Mail setup for deSEC

2025-07-18

This page describe the cli mail setup for deSEC

Introduction

This post describes how to configure the mail DNS setup for deSEC via commandline.

Prepare

General information

It's recommended to install the certbot-dns-desec because this makes the interaction with let's encrypt (also written as "LE") much easier.

In this post and for the setup are the following tools required `bash`, `curl`, `jq`, `envsubst` and `certbot`.

For the beginning will we export the two environment variables below.

```
export DNSSEC_TOKEN='Your-DNSSEC-API-Token'
export DOMAIN_NAME='Your-Domain'
```

The value for the `DNSSEC_TOKEN` can be created via the Web UI or via API as documented at Creating a Token.

Create Certificate

The creation of the file `/etc/letsencrypt/secrets/$DOMAIN_NAME.ini` is documented on the certbot-dns-desec
We can now create the LE certificate and key after the preparation above.

> ⚠ **Warning**
>
> Please be aware that an wildcard certificate will be created with the call below!

```
certbot certonly \
  --authenticator dns-desec \
  --dns-desec-credentials /etc/letsencrypt/secrets/$DOMAIN_NAME.ini \
  -d '$DOMAIN_NAME' \
  -d '*.$DOMAIN_NAME'
```

There must be an directory `/etc/letsencrypt/live/${DOMAIN_NAME}/fullchain.pem` after the successfully execution of the certbot call.

The deSEC json file

To be able to create the DNS Resource Record Sets (RRsets) is a json file required. The json file below can be used to create the RRsets via the API call, which is shown at the end.
Save the json text into the file `mail-setup.json`.

```
[
    {
        "subname": "",
        "type": "MX",
        "ttl": 3600,
        "records": [
            "5 mail.${DOMAIN_NAME}."
        ]
    },
    {
        "subname": "",
        "type": "TXT",
        "ttl": 3600,
        "records": [
            "\"v=spf1 mx -all\""
        ]
    },
    {
        "subname": "",
        "type": "A",
        "ttl": 3600,
        "records": [
            "IPv4 Adress"
        ]
    },
    {
        "subname": "",
        "type": "AAAA",
        "ttl": 3600,
        "records": [
            ["IPv6 Adress"]
        ]
    },
    {
        "subname": "",
        "type": "CAA",
        "ttl": 3600,
        "records": [
```

```
            "0 iodef mailto:postmaster@${DOMAIN_NAME}",
            "0 issue letsencrypt.org"
        ]
    },
    {
        "subname": "_25._tcp.mail",
        "type": "TLSA",
        "ttl": 3600,
        "records": [
            "3 1 1 ${TLSA_SHA}"
        ]
    },
    {
        "subname": "_dmarc",
        "type": "TXT",
        "ttl": 3600,
        "records": [
"\"v=DMARC1;p=quarantine;pct=100;rua=mailto:postmaster@${DOMAIN_NAME};ruf=mailto:postmast
        ]
    },
    {
        "subname": "_mta-sts",
        "type": "TXT",
        "ttl": 3600,
        "records": [
            "\"v=STSv1; id=2023042102;\""
        ]
    },
    {
        "subname": "_smtp._tls",
        "type": "TXT",
        "ttl": 3600,
        "records": [
            "\"v=TLSRPTv1; rua=mailto:postmaster@${DOMAIN_NAME}\""
        ]
    },
    {
        "subname": "dkim._domainkey",
        "type": "TXT",
        "ttl": 3600,
        "records": [
            "\"v=DKIM1; p=${DKIM_PUB_KEY}\""
        ]
    },
    {
```

```
        "subname": "mta-sts",
        "type": "CNAME",
        "ttl": 3600,
        "records": [
            "mail.${DOMAIN_NAME}."
        ]
    }
]
```

Add Data to deSEC

There are now two more environment variables necessary before the call to the deSEC API is possible.

- `DKIM_PUB_KEY` is the dkim key from your mail system.
  - for example with amavis `/usr/sbin/amavisd showkeys ${DOMAIN_NAME}`
- `TLSA_HASH` is the sha256 hash for the `TLSA` Record.

example call

The shell code below shows how such a shell session looks like.

> ⚠ **Example**
>
> You will need to adopt the environment variables.

```
export DNSSEC_TOKEN='Your-DNSSEC-API-Token'
export DOMAIN_NAME='Your-Domain'
export DKIM_PUB_KEY='Your-DKIM-Public-Key'
export TLSA_HASH=$(openssl x509 -noout -pubkey \
    -in /etc/letsencrypt/live/${DOMAIN_NAME}/fullchain.pem \
    | openssl pkey -pubin -outform DER \
    | openssl sha256 2>&1 \
    | cut -f2 -d ' ')

envsubst < mail-setup.json|curl -sSLX PUT \
  https://desec.io/api/v1/domains/${DOMAIN_NAME}/rrsets/ \
  --header "Authorization: Token ${DNSSEC_TOKEN}" \
  --header "Content-Type: application/json" --data @- | jq
```