Measuring the resource requirements of **DNSSEC**

Olaf M. Kolkman *

RIPE NCC / NLnet Labs
October 5, 2005

RIPE-352

Executive Summary

We measured the effects of deploying DNSSEC on CPU, memory and bandwidth consumption of authoritative name servers. We did this by replaying query traces captured from `ns-pri.ripe.net` and `k.root-servers.net` in a controlled lab environment.

We concluded that deploying DNSSEC on `k.root-servers.net` can easily be done with the currently deployed systems. In fact using the implementation today, the total increase in memory footprint is less than 5%. CPU usage would grow from 4 to 5% and the increase in bandwidth usage is around 10%.

Deployment of DNSSEC with all zones on `ns-pri.ripe.net` would cause a significantly higher consumption of memory and bandwidth usage while CPU would increase slightly. But growth is also well within the boundaries set by specifications of the currently deployed systems. The difference in bandwidth consumption between the `k.root-server.net` and `ns-pri.ripe.net` experiment was mainly due to the difference in the fraction of packets that requested DNSSEC information.

For `k.root-servers.net`, we also examined what the upper level of bandwidth consumption would be (*i.e.* what would happen if each request caused DNSSEC processing). The amount of bandwidth needed to answer the queries increased by 2 or 3 times, depending on some implementation properties. This growth is also within boundaries of currently deployed systems.

We recommend an implementation choice to minimise bandwidth consumption.

Increase of DNSSEC key sizes does not increase the amount of answers with the truncation (TC) bit set.

* (olaf@nlnetlabs.nl)

1 Introduction

DNSSEC [RFC4033, RFC4034, RFC4035] provides authentication and integrity checking to the domain name system. DNS Resource Records (RR) 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.

In addition to the DNSKEY and RRSIG RRs, DNSSEC introduces the NSEC RR that is used to prove the non-existence of data, and the DS RR that is used to delegate signing authority from one zone to another.

The introduction of these records causes zones to grow significantly. It is expected that this growth would have an effect on disks, memory and bandwidth usage. Besides DNSSEC aware servers have to do special processing to include the appropriate DNSSEC data. This might increase CPU load.

In this paper, we have set out to address the following question: *What would be the immediate and initial effect on memory, CPU and bandwidth resources if we were to deploy DNSSEC on k.root-servers.net and ns-pri.ripe.net?*

We performed a number of experiments in a test lab. Focusing on the immediate effects we used packet traces captured from the `ns-pri.ripe.net` and `k.root-servers.net` production servers. These traces were replayed — mimicking real-time behaviour— in the lab. The name server answering the queries carried zones that were signed with various sized keys. We measured memory consumption, packet counts and performed bandwidth measurements. These metrics in addition to some observations about the data contained in the traces provided enough information for provisioning the server infrastructure for DNSSEC.

In section 2 we describe the lab environment. In section 3 we perform an analysis of the query traces that were used in the experiment. The EDNS0 properties[RFC2671] of these traces is an important factor in the final bandwidth usage.

In section 4 we describe the impact of loading signed zones on the memory usage of the name servers and provide the parameters that can be used to estimate memory increase when loading signed zones.

In section 5 we assess the impact on CPU load when serving secured zones. Section 6 focuses on the bandwidth and the packet statistics for the various measurement configurations.

In section 7 we discuss the results and provide recommendations.

Appendix B provides a somewhat academic taxonomy of the replies for some of the experiments. We look at size distribution and discuss some of the phenomena that can be observed in these size distributions. This section contains the count of replies with the TC bit.

The paper by [ADF05] contains a similar analysis. Their experiments were more extended. They involved a caching forwarding name server. Their analysis used traces for which all the queries had been modified to include the EDNS0 OPT RR with the DO bit set. They measured the 'per packet' overhead. In this paper we will divert into a similar measurement — not through the

modifications of the query but through the modification of the name servers. We also measure bandwidth increase instead of per packet increase.

We assume the reader is familiar with DNS and DNSSEC jargon.

The figures and graphs are clearest when viewed in colour.

2 The Test Setup

2.1 The DISTEL Lab

The tests are performed on the Domain Name Server testing lab "DISTEL" test lab [KYLK02]. This lab was designed and implemented by the RIPE NCC to perform regression tests during the development of NSD. It is currently maintained by NLnet Labs.

The lab (Figure 1) consists of three identical machines containing AMD Athlon(TM) XP 2000+ (1670.46-MHz 686-class) CPUs. Two machines, labelled `player` and `recorder` contain 256 MB of memory. One machine, labelled `server`, contains 1.5GB of memory and acts as DNS name server. All machines run FreeBSD 6.0-CURRENT (13 June 2005).

The machines are connected through a 100baseT full duplex Ethernet network. The interfaces on that network are configured with RFC1918 addresses. The experiment is controlled through the player.

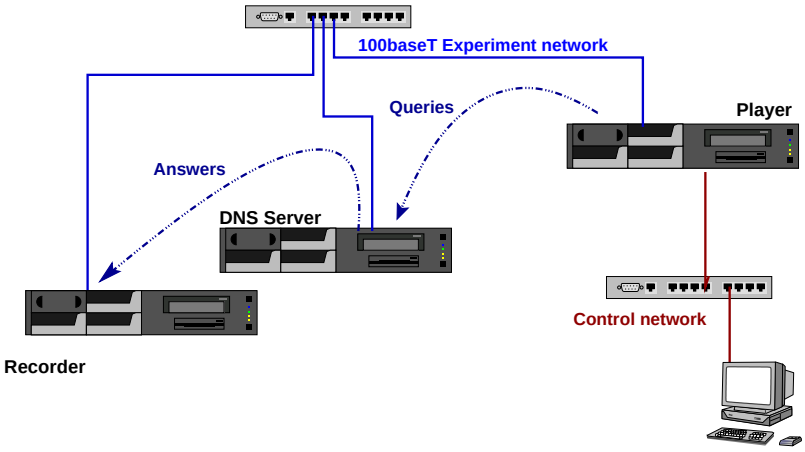

Figure 1: The DISTEL Test Lab

The `DNS server` has a host route configured for the `player` and has its default route configured to `recorder`. `recorder` has host routes for the other two machines and a default route to `/dev/null`.

Experiments are performed by taking `libpcap` traces from the production machines. After modifying the destination IP and MAC addresses to be those of the `server` the traces were replayed using a modified version of `tcpreplay`. The modifications to `tcpreplay` were done to avoid packet burst caused by problems with `usleep` having limited resolution and sometimes skipping a beat. See [KYLK02] for details.

Because of the default route configured on `server` the answers end up at `recorder` on which a `tcpdump` is collecting the answers before they end up at `/dev/null`.

In this setup, we can only study the effects of UDP traffic. The server used in the experiment has slightly lower hardware specifications, and runs a different operating system from the `ns-pri.ripe.net` and `k.root-servers.net` production systems.

2.2 The Server Software

For these experiments, the `DNS sever` ran BIND 9.3.1[BIND] and NSD 2.3.0 [NSD]. BIND was compiled with open-ssl and without multi threading[1], NSD 2.3.0 was compiled with the default options.

As well as the stock version of these name servers we used modified versions of both of the servers for a subset of the experiments. The modifications were designed to make the servers behave as if every query that had an EDNS0 option without the DO bit set actually had the DO bit set. It also acted as if every query without EDNS0 options actually contained an EDNS0 packet advertising 2048 UDP defragmentation capabilities of the query and with the DO bit set.

`named` was configured without recursion or any logging[2].

2.3 The Zone Configuration

We performed the experiment on two configurations. One configuration `ns-pri.ripe.net` and one that was to mimic `k.root-server.net`.

For the `ns-pri.ripe.net` type experiments, we used the configuration from 15 June 2005. At this date `ns-pri.ripe.net` was authoritative for zones like 193.in-addr.arpa (/8 reverse zones), 000.193.in-addr.arpa (/16 reverse zones) and their IPv6 variants whose content is dominated by delegation NS RRs ($3.94 * 10^5$ in total). Besides these 'delegation-only domains' there are a number of /24 reverse zones and a couple of small forward zones that relate to the RIPE NCC infrastructure. These account for about $0.05 * 10^5$ RRs roughly equally spread between PTR, CNAME, A and other resource records.

For the `k.root-servers.net` type experiments, we created a configuration where the server was authoritative for the root zone with SOA serial 2005070400. The server was not authoritative for other zones like the 'arpa' and 'root-servers.net' that are normally hosted by `k.root-servers.net`.

[1] BIND 9.3.1 that came with FreeBSD 6.0 CURRENT was compiled with multi-threading and that showed a clear performance limit at 3000 packets per second answer rate.

[2] During the preparations of experiments we noticed performance problems that were caused by having the `default` logging category logged to a channel that was configured with severity `debug 6`.

2.4 Zone Signing

We created a 2048 bits RSASHA1 key signing key (KSK) and two zone signing keys (ZSK) varying from 512 to 2048 bits. The ZSKs and KSK were included in the zone before it was signed with one ZSK and one KSK, using `dnssec-signzone` from the BIND 9.3.1 distribution. In the signed zone all RR sets are signed with one ZSK. Only the DNSKEY RR set is signed with two keys – the KSK and one ZSK.

Having two ZSKs in the DNSKEY RR set is expected to be a common situation for pre-publish zone signing key rollovers as in section 4.2.1 from [KG05]. In the pre-publish ZSK rollover model, the DNSKEY RRset grows during a ZSK rollover while the number of signatures over the RR sets in the zone remains constant.

3 Properties of the Query Traces

To perform the experiment, we obtained traces from the production servers. The traces we used contained UDP packets directed at port 53 of the server machines. The amount of data was roughly equivalent to about 10 minutes elapsed time and was taken at a random time for which the query stream did not seem to show anomalous behaviour.

We used two traces. One trace from `ns-pri.ripe.net` and one from `k.root-servers.net`. Table 1 summarises the properties of the traces.

The first column of the table lists the identifier used for the trace during the experiment. The second column lists the time we started recording the trace. The 3rd column the amount of UDP packets to port 53 contained in the trace. The 4th column lists the amount of DNS packets that the analysis script interprets as DNS packets. The packets that have their opcode set to "QUERY" and have not been truncated and do not have their query response flag set are probably bona-fide queries. We analysed those remaining packets on their EDNS0 content. The results are in Table 2 and the pie charts in Figure 2.

The left pie charts show the distribution between queries without the EDNS0 extensions, with the EDNS0 extension but without the DO-bit set and with the EDNS0 extension and with the DO bit set. The middle pie charts show the UDP defragmentation sizes advertised by the clients in the set of packets with an EDNS0 extension. The pie charts at the right show the sizes for those EDNS0 packets with the DO bit set.

The EDNS0 size distributions clearly differ for the query streams against the `k.root.servers.net` (top) and `ns-pri.ripe.net` (bottom). While the contribution of the "4096" size to the total of EDNS0 packets is roughly one third for all traces there is a distinct difference between the ratio of 1280 versus 2048 EDNS0 sizes. EDNS0 size 1280 makes up for almost 15% of the EDNS0 packets for the k.root trace while for the ns-pri traces, the contribution is less than 1 %. The EDNS0 size distribution is not the same for the queries with and without the DO bit set. EDNS0 size 1280 is not present for these queries.

The properties of the query streams clearly demonstrate selection effects.

We have not investigated what causes the difference between the properties of the queries to `k.root-servers.net` and `ns-pri.ripe.net`. If we were to undertake such investigation, our working hypothesis would be that most of the queries to `ns-pri.ripe.net` are targeted to sub-domains of `in-addr.arpa` and that those queries originate from environments where there is a preference for certain name server implementations.

Trace ID	Trace times	DNS packets	OPCODE QUERY	TC set	QR set	Remaining DNS packets
k.root	04/05/2005 5 23:57:54.338923 00:08:20.165730	2241766	2239638	6	37	2239598
ns-pri	06/15/2005 11:39:21.592356 11:49:09.123635	1711346	1705551	0	0	1705551

Table 1: Trace Properties

ID		Number	Fraction of total	EDNS0 size distribution 512	1024	1280	1500	2048	4000	4096	16384
k.root	E:	742275	34.5%	215	2	109000	0	330591	372	302088	7
	D:	227283	10.1%	141	0	0	0	89114	372	137655	0
ns-pri	E:	1202885	70.5%	1259	4	8353	9	745270	0	447990	0
	D:	476504	27.8%	538	0	0	9	234074	0	242062	0

The rows marked 'E:' show the statistics for the DNS packets with an EDNS0 OPT RR. The rows marked 'D:' show the statistics for the DNS packets that have the DO bit set.

Table 2: EDNS0 Properties of the Traces

We were also interested in the scenario for which all queries have the DO bit set. We simulated this by modifying the NSD code, to cheat the server into believing that queries with EDNS0 but without the DO bit had their DO bit set, and that queries without the EDNS0 extension had an EDNS0 section with the DO bit set and an EDNS0 size of 2048. ([ADF05] chose to use the minimum value of 1220 octets. We use 2048 because that is the minimum value that we've seen in the captured queries that have the DO bit set.). We only performed measurements with this modified name server using the `k.root-servers.net` configuration. These experiments are marked "k.modified" and referred to as experiments on the "modified server".

4 Memory Load

To assess the effect of introducing DNSSEC on the memory load of the servers, we created 100 configuration files that contained mixtures of different fractions of signed as opposed to unsigned zones. This allowed us to measure memory usage from the `named` process for different numbers of NSEC and RRSIG records.

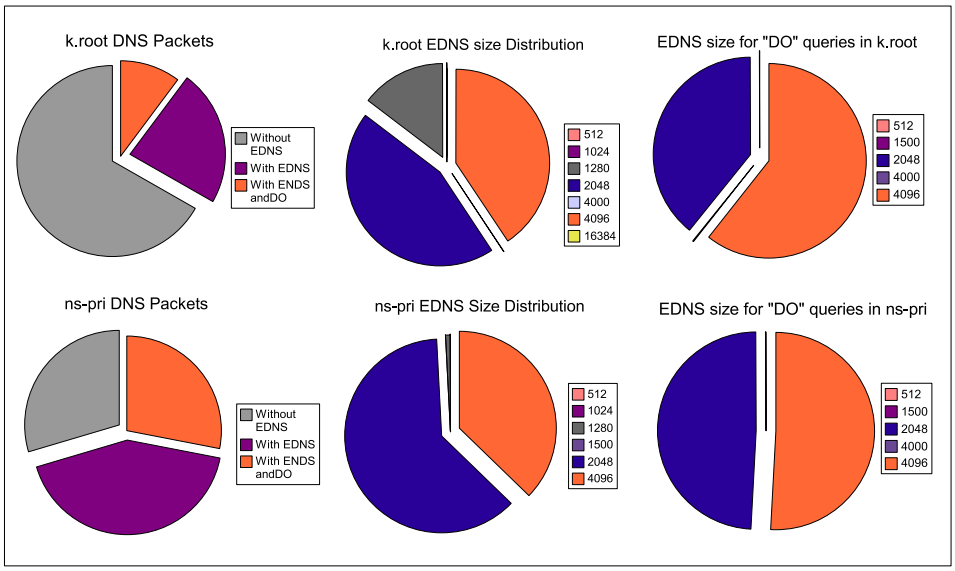

Figure 2: EDNS0 Properties for the k.root (top) and ns-pri Traces (bottom).

We loaded these configurations and measured the virtual memory size (VSZ) as reported by `ps`. We repeated the experiment for different sizes of zone signing keys and for two different operating systems.

- FreeBSD 6.0 on the server system described in Section 2.1.
- Linux 2.6.10-5-686-smp (Ubuntu distribution) on a Intel P4/Xeon 3Ghz machine with 1GB of memory.

The results are shown in Figure 3 in which the VSZ is plotted against the number of signatures in the zones loaded. The increase in zone size is a function of the number of RRSIGs and the number of NSEC RRs introduced during the signing. Because the content of our zone files (mostly delegation records) the ratio of NSEC to RRSIG records is reasonably constant when it reaches a large number of RRSIGs.

It is clear from the graphs that memory consumption is depended on OS and implementation.

For the FreeBSD system there is no difference in BIND's memory consumption for ZSKs between 512 and 1280 bits, and for ZSKs between 1596 and 2058 bits. The step function in memory load for BIND on FreeBSD is most probably caused by different alignment properties of the various implementations of `malloc`.

Because we did not have both operating systems available in our lab, we did not investigate any performance differences when running the servers on different operating systems.

Assuming linear increase of the memory usage as function of the number of NSECs and RRSIGs loaded by `named` we have performed a least square fit to the following function: $VSZ(NSEC, RRIG) = A * RRSIG + B * NSEC + C$ The

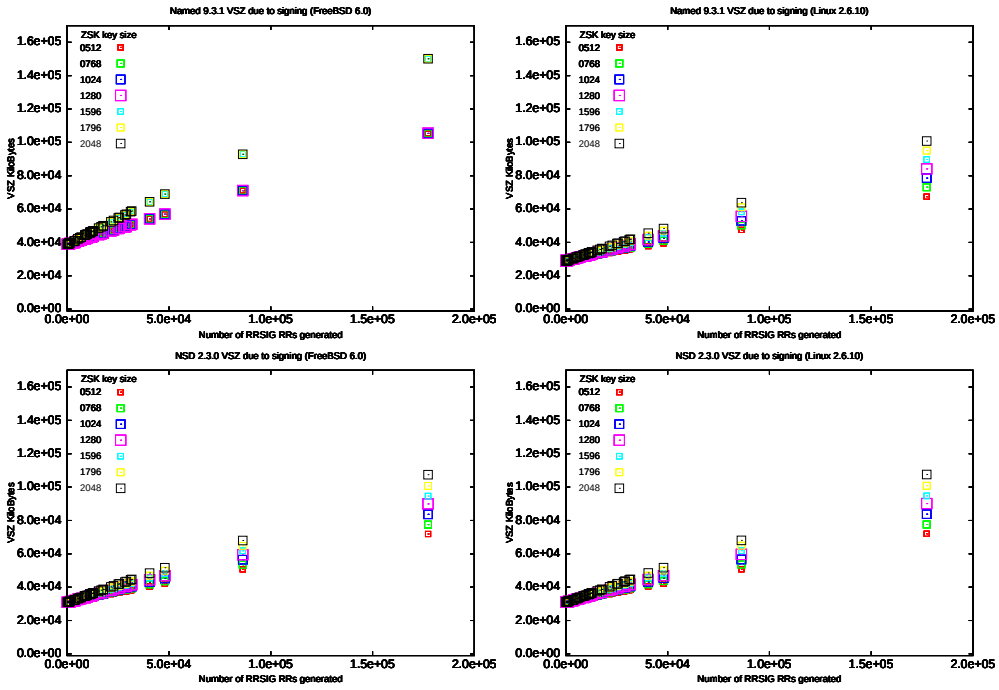

Figure 3: VMS for Different Key Sizes
Left column on FreeBSD. Right column on Linux 2.6.10, top row measurements for `named` *9.3.1 and bottom row measurements for* `nsd`*.*

results for BIND are tabulated in table 3 The units for the parameters are kilo bytes (1024bit).

The values from these tables can be used for estimates of the zone size increase[3].

The memory increase after signing the root zone was measured to be about 156 KB for `named` 9.3.1 on FreeBSD 6.0. This is exactly what would be expected from using the table above with 262 NSEC RRs (one for the apex and one for each delegation) and 267 RRSIGs (over the apex RRs and over each NSEC).

5 CPU load

CPU load was measured by reading the weighted CPU from `top` after the value had flattened out, a couple of minutes after the start and the initial increase in the load due to the loading of zones of the experiment. The variation of the CPU throughout the experiment was typically 2-3% for BIND and about 1% for NSD.

We did not see significant between the CPU load between serving signed and unsigned zones.

[3]The fits shows very different results for the amount of memory consumed per NSEC (the B parameter in Table 3). We do not expect that parameter to change between the various fits. The cause is that the number of NSEC RRs and the number of RRSIGs is not completely independent. The number of NSEC RRs is never larger than the number of RRSIG RRs, and the ratio for the zones for which ns-pri.ripe.net is authoritative is between 1:1 and 1:2. As a result we are trying to fit a plane to an almost straight line in the $(VSZ, NSEC, RRSIG)$ space and there are to little data points away from the line to reliably determine $NSEC$.

named 9.3.1 on Linux 2.6.10									
ZSK	A			B			C		
size	value	+/-	%	value	+/-	%	value	+/-	%
512	0.114	0.014	12.2	0.105	0.014	13.6	2.905e+04	7	0.024
768	0.161	0.016	10.0	0.091	0.016	18.1	2.898e+04	8	0.028
1024	0.211	0.013	6.4	0.070	0.014	19.8	2.904e+04	7	0.023
1280	0.230	0.016	6.8	0.082	0.016	19.5	2.903e+04	8	0.027
1536	0.292	0.016	5.3	0.052	0.016	30.8	2.901e+04	8	0.027
1792	0.280	0.014	5.0	0.096	0.014	15.0	2.905e+04	7	0.024
2048	0.326	0.014	4.3	0.081	0.014	17.5	2.904e+04	7	0.024
named 9.3.1 on Freebsd 6.0									
ZSK	A			B			C		
size	value	+/-	%	value	+/-	%	value	+/-	%
512	0.236	0.004	1.6	0.143	0.004	2.7	3.888e+04	2	0.005
768	0.237	0.004	1.6	0.142	0.004	2.8	3.887e+04	2	0.005
1024	0.242	0.005	2.0	0.137	0.005	3.6	3.887e+04	2	0.006
1280	0.244	0.004	1.5	0.135	0.004	2.9	3.887e+04	2	0.005
1536	0.494	0.004	0.8	0.137	0.004	3.0	3.888e+04	2	0.005
1792	0.494	0.004	0.8	0.137	0.004	3.1	3.888e+04	2	0.005
2048	0.494	0.004	0.8	0.137	0.004	3.0	3.888e+04	2	0.005
nsd 2.3.0 on Linux 2.6.10									
ZSK	A			B			C		
size	value	+/-	%	value	+/-	%	value	+/-	%
512	0.177	0.009	4.9	0.055	0.009	16.2	3.100e+04	5	0.015
768	0.214	0.008	4.0	0.050	0.009	17.5	3.099e+04	4	0.014
1024	0.219	0.006	2.9	0.081	0.007	8.2	3.100e+04	3	0.011
1280	0.254	0.007	2.8	0.082	0.007	9.0	3.098e+04	4	0.012
1536	0.292	0.007	2.4	0.069	0.007	10.4	3.099e+04	4	0.012
1792	0.318	0.006	2.0	0.079	0.007	8.4	3.097e+04	3	0.011
2048	0.350	0.007	2.1	0.085	0.008	9.1	3.097e+04	4	0.013
nsd 2.3.0 on Freebsd 6.0									
ZSK	A			B			C		
size	value	+/-	%	value	+/-	%	value	+/-	%
512	0.167	0.003	1.6	0.066	0.003	4.1	3.091e+04	1	0.004
768	0.199	0.003	1.4	0.065	0.003	4.5	3.091e+04	1	0.005
1024	0.231	0.003	1.1	0.069	0.003	3.9	3.091e+04	1	0.004
1280	0.254	0.003	1.2	0.082	0.003	3.9	3.091e+04	2	0.005
1536	0.295	0.003	1.1	0.066	0.003	4.8	3.091e+04	2	0.005
1792	0.321	0.003	1.0	0.075	0.003	4.3	3.091e+04	2	0.005
2048	0.348	0.003	1.0	0.086	0.004	4.2	3.091e+04	2	0.006

Table 3: Memory Usage Parameters

trace	server	ZSK size	WCPU	
ns-pri	`named` 9.3.1	0000	ca 14%	
ns-pri	`named` 9.3.1	2048	ca 18%	
k.root	`named` 9.3.1	0000	ca 38%	
k.root	`named` 9.3.1	2048	ca 42%	
k.root	`named` 9.3.1	2048	ca 50%	(modified server)
k.root	`nsd` 2.3.0	0000	ca 4%	
k.root	`nsd` 2.3.0	2048	ca 4%	
k.root	`nsd` 2.3.0	2048	ca 5%	(modified server)

Table 4: CPU Usage on FreeBSD 6.0

CPU load does not seem to be a function of the zone signing key size. For `named`, the CPU load does seem to correlate with the amount of packets for which DNSSEC processing needs to be done (as simulated by the modified server). For `nsd` the differences between the CPU load for the server with and without modifications to perform DNSSEC processing for all queries was too small to lead to any conclusions.

6 Bandwidth and Packet Counts

To collect bandwidth statistics, we ran the `iostat` command on the server machine while the query stream was replayed. As a side result we obtained egress packet counts.

The `iostat` program was started shortly before the query stream was replayed therefore there might be small offsets on the time-axis in the plots presented in this section.

6.1 ns-pri traces

Figure 4 indicates that the egress bandwidth for measurements with the ns-pri traces against `named` doubles for small and triples for larger zone signing keys. If we look at the packet counts in Figure 5, we see that that the amount of Ethernet packets sent out for 512 ZSK signed zones is the same as for unsigned zones. The amount of packets needed for 768 to 2048 bit ZSK signed zones is about 10% more. This is an indication for IP fragmentation. And confirms the what we saw in section B.1.

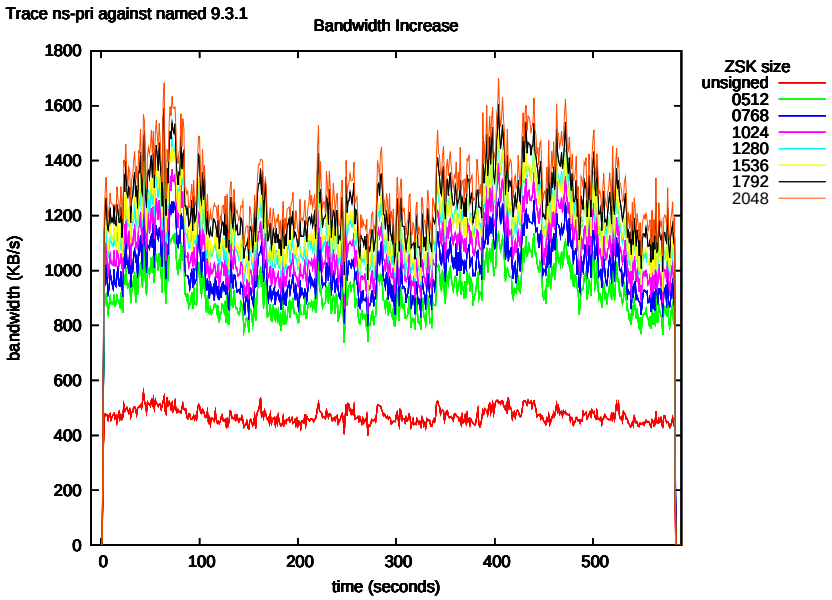

Figure 4: Bandwidth as Function of Key Size for Trace ns-pri

6.2 K.root Traces

Bandwidth increase due to zone signing for the root is shown in Figures 6 and 7.

The egress packet counts are plotted in Figures 8, and 9. What can be observed from these plots is that the packet counts on the unmodified versions do not show significant offsets. All lines overlap.

The response packets from the modified `named`, in the left graph of Figure 8, grow above the Ethernet MTU. This shows through the offsets in the packet counts.

All response packets from the modified `nsd` have sizes lower than the Ethernet MTU therefore all lines in Figure 9 overlap.

7 Discussion

Because the number of packets with the "DO" bit set is relatively low, there would be no significant impact on the bandwidth consumption for `k.root-servers.net` if a signed root zone were to be served today. Even in a worst-case scenario, when all queries to the root would have the "DO" bit set, the bandwidth usage would only grow by a factor 2 to 3 depending on key size and implementation used.

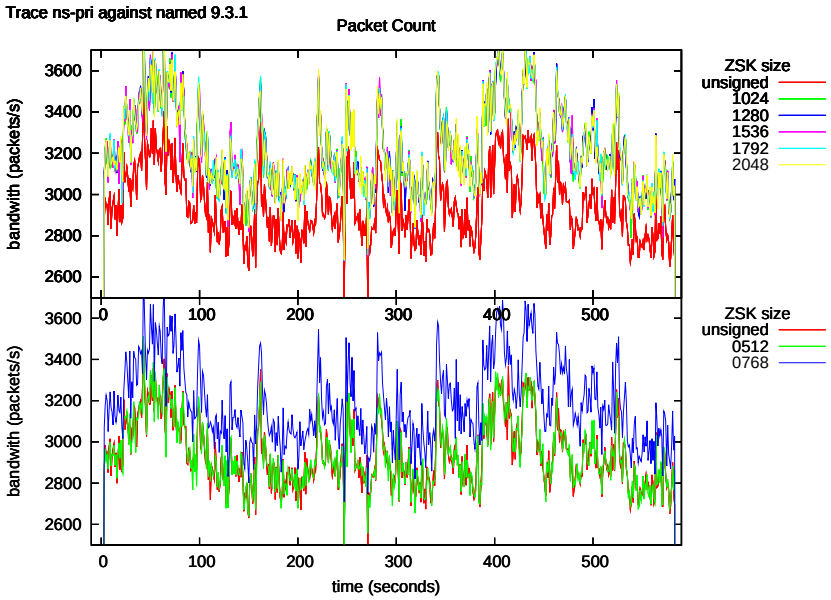

Figure 5: Packet Counts as Function of Key Size For Trace ns-pri

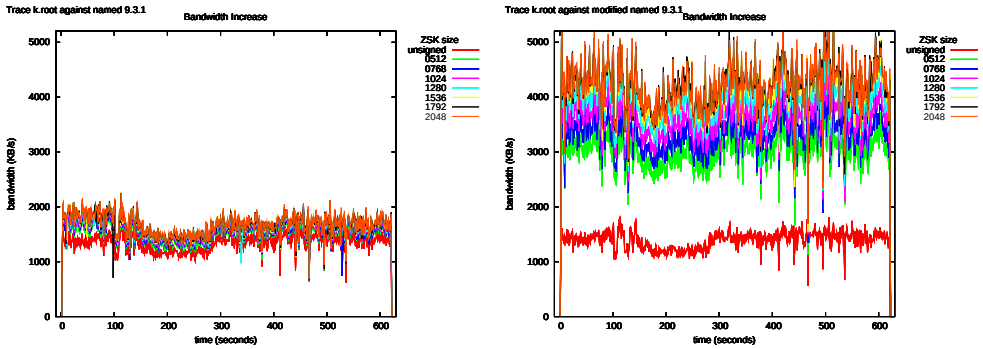

Figure 6: Bandwidth Increase as Function of Key Size for the k.root Trace Against `named` 9.3.1 and Modified `named` 9.3.1

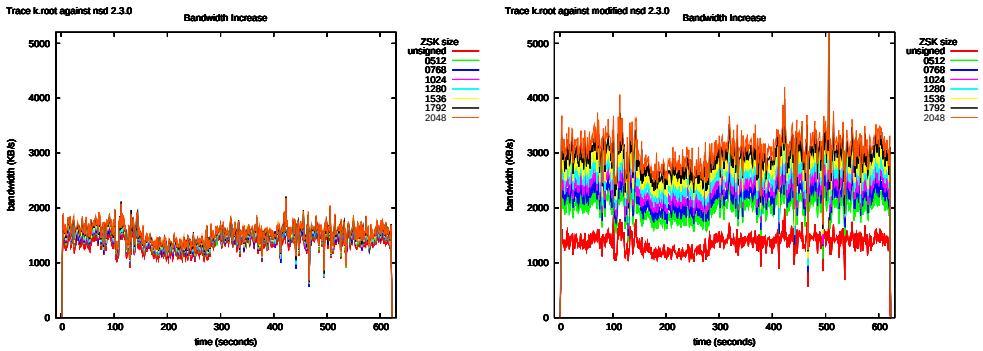

Figure 7: Bandwidth Increase as Function of Key Size for the k.root Trace Against `nsd` 2.3.0 and Modified `nsd` 2.3.0

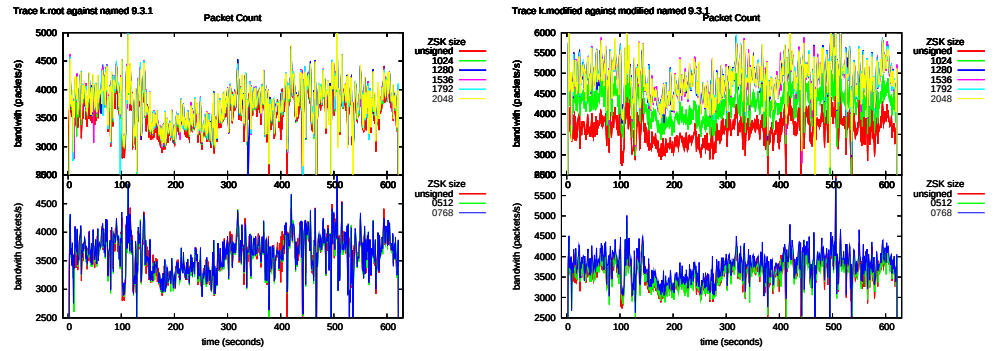

Figure 8: Packet counts as function of key size for trace k.root against **`named`** *9.3.1 and modified* **`named`** *9.3.1*

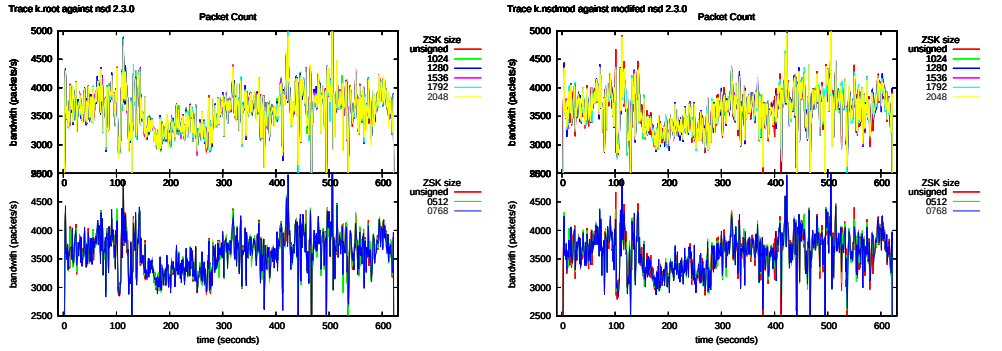

Figure 9: Packet counts as function of key size for trace k.root against **`nsd`** *2.3.0 and modified* **`nsd`** *2.3.0*

For `ns-pri.ripe.net` signing all its zones would increase bandwidth use by a factor 2 to 3 depending on the key size. The increase in CPU and memory usage will not cause problems on the production system.

It is safe to conclude that the impact of DNSSEC deployment on CPU load is of no concern. The impact of memory consumption can easily be estimated from the zone content. The growth in memory consumption is significant, but we do not think that the total amount of memory needed will be an issue for most authoritative servers running on commodity hardware; when serving of the order of 10^6 resource records the four gigabyte limit on 32bit Intel architectures may become a barrier.

The impact on bandwidth usage is much harder to estimate.

7.1 Estimating Bandwidth Usage

The effects of enabling DNSSEC is depends on many variables. These are, roughly in order of significance:

- the fractions of queries with the “DO” bit set;
- the inclusion of a DNSKEY RRs in the additional section; (See the recommendation below.)
- the amount of queries for existent and non-existent domain names; (The size difference between an unsigned Name Error packet and a signed Name Error

packet is significant, also see [ADF05].)

- the size of the ZSK and KSK. The effect of the size of the KSK is not measured here, but it makes up a significant fraction of the DNSKEY RR set and its signatures;
- the amount of KSKs and ZKSs in the DNSKEY RRset; (Depending on the rollover scheme used there can be multiple DNSKEY RRs in the zone and multiple RRSIGs over these keys. If multiple algorithms are used, which could be the case when a transition from RSASHA1 to ECC is made, the size of these RR sets can grow quite significantly.)
- the label depth of the records in the zone and therefore the need to include a second NSEC RR, with corresponding RRSIG, to prove the non existence of wildcards;

Because of these variables, there is no easy method to determine the impact of signing zones in generic environments.

The two configurations we tested were biased towards "delegation only" zones. For a delegation, two RRs are added to positive responses when serving a signed zone, an NSEC or DS RR and a RRSIG. When serving "end-node" data only a RRSIG is added for positive responses.

When estimating the impact of DNSSEC deployment, you should first look at the fractions of queries that have the DO bit set. If that fraction is small the impact of DNSSEC deployment will not be significant.

7.2 Effects Not Measured

These experiments only provide insight to the first order effects. We do not know what happens on the Internet when DNSSEC answers are returned to the clients.

The following scenarios seem likely:

Scenario A – A client behind a middle box (e.g. a firewall) sets the DO bit. The query crosses the middle box and is answered by the server. The answer that contains DNSSEC information does not cross the middle box, simply because the middle box does not implement DNSSEC or is not configured to deal with fragmented UDP on port 53.

scenario B – The Client sets the DO bit, but does not know what to do with the DNSSEC information in the answer and continues to query.

Scenario A is most likely. The EDNS0 specifications [RFC2671] do not demand that a resolver should retry without EDNS0 when there has not been an answer from an EDNS0 query. However, BIND developers have explained that BIND9 will re-query without EDNS0 and will cache the EDNS0 capability of the server. In other words, BIND9 resolvers will fall back to non-EDNS0 behaviour, resend their query without the DO-bit and receive non-DNSSEC answers[4]

We are not aware of any clients that set the DO bit and would not know how to process the DNSSEC replies.

Appendix A provides an overview of which type of clients queried k.root-servers.net. Most of the DO queries are being sent by BIND servers. Only an analysis of query patterns of deployment on signed zones on the Internet can tell us if we should worry about non-linear effects.

[4] As soon as these client configure a trust-anchors they will see validation failures and will not be able to resolve signed zones at all.

7.3 Recommendation

The distinct difference between `named` and `nsd` is that the latter does not add DNSKEY RRs and related RRSIGs in the additional section for a Name Error response. Both servers comply to the specification since *a security-aware authoritative name server [...] MAY return the zone apex DNSKEY RRset in the Additional section* (section 3.2.1 [RFC4035]). This different behaviour is the main cause for the difference in bandwidth consumption between the two implementations.

It can be argued that at the start of DNSSEC deployment, while the amount of validation DNS clients is small, there is little need to add the DNSKEY RR set to the additional section. Most DNS clients that set the DO bit will not use the information anyway and the minority of clients that do need the key information can afford the extra query. Our recommendation is to make the inclusion of DNSKEY RRs in the additional section, a configurable option, at least during early deployment of DNSSEC.

Stripping the RRSIG RR set from the additional section without stripping the DNSKEY RRs increases bandwidth consumption, while validators that need the DNSKEY RRs for validation will still need to re-query to obtain the missing signatures.

8 Conclusion

Serving signed zones has little impact on CPU resource use. The impact on memory usage is predictable and well within the specification of the production machines that are authoritative for the zones we used in this experiment.

Bandwidth usage increase can be significant, an increase by a factor of 2 to 3 was measured. To reduce the impact of bandwidth consumption, we recommend configuring your name server to not include DNSKEY RRs in the additional section.

Appendices

A DNS Clients

We took a sample of 3927 unique IP addresses from the traces and used a DNS fingerprinting tool [AS] to see what type of clients set the DO-flag. We could determine the client type for 2373 of the IP addresses.

This table only provides an indication of the resolvers that queried the root server. The fingerprinting was done at a different time to when we made the original query. This could explain the appearance of implementations that, as far as we know, do not have a DNSSEC support.

We performed a visual inspection on the queries sent by implementations other than BIND. These packets were all valid DNS packets with EDNS0 OPT RRs in the additional section that were 'properly' formatted *i.e.* they had no other flags than the DO-flag set, advertised 2048 or 4096 sizes, etc.

As mentioned in 7.2 the behaviour of BIND9 in situations where middle boxes drop replies is well understood.

B Analysis of the Recorded Replies

The replies recorded by the `recorder` contain much information. Since we tried to limit the scope of this paper and because of the lack of dedicated tools, we did not correlate

Client implementation	no.
ATOS Stargate ADSL	1
BIND 8.3.0-RC1 – 8.4.4	2
BIND 8.3.0-RC1 – 8.4.4 [recursion enabled]	8
BIND 8.3.0-RC1 – 8.4.4 [recursion local]	4
BIND 9.0.0b5 – 9.1.3 [recursion local]	3
BIND 9.1.0 – 9.1.3	17
BIND 9.1.0 – 9.1.3 [recursion enabled]	71
BIND 9.2.0a1 – 9.2.0rc3 [recursion enabled]	4
BIND 9.2.0a1 – 9.2.2-P3 [recursion enabled]	29
BIND 9.2.0a1 – 9.2.2-P3 [recursion local]	1
BIND 9.2.0rc4 – 9.2.2-P3 [recursion enabled]	7
BIND 9.2.0rc7 – 9.2.2-P3	173
BIND 9.2.0rc7 – 9.2.2-P3 [recursion enabled]	836
BIND 9.2.0rc7 – 9.2.2-P3 [recursion local]	47
BIND 9.2.3rc1 – 9.4.0a0 [recursion enabled]	828
ISC BIND 9.2.3rc1 – 9.4.0a0	309
Microsoft Windows 2000	3
Microsoft Windows 2003	1
Microsoft Windows NT4	1
MyDNS	18
PowerDNS 2.9.4 – 2.9.11	4
Runtop Implementation	3
TinyDNS 1.05	2
totd	1
No match found	325
TIMEOUT	1229

Table 5: Implementations Setting the DO Bit

the queries that were sent with the answers received. Neither did we investigate the variations in response times for different key sizes.

In Table 6, we present the core parameters of the recorded reply traces. In the second column of each of the four sub-tables we list the amount of DNS packets we counted in the reply trace. The third column lists how big a fraction this is from the number of DNS packets that we've listed in the third column of Table 1.

The decreasing percentage of DNS packets received for growing key sizes is not caused by the servers dropping packets. The analysis script, that was written to measure *first order effects*, does its own IP packet defragmentation. It is likely that it fails to to do defragmentation in all cases where IP fragmentation has occurred.

The fourth column indicates the amount of packets with the truncated (TC) bit set. What can be seen from this table is that the introduction of DNSSEC will cause packet truncation for about one packet each second. Wider application of EDNS0 would reduce truncations for unsigned zones (e.g. the `k.modified` query has 0 truncated packets compared to 0.5k packets truncated for replies from the unmodified server.)

The fifth column presents a count of the number of packets that had at least one RRSIG resource record in any of the sections of the packet.

B.1 Size Distribution

We measured the sizes of the packets in the response traces which had OPCODE QUERY and did not have the TC bit set in the header.

ns-pri against named 9.3.1			
ZSK size	DNS packets	TC set	with RRSIGs
0000	1703986 (99.91%)	0	0
0512	1703993 (99.91%)	199	475803
0768	1703972 (99.91%)	199	475837
1024	1703837 (99.90%)	199	475786
1280	1704008 (99.91%)	205	475828
1536	1703957 (99.91%)	205	475829
1792	1704119 (99.92%)	205	475843
2048	1703858 (99.90%)	209	475771

k.root against named 9.3.1			
ZSK size	DNS packets	TC set	with RRSIGs
0000	2235229 (99.80%)	0	0
0512	2233248 (99.71%)	518	226756
0768	2228432 (99.50%)	548	226208
1024	2230825 (99.61%)	550	226496
1280	2228756 (99.51%)	583	226189
1536	2229255 (99.54%)	603	226192
1792	2227160 (99.44%)	603	225811
2048	2230531 (99.59%)	628	226218

k.root against modified named 9.3.1			
ZSK size	DNS packets	TC set	with RRSIGs
0000	2234747 (99.78%)	0	0
0512	2231873 (99.65%)	1	2227936
0768	2229299 (99.54%)	68	2225284
1024	2227767 (99.47%)	75	2223773
1280	2227267 (99.45%)	119	2223217
1536	2225612 (99.37%)	648	2221042
1792	2226255 (99.40%)	684	2221640
2048	2225794 (99.38%)	705	2221160

k.root against nsd 2.3.0			
ZSK size	DNS packets	TC set	with RRSIGs
0000	2239770 (100.01%)	518	0
0512	2236494 (99.86%)	520	226882
0768	2235973 (99.84%)	549	226730
1024	2237481 (99.90%)	550	226853
1280	2237250 (99.89%)	585	226796
1536	2237242 (99.89%)	606	226809
1792	2236459 (99.86%)	610	226782
2048	2237055 (99.88%)	634	226749

k.root against modified nsd 2.3.0			
ZSK size	DNS packets	TC set	with RRSIGs
0000	2240141 (100.02%)	0	0
0512	2238062 (99.93%)	0	2234120
0768	2238556 (99.95%)	0	2234612
1024	2238696 (99.96%)	0	2234752
1280	2237768 (99.92%)	0	2233825
1536	2236664 (99.87%)	519	2232201
1792	2235883 (99.83%)	519	2231424
2048	2234545 (99.77%)	517	2230096

Table 6: Core Characteristics of the Recorded Response Traces

B.1.1 Size Distributions for ns-pri Experiment

Figure 10 shows the size distribution of four different key sizes on a linear scale. In this plot, you can distinguish distinct peaks that we will refer to as *harmonics*. To study these more closely, we created a plot with a logarithmic scale in Figure 11. In this plot, we marked a number of the *harmonics*. The properties of these *harmonics*, that are specific for the ns-pri setup are described below. These *harmonics* are defined by observed peaks. The amount of responses (the *signal*) in these peaks is determined by zone content and query distributions and are therefore specific to the system under observation.

Below we describe the properties of some of the *harmonics* found in the ns-pri replies.

Harmonic 1 is caused by packets that contain delegations in reverse address space that contain two NS, one NSEC and one RRSIG resource record in the authority section and have an empty answer and additional section.

Harmonic 2 is caused by by packets that contain a delegation for a /16 in-addr.arpa zone for which ns-pri is an authoritative server. So, in addition to the two NS one

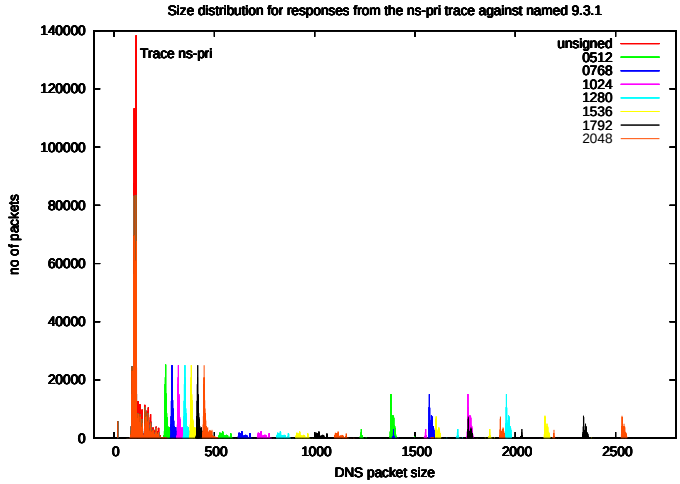

Figure 10: DNS Packet Size Distributions for ns-pri Traces Against `named` *9.3.1*

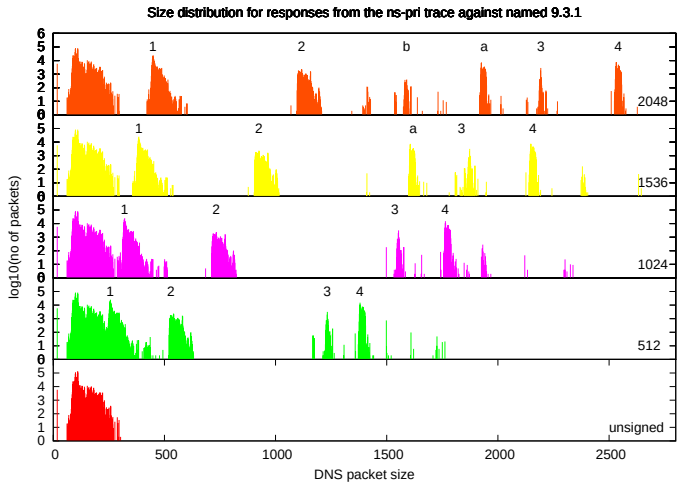

Figure 11: Logarithmic DNS Packet Size Distributions for ns-pri Traces Against `named` *9.3.1*

id	512	1024	1536	2048
1	289	353	385	449
2	530	733	925	1117
3	1233	1536	1837	2193
4	1380	1764	2158	2542
a			1619	1939
b				1590

Table 7: Harmonics Found in the Signed Responses for the ns-pri Trace. Also see Figure 11

NSEC and one RRSIG resource records in the authority section the additional section contains an A and AAAA RR for ns-pri.ripe.net plus the RRSIGs over those records.

Harmonic 3 is caused by Name Error responses (NXDOMAIN). These packets contain a SOA, an NSEC and their RRSIG Resource records in the authority section and the DNSKEY RR set (containing three keys) with its RRSIGs (one with the KSK and one with the ZSK).

Harmonic 4 is also caused by Name Error responses. These include the SOA and two NSEC RRs with their RRSIGs in the authority section, and the DNSKEY RRs with RRSIGs in the additional section. The inclusion of two NSEC RRs is needed to deny the existence of wildcards.

Harmonic A is a *harmonic 4* type packet with the RRSIGs over the DNSKEY RR set in the additional section removed.

Harmonic B is a *harmonic 3* type packet with the RRSIGs over the DNSKEY RR set in the additional section removed.

The difference between *Harmonic 3* and *4* is that *Harmonic* 3 contains a NSEC RR that immediately proves the non existence of a 'closer encloser' like in

```
;; QUESTION SECTION (1 record)
;; 215.103.43.84.in-addr.arpa.  IN      PTR
(...)
9.42.84.in-addr.arpa. 7200  IN      NSEC    128.43.84.in-addr.arpa (
                                            NS NSEC RRSIG )
```

while *Harmonic* 4 packets need two NSECs for such proof as in

```
;; QUESTION SECTION (1 record)
;; 47.112.102.80.in-addr.arpa.  IN      PTR
(...)
80.in-addr.arpa.      7200  IN      NSEC    0.80.in-addr.arpa (
                                            DNSKEY NS NSEC RRSIG SOA )
100.80.in-addr.arpa.  7200  IN      NSEC    104.80.in-addr.arpa (
                                            NS NSEC RRSIG )
```

See section 3.1.3.2 of [RFC4035] for details.

The *harmonics* do not always contain the same amount of packets. As soon as the size of packets of a given *harmonic* become larger than that advertised by a significant fraction of the clients, some of the *signal* from that *harmonic* is transferred to another one. For instance, at ZSK size 1536 *signal* from *Harmonic 4* is transferred to *Harmonic A* since the *harmonic* occurs at a size larger than 2048. On the other hand there is no sign of an equivalent to *harmonic b* since *harmonic 3* is still below the 2048 size limit.

Since the amount of signal is hard to judge from these logarithmic plots, we show the fraction of the total amount of packets that is smaller than a given size for the different ZSK sizes in Figure 12. In this kind of plot, the *harmonics* can be identified by the steep rises in the curves. You can clearly see the transfer of *signal* for ZSK sizes of 1536 bits and larger. The last step in the graph corresponds to *Harmonic 4*. For smaller keys this *harmonic* accounts

for slightly less than 10% while for larger keys some about half of these packets are moved to *Harmonic A*.

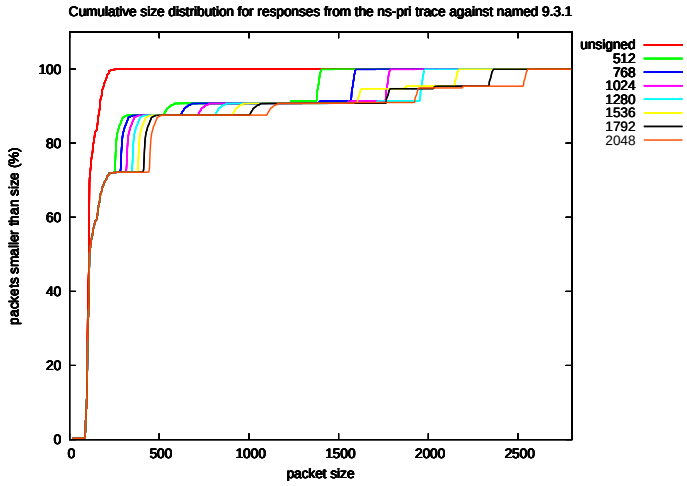

Figure 12: Cumulative DNS Packet Size Distributions for ns-pri Traces Against `named` *9.3.1*

B.1.2 Size Distributions for k.root Experiments

For each of the four experiments we performed using the k.root trace we created the various size distribution plots (Figure 13, 14 and 15). In these plots, the left column presents the measurements for the `named` 9.3.1 and the right column the measurements for `nsd` 2.3.1. The top row shows the effect of replaying the k.root trace as it stands, the bottom row shows the effects of replaying the trace against the modified versions of the server.

Since the fraction of packets in the k.root trace with the DO bit set is roughly 10%, the effects of signing are not as extreme as for the ns-pri trace.

To study the effect of signing on packet content, we use the output of the modified `named` server and define a number of *harmonics*. These *harmonics* are listed in Table 8 and indicated in the bottom left plot of Figure 14.

Below we describe the properties of some of the *harmonics* found in the replies from the modified `named` 9.3.1 server.

Harmonic 1 are delegations to the arpa domain containing 12 NS RRs, on NSEC and one RRSIG over the NSEC in the authority section and 12 A type glue RRs in the additional section.

This *harmonic* would not occur on the production system as `k.root-servers.net` is authoritative for the 'arpa' domain. It would return a delegation to the relevant subsequent domain or return a name error.

Harmonic 2 are delegations to the gtld-servers. The authority section contains 13 name servers, one NSEC and one RRSIG. The additional section

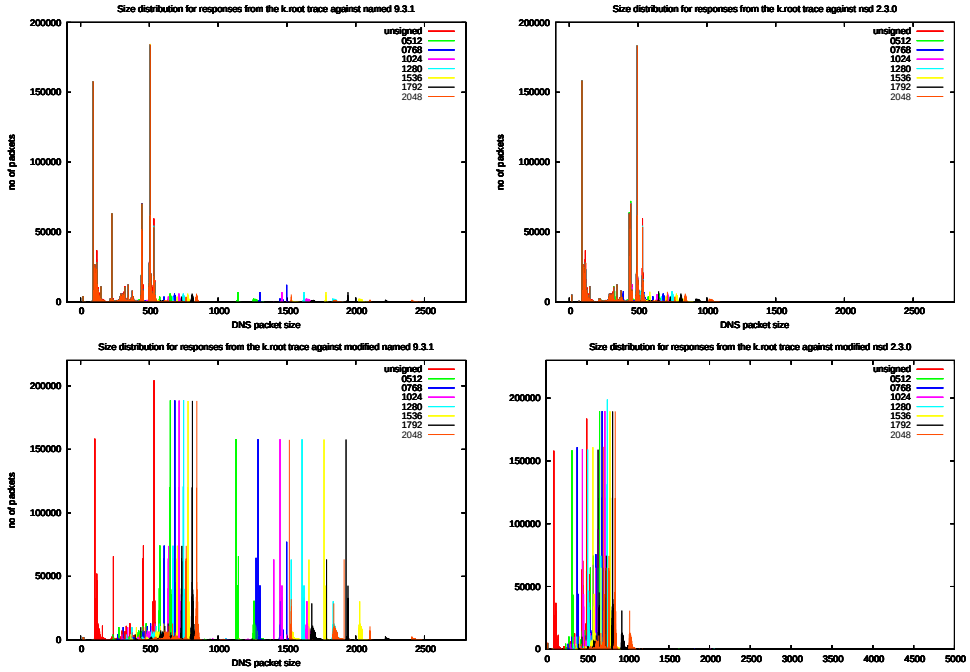

Figure 13: Size Distribution for Different ZSKs of Response Packets after Playing the k.root Trace Against **named** *9.3.1 (top left),* **nsd** *2.3.0 (top right), Modified* **named** *9.3.1 (bottom left) and Modified* **nsd** *2.3.0 (bottom right).*

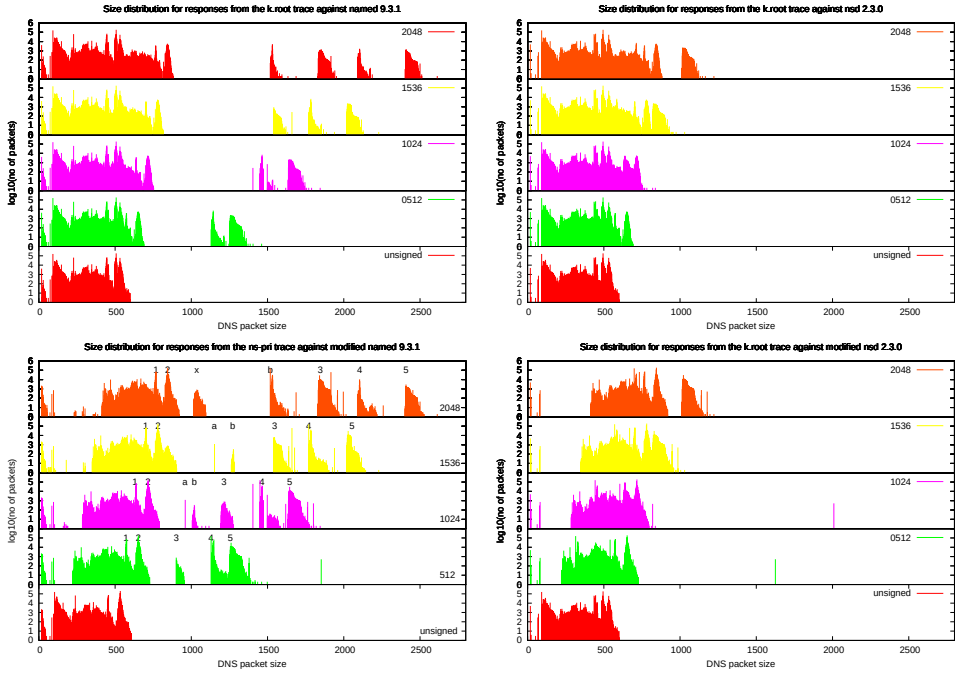

Figure 14: Logarithmic Size Distribution of Response Packets after Playing the k.root Trace Against **named** *9.3.1 (top left),* **nsd** *2.3.0 (top right), Modified* **named** *9.3.1 (bottom left) and Modified* **nsd** *2.3.0 (bottom right).*

id	512	1024	1536	2048
1	573	630	702	768
2	653	717	781	845
3	901	1215	1546	1847
4	1130	1464	1770	2104
5	1256	1645	2054	2408
a	—	960	1152	—
b	—	1020	1272	1518
x	—	—	—	1036

Table 8: Harmonics Found in the Signed Responses for the k.root Trace Against the Modified `named` *9.3.1 name server. Also see Figure 14*

contains 13 A type glue RRs and 2 AAAA type glue RRs.

Harmonic 3 are typically Name Error responses.

They contains the root's SOA, two NSEC RR and their RRSIGs in the authority section and the signed DNSKEY RR *without* a RRSIG RR set in the additional section.

The two NSEC RRs are needed to prove non-existence of the exact match (pointing from NR. to NU.) and one to prove non existence of a wildcard.

If the RRSIGs had been included, these packets would be included in *Harmonic 5*

This *harmonic* can not be distinguished in the responses from `nsd` (bottom right plot). `nsd` does not include the DNSKEY RR set and the data remains hidden between the majority of queries in the band around 500 octets size.

This *harmonic* can also not be distinguished in the responses from the unmodified name server for 512 bit ZSK (see the plot on the top left). They are significant in plot for the modified server. The *third harmonic* in the 512 ZSK measurements is mostly caused by queries for `_ldap._tcp.dc._msdcs.ntserver. IN SRV`.

This harmonic appears because on the modified server the original EDNS0 size is honoured and a significant fraction of the clients advertise that they can handle 1280 sized responses.

Harmonic 4 are typically responses to `. IN A` queries. The responses typically contain the SOA and an NSEC RR (owned by '.') and their RRSIGs in the authority section and the signed DNSKEY RR set in the additional section.

Harmonic 5 are typically Name Error responses that contain the root's SOA, two NSEC RR and their RRSIGs in the authority section and the signed DNSKEY RR with their RRSIG RRs in the additional section.

Harmonic A only occurs in the plots for ZSKs 1024 and 1596.

Packets in this *harmonic* are typically responses to the priming query `. IN NS`. With the roots NS RRset and its RRSIG in the answer section and a DNSKEY RR set *without* RRSIG in the additional section.

This *harmonic* appears and disappears because it is a by-product of the 1280 size honoured by the modified server.

Harmonic B are typically Name Error responses to queries for records in numerical top level domains with only one NSEC RR and corresponding RRSIG in the authority section. (The NSEC that denies existence of numerical TLDs also denies the existence of the '*') for ZSK 2048 *Harmonic B* also includes the packets from *Harmonic 4* with their signatures stripped.

Harmonic X only occurs responses from the zone signed with a 2048 bit ZSK.

These packets typically contain a Name Error response. It has a two NSEC RRs proof for non existence in the authority section. It does not have DNSKEYs in the additional section. These are the packets that did not fit in *Harmonic 3*.

It is *Harmonic X* that shows the distinct peak in packet distribution for 2048 ZSK zones served by the modified `nsd` 2.3.0 server (bottom right plot of figure 14. For smaller key sizes this harmonic is convolved with other harmonics.

In the figures for the modified `nsd` responses there is one distinct peak at 1625 and 2009 octets for the 512 and 1024 bits ZSK distributions(Figure 14, bottom left plot, second and third frame from the bottom).

Those peaks are caused by responses to the `. IN ANY` query. The amplitude of the peak is 518 packets. The reason for this peak not to occur for the 1592 bits ZSK responses is that these responses do not fit and the packets are truncated. This is confirmed in Table 6 where these packets show up as being truncated.

The peak also shows in the 512 ZSK graph (at 1852 octets) and occurs at 1801 octets in the 1024 ZSK graph for `named`. The content – or the sizes – of the responses to the ANY query are different. The `named` comes with an empty additional section while `nsd` adds A glue records to the additional section.

These peaks do not occur in the unsigned zone either. We did not include the DNSKEY RRset in the the unsigned version of the zone. The response to the ANY query then only measures 296 octets for `named` and 493 octets for `nsd`, the latter including glue.

The distributions for the unsigned zones differ slightly between the unmodified and modified servers (compare the bottom frames of the same row). This is caused by the fact that the modified treats all clients as if they are able to deal with responses of 1280 bytes or more. The unmodified server will need to honour the 512 byte size limit for the 65.5% of the queries that do not have the EDNS0 extension.

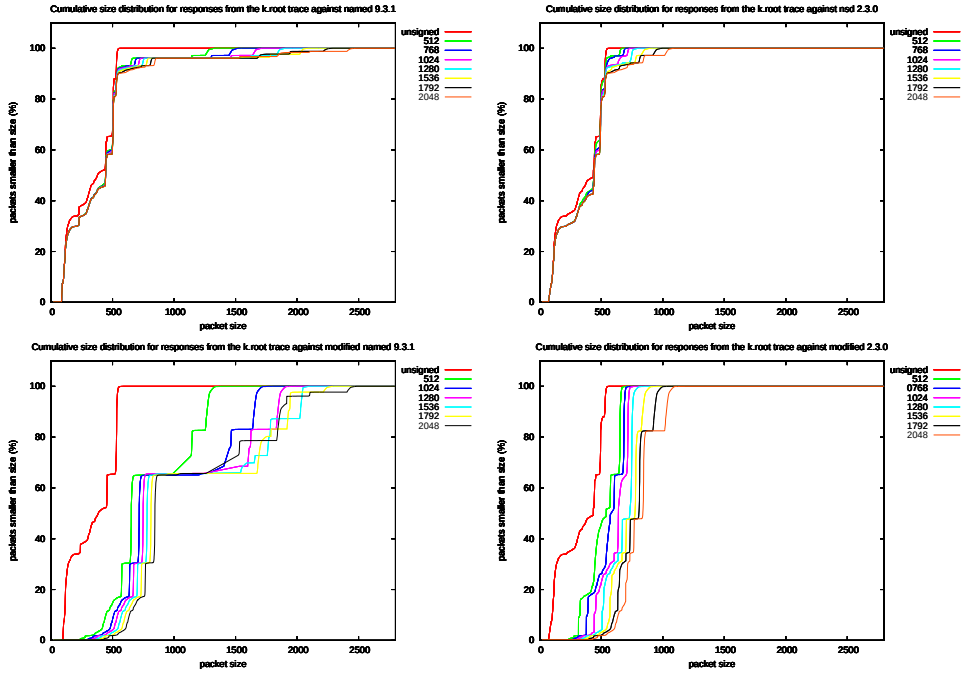

Figure 15: Cumulative Packet Sizes for Replies From the k.root Traces

Figure 15 shows the cumulative size distributions.

The graphs indicate that the amount of signal in the *harmonics* is rearranged for larger key sizes.

The most important conclusion that can be drawn from the graph is that when the DNSKEY RRs with their signatures are not included in the responses, the `nsd` behaviour packet sizes will not grow beyond 1259 octets, well beyond the MTU on Ethernet networks.

Acknowledgements

Thanks Daniel Karrenberg for the design and implementation of the DISTEL lab, proofreading of the manuscript and the *Vlaai*[5] during his instructions on how to operate the lab. Erik Rozendaal for coding the modifications to NSD. Mark Andrews for explaining the EDNS0 behaviour for BIND. People that helped me by asking clever questions or suggesting directions are Joao Damas, Lorenzo Colitti and Mark Santcroos. Adrian Bedford and Jaap Akkerhuis for proofreading the manuscript.

Most of this work was done while I was employed by the RIPE NCC.

References

[ADF05] Bernhard Ager, Holger Dreger, and Anja Feldmann. *Exploring the Overhead of DNSSEC*, April 2005. `http://www.net.informatik.tu-muenchen.de/~anja/`

[5] A treat from the province of Limburg

feldmann/papers/dnssec05.pdf, Work in progress.

[AS] Roy Arends and Jakob Schlyter. *fpdns - Fingerprinting DNS servers*. FPDNS webpages. http://www.rfc.se/fpdns/.

[BIND] ISC BIND nameserver webpage. http://www.isc.org/index.pl?/sw/bind/.

[KG05] Olaf Kolkman and Miek Gieben. *DNSSEC Operational Practices <draft-ietf-dnsop-dnssec-operational-practices-05.txt>*, September 2005. ftp://ftp.ietf.org/internet-drafts/draft-ietf-dnsop-dnssec-operational-practices-05.txt, DNSOP WG Internet draft, drafts are subject to change and have a limited lifetime.

[KYLK02] Daniel Karrenberg, Alexis Yushin, Ted Lindreen, and Olaf Kolkman. *DISTEL-Domain Name Server Testing Lab*, November 2002. http://www.ripe.net/ripe/meetings/ripe-43/presentations/ripe43-dnr-distel/.

[NSD] NLnet Labs NSD nameserver webpage. http://www.nlnetlabs.nl/nsd/.

[RFC2671] P. Vixie. *RFC 2671: Extension Mechanisms for DNS (EDNS0)*. IETF, August 1999. ftp://ftp.ietf.org/rfc/rfc2671.txt.

[RFC4033] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *RFC4033: DNS Security Introduction and Requirements*. IETF, March 2005. ftp://ftp.ietf.org/rfc/rfc4033.txt.

[RFC4034] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *RFC4034: Resource Records for the DNS Security Extensions*. IETF, March 2005. ftp://ftp.ietf.org/rfc/rfc4034.txt.

[RFC4035] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. *RFC4035: Protocol Modifications for the DNS Security Extensions*. IETF, March 2005. ftp://ftp.ietf.org/rfc/rfc4035.txt.

Note: URLs may be subject to change. For up to date versions of Internet Drafts please consult https://datatracker.ietf.org/public/idindex.cgi

Document history: This RIPE document is based on *Revision* : 33 of the manuscript source.