



Digital Forensic Investigation of Linux SSH Authentication Logs Using NIST SP 800-86 with GeoIP and ASN Enrichment

Danur Wijayanto✉

(Universitas 'Aisyiyah Yogyakarta,
Yogyakarta, Indonesia)

Ahmad Widad Saksana

(Universitas 'Aisyiyah Yogyakarta,
Yogyakarta, Indonesia)

OPEN ACCESS

ARTICLE HISTORY

Received: June 10, 2026

Revised: July 19, 2026

Accepted: August 17,
2026

KEYWORDS

ASN;
Digital forensics;
GeoIP enrichment;
NIST SP 800-86;
SSH authentication logs

ABSTRACT

Purpose – This study applies the National Institute of Standards and Technology Special Publication (NIST SP 800-86) framework to investigate SSH authentication logs from the Lensa UNISA Server while preserving evidence integrity and enriching forensic findings using GeoIP and Autonomous System Number (ASN) information.

Methods – The investigation followed the Collection, Examination, Analysis, and Reporting phases of NIST SP 800-86. The auth.log.1 file was acquired using Secure Copy Protocol (SCP), verified through SHA-256 hashing, and analyzed using standard Linux utilities. GeoIP and ASN information was obtained using MaxMind GeoLite2 and RDAP.

Findings – SHA-256 verification confirmed identical hashes between the original evidence and working copy. The analysis identified 96,182 OpenSSH-related events, including 15,522 failed authentication attempts, 4,604 invalid-user messages, and eight successful sessions. Systematic username enumeration targeting institutional account names was observed. GeoIP and ASN enrichment showed that substantial authentication activity originated from source IP addresses in the Netherlands and Turkey, particularly AS43350, AS48090, and AS154383, consistent with activity associated with hosting infrastructure.

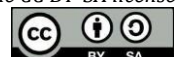
Research Implications – The findings demonstrate the practical applicability of NIST SP 800-86 for SSH log investigation using standard Linux tools. GeoIP and ASN enrichment provide additional context for interpreting authentication artifacts.

Originality – This study presents an operational forensic case study integrating NIST SP 800-86, Linux-based log analysis, and GeoIP–ASN enrichment.

Correspondence Author:✉ danurwijayanto@unisayogya.ac.id

To cite this article: Wijayanto, D., & Saksana, A. W. (2026). Digital Forensic Investigation of Linux SSH Authentication Logs Using NIST SP 800-86 with GeoIP and ASN Enrichment. *Journal of Deep Learning, Computer Vision and Digital Image Processing*, 4(3), 54–76. <https://doi.org/10.61255/decoding.v4i3.1758>

This is an open access article under the CC BY-SA license



INTRODUCTION

Secure Shell (SSH) is a cryptographic network protocol that provides secure communication for remote system administration, secure file transfer, and user authentication over unsecured networks. Owing to its strong security mechanisms, cross-platform compatibility, and broad adoption in Linux-based infrastructures, SSH has become the de facto standard for remote server administration across academic institutions, government agencies, cloud computing environments, and enterprise data centers [1], [2], [3]. Despite its widespread deployment, SSH services remain one of the primary targets of cyberattacks because they are continuously exposed to the Internet and frequently utilized as the main entry point for system administration. Numerous studies have reported that automated brute-force attacks, password guessing, credential stuffing, and username enumeration continue to threaten SSH services, often involving thousands of repeated authentication attempts originating from distributed networks and botnets [1], [2], [4], [5]. These attacks not only aim to obtain unauthorized access but also generate substantial authentication records that may contain valuable digital evidence for reconstructing malicious activities. Consequently, protecting SSH services requires not only robust authentication mechanisms but also systematic collection and preservation of authentication logs to support security auditing, incident response, and digital forensic investigations.

On Linux-based operating systems, SSH authentication activities are automatically recorded through the system logging mechanism, generating authentication logs that document every authentication event occurring on the server. SSH authentication logs record authentication attempts, usernames, source IP addresses, authentication outcomes, and other log information that can be analyzed to identify attack patterns and malicious login activities [6], [7], [8]. Unlike volatile evidence that may disappear after a system reboot, authentication logs preserved in the `/var/log/` directory constitute non-volatile digital evidence capable of supporting forensic reconstruction of security incidents. Previous studies have demonstrated that authentication logs enable investigators to reconstruct attack timelines, identify suspicious authentication patterns, correlate repeated login attempts, and determine the origin of malicious activities based on recorded authentication artifacts [6], [7], [9]. Therefore, authentication logs are not merely operational records for system administrators but also represent an important source of digital evidence that supports systematic forensic investigations following cybersecurity incidents.

Digital forensics provides a scientific methodology for identifying, preserving, examining, analyzing, and presenting digital evidence while maintaining its integrity and evidential value throughout the investigation process [10]. Among various forensic frameworks, the National Institute of Standards and Technology (NIST) Special Publication 800-86 is widely recognized as one of the most established guidelines for integrating forensic techniques into incident response. The framework organizes digital forensic investigations into four sequential phases, namely Collection, Examination, Analysis, and Reporting, thereby ensuring that digital evidence is acquired, processed, interpreted, and documented in a structured and reproducible manner [10], [11], [12]. In addition to supporting evidence integrity, the framework facilitates transparent documentation of investigative procedures, allowing forensic findings to be verified and reproduced by other investigators. Accordingly, NIST SP 800-86 has been extensively adopted in digital forensic investigations involving various categories of digital evidence, including storage media, operating systems, web browsers, and application artifacts [11], [12], [13].

The present study uses SSH authentication logs collected from the operational Lensa Server of Universitas 'Aisyiyah Yogyakarta (UNISA) as a real-world case study for digital forensic investigation. The analyzed evidence consists of the `auth.log.1` file generated through the Linux log rotation mechanism and obtained from the `/var/log/` directory. Since the acquired log file represents a static copy of historical authentication records, forensic acquisition and examination can be performed without interrupting server operations or altering the original evidence. The investigation therefore focuses on reconstructing authentication activities preserved in the log rather than assessing the current security posture of the server. This approach allows investigators to analyze authentication

artifacts while preserving evidence integrity in accordance with established digital forensic principles [10], [11], [13].

Previous studies have investigated SSH authentication logs from several research perspectives. The first group of studies focuses on rule-based detection, where authentication logs are analyzed to identify brute-force attacks, password guessing, and username enumeration by examining authentication frequency, login failures, and repeated access attempts [1], [2], [4]. These approaches are effective for detecting suspicious authentication behavior because they utilize predefined rules derived from authentication patterns. However, rule-based techniques generally emphasize attack detection rather than forensic reconstruction of authentication events after an incident has occurred. A second research direction utilizes SSH honeypots to capture real-world attack activities and analyze attacker behavior. Honeypot-based studies have provided valuable datasets containing millions of SSH authentication attempts, enabling researchers to investigate attack frequency, attacker persistence, credential usage, command execution, and geographical distributions of malicious IP addresses [14], [15], [16], [17]. These studies have substantially improved the understanding of Internet-wide SSH attacks by revealing common attack strategies and identifying long-term trends in automated attacks. Nevertheless, their primary objective is to characterize attacker behavior and collect threat intelligence instead of performing structured forensic investigations based on preserving authentication evidence.

More recently, several researchers have introduced machine learning and statistical approaches to improve the analysis of SSH authentication logs. These studies employ supervised learning, anomaly detection, and predictive modeling techniques to classify authentication events, distinguish malicious activities from legitimate access, and improve detection accuracy [18], [19], [20]. Although machine learning significantly enhances automated detection capabilities, most existing studies continue to treat SSH authentication logs as operational security data for intrusion detection or attack classification. Comparatively little attention has been devoted to treating authentication logs as digital evidence that should be collected, preserved, analyzed and reported through a structured forensic investigation process.

A previous study co-authored by one of the authors applied the NIST SP 800-86 framework to investigate PDF document forgery through structured forensic procedures [21]. The study demonstrated that systematic evidence collection, examination, analysis, and reporting improve the reliability and reproducibility of forensic investigations involving digital documents. Unlike the previous work, the present study extends the application of the NIST SP 800-86 framework to SSH authentication logs obtained from an operational Linux server. Furthermore, this study incorporates GeoIP and Autonomous System Number (ASN) enrichment to provide contextual information regarding the geographical origin and network ownership of authentication attempts, thereby supporting more comprehensive interpretation of authentication artifacts during forensic analysis [22], [23].

Despite the considerable progress reported in previous studies, several research gaps remain. First, existing research predominantly investigates SSH authentication logs from the perspectives of intrusion detection, threat monitoring, honeypot analysis, or machine learning-based classification rather than digital forensic investigation. Second, although the NIST SP 800-86 framework has been extensively adopted in digital forensic investigations involving storage media, operating systems, and application artifacts, its application to SSH authentication logs collected from operational Linux servers remains relatively limited [10], [11], [12], [13]. Third, contextual enrichment techniques such as GeoIP and ASN information have primarily been employed for cyber threat intelligence and attack attribution instead of supporting forensic interpretation of authentication evidence [22], [23], [24], [25]. Consequently, there remains a need for a structured and reproducible digital forensic workflow capable of integrating evidence acquisition, authentication log analysis, GeoIP and ASN enrichment, and systematic forensic reporting within a unified investigation process.

Based on these research gaps, this study applies the NIST SP 800-86 framework to conduct a structured digital forensic investigation of SSH authentication logs collected from the operational Lensa Server of Universitas 'Aisyiyah Yogyakarta (UNISA). The investigation follows the Collection,

Examination, Analysis, and Reporting phases to preserve evidence integrity, extract authentication artifacts, reconstruct authentication activities, and document forensic findings systematically. In addition, GeoIP and ASN enrichment are integrated to provide contextual information for each source IP address, thereby facilitating interpretation of attack origins and network ownership during forensic analysis. This study provides an operational forensic case study demonstrating the application of NIST SP 800-86 to SSH authentication logs, using standard Linux utilities for log extraction and GeoIP-ASN enrichment for contextual interpretation. This study does not develop or evaluate a machine-learning or anomaly-detection method. The structured extraction and session reconstruction procedures are used solely to support the forensic analysis of the investigated SSH authentication evidence. The resulting structured authentication data may be used as input for future studies involving machine-learning or anomaly-detection methods.

METHOD

This research applies a case study method by adopting the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-86 framework as a digital forensic investigation method [10]. The case study approach was chosen because the research was conducted on digital evidence obtained from a Linux-based operational server, so that the investigation process could be carried out in a real environment without changing the conditions of the digital evidence being analyzed. The NIST SP 800-86 framework provides digital forensic investigation procedures consisting of four main stages, namely Collection, Examination, Analysis, and Reporting [10]. These four stages are designed to ensure that the process of acquiring, examining, analyzing, and reporting digital evidence is carried out systematically with documented evidence-preservation procedures [10].

In this study, the NIST SP 800-86 framework was used to investigate SSH authentication artifacts recorded in authentication log files. The focus of the study was not to build an attack detection system or classify network activity, but rather to reconstruct authentication activity based on available digital evidence while preserving the original evidence and performing subsequent analysis on the verified working copy. This study was conducted with the approval and authorization of the Information Systems and Technology Development Agency (BPTSI) of Universitas 'Aisyiyah Yogyakarta (UNISA), which manages the UNISA Lensa Server. Access to the operational server and the auth.log.1 file was officially granted for research purposes only. During the investigation, the researcher made no changes to the server configuration or the content of digital evidence to maintain the integrity of the operational environment.

Investigation Framework

This study adopts the NIST SP 800-86 framework, which consists of four main stages, namely Collection, Examination, Analysis, and Reporting [10]. The framework provides a systematic procedure for acquiring, examining, analyzing, and reporting digital evidence with documented evidence-preservation procedures. The investigative flow applied in this study is shown in Figure 1.

Digital Forensic Investigation Procedure (NIST SP 800-86)

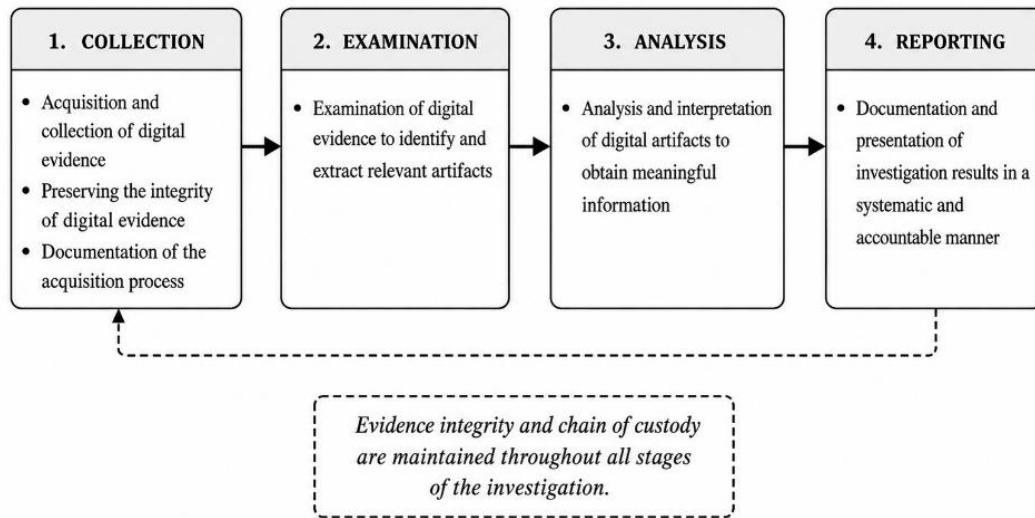


Figure 1. Digital forensic investigation flow based on the NIST SP 800-86 framework.

In the Collection phase, digital evidence metadata is documented using the stat utility, while the integrity of the acquisition results is verified using the sha256sum utility with the SHA-256 algorithm. Next, in the Examination phase, the grep utility is used to extract SSH authentication artifacts, then processed using awk, sort, uniq, and wc to produce structured authentication data. Because the digital evidence was obtained from an operational server that remained active during the research, SHA-256 verification in this study was used to ensure that the acquired copy (working copy) was identical to the original file after the transfer process. Thus, this verification ensures the integrity of the acquisition process, but is not intended to prove the historical integrity of the log files before the acquisition process was carried out.

The Examination phase is performed by extracting SSH authentication artifacts from the auth.log.1 file using the grep utility. Successfully extracted artifacts are then processed using the awk, sort, uniq, and wc utilities to produce a structured dataset containing information on the time of the event (timestamp), source IP address, username, authentication status, and authentication message (log message). This dataset forms the basis for the analysis phase. The analysis phase aims to identify the characteristics of authentication activities based on the extracted artifacts. Analysis was conducted on the distribution of authentication activities, the frequency of username usage, the distribution of source IP addresses, and successful and failed authentication activities. To enhance forensic interpretation, this study also conducted enrichment using GeoIP and Autonomous System Number (ASN) information through the MaxMind GeoLite2 database and RDAP service [26].

The reporting phase is the final stage of the investigation, focusing on systematically documenting the entire process and results. The analysis results are presented in tables, graphs, and narratives to facilitate interpretation of the recovered artifacts. The Investigation Finding Matrix aims to improve traceability between digital evidence, extracted artifacts, the analysis process, investigative findings, and recommendations, ensuring that each investigation result can be verified and reproduced in similar research.

Research Objects and Digital Evidence

This study uses the Lensa Server of Universitas 'Aisyiyah Yogyakarta (UNISA) as the object of investigation. The server is a Linux-based operational server that utilizes Secure Shell (SSH) services as a system administration medium. The selection of the operational server aims to evaluate the application of digital forensic investigations in a real environment so that the research results can

represent actual operational conditions. The digital evidence used is the `auth.log.1` file obtained from the `/var/log/` directory on July 23, 2026. The file is the result of a log rotation mechanism, so the file content is no longer changed after the log rotation process is completed and can be treated as static digital evidence during the investigation process.

The digital evidence analyzed in this study was limited to SSH authentication entries stored in the `auth.log.1` file. Inclusion criteria included log entries generated by the OpenSSH service and containing information on the time of occurrence (timestamp), source IP address, username, process identifier, authentication status, and log message. Conversely, log entries unrelated to SSH authentication, duplicate pre-processed data, and other Linux system logs outside the scope of the study were excluded from the analysis process. These characteristics allow the investigation process to be conducted without affecting server operations or altering the condition of the digital evidence being analyzed. Prior to the investigation, file metadata is documented and the integrity of the digital evidence is verified to ensure that all stages of the analysis are conducted on authentic evidence in accordance with digital forensic investigation principles [10]. This research does not assume that the server has experienced a security compromise, but rather focuses on investigating SSH authentication artifacts recorded in the `auth.log.1` to identify the characteristics of authentication activity based on available digital evidence.

The analyzed `auth.log.1` file contains continuous authentication records spanning an observation period of approximately 46 hours. Specifically, the log covers the period from 19 July 2026 00:00:00 UTC+7 to 20 July 2026 22:11:02 UTC+7. The server clock was configured to the Asia/Jakarta (UTC+7) time zone and synchronized using the system time service throughout the observation period. The authentication log was generated under the default Linux log rotation mechanism, where `auth.log.1` represents the rotated log from the previous logging cycle. Prior to acquisition, the file status was verified to confirm that no active process was modifying the rotated log. The authentication log was managed using the default Ubuntu `rsyslog` log rotation policy. Based on the server configuration, the `auth.log` file was rotated on a weekly basis with four archived log generations retained. The analyzed `auth.log.1` file therefore represented the previous rotation cycle and was verified before acquisition to ensure that it was no longer being actively modified.

Investigation Environment

The investigation in this study was conducted in a Linux-based operating system environment, utilizing standard utilities to support each stage of the digital forensic investigation. This approach was chosen to increase the reproducibility of the study, as the entire process can be reapplied to similar Linux distributions without the need for commercial forensic software. All utilities used are aligned with the investigation stages in the NIST SP 800-86 framework, from digital evidence acquisition to the preparation of investigation results [10]. Thus, each stage of the investigation can be carried out consistently using the built-in utilities of the Linux operating system.

The forensic investigation was conducted on Ubuntu 24.04.4 LTS (Noble Numbat) running OpenSSH 9.6p1 (Ubuntu-3ubuntu13.18) with OpenSSL 3.0.13. GeoIP enrichment was performed using the MaxMind GeoLite2 City database (release: 17 July 2026) and the GeoLite2 ASN database (release: 20 July 2026). ASN ownership information was obtained through RDAP lookups performed on 20 July 2026. Recording the operating system, software version, GeoLite2 database releases, and RDAP lookup date improves the reproducibility and transparency of the forensic workflow by documenting the software environment under which the evidence was acquired and analyzed.

The Collection phase begins with the acquisition of digital evidence in the form of the `auth.log.1` file from the `/var/log/` directory on the Lensa Server of Universitas 'Aisyiyah Yogyakarta (UNISA) using the Secure Copy Protocol (SCP). After the acquisition process is complete, the digital evidence metadata is documented using the `stat` utility, while the file integrity is verified using the SHA-256 algorithm through the `sha256sum` utility. Since the digital evidence was obtained from an operational server that remained active during the research, SHA-256 verification was used to ensure that the acquired copy (working copy) was identical to the original file after the transfer

process. Thus, this verification ensures the integrity of the acquisition process, but is not intended to prove the historical integrity of the log file before the acquisition process was carried out.

Table 1. Digital Forensic Investigation Tools Based on the NIST SP 800-86 Framework

Investigation Stage	Tool	Function
Collection	stat	Identifying digital evidence metadata.
Collection	sha256sum	Verifies file integrity using the SHA-256 algorithm.
Examination	grep	Extracting SSH authentication artifacts.
Examination	awk, sort, uniq, wc	Processing and grouping authentication artifacts into structured data.
Analysis	mmdblookup	Identifying GeoIP and ASN information using the MaxMind GeoLite2 City and GeoLite2 ASN databases.
Analysis	RDAP	Verifying ASN registration information.
Reporting	Microsoft Excel	Presenting the results of the investigation in the form of tables and graphs.

The Analysis stage enriches the extracted source IP addresses using the MaxMind GeoLite2 City and GeoLite2 ASN databases through the mmdblookup utility. GeoLite2 City provides geographical information, whereas GeoLite2 ASN identifies the corresponding Autonomous System Number (ASN) and the registered network organization. ASN registration information was further verified using the Registration Data Access Protocol (RDAP) during the analysis stage [26]. The analysis results were then presented in tables and graphs using Microsoft Excel to facilitate interpretation and reporting of the investigation results. Details of the tools used at each stage of the investigation and their functions are presented in Table 1.

RESULTS AND DISCUSSION

This section presents the findings of a forensic investigation of the SSH authentication logs on the UNISA Lensa Server following the NIST SP 800-86 framework. The analysis was conducted systematically, from evidence preservation to activity reconstruction, to ensure the accountability of the research results [10].

Collection

The Collection phase focuses on securely acquiring and preserving digital evidence. The original evidence is maintained on an operational server, whereas all examination procedures are performed on a working copy to prevent any modification of the original evidence. The characteristics of the acquisition environment are summarized in Table 2. In this study, digital evidence in the form of the auth.log.1 file was obtained from the Lensa Server of Universitas 'Aisyiyah Yogyakarta (UNISA), which functions as an operational server. The file is located in the /var/log/ directory and contains OpenSSH service authentication activities. Because the operational server is still used to support the service, the inspection process was not performed directly on the production system. Instead, the entire investigation process was conducted using a working copy on the research server, while the original evidence was retained on the operational server to maintain the authenticity of the digital evidence.

Evidence Acquisition

The auth.log.1 file was acquired using the Secure Copy Protocol (SCP) to ensure encrypted file transfer, as shown in Figure 2. The command used for evidence acquisition is presented below.

```
scp root@<157.xxx>:/var/log/auth.log.1 ~/penelitian/
```

```

root@srv-ti:~/penelitian# scp root@157.x.x.x:/var/log/auth.log.1 ~/penelitian/
root@157.x.x.x's password:
The authenticity of host '157.x.x.x (157.x.x.x)' can't be established.
ED25519 key fingerprint is SHA256:S9RVK6gttnVq3S4Vly2ZG8xPyGnZHN0GN40D00pJB+PI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '157.x.x.x' (ED25519) to the list of known hosts.
auth.log.1                                100% 15MB 242.5MB/s 00:00
root@srv-ti:~/penelitian# █

```

Figure 2. Acquisition of digital evidence using SCP.

To clarify the investigative environment used, a comparison between the original evidence and the working copy is presented in Table 2.

Table 2. Investigation environment.

Parameter	Operational Server	Research Server
Function	Source of Digital Evidence	Investigation environment
Digital Evidence	Original Evidence	Working Copy
Acquisition Method	SCP over SSH	SCP transfer results

Metadata Comparison Using stat

The metadata identification stage was performed using the stat utility to ensure the integrity and consistency of the basic attributes of the analyzed log files. Based on the examination results, the file size between the original evidence on the operational server and the working copy on the research server was proven to be the same, namely 15,331,173 bytes. Authentic evidence of the examination execution on both server environments can be seen in Figure 3, while the full comparison of metadata attribute values is presented in Table 3.

Server Operational

Server Penelitian

```

root@lensaproxmox:~# stat /var/log/auth.log.1
File: /var/log/auth.log.1
Size: 15331173      Blocks: 29944      IO Block: 4096   regular file
Device: 8,1      Inode: 2784      Links: 1
Access: (0640/-rw-r-----)  Uid: ( 102/ syslog)   Gid: (   4/  adm)
Access: 2026-07-23 13:18:04.409018971 +0700
Modify: 2026-07-20 22:11:02.026391801 +0700
Change: 2026-07-20 22:11:20.337803619 +0700
Birth: 2026-07-20 21:56:23.156866467 +0700
root@lensaproxmox:~# █

```

```

root@srv-ti:~/penelitian# stat auth.log.1
File: auth.log.1
Size: 15331173      Blocks: 29944      IO Block: 4096   regular file
Device: 8,1      Inode: 1321946     Links: 1
Access: (0644/-rw-r--r--)  Uid: (   0/  root)   Gid: (   0/  root)
Access: 2026-07-22 20:53:26.972138669 +0700
Modify: 2026-07-23 13:18:04.469415588 +0700
Change: 2026-07-23 13:18:04.469415588 +0700
Birth: 2026-07-21 13:07:28.832570304 +0700
root@srv-ti:~/penelitian# █

```

Figure 3. Metadata identification with the stat utility.

Based on Table 3, the working copy has an identical file name and file size to the original evidence. Meanwhile, several other metadata attributes, such as owner, permissions, and timestamp, show differences because the acquisition process using SCP produces a new copy on the research server with different file system characteristics. These differences are a normal consequence of the file copying mechanism and do not directly indicate any changes to the file content. Therefore, an integrity verification process using the SHA-256 algorithm is required to ensure that the content of the working copy remains identical to the original evidence. Because SCP creates a new file on the destination filesystem, metadata such as inode number, ownership, file permissions, and filesystem timestamps are expected to differ even when file contents remain identical.

Table 3. Metadata comparison results

Parameter	Original Evidence (Operational Server)	Working Copy (Research Server)
File Name	auth.log.1	auth.log.1
File Size	15,331,173 bytes	15,331,173 bytes
Inode	2784	1321946
Permissions Mode	0640 (-rw-r-----)	0644 (-rw-r--r--)

Parameter	Original Evidence (Operational Server)	Working Copy (Research Server)
User / Group Ownership	102 (syslog) / 4 (adm)	0 (root) / 0 (root)
Access Timestamp	2026-07-23 13:18:04.409018971 +0700	2026-07-23 13:19:15.972138669 +0700
Modification Timestamp	2026-07-20 22:11:02.026391801 +0700	2026-07-23 13:18:04.469415588 +0700
Change Timestamp	2026-07-20 22:11:20.337803619 +0700	2026-07-23 13:18:04.469415588 +0700
SHA-256 Hash	d1f9ebb0af7bcd1bd64e17b813a794c975f8756f1c30c8a9db34b1f351d0c863	d1f9ebb0af7bcd1bd64e17b813a794c975f8756f1c30c8a9db34b1f351d0c863

Integrity Verification Using sha256sum

The calculated SHA-256 hash values were identical for both the original evidence and the working copy, confirming that no modification occurred during the acquisition process [10]. Screenshots of the verification command execution in both environments are presented in Figure 4, while the details of the hash value comparison can be seen in Table 4.



Figure 4. SHA-256 hash verification.

Based on Table 4, the SHA-256 values of the original evidence and the working copy are identical. This matching hash value indicates that the SCP acquisition process did not alter the contents of the auth.log.1 file during transfer. Thus, the working copy was verified to have identical content to the original evidence at the time of acquisition and was subsequently used for the Examination, Analysis, and Reporting stages under documented access controls, in accordance with the evidence-preservation principles of NIST SP 800-86 [10].

Table 4. Integrity verification results

Proof	SHA-256	Status
Original	d1f9ebb0af7bcd1bd64e17b813a794c975f8756f1c30c8a9db34b1f351d0c863	Valid
Working Copy	d1f9ebb0af7bcd1bd64e17b813a794c975f8756f1c30c8a9db34b1f351d0c863	Valid

The entire investigation process is conducted on a working copy on the research server, while the original evidence remains on the operational server. This approach aligns with the evidence preservation principle in NIST SP 800-86, which maintains the original digital evidence so that it remains unchanged during the investigation. To strengthen evidence traceability and ensure forensic accountability, the acquisition and handling of the digital evidence were documented throughout the investigation. Information regarding the evidence identifier, acquisition details, investigator, storage location, integrity verification, and access control was recorded as part of the chain-of-custody documentation in accordance with the evidence preservation principles described in NIST SP 800-86. To maintain evidence traceability throughout the investigation, all evidence acquisition and handling activities were documented using a Chain of Custody record. The documentation includes the evidence identifier, acquisition details, integrity verification, storage location, and access control information, as presented in Table 5.

Table 5. Chain of Custody Record

Item	Description
Case Identifier	UNISA-SSH-2026-001
Evidence Identifier	EV-SSH-001
Evidence Source	/var/log/auth.log.1

Item	Description
Investigator	Danur Wijayanto
Acquisition Date and Time	23 July 2026 13:18 UTC+7
Acquisition Method	Secure Copy Protocol (SCP) over SSH
Source Server	Lensa UNISA Linux Server
Time Zone	Asia/Jakarta (UTC+7)
Integrity Verification	SHA-256 hash comparison
Evidence Storage	Dedicated forensic research workstation
Access Control	Accessible only to the authorized investigator

Following acquisition, the integrity-verified working copy was stored on a dedicated forensic research workstation with restricted investigator access. The original evidence remained on the operational server without modification throughout the investigation. All forensic examination, parsing, GeoIP enrichment, ASN analysis, and reporting activities were performed exclusively on the verified working copy to preserve the integrity of the original evidence.

Examination

The Examination phase aims to extract and organize raw artifacts into a structured dataset. This process utilizes Linux utilities (grep, awk, sort, uniq, head, wc) to filter OpenSSH activity relevant to the investigation's objectives.

Authentication Log Identification

The Examination phase begins with isolating the OpenSSH service log using grep. A total of 96,182 authentication activity entries were identified in the log file, as shown in Figure 5. The command used is shown below.

grep "sshd" auth.log.1 | wc -l

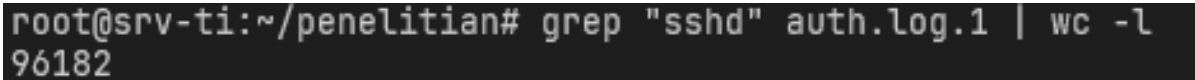


Figure 5. Identifying OpenSSH entries.

This command combines the grep utility with wc -l. The wc (word count) utility is used to count the number of extracted lines to obtain the total number of SSH authentication events recorded in the log file. The inspection yielded 96,182 authentication activity entries related to the OpenSSH service. These entries served as the primary data source for subsequent inspections.

Authentication Artifact Extraction

Authentication artifact extraction was performed using a combination of the grep and awk utilities to automatically extract usernames from failed SSH authentication events. The extraction process preserved the original log structure while separating usernames associated with both valid and invalid authentication attempts. The command used for artifact extraction is presented in Code Listing 1, and its execution is illustrated in Figure 6.

Code Listing 1. Log Extraction and Analysis Script	
Script / Command	Description
grep "Failed keyboard-interactive/pam" auth.log.1 \n awk '{ for(i=1;i<=NF;i++){ if(\$i=="invalid" && \$(i+1)=="user"){ print \$(i+2); next; } if(\$i=="for" && \$(i+1)!="invalid"){ print \$(i+1); next; } } }' \n grep -v '^#\$' \n	Extracts usernames from failed authentication logs and calculates the top 10 users.

Script / Command	Description
sort \	
uniq -c \	
sort -nr \	
head -10	

Figure 6 illustrates the execution of the username extraction process using the awk utility. The extracted usernames constitute one of the authentication artifacts used during the Examination phase and are subsequently utilized for frequency analysis and behavioral interpretation in the Analysis phase.

```
root@srv-ti:~/penelitian# grep "Failed keyboard-interactive/pam" auth.log.1 \
| awk '{
  for(i=1;i≤NF;i++){
    if($i="invalid" && $(i+1)="user"){
      print $(i+2);
      next;
    }
    if($i="for" && $(i+1)≠"invalid"){
      print $(i+1);
      next;
    }
  }
}' \
| grep -v '^#\$' \
| sort | uniq -c | sort -nr | head -10
1473 unisa
1472 unisayk
1472 unisa_yk
1472 unisa123
1472 unisa-yk
885 unisa.jogja
742 unisa_aisyiyah
742 unisa.yogyakarta
741 aisyyiyah
737 aisyyiyah-unisa
root@srv-ti:~/penelitian#
```

Figure 6. Artifact extraction with the awk utility.

Authentication Data Set Preparation

In the dataset preparation phase, the extraction of the ten IP addresses with the highest frequency of authentication activity was performed through a series of Bash command pipelines in the system environment. The first step involved the grep utility to filter log entries specifically related to the sshd service. Next, a regular expression (Regex) with the pattern '([0-9]{1,3}\.){3}[0-9]{1,3}' was applied to extract IPv4 IP address entities from the raw log text. The extracted IP address data was then sorted using the sort command, grouped, and calculated for total frequency using the uniq -c utility. Finally, the dataset was sorted again in descending order of frequency using sort -nr, and limited to the top ten entities using the head -10 utility. A screenshot of this command execution is shown in Figure 7, while the final IP address data structure and its frequency are comprehensively summarized in Table 6.

```
root@srv-ti:~/penelitian# grep "sshd" auth.log.1 \
| grep -oE '([0-9]{1,3}\.){3}[0-9]{1,3}' \
| sort \
| uniq -c \
| sort -nr \
| head -10
47650 185.107.56.76
20438 185.107.56.251
953 176.53.159.197
519 45.148.10.151
512 176.53.159.198
496 45.148.10.157
447 62.60.130.219
437 45.148.10.152
412 45.148.10.147
412 45.148.10.141
root@srv-ti:~/penelitian#
```

Figure 7. Authentication data aggregation.

The results of this process are not analytical results, but rather a structured collection of authentication data ready for use in the Analysis phase. This data set forms the basis for identifying authentication characteristics, username usage patterns, IP address distribution, and geographic and network information enrichment. The extraction process produces SSH authentication artifacts containing information on the time of the event (timestamp), source IP address, username, authentication status, and authentication message (log message). An example of the results of authentication artifact extraction is shown in Table 6.

Table 6. Authentication Data Attributes of Extracted Results

Timestamp	Source IP	Username	Authentication Status	Log Message
Jul 20 00:31:05	185.107.56.76	Unisa	Invalid User	Invalid user unisa from 185.107.56.76 port 59844
Jul 19 23:02:07	185.107.56.251	root	Disconnected	Received disconnect from 185.107.56.251 port 21154:11: Bye Bye [preauth]
Jul 19 01:58:34	176.53.159.197	support	Invalid User	Invalid user support from 176.53.159.197 port 14844

Table 6 shows examples of extracted authentication artifacts used as input in the Analysis phase. All successfully extracted artifacts were then processed to identify the characteristics of SSH authentication activity based on frequency of occurrence, username usage patterns, source IP address distribution, and contextual information derived from GeoIP and ASN enrichment.

Analysis

The Analysis phase interprets relationships between artifacts to reconstruct activity patterns. The analysis focuses on identifying anomalies, identifying target accounts, and distributing attack sources based on verified digital evidence.

Authentication Event Analysis

Event distribution analysis shows a predominance of authentication failures. The grep and wc utilities were used to calculate the frequency of each log category Figure 8 and Table 7. In this study, the identification process was performed using the grep utility combined with wc to calculate the frequency of occurrence of each type of authentication event. This approach allows investigators to determine the distribution of authentication activity based on the log categories recorded by the system. For example, the number of successful authentications is counted using the following command.

grep "Accepted keyboard-interactive/pam" auth.log.1 | wc -l

While failed authentication is counted using the following command.

grep "Failed keyboard-interactive/pam" auth.log.1 | wc -l

Similar commands are used to count artifacts of authentication failures, invalid users, pending authentications, disconnections received, connections closed, connections reset, and authentications rejected.

```

root@srv-ti:~/penelitian# grep "Accepted keyboard-interactive/pam" auth.log.1 | wc -l
15522
root@srv-ti:~/penelitian# grep "Failed keyboard-interactive/pam" auth.log.1 | wc -l
19144
root@srv-ti:~/penelitian# grep "authentication failure" auth.log.1 | wc -l
4684
root@srv-ti:~/penelitian# grep "Invalid user" auth.log.1 | wc -l
892
root@srv-ti:~/penelitian# grep "Connection closed" auth.log.1 | wc -l
2812
root@srv-ti:~/penelitian# grep "Received disconnect" auth.log.1 | wc -l
485
root@srv-ti:~/penelitian# grep "Connection reset" auth.log.1 | wc -l
89
root@srv-ti:~/penelitian# grep "Authentication refused" auth.log.1 | wc -l
8

```

Figure 8. Authentication event calculation.

Because multiple OpenSSH log messages may be generated during a single authentication session, the frequencies presented in Table 7 represent message occurrences rather than independent SSH sessions.

Table 7. Summary of SSH Event Frequency

Event Category (SSH)	Frequency
Total SSH Event	96,182
Authentication Failure	19,144
Postponed Keyboard-Interactive	15,530
Failed Keyboard-Interactive/PAM	15,522
Invalid User	4,604
Received Disconnect	2,812
Connection Closed	892
Connection Reset	485
Authentication Refused	89
Accepted Keyboard-Interactive/PAM	8

It should be noted that counting individual OpenSSH log messages does not accurately reflect the number of independent SSH sessions, as a single connection can generate multiple overlapping lifecycle messages. To reconstruct SSH sessions, distinct sshd Process Identifiers (PIDs) were used as the sessionization unit. Because the available authentication log does not provide a dedicated session identifier, log messages associated with the same sshd PID were grouped as one reconstructed session. This procedure identified 23,544 reconstructed SSH sessions from the 96,182 OpenSSH-related log entries. The following command was used to identify and count the distinct sshd Process Identifiers (PIDs): `grep "sshd" auth.log.1 | grep -o 'sshd\[([0-9]*)\]' | sort | uniq | wc -l`.

The command identified 23,544 unique sshd PIDs in the authentication log. For this investigation, grouping log messages by sshd PID was used as a practical sessionization approach because the available authentication log does not provide a dedicated session identifier. This approach provides a reasonable approximation of individual SSH sessions and is consistent with the use of process identifiers as sessionization units when dedicated session identifiers are unavailable [26]. The reconstructed session analysis identified 23,544 SSH sessions, including 8 successful sessions and 21,757 sessions containing failed-authentication indicators. A total of 270 unique source IP addresses were identified. The two dominant source IP addresses, 185.107.56.76 and 185.107.56.251, were associated with 12,449 and 5,303 reconstructed sessions, respectively. These session-level results were analyzed separately from the raw OpenSSH message frequencies reported in Table 7.

Username Analysis

Username analysis revealed the use of variations of organized institutional names Table 8. This confirms the activity of username enumeration to search for valid accounts Figure 9. User name extraction is performed using a combination of the grep, awk, sort, and uniq utilities. The awk utility is used to separate usernames from each log line, while sort and uniq are used to group usernames based on their frequency of occurrence. The command used is shown below.

Code Listing 2. Username Analysis Script

Script / Command	Description
<pre>grep "Failed keyboard- interactive/pam" auth.log.1 awk '{ for(i=1;i<=NF;i++){ if(\$i=="invalid" && \$(i+1)=="user"){ print \$(i+2); next; } if(\$i=="for" && \$(i+1)!="invalid"){ print \$(i+1); next; } } }' sort uniq -c sort - nr</pre>	Extracts usernames from failed authentication logs, then sorts and counts their occurrence frequency in descending order.

```

root@srv-ti:~/penelitian# grep "Failed keyboard-interactive/pam" auth.log.1 | awk '{
    for(i=1;i≤NF;i++){
        if($i="invalid" && $(i+1)="user"){
            print $(i+2);
            next;
        }
        if($i="for" && $(i+1)≠"invalid"){
            print $(i+1);
            next;
        }
    }
}' | grep -v '^#\$' | sort | uniq -c | sort -nr
1473 unisa
1472 unisayk
1472 unisa_yk
1472 unisa123
1472 unisa-yk
885 unisa.jogja
742 unisa_aisyiyah
742 unisa.yogyakarta
741 aisyyiah
737 aisyyiah-unisa
732 aisyyiah_unisa
731 aisyyiahjogja
730 aisyyiahogyakarta
430 root
80 support
71 admin
16 user
8 test
5 user3
5 ui
5 ubnt

```

Figure 9. Distribution of target usernames.

Based on Table 8, most of the usernames used in authentication attempts were variations of institutional names, such as unisa, unisa-yk, unisayk, unisa_yk, and unisa123. This pattern indicates that authentication attempts were not conducted randomly, but utilized variations of usernames related to organizational identities. These characteristics indicate username enumeration activity, namely the attempt to try various combinations of account names that are considered valid before the authentication process is successful. Analysis of successful authentications revealed that only two accounts successfully logged in: root (7 times) and eduunisa (1 time). This finding suggests that legitimate authentication activity was performed by only a small number of accounts, while the majority of recorded activity was failed authentication attempts. The observed naming pattern indicates that attackers deliberately targeted usernames associated with the institutional identity rather than randomly generated account names.

Table 8. Username identification results

No.	Username	Frequency
1	unisa	1,473
2	unisayk	1,472
3	unisa_yk	1,472
4	unisa123	1,472
5	unisa-yk	1,472
6	unisa.jogja	885
7	unisa_aisyiyah	742
8	unisa.yogyakarta	742
9	aisyiyah	741
10	aisyiyah-unisa	737

Source IP Address Analysis

Source IP address mapping identified addresses with anomalous connection intensity Figure 10. The dominant IP addresses are summarized in Table 9. The identification process is performed using a combination of the grep, sort, uniq, and head utilities. The grep utility is used to extract IP addresses from each log entry using regular expression (regex) patterns. Next, the sort utility is used to sort the IP addresses, uniq -c is used to count the frequency of each IP address, and head is used to display the ten IP addresses with the highest frequency of activity.

Code Listing 3. IP Address Mapping Script

Script / Command	Description
grep "sshd" auth.log.1 \ grep -oE '([0-9]{1,3}\.){3}[0-9]{1,3}' \ sort \ uniq -c \ sort -nr \ head -10	Extracts IPv4 address entities from sshd log entries using regular expression (regex) patterns, then sorts and displays the 10 IP addresses with the highest activity frequency.

The identification results show that several IP addresses had higher authentication activity frequencies than others. The ten IP addresses with the highest authentication activity frequencies are presented in Table 9.

Table 9. List of IP Addresses and Highest Frequencies

No.	Source IP Address	Frequency
1	185.107.56.76	47,650
2	185.107.56.251	20,438
3	176.53.159.197	953
4	45.148.10.151	519
5	176.53.159.198	512
6	45.148.10.157	496
7	62.60.130.219	447
8	45.148.10.152	437
9	45.148.10.147	412
10	45.148.10.141	412

Based on Table 9, IP address 185.107.56.76 is the Source IP Address with the highest frequency of authentication activity, with 47,650 occurrences, followed by 185.107.56.251 with 20,438 occurrences. The other eight Source IP Addresses have lower activity frequencies, ranging from 412 to 953 occurrences. These results indicate that most of the authentication activity recorded during the observation period originated from these two Source IP Addresses. However, Source IP Address information obtained from authentication logs cannot be used as a basis for directly identifying the perpetrator's identity. Therefore, GeoIP and Autonomous System Number (ASN) enrichment processes are required to obtain additional information regarding the geographic location and network organization associated with each Source IP Address before further interpretation can be carried out.

```
root@srv-ti:~/penelitian# grep "sshd" auth.log.1 \
| grep -oE '([0-9]{1,3}\.){3}[0-9]{1,3}' \
| sort \
| uniq -c \
| sort -nr \
| head -10
47650 185.107.56.76
20438 185.107.56.251
953 176.53.159.197
519 45.148.10.151
512 176.53.159.198
496 45.148.10.157
447 62.60.130.219
437 45.148.10.152
412 45.148.10.147
412 45.148.10.141
root@srv-ti:~/penelitian#
```

Figure 10. Identifying dominant source IP addresses.

The operational server was configured to use the Asia/Jakarta (UTC+7) time zone. At the time of evidence acquisition, the system clock was synchronized through the active Network Time Protocol (NTP) service, ensuring consistent timestamp generation throughout the observation period.

GeoIP and Autonomous System Number (ASN) Analysis

Enrichment of geographic and Autonomous System Number (ASN) enrichment was performed using the mmdblookup utility with the MaxMind GeoLite2 City database (release: 17 July 2026) and the GeoLite2 ASN database (release: 20 July 2026). The GeoLite2 City database was used to identify the geographical location associated with each source IP address, whereas the GeoLite2 ASN database was used to identify the corresponding Autonomous System Number (ASN) and registered network organization. Where necessary, ASN registration information was further verified through the Registration Data Access Protocol (RDAP) during the analysis stage. The commands used for GeoIP and ASN identification are presented in Code Listing 4.

Code Listing 4. Geolocation and ASN Identification Script

Script / Command	Description
mmdblookup --file GeoLite2-City.mmdb --ip 185.107.56.76	Performs geolocation information lookup (such as city and country) of anomalous source IP addresses using the GeoLite2-City database.
mmdblookup --file GeoLite2-ASN.mmdb --ip 185.107.56.76	Extracts Autonomous System Number (ASN) information and the identity of the associated internet service provider organization from a source IP address.

```
root@srv-ti:~/penelitian# mmdblookup \
--file GeoLite2-City_20260717/GeoLite2-City.mmdb \
--ip 185.107.56.76
{
  "continent":
  {
    "code":
    "EU" <utf8_string>
    "geoname_id":
    6255148 <uint32>
    "names":
    {
      "de":
      "Europa" <utf8_string>
      "en":
      "Europe" <utf8_string>
      "es":
      "Europa" <utf8_string>
      "fr":
      "Europe" <utf8_string>
      "ja":
      "ヨーロッパ" <utf8_string>
      "pt-BR":
      "Europa" <utf8_string>
      "ru":
      "Европа" <utf8_string>
      "zh-CN":
      "欧洲" <utf8_string>
    }
  }
}
```

Figure 11. GeoIP Identification.

```
root@srv-ti:~/penelitian# mmdblookup \
--file GeoLite2-ASN_20260720/GeoLite2-ASN.mmdb \
--ip 185.107.56.76
{
  "autonomous_system_number":
  43350 <uint32>
  "autonomous_system_organization":
  "NForce Entertainment B.V." <utf8_string>
}
root@srv-ti:~/penelitian#
```

Figure 12. ASN identification.

The results of the information enrichment process for ten IP addresses with

Table 10. IP Address, Geolocation, and ASN Mapping

No.	Source IP Address	Frequency	Country	ASN	Organization
1	185.107.56.76	47,650	The Netherlands	AS43350	NForce Entertainment BV
2	185.107.56.251	20,438	The Netherlands	AS43350	NForce Entertainment BV

No.	Source IP Address	Frequency	Country	ASN	Organization
3	176.53.159.197	953	Turkey	AS154383	ZORNTECH WEB SOLUTIONS
4	45.148.10.151	519	The Netherlands	AS48090	Techoff Srv Limited
5	176.53.159.198	512	Turkey	AS154383	ZORNTECH WEB SOLUTIONS
6	45.148.10.157	496	The Netherlands	AS48090	Techoff Srv Limited
7	62.60.130.219	447	Iran	AS215930	Cipher Operations Doo Beograd – Novi Beograd
8	45.148.10.152	437	The Netherlands	AS48090	Techoff Srv Limited
9	45.148.10.147	412	The Netherlands	AS48090	Techoff Srv Limited
10	45.148.10.141	412	The Netherlands	AS48090	Techoff Srv Limited

Based on Table 10, the GeoIP and Autonomous System Number (ASN) enrichment process successfully provided contextual information for the ten Source IP Addresses with the highest frequency of authentication activity. The two Source IP Addresses with the highest frequency of activity, namely 185.107.56.76 and 185.107.56.251, were identified as originating from The Netherlands and are located in Autonomous System AS43350 (NForce Entertainment B.V.). In addition, 176.53.159.197 and 176.53.159.198 were identified as originating from Türkiye and are located in AS154383 (ZORNTECH WEB SOLUTIONS). Five other Source IP Addresses, namely 45.148.10.151, 45.148.10.157, 45.148.10.152, 45.148.10.147, and 45.148.10.141, originate from The Netherlands and are associated with AS48090 (Techoff Srv Limited). Meanwhile, 62.60.130.219 is identified as originating from Iran and is located on AS215930 (Cipher Operations Doo Beograd – Novi Beograd). GeoIP and ASN information are used to enrich the context of the investigation by providing information about the geographic location and network organization associated with authentication activity, not to directly identify the perpetrator's identity. Integration of authentication logs with GeoIP and ASN information provides a more comprehensive understanding of connection patterns on OpenSSH services and supports the analysis process within the NIST SP 800-86 framework [26].

It should be noted that the IP address 62.60.130.219 indicates a geographic location in Iran based on the GeoLite2 database, while the ASN information identifies the organization Cipher Operations Doo Beograd – Novi Beograd, registered in Serbia. This discrepancy indicates that the geographic location of the GeoIP results does not always correspond to the country where the ASN organization is registered. This can be influenced by the use of cross-border network infrastructure, such as VPN services, international hosting providers, or other network addressing mechanisms. Therefore, GeoIP and ASN information need to be interpreted in a complementary manner to support forensic analysis.

During the analysis phase, seven successful SSH root authentication events were identified in the authentication log. Because successful root logins represent security-sensitive events, these records were examined separately from the unsuccessful authentication attempts discussed in the previous sections. Table 11 summarizes the timestamps, source IP addresses, source ports, authentication methods, and forensic observations associated with the successful root authentication events.

Table 11. Summary of All Successful Authentication Events

No.	Timestamp	Source IP	Source Port	Authentication Method	Observation
1	2026-07-19 12:53:53	157.85.xxx.xxx	25262	Keyboard- interactive/PAM	Successful root authentication

No.	Timestamp	Source IP	Source Port	Authentication Method	Observation
2	2026-07-19 13:42:29	157.85.xxx.xxx	4212	Keyboard- interactive/PAM	Successful root authentication
3	2026-07-19 13:49:38	157.85.xxx.xxx	15605	Keyboard- interactive/PAM	Successful root authentication
4	2026-07-19 18:01:59	157.85.xxx.xxx	4198	Keyboard- interactive/PAM	Successful root authentication
5	2026-07-19 22:36:28	157.85.xxx.xxx	7798	Keyboard- interactive/PAM	Successful root authentication
6	2026-07-20 18:27:08	157.85.xxx.xxx	3712	Keyboard- interactive/PAM	Successful root authentication
7	2026-07-20 21:55:36	157.85.xxx.xxx	4223	Keyboard- interactive/PAM	Successful root authentication
8	2026-07-20 13:53:51	172.16.xxx.xxx	56702	Keyboard- interactive/PAM	Successful eduunisa authentication

Table 11 shows that the seven root authentication events originated from the same source IP address, while the single successful login for the eduunisa account originated from a distinct IP segment. Based on the available authentication log, no direct evidence indicates that the observed brute-force authentication activity resulted in unauthorized access during the observation period. The available authentication log does not contain sufficient evidence to determine whether the successful root authentication events originated from an authorized administrator or another trusted management host. Verification of the operational context would require administrative records or infrastructure documentation, which were outside the scope of the present study. Therefore, the successful root authentication events were reported as observed forensic artifacts without attributing them to either legitimate administrative activity or unauthorized access.

Reporting

The Reporting stage is the final stage in the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-86 framework, which aims to document the entire investigation process along with the results obtained during the research [10]. All findings are compiled into a technical report that systematically contains digital evidence, extracted artifacts, analysis results, and forensic interpretation. The report is prepared to ensure that each stage of the investigation can be traced back and to support the verification and reproducibility of the research.

Based on the Collection stage, the research successfully obtained the auth.log.1 file as digital evidence from the Lensa Server of Universitas 'Aisyiyah Yogyakarta (UNISA). The integrity of the evidence was then verified using the SHA-256 algorithm, which showed that the working copy had an identical hash value to the original evidence. Furthermore, in the Examination stage, the authentication artifact was successfully extracted using the grep, awk, sort, uniq, head, and wc utilities to obtain a collection of authentication data consisting of timestamp, username, source IP address, authentication status, and log message as the basis for the analysis.

The analysis phase revealed that the auth.log.1 file contained 96,182 activities related to the OpenSSH service. The analysis revealed a predominance of failed authentication activities, the use of username variations that resembled institutional identities, and several IP addresses that had significantly higher activity frequencies than others. Furthermore, the enrichment process using GeoIP and Autonomous System Number (ASN) information successfully provided additional information regarding the country of origin and the network organization managing the IP addresses, thus supporting the interpretation of the characteristics of the authentication activities found during the investigation.

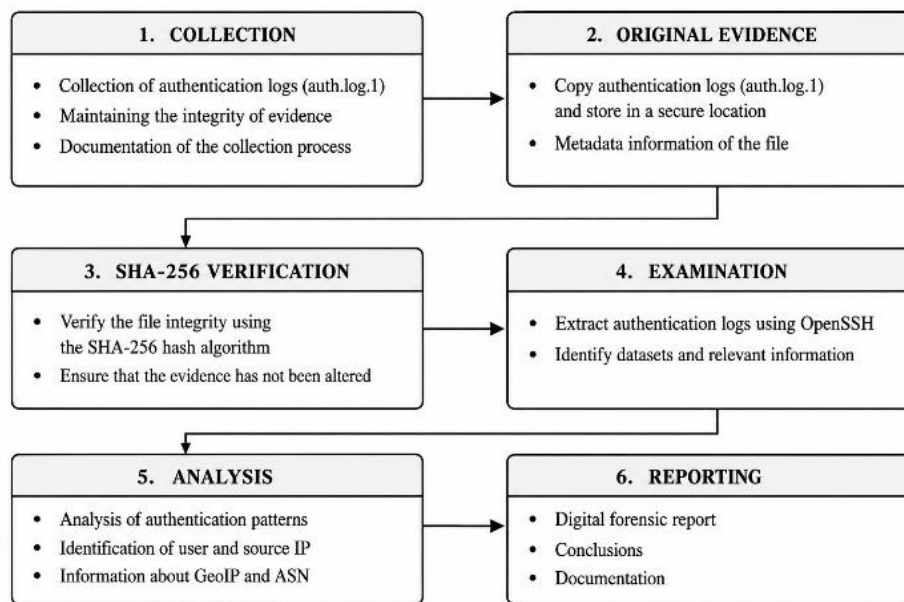


Figure 13. Implementation of the NIST SP 800-86 framework.

To provide a summary of the investigation results, all the main findings of the research are presented in Table 12. This summary shows the relationship between each stage of the investigation and the results obtained, thus facilitating the process of tracing back the digital evidence and artifacts analyzed.

Table 12. Summary of Main Findings of the Investigation	
Stages	Key Findings
Collection	The integrity of digital evidence is verified by identical SHA-256 hash values between the original evidence and the working copy.
Examination	Authentication artifacts were successfully extracted and compiled into a structured dataset consisting of timestamp, username, source IP address, authentication status, and log message.
Analysis	96,182 OpenSSH activities were identified, dominated by failed authentication events, repeated automated authentication attempts, and GeoIP and ASN enrichment results.
Reporting	All investigation results were successfully documented in the form of technical reports and Investigation Finding Matrix.

Based on the summary from the Reporting stage, all investigative findings are then comprehensively mapped into an Investigation Finding Matrix. This matrix serves to document the common thread between analyzed digital evidence, extracted artifacts, technical analysis results, and proposed security recommendations for incident mitigation. Details of the matrix are presented in Table 13.

Table 13. Investigation Finding Matrix				
Evidence	Artifact	Analysis	Finding	Recommendation
auth.log.1	Failed Keyboard-Interactive/PAM	15,522 Failed Keyboard-Interactive/PAM messages	Repeated automated authentication attempts	Fail2Ban implementation and Public Key based authentication

auth.log.1	Invalid User	4,604 Invalid User messages involving variations of institutional names	Username Enumeration activity	Rate limiting of login attempts
auth.log.1	Source IP & GeoIP	High frequency anomaly from the Netherlands (AS43350)	Activity consistent with automated authentication attempts originating from hosting infrastructure	Prioritize Fail2Ban and rate limiting; apply context-dependent IP range blocking (Geo-blocking) narrowly based on operational evidence

Based on the findings matrix above, the investigation results show a pattern of authentication activity consistent with repeated automated authentication activity on the Lensa Server of Universitas 'Aisyiyah Yogyakarta (UNISA). However, attribution regarding the level of organization or the perpetrator's intent cannot be determined solely based on authentication log analysis and requires additional sources of evidence, such as threat intelligence or other digital artifacts. The high frequency of recorded authentication activity from the same IP address within a relatively short timeframe is consistent with the use of automated tools to conduct repeated login attempts [1], [2]. These findings are in line with research by Agghey et al. [1], which explains that attacks on SSH services generally begin with a username enumeration process through various combinations of commonly used accounts. A similar pattern is also seen in this study through the large number of invalid user entries using variations of institutional names. The results of this study are also consistent with a survey conducted by Hamza and Al-Janabi [18], which explains that brute-force attacks on SSH services remain one of the most common threats on Linux-based systems and can be identified through repeated authentication patterns in authentication logs. However, this research focuses on digital forensic investigation of authentication artifacts, not on the application of machine learning-based detection methods.

Furthermore, the high number of failed authentication events is consistent with repeated automated authentication attempts. However, because SSH authentication logs do not record the passwords used during unsuccessful login attempts, the available evidence does not permit confirmation that a dictionary attack was performed. Therefore, the observed events should be interpreted as repeated authentication attempts rather than evidence of a specific password-guessing technique. Authentication logs containing repeated abnormal authentication events remain valuable forensic artifacts for identifying suspicious system activities and supporting further forensic investigations [27]. In addition, Alkhwaja et al. [28] demonstrated that brute-force and dictionary attacks remain highly effective against weak authentication credentials, emphasizing the importance of strong password policies and authentication hardening. Although most authentication attempts were unsuccessful, the high volume of recorded events has the potential to cause log flooding, making it more difficult for administrators to identify genuinely malicious activity.

When compared with the research of Wu et al. [5], the geographical distribution of IP addresses originating from the Netherlands and Turkey indicates that international cloud infrastructure is still frequently exploited as a jumping-off point in attacks against SSH services. Based on these results, the implementation of active defense mechanisms such as Fail2Ban, public key-based authentication, and SSH access restrictions are relevant steps to reduce the risk of attacks [28], [29]. In addition, the implementation of the NIST SP 800-86 framework in this study provided a systematic investigation process in which the original evidence was preserved and subsequent analysis was performed on the verified working copy under documented access controls. The integration of standard Linux utilities with GeoIP and ASN information enrichment also improves the quality of forensic interpretation and supports the reproducibility of research in Linux-based digital forensic investigations.

CONCLUSION

This research successfully applied the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-86 framework to conduct a digital forensic investigation of the OpenSSH service authentication log on the Lensa Server of Universitas 'Aisyiyah Yogyakarta (UNISA). The

investigation process was carried out systematically through the stages of Collection, Examination, Analysis, and Reporting, so that each process can be traced and documented. Verification using the SHA-256 algorithm confirmed that the transferred working copy was identical to the original evidence exactly at the time of acquisition. The working copy was subsequently analyzed under documented access controls to protect the investigation environment. The investigation revealed that the `auth.log.1` file contained authentication activity dominated by failed authentication messages, with only a very limited number of successful authentications. Further analysis revealed that two IP addresses belonging to Autonomous System AS43350 (NForce Entertainment BV, The Netherlands) accounted for more than 70% of all recorded SSH authentication activity. Furthermore, enrichment using GeoIP and Autonomous System Number (ASN) successfully provided contextual information regarding the geographic location and network organization associated with the source IP address without attributing the perpetrator's identity.

This study demonstrates the practical application of the NIST SP 800-86 framework to an SSH authentication log from an operational Linux server. The use of standard Linux utilities supported the extraction and reconstruction of authentication artifacts, while GeoIP and ASN enrichment provided contextual information for interpreting source IP activity. The contribution of this study is therefore limited to an operational forensic case study rather than the development of a new forensic method or machine-learning detection approach. This approach preserves the original evidence while enabling subsequent forensic analysis to be performed on the verified working copy under documented access controls. This research still has several limitations because the investigation was only conducted on one type of digital evidence, namely the `auth.log.1` file, so the analysis results are limited to authentication activities recorded by the OpenSSH service. In addition, the enrichment process only utilizes the GeoLite2 City and GeoLite2 ASN databases so that the contextual information obtained depends on the scope and update of these databases. Further research is recommended to integrate various other Source of Digital Evidence, such as system logs, audit logs, firewall logs, and network traffic capture, and combine them with threat intelligence, machine learning, or cross-source log correlation techniques to produce a more comprehensive investigation process and support a more in-depth analysis of information security incidents [23]. The findings also illustrate how standard Linux utilities and contextual GeoIP–ASN enrichment can support forensic analysis of SSH authentication activity in an operational server environment. This observation is limited to the investigated case and does not imply a general-purpose forensic method or security monitoring system.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Agency for Technology and Information Systems Development (BPTSI) of Universitas 'Aisyiyah Yogyakarta (UNISA) for their support, permission, and the provision of the Universitas 'Aisyiyah Yogyakarta (UNISA) Lensa Server infrastructure and access to the digital evidence used in this research. This support was very helpful in the process of data collection, digital evidence acquisition, and the implementation of digital forensic investigations in accordance with the research objectives. The authors also thank all individuals and institutions that contributed, directly or indirectly, to the completion of this research.

AUTHOR CONTRIBUTION STATEMENT

DW: Conceptualization, supervision, methodology development, validation, review and editing of the manuscript, and final approval of the manuscript. AWS: Data acquisition, investigation, data analysis, method implementation, visualization, writing the initial draft, and revising the manuscript.

AI DISCLOSURE STATEMENT

The authors used Claude (Anthropic) and ChatGPT (OpenAI) during the preparation of this manuscript to assist with restructuring, correcting grammar, enhancing clarity, and reformatting the draft to conform to the journal's intended format. Furthermore, both tools provided input on the manuscript's organization and presentation of the research findings. Following their use of these

tools, the authors reviewed, verified, and re-edited all content as needed and take full responsibility for the content, accuracy, interpretation, and conclusions presented in this publication.

REFERENCES

- [1] A. Z. Agghey, L. J. Mwinuka, S. M. Pandhare, M. A. Dida, and J. D. Ndibwile, "Detection of username enumeration attack on ssh protocol: Machine learning approach," *Symmetry (Basel)*, vol. 13, no. 11, p. 2192, Nov. 2021, doi: 10.3390/sym13112192.
- [2] 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," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jan. 2021, pp. 127–132. doi: 10.1145/3427477.3429772.
- [3] OpenSSH Project, "OpenSSH," OpenBSD Project. Accessed: Aug. 04, 2026. [Online]. Available: <https://www.openssh.com/>
- [4] G. Giridhara Maanas, P. Hegde, and G. Kannan Sadasivam, "Detection of SSH Password Guessing Attacks using Classification Algorithms," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 9, pp. 4277–4283, Sep. 2023, doi: 10.17762/ijritcc.v11i9.9883.
- [5] Y. Wu, P. M. Cao, A. Withers, Z. T. Kalbarczyk, and R. K. Iyer, "Mining Threat Intelligence from Billion-scale SSH Brute-Force Attacks," in *Proceedings 2020 Workshop on Decentralized IoT Systems and Security*, Reston, VA: Internet Society, 2020, doi: 10.14722/diss.2020.23007.
- [6] M. Davis, B. McInnes, and I. Ahmed, "Forensic investigation of instant messaging services on linux OS: Discord and Slack as case studies," *Forensic Science International: Digital Investigation*, vol. 42, Jul. 2022, doi: 10.1016/j.fsidi.2022.301401.
- [7] S. Sentanoe and H. P. Reiser, "SSHkex: Leveraging virtual machine introspection for extracting SSH keys and decrypting SSH network traffic," *Forensic Science International: Digital Investigation*, vol. 40, p. 301337, Apr. 2022, doi: 10.1016/j.fsidi.2022.301337.
- [8] K. Rasmus, T. Kokkonen, and T. Hämäläinen, "Distribution of Invalid Users on an SSH Server," 2024, pp. 139–151. doi: 10.1007/978-3-031-60227-6_12.
- [9] A. Azahari and D. Balzarotti, "On the inadequacy of open-source application logs for digital forensics," *Forensic Science International: Digital Investigation*, vol. 49, p. 301750, Jun. 2024, doi: 10.1016/J.FSIDI.2024.301750.
- [10] K. Kent, S. Chevalier, T. Grance, and H. Dang, "Special Publication 800-86 Guide to Integrating Forensic Techniques into Incident Response Recommendations of the National Institute of Standards and Technology," 2006.
- [11] R. A. Ramadhan, P. Rachmat Setiawan, and D. Hariyadi, "Digital Forensic Investigation for Non-Volatile Memory Architecture by Hybrid Evaluation Based on ISO/IEC 27037:2012 and NIST SP800-86 Framework," *IT Journal Research and Development*, pp. 162–168, Feb. 2022, doi: 10.25299/itjrd.2022.8968.
- [12] F. Al Rasyid and B. Parga Zen, "Dead Forensic Analysis Of Qutebrowser And Librewolf Browsers Using The Nist 800-86 Method," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 5, pp. 1009–1019, Oct. 2023, doi: 10.52436/1.jutif.2023.4.5.688.
- [13] P. Balqis and R. Rahman, "Analisis Keamanan Layanan SSH terhadap Brute Force Attack," *Jurnal Riset Sistem Informasi dan Teknik Informatika*, vol. 3, no. 4, 2025, doi: 10.61132/mercurius.v3i4.949.
- [14] M. F. M. Fuzi, M. F. Mazlan, M. N. F. Jamaluddin, and I. H. A. H. Halim, "Performance analysis of network intrusion detection using T-Pot honeypots," *Journal of Computing Research and Innovation*, vol. 9, no. 2, pp. 348–360, Sep. 2024, doi: 10.24191/jcrinn.v9i2.477.

- [15] C. Munteanu, Y. B. Suriyanarayanan, G. Smaragdakis, A. Feldmann, and T. Fiebig, "Attacks Come to Those Who Wait: Long-Term Observations in an SSH Honeynet," *Association for Computing Machinery (ACM)*, Oct. 2025, pp. 628–644. doi: 10.1145/3730567.3764475.
- [16] M. Rabzelj and U. Sedlar, "Beyond the Leak: Analyzing the Real-World Exploitation of Stolen Credentials Using Honey pots," *Sensors*, vol. 25, no. 12, Jun. 2025, doi: 10.3390/s25123676.
- [17] M. C. Ghanem, E. Almeida Palmieri, W. Sowinski-Mydlarz, S. Al-Sudani, and D. Dunsin, "Weaponized IoT: A Comprehensive Comparative Forensic Analysis of Hacker Raspberry Pi and PC Kali Linux Machine," *Internet of Things*, vol. 6, no. 1, Mar. 2025, doi: 10.3390/iot6010018.
- [18] A. Ali Hamza and J. s Urayh Al-Janabi, "Detecting Brute Force Attacks on SSH and FTP Protocol Using Machine Learning: A Survey," *Journal of Al-Qadisiyah for Computer Science and Mathematics*, vol. 16, no. 1, Mar. 2024, doi: 10.29304/jqcs.2024.16.11432.
- [19] J. Lee and H. Lee, "An SSH predictive model using machine learning with web proxy session logs," *Int. J. Inf. Secur.*, vol. 21, no. 2, pp. 311–322, Apr. 2022, doi: 10.1007/s10207-021-00555-6.
- [20] R. Flynn and O. Olukoya, "Using approximate matching and machine learning to uncover malicious activity in logs," *Comput. Secur.*, vol. 151, p. 104312, Apr. 2025, doi: 10.1016/J.COSE.2025.104312.
- [21] A. Firdonsyah and D. Wijayanto, "Analisis Forensik Rekayasa Dokumen PDF dengan Metode NIST," *Informatics Journal*, vol. 7, no. 2, p. 121, 2022.
- [22] Inc. MaxMind, "GeoLite2 Free Geolocation Data," MaxMind. Accessed: Aug. 02, 2026. [Online]. Available: <https://dev.maxmind.com/geoip/geolite2-free-geolocation-data>
- [23] X. Xiang, H. Liu, L. Zeng, H. Zhang, and Z. Gu, "IPAttributor: Cyber Attacker Attribution with Threat Intelligence-Enriched Intrusion Data," *Mathematics*, vol. 12, no. 9, May 2024, doi: 10.3390/math12091364.
- [24] J. Jow, J.-W. Park, S.-H. Kim, and B. Cho, "A Study on Dynamic Configuration of Attacker Blocking Time through Linux Log Analysis," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 49, no. 10, pp. 1397–1409, Oct. 2024, doi: 10.7840/kics.2024.49.10.1397.
- [25] M. B. Jiménez, D. Fernández, J. Eduardo Rivadeneira, and R. Flores-Moyano, "A Filtering Model for Evidence Gathering in an SDN-Oriented Digital Forensic and Incident Response Context," *IEEE Access*, vol. 12, pp. 75792–75808, 2024, doi: 10.1109/ACCESS.2024.3405588.
- [26] Bruce Nikkel, *Practical Linux Forensics: A Guide for Digital Investigators*. San Francisco, CA: No Starch Press, 2021.
- [27] H. Studiawan and F. Sohel, "Anomaly detection in a forensic timeline with deep autoencoders," *Journal of Information Security and Applications*, vol. 63, p. 103002, Dec. 2021, doi: 10.1016/J.JISA.2021.103002.
- [28] H. Touil, F. Hdioud, N. E. L. Akkad, and K. Satori, "The Security of SSH Protocol Public Key Sharing in the Post-Quantum Era," *International Journal of Computing*, vol. 23, no. 3, pp. 317–323, 2024, doi: 10.47839/ijc.23.3.3650.
- [29] A. D. Batistuta, A. H. Hendrawan, and Ritzkal, "Analisis Keamanan Jaringan Server Terhadap Serangan Dictionary Menggunakan Tools Fail2ban Dengan Notifikasi Telegram," *INFOTECH journal*, vol. 10, no. 1, pp. 64–73, Feb. 2024, doi: 10.31949/infotech.v10i1.8730.