Catch Me If Can: A Cloud-Enabled DDoS Defense

Quan Jia, Huangxin Wang, Dan Fleck, Fei Li,
Angelos Stavrou, Walter Powell

Presented by
Surya Mani

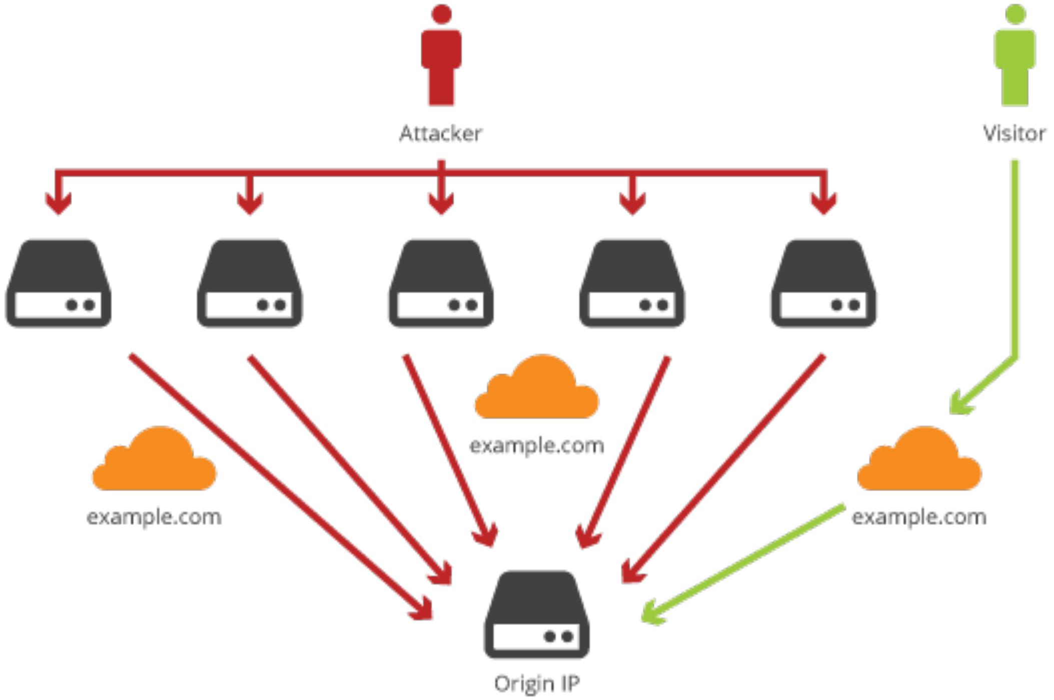
Content

- Motivation
- Related Work
- Cloud-enabled DDoS Defense
- Shuffling Based Segregation
- Experimental Evaluation

Motivation

- DDoS attacks is severest security threat to Internet Security
- Drawbacks in Present Defense Schemes

What is DoS and DDoS?



Related Work

- Filtering-based Approach And Capability Oriented Mechanism
- Overlay-based Defense
- Moving Target Defense
- Fast Flux Technique
- MOVE – Migration OVErlay
- MOTAG – Moving Target defense

Cloud- Enabled DDoS Defense

- Improvement over MOTAG system
  - Securing Internet services that support both authenticated and anonymous users against network and computational DDoS attacks
- Selective Server Replication
  - By replicating the server, the attacked server is taken offline and recycled
- Intelligent Client Reassignment
  - Shuffling: intelligently assigns client to the new replica server

System and Threat Model

- Network DDoS attacks
- Computational DDoS attacks
- Attacks performed by Attacker-Controlled Botnets
  - Naïve bots
  - Persistent bots
- DDoS detection- uses indicators or advanced traffic analysis technique
- Cloud-Enabled DDoS Defense is deployed

System Architecture and Components

Key Components

- Load Balancer
- Replica Servers
- Coordination Server

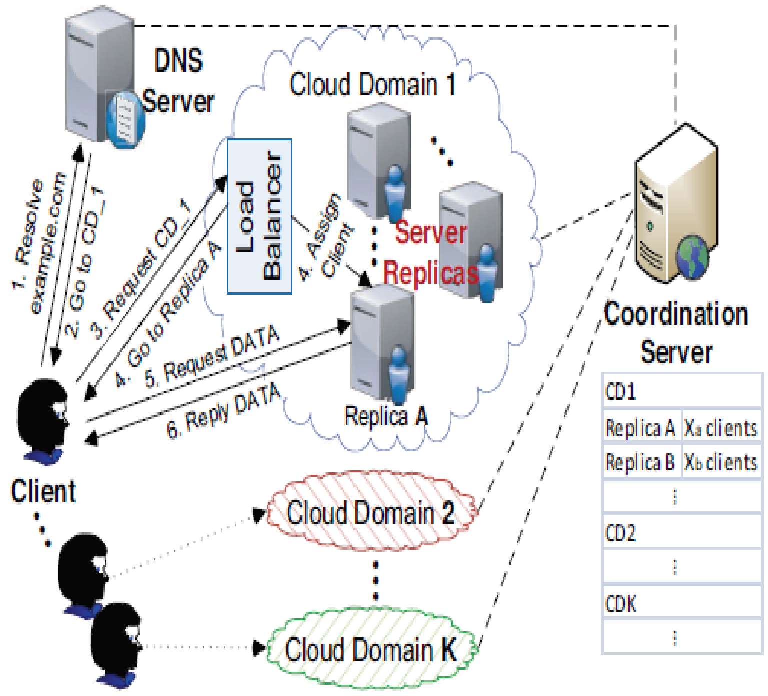


Figure 1. Architecture and Components

1. Load Balancer

- Client redirection
- Client-to-server assignment using Load balancing algorithm
- Keeps track of active replica servers
- Like Round-Robin DNS load balancing

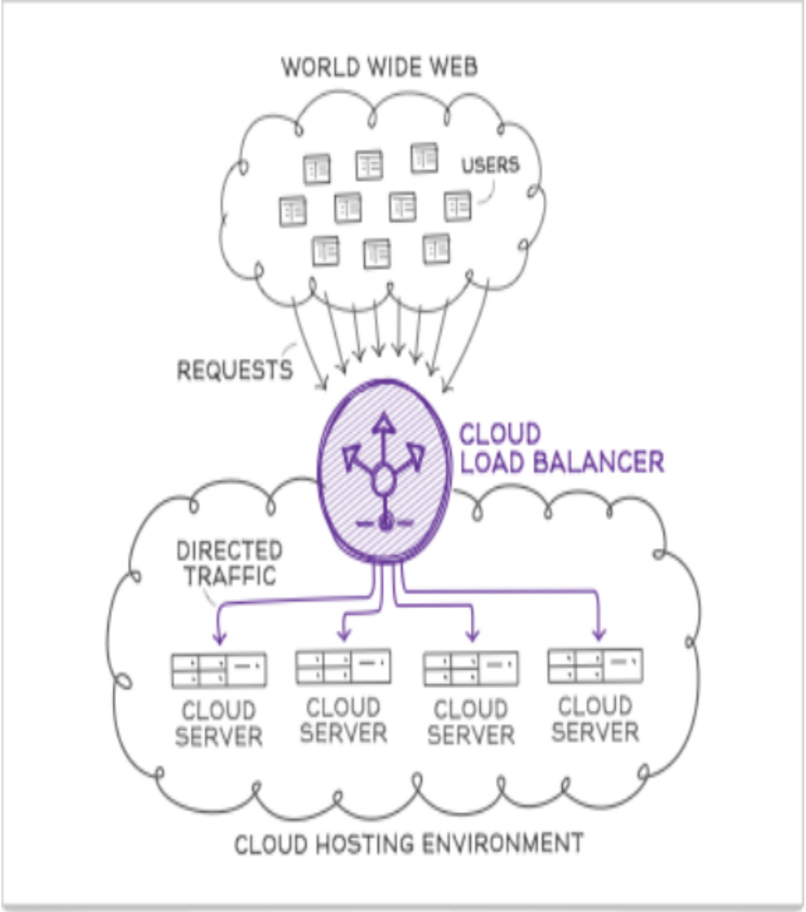


Image source: https://www.rackspace.co.uk/

2. Replica Server

- Replicate the protected servers
- Enforce Whitelist-based filtering
- When bombarded by DDoS attack, client-to-server shuffling takes place
- Attacked replica server is recycled
- Shuffling and non-shuffling replicas

3. Coordination Server

- Directs real-time actions against DDoS attacks
- Keep tracks of client-to-server assignment
- Respond to DDoS attack by computing optimal shuffling plan
  - Decides the number of clients to be reassigned to new replica server
- Communicates via a dedicated command and control channel

Shuffling Based Segregation - Structured method

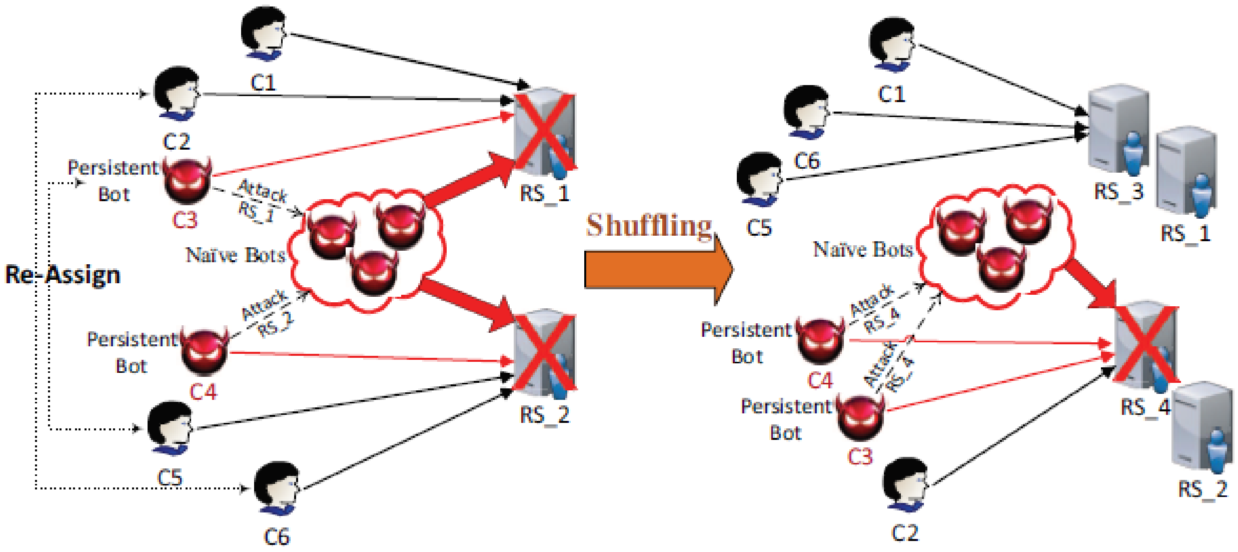


Figure 2. An example of client-to-server shuffling

Shuffling Based Segregation – Cont.

- Coordination server’s decision for reassignment of clients to new replica server is by using
  - Dynamic Programming algorithm
  - Greedy choice algorithm

Notations

Table I
NOTATIONS USED IN THIS PAPER AND THEIR MEANINGS

Notation	Meaning
N	number of clients (including benign clients and bots)
M	number of persistent bots
P	number of shuffling replicas
S	number of clients to be saved
p_i	probability that the i-th shuffling replica is not under attack
x_i	number of clients assigned to the i-th shuffling replica

Theoretical problem modeling

- Shuffling is determined randomly so we use probabilistic analysis
- E(S) – expected number of benign clients to be saved in one round

$$\max E(S) = \sum_{i=1}^{P} p_i \cdot x_i = \frac{\sum_{i=1}^{P} \binom{N-x_i}{M} x_i}{\binom{N}{M}}$$

$$\text{subject to} \sum_{j=1}^{P} x_j = N \qquad (1)$$

$$p_i = \frac{\binom{N-x_i}{M}}{\binom{N}{M}}.$$

Optimal Solution

- Solve max {S(a,b,1)+S(N-a,M-b,P-1)}

- Dynamic programming approach(bottom-up)

$$S(a,b,1) = \begin{cases} a, & b = 0 \\ 0, & b > 0 \end{cases} \tag{2}$$

$$S(N,M,P) = \max_{1 \le a \le n-1} \{ \sum_b \Pr(b) \times [S(a,b,1) + S(N-a, M-b, P-1)] \}$$

where

$$\Pr(b) = \frac{\binom{M}{b}\binom{N-M}{a-b}}{\binom{N}{a}}, b \in [0, \min(a, M)] \tag{3}$$

Algorithm

- Runtime – O(N^3.M^2.P) Space – O(N.M.P)

Algorithm 1 Optimal-Assign(N, M, P)

Initialize $save\_no[0, \cdots, N, 0 \cdots, M, 0 \cdots, P]$ and $assign\_no[0 \cdots, N, 0 \cdots, M, 0 \cdots P]$

for $i \leftarrow 1, N$ **do**

    for $j \leftarrow 1, M$ **do**

        for $k \leftarrow 1, P$ **do**

            compute $S(i, j, k)$ using Equations 2 and 3, with $a \in [1, i-1]$ and $b \in [1, \min\{j, a\}]$;

            select $a = \alpha$ that maximize $S(i, j, k)$;

            update table entry $assign\_no[i, j, k] = \alpha$ and $save\_no[i, j, k] = S(i, j, k)$.

Greedy Algorithm (Top-down approach)

- Dynamic programming algorithm is inadequate for making real-time decisions
- Greedy performs runtime shuffling decisions one replica server at a time
- Makes a greedy choice by selecting one locally optimal solution and then solving the remaining sub problem
- Runtime- O(N.M)
- Space – O(P)

Algorithm evaluation

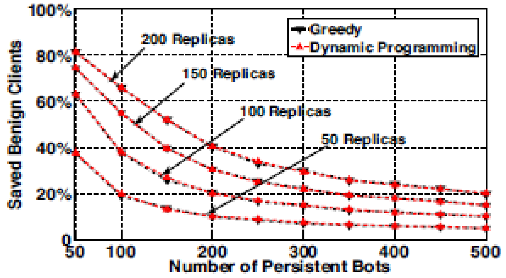


Figure 3. Compare the effectiveness of greedy algorithm and dynamic programming algorithm for one shuffle with 1000 clients. (*Curves are overlapping.*)

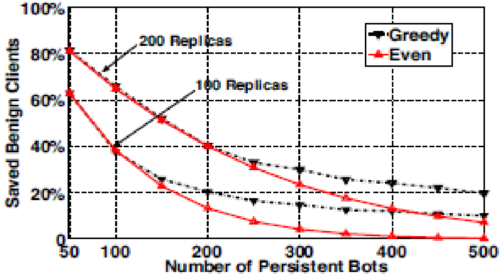


Figure 4. Compare the effectiveness of greedy algorithm and even distribution for one shuffle with 1000 clients.

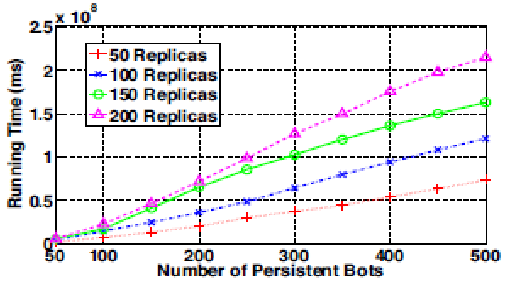


Figure 5. Running time of the dynamic programming algorithm with 1000 clients.

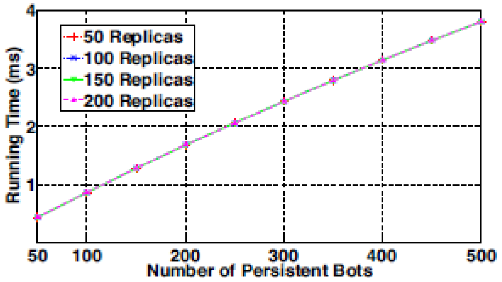


Figure 6. Running time of the greedy algorithm with 1000 clients.

Maximum Likelihood Estimation(MLE) Algorithm

- Used to estimate the probability of M(Persistent bots) going to attack X servers. I.e. X<=M<=N

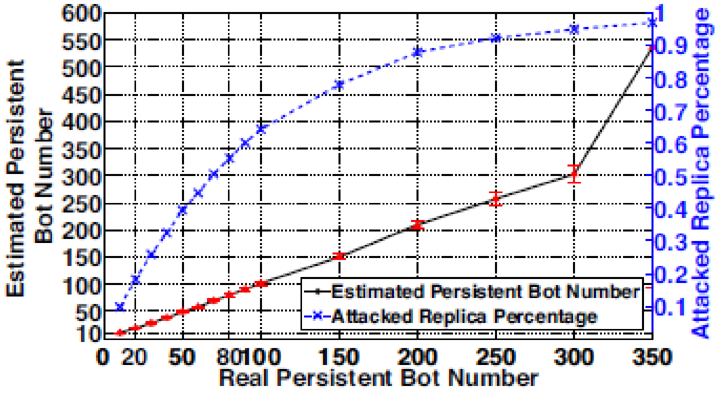


Figure 7. Evaluate MLE algorithm through examples (10000 clients, 100 shuffling replica servers)

Experimental Evaluation

- Prototype-Based evaluation

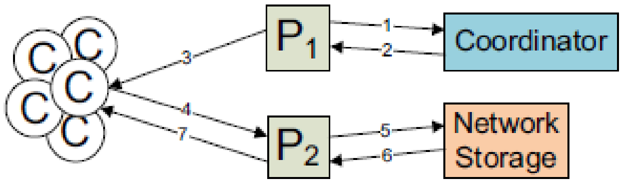


Figure 11. System prototype (C – Client, P – Replica Server)

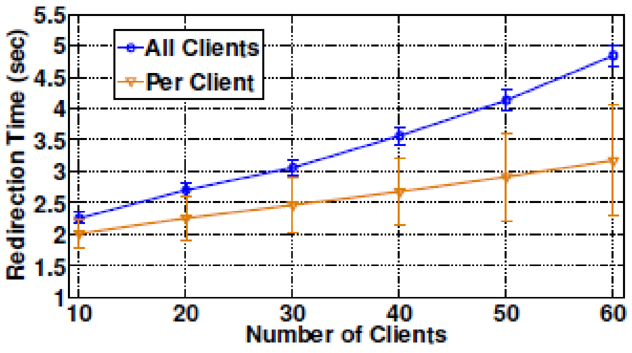


Figure 12. Client migration time between two replica servers

- Simulation-Based Evaluation

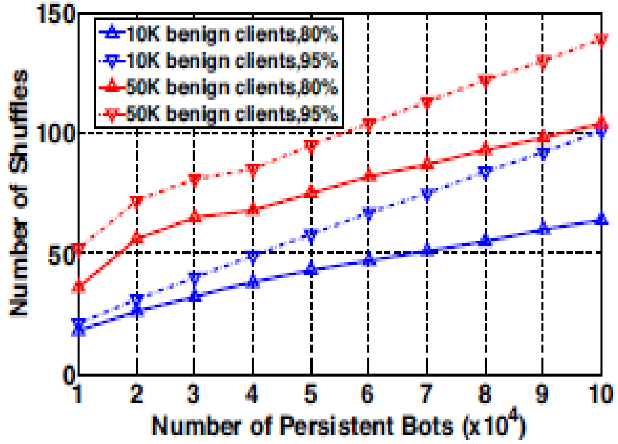


Figure 8. Number of shuffles to save 80% and 95% of 10^4 and 5×10^4 benign clients, with 1000 shuffling replica servers, and varying persistent bot numbers.

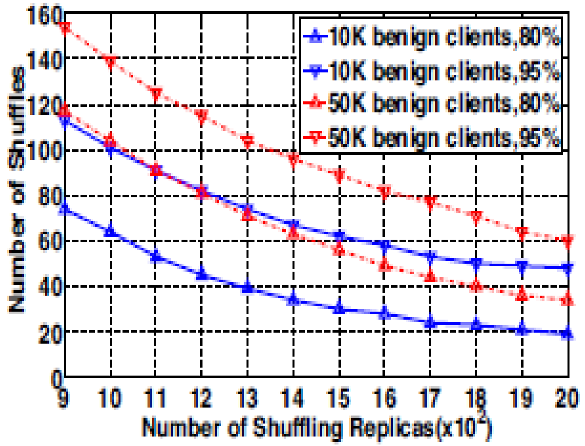


Figure 9. Number of shuffles to save 80% and 95% of 10^4, and 5×10^4 benign clients, with 10^5 persistent bots and varying shuffling replica server numbers.

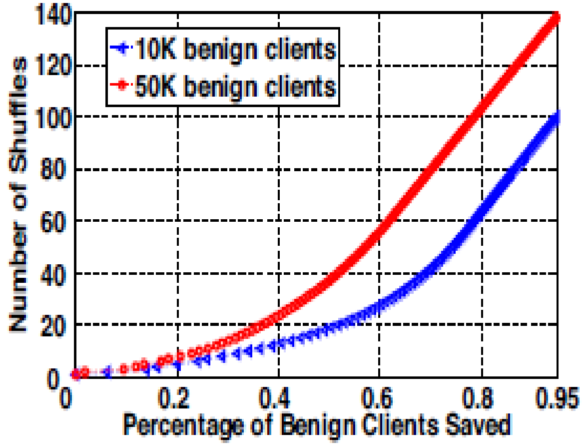


Figure 10. Cumulative percentage of saved benign clients vs. number of shuffles, with 10^5 persistent bots, 10^4, and 5×10^4 benign clients.

THANK YOU