

UNIX LAB 6

IPTABLES

```
GNU nano 6.2 iptables.sh
#!/ bin / sh
#
# IPTABLES SCRIPT
# <Administration av UNIX-lika System > <HT21 DT149G Datateknik GR (B)> - ASSIGNMENT 6
# <Filip Stenegren>

# Creating a macro that specifies the location of iptables .
IPTABLES = /sbin/iptables
echo " Flushing existing tables and setting default policies "
$IPTABLES -F
$IPTABLES -P INPUT DROP
# Make sure that you replace iptables with $IPTABLES , this way you will use the macro defined above .
echo " Setting up INPUT chains "

# Drop all incoming packets
$IPTABLES -P INPUT DROP

# Allow all traffic to and from the local network
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A OUTPUT -o lo -j ACCEPT

#SAMBA/CIFS
$IPTABLES -A INPUT -p tcp --dport 137:139 -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 445 -j ACCEPT

#FTP
$IPTABLES -A INPUT -p tcp --dport 20:21 -j ACCEPT

#DNS
$IPTABLES -A INPUT -p udp --dport 53 -j ACCEPT

#SMTP
$IPTABLES -A INPUT -p tcp --dport 25 -j ACCEPT

#SECURE SMTP
```

```
GNU nano 6.2 iptables.sh
#SECURE SMTP
$IPTABLES -A INPUT -p tcp --dport 465 -j ACCEPT

#POP
$IPTABLES -A INPUT -p tcp --dport 110 -j ACCEPT

#SECURE POP3
$IPTABLES -A INPUT -p tcp --dport 995 -j ACCEPT

#TCP PORT 143
$IPTABLES -A INPUT -p tcp --dport 143 -j ACCEPT

#SECUR EIMAP
$IPTABLES -A INPUT -p tcp --dport 993 -j ACCEPT

#DNS
$IPTABLES -A INPUT -p tcp --dport 53 -j ACCEPT

#HTTP
$IPTABLES -A INPUT -p tcp --dport 80 -j ACCEPT

#HTTPS
$IPTABLES -A INPUT -p tcp --dport 443 -j ACCEPT

#ICMP request & reply
$IPTABLES -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
$IPTABLES -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

#Outbound packets
$IPTABLES -A OUTPUT -j ACCEPT

#stateful packet inspection
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "Firewall configs done"
```

Iptables.sh with the required policies.

4.2 securing dns

1. See 2.
2. Appending the following lines into named.conf.options (i initially had it in named.conf which meant I couldn't run the bind9-server later on)

```
dnssec-enable yes;  
dnssec-validation yes;
```

3. Generating zsk and ksk

```
fillesteng@fillesteng:/var/snap/bind9-jdstrand/current/etc/bind$ sudo dnssec-keygen -a RSASHA256 -b 1024 -n ZONE cali.  
Generating key pair.....  
Kcali..+008+53806  
fillesteng@fillesteng:/var/snap/bind9-jdstrand/current/etc/bind$ sudo dnssec-keygen -a RSASHA256 -b 2048 -f KSK cali.  
Generating key pair.....  
Kcali..+008+18370  
fillesteng@fillesteng:/var/snap/bind9-jdstrand/current/etc/bind$
```

According to <https://linux.die.net/man/8/dnssec-keygen> RSA keys must be between 512 and 2048. And it also mentions that the default bit size for ZSK is 1024 and KSK 2048, which I picked. Seeing as it is the standard it probably is the ideal size for getting sufficient security with low resources needed.

4. Dnssec-keygen generates two pairs:

.key contains public key and a DNS KEY record.

.private contains the private key and data regarding algorithm.

ZSK pair:

```
fillesteng@fillesteng:/var/snap/bind9-jdstrand/228/etc/bind$ sudo cat Kcali..+008+53806.key  
; This is a zone-signing key, keyid 53806, for cali.  
; Created: 20231123153617 (Thu Nov 23 16:36:17 2023)  
; Publish: 20231123153617 (Thu Nov 23 16:36:17 2023)  
; Activate: 20231123153617 (Thu Nov 23 16:36:17 2023)  
cali. IN DNSKEY 256 3 8 AwEAAeB0IV8gyTLo2za8Pb8GqkvkX1Fp9od92ejhrmhgJuub+MpkHgU f7u0pYHOz7duRI9+QAe487ILkzBTntE+Bm0ShPncJ+sWKL204iU2iF  
DS w/c1Sz57D3LS2DgljovBVitw9LIu6nbk6IOyJ8CUfvwJl0FKYTEJ78eq 5VvHXArH  
fillesteng@fillesteng:/var/snap/bind9-jdstrand/228/etc/bind$ sudo cat Kcali..+008+53806.private  
Private-key-format: v1.3  
Algorithm: 8 (RSASHA256)  
Modulus: 4E7VxyDJOwjbNrw9vwageS+RfUwn2h33Z60GuaGAm65v4ymQeBR/u7SLgc7Pt25Ej35AB7jzsguTMFoA0T4GY5KE81wn6xYqXY7iJTaIUNLD9zXnPnsPcuzYOCWM68  
FWK1b2Ui7qduTog7InwJR+/AmXQWRhMQnvx6rLW8fECsc=  
PublicExponent: AQAAB  
PrivateExponent: jt0G/2C1aRBE1rbMKPFI6giHBJ+xybVeVy8K5V5bEGzgzHEzp5d+x4HGDEg5eh4MIsyJYaHF7Pd8VsJZD740eHmaK9j6GmYWRLha8WoSNCzavHicBxdMq  
BT6W/KbZx9LNGF90+3yCWq6321SLbtz3f0wv+CYAD4QFnXbwft49k=  
Prime1: 8r92PlpAW709KGXSeoDmITwBTxQIyx6Ifg4yAmEGsCkmhP8YK6II0Kz3nfefTg6h6JuhLuRwLxzMR07FnC09PQ==  
Prime2: 7I2pslvPFR69sqEvLMgN/DdBmzVNSEjL4VIA598q8ppXFU1JZX9GY1piG0LEIEI3L9IbLHxAmrofIIZxXDwUw==  
Exponent1: 2EpudY/RfYewxIGunSDmHU4zVcShrLL8NWBjBGCCHbSa/BHfWk+vhEQCMaMQSa0l0eydG/crZfBRiAhfwloGQ==  
Exponent2: sDxSoiKXWozengjQFrLTCG06aaHqc+IYU0RSW1G+s4RM70fnWMDb9gVodjTly9gl8wq3sclGn6B6mNhsztZ3lQ==  
Coefficient: 6c4FiM4FD0dGist0ISmKgzR75BXIoRn+E2dwhjgQ5l6k65JHZHPnr85nJWqvoh1mhcgHHJtPVZrtLDnXQypp5g==  
Created: 20231123153617  
Publish: 20231123153617  
Activate: 20231123153617  
fillesteng@fillesteng:/var/snap/bind9-jdstrand/228/etc/bind$
```

KSK pair:

```
fillesteng@fillesteng:/var/snap/bind9-jdstrand/228/etc/bind$ cat Kcali..+008+18370.key  
; This is a key-signing key, keyid 18370, for cali.  
; Created: 20231123153700 (Thu Nov 23 16:37:00 2023)  
; Publish: 20231123153700 (Thu Nov 23 16:37:00 2023)  
; Activate: 20231123153700 (Thu Nov 23 16:37:00 2023)  
cali. IN DNSKEY 257 3 8 AwEAAa26mWfKrgw03sPdwcS1YVInlwksFJGkmve/QI6Bu+4fm6ZD/xu ACD8VLbskebXYKMsJrVfhkL15LQI0dPsSyjxFDCIkSaPffp5dYsme  
Sy acJyX1d2pkpLb6APVgt32wo00+v8cQctb0T70sfq9lQWYjzUQISG4mK Uf0Vh3rDGLNPr1FIfa1L5JLmIpqbAFDzx1WqQBFQ/vVLP3rBUPew2g wiaeYgjbH9G0YLD9M  
vch9g+m8MJMKWQ4bHfokScZ5lq6ShwVsp68arX YJ9Cj1fbgsgAgop0x1Vt9XUDLnUmPPeyHs09zLVhn7bFvOkWnkLZ6d5r fvo+/W2SCF8=  
fillesteng@fillesteng:/var/snap/bind9-jdstrand/228/etc/bind$ cat Kcali..+008+18370.private
```

```

fillesten@fillesten: /var/snap/bind9-jdstrand/228/etc/bind$ sudo cat Kcali.+008+18370.private
Private-key-format: v1.3
Algorithm: 8 (RSASHA256)
Modulus: rbbqA18quDDTew38xLVhUieXCSwUkaSa979AjoG77h+bpkP/G4AIPxUtuyR5tdgoywmtV+GSUVXktAjR0+xlKPEUMKKRJo99+n11yZ7nJoKNhfV3amSktvoA+9WC
3FbG476/xxAK1s5P56x+2VBZ1PNRCJ1biYpR/RMHesMYs0+vUUh9rUvkvuYmSpsAUPPHVaQF9D+9UuLlesFQ95baDDVp5gaNuMf0bRIUP0y8KH2D4zwwk0pZDhsd+itKLpm
KrrnHBWynrxqtdgn1yPV9uCyocClnTHVW31dQMudS5897Iew730W8eftsW86RaeSVnp3mt++j79bZ1J/w==
PublicExponent: AQAB
PrivateExponent: MITADBYE+UHxnm4LUomRb39NCcu+ZtqAYRypsmL7GErNjobI9WYxZ2TDy9MpbQci5WmDdtqxoeegqs/5p7Ect+5USfkh8/NUChsUYDGB5PgChWYkDI
9S0nHy4HCXSD6HgDE0JRjYpB14/kVvGhb9mmjRxcIAXM47jaH4n0jtZ39ALmFRQa3MDd44io/4feBFysgfwU2PpHwqlT7Ho8dNjHSXbwDYTLmgwRe4Rx80+B14FS3+J8Wbo/B6
LNmGUJveRCOC7WGIC+F0gBuva1q7Jd6AsRHULLr22kd+y14Luh/f/r39u4uDLkP56UTnM1b8RLOfkk8ePj05gaQ==
Prime1: 3eHSnttK0XxtTwux3kg9RX2Vv1Fr6ND2W9DUAs6z7eXpQKTEq30g+xMXLdmjonb3uP/5jYm0AHDysEID02wtHly9gcB2aIBcrvZXBSOXPF3syrspGg94WHWZFCAT
HpKV8tqpqP+79zTPQbrDAQKZdtBVPbFmKjuZaCIjajF0=
Prime2: yHFHp2fcd1UTnmLuvzFoMP4cB8gHpuTiaM3B5iIaD0+nYx1a3uQyJ+3LGTUhap2D36K+0BE660DSruiBex4ELGK7gmnGB0QL/igZudd7ohNj3vfH1zk29n2Wv7wPbb
USyzqS/p4uyhnPTJURMnSW/jN7dw0k97Md+hr6RFV6gs=
Exponent1: RXV//TRA1CQanfrqK6S/8IESt0MHeeyF6QT+kHEM2phyoI30E0grS038/5l2JFH0sQvNx+aZTN003yaLJHsE1MFkwc2ZL6GFYhYzWchJqWifaLMzSXRoirm6B8
U7xUPTKhC/3lLQZ/SVENGr+yL4ZuCoXsXszjdGUAznfPE=
Exponent2: elK9SejhsxUBfJRDq0WwvjsjSdg3FM7rRUI7NkV9RCWLFunsV0xVcQF0tiYkKbMTAp2xQmzXqCUWankFwVoDJf6CuULiAgvLE3esN3+KdDR78Q4WNQFEB2dFR
TCxXKwfgf74EvaE42xbkTeKVH2qQMEDAZ34TSoG5FI7WJ8=
Coefficient: iqn8Fns7lX/Cf3EVdE+ll1cvaSpTC1U5vX5KBxY37sg1uLFZhfuzW/aI/BFWLCSMB/Y850MfAhSEp0n1ICbRVv4wuf1V3kyjdDtG0rvTVY3qhCWD5TH7wvkvk
+MDRC6k94RH5F3CRBFxGdEypNFwa3Kp3sFFTRwxuqGeTf1ik=
Created: 20231123153700
Publish: 20231123153700
Activate: 20231123153700
fillesten@fillesten: /var/snap/bind9-jdstrand/228/etc/bind$

```

5. Include KSK AND ZSK. HOWEVER I used cat instead.

cat Kcali.+008+18370.key >> cali.db

cat Kcali.+008+53806.key >> cali.db

```

GNU nano 6.2 cali.db
@ IN SOA ns1 mail(
2023103001 ; serial
3600 ; refresh
1800 ; retry
604800 ; expire
3600 ) ; minimum

@ IN NS ns1
ns1 IN A 10.0.2.16

duckburg IN NS ns1.duckburg.cali
ns1.duckburg.cali IN A 10.0.2.17

mail._domainkey IN TXT ( "v=DKIM1; h=sha256; k=rsa; t=y; "
"p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAywwOgNyl/2RbSF5NQAAsGdGZBJ0Bz3A9NkVN+NX5xIagBBBjBMfk1mtKLxvBxu1Xg+VMJAb+7B
"2ycebaOKT4LizE0k0rncXJ901IJ7NMULvAHRIdz8DzRpvzPzQLD24HUAOTyAPW2/f3N0x9hldbdANS1iInZekGJ7L4GagxhoWdsvZsxN5a64/bhwHvW153r8kG
; This is a key-signing key, keyid 18370, for cali.
; Created: 20231123153700 (Thu Nov 23 16:37:00 2023)
; Publish: 20231123153700 (Thu Nov 23 16:37:00 2023)
; Activate: 20231123153700 (Thu Nov 23 16:37:00 2023)
cali. IN DNSKEY 257 3 8 AwEAAa26mNfKrgw03sPdwS1YVInLwksFJGkmve/QI6Bu+4fm6ZD/xu ACD8VLbskebXYKMsJrVfhkL15LQI0dPsSyjxFDCIkSaPfp5dYs
; This is a zone-signing key, keyid 53806, for cali.
; Created: 20231123153617 (Thu Nov 23 16:36:17 2023)
; Publish: 20231123153617 (Thu Nov 23 16:36:17 2023)
; Activate: 20231123153617 (Thu Nov 23 16:36:17 2023)
cali. IN DNSKEY 256 3 8 AwEAAeB01V8gyTlo2za8Pb8GqnkvKX1Fp0d92ejhrmhgJuub+MpkHgU f7u0pYH0Z7duRI9+QAe487ILkzBTmtE+Bm0ShPncJ+swKL204iU2i
Wrote 30 lines

```

6. Signing the zone following using commands from this guide

(<https://www.digitalocean.com/community/tutorials/how-to-setup-dnssec-on-an-authoritative-bind-dns-server-2>)

```

fillesten@fillesten: /var/snap/bind9-jdstrand/228/etc/bind$ sudo dnssec-signzone -A -3 $(head -c 1000 /dev/random | shasum | cut -b 1-16) -N INCREMENT -o cali. -t cali.db
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
cali.db.signed
Signatures generated: 15
Signatures retained: 0
Signatures dropped: 0
Signatures successfully verified: 0
Signatures unsuccessfully verified: 0
Signing time in seconds: 0.020
Signatures per second: 749.662
Runtime in seconds: 0.032

```

This creates a cali.db.signed file

```

GNU nano 6.2 cali.db.signed
File written on Thu Nov 23 20:57:26 2023
; dnssec-signzone version 9.18.18-0ubuntu0.22.04.1-Ubuntu
cali. 2592000 IN SOA ns1.cali. mail.cali. (
2023103002 ; serial
3600 ; refresh (1 hour)

```

7. What must be added to parent zone to ensure a chain of trust?

To make sure that the chain of trust is validated the parent zone needs to acquire a DS record. DNSSEC has a DS record which transfers the trust from the parent zone to the child zone. The DS record is a hashed DNSKEY record which contains the KSK. This continues until the root parent has received a DS from their child zone.

8. When is it necessary to enable the the dnssec-lookaside option?

if the parent is not signed or doesn't publish DS's unlike the child zone the lookaside option is enabled to still validate the child.

"DNSSEC Lookaside Validation (DLV) is a mechanism for publishing DNS Security (DNSSEC) trust anchors outside of the DNS delegation chain. It allows validating resolvers to validate DNSSEC-signed data from zones whose ancestors either aren't signed or don't publish Delegation Signer (DS) records for their children." - <https://www.rfc-editor.org/rfc/rfc5074>

9. present key rollover for ZSK and KSK and how to achieve it in a secure manner for both.

ZSK rollover is relatively straightforward and does not involve your parent zone or any trust anchor issues. The only tricky part is the timing. Keys have an expiration time, so rollover must occur well before that time. However, keys also have a TTL, defined in the zone file. To illustrate, assume that the TTL is one day and that keys don't expire for another week." (Unix and Linux System Administration handbook 5th edition, page 565)

It also describes the steps to solve it, these are the steps:

- **Generate a new ZSK.**
- **Include it in the zone file.**
- **Sign or re-sign the zone with the KSK and the *old* ZSK.**
- **Signal the name server to reload the zone; the new key is now there.**
- **Wait 24 hours (the TTL); now everyone has both the old and new keys.**
- **Sign the zone again with the KSK and the *new* ZSK.**
- **Signal the name server to reload the zone.**
- **Wait another 24 hours; now everyone has the new signed zone.**
- **Remove the old ZSK at your leisure, e.g., the next time the zone changes.**

However, regarding KSK it utilizes a mechanism called double signing. It includes communicating the DS record to the parent, with a positive acknowledgement from the parent beforehand. the steps for KSK rollover are

- **Create a new KSK.**
- **Include it in the zone file.**
- **Sign the zone with both old and new KSKs and the ZSK.**
- **Signal the name server to reload the zone.**
- **Wait 24 hours (the TTL); now everyone has the new key.**
- **After confirmation, delete the old KSK record from the zone.**
- **Re-sign the zone with the new KSK and ZSK.**

