

# Denial of Service: Another Example

Jianxin Jeff Yan

*Computer Laboratory, Cambridge University, UK.*

Jeff.Yan@cl.cam.ac.uk

**Abstract:** Although denial of service attack has been becoming a fast-growing concern in security research, previous work focused on a type of classical denial of service caused by resource exhaustion. In this paper, a different type of network denial of service attack is discussed. Since traditional models and countermeasures are not applicable, we discuss solutions that can defend this non-classical service denial attack.

**Key words:** Denial of Service, Resource Exhaustion, Unblockable Internet Service

## 1. INTRODUCTION

Security researchers have extensively studied confidentiality and authenticity, but they have much less studied and understood availability issues. As denial of service (DoS) attack has become an increasingly prevalent security threat, people realize that protecting systems against DoS attack is also one of the key security issues. In some circumstances, like the burglar alarm case illustrated by Needham [10], DoS attack was even the only security threat that must be addressed.

Although DoS attack is becoming a fast-growing concern, most research has focused on only one type of DoS attack, where an attacker exploits a design flaw or system bug to exhaust a resource of a victim system, and thus prevent users from accessing the system service, or degrade the service quality that they can get. For example, the early work of DoS in operating systems was about this type of resource exhaustion attack. So was the later work on network DoS attacks and the latest on the distributed DoS attack.

What constitutes a DoS attack? If a service is supposed to be available but it is not, then this service is said to have been denied. Nonetheless, service denial caused by an unintentional behaviour is not a DoS attack, since only an intentional action constitutes an attack. Therefore, a DoS attack occurs whenever access to a computer or network resource, e.g. a user account or network connection, is intentionally prevented or degraded as a result of a malicious action. The attack intentionally compromises the availability of the resource, and it is typically against the will of affected users. Resource exhaustion has been the most popular method to materialise a DoS attack, but it is not necessarily the unique one. In this paper, we look into a type of network DoS attack that is different from the classical resource exhaustion form.

## **2. ANOTHER TYPE OF DENIAL OF SERVICE ATTACK**

This non-classical service denial attack is not new in real life, and it happens everyday in the Internet: some authorities have been keen in blocking Internet-based services such as bulletin board systems (BBS), ftp and web services, due to political or other reasons, by deploying real-time filters, e.g., filtering routers, firewalls or proxy servers. Typically, they blacklist servers hosting services that they do not like, and check each packet on the fly by a real-time filter sitting between the servers and users. The filter drops packets sent from or to those blacklisted addresses, and thus the communication channels are cut off between the servers and the users. The authorities do not exhaust the resource of systems that provide those services. Nonetheless, they do deny users, who are under their control, access to those services by selectively cutting off virtual communication channels. In some scenarios, this service denial does not necessarily constitute a DoS attack. For example, the terms of contract between an Internet service provider (ISP) and a subscriber might have access restrictions because they want to protect themselves from liabilities (specially in a litigious environment), and the subscriber accepts these restrictions. Or, the ISP may simply be courting the custom of subscribers who seek filtering of some sort of contents, e.g. parents who are concerned about their children's Internet surfing. Nonetheless, this service denial is commonly used to seek censorship, asymmetric information advantage or other unfair benefits, and it does constitute a DoS attack in these scenarios. For many Internet based systems that prompt free flow of information or seek for guaranteed service delivery, this blocking DoS attack is a severe security threat to be considered.

Technically, there are two typical properties of this blocking DoS attack: 1) it is a DoS attack where only partial users, i.e. people under the control of the authority, are denied access to the attacked service; 2) the service itself runs correctly all the time, no matter whether this partial DoS attack happens or not. To the contrast, when a service is hit by an effective DoS attack of resource exhaustion form, it will malfunction in certain way, and any user could be deprived of service access.

A simple framework may be used to clarify different DoS threat models as follows.

(a) Traditional DoS Model

Assume a service,  $X$ , and a set of users {user 1, user 2, ... , user  $n$ }. In this model, all users are denied access to  $X$ . There is no selection made regarding which user is denied access.

(b) User-focused DoS Model

Given the same scenario as in (a), selected users (e.g., user 1 and user 2, but not users 3 to  $n$ ) are denied access to  $X$ . The motivation for this attack is to deny certain users access to service or information.

(c) Service-focused DoS Model

Given the same user base, but assume the existence of a set of services  $\{X_1, X_2, \dots, X_n\}$ . Access to a particular service could be denied. For example, access to  $X_2$  could be blocked, but  $X_1, X_3$  to  $X_n$  are all accessible.

(d) Hybrid DoS Model

This is a combination of (b) and (c). Certain users could be denied access to a particular service.

This informal framework helps highlight where the threats are. For example, in (a) and (c), the DoS attack attempts to block access to a service. There is no particular concern as to the users who access the service, but a service's identifier is crucial in the attack. While in scenarios (b) and (d), user identifier is used in the attack.

### 3. DEFENCES

Previous research mainly tackled denial of service attack as a resource exhaustion problem, and most (if not all) proposed defences were about resource allocation. For example, as the first piece of work in DoS, Gligor showed that inter-user dependency was a common cause of service denial in

centralised multi-user operating systems, and this kind of DoS attack should be approached through resource allocation rather than access control [5]. Later, Millen proposed a resource allocation model for DoS [8]. Similarly, the later network DoS attacks - the TCP SYN flooding [12] was the most well known example - have been considered a problem of server resource exhaustion. Many techniques were proposed to design DoS resistant network protocols. For example, stateless protocol [3] tried to trade off the need for state storage space in the server side to an increasing protocol message size and time-consuming cryptographic computations. Client puzzle [7] managed to force a client to commit his resource first, and a server only allocated resource after the client commitment was verified. Distributed DoS (DDoS) was the latest network service denial attack. Except the attacking packet flood was generated in a distributed way, and coordinated, a DDoS attack exploited much the same vulnerabilities as previous network service denial attacks. XenoService [15] was designed to respond to a DDoS attack on a service by replicating it rapidly and widely to a distributed network of XenoServers, although it considered the economic incentive issue, which was a new angle about DDoS defence, its crucial design philosophy was still about resource allocation. Unfortunately, all of these countermeasures addressing resource exhaustion cannot provide an effective defence against the blocking DoS attack discussed in this paper.

The Eternity Service [1] was a seminal work on availability of online publication system, and it proposed a solution to fight technical and legal censorships. Nonetheless, the main security threat it addressed was deletion of online published materials, and its design was targeted at anonymous and persistent storage rather than frequent querying. The blocking DoS attack imposes a different threat model. On the other hand, the Eternity Service could not guarantee those undeletable materials freely accessible to all users, and it might also be vulnerable to the blocking DoS attack.

New remedies are needed. Our solution against service blocking is to fool or wear off the blacklisting mechanism. In this section, we discuss how to implement **unblockable Internet services** to defeat the IP blocking DoS attack. We take web service as our example. Several possible schemes are as follows.

### 1. HTTP Redirect

The HTTP protocol since version 1.0 [4] has supported “HTTP redirect”, where a browser seeking a particular resource is told to go elsewhere for it. With the help of an intelligent portal server, this capability can be used to automatically guide the browser to an appropriate place. Once this transfer of control takes place, the browser will continue to communicate with the

second server for the rest of the session. Many big websites have been using “HTTP redirect” to distribute system load to different servers with different DNS names and different IP addresses. In a same way, HTTP redirect can be used to bypass devices used for blacklisting. If only the IP address of the portal server is blacklisted, users may have access to the service by directly querying anyone of those redirecting servers.

## 2. Dynamic DNS

Commonly, a website has a unique domain name (i.e. DNS name), which matches a unique IP address. A website is blocked when its server IP address is blacklisted. Dynamic DNS [14] enables different web servers with different IP addresses to share a single DNS name, and this technique is also useful for defending the blocking DoS attack.

DNS servers are usually implemented in such a way that all servers for a particular name zone carry the same data, so when anybody issues a DNS query at any place all over the world, he will get a same IP address for the inquired DNS name. It, however, is valid to setup many name servers, each serving an authoritative but different IP address for a same DNS name. Thus, when a user browses a website, the web server he will access depends on which authoritative name server his local DNS server happens to ask. By using dynamic DNS update [14], each particular name server can be instructed to dynamically update its record for a web server with a different IP address where users are expected to go. In case a web server is blocked, its authoritative name server can be instructed to configure that web server to another valid IP address. Therefore, only blocking one IP address for a DNS name does not work any more.

## 3. Volunteering relay

There are two disadvantages with HTTP redirect and dynamic DNS update: 1) they require lots of resources that are usually beyond the affordability of small organisations, and 2) when an authority detects all IP addresses used for redirecting servers or DNS update, he can easily block the service. Volunteering relay provides a better solution as shown in Figure 1. Each  $Proxy_i$  ( $i = 1, 2, \dots, n$ ) is a relay proxy installed in a volunteering relay machine. When a web server receives a request from a browser, it will randomly choose a  $Proxy_j$  ( $1 \leq j \leq n$ ) to relay the communications between itself and the browser. Thus, for a real-time filter, it appears that the traffic is from an “innocent” server, rather than a blacklisted one. It is like that the real IP address of the web server is hidden by spoofing. Therefore, it is impossible to block a service by blacklisting its server IP address only.

When  $n$  is big enough, it is impractical to block all volunteering relays. Experiments showed that a CISCO enterprise router lost much performance when its entire 200 lines of filtering rules were fully utilized [9]. To block many volunteering relay machines will unavoidably cause a severe or even unacceptable network performance degradation.

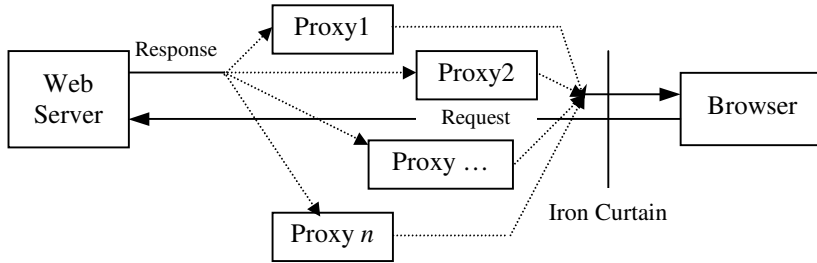


Figure 1. Volunteering relay

When enough volunteers participate, the scheme of volunteering relay may provide a cheap solution to achieve unblockable Internet service

4. Virtual proxies with randomised IP addresses

Another defence scheme is shown in Figure 2. When a web server receives a request from a browser, it initiates a virtual proxy with a randomly chosen IP address, and the proxy relays between the browser and the server so as to hide the server IP address in the traffic. Because each virtual proxy uses a random address each time, it is impossible to block all virtual proxies provided that the used address pool is big enough.

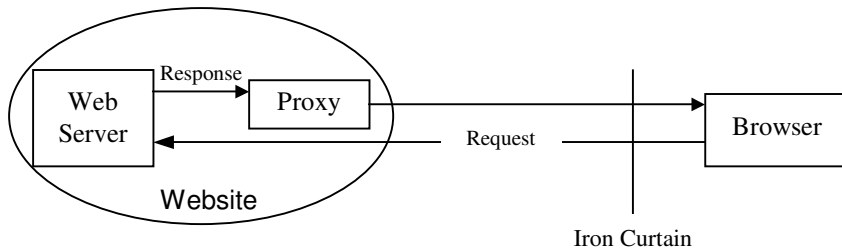


Figure 2. Virtual proxies with randomised IP addresses

When discussing the above four schemes, we assumed that user requests are not blocked, because it will cause severe performance penalty to check each outbound packet. This performance penalty is also one of the reasons

that few ISPs deploy egress filtering<sup>1</sup> in practice, even though they are in the danger of being involved with DDoS attacks [15]. In case an authority even blocks outbound requests, (volunteering) proxies are needed to forward client requests to appropriate servers.

#### 4. ANONYMITY AND AVAILABILITY

Anonymity can improve system availability somehow, since it can be used to hide identifiers, which a selective DoS attack exploits. Needham first pointed out that anonymity of communication would make it difficult to attack a particular user without attacking all users [10]. Nonetheless, an anonymity method protecting privacy well does not necessarily defeat the blocking DoS attack discussed in this paper, because it might address a different threat model and different goal. For example, the Crowds system [11] implemented an idea of “blending into a crowd” in the client side to hide IP address and identity of a browser from a web server. It protected client privacy, but did not prevent the blocking DoS attack.

Some anonymity systems could help users access blocked Internet services, but they themselves, unfortunately, might be vulnerable to the blocking DoS attack. For example, as a proxy service, Anonymizer [2] could help a user browse websites blocked by an authority, because this anonymity system would hide the blacklisted IP addresses in the traffic from and to the user. The authority, however, could blacklist the Anonymizer service itself, and thus deny users access to the Anonymizer (and other blocked services). On the other hand, each Anonymizer proxy was also vulnerable to DDoS and other network DoS attacks. Onion routing [13] provided anonymous connections that were resistant to eavesdropping and traffic analysis, but an application specific proxy (e.g., a web proxy) was needed to bridge a client (e.g., a browser) to the onion routing network. Similarly, this proxy could help users access to blocked services as Anonymizer did, but it also suffered the same attacks. The Freedom network [6] enabled a user to anonymously browse websites with a nym, and it supported encrypted anonymous communication between a browser and a web server. So, the Freedom network could prevent the both IP based and content based blocking. Nonetheless, each Freedom Server, which was supposed to bridge clients to the Freedom network, could suffer the same attacks that might happen to an Anonymizer proxy.

## 5. CONCLUSION

Authorities use real-time filters such as firewalls and filtering routers to block particular Internet services, which, in many scenarios, will lead to a type of DoS attack that is different from the traditional one of resource exhaustion form. This blocking DoS attack imposes a severe security threat to many Internet applications that seek for free flow of information or guaranteed service delivery. Traditional countermeasures, however, are not applicable. Neither anonymity means is necessarily effective in tackling this threat, because it might address a different threat model and security goal, and moreover, an anonymity system itself might also be vulnerable to this blocking DoS attack. Schemes are proposed to implement unblockable Internet services as defence of this blocking threat, and it appears that volunteering relay and virtual proxy with randomised address are feasible and cheap solutions.

## ACKNOWLEDGEMENTS

This work was done by April 2000. The author is grateful to anonymous reviewers of ACM NSPW'01 (New Security Paradigm Workshop) and IFIP SEC2002 for their helpful comments.

## REFERENCE

1. Ross J Anderson. The Eternity Service, Pragocrypt 96, 1996
2. Anonymizer, <http://www.anonymizer.com>
3. T Aura and P Nikander. Stateless Protocols. Proc. ICICS'97, LNCS 1334. Springer-Verlag, 1997.
4. T Berners-Lee, R Fielding, H. Frystyk. RFC-1945: Hypertext Transfer Protocol -- HTTP/1.0, May 1996
5. Virgil Gligor. A Note on Denial-of-Service in Operating Systems, IEEE Trans. on Software Engineering, Vol. SE-10, No.3, May 1984, pp320-324
6. Ian Goldberg, Adam Shostack. Freedom Network 1.0 Architecture and Protocols, white paper of Zero-Knowledge Systems, Inc., 1999
7. A. Juels and J. Brainard. Client Puzzles: A Cryptographic Defense Against Connection Depletion Attacks. In S. Kent, editor, *Proceedings of Networks and Distributed Security Systems (NDSS '99)*, pages 151-165, 1999.
8. Jonathan Millen. A Resource Allocation Model for Denial of Service, Proc of IEEE Symposium on Security and Privacy 1992, IEEE computer society, pp137-147
9. Peter Morrissey. The Cost of Security on Cisco Routers, <http://www.networkcomputing.com/1004/1004ws2.html>, Feb 1999
10. Roger M Needham. Denial of Service: An Example, CACM Vol. 37, No. 11, Nov. 1994

11. M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security* 1(1), November 1998, pp. 66-92.
12. Christoph Schuba, Ivan Krsul, Markus G. Kuhn, Eugene Spafford, Aurobindo Sundaram, and Diego Zamboni. Analysis of a Denial of Service Attack on TCP, *Proc. of the 1997 IEEE Symposium on Security and Privacy*, 1997
13. Paul F. Syverson, David M. Goldschlag, Michael G. Reed. Anonymous Connections and Onion Routing, *Proc. of the 18th Annual Symposium on Security and Privacy*, IEEE CS Press, 1997, pp. 44-54.
14. P Vixie (ed.), S Thompson, Y Rekhter, J Bound. RFC-2136: Dynamic updates in the domain name system (DNS UPDATE), 1997
15. Jianxin (Jeff) Yan, Stephen Early and Ross Anderson. The XenoService - A Distributed Defeat for Distributed Denial of Service, *Proc. of Information Survivability Workshop 2000*, Boston, USA, CMU CERT and IEEE Computer Society, 2000

---

<sup>1</sup> IP spoofing is a basic technique exploited by many network DoS attacks, and it is an important factor that DDoS attack is devastating and difficult to trace. Egress filtering can effectively eliminate spoofed IP traffic.