Derrick Murphy

June 28, 2018

Stress Testing

In my 10th grade year in high school I made a major transition into my love of Technology. Before this point I was interested in designing video games and becoming a professional gamer in Call of Duty. Through competitive gaming I was introduced to Denial of Service(DOS) attacks aka booters, ethically titled Server Stress Testing. The power that it gave users caught my attention immediately. This lead to me purchasing booters build on Linux operating systems, the first builds contained Perl scripts that performed denial of service attacks and next web-based models.

After understanding what I was doing and spending money on others service, I decided to build my own booter. Putty is the application used to connect SSH to the server. I joined forums that discussed hacking methods and other topic regarding technology. I began by doing research and messaging member to gain knowledge. I gathered scripts, instructions, and purchased a $5 a month virtual private server for my first service. For a few months this was my method, I would install the libraries I needed and add my scripts. Logins sold for $5 a month or $15 lifetime access of that service. Once a customer payment was deposited, I created the requested username, password, uploaded the scripts, and changed the permissions to run the scripts.

I purchased the servers under random information and on a virtual private network. On a server I added typically 5 users, Perl scripts used bandwidth quickly and was easily detected. So sometime servers were shut down before the month was over. When this happened and users wanted to continue with their subscription, I bought multiple servers. I created new logins and by spreading out the users it help everything run longer.

Soon new attack scripts coded in C became public, introducing me to new DOS methods. One SYN attacks and reflection attacks, properly known as distributed denial of service attacks. These were not easily detected and therefore expanding the lifetime of the booter. Again I did the research to learn about and use these scripts, browsing the internet and messaging forum members. Finding the libraries and how to run the scripts. SYN contained a parameter for number of threads and the DDOS attack, also included a parameter for DNS list.

I downloaded and reprogrammed a leaked bash graphic user interface to make the booter user friendly. I integrated my booter title, color scheme, and the new attack scripts. The GUI displayed options for the different menus(figure 1). Users could chose 1 of 4 attack menus, HTTP flood, UDP flood, SYN flood, or UDP-lag flood; then enter a IP, and time. I coded limits for time, the DNS list, and number of threads; users never needed to call the entire script manually. Customers could also see their subscription duration and subscription title.

Business was flowing okay and I had to provide around the clock customer service. Either there were new customers, server crashes, or subscription renewals to tend to. As business grew I purchased more VPSs to retain attack strength, remain undetected, and expand my clientele. I would be in high school during the day, during this time I would communicate through aim, skype, or private message. When I got home I tended back to my service. Some of the money I earned I spent on myself and my family buying everyone phone cases or other small knick-knacks, the rest was focused on improving my business. Upgrading to dedicated servers, trying new VPSs from different offshore providers, and moving towards a web-based service.

The web-based service was my next project and a more successful model. This project was built with leaked source and reprogrammed to my needs. I had to do more research, and paid a forum member to code some custom scripts and show me how everything worked. I learned to manage a control panel, primarily I needed the SQL databases. There was the information on my customers and place to add my servers. To power my new project I moved my servers from the putty booter, and install apache to create a SSH2 connections. Some customers preferred the putty booter so I ran two services and they kept their logins. For additional strength I bought APIs that performed DDOS attacks also and fresh DNS list. I purchased a domain matching my project title and Cloudflare protection to prevent it from being booted offline.

The web booter had more features, including support on mobile devices, friends & enemies list, skype resolver, Domain to IP, and geolocation tool. Plan options were either paid monthly or lifetime of the service. Both containing bronze, silver, gold, platinum, supreme, and ultimate categories; the differences were the attack length(figure 2).

At the same time I produced web a resolver tool. It could be purchased $1 daily, $3 monthly, or $5 lifetime with no limit on resolves. It offered a skype resolver, IP logger, Down or Not Pinger, Domain to IP, Cloudflare resolver, Geo-location, and friends and enemies organizer. It was offered on the web to avoid the need for port forwarding and installations, different than most other models.

Next, to cut the cost of buying DNS list I learned to create my own. To maintain the power of the booter I frequently bought fresh DNS list, because they were not permanent and eventually burnt out. I started by using public scanners, programmed in Perl and C. I researched how to use the scanners, understanding the libraries I needed and how to run the scripts. The C scanner ran more efficient than the Perl, but the lists produced were not strong like the ones I purchased. I messaged the person I bought DNS lists from asking for help. He taught me his method and gave me another shell script to parse the needed data. He used the same C scanner as me, which found recursive DNSs. Recursive DNSs are the most important aspect, but he taught me that does not make a strong list. We joined together on a TeamViewer session and he moved through the entire process, I recorded everything to study what was done.

I chose and range to scan, which returned open recursive resolvers and put those into a text file. There was still a large list of recursive DNSs, so that list was split using the command line tool offered in Unix. Start by attacking the unfiltered DNS list on another, stronger, servers with an IP traffic monitor installed to log the data. After a few rounds going through the entire list, the traffic was logged in another text file. The shell script parsed for servers responding with the peak data and put that into another concentrated list. The new filtered list then ran through the same process, because you wanted to filter out the most temporary addresses. After a few iterations, there was a large list still left and you simply split the text file into 50s.

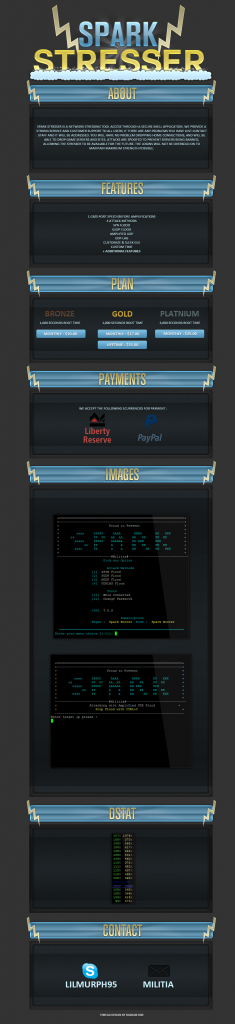
Now I could make fresh list I began to sell those as well. Depending on how much filtering was done I could create stronger lists. My list strengths basically ranged from 1Gb/s to over 3Gb/s output from a 100Mbit VPS. The opportunity to create list was abundant, every IP address from 0.0.0.1 to 255.255.255.254 could be scanned. With the exception of a few reserved addresses, nothing compared to what is available. As a result I started building my own API service to power other web booters.

During my ventures I have lost money from hopes of remaining anonymous, bad server purchases, servers being shut down, and twice through scams. On my monetary accounts I did not verified my information, if they required social security information and ID. On one of my accounts I attempted to transfer money out after receiving payments and it required me to verify my information first. I considered that a loss and continued to receive payments on other accounts. Another account was gone after a company closed its doors and shut the website down. For my last API project I wanted a custom built website source that was for sell on a forum I used. I sent my payment and promised I would be emailed the source. Time went by and I understood it was a scam and filed a report. Other members confirmed this by commenting on my report that I was not the only person they scammed. Soon I saw those scamming members accounts banned and receiving bad reputation. I lost hundreds of dollars on buying servers that did not allow header modification, so could did not run the reflection scripts. Bad sever purchases and server shutdowns were the top reason for me exiting stress testing services. I would then just use them as hosts for my website and back up storage.

By the beginning of my senior year I moved on to my final major project offering a API to other web-based booters. I found an partner that was an excellent coder. I found a leaked source made for a web booter and had my partner reprogram the layout to fit what we were trying to accomplish. I had an idea of a rotational system for our service. With the rotational system once the top server was called it moved to the bottom of our inventory of attack servers, he coded this into the project. When attacks initiate concurrent on the same server the strength is reduced, because each attack is sharing resources. This ensured our attacks were at the peak performance available. He taught me a lot about coding and help me improve my coding. So, I helped change some pages resulting in our site being ready sooner.

The new GUI contained manager and customer page layouts. The customer layout was user friendly and they could view their concurrent attacks, stop attacks, view their plans, see available servers, and update their domains. The API was locked to one domain for every customer, which we allowed to be updated in their settings, and required a key they chose when they registered. There was only monthly based plans, prices varied with attack length and number of concurrent attacks.

A major shut down to from my primary server provider swayed me to stop my stress testing conquest. Most of my attack servers were based out of the same provider. Since I began using this provider none of my servers had ever been shut down and I deemed them my go to. Once they did shut down my servers, I only had a few from other providers. Those were not enough to support my service, so I tried to adjust quickly. I began purchasing servers from new providers, some did not produce the reflection attacks or got shut down quickly, this discouraged me. Though I bought some good servers they took a lot away from my profit, because they cost much more. Eventually, unable to support my service like I wanted I chose to leave the projects after my current servers ended. I was in my senior year graduating, the basketball coach always wanted me to play, so I joined the basketball team. I still received messages from potential customer and those wanting me to continue, but I chose not to. Applying for college I thought to pursue a Computer Engineering degree, but ended up graduating with a Bachelors in Computer Science.



(figure 1. Thread of first major putty booter)



(figure 2. Thread of first web booter)