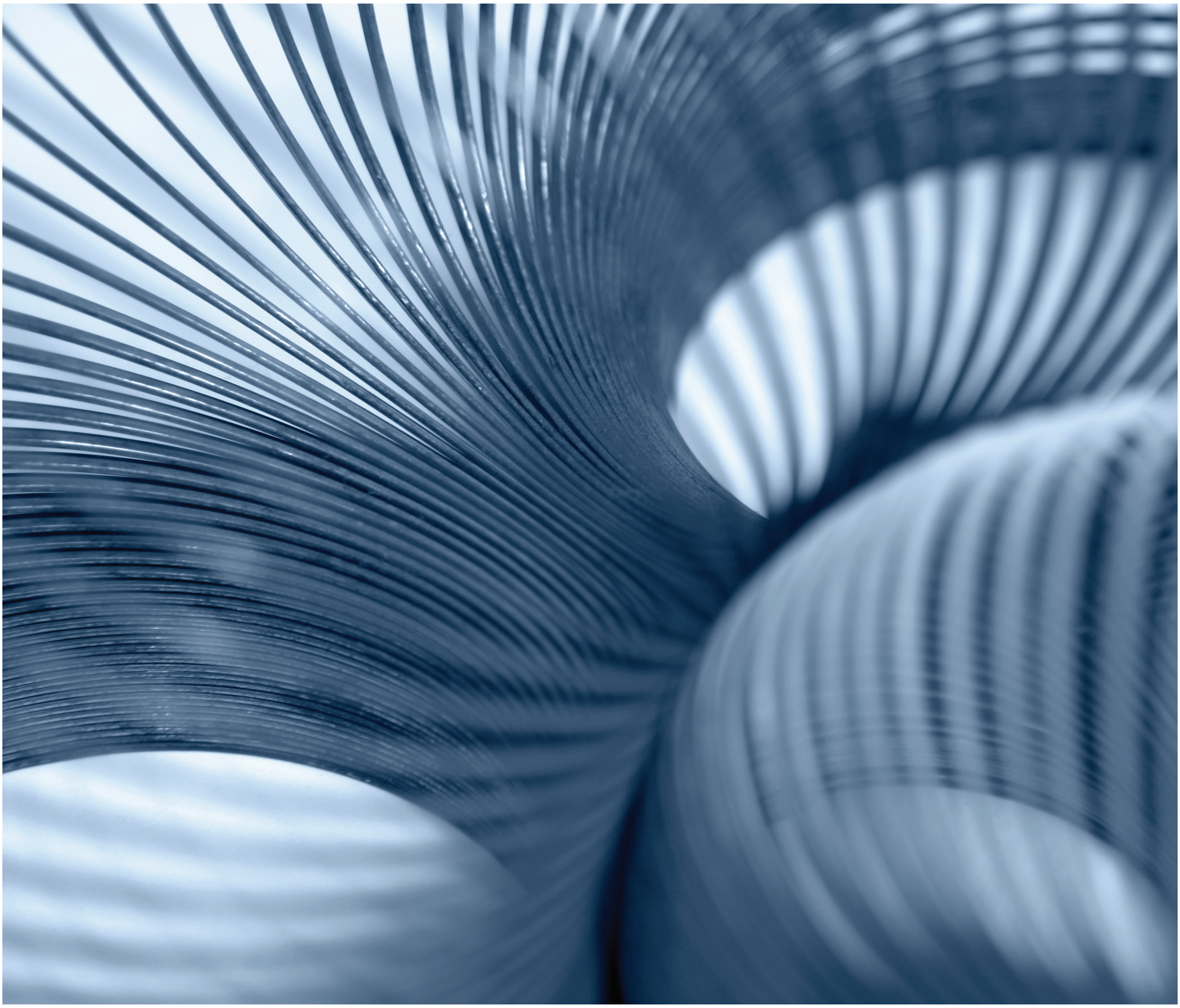

Towards Improving DNS Security, Stability, and Resiliency

David Conrad

www.internetsociety.org

Towards Improving DNS Security, Stability, and Resiliency

DNS Resiliency

Towards Improving DNS Security, Stability, and Resiliency

Executive Summary

The Domain Name System, continually evolving since its invention 30 years ago, is a core component of the Internet. Translation services provided by the DNS create a mapping between human friendly names and machine-preferred numbers (and vice versa). The DNS is used by the majority of services and applications available today in the Internet. As the Internet has become a critical resource with constant security attacks and threats, the DNS has also been attacked and threatened.

While the threats to the DNS are significant, mitigations can either eliminate or limit many of the risks to the DNS. At the same time, new protocol developments and operational best practices have increased the resilience, stability and security of the DNS protocol and the global DNS infrastructure.

The goal of this paper is to produce a comprehensive view on the DNS threats, their potential impacts, and available mitigation technologies and strategies. This paper begins by providing an overview of the DNS and its evolution. Then, threats to and from the DNS are described, followed by the discussion of mitigation technologies and strategies. This discussion is summarized at the end in a set of recommendations aimed at addressing the risks associated with the Internet's DNS.

This paper provides background information for the continuing dialog on the challenges the DNS faces, ways to further improve and evolve the DNS, and how to increase the security, stability, and resilience of the DNS.

Within the context of DNS security, stability, and resilience, the most significant threats to the operation of the DNS and their affects on the security, stability, and resiliency of the DNS are summarized in the table below.

Class of Threat	Threat Description
Denial of Service (DoS)	DoS attacks are characterized as either *resource starvation* in which all available resources are consumed or resource disruption in which resources are made unavailable until an external event occurs. Both resource starvation and *resource disruption* Denial of Service incidents can significantly impact the security, stability, and resiliency of the DNS.
Data Corruption	Data corruption threats affect integrity of data provided through the DNS. Data corruption attacks can strike at many points within the DNS infrastructure. Data Corruption has most impact on DNS security, but can also have significant impact on DNS stability and resiliency. (See Figure 3 on page 16 for a definition of these terms.)
Information Exposure	In Information Exposure threats, DNS data is overly exposed. Information Exposure attacks have the least impact on DNS security, stability, and resiliency since the DNS was never designed with a requirement for confidentiality of data. However, Information Exposure can affect the trust individuals have in the DNS, may result in changes in how DNS and the Internet are used.

The DNS has also been used as a vector for attack on other parts of the Internet, including DNS amplification attack (using DNS as a tool for a DoS attack on a third party), Fast Flux DNS (using the DNS to hide the source of criminal activities), and DNS as a Covert Channel.

Fortunately, mitigating techniques can reduce the risks effectively and allow for the continued secure, stable, and resilient operation of this core component of the Internet.

Denial of Service (DoS) attacks are most likely to have widespread effects on the stability of the Internet. A DoS attack could be against a single domain name, against the root servers that glue together the Internet's DNS, or against any part of the infrastructure in between. The impact of DoS would vary with the target. For example, without the root servers, the Internet would effectively cease to function, albeit not immediately.

Attackers intent on a denial of service can bring massive numbers of compromised systems as unwilling participants in their attack. The main strategy for repelling a DoS attack is *overprovisioning*, ensuring that the target has more resources available than the attacker can consume. DNS operators who understand the critical nature of the service they provide have designed the DNS very carefully, hardening systems against attacks, and using large-scale replication technology to help ensure survival in the face of a DoS attack.

Data Corruption attacks can be used to redirect Internet users to forged information resources. When successful, they may result in theft, fraud, misdirection and expose a user to significant risks, and in general reduce confidence and trust in the Internet. A new technology, DNSSEC, extends the DNS to help prevent many data corruption attacks. DNSSEC is being integrated into the Internet's DNS very slowly because it requires changes to every system using the DNS, including the resolving DNS servers at ISPs throughout the world. When fully in place, DNSSEC will significantly reduce the risk of data corruption.

Information Exposure is the most nebulous of the threats to the DNS. Information exposure through the DNS can be damaging to some individuals and organizations, and can threaten the trust the people have placed in the Internet. Information exposure can be minimized by following best practices, including securing the network paths used for DNS and access control restrictions.

The DNS itself has also been used as a vector for attack. While threats to the Internet from the DNS have less impact on its resilience, stability and security, DNS operators are taking these threats seriously. Many have put into place policies and procedures intended to minimize this type of threat.

While the threats both to and from the DNS are significant, mitigations are available to either eliminate or limit the risks experienced by DNS operators and end users alike.

Table of Contents

List of Figures, Graphics, and Tables

Introduction

The Domain Name System (DNS) provides translation from human-friendly names to data in other formats. It is a globally distributed database and is a critical component of the Internet. The most common use of the DNS is the translation from names such as “www.example.com” to the “dotted-quads” of IPv4 addresses, such as 192.168.1.64, or “colon separated hex” of IPv6 addresses like fd63:fad8:482a:65d3::0:f0cc. However, the DNS is used in the modern Internet for much more than that and now acts as a form of “directory assistance operator” for both human-to-machine as well as machine-to-machine interactions. In addition to IP addresses, the DNS is used to look up mail servers, cryptographic keys, latitude and longitude values, and other diverse types of data. The vast majority of uses of the Internet are critically dependent on the reliable, trustworthy, and responsive operation of the DNS.

Since its invention nearly three decades ago, the DNS has been continuously improved, becoming more capable, resilient, and secure. Still, the DNS is subject to a variety of threats and attacks. One goal of this document is to help readers understand today’s threats to the DNS and how those threats can be mitigated, whether through capabilities inherent in the DNS, via operational practices, or through policy implementation.

Because the DNS is critical to the operation of the Internet, we must engage in continued dialog on the threats and risks to the DNS. This document provides valuable background material on existing threats to the DNS, risks of system failure, mitigation of the risks, and areas where further work is required to more completely mitigate risks to the DNS.

Overview of the DNS

The DNS is both a set of protocols and the global distributed database used on the Internet.[1] The database is a network of more than 16 million servers, all cooperating to provide a globally distributed “directory assistance operator” service, translating between human-oriented alphanumeric labels and machine-oriented numbers. While the Internet Protocol (IP) itself does not use the DNS, a large-scale or systemic failure of the DNS would make the Internet unusable. Internet users and systems would have to stop using human-friendly names and start using IP addresses for everything. Web and email traffic would grind to a halt. DNS failures, both accidental and malicious, have taken companies such as Microsoft[2] and Amazon[3] and even countries such as Sweden[4] and Germany[5] partially or fully off the Internet.

Beyond facilitating entertainment and commerce, the Internet has become a key component of many nations’ telecommunications infrastructure. Internet transactions now make a significant contribution to many country’s economies[6]. As a foundation of the Internet, any threat to the DNS deserves significant attention because of its potential to interrupt the flow of information with devastating effects on business operations, both at the organizational and national level.

History of the DNS

In the earliest days of the “network or networks” that would become the Internet, names of systems connected to the network were assigned locally, and there was no DNS. “The NIC” or the Network Information Center kept track of the names, but was often notified of name assignments or changes after the fact. RFC 597, “Host Status,” published in December 1973, was the first official collection of hostnames - all 90 of them - that were on the network at the time. Each system manager was expected to use RFC 597 (and those that would come after) to keep their local list of host-to-address mappings up to date.

[1] The DNS protocols are defined in a series of Internet RFCs (see “History of the DNS” in this document). In addition to the Internet, those protocols can also be used inside of an organization to create a separate distributed database, not connected to the Internet at all. The case of a disconnected private DNS is not considered in this document.

[2] http://www.microsoft.com/presspass/press/2001/jan01/01-24dnspr.mspx

[3] http://www.pcworld.com/businesscenter/article/185458/ddos_attack_on_dns_hits_amazon_and_others_briefly.html

[4] http://www.iis.se/en/pressmeddelanden/felaktig-dns-information-2

[5] http://www.securityweek.com/content/reports-massive-dns-outages-germany

[6] http://www.eg8forum.com/fr/documents/actualites/McKinsey_and_Company-internet_matters.pdf

An online file maintained by the NIC containing the official name to address mappings was proposed in RFC 606, “Host Names On-Line,” published December 1973. L. Peter Deutsch, the author of RFC 606, wrote:

> *Now that we finally have an official list of host names, it seems about time to put an end to the absurd situation where each site on the network must maintain a different, generally out-of-date, host list for the use of its own operating system or user programs.*

Unlike DNS names used today, early names as defined in RFC 606 were simple labels. For example, the system at MIT’s Artificial Intelligence lab was called “MIT-AI.” The simple label approach to naming hosts on the network lasted for nearly a decade.

However, it was clear that the simple label approach would not work forever. In September 1981, D. L. Mills noted in RFC 799, “Internet Name Domains”:

> *In the long run, it will not be practicable for every internet host to include all internet hosts in its name-address tables. Even now, with over four hundred names and nicknames in the combined ARPANET-DCNET tables, this has become awkward. Some sort of hierarchical name-space partitioning can easily be devised to deal with this problem; however, it has been wickedly difficult to find one compatible with the known mail systems throughout the community.*

RFC 805, “Computer Mail Meeting Notes,” details a February 1982 meeting at which the decision was made to move to a “hierarchy of domains”. This new approach to host naming, described as the “Domain Naming Convention for Internet User Applications” was codified and introduced in August 1982 with the publication of RFC 819, by Zaw-Sing Su and Jon Postel. The Domain Naming Convention was intended to use hierarchy as a way of distributing administrative management of the namespace. This would eliminate name collisions, such as when two different Computer Science Departments at two different universities named their Digital VAX-11/750 computers “csvax.”

RFC 819 provides the general outline of what would become the DNS, including the ideas of naming authorities, registrars, and iterative and recursive resolvers. RFC 819 states:

> *The intent is that the Internet names be used to form a tree-structured administrative dependent, rather than a strictly topology dependent, hierarchy.*

RFC 819 also defined the first top-level domain, .ARPA, as “the set of organizations involved in the Internet system through the authority of the U.S. Defense Advanced Research Projects Agency.”

In October 1982, RFC 830 was published, describing "A Distributed System For Internet Name Service". The author of the RFC, Zaw-Sing Su, described an architectural view of a name resolution service "provided through the cooperation among a set of domain name servers (DNSs)" and discussed system components such as the database, caching of names, application interfaces, and protocols for inter-process communication necessary to implement a distributing naming system. The goal of RFC 830 was to focus discussion on the increasingly important topic of Internet names and to progress work towards standardization.

Progress after Zaw-Sing Su's architectural straw man was rapid. By November 1983, Paul Mockapetris had published RFCs 882 "Domain Names – Concepts and Facilities" and RFC 883 "Domain Names – Implementation and Specification," giving the initial specifications for the DNS as we know it today.

The structure of names in the DNS was also defined very early in the history of the Internet. Jon Postel and Joyce Reynolds published RFC 920, "Domain Requirements," in October 1984. This RFC defined the original top-level domains (ARPA, GOV, EDU, COM, MIL, and ORG), the two letter country domains (such as .DE for Germany and .CH for Switzerland), and opened the possibility of other top-level domains for "multiorganizations," large international organizations-of-organizations, which could be entitled to become top-level domains.

In 1987, RFCs 1034 and 1035 were published to replace the original DNS RFCs 882 and 883, respectively. These RFCs updated the DNS specifications with experiences gained from implementations of RFCs 882 and 883. RFCs 1034 and 1035 remain the core standards on which the DNS is based.

Over time, the DNS has been modified to address new requirements and changes to the DNS operational environment. The most significant changes to the DNS are summarized in the table below.

RFCs	Importance
1591, “Domain Name System Structure and Delegation” (March/1994)	Documents the structure of the top-level of the DNS, discusses categorization the initial 7 top-level “world wide generic domains” and outlines how delegated domains are to be administered
1886, “DNS Extensions to support IP version 6” (December/1995)	Defines DNS support for IPv6
2065, “Domain Name System Security Extensions” (January/1997)	Provides specification for security extensions necessary to assure the integrity and authenticity of data supplied by the DNS. These specifications were revised and enhanced several times, ultimately resulting in the publication of RFCs 4033, 4034, and 4035 in March 2005, collectively known as the “DNSSEC” specification.
2825, “A Tangled Web: Issues of I18N, Domain Names, and the Other Internet Protocols” (May/2000)	Discusses the issues involved in allowing for the internationalization of the DNS, extending the characters used for DNS names. This led to RFCs 3490, 3491, and 3492 (March/2003), which provide the mechanisms by which international characters could be used in the DNS.
2826, “IAB Technical Comment on the Unique DNS Root” (May/2000)	Explains the architectural necessity for having a single root in the DNS to ensure a coherent namespace.
2860, “Memorandum of Understanding Concerning the Technical Work of the Internet Assigned Number Authority (IANA)” (June/2000)	Documents the relationship between the Internet Corporation for Assigned Numbers and Names (ICANN) and the Internet Engineering Task Force (IETF) with respect to the Internet operations, including the creation of new top-level domains.

Figure 1 Significant Changes to the DNS since 1987

In all, between 1987 when RFCs 1034 and 1035 were published and July 2011, 516 RFCs relating in some way to the definition of the DNS or how it operated have been published.

Appendix A – *DNS Enhancements since RFC 1034/1035* provides a list of updates, modifications, and revisions to the DNS.

The DNS in Operation

The DNS is a conceptually simple system that allows a string of labels (such as "www," "isoc," and "org") joined by dots into a "domain name" to be looked up in a database distributed across multiple DNS servers. The domain name lookup results in an answer, or the answer that "there is no answer." Answers in DNS are known as "resource records," abbreviated as "RRs."[7]

The dots in a domain name are important because they represent potential administrative boundaries. For example, the dot between "isoc" and "org" in the domain name "www.isoc.org" represents the administrative boundary between the "org" top-level domain and ISOC, the organization responsible for "isoc.org." The Internet's DNS is a single large tree, read right-to-left, with progressively more specific administrative units to the left.[8] The term "zone" is used to indicate administrative units within the DNS tree. For example, the "isoc.org" zone is the piece of the DNS tree including all names ending in ".isoc.org." Further subdivisions are common, even within a single organization, and "isoc.org" might have multiple zones, such as "ch.isoc.org," "de.isoc.org" and "us.isoc.org."

The most common type of DNS lookup is for IP addresses. This is the type of lookup that occurs each time a user types a URL into a web browser, for example. Normally, the individual application (such as the web browser) does not perform the full lookup, which involves several steps. Instead, the application uses an intermediate system called a "resolver" (because it *resolves* DNS name lookups) to navigate the DNS distributed database to retrieve the information requested.

Types of DNS Servers

Resolvers are one of two types of servers that support the DNS. Resolvers make queries on behalf of applications and (usually) cache the responses to improve DNS performance and scalability. In the case of smaller enterprises and end users, Internet service providers

[7] Resource Records are also used for the lookup string in the DNS. Readers familiar with databases may want to mentally think of "resource records" as "tuples," as they are very similar in concept.

[8] In Internationalized Domain Names (IDNs), the choice of a right-to-left alphabet (as is used in Hebrew and Arabic alphabets) and the way it is represented to the user may reverse the traditional right-to-left hierarchy of the elements of a DNS name.

typically operate resolvers. In the case of larger enterprises, the resolvers are usually operated by the enterprises themselves or by large-scale DNS hosting providers.

Authoritative servers are the other type of DNS server. An authoritative server responds to lookup requests with one of:

- A positive response in which an answer to the question is provided;
- A negative response indicating the answer does not exist; or
- A referral providing an indication of where further information may be obtained.

Authoritative servers are typically operated by or on behalf of zone administrators. ISPs, DNS registrars, and hosting providers often operate authoritative servers on behalf of their customers. The authoritative DNS infrastructure, particularly for "high value" zones such as top-level domains, is being increasingly outsourced to DNS-focused service providers, such as Verisign, Afilias, and Neustar.

These two types of servers, recursive and authoritative, work together to lookup names in the DNS and return the results to applications.

A DNS Query in Detail

Figure 2, "DNS resolution of www.isoc.org" (below) shows the process by which an application looks up the IPv4 address associated with the name "www.isoc.org." The process begins when the user enters a domain name, e.g., www.isoc.org, into the application, such as a web browser. The web browser will submit the name "www.isoc.org[9]" to the DNS via a "resolver" and ask or *query* for the IP address(es) associated with that website (step 1).

The resolver will first[10] ask one of the set of DNS name servers known as the *root name servers* for the translation (step 2), and that root name server will respond with a "referral", telling the resolver to query the DNS name servers for the .ORG domain for the answer to the lookup (step 3).

The resolver next asks one of the .ORG DNS name servers for the translation (step 4). The .ORG DNS name server responds by referring the resolver to the DNS name servers for the ISOC.ORG domain for the answer (step 5).

[9] or WWW.ISOC.ORG, which is equivalent since DNS names are case insensitive.

[10] The resolution scenario described here assumes an empty cache, e.g., when the resolver first starts up. In most cases, many of the steps described here will be skipped because the relevant DNS data has already been fetched by a previous query and stored in the resolver's cache.

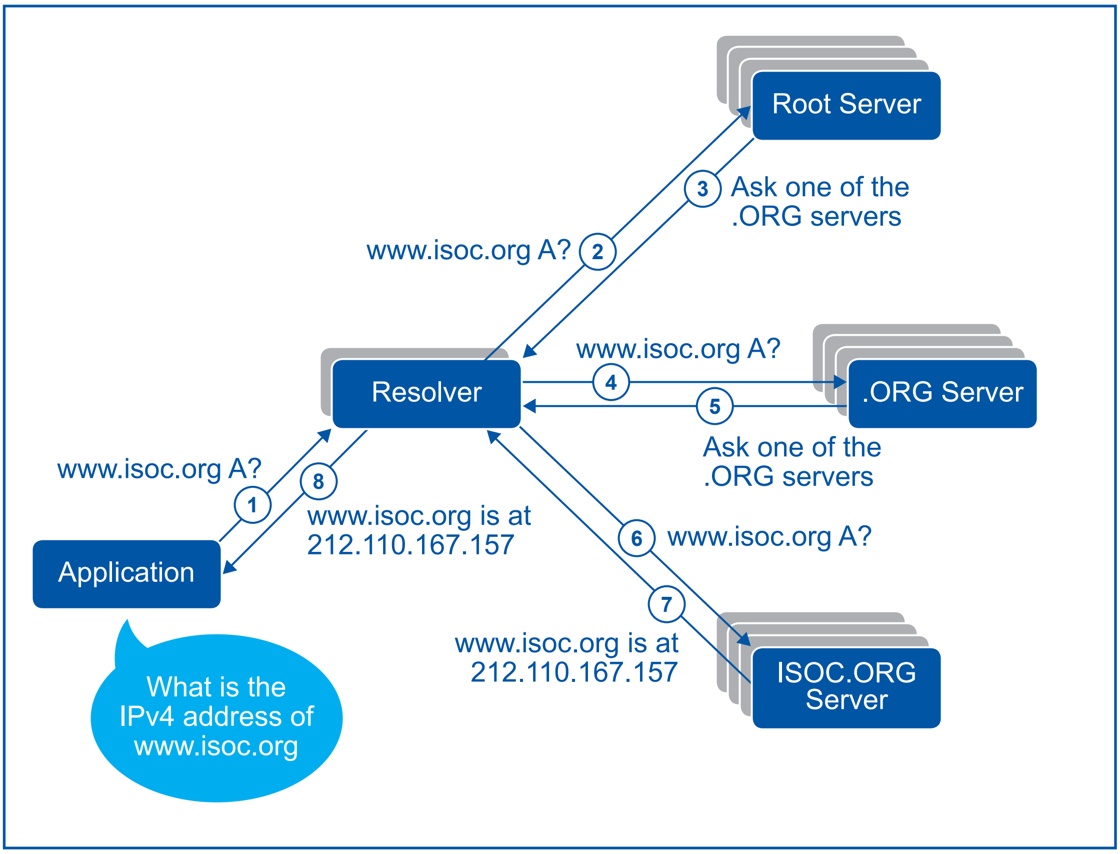
Root Server
3 Ask one of the .ORG servers
www.isoc.org A? 2
Resolver
www.isoc.org A? 4
.ORG Server
5
Ask one of the .ORG servers
www.isoc.org A?
1
8
www.isoc.org is at 212.110.167.157
Application
6 www.isoc.org A?
7
www.isoc.org is at 212.110.167.157
ISOC.ORG Server
What is the IPv4 address of www.isoc.org

Figure 2 DNS resolution of www.isoc.org

The resolver continues by asking one of the DNS name servers for the ISOC.ORG domain for the answer (step 6). In this case, the DNS name server sends a response with resource records containing result of the lookup: the IP address associated with WWW.ISOC.ORG (step 7).

Now that the resolver has the answer it is looking for, it can send the result of the lookup back to the original application. The resolver passes the address to the browser (step 8), allowing the browser to open a connection to the ISOC web server.

Caching is an important part of the operation of the DNS. At each stage along the way, from application to resolver to name server, information may be cached. In normal DNS operation, to improve performance, the resolver will store the responses and referrals it has received so that future lookups for the same information can be answered immediately. Caching is used to avoid sending queries across the network to DNS servers, speeding response time. Because caching is an expected part of DNS operation, each domain name response (resource record) includes a "time to live" (TTL) value indicating how long information may be cached[11].

DNS Security Extensions (DNSSEC)[12]

DNSSEC will be mentioned many times in this document, as the DNS Security Extensions are an important tool in increasing the security of the Internet's Domain Name System. Standardized in 2005, DNSSEC is not commonly used even in 2012. However, adoption of DNSSEC is expected to accelerate since the root DNS zone was signed in July 2010.

DNSSEC is a set of extensions to the DNS that provide authentication and integrity checking of DNS data. Authentication ensures that zone administrator can provide authoritative information for any particular DNS domain,[13] while integrity checking ensures that information in the DNS cannot be modified (accidentally or maliciously) while in transit or in storage.

DNSSEC requires both compliant DNS servers and security-aware DNS resolvers. DNS servers compliant with DNSSEC must support the additional types of DNS records needed for DNSSEC. Security-aware DNS resolvers must be able to detect the new DNSSEC extensions, and must check DNS data for authentication and data integrity.

DNSSEC was designed to provide a strong cryptographic signature of DNS data that security-aware (DNSSEC compliant) resolvers can verify to ensure data received over the network hasn't been modified since the data was signed. DNSSEC-signed DNS data can be retrieved from anywhere, regardless of any insecurities in the networks over which the DNS may travel or intermediate systems in which the data may reside. Any modification of the DNS data from what was originally signed at the authoritative source can be detected, thereby allowing a security-aware DNS resolver to discard corrupt or unauthorized data.

[11] The TTL value is specified by the domain name's administrator, the individual that manages the zone on behalf of the domain name owner.

[12] DNS Security Introduction and Requirements http://www.ietf.org/rfc/rfc4033.txt See also RFC 4034 ("Resource Records for DNS Security Extensions") and RFC 4035 ("Protocol Modifications for the DNS Security Extensions")

[13] Zone administrator as indicated by holding the private zone signing key.

The Root of the DNS

The Internet DNS is a tree-shaped hierarchy, with various branches starting from the well-known top-level domains such as ".COM," ".ORG," ".MX" and ".UK." The base of the DNS tree is called the "root" of the tree (shown as "Root Server" in Figure 1). It is the piece of the DNS tree that points to each of the branches. The root of the DNS tree is the single point required for operation of the DNS. A failure of the root or its administration would—in theory—result in a failure of the DNS system as a whole. However, in practice, there are numerous safeguards protecting the root from failure.

The root of the DNS tree is composed of two parts: a root zone file, and the many name servers that act as the "Root Servers" for the Internet.

The root zone file is a relatively small file that lists all top-level domains and the name servers for those domains. For example, the root zone file has an entry for ".CH" (the top-level domain for Switzerland), lists the names of the 6 name servers responsible for the ".CH" domain, and has the IP addresses of those name servers. Without this entry, names ending in ".CH" could not be looked up in the DNS, as no name server would know where to find the information.

The root zone file is managed by three different organizations: ICANN, the U.S. Dept. of Commerce, and Verisign. ICANN[14] accepts and validates changes from the various top-level domain authorities, and then proposes specific modifications to the root zone file. The U.S. Dept. of Commerce's National Telecommunications and Information Administration (NTIA) authorizes the modifications, and then Verisign applies the updates, signs the zone using DNSSEC, and publishes the revised root zone file on a "distribution master" server.

The 13 root name servers, called "a.root-servers.net" through "m.root-servers.net" are operated by the 12 independent organizations shown in Figure 3[15]. These root name server operators fetch the root zone file from the distribution master maintained by Verisign, and publish the information on the root name severs they independently operate. Most of these root name servers are actually clusters of machines, many of which are distributed globally to multiple sites and make use of a routing technique known as "Anycast" (described in more detail later).

[14] In this case, ICANN is acting in the role of the Internet Assigned Numbers Authority. This role could be moved to another organization, but the function would be the same.

[15] Derived from information available at http://www.root-servers.org/

Root	Organization	Sites[16]
A	Verisign, Inc.	6
B	University of Southern California, Information Sciences Institute	1
C	Cogent Communications	6
D	University of Maryland, College Park	1
E	U.S. NASA Ames Research Center	1
F	Internet Systems Consortium, Inc.	49
G	U.S. Department of Defense, Network Information Center	6
H	U.S. Department of Defense, Army Research Lab	2
I	Netnod	38
J	Verisign, Inc.	70
K	RIPE NCC	18
L	ICANN	39
M	WIDE Project	6

Figure 3 Root Server Operators

[16] The numbers given in the “Sites” column were accurate when this document was written, in Fall, 2011, but are constantly growing. More and more “Anycast” instances are also being deployed as part of the Internet’s DNS. More current information is available at http://www.root-servers.org/

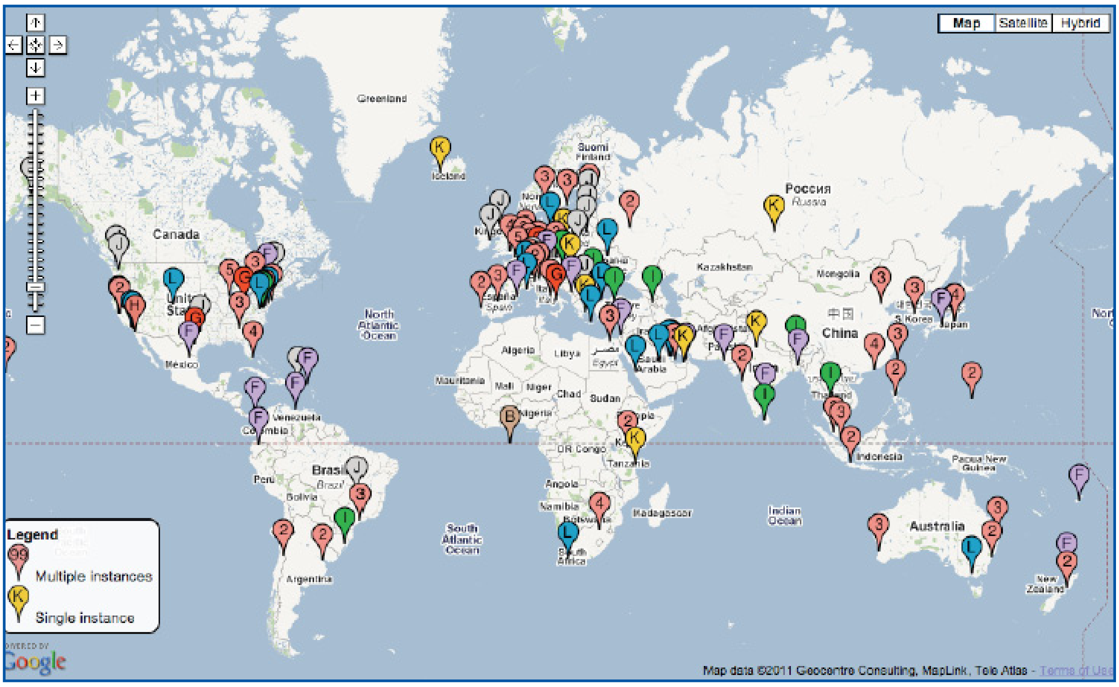

Figure 4 Global Root Server Distribution

In Figure 4 (“Global Root Server Distribution”), geographic distribution of the root name servers is shown with color-coding indicating the different organizations managing the root name servers. Considerable care has been taken in the design and deployment of the root name servers to minimize the risk of failure.

DNS Security, Stability, and Resilience

As shown in Figure 2 (“DNS resolution of www.isoc.org”), a lookup for any domain name involves multiple parties. In the example, five different parties all cooperate to answer the simple question, “what is the IP address of www.isoc.org,” including:

- the original query submitter (the user entering www.isoc.org in the browser),
- the operator of the DNS resolver (typically the user’s ISP),
- the operators of the root DNS server,
- the operators of the .ORG DNS server, and,
- the operator of the ISOC.ORG domain server.

In addition, any DNS lookup may also involve the operators of numerous Internet-connected networks, physical and virtual servers, support and back-office systems, and related infrastructure.

The many parties and components involved in every single DNS lookup multiply the potential risks to the security, stability, and resilience of the DNS. Due to the importance of the DNS for the operation of the Internet, any event that negatively impacts DNS Security, Stability, or Resiliency would have significant impact on the Internet.

Term	Definition
DNS Security	“The ability of the components of the DNS to protect the integrity of DNS information and critical DNS system resources.” Computer security is “the ability of a system to protect information and system resources with respect to confidentiality and integrity,”[17] but the DNS has not historically had requirements for confidentiality.
DNS Stability	“The ability of the entire name resolution system and its component parts to be able to respond to DNS queries.”[18] This is also known as “DNS System Stability.”
DNS Resiliency	“The ability of the DNS to provide and maintain an acceptable level of name resolution service in the face of faults and challenges to normal operations.”[19]

Figure 5 Definition of Security, Stability, and Resiliency

[17] Taken from “Computer Security: A Practical Definition” available from: http://www.albion.com/security/intro-4.html

[18] Two other definitions of DNS stability also exist. “DNS Name Stability” is consistency of names within a domain over time. That is, if names within a domain change with high frequency, the domain is unstable. “DNS Resolution Stability” is consistency in relation to performance. For example, if a query takes 10 milliseconds to respond in one instance and 1000 milliseconds to respond in a second instance, resolution time is unstable.

[19] Based on a definition from https://wiki.ittc.ku.edu/resilinets_wiki/index.php/Definitions

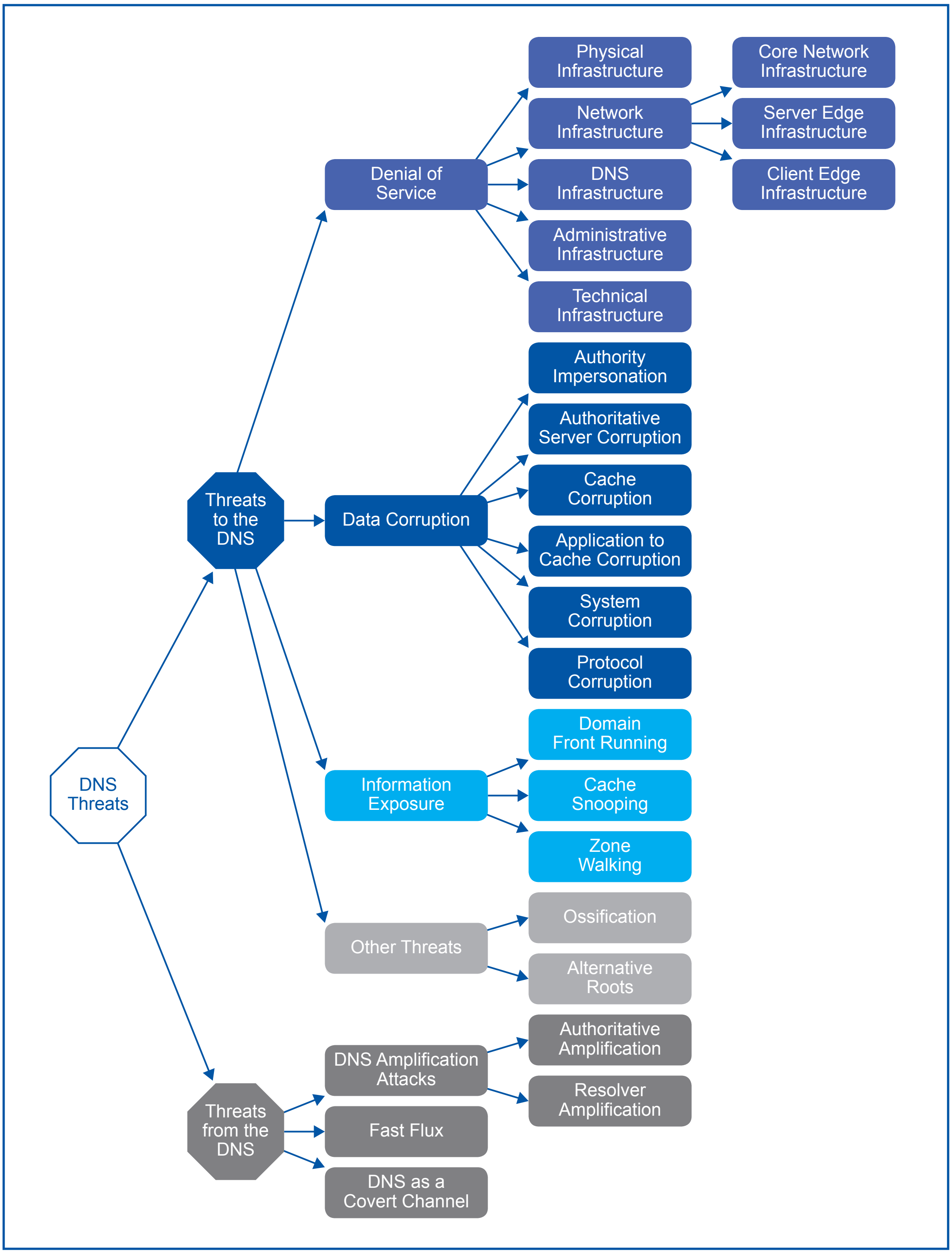

Figure 6 Common DNS Threats

DNS Threats

Internet researchers have identified a wide variety of security threats involving the DNS. (See Figure 6, which summarizes the most commonly identified threats[20].) In this section, we will provide an overview of threats to the DNS (attacks in which the DNS itself is threatened) and threats originating in the DNS (attacks in which components within the DNS infrastructure, such as DNS servers, are used to attack other assets).

Threats to the DNS

Some threats to the DNS have the potential to severely disrupt the operation of the Internet as a whole, while other threats to the DNS are more targeted and could be used against an individual organization. To help organize these different types of threats to the DNS, we have categorized them as:

- Denial of Service (keeping Internet users from using the DNS),
- Data Corruption (unauthorized change of information in the DNS), and
- Information Exposure (disclosure of information about Internet users by examination of their DNS traffic).

Denial of Service

The most significant threat to the DNS is Denial of Service, which is also the hardest to defend against. When DNS is slow or inaccessible because of a Denial of Service (DoS) attack, this can be the result of malicious activities in which an attacker purposefully tries to disrupt service, or an accident in which some aspect of the DNS is impacted due to infrastructure failure, mistakes, or acts of nature. The worst-case result of a DoS attack is complete disruption of all DNS-related and DNS-dependent services, that is, most services in use on the Internet today. In less severe cases, users of the DNS (both human as well as automated systems) can face increased delays, timeouts, and other performance-related issues.[21]

Denial of Service can have an impact on DNS security, stability, and resiliency of each of these component parts of the DNS. A DoS incident can impact the integrity of DNS information and system resources by making that information and those resources unavailable, either for use or as a subject of management.

[20] Adapted from Figure 1 of “DNS Threat Analysis”, available from: http://nlnetlabs.nl/downloads/se-consult.pdf

[21] Denial of Service attacks may come from a single system, or they can be distributed among a large number of attacking systems. These are known as Distributed Denial of Service, or DDoS, attacks. In this document, we will use the term DoS to include both single system and multi-system (DDoS) attacks.

Type of Denial of Service	Definition	Example(s)
Resource Starvation	Insufficient resources (such as Internet bandwidth, server CPU, or memory) are available to provide services.	An attacker floods the network with traffic, blocking reliable transmission of DNS requests and replies; attacker is able to disrupt normal operations of routers or switches.
Resource Disruption	An event makes the resource unavailable until some external event restores the resource.	Power failure (intentional or accidental); attacker is able to crash DNS servers.

The threat of DoS applies to all components of the DNS including:

- Physical and network infrastructure: buildings, power supplies, network connections;
- Server infrastructure: servers and related system that allow for DNS queries to be sent and received;
- Management infrastructure: processes that allows for the creation, modification, and deletion of DNS content; and
- Administrative infrastructure: support agreements and fault escalation procedures, staffing, invoicing, and similar arrangements.

Figure 7and Figure 8 below summarize the effects that the different types of DoS attacks would have on the security, stability, and resilience of DNS.

Impact of: Resource Starvation Denial of Service		
Type of Impact	**Level of Impact**	**Notes**
DNS Security	Low	As soon as DoS attack is stopped, normal operations should resume.
DNS Stability	Medium to High	As long as the attack continues, names affected by that DoS are either unavailable or degraded. However, overall DNS system stability could be severely affected by cascading effects in which a DoS against one system results in overload causing that first system to fail, resulting in increased load in associated systems, causing them to fail, etc. For example, if a DoS targets only one of a set of busy servers, the automatic failover mechanism of DNS resolvers may shift “non-DoS” load to other servers, increasing their load beyond their capabilities, and causing them to fail.
DNS Resiliency	Medium	Components affected by the DoS would be unable to provide service during the attack, but would return to normal as soon as the DoS was terminated.

Figure 7 Impact of Resource Starvation Attacks

Impact of: Resource Disruption Denial of Service		
Type of Impact	**Level of Impact**	**Notes**
DNS Security	Medium	Disruptions could cause corruption of data. For example, a sudden power failure may cause hard disk data to be scrambled.
DNS Stability	High	As long as the attack continues, names affected by that DoS are either unavailable or degraded until a system reset occurs. The same cascading effects seen in Resource Starvation DoS attacks can also occur.
DNS Resiliency	High	Components affected by the DoS would be unable to provide service even after the attack until a reset occurs.

Figure 8 Impact of Resource Disruption Attacks

Data Corruption

Data Corruption is a broad threat to the DNS that can have many different causes. Data Corruption occurs when DNS responses don't match the intended, published data. Data corruption can occur when someone has intentionally (or accidentally) changed the data in DNS servers in unexpected ways. Data corruption can also occur as DNS queries and responses pass over the Internet, for example, if an intermediate DNS server (resolver) has corrupt or incorrect data inserted into its cache (known as cache poisoning). Data corruption can also occur when an attacker sends answers to queries faster than the legitimate servers can answer, providing erroneous data to the application.

Impact of: Data Corruption

Type of Impact	Level of Impact	Notes
DNS Security	High	Corruption directly affects the integrity of DNS information.
DNS Stability	Low to High	Any corruption of DNS data affects the ability of the system to properly resolve names. In particular, if DNSSEC were being used, any data corruption would become a Denial of Service attack, with DNS resolvers discarding corrupt responses that fail validation. Name stability can be affected, for example, if corrupt data indicated that a valid name or zone doesn't exist. Resolution stability could also be affected if bogus referrals were injected into a response stream, resulting in queries being directed to inappropriate name servers.
DNS Resiliency	Low	Data corruption does not attack the infrastructure directly because the intent is to provide misleading information as a prelude to other forms of attack.

Figure 9 Impact of Data Corruption Attacks

Information Exposure

The DNS protocol was defined as a mechanism to associate named resources to underlying addresses or other data. Privacy protection was not a required feature. There is, however, a growing expectation that individuals should be able to use the DNS (and the related WHOIS protocol) in an anonymous fashion. Anonymous use of DNS would let individuals perform DNS queries without having their requests observed, aggregated and correlated with their identity[22].

DNS queries and responses are currently transmitted without any form of encryption, and thus can be observed at multiple points, from the network where the initial query occurs all the way to the authoritative servers responding to the query. DNS operators may also be capable of correlating requests and using them for other purposes (e.g., for advertising segmentation) without the knowledge and consent of the individual issuing the queries. In many jurisdictions, this would be considered a breach of those individuals' rights to privacy.

Impact of: Information Exposure		
Type of Impact	**Level of Impact**	**Notes**
DNS Security	High	As integrity of the DNS is dependent on trust in the system, inappropriate information exposure can impact DNS security.
DNS Stability	Low	Information exposure is largely passive with respect to the DNS system, so does not affect stability or resiliency.
DNS Resiliency	Low	Information exposure is largely passive with respect to the DNS system, so does not affect stability or resiliency.

Figure 10 Impact of Information Exposure Attacks

[22] The term "identity" here refers to the combination of attributes, assertions, or other observable traits and behaviors (e.g. online transactions) that can reasonably identify a natural person.

Other Threats to the DNS

The DNS is also subject to risks that may impact the evolution and use of the DNS, but which are not immediate operational threats in the way that data corruption or denial of service are.

Ossification

The DNS is now over 24 years old. Because the Internet is so dependent on the DNS, Internet engineers have found rolling out enhancements and extensions to be incredibly difficult. The need to maintain backwards compatibility with existing DNS implementations (and widely-deploy mis-implementations and shortcuts) means that changes standardized more than a decade ago can't be relied upon to work.

This situation has been termed ossification, a term which has as one of its definitions

> A tendency toward or state of being molded into a rigid, conventional, sterile, or unimaginative condition[23]

Ossification of the DNS has resulted in a system less able to adjust to future changes in the Internet. Ossification impacts DNS by delaying deployment of technologies needed to increase the security, stability and resilience of the DNS.

Alternate Roots

The DNS relies a hierarchy based on a single, globally unique root for both technical and usability reasons. As the Internet Architecture Board stated in RFC 2826[24]:

> To remain a global network, the Internet requires the existence of a globally unique public name space. The DNS name space is a hierarchical name space derived from a single, globally unique root. This is a technical constraint inherent in the design of the DNS. Therefore it is not technically feasible for there to be more than one root in the public DNS. That one root must be supported by a set of coordinated root servers administered by a unique naming authority.

However, there are differing views about the negative impact of having different roots with different information in them. This has resulted in numerous efforts to create "alternative roots". As discussed in "Alternative TLD Name Systems and Roots: Conflict, Control and Consequences"[25], alternative roots rely on resolver operators to modify name server con-

[23] http://www.merriam-webster.com/dictionary/ossification, third definition.
[24] http://www.rfc-editor.org/rfc/rfc2826.txt
[25] http://www.icann.org/en/committees/security/alt-tlds-roots-report-31mar06.pdf

figuration to use an alternative set of root name servers (which serve a root zone different than the one maintained by the IANA) or require end users to install special software that intercepts DNS queries and redirects them to alternative servers.

The threat to the DNS represented by alternative roots derives from the potential collisions, either accidental or intentional, of names within the Internet. Today, any particular domain name is controlled by a single domain owner. Alternative roots risk more than one domain owner holding the same domain name, or having a domain name exist through one root but not through another. How that domain name would resolve those names would depend on which root system was consulted. Since it is relatively rare for end users to control the resolution process (e.g., to run a resolver on their laptop), the response to DNS queries for each name would depend on which network the end user was connected to, the configuration of the network, the upstream ISP and other time- and location-sensitive factors. As such, alternative roots can be seen to potentially have significant impact on the security, stability, and resilience of the DNS.

Threats from the DNS

As a ubiquitous and core component of the Internet, DNS traffic often receives a lower level of scrutiny and filtering by routers and firewalls. As a result, DNS traffic has been used as a vector for several forms of attack. This section discusses some of these vectors.

DNS Amplification Attacks

Amplification attacks occur when an attacker sends a small request to a DNS server resulting in that server sending a response ten to one hundred times larger. Because the responses are larger than the initial requests, the attacker's resources are amplified, increasing the likelihood that the attacker can exhaust the resources of the victim. The DNS protocols are especially suitable for amplification attacks because:

- Responses are generally larger — sometimes much larger — than requests, amplifying the ability of the attacker;
- DNS does not require the establishment of a TCP connection, which allows the attacker to easily redirect the responses away from the attacker's network to a victim host by spoofing the source IP address as the victim's address; and
- DNS queries are small enough that the attacker's queries to multiple DNS servers are likely to be undetectable, while generating massive distributed denial of service attacks to a victim host.

- When a spoofed address is used, these attacks are very difficult to trace using current Internet mechanisms.

While the DNS has been used for amplification attacks for some time[26], the deployment of DNSSEC exacerbates this risk. DNSSEC-signed responses can be significantly larger than non-DNSSEC responses, and may be harder to filter than other records (such as large text records) used in DNS amplification attacks. Daniel Bernstein reports[27] that a 36-byte DNS query can result in a 3995-byte DNS response.

Fast Flux DNS

ICANN's Security and Stability Committee provides the following definition:

> "Fast flux" is an evasion technique that cyber-criminals and Internet miscreants use to evade identification and to frustrate law enforcement and anticrime efforts aimed at locating and shutting down web sites used for illegal purposes.[28]

In Fast Flux DNS, networks of servers (typically systems compromised by malware) are used as name servers. This allows for very rapid changes to DNS-related data, which helps cyber-criminals and miscreants delay or evade detection and mitigation of their activities.

Fast Flux exploits the stability and resilience of the DNS to make it difficult to eliminate systems being used for criminal activities. Fast Flux can frustrate both administrative remedies and technical remedies. Fast Flux isn't a threat to any component of the DNS infrastructure, but it is a threat to Internet users that is facilitated by the DNS.

DNS as a Covert Channel

DNS requests and responses can be used as a channel for covert communications[29]. In several documented cases, the DNS has been used as the mechanism for communications between botnet command and control servers the systems that make up the botnet.[30]

In addition to these existing attacks, protocols have been developed that use the DNS as a general-purpose (IP over DNS or TCP over DNS) tunneling protocol, allowing arbitrary data to be passed over the DNS protocol[31]. This use of the DNS as a Covert Channel

[26] http://www.isotf.org/news/DNS-Amplification-Attacks.pdf suggests widespread DNS amplification attacks started around 1999.

[27] http://dnscurve.org/amplification.html

[28] http://www.icann.org/en/committees/security/sac025.pdf

[29] http://www.oe.energy.gov/DocumentsandMedia/DNS_Exfiltration_2011-01-01_v1.1.pdf

[30] A "botnet" is a network of compromised systems under control of a single individual, sometimes called the "bot herder." Botnets can have hundreds to hundreds of thousands of systems, giving the bot herder the capability to launch very large and powerful distributed denial of service (DDoS) attacks.

[31] http://www.loria.fr/~lnussbau/files/tuns-sec09-article.pdf provides an overview of challenges of using the DNS as

could allow an attacker to bypass network security mechanisms, facilitating compromise of internal systems.

Application Corruption Attacks

In some cases, DNS responses can cause unexpected application behaviors. When these responses are maliciously constructed, they may be used to compromise systems. For example, one documented attack used reverse DNS names created with strings that embedded characters significant to Unix operating system shell scripts. These names were then used to compromise systems that expected the results of certain types of DNS lookups to be safe.[32]

As web application designers have learned from the huge wave of SQL injection and cross-site scripting attacks, all applications should assume that any data obtained over a network channel could be intentionally malicious or accidentally malformed. In cases where application or library source code is unavailable[33], some caching resolvers have configuration options that allow filters to be applied to responses to reduce the risk of malicious data being supplied to applications.

a transport protocol and provides an implementation.

[32] In this case, IP address to name lookups, commonly called “reverse” or “pointer” lookups.

[33] For example, http://www.kb.cert.org/vuls/id/844360

Mitigating DNS Threats

The DNS has a number of features that make it resilient to many forms of disruption and attack. Some of these features were part of the original design, while others have been added over time. For example, recent additions to the DNS such as "DNS Security Extension" (known as DNSSEC) and operational practices such as the deployment of "Anycast" [34]provide increased protection against many of the most common threats to the DNS. This section will explore these and other ways to mitigate threats associated with the DNS.

Denial of Service Mitigations

The impact of Denial of Service (DoS) attacks are primarily reduced in two ways, hardening and resource distribution. This is true whether the attacks are intentional or accidental.

Mitigating DoS through Hardening

"Hardening" of systems and networks involves provisioning against resource depletion attacks (sometimes called "over-provisioning") and increasing the protection that surrounds systems. For example, if the maximum anticipated DNS query load for a particular service is 100 queries per second, then designers might choose to over-provision systems and networks to support a load ten to one-hundred times higher. This over-provisioning would provide a level of protection against a resource degradation DoS attack. Similarly, if a single fiber provides the connectivity for a set of DNS servers, hardening that fiber by encasing it in concrete conduit would offer some protection against a resource disruption DoS attack, e.g., that fiber being cut.

Hardening also involves putting protective measures around DNS servers to make them less susceptible to known attacks. For example, securely configuring operating systems and applications according to best practices, filtering malformed or suspicious DNS traffic, and deploying real-time event monitoring and Intrusion Prevention Systems.

[34] http://www.ietf.org/rfc/rfc4786.txt

Mitigating DoS through Distribution

Distribution replicates facilities, systems, or services to multiple physical locations to reduce the likelihood of overall system failure. For example,

- diverse physical locations will reduce the impact of a fire or natural hazards such as flooding;
- redundancy in electricity supply can be achieved using independent power sources and distribution facilities, in combination with uninterruptible power supplies (e.g., batteries and on-site generators); and
- redundancy in telecommunications can be achieved using multiple independent facilities, such as multiple terrestrial and wireless providers;

Distribution avoids single points of failure. This strategy is most effective when the elements of the distributed infrastructure are independent of each other. If the distributed systems have some dependency, then the failure of one element may trigger failures of other elements, thereby increasing the overall risk of failure.

Designing a DoS-resistant DNS Infrastructure

Both authoritative DNS servers and DNS resolvers are subject to denial of service attacks. Mitigating DoS attacks against DNS servers and resolvers uses three main techniques: over-provisioning, geographical distribution, and the use of “Anycast”. The DNS commonly uses a connectionless protocol (UDP[35]) for most lookups and is thus particularly well suited to all three mitigations.

The DNS specifications support replication of the DNS service, including multiple name servers and a hierarchy of authoritative servers. It is easy to replicate authoritative DNS servers and DNS resolvers across multiple machines in geographically distributed areas.

Anycast allows for DNS query load to be distributed across a number of servers with each query being sent to the server that is network-topologically closest[36] to the source of the query. During a DoS attack, Anycast localizes the load of DNS queries to the servers closest to the attacker. While this increases the likelihood that those close servers will be overwhelmed, servers further away will receive fewer queries, thereby increasing the

[35] http://www.ietf.org/rfc/rfc0768.txt

[36] “Network-topologically closest” refers to the routing graph connecting the source to the destination, that is, the smallest set of networks that are needed to get a packet from the sender to the receiver. This path may result in the selection of a destination that is not geographically closest. In Anycast terms, “closer” and “farther” are relative to the network topology, rather than geographic topology.

chances that the DNS service as a whole will be able to continue to respond to queries.

Another helpful technique to mitigate DoS attacks is to use different implementations of DNS server software. In some cases, specially crafted packets[37] have been able to trigger bugs that have crashed DNS servers, effectively disabling parts of the DNS infrastructure. The use of multiple (independent) implementations of DNS server software can mitigate this type of DoS attack, as it is unlikely that independent developers will code the same bug.

Another mitigation strategy for DoS against DNS infrastructure is the use of outsourced DNS providers. These providers typically have hardened and geographically distributed infrastructure and are more able to withstand DoS attacks. Use of service providers does have a downside, however, as multiple DoS attacks, each targeting a different customer, could overload the provider, resulting in a DoS to customers that weren't under direct attack.

Protecting Management and Administrative Infrastructure against DoS attacks

"Management infrastructure" includes the facilities that allow for the creation, modification, and deletion of domain names outside of the DNS system itself. This includes the registries that maintain the zone files and the registrars that request modifications to the zone file on behalf of domain name holders (known as "registrants"). A successful DoS attack against the DNS management infrastructure would result in registrants being unable to manage domain names or registrars being unable to request modifications of the registries.

Mitigating management infrastructure DoS attacks is more challenging than mitigating the DNS infrastructure DoS attacks. The management infrastructure typically makes use of the connection-oriented TCP protocol for its web servers, "Whois"[38] servers, and Extensible Provisioning Protocol (EPP)[39] servers. Anycase is not always compatible with TCP protocol, which means that more complex and expensive mitigations must be used.

Well-funded registries and registrars generally do have the tools, people and systems to cope with this problem. However, not every registry and registrar is well funded, and there is a wide gap in mitigation capabilities between the top and bottom. In part, this is a consequence of economies of scale and other financial considerations. In other cases, informal best-effort arrangements are considered reasonable and proportionate to meet the needs of the stakeholders.

[37] These are sometimes called "Packets of Death" because a single packet can (figuratively) kill a DNS server.
[38] http://tools.ietf.org/html/rfc3912
[39] http://www.faqs.org/rfcs/rfc3730.html

The functions of management infrastructure are typically not time critical and can occur "out of band" of the Internet. This means that use of phones, emails, or FAXes can mitigate management infrastructure DoS attacks. To affect the management infrastructure, any DoS would have to be sustained for a longer period of time, providing the opportunity to use other mitigation techniques, such as working with network infrastructure providers to reduce or stop DoS traffic. However vulnerabilities in registrar infrastructure may well become more time-critical as DNSSEC deployment increases. In an emergency, registrants may need to change their DNSSEC keys and have this reflected in the DNS almost immediately, which would require time critical response on the part of registrars and registries.

Administrative issues also present a number of potential infrastructure weaknesses. For example, it is quite common for multiple entities to cooperate to provide DNS service (such as with primary and secondary DNS servers) without written agreements or service level commitments. When primary and secondary DNS are only loosely coupled, out-of-date information can make it difficult to contact responsible parties or have both operators use the same change windows to patch software, while changes by server operators could inadvertently introduce single points of failure.

Organizational administrative issues can cause a self-inflicted Denial of Service. For example, service may be disrupted if invoices are not paid on time or if payments are not processed. Since these invoices are usually for small amounts of money (typically a few tens of US dollars) they may not get the attention they deserve by the relevant finance departments. In 2003, Microsoft failed to renew hotmail.co.uk, following a similar mistake by the company in 1999 that resulted in its Hotmail service being degraded.[40] A lack of documentation, business continuity and disaster recovery planning, critical staff knowledge, or authentication credentials may also cause self-inflicted DoS.

These types of administrative DoS issues can be mitigated through established best practices, such as proper service documentation, establishment of service level agreements, hot standby services, failover procedures, and regular drills or exercises. There is nothing special about DNS in this regard: the same principles that apply to other critical IT systems apply to an organization's DNS infrastructure.

[40] http://www.theregister.co.uk/2003/11/06/microsoft_forgets_to_renew_hotmail/

Summary on Mitigating Denial of Service Threats

DoS attacks against the DNS are likely to increase in frequency for several reasons:

- tools to implement DoS attacks are easier for attackers to find and use;
- the various systems that provide the DNS are becoming more and more interconnected; and
- the DNS as a whole is becoming more complex and hence more fragile.

Intentional DoS attacks are especially challenging when they are actually Distributed DoS (DDoS) attacks. The resources that a DDoS attacker can bring to bear during their attack may include thousands or tens of thousands of compromised machines

The mitigations for all forms of DoS attacks follow the same prescription:

- Hardening: over-provision the infrastructure to withstand attacks;
- Distribution: spread the infrastructure out geographically, using independent facilities, systems, Anycast, and other technologies to remove the disruptable choke points; and
- Best Practices: utilize operational best practices that apply to other core IT systems, with well-documented and mature processes and procedures that are reviewed and updated regularly.

These mitigations, when properly applied to the multiple technical, management, and administrative layers of the DNS, can greatly reduce the impact of DoS attacks.

Data Corruption Mitigations

DNS data may be corrupted several different ways, including:

- Impersonation of the authority and corruption of the data at its source
- Corruption of the data on the authoritative server
- Corruption of the data within a DNS cache
- Corruption of the data in transit, either between the authority and the database, between the authoritative server and a DNS resolver, or between a resolver and the requesting application.

In this section, we will discuss how to mitigate the risk of these various forms of corruption.

Authority Impersonation

When an unauthorized entity creates, modifies, or deletes DNS data, this is “Authority Impersonation.” For example, if an attacker were able to obtain a registrant’s credentials at a registrar or DNS service provider, the attacker would then be able make any sort of DNS change desired. Credential theft isn’t required in every case. The domain sex.com[41] was stolen when an attacker convinced a registrar they represented the owner. This form of attack is relatively common on the Internet today and is typically implemented using social engineering (e.g., phishing)[42] or by attacks against a registration portal[43].

Mitigations for this form of Data Corruption would include increased vigilance or monitoring of the contents of a domain by the domain owner (i.e., verifying that the resource records within the domain are correct) as well as increased security by the registrar (e.g., the use of strong or two-factor authentication).

The routing system can also be a vector for authority impersonation if the IP addresses for authoritative servers are hijacked and redirected to another machine.

The most effective mitigation of this form of authority impersonation data corruption would be additional security in the routing system upon which the DNS relies. Efforts at the IETF [44]and within the Regional Internet Registries[45] may provide the tools and underlying infrastructure to secure the routing system in the future.

Deployment of DNSSEC also would make it infeasible for IP address hijackers to present corrupt data. DNSSEC-compliant resolvers check cryptographic signatures to detect unauthorized changes to DNS data. Because improperly signed data cannot be substituted by an IP address hijacker, DNSSEC-signed zone data changes the threat of IP address hijacking from “data corruption” to “denial of service”.

[41] http://www.circleid.com/posts/to_fight_domain_name_theft_sexcom_gives_birth_to_a_new_property_right/

[42] http://www.icann.org/en/committees/security/sac028.pdf

[43] http://www.icann.org/en/committees/security/sac040.pdf

[44] The Secure Inter-Domain Routing Working Group, see https://datatracker.ietf.org/wg/sidr/charter/

[45] Development of the Resource Public Key Infrastructure, see e.g., http://www.apnic.net/services/services-apnicprovides/resource-certification/RPKI

Authoritative Server Corruption

The DNS has a concept of a primary (also known as a master) server that is the authoritative source of all the information for a particular domain. Other servers copy the data from the primary authoritative server. A corruption on the primary server is called "authoritative server corruption." This corruption can be the result of malicious or accidental activity (such as disk errors). For example, the German zone .DE was corrupted in 2010 by accidental truncation (shortening) of the zone file.[46]

Zone monitoring is an effective way to mitigate authoritative server data corruption. For example, zone generation software can append a sentinel to the end of a zone. This provides a mechanism for the DNS software to detect data corruption prior to publication when the sentinel is not found.[47]

Cache Corruption

DNS resolvers cache DNS information to speed performance and reduce network load. Cache corruption inserts erroneous data in the resolver's cache to be handed out in response to subsequent queries. One of the earliest well-publicized cache corruption attacks against the DNS inserted an unauthorized address (the address of "ALTERNIC.NET") for the domain name "INTERNIC.NET."[48] While the bug that facilitated the ALTERNIC.NET cache corruption was addressed, the release of a technique for cache corruption by Dan Kaminsky[49] increased the necessity for a more fundamental fix offered by DNSSEC.

DNSSEC addresses cache corruption in resolvers by allowing resolvers to detect any unauthorized modification of DNSSEC-protected data and discard the data instead of inserting it into its cache.

Application to Resolver Corruption

Data fetched by a resolver must be returned to the application that has requested the DNS lookup. Since DNSSEC coverage stops at the resolver, the path between the application and the DNS resolver is subject to corruption that cannot be mitigated through the use of DNSSEC.

[46] http://www.denic.de/en/denic-in-dialogue/news/2731.html?cHash=ed9220a2f040569a255b97b88141b358

[47] VeriSign used to insert a record in the root zone with the name "vrsn-end-of-zone-marker-dummy-record.root." for exactly this purpose.

[48] http://news.cnet.com/2100-1033-201382.html

[49] http://news.cnet.com/8301-1009_3-10009827-83.html

Mitigation of Application to Resolver DNS data corruption requires protecting the communication channel between the application and the resolver. One form of protection would be to co-locate the application and the DNS resolver on the same machine, something that is increasingly feasible as machines become less constrained in terms of CPU power, memory, and network connectivity. However, a downside of this mitigation is the reduction in site-wide cache hit rate (since caches would no longer be shared among multiple end users), resulting in an increased burden on authoritative servers, particularly those at higher layers in the name hierarchy such as the root servers. This downside can be partially ameliorated by the use of "forwarding caches", that is, creating a multi-layer caching hierarchy in which individual machine caches query in a site-wide cache. These mitigations help to ensure that DNSSEC operates fully end-to-end, giving the end-user a significant amount of information about the trust level of DNS information as they browse the Internet or run other Internet applications. However, these architectural changes to the design of LANs also increase the complexity of the DNS lookup mechanisms that could impact overall system resiliency. When considering these issues, readers should consider the level of risk, compared to the level of effort of the mitigation, and the potential for downstream costs and potential system failures

In many cases the network over which applications make queries is trusted, such as a corporate LAN. In these cases, firewalls that disallow DNS responses being sent from external networks are generally sufficient to protect the application to resolver communications channel. Where the network is untrusted, techniques such as the use of IPSec[50] or Transaction Signature (TSIG)[51] could be used, although the effort required may be disproportionate to the benefit.

System Corruption

System corruption occurs when one or more systems used in DNS services are compromised, allowing for the data to be altered. The table below summarizes the main opportunities, and mitigations.

[50] http://tools.ietf.org/html/rfc4301
[51] http://www.ietf.org/rfc/rfc2845.txt
[52] http://www.citi.umich.edu/u/provos/papers/ndss08_dns.pdf
[53] http://www.kb.cert.org/vuls/id/844360
[54] For example, http://www.cert.org/advisories/CA-2002-31.html

Type of Corruption	Description	Mitigation
Stub Resolver Corruption	Altering the operation of the resolver libraries or configuration upon which applications depend. For example, the "Corrupted DNS Resolution Paths" attack[52] changes Microsoft Windows registry settings for the resolver applications use to do lookups. Another form of this type of corruption is when the stub resolver itself can be corrupted.[53]	A key mitigation technique is to ensure that the stub resolver in use and the network path to it is trusted. Verifying that the IP addresses specified for DNS resolution services (in /etc/resolv.conf on Unix-related systems and in the registry in Windows systems) are expected values is important. DNSSEC mitigates some of corrupted resolution paths if the application performs the DNSSEC validation itself (discussed above in "Application to Resolver Corruption")
Caching Resolver Corruption	Specially crafted responses to queries and unsolicited DNS messages have triggered bugs in DNS resolver implementations[54]. In some cases, these bugs have allowed attackers to gain control of the server software or the server itself.	Keeping critical system and application software up to date by applying security patches and system updates in a timely fashion is important to ensure those systems are not subject to known attacks. Another mitigation is separation of authoritative name service from caching resolution service. While most name server implementations allow a server to act as both a caching resolver and an authoritative server simultaneously, this has led to various forms of data corruption such as cache poisoning or inappropriate response to authoritative queries. Separating these two functions is considered a best practice.
Intermediate System ("Middlebox") Corruption	As DNS messages transit the network, they traverse intermediate devices such as routers, switches, and firewalls. These systems are subject to attack and DNS data that crosses through them can be corrupted. An example of this attack is "Drive-By Pharming"[55] in which an attacker causes a customer-premises device such as a broadband router to provide the IP address of an attacker's machine for the customer's DNS server.	Middleboxes should be considered critical system and be updated in a timely fashion. DNSSEC addresses this form of corruption directly. With a DNSSEC-signed response, any modification of that response in transit will be detectable by the validating resolver. Unfortunately, the use of DNSSEC as a mitigation for Intermediate System Corruption is hampered by middleboxes that inhibit used of DNSSEC by incorrectly handling requests that ask for DNSSEC-signed responses or the responses themselves.[56]
Authoritative Server Corruption	This is similar to caching resolver corruption, where specially crafted DNS messages can trigger bugs in DNS server software[57].	In addition to mitigations previously discussed, it is important to ensure the DNSSEC keying material is managed to minimize the chance that an attacker can steal keying material and impersonate an authoritative server. Best practices for DNSSEC deployment use "offline signing keys" minimizing the possibility of key theft.

[55] http://www.symantec.com/avcenter/reference/Driveby_Pharming.pdf
[56] http://www.icann.org/en/committees/security/sac035.pdf
[57] For example, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-1910

Protocol Corruption

Protocol Corruption takes advantage of limitations or vulnerabilities in the DNS protocol itself to corrupt DNS data. There are three broad categories of Protocol Corruption:

Query Prediction: as discussed in "DNS Threat Analysis"[58], every message in the DNS has a 16-bit query identifier that is used to match responses to queries. In combination with the 16-bit source port, this gives a total of 32 bits to uniquely identify a DNS transaction between any given source and destination. As early as 1986, it was recognized that this provided only a weak defense against injection of bogus responses by a malicious third party. In particular, it is possible to take advantage of "the Birthday Paradox"[59], to predict a query identifier and then inject a response[60].

Man-in-the-Middle: DNS traffic is not encrypted. An attacker with control over the intermediate network can implement a variety of man-in-the-middle attacks.

Cache Poisoning: predicting queries and man-in-the-middle attacks allow for an attacker to insert bogus data into a cache, an attack known as cache poisoning, discussed in detail above.

The key mitigation for all of these protocol corruption attacks is DNSSEC, which allows a security-aware resolver to verify that the DNS data have not been modified in flight. With this capability, it no longer matters that an attacker can predict query identifiers, can sit in the middle of a DNS transaction, or can attempt to poison the cache since any attempt to modify the DNS data will be detectable.

Summary on Mitigating Data Corruption

The most important mitigation for data corruption is DNSSEC, since the primary reason for the creation of DNSSEC is to ensure that published DNS data are not corrupted.

Where DNSSEC does not apply, specifically before the data is DNSSEC-signed or after it has been validated, other mitigation techniques are necessary, such as increased vigilance of system resolvers and ensuring that critical systems have all security patches applied and are kept up to date.

[58] http://tools.ietf.org/html/rfc3833
[59] http://www.howstuffworks.com/question261.htm
[60] http://www.secureworks.com/research/articles/dns-cache-poisoning

Information Exposure Mitigations

There is a growing expectation that individuals should be able to use the DNS (and related WHOIS protocol) anonymously. The goal would be that individuals could perform DNS queries without their requests being observed, aggregated and correlated with their identity[61].

Information Exposures are unauthorized releases of personal and other data associated with the DNS and DNS queries. These exposures can occur at both the administrative level, e.g., “Domain Front Running”, as well as at the network and system level with attacks known as “Cache Snooping” and “Zone Walking”. The latter, “Zone Walking”, which depends on DNSSEC, will likely become more common as DNSSEC sees greater deployment. A general description of Information Exposure threats and mitigations is discussed below.

[61] The term “identity” here refers to a combination of attributes, assertions, or other observable traits and behaviors (e.g. online transactions) that can reasonably identify a natural person.

Type of Information Exposure	Potential Mitigations
Domain Front Running: before registering a domain, potential buyers often use "WHOIS" tools to discover which domains are still available. Front Runners, it is believed[62], gather this information and rush to register domains before the original potential buyer, hoping for a quick profit by re-selling the domain.	Contractual in nature. ICANN could enforce and the registries and registrars abide by restrictions prohibiting such behavior.
Cache Snooping: an unauthorized party can observe DNS data as it is placed in or requested from caching DNS servers, either because they have access to the servers or the underlying network.	As discussed in *"DNS Cache Snooping or Snooping the Cache for Fun and Profit"*[63], which analyzes Cache Spoofing attacks against caching resolvers, several mitigations can protect against cache snooping on the resolver side of the DNS: 1. Access to caches should be limited to trusted clients. 2. DNS caching resolvers should be used that can be configured to ignore queries in which the "recursion desired" bit is not set. 3. DNS caching resolvers that can be configured to randomly reduce the amount of time an entry is stored in the cache by a small amount should be used.
Zone Walking: a feature of DNSSEC, specifically NSEC RRs, can be used to enumerate the contents of a DNSSEC-signed zone. Most authoritative name servers deny zone transfers to all but authorized requesters. However, as a side effect of the operation of DNSSEC, it has become easy to obtain the contents of a zone, even when the owner desires to deny zone transfers.	A purpose-designed DNSSEC protocol enhancement, NSEC3, can be used to mitigate this risk and make it very difficult to determine the contents of the zone. It is always possible to guess names by brute force, albeit at the cost of a very large number of queries[64],[65].
DNS Query Tracking: DNS queries made by an individual are visible to their ISP. These queries may contain personal data, revealing information about individuals and the sites they visit. Further, as there is increasing interest in various "do-not-track" proposals that intend to limit traditional Web-based tracking (e.g. cookies), it is likely that DNS queries may become the new target.[66]	Mitigation of unwarranted DNS query tracking can be undertaken via general system security and law. Effective physical security (preventing access to the servers from external attackers as well as from unauthorized internal access) and network security minimize the opportunities for the DNS and DNS queries being compromised. A legal framework, however, needs to be in place to ensure DNS operators are not undermining the trust in the system.
NXDOMAIN Redirection: When a DNS server is unable to resolve a domain name from a query into a known address, it can result in a "non-existent domain" response, called an NXDOMAIN reply. Many ISPs have started to intercept and replace the NXDOMAIN replies to redirect the client to another domain they control.[67] This redirection could be harmless, or could be used to deliver advertising or malware by replacing a trusted site.	This privacy threat is largely under the control of the ISP, and the mitigation is primarily legal. ISPs need to understand the potential threat to privacy and take appropriate steps to protect it. If an ISP decides to leverage NXDOMAIN Redirection, they should ensure they understand the ramifications of any advertising displayed (e.g. behavioral targeting tools are disabled) and take appropriate technical precautions to ensure privacy is not compromised.

[62] http://www.icann.org/en/committees/security/sac022.pdf

[63] http://www.rootsecure.net/content/downloads/pdf/dns_cache_snooping.pdf

[64] http://cr.yp.to/talks/2009.08.10/slides.pdf

[65] It should also be noted that tools such as "Phreebird" by Dan Kaminsky (http://dankaminsky.com/phreebird/) can alleviate the risk inherent in offline NSEC3 signing, albeit since the DNS database is public, names within it will always be susceptible to brute force attacks.

[66] http://tools.ietf.org/html/draft-cooper-web-tracking-opt-outs-00

[67] http://www.icann.org/en/topics/new-gtlds/nxdomain-substitution-harms-24nov09-en.pdf

Summary of Information Exposure Mitigations

Information Exposure threats are ways in which information relating to the DNS can leak out, potentially benefitting those who would use that information in inappropriate ways. In some cases, such as using NSEC3 instead of NSEC to combat Zone Walking, mitigations can be extremely effective. In other cases, such as protecting against Cache Snooping related to authoritative servers, mitigation techniques are limited to those used for the general protection of data including secure data paths and methods to protect against system compromise.

Mitigating Other Threats to the DNS

While the threats of ossification and alternate roots are lower than many of the previously described threats, mitigations do exist.

Mitigating Ossification

Increased conformance to the DNS protocol standards would help mitigate DNS ossification. As the Internet has grown, the DNS has been modified to adjust to that growth. However, many DNS implementations have been lax at conforming to the standards that did exist, much less newer standards. For example, deployment of DNSSEC is dependent upon support of new protocol extensions, but we have found that common network infrastructure equipment that is DNS-aware does not support the protocol extensions, making use of DNSSEC impossible in environments using this equipment. For example, an ICANN report investigating common broadband routers used for home Internet connections found that many devices had very poor compliance to existing DNS standards.[68]

Mitigating Alternative Roots

The threat of alternative roots waxes and wanes over time, usually for political, rather than technical, reasons. If the Internet community addressed the underlying causes that make alternative roots useful, this would mitigate the threat. For example, one of the drivers for alternative roots has been the desire to use domain names with local character sets, such as Cyrillic characters. However, with ICANN’s deployment of International Domain Names (IDNs) beginning in 2007, the pressure for alternative roots to support IDNs was significantly lessened. New generic top-level domain names (gTLDs) have also helped to mitigate this threat.

[68] SSAC035, see also SSAC17, and SSAC18, available from http://www.icann.org/en/committees/security/ssacdocuments.htm.

DNS Amplification Attack Mitigations

DNS amplification attacks use the DNS to amplify the power of an intruder to perform a Denial of Service attack on a third party. Mitigating amplification attacks requires action on the part of the DNS server operator.

Authoritative Amplification is when an authoritative server is used to reflect traffic to a target. Resolver Amplification is when an attacker reflects DNS traffic against a resolver instead of an authoritative server. For example, if an attacker sends a 28-byte query to either type of DNS server for the NS records associated with the root servers with DNS-SEC enabled, the response will be over 800 bytes. The attacker also spoofs the source address of the query so that the amplified response will be sent to the target, performing a Denial of Service attack on the target.

The challenge in mitigating amplification attacks is that attack traffic is usually indistinguishable from regular DNS traffic, since an amplification attack query is indistinguishable from a normal query.

Arguably the best mitigation of Amplification attacks would be the ubiquitous implementation of "Network Ingress Filtering,"[69] which would block the attacker from spoofing the source address of the query.

However, since Network Ingress Filtering is not universally deployed, other mitigations are needed. For example, servers could rate-limit responses so that a specific IP address will only receive a limited number of responses (or amount of traffic) per time period regardless of the number of queries received.[70] Some DNS service providers operating resolvers, such as Google[71], already implement rate limiting.

Internet Service Providers could manually block queries from source addresses that are the target of attack. Unfortunately, this mitigation also creates a denial of service to the target, although because of filtering rather than resource starvation.

Resolver Amplification attacks can also be mitigated to a great extent by limiting resolver queries to trusted clients, e.g., clients on the local LAN or within a site's administrative boundaries. Historically, DNS resolvers by default responded to all DNS queries regardless of their source address. Resolvers configured this way are referred to as "Open

[69] http://tools.ietf.org/html/bcp38

[70] This particular mitigation has the additional benefit of protecting the bandwidth of the server, if an aggressive client accidentally or maliciously floods the server with queries.

[71] See the section on "Rate Limiting Queries" at http://code.google.com/speed/public-dns/docs/security.html

Resolvers". While the practice of having Open Resolvers is discouraged[72], according to the latest open resolver survey performed by The Measurement Factory[73], there are over 190,000 Open Resolvers on the Internet. Recent versions of BIND, one of the most widely used DNS resolving server software packages, no longer behave as an Open Resolver by default, instead only providing recursive service to the IP addresses of the local networks.

Summary of DNS Amplification Attack Mitigations

DNS Amplification attacks make use of either authoritative servers or resolvers to reflect data towards a target. Both forms of attack can be mitigated at the source by implementing "Network Ingress Filtering" however this has proven to be difficult: it has been over a decade since Network Ingress Filtering was declared a Best Current Practice, yet implementation is lagging. In contrast to Authoritative Amplification attacks, Resolver Amplification attacks can be mitigated at the reflection point by ensuring resolvers are not open, that is, that they only respond to queries from known and trusted hosts, ideally on the same local area network as the resolver itself and that the resolvers rate limit responses on a per-client IP address basis.

Mitigating the Threat of Fast Flux DNS

The first step in mitigating Fast Flux is detecting it. This can be very difficult as the entire point of Fast Flux is to make use of the DNS to minimize chances of detection. As described by The Honeynet Project[74]:

> The detection of domain names being served by a fast-flux service network depends upon multiple analytical passes over DNS query results, with increasing flux detection accuracy gained by employing a scoring mechanism to evaluate multiple relatively short lived DNS records, taking into account including the number of A records returned per query, the number of NS records returned, the diversity of unrelated networks represented and the presence of broadband or dialup networks in every result set.

The Honeynet Project lists some steps that can be taken to most realistically mitigate Fast Flux DNS, including blocking connections from the Internet to end-user systems on ports 80 (WWW) and 53 (DNS) and to Fast Flux DNS controllers, improving domain registrar procedures including auditing to reduce fraud, and increasing service provider awareness to foster understanding of the threat and to share processes and knowledge.

[72] http://www.ietf.org/rfc/rfc5358.txt
[73] http://dns.measurement-factory.com/surveys/openresolvers/ASN-reports/latest.html
[74] http://www.honeynet.org/node/144

Mitigating DNS as a Covert Channel

As with Fast Flux DNS attacks, mitigation of covert channel use of DNS requires detection as a first step. For example, monitoring may be able to detect unusual use patterns, such as a large number of TXT responses being sent to a particular server.

Monitoring and analysis of DNS traffic within an enterprise can provide a baseline to make covert channel DNS stand out and be identified.

In cases where the DNS is being used as a covert channel for botnet command and control systems, mitigation typically depends on analysis of the botnet code or behavior to establish the domains a compromised machine will use to synchronize with the "botnet herder". In several cases, botnets were effectively disabled when the domain names used by the botnet were identified and redirected away from malicious parties[75],[76].

[75] http://www.confickerworkinggroup.org/wiki/pmwiki.php/TLD/TLDOperators

[76] http://newhaven.fbi.gov/dojpressrel/pressrel11/pdf/nh041311_2.pdf

Recommendations

The DNS is a critical component of the Internet and will continue to be subject to accidental disruption and malicious attack. However, due to its design and the continued evolution of DNS protocols, systems, and operational practices many of the potential threats to the smooth operation of the DNS can be mitigated.

The DNS has evolved to meet the changing requirements of the Internet. As the Internet has grown, the need for security, stability, and resilience in the DNS has also grown. The risks both to and from the DNS have become more apparent. Risks to the DNS include Denial of Service, Data Corruption, and Information Exposures and risks from the DNS include DNS Amplification and the use of the DNS as a covert channel for communications.

Fortunately, mitigating techniques, summarized below, can reduce the risks effectively and allow for the continued secure, stable, and resilient operation of this core component of the Internet.

Denial of Service

- Avoid single points of failure within any of the various infrastructures, systems, implementations, and facilities providing or supporting DNS services
- Hardening:
  - Limit, as much as possible, potential bottlenecks for resources such as CPU, memory, network bandwidth, disk bandwidth, etc.
  - Over-provision systems and services beyond anticipated worst-case load
  - Physically protect systems from attack or accident
- Distribution
  - Replicate facilities, systems, and services in multiple physical locations and using independent infrastructures
- Use multiple independent implementations for systems and services
- Use “Anycast” routing for DNS to reduce the risk that a denial of service (accidental or malicious) will completely disable the DNS

Data Corruption

- Increase vigilance to ensure corruptions are detected
- Use DNSSEC to protect data in transit and in storage, reducing the risk of cache poisoning and man-in-the-middle attacks
- Ensure all systems have security patches applied, are up-to-date and are configured using best practices

Information Exposure

- Secure the transit path of DNS messages as much as is feasible
- Restrict domain name zone transfers to authorized parties
- Use NSEC3 in DNSSEC to prevent Zone Walking

DNS Amplification

- Follow Network ingress filtering recommended best practices
- Rate limiting
  - Limit ingress of packets to protect your infrastructure
  - Limit egress of packets to protect your neighbors
- Limit the source addresses that can query resolvers to those within specified networks

DNS as a Covert Channel

- Monitor for DNS traffic that deviates significantly from baseline traffic
- Detect and disable (or redirect) domains used for botnet command and control synchronization

Appendix A – DNS Enhancements since RFC 1034/1035

The following table provides a list of changes to DNS protocol standards or operational conventions since the publication of RFCs 1034 and 1035.

Date	RFC	Modification
Apr 1989	1101	Described how to encode network names in the DNS
Oct 1989	1122	Provided requirements for the communication layers of Internet hosts
Oct 1989	1123	Provided requirements for applications and support systems on Internet hosts
Oct 1990	1183	Defined the AFSDB, RP, X25, ISDN, and RT RRs
Jul 1992	1348	Defined the NSAP and NSAP-PTR RRs for looking up OSI CLNP addresses
Jun 1993	1386	Defined the structure for the .US domain
Jan 1993	1401	IAB recommendation to DISA to discontinue the use of HOSTS.TXT
May 1993	1464	Discussed how to store arbitrary strings in the DNS
Jun 1993	1480	Revised the structure of the .US domain
Oct 1993	1535	Recommended revision of common resolver search heuristics
Oct 1993	1536	Described and recommended fixes for common DNS implementation errors
Oct 1993	1537	Described and recommended fixes for common DNS configuration errors
Mar 1994	1591	Provided the basis for DNS systems structure and delegations
May 1994	1611	Defined management parameters for authoritative DNS servers
May 1994	1612	Defined management parameters for DNS resolvers
Jun 1994	1637	Revised the definition of the NSAP RR
Aug 1994	1664	Defined the PX RR for use in mapping between X.400 and Internet e-mail

Date	RFC	Modification
Oct 1994	1706	Revised the NSAP RR again
Nov 1994	1712	Defined the GPOS RR for geographical location enabling the encoding of latitude, longitude and altitude
Nov 1994	1713	Provided an overview of DNS debugging tools
Apr 1995	1794	Described using the DNS for load balancing
Jun 1995	1811	Defined the structure of the .GOV domain
Aug 1995	1816	Redefined the structure of the .GOV domain
Jan 1996	1876	Defined the LOC RR for geographical location and size of an object with a resolution of centimeters.
Dec 1995	1886	Provided DNS support for IPv6, defining the AAAA RR and the IP6.INT domain.
Feb 1996	1912	Described and recommended additional fixes for common DNS operational and configuration errors.
Jun 1996	1956	Defined the structure of the .MIL domain
Aug 1996	1982	Described serial number arithmetic as used in the DNS
Aug 1996	1995	Defined the incremental zone transfer protocol extension, allowing for only changes in a zone to be transferred instead of the full zone.
Aug 1996	1996	Defined the NOTIFY protocol extension that permitted master servers to inform slave servers that zone contents had changed.
Oct 1996	2010	Described operational criteria for root name servers.
Oct 1996	2052	Defined the SRV RR that allowed for the lookup of protocol services in the DNS
Oct 1996	2053	Defined the procedures for registration in the .AM domain
Jan 1997	2065	First attempt at defining DNS security enhancements (DNSSEC) that would allow for data integrity and authenticity assurance
Apr 1997	2136	Defined mechanisms to allow for dynamically updating a zone
Apr 1997	2137	Defined mechanisms to secure dynamic updates defined in RFC 2136

Date	RFC	Modification
May 1997	2146	Redefined the registration procedures in the .GOV domain and proposed the first steps of migration to .FED.US
Jan 1998	2163	Revised the methods used to map between X.400 and Internet e-mail and the use of the PX RR
Jun 1997	2168	Specified how to resolve uniform resource identifiers using the DNS
Jul 1997	2181	Clarified numerous ambiguities in the DNS specifications
Jul 1997	2182	Provided recommendations on the selection and operation of secondary servers
Oct 1997	2219	Suggested conventions for aliasing services for sites, e.g., the use of “www” in a domain name to represent HTTP service
Nov 1997	2230	Defined the Key Exchange (KX) RR used to provide key delegation for the use in the IP Security protocols.
Nov 1997	2240	Discussed a legal basis for domain name allocation
Jan 1998	2247	Described how to use domains in LDAP/X.500 so that LDAP can contain DNS information
Mar 1998	2308	Defined enhancements to the DNS to make negative caching of names more effective
Mar 1998	2317	Explained a convention that would allow for “classless” addressing to be represented in the DNS reverse tree
May 1998	2352	Revised discussions of a legal basis for domain name allocation
Sep 1998	2377	Defined a naming plan for LDAP directories based on the top levels of the DNS
Mar 1999	2535	Revised specifications for the DNS security enhancements
Mar 1999	2536	Defined the use of DSA keys and signatures in the DNS
Mar 1999	2537	Defined the use of RSA/MD5 keys and signatures in the DNS
Mar 1999	2538	Defined how to store X.509 Certificates in the DNS
Mar 1999	2539	Defined how to store Diffie-Helman keys in the DNS

Date	RFC	Modification
Mar 1999	2540	Defined a format for archiving retrieved DNS data
Mar 1999	2541	Described operational considerations for using the DNS security enhancements
Jun 1999	2606	Listed reserved top-level domains
Aug 1999	2671	Defined an extension mechanism known as EDNS0 for the DNS protocol that allowed for connectionless DNS messages to be larger than 512 bytes
Aug 1999	2672	Described a protocol enhancement that would allow for the aliasing of an entire DNS tree, not just the last node in a tree and defined the DNAME RR
Aug 1999	2673	Allowed for the use of binary labels in the DNS
Sep 1999	2694	Discussed interactions between the DNS and network address translators (NATs)
Feb 2000	2782	Revision of the specification for the SRV RR
May 2000	2825	Discussed issues involved in the use of internationalized characters in the DNS
May 2000	2826	Explained the need for a single root in the DNS name space.
May 2000	2845	Defined a way of authenticating DNS transactions with shared keys and defined the TSIG RR
Jun 2000	2860	Placed on record the text of the MoU between ICANN and the IETF regarding IANA-related work
Jun 2000	2870	Revised the root server operational requirements
Jul 2000	2874	Standardized mechanisms to support IPv6 address aggregation and the A6 RR
Sep 2000	2915	Defined the Naming Authority Pointer (NAPTR) RR for use in looking up naming authorities
Sep 2000	2916	Provided a way of mapping telephone number (E.164 addresses) into the DNS using NAPTR RRs
Sep 2000	2929	Discussed IANA actions relating to the DNS

Date	RFC	Modification
Sep 2000	2930	Described a protocol enhancement to allow for sharing of (private) secret keys and the TKEY RR
Sep 2000	2931	Standardized a mechanism to provide for transaction signatures using public keys via DNSSEC
Nov 2000	3007	Revised the protocol to do dynamic updates in a secure fashion
Nov 2000	3008	Revised the way signing authority was done in DNSSEC
Feb 2001	3071	Commented on how ccTLD were operated
Mar 2001	3090	Clarified the status of whether zones are secure or not in DNSSEC
May 2001	3110	Standardized how RSA/SHA-1 signatures and RSA keys are specified for use with DNSSEC
Jun 2001	3123	Defined a way to list Address Prefixes in the DNS and the APL RR
Jun 2001	3130	Discussed the state of DNSSEC
Sep 2001	3152	Discussed the need for the delegation of IP6.ARPA
Sep 2001	3172	Described the management and operation requirements for the “address and routing parameter area” (ARPA) domain
Nov 2001	3197	Deprecated the DNS management information base RFCs (1611 and 1612)
Dec 2001	3225	Specified a way in which a resolver could inform an authoritative server that the resolver understood DNSSEC-related RRs
Dec 2001	3226	Documented the DNS message size requirements to support DNSSEC and IPv6
Apr 2002	3258	Described how to use a shared unicast address to distribute authoritative name servers
Aug 2002	3363	Clarified how IPv6 addresses are to be represented in the DNS
Aug 2002	3364	Explained the pros and cons of the two ways of representing IPv6 addresses in the DNS
Oct 2002	3403	Defined the DNS portion of the Dynamic Delegation Discovery System

Date	RFC	Modification
Nov 2002	3425	Deprecated the IQUERY (Inverse Query) operation code
Dec 2002	3445	Clarified the KEY RR is to be used exclusively for DNSSEC purposes
Dec 2002	3454	Defined how to prepare strings in the DNS for internationalization
Feb 2003	3467	Offered an interpretation of the role of the DNS
Mar 2003	3490	Described how to internationalize domain names in applications (IDNA)
Mar 2003	3491	Defined how to prepare internationalized names for use in the DNS (NamePrep)
Mar 2003	3492	Defined how to encode Unicode strings into ASCII for use in the DNS
Oct 2003	3596	Standardized the DNS extensions used to support IPv6
Sep 2003	3597	Described how unknown DNS RRs are to be handled
Oct 2003	3645	Defined the Generic Security Service transaction signatures (GSS-TSIG)
Dec 2003	3646	Specified a way to allow DHCPv6 to set DNS configuration options
Nov 2003	3655	Clarified the mean of the “Authenticated Data” bit in the DNS protocol message header
Dec 2003	3658	Defined the Delegation Signer (DS) RR for creating the chain of trust in DNSSEC
Jan 2004	3681	Discussed the need for the delegation of E.F.F.3.IP6.ARPA in order to do reverse address-to-name mappings of 6Bone addresses
Apr 2004	3743	Provided guidelines for internationalized domain name registration and administration for Chinese, Japanese, and Korean names
May 2004	3755	Described how to deal with backward compatibility with DNSSEC-aware name servers, creating new RR types (DNSKEY, RRSIG, and NSEC)
Apr 2004	3757	Defined a bit in the DNSKEY RR to indicate whether the DNSKEY is to be used as a secure entry point, allowing for the separation of key signing keys and zone signing keys in DNSSEC.
Apr 2004	3761	Discussed the use of the DNS for identifying the services available at E.164 (telephone) numbers

Date	RFC	Modification
Apr 2004	3762	Registered a telephone number mapping service in the DNS for H.323
Apr 2004	3764	Registered a telephone number mapping service in the DNS for SIP
Jul 2004	3832	Described how to use DNS SRV RRs for remote service discovery with the Service Location Protocol
Aug 2004	3833	Provided the threat analysis of the DNS
Aug 2004	3845	Redefined how NSEC type bit map is to be used
Sep 2004	3901	Provided guidelines and best current practices on how the DNS should be operated in an environment with both IPv4 and IPv6 transports
Jan 2005	3958	Defined a generalized mechanism for application service naming using the DNS
Mar 2005	4025	Defined a method for storing IPSec keying material in the DNS
Apr 2005	4027	Specified the MIME media types used to describe DNS data
Mar 2005	4033	Provided an introduction to and requirements for the revised DNSSEC specifications
Mar 2005	4034	Defined the resource records used for the revised DNSSEC specification
Mar 2005	4035	Specified the DNS protocol modifications for the revised DNSSEC specification
May 2005	4074	Discussed common misbehaviors when authoritative name servers are queried for IPv6 records
Sep 2005	4183	Suggests a convention for using the DNS to determine the network that contains a specified IP address, network mask, and first-hop router
Oct 2005	4185	Discussed the motivations, mechanisms, and constraints of putting internationalized characters into the DNS
Jan 2006	4255	Specified the method for putting “Secure Shell” (SSH) key fingerprints into the DNS and defined the SSHFP RR
Dec 2005	4310	Defined extensions for the Extensible Provisioning Protocol to support DNSSEC
Feb 2006	4339	Described approaches used by IPv6 hosts to configure DNS information

Date	RFC	Modification
Jan 2006	4343	Clarified DNS case insensitivity
Feb 2006	4367	Discussed some of the false assumptions made about DNS names
Mar 2006	4398	Redefined how cryptographic public keys are published in the DNS
Feb 2006	4431	Specified a way in which DNSSEC “islands of trust” can be looked up securely and defined the DLV RR
Apr 2006	4470	Described how to create NSEC RRs on demand to effectively stop zone walking
Sep 2006	4471	Proposed two methods for deriving the predecessor and successor of a DNS name for NSEC record synthesis
Apr 2006	4472	Summarized operational IPv6 DNS considerations
May 2006	4501	Defined Uniform Resource Identifiers for Domain Name resources
May 2006	4509	Specified how to use the SHA-256 digest type in DS RRs for DNSSEC
Jul 2006	4592	Clarified the role of wildcards in the DNS
Aug 2006	4635	Specified the HMAC SHA transaction signature algorithm identifiers
Sep 2006	4641	Discussed DNSSEC operational practices
Sep 2006	4690	Provided recommendations for the definition of internationalized domain names (IDNs)
Oct 2006	4697	Documents DNS resolution misbehavior
Oct 2006	4701	Defined a resource record for encoding DHCP information in the DNS and specifies the DHCID RR
Oct 2006	4703	Described situations in which conflicts can arise in the use of DHCP in clients and servers when dynamically updating name to address and address to name mappings in the DNS
Jan 2007	4795	Defined a DNS-like protocol for resolving names on local network (Link Local Multicast Name Resolution)
May 2007	4870	Described a historic approach to domain-based email authentication using public keys advertised in the DNS

Date	RFC	Modification
Jun 2007	4892	Discussed the requirements for a mechanism to identify a name server instance in an Anycast cloud
Jul 2007	4955	Documents experiments using DNSSEC
Jul 2007	4956	Proposed a way of opting in to DNSSEC
Aug 2007	4986	Explored the requirements for rolling DNSSEC trust anchors
Aug 2007	5001	Defined an option to cause a name server to identify itself
Sep 2007	5011	Described the methodology to automate updates of DNSSEC trust anchors
Nov 2007	5074	Clarified the mechanisms used to allow islands of trust to be validated using DNSSEC
Mar 2008	5155	Defined DNSSEC hashed authenticated denial of existence and specifies the NSEC3 RR
Mar 2008	5158	Specified the reverse DNS delegation for the 6to4 IPv6 transition mechanism
Apr 2008	5205	Defined DNS extensions for the Host Identity Protocol (HIP)
Oct 2008	5358	Discussed how to prevent the use of recursive name servers in reflector attacks
Nov 2008	5395	Clarified IANA actions related to the DNS
Jan 2009	5452	Discussed measures to make the DNS more resilient against forged responses
Apr 2009	5507	Provided recommendations when designers are considering expanding the DNS
Apr 2009	5509	Defined SRV resource records for use with SIP instant messaging and presence registration
Aug 2009	5625	Offered guidelines on the implementation of DNS proxies
Dec 2009	5679	Discussed how to locate IEEE 802.21 mobility services using the DNS
Oct 2009	5702	Defined the use of SHA-2 algorithms with RSA in DNSKEY and RRSIG RRs for DNSSEC

Date	RFC	Modification
Feb 2010	5782	Described the use of DNS in blacklists and whitelists
May 2010	5855	Specified a stable naming scheme name servers for the IN-ADDR.ARPA and IP6.ARPA zones
Apr 2010	5864	Defined SRV resource records for use with AFS
May 2010	5910	Re-specified the use of EPP for DNSSEC
Jul 2010	5933	Specified how to use of GOST signature algorithms in DNSKEY and RRSIG RRs for DNSSEC
Jun 2010	5936	Clarified the DNS zone transfer (AXFR) protocol
Aug 2010	5966	Clarified the requirements for the use of TCP with DNS
Nov 2010	6014	Described cryptographic algorithm identifier allocation for DNSSEC
Apr 2011	6147	Defined DNS extensions for NAT from IPv6 clients to IPv4 servers
May 2011	6168	Discussed the requirements for the management of name servers
Mar 2011	6195	Provided more clarification on IANA actions relating to the DNS

Internet Society
Galerie Jean-Malbuisson 15
CH-1204 Geneva, Switzerland
Tel: +41 22 807 1444
Fax: +41 22 807 1445
http://www.isoc.org

1775 Wiehle Ave. Suite 201
Reston, VA 20190, USA
Tel: +1 703 439 2120
Fax: +1 703 326 9881
Email: info@isoc.org

www.internetsociety.org