Server Authentication with DNSSEC

M.Vanderveen

A client-server web model is assumed. Clients are defined as users operating via a machine/device. For Web services, various degrees of client authentication are in use: from anonymous to strongly authenticated (e.g. via RSA token). Thus it is difficult to make broad requirements regarding client authentication -- be that user or machine/device. We therefore conveniently choose not to address this issue. However, on the server side, we submit that strong authentication should always be provided. The client may or may not be able/willing to actually verify the server's credentials- but that ability should not affect the decision to protect the client from rogue or masquerading servers.

In the case that the client device operates without direct or immediate user input, then that device can presumably be provisioned via secure and/or out-of-band means with the certificate (full or hash or in other forms) of the server(s) with which that the device is to communicate. Credential verification is then automated and if implemented correctly should not have any security vulnerabilities beyond those that might be part of the underlying secure pipe establishment protocol, e.g. TLS. For example, a mobile device should periodically retrieve the user's email securely from the right mail server.

In the case the client device must establish a connection to a server because of user initiation, the same server authentication is required. Yet some part of the web security community does not believe that the user can be trusted to verify the server's identity as it appears in its certificate- e.g. check the SubjectName and issuing CA fields (assume the rest of the certificate fields are verified by the device client). The upside is that this user-led trust establishment task need only happen once during the lifetime of the server's certificate.

A case may be made that server identities are not important to be verified for some applications such as public consumer-originated content (e.g. blogspots), but on the other hand some Internet engineers believe that all content should come with assurance of its origin. In an ideal world, all servers providing services to web clients should be able to prove their identity via digital certificates tied to identities that the clients can verify.

Users identify web servers via domain names. Fortunately now that the DNS root is signed and the procedure controlled by ICANN, there is hope that the DNS, via DNSSec, may serve as a root of trust for server identities. This of course does not absolve the user from having to ensure that they are connecting to the right domain name (e.g. correct spelling and the right TLD). But at least once the domain name is verified, trust in the server located at that domain name can be established without further user input.

The details of a method or methods of using DNSSec to tie server domain names to their certificates are being developed in the "DANE" IETF Working Group. Also due to lack of space, we choose to not address herein the following related but important aspects of server identity authentication: policy (is the user allowed to connect to that server), scalability (can the server now handle as many requests), and variable/evolvable connection security.

In summary, the author's personal opinion is that server authentication is a requirement for most if not all web applications, and that server authentication may eventually be solvable via DNSSec.