Combating DNS amplification attacks using Cookies

Supervisor:
Roland van Rijswijk
SURFnet

By:
Sean Rijs
Agenda

● I am going to do my presentation
DNS amplification attacks

1) query ANY delaat.net
2) query ANY delaat.net
   response 1.1.1.1.... (and cache)
3) Response 1.1.1.1....

Response size
---
Query size

Table by Rijswijk-Deij et al. [DNSSEC and its potential for DDoS attacks]
DNS Cookies
IETF Internet Draft

- By Donald Eastlake 2006-2014
  - Authentication of source IP
  - Off-path

- No pre-configuration required

- Research question:
  - Is the draft effective against DNS amp. attacks?
Terminology confusing?
Cookies
OPT RR
(EDNS0)

| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| OPTION-CODE = TBD | OPTION-LENGTH = 18 |
| Resolver Cookie | hash(Resolver Secret | Server IP Address) |
| Server Cookie | hash(Server Secret | Query IP Address | Resolver Cookie) |
| Error Code |

- May occur once
- Proposed hash = FNV-64
- Max. 22 bytes
• Costs?
  - Initially 2x RTT
  - Hashing
  - Caching
What if?

Stub resolver (Attacker) -> Recursive server (Open Resolver) -> Authoritative server

1

2

3

③ just contains small error messages, no big amplification

Target
Policy

- **Disabled**: do nothing with cookies
- **Enabled**: opportunistic (recommends RRL on server side)
  - Not a solution for recursive servers
- **Enforced**: Ignore everything without Cookies
  - Not gonna happen (in the near future)
- **Policy is important**, as it determines the incremental implementation
Source Identity Token (SIT)

- BIND 9.10-P1 (two months ago)

2x RTT has disappeared?
Differences SIT / Internet Draft

- Similar except:
  - Hashing: FNV-64, AES-MAC, SHA1, SHA256
  - RRL: whitelists valid clients
  - Policy: no one is going to use it
Analysis of impact

- Stub resolvers are stateless
- A lot of end devices: bound by release cycles
- Recursive server and authoritative are stateful
- Already use RRL
Measurements

• What do want to find out?
  – Do we need EDNS0 for normal use?
  – Do we need large response sizes for normal use?

• How?
  – PCAPs and EEMO
Measurement sources

- **Stub resolver:** [www.nu.nl](http://www.nu.nl) (with its adds) using:
  - Windows - Internet Explorer
  - OS X - Safari
  - Ubuntu Linux – Firefox
- **Stub resolver:** Alexa top 10 using:
  - Ubuntu Linux - Firefox
- **Recursive server:** SURFnet
  - 1500 – 2000 queries per second
  - 10m during a workday on noon
Stub resolver

- No EDNS0 found
- No large response responses:
  - Size <= 512 bytes
  - truncated/TCP communication = 0
Recursive server

- 22% EDNS0
- Average size
  - 133 B
- 99% ≤ 240 B
Conclusion/Discussion

- Based on our results, we suggest unauthenticated stub resolvers should be limited to a max. response size of 240 bytes.
- Amplification reduced further:
  - 240 bytes = 6 amplification factor
  - 100M = 600 Mbit/s

<table>
<thead>
<tr>
<th>Q-Size</th>
<th>R-Size</th>
<th>Amplification factor</th>
<th>Attacker</th>
<th>Victim</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>512</td>
<td>12.8</td>
<td>100M</td>
<td>1.28G</td>
</tr>
<tr>
<td>40</td>
<td>1472</td>
<td>36.8</td>
<td>100M</td>
<td>3.68G</td>
</tr>
<tr>
<td>40</td>
<td>4096</td>
<td>102.4</td>
<td>100M</td>
<td>10.24G</td>
</tr>
</tbody>
</table>

Table by Rijswijk-Deij et al. [DNSSEC and its potential for DDoS attacks]
Conclusion

- RRL should not be used
  - Especially on recursive server
  - But authoritative can also be effected
- Policy for incremental implementation must be changed
- Terminology:
  - stub/recursive/authoritative
  - The cookie is actually a Message Authentication Code (MAC) and not just a hash
Discussion

• Do we need to authenticate the server?
• Yes, it provides off-path defense against:
  – Last mile problem in DNSSEC
  – Cache poisoning (by Kaminsky)
Future research

• Need more measurements
  – to confirm suggested DNS maximum response size

• FNV-64
  – The non-standard and untested hashing algorithm, which could provide performance gain. Is a performance gain required?
Questions