Derrick Murphy

07/01/2018

Server 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. One of my gaming friends introduced me to Denial of Service (DOS) attacks known as booters. In the realm of competitive gaming booters were popular, they disconnected opponents from matches to create an advantage. My friend gave me a quick lesson on booters and showed me where to buy them. Once I my own research I bought one. At first I bought a web booter just to experience having a booter. Next, I started buying putty booters. Those are servers installed with a Linux operating systems, DOS scripts are added, then accessed with the putty application. The putty application created a secure shell (SSH) connection to access the server. Ethically titled Server Stress Testing, the power that it gave users caught my attention immediately.

After understanding what I was spending my money on, I started my own service. For my first service I created a putty booter. I joined forums that discussed hacking methods and other technology topic. I did research and messaged member to learn more. Then gathered instructions, attack scripts, and bought a Virtual Private Server (VPS). For a few months this was my method, I would install the libraries I needed and add my scripts. I sold logins for $5 a month, $15 for lifetime access, and also would negotiate. Once a customer payment was deposited their user login was created with the requested username and password. The first scripts were coded in Perl, these used a lot of system resources and bandwidth quickly. Those scripts were easily detected and often got servers were shut down early. When this happened and customers wanted to continue with their subscription, I bought multiple VPSs to relieve the pressure. I only paid $5 for my VPSs, so I had enough profit to increase my network. This was the main issue until new attack scripts were released.

After a few months new scripts coded in C became public, introducing me to new attack methods. They were they a UPD reflection attack and SYN attack. Again I did my research to learn what these scripts were and how to use them. I understood the parameters for each script and installed the required libraries to run them. Both included a new parameter for number of threads, but the reflection attack required a Domain Name Server (DNS) list. The reflection attack created a Distributed Denial of Service (DDOS) attack, stronger than just a normal DOS attacks. Both were required less resources than the Perl scripts and were not detected as easy. Therefore expanded the lifetime of the booter and satisfying customers.

I downloaded a public bash Graphic User Interface (GUI) and reprogrammed it with some of my own ideas. I integrated my booter title, a different color scheme, and the new attack scripts. The GUI displayed different menu options, the customer subscription duration, and limits for their time in the attack menus (figure 1). Customers could chose 1 of 4 attack menus, HTTP flood, UDP flood, SYN flood, or UDP-lag flood. For each attack customers only needed to enter an IP and time, because I physically coded in the other parameters. To create ease of use, I did not want customers call the entire script manually every time.

As my clientele grew I focused my profits toward improving my business. I had to maintain attack strength, remain undetected, and expand my network. I started to try new VPS providers and dedicated servers. I had to provide around the clock customer service, either there were new customers, server crashes, or subscription renewals to tend to. Being in high school during the day, I communicated through instant message or private message. When I got home I tended back to my service. I wanted to continue improving, so I moved toward creating a web-based booter.

The web booter was a successful project and I was proud of my results. I did more research and found useful files. The service was built from a leaked source and reprogrammed to my needs. I paid a forum member to teach me how everything worked and code a custom SSH2 script. I purchased a domain matching my project title and Cloudflare protection to prevent my service from being booted offline. I learned to manage a control panel, primarily I needed the SQL databases. There was the information on my customers and tables to add my servers. Each attack method required its own server. New attack APIs were on the rise, so I used those and my own servers to power my service. Available servers from the putty booter had the necessary libraries installed and apache to create a SSH2 connections. Some customers preferred the putty booter so I ran both services. 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.

DNS lists were not permanent and eventually lost strength, I to cut my cost learned to make my own list. Reflection attacks were the most efficient attack method, to maintain the power I regularly bought fresh list. I did my research to understand what the lists were and how to make them. These were lists of open recursive DNSs and I found public scanners that skimmed the internet for them, programmed in Perl and C. I learned how to use the scanners and what was needed to run them. The C scanner ran more efficient than the Perl, but the lists were still not strong like the ones I purchased. I messaged the seller I bought DNS lists from asking for help, unexpectedly he taught me his method for free. He used the same C scanner as me and gave me a shell script that parsed data. Recursive DNSs are the most important aspect, but he taught me how make a strong list. We joined together in a TeamViewer session and moved through the entire process, I recorded everything to reference anytime.

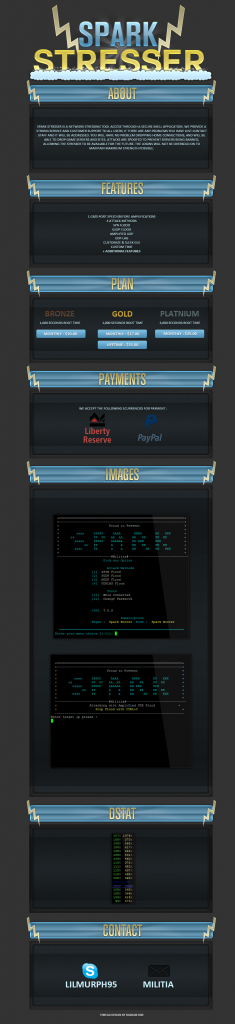
First, I chose and range to scan and the script put what was returned into a text file. There was a large list more than likely, so the text file was split using the standard command line tool. Attacking the unfiltered DNS list on another, stronger, servers with an IP traffic monitor installed to log 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 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 rounds you split the text file into lines of 50 and I was finished. After learning to scan for my own list I opened a service to sell them. Depending on how much filtering was done the lists strength varied. My list ranged from 1Gb/s to over 3Gb/s output from a 100Mbit VPS. With the exception of the few reserved addresses, the entire internet could be scanned. The opportunity to create lists are abundant. 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 scam. On my monetary accounts I did not verified my information, if they required social security information and ID. Once an account I used required me to verify my information first to transfer money out, I had about a $100 in payments. I considered this a loss and continued to receive payments on other accounts. Another account was deleted after a company closed its doors and shut their website down. For my API project I wanted a custom built website source that was for sell on a forum I used. I sent my payment and was promised I would be emailed the source. Time was going by and I spoke with the sellers a few times. After a while I understood it was a scam and filed a report. Other members confirmed this by commenting on my report, they said that I was not the only person scammed for that source. Soon I saw those members accounts banned and receiving bad reputation. I spent hundreds of dollars on servers that did not allow header modification, which is the function that enables DNS lists to be used. That left those servers to only servers to be used as hosts or backup storage. My server shutdowns and bad purchases were the top reason for me exiting stress testing.

By the beginning of my senior year I moved on to my final major project, an attack API to power other web booters. I found an partner that was an excellent coder. He helped me improve my coding skills and I helped get our service ready sooner. The project was built on a leaked source made for a web booter and reprogrammed to our needs. My favorite idea I had for the project was a rotational system for our servers, because concurrent attacks shared resources; reducing strength. The idea was to pop the top server and move it to the bottom of the inventory of once an attack began. My partner coded this into the project guaranteeing our attacks were at the peak performance available. For security the API was domain locked and required a key customers chose at registry. The website included a manager layout, customer page layouts, and a front page server count. The customer layout was user friendly with options to view their attacks history, stop attacks, view plans, or update their domain. The manager layout had a options to add customers, add servers, or view all attack history. Only monthly based was offered, prices varied with attack length and number of concurrent attacks.

A major shut down to from my primary server provider discouraged me into stopping my stress testing ventures. Most of my attack servers were based out of one provider. Since I began using this provider none of my servers had ever been shut down, I considered them my go to. Once they final shutdown my servers I only had a few from other providers. Those were not enough to support my service, so I tried to adjust. I purchased servers from new providers, many did not produce the reflection attacks or got shut down quickly. I bought some good servers, but they were expensive and reduced my profits. Eventually, I closed the projects after my current servers ended.

I was in my senior year and planning on going to college. As a result of everything I done, I intended on majoring in Computer Engineering. To fill in my free time I joined the basketball team. The coach always wanted me to play, so I tried out. I checked my accounts occasionally and still received messages from potential customer, but I chose leave everything alone. After graduating I enrolled at Winston-Salem State University along with some of my peers. Computer Engineering was not offered, but graduated with a notable Bachelors in Computer Science.



(figure 1. Thread of first major putty booter)



(figure 2. Thread of first web booter)