Date: 9th August-2026

DEVELOPMENT OF THE CONCEPT OF LOG COLLECTION AND ANALYSIS SYSTEM AND TEST ENVIRONMENT ARCHITECTURE

Narzikulova Sevinch Aktamovna
Faculty of Computer Engineering
Student of Information Systems and Technologies

Annotation: This thesis project is devoted to the study of methods for detecting security breaches through the analysis of log files in information systems. The research examines the role of log files in identifying unauthorized access, brute-force attacks, repeated authentication failures, suspicious IP addresses, and attempts to access privileged accounts.
Keywords: log files, log analysis, information security, cybersecurity, security breach, SSH, authentication, brute-force attack, unauthorized access, failed password, accepted password, root account, Fail2ban, firewall, SSH keys, IP address, SIEM, security monitoring, Linux, Ubuntu Server.

It is important to collect, store and analyze log files in detecting security events that occur in information systems. Every user activity, login attempts, service startups, network connections, and errors are logged in. With the help of these records, it is possible to identify potential security incidents occurring within the system, determine the source of the attack, and reconstruct the sequence of events. In the practical part of this thesis, the concept of detecting security breaches by analyzing authentication logs on Linux systems was developed. Case studies were organized based on the Ubuntu Server operating system, SSH service, auth.log file, and journalctl log system. Through this approach, brute-force attack, failed authentication attempts, the possibility of unauthorized access, and time-based attack detection capabilities are created.

The concept of log collection and analysis. The main purpose of the proposed concept is to log authentication events that occur on a Linux server, analyze them, and identify suspicious security events. At the same time, the focus is focused on monitoring the login attempts via SSH service. The concept of log collection and analysis consists of the following stages:

1. organization of the test environment;
2. Running SSH on Ubuntu Server
3. attempt to connect via SSH from another Linux device;
4. Produce failed authentication events by entering the wrong password;
5. make the connection with the correct password;
6. Analysis of auth.log and journalctl entries;
7. Separation of Failed password and Accepted password records;
8. Identify suspicious IP addresses;
9. Form a timeline;
10. Determination of the type of attack and development of defensive measures.

This concept allows detecting signs of cyberattacks through log analysis in a simple laboratory environment. Specifically, the auth.log file is the primary source for detecting brute-force attacks against the SSH service.

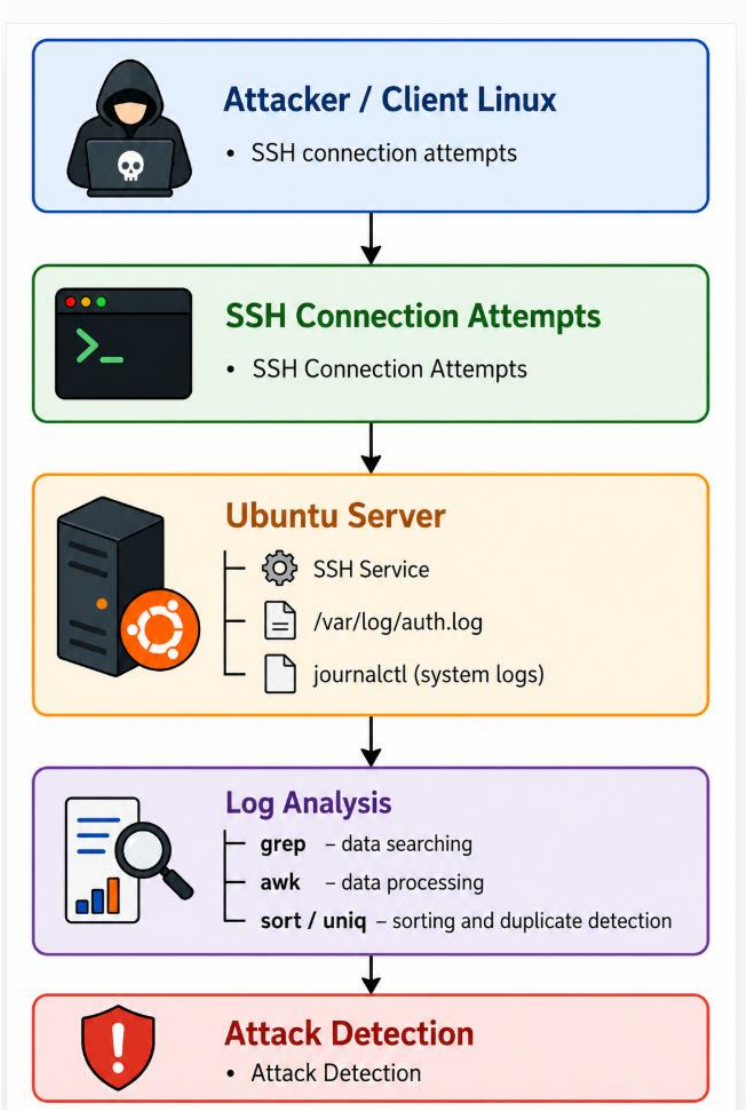

Figure 1.1. Architecture of a Test Environment for Log Analysis

Test environment architecture. A local test environment was established to conduct the case study under safe and controlled conditions. The test environment will consist of two main devices: Ubuntu Server, which acts as a server, and a client device with the Linux operating system that simulates the attack.

Ubuntu Server enables SSH and accepts connections over the network. Attempts to connect client device to server via SSH. In some attempts the password is deliberately entered incorrectly, which results in "Failed password" entries appearing in the auth.log file on the server. The correct password is then entered and a successful authentication event is generated. This means that the "Accepted password" entry is saved in the log file.

Through this architecture, the events that occur in the authentication process are observed in a practical way. The log entries generated on the server are then analyzed using command line utilities.

Components used in the test environment. Several software and hardware components are used when building a test environment. Each of these components

performs a unique function in the entire process of collecting, storing, processing, and analyzing logs. The effective operation of the system depends on the interaction of these components.

The first major component was chosen the Ubuntu Server operating system, which will also act as a server in your test environment. The SSH service is started on the server and all authentication-related events are logged in log files. Ubuntu Server is extremely usable, stable, and provides support for security tools, making it a easy platform for research.

The second component is a client device with the Linux operating system. Attempts to connect from this device to the server via SSH. With the help of the client device, successful and unsuccessful authentication cases are artificially generated, resulting in the server logs that are necessary for analysis. The SSH (Secure Shell) service is one of the central elements of your test environment. It provides the possibility of remote control and secure connection through the network. Every connection attempt to SSH, password checks, and authentication results are logged in log files. The primary source of log data is /var/log/auth.log. This file keeps track records of user login attempts, incorrect password type, authentications, and other security events. Which is why the auth.log file is one of the most important sources of security analysis. In addition, the journalctl tool maintained by the systemd journal system is also used. With this tool, service activity, system events, and SSG-related records can be viewed and filtered by time-range. This makes it possible to determine the sequence of events and form an attack timeline.

Table 2.1

Components used in the test environment

Component	**Function:**
Ubuntu Server	It was used as the primary server in the research environment. SSH has the task of running SSH, storing authentication logs, and logging of attacks.
Linux Client	Used to perform SSH connection attempts. Allows for modeling of brute-force and unauthorized entry scenarios.
SSH Service (OpenSSH)	Provides a remote secure connection between server and client devices. Authentication processes are done through this service.
/groom/people/auth.log	On a Linux system, it is a primary log file that logs all events related to authentication and authorization. This is where SSH login attempts are stored.
journalctl	Systemd is a tool for viewing and analyzing log entries. Provides detailed information about services activities and system events.
Grep	It is a command-line utility used to filter and search for necessary entries from log files. Allows for quick detection of specific events.

AWK	Logs are used to extract and process information, such as time, IP address, username, etc.
sort / uniq	Used to sort through log data and group recurring events. Especially useful when detecting a large number of attempts from the same IP address.

Command-line utilities like grep, awk, sort, and uniq are used when analyzing logs. While grep allows for a quick search and filtering of the necessary records, AWK serves to extract important fields from log rows. And the sort and uniq commands allow grouping IP addresses or events and counting their number of times. Together, the components used in the test environment form the complete lab infrastructure needed to collect, process authentication logs, and detect security leaks.

When these components work together, a sufficient lab environment is formed to analyze authentication logs and detect security breaches.

Log sources. The test environment uses auth.log and journalctl entries as the primary log sources.

auth.log. The auth.log file logs authentication-related events on Ubuntu Server. Login attempts via SSH, incorrect password entry, successful authentication, and sudo actions are stored in this file. For example:

Failed password for server from 192.168.0.108

Accepted password for server from 192.168.0.108

With the help of these records, it is possible to determine which IP address the login attempts were made and whether those attempts succeeded or failed.

Journalctl. journalctl is used to analyze systemd journal entries. With it, SSH service-specific events can be filtered by time, isolated records from specific time intervals, and generate an attack timeline. Example:

sudo journalctl -u ssh

or:

sudo journalctl -u ssh --since "2025-09-18 01:00" --until "2025-09-18 04:00"

By using these commands, all SSH service-related events are addressed, or at some time interval.

A Mechanism for Analyzing Logs. The logging process prioritizes authentication-related records that are isolated. To do this, the command grep is used.

sudo grep "Failed password" /var/log/auth.log

This command shows all failed authentication attempts. In the next step, IP addresses are isolated and their duplicate count is calculated.

grep "Failed password" /var/log/auth.log |grep -oE '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+' | black | uniq -c

This command groups cases of incorrect password entering by IP address and shows how many attempts each IP address made. The following command is used to analyze events over time:

sudo grep "Failed password" /var/log/auth.log | awk '{print $1,$2,$3}'

This command extracts time-sensitive information from log records and is used to generate an attack timeline.

Criteria for detecting attacks. Several criteria are used to detect security breaches in test environments. These criteria are based on recurrence in log entries, time intervals, and IP address.

Based on these criteria, log records are analyzed and security incidents are assessed. The proposed test environment offers several advantages:

1. simple and understandable structure;
2. the ability to work on the basis of real Linux logs;
3. hands-on analysis of SSH authentication events;
4. the ability to draw up an attack timeline;
5. perform pre-analysis even without an additional SIEM system;
6. the option to expand later with Fail2ban, Wazuh, or ELK Stack.

Table 1.2

Criteria for logging-based attack detection

Criterion	**Description**	**Possible attack type**
Many attempts in a short time	Logging of many *Failed Password* or *Authentication Failure* records in short intervals from one IP address.	Brute-force hujumi
Try rooting account	Detection of attempts to log into root or administrator accounts *via SSH or other authentication services*.	Attack on admin account
Failed Login'dan keyin Successful Login	Record of successful login after multiple failed authentication attempts.	Unauthorized access or account compromise
Constant time-interval attempts	Observation of repeated login attempts every 5–10 minutes or at a certain interval.	Time-based brute-force hujumi
Logging in from an unknown IP address	The system detects connections from a previously untracked or untrusted geographic area.	Suspicious Activity or Unauthorized Access

This approach is practically convenient for a bachelor's degree project and allows you to demonstrate the process of detecting security breaches by analyzing log files in a simple and understandable way. In this paragraph, the concept of a log collection and analysis system and the architecture of the test environment are developed. The test environment consisted of Ubuntu Server, SSH service, auth.log, journalctl, and a Linux client device. In this environment, authentication events are thrown and log entries are analyzed using GREP, AWK, SORT, and UNIQ tools. The proposed architecture would provide an adequate framework for detecting brute-force, unauthorized access, and time-based attack marks. The next paragraph provides a practical overview of the process for detecting security breaches based on Linux authentication logs in this test environment.

CONCLUSION

In this thesis project, the topic "Detection of security breaches by analyzing log files" was studied. In the course of the work, the concept of log files, their types, the main sources of logs in Windows and Linux operating systems, and the importance of logs in information security were analyzed. Logs are an important source of information that records events that occur in the system, and it was substantiated to substantiate user actions, authentication processes, system failures, and security breaches.

In the theoretical part, methods to detect cyberattacks based on security incidents and logs are examined. In particular, the log records of attacks such as brute-force, unauthorized access, privilege escalation, and time-based brute-force were highlighted in the logs. Also, the functions of log monitoring and analysis tools such as auth.log, syslog, journalctl, Event Viewer, SIEM systems, grep, awk, tail are studied. Possibility to identify security insiders via log collection, filtering, timesheet generation, and event correlation has been demonstrated.

The results of the diploma project showed that the systematic collection, storage and analysis of log files is one of the effective ways to detect security breaches in information systems. Brute-force and unauthorized login attempts can be detected at an early stage, especially by analyzing SSH authentication events based on auth.log and journalctl records on Linux systems. The proposed approach can be used as a practical solution for organizing security monitoring, checking for insiders, and strengthening system protection in small and medium-sized information systems.

REFERENCES:

1. Theis, M. C., Trzeciak, R. F., Costa, D., Moore, A. P., Miller, S., Cassidy, T., Claycomb, W. R. *Common Sense Guide to Mitigating Insider Threats, Sixth Edition*. Carnegie Mellon University, Software Engineering Institute, 2019.
2. IBM Security, Ponemon Institute. *Cost of a Data Breach Report 2025*. IBM, 2025.
3. Ponemon Institute / DTEX Systems. *2025 Cost of Insider Risks Report* bo‘yicha tahliliy ma’lumotlar.
4. Kent, K., Souppaya, M. NIST SP 800-92: Guide to Computer Security Log Management. National Institute of Standards and Technology, 2006.
5. NIST CSRC. Security Information and Event Management - Glossary. National Institute of Standards and Technology.
6. IETF. RFC 5424: The Syslog Protocol. Internet Engineering Task Force.
7. Microsoft Learn. Sysmon - Sysinternals. Microsoft documentation
8. NIST. Cybersecurity Framework 2.0, 2024.
9. NIST. SP 800-207: Zero Trust Architecture, 2020.
10. CISA. Zero Trust Maturity Model Version 2.0, 2023.
11. CISA. More than a Password: Multifactor Authentication.
12. MITRE. MITRE ATT&CK: Lateral Movement, TA0008.
13. Labor Code of the Republic of Uzbekistan. RU-798, 28.10.2022.

14. Law of the Republic of Uzbekistan "On Labor Protection". RU-410, September 22, 2016.
15. Law of the Republic of Uzbekistan "On Fire Safety". RU-226 No. 30.09.2009.
16. ISO 45001:2018. Occupational health and safety management systems — Requirements with guidance for use. International Organization for Standardization, 2018.
17. National Institute for Occupational Safety and Health (NIOSH). Hierarchy of Controls. Centers for Disease Control and Prevention.
18. Occupational Safety and Health Administration (OSHA). Computer Workstations eTool. United States Department of Labor.