Taylor & Francis
Taylor & Francis Group

Information Security Journal: A Global Perspective

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uiss20

SSHAA: A Python Package Index for visualizing features of SSH attacks with text mining in classification

Yuki Nakamura, Tsukasa Fukuda, Xuanzhou Yang & Yoshiyasu Takefuji

INFORMATION SECURITY JOURNAL: A GLOBAL PERSPECTIVE
https://doi.org/10.1080/19393555.2024.2357317

SSHAA: A Python Package Index for visualizing features of SSH attacks with text mining in classification

Yuki Nakamura, Tsukasa Fukuda, Xuanzhou Yang, and Yoshiyasu Takefuji

Faculty of Data Science, Musashino University, Tokyo, Japan

ABSTRACT

With the proliferation of the Internet, the number of cyber-attacks has increased worldwide. To prevent cyber-attacks, network administrators need to use analytical tools to disclose six new critical statistics such as Internet Protocol (IP) authenticity and classification, IP ownership, frequency of attacks, time of day, day of the week, and country of attack. There are four SSH attack analysis tools available, but existing tools lack visualization and cannot provide the necessary attack statistics. This paper proposes SSHAA, the first open source PyPI analysis tool that meets the requirement. Text mining technology is used for extracting data from SSH log files, classifying IPs with identified country names and calculating a variety of statistics. PyPI allows SSHAA to run on Windows, MacOS, and Linux operating systems. This paper highlights how to debut a PyPI application using SSHAA which will be the world's first tutorial in the security journals for maximum software dissemination. According to PePy, SSHAA has been downloaded 7668 times worldwide. The substantial number of downloads signifies the efficacy of the proposed tool. The proposed system offers ten functional statistics for SSH attack analysis. The six new functional statistics to information security are significant.

KEYWORDS
Cyber-attacks; DDoS attacks; genuine and fake IP; SSH

1. Introduction

Cyberattacks have been raging around the world. According to the Norton follow-up survey in Japan (Norton, 2021), more than 18 million people have experienced cybercrimes in the last 12 months since 2020, and the total amount of damage is estimated to be 194 billion US dollars. Furthermore, not only individual companies, but also companies that have been hit by cyber-attacks often suffer fatal damages such as loss of credibility and compensation issues. To illustrate cyber-attacks on businesses and organizations in depth, several case studies will be described in this paper.

According to bbc.com (BBC, 2021), the website of the New Zealand Stock Exchange was the target of a threatening DDoS attack in 2020. The company was taken offline for several days. Additionally, other threatening e-mails were sent to PayPal, Braintree, and several other financial institutions demanding amounts in Bitcoin. In 2016 at Osaka University, a month-and-a-half-long cyber-attack occurred on a database containing information on approximately 80,000 people. After the attack, the database was illegally accessed via an overseas IP address, and it was confirmed that the information of 69,549 people had been downloaded (CyberSecurity, 2021). This paper will show how people are very unaware and uncaring about cybercrimes. Other examples of recent cyber-attack incidents are, for example, blockchain startup MonoX Finance had announced that hackers have stolen 31 million USD by exploiting a bug in the software. In smart contracts, many developers do not do the work of defining the security properties of the code, and this is why such attacks are often carried out (Canon, 2020). For another example, there was a "SIM swap attack" in 2021 In the United States and $46 million cryptocurrency was stolen. The employee of a mobile phone company was tricked into duplicating his phone number on a hacker's SIM card to hijack his account by blocking his request for two-factor authentication. The suspect is said to be a young man living in Hamilton (Stephanie, 2021).

This paper introduces a novel tool for SSH analysis, capable of examining six new crucial statistics such as classifications of attack attempts by time of occurrence, day of the week, username,

CONTACT Yoshiyasu Takefuji takefuji@keio.jp Faculty of Data Science, Musashino University, 3-3-3 Ariake Koto-ku, Tokyo 135-8181, Japan

port number, and identification of the top 10 countries associated with IPs involved in SSH attacks. System administrators bear the responsibility of safeguarding their machines from harmful attacks. Secure Shell (SSH), a network protocol, facilitates a secure connection between two entities, typically used for remote server control and file transfer. It employs cryptography for device connection authentication and encryption. The importance of SSH protection stems from the fact that despite the protocol's inherent security, its authentication mechanism and client-server configuration are susceptible to misuse. Poor SSH key management can lead to considerable digital security threats for organizations. For example, unsecured SSH keys can result in key sprawl, lost keys, policy enforcement failure, and data breaches. Furthermore, SSH is often individually managed by administrators for the servers under their control. This absence of centralized supervision can lead to significant security risks. If an attacker gains access to one server, it could potentially open up access to others. Hence, administrators must shield SSH from malicious attacks to uphold a secure and efficient machine identity management program.

Computing systems are typically classified into two main categories: stationary systems, which encompass desktop computers and servers, and mobile devices that include laptops, smartphones, and vehicular computers. This paper primarily concentrates on the cybersecurity of stationary systems. However, it commenced with a literature review on the security of vehicular devices, followed by a comprehensive review of cybersecurity in stationary systems.

Al-Shareeda et al. proposed a COVID-19 vehicle using an efficient mutual authentication scheme for 5 G-enabled vehicular fog computing to combat the pandemic's spread via transportation systems (Al-Shareeda & Manickam, 2022a). Their scheme, which includes flags for normal and COVID-19 vehicles, enhanced privacy, security, and healthcare solutions, and outperformed recent works in communication and computation costs (Al-Shareeda & Manickam, 2022a). However, they did not specify the entities attacking the vehicles or the origins of these attacks. This paper provides a comprehensive analysis of server log files.

Al-Shareeda et al. proposed a Secure and Efficient Conditional Privacy-Preserving Authentication (SE-CPPA) scheme for Vehicular Ad-Hoc Networks (VANETs) (Al-Shareeda et al., 2021a). Their scheme, based on cryptographic hash function and bilinear pair cryptography, resisted impersonation attacks and improved performance efficiency. It outperformed existing schemes in computation and communication costs, reducing them by 99.95%, 35.93%, and 27.3% respectively. However, their security methods are passive and cannot identify and locate attackers. This paper can provide information to identify attackers as much as possible and to prepare for future attackers.

Al-Shareeda et al. proposed enhancements to an identity-based conditional privacy-preserving authentication scheme for Vehicular Ad Hoc Networks (VANETs) (Al-Shareeda et al., 2021b). Their scheme addressed security vulnerabilities, improved communication efficiency, and has been proven secure under the random oracle model. Performance evaluations showed it outperforms existing schemes in signing and verifying VANETs messages. However, their methods cannot identify and locate attackers. This paper can provide information to identify attackers as much as possible and to prepare for future attackers.

Al-Shareeda et al. proposed a fog computing-based pseudonym authentication (FC-PA) scheme for 5 G-enabled vehicular networks (Al-Shareeda et al., 2023). The FC-PA scheme, which uses a single scalar multiplication operation of elliptic curve cryptography, addresses privacy and security issues in vehicular communications. It offers improved efficiency and security trade-offs compared to existing schemes. However, their approach is defensive and cannot identify and locate attackers while this paper can provide information of attackers and to prepare for future attackers.

Al-Shareeda et al. proposed a 5 G-enabled vehicular network scheme, MSR-DoS, to improve traffic safety and efficiency (Al-Shareeda & Manickam, 2022b). It addressed privacy, security, and DoS attacks, which are challenges in these networks. The scheme used modular square root-based operations, reducing computational costs and overhead by up to 99.80%. It ensures message integrity, source authenticity, and privacy while being traceable and revocable. The security is proved under BAN logic. While their standalone protection scheme doesn't prevent future attacks, this paper offers detailed insights into potential attacks and prepares strategies to counter them.

If the characteristics of attackers were incorporated into vehicular network security, this valuable information could be utilized to enhance the proposed tool in the future.

A comprehensive review of cybersecurity in stationary systems was conducted. Kotenko et al. presented a two-level attacker model for risk analysis, constructed from high-level attributes derived from low-level attributes based on raw security data (Kotenko et al., 2023). Experiments with attack datasets showed that attacker profiles can be differentiated using nominal parameters like bash history logs. However, accurate profiling requires expanding the list of low-level attributes. This paper enhances their approach by proposing a method that can extract low-level attributes from SSH log files.

Shandilya et al. presented a comprehensive dataset of real-time network data during various cyber-attacks (Shandilya et al., 2022). The data, collected using Wireshark and tcpdump, includes attack scenarios on older software versions and unprotected ports. The dataset emphasized the importance of system updates and patches and includes various attack types such as Distributed Denial of Service, SQL Injection, and more. It also demonstrated the impact of insider threats on network security. However, their approach lacks visualization for potential analysis. Their collected data can be used for enhancing the proposed method in the future.

Wawrowski, et al. focused on enhancing network traffic safety through continuous monitoring and anomaly detection (Wawrowski et al., 2023). Their proposed solution, an anomaly detection module, offered a novel strategy for selecting and tuning model combinations in a faster offline mode. Their approach, aimed at public institutions, has achieved 100% balanced accuracy in detecting specific attacks. Their solution can be used to enhance the proposed method in the future update.

Alani et al. introduced a machine-learning-based system for detecting reconnaissance attacks on IoT devices (Alani & Damiani, 2023). Their system, based on an explainable ensemble model, aimed to counter attacks at an early stage. It was designed to be efficient and lightweight, suitable for resource-constrained environments. The system achieved 99% accuracy with low false positive and negative rates of 0.6% and 0.05%, respectively. Combining their approach and the proposed method can enhance identifying attackers and mitigate future attacks.

Noman et al. analyzed code injection attacks in IoT, particularly in the wireless domain (Noman & Abu-Sharkh, 2023). It revealed how these attacks exploit vulnerabilities in IoT systems and wireless technologies, leading to severe consequences like data breaches. Their study included a framework illustrating these attacks, proof of concept on Wi-Fi devices, and detection of malicious codes in firmware. It emphasized the need for understanding these vulnerabilities to develop more secure IoT systems. Their solution can be embedded in the proposed method in the future update.

From the past to now, there have been inadequate countermeasures against cyber-attacks, which continues to be a major problem and should be solved immediately. In order to know about security, it is essential to know about the problems occurring on the Internet. The internet is based on open-source software such as Linux operating systems. In other words, Linux is an indispensable part of the Internet. In 2020, 100% of the world's top 500 supercomputers run on Linux (Abhishek, 2020), 96.3% of the world's top 1 million servers run on Linux (Steven, 2015), and 90% of all cloud infrastructures operate on Linux (Nick, 2021). According to *Nik* (2021), there are more than one billion websites in the world.

Another representative open-source software is a web server. Wappalyzer.com (Wappalyzer, 2022) implies that 41% of the world's web servers run on Apache, 39% on Nginx, 7% on IIS, and 5% on LiteSpeed. Apache and Nginx are based on open-source software, so without Linux and open-source software, the Internet would not exist.

The protocol for providing secure remote access is called SSH, and it is the most widely and frequently used technology for accessing servers in remote locations. According to Tatu, Y. in 2017 (Tatu, 2017), It is now used in most data centers around the world, and more than half of the world's web servers are managed using SSH. To highlight its essential functionality, command lines, such as SSH are essential for researchers (Perkel, 2021). In 2021, Q-Success conveyed a study that showed that about 91% of the top 1,000 most popular websites use HTTPS as their default protocol (Techplus, 2021).

There are two types of IPs: true IPs and fake IPs. Fake IPs include forged IPs, springboard IP, and spoofed IPs. Fake IPs are difficult to identify, making it difficult to deal with DDOS (Distributed Denial of Service) attacks (Hayato, 2021). Using these mixed IPs makes it difficult to deal with DDoS attacks. In the next section, we will show that many of the incidents are caused by DDoS attacks, and the amount of damage is increasing.

According to Shizuka (Shizuka, 2021), although the total attack number has decreased, the amount of damage is rising due to the viciousness of the attacks. Furthermore, Shizuka implies that the global honeypots (decoy servers used to tempt attackers) that F-Secure has set up to gather information have seen about 2.8 billion attack events in the second half year, which when combined with 2.9 billion attacks in the first half of 2019, total amount goes up to 5.7 billion attacks for the full year. This research report is published semiannually, and previous findings have reported 1 billion attacks in 2018 and 800 million attacks in 2017.The majority of the surge in attack traffic in 2019 came from DDoS attacks, accounting for two-thirds of the total.

A DDoS attack is a cyber-attack conducted from multiple computers at once, thus transmitting even more data than a DoS attack and placing a greater load on the attack target. Currently it is hard to distinguish true IPs and spoofed IPs. Therefore, it is hard to mitigate a DDoS attack (Hayato, 2021).

To counter this, there are some famous countermeasure tools against SSH attacks today. Such as the MFA tools (Google, 2022), ingress filtering, egress filtering, DenyHosts (denyhosts, 2022), Fail2ban (fail2ban, 2022). These tools prevent unauthorized access by complicating login procedures, disabling a large number of login attempts from the same access source, checking for spoofing of the internal network. Analysis can also be done by using log analysis tools such as Fluentd, LOGalyze, Elastic Stack. There is no disputing that these tools are of course effective and actually prevent a variety of malicious attacks, it could be said that these existing tools lack the ability to fully analyze the source of access and are likely to end up being symptomatic measures.

In today's world where IP spoofing technology is widely recognized, these tools are not able to detect unauthorized access or are likely to breach security if the IP is spoofed to a trusted destination or a trusted IP is spoofed or an IP is used in a springboard attack.

In this paper, we introduce SSHAA, a new open-source tool for analyzing and visualizing SSH attacks in order to compensate for the weaknesses of the existing tools. Conventional tools have a low degree of freedom in analysis and can only obtain information to the extent of analysis by IP address.

SSH log is analyzed with the SSHAA tool. Weekly SSH log files, including auth.log and other archived files, can be found in the/var/log/ directory on Ubuntu and Debian Linux systems. The file size varies based on events, such as external attacks. For instance, the log file for the week of October 29, 2023, is 29MB (29,682,844 bytes). The auth.log file is a crucial system log that documents all attempts at authentication. It meticulously records each remote login attempt to the server, including the account used, and the date and time of the attempt. Additionally, it logs every instance where a user password is prompted, like when using the sudo command, and notes if the authentication was successful or not. This log file plays a vital role in maintaining server security by aiding in the identification of unauthorized access or any activities that raise suspicion.

If we can get various information about the access source (e.g., country name, name of the organization, contact information, etc.) other than the IP address, it can be used for effective analysis of SSH to mitigate future attacks.

The SSHAA is available to the public and can be installed by PyPI packaging. This tool also has a separate aggregation function for specific IPs and subnet masks, allowing analysis of access from specific IPs. By using this program, it is possible to graph the number of attempts for each source IP for malicious access attempts, and analyze the attack trend for information such as the attack source IP, access destination port, and username. In addition, WHOIS processing can be incorporated into the access source IP, and information about the country affiliation and the owner of the IP that has been made public can be summarized as a report. A report containing the graphed data can be output as a CSV file.

As an additional study using this program, we analyzed the trend of unauthorized access attempts by day of the week. As a result, a significant trend was observed, such as a decrease in the number of accesses on Saturday, but a clear increase on Sunday. By analyzing the information, it will be possible to analyze the attacker's method on SSH-attack.

Thus, it can be said that this tool could be used for analyzing the pattern of SSH attacks. The analysis of the attacker's intentions will provide information for countermeasure policies when spoofing or springboard attacks are assumed. However, since this information includes all kinds of information about the SSH daemon, it is not used in most cases.

An effective method against IP spoofing and springboard attacks is to detect spoofed information, but comprehensive countermeasures are difficult with the current Internet technology (Hayato, 2021). By using an SSH log analysis tool, it is possible to share information about the misuse of the system at an early stage by sharing the unauthorized access attempts made on individual PCs.

Conventional tools only provide information about the attacker's IP address and the number of attacks. In this paper, we propose a new open-source tool that will be beneficial in defending against attackers by providing new features that conventional analysis tools cannot provide. New features include country name and institutional affiliation information associated with IP addresses, time of day and day of week on analysis of attack frequency, and detailed analysis for specific IP addresses. It also includes visualization of attack attempts with circular graphs, and attack frequency analysis for attacked usernames and port numbers.

The SSHAA can be installed by PyPI packaging with pip command "pip install SSHaa" and has been downloaded 7668 times worldwide. This fact shows that the applicability, practicality, and usefulness of the proposed tool was justified.

This paper will be the first tutorial in security journals on how to debut a PyPI packaging application. Log data files are used with the text mining technology for extracting IPs, date and time and identifying the country names from IPs and calculating a variety of statistic features such as the most frequent time of day, the number of attacks per day of the week, the identified country of attack, and the owner of the IP.

While there are numerous SSH analysis tools available such as MFA tools (Google, 2022), DenyHosts (denyhosts, 2022), Fail2ban (fail2ban, 2022), and SSH TRACKER (Solnichkin, 2019), they often lack a comprehensive analysis that includes categorizing attack attempts based on the time of occurrence, day of the week, username, port number, and identifying the top 10 countries associated with the IPs involved in SSH attacks. The tool proposed in this paper aims to fill this gap by providing a more thorough analysis. The detailed analysis of attacker characteristics plays a pivotal role in future attack mitigation. This is achieved by investigating the original owners of the attacking machines and subsequently informing them.

2. Materials and methods

In this section, we will analyze the SSH log data of five servers that are actually in operation using the functions included in SSHAA and verify its effectiveness.

2.1. Dataset

In Linux-like operating systems, information about all login history, including failed and successful attempts, is stored in a specific log file. For Debian OS and its derivative Ubuntu OS, it is "/var/log/auth.log," and for CentOS servers, it is "/var/log/secure." The Raspberry Pi OS (formerly Raspbian), the standard OS of the Raspberry Pi, which is currently widely used as an embedded PC for research and development, falls into the former category, for example:

```
Apr 25 00:00:09 neuro SSHd[4186563]: Failed password for root from
   113.88.13.132 port 40,206 SSH2
Apr 25 00:00:12 neuro SSHd[4186709]: Failed password for root from
   167.99.41.147 port 59,518 SSH2
Apr 25 00:00:12 neuro SSHd[4186793]: Invalid user gpu from
   109.134.162.71 port 38,884
```

The above extracts the part of the log data related to login attempts to SSH. In the log file, the date and time of the event, server name, process name, username, IP address of the access source, port number, and version of SSH are recorded for each login attempt. In this study, we will analyze using this information.

We will use the data recorded on the actual server as the data set. The target servers are from "Server-1" to"Server-4," which are mainly used for data analysis and web server purposes in Japan. The log data obtained from the servers covers the period from 2021/11/21 0:00:00 to 2021/11/25 19:11:00, and contains 196,577 lines 89,132 lines, 167,650 lines, and 166,007 lines of events, respectively. Of these, lines 21,907, 20229, 21864, and 22,807 were recorded as failed login attempts, which are the subject of this analysis. It is estimated that an average of 1.65 million login attempts occur per server per year, assuming that the same trend continues for a year. In addition, as "Server-5," we prepared a log file for the day-of-week analysis.

2.2. Existing approaches

Currently, MFA tools (Google, 2022), DenyHosts (denyhosts, 2022), and Fail2ban (fail2ban, 2022) are the most widely known existing tools for analyzing SSH logs. Although fail2ban can check the records of access blocking, the information that can be checked is limited such as the number of failed login attempts and the number of denied accesses from IP addresses registered in the denial list. There is not enough information to analyze SSH attacks. An analysis tool that specializes in visualizing access attempts is SSH TRACKER (Solnichkin, 2019). This program maps the failed login attempts on a map. The larger the number of login attempts, the larger the circle is displayed, and it is a tool that can visually show the source and scale of an attack in an easy-to-understand manner. However, the tool lacks sufficient data on trends like "day of the week" and "time of day." Additionally, registration is required to utilize this tool.

2.3. Research approach

In the previous section, we introduced the existing tools for blocking unauthorized access and their visualization approaches. The blocking of unauthorized access by existing tools is realized by the IP table of the unauthorized access source.

However, the number of attacks using spoofing IPs is increasing today. Spoofing IPs are used for distributed denial-of-service (DDoS) attacks and to slip through authentication (Veracode, 2022). As mentioned above, these existing tools use IP tables to block unauthorized access. This means that legitimate access from the outside may be blocked depending on the forged IP.

There are also cases in which a trusted server with a malicious process embedded by some means is used as a springboard for an attacker to attempt unauthorized access. Forged IPs can be countered by these existing tools by setting the registration period of the denied IP table.

However, the fundamental problem is that it is difficult to completely block the attacks. This is due to the fact that IP addresses can be forged and there is a limit to the information that can be obtained from packets. As a next best solution, Ingress filtering in the internal network is often taken as a basic security measure, but it is difficult to solve at the software level with the current SSH protocol.

In order to solve these problems, it is necessary not only to detect unauthorized access by simple IP address, but also to analyze the information obtained from log files to the maximum extent possible and incorporate it into security measures, to detect unauthorized access from a trusted network as soon as possible, and to eliminate the cause. In this paper, we discuss these situations. This paper adds new SSH log analysis functions such as trends of frequency, day of the week, popular forged names and others.

2.4. SSH log analysis

The target of this analysis is the date and time of event occurrence, username, IP address of the access source, and port number among the login

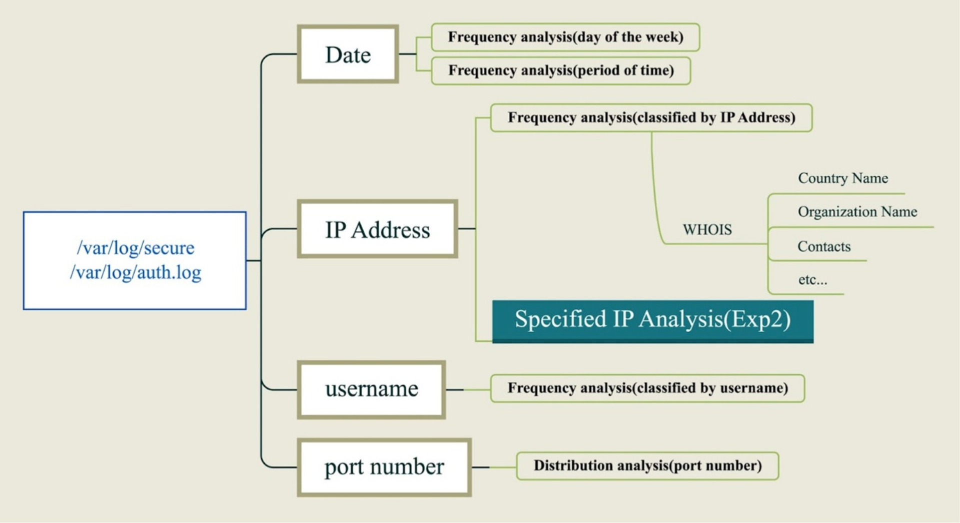
/var/log/secure
/var/log/auth.log
Date
Frequency analysis(day of the week)
Frequency analysis(period of time)
IP Address
Frequency analysis(classified by IP Address)
WHOIS
Country Name
Organization Name
Contacts
etc...
Specified IP Analysis(Exp2)
username
Frequency analysis(classified by username)
port number
Distribution analysis(port number)

Figure 1. Map of data that can be retrieved from auth.Log and secure files.

attempt records in the log data file as shown in Figure 1. First, as Experiment 1, we will conduct an access frequency analysis of IP addresses, login usernames, and access destination port numbers as a basic analysis method. In particular, the frequency analysis of the access source IP address is one of the basic countermeasure approaches against attacks and has been implemented in existing tools.

WHOIS is a protocol that can be used to search for the owner of an IP address. By querying an IP address, it is possible to obtain information about the organization to which the IP address belongs. The information obtained includes the name of the country, the name of the organization, and contact information. WHOIS APIs typically exhibit a sluggish response time and are capable of processing only a restricted number, approximately in the tens, of queries every minute.

In the proposed tool, the WHOIS API is composed of multiple APIs and they are rotated to eliminate limited queries. The slow processing and limitation of queries makes these APIs unsuitable for integration into IP address analysis, but querying IPs can be made more efficient by consolidating access histories from the same IP, or by allowing local access to WHOIS results for IPs that have been queried once as a preprocessing step.

While this information is highly beneficial for security protocols, its integration into log analysis, which handles substantial data volumes, is impractical. The current WHOIS service can only handle a finite number of queries per second, yet each instance requires processing a significant number of queries. Therefore, we adopted a system that stores the queried information in a local database for a certain period of time and refers to the local data for the second and subsequent queries, in order to reduce the burden on the query API and speed up the analysis.

The analysis of access frequency to usernames and port numbers is expected to provide information on brute force attacks, dictionary attacks, and whether the attacker already has access to the information of the server even in fragments. As analysis related to the date and time of event occurrence, we also conducted analysis by day of the week and by time of day.

Next, as Experiment 2, we simulated a detailed analysis to take countermeasures against specific malicious IP addresses based on the analysis of Experiment 1.

This is the analysis map of this experiment.

3. Results

First, as a basic analysis method, we performed an analysis of IP addresses, login usernames, and access destination port numbers. As shown in 3.1, the total number of illegal accesses to “Server 1” was 21,907, and the following table shows that 30% of them were in the top 4. The table below shows that 30% of the total number of malicious accesses

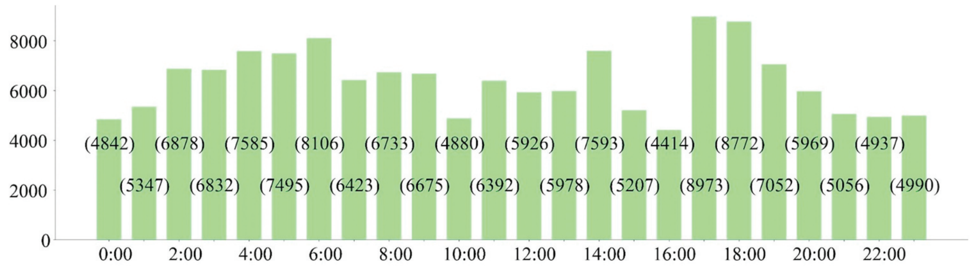

Figure 2. Attack attempts classified by time of occurrences.

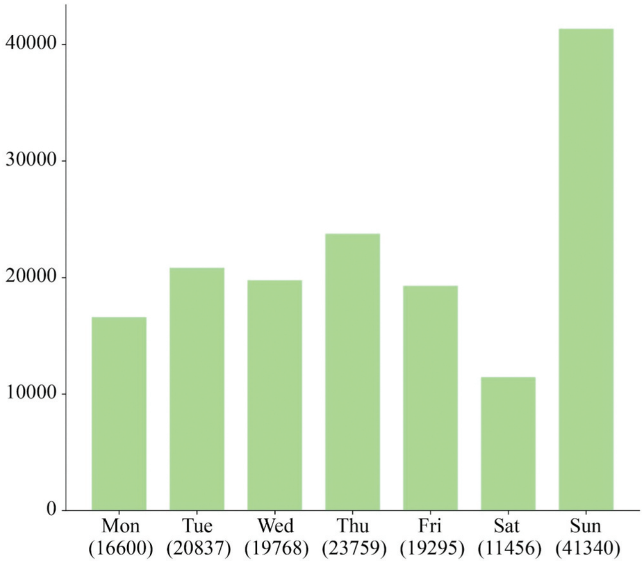

Figure 3. Attack attempts classified by the day of the week.

to server 1 is accounted for by the top four. In addition, by performing a WHOIS lookup on these IP addresses, we can obtain information on the organization that owns the IP address and the country it belongs to.

Table 1 indicates IP address, country name and the number of failed login attempts. While the auth.log file in Linux is typically a reliable source for pinpointing the origin of a connection attempt, its accuracy can be compromised by advanced IP hiding techniques such as VPNs, proxy servers, the Tor network, and IP spoofing. These methods can disguise a user's true IP address, giving the illusion that the connection originates from a different location. As such, even though auth.log is instrumental in monitoring authentication attempts, it's crucial to remember that savvy users might use these techniques to hide their real IP addresses. In these instances, extra security precautions and analyses may be necessary to accurately determine the source of suspicious activities. Interestingly, even with the use of IP hiding techniques, it's sometimes possible to detect and track behavioral patterns.

Table 2 below shows the aggregate results for the login usernames. The table shows the top six positions in the number of login attempts. It can be read that more than half of the login attempts are attacks against root.

Next, we show the aggregate results of the access destination port numbers. Table 3 shows the top six positions in the number of login attempts. Compared to the results of the previous aggregation, the access destination port numbers varied widely, and no significant trend was observed.

The date and time of the access attempt are also recorded in these log files. The following graph shows the statistical results of the timing of the login attempts.

The following graphs as shown in Figure 2 show the results of the analysis on the time of day when the accesses were received, showing that the number of unauthorized accesses peaked at 0:00, 7:00, 11:00, and 17:00. It was found that there was a significant trend in the access time period.

Next, we classify the number of accesses based on the day of the week. There is no significant change in the number of attempts from Sunday to Friday, but it shows that the number of attempts on Saturday is about half that of the other days.

And then, we classify the number of accesses based on the day of the week using a "Server 5" log file as shown in Figure 3. There is no significant change in the number of attempts from Sunday to

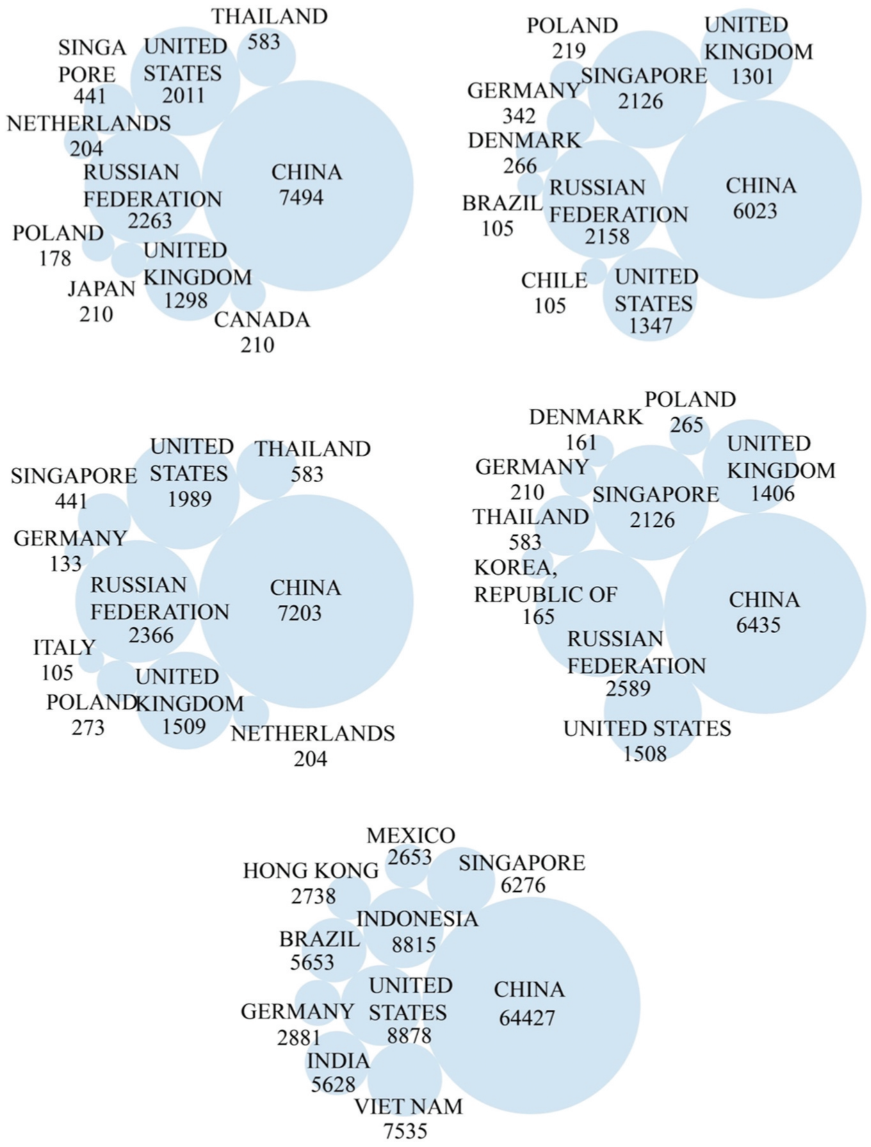

Figure 4. Top 10 countries of IPs with SSH attacks.

Table 1. IP address and country name with login attempts.

IP address	Country	Login attempts
..***.***	CHINA	2166
..***.***	RUSSIAN FEDERATION	2158
..***.***	UNITED KINGDOM	1298
..***.***	UNITED STATES	1129
..***.***	CHINA	471

Table 2. Username and attempts.

Username	Attempts	Username	Attempts	Username	Attempts
root	12124	user	437	test	259
admin	1173	hadoop	364	ubuntu	218

Table 3. Port number and attempts.

Port No.	Attempts	Port No.	Attempts	Port No.	Attempts
54340	11	20561	9	34068	9
56534	11	33365	9	41004	9

Friday, but it shows that the number of attempts on Saturday is about half that of the other days.

The results of the same analysis for “Servers 2” to”Server 4” are shown in Table 4, Table 5, Table 6, Table 7 and Figure 4.

By using the WHOIS protocol, it is possible to obtain the country of the owner from the access source IP. The above figure visualizes the number of attack-attempts per country of affiliation between “Server-1” and “Server-5.”

4. PyPI packaging

According to TIOBE in 2022, Python is the most popular programming language in the world.

Table 4. Attack attempts classified by country.

Country	Attempts	Country	Attempts
Server 1		**Server 2**	
China	7494	China	6023
Russian Federation	2263	Russian Federation	2158
United States	2011	Singapore	2126
United Kingdom	1298	United States	1347
Thailand	583	United Kingdom	1301
Server 3		**Server 4**	
China	7203	China	6435
Russian Federation	2366	Russian Federation	2589
United States	1989	Singapore	2126
United Kingdom	1509	United States	1508
Thailand	583	United Kingdom	1406

Table 5. Attack attempts classified by IP address.

Country	Attempts	Country	Attempts
Server 1		**Server 2**	
China	2166	Russian Federation	2158
Russian Federation	2158	China	2019
United Kingdom	1298	Singapore	1685
United States	1129	United Kingdom	1301
China	471	United States	1129
Server 3		**Server 4**	
China	2377	China	2166
Russian Federation	2156	Russian Federation	2158
United Kingdom	1299	Singapore	1685
United States	1129	United Kingdom	1301
China	852	United States	1129

Table 6. Attack attempts classified by username.

Username	Attempts	Username	Attempts
Server 1		**Server 2**	
root	12124	root	10150
admin	1173	admin	1101
user	437	user	794
hadpoop	364	hadoop	338
test	259	test	242
Server 3		**Server 4**	
root	12255	root	11708
admin	1219	user	979
username	406	admin	957
hadoop	318	hadoop	305
test	236	test	247

Table 7. Attack attempts classified by port number.

Port number	Attempts	Port number	Attempts
Server 1		**Server 2**	
54340	11	34767	9
56534	11	25400	8
20561	9	34762	8
33365	9	35090	8
34068	9	35594	8
Server 3		**Server 4**	
2072	70	56762	12
39106	10	38172	10
41604	9	56184	10
60428	9	57023	10
13452	8	40772	9

PyPI is the de facto Python Package. PyPI has 368,708 projects, 3,374,362 releases, 5,881,492 files, and 585,009 users in the world. In other words, PyPI allows programmers to maximize new software dissemination worldwide. However, there is no tutorial how to debut a PyPI packaging in security journals. The paper's contribution is significant. Remember that PyPI allows SSHaa to run on Windows, MacOS, and Linux operating systems without being aware of operating systems as long as Python is installed on the system.

PyPI packaging needs three files such as READEME.md, setup.py and application (SSHaa.py). In order to debut a PyPI application, GitHub account and PyPI account are needed. The following is the summarized procedure to debut your PyPI application.

(1) Create the README.md file with GitHub site creating a new repository with a README file option. You can download the completed README.md later.

(2) Download the compressed file and expand it in order to see the template of setup.py file from PyPI site:

https://files.pythonhosted.org/packages/99/d1/bacccc6d9ad590b76c6ef765d05b757249e3cb4a4f247045473cdc3f6cf3/SSHaa-2.1.1.tar.gz

```
$ tar xvf SSHaa-2.1.1.tar.gz
$ cd SSHaa-2.1.1
$ cat setup.py
  import setuptools
  with open(“README.md,” “r,” encoding=“utf-8”) as fh:
   long_description = fh.read()
 def _requires_from_file(filename):
   return open(filename).read().splitlines()
 setuptools.setup(
    name=“SSHaa,”
    version=“2.1.1,”
    author=“author_name,”
    author_email=“email_of_author,”
```

```
        description=“Analyze auth.log,”
        long_description=long_description,
        long_description_content_type=“text/markdown,”
        url=“https://github.com/name_of_author/SSHaa,”
        project_urls={
         “Analyze SSH auth.log:” “https://github.com/name_of_author/SSHaa,”
        },
        classifiers=[
          “Programming Language : Python : 3,”
          “License : OSI Approved : MIT License,”
          “Operating System : OS Independent,”
        ],
        package_dir={”“: “src”},
        py_modules=[“SSHaa”],
        packages=setuptools.find_packages(where=“src”),
        python_requires=“≥3.6,”
        install_requires=_requires_from_file(“requirements.txt”),
        entry_points={
         “console_scripts:” [
         “SSHaa = SSHaa:main”
        ]
       },
)
```

The shaded 10 lines in setup.py template file should be modified for your application.

3. In order to upload three files, you need to install twine.

   ```
   $ pip install twine
   ```

   You should create the (.pypirc) file in your home directory for authentication. The file is with three lines. The third line should be modified with your API-token which can be generated by clicking “Add 2FA with authentication application” button in account setting. The API token starts from “pypi-”:

   ```
   [pypi]
   username = __token__
   password = <API-token>
   ```

4. Finally, you can upload files and debut your PyPI packaging application with the following commands. The structure of directory (src) and files (README.md, setup.py, SSHaa.py) is as follows.

   ```
   $ tree.

   ├── README.md
   ├── setup.py
   └── src
   └── SSHaa.py
   $ python setup.py install
   $ python setup.py sdist bdist_wheel
   $ twine upload dist/*
   ```

If your PyPi packaging is successful, the URL address will be shown in your terminal.

5. Discussion

There has been a noticeable uptick in the frequency of SSH attacks in recent times. However, existing tools such as MFA tools (Google, 2022), DenyHosts (denyhosts, 2022), Fail2ban (fail2ban, 2022), and SSH TRACKER (Solnichkin, 2019) fall short in providing critical statistics such as IP authenticity and classification, IP ownership, attack frequency, time of day, day of the week, and country of attack. Conventional tools have limited analytical capabilities and can only analyze to the extent of the IP address. Remember that SSHAA, being a PyPI application, can be installed and used without registration simply by utilizing the pip command.

To address this, the proposed tool improves upon the weaknesses of existing tools and introduces new functions for better visualization of attacks. It charts the number of malicious access times for each source IP address and analyzes the attack trend of the source IP address, destination port, and attempted username. It also incorporates WHOIS processing into access to source IP and aggregates public country and IP owner information into a report. Furthermore, it can analyze trends such as which days of the week have the most attacks, aiding in determining the identity of the attacker.

Future work could involve integrating the proposed method with cutting-edge research techniques such as continuous monitoring and anomaly detection (Wawrowski et al., 2023), machine-learning-based detection of reconnaissance attacks (Alani & Damiani, 2023), and detection of code injection attacks in the wireless domain (Noman & Abu-Sharkh, 2023). Additionally, data from cyber-attacks, including Distributed Denial of Service and SQL Injection (Shandilya et al., 2022), could be leveraged to further enhance the proposed method.

6. Conclusion

The proposed tool, SSHAA, is highly effective due to new features such as such as classifications of attack attempts by time of occurrence, day of the week, username, port number, and identification of the top 10 countries associated with IPs involved in SSH attacks for analyzing SSH attack statistics. The information obtained to analyze log files is maximally integrated into security measures to detect unauthorized access from a trusted network as soon as possible and eliminate the cause. SSHAA can output reports containing graphed data as CSV files for sharing with others and can accumulate information among other users to block the attack source and prevent attacks. With 7668 downloads worldwide, the applicability, practicality, and usefulness of SSHAA are well-justified.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research has no fund.

ORCID

Yoshiyasu Takefuji http://orcid.org/0000-0002-1826-742X

References

Abhishek, P. (2020). Linux runs on all of the top 500 supercomputers, again! *https://itsfoss.com/linux-runs-top-supercomputers/*

Al-Shareeda, M. A. (2023). FC-PA: Fog computing-based pseudonym authentication scheme in 5g-enabled vehicular networks. *IEEE Access*, 11, 18571–18581. https://doi.org/10.1109/ACCESS.2023.3247222

Al-Shareeda, M. A., Anbar, M., Manickam, S., & Hasbullah, I. H. (2021a). SE-CPPA: A secure and efficient conditional privacy-preserving authentication scheme in vehicular ad-hoc networks. *Sensors (Basel, Switzerland)*, *21*(24), 8206. https://doi.org/10.3390/s21248206

Al-Shareeda, M. A., Anbar, M., Manickam, S., & Hasbullah, I. H. (2021b). Towards identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Access*, 9, 113226–113238. https://doi.org/10.1109/ACCESS.2021.3104148

Al-Shareeda, M. A., & Manickam, S. (2022a). COVID-19 vehicle based on an efficient mutual authentication scheme for 5G-Enabled vehicular fog computing. *International Journal of Environmental Research and Public Health*, *19*(23), 15618. https://doi.org/10.3390/ijerph192315618

Al-Shareeda, M. A., & Manickam, S. (2022b). MSR-DoS: Modular square root-based scheme to resist denial of service (DoS) attacks in 5g-enabled vehicular networks. *IEEE Access*, 10, 120606–120615. https://doi.org/10.1109/ACCESS.2022.3222488

Alani, M. M., & Damiani, E. (2023). XRecon: An explainbale iot reconnaissance attack detection system based on ensemble learning. *Sensors (Basel, Switzerland)*, *23*(11), 5298. https://doi.org/10.3390/s23115298

BBC. (2021). *New Zealand stock exchange halted by cyber-attack*. https://www.bbc.com/news/53918580

Canon. (2020). *The reality of DDoS attacks is becoming more sophisticated and evolving*. https://eset-info.canon-its.jp/malware_info/special/detail/200618.html.

CyberSecurity. (2021). *Damage cases caused by cyber-attacks (crimes) and malware*. https://cybersecurity-jp.com/column/14634

denyhosts. (2022). denyhosts https://github.com/denyhosts/denyhosts

fail2ban. (2022). *fail2ban*. https://github.com/fail2ban/fail2ban

Google. (2022). google-authenticator–libpam. https://github.com/google/google-authenticator-libpam

Hayato, A. (2021). *What is a DDoS attack? Explanation of the purpose and types of attacks, actual examples and countermeasures*. https://business.ntt-east.co.jp/content/cloudsolution/column-185.html#section-02

Kotenko, I., Fedorchenko, E., Novikova, E., & Jha, A. (2023). Cyber attacker profiling for risk analysis based on machine learning. *Sensors (Basel, Switzerland)*, *23*(4), 2028. https://doi.org/10.3390/s23042028

Nick, G. (2021). *Best cloud hosting providers for 2021*. https://hostingtribunal.com/best/cloud-hosting-providers/

Nik, H. (2021). *How many websites are there in theWorld?*. https://siteefy.com/how-many-websites-are-there/

Noman, H. A., & Abu-Sharkh, O. M. F. (2023). Code injection attacks in wireless-based internet of things (IoT):

A comprehensive review and practical implementations. *Sensors (Basel, Switzerland)*, *23*(13), 6067. https://doi.org/10.3390/s23136067

Norton. (2021). *Norton cybercrime research report*. https://prtimes.jp/main/html/rd/p/000000004.000069936.html

Perkel, J. M. (2021). Five reasons why researchers should learn to love the command line. *Nature*, *590*(7844), 173–174. https://doi.org/10.1038/d41586-021-00263-0

Shandilya, S. K., Ganguli, C., Izonin, I., & Nagar, P. A. K. (2022). Cyber attack evaluation dataset for deep packet inspection and analysis. *Data in Brief*, *46*, 108771. https://doi.org/10.1016/j.dib.2022.108771

Shizuka, K. (2021). *Total number of ransomware attacks declines, but damage per attack rises due to maliciousness*. https://cloud.watch.impress.co.jp/docs/news/1240532.html

Solnichkin, A. (2019). *Geolocating SSH hackers in real-time*. https://medium.com/schkn/geolocating-SSH-hackers-in-real-time-108cbc3b5665

Stephanie, H. (2021). Hamilton youth arrested in alleged $46 million crypto theft following joint probe with FBI. https://financialpost.com/fp-finance/cr yptocurrency/hamilton-youth-arrested-in-alleged-46-million-crypto-theft-following-joint-probe-with-fbi.

Steven, J. (2015). *Can the internet exist without Linux?* https://www.zdnet.com/article/can-the-internet-exist-without-linux/

Tatu, Y. (2017). *Security management -Five SSH Facts*. https://www.asisonline.org/security-management-magazine/articles/2017/03/five-SSH-facts/

Techplus. (2021). *91% of the top 1000 sites use HTTPS as their default protocol*. https://news.mynavi.jp/techplus/article/20210205-1689131/

Veracode. (2022). *Protection against spoofing attack*. IP, DNS & ARP. https://www.veracode.com/security/spoofing-attack

Wappalyzer. (2022). *Web services*. https://www.wappalyzer.com/technologies/web-servers\

Wawrowski, Ł., BiałBiałAs, A., Kajzer, A., Kozłowski, A., Kurianowicz, R., Sikora, M., Szymańska-Kwiecień, A., Uchroński, M., Białczak, M., Olejnik, M., & Michalak, M. (2023). Anomaly detection module for network traffic monitoring in public institutions. *Sensors (Basel, Switzerland)*, *23* (6), 2974. https://doi.org/10.3390/s23062974