IPsec's Appeal: Protecting DNS Under the Covers

[Verisign Labs Technical Report #1130006]

Eric Osterweil
eosterweil@verisign.com

Danny McPherson
dmcpherson@verisign.com

Abstract

IPsec is a network-layer protection suite that has met with limited deployment success. While there are clearly some applications for which it is well suited (such as Virtual Private Networks, VPNs), there is often debate over the idea of a wider role for it. Indeed, when applications and higher level protocols are secured by semantics above the network layer, such as by SSL/TLS, DNSSEC, etc. the network layer's protections are often left unaddressed. We propose that security assurances should be embraced by as many layers as possible, and that securing semantics above the network layer should not obviate one from securing the network layer itself. Specifically, we propose that there are very tangible benefits to be gained by augmenting DNSSEC's protections with IPsec. In this work we will outline the specific ways in which IPsec can be used to augment DNSSEC and how their protections are complimentary.

1 Synergy

Internet security is often described as nonexistent, or fundamentally broken. Indeed, there are many problems that range from a diffused and insecure PKI model used by browsers, to insecure Internet routing system, and more. Indeed, many of the protocols and systems that have attempted to add integrity functions to existing Internet protocols (e.g., as DNSSEC provides for DNS) have tended to result in more ways to fail; i.e., more ways for wrong answers to be generated, and often times they have made their target systems more brittle as a result [9]. However, there are many cases where operators can enhance the security of their own networks by embracing and using *more than just one* of the security protocols that already exist. Adding lower layer protections (for example, IPsec at the network layer) can prevent spoofed or manipulated packets from finding their way into the application-level validation component, thereby adding a necessary additional assurance to the transaction.

The IP security architecture, known as IPsec [7], has been a standard security extension to the Internet's network layer since the mid 1990's. However, since then, this security mechanism has gained limited deployment traction. Today, its main use is in constructing secure network-layer tunnels for Virtual Private Networks (VPNs). One reason for IPsec's stunted deployment likely comes from the challenge its users face when trying to enable arbitrary parties in the Internet to discovering each others' IPsec keys in a secure manner. That is, when running a VPN, both the clients and the servers pre-configure the tunnel's IPsec key(s) because they have an out of band relationship. However, traditionally there has not been a way for anonymous key learning in the Internet. However, the IETF has recently produced a new standard for securely learning keys using DNSSEC. The working group responsible for this is called the DNS-based Authentication of Named Entities (DANE) [1] working group, and the details of the protocol are illustrated here [8].

In lieu of this standard, many popular applications (such as HTTP, email, DNS, etc.) have traditionally used security protocols such as the Transport Layer Security protocol (TLS) [6], or (more recently) the DNS Security Extensions (DNSSEC) [2, 4, 3] (because of the lack of an Internet-scale key learning

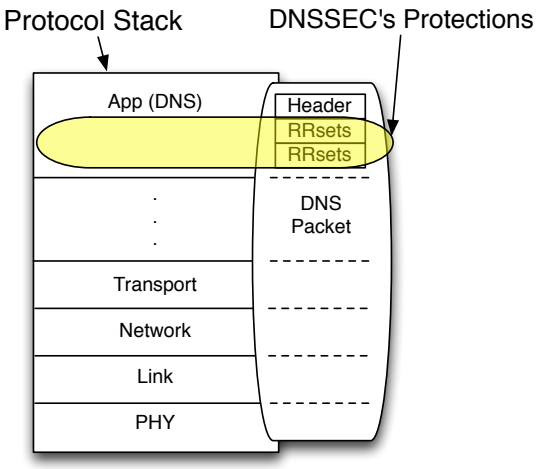

Figure 1: The scope of DNSSEC's protections.

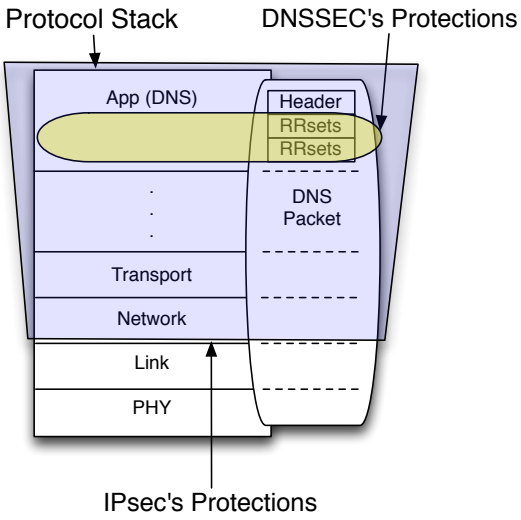

Figure 2: The scope of IPsec's protections, combined with DNSSEC's protections.

facility). Indeed, protecting the higher layers of the protocol stack does not necessarily mean that there is nothing to gain from protecting the lower layers as well, as layered security models and defense in depth are particularly valuable in these applications.

As DNSSEC is becoming an operational reality, securing the DNS infrastructure (in general) is increasingly relevant. DNSSEC's design goals focus only on origin authenticity, data integrity, and secure denial of existence of DNS objects (RRsets) only. It does not attempt other goals, and its protections are narrowly focused on just DNS data, but it *does* protect DNS data beyond the ephemeral query/response transaction (in caches, for example). However, there is clearly more involved in the actual query process than just DNS RRsets. Indeed, as Figure 3 shows, DNSSEC does not attempt to protect the confidentiality of DNS transactions, data outside of the RRsets (including the DNS header and `OPT` record) are not protected, and its DNS-specific focus allows adversaries to identify DNS traffic separately from other traffic in order to target DNS during an attack. In other words, a clever adversary can target *just* DNS traffic, discover the nature of a user's queries, and interfere with the delivery of DNS messages (though the DNS objects are not forgeable).

For example, consider an adversary who oversees outgoing DNS queries near a cache, and spoofs DNS responses. In this case, DNSSEC would protect the cache's integrity, but could enable a Denial of Service (DoS) vector. If an adversary spoofs unvalidatable DNS responses to DNSSEC queries, the cache will not be listening for the genuine responses when they come. Thus, an adversary can leave a caching resolver unable to receive valid responses. Another concern could be one of information leakage, or *disclosure attacks*. An adversary can not only learn what Second Level Domains (SLDs) a user is querying (such as their bank, a funding agency, a security contractor, etc.), but can inspect the fully qualified domain name to also learn the specific hosts visited, or in the case of certain DNS-based chat programs (like Facebook's), information about ongoing conversations. Clearly, this does not invalidate the benefits of DNSSEC, because it still protects the actual DNS objects, eliminates cache poisoning attacks, and more. Rather, we simply aim to illustrate that there is more at stake than just valid DNS data.

To this end, we consider how IPsec's protections can be used to *augment* the protections offered by DNSSEC. If operators were to run IPsec between an authoritative zone's name servers, and the client resolvers that query them, then the clients' security would be augment in the following ways: 1) IPsec's packet-level encryption would protect the confidentiality of both resolvers' queries and the name servers' responses, and 2) while DNSSEC could already be protecting the mapping between a server's domain name and its IP address(es) (via signatures over `A` and `AAAA` RRsets) during name resolution, IPsec would be able to provide the same protections during the actual connection and data transmission. In other words, as Figure 4 shows, DNSSEC would ensure a mapping, and IPsec would ensure the subsequent connection. Moreover, this would include the unprotected DNS header and would also eliminate spoofed responses (like those used in the Kaminsky attack [5]). This overlapping set of protections serves as an illustration that DNSSEC and IPsec are actually complimentary and that their mutual deployment offers a very palpable synergy that neither can achieve without the other.

Our intention in this writing is to demonstrate that securing any single protocol (or protocol layer) does not necessarily mean that nothing can be gained by securing other protocols or layers as well. DNSSEC's charter specifically avoids attempting to offer protections like confidentiality, because these are not goals of the DNS protocol. However, protocols like IPsec not only do this, but help "double-down" on the protections that DNSSEC *is* providing. Similarly, DNSSEC's non-repudiation and resource learning infrastructure offers protections that are beyond the design goals of IPsec. This only further supports the position that as one secures more layers of the security onion, protections can begin to reinforce each other.

2 Using DNSSEC

As DNSSEC is becoming an operational reality, securing the DNS infrastructure (in general) is increasingly relevant. DNSSEC's design goals focus only on origin authenticity, data integrity, and secure denial of existence of DNS objects (RRsets) only. It does not attempt other goals, and its protections are narrowly focused on just DNS data, but it *does* protect DNS data beyond the ephemeral query/response transaction (in caches, for example). However, there is clearly more involved in the actual query process than just DNS RRsets. Indeed, as Figure 3 shows, DNSSEC does not attempt to protect the confidentiality of DNS transactions, data outside of the RRsets (including the DNS header and `OPT` record) are not protected, and its DNS-specific focus allows adversaries to identify DNS traffic separately from other traffic in order to target DNS during an attack. In other words, a clever adversary can target *just* DNS traffic, discover the nature of a user's queries, and interfere with the delivery of DNS messages (though the DNS objects are not forgeable).

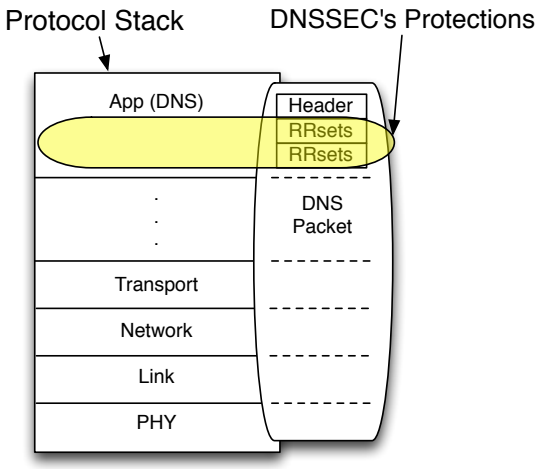

Figure 3: The scope of DNSSEC's protections.

For example, consider an adversary who oversees outgoing DNS queries near a cache, and spoofs DNS responses. In this case, DNSSEC would protect the cache's integrity, but could enable a Denial of Service (DoS) vector. If an adversary spoofs unvalidatable DNS responses to DNSSEC queries, the cache will not be listening for the genuine responses when

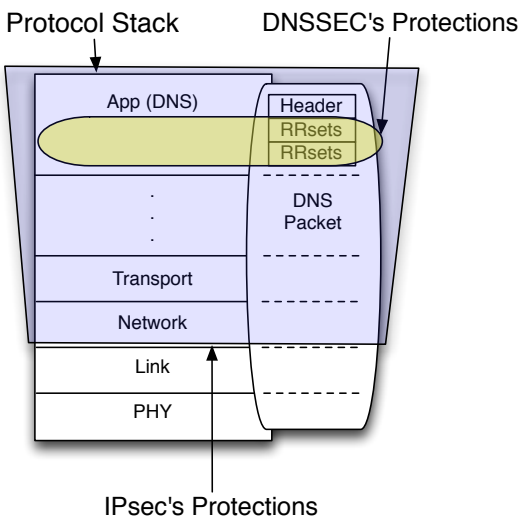

Figure 4: The scope of IPsec's protections, combined with DNSSEC's protections.

they come. Thus, an adversary can leave a caching resolver unable to receive valid responses. Another concern could be one of information leakage, or *disclosure attacks*. An adversary can not only learn what Second Level Domains (SLDs) a user is querying (such as their bank, a funding agency, a security contractor, etc.), but can inspect the fully qualified domain name to also learn the specific hosts visited, or in the case of certain DNS-based chat programs (like Facebook's), information about ongoing conversations. Clearly, this does not invalidate the benefits of DNSSEC, because it still protects the actual DNS objects, eliminates cache poisoning attacks, and more. Rather, we simply aim to illustrate that there is more at stake than just valid DNS data.

3 Using IPsec

To this end, we consider how IPsec's protections can be used to *augment* the protections offered by DNSSEC. If operators were to use DANE to provision IPsec keys in their reverse zones (for their resolvers and name servers), they could then run IPsec between an authoritative zone's name servers, and the client resolvers that query them. Then the clients' security would be augment in the following ways: 1) IPsec's packet-level encryption would protect the confidentiality of both resolvers' queries and the name servers' responses, and 2) while DNSSEC could already be protecting the mapping between a server's domain name and its IP address(es) (via signatures over `A` and `AAAA` RRsets) during name resolution, IPsec would be able to provide the same protections during the actual connection and data transmission. In other words, as Figure 4 shows, DNSSEC would ensure a mapping, and IPsec would ensure the subsequent connection. Moreover, this would include the unprotected DNS header and would also eliminate spoofed responses (like those used in the Kaminsky attack [5]). This overlapping set of protections serves as an illustration that DNSSEC and IPsec are actually complimentary and that their mutual deployment offers a very palpable synergy that neither can achieve without the other.

4 Summary

Our intention in this writing is to demonstrate that securing any single protocol (or protocol layer) does not necessarily mean that nothing can be gained by securing other protocols or layers as well. DNSSEC's charter specifically avoids attempting to offer protections like confidentiality, because these are not goals of the DNS protocol. However, protocols like IPsec not only do this, but help "double-down" on the protections that DNSSEC *is* providing. Similarly, DNSSEC's non-repudiation and resource learning infrastructure offers protections that are beyond the design goals of IPsec. This only further supports the position that as one secures more layers of the security onion, protections can begin to reinforce each other.

References

[1] DANE. `https://datatracker.ietf.org/wg/`

`dane/charter/`.

[2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirement. RFC 4033, March 2005.

[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, March 2005.

[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, March 2005.

[5] CERT. Cert vulnerability note vu#800113, 2008.

[6] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008.

[7] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), December 2005.

[8] Eric Osterweil, Burt Kaliski, Matt Larson, and Danny McPherson. Reducing the X.509 Attack Surface with DNSSEC's DANE. In *Securing and Trusting Internet Names, SATIN '12*, 2012.

[9] Eric Osterweil, Dan Massey, and Lixia Zhang. Managing trusted keys in internet-scale systems. In *The First Workshop on Trust and Security in the Future Internet (FIST'09)*, 2009.

