.ca Signing Metrics

R. Gieben[*]
NLnet Labs

NLnet Labs document 2006-001 May 2, 2006

Abstract

In this report we look at the signing, incremental signing and parallel signing characteristics of the .ca zone. Although the .ca zone is used for the tests the results should be applicable to other large (TLD) zones.

Signing of .ca takes about half an hour on commodity hardware. The amount of time the signing process takes depends on a number of factors; key sizes, size of the zone and the amount of (re)generated signatures. In this paper the .ca is signed in a number of different ways to show the impact these factors have on the signing time and the resulting zone size.

Contents

1 Introduction

DNSSEC [1, 3, 2] is a security extension to the Domain Name System (DNS). It provides authentication and integrity through signatures over resource records (RRs).

In DNSSEC DNS resource records are signed using private keys. The signatures are published in the DNS as RRSIG resource records. The public keys that are needed to validate the signatures are published as DNSKEY resource records.

[*] miek@nlnetlabs.nl

File information:Id: ca-reg.tex 155 2006-01-31 08:39:08Z miekg

Signed zones are created by taking plain DNS zones and running these through a *signer*. This document looks into the characteristics of signing and resigning of the `.ca` zone. Though we looked at this specific zone, the results are generic enough to be applicable to other TLD zones. The signer used to carry out the experiments was from BIND version 9.3.1.

1.1 Zone (Re)signing

Each generated signature has a cut-off date (signature expiration date), after which the signature becomes invalid. To prevent signatures from being regenerated each time one signs the zone, `dnssec-signzone` has feature which looks at this expiration date and then decides if signatures need to be recalculated.

This feature (called incremental signing) can significantly speedup the signing process because signature generation is an expensive operation.

The process of zone signing adds various records to the zone (RRSIG and NSEC records). Because of these additions the zone size grows considerable, depending on the key size (which directly influences the generated RRSIG size), this can be several orders of magnitude. See [5] for more details on this.

2 Experimental Setup and Tools

The hardware used for the experiment is a dual Pentium III 1400 Mhz, with 3 GB of RAM and running Debian GNU/Linux with a 2.6 kernel. This is reasonable commodity hardware.[1]

The `.ca` zone is from the 15th of December 2005. It amounts to 63 MB (on disk) and has 1,361,002 resource records. The amount of RRsets in this zone is 612,409[2]. Which should roughly match the number of delegated names under `.ca`.

With `dnssec-keygen` a number of keys were generated, each with a different length. During the signing the device `/dev/urandom` was used.

In total three kinds of experiments are run:

1. Sign the entire zone, with different key sizes, see section 3.

2. Incremental sign the `.ca` zone, see Section 4.

3. Parallel sign the zone. No actual testing is done, see Section 5.

Each section starts of with an outline of the experimental setup. After that we detail the gathered data.

3 Signing of the entire `.ca` Zone

We signed the `.ca` zone with a number of keys with a different length to show the correlation between key length and sign time. Each time the zone was fully signed, i.e. no incremental signing was done.

key size (b)	time elapsed (m)	size (MB)	size increase
1024	29:09	238	3.7
1280	52:19	266	4.2
1584	1:24:50	300	4.8

Table 1: Signing metrics.

Table 1 shows a rundown of the metrics found.

Note how a small key size increase greatly effects the sign time. A rule of thumb for RSA's time requirements [3] is:

- n^2 for public key operations (verification).
- n^3 for private key operations (signing).
- n^4 for key generation.

Where n is the key size.

One way to use larger keys is to use a larger key as the key signing key and a smaller one as the zone signing key. See [4] for more information on this subject.

4 Incremental Signing of the `.ca` Zone

To measure the incremental signing times we have split up the zone and each part was then signed with a different expiration date. For the splitting and reassembly of the `.ca` zone two tools based on the `ldns` [4] library are used:[5]

`ldns-zsplit`: Splits a zone into smaller pieces. Each piece is a complete zone with the original SOA placed at the apex.

`ldns-zcat`: Take a number of previously (z)split up zones and create a new zone.

These two utilities allow for parallel signing of large zones and the can be used to test the incremental signing capabilities of the signer.

The `.ca` zone was cut into five equivalent sized pieces. Each piece holds about 272205(+/- 10) RRs. Each piece was then signed with `dnssec-signzone`. Because we had five different parts of the `.ca` zone we could change the signature expiration time in each part, thereby allowing us to test `dnssec-signzone` with a zone that needed: no, 20%, 40% and 80% resigning.

[1] RIPE NCC provided us with this equipment.

[2] Calculated with: `cat ca.today | awk ' { print $1" "$4 } ' | uniq | wc`.

[3] Personal (email) conversation with Sam Weiler.

[4] `http://www.nlnetlabs.nl/ldns/`

[5] The tools will be distributed with ldns in the future and currently available from the subversion repository: `http://www.nlnetlabs.nl/ldns/svn/trunk/examples/`

4.1 Incremental Signing Metrics

DNSSEC requires that a zone is sorted. If the signer signs an unsorted zone the sign time can become unacceptable long. Sorting a zone before signing helps in this regard. As the `.ca` zone was already sorted this hasn't been a problem.

The actual splitting of the `.ca` zone took about one (1) minute on our hardware. The reassembly of (the now signed and larger) parts took about 2.5 minutes. Table 2 shows how the to amount of regenerated signatures relate to the elapsed sign time.

regenerated sigs (%)	key size (b)	time elasped (m)
0	1024	3:22
20	1024	8:50
40	1024	14:13
80	1024	25:03
100	1024	29:09

Table 2: Incremental signing metrics.

Note that the time increase is constant. As we double the amount of signatures to be regenerated, the time increases by a factor of 1.6. This factor stays constant in the experiment performed.

5 Parallel Signing of the `.ca` Zone

When a zone file is cut into pieces and each part is separately signed, the speedup can be significantly. The correlation in zone size and sign time is linear: split the zone in two parts and each part is signed in half the time. In our test from the previous section, we divided `.ca` into five parts. Each part can be signed in roughly six minutes. If we take the splitting up and regeneration of `.ca` into account the total signing time would measure about 9.5 minutes: (elapsed signing time per piece + splitting + regeneration $= 6 + 1 + 2.5$).

6 Conclusion

Full signing of the `.ca` zone takes about half an hour on the hardware at our disposal. This is of course related to the keysize; the longer the key, the larger the resulting zonefile and the longer the sign time. Table 1 shows how different key lengths effect this.

The signing time is also related to the amount of signatures being generated as shown by the incremental signing table (Table 2). The various techniques used in this report can bring down the sign time some more. By utilizing parallel signing the sign time can come down to about 9.5 minutes.

7 Acknowledgments

Olaf Kolkman was invaluable for proofreading this report and providing corrections.

References

[1] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *DNS Security Introduction and Requirements*. RFC 4033 (Proposed Standard), March 2005. http://www.ietf.org/rfc/rfc4033.txt.

[2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *Protocol Modifications for the DNS Security Extensions*. RFC 4035 (Proposed Standard), March 2005. http://www.ietf.org/rfc/rfc4035.txt.

[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *Resource Records for the DNS Security Extensions*. RFC 4034 (Proposed Standard), March 2005. http://www.ietf.org/rfc/rfc4034.txt.

[4] O. Kolkman and R. Gieben. *DNSSEC Operational Practices <draft-ietf-dnsop-dnssec-operational-practices-06.txt>*, September 2005. ftp://ftp.ietf.org/internet-drafts/, (DNSOP WG Internet draft, drafts are subject to change and have a limited lifetime.).

[5] Olaf Kolkman. *Measuring the resource requirements of DNSSEC*. RIPE NCC web pages. http://www.ripe.net/ripe/docs/ripe-352.html.