An Authentication technique to handle DDoS attacks in Proxy-Based Architectures

Poonam Dharam (Advisor & Associate Professor)
Dept. of Computer Science & Information Systems
pdharam@svsu.edu

Jarin Musarrat (Undergraduate Student)
Dept. of Computer Science & Information Systems
jmusarra@svsu.edu

What is a DDoS Attack?

A distributed denial-of-service (DDoS) attack occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers. Such an attack is often the result of multiple compromised systems (for example, a botnet) flooding the targeted system with traffic.

History of DDoS attacks

- 17% of all internet attacks are based on DoS attacks
- The biggest DDoS attack to date took place in February of 2018 on GitHub. At its peak, this attack had incoming traffic at a rate of 1.3 terabytes per second, sending packets at a rate of 126.9 million per second.
- The 2016 Dyn attack, The 2013 Spamhaus attack, The 2007 Estonia attack, The 2000 Mafiaboy attack

Targets and Cause of DDoS Attacks

- Ideology - “hacktivists” use DDoS attacks as a means of targeting websites they disagree with ideologically.
- Business feuds - Businesses and organization can use DDoS attacks to strategically take down competitor websites or business
- Extortion - In this case the threat of DDoS attacks is performed as a means of extorting money from their targets.

Targets and Cause of Attacks

- Boredom - “script-kiddies” use prewritten scripts to launch DDoS attacks and these attacks are typically performed for personal interest or to test own skills.

- Cyber warfare - Government authorized DDoS attacks can be used to both cripple opposition websites and an enemy country’s infrastructure.

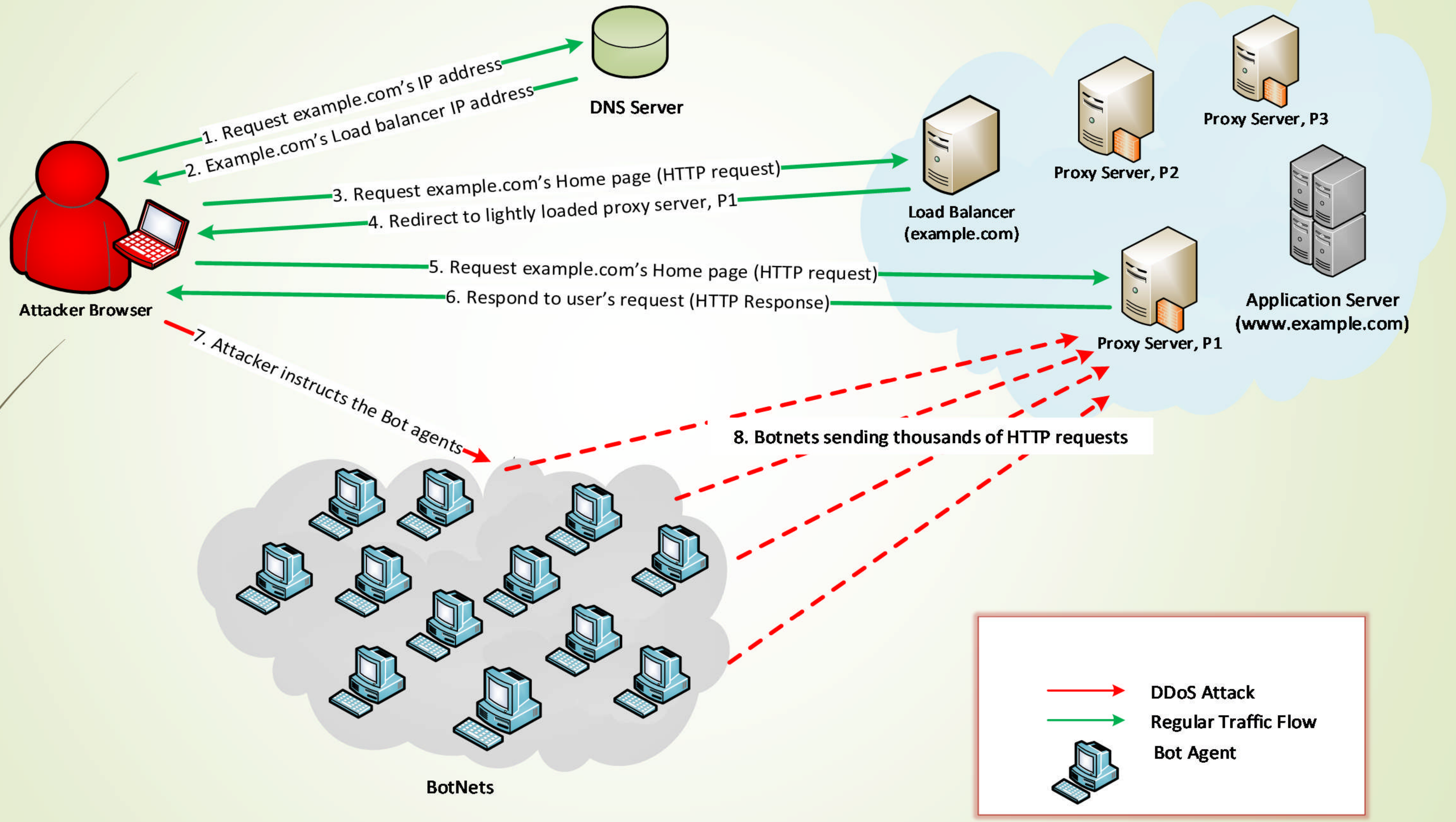

1. Request example.com's IP address
2. Example.com's Load balancer IP address
DNS Server
3. Request example.com's Home page (HTTP request)
4. Redirect to lightly loaded proxy server, P1
Load Balancer (example.com)
Proxy Server, P2
Proxy Server, P3
5. Request example.com's Home page (HTTP request)
6. Respond to user's request (HTTP Response)
Attacker Browser
Proxy Server, P1
Application Server (www.example.com)
7. Attacker instructs the Bot agents
8. Botnets sending thousands of HTTP requests
BotNets
DDoS Attack
Regular Traffic Flow
Bot Agent

Fig1: Example of a DDoS attack in proxy-based architecture.

The Attack

- To understand the limitations of using proxy-based architecture in handling DDoS attacks, we consider a Web service provided by a combination of proxy servers and a backend application server as shown in Figure 1.
- Here the browser communicates with the Load Balancer (Load balancing is a method of distributing network traffic among multiple servers.).
- Also, we consider a client trying to access a Web page www.example.com. Next, we will list the sequence of steps that take place.

Behind The Scene

1.The client types in the URL in the browser;

2.The browser resolves the domain name by talking to DNS server and getting an equivalent IP address (that actually corresponds to the Load balancer's IP address);

3.The browser then sends an HTTP request to the Load Balancer (LB);

4.The LB then finds a proxy server that is lightly-loaded and redirects the user to the assigned proxy server;

5.The client then directly talks to the proxy server.

Attack Explained

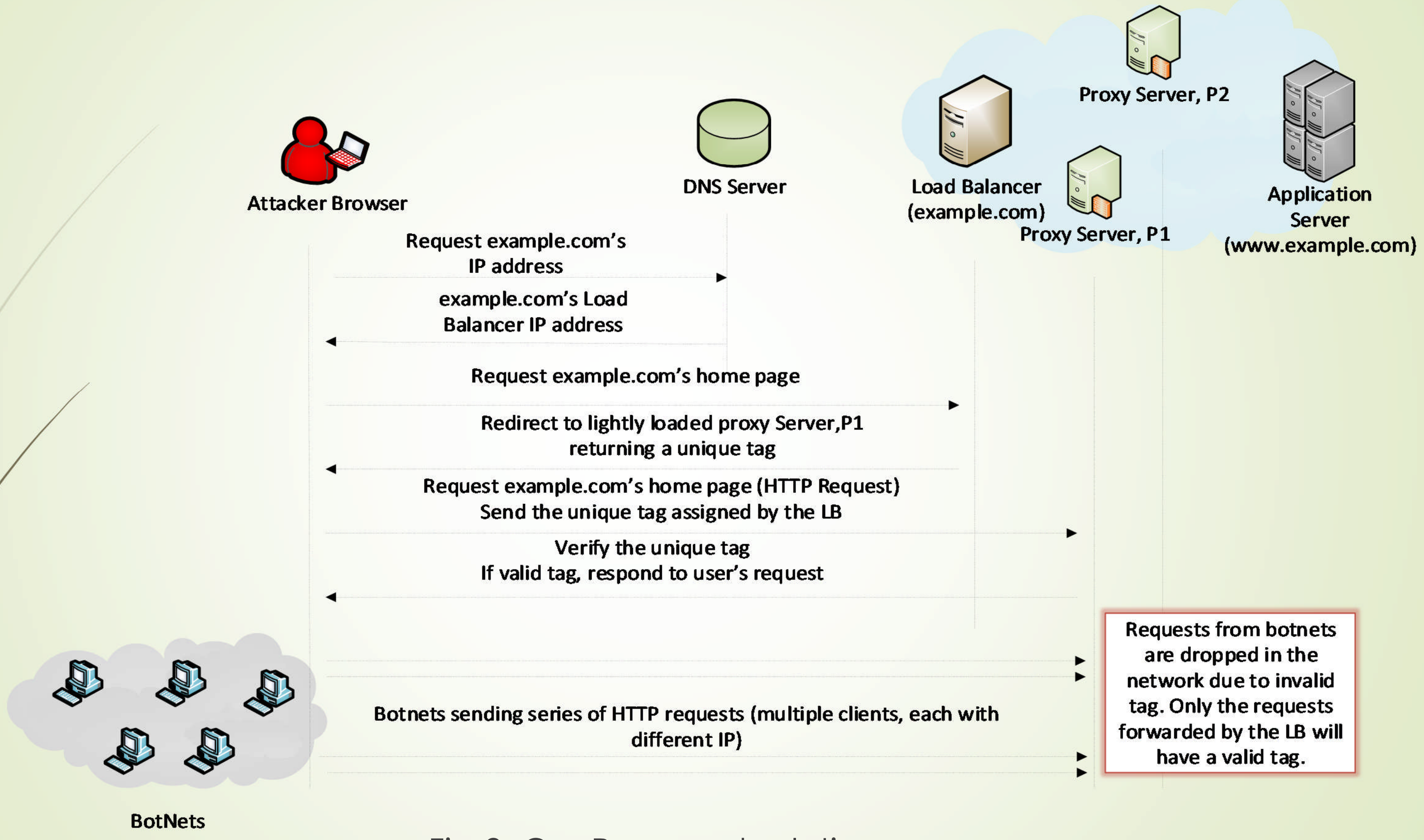
- The DDoS attack exploits the LB-to-proxy redirection scheme.
- LB-to-proxy redirection by domain name requires that clients obtain proxy details by DNS and then contact their proxies.
- Through this process, the attacker learns the IP address of an active proxy. After gaining the IP address of the proxy server, a DDoS attack can be launched by the botnets; generating a huge number of packets with the proxy server's IP address to the destination.
- ***In this case, we focus on handling DDoS attacks in proxy-based architecture.***

Related Work

- Most of the existing solutions [1] – [5] in this area focus either on (a) Moving Target Defense (MTD), or (b) extending the available resources to support increased user requirements.
- In spite of existence of various mitigation techniques, DDoS attacks in proxy-based architecture still continue to exist.
- One of the main reasons is the overhead involved in either migrating the existing clients or spawning additional resources to handle additional traffic.

Our Proposed Solution

- To overcome the identified challenges, in this paper, we design a client-to-proxy assignment and authentication scheme that finds a lightly loaded proxy and returns the IP address along with the unique ID to the client.
- The client then talks to the proxy server by exchanging its unique ID. Only the user with a valid ID is allowed to communicate with the proxy. We thus make sure that every user directed to a proxy is authenticated by the LB.



Proxy Server, P2
Attacker Browser
DNS Server
Load Balancer (example.com)
Proxy Server, P1
Application Server (www.example.com)
Request example.com's IP address
example.com's Load Balancer IP address
Request example.com's home page
Redirect to lightly loaded proxy Server,P1 returning a unique tag
Request example.com's home page (HTTP Request) Send the unique tag assigned by the LB
Verify the unique tag If valid tag, respond to user's request
Botnets sending series of HTTP requests (multiple clients, each with different IP)
Requests from botnets are dropped in the network due to invalid tag. Only the requests forwarded by the LB will have a valid tag.
BotNets

Fig 2: Our Proposed solution

Our Proposed Solution

- Bob's browser talks to the DNS server requesting for example.com's IP address; DNS returns the IP address corresponding to the load balancer (LB);
- Bob's browser then sends an HTTP request to the LB, requesting to access example.com's home page;
- The LB generates a unique tag and returns it to the user along with the IP address of the proxy. The unique tag is generated as a function of proxy servers', LB's, and client's IP address. To avoid the tag being forged, the tag is encrypted using a secret key-shared between the proxy and LB

Our Proposed Solution

- The user then sends an HTTP request to the proxy server along with the unique tag assigned to it.
- The proxy first decodes the unique tag and verifies the credentials present in the tag. On successful verification of the client, the proxy sends a corresponding HTTP response; otherwise, the client request/ connection is dropped;

Our Experimental Setup

For our experiments, we simulate a simple network using socket programming in Java. Each component (LB, clients, proxy servers, and application servers) is a Java class running at localhost i.e., 127.0.0.1.

For our experimental setup, we have a LB that processes incoming requests from the client, and four proxy servers which are in turn connected to the application servers.

Performance Evaluation

We simulated about 50 valid user requests by hosting client programs and sending a simple HTTP GET requests for a valid document available at the Application server.

Additionally, we simulated about 30 requests which were mainly attack traffic.

The way we simulated the attack traffic is described below. We assume user-1 is the attacker.

- A valid user request is sent to the LB from user-1;
- The LB then finds a proxy server with the lowest load and returns the IP address of the proxy with a unique tag that contains user-1's IP address and port number;
- User-1 then generates 30 requests directed to the proxy servers chosen in step-2;

Implementation

- First, the user request arrives at the LB, it generates a unique tag which is a function of source/client, LB, and proxy server's IP address and encode it into a secret tag;
- Next, the request is redirected by the LB using HTTP response 302 Found. The generated secret tag is placed as a part of the Location field in the HTTP response, along with the proxy server's IP address;
- The redirected user request then arrives at the proxy server, where the server first extracts the unique tag and verifies the IP addresses;
- If the secret tag is valid, the user request is processed; else the request is dropped due to invalid/missing secret tags.

Performance Evaluation

- For performance evaluation, our primitive implementation resulted in a uniform load distribution.
- Additionally, our proposed model was able to detect malicious DDoS traffic successfully. All the valid requests were successfully redirected by the LB to available proxy servers.
- In case of attack traffic, the user requests with unique valid tags were successfully authenticated and processed by proxy servers, whereas the attack traffic without valid tags were dropped.

Future Work

The encryption process in our proposed solution might add a little bit of overhead We intend to study its effects on performance in terms of

(a) time take to direct an incoming client to one of the proxy servers,

(b) time taken to process the client’s HTTP request, and

(c) false positives and negatives during DDoS detection

Our future work will focus on extending our initial proposed solution to improve the above identified performance factors

Conclusion

We propose an authentication mechanism that ensures each client request arriving at a proxy server is directed by the Load Balancer.

A proxy server will only serve those clients that are originally redirected by the Load Balancer. Since the Load Balancer's job is to uniformly distribute the incoming client traffic among existing proxy servers, the chances of a DDoS attack due to huge amount of incoming traffic is mitigated. Thus, the DDoS attacks caused due to botnets can be easily handled.

Reference

[1] S. Venkatesan, M. Albanese, K. Amin, S. Jajodia, and M. Wright, "A Moving Target Defense Approach to Mitigate DDoS Attacks against Proxy-Based Architectures," in Proceedings of the Communications and Network Security (CNS), 2016, pp. 198-206.

[2] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, "Catch me if you can: A cloud-enabled DDoS defense," in Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2014), Atlanta, GA, USA, June 2014, pp. 264 – 275.

[3] H. Wang, Q. Jia, D. Fleck, W. Powell, F. Li, and A. Stavrou, "A Moving Target DDoS Defense Mechanism," Computer Communications, vol. 46, pp. 10 – 21, June 2014.

[4] "DDoS Attack Types & Mitigation Methods: Imperva." *Learning Center*, Imperva, 7 July 2020, www.imperva.com/learn/ddos/ddos-attacks/.

[5] A. Stavrou, A. D. Keromytis, J. Nieh, V. Misra, and D. Rubenstein, "MOVE: An end-to-end solution to network denial of service," in Proceedings of the Network and Distributed System Security Symposium (NDSS 2005), San Diego, CA, USA, February 2005, pp. 81–96.