

DDA: An Approach to Handle DDoS (Ping Flood) Attack

Virendra Kumar Yadav, Munesh Chandra Trivedi and B.M. Mehtre

Abstract Distributed denial of service attack (DDoS) is an attempt by malicious hosts to overload website, network, e-mail servers, applications, network resources, bandwidth, etc. Globally DDoS attacks affected four out of ten organizations (around 41 %) over the past few years. Challenges involved in taking counter measures against DDoS attacks are network infrastructure, identifying legitimate traffic from polluted traffic, attacker anonymity, large problem space, nature of attacks, etc. Several approaches proposed in the past few years to combat the problem of DDoS attacks. These approaches suffer for many limitations. Some of the limitations include: implementing filtering at router (firewall enabled) will create bottleneck, additional traffic, no means of sending alert to an innocent host acting as a bot, etc. Ping flood attack is one kind of DDoS attack. In this paper, ping flood attack is analyzed and a new approach, distributed defence approach (DDA) is proposed to mitigate ping flood attack. Distributed defence is applied with the help of routers connected to network when count of PING request crosses a threshold limit or packet size is greater than normal ping packet size. Concept of the proposed approach is to help the end router by putting less load during filtering attack packets, enhancing the speed of processing and informing the innocent host acting as bot simultaneously making the DDoS attack ineffective.

Keywords Distributive defence approach (DDA) • PING • Intrusion prevention system (IPS) • Message to source address (MtSA) • Next reset time (NRT)

V.K. Yadav (✉) • M.C. Trivedi
CSE Department, ABES Engineering College, Ghaziabad, Uttar Pradesh, India
e-mail: virendrashines@gmail.com

M.C. Trivedi
e-mail: Munesh.trivedi@gmail.com

B.M. Mehtre
IDRBT, Hyderabad, Andhra Pradesh, India
e-mail: bmmehetre@idrbt.ac.in

1 Introduction

DDoS attack is a kind of attack in which the attacker target the victim network resources such as bandwidth, memory, etc., so that victim may stop responding to legitimate users. The flood created by attacker forces victim to shut down for its legitimate user thus causing denial of service to its legitimate user. DDoS attack is also known as bandwidth attacks. DDoS attack can target many different network components such as firewalls, routers, ISPs, data centers, servers, appliances, etc. In DDoS attack, attacker creates the networks of bots also known as zombies by spreading malicious softwares. Sources of spreading malicious software could be emails, social media, Trojan viruses, malware, etc. Once infected, the machine will act as bot following attacker instructions remotely without their owner's knowledge. The collection of bots is commonly known as botnets. If number of bots involved during the DDoS attack is high, situation become more complex. Bots amplifies the power of attacker simultaneously making defence more complicated.

PING stands for Packet InterNet Groper. Mike Muuss has written the PING program to check the reachability of another host. PING uses two ICMP query messages: ICMP (ECHO request) and ICMP (ECHO reply). When a source make ICMP (ECHO (PING) request)) to another host, according to RFC 0792 guidelines, that host must reply with ICMP (ECHO (PING) reply)) after receiving the request from source. In Ping flood attack, attacker with the help of bots send several ICMP echo requests to victim without waiting for reply (Fig. 1). Now victim according to guidelines of RFC 0791 after receiving the ICMP echo request tries to reply with ICMP echo reply packets to source. Attacker sends request packets as fast as possible to consume bandwidth or network resources of victim, forcing victim to shut down or slowdown.

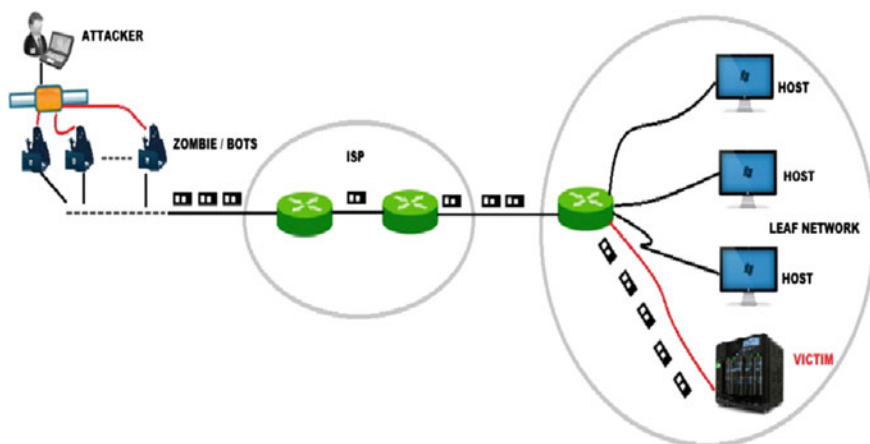
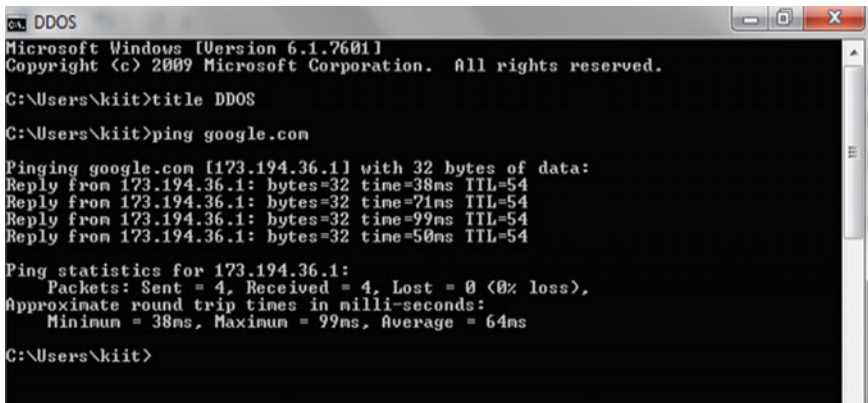


Fig. 1 DDoS attack overview



```

CA DDOS
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\kiit>title DDOS
C:\Users\kiit>ping google.com

Pinging google.com [173.194.36.1] with 32 bytes of data:
Reply from 173.194.36.1: bytes=32 time=38ms TTL=54
Reply from 173.194.36.1: bytes=32 time=71ms TTL=54
Reply from 173.194.36.1: bytes=32 time=99ms TTL=54
Reply from 173.194.36.1: bytes=32 time=50ms TTL=54

Ping statistics for 173.194.36.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 38ms, Maximum = 99ms, Average = 64ms

C:\Users\kiit>

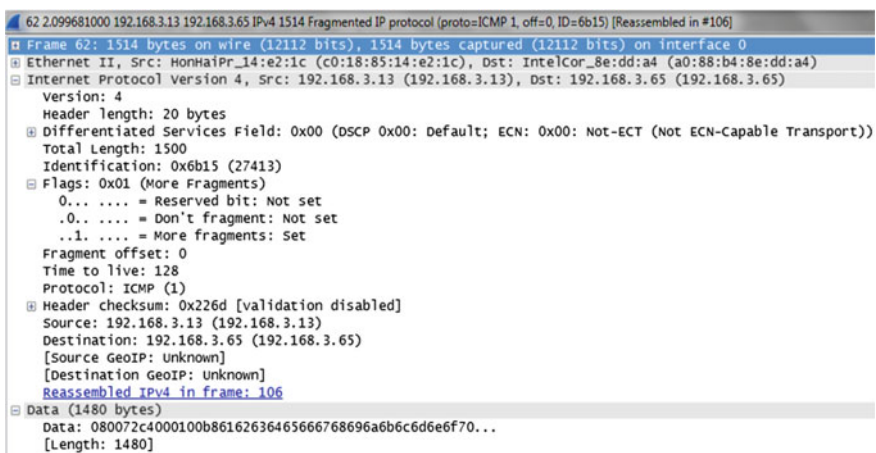
```

Fig. 2 Ping command overview

By default, the ICMP echo request packet contains data of 32 bytes under Windows (Fig. 2) and 56 bytes under Linux. But attacker can send data which can be greater than 32 bytes.

By default, the ICMP echo request packet contains data of 32 bytes under Windows (Fig. 2) and 56 bytes under Linux. But attacker can send data which can be greater than 32 bytes.

The length of an IP packet is 1514 bytes, maximum packet size supported by Ethernet. If attacker sends data in ICMP echo request which is more than 1500 bytes then sender or router will make fragments of this packet and set flag equal to one in flags field (Fig. 3). For example, suppose an attacker send 65,500 bytes data in each ICMP echo packet then it will get fragmented and receiver host (victim) should reassemble these IPv4 packets at their side. Sometimes when attacker sends

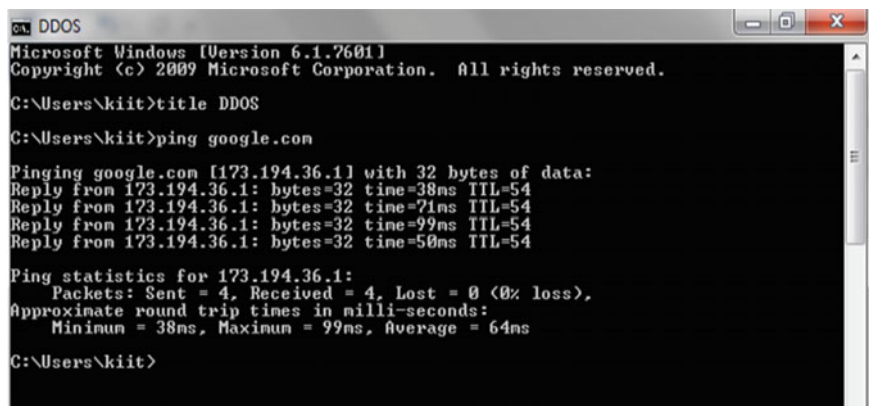


```

62 2.099681000 192.168.3.13 192.168.3.65 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=6b15) [Reassembled in #106]
  Frame 62: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
    Ethernet II, Src: MonMaiPr_14:e2:1c (c0:18:85:14:e2:1c), Dst: Intelcor_8e:dd:a4 (a0:88:b4:8e:dd:a4)
    Internet Protocol Version 4, Src: 192.168.3.13 (192.168.3.13), Dst: 192.168.3.65 (192.168.3.65)
      Version: 4
      Header length: 20 bytes
      Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
      Total Length: 1500
      Identification: 0x6b15 (27413)
      Flags: 0x01 (More Fragments)
        0... .... = Reserved bit: Not set
        .0... .... = Don't fragment: Not set
        ..1. .... = More fragments: Set
      Fragment offset: 0
      Time to live: 128
      Protocol: ICMP (1)
      Header checksum: 0x226d [validation disabled]
      Source: 192.168.3.13 (192.168.3.13)
      Destination: 192.168.3.65 (192.168.3.65)
      [Source GeoIP: Unknown]
      [Destination GeoIP: Unknown]
      Reassembled IPv4 in frame: 106
    Data (1480 bytes)
      Data: 080072c4000100b86162636465666768696a6b6c6d6e6f70...
      [Length: 1480]

```

Fig. 3 Captured packets (ICMP echo request)



```

C:\ DDOS
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\kiit>title DDOS
C:\Users\kiit>ping google.com

Pinging google.com [173.194.36.1] with 32 bytes of data:
Reply from 173.194.36.1: bytes=32 time=38ms TTL=54
Reply from 173.194.36.1: bytes=32 time=71ms TTL=54
Reply from 173.194.36.1: bytes=32 time=99ms TTL=54
Reply from 173.194.36.1: bytes=32 time=50ms TTL=54

Ping statistics for 173.194.36.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 38ms, Maximum = 99ms, Average = 64ms

C:\Users\kiit>

```

Fig. 4 Windows ping valid range (screen shots)

data which is more than victim machine can handle, then victim machine can crash (ping of death). For security reasons, Windows has fixed this size up to 65,500 bytes (Fig. 4).

1.1 Additional Information Required by the Volume Editor

If you have more than one surname, please make sure that the Volume Editor knows how you are to be listed in the author index.

1.2 Copyright Forms

The copyright form may be downloaded from the For Authors section of the LNCS Webpage: www.springer.com/lncs. Please send your signed copyright form to the Contact Volume Editor, either as a scanned pdf or by fax or by courier. One author

The important question is what are the consequences of DDoS attack? The consequences depend upon the intentions of the attacker and his success rate. According to Ponemon institute study, the average cost due to DDoS attack is \$22,000 when downtime is equal to 1 min [1]. There are several variables to determine these costs; for example, volume of online business, brand value, competitors and business segment. Latest impact of DDoS attack is published in the South China morning post Hong Kong titled ‘Cyberattack threatens to derail Hong Kong’s unofficial vote on universal suffrage’ (Fig. 5) [2].

The work in this paper presents distributed defence approach to prevent ping flood attack with the help of neighbouring routers simultaneously maintaining the efficiency of end router of victim network. The rest of the paper is organized as

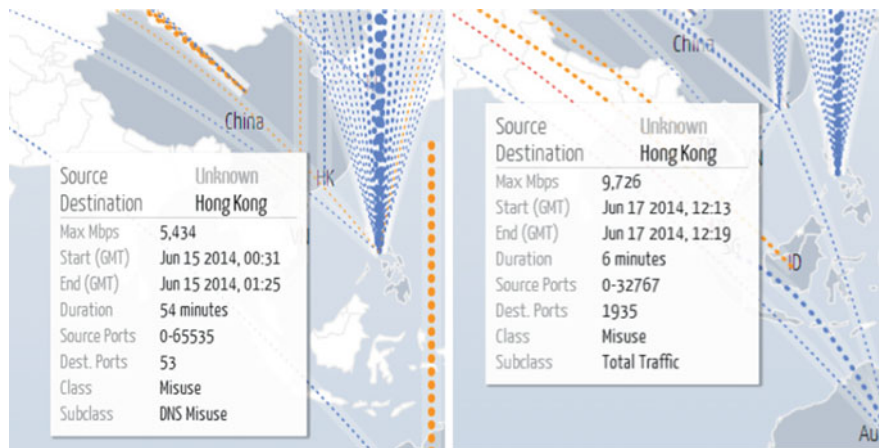


Fig. 5 Ping flood attack at Hong Kong [3]

follows: Sect. 2 contains the related works, Sect. 3 contains the proposed concept, Sect. 4 contains the proposed algorithm and Sect. 7 contains the conclusion of the proposed work (Figs. 6, 7 and 8).

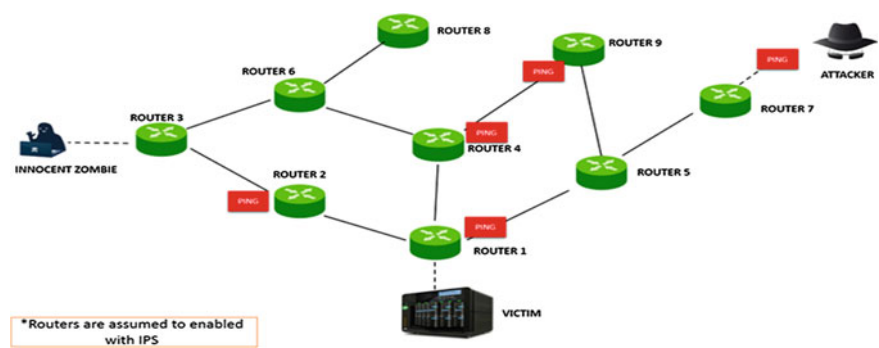


Fig. 6 Ping flood attack

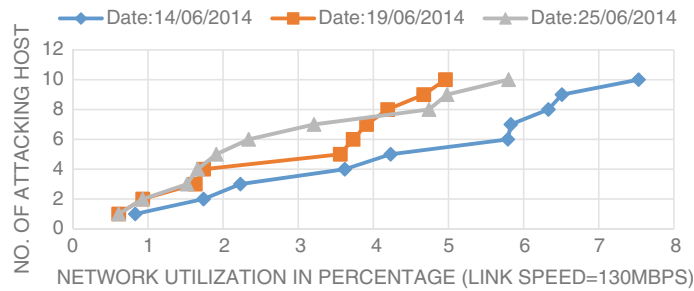


Fig. 7 Bandwidth consumption during PING flood attack

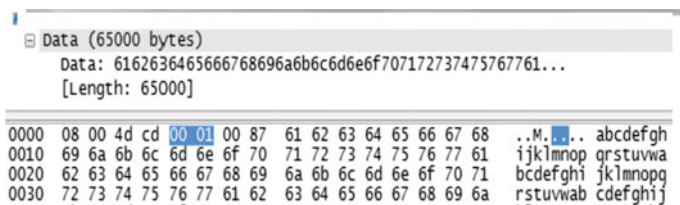


Fig. 8 Wireshark screen shot-1

2 Related Work

DDoS attacks are performed by attacker for denying end-user service. Several researchers are working in this domain since past few years. Authors in the paper titled ‘Towards User Centric Metrics for Denial of Service Measurement’ introduced the concept of DoS impact metrics. With the help of these metrics, one can measure the QoS that is experienced by end user when it is receiving DDoS attack [1]. According to authors, measurement approaches that are followed during DDoS attacks are imprecise and incomplete. Percentage of failed transactions (PFT) is the main impact measure for each category of applications. PFT means the percentage of failed transactions that have occurred during the DDoS attack. In this paper, they also define the threshold model whose concept is developed, based on the past findings.

Concept of Hop-Count Filtering was proposed by authors to distinguish the spoofed IP packets from legitimate packets [2]. Server on the basis of time-to-live (TTL) value in the IP header and IP addresses, creates mapping. This mapping helps to identify the spoofed IP packets.

In the article, titled ‘Survey of network-based defence mechanisms countering the DoS and DDoS problems’, authors conducted the survey about distributed denial of service attack. They discussed the various kinds of DDoS attack such as protocol-based bandwidth attacks (SYN flood and ICMP flood), application-based bandwidth attack (HTTP flood and SIP flood), distributed reflector attacks, DNS amplification attacks and infrastructure attacks. The article also presents methods or strategies available to defend against the DDoS attack. Comparison of each method has also provided by authors [3].

Apple and Windows in late 2000s released Snow Leopard and Windows 7 respectively. According to them, the developed OS provides user a reliable and safer operating system. No experiments were conducted which evaluate the reliability of these operating system. So authors put efforts in conducting experiments, i.e. how both the OS faced against DDoS attacks. Based on the experimental results, authors concluded that Window 7 OS is more reliable than Snow Leopard in limiting adverse effects of DDoS attacks [4].

Authors in the paper, titled ‘Defending Against Meek DDoS Attacks By IP Traceback-based Rate Limiting analyses a rate limit algorithm’, i.e. maxmin-based

rate limit algorithm. They try to put the algorithm fairness during the meek DDoS attack. Meek DDoS attack takes place when the bot or zombie behaves like a legitimate user. In that case, it is quite difficult to differentiate the legitimate traffic with the polluted traffic. Based on the analysis, authors proposed IP traceback-based rate limiting algorithm [5].

Subramani Rao and Sridhar Rao in their paper concluded with the help of experimental analysis about the role of network topology. According to them, topology of networks decides few important things such as: traffic amount passing through it, number of network elements, rate limiting, etc. [6]. Authors on the page number 39 mentioned one important line, i.e. in case of IPv6, DDoS attack would be stronger (88 %) in comparison with IPv4 [6].

Udaya Kiran Tupakula, Vijay Varadharajan in their paper analyzed the popular traceback technique. They also consider the real-time situation in which they raise certain issue such as how the attacker remains anonymous and remain untraced if any of these traceback techniques have been applied. Some of the IP traceback techniques considered in this paper are: single-packet IP traceback, IP packet marking technique and ICMP traceback technique, etc. [7].

Authors in this survey presented the study about the botnet. Nowadays, the 40 % of the host which are on the internet are infected and follow the instructions given by the attacker. Many of them are also unaware that they are acting as bot. In the study of botnet, focus has been put mainly in three areas: botnet understanding, tracking, detecting botnet and countering against botnet [8].

The authors in the paper proposed the system DoSTRACK. DoSTRACK system according to authors can handle the TCP SYN and reflection DDoS attacks [9].

Although there are several techniques available in the today's world, to mitigate with the DDoS attack or to prevent attack, still they have certain limitations.

- Implementing certain techniques will result in boosting DDoS attack traffic, for example, ICMP traceback.
- Implementing certain technique will cause the traffic when there is no DDoS attack, which is not encouraging, for example: single IP traceback.
- DoSTRACK approach seems to encouraging but it also has some limitations such as
- It works well for spoofed address but what will the case when attacker does not use the spoofed address. As nowadays, attacker needs not to spoof the source address.
- Also victim has to wait for certain threshold to initiate the attack prevention. Generally, ICMP (ECHO (PING) request) packet by default contains 32 bytes of data. If we receive the ping request packet which is carrying a payload for example say 1000 bytes, it cannot be considered as normal ping packet. So after receiving the ping packet of such payload, the immediate action is required without waiting for the certain threshold.
- How to send alert to a host who is unknowingly acting as bot?

The proposed technique will focused on ICMP ping flood attack. The technique tries to overcome the limitations of previous proposed concepts.

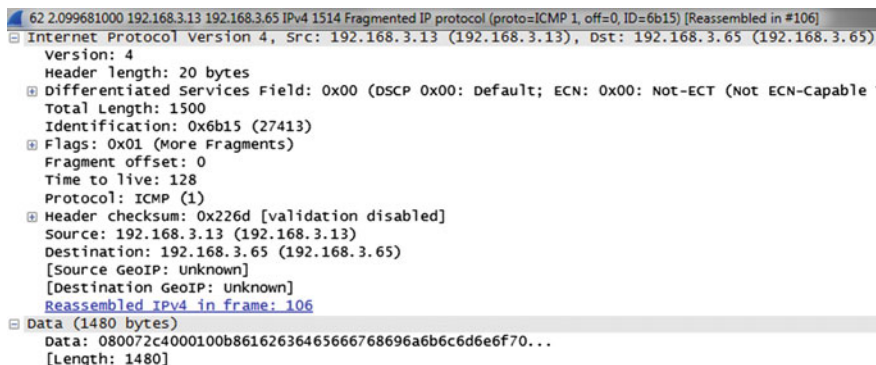


Fig. 9 Wireshark screen shot-2

3 Proposed Work

This section contains the description of our proposed concept. Approach is to follow the distributive defence, when victim is receiving ping flood attack (DDoS attack). To understand the proposed work let us consider the Fig. 9 given below:

Before we start discussing the proposed model, consider the following assumptions:

- Routers are not compromised.
- Routers have capability to inspect the packet or routers are enabled with intrusion prevention system (IPS).

3.1 Case 1

When victim started receiving several ICMP echo request.

Subcase-1: when attacker does not spoofed the source address of botnets.

Suppose victim started receiving the ping flood attack. The end router-1 (R_1) which is connected with victim will generate an ICMP echo reply which will contain an alert message. R_1 will generate an alert message depending on certain conditions such as if it crosses a threshold of count, i.e. T_{count} or packet size is greater than normal ping packet size (by default windows ping with 32 bytes of data). R_1 will write three things in alert message, i.e. ACTION, message to source address (MtSA) and next reset time (NRT). R_1 will write these three things in

ICMP echo reply (data portion of the packet which contains default data. For example, suppose we are using windows OS, when we ping, the default data which a ping packet contains is abcd...w, repeatedly). In the data portion of ping packet, we can write these three things. R_1 will issue an alert message to all those router through which it is receiving the attack. Here in this case R_1 will issue an alert message to R_2 , R_4 and R_5 .

After receiving the alert message from R_1 , R_2 will start dropping the ping packets whose destination address is victim IP address (VIP). R_2 will forward this packet to its upstream router, i.e. R_3 . R_3 after receiving the ICMP echo reply packet (which contains the alert message) will start dropping the ping packets whose destination address is same as VIP. From the above example, it is clear that one or more innocent bots are connected to this router, i.e. R_3 . Suppose IP address of bots (host) is not spoofed by the attacker. It means here we can assume that ICMP echo reply will reach to these bot or their end router. In this case, R_3 which is IPS enabled, will inspect the data field ICMP echo reply packet and read the ACTION, MtSA, NRT. R_3 will take necessary action that will ensure that in future or at least at that time the host will be prevented from acting as bot. The ICMP echo reply (alert message) will reach to its destination only, if message generating host, i.e. R_1 (in this case), how it chooses TTL field.

Subcase-2: when attacker spoofed the source address of botnets.

Here we have assumed that attacker does not spoofed IP addresses of its bots. Now consider the case, if addresses of bots are spoofed by the attacker; in this case, it is difficult to locate the bot and MtSA is of no use and ICMP echo reply will reach to spoofed address which is not participating in attack. Since ICMP echo reply is small packet, will not create trouble for the spoofed address who receive this. But it will alert the routers through which it pass to drop ping packets.

In similar fashion, all the routers will follow the same strategy during the ping flood attack.

3.2 Case 2

When victim started receiving several ICMP echo reply (reflector attacks).

This is a case in which it is receiving ICMP echo reply from several bots. In such cases, router R_1 will follow the same procedure as in case 1, but here it has to send an extra packet as alert message.

4 Proposed Algorithm

4.1 Algorithm-1 Victim Router

```

if ((length PING (ICMP(echo request))) ≥ 1500 bytes OR
    COUNT = THIGH)
{set (flag = ALERT)

    ICMP(echo reply(data)) ← write (ACTION, MtSA, NRT)
                                // this is one time reply send by route to alert
                                neighbouring nodes

    DROP_ICMP echo request
}
Else {echo reply}

```

4.2 Algorithm-2 Intermediate Router

```

if (ICMP echo reply (flag == ALERT))
{  INSPECT ICMP echo reply (data)
    // check for spoofed source address (Egress Filtering)
If(INTERFACE_ROUTER (ICMP echo request=INTERFACE_ROUTER
allowed))
{FORWARD ICMP echo reply
perform ACTION
    // perform action according to mentioned in the data field
    of ICMP echo reply packet
}
else { DROP
    // ICMP echo reply
}}
else
{ FORWARD ICMP echo reply }

```

4.3 Algorithm-3 Attack Machine

```

if RECEIVED ICMP echo reply (flag == ALERT)
{
    INSPECT ICMP echo reply (data)
    take ACTIONS
}
else do nothing

```

5 Experimental Observations

To better understand the problem of ping flood attack, experiments have been conducted in MBS lab with the help of ten hosts in controlled environment.

The practice of ping flood is performed on three different days (Fig. 4). From the above graph, it is clear that on increasing the number of attacking hosts on y-axis, increase in network utilization is observed on x-axis. It means if the number of attacking host will increase, results in bandwidth consumption of victim host or server will increase.

We also recorded some more observation with the help of Wireshark. Some of the interesting observations are:

- The data which an ICMP echo request carries is default data which contains no meaningful information, for example: a, b, c, d...repeatedly.
- If attacker sends the packet which is of larger size say 65,500 bytes, then it gets fragmented and the ICMP request packet will contain the information about the frame number after which all packets will get reassemble. Consider the screen shots given below:

Conclusion of point number 2 is sending big volume of packet by the attacker to victim, to consume the processor time in rearrangements of packets.

Based on the observations which was obtained through experiments, we have developed some concepts, already been discussed in proposed work.

6 Efficiency (in Terms of Time, Bandwidth and Buffer Size)

Suppose number of packets reaching to end leaf router are $= n$. Let us assume that each packet take t time to drop. Total time consumed by these packets $= n \times t$. Consider two one step upstream router are started dropping. Suppose router

one and two drop x and y packets, respectively, then number of packets reaching to end leaf router are $n - (x + y)$

Conclusion:

- $n > n - (x + y)$ //number of packets now end router has to process is less.
- less packets \rightarrow less processing time
- less packets \rightarrow less buffer size needed to store, avoid legitimate packet to drop
- also alert bot to take security measures.
- No extra packet is needed

7 Conclusion

Conclusions are as follows:

- Approach is distributive, as we are not creating bottleneck at router R1 (considered example) during ping flood. As in some initial approaches, all the filtering is applied at end router.
- We are not creating any extra packet for generating the alert message, i.e. sending alert message in ICMP echo reply (data field).
- If address is not spoofed, then alert message will reach to bot who is unaware about its activity. The bot IPS or end router which is IPS enabled will take certain steps not to participate in attack at present, also in future.
- In case if address of bot is spoofed to fake source address, then it seems that alert message is of no use. But in that case it will also inform the router through which it passes, to drop the ping packet whose destination address is same as VIP. Thus helping in mitigating against the DDoS attack.

Acknowledgments I would like to thanks my guru Dr. B.M. Mehtre whose is also co-author of this paper, for his kind nature, faith and of course the guidance from time-to-time during the fellowship program. I would also like to thanks my parents, whose blessings are constant inspiration to me and also to Varsha Yadav for sharing his valuable time to share her knowledge.

References

1. Mirkovic, J., Hussain, A., Wilson, B., Fahmy, S., Reiher, P., Thomas, R., Yao, W., & Schwab, S. (2007). Towards user centric metrics for denial of service measurement. ExpCS'07, San Diego, CA. 13–14, June 2007. Copyright 2007 ACM.
2. Jin, C., Wang, H., & Shin, K.G. (2003). Hop-count filtering: an effective defense against spoofed DDos traffic. CCS'03, October 27–31, 2003, Washington, DC, USA. Copyright 2003 ACM 1-58113-738-9/03/0010.
3. Peng, T., Leckie, C., & Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the DoS and DDos problems. *ACM Computing Surveys*, 39(1), Article 3 (April 2007), p. 42. doi:[10.1145/1216370.1216373](https://doi.org/10.1145/1216370.1216373).

4. Kumar, S., & Surisetty, S. Microsoft versus apple: resilience against distributed denial-of-service attacks. *IEEE Security and Privacy*, 10(2).
5. Jing, Y., Wang, X., Xiao, X., & Zhang, G. (2006). Defending against meek DDos attacks by IP traceback-based rate limiting. *Global Telecommunications Conference, 2006. GLOBECOM '06*. IEEE, November 27 2006–December 1 2006.
6. Rao, S., & Rao, S. (2011). Denial of service attacks and mitigation techniques: Real time implementation with detailed analysis. This paper is from the SANS Institute Reading Room site©.
7. Kiran Tupakula, U., & Varadharajan, V. (2006). Analysis of traceback techniques. This paper appeared at the Fourth Australasian Information Security Workshop (AISW-NetSec 2006), Hobart, Australia. *Conferences in Research and Practice in Information Technology (CRPIT)*, vol. 54.
8. Zhu, Z., Lu, G., Chen, Y., Judy Fu, Z., Roberts, P., & Han, K. (2008). Botnet research survey. *Annual IEEE International Computer Software and Applications Conference*, © 2008 IEEE.
9. Kiran Tupakula, U., Varadharajan, V., & Pandalaneni, S. R. (2009). DoSTRACK: a system for defending against DoS attacks. *SAC '09 Proceedings of the 2009 ACM Symposium on Applied Computing* (pp. 47–53). New York, NY, USA: ACM: ©2009.
10. Yuan, D., & Zhong, J. (2008). A lab implementation of SYN flood attack and defense. *SIGITE '08 Proceedings of the 9th ACM SIGITE Conference on Information Technology Education* (pp. 57–58). New York, NY, USA: ACM ©2008.
11. Bolz, C., Romney, G. W., & Rogers, B. L. (2004). Safely train security engineers regarding the dangers presented by denial of service attacks. *Proceeding CITC5 '04 Proceedings of the 5th Conference on Information Technology Education* (pp. 66–72) New York, NY, USA: ACM ©2004.
12. Batham, S., Yadav, V. K., & Kumar Mallik, V. K. (2014). ICSECV: An efficient approach of video encryption. In *Proceedings Contemporary Computing (IC3), 2014 Seventh International Conference*, 7–9 August 2014, pp. 425–430 (Available at IEEE Xplorer and DBLP, indexed by SCOPUS).
13. Batham, S., & Yadav, V. K. et al. (2013). A new video encryption algorithm based on indexed based chaotic sequence. In *Proceedings Fourth International conference Confluence 2013: The Next Generation Information Technology Summit*, September 27–28 (pp. 139–143). Available at IET and IEEE xplorer.
14. Yadav, V. K., & Batham, S. et al. (2013). Hiding large amount of data using a new approach of video steganography. In *Proceedings Fourth International Conference Confluence 2013: The Next Generation Information Technology Summit*, Sept 27–28, pp. 337–343 (Available at IET and IEEE xplorer).

Webliography

15. <http://blog.radware.com/security/2013/05/how-much-can-a-DDos-attack-cost-your-business/>.
16. <http://www.scmp.com/news/hong-kong/article/1534725/cyberattack-threatens-derails-occupycentrals-unofficial-referendum>.
17. <http://www.digitalattackmap.com/#anim=1&color=2&country=ALL&time=16238&view=map>.

Proceedings of International Conference on ICT for
Sustainable Development

ICT4SD 2015 Volume 1

Satapathy, S.C.; Joshi, A.; Modi, N.K.; Pathak, N. (Eds.)

2016, XX, 757 p. 426 illus. in color., Softcover

ISBN: 978-981-10-0127-7