清華大學
Tsinghua University

[NDSS 26'] Should I Trust You? Rethinking the Principle of Zone-Based Isolation DNS Bailiwick Checking

Yuxiao Wu[1], Yunyi Zhang[2], Chaoyi Lu[3], Baojun Liu[2]

[1]Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University

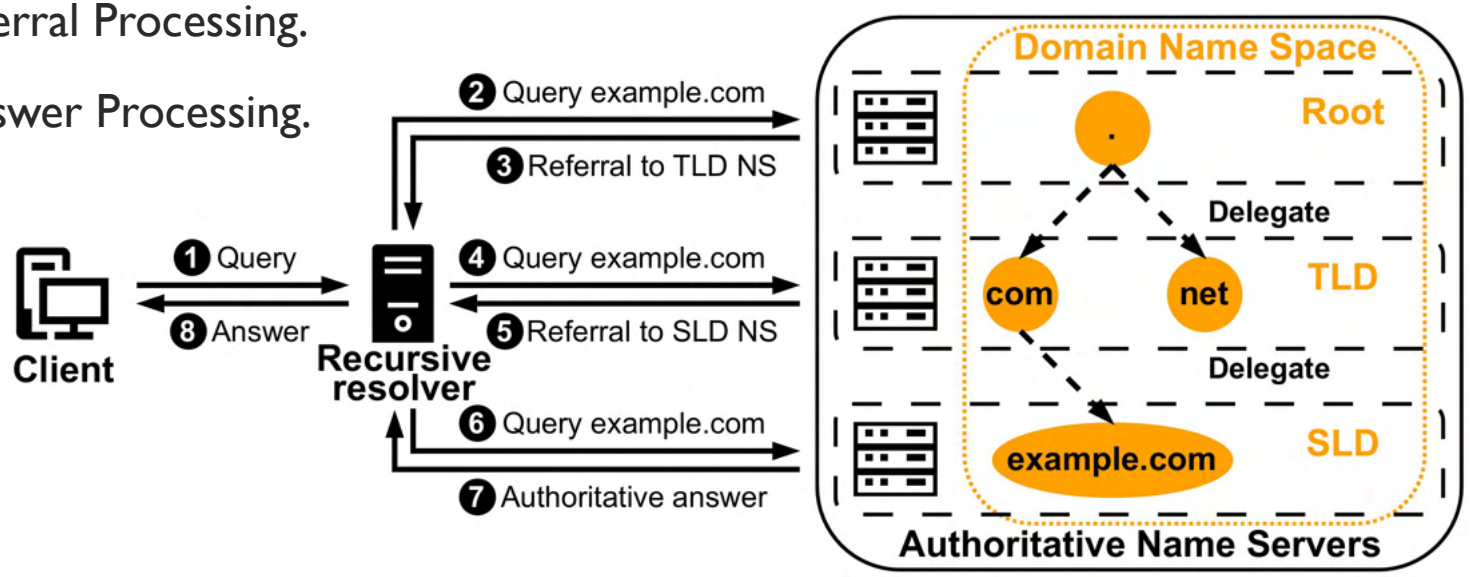
[2]Tsinghua University

[3]Zhongguancun Laboratory

February 2026

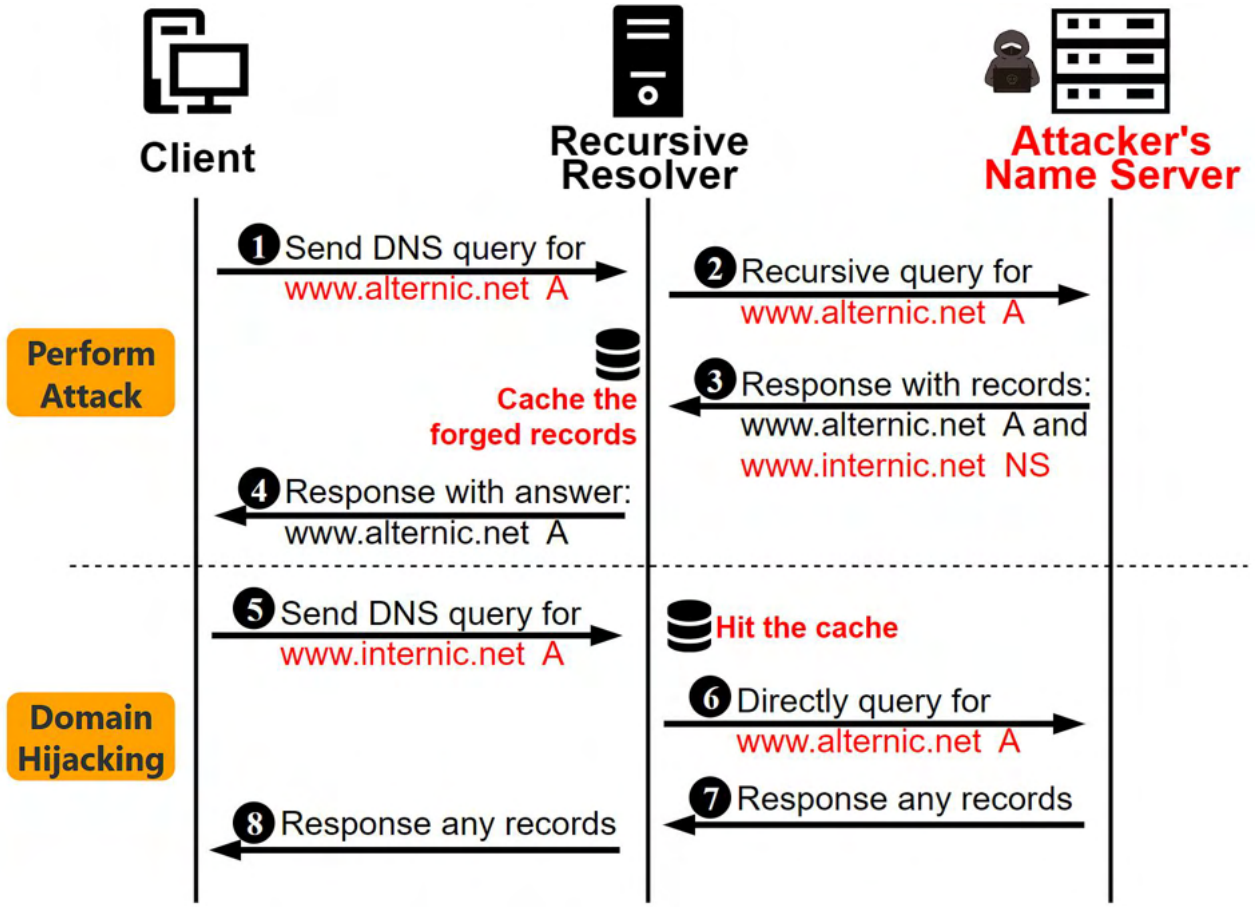
What is DNS?

- The Domain Name System:
  - Transform "example.com" to 23.215.0.138.
- Three Steps to get answer for resolvers:
  - Step 1. Query Receiving.
  - Step 2~5. Referral Processing.
  - Step 6~8. Answer Processing.



What is DNS Poisoning Attack?

- Kashpureff Attack (1997):
  - **Method: Directly forge resource records** for other domains in response;
  - **Result:** All subsequent queries will be **redirected to the attacker's server**.
  - **Mitigation:** Major DNS software have proposed and implemented the **Bailiwick principle**.



What is DNS Bailiwick Checking?

GOAL: Prevent **malicious users** from carrying **records of other users' domain** in response.

Spoofing Answer Section

Add **any type** of records for other domains.

Header Flags: QR AA
Question Section: attacker.com. A
Answer Section: **victim1.com. A a.t.k.r**
Authority Section: (Empty)
Additional Section: (Empty)

(c) Spoofing Answer Section

Spoofing Authority Section

Add **NS** records for other domains.

Header Flags: QR AA
Question Section: attacker.com. A
Answer Section: attacker.com. A a.b.c.d
Authority Section: **victim2.com. NS ns.atkr.com.**
Additional Section: (Empty)

(d) Spoofing Authority Section

Spoofing Additional Section

Add **glue** records for target name server.

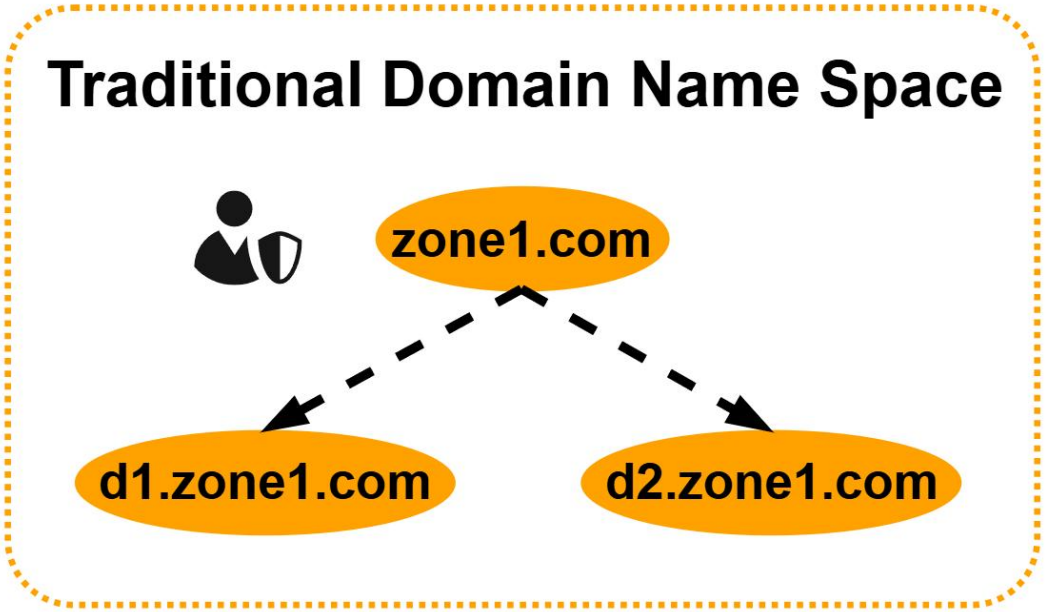
Header Flags: QR AA
Question Section: attacker.com. A
Answer Section: attacker.com. A a.b.c.d
Authority Section: com. NS a.gtld-servers.net.
Additional Section: **a.gtld-servers.net. A a.t.k.r**

(e) Spoofing Additional Section

Question

Traditionally, DNS zone was controlled by **a single administrator**.

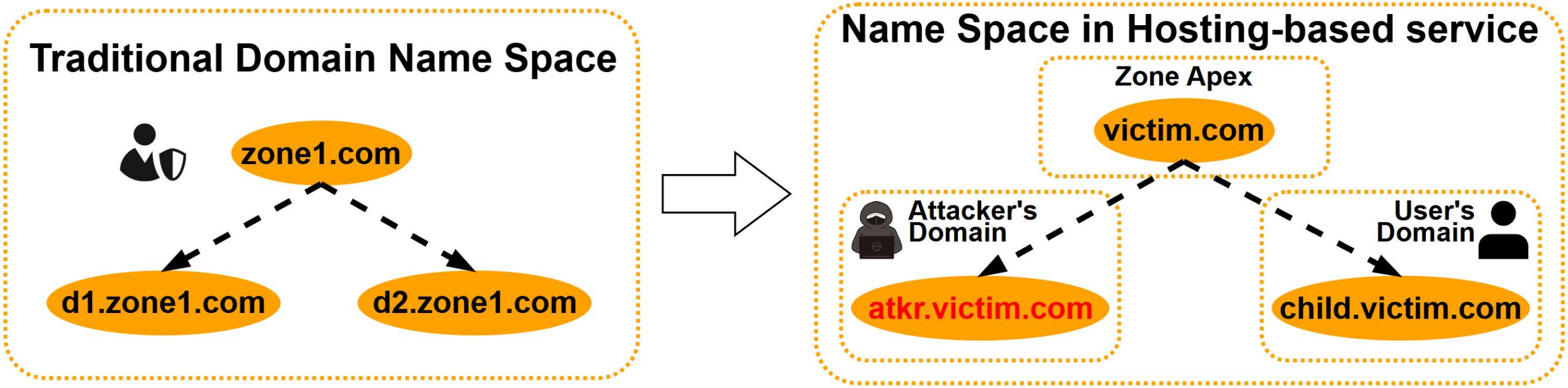
Zone-based Bailiwick checks have performed well.



Question

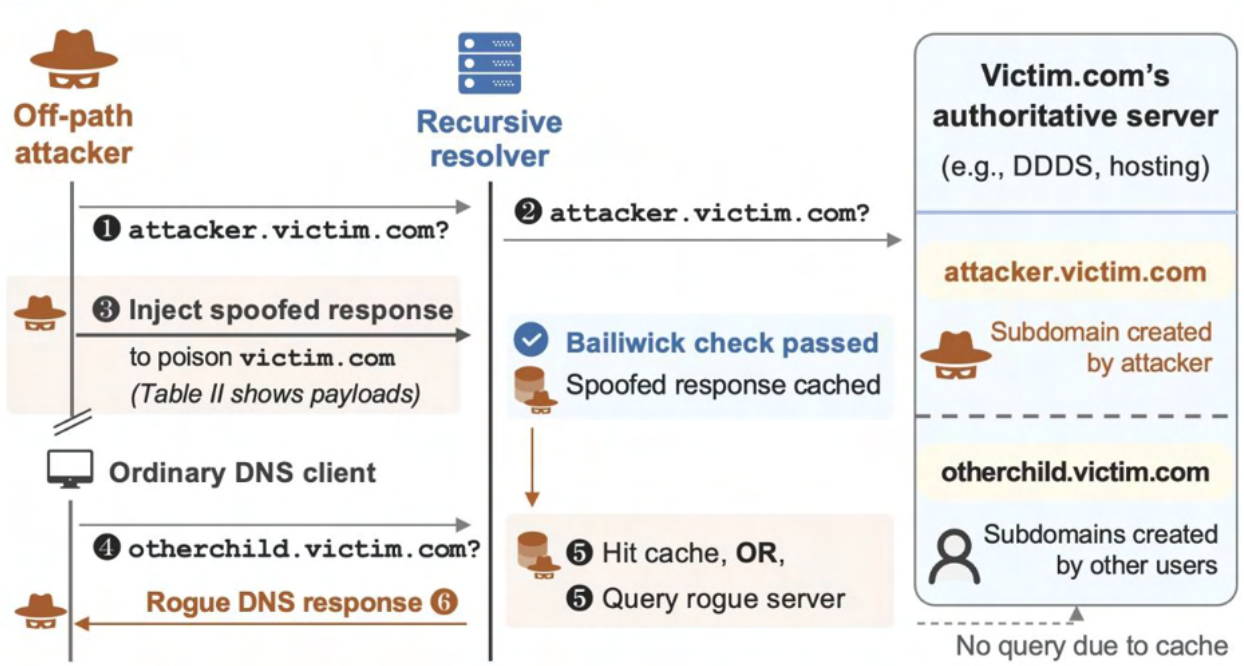
However, third-party hosting services, such as DDNS, result in **domains** within the same zone being **controlled by different users**.

Is bailiwick still effective?



Threat Model & Attack Workflow

- **Attacker Ability:**
  - Perform **Port-inference attacks.**
  - **Control subdomains within a zone** and configure records to generate **IP fragments**.
- **Attack Workflow:**
  - **Query the target resolver for the controlled domain**,
  - Use an off-path attack to **inject a spoofed response** into the resolver.
- **Result:**
  - Poison the caches of ***QNAME***'s **sibling** or even **parent** domains.



Attack Workflow of *Cuckoo Domain*

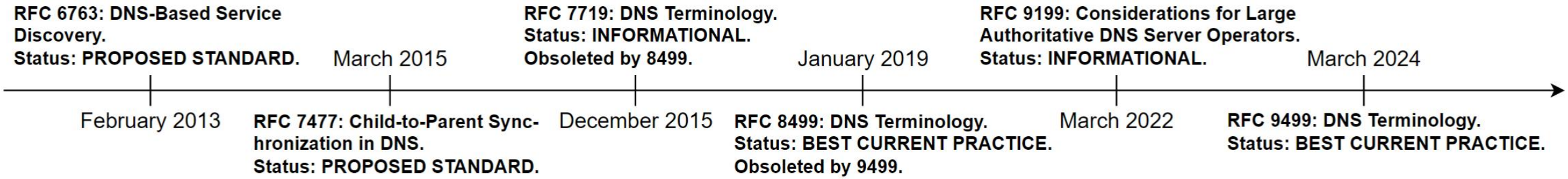
The Bailiwick in RFC

Imposing Requirement Constraints

- RFC 6763 requires recursive resolver to perform bailiwick checks;
- RFC 7477 and 9199 use “in-bailiwick” and “out-of-bailiwick” NS records to validate the legitimacy of glue records.

Providing Definitions

- RFC 7719 uses “in-bailiwick” to describe the name server located in a zone;
- RFC 8499 further subdivides in-bailiwick into “in-domains” and “sibling domains” based on RFC 7719, aligning with the definition of glue records;
- RFC 9499 notes that the definition of “bailiwick” has been observed to cause more confusion than clarity for this usage, and obsolete RFC 8499.



The Bailiwick Checking in DNS Software

- **Stage Ⅰ. Query Initialization:**
  - Get ***QNAME*** and ***QTYPE*** from client's query.
  - If the cache exists, ***QZONE*** is initialized to the domain closest to ***QNAME***.
  - Otherwise, ***QZONE*** is initialized to ".".

- Two key questions for next:
  - **Check** the record, or simply **discard** it in a specific section?
  - **Which information** will be used to perform **bailiwick checking** for each record?

The Bailiwick Checking in DNS Software

- **Stage Ⅱ. Referral Processing:**
  - **All resolvers check** records from the ***authority*** and ***additional*** sections.
  - **4 resolvers** only require ***RNAME < QZONE***, ignoring ***QNAME***:
    - querying the subdomain may overwrite the **sibling domain's** cache.
  - **2 resolvers** consider ***RNAME = QZONE*** to be legal:
    - querying the subdomain may overwrite the **parent domain's** cache.

Functional implementation	BIND9[12]	Knot[56]	Unbound[101]	PowerDNS[80]	Technitium[99]	MaraDNS[70]	Microsoft DNS[98]	Simple DNS Plus[79]
Version	9.18.32	6.0.9	1.22.0	4.9.9	13.3	3.5.0036	2025	9.1.116
Check NS from a referral response	✓	✓	✓	✓	✓	✓	✓	✓
Check AR from a referral response	✓	✓	✓	✓	✓	✓	✓	✓
Cache sibling record in referral response	✗	✗	✗	✓	✓	✗	✓	✓
Update delegation in referral response	✗	✓	✗	✓	✗	✗	✗	✗
Cache updates with the same level data	✓	✗	✓	✓	✓	✓	✓	✓

The Bailiwick Checking in DNS Software

- **Stage Ⅲ. Answer Sanitization:**
  - **6 resolvers check** and try to use records from the ***authority*** and ***additional*** sections.
  - **2 resolvers** perform bailiwick checks in the ***answer*** section using ***RNAME < QZONE***, while the others strictly require ***RNAME = QNAME***.
  - **4 resolvers** perform bailiwick checks in the ***authority*** and ***additional*** sections using ***RNAME ≤ QZONE***.
  - **2 resolvers** perform bailiwick checks in the ***authority*** and ***additional*** sections using ***QNAME ≤ RNAME ≤ QZONE*** and ***RNAME < QZONE*** respectively.

Functional implementation	BIND9[12]	Knot[56]	Unbound[101]	PowerDNS[80]	Technitium[99]	MaraDNS[70]	Microsoft DNS[98]	Simple DNS Plus[79]
Version	9.18.32	6.0.9	1.22.0	4.9.9	13.3	3.5.0036	2025	9.1.116
Check matched RTYPE records in AN	✓	✓	✓	✓	✓	✓	✓	✓
Check NS from an answer response	✓	✓	✓	✓	✗	✗	✓	✓
Check AR from an answer response	✓	✓	✓	✓	✗	✗	✓	✓
Cache unmatched RNAME in AN	✗	✗	✗	✓	✗	✗	✓	✗
Cache sibling record in answer response	✓	✓	✗	✓	-	-	✓	✓
Update delegation in answer response	✓	✓	✓	✓	-	-	✓	✗
Cache updates with the same level data	✓	✗	✓	✓	✓	✓	✓	✓

Test Payload Design

- **Goal of the attacker**:
  - **T1: Arbitrary Record Injection**.
    - Inject forged records for sibling or parent domains with ***matched RTYPE*** in ***answer*** section.
  - **T2: Authority Record Takeover**.
    - Using fake **CNAME** or **NS** record to hijacking the target domain.
  - **T3: Glue Record Poisoning**.
    - Using fake **glue records** in ***additional*** section to indirectly hijack the domain.

Vulnerable	BIND9[12]	Knot[56]	Unbound[101]	PowerDNS[80]	Technitium[99]	MaraDNS[70]	Microsoft DNS[98]	Simple DNS Plus[79]
Version	9.18.32	6.0.9	1.22.0	4.9.9	13.3	3.5.0036	2025	9.1.116
T1 Attack	✗	✗	✗	✓	✗	✗	✓	✗
T2 Attack	✓	✓	✓	✓	✓	✗	✓	✓
T3 Attack	✓	✓	✓	✓	✗	✗	✓	✓

Evaluating DNS Resolvers in the Wild

- Open Resolvers:
  - We use XMAP to conduct a scan of IPv4 address, and collect **588K stable open resolvers**.
- Public DNS Providers:
  - We collect a list of 30 popular public DNS vendors, and **21 of them have abnormal behavior.**

Attack Payload	# IP	%	Attack Payload	# IP	%
DNS resolver on Jun. 1 2025	1,044,825	-	DNS resolver alive on Jun. 7 2025	588,624	100%
1. AN-a	28,611	4.86%	2. AN-a-c	26,216	4.45%
3. AN-txt	28513	4.84%	4. AN-txt-c	28,545	4.85%
5. AN-cname	28,697	4.88%	6. AN-cname-c	26,061	4.43%
7. AN-cname2	28,607	4.86%	8. AN-cname2-c	26,239	4.46%
9. NS-in-domain	179,175	30.44%	10. NS-in-domain-c	106,049	18.02%
11. NS-in-domain -referral	46,776	7.95%	12. NS-in-domain -referral-c	39,224	6.66%
13. NS-out-zone	105,837	17.98%	14. NS-out-zone-c	99,886	16.97%
15. NS-out-zone -referral	5,850	0.99%	16. NS-out-zone -referral-c	24,636	4.19%
17. AR-victim-domain	89,591	15.22%	18. AR-attack-domain	46,309	7.87%
19. AR-victim-domain -referral	76,121	12.93%	20. AR-victim-attack -referral	54,961	9.34%
T1 Attack	**41,104**	**6.98%**	**T2 Attack**	**235,171**	**39.95%**
T3 Attack	**138,856**	**23.59%**	**Total Vulnerability**	**262,779**	**44.64%**

Public DNS Vendors	IPv4 Address	Vulnerable?		
		T1	T2	T3
CNNIC sDNS [90]	1.2.4.8	✗	✓	✓
Quad9 DNS [82]	9.9.9.9	✗	✓	✓
Strongarm DNS [96]	52.3.100.184	✓	✓	✓
Hurricane Electric DNS [31]	74.82.42.42	✓	✓	✓
ControlD DNS [22]	76.76.2.0	✓	✓	✓
LibreDNS [67]	88.198.92.222	✓	✓	✓
Safe Surfer DNS [86]	104.155.237.225	✓	✓	✗
OneDNS [77]	117.50.10.10	✗	✓	✓
Clean Browsing DNS [18]	185.228.168.10	✓	✓	✓
Dyn DNS [28]	216.146.35.35	✗	✓	✓

Third-party hosting services

- Dynamic DNS (DDNS)
  - Allows users to update their DNS records dynamically.
- Free subdomain service
  - Provide free subdomains within the zone to users for various purposes, such as personal websites and custom emails.
- Load balancing service
  - Offer GTM or CDN services.
- Others
  - Some IoT vendors and large companies assign different subdomains to different internal users or devices.

Vendor	Available Domain	Name Server	Number of Subdomains	Daily Queries
No-IP[1] [76]	∗.ddns.net ∗.zapto.org ∗.hopto.org ∗.sytes.net ∗.ddns.me	nf1.no-ip.com nf2.no-ip.com nf3.no-ip.com nf4.no-ip.com	862,426 310,049 305,116 300,657 274,537	14,986,250.7 4,402,439.6 4,197,433.4 1,337,583.6 255,232.4
Dynv6 [30]	∗.dynv6.net ∗.dns.army ∗.v6.army ∗.dns.navy ∗.v6.rocks ∗.v6.navy	ns1.dynv6.com ns2.dynv6.com ns3.dynv6.com ns2.dynv6.net[2] ns3.dynv6.net[2]	31,337 9,058 3,390 3,364 3,111 2,140	2,749,345.5 3,114,435.7 693,529.5 734,668.9 428,565.6 542,523.8
DNSExit [27]	∗.linkpc.net ∗.publicvm.com ∗.work.gd ∗.run.place	ns10.dnsexit.com ns11.dnsexit.com ns12.dnsexit.com ns13.dnsexit.com	9,339 7,838 7,825 2,776	4,548,094.2 162,293.4 199,321.1 14,355.8
ClouDNS [20]	∗.ip-ddns.com ∗.ddns-ip.net	ns61.cloudns.net ns62.cloudns.com ns63.cloudns.net ns64.cloudns.uk	17,684 5,202	149,697.8 74,109.3
Akamai [6]	∗.akadns.net	a1-128.akadns.net[4] a18-128.akagtm.org	124,354	3.282×10^9
Synology[5] [97]	∗.myds.me ∗.synology.me ∗.i234.me ∗.dsmynas.com ∗.dscloud.biz	ddns-ns1.quickconnect.to ddns-ns2.quickconnect.to ddns-ns3.quickconnect.to ddns-ns4.quickconnect.to	1,481,000 655,159 32,338 10,386 7,116	8,925,267.2 15,916,461.5 5,028,393.2 1,095,862.7 2,860,361.9
ASUSTOR [9]	∗.myasustor.com	ns1.myasustor.com ns2.myasustor.com	11,681	14,576.2
Total Vul.[6]	-	-	**6,400,327**	**3.3×10^9**

Mitigation

- For DNS software vendors:
  - Validating the ***RNAME*** against the ***QNAME*** when caching records.
    - **BIND9** and **Technitium** adopted
  - Not using the name server information in the ***authority*** and ***additional*** sections of the answer response.
    - **PowerDNS** and **Unbound** adopted

- Collected responses during querying the Tranco Top 100K domains:
  - Only a few domains exhibiting abnormal resolution behavior.
  - Mitigations do not impact normal resolution processes.

Lessons Learned

- For the design of the Principle:
  - **Continuous effort is required to safeguard the security** of the DNS, and even the whole world.
  - Bailiwcik checking serves as a cornerstone of DNS security, and introduced in 1990s.
  - We must re-examine whether traditional implementations can effectively defend against new threats.
- For the software implementation:
  - Greater effort is required to **bridge the gap between theory and practical implementation**.
  - BIND9 wishes to cache records for subdomains of **QNAME**, but it actually caches records for **QZONE**.
  - Knot may have realized our attack and wanted to avoid T2 attack. However, it remains vulnerable.

Conclusion

- **Systematic analysis of bailiwick principle**
  - A significant divide between the protocol standards and their implementation in practice, by reviewing 470 RFCs and auditing 8 major DNS software.
- **New threat model**
  - A novel threat model that allows attackers to poison the cache of their sibling or even parent domains in the same zone by controlling any subdomain within the zone.
- **Comprehensive evaluation of new attacks**
  - A comprehensive threat assessment across major software implementations, third-party service providers and open resolvers.

清華大學
Tsinghua University

Thanks for listening!
Q & A

Yuxiao Wu[1], Yunyi Zhang[2], Chaoyi Lu[3], Baojun Liu[2]

[1]Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University

[2]Tsinghua University

[3]Zhongguancun Laboratory

February 2026