Reintroducing Client Puzzles for DDoS Mitigation

Introducing DDoS:
The Volumetric Distributed Denial of Service Attacks (DDoS) is one of the most common problems in network security. Volumetric DDoS Attacks occur when a server is flooded with so much fake traffic that the server cannot serve requests from legitimate clients.

Current Solutions:

- **PUZZLES**
  - Puzzles are cryptographical challenges imposed on clients trying to access a webserver. Clients can solve puzzles by hashing through a nonce $2^{d+1}$ times.
  - Therefore, clients will have to provide more computational work for a larger $d$.

Strengths:
- All devices must use computational resources to communicate with a server
- Stateless
- Makes attacks more expensive to run (60% more CPU utilization)
- Computer handles the work, not the user

Weakness:
- Fairness issue between devices with strong and weak computational resources
- Computation must happen at kernel level

Ours Argument

Client puzzles with scalable difficulty deployed on a flexible network can mitigate the affects of a modern DDoS attack.

Where $R$ is the number of resources the server would need to use to fulfill a client’s request, we can scale the difficulty ($d$) like so

$$R \uparrow \text{ than } d \uparrow \text{ and } if \ R \downarrow \text{ than } d \downarrow$$

Implementing this client puzzle protocol system is even easier today with a cloud architecture using software like intel’s DPDK that runs applications at kernel level.

The problem
- DDoS attacks are getting cheaper and cheaper to run
- Current solutions would fail if an attack exceeded the current bandwidth available by mitigation providers.
- The only way to prevent DDoS attacks from impacting victims is to have more resources than attackers can get ahold of

A Solution
- Make attacks more expensive to run by exhausting the attackers’ resources
- Improve filtering by collecting more information on the source of the traffic

Andrew Walkowski
Undergraduate Computer Science Major

Theodore Yin
Undergraduate Computer Science Major

Muhammad Noureddine
Assistant Professor of Computer Science and Software Engineering