IETF Hackathon

Knowledge Graph–Driven Threat Intelligence Analysis and Network Configuration Generation

IETF 123
19–20 July 2025
Madrid, Spain

Motivating Example

Threat Intelligence of HTTP 2.0 Rapid Reset DDoS Attack

Overview of HTTP/2 Rapid Reset Attack Mechanism:

- The attacker initiates a large number of requests using HTTP/2 , then immediately send RST_STREAM frames to terminate the connections.
- The server expends significant resources to process these requests and cancellations.

How to defend?

- Server operators should identify abuse based on connection patterns—e.g., a **TCP connection with >100 requests and >50% cancellations may be abusive**. Responses can vary from sending a GOAWAY frame to terminating the connection.

1. Which devices/software should I use for defense?
Router/WAF/Firewall/Nginx/Snort？
2. How should I configure these devices/software?

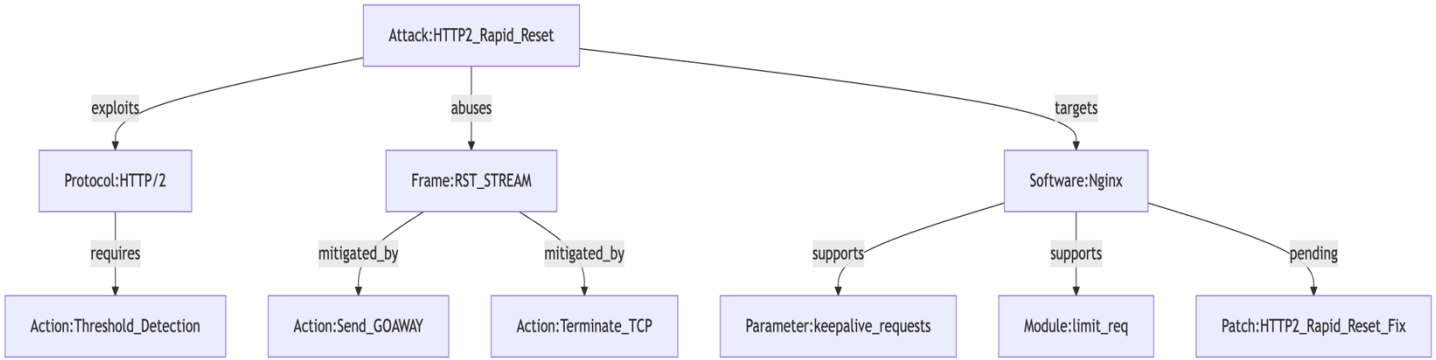
Hackathon Plan

- <What problem were you working on?>
  - Operators need to learn the intelligence and manually type commands to achieve specific intents
    - A **learning curve** to understand both the intelligence and network device configuration
    - **Expert experience** in selecting the most appropriate solution
    - **Risks of typos or fat-finger** during practical operation

Hackathon Plan

We use **Large Language Model (LLM)** to construct an intelligence **knowledge graph**,

and use **Graph Retrieval-Augmented Generation** technique to generate the configuration



For example, HTTP 2.0 rapid reset <--exploits--> TCP connection >100 <--target--> Nginx <--supports--> keepalive_requests 100;

What got done

- <What you achieved? (key results)>
  - Delivered an end-to-end LLM+KG demo that parses DDoS intel and generate config commands.
  - GitHub Link
    - https://github.com/nobrowning/KG-Intent_Conf_Gen
    - The detailed solution workflow can be found here

What got done

Intel

Overview of HTTP/2 Rapid Reset Attack Mechanism:
The attacker initiates a large number of requests using HTTP/2, then immediately send RST_STREAM frames to terminate the connections.
The server expends significant resources to process these requests and cancellations.
How to defend?
Server operators should identify abuse based on connection patterns—e.g., a TCP connection with >100 requests and >50% cancellations may be abusive.
Responses can vary from sending a GOAWAY frame to terminating the connection.

Start Run

Input

nginx

```nginx
http {
    # Define rate limiting zone
    limit_req_zone $binary_remote_addr zone=req_limit:10m rate=30r/s;

    server {
        listen 443 ssl http2;

        # Limit requests per connection
        keepalive_requests 100;

        # Apply rate limiting to all requests
        limit_req zone=req_limit burst=50 nodelay;

        # Other server configurations...
    }
}
```

Output

What we learned

- <Lessons learned from this hackathon>
  - <**New work** to take to WG?>
    - **Frameworks of LLM and KG in network management**
    - https://datatracker.ietf.org/doc/draft-cui-nmrg-llm-nm/
    - https://datatracker.ietf.org/doc/draft-cui-nmrg-llm-benchmark/
    - https://datatracker.ietf.org/doc/draft-cui-ai-agent-task/

Wrap Up

Team members:

Yong Cui (cuiyong@tsinghua.edu.cn)
Mingzhe Xing (xingmz@zgclab.edu.cn)
Chenguang Du (ducg@zgclab.edu.cn)

First Timers@IETF Hackathon

- Mingzhe Xing
- Chenguang Du

We thanks the comments from hackathon!