DoH:

AN EXPLAINER

moz://a

CONTENTS

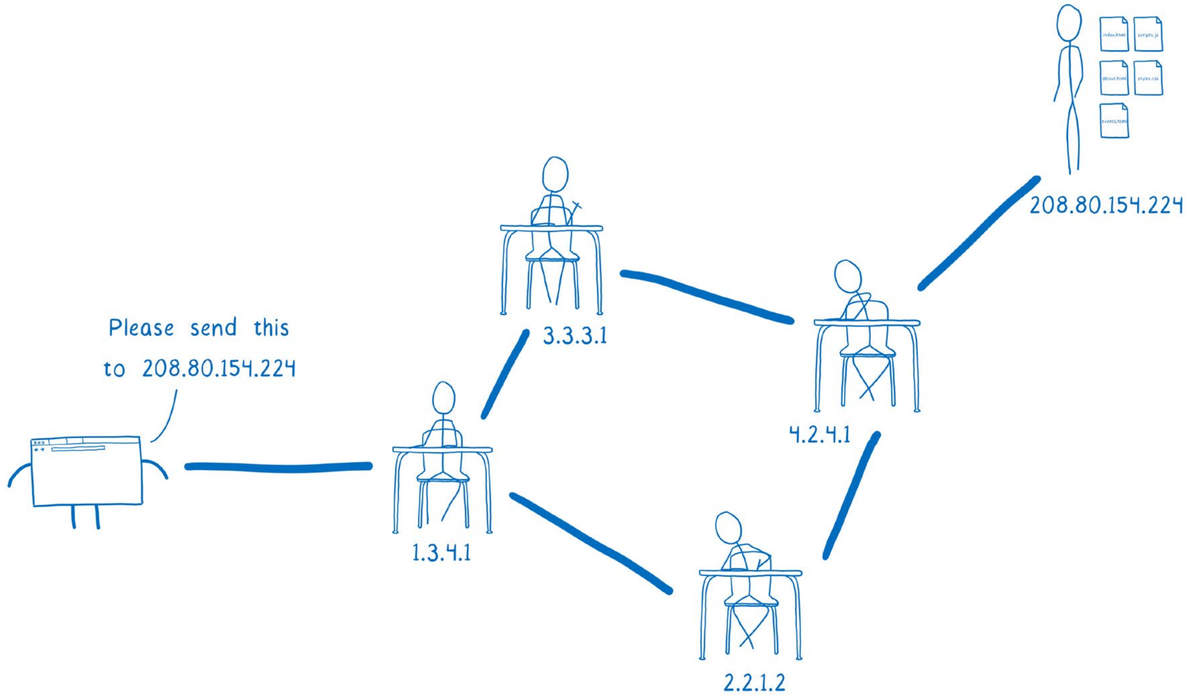
DNS

THE DOMAIN NAME SYSTEM

HTTP requests from your browser need to state who they are going to, but that can't simply be a name. Instead, you have to use an **IP address**.

That's how the routers in between (that carry your request forward) know which server you want to send your request to.

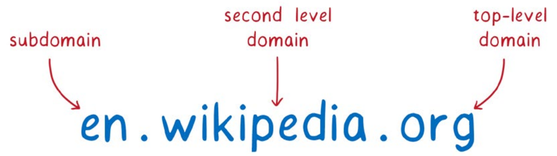
Since remembering a catchy website name is easier than memorising IP addresses, we have the **domain name system (DNS)**.



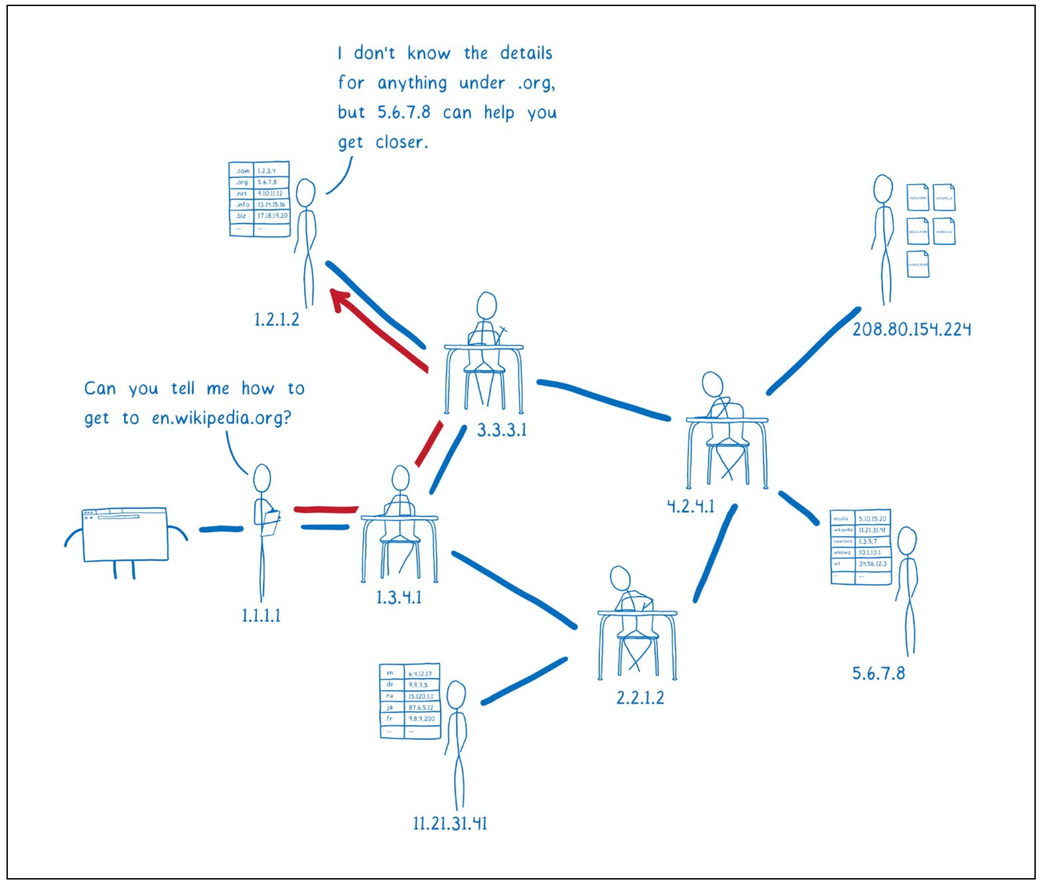
Your browser uses DNS to convert any website name to an IP address. This process — converting the domain name to an IP address — is called domain name resolution.

en.wikipedia.org = 208.80.154.224

We can split this domain into parts.



The tool that will find the right IP address for you is called a resolver.



1.

The resolver talks to a service called the **Root DNS.** It knows of a few different Root DNS servers, so it sends the request to one of them.

2.

The resolver asks the Root DNS where it can find more info about addresses in the .org top-level domain.

3.

The Root DNS will give the resolver an address for a server that knows about .org addresses.

4.

This next server is called a **top-level domain (TLD)** name server. The TLD server knows about all of the second-level domains that end with .org.

5.

It doesn't know anything about the subdomains under wikipedia.org, though, so it doesn't know the IP address for en.wikipedia.org.

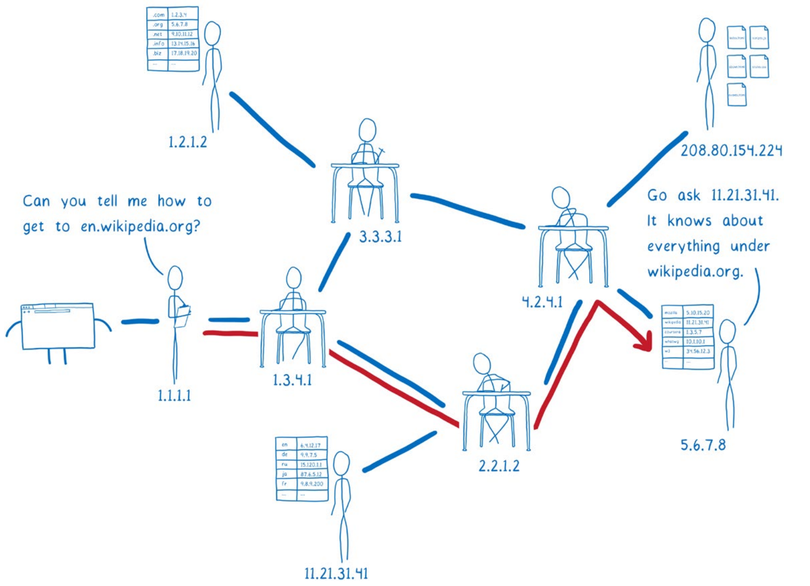
6.

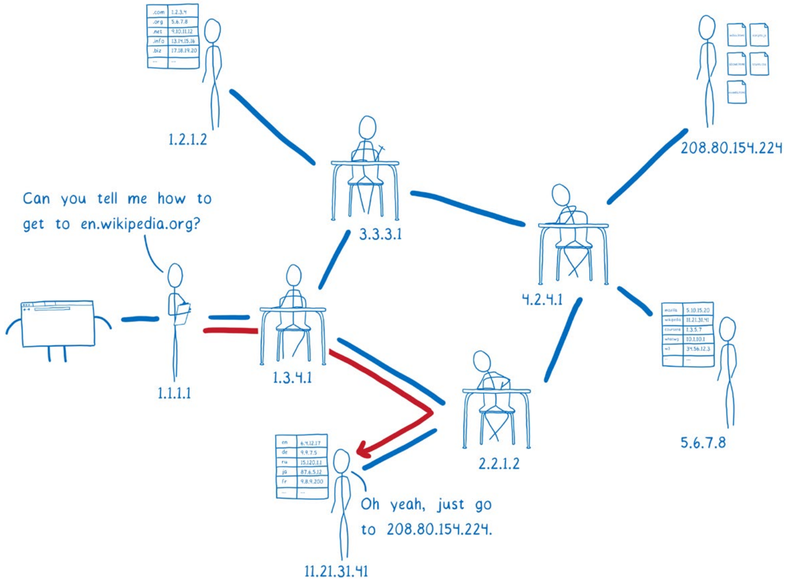
Instead, the TLD name server will tell the resolver to ask Wikipedia's name server.

7.

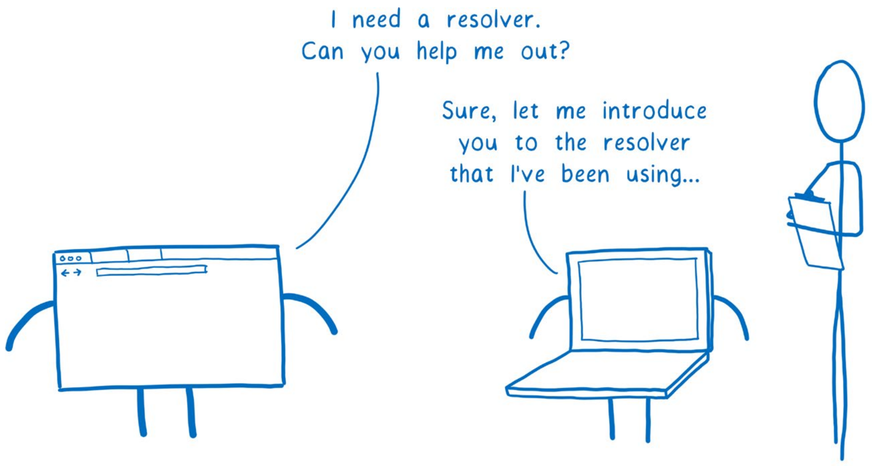
Wikipedia's name server is what's called the **authoritative server**. It knows about all of the domains under wikipedia.org. The authoritative server tells the resolver which IP address has the HTML files for the site.

The resolver will now return the IP address for en.wikipedia.org to the operating system.





This process is called recursive resolution, because you have to go back and forth asking different servers what's basically the same question.



How does the browser find this resolver?

In general, it asks the computer's operating system (OS) to set it up with a resolver that can help.

How does the operating system know which resolver to use?

There are two possible ways. You can configure your computer to use a resolver you trust. But very few people do this. Instead, most people just use the default.

And by default, the OS will just use whatever resolver the network told it to.

When the computer connects to the network and gets its own IP address, the network recommends a resolver to use.

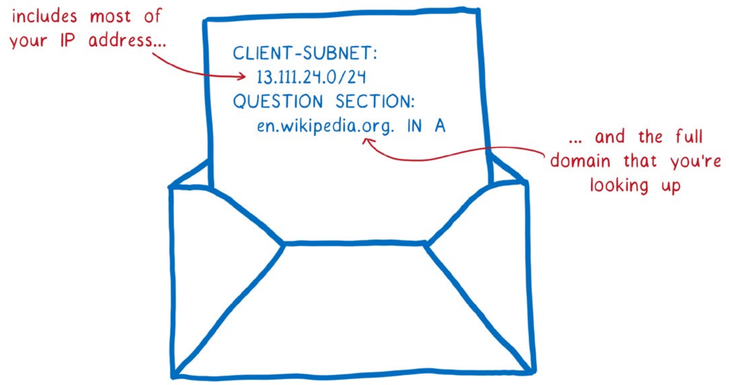
HOW CAN DNS BE EXPLOITED?

HOW CAN THIS SYSTEM MAKE USERS VULNERABLE?

Usually a resolver will tell each DNS server what domain you are looking for. This request sometimes includes your full IP address.

Or if not, increasingly often the request includes most of your IP address. **This data can easily be combined with other information to figure out your identity.**

This means that every server that you ask to help with domain name resolution sees what site you're looking for. But more than that, it also means that **anyone on the path to those servers sees your requests, too.**



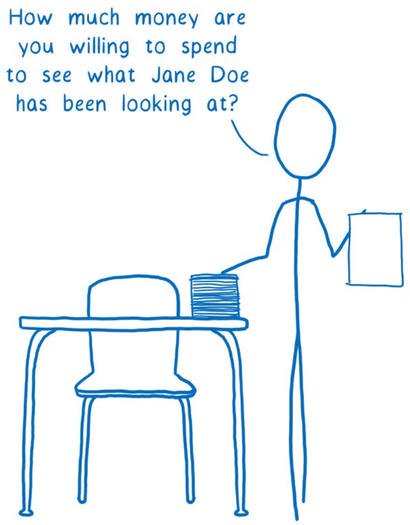
There are a few ways that this system puts users' data at risk. The two major risks are tracking and spoofing.

TRACKING

The risk to user data comes from the resolver itself — the one that the network gives to you. It could be untrustworthy.

Since it is so easy for them to collect the data, the DNS server and anyone along the path to that DNS server — called on-path routers — can create a profile of you.

They can create a record of all of the web sites that they've seen you look up. That data is valuable—many people and companies will pay lots of money to see what you are browsing for.



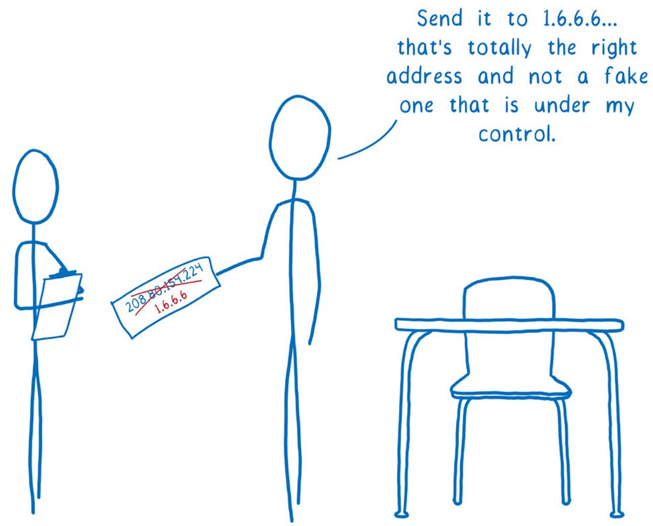
Even if you trust your network's recommended resolver, you're probably only using it when you're at home.

Whenever you go to a coffee shop or hotel or use any other network, you're probably using a different resolver. And who knows what its data collection policies are?

SPOOFING

With spoofing, someone on the path between you and the DNS server changes the response.

Instead of telling you the real IP address, a spoofer will give you the wrong IP address for a site. This way, they can block you from visiting the real site or send you to a scam one.



Again, this is a case where the resolver itself might act nefariously.

For example, let's say you're out shopping for something at Megastore. You want to do a price check for an item to see if you can get it cheaper at a competing online store, big-box.com.

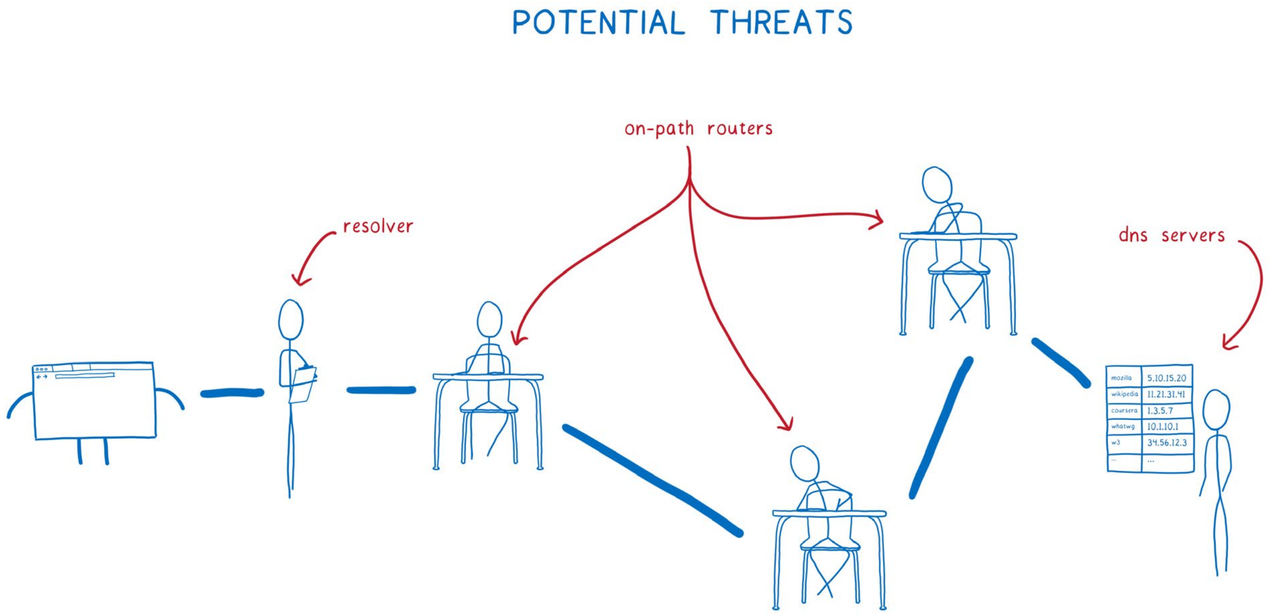
But if you're on Megastore WiFi, you're probably using their resolver. That resolver could hijack the request to big-box.com and lie to you, saying that the site is unavailable.

HOW CAN WE FIX THIS?

TRUSTED RECURSIVE RESOLVERS (TRRS) AND DNS OVER HTTPS (DOH)

At Mozilla, we feel strongly that we have a responsibility to protect our users and their data. We've been working on fixing these vulnerabilities.

We have been working on two things to fix this — an industry-wide technology called **DNS over HTTPS (DoH)** and a special Mozilla program called **Trusted Recursive Resolvers (TRRs).**



There are three threats here:

1. You could end up using an untrustworthy resolver that tracks your requests, or tampers with responses from DNS servers.

2. On-path routers can track or tamper in the same way.

3. DNS servers can track your DNS requests.

So how do we fix these?

1.

Avoid untrustworthy resolvers by using one of Mozilla's Trusted Recursive Resolvers

Networks can get away with providing untrustworthy resolvers that steal your data or spoof DNS because **very few users know the risks or how to protect themselves.**

Even for users who do know the risks, it's hard for an individual user to negotiate with their ISP or other entity to ensure that their DNS data is handled responsibly.

Mozilla has signed agreements with several resolvers that our users can trust to protect your privacy. This means Firefox can use these trusted resolver so that you don't have to worry about rogue resolvers selling your data or tricking you with spoofed DNS.

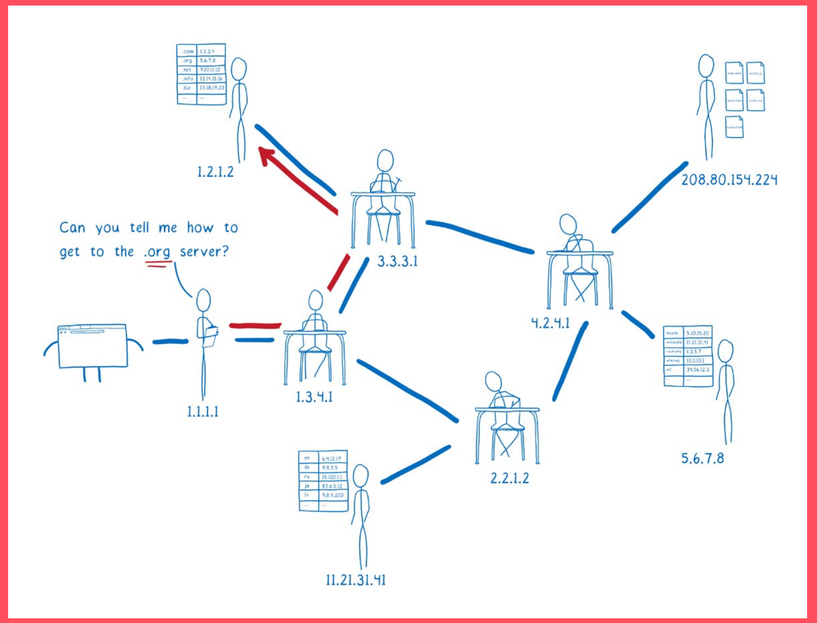
But this doesn't mean you have to use one of the Trusted Recursive Resolvers. Because we respect user choice, you can configure Firefox to use whichever DoH-supporting recursive resolver you want.

2.

Protect against on-path eavesdropping and tampering using DNS over HTTPS

The resolver isn't the only threat, though. On-path routers can track and spoof DNS because they can see the contents of the DNS requests and responses. But the Internet already has technology for **ensuring that on-path routers can't eavesdrop** like this. It's the encryption that we talked about before.

By using **HTTPS to exchange the DNS packets**, we ensure that no one can spy on the DNS requests that our users are making.



3.

Transmit as little data as possible to protect users from deanonymization

Normally, a resolver would send the whole domain name to each server—to the Root DNS, the TLD name server, the second-level name server, etc. But our **TRRs will be doing something different.** They will only send the part that is relevant to the DNS server they're talking to at the moment. This is called **QNAME minimization.**

WHAT ISN'T FIXED BY DOH AND TRRS?

With these fixes, we've reduced the number of people who can see what sites you're visiting. But this doesn't eliminate data leaks entirely.

After you do the DNS lookup to find the IP address, you still need to connect to the web server at that address. To do this, you send an unencrypted initial request. This contains a server name indication (SNI), which specifies which site on the web server you want to connect to.

That means that your ISP and the routers on the way can still figure out which sites you're visiting, because it's right there in the SNI. The proposed solution to this is called Encrypted SNI, which Mozilla is working with industry stakeholders to standardize.

We are aware that no one technology will solve all privacy and security problems on the Internet. **However, DNS over HTTPS is an important piece in the puzzle and will help increase trust in the Internet ecosystem.**

Text and Images -

Lin Clark
A cartoon intro to DNS over HTTPS