# SlowHTTPTest Package Description

SlowHTTPTest is a highly configurable tool that simulates some Application Layer Denial of Service attacks. It works on majority of Linux platforms, OSX and Cygwin – a Unix-like environment and command-line interface for Microsoft Windows.

It implements most common low-bandwidth Application Layer DoS attacks, such as slowloris, Slow HTTP POST, Slow Read attack (based on TCP persist timer exploit) by draining concurrent connections pool, as well as Apache Range Header attack by causing very significant memory and CPU usage on the server.

Slowloris and Slow HTTP POST DoS attacks rely on the fact that the HTTP protocol, by design, requires requests to be completely received by the server before they are processed. If an HTTP request is not complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this creates a denial of service. This tool is sending partial HTTP requests, trying to get denial of service from target HTTP server.

Source: https://github.com/shekyan/slowhttptest
SlowHTTPTest Homepage | Kali SlowHTTPTest Repo

- Author: shekyan
- License: Apache 2.0

## Tools included in the slowhttptest package

### slowhttptest – A tool to test for slow HTTP DoS vulnerabilities

```
root@kali:~# slowhttptest -h


slowhttptest, a tool to test for slow HTTP DoS vulnerabilities - version 1.6
Usage: slowhttptest [options ...]
Test modes:
  -H              slow headers a.k.a. Slowloris (default)
  -B              slow body a.k.a R-U-Dead-Yet
  -R              range attack a.k.a Apache killer
  -X              slow read a.k.a Slow Read

Reporting options:


  -g              generate statistics with socket state changes (off)
  -o file_prefix   save statistics output in file.html and file.csv (-g required)
  -v level        verbosity level 0-4: Fatal, Info, Error, Warning, Debug

General options:

  -c connections   target number of connections (50)
  -i seconds       interval between followup data in seconds (10)
  -l seconds       target test length in seconds (240)
  -r rate          connections per seconds (50)
  -s bytes         value of Content-Length header if needed (4096)
  -t verb          verb to use in request, default to GET for
                   slow headers and response and to POST for slow body
  -u URL           absolute URL of target (http://localhost/)
  -x bytes         max length of each randomized name/value pair of
                   followup data per tick, e.g. -x 2 generates
                   X-xx: xx for header or &xx=xx for body, where x
                   is random character (32)

Probe/Proxy options:

  -d host:port     all traffic directed through HTTP proxy at host:port (off)
  -e host:port     probe traffic directed through HTTP proxy at host:port (off)
  -p seconds       timeout to wait for HTTP response on probe connection,
                   after which server is considered inaccessible (5)

Range attack specific options:

  -a start         left boundary of range in range header (5)
  -b bytes         limit for range header right boundary values (2000)

Slow read specific options:

  -k num           number of times to repeat same request in the connection. Use to
                  multiply response size if server supports persistent connections (1)
  -n seconds       interval between read operations from recv buffer in seconds (1)
  -w bytes         start of the range advertised window size would be picked from (1)
  -y bytes         end of the range advertised window size would be picked from (512)
  -z bytes         bytes to slow read from receive buffer with single read() call (5)
```

## slowhttptest Usage Example

Use 1000 connections ***(-c 1000)*** with the Slowloris mode ***(-H)***, and generate statistics ***(-g>*** with the output file name ***(-o slowhttp)***. Use 10 seconds to wait for data ***(-i 10)***, 200 connections ***(-r 200)*** with GET requests ***(-t GET)*** against the target

URL ***(-u http://192.168.1.202/index.php)*** with a maximum of length of 24 bytes ***(-x 24)*** and a 3 second time out ***(-p 3)***:

```
root@kali:~# slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 200 -t GET -u http://192.168.1.202/index.php -x 24 -p 3
Sat May 17 10:45:26 2014:
Sat May 17 10:45:26 2014:
    slowhttptest version 1.6
 - https://code.google.com/p/slowhttptest/ -
test type:                        SLOW HEADERS
number of connections:            1000
URL:                              http://192.168.1.202/index.php
verb:                             GET
Content-Length header value:      4096
follow up data max size:          52
interval between follow up data:  10 seconds
connections per seconds:          200
probe connection timeout:         3 seconds
test duration:                    240 seconds
using proxy:                      no proxy

Sat May 17 10:45:26 2014:
slow HTTP test status on 0th second:

initializing:        0
pending:             1
connected:           0
error:               0
closed:              0
service available:   YES
```