Slow Client

https://campus.barracuda.com/doc/98225913/

In a slow client attack, an attacker deliberately sends multiple partial HTTP requests to the server to carry out an HTTP DoS attack on the server. The client attempts to slow the request or response so much that it holds connections and memory resources open on the server for a long time, but without triggering session time-outs. Common ways to carry out this attack include:

- **Slow HTTP Headers Vulnerability (Slowloris)** – As described in Slowloris HTTP DoS (http://ha.ckers.org/slowloris/), using this technique the client never completes sending the headers. It sends headers one-by-one at regular intervals to keep sockets from closing and the web servers thereby tied up. In particular, threading servers tend to be vulnerable when they try to limit the amount of allowed threading. Slowloris must wait for all of the sockets to become available before successfully consuming them, so for high traffic websites, it may take awhile for the site to free up its sockets.
- **Slow HTTP POST Vulnerability (R-U-Dead-Yet or RUDY)** – Using this technique, the client attempts to DoS the server using long form field submissions. The client sends all of the HTTP headers, one of which is a legitimate Content-Length header with a large value. The client then iteratively injects data into the form's post field at a very slow rate, so the web application keeps waiting for the full data to arrive. Once multiple threads are tied up by waiting, the server eventually runs out of resources and gets DoS'ed. More technical details about layer-7 DDoS attacks can be found in the OWASP lecture: OWASP-Universal-HTTP-DoS (http://www.hybridsec.com/papers/OWASP-Universal-HTTP-DoS.ppt).
- **Slow Read DoS Attack** – Using this attack technique, the client request completes fully. When the server responds, the client advertises very small windows for accepting response data. For a large response (a file download, for example) the client's slow reception rate ties up server resources for a long time. Multiple requests of this type can eventually take the server down.

These requests are layer 7 DoS attacks. They are typically legitimate from a protocol compliance point of view and are therefore not detected by network layer DDoS devices, by IPS/IDS, or even by your ISP. Clients can DoS the server stealthily and slowly, without consuming any significant bandwidth on the network, so they remain otherwise undetected.

How does Slow Client Attack Prevention Work?

The following settings allow the identification of prevention of a slow client request or response attack:

Max Request Timeout

The maximum time allowed to receive a request from a client. If a request does not complete in this time, the connection is terminated, FIN is sent to the client, and further requests are blocked.

Incremental Request Timeout

This value specifies the initial timeout window a client has in which to complete a request. The system then progressively shrinks the window using an adaptive algorithm. If the client repeatedly fails to complete a request in the shrinking window, the request timeout window converges to zero and the connection is dropped. If the client begins to send data at a healthy rate, the window is progressively expanded.

This adaptive algorithm ensures that temporary network delays do not affect genuine clients, but persistent slow clients are detected and denied.

Max Response Timeout
The maximum time allowed to send a response to the client. If the response transfer is not complete in this time, the connection is terminated, Fin is sent to the client, and further responses to the client are not sent.

Incremental Response Timeout
This value specifies the initial timeout window a client has in which to receive a response. The system then progressively shrinks the window using an adaptive algorithm. If the client repeatedly fails to receive the response in the shrinking window, the response timeout window converges to zero and the connection is dropped. If the client begins to receive data at a healthy rate, the window is progressively expanded.

This adaptive algorithm ensures that temporary network delays do not affect genuine clients, but persistent slow clients are detected and denied.

Data Transfer Rate
The minimum data transfer rate the Barracuda WAF-as-a-Service expects for requests from the client and responses to the client. Data transfer rates slower than this are considered slow.

Steps to Configure Slow Client Policy

1. On the WAF-as-a-Service web interface, go to the **APPLICATIONS** page and click on the application to which you want to configure the Slow Client policy.
2. On your application page, click **DDoS** in the left panel and then click **Slow Client**.
3. On the **Slow Client** page, do the following:
    1. **Slow Client Prevention** – Set to **ON** to enable Slow Client Prevention.
    2. Edit the default values if required and click **Save**.