Linux Log Monitoring and Analysis

Introduction

This project is a simple yet effective log monitoring and analysis system designed to track SSH login attempts, sudo access, and command history on a Linux system. The system consists of a Bash script to extract relevant logs and a Python script to parse and analyze the data.

The purpose of this project is to provide visibility into system activity, helping detect unauthorized access attempts and suspicious behavior. Given the importance of cybersecurity, monitoring system logs is a crucial aspect of maintaining system integrity and security.

Project Structure

The project consists of two main scripts:

1. **Bash Script (monitor_logs.sh)**
    - Extracts the last ten SSH login attempts from /var/log/auth.log
    - Extracts the last ten sudo access attempts from /var/log/auth.log
    - Extracts the last ten executed system commands from the user's Bash history
    - Outputs the extracted data to the terminal
2. **Python Script (parse_logs.py)**
    - Reads the extracted log data
    - Parses relevant fields such as timestamps, user activity, and IP addresses
    - Performs basic log analysis
    - Outputs structured data for further review or analysis

Installation and Setup

Ensure that you have a Linux system with Bash and Python installed.

For Python dependencies, install them within a virtual environment to prevent conflicts:

```
python3 -m venv venv
source venv/bin/activate
pip install pandas scikit-learn matplotlib
```

Alternatively, if you prefer to install system-wide:

```
pip install --break-system-packages pandas scikit-learn matplotlib
```

Execution

The scripts can be executed separately or together depending on the use case. To run them manually:

```
chmod +x monitor_logs.sh
./monitor_logs.sh
python3 parse_logs.py
```

If you want the Bash script to automatically call the Python script after extracting logs, modify monitor_logs.sh to include:

```
python3 parse_logs.py
```

This ensures a seamless workflow where log extraction and analysis occur in sequence.

Why This Matters

System logs provide a detailed record of activities, making them essential for security monitoring and forensic analysis.

- **SSH Logs:** Help detect unauthorized access attempts and brute-force attacks.
- **Sudo Logs:** Provide insight into administrative actions, ensuring accountability.
- **Command History:** Reveals user behavior, aiding in forensic investigations.

By implementing this system, security professionals and system administrators can enhance their ability to track and respond to suspicious activities.

Troubleshooting

Externally Managed Environment Error

If you encounter:

```
error: externally-managed-environment
```

This occurs due to Ubuntu's package management restrictions (The OS I use. Other OS may be different). Either install dependencies using apt or create a virtual environment as shown earlier.

Empty Output

If the Python script produces an empty dataframe:

Ensure the Bash script is extracting logs properly by running:

```
 cat /var/log/auth.log | grep sshd
```

- Check if the log file path is correct.
- Ensure you have sufficient permissions to read log files (sudo may be required).

Permission Denied

If executing monitor_logs.sh results in:

```
permission denied
```

Grant execute permissions with:

```
chmod +x monitor_logs.sh
```

Script Does Not Detect SSH or Sudo Logs

If expected log entries do not appear:

- Ensure your system is configured to log SSH and sudo activity.
- Check if logs are stored in /var/log/auth.log or another location.
- Try modifying the grep commands in monitor_logs.sh to refine search criteria.

Conclusion

This log monitoring system serves as an essential tool for security-conscious users who want to track authentication attempts and administrative activity. The modular approach ensures flexibility, allowing users to modify or extend functionality as needed. By combining Bash scripting with Python's analytical capabilities, this project provides an effective solution for monitoring Linux system activity.