Gatekeeper

The First Open Source DDoS Protection System

hostnet

Michel Machado

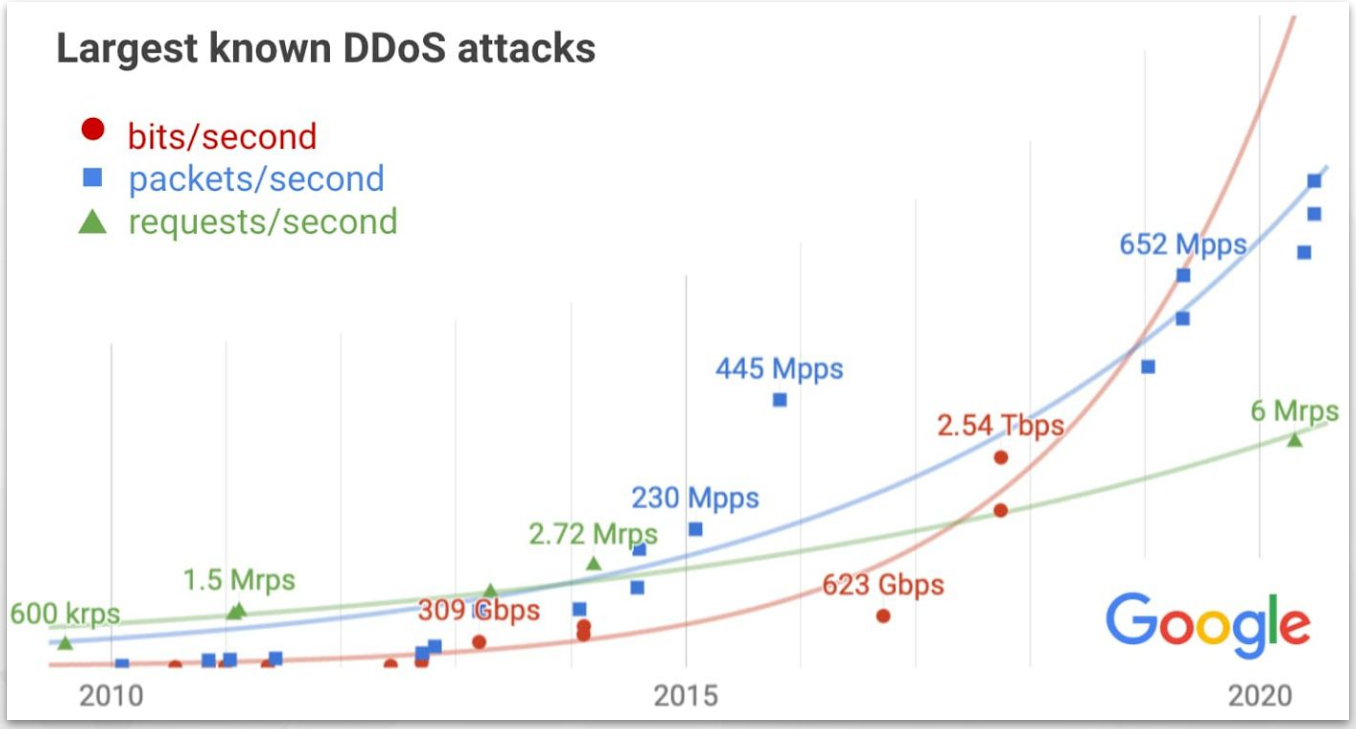
BOSTON UNIVERSITY

Cody Doucette*
Qiaobin Fu**
John W. Byers

* Current affiliation: Cloudflare
** Current affiliation: Google

APRICOT 2022
March 3rd, 2022

Motivation -- Relevance of DDoS attacks



Gatekeeper

Motivation -- Why Gatekeeper?

Unparalleled multi-vector protection

⇒ All flows are monitored and all filters are active;
alternative solutions have limited filtering capacity;
See paper "The Catch-22 Attack" for details

Scalable and deployed

⇒ Production deployments ranging from 10 Gbps to 1 Tbps

Mitigation in seconds

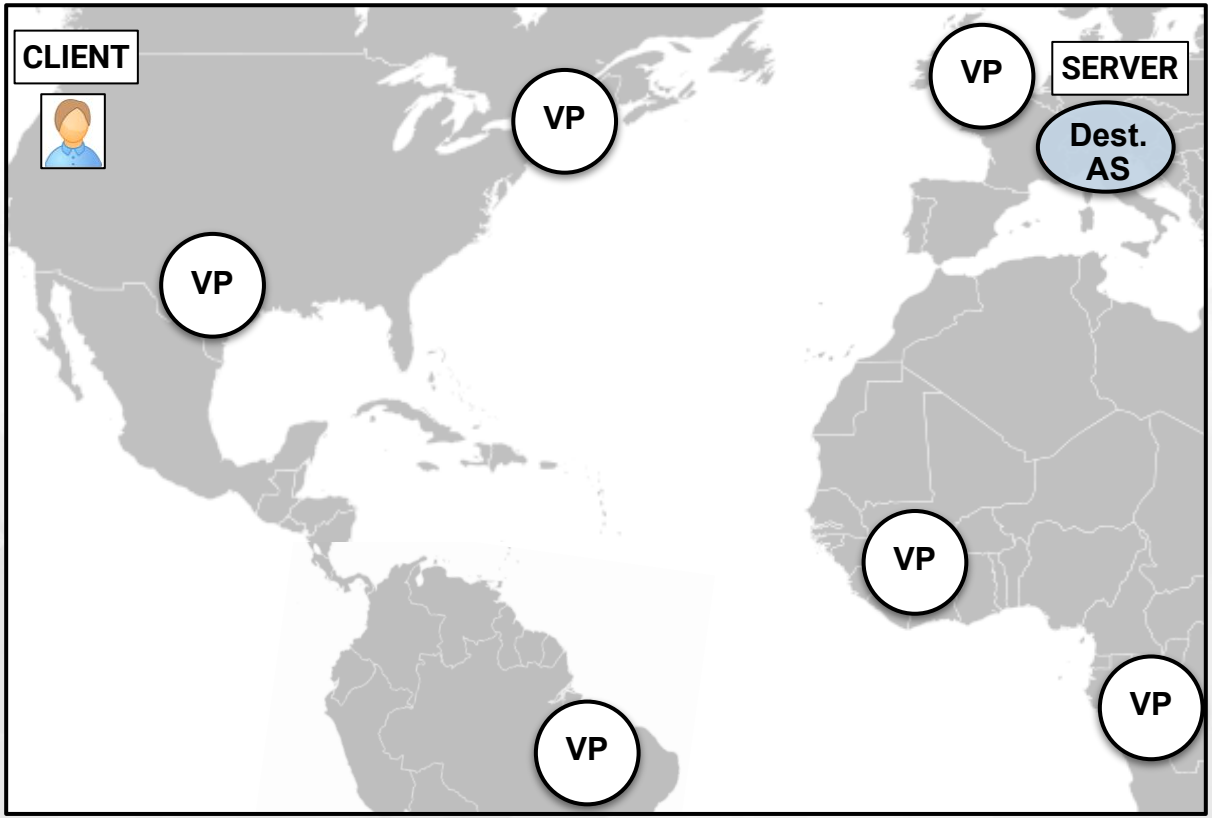
⇒ More than 80% of attacks last ≤ 4 min according to Kaspersky;
There is not much time for human intervention

Gatekeeper

Outline

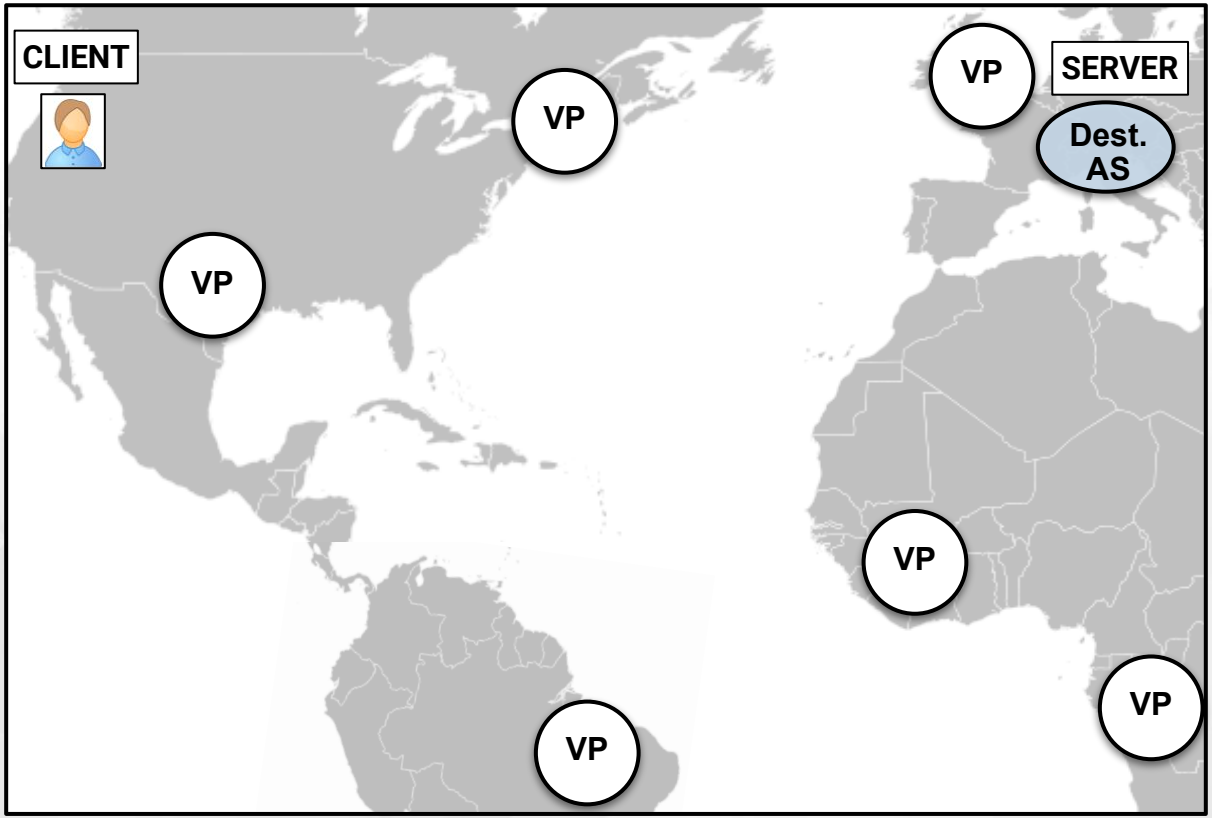
- ✓ Motivation
- How Gatekeeper works
- How to write a destination policy
- Conclusion

Gatekeeper's components



Vantage points: well-provisioned and geographically distributed locations

Gatekeeper's components

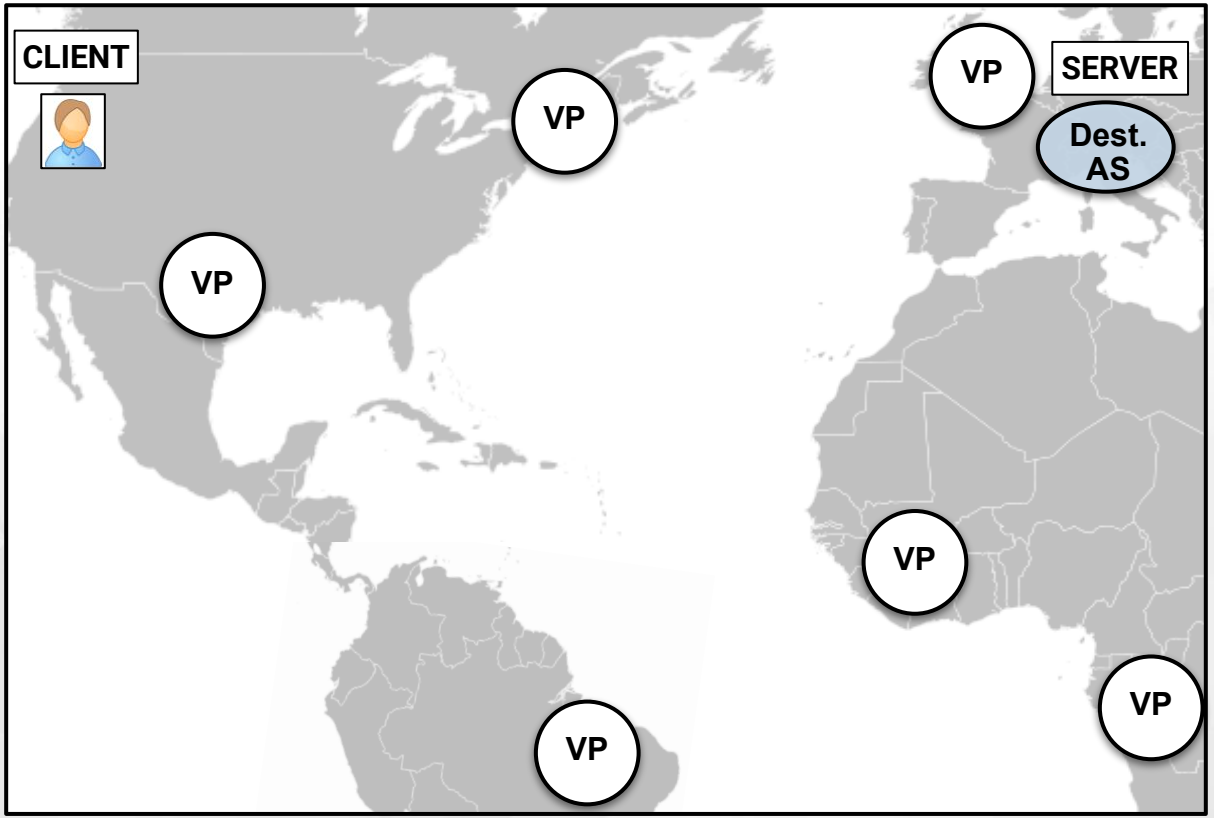


Vantage points: well-provisioned and geographically distributed locations

Requirements:

- computing capacity
- cheap ingress bandwidth
- BGP peering
- private links to the protected AS

Gatekeeper's components



Vantage points: well-provisioned and geographically distributed locations

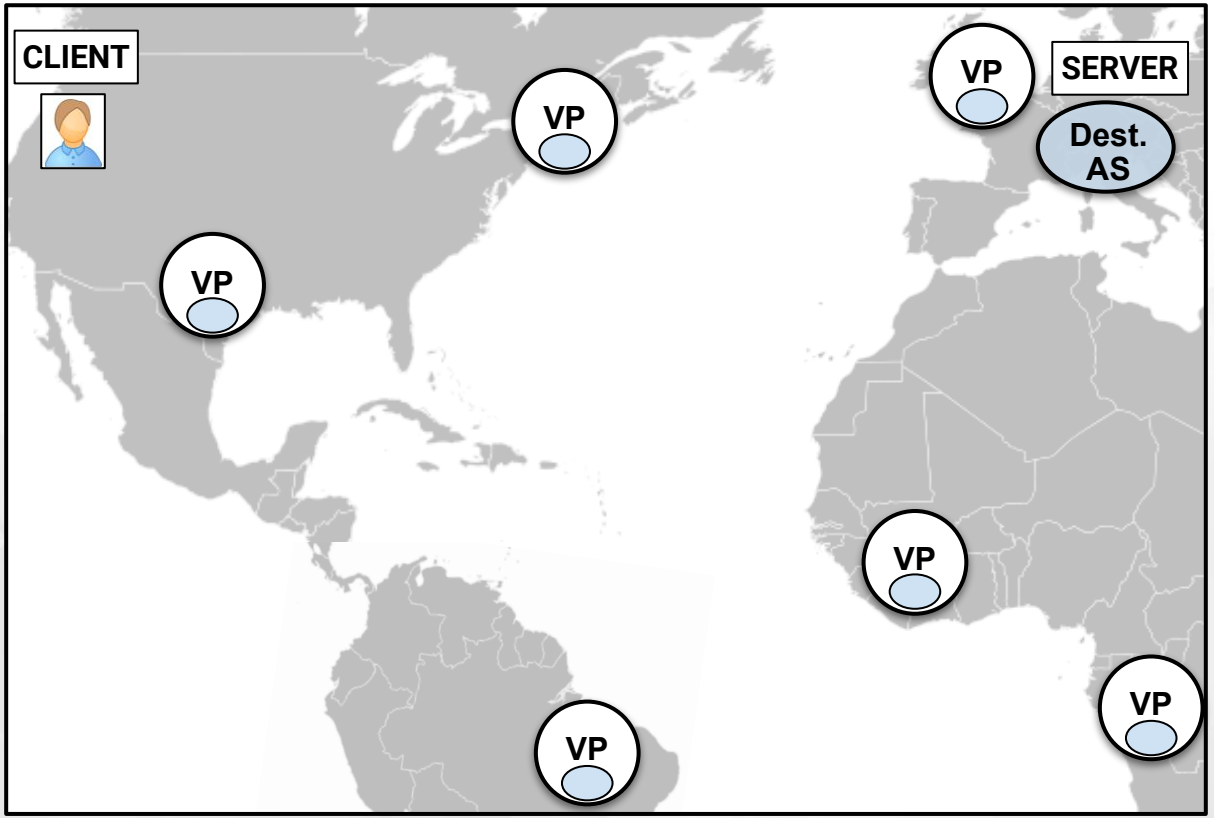
Requirements:

- computing capacity
- cheap ingress bandwidth
- BGP peering
- private links to the protected AS

Examples:

- Internet exchanges
- Peering link
- Some cloud providers

Gatekeeper's components



Vantage points: well-provisioned and geographically distributed locations

Requirements:

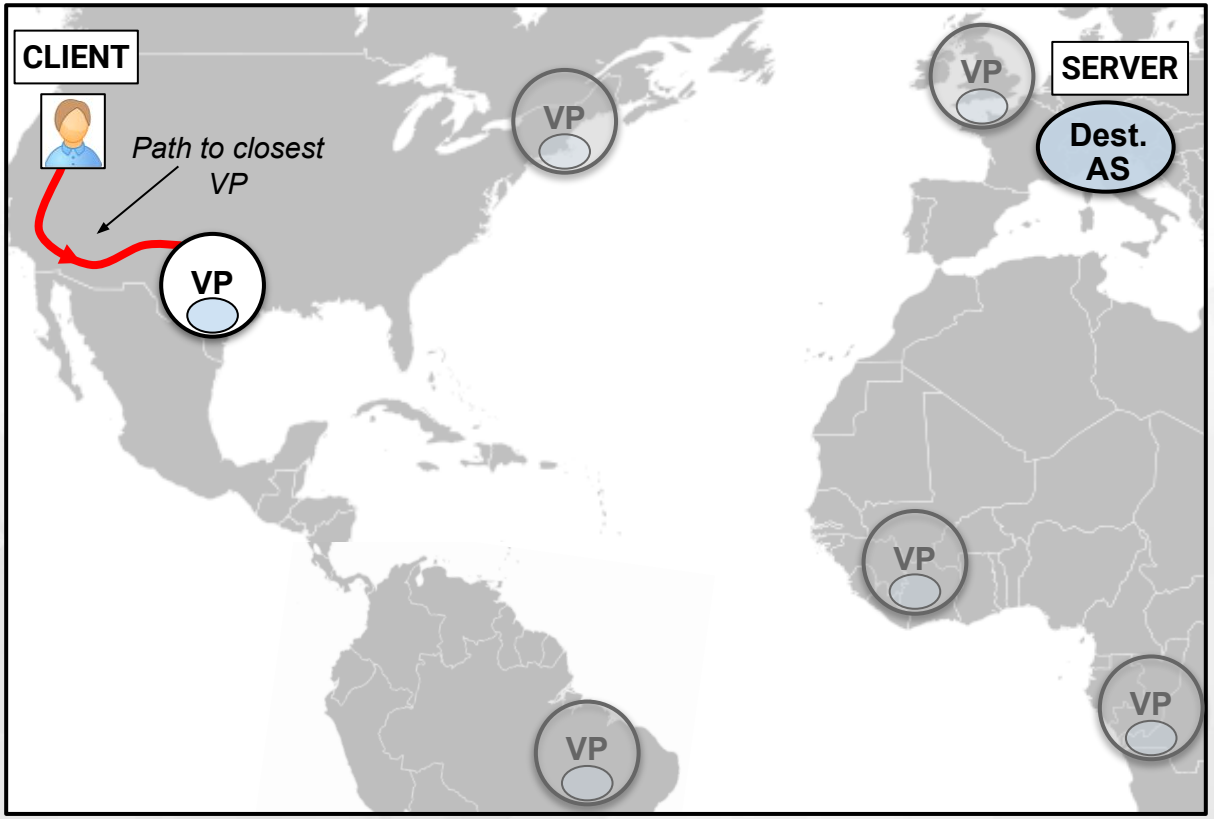
- computing capacity
- cheap ingress bandwidth
- BGP peering
- private links to the protected AS

Examples:

- Internet exchanges
- Peering link
- Some cloud providers

Gatekeeper

Gatekeeper's components



Vantage points: well-provisioned and geographically distributed locations

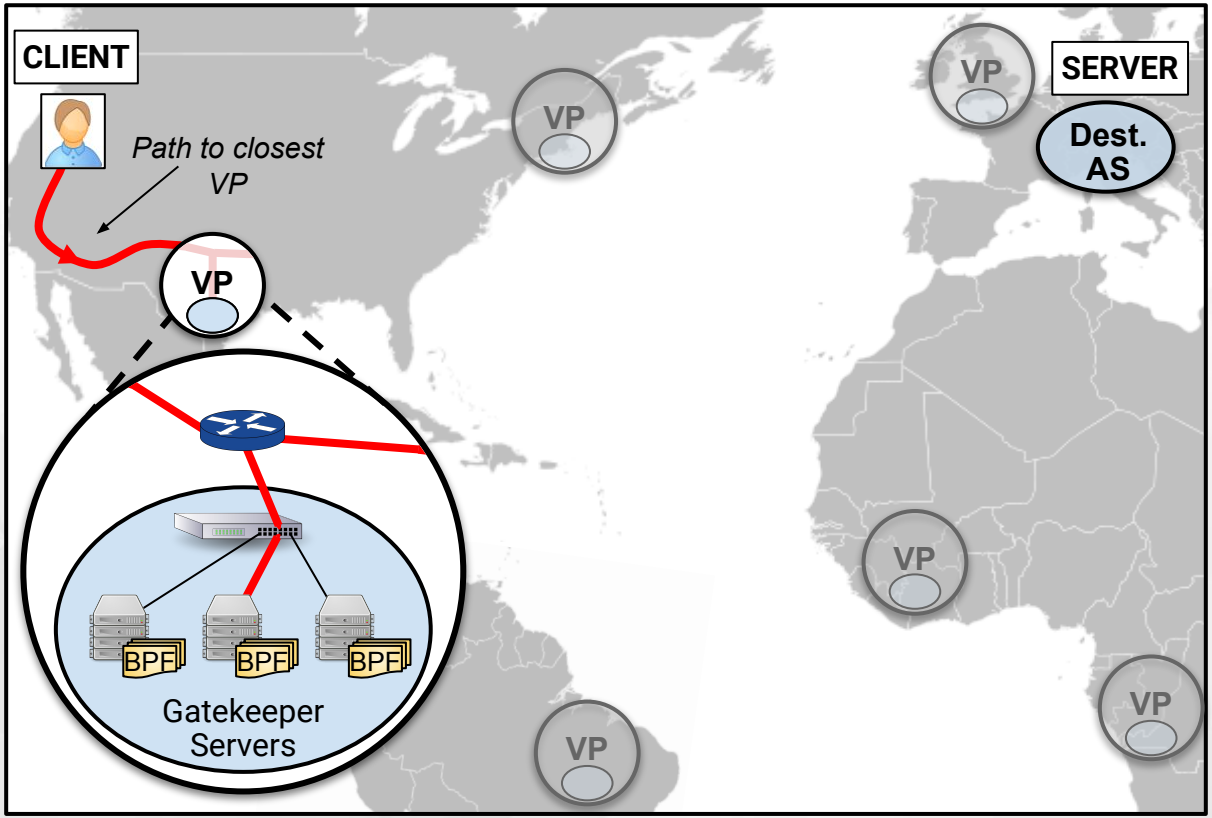
Requirements:

- computing capacity
- cheap ingress bandwidth
- BGP peering
- private links to the protected AS

Examples:

- Internet exchanges
- Peering link
- Some cloud providers

Gatekeeper's components

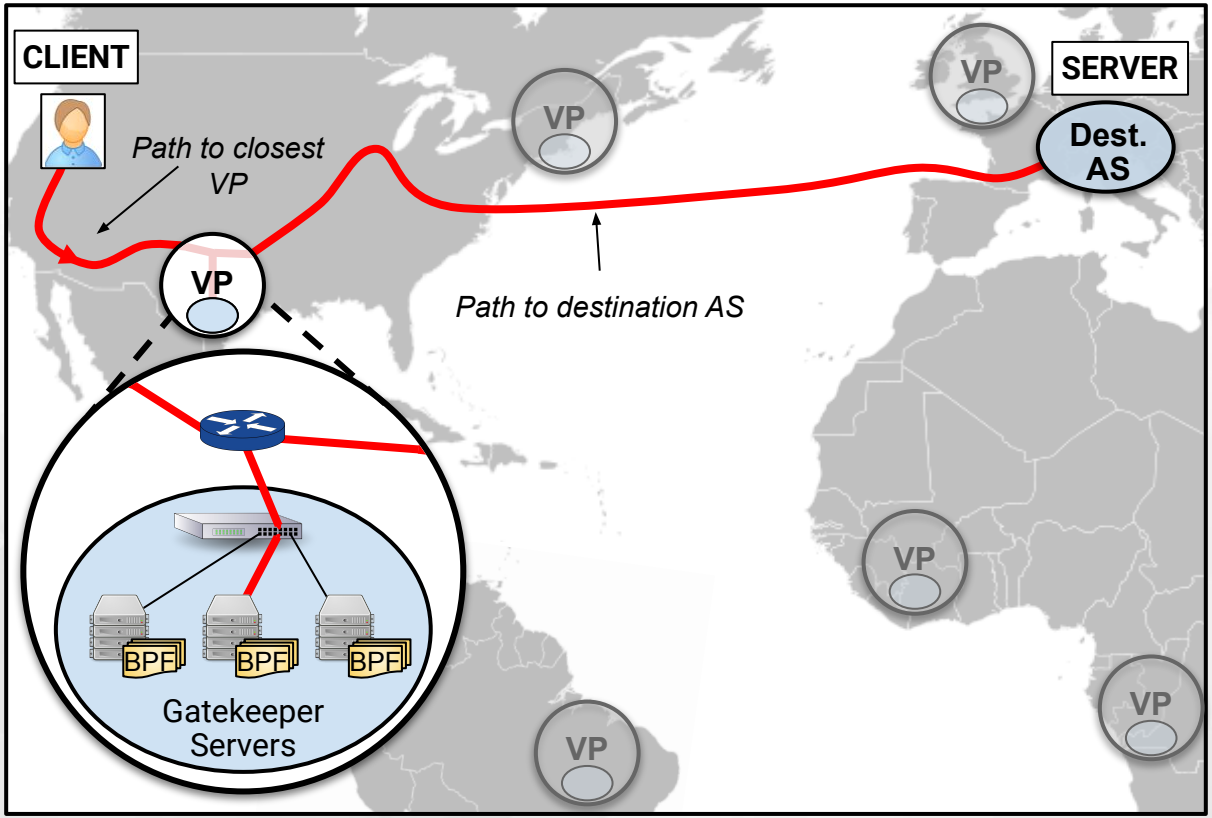


Gatekeeper servers: upstream policy enforcement

Responsibilities:

- Forwarding requests (new flows)
- Dropping or rate-limiting according to per-flow policy enforcement program
- Encapsulating

Gatekeeper's components

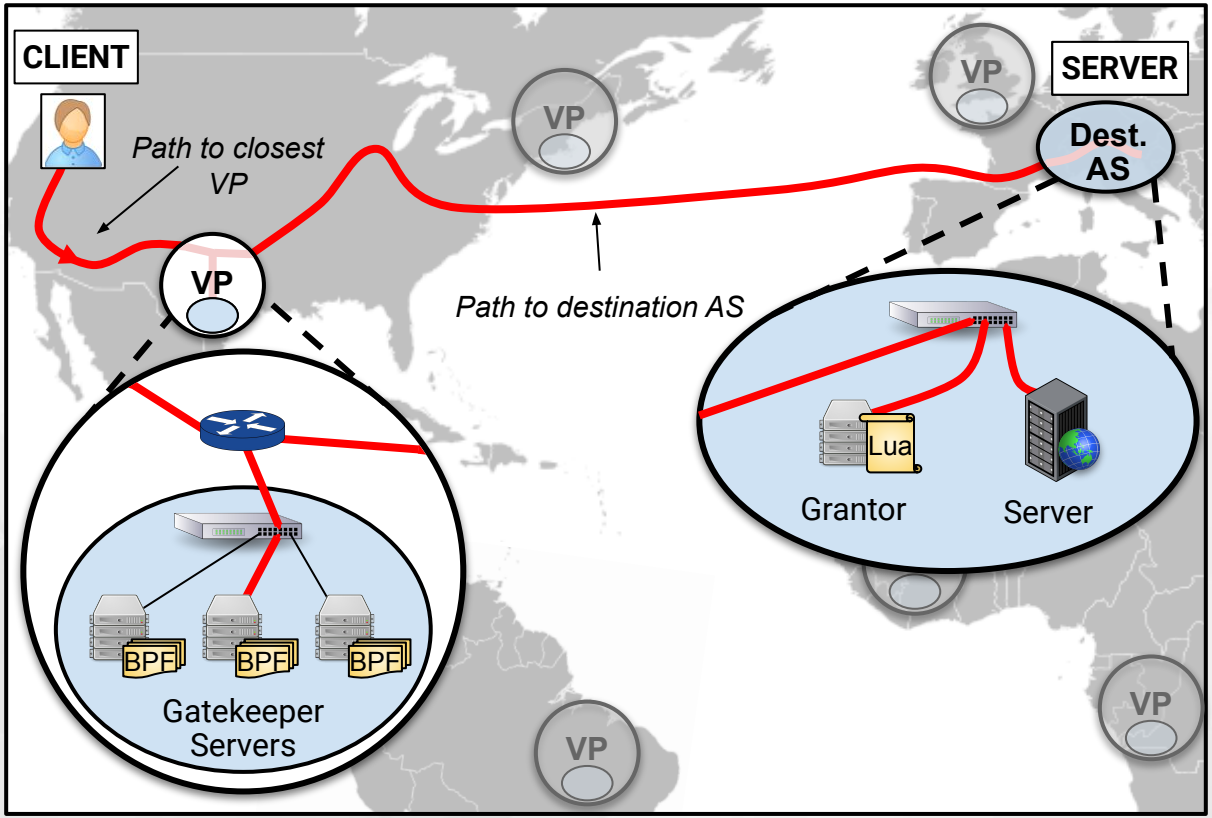


Gatekeeper servers: upstream policy enforcement

Responsibilities:

- Forwarding requests (new flows)
- Dropping or rate-limiting according to per-flow policy enforcement program
- Encapsulating

Gatekeeper's components



Grantor servers: centralized policy decision making

Responsibilities:

- Making policy decisions about requests and installing those decisions at Gatekeeper
- Decapsulating and sending to destination server

Gatekeeper

Quick summary

1. Packets from clients are forwarded to the closest VPs
2. Gatekeeper servers forward packets of new flows to Grantor servers, or run BPF programs to decide what to do
3. Grantor servers run a policy to map flows to BPF programs, and forward granted packets to destinations
4. Grantor servers notify Gatekeeper servers of all policy decisions
5. Gatekeeper servers enforce the police decisions

Gatekeeper

Quick summary

1. Packets from clients are forwarded to the closest VPs
2. Gatekeeper servers forward packets of new flows to Grantor servers, or run BPF programs to decide what to do
3. Grantor servers run a policy to map flows to BPF programs, and forward granted packets to destinations
4. Grantor servers notify Gatekeeper servers of all policy decisions
5. Gatekeeper servers enforce the police decisions

Gatekeeper

Outline

- ✓ Motivation
- ✓ How Gatekeeper works
- How to write a destination policy
- Conclusion

Gatekeeper

Step 1: identify ALL your network profiles

A profile may apply:

to a single server, a group of servers, or

to blocks of IP addresses

Example of a profile: outgoing email servers

- No listening sockets
- Very small ingress traffic footprints

Sources: config files, production servers, docs

Step 1: Network profiles ➔ Step 2: BPF programs ➔ Step 3: Lua Policy

Gatekeeper

Step 2: write an BPF program for each profile

Classify packets into one of these bins:

- **Primary:** main purpose of the service
- **Secondary:** needed packets (e.g. TCP SYN, ICMP)
- **Unwanted:** please guess :-)

Enforce primary bandwidth limit before classification

Enforce secondary bandwidth limit after classification on secondary packets

Step 1: Network profiles ➔ Step 2: BPF programs ➔ Step 3: Lua Policy

Gatekeeper

Step 3: map flows to your BPF programs

Just classify flows using the destination IP address

Example: 10.99.99.128/25 are outgoing email servers

This information is a byproduct of Step 1

Grantor servers run this part of the policy (Lua policy)

Gatekeeper

Step 3: map flows to your BPF programs (bonus)

Classify source IP addresses too!

- Reject bogons, abusers, malware
- Tune bandwidth to partners, countries, end users
- Return different profiles to CDNs, crawlers, offices

Manage all your IP ranges with Drib:
https://github.com/andrenth/drib

Gatekeeper

Example policy against a SYN flood

The request channel immediately limits the SYN flood to 5% of the link capacity

⇒ The request channel is out of the scope of this presentation

The secondary limit further limits the flood to <5%

⇒ Assuming all primary limits are less than the link capacity (typical)

The negative bandwidth blocks abusive flows

⇒ The secondary limit bounds the worst case

Gatekeeper

What kinds of attacks can this policy stop?

Typical infrastructure attacks

⇒ Floods (e.g. SYN, UDP, ICMP)

⇒ Amplifications (e.g. DNS, NTP, Memcached)

Enforcing application patterns

⇒ Cryptocurrencies, VoIP, online games, port knocking

Advanced attacks

⇒ Carpet-bombing, Catch-22, Crossfire

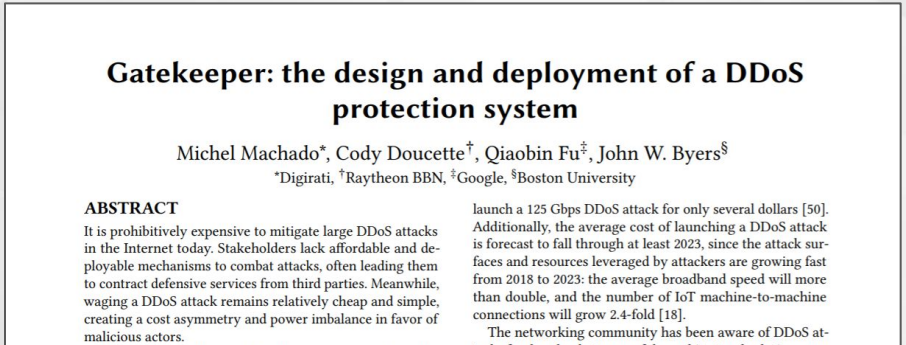
⇒ Arbitrary combinations of all kinds of attacks
(AKA multi-vector attacks)

More Details

We evaluated some of the types of attacks that Gatekeeper defends against in whitepapers

For more details, see our publications page

⇒ https://github.com/AltraMayor/gatekeeper/wiki/Publications



Gatekeeper: the design and deployment of a DDoS protection system

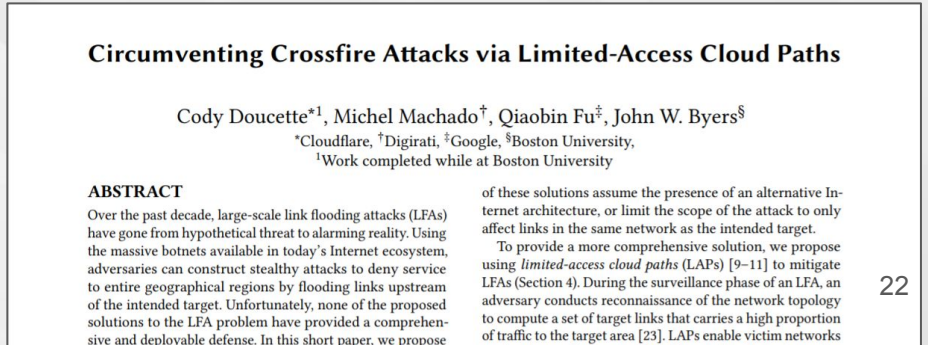
Michel Machado*, Cody Doucette†, Qiaobin Fu‡, John W. Byers§
*Digirati, †Raytheon BBN, ‡Google, §Boston University

ABSTRACT

It is prohibitively expensive to mitigate large DDoS attacks in the Internet today. Stakeholders lack affordable and deployable mechanisms to combat attacks, often leading them to contract defensive services from third parties. Meanwhile, waging a DDoS attack remains relatively cheap and simple, creating a cost asymmetry and power imbalance in favor of malicious actors.

launch a 125 Gbps DDoS attack for only several dollars [50]. Additionally, the average cost of launching a DDoS attack is forecast to fall through at least 2023, since the attack surfaces and resources leveraged by attackers are growing fast from 2018 to 2023: the average broadband speed will more than double, and the number of IoT machine-to-machine connections will grow 2.4-fold [18].

The networking community has been aware of DDoS at-



Circumventing Crossfire Attacks via Limited-Access Cloud Paths

Cody Doucette*[1], Michel Machado†, Qiaobin Fu‡, John W. Byers§
*Cloudflare, †Digirati, ‡Google, §Boston University,
[1]Work completed while at Boston University

ABSTRACT

Over the past decade, large-scale link flooding attacks (LFAs) have gone from hypothetical threat to alarming reality. Using the massive botnets available in today's Internet ecosystem, adversaries can construct stealthy attacks to deny service to entire geographical regions by flooding links upstream of the intended target. Unfortunately, none of the proposed solutions to the LFA problem have provided a comprehensive and deployable defense. In this short paper, we propose

of these solutions assume the presence of an alternative Internet architecture, or limit the scope of the attack to only affect links in the same network as the intended target.

To provide a more comprehensive solution, we propose using *limited-access cloud paths* (LAPs) [9–11] to mitigate LFAs (Section 4). During the surveillance phase of an LFA, an adversary conducts reconnaissance of the network topology to compute a set of target links that carries a high proportion of traffic to the target area [23]. LAPs enable victim networks

Gatekeeper

Outline

- ✓ Motivation
- ✓ How Gatekeeper works
- ✓ How to write a destination policy
- Conclusion

Gatekeeper

Planned releases

The latest stable version is v1.0

⇒ Released on September 15th, 2021

⇒ Deployed by Digirati and Mail.ru

Improvements coming in versions v1.1 and v1.2

⇒ Addressing production demands

Load balancing in policies planned for v2.0

⇒ Better return on investment

Wrap-up

Unparalleled multi-vector protection

Mitigation in seconds; added latency < 10μs

Scalable, open source, and ready for your deployment

Load balancing (and more) in store for the future

https://github.com/AltraMayor/gatekeeper