ISSN No:-2456-2165

Mitigating DoS and DDoS based Attacks: An Artificial Intelligence Approach

Jairaj Singh
Student M. Tech (Information Technology)
Birla Institute of Technology
Birla Institute of Technology, Mesra, Ranchi, Jharkhand

Abstract:- DoS and DDoS attacks are one of the most lethal attacks considered in the domain of cyber security. The prime reason behind is that it is easy to conduct such attacks with very limited tools and applications and also with very little effort. The main aim of this attack is to deny the user from the services of the machine. Users and Internet service providers (ISPs) are constantly affected by denial-of-service (DoS) attacks. This cyber threat continues to grow even with the development of new protection technologies. Developing mechanisms to detect this threat is a current challenge in network security.

The topic covers the role of Artificial Intelligence in combating DoS and DDoS type threats in cyber security by identifying DoS and DDoS attacks, learning about the attack and in future preventing such attacks. The agent is being run on a fast hardware, which can process voluminous data at high speeds and can act appropriately to counter the attacks. The AI agent is a combination of both hardware and software solutions and is self-sufficient to counter DoS and DDoS type attacks. This design can prove beneficial as it can replace traditional anti-virus solutions installed on the server as well as on the clients. By loading huge amounts of data into the AI agent and training the machine, the machine itself would be in a process to distinguish between the asset and attack. Section I is an introduction that gives a generic view of computer hacking and introduces AI into the realm of Cyber Security. Section II is the implementation and testing indicating how AI can be implemented in preventing DDoS attack. Section III Generic AI approach to counter some other cyber-attacks and Section IV is the conclusion followed by references.

I. INTRODUCTION

In recent years, distributed denial-of-service (DDoS) attacks have caused significant financial losses to industry and governments worldwide, as shown in information security reports. These records are in line with the growing number of devices connected to the Internet, especially driven by the popularization of ubiquitous computing, materialized through the Internet of Things (IoT) paradigm and characterized by the concept of connecting anything, anywhere, anytime. In most Internet scenarios, devices interact with applications that run remotely on the network, which enables malicious agents to take control of devices. In this way, it is possible to have the interruption of services or the use of devices as a launching point of attacks for diverse domains, as is the case of the DDoS attack, which has been consolidated for several reasons, such as

- simplicity and facility of execution, not requiring vast technological knowledge on the attacker side, and
- variety of platforms and applications for facilitated attack orchestration. Many of these attacks succeeded in disrupting essential Internet services such as DNS, affecting millions of users around the world, and commercial platforms such as the GitHub, prompting severe financial losses to the organizations that depend on those services.

The concept of computer hacking first started somewhere in Australia when a group of hackers were able to get into almost every single computer they targeted worldwide. Hacking although, was popular in the United States before but was mostly done in the form of phone-jacking. **Unix** had become a popular **Operating System** (in 1971) at that time and hackers were mostly using it since it was reliable and could support various forms of internet protocols (OSI layer) with ease. All that was required was to give the right kind of commands to make the system behave the way one wanted.

But technology has evolved quite a lot since then and today security has increased to a great extent. The hackers as well have grown much smarter and with the kind of tools available in the market and with the right kind of hardware they are able to launch sophisticated attacks on large networks, take down large organizations and also cause huge damage to the business. To counter such hackers and their malicious attacks, there are a lot of security protocols that are practiced like the **CIA** triad, the **ISA** (Integrated Security Approach) and many others. There are firewalls installed along at the network level and also there are anti-virus and anti-malware solutions deployed at the client level. But we need to understand that such measures are static and do not have the ability to change their internal mechanisms to counter rapidly changing dynamics of cyber-attacks. For example, if there is a CISCO firewall which has been set to block unwanted ping packets from an unauthorized source, the method is going to work only if the nature of packets remains the same. But this is not the case with the kind of tools we have. Hackers are quite smart in changing their locations on the fly and can also craft the packet intelligently to breach the premises of the firewall. What is required to be understood here is the fact that a **firewall** works on a particular rule and if that rule is

ISSN No:-2456-2165

understood by the hacker by constantly pinging different types of packets on the **target network,** it can be breached ultimately!

Thus, comes the role of **AI** which can boost security in the information domain to a great extent. AI has the ability to understand the data it is trained with and accordingly adapt to changing conditions. If we understand the above scenario by replacing the CISCO firewall , by an **AI agent** that has been trained in some way to block ping packets from unauthorized sources , the AI would be able to sense that the hacker is trying to change his strategies on every **failed attempt,** and would keep learning the way he is trying to infiltrate. In this way the AI would be able to learn almost every technique or creativity which the hacker has been using and would finally be able to block every attempt by the hacker. This is where **dynamics and adaption** have a huge role to play in order to counter constantly varying attacks and such can only be implemented by having a strong AI based cyber security solution.

II. IMPLEMENTATION AND TESTING

Before we go onto understand how AI can be implemented it's important to understand the concept of **seed AI**. Seed AI is the infant stage of the agent (both the program + hardware) where the program is ready to take inputs from the programmer. The concept of seed AI is important because it tries to mimic the human life cycle evolution. The human baby learns by observation and also by different types of input from his parents and later turns into a fully grown experienced and intelligent human who can now act on his own and tackle new problems based on previous knowledge and experience. In much the same way, the concept of seed AI has been used. Although it's a hypothesis, its implementation can turn the AI into a fully developed and experienced machine that would make less and less errors. We can also use a partially or already trained agent in this situation but we prefer using a new agent so that it has trained in the best way possible. In the field of AI **training and testing** are the two most fundamental processes. For both these processes, we have separate data available like test data and training data. Training data is the once using which the agent gets information from various resources and gets an idea about the ways to process it. Once the processing phase is complete, then comes the testing phase where the agent can be tested for its efficiency. The **efficiency** of the AI should be considerably high because it would otherwise give erroneous results. If it is not up to the mark the agent is trained again and again tested till the desired efficiency is reached. Once that is over the AI can be run on production data or in other words it can be deployed to handle real-time data.

Now let's see how an AI can play an important role in tackling DoS and DDoS type attacks.

In a **DoS** attack, the user is denied from the services he is currently using. The attack is mostly meant on the host server in a network that basically kicks out every user who is communicating with it. In this way the server is unable to process any incoming requests from the client machine and the users are denied from accessing the internet. The attack can also take a greater shape, if the network being targeted is done through a network of **"bots" or "zombies"**. These bots are themselves computers that the hackers might have compromised and the network is attacked using the combined processing power of these systems. This is a much dangerous version of the attack because the hackers can easily hide the location of these bot machines. They may use these connections over the TOR network giving them increased anonymity. In order to mobilize bots or zombie machines, the hackers can use the IRC (Internet Relay Chatting) connections used in secure text-based chatting. A lot of hosts can be mobilized in such a way because it's easier to make connections over the secure channel.

If we want to execute a DoS attack, we can use a pen testing platform (Operating System) named **Kali LInux** and use a tool called **hping3.** For a DDoS attack, one can use **SMURF**, which uses **the combined ip traffic of an entire** network to attack the victim. Popular way to use SMURF is to attack the victim using an ICMP echo attack where the packets are initially transferred to the victim and from the victim there is an **echo request** made to the entire network and the network as a response fires back all packets to the victim causing it to get out of service !!.

DoS and DDoS attacks can also be detected by using tools such as **Wireshark** that highlight packets as per the protocol they carry. For eg: if it is a **TCP SYN** attack, the packets would be read as SYN on the application, if it's a ICMP based DoS attack, the packets would be read as ICMP echo request and thereby the user can log them as a malicious DoS attempt. Another popular tool is **Snort** which is a network monitoring tool as well. Its **configuration file** consists of rules that can be modified to include different packets based on the protocol they are using. These protocols can then be filtered accordingly and can be verified for the variety of DoS/DDoS attacks performed.

DoS attacks are also very popular when it comes to ATM services. Hackers are always trying to launch MITM based attacks. Most ATM cards still today use SDA (Static Data Authentication) which shares or verifies the pin with the server in **plain text.** But in most cases when the hacking attempt is unsuccessful, the hackers try to bring down the cash dispensing machine by continuously attacking with ping packets on the server (in other words DoS) and bringing it down.

ISSN No:-2456-2165

DoS attacks can be prevented in many ways. The most fundamental way is to block the ping packets from any unidentified or spoofed IP address. The servers have a firewall that has a list of allowed IP addresses on a local file. These servers make use of this file in order to allow ping packets or reject them. But this method which although looks simple, does not work very well because the hackers can then use better methods like spoofing the IP address, crafting the ping packets in such a way that it reaches the targeted server. Therefore, we need to do better such as implementing traffic surge and congestion control algorithms to finally reach the **IP/hacker** that caused flooding.

➢ *Traffic Surge Algorithm:*

As the name suggests, this algorithm constantly monitors the packets that leave and enter a particular network. This algorithm works on a real-time basis and keeps track of all IPs registered within the network. If there is a sudden packet increase in the network more than the calculated threshold packet limit, then the surge alarm is set and the algorithm tries to locate the packet/packets that crossed the threshold limit. Below is the algorithm generic representation:

➢ *Traffic Surge (struct node * IPAddr, Boolean surge)*

I. Assume a network where the attack is going to take place storing the IP address of the server and all clients in a suitable data structure (IPAddr) that does not increase the space-time complexity of the algorithm.

- Monitor the incoming and outgoing packets of the network and calculate the max packet count (incoming + outgoing).
- Map the total number of IPs and the total number of packets exchanged.
- If the packet limit increases, set surge to "**True**", indicating that the max packet count has been violated.
- Once the surge has been determined, filter the packets as per the registered IPs of the network.
- Since the packets and IPs are mapped, all registered IPs can be separated from the ones not registered or intrusive IPs.
- Return the unregistered or **intrusive IP** along with its hopp.

The steps above give a generic picture of how a traffic surge algorithm works. Next step is to train this algorithm by running it on a considerably fast hardware (**a NVIDIA GPU** would suffice) using **ANN (Artificial Neural Network).** The layers of the neural network can be tuned as per the limitations of the hardware and also without violating the efficiency of the output.

ANN is a neuron-based model which tries to mimic a human brain and its cognitive behavior. The inputs are given into the neurons. Each neuron is assigned a certain weight that changes to match the efficiency of the output. The output is already known to the system from beforehand (**a supervised learning model)** and the **neurons** change their weights to match the efficiency of the known output. So, the output in our training model would be to keep the total packet exchange within the limit. If our ANN is able to keep the limit under control (the number of packets under control) then we can say that our ANN is successful in countering the DDoS attack. Now we design the input to be given into the AI.

The input given to the seed AI should comprise of the following:

- The different types of packets under consideration (**both incoming + outgoing**)
- The protocols which are getting targeted for example: **TCP, UDP and ICMP.**
- The attack vectors or the paths which the hacker can take to DDoS onto the target. For instance, if the hacker has some IP then the entire route which the packet takes would be the attack vector.
- The types of assets (client's computers + servers) that can be attacked should be considered as well.

The above steps are basically the anatomy of a DDoS attack and by giving such input to the seed AI would train it to classify the output whether the data being generated is out of bounds and the network has been attacked or whether the data is within the limit and the network is safe from DDoS.

➢ *Testing:*

In the testing phase, the seed AI is going to counter DDos attacks which would be carried out by ethical hackers and authorized pen testers from networks outside the main target network. The verification would be done on the following guidelines:

- How efficiently does the seed AI manage to detect packets for DDos?
- How efficiently is the seed AI able to detect the affected protocols?
- Is the seed AI able to recognize the entire attack vector right from the source (hacker) to the destination (target network)?
- Is the seed AI able to list down or recognize the target computers being brought down by the DDos attack and block ping packets?

In the testing phase initially, the seed AI is going to make a lot of errors because the data being generated by the network is very large and secondly due to the diversity within the inputs being given to the neuron. Therefore, every neuron would need to calculate the result based on every part of the input and in doing this it might tend to miss out on certain specifications (mentioned in the input section above). As a result, the expected output of filtering out the correct IP/sets of IPs leading to the DDoS attack can lead to erroneous results. But this problem can be fixed by regularly training the algorithm and modifying it based on the errors. It's obvious that the **fundamental logic** behind the traffic surge algorithm remains the same but certain variations might be required to reach the final result efficiently.

ISSN No:-2456-2165

These are the factors upon which the efficiency of the seed AI relies. In order to increase the efficiency of the agent, we might want to increase the following:

- Increase the input data by increasing the parameters within the input and then train the AI algorithm.
- Use better attack methods which the AI is not aware about and then by **back-propagation** principle on **ANN** rectify the weights to accommodate the correct output.
- Use of different attack vectors, for instance a group of pen testers can use a **TOR network** for a **"bot" or "zombie"** attack. In this way, the AI would initially not be able to separate the unregistered IPs / intrusive IPs because it won't be able to detect the hidden nodes. In order to counter such an attack, the algorithm can simply be modified with a condition to exclude all packets being generated from hidden nodes and thus in this way the massive number of packets can be reduced within the initial packet limit thus avoiding a massive and sophisticated DDoS attack.

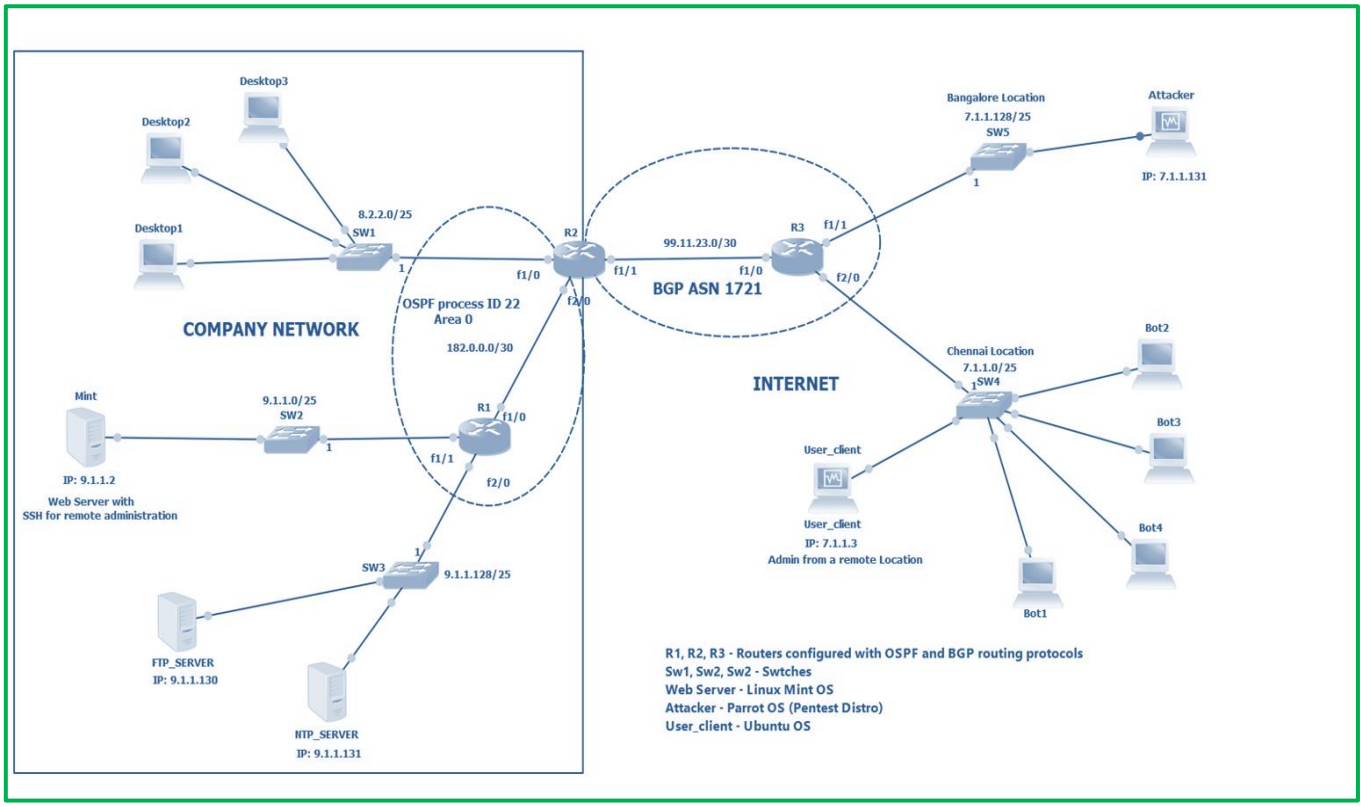

Fig 1:- A generic diagram of a corporate network mitigating against a DDoS attack
(courtesy: securityintelligence.com/bumper-to-bumper-detecting-and-mitigating-dos-and-ddos-attacks-on-the-cloud-part-2)

III. GENERIC AI APPROACH TO COUNTER SOME OTHER CYBER ATTACKS

As we can see that a DDoS attack can be prevented or mitigated using a seed AI, it can also be generalized to counter other cyber-attacks as well. Other attacks such as **XSS scripting, MITM (Man in the Middle Attack), Cryptographic Cipher attacks can also be mitigated using an AI.** But for that we need to design a generic algorithm with different use cases. The use cases can be designed on the basis of different categories of cyber-attacks and threats. The generic algorithm can then be used to branch to different cases automatically or by adaptation of the AI depending upon the type of intruder packet which got detected.

The algorithm would be designed for every attack and then would be called by the main generic code, in much the same way as in simple programming a main function calls the other methods. We can take up an example to highlight the above clause. Let's say that we are interested in an XSS type attack. So, to design the algorithm we need to first list down the steps needed to actually perform the attack on the victim machine. For instance, we need to first open the browser, find a link or a URL that we want the user to open, try to find a JavaScript vulnerability that would allow us to inject our XSS backdoor, and then use some of social-engineering skills to attract the user in order to click that infected link and ultimately cause an attack on the machine . Once this is done, we can automate the process, let's say by writing a relevant program in Python. We can use this program as a library or a module in the main AI program and call this whenever we want to detect and defend such an attack. This is probably the simplest method where we can have control over the AI as we know how the attacks are performed manually. The input part for XSS can be as simple as having different URLs to infect and the rest can be handled by the XSS program. The AI can also detect whether an XSS attack can be performed in the same way. Because we know the steps to launch an XSS attack, we can have a filter in the detection algorithm for every step. As soon as the AI detects any of the steps in the XSS

program, it would raise an alarm and abort the packet of the intruder. This goes to prove that use of AI has the potential to mitigate any type of cyber-attacks with ease.

IV. CONCLUSION

The process of detecting and mitigating any **DoS/DDoS** based cyber-attacks are usually carried out by industries by using static tools such as Snort, Wireshark etc. But using AI as we have seen above, we can really make the AI smart enough to detect and mitigate DoS/DDoS threats all by itself. The AI design behaves much like a human cognitive behavior. Once the AI has enough experience it can become much more powerful than normal applications and software available.

With the invention of AI and AI based algorithms, the fate of technology is changing. The idea behind Artificial Intelligence was to make machines think like humans and ultimately bridge the gap between man and machine. But there are issues like if the AI gets smarter than a human, there would be chances that the machine starts to dominate and the sole purpose of making our jobs easier would come back haunting at us. Machines designed or programmed to control cyber threats would start attacking our own organization instead of defending our infrastructure. Therefore, we need to have control over our technology and use it to our advantage. To ensure that the AI does not go out of control, we need to be careful while training the algorithm. The kind of data we input plays a crucial role in how the AI is going to function. With the era of **Super-computers** and **Quantum-computers** which would be in the business shortly, AI can be lethal if technology is not regulated. There have been instances where AI has managed to cause harm, for example in the case of accidents happening in autonomous cars. A few AI machines used in medical science happen to give the wrong diagnosis and completely horrific results that could prove fatal. But AI, nonetheless proves a promising future not only in the field of Cybersecurity but other areas as well. In the field of medical science, AI can work wonders. With the recent outbreak of Coronavirus, efficient use of AI and Machine Learning can help discover the suitable vaccine and thereby cure the infected people. In the domain of Robotic surgery as well, AI and 5th Generation wireless technology can do wonders with the amount of precision the machines can achieve.

REFERENCES

[1]. Tod Lamble, “Cisco Certified Network Associate”, Sixth Edition, pp.13-31
[2]. Keith Barker, Scott Morris, Kevin Wallace, Michael Watkins, “CCNA Security 640-554”, pp.221-2
[3]. Nick Bostrom, “Superintelligence-Paths, Dangers and Strategies”, pp.22-50
[4]. Congyingzi Zhang, Robert Green, “Communication Security in Internet of Things: Preventive Measure and Avoid DDoS Attack Over IoT Network”
[5]. Enn Tyugu, “Algorithms and Architectures of Artificial Intelligence”, IOS Press, pp.79-84
[6]. Anton Rager, Seth Fogie, “XSS attacks Cross Site Scripting Attack and Defense”, Synereses Publishing, Inc, pp.67-90
[7]. Srikanth K Ballal, “Bumper to Bumper: Detecting and Mitigating DoS and DDoS Attacks on the Cloud, Part 2”, covered in web link securityintelligence.com