Detecting Application Layer DDoS Attacks Using TLS Fingerprinting

GIAC (GCIA) Gold Certification

Author: Alejandro Aucestovar, aucestovara@gmail.com
Advisor: *Domenica Crognale*

Accepted: *12/10/2021*

Abstract

Application layer DDoS attacks are some of the most complex and devastating attacks on the modern internet. Unlike their lower-layer counterparts, application-layer DDoS attacks utilize the widely accepted TLS encryption, commonly used across the internet, to their advantage so that identification and mitigation do not happen easily. Previous research has had different levels of success at identifying DDoS attacks and differentiating them from legitimate human traffic as well as legitimate flash flood events. Adding TLS fingerprinting details of a client/server communication to the identification methodology previously used and tested by researchers will increase fidelity in identifying application-layer DDoS attacks. The combined identification methodology and JA3 fingerprinting technique were tested against the Canadian Institute of Cybersecurity's DDoS dataset created in 2019. TLS Fingerprinting successfully identified illegitimate traffic by providing details of the user/client operating system, browser information, and application components.

1. Introduction

The rise in consumer consumption and e-commerce via websites and applications has risen significantly in the last two decades and is a major revenue focus for many companies. Some companies like Amazon exist almost entirely online and rely on their systems and applications to be always available to the public. For this reason, distributed denial of service attacks have been a major concern for security experts, business leaders, and e-commerce and financial institutions worldwide for the last two decades. The financial impact of a DDoS attack on an e-commerce or financial website can range vastly depending on the duration of the attack and the assets of the victim company. To protect their online presence, companies are forced to decide how to mitigate DDoS attacks with the assistance of cyber security providers that provide protection or with cloud service providers that provide scalability, load balancing, and global traffic management.

Companies such as Akamai, Cloudflare, Radware, and NexusGuard are just a few cyber security providers that companies call upon for DDoS protection. These companies provide a variety of services that can identify and mitigate DDoS attacks. These services include serving a company's static content on the service provider's network, detecting high traffic rates or behavioral patterns via complex algorithms, or using proxies to inspect and detect abnormalities before the traffic reaches the real e-commerce website. Since DDoS protection is highly complex and does not always work, especially with the rise of amplification via botnets, companies have increasingly chosen to seek help from cloud providers such as Google and Amazon. Cloud service providers are often a good choice for companies since the cloud provider is usually responsible for identifying and mitigating DDoS attacks on their online systems. Cloud providers can also spin up new servers and provide load balancing to those new servers with almost no delay or impact to the company website. Even though cloud providers can provide availability services, they are not exempt from the effects of a DDoS attack, especially those initiated by botnets such as Mirai botnet. The best approach to combatting DDoS attacks is to take a hybrid approach where the company hires cyber security and cloud availability services for their e-commerce sites and applications.

Alejandro Aucestovar, aucestovara@gmail.com

DDoS attacks come in many different shapes and sizes, and their successful detection and mitigation must match the shape and size of those attacks. In other words, if a DDoS attack utilizes an open stack interconnection (OSI) layer four, then the detection and mitigation must occur within OSI layer four. Cyber security providers are particularly adept at detecting and mitigating layer four and layer five DDoS attacks. Cyber security providers attempt to profile traffic patterns that match certain typical human traits and behaviors and compare those behaviors to the behaviors of bots. DDoS attacks at the application layer or the OSI layer seven present a more complex problem for cyber security companies because of the additional packet inspection needed to view the data and establish normal behaviors of legitimate customers. The use of encryption via transport layer security (TLS) or secure sockets layer (SSL) makes this problem even more complex.

Police agencies across the globe have used fingerprinting as part of their investigation processes to tie perpetrators to crime scenes because of the unique characteristics individual humans possess. This research will explore using various aspects of the TLS handshake to fingerprint the traffic from legitimate and illegitimate users to DDoS attacks such as HTTP Get Flood over SSL. When used in conjunction with previously used characteristics of detection methods, TLS fingerprinting could provide higher fidelity in detecting application-layer DDoS attacks.

1.1. TLS Fingerprinting

The increase of cyber-attacks prompted companies to use TLS to encrypt their communications with their customers as early as 1999. However, TLS encryption was not widely adopted until much later. The purpose of TLS was to provide an upgrade to its predecessor, SSL 3.0, which was susceptible to POODLE attack, by using a standardization protocol that can be used to define cipher suites with symmetric cryptography, which include RC4, Triple DES, and AES (RFC 2246). Several subsequent versions of TLS (TLS1.1, TLS 1.2, TLS1.3) have been developed and superseded for various reasons and shortcomings, but the purpose of TLS, to provide communications privacy and prevent eavesdropping, has remained the same. RFC 2246 outlines the process undertaken by both client and server to establish secure

communication with encryption, authentication, and integrity through a handshake and record process. The primary function of the recording process is to encrypt and send the data using the information from the handshake process. The TLS handshake process begins after the transport connection protocol (TCP) handshake and is always transmitted in the clear with little disruption to the services. TLS fingerprinting takes advantage of the data transmitted in the clear during the handshake process to obtain details about the client and the server and preserve the messages' privacy.

1.2. Research Goals

Researchers and security professionals attempt to differentiate legitimate from illegitimate traffic in DDoS attacks by identifying normal user behavior and setting it as the baseline. Once this baseline is established, all traffic that does not fit the baseline is illegitimate, generated by bots, scripts, or non-human interaction. Flash flood traffic creates a problem for researchers since the traffic is legitimate but behaves like illegitimate bot traffic. Detection and mitigation of DDoS attacks have focused on the characteristics of either user attributes, traffic attributes, or traffic rates (Karanpreet et al., 2016). A combination of user attributes such as user-agents, header data, the number of pages visited, sequence of pages viewed, or rate of received/sent packets have been analyzed and subjected to rigorous mathematical algorithms to determine whether the traffic is normal, bot, or script generated, or flash flood. The notion that human users act differently than bots is at the center of this analysis.

Relationships between humans have always been based on how well we know each other and trust each other. Generally speaking, the more we know about each other, the more trust we provide. This concept derived from the human trust should be applied to our internet interactions to measure client/server trust. For example, in a banking website, a user who provides personal details and creates an account is provided more access to the website than a customer who does not. These customers with accounts are trusted more because the banking website knows more about them. The user even reciprocates this trust to the bank since the bank is holding their money. Researchers and security professionals apply this concept to incoming traffic as they focus on the user and traffic characteristics that make up the ‘recipe’ identifying legitimate, trusted traffic. This

concept has shown varying results in identifying and mitigating application-layer DDoS attacks (Park et al., 2021, Zhao et al., 2018, Wei et al., 2020, Suchacka et al., 2020). Still, TLS fingerprinting could add trust and fidelity to this process by providing more information about the user initiating the traffic. The more we (website owners, security professionals, business leaders, etc.) know who is coming to our websites and what they are doing, the better we will get to know them (users, clients, etc.) and attach the appropriate level of trust.

2. DDoS Identification Recipe

Application layer DDoS attacks can generally be categorized into high-rate and low-rate attacks (Mirkovic et al., 2004). Low-rate attacks such as Slowloris can begin at slow or constant rates but do not stop, which bottles up server resources. High-rate attacks such as HTTP Get Floods can vary with elevated rates but still impact the server's performance. Whether categorized as the low-rate or high-rate, application-layer DDoS attacks are far more complex than lower layer attacks because of all the valid components that the traffic contains, such as valid TCP connections, valid IPs, and valid requests. They do this under the protection of TLS encryption. Researchers and security professionals have sought help from web application firewalls (WAF) to try and detect and mitigate application-layer DDoS attacks. WAFs are incredibly helpful in identifying layer seven attacks and are considered "must-haves" when building security architecture around web applications— but they cannot catch everything. Even though most WAFs are well suited for signature detection, they cannot detect attacks based on behavioral analysis, which is needed in DDoS attacks. Researchers must rely on identifying user and traffic characteristics attributable to normal human traffic and classify other traffic as bot traffic.

2.1. Human, Bot, and Flash Flood Traffic

Botnets are groups of interconnected computers and devices hijacked and controlled by a remote user without the knowledge of its real user to conduct DDoS attacks (Park et al., 2021). Distinguishing between bot-related traffic and human-related traffic is the main objective in all DDoS attack research. Normal human interaction with

a website, no matter the purpose of the website, should be different from the interaction and traffic from bot-related traffic.

2.1.1. Human Versus Bot Traffic

According to Suchaka et al. (2020), technological advances have made bot traffic extremely difficult to distinguish from legitimate human web traffic. The bots, botnets, and shilling attacks try to mimic human traffic. Bots are similar to scripts in that they help humans with repetitive tasks that may not be very complicated but are high in the number of events to be completed. According to Lagopulous et al. (2017), bot traffic was identified by a high number of repeated requests, long session durations, and a high number of total requests. Similarly, Radware Bot Manager (n.d.) discusses how bot traffic is detected when there are many hits from a single IP during a short time. Human traffic would have a significantly lower number of repeated requests, uneven session durations, and fewer total requests. Further, Sreeram et al. (2019) identified that bot traffic had significantly lower entropy levels of IP geolocation compared to human traffic.

Human traffic resembles human behavior in that the session durations change from website to website, dictated by personal interest. Humans tend to request only interesting pages, whereas bots request all web pages or the same pages across different websites. Bot traffic, much like automated scripts, is continuously unchanging in duration, in the number of requests and the type of information sought. Karanpreet et al. (2017) conducted a systematic survey of approximately 63 research studies dedicated to application layer 7 DDoS attacks. They provided the following characteristics pertaining to human and bot traffic: “Some actions or features usually associated with the legitimate users are as follows:

- use bookmarks to open web pages;
- navigate to a web page through search engines;
- make use of hyperlinks to navigate among web pages;
- generate legible mouse click and scroll events on web pages;

- likely to request for popular web pages;
- hardly ever repeat their web access patterns;
- rely on legitimate web browsers to connect to the server;
- exhibit widely dispersed geographical distribution.

Some of the few properties related to attacking bots are as follows:

- infected by the presence of malware;
- await attacker's command before initiating an attack;
- provide false identity by faking user-agent strings;
- almost similar access patterns among all attacking bots;
- access behavior more often than not deviates from legitimate users;
- tend to re-iterate their access patterns;
- concentrated geographical distribution at various locations" (Karanpreet et al., 2017, pp7-8).

2.1.2. Flash Flood Events

Flash flood events are those events that are generated by a high number of requests from legitimate users, usually as a result of a marketing campaign, new web page or product, or new services but behave much like illegitimate DDoS traffic. In a sense, flash flood events mimic not only human behavior but also bot frequency and velocity. According to Khalaf et al. (2021), flash flood events can have the same impact as illegitimate DDoS traffic, overloading the server, computational performance, and exhausting resources. However, it is created by legitimate users trying to access the website. According to Singh et al. (2018), bot-generated DDoS traffic has almost identical inter-arrival time rates as flash flood traffic. Still, it is different in that the number of instances per IP address is much larger in DDoS attacks. In other words, although requests are coming in almost simultaneously in both DDoS and flash flood events, there are more instances, connections, or sessions per IP during a DDoS attack.

2.1.3. Selected Research to Emulate

Throughout the research process of this whitepaper, the work of Karanpreet et al. (2017) was a highly relevant contribution to the research and body of work on application-layer DDoS attacks. Karanpreet and associates conducted a wide-ranging survey of all available application layer DDoS research from 2005 to 2015, focusing on flood attacks. After reviewing all the selected studies within their work, only the newest studies conducted in 2015 were selected for review. Out of the five selected studies, only two studies used multiple detection attributes that had either a high likelihood of detection for typical human values or typical bot values based on Karanpreets' analysis. Saleh et al. (2015) used a combination of the total number of requests and the total number of hot pages and calculated a ratio which was then used to multiply by the log of the URI count and total count ratio. Liao et al. (2015) used a combination of all users' total number of requests and the number of total objects. They created a ratio that was then applied to the likelihood of switching between web pages. Since the dataset used for this research does not seem to use hot pages, this whitepaper will use the methodology described by Liao.

2.1.4. Analysis Tools and Test Data

Conducting research in the analysis of cybers attacks has always been difficult because it is highly unlikely that a company allows its data to be shared with the public and be seen by potential malicious persons. In the past, DDoS researchers have taken one of two approaches, creating their data in their control lab environments or obtaining publicly accessible data sets. The obvious problem with researchers creating their data in a controlled lab is that there is little trust the data resembles actual DDoS data in the wild. A tool such as hping3 from Kali Linux is one of the tools used by actors to generate DDoS attacks. Hping3 tools allow for a script to be developed and executed to generate hundreds of thousands of HTTP GET and POST requests to single or multiple targets. Another tool used by both professionals and bad actors is the HTTP Unbearable Load King (HULK) created by Barry Schteiman, which was designed to generate and send volumes of obfuscated traffic to overwhelm the web servers and act as a stress tester. Although the tools used in the lab environment are likely the same as those used by

attackers, professionals still distrust professionals that the lab traffic is the same as DDoS attack traffic.

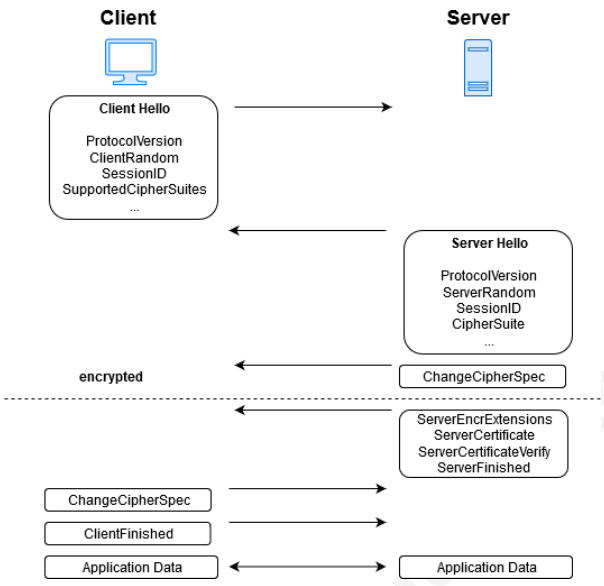
Publicly accessible data composed of DDoS attack traffic does not carry the same distrust from researchers and security professionals. The ability to analyze and take apart real DDoS traffic is a great advantage to researchers and professionals because they can look deep into the attack's process, generation, and logic. This deep insight into DDoS traffic can provide the sought-after answers that the community desire for the identification and mitigation of DDoS attacks, theoretically. Unfortunately, all the publicly accessible datasets are rather old, dating back to 1998 when the internet was just getting started. Sreeram et al. (2017) use a dataset created in 2007 by the Center for Applied Internet Data Analysis (CAIDA) from the University of California's San Diego Supercomputer Center to predict Application Layer DDoS using a bio-inspired bat algorithm. Behal et al. (2016) reviewed approximately 16 application-layer DDoS attacks that attempted to discriminate a DDoS attack from a flash flood event. Behal and associates found that all 16 studies used the 2007 CAIDA dataset or the 1998 FIFA World Cup dataset, which the Network Research Group maintained at Lawrence Berkeley National Laboratory, Berkley, CA. The internet has certainly changed since 1998 and 2007, from the increased use of internet services to more secure protocols such as TLS encryption. TLS 1.2 was released to the public in August 2008 and was not widely used until recent years. There is no question that researchers and security professionals need a more representative dataset to identify and mitigate application-layer DDoS attacks.

2.1.5. Canadian Institute for Cybersecurity (CIC) DDoS Dataset 2019

This research will utilize a newer dataset for experimentation, using some of the characteristics already found in previous application layer DDoS studies. It will also apply TLS fingerprinting techniques to increase fidelity in identifying these types of attacks. The CICDDoS2019 dataset was created by researchers from the University of New Brunswick after reviewing publicly available DDoS datasets since 2007 and finding numerous shortcomings. Researchers then used the CICFlowMeter-V3 tool to address many of the shortcomings of previous datasets to provide a more realistic and representative DDoS dataset of today's internet. The CICDDoS2019 dataset proposes a new detection and family classification approach for application-layer DDoS attacks. The dataset was created in two days which was comprised of a training day and a testing day. During the training day, traffic was generated for 12 DDoS attacks, including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP. During the testing day, seven attacks were generated, including PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and SYN. The researchers created an abstract behavior of approximately 25 legitimate users using profiles from HTTP, HTTPS, FTP, SSH, and email protocols.

3. TLS Fingerprinting Methods

As previously mentioned, TLS fingerprinting relies on certain characteristics of TLS communications that are required to take place per RFC and within the implementation of TLS 1.2. According to RFC 5246, TLS 1.2 has two parts, the handshake protocol, and the record protocol. The handshake protocol is a process where the client and server communicate to establish an encrypted channel by exchanging and negotiating their allowable cipher mechanisms and keys. The handshake process takes place in plaintext, is visible to anyone sniffing out the traffic, and is valuable to the TLS fingerprinting process. Gancheva et al. (2020) provide an excellent illustration of the TLS handshake process in Figure 2 below.



Gancheva et al (2020) Figure 2, p. 17

The exchange of hello messages and corresponding data is sent in plain text along with the server ChangeCipherSpec. Researchers can use these parameters to conduct TLS fingerprinting of both the client and the server, which can be applied to future connections and subsequent sessions.

3.1 **TLS Fingerprinting Methods**

The parameters of the TLS handshake can be used to create profiles or to fingerprint the clients and servers that are making connection requests to each other. The way a client makes a connection and executes the handshake process should be the same when making future or subsequent connection requests to the same server. Similarly, servers can be expected to also behave in the same manner when responding to future or subsequent requests from a previous client. This server behavior can be seen in the TLS fingerprinting technique known as Markov Chain fingerprinting. The Markov technique is complex but focuses on the server responses based on the message type sequence and applies statistical analysis to either first-order or second-order chains.

Another technique is the network-based HTTPS client identification TLS fingerprinting process. This technique focuses on the client hello messages by creating a dictionary where the cipher suites are paired with the user-agent. The dictionary is then

applied to either a host-based, which uses information from HTTP header once the server decrypts the connection, or flow-based, which uses cipher suites from HTTP connections and combines the user-agent from the HTTP connection.

A third TLS fingerprinting method is the JA3/JA3S method developed by Salesforce engineers John Althouse, Jeff Atkinson, and Josh Atkins. This method converts fields from the ClientHello message such as version, accepted ciphers, and other fields into decimals, and then MD5 hashed as the first step of the fingerprinting process. Secondly, the same process is conducted on the server response using the data from the ServerHello message (Althouse, 2019). Security professionals have received the JA3/JA3S fingerprinting technique with open arms, especially those researching malware analysis. Researchers can fingerprint the connection behaviors of well-known hacking tools such as Kali Linux and Metasploit and some indicators of compromise from malware embedded within the software. Unfortunately, bad actors are now fully aware of this TLS fingerprinting technique. They have found ways around the process by changing their TLS connection behaviors or by spoofing other characteristics or fields used in JA3/JA3S fingerprinting. Fortunately, for this research into application-layer DDoS attacks, it is unnecessary to worry about bad actors spoofing their JA3 fingerprints to evade specific malware signature detection. The goal is to determine if the client is a legitimate user such as a human or an illegitimate user such as a bot. In other words, even if the JA3/JA3S fingerprint does not match the signature of a well-known hacking tool or malware, the JA3/JA3S fingerprint should remain the same with multiple, subsequent, or future connections from the same client. Adding the details of the JA3/JA3S fingerprinting process as a property or characteristic of an illegitimate or legitimate user may provide fidelity to application layer DDoS detection.

4. Analysis

The CICDDoS2019 dataset was analyzed with Wireshark and Tcpdump. The detection attributes section focuses on the encrypted data extracted from the WebDDoS data using Tcpdump. TLS fingerprinting was conducted using a Python script that used

the JA3/JA3S fingerprinting technique and focused on the TLS handshake data extracted from the WebDDoS traffic.

4.1. WebDDoS Data Preparation

The entire CICDDoS2019 dataset is comprised of two days of DDoS attack traffic split into approximately 146 large PCAP files, which are timestamped directly on the CIC's dataset webpage. The WebDDoS was conducted from 13:18 to 13:29 on March 11, 2018, and was documented in a PCAP file named SAT-03-11-2018_0144. The victim network was 192.168.50.0/24 with only 192.168.50.6 and 192.168.50.8-9 active during the WebDDoS attack. The following illustration captures the web architecture of the victim network and the traffic flow of the DDoS attacks directly taken from the CICDDoS2019 website.

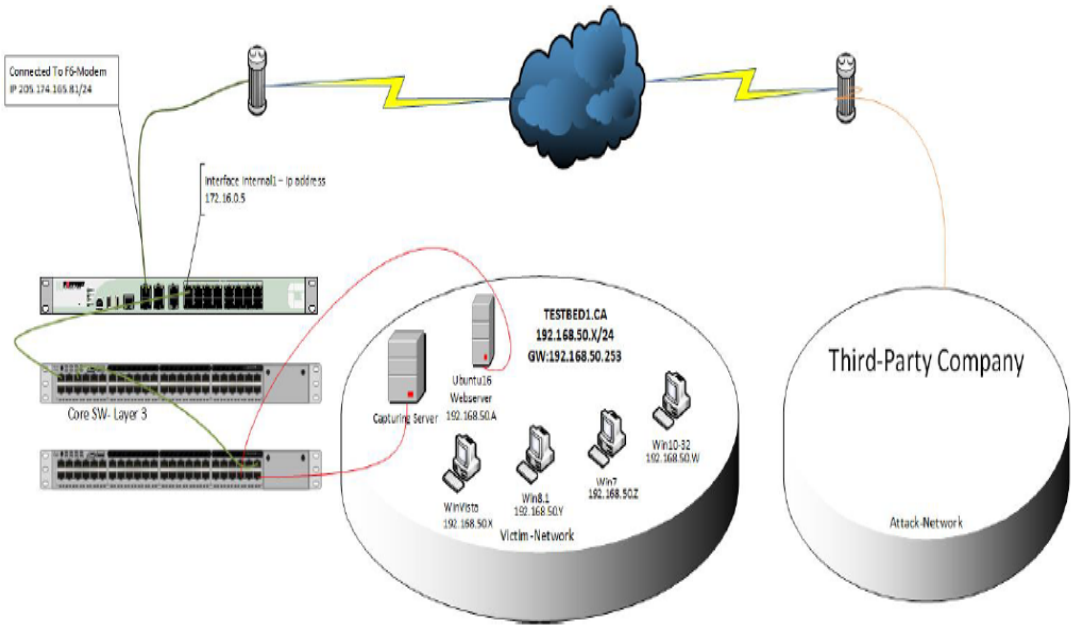


Figure 2: Testbed Architecture

The architecture design used by the CIC did not utilize a firewall, intrusion detection or intrusion prevention system, or any other modern security controls. There is an opportunity to use the F5-Modem as a firewall or filtering protocol that could have been utilized to inspect the traffic. Still, the researchers did not choose to do this without

explanation or comment. The CIC researchers did not provide a source internet protocol address or domain for the third party company, nor did they indicate legitimate users.

4.2. Detection Attributes

According to Liao et al. (2015), their detection algorithm was derived from two derived attributes: the request interval sequence and the request frequency sequence. In order words, the distinction between DDoS illegitimate traffic and legitimate human traffic depends on the average popularity of web objects and the average transition between each object. The average popularity of web objects is the ratio of all users' total number of requests over the number of total objects. The average transition between each object is the time between the last request and the following request and the likelihood of visiting or requesting those more popular web objects. Liao et al. (2015) rationalize that illegitimate DDoS traffic is generated by bots and will have a regular or rhythmic pattern, whereas legitimate human traffic is highly irregular and unpredictable. Furthermore, legitimate human traffic resembles human behavior in that only interesting web objects will be requested. In contrast, illegitimate bot traffic will request all web objects or a consistent number or pattern of objects, but consistently more than humans.

4.2.1. Request Frequency Sequence (RFS)

The request frequency sequence has always been a pivotal problem for researchers and security professionals in the HTTP Get Flood over SSL/TLS. Encryption hides the fine details of a client's request, and this shortcoming continued in this research. Without the SSL/TLS certificates used during the testing and creation of the CICDDoS2019 dataset, the HTTP requests to the victim network could not be inspected. The dataset creators were contacted to retrieve TLS certificates via personal and work email but did not respond. Even though the fine details of the requests were not available, those connections and requests to the victim network were still analyzed. Since this characteristic focuses on the request frequency sequence, all traffic during the TCP and TLS handshake must be excluded. Tcpdump was used to look at the traffic over port 443 to the victim network and isolated the encrypted application data by filtering the packets with a 17 in the first nibble of the TCP header. From 15:37:01UTC to 17:15:12UTC, there were 30274 requests to the victim network, from approximately 620

distinct source IPs, at a rate of approximately 308.91 requests per minute. During the same time frame, there were 18846 encrypted requests to 192.168.50.9, 4786 encrypted requests to host 192.168.50.8, 6340 requests to host 192.168.50.6, and 302 requests to 192.168.50.4.

4.2.2. Request Interval Sequence (RIS)

The request interval sequence was easier to determine since there was no need to decrypt any traffic, and the focus was on the timestamps of each request. Using Tcpdump, the data were extracted and transferred into a Microsoft Excel spreadsheet with time stamps and time deltas in microsecond precision (Tcpdump -ttt) into four separate files. The largest file contained all the requests sent to the victim network 192.168.50.0/24, and the other files contained the traffic sent to the other three victim hosts192.168.50.9-8-6. According to Liao et al. (2015), the distinction between legitimate and illegitimate traffic is that greater request intervals characterize legitimate traffic. Table 1 depicts statics based on the time interval between the current request and the last request.

Table 1. Request Interval Sequence Statistics

There were over 30,000 requests in all four files, and after visually inspecting all the requests, approximately 65% of the requests were sent to host 192.168.50.9. Even though most of the traffic was sent to host 50.9, three timeframes in each of the hosts (50.8 and 50.6) were compared to understand the request interval sequence. Table 2 depicts the one-minute time frames where the number of requests was higher in all three victim hosts. Compared to the overall victim network statistics, the average request interval sequence for both hosts, 50.8 and 50.6, was significantly smaller. Host 50.6 had a significantly faster request interval sequence during the timeframe 17:04 UTC. There were only five source IPs, 172.217.10.226, 172.217.10.2, 172.217.9.226, 172.217.11.2,

216.58.219.226, sending traffic during this time. These source IPs are good candidates to be classified as illegitimate.

Table 2. Request Interval Sequence for Select One-Minute Timeframes

The creators of the CICDDoS2019 dataset included the abstract behavior of 25 legitimate users. Some possible legitimate users were discovered searching the dataset via Tcpdump, focusing on single-source addresses with no more than 50 requests. Table 3 depicts request interval sequence statistics for possible legitimate users. Host 50.9 had 50 requests from host 13.35.78.80 during the entire duration of the WebDDoS attack, with an average of approximately 11.79 seconds between each request. Compared to the average request interval sequence of host 13.35.78.80, this request interval sequence is significantly larger and makes a good candidate to be classified as legitimate.

Table 3. Request Interval Sequence Statistics for Legitimate Users

4.3. JA3 Fingerprinting

The JA3/JA3S fingerprinting technique profiles both candidates' legitimate and illegitimate users from the CICDDoS2019 dataset. To calculate JA3/JA3S fingerprints, a Lua plugin for Wireshark was used. I then created two filters, "JA3" and "JA3S," which were then used to display as columns for easy tracking. The illegitimate candidate hosts sending traffic to host 50.6 were isolated and then exported into a smaller PCAP file to be examined with Tcpdump. The JA3/JA3S calculations for the candidate illegitimate hosts showed three distinct JA3/JA3S fingerprints that all the candidate hosts used. Table 4 depicts the details on the JA3 calculations and the server the traffic was sent to per the illegitimate candidate hosts.

Client	JA3 Fingerprint	Server
172.217.10.226	b20b44b18b853ef29ab773e921b03422 334da95730484a993c6063e36bc90a47 eca9b8f0f3eae50309eaf901cb822d9b	192.168.50.9 192.168.50.8 192.168.50.6
172.217.10.2	b20b44b18b853ef29ab773e921b03422 334da95730484a993c6063e36bc90a47 eca9b8f0f3eae50309eaf901cb822d9b	192.168.50.9 192.168.50.8 192.168.50.6
172.217.9.226 172.217.11.2	334da95730484a993c6063e36bc90a47 b20b44b18b853ef29ab773e921b03422 eca9b8f0f3eae50309eaf901cb822d9b	192.168.50.9 192.168.50.6
216.58.219.226	b20b44b18b853ef29ab773e921b03422 eca9b8f0f3eae50309eaf901cb822d9b	192.168.50.8 192.168.50.6

Table 4. JA3 Fingerprint calculations for illegitimate candidate hosts

Isolating the JA3 fingerprints used by all illegitimate candidate hosts as a filter in Wireshark showed the same calculation was made in other hosts' connections to the victim network, as shown in Image 1 below. The JA3 Fingerprint "eca9b8f0f3eae50309eaf901cb822d9b" was calculated in 540 different connections from 67 clients to the victim network. The other JA3 Fingerprint calculations observed from

the candidate illegitimate hosts were seen across thousands of connections and from over 100 client hosts. Further research was conducted by searching the online JA3 Fingerprint database accessible at ja3er.com found that JA3 fingerprint "b20b44b18b853ef29ab773e921b03422" showed the fingerprint matched fingerprints of the onion router (TOR) and multiple operating systems such as Ubuntu, Windows NT, and macOS.

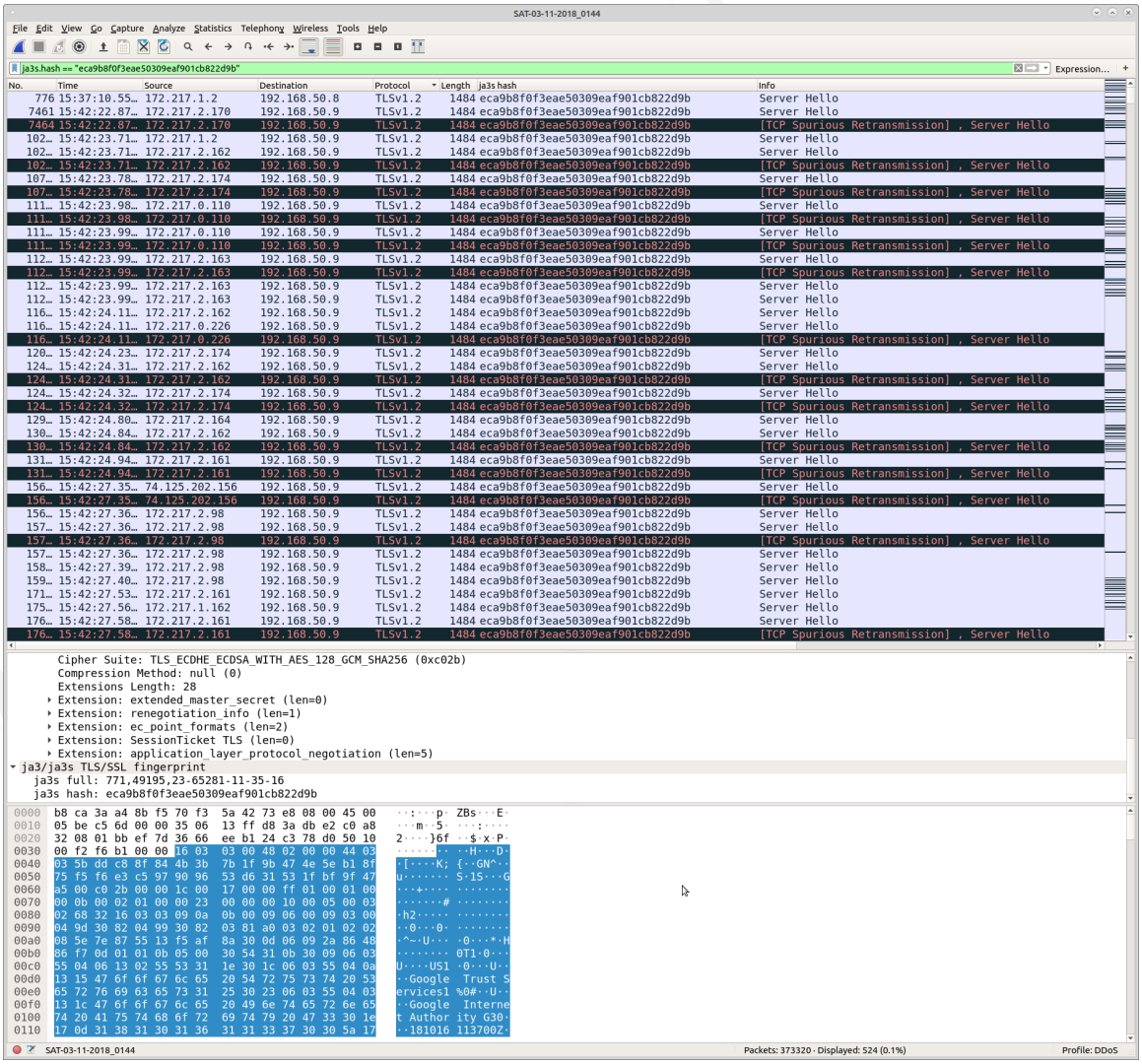


Image 1. Same JA3 Fingerprint calculation across multiple clients.

Isolating the traffic from the candidate legitimate host 13.35.78.80 in Wireshark showed the JA3 fingerprint for this host was "76cc3e2d3028143b23ec18e27dbd7ca9". Filtering by this JA3 Fingerprint in Wireshark revealed the same calculation was seen in

245 different encrypted connections originating from 26 different client hosts. Again, the online JA3 Fingerprint database at ja3er.com was searched for this fingerprint, but there was no information found. Image 2 depicts the details of this JA3 fingerprint shared by the 26 different hosts.

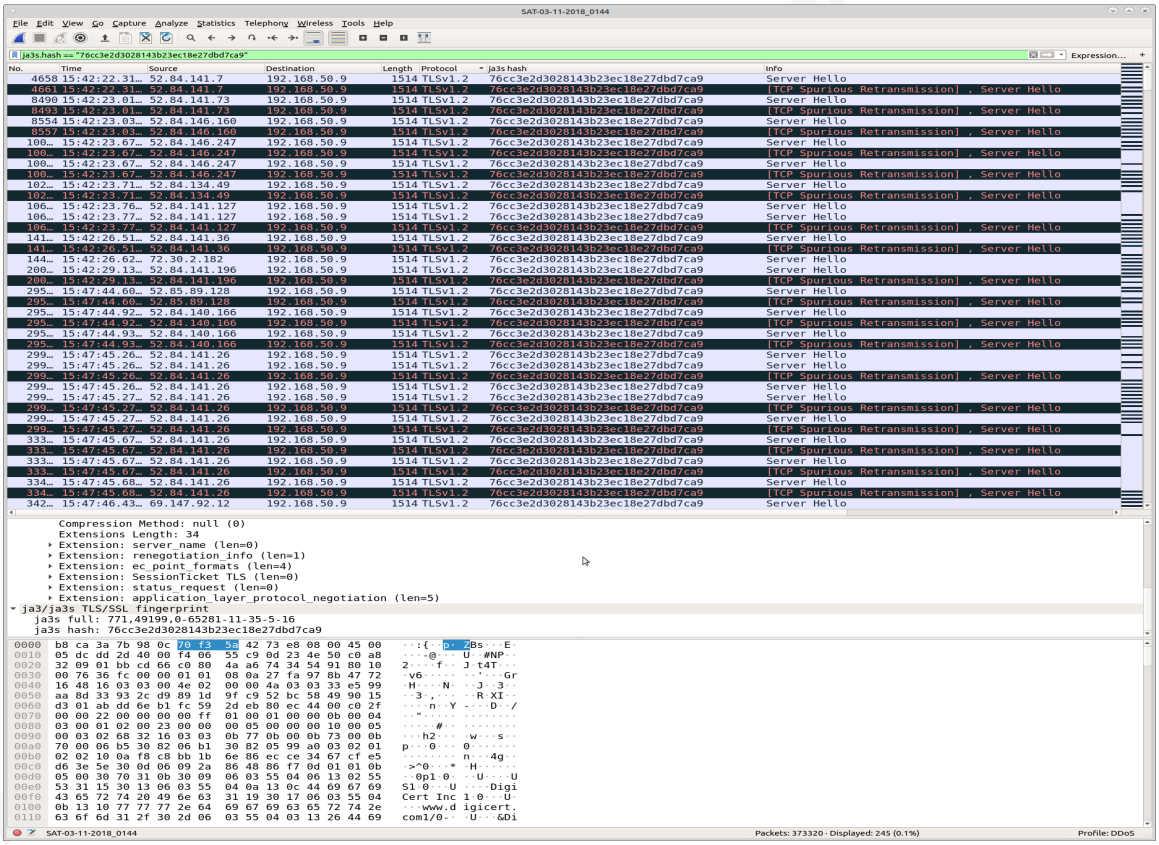


Image 2. Same JA3 fingerprint calculated from different clients.

5. Recommendations and Implications

5.1. Implications for Practice

The combination of the three characteristics, RIS, RFS, and TLS fingerprint, used in this paper resulted as good indicators that security practitioners could use in the layer seven DDoS attack area with some improvements. Security practitioners will benefit by getting to know their clients and users without breaking the trust of TLS encryption by creating profiles they can access and update as needed. Seeing clients or users that use TOR and older technology such as Windows NT may be acceptable for a small number of users, but when those same characteristics are seen in over 50 users with similar

internet protocol addresses tends to raise concerns. Previous research on layer seven DDoS attacks has also shown the lack of entropy in IP locations, a strong indicator of illegitimate DDoS traffic (Zhao et al., 2016, Swaminathan et al., 2014, Tongguang et al., 2013, Siracuscano et al., 2021). Utilizing the information and data used during the TLS handshake may have some ethical and contractual implications for security practitioners. Still, a greater concern would be the storage and access to that data for future use. It is advisable for security practitioners and business leaders to disclose data collection during all internet communications and gain consent from their users before actively collecting data.

Previous layer seven DDoS research has shown that attackers are imitating human behavior to bypass security and corporate security, and the same trend is happening with TLS Fingerprinting. Attackers are masking their TLS fingerprints to avoid detection as they attempt to avoid detection and gain access into networks. In the arena of DDoS attacks, attackers changing or masking the TLS Fingerprint may make it easier to identify illegitimate traffic. Trust is at the center of the TLS Fingerprinting technique, where a client and server will always respond and communicate the same way, and if that changes, then trust is lost. When trust is lost, relationships are broken, contracts are negated, and unity is cast away.

5.2. Recommendations for Future Research

This research paper analyzed two known identifying characteristics of layer seven DDoS attacks and added TLS fingerprinting as a third characteristic. Still, it was not identical to previous research. Liao et al. (2015) used machine learning (ML) and artificial intelligence (AI) to calculate RIS and RFS using an algorithm they called a sparse vector decomposition and rhythm matching (SVD-RM). Applying the same technique in future research that Liao et al. used would be more effective in differentiating illegitimate DDoS traffic from legitimate flash flood traffic. To do this, future researchers would first need to find suitable legitimate and illegitimate traffic so the IA/ML can establish a baseline.

Future research should focus on looking at real-world data where TLS certificates are available so that the content of each request can be analyzed and taken into

consideration. Applying TLS Fingerprinting to DDoS analysis is part of the reasoning for providing more information about who the users and clients requesting web pages are. If researchers can see the content of each request, they can make better-informed decisions about legitimate or illegitimate traffic. Previous layer seven DDoS attack research focused on web page details such as a user requesting the robots.txt file, passing CAPTCHA challenges, hot pages, out of date pages, hyperlink depth, etc. (J. Wang, 2011, Behal et al., 2016, Park et al., 2021, K. Singh, 2016).

Future research could also study the applicability of TLS Fingerprinting with other identifying layers seven DDoS characteristics. Characteristics such as passing CAPTCHA challenges and access to hot pages could be combined with TLS Fingerprinting to increase fidelity in differentiating illegitimate from legitimate traffic. The fidelity of using TLS Fingerprinting alone as a characteristic of identifying layer seven DDoS attacks would also be intriguing for future research.

6. Conclusion

TLS Fingerprinting such as JA3 Fingerprinting can provide valuable insight about the users requesting web pages and web objects which can help security practitioners in their decisions about legitimate and illegitimate traffic. Like most applications in cyber security, the detection and mitigation of DDoS attacks must have multiple filters and multiple tests to ensure high fidelity of results. TLS Fingerprinting helps security practitioners identify layer seven DDoS attacks, but future research should also include mitigation techniques.

Alejandro Aucestovar, aucestovara@gmail.com

References

Althouse, John. (2019). *TLS Fingerprinting with JA3 and JA3S*. Medium. Retrieved September 21, 2021, from https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967

Behal, Sunny, Kumar, Krishan, & Sachdeva, Monika. (2016). Discriminating Flash Events from DDoS Attacks: A Comprehensive Review. *International Journal of Network Security, Vol 19, No 5, 2017, pp 734-741*. https://doi.org/10.6633/IJNS.201709.19(5).11

Defending HTTP web servers against DDoS attacks through busy period-based attack flow detection. (2014). *KSII Transactions on Internet and Information Systems*, *8*(7). https://doi.org/10.3837/tiis.2014.07.018

Gancheva, Z., Sattler, P., & Wustrich, L. (2020). TLS Fingerprinting Techniques. *Network Architectures and Services*, *WS*(19), 15–29. https://doi.org/10.2313/NET-2020-04-1_04

How to detect bot traffic and identify bot threats. Radware Bot Manager. (n.d.). Retrieved September 17, 2021, from https://www.radwarebotmanager.com/how-to-detect-bot-traffic/.

Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A. Ghorbani. (2019) Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy, *IEEE 53rd International Carnahan Conference on Security Technology*, Chennai, India.

Jaafar, G. A., Abdullah, S. M., & Ismail, S. (2019). Review of recent detection methods for an HTTP DDOS attack. *Journal of Computer Networks and Communications*, *2019*, 1–10. https://doi.org/10.1155/2019/1283472

J. Wang, X. Yang, and K. Long, "Web DDoS detection schemes based on measuring user's access behavior with large deviation," In *Proc. IEEE Global Telecommun. Conf.*, 2011, pp. 1-5

Karanpreet, Singh., Paramavir, Singh., and Kumar, Krishan S. (2017). Application Layer HTTP-Get Flood DDoS Attacks: Research landscape and challenges. *Computers and Security*. https://doi.org/10.1016/j.cose.2016.10.005

K. Singh, P. Singh, and K. Kumar, "A systematic review of IP traceback schemes for denial of service attacks," *Comput. Security*, vol. 56, pp. 111-139, 2016

Liao, Qin., Li, Hong., Kang, Songlin., and Liu, Chuchu. (2015). Application layer DDoS attack detection using a cluster with a label based on sparse vector decomposition and rhythm matching. Security and Communications Networks, 8(17), 3111-3120. https://doi.org/10.1002/sec.1236

Park, S., Kim, Y., Choi, H., Kyung, Y., & Park, J. (2021). HTTP DDoS flooding attack mitigation in software-defined networking. *IEICE Transactions on Information and Systems*, *E104.D*(9), 1496–1499. https://doi.org/10.1587/transinf.2021edl8022

Prasad, K. M., Reddy, A. R., & Rao, K. V. (2016). Anomaly-based real-time prevention of underrated app-DDOS attacks on the web: An experiential metrics-based machine learning approach. *Indian Journal of Science and Technology*, *9*(27). https://doi.org/10.17485/ijst/2016/v9i27/87872

Rahman, O., Quraishi, M. A., & Lung, C.-H. (2019). DDoS attacks detection and mitigation in SDN using machine learning. *2019 IEEE World Congress on Services (SERVICES)*. https://doi.org/10.1109/services.2019.00051

Ramezani, A., Khajehpour, A., & Siavoshani, M. J. (2020). On multi-session website fingerprinting over TLS handshake. *2020 10th International Symposium OnTelecommunications (IST)*. https://doi.org/10.1109/ist50524.2020.9345817

Saleh, M. A., & Abdul Manaf, A. (2015). A novel protective framework for defeating HTTP-based denial of service and distributed denial of service attacks. *The Scientific World Journal*, *2015*, 1–19. https://doi.org/10.1155/2015/238230

Singh, K. J., Thongam, K., & De, T. (2018). Detection and differentiation of application-layer DDoS attack from Flash events using fuzzy-ga computation. *IET Information Security*, *12*(6), 502–512. https://doi.org/10.1049/iet-ifs.2017.0500

Singh, K., Singh, P., & Kumar, K. (2017). Application layer HTTP-get floods DDoS attacks: Research landscape and challenges. *Computers & Security*, *65*, 344–372. https://doi.org/10.1016/j.cose.2016.10.005

Sreeram, I., & Vuppala, V. P. (2019). HTTP flood attack detection in application layer using machine learning metrics and bio-inspired bat algorithm. *Applied Computing and Informatics*, *15*(1), 59–66. https://doi.org/10.1016/j.aci.2017.10.003

Suchacka, G., & Iwański, J. (2020). Identifying legitimate web users and bots with different traffic profiles — an information bottleneck approach. *Knowledge-Based Systems*, *197*, 105875. https://doi.org/10.1016/j.knosys.2020.105875

Wei, F., & Yuqin, W. (2020). DDoS attack real-time defense mechanism using Deep Q-Learning Network. *International Journal of Performability Engineering*, *16*(9), 1362. https://doi.org/10.23940/ijpe.20.09.p5.13621373

Zhao, Y., Zhang, W., Feng, Y., & Yu, B. (2018). A classification detection algorithm based on joint entropy vector against Application-layer DDoS attack. *Security and Communication Networks*, *2018*, 1–8. https://doi.org/10.1155/2018/9463653