Co-operative DDoS Mitigation

draft-reddy-dots-transport-03

April 2016
IETF 95
Authors: T. Reddy, D. Wing, P. Patil, M. Geller,
M. Boucadair, R. Moskowitz
Presenter: Dan Wing
Changes to draft

• Meets most of the requirements in draft-ietf-dots-requirements-01
  ▪ Happy Eyeballs-like technique for DOTS Signal Channel (v6/v4, UDP/TCP)
  ▪ DOTS Signal Channel, UDP/TCP
  ▪ DOTS Data Channel, TCP
  ▪ CoAP for lightweight communication
  ▪ Performance considerations
Happy Eyeballs-like technique for DOTS Signal Channel

DOTS client

- DTLS ClientHello, IPv6
- TCP SYN, IPv6
- DTLS ClientHello, IPv4
- TCP SYN, IPv4
- TCP SYN ACK, IPv4
- DTLS ClientHello, IPv4
- TCP ACK, IPv4

DOTS server

- TLS Session
- DOTS signal
Happy Eyeballs for DOTS Signal Channel

• Order of preference (aligns with RFC6724)
  ▪ IPv6 over DTLS over UDP
  ▪ IPv6 over TLS over TCP
  ▪ IPv4 over DTLS over UDP
  ▪ IPv4 over TLS over TCP

• DNS lookup during peacetime.
  ▪ DNS-SD will be aligned with requirements
DOTS signal channel

- DOTS client
- DOTS server

(D)TLS Session

POST : request to convey DOTS signal
200 OK

PUT : efficacy update from DOTS client
200 OK

GET : status of attack
200 OK {"status":"attack stopped"}

DELETE : withdraw DOTS signal
200 OK
DOTS data channel

POST: filtering rules to black/white-list-list traffic

Delete: remove filtering rules

200 OK
Why CoAP for DOTS?

- Constrained Application Protocol (CoAP)
- CoAP runs over both DTLS over UDP (RFC7252) and TLS over TCP (draft-ietf-core-coap-tcp-tls).
- CoAP is designed according to the REST architecture.
- CoAP integrates with JSON, CBOR or any other data format.
- Asynchronous message exchanges.
- CoAP proxy.
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Code Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOTS</td>
<td>No code</td>
</tr>
<tr>
<td>CoAP</td>
<td>Code available</td>
</tr>
<tr>
<td>TLS</td>
<td>Code available</td>
</tr>
<tr>
<td>DTLS</td>
<td>Code available</td>
</tr>
<tr>
<td>TCP</td>
<td>Code available</td>
</tr>
<tr>
<td>UDP</td>
<td>Code available</td>
</tr>
<tr>
<td>IP</td>
<td>Code available</td>
</tr>
</tbody>
</table>

http://coap.technology/impls.html
# Use Cases

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Addressed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.1</td>
<td>Request mitigation, from mitigator</td>
<td>Yes</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Request mitigation, from network infrastructure</td>
<td>Yes</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Request mitigation, from telemetry system</td>
<td>Yes</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Request mitigation, from targeted application</td>
<td>Yes</td>
</tr>
<tr>
<td>3.1.5</td>
<td>Request mitigation, from web portal</td>
<td>Yes</td>
</tr>
<tr>
<td>3.1.6</td>
<td>Request mitigation, from mobile device application</td>
<td>Yes</td>
</tr>
<tr>
<td>3.1.7</td>
<td>Unsuccessful mitigation request</td>
<td>Yes</td>
</tr>
<tr>
<td>3.2.1</td>
<td>DOTS client registration</td>
<td>Yes</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Auto-provisioning of DDoS countermeasures</td>
<td>Yes</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Attack notification to 3(^{rd}) party</td>
<td>NO</td>
</tr>
</tbody>
</table>

Reference: draft-ietf-dots-use-cases-01
Performance considerations

• (D)TLS session resumption without server-side state [RFC5077].
• TLS False Start [I-D.ietf-tls-falsestart].
• Cached Information Extension [I-D.ietf-tls-cached-info].
• Raw public keys [RFC7250].
• (D)TLS Heartbeat.
• TCP FastOpen [RFC7413].
Draft Reddy Dots Transport 03

- Consensus on Happy Eyeballs-like technique?
- Consensus on CoAP?