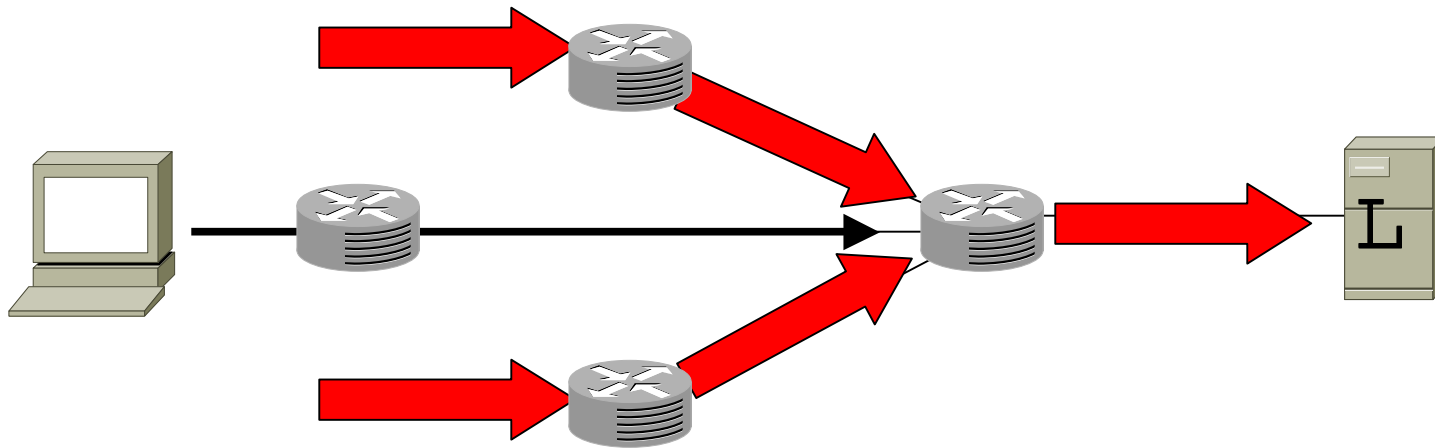


Preventing Internet Denial-of-Service with Capabilities

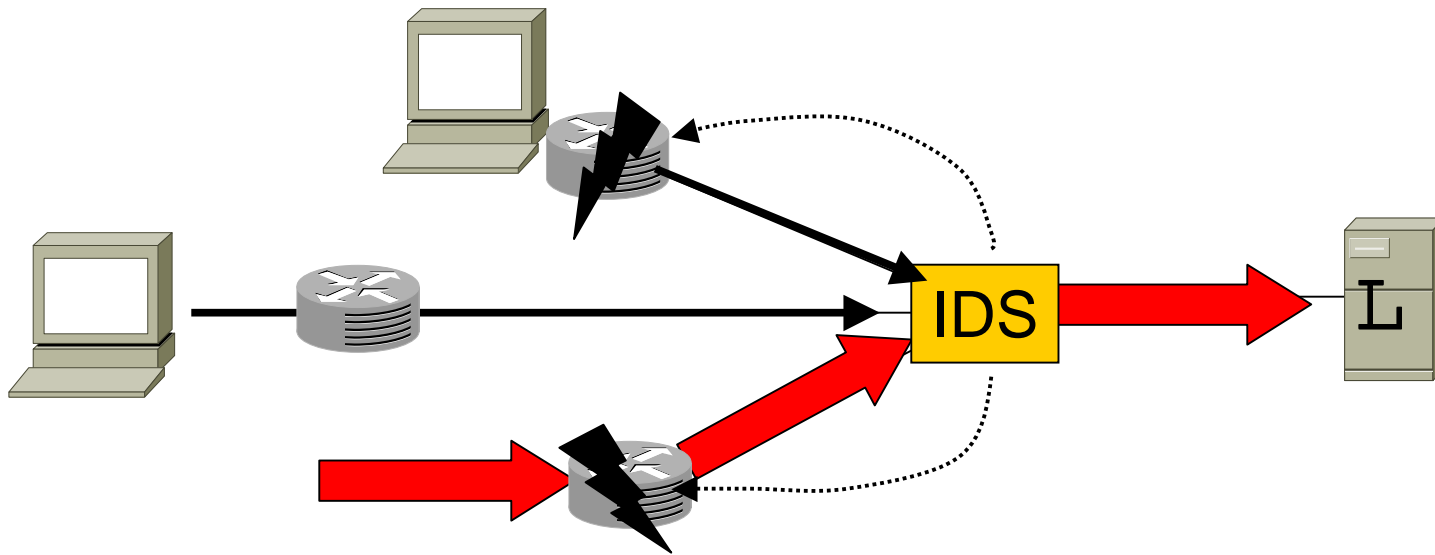
Tom Anderson, Timothy Roscoe, David Wetherall
University of Washington and Intel Research.

DDOS isn't even close to solved



- Address validation is insufficient (zombies)
- Traceback is too little too late (detection)
- Pushback lacks discrimination (E2E encryption)
- Overlay filtering can be undermined (router ACLs)

DDOS responses threaten openness



- Automated IDS alarms on unusual traffic ...
 - Clamps down on attacks and new applications alike
 - In the limit this leads to a closed system without innovation.

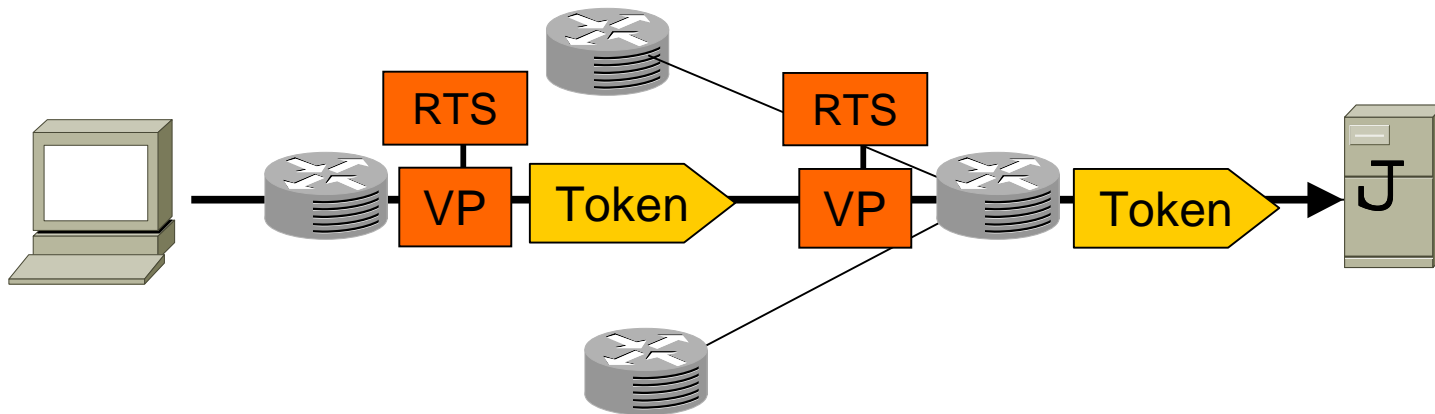
Our position

- It's time to rethink a basic premise – that anyone can send packets anywhere, any time. If not, the long-term consequence will be no security and no openness.
- We argue for a capability-based architecture that contains the damage of DDOS (security) yet allows applications to exchange any packets they want (openness)

The Need for Capabilities

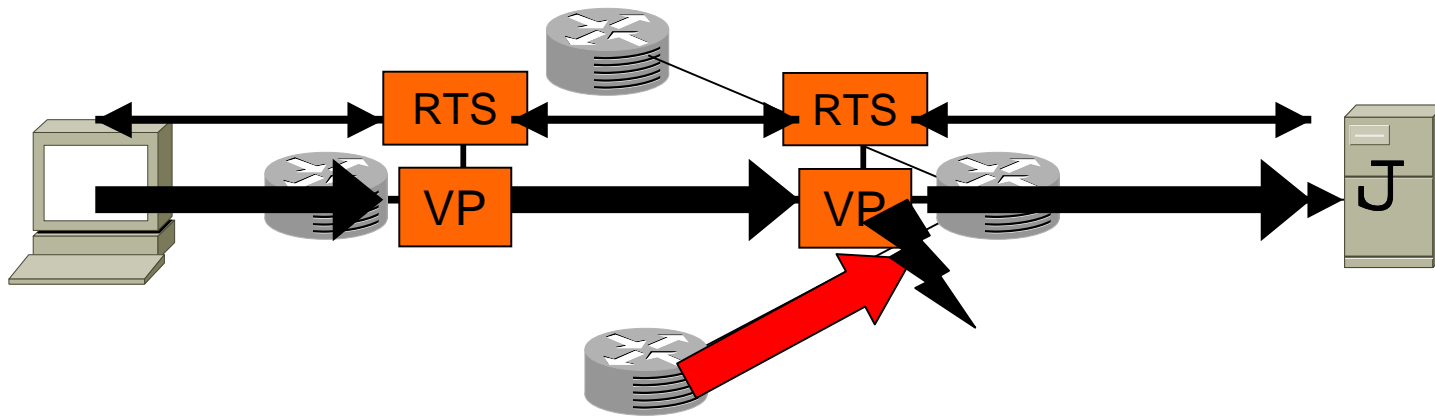
- Observe that:
 - Only destinations know which packets are legitimate
 - Only the network can shed load before it is excessive
- End result:
 - Network filtering must be based on destination control
 - Authorization needs to be explicit so it can be checked throughout the network, i.e., packets carry capabilities

Key Idea: A Capability-based Internet



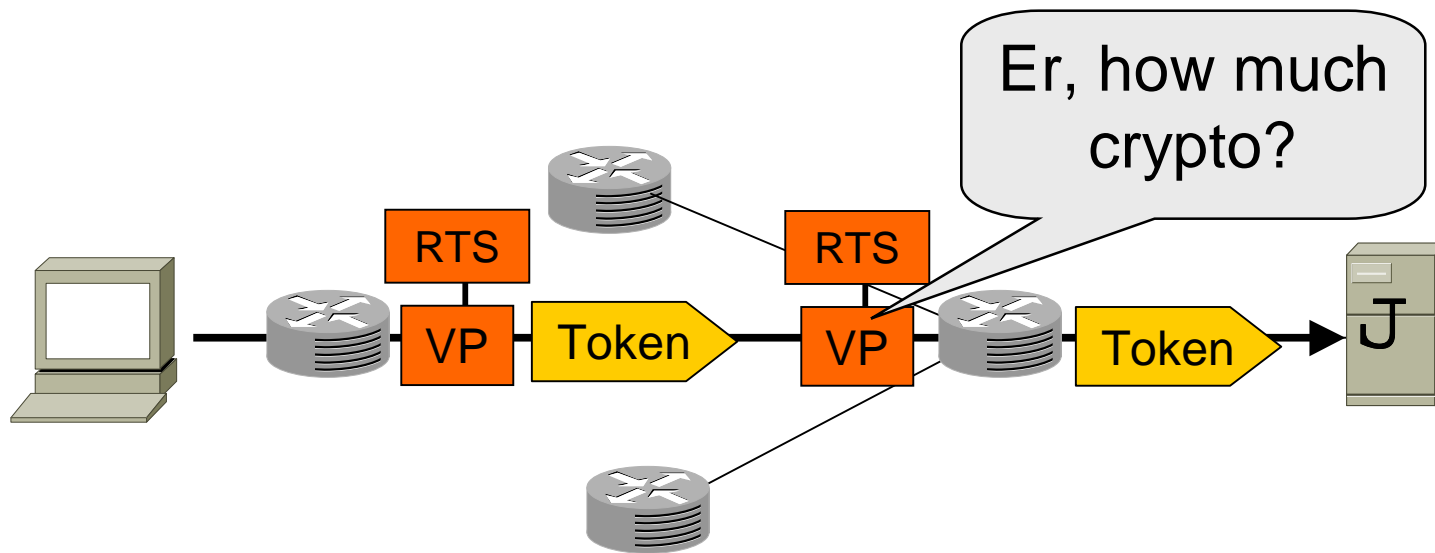
- Goal is to tightly contain the impact of attackers.
- Add two elements to the architecture:
 - Tokens, short-lived capabilities carried on packets
 - Boxes (RTS/VP), to grant tokens and filter packets

Sketch of Using Capabilities



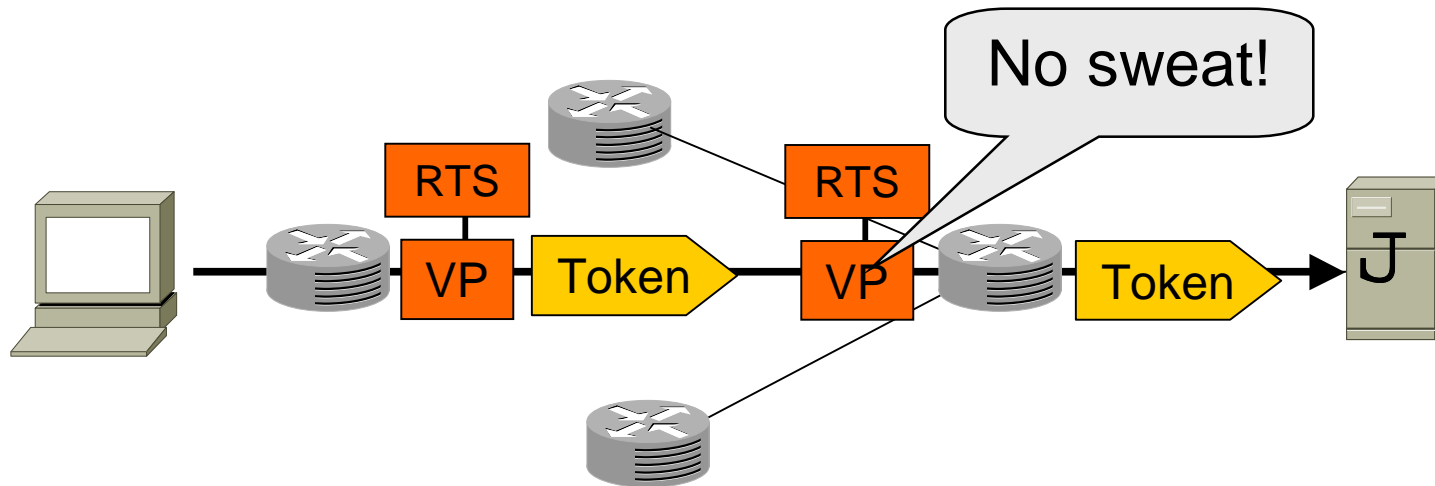
1. Source requests permission to send via RTS servers
2. Destination authorizes sources for limited transfer.
3. Source places tokens on packets and sends them
4. Network filters packets at VPs based on tokens

Encoding Capabilities with Tokens (1)



- Tokens could be PKI signatures ... but too heavyweight.
 - Would require per packet public key signature verification

Encoding Capabilities with Tokens (2)



- So just let tokens be opaque 64-bit numbers
 - Install tokens along a fixed path to prevent theft
 - OK since token is short-lived secret (e.g., 50 packets in the next 10 seconds)
 - See paper for hash-chain trick to cheaply grant next tokens
 - VPs can use simple operations to check tokens

Protecting the RTS Channel

- Must protect the RTS channel from jamming
 - Easier problem than protecting the datapath
- RTS channel has highly constrained usage
 - Only RTS traffic to/from adjacent servers/hosts
 - Use this to filter out traffic that isn't valid
- RTS channel has limited bandwidth
 - One to two orders of magnitude below datapath rates
 - Can apply more processing, e.g., per- RTS/prefix queuing

Policy Issues

- Destinations decide whether or not to authorize
 - So what's the policy?
- Simple policies can be effective
 - We have policies today, e.g., firewalls, but enforcement is weak
View capabilities as programming the network-wide firewall
- Destinations can cut off misbehaving sources
 - Just don't issue more tokens

Incremental deployment

- How do we get from here to there?
 - Look for benefit from small deployments
- Tokens need to be carried on packets
 - Most likely a shim protocol on top of IP with translation
- RTS/VPs start at ingress/egress and move outwards
 - VPs are “bumps-in-the-wire” on key datapaths; RTSes nearby
 - Advertise RTS servers via BGP

Conclusion

- It's time to rethink a basic Internet premise – that anyone can send to anywhere, any time – to improve security and preserve openness
- We argue for a new capability-based architecture:
 - Tokens make destination authorization decisions explicit
 - Packets can be checked for permission throughout the network