A Study on LR-DDoS and the Enhancement of General-Purpose Python Socket Module to Provide Defense Mechanisms for Multi-Thread Servers

By 106321035 周以恆

OUTLINE

- LR-DDoS
- Defense Mechanism on Web Severs
- Original Python Socket Server
- Proposed Solution

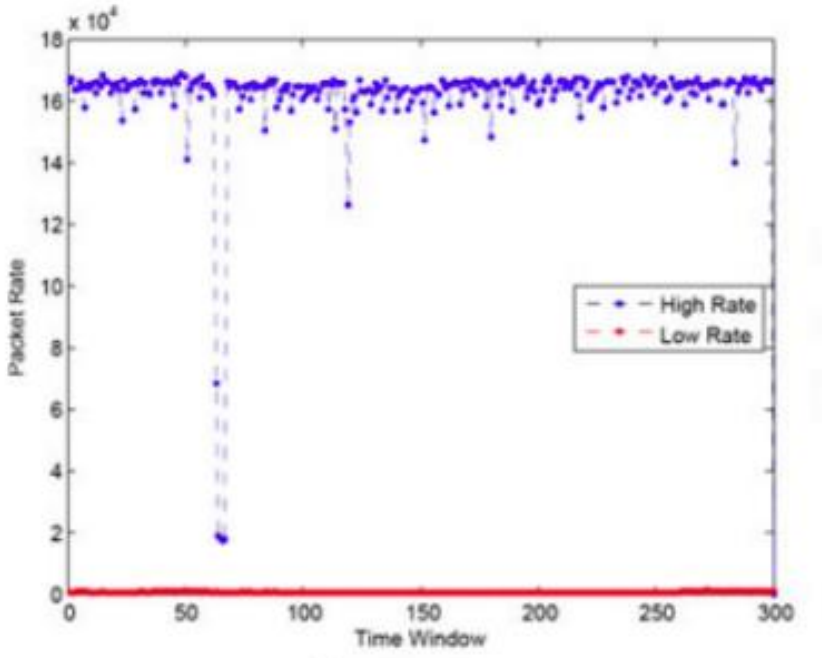
High-Rate vs. Low-Rate

High-Rate

1. Volume Based (Bandwidth)
2. Packet Burst
3. Network Congestion

Low-Rate

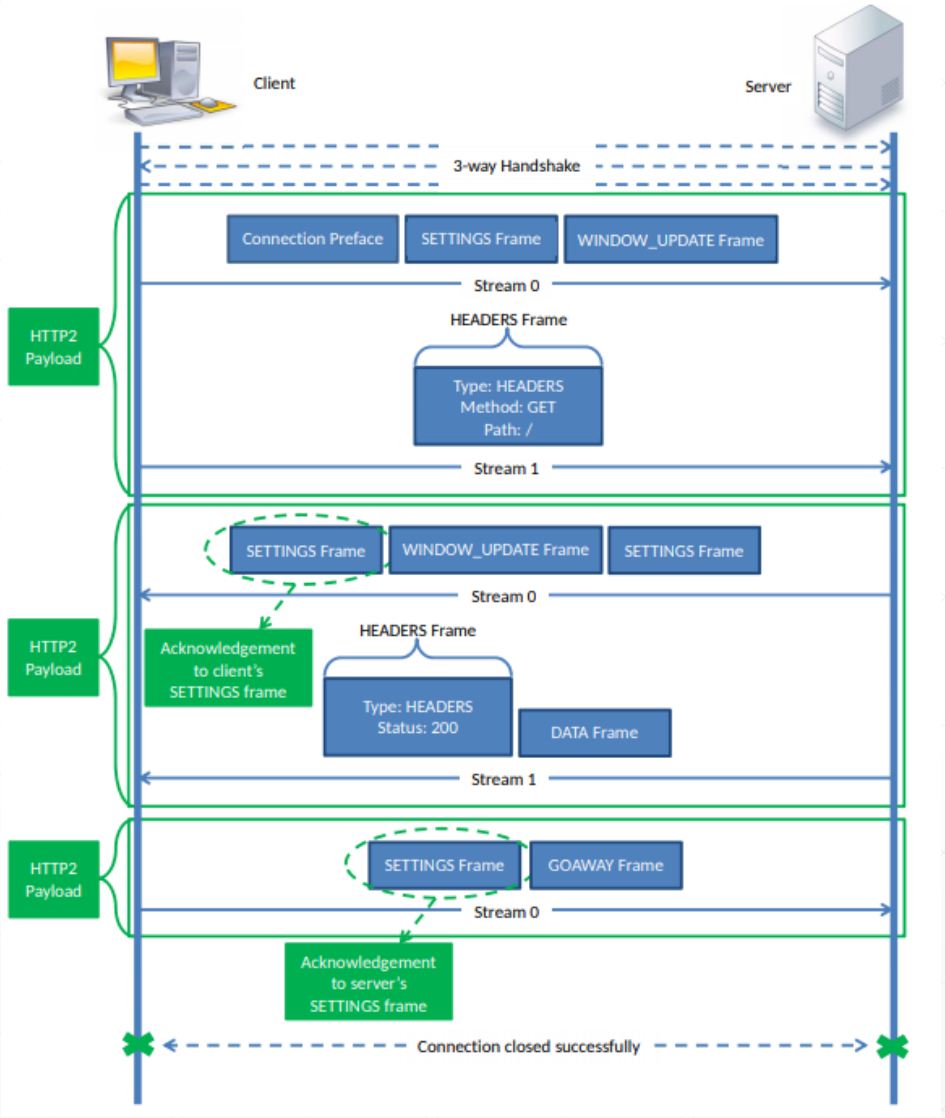
1. Low Bandwidth Needed
2. Hard to Detect (Behave like a Normal User)



(a) Packet rate

[1]

LR-DDoS on HTTP (LowRate-DDoS)



HTTP/2 Payload [2]

LR-DDoS (Example 1)

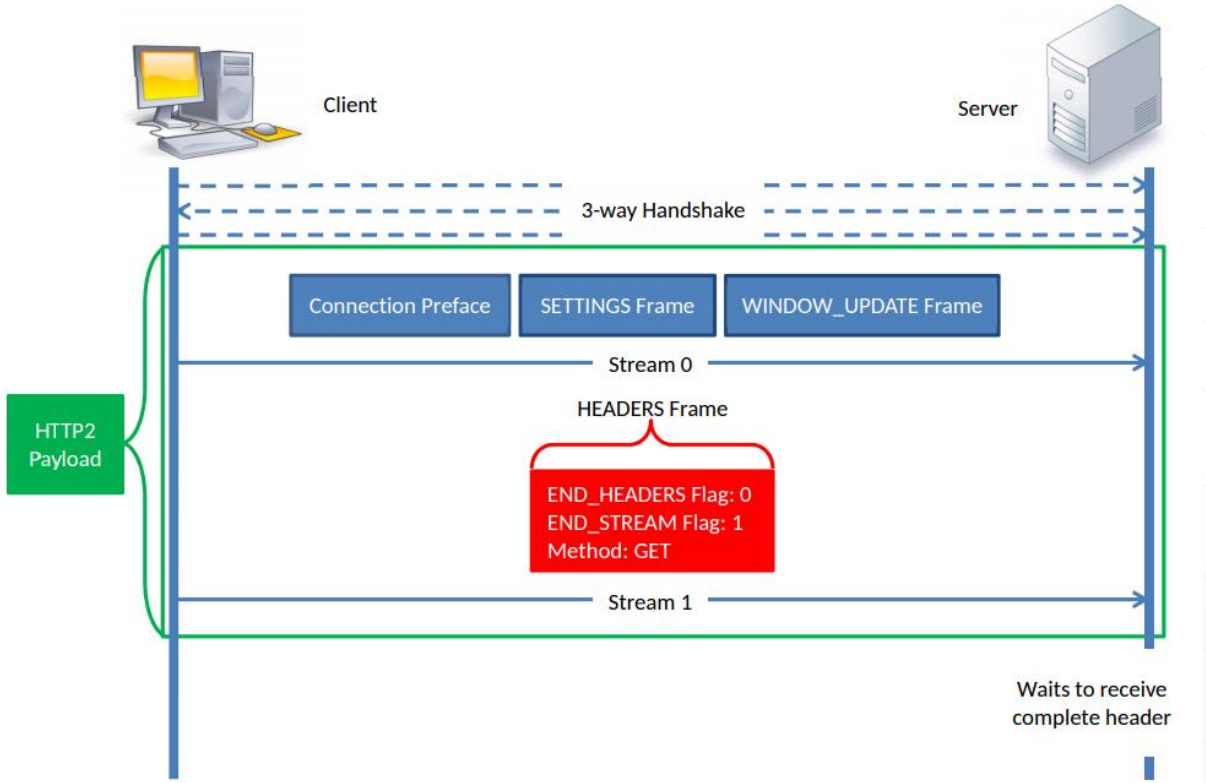
Slow Header

Malicious client send HEADER frame with these flags.

END_HEADERS : reset(0)

END_STREAM : set(1)

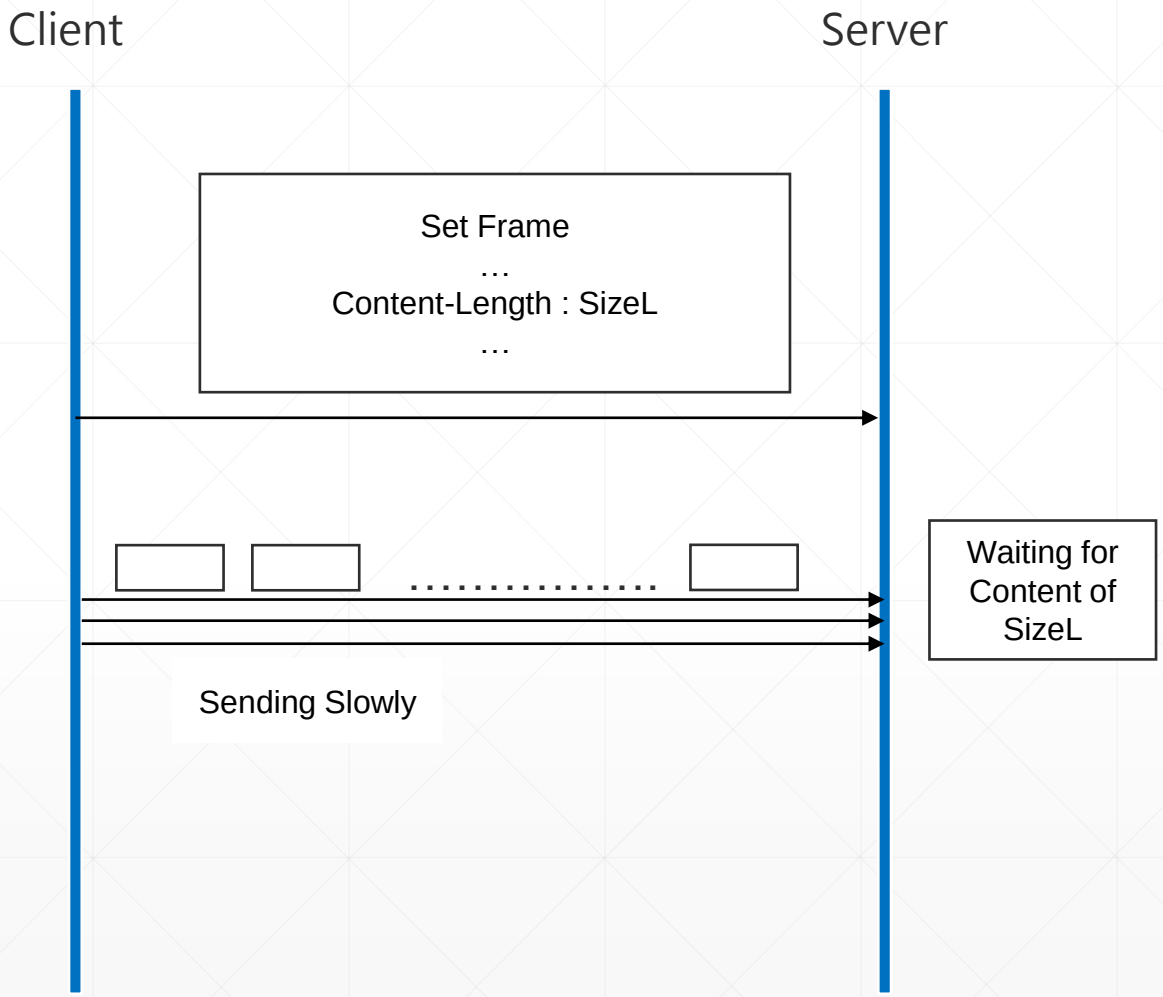
Implies there are more frames coming



LR-DDoS (Example 2)

Slow Body

Send a respectively long request and send content slowly to let Server waiting for complete.



Defense Mechanism
Apache II

Apache有多個Module可供開啟，每個都只需下載並啟用即可。

mod_reqtimeout：限制Headers與Body的傳送時限。

mod_qos：當Server較繁忙時，會關閉HTTP KeepAlive。
此外也會限制單個IP的同時連線數和最低的資料傳送速率。

mod_antiloris：限制單個IP最多同時在等候資料的連線數。

Defense Mechanism
Nginx

Nginx Server 內擁有可以自行調整的參數，用來避免此類的攻擊。

large_client_header_size : Max Header Length。
client_max_body_size : Max POST content Length。
client_header_timeout : Timeout for Header。
client_body_timeout : Timeout for Content。
send_timeout : Timeout for each frame。
CPS : Limit to connections per second. If the limit is exceeded, sleep for few seconds.。

Original Python Socket Sever

Common Problems

Too Many Open Files

Memory Waste

CPU Waste

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(('127.0.0.1', 8080))
sock.listen(1)
accept, _ = sock.accept()
```

Common Server with socket python module

Proposed Solution

LIMIT_bps : Minimum Data Rate after first byte.
LIMIT_per_source : Simultaneous open connection.
LIMIT_cps : Maximum accept connections per second.

Receiving:

Wait the first byte arrive before timeout.
After that, calculate minimum Data Rate LIMIT_bps (Bytes per second).
If Data Rate slower than LIMIT_bps, then drop connection.

Accept Connection:

If current connections from source is more than LIMIT_per_source,
accept connection for only “few” seconds then drop.

Only accept amount of connections per second.

DEMO

Reference

[1] Hoque, Nazrul & Bhattacharyya, Dhruba K & Kalita, Jugal. (2016). FFSc: A novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis. Security and Communication Networks. 9. 10.1002/sec.1460.

[2] Tripathi, Nikhil & Hubballi, Neminath. (2017). Slow Rate Denial of Service Attacks Against HTTP/2 and Detection. Computers & Security. 72. 10.1016/j.cose.2017.09.009.