The Maginot Line: Attacking the Boundary of DNS Caching Protection

Balaji Balachandran

A little bit about conditional DNS (CDNS)

- Acts as a recursive resolver and forwarder
- All queries fit into one of two categories
    - Recursive DNS zones, Z_R
    - Forwarding DNS zones, Z_F
- Shared global cache between resolver and forwarder

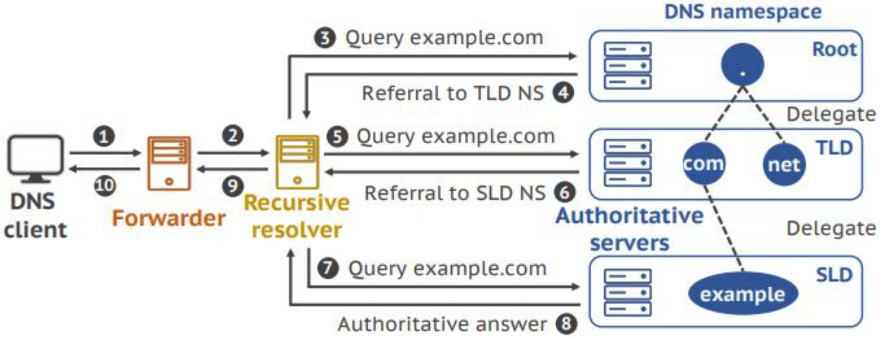
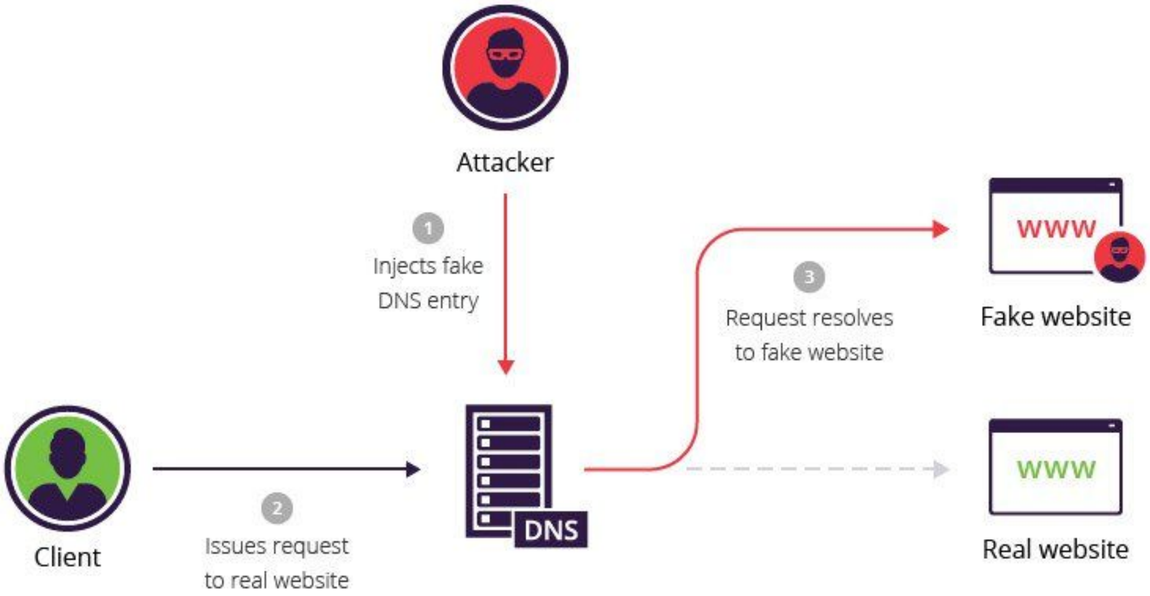


Figure 1: A standard DNS resolution process for domain `example.com` under the DNS namespace.

- *MaginotDNS targets queries for domains in the forwarding DNS Zone*
    - *i.e.* $d_{attack} \in Z_F$

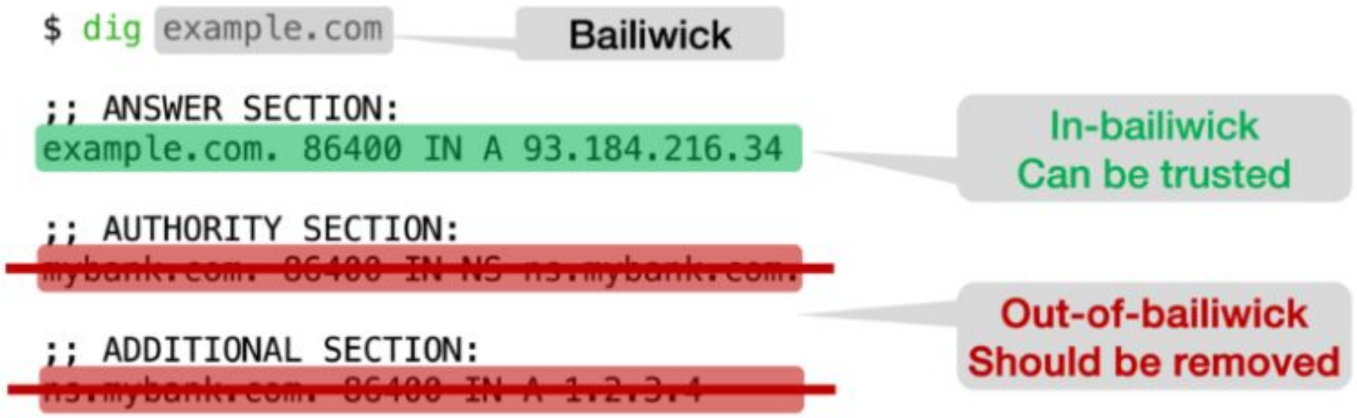
Cache Poisoning



https://www.imperva.com/learn/application-security/dns-spoofing/

Bailiwick Rules

- Don’t accept responses from an authoritative DNS that fall outside the scope of authority
- Prevent malicious authoritative servers from providing DNS mappings



https://blog.apnic.net/2023/09/26/maginotdns-attacking-the-boundary-of-dns-caching-protection/

Attack Taxonomy

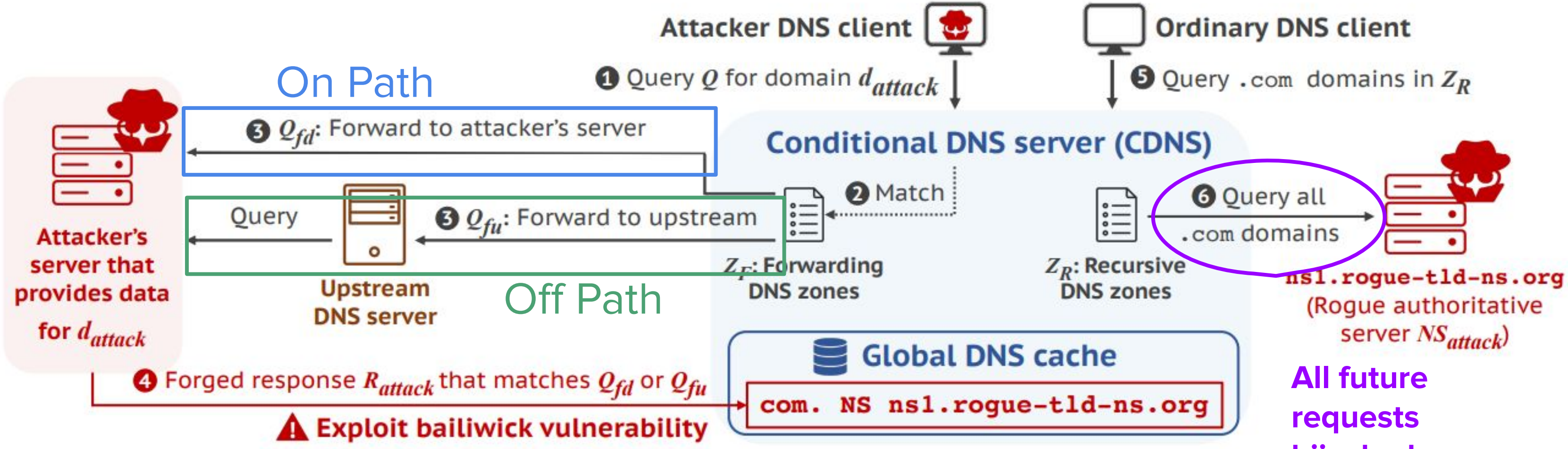


Figure 4: Threat model and attack workflow of MAGINOTDNS.

All future requests hijacked

- Bailiwick seems like a reasonable defense against cache poisoning
- Bailiwick checks are adequately enforced for recursive resolvers…
- …not so much for forwarders
- When we leverage the shared cache of a forwarder and resolver, we can manipulate the forwarder and enable cache poisoning

Maginot Line: “A defensive barrier that inspires a false sense of security”[1]

- “Cross the boundary”

[1] Merriam Webster, https://www.merriam-webster.com/dictionary/Maginot%20Line

Table 1: DNS operational modes and functionalities available in mainstream implementations.

DNS software		Server role					Cache protection			Cache poisoning defense	
Brand	Version	Auth[1]	Recur[2]	Fwder[3]	CDNS	Fall-back	Bailiwick checking	Trust level	Shared cache	DNSSEC	0x20
BIND [12]	9.18.0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Knot Resolver [77]	5.5.2	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓
Unbound [91]	1.16.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PowerDNS Recursor [75]	4.7.1	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓
Microsoft DNS [87]	2022[4]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Technitium [89]	7.0	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓
Simple DNS Plus [73]	9.1.108	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
MaraDNS [67]	3.5.0022	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓
CoreDNS [22]	1.9.3	✓	✓[5]	✓	✓[5]	✓[5]	✗	-[6]	✗	✓	✗
Dnsmasq [33]	2.86	✗	✗	✓	✗	-	✗	✗	-	✓	✗
DNRD [26]	2.20.3	✗	✗	✓	✗	-	✗	✗	-	✗	✗
YADIFA [94]	2.5.4	✓	✗	✗	✗	-	-	-	-	✓	✓
NSD [72]	4.6.0	✓	✗	✗	✗	-	-	-	-	✓	✓

[1] Authoritative server. [2] Recursive resolver. [3] Forwarder. [4] OS build 20348.740. [5] Available only when compiled with extra Unbound extensions. [6] "-" means not applicable.

Pulling it off

1) Probe or use software fingerprinting to find CDNSes
2) Craft DNS response with enough trust level to overwrite the cache
3) Manipulate future queries

Finding vulnerable DNS ports

- Attack in ‘rounds’
- Brute force attacking to determine vulnerable dns ports
    - Relies on the birthday paradox
- On average <15 minutes to execute the attack
- Traffic rate is significant. Should this be a red flag to DNSes?

Table 3: Microsoft DNS and BIND off-path attack results.

Software	**Time of each round**	**Avg time taken**	**Max traffic rate**	**Success rate**
MS DNS	5s	802s	216Mbps	20/20
BIND	1.2s	790s	54Mbps	20/20

$$1 - [(28,232 - 50)/28,232]^{3600} = 99.8\% \qquad (2)$$

Identifying CDNSes

- Probe a subset of DNS zones to determine when CDNSs
  - Use Alexa's Top 10k sites
- Of the **370,512** DNS that support cache probing, **154,955** could be identified as CDNSes (41.8% of probed)
- **54,949** vulnerable CDNSes (14.8% of probed)
  - All vulnerable to on path attacks
  - 88.3% vulnerable to off path attacks

Table 4: Open DNS servers and CDNS statistics.

DNS Server Type	# IP	% of Probed	% of CDNS	% of Vuln.
DNS servers on Feb. 14, 2022	1,499,110	–	–	–
DNS servers alive on Mar. 14, 2022	**1,215,918**	–	–	–
– Not following non-recursive	839,017	–	–	–
– Using multiple caches	401,186	–	–	–
– Supports cache-probing	**370,512**	**100%**	–	–
– Version identifiable	237,835	64.2%	–	–
– DNSSEC validation	86,955	23.5%	–	–
– 0x20 encoding	1,619	0.4%	–	–
CDNSes identified by probing	**154,955**	**41.8%**	**100%**	–
– Version identifiable (in CDNS)	117,306	31.7%	75.7%	–
– by `version.bind`	59,419	16.0%	38.3%	–
– by `fpdns`	57,887	15.6%	37.4%	–
– OS identified for BIND (in CDNS)	19,995	5.4%	12.9%	–
– DNSSEC validation (in CDNS)	34,424	9.3%	22.2%	–
– 0x20 encoding (in CDNS)	1,119	0.3%	0.7%	–
Vulnerable CDNSes	**54,949**	**14.8%**	**35.5%**	**100%**
– On-path attack possible*	**54,949**	**14.8%**	**35.5%**	**100%**
– BIND	24,287	6.6%	15.7%	44.2%
– Microsoft DNS	30,662	8.3%	19.8%	55.8%
– Off-path attack possible*	**48,539**	**13.1%**	**31.3%**	**88.3%**
– BIND (OS exploitable)	17,877	4.8%	11.5%	32.5%
– Microsoft DNS	30,662	8.3%	19.8%	55.8%
– Recursive-default	10,445	5.0%	11.9%	33.4%
– Forwarding-default	36,581	9.9%	23.6%	66.6%

* On-/Off-path attack possible: CDNSes equipped with non-empty Z_F and vulnerable software versions/OSes. Because we lack vantage between CDNSes and upstream servers, we can only confirm they are vulnerable to on-/off-path attacks, but cannot further identify which domains in Z_F can be actually exploited by each type of attack.

Implies trust level 6

```
Flags: QR AA RD;
Question section:
attacker.com. A
Answer section:
attacker.com. A a.t.k.r
Authority section:
com. NS ns1.rogue-tld-ns.org.
Additional section:
ns1.rogue-tld-ns.org. A a.t.k.r
```

(a)

Microsoft DNS / BIND

```
Flags: QR AA RD;
Question section:
attacker.com. NS
Answer section:
attacker.com. CNAME com.
Authority section:
(Empty)
Additional section:
(Empty)
```

(b)

Knot pt. 1

```
Flags: QR AA RD;
Question section:
com. NS
Answer section:
com. NS ns1.rogue-tld-ns.org.
Authority section:
(Empty)
Additional section:
ns1.rogue-tld-ns.org. A a.t.k.r
```

(c)

Knot pt. 2

```
Flags: QR AA RD;
Question section:
attacker.com. A
Answer section:
(Empty)
Authority section:
com. NS ns1.rogue-tld-ns.org.
Additional section:
ns1.rogue-tld-ns.org. A a.t.k.r
```

(d)

Technetium

```
Flags: QR AA RD;
Question section:
attacker.com. A
Answer section:
attacker.com. A a.t.k.r
Authority section:
(Empty)
Additional section:
(Empty)
```

(e)

Prevents Fallback

Attack Impact

- Attackers can take over entire DNS zones
    - Including top level domains (.net, .com, .edu, etc.)
- Poisoned cache relinquishes control to attackers
- Can insert malware, phishing, etc.

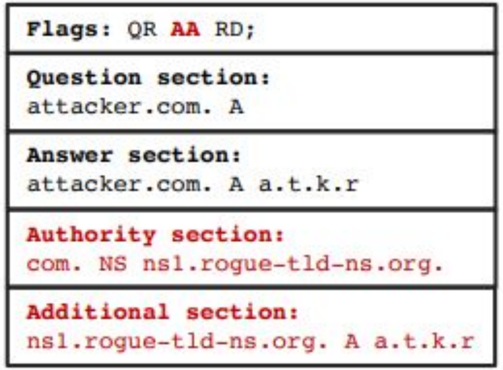
Mitigation

- 0x20 encoding
  - Randomly change the case of each character in a query
  - Difference between uppercase and lowercase is the 6th bit in ASCII (0x20)
  - defends against MaginotDNS Off-path
- DNSSEC validation
  - Validates the sender
  - defends against On-path and Off-path MaginotDNS attacks
  - When probed, simply returns a *SERVFAIL*

Discussion

- Is this a large threat? DNSSEC is an effective countermeasure already, does this take away from the novelty of the attack?
- All DNS vendors have acknowledged and have now remediated all issues
- ~70% of the world's DNS servers are running BIND. Is that an issue?
- Why was this discovered just recently? Microsoft DNS and BIND are mature products.
- Why isn't DNSSEC used extensively in practice?

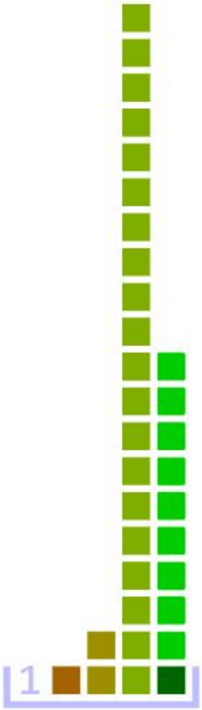
- Why is it so easy to spoof trust with the AA flag?
- RFCs specify bailiwick checks at a high-level. Why the implementation to standard gap?
- DNSSEC requires overhead to verify responses. Is the attack serious enough to be worth the tradeoff?
- This research was supported in part by Microsoft

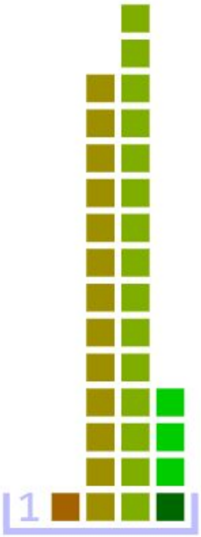


(a)

General Consensus

“Not so novel, mitigation techniques exist already”





“Thorough in their analysis, attack is interesting”