Aalto University
School of Science
Bachelor's Programme in Science and Technology

DNS Enumeration Techniques

Bachelor's Thesis

April 7, 2021

Tuomas Välimäki

Aalto-yliopisto
Perustieteiden korkeakoulu
Teknistieteellinen kandidaattiohjelma

KANDIDAATINTYÖN
TIIVISTELMÄ

Tekijä:	Tuomas Välimäki
Työn nimi:	DNS Enumeration Techniques
Päiväys:	7. huhtikuuta 2021
Sivumäärä:	24
Pääaine:	Tietotekniikka
Koodi:	SCI3027
Vastuuopettaja:	Professori Tuomas Aura
Työn ohjaaja(t):	Titteli Jacopo Bufalino (Tietojenkäsittelytieteen laitos)

Domain name system (DNS) eli nimipalvelinjärjestelmä on oleellinen osa nykyistä Internetiä. Nimipalvelinpalvelimille tehtävien kyselyiden avulla saadaan selville verkkotunnusta vastaava IP osoite. Nimipalvelinjärjestelmä toisaalta mahdollistaa sen, että penetraatiotestaaja tai hyökkääjä pystyy kartoittamaan kohteen verkkoinfrastuktuurin nimipalvelinkyselyiden avulla, nimipalvelintietojen ollessa julkisia ja helposti saatavilla.

Tutkielma käsittelee erilaisia menetelmiä kohteen verkkoinfrastuktuurin kartoittamiseksi nimenomaan nimipalvelinkyselyitä hyväksi käyttäen. Tutkielma on pääosin kirjallisuuskatsaus, jonka keinoin perehdytään erilaisten palvelukartoitustekniikoiden toimintaperiaatteisiin.

Tutkimuksen edetessä ilmeni että alunperin vuodelta 1987 oleva nimipalvelinjärjestelmä ei pitänyt sisällään minkäänlaista tietoturvaa. Tästä syystä tutkielmassa on palvelukartoitustekniikoiden lisäksi käsitelty sitä, minkälaisia tietoturvalaajennuksia alkuperäiseen nimipalvelinjärjestelmään on tehty vuosien varrella.

Palvelukartoitustekniikoiden lisäksi tutkielmassa esitellään nimipalvelinjärjestelmän toiminta lyhyesti. Lisäksi käsitellään sitä miten nimipalvelinjärjestelmä on kehittynyt ajan saatossa erityisesti tietoturvan näkökulmasta sekä millaisia muutoksia nimipalvelinjärjestelmässä on tapahtunut viimeaikoina. Tutkielman lopussa käsitellään lyhyesti pilvipalveluita palvukartoituksen näkökulmasta.

Avainsanat:	DNS, domain name system, nimipalvelinjärjestelmä, palvelukartoitus
Kieli:	Suomi

Aalto University
School of Science
Bachelor's Programme in Science and Technology

ABSTRACT OF
BACHELOR'S THESIS

Author:	Tuomas Välimäki
Title of thesis:	DNS Enumeration Techniques
Date:	April 7, 2021
Pages:	24
Major:	Tietotekniikka
Code:	SCI3027
Supervisor:	Tuomas Aura, Professor
Instructor:	Jacopo Bufalino, titleOfInstructor (Department of Information and Computer Science)

Domain Name System (DNS) is an integral part of modern Internet which allows resolving domain names into IP addresses. Due to publicity of DNS records and their easy access, DNS makes it possible for a penetration tester or an attacker to acquire information about the target network infrastructure using DNS queries.

This thesis discusses different methods of using DNS queries or DNS enumeration to map the target network infrastructure. Research method is mainly literature review focusing on different techniques and their operating principles.

In addition to DNS enumeration techniques the thesis presents an overview of DNS, discusses how DNS has evolved from a security point of view and gives a brief overview of recent developments in DNS. In the end enumerating cloud services is briefly discussed.

Keywords:	DNS, domain name system, DNS enumeration, service enumeration
Language:	English

Contents

1 Introduction

The first stage of committing a cyber attack or a penetration test is reconnaissance or a network footprinting. The aim of reconnaissance is to acquire information about the target network infrastructure. Various footprinting techniques may be used to query publicly available data, such as using advanced search features of search engines and WHOIS queries. One such technique is Domain Name System (DNS) enumeration which uses public DNS records to gather information about the target without actually probing the target network. The Domain Name System may reveal information such as IP addresses, domains, subdomains, mail exchange services and services.

Objective of this thesis is to conduct a literary review about the existing DNS enumeration techniques and their operating principles and related security enhancements that has been made to DNS.

The goals of this thesis is as follows:

- Conduct a literary review about existing DNS enumeration techniques
- Discuss modifications that have been made to the Domain Name System to implement security
- Present some use cases of the existing methods.

DNS has been and still is vulnerable to attacks such as DNS cache poisoning where the attacker tricks a DNS server or a resolver to store false information in a DNS cache e.g., to route a client to a phishing site. DNS Security Extensions or DNSSEC has been developed to prevent cache poisoning attacks but the way DNSSEC is implemented has opened up new ways to conduct DNS enumeration. Therefore DNSSEC will be discussed but DNS cache poisoning attacks are out of the scope of this study. In addition recent developments in DNS, enumerating cloud services are briefly discussed.

2 The Domain Name System

The Domain Name system or DNS [24][23] is a fundamental part of the Internet infrastructure. This chapter gives a brief overview of DNS, related security issues and recent developments related to this study.

2.1 Overview

Domain Name System is a distributed database implemented in a hierarchy of DNS servers with delegated authority that stores information about services and other resources on the Internet. Notably, DNS stores information that maps host names to IP addresses. The Domain Name System as a whole consists of specification for the functionality of DNS servers and the DNS protocol. The DNS protocol for DNS queries is for the most part based on the UDP protocol. DNS protocol is part of the Internet Protocol Suite and as an application layer protocol is commonly employed by other application layer protocols such as HTTP, SMTP or FTP to resolve domain names to IP addresses. [17][35]

When an application makes a query to retrieve an IP address for a host name, the corresponding IP address is resolved in an iterative manner by a DNS resolver. For most Internet users DNS resolver is a service provided by Internet Service Provider. From an application perspective query is executed in a recursive manner, see Figure 1.

At the top level of the name server infrastructure there are 13 logical root name servers distributed across the globe named from A to M. If an answer for a query is not found in cache of the resolver a root name server is queried first which delegates the query for a top level domain such as .com., to a Top Level Server Domain server or TLD server. The iteration continues until an Authoritative Name Server is found for the host. The query result is returned to the client via DNS resolver. [17] [35]

There is another class of DNS servers called a local or private DNS Server which is strictly not part of the DNS hierarchy, but is part of the DNS infrastructure. A private DNS server may be used in a private network. [17]

2.2 DNS Resource Record Types

A DNS database or a DNS record has multiple queryable resource record types. Most common and relevant for this study are listed in Figure 1. Among A and AAAA records, PTR record has a special interest related to this study since it allows reverse DNS lookup which maps an IP address to a hostname. Main uses for PTR records are anti-spam filtering and logging [5]. NSEC and NSEC3 resource records can be used to execute an enumeration method called Zone Walking as is discussed later on.

2.3 DNS Zones

A DNS zone is a distinct part of domain namespace which is delegated to an entity such as an organization. DNS zones should not be associated with domain namespaces, meaning there is no one to one correspondence between a zone and a domain. A single DNS server

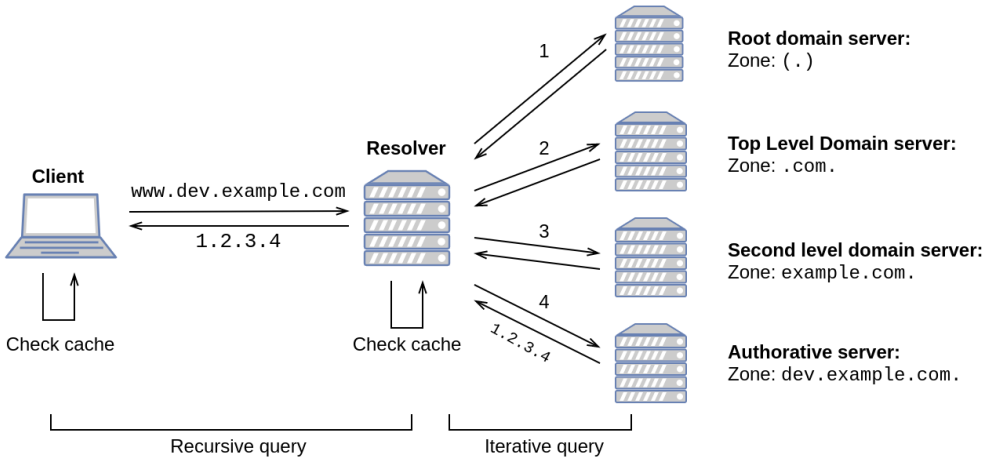

Figure 1: A simplistic presentation of resolving an IP address for a host `www.dev.example.com`. An Iterative query is performed by the DNS resolver until an authoritative name server for subdomain `dev.example.com` is found. The authoritative server has a type A resource record which maps the name `www.dev.example.com` into an IP address. Explanation for zones, see Section 2.3.

may be responsible for multiple zones, and for replication, load distribution and security (e.g., to deflect DDoS attacks) there are usually multiple domain name servers for each zone. [35]

In the example presented in Fig. 1 second level domain name server for `example.com` has delegated the responsibility of subdomain `dev.example.com` for another zone.

If there are multiple domain name servers for a zone there is a primary DNS server which is replicated by secondary servers. A secondary DNS server maintains a read-only copy of the DNS zone file maintained in the primary DNS server. If modifications are made they are made into the DNS zone file of the primary DNS server. The process of secondary DNS server requesting an update and receiving this information is called a Zone Transfer. [35]

2.4 Publicity and Security of DNS

The intrinsic property of DNS records is that they are public by design and normal DNS queries are not encrypted. In the initial design of DNS there were no security features implemented. The DNS system has a long history of being vulnerable for DNS spoofing, also referred to as DNS cache poisoning, where the attacker tricks a DNS server to store false information in a DNS cache. [35][4]

DNS Security Extension or DNSSEC (defined originally in now obsolete [1]) has been developed to provide end-to-end data integrity and source authenticity check between

Table 1: Most common or most the relevant DNS Record types for this study.

Record Type	Description
A	IPv4 address for a host. [24]
AAAA	IPv6 address for a host [36]
MX	Mail exchange server. [24]
CNAME	Canonical Name. Maps an "alias" to a canonical name. [24]
SOA	Start of Authority. Specifies authoritative information about a DNS zone, including the primary name server for the zone. [24]
PTR	A Pointer Record. Maps an IP address to a hostname. [24]
NSEC	Provides an authenticated denial of existence for DNS Resource Record Sets. Points to a next Domain in canonical order. [16].
NSEC3	Similar to NSEC but hashed values of domain names are used instead of plain text. [19]
NSEC3PARAM	Holds cryptographic information related to NSEC3. [19]
RRSIG	Resource record signature for a resource record set. [2]
SRV	Specifies hostname and port number of servers for specified services. [13]

the source of DNS data and the client. To achieve this, DNSSEC uses public/private key pair signatures (RRSIG resource record). Threaths of data disclosure were ruled out of scope of DNSSEC [4]. Although DNSSEC was standardized already in 1997 it is still not widely used but the adoption is slowly progressing [33]. In 2013 ICANN started a process of approving new TLDs with the additional requirement that they are secured by DNSSEC [15]. DNSSEC relies on authentication chain of trust. All root level domain name servers use DNSSEC signatures, but if a TLD does not implement DNSSEC there can not be DNSSEC for any domain under such TLD since the chain of trust is broken [33].

Another security measure, Transmission Signatures or TSIG [38] use shared private keys to provide authentication of servers to secure actions such as zone transfers and dynamic updates of resource records.

2.5 Recent developments in DNS

Even though DNS was initially designed to translate domain names to IP addresses, the DNS has evolved into a complex system and has many uses beyond its original intent. A common dns query for example for `TXT` resource records reveal much information such as Google site verification and whitelisting email for spam verification. In the current form DNS may even be considered to be a distributed key value database for myriad of applications.

Currently, DNSSEC provides data integrity and source authenticity, but DNS lacks confidentiality since DNS queries and responses are visible to network elements on the path. In recent years new standards for encrypting DNS query traffic has been proposed such as DNS over TLS [RFC 7858], DNS Queries over HTTPS [RFC 8484] and DNS over Datagram Transport Layer Security [RFC 8094].

For some domains, it can be desirable to provide web sites or other services at a bare domain name, such as `domain.com` instead of `www.domain.com`. Use of CNAME record type for this purpose is in conflict with DNS specifications. RFC 1034 [23] states that if there is a CNAME RR in a zone for a name there can not be any other RR for that name in the zone. If the zone is a so-called apex zone i.e., the zone contains SOA and NS resource records for the domain name, according to [23] use of CNAME for the domain name in this zone is prohibited. To overcome this limitation some managed DNS providers such as Route 53 by Amazon [31], DNSimple [7] and ClouDNS [6] use a non-standard ALIAS resource record that can coexist with other resource records for a name. The ALIAS record is not returned to a DNS resolver but is resolved instead internally by the DNS server into an IP address to achieve compliance with DNS specifications. Due to use of a non-standard ALIAS resource record by multiple DNS providers a standard for new resource record (ANAME) with similar functionality is being developed [8].

As internet services are moving to the cloud, websites are turning to Content Delivery Networks (CDN) to provide services. CDNs are dynamic and change IP addresses based on the users' location and server load. Instead of static DNS records managed DNS providers nowadays use an anycast network and provide dynamic DNS records with features such as geolocation, failover and even load balancing on a DNS level [9].

3 DNS Enumeration

Due to the original design philosophy and easy access principle, DNS is vulnerable to information leakage and attacks such as DNS cache poisoning. Since DNS records are public DNS queries allow attackers the gather information about the target networks using various methods of DNS Enumeration such as using a dictionary attack to enumerate subdomains. DNS Enumeration is part of the reconnaissance stage where an attacker gathers information about the target network. DNS enumeration includes gathering information about IP addresses, domains and subdomains.

DNS Enumeration is considered to be part of the Open Source Intelligent or OSINT methodology where publicly available (open source) information is used in an intelligence context [22]. DNS enumeration is commonly used jointly with other OSINT methods such as `whois` queries and exploiting advanced search engine features such as "Google Dorks" [22].

This chapter will review the most common methods of DNS enumeration and in the last section (3.6) cloud services are briefly discussed.

3.1 Basic DNS lookup

Every resource record or RR can be considered a five-tuple (Name, Type, Class, Time to Live, Value) [24]. Some of the Types were listed in Table 1. In addition to Types (TYPE), DNS defines query types (QTYPE) which is a superset of types. The DNS protocol also defines a query class (QCLASS) as a superset of class (CLASS). Every query contains QNAME (fully qualifed domain name or FQDN), QTYPE and QCLASS [24].

`Nslookup` and 'Domain Internet Grober' (`dig`) are common tools found in most Unix enviroments to carry out basic DNS queries. For example `dig -q www.example.com -t AAAA -c IN` returns an IPv6 address for `www.example.com` with query class `IN` (Internet). In practise the query class is now redundant since the query class `IN` is the only one in use but is required by the protocol definition.

Above example which returns an IP address for a host name is called a forward lookup. A reverse lookup is a query returning a host name for an IP address (PTR record).

3.2 Zone Transfer Attack

As explained in Section 2.3 each DNS zone is composed by a primary server and secondary servers. Secondary servers maintain read-only copies of the primary servers zone file. DNS protocol defines a special type of query, Asynchronous Transfer Full Range or AXFR

which requests for a transfer of an entire list of resource records for zone or a Zone File [23]. The DNS zone information may include sensitive information e.g., the internal infrastructure of a network of an organization. If a DNS server is misconfigured, the attacker may retrieve the zone file from a DNS server. The zone file is a plain text list of all resource records of the zone.

Although succeeding in unauthorized zone transfer requests is considered by some as a thing for the past [22], on a global-scale vulnerability assessment Skwarek et al. [34] were able to carry out approximately 11 million zone transfers containing information such as HINFO (information about OS and CPU [24]) records, and domains with `test.` and `dev.` prefixes etc.

The AXFR offers no authentication mechanism [24][20]. In order to prevent the vulnerability from occurring the DNS server should be configured to only allow AXFR requests from trusted IP addresses. Additionally the use of TSIG or SIG(0) are recommended for authentication of trusted servers[20]. Since AXFR uses TCP [23][24] a firewall could be configured to block TCP traffic to port 56 for untrusted clients but since DNS queries over 512 bytes will be transferred over TCP this is not practical [21].

3.3 Reverse DNS sweeping

Once IP address ranges used by the target are discovered one can use reverse DNS sweeping. IP network blocks can be discovered for example using whois queries or other reconnaissance methods (see e.g., [22]) or even by an educated guess. Reverse DNS sweeping in itself is quite a simple method. Reverse DNS queries are performed for a range of IP addresses.

As an example DNS forward lookup reveals four A resource records for domain `hs.fi`:

```
---
;; ANSWER SECTION:
hs.fi. 60 IN A 13.32.143.49
hs.fi. 60 IN A 13.32.143.98
hs.fi. 60 IN A 13.32.143.126
hs.fi. 60 IN A 13.32.143.119
---
```

One can infer (or quess) that maybe there is a reserved IP block, `13.32.143.0/24`. Reverse sweep can be executed for example using `nmap` tool:

```
> nmap -sL 13.32.143.0/24 | grep "13" | awk '{printf("%s %s\n",$5,$6);}'
```

```
server-13-32-143-0.hel50.r.cloudfront.net (13.32.143.0)
server-13-32-143-1.hel50.r.cloudfront.net (13.32.143.1)
...
server-13-32-143-254.hel50.r.cloudfront.net (13.32.143.254)
server-13-32-143-255.hel50.r.cloudfront.net (13.32.143.255)
```

From the domain names one can infer that the service is provided by Amazon Web Services content delivery network. Further investigation reveals that ip addresses are for Cloudfront edge servers [29].

3.4 Zone Enumeration

DNSSEC introduced the ability for a hostile party to enumerate all names in a zone. Initially, ability to enumerate all the names in a zone was not considered as an error but was recognized as a drawback of DNSSEC [3].

As explained in section 2.4 the purpose of DNSSEC is to provide end-to-end data integrity and source authenticity check in the Domain Name System. This section will not cover DNSSEC in detail but an important part of DNSSEC which is called an authenticated denial of of existence in DNS [10]. The section will further discuss how the authenticated denial of existence can be leveraged to execute a DNS enumeration technique called Zone Walking or Zone Enumeration.

3.4.1 Denial of Existence in DNS

DNSSEC signatures are used only for signing the Resource Records in the DNS response message or packet. The DNS message header is not signed. [10]

The DNS response message header contains a status code. If a DNS query is made for an existing name (QNAME), status code of the response header is NOERROR and the answer section contains Resource Records for the name in question, which are signed if DNSSEC is supported. There are two cases of status codes corresponding to situations where requested Resource record does not exits [10]:

- If a query is made for a name that does not exist the status code of the response is `NXDOMAIN` (Non-Existing Domain).
- If a query is made for a name that does exist but the requested resource record does not exist the status code of the response is `NOERROR`. Based on the response the resolver can infer that the type of response is of type called `NODATA`.

Since DNSSEC signature is not used to sign DNS message header the NXDOMAIN status cannot be trusted. In addition the DNSSEC signatures are precomputed. The original design of DNSSEC excluded online signing due to security reasons (to keep the private key offline) and the computational overhead. Therefore, offline signing the header is not feasible since it would require precomputing signatures for all conceivable nonexisting answers. [10]

Above elaborates the issue with DNSSEC and authenticated denial of existence: if an empty answer is returned there are no resource records to sign. The first attempt to specify authenticated denial of existence was NXT resource record. NXT was introduced in [1] but has been superceded by NSEC resource record [16].

NSEC resource record indirectly lets the resolver know that the name does not exist in the zone. Zone is sorted into canonical order and NSEC describes an interval between names [2]. For example if a zone contains domain names `a.domain.org` and `c.domain.org` and in canonical ordering there are no domain names in between in the zone, DNS query for a domain name `b.domain.org` returns (class and time to live omitted for simplicity):

```
Name              Type    Value
a.domain.org      NSEC    c.domain.org NS SOA TXT RRSIG NSEC DNSKEY
```

The answer above is returned since `b.domain.org` resides in the interval between `a.domain.org` and `c.domain.org`. The value returned contains the endpoint of the interval and a list of resource records types where RRSIG, NSEC and DNSKEY are related to DNSSEC. NSEC records are signed as are other resources in DNSSEC.

NSEC records are also used in `NODATA` responses as is explained in [10].

3.4.2 Zone Walking

The drawback of NSEC is that it permits DNS enumeration method called Zone Walking or Zone Enumeration. An enumerated zone can be used, for example, as a source of open mail servers for spam. NSEC records point from one name to another and this allows enumeration of the whole zone thus defeating attempts to block zone transfers. An attacker can query these NSEC RRs in sequence to obtain all the names in a zone. [26] [21] [10]

For example using `dig` tool it is easy to see if a top level domain uses NSEC. In the example below one can see that when that when making a DNS query for a TLD name server (`a.fi`) responsible for top level domain `fi.`, the authority section reveals that NSEC3 is used for denial of existence, see section 3.4.3. Dig tool offers an option `+dnssec` which sets the `DO` bit ("`DNSSEC OK`") to inform a DNS server that the client is DNSSEC-aware [RFC4035].

```
> dig +dnssec @a.fi fi.
---
;; AUTHORITY SECTION:
---
uhirv2ck3kkgn20e3fnqecj1nfu6eeko.fi. 86400 IN NSEC3 1 1 5 7B7EE2F28EAFC626
 UHKSO58SOE3BAFENJNSU6QKL13RGLTP4
---
```

In the previous result `uhirv2ck3kkgn20e3fnqecj1nfu6eeko` is the hash value for `fi.`, `7B7EE2F28EAFC626` a salt used in hashing and `UHKSO58SOE3BAFENJNSU6QKL13RGLTP4` the next hash value for a domain in the canonical list of hashes. The number of iterations used for computing the hash is 5.

On the other hand, using `dig` to query name server (`a.ns.fi`) responsible for TLD `se.` the authority section reveals that NSEC is used which allows zone walking.

```
> dig +dnssec @a.ns.se se.
---
;; AUTHORITY SECTION:
---
se.   7200    IN NSEC  0.se. NS SOA TXT RRSIG NSEC DNSKEY
---
```

In the previous answer `se.` is the start of an interval and `0.se.` is the end of an interval. Requesting DNSSEC resource records for the domain name `0.se.` one can find out the next domain name:

```
> dig +dnssec @a.ns.se se. NSEC
---
;; ANSWER SECTION:
---
0.se.   7200    IN NSEC  0-0.se. NS DS RRSIG NSEC
---
```

Previous answer reveals that the next interval is from `0.se` to `0-0.se`. Iterating previous procedure and listing the domain names is the operating principle of Zone Walking. To illustrate how easily NSEC can be leveraged to perform zone a small Python script was written to execute zone walking. An example of an output, starting the enumeration from domain `aftonbladet.se.` is presented below.

```
aftonbladet.se.
aftonbladet-cdn.se.
aftonbladet-cloudflare.se.
aftonbladet5.se.
aftonbladet6.se.
aftonbladeta.se.
aftonbladetbingo.se.
---
```

Notable is that the Internet Foundation in Sweden [15] intends the `.se` zone to be public and the zone file can be transferred over AXFR request from a name server `zonedata.iis.se`.

3.4.3 NSEC3

NSEC is implemented using a sorted linked list of clear text names in the zone. RFC 5155 [19] introduces NSEC3 as an alternative for NSEC which aims to make zone enumeration considerably harder. NSEC3 still works the similar way but instead of using clear text names, hashed names are used to construct the (linked) list for names in the zone (see Fig. 2). NSEC3 was developed to attempt to obstruct zone walking using obfuscated or hashed names.

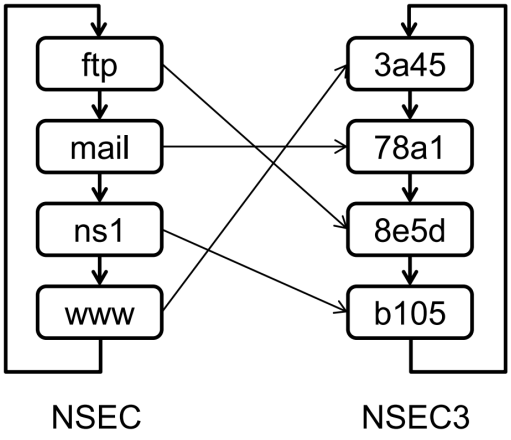

Figure 2: A simple representation of how NSEC and NSEC3 differ. In the canonical ordering NSEC uses a linked list of plaintext domain names in lexicographic order. NSEC uses a linked list of hashed domain name values. Image from [40].

Study conducted by Wander et al. [40] showed that even with NSEC3 zone enumeration is still possible using GPU-based NSEC hash breaking. The algorithm had two parts. Since

querying NSEC3 records directly is impossible the first phase was to query non-existing random names until all NSEC3 records (or NSEC3 intervals) had been retrieved and NSEC3 intervals formed a fully connected and circular path so crawling NSEC3 intervals was finished.

After crawling the NSEC3 intervals brute forcing and dictionary based attack was used for hash breaking. The result was that approximately 64 % of all NSEC3 hashes of the top level domain `com.` was reversed after 4.5 days of computation [40]. The Technique presented by Wander et al. [40] differs from using a brute force attack to directly query (or "bombard") a DNS server for domain names in a sense that the technique was carried out mostly offline. Notable is that the `com.` TLD NSEC3 hashing used no salt in addition to zero iterations (hash computed only once) when computing the hash value for a domain name.

In addition in the study Wander et al. [40] learned that 52 % of the TLDs did not change their salt value during the 1 year of observation period and 5 % of the TLDs did not use salt at all making both cases vulnerable to "rainbow table" attacks i.e., table of precomputed hash values.

Since zone enumeration is feasible even with NSEC3, a variant of NSEC3, NSEC5 [37] has been suggested to overcome the existing vulnerabilities in an authenticated denial of existence in NSEC and NSEC3. This thesis will not discuss NSEC5 but suffice to say that NSEC5 is designed so that it will prevent zone enumeration [11].

3.5 Subdomain enumeration

The first stage in an attempt to enumerate subdomains is to try zone transfer attack but the probability of succeeding is low [40]. This section discusses the few methods which may be leveraged to enumerate subdomains. Enumeration of subdomains using methods described in this section can be used in conjunction with advanced search engine features e.g., "Google Dorks" to gather information about the subdomains [21].

3.5.1 TTL/TLS certificates

Information about subdomains can ge gathered by examining the TLS certificate for a top-level domain. A TLS certificate contains a field Certificate Subject Alternative Name which indicates all of the domain names and IP addresses that are covered by the certificate [21]. As an example examining the certificate for `www.aalto.fi` one can see that the Subject Alternative Name field contains nearly 30 domain names:

```
DNS Name: aalto.fi
```

```
DNS Name: 5g-research.aalto.fi
DNS Name: abe.aalto.fi
DNS Name: accounting.aalto.fi
DNS Name: acre.aalto.fi
DNS Name: act.aalto.fi
DNS Name: aef.aalto.fi
DNS Name: aiic.aalto.fi
DNS Name: alumni.aalto.fi
...
DNS Name: www.aalto.fi
```

Of course this is not a DNS enumeration technique per se but presents a simple yet effective way to enumerate subdomains in certain cases.

3.5.2 Brute forcing

As the name suggests simple dictionary based brute forcing for subdomains is not a very finessed method. On the other hand, the ”guessing” of subdomain names is feasible to a certain degree since certain services tend to have names which reflect their purpose. The downside is that simplistic dictionary based brute forcing methods tend not to give reliable results since the method depend of a huge list of known words and thus, will not work against unknown names [39]. Most commonly known tools for enumerating subdomain names are `DNSEnum` [27] and `Fierce` [28] both using dictionary based brute forcing to enumerate subdomains in addition to methods described in previous sections.

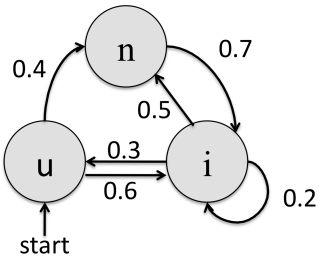

Figure 3: A simplistic illustration of generating subdomain names using a simple Markov chain model i.e., a state machine model. Using training data the transition probabilities were calculated by Wagner et al. [39] and the model was used to generate subdomain names. The initial character was chosen randomly based on the distribution of initial characters in the training data and subsequent characters were generated by the model.

Instead using dictionary based attacks, Wagner et al. [39] demonstrated that applying a relatively simple language generation model achieved better results than a basic dictionary

based brute forcing. The training data was gathered by passive DNS monitoring activity for generating the domain names, see Figure 3 for more details.

The study conducted by Wagner et al. [39] is relatively old and one could argue that now fashionable autoregressive language models could be leveraged to generate subdomain names but no related studies were found.

3.6 Enumerating Cloud Services

Services in the cloud may not reveal much information about the service as the internal infrastructure is not visible from outside. As an example, Figure 4 illustrates a recommended three tier architecture for a cloud infrastructure in a case of Wordpress application in Amazon Web Services (AWS) [30]. In the architecture security critical components such as databases and volumes have been ”hidden” in private subnets of the Virtual Private Cloud (VPC) with no access to the Internet thus making accessing and enumeration from the outside impossible. Enumeration of the internal structure would require the attacker to gain an access to an internal component such as an EC2 instance (virtual machine) in the case of AWS.

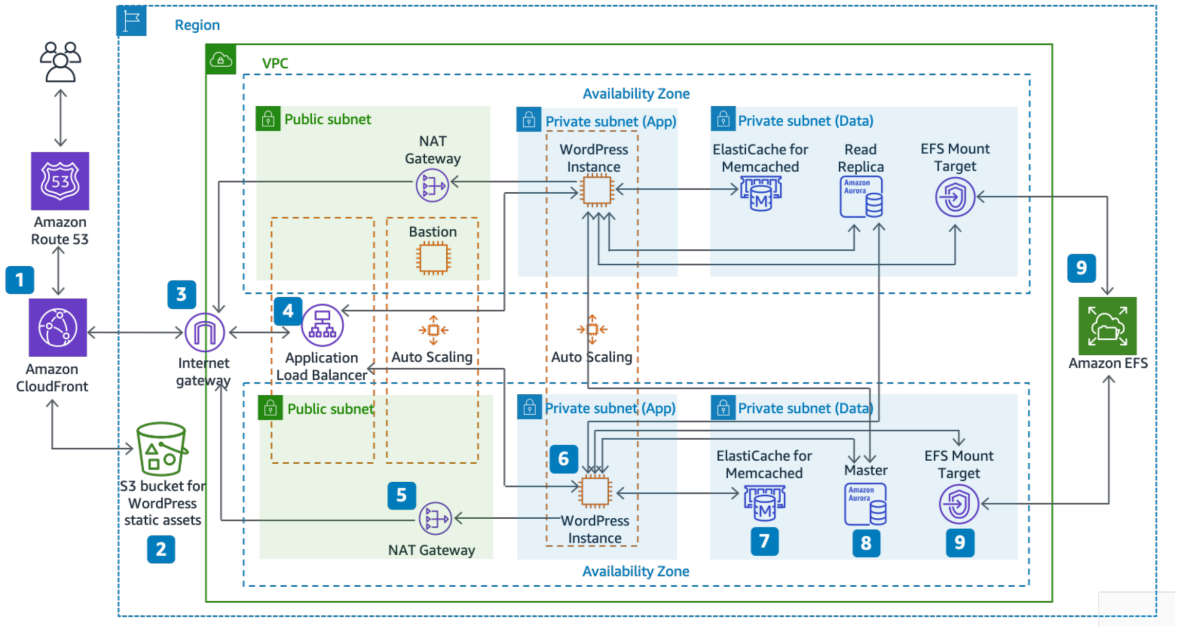

Figure 4: A three tier achitecture suggested by Amazon for a Wordpress application in Amazon Web Services [30].

For a content such as images for web sites or other uses for a storage AWS provides a service called Simple Storage System (S3) [32]. Google Cloud platform offers a similar solution called Cloud Storage [12] both solutions being referred commonly as ”buckets”.

The use of a bucket can range from serving static files for a web site to database backups.

Enumerating AWS S3 or Google Cloud Storage is popular as there are multiple AWS S3 and Google Cloud Storage enumerators available [18][14]. Contents of a bucket may be configured to be either private or public. Many of the enumerators try to find misconfigured storages by brute forcing bucket names to find buckets in order to access information not intended to be public. Misconfigured buckets in some cases even allow a priviledge escalation attack [18].

Using DNS queries to enumerate buckets in both cases is not possible. For Google Storage the domain name for accessing a bucket is the same for each of the buckets (`storage.googleapis.com` [12]) and path is used to access a specific bucket with a globally unique bucket name. All S3 buckets have a globally unique name as well and the domain name for a bucket is `<bucket-name>.s3.amazonaws.com` [32].

The latter gives a hint of a possibility to brute force buckets using DNS queries but this is not possible any more [25] since every DNS query for a non-existing bucket name returns a result which is indistinguishable from a result for an existing bucket, possibly to prevent brute forcing using DNS queries. Instead enumerators available leverage either HTTP queries to bucket endpoints directly or use cloud provider command line interface software to query possible bucket names.

4 Conclusion and Future Work

The Domain Name System is an integral and critical part of the Internet infrastructure and without it the Internet would be considerably less user friendly. The Domain Name System originates from a time when the Internet was relatively new and there was not much emphasis on security. The DNS has a long history of being vulnerable to cache poisoning and DDoS attacks and still to this day the DNS traffic is unencrypted which allows passive information gathering. In addition to being vulnerable to cyber attacks such as cache poisoning, publicity of DNS records allows enumerating target network infrastructure.

Later on the different security measures to DNS have been introduced. Transmission signatures or TSIG use shared private keys to provide authentication of servers to secure actions such as zone transfers and dynamic updates of resource records. DNS Security Extensions or DNSSEC have been introduced early on to provide end-to-end data integrity and source authenticity check between the source of DNS data and the client but the adoption process is still underway. Interestingly, DNSSEC opened up a possibility for a new kind of DNS enumeration technique called Zone Walking. Zone walking was not originally considered a security threat but later on a consensus was reached that enabling

a possibility to enumerate the whole DNS zone using DNSSEC records was an undesirable feature. NSEC3 was introduced to make zone walking considerably harder but due to the increase of computing power and its easy access, NSEC3 turned out to be vulnerable to zone walking, thus NSEC5 has been proposed.

The DNS enumaration methods discussed are methods are Zone Transfer Attack, reverse DNS sweeping, Zone Walking and brute forcing subdomain names. If the Zone Transfer attack fails - which usually is the case - there's no guarantee that an attacker can map the whole target infrastructure especially in case of enumeration of subdomain names. But in addition to "traditional" DNS enumeration methods, attacker can use other methods in conjuction such as advanced search engine features and inspecting TLS certificates for a certain domain. The most comprehensive result is achieved by using multiple methods.

Even though this thesis discussed how to use DNS enumeration techniques to query DNS servers, an attacker does not have to query DNS servers directly. There are online resources that keep records of domains and their subdomains. The information available has been gathered using multiple different methods described in this study.

The focus of this study has been "traditional" DNS enumeration. Since services move to the cloud and services may consist of multiple microservices this opens up new possibilities to leverage DNS or more generally service enumeration techniques. As discussed in the Chapter 3.6 from outside services may not reveal much information but if an attacker gains a foothold in the system e.g., Kubernetes cluster, EC2 instance of Amazon Web Services or a Compute Engine instance of Google Cloud Platform, this opens up new ways for enumerating services from the inside. Unfortunately discussing this aspect was not possible due to the time restrictions but it would be an interesting topic for a further study.

References

[1] D. Eastlake 3rd and C. Kaufman. Domain Name System Security Extensions. Proposed standard, IETF, January 1997. URL http://www.rfc-editor.org/rfc/rfc2065.txt.

[2] R. Arends, R. Austein, M. Larson and S. Rose D. Massey. Resource Records for the DNS Security Extensions. Proposed standard, IETF, 2005. URL http://www.rfc-editor.org/rfc/rfc4034.txt.

[3] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose. DNS Security Introduction and Requirements. Proposed standard, IETF, 2005. URL http://www.rfc-editor.org/rfc/rfc4033.txt.

[4] D. Atkins and R. Austein. Threat Analysis of the Domain Name System (DNS). Draft standard, IETF, August 2004. URL http://www.rfc-editor.org/rfc/rfc3833.txt.

[5] Cloudflare. What is a DNS PTR record?, 2021. URL https://www.cloudflare.com/learning/dns/dns-records/dns-ptr-record. Viewed 30 March 2021.

[6] ClouDNS. What is an ALIAS Record?, 2021. URL https://www.cloudns.net/wiki/article/18. Viewed 4 April 2021.

[7] DNSimple. Alias Records, 2021. URL https://support.dnsimple.com/articles/alias-record. Viewed 4 April 2021.

[8] T. Finch, E. Hunt, P. van Dijk, A. Eden and W. Mekking. Address-specific DNS aliases (ANAME). Standards track, IETF, July 2019. URL https://tools.ietf.org/html/draft-ietf-dnsop-aname-04.

[9] Z. Gao and A. Venkataramani. Measuring update performance and consistency anomalies in managed dns services. *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 2206–2214, 2019. doi: 10.1109/INFOCOM.2019.8737568.

[10] R. Gieben and W. Mekking. Authenticated Denial of Existence in the DNS. Informational, IETF, February 2014. URL http://www.rfc-editor.org/rfc/rfc7129.txt.

[11] S. Goldberg and M. Naor. NSEC5: Provably Preventing DNSSEC Zone Enumeration. San Diego, CA, February 2015. NDSS. URL https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/shulman.

[12] Google. Cloud Storage, Quickstart: Using the Console, 2021. URL https://cloud.google.com/storage/docs/quickstart-console. Viewed 6 April 2021.

[13] A. Gulbrandsen, P. Vixie and L. Esibov. A DNS RR for specifying the location of services (DNS SRV). Proposed standard, IETF, February 2000. URL http://www.rfc-editor.org/rfc/rfc2782.txt.

[14] J. Helmus. *AWS Penetration Testing*. Packt Publishing, 2020. ISBN 9781839216923.

[15] Internet Foundation in Sweden. Access to Zone Files for .se and .nu, No Date. URL https://www.iis.se/english/domains/tech/zonefiles. Viewed 31 March 2021.

[16] Ed. J. Schlyter. DNS Security (DNSSEC) NextSECure (NSEC) RDATA Format. Proposed standard, IETF, August 2005. URL http://www.rfc-editor.org/rfc/rfc3845.txt.

[17] J.F. Kurose and K.W. Ross. *Computer networking : a top-down approach*. Pearson, Boston, 2013. ISBN 9780273775638.

[18] Rhino Security Labs. Google Cloud Platform (GCP) Bucket Enumeration and Privilege Escalation, No Date. URL https://rhinosecuritylabs.com/gcp/google-cloud-platform-gcp-bucket-enumeration. Viewed 5 April 2021.

[19] B. Laurie, G. Sissiona and R. Arends. DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. Proposed standard, IETF, 2008. URL http://www.rfc-editor.org/rfc/rfc5155.txt.

[20] E. Lewisa and Ed. A. Hoenes. DNS Zone Transfer Protocol (AXFR). Proposed standard, IETF, June 2010. URL http://www.rfc-editor.org/rfc/rfc5936.txt.

[21] S. McClure. *Hacking Exposed*. McGraw-Hill, 7th edition, 2012. ISBN 9780071780285.

[22] C. McNab. *Network Security Assessment*. O'Reilly Media, Inc., third edition, 2016. ISBN 9781491910955.

[23] P. Mockapetris. Domain Names - Concepts and Facilities. Internet standard, IETF, November 1987. URL http://www.rfc-editor.org/rfc/rfc1034.txt.

[24] P. Mockapetris. Domain Names - Implentation and Specification. Internet standard, IETF, November 1987. URL http://www.rfc-editor.org/rfc/rfc1035.txt.

[25] K. Rouwhorst. S3enum, 2019. URL https://github.com/koenrh/s3enum. Viewed 6 April 2021.

[26] H. Schulzrinne, S. Casner, R. Frederic and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. Draft standard, IETF, July 2003. URL http://www.rfc-editor.org/rfc/rfc7129.txt.

[27] Offensive Security. Dnsenum package description, No Date. URL https://tools.kali.org/information-gathering/dnsenum. Viewed 6 April 2021.

[28] Offensive Security. Fierce package description, No Date. URL https://tools.kali.org/information-gathering/fierce. Viewed 6 April 2021.

[29] Amazon Wab Services. Locations and IP address ranges of CloudFront edge servers, 2021. URL https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/LocationsOfEdgeServers.html. Viewed 27 March 2021.

[30] Amazon Web Services. Best Practices for WordPress on AWS, 2019. URL https://d1.awsstatic.com/whitepapers/wordpress-best-practices-on-aws.pdf. Viewed 5 April 2021.

[31] Amazon Web Services. How internet traffic is routed to your website or web applications, 2021. URL https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/welcome-dns-service.html. Viewed 4 April 2021.

[32] Amazon Web Services. Working with Amazon S3 Buckets, No Date. URL https://docs.aws.amazon.com/AmazonS3/latest/dev-retired/UsingBucket.html. Viewed 6 April 2021.

[33] H. Shulman and M. Waidner. One key to sign them all considered vulnerable: Evaluation of DNSSEC in the internet. *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 131–144, Boston, MA, March 2017. USENIX Association. ISBN 978-1-931971-37-9. URL https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/shulman.

[34] M. Skwarek, M. Korczynski, W. Mazurczyk and A. Duda. Characterizing vulnerability of dns axfr transfers with global-scale scanning. *2019 IEEE Security and Privacy Workshops (SPW)*, pages 193–198, 2019. doi: 10.1109/SPW.2019.00044.

[35] A.S. Tanenbaum and D. Wetherall. *Computer networks*. Pearson, Harlow, fifth edition, 2014. ISBN 9781292037189.

[36] S. Thomson, C. Huitema, V. Ksinant and M. Souissi. DNS Extensions to Support IP Version 6. Internet standard (changed from draft standard may 2017), IETF, October 2003. URL http://www.rfc-editor.org/rfc/rfc3596.txt.

[37] J. Vcelak and S. Goldberg. NSEC5, DNSSEC Authenticated Denial of Existence. Internet draft, IETF, 2018. URL `https://tools.ietf.org/id/draft-vcelak-nsec5-06.html`.

[38] P. Vixie and O. Gudmundsson. Secret Key Transaction Authentication for DNS (TSIG). Proposed standard, IETF, May 2000. URL `http://www.rfc-editor.org/rfc/rfc2845.txt`.

[39] C. Wagner, J. François, A. Dulaunoy, R. State and T. Engel. SDBF: Smart DNS brute-forcer. April 2012. doi: 10.1109/NOMS.2012.6212021.

[40] M. Wander, L. Schwittmann, C. Boelmann and T. Weis. GPU-Based NSEC3 Hash Breaking. *2014 IEEE 13th International Symposium on Network Computing and Applications*, pages 137–144, 2014. doi: 10.1109/NCA.2014.27.