Latest updates: https://dl.acm.org/doi/10.1145/3719027.3744832

RESEARCH-ARTICLE

RebirthDay Attack: Reviving DNS Cache Poisoning with the Birthday Paradox

XIANG LI, Nankai University, Tianjin, China

MINGMING ZHANG, Beijing Institute of Technology, Beijing, China

ZUYAO XU, Nankai University, Tianjin, China

FASHENG MIAO, Tsinghua University, Beijing, China

YUQI QIU, Nankai University, Tianjin, China

BAOJUN LIU, Tsinghua University, Beijing, China

View all

Open Access Support provided by:

Nankai University

Tsinghua University

Beijing Institute of Technology

RebirthDay Attack:
Reviving DNS Cache Poisoning with the Birthday Paradox

Xiang Li*
Nankai University
Tianjin, China
lixiang@nankai.edu.cn

Mingming Zhang
Zhongguancun Laboratory
Beijing, China
zhangmm@mail.zgclab.edu.cn

Zuyao Xu
Nankai University
Tianjin, China
tochusc@gmail.com

Fasheng Miao
Tsinghua University
Beijing, China
mfs24@mails.tsinghua.edu.cn

Yuqi Qiu
Nankai Univeristy
Tianjin, China
qiuyuqi22@mails.ucas.ac.cn

Baojun Liu
Tsinghua University
Beijing, China
lbj@tsinghua.edu.cn

Jia Zhang
Tsinghua University
Beijing, China
zhangjia2017@tsinghua.edu.cn

Xiaofeng Zheng[†]
Tsinghua University
Beijing, China
zxf19@mails.tsinghua.edu.cn

Haixin Duan[‡]
Tsinghua University
Beijing, China
duanhx@tsinghua.edu.cn

Zheli Liu
Nankai Univeristy
Tianjin, China
liuzheli@nankai.edu.cn

Yunhai Zhang
NSFOCUS Technologies Group
Beijing, China
zhangyunhai@nsfocus.com

Dunqiu Fan
NSFOCUS Technologies Group
Beijing, China
fandunqiu@nsfocus.com

Abstract

DNS cache poisoning is a persistent game of attack and defense, posing an enduring challenge for the DNS community. Significant efforts have been made to uncover, detect, and mitigate vulnerabilities that increase the risk of cache poisoning. However, no work has systematically revisited whether the original cache poisoning attack based on the Birthday Paradox remains effective. In this work, we introduce RebirthDay, a novel DNS cache poisoning attack targeting recursive resolvers and forwarders, reviving the classic DNS Birthday attack that no longer works since 2002. RebirthDay exploits newly uncovered, protocol-compliant vulnerabilities in DNS extension implementations to bypass the query aggregation mechanism intended to prevent DNS Birthday attacks that has not been well understood. We uncovered that 18 out of 22 mainstream DNS software are vulnerable due to weaknesses in the processing of a DNS extension (i.e., ECS option), specifically lacking or incorrectly implemented ECS coherence checks when handling DNS queries and responses, demonstrating the widespread susceptibility to RebirthDay. These flaws could be exploited to circumvent the query aggregation mechanism and launch RebirthDay attacks. Through comprehensive evaluation, we showed that RebirthDay attacks are highly practical and can have significant real-world impact, affecting 16 router vendors, 14 public DNS services, and 365K (15%) open DNS resolvers. We have reported the identified vulnerabilities to affected vendors and discussed mitigation solutions with them. To date, we have received acknowledgments from 8 vendors, including BIND, Unbound, PowerDNS, and Quad9, and have been assigned 50 CVE-ids. Our study emphasizes the need for greater attention to the importance of ECS verification and DNS extension implementations, revealing new security risks introduced by them.

CCS Concepts

• **Networks → Naming and addressing**; • **Security and privacy → Network security**; **Authentication**.

Keywords

Domain name system; DNS security; Cache poisoning attack

*Corresponding author.

[†]Also with QI-ANXIN Technology Research Institute.

[‡]Also with Quan Cheng Laboratory.

1 Introduction

The Domain Name System (DNS) provides translation between human-readable domain names and machine-readable IP addresses.

It serves as a fundamental infrastructure for locating Internet services through the domain name space and DNS resolution. The resolution process is simple and robust in design but vulnerable to manipulation due to unencrypted communication. Extensive incidents have shown that DNS cache poisoning attacks remain a significant threat to the Internet. These attacks exploit vulnerabilities in the resolution process to inject rogue resource records into the DNS resolver's cache and redirect user access to malicious targets, leading to catastrophic security failures for national ISPs [62], public key infrastructure [6], and other critical services [13].

Over the past 30 years, DNS cache poisoning attacks have evolved from basic brute-force guessing of source ports and TxIDs to more advanced techniques exploiting implementation vulnerabilities [15, 51, 59, 64], protocol flaws [20, 63, 73], and side channels [21, 44]. This evolution has prompted DNS defenses such as source port and TxID randomization, Birthday protection, 0x20 encoding, and DNSSEC, which effectively mitigate off-path cache poisoning [15]. However, researchers have also identified flaws in the implementation of these defenses, creating new attack surfaces or allowing bypass of the protections [18, 19, 37, 40, 42, 44, 45]. After systematically reviewing the evolution of DNS cache poisoning threats, we noticed that DNS Birthday attacks [59, 64] have received limited follow-up attention, resulting in an oversight of their mitigation effectiveness, which now needs to be studied in depth.

Our Study. We identify new protocol-compliant vulnerabilities in the DNS extension mechanisms as new attack surfaces, which enable a novel DNS cache poisoning attack, termed REBIRTHDAY. Specifically, the attack exploits weaknesses in the EDNS Client Subnet (ECS) mechanism's option coherence checks to bypass DNS query aggregation policies adopted by resolvers to protect against DNS Birthday attacks. By bypassing query aggregation, attackers can trigger numerous identical rogue DNS responses from different source ports. This increases the likelihood of colliding source ports and injecting malicious responses into the resolver cache. Via controlled experiments, we demonstrated that REBIRTHDAY is totally practical and can lead to prevalent real-world impacts.

Through code review and active testing, we analyzed ECS processing implementations across 22 mainstream DNS software, including 9 recursive resolvers and 13 forwarders (Section 4). We identified a novel vulnerability in ECS processing, where DNS queries with client subnets are cached without proper coherence checks of subnet fields in responses. This flaw can bypass DNS query aggregation mechanisms and revive Birthday Paradox-based cache poisoning. Affected software includes BIND, Unbound, PowerDNS, Technitium, Dnsmasq, Pi-hole, and others. Additionally, we found that 11 DNS software lacks effective query aggregation mechanisms and one uses predictable TxID. These vulnerabilities collectively expose 18 software to exploitable DNS cache poisoning attacks.

To evaluate the capability of REBIRTHDAY, we conducted cache poisoning experiments across four representative vulnerable DNS software implementations (Section 5), achieving a 20/20 success rate by costing 358s on average. We then discovered the prevalence of vulnerable DNS resolvers (Section 6) in real-world environments. With ethical considerations in mind, we analyzed 21 prominent Wi-Fi routers, 6 router OSes, 45 public DNS services, and approximately 2.4M open DNS resolvers collected via Internet-wide scanning. Our findings revealed that a significant portion of in-use resolvers are

vulnerable to REBIRTHDAY. We also identified 16 router vendors (e.g., ASUS, TP-Link, and ZTE), 14 DNS services (e.g., AdGuard, Ali, and Quad9), and 365K (15%) open DNS resolvers as vulnerable.

Disclosure and Mitigation. We claim the root cause of REBIRTHDAY lies in the DNS protocol standards (RFC 7871), which specify that "a response that does not include the ECS option is still considered valid". We reported vulnerabilities to affected parties, including BIND, Unbound, PowerDNS, Google, Quad9, and Level3. We have received acknowledgments from 8 vendors, including BIND, Unbound, PowerDNS, Technitium, Dnsmasq, YogaDNS, Quad9, Adguard, and been assigned 50 CVEs. We discussed the attack details and solutions with them. For mitigating REBIRTHDAY, we suggest the stakeholders to validate the response's ECS option and aggregate queries according to the ECS responding state of nameservers.

Contributions. Our study makes the following contributions:

(1) We offered a comprehensive survey of previous DNS cache poisoning attacks and identified new vulnerabilities.
(2) We proposed a novel threat model, REBIRTHDAY, that revives the classic DNS Birthday attacks, affecting 18 DNS implementations with newly uncovered vulnerabilities.
(3) We demonstrated REBIRTHDAY could cause prevalent real-world threats, affecting 16 famous router vendors, 14 DNS service providers, and around 365K open DNS resolvers.
(4) We introduced mitigation solutions, responsibly reported findings to affected vendors, and assisted in resolving vulnerabilities with an online DNS-OARC discussion group.

2 Background

In this section, we introduce the basic concepts of DNS, its resolution process, and packet formats [49, 50]. Additionally, we present and analyze all types of DNS cache poisoning attacks to identify potential areas that have not been thoroughly studied.

2.1 DNS Overview

2.1.1 DNS Concepts and Resolution. The Domain Name System (DNS) improves the usability of IP-based applications by mapping domain names to IP addresses and vice versa. It consists of two core components: the DNS namespace and the resolution process.

The DNS namespace is a hierarchical and distributed database that organizes domain names into zones, separated by periods ("."). Each zone is managed by an authoritative nameserver responsible for maintaining authoritative data, referred to as resource records. These include A (IPv4 address) and AAAA (IPv6 address) records for domain-to-IP mappings, and NS (nameserver) records, which link parent zones to their child zones. As shown in Figure 1, the domain example.com consists of three zones: the Root zone ("."), the Top-Level Domain (TLD) zone (".com"), and the Second-Level Domain (SLD) zone ("example.com").

The DNS resolution process translates domain names into IP addresses iteratively. Figure 1 illustrates this process in detail. When a user accesses a website like example.com, the client application interact with the stub resolver, such as systemd-resolved [65], within the TCP/IP stack. The stub resolver formulates a DNS query and sends it to a pre-configured DNS forwarder, typically embedded in home or Wi-Fi routers [11, 36], to enhance performance by caching responses locally. The forwarder then relays the query to
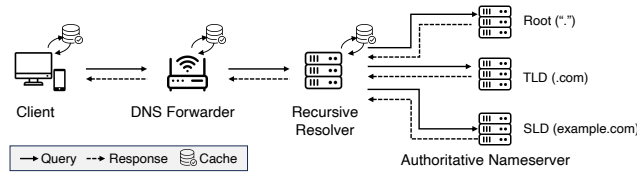
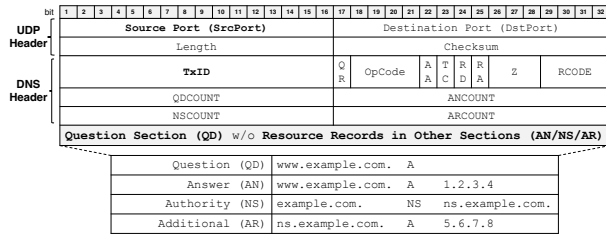**Figure 1: General DNS Resolution Process.**

Figure 2: DNS Packet Format.

a recursive resolver, which handles the iterative resolution process. The resolver starts at the Root nameserver, then queries the TLD and SLD nameservers sequentially. Each nameserver provides referral information, guiding the resolver to the next "closer" authoritative nameserver. Finally, the authoritative nameserver for `example.com` returns requested resource records, completing the resolution. If caching is enabled, the recursive resolver stores the response in its local cache database for faster resolution of subsequent queries.

2.1.2 DNS Packets. As shown in Figure 2, DNS typically uses UDP to transmit query and response payloads. A DNS packet consists of a header and a body. The header includes the transaction ID (`TxID`), the query/response flag (`QR`), operation code (`OpCode`), return code (`RCODE`), and the count of resource records. Specially, the TxID uniquely identifies each query-response pair, ensuring that resolvers can correctly associate incoming responses with their corresponding queries. The body contains four sections: the question section (present in both queries and responses, specifying the query name and type), the answer section (providing resolution results from authoritative nameservers with the AA flag), and the authority and additional sections (providing referral information without the AA flag, from higher-level nameservers).

2.2 DNS Cache Poisoning Attack

The DNS cache poisoning attack [12] aims to inject falsified resource records into a resolver's cache to manipulate subsequent queries. By exploiting vulnerabilities in the resolution process, attackers can redirect users to malicious domains, disrupt network services, or conduct man-in-the-middle attacks. These attacks have evolved over the years (1990 - 2025), transitioning from brute-force methods to sophisticated techniques that leverage protocol weaknesses or implementation flaws. This section provides a comprehensive overview of DNS cache poisoning attacks, covering fundamental principles, methodologies, and the evolution of attack strategies.

Basic Attack. Cache poisoning attacks exploit the DNS resolution process by injecting bogus response packets into a resolver's cache. As outlined in Section 2.1, DNS queries and responses depend on matching the source port and TxID. A resolver accepts a response
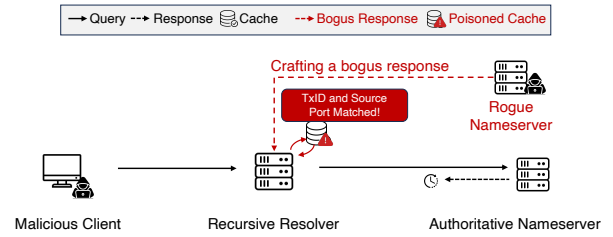

Figure 3: General DNS Cache Poisoning Model.

only if its source port and TxID match the query's source port and TxID. According to Figure 3, a general DNS cache poisoning attack involves the following steps:

(1) *Query Initiation or Interception:* The attacker initiates or intercepts a legitimate DNS query sent by the resolver to an authoritative nameserver.
(2) *Crafting a Bogus Response:* The attacker forges a DNS response packet, setting the TxID and source port to match those in the intercepted or guessed query.
(3) *Race Condition Exploitation:* The attacker delivers spoofed responses to the resolver before the legitimate response arrives from the authoritative nameserver.
(4) *Cache Injection:* Upon accepting the spoofed response, the resolver caches the falsified resource records, causing it to return incorrect data for future queries.

On-Path-Based Attack Stage. Early DNS cache poisoning attacks were predominantly conducted by the on-path attacker who exploited their position within the communication path to inject falsified resource records directly.

In 1990, Bellovin [5] identified a vulnerability in the file server that relied on hostname whitelists for user login verification. These servers performed a reverse DNS lookup using the `gethostbyaddr` function to obtain a hostname for the login host's IP address and then verified the hostname against a whitelist to grant access. Attackers bypassed this verification by modifying the reverse lookup PTR record of their IP address to return a whitelisted hostname. In 1993, Schuba [61] demonstrated how attackers could proactively inject falsified PTR records into the DNS cache using the additional section of DNS packets. This allowed attackers to poison subsequent reverse DNS lookups and bypass hostname-based authentication mechanisms. In 1995, Vixie [66] conducted a comprehensive analysis of DNS protocol design flaws, underscoring the insecurity of hostname-based authentication [8]. To mitigate such vulnerabilities, Vixie introduced the *Bailiwick Rules* and *Credibility Rules*, which were incorporated into newer versions of BIND. These measures restricted the acceptance of invalid resource records from the additional section, significantly enhancing DNS security.

The most infamous example of on-path cache poisoning occurred in 1997 when Kashpureff [13] exploited recursive resolvers lacking Bailiwick Rules enforcement. By returning falsified NS records from the on-path-controlled AlterNIC authoritative nameserver, he redirected subsequent queries for InterNIC domains to a server under his control, successfully poisoning the DNS cache. This incident catalyzed the widespread adoption of Bailiwick Rules by resolvers [64], which mitigated on-path cache poisoning attacks. Since then, these attacks have become infeasible in modern DNS.

However, DNS remained vulnerable to prediction-based attacks. In 1997, researchers [9, 51] discovered that the TxID used by BIND for external queries could be predicted due to its poor implementation. Coupled with the fixed source port for queries originating from the same client IP, the attacker were able to conduct off-path cache poisoning attacks by successfully guessing the TxID.

Exploitation of Randomness Weakness Stage. Beginning in 2000, the security community identified multiple DNS vulnerabilities that could be exploited through weaknesses in TxID and source port verification mechanisms.

In 2000, researchers [52] discovered a flaw in the TxID generator within the GNU C Library, rendering TxID values predictable and enabling attackers to perform cache poisoning attacks. In 2001, Zalewski [70] conducted an extensive analysis of pseudorandom number generator (PRNG) randomness across major operating systems, visually illustrating the predictability of TxID values and TCP initial sequence numbers [10]. In 2002, Sacramento [59] proposed leveraging the "*Birthday Paradox*" to enhance brute-force attacks against TxID. Exploiting the behavior of BIND resolvers, which initiated multiple queries for repeated domain requests, attackers increased their success probability by issuing multiple resolution requests simultaneously. Theoretically, sending 425 forged packets achieved a 50% success rate [23], effectively completing a cache poisoning attack [53]. In 2005, Kaminsky [28] highlighted the persistence of vulnerable configurations through network scans, discovering that over 230K of 2.5 million tested resolvers were still using BIND8, which lacked sufficient defenses against cache poisoning. Between 2007 and 2008, Klein revealed that TxID generation remained predictable in several DNS implementations, including BIND8 [31], BIND9 [32], PowerDNS [35], Windows [34], and OpenBSD [33], facilitating cache poisoning attacks.

The most prominent off-path attack during this stage was proposed by Kaminsky in 2008 [29]. This off-path attack employed TxID brute-forcing alongside randomized domain names to bypass cache limitations, enabling multiple rounds of attacks and the injection of falsified NS records. The underlying vulnerability was attributed to the lack of source port randomization, which affected nearly all domain resolver software at that time [54]. In 2009, Dagon et al. [15] developed a comprehensive cache poisoning attack model, analyzing various defense techniques, including source port randomization, TxID randomization, and 0x20 case randomization [16]. Their findings indicated that 57% of the 0.9 million tested resolvers had not implemented source port randomization, and vulnerabilities such as susceptibility to Birthday attacks persisted in systems like djbDNS. Since then, widespread adoption of source port and TxID randomization by resolvers [15] has substantially mitigated the feasibility of off-path DNS cache poisoning attacks.

Randomness Reduction Stage. Starting in 2010, the adoption of source port and TxID randomization rendered brute-force techniques for DNS cache poisoning increasingly infeasible. Consequently, the security community began exploring new methods to reduce the guessing space and achieve cache poisoning.

In 2012, Herzberg et al. [19] identified flaws in the source port allocation strategies for DNS queries and the authoritative nameserver IP selection in *NAT* network environments. These vulnerabilities allowed attackers to predict source ports and target authoritative nameserver IPs for several NAT devices, including those

running Linux and Windows, leading to successful cache poisoning attacks. By 2013, Herzberg et al. [22] highlighted that many DNS proxies, such as forwarding resolvers, had not fully adopted source port randomization. These proxies often employed fixed or sequentially increasing source port allocation, exposing them to cache poisoning risks. That same year, Herzberg et al. [21] introduced a cache poisoning method leveraging *Socket Overloading*. By sending high volumes of traffic to the victim host, attackers could detect packet loss, indicating open source ports, and then inject falsified replies by enumerating TxID values. In 2014, Shulman et al. [63] utilized DNS fragmentation as a timing side channel to infer source ports. Attackers sent the first fragment of a DNS response containing a source port to the target resolver. If the source port was correct, the resolver accepted the response; otherwise, it rejected the response and initiated a new request. By measuring response times, attackers could determine the correctness of the source port.

In addition to guessing source ports, Herzberg et al. [20] in 2013 proposed an alternative fragmentation-based cache poisoning attack. By predicting the IPID value and sending a false second fragment to the target resolver, attackers bypassed the need to guess source ports and TxIDs, as the DNS verification fields were located in the first fragment. In 2018, Brandt et al. [6] extended fragmented cache poisoning to certificate application verification. They demonstrated that off-path attackers could issue falsified certificates, compromising the security of the public key infrastructure.

Finally, in 2019, Alharbi et al. [1] introduced a cache poisoning attack targeting DNS clients. By deploying malware to occupy local source ports, they forced DNS queries to use the only available reserved port. Combined with TxID enumeration, this technique enabled attackers to execute cache poisoning successfully.

Revival Through Protocol Vulnerabilities Stage. Since 2020, the security community has leveraged various protocol vulnerabilities to propose new techniques for reviving DNS cache poisoning attacks rather than traditional guessing-style methods.

Following the development of fragmented cache poisoning attacks, academia and industry suggested minimizing fragmentation as a mitigation measure. However, to bypass this limitation, Zheng et al. [73] in 2020 exploited the use of extended CNAME resolution chains to increase DNS packet sizes, forcing recursive resolvers to fragment them. Using fragmented cache poisoning techniques, they successfully injected false resource records into DNS forwarders that accepted fragments and lacked bailiwick checking.

In 2020 and 2021, Man et al. [44, 45] introduced two novel cache poisoning attacks based on ICMP side channels. These attacks allowed rapid source port guessing and false reply injection by enumerating TxID values within a short time frame. The side channels exploited the global ICMP error message rate limiting counter, as well as fragmentation and redirection messages in the Linux operating system. By observing the presence or absence of side channel information, attackers could efficiently determine the source port used in DNS queries within tens to hundreds of milliseconds.

In 2021, Klein [37] analyzed the pseudorandom number generator (PRNG) of the Linux operating system. By observing fields such as the IPv6 flow label and the IPv4 IPID in packets, Klein was able to reverse-engineer the initial state of the PRNG, calculate the source port number, and successfully complete a cache poisoning attack. Between 2021 and 2022, Jeitner et al. [26, 27] proposed two new

cache poisoning attacks based on ambiguities in special character parsing. By leveraging inconsistencies in how resolvers processed special characters like "." and "000", they demonstrated how on-path controlled authoritative nameservers could inject false replies. These parsing ambiguities caused the same string to be decoded into different domain names, enabling cache poisoning.

In 2023, Heftrig et al. [18] discovered vulnerabilities in DNSSEC validation implemented by certain well-known public DNS resolution services. These vulnerabilities exposed domains protected by DNSSEC signatures to the risk of cache poisoning, highlighting new security risks even in environments utilizing DNSSEC. That same year, Li et al [40] introduced a new DNS cache poisoning attack, which exploits vulnerabilities in bailiwick checking algorithms to perform DNS cache poisoning. This attack targets DNS servers acting as both recursive resolvers and forwarders, allowing attackers to take control of entire DNS zones.

Most recently, in 2024, Li et al. [42] introduced a novel DNS cache poisoning technique by exploiting logic vulnerabilities in DNS response pre-processing. They demonstrated that malformed DNS packets, when processed incorrectly, could be leveraged as a side channel to inject malicious DNS records into the cache within 1s, bypassing traditional defenses such as source port randomization. **Insights from the Evolution.** Existing DNS cache poisoning attacks, such as on-path and off-path methods leveraging source port and TxID guessing, have garnered significant attention and extensive research, leading to numerous attack strategies and network measurements. In contrast, DNS Birthday attacks [59], proposed and mitigated in 2002, lack systematic follow-up studies, leaving critical questions about their impact on implementations unanswered. Therefore, in this study, we systematically examine the overlooked DNS Birthday attack vector, uncovering previously unidentified vulnerabilities that enable the revival of DNS cache poisoning attacks and highlighting their extensive and widespread implications across modern DNS software, resolvers, and services.

3 Overview of REBIRTHDAY Attack

In this paper, we present a novel DNS cache poisoning attack, named REBIRTHDAY attack, which leverages the DNS protocol extension as the new attack vector to revive the classic DNS Birthday attack. These protocol extensions, while compliant with DNS specifications, introduce new vulnerabilities, which bypass the mitigation strategy: Query Aggregation. *With the new vulnerabilities reviving the DNS Birthday attack that was no longer workable since 2002, we dubbed them the REBIRTHDAY attack ("Rebirth").*

This section begins by examining the fundamental of the DNS Birthday attack, followed by an introduction to the new threat model and the basic attack workflow. Further attack details and experiment results are provided in Section 4, Section 5, and Section 6.

3.1 DNS Birthday Attack

The DNS Birthday attack leverages the statistical principle known as the "Birthday Paradox" [64] to significantly enhance the probability of brute-forcing DNS TxIDs. Initially proposed by Sacramento in 2002 [59], the attack exploits resolver behaviors such as issuing multiple identical queries for repeated domain name requests, thus allowing attackers to amplify the chances of successfully injecting
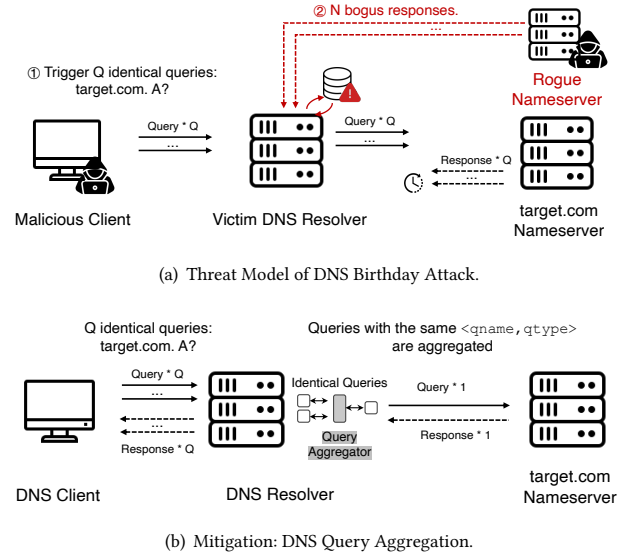

(a) Threat Model of DNS Birthday Attack.

(b) Mitigation: DNS Query Aggregation.

Figure 4: DNS Birthday Attack and Mitigation Strategy.

malicious responses. This strategy effectively transforms the classic brute-force attacks on TxIDs into a practical cache poisoning technique by utilizing the Birthday Paradox.

3.1.1 Birthday Paradox. The Birthday Paradox [46, 47] is a counter-intuitive statistical phenomenon which states that the probability of two people sharing the same birthday in a group grows rapidly with the group size, even when the total possible outcomes (365 days) remain constant. In the context of DNS, this principle is applied to increase the likelihood of TxID collisions. By generating multiple identical queries and spoofing a large number of forged responses, attackers can probabilistically achieve a collision between a legitimate TxID and a spoofed one. For example, sending 425 forged packets theoretically yields a 50% success rate for collision [23], which contributes to the DNS cache poisoning attack.

3.1.2 Attack Steps. The threat model of DNS Birthday attacks is demonstrated in Figure 4(a). To initialize an attack, an attacker needs to follow the following steps:

(1) The attacker identifies a vulnerable DNS resolver that issues multiple queries for repeated domain name requests.
(2) The attacker floods the resolver with numerous identical DNS queries for the target domain name, forcing it to initiate multiple simultaneous resolution requests with different used TxIDs to an authoritative nameserver.
(3) The attacker simultaneously sends a large number of forged DNS responses with randomly generated TxIDs, hoping to match one of the TxIDs generated by the resolver.
(4) Upon achieving a TxID collision, the attacker could inject a malicious resource record into the resolver's cache, indicating a successful DNS cache poisoning attack.

3.1.3 Attack Metrics. To analyze the success probability of the DNS Birthday attack, we introduce the following key metrics and equations, as well as their relationships.

Metrics Definition. The key metrics are defined below:

- Q: Total number of DNS queries simultaneously triggered for the same domain (using different source ports or TxIDs).
- T: Total size of the randomness (source port or TxID) space, typically $T = 2^{16}$ for the source port or TxID.
- n: Number of unique source port or TxID collisions needed for successful cache poisoning.
- P_{success}: Probability of a successful DNS Birthday attack after r attack rounds.

Success Probability. The probability of success in a single attack round is derived from the classic birthday problem:

$$P_{\text{single}} = 1 - \prod_{i=0}^{n-1} \left(\frac{T - Q \cdot i}{T} \right) \tag{1}$$

where Q represents the number of triggered queries, and T represents the size of the randomness space.

Cumulative Success Probability. To improve the overall success rate, an attacker can repeat the attack multiple times. The cumulative success probability after r rounds is given by:

$$P_{\text{success}} = 1 - \left(1 - P_{\text{single}} \right)^r \tag{2}$$

3.1.4 Attack Effectiveness. The effectiveness of the DNS Birthday attack lies in its ability to amplify the collision probability using a high volume of simultaneous queries and forged responses. This attack was demonstrated against BIND resolvers, where the attacker could achieve a 50% success probability with 425 forged packets to collide TxIDs due to the lack of source port randomization [23, 59]. By exploiting resolver behaviors compliant with the DNS protocol, the attack significantly reduces the effort required for successful cache poisoning, posing a severe risk to DNS security.

3.1.5 Mitigations. To address the vulnerabilities exploited by the DNS Birthday attack, an effective mitigation strategy involves implementing the *Query Aggregation* mechanism [23, 53]. As shown in Figure 4(b), the mechanism works by merging identical DNS requests (identified if they share the same key: `<qname, qtype>`) for the same domain name into a single query. Instead of sending out separate queries for each identical request, the resolver combines them and processes a single resolution workflow. Once the DNS response is received, it is distributed to all pending queries, ensuring consistency and minimizing the attack surface. This mechanism eliminates the creation of multiple DNS queries that an attacker could exploit using the Birthday Paradox, effectively mitigating the probability of a successful cache poisoning attack.

However, after discovery and mitigation, the DNS Birthday attack was confirmed to affect a set of DNS software implementations, such as BIND and Microsoft DNS. Since then, despite significant advancements in DNS protocol standards and resolver implementations, there has been a noticeable lack of in-depth and systematic analysis on the broader applicability and feasibility of the attack. Specifically, questions such as whether this attack could still impact other DNS software or whether it remains a viable threat in today's Internet have remained largely unexplored. As a result, the overall picture of the DNS Birthday attack and its potential impact across modern DNS systems remains unclear, leaving a significant gap in understanding and an opportunity for further investigation.
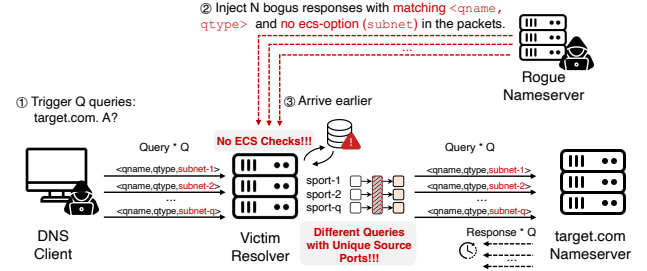

Figure 5: Threat Model of the REBIRTHDAY Attack.

3.2 Threat Model of REBIRTHDAY Attack

Figure 5 illustrates the threat model of the REBIRTHDAY attack. In this model, we assume that an attacker is a DNS client that can trigger domain queries to target DNS resolvers and obtain their egress IP addresses. The attacker aims to inject malicious DNS responses into the target resolvers' DNS cache. Specifically, both DNS forwarders and recursive resolvers are affected by REBIRTHDAY.

When sending DNS queries, attackers do not need to spoof the source IP address, as most resolvers do not verify the client's IP or subnet information (see Section 6.2). For open resolvers distributed across the Internet, attackers can directly send DNS queries to the resolver from any location [60]. For DNS resolvers serving more limited networks, such as those in home or enterprise networks, attackers can leverage large-scale measurement platforms [57] or residential proxy networks [43, 48] to generate DNS queries. Before launching the REBIRTHDAY attack, an attacker must collect the egress IP address of the target resolver to inject responses. This can be achieved by querying the target resolver, and then obtaining the egress IP address from the perspective of the authoritative nameserver controlled by the attacker.

Additionally, we assume that the attacker is off-path and needs to spoof the source IP address of a forged response. According to recent data (Dec. 2024) from CAIDA [7], over 19% of IPv4 ASes are classified as IP-spoofable, making it still feasible for an attacker to use bulletproof hosting services [2] within these ASes to spoof the source address. Besides, to ensure that malformed packets are delivered before legitimate responses, we assume the attacker can generate response packets from a neighboring host. Specially, we exclude resolvers that enable DNSSEC validation [4] and 0x20 encoding [16] from our consideration.

3.3 Attack Workflow

REBIRTHDAY revitalizes the traditional DNS Birthday attack by leveraging a novel attack vector stemming from protocol-compliant DNS extensions. This newly identified vulnerability allows attackers to bypass the query aggregation mechanism of resolvers, forcing the target resolver to issue multiple DNS queries for the same domain by utilizing distinct source ports. This breakthrough significantly increases the effectiveness of DNS Birthday attacks, even facing the widespread adoption of mitigation strategies.

3.3.1 Step-by-Step Workflow. The REBIRTHDAY attack operates through three steps, as depicted in Figure 5, which exploit the newly identified vulnerability to achieve a high success rate.

Step ①: Triggering Multiple Queries. The attacker begins by initiating multiple carefully crafted DNS queries for a same domain name to the victim resolver. By exploiting the identified logic flaw in the DNS extension implementation, the resolver generates multiple identical queries for the same domain instead of aggregating them into a single query, using different source ports. This behavior directly bypasses the query aggregation defense, laying the groundwork for the subsequent steps of the attack.

Step ②: Injecting Malicious Responses. The attacker follows by injecting a large number of bogus DNS responses. These responses exploit the resolver's response handling vulnerabilities, increasing the collision probability between the attacker's bogus responses and the resolver's legitimate queries. By manipulating the source port (guessing a small range) and TxID (brute-forcing all 65,536), the attacker maximizes the effectiveness of the attack.

Step ③: Returning before Legitimate Response. To succeed, the attacker ensures that the forged responses arrive before the legitimate responses. This is critical to poisoning the resolver's cache with the malicious DNS record. If a single attack attempt is unsuccessful, the process is repeated starting from step ① with a new domain name to bypass the impact of caching and scope-prefix.

3.3.2 Key Innovation. The critical innovation of the REBIRTHDAY attack lies in exploiting a protocol-compliant vulnerability to bypass query aggregation and trigger multiple DNS queries for a same domain name. By systematically exploiting this novel vulnerability, the REBIRTHDAY attack reintroduces the DNS Birthday attack as a significant threat to modern DNS infrastructures.

4 Bypassing DNS Query Aggregation

DNS query aggregation has been recognized as an effective defense mechanism against the DNS Birthday attack since its widespread adoption by resolvers in 2002 [59, 64]. However, to date, no work has conducted a systematic and in-depth analysis of its software-specific implementation and potential weaknesses. We find that the introduction of DNS extensions, particularly EDNS(0) and Client Subnet (ECS) mechanism, opens new attack surfaces for attackers, enabling bypassing of DNS query aggregation and reigniting the potential of DNS Birthday attacks.

In this section, we thoroughly analyze the EDNS(0) and ECS mechanisms, including a summary of protocol specifications and an extensive examination of their implementation in 22 mainstream DNS software. These include 9 recursive resolvers and 13 DNS forwarders that are widely studied in prior works [38, 40–42, 69, 72], as listed in Table 1. Our approach involves both source code inspection and local testing to elucidate the basic workflow of ECS processing and to identify implementation-specific differences across various software. Through detailed analysis and testing, we uncovered novel vulnerabilities that facilitate bypassing DNS query aggregation, thereby enabling the reconstruction of DNS Birthday attacks (REBIRTHDAY) to poison DNS caches effectively.

4.1 EDNS(0) and ECS Mechanisms

4.1.1 EDNS(0): Extended DNS Mechanisms. EDNS(0), introduced in RFC 2671 [67] and updated in RFC 6891 [17], is an extension to the DNS protocol designed to overcome the limitations of the original DNS specification, which restricted message sizes to 512
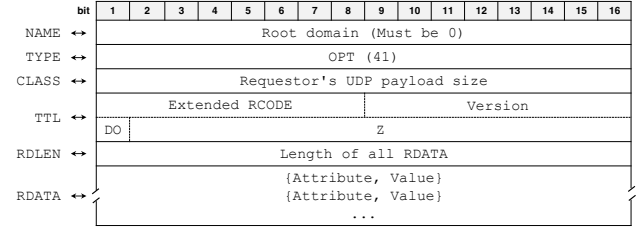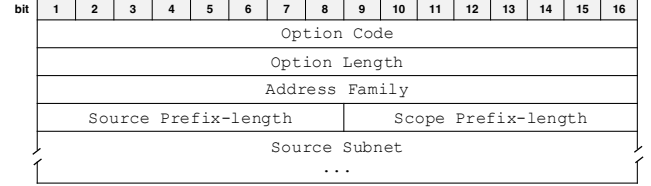

Figure 6: DNS EDNS(0) Data Format.

Figure 7: ECS Option Format.

bytes. By enabling larger DNS message sizes, EDNS(0) facilitates the transmission of additional information and supports advanced DNS features. EDNS(0) achieves this by adding optional fields in the DNS header, known as OPT pseudo-records, which allow extended functionality while maintaining backward compatibility.

The primary enhancement of EDNS(0) is its support for DNS messages exceeding the original size limit, improving DNS efficiency and enabling extensions such as DNSSEC and ECS (EDNS Client Subnet) options. Furthermore, EDNS(0) permits the specification of extended flags and options in DNS queries and responses, offering a flexible framework for future enhancements to the DNS protocol. **Data Format.** EDNS(0) extends the traditional DNS protocol by introducing an optional OPT pseudo-record in the DNS additional section of DNS messages. The data format of such an OPT record is shown in Figure 6. It includes fields such as the Root name (set to 0), the fixed type (OPT, value 41), the UDP payload size, extended flags (e.g., DNSSEC signaling), and flexible option data structures for encoding new features. This backward-compatible extension enhances DNS capabilities while maintaining interoperability with non-EDNS(0)-aware systems.

4.1.2 ECS: EDNS Client Subnet. The EDNS Client Subnet (ECS) mechanism, standardized in RFC 7871 [14], is an extension option of EDNS(0) that aims to optimize DNS resolution for geographically distributed clients. ECS allows DNS resolvers to include a portion of the client's IP address (named as the client subnet) in DNS queries sent to authoritative nameservers. This information enables the authoritative nameserver to tailor DNS responses based on the client's geographical location, improving content delivery and reducing latency for end-users.

Option Format. The option format of ECS is shown in Figure 7. This option is structured as follows: it begins with an option code identifying it as ECS, followed by a length field. It then specifies the address family (IPv4 or IPv6) and the source prefix length of the client's IP address, allowing the authoritative nameserver to use the client's subnet for more localized responses. The client's subnet data (address) is included based on the specified scope prefix length from the authoritative nameserver.

ECS Processing Steps. The following outlines how the ECS interacts with resolvers, authoritative nameservers, and response caching mechanisms during a DNS query-response cycle [14].

(I) Resolver Originating Query. When a resolver receives a query from a client, it determines whether to include an ECS option based on its configuration and policy. Typically, the client's source IP address is truncated to a predefined prefix length (e.g., 24 bits for IPv4 or 56 bits for IPv6), and this truncated subnet is included in the ECS option. If the client's query already contains an ECS option, the resolver uses the subnet specified in the original ECS option. The ECS option encodes this subnet with fields specifying the address family, source prefix length, and the truncated IP address.

(II) Authoritative Nameserver Generating Response. When an authoritative nameserver receives a query containing the ECS option, it determines whether to use the included subnet for its response. If ECS processing is supported, the nameserver tailors its response based on the subnet, potentially providing more geographically or topologically relevant answers (e.g., selecting a closer content delivery node). The nameserver includes an ECS option in its response, reflecting the scope of the provided answer by specifying a scope prefix length, which may differ from the original query. If the nameserver does not support ECS processing, it must omit the ECS option in the response, signaling to the client that ECS was not used to generate the reply (Null ECS option).

(III) Resolver Handling Response. When a resolver processes a response containing the ECS option, it checks the `Family`, `Address`, `Source Prefix-Length` fields from the response against those in the original query. If there is any mismatch, the entire response is discarded to maintain consistency. If the ECS option matches, the resolver will use both the source and scope prefix length to determine the subnet. If the response does not include an ECS option, it is treated as applicable to all client addresses.

(IV) Caching and Subsequent Queries. When caching responses with ECS, the resolver associates the cached entry with the specific subnet used in the query. For subsequent queries from different subnets, the resolver either uses an existing cache entry that matches the new subnet or issues a new query to the upstream.

4.1.3 Software Implementations. We summarize the implementations related to ECS processing in Table 1 based on our analysis of 22 DNS software (9 resursives and 13 forwarders).

- **BIND** has two major versions with differing support for ECS processing. The Open Source version does not support ECS functionality. When responding to client queries, it returns the ECS option in the response exactly as it appeared in the client query, without further processing. The Subscription Edition (BIND -S), on the other hand, supports ECS. Upon receiving a client query that includes an ECS option, it extracts the ECS information and uses it to construct the resolver query sent to the authoritative nameserver, ensuring that the query is tailored based on the ECS subnet specified in the original client request.
- **Unbound** provides support for ECS (Disabled by default). To activate ECS, the `subnetcache` module must be added, and the `client-subnet-always-forward` option must be enabled. Upon receiving a client query that includes an ECS

option, Unbound leverages this option to generate corresponding queries sent to the authoritative nameserver. Unbound maintains distinct cache entries for different subnets.

- **PowerDNS** supports ECS but disables it by default. It can be enabled by configuring `use-incoming-edns-subnet` and `edns-subnet-allow-list`. Like Unbound, PowerDNS utilizes ECS data from client queries for resolver queries, optimizing responses based on the specified subnet.
- **Knot**, **Microsoft**, **Simple DNS Plus**, **MaraDNS**, and **HickoryDNS** do not offer ECS functionality to handle ECS options.
- **Technitium** implements ECS, disabled by default. By enabling the ECS option, Technitium honors client ECS data for resolution and maintains separate caches for each subnet.
- **Dnsmasq** and **Pi-hole**, by default, only return ECS option to clients. When `-add-subnet` is enabled, they forward client's ECS data to the upstream while ignoring the cache.
- **CoreDNS** does not implement ECS functionality.
- **DNSDist**, **Acrylic DNS**, **AdGuard**, **AdGuard Home**, **DNS Safety**, **Dual DHCP DNS**, and **YogaDNS** supports both replying with ECS data to clients and forwarding the ECS option to upstream resolvers by default.
- **SmartDNS** enables ECS functionality by default and follows the general ECS processing logic shown above for utilizing the ECS option in the client query.
- **pdnsd** only supports returning the ECS option to the client.
- **NxFilter** only supports forwarding ECS options to upstreams.

4.2 Vulnerability in DNS Query Aggregation

We identified new DNS query aggregation vulnerabilities and other poor source port or TxID randomization implementations.

Vulnerabilities in ECS Processing. As shown in Table 1 (Column "*No Query Aggregation*" and "*Vulnerable*"), through analysis and testing, we identified a novel vulnerability affecting six DNS software that support ECS: *BIND -S*, *Unbound*, *PowerDNS Recursor*, *Technitium DNS*, *Dnsmasq*, and *Pi-hole*. This vulnerability arises from two key issues in their ECS processing mechanisms.

Firstly, when handling client queries containing ECS subnets, these implementations verify the presence of a cache entry associated with the subnet. If such a cache exists, the query is directly resolved from it. If no cache entry exists, the software determines whether there is an ongoing query for the same subnet. If no such query exists, the resolver initiates a new query using the specified subnet. Consequently, the identifier for determining identical queries expands from the two-tuple <qname, qtype> to the three-tuple <qname, qtype, subnet>. This extension inadvertently permits attackers to bypass query aggregation defenses by appending different, spoofed subnets to otherwise identical queries, forcing the resolver to issue a large volume of queries for the same domain.

Secondly, upon receiving responses from authoritative nameservers, these implementations should validate that the subnet in the response aligns with the original query. However, as specified by DNS protocol standards (RFC 7871), a response lacking the ECS option is still considered valid, indicating that the authoritative nameserver does not support ECS. Attackers can exploit this weakness by injecting forged responses without the ECS option, relying solely on the two-tuple <qname, qtype> to bypass validation.

Table 1: ECS Processing Implementations of 22 DNS Software (18 Vulnerable).

Resolver				ECS		No Query Aggregation		# of Rate-limit	Vulnerable
Actor	Index	Software	Version	Reply	Request	Without ECS	With ECS		
Recur-sive	#1	BIND -S	9.18.37	✓	✓	✗	✓	100	✓
	#2	Unbound	1.23.0	✓	✓	✗	✓	20k	✓
	#3	PowerDNS Recursor	5.2.2	✓	✓	✗	✓	500	✓
	#4	Knot Resolver	5.7.5	✗	✗	✗	✗	30k	✗
	#5	Microsoft DNS	2025	✗	✗	✗	✗	30k	✗
	#6	Technitium DNS	13.3	✓	✓	✗	✓	30k	✓
	#7	Simple DNS Plus	9.1.116	✗	✗	✗	✗	3k	✗
	#8	MaraDNS	3.5.0036	✗	✗	✗	✗	7	✗
	#9	HickoryDNS	0.26.0	✗	✗	✓	✓	800	✓
Forw-sarder	#10	Dnsmasq	2.91	✓	✓	✗	✓	150	✓
	#11	CoreDNS	1.12.0	✗	✗	✓	✓	3k	✓
	#12	DNSDist	2.0.0	✓	✓	✓	✓	450	✓
	#13	SmartDNS	46.1	✓	✓	✗	✗	1k	✓*
	#14	Pi-hole	6.1.2	✓	✓	✗	✓	150	✓
	#15	pdnsd	1.2.9a	✓	✗	✓	✓	40	✓
	#16	Acrylic DNS	2.2.1	✓	✓	✓	✓	14k	✓
	#17	AdGuard	7.19	✓	✓	✓	✓	9k	✓
	#18	AdGuard Home	0.107.62	✓	✓	✓	✓	20	✓
	#19	DNS Safety	2.1.0	✓	✓	✓	✓	7k	✓
	#20	Dual DHCP DNS	8.01	✓	✓	✓	✓	1.5k	✓
	#21	NxFilter	4.7.1.9	✗	✓	✓	✓	3.4k	✓
	#22	YogaDNS	1.47	✓	✓	✓	✓	10k	✓

✓: Yes. ✗: No. ✓: Vulnerable. ✗: Not vulnerable. *: Vulnerable due to poor source port or TxID randomization. # of Rate-limit is per IP.

These flaws collectively introduce a novel attack vector that effectively circumvents query aggregation mechanisms and resurrects the DNS Birthday attack, enabling DNS cache poisoning.

Flaws in Query Aggregation Implementations. We also find that several DNS software implementations remain vulnerable to cache poisoning due to insufficient defenses against traditional DNS Birthday attacks. Specifically, query aggregation mechanisms are absent or ineffective in *HickoryDNS*, *CoreDNS*, *DNSDist*, *pdnsd*, *Acrylic DNS*, *AdGuard*, *AdGuard Home*, *DNS Safety*, *Dual DHCP DNS*, *NxFilter*, and *YogaDNS*.

Flaws in Randomization Implementations. Furthermore, vulnerabilities in source port and TxID randomization still significantly exacerbate the risk of exploitation. For instance, *SmartDNS* exhibits inadequate source port randomization over extended periods. *DNS-Dist* uses fixed source ports and sequential TxIDs for queries to the same upstream resolver. *Acrylic DNS* and *YogaDNS* employ incrementing source ports, while *AdGuard* combines incrementing source ports with client query TxIDs for self-requests, and *AdGuard Home* exclusively relies on client query TxIDs too. Additionally, *Dual DHCP DNS* uses fixed source ports in conjunction with client query TxIDs for resolution. These flaws collectively expose these implementations to easily exploitable cache poisoning attacks, undermining DNS security and corresponding Internet services.

In summary, we discovered new DNS query aggregation vulnerabilities that could be exploited to conduct REBIRTHDAY attacks, and classic flaws in source port or TxID randomization. Those vulnerabilities could bypass the randomization defenses and enable DNS cache poisoning attacks.

5 Evaluation of REBIRTHDAY Attack

In this section, we develop end-to-end cache poisoning attack experiments exploiting the REBIRTHDAY technique described above and evaluate them across four vulnerable DNS software implementations: Unbound, PowerDNS Recursor, Technitium DNS, and CoreDNS. For ECS-enabled resolvers, we targeted three mainstream software: Unbound, PowerDNS Recursor, and Technitium DNS. Although the BIND Subscription Edition is not publicly available, we verified its vulnerability through testing with industry partners (Quad9 DNS) utilizing this version. For non-ECS-supporting resolvers, we selected CoreDNS due to its widespread use. While Dnsmasq is also affected by REBIRTHDAY, it does not enable caching for ECS, making it applicable only for attacks targeting unqueried or nonexistent domains [71]. Other DNS software behave consistently with CoreDNS in our evaluations.

5.1 Attack Design

5.1.1 Experiment Setup. For our evaluations, we installed Unbound, PowerDNS, and CoreDNS on machines running Ubuntu 24.04 as the host operating system, while Technitium was deployed on Windows Server 2022. Each DNS resolver was configured to enable ECS support. We utilized the domain victim.com for testing, configuring it under a controlled authoritative nameserver. We deployed attackers and clients on machines within the same local network as the DNS resolvers. Each experiment involved three to five clients issuing queries to the resolvers, while the attacker machine sent multiple queries with different ECS subnets to the target resolver and forged malicious responses targeting these queries. The query packets,
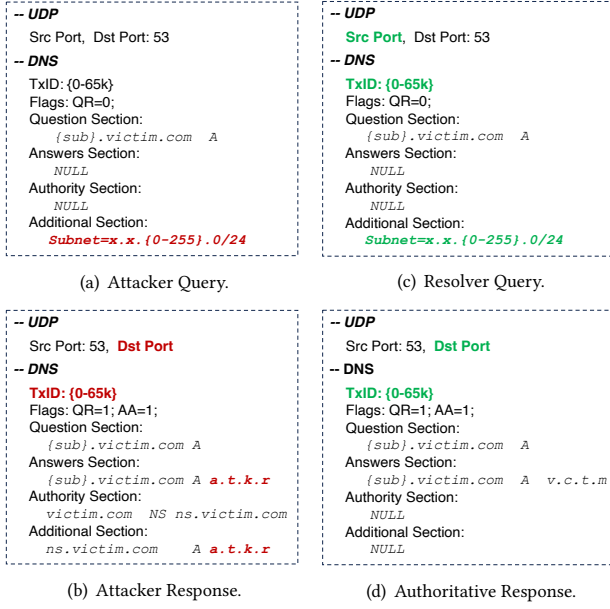
(a) Attacker Query.

(c) Resolver Query.

(b) Attacker Response.

(d) Authoritative Response.

Figure 8: DNS Query and Response Packets.

legitimate authoritative response packets, and forged attacker response packets are illustrated in Figure 8, respectively. Here, we use victim.com for anonymity purposes. In actual experiments, real domain names are used. If the attack succeeds, victim.com will be hijacked to a.t.k.r controlled by attackers. The network was configured with a public IP assigned to the resolver, allowing access from both legitimate clients and potential external traffic.

5.1.2 Rate-limit Testing. The feasibility of REBIRTHDAY is significantly influenced by the number of queries a resolver accepts from a single client. As discussed in Section 3.1, the probability of a successful attack increases with the number of queries that can be triggered. To investigate this aspect, we conducted a systematic evaluation of the rate-limiting configurations across 22 DNS software (Column "*Rate-limit*" in Table 1), measuring the maximum number of queries permitted per client. Most impacted software allowed a rate limit exceeding 200 queries per client. For resolvers with lower thresholds, attackers could bypass these limitations in practical scenarios by employing multiple clients or spoofing source IP addresses to increase the number of identical queries generated.

5.1.3 Attack Steps. The core of REBIRTHDAY involves the following steps. In each round, the attacker first sends multiple queries containing the ECS option to the target resolver. In Figure 8(a), the attacker queries for the A record of {nonce}.victim.com, where nonce is a random string used to bypass the cache. The query also contains a unique subnet value set to x.x.{0-255}.0/24. Secondly, upon receiving the query, the target resolver will handle it based on the three-tuple <qname, qtype, subnet>. The attacker uses this to bypass query aggregation and triggers a series of resolver queries to the authoritative nameserver, where each query uses a different source port, as illustrated in Figure 8(c). Thirdly, the attacker then injects fake responses to the target resolver before

Table 2: REBIRTHDAY Attack Results.

Software	Avg. Round Taken	Avg. Time Taken	Success Rate
Unbound	263	593s	20/20
PowerDNS Recursor	328	237s	20/20
CoreDNS	20	245s	20/20

the legitimate response is received, as shown in Figure 8(b) and Figure 8(d). The attack strategy involves guessing one or more source ports and brute-forcing 65,536 possible TxIDs. This is made feasible by the Birthday Paradox, where the greater the number of queries sent by the resolver (i.e., the more source ports used), the higher the probability of success. Lastly, after injecting the response, the attacker checks if the returned IP is a.t.k.r, indicating a successful attack. If the result is v.c.t.m, the attack has failed; the attacker then repeats the process by starting a new round of queries.

Success Probability. According to the rate-limit testing results, we send 200 queries in each round (200 different source ports) and guess only one random source port with brute-forcing 65,536 TxIDs. The success probability of the attack after 1,800 rounds, based on the Birthday Paradox, is calculated as follows (around 99.6%):

$$P_{\text{success}} = 1 - \left(1 - \frac{200}{65,536}\right)^{1,800} \approx 0.99592$$

Where *200* is the number of queries (or source ports) per round and *65536* is the total number of possible source ports.

5.2 End-to-End Attacks

We conducted the REBIRTHDAY attack 20 times using programs developed in Golang, which implement the techniques above. To introduce query latency, we delayed the response from each software by a specified duration before returning the legitimate response. The experiment results are presented in Table 2. The attack success rate is 20/20, except for Technitium. The maximum bandwidth required to inject 65,536 packets with different TxIDs is about 119Mbps.

5.2.1 Attacking Unbound. Due to Unbound's retry algorithm that uses intervals starting from 0.1s, progressing to 0.5s, then 1.5s, and eventually increasing further, we introduce a 0.5s delay per round. To enhance performance, we set the queries per thread to 200 (default is 30). On average, the attack takes only *263* rounds (*593s*). The minimum time is 23s while the maximum time is 2,426s.

5.2.2 Attacking PowerDNS Recursor. PowerDNS performs only one retry with a timeout window of 1.5s. To account for this, we introduce a 1s delay per round. Additionally, PowerDNS employs a spoof-nearmiss-max setting to detect spoofed responses (default 1). For testing, we modify the value to 0. On average, the attack requires *328* rounds (*237s*). The attack took between 60s and 480s.

5.2.3 Attacking CoreDNS. Given CoreDNS's 6s query timeout, we introduce a 900ms delay per round. With a rate limit of 3k queries, we increase the queries per round to 2,000 to optimize efficiency. Theoretically, with 2,000 queries, 300 rounds are required for a *99.9%* success rate. In practice, the attack completes in an average of *20* rounds (*245s*). The attack duration varied from 11s to 1,479s.

5.2.4 Attacking Technitium DNS. Technitium performs a single retry with a timeout of 1.5 seconds. We observed that Technitium deploys *Delegation Revalidation* [24] to validate NS records and only accepts NS records from referrals or authoritative answers. During the experiment, when Technitium received a high volume of random domain queries and fake responses, it initiated numerous revalidation queries but failed to respond to legitimate ones, making it impossible to determine the success of the REBIRTHDAY. However, this resulted in a DoS effect.

5.2.5 Discussion. For practical attacks, we need to consider the following factors that affect the attack result in the real world.

Multiple Nameservers. Many domains are configured with multiple nameserver IPs, which can potentially complicate the attack surface. Our analysis of the latest zone files for `.com` and `.net` domains [25], revealed the median number of nameserver IPs per domain is 4. Given that our attack requires a maximum of 65,536 packets to brute-force, attackers could exploit this by spoofing the IP addresses of all four nameservers simultaneously, sending 65,536×4 packets.

Multiple Backend IPs. Public DNS services typically operate with multiple backend IPs. This increases the injection space for attackers. However, the selection of backend IPs is often influenced by factors such as geolocation and load balancing. In our experiments, we found that affected public DNS services (tested in Section 6.2) had limited backend IPs (e.g., 1-4) in use from a specific geolocation. This simplifies the attacker's task, as they only need to spoof a few select IP addresses at once.

6 Measurement of Vulnerable Resolvers

In this section, we present a comprehensive measurement of the REBIRTHDAY in real-world environments, analyzing 21 Wi-Fi routers, 6 router OSes, 45 public DNS services, and 2.4M open DNS resolvers. Our extensive testing demonstrates that a significant portion of the resolver population remains vulnerable to REBIRTHDAY, highlighting the widespread applicability and impact of REBIRTHDAY.

Testing Design. To evaluate vulnerable resolvers, we focus on assessing the key attack factors (ECS support and query aggregation) rather than conducting the attack itself. We perform two sets of tests to measure these factors. Firstly, we send DNS queries containing the ECS option to the target resolver and observe whether the resolver correctly forwards the ECS information to the upstream DNS servers and whether the ECS option is subsequently returned to the client. Secondly, we send 200 DNS queries to each resolver to examine whether the resolver aggregates these queries, which could potentially impact its vulnerability to attacks. On the nameserver side, we could observe the number of DNS queries from each resolver. We can also identify how many backend IPs each resolver uses. If no query aggregation is observed, we classify the target resolver as vulnerable to the attack.

6.1 Wi-Fi Routers and Router-OSes

6.1.1 Router and OS List. We collected 21 popular Wi-Fi routers and 6 router OSes listed in [58, 68] to evaluate, shown in Table 3.

6.1.2 Testing. We connected our client machine to these routers (OSes) and configured their upstream DNS servers to point directly

Table 3: Results of 27 Wi-Fi Routers and OSes (16 Vulnerable).

Index	Vendor	Version	No Q. Agg.	Vul.
#1	360 WI-FI6 T7	4.2.4	✗	✗
#2	ASUS RT-AC66U	384.18	✓	✓
#3	CISCO Router	1.2.1.7	✓	✓
#4	D-Link 7001	17.01.11A1	✓	✓
#5	Fast FAC1200R	1.0.0	✓	✓
#6	Fiberhome SR4201SA	RP0100	✗	✓*
#7	H3C Magic NX15	00R012	✗	✗
#8	Honor X4 Pro	16.0.0.38	✗	✗
#9	Huawei AX3	4.0.0.19	✗	✗
#10	iKuai IX-Q3600	3.7.15	✗	✗
#11	Linksys	2.0.4.215745	✓	✓
#12	Mercury D191G	2.0.2	✓	✓
#13	Netgear AX5	1.0.8.82_1	✗	✓*
#14	PGY X4C-5131G	6.5.0	✗	✗
#15	Redmi AX3000	1.0.68	✓	✓
#16	Skyworth WR9651X	1.1.0	✓	✓
#17	Tenda V1	16.03.29.50	✓	✓
#18	TP-Link XDR3230	1.0.22	✓	✓
#19	TP-Link XDR5430	1.0.14	✓	✓
#20	Xiaomi 4C	5.15.150	✗	✗
#21	ZTE E2633	1.0.4	✓	✓
#22	DD-WRT	v3.0-r39296	✗	✗
#23	Gargoyle	1.14.0	✗	✗
#24	iKuai OS	3.7.17	✗	✓*
#25	libreCMC	1.5.15	✗	✗
#26	OpenWrt	23.05.3	✗	✗
#27	RouterOS	7.16.2	✗	✓*

✓: Yes. ✗: No. ✓: Vulnerable. ✗: Not vulnerable. No Q. Agg.: No query aggregation. *: Vulnerable due to poor randomization.

to our authoritative nameserver. We then observed whether the routers forward the ECS option and whether they aggregate queries.

6.1.3 Results. We found that all tested routers successfully supported forwarding the ECS option. However, we identified key differences regarding query aggregation. Specifically, 12 routers from ASUS, CISCO, D-Link, Fast, Linksys, Mercury, Redmi, Skyworth, Tenda, TP-Link, and ZTE did not support query aggregation. Furthermore, we found that Fiberhome, Netgear, iKuai OS, and RouterOS were vulnerable due to predictable source ports or TxIDs. These vulnerabilities rendered 16 routers to be exploitable by the REBIRTHDAY attack.

6.2 Public DNS Services

6.2.1 Public DNS Service List. Based on statistics from APNIC [3], we selected 45 widely-used public DNS services and their corresponding IP addresses for testing, as shown in Table 4.

6.2.2 Testing. The testing was conducted using our own domain and involved multiple test rounds. Additionally, we measured their rate-limiting behaviors that affected the attack success probability.

Table 4: Results of 45 Public DNS Services (14 Vulnerable).

Public DNS Service			ECS		No Query Aggregation		# of Rate-limit	Vulnerable
Index	Vendor	IP	Reply	Request	Without ECS	With ECS		
#1	114DNS	114.114.114.114	✗	✗	✗	✗	198	✗
#2	360 Secure DNS	101.226.4.6	✓	✗	✗	✗	150k+	✓*
#3	AdGuard DNS	94.140.14.14	✓	✓	✗	✓	2.7k	✓
#4	AhaDNS	5.2.75.75	✓	✓	✗	✗	11	✗
#5	Akamai Vantio DNS	23.56.160.142	✓	✗	✗	✗	10k+	✗
#6	Ali DNS	223.5.5.5	✓	✓	✗	✓	50k+	✓
#7	Alternate DNS	76.76.19.19	✓	✗	✗	✗	35	✗
#8	Baidu DNS	180.76.76.76	✗	✗	✗	✗	50k+	✗
#9	ByteDance DNS	180.184.1.1	✓	✓	✗	✓	1k+	✓
#10	CenturyLink DNS	205.171.3.66	✓	✗	✗	✗	2.5k	✗
#11	CIRA Shield DNS	149.112.121.10	✓	✓	✗	✓	50k+	✗
#12	Cisco OpenDNS	208.67.222.222	✓	✓	✗	✗	25k	✓*
#13	CleanBrowsing	185.228.168.10	✗	✓	✗	✗	400	✗
#14	CloudFlare DNS	1.1.1.1	✗	✓	✗	✗	50k+	✗
#15	CNNIC sDNS	1.2.4.8	✗	✗	✗	✗	10k+	✗
#16	Comodo Secure	8.26.56.10	✓	✓	✗	✗	20k+	✗
#17	Comss.one DNS	195.133.25.16	✗	✓	✓	✓	1k+	✓
#18	ControlD DNS	76.76.2.5	✓	✓	✗	✓	15k+	✓
#19	CZ.NIC ODVR	193.17.47.1	✗	✗	✗	✗	120	✗
#20	DNS for Family	94.130.180.225	✓	✓	✓	✓	5k+	✓
#21	DNS Forge	176.9.93.198	✗	✓	✗	✗	1k	✗
#22	DNS.SB	185.222.222.222	✗	✗	✗	✗	50k+	✗
#23	DNS.WATCH	84.200.69.80	✓	✗	✗	✗	5k+	✗
#24	DNSPod Public DNS+	119.28.28.28	✓	✓	✗	✓	50k+	✓
#25	Dyn DNS	216.146.35.35	✗	✓	✗	✗	600	✗
#26	FDN DNS	80.67.169.12	✓	✓	✗	✗	10k+	✗
#27	G-Core DNS	95.85.95.85	✓	✓	✓	✓	10k+	✓
#28	Google DNS	8.8.8.8	✓	✓	✓	✓	500	✓
#29	Hurricane DNS	74.82.42.42	✗	✗	✗	✗	1k	✗
#30	Level3 DNS	4.2.2.1	✗	✗	✗	✗	3.5k	✓*
#31	LibreDNS	88.198.92.222	✗	✗	✗	✗	10k+	✗
#32	Neustar UltraDNS	156.154.70.1	✓	✓	✗	✗	20k+	✗
#33	NextDNS	45.90.30.118	✗	✗	✗	✗	2.5k	✗
#34	Norton ConnectSafe	199.85.126.10	✓	✗	✗	✗	20k+	✗
#35	OneDNS	52.80.66.66	✓	✓	✗	✗	540	✗
#36	OpenNIC DNS	103.1.206.179	✓	✓	✗	✗	15k+	✗
#37	Quad101 DNS	101.101.101.101	✓	✗	✗	✗	220	✗
#38	Quad9 DNS	9.9.9.11	✓	✓	✗	✓	5k+	✓
#39	SafeDNS	195.46.39.39	✗	✗	✗	✗	15k+	✗
#40	SafeSurfer DNS	104.197.28.121	✗	✗	✗	✗	3k+	✗
#41	SkyDNS	193.58.251.251	✗	✗	✗	✗	15k+	✗
#42	Strongarm DNS	52.3.100.184	✓	✗	✗	✗	150	✗
#43	Tiarap Public DNS	174.138.21.128	✗	✗	✗	✗	500	✗
#44	Verisign Public DNS	64.6.65.6	✓	✗	✗	✗	10k+	✗
#45	Yandex DNS	77.88.8.1	✓	✗	✓	✓	2.5k	✓

✓: Yes. ✗: No. ✓: Vulnerable. ✗: Not vulnerable. *: Vulnerable due to poor query aggregation. # of Rate-limit is per IP.

6.2.3 *Results.* Our results showed that 32 services supported ECS. Specifically, 10 of these services only supported returning ECS to the client, 5 only supported querying upstream with ECS, and 17 supported both querying and responding with ECS. We also found that 11 services did not implement query aggregation that were vulnerable to REBIRTHDAY. Among these, 8 services, including AdGuard DNS, Ali DNS, ByteDance DNS, CIRA Shield DNS, Comss.one DNS, ControlD DNS, DNSPod Public DNS+, and Quad9

DNS, were affected due to their vulnerable ECS implementations. Besides, 3 services, including 360 Secure DNS, Cisco OpenDNS, and Level3 DNS, were vulnerable because of poor query aggregation (querying 25 - 100 times). The inconsistent multiple queries from resolvers are likely due to retry mechanisms for performance optimization or handling network issues. Specifically, only CIRA Shield DNS performs client IP and subnet verification; exploiting it would require attackers to spoof the client's IP address. These vulnerabilities significantly increased the risk of the DNS Birthday attack. These findings demonstrate the critical impact of ECS in amplifying the risks of the REBIRTHDAY attack.

6.3 Open DNS Resolvers

6.3.1 Open DNS Resolver List. Since the open DNS resolver is highly volatile [60], we aimed to acquire the most recent snapshot of the Internet by scanning the IPv4 UDP port 53 on our controlled domain using XMap [39]. We discarded any results that were erroneous. Between October and December 2024, we discovered over 2.4 million open DNS resolvers, which are associated with 232 regions and 25,711 autonomous systems (ASes). The top three regions with the highest number of resolvers are China, India, and Russia.

6.3.2 Testing. Instead of conducting actual attacks, we focused on testing whether the resolvers support ECS and query aggregation. For ethical considerations and based on the rate-limiting results from public DNS services, we sent 200 queries for the same domain in 1s. We then observed queries from our authoritative nameserver.

6.3.3 Results. Among the 2.4 million open DNS resolvers, approximately *14.1%* (*343,133*) supported querying the authoritative nameserver with the ECS option, while about *29.0%* (*705,176*) returned the ECS option to the client. Around *5.2%* (*127,561*) supported forwarding ECS to both clients and upstream resolvers. The results of the query aggregation tests are shown in Figure 9. Upon receiving 200 queries for the same domain, over *80%* of resolvers exhibited no significant query aggregation (less than three queries). Specially, more than *15%* of resolvers made 25 or more queries, and over *10%* made 50 or more queries. According to the Birthday Paradox, resolvers that queried 25 times would have a 50% success rate after 1,800 rounds of attacks, while those querying 50 times could reach a 50% success rate after only 900 rounds. For resolvers supporting ECS, the impact was even more significant. Over *30%* of ECS-supporting resolvers made 25 or more queries, and approximately *20%* made 50 or more queries. Consequently, **the support for ECS significantly increased the number of resolvers (by at least *100K*) vulnerable to Birthday attacks.** We conclude that at least *365K* (*15%*) open DNS resolvers are vulnerable to Birthday attacks due to no query aggregation.

7 Discussion and Mitigation

Ethical Considerations. We follow the ethical guidelines of the Menlo Report [30] and best network measurement practices [55]. Firstly, we install all DNS software on our local machines. For the public DNS services, we limit queries to avoid exceeding their rate limits ($\leq$ 1K for rate-limits over 150K). For measurements, we cap the scanning rate at 5K pps to minimize network impact and perform random-enumerating scans with our own domains, identifying
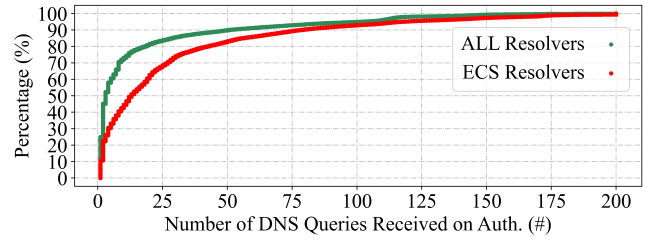

Figure 9: The Fraction of Query Aggregation Results.

a lower thresholds (200). We configure the PTR record and a website to show our research objectives, and no opt-out requests have been received. Finally, we report findings to all vendors. Notably, Quad9 actively provided services for us to test.

Mitigation Solutions. REBIRTHDAY exploits vulnerabilities in the DNS Extension protocol, particularly with ECS processing. The root cause lies in RFC 7871, which allows responses without ECS to be considered valid. To mitigate this attack, we propose an enhanced query aggregation strategy and urge additional defenses against cache poisoning. Firstly, we recommend that resolvers verify ECS consistency in responses, marking authoritative nameservers as invalid if there is no ECS or the Scope Prefix-Length is 0. Resolvers should aggregate all subsequent queries containing ECS subnets for that server and domain, and also aggregate queries within the same ECS scope to ensure consistency. Secondly, to defend against cache poisoning, we urge enabling 0x20 encoding [16] to thwart case manipulation, and implementing DNSSEC [4] to cryptographically sign DNS responses. Finally, resolvers could block malicious response injections through anomaly detection and rate limiting like PowerDNS [56], further securing the DNS system.

Disclosure. We reported vulnerabilities to affected vendors and are discussing solutions with them. 8 vendors have confirmed REBIRTHDAY, including BIND, Unbound, PowerDNS, Technitium, Dnsmasq, YogaDNS, Quad9, and Adguard. BIND, Unbound, and PowerDNS have implemented a patched version based on our suggestions. They will resend the DNS query without ECS or switch to TCP when an ECS mismatch is detected. 50 CVE-ids were assigned.

8 Conclusion

In this study, we present a novel DNS poisoning attack model, REBIRTHDAY, which exploits vulnerabilities in the poorly implemented ECS mechanism. This flaw bypasses the query aggregation policy designed to prevent DNS Birthday attacks, making DNS cache poisoning feasible once again. A solution involves enforcing strict ECS verification between DNS queries and responses. Besides, **we found other extension-based risks and are conducting a comprehensive analysis of DNS extensions to identify new weaknesses as our future work.**

Acknowledgments

References

[1] Fatemah Alharbi, Jie Chang, Yuchen Zhou, Feng Qian, Zhiyun Qian, and Nael B. Abu-Ghazaleh. Collaborative Client-Side DNS Cache Poisoning Attack. In *Proceedings of 2019 IEEE Conference on Computer Communications (INFOCOM '19)*, 2019.

[2] Sumayah Alrwais, Xiaojing Liao, Xianghang Mi, Peng Wang, XiaoFeng Wang, Feng Qian, Raheem Beyah, and Damon McCoy. Under the Shadow of Sunshine: Understanding and Detecting Bulletproof Hosting on Legitimate Service Provider Networks. In *Proceedings of 2017 IEEE Symposium on Security and Privacy (S&P '17)*, 2017.

[3] APNIC. DNS Resolvers Use. https://stats.labs.apnic.net/rvrs, 2023.

[4] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose. RFC 4033: DNS Security Introduction and Requirements. *RFC Proposed Standard*, 2005.

[5] Steven M. Bellovin. Using the Domain Name System for System Break-ins. In *Proceedings of the 5th USENIX Security Symposium (USENIX Security '95)*, 1995.

[6] Markus Brandt, Tianxiang Dai, Amit Klein, Haya Shulman, and Michael Waidner. Domain Validation++ For MitM-Resilient PKI. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, 2018.

[7] CAIDA. State of IP Spoofing. https://spoofer.caida.org/summary.php, 2025.

[8] CERT/CC. 1996 CERT Advisories. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=496170, 1996.

[9] CERT/CC. 1997 CERT Advisories. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=496174, 1997.

[10] CERT/CC. 2001 CERT Advisories. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=496190, 2001.

[11] Kenjiro Cho, Kensuke Fukuda, Vivek Pai, Neil Spring, Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *Proceedings of the 2015 ACM Internet Measurement Conference (IMC '15)*, 2015.

[12] Cloudflare. What is DNS cache poisoning? | DNS spoofing. https://www.cloudflare.com/learning/dns/dns-cache-poisoning/, 2025.

[13] CNET. AlterNIC Takes over InterNIC Traffic. https://www.cnet.com/tech/mobile/alternic-takes-over-internic-traffic/, 1997.

[14] Carlo Contavalli, Wilmer van der Gaast, David C Lawrence, and Warren Kumari. RFC 7871: Client Subnet in DNS Queries. *RFC Informational*, 2016.

[15] David Dagon, Manos Antonakakis, Kevin Day, Xiapu Luo, Christopher P. Lee, and Wenke Lee. Recursive DNS Architectures and Vulnerability Implications. In *Proceedings of the 16th Annual Network and Distributed System Security Symposium (NDSS '09)*, 2009.

[16] David Dagon, Manos Antonakakis, Paul Vixie, Tatuya Jinmei, and Wenke Lee. Increased DNS Forgery Resistance through 0x20-bit Encoding: Security via Leet Queries. In *Proceedings of the 15th ACM Conference on Computer and Communications (CCS '08)*, 2008.

[17] Joao Damas, Michael Graff, and Paul Vixie. RFC 6891: Extension Mechanisms for DNS (EDNS(0)). *RFC Best Current Practice*, 2013.

[18] Elias Heftrig, Haya Shulman, and Michael Waidner. Downgrading DNSSEC: How to Exploit Crypto Agility for Hijacking Signed Zones. In *Proceedings of the 32nd USENIX Security Symposium (USENIX Security '23)*, 2023.

[19] Amir Herzberg and Haya Shulman. Security of Patched DNS. In *Proceedings of the 17th European Symposium on Research in Computer Security (ESORICS '12)*, 2012.

[20] Amir Herzberg and Haya Shulman. Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org. In *Proceedings of the 1st IEEE Conference on Communications and Network Security (CNS '13)*, 2013.

[21] Amir Herzberg and Haya Shulman. Socket Overloading for Fun and Cache-Poisoning. In *Proceedings of the 29th Annual Computer Security Applications Conference (ACSAC '13)*, 2013.

[22] Amir Herzberg and Haya Shulman. Vulnerable Delegation of DNS Resolution. In *Proceedings of the 18th European Symposium on Research in Computer Security (ESORICS '13)*, 2013.

[23] Allen Householder and Ian A Finlay. Various DNS Service Implementations Generate Multiple Simultaneous Queries for the Same Resource Record. https://www.kb.cert.org/vuls/id/457875, 2002.

[24] Shumon Huque, Paul Vixie, and Ralph Dolmans. Draft: Delegation Revalidation by DNS Resolvers. *RFC Draft*, 2022.

[25] ICANN. Centralized Zone Data Service. https://czds.icann.org/, 2023.

[26] Philipp Jeitner and Haya Shulman. Injection Attacks Reloaded: Tunnelling Malicious Payloads over DNS. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security '21)*, 2021.

[27] Philipp Jeitner, Haya Shulman, Lucas Teichmann, and Michael Waidner. XDRI Attacks - and - How to Enhance Resilience of Residential Routers. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security '22)*, 2022.

[28] Dan Kaminsky. Black Ops of TCP/IP 2005. https://www.blackhat.com/presentations/bh-jp-05/bh-jp-05-kaminsky/bh-jp-05-kaminsky.pdf, 2005.

[29] Dan Kaminsky. It's the End of the Cache as We Know It. https://www.blackhat.com/presentations/bh-jp-08/bh-jp-08-Kaminsky/BlackHat-Japan-08-Kaminsky-DNS08-BlackOps.pdf, 2008.

[30] Erin Kenneally and David Dittrich. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *SSRN Electronic Journal*, 2012.

[31] Amit Klein. BIND 8 DNS Cache Poisoning. *Trusteer*, 2007.

[32] Amit Klein. BIND 9 DNS Cache Poisoning. *Trusteer*, 2007.

[33] Amit Klein. OpenBSD DNS Cache Poisoning and Multiple O/S Predictable IP ID Vulnerability. *Trusteer*, 2007.

[34] Amit Klein. Windows DNS Server Cache Poisoning. *Trusteer*, 2007.

[35] Amit Klein. PowerDNS Recursor DNS Cache Poisoning. *Trusteer*, 2008.

[36] Amit Klein. DNS Record Injection Vulnerabilities in Home Routers. http://www.icir.org/mallman/talks/schomp-dns-security-nanog61.pdf, 2014.

[37] Amit Klein. Cross Layer Attacks and How to Use Them (for DNS Cache Poisoning, Device Tracking and More). In *Proceedings of 2020 IEEE Symposium on Security and Privacy (S&P '21)*, 2021.

[38] Xiang Li, Baojun Liu, Xuesong Bai, Mingming Zhang, Qifan Zhang, Zhou Li, Haixin Duan, and Qi Li. Ghost Domain Reloaded: Vulnerable Links in Domain Name Delegation and Revocation. In *Proceedings of the 30th Annual Network and Distributed System Security Symposium (NDSS '23)*, 2023.

[39] Xiang Li, Baojun Liu, Xiaofeng Zheng, Haixin Duan, Qi Li, and Youjun Huang. Fast IPv6 Network Periphery Discovery and Security Implications. In *Proceedings of the 2021 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '21)*, 2021.

[40] Xiang Li, Chaoyi Lu, Baojun Liu, Qifan Zhang, Zhou Li, Haixin Duan, and Qi Li. The Maginot Line: Attacking the Boundary of DNS Caching Protection. In *Proceedings of the 32nd USENIX Security Symposium (USENIX Security '23)*, 2023.

[41] Xiang Li, Dashuai Wu, Haixin Duan, and Qi Li. DNSBomb: A New Practical-and-Powerful Pulsing DoS Attack Exploiting DNS Queries-and-Responses. In *Proceedings of 2025 IEEE Symposium on Security and Privacy (S&P '24)*, 2024.

[42] Xiang Li, Wei Xu, Baojun Liu, Mingming Zhang, Zhou Li, Jia Zhang, Deliang Chang, Xiaofeng Zheng, Chuhan Wang, Jianjun Chen, Haixin Duan, and Qi Li. TuDoor Attack: Systematically Exploring and Exploiting Logic Vulnerabilities in DNS Response Pre-processing with Malformed Packets. In *Proceedings of 2025 IEEE Symposium on Security and Privacy (S&P '24)*, 2024.

[43] Baojun Liu, Chaoyi Lu, Hai-Xin Duan, Ying Liu, Zhou Li, Shuang Hao, and Min Yang. Who Is Answering My Queries: Understanding and Characterizing Interception of the DNS Resolution Path. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security '18)*, 2018.

[44] Keyu Man, Zhiyun Qian, Zhongjie Wang, Xiaofeng Zheng, Youjun Huang, and Haixin Duan. DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*, 2020.

[45] Keyu Man, Xin'an Zhou, and Zhiyun Qian. DNS Cache Poisoning Attack: Resurrections with Side Channels. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21)*, 2021.

[46] MathWorld. Birthday Attack. https://mathworld.wolfram.com/BirthdayAttack.html, 2025.

[47] MathWorld. Birthday Problem. https://mathworld.wolfram.com/BirthdayProblem.html, 2025.

[48] Xianghang Mi, Xuan Feng, Xiaojing Liao, Baojun Liu, XiaoFeng Wang, Feng Qian, Zhou Li, Sumayah Alrwais, Limin Sun, and Ying Liu. Resident Evil: Understanding Residential IP Proxy as a Dark Service. In *Proceedings of 2019 IEEE Symposium on Security and Privacy (S&P '19)*, 2019.

[49] Paul V. Mockapetris. RFC 1034: Domain Names - Concepts and Facilities. *RFC Internet Standard*, 1987.

[50] Paul V. Mockapetris. RFC 1035: Domain Names - Implementation and Specification. *RFC Internet Standard*, 1987.

[51] NIST. CVE-1999-0024. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0024, 1997.

[52] NIST. CVE-2000-0335. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0335, 2000.

[53] NIST. CVE-2002-2211. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-2211, 2002.

[54] NIST. CVE-2008-1447. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1447, 2008.

[55] Craig Partridge and Mark Allman. Ethical Considerations in Network Measurement Papers. *Communications of the ACM*, 2016.

[56] PowerDNS. spoof-nearmiss-max. https://docs.powerdns.com/recursor/settings.html#spoof-nearmiss-max, 2025.

[57] RIPE. RIPE Atlas. https://atlas.ripe.net/, 2025.

[58] RouterChart. Popular Routers. https://routerchart.com/brands, 2023.

[59] Vagner Sacramento. Vulnerability in Requests Control of BIND Versions 4 and 8 Allows DNS Spoofing. https://lists.isc.org/pipermail/bind-users/2002-November/043141.html, 2002.

[60] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. On Measuring the Client-side DNS Infrastructure. In *Proceedings of the 2013 Internet Measurement Conference (IMC '13)*, 2013.

[61] Christoph Schuba and Eugene H Spafford. Addressing Weaknesses in the Domain Name System Protocol. Master's thesis, Purdue University, 1993.

[62] SecureList. Massive DNS Poisoning Attacks in Brazil. https://securelist.com/massive-dns-poisoning-attacks-in-brazil/31628/, 2011.

[63] Haya Shulman and Michael Waidner. Fragmentation Considered Leaking: Port Inference for DNS Poisoning. In *Proceedings of the 12th International Conference on Applied Cryptography and Network Security (ACNS '14)*, 2014.

[64] Sooel Son and Vitaly Shmatikov. The Hitchhiker's Guide to DNS Cache Poisoning. In *Proceedings of the 6th International ICST Conference on Security and Privacy in Communication Systems (SecureComm '10)*, 2010.

[65] systemd. systemd-resolved.service and VPNs. https://systemd.io/RESOLVED-VPNS/, 2025.

[66] Paul Vixie. DNS and BIND Security Issues. In *Proceedings of the 5th USENIX Security Symposium (USENIX Security '95)*, 1995.

[67] Paul Vixie. RFC 2671: Extension Mechanisms for DNS (EDNS(0)). *RFC Proposed Standard*, 1999.

[68] Wikipedia. List of Router Firmware Projects. https://en.wikipedia.org/wiki/List_of_router_firmware_projects, 2023.

[69] Wei Xu, Xiang Li, Chaoyi Lu, Baojun Liu, Jia Zhang, Jianjun Chen, Tao Wan, and Haixin Duan. TsuKing: Coordinating DNS Resolvers and Queries into Potent DoS Amplifiers. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)*, 2023.

[70] Michal Zalewski. Strange Attractors and TCP/IP Sequence Number Analysis. *RAZOR/BindView Corporation*, 2001.

[71] Mingming Zhang, Xiang Li, Baojun Liu, Jianyu Lu, Jianjun Chen, Yiming Zhang, Xiaofeng Zheng, Haixin Duan, and Shuang Hao. DareShark: Detecting and Measuring Security Risks of Hosting-Based Dangling Domains. In *Proceedings of the 2023 ACM Special Interest Group on Measurement and Evaluation (SIGMETRICS '23)*, 2023.

[72] Qifan Zhang, Xuesong Bai, Xiang Li, Haixin Duan, Qi Li, and Zhou Li. ResolverFuzz: Automated Discovery of DNS Resolver Vulnerabilities with Query-Response Fuzzing. In *Proceedings of the 33rd USENIX Security Symposium (USENIX Security '24)*, 2024.

[73] Xiaofeng Zheng, Chaoyi Lu, Jian Peng, Qiushi Yang, Dongjie Zhou, Baojun Liu, Keyu Man, Shuang Hao, Haixin Duan, and Zhiyun Qian. Poison Over Troubled Forwarders: A Cache Poisoning Attack Targeting DNS Forwarding Devices. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security '20)*, 2020.