CSE3NS/5NS
Project 4

“Slow” DoS attacks on Web Servers (100 points)
Due 22/11/2014

[This project is of 2 weeks duration, but being released early due to student suggestions given the asynchronous nature of the course]

We have studied in class how to use a variety of “slow” attacks to do a DoS attack on Web Servers. These attacks include SlowLoris, RUDY, Slow Reads etc. Perhaps the easiest such attack is SlowLoris. Recall that in this attack, a client sets up a connection to the server, starts a valid HTTP GET request, and then keeps sending headers (including junk headers) every so often, without finishing. This ties up a server thread. If we tie up all server threads, we’ve DOS-ed the server!

For this project, you need to have a vulnerable version of HTTPD running in a VM. The stuff you get by default (typically 2.4.7 these days) will resist these attacks. You will research to find out which versions of HTTPD are vulnerable to these attacks -- older versions from 3-4 years should work. In a different VM, you will need to run the attack. Make sure that these VMs are both on the machine provided to you by the institute, or on your own machines and NOT talking to the outside network.

In the first part of the attack, I want you to download a tool called slowhttptest. For the attack, I want you to understand and then code a client for the the slowloris attack. One is built into the slowhttptest utility, and its code is freely available from Google code. There are other implementations as well. *The intent is NOT for you to copy this code and paste it in!* I want you to understand the attack and write it yourself in your favourite language. Feel free to look at the existing code for help, but acknowledge that explicitly in your readme by saying what parts you coded yourself and where you had help. It is OK to discuss the high level elements of the code with your friends, but NOT the code itself.
Before you proceed, please remember the discussion in the very first class around Ethics. This is ONLY to be done on the resources that have been assigned for this project . Please do not try this on any other machine. If you do this on any live network, you’re probably breaking several criminal statutes that attract strict penalties.

Submission

You will turn in

- A detailed report that shows
    - Current httpd doesn’t succumb to the attacks build into slowhttptest
    - What you did (older version, parameter tinkering) to make it susceptible
    - Analysis of your data regarding the parameters used etc and how they were influenced by the number of threads running in the server. It would be nice to have a graphical analysis -- graphs that show the success of failure of an attack as some parameter is varied, graphs that show the tradeoff of parameter value and number of threads to get the attack to succeed etc.

- Your actual attack client code and a readme file to go with it.