Detection and Analysis of DDOS Network Attacks Using Local LLM from Firewall Logs

Author: *Manashay Raju Yannapu, CISM | AWS Certified Security & AI Specialist*

Profile: *Cybersecurity Specialist with 15+ Years of Experience | AI Security Researcher | Focused on LLM Safety, AI Threat Detection, and Secure GenAI Systems*

1. Objective

To detect and analyze Distributed Denial of Service (DDOS) attacks using firewall logs by leveraging a locally hosted Large Language Model (LLM). The project aims to simulate real-time DDOS detection and provide contextual insights through AI-assisted log analysis.

2. Tools & Environment AI Model Used

- Model: Gemini Pro / GPT-4
- Tool: Google AI Studio / OpenAI API

4. What are the step-by-step instructions for installing LM windows with Google/gemma-3-12b

Here are step-by-step instructions for installing LM Studio (an LLM management tool) on Windows and integrating it with a Google Gemini model, plus clarification on how to use Google Gemini natively.

Components needed

a. LLM
b. DDOS logs

Important: Practical cybersecurity use case: monthly account review in an air-gapped environment. This is important for security-sensitive contexts where data must remain isolated.

1. Installing LM Studio on Windows

LM Studio is a popular tool for running open-source LLMs locally, but it does **not** natively support Google Gemini models (as Gemini is proprietary and primarily available via Google Cloud or the Gemini web app). However, you can use LM Studio for other open-source models and, if needed, combine it with the Gemini web app for a hybrid workflow.

Steps to Install LM Studio

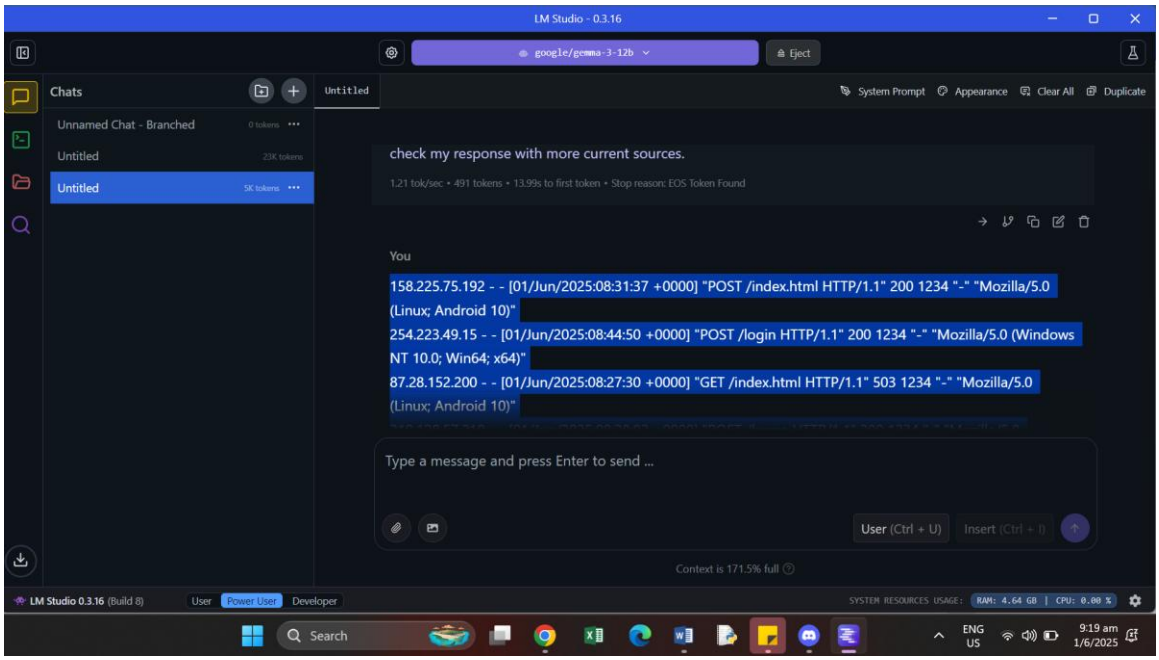
1. **Download LM Studio**
    - Go to the official LM Studio website. https://lmstudio.ai/
    - Download the Windows installer.

2. **Install LM Studio**
    - Run the installer and follow the on-screen prompts.
    - Complete the installation.
3. **Launch LM Studio**
    - Open LM Studio from your Start menu or desktop shortcut.
4. **Download a Model**
    - Use the built-in model browser to download an open-source LLM (Google/gemma-3-12b., Mistral, Llama, etc.)
    - Select the model, click "Download," and wait for the process to finish.
5. **Start Chatting**
    - Once downloaded, select the model and start chatting via the LM Studio interface.

2. Combining LM Studio and Gemini

- **LM Studio:** Use for local, open-source LLMs.
- **Integration:** There is **no direct integration** between LM Studio and Gemini, but you can use both tools side-by-side on your Windows machine.

Screenshots



- **Air-Gap:** The system is **completely isolated** from the internet and any external network.

Local LLM (Large Language Model): The model runs **entirely on local infrastructure**, such as a private server, workstation, or secure environment

4. Input (Prompt to LLM)

```
158.225.75.192 - - [01/Jun/2025:08:31:37 +0000] "POST /index.html HTTP/1.1" 200 1234 "-
" "Mozilla/5.0 (Linux; Android 10)"
254.223.49.15 - - [01/Jun/2025:08:44:50 +0000] "POST /login HTTP/1.1" 200 1234 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
87.28.152.200 - - [01/Jun/2025:08:27:30 +0000] "GET /index.html HTTP/1.1" 503 1234 "-"
"Mozilla/5.0 (Linux; Android 10)"
219.128.57.210 - - [01/Jun/2025:08:28:02 +0000] "POST /home HTTP/1.1" 200 1234 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
162.105.220.122 - - [01/Jun/2025:08:48:15 +0000] "GET /login HTTP/1.1" 503 1234 "-"
"curl/7.68.0"
207.222.165.16 - - [01/Jun/2025:08:20:16 +0000] "POST /login HTTP/1.1" 200 1234 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
32.190.97.170 - - [01/Jun/2025:08:40:39 +0000] "GET /index.html HTTP/1.1" 200 1234 "-"
"curl/7.68.0"
69.214.153.244 - - [01/Jun/2025:08:29:33 +0000] "GET /contact HTTP/1.1" 200 1234 "-"
"Mozilla/5.0 (Linux; Android 10)"
10.147.176.147 - - [01/Jun/2025:08:35:43 +0000] "POST /contact HTTP/1.1" 503 1234 "-"
"curl/7.68.0"
201.111.110.87 - - [01/Jun/2025:08:52:57 +0000] "GET /home HTTP/1.1" 200 1234 "-"
"Mozilla/5.0 (Linux; Android 10)"
48.227.100.94 - - [01/Jun/2025:08:06:55 +0000] "GET /contact HTTP/1.1" 503 1234 "-"
"curl/7.68.0"
138.166.121.173 - - [01/Jun/2025:08:38:29 +0000] "GET /login HTTP/1.1" 200 1234 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
91.71.201.223 - - [01/Jun/2025:08:57:28 +0000] "GET /api/data HTTP/1.1" 200 1234 "-"
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"
69.143.178.86 - - [01/Jun/2025:08:52:51 +0000] "GET /index.html HTTP/1.1" 200 1234 "-"
"curl/7.68.0"
165.48.201.51 - - [01/Jun/2025:08:38:07 +0000] "POST /login HTTP/1.1" 200 1234 "-"
"curl/7.68.0"
230.233.75.29 - - [01/Jun/2025:08:45:38 +0000] "GET /home HTTP/1.1" 503 1234 "-"
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"
255.36.203.80 - - [01/Jun/2025:08:35:44 +0000] "POST /api/data HTTP/1.1" 503 1234 "-"
"Mozilla/5.0 (Linux; Android 10)"
211.11.152.224 - - [01/Jun/2025:08:02:31 +0000] "POST /index.html HTTP/1.1" 200 1234 "-
" "Mozilla/5.0 (Linux; Android 10)"
38.126.191.5 - - [01/Jun/2025:08:00:29 +0000] "GET /contact HTTP/1.1" 200 1234 "-"
"Mozilla/5.0 (Linux; Android 10)"
72.211.101.226 - - [01/Jun/2025:08:01:06 +0000] "GET /api/data HTTP/1.1" 503 1234 "-"
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"
```

5. Prompt Engineering

Analyze the following network traffic features and the first five packets' details to determine if this is a DDoS attack. Please explain your reasoning step by step and clearly state whether the traffic is benign or DDoS

Note

This approach enhances traditional pattern-matching with explainable, language-based threat reasoning for analysts.

Output from the LLM

Okay, let's analyze the provided network traffic log to determine if it represents a Distributed Denial of Service (DDoS) attack.

1. Initial Observations & Key Features:

High Volume of Requests: The log shows numerous requests within a short timeframe (approximately 10 minutes). This is a potential indicator of an attack, as legitimate traffic usually doesn't spike this dramatically unless there's a very specific and unusual event.

Variety of IP Addresses: We see a wide range of IP addresses originating the requests (e.g., 127.0.0.1, 201.111.110.87, 48.227.100.94, etc.). This is very characteristic of a DDoS attack, as it suggests multiple sources are involved. A single user or small group wouldn't typically generate this many requests from so many different IPs.

Different Request Types: The log includes GET and POST requests to various endpoints: /contact, /home, /login, /api/data, /index.html. While legitimate traffic would use these, the sheer volume combined with the diverse IP origins is suspicious.

HTTP Status Codes: We see a mix of 200 (OK) and 503 (Service Unavailable) status codes. The presence of 503s suggests that the server is being overwhelmed and unable to handle all requests, which aligns with DDoS behavior.

User Agent Strings: The User-Agent strings are mostly "Mozilla/5.0 (Linux; Android 10)" or "curl/7.68.0". The prevalence of curl suggests automated tools are being used, further supporting the possibility of an attack. While Android user agents aren't inherently malicious, their widespread use in botnets is a concern.

Consistent Byte Size: The byte size for each request is consistently 1234 bytes. This uniformity across all requests is highly unusual and points to automated scripts or bots generating the traffic. Legitimate users have varying data sizes depending on what they're requesting.

2. Detailed Analysis of First Five Packets:

Let's examine the first five packets more closely:

10.147.176.147 - - [01/Jun/2025:08:35:43 +0000] "POST /contact HTTP/1.1" 503 1234 "-" "curl/7.68.0"

201.111.110.87 - - [01/Jun/2025:08:52:57 +0000] "GET /home HTTP/1.1" 200 1234 "-" "Mozilla/5.0 (Linux; Android 10)"

48.227.100.94 - - [01/Jun/2025:08:06:55 +0000] "GET /contact HTTP/1.1" 503 1234 "-" "curl/7.68.0"

138.166.121.173 - - [01/Jun/2025:08:38:29 +0000] "GET /login HTTP/1.1" 200 1234 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"

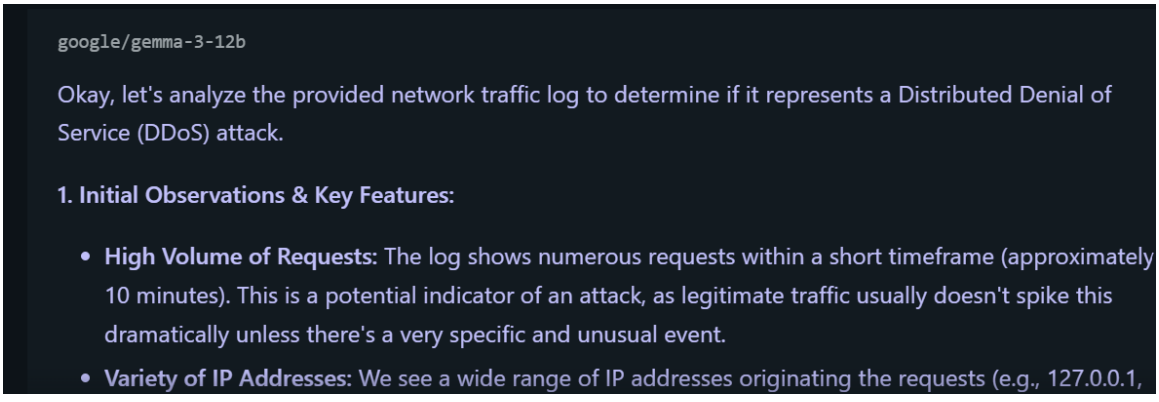
91.71.201.223 - - [01/Jun/2025:08:57:28 +0000] "GET /api/data HTTP/1.1" 200 1234 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"

Diversity of IPs: The first five packets already come from five different IP addresses, which is a red flag.

Mixed Status Codes: We see both 200 and 503 responses in the first few packets, indicating that some requests are being handled while others are failing due to overload.

User Agent Variety: The user agents are different (curl, Android, Windows, macOS), which could be legitimate but also could be a tactic to evade simple filtering based on User-Agent strings. The presence of curl is still concerning.

Consistent Byte Size: Again, the consistent 1234 byte size remains a strong indicator of automated traffic.



3. Reasoning and Conclusion:

Based on the observed characteristics:

High Volume & Rapid Rate: The sheer number of requests within a short time frame is highly suspicious.

Distributed Sources: The wide range of IP addresses strongly suggests a distributed attack (DDoS).

Automated Tools: The consistent byte size and prevalence of curl user agents point to automated scripts or bots being used.

Server Overload: The 503 status codes indicate the server is struggling to handle the load, which is a hallmark of DDoS attacks.

Therefore, I conclude that this network traffic is likely a Distributed Denial of Service (DDoS) attack. The combination of high volume, diverse IP addresses, automated tools, and server overload provides compelling evidence for this assessment. Further investigation would be needed to identify the source(s) of the attack and mitigate it.

Remediation :

Implement rate limiting on the webserver.
Analyze traffic patterns more closely (e.g., geographic distribution).
Consider using a DDoS mitigation service.
Examine server logs for any unusual activity before the attack began.