NIST Special Publication 800
NIST SP 800-81r3

Secure Domain Name System (DNS) Deployment Guide

Scott Rose
Cricket Liu
Ross Gibson

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-81r3

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication 800
NIST SP 800-81r3

Secure Domain Name System (DNS) Deployment Guide

Scott Rose
Wireless Networks Division
Communications Technology Lab

Cricket Liu
Ross Gibson
Infoblox Inc.

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-81r3

March 2026

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, NIST Director and Under Secretary of Commerce for Standards and Technology (Acting)

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications

NIST Technical Series Policies
Copyright, Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History
Approved by the NIST Editorial Review Board on 2026-03-04
Supersedes NIST SP 800-81-2 (Sept. 2013) https://doi.org/10.6028/NIST.SP.800-81-2

How to Cite this NIST Technical Series Publication
Rose S, Liu C, Nodurft R (2026) Secure Domain Name System (DNS) Deployment Guide. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-81r3. https://doi.org/10.6028/NIST.SP.800-81r3

Author ORCID iDs
Scott Rose: 0000-0002-3105-7427
Cricket Liu: 0009-0004-3351-4734
Ross Gibson: 0009-0008-4433-1705

Contact Information
SP800-81@nist.gov

National Institute of Standards and Technology
Attn: Wireless Networks Division, Communication Technology Laboratory
100 Bureau Drive (Mail Stop 6730) Gaithersburg, MD 20899-6730

Additional Information
Additional information about this publication is available at https://csrc.nist.gov/pubs/sp/800/81/r3/final, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

The Domain Name System (DNS) is an integral part of any enterprise network architecture. An attack against the DNS infrastructure of an enterprise threatens every network operation in that enterprise. DNS operations are composed of different roles that each have their own set of security considerations. This document provides DNS deployment guidelines to secure the DNS protocol and infrastructure, mitigate misuse or misconfiguration, and provide an additional layer of network security as part of a zero trust and/or defense-in-depth security risk management approach.

Keywords

Authoritative name server; Domain Name System (DNS); DNS logging; DNS Security Extensions (DNSSEC); encrypted DNS; protective DNS; recursive name server; Resource Record (RR).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Patent Disclosure Notice

NOTICE: ITL has requested that holders of patent claims whose use may be required for compliance with the guidance or requirements of this publication disclose such patent claims to ITL. However, holders of patents are not obligated to respond to ITL calls for patents and ITL has not undertaken a patent search in order to identify which, if any, patents may apply to this publication.

As of the date of publication and following call(s) for the identification of patent claims whose use may be required for compliance with the guidance or requirements of this publication, no such patent claims have been identified to ITL.

No representation is made or implied by ITL that licenses are not required to avoid patent infringement in the use of this publication.

Table of Contents

List of Tables

List of Figures

Preface

This revision of Special Publication (SP) 800-81 includes significant changes to the structure, format, and general objective of the document. Since the previous version was published, the ways in which DNS is used and deployed have changed significantly, and this revision provides an updated set of discussions and recommendations for securing modern DNS deployments.

Acknowledgments

The authors would like to thank Stephen Banghart for organizing the effort to produce this document. The authors would also like to thank Frank Hecker, Philip Parker, Jim Mozley, Ken Shade, and Paul Adair for their input and discussion of the contents of this document.

Executive Summary

The Domain Name System (DNS) [1][2] is a standardized way of mapping human-readable domain names (e.g., example.com) to machine-readable information, such as IP addresses (e.g., 192.0.2.1). It is commonly deployed within an organization's networks to facilitate the internal functions of those intranets. DNS is also deployed and maintained across critical internet infrastructure at a high level to enable the core functionalities of the internet on almost every network at every scale. Its centralized position enables it to act as a foundational layer of network security in zero trust and defense-in-depth security risk management approaches. Such DNS services are often referred to as *protective DNS*[1] deployments and are a key consideration in securing organizational networks. DNS infrastructure has become more than a service; it is also a security control that can be an important part of an enterprise security architecture.

This revision of Special Publication (SP) 800-81 acknowledges these changes in the role of DNS and provides modern guidelines on DNS deployments with the following high-level recommendations for network and security owners:

- Employ protective DNS wherever technically feasible to provide additional network-wide security capabilities that include:
    - Blocking harmful or malicious traffic in real time
    - Filtering out categories of traffic that do not conform to the organization's policies
    - Generating real-time and historical DNS query and response data to facilitate digital forensics and incident response
    - Integrating with the wider security ecosystem as part of a defense-in-depth or zero trust approach
    - Facilitating the organization's responsibility to comply with regulatory or contractual requirements for blocking traffic to disallowed sites (e.g., copyright violations, legal restrictions)
- Encrypt internal and external DNS traffic wherever feasible
- Deploy DNS Security Extensions (DNSSEC) to protect the integrity of DNS data
- Deploy dedicated DNS servers to reduce attack surfaces
- Follow all technical guidance on ensuring that DNS deployments and the DNS protocol are as secure and resilient as possible

[1] This refers to the general service and not the CISA program of the same name.

1. Introduction

This document provides Domain Name System (DNS) deployment guidelines to secure the DNS protocol and servers, mitigate the impacts of misuse and compromise, and utilize DNS as a foundational layer of security control across the organization. This introduction briefly discusses relevant contexts for DNS and examines the changing threat landscape that has warranted this updated approach to DNS deployment best practices.

1.1. Domain Name Systems

From the perspective of a user, each node or resource on the internet is identified by a unique domain name. From the perspective of networking components, however, the unique resource identifier is an Internet Protocol (i.e., IPv4 or IPv6) address. Accessing internet resources by domain names rather than IP addresses requires a system for translation, which is the primary task of the DNS. The process of translating domain names into IP addresses is called *name resolution*. For an in-depth overview of how DNS works and how its constituent components are arranged, see Appendix A.

When DNS was first developed and deployed, it focused on providing name resolution for resources on the internet. As more sophisticated internet and intranet infrastructure have developed, the scope of DNS has expanded to cover domain name translations inside of private networks. Today, DNS deployments must be able to perform name resolution for internet addresses and private network addresses at the same time to allow endpoints to simultaneously and seamlessly access both internal and internet resources. Well-configured DNS deployments can protect the digital identity and online fingerprint of a modern organization. As domain names have become synonymous with a given entity, it is important for these names to securely resolve to the correct addresses.

In a modern organization's network deployments, DNS is a core and critical part of internet operations. Almost all requests and connections rely on DNS servers, so it is crucial that these servers be configured correctly and managed securely. If a DNS namespace or server is compromised by malicious actors, it can become a threat vector for further security intrusions. The central role of DNS in modern internet infrastructure also allows it to be used as a tool for securing and protecting the rest of the organization. DNS servers can provide significant insight into the connections and dataflows of endpoints and can often prevent security incidents earlier than other systems. As such, this document provides guidelines on the use of DNS as a foundational layer of cyber defense in an organization.

1.2. Impact of DNS on Cyber Resiliency, Defense-in-Depth, and Zero Trust

DNS infrastructure is mission-critical. If it were to fail, entire networks and their applications would become unavailable or simply stop working. As a critical element in an organization's digital resiliency, DNS should be regularly assessed and/or reevaluated.

Recent developments in network security best practices have driven an increased focus on defense-in-depth, which refers to the idea that no defensive measure is infallible and that the

best defense comes from multiple layers of protection. This style of cyber defense yields a more flexible, scalable, and resilient system that is more resistant to compromise. It also aligns more closely with zero trust principles, which assume that no element, node, or service can be implicitly trusted [3]. However, DNS often remains an overlooked element within zero trust strategies, creating a significant gap when it is implicitly trusted. Organizations should verify their information sources by leveraging DNS to enforce security policies, prevent end users and systems from accessing malicious or unauthorized resources, and provide asset visibility for digital forensics and incident response (DFIR). This marks a shift in the role of DNS from a purely operational one to a wide-scale tool to improve internet security and bolster network resilience, but it requires additional considerations to utilize securely.

Zero trust presents a shift from a location-centric model to one that focuses on identity, context, and data with fine-grained security controls between users, systems, applications, data, and assets that change over time. As a DNS resolution may be open to end points that querying public domain names, organizations should secure DNS to prevent it from being used as a route to bypass zero trust network controls. This could involve the implementation of protective DNS services (see Sec. 2.1) and/or specific DNS resolver configurations for certain classes of end points. The DNS is also a source of enrichment data that can be used in access decisions or improvements to security posture. DNS query logs can be analyzed as part of an organization's threat detection program. An organization's DNS service serves as both an information source for policy decision points or policy creation and a policy enforcement point.

1.2.1. DNS Use Cases for Operational Technology, Internet of Things Devices, and Critical Infrastructure

As a ubiquitous part of internet infrastructure, DNS implementations are present in contexts where more advanced security paradigms may be difficult to utilize, particularly the operational technology present in manufacturing, critical infrastructure, and Internet of Things (IoT) deployments. Protecting these devices can often present serious difficulties due to low computing power, physical location or accessibility, bespoke system designs, aging legacy software, and extremely high uptime requirements, among other challenges.

DNS services can provide an additional layer of security and monitoring without negative impacts on operational devices that operate using an IP network. This "overlay" of risk mitigation on top of other existing strategies is a step toward a full defense-in-depth strategy, which is the preferred approach for critical infrastructure and other important high-uptime digitally controlled services.

Options to support the security of operational technology (OT) and IoT devices through the DNS service can include:

- Taking advantage of the capabilities of protective DNS (see Sec. 2.1)
- Implementing highly restricted DNS firewalls so that communication from OT/IoT devices is restricted to the systems needed for their secure operation

1.3. Using This Guide

This document provides deployment guidelines for organizations and administrators to secure DNS protocols, infrastructures, and services (i.e., protective DNS). It is organized as follows:

- Section 2 provides recommendations for using DNS as a component of an organization's security architecture.
- Section 3 discusses the threats to and recommendations for securing authoritative services.
- Section 4 discusses the threats to and recommendations for securing recursive services and stub resolvers (Section 5).

1.4. Audience

This document is intended for two general groups:

1. Cybersecurity executives, decision-makers, and organizational policy-setters
2. Operational networking and cybersecurity teams

While there may be overlap between these two roles, this document's sections are tailored to the distinct duties of each. Sections 1 and 2 are intended for the first group and provide high-level context and impact analysis to inform decision-making. The remaining sections provide technical guidelines to be implemented on the network.

1.5. Objective

This document provides guidelines for securing the DNS infrastructure (both data and protocol use) used to support enterprise operations and describes its role in an organizaton's overall security posture. This includes the deployment of protective DNS that — in addition to the core name resolution functions — provides proactive security functions, such as preventing communication pathways to malicious endpoints. This document also provides guidelines on protecting the integrity of DNS data using DNSSEC and protecting DNS query/response transactions using DNS over a secure transport layer.

2. DNS as a Component of an Organization's Security Strategy

DNS is critical to network connections, and its universal deployment makes it an effective security mechanism. The DNS platform is already in use by all types of clients on the network, including on-premises, in the cloud, and on IoT devices. Thus, any protection provided by DNS infrastructure benefits all clients that use that infrastructure for name resolution, regardless of the type of device.

The original intent of DNS was to distribute information (e.g., host and IP address mappings, mail routing information), so it has not traditionally been viewed as a tool for securing network communications. However, because DNS enables nearly all network communications, it is an effective tool for monitoring and managing those communications. In a typical network communication pattern, the first action performed is to use DNS to resolve the domain name of the target system to an IP address. The communication is unable to begin until that resolution is completed.

According to the Cybersecurity and Infrastructure Security Agency (CISA), "DNS infrastructure is a common threat vector for attack campaigns" [4]. Even when DNS traffic is encrypted, the system still needs to retain access to the internet. This allows malicious actors to set up authoritative servers for command and control (C2) and data exfiltration, where encryption can work to their advantage. Therefore, it is crucial to implement strict controls and auditing on an organization's secure resolvers to ensure that only resolvers with the appropriate policy configuration are permitted to communicate with the internet. Applying security in DNS infrastructure gives administrators the opportunity to identify and review potentially malicious communications before they begin and automatically prevent them from happening.

Two other advantages of using DNS are scale and efficiency. DNS has evolved over decades to support massive networks like the internet, so DNS security tools can handle a tremendous number of clients simultaneously. Name servers can load a large volume of authoritative and threat data. Taking protective action with DNS is also efficient. Because DNS queries precede network communication streams, enforcing policy with DNS prevents malicious or suspicious communication streams from starting. Protective DNS decreases unauthorized traffic on a given network, which benefits the entire network infrastructure by alleviating the burden on other security elements, such as infrastructure components (e.g., firewalls) and human resources (e.g., the Security Operations Center [SOC]).

2.1. Protective DNS

Protective DNS[2] is a DNS service that is enhanced with security capabilities to analyze DNS queries and responses and take action to mitigate threats. Protective DNS blocks access to malicious websites and prevents the delivery of malware, ransomware, phishing, and other attacks that attempt to distribute spyware and viruses. Protective DNS can be provided as a service from a vendor, deployed on internal DNS infrastructure, or a combination of the two.

[2] This is the commonly used name for the described service and does not indicate the CISA program of the same name.

There are potential benefits to using a combination of externally provided protective DNS with internally deployed protective DNS. For example, in most cases, cloud-provided services can leverage a larger amount of threat intelligence and provide more real-time analysis than an individual on-premises DNS server due to the cloud's greater resources. However, if the cloud-hosted service were unavailable for some reason, an organization that is configured to fail open (i.e., fall back to normal recursive resolution) would want on-premises protective DNS capabilities available to ensure protection. While this approach may not be applicable in all cases, this combined hybrid scheme should be utilized where feasible.

The goals of deploying protective DNS include:

- Using the DNS service as a policy enforcement point (PEP) by blocking harmful traffic in real time at the point of domain name resolution, typically before malicious activity starts and/or by preventing the misuse of the DNS protocol for nefarious purposes (e.g., data exfiltration over DNS)
- Blocking categories of traffic with DNS by categorizing domain names that do not conform to an organization's policies or matching against known bad actor lists
- Using the DNS service as an information source to improve a zero trust security posture by delivering visibility into real-time and historical DNS query and response data to facilitate DFIR
- Integrating with the wider security ecosystem as part of defense-in-depth, such as correlating an organization's data on assets (e.g., devices, cloud workloads) and users with the clients that sent queries that were blocked by protective DNS
- Facilitating an organization's responsibility to comply with regulatory or contractual requirements for blocking traffic to disallowed sites (e.g., copyright violations, legal restrictions)

2.1.1. Threat Intelligence and Telemetry

As new networking technologies emerge and attack surfaces evolve, DNS remains a constant security control point for protecting users in all environments (e.g., organizations, mobile endpoints) and monitoring and disrupting malicious communications. Unlike other mechanisms in the security stack, it is not limited to any single type of threat and can often stop complex, multi-stage attacks before they progress. DNS can also protect users and organizations from scams or credential theft via phishing, ransomware, and data exfiltration.

This approach requires integrating threat intelligence into the DNS resolver. Threat intelligence is most effectively leveraged in a DNS infrastructure in an actionable manner via DNS firewalls. DNS firewalls can be implemented in various ways from static or dynamic blocklists to response policy zones (RPZs) and can be seamlessly integrated into the DNS resolution chain via different architectures. Therefore, the consumption and deployment of threat intelligence services should be considered as part of any protective DNS deployment. There are additional ways to use DNS infrastructure to distribute threat intelligence data, such as DNS blocklists (DNSBLs) or reputation blocklists (RBLs), which use DNS to publish threat information that clients can use to

make policy decisions. The biggest difference is that DNSBLs/RBLs implement the enforcement mechanism on the clients (e.g., in email server software), whereas RPZs implement it within the DNS infrastructure, meaning that protection can be provided regardless of the features at the client.

2.1.2. Name Resolution Filtering

Name resolution filtering refers to DNS infrastructure that applies security-related policies to DNS resolution. For example, a protective DNS implementation might refuse to resolve a set of domain names that are known to be used in phishing campaigns or that identify malware command-and-control infrastructure. Instead of resolving these domain names to IP addresses or other types of data, protective DNS generally returns some other form of DNS response to indicate that the domain name does not exist, such as NXDOMAIN (i.e., “non-existent domain”). Protective DNS implementations can also log queries for domain names that trigger policy to indicate potential malware infection or other malicious activity.

Protective DNS is generally implemented by using DNS firewalls or RPZs configured on an on-premises DNS service; a cloud-based, secure recursive DNS service; or some combination of the two. There are far more ways to approach deploying protective DNS than can be addressed in this document, especially when hybrid deployments are considered. When considering the deployment of RPZs on-premises, there are two typical models: closest to the client or closest to the internet. Each model has its own advantages and drawbacks. Putting the RPZ systems closest to the internet often allows for the most concentrated deployment, which may control costs but may also sacrifice client visibility. In contrast, putting the RPZs closest to the client provides the greatest ability to attribute activity to individual clients but may be more costly or require using a smaller amount of threat intelligence on each device. Different architectures may be possible or desired, depending on an organization’s infrastructure or policy. Figure 1 shows an example enterprise that utilizes a cloud-based service but retains a local forwarder for hosts on the enterprise local network.

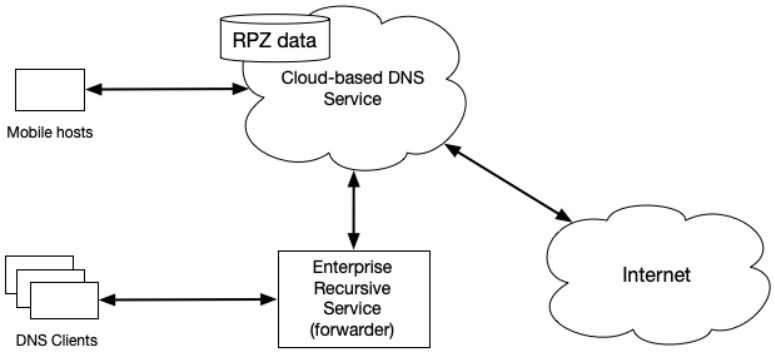

Fig. 1. Enterprise using cloud-based protective DNS with a local forwarder

DNS firewalls allow administrators to specify security policies that are enforced by the DNS service, such as policies that filter out specific domains or IP addresses. Many internet security organizations also provide “feeds” of their current threat data that can be used to dynamically update DNS firewalls or RPZs. Organizations can configure their name servers as secondary

name servers for RPZs and automatically synchronize their DNS resolution policy with the latest threat information. With an on-premises deployment of DNS, DNS firewalls (or locally managed RPZs) provide local control over these policies with very low latency, increased survivability, and minimal performance impact. With the ubiquity of DNS infrastructures and the properties of DNS firewalls, DNS can consistently mitigate threats across entire network infrastructures in a matter of minutes.

Secure recursive DNS services are also available on the internet. Organizations can configure their name servers to forward queries for internet domain names to the IP addresses operated by these services. To ensure the most complete coverage for the organization, a protective DNS service should be capable of analyzing all record types (e.g., MX, SOA, TXT). The services generally offer a web-based control panel that allows administrators to customize the resolution policy applied to queries from the organization's clients. The cloud approach offers greater scalability, storage, and computing power but has disadvantages, such as a loss of confidentiality, higher latency, and challenges in quickly and accurately attributing DNS queries to their sources. Organizations may consider using a combination of these to benefit from the lower latency of on-premises services and the greater scalability of cloud services while allowing for the granular tracking, logging, and attribution of connections and requests. The best choice for a given DNS deployment will vary depending on the specific needs of the network and its users and any regulatory or policy environment under which the enterprise may operate.

2.1.3. DNS for Digital Forensics and Incident Response

Government agencies and regulated enterprises should implement robust DNS traffic logging mechanisms to meet compliance requirements. Logging should capture both current and historical DNS traffic to enable DFIR. These DNS logs should be integrated with other system logs to facilitate correlation with cloud workloads and device or user activities and to enhance visibility and auditability.

Logging all DNS traffic can be resource-intensive. If not done efficiently, it may impact the performance and availability of an organization's DNS services. Alternatively, organizations may consider using cloud-based solutions, efficient logging methods (e.g., DNSTAP format [5]), or selective logging. However, DNS queries and the responses associated with domains that are classified as malicious or unauthorized by protective DNS services should always be logged to support security and compliance objectives. An organization may consider removing entries related to known secure domains from logs to reduce log volume and operating costs before sending them to a security information and event management (SIEM) system but should keep a complete log for future forensics. A known secure domain could be any zone maintained by the organization where there are other monitoring points in place or any other preconfigured zone.

Mapping an IP address to a compromised asset in the event of a cyber attack requires tracking key attributable metadata in real time as well as a history of its allocation to each asset and resource, such as a Dynamic Host Configuration Protocol (DHCP) lease history. To ensure the

rapid notification of queries that might indicate infection or malicious activity, organizations should integrate protective DNS logs from their name servers or their secure recursive DNS service with their SIEM or log analysis platform.

2.2. Protecting the DNS Protocol

DNS is a fundamental network service and must be left open to enable internet connections. As a result, it has been used by threat actors as a strategic vehicle to send malware, conduct data exfiltration, communicate with command and control infrastructure (C2), and other attacks. The Internet Engineering Task Force (IETF) published an early threat model of DNS [6] that primarily focused on its query/response role. However, there are other considerations that require additional protective measures.

If one or more DNS servers comprising a service are compromised, the service itself is compromised. There is little limit to the amount of short- or long-term damage that can be inflicted on clients of the service, often while avoiding detection. To that end, it is crucial to prevent bad actors from using DNS as a threat vector, and hardening the service deployment plays an important role in providing a DNS service. There are three equally important elements to accomplishing this:

1. Protecting internal and external authoritative and recursive DNS services against threats
2. Using Encrypted DNS and authentication to protect privacy and confidentiality
3. Using DNSSEC to protect DNS data integrity and authenticity

2.2.1. Protecting the Integrity of DNS Services

The DNS protocol refers to the standardized communications that carry DNS information between networked entities. Threats to the DNS protocol are numerous but well-studied. Section 3 discusses specific threats and protection approaches to name servers that host DNS information (i.e., authoritative service).

2.2.2. Using Encrypted DNS and Authentication to Protect the Protocol

There are tested mitigation methods available for securing the DNS protocol itself, including security focused on data (i.e., zone data) and channel security (i.e., DNS message transactions).

The DNSSEC is a standardized set of extensions to DNS for securing DNS protocol communications against compromise. It is one part of the wider field of DNS security and should be implemented alongside other best practices and protocol features. While DNSSEC can help protect against the compromise of DNS communications (i.e., integrity protection), it does not provide any confidentiality or privacy protection. Those capabilities are provided by other technologies, such as DNS over Transport Layer Security (TLS) (DoT), DNS over Secure Hypertext Transfer Protocol (HTTPS) (DoH), and DNS over Quick UDP Internet Connections (QUIC) (DoQ) (see Sec. 4.2.1).

The process of authenticating the source of a message and its integrity through hash-based message authentication codes (HMAC) is specified through a set of DNS specifications known collectively as transaction signatures (TSIG) (see Sec. 3.1). HMAC denotes both the message authentication code generated by using a keyed hash function and the hash function itself. HMAC is specified in Request for Comment (RFC) 6151 [7] and generalized in NIST Federal Information Processing Standards Publication (FIPS PUBS) 198-1 [8]. The HMAC function for TSIG is specified in RFC 8945 [9].

Later sections of this document categorize deployment guidelines for securing the DNS protocol by type, including the use of DNSSEC and TSIG:

- Authoritative in Sec. 3
- Recursive in Sect. 4.2
- Stub resolvers in Sec. 5.1

2.2.3. DNS Hygiene and Best Practices

Threat actors can exploit misconfiguration and lapsed domain/DNS name server registration to seriously compromise DNS integrity. Organizations should implement robust processes to continuously monitor and validate the integrity of their public domains and to raise the visibility of attempts to impersonate domains owned by the organization. Section 3 provides an in-depth breakdown of threats that occur due to a lack of proper DNS hygiene and a discussion of best protection approaches.

2.3. Protecting the DNS Service and Infrastructure

DNS software must run on some existing host platform, which includes the hardware, firmware, and software that are required for DNS services to operate. Compromising any of these can potentially compromise the DNS service and cascade into significant operational failures or the loss of integrity and confidentiality. The following non-exhaustive list provides examples of compromises that could jeopardize the integrity of the host, platform, or software on which any given DNS deployment relies:

- **Platform or component vulnerability:** The OS, any system software, or any other application software on the DNS host could be vulnerable to attacks that result in the denial of name resolution services or threats to the hypervisor or underlying cloud infrastructure if the DNS service is cloud-based.
- **Denial of service:** A malicious attacker could send a large volume of queries to perform a denial-of-service (DoS) attack against a server or service. The attacker could also use numerous third-party DNS servers to aid in the attack (i.e., in a distributed DoS or DDoS). Another DoS attack is resource exhaustion, which does not rely on the volume of queries but rather queries that result in significant processor or memory usage by the server, resulting in a degradation or failure of the service.

- **Unauthorized configuration change:** The platform-level configuration file that enables DNS communication could be corrupted or subject to unauthorized modifications due to inadequate protections, resulting in disruptions that vary between the breakdown of communication among DNS hosts to complete failure of the DNS service itself.

In this context, securing the platform and host refers to following the relevant best practices for securely deploying the non-DNS components on which the DNS relies (e.g., securely configuring the operating system that the service is running on, ensuring that the hardware supply chain is well-understood). Like any other network service, DNS requires a large stack of diverse components to function, and providing the best practices for securing all of them is outside of the scope of this document.[3] However, the high-level architectural design of an organization's network infrastructure should be carefully laid out to protect high-criticality elements, including DNS services. Because DNS is a core network service, organizations must provide robust and secure network connectivity to the DNS servers to ensure the uptime of DNS services.

2.3.1. Dedicated DNS Services

Cyber criminals and other actors will seek to amplify and maximize the disruption of any cyber incident by attacking mission-critical systems, especially targets that host multiple critical components. To ensure cyber resiliency, the coexistence of multiple mission-critical services on a single system should be limited (i.e., separation of duties).

Even if a DNS is run on a secure operating system, vulnerabilities in other software programs on that OS can be used to compromise the security and availability of DNS software. Hence, the infrastructure that hosts DNS services should be dedicated to that task and hardened to reduce the attack surface and ensure that adequate system resources are available to the DNS service. The infrastructure should include sufficient capacity for elements of the DNS service, such as logging, the support of encrypted DNS protocols, and protective DNS. This may be easier to accomplish on purpose-built DNS platforms, either as a service or via virtual or physical appliances. If having systems dedicated only to DNS is not feasible, then incorporating related core services (e.g., DHCP) may be appropriate.

Most name servers can be configured to perform authoritative or recursive functions or both. An authoritative name server produces resource records (RRs) from its own zone file or database. Since it is only meant to provide name resolution for the zones for which it has authoritative information, the security policy should have recursion turned off for this type of name server. In contrast, a recursive function would serve RRs from the name server's cache or by querying other name servers. It provides resolution services (i.e., resolving queries on behalf of clients), so protection can be ensured by restricting the number and location of clients from which it will accept queries.

A recursive name server should be run under a different security policy than that of an authoritative name server to mitigate attacks (e.g., cache poisoning). A name server instance that is directly accessible from or has direct access to the internet should be configured as

[3] Refer to relevant documentation, standards, and best practices for any hardware, firmware, software or network infrastructure that is a dependency for a given DNS deployment.

either an authoritative name server or a recursive name server, but any given instance should not be configured to serve in both roles. Name servers that are only accessible from inside an organization's network commonly perform both functions: hosting internal DNS zone data and providing recursion for clients. In that context, combining the functions on the same server is acceptable and practical.

An organization could also classify its authoritative name servers into two sets based on the types of clients and data they serve (i.e., public versus private). External name servers would be located within a DMZ or any public-facing hosting service. These would be the only name servers that are accessible by external clients and would serve zones and RRs that pertain to hosts with internet-facing services. Internal name servers are usually located within the firewall perimeter and should be configured to be unreachable from outside of the organization's network (i.e., provide name resolution services exclusively to internal clients).

DNS software should not run or be present on hosts that are not designated as name servers. Some OS versions may come with DNS server software or other resolution components by default. Hence, while taking an inventory of organizational software in workstations and servers as part of the security audit, remove DNS server implementations from hosts that are not functioning as name servers.

2.3.2. Resiliency and High Availability of DNS Servers

As a critical service, DNS servers should be made as resilient and available as possible to ensure business continuity. This can be accomplished by using network and geographic dispersion and implementing best practices for server backups and recovery.

Every zone must have an authoritative primary server and one or more authoritative secondary name servers. At least two of the authoritative name servers for an organization should be located on different network segments. This dispersion ensures the availability of an authoritative name server when a particular router or switch fails or during an attack on an entire network segment (e.g., a DoS/DDoS attack). Authoritative name servers should also be dispersed geographically (i.e., different physical sites). For example, organizations may locate some authoritative name servers on their own premises and others in a hosting service or partnering organizations. Other solutions could rely on cloud-hosted DNS services, which handle the availability of zone information for the organization.

If the organization hosts the zone information, a network administrator should use a "hidden" primary authoritative server and only have secondary servers visible on the network. A hidden primary authoritative server is an authoritative DNS server that does not appear in the name server resource record set (RRset) for a zone. All of the designated name servers that appear in the RRset must have a copy of the zone data, whether by zone transfer or some other method (e.g., database replication). In effect, all visible name servers are secondary servers, which prevents potential attackers from targeting the primary name server. A hidden primary should only accept zone transfer requests from a specified set of secondaries and refuse all other requests for zone transfers and, ideally, other DNS queries.

2.3.3. Interoperability of the Protective DNS Ecosystem

When an organization deploys a protective DNS service, it must ensure interoperability with the wider security ecosystem so that it is additive to the overall security model. This includes:

- Ensuring that the DNS service is part of a defense-in-depth approach and not an isolated control
- Logging query/response data and blocks/redirects to security operation functions (e.g., via a SIEM/SOAR) to be correlated with other security events
- Being able to access the threat intelligence deployed within the protective DNS service via APIs for use in assessments and DFIR
- Ensuring that DNS components use standardized and interoperable protocols and APIs, as defined by the relevant IETF RFCs
- Sharing and forwarding relevant DNS data to other components of the wider security ecosystem

3. Managing Threats to Authoritative Services

The primary role of an authoritative DNS server is to provide answers to queries for the zones for which it is authoritative. These servers contain zone files, which in turn contain specific information, such as name-to-address mappings (i.e., A RRs for IPv4 IP addresses or AAAA RRs for IPv6 addresses) or address-to-name mappings (i.e., PTR records). This authoritative role could be for external (i.e., internet-facing) clients, internal clients, or both. Most of the recommendations below assume that there is an authoritative service for zones linked to the global DNS tree or internal (i.e., LAN) networks. Certain virtualization platforms (e.g., Kubernetes) have their own DNS implementation for internal lookups and are not usually linked as a delegated zone from the larger, global DNS infrastructure. In such cases, these recommendations may not apply.

There are two types of authoritative name servers: primary name servers and secondary name servers. A primary name server contains zone files that are created and edited by the zone administrator. Sometimes, a primary name server is configured to allow the zone file to be dynamically updated by authorized DNS clients. A secondary name server also contains authoritative information for a zone, but its zone file is a replica of the one in the associated primary name server. The replication is most often enabled through a standardized transaction called a zone transfer, which transfers all RRs from the zone of a primary (or other secondary) name server to the secondary name server. The secondary name servers are notified of any changes to the contents of a zone file on the primary name server through a transaction called *DNS NOTIFY*. When a secondary name server receives this message, it initiates a zone transfer process. Depending on the circumstances, the zone transfer can contain the entire zone file (i.e., AXFR) [9] or the incremental changes since the last AXFR (i.e., IXFR) [10]. To improve fault tolerance, there are usually several secondary name servers in an organization.

When the authoritative role is provided by a cloud-based service, there is often no differentiation between primary and secondary roles. Cloud-based DNS authoritative hosting services may use means other than zone transfers to synchronize DNS data across cloud instances. From a client perspective, there is no difference between a primary or secondary role providing answers to a query because they both provide authoritative answers for the zone. The primary role is only relevant to clients when they are attempting to update DNS records via DNS UPDATE (sometimes called *nsupdate)* [12], which must be serviced by the primary role for the zone.

Domains that are hosted on authoritative DNS infrastructure must be protected against exploitation. Misconfigured or stale Canonical Name (CNAME) RRs or name server delegations allow threat actors to take control of an organization's external-facing domains. This allows the threat actor to use the positive reputation of those domains in attacks against the organization's own users (i.e., spear phishing) or as part of general malware campaigns. Threat actors also increasingly register "look-alike" domains that are not owned by the target organization but which users could easily assume to be associated with that organization.

3.1. Zone Transfer Threats and Protection Approaches

Zone transfers replicate zone data to multiple servers to provide a degree of fault tolerance in the DNS service provided by an organization. While threats from zone transfers have not been formally documented through any IETF RFCs, a common threat to any network transaction is the denial of service. A second common threat to any network service is based on the exploitation of knowledge gained from the information provided by zone transfers.

- Denial of service (DoS): Because zone transfers involve the transfer of entire zones, they place substantial demands on network resources relative to normal DNS queries. Errant or malicious frequent zone transfer requests from name servers can overload a primary or secondary zone server and result in the denial of service to legitimate users.
- Unauthorized zone modification: The zone transfer response message could be tampered with.

A DoS threat can be minimized if secondaries allowed to make zone transfer requests are restricted to a set of known entities. Configuring this restriction into the primary and secondary name servers requires a means of identifying those entities. IP addresses are commonly used but are not secure because they can be spoofed.

One way to protect the integrity of zone contents between primary and secondary servers is by including a hashed digest of the zone contents published as a special RRtype known as a ZONEMD RRtype [13]. This RR contains a hash of the zone file contents. A secondary authoritative server can use the ZONEMD RR to check whether the response to a zone transfer is the same as the zone information hosted by the primary authoritative server. One advantage of the ZONEMD RR is that it does not require a shared secret (see below).

The IETF developed an alternate mechanism called a transaction signature (TSIG) [9], where the mutual identification of servers is based on a shared secret key. Because the number of servers involved in a zone transfer is generally restricted to name servers in the same administrative domain of an organization, a bilateral trust model that is based on a shared secret key may be adequate for most organizations. TSIG specifies that the shared secret key be used for both mutual authentication and for signing zone transfer requests and responses to protect against tampering.

Asymmetric cryptography (i.e., public-key cryptography) can also be used to authenticate DNS transactions using a special type of RR referred to as SIG(0). The format of the SIG(0) RR [14] is similar to the resource record signature (RRSIG) RR and can be validated using a public key stored in the DNS instead of a shared secret key. While SIG(0) can be more computationally expensive to use, a previous trust relationship may not be necessary to use SIG(0)-signed messages. However, because most zone transfers occur between parties that have a previously established relationship, it is considered easier to implement TSIG for authenticating zone transfer transactions.

The use of transaction signatures (i.e., TSIG or SIG(0)) only provides authentication and integrity protection. The confidentiality of zone transfers can be provided by using TLS [15]. A lower-level network layer solution (e.g., IPsec or other secure network communication technologies)

may also provide confidentiality and remove the need for additional authentication at the application layer. However, this is usually provided by the underlying host/infrastructure and not the DNS implementation.

3.1.1. Restricting Zone Transfer Transaction Entities

Authoritative name servers (especially primary name servers) should be configured with an access control list (ACL) that designates the hosts from which zone transfer requests will be accepted. These restrictions address DoS threats and potential exploits from the unrestricted dissemination of information about internal resources. Hence, zone transfers from primary name servers should be restricted to secondary name servers. Servicing zone transfers should be completely disabled in secondary name servers unless they are also intended to provide zone transfers to other secondary name servers. In that case, the zone transfers on the secondary should also be protected with ACLs.

3.2. Zone Content Threats and Protection Approaches

DNS data is made up of zone information and configuration information. All of the security deployment options discussed in this section relate to zone file contents, particularly:

- Parameter values for certain key fields in RRs of various RRtypes
- The presence of certain RRs in the zone file

The various types of data in the zone file result in different security exposures and potential threats.

3.2.1. Lame Delegations

Incorrect zone delegation (i.e., lame delegation) (see Appendix B) can result in a child zone becoming unreachable (i.e., denial of service) or intermittently accessible if only one or more NS RRset entries points to a non-existing server. To mitigate this, authoritative DNS admininstrators should regularly check delegation information for correctness. This applies to both the zone's delegation from a parent zone to any child delegations within the zone.

3.2.2. Zone Drift and Zone Thrash

There may be a mismatch of data between the primary and secondary name servers if the Refresh and Retry fields in the start of authority (SOA) RR [2] of the zone are set too high and the zone file is changed frequently. This error is called *zone drift* and results in incorrect zone data at the secondary name servers. If the Refresh and Retry fields in the SOA RR are set too low, the secondary server will initiate zone transfers frequently. This error is called *zone thrash* and results in more workload on both the primary and secondary name servers. Such incorrect data or increased workload may result in degradation or the denial of service.

The Refresh value should be determined by the expected rate of change for a zone. Suggested values usually range from 1200 seconds (20 minutes) to 432000 seconds (12 hours) or even

864000 seconds (1 day). The Retry value is the time that a secondary should wait after a failed refresh before retrying and should be a fraction of the Refresh value for the zone.

3.3. Dynamic Update Threats and Protection Approaches

3.3.1. Dynamic Update Misuse

Dynamic updates involve DNS clients making changes to zone data in an authoritative name server in real time [12]. As with zone transfer transactions, the threats associated with dynamic update transactions are documented by the IETF in RFC 2136 [12]. However, the following common threats could be expected:

- **Unauthorized updates** could have several harmful consequences for the content of zone data, such as:
    - Adding illegitimate resources
    - Deleting legitimate resources
    - Altering delegation information (NS RRsets pointing to child zones)
- Update tampering could affect data in a dynamic update request.
- Replay attacks (i.e., update request messages captured and resubmitted later) could cause inappropriate updates or be part of a DoS attack.
- A large volume of dynamic updates could be used to degrade or disrupt services since dynamic update processing is more resource-intensive than answering queries and is prioritized higher than answering queries in some DNS server software. This is why the use of dedicated hidden primary servers for zones that allow dynamic updates is strongly encouraged.

3.3.2. Guidance on Securing Dynamic Updates

Unauthorized or tampered updates could be countered by authenticating the entities involved and providing a means to detect message tampering. The TSIG mechanism can meet the security objectives for zone transfer and protecting dynamic updates. Although the dynamic update message contains some replay attack protection in the prerequisite field of the message, TSIG provides an additional protection mechanism by including a timestamp field in the dynamic update request. This signed timestamp enables a server to determine whether the timing of the dynamic update request is within the acceptable time limits specified in the configuration. Security can also be enhanced using a lower-level network layer mechanism, such as IPSec.

Dynamic updates to a zone can only be directed to the zone's primary name server, which holds the writable copy of the zone file. Dynamic update requests generally originate from hosts (e.g., DHCP servers) that attempt to provision a new host name in the DNS zone when an IP address

is dynamically assigned to a client. Because dynamic update messages change the authoritative zone data, they should only be accepted from authorized senders (e.g., by TSIG or ACL).

3.4. DNS NOTIFY Threats and Protection Approaches

3.4.1. DNS NOTIFY Misuse Threats

DNS NOTIFY is a message sent by primary name servers that cause secondary servers to start a refresh operation (i.e., query for SOA RR to check the serial number) and perform a zone transfer if an update to the zone has occurred. Because the NOTIFY message is only a signal, there are only minor security risks in dealing with the message. The primary security risk to consider is spurious NOTIFY messages.

3.4.2. DNS NOTIFY Protection

Receiving spurious DNS NOTIFY messages results in an increased workload for secondary name servers because a zone transfer will only occur when an updated zone is on the primary server. Because this threat is low-impact, the required protection approach is to configure the secondary name servers to accept DNS NOTIFY messages only from the zone's primary name server or from a secondary for secondary name servers configured to transfer zones from another secondary. However, if TSIG is set up for use for all communication between a set of hosts, TSIG will be used with NOTIFY messages as well.

Once zone transfers have been set up between servers, secondary name servers should be informed about changes to zone file data through a notification message. By default, a DNS NOTIFY message is sent to every recognized secondary name server of the zone (i.e., name servers listed in the NS RRset in the zone) when a primary name server detects a change in the zone file. Most DNS server software provides the ability to also notify other servers, which allows the administrator to account for any hidden or stealth name servers for the zone (i.e., not listed in the NS RRset). DNS administrators should keep notifications on to allow updates to be propagated quickly to secondary name servers.

3.5. Minimizing Information Leakage

As part of operational security, it is important to minimize the amount of information that can be gathered from a DNS server without credentials. This information can be used to launch attacks on the DNS server or to quickly survey a larger organization's network.

Not all information leakage is preventable. Some information stored in DNS servers must be public in order for DNS to function correctly. This section provides recommendations on how to best reduce information leakage without compromising core functionality.

3.5.1. Resource Record Information

Attackers can map an organization by using RRtypes to learn about available services, such as mail exchange (MX), server selection (SRV), RRtypes to encode digital certificates used in other protocols (TLSA), and text strings (TXT). Types of RRs in the DNS that are meant to convey information to humans and applications about the network, hosts, or services include the responsible person (RP) record, host Information (HINFO) record, location (LOC) record, and catch-all TXT RRtype [2]. Although these record types are meant to provide information to users in good faith, they also allow attackers to gain knowledge about network hosts before attempting to exploit them. For example, an attacker may query for HINFO records to find hosts that list an OS or platform known to have exploits. Therefore, a best practice is to exclude these record types from internet-facing zones.

More careful consideration should be given to the TXT resource record type. A DNS administrator will have to decide whether the data contained in a TXT RR constitutes an information leak or is a necessary piece of information. For example, several authenticated email technologies (e.g., sender policy framework [SPF], domain keying for internet mail [DKIM], domain-based message authentication, reporting and conformance [DMARC]) use TXT RRs to store email sender policy information, such as valid email senders for a domain [16]. These judgments will have to be made on a case-by-case basis.

3.6. External Authoritative Domain Integrity

Threat actors often target legitimate public-facing DNS domains to reduce suspicion and improve the efficacy of their phishing and malware campaigns.

3.6.1. Dangling CNAME Exploitation

When a DNS CNAME record links two domain names together, there is a risk that the parent domain of the canonical name that the record points to does not remain registered by the target organization. As a result, threat actors can register the delegating parent zone and cause DNS resolutions to resolve to the threat actor's controlled domain. CNAME records can also be exploited if the canonical name resolves to an IP address that is no longer in use by the domain owner, and the attacker can gain control of that IP address to conduct attacks.

DNS administrators should develop policies and procedures to regularly monitor and assess the configurations and registrations of these domains. When they are no longer required, CNAME records should be deleted.

3.6.2. Lame Delegation Exploitation

A lame delegation can result in domain hijacking. When a subdomain is delegated to a DNS-hosting provider and the contract for providing DNS services for that domain lapses, threat actors could hijack resolution for that subdomain by contracting with the provider that controls the servers named in the delegation to host that subdomain. This then enables the threat actor

to redirect resolution requests to their own infrastructure. DNS administrators should actively validate that there are no lame delegations within their external authoritative domain name space and use DNS-hosting providers who apply safeguards.

A similar threat exists if a zone is deleted from a hosting service, but the original delegation is not removed from any registration service. This may allow an attacker to use the same hosting service to hijack resolution for the delegation. Domain administrators should follow a deletion procedure that involves removing all delegation information and the zone data itself.

3.6.3. Look-Alike Domain Exploitation

Threat actors extensively leverage look-alike or *typosquat* domains to impersonate target organizations. By leveraging the positive reputation of legitimate organizations, threat actors vastly increase the success rate of their phishing and malware campaigns. These look-alike domains can include subtle variations of legitimate domains or text or character substitution (e.g., homoglyphic characters from international scripts) to register a domain that appears to be owned by a legitimate organization.

A common best practice is to monitor new DNS registrations to detect this attack vector and defensively register look-alike domains if feasible. Gaining visibility into these activities will enable organizations to preemptively address a prevalent threat vector that targets their users and consumers. An attacker may also attempt to register retired delegations to impersonate the organization. The organization should monitor for these re-registrations or, if practical, maintain retired delegations as "parked" (i.e., registered but not appearing in the DNS) for a period of time.

3.7. Operational Recommendations

3.7.1. Resource Record TTL Value Recommendations

Each RRset in a zone has its own time-to-live (TTL) value that tells recursive DNS servers and clients how long (in seconds) the RRset data should be stored in its cache upon receipt, although not all clients cache responses or strictly obey TTLs. After a recursive server receives a DNS response to a query and performs all relevant checks, it stores the resulting RRsets in its cache. The server cache decrements the TTL value of each RRset in its cache, and that data is purged from the cache (excluding DNS services with prefetching or similar mechanisms) when the TTL for any RRset reaches zero. This prevents caches from growing too large, removes old and possibly incorrect DNS data from caches, and prevents stale results from being returned to future queries, except when a capability like serve-stale [17] is in use.

The zone administrator can assign the default TTL value for all RRs in the zone, but each RRset can have its TTL individually specified, and different RRsets in the same zone can have different values. The zone administrator should set a default TTL value that is long enough for the RRset to be useful for caches but short enough that any changes to the RRset will be propagated quickly through the DNS (and old information purged). DNSSEC signature validity periods should

also be taken into consideration. TTL values should be less than the validity period of the RRSIG RR that covers the RRset. DNSSEC-aware clients will decrement the TTL value of an RRset in its cache to the signature expiration date if that date is before the projected TTL. That way, the RRset will be purged before the signature expires and seen as BOGUS by DNSSEC validators. However, DNSSEC-unaware clients may not know to do this comparison, so there is a risk that invalid DNSSEC RRsets will be stored in DNSSEC-unaware caches.

TTL values should be in the order of hours with a recommended range of 1800 (i.e., 30 minutes) to 86400 (i.e., 1 day). If a zone administrator knows that the DNS data is likely to change frequently or shortly in the future, the TTL value could be set lower for those specific records to ensure that old, stale data is purged from server caches. If the zone administrator believes that the DNS data will not change frequently, then the TTL value can be set higher to optimize the benefits of caching in recursive servers. While some specialized load-balancing scenarios rely on much shorter TTL (i.e., 60 seconds or less), 30 minutes to 24 hours is sufficient for most DNS data. If the data is signed using DNSSEC, the value should always be long enough to ensure that the data will not be purged from caches before validating resolvers can validate it. Very low TTL values (i.e., 30 seconds or less) can cause problems with DNSSEC-validating caches and should be avoided for DNSSEC-signed RRsets. Even for non-signed zones, extremely low TTL values should be avoided [18]. In particular, a TTL of 0 should never be used because it has been known to cause a multitude of issues in some cache implementations. Even a TTL in the range of 5-30 is significantly better than 0.

Operators may wish for certain DNSSEC RRSets to have short TTLs for operations, such as Delegation Signer (DS) or DNSSEC keys (DNSKEY) RRs. Shorter TTL values allow for key changes to occur more quickly as old DNSSEC information is purged from client caches. This is especially important during an emergency key change because it limits the time when a potential or known key compromise can be exploited.

3.8. **DNSSEC Signing Considerations for Authoritative Service**

DNSSEC refers to a set of protocol extensions that add source authentication and integrity protection to DNS data [19]. DNSSEC was designed to protect the data itself from tampering. In DNSSEC, a digital signature covers a given RRset in a response and is encapsulated through a special RRtype called RRSIG. The keys used to validate these digital signatures are also stored in the DNS in DNSKEY RRsets. Trust in the public key is established by building a chain of validated signatures and public keys from signed DNS data to a trusted public key that is preconfigured on the system. These keys are sometimes operationally differentiated as zone-signing keys (ZSK) and key-signing keys (KSK). The preconfigured public key of the trusted zone is called the *trust anchor*.

DNSSEC can guarantee the integrity of name resolution response data to DNS clients that perform DNSSEC signature verification. DNSSEC does not protect the confidentiality of DNS query/response data. Confidentiality can be protected using DoH, DoT, or DoQ (see Sec. 4.2.1). Likewise, DoH, DoT, and DoQ only protect the query/response transaction and do not provide integrity protection and source authentication for DNS data in responses.

Deploying DNSSEC for an authoritative DNS service requires steps to digitally sign the zone data and configure the authoritative service [20]. The exact process for these steps depends on the implementation used by the organization.

Due to the naming convention, DNSSEC is often conflated with the wider and more general concept of DNS security. However, DNSSEC is only one component of a larger whole, and more tools should be utilized to achieve more comprehensive DNS security.

3.8.1. DNSSEC Key Considerations

DNSSEC is based on public-key cryptography to generate digital signatures over DNS data. The current set of cryptographic algorithms is defined in RFC 8624 [21], which lists the mandatory and optional algorithms supported by DNSSEC tools. Using RFC 8624 and recommendations for key strength and lifetimes in SP 800-57 [22] and SP 800-131A [23] produces the digital signatures algorithms shown in Table 1.

Table 1. DNSSEC key parameters based on algorithm

Signing Algorithm	DNSSEC code	Public Key Length (in bits)	Signature length (in bits, no RRSIG encoding)
RSA with SHA-256	8	2048 bits or greater	~2050-4100
ECDSA P-256 with SHA-256	13	256	512
ECDSA P-384 with SHA-384	14	384	768
Ed225516	15	256	512
Ed448	16	456	912

SP 800-57 recommends required security strengths and lifetimes for cryptographic key material based on intended use. A strength of 112 bits is acceptable until 2030, when it will be raised to 128 bits. This is for DNSSEC signing and not validating, as using keys (or algorithm suites) for backward compatibility with entities that do not have the same cryptographic requirements is allowed. Additionally, some organizations (e.g., federal agencies) have additional requirements regarding the use of FIPS 140-certified cryptographic modules [24].

As there are restrictions on the total size of a DNS response (without resorting to using TCP), administrators should be aware of how the key and digital signature size affect the DNS response size. Therefore, algorithms like ECDSA and ed448 are preferable to RSA, as they produce smaller key and signature sizes.

Table 1 does not include post-quantum cryptographic (PQC) algorithms. At the time of writing, the use of PQC algorithms for DNSSEC has not been specified. However, administrators should consider migrating to PQC algorithms once they have been specified and tools have been updated to include them. ECC algorithms with smaller key sizes would be more vulnerable to a quantum attack, as it would require a currently theoretical quantum computer with fewer qubits than would be required for an RSA key with the same cryptographic strength [25]. Since

DNSSEC does not provide confidentiality, the risk of "store and future decrypt" with quantum computers would not be a threat.

The key lifetime value is the period during which a key pair should be considered active and used. Afterward, the key is retired and no longer used to generate signatures. Signing keys used for DNSSEC are categorized as signature keys with a recommended maximum lifetime of 1-3 years [22]. Key lifetime values do not have a representation in DNSKEY or RRSIG RRsets. They are completely internal to the enterprise and set by local policy. The act of retiring a signing key pair and introducing a new signing key pair is called a *key rollover* in DNSSEC. As with other DNSSEC operations, there are automated tools available to perform ZSK key rollovers with minimal administrator intervention based on the enterprise's policy on key lifetimes. KSK key rollovers involve the (delegating) parent zone, which may be operated by a different organization. Since there are means to automate this process [27], implementation must be conducted by both the parent and child zones. Authoritative DNS zone operators should be aware of the preferred KSK rollover option for their delegating parent zone before starting a KSK rollover.

If practical, a hardware security module (HSM) is recommended to store private keys, usually the KSK. This provides additional protection for private keys and would allow for them to be transferred from one primary server to a backup if an issue develops. Using HSMs may not be practical for all zones but should be considered for important authoritative zones.

3.8.2. Using RRSIG Validity Periods to Minimize Key Compromise

The best way for a zone administrator to minimize the impact of a key compromise is by limiting the TTL and RRSIG validity period of the DNSKEY RRs in their zone and the corresponding DS in the parent zone. This strategy limits the time during which an attacker can take advantage of a compromised key to forge responses. An attacker that has compromised a ZSK can only use that key during the KSK's signature validity interval, and an attacker that has compromised a KSK can only use that key during the signature interval of the RRSIG covering the DS RR in the delegating parent. Therefore, DNSKEY RRISG validity periods should be kept short (e.g., 5-7 days), depending on the organization's policies. Validity periods should be significantly longer than the TTL value, as a value that is too low will require more frequent DNSSEC resigning operations.

3.8.3. Hashed Authenticated Denial of Existence

A side effect of the DNSSEC security extensions as they were first specified is the ability for a user to "walk" a zone by sending a series of queries for NSEC RRs. A client could send a query for an NSEC RR in the zone and "walk" the zone by sending a follow-up query for the NSEC RR to the next name indicated in the received NSEC RR. This would result in a client being able to enumerate the entire contents of a zone. While this is not an attack by itself (all DNS data is considered public), it would most likely be a prelude to an attack. An attacker could enumerate a zone to discover the IP addresses of servers to attack directly. This concern led to the creation of NSEC3, which uses hashed FQDNs in the response [28]. Using NSEC3 leads to an increased

burden on authoritative servers and clients to generate and validate responses that contain NSEC3 RRs. At best, NSEC3 can increase the cost of zone walking to an equivalent attack via brute force queries to the zone, but it is often still possible to obtain the full list of owner names in the zone [29].

This has brought into question whether the computational effort that NSEC3 requires is worthwhile. The more prudent approach is to use NSEC. If the use of NSEC3 is still required due to local policy or for other features that NSEC3 provides, the NSEC3 parameters should be set to minimize the DoS risk [30].

Additionally, the only hash algorithm specified for use with NSEC3 is the SHA-1 algorithm. Published research has shown that SHA-1 may be more vulnerable to attack than previously believed. If a new, stronger hash algorithm is specified for use with NSEC3, authoritative operators should plan to migrate to the new algorithm once it is available and widely supported by DNSSEC validators.

3.8.4. Compact Denial-of-Existence Responses as an Alternative

An alternative method of responding with a denial-of-existence answer is to use compact denial-of-existence responses, which are currently a work in progress in the IETF [31]. This method requires the authoritative server to generate a response with NSEC (or NSEC3) RRs that would normally appear when responding to a query where the queried for RRtype does not exist in the zone. The client behavior in these responses does not change, as this is still a failure case when the query was for data that does not appear in the DNS. There are some minor differences in some zone deployments, but using compact denial-of-existence responses may be an alternative to organizations seeking to mitigate zone walking threats.

3.8.5. DNSSEC Algorithm Migration

It may eventually be necessary to migrate to a new DNSSEC signing algorithm due to a discovered weakness in the currently used algorithm or overriding policy decisions. Migrating from the current DNSSEC algorithm to a new algorithm requires steps and delays while old data is removed from caches. A proposed process can be found in RFC 6781 [20]. To reduce the risk of a validator thinking it is under a downgrade attack, the signatures for the new algorithm are added before the DNSKEY RR with the new algorithm's public key or any DS RR in a delegating parent zone, which should be the last addition. Likewise, when removing the retiring algorithm, the public key DNSKEY RR is removed first, followed by the signatures.

DNSSEC-enabled responses can grow large enough to require TCP during the algorithm rollover period and while two RRSIG RRs are present for every RRset in the zone. Administrators need to consider the response size and the complexity of operations when initiating an algorithm rollover.

3.8.6. DNSSEC Signing Internal Zones

While DNSSEC signing of internal DNS zones is likely to be of limited value for most organizations, some circumstances (e.g., in government agencies) may require it. If internal zones must be signed, it is likely that the administrators already know that it is required. Otherwise, DNSSEC signing of internal zones is not recommended.

There are three major reasons why signing internal zones may prove difficult:

1. Depending on the namespace, it could be difficult or impossible to tie the internal zone back to the internet chain of trust. If there is no chain, validating the internal zones requires the management of additional trust anchors for each apex zone on all of the validating resolvers used within the organization.
2. If the zone is being dynamically updated, the zone will have to be re-signed each time it is updated, which can be computationally intensive. For large or busy zones, frequent re-signing could lead to the degradation or complete denial of DNS services for clients. Authoritative servers that support incremental signing (i.e., only generating RRSIGs that are impacted by the change) may mitigate this factor, depending on the availability and the rate of change in the zone.
3. DNSSEC-signing the internal zones serves no purpose when authoritative and recursive services are combined on the same DNS servers, which is a common deployment for internal DNS systems. In this type of architecture, some DNS server software will not perform validation because it is authoritative for the zone, and even if it did, the DNS server would only be validating its own authoritative data.

If there are requirements for signing internal zones, the organization needs to consider various trade-offs, depending on the DNS architecture in use and administrative complexity. For example:

1. Having a separate delegation for the internal zone is recommended. This includes a delegation from the external delegating parent zone. This makes a chain of trust to the DNS Root Zone possible, which avoids having to manage multiple trust anchors in validators. However, this leaks the name of the delegation and has the same threats to the validation chain as to external zones (e.g., disruption of service in upstream zones may result in validation errors of internal responses for internal hosts).
2. A separate delegation may not be delegated from the external zone. This hides the internal zone but requires any validating resolver to maintain the zone's public key for DNSSEC validation.

As with any internal zone, the organization's hosts need to be able to reach an internal recursive service or server that can reach internal authoritiative name servers.

3.8.7. Use of Authoritative Time Source

For DNSSEC, a signer and validator must know the correct time to encode and verify RRSIG RRs. This means that DNSSEC signers should have an authoritative source for time and use a secure

method to learn and maintain the correct time, such as Network Time Security (NTS) [32] with redundant time sources. Configuring this for a DNSSEC signing service depends on the service implementation and is beyond the scope of this document.

4. Managing Threats to Recursive/Forwarding Services

A recursive name server for an organization performs the name resolution function on behalf of a collection of clients. A recursive name server may also be called a caching name server or a recursive resolver [33]. The name resolution function is performed by a recursive name server in response to queries from a stub resolver. The search process for name resolution may involve searching its own cache, forwarding to other designated name servers, recursively querying various authoritative name servers through a set of iterative queries, or a combination of these methods (see Appendix A).

Some name server implementations can be configured to be both an authoritative and a recursive name server. In this configuration, the same name server provides authoritative responses for queries that pertain to authoritative zones while it performs the resolving functions for queries that pertain to other zones. To perform the resolving function, the server must be configured to allow recursive queries, which makes it more vulnerable to attack than a server that does not allow such queries. Since authoritative information might be compromised, it is not a good security practice to configure an internet-accessible name server (often referred to as an external name server) to perform both authoritative and recursive functions. However, it can be an acceptable and efficient configuration for purely internal name servers that are not internet-accessible.

A recursive service is accessed via an IP address, which could be a local recursive resolver or a cloud-based service (public or private). Cloud-based services can have the advantage of global availability and larger, more active caches since they serve millions of clients. However, there may be trade-offs in control and visibility when an organization decides to rely solely on public recursive services.

4.1. Threats to Recursive/Forwarding Services

Compromised transactional queries and incoming responses can threaten recursive servers. A forged or bogus response is different from those expected from legitimate authoritative name servers. Threats to the recursive service could include:

- A compromised authoritative name server (for queries that originate from a recursive name server)
- A poisoned cache of a recursive name server (for queries that originate from a stub resolver or a forwarding recursive name server)
- A recursive name server that is induced to query for a specific name, and forged responses are immediately sent to the server attempting to insert a malicious answer into the cache (mitigations have historically been put in place to prevent this)
- Specific queries that expose bugs in name server software implementations to launch a DoS attack or impact operations in some way
- A passive monitor that can observe DNS queries sent from a stub resolver, which could lead to a loss of privacy for end users of the host system or a monitor learning what

applications or services are running on the host in order to identify communication patterns between the host and network services

- Improper DNSSEC management by the domain owner (e.g., during key rollovers), although this is not due to malicious activity

The cache of a recursive (caching) name server could be poisoned by the following attacks:

- **Packet interception.** The attacker eavesdrops on a request and sends a response by spoofing a name server before the real response from the legitimate name server reaches the recursive name server.
- **ID guessing and query prediction.** The attacker guesses the ID field in the header of the DNS request message (e.g., using brute force guessing) and possibly the query name (QNAME) and query type (QTYPE). The attacker then injects bogus data into the network as a response by spoofing a name server [6].
- **Responses from a compromised authoritative name server.** A compromised authoritative name server is directed by a controlling adversary to send out bogus responses to queries from recursive name servers.
- **Incorrect expansion rules applied to wildcard RRs.** Many zones use wildcard RRs to economize on the volume of data in the zone file. The wildcard patterns are used for synthesizing RRs to generate responses to queries, as described in RFC 1034 [1] and RFC 4592 [34]. If synthesis rules are applied incorrectly in a name server, the RRs associated with existing organizational resources may not be generated or made available in a DNS response. This fault also results in a denial of service.
- **Removal of RRs from a response.** An attacker could also remove RRs from a response, which may result in a name resolution failure and consequent denial of service.

4.2. Recommendations for Protection

DNSSEC was designed to provide authentication for DNS data but does not protect the confidentiality of DNS query/response transactions. As originally specified, DNSSEC and non-DNSSEC query/response are sent unencrypted and are vulnerable to passive monitoring. This allows a third party to observe the queries being sent by a stub resolver on a host system, see what network services the host is communicating with, and develop a possible “fingerprint” of the client device (and associated end user) for later tracking. Some organizations’ security services also use DNS monitoring to detect unauthorized communication or potential malware operating within an organization.

4.2.1. Encrypted DNS

Communication between stub resolvers and the recursive DNS servers they query has traditionally been unencrypted. The DNS messages exchanged by stub resolvers and DNS servers have a binary encoding that is widely understood and easily decoded. This communication has, therefore, been subject to both interception and spoofing, which can

reveal sensitive information or allow an attacker to redirect unsuspecting users to malicious sites.

To address these threats, the IETF has developed several enhancements to DNS that are collectively known as Encrypted DNS. Each protocol enhancement can encrypt communications between stub resolvers and recursive DNS servers in slightly different ways:

- DoT [35] runs the traditional DNS protocol over TLS, which is the same layer of encryption used to secure traffic between web browsers and web servers that use HTTPS to communicate. DoT defines port tcp/853 for use to send queries.
- DoH [36] runs the DNS protocol over HTTP, which in turn runs over TLS using the defined ports tcp/443 and udp/443.
- DoQ [37] runs DNS over QUIC, which includes a encrypted transport layer that runs over UDP on port udp/853.

All of these protocols optionally allow recursive DNS servers to authenticate themselves to stub resolvers, addressing the threats of interception and spoofing.

4.2.1.1. Encrypted DNS Guidance and Recommendations

The U.S. Government requires the DNS infrastructure of Federal Civilian Executive Branch (FCEB) agencies to use encrypted DNS when communicating with agency endpoints, wherever technically supported [4].

Many organizations will find that their options to implement Encrypted DNS protocols are limited by the protocols that their applications or operating systems currently support. Organizations may also need to configure applications (e.g., browser software) that use encrypted DNS so that local resolvers that act as a point of control and logging are not bypassed. Some endpoints (e.g., IoT devices) may not have the necessary software modules to use encrypted DNS on their own and may require a forwarding DNS server that acts as a recursive service, as shown in Fig. 2.

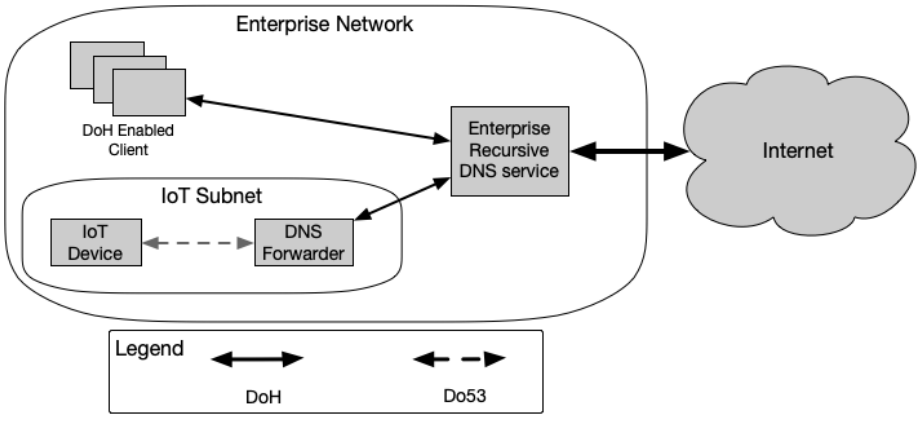

Fig. 2. Mixed use of Legacy DNS over UDP port 53 (Do53) and DoH

While this is not a complete zero trust architecture, it still allows for strict policies on outgoing DNS traffic. Also, depending on architectural requirements for responsiveness and availability of the DNS service, the Enterprise Recursive Resolver could be implemented as a cloud service.

4.2.1.2. Considerations for Using Encrypted DNS

Encrypted DNS is crucial for enhancing online privacy and security. Encryption helps protect sensitive information from being exposed or manipulated and reduces the risk of attacks (e.g., DNS spoofing, machine-in-the-middle attacks). It is a vital component in broader organizational strategies for securing internet communications.

However, Encrypted DNS introduces additional overhead, particularly on name servers, because of the need to perform encryption and decryption when sending and receiving DNS messages, respectively. Organizations should anticipate this and ensure that their name servers have sufficient resources to handle their query load before beginning any widespread deployment of Encrypted DNS.

The use of Encrypted DNS may also make troubleshooting more difficult because IT staff using network troubleshooting tools will not have ready access to the contents of DNS queries or responses, though that information will still be available on the name servers themselves.

4.2.1.3. Cryptographic Guidance

All varieties of Encrypted DNS support server authentication, but this requires the configuration of a server certificate on each name server that receives queries over Encrypted DNS. This certificate can be generated and supplied by either an internet or internal certificate authority.

If supported, name servers must be configured to only allow the cryptographic ciphers permitted in SP 800-52 [38], and cloud-based Encrypted DNS providers should be assessed for their support of this configuration. Additionally, some organizations may have requirements on the use of FIPS-140-approved cryptomodules for use with TLS or DTLS [24], which would also include Encrypted DNS.

4.2.2. Restricting the Use of DNS With Public Providers

In order to ensure that users do not use unauthorized public, internet-based DNS services, organizations should:

- Block outbound DNS from the internal network to the internet, except for name servers that are authorized to communicate directly with name servers on the internet (e.g., forwarders). This blocking can be implemented using firewall rules or router ACLs because DNS uses two well-known ports: UDP port 53 and TCP port 53.
- Restrict stub resolvers to only use encrypted DNS on authorized DNS services wherever possible.

- Block unauthorized DoT traffic from the internal network to the internet using firewall rules or router ACLs. DoT is straightforward to block because it uses the well-known TCP port 853.
- Block unauthorized DoH traffic from the internal network to the internet using RPZs and firewall rules. DoH is more difficult to block because it uses the same TCP port as TLS: 443. RPZs can help block the resolution of the domain names used to identify known DoH servers, and firewall rules can block access to public DoH services that run on dedicated IP addresses.
- Block DoQ traffic from the internal network to the internet using firewall rules or router ACLs. DoQ uses the well-known UDP port 853.
- Use mobile device management (MDM) or other central management solutions to prevent users from configuring non-approved external encrypted DNS services.

4.2.3. QNAME Minimization

Another feature to strengthen the privacy of DNS query/response transactions is query name (QNAME) minimization [39]. This is not a protocol extension but changes how a recursive service constructs iterative queries to obtain the final answer to a stub resolver's query. The goal is to only expose the minimum information needed to obtain a response that will produce the "next step" in the overall resolution process. While QNAME minimization does provide necessary information about the original query from the stub resolver, that information is limited to only what is needed to continue the resolution process.

To protect the privacy of stub resolvers, a recurisve service operator may enable QNAME minimization in the DNS server software used in their deployment. There is some cost of possible additional queries, but it is negligible over time as the cache is populated.

4.2.4. Detecting and Mitigating Data Exfiltration via DNS

Organizations should establish controls to detect and block unauthorized applications from tunneling data within DNS packets. Signature-based systems can detect well-known DNS tunneling tools, but customized DNS data exfiltration tools should also be considered. Such techniques often follow similar patterns, including:

- A larger than normal volume of queries from a host
- Unusual or changing pattern of queries
- Queries for seemingly odd QNAMES (detected by taking entropy measurments of QNAMEs)
- Queries for hostnames in known malicious domains (e.g., known via threat intelligence feeds)

How these measurements are done or configured depends on the tools used by the enterprise. Network-based tools that passively observe DNS traffic will be impacted by the use of encrypted DNS unless the tool is configured to act as a proxy or forwarder.

4.2.5. Enabling DNSSEC Validation

Configuring DNSSEC validation requires two tasks: configuring the server to perform validation and configuring one or more trusted public keys to act as trust anchors. The method for achieving this depends on the DNS implementation used.

The policy for determining which DNSSEC public keys to configure as trust anchors is beyond the scope of this guide and would be part of the organization's security policy. However, given the hierarchy used in DNS, the higher the public key in DNSSEC, the wider range of DNS responses can be validated using that key, as shown in Table 2.

Table 2. Trust anchor selection

Example Public Key	Could Validate
root "."	All signed TLDs and below
.gov	All signed names under ".gov"
Example.gov	Only names in the "example.gov" domain

Therefore, the internet root key should be configured as a trust anchor.

4.2.6. Maintaining DNSSEC Trust Anchors

When a zone updates its DNSSEC signing keys (i.e., performs a key rollover), any validating recursive resolver that has configured that zone's key as a trust anchor must obtain the new singing key. If they key is not updated, DNSSEC-signed responses from the zone (or delegated zones) will fail DNSSEC validation.

There is an automated process for a zone to signal to validating recursive resolvers that it is performing a key rollover [40]. Administrators may not get advanced out-of-band notifications of a rollover (e.g., via email), which makes relying on manual trust anchor updates risky. Administrators of validating recursive resolvers should enable automated trust anchor rollover and monitor logs to ensure stability in the recursive service.

4.3. Operational Recommendations

There are additional steps that an administrator can take when setting up a recursive server for an organization. Because it is unwise to allow queries from the internet to the recursive server, the recursive server can be placed behind a firewall that blocks inbound connections to UDP and TCP port 53 (used by DNS).

Recursive servers should be configured to only accept queries from internal hosts and perform recursion for them. Different recursive DNS server implementations may have features to

enable this ability. This could be done by implementation-specific ACLs or network infrastructure configuration, which is beyond the scope of this document.

4.3.1. Configuring an Authoritative Source for Time

DNSSEC validation and encrypted DNS certificate validation requires knowledge of the current time. Recursive servers that provide DNSSEC validation, DoH, DoT, or DoQ need to have an authoritative source for time. Servers should be configured to have a primary and backup set of time sources and a secure means to contact time servers (e.g., using NTS [32]). How this is configured depends on the implementation.

5. Managing Threats to Stub Resolvers

Software that requires access to the internet or internal network resources (e.g., web browsers, email clients, cloud applications, operating systems) use a DNS client called the client resolver, resolver library, or stub resolver. A stub resolver is often referred to as simply a client. The stub resolver formulates a name resolution query for the resource sought by the network-accessing software and sends it to a recursive name server in the enterprise (see Appendix A). Stub resolvers are generally configured to send queries to two or more recursive name servers to provide some fault tolerance for their operation.

The OS-layer stub resolver is a centralized DNS client within the operating system that handles DNS queries for all applications by sending them to a recursive resolver configured at the system level (e.g., via network settings). In contrast, an application-layer stub resolver operates independently within a specific application and bypasses the OS resolver to handle DNS queries directly. This is common in modern browsers or applications that support DoH or DoT for enhanced privacy and security. These two resolvers often interact when an application-layer stub overrides the system's default resolver settings, which can create conflicts or complementarities, depending on how they are configured. For instance, an application might use its own DNS settings for specific queries while still relying on the OS stub for others, creating a layered approach to DNS resolution.

5.1. Securing the Stub Resolver

Stub resolvers do not have many configuration options. Often, the only configuration necessary for an administrator is to enter the IP addresses of one or more recursive resolvers that the stub would rely on to resolve queries. It is a good idea for administrators to include the IP addresses of at least two recursive servers to increase the availability of the DNS service for end users. This can be done manually or using a network management protocol (e.g., DHCP, DHCPv6). Additionally, an administrator can lower the risk of client impact from the loss of any one DNS server by adding a layer of abstraction between the IP address of the DNS service and the individual recursive servers via anycast. This can be especially important for supporting certain applications that will only use one DNS server, regardless of the stub resolver configuration.

There is a known class of malware that attempts to change the settings on a system's stub resolver to direct queries to another (usually malicious) recursive server. The server may then direct end users to a malicious site where another attack takes place, or the server may simply direct users to a web page that serves ads or similar non-intended content. To combat this, administrators should consider blocking all outbound DNS traffic with the exception of known recursive servers or use other security tools to maintain secure endpoint configurations.

Implementing enterprise mobility management (EMM) software can ensure that stub resolvers are correctly configured to point to authorized DNS servers. Additionally, EMM facilitates the management and enforcement of policies regarding the approval and use of software with integrated stub resolvers within the enterprise environment.

5.2. DNSSEC Considerations for Stub Resolvers

Most stub resolvers cannot perform DNSSEC validation and may not understand DNSSEC at all. These stub resolvers will have to rely on an upstream validating recursive resolver to perform DNSSEC validation on their behalf. This does not pose a significant risk on a trusted organization's network because there are several options to protect the link between a validating recursive resolver and a stub resolver that cannot do DNSSEC validation processing.

If the link between the stub and its recursive service is secured (e.g., using one of the last hop mechanisms described earlier in this document), then the stub resolvers can be considered to be "using" DNSSEC. However, network administrators should be aware that DNSSEC validation failures could complicate the diagnosis of internet error messages. DNSSEC validation failures will be seen by the upstream validator, not the stub resolver that initiated the query (which currently receives a general error response). Network administrators should check validator logs when responding to resolution errors to rule out DNSSEC validation failures.

5.3. Recommendations for Providing Service to Mobile Hosts

Mobile or nomadic hosts present a particular challenge for network administrators. These systems often access a network outside of the trusted enterprise, so mobile hosts must either perform their own validation or have a trusted connection to an enterprise-approved DNS service. If the mobile hosts can perform their own validation, then the same policy for the enterprise validators should be applied for the mobile host. That is, the same trust anchors and validation policy should be set for mobile hosts as for validators on the enterprise network.

If a mobile system is able to perform its own DNSSEC validation, it may be necessary to have different configurations for its validator when migrating from the enterprise to an external network and vice versa. If the enterprise network is trusted, the mobile host can rely on the enterprise validator when on the enterprise network and perform its own validation when on external networks. Ideally, network administrators can avoid this problem by using different names for internal and external zones, thus having different trust anchors (if the internal zone is DNSSEC-signed).

If the mobile host cannot perform its own validation, it must either have a secure tunnel back to the enterprise network or a secure connection to an approved recursive DNS service. Many enterprises already have a means for mobile hosts to access internal resources (e.g., file servers), so a validating recursive server should be added as one of the services provided to mobile hosts through a secure channel. Alternatively, enterprise mobile hosts could be configured to use an approved cloud-based DNS recursive service to allow mobile hosts to use approved DNS recursive services without needing to have a connection to the enterprise infrastructure.

References

[1] Mockapetris P (1987) Domain Names - Concepts and Facilities. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 1034. https://doi.org/10.17487/RFC1034

[2] Mockapetris P (1987) Domain Names - Implementation and Specification. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 1035. https://doi.org/10.17487/RFC1035

[3] Rose S, Borchert O, Mitchell S, Connelly S (2020) Zero Trust Architecture. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-207. https://doi.org/10.6028/NIST.SP.800-207

[4] Cybersecurity and Infrastructure Security Agency (2025) Protective Domain Name System (DNS) Resolver. Available at: https://www.cisa.gov/resources-tools/services/protective-domain-name-system-dns-resolver

[5] dnstap community (2025) dnstap. Available at https://dnstap.info

[6] Atkins D, Austein R (2004) Threat Analysis of the Domain Name System (DNS). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 3833. https://doi.org/10.17487/RFC3833

[7] Turner S (2011) Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 6151. https://doi.org/10.17487/RFC6151

[8] National Institute of Standards and Technology (2008) The Keyed-Hash Message Authentication Code (HMAC) (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) 198-1, Change Notice 1 July 2008. https://doi.org/10.6028/NIST.FIPS.198-1

[9] Dupont F, Morris S, Vixie P, Eastlake D, Guðmundsson Ó, Wellington B (2020) Secret Key Transaction Authentication for DNS (TSIG). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8945. https://doi.org/10.17487/RFC8945

[10] Lewis E, Hoenes A (2010) DNS Zone Transfer Protocol (AXFR). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 5936. https://doi.org/10.17487/RFC5936

[11] Ohta M (1996) Incremental Zone Transfer in DNS. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 1995. https://doi.org/10.17487/RFC1995

[12] Vixie P, Thomson S, Rekhter Y, Bound J (1997) Dynamic Updates in the Domain Name System (DNS UPDATE). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 2136. https://doi.org/10.17487/RFC2136

[13] Wessels D, Barber P, Weinberg M, Kumari W, Hardaker W (2021) Message Digest for DNS Zones. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8976. https://doi.org/10.17487/RFC8976

[14] Eastlake D (2000) DNS Request and Transaction Signatures (SIG(0)s). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 2931. https://doi.org/10.17487/RFC2931

[15] Toorop W, Dickinson S, Sahib S, Aras P, Mankin A (2021) DNS Zone Transfer over TLS. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 9103. https://doi.org/10.17487/RFC9103

[16] Rose S, Nightingale JS, Garfinkel S, Chandramouli R (2019) Trustworthy Email. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-177r1. https://doi.org/10.6028/NIST.SP.800-177r1

[17] Lawrence D, Kumari W, Sood P (2020) Serving Stale Data to Improve DNS Resiliency. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8767. https://doi.org/10.17487/RFC8767

[18] Moura GCM, Heidemann J, Schmidt RdO, Hardaker W (2019) Cache Me If You Can: Effects of DNS Time-to-Live. *IMC '19: Proceedings of the Internet Measurement Conference* (ACM, Amsterdam, Netherlands), pp 101–115. https://doi.org/10.1145/3355369.3355568

[19] Hoffman P (2023) DNS Security Extensions (DNSSEC). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 9364. https://doi.org/10.17487/RFC9364

[20] Kolkman O, Mekking M, Gieben RM (2012) DNSSEC Operational Practices, Version 2 (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 6781. https://doi.org/10.17487/RFC6781

[21] Wouters P, Surý O (2019) Algorithm Implementation Requirements and Usage Guidance for DNSSEC. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8624. https://doi.org/10.17487/RFC8624

[22] Barker E (2020) Recommendation for Key Management: Part 1 – General. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-57pt1r5. https://doi.org/10.6028/NIST.SP.800-57pt1r5

[23] Barker E, Roginsky A (2019) Transitioning the Use of Cryptographic Algorithms and Key Lengths. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-131Ar2. https://doi.org/10.6028/NIST.SP.800-131Ar2

[24] National Institute of Standards and Technology (2025) Cryptographic Module Validation Program (CMVP). Available at https://csrc.nist.gov/projects/cryptographic-module-validation-program

[25] Roetteler M, Naehrig M, Svore KM, Lauter K (2017) Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms. *arXiv preprint*. https://doi.org/10.48550/arXiv.1706.06752

[26] Kumari W, Guðmundsson Ó, Barwood G (2014) Automating DNSSEC Delegation Trust Maintenance. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 7344. https://doi.org/10.17487/RFC7344

[27] Guðmundsson Ó, Wouters P (2017) Managing DS Records from the Parent via CDS/CDNSKEY. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8078. https://doi.org/10.17487/RFC8078

[28] Arends R, Sisson G, Blacka D, Laurie B (2008) DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 5155. https://doi.org/10.17487/RFC5155.

[29] Rose S, Nakassis A (2008) Minimizing Information Leakage in the DNS. *IEEE Network*, vol. 22, no. 2, pp 22-25. https://doi.org/10.1109/MNET.2008.4476067

[30] Hardaker W, Dukhovni V (2022) Guidance for NSEC3 Parameter Settings. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 9276. https://doi.org/10.17487/RFC9276

[31] Huque S, Elmerot C, Guðmundsson Ó (2025) Compact Denial of Existence in DNSSEC. Internet Draft Draft-Ietf-Dnsop-Compact-Denial-Of-Existence-07. Internet Engineering Task Force, 2025. Available at https://datatracker.ietf.org/doc/draft-ietf-dnsop-compact-denial-of-existence

[32] Franke DF, Sibold D, Teichel K, Dansarie M, Sundblad R (2020) Network Time Security for the Network Time Protocol. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8915. https://doi.org/10.17487/RFC8915

[33] Hoffman P, Fujiwara K (2024) DNS Terminology(Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 9499. https://doi.org/10.17487/RFC9499

[34] Lewis E (2006) The Role of Wildcards in the Domain Name System(Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 4592. https://doi.org/10.17487/RFC4592

[35] Hu Z, Zhu L, Heidemann J, Mankin A, Wessels D, Hoffman P (2016) Specification for DNS over Transport Layer Security (TLS). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 7858. https://doi.org/10.17487/RFC7858

[36] Hoffman P, McManus P (2018) DNS Queries over HTTPS (DoH). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8484. https://doi.org/10.17487/RFC8484

[37] Huitema C, Dickinson S, Mankin A (2022) DNS over Dedicated QUIC Connections. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 9250. https://doi.org/10.17487/RFC9250

[38] McKay K, Cooper D (2019) Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-52r2. https://doi.org/10.6028/NIST.SP.800-52r2

[39] Bortzmeyer S (2016) DNS Query Name Minimisation to Improve Privacy. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 7816. https://doi.org/10.17487/RFC7816

[40] StJohns M (2007) Automated Updates of DNS Security (DNSSEC) Trust Anchors. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 5011. https://doi.org/10.17487/RFC5011

[41] Internet Engineering Task Force (2025) RFC Editor. Available at https://www.rfc-editor.org/search/rfc_search_detail.php?title=DNS&page=All

Appendix A. DNS Protocol Tutorial

The term DNS can refer to the global namespace, the infrastructure used to operate that namespace, and the protocol used by hosts to resolve domain names to IP addresses or other information. This appendix presents a high-level overview of how a DNS operates.

A.1. DNS Namespace and Infrastructure

The DNS infrastructure is made up of computing and communication entities that are geographically distributed throughout the world. The domain name space is organized in a hierarchy. The topmost level in the hierarchy is the *root domain*, which is represented as a dot (“.”). The next level in the hierarchy is called the *top-level domain* (TLD). There is only one root domain, but there are many TLDs. Each TLD is called a child domain of the root domain. In this context, the root domain is the parent domain because it is one level above a TLD. Each TLD, in turn, can have many child domains. The children of TLDs are often called *second-level* or *enterprise-level domains*.

In a domain name representation, the symbol for the root domain is usually omitted. For example, consider the domain name marketing.example.com. The rightmost label in this domain name (“com.”) is a TLD. The next label to the left (“example”) is the second-level or enterprise-level domain. The leftmost label (“marketing”) is the third-level domain. It is also possible to have a fourth-level domain, fifth-level domain, and so on. Because each of the labels in marketing.example.com is called a domain (e.g., TLD, second-level domain, third-level domain), the concatenation of all of these labels from the current level to the root zone is a *fully qualified domain name* (FQDN). In this document, however, the FQDN is simply referred to as a domain name, and the level name is used to identify individual labels.

There are over a thousand TLDs categorized into the following three types:

- **Country-code TLDs (ccTLDs) —** Domains associated with countries and territories. There are more than 240 ccTLDs. Examples include .uk, .in, and .jp.
- **Sponsored generic TLDs (gTLDs) —** Specialized domains with a sponsor that represents a community of interest. These TLDs include .edu, .gov, .int, .mil, .aero, .coop, and .museum.
- **Unsponsored generic TLDs (gTLDs) —** Domains without a sponsoring organization. The list of unsponsored gTLDs includes .com, .net, .org, .biz, .info, .name, and .pro.

Figure 3 shows a partial DNS namespace hierarchy.

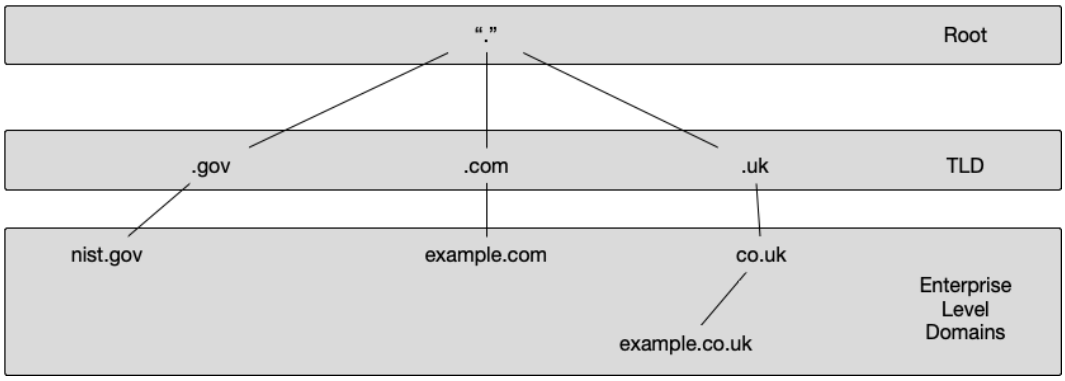

Fig. 3. DNS tree

A user (i.e., an individual or corporation) wishing to register a domain name must contact an authorized entity called a registrar (which may charge a fee, depending on the TLD in question). *Registrars* are companies that are authorized to register domain names in a particular TLD or sub-domain of a TLD (e.g., co.uk.) to end users. When the registrar receives the user's registration request, the registrar verifies that the name is available by checking with the registry that manages the corresponding TLD (or sub-domain under the TLD). If the name is available, the registrar registers the name with the appropriate registry. The registry then adds the new name to its registry database and publishes the new name in DNS. In some domains (e.g., country codes, gTLDs), the same organization acts as the registry and registrar for names in the domain.

Organizations that register and obtain an enterprise-level domain can then create child domains to properly identify internet resources associated with various functional units. For example, the owner of the domain name example.com might create the subdomain "shipping" to create and identify resources associated with the shipping department of the organization.

To facilitate this grouping, the DNS defines the concept of a zone. A zone may be an entire domain or a domain with one or more subdomains. It is a configurable entity within a name server under which information on all internet resources pertaining to a domain and a selected set of subdomains is described. Thus, zones are the administrative building blocks of the DNS namespace just as domains are the structural building blocks. As a result, the term *zone* commonly refers to a domain that is managed as a stand-alone administrative entity (e.g., the root zone, the .com zone).

A.2. DNS Queries and Responses

When a user types the URL www.example.com into a web browser, the browser program contacts a type of resolver called a *stub resolver* that then contacts a local recursive name server, as shown in Fig. 4.

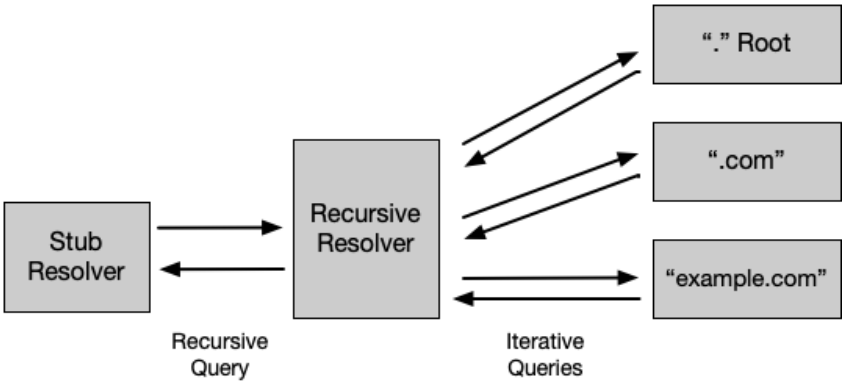

Fig. 4. DNS resolution

The resolving name server checks its cache to determine whether it has valid information to provide the IP address for the internet resource (e.g., www.marketing.example.com). If not, the resolving name server checks the cache to determine whether it has information regarding the name server for the zone marketing.example.com, since this is the zone that is expected to contain the resource. If the name server's IP address is in the cache, the resolver's next query will be directed to that name server or a name server higher up in the hierarchy.

If a complete search of the cache does not yield the required information, the resolving name server must start its search by querying the name server in the topmost zone in the DNS namespace hierarchy (i.e., the root server). Contact with the root servers is enabled by the "root hints" file that is usually present in every name server in DNS. The root server will contain information about the name servers for its child zones (i.e., TLDs). A TLD (e.g., .com) will contain name server information about its child zones (e.g., example.com). The name server information about its child zones that is carried in a zone is called *delegation information*. The delegation information is used by a zone to refer name resolution requests for a resource that is lower than it in the domain name hierarchy. Since the name resolution request in this example pertains to a resource in the third-level domain, the root server must refer the request to a lower-level name server.

The response to the resolving name server that involves sending this delegation information is called the *referral*. The referral provides the name and IP address for the name servers for the TLD zone that is relevant to the request (i.e., the .com zone). Using this referral, the resolving name server then formulates and sends a query to the .com zone name server. This server will provide the referral for example.com's name server. If the marketing.example.com domain is included in example.com's zone, querying the name server for example.com will provide the IP address for the resource marketing.example.com.

As the description of the name resolution process makes clear, a name server provides:

- A referral to a child zone
- A mapping from the domain name to the IP address (i.e., domain name resolution) or vice versa (i.e., inverse resolution)
- An error message if the query is for a DNS entry that does not exist

The name server performs these three functions with the same DNS database, which is called a *zone file*. The mapping function is performed by a class of information in a zone file called *authoritative information* and is provided by a set of records that list the resources in that zone, its domain name, and its corresponding IP address. Because the resources belong to that zone, the information provided is deemed authoritative. Thus, a zone file contains two categories of information: authoritative information (i.e., information about all resources for all domains in the zone) and delegation information (i.e., information about name servers for child zones). The locations in the zone file where delegation information appears are called *delegation points*. The level of a zone file is the level of the topmost domain for which it contains authoritative information. In the previous example, the zone file in the name server of example.com is the enterprise-level zone file, and the corresponding name server is called the enterprise-level name server.

To fully understand the DNS protocol, readers are urged to consult the DNS protocol specifications maintained by the IETF [41].

Appendix B. Glossary

The general terms related to DNS are collected and defined by the IETF in RFC 9499 [33]. Terms that do not appear there or require additional context include the following.

dangling CNAME
A CNAME record maps an alias to a canonical name, which in turn can be resolved to an IP address via a record of the canonical name. If the record for the canonical name is removed, the CNAME is left "dangling" because what it refers to is no longer present in the DNS. If the CNAME refers to a canonical name outside of the organization's control, a malicious actor may gain administrative control of the canonical name (e.g., if the registration of the canonical domain has expired and potentially appears as part of the organization owning the CNAME).

DNS hygiene
The practice of maintaining a secure, resilient, and well-configured DNS environment. This includes proactively managing and monitoring DNS configurations to eliminate vulnerabilities (e.g., lame delegations, dangling CNAME records). It also involves the removal of any RRsets that are no longer in use. Additionally, DNS hygiene encompasses the continuous monitoring and mitigation of look-alike domains to protect against phishing, impersonation, and other malicious activities.

encrypted DNS
A collective term for the use of DNS over TLS (DoT), DNS over HTTPS (DoH), DNS over QUIC (DoQ), and other means to encrypt communication between stub resolvers and recursive DNS servers.

forged responses
A response that is different from the one that is expected from a legitimate authoritative name server. A bogus response can originate from a compromised authoritative name server or a poisoned cache of a recursive name server.

lame delegation
Delegating a zone to any name server that is not authoritative for that zone. This can happen if the subdomain expires and is not removed from the parent domain, or name servers for the delegated domain change but the changes are not synchronized with the parent domain. [33]

look-alike domains
Domain names that are created for malicious purposes or to visually appear as a trusted domain name to end users. They visually look like a legitimate domain name but may include characters that are substituted from a different alphabet script.

protective DNS
A DNS service that is enhanced with security capabilities to analyze DNS queries and responses and take action to mitigate threats.

typosquat
A domain that is close to a valid domain but contains a misspelling or some other error when typing in the domain. It may be registered by an attacker who hopes to direct a user to a malicious site (e.g., register "xample.com" and "exampel.com" in an attempt to redirect users who mean to enter "example.com").