Thomas Bakaysa
DNSSEC lab

Task 1 : Set Up the example.edu Domain

Task 1a : Generate keys

Creating the ZSK and the KSK for example.edu

```
[03/27/25]seed@VM:~/.../edu.example$ ll
total 24
-rw-rw-r-- 1 seed seed  488 Jul 19  2022 example.edu.db
-rw-r--r-- 1 seed seed  431 Mar 27 15:09 Kexample.edu.+008+15358.key
-rw------- 1 seed seed 1012 Mar 27 15:09 Kexample.edu.+008+15358.private
-rw-r--r-- 1 seed seed  605 Mar 27 15:09 Kexample.edu.+008+33034.key
-rw------- 1 seed seed 1776 Mar 27 15:09 Kexample.edu.+008+33034.private
-rw-rw-r-- 1 seed seed   86 Jul 18  2022 named.conf.seedlabs
```

Created a signed zone file using the example.edu.db nameserver

```
[03/27/25]seed@VM:~/.../edu.example$ dnssec-signzone -S -o example
.edu example.edu.db
Fetching example.edu/RSASHA256/15358 (ZSK) from key repository.
Fetching example.edu/RSASHA256/33034 (KSK) from key repository.
Verifying the zone using the following algorithms:
- RSASHA256
Zone fully signed:
Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                      ZSKs: 1 active, 0 stand-by, 0 revoked
example.edu.db.signed
[03/27/25]seed@VM:~/.../edu.example$ ll
total 36
-rw-rw-r-- 1 seed seed   96 Mar 27 15:11 dsset-example.edu.
-rw-rw-r-- 1 seed seed  488 Jul 19  2022 example.edu.db
-rw-rw-r-- 1 seed seed 5790 Mar 27 15:11 example.edu.db.signed
```

Task 1b : Sign example.edu domain's zone file

The original zone file simple contained mappings for hostnames to ip addresses.

```
; Records for this nameserver (you need to make changes)
@               IN   NS    ns.example.edu.
ns.example.edu. IN   A     10.9.0.65

; IP addresses for the hostnames in the example.du domain
@       IN      A     1.2.3.5
www     IN      A     1.2.3.5
xyz     IN      A     1.2.3.6
*       IN      A     1.2.3.7
```

After signing the zone file, we now have DNSkeys and the RRSIGs made with said keys. This can be used to ensure that the data has not been tampered with. Below is the beginning of the signed DNS zone file.

```
; File written on Thu Mar 27 15:11:32 2025
; dnssec_signzone version 9.18.30-0ubuntu0.20.04.2-Ubuntu
example.edu.            259200  IN SOA  ns.example.edu. admin.example.edu. (
                                        2008111001 ; serial
                                        28800      ; refresh (8 hours)
                                        7200       ; retry (2 hours)
                                        2419200    ; expire (4 weeks)
                                        86400      ; minimum (1 day)
                                        )
                        259200  RRSIG   SOA 8 2 259200 (
                                        20250426181132 20250327181132 15358 example.edu.
                                        FM5sgt2n5S1osOGlNtR/AGI9QDvrMI47o6ER
                                        LMtyTzs00D/mkLbnBukrCDH91y7Q9tcLJb7R
                                        dhw39jEqhqTJtxAJQi37/M/PqK00JMoDvrt1
                                        zPrRvaAmPWFIKh/Mwhmg2sqBH1bVJ7TxMgCZ
                                        JqIZKXucr2JZ4T8N4x2Yei1+76U= )
                        259200  NS      ns.example.edu.
                        259200  RRSIG   NS 8 2 259200 (
                                        20250426181132 20250327181132 15358 example.edu.
                                        WMOEXSyub/+RzK4EEIfKSv5aNS9AEzjqqqBY
                                        9EnfeFQ/w/C3S5OvTtSNf3DIiQ5E9buJyWtg
                                        DOEXfdmaX0BF5Z2ePsHqo8z6X7DBPwqNN5kR
                                        rAHJ+/0M3VW5N5ChDTT7ggEM9HcqTRI4UuIm
                                        kGvotO89G3+SV7GfJrTkLCKgwBo= )
                        259200  A       1.2.3.5
                        259200  RRSIG   A 8 2 259200 (
```

DNS config file modified to use the newly signed zone file

```
[03/27/25]seed@VM:~/.../edu.example$ cat named.conf.seedlabs
zone "example.edu" {
        type master;
        file "/etc/bind/example.edu.db.signed";
};
```

Task 1c : Testing

Looking for the public DNS keys and RRSIG for the various servers in our containers. I outputted the results into files for easier access and readability.

```
/DNSSEC-lab$ dig @10.9.0.65 example.edu DNSKEY +dnssec | cat > example.DNS.results
```

$ dig @10.9.0.65 example.edu DNSKEY +dnssec

For the command [dig @10.9.0.65 example.edu DNSKEY +dnssec] it brought up the ZSK (zone signing key) and the KSK (key signing key) used by DNSSEC to verify the integrity of the DNSSEC records.

```
;; ANSWER SECTION:
example.edu.            259200  IN      DNSKEY256 3 8 AwEAAcTnFPcedTPoq72
Ily8ASzpS5bgwNZMSiupYGOkdODKQRXpUwRM2 q30b+7eLT4OdGW1Hv3EMJkh9IhU1oOgnS1Q
LJZz/wIT5RWkkrecBHG1q aKJidT6GQ5e5rf7DnQFVOYMh6K07ScPbbnBEsIwQuU/1XI/HMmY
A8xNC FiWBZegl
example.edu.            259200  IN      DNSKEY257 3 8 AwEAAaDXuXB5YhEBh42
zxpVdQLjagAFvkPXZuH9pqZwtotq91E2UXXBf z99TuldhTnKSAGxPAVCckUeyALf60txiQz8
ELHNobEqX9GBNRp6uTP8b +670Z1/sOT0tWQWlUAHH/LC3JYjYc2j42FWgP8mjg4RMcRlxtJp
ZE1qJ cgCWvHou/pGq01E4Hj097W5OTdXnRdKQygQsWyzAkhN0IaGzKHhz0DzR OR8U43RXdr
eDracl1S9AjmatAumTfDR4sPHR280c+9kx9rPY8CA1NmdK hIstvoOh3bv6Sllv0HtxHU6fM7
OK3z836aYEXw9z361DRbERpkU5iVb1 FdY1z8bBrTM=
```

Taking a closer look, the first entry has DNSKEY256, which is the zone signed key. The long string is the public key that can be used to authenticate entries. The second entry, DNSKEY257, is the key signing key.

```
example.edu.            259200  IN      RRSIG DNSKEY 8 2 259200 202504261
81132 20250327181132 15358 example.edu. V5EEFqW4AeOUHYdMLvDIkHfCvdbOPCzau
xIwBotv12nCdFzr0n1nbWaD IZ6aFDyrdjLFJ7yISHr2sB8emp0Hc8BVZAesh2vFZBbY3WjOd
ZAHsL2h DpZ/hNdIq35yTlJhibJDKoDpGDlciIlVj1Mir3O9v5QYuuplvqL5KRiB P0o=
example.edu.            259200  IN      RRSIG DNSKEY 8 2 259200 202504261
81132 20250327181132 33034 example.edu. XTPeegWnAxlurP8f3pnUddNghGGBonDMO
aG10+ecnT/1sA6xJhzo5rX0 obdI4PKp9swPz/JmQn5VwYIe9D3ERr3XysY6NdmzxD2UHfCzi
knC6DIT lAZjdqp4JxIwFEl1NKxBox1im7VP09P02f5lGJKVFRQnl9tHI/gI7nZ9 pZ3VSjZD
wqBjCb1Zmato3zwoyjLDtmDJP9f7ArLHk3pF7s5/oH4NNUtF 9U3rsTHBr7E7hZ+musTgC3dS
pVsz+nJvetfoAKYcRKtU19MgMfis33Fk HERVSm5Qmg8KGPWSdTybzcGwnLEk25idDPRgC8IV
Whvef3mJ98CCwvos nZBXsw==
```

This second part of the results show two RRSIGs, the signatures made from signing the two keys (known as an RRset). This specific record shows that we signed the RRset with both the ZSK and the KSK. However, in the RRSIGs it actually shows the Key ID used to sign. This Key ID is a hash of the DNSKEY itself. I didn't feel like doing that so it's a good thing the signed zoned file has the computed value already. If we take a look there and compare we find that the first RRSIG entry was signed with the ZSK key while the second was signed with the KSK. The purple is the RRSIG while the dark green is the DNSKey entry in the signed zone file.

```
81132 20250327181132 33034 example.edu. )
) ; KSK; alg = RSASHA256 ; key id = 33034
81132 20250327181132 15358 example.edu.
) ; ZSK; alg = RSASHA256 ; key id = 15358
```

$ dig @10.9.0.65 example.edu NS +dnssec

With this we get the DNSSEC record of the NS (name server). As shown below, this command returns the NS record and the A record (ip address) as an additional record. Since these are RRSIGs, that means they need to be signed by a DNSKEY, the format is the same as the earlier RRSIGs and so we can see that these signatures are signed using the ZSK (key 15358), which

makes sense. The ZSK is the weaker of the two keys and gets rotated out frequently for security purposes. The KSK is usually stronger and really only used to authenticate the ZSK. This is because PKI (public key infrastructure) is - relatively - computationally expensive.

```
;; ANSWER SECTION:
example.edu.            259200  IN      NS      ns.example.edu.
example.edu.            259200  IN      RRSIG   NS 8 2 259200 20250426181
132 20250327181132 15358 example.edu. WMOEXSyub/+RzK4EEIfKSv5aNS9AEzjqqqB
Y9EnfeFQ/w/C3S5OvTtSN f3DIiQ5E9buJyWtgDOEXfdmaX0BF5Z2ePsHqo8z6X7DBPwqNN5k
RrAHJ +/0M3VW5N5ChDTT7ggEM9HcqTRI4UuImkGvotO89G3+SV7GfJrTkLCKg wBo=

;; ADDITIONAL SECTION:
ns.example.edu.         259200  IN      A       10.9.0.65
ns.example.edu.         259200  IN      RRSIG   A 8 3 259200 202504261811
32 20250327181132 15358 example.edu. C4eSdwPyeTTVeKhmL6hxgF6VK7mOUaGRbzfJ
fUNQBLP3snqv0OozK/wT LDMYPrzXnbyT9pOxHQ5znzY4q5lAOq9WOizWR15THx8ZEX8bSPLA
eXHG mrscK3X8GTS0ntTrPJo70LWiuFLiGyyB+2+muw2jjHxRtwsDfrHmz6BA LYo=
```

$ dig @10.9.0.65 www.example.edu A +dnssec

This command retrieves the A record for this address. Again, the DNSSEC RRSIG issigned with the ZSK.

```
;; ANSWER SECTION:
www.example.edu.        259200  IN      A       1.2.3.5
www.example.edu.        259200  IN      RRSIG   A 8 3 259200 202504261811
32 20250327181132 15358 example.edu. NKMGmr1xAVONxyFUXgt9kNzoPQxyK1jhfHnb
p+qta1cbPtn2/Qxpwpyn pZkWNXcvY7oMQR9VrqCW4HLpUwNvytuqp6UaJ2MJ+99hEJMNHlAk
Qf5e bYU2G+X5qmlrLoQk2vNX6qhyq0VkJRKlcB30z4HBsa2ll8AhRhHO9BeK Gms=
```

By using only the ZSK to sign zone records, we minimize the exposure of the KSK, which is far more important than the ZSK. The private key pairs for both are these however, should always be stored in a secure location.

```
[04/09/25]seed@VM:~/.../edu.example$ ll
total 36
-rw-rw-r-- 1 seed seed   96 Mar 27 15:11 dsset-example.edu.
-rw-rw-r-- 1 seed seed  488 Jul 19  2022 example.edu.db
-rw-rw-r-- 1 seed seed 5790 Mar 27 15:11 example.edu.db.signed
-rw-r--r-- 1 seed seed  431 Mar 27 15:09 Kexample.edu.+008+15358.key
-rw------- 1 seed seed 1012 Mar 27 15:09 Kexample.edu.+008+15358.private
-rw-r--r-- 1 seed seed  605 Mar 27 15:09 Kexample.edu.+008+33034.key
-rw------- 1 seed seed 1776 Mar 27 15:09 Kexample.edu.+008+33034.private
-rw-rw-r-- 1 seed seed   93 Mar 27 15:18 named.conf.seedlabs
```

Currently the key pairs live in the same directory as the signed zone file.

Task 2 : Set up the edu server

Task 2a : Find and add the DS record

Retrieving the DS record, which is stored in the same directory as the signed zone file.

```
[04/09/25]seed@VM:~/.../edu.example$ cat dsset-example.edu.
example.edu.            IN DS 33034 8 2 BACEF2F6890E2FF5C4FB8C
```

KeyID 33034 indicates that this DS (Delegation Signer) was itself signed by the KSK. The truncated string at the end is the signature to verify the record.

Task 2b : Set up the edu server

This task wants me to add the DS record to the signed zone file. Because this modifies the content of the zone's files, re-singing the zone file is required to maintain validity and functionality..
I did use the [dnssec-dsfromkey] command to see if the current DS record was made correctly. Which they did, as can be seen here.

```
[04/09/25]seed@VM:~/.../edu.example$ cat dsset-example.edu.
example.edu.            IN DS 33034 8 2 BACEF2F6890E2FF5C4FB8C724D42BB9BFF6708A7CE8CEBD60175
60E8 E649CCD3
[04/09/25]seed@VM:~/.../edu.example$ dnssec-dsfromkey Kexample.edu.+008+33034.key
example.edu. IN DS 33034 8 2 BACEF2F6890E2FF5C4FB8C724D42BB9BFF6708A7CE8CEBD6017560E8E649CCD
3
```

I then attached it to the signed zone file. By using the $INCLUDE line, we don't have to worry about copying the actual values. Especially good since a robust DNSSEC policy requires keys to be rotated periodically.
The $INCLUDE must be at the bottom of the file so as to not confuse the RRset for this zone file.

```
[04/09/25]seed@VM:~/.../edu$ cat edu.db
$TTL 3600
@       IN      SOA     ns1.edu. admin.ns1.edu. (
example.edu.            IN      NS      ns.example.edu.
ns.example.edu.         IN      A       10.9.0.65

$INCLUDE ../edu.example/dsset-example.edu.
```

Creating the keys for [edu.db]. At first I had named the keys wrong, using edu.db instead of edu.

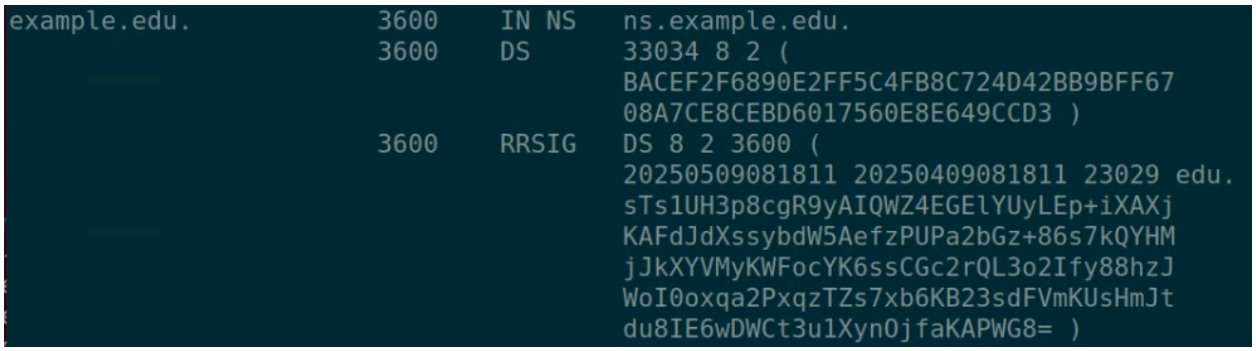
```
-rw-r--r-- 1 seed seed  589 Apr  9 05:17 Kedu.+008+12526.key
-rw------- 1 seed seed 1776 Apr  9 05:17 Kedu.+008+12526.private
-rw-r--r-- 1 seed seed  415 Apr  9 05:17 Kedu.+008+23029.key
-rw------- 1 seed seed 1012 Apr  9 05:17 Kedu.+008+23029.private
```

```
[04/09/25]seed@VM:~/.../edu$ dnssec-signzone -S -o edu edu.db
Fetching edu/RSASHA256/12526 (KSK) from key repository.
Fetching edu/RSASHA256/23029 (ZSK) from key repository.
```

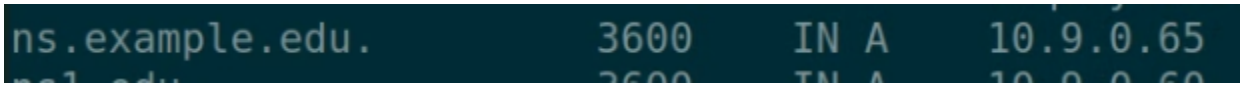
A quick look at the newly signed zone file.

Specifically, at the new zones KSK and ZSK. They differ from the ZSK and KSK used in the example.edu zone.

Scrolling down the file, we can also see the example.edu DS record we included. The key used to sign it (33034), which is the example.edu's KSK and an RRSIG signed with this (edu) zone's ZSK.



An important link is between the example.edu NS record which points to the ns.example.edu A record. This properly routes traffic heading for example.edu to the correct zone. The record above is paired with the record below in order to facilitate this.



Task 2c : Testing

This task took a while to debug, for some reason the named.config.seedlabs - which I modified to point to the edu.db.signed file so that DNSSEC has access to the signed zone file - was not loading into the container properly. After a bunch of dcbuilds and a few restarts of the VM I was able to get it to work. I verified that the file loaded properly by sh'ing into the container and manually checking the /etc/bind/named.config.seedlabs file.

$ dig @10.9.0.60 edu DNSKEY +dnssec

```
edu.                    3600    IN      DNSKEY  256 3
8 AwEAAdcsFzBiAMLZ5paDMDQ1H6I09Ui42jflGaVs07a7RH42r5Ei
WMPf TC1iuY8x0oBec1KPMaFGsYXf8ilLHBP/yFDnwLwfHLX1X/BnJ
67mK5pr ptAKffPUzlXOS4zILMQGI9oUYGcOZLFsIMExnIj8BbKW7q
LSgpVOdUBF HD8O+oGP
edu.                    3600    IN      DNSKEY  257 3
8 AwEAAdN4SKe16pjkBAwBj1uQMx5jU0GuyHjbeWrfPdtPqgMw2t4D
/fvh x61QQ5SZ2RYU/kEnYd3hc0a/NUi1U7EZJ9LF7fJyGBTwW4W5o
/N71r1o 3Z3xCGTUO1zqrFukKRIATRx493sxvNuwRQ2yr/ARTCnPms
3qQvTn1FAF k+TMDNpkFiER6ZS1OxwnHol2V2RJVa4A1MktwlBarJC
iq9/Z3tb9rHwc urxuLNDDHAywFnQ0UJUXblz9fvGsaLb7/28oyiS4
+NIJZkbekkYmQV9I fKivS7fYzOVubQrULYCi722LfH+GG3SpB7GTF
S0FPT6EjF1vL3QkO2t+ AGgA0jUqGvM=
```

```
edu.                    3600    IN      RRSIG   DNSKEY
 8 1 3600 20250509093702 20250409093702 12526 edu. Z/6
wZS20MjT8oWEMg8xtMhWVjZ1Nf5EoU+IhA65jVm9VKaufzRtMS1Fn
BZxQbMWc0a46ZTxBJkLystplMj9VW34lGCfz2ph1LKpKaWJre450fO
pO McWJUbtS1QgoKmeDtLVJhQkxF3di0ZlOg+tnQBcdp/xNOHS51/5
l4ANA vimyll8n/K5uoCNd0eHJf93d+0KUJokz4iNF/AH2pVVgHXUw
k+M+DF0u y/hZGKKu1vRlm4nZnHCO5Z1lDKZnLk2fzWS6GIdSdlM6y
02M3/UdvvgU 3YlnR8AOVYBM85B358xdN3IO/NbLsg8Cme0wBjQUZL
PFtu8yjSQKj/VC NaPzXA==
edu.                    3600    IN      RRSIG   DNSKEY
 8 1 3600 20250509093702 20250409093702 23029 edu. X2q
+mSmicaT9kdILzAfXUKsOZ2Jn1ghvk1uo6E0IIG+9GagzPeZKD2VQ
PuzO5iTrGTrcTex7G1p742Tc3SaiFehrBOJGMIIGREvGfjxsQaorCL
2i VuQzGF3MqM9dpbw9m+ak08sOg7kEWl3FaeqExyIRCeVsSqLy1/q
fS/Pi aPo=
```

Shows us the public DNSKeys for the edu zone. DNSKEY 256 is edu's ZSK and DNSKEY is edu's KSK. The RRSIGs are the signed RRSets, the first record signed with the KSK (KeyID 12526) while the second record is signed with the ZSK (KeyID 23029)

$ dig @10.9.0.60 edu NS +dnssec

```
;; ANSWER SECTION:
edu.                    3600    IN      NS      ns1.edu.
edu.                    3600    IN      RRSIG   NS 8 1 3600 2025
0509093702 20250409093702 23029 edu. Y1rucxxEmk2g39H8vbEkf6NdU9Z
sOXEVAJnDmA9Oa0Y5GYt3J5XZk4Lw VgAtYOoHjkVJI3jVxw4VCOpmUY1WyXdRGq
pRYws+HHxBzjQKVMwD0cA1 8AaZbhsIwg+MXm/i09ELsv7Z/nlfzcu6qYbMp6/N1
JwdLsNn6jUnZ+x+ 11U=

;; ADDITIONAL SECTION:
ns1.edu.                3600    IN      A       10.9.0.60
ns1.edu.                3600    IN      RRSIG   A 8 2 3600 20250
509093702 20250409093702 23029 edu. bdo4FK6wuXU6H8rAbo1rlK/dpeZ8
mMrqlUEf0owiK718IbDGUi5UcIos 4DesHoQ99odXH8LizvznWR/C8PalkAlD+H2
+Lt4UcQCH2tZvQA33hpSJ A4fytbIsP1shOSjFKUV9wSDnR9YU6AQwQMBCJRZUNs
RvTjCMd1tus9YL bX8=
```

Retrieves the name server that this zone uses, which ns1.edu. The additional sections provides the A record for the name server.Both have an attached RRSIG signed using the edu zone's ZSK. `23029 edu.`

$ dig @10.9.0.60 example.edu +dnssec

```
;; AUTHORITY SECTION:
example.edu.            3600    IN      NS      ns.example.edu.
example.edu.            3600    IN      DS      33034 8 2 BACEF2
F6890E2FF5C4FB8C724D42BB9BFF6708A7CE8CEBD6017560E8 E649CCD3
example.edu.            3600    IN      RRSIG   DS 8 2 3600 2025
0509093702 20250409093702 23029 edu. eLC5Oae6SDW8XupiUdnvqs0199h
ZbMQ8OLInledFDcbHguues7RvWtV4 BS1sFHtwqDJFXIIYNCKyGTJCdy+0Go+zPh
riE7Hl89hGde4u4L2ERF5X epSTeLUeerQ/DpALM47aHDMJ5+SSXevn90WY94k0h
ZZpFYJZNHAe5ezv 524=

;; ADDITIONAL SECTION:
ns.example.edu.         3600    IN      A       10.9.0.65
```

This shows that this zone is directing traffic heading to example.edu to the name server ns.example.edu, the top being the NS record for this. Followed by the DSset, the child (example.edu) zone's KSK KeyID and a signature to verify its authenticity. This is how a parent zone provides authentication for its child zones. We then have an RRSIG of the DS record to then authenticate that too using the parent (edu) zone's ZSK.

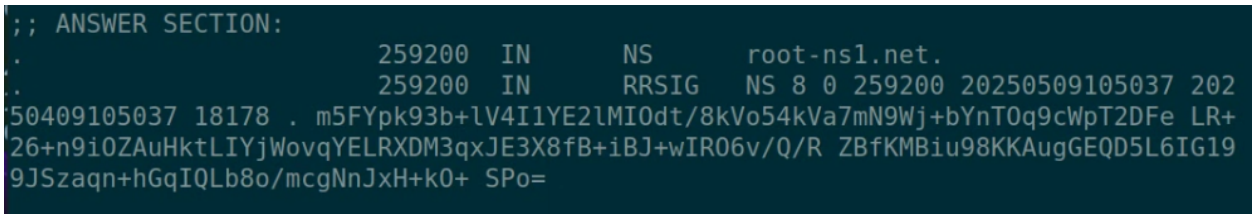
Task 3 : Set up the root server

Alright, so the root's ZSK ID is 18178 while its KSK ID is 59155, which I found in the signed zone file.

$ dig @10.9.0.30 . DNSKEY +dnssec

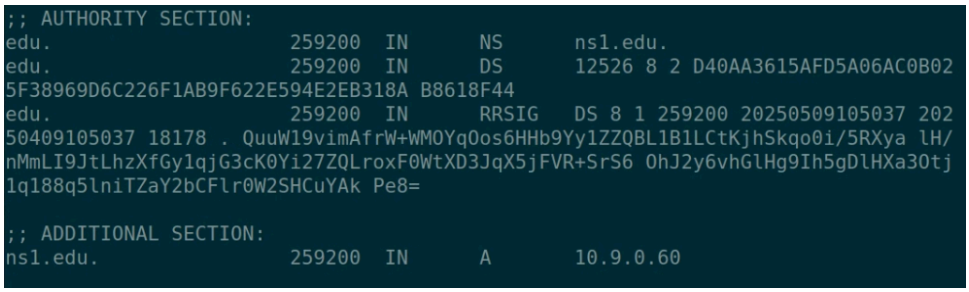
Same as the last time, this retrieves the roots KSK and ZSK public keys including the RRSIGs of the RRset.

$ dig @10.9.0.30 . NS +dnssec



Different from the others, this record is pointing to a name server that does not live in root's zone. This is the reason why we don't have an attached A record pointing to the address of the NS, since this name server lives elsewhere and we don't blindly send out A records outside our zone.
If someone needed to resolve an address in any of our zones, a higher level DNS server would have a record with the NS of our root and an A record pointing to it, since our server would live inside their zone.

$ dig @10.9.0.30 edu +dnssec



Here we see that we have an entry pointing to name server for edu's zone. It includes the DSset to verify the child's KSK and a RRSIG of the DS record signed with root's ZSK for

authentication. Included is the additional A record pointing to the actual IP address of edu's name server, since this lives within root's zone we have the ability to verify that nothing has been tampered with.

$ dig @10.9.0.30 example.edu +dnssec

```
;example.edu.                   IN      A

;; AUTHORITY SECTION:
edu.                    259200  IN      NS      ns1.edu.

;; ADDITIONAL SECTION:
ns1.edu.                259200  IN      A       10.9.0.60
```

Here we see that the command actually returned the NS and A record for edu instead of example.edu. This is because we asked the root zone, which typically does not allow recursion.

```
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 2
;; WARNING: recursion requested but not available
```

So, because root doesn't know where example.edu lives, but knows where edu does, it returns that and nothing more. In the real world this is done because root servers are incredibly important and shouldn't be looking into every DNS request, it returns the authoritative zone for the next zone down on the hierarchy and has the requester sort the rest.

Task 4 : Set up the local DNS server

Failure after failure kept following around on this task. The dig command was giving everything correct, the records were as they were supposed to be and it could traverse the zones without problem. Except, when using the DNS local server it just wouldn't set the ad flag.
The solution I found, after redoing every key and every signed zone file at least twice, was that the named.conf in the local_dns_server directory needed the line [include "/etc/bind/bind.keys";] to actually use the root's KSK as a trusted authority. After that, I got the ad flag.

```
[04/09/25]seed@VM:~/.../local_dns_server$ dig @10.9.0.53 www.example.edu +dnssec

; <<>> DiG 9.18.30-0ubuntu0.20.04.2-Ubuntu <<>> @10.9.0.53 www.example.edu +dnssec
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5438
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: 0d6dc6f6e897847e0100000067f6a8c80c3dbbb065280f52 (good)
;; QUESTION SECTION:
;www.example.edu.               IN      A

;; ANSWER SECTION:
www.example.edu.        259200  IN      A       1.2.3.5
www.example.edu.        259200  IN      RRSIG   A 8 3 259200 20250509133856 202504091
```

```
IN NS    ns.thomas2025.edu.
DS       7399 8 2 (
         304A756269DD5D69B5
         E6F6C6805B6B36CB01
RRSIG    DS 8 2 3600 (
         20250509170238 202
         HvpTTlg4LSIvCZGynn
         pkqgQ2V8FVmgKJzLz6
         9Xt2qvKXau33c64OhsY
         vNRyFGBBum+dqiPjU1
```

```
IN NS    ns.thomas2025.edu.
DS       7399 8 2 (
         304A756269DD5D69B5
         E6F6C6805B6B36CB01
RRSIG    DS 8 2 3600 (
         20250509170238 202
         HvpTTlg4LSIvCZGynn
         pkqgQ2V8FamgKJzLz6
         9Xt2qvKXau33c64OhsY
         vNRyFGBBum+dqiPjU1
```

To test if DNSSEC catches inconsistencies, I modified a single Byte in the RRSIG for my site's DS entry in the edu signed zone file.
This causes the dig command to fail, since DNSSEC doesn't trust the information store due to the altered RRSIG for thomas2025's DS record.
While accessing edu's DS record can still be done.

```
; <<>> DiG 9.18.30-0ubuntu0.20.04.2-Ubuntu <<>> @10.9.0.53 www.thomas202
sec
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 60378
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
```

```
; <<>> DiG 9.18.30-0ubuntu0.20.04.2-Ubuntu <<>> @10.9.0.53 edu DS +c
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32280
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL
```

Task 5 : Adding another domain nameserver

```
25$ ll

4:02 dsset-thomas2025.edu.
3:26 Kthomas2025.edu.+008+07399.key
3:26 Kthomas2025.edu.+008+07399.private
3:25 Kthomas2025.edu.+008+56709.key
3:25 Kthomas2025.edu.+008+56709.private
4:00 named.conf.seedlabs
3:40 thomas2025.edu.db
4:02 thomas2025.edu.db.signed
```

This task was relatively simple considering I already had an example.edu to work with.
I generated the requisite keys and modified the db file to reflect this new zone.
Below is the db file, modified for this particular zone.

```
[04/09/25]seed@VM:~/.../edu.thomas2025$ cat thomas2025.edu.db
$TTL 3D
@       IN      SOA   ns.thomas2025.edu. admin.thomas2025.edu. (
                      2008111001
                      8H
                      2H
                      4W
                      1D)

; Records for this nameserver (you need to make changes)
@                 IN   NS    ns.thomas2025.edu.
ns.thomas2025.edu.  IN   A    10.9.0.66


; IP addresses for the hostnames in the thomas2025.edu domain
@       IN      A     3.2.3.5
www     IN      A     3.2.3.5
*       IN      A     3.2.3.7
```

Big changes - aside from changing the record names to reflect this zone - was pointing the A record to the actual IP address this zone will live on. This container that will implement this zone can be found at the root lab folder, in the docker compose file.
Another important modification was changing the entry and the zone name in the named.conf.seedlabs file. Zone “thomas2025.edu” lets the DNS server know that this db file is good for the thomas2025.edu zone. I was running into troubles because it was still smith20something, even though I had changed the file path.

```
[04/09/25]seed@VM:~/.../edu.thomas2025$ cat named.conf.seedlabs
zone "thomas2025.edu" {
        type master;
        file "/etc/bind/thomas2025.edu.db.signed";
};
```

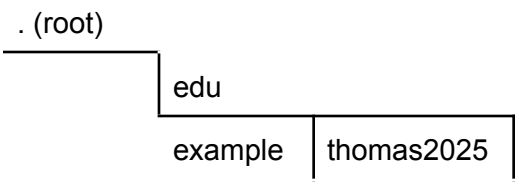
I then created the signed zone file, which also created the dsset file. I included that path in the edu db file while also including thomas2025’s A and NS record to actually allow the zone to be found.

```
; Records for the example.edu domain
example.edu.            IN      NS   ns.example.edu.
thomas2025.edu.         IN      NS   ns.thomas2025.edu.
ns.example.edu.         IN      A    10.9.0.65
ns.thomas2025.edu.      IN      A    10.9.0.66
```

Both the NS and A record are important. At the bottom I also appended the INCLUDE for thomas2025 zone’s dsset so that these records can be verified.

```
$INCLUDE ../edu.example/dsset-example.edu.
$INCLUDE ../edu.thomas2025/dsset-thomas2025.edu.
```

I then signed the edu zone’s file, since it was changed and the new signed file needs to reflect the added zone. This changes the RRSIG in the signed file and the dsset file as well, which means we have to goto the root file and sign that one as well.
It is important that we sign the zones from the lowest node on the hierarchy first, and then move up from there. This is because the upper zones incorporate the information of the zones right below them in their signed zone files.
So for ours, a good way to visualize it would be like so.



So edu incorporates example and thomas2025 in its signed zone file, while root incorporates edu. Thus, you have to start signing from the bottom, else DNSSEC will catch the inconsistency.

```
[04/09/25]seed@VM:~/.../edu.thomas2025$ dig @10.9.0.53 www.thomas2025.edu +dnssec

; <<>> DiG 9.18.30-0ubuntu0.20.04.2-Ubuntu <<>> @10.9.0.53 www.thomas2025.edu +dnssec
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59420
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; COOKIE: f59b8e80013340c10100000067f6b6ad00a97170c9b0aecd (good)
;; QUESTION SECTION:
;www.thomas2025.edu.            IN      A

;; ANSWER SECTION:
www.thomas2025.edu.     259200  IN      A       3.2.3.5
www.thomas2025.edu.     259200  IN      RRSIG   A 8 3 259200 20250509170231 202504091
70231 56709 thomas2025.edu. AYI41lhIfU7q5UCTgC2Hf8/sV5QF0aM4qx31MRRm77DFPwv3GLAVz26u
N9qUpfvVw7b0CzM8TbhBdtBCjaMs4nZDPlKXCNIO80uEHb5dYL4UUBS4 1UAKSc7EJJ0l5agc+jtmsSbgReYp
Nowj8tJhf7cr0fsHRoZ/SzVDS9Rj LFG8
```

The ad flag is set, thus proving that my custom zone within the edu domain is considered trustworthy.