JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

COMBATING DDOS ATTACKS WITH TURING TESTS: A FRAMEWORK FOR ENHANCED NETWORK SECURITY

Sonali Bharat Shirke
PhD Scholar
Department of Computer Sci. and Engg
Kalinga University

Uruj Jaleel
Professor
Department of Computer Sci. and Engg
Kalinga University

Sandeep Kadam
Professor
Department of Computer Engineering
Keystone School of Engineering

Aparna Atul Junnarkar
Associate Professor
Department of Information Technology
Vishwakarma Institute of Information Technology.

ABSTRACT

Distributed denial of service assaults, often known as DDoS attacks, continue to pose a significant risk to the reliability and safety of websites and other online platforms. An technique that utilizes CAPTCHA-based Turing tests and cache-based monitoring systems is proposed in this research as a multi-faceted method for identifying and combating distributed denial of service assaults (DDoS). The framework provides the early identification of harmful activity by classifying network traffic into blacklist, pinklist, and whitelist categories. This configuration enables the framework to take prompt action against sources that are deemed to be suspicious. According to the findings of the simulation, the use of this framework results in increased throughput, a higher packet delivery ratio, and a reduction in both packet loss and latency. With the use of sophisticated detecting modules and algorithms, the purpose of this study is to develop a system that is both scalable and effective in reducing the damage that is caused by distributed denial of service assaults.

KEYWORDS: *Blacklist, DDos, Algorithms, Simulation, Pinklist, Phishing*

1. INTRODUCTION

In a typical online assault known as "phishing," the perpetrator pretends to be a trustworthy entity in order to trick victims into divulging sensitive information such as login credentials, passwords, and credit card details. In order to steal sensitive information, attackers often pose as legitimate businesses, government agencies, or social media platforms. Email and SMS messages are common vectors for phishing attacks. By using a technique that is very similar to the genuine thing, phishing encourages users to input sensitive information on a fake website. [1]

Effective and financially motivated breaches that steal customers' confidential data and confirmation certificates are known as phishing assaults. Not only do they cause substantial financial damage to individuals, businesses, and financial institutions, but they also erode customers' faith in online shopping. [2]In 2007, 3.6 million people in the US fell victim to phishing attacks, which cost over $3.2 billion USD, according to researchers from Gartner. In 2006, e-commerce and managing account exchanges lost two billion USD due to customer anxiety about web security. Phishing assaults, despite several attempts at defence, have been steadily increasing in both scope and sophistication. Between June 2006 and April 2007, the number of exposed phishing sites increased fivefold, going from 10047 to 55643. The actual number might be much higher because not all recent phishing attempts have been detected and documented. These include intentional setting phishing attacks, malware-based phishing attacks, and ongoing man-in-the-center phishing attacks against one-time passwords. [3]

Phishing is a common tactic for cybercriminals to get sensitive information like bank account or standardised savings account numbers. A typical phishing attack involves a cybercriminal creating a phoney website that mimics the login page of a legitimate financial institution. Then, the criminal sends out an overwhelming amount of emails in an attempt to trick victims into entering their personal information on the phoney site. Offender acquisition costs are minimal, and they can cover their tracks and complete an assault cycle in a short amount of time. Phishing assaults have grown in popularity because of these certainties. [4]

Typically, when people think of a network assault, they picture someone interfering with a network's foundation in order to steal information or exploit existing vulnerabilities, such as open ports or unauthorised access to assets. [5]

Managing an aloof assault occurs when the goal of the attack is only to collect data from the framework, without modifying or harming any of the framework's assets. [6] When an attacker has access to sensitive data or assets, they might launch a dynamic assault by changing the handicaps or destroying them. An outside attack is one that an unauthorised element may launch from outside the organisation. And the same thing may happen within an organisation by a "insider" who already has significant network access; this kind of attack is called an indoor attack. [7]

At all times, the network crashes into itself and becomes infected with malware that targets certain frameworks. This research will include assaults like phishing and social engineering that target end users. [8] Although they aren't often called network assaults, they must be considered for fulfilment reasons and because they are almost limitless. There are a variety of ways in which network assaults may be structured, depending on the techniques employed or the vulnerabilities exploited. Only the most well-known and far-reaching forms of assault are covered and shown here. [9]

One kind of cyberattack is known as "phishing," and it aims to steal sensitive information from unsuspecting internet customers, such as login credentials and financial details. When an attacker uses email or a website page to pretend to be a trustworthy entity, they are engaging in phishing. By using spoof emails, instant messenger/social media, or other ways, exploited individuals are directed to fake internet pages that seem legitimate. Email spoofing is one common tactic used to make communications seem to have come from a legitimate sender or to hide the true website behind a lengthy, complicated subdomain. According to RSA, an insurance company, $1.5 billion in losses in 2012 were attributable to phishing. [10]

2. RESEARCH METHODOLOGY

- **Proposed Work**

1) Text Based Turing Testing

The bundle originating from the starting place is initially come to the Source Monitoring and Counting phase where the source is tested and confirmed. In the event that the starting place is unapproved or suspected the parcel is coordinated to the Text-Based Turing Module. In this, the source is tested by reacting to the CAPTCHA showed to proceed with the association. The Text Based Turing Module is upheld by the CAPTCHA generation module by intermittently redesigning the inquiry pool. The DDoS Attack Detection Module keep running in collaboration with Source Monitoring and counting module to discover any conceivable DDoS assault. The Source Checking and Counting Module note down the entry point and important data for DDoS recognition (Dou and Wanchun, 2013).

2) Cache Based and Question Generation

The Source Monitoring and Counting Module is where the client-side bundle will reach its foundation, after which it will be certified. Clients are sent to the Cache-Based Turing Module in the event that they exhibit suspicious behaviour. “The Cache-Based Turing Module verifies the client by accessing the client's temporary data stored in the client's framework. To identify more Denial of Service attacks, the Detection area will be used. All of the critical information for attack identification is remembered by the Source scrutiny and Counting Module. Furthermore, DDoS prevention makes use of the Question era module.

A) Source Monitoring and Counting Module

It is a module that another module uses as a facilitator. Monitoring Module for Sources It is the responsibility of this module to sort packages according to their state. For the other modules, this one acts as a coordinator. Using this module, bundles are organised into the following list:

- **Black list:** Here, the client's location is verified by the Source Checking Module. If it's in the boycott database, it will block the shipment associated with the specified client's address. Otherwise, the package will be sent to the pink or white rundown.
- **Pink list:** This part of the process involves running the Cache-Based or Text-Based Turing Test on the packages once more. Considering the reserve data, it will investigate the suspiciousness of the bundle. If the package detects anything fishy, it will be sent to white rundown for further investigation.
- **White list:** This summary only stores client locations that have been accepted following full confirmation by the Turing module.

Counting Module

The location of the source and destination bundles are stored in the tallying module. The entry time of the request is also stored. Debilitation is the default way of module numbering. The quality of any suspicious package is changed to enable from incapacitate if the DDoS Attack Detection Module detects one. At intervals, the quality is reset throughout the tally phase.

B) DDoS Attack Detection

The main objective of this part is to identify potentially malicious sources and report them to the boycott vault. The Turing Module verifies the provided source by running tests on it to determine the inquiry's validity. It goes through the following processes to identify the questionable source, as listed below:

Stage i: Here, the recognition module acts as a screen mode, often gathering data as usual and estimating the association/approaching bundles/approaching bytes per second, as well as recognising the source activities. In order to identify the malicious source, the stored data relates to all VC (Virtual Controller) operations.

Stage-ii: At this point, the process from Stage-i is still feverishly compiling the current VC movement data for identifying the malicious source. At this stage, the attack discovery module

evaluates the value of the present movement relative to the one from the past measurement for each virtual controller. The identification state advances to Stage-iii, and the Counting Module is able to count the approaching movement of the particular virtual controller, if the current activity worth is more than the prior measurement one.

Stage-iii: Four fundamental parameters are utilized which are given underneath:

- **Only the most severe edge value is considered by TH:** Threshold. This characteristic may serve as the foundation for the relationship between the client and virtual controller.
- **NUM Period:** When the sum of all client-sent packages exceeds the specified threshold, an upper limit is imposed in this section. In such a case, the DDoS Attack Detection Module added the specific IP address to the Pink rundown database. Afterwards, the validation area is completed by the Turing Module that is supported by Cache-Based or CAPTCHA.

If the amount of association time is more important than MXTH, it is also an edge quality that is set. Assuming its quality is 90% of Apache's Server execution or TH, the certain IP location is simultaneously linked to the Pink rundown database under these conditions.

If the amount of IP source association is more than maximal, an additional edge value called Node TH is set. In this scenario, the framework swiftly changes 50% of the IP address associated with the Pink database. In order to avoid a framework crash caused by a virtual controller blockage, the specified region must be completed. In the event that no IP address is included in the pink list for the specified NUM Period value, the DDoS Attack Detection Module status will revert to Stage II, and the counting module will become inoperable. (Huang, V. S, 2013).

Stage-iv: Here, 90–95 percent of the virtual controller's incoming and outgoing system transfer speed is consumed by the enormous amount of activity going to and from the virtual controller. Under these conditions, any investigation can lead to a system crash or slower performance. Consequently, in order to avoid this situation, we have prevented the client-side HTTP associations by adding the global public IP address to the destination piece list. The IP address of the virtual controller is added sequentially and prevented from accessing HTTP associations until it stops moving. The next step is to get to the Turing part, where the customer's verification takes place, till then.

C) Text-based Turing Testing Module for first Proposed Framework

In Turing based system like CAPTCHA, if the client gives or enter legitimate data that client is utilized from pink listing to white listing. There is no compelling reason to stress over spending such a great amount of cash on such administrations, as the best suppliers offer at sensible rates.

D) Cache Based and Question Generation Turing Testing Module for Second proposed

Framework Reserve Based Turing Cache is such a check innovation in which less exertion is required and a protected side service is included. This empowers client to check through a protected server. Despite the fact that various exchange of service is required. It incorporates a couple f secure information relocation. This innovation is secure according to the outcome and in addition generally reliable or solid. This Turing is to get quick data about the client. The destination location stores various secure different destinations. The client is being requested to give access to these destination addresses. In the event that it is found there it moved from the black list to white list.

This Cached based Turing Test composed of subsequent Steps

1. Using a secure server side, the server connects to the client and retrieves the client's current association in the store. The solicitation response structure is ready to go whenever a customer requires a service. The client's request is sent to the server after the client's check is complete. Now the server is looking for the data stored in the client framework. The data stored in this cache is saved in the content configuration as a temporary entry in the framework CatLog, where it is saved as a name-quality pair. There is coordination between the stored information and the client-filled data. Once the client's data is properly integrated with the store's data, they will be able to approve the service.
2. Utilising the Client-Received Qualification, the Server Communicates with the Present Client. The current server is used to verify the client at this point. The client was successfully validated by the existing server using the framework's stored reserve data.
3. Verify that the client data stored on the existing server at day's end.
4. The process, the current server updates the server on the status. Based on the status that the server receives, it decides whether to update the cache store again or not to communicate with the client.

Algorithm: Step-by-Step Procedure involved in Preventing the DDos Attack

Step 1 : The hitcounter property is established by recording the number of hits or records per second as soon as the client becomes connected to the server with the number of requests. An initial value of 1 will be set to the hit counter for first-time visitors. Whereas, if it is not, the hit count is increased by one.

Step 2: To create or verify the existence of a log file to document the present status of the traffic monitoring and to record the IP addresses of the zombie computers. The white list is a file that is either produced or verified to ensure that only legitimate users are logged in.

Step 3: Commit the time to paper using both the seconds and the date format.

Step 4: In order to identify DDoS attacks, the server's defence time is set to 5 to 10 seconds and the threshold value, which is the number of requests per second, is set to a configurable figure.

Step 5: In order to identify a distributed denial of service (DDoS) assault, the source's IP address is logged or kept.

Step 6: To stop the connection with the client and refuse them access to the requested website if the IP address is already in the blacklist. Additionally, you may ban people from the blacklist.

Step 7: The current visit is recorded and the flag value is set as true if the IP address is mentioned in the white listing. We then proceed to further discover any issues. Upon initial connection, the user's session will be initiated; otherwise, connections will be checked and updated.

Step 8: Keep an eye on the traffic using the tallying module to get the last session request and the total number of requests.

Step 9: To determine how long the desired session is in relation to the current time and the defence time.

a) Check if request count is less than the threshold value set. If true, the threshold is increment to one and the service for the request is done.

b) If request count is greater than the threshold value set, and the IP address of the client is not found in white list, then block and add the connection to the black list as it reaches the given threshold and redirect the client to the access denied page.

c) Check if the last request count is equal to the threshold value and the IP address of the client is not in the white list, then the client is redirected to turing test page where the further verification or turing is done.

If the user sends more number of requests and the information filled in is matched with the cache data, the server then asks the client to respond to the randomly generated question to check the attack strength. If he is unable to answer the question properly then the client is directed to the access denied page. Even if the client tries to or attempts to login again to gain access to the requested page, he will be blocked permanently.

- **Materials**

Network experiments in computer science and circulation include programming a programme to mimic the behaviour of a network, either by simulating its constituent parts or by determining how those parts interact with one another. The use of mathematical formulas or, alternatively, the actual acquisition and examination of manufacturing network. In a controlled environment, we can examine the network's functioning and the many applications and services it supports. We can also manipulate the environment to see how the network reacts in different settings." This method, which is also known as network emulation, involves combining a simulation programme with actual apps and services to maintain end-to-end achievement on the user's desktop.

- **Glomosim/Qualnet**

Glomosim it is called as Global Mobile Information System simulator and is a public domain simulator established using UCLA. It has scalable environment for big communication and the wireless to wire line networks. GlomoSim it utilized for parallel discrete-event simulation. This ability it is provided through Parsec. Business tool of GloMosim it is called as QuaNet. QuaNet is derived from the GloMoSim and that was first established in the year of 2000 through SNT. The important comparison between the GloMoSim and QualNet is GloMoSim are establish from C language and the Qualnet is established from C++ language. GloMosim divided an open-source license and also controlled through parallel computing but the Qualnet is a private enterprise product and its maintained through SNT. QualNet Model Libraries: The QualNet help a multiple numbers of model libraries that enable to implement network through definite protocol models. These consists of Multimedia and Enterprise, wireless model Libraries and Developer. The extra libraries also applicable such as cellular networks these network are, WiMAX, military radio networks, UMTS, satellite networks, enhance propagation model library and sensor networks.

OMNet++

It is a framework for distinct event simulators that is component-based, modular, and open source. Although it is exposed for queuing network simulation, this model has many important applications in network simulation. Academic Public Licences, which permit the GNU Public Licence, govern the OMNeT++ open-source paradigm.

JSIM

The DRCL team has published JSim. The National Science Foundation (NSF), the DARPA Information Technology Office, the Air Force Office of Scientific Research's Multidisciplinary University Research Initiative, Ohio State University, and the University of Illinois at Urbana-Champaign have all provided funding for this research.

QualNet

Technologies for Scalable Networks During the years 2000 and 2001, I supplied commercial network simulator, Inc. The S/W is a kind of network simulation software that can expand the number of networking devices and simulate wireless, wired, and mixed-platform networks. These independent applications are network-capable.

NS2

Students from UC Berkeley, USC's Information Science Institute, Lawrence Berkeley National Laboratory, Xerox Palo Alto Research Centre, and the Virtual Inter Network Testbed collaborated on the development of Network Simulator 2. In the year 1995. Both the National Science Foundation and the Defence Advanced Research Projects Agency are highly inspired participants.

NS3

In the research and educational primarily use ns-3 simulator and that is mixed-event network. These are started in the year of

2006, ns-3 is an open source implementation framework. The aim of this paper is introduce the proposed ns-3 customer to the system scheme way.

If recent user to get the need information from detailed manuals that time it is hard and to convert this data into working simulations. Following the main point about the ns-3.

- Ns-3 it is an open-source framework, and application try to organize an open arounding to research worker that give the S/W.
- Ns-3 it is a new simulator; and these are not back-sutaible addition of ns-2. Ns-1 and ns-2 simulators are developed by using C++ language.

For the purpose of networking research and teaching, the Ns-3 network simulator was released as an open-source, expandable platform for network simulation. Descriptively, ns-3 offers a model of the functioning and performance of data packet information networks and a simulation engine allowing customers to do actual simulations.

The disposable model set in ns-3 focuses on simulating how the network and IP (Internet protocols) function, although they are only applicable to systems that are dependent on the Internet. Some customers are utilising ns-3 to model systems that are not dependent on the Internet. Research on networks may benefit from the availability of several numerical simulation tools. Here are some characteristics of ns-3

- Network simulation-3 it is scheme as group of libraries and these can be attached corporately and also with outer S/W libraries. While several simulation sides provide customer with an only one, combined GUI arrounding and they are each and every outcomes are find out.
- Network simulator-3 is mainly used on Linux operating system, and that help survive for FreeBSD, Cygwin and native windows Visual support is in the method of being published.
- Ns-3 it is does not help for S/W product of the any another industry. These are work on user best-effort.

The network makes it possible to see the imitation output from Ns-2. The ns-2 animation programmes and names are OTcl scripts. Executing a spoof in ns-2 is not doable. The C++ language also contributed to ns-2, and OTcl has preserved some of its features. Built in C++ with optional Python bindings, the ns-3 is an imitator. Other vulnerabilities are used to analyse the PCAP packet trace files, which are generated by ns-3. "Should it stay with ns-2 or switch to ns-3?" is a common concern. A client will be more creative with ns-3 for the following points, regardless of the author's judgement, unless it is somehow vested in ns-2:

- While ns-2 has been slowly organised and has not seen substantial growth in its major code tree, ns-3 is mostly organised and has an active, responsive customer mailing list.
- While the assembly code runs in the background, Ns-3 provides features that are irrelevant to ns-2.
- The third iteration of network simulation differs somewhat from the second, releasing it in a way that is most in line with the way actual systems are constructed; it offers a basic degree of engagement. Some limitations are discovered in the ns-2 model. These constraints are addressed in the ns-3 model.

Because of its lengthy history, Ns-2 has a very different set of displayed modules than doe's ns-3. However, with the aid of assembly code, it announces a wide range of high-fidelity approaches, and network simulation-3 has a vast discussion approach in many study areas. It is possible to encapsulate the whole Linux networking stack in a ns-3 node after preparing ns-2 models in C++, which allows for irregular transformations into Ns-3 models using Direct Code Execution (DCE).

- **NS-3 Resources**

Getting Code

There are distinct main resource of which any ns-3 user it must be attentive. The main web site is related at http://www.nsnam.org and provided access to fundamental data abut the ns-3 model. The reported documentation is relevant using the significant web site at http://www.nsnam.org/documentation/. By the use of this site page can recognize document associated to the model architecture. These are also recognize customer and developer FAQ there, and destroy lead, third-party grant code etc. This source code can be recognize and browsed at above website http://code.nsnam.org/. In the depositary named ns-3-dev you will detect the present development tree.

NS3 Outputs

Performance Measurement

To evaluate the efficacy of wireless networks, NS3 employs a flow-monitor paradigm. In the directory src/flow-monitor, you can find the new module's source code.

The Flow Monitor module's primary objective is to facilitate the flexible system's determination of the network's throughput. You may find a plethora of parameters in the modules by using the probes that are generated in the network nodes to monitor the packet exchanges between them. “This flow describes the probes function, the IP address and ports of the source node, and the IP address and ports of the destination node, all based on the node's position in the network.

For XML-based flow forwarding, the statistics are consistent. In addition, the user may directly contact the probes to obtain particular statistics flow.

Design

There is a modular approach to implementing the Flow Monitor module. You may scale these using the ns3::Flow Classifier and ns3::Flow Probe parameters.

Flow Monitor discusses the whole module design.

3. RESULTS

- **Visualization Results**

We used the NS3 visualizer paradigm for network visualisation. Figures below show snippets of the visualisation. In the system architecture shown in Figure, fifty travelling users are distributed randomly between two servers, one of which serves as a home server and the other as a foreign server. When nodes move from one server to another, these servers take on these roles. These two servers make up the network's two end nodes, which are linked by a gray-colored connection. The sensing patterns and connections in this network are shown in Figure For the sake of communication, every node in an IP network is always monitoring the nodes in its immediate vicinity. The red colour indicates packet loss in Figure. We built the system with the idea of hostile actors inflicting network packet loss in mind. In a similar vein, figure shows that packet loss occurs when a user moves from a local server to an international one.

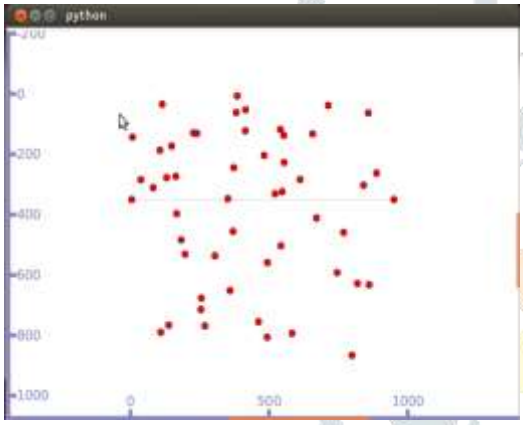

Figure 1: A network with two servers (one at home and one abroad) and fifty roaming users

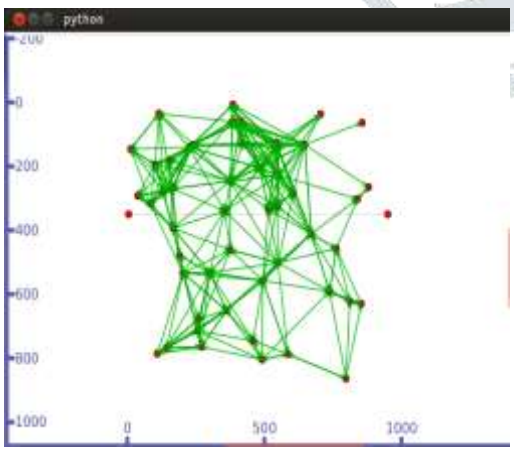

Figure 2: Network Connectivity Result

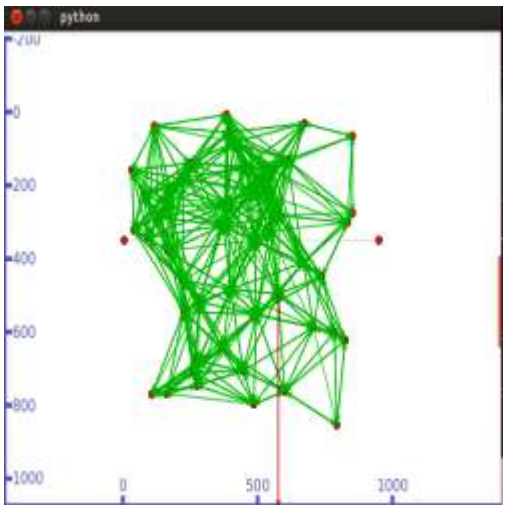

Figure 3: Packet Loss Indication in the presence of Malicious Attacker

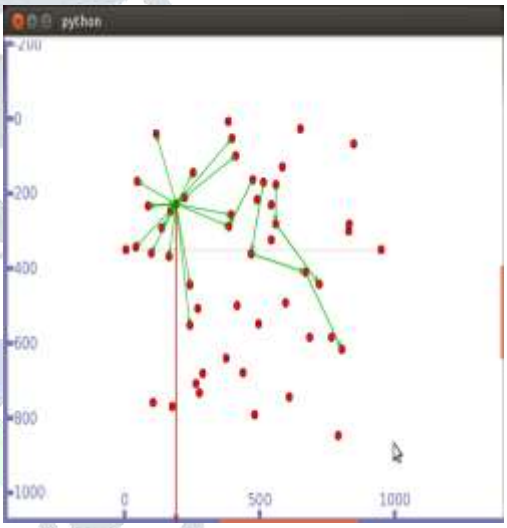

Figure 4: Packet Drop Indication While Roaming in the presence of Malicious Attacker

Nothing stands out as performance metrics when looking at it via a visualisation lens. As a result, research visualisation is often quite narrow in scope, with an emphasis on measuring performance metrics using simulation traces. Each contribution's results for four performance measures are shown in the sections below. All of the aims and objectives of this study are met by Contribution 2. This is the part where we put our new, HCIF-based approach into action and create it from scratch. The development and use of techniques for authentication and privacy preservation as a routeing protocol for mobile IP networks was the innovative aspect of contributions 1 and 2.

Varying Nodes Scenario

Table 1: Average Throughput Analysis for Contribution 2

	RBP Boost	HCIF	Proposed
0	60	100	150
10	78	90	150
	120	95	157
20	90	105	145
30	100	100	133

Table 2: Packet Delivery Ratio Analysis for Contribution

	RBP Boost	HCIF	Proposed
0	45	78	98
10	60	70	97
	80	77	95
20	70	82	90
30	80	80	88

Table 3: Packet Loss Ratio Analysis for Contribution

	RBP Boost	HCIF	Proposed
0	138	58	4
10	104	75	5
	55	65	10
20	75	75	12
30	55	45	16

Table 4: Delay Analysis for Contribution 2

	RBP Boost	HCIF	Proposed
0	0.04	0.12	0.018
10	0.07	0.05	0.018
	0.1	0.085	0.018
20	0.09	0.095	0.019
30	0.138	0.058	0.02

The proposed HCIF routeing protocol achieves improved throughput performance. The HCIF team is confident in their ability to stop rogue nodes from compromising wireless network security by effectively handling and mitigating their threats. When compared to all prior approaches, the throughput performance is around 25% better.

Unfortunately, hackers often make it so that data doesn't get it to its intended destination. Better PDR performance is being produced by HCIF, which is resolving this issue. With its two-day security measure, HCIF eliminates the possibility of data loss caused by a mix of privacy preservation and secure communication techniques. With the suggested approach, packet loss is minimal. Approximately 15% of packet loss performance is reduced. When compared to HCIF, the previous method's low delay performance is clearly an issue.

Varying Number of Malicious Attackers

Table 5: Analysis of Throughput Performance with Changing Attacker Counts

	RBP Boost	HCIF	Proposed
1	85.504	103.936	119.808
3	96.25	115.712	124.41
5	65.024	76.8	89.6
7	65.024	80.896	82.94
9	61.22	74.22	79.2

Table 6: Evaluating the Efficiency of Packet Delivery Ratio under Different Attacker Scenarios

	RBP Boost	HCIF	Proposed
1	66.8	81.2	93.6
3	75.2	90.4	97.2
5	50.8	60	70
7	50.8	63.2	64.8
9	47.45	57.2	60.34

Table 7: Examining the Effect of Changing the Attacker Count on the Packet Loss Ratio

	RBP Boost	HCIF	Proposed
1	83	47	16
3	62	24	7
5	123	100	75
7	123	92	88
9	135	118	100

Table 8: Analysing Performance of Delay with Different Attacker Numbers

	RBP Boost	HCIF	Proposed
5	4.52	4.53	4.01
10	5.06	5.06	4.17
15	5.28	5.29	4.46
20	5.6	5.55	5.09
25	7.02	7.04	5.53

4. CONCLUSION

Distributed denial of service attacks demonstrates a real problem with the Internet and put its growth rate, widespread recognition by the public, and sceptical public and commercial organisations to the test. A botnet "zombie army," a collection of Internet-connected PCs, is often used in HTTP surge assaults, which are volumetric attacks. Of which, malicious actors have taken control, sometimes with the use of Trojan Horses and other forms of malware. To ensure the suggested work is efficient and to assess the effectiveness of the systems in mitigating network DDoS traffic, the HCIF and RBPBoost defence frameworks are run." Additionally, a Confidence-Based filtering method for DDoS Attack Defence in Web applications is suggested, together with an HTTP GET flooding location, to overcome HTTP Flood issues. It is linked to an early stage method for detecting HTTP GET flooding. Using the server's available resources (CPU, memory, I/O, transmission capacity, etc.) in a way that mitigates DDoS assaults on specific clients is the job of the dynamic asset component.

REFERENCES

1. Kalaikavitha E, Juliana Gnanaselvi. Secure Login Using Encrypted One Time Password (OTP) and Mobile Based Login Methodology. International Journal of Engineering and Science. 2013 Apr; 2(10):14-17.

2. Jesudoss A, Subramaniam NP. EAM: Architecting Efficient Authentication Model for Internet Security using ImageBased One Time Password Technique. Indian Journal of Science and Technology. 2016 Feb; 9(7):1-8.

3. Keunsoo Lee, Juhyun Kim, Ki Hoon Kwon, Younggoo Han, Sehun Kim. DDoS attack detection method using cluster analysis. Expert Systems with Applications. 2018 Apr; 34(3):1659–65.

4. Mirkovic J, Peter Reiher. D-WARD: A Source-End Defense against Flooding Denial-of-Service Attacks. IEEE Transactions on Dependable and Secure Computing. 2019 Aug; 2(3):216-32.

5. Yau DKY, Lui JCS, Feng Liang. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. ACM Transaction on Networking. 2020 Feb; 13(1):29-42.

6. Countering Denial-of-Service Attacks Using Congestion Triggered Packet Sampling and Filtering. Date Accessed: 15/10/2021: Available from: http://ieeexplore.ieee.org/document/95630 9 /?reload=true & a number=956309.

7. Ganesh Kumar K, Arivazhagan D. Generating A Digital Signature Based On New Cryptographic Scheme For User Authentication And Security. Indian Journal of Science and Technology. 2022 Oct; 7(S6):1-5.

8. Rosario Gennaro, Yehuda Lindell. Springer Berlin Heidelberg: A framework for password-based authenticated key exchange. 2023 May; p. 524-43.

9. William G Morein, Angelos Stavrou, Debra L Cook, Angelos Keromytis, Vishal Misra D. Using Graphic Turing Tests to Counter Automated DDoS Attacks Against Web Servers. Proceedings of 10th ACM conference on Computer and communications security. 2023 Sep; p. 8-19

10. Yongdong Wu, Zhigang Zhao, Feng Bao, Robert H Deng. Software Puzzle: A Countermeasure to ResourceInflated Denial-of-Service Attacks. IEEE Transactions on Information forensics and security. 2019 Jan; 10(1):168-77.