Smart Shield: Real-Time Intelligent DDoS Detection and Mitigation

Omkar Gade[1], Shreeja Gundlur[2], Abhishek Wagavekar[3], Aditya Borawake[4], Prof. Rohidas Sangore[5]
Computer Science and Engineering, MIT ADT University Pune, India

***Abstract*—The SmartShield system is designed to identify and mitigate DDoS attacks using real-time, intelligent filtering and detection techniques while monitoring network traffic. The configuration utilizes three virtual machines, including Kali Linux as the attacker, the victim running on an Ubuntu Server, and the monitoring and response unit on an Ubuntu Desktop. SmartShield also utilizes packet analysis with Wireshark[11], intrusion detection with Snort, and automated banning of malicious IPs via Fail2Ban[2]. A simulated DDoS environment is created by traffic HOIC or hping3, which is then sent over the network. The system captures data packets, analyzes them for anomalies using Snort[6], and mitigates the attacks using fail2ban policies[2]. The practical implementation of this architecture showcases a very affordable hands-on approach to teaching real-time network cybersecurity along with intrusion detection and mitigation within controlled environments [1][6][12].**

***Keywords*—SmartShield, Wireshark, VirtualBox, DDoS Simulation, Snort, Fail2Ban, Network Security, Ubuntu Server, Intrusion Detection, Kali Linux.**

I. INTRODUCTION

The rapid proliferation of interconnected systems and services in the modern world has transformed communication, commerce, and information retrieval. However, this connectivity renders infrastructure vulnerable and increases the potential points of attack for threat actors. One of the persistently impactful threats to networked infrastructure remains the DDoS attack (Distributed Denial-of-Service)[5][8]. Strategic attacks of this nature are not aimed at gathering information, rather they seek to make services unavailable by perpetually flooding server systems which renders the systems out of reach for users who are authorized. These sorts of cyber assaults do not require intricate infiltration or taking advantage of system weaknesses, instead they are DDoS attacks that are simplistic in nature and can be executed easily which leads to unprecedented destruction. Everything from degrading services to total halting of critical system operations is possible with, making them . All hail the almighty DDoS attack! Attacks are an unrivalled tactic for hacktivists, cybercriminals, ... even state-funded hacktivists. In the past couple of years, the number and severity of DDoS attacks has DRASTICALLY increased because of the thousands of IoT devices and botnets [5][9] available for use, therefore raising the need for instantaneous intelligent defences. To address this newly emerged challenge, we propose Smartshield, a new intelligent system that detects and alleviates DDoS attacks in real time. Unlike typical static defense systems that depend on outdated signatures and delayed action systems, Smartshield staticehncsmrsmrmershshsgfmgrs Nothing. Offers dynamic, proactive, and flexible defense methods. Combines behavior bashed on system monitoring, real-time network traffic, and automated firewall rule update to detect malicious activities and remove them before they cause serious damage. Smart shield is designed to work in a virtualized lab, which allows for flexibility in classroom settings and practicality in real world testbeds. The architecture utilizes a collection of freely available software, such as tcpdump for packet capture, iptables for controlling cross level access to the network, and bash or python scripts to trigger autotomatic commands (for detection and mitigation), to achieve the lowest response time to any given threat while remaining highly adjustable to new patterns of attack[1][12].

II. LITERATURE REVIEW

Signature-Based Detection Methods: Classic Intrusion Detection Systems (IDS) like Snort and Suricata are built on a signature detection framework where incoming traffic is checked against known attack signatures. These systems, however, are only effective against previously discovered threats and are incapable of recognizing zero-day or variable attacks that do not conform to expected patterns. In addition, defensive

mechanisms in these systems are real-time, but because signature updates are manual and usually lag behind, protection is rendered inadequate. [7]

Anomaly Detection Systems: This type of detection has gained considerable attention using statistical or machine learning because of the need to detect anomalies. Algorithms such as Support Vector Machines (SVM), Neural Networks, and Decision Trees have been employed to achieve this. [7][10]

DDoS Mitigation using Software-Defined Networking (SDN): Traffic Management and mitigation techniques have been actively implemented using Software-Defined Networking (SDN) in several recent works. SDN decouples control and data planes which facilitate central control and offer agile routing to mitigate DDoS attacks. [8][9]

Mitigation Using Firewalls DDoS: protection mechanisms generally include the use of firewalls and specific rules made in them, for example, in iptables under Linux. Firewalls ban certain IP addresses, specific protocols or even certain volumes of traffic. As with all attempts at structuring firewall rules while under attack, doing this step manually is both tedious and prone to unexpected complications.[4]

Simplistic Mitigation Strategies Through Scripting: For academic lab scenarios or other limited scoped setups, some researchers have applied tcpdump along with custom Bash/Python scripts to track traffic and pinpoint abnormal packet movements. These systems are flexible and efficient with system resources.[3][13].

III. METHODOLOGY

Smart Shield functions as a real-time Protective System for Intelligent Detection and Mitigation of DDoS attacks employing traffic analysis at the system level using automated traffic banning. The revised methodology incorporates Fail2Ban, a log monitoring and banning program, thus making the automated mitigation process more efficient. This enhances the robustness, scalability, and ease of management of the system while still achieving real-time responsiveness. [2][3][11][13]

1. Setting up the system and installing required tools:
A virtual lab setup creates a simulated environment consisting of a client, an attacker, and observation nodes.
An Ubuntu distribution is selected as the host operating system to facilitate command-line operations as well as the use of Fail2Ban.

The following tools are installed:
For live traffic analysis, tcpdump and wireshark are installed.
For IP banning automation, Fail2ban is installed to monitor logs.[3][11][2]

2.Traffif monitoring with tcpdump: Tcpdump operates nonstop to fetch the incoming traffic on ports or interfaces. The information comprises of: Source IP and Destination IPs ICMP, SYN, etc. are some examples of PDU types Packets' frequency As for capturing and logging the data, System will analyze the text data and save it in a ddos.log. So we parse and log this file perfomed (captured data) this log will be monitored by Fail2ban.[3]

3. Log File Creation and Formatting: Wireshark does real time monitoring of the data that comes from the tcpdump. The script uses the "Fail2Ban compatible format" to log the suspicious activities that would be over 100 requests from a single IP in a minute, to avoid log file bloat.[2]

4. Actions Against Attacks using Fail2Ban: Configure Fail2Ban to check ddos.log and punish violators based on a predefined filter. Identified patterns or multiple alerts from the same IP will be acted upon by: Initiating a ban action Set iptables to deny all incoming request from the provided address. This action will continue until the defined duration for logs, disable and retry, is reached. All parameters for configurable to the fail2ban configuration. Banning duration can also be set. [2][4]

5. Always On Monitoring and Recovering Countable Events Spam: Smart Shield do automatic: Traffic Behavior logging Suspicious behavior logging Banning or unbanning through Fail2Ban log decision logs User defined or timed unban Smart Shield automated features that run continuously include traffic behavior typing logging.[2]

IV. ARCHITECTURE AND IMPLEMENTATION

SmartShield has a modular architecture that is streamlined for real-time DDoS attack detection and mitigation in automated enterprise or virtual lab settings. It is composed of traffic monitoring, intelligent detection systems, log-based event generation, and automated mitigation via Fail2Ban. [6][1][12]

Primary Components:

1. DDoS Attack:
Tool: ddosify
Execute a DDoS attack

2. Traffic Capture Module:
Tool: wireshark, tcpdump
Collects live network traffic from controlled environments.

3. Custom Log File:
File: ddos.log
Contain alerts on suspicious IP accumulative ban in Fail2Ban formatted readable logs.

4. Fail2Ban Engine:
Analyses ddos.log through tailored filters.
Upon parameters' fulfilments, applies predefined rules to block source IP using iptables.

V. CONCLUSION

Smart Shield's response delay DDoS mitigation arms the defender with speed and flexibility against attack DDoS attack strategies, shifting the balance of power towards the defender's advantage. Incorporating simple tools like tcpdump, Python scripts, alongside the intelligent features of Fail2Ban makes the system economical and easy to deploy, all while using basic resources to protect critical infrastructure from highly prevalent and disruptive attacks. The model streamlined principles translates into ease of use makes SmartShield especially appealing for small to medium-sized business (SMBs), education and research, as well as cyber training environments. We The architecture allows for effective protection without requiring supervision, real-time responsiveness and resource efficiency. SmartShield demonstrates the application of basic cybersecurity principles such as live traffic monitoring, log analysis, and IP banning in a practical system.

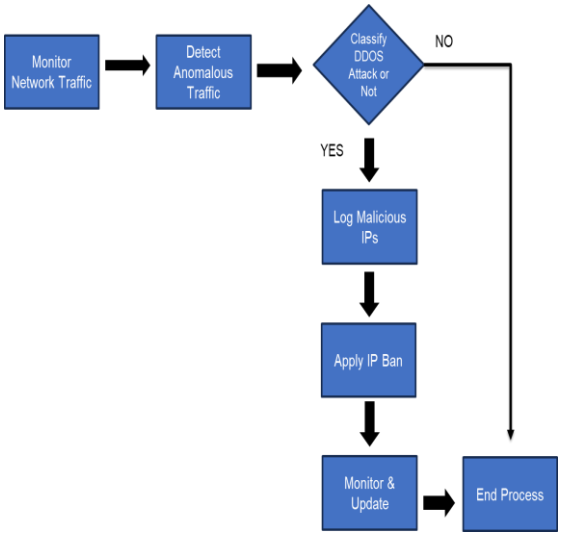

Fig 3.1 Working of the project

VI. RESULTS

Fig 6.1 Ubuntu server DDoS attack victim machine

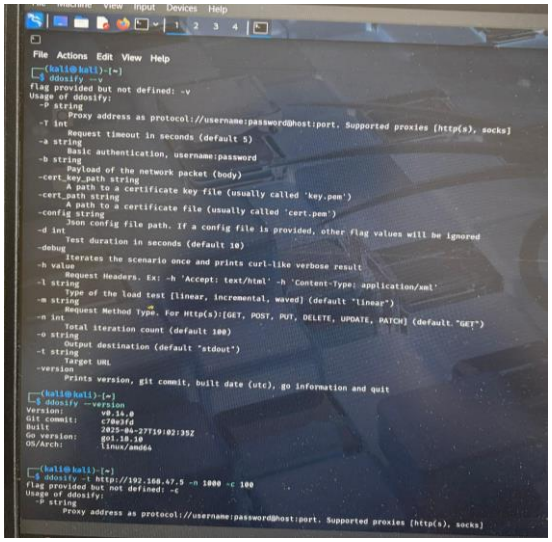

Fig 6.2 Kali Linux attacker machine for simulating ddos attack

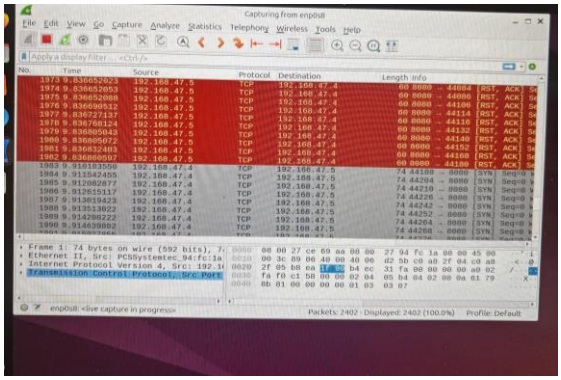

Fig 6.3 Wireshark Network capture

VII. ACKNOWLEGEMENT

This is where we convey our respects to Pro VC DR. Ramchandra Pujer, Dean Dr. Rajneeshkaur Sachdeo, Director Dr. Vipul Dalal, Head of the Department Dr. Ramesh Mali, and extend appreciation to the team members for accomplishing the milestone as it could not have been done if we did not fully depend on all of us together.

We appreciate the provisions extended MIT ADT University, Pune in respect of School of computing, other unidentified facilities along with the immense support extended by the family and friends which has been imperative in achieving the project objectives.

REFRENCES

[1] Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). NIST. https://nvlpubs.nist.gov

[2] Fail2Ban Project. Documentation. https://www.fail2ban.org

[3] tcpdump/Libpcap Project. Official Documentation. https://www.tcpdump.org

[4] Linux iptables Manual. The Netfilter Project. https://netfilter.org

[5] Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms. Computer Networks, 44(5), 643–666.

[6] Roesch, M. (1999). Snort - Lightweight Intrusion Detection for Networks. USENIX LISA. https://www.snort.org

[7] Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against DDoS flooding attacks. IEEE Comms Surveys, 15(4), 2046–2069.

[8] Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and defense mechanisms. ACM SIGCOMM CCR, 34(2), 39–53.

[9] Sommer, R., & Paxson, V. (2010). Using ML for Network Intrusion Detection. IEEE Symposium on Security and Privacy.

[10] Choraś, M., & Kozik, R. (2013). Machine Learning Techniques for Intrusion Detection. Springer.

[11] Wireshark Foundation. Wireshark User's Guide. https://www.wireshark.org/docs/wsug_html_chunked/

[12] OWASP. Denial of Service (DoS) Attacks. https://owasp.org/www-community/attacks/Denial_of_Service

[13] Kreibich, C., & Crowcroft, J. (2004). Honeycomb – Intrusion Detection Signatures Using Honeypots. ACM SIGCOMM CCR.