dprive
Internet-Draft
Intended status: Informational
Expires: 27 August 2021

T. Pauly
Apple Inc.
E. Rescorla
Mozilla
D. Schinazi
Google LLC
C.A. Wood
Cloudflare

23 February 2021

Signaling Authoritative DNS Encryption
draft-rescorla-dprive-adox-latest-00

Abstract

This document defines a mechanism for signaling that a given authoritative DNS server is reachable by encrypted DNS.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the DNS PRIVate Exchange Working Group mailing list (dns-privacy@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dns-privacy/.

Source for this draft and an issue tracker can be found at https://github.com/ekr/draft-rescorla-dprive-adox.

Table of Contents

1. Introduction
2. Conventions and Definitions
3. Overview of Operation
4. Use of SVCB Records to Signal Encrypted Transport
   4.1. Caching and lifetime
   4.2. Authenticating the Server
5. Example
6. Security Considerations
7. IANA Considerations
8. References
   8.1. Normative References
   8.2. Informative References
Acknowledgments
Authors' Addresses

1. Introduction

The IETF has defined a number of mechanisms for carrying DNS queries over encrypted transport [DOH] [DOT] [DOQ]. However, there is no scalable way for a recursive resolver to know that a given authoritative resolver supports encrypted transport, which inhibits the deployment of encrypted DNS for queries from recursive resolvers. This specification defines a mechanism for carrying that signal, using the DNS SVCB [SVCB] record.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Overview of Operation

The mechanism defined in this document works by using the DNS SVCB [SVCB] record to indicate that a given server supports TLS. The recursive resolver can obtain these records in two distinct ways:

* In the additional data block of the response that referred the

recursive to the target authoritative server.

* By directly resolving a SVCB query for the target authoritative resolver.

As a practical matter, the first of these options is preferred as it allows the recursive to learn that the authoritative server supports encrypted transport without an additional round trip, as shown below:

```
Recursive               .com                    ns.example.example
                  Authoritative Server      (Authoritative for example.com)
NS example.com? ------------>

<----- example.com NS ns.example.example
       ns.example.example A 192.0.2.1
       _dns.ns.example.example SVCB alpn=dot

<--------------  TLS connection to ns.example ------------>
A example.com? ------------------------------------------->
```

The recursive resolver starts by contacting the authoritative server for .com and asks for the NS records for example.com. Note that .com is not authoritative for the example.com apex, and will not sign the NS RRset; see [RFC4035], Section 2.2, and Section 6 for details. The authoritative returns the NS record pointing at ns.example.example and also returns a glue records for ns.example.example indicating that it supports DNS over TLS (DoT), in much the same way as it might have sent an IP address for that server. This additional record is the only difference from the current situation, and allows the recursive resolver to know that it can reach ns.example.example over encrypted transport.

Note: SVCB is not presently permitted at the root [REGISTRY]. In the interim, recursive resolvers may be preconfigured with the TLS status of the resolvers for TLDs. [[OPEN ISSUE: Do we want to invent some other sentinel as a temporary measure?]]

4. Use of SVCB Records to Signal Encrypted Transport

Any given authoritative server name can have one or more DNS Server SVCB records, as defined in [I-D.schwartz-svcb-dns].

For instance, the following records would indicate that ns.example.example could be reached by either DoT or DoH (over both TCP and QUIC).

```
_dns.ns.example.example. 7200 IN SVCB 1 ns.example.example. alpn=dot,h2,h3
dohpath=/dns-query{?dns}
```

Upon determining that a given nameserver supports a compatible encrypted transport, an implementation MUST only use encrypted transport for the rest of the cache lifetime for that information and

MUST hard fail with error if it is unable to establish a connection. If multiple encrypted transports are available, an implementation SHOULD try all of them before declaring failure.

[[OPEN ISSUE: figure out error details]]

4.1. Caching and lifetime

Note that in the common case where the name of the target authoritative server is out-of-bailiwick [RFC7719] for the referring resolver, then the SVCB record may not be retained for future queries. This can create a situation in which a given authoritative server will be queried over encrypted transport for one name and over unencrypted transport for another. This is not the end of the world (HTTPS has historically operated in this way, with the security properties being attached to the reference), but is also not ideal. In order to prevent this, resolvers which are also authoritative for their own name SHOULD send SVCB glue records in the additional data section so that they can be properly cached, and the TTL for these SVCB records SHOULD match that of the corresponding NS records in the same RRset.

[[OPEN ISSUE: How often is the case where ns.example.example is not authoritative for itself? Should we encourage people to accept out-of-bailiwick responses in that case?]]

4.2. Authenticating the Server

Recursive servers MUST authenticate the authoritative server using the procedures associated with the relevant protocol, [RFC6125] and [RFC2818] for DoT and DoH respectively. This is in principle compatible with having the server authenticated either with the WebPKI or with TLSA records [DANE], or both. In order for this to work properly, however, the recursive resolver must know at the time it connects whether it will be willing to accept the authoritative server's credentials.

This can be addressed in several ways:

1. Require a particular form of authentication (e.g., the WebPKI or TLSA records) as mandatory.

2. Have the SVCB record indicate what kind(s) of authentication the authoritative server supports, allowing the recursive to filter out incompatible advertisements. For instance, the SVCB record could contain a key that stated that it only had a WebPKI certificate, in which case the resolver could ignore that entry.

[[OPEN ISSUE: If we have the kind of advertisement indicated in (2) above, is not necessary to have an MTI, but it might be desirable to promote interoperability.]]

One challenge with TLSA records in this context is that they may not be in the recursive resolver's cache at the time when it wants to connect to the authoritative. This can create added latency if the recursive resolver must then first retrieve TLSA records for the authoritative. If we wish for servers to authenticate with DANE, we will also probably want some mechanism to carry the TLSA records in the TLS handshake, as, for instance defined in [I-D.ietf-tls-dnssec-chain-extension].

[[OPEN ISSUE: Resolve this.]]

5. Example

A complete example is shown below.

```
Recursive  x.root-servers.org  ns.a.example         ns.example.example
             (Auth. for .)     (Auth for .example)  (Auth for .example.example)

<== TLS handshake ==>
NS .example? ------->
<- .example NS ns.op.example
   ns.op.example A 198.51.100.1
   _dns.ns.op.example SVCB alpn=dot


<============ TLS handshake ===========>
NS example.example? ------------------->
<----- example.example NS ns.example.example
       ns.example.example A 203.0.113.1
       _dns.ns.example.example SVCB alpn=dot

<====================== TLS Handshake ======================>
A www.example.example? ----------------------------------->
<--------------------------- www.example.example A 192.0.2.1
```

In this case, the recursive wants to resolve www.example.example.

Resolution proceeds in three phases.

Initially, the recursive connects to the root server. We assume that the recursive knows that the root server is able to do DoT, either because it has been preconfigured with his information or because it has connected to that root server before. It performs an NS query for ".example" (we are assuming QMIN) and receives:

* An NS record pointing to ns.op.example

* A glue A record for ns.op.example = 198.51.100.1

* A SVCB record stating that ns.op.example speaks DoT

Next, the recursive resolver forms a TLS connection to ns.op.example

and requests an NS record for example.example. It receives:

* An NS record pointing to ns.example.example

* A glue A record for ns.example.example = 203.0.113.1

* A SVCB record stating that ns.example.example speaks DoT

Finally, the recursive resolver forms a TLS connection to ns.example.example and request an A record for www.example.example and receives the A record of 192.0.2.1.

6. Security Considerations

The primary security property delivered by this mechanism is confidentiality of the query and response. As long as (1) all queries in the resolution chain, including to the authoritative server are encrypted and (2) all resolvers in the resolution chain are trustworthy, then even an on-path attacker cannot discover the name being resolved or its response. However, if either of these conditions is violated, then an attack is possible:

* If the connection to the authoritative server is not encrypted, then the request and response can be read directly.

* If one of the earlier connections is not encrypted, then the attacker can substitute their own NS records. records from the additional_data, forcing the resolution back to unencrypted mode.

* If one of the resolvers is untrustworthy, then they can substitute their own NS records.

DNSSEC signing only partly mitigates these issues because delegations at top-level zones are not signed, as per [RFC4035], Section 2.2. In practice, this means a recursive resolver attempting to resolve a zone apex query, such as example.com in Section 3, cannot assume the NS answer is authentic. While NS records received from the authoritative server may be signed, in order to retrieve them, the recursive resolver will have to contact the servers listed by the unverified NS records received from the referring server, at which point it has leaked the zone apex to the (potentially fake) authoritative server (as well as to the referring server).

If the recursive resolver is attempting to resolve a specific subdomain from the resolver (e.g., server-1234.example.com), then it may be able to protect against this attack by (1) using query minimization [QMIN] and (2) querying the (alleged) authoritative for its DNSSEC-signed NS and SVCB records and only once it has received those, attempting to retrieve the actual subdomain. If the domain is DNSSEC signed, then this prevents a malicious referring resolver from redirecting the recursive resolver to their own authoritative and learning the true subdomain. However, if, as is common, the

recursive is just trying to resolve the apex name or one of the common "service" names such as "www.example.com", then this procedure does not provide additional protection.

Encryption does not mitigate all leakage. In some circumstances, an on-path attacker may learn the identity of the authoritative server if, for example, that server only serves a small number of domains. The attacker can learn information about what is being resolved by observing whether or not server is queried.

As a secondary property, the mechanism in this document can provide some level of integrity for DNS responses, again under the condition that each resolver in the chain is trustworthy. By contrast, DNSSEC provides integrity even if the resolvers are untrustworthy.

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[DANE] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://datatracker.ietf.org/doc/html/rfc6698>.

[DOH] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://datatracker.ietf.org/doc/html/rfc8484>.

[DOT] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://datatracker.ietf.org/doc/html/rfc7858>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://datatracker.ietf.org/doc/html/rfc2119>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://datatracker.ietf.org/doc/html/rfc2818>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://datatracker.ietf.org/doc/html/rfc6125>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://datatracker.ietf.org/doc/html/rfc8174>.

8.2. Informative References

[DOQ] Huitema, C., Mankin, A., and S. Dickinson, "Specification of DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnsoquic-02, 22 February 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnsoquic-02.txt>.

[I-D.ietf-tls-dnssec-chain-extension] Shore, M., Barnes, R., Huque, S., and W. Toorop, "A DANE Record and DNSSEC Authentication Chain Extension for TLS", Work in Progress, Internet-Draft, draft-ietf-tls-dnssec-chain-extension-07, 21 March 2018, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-dnssec-chain-extension-07.txt>.

[I-D.schwartz-svcb-dns] Schwartz, B., "Service Binding Mapping for DNS Servers", Work in Progress, Internet-Draft, draft-schwartz-svcb-dns-02, 17 February 2021, <https://datatracker.ietf.org/doc/html/draft-schwartz-svcb-dns-02.txt>.

[QMIN] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <https://datatracker.ietf.org/doc/html/rfc7816>.

[REGISTRY] ICANN, "ICANN Registry Agreement", July 2017, <https://newgtlds.icann.org/sites/default/files/agreements/agreement-approved-31jul17-en.html#exhibitA.1>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://datatracker.ietf.org/doc/html/rfc4035>.

[RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <https://datatracker.ietf.org/doc/html/rfc7719>.

[SVCB] Schwartz, B., Bishop, M., and E. Nygren, "Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)", Work in Progress, Internet-Draft, draft-ietf-dnsop-svcb-https-03, 17 February 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-svcb-https-03.txt>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Tommy Pauly
Apple Inc.

Email: tpauly@apple.com

Eric Rescorla
Mozilla

Email: ekr@rtfm.com

David Schinazi
Google LLC

Email: dschinazi.ietf@gmail.com

Christopher A. Wood
Cloudflare

Email: caw@heapingbits.net