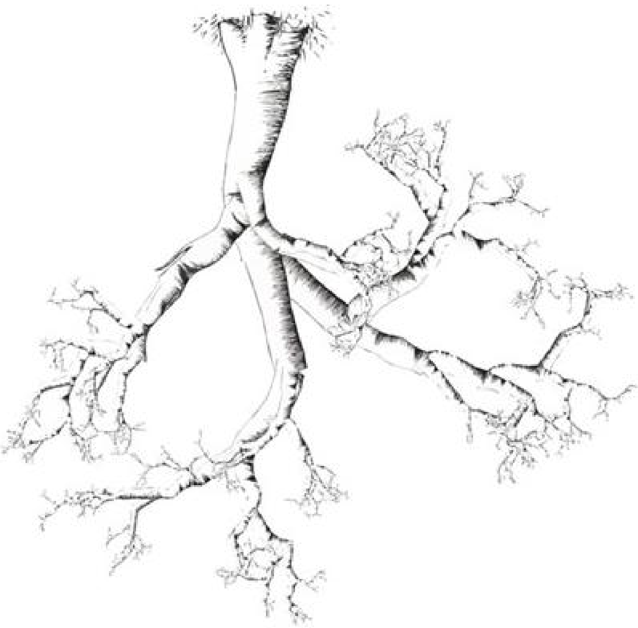
A DNS Primer



□ Every Sunday I put out a curated list of the week's most interesting stories in infosec, technology, and humans. You can subscribe to it here.

[NOTE: For more primers like this, check out my tutorial series.]

WHAT IS THE DNS?

The Domain Name System (DNS) makes the Internet usable to humans by providing a naming structure for online resources and mapping those names to the addresses where the resources reside. Without it, websites would be accessible only by entering long strings of numbers, such as: “120.238.104.535”.

Humans aren’t good at retaining such things, but remembering “npr.org” is fairly manageable. Enter the DNS.

HISTORY

The DNS was created by Paul Mockapetris at UC Irvine in 1983. Before then, people were mapping names to addresses by sharing a big text file called `hosts.txt`. This is why most operating systems have a hosts file even today.

HOW IT WORKS

Let’s look at how the DNS works.

BASICS

For those who prefer a walkthrough approach, the video above describes DNS resolution visually.

As we said in the intro, the DNS is a system that finds resources for you by name. You ask where a name is, and it returns you an IP address. This is done through a distributed database system whereby requests for names are handed off to various tiers of servers which are delineated by the dot (.) in the name you're looking for. It uses the client-server model.

The structure is hierarchical, and moves from right to left like so:

- The root domain (dot)
- The top-level domain (TLD)
- The second-level domain
- The subdomain
- The host/resource name

Clients, like your laptop or desktop, are usually configured with a DNS server that it uses to get names resolved. Clients usually make recursive queries, meaning that they just want the final answer—leaving the DNS server to do the work of walking the tree.

Name resolution for a typical client follows the steps described below:

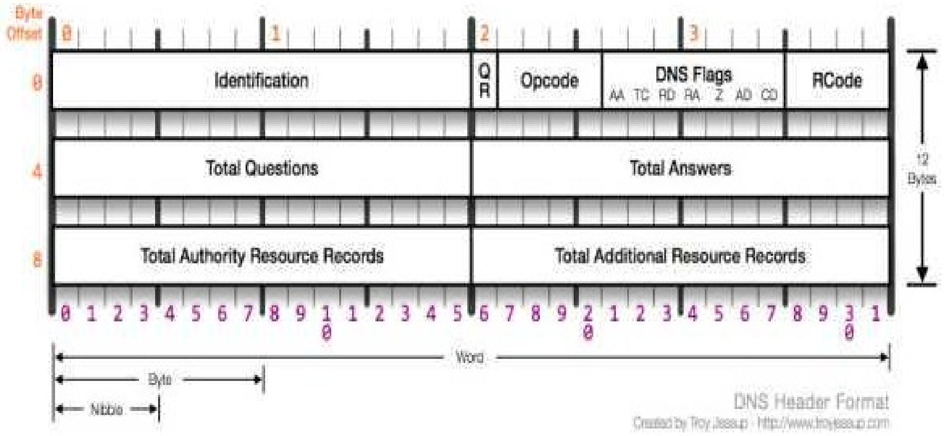
1. A DNS server is configured with an initial cache (so called hints) of the known addresses of the root name servers. The hint file is updated periodically in a trusted, authoritative way.
2. When a client makes a (recursive) request of that server, it services that request either through its cache (if it already had the answer from a previous lookup) or by performing

the following steps on the client's behalf.

3. A query is made to one of the root servers to find the server authoritative for the top-level domain being requested.
4. An answer is received that points to the nameserver for that resource.
5. The server "walks the tree" from right to left, going from nameserver to nameserver, until the final step which returns the IP address of the host in question.
6. The IP address of the resource is then given to the client.

[**NOTE**: The name resolution process is different for recursive vs. iterative queries. See that section for more detail.]

THE PROTOCOL



The DNS protocol is relatively light (see the image above) and thus sits on top of UDP so that queries can happen quickly and without much overhead. Queries over 512 bytes, and certain heavier operations such as Zone Transfers, however, switch to using TCP so that delivery will not become an issue.

Fields

The protocol has the following fields:

- **The Identifier**: a 16-bit ID field that matches requests and responses.
- **The Query/Rsponse Flag**: a 4-bit field that designates whether the packet is a request or a response.
- **Opcode**: Specifies the type of message being carried. Options include: 0 for standard query, 1 for an inverse query (obsolete), 2 for server status, 3 is reserved and unused, 4

is notify message, and 5 is an update (used for Dynamic DNS).

- **AA**: This is a 1-bit field indicating an authoritative answer. The bit is set to 1 if it's authoritative, meaning that the server who gave the answer is authoritative for the domain in question. If it's set to 0 it's a non-authoritative answer.
- **TC**: A 1-bit field for truncation, yes or no. Usually indicates it was sent via UDP but was longer than 512 byes.
- **RD**: A 1-bit field called "Recursion Desired", meaning that the client is asking the server to walk the tree on the client's behalf and just return the answer as opposed to telling it where to look next.
- **RA**: A 1-bit field called "Recursion Available", in which a DNS server tells a client that it support recursion or not.
- **Z**: Three reserved bits that are always set to zeroes.
- **RCode**: A 4-bit field that's set to zero in queries (because they're not responses) with the following options: 0 is no error, 1 is format error, 2 is server failure, 3 is name error, 4 is not implemented, 5 is refused, 6 the name exists but it shouldn't, 7 a resource record exists that shouldn't, a resource record that should exist doesn't, 9 the response is not authoritative, 10 the name in the response is not within the zone specified.
- **QDCount**: How many questions in the question section.
- **ANCount**: How many answers in the answer section.
- **NSCount**: How many resource records in the authority section.
- **ARCount**: How many resource records in the additional section.

DNS CACHES VS. DNS SERVERS VS. DNS RESOLVERS

One of the biggest points of confusion regarding DNS comes from the difference between DNS caches, DNS servers, and DNS resolvers.

DNS Caches

A DNS cache can mean a couple of different things, which is why it's confusing.

1. The list of names and IPs that you've resolved recently, which are "cached" such that if you ask the question again you'll get the same answer without generating network traffic.
2. A DNS server that doesn't have any authoritative names itself, but just performs recursive queries and caching (saving those answers for future requests within a certain

amount of time).

So when someone says they need to clear their DNS cache, they’re probably talking about their local cache. If they’re talking about setting up a DNS Cache, they’re probably talking about a DNS server that just makes DNS queries faster for the network.

DNS Servers

A DNS server is software that serves DNS requests for clients. It can be a cache (see above) which doesn’t have any names of its own and just performs recursive queries (and caching), or it can be a “real” server, meaning that it does hold the authoritative answers for certain resources.

DNS Resolvers

DNS resolvers are just DNS clients. They can make two main types of queries: iterative, and recursive. See that section below.

RECURSIVE VS. ITERATIVE QUERIES

As mentioned above, recursive queries are queries where the client asks the server to do all the work for it. It sends in its query the RECURSION DESIRED flag, and the DNS server will either honor that or not.

Iterative queries are the opposite of recursive queries. When they’re used, the server doesn’t go find the answer for the client (unless it’s on the first question and response), but rather tells the client where to look next. So if the client asks for chat.google.com, it tells the client to check with the .com servers and considers its work done.

AUTHORITATIVE VS. NON-AUTHORITATIVE RESPONSES

Authoritative responses are responses that come directly from a nameserver that has authority over the record in question. Non-authoritative answers come second-hand (or more), i.e., from another server or through a cache.

REVERSE VS. FORWARD QUERIES

Reverse queries simply reverse the direction of DNS lookups, i.e. going from IP to name instead of name to IP. Forward queries are another name for normal name-to-IP queries.

ZONE TRANSFERS

Zone Transfers are the means by which slave servers pull records from master servers for backup and redundancy purposes. They take place over TCP because the data being transfered is usually substantial (and mostly likely over 512-bytes). During the operation, the client sends a query type of IXFR instead of AXFR.

Zone Transfers are sensitive from a security standpoint because when someone knows what and where your resources are, it helps them plan an attack against you. Zone Transfers should only be allowed by approved systems.

Performing a Zone Transfer

When you perform a zone transfer you basically want to define two things:

1. The server you're asking
2. The domain you're trying to pull

You can perform the actual transfer using a number of tools.

Using the `host` command

```
# host -la $DOMAIN
```

[**NOTE**: Keep in mind that there are two pieces to doing a Zone Transfer: defining the server you're asking, and defining the zone you're trying to pull. With `host`, you need to define the first piece by using your resolv.conf file, i.e. by setting your DNS server to be the target.]

Using the `dig` command you can do it in one step…

```
# dig @server $DOMAIN axfr
```

You can also use `nslookup` to perform a Zone Transfer, but it's an elaborate process and therefore silly. Use `host` or `dig`.

[**NOTE**: Performing a Zone Transfer against a domain without permission may be considered "hacking" to some. You should either have permission or be prepared to face consequences.]

ANYCAST DNS

Anycast is a brilliant protocol that allows the same IP to be served from multiple locations. The network then decides intelligently which location to route a given user request to, based on distance, latency, network health conditions, etc.

Anycast DNS does this for DNS, making it almost like a CDN for your DNS. If you have a site that is accessed from many parts of the world, and where speed is a consideration, you should consider using a DNS provider that has an Anycast option.

[**NOTE**: This site uses Anycast DNS (through DYN) as part of its /stack.]

WILDCARD DNS

Wildcard DNS is a type of DNS record that will respond to non-existent subdomains/hosts within a zone. So if you have a wildcard for “*.danielmiessler.com”, then someone going to “**aargghhh**.danielmiessler.com” will be directed to wherever I point that wildcard record to.

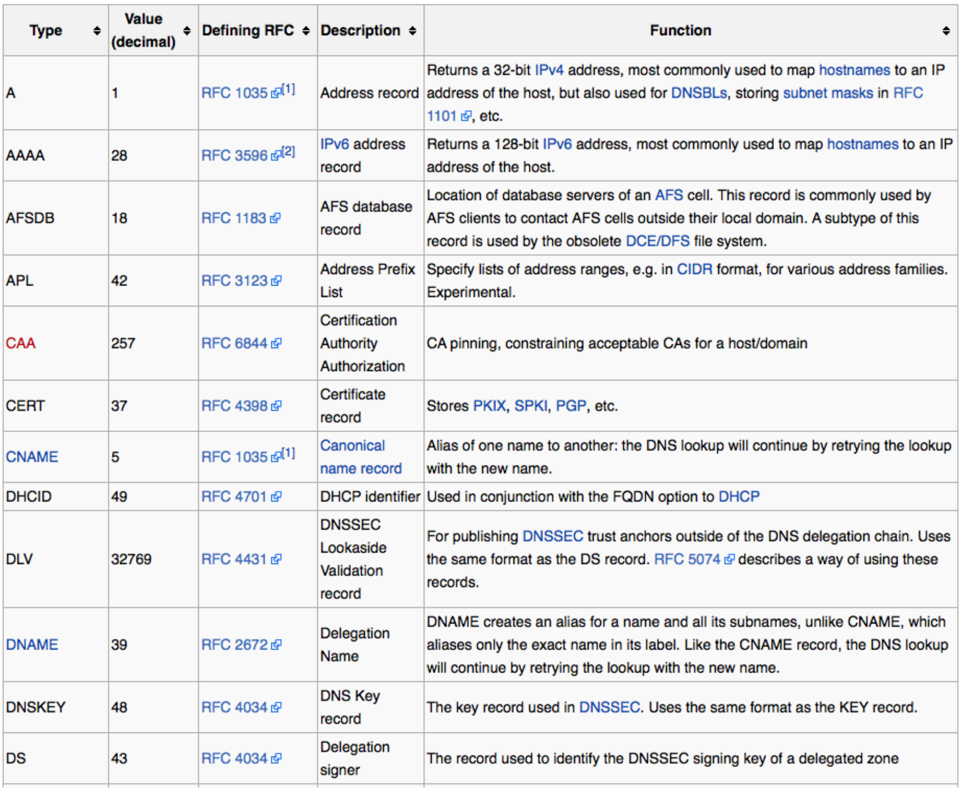
For example:

```
*.danielmiessler.com.   3600   TXT   "This is a wildcard."
```

DYNAMIC DNS

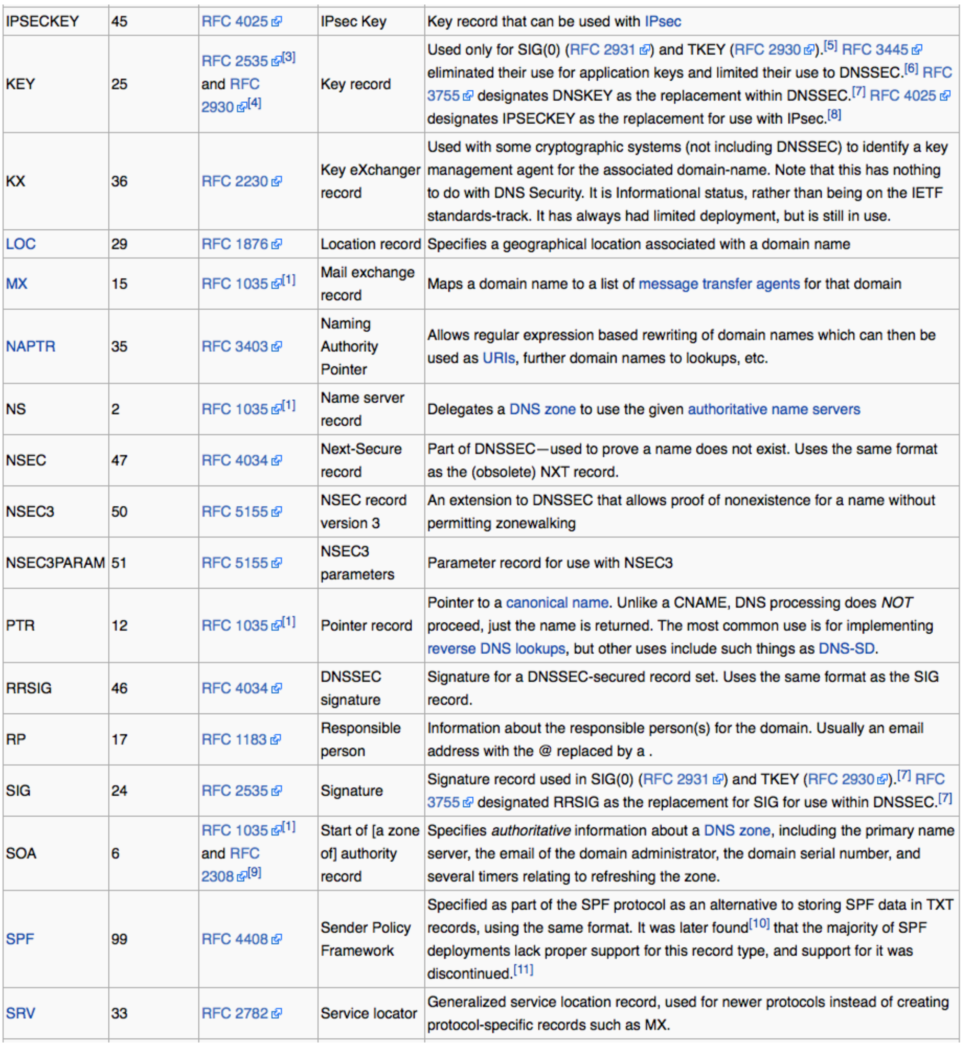
Dynamic DNS allows clients with changing DHCP addresses to update a DNS server with their latest IP so that it can be found by name at its current location.

RECORD TYPES



Type	Value (decimal)	Defining RFC	Description	Function
A	1	RFC 1035[1]	Address record	Returns a 32-bit IPv4 address, most commonly used to map hostnames to an IP address of the host, but also used for DNSBLs, storing subnet masks in RFC 1101, etc.
AAAA	28	RFC 3596[2]	IPv6 address record	Returns a 128-bit IPv6 address, most commonly used to map hostnames to an IP address of the host.
AFSDB	18	RFC 1183	AFS database record	Location of database servers of an AFS cell. This record is commonly used by AFS clients to contact AFS cells outside their local domain. A subtype of this record is used by the obsolete DCE/DFS file system.
APL	42	RFC 3123	Address Prefix List	Specify lists of address ranges, e.g. in CIDR format, for various address families. Experimental.
CAA	257	RFC 6844	Certification Authority Authorization	CA pinning, constraining acceptable CAs for a host/domain
CERT	37	RFC 4398	Certificate record	Stores PKIX, SPKI, PGP, etc.
CNAME	5	RFC 1035[1]	Canonical name record	Alias of one name to another: the DNS lookup will continue by retrying the lookup with the new name.
DHCID	49	RFC 4701	DHCP identifier	Used in conjunction with the FQDN option to DHCP
DLV	32769	RFC 4431	DNSSEC Lookaside Validation record	For publishing DNSSEC trust anchors outside of the DNS delegation chain. Uses the same format as the DS record. RFC 5074 describes a way of using these records.
DNAME	39	RFC 2672	Delegation Name	DNAME creates an alias for a name and all its subnames, unlike CNAME, which aliases only the exact name in its label. Like the CNAME record, the DNS lookup will continue by retrying the lookup with the new name.
DNSKEY	48	RFC 4034	DNS Key record	The key record used in DNSSEC. Uses the same format as the KEY record.
DS	43	RFC 4034	Delegation signer	The record used to identify the DNSSEC signing key of a delegated zone

<

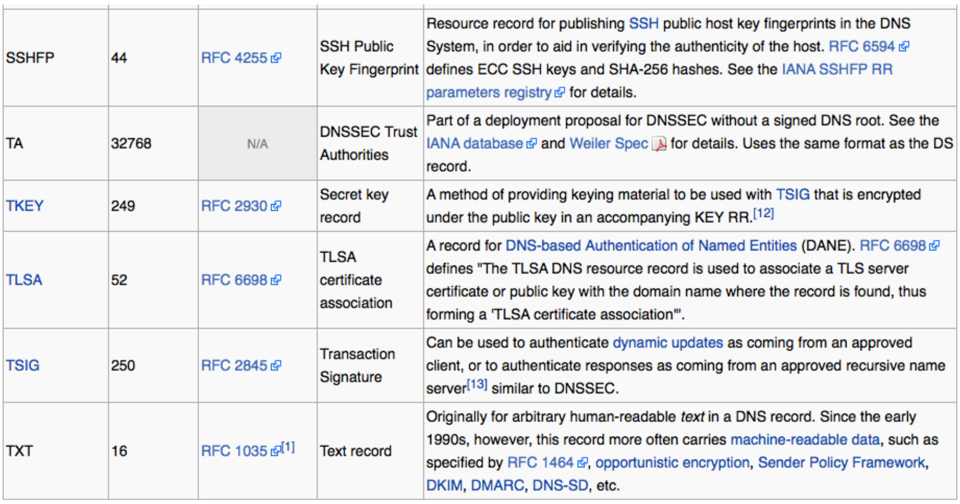


IPSECKEY	45	RFC 4025	IPsec Key	Key record that can be used with IPsec
KEY	25	RFC 2535[3] and RFC 2930[4]	Key record	Used only for SIG(0) (RFC 2931) and TKEY (RFC 2930).[5] RFC 3445 eliminated their use for application keys and limited their use to DNSSEC.[6] RFC 3755 designates DNSKEY as the replacement within DNSSEC.[7] RFC 4025 designates IPSECKEY as the replacement for use with IPsec.[8]
KX	36	RFC 2230	Key eXchanger record	Used with some cryptographic systems (not including DNSSEC) to identify a key management agent for the associated domain-name. Note that this has nothing to do with DNS Security. It is Informational status, rather than being on the IETF standards-track. It has always had limited deployment, but is still in use.
LOC	29	RFC 1876	Location record	Specifies a geographical location associated with a domain name
MX	15	RFC 1035[1]	Mail exchange record	Maps a domain name to a list of message transfer agents for that domain
NAPTR	35	RFC 3403	Naming Authority Pointer	Allows regular expression based rewriting of domain names which can then be used as URIs, further domain names to lookups, etc.
NS	2	RFC 1035[1]	Name server record	Delegates a DNS zone to use the given authoritative name servers
NSEC	47	RFC 4034	Next-Secure record	Part of DNSSEC—used to prove a name does not exist. Uses the same format as the (obsolete) NXT record.
NSEC3	50	RFC 5155	NSEC record version 3	An extension to DNSSEC that allows proof of nonexistence for a name without permitting zonewalking
NSEC3PARAM	51	RFC 5155	NSEC3 parameters	Parameter record for use with NSEC3
PTR	12	RFC 1035[1]	Pointer record	Pointer to a canonical name. Unlike a CNAME, DNS processing does *NOT* proceed, just the name is returned. The most common use is for implementing reverse DNS lookups, but other uses include such things as DNS-SD.
RRSIG	46	RFC 4034	DNSSEC signature	Signature for a DNSSEC-secured record set. Uses the same format as the SIG record.
RP	17	RFC 1183	Responsible person	Information about the responsible person(s) for the domain. Usually an email address with the @ replaced by a .
SIG	24	RFC 2535	Signature	Signature record used in SIG(0) (RFC 2931) and TKEY (RFC 2930).[7] RFC 3755 designated RRSIG as the replacement for SIG for use within DNSSEC.[7]
SOA	6	RFC 1035[1] and RFC 2308[9]	Start of [a zone of] authority record	Specifies *authoritative* information about a DNS zone, including the primary name server, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.
SPF	99	RFC 4408	Sender Policy Framework	Specified as part of the SPF protocol as an alternative to storing SPF data in TXT records, using the same format. It was later found[10] that the majority of SPF deployments lack proper support for this record type, and support for it was discontinued.[11]
SRV	33	RFC 2782	Service locator	Generalized service location record, used for newer protocols instead of creating protocol-specific records such as MX.

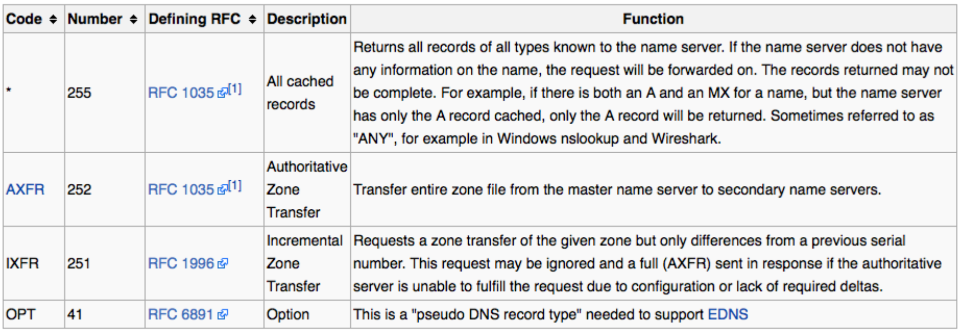
p style=”text-align:center”>

<

p style="text-align:center">



SSHFP	44	RFC 4255	SSH Public Key Fingerprint	Resource record for publishing SSH public host key fingerprints in the DNS System, in order to aid in verifying the authenticity of the host. RFC 6594 defines ECC SSH keys and SHA-256 hashes. See the IANA SSHFP RR parameters registry for details.
TA	32768	N/A	DNSSEC Trust Authorities	Part of a deployment proposal for DNSSEC without a signed DNS root. See the IANA database and Weiler Spec for details. Uses the same format as the DS record.
TKEY	249	RFC 2930	Secret key record	A method of providing keying material to be used with TSIG that is encrypted under the public key in an accompanying KEY RR.[12]
TLSA	52	RFC 6698	TLSA certificate association	A record for DNS-based Authentication of Named Entities (DANE). RFC 6698 defines "The TLSA DNS resource record is used to associate a TLS server certificate or public key with the domain name where the record is found, thus forming a 'TLSA certificate association'".
TSIG	250	RFC 2845	Transaction Signature	Can be used to authenticate dynamic updates as coming from an approved client, or to authenticate responses as coming from an approved recursive name server[13] similar to DNSSEC.
TXT	16	RFC 1035[1]	Text record	Originally for arbitrary human-readable *text* in a DNS record. Since the early 1990s, however, this record more often carries machine-readable data, such as specified by RFC 1464, opportunistic encryption, Sender Policy Framework, DKIM, DMARC, DNS-SD, etc.



Code	Number	Defining RFC	Description	Function
*	255	RFC 1035[1]	All cached records	Returns all records of all types known to the name server. If the name server does not have any information on the name, the request will be forwarded on. The records returned may not be complete. For example, if there is both an A and an MX for a name, but the name server has only the A record cached, only the A record will be returned. Sometimes referred to as "ANY", for example in Windows nslookup and Wireshark.
AXFR	252	RFC 1035[1]	Authoritative Zone Transfer	Transfer entire zone file from the master name server to secondary name servers.
IXFR	251	RFC 1996	Incremental Zone Transfer	Requests a zone transfer of the given zone but only differences from a previous serial number. This request may be ignored and a full (AXFR) sent in response if the authoritative server is unable to fulfill the request due to configuration or lack of required deltas.
OPT	41	RFC 6891	Option	This is a "pseudo DNS record type" needed to support EDNS

DNS SECURITY

There are a number of things to think about from a security perspective when it comes to DNS. First among these is the fact that if someone controls where you are sent when you ask for a given name, they control something quite powerful.

SPOOFING A LEGITIMATE SITE

Modifying DNS servers for clients is often a primary objective of an attacker after gaining control to a system or network. This means changing the DNS resolution so that certain sensitive names (like bankofamerica.com, for example), or even *all* names, are redirected to a server that the attacker controls.

This enables an attacker to present a login form that looks similar to (or even identical to) the real thing. If the user signs into the fake site it means the attacker now has stolen their credentials.

It’s critical, therefore, that the nameserver responding to client requests in your environment is legitimate and is not compromised.

DNSSEC

By default, DNS is fairly easy to spoof because it’s based on UDP. In many cases You can simply send a response to a client and it will assume that you made a previous request and update the record in the cache.

DNSSEC is a set of security-oriented DNS extensions designed to address a number of issues with DNS. It is primarily concerned with helping resolvers (clients) ensure that DNS data in fact came from an authorized origin.

DNSSEC works by digitally signing responses using public-key cryptography and uses several new resource records, shown below.

- RRSIG – contains the DNSSEC signature for a record set. DNS resolvers verify the signature with a public key, stored in a DNSKEY-record.
- DNSKEY – contains the public key that a DNS resolver uses to verify DNSSEC signatures in RRSIG-records.
- DS – holds the name of a delegated zone. You place the DS record in the parent zone along with the delegating NS-records. references a DNSKEY-record in the sub-delegated zone.
- NSEC – Contains a link to the next record name in the zone and lists the record types that exist for the record’s name. DNS Resolvers use NSEC records to verify the non-existence of a record name and type as part of DNSSEC validation.
- NSEC3 – Contains links to the next record name in the zone (in hashed name sorting order) and lists the record types that exist for the name covered by the hash value in the first label of the NSEC3 -record’s own name.These records can be used by resolvers to verify the non-existence of a record name and type as part of DNSSEC validation. NSEC3 records are similar to NSEC records, but NSEC3 uses cryptographically hashed record names to avoid the enumeration of the record names in a zone.
- NSEC3PARAM – Authoritative DNS servers use this record to calculate and determine which NSEC3-records to include in responses to DNSSEC requests for non-existing names/types.

When DNSSEC is used, each answer to a DNS lookup contains an RRSIG DNS record, in addition to the record type that was requested. The RRSIG record is a digital signature of the answer DNS resource record set. The digital signature is verified by locating the correct public key found in a DNSKEY record.

DNS ATTACKS

There are number of attack types associated with DNS. We'll cover just a few of them.

1. **Changing Your DNS Server**: An attacker who can change your DNS server can control where you are taken when you request sensitive resources, like your bank, etc.
2. **Distributed Denial of Service**: Because DNS uses UDP, you can request from thousands, or millions, of DNS servers the address for a given name, but do so with the spoofed source address of your victim. This results in that victim being melted by all the response traffic. Many tools exist for doing this.
3. **The Kaminsky Attack**: The Kaminsky attack was an issue with predictable DNS IDs that allowed attackers to flood a given system with responses that would then be written and passed onto clients.For more on this attack, see this excellent resource.

DNS SOFTWARE

The primary players in DNS software are:

1. Bind: Ubiquitous, powers most of the Internet.
2. DJBDNS: Focused on security, less used

PROVIDERS

A number of DNS providers exist that provide various benefits.

1. **DYN**: DYN is a DNS provider that allows you to do registration and host your zones. It also provides Anycast and other advanced services.
2. **DNSMadeEasy**: DNSMadeEasy is another big name in DNS that provides a number of related services.
3. **GoDaddy**: Godaddy is a rather popular DNS service that has a number of services in the space.
4. **OpenDNS**: OpenDNS is an interesting service that provides content and security filtering based on monitoring and blocking DNS requests. You configure your hosts or network to use OpenDNS's DNS servers, and it will block requests for certain categories of site and/or for malware before your browser even starts going there.

This author has always used DYN, but any of the top tier services are likely good options. The key to remember when choosing a provider is that if your DNS is compromised for a site that is sensitive, you will likely experience similar negative effects to having your site compromised. Choose wisely.

DNS TOOLS

There are a few DNS tricks and tools that you always want to have available to you.

UTILITIES

1. `dig`: `dig` is supremely valuable for performing all manner of DNS tasks. I'll be doing a primer on it soon.
2. `nslookup`: `nslookup` is my least favorite DNS tool because I dislike the syntax, but it is on most systems and thus should be learned to some degree.

ACTIONS

Here are a couple things you'll want to be able to do on any computer related to DNS.

Change Your DNS Server

Sometimes you need to be able to change your DNS server.

1. **Windows** Go to your network configuration and change your DNS server(s) and apply/exit.
2. **OSX**: Open your Network Preferences and click the Advanced button for the connection you're on.
3. **Linux**: Modify your `/etc/resolv.conf` file.

Clear Your DNS Cache

There are times when you have previously resolved a name for a resource, which has now changed, and you need to get the entry updated in your cache. This is how to do it for the three main operating systems.

1. **Windows** `ipconfig /flushdns`
2. **OSX**: `sudo killall -HUP mDNSResponder`
3. **Linux**: `sudo /etc/init.d/nscd restart`

DNS INTERVIEW QUESTIONS

Q: What port does DNS work over?

A: 53

Q: What protocol does DNS work over?
A: UDP

Q: Does DNS always work over UDP?
A: No, it switches to TCP if the content is greater than 512 bytes.

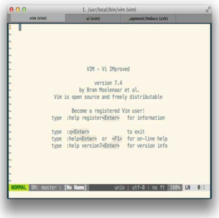
Q: What's the primary role of DNSSEC?
A: To assure clients that the answers received came from the authorized server.

Q: What's an example of a security problem related to DNS?
A: Having someone change your DNS server to a malicious one, having someone send you malicious DNS replies that get accepted as legitimate, using DNS to DDoS someone.

REFERENCES

1. Resolution actually starts at the far right dot (.), not at the TLD, but this is a minor technical detail.
2. The Wikipedia article on DNS.
3. Definitely check out my friend Steve's phenomenal description of the Kaminsky DNS Attack: http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html.

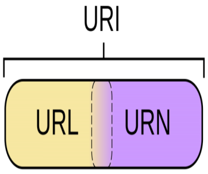
YOU'LL PROBABLY LIKE THESE AS WELL...



```
1. /usr/local/bin/vim (vim)
vim (vim)          vi (vim)          ..opment/htdocs (zsh)
1
~
~
~
~                     VIM - Vi IMproved
~
~                       version 7.4
~                  by Bram Moolenaar et al.
~         Vim is open source and freely distributable
~
~                Become a registered Vim user!
~       type  :help register<Enter>   for information
~
~       type  :q<Enter>               to exit
~       type  :help<Enter>  or  <F1>  for on-line help
~       type  :help version7<Enter>   for version info
~
~
NORMAL  BR: master | [No Name]      unix | utf-8 | no ft  100%  LN   0:1
```

A vim Tutorial and Primer

Intro Why Vim? Approach Configuration vim as Language Getting Things Done Working With Your File…



A Security-focused HTTP Primer

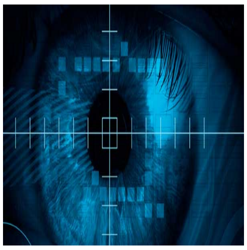
What follows is a primer on the key security-oriented characteristics of the HTTP protocol. It's…

Ideas

If you're like me you've had a number of ideas throughout your life. Most of…

Information Security Interview Questions

Before You Start General Questions Network Security Application Security Corporate/Risk The Onion Model The Role-playing…



How to Build a Successful Information Security Career

I've been writing about infosec for a while now, so I get a good amount…

The Ultimate Speed Guide for WordPress on NGINX

I typically serve web pages from my my Nginx/Wordpress-based stack in 100-300ms, and enough people…

NEWSLETTER

Every Sunday I put out a curated list of the most interesting stories in infosec, technology, and humans.

I do the research, you get the benefits. Over 5K subscribers.

email address

SUBSCRIBE

Search this website …

:: RSS
:: Twitter
:: Github

Recommended.

A vim Tutorial and Primer

A Security-focused HTTP Primer

Ideas

Information Security Interview Questions

How to Build a Successful Information Security Career

NEWSLETTER

Every Sunday I put out a curated list of the most interesting stories in infosec, technology, and humans.

I do the research, you get the benefits. Over 5K subscribers.

email address

SUBSCRIBE