Bitflipping root-servers.net

Peter Thomassen <peter@desec.io>

IPEG pre IETF 125 – Shenzhen – March 15, 2026

Bitflips

*"Computer hardware, especially RAM, can suffer from **random errors** that manifest as **corruption of one or more bits**. The causes of these errors range from manufacturing defects to **environmental factors such as cosmic rays and overheating**.*

*While the **probability of a single error is small**, the total error amount in **all RAM connected to the Internet is significant**."*

https://media.blackhat.com/bh-us-11/Dinaburg/BH_US_11_Dinaburg_Bitsquatting_WP.pdf

Bitflip Causes

- Common assumption: hardware operates correctly

But:

- manufacturing defects
- operation outside spec
  - temperature, humidity, electric parameters → fault injection
- radiation / radiocontamination

- “some studies conclude that cosmic rays are actually the primary source of bit-errors in DRAM at ground level”
  - Some studies demonstrated altitude dependency

Bitflip Characteristics

- No reverse engineering or manipulation needed
- Largely independent of hardware and software architecture
- Can occur in various pieces of software
  - Application (e.g., browser)
  - Operating system library (e.g., DNS)
  - Middlebox (e.g., router)
  - Application server (e.g., HTML cache)
  - …
- Security: Confidentiality, ~~Integrity~~, Availability
  - most of them useless, but some pave the way for other attacks
- Mitigation: error detection / correction (such as ECC memory)

Warren's Experiment

- Registered a number of bitflip variants for common domains, recorded HTTP

  → Extracted hostnames, SNI names, cookies, … 🤔

- Tried to choose names which are "far" on the keyboard.
- Expanded names:
  - microsont.com ->
  - msedge.b.tlu.dl.delivery.mp.microsont.com
- API / "System" names:
  - access-point.cloudmessaging.edge.microsont.com
- SNI names:
  - tlu.dl.delivery.mp.microsoft.com

Check out Warren's talk (ICANN 83)

So far, so good.

RFC 9609: Priming Queries

2. Description of Priming

Priming is the act of finding the list of root servers from a configuration that lists some or all of the purported IP addresses of some or all of those root servers. In priming, a recursive resolver starts with no cached information about the root servers, and it finishes with a full list of their names and addresses in its cache. [...]

The configured list of IP addresses for the root servers usually comes from the vendor or distributor of the recursive server software. Although this list is generally accurate and complete at the time of distribution, it may become outdated over time.

The domain names for the root servers are called the "root server identifiers". Although this list has remained stable since 1997, the associated IPv4 and IPv6 addresses for these root server identifiers occasionally change. Research indicates that, following such changes, certain resolvers fail to update to the new addresses [...]

Therefore, it is important that resolvers are able to cope with change, even without relying upon configuration updates to be applied by their operator. **Root server identifier and address changes are the main reasons that resolvers need to use priming** to get a full and accurate list of root servers, instead of just using a statically configured list.

RFC 9609: Priming Queries

3. Priming Queries

A priming query is a DNS query whose response provides root server identifiers and addresses. It has a QNAME of ".", a QTYPE of NS, and a QCLASS of IN; it is sent to one of the addresses in the configuration for the recursive resolver. [...]

3.1. Repeating Priming Queries

The recursive resolver SHOULD send a priming query only when it is needed, such as **when the resolver starts with an empty cache or when the NS resource record set (RRset) for the root zone has expired**. [...]

6. Security Considerations

[...] An on-path attacker who sees a priming query coming from a resolver can inject false answers before a root server can give correct answers. If the attacker's answers are accepted, this can set up the ability to give further false answers for future queries to the resolver. False answers for root servers are more dangerous than, say, false answers for TLDs, because the root is the highest node of the DNS. [...]

Priming Queries: How do they look like?

- DNS is hierarchical: queries descend from the root zone (root domain: ".")

```
$ dig +noall +answer NS .
.           518400   IN   NS   a.root-servers.net.
.           518400   IN   NS   b.root-servers.net.
…
.           518400   IN   NS   m.root-servers.net.
```

- DNS resolvers use hardcoded nameservers + IP addresses for the root ("hints file": https://www.internic.net/domain/named.root)

```
.                        3600000      NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:ba3e::2:30
```

- But their addresses may change → "priming queries"

RFC 9609 again

3.3. DNSSEC with Priming Queries

> [...] An attacker controlling a rogue root name server also has complete control over all unsigned delegations and over the entire domain name space in the case of non-DNSSEC validating resolvers. [...]

The w00t-servers.net Experiment

Idea: Can we bitsquat on priming queries to hijack the root?

- Start by registering 47 (+1) bitflip variants of `root-servers.net`
  - There are 56 valid variants total: 36× bit cleared, 20× bit set – 8 domains already taken
  - Nameservers: `a.w00t-servers.net` / `b.w00t-servers.net`
- Dataset – thanks to Rutgers University for analysis resources
  - early July 2025 till mid-January 2026
  - 16 GB pcap files each recorded on `[ab].w00t-servers.net`
    … includes 8,302,445 A/AAAA queries for `[a-m].bitflip(root-servers).net`
  - 219 GB pcap files recorded on fake root server
  - Lots of crap, needs sanitization
- Step 1: Look at fake A/AAAA records for `[a-m].bitflip(root-servers).net`

Experimental Setup

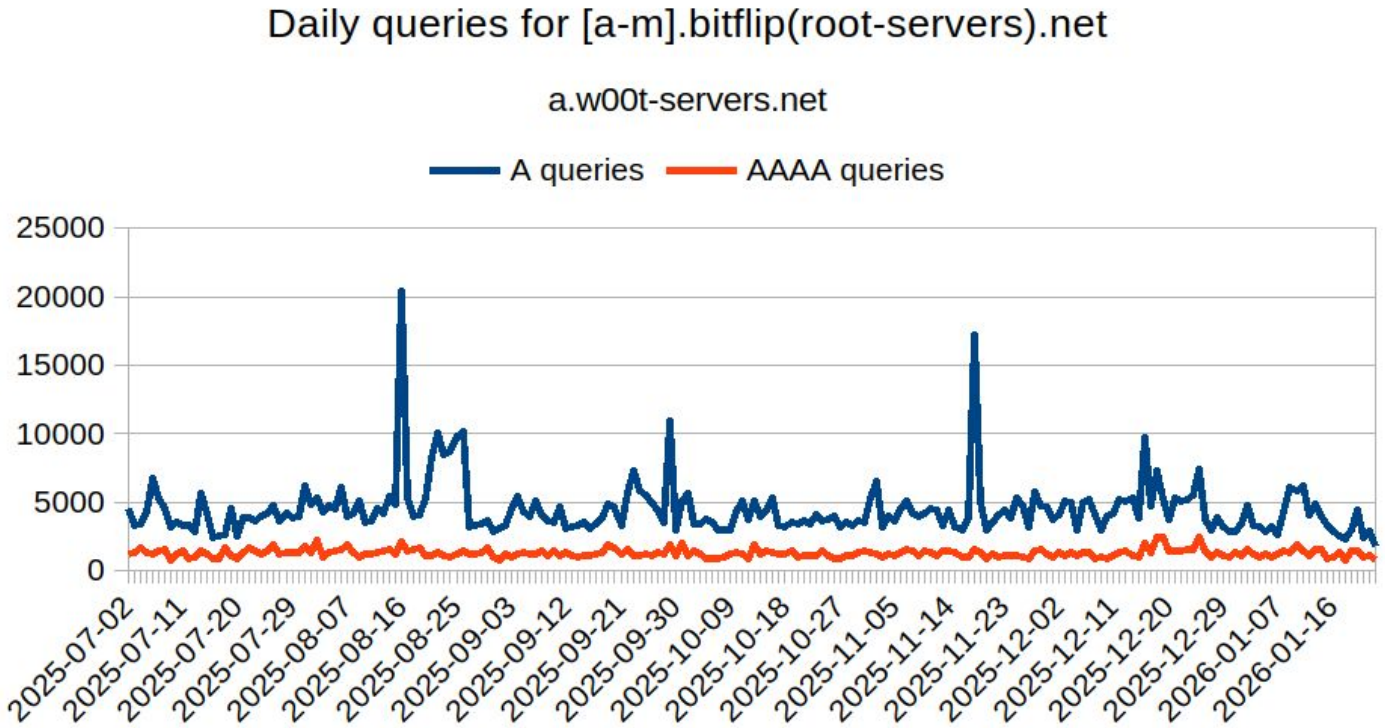
VM 1

```
$ host a.w00t-servers.net
a.w00t-servers.net has address 91.99.151.56
a.w00t-servers.net has IPv6 address 2a01:4f8:c012:2f9e::1
```

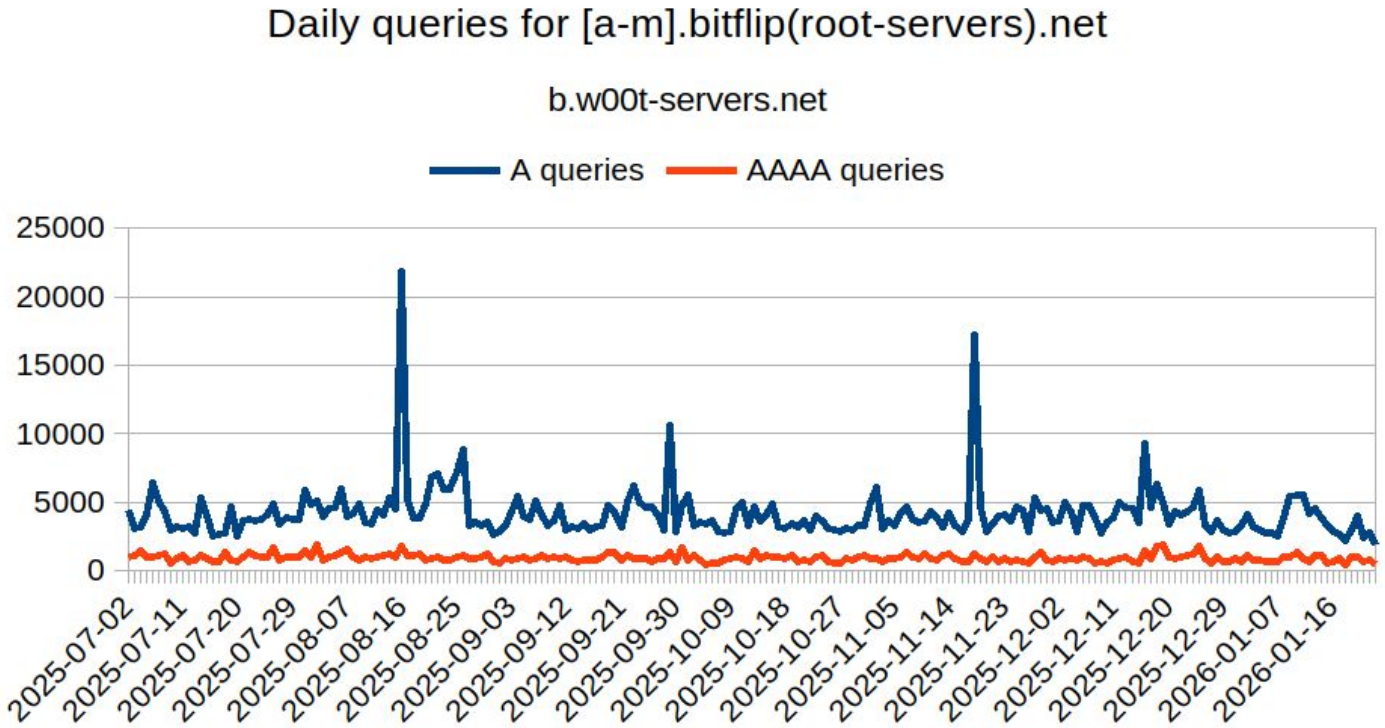
VM 2

```
$ host b.w00t-servers.net
b.w00t-servers.net has address 37.27.28.18
b.w00t-servers.net has IPv6 address 2a01:4f9:c013:c86a::1
```

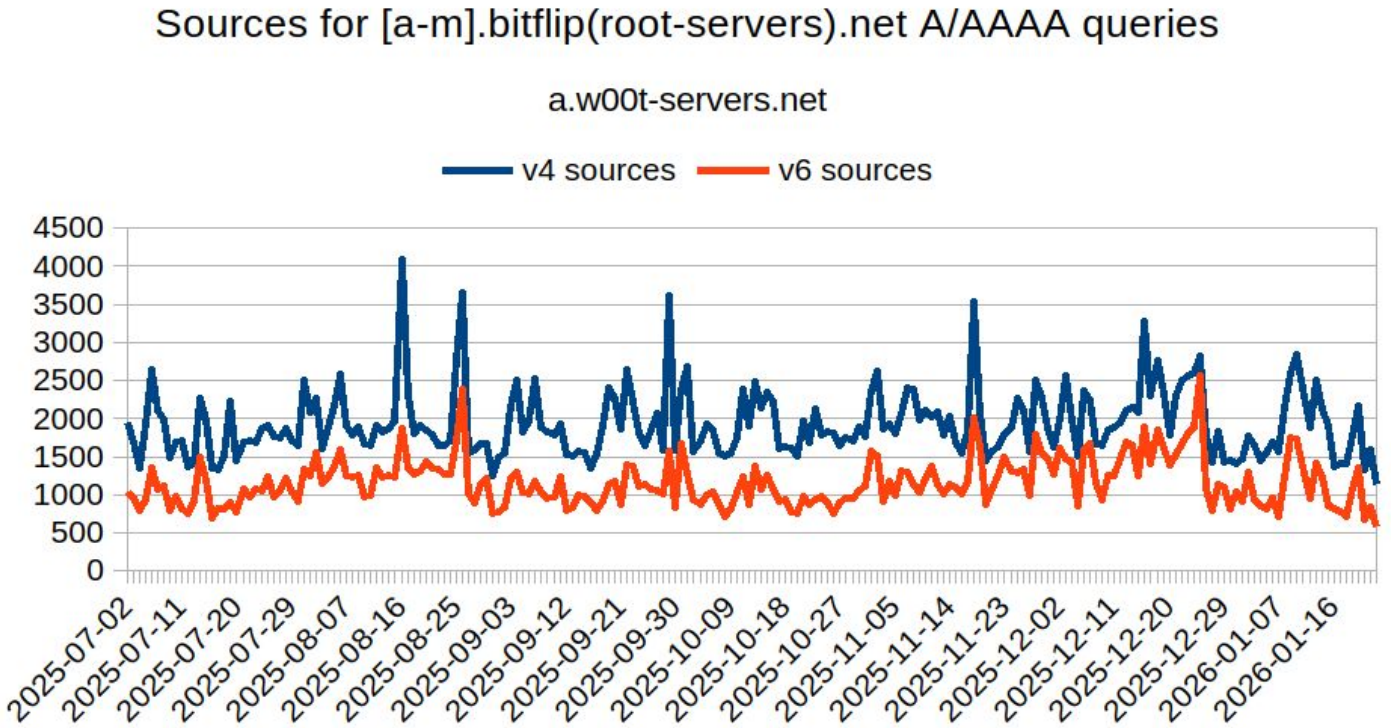
Queries recorded



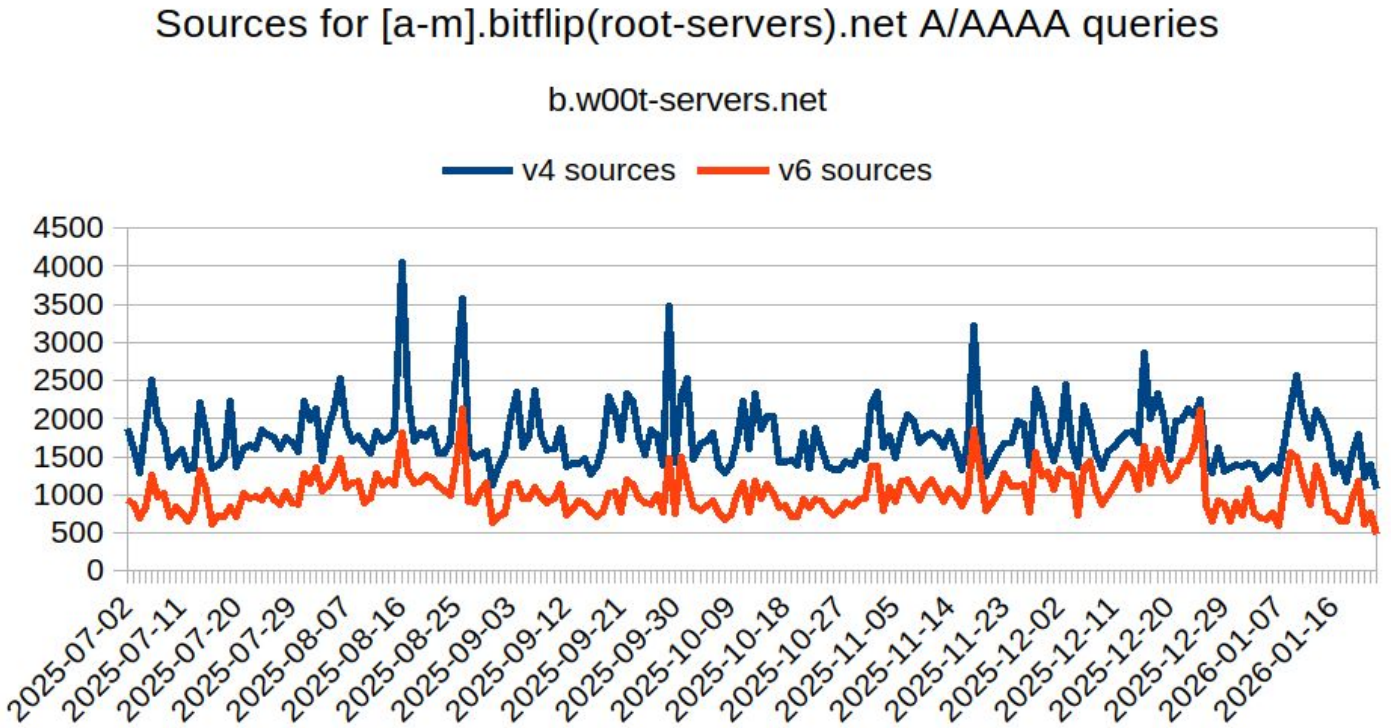
Queries recorded



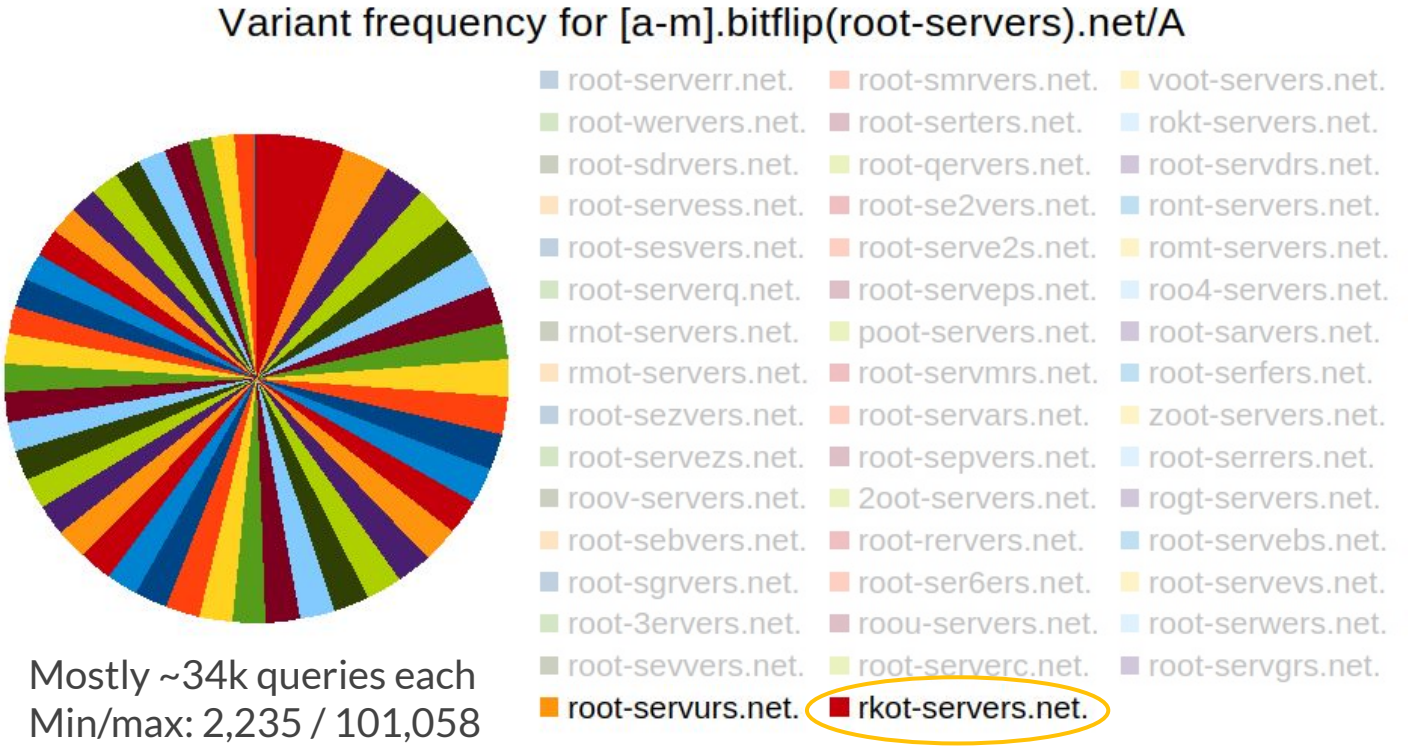
Number of Query Sources



Number of Query Sources



Frequency of bitflip variants (A queries)




```
01001111 O
01001011 K

01101111 o
01101011 k

01000101 E
01010101 U

01100101 e
01110101 u
```

- Mostly ~34k queries each
- Min/max: 2,235 / 101,058
- 3 times r**k**ot-ser**r**ers.net!

deSEC

Frequency of bitflip variants (AAAA queries)


```
01001111 O
01001011 K

01101111 o
01101011 k

01000101 E
01010101 U

01100101 e
01110101 u

01010010 R
01010011 S

01110010 r
01110011 s
```

- Mostly ~10k queries each
- Min/max: 1,279 / 18,614

deSEC

Prefix frequencies

a.w00t-servers.net			b.w00t-servers.net	
A queries	AAAA queries		A queries	AAAA queries
184,340	25,771	**a**	168,956	16,835
53,610	18,051	**b**	50,585	13,988
80,868	20,620	**c**	75,698	15,520
62,180	23,436	**d**	63,158	15,209
60,377	19,546	**e**	56,285	14,892
54,173	17,249	**f**	49,129	13,316
56,853	15,553	**g**	51,520	12,264
51,314	15,783	**h**	46,664	12,104
63,232	19,486	**i**	57,771	14,745
74,179	21,047	**j**	68,531	15,861
52,500	16,915	**k**	47,961	13,191
58,712	19,105	**l**	54,089	14,228
73,230	36,497	**m**	73,903	26,234

prefix	count	prefix	count
U	14	6	64
P	16	z	64
W	16	4	65
)	32	@	68
Q	38	8	83
R	41	n	94
}	42	q	150
Y	48	p	187
v	58	o	224
y	58	9	435
x	59	r	499
0	60	_	1,623
w	60	2	1,826
3	61	!	16,574
5	62	s	19,492
7	62	1	21,072
t	62	`	39,973
u	63	*	2,227,058

Step 2: Serve Real Root Zone on Bitflip Root Server

- Raw dataset has 145,620,242 queries
  - 75.3% of these are ANY queries: 38M for europa.eu, 37M for esc5.net, 4M for up.pt, 41k for .sl
  - 24.6% are TXT queries: 24M for cisco.com *(large one!)*, 400k each for atlassian.com, gwu.edu

  → looks like attempts of amplification attacks
- After pruning other invalid queries (e.g., CH class), only **59,364 queries remain**

  → Hard to tell what that means
- Experimental refinement may be needed

Some Interesting Queries from a European Mobile ISP

```
2025-07-11T05:25:48 XXX.YYY.58.2
                    geolocation.garmin.com.cdn.cloudflare.net. A

2025-07-26T14:32:49 XXX.YYY.63.19 3.cep.eu.prod.c3.volvocars.com. A

2025-09-09T23:58:03 XXX.YYY.55.57
                    eu-sest-1a.k8s-vehiclee-vehiclee-7e05e6b8cc-
                    b4f3ff8b40ff2dc4.elb.eu)west-1.amazonaws.com. A
2025-09-09T23:58:09 XXX.YYY.55.57
                    eu-west-1a.k8s-vehiclee-vehiclee-7e05e6b8cc-
                    b4f3ff8b40ff2dc4.elb.eu-west-1.amazonaws.com. A

2025-10-22T00:27:57 XXX.YYY.19.2 ns-1786.awsdns-31.co.uk. A
2025-10-22T00:27:59 XXX.YYY.19.2 ns-1786.awsdns-31.ck.uk. AAAA
```

Step 3: Improved Experimental Setup (captured Feb 5-27, 2026)

- Do not drop NS and DNSKEY queries for `[ab].w00t-servers.net`

Step 3: Improved Experimental Setup (captured Feb 5-27, 2026)

- Do not drop NS and DNSKEY queries for `[ab].w00t-servers.net`
- Add real root server name when asked for `[a-m].bitflip(root-servers).net`

```
$ dig A [a-m].root-se2vers.net
a.root-se2vers.net.  86400   IN   A     78.47.111.199          # p.w00t-servers.net
a.root-servers.net.  86400   IN   A     46.62.230.96           # o.w00t-servers.net
         ^

$ dig AAAA [a-m].root-se2vers.net
a.root-se2vers.net.  86400   IN   AAAA  2a01:4f8:1c1a:3ca0::1  # p.w00t-servers.net
a.root-servers.net.  86400   IN   AAAA  2a01:4f9:c012:5daa::1  # o.w00t-servers.net
         ^
```

Different IP addresses (two additional VMs) → distinguish corruption modes

Step 3: Improved Experimental Setup (captured Feb 5-27, 2026)

- Priming queries were repeatedly seen from various clients: `. /NS`
  - Used by resolvers to refresh set of root NS hostnames and glue
- How about responding with a tailored priming response?

```
$ dig NS . @[fake root server]

;; ANSWER SECTION:
.                       86400   IN   NS     o.w00t-servers.net.
.                       86400   IN   NS     p.w00t-servers.net.

;; ADDITIONAL SECTION:
o.w00t-servers.net.      3600   IN   A      95.217.243.58
p.w00t-servers.net.      3600   IN   A      49.13.33.167
o.w00t-servers.net.      3600   IN   AAAA   2a01:4f9:c012:5daa::14
p.w00t-servers.net.      3600   IN   AAAA   2a01:4f8:1c1a:3ca0::14
```

→ Might cause resolvers to latch on to our hostnames/glue (different IPs!)

Some Interesting Queries after Improvements

Ford Transportation Mobility Cloud

```
2026-02-17T23:00:47.764043 XXX.YYY.121.254 78.47.111.199 . NS
2026-02-17T23:01:48.204300 XXX.YYY.121.254 78.47.111.199 . NS
2026-02-17T23:18:36.241081 XXX.YYY.121.254 95.217.243.58 mq.autonomic.ai. A
2026-02-17T23:18:37.023560 XXX.YYY.121.254 49.13.33.167  mq.autonomic.ai. A
2026-02-17T23:18:37.841542 XXX.YYY.121.254 95.217.243.58 mq.autonomic.ai. A
2026-02-17T23:18:38.630916 XXX.YYY.121.254 49.13.33.167  mq.autonomic.ai. A
2026-02-17T23:18:39.603309 XXX.YYY.121.254 95.217.243.58 mq.autonomic.ai. A
2026-02-17T23:18:41.198863 XXX.YYY.121.254 49.13.33.167  mq.autonomic.ai. A
2026-02-17T23:18:42.804587 XXX.YYY.121.254 95.217.243.58 mq.autonomic.ai. A
```

Some Interesting Queries from a US Mobile ISP

Volvo / G-Book

```
2026-02-21T09:42:49.829276 XXX.YYY.113.108 78.47.111.199 . NS
2026-02-21T09:42:50.679099 XXX.YYY.113.108 78.47.111.199 . NS
2026-02-21T09:42:50.939392 XXX.YYY.113.108 78.47.111.199 cc-sec-t-svcs.cctsl.aom. A
2026-02-21T09:42:51.375212 XXX.YYY.113.108 78.47.111.199 cc-sec-t-svcs.cctsl.com. A
```

Note: `cc-sec-t-svcs.cctsl.com` is a CNAME for `d3g9vvpbff5etf.cloudfront.net` → queries follow

```
2026-02-21T09:42:57.410949 XXX.YYY.113.108 49.13.33.167 tscgt17cy03.g-book.cmm. A
2026-02-21T09:42:57.796686 XXX.YYY.113.108 49.13.33.167 tscgt17cy03.g-book.com. A
2026-02-23T13:31:07.834561 XXX.YYY.64.126  78.47.111.199
                           3.cep.us.prod.c3.voltocars.com. A
2026-02-26T20:10:03.842369 XXX.YYY.18.208  78.47.111.199 udns2.cscdns.uk. AAAA
2026-02-26T20:10:04.224047 XXX.YYY.18.208  78.47.111.199 udns2.cscdns.uk. A
```

`udns2.cscdns.uk` is NS of `volvocars.com` with NS `dns1.cscdns.net` which has `udnq2.cscudns.org`

Consequently, `gtld-servers.net` questions are seen as well

```
2026-03-03T16:14:09.821471 XXX.YYY.113.142 78.47.111.199 udnq2.cscudns.org. AAAA
2026-03-03T16:14:09.821499 XXX.YYY.113.142 78.47.111.199 udnq2.cscudns.org. A
```

Some Interesting Queries from a US Mobile ISP

Volvo / G-Book

```
2026-02-21T09:42:49.829276 XXX.YYY.113.108 78.47.111.199 . NS
2026-02-21T09:42:50.679099 XXX.YYY.113.108 78.47.111.199 . NS
2026-02-21T09:42:50.939392 XXX.YYY.113.108 78.47.111.199 cc-sec-t-svcs.cctsl.aom. A
2026-02-21T09:42:51.375212 XXX.YYY.113.108 78.47.111.199 cc-sec-t-svcs.cctsl.com. A
```

Note: `cc-sec-t-svcs.cctsl.com` is a CNAME for `d3g9vvpbff5etf.cloudfront.net` → queries follow

```
2026-02-21T09:42:57.410949 XXX.YYY.113.108 49.13.33.167 tscgt17cy03.g-book.cmm. A
2026-02-21T09:42:57.796686 XXX.YYY.113.108 49.13.33.167 tscgt17cy03.g-book.com. A
2026-02-23T13:31:07.834561 XXX.YYY.64.126  78.47.111.199
                           3.cep.us.prod.c3.voltocars.com. A
2026-02-26T20:10:03.842369 XXX.YYY.18.208  78.47.111.199 udns2.cscdns.uk. AAAA
2026-02-26T20:10:04.224047 XXX.YYY.18.208  78.47.111.199 udns2.cscdns.uk. A
```

`udns2.cscdns.uk` is NS of `volvocars.com` with NS `dns1.cscdns.net` which has `udnq2.cscudns.org`

Consequently, `gtld-servers.net` questions are seen as well

```
2026-03-03T16:14:09.821471 XXX.YYY.113.142 78.47.111.199 udnq2.cscudns.org. AAAA
2026-03-03T16:14:09.821499 XXX.YYY.113.142 78.47.111.199 udnq2.cscudns.org. A
```

Some Interesting Queries from a US Mobile ISP

Volvo / G-Book

```
2026-02-21T09:42:49.829276 XXX.YYY.113.108 78.47.111.199 . NS
2026-02-21T09:42:50.679099 XXX.YYY.113.108 78.47.111.199 . NS
2026-02-21T09:42:50.939392 XXX.YYY.113.108 78.47.111.199 cc-sec-t-svcs.cctsl.aom. A
2026-02-21T09:42:51.375212 XXX.YYY.113.108 78.47.111.199 cc-sec-t-svcs.cctsl.com. A
```

Note: `cc-sec-t-svcs.cctsl.com` is a CNAME for `d3g9vvpbff5etf.cloudfront.net` → queries follow

```
2026-02-21T09:42:57.410949 XXX.YYY.113.108 49.13.33.167 tscgt17cy03.g-book.cmm. A
2026-02-21T09:42:57.796686 XXX.YYY.113.108 49.13.33.167 tscgt17cy03.g-book.com. A
2026-02-23T13:31:07.834561 XXX.YYY.64.126  78.47.111.199
                           3.cep.us.prod.c3.voltocars.com. A
2026-02-26T20:10:03.842369 XXX.YYY.18.208  78.47.111.199 udns2.cscdns.uk. AAAA
2026-02-26T20:10:04.224047 XXX.YYY.18.208  78.47.111.199 udns2.cscdns.uk. A
```

`udns2.cscdns.uk` is NS of `volvocars.com` with NS `dns1.cscdns.net` which has `udnq2.cscudns.org`
Consequently, `gtld-servers.net` questions are seen as well

```
2026-03-03T16:14:09.821471 XXX.YYY.113.142 78.47.111.199 udnq2.cscudns.org. AAAA
2026-03-03T16:14:09.821499 XXX.YYY.113.142 78.47.111.199 udnq2.cscudns.org. A
```

Summary

- ~10M queries for `[a-m].bitflip(root-servers).net` seen
- ~100k queries seen on fake root servers after pruning (150M total)
- ~10 hijackable resolution cascades observed with improved setup (3 weeks)
  - Did not fake referrals – would likely have caused higher number
- Analysis could be refined by looking into …
  - queries by country
  - origin AS
  - whether sources are resolvers (so far none noticed)
  - DO bit for priming responses (requests DNSSEC)
- Worth pursuing?

deSEC

Peter Thomassen
peter@desec.io