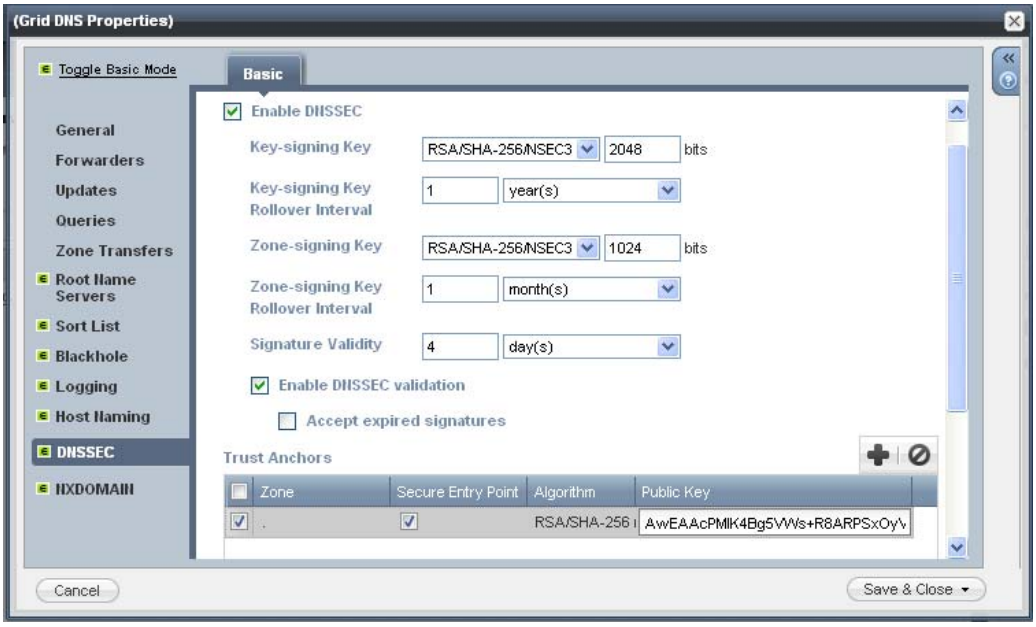
Set up of the trust anchors on NIOS GUI



▼Verification item <F-194>: Self-signed DNSKEY with the REVOKE bit should be revoked.

▼Verification item <F-196>: Revoked DNSKEY should not be used as a trust anchor.

▼Verification item <F-201>: New keys should be added to the trust anchor when the time limit has passed.

▼Verification item <F-202>: New keys should not be added to the trust anchor before the time limit has passed.

The test was not conducted in this technology experiment for verification items: <F-194>, <F-196>, <F-201> and <F-202>.

▼Verification item <F-229>: The NSEC3PARAM record should exist at the apex of the zone.

[root@centos bin]# ./dig +dnssec @192.168.200.130 jp NSEC3PARAM

; <<>> DiG 9.6.2-P2 <<>> +dnssec @192.168.200.130 jp NSEC3PARAM
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34689
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;jp. IN NSEC3PARAM

```
;; ANSWER SECTION:
jp.                     0       IN      NSEC3PARAM 1 0 10 CAFE
jp.                     0       IN      RRSIG   NSEC3PARAM 8 1 0 20100916030235
20100817030235 54813 jp.
kEmt4tYBDaH4pUVNl/M++G3q/QTZQLTwk8QEdVIeCYTTKAdfA8gxMpvw
tMpcxng2swE65XP5PRNwWRi133Ku4RbCfu1CcgC+7pkC7T6y/SkPNqHf
9NxjAFkGpXGNdigyKaet+JrOV9ItCUmUVL/bs+G8ei50gGSclJ/HAneI
oxRufg6afqs6ohLzGfeqWK+A7BTCJvbR+mpNr3yeazpBqmg6aX89fb3w
/3sLqUPrOJCkif4+Z1GFiTiv+8vpUiHqZ3uHaj23gCythrhd5r1SBL2Z
QjBvXkLWRUnjN1oVJNs+34L3ZUDoMM3gPtqzqpBMm0uK2hiun5jrYNMA c5p9ng==

;; Query time: 13 msec
;; SERVER: 192.168.200.130#53(192.168.200.130)
;; WHEN: Tue Aug 17 20:37:34 2010
;; MSG SIZE  rcvd: 340
```

■ Case Study 5: Obtained Findings

▼ Infoblox NIOS compatibility with the RSA/SHA-2 signature algorithm

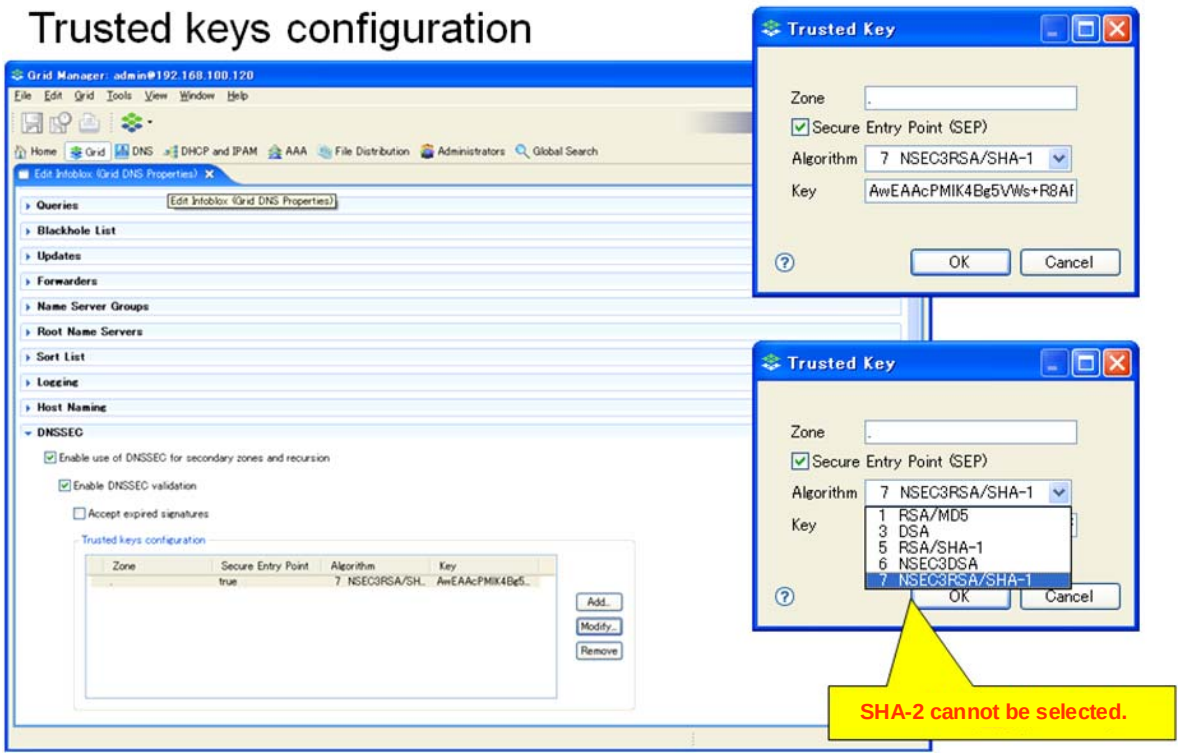
Infoblox NIOS (4.3r3 and above) supports the secondary name server for the DNSSEC zone and the import of trust anchors for the cache (full-service resolver) name server, and NIOS (5.0r1 and above) supports the key management of the primary name server for the DNSSEC zone and a signature function. However, only NIOS (5.1r2 and above) supports the RSA/SHA-2(SHA-256/SHA-512) signature algorithm which is used in root zone.

In the case of implementing the DNSSEC-compliant cache name server using the Infoblox appliance in the future, it is recommended to deploy NIOS (5.1r2 and above) which supports the RSA/SHA-2 signature algorithm.

▼ Example of a validation failure in the cache server due to inconsistency in the signature algorithm

Test result for NIOS (4.3r6)

The RSA/SHA-2 signature algorithm is used for the pseudo root server.



When the DNSSEC validation of a cache server is enabled, the test fails as follows.

```
C:¥dig>dig @192.168.200.120 jprs.jp a +dnssec

; <<>> DiG 9.3.2 <<>> @192.168.200.120 jprs.jp a +dnssec
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 1797
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;jprs.jp.                       IN      A

;; Query time: 187 msec
;; SERVER:  192.168.200.120#53(192.168.200.120)
;; WHEN: Fri May 21 14:07:18 2010
;; MSG SIZE  rcvd: 36
```

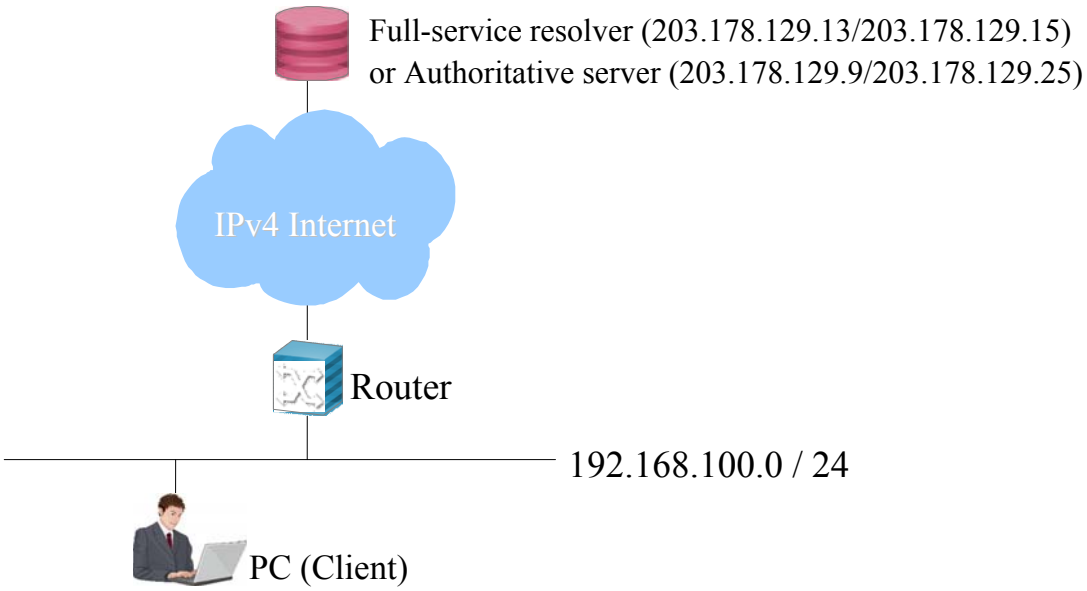
SERVFAIL is returned.

Validation fails due to inconsistency in the signature algorithm.

```
May 21 14:11:55 (none) named[23718]: client 192.168.1.23#3712:  query: jprs.jp IN A +ED
May 21 14:11:55 (none) named[23718]: validating @0x85a2258: jprs.jp A: no valid signature found
May 21 14:11:55 (none) named[23718]: validating @0x85b02c8: jp DS: no valid signature found
May 21 14:11:55 (none) named[23718]: no valid RRSIG resolving 'jp/DS/IN': 203.178.129.5#53
May 21 14:11:55 (none) named[23717]: validating @0x85b02c8: jp DS: no valid signature found
May 21 14:11:55 (none) named[23717]: no valid RRSIG resolving 'jp/DS/IN': 203.178.129.4#53
May 21 14:11:55 (none) named[23717]: no valid DS resolving 'jprs.jp/A/IN': 203.178.129.9#53
May 21 14:11:55 (none) named[23717]: validating @0x85ae2b8: jprs.jp A: no valid signature found
May 21 14:11:55 (none) named[23717]: validating @0x85b02c8: jp DS: no valid signature found
May 21 14:11:55 (none) named[23717]: no valid RRSIG resolving 'jp/DS/IN': 203.178.129.4#53
May 21 14:11:55 (none) named[23718]: validating @0x85b02c8: jp DS: no valid signature found
May 21 14:11:55 (none) named[23718]: no valid RRSIG resolving 'jp/DS/IN': 203.178.129.5#53
May 21 14:11:55 (none) named[23718]: no valid DS resolving 'jprs.jp/A/IN': 203.178.129.8#53
```

Verification of Functionality: Case Study 6

■ Case Study 6: Experimental Environment



* CentOS5.4 is used for PC (Client).
  BIND-9.7.0rc1 is installed.

■ Case Study 6: Summary of Experimental Results

By testing the following verification items on a full-service resolver and an authoritative server for both cases with and without a signature, it has been verified that the router's behaviours do not interfere with the DNSSEC communication during the transparent transmission using a router with a standard filter which sufficiently meets DNS requirements from a client.

- <F-85>: Full-service resolver (security-compliant) should have a UDP communication capability via EDNS0.
- <F-86>: Full-service resolver (security-compliant) should support UDP messages of 1220 bytes.
- <F-87>: Full-service resolver (security-compliant) should support UDP messages of 4000 bytes.
- <A-85>: Authoritative server (security-compliant) should have a UDP communication capability via EDNS0.
- <A-86>: Authoritative server (security-compliant) should support UDP messages of 1220 bytes.
- <A-87>: Authoritative server (security-compliant) should support UDP messages of 4000 bytes.

■ Case Study 6: Detailed Experimental Results

▼Make queries to the full-service resolver (with a signature).

○ Verification item <F-85>: Full-service resolver (security-compliant) should have a UDP communication capability via EDNS0.

1. Make queries whose response results will exceed 512 bytes to the full-service resolver.

- Make queries by adding a <+dnssec> option to the dig command.
- Add the full-service resolver's address after @.
- Specify a zone name for the signature zone.
- Specify < DNSKEY> for a record type. Verify the following.

Verified the following items:

- Verify that DNSKEY records and RRSIG records for the zone are included in the response.
- Verify that the data volume of the response results (MSG SIZE rcvd:) exceeds 512 bytes.
- Verify that <;; Truncated, retrying in TCP mode> is not shown just below the results of the dig command.
- Verify that the udp of the OPT PSEUDOSECTION section of the response results shows 4096.

2. Perform queries by adding the <+bufsize=512> option.

Verified the following item:

- Verify that <;; Truncated, retrying in TCP mode> is shown just below the results of the dig command. This is because the data volume of the server's response to the query exceeded the maximum size of the UDP specified by the PC.

○ Verification item <F-86>: Full-service resolver (security-compliant) should support UDP messages of 1220 bytes.

Make queries whose response results will exceed 1220 bytes to the full-service resolver.

Verified the following items:

- Verify that DNSKEY records and RRSIG records for the zone are included in the response.
- Verify that the data volume of MSG SIZE rcvd: exceeds 1220 bytes.
- Verify that <;; Truncated, retrying in TCP mode> is not shown just below the results of the dig command.

○ Verification item <F-87>: Full-service resolver (security-compliant) should support UDP messages of 4000 bytes.

By setting another full-service resolver (Fedora10, BIND-9.7.0rc1) locally, ensure that the MSG SIZE will exceed 4000 bytes.
Verified the following item:

- Verify that normal response could be obtained for queries whose response results may exceed 4000 bytes.

▼ Make queries to the full-service resolver (without a signature).

○ Verification item <F-85>: Full-service resolver (security-compliant) should have a UDP communication capability via EDNS0.

1. Make queries whose response results will exceed 512 bytes to the full-service resolver.

- Make queries by adding a <+dnssec> option to the dig command.
- Add the full-service resolver's address after @.
- Specify a zone name without a signature.
- Specify < DNSKEY> for a record type. Verify the following.

Verified the following items:

- Verify that DNSKEY records and RRSIG records for the zone are not included in the response.
- Verify that <;; Truncated, retrying in TCP mode> is not shown just below the results of the dig command.
- Verify that the udp of the OPT PSEUDOSECTION section of the response results shows 4096.

○ Verification item <F-86>: Full-service resolver (security-compliant) should support UDP messages of 1220 bytes.

Could not verify this because there was no signature and it was not possible to make the MSG SIZE larger.

○ Verification item <F-87>: Full-service resolver (security-compliant) should support UDP messages of 4000 bytes.

Could not verify this because there was no signature and it was not possible to make the MSG SIZE larger.

▼ Make queries to the authoritative server (with a signature).

・ ○ Verification item <A-85>: Authoritative server (security-compliant) should have a UDP communication capability via EDNS0.

1. Make queries whose response results will exceed 512 bytes to the authoritative server.

- Make queries by adding a <+dnssec> option to the dig command.
- Add the authoritative server's address after @.
- Specify a zone name for the signature zone.
- Specify < DNSKEY> for a record type.

Verified the following items:

- Verify that DNSKEY records and RRSIG records for the zone are included in the response.
- Verify that the data volume of the response results (MSG SIZE rcvd:) exceeds 512 bytes.
- Verify that <;; Truncated, retrying in TCP mode> is not shown just below the results of the dig command.
- Verify that the udp of the OPT PSEUDOSECTION section of the response results shows 4096.

2. Perform queries by adding the <+bufsize=512> option.

Verified the following item:

- Verify that <;; Truncated, retrying in TCP mode> is shown just below the results of the dig command. This is because the data volume of the server's response to the query exceeded the maximum size of the UDP specified by the PC.

○ Verification item <A-86>: Authoritative server (security-compliant) should support UDP messages of 1220 bytes.

Make queries whose response results will exceed 1220 bytes to the authoritative server.

Verified the following items:

- Verify that DNSKEY records and RRSIG records for the zone are included in the response.
- Verify that the data volume of MSG SIZE rcvd: exceeds 1220 bytes.
- Verify that <;; Truncated, retrying in TCP mode> is not shown just below the results of the dig command.

○ Verification item <A-87>: Authoritative server (security-compliant) should support UDP messages of 4000 bytes.

Could not verify this because the test was not possible without setting up another authoritative server locally.

▼Make queries to the authoritative server (without a signature).

○ Verification item <A-85>: Authoritative server (security-compliant) should have a UDP communication capability via EDNS0.

1. Make queries whose response results will exceed 512 bytes to the authoritative server.

- Make queries by adding a <+dnssec> option to the dig command.
- Add the authoritative server's address after @.
- Specify a zone name for the authoritative zone.
- Specify < DNSKEY> for a record type.

Verified the following items:

- Verify that DNSKEY records and RRSIG records for the zone are included in the response.
- Verify that <;; Truncated, retrying in TCP mode> is not shown just below the results of the dig command.
- Verify that the udp of the OPT PSEUDOSECTION section of the response results shows 4096.

2. Perform queries by adding the <+bufsize=512> option.

Verified the following item:

- Verify that <;; Truncated, retrying in TCP mode> is shown just below the results of the dig command. This is because the data volume of the server's response to the query exceeded the maximum size of the UDP specified by the PC.

○ Verification item <A-86>: Authoritative server (security-compliant) should support UDP messages of 1220 bytes.

Could not verify this because there was no signature and it was not possible to make the MSG SIZE larger.

○ Verification item <A-87>: Authoritative server (security-compliant) should support UDP messages of 4000 bytes.

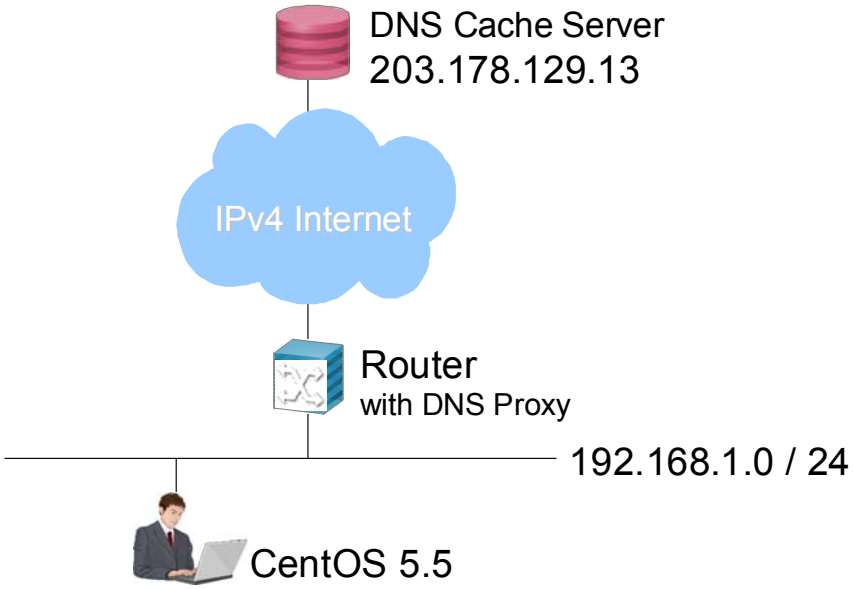
Could not verify this because there was no signature and it was not possible to make the MSG SIZE larger.

■ Case Study 6: Obtained Findings

Although some verification items could not be tested, our findings generally suggest that the router's filter and transfer functions do not interfere with the DNSSEC communication for a full-service resolver and an authoritative server with or without a signature.

Verification of Functionality: Case Study 7

■ Case Study 7: Experimental Environment



■ Case Study 7: Summary of Experimental Results

Verification of the DNS Proxy function for in-house broadband routers (for corporate and private use) was conducted in reference to <F-85>, <F-86> and <F-87> of the "DNSSEC Verification of Functionality: Procedure Manual" and the [RFC5625] DNS Proxy Implementation Guidelines.

■ Case Study 7: Detailed Experimental Results

(1) Verification was conducted to see whether the devices could process OPT RR and various flags transparently.

=> It was verified that all devices were normal.

(2) Verification was conducted to see whether the devices were EDNS0-compliant and could process packets of 1,220 bytes properly.

=> It was verified that some devices could not process properly.

(3) Verification was conducted to see whether the devices were EDNS0-compliant and could process packets of 4,000 bytes properly.

=> It was verified that some devices could not process properly.

(4) Verification was conducted to see whether the TCP Fallback was functioning properly.

=> It was verified that the TCP Fallback was not functioning properly for some devices.

(5) Verification was conducted to see the impact of the DNS cache on the devices.

=> It was verified that the DNS cache impacted on some devices negatively.

■ Case Study 7: Obtained Findings

It was verified that packets exceeding 512 bytes could not be processed by some devices and that there were some conditions in which the DNSSEC validation could not be conducted in the case of a cache implementation only for specific RRs.

Although it is desirable to implement a cache function not only for specific RRs but also for all kinds of RRs, we think that it is realistic not to implement a cache function itself as there are too many things to consider.

(1) Issue of not being able to process packets exceeding 512 bytes

Although the packet size is cut down to 512 bytes for the transfer when response packets exceeding 512 bytes are received, the stub resolver cannot fallback to TCP as packets are transferred using “TC=0.”

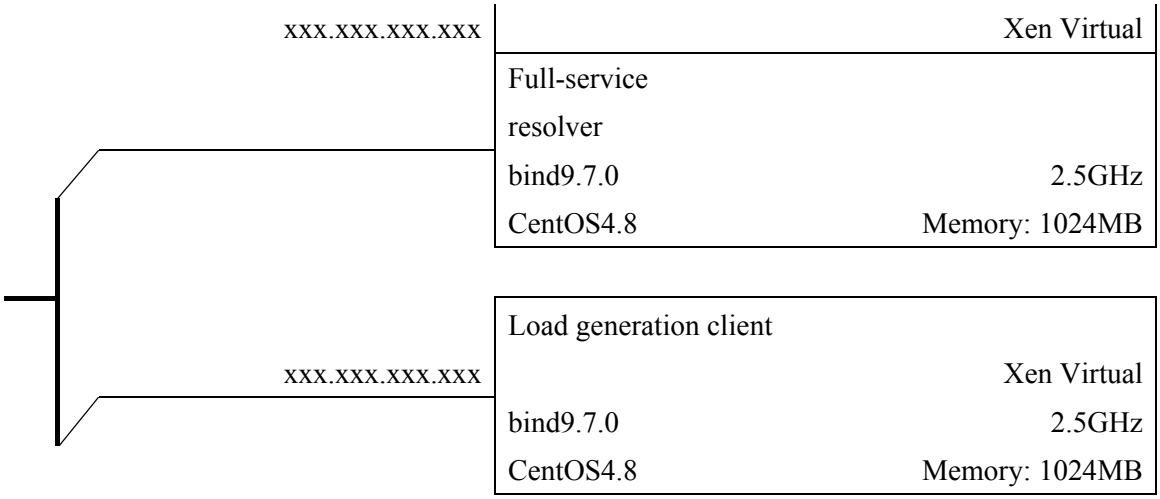
(2) Issue of the cache implementation for specific RRs only

In the cache implementation for specific RRs only, the RRSIG RRs cannot be cached. In this condition, the DNSSEC validation cannot be conducted until the cache entries expire (TTL expiry) because only cached RRs are returned as a response when queries are sent from other devices.

Results of Performance Verification

Performance Verification: Case Study 1

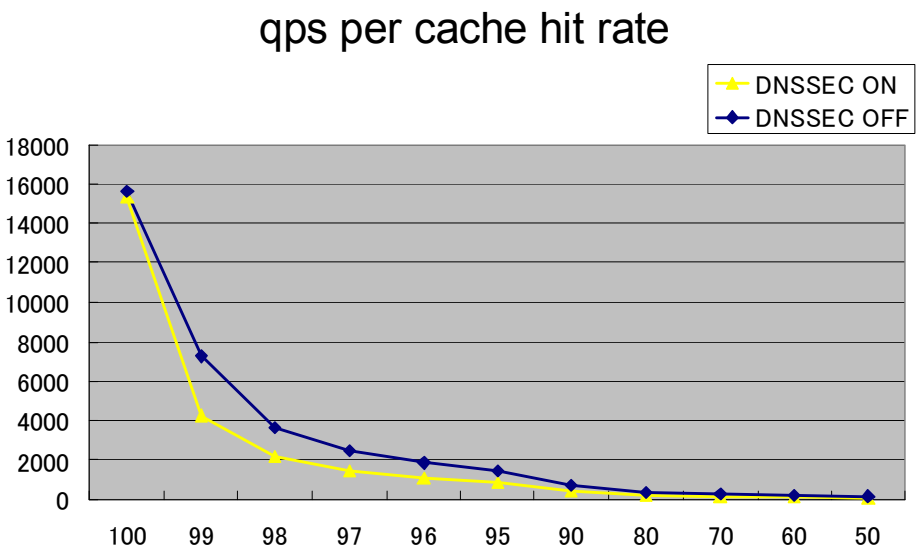
■ Case Study 1: Experimental Environment

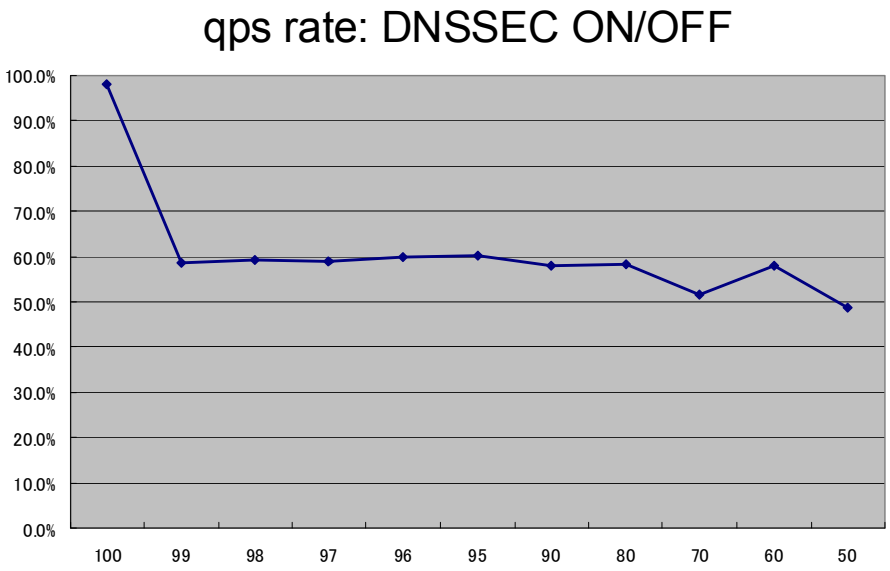


■ Case Study 1: Summary of Experimental Results

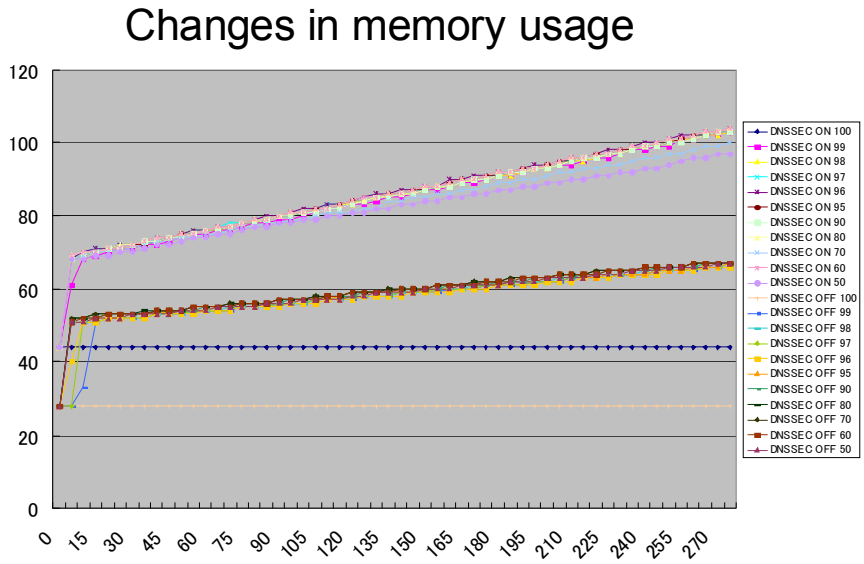
It was verified that the full-service resolver's CPU usage rate and memory usage increased and the query processing capability decreased by enabling DNSSEC.

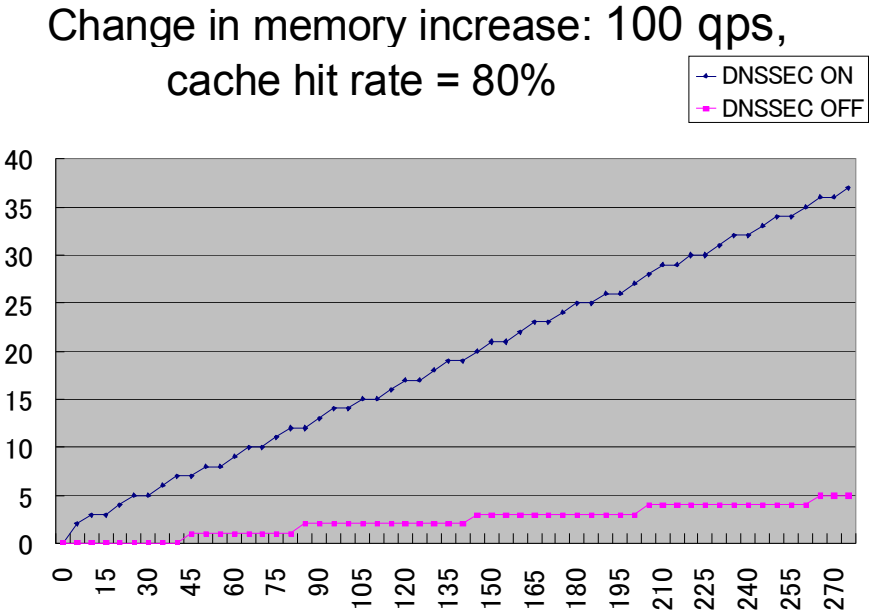
■ Case Study 1: Detailed Experimental Results(1)





■ Case Study 1: Detailed Experimental Results(2)





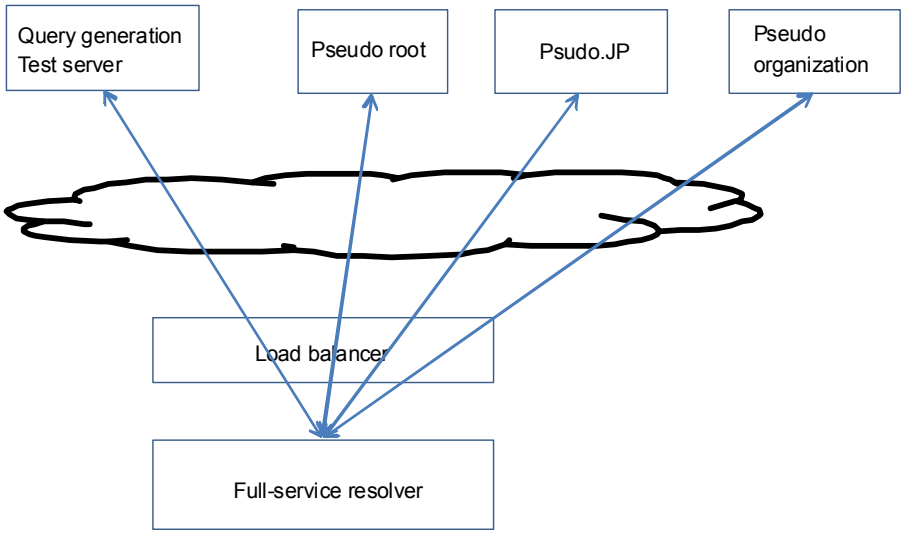
■ Case Study 1: Obtained Findings

The memory usage increased by eight times when NSEC3 was used and the query results were mostly "NXDOMAIN."

The query processing capability decreased to 60% when DNSSEC was ON.

Performance Verification: Case Study 2

■ Case Study 2: Experimental Environment



Functionality	HW	Application	Note
Full-service resolver	SUN X2100 Solaris10	bind-9.7.0	IPv4 only
Load balancer	A10networks AX2500		IPv4 only
Query generation	SUN NetraT1	queryperf (modified version)	IPv4 only

■ Case Study 2: Summary of Experimental Results

The following tests were conducted using a cache server based on the “5. Measurement Procedures” in the “DNSSEC Performance Verification: Procedure Manual (Ver. 1.1)” by JPRS.

5. Measurement Procedures

5.1. Measurement of Behavioral Changes for Various Patterns of the Validator

a) DNS name resolution by a <dig> command and verification of the DNSSEC validation

1. The following configurational changes should be made depending on the pattern.

- Changes to the network configuration of the validator server (MTU, TCP, fragment)
- Changes to the configuration file of the validator server (DO=0/1, TA configuration)
- Changes to the zone data which will be configured in the authoritative server (ZSK=1024/2048, without a signature)

2. After the above configurational changes are made, verifications should be conducted on the validator server by using the following <dig> commands when both the validator and

authoritative servers are in operation. It should be verified whether the name resolution and the validation are successful or not by looking at the output results of the <dig> command.

Without a signature:
dig @localhost example.jp. A
With a signature:
dig @localhost +dnssec example.jp. A

b) The validator load and the queries to the authoritative server should be measured for the pattern with the name resolution.

1. As in a), the network and server configurations should be changed depending on the pattern.
2. The load measurement tool (*) should be activated on the validator server and the CPU usage rate and the memory usage should be measured.

* Scripts to measure the CPU usage rate, memory usage, load average, etc. should be prepared.

3. DSC (DSC Collector) should be activated on the validator and authoritative servers.
4. Load tests by queryperf (modified version) should be conducted on the query generator. Commands should be changed by DO bits.

Transmission intervals should be specified on the millisecond time scale for the <-i> option. In the following example, the transmission interval was “10,000 qps” as “0.1ms” was specified.

DO=0
queryperf -d query.txt -s 192.0.2.1 -l 300 -i 0.1
DO=1
queryperf -d query.txt -s 192.0.2.1 -D -l 300 -i 0.1

5. After the load tests are finished, the load measurement tool/DSC should be turned off on the validator and authoritative servers.

In the above procedures, switching of patterns during the validation and measurements should be conducted as follows.

■ Case Study 2: Detailed Experimental Results

RE: “5.1. a)-1.”of the Measurement Procedures

- The network configurational changes (MTU, TCP, fragment, etc.) were not conducted because the experimental environment coexisted with the production environment and the changes could not be made.
- With regard to the configuration file change of the validator, behavioral changes were monitored with and without the TA configuration.
- The tests were not conducted for the authoritative server because it was not set up.

RE: “5.1. b)” of the Measurement Procedures

A test query by queryperf which would not cause a cache hit for five consecutive minutes could not be prepared. Therefore the load to the server due to the DNSSEC validation decreased to the level which was lower than expected after a while, which was believed to be due to the cache hit. Consequently, a prominent difference in the load to the server could not be identified with or without the DNSSEC validation.

However, it was verified that the cache size was more than doubled when accessed a pseudo tree with a signature compared to when accessed a pseudo tree without a signature.

■ Case Study 2: Obtained Findings

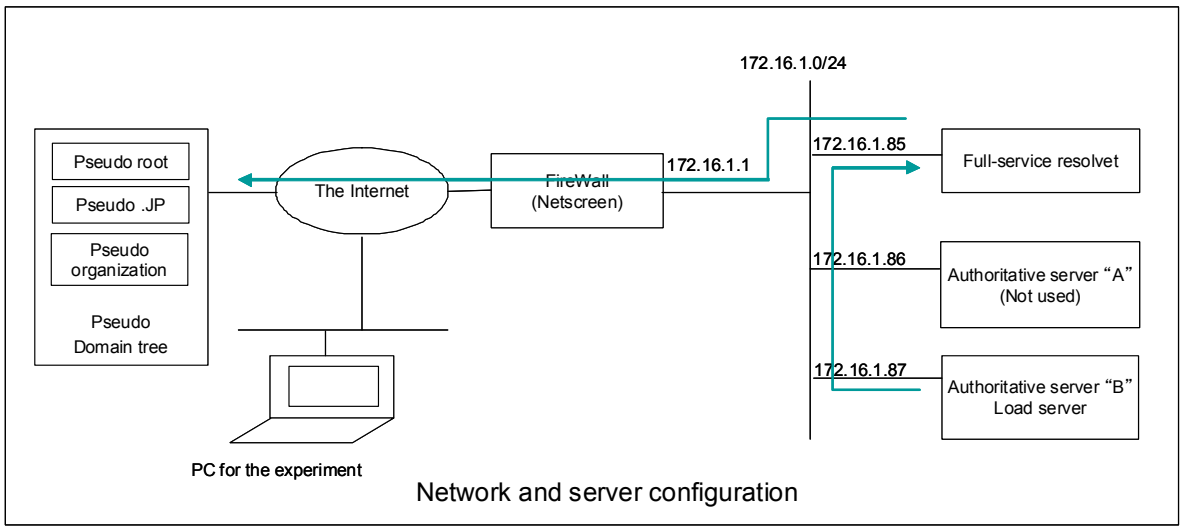
A prominent difference in the server load could not be identified with or without the DNSSEC validation.

However, as the cache size is expected to be more than double when DNSSEC is on, we think that it is necessary to replace servers, as appropriate, in light of the server load and the processing capability of servers in the service environment.

Performance Verification: Case Study 3

■ Case Study 3: Experimental Environment

・Experimental Environment Configuration



・ Software configuration:

OS:Solaris10　BIND:9.7.0-P1

・ Load tool:

queryperf　dnsperf

■ Case Study 3: Summary of Experimental Results

<Experiment Procedures>

・ Processing status is monitored by deploying "dnsperf/queryperf" in the testing device and loading the full-service resolver.

・ The DNS query information is obtained as a sample and processed into a data format suitable for the test.

・ 100,000 records are used as a data for the test. (Not all of the records are uniq.)

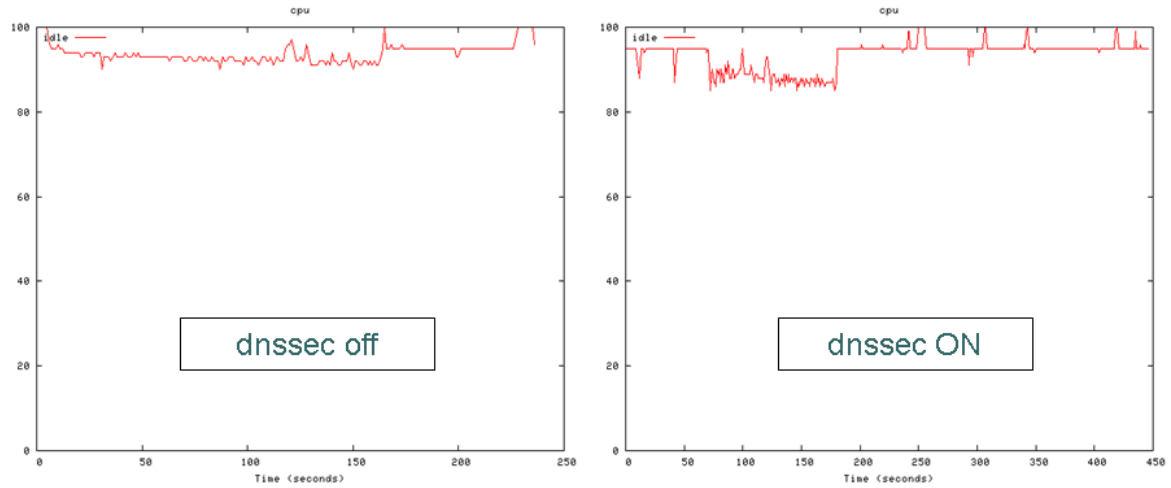
・ "vmstat" and "top" are used to measure the performance.

<Result>

・ The load to the full-service resolver and the memory usage increased due to the processing of the DNS request for DNSSEC.

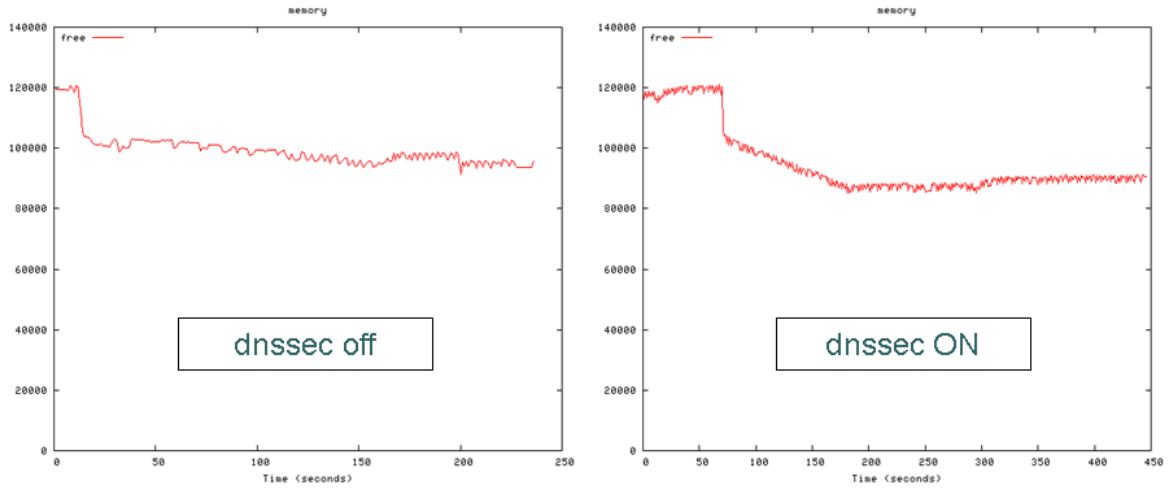
■ Case Study 3: Detailed Experimental Results

● Comparison on the CPU value: dnssec On/Off



It was verified that the CPU load increased when DNSSEC was on compared to when DNSSEC was off. (Due to the band width restrictions of the upper network, the processing number of DNS decreased and the full CPU capability could not be utilized.)

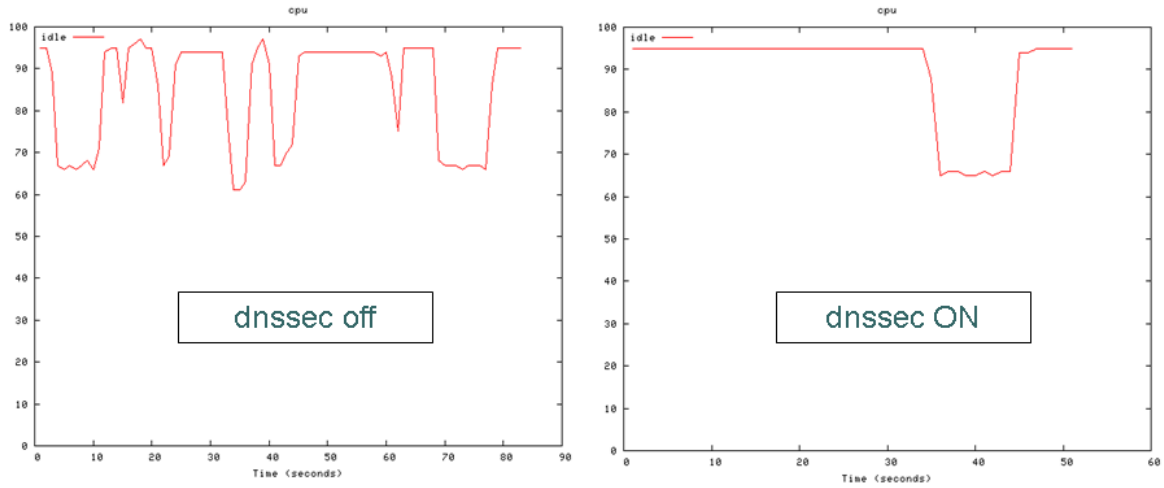
● Comparison of the memory value: dnssec On/Off



A prominent difference in memory consumption was not observed when DNSSEC was on compared to when DNSSEC was off.

(There is a possibility that the difference in data was not observed because the number of <uniq> queries used for the test was too few.)

● Comparison of CPU: full cache



The CPU load with the cache hit rate of 100% showed the same value when DNSSEC was on and when DNSSEC was off.

It seemed that there is only little (or no) impact of the DNSSEC validation on cached information.

■ Case Study 3: Obtained Findings

No significant differences due to the testing environment or restrictions of network configurations were observed as a result of the performance comparison in the testing environment.

However, it was verified that the load increased marginally when DNSSEC was on as a result of this experiment.

Since the load to the full-service resolver changes depending on the DNSSEC usage rate, we think that it is necessary to monitor the following for the full-service resolver to be used for service on an ongoing basis and reinforce and expand the system.

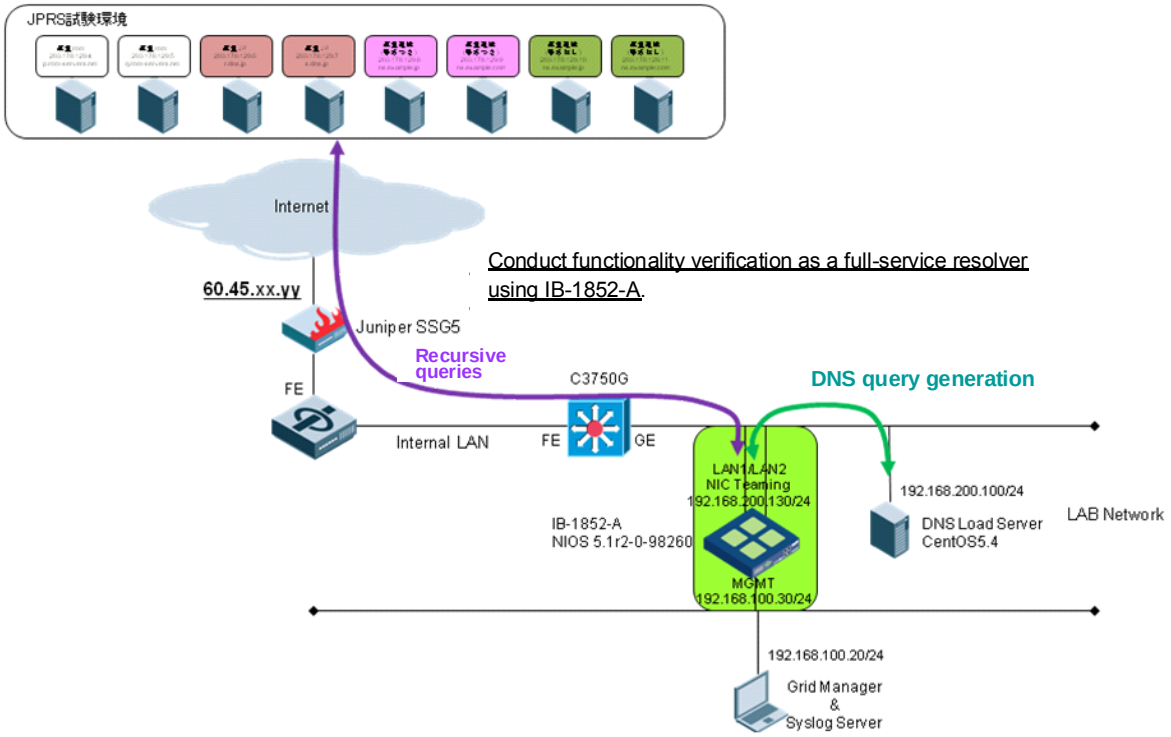
- Memory usage
- Cache hit rate
- Server CPU usage
- Network band width usage
- Number of connections

Furthermore, a prominent difference in performance was observed due to the cache hit rate when DNSSEC was on compared to when DNSSEC was off.

It seemed that it is necessary to consider avoiding busy hours when stopping/starting the server (or clearing cache). It is also necessary to establish a method for warming up.

Performance Verification: Case Study 4

■ Case Study 4: Experimental Environment



DNS APPLIANCE DEVICE: Infoblox 1852-A Network Service Appliance
Software version: NIOS 5.1r2-0-98260 * NIOS = Name of the internal OS of the Infoblox appliance

■ Case Study 4: Summary of Experimental Results

▼ The impact on appliance load was verified with the cache hit rate of 100% with and without the DNSSEC validation. The test was conducted by sending a test query (www.xxx.co.jp A) of 100 unique domains from a Linux load generation server to the IB-1852-A DNS cache server by using "queryperf."

The test was conducted only for the IPv4 address by using the RSA 2024 bit-type testing environment of a pseudo tree under the DNS technology experimental environment.

* Verification items in accordance with the "DNSSEC Performance Verification: Procedure Manual (Ver. 1.2) were not conducted.

▼ Experimental result:

With regard to the degree of impact on performance of a cache server, it was verified that the performance was down by approximately 13% when the DNSSEC validation was on compared to when the DNSSEC validation was off.

It was also verified that the performance was impacted by the additional message size of the RRSIG record for the DNS query response.

■ Case Study 4: Detailed Experimental Results

▼IB-1852-A, without the DNSSEC validation, 100% cache hit rate

Queryperf, without the <-D> option

1st time: 141238.4 qps Appliance CPU load average: 81%

2nd time: 142645.4 qps Appliance CPU load average: 84%

3rd time: 142131.4 qps Appliance CPU load average: 83%

4th time: 141813.0 qps Appliance CPU load average: 82%

5th time: 141439.3 qps Appliance CPU load average: 82%

Without the DNSSEC validation, example of a <dig> response

; <<>> DiG 9.3. 6-P1-RedHat-9.3. 6-4.P1.el5_4.2 <<>> @192.168.200.130 www.xxxxx.co.jp A

; (1 server found)

;; global options: printcmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10029

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:

;www.xxxxx.co.jp. IN A

;; ANSWER SECTION:

www.xxxxx.co.jp. 861 IN A 192.0.2.1

;; Query time: 0 msec

;; SERVER: 192.168.200.130#53(192.168.200.130)

;; WHEN: Tue Aug 24 20:44:34 2010

;; MSG SIZE rcvd: 49

▼IB-1852-A, with the DNSSEC validation, 100% cache hit rate

queryperf, with the <-D> option

1st time: 123276.8 qps Appliance CPU load average: 83%

2nd time: 123065.5 qps Appliance CPU load average: 82%

3rd time: 123418.6 qps Appliance CPU load average: 84%

4th time: 122981.4 qps Appliance CPU load average: 81%

5th time: 123616.6 qps Appliance CPU load average: 84%

Without the DNSSEC validation, example of the <dig> response

; <<>> DiG 9.3. 6-P1-RedHat-9.3. 6-4.P1.el5_4.2 <<>> +dnssec @192.168.200.130 www.xxxxx.co.jp A

; (1 server found)

;; global options: printcmd

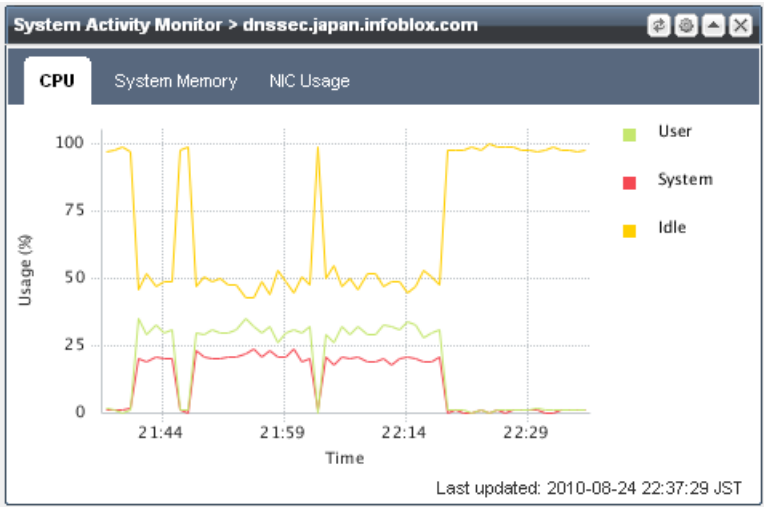
```
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25119
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.xxxxx.co.jp.                IN      A

;; ANSWER SECTION:
www.xxxxx.co.jp.        666     IN      A       192.0.2.1
www.xxxxx.co.jp.        666     IN      RRSIG   A 8 4 900 20101225225717 20091225215717
24018 xxxxx.co.jp. GPTQObu3iCAksBwl5qAo+epHdEuIfnA8dYW6MWGWLptfwMpZ/nJaYnur
GKc2MQh6zD5Q8RFFpdZrXWOWrqW9W8ffry5mmrdaEQxhSibmsoshw3GA
ymaM/J9F1UAfNQFPKLLHCGJUtdMbMxD5LtxaSBwRI07rZFyGKPYeXgs2 HHs=

;; Query time: 0 msec
;; SERVER: 192.168.200.130#53(192.168.200.130)
;; WHEN: Tue Aug 24 20:41:58 2010
;; MSG SIZE  rcvd: 231

(*Reference: shell scripts used for the test)
#!/bin/sh
SECS=300
INPUT=cached_test.data
SERVER=192.168.200.130
NUM=60
./queryperf -s $SERVER -d $INPUT -l $SECS -q $NUM -D > out1 2>&1 &
./queryperf -s $SERVER -d $INPUT -l $SECS -q $NUM -D > out2 2>&1 &
./queryperf -s $SERVER -d $INPUT -l $SECS -q $NUM -D > out3 2>&1 &
./queryperf -s $SERVER -d $INPUT -l $SECS -q $NUM -D > out4 2>&1 &
./queryperf -s $SERVER -d $INPUT -l $SECS -q $NUM -D > out5 2>&1 &
./queryperf -s $SERVER -d $INPUT -l $SECS -q $NUM -D > out6 2>&1 &
wait
grep 'Queries per' out? | awk 'BEGIN { sum=0;} { sum += $5;} END {printf("Total:
%.1f qps¥n", sum);}'
```

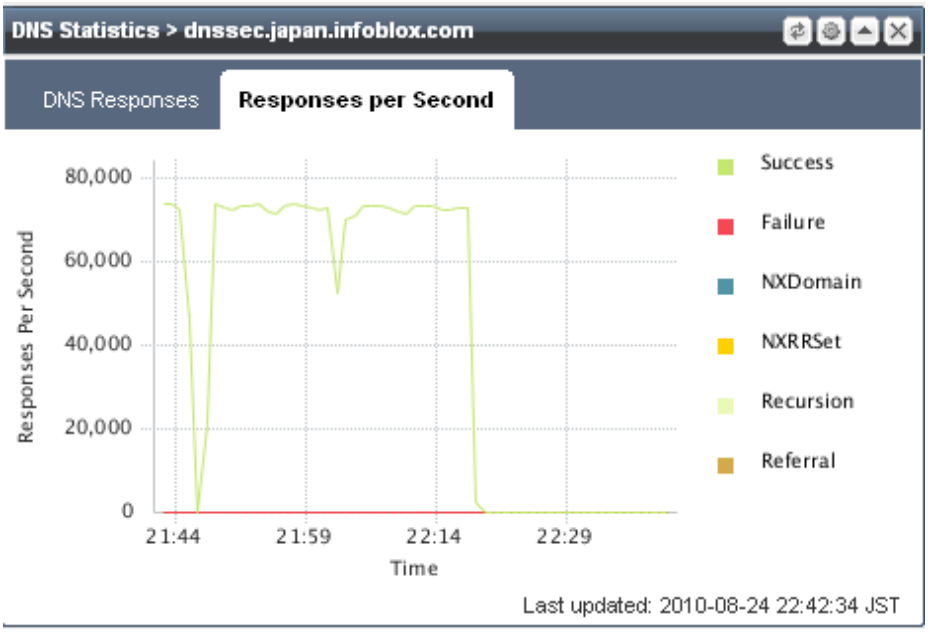
■ Case Study 4: Obtained Findings

This experiment was conducted with the cache hit rate of 100% and with and without the DNSSEC validation. When deploying DNSSEC into the production environment, it was necessary to examine the impact on performance of the cache server in consideration of the rate of incursive queries, average size of responses when the DNSSEC validation is on, etc.

(Reference)

NOIS (5.1r2 and above) has an expanded function and users can check the CPU/Memory/NIC usage rate and DNS query response statistics in a graph on the Infoblox Grid Administration page.





Performance Verification: Case Study 5

Changes in the load to the authoritative DNS server with the DNSSEC validation

■ Case Study 5: Experimental Environment

Two types of hardware were prepared for the DNS server. One server was relatively new and the other was relatively old.

	CPU	OS
Server “A”	Xeon E5540 (2.53GHz) × 2	CentOS 5.5
Server “B”	Pentium-III 1.26GHz	FreeBSD 8.0

Changes in response performance were measured with and without DNSSEC by activating BIND 9.7.x (“named”) in these servers and measuring the response performance with the load by <dnsperf> from another server connected to the LAN. Furthermore, the NSEC method and the NSEC3 method were compared when DNSSEC was on.

Changes in the load to the authoritative DNS server with the DNSSEC validation

Data used for the measurement:

- Zone data to be measured
  The zone data of a small-scale domain name in actual operation (total number of resource records = 244) was used almost as is.
- Query data used for <dnsperf>
  The data which was generated from the query log to the DNS server in the aforementioned zone
  - Although actual queries were used, DNSKEY queries, were also included because the DLV environment was used for the DNSSEC validation.
- DNSSEC parameters
  - Encrypted algorithm: RSASHA256
  - KSK key length: 2048 bits
  - ZSK key length: 1024 bits

■ Case Study 5: Experimental Results

Result 1: Comparison of response performance in each method (Unit: number of queries per second)

	Method	Server "B" (Pentium-III)		Server "A" (E5540)	
		Existence	Denial of existence	Existence	Denial of existence
Without DNSSEC	N/A	9345	8855	58423	58248
With DNSSEC	NSEC	8352	7433	57279	56642
	NSEC3	7309	3364	57122	41437

Existence: Existing domain names were extracted from the query logs.

Denial of existence: Authenticated denial of existence records generated from existence

Iterations of NSEC3: 5

The CPU usage rate of Server "A" was around 30 to 45%. Although the server in which <queryperf > was activated had the same specs with Server "A," there is a possibility that sufficient amount of the load could not be generated because <dnsperf > is a single thread. Therefore, it can be assumed that maximum capabilities of Server "A" were higher. The CPU usage rate for Server "B" was 100% for every test and it was verified that the amount of the load was sufficient.

Result 2: Average DNS response size for each method

	Method	Normal	Existence	Denial of existence
Without DNSSEC	N/A	115	115	112
With DNSSEC	NSEC	602	598	648
	NSEC3	637	604	884

Normal: The query logs were applied as is. (Denial of existence rate: approximately 8%)

Existence: Existing domain names were extracted from the query logs. (including DNSKEY)

Denial of existence: Authenticated denial of existence records generated from existence

As a reference, the DNS query size was 45 bytes on average.

■ Case Study 5: Obtained Findings

The response performance of the authoritative DNS server decreased to a certain degree when DNSSEC

was on. The rate of decease was around 10 to 20% for the response of existing names. The denial of existence response of NSEC3, in particular, could cause more than 50% of drop in processing capability.

The number of DNS response packets from the authoritative DNS server increased by five to eight times as a result of the DNSSEC validation.

Performance Verification: Case Study 6

Changes in response performance of the NSEC3 method in accordance with the number of iterations

It is known that when the NSEC3 method is used for the DNSSEC validation, the load to the server increased in response to the number of iterations. The test was conducted to measure such changes. The measurement environment was as described in the previous section. The number of iterations was increased from 0 to 100 in sequence when generating zone data , <named> was set, and the response performance by the <queryperf> command was measured repeatedly.

■ Case Study 6: Experimental Environment

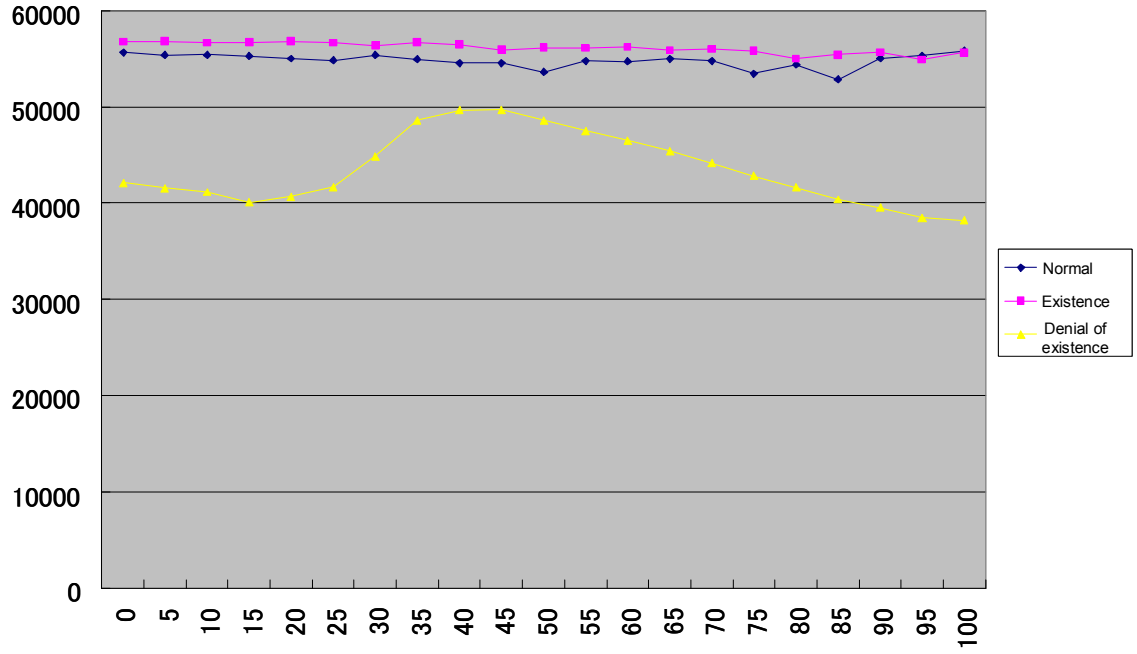
The experimental environment was the same as that of the “Performance Verification: Case Study 5.”

■ Case Study 6: Experimental Results

“Normal,” “Existence,” and “Denial of existence” in the following graphs refer to the following.

Normal: The query logs were applied as is. (Denial of existence rate: approximately 8%)
Existence: Existing domain names from the query logs were used. (including DNSKEY)
Denial of existence: Queries of the authenticated denial of existence records generated from existence

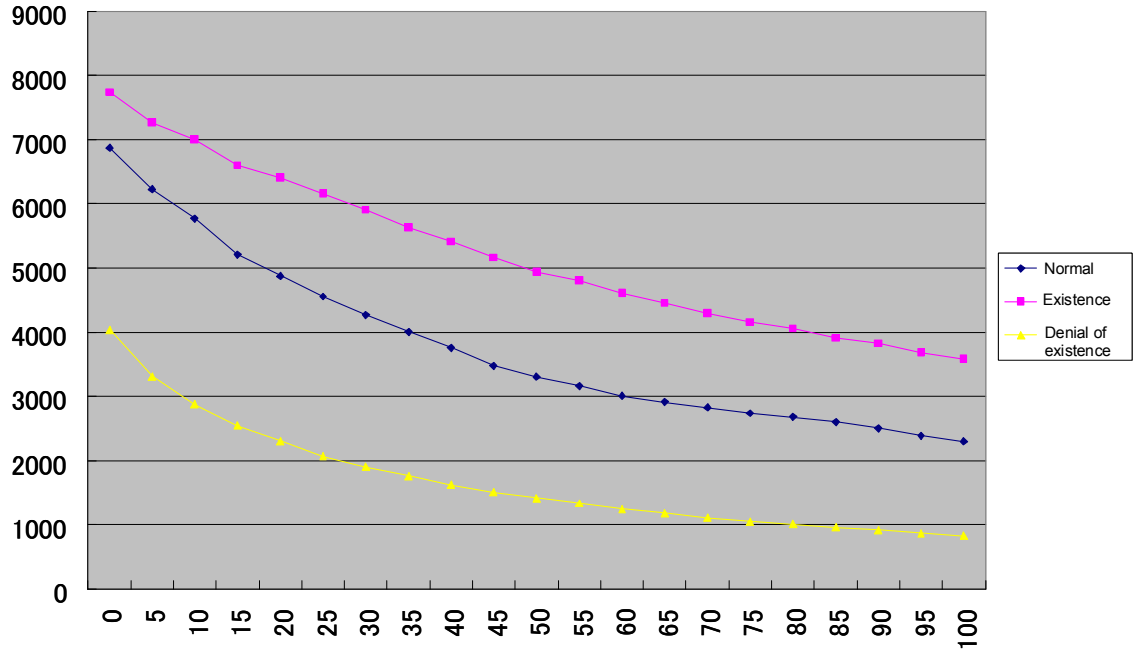
Iterations and changes in response performance for Server “A”



X-axis: Iterations; Y-axis: Response performance (qps)

During the measurement, the maximum CPU occupancy rate of <named> of Server “A” reached approximately 90% (for a response of denial of existence for 100 iterations.) Furthermore, changes in response performance of denial of existence in accordance with the changes in the number of iterations showed similar results. Although it was unnatural, we assume that it was caused by some kind of impact such as CPU characteristics.

Changes in iterations and response performance for Server “B”



X-axis: Iterations; Y-axis: Response performance (qps)

It was verified that the existence response performance of Server “B” deteriorated in accordance with the increase in the number of iterations. We assume that it was impacted by the fact that NSEC3 needs to calculate hash even for the existence responses.

■ Case Study 6: Obtained Findings

It is not desirable to make the number of iterations for the NSEC3 method extremely large because it could negatively impact the response performance. We think that “10” should cause no material impact.