DNSOP
Internet-Draft
Updates: 4035 (if approved)
Intended status: Informational
Expires: November 1, 2020

P. Wouters
Red Hat
W. Hardaker
USC/ISI

April 30, 2020

The DELEGATION_ONLY DNSKEY flag
draft-pwouters-powerbind-04

Abstract

This document introduces a new DNSKEY flag called DELEGATION_ONLY that indicates that the particular zone will never sign zone data across a label. That is, every label (dot) underneath is considered a zone cut and must have its own (signed) delegation. Additionally, it indicates the zone is expecting its parent to never bypass or override the zone. DNSSEC Validating Resolvers can use this flag to mark any data that violates the DELEGATION_ONLY policy as BOGUS.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 1, 2020.

Table of Contents

1. Introduction

The DNS Security Extensions [DNSSEC] use public key cryptography to create an hierarchical trust base with the DNSSEC root public keys at the top, followed by Top Level domain (TLD) keys one level underneath. While the root and most TLD zones are asumed to be exclusively delegation-only zones, there is currently no interoperable and automatable mechanism for auditing these zones to ensure they behave as a delegation-only zone. This creates a target for malicious use of these zones - either by their owners or through coercion.

This document defines a mechanism for delegation-only zone owners to create a DNSKEY that indicate they will only delegate the remainder of the DNS tree to lower-level zones. This allows for easier delegation policy verification and logging and auditing of DNS responses served by their infrastructure.

Specifically, this document introduces a new DNSKEY flag allowing zone owners to commit to only signing records relating to delegation.

If a DNSSEC validator discovers any non-delegation zone data signed by a flagged key, this data can be interpreted as BOGUS.

2. Terminology

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. The Deep Signing problem

The hierarchical model of DNS and DNSSEC ([RFC1034], [RFC1035], [RFC4033], [RFC4034] and [RFC4035]) comes with the property that a zone at one point in the hierarchy can define, and therefor override, everything below it in the DNS tree. And this is possible to achieve on a per-client basis.

For example, the owner of the DNSSEC root key could generate a specially crafted zone file that ignores the intended NS records for ".org" and "example.org". It could place a AAAA record for "www.example.org" directly into the specially crafted zone, with a corresponding RRSIG signed by the root DNSKEY itself. Validating resolvers would find this record perfectly acceptable, as it was signed by a key within the proper chain of trust (in this case, a root DNSKEY). This specially crafted zone could then even be served to specific clients in an effort to subvert a portion of the DNS ecosystem for a portion of the Internet.

Similarly, the TLD "example" could circumvent company.example for certain clients. It is important to note that this can be done without changing the upstream DS record or trust anchor -- the DNSKEY is (again) already in the trust path and is merely signing deeper DNS records than the zone owner's clients may have expected it to sign.

It is important to note that this "feature" has always been present. Since the creation of the DNS, it has always been possible to serve "split zones". Specifically, it is not a problem created by DNSSEC -- DNSSEC was not designed to protect against this use case.

Exposing such targeted attacks requires a transparency audit infrastructure similar to what is deployed for PKIX ([RFC6962]). However, a DNSSEC version would need to log significantly more data, as all signed DNS data used by a DNSKEY must be recorded in order to prove that data signed by a parent zone's DNSKEY was out of expected policy. The very distributed nature of the DNS combined with the

typically frequent zone resigning rate makes such transparency logs prohibitively expensive and nearly impossible to operate.

Additionally, it would require zone owners to expose all their zone data to any public log operators, thereby introducing privacy implications and exposing all relevant DNS data to a public archive. This may be acceptable for some domains, such as the root, where DNS data is already considered public. However, other delegation domains have legal implications that prohibit them from participating in such a system.

Furthermore, there is no signaling mechanism that lets validating resolvers know which zones are supposedly delegation-only, and what zones can be logged. Today there are over 1500 TLDs in the root zone, some of which may be considered delegation-only while others may not be. At the time of this writing, the list of entries in the public suffix list contains over 8800 entries as well, with 73 wild-card entries (prefixed with a "*") indicating that all of their (unknown number of) children are public registration points. In the absence of an interoperable mechanism (like this draft provides), it is infeasible that a validating resolver or auditing log could know which of these zones are delegation-only without individual policy statements from each of them. [todo: xref psl]

3.1. Affected parties and their roles

Upon deployment of this specification, the following parties would be potentially benefit or be affected by:

Authoritative parent: If (and only if) an authoritative parent is a "delegation only" zone, it could generate a DNSKEY with the DELEGATION_ONLY flag set, indicating a verifiable promise to the world that will not sign any records other than delegation records.

Authoritative Child / Delegated Zone: child zones existing underneath a marked delegation-only zone get the added benefit of knowing their parent will not (and cannot) sign DNS records within the child's portion of the DNS tree using the marked DNSKEY.

Validating Resolver: A validating that supports verifying the DELEGATION_ONLY flag is capable of rejecting or discarding any data from an authoritative parent that incorrectly signs non-delegation records too low in the DNS tree. If the validating resolver supports a (future-defined) DNSSEC transparency audit log as well, it may submit the appropriate data to a DNSSEC transparency log that appropriately tracks DNSSEC signatures.

DNSSEC Transparency Log (optional future): a DNSSEC transparency log would create a non-modifiable trace of log entries submitted to it, for public verification, similar to [RFC6962]. What it chooses to accept into its log might be only certain zone data, or any zone with a marked DNSKEY.

Note that without a DNSSEC Log, the DELEGATION_ONLY flag is still useful per the discussion in the Validating Resolvers role: the resolver will reject incorrectly signed, non-delegation data. However, malicious parent zones are still capable of creating two (or more) DNSKEYs, one with the DELEGATION_ONLY flag and one without. However, they would also have to publish those DS records as well, which is detectable by DNSSEC monitoring platforms, regardless of the existence of a DNSSEC Transparency Log. Any zone with multiple DS records that link to both a DELEGATION_ONLY marked and an unmarked DNSKEY would be considered suspicious (or at least in transition). Circumventing this through obfuscation would require the collusion of their parent as well. Finally, a DELEGATION_ONLY flagged DNSKEY for the root zone cannot be overridden easily, as it would require a trust anchor update in all validating resolvers.

4. The DELEGATION_ONLY DNSKEY flag

This document introduces a new DNSKEY flag called DELEGATION_ONLY. When this flag is set on a DNSKEY with its Secure Entry Point (SEP) flag set, the zone owner commits to not sign any data that crosses a label down in the hierarchy. This commits a parent in the DNS hierarchy to only publish signed DS records and unsigned glue records (NS and A/AAAA) for its child zones. It will no longer be able to ignore (or briefly delete, see below) a child delegation and publish data crossing zone labels by pretending the next label is not a zone cut.

For such a parent to take over data that belongs to its child zone, it has two choices. It can (temporarilly) remove its own DNSKEY DELEGATION_ONLY flag or it can replace the NS and DS records of its child zone with its own data (destinations and key references) so it can sign DNS data that belongs to its own child zone. However, both of these actions cannot be hidden, thus exposing such malicious behavior when combined with DNSSEC Transparency logs.

A zone that publishes a DNSKEY with the DELEGATION_ONLY flag also signifies that it is not expecting its own parent to skip it, thereby bypassing its DELEGATION_ONLY flag.

5. _underscore label exception

Some protocols, such as the DANE protocol [RFC6698] use a number of labels that start with an underscore (_) prefix to publish information about the zone itself. For example, the TLSA record for www.example.com is published at the location _443._tcp.www.example.com. These records are semantically part of the zone itself and are not delegated child zones. Any chain of labels that each start with an underscore (_) is not considered to violate the DELEGATION_ONLY flag limitation of being DELEGATION_ONLY, as this data is logically part of the zone itself and is never meant to be interpreted as an independent delegated child zone.

6. Parental Transparency

A parent zone, such as the root zone, a TLD or any public suffix list delegation point, that has published a key with the DELEGATION_ONLY flag can no longer make an exception for a single delegated zone without removing the DELEGATION_ONLY flag, switching off its published policy. This action would be highly visible, and for some domains such as the root or TLDs, require human interaction to notify the stake holders to prevent loss of trust.

Removing the DELEGATION_ONLY flag from a DNSKEY requires that the zone first publishes an additional updated DS record to its parent.

In the case of the root key, it would require updating out-of-band root key meta information and/or perform an [RFC5011] style rollover for the same key with updated DNSKEY flags. Due to the timings of such a rollover, it would take at least 30 days for the first validating resolvers to pick up this policy change. It would also be a highly visible event.

Replacing the NS and DS records of a child zone can still be done in a targeted attack mode, but these events are something that can be easilly tracked by a transparency infrastructure similar to what is now in use for the WebPKI using [RFC6962](bis). With client implementations of transparency, all DELEGATION_ONLY flag changes would be logged and become visible to the owner of attacked child zones, exposing a parent's malicious behaviour.

7. Marking zone keys DELEGATION_ONLY

Even before a parent DNSKEY (or the root key) has set the DELEGATION_ONLY flag, zones can already signal their own willingness to commit to being DELEGATION_ONLY zones. Any changes of that state in a zone DNSKEY will require those zones to submit a new DS record to their parent. Setting the DELEGATION_ONLY flag would trigger

DNSSEC Transparency clients to start monitoring for actions by the zone or its parents that would be bypassing the DELEGATION_ONLY policy of the zone. Validating resolvers would mark any data in violation of the DELEGATION_ONLY policy as BOGUS.

7.1. Marking the Root DNSKEY DELEGATION_ONLY

Once the Root DNSKEY is marked with a DELEGATION_ONLY flag and deployed resolvers are configured with the new DNSKEY, all TLDs will be ensured that the Root DNSKEY can no longer be abused to override child zone data. Until the Root KSK DNSKEY sets this flag, software SHOULD imply this flag is always set, as this is the current expectation of the Root Zone.

7.2. Migrating to and from DELEGATION_ONLY

There might be multiple DNSKEYs with the SEP flag set in a zone. For the purpose of declaring a zone as DELEGATION_ONLY, only those DNSKEY's that have a corresponding DS record at the parent MUST be considered. If multiple DS records appear at the parent, some of which point to DNSKEYs with and some of which point to DNSKEYs without the DELEGATION_ONLY flag set, the zone MUST be considered DELEGATION_ONLY. This situation will occur when a zone is rolling its DNSKEY key at the same time as it is committing to a DELEGATION_ONLY zone (or the reverse).

8. Operational Considerations

Setting or unsetting the DELEGATION_ONLY flag must be handled like any other Key Signing Key rollover procedure, with the appropriate wait times to give resolvers the chance to update their caches.

Some TLDs offer a service where small domains can be hosted in-zone at the TLD zone itself. In that case, the TLD MUST NOT set the DELEGATION_ONLY flag. Another solution for such TLDs is to create delegations for these child zones with the same or different DNSKEY as used in the parent zone itself.

If a zone is publishing glue records for a number of zones, and the zone that contains the authoritative records for this glue is deleted, a resigning of the zone will make this orphaned glue authoritative within the zone. However, with the DELEGATION_ONLY flag set, this (signed) DNSSEC data will be considered BOGUS as it violations the commitment to only delegate. This may impact domains that depended on this unsigned glue. Note that glue handling differs per zone. Some TLDs already remove the glue records if no authoritative child is left in its zone that matches these glue records.

For example, if "example.com" and "example.net" use NS records pointing to "ns.example.net", then if "example.net" is deleted from the ".net" zone, and the previously unsigned glue of "ns.example.net" is now signed by the ".net" zone, the "example.com" zone will lose its NS records and fail to resolve.

If a domain uses Empty Non Terminals (ENT), that is uses multiple labels where the label is not covered by its own delegation, then the DELEGATION_ONLY flag cannot be set. For example, some domains allow registrations straight into their zone (eg "child.example") while others use an ENT to categorize these (eg "child.co.example" and "child.ac.example"). Some TLDs contain a few ENTs marking some administrative or geographic region. Such TLDs must first migrate their ENT to fully delegated child zones before enabling the DELEGATION_ONLY flag.

Some TLDs publish their nameserver (NS) records straight within their TLD (eg "ns1.example") which makes these names indistinguishable from real relegations with respect to the DELEGATION_ONLY flag. These NS entries would have to be moved to their own delegation zone (eg "ns1.nic.example")

Some TLDs have a requirement for certain Fully Qualified Domain Names (FQDN) within their TLD, such as "whois.example" or "nic.example". These usually appear as signed data of the TLD and not as a delegated child zone. These names would have to be converted to delegated zones before enabling the DELEGATION_ONLY flag.

The bind DNS software has an option called "delegation_only zones" which is an option that means something completely different. It refers to ignoring wildcard records in specified zones that are deemed delegation-only zones.

9. Security Considerations

Some parental attacks cannot be detected when the validating resolver's cache is empty. Care should be taken by resolvers to not unneccessarily empty their cache. This is specifically important for roaming clients that re-connect frequently to different wireless or mobile data networks.

This DELEGATION_ONLY mechanism is not designed to completely foil attacks (since parent's can simply change a child's referral data), but rather to empower transparency logging mechanisms.

10. Privacy Considerations

Some of the protection offered by the DELEGATION_ONLY flag is only available when DNS resolvers report changes in the signing depth of high level (root or TLD) DNSKEYs to gain DNSSEC Transparency. This reporting can reveal that a particular node is trying to access a certain DNS name. Defensive measures to prevent exposing users should be taken when implementing DNSSEC Transparency. It is expected that DNSSEC Transparency behaviour will be written up in a separate document.

11. Human Rights Considerations

The DNS protocol's hierarchy limits zones authority to themselves and their child zones only. While this provides a finer grained trust model compared to a simple list of trusted entities, such as used in the WebPKI, it consolidates a lot of power in the top of the DNS hierarchy. With the increased reliance on DNSSEC for securely identifying resources, such as DANE records, it becomes very important to audit those entities high up in the hierarchy to not abuse or be co-erced into abusing control of the independent child zones. The protocol extension specifically aims at increasing parental transparency and blocks some parental attacks from those parents who have publicly claimed to never override their child zone data.

Parents using the DELEGATION_ONLY flag publication to increase their public trust are still able to remove child zones from their zone, for example in cases of legal compliance or to prevent malicious activity happening in its child zone. But these parents can only do so publicly and can no longer surreptitiously take control of their own child zones.

12. IANA Considerations

This document defines a new DNSKEY flag, the DELEGATION_ONLY flag, whose value [TBD] has been allocated by IANA from the DNSKEY FLAGS Registry.

13. Acknowledgements

The authors wishes to thank Thomas H. Ptacek for his insistence on this matter.

Thanks to the following IETF participants: Viktor Dukhovni, Shumon Huque, Geoff Huston, Rick Lamb and Sam Weiler.

14. References

14.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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://www.rfc-editor.org/info/rfc4035>.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <https://www.rfc-editor.org/info/rfc5011>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

14.2. Informative References

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC6698] 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://www.rfc-editor.org/info/rfc6698>.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/info/rfc6962>.

Authors' Addresses

Paul Wouters
Red Hat

Email: pwouters@redhat.com

Wes Hardaker
USC/ISI
P.O. Box 382
Davis, CA 95617
US

Email: ietf@hardakers.net