Countering Denial of Service
(and why it’s hard)

Katerina Argyraki, EPFL
Network systems

State

where is it stored

how is it managed

how much does it cost
Denial of service

Twitter down for hours (2009)

South Ossetian news portals down for days (2007)

BetCris.com down for months (2003)

It’s a big, unsolved problem
Denial of service

Target: tail-circuit bandwidth
Network filtering

State: \{A, R\}

Code: if ( \{packet.src, packet.dst\} in State) block packet;

Block attackers at the receiver’s gateway
State

State: \{attacker, receiver\} pairs
Where: receiver’s gateway
Managed: locally
Internet routers

control plane
(cheap) DRAM

data plane
(expensive) SRAM

Network filtering is expensive
Distributed denial of service

Target: filtering resources + tail circuit
Distributed filtering

Identify routers close to attack sources
Ask them to block attack traffic

(AlITE, Argyraki and Cheriton, 2005)
Filter propagation

filter A to R
Filter propagation

filter A to R

filter A to R
Filter propagation

filter $A$ to $R$

filter $A$ to $R$
Malicious filtering requests?

\[ R \xrightarrow{\text{filter } A \text{ to } R} G_A \]
Filter propagation continued

filter A to R

filter A to R

ACK + ticket

ticket
Filter propagation continued

$G_R$ proves it is on the path by 3-way handshake
Busy attackers?
Busy attackers?
Busy attackers?
Filter propagation continued

Keep in-network filters temporarily
Filter propagation continued

Disconnection = cheap filtering
Repeat offenders?

filter A to R
Repeat offenders?

\[ R \xrightarrow{\text{filter A to } R} A \text{ ACK + secret} \xleftarrow{\text{secret}} R \]
Repeat offenders?

- Filter $A$ to $R$
- ACK + secret
- secret
Repeat offenders?

Keep filtering state in the control plane
Non-cooperative networks?
Non-cooperative networks?

... get disconnected from R
State: \{attacker, receiver\} pairs

Where: control plane of attacker’ gateway

Managed: filter-propagation protocol
Ticket-based authorization

Give tickets to well behaved senders
Verify tickets inside the network

Need ticket distribution and verification

(SKEF, Yasar, Perrig and Song, 2004)
Ticket distribution

ticket request

ticket response
Ticket verification

Verified data + ticket

ticket request
ticket response

Thursday, October 28, 2010
Ticket verification

not verified!
data + ticket
ticket request
denied!
Stateless filtering

State: -

Code: if ( not verify(ticket) ) 
block packet;
Ticket construction

Remove filtering state from the network
State:

{sender, receiver} pairs

Where: senders

Managed: ticket-distribution protocol
Denial of ticket

Target: tail circuit + ticket distribution
Tickets + network filtering

Block attackers in the network

(TVA, Yang, Wetherall and Anderson, 2005)
Distributed denial of ticket

Target: filtering resources + tail circuit + ticket distribution
Tickets + distributed filtering

Need a filter-propagation protocol
Outsource ticket distribution

(R, S)

(Portcullis, Parno and Perrig, 2007)
Outsource ticket distribution

DNS infrastructure
Outsource ticket distribution

Target: the DNS infrastructure
State

State: \{\text{sender/attacker, receiver}\} \text{ pairs}

Where: \text{senders} + \text{network}

Managed: \text{ticket distribution} + \text{filtering propagation}
Fair-share the Internet

Fixed number of connections per sender
Reduces filtering state

Changes the nature of the Internet
Conclusion

Identify minimum state

Move it where it is easy to manage

Beware of hidden state!