HTTPrej ect: Handling Overload Situations without Losing the Contact to the User

Jörg Schneider
Sebastian Koch
Flash effect based DDOS

- Flash effect
  - High increase in requests by *legitimate* user.
  - User retry until success

- Examples
  - Transport network break down (ash cloud)
  - Important news (accidents, votes)
  - “slashdot effect”
Idea: Stop the user

Assumption:

- The user will stop, when he gets the information requested.
- … or at least an error message

Low resource IP level filter to block requests with small HTTP messages
Related work

- Cloud: dynamic capacity
  - Website needs to be „cloudified“
  - Who pays the bill?
- Content distribution networks / P2P Cache
- DDoS attacks
  - Filtering packets to reduce request rate
  - SYN cookies
Ideal HTTP request

TCP SYN
TCP SYN/ACK
TCP ACK*
HTTP GET
HTTP answer
TCP ACK
TCP FIN
TCP FIN/ACK
TCP ACK
Stateless HTTP/TCP

1. Wait for next packet
2. Check if SYN set
   - YES: Answer with SYN/ACK and seq.nr. n
     - YES: Answer with HTTP answer and ACK received packet
     - NO: Answer with FIN/ACK
   - NO: Check if payload.length > 0
     - YES: ACK Seq. = n+1+payload.length
     - NO: Just ACK the received packet
Performance evaluation

Server
- Candidates
  - HTTPreject
  - Apache 2.2
  - lighttpd 1.4
- Deliver same website

Simulated Clients
- 16 processes
  - Own IP
  - Up to 1024 simultaneous connection
  - Fixed rate of connections: 20,800 con/s

- 4 Xeon cores
- 16 Xeon cores
Results: Server Load

server load (%)

- apache: 98.47%
- lighttpd: 42.88%
- httpreject: 3.54%
Results: Connection time

avg. connection time (s)

- Apache: 0.43
- Lighttpd: 0.32
- Httprej: 0.01
Conclusion and Outlook

- Blocking HTTP requests on IP level
  - Meaningful short message
  - No state/context needed
  - Low resource usage
  - Simplified TCP and HTTP

- Future work
  - User study – Will they stop reloading?
  - Broader compatibility study
HTTPrejpect: Handling Overload Situations without Losing the Contact to the User

Jörg Schneider <komm@cs.tu-berlin.de>
Sebastian Koch <seb@cs.tu-berlin.de>
http://www.kbs.tu-berlin.de/