DANE und DNSSEC

Universität Potsdam
Institut für Informatik und Computational Science

Potsdam, den 11. März 2020

Warum reicht TLS allein nicht aus?

Motivation

- Transport Layer Security (TLS) ist das Standardverfahren zum Verschlüsseln des Datentransports
- Über eine PKI können digitale Zertifikate ausgestellt, verteilt und geprüft werden
- Die Authentizität der verwendeten Zertifikate kann nicht immer gewährleistet werden

Gliederung

1. DANE
2. DNSSEC
3. BIND9
4. Postfix
5. Key Rollover
6. Key Management
7. Ausblick
8. Fazit

1. DANE

DNS-based **A**uthentication of **N**amed **E**ntities:

- DANE führt den TLSA-Record ein
- Prüfen von TLS-Zertifikaten über DNS
- Auch für SSH, PGP und S/MIME einsetzbar
- Setzt auf DNSSEC auf und ist im Prinzip eine eigene PKI
- Root DNSSEC Key wird von der ICANN verwaltet

1. DANE: TLSA Record

_25._tcp.mail.dnssec-uni-potsdam.de. IN TLSA **3** 1 1 […]

TLSA Certificate Usages:

- 0: CA Constraints: Das Zertifikat muss von der angegebenen CA stammen. Hash aus dem Public Certificate generiert. Trust chain muss gültig sein.
- 1: Certificate Constraints: Nur das angegebene Zertifikat darf mit der Domain eingesetzt werden. Hash aus diesem Zertifikat.
- 2: Trust anchor assertion: Das Zertifikat muss von der angegebenen CA stammen. Hash aus Public Cert der CA. Keine Trust chain-Überprüfung.
- 3: Domain-Issued certificates: Nur das angegebene Zertifikat darf mit der Domain eingesetzt werden. Hash aus dem eigenen Public Cert. Keine Trust chain-Überprüfung.

1. DANE: TLSA Record

_25._tcp.mail.dnssec-uni-potsdam.de. IN TLSA 3 **1** 1 […]

TLSA-Selectors:

- 0: Gesamtes Zertifikat wird gehashed
- 1: Nur die “SubjectPublicKeyInfo” wird gehashed

1. DANE: TLSA Record

_25._tcp.mail.dnssec-uni-potsdam.de. IN TLSA 3 1 **1** […]

TLSA Matching Types:

- 0: Kein Hash: Zertifikatsdaten werden direkt verglichen.
- 1: SHA-256
- 2: SHA-512

1. DANE

Bisher: PKI und Vertrauen auf vorinstallierte Root-Zertifikate

Jetzt: DNS-Abfrage und Vertrauen in den DNS-Root-Key

➡ Anstatt Vertrauen auf Root-Zertifikate Die Zugehörigkeit vom Public Key und Dienst können über den Hashwert im TLSA-RR validiert werden

2. DNSSEC

Domain **N**ame **S**ystem **S**ecurity **E**xtensions:

- Gewährleistet Authentizität und Integrität der DNS-Resource Records durch Signaturen
- RFC 2535 (1999)
- …
- RFC 6781 (2012)

2. DNSSEC: Protocol Flags

dig dnssec-uni-potsdam.de +dnssec +multiline

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40631
;; flags: qr rd ra **ad**; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1
; EDNS: version: 0, flags: **do**; udp: 4096

- AD: Authenticated Data
  - Der Resolver hat die DNSSEC-Validierung durchgeführt und als authentisch bestätigt
- CD: Checking Disabled
  - Client wünscht keine DNSSEC-Validierung durch den Resolver
  - Client will Records selber prüfen
- DO: DNSSEC OK
  - Resolver versteht DNSSEC-Records

2. DNSSEC: Resource Records

DNSKEY	Schlüssel	ZSK (256) oder KSK (257), (fixed) Protocol (3), Algorithm, Hash
RRSIG	Signatur über RRset	Eintrag UND ZSK erstellt eine Signatur
DS	Delegated Signer	Public Teil des KSK der "Child"-Zone
NSEC	Nächster Eintrag	Nächster sicherer Eintrag in der Zone (als Ring: der letzte verweist auf den ersten)
NSEC3	Nächster Eintrag (gehasht)	Wie NSEC, nur sind die Namensbezeichner (mehrfach) gehasht

2. DNSSEC: Zone Signing Key

- Signiert alle Records
- Flag ID: 256
- Schwächerer Key: min. 2048 Bits RSA/SHA-256
- Aktuelle Empfehlung: ECDSA Curve P-256 mit SHA-256
- Automatischer Key Rollover
- Lebensdauer: 6-12 Monate

2. DNSSEC: Key Signing Key

- Signiert nur DNSKEYs (ZSKs und KSKs)
- Flag ID: 257
- Starker Key: min. 4096 Bit RSA/SHA-256
- Aktuelle Empfehlung: ECDSA Curve P-256 mit SHA-256
- Lebensdauer: 1-2 Jahre

2. DNSSEC: DNSKEY

- Öffentlicher KSK und ZSK
- Key-Tag zur Identifikation
- zugehöriger Algorithmus
- spezifische Flag-Werte

2. DNSSEC: DNSKEY

dig dnssec-uni-potsdam.de +dnssec +multiline +noall +answer DNSKEY

dnssec-uni-potsdam.de. 2057 IN DNSKEY 257 3 8 (

- AwEAAfN/i7XDXwk76MJB[...]

) ; KSK; alg = RSASHA256; key id = 30955

dnssec-uni-potsdam.de. 2057 IN DNSKEY 256 3 8 (

- AwEAAbEZYWsapkildIT4[...]

) ; ZSK; alg = RSASHA256; key id = 21681

2. DNSSEC: RRSIG

Resource Record Signature:

- Private Key signiert RRSet
- Mitgeliefert zum angefragten Record
- Resource Records eines Types werden als RR-Sets zusammengefasst

2. DNSSEC: DS

Delegation Signer:

- Hash vom DNSKEY (KSK)
- Authentifiziert DNSKEY einer Child Domain
- Wird in der Parent Zone eingetragen, daraus resultiert die Vertrauensstellung

2. DNSSEC: NSEC/NSEC3

Next SECure Entry:

- Nicht existente Einträge: NXDOMAIN
- Angreifer könnte Antworten unterdrücken
- Nicht vorhandene Hostnamen kryptographisch verifizieren
- NSEC3 hashed die Einträge mehrfach

2. DNSSEC: NSEC/NSEC3

aaa IN A 10.1.1.1 NSEC bbb

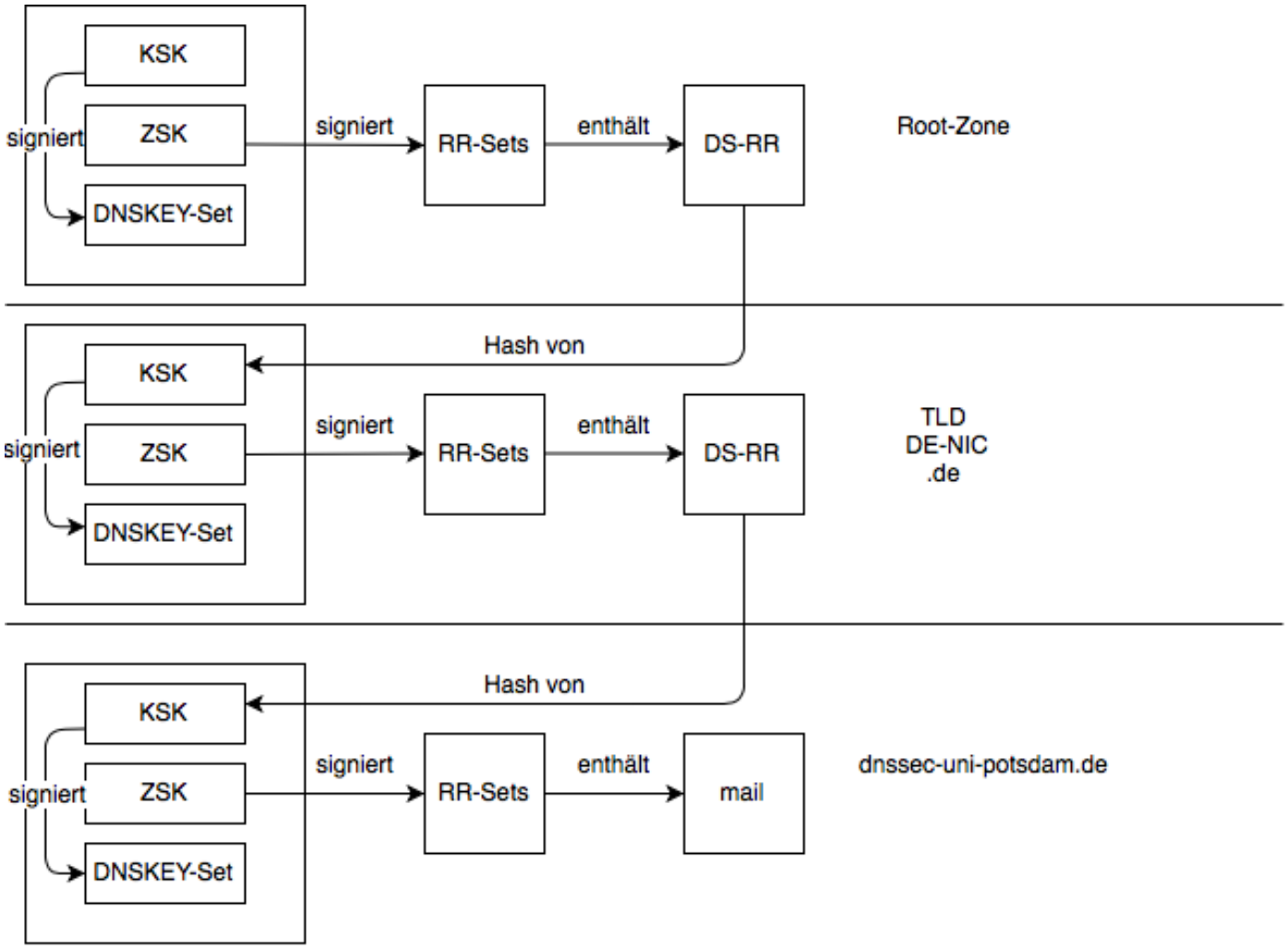
bbb IN A 10.1.1.2 NSEC ddd

ddd IN A 10.1.1.3 NSEC aaa

Question: dig ccc +dnssec +multiline

Answer: bbb IN NSEC ddd

2. DNSSEC


KSK
ZSK
DNSKEY-Set
signiert
signiert
RR-Sets
enthält
DS-RR
Root-Zone
Hash von
KSK
ZSK
DNSKEY-Set
signiert
signiert
RR-Sets
enthält
DS-RR
TLD
DE-NIC
.de
Hash von
KSK
ZSK
DNSKEY-Set
signiert
signiert
RR-Sets
enthält
mail
dnssec-uni-potsdam.de

2. DNSSEC

A-Record mit seiner digitalen Unterschrift:

mail.dnssec-uni-potsdam.de. 86400 IN A 141.89.59.180

mail.dnssec-uni-potsdam.de. 85992 IN RRSIG A 8 3 86400 20200226081901 20200127081112 21681 dnssec-uni-potsdam.de. ST2jMhz67y5V5vRxJPJJ[...]

2. DNSSEC

Mit dem Public Key (dnssec-uni-potsdam.de.) kann die digitale Unterschrift geprüft werden:

mail.dnssec-uni-potsdam.de. 86400 IN A 141.89.59.180

mail.dnssec-uni-potsdam.de. 85992 IN RRSIG [...]

dnssec-uni-potsdam.de. 85777 IN DNSKEY 256 3 8

AwEAAbEZYWsapkildIT4[...]

2. DNSSEC

Die nächsthöhere Ebene (de.) publiziert unseren Key und signiert ihn mit ihrem Key:

dnssec-uni-potsdam.de. 86400 IN DS 30955 8 2 0F3E5F57229C1693371[...]

dnssec-uni-potsdam.de. 86400 IN RRSIG DS 8 2 86400 20200218121150 20200204104150 15771 de. uh1rlTR+E6/+gXZAEH6H[...]

2. DNSSEC

Mit dem Public Key (de.) kann die digitale Unterschrift geprüft werden:

dnssec-uni-potsdam.de. 86400 IN DS 30955 8 2 [...]

dnssec-uni-potsdam.de. 86400 IN RRSIG DS 8 2 86400 [...]

de. 300 IN DNSKEY 256 3 8 cf77MuLY33[…]

3. Bind9: Schlüssel generieren

- Verzeichnis „keys“ erstellen:
  - mkdir /etc/bind/keys
  - chmod 775 /etc/bind/keys
  - chown root:bind /etc/bind/keys
- Keys generieren:
  - cd /etc/bind/keys
  - dnssec-keygen -3 -a ECDSAP256SHA256 -r /dev/random -f KSK dnssec-uni-potsdam.de
  - dnssec-keygen -3 -a ECDSAP256SHA256 -r /dev/random dnssec-uni-potsdam.de
- Berechtigung setzen:
  - chmod 644 *.key
  - chmod 600 *.private
  - chown bind:bind *

3. Bind9: DS Record generieren

- Herausfinden welcher der KSK ist:
  - cat *.key l grep “\-signing"
- DS-Record generieren:
  - dnssec-dsfromkey -a SHA-1 Kdnssec-uni-potsdam.de.+008+27511.key >> /etc/bind/keys/dsset-dnssec-uni-potsdam.de
  - dnssec-dsfromkey -a SHA-256 Kdnssec-uni-potsdam.de.+008+27511.key >> /etc/bind/keys/dsset-dnssec-uni-potsdam.de

3. BIND9: Generate TLSA Record

https://www.huque.com/bin/gen_tlsa

3. BIND9: named.conf.options

```
options {
    // When performing dynamic update of secure zones, the directory
    // where the public and private DNSSEC key files should be found.
    key-directory "/etc/bind/keys";
    // To enable named to validate answers from other servers,
    // the dnssec-enable option must be set to yes, and the
    // dnssec-validation options must be set to yes or auto.
    dnssec-enable yes;
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.
    // See https://www.isc.org/bind-keys.
    dnssec-validation auto;
```

3. Bind9: named.conf.public-zones

```
// Domain: dnssec-uni-potsdam.de
zone "dnssec-uni-potsdam.de" {
    type master;
    update-policy local;
    file "/etc/bind/zones/db.dnssec-uni-potsdam.de";
    auto-dnssec maintain;
    inline-signing yes;
};
```

3. Bind9: Signieren der Zone

Beim erstmaligen Laden einer Zonendatei mit vorhandenen DNSSEC-Schlüsseln, wird die Datei automatisch signiert:

- rndc*loadkeys dnssec-uni-potsdam.de
- rndc* reload dnssec-uni-potsdam.de

*__rndc__ stands for Remote Name Daemon Control

3. Bind9: Signieren der Zone

Die automatisch generierten (signierten) Zonendateien und die Journaldateien sind nicht mehr direkt lesbar:

- named-checkzone -D -j -s relative -f raw dnssec-uni-potsdam.de /etc/bind/zones/db.dnssec-uni-potsdam.de.signed
- named-journalprint /etc/bind/zones/db.dnssec-uni-potsdam.de.signed.jnl

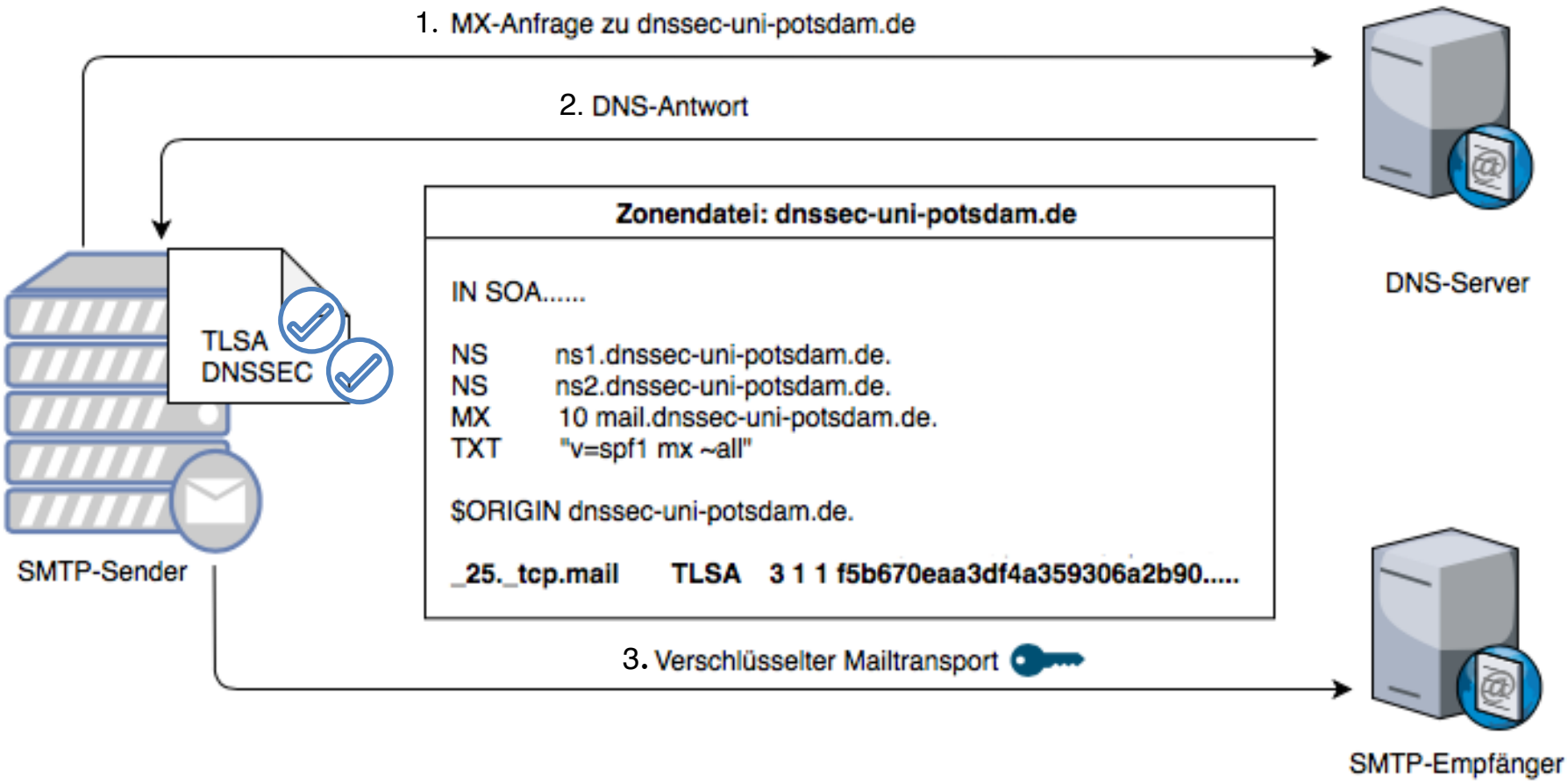
3. Bind9: Änderungen der Zone

- DNS Updates:
  - rndc freeze
  - rndc sync -clean
- Bei Änderungen an den Zonen, muss Seriennummer angepasst und Konfiguration überprüft werden:
  - named-checkconf -z
- Zonendatei neu laden:
  - rndc reload
- DNS-Updates fortsetzen:
  - rndc thaw

4. Postfix: main.cf

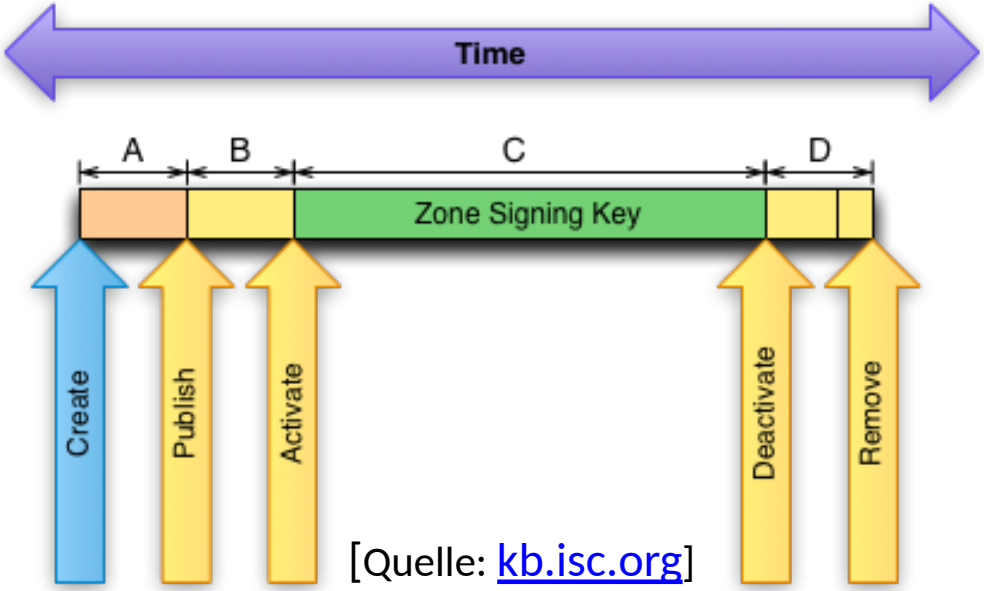
- DANE ausgehend mit Postfix einrichten:
  - smtp_tls_security_level = dane
  - smtp_dns_support_level = dnssec
- DANE eingehend mit Postfix einrichten:
  - smtpd_tls_security_level = may
  - smtpd_tls_security_level = encrypt (nur TLS erlauben)

4. Postfix: Mailtransport



5. Key Rollover: Key Timer

- A: Schlüssel ist erstellt
- B: Schlüssel in Zone veröffentlicht (TTL DNSKEY)
- C: Schlüssel in ZONE und zum Signieren verwendet
- D: Schlüssel noch in der Zone, jedoch inaktiv
- E: Schlüssel wird aus der Zone entfernt



[Quelle: kb.isc.org]

5. Key-Rollover: Key Metadaten

Filenames: Kzone+alg+tag.{key|private}

cat Kdnssec-uni-potsdam.de.+008+21681.key

; This is a zone-signing key, keyid 21681, for dnssec-uni-potsdam.de.

; Created: 20200108131355 (Wed Jan 8 14:13:55 2020)

; Publish: 20200109235958 (Fri Jan 10 00:59:58 2020)

; Activate: 20200110000000 (Fri Jan 10 01:00:00 2020)

; Inactive:

; Delete:

dnssec-uni-potsdam.de. IN DNSKEY 256 3 8 AbEZYWsaildIT4[...]

5. Key-Rollover: ZSK

Pre-Publish Methode:

- Neuen ZSK in Zonendatei veröffentlichen, bevor er verwendet wird
- Mindestens eine TTL abwarten
- Den alten Schlüssel nicht mehr verwenden und neuen RRSIG generieren
- Mindestens eine weitere TTL abwarten
- Den alten Schlüssel entfernen

5. Key Rollover: Alter ZSK

root@mail: /etc/bind/keys

dnssec-settime -I 20200110 -D 20200112 kdnssec-uni-potsdam.de.+008+06924.key

```
root@mail:.../bind/keys.SV # cat Kdnssec-uni-
potsdam.de.+008+06924.key
; This is a zone-signing key, keyid 6924, for dnssec-
uni-potsdam.de.
; Created: 20190913071920 (Fri Sep 13 09:19:20 2019)
; Publish: 20190913071920 (Fri Sep 13 09:19:20 2019)
; Activate: 20190913071920 (Fri Sep 13 09:19:20 2019)
; Inactive: 20200110000000 (Fri Jan 10 01:00:00 2020)
; Delete: 20200112000000 (Sun Jan 12 01:00:00 2020)
dnssec-uni-potsdam.de. IN DNSKEY 256 3 8
AwEAAdcHLz0owAalyUqwPCijIW1vdd6oWwT7Oo42J8bXt+PLRdnqyG
QU Q4CGO30UEgyA3CpErLy0ZHugM0j9l/
phqKAGzHHjHWyKbZVqNSHDbG2V nyO6cQmXj…
```

- -I 20200110: ZSK ist ab 10.01. in der Zone, aber Inaktiv
- -D 20200112: Schlüssel kann ab 12.01. gelöscht werden

5. Key Rollover: Neuer ZSK

root@mail: cd /etc/bind/keys

dnssec-keygen -S Kdnssec-uni-potsdam.de.+008+06924.key -i 2 d

Neuen ZSK generieren

```
root@mail:.../bind/keys.SV # cat Kdnssec-uni-potsdam.de.+008+21681.key
; This is a zone-signing key, keyid 21681, for dnssec-uni-potsdam.de.
; Created: 20200108131355 (Wed Jan  8 14:13:55 2020)
; Publish: 20200109235958 (Fri Jan 10 00:59:58 2020)
; Activate: 20200110000000 (Fri Jan 10 01:00:00 2020)
dnssec-uni-potsdam.de. IN DNSKEY 256 3 8
AwEAAbEZYWsapkildIT4O7JLgTqXrvxyPtU+pT3YH8J2ri2IUS9/m6z2
cCPSG5bZ6lMzsA1h6zW92gLV4qrOyXx7beXzSaRV3T2TXfKdib/WgNAn
2SioaqC2zYXuR65UygpknPIirfesoN66LrPK2ow6RM7XJ2hSnO8h3aTl
skuSr0Cg3NNXiEcWBME5801e297ODMpqALV8fcI6aJU44PuWfzh+5Ca8
srLncq0DloDAHGbhi9Is5enBjY2G8F1rq6yuCalJy0mv5OD9Ir8yNQPu
mUagQOF8LdiIXXaT7PFv7nLjzfL+KfHMmKAFOBPDXfIz82sx1oirUGoP..
```

Rechte setzen: chmod g+r *

chown root:bind /etc/bind/keys

ZSK in die Zone laden: rndc loadkeys dnssec-uni-potsdam.de

5. Key-Rollover: ZSK

- Herausfinden welcher der ZSK ist:
  - dig +noall +answer +noclass +nocrypto +dnssec +multiline DNSKEY dnssec-uni-potsdam.de
- Datum des vorhandenen Schlüssels ändern:
  - cd /etc/bind/keys
  - dnssec-settime -I +2d -D +4d Kdnssec-uni-potsdam.de.+008+06924.key
- Schlüssel generieren:
  - cd /etc/bind/keys
  - dnssec-keygen -i 2d -3 -a ECDSAP256SHA256 -r /dev/random dnssec-uni-potsdam.de
- Berechtigung setzen:
  - chmod 644 /etc/bind/keys/*.key
  - chmod 600 /etc/bind/keys/*.private
  - chown bind:bind /etc/bind/keys/*
- Schlüssel für die Zone neu laden:
  - rndc loadkeys dnssec-uni-potsdam.de

5. Key-Rollover: KSK

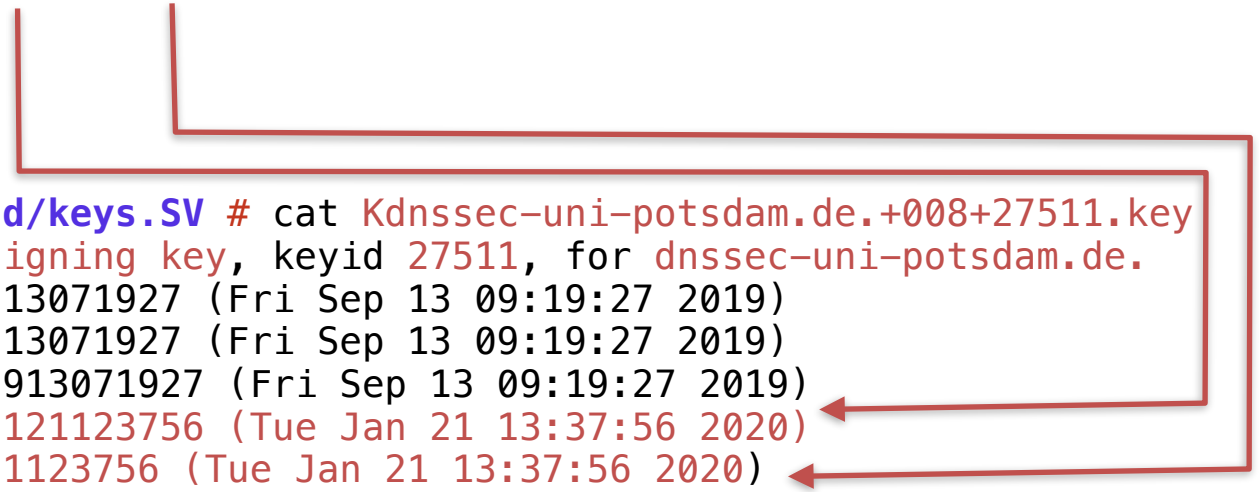
Double Signature Methode:

- KSK Rollover mit der Double Signature Methode
- Den neuen KSK und neue RRSIG veröffentlichen, die Zone ist nun doppelt so groß
- Mindestens eine TTL abwarten
- Den alten KSK entfernen

5. Key Rollover: Alter KSK

root@mail: cd /etc/bind/keys

dnssec-settime -I +1w -D +1w kdnssec-uni-potsdam.de.+008+27511.key



```
root@mail:.../bind/keys.SV # cat Kdnssec-uni-potsdam.de.+008+27511.key
; This is a key-signing key, keyid 27511, for dnssec-uni-potsdam.de.
; Created: 20190913071927 (Fri Sep 13 09:19:27 2019)
; Publish: 20190913071927 (Fri Sep 13 09:19:27 2019)
; Activate: 20190913071927 (Fri Sep 13 09:19:27 2019)
; Inactive: 20200121123756 (Tue Jan 21 13:37:56 2020)
; Delete: 20200121123756 (Tue Jan 21 13:37:56 2020)
dnssec-uni-potsdam.de. IN DNSKEY 257 3 8 AwEAAehgMIZIdWY8P2wGFNk1l7R0FdVIwtx9uT/
TAmTXJUQN7TLCyLum ekF2K3Wc/3PJFjs2iT8tcIBA4hihXCkcoHsnod+kyiU/I9H3CUQtopqV
PIc732LQvzPKDPLEBs3Je74Sqh1Vlhq7odWf43ofWVDlaZd9RwbPJP4c tXpSodBCzYu2N/GbL/
x8QRMVOTGh5KOmp03DGtPZTBpo7sWtD+MYD/s8
6GJwjows0tay8jjUojt64IHwGrIBWmNYSZyPHrQ8N9sm8wTJbpO5HJ7z
Sr9UxGlVh8aOZHvv6DDeFSGxbX53BalTLAh/ts1nRsl5K263EXyGmbwe 0mGe0irswP0=
```

5. Key Rollover: Neuer KSK

root@mail: cd /etc/bind/keys

dnssec-keygen -a RSASHA256 -b 2048 -f KSK -n ZONE dnssec-uni-potsdam.de

```
root@mail:.../bind/keys.SV # cat Kdnssec-uni-potsdam.de.+008+30955.key
; This is a key-signing key, keyid 30955, for dnssec-uni-potsdam.de.
; Created: 20200114120713 (Tue Jan 14 13:07:13 2020)
; Publish: 20200114120713 (Tue Jan 14 13:07:13 2020)
; Activate: 20200114120713 (Tue Jan 14 13:07:13 2020)
dnssec-uni-potsdam.de. IN DNSKEY 257 3 8 AwEAAfN/
i7XDXwk76MJBJvTpNWjvJyWqIE2dT4KqzDEc9822D7ucugAz vcFsUwJcW6BXLIhviI/
yjrvCvuHBj++6crTfchTnvZmf6GqNdLpJGlp2 TGxC+45hX+Xr7eeHk/
sxfThDSyNhdxQhyeKKJ5SMNHWAZFlaqy4qpucG SvKKoFe7yIEHGVRfOkbkRW/
QpMotBUScACZANCj2CXilkdclr7uaOYol 5rhBQU4FXtiKnKk1u8pO/
hW4zejBUNdZLWiZ7aUi6w2hPgpoLu562Bdc N+soVguSE3uEAizXxs+UkD/
dsTQg0N8E7tx0pGDzKpFMi6i23+vEpjue jm+wlXag+4M=
```

5. Key-Rollover: KSK

- Herausfinden welcher der KSK ist:
  - dig +noall +answer +noclass +nocrypto +dnssec +multiline DNSKEY dnssec-uni-potsdam.de
- Datum des vorhandenen Schlüssels ändern:
  - dnssec-settime -I +1w -D +1w Kdnssec-uni-potsdam.de.+008+27511.key
- Schlüssel generieren:
  - cd /etc/bind/keys
  - dnssec-keygen -3 -a ECDSAP256SHA256 -r /dev/random -f KSK dnssec-uni-potsdam.de
- DS-Record vom KSK generieren:
  - dnssec-dsfromkey -a SHA-1 Kdnssec-uni-potsdam.de.+008+30955.key >> /etc/bind/keys/dsset-dnssec-uni-potsdam.de
  - dnssec-dsfromkey -a SHA-256 Kdnssec-uni-potsdam.de.+008+30955.key >> /etc/bind/keys/dsset-dnssec-uni-potsdam.de
- Berechtigungen setzen:
  - chmod 644 /etc/bind/keys/*.key
  - chmod 600 /etc/bind/keys/*.private
  - chown bind:bind /etc/bind/keys/*
- Schlüssel für die Zone neu laden:
  - rndc loadkeys dnssec-uni-potsdam.de

6. Key Management

- Bind 9.11
- Phython-basierte Wrapper kombiniert „dnssec-keygen“ und „dnssec-settime”
- Automatisierung umfasst allgemeine/ zonenspezifische Richtlinien in /etc/dnssec-policy.conf
- dnssec-keymgr -c /etc/dnssec-policy.conf

6. Key Management

```
policy default {
  algorithm RSASHA256;
  directory "/etc/bind/keys";
  key-size zsk 2048;
  key-size ksk 4096;
  roll-period zsk 6mo;
 # roll-period ksk 2y;
  pre-publish zsk 1mo;
  pre-publish ksk 1mo;
  post-publish zsk 1mo;
  post-publish ksk 1mo;
  coverage 1y;
};
zone dnssec-uni-potsdam.de {
  policy default;
  algorithm ECDSAP256SHA256;
};
```

6. Key Management

Mit einem Cron-Skript wird dnssec-keymgr regelmässig aufgerufen.

/etc/cron.daily/dnssec-keymgr:

```
#!/bin/sh

chown bind:bind /etc/bind/keys/*

runuser -u bind -- /usr/sbin/dnssec-keymgr -c \
/etc/dnssec-policy.conf
```

6. Key Management

Mit dnssec-coverage wird überprüft, ob für die DNSSEC-Schlüssel, für eine bestimmte Zone Timing-Metadaten festgelegt wurden

dnssec-coverage -d 1d -m 1d -K /etc/bind/keys/ dnssec-uni-potsdam.de

7. Ausblick

- Ab BIND 9.11
  - dnssec-keymgr
  - Manuelles Hochladen des KSK notwendig
- Ab BIND 9.12
  - CDS- und CDNSKEY-Records eingeführt
  - Kein manuelles Hochladen notwendig

7. Ausblick

BIND 9.12: CDS/CDNSKEY:

- Automatisches Registrieren in der übergeordneten Zone
- Neues Schlüsselmaterial über DNSSEC kann in-Band in übergeordente Zone übertragen werden
- Regelmäßiges durchsuchen des Betreibers der übergeordneten Zone

8. Fazit

- Mangelhaft Sicherheit mit Interaktion der Provider
- Statt CAs kann der DNS befragt werden
- Über die Prüfsumme kann das Ziel zweifelsfrei identifiziert werden
- Über DNSSEC gesicherte DNS-Abfrage wird geprüft, ob sein Gegenüber TLS unterstützt
- Bietet der Empfänger trotz TLSA-Record kein STARTTLS, bricht Ihr Mailserver sofort ab (Downgrade-Attacke)

Tools

- https://dnsviz.net/
- https://dnslookup.org/
- https://dnssec-debugger.verisignlabs.com/
- https://dane-test.had.dnsops.gov/server/dane_check.cgi
- dig dnssec-uni-potsdam.de +dnssec +multiline
- posttls-finger mail.dnssec-uni-potsdam.de
- **Webseite:** https://www.cs.uni-potsdam.de/bs/research/projectSecurity.html#dnssec

Questions and Answers

Quellen

- https://kb.isc.org/docs/aa-00822, (Zuletzt Aufgerufen 05.03.2020)
- doku.lrz.de, (Zuletzt Aufgerufen 05.03.2020)
- G. Haslinger, Sicherer E-Mail-Dienste-Anbieter, https://hitco.at/blog/wp-content/uploads/Sicherer-E-Mail-Dienste-Anbieter-DNSSecDANE-HowTo-2016-04-28.pdf, (Zuletzt aufgerufen 03.03.2020)
- DFN-CERT, Empfehlungen für den Einsatz von Transportverschlüsselung zwischen Mailservern, www.dfn-cert.de/dokumente/smtp-transportverschluesselung.pdf, (Zuletzt aufgerufen 05.03.2020)
- https://arstechnica.com/information-technology/2020/03/lets-encrypt-revoking-https-certs-due-to-certificate-authority-bug/, (Zuletzt aufgerufen 05.03.2020)