Zone-Hopping: Sensitive Information Leakage Prevention for DNSSEC-NSEC

Fatema Bannat Wala
ESnet
Lawrence Berkeley National Lab
Berkeley, USA
fatemabw@es.net

Stephan Bohacek
Electrical And Computer Engineering
University of Delaware
Newark, USA
bohacek@udel.edu

***Abstract*—DNSSEC (DNS Security Extension) was developed to address significant security integrity flaws in DNS. We explore information leakage stemming from a known DNSSEC vulnerability that facilitates a process known as zone walking, which enables the efficient collection of all FQDNs. This paper is divided into two main sections. First, we show that the information leaked by zone walking is sometimes private and should be protected. We demonstrate this by identifying instances where private information is disclosed through zone-walking, emphasizing that zone walking is an effective replacement for network scanning in IPv6 networks, and that FQDNs often constitute de facto private information. This last point is addressed by comparing data collected from DomainTools' passive DNS database with data collected from zone walking. We found that more than 50% of FQDNs found from zone walking are not in the passive DNS dataset. This indicates that these FQDNs are not used publicly, despite being accessible on publicly available DNS servers. In the second part of this paper, we propose and test a novel and practical approach - Zone-Hopping (ZH) to mitigate this information leakage issue, all the while preserving the integrity of the DNS/DNSSEC protocol. In some cases, ZH can be implemented through only small changes to a configuration file. Moreover, in contrast to other solutions, the proposal solution does not impose any extra computational load on authoritative DNS servers, and hence, does not open new avenues for DoS attacks.**

***Index Terms*—DNSSEC, Zone-walking, Security, information leakage**

I. Introduction

DNS, one of the oldest and most important protocols developed by the internet community, supports the World Wide Web and access to some critical services such as web browsing, email, VPN (Virtual Private Network), IoT applications, Instant messaging, etc. However, it was not developed with security and privacy as the primary objectives, and hence several new protocols have been designed to enhance DNS security. One of such protocol, DNSSEC (DNS Security Extension protocol), was publish in 1999 [1]. DNSSEC ensures the integrity of DNS messages, specifically protecting against malicious or forged answers. This is achieved by adding digital signatures to each DNS record hosted by the authoritative DNS server. To this end, DNSSEC introduces some new record types: RRSIG (digital Resource Record SIGnature), DNSKEY (public key record), DS (Delegation Signer, to establish chain of trust), NSEC (Next Secure, proof of nonexistence), NSEC3 (Hashed Next Secure) [2]. The signatures use public key cryptography to provide verifiable DNS messages. Authoritative name servers generate a key pair; the private key signs each resource record in a zone file, thus creating a digital signature (RRSIG), and the corresponding public key is then published in a DNSKEY record, which is also signed with the private key and accompanied by its own RRSIG record. The chain of trust is completed by the DS record, which contains verifiable information generated from the public key and uploaded is to the parent zone. The parent zone publishes its DS record to its parent; this hierarchical sequence of DS records continues up to the root server. The resolver only needs to trust one public key, known as the "trust-anchor", which is typically the root server's public keys. In a properly DNSSEC-enabled name server, each DNS resource record is paired with a corresponding RRSIG record.

A primary motivation for DNSSEC is to ensure that the resolved IP address is indeed the IP address of the requested domain. Thus, DNSSEC offers a significant advantage over DNS, which is vulnerable to integrity attacks [3] [4] [5]. Additionally, DNSSEC protects against "denial of existence" attacks. In such attacks, an attacker could maliciously generate a reply to a DNS query, falsely indicating that the requested domain does not exist. The NSEC record was introduced to authenticate the non-existence of a domain, thereby providing an authenticated denial of existence. Specifically, when a client requests a non-existent domain, an NSEC record is returned in response. This record lists the domains that are lexicographically adjacent to the queried domain – the one immediately preceding it (NSEC's owner name) and the one immediately following (next available name). This record proves that no names exist in the "span" between the NSEC's owner name and the next available name. Thus, by validating the NSEC record's RRSIG, the client can verify the authenticity of the response, confirming the non-existence of the domain.

The impact of a successful denial of existence attack is that the client is unable to resolve the requested domain name. This result can be also achieved through a standard DoS attack that disables the authoritative server or the DNS resolver. On the other hand, DNSSEC's NSEC implementation allows information leakage through a process known as zone walking [6]. Specifically, zone walking allows all fully

2833-292X/24/$31.00 ©2024 IEEE
DOI 10.1109/DSN-S60304.2024.00035

qualified domain names to be identified within a domain. That is, while protecting against denial of existence attack is beneficial, the approach used by DNSSEC comes with new security issues that, in some case, might outweigh the benefits provided by such protection. This paper explores this trade-off. The important contributions are as follows.

- We clarify the importance of protecting FQDNs. First, we argue that FQDNs sometimes contain private information such as employee names and descriptive strings (e.g., VPN and CAM) that can assist attackers. Second, in IPv6 networks where scanning is infeasible, zone walking facilitates reconnaissance that can assist attackers penetrate the network and move laterally. And third, most FQDNs are de facto private information. By this we mean that without zone walking, it is difficult to determine FQDNs. To demonstrate this, we examine the data collected from a vast network of passive sensors that collect DNS queries. Our analysis shows that less than half of the FQDNs identified via zone walking are also detected through passive sensing. This indicates that the majority of these FQDN are not used in the Internet. Thus, barring the existence of zone walking, these FQDNs are private information.
- We propose and develop Zone-Hopping (ZH), a novel solution that prevents the exposure of sensitive information through zone walking. Implementing this solution is straightforward. For instance, in BIND9, it merely requires a change in the configuration files.

Following the Related Work section, this paper is divided into two main parts. In the first part, we demonstrate how zone walking constitutes a critical source of information leakage. In the second part, we introduce a straightforward, easily implementable solution that enables network administrators to decide between protection against denial of existence attacks and information leakage on a per-FQDN basis. This solution, when added to existing solutions, results in a complete spectrum of solutions. This suite of solutions enables the network administrator to balance protections against zone walking, denial of existence, and, as will be discuss, susceptibility to DoS attacks. The vulnerability to DoS attacks stems from an increased computational load on the authoritative server when handling non-existent FQDNs.

II. Related Work

When DNSSEC was first deployed in 2005, the challenge of proving that a domain doesn't exist was resolved by introducing NSEC records, which have owner name and next available domain name in a lexicographically sorted zone file. The NSEC records introduced the ability to zone walk a domain [6]. To prevent this information leakage, NSEC3 records were introduced in early 2008, as defined in RFC 5155 [7]. Instead of returning the plain-text owner name and next available domain name in NSEC record, NSEC3 hashed the values of the owner name and the next existing domain name. This solves the problem of retrieving the clear text names from an NSEC record. However, the lexicographically preserving hash used in NSEC3 is susceptible to the hash-cracking attacks [8] [9] . For example, in [10], the authors demonstrate that NSEC3 hashes can be cracked with a success rate of 88%.

RFC 4470 [11] was introduced, in early 2006, to solve the zone walking problem. This RFC introduced the concept of "Minimally Covering NSEC Records and DNSSEC Online Signing", which describes a way to construct DNSSEC NSEC resource records that cover a smaller range of names. According to the RFC - "Whenever an NSEC record is needed to prove the non-existence of a name, a new NSEC record is dynamically produced and signed. The new NSEC record has an owner name lexically before the QNAME (queried name) but lexically following any existing name and a "next name" lexically following the QNAME but before any existing name." The new NSEC record hence generated would still cover the non-existent query, but with the fake previous-name and the next name, effectively preventing the disclosure of zone contents. There were two adaptations found that were based on RFC 4470:

NSEC3 White Lies, [12], was introduced in 2014 for prevention of zone walking specifically in NSEC3, where fake NSEC3 records are generated on-the-fly that surround the requested name, as described in RFC 4470. The new NSEC3 record will comprise of the hash of the QNAME minus one as owner name and the hash of the QNAME plus one as the next name and therefore existing records that span the NSEC3 hash of the requested QNAME are not disclosed. White lies also requires online-signing of the newly created fake NSEC3 records on the request basis.

Black Lies (BL), described in RFC draft [13] in 2016, is implemented by utilizing a different secure negative response "NODATA" instead of "NXDOMAIN". When a client queries for a non-existence domain, an NSEC record is dynamically created where the owner name is the same as the QNAME, and the next name as the immediate lexicographic successor of the QNAME (which is generated on-the-fly and doesn't match any existing domain names in the zone). The response message have RCODE NOERROR, as opposed to NXDOMAIN, since a record matching the QNAME is being returned (of type NSEC). This method requires online-signing, as oppose to the offline signing in NSEC and NSEC3 implementations. This approach provides protection against zone-walking while still providing integrity of FQDN to IP mapping. Consequently, this method is used by several DNS service providers such as Cloudflare [14] and Amazon Route 53 [15].

As mentioned in RFC 4470 section 5 [11], there are some security risks associated with the methods described above. First, in order to sign NSEC records on-the-fly, the private key needs to be available on the internet-accessible zone's authoritative servers. Any unintended disclosure of the private key can compromise the whole zone. Second, generating signatures of NSEC records is computationally expensive and makes authoritative servers vulnerable to a denial of service (DoS) attacks. As mentioned, NSEC was added specifically to resolve denial of existence attacks and therefore ensure client's requested domain is resolved. Ironically, these online signing

approaches increase the computational load on the servers and therefore opens an avenue for DoS, which have the same impact as the attack that NSEC was designed to solve. In fact, AWS Route 53 documentation discusses that DNS queries will fail when zone walking behavior is detected [16]. While the documentation does not specify why throttling is requires, we can assume that this throttling is a consequence of the high computational load required to generate signed NSEC records on-the-fly.

The AWS documentation highlights a significant problem with the state-of-the-art: BL seeks to protect against zone walking while preserving defense against denial of existence attack. However, this defense against denial of existence attacks opens a vulnerability to DoS, which has the same impact as denial of existence. Moreover, denial of existence attacks require a MitM attack, which are difficult to execute, while a DoS attack on a BL enabled DNS server is a significantly less sophisticated attack. This demonstrates the critical need for an alternative solution such as the one proposed here that relies on offline signing, and therefore does not increase susceptibility to DoS attack or private key exposure, and also protects against information leakage.

III. Data Leakage from Zone Walking

This section discusses how zone walking allows information be leaked and why this leakage is a security risk. First, we describe zone walking and demonstrate the ease at which DNSSEC enables complete reconnaissance of the domain's FQDNs. Second, we discuss why this information leakage is a critical security leak in IPv6 networks. Third, we demonstrate the incorrectness of the common belief that domain names are de facto public information. Specifically, we find that domain names are, in fact, private.

A. Zone Walking

Zone walking is a straightforward process that utilizes NSEC records [17]. The zone walker generates a DNSSEC query for a random FQDN under the desired apex domain. For example, if the domain example.com is to be walked, we might start with requesting the DS for \000.example.com. Assuming this domain does not exist, the DNSSEC server will reply with an NSEC record that include the next existing FQDN and the preceding existing FQDN from the requested FQDN. Here, "next" and "preceding" are based on lexicographic order. Therefore, two existing domains have been found. Suppose the next FQDN is a.example.com. The zone walker then queries for a\000.example.com, which is a string that is slightly after the just found a.example.com. The DNSSEC server replies with the preceding FQDN (which is a.example.com) and the next existing FQDN. The zone walker continues following this process until the next existing FQDN is the apex domain (SOA), signaling that there is no next FQDN.

Several reconnaissance tools, such as Kali Linux [18] and DNSRecon [19], include built-in functionality to enumerate an entire zone through zone walking. Additionally, there is open-source software, such as 'ldns' [20], that can be used for zone walking.

It is important to highlight the efficiency of zone walking, as it enables the retrieval of N Fully Qualified Domain Names (FQDNs) with approximately N DNS queries. Consequently, methods to block zone walking based on detecting multiple queries for non-existing FQDNs (e.g., [21]) would likely fail to stop a stealthy zone walker from enumerating the entire domain.

B. IPv6 Address Discovery

For IPv6 networks, zone-walking is a critical tool for reconnaissance [22] [23] [24]. According to the Office of Management and Budget (OMB) memorandum M-21-07, issued in Nov. 2020, the Federal government plans to transition to IPv6 only networks by 2025 [25]. An important benefit of IPv6 is that network reconnaissance through scanning is infeasible since the address space is so large that scanning takes too long unless massive network traffic is generated, which is easily detected. For instance, the private address range is fc00::/7, allowing private addresses to be dispersed randomly within an address space of 2^{121} addresses. However, zone walking provides effective way to find all named hosts. Therefore, disabling zone walking will disable a crucial tool employed in malicious IPv6 network reconnaissance [22] [23].

C. The Privacy of Fully Qualified Domain Names

As previously discussed, a trade-off exists between protection against denial of existence attacks and protection against zone walking. Proponents of prioritizing defense against denial of existence attacks commonly argue that zone walking merely discloses information that is already public. This section will explore two facets of this argument.

The first component of this argument is that no confidential information is stored in FQDNs. However, through zone walking we have discovered that hosts such as laptops and personal workstations have names similar to <employee first name>-<employee last name>.example.com. Consequently, zone walking exposes the company's employee directory. Other types of information revealed includes names with descriptive sub-strings such as "dhcp.", "dev-", "test-" "cctv.", "VPN" and "cam." Note that these names are leaked even though, in many cases, the hosts are not reachable from the public Internet. That is, zone walking allows reconnaissance of hosts even when hosts are not reachable.

A second aspect of the argument that zone walking is permissible is the assertion that the information available via zone walking is public information and can be easily reconstructed using various available data sources such as Passive DNS (PDNS) [26]. However, through our analysis, we found that even extensive passive listening fails to retrieve most FQDNs. That is, while these FQDNs are stored on publicly accessible authoritative domain servers, these FQDNs were not queried by public machines. Therefore, they could not be determine through passive listening. Consequently, we argue that these FQDNs are essentially private, and zone

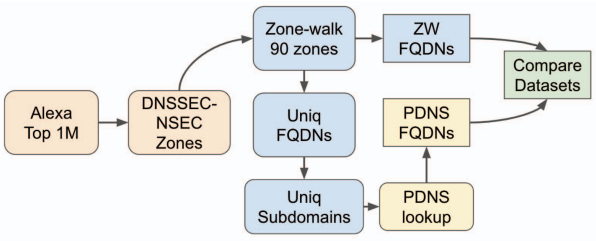

Fig. 1. Data collection pipeline

walking provides a straightforward method for uncovering this otherwise inaccessible information.

To assess whether passive listening could result in the same information as zone walking, we compared data from zone walking with data from a well-known PDNS data source, specifically DomainTools' PDNS database known as DNSDB [27]. It is important to note that we are not investigating DNSDB as a security threat. On the contrary, DNSDB implements several security measures. Our objective is to determine whether an attacker, by building a network of passive DNS sensors that mimics DNSDB, could identify all FQDNs or if some FQDNs would remain undiscovered.

1) Data Collection via Zone Walking: To perform zone-walking, we utilized the open-source tool called "ldns-walk" [20]. This tool requires a zone name (e.g., example.com) as input and the -f option to perform a full zone walk. We initiated our study with the first 25,000 second-level domains from the Alexa Top 1 Million domains list. Among these, 90 were found to be using DNSSEC-NSEC; and remainder either failed, were not using DNSSEC (19,064), or were using BL by AWS Route 53 or Cloudflare (151). We categorized those 90 domains into three categories: Government (34 zones), Academia (15 zones), and Other (41 zones). Overall 561,332 unique FQDNs were harvested via zone-walking on the 90 zones, with 458,227 unique subdomains. The data collection period via ldns was around September 2023. The data pipeline is illustrated in Figure 1.

2) Data Collection via PDNS: Passive DNS is a technique used to record the DNS responses collected across the Internet through sensors place between the recursive resolvers and authoritative resolvers. DomainTools, a commercial organization, acquired Farsight and now offers PDNS datasets under the named DNSDB as a commercial solution. The PDNS dataset is collected from "far more than 400" organizations around the world [28]. As a courtesy and to support advance research, we have been granted access to DNSDB.

While 90 zones are identified using zone walking, we were unable to lookup all zones using DNSDB. This limitation arises because DNSDB aims to prevent malicious use by restricting wildcard searches, such as *.example.com. Nevertheless, DNSDB permits wildcard searches for subdomains, like *.apis.example.com. However, some zones have large number of subdomains such as XXX.example.com. For example, stanford.edu has 174,660 subdomains such as *.XXX.stanford.edu.

TABLE I
COMPARISON OF PDNS AND ZW FQDNS

Category	Zones	Subdomains	ZW FQDNs	PDNS FQDNs	pdns Coverage
Gov.	34	30,552	51,845	29,129	56.18%
Academia	14	101,680	147,115	77,265	52.52%
Other	40	45,145	152,654	35,958	23.55%
Total	88	177,377	351,614	142,352	40.48%

Our access to DNSDB only permitted a limited number of queries. Therefore, we restricted our attention to 88 zones that had a more manageable number of third level domains and were able to query a total of 351,614 FQDNs (a total of 177,377 subdomains) across these zones, as shown in Table I.

3) Results: Comparing the DNSDB results to the zone walking results across the 88 zones, we found less than 50% of coverage overall, with 56.18%, 52.52% and 23.55% coverage in individual categories, as shown in Table I. These results show that data exposed through zone walking is not available from passive listening systems. The fact that zone walking detects far more FQDNs than PDNS could be attributed to the size of PDNS data collection network. However, PDNS is recognized to be very large and yet we find that most domains are not detected. An alternate explanation could be that PDNS misses certain FQDNs because they are private and not queried by external hosts. That is, while the missed FQDNs are available in publicly accessible DNS servers, no host uses these FQDNs. Consequently, these FQDNs are de facto not publicly known, and yet, these FQDNs are easily retrievable via zone walking.

IV. PROPOSED SOLUTION - ZONE-HOPPING (ZH)

The objective of our solution is to disable zone walking for specific FQDNs, while still allowing zone walking to find FQDNs that we seek to protect from denial of existence attacks. It is important to note that FQDNs protected from zone walking are not protected from denial of existence attacks. On the other hand, since all signatures are generated offline, this approach does not increase the computational load on the DNS server. Additionally, this method is trivially implemented in BIND9 by simply modifying a configuration file.

To understand ZH, recall that DNSSEC's NSEC records are responsible for zone walking and protection against denial of existence attacks. Our approach involves adjusting the NSEC records so that they never contain sensitive FQDNs. Suppose the domain has three FQDNs, namely public.example.com, sensitive.example.com, and x-posed.example.com. Here, public.example.com and x-posed.example.com accessible to the public, whereas sensitive.example.com is private. In such a scenario, we generate an NSEC with public.example.com as the owner-name and x-posed.example.com as the next FQDN. Note that in lexicographical order, public.example.com $<$ sensitive.example.com $<$ x-posed.example.com. Consequently, if a zone walking query for any FQDN falling between public.example.com and x-posed.example.com, the aforemen-

tioned NSEC record is returned, concealing the existence of sensitive.example.com.

ZH can be easily implemented via BIND9. The proposed solution modifies the NSEC records for instantiated names, which are pre-generated and signed in advance (off-line), but in a slightly different manner: public and sensitive records are separated in two files and each is signed separately with the same key, thus creating two zone-files. Then, the NSEC records from the sensitive zone file are deleted, and this modified file is merged back into the public zone file. This process yields a final zone file devoid of and NSEC records with sensitive data, effectively preventing their disclosure through zone walking while still serving any requests for the sensitive domains with complete authenticity. This solution offers best of both worlds: sensitive information is handled just like traditional DNS (i.e. traditional NXDOMAIN negative answers as opposed to NSEC), but at the same time, digital signatures (RRSIG) are available to prove the authenticity of the records.

A. Implementation

a) Setup: For testing the solution, we setup a test environment with two GCP instances running BIND9 [29] on Ubuntu Linux. One test server was serving as an authoritative name server for the signed (DNSSEC enabled) domain "gotpcap.com" and another test server was acting as recursive resolver with DNSSEC validation enabled.

b) Algorithm steps: We followed steps mentioned below to customize our test zone-file to remove sensitive information. Note that the records in the zone files shown below are truncated for the better readability.

- First, we created a public zone-file with records and signed it using our Zone Signing Key (ZSK). The resulted signed zone is as follows.

```
; gotpcap.signed.db.public
; File written on Tue Mar 12 17:52:50 2024
; dnssec_signzone version 9.18.18
; NOT SHOWING SOA AS IT IS SAME IN ALL
a.gotpcap.com.
            604800 IN A     192.168.2.3
            604800 RRSIG  A 13 3 604800
            604800 NSEC   dns.gotpcap.com.
            604800 RRSIG  NSEC 13 3 604800
dns.gotpcap.com.
            604800 IN A     34.125.87.209
            604800 RRSIG  A 13 3 604800
            604800 NSEC   z.gotpcap.com.
            604800 RRSIG  NSEC 13 3 604800
z.gotpcap.com.
            604800 IN A     192.168.2.4
            604800 RRSIG  A 13 3 604800
            604800 NSEC   gotpcap.com.
            604800 RRSIG  NSEC 13 3 604800
```

- Next, we created a sensitive zone-file with all sensitive records, and again signed it with the same ZSK. The resulted signed zone is as follows.

```
; gotpcap.signed.db. sensitive
; File written on Tue Mar 12 18:06:31 2024
; dnssec_signzone version 9.18.18
; NOT SHOWING SOA Record
b- sensitive .gotpcap.com.
            604800 IN A    10.10.0.1
            604800 RRSIG  A 13 3 604800
            604800 NSEC   c- sensitive .gotpcap.com.
            604800 RRSIG  NSEC 13 3 604800
c- sensitive .gotpcap.com.
            604800 IN A    10.10.0.2
            604800 RRSIG  A 13 3 604800
            604800 NSEC   gotpcap.com.
            604800 RRSIG  NSEC 13 3 604800
```

- Then we deleted all NSEC records from the signed sensitive zone-file. And finally, we combined the resulting file with the public zone-file. The final signed zone is as follows.

```
; gotpcap.signed.db. final
; File written on Tue Mar 12 18:06:31 2024
; dnssec_signzone version 9.18.18
gotpcap.com.
            604800 IN SOA gotpcap.com.
            604800 RRSIG  SOA 13 2 604800
            604800 NS     dns.gotpcap.com.
            604800 RRSIG  NS 13 2 604800
            604800 NSEC   a.gotpcap.com.
            604800 RRSIG  NSEC 13 2 604800
            604800 DNSKEY 257 3 13 xS9zESD
            604800 RRSIG  DNSKEY 13 2 604800
a.gotpcap.com.
            604800 IN A     192.168.2.3
            604800 RRSIG  A 13 3 604800
            604800 NSEC   dns.gotpcap.com.
            604800 RRSIG  NSEC 13 3 604800
dns.gotpcap.com.
            604800 IN A     34.125.87.209
            604800 RRSIG  A 13 3 604800
            604800 NSEC   z.gotpcap.com.
            604800 RRSIG  NSEC 13 3 604800
b- sensitive .gotpcap.com.
            604800 IN A     10.10.0.1
            604800 RRSIG  A 13 3 604800
c- sensitive .gotpcap.com.
            604800 IN A     10.10.0.2
            604800 RRSIG  A 13 3 604800
z.gotpcap.com.
            604800 IN A     192.168.2.4
            604800 RRSIG  A 13 3 604800
            604800 NSEC   gotpcap.com.
            604800 RRSIG  NSEC 13 3 604800
```

B. Testing and Results

We used ldns to zone walk "gotpcap.com". Below are the results before applying the solution.

```
#ldns-walk gotpcap.com
gotpcap.com.      gotpcap.com. NS SOA RRSIG NSEC DNSKEY
a.gotpcap.com.        A RRSIG NSEC
b- sensitive .gotpcap.com. A RRSIG NSEC
c- sensitive .gotpcap.com. A RRSIG NSEC
dns.gotpcap.com.      A RRSIG NSEC
z.gotpcap.com.        A RRSIG NSEC
```

As can be seen below, the sensitive FQDNs are not longer discovered via zone-walking after we implemented the solution:

```
#ldns-walk gotpcap.com
gotpcap.com.      gotpcap.com. NS SOA RRSIG NSEC DNSKEY
a.gotpcap.com.        A RRSIG NSEC
```

```
dns.gotpcap.com.      A RRSIG NSEC
z.gotpcap.com.        A RRSIG NSEC
```

However, querying the sensitive FQDNs generates a valid response from the name-server.

```
# dig @34.125.255.60 c-sensitive.gotpcap.com a +multi +dnssec
;;
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43569
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0,
    ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
;; QUESTION SECTION:
;c-sensitive.gotpcap.com. IN A

;; ANSWER SECTION:
c-sensitive.gotpcap.com. 604709 IN A 10.10.0.2
c-sensitive.gotpcap.com. 604709 IN RRSIG A 13 3 604800
    (20240411170631 20240312170631 11305 gotpcap.com.
knx4IK1axo/yZhmw5GfP7m39acCwGdz60NGLPlKxjoFy
h89ixC0TZ/pwEzWTF4bG+s+xPCcXdW2F2FDmuuyDBg== )
```

C. Limitations and Spectrum of solutions

The proposed method of ZH is susceptible to denial of existence attacks for sensitive FQDNs via record replay. This attack could proceed as follows. Suppose that sensitive.example.com is private, therefore is not included in any NSEC record. Instead, there is an NSEC record that has the preceding FQDN as public.example.com and the next FQDN as x-posed.example.com, hence public.example.com $<$ sensitive.example.com $<$ x-posed.example.com. Now, if a validating DNS resolver requests the IP address for sensitive.example.com, a MitM attack could return this NSEC record, which verifies that sensitive.example.com does not exist.

Combining the solution described above with the other available solutions, we have the following spectrum of solutions that should satisfy most use-cases.

- DNSSEC-NSEC protects against denial of existence attacks, is not especially vulnerable to DoS attack, but is vulnerable to zone walking.
- BL protects against denial of existence attacks and zone walking, but is susceptible to DoS attacks, which have the same impact as denial of existence attacks.
- ZH allows zone walking only on public FQDNs and protects against denial of existence attacks on these public FQDNs. For sensitive FQDNs, zone walking is not possible and denial of existence is possible. Moreover, the ZH is not especially vulnerable to DoS attacks.

V. Conclusion

We demonstrated that information leakage via NSEC-based zone walking is an important threat. The importance of this threat is demonstrated by showing that some FQDNs are private and cannot be found through passive listening. Moreover, knowing FQDNs is critical for the reconnaissance of IPv6 networks. And furthermore, these FQDNs might include sensitive information such as employee names and descriptive names that are not intended to be publicly queried.

We proposed a practical solution, ZH, that is easy to implement and addresses these security risks. While there remains a trade-off between protecting against denial of existence attack and zone walking, the proposed method allows the administrator to precisely control which FQDNs should be public and receive protection against denial of existence attacks and which FQDNs should be private, but potentially suffer from denial of existence attacks. Moreover, unlike other methods, this method does not reply on creating signatures on-the-fly, which requires the private key to be located on a publicly accessible server, and protects against unintentional key disclosure. It also does not require computational effort that, in practice, makes the server vulnerable to DoS attacks, which has the same impact as a denial of existence attack.

VI. Discussion: ZH vs. Split-View DNSSEC

As was shown in Section III, zone walking can expose FQDNs from a private network to the public Internet. This vulnerability can be mitigated by using the Split-View DNS architecture, where a private DNS server is used to protect internal domains from being leaked to the public Internet [30] [31]. While this architecture addresses some aspects of the risks associated with zone walking, it does not entirely eliminate them. Specifically, even with a split-view DNS architecture, an external attacker can quickly discover all publicly facing servers and, upon gaining access to the private network, the attacker can efficiently discover internal servers, which facilitates lateral movement within the network.

Without ZH or some other protection against zone walking, employing a Split-View DNS architecture is essential to conceal the names of internal servers. However, even with zone walking countermeasures in place, a split-view DNS architecture is useful to prevent brute-force searches of all possible server names from finding internal FQDNs.

VII. Ethics

We consider the security and privacy of the hosts and zone owners with high regard. The data collected via zone walking and PDNS was only used for research purposes and handled with due diligence and care. Data was not leaked or shared with others, except with people working on this research. Nor did we misused data to get insights into other organizations, do any kind of unethical disclosures, or for any personal gain. We don't intend to share any data that was collected with the public in order to protect the privacy of owners and parties involved.

Acknowledgment

We would like to acknowledge the contribution of DomainTools for providing access to their PDNS database for research advancement and enabling researchers to perform analysis on the real world data collected from their network sensors. We appreciate their help and prompt support.

REFERENCES

[1] D. E. E. 3rd, "Domain Name System Security Extensions," RFC 2535, Mar. 1999. [Online]. Available: https://www.rfc-editor.org/info/rfc2535

[2] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends, "Resource Records for the DNS Security Extensions," RFC 4034, Mar. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc4034

[3] F. Alharbi, J. Chang, Y. Zhou, F. Qian, Z. Qian, and N. Abu-Ghazaleh, "Collaborative client-side dns cache poisoning attack," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1153–1161.

[4] D. Adamitis, D. Maynor, W. Mercer, M. Olney, and P. Rascagneres, "Dns hijacking abuses trust in core internet service," 2019.

[5] H. Nebuchadnezzar, "The collateral damage of internet censorship by dns injection," *ACM SIGCOMM CCR*, vol. 42, no. 3, pp. 10–1145, 2012.

[6] "Zone walking (zone enumeration via dnssec nsec records," https://www.domaintools.com/resources/blog/zone-walking-zone-enumeration-via-dnssec-nsec-records/, 2024.

[7] R. Arends, G. Sisson, D. Blacka, and B. Laurie, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence," RFC 5155, Mar. 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5155

[8] M. Wander, L. Schwittmann, C. Boelmann, and T. Weis, "Gpu-based nsec3 hash breaking," in *2014 IEEE 13th International Symposium on Network Computing and Applications*. IEEE, 2014, pp. 137–144.

[9] H. Mitchell, "Taking the dns for a walk; nsec3 prevalence and recoverability," https://harrisonm.com/whitepaper/nsec3-prevalence-and-recoverability.pdf.

[10] "Dnssection," https://infocondb.org/con/def-con/def-con-28/dnssection-a-practical-attack-on-dnssec-zone-walking, 2020.

[11] S. Weiler and J. Stenstam, "Minimally Covering NSEC Records and DNSSEC On-line Signing," RFC 4470, Apr. 2006. [Online]. Available: https://www.rfc-editor.org/info/rfc4470

[12] R. M. Gieben and M. Mekking, "Authenticated Denial of Existence in the DNS," RFC 7129, Feb. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7129

[13] O. G. F. Valsorda, "Compact dnssec denial of existence or black lies," https://datatracker.ietf.org/doc/html/draft-valsorda-dnsop-black-lies-00, 2016.

[14] "Economical with the truth: Making dnssec answers cheap," https://blog.cloudflare.com/black-lies, 2016.

[15] "Dnssec proofs of nonexistence in route 53," https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-configuring-dnssec-proof-of-nonexistence.html, 2016.

[16] "Dns zone walking," https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/best-practices-resolver-zone-walking.html.

[17] G. Huston, "Dnssec-the opinion," https://blog.apnic.net/2023/02/20/opinion-to-dnssec-or-not/, 2023.

[18] "Kali linux: dnswalk," https://www.kali.org/tools/dnswalk/.

[19] "Kali linux: Dnsrecon," https://subscription.packtpub.com/book/security/9781789952308/1/ch01lvl1sec09/zone-walking-using-dnsrecon, 2020.

[20] "ldns-walk," https://linux.die.net/man/1/ldns-walk, 2005.

[21] V. Hadjitodorov, "A novel zone-walking protection for secure dns server," *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, vol. 14, no. 1, pp. 1–15, 2022.

[22] H. Rafiee, C. Mueller, L. Niemeier, J. Streek, C. Sterz, and C. Meinel, "A flexible framework for detecting ipv6 vulnerabilities," in *Proceedings of the 6th International Conference on Security of Information and Networks*, ser. SIN '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 196–202. [Online]. Available: https://doi.org/10.1145/2523514.2527001

[23] Q. Hu, M. R. Asghar, and N. Brownlee, "Measuring ipv6 dns reconnaissance attacks and preventing them using dns guard," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, pp. 350–361.

[24] V. Hadjitodorov, "Security of ipv6 and dnssec for penetration testers," *Research Project, University of Amsterdam July*, 2011.

[25] "Completing the transition to internet protocol version 6 (1pv6)," https://www.whitehouse.gov/wp-content/uploads/2020/11/M-21-07.pdf, 2020.

[26] P. E. Hoffman, A. Sullivan, and K. Fujiwara, "DNS Terminology," RFC 8499, Jan. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8499

[27] Farsight, "Farsight dnsdb api version 2 documentation," https://www.domaintools.com/resources/user-guides/farsight-dnsdb-api-version-2-documentation/, 2024.

[28] "Dnsdb frequently asked questions (faq)," https://www.domaintools.com/resources/user-guides/dnsdb-frequently-asked-questions-faq/.

[29] "Dnssec-guide," https://bind9.readthedocs.io/en/latest/dnssec-guide.html, 2024.

[30] "Split-view dnssec operational practices," https://datatracker.ietf.org/doc/html/draft-krishnaswamy-dnsop-dnssec-split-viewsection-2.1, 2007.

[31] "Establishing local dns authority in validated split-horizon environments," https://datatracker.ietf.org/doc/draft-ietf-add-split-horizon-authority/, 2024.