Catch me if you can

A cloud based DdoS defense

Mikaël Fourrier

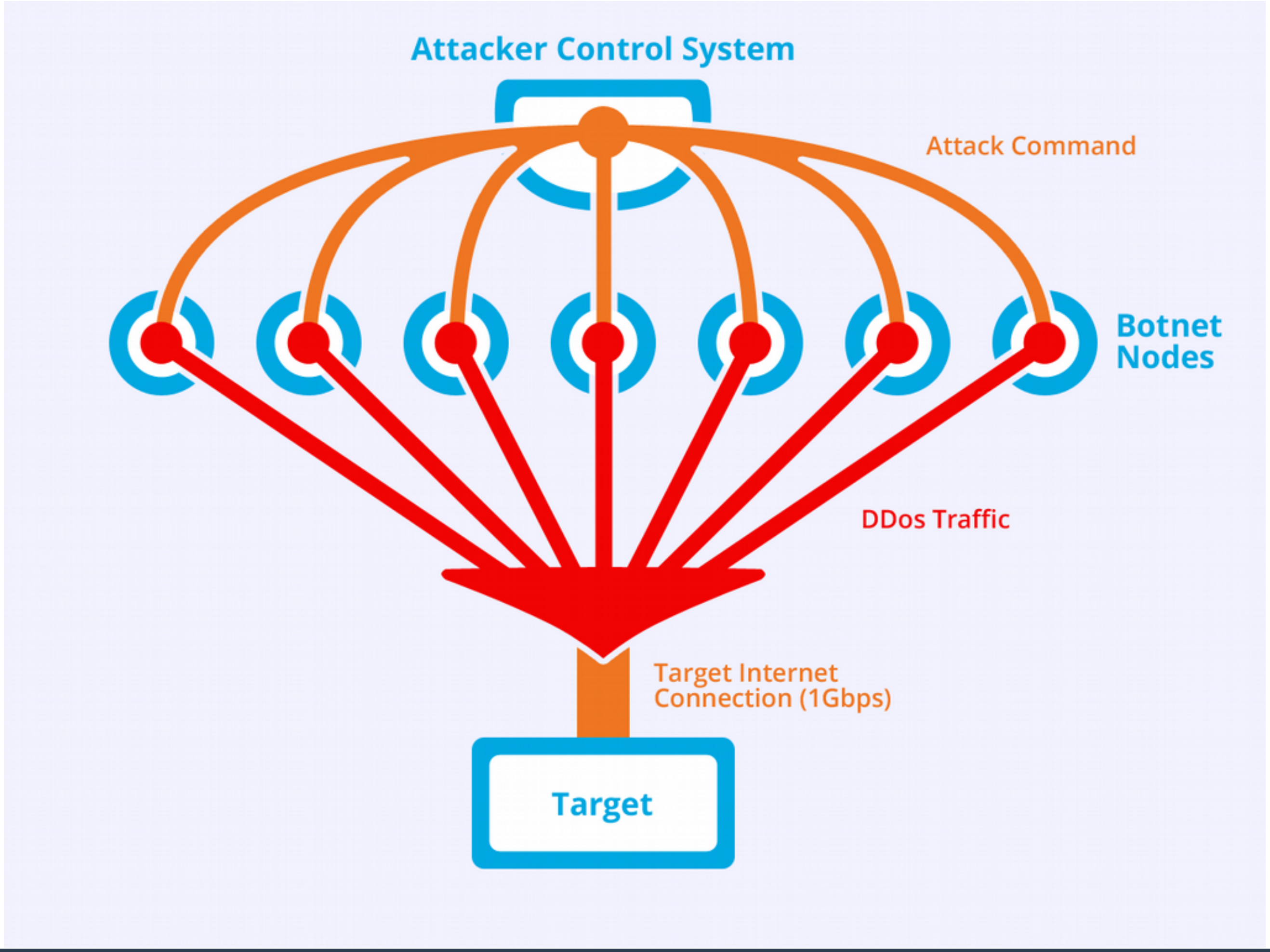
DDoS attacks

- **Goal: prevent the access to a computer system to it's legitimate users.**
- **DDoS: DoS attack on the network using a lot of different source IP**
- **Why:**
  - Reprisal (ex: Anonymous)
  - Cyber-war or cyber-terrorism
  - Extorsion

    ...

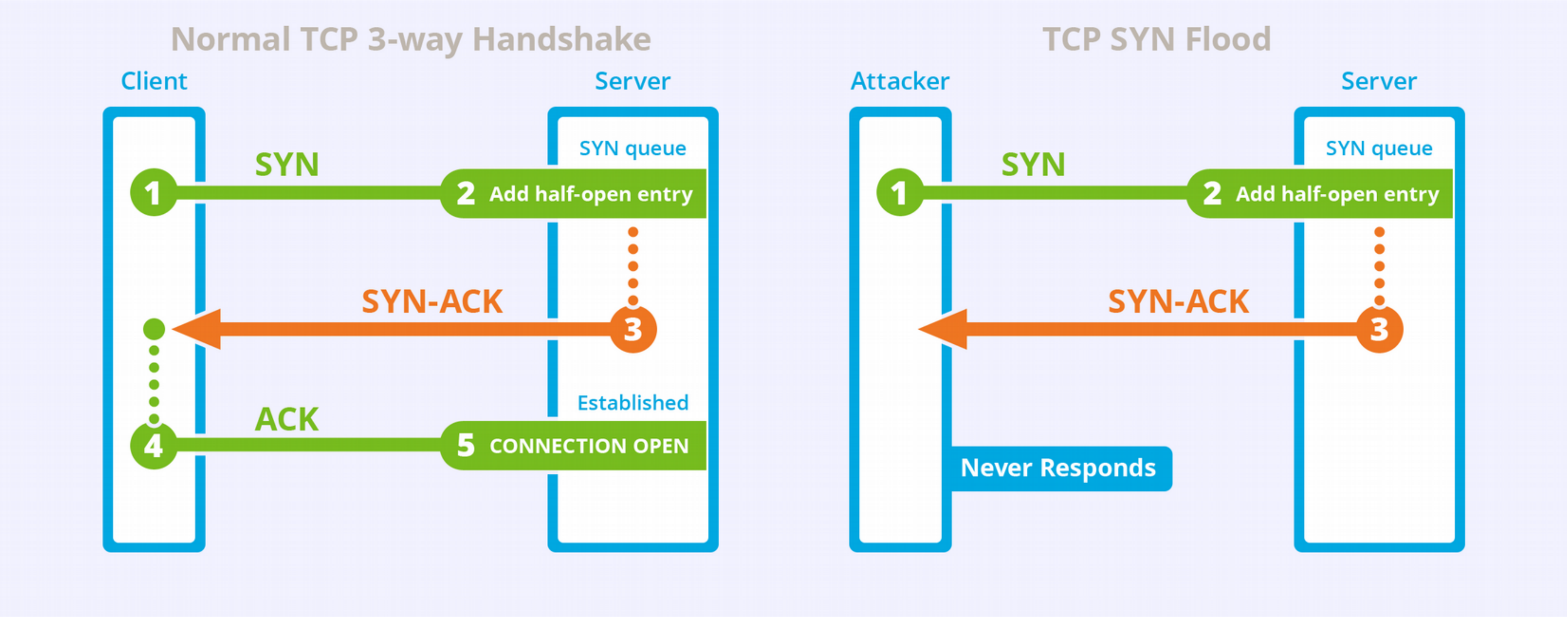
Examples of DDoS attacks

- **Volumetric attack**



Examples of DDoS attacks

- **SYN flood (layer 4)**



Examples of DDoS attacks

- **Application level (layer 7)**
  - Download large file
  - Make heavy database request
  - Hit CPU-intensive URL
  - Upload large file

Mitigations

- **Have more servers and bandwidth**
  - but useless the rest of the time
  - costs a lot
- **Firewall with an IP blacklist**
  - good for individual attacks, not so good against DDOS and dynamic IPs
- **More secure code**
  - only works against specifics attacks like file upload

Need a way to dynamically
add and remove new servers,
only when needed...

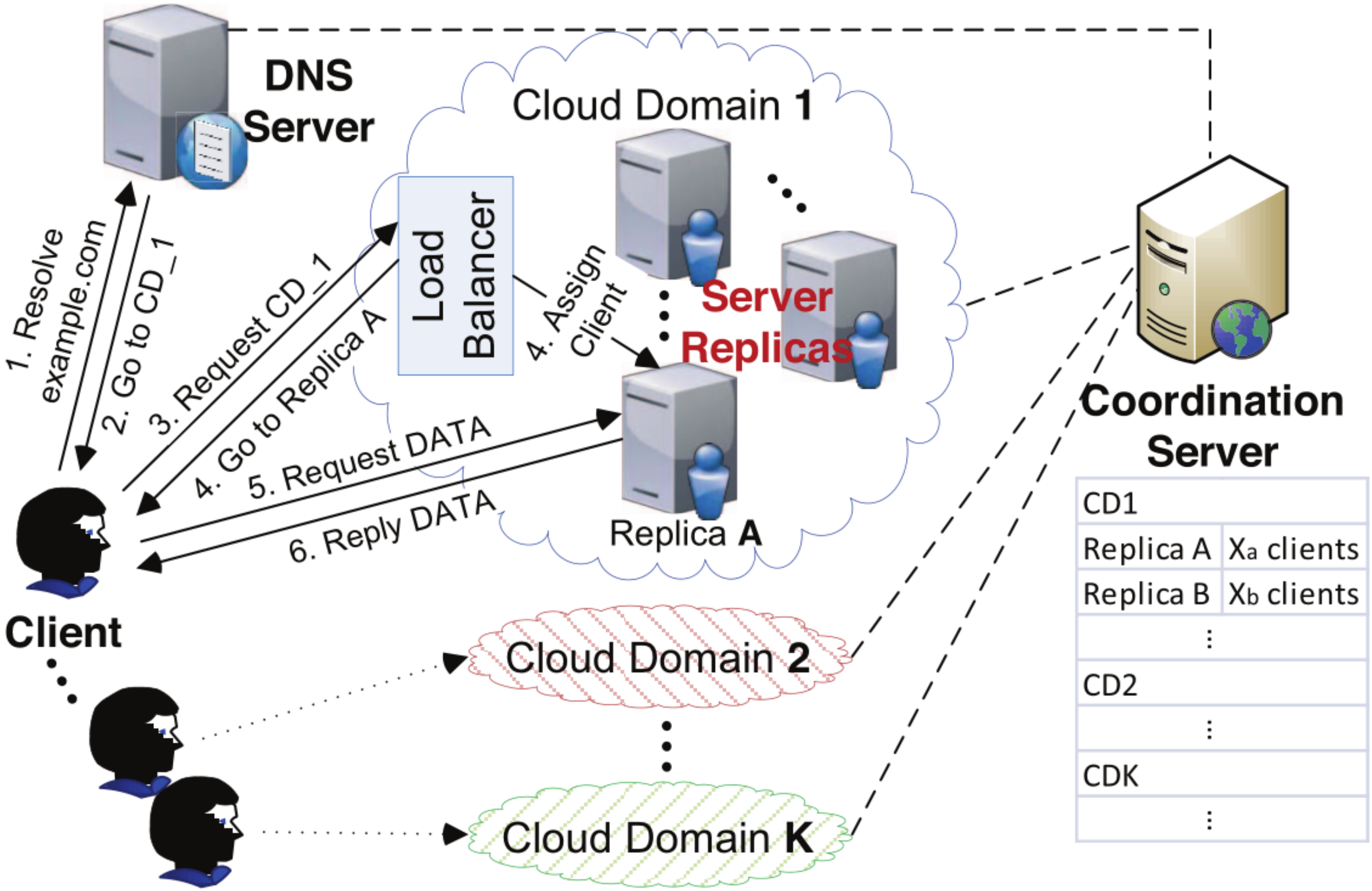
Enters the “cloud computing”

- **Exemple: Amazon EC2**
  - VM based
  - Auto-scaling
  - Quick start of new instances
  - Pay what you use
  - Very high total bandwidth and computing power

Catch me if you can

- “A cloud-enabled defense mechanism for Internet services against network and computational DDoS attacks”
- Uses a “shuffling” mechanism to segregate attackers and legitimate users
- Add and remove servers to present a moving target

Architecture



CD1	
Replica A	X_a clients
Replica B	X_b clients
⋮	
CD2	
⋮	
CDK	
⋮	

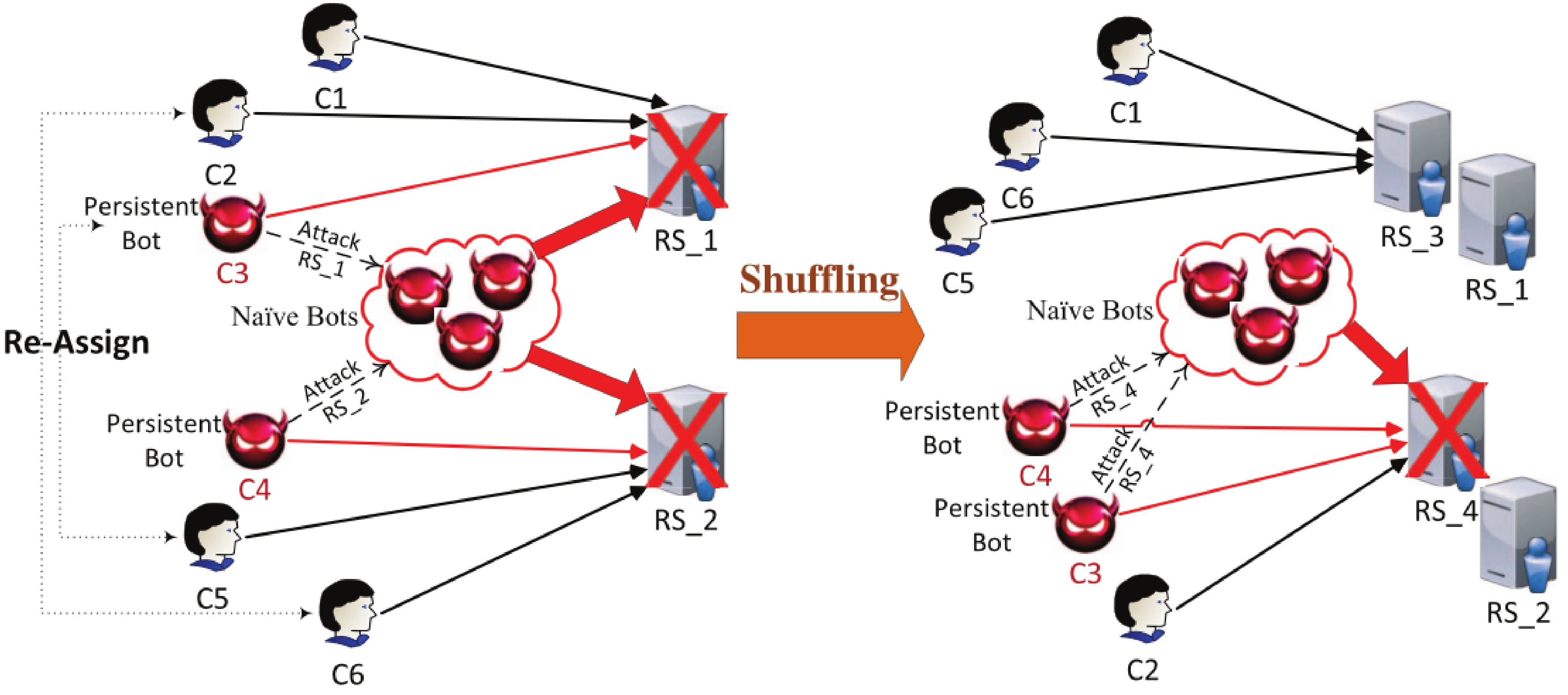
Points of failure

- **DNS servers**
  - still a problem
- **Load balancers**
  - not a problem with Amazon
- **Replica**
  - not a problem with auto-scaling
- **Coordination server**
  - not accessible from the internet so not a problem

Interlude: Persistent vs naïve bot

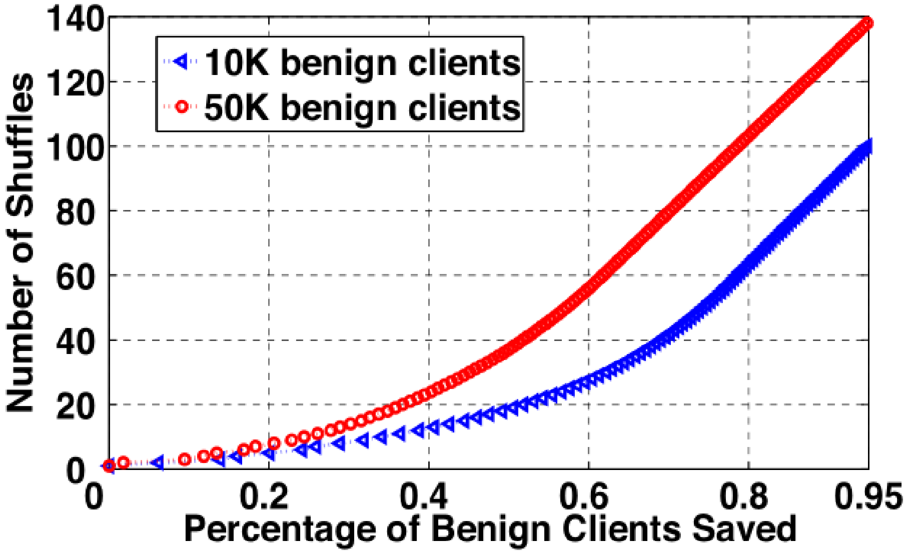
- **Naïve bot: dumb**
  - can only attack one IP
- **Persistent bot: adaptative**
  - can follow a target
  - understand HTTP redirect

Clients segregation

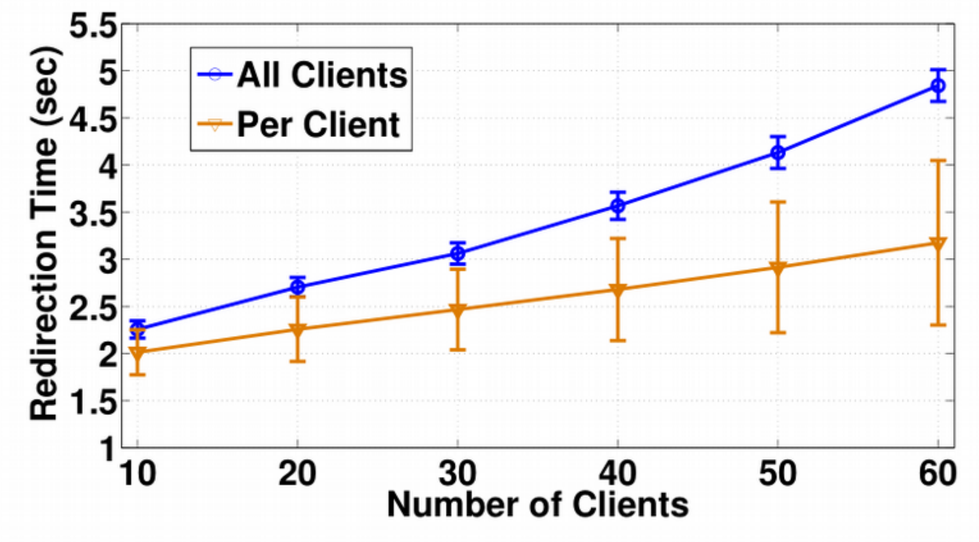


- RS_3 is bot free!
- Some naïve bots can still attack RS_1 and RS_2, so we shut them down
- Use WebSocket to redirect connected clients
- Use of an dynamic programing or greedy algorithm to make the best segregation possible to maximize the number of saved clients

Results



10K benign clients
50K benign clients
Number of Shuffles
0
20
40
60
80
100
120
140
0
0.2
0.4
0.6
0.8
0.95
Percentage of Benign Clients Saved



All Clients
Per Client
Redirection Time (sec)
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
10
20
30
40
50
60
Number of Clients

Discussion

- Dependent on Amazon infrastructure
- Worst case scenario used
- Catch non-aggressive attackers (stateless algorithm)
- Can catch re-entrant bots
- Cost effective and scalable
- Doesn't require an application modification

Credits

- Original paper: http://cs.gmu.edu/~astavrou/research/Catch_me_if_you_can_DSN14.pdf
- DdoSBootcamp (images): https://www.ddosbootcamp.com/