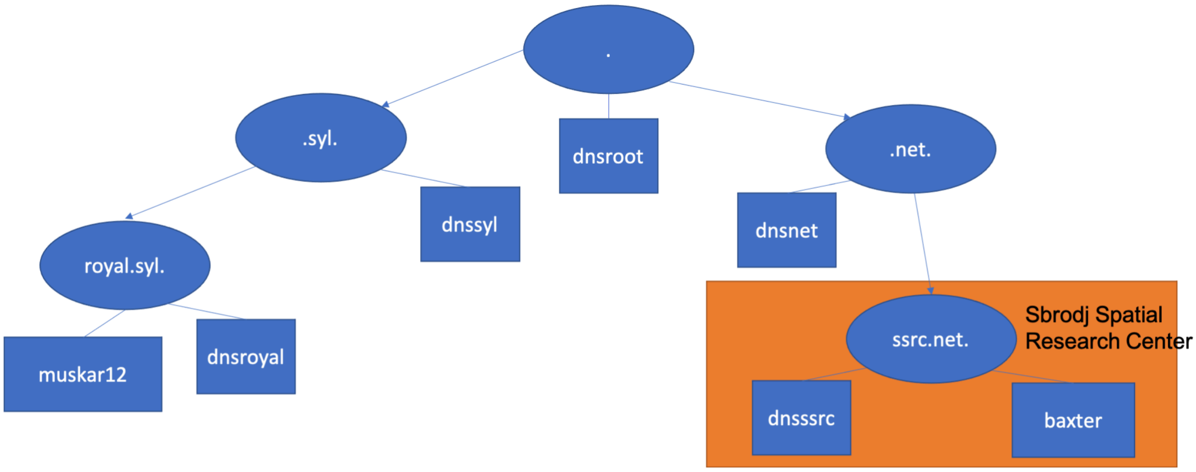
TP DNSSEC

Students:

Leonardo Araujo Fernandes
Gabriel Mauricio Molina Perez

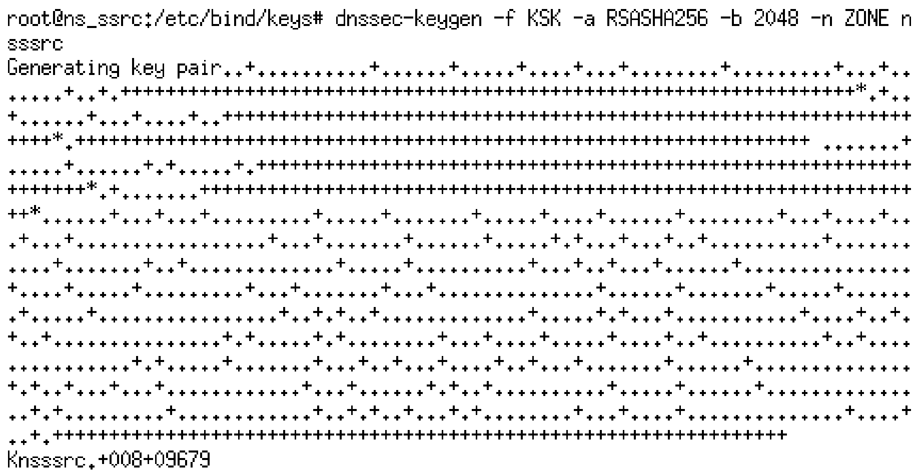


Objectif: Understand the DNS SEC architecture for securing the DNS infrastructure, its mechanisms and the consequences of this architecture.

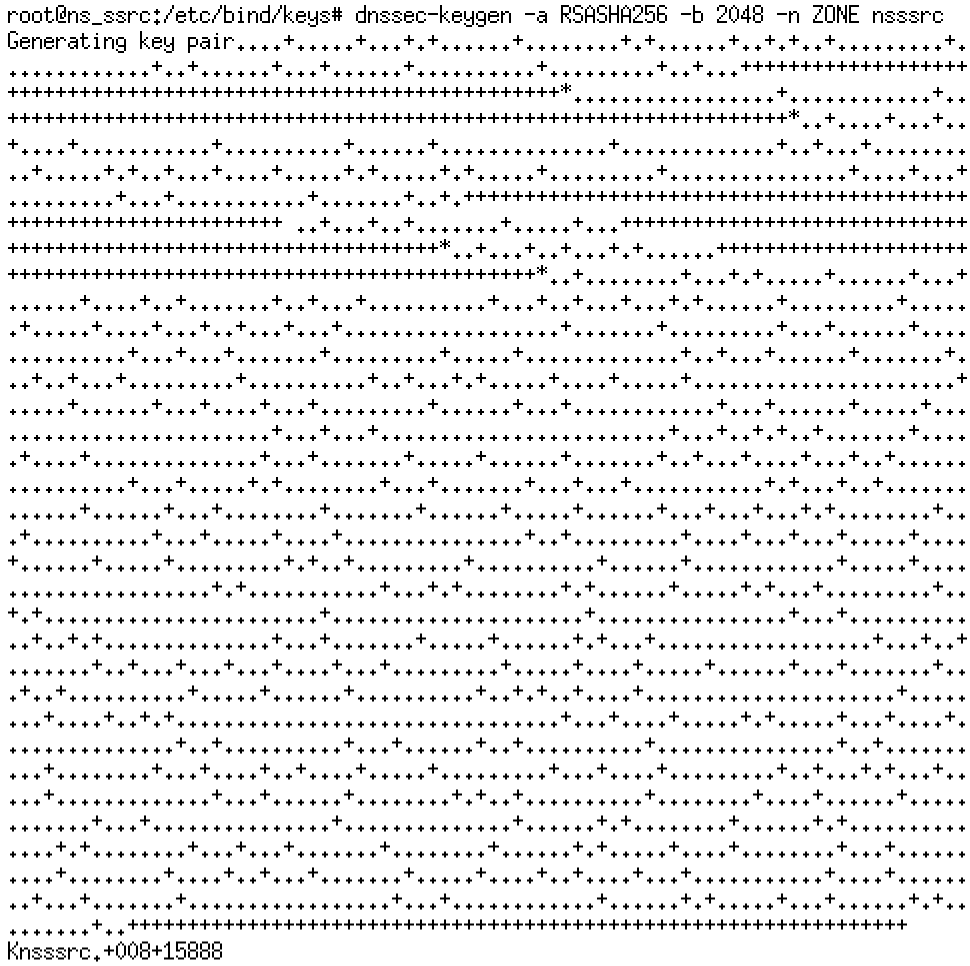
Describe the steps taken to get this DNSSEC PoC up and running

Environment: Kathara
Initial DNS server configuration: With the exception of nsssrc, all DNS servers in the lab are configured to support DNSSEC. This includes the generation of two types of key: the ZSK (Zone Signing Key) to sign DNS information and the KSK (Key Signing Key) to authenticate DNSSEC keys.

The KSK is the trusted root key that authenticates ZSK keys and guarantees the integrity of DNS records in a zone, in this case, *ssrc.net*, and is used to digitally sign the ZSK public key, creating a ZSK signature.
Command: dnssec-keygen -f KSK -a RSASHA256 -b 2048 -n ZONE *ssrc.net*



On the other hand, the ZSK is used to digitally sign DNS records in a specific zone, in this case, *ssrc.net*, such as A, CNAME, NS, SOA, etc. records that we have configured in the last TP. The ZSK signs all records in the zone, creating DNSSEC signatures for each record.
Command: dnssec-keygen -a RSASHA256 -b 2048 -n ZONE *ssrc.net*



Import keys into zone. Edit the file containing the ssrc.net zone (a priori "db.net.ssrc") with the necessary information (copy/paste from the previous TP) and also add the keys at the end with the following syntax.

So, we import the generated keys into the DNS zone by editing the zone file db.net.ssrc and adding the $INCLUDE lines for the KSK and ZSK keys.

$INCLUDE "/etc/bind/keys/Kssrc.net.+008+01371.key" ;

$INCLUDE "/etc/bind/keys/Kssrc.net.+008+07177.key" ;

```
$TTL    60000
@               IN      SOA     nsssrc.ssrc.net.
root.nsssrc.ssrc.net. (
```

```
                         2006031201 ; serial
                         28 ; refresh
                         14 ; retry
                         3600000 ; expire
                         0 ; negative cache ttl
                         )
@               IN      NS      nsroyal.royal.syl.
nsssrc         IN      A       192.168.0.22
baxter            IN       A      192.168.0.222
local           IN      A       192.168.0.110
ns              IN      CNAME   nsssrc

$INCLUDE "/etc/bind/keys/Kssrc.net.+008+01371.key" ;
$INCLUDE "/etc/bind/keys/Kssrc.net.+008+07177.key" ;
```

Including the KSK and ZSK keys in the DNS zone's records, along with sharing DS records with the parent zone, is essential for DNSSEC to function correctly and provide the intended security benefits, including *data authenticity* and *integrity validation*.

Then, we edit the named.conf configuration file to specify ssrc.net zone configuration, including key-directory definition.

```
zone "ssrc.net" IN {
    type master;
    file "/etc/bind/db.net.ssrc";
    inline-signing yes;
    auto-dnssec maintain;
    serial-update-method unixtime;
    key-directory "/etc/bind/keys";
};
```

Restart the BIND DNS server with

/etc/init.d/bind restart

It’s must then be restarted to take account of its new configuration

```
root@ns_ssrc:/# /etc/init.d/bind restart
Stopping domain name service...: namedwaiting for pid 39 to die
.
Starting domain name service...: named.
root@ns_ssrc:/#
```

```
root@ns_ssrc:/etc/bind# dig -4 +do baxter.ssrc.net

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> -4 +do baxter.ssrc.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55998
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
; COOKIE: 71413aa874ecf8690100000065a538f7857d38d59d8882bf (good)
;; QUESTION SECTION:
;baxter.ssrc.net.               IN      A

;; ANSWER SECTION:
baxter.ssrc.net.        60000   IN      A       192.168.0.222
baxter.ssrc.net.        60000   IN      RRSIG   A 8 3 60000 20240201075132 20240
115125135 1371 ssrc.net. Ic94Bq/nuRGLqzsCt+GwwZghYTZRIsZkl2fokWe26BIwBO8eAlpKuHv
Z JfXGl/dWsXMGZ2kq0A0blg/tVb/RLtiHnPfEGdACEcoMVweluHTH6wxe gWXgRkCcusB8dirUyiV2y
te3TA888pXxfmTWLjiiV/hZHZyYpa94pPYJ 1b8RsKSAG+CzrTpxXMX9wYgXxbaAf082/ZVZTcT7gqpQ
SQ6JIiF6qxoo CfLF5sPqzc+wRsCW2ZWZz/u/jqebtqWqZiKjWeYwq8TRhOu0iNJgq/9M uSyYfsomZV
ZB2jLE+btKxvuETrdOZ8WwbERYSEcVW7uLPKzwE+DLRcZR aNfzoA==

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Mon Jan 15 13:53:59 UTC 2024
;; MSG SIZE  rcvd: 384
```

The presence of the RRSIG record indicates that DNSSEC is in use for this query response. DNSSEC adds an additional layer of security by digitally signing DNS records to ensure their authenticity and integrity. The RRSIG record is used to verify the authenticity of the A record response, preventing DNS spoofing and tampering with DNS data.

Close chain of trust: nsnet configuration

In this step, we start creating a DS file using the dnssec-dsfromkey command to extract the DS corresponding to the KSK key:

```
root@ns_ssrc:/etc/bind/keys# dnssec-dsfromkey Kssrc.net.+008+07177.key
ssrc.net. IN DS 7177 8 2 7901B30B3954D9F23BBA3124125A06A057BF05DC20C7ED6BDFFA342
1FF78E61E
```

It's important to verify that is the 'Key-signing key',

```
root@ns_ssrc:/etc/bind/keys# cat Kssrc.net.+008+07177.key
; This is a key-signing key, keyid 7177, for ssrc.net.
; Created: 20240115124600 (Mon Jan 15 07:46:00 2024)
; Publish: 20240115124600 (Mon Jan 15 07:46:00 2024)
; Activate: 20240115124600 (Mon Jan 15 07:46:00 2024)
```

Rechargement de la configuration sur le serveur nsnet avec la commande rndc reload ou /etc/init.d/bind reload.

Perform a "delv" check

"Fully validated" means that DNSSEC validation has been successfully completed for all stages of the chain of trust, guaranteeing that the DNS response you have obtained has not been tampered with en route, and that it comes from an authorized and legitimate source.

This is an essential security mechanism for avoiding cache poisoning attacks and ensuring the integrity of DNS responses.

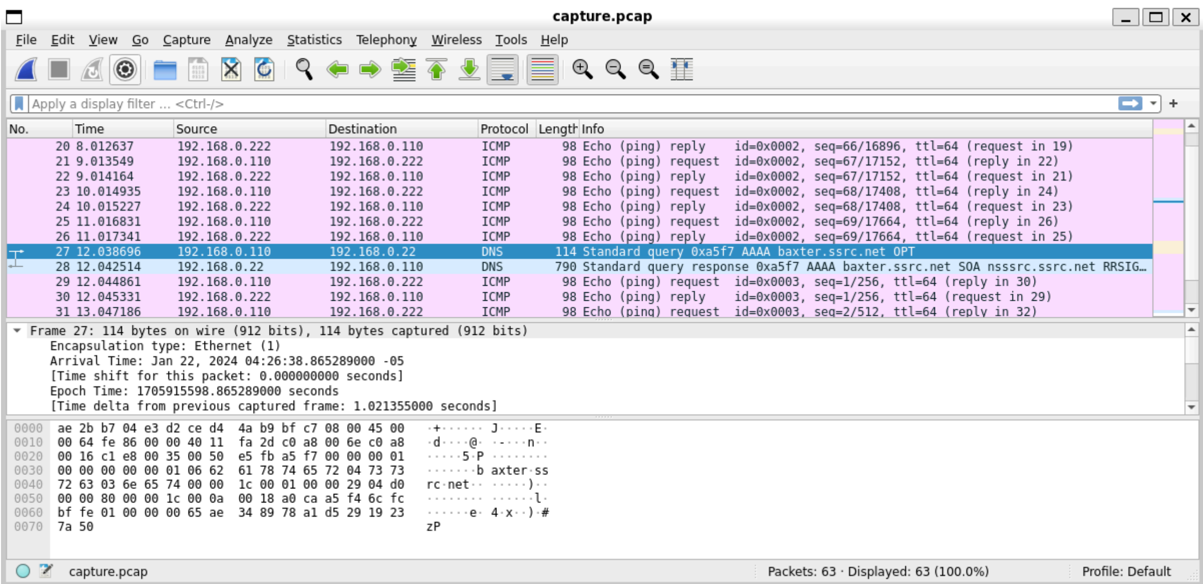
```
root@local_royal:/# delv -4 -t A baxter.ssrc.net +rtrace +root
;; fetch: baxter.ssrc.net/A
;; fetch: ssrc.net/DNSKEY
;; fetch: ssrc.net/DS
;; fetch: net/DNSKEY
;; fetch: net/DS
;; fetch: ./DNSKEY
; fully validated
baxter.ssrc.net.        60000   IN      A       192.168.0.222
baxter.ssrc.net.        60000   IN      RRSIG   A 8 3 60000 20240202223518 20240
120093635 1371 ssrc.net. OUrdhyTTv47WS9DkidGcEadaiOWfHR67hlnE59NrORKjVtqRnJCXRAs
t TLvjO78wiZPpBXIp4Wo8zz4AfJjkCIwl48+qw2HyqAqssXgRwnR93cbz RpcswsCeHNwrEn+B4C7bF
1Aor1e1AM27sFsazRfXtc1zB2tq1jJR0boO U1OSOv21VlyxXBFGWm7kPJdbENDLRSSf+cxd05Z4mc6I
7qYNJXlWgPYm 7GO+NOz6jOyOMqfIfgpGEnTOvQvRqsGyGZNIl9yk4SYA6LXMrA+AtVzs kAOLyk1tCJ
O+zn721NaD8ZfhZYWxlY6TGNm3H2cYd7RIm9SgqmT3mjDL qWSSPA==
root@local_royal:/#
```

We successfully reached the response, so we can affirm that the DNSSEC validation has been implemented.

Analysis with Wireshark

In the *local_royal* machine we execute a ping to the machine *ns_ssrc,* and in this last one, we're going to use a save command for tcpdump, which will allow us to analyze the capture using Wireshark.

tcpdump -i eth0 -w /shared/capturepcc.pcap &



Frame Number: 28, indicating this is the 28th frame captured in the sequence.
Frame Length: 790 bytes (or 6320 bits), both for the data on the wire and the captured data, suggesting no data was truncated during the capture.
Source IP Address: 192.168.0.22, in this case, *nssssrc.*
Destination IP Address: 192.168.0.110, in this case, *local_royal*.

Before going through the DNSSEC information, we will take a look into the DNS Response:

Transaction ID: 0xa5f7, used to match responses with requests.
Flags: Indicate a standard query response with no errors.
Questions: 1, asking for the IPv6 address (AAAA record) of baxter.ssrc.net.
Answers: 0, indicating no direct answer was provided in this response.
Authority RRs: 4, providing authoritative data about ssrc.net, including SOA and DNSSEC information.
Additional RRs: 1, likely for extended DNS features such as security.

Going into details, we have the Authoritative Nameservers and DNSSEC (DNS Security Extensions):

SOA (Start of Authority) record for ssrc.net, detailing the primary nameserver and various timing settings.
RRSIG (Resource Record Signature) records for DNSSEC, providing cryptographic signatures for verifying the response's integrity and authenticity, using the algorithm we define at the beginning *RSA/SHA-256.*

```
▾ ssrc.net: type RRSIG, class IN
      Name: ssrc.net
      Type: RRSIG (Resource Record Signature) (46)
      Class: IN (0x0001)
      Time to live: 0 (0 seconds)
      Data length: 284
      Type Covered: SOA (Start Of a zone of Authority) (6)
      Algorithm: RSA/SHA-256 (8)
      Labels: 2
      Original TTL: 60000 (16 hours, 40 minutes)
      Signature Expiration: Feb 21, 2024 04:14:55.000000000 -05
      Signature Inception: Jan 22, 2024 03:14:55.000000000 -05
      Key Tag: 1371
```

Finally, and important part of the packet we could analyze is the OTP record for EDNS0 which includes:

- A larger UDP payload size (1232 bytes).
- DNSSEC OK bit set (DO bit), indicating the server supports DNSSEC.
- A COOKIE option for additional security between client and server.

L'impact de l'utilisation de DNSSEC

Response size: DNSSEC responses are substantially larger than traditional DNS responses due to the addition of digital signatures and other security-related records (such as RRSIG, DNSKEY, and NSEC/NSEC3). This can increase DNS traffic, especially for the first queries that require the transmission of this additional information.
Without DNSSEC:

1	0.000000	192.168.0.110	192.168.0.22	ICMP	98 Echo (ping) request id=0x0007, seq=1/256, ttl=64 (reply in 2)
2	0.000544	192.168.0.22	192.168.0.110	ICMP	98 Echo (ping) reply id=0x0007, seq=1/256, ttl=64 (request in 1)
3	1.000889	192.168.0.110	192.168.0.22	ICMP	98 Echo (ping) request id=0x0007, seq=2/512, ttl=64 (reply in 4)
4	1.000936	192.168.0.22	192.168.0.110	ICMP	98 Echo (ping) reply id=0x0007, seq=2/512, ttl=64 (request in 3)
5	2.002477	192.168.0.110	192.168.0.22	ICMP	98 Echo (ping) request id=0x0007, seq=3/768, ttl=64 (reply in 6)
6	2.002526	192.168.0.22	192.168.0.110	ICMP	98 Echo (ping) reply id=0x0007, seq=3/768, ttl=64 (request in 5)
7	3.004095	192.168.0.110	192.168.0.22	ICMP	98 Echo (ping) request id=0x0007, seq=4/1024, ttl=64 (reply in 8)
8	3.004143	192.168.0.22	192.168.0.110	ICMP	98 Echo (ping) reply id=0x0007, seq=4/1024, ttl=64 (request in 7)
9	4.005169	192.168.0.110	192.168.0.22	ICMP	98 Echo (ping) request id=0x0007, seq=5/1280, ttl=64 (reply in 10)
10	4.005199	192.168.0.22	192.168.0.110	ICMP	98 Echo (ping) reply id=0x0007, seq=5/1280, ttl=64 (request in 9)
11	5.052150	be:cc:e9:7f:1e:a7	7e:0c:9f:be:e8:bb	ARP	42 Who has 192.168.0.110? Tell 192.168.0.22

The replies took less time to arrive without DNSSEC, between 1-10s compared to the 20-30 seconds that took with DNSSEC.
DNSSEC offers significant security benefits, but its deployment and use must be carefully planned to manage its impact on network traffic and server load.