cryptdomainmgr

automating Cert, TLSA, DKIM and many more

Stefan Helmert

Chaostreff Chemnitz

14. November 2018

Content

Motivation

→ **let's make a web app** ←

- DNS
- Webpage
- E-Mail
- Mailinglist
- **and the s for security**

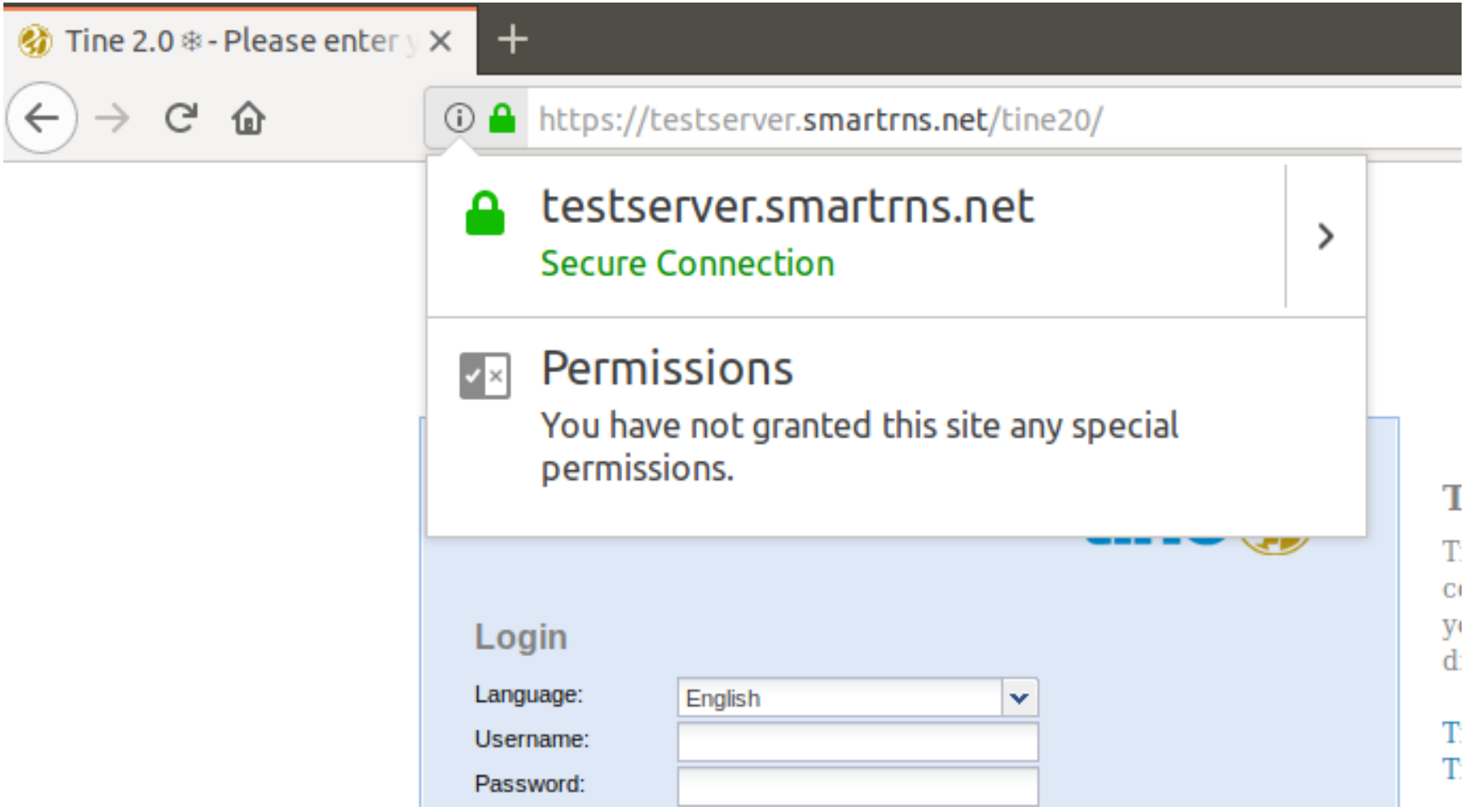
DeMotivation

→ **let's make a web app** ←

- DNS
  - SOA
  - DNSSEC
- Webpage
  - HTTPS
  - Certificate
  - HSTS
  - SRV
  - TLSA
- E-Mail
  - Spam
  - DKIM
  - SPF
  - ADSP
  - DMARC
  - SRV
- Mailinglist
  - SRS

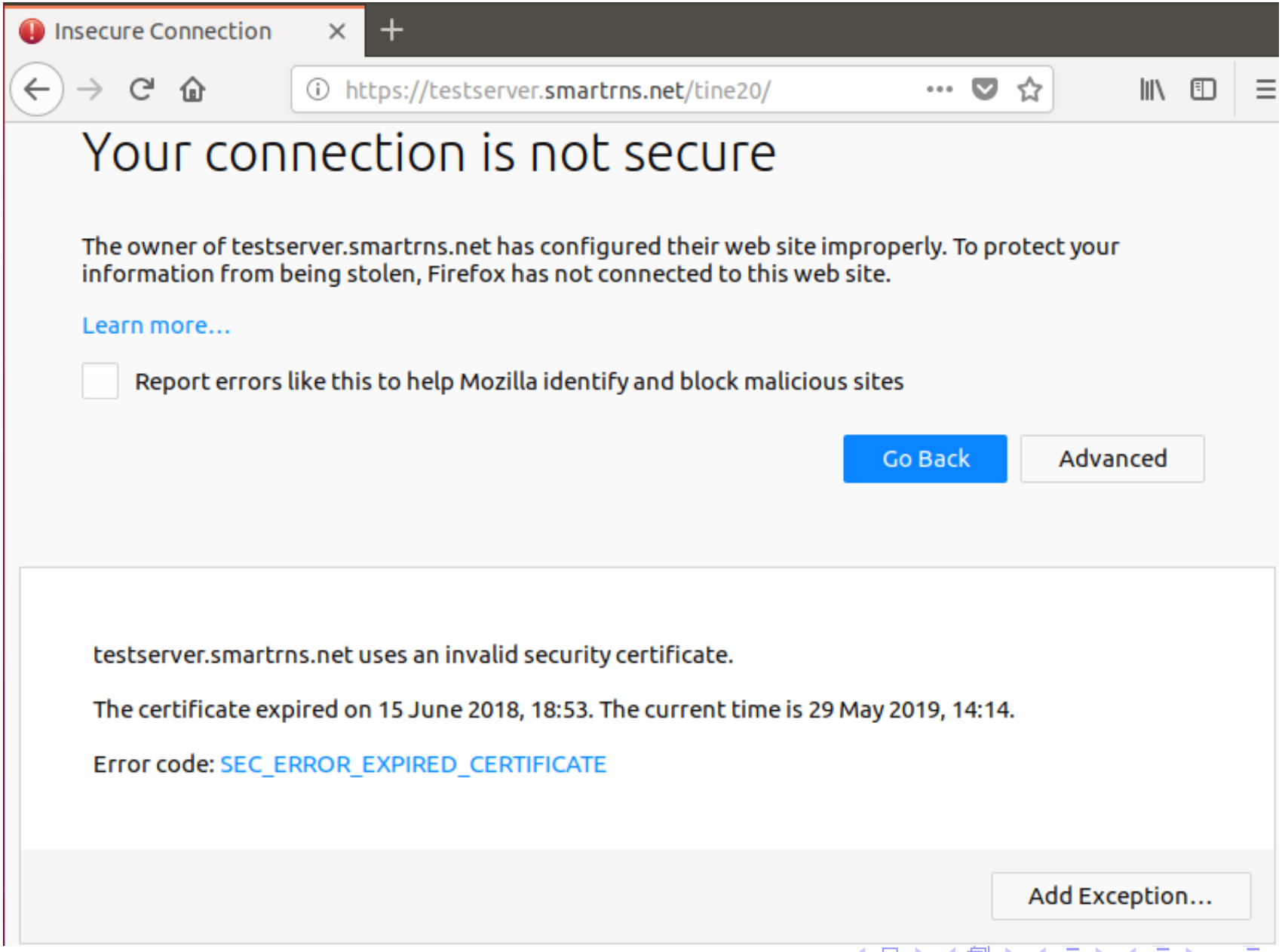
DeMotivation

fine



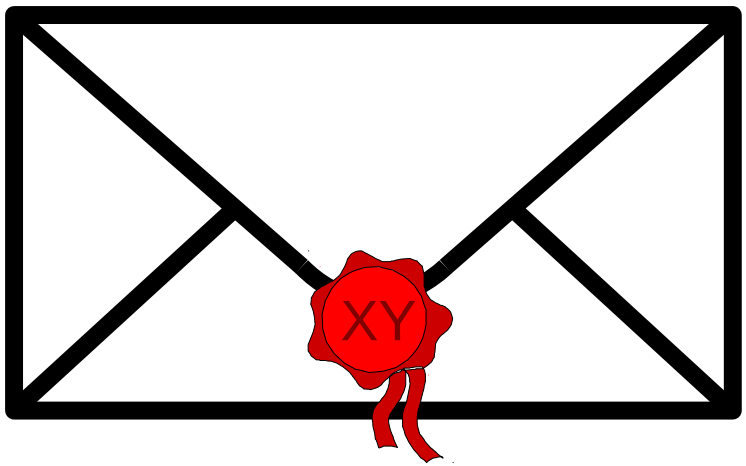
DeMotivation

not so fine



Basics

SSL Certifcate



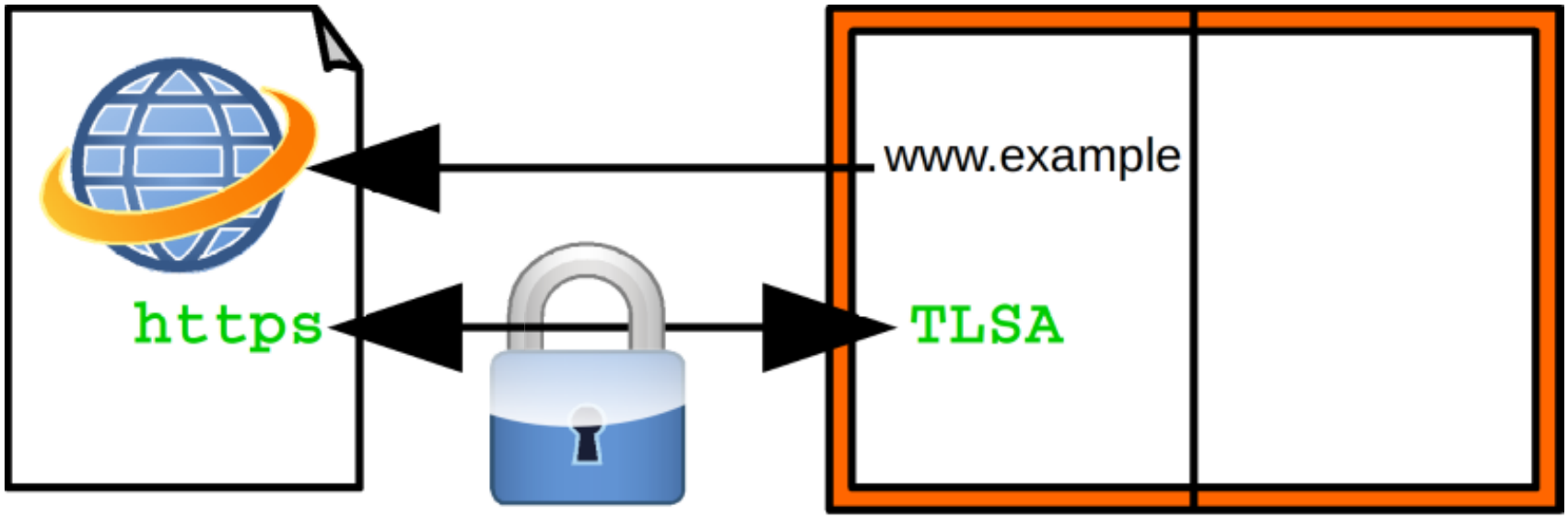
- authentication (phishing)
- integrity (man in the middle)
- privacy (spy)

$\rightarrow$ certbot renew

Basics

TLSA

DANE – DNS-based Authentication of Named Entities



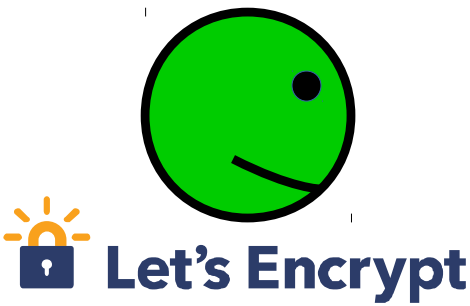
TLSA – Transport Layer Security Authentication

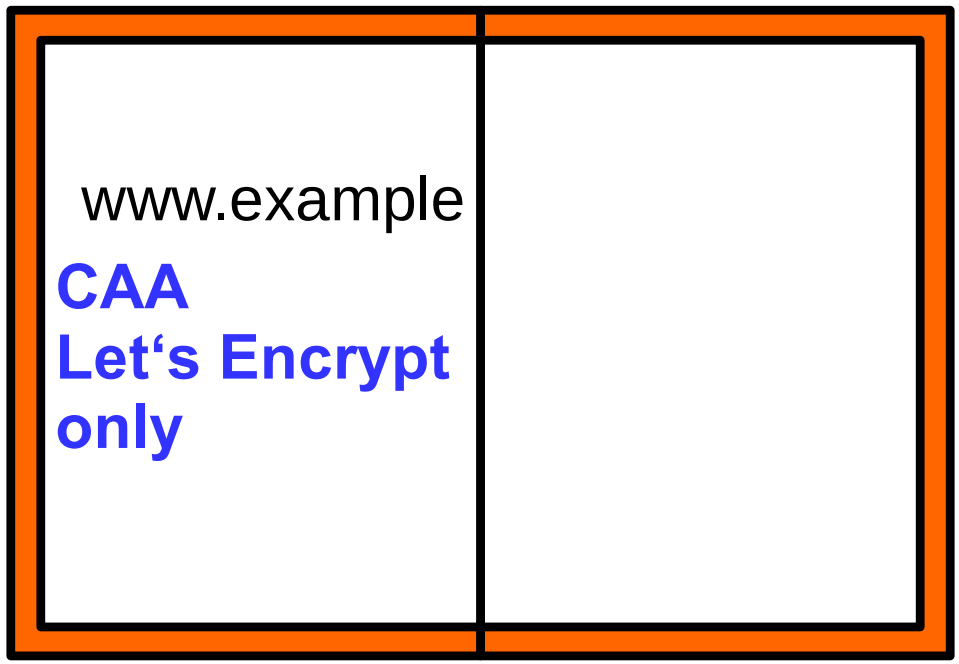
- locks certificate to domain/DNS (fraudulent CA, stolen cert)

→ to do

Basics

CAA



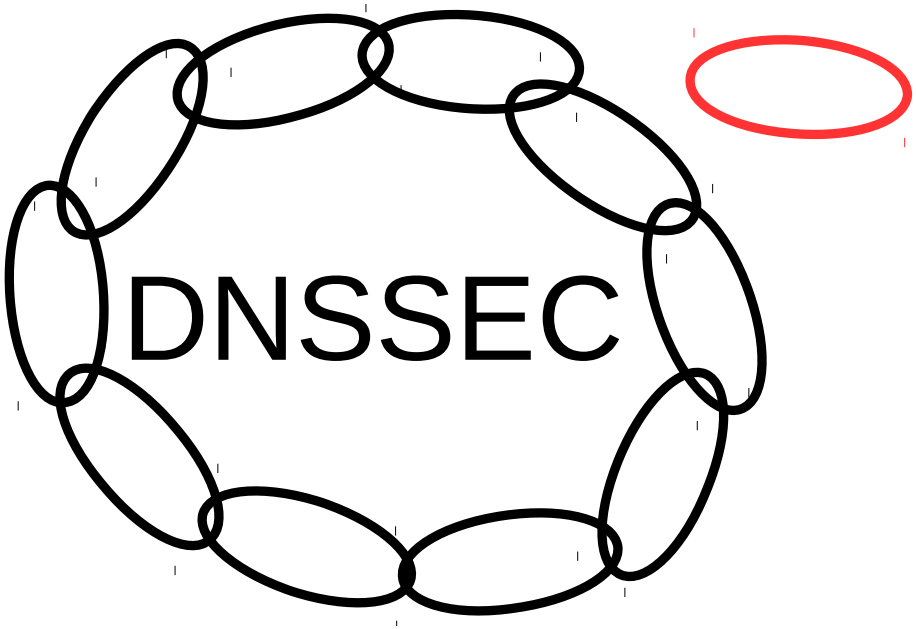


CAA – Certification Authority Authorization

- specifies allowed CA
- checked by CA

Basics

DNSSEC



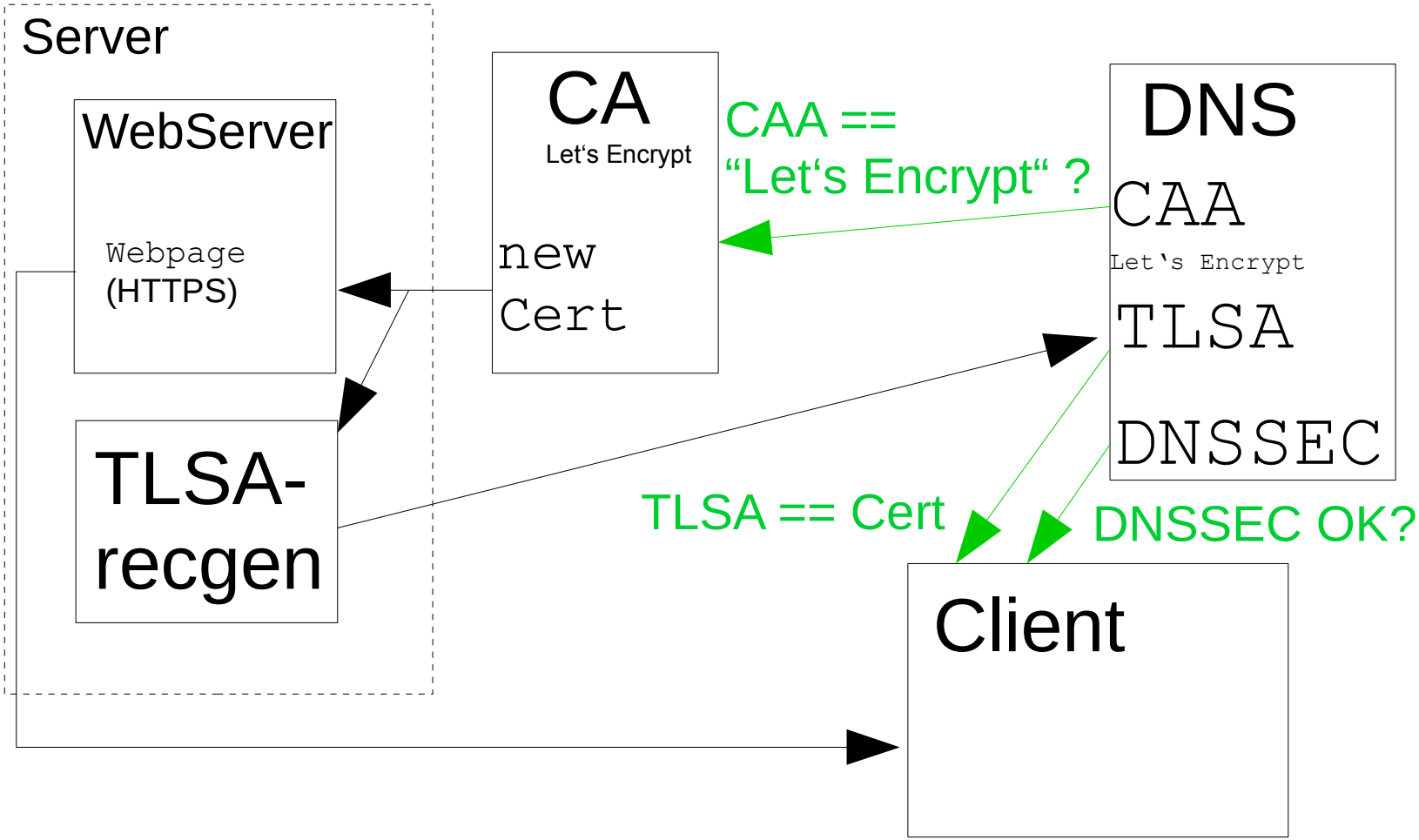
Domain Name System Security Extensions

- authenticate domain owner
- integrity (DNS cache poisoning)
- proof of nonexistence

$\rightarrow$ done by domain provider

Basics

DANE – all steps



Basics

DKIM

DomainKeys Identified Mail

- authenticate MTA (fake/spam server)
- integrity (man in the middle)

→ to do

Basics

additional DNS records

SPF – Sender Policy Framework

- which server is allowed to send email

ADSP – Author Domain Signing Practices

- defines, if email must be DKIM signed

DMARC – Domain-based Message Authentication, Reporting and Conformance

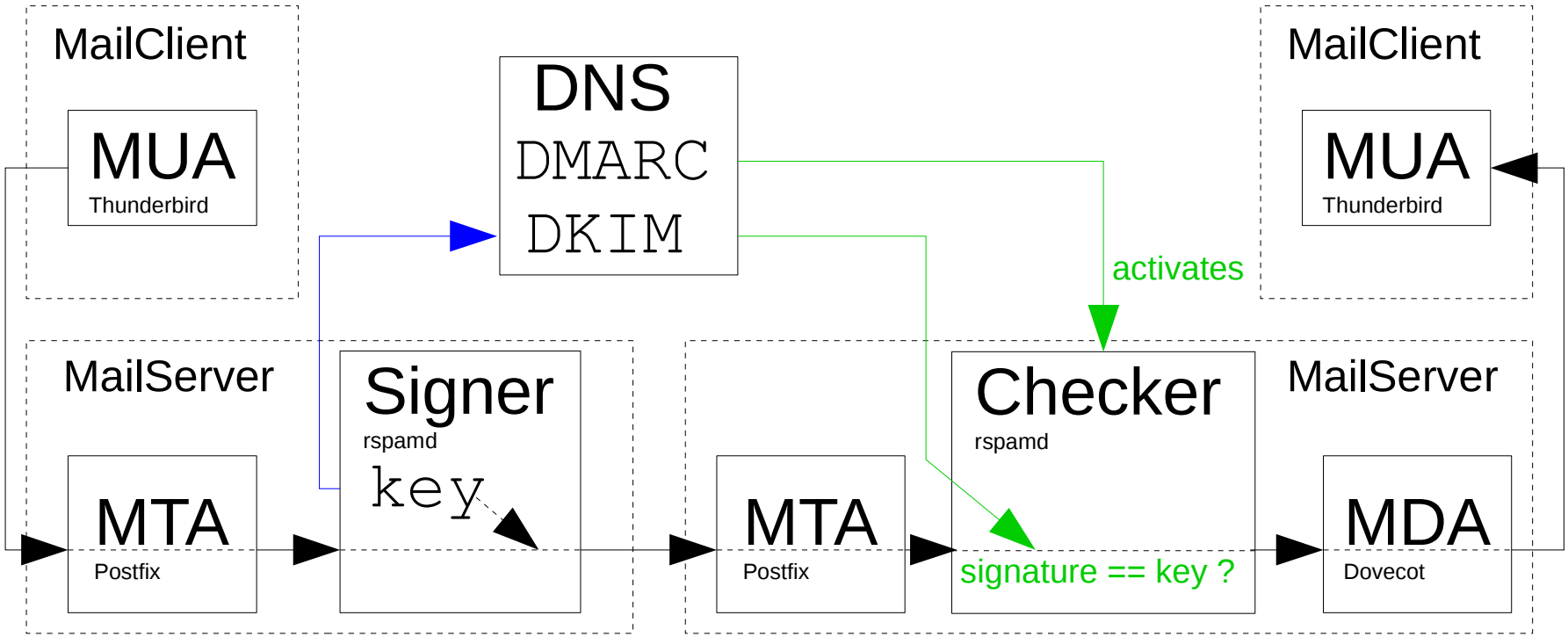
- successor of SPF and ADSP
- overrides SPF and ADSP
- additional parameters: report email

SRV – Service

- announces services

Basics

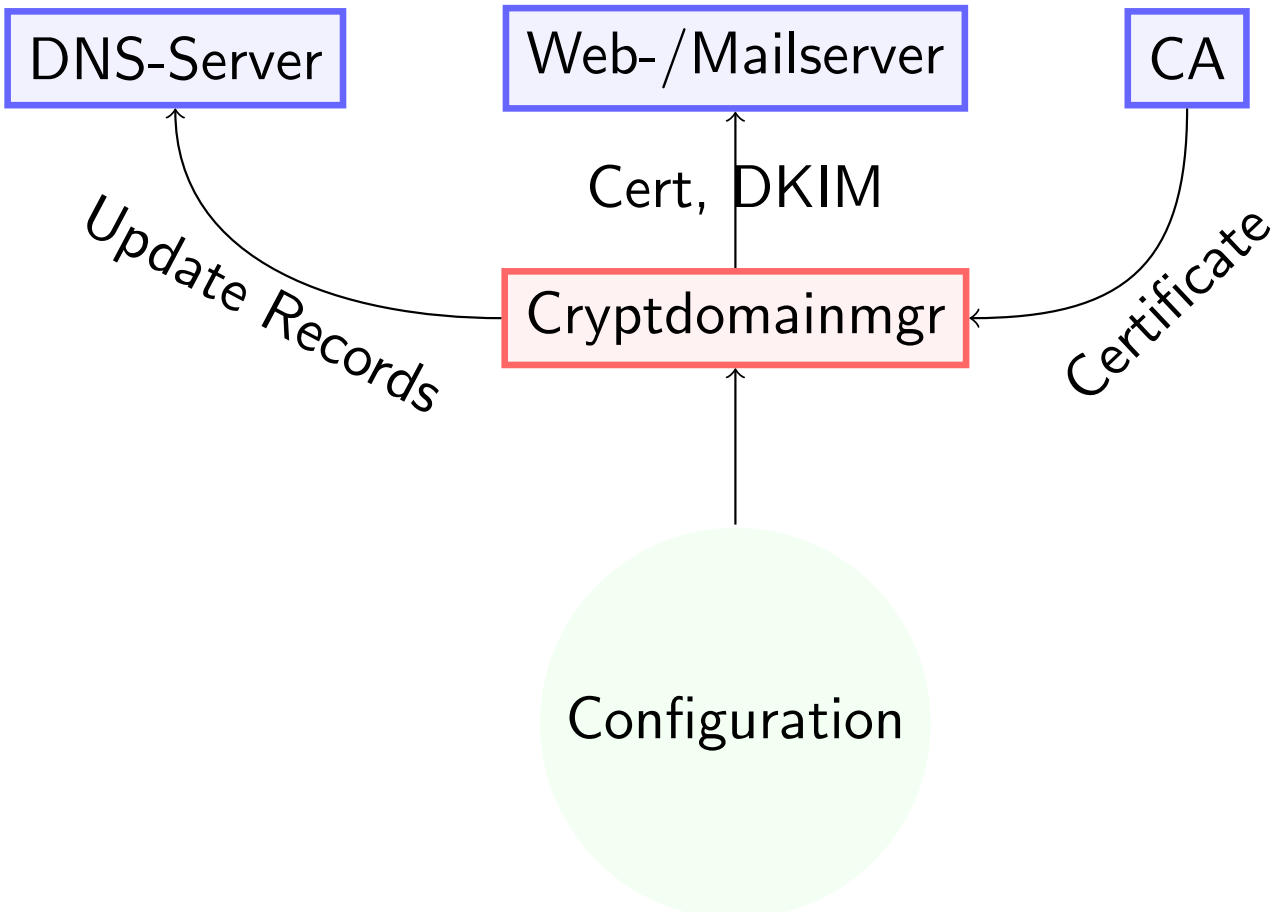
DKIM – overview



Cryptdomainmgr

dataflow

Infrastructure as Code!



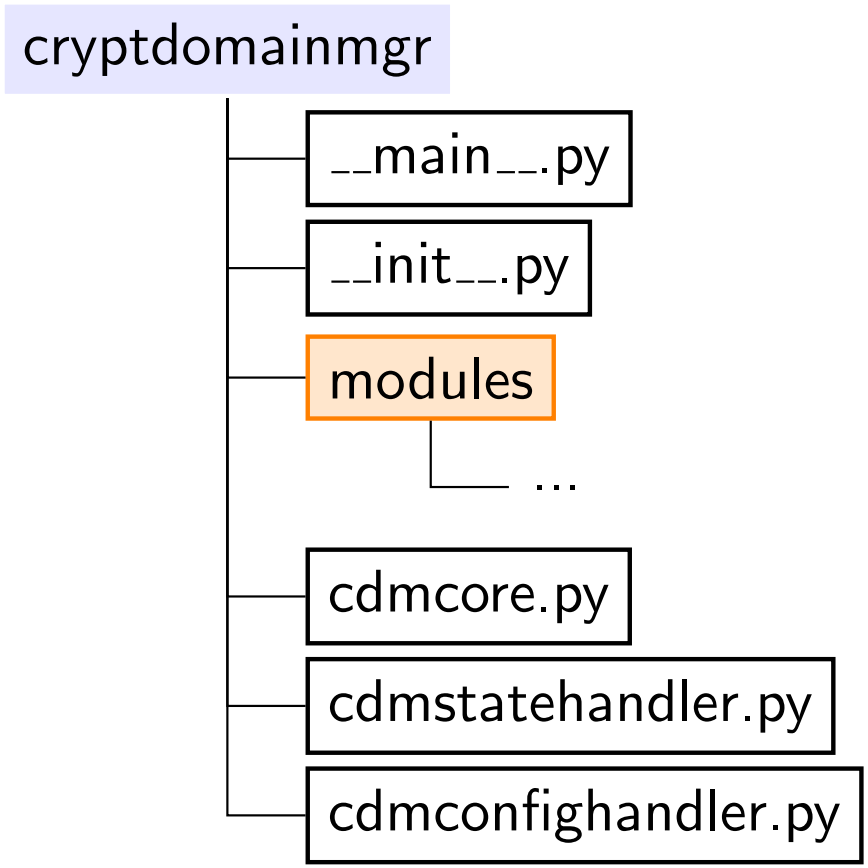
Cryptdomainmgr

autorenew process

- prepare
  - generate certificate
  - calculate TLSA from certificate
  - add TLSA RR
  - generate key pair for DKIM
  - calculate DKIM
  - add DKIM RR
- rollover
  - use new certificate
  - use new DKIM key
- cleanup
  - remove old TLSA RR
  - remove old DKIM RR
  - delete old certificates
  - delete old DKIM keys

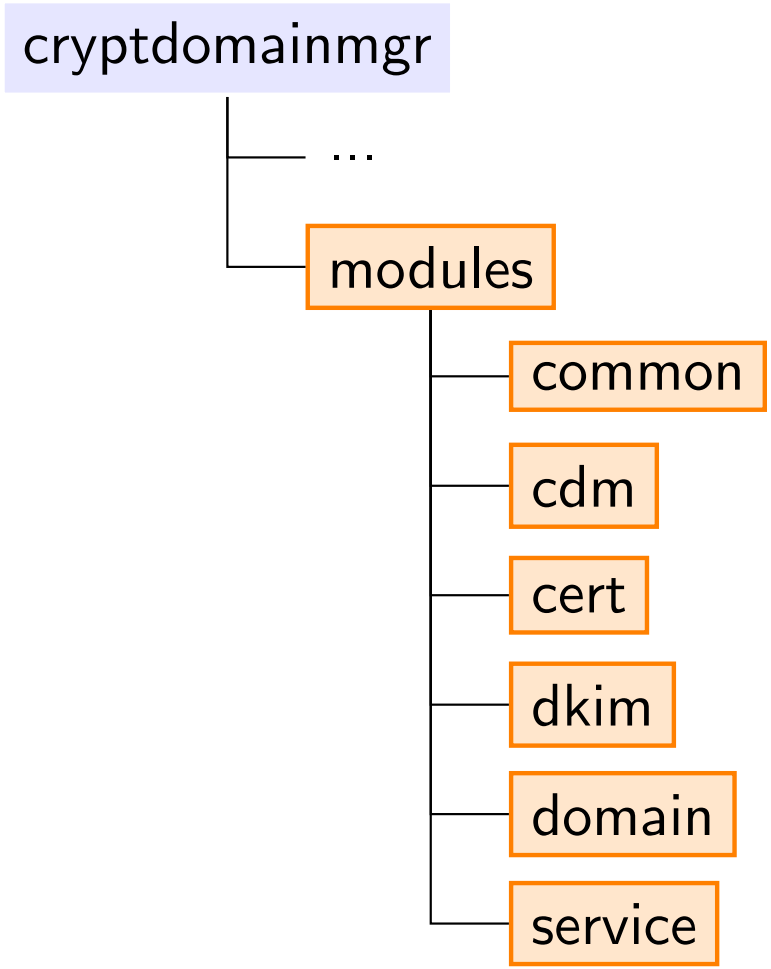
Cryptdomainmgr

structure



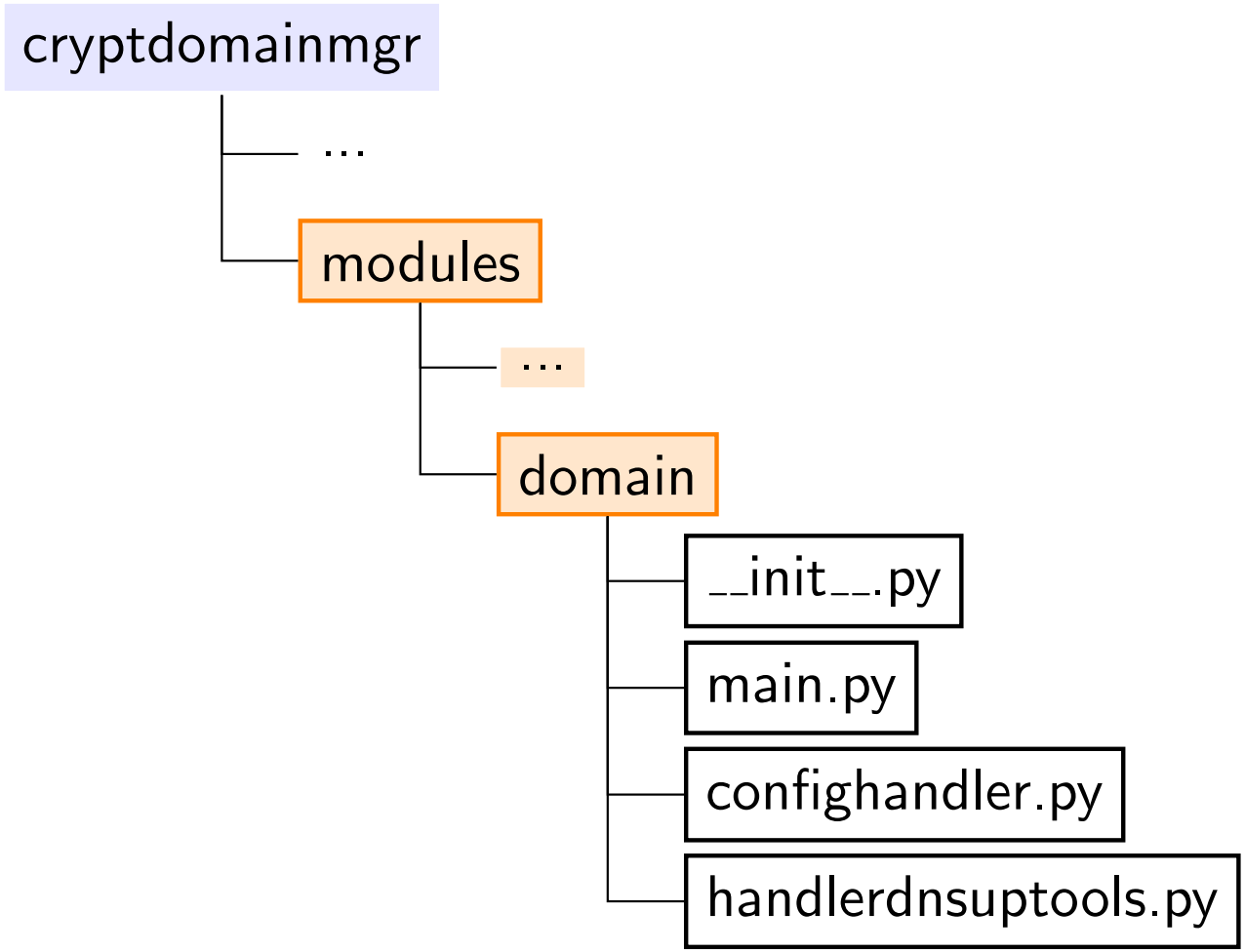
Cryptdomainmgr

structure



Cryptdomainmgr

structure



Cryptdomainmgr

update cycle

update – set a, aaaa, srv, dmarc, spf, adsp

```
$ python -m cryptdomainmgr --update cred.cnf exmpl.cnf
```

prepare cycle – generate Cert, TLSA, DKIM

```
$ python -m cryptdomainmgr --prepare cred.cnf exmpl.cnf
```

rollover cycle – use Cert, TLSA, DKIM

```
$ python -m cryptdomainmgr --rollover cred.cnf exmpl.cnf
```

cleanup cycle – remove outdated

```
$ python -m cryptdomainmgr --cleanup cred.cnf exmpl.cnf
```

Configuration

DNS credential

```
$ cat cred.cnf

[domain]
user = myusername
passwd = mypassword
```

Configuration

Certificates

```
$ cat exmpl.cnf

[cert]
handler = dehydrated
email = stefan.helmert@t-online.de
keysize = 4096

[cert:maincert]
destination = /etc/ssl
extraflags = --staging, -x
certname = fullchain.pem
```

- multiple domains using `maincert` → SAN certificate

Configuration

DKIM

```
$ cat exmpl.cnf

[dkim]
handler = rspamd

[dkim:maindkim]
signingConfTemplateFile
  = /etc/cryptdomainmgr/dkim_signing_template.conf
signingConfTemporaryFile
  = /etc/rspamd/dkim_signing_new.conf
signingConfDestinationFile
  = /etc/rspamd/local.d/dkim_signing.conf
```

Configuration

Domain

```
$ cat exmpl.cnf

[domain]
user = myusername
handler = dnsuptools/inwx

[domain:domain.example]
soa.hostmaster = stefan.helmert@t-online.de
soa.refresh = 7200

[domain:sub.domain.example]
ip4 = auto, 192.168.0.1
ip6+ = auto, 0ffc::0030
mx = mail20.domain.example:20, mail30.domain.example:30
mx.40 = mail40.domain.example, mail50.domain.example:50
mx.10+= mail10.domain.example
```

Configuration

Domain

set A record

```
$ cat exmpl.cnf

[domain:sub.domain.example]
ip4 = auto, 192.168.0.1
```

means:

- add external ip and `192.168.0.1` to sub.domain.example
- delete all other A records of sub.domain.example

Configuration

Domain

add A record

```
$ cat exmpl.cnf

[domain:sub.domain.example]
ip4+ = auto, 192.168.0.1
```

means:

- add external ip and `192.168.0.1` to sub.domain.example
- ~~delete all other A records of sub.domain.example~~

Configuration

Domain

set MX record

```
$ cat exmpl.cnf

[domain:sub.domain.example]
mx = mail20.domain.example:20, mail30.domain.example:30
```

means:

- add MX records
  - mail20.domain.example with prio 20
  - mail30.domain.example with prio 30
- delete all other MX records from sub.domain.example

Configuration

Domain

set MX record

```
$ cat exmpl.cnf

[domain:sub.domain.example]
mx.40 = mail40.domain.example, mail50.domain.example:50
```

means:

- add MX records
  - mail40.domain.example with prio 40
  - mail50.domain.example with prio 50
- delete all other MX records with prio 40 from sub.domain.example

Configuration

Domain

set SRV record

```
$ cat exmpl.cnf

[domain:sub.domain.example]
srv.service.proto.port.weight.prio
  = sub.domain.example:PRIO:WEIGHT:PORT:PROTO:SERVICE
```

Configuration

Domain

set DMARC entries

```
$ cat exmpl.cnf

[domain:sub.domain.example]
dmarc.p = quarantine
dmarc.rua = mailto:stefan.helmert@t-online.de
dmarc.ruf = mailto:stefan.helmert@gmx.net
```

- changes the entries `p`, `rua`, `ruf` of the `DMARC` record
- entries `adkim`, `aspf`, `pct` do not change
- „atomic“ operation
- only one `DMARC` record allowed!

Configuration

Domain

set DMARC record

```
$ cat exmpl.cnf

[domain:sub.domain.example]
dmarc =
dmarc.p = quarantine
dmarc.rua = mailto:stefan.helmert@t-online.de
dmarc.ruf = mailto:stefan.helmert@gmx.net
```

- changes the entries `p`, `rua`, `ruf` of the `DMARC` record
- remove all other entries of this record
- atomic operation
- at most one `DMARC` record allowed!

Configuration

Domain

set SOA entries

```
$ cat exmpl.cnf

[domain:domain.example]
soa.hostmaster = stefan.helmert@t-online.de
soa.refresh = 7200
```

- changes the entries `hostmaster`, `refresh` of the `SOA` record
- `primns`, `serial`, `retry`, `expire`, `ncttl` not changed
- atomic operation
- exact one `SOA` record in top level allowed!

Configuration

Domain

set SPF flags

```
$ cat exmpl.cnf

[domain:domain.example]
spf = -mx, a, ?all, +aaaa
```

- add given flags to SPF record
- remove all other flags from SPF record
- atomic operation
- at most one SPF record is allowed!

Configuration

Domain

set ADSP and CAA records

```
$ cat exmpl.cnf

[domain:domain.example]
adsp = all
caa =   0 issue letsdecrypt.org,
      128 issuewild examplecert.example
```

- atomic update ADSP record
- add the CAA records
- remove all other CAA records

configured by cert handler:

```
[domain:domain.example]
caa =   auto
```

Configuration

Domain

combine stuff – TLSA and DKIM

```
$ cat exmpl.cnf

[domain:sub.domain.example]
tlsa.tcp.443 = auto:3:0:1, auto:2:0:1
cert = maincert
dkim = maindkim
```

prepare cycle

- add `TLSA` and `DKIM` records

rollover cycle

- no DNS changes
- apply certificates and keys on server

cleanup cycle

- add `TLSA` and `DKIM` records (again)
- remove all other `TLSA` and `DKIM` records

Implementation

cryptdomainmgr

__main__.py command line interface

cdmcore.py core, brings everything together

cdmconfighandler.py reads/interpretes config (ini) files

cdmstatehandler.py manages dependencies, data transport, next run phase

modules/ plugins handling/interfacing dns update, certificate renewal, dkim renewal, service reload

external packages:

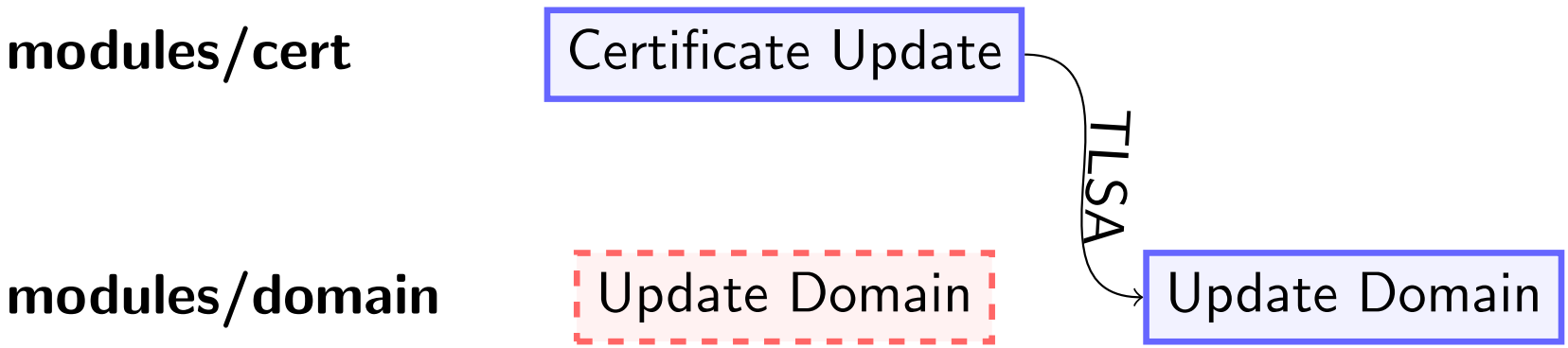
simpleloggerplus logging abstraction, password $\rightarrow$ *****

dnsuptools domrobot interface abstraction, TLSA, DKIM calculation

Implementation

cryptdomainmgr

Reactive: Domain update depends on TLSA record calculated based on new certificate.



Implementation

modules

modules/cert/main.py interface to handler, some helpers

modules/cert/handlerdehydrated.py interface todehydrated to create certificate

modules/cert/confighandler.py interpretes corrspondig parts of the config file

external package:

dehydrated handles acme api for letsencrypt

Implementation

simpleloggerplus

simpleloggerplus.py core, produces output

deepops.py deep dict/list operations, password $\rightarrow$ *****

Implementation

dnsuptools

dnsuptools.py core, high level, record change & query methods

dnsupdate.py interface to wrapper, low level

inwxwrapper.py interface to internetworx api, lowest level

dkimrecgen.py reads/interpretes dkim key file

tlsarecgen.py reads/interpretes certificate file

dnshelpers.py one helper function

external packages:

simpleloggerplus see simpleloggerplus 3

- inwxclient domrobot client

Discussion

???