TLS and DNS (Domain Name Service)

You wouldn’t think they had much to do with one another. However before TLS can kick in, your browser needs to be able to map URLs to IP addresses. And that is what DNS does. So when you start your browser a connection is made to the DNS server assigned by your ISP. This standard TCP/IP connection is kept open for your whole browser session, and as you connect to various websites, it does its mapping thing for you. At least that is how I think it works.

This implies that websites already have a relationship with the DNS servers, otherwise the DNS server would not be able to find them. But what has all that got to do with TLS?

DNS over TLS

Well the first connection between the two arises because, surprisingly, standard DNS does not use TLS. So your browser communicates with the DNS in plaintext. Which kind of reveals what your interests are to any snooper listening in. Hence DNS over TLS (DoT) has been suggested, as has DNS over Https (DoH) which is basically the same thing but using a different TCP/IP port. These are not yet widely adopted, but probably will be over time.

So TLS can do something for DNS. But what can the DNS do for TLS?

TLS using DNS

Note that before you make a TLS connection to any website, a DNS lookup is done against that website.

Now let us recall some of the shortcomings of TLS. The TLS handshake starts with a clientHello message from client to server. This is sent in plaintext, as the browser at this stage is not aware of the public key associated with the website – indeed it is only in the course of the TLS handshake that this information is handed over.

In the clientHello there will almost certainly be a reference to the URL being visited, in the SNI (Server Name Indication) extension field. Since this is sent in plaintext the snooper again gets to know which websites you are visiting.

To solve this conundrum (without resorting to IBE – see last blog), a solution is to somehow get a suitable public key from the server prior to activating TLS, and it seems the only way of doing that is to get it during the DNS lookup. So the DNS now does a bit more than merely supply an IP address, it is also expected to supply a public key associated with the server. This public key can then be used to encrypt the SNI part of the clientHello. This is called ESNI, for Encrypted SNI. In practise, for DNS servers that support this experimental feature, this public key is changed every hour or so. Therefore this is *not* the servers long term public key which will eventually be supplied along with its authenticating certificate chain.

There is a more recent proposal ECH (Encrypted Client Hello) to extend this idea to encrypt all of the clientHello, see “TLS Encrypted Client Hello” https://datatracker.ietf.org/doc/draft-ietf-tls-esni/

The TLS community seems quite excited by this idea, motivated by the commendable zeal of its members to place privacy and censorship concerns above all others.

Conclusion

It is certainly a clever idea. But complicated to realise. For a more in depth discussion and history of these ideas see Good-bye ESNI, hello ECH! (cloudflare.com). The ECH proposal is quite elaborate. And of course the idea must now be reconsidered with PQC in mind, where public keys can get much bigger. Is it expecting too much of an already overworked DNS server?

From a quick web search ESNI (despite a big push from Cloudfare) doesn’t appear to have achieved much traction. That could be because people are waiting for ECH. Browsers seem quite quick to support these ideas, but of course it is relatively easy for them. ECH will only be used in practise if DNS servers and websites play along.

Hard to tell at this point how all this will play out. I would expect some serious push-back from governmental/law enforcement agencies. Some people (like parents) would like to know which websites their children/grandchildren are visiting!