




Attackers' Profiling Based on Multi-Attack Patterns in SSH Service

Kriti Majumdar¹, Nitesh Kumar²^a, Anand Handa²^b and Sandeep K. Shukla²^c

¹Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, India

²C3i Hub, Indian Institute of Technology, Kanpur, India

{kritim, niteshkr, ahanda, sandeeps}@cse.iitk.ac.in

Keywords: SSH Security, Attackers' Profiling, Machine Learning.


Abstract: In the realm of cyber security, profiling attackers' behaviors provides critical insights that can enhance defensive strategies and improve the security of network services. This paper introduces a methodology for profiling attackers through the analysis of multi-attack patterns on Secure Shell (SSH) services. We develop a comprehensive framework that utilizes both predefined rule-based techniques and advance machine learning techniques to classify attack types and link them to specific attacker profiles. By analyzing logs from SSH services that comprise various SSH attack incidents, we identify common and distinct behavioral patterns that help in predicting future attacks and identifying the likely attributes of attackers. Our attacker profiling system addresses the five key 'wh' questions: who is causing the attack, when the attack occurred, how the attack was executed, from where the attack originated, and what type of attack was carried out. The results demonstrate that our approach is highly effective not only at detecting security threats but also at profiling them, which allows for the development of specific and effective countermeasures. This methodology significantly enhances the ability to anticipate and mitigate a wide range of attack vectors, strengthening overall cybersecurity resilience.


1 INTRODUCTION


SSH emerged in 1995 from the efforts of Tatu Ylönen, a visionary researcher at the University of Helsinki, who was propelled by a password-sniffing attack on his university's network to create a protocol that could safeguard remote login sessions and other network interactions over inherently insecure infrastructures (Barrett, 2005). However, as the utility of SSH has grown, so too has its attractiveness as a target for cyber attackers. Kaspersky Security Services (Securelist, 2023) reported in their 2023 Threat Report that in the first half of 2023, 2.09% of all password brute-force attempts recorded on honeypots targeted SSH services, while the majority targeted less secure protocols like Telnet. Despite the lower frequency compared to other protocols, the potential impact of SSH attacks on secure systems remains significant. The intensity and sophistication of these attacks are underscored by their latent consequences. Recent cybersecurity findings, including the 'Terrapin' vulnerability highlighted by arsTechnica in 2023 (Technica,

2024), affected approximately 11 million Internet-exposed SSH servers globally, emphasize the critical importance of monitoring SSH services. A successful SSH attack can lead to significant breaches, such as the T-Mobile incident in 2021 (Keytos, 2024), where personal data of over 54 million customers was compromised through SSH channels. In another significant security incident, GoDaddy disclosed that SSH credentials of nearly 28,000 users were compromised during a data breach in October 2019 (Roy, 2020). Similarly, historical breaches like the RSA Security incident in 2011 and Operation Aurora during 2009-2010 reveal that even well-secured systems are not immune to the misuse of SSH keys (Keytos, 2024).

The rapid evolution of cyber threats targeting SSH has introduced novel exploitation methods such as 'Proxyjacking' (Team, Year), where attackers hijack a victim's network bandwidth to generate passive income. Researchers of the Akamai Security Intelligence Response Team (Cimpanu, 2024) discovered that this emerging threat involves malicious actors leveraging compromised SSH servers to enroll devices into peer-to-peer (P2P) proxy networks like Peer2Profit without the owner's knowledge. This type of attack not only highlights the sophistication and

^a <https://orcid.org/0000-0003-0998-0925>

^b <https://orcid.org/0000-0003-0075-1165>

^c <https://orcid.org/0000-0001-5525-7426>

stealth of modern cyber threats but also underscores the financial motivations driving attackers to seek less detectable methods of exploitation.

SSH attacks can be systematically categorized into two main types: those that occur before SSH is compromised (BSC) and those that exploit after SSH is compromised (ASC). The pre-compromise attacks include brute force (BF), dictionary (DA) (Kaspersky, 2024), scanning (SC), denial of service (DoS) (Cloudflare, 2024a), and mixed activities (MA). These attacks aim to gain unauthorized access to SSH servers. Once SSH is compromised, the activities expand into more sophisticated and varied forms, such as reconnaissance, credential dumping, lateral movement, privilege escalation, backdoor installation, command and control (C2) operations, data exfiltration, man-in-the-middle (MitM) attacks (TechTarget, 2024), ransomware deployment (Cartwright and Bunter, 2019), phishing (Cloudflare, 2024b), DoS attacks and many more. These post-compromise actions utilize the compromised session to inflict further damage and penetrate deeper into network systems, indicating a critical phase where the attackers exploit the initial access to maximize impact.

Our paper addresses the gap in existing cyber security defenses by proposing a framework for attacker profiling based on multi-attack patterns observed in SSH services. Our analysis begins with a thorough review of the current challenges in SSH security and an examination of previous efforts to detect attacks on SSH. In addition to implementing predefined rule-based detection methods, we identify multiple attack patterns through meticulous analysis of recorded attack scenarios on SSH services. Building on this foundation, our methodology integrates sophisticated feature extraction from SSH log data with ensemble machine learning classifiers, enhancing our ability to discriminate between various types of attack behaviors effectively. The integration of rule-based techniques with advanced machine learning allows for a more robust defense mechanism, capable of adapting to both known and emerging threats. The implications of our findings are profound, suggesting that a nuanced understanding of attacker profiles can facilitate the development of more adaptive and dynamic security systems which are not only capable of withstanding current threats but are also agile enough to evolve in response to emerging tactics and strategies used by cyber attackers. Our approach holds the promise of significantly bolstering the defenses of SSH services against the sophisticated and continuously evolving landscape of cyber threats.

2 RELATED WORKS

In this section, we discuss the existing approaches that have been implemented to defend against potential SSH attacks. These approaches can generally be categorized into two types: Rule-based Approaches and Machine Learning (ML)-based Approaches.

Rule Based Approaches: In rule-based systems, researchers establish predefined rules to detect attacks or malicious activities. These rules are typically derived from known threats, user behaviors, and other indicative metrics that can be monitored through network logs. An exemplar of this methodology, as discussed in (Park et al., 2021), presents a model designed to detect and mitigate SSH brute-force attacks by analyzing router-generated logs. The proposed model aggregates and assesses logs indicative of failed SSH access attempts, extracting critical information such as IP addresses, timestamps, and error messages. This data forms the foundation for applying rules, where each element is weighted according to its assessed threat level. Upon detecting an attack, the model logs the involved IP addresses and restricts further access from these sources to prevent unauthorized activities. Additionally, the model employs a dynamic blacklist to restrict access from identified malicious IPs, which is continuously updated based on attack frequency, detection days, and geographical origin. The efficacy of this model is validated through a comprehensive analysis of logs collected over one year.

Another exemplar, as discussed in (Fahrnberger, 2022), proposes a Condition Monitoring System (CMS) designed to monitor and assess the risk of SSH brute-force attacks in real time. The CMS employs predefined rules combined with statistical analysis to evaluate the threat level of each failed authentication attempt. A distinctive feature of this system is its dynamic approach to risk assessment, which adjusts threshold values based on historical data and the evolving nature of attack patterns. The CMS dynamically updates its risk evaluation parameters to provide real-time alerts and notifications when suspicious activities are detected, enhancing the system's responsiveness to emerging threats. The effectiveness of the CMS was proven through experiments using real-world SSH log data collected over a year.

ML Based Approaches: In (Agghey et al., 2021), the authors explore the use of machine learning classifiers to detect username enumeration attacks (SSL, 2024) (UAE) on SSH protocols. These attacks serve as a preliminary step to brute-force attacks, enabling attackers to gather valid usernames. The study collected data from a controlled network environment

and utilized four machine learning classifiers—k-nearest neighbor (KNN), naïve Bayes (NB), random forest (RF), and decision tree (DT)—to evaluate their effectiveness. The researchers used a total of seven features, including packet duration, packet length, and port information etc. Though the findings indicated that their machine learning models could successfully identify username enumeration attacks, with improved performance, the study lacked testing with real-world data.

The authors in (Hynek et al., 2020) presents a novel approach for detecting SSH brute-force attacks in high-speed networks using machine learning. The detection system architecture includes data preprocessing, an ML-based detector, and a knowledge base for post-processing detected events. Unlike host-based methods, this network-level approach captures detailed traffic information, including packet lengths and inter-packet times etc. The authors created a dataset from real network traffic with over 30,000 labeled SSH biflow records, half of which are brute-force attacks. They evaluated over 70 features and selected 11 that provided good detection accuracy using the AdaBoosted Decision Tree model.

The paper described in (Wanjau et al., 2021) proposes a CNN-based model to detect brute-force attacks on SSH logs. They identified the increasing difficulty of detecting these attacks due to the high speed and volume of network traffic, which often obscures malicious activities. The model is trained using the CIC-IDS 2018 dataset, which includes contemporary benign and malicious network activities. The researchers employ feature selection and data normalization techniques to preprocess the data, transforming it into images suitable for CNN processing. The results show that the CNN-based model significantly outperforms traditional machine learning methods such as Naive Bayes, Logistic Regression, Decision Tree, k-Nearest Neighbour, and Support Vector Machine in detecting SSH brute-force attacks.

The paper described in (Garre et al., 2021) proposes a machine learning-based approach for detecting SSH botnet infections. This research addresses the exponential increase in botnet activity, exacerbated by zero-day attacks and obfuscation techniques, which traditional detection methods struggle to manage. The authors utilized High-Interaction Honeypots (HIH) to capture detailed attack behaviors and log data, creating a dataset consisting of executed commands and network information during SSH sessions. This dataset was used to train a supervised learning model to identify botnet infections during the initial infection phase. This study underscores the poten-

tial of machine learning techniques in enhancing early botnet detection and preventing compromised devices from participating in malicious activities.

Our observations on the past approaches to SSH attack detection are as follows:

- Most proposed solutions, regardless of the technology used, focus on detecting individual attacks separately. To the best of our knowledge, none of them consider the entire spectrum of attack scenarios possible on SSH.
- Rule-based approaches are less complex to implement and can effectively detect malicious behavior. However, they are not robust against sophisticated and obfuscated attack strategies.
- Rule-based approaches are less complex to implement and can effectively detect malicious behavior. However, they are not robust against sophisticated and obfuscated attack strategies.

3 PROPOSED METHODOLOGY

In this section, we describe the architecture of our proposed SSH log-based attack detection and classification system for attacker profiling. Our methodology integrates rule-based techniques with machine learning algorithms to create a robust, multi-faceted defense mechanism. The architecture, illustrated in Figure 1, is designed to parse, process, and analyze SSH logs using a dual analytical strategy. It employs predefined security rules for immediate threat identification, while simultaneously using predictive machine learning models for deeper analysis and classification.

Data Collection – The raw SSH log data for four months, spanning from June 16, 2021, to October 10, 2021, are collected from an SSH server hosted in the cloud. Additionally, we gather Cowrie log data for six months, from March 1, 2023, to August 23, 2023. Both datasets include various types of attacks, which could be either manual or automated. We designate the Cowrie Honeypot data as D0, which comprises 5,941,378 log entries. We divide the SSH server-generated log dataset into two parts: D1 and D2. D1 contains data from June 16, 2021, to September 17, 2021, amounting to 3,312,998 log entries, while D2 encompasses data from September 18, 2021, to October 10, 2021, with 199,853 log entries. Initially, we use the D0 dataset for pattern-based feature extraction. Ultimately, the D1 dataset is employed for initial training and testing, and the D2 dataset is used to evaluate the performance of predictive models on previously unseen data.

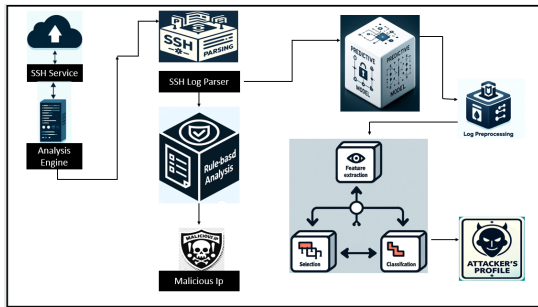


Figure 1: Architecture of our Proposed Methodology.

Log Parsing – Log parsing is the essential first step in our analysis engine. During this phase, we process the raw SSH log data, which is typically large and cluttered, to filter out unnecessary content. This filtering removes irrelevant details such as routine server starts, protocol initiations, and other non-essential data, while formatting the entries by eliminating redundant information. This helps us focus on the critical events that could indicate security threats or attempted intrusions. After cleaning, the data is organized and prepared for deeper analysis using rule-based and machine learning algorithms.

Rule-Based Analysis – In the rule-based analysis phase of our system, we apply predefined security rules to the parsed SSH log data to flag activities suggestive of potential threats. These rules are based on specific criteria and known threat indicators.

One rule targets IP addresses known for malicious activities. By cross-referencing log entries with external databases such as Maxmind's list (MaxMind, Inc., 2024b) and Ipsum by Stampum (Stampar, 2024), we identify and blacklist threats from these known problematic sources. Geolocation analysis is another key aspect of our rule-based approach. Utilizing Maxmind's GeoIP2-City and GeoIP2Country (MaxMind, Inc., 2024a) databases, we convert IP addresses to geolocations. This geographic information is compared against the CTI's list of ten countries known for heightened cybersecurity risks (CyberProof, 2024), augmented by additional countries based on geopolitical relations with India. Any activity from IP addresses located in these countries is flagged for further review.

Additional rules detect suspicious behaviors such as rapid port changes, excessive failed login attempts from one or multiple IPs in quick succession, and repeated failed attempts to access the root account. We also monitor unusual login times and the use of rarely used or new usernames, which can indicate coordinated attacks or unauthorized access attempts. Through this rule-based analysis, our system swiftly identifies and responds to a range of potential threats,

enabling subsequent analysis through machine learning models to be effective on the most pertinent security events.

ML-Based Analysis – In this section, we discuss the ML-based analysis which consists of preprocessing, feature engineering, feature selection, and classification methodology. In the preprocessing stage, we segment the dataset by unique IP addresses. Each set of log entries associated with a distinct IP address is grouped and stored in individual files. For instance, in dataset D1, which contains 17,599 unique IP addresses, we generated the same number of separate files following preprocessing. We then further process these files to extract relevant features. Feature extraction is crucial in the development of our model. To train our machine learning models for multilabel classification of attacks, we primarily focus on extracting two key types of features – statistical information-based and pattern-based. These features are integrated together to enhance the training and testing tasks, providing a robust foundation for accurately identifying various attack vectors. We explain both the types of features as follows –

Statistical Information-Based Features – In our classification model, features based on statistical information are primarily derived from network interactions. These features capture various dimensions of network activity linked to individual IP addresses. Specifically, we track the total number of connection requests made by an IP in a single day (Feature 1) and monitor invalid username attempts by that IP on that day (Feature 2). We also measure the total number of failed password attempts for valid users (Feature 3) and for all attempts including both valid and invalid usernames made by an IP in a day (Feature 4). Additionally, we aggregate the number of failed login attempts (Feature 5) and successful login attempts (Feature 6), along with the ratio of failed to successful attempts (Feature 7) for an IP on a particular day, providing a comprehensive view of authentication outcomes. Furthermore, we calculate the total number of instances where the maximum authentication limit is exceeded for an IP in a day (Feature 8) and count the number of disconnect requests initiated by an IP in a day (Feature 9). Together, these features form a comprehensive dataset that aids in detecting and analyzing suspicious activities indicative of potential security threats.

Pattern-Based Features – Pattern-based features involve recognizing and encoding specific behavioral patterns evident in SSH log entries.

Phase 1 – In this phase, we work with dataset D0. By analyzing the logs of this dataset, we found out 2 common patterns used by the attackers to per-

Algorithm 1: Detection of Patterns found in Phase 1.

```

Input : log_entries
Output: Pattern Identifier (1 or "Unknown")
Procedure DetectPatterns (log_entries) :
    // Check Pattern I
    if "Connection request" followed by
        repeated "Login Failed" followed by
        "Successful Login" followed by
        repetition of "Executing Unix
        commands" followed by "Remote
        close" followed by "Channel Open"
        followed by "Executing Unix
        commands" then
        | return 1;
    end
    // Check Pattern II
    if "Connection request" followed by
        "Successful Login" followed by
        repetition of "Executing Unix
        commands" followed by "Remote
        close" followed by "Channel Open"
        followed by "Executing Unix
        commands" then
        | return 1;
    end
    return "Unknown";
    
```

```

Input : log_entries
Output: Pattern Identifier (1 or "Unknown")
Procedure DetectPatterns (log_entries) :
    // Check Pattern 1
    if "Connection request" → "Failed
    password" → "disconnect" or
    "Connection request" → "Invalid user"
    → "Failed password" → "disconnect"
    then
        | if Repeated several times then
        | | return 1;
        | end
    end
    // Check Pattern 2
    if consecutive "Connection request" →
    "Invalid user" and/or "Failed
    password" then
        | return 1;
    end
    // Check Pattern 3
    if "Connection request" → consecutive
    "Failed password" for a user →
    "disconnect" or "max auth attempts
    exceeded" then
        | return 1;
    end
    
```

form attack or execute any sort of malicious activity. Additionally, we identify specific commands used by attackers for malicious purposes. These commands were categorized based on their intended actions and mapped to corresponding attack or malicious activity categories. Algorithm 1 delineates the characteristics of two common patterns, referred to as Pattern I and Pattern II, identified in these logs, and explicates the method used for their detection. Pattern I demonstrates how, after several attempts, an adversary successfully authenticates and begins executing commands for malicious purposes. In contrast, Pattern II depicts scenarios where the attacker logs in effortlessly and immediately runs commands. This nature seems to be benign but based on the commands they execute, we should decide whether this is a normal user behavior or a malicious activity. A few commands frequently used by attackers in our logs, along with their purposes and associations with malicious activities or attacks are as follows:

- Activity: Reconnaissance commands
 1. CPU Information: `cat /proc/cpuinfo — grep name — wc -l`
 2. System information: `uname -m , uname -a`
- Activity : Privilege Escalation commands

1. searching for SUID binaries: `find / -perm -o+w -type f 2>/dev/null`
 2. trying to list all users with UID 0 (root): `awk -F: '($3 == 0) {print}' /etc/passwd`
- Activity : Changing SSH Keys commands
 1. Removing legitimate SSH keys: `echo "" >/.ssh/authorized_keys`

Phase 2 – When analyzing the D1 dataset, we discover that attacks can occur in numerous other ways in real-life scenarios. We conclude that, since the credentials of honeypots are typically simple and easy to guess, adversaries can more easily compromise or authenticate to them. In contrast, real-time SSH servers often have stronger credentials, requiring significantly more effort to perform malicious activities. Therefore, the patterns identified in the Cowrie log dataset (D0) are not universally applicable to all real-time scenarios. Additionally, it was challenging to accurately distinguish patterns that exclusively exhibit benign behavior. As a result, a model trained on this dataset may lack the robustness required to account for all potential attack vectors and variations in real-world conditions. Consequently, we chose not to train our model using dataset D0 to ensure comprehensive coverage and reliability in detecting and responding

Algorithm 2: Detection of Patterns found in Phase 2.

```

Procedure DetectPatterns (log_entries):
  // Check Pattern 4
  if "Connection request" → consecutive
    "Failed password for root" →
    "disconnect" or "max auth attempts
    exceeded" then
    | return 1;
  end
  // Check Pattern 5
  if "Connection request" → "Invalid
  user" → "Failed password" then
    | if Repeated several times then
    | | return 1;
    | end
  end
  // Check Pattern 6
  if Several consecutive "Connection
  request" entries then
    | return 1;
  end
  // Check Pattern 7
  if several consecutive "Connection
  request" → "Invalid user" or "Failed
  password" → "disconnect" → several
  consecutive "Connection request" then
    | return 1;
  end
  // Check Pattern 8
  if Any of patterns 1–7 followed by
  "Accepted password" and "User
  executed command:" then
    | return 1;
  end
  // Check Pattern 9
  if None of patterns 1–8 and "Connection
  request", "Invalid user", or "Failed
  password" present then
    | return 1;
  end
  // Check Pattern 10
  if "Connection request" → "Accepted
  password" without any "Invalid user"
  or "Failed password" then
    | return 1;
  end
  return "Unknown";

```

to a wider range of security threats. By systematically analyzing both attack and normal SSH logs in D1, we have identified ten distinct patterns that effectively characterize more complex user behavior and interactions with the SSH service. Algorithm 2 illus-

trates the characteristics of the patterns identified in Phase 2 and outlines the method employed for their detection. The following patterns were identified:

- **Pattern 1 : Sequential Connection Attempts with Failed Authentication and Disconnect Request**
Attacks or Activities Associated with this Pattern: This pattern indicate brute-force (BF) attacks, characterized by repeated attempts to guess passwords.
- **Pattern 2: Multiple SSH Connection Attempts Followed by Multiple Authentication Failures**
Attacks or Activities Associated with this Pattern: This pattern indicate brute-force (BF) attacks, characterized by repeated attempts to guess passwords.
- **Pattern 3: Persistent SSH Connection Attempts with Known Username**
Attacks or Activities Associated with this Pattern: This pattern is associated with dictionary attacks (DA), which use predefined lists of usernames and passwords to gain access.
- **Pattern 4: Repetitive SSH Connection Attempts Targeting the Root Account**
Attacks or Activities Associated with this Pattern: This pattern is associated with dictionary attacks (DA), which use predefined lists of usernames and passwords to gain access.
- **Pattern 5: Rapid Sequential SSH Connection Attempts with Authentication Failures**
Attacks or Activities Associated with this Pattern: This pattern represents denial of service (DoS) attacks, aimed at overwhelming the SSH server with a flood of connection requests.
- **Pattern 6: Multiple Connection Requests Without Login Attempts**
Attacks or Activities Associated with this Pattern: This pattern indicate scanning activities, where attackers probe for server vulnerabilities by rapidly initiating connection requests.
- **Pattern 7: Multiple SSH Connection Attempts with Intermittent Authentication Failures and Disconnects**
Attacks or Activities Associated with this Pattern: This pattern indicate scanning activities, where attackers probe for server vulnerabilities by rapidly initiating connection requests.
- **Pattern 8: SSH Compromise Leading to Severe Exploitation**
Attacks or Activities Associated with this Pattern: This pattern represents attacks after

Table 1: 10-fold Cross Validation Results in %.

Classifier	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10	Mean CV Accuracy
Random Forest	99.43	99.47	99.48	99.49	99.47	99.48	99.46	99.47	99.96	99.47	99.57
Decision Tree	98.30	98.43	98.28	98.50	98.14	98.28	98.14	98.50	98.71	98.57	98.39
SVM	97.40	97.73	97.11	97.94	97.93	97.23	97.10	97.80	97.63	97.59	97.55
KNN	98.65	98.50	98.72	98.58	98.15	98.87	98.51	98.29	98.51	98.22	98.50
Logistic Regression	98.86	99.43	98.72	99.36	98.79	98.86	98.93	99.00	99.07	98.15	98.82
Gradient Boosting	98.57	98.64	98.36	98.57	98.43	98.36	98.36	98.57	98.91	98.79	98.55

Table 2: Test Results in % on D1 for various type of attack scenarios.

Test Results in % on D1: Brute Force Attack							Test Results in % on D1: Dictionary Attack						
Classifiers	RF	DT	SVM	KNN	LR	GB	Classifiers	RF	DT	SVM	KNN	LR	GB
Accuracy	99.47	99.45	98.58	99.41	99.44	99.23	Accuracy	99.38	99.33	98.84	99.37	99.28	99.36
Precision	99.95	99.93	99.79	99.84	99.91	99.79	Precision	99.85	99.78	99.62	99.83	99.71	99.80
F1-score	99.46	99.45	98.55	99.41	99.44	99.23	F1-score	99.38	99.33	98.83	99.37	99.27	99.36
TPR	98.98	98.97	97.35	98.97	98.96	98.67	TPR	98.91	98.88	98.06	98.91	98.84	98.91
FPR	0.05	0.07	0.20	0.16	0.09	0.21	FPR	0.15	0.22	0.37	0.17	0.29	0.20
TNR	99.95	99.93	99.80	99.84	99.91	99.79	TNR	99.85	99.78	99.63	99.83	99.71	99.80
FNR	1.02	1.03	2.65	1.03	1.04	1.33	FNR	1.09	1.12	1.94	1.09	1.16	1.09
Test Results in % on D1: Scanning Attack							Test Results in % on D1: DoS Attack						
Classifiers	RF	DT	SVM	KNN	LR	GB	Classifiers	RF	DT	SVM	KNN	LR	GB
Accuracy	99.96	99.94	99.73	99.95	99.91	99.93	Accuracy	99.43	99.30	99.10	99.33	99.18	99.42
Precision	99.96	99.90	99.87	99.93	99.91	99.92	Precision	99.33	99.12	98.98	99.30	99.27	99.31
F1-score	99.96	99.94	99.73	99.95	99.91	99.93	F1-score	99.43	99.30	99.10	99.33	99.18	99.42
TPR	99.98	99.97	99.58	99.97	99.91	99.93	TPR	99.52	99.48	99.22	99.35	99.09	99.52
FPR	0.07	0.10	0.13	0.07	0.09	0.08	FPR	0.67	0.89	1.02	0.70	0.73	0.69
TNR	99.93	99.90	99.87	99.93	99.91	99.92	TNR	99.33	99.11	98.98	99.30	99.27	99.31
FNR	0.02	0.03	0.42	0.03	0.09	0.07	FNR	0.48	0.52	0.78	0.65	0.91	0.48
Test Results in % on D1: Mixed Activity							Test Results in % on D1: After SSH compromise						
Classifiers	RF	DT	SVM	KNN	LR	GB	Classifiers	RF	DT	SVM	KNN	LR	GB
Accuracy	99.23	99.19	97.97	99.37	99.18	99.13	Accuracy	98.15	97.72	97.34	98.06	98.14	98.09
Precision	99.30	99.20	98.16	99.28	99.17	99.26	Precision	98.02	97.65	97.64	97.99	98.16	98.11
F1-score	99.23	99.19	97.97	99.37	99.18	99.13	F1-score	98.15	97.91	97.33	98.06	98.14	98.09
TPR	99.16	99.18	97.78	99.37	99.18	99.13	TPR	98.28	98.17	97.02	98.13	98.11	98.07
FPR	0.70	0.80	1.84	0.72	0.83	0.74	FPR	1.99	2.36	2.34	2.00	1.84	1.89
TNR	99.30	99.20	98.16	99.28	99.17	99.26	TNR	98.01	98.64	97.66	98.00	98.16	98.11
FNR	0.84	0.92	2.22	0.55	0.92	1.00	FNR	1.72	1.83	2.98	1.87	1.89	1.93

SSH gets compromised (ASC), where attackers achieve unauthorized access through a successful login and then execute commands on the compromised system.

• Pattern 9: Randomized Connection Attempts and Authentication Failures

Attacks or Activities Associated with this Pattern: This pattern indicates mixed activity (MA), involving random combinations of connection requests, failed login attempts, and disconnect requests.

• Pattern 10: Benign SSH Connection and Interaction

Attacks or Activities Associated with this Pattern: Lastly, Pattern 10 represents benign activity, characterized by legitimate connection requests, successful logins, and normal command execution followed by a proper disconnect or session timeout.

Data Labeling and Distribution – After extracting both statistical and pattern-based features, we pro-

ceed to label the preprocessed files accordingly. The statistical features provided numeric values as discussed previously. On the other hand, pattern-based features were binary (0/1), indicating the presence or absence of specific behaviors in the log entries. If a file exhibited characteristics corresponding to a particular pattern, the entry for that pattern’s column in the csv file was marked as 1. In total, we utilize 19 features to label the data, enabling a comprehensive classification of the logs.

The distribution of labels revealed significant insights into the nature of the data. The dataset D1, used for initial training and testing, comprised 3,232,188 log entries with 17,599 unique IP addresses, which were categorized as follows after feature extraction and labeling: Benign - 5,749, Brute Force (BF) - 3,221, Scanning - 2,310, Dictionary Attack (DA) - 2,189, and Denial of Service (DoS) - 1,137, Mixed Activity (MA) - 1,290, after SSH Compromise (ASC)-1703. The dataset D2, used for final testing, comprised 113,696 log entries with 7,029 unique IP addresses, which were categorized as fol-

Table 3: Test Results in % on D2 for various type of attack scenarios.

Test Results in % on D2: Brute Force Attack							Test Results in % on D2: Dictionary Attack						
Classifiers	RF	DT	SVM	KNN	LR	GB	Classifiers	RF	DT	SVM	KNN	LR	GB
Accuracy	99.47	99.26	98.67	99.39	99.29	99.43	Accuracy	99.46	99.35	98.67	99.40	99.37	99.32
Precision	99.01	98.82	98.02	98.92	98.85	98.97	Precision	99.00	98.90	98.02	98.98	98.97	98.97
F1-score	99.47	99.26	98.68	99.39	99.28	99.43	F1-score	99.46	99.35	98.68	99.40	99.38	99.33
TPR	99.93	99.71	99.35	99.87	99.73	99.89	TPR	99.93	99.81	99.35	99.83	99.79	99.70
FPR	1.00	1.19	2.01	1.09	1.16	1.04	FPR	1.01	1.11	2.01	1.03	1.04	1.04
TNR	99.00	98.81	97.99	98.91	98.84	98.96	TNR	98.99	98.89	97.99	98.97	98.94	98.94
FNR	0.07	0.29	0.65	0.13	0.27	0.11	FNR	0.07	0.19	0.65	0.17	0.21	0.30
Test Results in % on D2: Scanning Attack							Test Results in % on D2: DoS Attack						
Classifiers	RF	DT	SVM	KNN	LR	GB	Classifiers	RF	DT	SVM	KNN	LR	GB
Accuracy	99.95	99.93	99.82	99.91	99.90	99.92	Accuracy	98.94	98.83	98.42	98.84	98.51	99.09
Precision	99.97	99.95	99.89	99.91	99.93	99.95	Precision	98.78	98.62	98.01	98.68	98.02	98.32
F1-score	99.95	99.93	99.82	99.91	99.90	99.92	F1-score	98.94	98.83	98.43	98.84	98.51	99.09
TPR	99.92	99.90	99.75	99.91	99.87	99.89	TPR	99.11	99.04	98.85	99.00	99.01	99.89
FPR	0.03	0.05	0.11	0.09	0.07	0.05	FPR	1.23	1.39	2.01	1.33	2.00	1.71
TNR	99.97	99.95	99.89	99.91	99.93	99.95	TNR	98.77	98.61	97.99	98.67	98.00	98.29
FNR	0.08	0.10	0.25	0.09	0.13	0.11	FNR	0.89	0.96	1.15	1.00	0.99	1.11
Test Results in % on D2: Mixed Activity							Test Results in % on D2: After SSH compromise						
Classifiers	RF	DT	SVM	KNN	LR	GB	Classifiers	RF	DT	SVM	KNN	LR	GB
Accuracy	98.43	98.31	97.33	98.37	98.40	98.43	Accuracy	97.91	97.40	96.72	97.41	97.47	97.41
Precision	98.01	97.89	96.96	97.93	97.97	98.01	Precision	97.85	97.36	96.65	97.38	97.40	97.39
F1-score	98.44	98.31	97.34	98.38	98.41	98.43	F1-score	97.91	97.39	96.71	97.41	97.46	97.41
TPR	98.87	98.78	97.72	98.83	98.85	98.86	TPR	97.96	97.43	96.78	97.45	97.53	97.44
FPR	2.01	2.13	3.07	2.09	2.05	2.01	FPR	2.15	2.64	3.34	2.62	2.59	2.61
TNR	97.99	97.83	96.93	97.91	97.95	97.99	TNR	97.85	97.36	96.66	97.38	97.41	97.39
FNR	1.13	1.22	2.28	1.17	1.15	1.14	FNR	2.04	2.57	3.22	2.55	2.47	2.56

Table 4: Comparison with existing approaches.

Authors	Detected Attack	Used dataset	Approach	Accuracy
Jeo Park et al. (Park et al., 2021)	BF	Router log	Rule-based	Not reported
Fahrnberger et al. (Fahrnberger, 2022)	BF	SSH log	Rule-based	Not reported
Stephen Wanjau et al. (Wanjau et al., 2021)	BF	CIC-IDS 2018	ML-based	85.2%
Abel Z. Agghey et al. (Agghey et al., 2021)	UEA	Network Traffic	ML-base	KNN- 99.93%
Jose Tomas et al. (Garre et al., 2021)	Botnet infection	Novel dataset	ML-based	98.1%
Karel Hynek et al. (Hynek et al., 2020)	BF	Network Traffic	ML-based	100%
Our Approach	BSC and ASC	SSH log	Rule and ML based	Rule-based-99.92% RF = 97.9%

lows after feature extraction and labeling: Benign - 2394, Brute Force (BF) - 1179, Scanning - 784, Dictionary Attack (DA) -853, and Denial of Service (DoS) - 368, Mixed Activity (MA) - 548, After SSH Compromise (ASC)-903.

Feature Selection and Classification – In our proposed methodology, feature selection and classification play pivotal roles in accurately identifying and categorizing various attack types. We utilize a Random Forest (RF) classifier for feature selection due to its robustness and ability to handle large datasets effectively. The RF classifier helps in identifying the most significant features that contribute to accurate classification. Once the essential features are selected, we employ multiple machine learning classifiers for the classification task. These classifiers included Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), k-Nearest Neighbors (KNN), Logistic Regression (LR), and Gradient Boosting (GB). Each of these classifiers are trained and tested on the datasets D1 and D2 to evaluate their

performance. The results of these evaluations, detailing the effectiveness of each classifier, are presented in section 4.

4 EXPERIMENTAL RESULTS

In this section, we discuss the evaluation of both rule-based and ML-based approaches, which are as follows –

Evaluation of Rule-Based Approach – For this evaluation, we utilize dataset D1. Initially, feature extraction is performed using a prediction-based engine, which segregates the data into benign and various types of malicious activities (Brute force, mixed activity, scanning, dictionary attack, and denial of service), totaling 10,850 malicious IPs. Concurrently, the rule-based engine processed the same dataset and listed 10,842 IPs as malicious. These outputs are then compared, revealing a detection accuracy of 99.92%

for the rule-based engine.

Evaluation of ML-Based Approach – After feature extraction and selection, we evaluate the multi-label classification accuracy of each classifier using 10-fold cross-validation. The k-fold cross-validation results, demonstrating the multi-class classification accuracy for each classifier, are presented in Table 1. These detailed evaluations provide insights into each classifier’s ability to detect and differentiate between various types of cyber threats effectively. The corresponding results are systematically detailed in Table 2 which present a breakdown of these metrics for each attack type. After the initial training-testing using D1 dataset, we have saved our trained classifiers for subsequent testing with an independent dataset, D2, which was not included in the training phase. We evaluated the performance of all classifiers across each label in the dataset. The results are detailed in Table 3. In Table 4, we present a comparison of our approach with existing methodologies in terms of the datasets used, detected attacks, approaches applied, and accuracy. This comparison demonstrates how our proposed methodology overcomes the limitations found on existing approaches.

5 CONCLUSION

In this work, we propose a hybrid methodology that combines rule-based and machine learning (ML) approaches to detect various activities within SSH logs and profile attackers based on our model’s findings. Our rule-based approach utilized predefined and time-dependent rules to quickly identify suspicious activities, providing immediate heuristic-based insights. Complementing this, the ML-based approach extracted statistical and pattern-based features from the logs, enabling a detailed analysis of activities with respect to unique IP addresses. In terms of classification, while all classifiers demonstrated strong performance, the Random Forest (RF) and Gradient Boosting (GB) classifier consistently outperformed others, particularly in classifying unknown data. By integrating rule-based and ML-based approaches, we achieved a robust and accurate attacker profiling system. This comprehensive methodology significantly enhances the security and resilience of SSH servers against a wide range of attack vectors. The dataset used in the paper is available on request.

REFERENCES

- Agghey, A. Z., Mwinuka, L. J., Pandhare, S. M., Dida, M. A., and Ndibwile, J. D. (2021). Detection of username enumeration attack on ssh protocol: Machine learning approach. *Symmetry*, 13(11):2192.
- Barrett, D. J. (2005). Ssh: The secure shell - the definitive guide.
- Cartwright, E. and Bunter, M. (2019). Cyber fraud in the uk: Causes, consequences and the role of the consumer. *Crime Science*, 8(1):3. Accessed: 2024-05-15.
- Cimpanu, C. (2024). Ssh servers hit in 'proxyjacking' cyberattacks.
- Cloudflare (2024a). What is a denial of service (dos) attack? <https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/>. Accessed: 2024-05-15.
- Cloudflare (2024b). What is a phishing attack? <https://www.cloudflare.com/learning/access-management/phishing-attack/>. Accessed: 2024-05-15.
- CyberProof (2024). Which Countries Are Most Dangerous. <https://blog.cyberproof.com/blog/which-countries-are-most-dangerous>. Accessed: 2024-05-14.
- Fahrnberger, G. (2022). Realtime risk monitoring of ssh brute force attacks. In -, pages 75–95.
- Garre, J. T. M., Pérez, M. G., and Ruiz-Martínez, A. (2021). A novel machine learning-based approach for the detection of ssh botnet infection. *Future Generation Computer Systems*, 115:387–396.
- Hynek, K., Beneš, T., Čejka, T., and Kubátová, H. (2020). Refined detection of ssh brute-force attackers using machine learning. In *ICT Systems Security and Privacy Protection: 35th IFIP TC 11 International Conference, SEC 2020, Maribor, Slovenia, September 21–23, 2020, Proceedings 35*, pages 49–63. Springer.
- Kaspersky (2024). What is a dictionary attack? <https://www.kaspersky.com/resource-center/definitions/what-is-a-dictionary-attack>. Accessed: 2024-05-15.
- Keytos (2024). Biggest ssh vulnerabilities to consider in 2024: Learning from previous data breaches.
- MaxMind, Inc. (2024a). GeoIP2 and GeoLite2 City and Country Database Documentation. <https://dev.maxmind.com/geoip/docs/databases/city-and-country>. Accessed: 2024-05-14.
- MaxMind, Inc. (2024b). High-Risk IP Sample List. <https://www.maxmind.com/en/high-risk-ip-sample-list>. Accessed: 2024-05-14.
- Park, J., Kim, J., Gupta, B. B., and Park, N. (2021). Network log-based ssh brute-force attack detection model. *Computers, Materials & Continua*, 68(1).
- Roy, A. (2020). Ssh credentials targeted in data breach of hosting giant godaddy.
- Securelist (2023). Iot threat report 2023.
- SSL, R. S. (2024). What are user enumeration attacks? <https://really-simple-ssl.com/definition/what-are-user-enumeration-attacks/>. Accessed: 2024-05-15.

- Stampar, M. (2024). ipsum: Daily feed of bad IPs (with blacklist hit scores). <https://github.com/stamparm/ipsum>. Accessed: 2024-05-14.
- Team, A. S. I. R. (Year). Proxyjacking: A new campaign for the cybercriminal side hustle.
- Technica, A. (2024). Millions still haven't patched terrapin ssh protocol vulnerability.
- TechTarget (2024). What is a man-in-the-middle (mitm) attack? <https://www.techtarget.com/iotagenda/definition/man-in-the-middle-attack-MitM>. Accessed: 2024-05-15.
- Wanjau, S. K., Wambugu, G. M., and Kamau, G. N. (2021). Ssh-brute force attack detection model based on deep learning. -.

