

ABC Inc. Firewall Log Analyzer

Submitted by
Kasun Priyashan

Table of Contents

Introduction	3
Methodology for create Log analyzer	4
Design.....	6
Implementation	7
Results and discussion	9

Introduction

What is firewall and firewall logs?

A firewall is an important part of network security because it keeps a trusted internal network separate from a less trustworthy external network, like the internet. The main job of this software is to keep an eye on and manage all incoming and outgoing network data based on the security rules and guidelines set by the company.

some key aspects of firewalls:

- Packet filtering: Firewalls look at network data and decide whether to let them through or not based on rules that have already been set. Source IP address, target IP address, port numbers, and protocols are just some of the things that can be used to make these rules.
- Stateful Inspection: Stateful inspection is a way for modern firewalls to keep track of what state live links are in. This lets them make better choices by letting them know more about the traffic, like if it's part of an established link.
- Some more advanced firewalls can look at data at the application layer, which is Layer 7 of the OSI model. This gives them more precise control over network data by letting them find and block specific apps or services.

Now let's talk about logs from the firewall:

Firewall logs are records that a firewall keeps of what it does and how it interacts with network data. For a number of reasons, these logs are very useful for network managers and security experts:

Monitoring security: Firewall logs show in great detail all the data that comes in and goes out. These logs can help security teams find and look into possible security incidents or efforts to get in without permission

How to analyze firewall logs?

The analysis of firewall logs is an essential responsibility for network administrators and security professionals. In order to conduct a comprehensive analysis of firewall logs, it is imperative to initiate the process by gathering and consolidating the logs in a safe and centralized repository. This task involves comprehending the log formats employed by the firewall, filtering and standardizing the data, and subsequently detecting patterns and anomalies within the log entries. This approach involves establishing correlations between events derived from many sources, utilizing threat intelligence feeds to identify potential risks, and establishing benchmarks for typical network behavior. Establish alerts and notifications in accordance with pre-established guidelines in order to promptly address security incidents. In the event of incidents, it is imperative to conduct a comprehensive investigation, meticulously record the discovered evidence, and adhere to the incident response strategy established by your organization. It is imperative to consistently refine analysis procedures, remain updated on emerging threats, and adapt firewall rules and policies in order to bolster network security. The analysis of firewall logs constitutes a continuous and proactive endeavor aimed at safeguarding one's network against potential attacks and weaknesses.

Methodology for create Log analyzer

In order to write a script for ABC Inc. that does simple analysis of firewall logs, we need to take into consideration the following steps:

- The format of the log needs to be understood. The firewall logs generated by iptables on Linux and those generated by AWS security groups have different forms. To extract useful data, it is essential to have a working knowledge of these formats.
- The process of extracting the pertinent information from these logs and preparing it in preparation for analysis is referred to as data extraction and preprocessing.
- Analysis and Pattern Detection: Examining these logs in order to locate potential dangers and irregularities, as well as to gain an understanding of traffic patterns.

The provided log file appears to be structured with a clear header line, followed by individual log entries. Each entry contains several fields, as outlined in the header:

- Date and Time: The date and time of the logged event.
- Action: The action taken by the firewall (e.g., ALLOW, BLOCK).
- Protocol: The network protocol used (e.g., TCP, UDP, ICMP).
- Src IP: Source IP address.
- Dst IP: Destination IP address.
- Src Port: Source port number.
- Dst Port: Destination port number.
- Size: Size of the packet.
- TCP Flags: TCP flags, if applicable.
- Info: Additional information about the event (e.g., type of request, nature of traffic).

Based on this structure, the script can process these logs to:

Categorize Traffic: Classify traffic based on action (ALLOW, BLOCK), protocol, and other attributes.

Identify Anomalies: Look for unusual patterns, such as repeated blocks from the same source IP or unusual traffic on sensitive ports.

Traffic Analysis: Determine the most common types of traffic, frequent sources of blocked traffic, etc.

Security Incidents Identification: Flag potential security incidents like repeated SSH or SQL Server access attempts from external sources.

Design

I created a simple python script for analyze firewall logs using flask environment .for creation I used following steps.

1. Importing Libraries:

The code starts by importing necessary libraries, including Flask, os, pandas (for data manipulation), and re (for regular expressions).

2. Configuration:

It defines some configuration settings for the Flask application, including the location where uploaded log files will be stored temporarily (UPLOAD_FOLDER) and a set of allowed file extensions (ALLOWED_EXTENSIONS).

3. Creating a Flask App:

The Flask application is created using Flask(__name__).

4. Parsing Log Lines:

The code defines a function parse_log_line(line) to parse a single line of a log file. It uses regular expressions to split each line into fields and extracts relevant information such as date, time, action, protocol, source and destination IP addresses, ports, size, TCP flags, and additional information.

5. Analyzing Uploaded Log File:

Another function analyze_uploaded_log(file) is defined to analyze the uploaded log file. It reads the log file, skips the header line, and iterates through the lines to parse and extract data.

The extracted log data is then converted into a Pandas DataFrame for further analysis.

Basic log analysis is performed, including counting actions, identifying top blocked ports, and finding suspicious source IPs.

Several custom functions are defined within this function to identify potential threats such as DDoS attempts, brute force attacks, port scanning, horizontal scanning, uncommon port traffic, repeated actions, unauthorized SSH access attempts, and SQL Server access attempts.

These threat identification functions return relevant information based on predefined criteria and thresholds.

6. Checking Allowed File Extension:

A function `allowed_file(filename)` checks if the uploaded file has a valid log file extension (in this case, '.log').

7. Uploading Log File:

The main route '/' is defined for the web application. It handles both GET and POST requests.

When a POST request is made (usually after uploading a log file), it checks if a file was uploaded, validates the file extension, and saves the uploaded file to the specified `UPLOAD_FOLDER`.

It then calls the `analyze_uploaded_log` function to perform log file analysis.

Finally, it renders a results template with the analyzed data and presents it to the user.

Implementation

When using the interface user can upload the log to the system and need only click analyze button

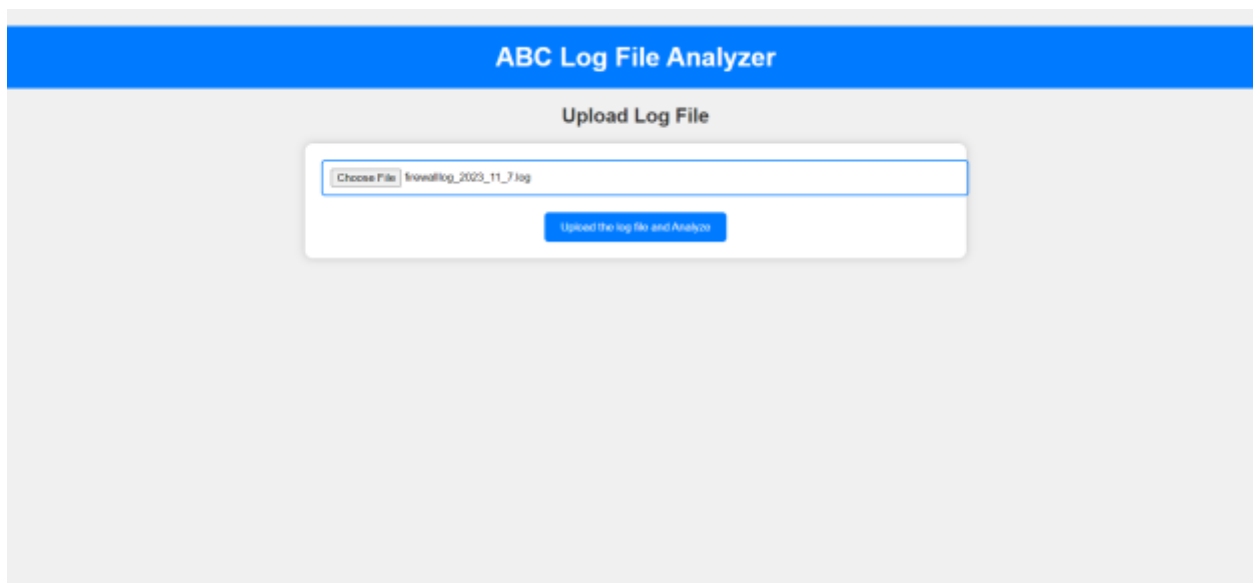
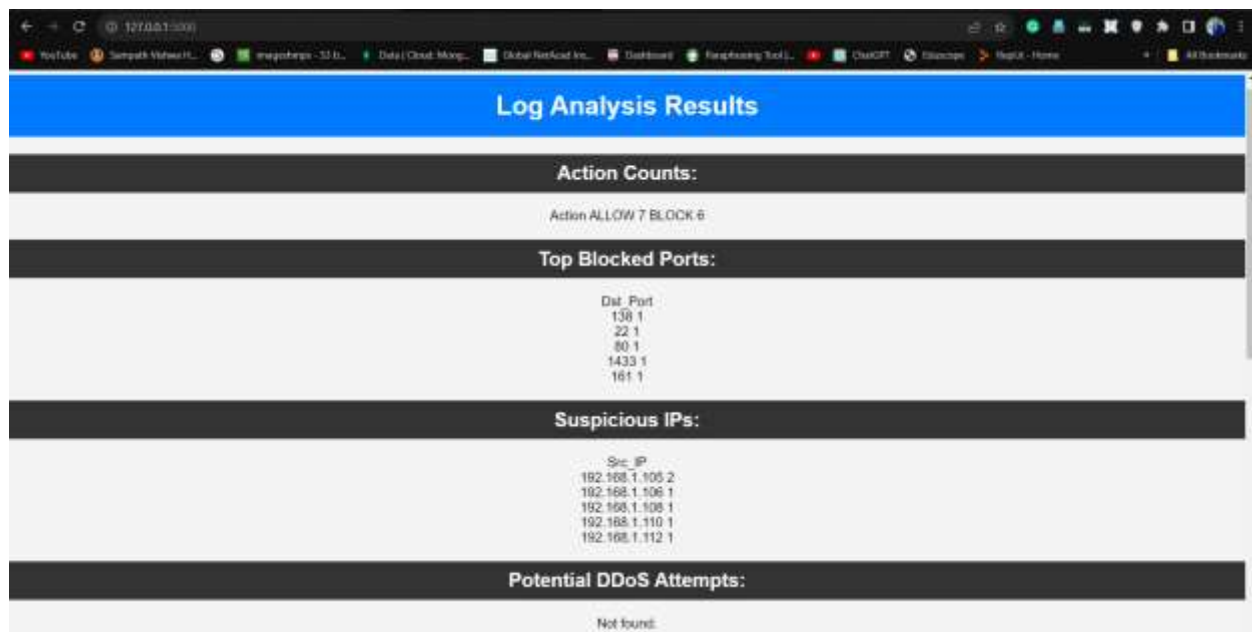


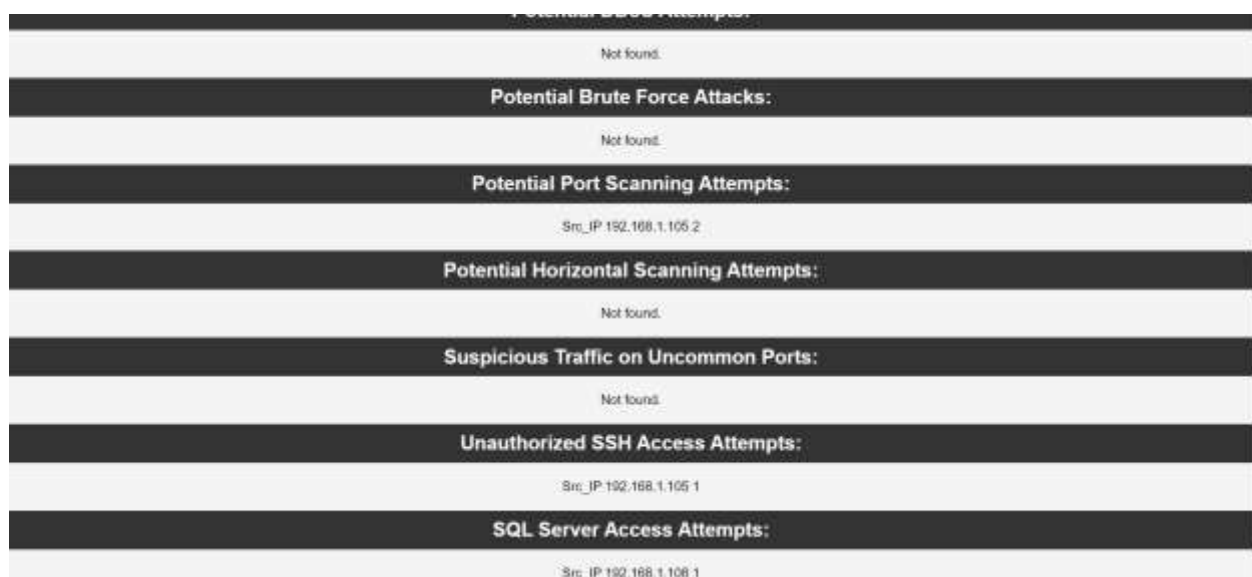
Figure 1 main interface for upload the log file

Then user can see following results



Log Analysis Results	
Action Counts:	
Action ALLOW 7 BLOCK 6	
Top Blocked Ports:	
Dst_Port	
139	1
22	1
80	1
1433	1
161	1
Suspicious IPs:	
Src_IP	
192.168.1.105	2
192.168.1.106	1
192.168.1.108	1
192.168.1.110	1
192.168.1.112	1
Potential DDoS Attempts:	
Not found.	

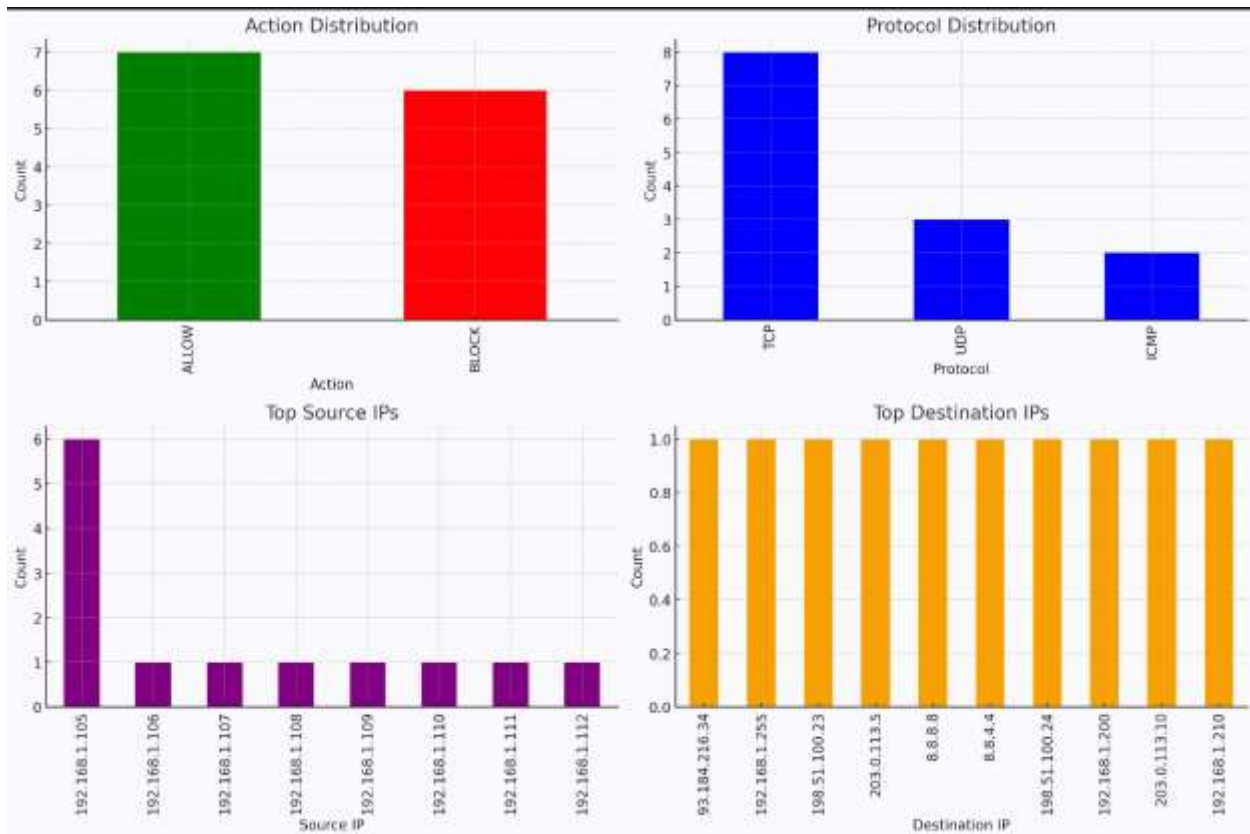
Figure 2 results after the analysis



Potential Brute Force Attacks:	
Not found.	
Potential Port Scanning Attempts:	
Src_IP	192.168.1.105 2
Potential Horizontal Scanning Attempts:	
Not found.	
Suspicious Traffic on Uncommon Ports:	
Not found.	
Unauthorized SSH Access Attempts:	
Src_IP	192.168.1.105 1
SQL Server Access Attempts:	
Src_IP	192.168.1.108 1

Figure 3 results after the analysis

Results and discussion



We'll focus on the following key areas in the analysis:

- Distribution of allowed vs. blocked traffic.
- Most common protocols used in the traffic.
- Frequent source and destination IPs for blocked traffic.
- Detection of potential security incidents (e.g., repeated access attempts).
- General traffic patterns and anomalies.

Based on the analysis of the firewall log file, here are some key insights:

Total Entries: The log contains 16 entries.

Actions: There were 7 'ALLOW' actions and 6 'BLOCK' actions.

Protocols: The top protocols used were TCP (8 times), UDP (3 times), and ICMP (2 times).

Source IPs: The IP '192.168.1.105' was the most frequent source, appearing in 6 entries. Other source IPs appeared once.

Destination IPs: Each destination IP listed appeared only once, indicating a variety of destination addresses.

Source Ports: A variety of source ports were used, each appearing only once in the log.

Destination Ports: The ports 80 and 22 were the most common destination ports, each appearing twice.

Observations :

SSH and SQL Server Attempts: The blocked attempts on SSH (port 22) and SQL Server (port 1433) could indicate targeted attempts to access sensitive services. It's advisable to further investigate these incidents for potential security threats.