Workgroup: Network Working Group
Internet-Draft:
draft-ietf-dnsop-nsec3-guidance-07
Published: 7 April 2022
Intended Status: Best Current Practice
Expires: 9 October 2022
Authors: W. Hardaker V. Dukhovni
USC/ISI Bloomberg, L.P.

Guidance for NSEC3 parameter settings

Abstract

NSEC3 is a DNSSEC mechanism providing proof of non-existence by promising there are no names that exist between two domainnames within a zone. Unlike its counterpart NSEC, NSEC3 avoids directly disclosing the bounding domainname pairs. This document provides guidance on setting NSEC3 parameters based on recent operational deployment experience.

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 9 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction
  1.1. Requirements Notation
2. NSEC3 Parameter Value Considerations
  2.1. Algorithms
  2.2. Flags
  2.3. Iterations
  2.4. Salt
3. Recommendations for Deploying and Validating NSEC3 Records
  3.1. Best-practice for Zone Publishers
  3.2. Recommendation for Validating Resolvers
  3.3. Recommendation for Primary / Secondary Relationships
4. Security Considerations
5. Operational Considerations
6. IANA Considerations
7. References
  7.1. Normative References
  7.2. Informative References
Appendix A. Deployment measurements at time of publication
Appendix B. Computational burdens of processing NSEC3 iterations
Appendix C. Acknowledgments
Appendix D. Github Version of This Document
Appendix E. Implementation Notes
  E.1. OpenDNSSEC
  E.2. PowerDNS
  E.3. Knot DNS and Knot Resolver
  E.4. Google Public DNS Resolver
  E.5. Google Cloud DNS
Authors' Addresses

1. Introduction

As with NSEC [RFC4035], NSEC3 [RFC5155] provides proof of non-existence that consists of signed DNS records establishing the non-existence of a given name or associated Resource Record Type (RRTYPE) in a DNSSEC [RFC4035] signed zone. In the case of NSEC3, however, the names of valid nodes in the zone are obfuscated through (possibly multiple iterations of) hashing (currently only SHA-1 is in use within the Internet).

NSEC3 also provides "opt-out support", allowing for blocks of unsigned delegations to be covered by a single NSEC3 record. Use of the opt-out feature allow large registries to only sign as many NSEC3 records as there are signed DS or other RRsets in the zone - with opt-out, unsigned delegations don't require additional NSEC3

records. This sacrifices the tamper-resistance proof of non-existence offered by NSEC3 in order to reduce memory and CPU overheads.

NSEC3 records have a number of tunable parameters that are specified via an NSEC3PARAM record at the zone apex. These parameters are the hash algorithm, processing flags, the number of hash iterations and the salt. Each of these has security and operational considerations that impact both zone owners and validating resolvers. This document provides some best-practice recommendations for setting the NSEC3 parameters.

1.1. Requirements Notation

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.

2. NSEC3 Parameter Value Considerations

The following sections describe recommendations for setting parameters for NSEC3 and NSEC3PARAM.

2.1. Algorithms

The algorithm field is not discussed by this document.

2.2. Flags

The NSEC3PARAM flags field currently contains no flags, but individual NSEC3 records contain the "Opt-Out" flag [RFC5155], which specifies whether or not that NSEC3 record provides proof of non-existence or not. In general, NSEC3 with the Opt-Out flag enabled should only be used in large, highly dynamic zones with a small percentage of signed delegations. Operationally, this allows for fewer signature creations when new delegations are inserted into a zone. This is typically only necessary for extremely large registration points providing zone updates faster than real-time signing allows or when using memory-constrained hardware. Smaller zones, or large but relatively static zones, are encouraged to use a flags value of 0 (zero) and take advantage of DNSSEC's proof-of-non-existence support.

2.3. Iterations

NSEC3 records are created by first hashing the input domain and then repeating that hashing algorithm a number of times based on the iteration parameter in the NSEC3PARM and NSEC3 records. The first

hash is typically sufficient to discourage zone enumeration performed by "zone walking" an NSEC or NSEC3 chain. Only determined parties with significant resources are likely to try and uncover hashed values, regardless of the number of additional iterations performed. If an adversary really wants to expend significant CPU resources to mount an offline dictionary attack on a zone's NSEC3 chain, they'll likely be able to find most of the "guessable" names despite any level of additional hashing iterations.

Most names published in the DNS are rarely secret or unpredictable. They are published to be memorable, used and consumed by humans. They are often recorded in many other network logs such as email logs, certificate transparency logs, web page links, intrusion detection systems, malware scanners, email archives, etc. Many times a simple dictionary of commonly used domain names prefixes (www, ftp, mail, imap, login, database, etc) can be used to quickly reveal a large number of labels within a zone. Because of this, there are increasing performance costs yet diminishing returns associated with applying additional hash iterations beyond the first.

Although Section 10.3 of [RFC5155] specifies upper bounds for the number of hash iterations to use, there is no published guidance for zone owners about good values to select. Recent academic studies have shown that NSEC3 hashing provides only moderate protection [GPUNSEC3][ZONEENUM].

2.4. Salt

NSEC3 records provide an additional salt value, which can be combined with an FQDN to influence the resulting hash, but properties of this extra salt are complicated.

In cryptography, salts generally add a layer of protection against offline, stored dictionary attacks by combining the value to be hashed with a unique "salt" value. This prevents adversaries from building up and remembering a single dictionary of values that can translate a hash output back to the value that it derived from.

In the case of DNS, the situation is different because the hashed names placed in NSEC3 records are always implicitly "salted" by hashing the fully-qualified domain name from each zone. Thus, no single pre-computed table works to speed up dictionary attacks against multiple target zones. An attacker is always required to compute a complete dictionary per zone, which is expensive in both storage and CPU time.

To understand the role of the additional NSEC3 salt field, we have to consider how a typical zone walking attack works. Typically, the attack has two phases - online and offline. In the online phase, an

attacker "walks the zone" by enumerating (almost) all hashes listed in NSEC3 records and storing them for the offline phase. Then, in the offline cracking phase, the attacker attempts to crack the underlying hash. In this phase, the additional salt value raises the cost of the attack only if the salt value changes during the online phase of the attack. In other words, an additional, constant salt value does not change the cost of the attack.

Changing a zone's salt value requires the construction of a complete new NSEC3 chain. This is true both when resigning the entire zone at once, or when incrementally signing it in the background where the new salt is only activated once every name in the chain has been completed. As a result, re-salting is a very complex operation, with significant CPU time, memory, and bandwidth consumption. This makes very frequent re-salting impractical, and renders the additional salt field functionally useless.

3. Recommendations for Deploying and Validating NSEC3 Records

The following subsections describe recommendations for the different operating realms within the DNS.

3.1. Best-practice for Zone Publishers

First, if the operational or security features of NSEC3 are not needed, then NSEC SHOULD be used in preference to NSEC3. NSEC3 requires greater computational power (see Appendix B) for both authoritative servers and validating clients. Specifically, there is a nontrivial complexity in finding matching NSEC3 records to randomly generated prefixes within a DNS zone. NSEC mitigates this concern. If NSEC3 must be used, then an iterations count of 0 MUST be used to alleviate computational burdens. Note that extra iteration counts other than 0 increase the impact of CPU-exhausting DoS attacks, and also increase the risk of interoperability problems.

Note that deploying NSEC with minimally covering NSEC records [RFC4470] also incurs a cost, and zone owners should measure the computational difference in deploying both RFC4470 or NSEC3.

In short, for all zones, the recommended NSEC3 parameters are as shown below:

```
; SHA-1, no extra iterations, empty salt:
;
bcp.example. IN NSEC3PARAM 1 0 0 -
```

For small zones, the use of opt-out based NSEC3 records is NOT RECOMMENDED.

For very large and sparsely signed zones, where the majority of the records are insecure delegations, opt-out MAY be used.

Operators are encouraged to forgo using a salt entirely by using a zero-length salt value instead (represented as a "-" in the presentation format).

If salts are used, note that since the NSEC3PARAM RR is not used by validating resolvers (see [RFC5155] section 4), the iterations and salt parameters can be changed without the need to wait for RRsets to expire from caches. A complete new NSEC3 chain needs to be constructed and the zone resigned.

3.2. Recommendation for Validating Resolvers

Because there has been a large growth of open (public) DNSSEC validating resolvers that are subject to compute resource constraints when handling requests from anonymous clients, this document recommends that validating resolvers change their behavior with respect to large iteration values. Specifically, validating resolver operators and validating resolver software implementers are encouraged to continue evaluating NSEC3 iteration count deployments and lower their default acceptable limits over time. Similarly, because treating a high iterations count as insecure leaves zones subject to attack, validating resolver operators and validating resolver software implementers are further encouraged to lower their default and acceptable limit for returning SERVFAIL when processing NSEC3 parameters containing large iteration count values. See Appendix A for measurements taken near the time of publication and potential starting points.

Validating resolvers MAY return an insecure response to their clients when processing NSEC3 records with iterations larger than 0. Note also that a validating resolver returning an insecure response MUST still validate the signature over the NSEC3 record to ensure the iteration count was not altered since record publication (see [RFC5155] section 10.3).

Validating resolvers MAY also return a SERVFAIL response when processing NSEC3 records with iterations larger than 0. Validating resolvers MAY choose to ignore authoritative server responses with iteration counts greater than 0, which will likely result in returning a SERVFAIL to the client when no acceptable responses are received from authoritative servers.

Validating resolvers returning an insecure or SERVFAIL answer to their client after receiving and validating an unsupported NSEC3 parameter from the authoritative server(s) SHOULD return an Extended DNS Error (EDE) {RFC8914} EDNS0 option of value (RFC EDITOR: TBD).

Validating resolvers that choose to ignore a response with an unsupported iteration count (and do not validate the signature) MUST NOT return this EDE option.

Note that this specification updates [RFC5155] by significantly decreasing the requirements originally specified in Section 10.3 of [RFC5155]. See the Security Considerations for arguments on how to handle responses with non-zero iteration count.

3.3. Recommendation for Primary / Secondary Relationships

Primary and secondary authoritative servers for a zone that are not being run by the same operational staff and/or using the same software and configuration must take into account the potential differences in NSEC3 iteration support.

Operators of secondary services should advertise the parameter limits that their servers support. Correspondingly, operators of primary servers need to ensure that their secondaries support the NSEC3 parameters they expect to use in their zones. To ensure reliability, after primaries change their iteration counts, they should query their secondaries with known non-existent labels to verify the secondary servers are responding as expected.

4. Security Considerations

This entire document discusses security considerations with various parameters selections of NSEC3 and NSEC3PARAM fields.

The point where a validating resolver returns insecure vs the point where it returns SERVFAIL must be considered carefully. Specifically, when a validating resolver treats a zone as insecure above a particular value (say 100) and returns SERVFAIL above a higher point (say 500), it leaves the zone subject to attacker-in-the-middle attacks as if it was unsigned between these values. Thus, validating resolver operators and software implementers SHOULD set the point above which a zone is treated as insecure for certain values of NSEC3 iterations counts to the same as the point where a validating resolver begins returning SERVFAIL.

5. Operational Considerations

This entire document discusses operational considerations with various parameters selections of NSEC3 and NSEC3PARAM fields.

6. IANA Considerations

This document requests a new allocation in the "Extended DNS Error Codes" of the "Domain Name System (DNS) Parameters" registration table with the following characteristics:

*INFO-CODE: (RFC EDITOR: TBD)

*Purpose: Unsupported NSEC3 iterations value

*Reference: (RFC EDITOR: this document)

7. References

7.1. Normative References

[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>.

[RFC4470] Weiler, S. and J. Ihren, "Minimally Covering NSEC Records and DNSSEC On-line Signing", RFC 4470, DOI 10.17487/RFC4470, April 2006, <https://www.rfc-editor.org/info/rfc4470>.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <https://www.rfc-editor.org/info/rfc5155>.

7.2. Informative References

[GPUNSEC3] Wander, M., Schwittmann, L., Boelmann, C., and T. Weis, "GPU-Based NSEC3 Hash Breaking", DOI 10.1109/NCA.2014.27, 2014, <https://doi.org/10.1109/NCA.2014.27>.

[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>.

[ZONEENUM] Wang, Z., Xiao, L., and R. Wang, "An efficient DNSSEC zone enumeration algorithm", n.d..

Appendix A. Deployment measurements at time of publication

At the time of publication, setting an upper limit of 100 iterations for treating a zone as insecure is interoperable without significant problems, but at the same time still enables CPU-exhausting DoS attacks.

At the time of publication, returning SERVFAIL beyond 500 iterations appears to be interoperable without significant problems.

Appendix B. Computational burdens of processing NSEC3 iterations

The queries per second (QPS) of authoritative servers will decrease due to computational overhead when processing DNS requests for zones containing higher NSEC3 iteration counts. The table (Appendix C) below shows the drop in QPS for various iteration counts.

Iterations	QPS [% of 0 iterations QPS]
0	100 %
10	89 %
20	82 %
50	64 %
100	47 %
150	38 %

Appendix C. Acknowledgments

The authors would like to thank the dns-operations discussion participants, which took place on mattermost hosted by DNS-OARC.

Additionally, the following people contributed text or review comments to the draft:

*Vladimir Čunat

*Tony Finch

*Paul Hoffman

*Alexander Mayrhofer

*Matthijs Mekking

*Florian Obser

*Petr Špaček

*Paul Vixie

*Tim Wicinski

Appendix D. Github Version of This Document

While this document is under development, it can be viewed, tracked, issued, pushed with PRs, ... here:

https://github.com/hardaker/draft-hardaker-dnsop-nsec3-guidance

Appendix E. Implementation Notes

The following implementations have implemented the guidance in this document. They have graciously provided notes about the details of their implementation below.

E.1. OpenDNSSEC

The OpenDNSSEC configuration checking utility will alert the user about nsec3 iteration values larger than 100.

E.2. PowerDNS

PowerDNS 4.5.2 changed the default value of nsec3-max-iterations to 150.

E.3. Knot DNS and Knot Resolver

Knot DNS 3.0.6 warns when signing with more than 20 NSEC3 iterations. Knot Resolver 5.3.1 treats NSEC3 iterations above 150 as insecure.

E.4. Google Public DNS Resolver

Google Public DNS treats NSEC3 iterations above 100 as insecure since September 2021.

E.5. Google Cloud DNS

Google Cloud DNS uses 1 iteration and 64-bits of fixed random salt for all zones using NSEC3. These parameters cannot be adjusted by users.

Authors' Addresses

Wes Hardaker
USC/ISI

Email: ietf@hardakers.net

Viktor Dukhovni

Bloomberg, L.P.

Email: ietf-dane@dukhovni.org