources, ultimately resulting in a deterioration of service quality or rendering the system inoperable [11].

SSH brute force attack is one of the simplest, and the oldest attack type in computer security, but its consequences may have a significant impact. In 2007, SANS reported the brute force attacks as one of the most common forms of attack type in Institute Security Risk Report [11]. In 2021, The Open Worldwide Application Security Project (OWASP) declared that Broken Access Control is the most common vulnerability in 2021 [3].

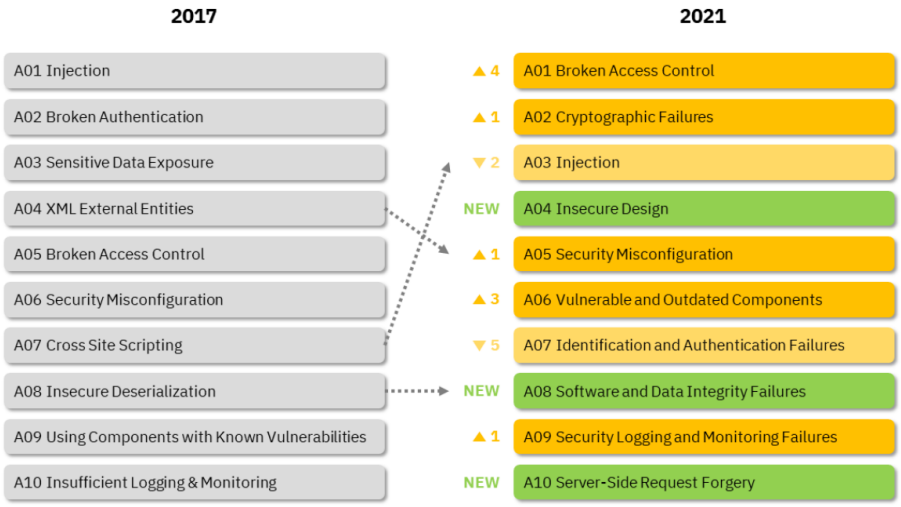

Figure 2.2: OWASP Top 10 (2021) [3]

According to Google's Threat Horizon's report prepared by Google's Cybersecurity Action Team in July 2022, attackers are actively scanning for and exploiting misconfigured cloud resources with the primary objective crypto mining and ransomware. According to the report, in the first quarter of 2022, brute force attacks accounted for 51 percent of the most common attack vectors across cloud providers. Recent data breaches and the ongoing use of weak passwords have provided hackers with the chance to amass extensive login information and carry out brute force attacks on cloud admin accounts and privileged users [4].

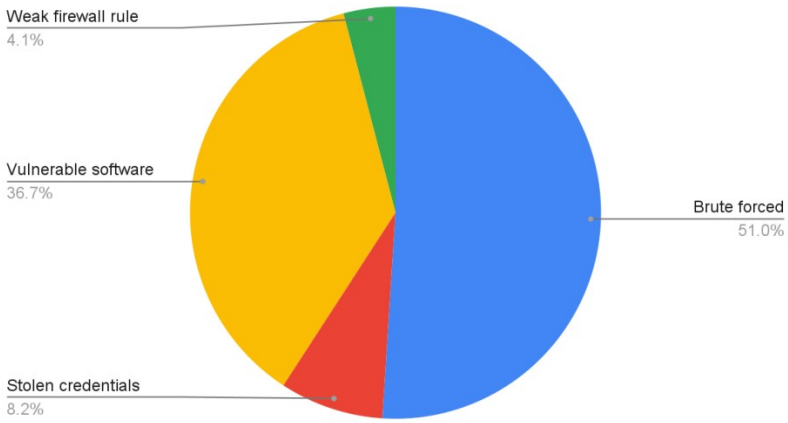

Figure 2.3: Google Threat Horizon Report 2022 [4]

2.2.1 Brute Force Attack Types based on Password Guessing

Brute force attacks come in various types, each distinguished by its specific method and approach to crack the passwords. Humans have a tendency to choose relatively short and simple passwords that is easier to remember. In contrast, servers generate stronger passwords, adhering to specific security requirements [12]. To crack the passwords either they are human generated or server generated different methodologies have been improved [13] [14]:

- **Simple Brute Force Attacks:** Hackers attempt to guess passwords logically without relying on any software or tool. This method is used for simple passwords.

- **Dictionary Attacks:** In this attack, a dictionary is employed, and each word and character combination within the dictionary is systematically tested to determine if it matches a valid password.

- **Hybrid Brute Force Attacks:** A hybrid attack typically combines dictionary and brute force methods, aiming to decipher passwords that blend familiar words with random characters.

- **Reverse Brute Force Attacks:** In a reverse brute force attack, the attack strategy is inverted, commencing with a known password. Hackers then search through millions of usernames, persisting until they discover a matching pair. Frequently, cybercriminals initiate this approach using leaked passwords sourced from existing data breaches, available online.

- **Spraying Attacks:** This type of attack is designed to circumvent lockout mechanisms. A vast list of targeted user accounts is initially compiled. A password is then selected and systematically tested against every account on the list. This process repeats with successive passwords until the desired result is achieved.

- **Credential Stuffing:** When a hacker possesses a username-password combination that proves effective on one website, they often employ it across numerous others. This strategy specifically targets users who tend to reuse their login credentials across multiple websites.

- **Rainbow Table Attacks:** This attack aims to unveil a password from its

hash. The most time-consuming aspect of this process is generating the hash from the known passwords.

2.2.2 Types of Brute Force Attacks based on Source

Brute force attacks exhibit variations according to their sources, either attacking from a single source or involving distributed forces. The distinction between these types lies in their detection complexity, with single-source attacks being more simple to identify compared to their distributed ones. Understanding these differences in the types of brute force attacks based on their sources is important for implementing effective security strategies that can detect and mitigate these threats [15].

- **Single Source Brute Force Attacks:** In this scenario, attackers use a single machine to execute the brute force attack. The process involves systematically trying various credentials over time. While these attacks are relatively easier to detect, they can still pose a significant threat when the attacker employs evasion techniques.

- **Single Domain Brute Force Attacks:** Attackers expand the scope by incorporating several machines which are located in similar regions in terms of same organization, same city, same country that share similar IP addresses.

- **Distributed Brute Force Attacks:** Distributed brute force attacks are more sophisticated that attackers orchestrate a network of machines distributed globally that do not share similar IP prefixes. These attacks may be hard to detect if the attackers spread their attack in long term.

2.2.3 Tools for Brute Force Attacks

Numerous brute force attack tools are presently available, each providing specialized support for different sets of protocols and operating systems. Examples include BruteX, Secure Shell Bruteforcer (SSB), Patator, and Ncrack, which specifically target the brute forcing of SSH protocols [16].

New tools are constantly emerging in this field of cybersecurity. As of mid-June 2022, a new Mirai-based botnet, "RapperBot," has surfaced, focusing on Linux servers and utilizing 3,500 unique IPs from global IoT devices [17]. Furthermore, a recently introduced Python-based credential harvesting tool named "Legion," in 2023, includes a brute forcing mechanism for exploiting vulnerabilities in poorly managed web servers [18].

2.2.4 Brute Force Attack Mitigation Techniques

Building preventive measures requires a thorough understanding of phases of brute force attacks. Before an attack starts to a host, it must be discovered. This is done by network scanning tools like nmap or by using publicly available lists of potential targets as provided on PasteBin (pastebin.com) or acquired by services like Shodan (shodan.io). After the discovery, brute force attack can take place, and if the valid credentials are matched compromise phase starts. In short, brute force attacks consists of three main phases: scanning phase, followed by a brute force attack phase and ended by a compromise phase if the attack was successful [19].

Various passive and active prevention methods exist to countermeasure brute force attacks, each with its distinct advantages and disadvantages. Achieving robust and in-depth security involves employing a combination of these methods based on specific goals and needs, while considering factors security, accessibility, and cost.

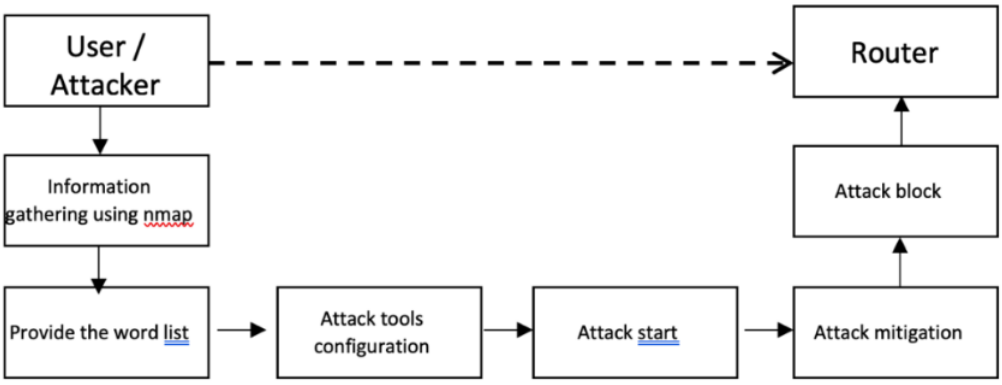

Figure 2.4: Attack Schema [5]

- **Encourage the use of robust, lengthy passwords:** Password length and complexity effects the time of the cracking the password. The table illustrates the time required to crack a password, factoring in its length and complexity, using a cracking power of 632 GH/s equivalent to 632 billion different passwords per second [6]. It is worth noting that dictionary attacks operate differently; if your password is found in the dictionary, it can be compromised instantly, regardless of its complexity [20].

- **Implement limits on the number of failed login attempts:** After a series of unsuccessful login attempts surpass a predefined threshold, the account is subject to temporary lockout or other predefined security actions like delayed response or employed CAPTCHA challenges. However, it's essential to recognize that this countermeasure can be susceptible to DoS and DDoS attacks, where malicious actors can intentionally lock legitimate users out of their accounts. Another potential drawback is the risk of users inadvertently locking themselves out due to password typos. In such cases, users typically require assistance from IT services, which can potentially strain service providers due to an influx of support requests. Therefore, the threshold for restricting incor-

Password Length	Numerical 0-9	Upper & Lower case a-Z	Numerical Upper & Lower case 0-9 a-Z	Numerical Upper & Lower case Special characters 0-9 a-Z %$
1	instantly	instantly	instantly	instantly
2	instantly	instantly	instantly	instantly
3	instantly	instantly	instantly	instantly
4	instantly	instantly	instantly	instantly
5	instantly	instantly	instantly	instantly
6	instantly	instantly	instantly	20 sec
7	instantly	2 sec	6 sec	49 min
8	instantly	1 min	6 min	5 days
9	instantly	1 hr	6 hr	2 years
10	instantly	3 days	15 days	330 years
11	instantly	138 days	3 years	50k years
12	2 sec	20 years	162 years	8m years
13	16 sec	1k years	10k years	1bn years

Figure 2.5: Password Cracking Duration [6]

rect password attempts should be thoughtfully determined, striking a balance between security and user accessibility [20].

- **Implement a throttle rate for repeated login attempts:** This approach will slow down the attacker's each attempt, but in case of forgetting the password it will take exponential time to access the system.

- **Use two factor authentication:** While implementing a two factor authentication is highly effective in enhancing security, it involves additional complexities in terms of setup, ongoing management, and associated costs.

- **Use an IP deny list:** IP deny lists are effective against known threats and quick response to incidents. However, due to the nature of dynamic IP assignments, deny lists may not be sufficient and/or attackers can manipulate the IP addresses.

- **Use public key based authentication:** Using public key based authentication may help prevent any brute force attack provided the security of the public

key remains uncompromised. However, this method necessitates securely sharing public keys in advance [21].

- **Monitor the logs:** Log monitoring allows for the early detection of brute force attacks. It enables timely intervention before unauthorized access is achieved, helps trace the source of brute force attacks, block specific IP addresses, identify attackers' tactics, techniques, and procedures.

2.3 Related Work

The idea of detection intrusion in network systems goes back to 1980 and since then there has been lots of developments and improvements in intrusion detection systems. In recent years, the researchers have explored the effectiveness of Machine Learning and Deep Learning models in network traffic to detect malicious activities. While ML based detection systems depends primarily on feature engineering to predict the output, deep learning-based systems operate on raw data, incurring higher costs and providing less explainable feature information [7].

As mentioned in the introduction section, there are two primary detection methods in IDS; signature-based and anomaly detection-based. Both system have their own advantages and disadvantages. Signature-based systems proves their effectiveness against to known threats and they are faster, but fails with novel attacks [22]. In contrast, anomaly-based systems excels in detecting zero-day attacks but often comes with a higher false alarm rate, thereby increasing investigation time and incurring additional costs.

The study [7] surveyed papers in the field of network intrusion detection systems from 2017 to April 2020 focusing on machine learning and deep learning approaches. According to the paper, AutoEncoders (AE) and Deep Neural Networks (DNN) are

the predominant models deployed where Deep Learning models constitute 60 percent of the studies surveyed. Additionally, the study highlights KDD Cup'99 and NSL-KDD datasets are mainly tested in researches where these datasets are quite old to address modern network attacks.

The study [23] utilized honeypot data and deployed various Machine Learning (ML) models for analysis. Decision Tree (DT) and Random Forest (RF) demonstrated the best performance, achieving an F1 score of 0.92. The acquired dataset has the information of IP address, user, isroot, command, and DateTime, but lack of information about the size and distribution of the data.

In the conducted study [24], researchers evaluated the performance of an Artificial Neural Network (ANN) with a softmax classifier on the CIC-IDS 2018 dataset. The obtained results indicated superior performance compared to previous studies, with precision at 100, accuracy at 99.9, and an F1 score of 98.3. However, the study lacks clarity regarding the specific method used for splitting the data into training, validation, and testing sets. If a conventional splitting method was employed, there exists a risk of over-fitting, potentially leading to the model memorizing known labels rather than generalizing well to new, unseen data.

The paper [25] conducted tests on the performance of LightGBM and XGBoost models using the CIC-IDS 2018 dataset. During the feature selection phase, the number of features was reduced to 66 out of 79, with the top 20 highest-ranked features being selected, but excluding DateTime which we believe that important feature has been lost. The study found that this feature selection process led to improved results and reduced computational time. Notably, LightGBM, benefiting from built-in support for categorical features, demonstrated superior performance when the port destination feature was added as a categorical variable.

In study [26] a broad scope of attack strategies has been investigated in billion-

scale SSH brute force attacks at the National Center for Super-computing Applications. The data has been collected for 1000 days, and honeypot recorded 4.5 million unique, globally distributed IP addresses. This study has lots of interesting findings that can help design the intrusion detection systems especially in terms of anomaly detection and proves that how leaked passwords and stolen keys are important for SSH brute force attacks.

Paper [27] centers on evaluating the effectiveness of feature selection in CIC-IDS 2018 dataset, and tried different combination of features to find the smallest number of features to achieve consistently strong classification performance. Their findings show that with just two features BwdPacketss and minsegsize, decision tree model outperformed deep learning models in terms of classification performance.

Fernández, Gabriel C. and Xu, Shouhuai conducted a study [28] involving Deep Neural Networks (DNN) using the ICS IDS 2017 dataset. Their emphasis on DNNs stems from the models' ability to handle tabular data and categorical variables with high cardinality. In their research, they utilized 74 different features and transformed each distinct source IP, destination IP, source port, and destination port feature into low embedded dimensions. The DNN model, when considering IP addresses, exhibited a remarkable True Positive Rate (TPR) of 0.9992 and a low False Positive Rate (FPR) of 0.0003. Even without IP addresses, the DNN demonstrated a TPR of 0.9677 and a FPR of 0.052. A contribution of their study is the recognition that the first three octets of IP addresses can be effective in addressing the challenges associated with dynamic IP addresses.

Tiwari, Namrata and Hubballi, Neminath developed a mathematical model named Petri-Net instead of using the standard features of ICS IDS 2018 dataset [15]. In particular, the model uses the weighted average of Number of Packets per Flow, Number of Bytes per Flow, and Flow Duration. They show that these parameters are vital in

differentiating normal and failed login attempts. The model's methodology involves a series of checks, starting with verifying if the new connection is within a Blacklisted IP Address, confirming the port number corresponds to the SSH port, and ensuring the SSH connection rate is below a defined threshold. Additionally, they introduced the concept of IP prefix count, aggregating IPs in similar regions. According to their results, Petri-Net can adopt to different behaviors and detect attacks with minimum number of errors, but it needs additional information of blacklisted IP addresses.

In the study referenced as [29], the machine learning model k-NN is chosen because of its potential for pattern recognition, and decision tree is selected because of its ability to visually represent decision trees. The evaluation is conducted using the CICIDS17 dataset from the Canadian Institute for Cybersecurity, with the specific selection of features packet length mean, destination port, flow packet/s, flow duration, and init win bytes forward. Results revealed that the decision tree outperformed k-NN, achieving a F1 score of 99.98.

Paper [30] approached to the problem in a different perspective, instead of a model that aims to achieve a high detection rate and a low false alarm rate, they visualized the netflow traffic that can support decision making process for the human experts. Their evaluations show that the SSH brute force attacks have their own behavioral patterns that that are different from the behavioral patterns of the normal SSH netflow in terms of duration, traffic, and density.

Reference [31] is focused on predicting malicious hosts based on the density of the known blacklisted IPv4 addresses. The underlying concept is that IP addresses that appear close to those that were previously identified as malicious are more likely to become malicious in the future. The predictive model developed achieves a F1 score of 0.908. This valuable information serves as a crucial component in constructing our model for detecting malicious activities.

In summary, intrusion detection has been a focus of researchers since 1980's. The comprehensive survey presented in [7] sheds light on studies conducted between 2017 and 2020 where KDD Cup'99 and NSL-KDD datasets are used in 60 percent of related papers. Recent studies also focuses on machine learning and deep learning models, with the CIC-IDS 2018 dataset becoming a prominent choice. However, results vary due to differences in features and evaluation metrics. An important issue is the lack of clear details in studies regarding how they split and train their data

Chapter 3

Detection of SSH Brute Attacks

This study conducts a systematic approach to the dataset collected from the University of Nevada, Reno server [elastic-0.eng.unr.edu] during the period of March 1-31, 2022. It has three phases, data collection, extraction and cleaning, analysis of the raw data, and the machine learning implementation.

The initial form of the data was as a btmp file under var/log/ directory. As mentioned earlier, btmp file records all unsuccessful (bad) login attempts in Linux operating systems. utmpdump command is used to read and extract the data. utmpdump is a Linux program to dump btmp, utmp, and wtmp files in raw format, so they can be examined [32]. Raw data has 8 features:

- Type of record (type)
- PID of login process (pid)
- Terminal name suffix (id)
- Username (user)
- Device name or tty - "/dev/" (line)

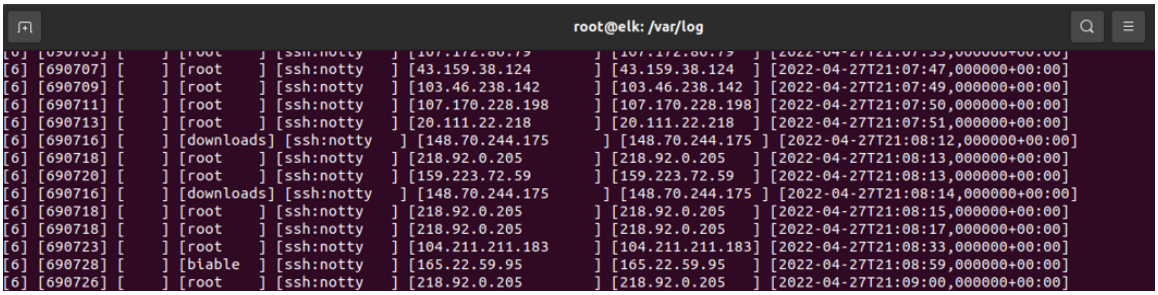

Figure 3.1: Raw btmp file.

- Hostname for remote login (host)
- Internet address of remote host (addr v6)
- Time entry was made (time)

In the first phase, the data is parsed using a python regex code, cleaned and pre-processed to analyze. The ultimate dataset is composed of 470,149 rows, with a total of 4,775 unique IP addresses. Then, all IPs' information is retrieved from the IPWHOIS website using an API connection. This retrieval includes important information like the Internet Service Provider (ISP), geolocation, country, continent, and other relevant features. Subsequently, this information is concatenated with the cleaned data.

In the second phase, the data is analyzed and visualised from various perspectives to derive meaningful insights.

In the final phase, the data is manually labeled, new rows are added to address the imbalance situation in the dataset, and various feature inputs are applied to increase the performance of the different machine learning models.

3.1 Analysis of the Dataset

In this section, we will analyze our dataset from the perspectives of various distributions; geographic location, ISP, time, frequency, and username.

In a SSH brute force attack, the main threat is originated from the IP addresses. If an IP address is blacklisted, it is easier to prevent the attack. However, blacklisting has several downsides. First, every blacklist is limited, there is not a system that checks every IP addresses in a limited reasonable time. Second, it only deals with the previously reported threats. Third, a stolen IP can be returned to original owner or an attacker can switch to the new IP addresses. So, predictive blacklisting is an alternative that predicts the possible malicious IP addresses in the future based on different features like ISP, the country or city where IP address is located, and similarity of the previously known malicious IP addresses. The performance of the prediction varies based on the different use cases and hypothesis behind the model [31].

3.1.1 Geographical Distribution

Understanding the geographic distribution of IPs is crucial when monitoring network connections to our entity and selectively blocking/ allowing IP addresses within or outside specific locations. In our database, all IP addresses, except one, are associated with the University of Nevada, Reno network. We visualized the geographical locations of the IP addresses using the Python folium library. In a continental bases, Asia shows the highest number of attacks, with North America and Europe following behind.

In a country-based classification, China leads, closely followed by the United States. Hong Kong, India, and Singapore follow in succession. In Europe, Germany and the Netherlands, and in South America, Brazil, are the primary sources of

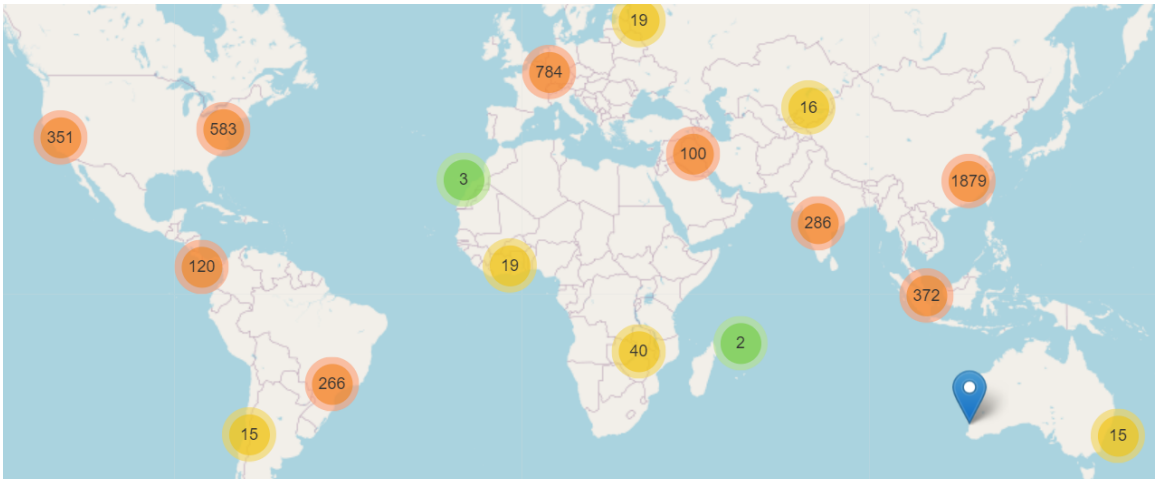

Figure 3.2: Geographic Distribution of IP Attacks

most attacks.

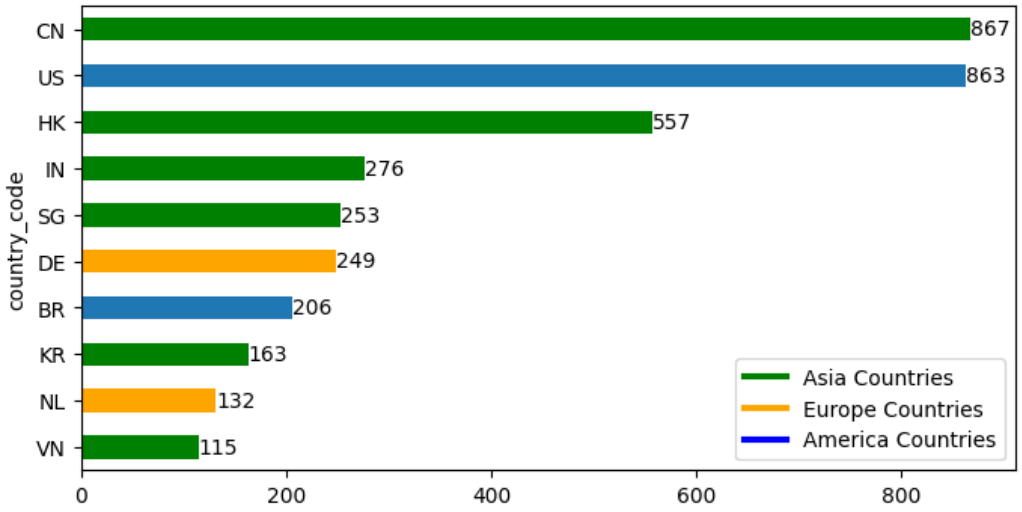

Figure 3.3: Distribution of Attacks Categorized by Countries

3.1.2 ISP Distribution

An Internet Service Provider (ISP) is a company that provides individuals or organisations access to the Internet. Customers choose ISPs based on factors such as location, service quality, cost, and specific needs. The variations in regulations,

law enforcement, and security controls among different ISPs impact the likelihood of attackers targeting or exploiting vulnerabilities within a particular ISP's network. According to our research US based Digital Ocean LLC and Singapure based 6 Collyer Quay are recognized for leading a bad reputation consisting of 31 percent of malicious IPs. However, there are 1096 different ISP providers that shows that attackers are trying to compromise from different providers.

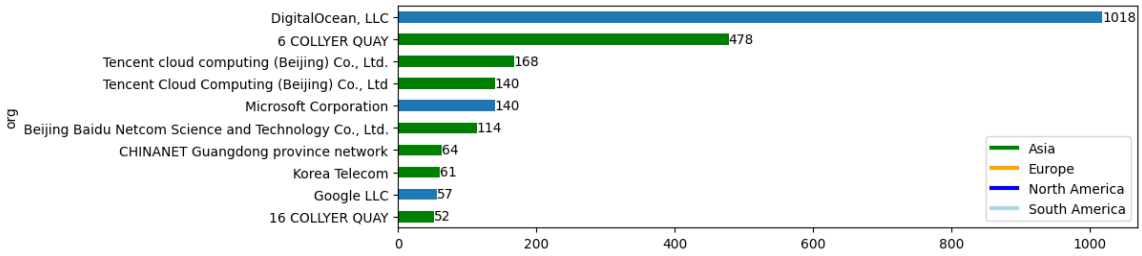

Figure 3.4: Distribution of Attacks Across ISPs

3.1.3 Time Distribution

We analyzed our dataset based on number of daily and hourly intrusions and specific to the countries. We found no correlation between the week days, but total count of intrusion attempts on Saturday are more than Sunday.

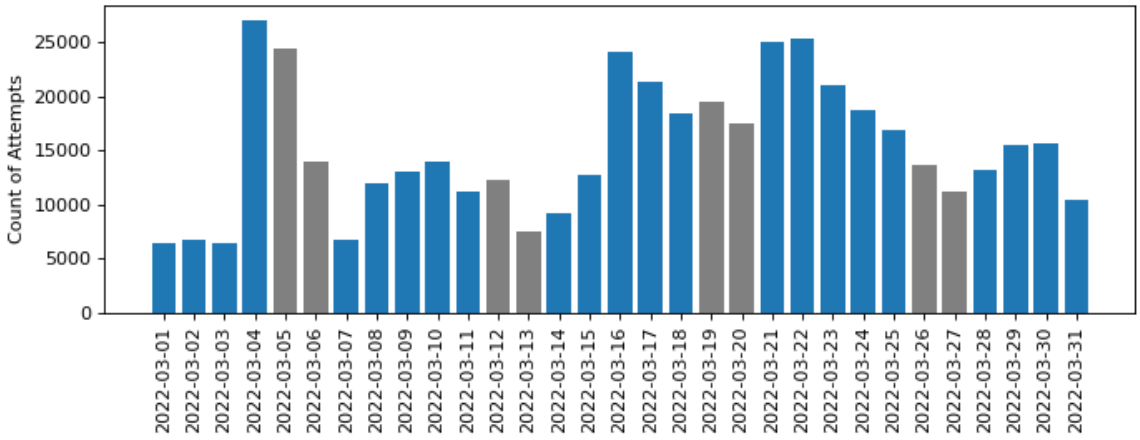

Figure 3.5: Count of Daily Intrusion Attempts

There is no meaningful difference between the count of hourly intrusions in a month. Even the count of hourly attempts per day are not distributed evenly, the total of hourly attempts in a month is close to each other. Since, intrusion attempts are from all over the world, and most of the attacks source are botnets, the result is as expected.

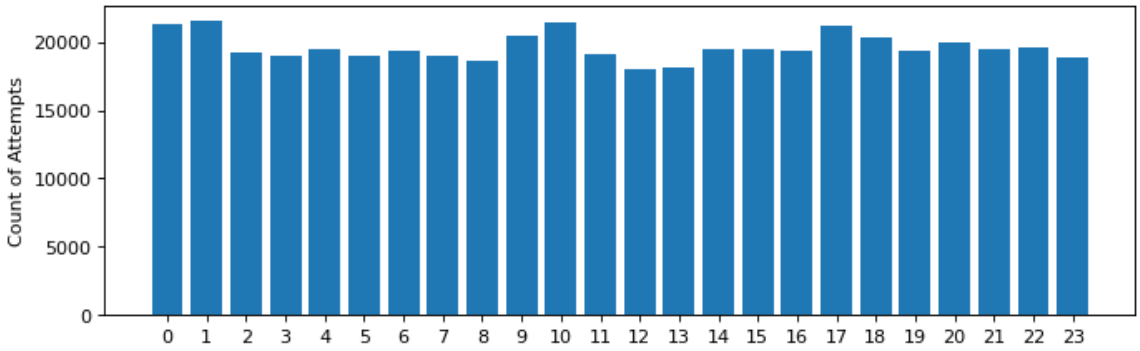

Figure 3.6: Total Count of Hourly Intrusion Attempts

We also analyzed the time distribution according to the countries where time difference may effect the hourly distribution. In countries with a high count of attacks, such as the US, China, and Hong Kong, we observed minimal variation in the hourly

attempts. However, in countries with lower attack volumes, such as Italy, France, and Indonesia, significant differences were noted in the hourly attempts. This discrepancy suggests that in countries with lower attack density, the patterns may be more indicative of human-controlled interactions compared to regions with higher attack frequencies.

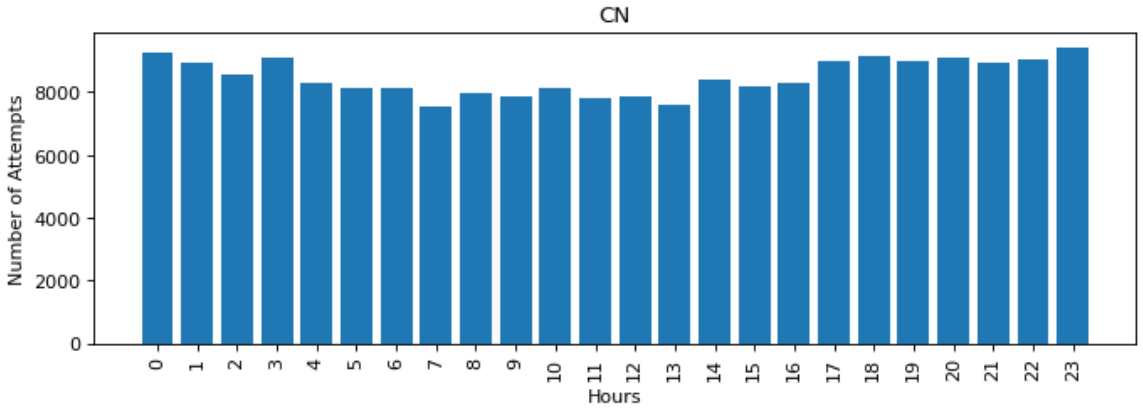

Figure 3.7: Hourly Intrusion Attempts from China

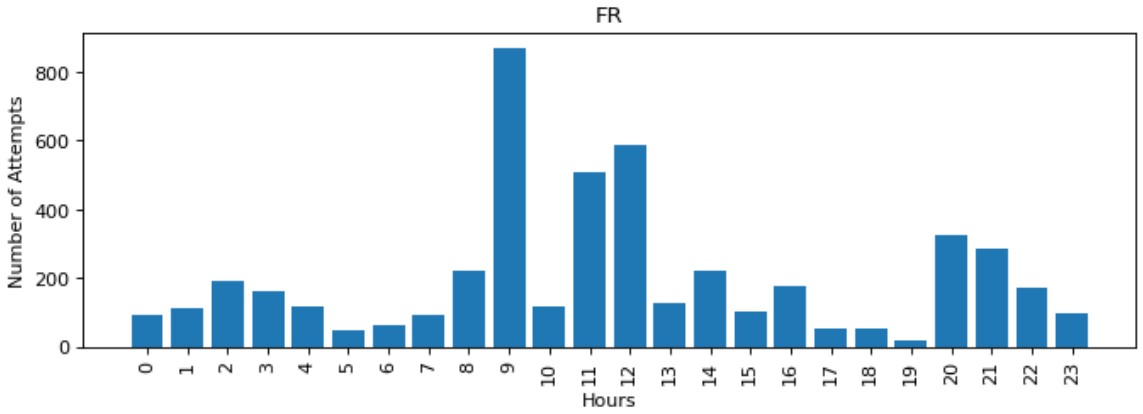

Figure 3.8: Hourly Intrusion Attempts from France

3.1.4 Frequency Distribution

In a btmp file where failed login logs are recorded, the number of repeating unique IP addresses should be limited in case there is not a SSH brute force attack. The only records should be the user entered the wrong credentials. In figure number 3.9, the x-axis represents the number of occurrences of each unique IP address, while the y-axis displays the count of IP addresses falling into each occurrence category. As an example, 20 IP addresses is showed up 680 times and 100 IP addresses is showed up 500 times. This information can be used to identify the botnets.

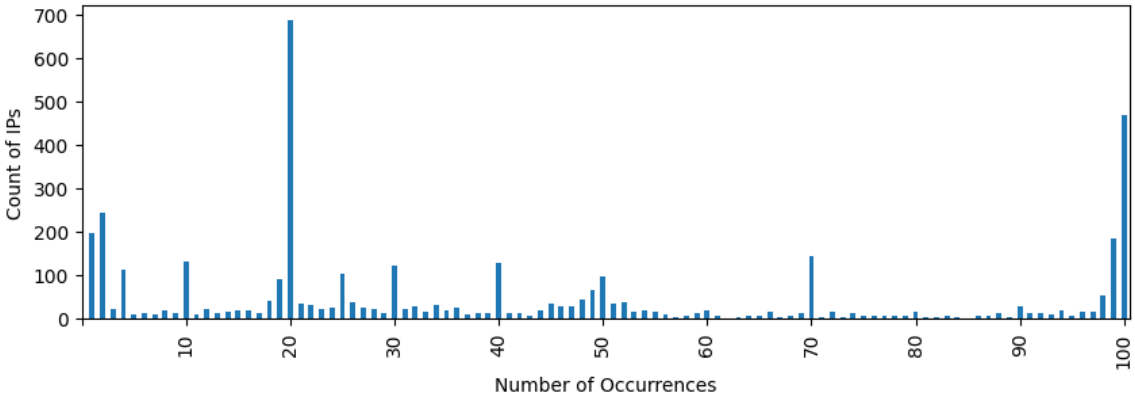

Figure 3.9: Distribution of IP Occurrences

3.1.5 UserName Distribution

The username plays an important role in classifying SSH brute force attacks in our dataset. Instances where multiple usernames originate from a single IP address or where the same username is tried across different IPs serve as indicators of a SSH brute force attack. In our dataset username "root" has been used 245867 times, which constitutes more than half of the data. This underscores a security vulnerability indicating that the use of the 'root' name poses risks due to its inherent privileges. admin, user, test, ubuntu,user2, postgres, oracle, ftpuser, ang git are used as usernames over

a thousand times. These usernames match with the SSH brute force attack dictionaries and to proactively address this threat, a recommended countermeasure involves restricting the use of frequently encountered usernames. Beyond that, compromised credentials are the weakest point to access the system until the administrators are aware of these stolen information.

3.2 Machine Learning Implementation

3.2.1 Data Labelling

We implemented supervised machine learning models in our dataset where the data labelling phase is the critical to acquire the accurate results. Our data labeling process has three stages. Initially, all IP addresses were visualized to check for patterns based on hour, minute, and second scales. If no pattern was detected, the specific location of the IP in the dataset was analyzed. Figure number 3.10 shows a clear indication of SSH brute force attack pattern.

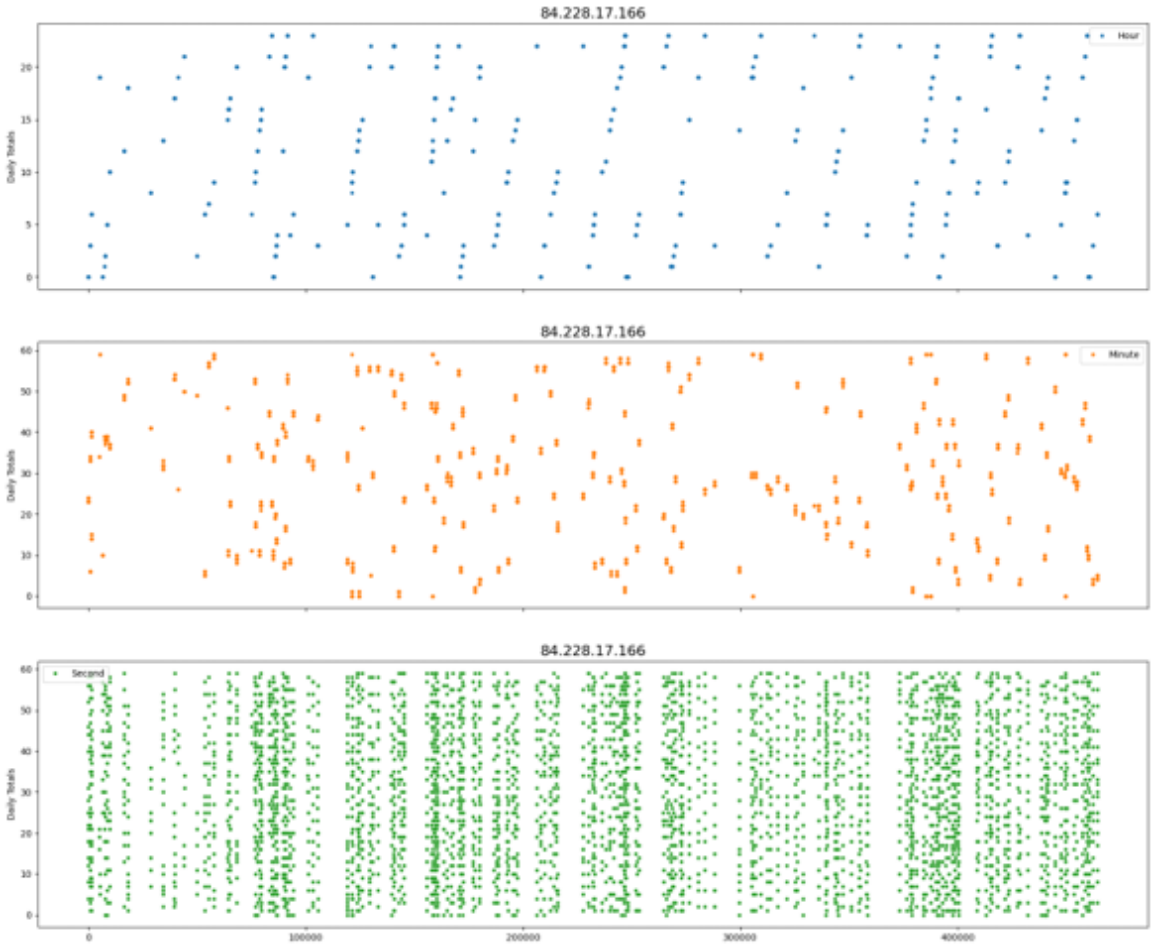

Figure 3.10: Visualizing SSH Brute Force Attacks by IP Address

Thirdly, the IP address was cross-referenced with the ABUSEIPDB website. AbuseIPDB is a project with mission to help make web safer by providing a central blacklist for users to report and find IP addresses that have been associated with malicious activity online [33]. If a record was found with a reported malicious IP on the specified date, the IP was labeled as malicious. However, five IP addresses are not reported in ABUSEIP website, but classified as malicious due to their recurrent times which is less than 4. In total, except 1 IP address, all IP addresses are classified as malicious.

Being the dataset is unbalanced, 299 benign data entries, each associated with one of 94 unique IP addresses are inserted into the dataset artificially. Newly generated

8 unique IPs has username as root where half of them inserted in a SSH brute force attack where root username is recorded. Finally, the dataset has a size of 470758 rows with 470459 malicious and 299 benign IP addresses.

3.2.2 Feature Engineering

Username, Hostname, and Time information is selected out of 8 different features in the raw dataset. The other features do not have any effect in the models. Additionally, five different features were generated for the Machine Learning algorithms from three different variables:

- **FromPrevAttemptHostName:** Time difference between occurrences of the same previous hostname. If no previous hostname is available, a default time difference of 5 seconds is assigned for data entry. In cases where the time difference from the same hostname exceeds 300 seconds, the difference is limited at 300 seconds. This adjustment helps mitigate data normalization errors. As depicted in Figure 3.11, it is clear that there are no instances of benign IPs with a previous same IP time difference of under 5 seconds. Conversely, in Figure 3.12 the density of malicious IPs reaches its peak when the time difference is below 5 seconds. The discrimination will contribute to balance the malicious and benign data.

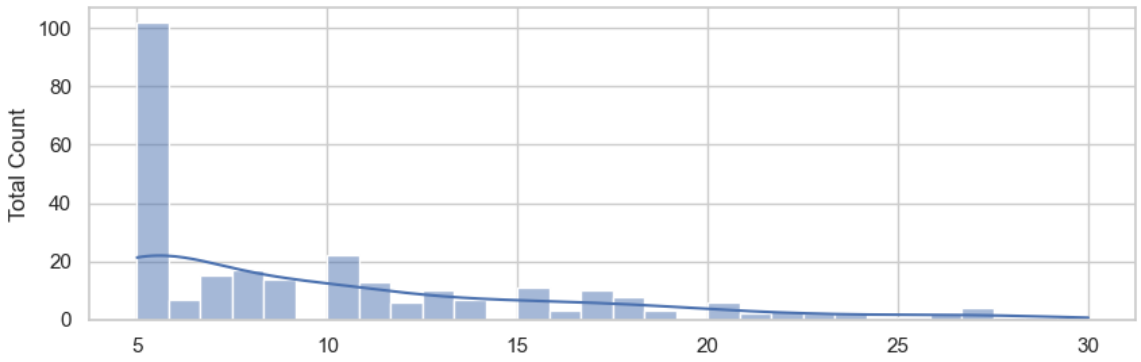

Figure 3.11: Benign IP Time Difference in Seconds

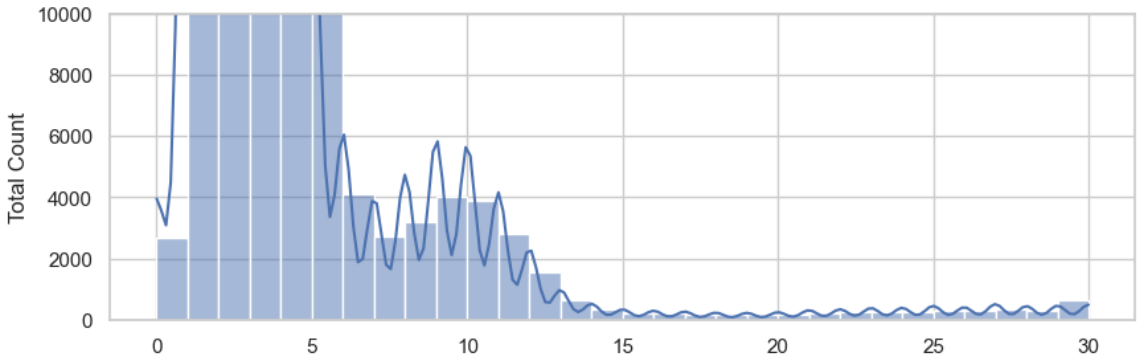

Figure 3.12: Malicious IP Time Difference in Seconds

- **FromPrevAttemptUserName:** Time difference between occurrences of the same usernames with the same hostname. The same computation in FromPrevAttemptHostName is applied to the feature "FromPrevAttemptHostName.
- **HostObservedSofar:** Count of occurrences of the same hostname. As far as same hostname appears, the count of HostObservedSofar increases by 1.
- **UserHostObservedSofar:** Count of usernames from the same hostname. The count of occurrences for each username from the same hostname is tracked in the column 'UserHostObservedSoFar.' As the same hostname reappears, the 'UserHostObservedSoFar' count increases by 1.

- **HostBase:** The first two octets of the hostname. This feature is based on the concept that IP addresses proximate to those previously associated with malicious activities are more prone to become malicious [31]. In our dataset, 25 IP addresses showed up once and than an attack started later as in Figure 3.13

```
df[df['HostName'].str.startswith('101.34')]
```

	Date_Time	PID	UserName	HostName	Label
17094	2022-03-03 15:23:14	113585	root	101.34.58.18	m
186353	2022-03-16 03:00:09	418688	alfresco	101.34.155.60	m
186355	2022-03-16 03:00:10	418688	alfresco	101.34.155.60	m
186505	2022-03-16 03:11:00	418872	admin	101.34.155.60	m
186506	2022-03-16 03:11:01	418872	admin	101.34.155.60	m

Figure 3.13: One Time Showed Up IP and Related SSH Brute Force Attack

Depending on the model's optimal performance, all features, excluding HostBase, are grouped and used either as categorical data or as continuous data.

3.2.3 Dealing with Unbalanced Data

Our dataset comprises 470,758 rows, including 470,459 malicious instances with 4,871 unique IP addresses and 299 benign instances with 94 unique IP addresses. Given the relational nature of the data, traditional methods like upsampling , down-sampling, SMOOTH cannot be applied to our dataset. In our analysis of time difference distribution in Figure 3.11 and 3.12, we observed that there are no instances of benign data with a time difference interval under 4 seconds where the time frame aligns with the peak density of malicious data. This is attributed to the fact that login re-attempts typically take more than 4 seconds, allowing us to disregard data falling within this time frame.

Leveraging this insight, we filtered out malicious data with time difference intervals below 4 seconds. Consequently, the dataset was refined to 39,296 instances, with 1,492 unique malicious IPs, while the benign data remained unchanged.

3.2.4 Cross Validation

In the articles of related works section, training and test data split phase is not clearly explained. Data splitting is an approach for model validation, where a dataset is splitted into two parts: training and testing. The machine learning models are initially fitted on the training set and then validated using the testing set. This process is done to predict the performance of models. There is not a consensus about data splitting ratio, but a commonly used ratio is 80:20, which means 80 percent of the data is for training and 20 percent for testing. Most commonly used approach for data splitting is random sampling, that is, randomly sampling without replacing some rows of the dataset for testing and keeping the rest for training. [34]. However, employing random selection for test data is not an optimal choice in our dataset, given that intrusions in SSH brute force attacks are often connected to previous attempts. Randomly selecting test data could compromise the integrity of our dataset. Additionally, if the test data coincides with the training data, containing the same IP addresses, over-fitting may occur. In order to address this issue, we implemented a time-based splitting approach. Our model utilizes time-based cross-validation, as illustrated in Figure 3.14. We employed a ten-day training period, cross-validated over three days, and conducted testing on the subsequent day.

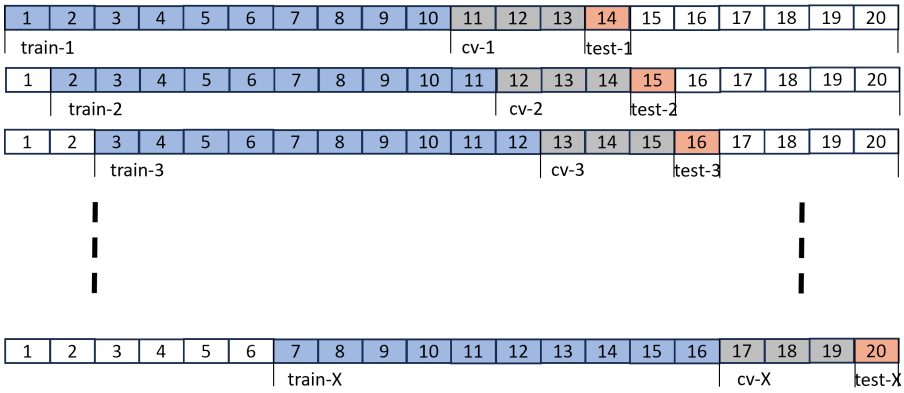

Figure 3.14: Time Based Cross Validation

This approach has contributed to preserving the integrity of the relational data while providing the flexibility to train the data over the intended duration which can be advantageous in dynamic IPs scenarios.

3.2.5 Machine Learning Models

XGboost, Support Vector Machine (SVM), Logistic Regression, and Random Forest models are used to test our data.

- **XGBoost:**

  XGBoost is a fast scalable and memory efficient machine learning algorithm that uses an ensemble of decision trees and gradient boosting to make predictions. It also handles sparse data where in our dataset HostBase feature is one-hot encoded.

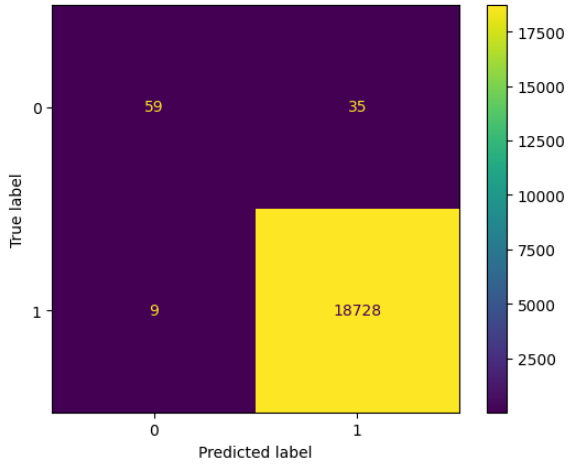

Figure 3.15: Confusion Matrix of XGBoost

The XGBoost model initially classifies 35 instances False Positive (FP) in its first interaction with previously unseen IP addresses. However, upon subsequent attempts from the same IPs, the model rectifies its classification with 100 percent. In the context of False Negatives (FN) model misclassifies 9 instances and out of 1 instance, all usernames are root. Two of the misclassified IP addresses appear only 3 times in the entire dataset. The rest of the FN IPs are misclassified initially, but then rectified. Overall, the XGBoost model performed well with zero occurrences of False Negatives and two instances classified as True Negatives (TN).

- **Decision Tree:** A decision tree algorithm is a supervised machine learning algorithm that uses a tree-like model for predictions. The algorithm works by recursively splitting the data into subsets where each node shows a feature (attribute), each link (branch) shows a decision (rule) and each leaf shows an outcome (categorical or continues value) [35].

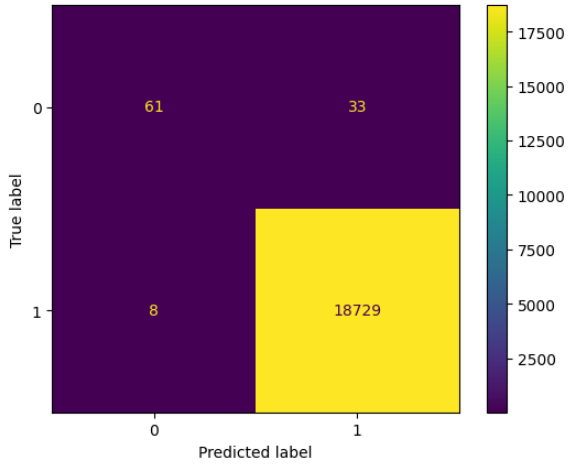

Figure 3.16: Confusion Matrix of Decision Tree

Decision Tree model performed slightly better than XGBoost. The results are the almost the same with XGBoost model. In the form of False Negatives (FN) model misclassifies 8 instances and out of 1 instance, all usernames are root. Two of the misclassified IP addresses appear only 3 times in the entire dataset. The rest of the FN IPs are misclassified initially, but then rectified. Overall, the Decision Tree model performed well with zero occurrences of False Negatives and two instances classified as True Negatives (TN).

- **Logistic Regression:** Logistic regression is a supervised machine learning model that attempts to distinguish between the classes by estimating the probability of an event occurring. It has the advantage of being simple, understandable, and interpretable [36]. In our problem, we applied different feature sets and parameters to the logistic regression. However, the model faces challenges in predicting False Negatives (FN) when the class weight is set to *balance* and struggles with True Negatives (TN) prediction when the weight is set to *None*.

- **Support Vector Machines:** SVM is a supervised machine learning algorithm

where the goal is to find a hyperplane that effectively separates two classes. The main difference between SVM and logistic regression is that logistic regression based on on a probabilistic approach, while SVM relies on statistical methods [37].

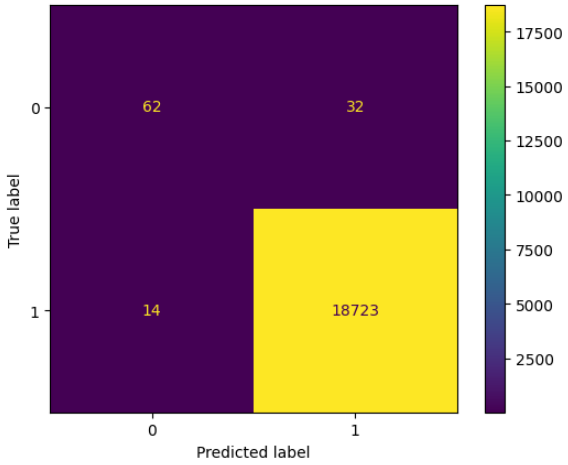

Figure 3.17: Confusion Matrix of SVM

SVM didn't perform well with the continuous data in our dataset. To address this problem, we categorized our features into distinct groups based on dataset observations. Subsequently, we applied one-hot encoding to these features. For instance, 'FromPrevAttemptHostName' and 'FromPrevAttemptUserName' were categorized into three groups: 0 to 4, 4 to 60, and 60 to 300. Similarly, 'UserHostObservedSofar' and 'HostObservedSoFar' were categorized into three groups: 0 to 10, 10 to 20, and over 20. In terms of detecting false positives, SVM performed better than the other models. However, the number of false negatives increased to 14.

- **Random Forest:** Random Forest is a supervised machine learning model, which combines several randomized decision trees and aggregates their predictions by averaging. It is known for its accuracy and ability to deal with small

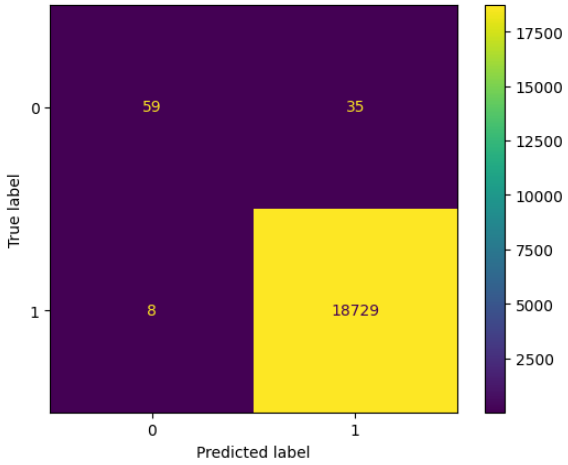

Figure 3.18: Confusion Matrix of Random Forest

sample sizes and high-dimensional feature spaces [38].

The Random Forest model performed as well as XGBoost but exhibited slightly lower performance than a decision tree.

In conclusion, we carefully labeled our data and added new benign data. We have applied feature engineering and generated new features from the known variables. To handle the data imbalance, we removed obviously malicious data based on our analysis. We split the data using a time-based method for evaluation. We tried different machine learning models and checked their performance using confusion matrices.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

Despite being considered old-fashioned, SSH brute force attacks persist a potent threat, exploiting weak passwords and inadequate security measures. Studies addressing this issue primarily concentrate on network-based machine learning detection due to its scalability. Most of the studies before 2020 used utilized KDD Cup'99 and NSL-KDD datasets, and ICS IDS 2018 dataset after 2020. Researchers have explored various machine learning and deep learning models with different feature configurations. Interestingly, similar results have been achieved using as few as 2 features or as many as 80 features. The consensus for a new, comprehensive dataset is widely acknowledged. Given the inherent imbalance in many datasets, achieving 99.99 percent accuracy may not carry significant meaning. We believe that, the most explanatory metric is confusion matrix for model comparison using the same dataset.

To the extend of our research, our study is the first in host-base machine learning detection. It's low dimension makes it fast, interpretable and easy to visualize. During

our data analysis, a noteworthy observation is discovered – a preliminary, one-time scan from a similar IP often precedes a SSH brute force attack. Recognizing this initial scan becomes crucial for monitoring potential future attacks, whether in honeypots or real environments.

In addition, a thorough examination of IP addresses from the ABUSEIPDB website revealed five IPs that had not been reported at the time of our data collection. This underscores a key advantage of investigating failed login logs. Such information can significantly enhance the detection of stealth attacks, characterized by their slow and steady nature, making them challenging to identify.

We introduced five distinct features to our dataset, including the time difference between the same hostname and username, the aggregation of count of hostnames and usernames, and the first two octets of the IP addresses. What sets our approach apart is the implementation of a time-based splitting strategy, a method not observed in previous studies. This approach safeguards the integrity of data and also mitigates the risk of over-fitting.

During the evaluation phase, tree-based models demonstrated better performance than the other models, but our models faced challenges in terms of detecting false positives. This bias is attributed to the model's tendency to classify newly seen data as malicious, especially the first appearance of an IP address. However, with subsequent attempts from the same IPs, the model rectifies its classification and delivers outstanding performance.

In conclusion, our research distinguishes itself through a unique methodology that differs from previous studies in problem approach, feature engineering, and data splitting. We achieved comparable or even superior results compared to network-based approaches.

4.2 Future Work

In considering future work, we've identified two remarkable areas for exploration. Firstly, while the applied machine learning and deep learning models exhibit proficiency in detecting machine-generated attacks, their efficacy may be limited when dealing with human-interacted attacks, particularly insider threats. To address this, we are considering the inclusion of password similarity analysis, which could significantly enhance the detection of insider threats.

The second area for improvement involves the current models classifying instances individually without capturing the relationships or connections between entities. To address this limitation, the integration of graph-based neural networks emerges as a promising solution. This approach could be effective in link prediction, and botnet detection where understanding the relationships between entities is essential for uncovering hidden patterns and enhancing overall model performance.

Bibliography

[1] N. Sultana, N. K. Chilamkurti, W. Peng, and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, pp. 493–501, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:67279885

[2] B. A. Forouzan, "Remote login: Telnet and ssh," in *TCP/IP Protocol Suite Fourth Edition*, 2010, pp. 610–628.

[3] "Top 10 Web Application Security Risks," [https://owasp.org/www-project-top-ten/], 2021.

[4] "Threat horizons cloud threat intelligence," 2022. [Online]. Available: https://services.google.com/fh/files/blogs/gcat_threathorizons_full_july2022.pdf

[5] C. Pamungkas, P. Hendradi, D. Sasongko, and A. Ghifari, "Analysis of brute force attacks using national institute of standards and technology (nist) methods on routers," *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, vol. 5, no. 2, pp. 115–125, 2023.

[6] "Password cracking speed," [https://thesecurityfactory.be/password-cracking-speed/], 2020.

[7] Z. Ahmad, A. S. Khan, W. S. Cheah, J. bin Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225153435

[8] "Secure Shell," [https://en.wikipedia.org/wiki/Secure_Shell], 2023.

[9] "SSH History," [https://www.ssh.com/about/history/], 2023.

[10] S. K. Wanjau, G. M. Wambugu, and G. N. Kamau, "Ssh-brute force attack detection model based on deep learning," *International Journal of Computer Applications Technology and Research*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238150632

[11] "Brute Force Attack," [https://campus.barracuda.com/product/webapplicationfirewall/doc/42049329/brute-force-attack/], 2019.

[12] R. Kirushnaamoni, "Defenses to curb online password guessing attacks," *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 317–322, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:15146905

[13] "Brute Force Attack: Definition and Examples," [https://usa.kaspersky.com/resource-center/definitions/brute-force-attack], 2023.

[14] S. Wickramasinghe, "Brute Force Attacks: Techniques, Types & Prevention," [https://www.splunk.com/en_us/blog/learn/brute-force-attacks.html], 2023.

[15] N. Tiwari and N. Hubballi, "Secure socket shell bruteforce attack detection with petri net modeling," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 697–710, 2023.

[16] Z. Talab, "11 Brute-force Attack Tools for Penetration Test," [https://geekflare.com/brute-force-attack-tools/], 2022.

[17] B. Toulas, "New Linux malware brute-forces SSH servers to breach networks," [https://www.bleepingcomputer.com/news/security/new-linux-malware-brute-forces-ssh-servers-to-breach-networks/], 2022.

[18] ——, "Legion: New hacktool steals credentials from misconfigured sites," [https://www.bleepingcomputer.com/news/security/legion-new-hacktool-steals-credentials-from-misconfigured-sites/], 2023.

[19] R. Hofstede, A. Pras, A. Sperotto, and G. D. Rodosek, "Flow-based compromise detection: Lessons learned," *IEEE Security & Privacy*, vol. 16, no. 1, pp. 82–89, 2018.

[20] Y. Yang, K. C. Yeo, S. Azam, A. Karim, R. Ahammad, and R. Mahmud, "Empirical study of password strength meter design," *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 436–442, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:220568597

[21] P. Khandait, N. Tiwari, and N. Hubballi, "Who is trying to compromise your ssh server? an analysis of authentication logs and detection of bruteforce attacks," *Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229716056

[22] M. D. Hossain, H. Ochiai, F. Doudou, and Y. Kadobayashi, "Ssh and ftp brute-force attacks detection in computer networks: Lstm and machine learning approaches," in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, 2020, pp. 491–497.

[23] M. A. Kristyanto, I. Krisnahati, F. Rawung, D. Dzhalila, B. D. Nurwibawa, W. Murti, B. A. Pratomo, and A. M. Shiddiqi, "Ssh bruteforce attack classification using machine learning," *2022 10th International Conference on Information and Communication Technology (ICoICT)*, pp. 116–119, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253047007

[24] N. Alotibi and M. Alshammari, "Deep learning-based intrusion detection: A novel approach for identifying brute-force attacks on ftp and ssh protocol," *International Journal of Advanced Computer Science and Applications*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259311075

[25] J. L. Leevy, J. T. Hancock, R. Zuech, and T. M. Khoshgoftaar, "Detecting cybersecurity attacks using different network features with lightgbm and xgboost learners," *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*, pp. 190–197, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:231683552

[26] Y. Wu, P. Cao, A. Withers, Z. T. Kalbarczyk, and R. K. Iyer, "Mining threat intelligence from billion-scale ssh brute-force attacks," 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:211139163

[27] J. Hancock, T. M. Khoshgoftaar, and J. L. Leevy, "Detecting ssh and ftp brute force attacks in big data," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021, pp. 760–765.

[28] G. C. Fernández and S. Xu, "A case study on using deep learning for network intrusion detection," in *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, 2019, pp. 1–6.

[29] M. F. K. Shah, M. Md-Arshad, A. A. Samad, and F. A. Ghaleb, "Comparing ftp and ssh password brute force attack detection using k-nearest neighbour (k-nn) and decision tree in cloud computing," *International Journal of Innovative Computing*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258997261

[30] C. Yao, X. Luo, and A. N. Zincir-Heywood, "Data analytics for modeling and visualizing attack behaviors: A case study on ssh brute force attacks," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–8.

[31] D. Likhomanov and V. Poliukh, "Predicting malicious hosts by blacklisted ipv4 address density estimation," in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 2020, pp. 102–109.

[32] "utmpdump(1) — linux manual page." [Online]. Available: https://man7.org/linux/man-pages/man1/utmpdump.1.html

[33] "Abuseipdb." [Online]. Available: https://www.abuseipdb.com/

[34] V. R. Joseph, "Optimal ratio for data splitting," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 15, pp. 531 – 538, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:246634017

[35] H. Patel and P. Prajapati, "Study and analysis of decision tree based classification algorithms," *International Journal of Computer Sciences and Engineering*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:59515422

[36] "What is logistic regression?" [Online]. Available: https://www.ibm.com/topics/logistic-regression

[37] "Guide on support vector machine (svm) algorithm." [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

[38] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, pp. 197 – 227, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:14518730