AI-Powered DDoS Detection Final Report

George Wiedemann, Danh C. Le, Ivan Javier Sosa, Henok Ketema, Daniel deFreese

Introduction

Technology has grown increasingly advanced year after year. New ideas and implementation being released for public use, thus promoting an increase in productivity worldwide, significantly impact developing countries. Though the growth of technology has brought numerous advancements, its potential downfall raises concerns, “how secure is it?” It’s an extremely difficult task to analyze the details of the technology in the question without spending years of research to improvise for its disadvantages and security flaws. In the old days, many internet vulnerabilities were used to disrupt networks and caused harm to many individuals, including loss of organization. Over the course of many decades, the evolution of the current internet infrastructure has undergone numerous stages, accompanied by extensive studies aimed at preventing future security issues. Even so, these vulnerabilities appear to resurface repeatedly when new versions of technologies are introduced to the public. In this report, we will explore a specific type of internet attack known as “*Denial of Service”* or in a large scale, “*Distributed Denial of Service.”* Using our understanding of the attack mechanisms, we will be able to train an Artificial Intelligence to analyze the details of an attack and detect it in a system moving forward.

What’s the problem?

A *Distributed Denial of Service, DDoS,* attack goal is to flood the target server with a large amount of traffic in a short amount of time. *DDoS* attacks can flood the target with many tools, some of these include spoofed IP addresses, a high volume of packets in a short amount of time, and using packets with large amounts of data in them. There are many types of attacks but they fall under three main categories. First is volume-based attacks. These are attacks that rely on the idea of flooding the target with so much data it consumes the bandwidth of the target. Some example attacks that are in this category are UDP floods, ICMP floods, and other spoofed-packet floods. One thing to notice is how all of these attacks end with the word flood showing the goal is to use all of the bandwidth. These attacks are measured in bits per second to show how much bandwidth is being used.

The second category is Protocol attacks. These attacks focus on the use of certain protocols that are vulnerable to a flood of data. The goal is to attack a protocol so that the target’s system stulls, crashes, or hangs. Some examples of this type of attack are SYN floods, fragmented packet attacks, Ping of Death, and Smurf DDoS. These are protocols that when attacked with a flood of data will limit the server's resources and lead to failure of the system. These attacks are measured in packets per second.

The final category of attack is Application layer attacks. These attacks aim to crash the web server by sending more requests than it can handle. Some examples of this are low-and-slow attacks, GET/POST floods, and attacks that target Apache. “Composed of seemingly legitimate and innocent requests, these attacks aim to crash the web server” [12]. These attacks are measured in requests per second because they are not sending packets they are now sending requests.

Why is this problem important?

The use of *DDoS* can be frequently seen increasingly leaning towards online businesses. As of 2020 the percentage of all business conducted online was 25.8%, with a projected increase of about 28.8% by 2024. These businesses depend on network services to allow availability to their e-business site. According to Google nearly 60% of the world's largest 1,000 companies depend on their Google Cloud services[1]In September 2017 a *DDoS* attack cocking in at 2.54Tbps was launched at Google's cloud services. The attack lasted six months, making it one of the largest *DDoS* attacks.[2]

With an almost unlimited access to resources to deliver these attacks from and the difficulties that come in mitigating attacks, the issue of these attacks are quite important. Attacks are performed via a compromised host, consisting of a possible "millions of unprotected computers that access the internet through high bandwidth and always available connections".[3] This essentially means there are unlimited resources for attackers to choose from when conducting their attacks. With the sources(host spanning large geographical areas makes it hard to locate where the attacks were conducted from. Not including spoofing which while attacks done with spoofing are seemingly decreasing, the magnitude and distributed structure of these attacks' traffic make mitigating them extremely difficult.

Recent Attacks

In August of two thousand and twenty three the largest *DDoS* attack started. Nicknamed "*Rapid Reset*" it attacked Google. Two massive waves were detected about 1 minute apart from each other in early October. The largest one being at 392 million requests per second. This attack was also directed at Amazon who wasn't hit as hard. The attack used an exploit in the HTTP/2 Request transfer system. [10]

Since the first quarter of two thousand and twenty four Sweden's application to join NATO has been pending and likely to go through.. They have had a huge increase in Ddos attacks. They had approx. 4.5 million attacks in the first quarter of the year. Many of which exceeded 1Tbps, and 1 recorded instance of a 2Tbps attack. This is very similar to what Finland experienced when they joined NATO. [11]

The Challenges To Detect Attacks

In order to detect *DoS* and *DDoS*, there must be some type of network monitoring system for continuous packet filtering. The main challenges for a system to determine whether a traffic is considered "normal" or "malicious" is dependent fully on the information analysis of the past. In previous years, *DoS* and *DDoS* attacks were used to fill a network or a system with requests to create disruption. But the attacker often needed to hide themselves to avoid trace back, they used the method of IP Spoofing.

The attackers are able to manipulate the disadvantages of Internet Protocol, without authentication themselves, they are able to make many requests and use any arbitrary IP address at their disposal. IP Spoofing refers to, "*creating an IP packet containing fake information." (Tao Peng, 2007, p.7). [8]"* The attacker uses the fake contents in the packet to trick a legitimate system into thinking they need to process such information, this fills the system with a forgery packet on top of normal traffic. When the system is unable to process the request

any longer, caused by a large amount of traffic in the stack, it will shut itself down or “hang” all requests altogether. For the system to work again, the maintainer would have to restart the system with the cost of losing all current requests. In fact, IP Spoofing is just a top level term to describe the attack. The existing variations of this type of attacks are SYN-ACK, Usage of Zombies and Botnets, and Arbitrary Source IP Address.

In our first variation, the malicious intent of SYN-ACK attack is described as, “*exhaust the server SYN queue with spoofed packets, in order to make the server deny any new connection. (Dr.Ibraheem & Sarah Admad, 2013, p.787)*. [6]” When the server is filled with a spoofed SYN request, the SYN stack in the system will be out of bounds causing the system to hang up on all incoming requests. The originator can not be traced back due to IP Addresses being arbitrary. It is a challenge to detect SYN attacks or determine the point of system vulnerability. There are researches that find ways to prevent the attack but there will be trade-offs such that the system will need more storage for large scale production. Dr.Ibraheem’s research suggested that by using SYN-Cookie, the system will be able to note an attack, but it comes at a cost of memory and storage alongside with false positives for denied attacks.

The second variation is Zombies and Botnets. A study defined this attack as, “*gain access to a large number of computers by exploiting their vulnerabilities to set up attack armies (Saman Zargar & David Tipper, 2013, p.2046). [15]”* It is continuing to be an intellectual challenge due to requests in this variation coming from infected systems, viewed as legitimate systems. An attacker can rent or compromise systems with embedded programs to infect other users to take over their system. As a server, receiving requests from these machines will be viewed as legitimate systems, a trace back will not be effective against this variation of spoofing.

By complying with the Internet Protocol and Transmission Control Protocol, the server will have a hard time to prevent an attack and filter out what can be “malicious” and what is “normal.” When an attacker uses different variations of spoofing to attack the server, it’s easy to trace back to the originator if the attacker exposed themselves using their home IP Address. As an attacker, they understand that exposing themselves to vulnerabilities will be risking their professions, and often will hide themselves in a crowd where trace back is almost impossible. Due to the idea, the server is encouraged to continue monitoring their server and prevent any attacker rather than putting the effort into tracing back to the originator, which is almost impossible and a waste of resources. We learned that tracing back is difficult if the attack is using compromised systems.

Besides IP Spoofing, there are other methods such as Amplification attack, which can be combined with IP spoofing to create disruption on a large scale. In a paper by Tao Peng, he mentioned, “*force the target to execute expensive operations [...] exploit this application by sending a large number of queries to a Web site’s search engine. (Tao Peng, 2007, p. 10). [8]*” This research explained that amplification is used as a tool to exploit the system to overload itself in the process of executing too many tasks. Similarly to the DDoS concept, this technique is used to overload the system with hardware requests and hang up to process legitimate requests rather than relying on network overload. Amplification attack can be combined with IP spoofing to attack a target system in many different layers of network and hardware.

In an application based environment, the attacker will focus on creating a disruption in the hardware such that CPU and Memory will be utilized to the maximum. An attacker can execute an embedded program inside the application to create an intensive CPU utilization. The

concept is mainly to target the system to not be able to have any CPU clock cycle left for other tasks in queue that needed to be completed.

In the network based environment, the attacker will focus on overloading the server's connection to other clients. Forcing the bandwidth of the server to be tighter and not accepting any legitimate requests. The paper referred to this as "*Network Traffic Congestion,"* where requests are being reflected off another compromised system directing to the target system with a large bytes size packet. One of the examples of this attack would be ICMP-flooding, ICMP packets does not need any packet header and it's just an echo-request. By sending the echo-request using the target's IP address as the source, the legitimate systems will echo-reply to the source's IP address, performing this on a large scale can cause network traffic congestion. Although ICMP requests can be filtered altogether, there can be different variations where other protocols can be used. To filter out all protocols and determine which traffic is malicious will continue to be a challenge.

Existing Solutions

Traffic Analysis is the simple task of observing the incoming packets and watching the rate at which they come in. The simplest it will observe is a constant rate of packets. It will also detect if there is a scaling amount of packets coming in, either consistently growing or exponentially. Another way that it will look at the traffic patterns is if it picks up at a specific time of day every day. Another way to put it is as a periodic attack. There is a mathematical model that is used to predict whether traffic is malicious or not. It then has a threshold that if hit will consider the traffic as an attack. This method works for TCP, SYN, and HTTP protocols. [5]

Another method is Flow Correlation. Its main focus is detecting between flash crowds and a Ddos attack. It accomplishes this by looking at multiple entry points for a Ddos attack and will analyze them similarly to how traffic analysis does. A flash crowd will also have a lot of deviations from a constant rate of traffic, whereas a Ddos attack tends to have some pattern. Using the Network Flow, Flow Strength, Flow Fingerprint, and the Flow Correlation Coefficient it is able to determine if it is an attack or just a flash crowd. Network flow clusters all of the packets together that are going to the same address. Flow Strength is how long the Network flow lasts, Flow Fingerprint is what makes the Network Flow unique. The Flow Correlation Coefficient is the comparison of two Network Flows. [4]

Limitations for Existing Solutions

While the existing solutions are very good now there are limitations in the existing solutions. In Flow Correlation , It can be disabled by circumventing some conditions.*"Based on our knowledge of current botnets, the above assumptions are applicable in practice. However, attackers may disable our detection method by circumventing some conditions (e.g., the size of live bots) once our strategy is known to them."(Yu, Shui,pg 1076)[4]*. Attackers can also create a super botnet then individual bots within it could emulate the legitimate behavior of genuine users, evading detection. *"First of all, if attackers are able to organize a super botnet, in which the number of live bots is the same or close to the number of concurrent users of a flash crowd, then, one bot can mimic the legitimate behavior of one user"(Yu, Shui,pg 1076)[4].*

In Traffic Analysis If correlation thresholds are set too high or too low it may allow attack packets to go through or regular packets to not go through at all. *“adjust τU too high and τL too low, our detection system may fail, and as a result, the defense system may allow most attack packets to get through. On the contrary, if we adjust τU too low and τL too high, we may confront the DoS condition earlier because most packets would be considered a predictable attack. Since the two thresholds are important, the adjusted values may rely on how much confidence we have.”(Thapngam, Theerasak, pg 352)[5].* Also Traffic Analysis is intended to be used as aids in order to make quick estimates and preliminary process designs, which could lead to inaccurate results. ”*They are intended to be used as aids in order to make quick estimates and preliminary process designs.”(Thapngam, Theerasak, pg 348)[5].* Overall while existing solutions are very good, there are still limitations that can be exposed by hackers.

System Architecture

For our group to create a detection system to train our Artificial Intelligence to determine which traffic is considered as malicious and what a normal traffic pattern looks like. Our research landed us on a dataset created by a simulated environment. Our dataset was produced using the RYU component as a controller switch and in a simulated environment using Python codes to extract all the features, the data set we are using is known as, “DDOS Attack SDN Dataset” completed by Ahuja, Nisha [9]. In this dataset, the traffic was monitored with a Ryu switch, “*a component-based software defined networking framework [...] make it easy for developers to create new network management and control applications. (ryu-sdn.org, ‘What's Ryu’). [16]”* Then in a simulated traffic provided by tools such as *Mininet*, a network traffic was established in a controlled manner. The dataset contained what is classified as malicious from analyzing various packet’s information such as byte counts per packet counts, the flow of the network, the protocols coming from a group of packets in a timestamp, and correlate these packet’s information with the duration of the received packets. By analyzing packets in groups, labeling from the knowledge of DoS/DDoS, ‘label’ is used to classify as malicious or benign. Our group understood that this is a simulated environment, since we do not have a group’s deep knowledge of building an Artificial Intelligence, this would be a good learning curve for us and big enough scope to complete a beginner project.

Since we decided and agreed upon an official dataset, we analyzed the dataset to determine the type of AI classification we need to use to establish the best accuracy result. In a survey studied by Boyang Zhange and Tao Zhang, “*we recommend that Random Forest Tree and Naive Bayes are used to classify malicious traffic and normal traffic for their better performance (B. Zhang & T. Zhang, 2017, p.1278).[14]”* The study was done for detecting malicious traffic in analyzing features such as “... *number of packets, average of packet size, time interval variance, packet size variance, number of bytes, packet rate and bit rate (B. Zhang & T.Zhang, 2017, 1277).[14]”* These features studied fit our SDN dataset which we decided early on. Therefore, we concluded that we will use Naive Bayes as our malicious traffic detection. Though, another survey for Naive Bayes performance completed by H.Setia show in an Advantages and Disadvantages table fashion stated, “*Disadvantages : Significant shortcomings in identifying DDoS attacks (H.Setia & A. Chhabra, 2024, p.3)*.[13]“ The following indicate that we should only build the model to detect malicious traffic rather than identifying the type of

DDoS attack there is. Since Naive Bayes classification is built on Probabilistic, it is less likely it will be able to determine the specific type of attack in our dataset.

As we do further analysis on our features, we have determined to focus mainly on Naive Bayes Categorical classification because our packet's information is sorted in categorical format. The preprocess of data will mainly be formatting the String type of Protocol into columns of One Hot Encoder provided by Scikit-Learn. We will exclude fields that are depending too much on each other and select those that stand independent with our y-axis to be labeled malicious or normal represented by 0 and 1 (for such 1 is malicious and 0 is not). For our packet's source IP Address and destination IP Address, these two fields will be fully removed because categorically, when encoding these String types, too many columns will be added for each unique IP Address in the dataset. In our real world problem, IP Address should have its own topic in determining if an IP Address is considered to be malicious or not. In our case, we are solely focused on determining if a traffic is malicious or otherwise.

Our hypothesis concluded that using Naive Bayes Categorical classification will be the best option according to our research above. Even so, we want to further test our hypothesis to conclude the best result in detection. Scikit-Learn provide us with different Naive Bayes's model implemented for easy usage. We decided to compare Categorical Naive Bayes with other Naive Bayes such as Multinomial, Bernoulli, and Gaussian. We want to measure the time for the model to train (80%) and the accuracy it produced by using our 20% unseen data.

The system specification used to generate the following data is:

- *Intel i9-13900K (24 cores, 32 threads)*
- *Geforce RTX 4070-Ti*
- *Python 3.12*

Model	Time to train(ms)	Score/Accuracy
Categorical Naive Bayes	30,297	96.22%
Multinomial Naive Bayes	16	64.4%
Bernoulli Naive Bayes	15	62.2%
Gaussian Naive Bayes	16	67.1%

From the above benchmark, we can observe that Categorical took the longest to train and to produce results. Although, it produced the most accurate result compared to other models. In a real environment, we would have to take the time it takes to train and produce results into consideration where time is our constraint. For our report, there aren't any constraints on our research; therefore, we will use Categorical to produce the most accurate results for the report.

In a larger scope, instead of detecting malicious traffic or normal traffic, we would want to explore more dataset and different types of models to discuss further on the topic of what type of attack is being performed on a server's traffic. It's difficult to train models without realistic, in an uncontrolled environment, with real time delta of traffics. We can not do further research on

the identifying but to keep the scope large enough for the project, we will only determine if a traffic is malicious or normal in a controlled environment.

Conclusion

There are various methods DDoS attacks can come in (volume based, protocol, application layer). As well as things such as size duration. All this along with our previously mentioned hinerances(e.g. Spoofing, using compromised systems, etc) that come with trying to locate the person/s behind DDoS attacks. When mitigating attacks, differentiating legitimate network traffic from an attack seems to be the problem of contention. Looking over sections of our selected dataset, comparing an attack's duration, byte count and protocol to that of regular traffic attacks are as easily differentiated. In the end we found our Categorical Naive Bayes model was able to detect attacks at what could be deemed as an acceptable rate and shows potential to help combat against DDoS attacks.

References

[1] IEEE Xplore, ieeexplore.ieee.org/Xplore/home.jsp. Accessed 24 Apr. 2024. Customers | google cloud. (n.d.-a). Retrieved from https://cloud.google.com/customers “Google Says It Mitigated a 2.54 Tbps Ddos Attack in 2017, Largest Known to Date.”

[2]ZDNET,www.zdnet.com/article/google-says-it-mitigated-a-2-54-tbps-ddos-attack-in-2017-largest-known-to-date/. Accessed 24 Apr. 2024.

[3] Sachdeva, Monika, et al. “DDoS Incidents and their Impact: A Review.” The International Arab Journal of Information Technology, vol. 7, no. 1, 1 Jan. 2010, pp. 14–22.

[4] Yu, Shui, et al. “Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient.” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 6, 2012, pp. 1073–80, https://doi.org/10.1109/TPDS.2011.262.

[5] Thapngam, Theerasak, et al. “Distributed Denial of Service (DDoS) Detection by Traffic Pattern Analysis.” Peer-to-Peer Networking and Applications, vol. 7, no. 4, 2014, pp. 346–58, https://doi.org/10.1007/s12083-012-0173-3.

[6] Dr. Ibraheem K. Ibraheem, et al. “Enhancement of the Detection of TCP SYN Flooding (DDoS) Attack.” , Volume 19, June 2013, no. 6, Journal of Engineering, https://joe.uobaghdad.edu.iq/index.php/main/article/view/2119/1206

[7] S.T Zargar, J. Joshi and D. Tipper, “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks,” IEEE Communications Sur. & Tutorials, Fourth Quarter 2013, https://doi.org/10.1109/SURV.2013.031413.00127

[8] Tao Peng, et al. “Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems.” ACM Computer Survey. 39, 1, Article 3, April 2007, https://doi.org/10.1145/1216370.1216373

[9] Ahuja, Nisha. “DDOS Attack Sdn Dataset.” Mendeley Data, Mendeley Data, 27 Sept. 2020, data.mendeley.com/datasets/jxpfjc64kr/1.

[10] “Internet Companies Tackle the Biggest Ever Denial of Service Attack.” World Economic Forum, www.weforum.org/agenda/2023/10/internet-cyber-attack-record/. Accessed 1 May 2024.

[11] Fadilpašić, Sead. “DDoS Attacks Saw a Huge Surge in the First Part of 2024, with One Particular Country Badly Hit.” TechRadar, TechRadar pro, 17 Apr. 2024, www.techradar.com/pro/security/ddos-attacks-saw-a-huge-surge-in-the-first-part-of-2024-with-one-particular-country-badly-hit.

[12] Booters — an Analysis of Ddos-as-a-Service Attacks | IEEE Conference Publication | IEEE Xplore, ieeexplore.ieee.org/abstract/document/7140298. Accessed 24 Apr. 2024.

[13] Himanshu Setia, Amit Chhabra, et al. “Securing the Road Ahead: Machine Learning-Driven DDoS attack detection in VANET cloud environments.” Cyber Security and Applications 2, 2024, https://doi.org/10.1016/j.csa.2024.100037

[14] Boyang Zhang, Tao Zhang, et al. “DDoS Detection and Prevention Based on Artificial Intelligence Techniques.” 3rd IEEE International Conference on Computer and Communications, 2017, https://doi.org/10.1109/CompComm.2017.8322748

[15] Saman Taghavi Zargar, James Joshi, and David Tipper, “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks.” IEEE Communications Surveys & Tutorials, Vol 15, No.4, Fourth Quarter 2013. https://doi.org/10.1109/SURV.2013.031413.00127

[16] Ryu-SDN, “What’s Ryu?.” https://ryu-sdn.org