TẠP CHÍ KHOA HỌC ĐẠI HỌC HẠ LONG
Scientific Journal of Ha Long University
http://uhl.edu.vn/

TẤN CÔNG TỪ CHỐI DỊCH VỤ VÀ CƠ CHẾ PHÒNG THỦ

Ngô Hải Anh[1*], Lương Khắc Định[2]

[1]Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam
[2]Trường Đại học Hạ Long

** Email: ngohaianh@ioit.ac.vn*

Ngày nhận bài: 23/05/2024 *Ngày nhận bài sửa sau phản biện: 12/02/2025* *Ngày chấp nhận đăng: 19/02/2025*

TÓM TẮT

Trong bài báo này, nhóm tác giả nghiên cứu một số phương pháp tấn công từ chối dịch vụ (DoS), tấn công từ chối dịch vụ phân tán (DDoS) và giới thiệu một số cơ chế phát hiện và ngăn chặn. Tấn công DoS/DDoS cho đến nay vẫn là kiểu tấn công rất khó để ngăn chặn hoặc phòng tránh hoàn toàn. Các cơ chế được tìm hiểu trong bài báo này chủ yếu có hiệu quả với các cấu trúc mạng cỡ nhỏ, nhưng cũng có thể được tham khảo để áp dụng mở rộng sang các mạng có quy mô lớn.

Từ khóa: *tấn công từ chối dịch vụ, tấn công từ chối dịch vụ phân tán, cơ chế phòng thủ, truy vết địa chỉ IP, bộ lọc gói.*

DENIAL OF SERVICE ATTACKS AND DEFEATING MECHANISMS

ABSTRACT

This paper investigates some Denial-of-Service (DoS) and Distributed DoS (DDoS) attack methods and introduces detection and defeating mechanisms. Detecting and preventing denial of service attacks is generally quite complex; to date, no solution can completely prevent these attacks. The solutions examined in this paper are only relatively effective for small-structure networks but can also apply to large networks.

Keywords: *DoS, DDoS, defeating mechanisms, IP traceback, packet filtering.*

1. INTRODUCTION

Today, DoS (Denial of Service) attacks are part of every Internet user's life. They are happening all the time, and all Internet users, as a community, have some part in creating them, suffering from them, or even losing time and money. DoS attacks can take several forms and be categorized according to several parameters. The main distinction between the DoS attack types considers where the attack's origin is being generated.

"Normal" DoS attacks are generated by a single host (or a small number of hosts at the exact location). The only way for DoS attacks to impose a real threat is to exploit software or design flaws. Such flaws can include, for example, wrong implementations of the IP stack, which crash the whole host when receiving a non-standard IP packet (for example, ping-of-death).

Such an attack would generally have lower volumes of data. Unless some exploits exist at the victim hosts, which have not been fixed, a DoS attack should not pose a real threat to high-end services on today's internet.

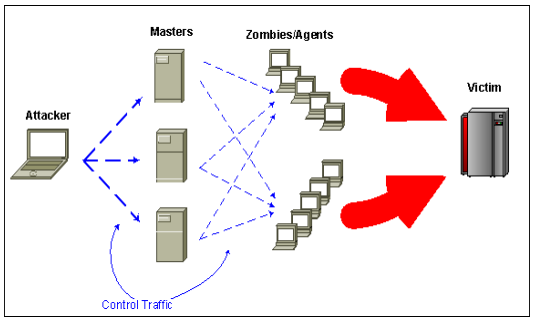

Figure 1. DDoS attacks

Many hosts would usually generate DDoS (Distributed Denial of Service) attacks. These hosts might be amplifiers, reflectors, or "zombies" planted on remote hosts and waiting for the command to "attack" a victim. It is common to see attacks generated by hundreds of hosts, generating hundreds of megabits per second floods. The primary tool of DDoS is bulk flooding, where an attacker or attackers flood the victim with as many packets as possible to overwhelm the victim (Dhanapal & Nithyanandam, 2019).

DDoS attacks are standard today, and they pose the main threat to public services because when a distributed attack is being generated against an internet service, it is hard to block thousands of hosts sending flood data or even legitimate requests. Another aspect of most DDoS is that they consume vast resources from the network infrastructure, such as ISP networks and network equipment. This fact makes such attacks even more troublesome because a single attack targeted against a minor web server might bring the whole ISP's network down and, with it, affect service for thousands of users (Verma et al., 2021).

2. TYPES OF DDOS ATTACKS

DoS attacks can be classified into several major attack types. Classification can be made in several ways. A common way to classify DoS attacks is:

– Application attacks – would usually include attacks that target specific applications. Usually, such attacks would exploit known vulnerabilities in the attacked application, sending malformed application requests or any other type of traffic known to break the application. Usually, such attacks would not require massive amounts of network or computing resources (Mohammed Sharif & Beitollahi, 2023).

– Operating system attacks – would usually exploit known vulnerabilities in specific operating systems. OS attacks would usually exploit some implementation loopholes in the IP stack, sending malformed IP packets to crash the operating system (Değirmenci et al., 2023).

– Network device attacks typically target devices to either degrade or crash their performance. Usually, attacks against network devices would try to exhaust hardware resources on the network device, such as buffer memory or CPU. Network devices include routers, firewalls, gateways, etc. (Ye et al., 2018).

– Flood or Overload attacks – attacks that may use any exploit from the previous attacks' types or any other type of traffic and create a massive volume. In many cases, the mere volume of the attack would be enough (Bawany et al., 2017; Prasad et al., 2014).

3. DEFEATING AGAINST DDOS ATTACKS

Because DoS and DDoS attacks have become a real threat to public and private services on the internet, there is a constant demand to fight back against attacks and try to block them. We would explore the means available to block attacks or make life much more challenging for an attacker to bring services down. This is a continuing effort; attack and defense methods change and evolve with time and experience.

3.1. Patching up the kernel

As we have seen, some DoS attacks take advantage of known operating system weaknesses, protocols, and other mechanisms. Many weaknesses have been fixed or made more robust to make systems more secure. Also, we have seen that many DDoS attacks are possible because many Internet hosts are prone to be exploited and broken into. Such hosts can become infected

with trojan software and a DDoS zombie host. In many cases, fixes to these exploits would be released by the software vendor or by some other third party.

The general idea is that systems become more secure as new updates and fixes emerge. However, the system would not be secure, even if the vendor would provide all fixes quickly. Still, system administrators all over the world must update the systems. To make a system secure, it must be maintained and updated constantly. The system administrator must follow announcements about new releases and updates and install them quickly.

3.2. Finding the source of the attack

One of the issues with DoS attacks is finding out who the attacker is so that we can stop the attack at its source and maybe even prosecute the offender. The problem with finding the source of the attack is that it is easy to hide the actual origin of the attack. It is very common to see attacks coming in with many source IP addresses. Many times, these addresses are being spoofed and generated randomly. This makes it quite hard to know where the attack came from because we do not have any information about the actual source of the attack. We would explore a few methods that help trace the origin of a traffic flood. Some of these methods can and are being used by network operators. Other methods were suggested but have not been made possible for different reasons.

3.2.1. Router by Router, Hop by Hop

When an attack flood arrives at a victim host, it is easy to determine which router is handing off the traffic to the victim. It would usually be the segment's gateway where the victim is connected.

This router can be the first clue for tracing the attack. The idea is to look where the flood enters this router to find the previous hop passing the flow. If we locate the interface bringing the flow into the router, we can look up the router connected at the other side of this link.

This process can be repeated on any router along the path. The last router in this chain would reveal the actual originator of the attack's flow.

Many routers offer tools that enable network administrators to accomplish this task.

For example, Cisco routers support a feature called NetFlow (Cisco, 1999). This feature enables the collection of real-time information about all flows flowing through a router. The NetFlow feature enables network administrators to analyze the collected information to identify attack flows and trace them back by following the origin of the flow at each router.

3.2.2. Packet marking

To be able to trace back the origin of an attack, it is possible to generate additional information generated by routers in the path of the attack flow to provide the destination of the flow (the attack victim) with enough information to reconstruct the flow’s path, and eventually find the actual source of the attack.

Several methods have been offered to provide the infrastructure for this network feature. None has been commercially deployed. We would explore 2 of the suggested algorithms to explore their possibilities.

Savage, Wetherall, Karlin, and Anderson have devised a method called Traceback or edge marking (Savage et al., 2000). This method requires routers in the path of a flow to add information into the packets as they are being forwarded to the victim. A probability p is defined at all routers. A packet would be marked with additional information using this probability, ensuring that for large flows with many packets, we would have enough packets marked by any router in the path.

To avoid adding more overhead to the IP traffic, the Identification field in the IP packet is used to add the path information. This is a 16-bit field that usually does not have any actual use (except with fragmentation – which can create back compatibility issues). The idea is to inject information about edges

(or links) in the packet's path. An edge comprises the IP addresses of 2 adjacent routers and a distance counter, which would tell the victim the distance to this edge. Because all the information is longer than 16 bits (we need enough room for 2 IP addresses and an 8-bit counter, which requires a total of at least 72 bits), the information is compressed in such a way that the whole information block would be delivered to the victim only over a few separate packets.

The compression also adds a hash value used to authenticate the information passed on by the routers, making it hard for the attacker to falsify the information.

Another path reconstruction method was offered as an IETF draft (Bellovin et al., 2003). This method defines a new ICMP message, which is generated by routers in the path of the flow (ICMP Traceback message). Each router should generate these ICMP messages with a low probability (the draft offers a probability of 1/20,000) for every flow. The ICMP message would include information about the routers one hop away from the originating router in both directions. In such a way, the victim host would receive information about edges in the path of the packets in the flow. Also, some original packets would be placed into the ICMP message so the victim host could correlate all the information. Some authentication data is also suggested to make the whole system as secure as possible.

3.2.3. Pushback

Another active mechanism offered to allow automatic filtering of DoS attacks is pushback. Pushback does not offer a real solution for finding the source of a DoS attack but lays an infrastructure for blocking such attacks as close as possible to their source or sources. Pushback has been offered by Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker as a method of detecting and controlling DoS traffic on the internet (Mahajan et al., 2002).

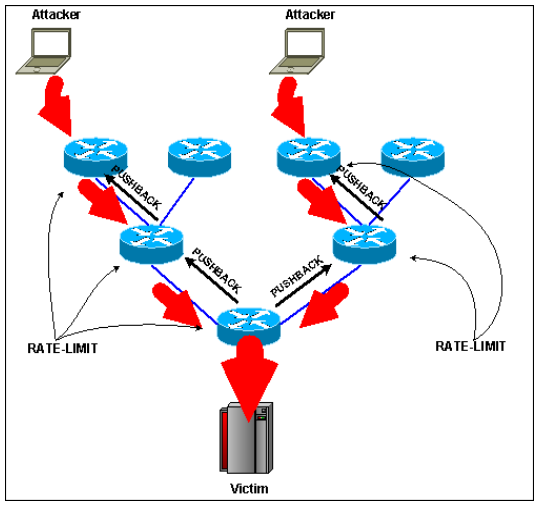

Figure 2. Pushback

Pushback is divided into three main functions. The first stage in the process is to identify which flows generate most of the traffic and classify these flows into manageable groups.

After classification has been made, the next stage is to start rate-limiting the offending flows on the local router while monitoring the effect of the rate limit locally.

The last stage is the actual pushback mechanism. The local router would advertise the information about the offending flows to the upstream routers, requesting them to rate limit the offending flows closer to their origin.

This algorithm can be activated recursively, propagating through the upstream links till it reaches the routers closest to the source or sources of the DoS attack, effectively blocking the attack (see Figure 2).

3.3. Using routers to fight against DoS attacks

As we have seen, DoS attacks are a reality on the Internet. If we examine the Internet's components, we will find that routers are the most common component (maybe except computers…). Router devices are the real core of the internet, passing IP traffic all over it. Routers, critical components for regular Internet operation, are also critical for DoS attacks. All DoS attacks pass through at least

a few routers, while the packets travel from the source of that attack towards the victim. Recognizing this fact, we can use the routers as a tool against DoS attacks by carefully using existing tools inside the core and edge routers to construct the Internet. In this part, we will explore a few functions a router can perform to fight against DoS attacks.

3.3.1. Ingress filtering and good policy

Any network connected to the Internet has to use some routing equipment to forward packets from and to the Internet. This router should be able to provide essential services such as filtering traffic according to a pre-defined policy (Baker, 1995). Let us examine the traffic usually passing through the Internet. We can map a large set of rules that would describe types of traffic that are not expected to be routed on the Internet, and filtering it would not affect everyday user applications. This can be done by analyzing the applications used on the Internet and the kind of traffic they require. Another way of creating the filtering policy is to examine the kinds of traffic used in DoS and DDoS attacks and try to generate some filtering rules to block this kind of traffic explicitly.

After constructing such a set of rules, we can activate this policy on the ingress of all networks connected to the Internet, blocking as much malicious traffic as possible from being propagated. This policy is called Ingress Filtering and is suggested in RFC 2827 (Senie & Ferguson, 2000).

Examining the applications used by users on the Internet, we can find a set of rules that would describe traffic that is not expected to be received on the edge of any network.

Such traffic includes the following rules:

– Spoofed IP addresses – Any network connected to the Internet is assigned some range of registered IP addresses. It is impossible that any other network was assigned the same range, so we can expect that this range of IP addresses will not be used outside the local network. Any packet arriving at the network's edge using a source address with the local IP range can be safely filtered.

– Private IP addresses – RFC 1918 has defined a few ranges of IP addresses that can be used for several purposes (such as NAT, etc.). The RFC defines such addresses as being filtered on the Internet core. No packets using such an IP address as source or destination are legal and, therefore, can be safely filtered at any network edge.

– Unique IP addresses – Many other IP addresses have been assigned for special uses. Examples of such IP addresses are the 127.0.0.0 range used for loopback addressing, class D addresses assigned for multicasting, etc. These IP addresses are not expected to be used as source IP addresses for incoming traffic and can be safely filtered (IANA, 2002).

– Illegal IP address – IP addresses such as 0.0.0.0, 0.1.1.1, or even 255.1.1.1 are not permitted and can be filtered. Also, many IP ranges are known to be bogus or simply unallocated and can be filtered.

Using ingress filtering on the edge of a network will not block all kinds of DoS attacks but may provide a good policy that can, in some cases, reduce the amount of traffic generated by DoS attacks. Ingress filtering is one of the best ways to reduce spoofed Internet traffic. If service providers and network administrators follow this policy and deploy such filtering schemes, many kinds of attacks that rely on the ability to spoof IP addresses will become obsolete.

3.3.2. Null routing

One of the most effective ways of blocking DoS and DDoS attacks is by blocking them on core and edge routers as close as possible to the attack's origin. When an attack is found, we would like to filter all its traffic so that downstream routers can operate normally. Examining the tools available on current routers, we see that almost all routers offer some filtering tool based on a set of rules, such as access lists.

Using these tools may provide a good solution for low-volume DoS attacks but may become ineffective when the volume of an attack is very high. Usually, such filtering

policies are implemented in software and require that all traffic traverse a relatively slow and CPU-intensive path inside the router. When dealing with high-volume attacks, this slow forwarding path may present a real burden for the router and even might crash it. To avoid this problem, we would like to block such attacks without requiring any special treatment by the router for the attack's traffic. This can be accomplished by modifying the router's forwarding table, which is used for regular packet switching. Because of the router's highly optimized packet forwarding process, we would not add any overhead by applying a blocking policy.

The way we change the routing table is crucial for blocking an attack. Many routers include a bit-bucket interface (such as the Null 0 interface on Cisco routers). All packets that are destined for this interface will be dropped by the router.

For example, if we enter the following command on a Cisco router, all packets sent to the 192.168.1.0 prefix will not be routed:

router(config)#ip route 192.168.1.0 255.255.255.0 null 0

The problem with this blocking method is that it operates on the network layer and would not allow the network administrator to block packets based on TCP port numbers or other transport or application layer information. This may cause a problem if a web server is flooded with UDP packets. We would have to block all traffic to it, including packets with HTTP traffic, effectively blocking the attack and the service itself. Another issue with this blocking scheme is that it can block traffic only based on destination IP addresses because routers perform routing based only on this field.

Some extensions provide the same ability for filtering based on source IP addresses, but they are less commonly used (Greene & Jarvis, 2001). This blocking method has been made even more helpful by a solution devised by UUNet engineers, which UUNet is using on their North American network (AS701). Barry Raveendran Greene describes this method in Remote Triggering Black Hole Filtering (Greene, 2006).

4. CONCLUSIONS

As we have seen, denial of service attacks is a living threat to the Internet. It is easy to mount a massive attack against anyone. There are many available tools today that make this task as easy as downloading a file from the Web. On the other hand, protecting against DoS attacks is challenging, requiring a deep understanding of the protocols and applications involved and coordinating a network-wide effort. Many methods have been offered to enable the Internet community to deal with DoS attacks. These methods have not been implemented for many technical and social reasons. Other methods have been deployed to help in the day-to-day war against DoS. As we have shown, typical end users on the Internet hold many aspects of DoS, especially DDoS. Most of these aspects are related to good policy and responsible behavior. If the millions of users represented by normal home users and more extensive enterprise networks would employ a strict security policy, using anti-virus and firewall products, the vast amount of resources available today for DDoS would diminish, and with it, the danger of such attacks would become less noticeable. The solution for DoS attacks is not entirely technological. The same technology is available to both attackers and victims; any advantage is temporary. The accurate and complete solution must come from the social aspect of the end user grasping his need to use a good security policy, denying the attacker vast attack resources.

REFERENCES

Baker, F. (1995). *Requirements for IP Version 4 Routers* (Request for Comments RFC 1812). Internet Engineering Task Force. https://doi.org/10.17487/RFC1812.

Bawany, N. Z., Shamsi, J. A., & Salah, K. (2017). DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions. *Arabian Journal for Science and Engineering*, *42*(2), 425–441. https://doi.org/10.1007/s13369-017-2414-5.

Bellovin, S. M., Leech, M., & Taylor, T. (2003). *ICMP Traceback Messages*. Columbia University Academic Commons. https://doi.org/10.7916/D8FF406R.

Cisco. (1999). *NetFlow Services and Applications* [White Paper].

Değirmenci, E., Kırca, Y. S., Yolaçan, E. N., & Yazici, A. (2023). An Analysis of DoS Attack on Robot Operating System. *Gazi University Journal of Science*, *36*(3), Article 3. https://doi.org/10.35378/-gujs.976496.

Dhanapal, A., & Nithyanandam, P. (2019). The Slow HTTP DDOS Attacks: Detection, Mitigation and Prevention in the Cloud Environment. *Scalable Computing: Practice and Experience*, *20*(4), Article 4. https://doi.org/10.12694/scpe.v20i4.1569

Greene, B. R. (2006). *Remote Triggering Black Hole Filtering* [White Paper]. Cisco Systems.

Greene, B. R., & Jarvis, N. (2001). *Unicast Reverse Path Forwarding (uRPF) Enhancements for the ISP-ISP Edge* [White Paper]. Cisco Systems.

IANA. (2002). *Special-Use IPv4 Addresses* (Request for Comments RFC 3330). Internet Engineering Task Force. https://doi.org/10.17487/RFC3330.

Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V., & Shenker, S. (2002). Controlling high bandwidth aggregates in the network. *ACM SIGCOMM Computer Communication Review*, *32*(3), 62–73. https://doi.org/10.-1145/571697.571724.

Mohammed Sharif, D., & Beitollahi, H. (2023). Detection of application-layer DDoS attacks using machine learning and genetic algorithms. *Computers & Security*, *135*, 103511. https://doi.org/10.-1016/j.cose.2023.103511.

Prasad, K. M., Reddy, A. R. M., & Rao, K. V. (2014). DoS and DDoS Attacks: Defense, Detection and Traceback Mechanisms -A Survey. *Global Journal of Computer Science and Technology: E Network, Web & Security*, *14*(7).

Savage, S., Wetherall, D., Karlin, A., & Anderson, T. (2000). Practical network support for IP traceback. *ACM SIGCOMM Computer Communication Review*, *30*(4), 295–306. https://doi.org/-10.1145/347057.347560.

Senie, D., & Ferguson, P. (2000). *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing* (Request for Comments RFC 2827). Internet Engineering Task Force. https://doi.org/10.17487/RFC2827.

Verma, P., Tapaswi, S., & Godfrey, W. W. (2021). A request aware module using CS-IDR to reduce VM level collateral damages caused by DDoS attack in cloud environment. *Cluster Computing*, *24*(3), 1917–1933. https://doi.org/10.1007/-s10586-021-03234-2.

Ye, J., Cheng, X., Zhu, J., Feng, L., & Song, L. (2018). A DDoS Attack Detection Method Based on SVM in Software Defined Network. *Security and Communication Networks*, *2018*(1), 9804061. https://doi.org/10.1155/2018/9804061