International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 6, Issue 2, March - April 2017 ISSN 2278-6856

# Fortified HoneyWall: Virtual Honeypots to Detect and Prevent Distributed Denial of Services Attack

**Sonal Sinha**

Principal Scientific Officer, ITRA- Media Lab Asia,
Ministry of Electronics & IT, New Delhi, India

## Abstract

***Nowadays, internet and web services have become inseparable parts of our lives. Thus major thrust is on continuous and efficient delivery of internet services as the success of any organization is based on it. These services are often intermittent as there are numerous security threats which affect them. Thus management of optimal protection is a major concern for organizations. One such major attack on the services is Distributed Denial of Services Attack (DDoS) which results in temporary slowdown in services and could escalate to complete non-availability of the service. Fortified HoneyWalls are firewalls which are embedded with multiple virtual honeypots. These honeypots are interconnected to form a mesh and have backup honeypots for fault tolerance. Also virtual honeypots are installed at production servers as daemon processes. These honeypots allure attackers to interact with them giving an illusion of an actual system. These can detect, deflect and prevent malicious accesses and attacks to ensure continuous delivery of requisite services. This paper gives an insight into DDoS, various prevention mechanisms for DDoS, Honeypots and their deployment for prevention of Denial of Services attacks.***

**Keywords—**Distributed Denial of Service, attack source, victim server, firewall, honeypot, virtual machines, honey daemons, challenge response, virtual network, fortified HoneyWall, Honeymesh.

## 1.Introduction

In today’s fast paced world continuous uninterrupted efficient service is the bedrock for all service organizations. The success of any new or existing venture is critically dependent on reliability and continuous availability of service. Gradually each and every individual is becoming more and more dependent on the web for efficient and timely fulfillment of his need. This raises the bar for quality too high for the service delivery organizations. They need to be extra cautious while hardening their security infrastructure. Various kinds of threats and attacks are continuously trying to breach their security structure. One of the most difficult attacks to prevent is the Distributed Denial of Services (DDoS) Attack since it has direct effect on the service availability to a consumer.

## 2.Denial of Services Attack

A denial of service (DoS) attack is an attempt to make a service, usually one offered over internet, unavailable to its legitimate users. A denial of service attack results in a temporary or long-term non-availability of a service to its intended users by the way of either crashing or delaying the service. Denial of services attack is an attack in which the victim server is flooded with bogus service requests. Due to this legitimate traffic to the service is blocked and authentic users are denied service. [1]

There are four different ways to defend against DoS attacks:

- Attack Prevention
- Attack Detection
- Attack Source Identification and
- Attack Reaction.

## Distributed Denial of Services (DDoS) Attacks

According to the WWW Security FAQ on Distributed Denial of Service (DDoS) attacks: ‘‘A DDoS attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the DoS significantly by harnessing the resources of multiple unwitting accomplice computers, which serve as attack platforms’’. It is distinguished from other attacks by its ability to deploy its weapons in a ‘‘distributed’’ way over the Internet and to aggregate these forces to create lethal traffic. DDoS attacks never try to break the victim’s system, thus making any traditional security defensemechanism inefficient. The main goal of a DDoS attack is to cause damage on a victim either for personal reasons, either for material gain, or for popularity. Distributed Denial of service has the cohesive strength of many compromised systems working towards a single cause. The first stage of this attack is to build its platform with many host systems that can work under remote commands. The attacker group would first scan networks to hunt for vulnerable systems that are weak in security features. According to researchers there are millions of host machines that are vulnerable without secure patches and proper updates that often fall victims to these attackers. Once the scanning procedure is completed, attackers would bring these hosts into control using software exploitations like buffer overflow, dangling pointers, code injection etc [2]. Special root kits are also used in many cases that are installed in a host system to incur these software exploitations. After having sufficient hosts under control, attackers also create backdoors that allows special access that is used for future entry. The attackers also update the hosts and tighten its security so that another attacker does not use the same host. Any future entry would be done using the back entry that has been specially crafted.

## 3.Prevention of DDoS

The countermeasures proposed for preventing a DDoS attack are currently partial solutions at best. There is currently no comprehensive method to protect against all known forms of DDoS attacks. Also, many derivative DDoS attacks are continually being developed by attackers to bypass each new countermeasure employed. We propose a preliminary taxonomy of DDoS Countermeasures in Figure 1.

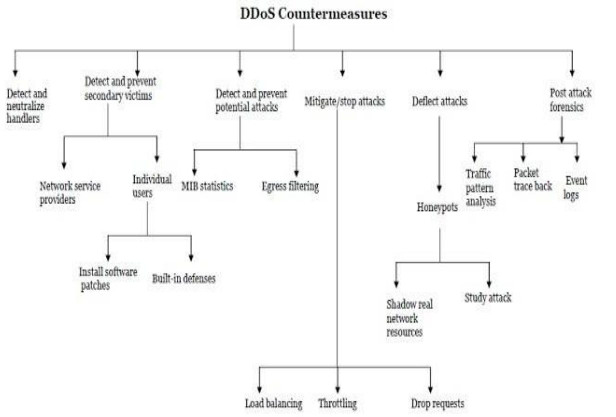

**Figure1:** DDoS Countermeasures

There are three categories of DDoS countermeasures. First: preventing the setup of the DDoS attack network, including preventing secondary victims and detecting and neutralizing handlers. Second: dealing with a DDoS attack while it is in progress, including detecting or preventing, mitigating or stopping, and deflecting the attack. Third: there is the post-attack category involving network forensics.

## Prevention of DDoS through Honeypots

Some solutions also propose the usage of honeypots to mitigate DDoS attacks [3]. Honeypots are systems which give the look and feel of actual systems and attract intruders to interact with them. A cluster of physical honeypot servers can be implemented that mimic the activities of real servers [4]. This solution is expensive since every honeypot needs a separate physical server which results in wastage of resources and high maintenance costs. Use of “Active Servers” (AS) was proposed to mitigate denial of service [5]. A production server is hidden behind an AS that acts like a gateway to the production server. Only legitimate traffic is passed on to real server and malicious traffic is restricted. For malicious traffic, an AS acts like a honeypot thus protecting real server from being compromised. This solution is robust and secure but it slows down processing of requests for real users as each and every request needs to pass through an additional gateway. Moreover, a separate honeypot server for each production server wastes resources and again this solution proves to be expensive. Also flooding attacks with thousands of requests can clog these gateway servers thereby greatly slowing down the access to production servers. Another solution was proposed to mitigate denial of service attacks where honeypots and production servers are frequently shuffled within the network [6]. Honeypots are used to detect and prevent DDoS attacks. This solution is effective when most incoming requests are DDoS requests. But if majority of traffic is legitimate and only few requests are DDoS attacks, the solution is ineffective since a certain number of servers function as honeypots irrespective of the traffic. This again wastes resources and constant shuffling of honeypots and production servers in fact slows down service for intended users.

## Fortified HoneyWall

The proposed solution for protection from DDoS attacks would use a combination of virtual honeypots to protect the system. Most of these honeypots would be installed inside firewalls so that a better management and control may be applicable. We may term this group of virtual honeypots as Honey Farm. Although a firewall with honeypots works in opposition of the natural firewall behavior i.e. instead of preventing or restricting unauthorized access it attracts attackers to interact with the system for longer durations. This helps identify the attacker and his intention. These virtual systems lure attackers and help administrators with the following

i) The administrator may get to know the vulnerabilities of the system thus security can be hardened

ii)An attacker would be caught and stopped before he could exploit the vulnerabilities of the system

iii) Network designers may gain insight by studying the activities of the attacker to better design the system and to make it more robust.

The firewall with honeypots would also actas a Gateway Server which will filter the traffic and further guide the malicious requests to the virtual honeypots installed on various servers. The Firewall with virtual honeypots is hardened with these auxiliary virtual honeypots installed at the active servers (AS). This system will provide fault tolerance and robustness thus a Fortified HoneyWallis constructed.

Unlike earlier solutions that used explicit servers as honeypots to function as gateways and mimic a real server, this solution proposes to implement honeypots as virtual machines (VM) that can be hosted on a few physical servers [7] and the firewall. Since VM’s share resources, multiple honeypots can be hosted on a single server [8] and firewall.

In opposition to the solutionwhere separate honeypot servers function as gateways to individual production servers [5], the gateway honeypot can run as a daemon process on the active server. This honeypot daemon, would work as a gateway and perform initial authentication before passing on the information to the actual server. Thus even if the honey farm fails to detect an attack, honey daemon present within the server provides an additional layer of security. This, together with the hybrid network of honey VM's functions like a mesh of virtualized honeypots and ensures effective detection and prevention of possible DDoS attacks.

## 3.Attack Detection and Prevention

Honeypot VM's in the honey farm use machine learning algorithms to perform a behavioral analysis of incoming traffic [9]. Since each production server receives different types of requests, appropriate honey VM's can be customized for different servers. For example, one honey VM can analyze web server traffic while another can examine file server requests. After analyzing a few thousand requests, each honey VM generates a baseline model of expected traffic. Incoming requests are compared against the baseline. If any deviation is observed, the request is further probed to confirm if it actually constitutes a DDoS attack.

Once the honeypot suspects a particular request based on behavioral analysis, it needs to verify that the suspicious request is actually a DDoS attack. For this, the honey VM then employs a challenge-response model to gather more information. This is accomplished by sending a set of challenge queries to suspicious source [10]. Based on the responses received, the honey VM decides whether further investigation is necessary. If yes, more sophisticated challenges are sent to the source. Based on the responses received and an intelligent behavioral mechanism, the honey VM can conclude whether the requests are part of a DDoS attack. This process is fully automated and happens without human intervention thereby guaranteeing excellent service for legitimate users. Similar mechanisms can be built into the honey daemons that run on production servers. This ensures that even if the honey farm misses out on a potential attack, it is reexamined by honey-d's running on respective servers. This provides an additional level of authentication and intrusion detection.

## 4.Preventing Flooding Attacks

Once an attack is discovered, the routing information in the internal routers is modified so as to redirect all incoming traffic from the attack source to the honey farm. Since malicious traffic now flows to the honey farm, it ensures that the production network is shielded from flooding attacks. Honey VM's in the farm keep the attacker engaged through a set of challenge-response queries further slowing down the attacker [10]. Also, once the attack source is confirmed, all incoming traffic from that source is blocked at the firewall itself. This mechanism mitigates the impact of flooding attacks to a great extent.

## 5.Conclusion

Distributed denial of service (DDoS) attacks are dangerous and can potentially render the production site unusable either by flooding the server network with thousands of malicious requests or crashing the server by exploiting the vulnerabilities in its software. Several solutions have been proposed to deal with DDoS attacks. However, these solutions are either expensive due to usage of multiple physical servers for honeypots or do not successfully address the issue of flooding type of DDoS attacks. The new solution proposes to create a virtual network or mesh of honeypot VM's and honey daemon processes to provide multiple levels of security checks and intrusion detection using behavioral analysis and challenge response models. Also, malicious traffic is routed to honey farm thereby protecting the production server and internal networks from both crashing and flooding type of DDoS attacks. Since all virtual honeypots are integrated/embedded in the firewall and others run as virtual daemons the Fortified HoneyWall very robust and provides multiple levels of security checks and intrusion detection mechanisms to effectively detect deflect and prevent possible DDoS attacks.

## References

[1] Christos Douligeris and AikateriniMitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art", Computer Networks: The Int. Journal of Computer and Telecommunications Networking, vol. 44, no. 5, Apr. 2004, pp. 643–666

[2] Jun Xu; Wooyong Lee; "Sustaining availability of Web services under distributed denial of service attacks," Computers, IEEE Transactions on , vol.52, no.2, pp. 195- 208, Feb. 2003 doi; 10.1109/TC.2003.1176986

[3] Kumar Shridhar and Nikhil Gautam, "A Prevention of DDos Attacks in Cloud Using Honeypot ", International Journal of Science and Research, Volume 3 Issue 11, November 2014, pp. 2378-2383.

[4] Natalie Weiler, "Honeypots for distributed denial-of-service attacks", Proceedings of Eleventh IEEE International Worksops on Enabling Technologies, 2002. [11] Vinu V. Das, "Honeypot Scheme for Distributed Denial-of-Service", Proceedings of the 2009 International Conference on Advanced Computer Control, January 2009, pp. 497-501.

[5] Vinu V. Das, "Honeypot Scheme for Distributed Denial-of-Service", Proceedings of the 2009 International Conference on Advanced Computer Control, January 2009, pp. 497-501

[6] Sherif M. Khattab, ChatreeSangpachatanaruk, Daniel Moss, Rami Melhem and TaiebZnati, "Roaming Honeypots for Mitigating ServiceLevel Denial-of-Service Attacks", Proceedings of the International Conference on Distributed Systems, March 2004, pp. 328–337.

[7] Xuxian Jiang and Xinyuan Wang, "Out-of-theBox Monitoring of VMBased High-Interaction Honeypots", Proceedings of the International

Conference on Recent Advances in Intrusion Detection, September 2007, pp. 198-218.

[8] Yu Adachi and Yoshihiro Oyama, "Malware Analysis System using Process-Level Virtualization", Proceedings of IEEE Symposium on Computers and Communications, July 2009, pp. 550-556.

[9] YiZhang, QiangLiu and Guofeng Zhao, "A Real-Time DDoS Attack Detection and Prevention System Based on per-IP Traffic Behavioral Analysis", IEEE 3rd International Conference on Computer Science and Information Technology (ICCSIT '10), April 2010, pp. 163–167.

[10] Aamir, M. and Arif, M., "Study and performance evaluation on recent DDoS trends of attack & defense", International Journal of Information Technology and Computer Science, 2013, pp. 54–65.

**AUTHOR**

**Sonal Sinha** received B.Tech. degree in Information Technology and M.Tech. in Computer Science and Engineering from U P Technical University in 2004 and 2014, respectively. During 2005-2015, she had served as Assistant Professor in Computer Science and Engineering. At various engineering colleges under U P Technical University. She is currently serving as a Principal Scientific Officer at ITRA, Media Lab Asia under the Ministry of Electronics and IT, Govt. of India.