efficient iP™
DEFINING SMART DDI

Your Cache DNS server also requires your attention

Nico CARTRON nc@efficientip.com

SwiNOG, Bern 5 November 2015

What we’re gonna talk about

Quick Facts About Cache & Recursive DNS

Sloth DNS Attack – What is it ?

Sloth DNS attack – How to protect the DNS Server

Who am I and who is EfficientIP?

Nico CARTRON:

Global Telco Specialist at EfficientIP
10+ years experience in this market

EfficientIP:

Software vendor specialising in DDI (DNS/DHCP/IPAM)

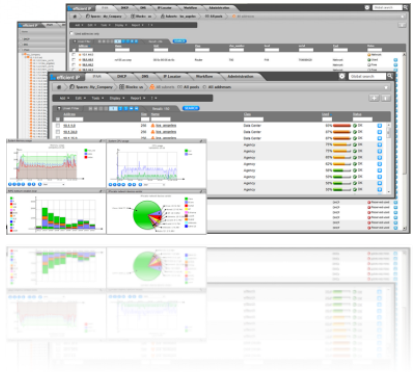
Americas & European Headquarters – (USA & France)

Offices: USA, UK, Germany, Spain & France

SOLID™ server

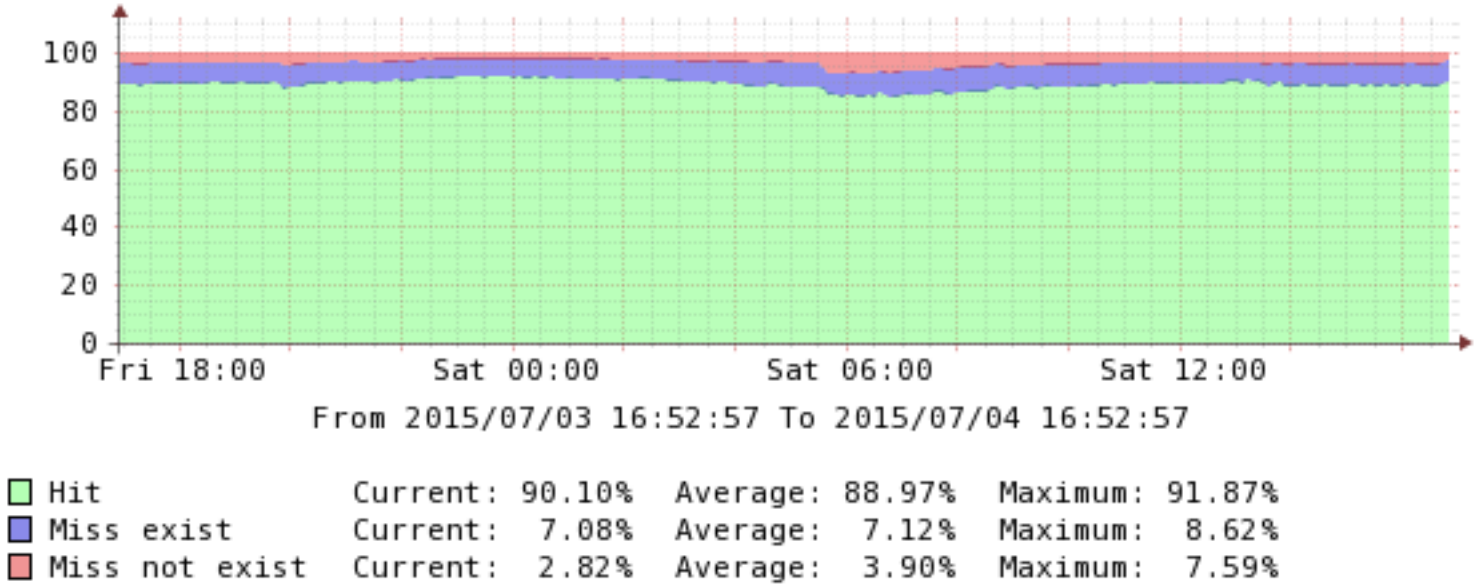
Features:

- IP addressing & VLAN Plan Management
- Network Services Engines : DNS-DHCP-NTP-TFTP
- Multi-Vendor DNS&DHCP Services Management
  (Microsoft – ISC – SOLIDServer)
- Device Deployment Management
- Network Discovery & Configuration Management



Quick facts about Cache & Recursive DNS

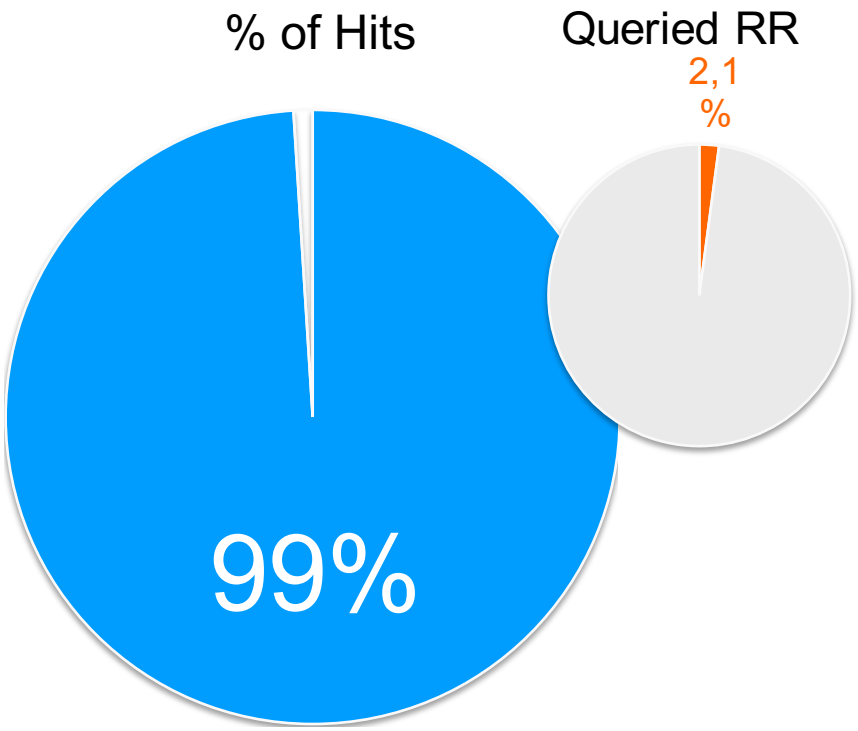
This is How DNS Traffic on an ISP Typically Looks Like

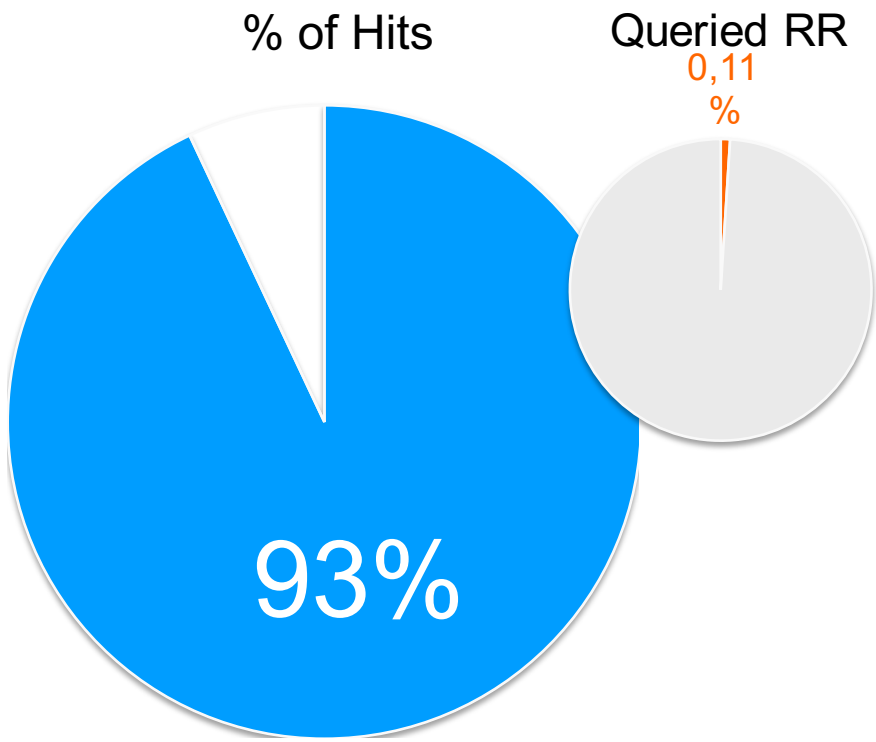


SO...Nothing Really New, Right?

Cache Hit: 85 to 91% (avg 87%)
Cache Miss Exist: 7 to 8%
Cache Miss Not Exist: 4 to 5%

Very Few Number of RRs Deliver the Most Part of the Answers

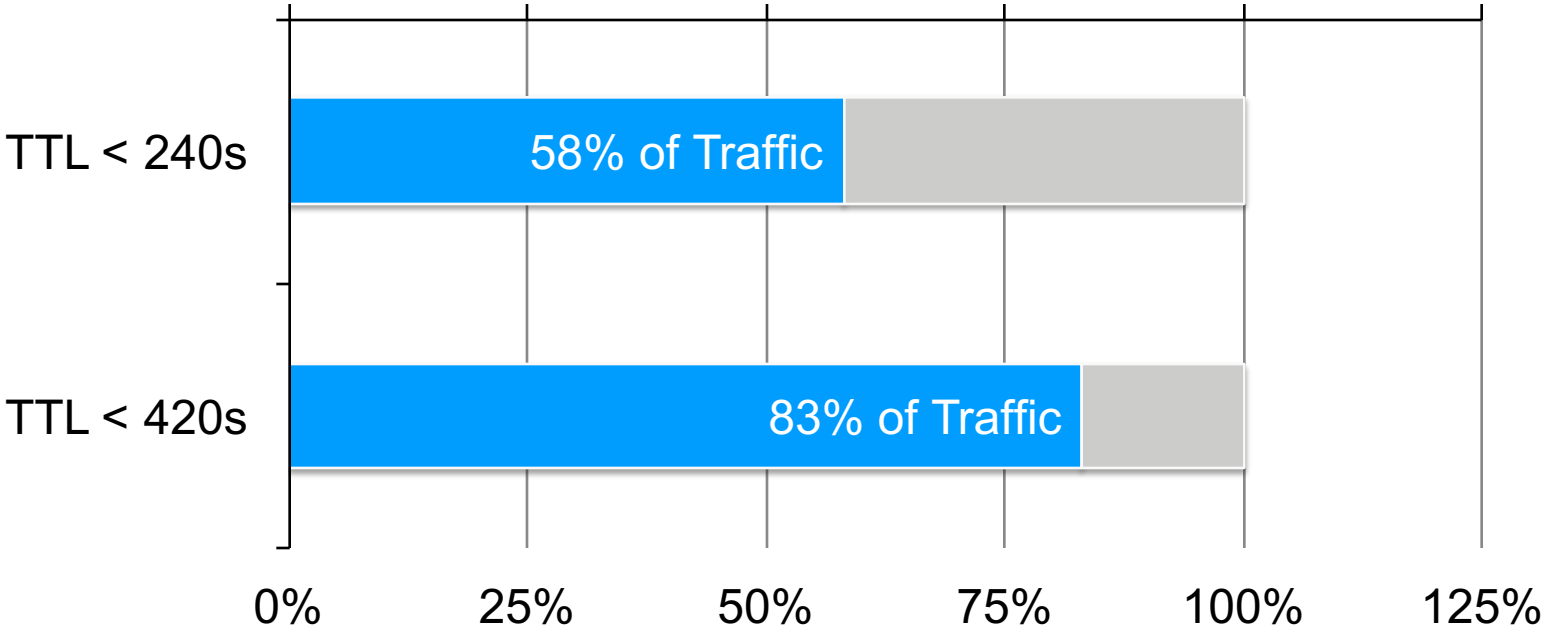




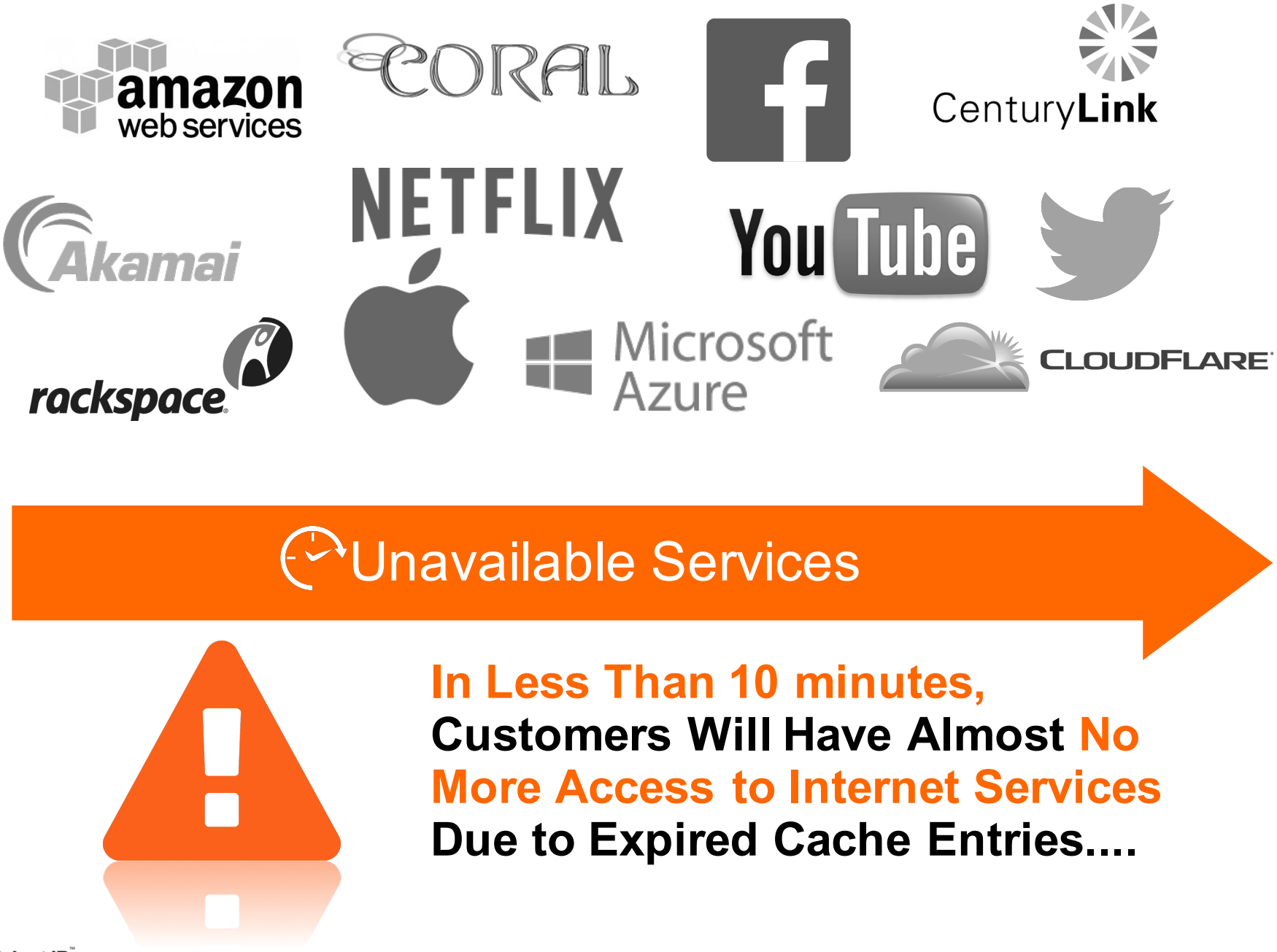
Sample: 3,6 billions requests on 14,5 million RRs

TTL of Top Requested RRs is Short…

Even Very Short!



…Cloud Services Accelerate Short TTL Deployments For Resiliency Purposes…



In Less Than 10 minutes,
Customers Will Have Almost No More Access to Internet Services Due to Expired Cache Entries....

DNS Recursive Function is Highly Vulnerable to High & Low-Volume DoS Attacks

DNS Recursive Function Performance is Limited By Design: Maximum of 10K simultaneous recursions

Few Compromised Clients Can Quickly Cause Exhaustion of DNS Server Capacity

Performance Imbalance Between Clients & Server Recursion

Low-Volume Attack Threats On DNS Recursive Needs Your Attention!

Huge Business Impacts

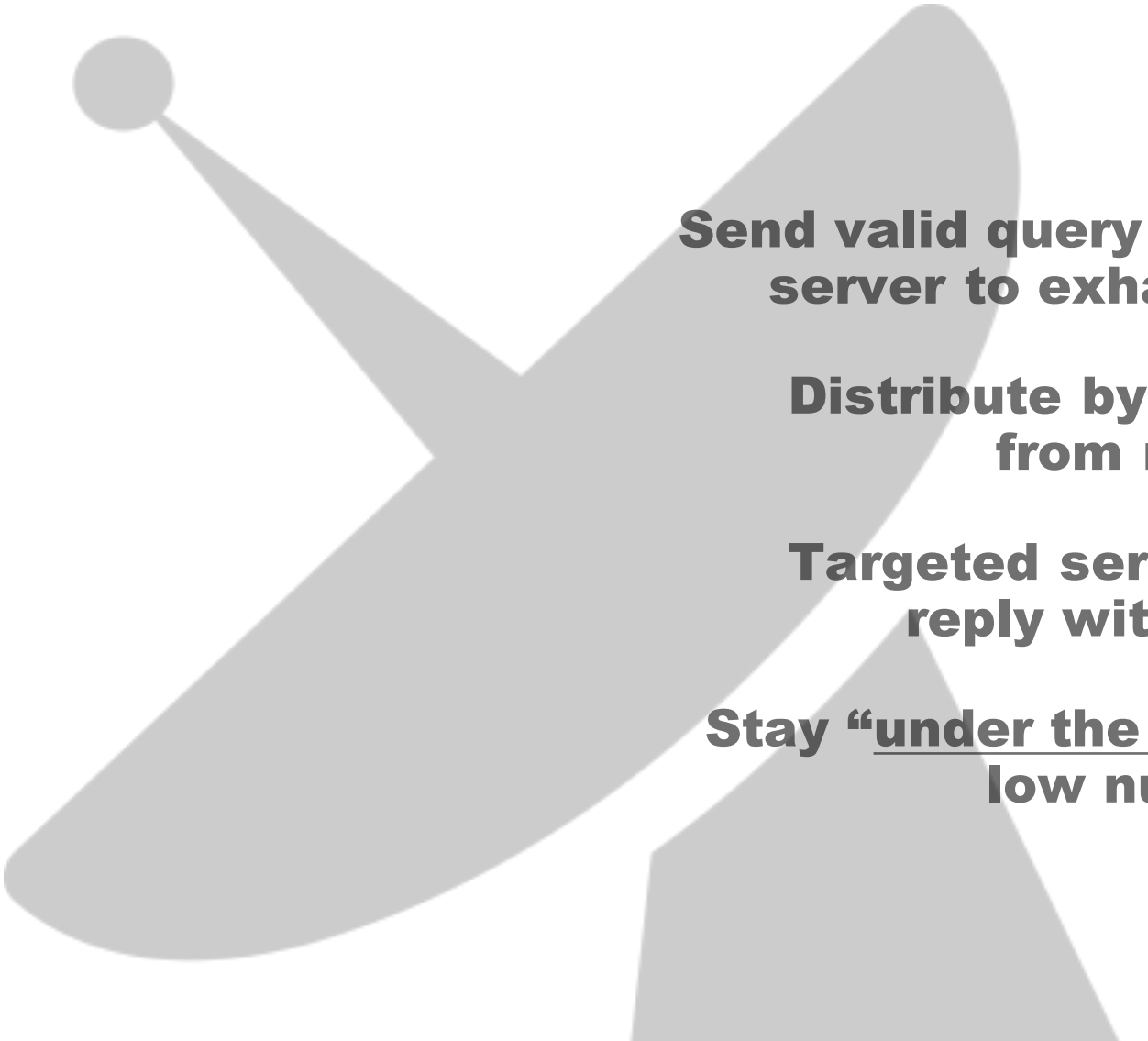
Low-Volume DNS DoS Attacks Are Less Complex to Setup

Very Difficult To Mitigate

Can be distributed

Sloth DNS Attacks
What is it?

Objective: Stealthy Low-volume DoS Attacks on DNS Recursive Function



Send valid query to targeted DNS server to exhaust its capacity

Distribute by sending queries from multiple sources

Targeted server must always reply with a valid answer

Stay "under the radar" with very low number of queries

...So... How Do We Do This? Make it Slow!

1. **Setup an Authoritative DNS server**
   Say a "normal" BIND
   With "max-recursive-clients" set to 10.000 (already quite high)

2. **Patch the code to introduce latency for each answer**

```
#include <unistd.h>
unsigned int sleep(unsigned int seconds);
```

3. **Then launch some queries on the targeted Recursive server**

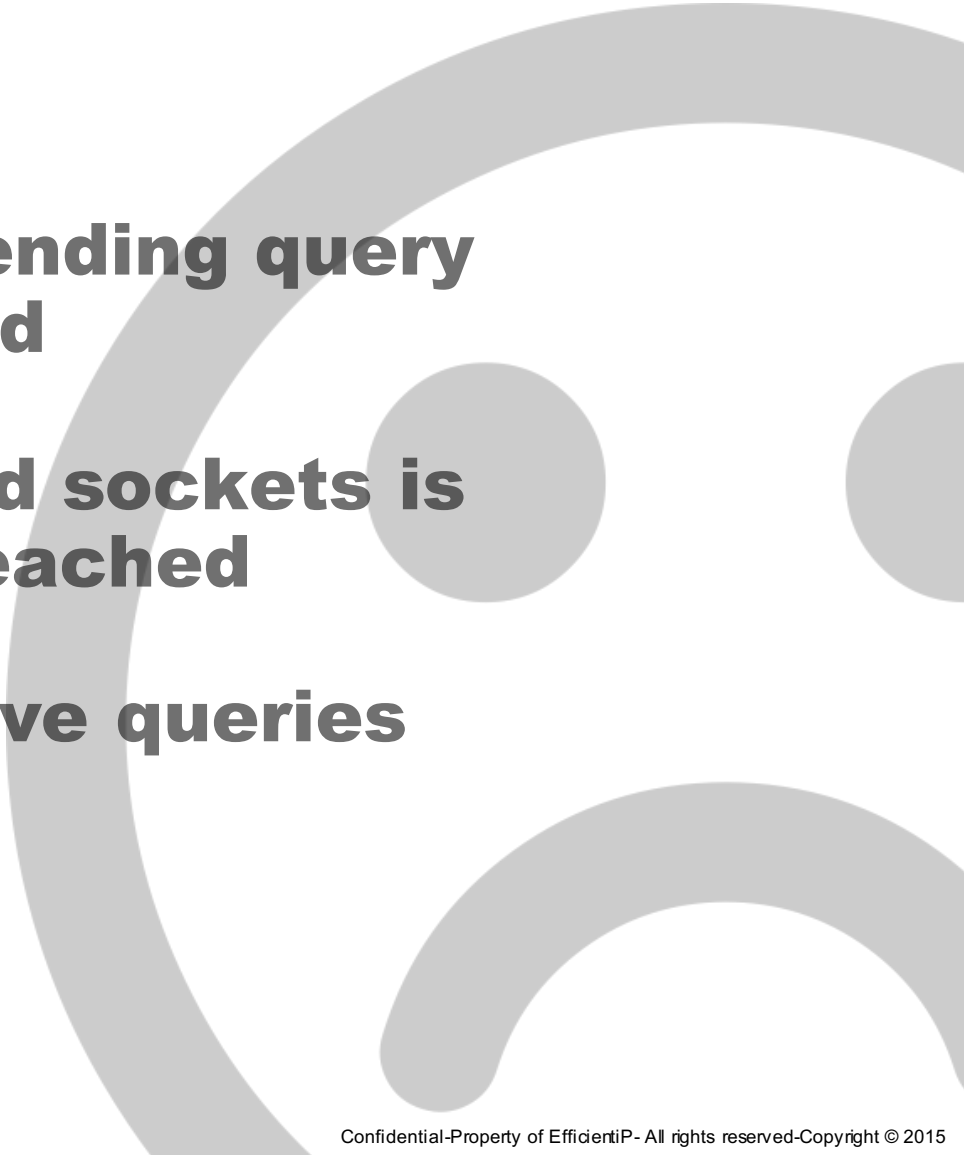
"Some" being between 5k and 10k QPS, according to the latency introduced 10.000 if latency = 1", 5000 if latency = 2", ...

Why the DNS Stops Resolving?

For each recursive pending query a UDP socket is bound

The number of opened sockets is limited and quickly reached

Then no more recursive queries can be processed



Sloth DNS Attacks
How to protect the DNS server?

We need to rethink the indicators used to check Cache/Recursive DNS servers performances

Usual technics do not work: not a Volumetric attack

Per IP statistics on time spent on recursion
According to the results, take decisions

Tricky part is that only taking into account time spent on recursion
Example when querying a public Recursive DNS server in Vietnam:

dig mit.edu @115.78.230.81
>300ms every time, even for cached requests!

Shall we let the Recursive layer stay the bottleneck?

Let the attack on the Recursive layer overwhelm the Cache part?

Graduated answer:

- blocking,
- forbid recursion,
- answer from the cache

=> Less false-positive risks (see CPE)

The bigger cache we have, the more efficient this technic is.

Conclusion

Most of the efforts against DNS DDoS attacks = Authoritative servers
Very little done to protect the Cache/Recursive Layer
=> Because it's much more complex (much more RR to handle)

How about automatically block, forbid recursion, answering from cache?
=> Less false-positive risks (see CPE)

ISC has integrated this approach since BIND 9.10.3 (–enable-fetchlimit -> fetches-per-server & fetches-per-zone), **while still disabled by default because of the collateral damage**

Questions?!

nc@efficientip.com