Kim *et al. J Surveill Secur Saf* 2020;1:34-60
DOI: 10.20517/jsss.2020.14

Journal of Surveillance, Security and Safety

Original Article

Open Access

A survey of domain name system vulnerabilities and attacks

Tae Hyun Kim, Douglas Reeves

Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA.

Correspondence to: Prof. Douglas Reeves, Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA. E-mail: reeves@ncsu.edu

How to cite this article: Kim TH, Reeves D. A survey of domain name system vulnerabilities and attacks. *J Surveill Secur Saf* 2020;1:34-60. http://dx.doi.org/10.20517/jsss.2020.14

Received: 20 Apr 2020 **First Decision**: 9 Jun 2020 **Revised**: 13 Jul 2020 **Accepted**: 20 Jul 2020 **Available online**: 12 Sep 2020

Academic Editor: Fei Gao **Copy Editor**: Cai-Hong Wang **Production Editor**: Jing Yu

Abstract

Aim: The Domain Name System (DNS) plays an integral role in the functionality of the Internet. Clients receive Internet service by mapping domain names into internet protocol addresses, which are routable. DNS provides a scalable and flexible name resolution service to clients easily and quickly. However, DNS was initially developed without security, and the information is not secured. Although DNS security extensions was released in 1999 to protect against vulnerabilities, it is not widely deployed, and DNS continues to suffer from a variety of attacks. The purpose of this study is to provide a comprehensive survey of DNS security.

Methods: We describe an overview of DNS vulnerabilities, DNS attacks, and even mitigation systems. In detail, attacks are classified by purpose and methods for defending against these attacks are introduced and assessed. Finally, we conclude with a summary of the current state of DNS security.

Results: The main findings of this study is to introduce fundamental vulnerabilities of DNS and classify representative DNS attacks into four categories to efficiently analyze them. Moreover, we describe and assess mitigation systems to defense these attacks.

Conclusion: We conclude that DNS is an integral part of Internet operations but is still exposed to various attacks due to its vulnerabilities, low deployment of available mitigation techniques, and limitations of such techniques.

Keywords: Survey paper, Domain Name System, DNSSESC, network security, DNS attacks, DNS mitigation system

1. INTRODUCTION

Over the past 30 years, we have experienced more convenient Internet services through the human-friendly Domain Name System (DNS) functionality, which maps domain names to internet protocol (IP) addresses using globally distributed hierarchical name servers. Internet users with domain addresses can utilize various Internet services, such as web surfing, e-mail, and even mobile services without entering machine-recognized IP addresses. However, DNS was first developed without consideration of cybersecurity and caused many problems[1,2]. There is no doubt that there are many cyber attacks on DNS in the wild. In a recent attack, for instance, attackers redirected DNS lookup for MyEtherWallet.com to a malicious website that looked like an authentic website, for hijacking victims' account information[3].

To overcome such various DNS security problems (i.e., directory lookup) and reinforce cybersecurity, the DNS security extensions (DNSSEC) protocol was developed. DNSSEC implanted the digital signature mechanism of public-key cryptography into the DNS system[4-7]. DNSSEC extends DNS based on the hierarchical public key infrastructure (PKI) to protect data published in DNS. Certificates for the public keys are issued by trusted certificate authorities (CAs), which certify the ownership of the public keys. Thus, clients and resolvers can verify that DNS responses have not been forged or altered, using DNSSEC. However, DNSSEC still suffers from deployment issues in the current Internet. Chung *et al.*[8] found that 31% of domains supporting DNSSEC failed to publish all relevant records required for validation and 39% of domains used an insufficiently strong key-signing key. They also found that 82% of the resolvers requested DNSSEC records, but only 12% of them attempted to validate the DNSSEC records. Additionally, several studies have been performed to scrutinize the CA model for lack of transparency and choice of trusted CA sets[9,10]. If one of the CAs acting as a trust anchor is compromised, all information certified by the CA may be falsified.

The 2016 Dyn cyberattack was a significant event indicating serious DNS risk. Dyn, which is a popular DNS provider, was attacked by two large and complex distributed denial-of-service(DDoS) attacks against the DNS infrastructure[11]. Eventually, several major Internet services and banking systems were paralyzed. Figure 1[12] shows the map of the Internet disabling in North America by the Dyn cyberattack. An interesting issue with this attack is that a large part of the US was impacted by attacking Data Centers in only certain parts of the US. That is, the attack directly targeted only a locally distributed DNS with a local Botnet. Moreover, the Cyber Security Report[13], released in 2018, describes DNS as the largest (82%) Internet service target of application-layer attacks. Despite efforts to improve DNS's security problems, DNS is still a popular target for cyberattacks because of its essential role on the Internet, and its vulnerability.

This paper is a comprehensive survey of vulnerabilities of DNS (and DNSSEC), attacks exploiting those vulnerabilities, and mitigations proposed or deployed to address such attacks. There have been previous surveys on more restricted aspects of DNS security[14], a broader security context that includes DNS[15], or the use of DNS to combat specific types of attacks[16,17]. The contributions of this paper are: (1) first, the problems of DNS and DNSSEC security are described and classified as fundamental, structural, and systematic vulnerabilities. Also, the increasing seriousness of DNS attacks is discussed; second, various DNS attacks are discussed and classified by purpose, to understand and analyze them; finally, defenses against DNS attacks are described, and the effectiveness of current DNS attack mitigation is assessed.

The paper is organized as follows. Section 2 provides background on DNS and DNSSEC. Section 3 describes the security vulnerabilities of DNS and DNSSEC. Section 4 explains typical DNS attacks that currently threaten Internet users, assesses these attacks according to seriousness and classifies DNS attacks by purpose. Section 5 explores DNS attack mitigation methods and assesses their strengths and weaknesses. Section 6 concludes with the implications of this study and opportunities for research.

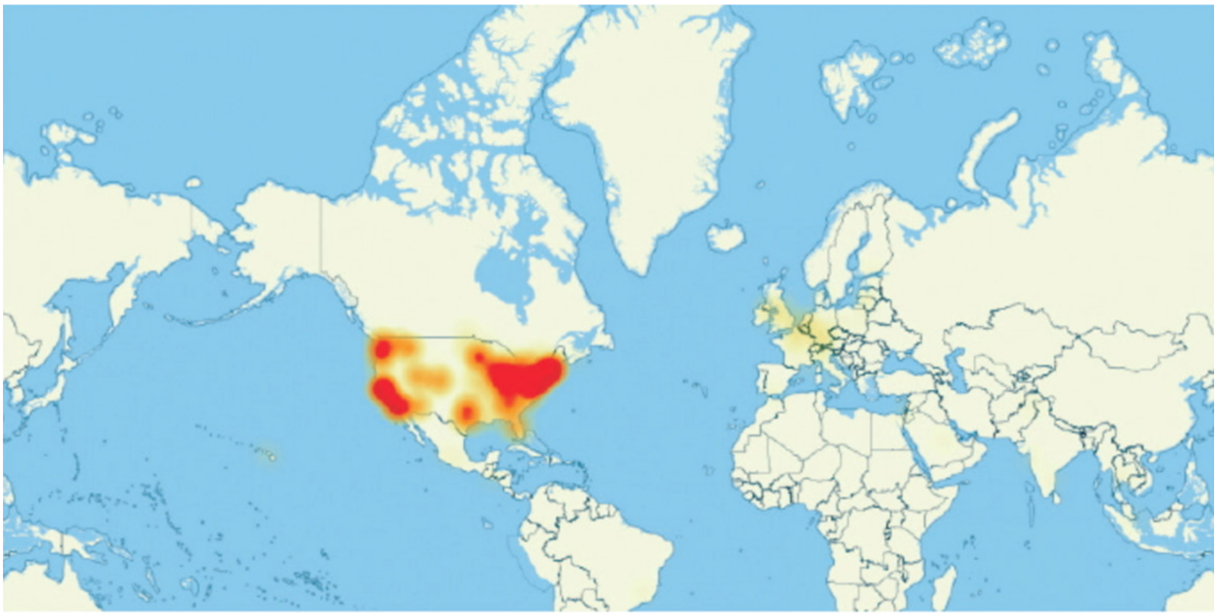


Figure 1. Map of Internet disabled in US by the Dyn Attack

2. BACKGROUND

2.1 DNS

DNS is an Internet system to map alphabetic domain names to numeric IP addresses[1,2,18]. In this paper, DNS is defined as the following:

Service: DNS is a name resolution service. The domain name can be matched to the IP address through DNS.

System: DNS is a distributed database system for the naming service as technical support. The DNS servers are located globally.

Server (Structure): DNS name servers are organized in a top-down tree structure to support an efficient naming service.

2.1.1 DNS history

In 1983, domain names were first translated to addresses through a local service, managed by the Operating System (OS). The host file in the OS stored these translations. Initially, only about 15 organizations used a single network, so keeping these files consistent and updated was straightforward, but not scalable. To address this inefficiency, the Stanford Research Institution Network Information Center (SRI-NIC) developed a new naming mechanism. The previous name service within the OS was transformed into a system that was managed and deployed collectively by SRI-NIC. The host file containing translation information (host name and numeric address) was hosted online by SRI-NIC and could be downloaded over FTP. However, as the Internet grew the difficulties of keeping the file updated, and the size of the file, became impractical. This resulted in poor search performance and traffic bottlenecks. To overcome these drawbacks, a new type of name system was introduced in 1987 as the IETF Request for Comments (RFC) 1034[2]. The DNS system was standardized and widely implemented and started to manage domain names on hierarchically-organized servers, growing into the current DNS system.

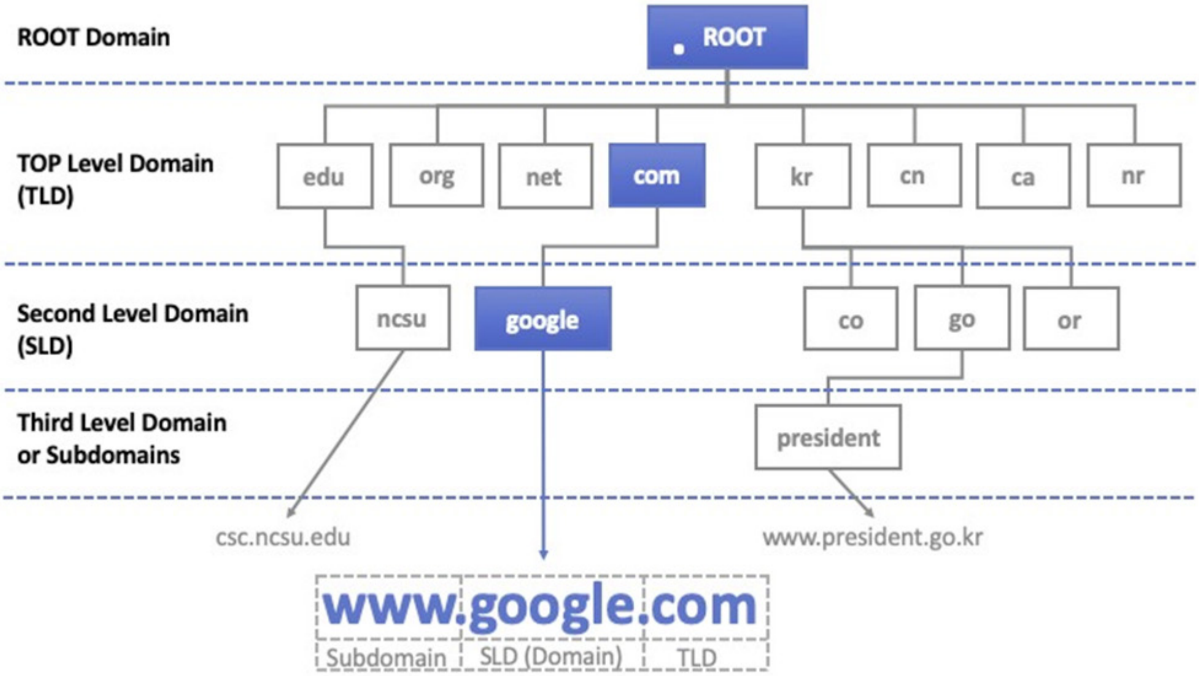


Figure 2. Domain Name System structure

2.1.2 DNS philosophy

Technically, DNS is a hierarchical name server system that uses a globally distributed database system that holds information about each domain. The DNS information is stored in distributed DNS servers, and the information can be searched at any time upon user request.

Figure 2 illustrates the hierarchical DNS structure via a common domain name. DNS begins with the .(Root) domain at the top. .com is a TLD (Top Level Domain) whose parent is the .(Root) domain. .google is an SLD (Second Level Domain) whose parent is the .com domain. Finally, .www (i.e., a web service) is a subdomain of .google.com.

As the top level of DNS, Root name servers are a global network with 13 redundant servers located in various countries, which manage all TLDs. The TLD comprises two types: the country code Top Level Domain (ccTLD) and the general Top Level Domain (gTLD). The ccTLD stands for the country domain name, such as .kr (South Korea) and the gTLD stands for the general domain type, such as .com (Company) or .org (Organization). As the number of domains increased, the number of available TLDs became insufficient, and ICANN announced a new set of TLDs in 2014. Currently, the number of TLD servers around the world is approximately 1,500 (maintained by IANA). Such vertical tree structure enables DNS not only to facilitate the management of each domain information but also to distribute numerous DNS requests efficiently.

The process of translating IP addresses to corresponding domain names through DNS is called name resolution or DNS resolution[1]. DNS resolution begins with a client's DNS request. Figure 3 illustrates how a client obtains the IP address for a web server via DNS resolution, allowing it to receive web services.

(1) A client requests an IP address www.google.com from a local recursive DNS resolver.
(2) The recursive DNS resolver first checks the address translation in its local cache.
(3) If there is no information in the cache, the recursive DNS resolver requests the IP address of the TLD

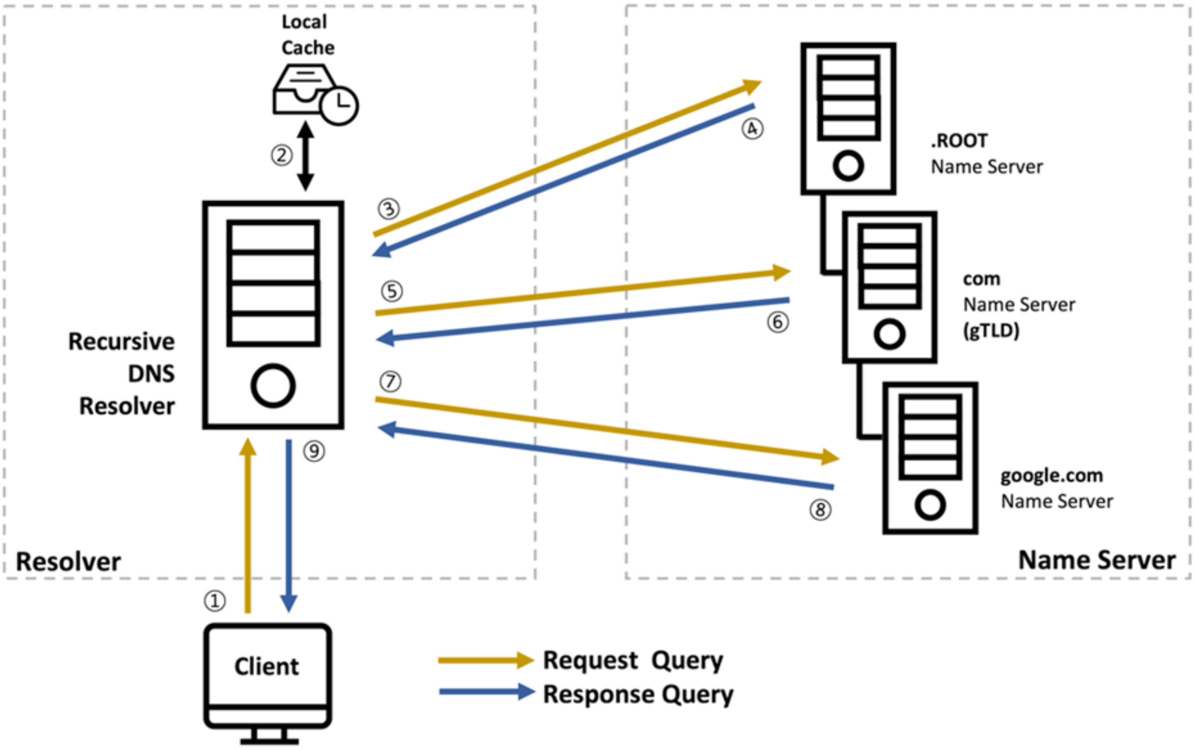


Figure 3. DNS architecture. DNS: Domain Name System; gTLD: general Top Level Domain

nameserver from the Root name server.
(4) The Root name server sends back the IP address of the .com name server as a response.
(5) Using this IP address, the recursive DNS Resolver requests the IP address of the SLD nameserver from the .com name server.
(6) The .com name server sends back the IP address of the .google.com name server as a response.
(7) With the IP address, the recursive DNS Resolver requests the IP address for www.google.com from the .google.com name server.
(8) The .google.com name server sends back the own IP address of www.google.com to the recursive DNS resolver.
(9) The recursive DNS resolver sends back the IP address of www.google.com to the client as a response. Finally, with the IP address (172.217.7.197 in this example), the client connects to the www.google.com server.

The DNS framework consists of the following three parts:
(1) Client: They request IP addresses with domain names through the stub resolver, a client of DNS, and transmits the request to the local DNS server address set on its device.
(2) Local DNS Server (Recursive DNS Resolver): They receive the DNS query from clients and obtains the IP address for the domain name from domain name servers. Also, the IP address once found is stored in memory for a certain period. So, it is called Caching Resolver.
(3) Domain Name Server (Authoritative Name Server): They have and manage IP addresses for the domain names as well as the information related to the IP addresses. The Authoritative Name Server is composed of more than 3-levels (Root, TLD, Lower-level Domain). Each domain server consists of a single master server and several slave servers.

In addition to the basic information regarding IP addresses for domain names, DNS databases provide additional information for a variety of services. DNS resource records (RR) have additional information

related to domain names as a DNS server database element, which is used to respond to DNS client queries. RRs are added to the DNS namespace generated by the DNS server and consist of various types, including the following:
(1) A and AAAA: A - IPv4 address or AAAA - IPv6 address.
(2) CNAME (Canonical Names): domain name aliases, used for mapping an alias to a domain name.
(3) NS (Name Server): indicates a specific authoritative name server or a name server address.
(4) Others: MX (Mail Exchange) - mapping the domain to an SMTP email server, PTR (Pointer) - Reversing IP address to Domain name resolution (reverse DNS lookup), and TXT - readable information.

2.1.3 DNS limitations

The major vulnerability in DNS is the lack of security. The original DNS protocol did not consider this issue in depth. Thus, DNS data could be forged to translate to a malicious IP address, so that Internet users would connect to a non-authorized site. This could, for example, be used to distribute false information or to surreptitiously collect personal information. DNS does not provide a way to verify that the received IP address translation is authentic. A corrupted or intercepted DNS response may provide false information to any requester. DNSSEC has been developed to overcome this fundamental security vulnerability of DNS[4,7].

2.2 DNSSEC

DNSSEC, which is an Internet standard technology, aims to eliminate this vulnerability of DNS. DNSSEC was originally standardized in 2005 as IETF RFCs 4033 through 4035[4-7]. Using two keys - the Zone Signing Key and Key Signing Key (KSK) - to create digital signatures with Public Key Cryptography, DNSSEC guarantees integrity and authentication for DNS data.

2.2.1 DNSSEC purpose

DNSSEC significantly enhances DNS security by adding Public Key Cryptography to the existing DNS. The DNS cache poisoning attack, for instance, configures an ISP's local DNS resolvers and their cache to map specific domain names to malicious IP addresses. As a solution to such DNS fundamental security problems, DNSSEC provides strong authentication using digital signatures, based on Public Key Cryptography[4,7].

2.2.2 DNSSEC philosophy

Figure 4 shows the basics of data authentication using public-key cryptography.
(1) Alice generates an asymmetric key pair, composed of a Public and a Private key.
(2) Alice distributes the Public key to the Internet.
(3) Alice creates "signature" by signing the plain text with her Private key.
(4) Alice transmits "signature" along with "original data" to Bob.
(5) Bob receives "original data" with "signature" from Alice
(6) Bob looks up the Public key of Alice
(7) Bob performs the signature validation of "original data" with "signature", using Alice's Public key.
(8) If the signature is successfully verified, then Bob is assured that the original data purportedly from Alice is correct.

DNSSEC applies the digital signature mechanism to resource records (RRs) to protect the data itself, which is set in each section of the response message. DNSSEC has added four new RR types to existing DNS records; these are Resource Record Signature (RRSIG), DNS Public Key (DNSKEY), NSEC/NSEC3, and DS. These record types support the digital signatures and the signature verification process[6,19].
(1) RRSIG: This RR has a signature for a DNSSEC-secured record set.
(2) DNSKEY: This RR contains the public key to verify the signature in RRSIG records.
(3) NSEC/NSEC3: This RR is for the explicit denial-of-existence of a DNS record.

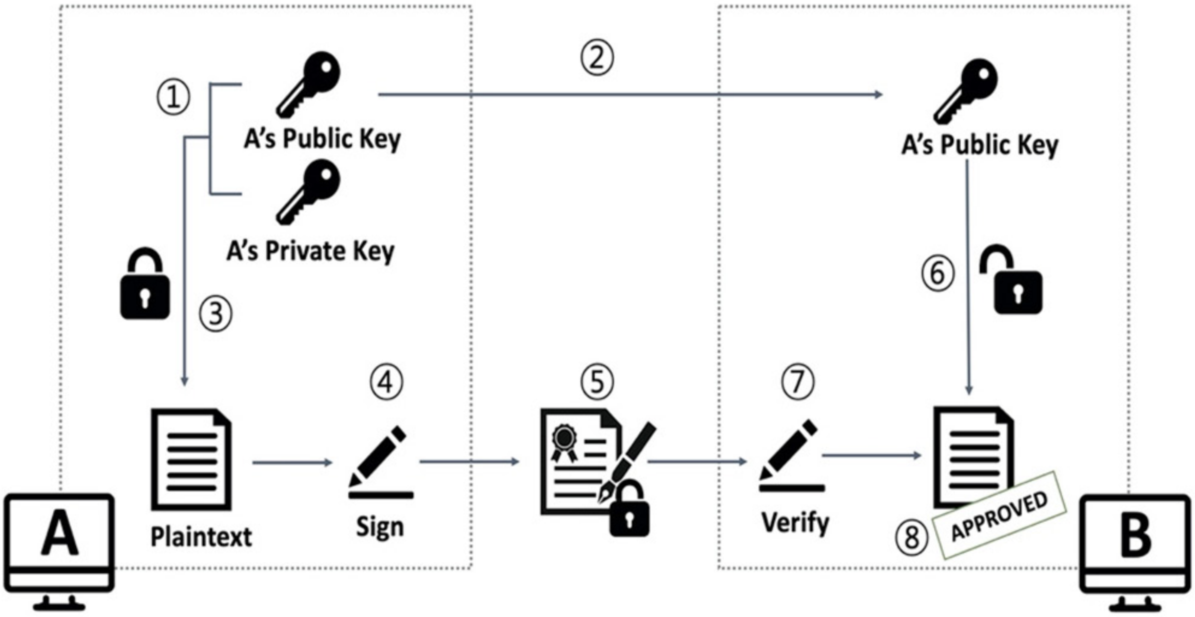


Figure 4. Public Key Cryptography Architecture

(4) DS (delegation signer): This RR holds the name of a delegated zone. The DS record is placed in the parent zone along with the delegating NS records for the authentication chain between the parent zone and child zone.

The DNSSEC protocol uses a Chain of Trust due to a strong, reliable connection between DNS servers. Figure 5 shows how DNSSEC works as the Chain of Trust. Compared with Figure 3, the IP address request of DNSSEC is the same as that of DNS. However, the verification process is added to the existing DNS. DNS servers verify each other with digital signatures from trusted CAs. Thus, DNS servers maintain a strong security chain between each other to guarantee the integrity and authentication of DNS data[7].
(1) A DNS resolver first sets a "Trust Anchor" that corresponds to the public key from a Root domain zone, as the KSK over DNSKEY record.
(2) The "Trust Anchor" is the starting point for verifying the signature in the signed DNS data, as the basis for ensuring "Trust" for Data Integrity.
(3) The DNS resolver performs signature verification from the Root domain zone to the A record data, which is the final node of verification, and then trusts the data.

DNSSEC adds strong security to authenticate DNS responses. Thus, DNSSEC assures users where the DNS data originated from, that is not forged in transit, and verifies whether a domain exists or not.

2.3 Multicast DNS

The multicast DNS (mDNS) protocol, described by RFC 6762[20], is a DNS service to resolve the hostname to IP address in small networks without a local name server. Unlike conventional unicast DNS, mDNS uses the IP multicast user datagram protocol (UDP) packet. Thus, every node on the network subscribing to that multicast address receives the request to resolve a hostname. The host owning that domain name responds, also using multicast, with its IP address. All nodes subscribing to the multicast address can update their DNS cache with the response. Figure 6 illustrates the basic mDNS protocol.

With the advent of IPv6 and the use of numerous embedded devices (e.g., IoT devices) greatly increasing, the normal, somewhat complex DNS infrastructure is inconvenient for local services configuration. To

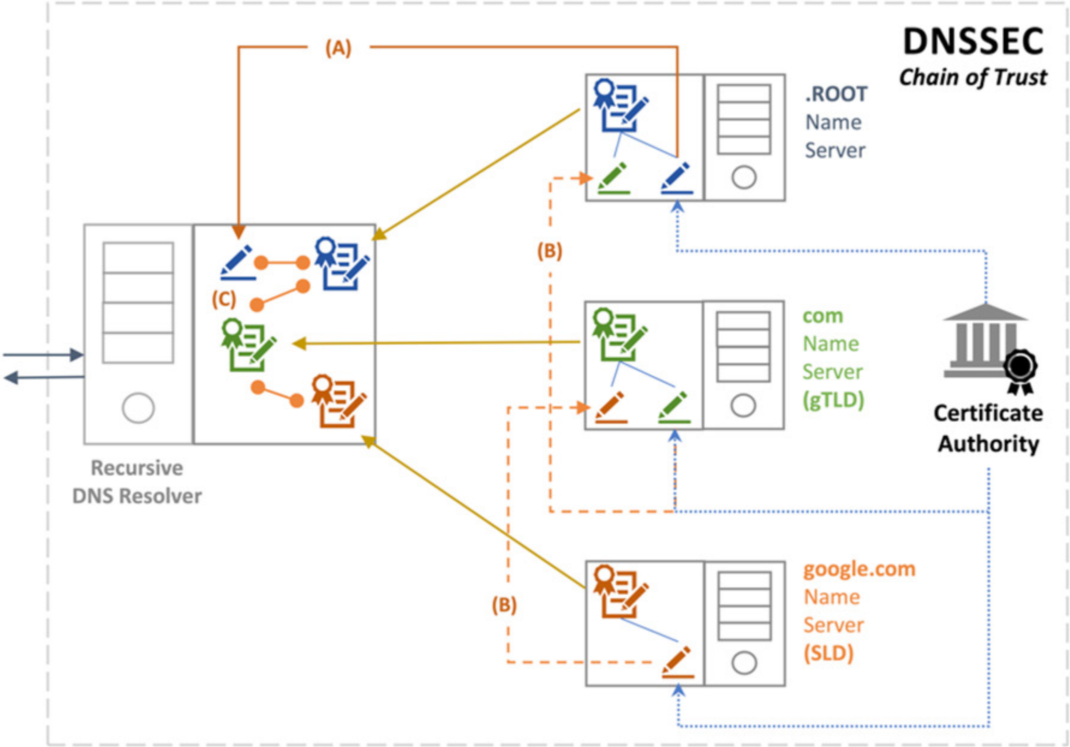


Figure 5. DNSSEC Architecture. DNS: Domain Name System; DNSSEC: DNS security extensions

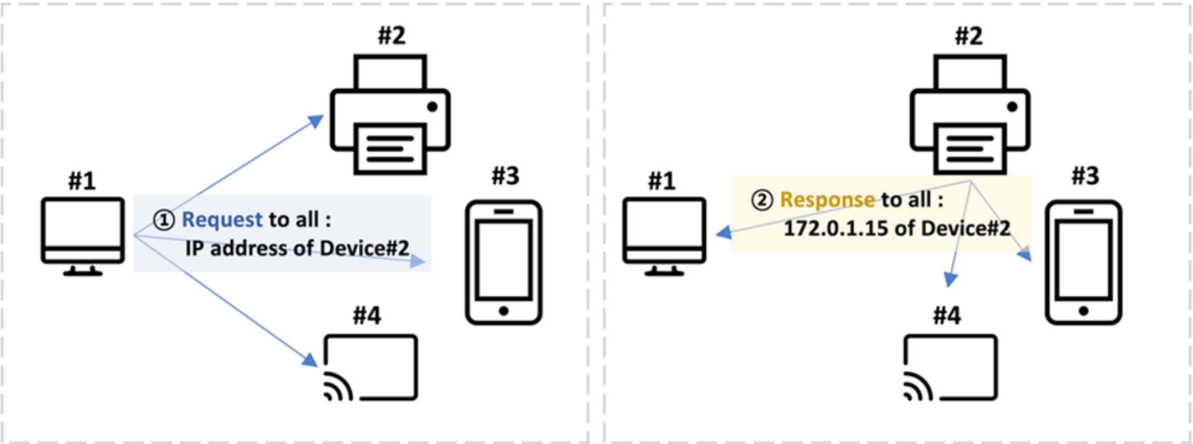


Figure 6. multicast DNS Architecture. DNS: Domain Name System

address this problem, mDNS was implemented by Apple Bonjour[20] and the Microsoft Link-local Multicast Name Resolution[21]. Initially, mDNS was intended to search for printer devices within a network but later expanded to the ability to resolve local hostnames.

The major benefits of mDNS are a zero-configuration and no infrastructure. It is available without conventional DNS settings and does not require a local name server. Also, users can connect and use devices in the network more conveniently because access to devices is intuitive.

mDNS has several weaknesses. First, if mDNS is exposed to the Internet, an attacker can easily collect information about devices and services on the network. Multicasting is inherently a powerful means of mounting Denial of Service attacks. Since mDNS is a UDP-based protocol, it can be vulnerable to amplification attacks using mDNS queries, and spoofing attacks are trivial.

3. VULNERABILITIES

Cybersecurity is a defense mechanism to protect the system from various malicious attacks; cyberattacks disable or avoid these defenses. Vulnerabilities or weaknesses enable such attacks. This section looks specifically at DNS and DNSSEC vulnerabilities.

3.1 DNS vulnerabilities

DNS vulnerabilities can be viewed in 3 ways: by concept, by structure, and by communication.

3.1.1 Conceptual view

The CIA Triad is a conceptual model of information security, consisting of three factors: confidentiality, integrity, and availability[22]. The following is an assessment of DNS in terms of information security.

(1) Confidentially: DNS requests and responses are in most cases sent via the UDP protocol, which is light and fast, but normally unencrypted, allowing eavesdropping on all messages. Besides, the information stored by DNS servers is necessarily public, as name to address bindings must be served on demand.
(2) Integrity: DNS without modification does not have a mechanism sign data cryptographically, which is its single greatest weakness; anyone can tamper with or forge DNS data.
(3) Availability: the hierarchical structure of DNS, unless augmented with redundancy, is very much subject to attacks on DNS servers, or to failures of those servers.

3.1.2 Structural view

DNS servers have a hierarchical tree structure ranging from the Root to a specific domain name server. However, such a DNS feature includes structural problems, which can affect DNS vulnerabilities. The structural problems in DNS are as follows:
(1) Lack of redundant DNS[23]: The hierarchical DNS structure distributes and processes DNS queries more efficiently. Users can request an IP address of the desired domain step by step and obtain the response. Although DNS is designed to be distributed, traffics is concentrated because of the centralization. The centralized DNS structure makes it easier for an attacker to attack multiple Internet services used by many Internet users. For example, in 2016, a DYN attack exploiting such vulnerability made many users unable to receive normal DNS responses, as well as Internet services unavailable[11]. DNS above the SLD level, and major domain nameservers, have evolved over the years into a highly redundant system through numerous studies and cases. However, lower-level DNS servers remain exposed to threats due to a lack of redundancy. Resilient and reliable DNS support is possible if more domains adopt and support secondary DNS configurations[23].
(2) DNS server information exposure[24]: Because the fundamental security configuration of the DNS server is insufficient, the server information (e.g., server list, version) can be exposed through DNS servers of many companies. If such information is exploited, not only DNS operation but also server operation inside the companies can be exposed to the risk by attackers. The leakage of DNS server information allows malicious DNS data to be sent and the user to trust wrong DNS information. Additionally, attackers can collect information by reconnaissance attack and finally attack the server. Therefore, the security configuration of restricted server information transmission needs to be set up in each company's DNS servers.

3.1.3 Communication view

Responses to queries are only weakly protected in DNS. DNS uses the IP address, destination and source port numbers, and transaction ID in responses to match them with queries. It is relatively straightforward

for attackers to craft responses that pass these tests, as follows:
(1) No secured packet through UDP[25]: The basic query of DNS is delivered over the UDP protocol, which is unencrypted. An attacker could first capture a DNS query packet and forge a response from the name server in a malicious response before the resolver receives a valid response. This attack is made easier if routers are subverted as well.
(2) Transaction ID prediction[26]: The transaction ID is unique among several parameters that match DNS responses to requests. However, if the transaction ID is predictable, it makes it easier to forge a DNS response. The transaction ID is a 16-bit field in the DNS header and issued by the DNS algorithm. The ID value has a range of 32,768 values, but it is easier to predict if DNS randomization is poorly done (e.g., overload in cache). It is also predictable just by observing the request ID. Thus, attackers can easily guess the transaction ID and have their DNS response accepted as valid. For Berkeley Internet Name Domain (BIND) versions 4 and 8, a sequential transaction ID method is used, allowing the response ID simply to add 1 to the request ID. BIND version 9 and later adopts all randomized transaction ID and does not re-use the same ID for the same domain name. and predict the next transaction ID.
(3) Caching problems[27]: Caching is used for DNS efficiency. By storing the IP for the domain for a period of time, unnecessary IP address requests and access time to that domain can be reduced. Cache Poisoning, a typical DNS attack using such vulnerability, is one of the major threats to DNS. In cache poisoning, an attacker injects a malicious IP address into the DNS cache, causing users to receive false translation information for an extended period.
(4) Lack of protection against DDoS: About 93% of all cyberattacks on the Internet are reported as DDoS attacks[13]. DNS is also vulnerable to this attack. If DNS request floods occur, the DNS name server that handles the requests cannot respond to all requests making DNS service unavailable. As a consequence, all users using the DNS name server are unable to use the Internet. Due to the absence of a mechanism to block and prevent such attack patterns, DNS is currently suffering from many DDoS attacks.

3.2 DNSSEC vulnerabilities

As shown in Section II, DNSSEC has enhanced security for authentication and integrity by adding digital signatures using public and private keys to existing DNS to overcome known DNS vulnerabilities. However, DNSSEC is still suffering from various attacks through vulnerabilities and limitations.

3.2.1 Overhead

DNSSEC adds four record types to the DNS: RRSIG, DNSKEY, Delegation Signer (DS), and Next Secure (NSEC). Because of these extended records, DNSSEC requires more overhead than traditional DNS and increases processing time and packet size. The size of the DSSEC packet is up to 2000 bytes, while the UDP size specified by the RFC is 512 bytes. Therefore, the packets in DSSEC are fragmented, which may result in DNS fallback. For example, if the fragmented DNSSEC packets are not delivered properly and a public key that was previously verified during a key rollover is still stored in the local cache and a DNS data packet signed with a new key is received, verification of the new packet will eventually fail and be ignored. As a result, the user is provided neither with the DNS service nor authentication[28].

3.3.2 Complexity

The implementation of DNSSEC has been found to have problems in deployment. Misconfiguration may be increased because DNSSEC significantly increases the complexity of the existing DNS infrastructure[29]. The misconfiguration may result in incorrect DNSSEC RRs and authentication problems such that the data is regarded as fake, even though it is correct, causing name translation to fail[30].

3.2.3 Untrustworthy resolver

Assuming a reliable DNSSEC system is built on DNS, most of the DNS responses are trustworthy. However, if there are unreliable resolvers to deliver the final DNS response provided by the secure DNS server, Internet users are exposed to DNS threats despite the robust DNSSEC[31]. Usually, most people do not consider how much they trust the local DNS resolver that is set up for them but simply use the default local DNS resolver provided by the network. For example, if a typical user connects to the Internet over public Wi-Fi, the DNS resolver is automatically configured as the default. Exploiting such a problem, an attacker may intercept the request and configure a malicious DNS resolver that delivers false DNS data to the victim. To counteract this, the chain of trust should be extended from the DNS resolver to the users. Dynamic Host Configuration Protocol (DHCP) with authorization tickets is one way to identify DNS resolvers that are trustworthy[32]. However, if the DHCP server is disabled, or untrustworthy itself, all users in the network could be affected.

3.2.4 Zone list exposure

The DNS database is broken into zones of records. Each zone contains not only a domain's records but may also contain its subdomains and related records. DNSSEC has a security function that can digitally prove a domain or resource record that does not exist, using the NSEC (Next Secure) record type. This, however, makes it possible for an outsider to find the names in an entire zone, a process known as zone enumeration. To address this issue, the standardization of the NSEC3 RR has been completed, but can still be seriously impacted by malicious NSEC3 and DNS servers that do not implement the standard[33].

Also, zone transfer is used to synchronize zone files between primary and secondary DNS servers. To synchronize zone files between DNS servers, it is often accomplished using NFS, or a specialized zone-transfer function. Although zone file transfers are necessary, misconfiguration of the transfer may pose a serious threat of leaking information.

3.2.5 Low deployment of DNSSEC

DNSSEC provides much stronger security for DNS, but it is currently plagued by the slow deployment of DNSSEC. According to an Internet Society Report in 2016[34], TLDs zones signed with DNSSEC were about 90%, while SLDs were only 65% of DNSSEC-enabled zones. In addition, considering that the usage of DNSSEC-validating resolvers is approximately 26%, the percentage of deployment might be lower. The report also points out that DANE's deployment, which enhances the DNSSEC's vulnerability, is also low.

3.2.6 Amplification and reflection DDoS threat

DNSSEC is still a possible vehicle for amplification and reflection attacks[35]. Due to the additional information caused by complex digital signatures, DNSSEC's record is significantly larger than a normal DNS response. On average, the size of an "ANY" response from DNSSEC is 28 times larger than a normal DNS "ANY" response[36], making amplification and reflection attacks even more damaging.

4 ATTACKS

This section presents the state-of-the-art for DNS attacks, classifies, and assesses them. Generally, the DNS attack is an attack that targets multiple DNS servers on the Internet, using the DNS and DNSSEC vulnerabilities described in the previous section. The goal of the DNS attack is to deplete the targeted system resource or to corrupt the data, make the DNS system unavailable, or exploit the system to achieve the final attack. As of now, the attacks are received considerable attention from researchers, governments and also industry, but they still cause a significant risk for Internet users.

DNS attacks may be separated into four categories: DNS data tampering, DNS data flooding, abuse of DNS, and DNS server structure. Figure 7 shows the list of 11 DNS attacks that are categorized.

4.1 DNS data tampering

DNS Data Tampering occurs when an attacker hijacks and/or compromises unencrypted DNS data in the middle between users and DNS servers, and then users receive false address translation information. The

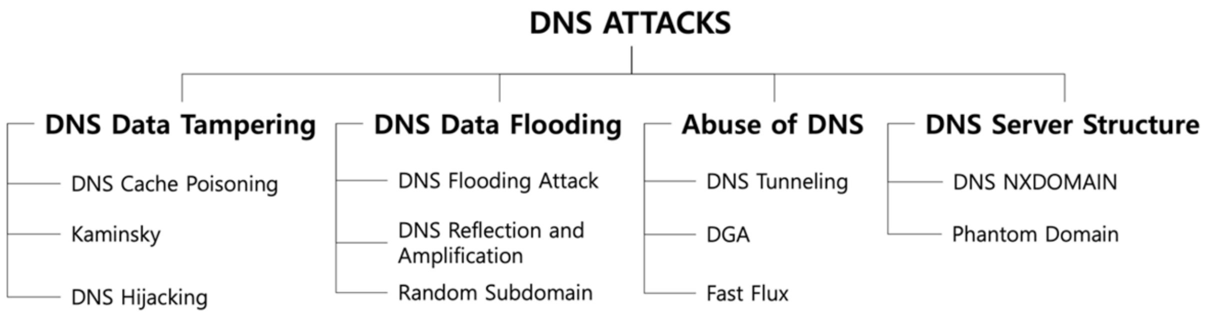


Figure 7. 11 DNS attacks. DNS: Domain Name System; DGA: domain generation algorithm

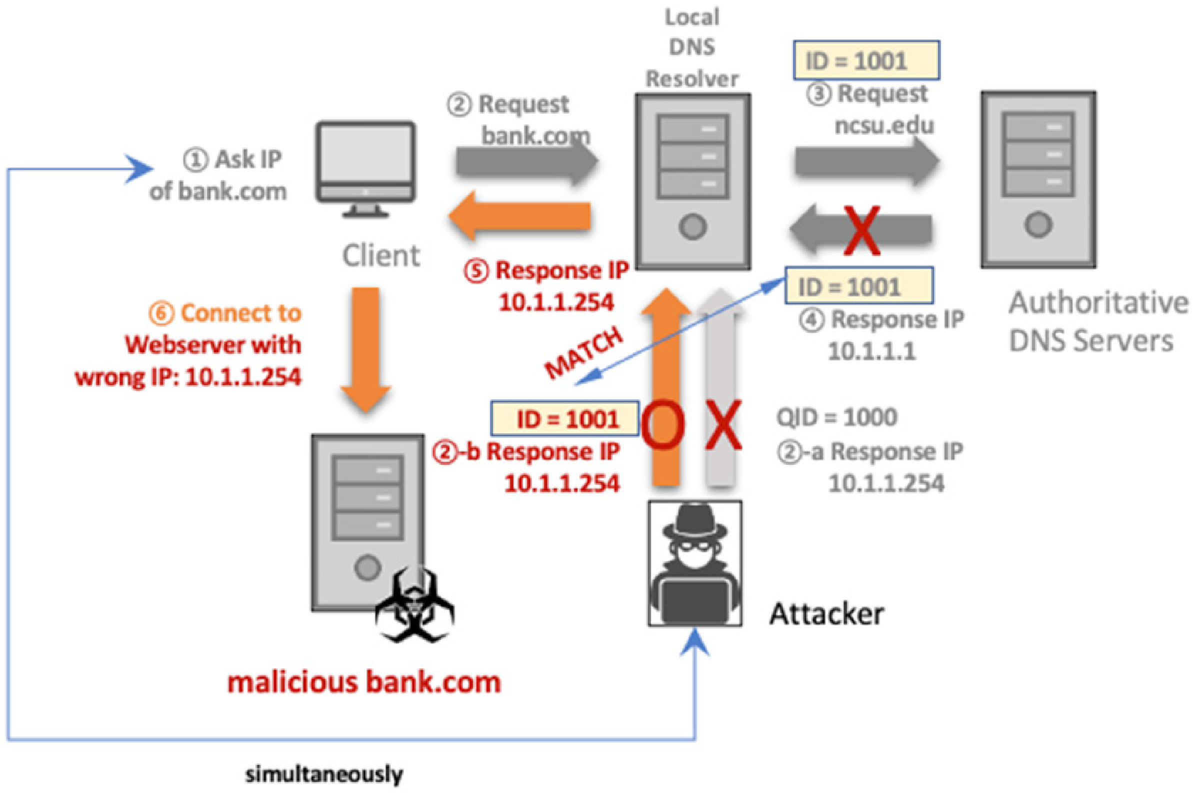


Figure 8. DNS attack: DNS data tampering. DNS: Domain Name System; QID: Query ID; "-a", "-b": the process order

attack is based on the vulnerability of insecure DNS data. Figure 8 shows how a typical DNS data tampering attack occurs. DNS attacks using data tampering are listed below.

4.1.1 DA01. DNS cache poisoning

DNS cache poisoning attack corrupts the data in the DNS cache. An attacker first queries a recursive DNS server for a domain. If the recursive DNS server (A) does not have an IP address corresponding to the requested domain in its cache, A sends queries to the authoritative name server (B). Before B can send an NXDOMAIN response, the attacker sends a large number of spoofed responses to A that appear to come from B. If the DNS response matches the DNS query, A will accept a spoofed response from the attacker and keeps the resource records (RRs) provided in that response in its cache. At a later time, a user asking for the translation of this same domain name contacts the A, which will provide the cached malicious IP address to the user[27].

Alharbi *et al*.[37] did a study on the risk of client-side DNS cache poisoning attack and discovered that a new type of DNS poisoning attack using vulnerabilities to caching within the end-user's operating system is feasible. Such vulnerability is still exposed because the client side is not considered as part of the DNS framework and, therefore, not considered in mitigations to the DNS cache poisoning attack.

4.1.2 DA02. Kaminsky

To protect against conventional cache poisoning attacks, DNS resolvers use a technique known as "bailiwick checking". To protect against malicious DNS additional records, the DNS resolver accepts only basic information and ignores additional information. To overcome this, attackers exploited the authoritative name server to poison resolver caches. Dating from Steven Bellovin's study in 1990, DNS hijacking and poisoning attacks developed into attacks based on the "birthday paradox", and eventually evolved into Kaminsky attacks in 2008[14,38].

Kaminsky attack hijacks the authoritative records instead of RRs. To succeed in the attack, the attacker should configure a domain name server that is authoritative for the malicious website zone, including all records, as a precondition. Kaminsky attack consists of two steps: Step 1: The attacker requests fake DNS queries about a random name within the target domain to local DNS servers. Since the local DNS server does not have the information in its cache, it will generate subsequent queries to authoritative name servers. Step 2: The attacker sends a barrage of forged answers to the local DNS server. Instead of fake RRs, it delegates to another name server, using the malicious authority record.

Finally, an attacker owns an authoritative name server for the specific website and provide users with malicious IP addresses for normal DNS requests of the domain through the DNS resolver. This attack is a higher level of attack than DNS Cache Poisoning Attack because it can affect not only the domain but also the subdomain.

4.1.3 DA03. DNS hijacking

DNS hijacking modifies DNS record settings (most often at the domain registrar) to point to a bogus DNS server or domain. Attackers hack the vulnerable DNS servers to change the IP address and the mapped domain address[39]. Cisco Talos discovered a new DNS hijacking attack called "DNSpionage"[40]. The main feature of this attack is to keep it as inconspicuous as possible during the attack. DNSpionage uses malicious Microsoft Office files with embedded malware, which provides HTTP and DNS communication with the attackers. Finally, malicious DNS redirection works when a user opens a forged document or malicious site. The main feature of this attack is to be as inconspicuous as possible during the attack.

4.2 DNS data flooding

In general, the goal of flooding attacks is to disable the user-server function by overwhelming the server, thereby hampering the DNS name resolution for its zone. Through the DNS data flooding attack, the attacker tries to exhaust server resources with an enormous amount of apparently valid queries, overwhelming server resources, and impeding the server's ability to respond to legitimate requests. Figure 9 describes the specific method of DNS data flooding.

4.2.1 DA04. DNS flooding attack

DNS flooding attack attempts to exhaust server-side resources through a flood of UDP requests from multiple machines contaminated by malware. DNS servers, which rely on UDP protocol for name resolution, may not be able to distinguish large UDP packets from normal requests. Attackers send a large volume of packets, mimicking legitimate DNS requests to a DNS server, causing the DNS server to run out of resources to handle legitimate requests[41].

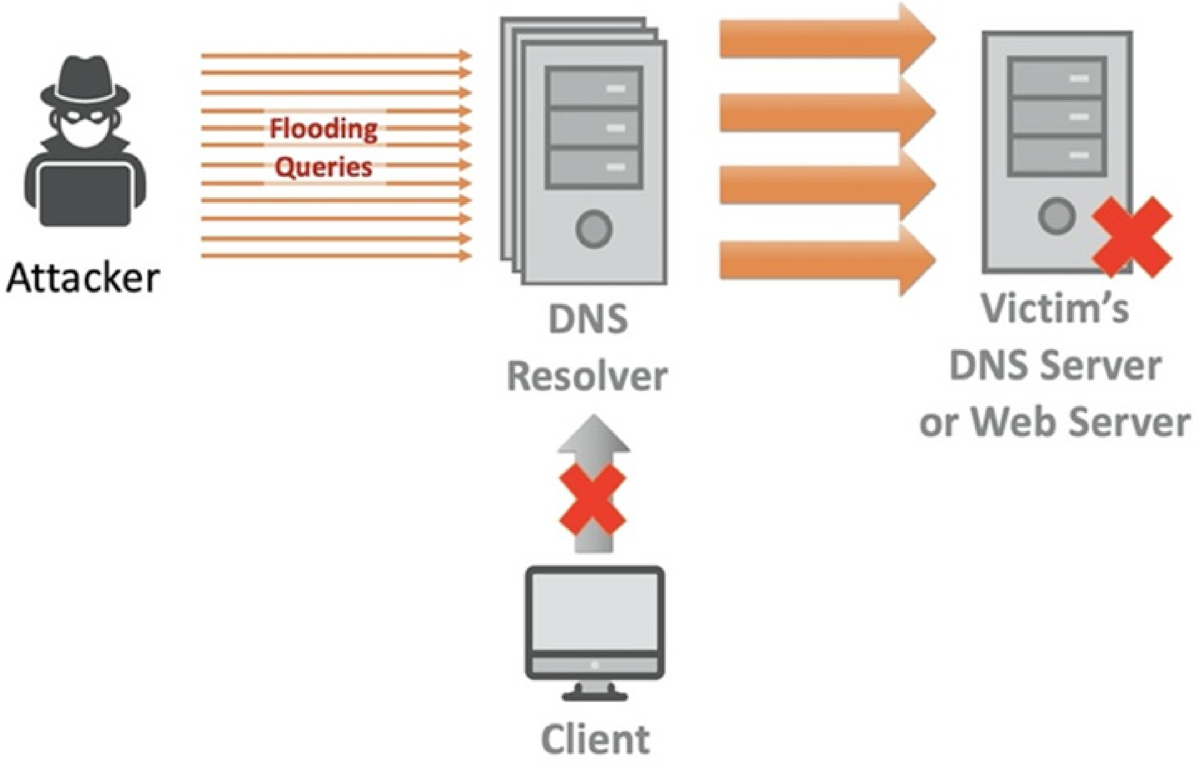


Figure 9. DNS attack: DNS Data Flooding. DNS: Domain Name System

4.2.2 DA05. DNS reflection and amplification DDoS attack

The obvious difference between DNS reflection/amplification DoS attack and DNS flooding attack is in the target of attacks[42]. While DNS flooding attack depletes DNS server's ability, DNS reflections and amplification attack attempts to saturate network capacity with heavy bandwidth traffic. This attack takes advantage of the vulnerability of third-party open resolvers in the network that combines reflection and amplification. An attacker sends out small request queries to multiple open recursive DNS servers, with a spoofed source IP address. The request is crafted to cause a large response packet. Through simultaneous reflection and amplification attack, the open recursive DNS servers generate a number of legitimate DNS responses, and finally, the victim server is attacked by DDoS. To mitigate such a DNS amplification attack, several security guidelines[43] have been issued, but still, amplification attacks have been widespread in recent years.

4.2.3 DA06. Random Subdomain

The random sub-domain attack is another type of DNS data flooding attack, sending a flood of randomized DNS requests for non-existent domains[44]. To succeed in the random subdomain attack, an attacker first infects numerous clients. Infected clients create request queries by adding randomly generated subdomain strings to the victim's target domain. Each client sends these numerous queries to a DNS recursive server, which attempts to resolve them with another server. Because this server continuously responds that the domain is nonexistent, the requests for random lookups eventually exhaust the limited resources, which delays or stops responses of legitimate lookups and all domains under the DNS server control. These attacks are used for DDoS attacks against domain name servers.

4.3 Abuse of DNS

The latest cyber attacks are active in botnets using Command Control (C&C) servers. A C&C server is a server that controls communication between attackers and zombie PCs (called Botnets) to attack a target.

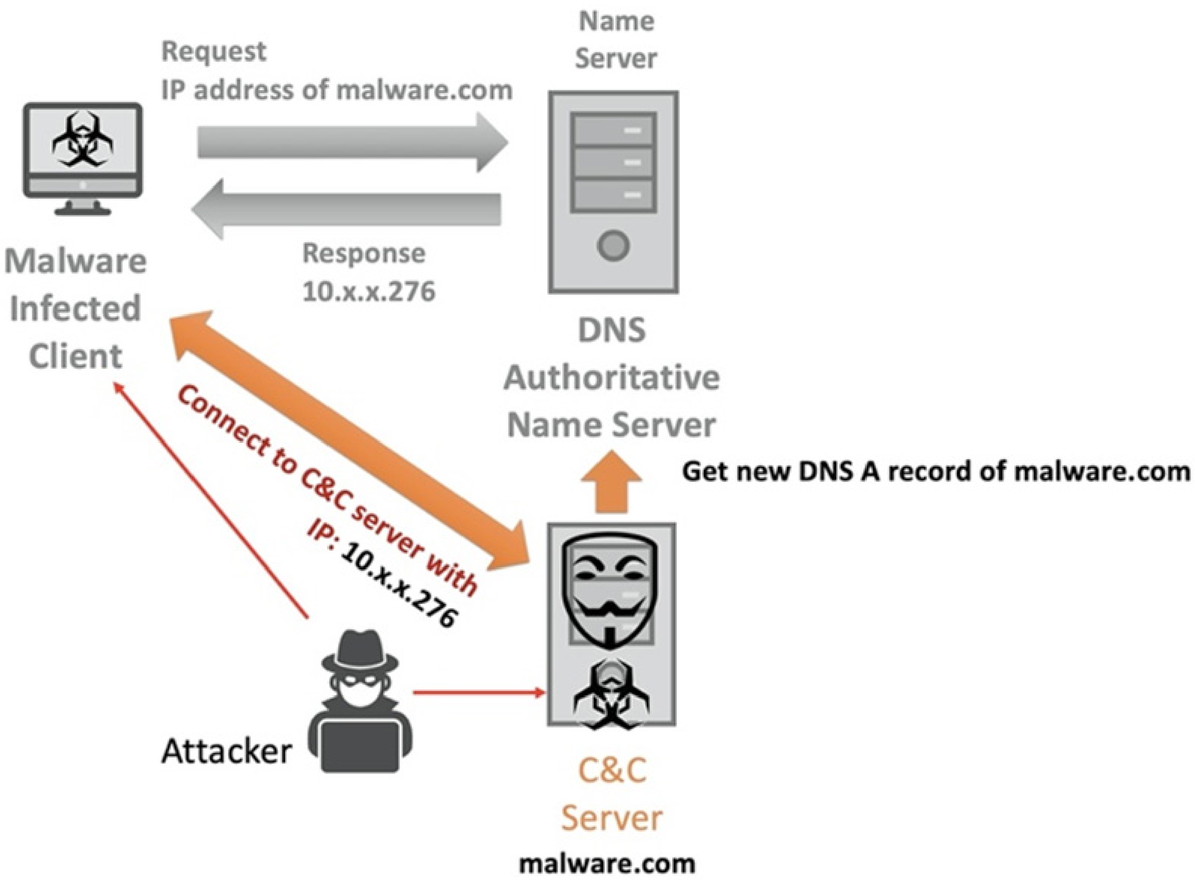

Request
IP address of malware.com
Name Server
Response
10.x.x.276
Malware Infected Client
DNS Authoritative Name Server
Get new DNS A record of malware.com
Connect to C&C server with
IP: 10.x.x.276
Attacker
C&C Server
malware.com

Figure 10. Demonstrating the DNS attack using abuse of DNS. DNS: Domain Name System

An attacker uses a C&C server to make it difficult to find the source of an attack and to scale to large numbers of bots. To counteract the development of methods for detecting C&C servers, an attacker exploits DNS to hide the location of C&C servers or to exfiltrate traffic to conceal the attack. To bypass firewalls, an attacker attempts to send malicious commands from inside a network to an external C&C server. In such a case, an attacker could conceal the information of the C&C server by using seemingly innocuous DNS (DNS TTL, NXDOMAIN) records, as shown in Figure 10.

4.3.1 DA07. DNS tunneling

DNS Tunneling is a type of bypass technology that allows an attacker to send attack commands and receive the results without blocking by the defense system. DNS requests may use up to 255 characters for a domain name, and subdomains separated by "." can be up to 63 characters. For example, if an attacker sends a DNS query of "ghAAAAATTTAAAACCCKKakdg.malware.com", the malware.com name server, as the C&C server, accepts the query as a malicious attack command. Conversely, the malware.com name server exploits records (A, CNAME, TXT) of the DNS response query to include the results for that attack command. Since an attacker and a C&C server communicate through DNS port 53, DNS tunneling may evade a defensive system[45,46].

4.3.2 DA08. domain generation algorithm

Domain generation algorithm (DGA) is an algorithm that randomly generates a large number of domains (from hundreds to tens of thousands)[47]. An attacker uses DGA to support malware attacks. First, an attacker attempts an attack by sending malicious commands to many botnets infected with malware

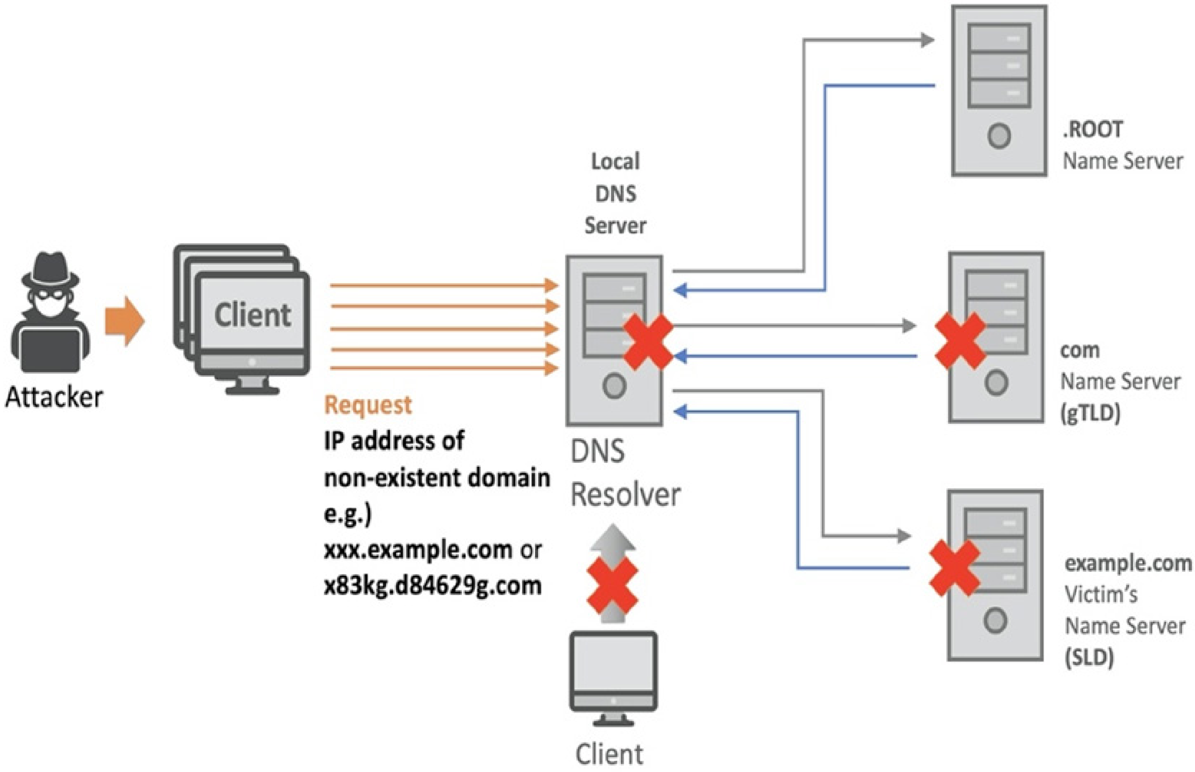


Figure 11. DNS Attack: DNS Server Structure. DNS: Domain Name System

through a C&C server. However, security devices or agencies may block the IP address of the C&C server to prevent communication. Some malware (such as the Necurs Botnet[48]) applies numerous domain names generated by DGA to continuously change the domain of the C&C server. This evades a domain reputation defense to hide the location of the C&C server.

4.3.3 DA09. fast flux

Fast Flux is a method of allocating multiple IP addresses to one domain. By setting the DNS response TTL (Time to Live) to a minimum value (typically within five minutes) and changing the DNS record on the DNS server periodically, the corresponding IP address of the C&C server may be changed repeatedly in a short time interval. This usually relies on a DNS server controlled by the attacker. If a security manager confirms access to a malicious domain and blocks the IP address of that C&C server on the firewall, Fast Flux attempts to bypass this defense[49].

4.4 DNS server structure

As we mentioned in the previous section, DNS has its structural problems. In the hierarchical structure, if a domain on the lowest level does not exist or has a problem, the DNS query processed from the top level may be contaminated. Due to the structural weakness, DNS can easily be attacked, resulting in a large number of victims connected to the DNS server. Figure 11 explains how the DNS attack with the DNS server structure vulnerability works.

4.4.1 DA10. DNS non-existent domain

Non-existent domain (NXDOMAIN) is one of the DNS response queries, which means that a domain does not exist. An attacker sends numerous queries to DNS servers for non-existent domains. The DNS servers try to process the queries to find non-existing domains, but they send back the NXDOMAIN queries because the domains do not exist. Eventually, the cache in the recursive DNS server could be filled with NXDOMAIN results and users will experience slower DNS server response times for legitimate DNS requests. The authoritative DNS servers also spend valuable resources due to the multiple recursive queries to obtain resolution results[50].

4.4.2 DA11. phantom domain

The phantom domain attack is similar to the DNS NXDOMAIN attack. However, the major difference is that attackers use multiple phantom domains to interfere with normal DNS resolution. First, an attacker sets up several phantom domains which either respond very slowly or do not respond to DNS requests. Then, numerous bots send malicious DNS queries for the phantom domains to DNS resolvers. The DNS resolvers handle and deliver the queries to the authoritative servers. However, under the phantom domain attack, the DNS resolvers will continue to wait for responses and continue to query the unresponsive servers, which consumes their resources. As a result, the DNS resolvers' resources are used to process the queries for the phantom domain, and users could be delayed or unable to receive responses to normal DNS queries[51].

4.5 Assessment of DNS attacks

To classify DNS attacks, the types of attacks first are evaluated for each factor. Figure 12 shows the assessment of the 11 DNS attacks introduced in this paper. There are five criteria for evaluating DNS attacks. First is the Attack Method, as described above. The Effect factor classifies attacks according to their intended outcome. The Attack Mode factor refers to whether the attack is passive (i.e., takes place in response to a user-initiated query) or aggressive (launched by the attacker). The Attack Source/Target classifies the multiplicity of attack source(s) and target(s). The Location of Attack Target factor means the location where the attack is executed. If an attacker attempts to attack the DNS infrastructure directly, it is labeled "Internal". Otherwise, if an attacker attempts to attack a target using the DNS infrastructure, it is labeled "External".

The assessment for each factor is a filled circle, meaning fully or completely, half-filled circle, meaning partially, and empty circle, indicating does not apply or not at all. DNS attacks have a variety of purposes. Hijacking/poisoning-based attacks (DNS cache poisoning, Kaminsky, and DNS hijacking) mainly have attack targets to lead to specific malicious sites, while flooding-based attacks (DNS reflection and amplification, DNS flooding, Random sub-domain, DNS NXDOMAIN, and Phantom domain) have the purpose to exhaust DNS servers' resources through direct and aggressive attacks from malware-infected Botnets. van Rijswijk-Deij *et al.*[35] found that DNSSEC could be exploited as DNS reflection attacks. Thus, this attack can target specific servers as well as DNS servers. Finally, attacks that hide their attacks in normal DNS packets or procedures have the purpose of exploiting DNS.

Based on the assessment, Figure 13 shows the classification of DNS attacks by purpose.
(1) DNS Server Unable/Slow: These attacks target DNS servers. The attacker sends a flood of queries to a DNS server, and then the DNS server is forced to exhaust server resources to handle the enormous queries. Eventually, the DNS server will not function normally and not be able to provide the domain service to the user.
(2) Specific Target Server Unable: These attacks target a specific server. The attacker attempts to send heavy traffic to the target server through flooding from the DNS servers. Attackers exploit open DNS resolvers to amplify heavy traffic volume, as a third party[52]. The victim server receives a number of legitimate DNS responses and finally, is subjected to a denial of service attack.
(3) Malicious Website: These attacks provide malicious websites to victims despite requests with normal domains is a DNS Poisoning attack. By manipulating normal response queries, an attacker can illegally acquire and exploit user information by providing bogus IP addresses to the user.

		DNS Cache Poisoning	*Kaminsky*	*DNS Hijacking*	*DNS Reflection and Amplification DoS Attack*	*DNS Flooding Attack*	*Random Sub-domain*	*DNS NXDOMAIN*	*Phantom Domain*	*Domain Generation Algorithms*	*DNS Tunneling*	*Fast Flux*
Attack Method	**Flooding**	○	○	○	●	●	●	●	●	○	○	○
	Poisoning / Hijacking	●	●	●	●	○	○	○	○	○	○	○
	Malware	○	○	○	●	●	●	●	●	●	●	●
	DNS Server Attack	◐	◐	◐	●	●	●	●	●	◐	◐	◐
Effect	**Target Server disable**	○	○	○	●	○	○	○	○	◐	◐	◐
	Move to maricious site	●	●	●	○	○	○	○	○	○	○	○
	aim to hide attack	○	○	○	○	○	○	○	○	●	●	●
	DNS Server disable	○	○	○	●	●	●	●	●	○	○	○
Attack Mode	**passive**	●	●	●	○	○	○	○	○	●	●	●
	aggressive	○	○	○	●	●	●	●	●	○	○	○
Attack Source / Target	**One to One**	○	○	○	○	○	○	●	○	○	○	○
	Many to One	○	○	○	●	●	●	○	●	○	○	○
	One to Many	●	●	●	○	○	○	○	○	●	●	●
Location of Attack Target	**DNS Internal**	●	●	●	○	○	○	○	○	●	●	●
	DNS External	○	○	○	●	●	●	●	●	○	○	○

Figure 12. DNS attacks Assessment. DNS: Domain Name System

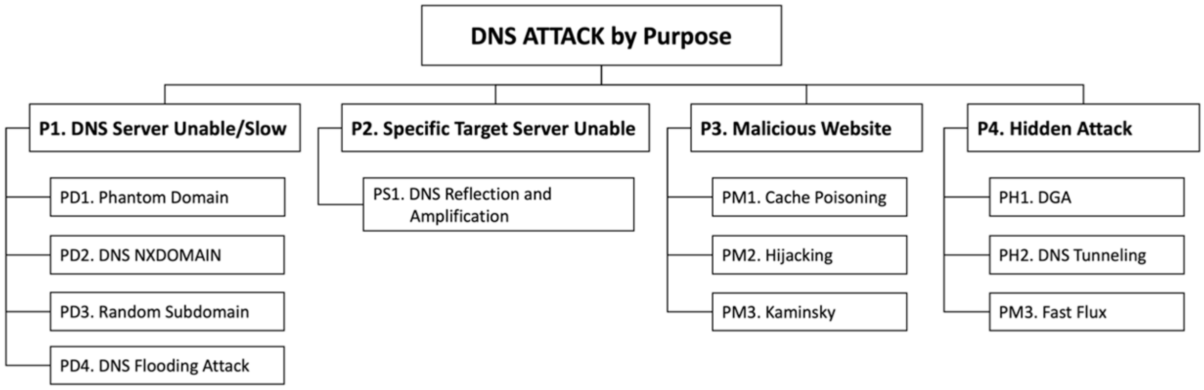


Figure 13. Classification: DNS Attacks by Purpose. DNS: Domain Name System

(4) Hidden Attack: These attacks abuse DNS servers to hide their attack location or attack message. The attacker tries to conceal the location of C&C servers or to exfiltrate the botnet command from C&C, using a vulnerability in internal DNS.

5. MITIGATION

Although DNS has suffered from many attacks, researchers' efforts to mitigate these attacks are ongoing. In particular, DNSSEC, which is the product of their efforts, has helped ensure the integrity of the unreliable DNS data as the main vulnerability of DNS. Additionally, various advanced methods have been introduced to overcome a number of limitations. This section briefly describes them.

5.1 DNSSEC and redundant DNS

Common DNS attacks, such as cache poisoning and spoofing attacks, occur easily by forging DNS data and disguising fake DNS queries. Designed to overcome these problems, DNSSEC uses digital signatures to authenticate the contents of DNS responses, preventing the use of forged DNS data and enhancing the reliability of DNS queries.

As discussed in Section III, DNSSEC suffers from technical complexity, overhead, and low deployment[8]. In 2018, NS1[53] has developed DNSSEC guidelines, so that DNSSEC can be configured correctly and used more easily. However, this does not solve all DNS security issues, including vulnerability to DDoS attacks. The additional length of DNSSEC responses exacerbates the problems of reflection and amplification (DDoS attacks). This dilemma is a major challenge for DSSEC to address in the future.

Redundant DNS servers are one solution to attacks on availability. The DNS standard specified that up to eight spare servers may be used for redundancy[54], so that if a server is unreliable or unavailable, another server can provide name lookup for the user[55]. However, these settings are rarely used in practice by enterprises and ISPs[56], although redundancy has been recommended for a long time.

Ansari *et al.*[57] introduced a new technique to overcome the limitation of DNSSEC and reinforce DNS security, based on using Cloud services for availability and reliability. The redundancy, flexibility, and managed nature of the cloud make it a promising solution for DNS security.

5.2 Existing DNS mitigation systems

A number of approaches for securing DNS have been proposed. We describe these systems by grouping them into three categories: Monitoring and Detection Systems, security extensions on DNS records, and Advanced DNS with additional security functions.

5.2.1 Monitoring and detection systems

DNS is vulnerable to the threat of counterfeited data. One approach is to detect and monitor forged data to distinguish reliable DNS data. The following systems are representative DNS defense systems that include these functions.

(1) Kopis System[58]: Independently detects malware-related domains at the higher levels of the DNS hierarchy (e.g., TLD level) by monitoring network traffic at a high level of the DNS hierarchy. In particular, the Kopis System analyzes the streams of DNS queries and responses at authoritative name servers. From the monitored DNS traffic, they extract the statistical features such as the diversity in the network locations and the reputation of the IP space into which the domain name resolves. Kopis can predict malware-related domains based on monitored traffic patterns with a statistical classification which is determined from higher DNS levels' information. This feature is different from existing detection systems such as Notos[59] (see below) or Exposure[60]. Even without current IP reputation information, Kopis can accurately detect

malware-related domains.
(2) Domain Watcher System[61]: A detection system that detects malicious domain names with local and global textual-based features based on machine learning. This system utilizes three textual features of domains - Lexical features, imitation features, and bi-gram features. First, they use the lexical features to combine the existing characteristic data provided by systems such as EXPOSURE[60] or Detection of Phishing Attacks[62] and new characteristics, such as the number of special characters and numeric characters in the domain name or the number of continuous numeric characters, to easily fetch and normalize the pattern. Imitation features and bi-gram features both utilize the domain information, but imitation looks at the distance between domain names, while bi-gram looks at the similarity of the distribution of letters in domain names.
(3) Anax[63]: A DNS protection system that detects the cache poisoning attack using a large set of open recursive DNS servers (ORDNSs), identifying poisoned DNS caches through DNS records. An infrastructure is added to intercept DNS responses (DNS Scanning Points) and collect and process the resulting data (DNS Data Collector). A Data Preparation Engine analyzes and labels this data, offline, in training mode. A Detection Engine then monitors in real-time DNS responses and flags suspicious responses as poisoning attempts.
(4) Notos-Dynamic Reputation System for DNS[59]: a dynamic reputation system to compute scores of domain names. The goal is to determine if a domain is legitimate or malicious using malicious domains' distinctive features or characteristics.

Other methods of DNS attack detection have been proposed. Zhang *et al.*[64] introduces a new detection method based on machine learning and hybrid methods, which obtains DNS data through active domain name data or passive domain name data. Palau *et al.*[65] proposes an approach to detect DNS tunneling, based on a Convolutional Neural Network (CNN) with a minimal architecture complexity. Also, they use their dataset that contains DNS Tunneling domains generated with five well-known DNS tools. The resulting CNN model correctly detected more than 92% of total Tunneling domains with a false positive rate close to 0.8%. Rajendran *et al.*[66] uses specific properties of DNS amplification and DNS tunneling attacks and presents a number of countermeasures and mitigation techniques to protect against these attacks on the DNS infrastructure.

Fast Flux generates a variety of domain names based on specific algorithms to avoid suppression. Normal DNS-based detection approaches and blacklist filtering are ineffective against the Fast Flux attack. Methods for analyzing new DNS traffic patterns using these Fast Flux characteristics have been developed. These methods recognize the overwhelmingly large or abnormal DNS traffic, filtering the suspicious DNS mapping, and detecting domains of pseudorandom strings generated by the algorithm compared with legitimate domain patterns[67-69]. In particular, DNSMap[67] can quickly identify excessive DNS traffic in real-time by analyzing the DNS mapping of abnormal domains and IP addresses through graphical analysis, unlike conventional methods of domain analysis based on machine learning.

5.2.2 Security extension of DNS records

DNS records provide information about domains that are needed by users. More information may be added to provide data integrity and improve/extend trust. Several methods attempt to do so with less overhead than DNSSEC.

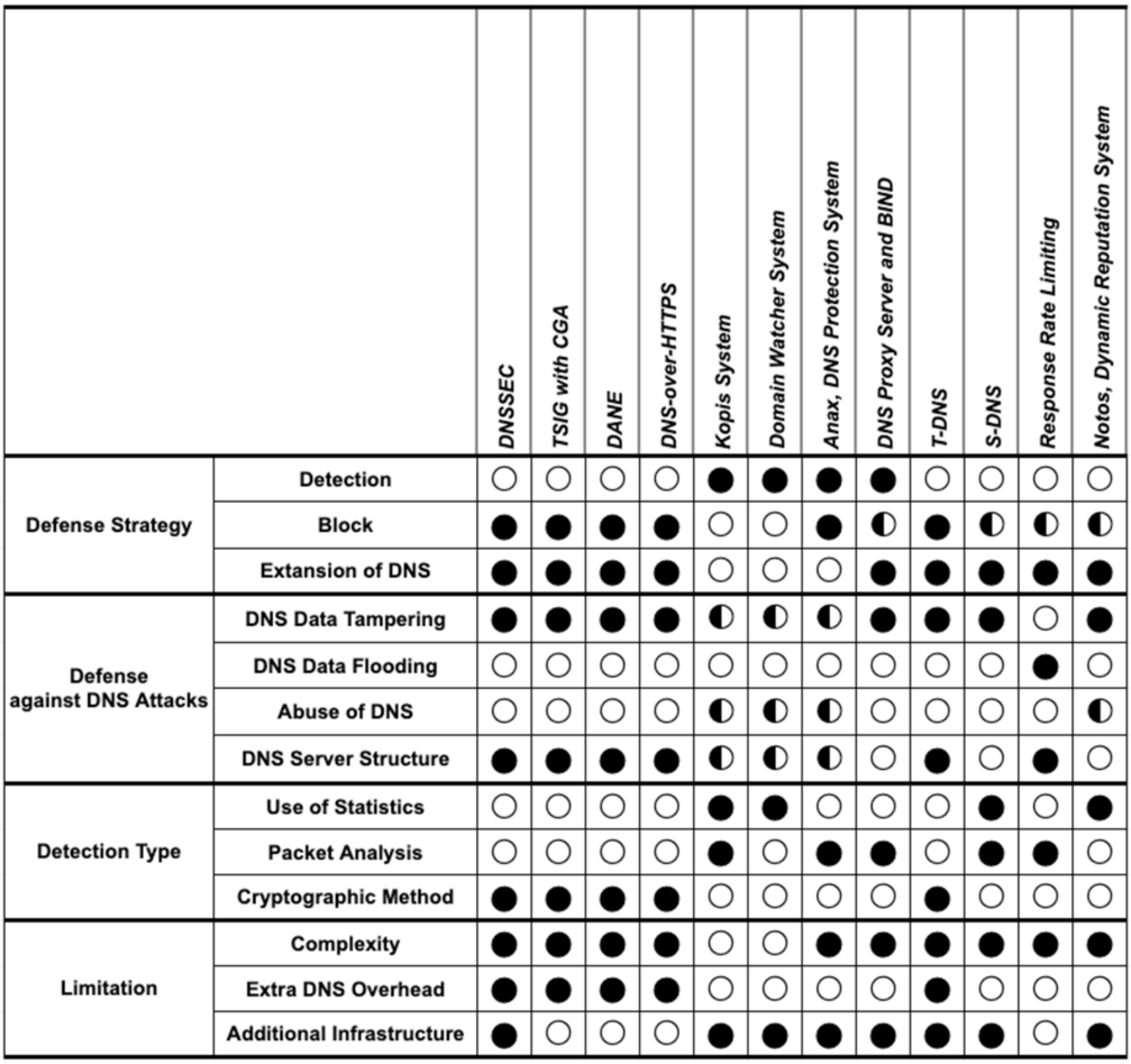
(1) The Transaction SIGnature (TSIG) using CGA (Cryptographically Generated Addresses) Algorithm in IPv6[70]: DNS has a security problem between the client and the DNS resolver due to the untrustworthy resolver as discussed in the 'Vulnerabilities' section. To address this issue, TSIG is used. TSIG establishes a trust relationship between a client and a DNS server. This process provides not only end-to-end authentication but also data integrity between each other through a one-way hash algorithm and shared

keys. However, TSIG faces one problem that it requires the keys is exchanged manually. A solution to the key distribution problem is TSIG using CGA. TSIG-CGA provides an automated way for the negotiation of a shared secret key, with authentication of the host via IPv6's CGA algorithm.
(2) DNS-Based Authentication of Named Entities (DANE)[71-73]: DANE takes advantage of the source of trust provided by DNSSEC to authenticate transport layer security (TLS) certificates. Through TLSA records in the DNS hierarchy, DNSSEC can verify the integrity of DNS data. DANE was designed to provide a stronger trust anchor using DNS as the root. Especially, DANE uses the DNSSEC chain of trust to authenticate X.509 certificates used for transport layer security (TLS) and, as it relies on DNSSEC infrastructure, it can support authentication and data integrity. DANE allows domain owners to issue their certificates without CAs. Using the DNS hierarchy as a single trust anchor instead of many existing CAs, DANE greatly reduces the attack surface. DANE can be used to solve issues related to CAs' vulnerability through the use of a new DNS resource record type, TLSA, signed with DNSSEC. As a result, DANE allows TLS users to better control certificate validation.
(3) DNS-over-HTTPS (DoH)[74]: DoH is a standard web protocol to send DNS traffic over HTTPS. DoH is developed to prevent fundamental DNS privacy problem of unencrypted communication between users and DNS resolvers. As shown in the previous section, without a trusted DNS resolver, DNS queries cannot be guaranteed. In DoH, by using HTTPS's security platform, DNS queries and responses are protected. Moreover, DNS traffic and requests are not directly observable because DoH applies the same port 443 used by HTTPS traffic. Additionally, DoH can be provided by existing DNS servers using a built-in web server. Starting with Mozilla Firefox and Google Chrome in 2018, most major web browsers support or plan to support DoH. Despite this, there are some drawbacks to DoH. First, DNS traffic is encrypted, making it difficult to track/analyze. Mitigation systems that detect DNS attacks based on DNS data analysis will fail to function. Second, the prerequisite for DoH is the support of a trusted DNS resolver. Each web browser, such as Firefox-Cloudflare and Chrome-Google OpenDNS, provides a trusted open DNS resolver. However, traffic is centralized with a few DNS resolvers, with corresponding privacy and performance concerns. Finally, the policies of these enterprises will be difficult to ensure transparency in DNS operations.

5.2.3 Advanced DNS with additional secure functions

According to the DNSSEC deployment tracking system SecSpider[75], current DNSSEC-enabled zones number approximately 3.3 million. It seems that the full deployment of DNSSEC will take considerable time despite many efforts. Thus, additional security functions for DNS are required. The following are methods for improving DNS security.

(1) DNS Proxy Server (DPS) and BIND[76]: a new approach to detect cache poisoning attacks and then send an additional request for the same DNS Resource Record using a local proxy for the BIND caching server. This defensive system makes cache poisoning attacks more difficult.
(2) T-DNS[77]: DNS uses unconnected UDP as the standard protocol. However, because of the poorly secured UDP protocol, DNS is subject to attacks such as spoofing and flooding. T-DNS uses TCP and TLS to provide DNS security. T-DNS provides more secure DNS data through TCP encryption, reduces the impact of DoS attacks by establishing mutual connections, and overcomes the limitations of UDP's response size. DNS based on TLS can provide more secure privacy, support large payload, and mitigate spoofing and reflection DDoS attacks compared to the use of existing UDP protocols. However, the fundamental problems of TCP, latency, and resource needs, remain.
(3) S-DNS[78]: A security solution to prevent DNS cache poisoning and spoofing attacks. Based on the predictability measures and timing analysis, S-DNS mitigates man-in-the-middle attacks in the DNS hierarchy. This protocol has effects on decreasing the probability of the attack and also provides a simple security mechanism with light-weight computation and overheads.
(4) Response Rate Limiting[43]: A defense mechanism to reduce the impact of DNS amplification attacks



		DNSSEC	TSIG with CGA	DANE	DNS-over-HTTPS	Kopis System	Domain Watcher System	Anax, DNS Protection System	DNS Proxy Server and BIND	T-DNS	S-DNS	Response Rate Limiting	Notos, Dynamic Reputation System
Defense Strategy	Detection	○	○	○	○	●	●	●	●	○	○	○	○
	Block	●	●	●	●	○	○	●	◐	●	◐	◐	◐
	Extansion of DNS	●	●	●	●	○	○	○	●	●	●	●	●
Defense against DNS Attacks	DNS Data Tampering	●	●	●	●	◐	◐	◐	●	●	●	○	●
	DNS Data Flooding	○	○	○	○	○	○	○	○	○	○	●	○
	Abuse of DNS	○	○	○	○	◐	◐	◐	○	○	○	○	◐
	DNS Server Structure	●	●	●	●	◐	◐	◐	○	●	○	●	○
Detection Type	Use of Statistics	○	○	○	○	●	●	○	○	○	●	○	●
	Packet Analysis	○	○	○	○	●	○	●	●	○	●	●	○
	Cryptographic Method	●	●	●	●	○	○	○	○	●	○	○	○
Limitation	Complexity	●	●	●	●	○	○	●	●	●	●	●	●
	Extra DNS Overhead	●	●	●	●	○	○	○	○	●	○	○	○
	Additional Infrastructure	●	○	○	○	●	●	●	●	●	●	○	●

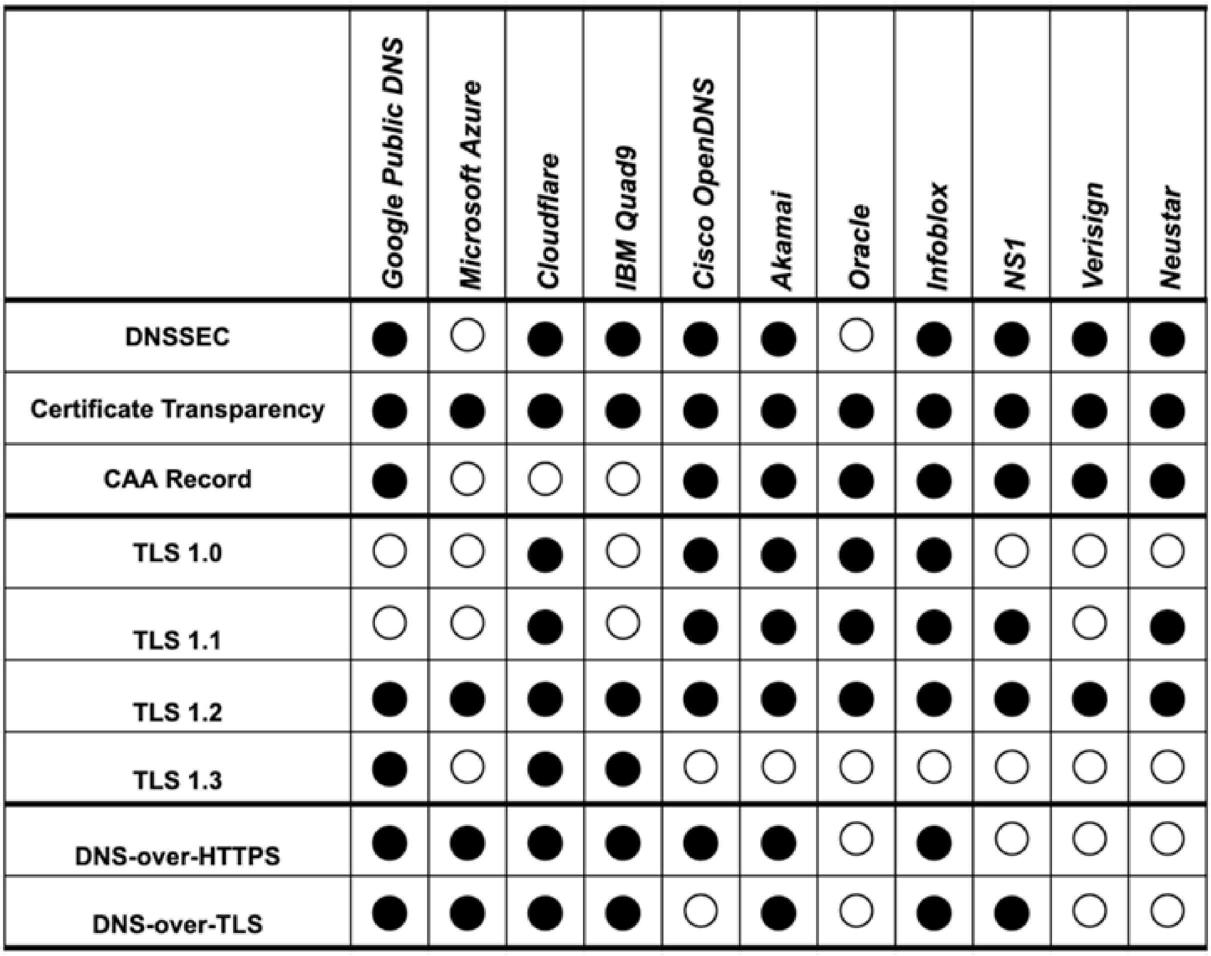
Figure 14. Assessment of DNS Mitigation. DNS: Domain Name System

and reflection attacks. The DNS server will respond a limited number of times to requests for a domain name resolution from a particular IP address, making it more difficult to flood the victim with traffic.

5.3 Overall assessment of DNS mitigation system

Figure 14 shows the assessment of whether the mitigation system can protect against DNS attacks.

A full circle denotes yes or fully, a half-circle denotes partially, and empty circles denote no or not at all. Each mitigation system was developed to solve specific vulnerabilities in DNS. Several key findings of our assessment are provided:
(1) DNSSEC is a major enhancement to DNS but can be exploited for DDoS attacks. According to the 2019 report released by Neustar[79], the number of DDoS attacks increased by 133% and the average DDoS attack size is 7.5 Gbps compared to 2018.
(2) Most monitoring and detection systems can observe the malicious DNS traffic, not protect against the attacks. But, using these mitigation systems, it is possible to filter or protect against the DNS data attacks.



	Google Public DNS	Microsoft Azure	Cloudflare	IBM Quad9	Cisco OpenDNS	Akamai	Oracle	Infoblox	NS1	Verisign	Neustar
DNSSEC	●	○	●	●	●	●	○	●	●	●	●
Certificate Transparency	●	●	●	●	●	●	●	●	●	●	●
CAA Record	●	○	○	○	●	●	●	●	●	●	●
TLS 1.0	○	○	●	○	●	●	●	●	○	○	○
TLS 1.1	○	○	●	○	●	●	●	●	●	○	●
TLS 1.2	●	●	●	●	●	●	●	●	●	●	●
TLS 1.3	●	○	●	●	○	○	○	○	○	○	○
DNS-over-HTTPS	●	●	●	●	●	●	○	●	○	○	○
DNS-over-TLS	●	●	●	●	○	●	○	●	●	○	○

Figure 15. List of the 10 Enterprise DNS providers. DNS: Domain Name System; TLS: transport layer security

(3) TSIG with CGA and DANE are solutions to overcome DNSSEC's limitations and are promising alternatives.
(4) Because most advanced DNS mitigation systems with additional security functions are focused on specific security problems in DNS, they do not cover all DNS attacks. On the other hand, T-DNS prevents most of the DNS attacks because they address the fundamental protocol problem in the DNS protocol. However, T-DNS, based on the TCP protocol, greatly helps improve DNS privacy, while its latency is the slower, and overall cost is significant compared to the UDP protocol.

5.4 Secure/enterprise DNS provider

Unlike these mitigation systems which provide additional security functions or monitor/analyze/detection techniques, an openDNS of major companies or organizations that ensure improved security, reliability and speed would be better option to defend against some of the DNS attacks. It is called Secure/Enterprise DNS, which is a fast and reliable DNS service from large organizations. Enterprise DNS centrally manages its security architecture that guarantees a more sophisticated and reliable DNS service.

To better understand the current Enterprise DNS situation, we provide and evaluate a list of 10 large Enterprise DNS providers, as shown in Figure 15. Each organization provides its open DNS and can be set up and used by anyone on their device. Except for Microsoft Azure and Oracle, most providers support DNSSEC. Azure and Oracle protect DNS through their systems.

Another factor is the support of the Certification Transparency and Certification Authority (CAA) records, which are techniques to compensate for weaknesses and defects in the PKI-certificate system. While all organizations provide Certification Transparency, some do not offer CAA records. Regardless of whether DoH or DoT is supported or not, it is judged as the support of a security solution for certificates.

Almost all providers support DoH and/or DoT, except for Oracle and Verisign. We expect that the support of the DoH/DoT would increase with time.

Finally, all providers offer TLS 1.2 for cipher transmission, especially Google, Cloudfare, and Quad9 that support DoH, up to the latest TLS 1.3. Therefore, these institutions are expected to provide more stable DoH based on TLS 1.3 in the future.

6. DISCUSSION

This paper presents a survey of DNS security. The background of basic DNS and DNSSEC was described, with an explanation for the motivation of DNSSEC. DNS is essential for proper operation of the Internet, but it is still subject to a variety of attacks, due to its vulnerabilities, lack of widespread adoption of available mitigation techniques, and limitations of those techniques. These vulnerabilities were described, and DNS attacks were classified based on those vulnerabilities. Also, several methods suggested in the literature for defending against such attacks were summarized.

This survey provides a novel and useful analysis to understand DNS and DNSSEC in terms of cybersecurity. Specifically, the classification of DNS attacks supports understanding and analysis of future DNS attacks. This paper provides the first DNS attack classification. The analysis of various mitigation systems also provides indicators for future DNS developments. Promising alternatives to DNSSEC include DANE/TLSA and DNS-over-HTTPS. Even lighter-weight approaches, suitable for deployment in the Internet of Things, are needed as well.

DECLARATIONS

Authors' contributions

Contributed to the design, survey, implementation, and analysis of the research and to the writing of the manuscript: Kim TH, Reeves D

Availability of data and materials

Not applicable.

Financial support and sponsorship

Not applicable.

Conflicts of interest

Both authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Mockapetris P. Domain names-implementation and specification. RFC1035 1987. Available from: https://tools.ietf.org/html/rfc1035. [Last accessed on 17 Aug 2020]
2. Mockapetris P. Domain names-implementation and facilities. RFC1034 1987.
3. Engel S. My ether wallet DNS attack explained. Available from: https://cryptovoid.net/mew-dns-attack-explained. [Last accessed on 17 Aug 2020]
4. Arends R, Austein R, Larson M, Massey D, Rose S. DNS security introduction and requirements. RFC 4033 2005:1-21. Available from: https://tools.ietf.org/html/rfc4033. [Last accessed on 17 Aug 2020]
5. Arends R, Austein R, Larson M, Massey D, Rose S. Resource records for the DNS security extensions. RFC 4034 2005:1-30. Available from: https://tools.ietf.org/html/rfc4034. [Last accessed on 17 Aug 2020]
6. Arends R, Austein R, Larson M, Massey D, Rose S. Protocol modifications for the DNS security extensions. RFC 4035 2005:1-54. Available from: https://tools.ietf.org/html/rfc4035. [Last accessed on 17 Aug 2020]
7. Eastlake 3rd D. Domain name system security extensions. RFC 2535 1999. Available from: https://tools.ietf.org/html/rfc2535. [Last accessed on 17 Aug 2020]
8. Chung T, an Rijswijk-Deij R, Chandrasekaran B, Choffnes D, Levin D, et al. A longitudinal, end-to-end view of the DNSSEC ecosystem. Proceedings of the 26th USENIX Conference on Security Symposium; 2017 Aug; Vancouver, BC. USENIX Association, USA; 2017. pp. 1307-22.
9. NIST. Estimating USG IPv6 and DNSSEC external service deployment status. Available from: https://fedv6-deployment.antd.nist.gov/cgi-bin/generate-gov. [Last accessed on 17 Aug 2020]
10. Roosa SB, Schultze S. Trust darknet: control and compromise in the internet's certificate authority model. IEEE Internet Comput 2013;17:8-25.
11. Wikipedia. 2016 Dyn cyberattack. Avaliable from: https://en.wikipedia.org/wiki/2016_Dyn_cyberattack. [Last accessed on 17 Aug 2020]
12. Downdetector. Internet outage map. Avaliable form: https://downdetector.com/status/centurylink/map/. [Lasted accessed on 27 Jul 2020]
13. NETSCOUT. NETSCOUT's 14th Annual Worldwide Infrastructure Security Report. Avaliable from: https://www.netscout.com/report/. [Last accessed on 17 Aug 2020]
14. Zhauniarovich Y, Khalil I, Yu T, Dacier M. A survey on malicious domains detection through DNS data analysis. ACM Computing Surveys (CSUR) 2018;51:1-36.
15. Fernandes D, Soares LFB, Gomes JV, Freire M, Inácio PRM. Security issues in cloud environments: a survey. Int J Inf Secur 2014;13:113-70.
16. Alieyan K, ALmomani A, Manasrah A, Kadhum, MM. A survey of botnet detection based on DNS. Neural Computing and Applications 2017;28:1541-58.
17. Peng T, Leckie C, Ramamohanarao K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Comput Surv 2007;39:31-42.
18. Casalicchio E, Caselli M, Coletta A. Measuring the global domain name system. IEEE network 2013;27:25-31.
19. Wikipedia. List of DNS resource records. Available from: https://en.wikipedia.org/wiki/List-of-DNS-record-types. [Last accessed on 17 Aug 2020]
20. Cheshire S, Krochmal M. Multicast DNS, RFC 6762 2013. Available from: https://tools.ietf.org/html/rfc6762. [Last accessed on 17 Aug 2020]
21. Aboba B, Thaler D, Esibov L. Link-local multicast name resolution (LLMNR), RFC 4795, January 2007. Available from: https://www.rfc-editor.org/info/rfc4795. [Last accessed on 17 Aug 2020]
22. Andress J. The basics of information security: understanding the fundamentals of InfoSec in theory and practice, 2nd ed. Syngress; 2014. p. 240.
23. Bates S, Bowers J, Greenstein S, Weinstock J, Xu Y, et al. Evidence of decreasing internet entropy: the lack of redundancy in DNS resolution by major websites and services. Available from: https://www.nber.org/papers/w24317. [Last accessed on 17 Aug 2020]
24. Schiffman M. Bound by tradition: a sampling of the security posture of the internet's DNS servers. LinuxSecurity 2003. Available from: http://packetfactory.openwall.net/papers/DNS-posture/DNS-posture-1.0.pdf. [Last accessed on 17 Aug 2020]
25. Migault D, Cédric G, Laurent M. A performance view on dnssec migration. 2010 International Conference on Network and Service Management (CNSM); 2010 Oct 25-29; Niagara Falls, Canada. IEEE; 2010. pp. 469-74.
26. Klein A. BIND 9 DNS cache poisoning. SecuriTeam 2007. Available from: https://securiteam.com/securitynews/5vp0l0um0a/. [Lasted accessed on 28 Jul 2020]
27. Yu X, Chen X, Xu F. Recovering and protecting against DNS cache poisoning attacks.2011 International Conference onInformation Technology, Computer Engineering and Management Sciences (ICM); 2011 Dec 26-28; Beijing, China. IEEE; 2011. pp. 120-3.
28. Ager B, Dreger H, Feldmann A. Predicting the DNSSEC overhead using DNS traces. In 2006 40th Annual Conference on Information Sciences and Systems. 2006, March 22-24, Princeton, NJ, USA. IEEE; 2006.
29. Van Adrichem NLM, Blenn N, Lua AR, Wang X, Wasif M, et al. A measurement study of DNSSEC misconfigurations. Secur Inform 2015;4:1-14.
30. Deccio C, Sedayao J, Kant K, Mohapatra P. Quantifying and improving dnssec availability. 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN); 2011 Jul 31-Aug 4; Lahaina, HI, USA. IEEE; 2011. pp. 1-7.
31. Clark L. A cartoon intro to DNS over HTTPS. Avaliable from: https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/. [Last accessed on 17 Aug 2020]

32. Droms R, Arbaugh W. Authentication for DHCP messages. RFC 3118. Avaliable from: https://tools.ietf.org/html/rfc3118. [Last accessed on 17 Aug 2020]
33. Bau J, Mitchell JC. A security evaluation of DNSSEC with NSEC3. Proceedings of the Network and Distributed System Security Symposium, 2010 Feb 28-Mar 3; San Diego, California, USA. NDSS; 2010. pp. 18.
34. Internet society. State of DNSSEC deployment 2016. Avaliable from: https://www.internetsociety.org/resources/doc/2016/state-of-dnssec-deployment-2016. [Last accessed on 17 Aug 2020]
35. van Rijswijk-Deij R, Sperotto A, Pras A. DNSSEC and its potential for DDoS attacks: a comprehensive measurement study.Proceedings of the 2014 Conference on Internet Measurement Conference; 2014 Nov; Vancouver, BC, Canada. ACM; 2014. pp. 449-60.
36. Loveless J. DNSSEC: how Savvy DDoS attackers are using our defenses against us, Security Research Report by Neustar 2016. Avaliable from: http://www.circleid.com/posts/20160818_how_savvy_ddos_attackers_are_using_dnssec_against_us/. [Last accessed on 17 Aug 2020]
37. Alharbi F, Chang J, Zhou YC, Qian F, Qian ZY, et al. Collaborative client-side DNS cache poisoning attack. IEEE INFOCOM 2019-IEEE Conference on Computer Communications. 2019. Apr 29 - May 2; Paris, France. IEEE, 2019.
38. Kaminsky D. Black ops 2008: It's the end of the cache as we know it. Black Hat USA 2008; 2. Avaliable from: https://www.blackhat.com/presentations/bh-jp-08/bh-jp-08-Kaminsky/BlackHat-Japan-08-Kaminsky-DNS08-BlackOps.pdf. [Last accessed on 17 Aug 2020]
39. Vissers T, Barron T, van Goethem T, Joosen W, Nikiforakis N. The wolf of name street: hijacking domains through their nameservers. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security; 2017 Oct; Dallas, Texas, USA. ACM; 2017. pp. 957-70.
40. Rascagneres P. Mercer W. DNSpionage campaign targets middle east. Available from: https://blogs.cisco.com/security/talos/dnspionage-campaign-targets-middle-east. [Last accessed on 17 Aug 2020]
41. Thornewell PM, Golden LM. DNS flood protection platform for a network. US Patent 2012;8,261,351. Available from: https://portal.unifiedpatents.com/patents/patent/US-8261351-B1. [Last accessed on 17 Aug 2020]
42. Rozekrans T, Mekking M, de Koning J. Defending against DNS reflection amplification attacks. University of Amsterdam System & Network Engineering RP1 2013. Available from: https://www.nlnetlabs.nl/downloads/publications/report-rrl-dekoning-rozekrans.pdf. [Last accessed on 17 Aug 2020]
43. Chandramouli R, Rose S. Secure domain name system (DNS) deployment guide. NIST Special Publication 2006;800:81-2.
44. Feibish SL, Afek Y, Bremler-Barr A, Cohen E, Shagam M. Mitigating DNS random subdomain DDoS attacks by distinct heavy hitters sketches. Proceedings of the fifth ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies; 2017 Oct; San Jose, California. New York, NY, USA: Association for computing Machinery; 2017. pp. 1-6.
45. Farnham G, Atlasis A. Detecting DNS tunneling. SANS Institute InfoSec Reading Room 2013;9:1-32.
46. van Leijenhorst T, Chin KW, Lowe D. On the viability and performance of DNS tunneling. The 5th International Conference on Information Technology and Applications (ICITA); 2008. pp. 560-6.
47. Zhou Y, Li Q, Miao Q,Yim K. DGA-based botnet detection using DNS traffic. JInternet ServInfSecur 2013;3:116-23.
48. Kessem L. The Necurs Botnet: a pandora's box of malicious spam. Security Intelligence. Acaliable from: https://securityintelligence.com/the-necurs-botnet-a-pandoras-box-of-malicious-spam/. [Last accessed on 17 Aug 2020]
49. Metcalf LB, Ruef, Spring JM. Open-source measurement of fast-flux networks while considering domain-name parking. The LASER Workshop: Learning from Authoritative Security Experiment Results (LASER 2017); 2017 Oct 18-19; USENIX Association; 2017. pp. 13-24.
50. Dagon D, Lee C, Lee W, Provos N. Corrupted DNS resolution paths: The rise of a malicious resolution authority. Proceedings of the 15th Network and Distributed System Security Symposium (NDSS); 2008 Feb 10-13; San Diego, California, USA. NDSS; 2008.
51. Mergenhagen P, Domain DP. Mainstreethost. Available from: https://www.mainstreethost.com/blog/deindexing-phantom-domains. [Last accessed on 10 Aug 2020]
52. Krämer L, Krupp J, Makita D, Nishizoe T, Koide T, et al. Amppot: monitoring and defending against amplification ddos attacks. International Symposium on Recent Advances in Intrusion Detection; 2015 Nov 2-4; Kyoto, Japan. Springer; 2015. pp. 615-36.
53. NS1. Enabling DNSSEC. Available from: https://ns1.com/knowledgebase/dnssec. [Last accessed on 27 Jul 2020]
54. Elz R, Bush R, Bradner S, Patton M. Selection and Operation of Secondary DNS Servers. RFC 2182 1997. Available from: https://tools.ietf.org/html/rfc2182. [Last accessed on 27 Jul 2020]
55. Yu Y, Cai J, Osterweil E, Zhang L. Measuring the placement of DNS servers in top-level-domain. Verisign Technical Report 2011. Available from: https://www.semanticscholar.org/paper/Measuring-the-Placement-of-DNS-Servers-in-Yu/4afb5d97b5002edc7f14708a51d7abb322d28f9a. [Last accessed on 27 Jul 2020]
56. Bisiaux JY. DNS threats and mitigation strategies. Network Security 2014;7:5-9.
57. Ansari A, Khan N, Rais Z, Taware P. Reinforcing security of DNS using AWS cloud. Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST); 2020 Apr 8-9; Mumbai, India. SSRN; 2020.
58. Antonakakis M, Perdisci R, Lee W, Vasiloglou N, Dagon D. Detecting malware domains at the upper DNS hierarchy. Proceedings of the 20th USENIX Conference on Security; 2011 Aug; USENIX Association. USA; 2011. pp. 1-16.
59. Antonakakis M, Perdisci R, Dagon D, Lee W, Feamster N. Building a dynamic reputation system for DNS. Proceedings of the 19th USENIX Conference on Security; 2010 Aug; USENIX Association. USA; 2010. pp. 273-89.
60. Bilge L, Kirda E, Kruegel C, Balduzzi M. EXPOSURE: finding malicious domains using passive DNS analysis. Proceedings of the Network and Distributed System Security Symposium, 2011 Feb 6-9; San Diego, California, USA. NDSS; 2011.
61. Zhang P, Liu T, Zhang Y, Ya J, Shi J, et al. Domain watcher: detecting malicious domains based on local and global textual features. ProcComputSci 2017;108:2408-12.

62. Muhammet B, Ziya GZ. Detection of phishing attacks. 2018 6th International Symposium on Digital Forensic and Security (ISDFS); 2018 Mar 22-25; Antalya, Turkey. IEEE; 2018. pp. 1-5.
63. Antonakakis M, Dagon D, Luo X, Perdisci R, Lee W, et al. A centralized monitoring infrastructure for improving dns security. Proceedings of the 13th International Conference on Recent Advances in Intrusion Detection; 2010 Sep; International Symposium, Raid, Ottawa, Ontario, Canada. Berlin: Springer-Verlag; 2010. pp. 18-37.
64. Zhang K, Ji W, Li N, Wang Y, Liao S. Detection of malicious domain name based on DNS data analysis. JPhysConfSer 2020;1544:012169.
65. Palau F, Catania C, Guerra J, Garcia S, Rigaki M. DNS tunneling: a deep learning based lexicographical detection approach. Cryptography and Security 2020.
66. Rajendran, B. DNS amplification & DNS tunneling attacks simulation, detection and mitigation approaches. 2020 International Conference on Inventive Computation Technologies (ICICT); 2020 Feb 26-27; Coimbatore, India. IEEE; 2020. pp. 230-6.
67. Berger A, D'Alconzo A, Gansterer WN, Pescape A. Mining agile dns traffic using graph analysis for cybercrime detection. Comput Netw 2016;100:28-44.
68. Perdisci R, Corona I, Giacinto G. Early detection of malicious flux networks via large-scale passive DNS traffic analysis. IEEE T Depend Secure 2012;9:714-26.
69. Yadav S, Reddy AKK, Reddy AN, Ranjan S. Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. IEEEACM TNetwork 2012;20:1663-77.
70. Vixie P, Gudmundsson O, Eastlake D, Wellington B. Secret key transaction authentication for DNS (TSIG). RFC28452000. Available from: https://www.bibsonomy.org/bibtex/fbdc74e947549d1d0939d567bd377f08. [Last accessed on 27 Jul 2020]
71. Barnes R. Use cases and requirements for DNS-based authentication of named entities (DANE). RFC 6394 2011. Available from: https://tools.ietf.org/html/rfc6394. [Last accessed on 27 Jul 2020]
72. Gudmundsson O. Adding acronyms to simplify conversations about DNS-based authentication of named entities (DANE). RFC 7218 2014. Available from: https://tools.ietf.org/html/rfc7218. [Last accessed on 27 Jul 2020]
73. Zhu L, Wessels D, Mankin A, Heidemann J. Measuring dane tlsa deployment. International Workshop on Traffic Monitoring and Analysis; 2015 Apr 21-24; Barcelona, Spain. Springer; 2015. pp. 219-32.
74. Hoffman P, McManus P. DNS queries over HTTPS (DoH). RFC 8484 2018. Available from: https://tools.ietf.org/html/rfc8484. [Last accessed on 27 Jul 2020]
75. SecSpider. Global DNSSEC deployment tracking. Available from: http://secspider.net/. [Last accessed on 17 Aug 2020]
76. Trostle J, van Besien B, Pujari A. Protecting against DNS cache poisoning attacks. 2010 6th IEEE Workshop on Secure Network Protocols; 2010 Oct 5-5; Kyoto Japan. IEEE; 2010. pp. 25-30.
77. Zhu L, Hu Z, Heidemann J, Wessels D, Mankin A, et al. T-DNS: connection-oriented DNS to improve privacy and security. ACM SIGCOMM CompCom 2014;44:379-80.
78. Bassil R, Hobeica R, Itani W, Ghali C, Kayssi A, et al. Security analysis and solution for thwarting cache poisoning attacks in the domain name system. 2012 19th International Conference on Telecommunications (ICT); 2012 Apr 23-25; Jounieh, Lebanon. IEEE; 2012. pp. 1-6.
79. Neustar. Q2, 2019 Cyber threats and trends report. Available from: https://www.home.neustar/resources/whitepapers/2019-cyberthreats-trends-report. [Lasted accessed on 17 Aug 2020]