Poster Abstract: DDoS Attacks Avoidance by Securely Hiding Web Servers

Mohamad Samir A. Eid
Dept. of Electrical Engineering and Information Systems
The University of Tokyo
Tokyo, Japan
mohamadsamir@aida.t.u-tokyo.ac.jp

Hitoshi Aida
Dept. of Electrical Engineering and Information Systems
The University of Tokyo
Tokyo, Japan
aida@ee.t.u-tokyo.ac.jp

ABSTRACT
To protect web servers from flooding Distributed Denial of Service (DDoS) attacks, it is required to stop undesired traffic far from the servers. To avoid non-practical assumptions of modifying the internet core infrastructure to provide such protection, the overlay network protection approach is adopted. However, hiding the web servers behind an overlay network with third party access-nodes raises concerns about the confidentiality and integrity of the data crossing these access-nodes. In this work, special purpose dummy public servers and access-nodes are designed, built and tested to demonstrate their compatibility with end-to-end connection encryption as well as attacks resilience.

Categories and Subject Descriptors
C.2.0 [**Computer-Communication Networks**]: Security and Protection.

General Terms
Security, Design, Reliability, Measurement, Experimentation, Performance.

Keywords: access control; DDoS protection; e-commerce; internet security; privacy.

1. INTRODUCTION
According to the study in [1] including 400 IT decision-makers from companies that operate a significant online business or enjoy and important online reputation, 74% reported that their organizations had been targeted by at least one DDoS attack in the year 2008 alone. Of these, 31% resulted in service disruption. Of the surveyed organizations, 87% will maintain or increase their current budget for DDoS protection in the foreseeable future.

A deployable DDoS defense scheme needs to have practical assumptions and also be compatible with the protected servers' demands. Several Internet based services, such as offered by banks and hospitals require end-to-end encryption, i.e., HTTPS based web servers. A practical and SSL compatible defense architecture is needed for protecting such services.

In this work, we demonstrate the design, implementation and deployment for testing of a practical DDoS protection architecture.

2. Securely Hiding Web Servers

2.1 Overview
Protected web servers are not modified; instead, are required to provide at least one additional public *dummy* server, while the protected server is hidden from direct access, inside a VPN [2]. Protected servers can be accessed by its users only through a set of access-nodes (ANs). The public server implements a *lightweight* protocol that handles the initial request from a client, selects and negotiates with one of the suitable ANs, and then redirects the client to that AN. The ANs are geographically distributed, implementing a special protocol transparent to the client and the protected server.

2.2 Client Connection Procedure
Figure 1 shows a simplified scenario, without loss of generality, with two clients C_1 and C_2. Clients C_1 and C_2 may represent two separate users running on two separate hosts while sharing the same network, or a single user with two separately opened sessions. In either case, each client needs to generate a request, originating from the same source IP address. Assume that the two requests are destined to two different web servers, server X (and server Y), at the same time. If the defense is switched ON; Stage 1: clients C1 and C2 ask the DNS about the IP address of server X (and server Y), respectively, not aware of the defense implementation. The DNS return the public IP address IPXp and IPYp, for the public servers Xp and Yp, respectively. Stage 2: After establishing TCP connection, both clients ask for some resources.

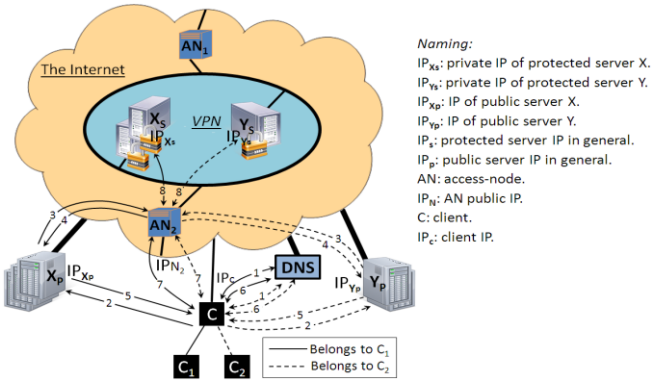

Figure 1 Proposed approach

Stage 3: both Xp and Yp happened to select the access-node AN2 at the same time not aware of each other's choice, and then inform AN2 about IPc and IPs, of Xs and Ys, respectively. This coincidence of selecting the same AN is to demonstrate the AN ability of differentiating between client-server pairs. Stage 4: AN2 replies to Xp and Yp with two distinctive port numbers to be able to differentiate between the two clients' connections originating at the same time from the same IP address (IPc), without having to open the application messages. Stage 5: Xp and Yp relay, back to the clients, the address for the selected AN plus the corresponding port for that connection(s) (i.e. client) in a standard HTTP redirection message. The TCP connection to the client is then closed by the public server. Stage 7: Each client is expected to establish a TCP connection to AN2 using the ephemerally assigned destination port. After the TCP connection is established, the clients now may ask their requested resources from the new location, while the assigned port can be reassigned by the AN to be reused with another client-server pair. Stage 8: AN2 connects to the corresponding servers and communication is carried on.

3. Evaluation

3.1 Prototype Implementation

System prototype was implemented, for the sake of concept verification and empirical service impact evaluation. All components' protocols are realized using JAVA. The prototype is deployed on a small scale experimental test bed of six hosts. An experimental test bed is constructed for the implemented system to be deployed on. Figure 2 shows the topology for the test bed.

3.2 Experiments

Several experiments on the prototype were carried on, evaluating; individual components processing times with and without attacks, as well as system performance impact under several flooding attacks. Tests on the system implementation show its ability to handle request rates much larger than a web server can handle without

Table 1 Test-bed components specifications

No.	Role	Model	CPU	RAM
M1	Public server	Dell Vostro 1200	2.00 GHz II	2GB
M2	Access-node	Dell OPTIPLEX 330	2.53GHz II	2GB
M3	Measurement source	IBM ThinkPad X41	1.5GHz I	1GB
M4	Attack source	Dell OPTIPLEX 330	2.53GHz II	2GB
M5	HTTP server	IBM ThinkPad X41	1.5GHz I	1GB
M6	Attack source	IBM ThinkPad X201i	2.13GHz II	2GB
R1	Public NW	Buffalo WZR-HP-G300NH	AR9132 @ 400MHz	64MB
R2	Hidden NW	I-O Data ETX-R	AMRISC 9041-G	16MB

I: Single core, II: Dual core

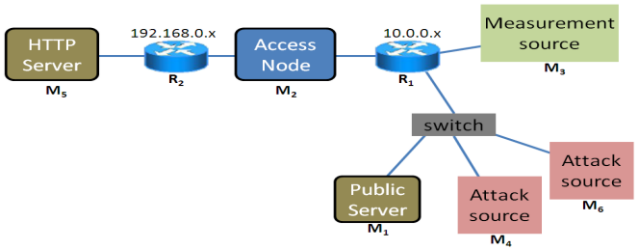

Figure 2 Test-bed network topology

performance degradation, even with implementing, traditional, victim-side, protection methods such as TCP proxy protection and SYN cookies. The measurements on the AN shows a constant service level even with ICMP and TCP based attacks like NAPTHA and SYN flooding. This is assuming the implementation of an efficient detection system at the victim side. The proposed defense mechanism also "raises the bar" for application level attacks; i.e., to achieve the same level of attack rates on the public server, a much larger botnet is required. Similarly, the amount of over-provisioning required at a the protected service is much less than what a non-protected service would require since it is only proportional with the expected clientele of the service, not the expected attack rate.

4. CONCLUSION

The problem of DDoS protection was addressed while stressing on practicality for a more plausible deployability and the importance of compatibility with SSL. Such protection service can belong to a provider with a globally distributed set of data centers, therefore, requiring no modifications to legacy network equipment or protocols. Experiments, so far, show system concept soundness. This is a powerful and secure method of thwarting DDoS attacks, and is most suitable for web servers that serve personal transaction data.

5. REFERENCES

[1] Forrester Consulting, "The trends and Changing Landscape of DDoS Threats and Protection", A study on behalf of VeriSign, Inc., July 2009.

[2] M. S. A. Eid, and H. Aida, "Securely Hiding the Real Servers from DDoS Floods", IEEE 10th Annual Int. Symp. on App. and the Internet (SAINT), July 2010.