Adventures in Security DNS Episode: Script

[*This is the script for the May 2018 episode on domain name services. Next to each slide heading, I placed the location (minutes and seconds) where you can watch the relevant video clip. The "*[…]*" represents where the screen changes. This is helpful if you are not listening to the audio, but instead are following along with only the script.*]

Slide 1: Intro

Welcome, travelers, to adventures in security. My name is Tom Olzak, and I'll be your guide. In this episode, we take a look at how domain name services work. This is sort of a part one. I will cover DNS security issues in a later episode.

[…]

Slide 2: Agenda

First, we'll define and apply the various components of DNS: on the internet and on the desktop. I will then step through a name resolution process.

[…]

Slide 3

[No content]

[…]

Slide 4: Fully Qualified Domain Names [0:29]

First, let's look at what we call a fully qualified domain name. For the purposes of this episode, a fully qualified domain name is the minimum you must enter into a browser to reach a desired webpage.

Fully qualified domain names are divided into parts. The first is the top-level domain.

[…]

Over 1000 top-level domains exist on the internet, although most users only use a few, like .com, .edu, and .org. Top-level domains can be used by everyone, like .com, or they can be privately owned by an organization.

[…]

When an organization obtains a domain name for putting up a website, it adds its own identifier to the top-level domain. It can be the organization's name, as with Microsoft in this case, or any other identifier the organization… or person… selects. This identifier and the top-level domain are usually enough to get a customer or other interested party to the main page of the website. However, large organizations, like Microsoft, often have pages users want to directly visit to avoid searching and clicking. This is the purpose of the subdomain.

[…]

Once an organization or individual has a domain, they can assign any number of subdomains to it. Each subdomain is typically associated with a specific service or body of information within a larger site. In this example, a Microsoft customer can go directly to the Microsoft support page.

But we have a problem with domain names… routers that direct traffic across the internet do not know where Microsoft.com or any other domain resides. They only understand IP addresses. This is the reason we need DNS.

[…]

Slide 5: DNS Overview [2:30]

DNS converts the domain name to an IP address, so the internet services can find and establish sessions with websites.

[…]

Here, terra.edu would be resolved by DNS to 199.218.41.5.

[…]

One more thing before we continue. WWW was frequently appended to front of domain names in the past. However, this is not necessary. WWW can be used, but it is actually just another subdomain.

[…]

Slide 6: Resolver [3:09]

Next, we look at what is called a resolver. A resolver is software that organizes and performs the steps necessary to translate a domain name to an IP address.

[…]

Resolvers are found on user devices, business servers, and DNS servers.

[…]

Slide 7: Local Device Resolver [3:30]

So how does a resolver do its job?

When a user first enters a fully qualified domain name into a browser, the DNS name resolution process begins.

[…]

Browsers such as Chrome and Firefox contain resolvers that create browser resolver caches. The resolver cache contains the domain name and associated IP address for domains the user, or an application, visited recently. The cache reduces website access time by removing the need for the resolver to go find the IP address again. As you will see in a later slide, going to the internet to obtain a website IP address adds to user and system response times.

This resolver cache is from Google Chrome. Its resolver cache is the first place Chrome visits to see if it has already resolved the domain name. If not, a request is sent to the operating system resolver.

[…]

The operating system resolvers on Windows machines also create resolver caches by default. So a request from a browser results in Windows checking its own cache to see if the IP address is available. Note that Linux installations typically do not include resolver caching by default. It must be enabled.

This is an example of a resolver cache listing. We use the *ipconfig* command with a */displaydns* to view content.

You can see a time to live in this listing. When an administrator sets up a domain name for a website, she assigns a cache time to live. This means that entries in an operating system resolver cache might only stick around for a few minutes before another resolution is necessary.

The time to live is important. While domain names rarely change, IP addresses associated with those domain names change as needed by the organization. This means that we need to perform domain name resolution often to remain current.

[…]

This is what is known as a hosts file. This particular file is from a Windows 10 machine. If the operating system does not find the IP address it needs in the resolver cache, it checks here next.

A hosts file is used by an admin to statically assign IP addresses to domain names. This is not something we want to do very often because of the possibility of IP address changes. However, for testing or for use of internal web services, this is sometimes used. I added the two entries indicated by the red arrow. The rest of the commented lines are included in the hosts file by default.

[…]

Slide 8: Recursive Queries [6:30]

When the local resolver cannot find the domain name in the resolver cache or the hosts file, it sends a recursive query to a DNS server for resolution. A recursive query is defined as a request for which the local resolver expects either the IP address associated with the domain name or a message stating the domain cannot be found.

[…]

Slide 9: DHCP DNS Configuration [6:57]

So how does the local resolver know the IP address of the DNS server to which it should send the recursive query? This is one of the roles of the dynamic host configuration protocol, or DHCP.

DHCP is used to automatically provide an IP address to a device connecting to the network. It also provides the IP address of the first step in the path to reach the internet: the default gateway. However, it is also used to provide the IP address of one or more DNS servers that connected devices should use for recursive queries.

[…]

This is a screen shot of where in the DHCP configuration the admin would supply a DNS server IP address. The IP address could be for a local server, as in this example, or it could be for an external DNS service, like Cisco Umbrella. Further, the admin usually provides at least two IP addresses, so requesting devices can send requests to an alternate server if the first is slow to respond… or doesn't respond at all.

[…]

You can check the IP address of your DNS server on a Windows device by typing *Ipconfig /all* at a command prompt.

[…]

Side 10: DNS Server Types [8:18]

Finally, five types of DNS servers exist. The first is the root server. The root server stores and distributes the IP addresses of the top-level domain servers. Three hundred root servers exist on the internet. Their addresses are known, by default, by all DNS servers.

[…]

The Top-level domain servers contain the IP addresses for authoritative servers. Authoritative servers are typically managed by domain name owners and contain the domain names, sub-domain names, and associated IP addresses. In other words, the authoritative servers contain the information local resolvers need to connect to websites.

[…]

This is an example of how the domain name/IP address pairs are stored on authoritative servers. This is the "A" record for v-cso.com. I own this domain name, so I manage it on an authoritative DNS server. DNS servers have several record types, but we are only concerned with the "A" records in this episode. It is the content of the "A" record that a resolver would receive when looking for my website's IP address.

[…]

A recursive resolver server receives the recursive query requests from local resolvers. We will step through its role in the DNS process in the next slide. It is important to note that it is the recursive resolver server that ultimately returns the IP address and time to live (the "A" record content) to the local resolver.

The recursive resolver might also contain a resolver cache for all domain names it has translated. This, again, minimizes the time needed to resolve frequently used domain names on a network or on a cloud DNS service.

[…]

Finally, there is the forwarding and caching server. I separated this out because it is possible the server with which local resolvers communicate does not manage the domain name resolution process. Rather, it forwards the recursive query to a recursive resolver server. It would also contain a resolver cache containing all the domain name/IP address pairs provided to all devices that send it requests for which the time to live has not expired.

[…]

Slide 11: DNS Resolution Process Title Slide [10:45]

Now that you understand all the components of domain name resolution, let's step through an example.

[…]

Slide 12: Browser Domain Entry

In our example, a user enters the URL terra.edu into Google Chrome. For our purposes, the user has not recently visited this site.

[…]

Slide 13: DNS Iterative Resolution Process

[…]

The local resolvers for Chrome and for the operating system look in their resolver caches for terra.edu. It is not found. It is also not in the hosts file. So…

[…]

The local resolver sends a recursive query for the domain to the local DNS server whose address it obtained during the DHCP process.

[…]

The server, also known as the recursive resolver, checks its own resolver cache. No one using the server has recently accessed this domain, so the IP address is not listed in the cache.

[…]

The recursive resolver now sets up and manages what is known as the iterative resolution process. It begins by sending the domain name to a root server.

[...]

The root server has no idea where terra.edu is, but it does have the IP addresses of the .EDU top-level domain servers.

[...]

The root server returns the top-level domain server address to the recursive resolver. Using that address, the recursive resolver sends the domain name to the top-level domain server.

[...]

The top-level domain server does not have the address for the domain. In other words, it does not have the A record for terra.edu. However, it does know the address of the authoritative server for terra.edu.

[...]

It returns this address to the recursive resolver. Using that address, the recursive resolver now sends the domain name to the authoritative server.

[...]

The authoritative server returns the contents of the terra.edu "A" record to the recursive resolver: including the website IP address and time to live. The server adds this information to its resolver cache. If another user needs this information before the time to live expires, the local DNS server can quickly return the information without having to step through the iterative resolution process.

[...]

Finally, the local DNS server returns the IP address of terra.edu to the local resolver. It is given to the browser, and Chrome initiates a session with the website.

[...]

Slide 14: Summary

In this episode, we walked through the purpose for DNS: the translation, or resolution, of domain names to IP addresses.

We also looked at the DNS domain hierarchy, starting with root, stepping down to the top-level domains, and the makeup of fully qualified domain names.

We saw that authoritative servers, managed by domain owners, contain the IP addresses associated with the domain and all subdomains. This information is kept in "A" records.

Next, we stepped through the name resolution process: including recursive and iterative queries.

And we discussed resolvers on both local devices and DNS servers; how they are related to resolver caches and hosts files; and their importance in the resolution process.

[…]

Slide 15: Closing

That's it for this episode. If you found value in your time spent here, please subscribe by clicking the icon in the lower right corner. Until next time, be careful what you click…