NIST Special Publication 800
NIST SP 800-81r3 ipd

Secure Domain Name System (DNS) Deployment Guide

Initial Public Draft

Scott Rose
Cricket Liu
Ross Gibson

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

NIST Special Publication 800
NIST SP 800-81r3 ipd

Secure Domain Name System (DNS) Deployment Guide

Initial Public Draft

Scott Rose
Wireless Networks Division
Communications Technology Laboratory

Cricket Liu
Ross Gibson
Infoblox Inc.

April 2025

U.S. Department of Commerce
Howard Lutnick, Secretary

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

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

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

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

1 **Abstract**

2 This document provides Domain Name System (DNS) deployment guidelines to secure the DNS
3 protocol and infrastructure, mitigate misuse or misconfiguration, and provide an additional
4 layer of network security as part of a zero trust and/or defense-in-depth security risk
5 management approach. This introduction briefly discusses relevant context for DNS and
6 examines the changing threat landscape that has warranted an updated approach to DNS
7 deployment.

8 **Keywords**

9 Authoritative Name Server; DNS Logging; DNS Security Extensions (DNSSEC); Domain Name
10 System (DNS); Encrypted DNS; Protective DNS; Recursive Name Server; Resource Record (RR).

11

12 **Call for Patent Claims**

13 This public review includes a call for information on essential patent claims (claims whose use
14 would be required for compliance with the guidance or requirements in this NIST draft
15 publication). Such guidance and/or requirements may be directly stated in this NIST Publication
16 or by reference to another publication. This call also includes disclosure, where known, of the
17 existence of pending U.S. or foreign patent applications relating to this NIST draft publication
18 and of any relevant unexpired U.S. or foreign patents.

19 NIST may require from the patent holder, or a party authorized to make assurances on its
20 behalf, in written or electronic form, either:

21 a) assurance in the form of a general disclaimer to the effect that such party does not hold
22   and does not currently intend holding any essential patent claim(s); or

23 b) assurance that a license to such essential patent claim(s) will be made available to
24   applicants desiring to utilize the license for the purpose of complying with the guidance
25   or requirements in this NIST draft publication either:

26   i. under reasonable terms and conditions that are demonstrably free of any unfair
27   discrimination; or

28   ii. without compensation and under reasonable terms and conditions that are
29   demonstrably free of any unfair discrimination.

30 Such assurance shall indicate that the patent holder (or third party authorized to make
31 assurances on its behalf) will include in any documents transferring ownership of patents
32 subject to the assurance, provisions sufficient to ensure that the commitments in the assurance
33 are binding on the transferee, and that the transferee will similarly include appropriate
34 provisions in the event of future transfers with the goal of binding each successor-in-interest.

35 The assurance shall also indicate that it is intended to be binding on successors-in-interest
36 regardless of whether such provisions are included in the relevant transfer documents.

37 Such statements should be addressed to: SP800-81@nist.gov

38 **Table of Contents**

107 **List of Tables**

110 **List of Figures**

115

116

117 **Preface**

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

122 **Acknowledgments**

126 **Executive Summary**

127 The Domain Name System (DNS) [1][2] is a standardized way of translating machine-readable IP
128 addresses (e.g., 129.6.13.49) to human-readable ones (e.g., nist.gov). It is commonly deployed
129 within an organization's networks to facilitate the internal functions of those intranets. DNS is
130 also deployed and maintained across critical internet infrastructure at a high level to enable the
131 core functionalities of the internet on almost every network at every scale. Its centralized
132 position enables it to act as a foundational layer of network security in zero trust and defense-
133 in-depth security risk management approaches. Such DNS services are often referred to as
134 *Protective DNS*[1] deployments and are a key consideration in securing organizational networks.

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

138   • Employ Protective DNS wherever technically feasible to provide additional network-
139   wide security capabilities that include:

140   o Blocking harmful or malicious traffic in real time

141   o Filtering out categories of traffic that do not conform to the organization's
142   policies

143   o Generating real-time and historical DNS query and response data to facilitate
144   digital forensics and incident response

145   o Integrating with the wider security ecosystem as part of a defense-in-depth or
146   zero trust approach

147   o Facilitating the organization's responsibility to comply with regulatory or
148   contractual requirements for blocking traffic to disallowed sites (e.g., copyright
149   violations, legal restrictions)

150   • Encrypt internal and external DNS traffic wherever feasible

151   • Deploy dedicated DNS servers to reduce attack surfaces

152   • Follow all technical guidance on ensuring that DNS deployments and the DNS protocol
153   are as secure and resilient as possible

154

---

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

155 **1. Introduction**

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

161 **1.1. Domain Name Systems**

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

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

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

187 **1.2. Impact of DNS on Cyber Resiliency, Defense-in-Depth, and Zero Trust**

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

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

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

203 Zero trust presents a shift from a location-centric model to one that focuses on identity,
204 context, and data with fine-grained security controls between users, systems, applications,
205 data, and assets that change over time. As DNS resolution may be open to end points querying
206 any public domain names, organizations should secure DNS to prevent it from being used as a
207 route to bypass zero trust network controls. This could involve the implementation of
208 Protective DNS services (see Sec. 2.1.2) and/or specific DNS resolver configurations for certain
209 classes of end points.

210 **1.2.1. DNS Use Cases for Operational Technology, Internet of Things Devices, and Critical**
211 **Infrastructure**

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

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

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

225 • Taking advantage of the capabilities of Protective DNS (see Sec. 2.1.2)

226 • Implementing highly restricted response policy zones (RPZs) so that communication
227 from OT/IoT devices is restricted to the systems needed for their secure operation

228 **1.3. Using This Guide**

229 This document provides DNS deployment guidelines that enable the use of DNS as a security
230 control (i.e., Protective DNS) for organizations and users to secure the DNS protocol,
231 infrastructure, and services. It is organized as follows:

232 • Section 2 discusses using DNS as a component of an organization's security and provides
233 recommendations on how to best accomplish that goal.

234 • Section 3 discusses the threats to and recommendations for securing authoritative
235 services.

236 • Section 4 discusses the threats to and recommendations for securing recursive services
237 and stub resolvers.

238 **1.4. Audience**

239 This document is intended for two general groups:

240 1. Cybersecurity executives, decision-makers, and organizational policy-setters

241 1. Operational networking and cybersecurity teams

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

246

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

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

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

260 According to the Cybersecurity & Infrastructure Security Agency (CISA), "DNS infrastructure is a
261 common threat vector for attack campaigns" [30]. Even when DNS traffic is encrypted, the
262 system still needs to retain access to the internet. This allows malicious actors to set up
263 authoritative servers for command and control (C2) and data exfiltration, where encryption can
264 work to their advantage. Therefore, it is crucial to implement strict controls and auditing on an
265 organization's secure resolvers to ensure that only resolvers with the appropriate policy
266 configuration are permitted to communicate with the internet. Applying security in DNS
267 infrastructure gives administrators the opportunity to review potentially malicious
268 communications before they begin and automatically prevent them from happening.

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

278 **2.1. Protective DNS**

279 Protective DNS[2] is enhanced with security capabilities to analyze DNS queries and responses
280 and take action to mitigate threats. Protective DNS blocks access to malicious websites and
281 prevents the delivery of malware, ransomware, phishing, and other attacks that attempt to
282 deliver spyware and viruses. Protective DNS can be provided as a service from a vendor,
283 deployed on internal DNS infrastructure, or a combination of the two. There are potential
284 benefits to using a combination of externally provided Protective DNS with internally deployed

---

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

285 Protective DNS. While this approach may not be applicable in all cases, this combined hybrid
286 scheme should be utilized where feasible.

287 The goals of deploying Protective DNS include:

- Blocking or redirecting harmful traffic in real time at the point of domain name
  resolution, typically before malicious activity starts

- 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

- Delivering visibility into 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 defense in depth, such as
  correlating an organization's data on assets (e.g., devices, cloud workloads) and users
  with the IP addresses of blocked queries

- 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

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

307 This approach requires integrating threat intelligence into the DNS resolver. Threat intelligence
308 is leveraged in a DNS infrastructure via mechanisms such as RPZs and can be seamlessly
309 integrated into the DNS resolution chain via different architectures. Therefore, the
310 consumption and deployment of threat intelligence services should be considered as part of
311 any Protective DNS deployment.

2.1.2. Name Resolution Filtering

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

321   Protective DNS is generally implemented by using RPZs configured on an on-premises DNS
322   service; a cloud-based, secure recursive DNS service; or some combination of the two. Figure 1
323   shows an enterprise that utilizes a cloud-based service but retains a local forwarder for hosts on
324   the enterprise local network.

325

326
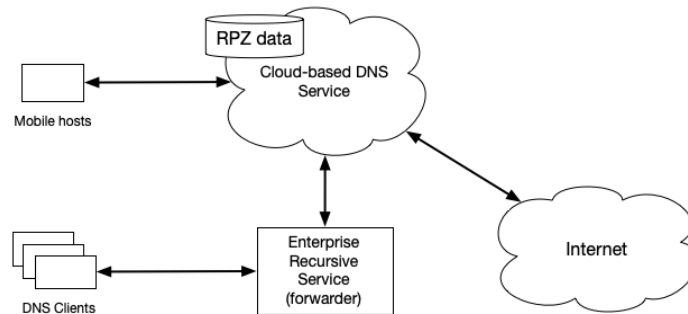
327   **Fig. 1. Enterprise using cloud-based Protective DNS with a local forwarder**

328   RPZs allow administrators to specify security policies that are enforced by the DNS service. For
329   example, these policies could filter out specific domains or IP addresses. Many internet security
330   organizations also provide "feeds" of their current threat data as RPZs. Organizations can apply
331   those feeds to configure their name servers as secondary name servers for those RPZs and
332   automatically synchronize their DNS resolution policy with the latest threat information. With
333   an on-premises deployment of DNS, RPZs provide local control over these policies with very low
334   latency, increased survivability, and minimal performance impact. With the ubiquity of DNS
335   infrastructure and properties of RPZ, DNS is able to consistently mitigate threats across entire
336   network infrastructures in a matter of minutes.

337   Secure recursive DNS services are also available on the internet. Organizations can configure
338   their name servers to forward queries for internet domain names to the IP addresses operated
339   by these services. The services generally offer a web-based control panel that allows
340   administrators to customize the resolution policy applied to queries from the organization's
341   clients. The cloud approach offers greater scalability, storage, and computing power but has
342   disadvantages, such as a loss of confidentiality, higher latency, and challenges in quickly and
343   accurately attributing DNS queries to their sources. Organizations may consider using a
344   combination of these to benefit from the lower latency of on-premises services and the greater
345   scalability of cloud services while allowing for granular tracking, logging, and attribution of
346   connections and requests. The best choice for a given DNS deployment will vary depending on
347   the specific needs of the network and its users.

348   **2.1.3. DNS for Digital Forensics and Incident Response**

349   Government agencies and regulated enterprises should implement robust DNS traffic logging
350   mechanisms to meet compliance requirements. Logging should capture both current and
351   historical DNS traffic to enable digital forensics and incident response. These DNS logs should

352   be integrated with other system logs to facilitate correlation with cloud workloads and device
353   or user activities and to enhance visibility and auditability.

354   Logging all DNS traffic can be resource-intensive. If not done efficiently, it may impact the
355   performance and availability of an organization's DNS services. If full DNS traffic logging is
356   determined to be too resource-intensive, organizations may consider using cloud-based
357   solutions, efficient logging methods (e.g., DNSTAP format [4]), or selective logging. However,
358   DNS queries and the responses associated with domains that are classified as malicious or
359   unauthorized by Protective DNS services should always be logged to support security and
360   compliance objectives. An organization may remove known secure domains from logs to reduce
361   volume and operating costs before sending them to a security information and event
362   management (SIEM) system but should keep a complete log for future forensics.

363   Mapping an IP address to a compromised asset in the event of a cyber attack requires tracking
364   key attributable metadata in real-time as well as a history of its allocation to each asset and
365   resource, such as a Dynamic Host Configuration Protocol (DHCP) lease history. To ensure rapid
366   notification of queries that might indicate infection or malicious activity, organizations should
367   integrate Protective DNS logs from their name servers or their secure recursive DNS service
368   with their SIEM or log analysis platform.

369   **2.2. Protecting the DNS Protocol**

370   DNS is a fundamental network service and must be left open to enable internet connections. As
371   a result, it has been used by threat actors as a strategic vehicle to send malware and conduct
372   data exfiltration, C2, and other attacks. The Internet Engineering Task Force (IETF) published an
373   early threat model of DNS [5] that primarily focused on its query/response role. However, there
374   are other considerations that require additional protective measures.

375   If a DNS server is compromised, there is little limit to the amount of short- or long-term damage
376   that can be inflicted, often while avoiding detection. To that end, it is crucial to prevent bad
377   actors from using DNS as a threat vector. There are two equally important elements to
378   accomplishing this:

379   1. Protecting internal and external authoritative and recursive DNS services against threats

380   2. Usage Encrypted DNS and authentication to protect privacy and confidentiality

381   **2.2.1. Protecting the Integrity of DNS Services**

382   The DNS protocol refers to the standardized communications that carry DNS information
383   between networked entities. Threats to the DNS protocol are numerous but well-studied.
384   Section 3 discusses specific threats and protection approaches.

385   **2.2.2. Using Encrypted DNS and Authentication to Protect the Protocol**

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

388 The DNS Security Extensions (DNSSEC) is a standardized set of extensions to DNS for securing
389 DNS protocol communications against compromise. It is one part of the wider field of DNS
390 security and must be implemented alongside other best practices and protocol features. While
391 DNSSEC can help protect against the compromise of DNS communications, it does not provide
392 any privacy protection (e.g., encryption). Those capabilities are provided by other technologies,
393 such as DNS over Transport Layer Security (TLS) (DoT), DNS over Secure Hypertext Transfer
394 Protocol (HTTPS) (DoH), and DNS over Quick UDP Internet Connections (QUIC) (DoQ) (see Sec.
395 4.2.1).

396 The process of authenticating the source of a message and its integrity through hash-based
397 message authentication codes (HMAC) is specified through a set of DNS specifications known
398 collectively as TSIG. The term "HMAC" denotes both the message authentication code
399 generated by using a keyed hash function and the hash function itself. HMAC is specified in
400 Request for Comment (RFC) 6151 [6] and generalized in the NIST document Federal Information
401 Processing Standards Publications (FIPS PUBS) 198-1 [7]. The HMAC function for TSIG specified
402 in RFC 8945 [8].

403 Later sections of this document categorize deployment guidance for securing the DNS protocol
404 by type (i.e., authoritative in Section 3.8, recursive in Section 4.2.4, and stub resolvers in
405 Section 5.1), including the use of DNSSEC and TSIG.

406 **2.2.3. DNS Hygiene and Best Practices**

407 Threat actors can exploit misconfiguration and lapsed domain/DNS resolver registration to
408 seriously compromise DNS integrity. Organizations should implement robust processes to
409 continuously monitor and validate the integrity of their public domains and take steps to raise
410 the visibility of attempts to impersonate domains owned by the organization. Section 3
411 provides an in-depth breakdown of threats to DNS Hygiene and a discussion of best protection
412 approaches.

413 **2.3. Protecting the DNS Service and Infrastructure**

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

420 • **Platform or component vulnerability:** The OS, any system software, or any other
421 application software on the DNS host could be vulnerable to attacks that result in the
422 denial of name resolution services or threats to the hypervisor or underlying cloud
423 infrastructure if the DNS service is cloud-based.

424 • **Distributed denial of service:** A malicious attacker could send a large volume of queries
425 to perform a denial-of-service (DoS) attack against a server or service. The attacker

426   could also use numerous third-party DNS servers to aid in the attack (i.e., distributed
427   DoS or DDoS).

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

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

440   ### 2.3.1. Dedicated DNS Services

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

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

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

---

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

462 A recursive name server should be run under a different security policy than that of an
463 authoritative name server to mitigate attacks (e.g., cache poisoning). A name server instance
464 that is directly accessible from or has direct access to the internet should be configured as
465 either an authoritative name server or a recursive name server. Name servers that are only
466 accessible from inside an organization's network commonly perform both functions — hosting
467 internal DNS zone data and providing recursion for clients. In that context, combining the
468 functions on the same server is acceptable and practical.

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

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

2.3.2. Resiliency and High Availability of DNS Servers

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

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

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

502 2.3.3. Interoperability of the Protective DNS Ecosystem

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

505 • Ensuring that the DNS service is part of a "strength in depth" approach and not an
506   isolated control

507 • Logging query/response data and blocks/redirects to security operations functions (e.g.,
508   via a SIEM/SOAR) to be correlated with other security events

509 • Being able to access the threat intelligence deployed within the Protective DNS service
510   via APIs for use in assessments, forensics, and incident response

511 • Ensuring that DNS components use standardized and interoperable protocols and APIs
512   as defined by the relevant IETF RFCs

513 • Sharing and forwarding relevant DNS data to other components of the wider security
514   ecosystem

515 **3. Managing Threats to Authoritative Services**

516 The primary role of an authoritative DNS server is to provide answers to queries for the zones
517 for which it is authoritative. These servers contain files known as zone files, which in turn
518 contain information, such as name-to-address mappings (i.e., A RRs for IPv4 IP addresses or
519 AAAA RRs for IPv6 addresses) or address-to-name mappings (i.e., PTR records).

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

534 When the authoritative role is provided by a cloud-based service, there is often no
535 differentiation between primary and secondary roles. Cloud-based DNS authoritative hosting
536 services may use means other than zone transfers to synchronize DNS data across cloud
537 instances. From a client perspective, there is no difference between a primary or secondary role
538 providing answers to a query since they both provide authoritative answers for the zone. The
539 primary role is only relevant to clients when they are attempting to update DNS records via DNS
540 UPDATE (sometimes called *nsupdate)* [11], which must be sent to the primary for the zone.

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

548 **3.1. Zone Transfer Threats and Protection Approaches**

549 Zone transfers replicate zone files to multiple servers to provide a degree of fault tolerance in
550 the DNS service provided by an organization. While threats from zone transfers have not been
551 formally documented through any IETF RFCs, a common threat to any network transaction is

552 the denial of service. A second common threat to any network packet is based on the
553 exploitation of knowledge gained from the information provided by zone transfers.

554 • Denial of service (DoS): Because zone transfers involve the transfer of entire zones, they
555   place substantial demands on network resources relative to normal DNS queries. Errant
556   or malicious frequent zone transfer requests on name servers can overload a primary or
557   secondary zone server and result in the denial of service to legitimate users.

558 • Unauthorized zone modification: The zone transfer response message could be
559   tampered with.

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

564 The IETF developed an alternate mechanism called a transaction signature (TSIG) [8], whereby
565 the mutual identification of servers is based on a shared secret key. Because the number of
566 servers involved in zone transfer is generally restricted to name servers in the same
567 administrative domain of an organization, a bilateral trust model that is based on a shared
568 secret key may be adequate for most organizations. TSIG specifies that the shared secret key be
569 used for both mutual authentication and for signing zone transfer requests and responses to
570 protect against tampering.

571 Asymmetric cryptography (i.e., public-key cryptography) can also be used to authenticate DNS
572 transactions. The format of the SIG(0) RR [12] is similar to the resource record signature (RRSIG)
573 RR and can be validated using a public key stored in the DNS instead of a shared secret key.
574 While SIG(0) can be more computationally expensive to use, a previous trust relationship may
575 not be necessary to use SIG(0) signed messages. However, because most zone transfers occur
576 between parties that have a previously established relationship, it is considered easier to
577 implement TSIG for authenticating zone transfer transactions.

578 A lower-level network layer solution (e.g., IPSec or other secure network communication
579 technologies) can also provide security and remove the need for authentication at the
580 application layer. The protocol for DNS zone transfer over TLS ensures that zone transfers are
581 encrypted [13].

582 **3.1.1. Restricting Zone Transfer Transaction Entities**

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

591 **3.2. Zone Content Threats and Protection Approaches**

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

594   • Parameter values for certain key fields in RRs of various RR Types

595   • The presence of certain RRs in the zone file

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

598 **3.2.1. Lame Delegations**

599 Incorrect zone delegation (i.e., lame delegation) (see Appendix B) can result in a child zone
600 becoming unreachable (i.e., denial of service) or intermittently accessible if only one or more
601 NS RRset entries points to a non-existing server.

602 **3.2.2. Zone Drift and Zone Thrash**

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

610 The Refresh value should be determined by the expected rate of change for a zone. Suggested
611 values (in seconds) usually range from 1200 (20 minutes) to 432000 (12 hours) or even 864000
612 (1 day). The Retry value is the time that a secondary should wait before a failed refresh and
613 should be a fraction of the Refresh value for the zone.

614 **3.3. Dynamic Update Threats and Protection Approaches**

615 **3.3.1. Dynamic Update Misuse**

616 Dynamic updates involve DNS clients making changes to zone data in an authoritative name
617 server in real time [11]. As with zone transfer transactions, the threats associated with dynamic
618 update transactions have not been officially documented by the IETF in an RFC. However, the
619 following common threats could be expected:

620   • **Unauthorized updates** could have several harmful consequences for the content of
621   zone data, such as:

622   o Adding illegitimate resources

623   o Deleting legitimate resources

624   o Altering delegation information (NS RRsets pointing to child zones)

625 • Update tampering could affect data in a dynamic update request

626 • Replay attacks (i.e., update request messages captured and resubmitted later) could
627   cause inappropriate updates

3.3.2. Guidance on Securing Dynamic Updates

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

638 Dynamic updates on a zone file can only be directed to the primary name server, which holds
639 the writable copy of the zone file. Dynamic update requests generally originate from hosts (e.g.,
640 DHCP servers) that attempt to provision a new host name in the DNS zone when an IP address
641 is dynamically assigned to a client. Because dynamic update messages change the authoritative
642 zone data, they should only be accepted from authorized senders (e.g., TSIG, ACL).

3.4. DNS NOTIFY Threats and Protection Approaches

3.4.1. DNS NOTIFY Misuse Threats

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

3.4.2. DNS NOTIFY Protection

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

657 Once zone transfers have been set up between servers, secondary name servers should be
658 informed about changes to zone file data through a notification message. By default, a DNS

659 NOTIFY message is sent to every recognized secondary name server of the zone (i.e., name
660 servers listed in the NS RRset in the zone) whenever a primary name server detects a change in
661 the zone file. Most DNS server software provides the ability to also notify other servers, which
662 allows the administrator to account for any stealth name servers for the zone (i.e., not listed in
663 the NS RRset). DNS administrators should keep notifications on because this configuration will
664 allow updates to be propagated quickly to secondary name servers.

665 **3.5. Minimizing Information Leakage**

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

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

673 **3.5.1. Resource Record Information**

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

683 More careful consideration should be given to the TXT resource record type. A DNS
684 administrator will have to decide whether the data contained in a TXT RR constitutes an
685 information leak or is a necessary piece of information. For example, several authenticated
686 email technologies (e.g., sender policy framework [SPF], domain keying for internet mail
687 [DKIM], domain-based message authentication, reporting and conformance [DMARC]) use TXT
688 RRs to store email sender policy information, such as valid email senders for a domain [14].
689 These judgments will have to be made on a case-by-case basis.

690 **3.6. External Authoritative Domain Integrity**

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

693 **3.6.1. Dangling CNAME Exploitation**

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

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

703 **3.6.2. Lame Delegation Exploitation**

704 A lame delegation can result in domain hijacking. When a subdomain is delegated to a DNS-
705 hosting provider and the contract for providing DNS services for that domain lapses, threat
706 actors could hijack resolution for that subdomain by contracting with the provider that controls
707 the servers targeted by the delegation to host that subdomain under their control. This then
708 enables the threat actor to redirect resolution requests to their own infrastructure.

709 DNS administrators should actively validate that there are no lame delegations within their
710 external authoritative domain name space and use DNS-hosting providers who apply
711 safeguards.

712 **3.6.3. Look-Alike Domain Exploitation**

713 Threat actors extensively leverage look-alike or *typosquat* domains to impersonate target
714 organizations. By leveraging the positive reputation of legitimate organizations, threat actors
715 vastly increase the success rate of their phishing and malware campaigns. These look-alike
716 domains can include subtle variations of legitimate domains or text or character substitution to
717 register a domain that appears to be owned by a legitimate organization.

718 A common best practice is to monitor new DNS registrations to detect this attack vector and to
719 defensively register look-alike domains if feasible. Gaining visibility into these activities will
720 enable organizations to preemptively address a prevalent threat vector that targets their users
721 and consumers.

722 **3.7. Operational Recommendations**

723 **3.7.1. Resource Record TTL Value Recommendations**

724 Each RRset in a zone has its own time-to-live (TTL) value that tells recursive DNS servers and
725 clients how long (in seconds) the RRset data should be stored in its cache upon receipt,
726 although not all clients cache responses or strictly obey TTLs. When a recursive server receives
727 a DNS response to a query and performs all relevant checks, it stores the resulting RRsets in its

728 cache. The server cache decrements the TTL value of each RRset in its cache, and that data is
729 purged from the cache (excluding DNS services with prefetching or similar mechanisms) when
730 the TTL for any RRset reaches zero. This prevents caches from growing too large, removes old
731 and possibly incorrect DNS data from caches, and prevents stale results from being returned to
732 future queries.

733 The zone administrator can assign the default TTL value for all RRs in the zone, but each RRset
734 can have its TTL individually specified, and different RRsets in the same zone can have different
735 values. The zone administrator should set the default TTL value long enough to ensure that the
736 RRset will be useful for caches but short enough that any changes to the RRset will be
737 propagated quickly through the DNS (and old information purged). DNSSEC signature validity
738 periods should also be taken into consideration. TTL values should be less than the validity
739 period of the RRSIG that covers the RRset. DNSSEC-aware clients will decrement the TTL value
740 of an RRset in its cache to the signature expiration date if that date is before the projected TTL.
741 That way, the RRset will be purged before the signature expires and seen as BOGUS by other
742 DNSSEC validators. However, DNSSEC-unaware clients may not know to do this comparison, so
743 there is a risk that invalid DNSSEC RRsets will be stored in DNSSEC-unaware caches.

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

757 3.8. **DNSSEC Signing Considerations for Authoritative Service**

758 DNSSEC refers to a set of protocol extensions that add source authentication and integrity
759 protection to DNS data [15]. In DNSSEC, a digital signature covers a given RRset in a response
760 and is encapsulated through a special RRtype called RRSIG. The keys used to validate these
761 digital signatures are also stored in the DNS in DNSKEY RRsets. Trust in the public key is
762 established by building a chain of validated signatures and public keys that are sometimes
763 operationally differentiated as zone-signing keys (ZSK) and key-signing keys (KSK) from signed
764 DNS data to a trusted public key that is preconfigured on the system. The preconfigured public
765 key of the trusted zone is called the *trust anchor*.

766 DNSSEC can guarantee the integrity of name resolution response data to DNS clients that
767 perform DNSSEC signature verification. DNSSEC does not protect the confidentiality of DNS

768   query/response data. Confidentiality can be protected using DoH, DoT, or DoQ (see Section
769   4.2.1).

770   Deploying DNSSEC for an authoritative DNS service requires a set of steps to digitally sign the
771   zone data and configure the authoritative service [16]. The exact process for these steps
772   depends on the implementation used by the organization.

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

3.8.1. DNSSEC Key Considerations

776

777   DNSSEC is based on public-key cryptography to generate digital signatures over DNS data. The
778   current set of cryptographic algorithms is defined in RFC 8624 [17], which lists the mandatory
779   and optional algorithms supported by DNSSEC tools. Using RFC 8624 and the guidance on key
780   strength and lifetimes in SP 800-57 [18] and SP 800-131A [19] produces the recommended
781   digital signatures algorithms shown in Table 1.

782

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

783   SP 800-57 gives required security strengths and lifetimes for cryptographic key material based
784   on intended use. A strength of 112 bits is acceptable until 2030, when it will be raised to 128
785   bits. This is for DNSSEC signing and not validating, as using keys (or algorithm suites) for
786   backward compatibility with entities that do not have the same cryptographic requirements is
787   allowed. Additionally, some organizations (e.g., federal agencies) have additional requirements
788   around the use of FIPS 140-certified cryptographic modules [20].

789   As there are restrictions to the total size of a DNS response (without resorting to using TCP),
790   administrators should be aware of how the key and digital signature size affect the DNS
791   response size. Therefore, algorithms like ECDSA and ed448 are preferable over RSA, as they
792   produce smaller key and signature sizes. Table 1 does not include post-quantum cryptographic
793   (PQC) algorithms. At the time of writing, the use of PQC for DNSSEC has not been specified.
794   However, administrators should consider migrating to PQC algorithm usage when the use of
795   PQC algorithms has been specified, tools have been updated to include the new algorithms, and
796   the majority of DNS clients support them. ECC algorithms with smaller key sizes would be more
797   vulnerable to a quantum attack, as it would require a currently theoretical quantum computer

798 with fewer qubits than would be required for an RSA key with the same cryptographic strength
799 [21]. Since DNSSEC does not provide confidentiality, the risk of "store and future decrypt" with
800 quantum computers would not be a threat.

801 The key lifetime value is the period during which a key pair should be considered active and
802 used. Afterward, the key is retired and no longer used to generate signatures. Signing keys used
803 for DNSSEC are categorized as signature keys with a recommended maximum lifetime of 1-3
804 years. The act of retiring a signing key pair and introducing a new signing key pair is called a *key*
805 *rollover* in DNSSEC. As with other DNSSEC operations, there are automated tools available to
806 perform key rollovers with minimal administrator intervention based on the enterprise policy
807 on key lifetimes. Key lifetime values do not have a representation in DNSKEY or RRSIG RRsets.
808 They are completely internal to the enterprise and set by local policy.

809 **3.8.2. Using RRSIG Validity Periods to Minimize Key Compromise**

810 The best way for a zone administrator to minimize the impact of a key compromise is by
811 limiting the validity period of RRSIGs in the zone and in the parent zone. This strategy limits the
812 time during which an attacker can take advantage of a compromised key to forge responses. An
813 attacker that has compromised a ZSK can only use that key during the KSK's signature validity
814 interval, and an attacker that has compromised a KSK can only use that key during the signature
815 interval of the RRSIG covering the DS RR in the delegating parent. Therefore, ZSK validity
816 periods should be kept short to require frequent resigning. KSK changes and changes with the
817 registrar should be infrequent (e.g., once per year). To reduce the chance that the KSK is
818 compromised, strong modern algorithms (e.g., ECDSA) should be used.

819 **3.8.3. Hashed Authenticated Denial of Existence**

820 A side effect of the DNSSEC security extensions as they were first specified is the ability for a
821 user to "walk" a zone by sending a series of queries for NSEC RRs. A client could send a query
822 for an NSEC RR in the zone and "walk" the zone by sending a follow-up query for the NSEC RR
823 to the next name indicated in the received NSEC RR. This would result in a client being able to
824 enumerate the entire contents of a zone. While this is not an attack by itself (all DNS data is
825 considered public), it would most likely be a prelude to an attack. An attacker could enumerate
826 a zone to discover the IP addresses of servers to attack directly. This concern led to the creation
827 of NSEC3, but the emergence of attacks like KeyTrap (CVE-2023-50387[4]) has brought into
828 question whether the computational effort that NSEC3 requires is worthwhile. It appears that
829 since the emergence of KeyTrap, the more prudent approach is to use NSEC. If the use of NSEC3
830 is still required due to local policy, the NSEC3 parameters should be set to minimize the DoS risk
831 [22].

---

[4] CVE-2023-50387 detail from NIST National Vulnerability Database: https://nvd.nist.gov/vuln/detail/cve-2023-50387 (accessed Jan 2025)

832 **3.8.4. DNSSEC Algorithm Migration**

833 It may eventually be necessary to migrate to a new DNSSEC signing algorithm due to a
834 discovered weakness in the currently used algorithm or overriding policy decisions. Migrating
835 from the current DNSSEC algorithm to a new algorithm requires a set of steps and delays while
836 old data is removed from caches.

837 A proposed process can be found in RFC 6781 [16], which outlines the basic steps. To reduce
838 the risk of a validator thinking it is under a downgrade attack, the signatures for the new
839 algorithm are added before the DNSKEY RR with the new algorithm public key. Likewise, when
840 removing the retiring algorithm, the public key DNSKEY RR is removed first, followed by the
841 signatures.

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

846 **3.8.5. DNSSEC Signing Internal Zones**

847 Using DNSSEC to sign internal zones is generally considered bad practice. In the few instances
848 where internal zones must be signed, it is likely that the administrators already know that it is
849 required and why. Any existing regulatory or contractual requirements that include the signing
850 of internal zones should be reviewed.

851 There are two distinct functions that DNSSEC can perform: signing and validating. Signing allows
852 a domain owner to provide its customers with assurance that they are getting the correct DNS
853 data from the true owner. Validating is performed by recursive resolvers to check the
854 authenticity and integrity of the DNS responses it receives. The signing of zones should be
855 confined to public-facing external namespaces only, and validation should be enabled on any
856 recursive servers.

857 There are three major reasons why signing internal zones is discouraged:

858   1. Depending on the namespace, it is generally difficult if not impossible to tie the internal
859   zone back to the internet chain of trust. Because the chain will be broken, validating the
860   internal zones will require the management of additional trust anchors for each zone on
861   all of the validating resolvers.

862   2. If the zone is being dynamically updated, the entire zone will have to be re-signed each
863   time it is updated, which can be computationally intensive. For large or busy zones,
864   frequent re-signing could lead to the degradation or complete denial of DNS services for
865   clients.

866   3. DNSSEC-signing the internal zones serves no purpose when authoritative and recursive
867   services are combined on the same DNS servers, which is a common and acceptable
868   deployment for internal DNS systems. In this type of architecture, the DNS server would
869   be doing nothing other than validating itself.

870 Therefore, the general guidance would be that internal-only zones should not be DNSSEC-
871 signed. Any exceptions to this guidance are beyond the scope of this document.

872

873   **4. Recursive/Forwarding Service and Stub Resolvers**

874   A recursive name server for an organization performs the name resolution function on behalf of
875   a collection of clients. A recursive name server may also be called a caching name server or a
876   recursive resolver [23]. The name resolution function is performed by a recursive name server
877   in response to queries from a stub resolver. The search process for name resolution may
878   involve searching its own cache, forwarding to other designated name servers, recursively
879   querying various authoritative name servers through a set of iterative queries, or a combination
880   of these methods (see Appendix A).

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

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

896   **4.1. Threats to Recursive/Forwarding Service**

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

900   • A compromised authoritative name server (for queries that originate from a recursive
901   name server)

902   • A poisoned cache of a recursive name server (for queries that originate from a stub
903   resolver or a forwarding recursive name server)

904   • A recursive name server that is induced to query for a specific name, and forged
905   responses are immediately sent to the server attempting to get a malicious answer in
906   the cache (mitigations have historically been put in place to prevent this)

907   • Specific queries that expose bugs in name server software implementations to launch a
908   DoS attack or impact operations in some way

909   • A passive monitor that can observe DNS queries sent from a stub resolver, which could
910   lead to a loss of privacy for end users of the host system or a monitor learning what

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

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

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

916   • **Packet interception.** The attacker eavesdrops on a request and sends a response by
917   spoofing an authoritative name server before the real response from the legitimate
918   authoritative name server reaches the recursive name server.

919   • **ID guessing and query prediction.** The attacker guesses the ID field in the header of the
920   DNS request message (e.g., using brute force guessing) and possibly the query name
921   (QNAME) and query type (QTYPE). The attacker then injects bogus data into the
922   network as a response by spoofing a name server.

923   • **Responses from a compromised authoritative name server.** A compromised
924   authoritative name server is directed by a controlling adversary to send out bogus
925   responses to queries from recursive name servers.

926   • **Incorrect expansion rules applied to wildcard RRs.** Many zones use wildcard RRs to
927   economize on the volume of data in the zone file. The wildcard patterns are used for
928   synthesizing RRs to generate responses to queries as described in RFC 1034 [1] and RFC
929   4592 [24]. If synthesis rules are applied incorrectly in a name server, the RRs associated
930   with organizational resources existing may not be generated or made available in a DNS
931   response. This fault also results in a denial of service.

932   • **Removal of RRs from a response.** An attacker could also remove RRs from a response,
933   which may result in a name resolution failure and consequent denial of service.

934 **4.2. Recommendations for Protection**

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

943 **4.2.1. Encrypted DNS**

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

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

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

953 • DNS over Transport Layer Security (TLS) (DoT) [25] runs the traditional DNS protocol
954   over TLS, which is the same layer of encryption used to secure traffic between web
955   browsers and web servers that use HTTPS to communicate. TLS, in turn, runs over TCP.

956 • DNS over HTTPS (DoH) [26] runs the DNS protocol over HTTP, which in turn runs over
957   TLS. TLS runs over TCP.

958 • DNS over Quick UDP Internet Connections (QUIC) (DoQ) [27] runs DNS over QUIC, which
959   is an encrypted transport layer that runs over UDP.

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

962 **4.2.1.1. Encrypted DNS Guidance and Recommendations**

963 The U.S. Government requires the DNS infrastructure of Federal Civilian Executive Branch
964 (FCEB) agencies to support the use of encrypted DNS when communicating with agency
965 endpoints, wherever technically supported [31].

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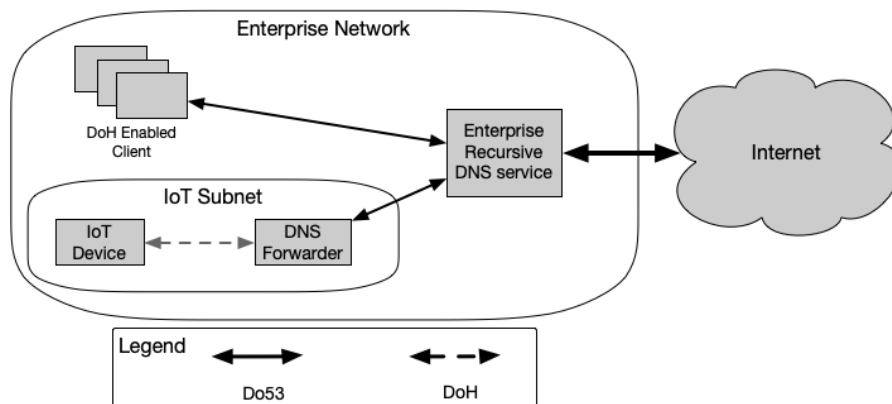972 recursive service, as shown in Fig. 2.

973

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

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

978   **4.2.1.2. Considerations for Using Encrypted DNS**

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

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

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

991   **4.2.1.3. Cryptographic Guidance**

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

995   If supported, name servers must be configured to only allow the cryptographic ciphers
996   permitted in SP 800-52 [28], and cloud-based Encrypted DNS providers should be assessed for
997   this. Additionally, some organizations may have requirements on the use of FIPS-140 approved
998   cryptomodules for use with TLS or DTLS [20], which would also include Encrypted DNS.

999   **4.2.2. Restricting the Use of DNS With Public Providers**

1000 In order to ensure that users do not use unauthorized public, internet-based DNS services,
1001 organizations should:

1002 • Block outbound DNS from the internal network to the internet, except for name servers
1003   that are authorized to communicate directly with name servers on the internet (e.g.,
1004   forwarders). This blocking can be implemented using firewall rules or router access
1005   control lists (ACLs) because DNS uses two well-known ports: UDP port 53 and TCP port
1006   53.

1007 • Restrict stub resolvers to only use authorized DNS services to leverage encrypted DNS
1008   wherever possible.

1009   • Block unauthorized DoT traffic from the internal network to the internet using firewall
1010   rules or router ACLs. DoT is straightforward to block because it uses the well-known TCP
1011   port 853.

1012   • Block unauthorized DoH traffic from the internal network to the internet using RPZs and
1013   firewall rules. DoH is more difficult to block because it uses the same TCP port as TLS:
1014   443. RPZs can help block the resolution of the domain names used to identify known
1015   DoH servers, and firewall rules can block access to public DoH services that run on
1016   dedicated IP addresses.

1017   • Use mobile device management (MDM) or other central management solutions to
1018   prevent users from configuring non-approved external encrypted DNS services.

1019   **4.2.3. Detecting and Mitigating Data Exfiltration via DNS**

1020   Organizations should establish controls to detect and block unauthorized applications from
1021   tunneling data within DNS packets. Signature-based systems can enable the detection of well-
1022   known DNS tunneling tools, but customized DNS data exfiltration tools should also be
1023   considered as threat actors increasingly turn to this tactic to avoid detection by signature-based
1024   systems.

1025   **4.2.4. Enabling DNSSEC Validation**

1026   Configuring DNSSEC validation requires two tasks: configuring the server to perform validation
1027   and configuring one or more trusted public keys to act as trust anchors. The method for
1028   achieving this depends on the DNS implementation used.

1029   The policy for determining which DNSSEC public keys to configure as trust anchors is beyond
1030   the scope of this guide and would be part of the organization's security policy. However, given
1031   the hierarchy used in DNS, the higher the public key in DNSSEC, the wider range of DNS
1032   responses can be validated using that key, as shown in Table 2.

1033   **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

1034   Therefore, the internet root key should be configured as a trust anchor.

1035   **4.2.5. Maintaining DNSSEC Trust Anchors**

1036   When a zone updates its DNSSEC signing keys (i.e., performs a key rollover), any validating
1037   recursive resolver that has configured that zone's key as a trust anchor must obtain the new
1038   singing key. If not, DNSSEC-signed responses from the zone (or delegated zones) will fail
1039   DNSSEC validation.

1040   There is an automated process for a zone to signal to validating recursive resolvers that it is
1041   performing a key rollover [29]. Administrators may not get advanced notification of a rollover,
1042   which makes relying on manual trust anchor updates risky. Validating recursive resolver
1043   administrators should enable automated trust anchor rollover and monitor logs to ensure
1044   stability in the recursive service.

1045   **4.3. Operational Recommendations**

1046   There are additional steps that an administrator can take when setting up a recursive server for
1047   an organization. Because it is unwise to allow queries from the internet to the recursive server,
1048   the recursive server can be placed behind a firewall that blocks inbound connections from UDP
1049   and TCP port 53 (used by DNS).

1050   Recursive servers should be configured to only accept queries from internal hosts and perform
1051   recursion for them. Different recursive DNS server implementations may have features to
1052   enable this ability. This could be done by implementation specific ACLs or network
1053   infrastructure configuration, which is beyond the scope of this document.

1054

1055   **5. Stub Resolvers**

1056   Software that require access to the internet or internal network resources (e.g., web browsers,
1057   email clients, cloud applications, operating systems) use a DNS client called the client resolver,
1058   resolver library, or stub resolver. A stub resolver is often referred to as simply a client. The stub
1059   resolver formulates a name resolution query for the resource sought by the network-accessing
1060   software and sends it to a recursive name server in the enterprise (see Appendix A). Stub
1061   resolvers are generally configured to send queries to two or more recursive name servers to
1062   provide some fault tolerance for their operation.

1063   The OS-layer stub resolver is a centralized DNS client within the operating system that handles
1064   DNS queries for all applications by sending them to a recursive resolver configured at the
1065   system level (e.g., via network settings). In contrast, an application-layer stub resolver operates
1066   independently within a specific application and bypasses the OS resolver to handle DNS queries
1067   directly. This is common in modern browsers or applications that support DoH or DoT for
1068   enhanced privacy and security. These two resolvers often interact when an application-layer
1069   stub overrides the system's default resolver settings, which can create conflicts or
1070   complementarities, depending on how they are configured. For instance, an application might
1071   use its own DNS settings for specific queries while still relying on the OS stub for others,
1072   creating a layered approach to DNS resolution.

1073   **5.1. Securing the Stub Resolver**

1074   Stub resolvers do not have many configuration options. Often, the only configuration necessary
1075   for an administrator is to enter the IP addresses of one or more recursive resolvers that the
1076   stub would rely on to resolve queries. It is a good idea for administrators to include the IP
1077   addresses of at least two recursive servers to increase the availability of the DNS service for end
1078   users. This can be done manually or using a protocol like DHCP. Additional resiliency can be
1079   provided to stub resolvers through the use of anycast for the DNS service. By adding a layer of
1080   abstraction between the IP of the DNS service and the individual servers via anycast, an
1081   administrator can lower the risk of client impact from the loss of any one DNS server. This can
1082   be especially important for supporting certain application stacks that will only use one DNS
1083   server, regardless of the stub resolver configuration.

1084   There is a known class of malware that attempts to change the settings on a system's stub
1085   resolver to direct queries to another (usually malicious) recursive server. The server may then
1086   direct end users to a malicious site where another attack takes place, or the server may simply
1087   direct users to a web page that serves ads or similar non-intended content. To combat this,
1088   administrators should make sure end user systems have the latest endpoint protection
1089   software and consider blocking all outbound DNS traffic with the exception of known recursive
1090   servers.

1091   Implementing enterprise mobility management (EMM) software can ensure that stub resolvers
1092   are correctly configured to point to authorized DNS servers. Additionally, EMM facilitates the
1093   management and enforcement of policies regarding the approval and use of software with
1094   integrated stub resolvers within the enterprise environment.

1095 **5.2. DNSSEC Considerations for Stub Resolvers**

1096 Most stub resolvers cannot perform DNSSEC validation and may not understand DNSSEC at all.
1097 These stub resolvers will have to rely on an upstream validating recursive resolver to perform
1098 DNSSEC validation on its behalf. This does not pose a significant risk on a trusted organization's
1099 network because there are several options to protect the link between a validating recursive
1100 resolver and a stub resolver that cannot do DNSSEC validation processing.

1101 If an organization's network is considered trusted (e.g., using one of the last hop mechanisms or
1102 similar), then the stub resolvers can be considered to be using DNSSEC. However, network
1103 administrators should be aware that DNSSEC validation failures could complicate the diagnosis
1104 of internet error messages. DNSSEC validation failures will be seen by the upstream validator,
1105 not the stub resolver that initiated the query. The stub resolver may only see a generic server
1106 failure message (SERVFAIL), which applications interpret differently. Network administrators
1107 should check validator logs when responding to network errors to rule out DNSSEC validation
1108 failures.

1109 **5.2.1. Recommendations for Providing Service to Mobile Hosts**

1110 Mobile or nomadic hosts present a particular challenge for network administrators. These
1111 systems often access a network outside of the trusted enterprise, so mobile hosts must either
1112 perform their own validation or have a trusted connection to an enterprise-approved DNS
1113 service. If the mobile hosts can perform their own validation, then the same policy for the
1114 enterprise validators should be applied for the mobile host. That is, the same trust anchors and
1115 validation policy should be set for mobile hosts as for validators on the enterprise network.

1116 It may be necessary for a mobile system to configure its validator when migrating from the
1117 enterprise to an external network and vice versa. If the enterprise network is trusted, the
1118 mobile host can rely on the enterprise validator when on the enterprise network and perform
1119 its own validation when on external networks. Ideally, network administrators can avoid this
1120 problem by using alternative names for internal and external zones, thus having different trust
1121 anchors.

1122 If the mobile host cannot perform its own validation, it must either have a secure tunnel back
1123 to the enterprise network or a secure connection to an approved DNS recursive service. Many
1124 enterprises already have a means for mobile hosts to access internal resources (e.g., file
1125 servers), so a validating recursive server should be added as one of the services provided to
1126 mobile hosts through a secure channel. Alternatively, enterprise mobile hosts could be
1127 configured to use an approved cloud-based DNS recursive service to allow mobile hosts to use
1128 approved DNS recursive services without needing to have a connection to the enterprise
1129 infrastructure.

References

[1] Mockapetris, Paul. "Domain Names - Concepts and Facilities." Request for Comments. Internet Engineering Task Force, November 1987. https://doi.org/10.17487/RFC1034

[2] Mockapetris, Paul. "Domain Names - Implementation and Specification." Request for Comments. Internet Engineering Task Force, November 1987. 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] dnstap community (2025) dnstap. Available at https://dnstap.info

[5] Atkins, Derek, and Rob Austein. "Threat Analysis of the Domain Name System (DNS)." Request for Comments. Internet Engineering Task Force, August 2004. https://doi.org/10.17487/RFC3833

[6] Turner, Sean. "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms." Request for Comments. Internet Engineering Task Force, March 2011. https://doi.org/10.17487/RFC6151

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

[8] Dupont, Francis, Stephen Morris, Paul A. Vixie, Donald E. Eastlake 3rd, Ólafur Guðmundsson, and Brian Wellington. "Secret Key Transaction Authentication for DNS (TSIG)." Request for Comments. Internet Engineering Task Force, November 2020. https://doi.org/10.17487/RFC8945

[9] Lewis, Edward P., and Alfred Hoenes. "DNS Zone Transfer Protocol (AXFR)." Request for Comments. Internet Engineering Task Force, June 2010. https://doi.org/10.17487/RFC5936

[10] Ohta, Masataka. "Incremental Zone Transfer in DNS." Request for Comments. Internet Engineering Task Force, August 1996. https://doi.org/10.17487/RFC1995

[11] Vixie, Paul A., Susan Thomson, Yakov Rekhter, and Jim Bound. "Dynamic Updates in the Domain Name System (DNS UPDATE)." Request for Comments. Internet Engineering Task Force, April 1, 1997. https://doi.org/10.17487/RFC2136

[12] Eastlake 3rd, Donald E. "DNS Request and Transaction Signatures (SIG(0)s)." Request for Comments. Internet Engineering Task Force, September 2000. https://doi.org/10.17487/RFC2931

[13] Toorop, Willem, Sara Dickinson, Shivan Kaul Sahib, Pallavi Aras, and Allison Mankin. "DNS Zone Transfer over TLS." Request for Comments. Internet Engineering Task Force, August 2021. https://doi.org/10.17487/RFC9103

[14] 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

[15] Hoffman, Paul E. "DNS Security Extensions (DNSSEC)." Request for Comments. Internet Engineering Task Force, February 2023. https://doi.org/10.17487/RFC9364

[16]   Kolkman, Olaf, Matthijs Mekking, and R. (Miek) Gieben. "DNSSEC Operational Practices, Version 2." Request for Comments. Internet Engineering Task Force, December 2012. https://doi.org/10.17487/RFC6781

[17]   Wouters, Paul, and Ondřej Surý. "Algorithm Implementation Requirements and Usage Guidance for DNSSEC." Request for Comments. Internet Engineering Task Force, June 2019. https://doi.org/10.17487/RFC8624

[18]   Barker E (2016) Recommendation for Key Management: Part 1: General. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-57pt1r4. https://doi.org/10.6028/NIST.SP.800-57pt1r4

[19]   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

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

[21]   Roetteler, Martin, Michael Naehrig, Krysta M. Svore, and Kristin Lauter. "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms." arXiv, October 31, 2017. https://doi.org/10.48550/arXiv.1706.06752

[22]   Hardaker, Wes, and Viktor Dukhovni. "Guidance for NSEC3 Parameter Settings." Request for Comments. Internet Engineering Task Force, August 2022. https://doi.org/10.17487/RFC9276

[23]   Hoffman, Paul E., and Kazunori Fujiwara. "DNS Terminology." Request for Comments. Internet Engineering Task Force, March 2024. https://doi.org/10.17487/RFC9499

[24]   Lewis, Edward P. "The Role of Wildcards in the Domain Name System." Request for Comments. Internet Engineering Task Force, July 2006. https://doi.org/10.17487/RFC4592

[25]   Hu, Zi, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman. "Specification for DNS over Transport Layer Security (TLS)." Request for Comments. Internet Engineering Task Force, May 2016. https://doi.org/10.17487/RFC7858

[26]   Hoffman, Paul E., and Patrick McManus. "DNS Queries over HTTPS (DoH)." Request for Comments. Internet Engineering Task Force, October 2018. https://doi.org/10.17487/RFC8484

[27]   Huitema, Christian, Sara Dickinson, and Allison Mankin. "DNS over Dedicated QUIC Connections." Request for Comments. Internet Engineering Task Force, May 2022. https://doi.org/10.17487/RFC9250

[28]   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

[29]   StJohns, Michael. "Automated Updates of DNS Security (DNSSEC) Trust Anchors." Request for Comments. Internet Engineering Task Force, September 2007. https://doi.org/10.17487/RFC5011

1217 [30] Eric Goldstein, Executive Assistant Director for Cybersecurity, CISA, CISA Launches its
1218 Protective DNS Resolver with General Availability for Federal Agencies (Sep. 27, 2022).
1219 Available at https://www.cisa.gov/news-events/news/cisa-launches-its-protective-dns-
1220 resolver-general-availability-federal-agencies
1221 [31] Cybersecurity & Infrastructure Security Agency (2025) Protective Domain Name System
1222 (DNS) Resolver. Available at: https://www.cisa.gov/resources-tools/services/protective-
1223 domain-name-system-dns-resolver
1224 [32] Internet Engineering Task Force (2025), RFC Editor. Available at: https://www.rfc-
1225 editor.org/search/rfc_search_detail.php?title=DNS&page=All
1226

1227 **Appendix A. DNS Protocol Tutorial**

1228 A DNS can refer to the global namespace, the infrastructure used to operate that namespace,
1229 and the protocol used by hosts to resolve domain names to IP addresses or other information.
1230 This appendix presents a high-level overview of how a DNS operates.

1231 **A.1. DNS Namespace and Infrastructure**

1232 The DNS infrastructure is made up of computing and communication entities that are
1233 geographically distributed throughout the world. The domain name space is organized in a
1234 hierarchy. The topmost level in the hierarchy is the *root domain*, which is represented as a dot
1235 ("."). The next level in the hierarchy is called the *top-level domain* (TLD). There is only one root
1236 domain, but there are many TLDs. Each TLD is called a child domain of the root domain. In this
1237 context, the root domain is the parent domain because it is one level above a TLD. Each TLD, in
1238 turn, can have many child domains. The children of TLDs are called *second-level* or *enterprise-*
1239 *level domains*.

1240 In a domain name representation, the symbol for the root domain is usually omitted. For
1241 example, consider the domain name marketing.example.com. The rightmost label in this
1242 domain name ("com.") is a TLD. The next label to the left ("example") is the second-level or
1243 enterprise-level domain. The leftmost label ("marketing") is the third-level domain. It is also
1244 possible to have a fourth-level domain, fifth-level domain, and so on. Because each of the labels
1245 in marketing.example.com is called a domain (e.g., TLD, second-level domain, third-level
1246 domain, etc.), the concatenation of all of these labels from the current level to the TLD is a *fully*
1247 *qualified domain name* (FQDN). In this document, however, the FQDN is simply referred to as a
1248 domain name, and the level name is used to identify individual labels.

1249 There is only one root domain. There are several hundreds (possibly thousands) of TLDs
1250 categorized into the following three types:

1251 • **Country-code TLDs (ccTLDs)** — Domains associated with countries and territories. There
1252   are more than 240 ccTLDs. Examples include .uk, .in, and .jp.

1253 • **Sponsored generic TLDs (gTLDs)** — Specialized domains with a sponsor that represents
1254   a community of interest. These TLDs include .edu, .gov, .int, .mil, .aero, .coop, and
1255   .museum.

1256 • **Unsponsored generic TLDs (gTLDs)** — Domains without a sponsoring organization. The
1257   list of unsponsored gTLDs includes .com, .net, .org, .biz, .info, .name, and .pro.

1258   Figure 3 shows a partial DNS name space hierarchy.
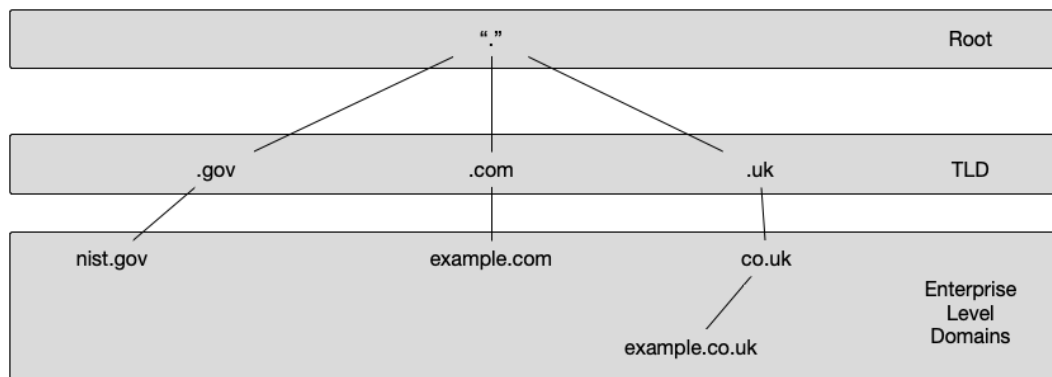
1259

1260   **Fig. 3. DNS tree**

1261   A user (i.e., an individual or corporation) wishing to register a domain name must contact an
1262   authorized entity called a registrar (which may charge a fee, depending on the TLD in question).
1263   *Registrars* are companies that are authorized to register domain names in a particular TLD or
1264   sub-domain of a TLD (e.g., co.uk.) to end users. When the registrar receives the user's
1265   registration request, the registrar verifies that the name is available by checking with the
1266   registry that manages the corresponding TLD (or sub-domain under the TLD). If the name is
1267   available, the registrar registers the name with the appropriate registry. The registry then adds
1268   the new name to its registry database and publishes the new name in DNS. In some domains
1269   (e.g., country codes, gTLDs), the same organization acts as the registry and registrar for names
1270   in the domain.

1271   Organizations that register and obtain an enterprise-level domain can then create child
1272   domains to properly identify internet resources associated with various functional units. For
1273   example, the owner of the domain name example.com might create the subdomain "shipping"
1274   to create and identify resources associated with the shipping department of the organization.

1275   To facilitate this grouping, the DNS defines the concept of a zone. A zone may be an entire
1276   domain or a domain with one or more subdomains. It is a configurable entity within a name
1277   server under which information on all internet resources pertaining to a domain and a selected
1278   set of subdomains is described. Thus, zones are the administrative building blocks of the DNS
1279   name space just as domains are the structural building blocks. As a result, the term *zone*
1280   commonly refers to a domain that is managed as a stand-alone administrative entity (e.g., the
1281   root zone, the .com zone).

1282   **A.2. DNS Queries and Responses**

1283   When a user types the URL www.example.com into a web browser, the browser program
1284   contacts a type of resolver called a *stub resolver* that then contacts a local recursive name
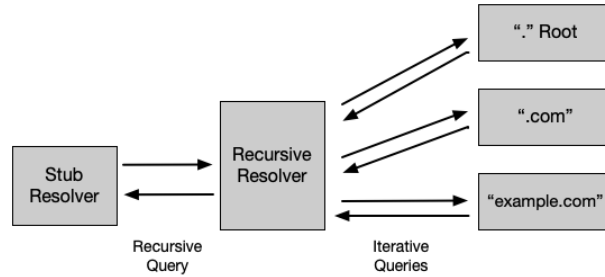1285   server, as shown in Fig. 4.

1286

Fig. 4. DNS resolution

1288 The resolving name server checks its cache to determine whether it has valid information to
1289 provide the IP address for the accessed internet resource (e.g., www.marketing.example.com).
1290 If not, the resolving name server checks the cache to determine whether it has information
1291 regarding the name server for the zone marketing.example.com, since this is the zone that is
1292 expected to contain the resource. If the name server's IP address is in the cache, the resolver's
1293 next query will be directed against that name server or a name server higher up in the
1294 hierarchy.

1295 If a complete search of the cache does not yield the required information, the resolving name
1296 server must start its search by querying the name server in the topmost zone in the DNS name
1297 space hierarchy (i.e., the root server). Contact with the root servers is enabled by the "root
1298 hints" file that is usually present in every name server in DNS. The root server will contain
1299 information about the name servers for its child zones (i.e., TLDs). A TLD (e.g., .com) will contain
1300 name server information about its child zones (e.g., example.com). The name server
1301 information about its child zones that is carried in a zone is called *delegation information*. The
1302 delegation information is used by a zone to refer name resolution requests for a resource lower
1303 than it in the domain name hierarchy. Since the name resolution request in this example
1304 pertains to a resource in the third-level domain, the root server must refer the request to a
1305 lower-level name server.

1306 The response to the resolving name server that involves sending this delegation information is
1307 called the *referral*. The referral provides the name and IP address for the name servers for the
1308 TLD zone that is relevant to the request (i.e., the .com zone). Using this referral, the resolving
1309 name server then formulates and sends a query to the .com zone name server. This server will
1310 provide the referral for example.com's name server. If the marketing.example.com domain is
1311 included in example.com's zone, querying the name server for example.com will provide the IP
1312 address for the resource www.marketing.example.com.

1313 As the description of the name resolution process makes clear, a name server provides:

1314 • A referral to a child zone

1315 • A mapping from the domain name to the IP address (i.e., domain name resolution) or
1316   vice versa (i.e., inverse resolution)

1317 • An error message if the query is for a DNS entry that does not exist

1318 The name server performs these three functions with the same DNS database, which is called a
1319 *zone file*. The mapping function is performed by a class of information in a zone file called

1320   *authoritative information* and is provided by a set of records that list the resources in that zone,
1321   its domain name, and its corresponding IP address. Because the resources belong to that zone,
1322   the information provided is deemed authoritative. Thus, a zone file contains two categories of
1323   information: authoritative information (i.e., information about all resources for all domains in
1324   the zone) and delegation information (i.e., information about name servers for child zones). The
1325   locations in the zone file where delegation information appears are called *delegation points*.
1326   The level of a zone file is the level of the topmost domain for which it contains authoritative
1327   information. In the previous example, the zone file in the name server of example.com is the
1328   enterprise-level zone file, and the corresponding name server is called the enterprise-level
1329   name server.

1330   To fully understand the DNS protocol, readers are urged to consult the DNS protocol
1331   specifications maintained by the Internet Engineering Task Force (IETF) [32].

1332

1333 **Appendix B. Glossary**

1334 The general terms related to DNS are collected and defined by the IETF in RFC 9499 [23]. Terms
1335 that do not appear there or require additional context include the following.

1336 **dangling CNAME**
1337 A CNAME record maps an alias to a canonical name, which in turn can be resolved to an IP address via a record of
1338 the canonical name. If the record for the canonical name is removed, the CNAME is left "dangling," as what it
1339 refers to is no longer present in the DNS. If the CNAME refers to a canonical name outside of the organization's
1340 control, a malicious actor may gain administrative control of the canonical name (e.g., if the registration of the
1341 canonical domain has expired, thus potentially appearing as part of the organization owning the CNAME).

1342 **DNS hygiene**
1343 The practice of maintaining a secure, resilient, and well-configured DNS environment. This includes proactive
1344 management and monitoring of DNS configurations to eliminate vulnerabilities (e.g., lame delegations, dangling
1345 CNAME records). It also involves the removal of any RRsets that are no longer in use. Additionally, DNS Hygiene
1346 encompasses the continuous monitoring and mitigation of look-alike domains to protect against phishing,
1347 impersonation, and other malicious activities.

1348 **Encrypted DNS**
1349 A collective term for the use of DNS over TLS (DoT), DNS over HTTPS (DoH), DNS over QUIC (DoQ), and other
1350 means to encrypt communication between stub resolvers and recursive DNS servers.

1351 **forged responses**
1352 A response that is different from the one that is expected from a legitimate authoritative name server. A bogus
1353 response can originate from a compromised authoritative name server or a poisoned cache of a recursive name
1354 server.

1355 **lame delegation**
1356 Delegating a zone to any name server that is not authoritative for that zone. This can happen if the subdomain
1357 expires and is not removed from the parent domain, or name servers for the delegated domain change but the
1358 changes are not synchronized with the parent domain. [23]

1359 **look-alike domains**
1360 Domain names that are created for malicious purposes or to catch user errors when typing domain names. They
1361 visually look like a legitimate domain name but may include characters that are substituted from a different
1362 alphabet script.

1363 **Protective DNS**
1364 A DNS service that is enhanced with security capabilities to analyze DNS queries and responses and takes action to
1365 mitigate threats. The outcomes of deploying Protective DNS should include:

1366 • Blocking or redirecting harmful traffic in real time at the point of domain name resolution, typically before
1367   malicious activity starts

1368 • Blocking categories of traffic with DNS via the categorization of domain names that do not conform to an
1369   organization's policies or matching against known bad actor lists

1370 • Delivering visibility into real-time and historical DNS query and response data to facilitate digital forensics
1371   and incident response

1372 • Integrating with the wider security ecosystem as part of defense in depth (e.g., correlating an
1373   organization's data on assets and users with the IP addresses of blocked queries).

1374 • Facilitating an organization's responsibility to comply with regulatory or contractual requirements for
1375   blocking traffic to disallowed sites (e.g., copyright violations, legal restrictions)

1376 **typosquat**
1377 A domain that is close to a valid domain but contains a misspelling or some other error when typing in the domain.
1378 It may be registered by an attacker who hopes to direct a user to a malicious site (e.g., register "xample.com" and
1379 "exampel.com" in an attempt to redirect users who mean to enter "example.com").