DNSSEC und DANE Einführungskurs

Sven Duscha (Leibniz Rechenzentrum)

Zeitplan

Mittwoch, 8.März

- 12:30-13:30 Mittagspause
- 15:00-15:20 Kaffeepause
- ~16:00 Ende des ersten Tages

- Abend: Kurs-Dinner

Donnerstag, 9.März

- 10:15-10:35 Kaffeepause
- 12:00-12:30 Kaffeepause
- 13:30 Ende / ggf. gemeinsames Mittagessen

DNSSEC/DANE Kurs am RRZE

Kurze Vorstellungsrunde

- Name, Universität / Hochschule
- Vorwissen über Kryptographie?
- Kenntnisse und Erfahrung mit DNS, DNSSEC?
- Was erwarten Sie sich von diesem Kurs?

Inhaltsübersicht

DNSSEC

- DNS Funktionsweise und Schwachpunkte
- Public Key Kryptographie Grundlagen
- DNSSEC Records und Zusammenhänge
- DNSSEC-Konfiguration am Beispiel BIND-9.9
- DNSSEC in der Praxis
- Zusammenfassung DNSSEC

Inhaltsübersicht

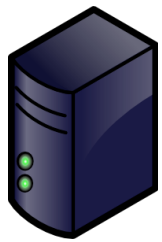
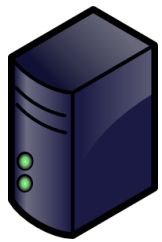
DANE

- DANE - Domain name-based authenticated named entity
- DANE Funktionsweise
- Anwendungen von DANE
- Beispiel Mailserver-Authentizität garantieren
- Zusammenfassung DANE


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
DNS - Funktionsweise und Schwachpunkte

DNS Funktionsweise

- DNS - Domain name system ordnet die IP Adressen Domainnamen zu (und umgekehrt)
- DNS ist dezentral, keine zentrale Datenbank
- Jeder Nameserver verwaltet seine Zone
- Abfragen durchlaufen hierarchisch den Domain tree

129.187.4.40

2001:4ca0:0:305::40

root NS → TLD NS → Domain NS → IP des Servers

- Autoritative und Resolving Nameserver
- Master und Slave Nameserver

DNS Abfrage-Beispiel: confluence.lrz.de

DNS Abfrage-Beispiel: confluence.lrz.de

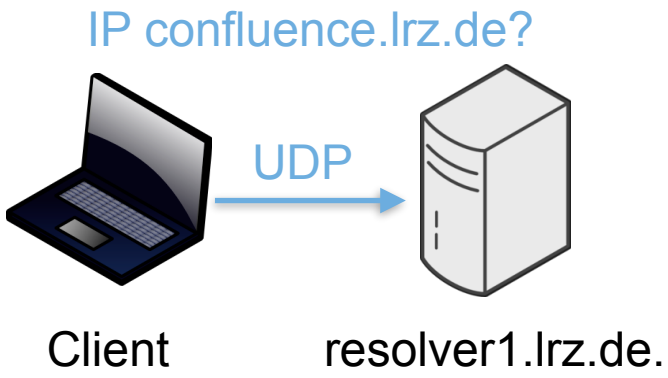
IP confluence.lrz.de?

1. Benutzer will auf: confluence.lrz.de

Client

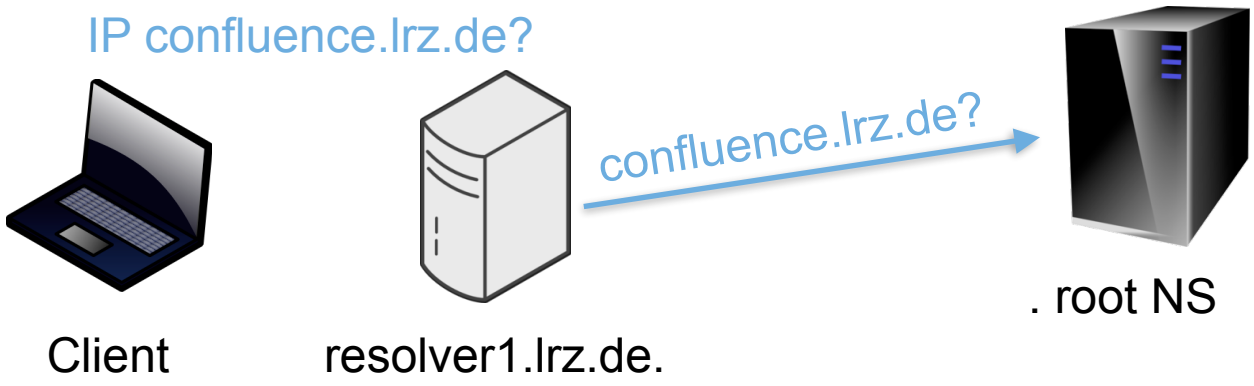
DNS Abfrage-Beispiel: confluence.lrz.de

1. Benutzer will auf: confluence.lrz.de
2. Browser fragt 10.156.33.53, resolver1.lrz.de



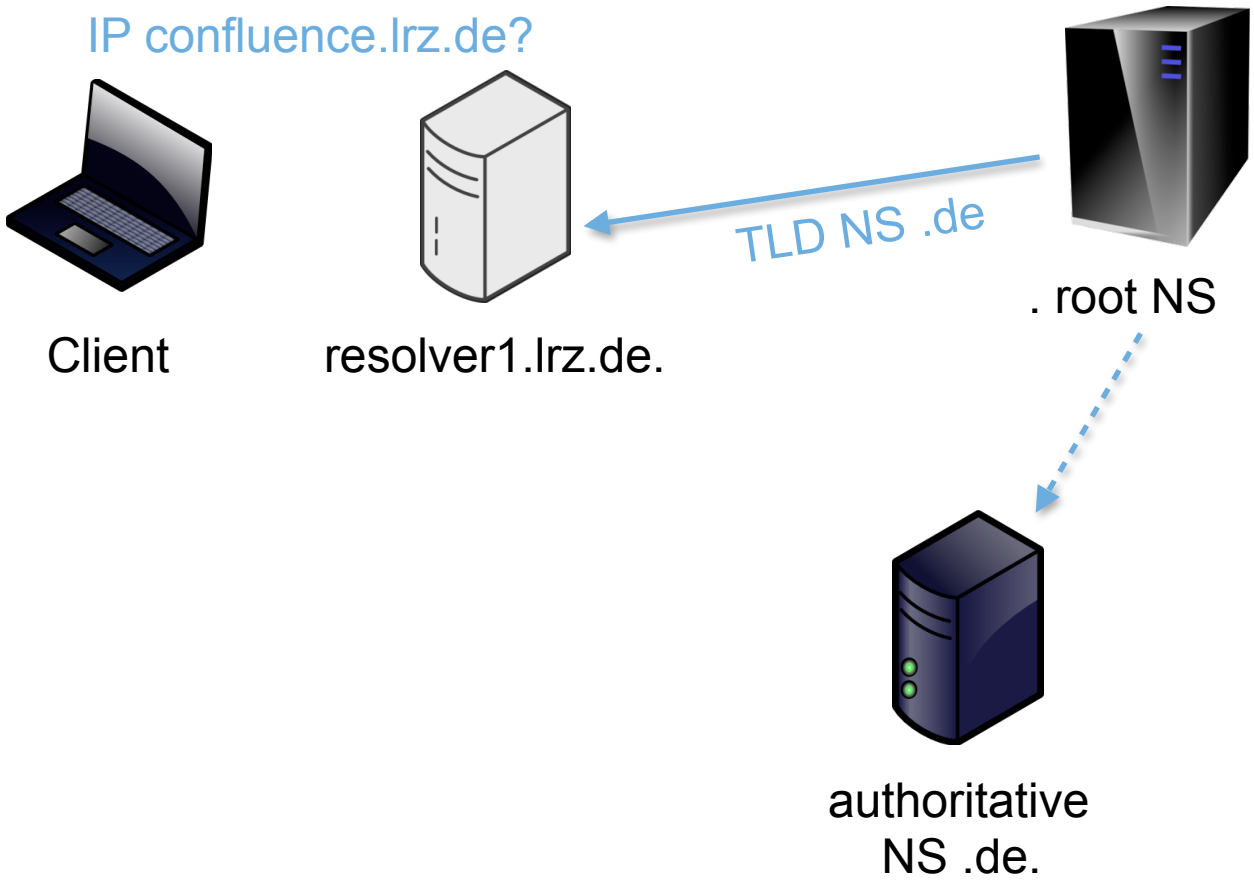
DNS Abfrage-Beispiel: confluence.lrz.de

1. Benutzer will auf: confluence.lrz.de
2. Browser fragt 10.156.33.53, resolver1.lrz.de
3. Resolving DNS fragt root Nameserver



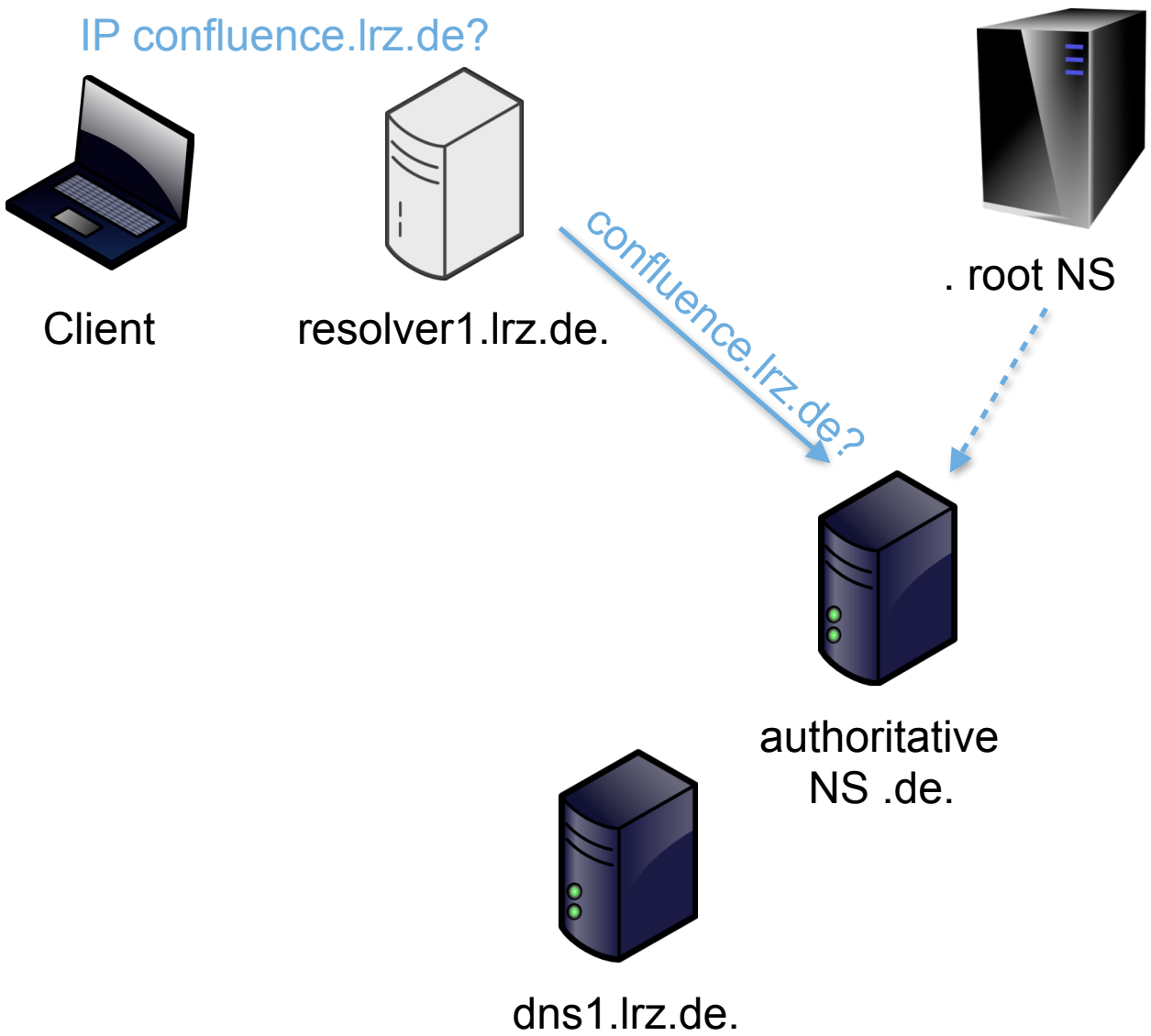
DNS Abfrage-Beispiel: confluence.lrz.de

1. Benutzer will auf: confluence.lrz.de
2. Browser fragt 10.156.33.53, resolver1.lrz.de
3. Resolving DNS fragt root Nameserver
4. Verweis auf .de. TLD authoritative Nameserver



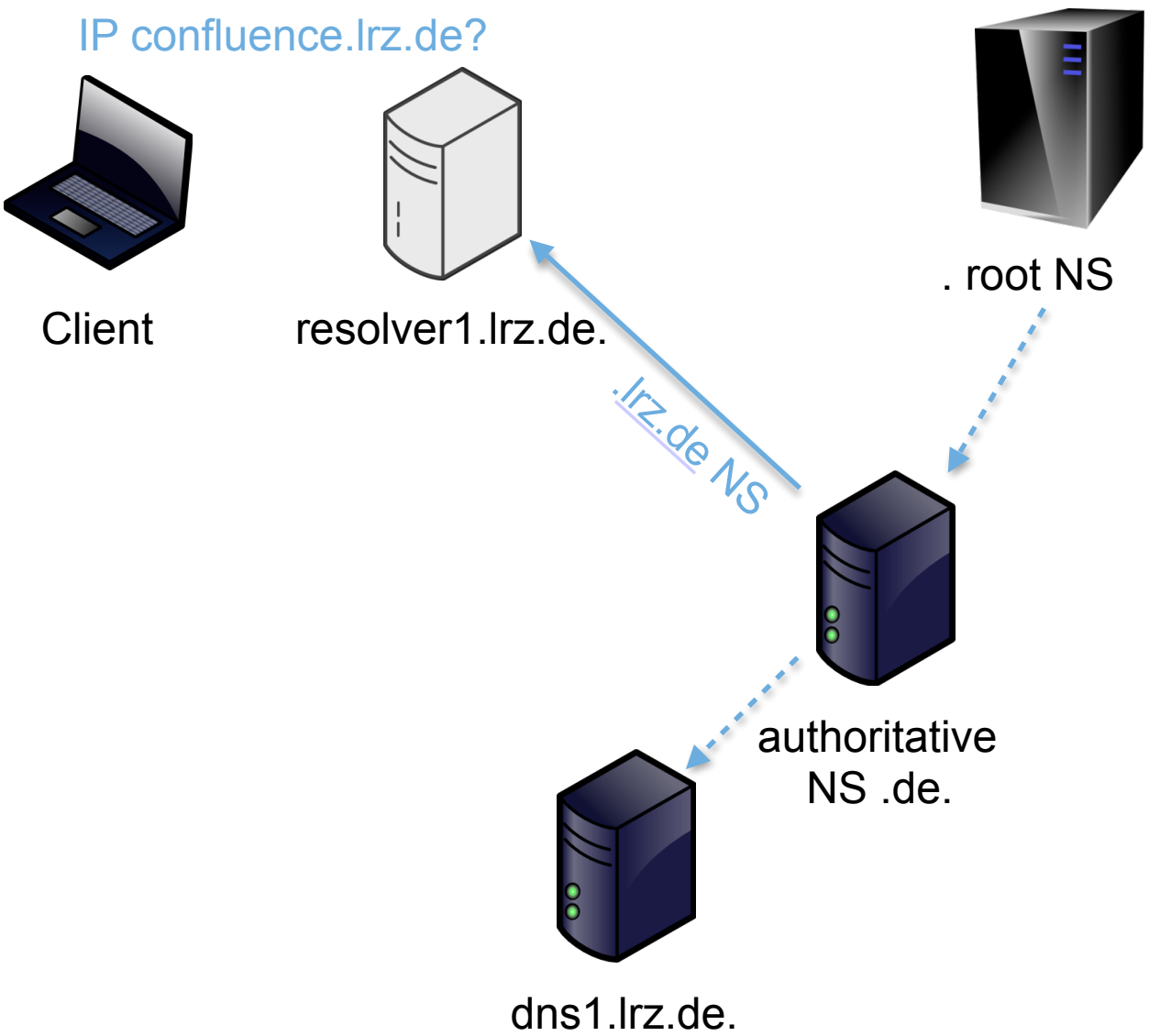
DNS Abfrage-Beispiel: confluence.lrz.de

1. Benutzer will auf: confluence.lrz.de
2. Browser fragt 10.156.33.53, resolver1.lrz.de
3. Resolving DNS fragt root Nameserver
4. Verweis auf .de. TLD authoritative Nameserver
5. Resolver frage .de. TLD NS



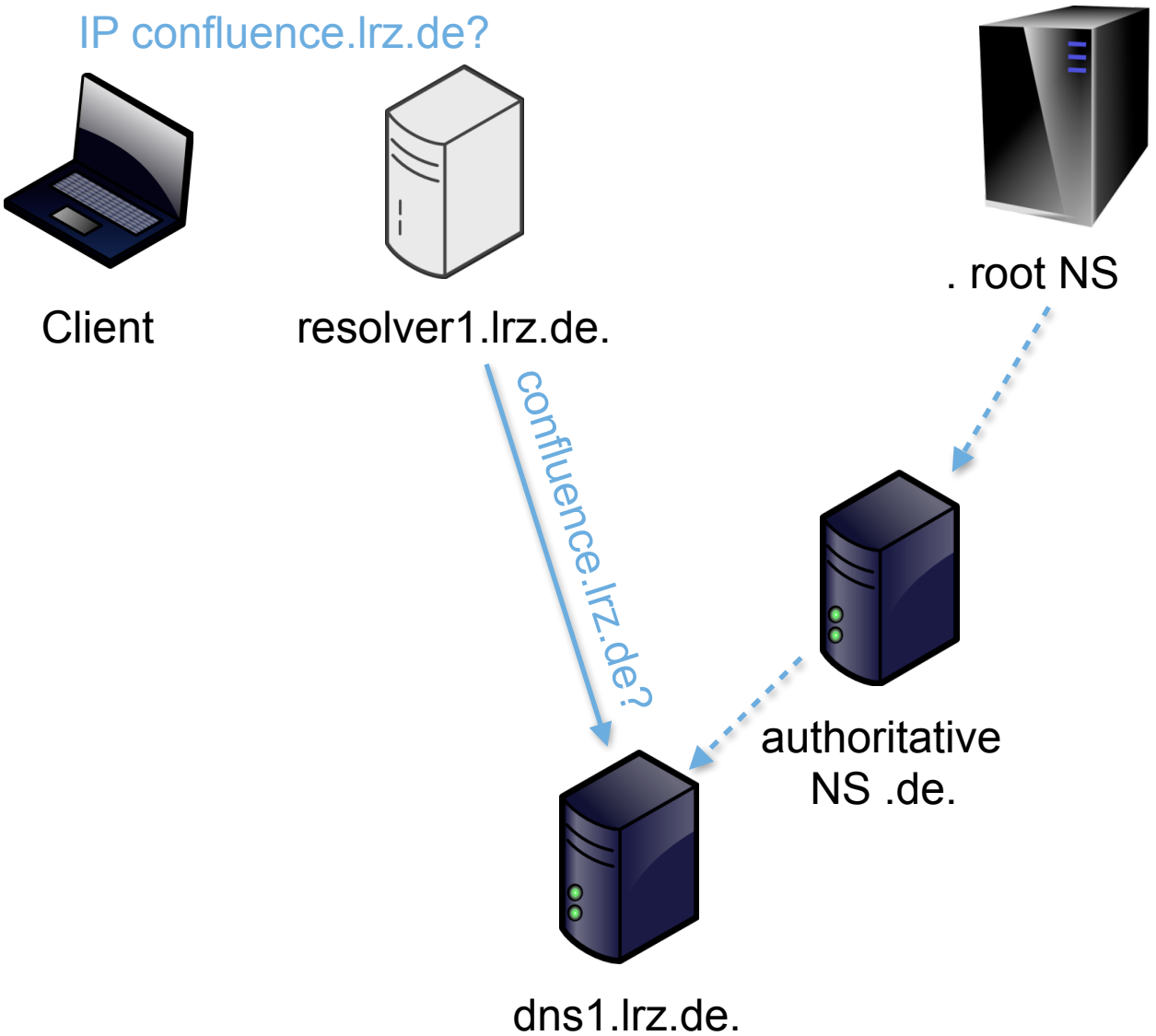
DNS Abfrage-Beispiel: confluence.lrz.de

1. Benutzer will auf: confluence.lrz.de
2. Browser fragt 10.156.33.53, resolver1.lrz.de
3. Resolving DNS fragt root Nameserver
4. Verweis auf .de. TLD authoritative Nameserver
5. Resolver frage .de. TLD NS
6. Verweis auf LRZ NS dns1.lrz.de



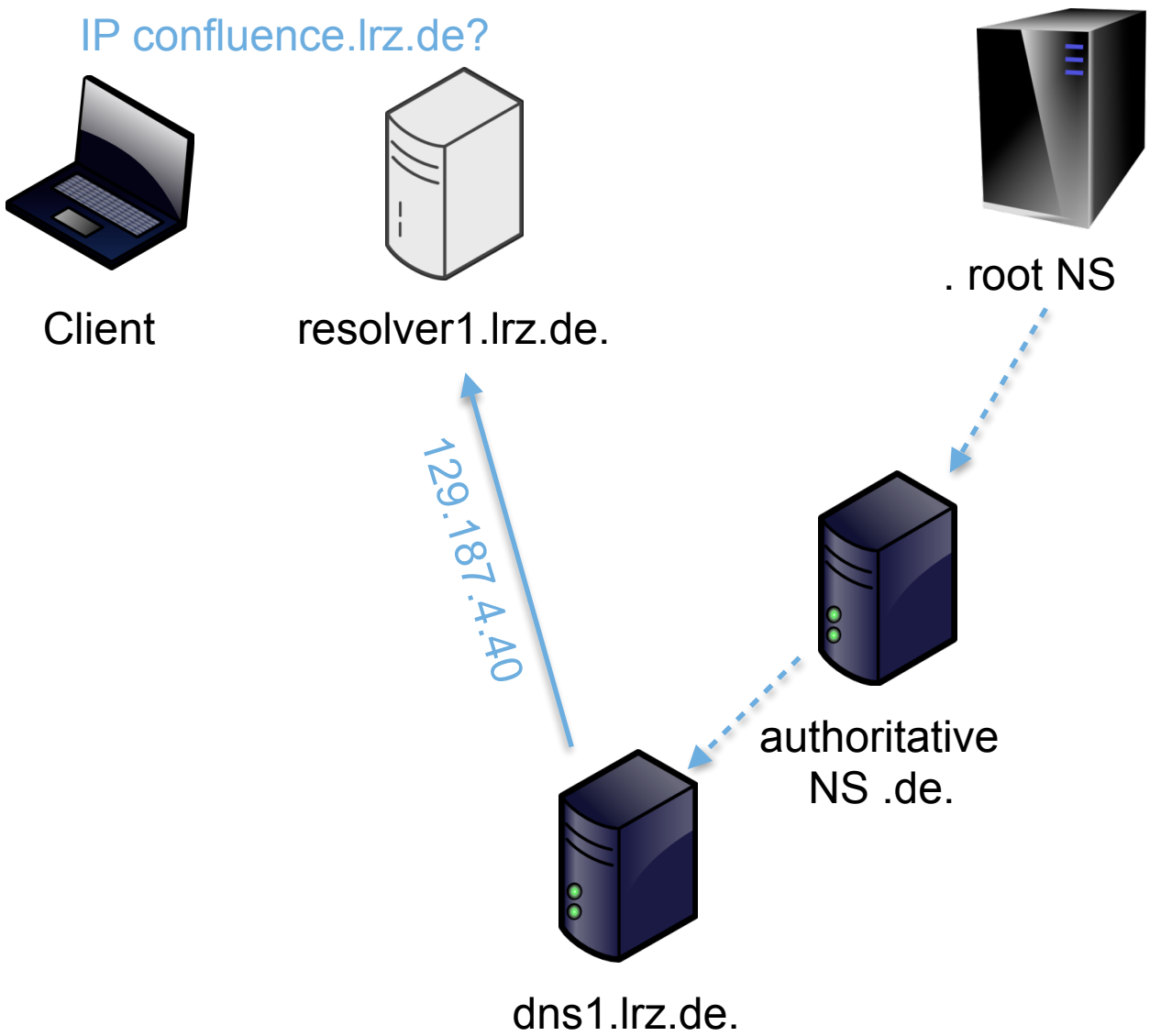
DNS Abfrage-Beispiel: confluence.lrz.de

1. Benutzer will auf: confluence.lrz.de
2. Browser fragt 10.156.33.53, resolver1.lrz.de
3. Resolving DNS fragt root Nameserver
4. Verweis auf .de. TLD authoritative Nameserver
5. Resolver frage .de. TLD NS
6. Verweis auf LRZ NS dns1.lrz.de
7. Resolver fragt dns1.lrz.de.



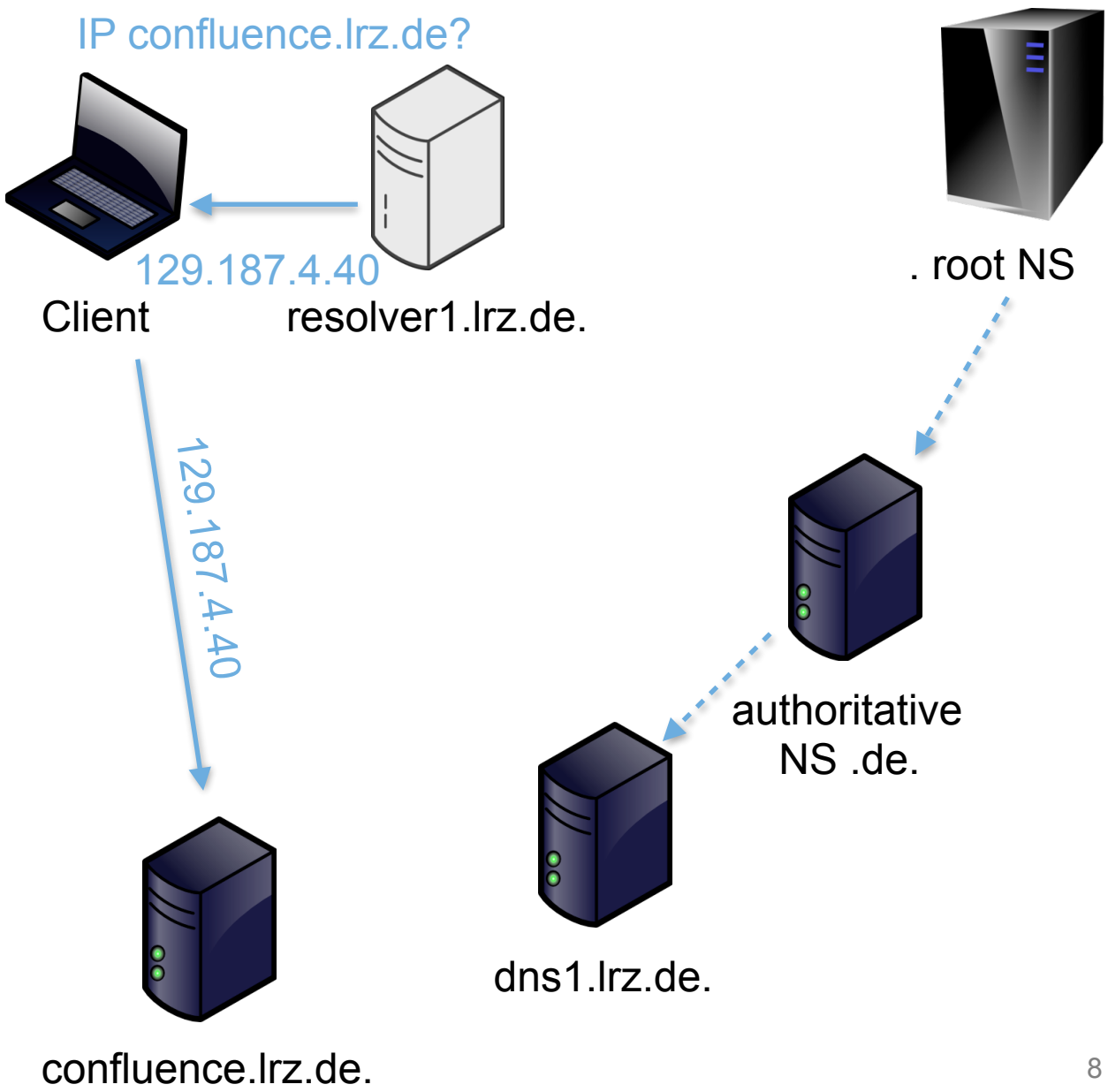
DNS Abfrage-Beispiel: confluence.lrz.de

1. Benutzer will auf: confluence.lrz.de
2. Browser fragt 10.156.33.53, resolver1.lrz.de
3. Resolving DNS fragt root Nameserver
4. Verweis auf .de. TLD authoritative Nameserver
5. Resolver frage .de. TLD NS
6. Verweis auf LRZ NS dns1.lrz.de
7. Resolver fragt dns1.lrz.de.
8. IP zu confluence.lrz.de.
   129.187.4.40
   2001:4ca0:0:305::40



DNS Abfrage-Beispiel: confluence.lrz.de

1. Benutzer will auf: confluence.lrz.de
2. Browser fragt 10.156.33.53, resolver1.lrz.de
3. Resolving DNS fragt root Nameserver
4. Verweis auf .de. TLD authoritative Nameserver
5. Resolver frage .de. TLD NS
6. Verweis auf LRZ NS dns1.lrz.de
7. Resolver fragt dns1.lrz.de.
8. IP zu confluence.lrz.de.
   129.187.4.40
   2001:4ca0:0:305::40
9. IP Antwort an Client



DNS Nameserver Zusammenspiel

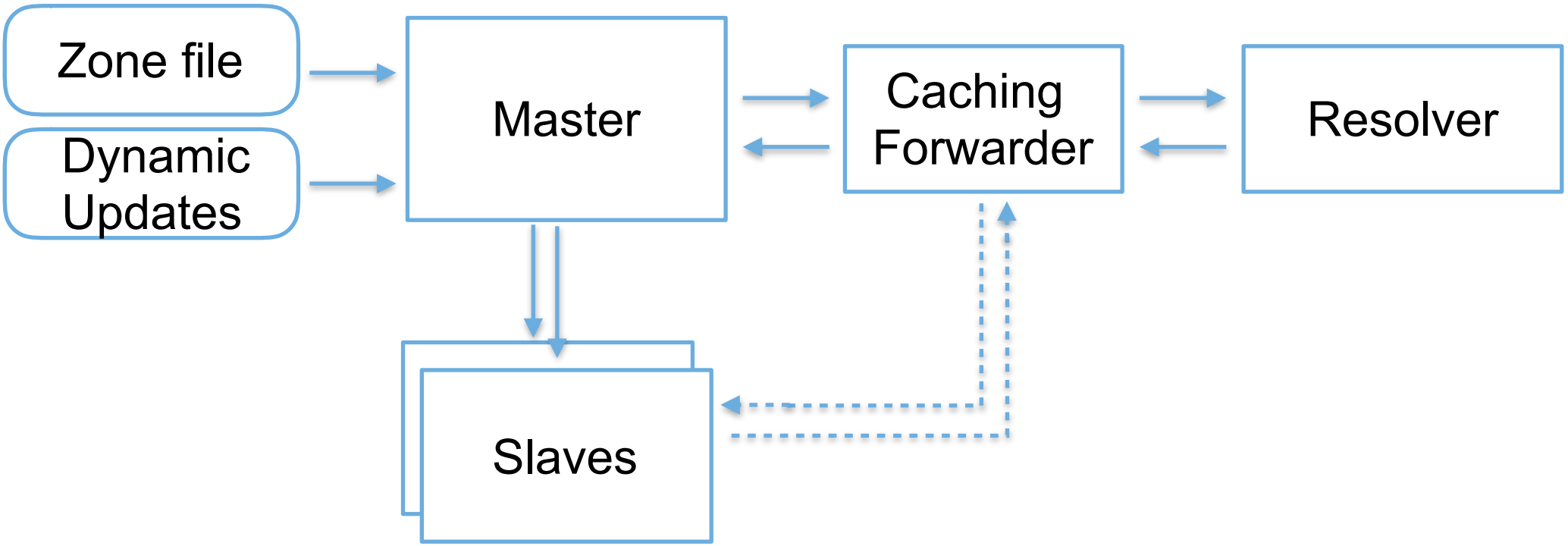


Bild: RIPE NCC

DNS - Grundsätzliche Probleme

- DNS ist plain text
- UDP keine sessions
- Baumstruktur mit Weiterreichen der Verantwortlichkeit

  → jede Instanz ist für ihre Zone zuständig
- Basiert auf gegenseitigem Vertrauen
- Resolving Nameserver sind Opfer von Fehlern, Übernahmen und Angriffen

DNS Nameserver - Angriffspunkte

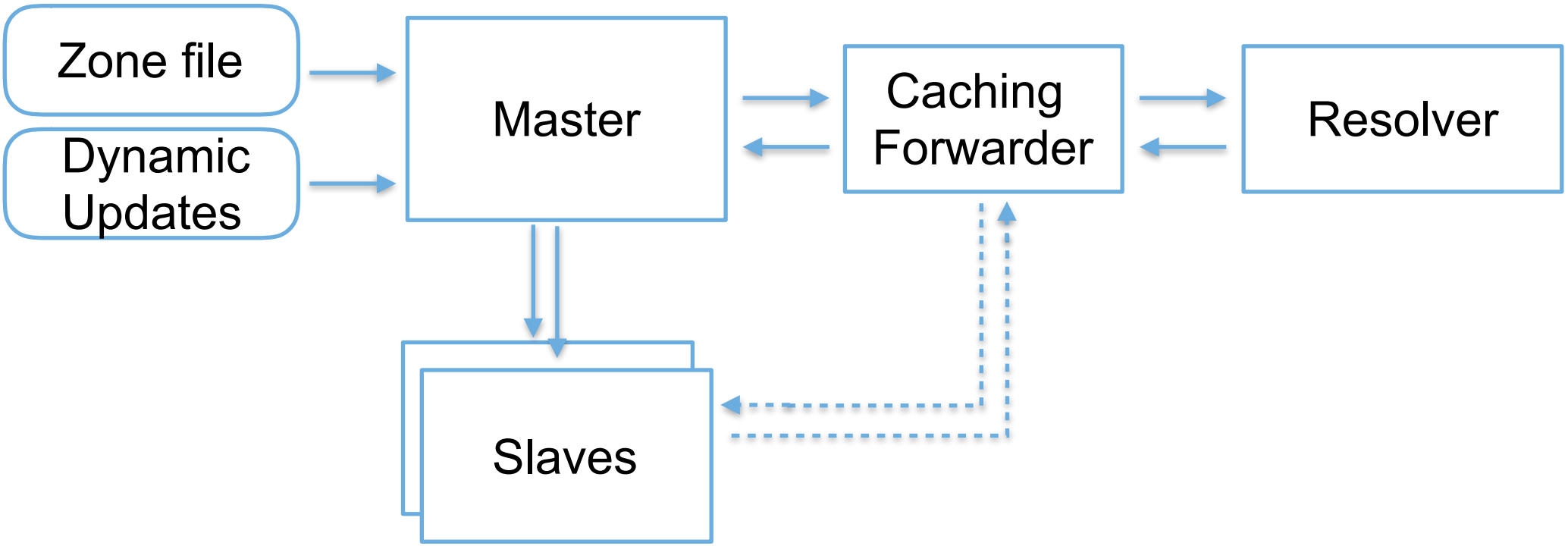


Bild: RIPE NCC

DNS Nameserver - Angriffspunkte

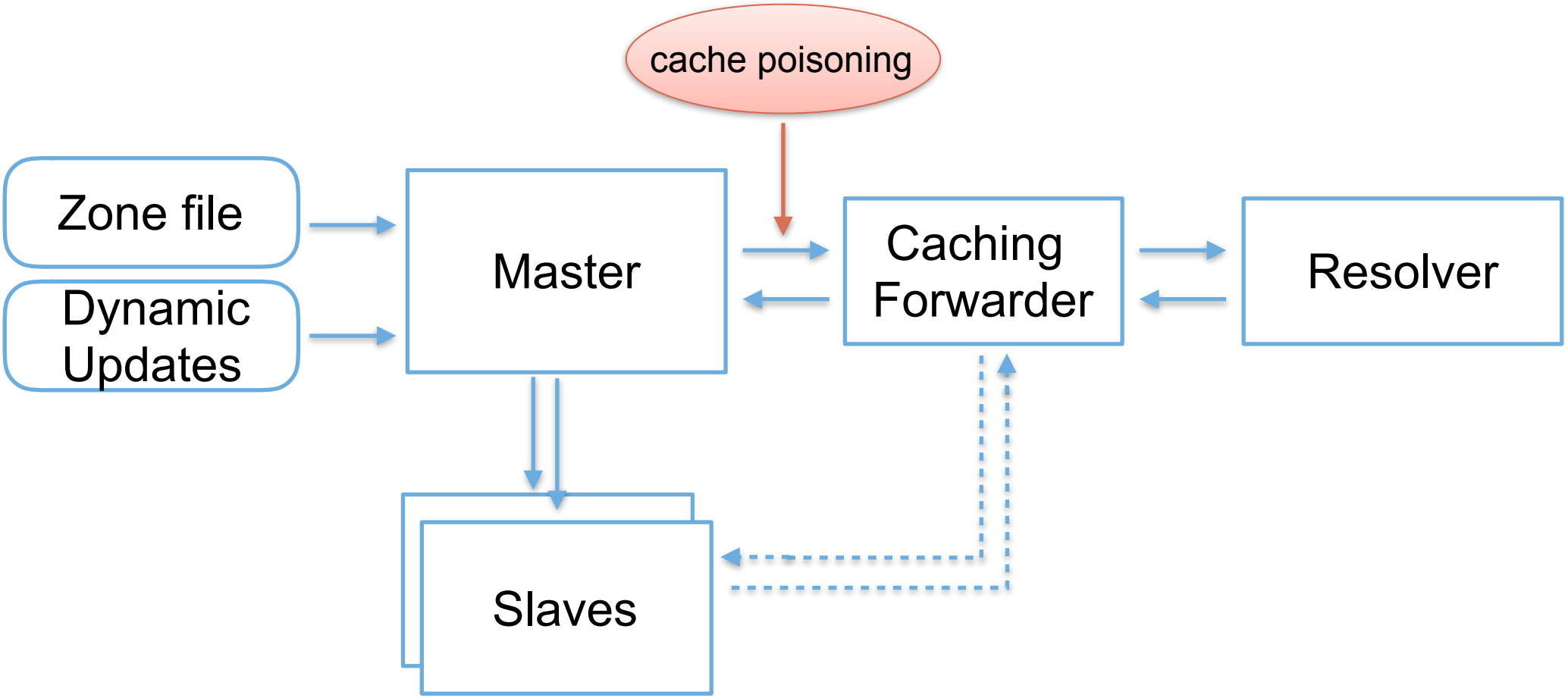


Bild: RIPE NCC

DNS Nameserver - Angriffspunkte

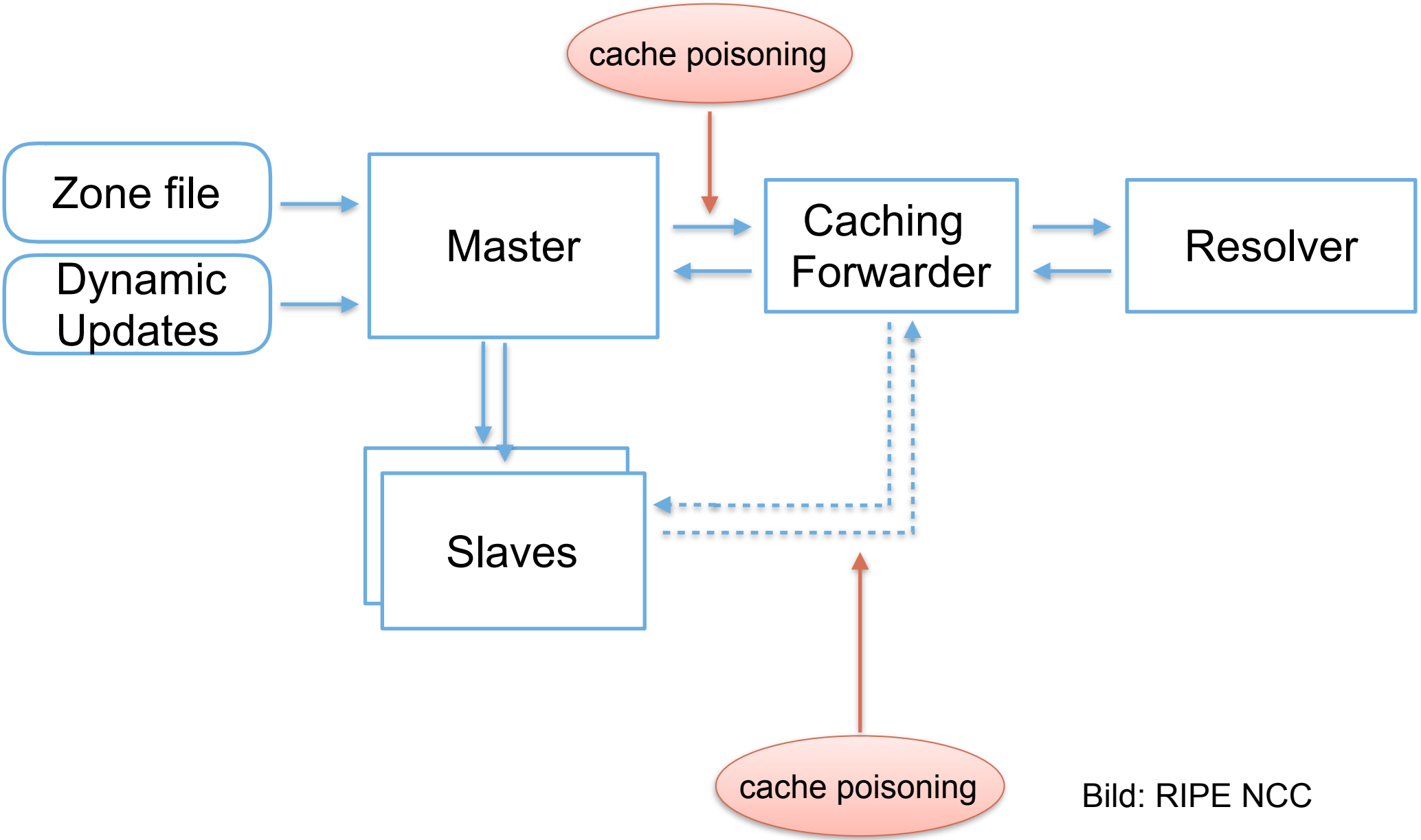


Bild: RIPE NCC

DNS Nameserver - Angriffspunkte

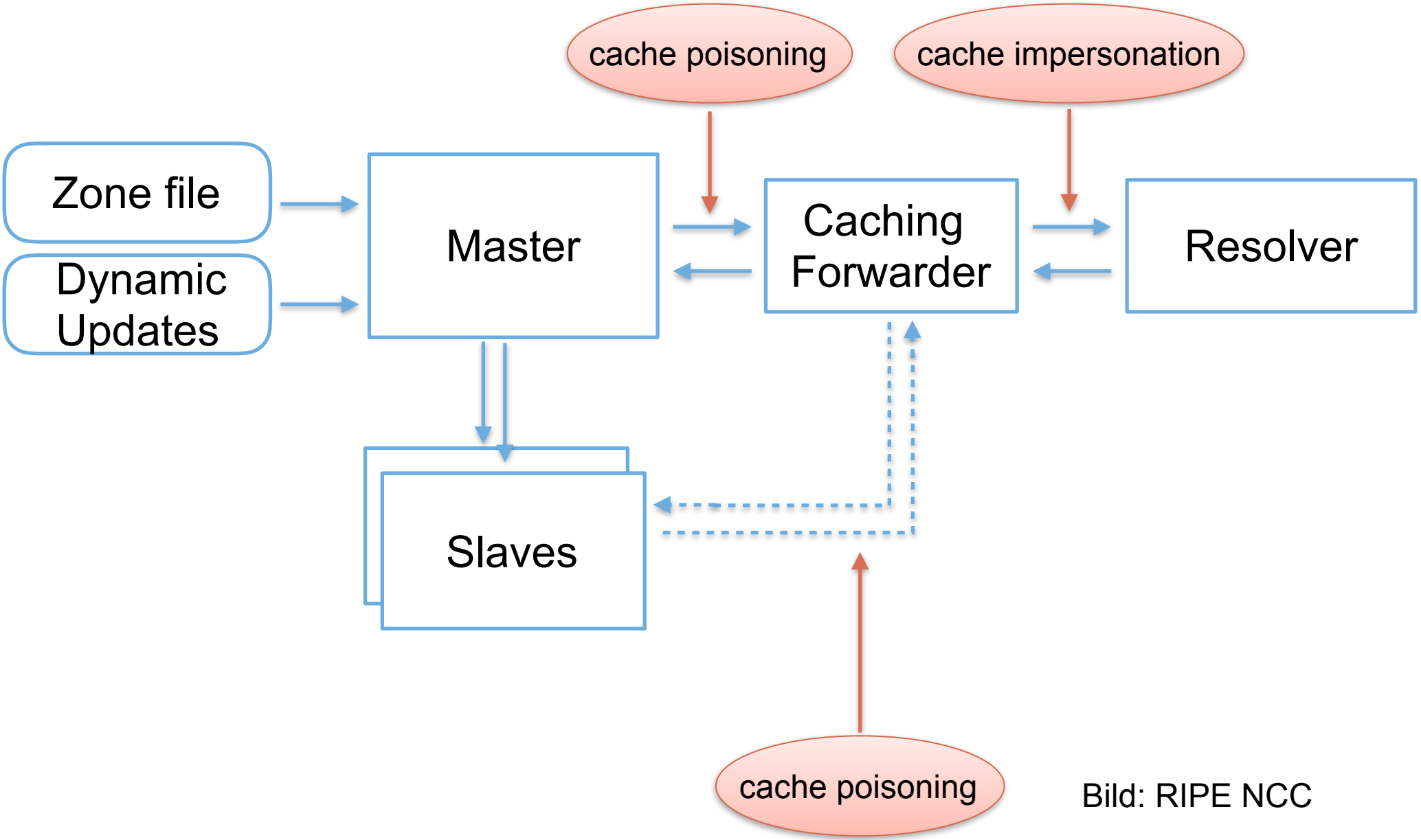


Bild: RIPE NCC

DNS Nameserver - Angriffspunkte

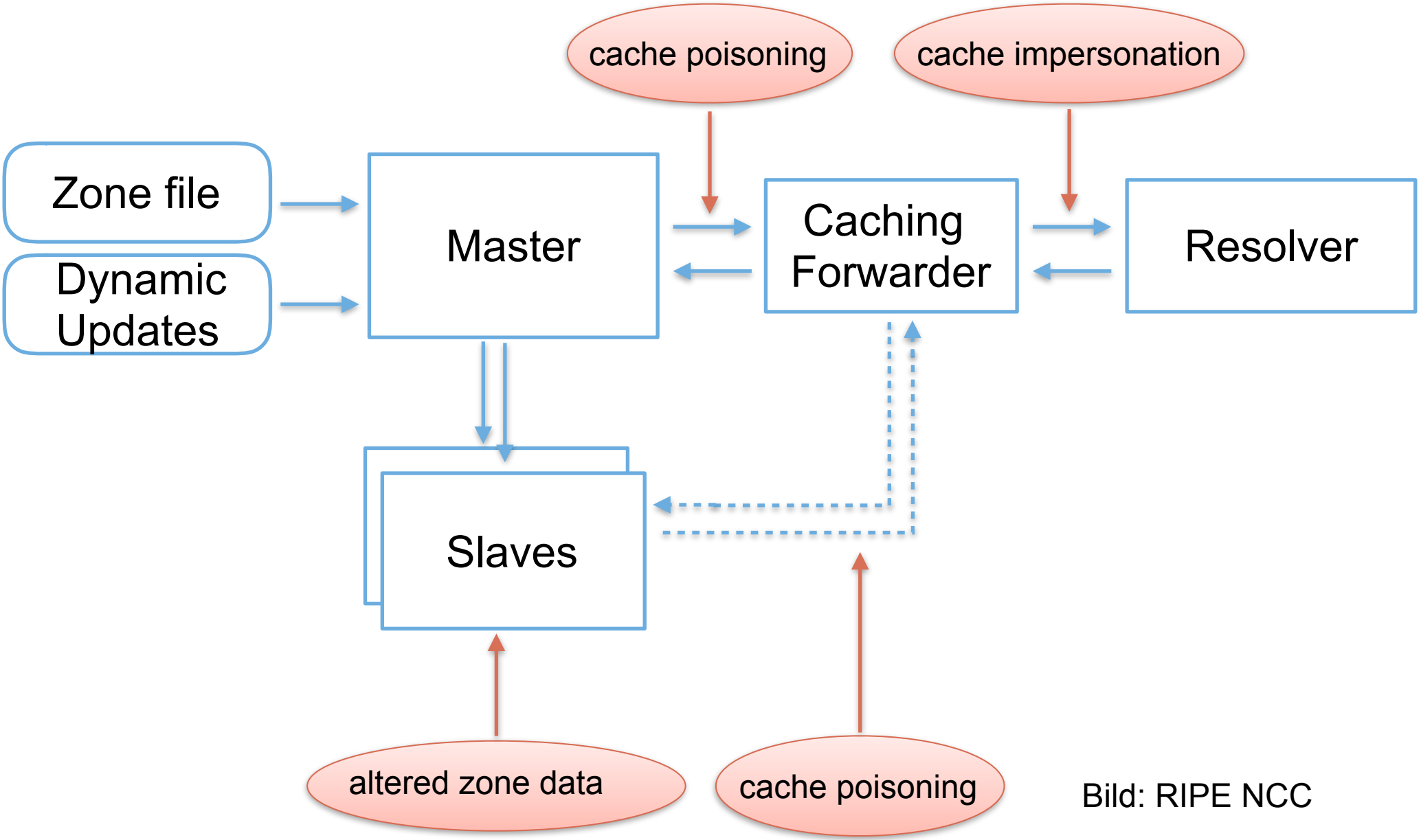
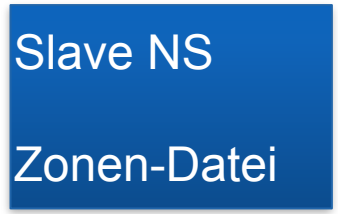
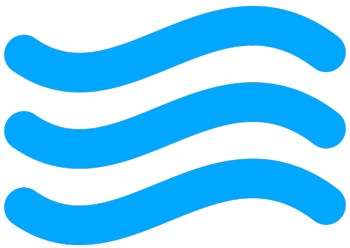


Bild: RIPE NCC

Angriffsvektoren auf DNS Abfragen/Server

- Cache impersonation
  = (Spoofed Address attack / Man-in-the-middle attack)
- Cache poisoning
- Geänderte Zonendateien auf Slaves
- TCP SYN Flood Attacks / UDP Flood Attack
- Zone Walking

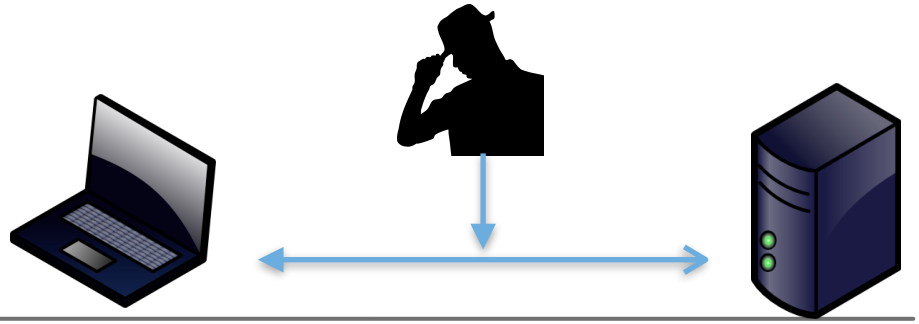




Angriffe sind kein Selbstzweck

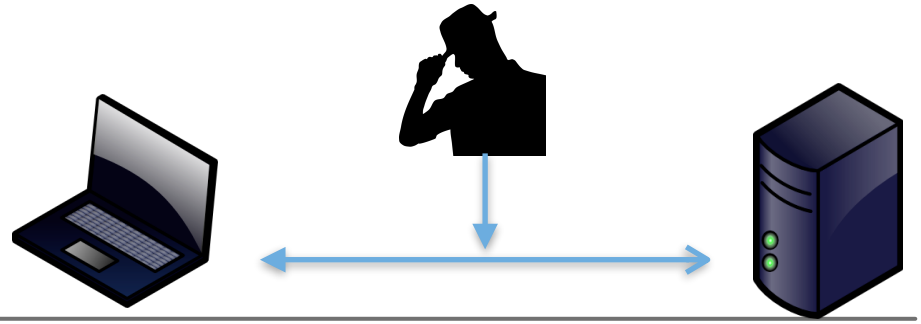
- sondern Vorbereitung für
  - Man-in-the-Middle Angriffe
  - Advertisement Fraud
  - Drive-by Angriffe für Malware
  - Hijacking von SMTP, dann Abgreifen von Passwortresetmails
- schwer zu erkennen, noch schwerer zu verhindern
- betrifft unter Umständen tausende von Nutzern auf einen Schlag

Man-in-middle attack



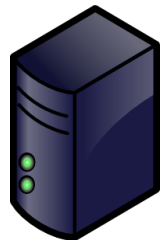
- Angreifer sendet ein IP Paket mit der Adresse des DNS-Servers als Absender
- Muss die korrekte DNS-ID-Nummer verwenden
- ID der Anfrage sniffen, muß aber schneller als der echte DNS NS antworten

Man-in-middle attack



- Angreifer sendet ein IP Paket mit der Adresse des DNS-Servers als Absender
- Muss die korrekte DNS-ID-Nummer verwenden
- ID der Anfrage sniffen, muß aber schneller als der echte DNS NS antworten

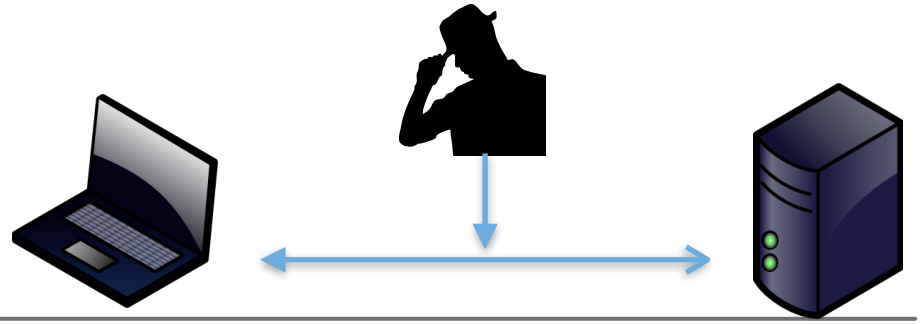
Client



echter
DNS NS

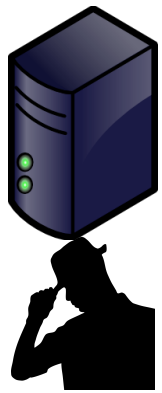
Lokales Netzwerk

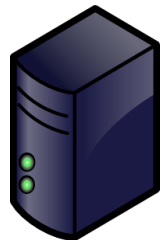
Man-in-middle attack



- Angreifer sendet ein IP Paket mit der Adresse des DNS-Servers als Absender
- Muss die korrekte DNS-ID-Nummer verwenden
- ID der Anfrage sniffen, muß aber schneller als der echte DNS NS antworten

Client

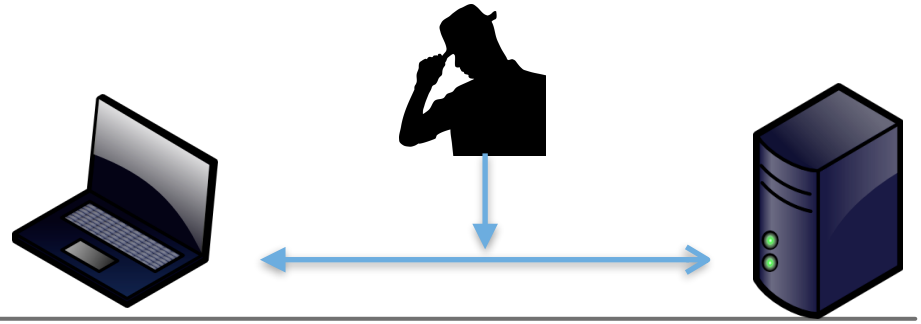




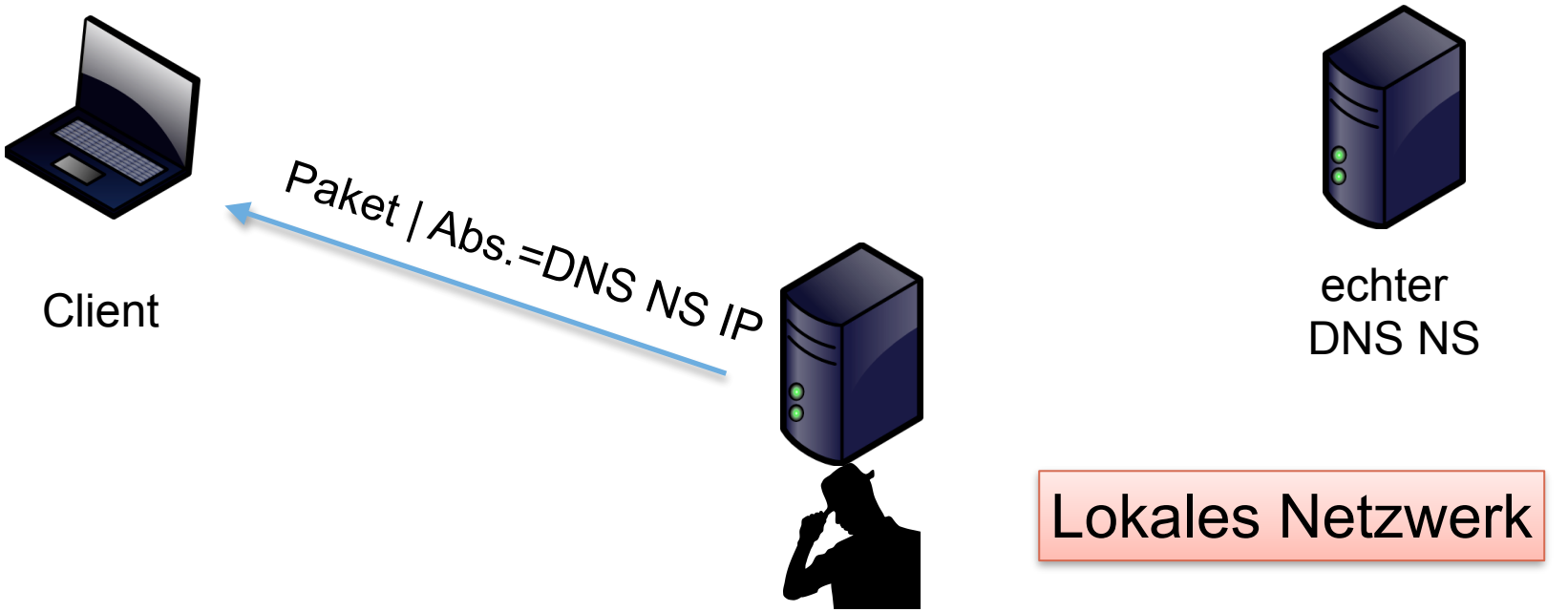
echter
DNS NS

Lokales Netzwerk

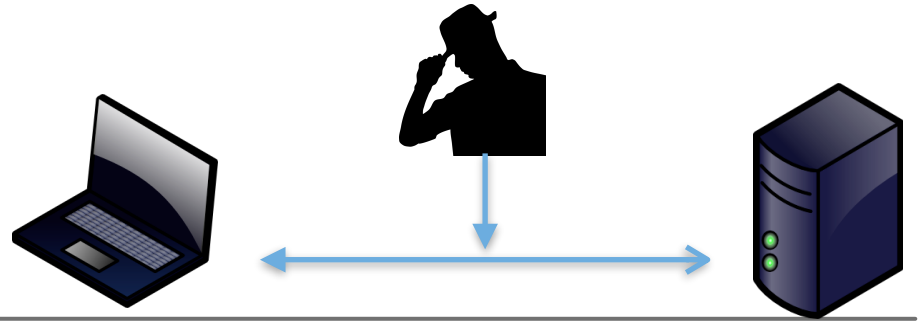
Man-in-middle attack



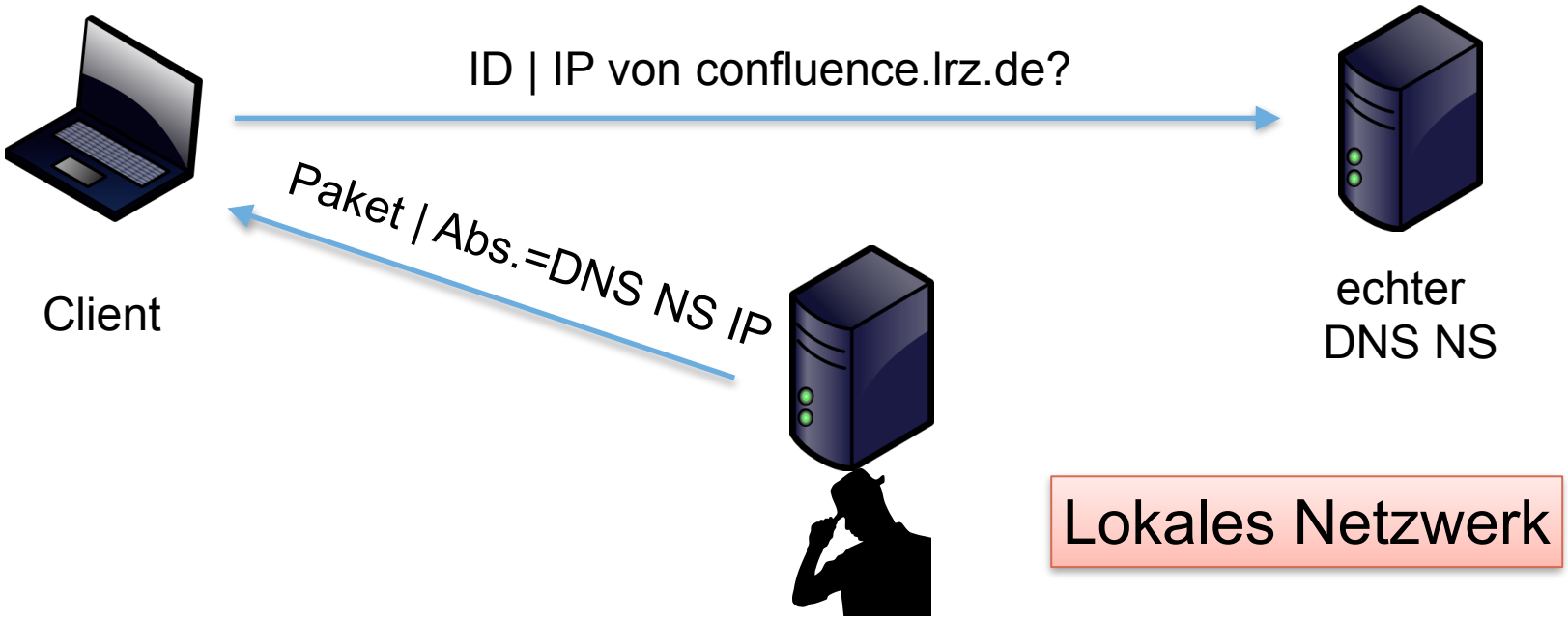
- Angreifer sendet ein IP Paket mit der Adresse des DNS-Servers als Absender
- Muss die korrekte DNS-ID-Nummer verwenden
- ID der Anfrage sniffen, muß aber schneller als der echte DNS NS antworten



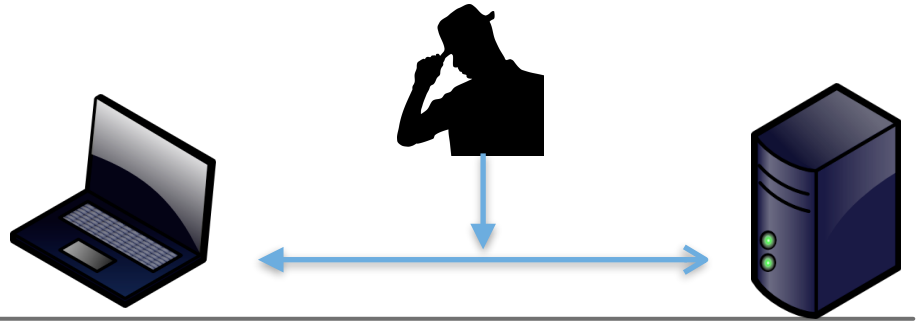
Man-in-middle attack



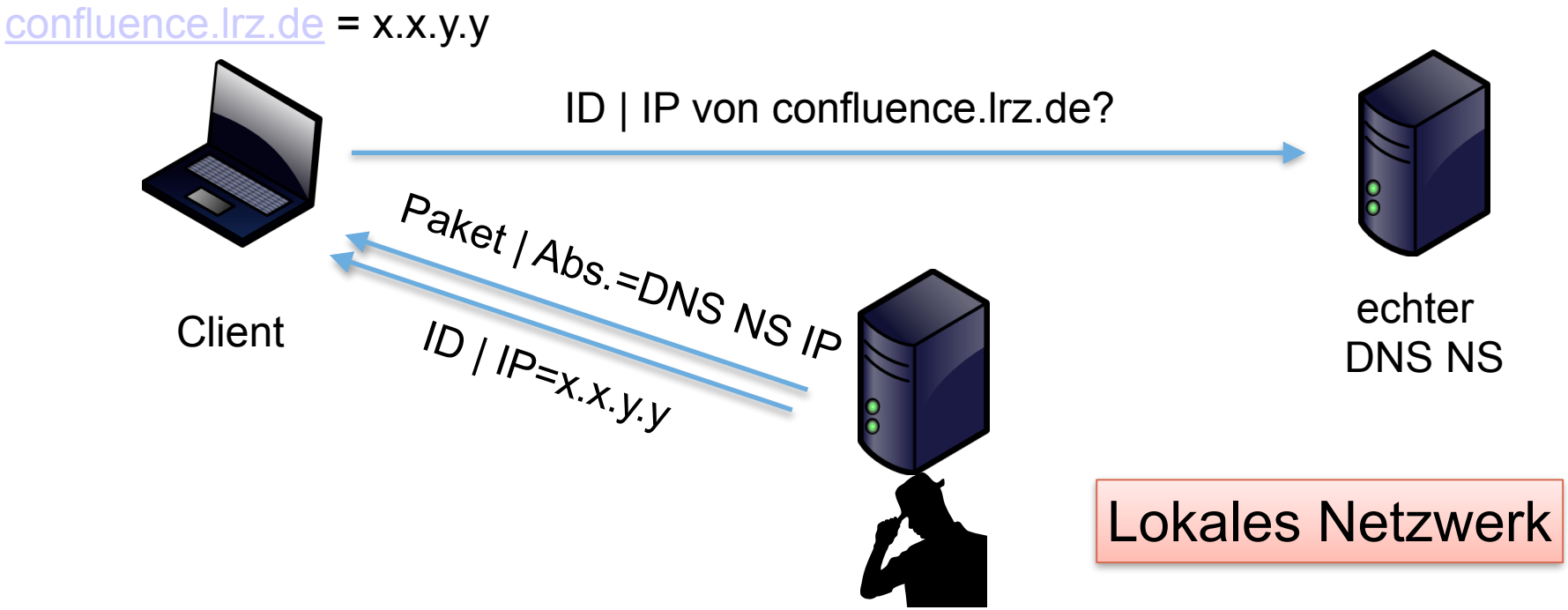
- Angreifer sendet ein IP Paket mit der Adresse des DNS-Servers als Absender
- Muss die korrekte DNS-ID-Nummer verwenden
- ID der Anfrage sniffen, muß aber schneller als der echte DNS NS antworten



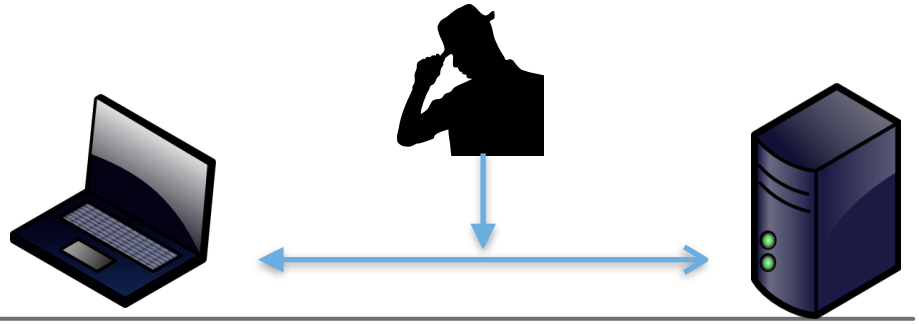
Man-in-middle attack



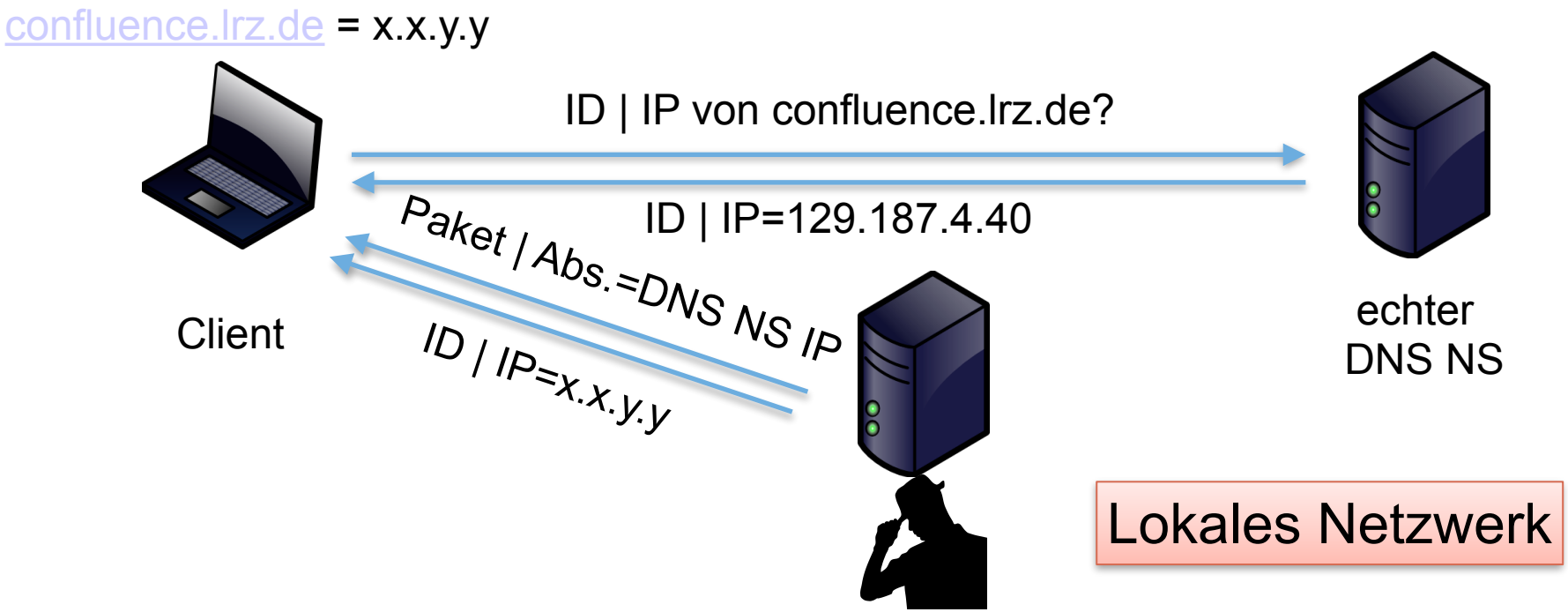
- Angreifer sendet ein IP Paket mit der Adresse des DNS-Servers als Absender
- Muss die korrekte DNS-ID-Nummer verwenden
- ID der Anfrage sniffen, muß aber schneller als der echte DNS NS antworten



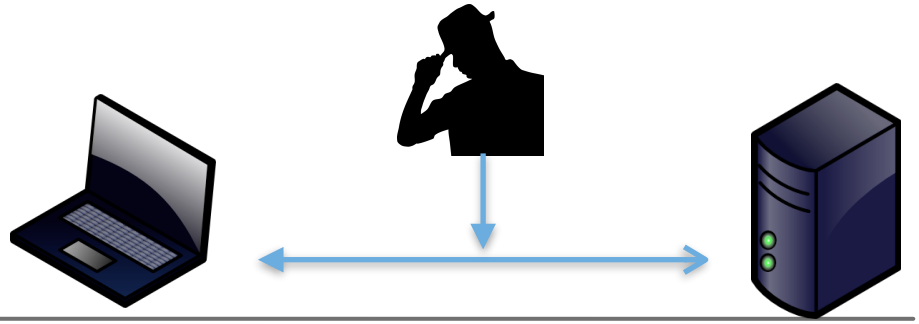
Man-in-middle attack



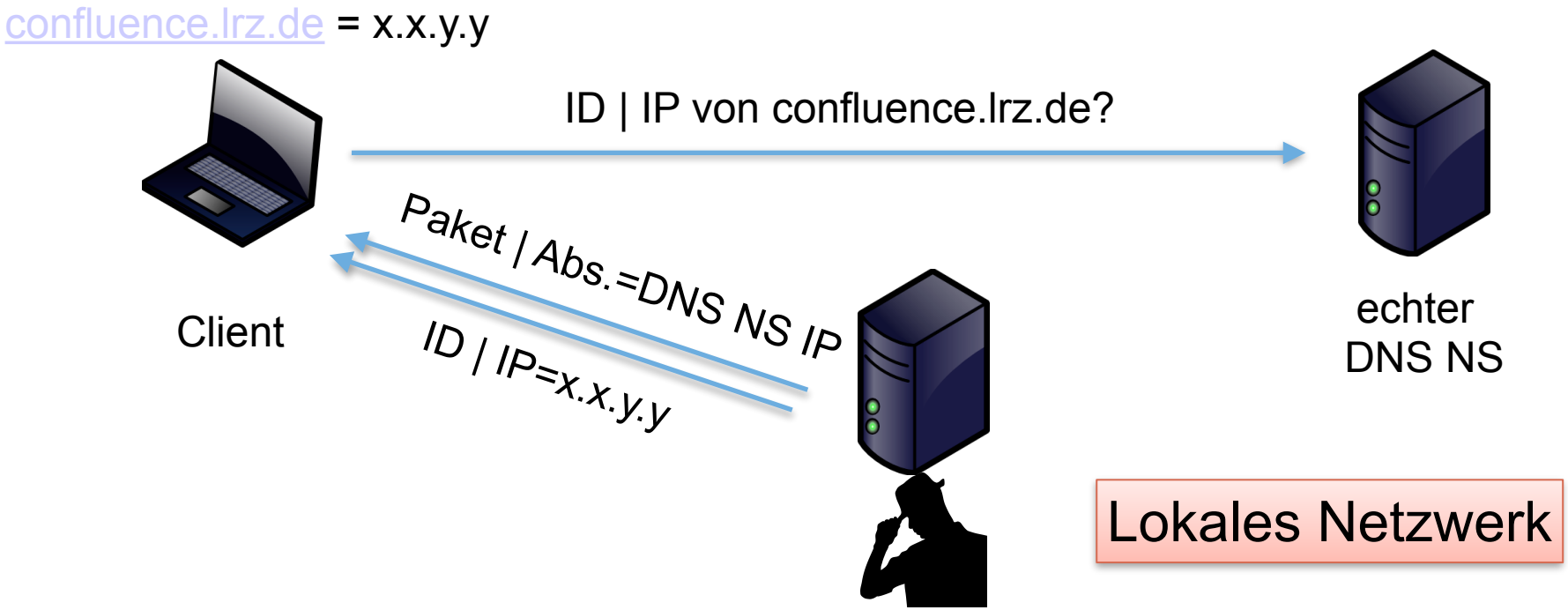
- Angreifer sendet ein IP Paket mit der Adresse des DNS-Servers als Absender
- Muss die korrekte DNS-ID-Nummer verwenden
- ID der Anfrage sniffen, muß aber schneller als der echte DNS NS antworten



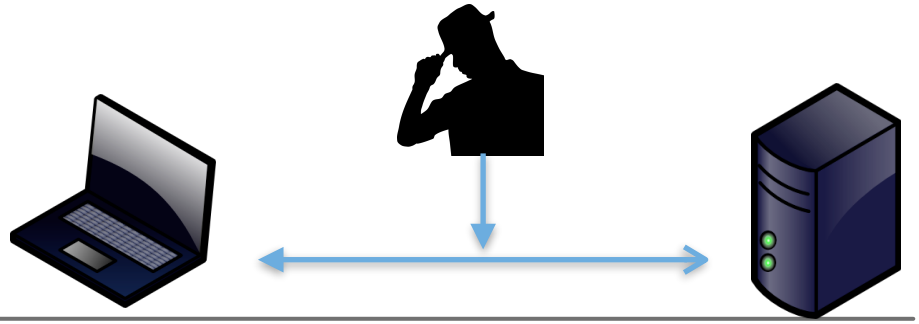
Man-in-middle attack



- Angreifer sendet ein IP Paket mit der Adresse des DNS-Servers als Absender
- Muss die korrekte DNS-ID-Nummer verwenden
- ID der Anfrage sniffen, muß aber schneller als der echte DNS NS antworten

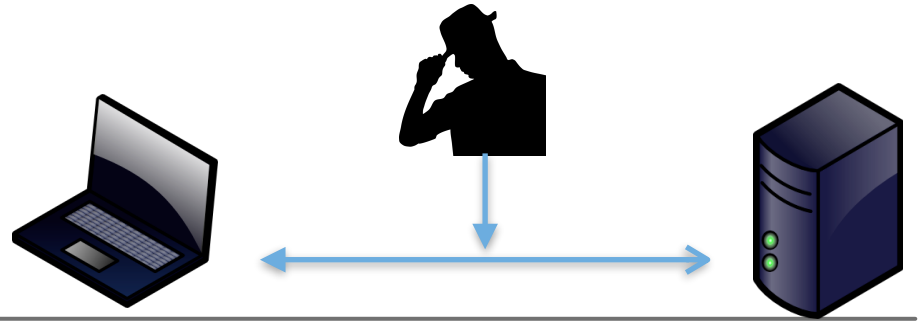


Man-in-middle attack



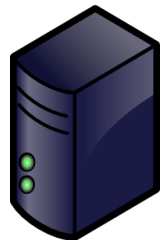
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen

Man-in-middle attack



- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen

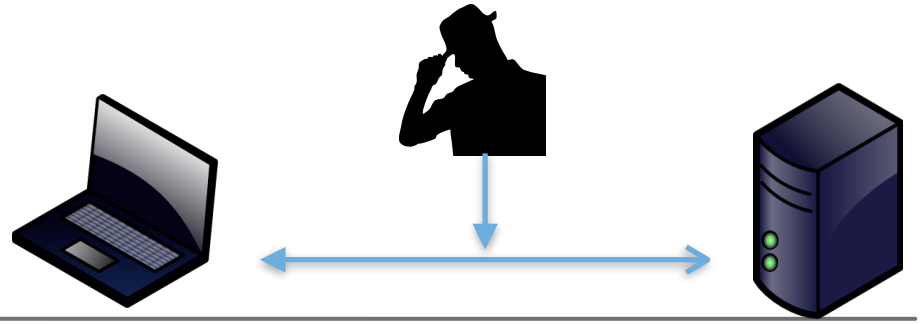
Client



echter
DNS NS

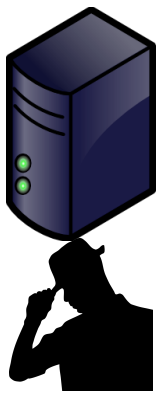
Internet

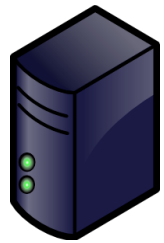
Man-in-middle attack



- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen

Client

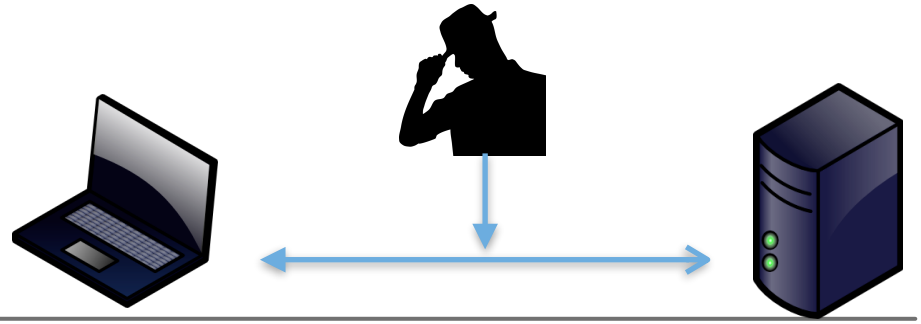




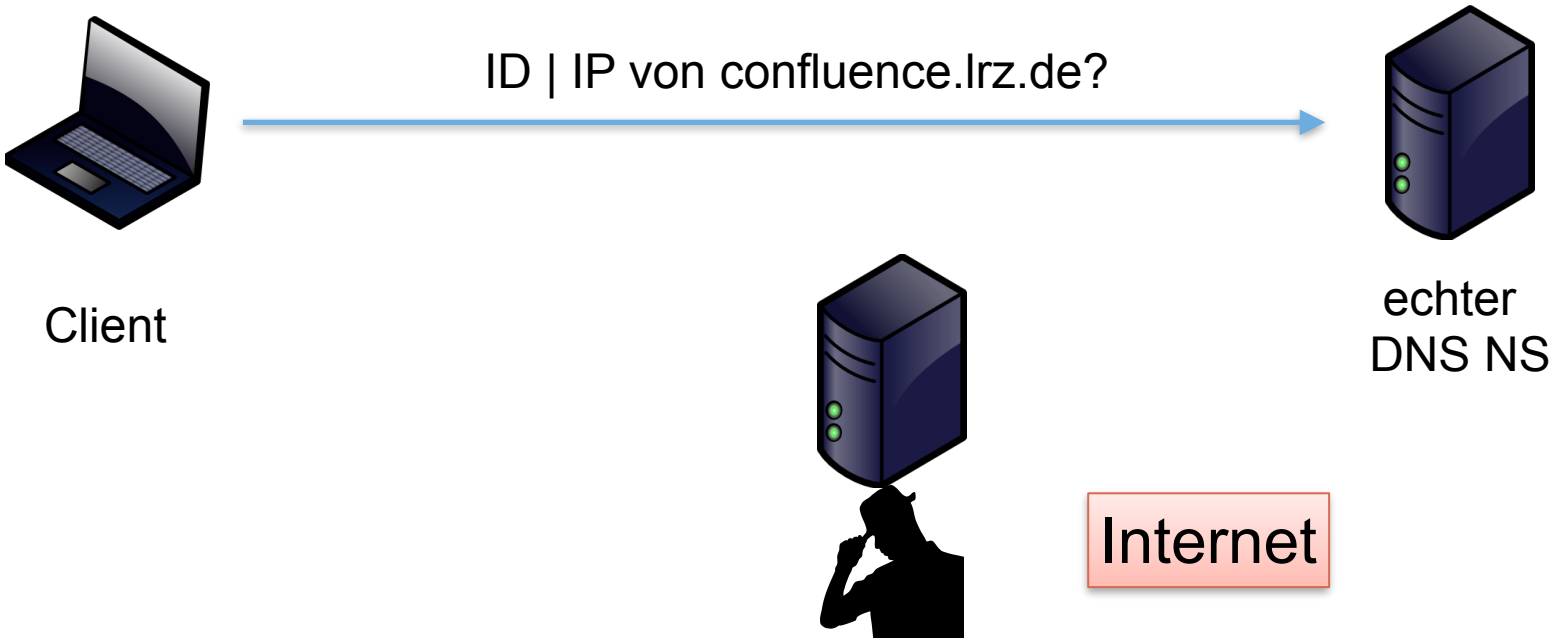
echter
DNS NS

Internet

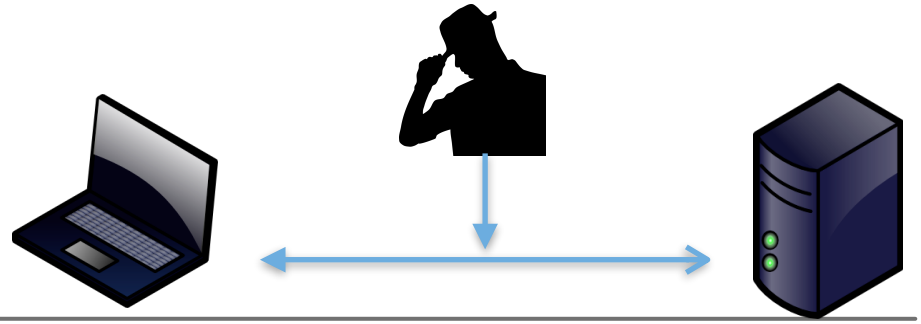
Man-in-middle attack



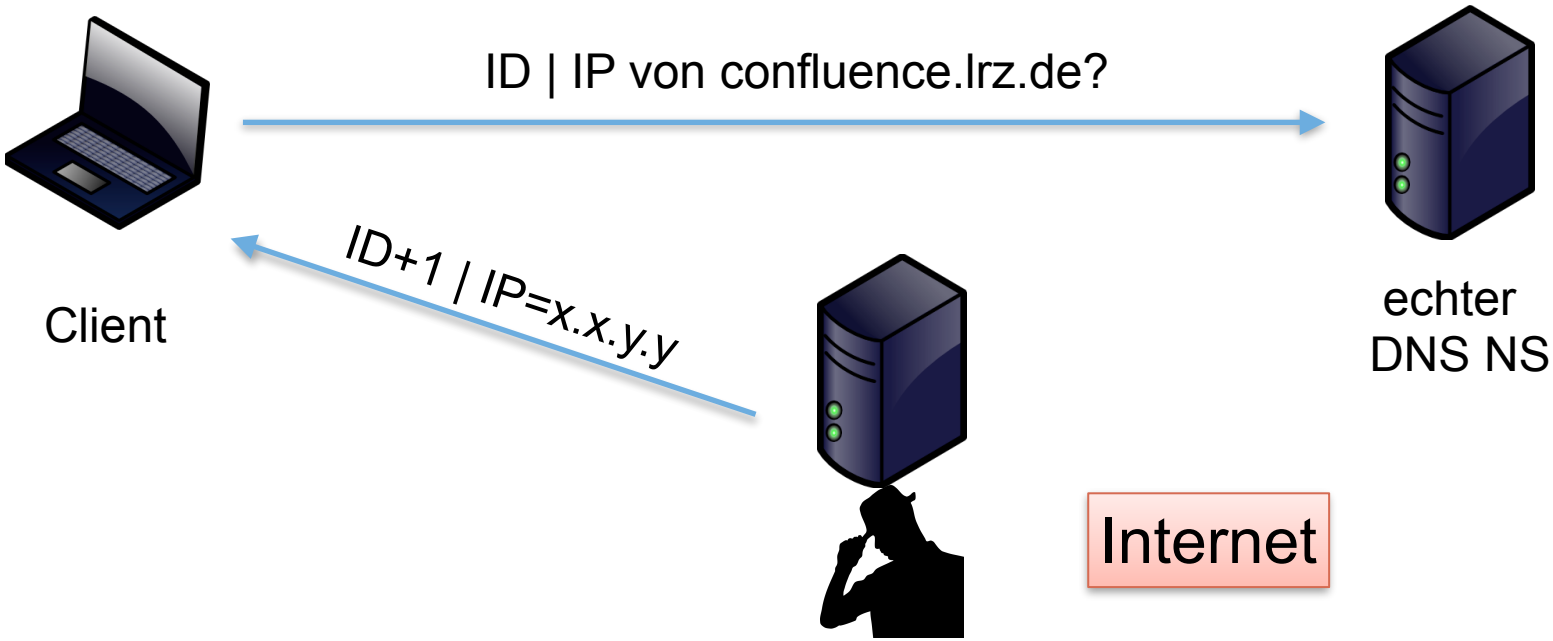
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



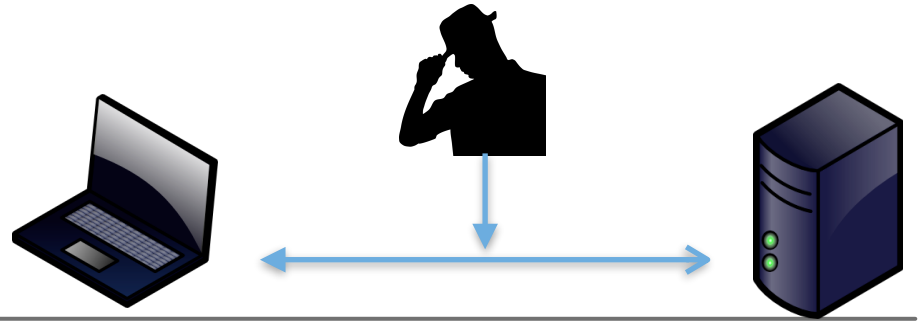
Man-in-middle attack



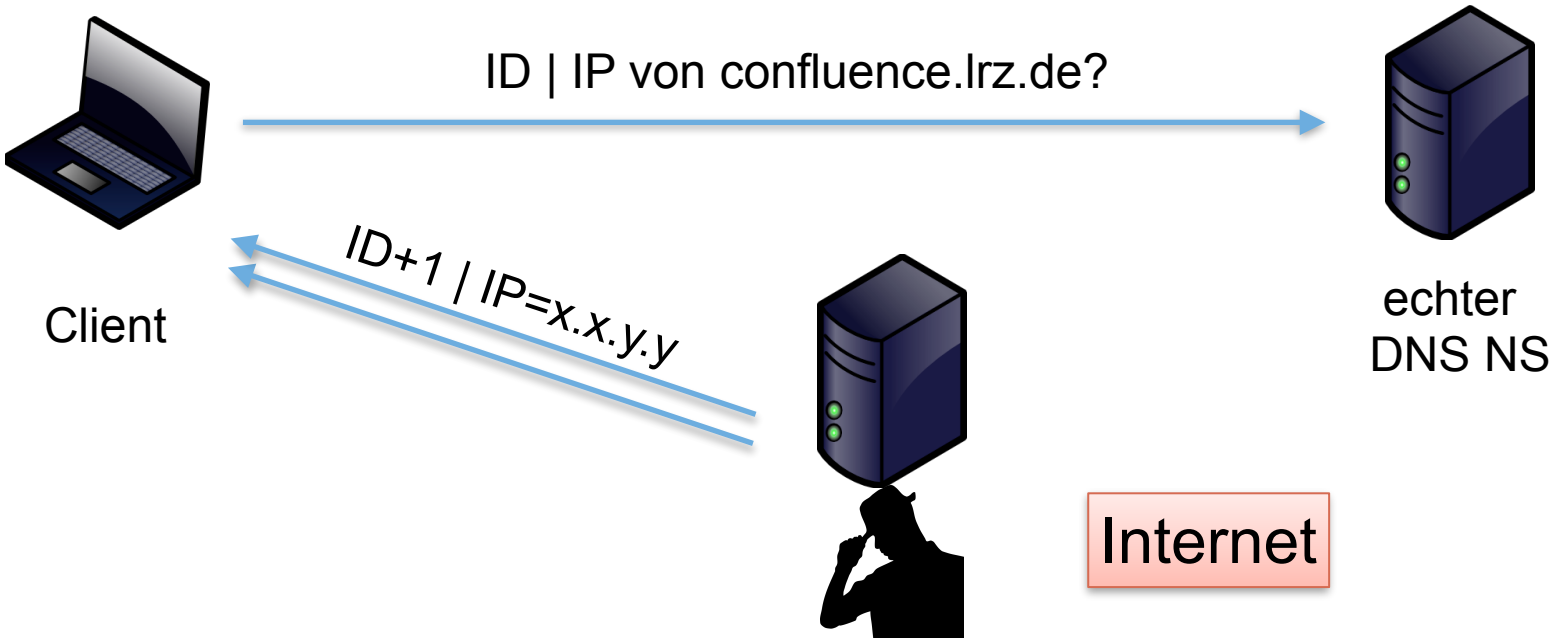
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



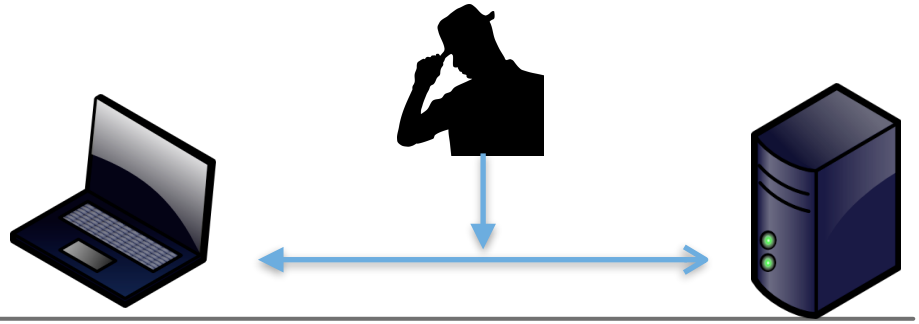
Man-in-middle attack



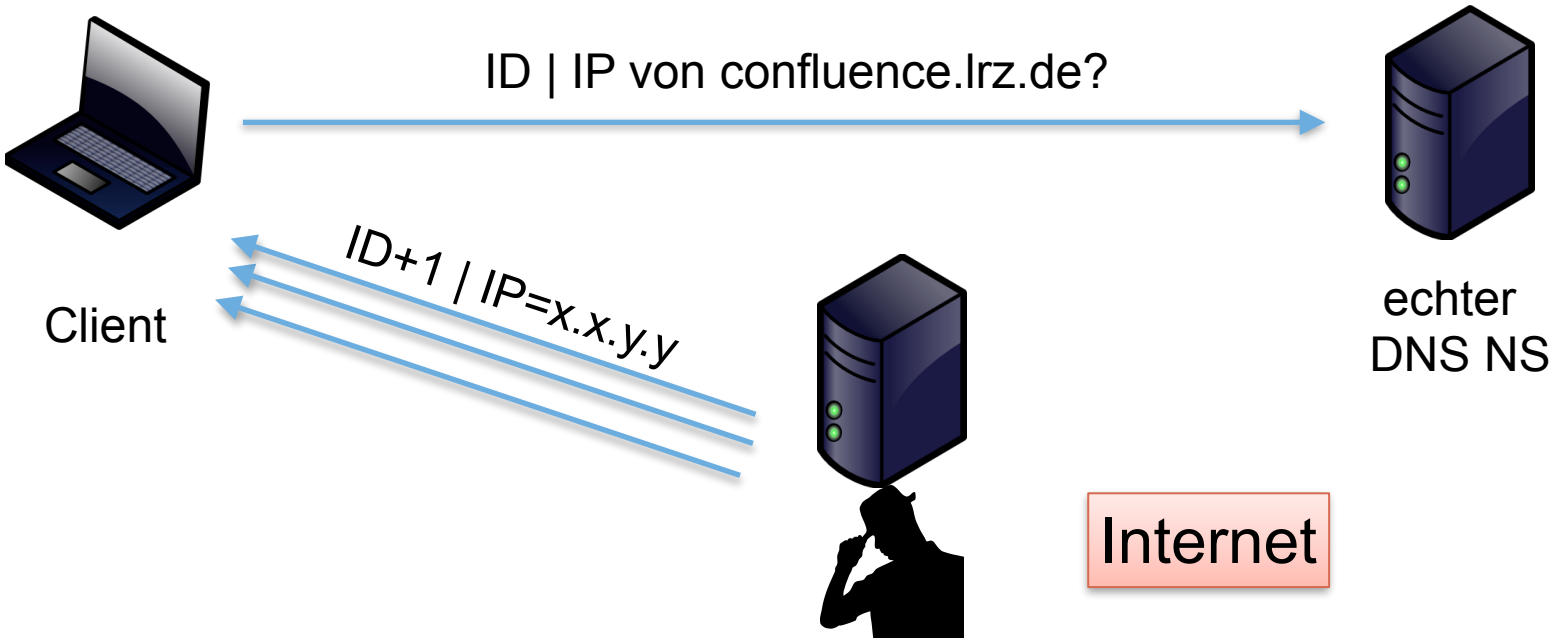
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



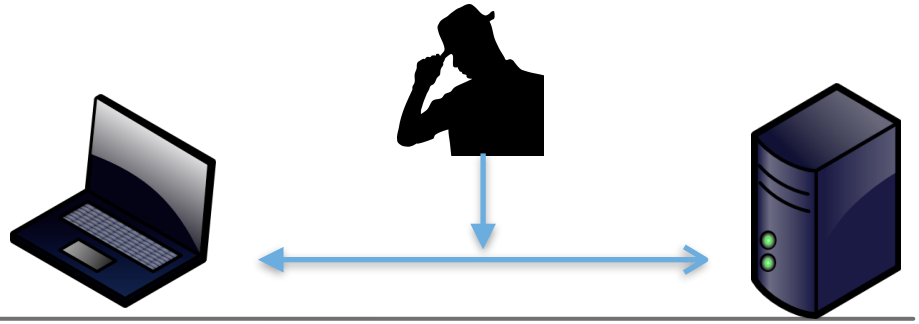
Man-in-middle attack



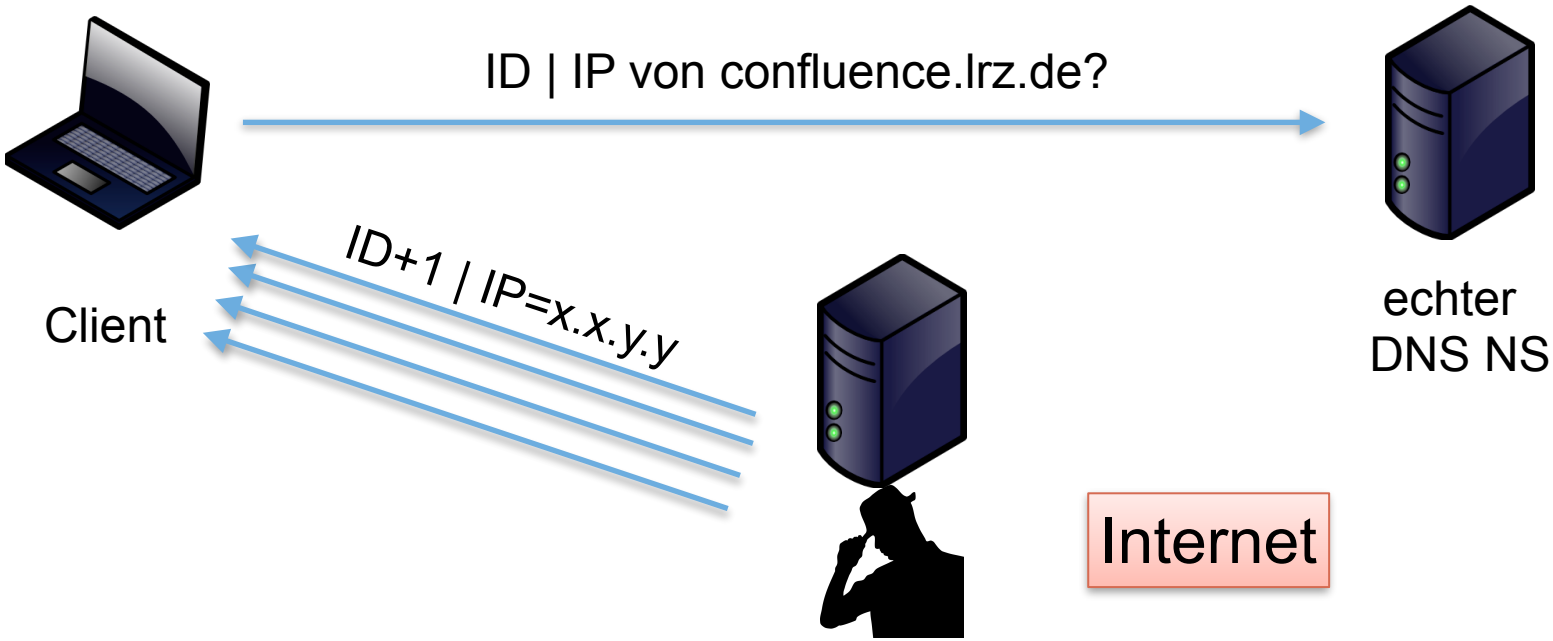
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



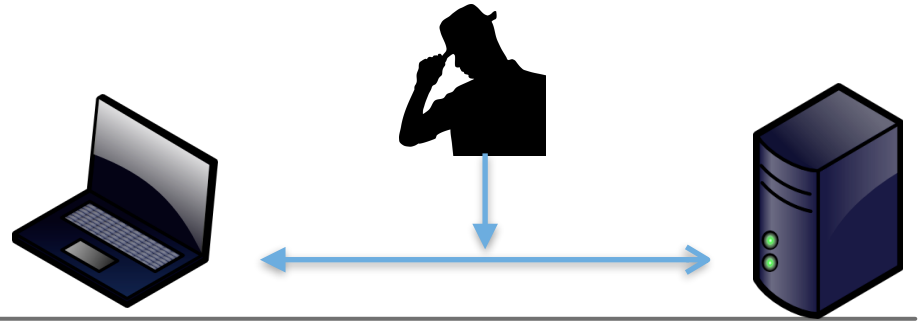
Man-in-middle attack



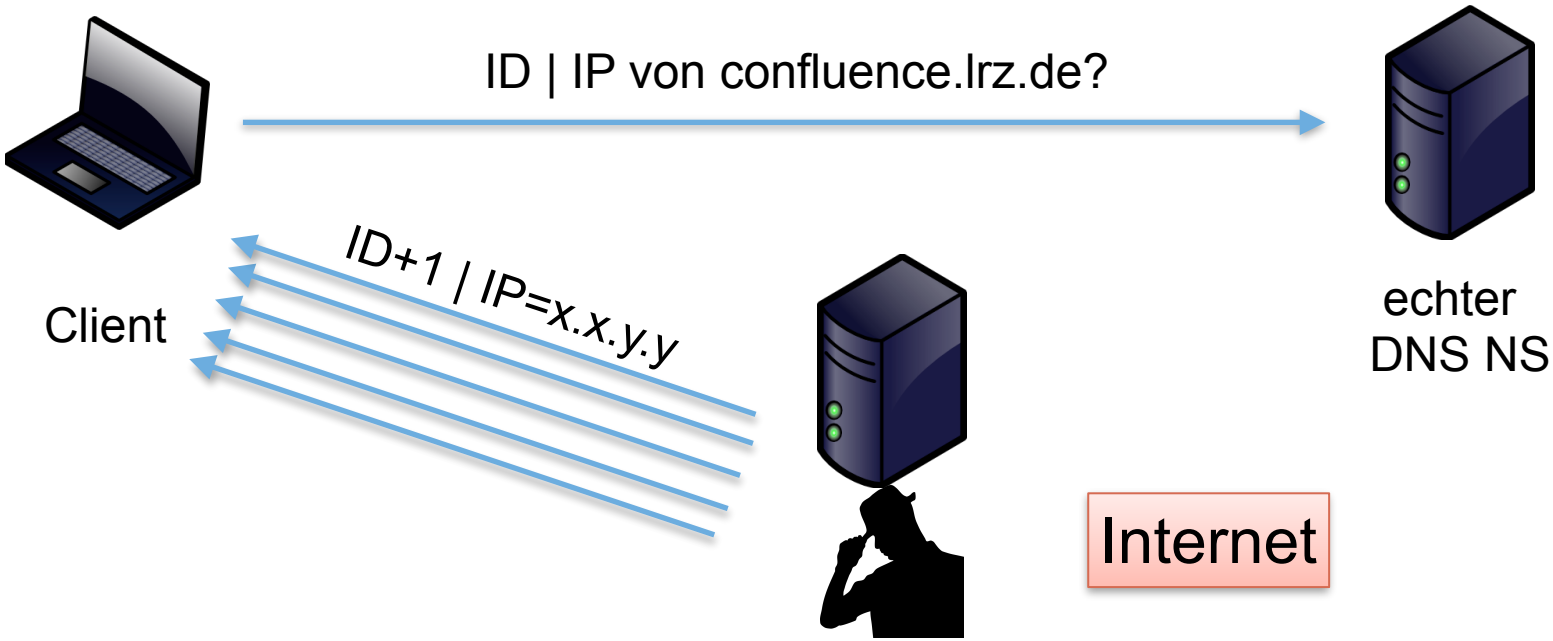
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



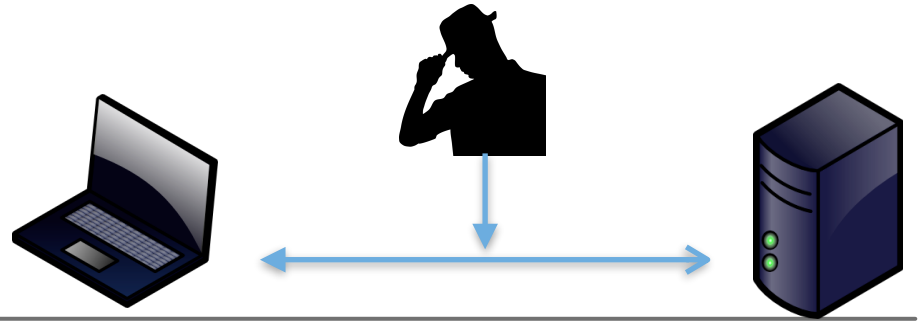
Man-in-middle attack



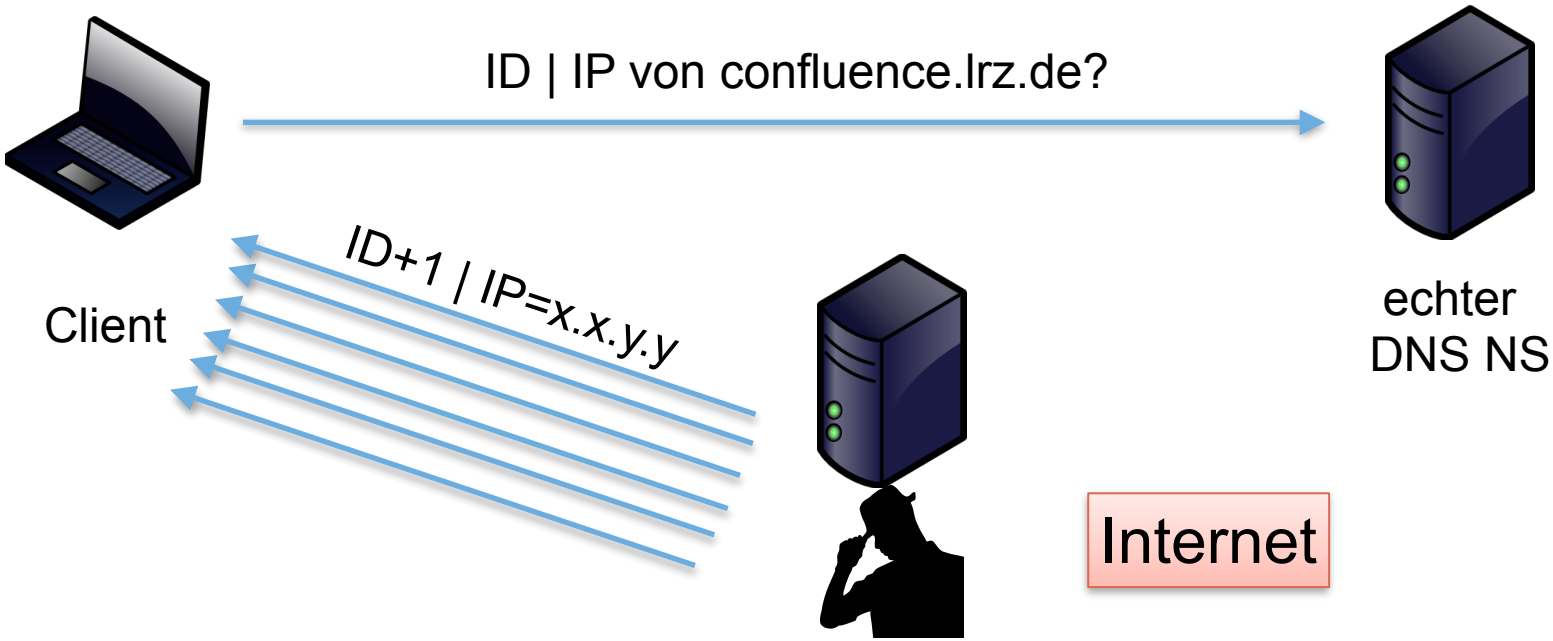
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



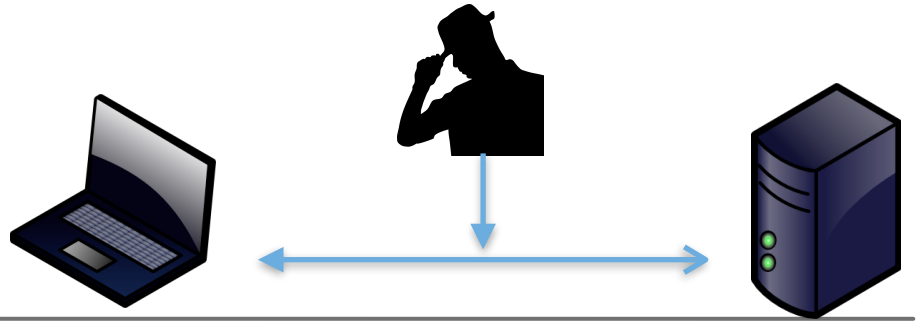
Man-in-middle attack



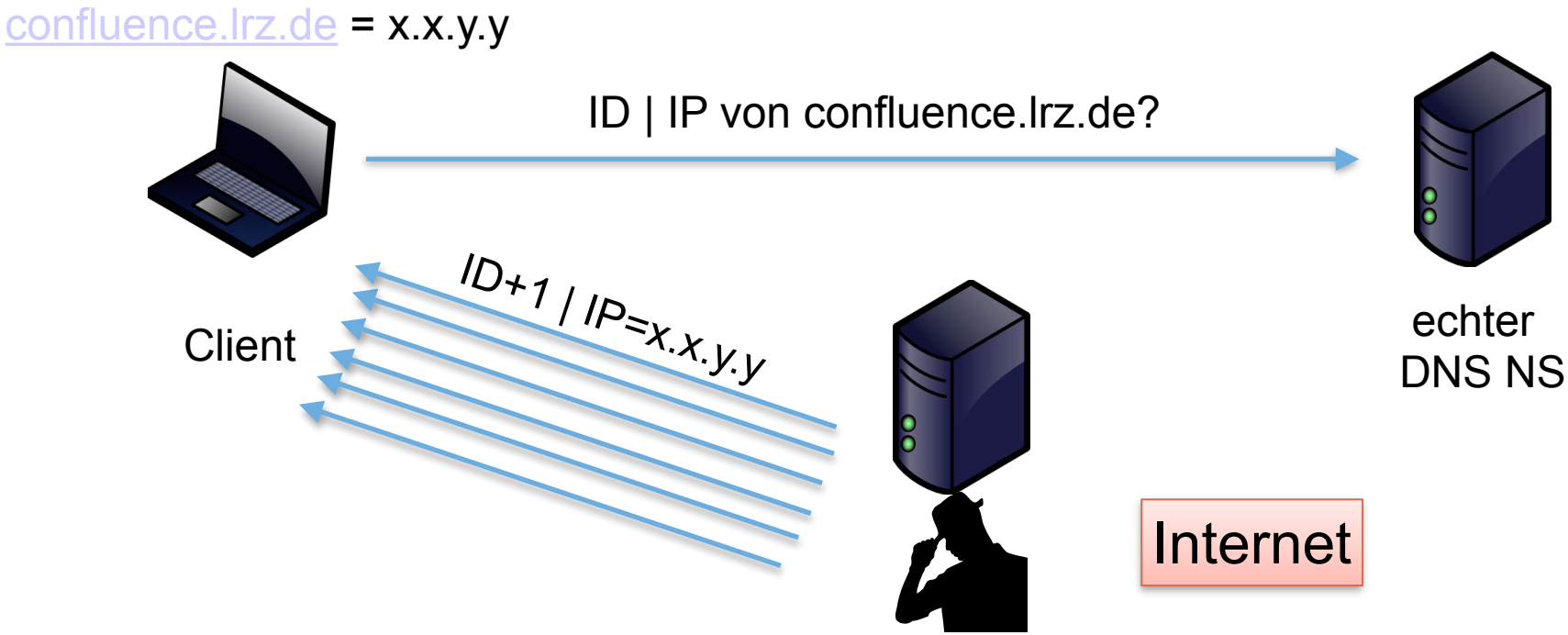
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



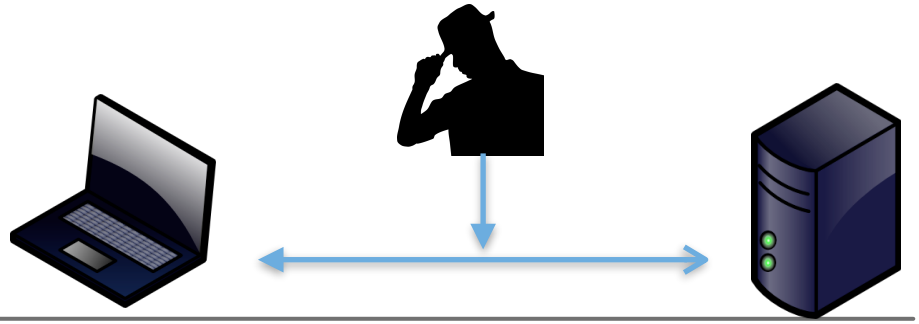
Man-in-middle attack



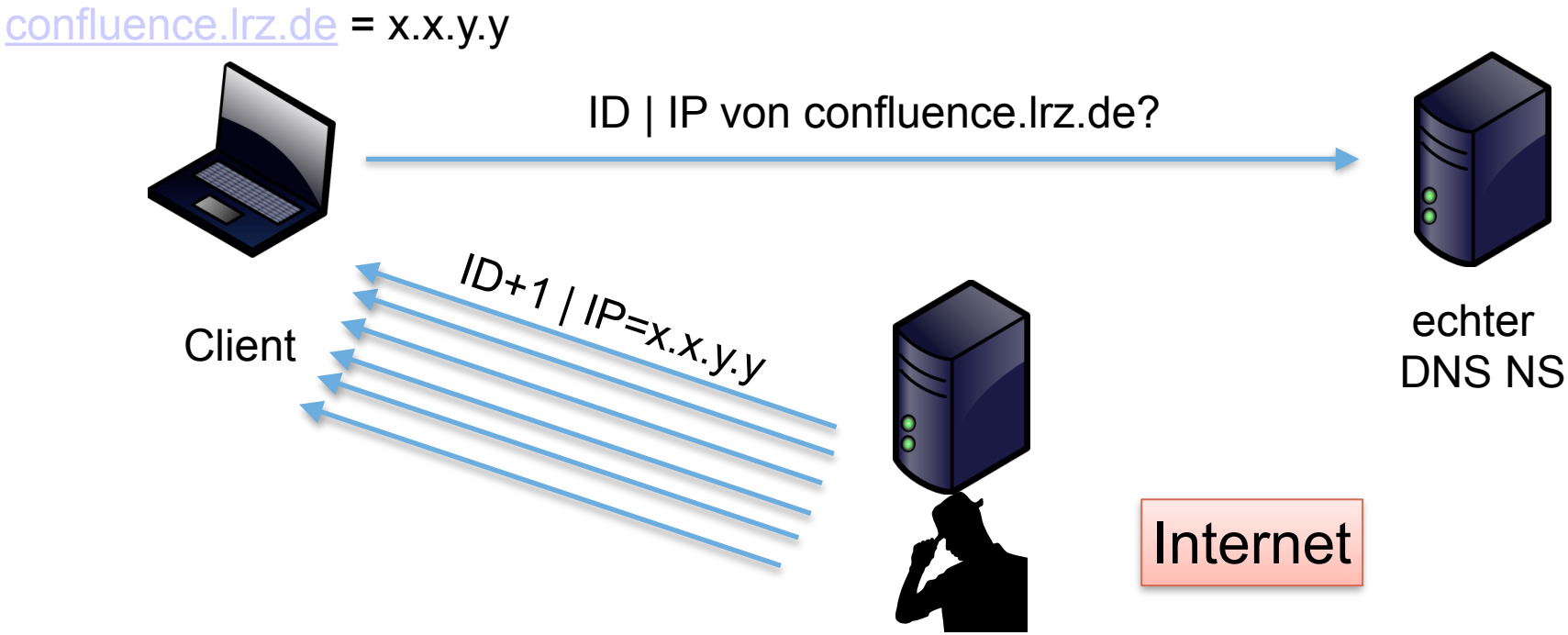
- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



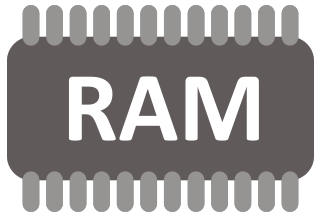
Man-in-middle attack



- Pakete mit zufälligen (allen) möglichen IDs in Paketen
- ~N x 100 Antworten senden, um die ID mit hoher Wahrscheinlichkeit zu treffen
- Einige Nameserver erhöhen einfach die ID um 1 von aufeinander folgenden Antworten auf Anfragen



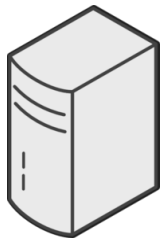
DNS Caching



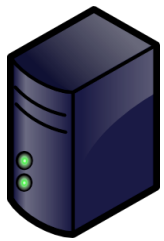
- DNS server caches IP-Domänenname-Zuordnung zur Optimierung zukünftiger Anfragen
- Angriffspunkt durch Cache-Poisoning
  - RR-Eintrag
  - DNS ID-Vorhersage
- DNS Cache enthält IP-Verweis auf den Rechner des Angreifers

DNS cache poisoning

confluence.lrz.de. IN A 129.187.4.40



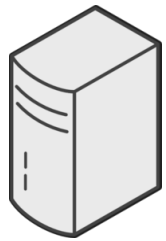
Wenn es der Angreifer durch Spoofing (MTM) schafft, unautorisierte Antworten in den Cache eines Resolvers zu bringen, spricht man von “Cache poisoning”.

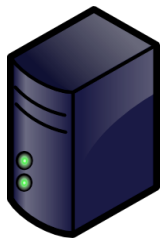


DNS cache poisoning

confluence.lrz.de. IN A 129.187.4.40

Wenn es der Angreifer durch Spoofing (MTM) schafft, unautorisierte Antworten in den Cache eines Resolvers zu bringen, spricht man von “Cache poisoning”.

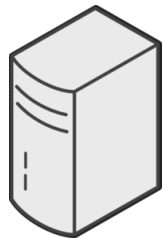




DNS cache poisoning

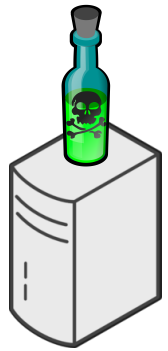
confluence.lrz.de. IN A 129.187.4.40

Wenn es der Angreifer durch Spoofing (MTM) schafft, unautorisierte Antworten in den Cache eines Resolvers zu bringen, spricht man von “Cache poisoning”.

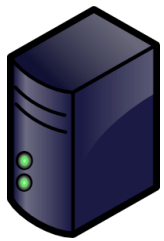


DNS cache poisoning

confluence.lrz.de. IN A x.y.w.z

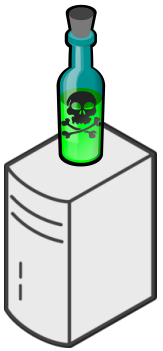


Wenn es der Angreifer durch Spoofing (MTM) schafft, unautorisierte Antworten in den Cache eines Resolvers zu bringen, spricht man von “Cache poisoning”.



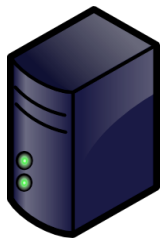
DNS cache poisoning

confluence.lrz.de. IN A x.y.w.z



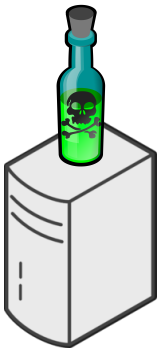
confluence.lrz.de.?

Wenn es der Angreifer durch Spoofing (MTM) schafft, unautorisierte Antworten in den Cache eines Resolvers zu bringen, spricht man von “Cache poisoning”.



DNS cache poisoning

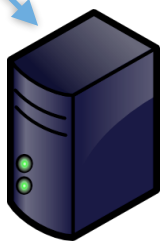
confluence.lrz.de. IN A x.y.w.z



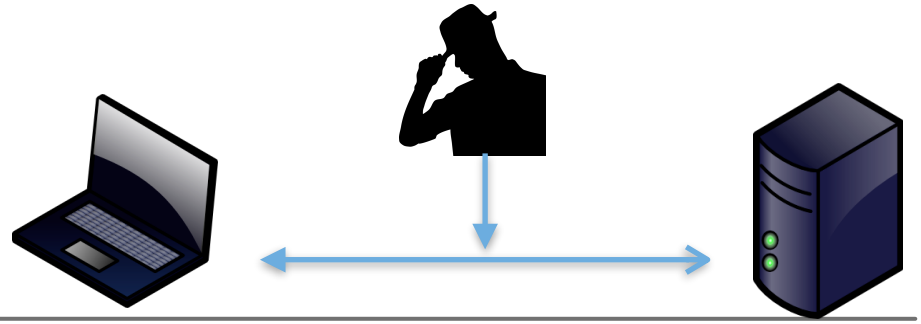
Wenn es der Angreifer durch Spoofing (MTM) schafft, unautorisierte Antworten in den Cache eines Resolvers zu bringen, spricht man von “Cache poisoning”.

IP | x.y.w.z

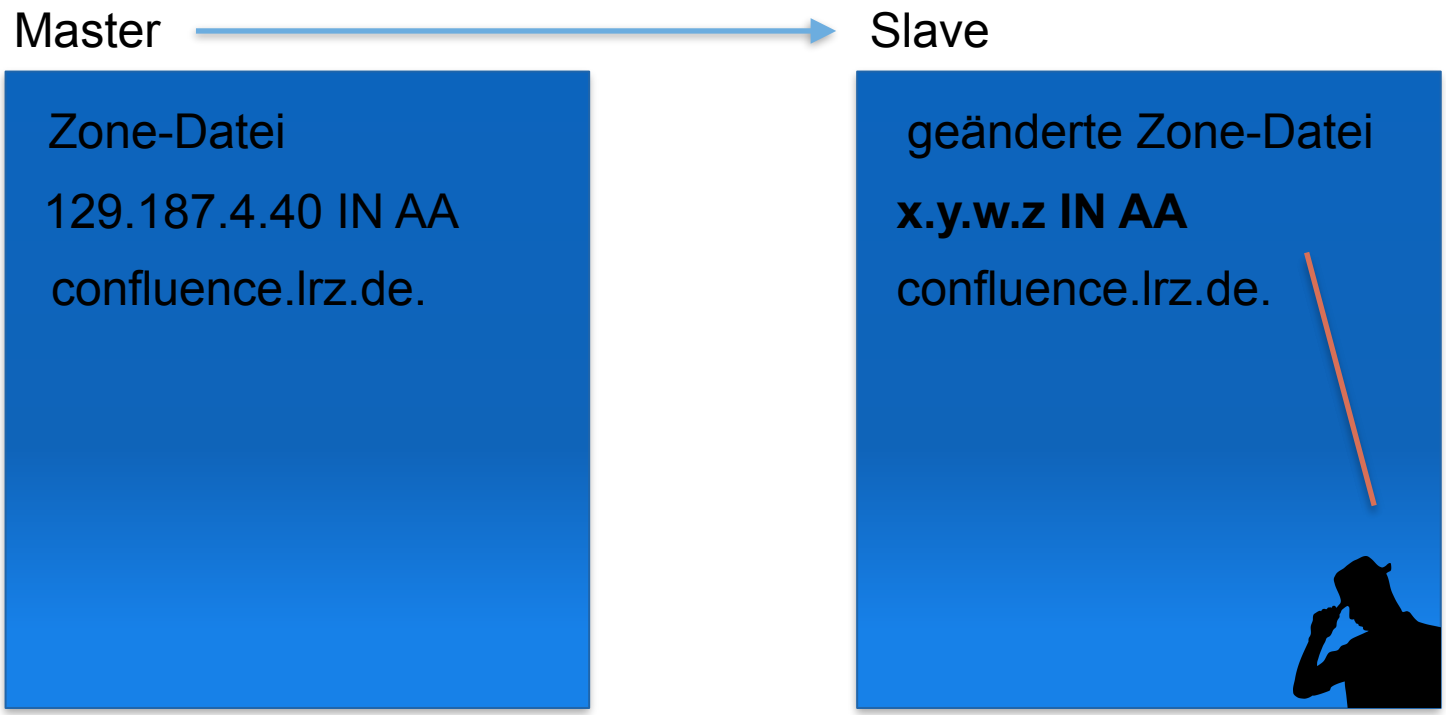
x.y.w.z



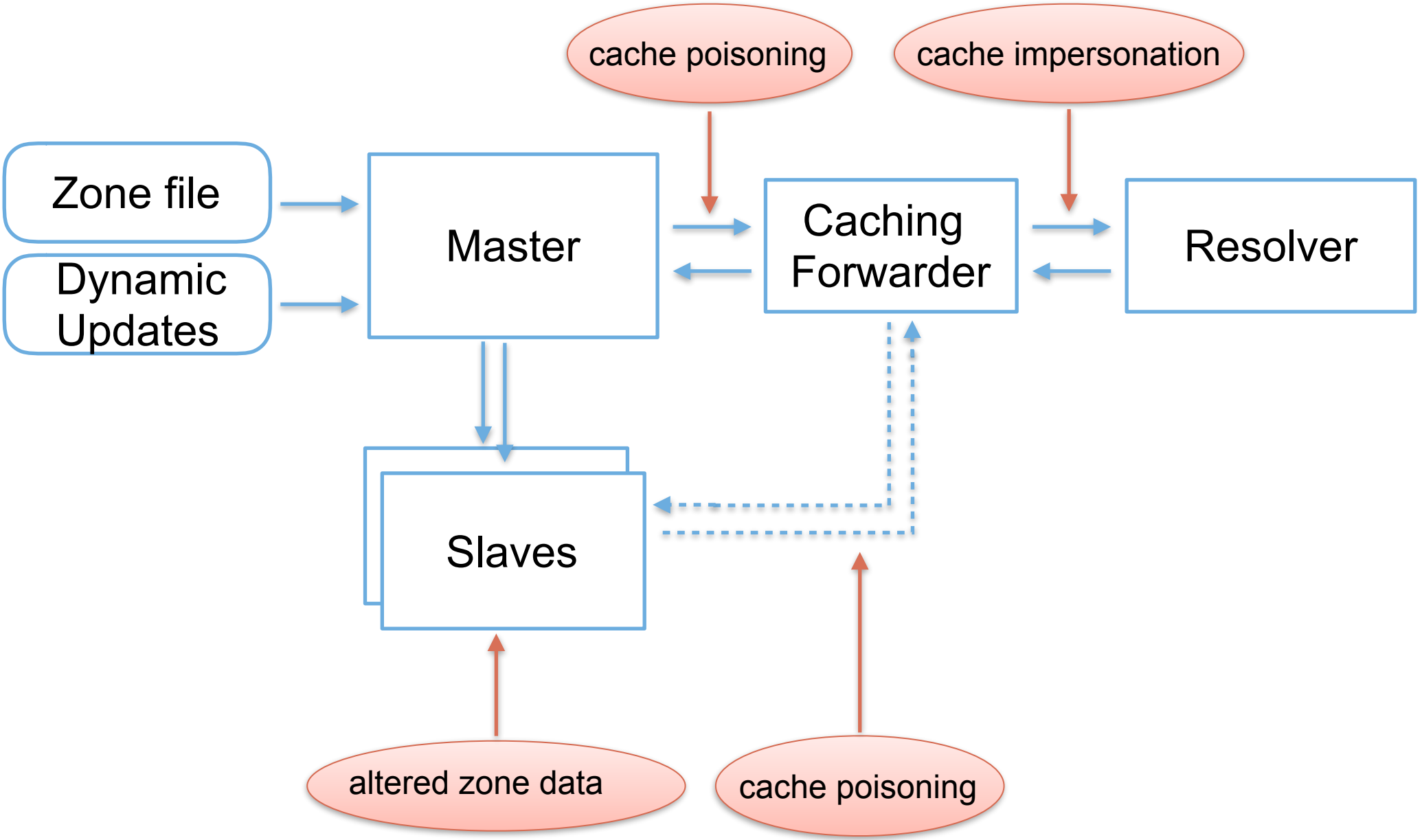
Altered Zone data



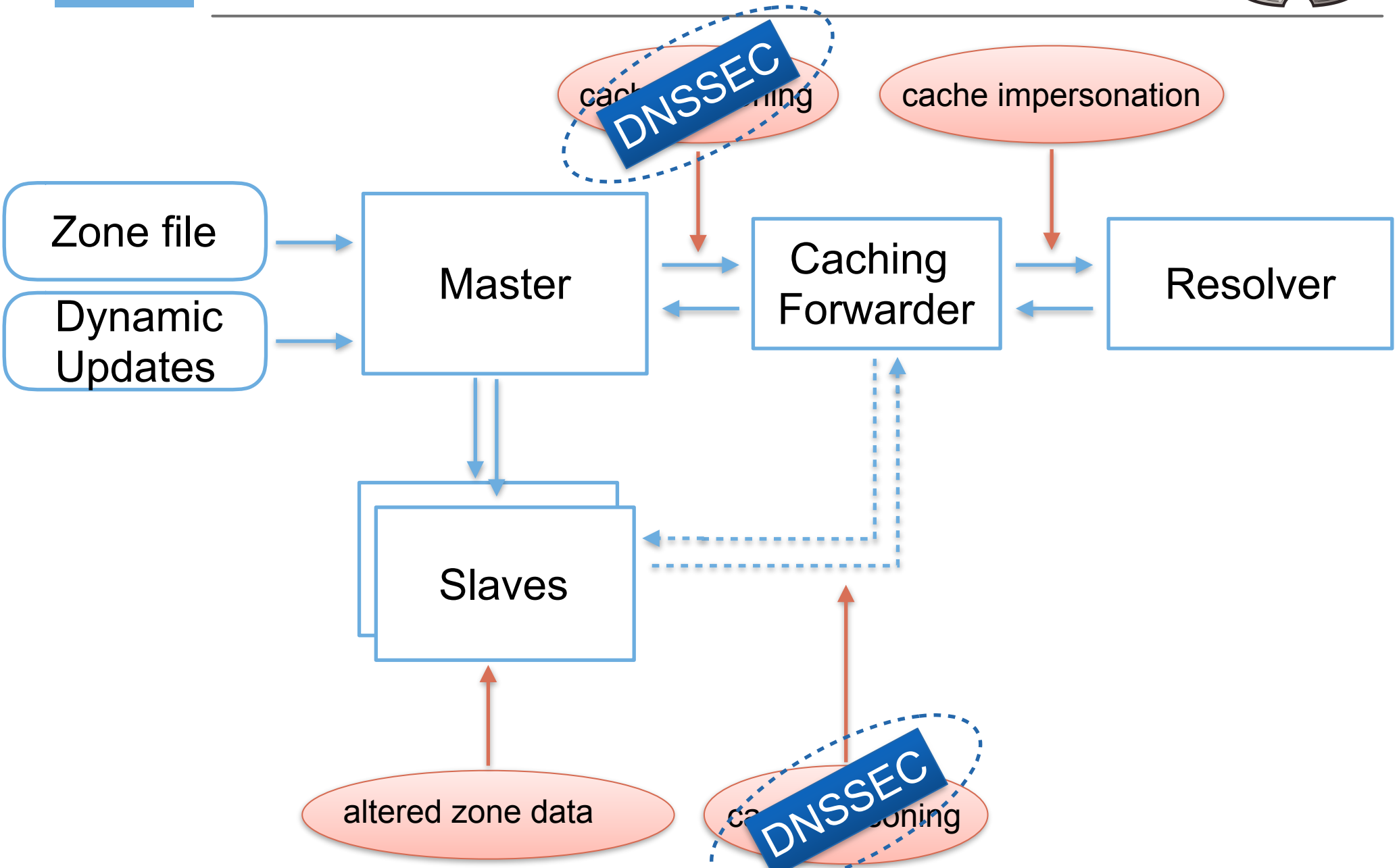
- Zone-Datei auf DNS Slave wird vom Angreifer geändert
- Response leitet auf einen Server des Angreifers um



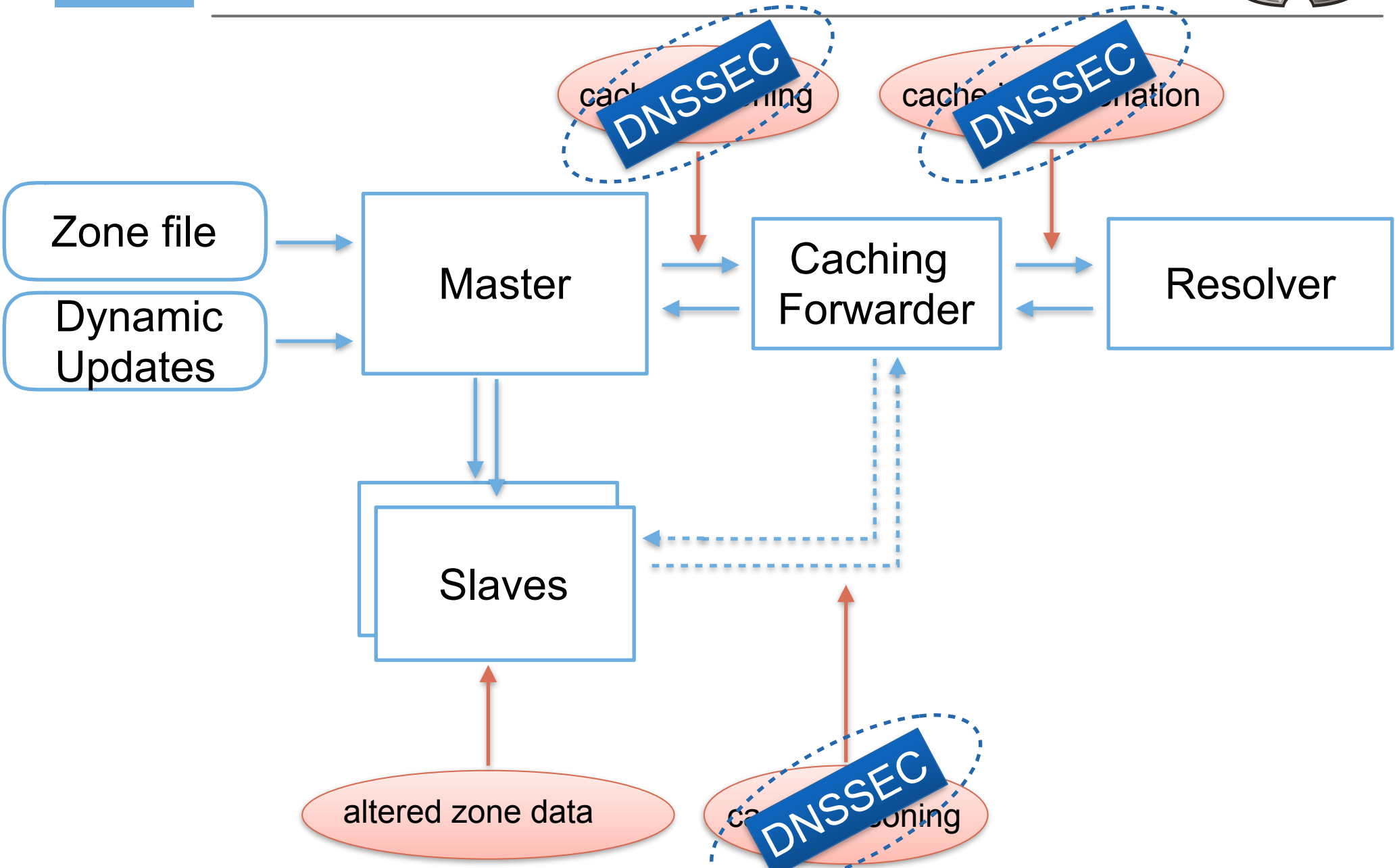
DNSSEC schützt vor diesen Attacken



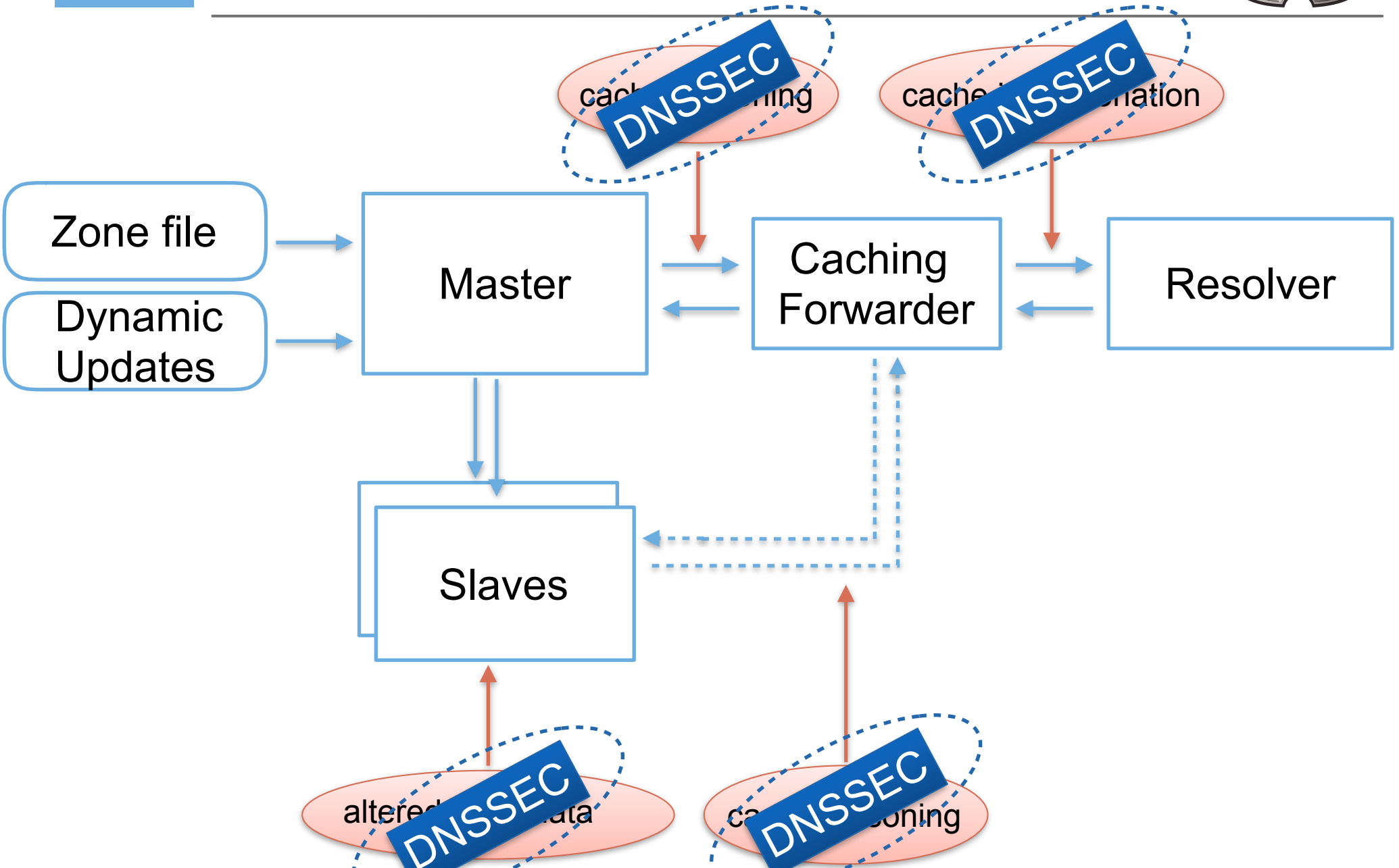
DNSSEC schützt vor diesen Attacken



DNSSEC schützt vor diesen Attacken



DNSSEC schützt vor diesen Attacken

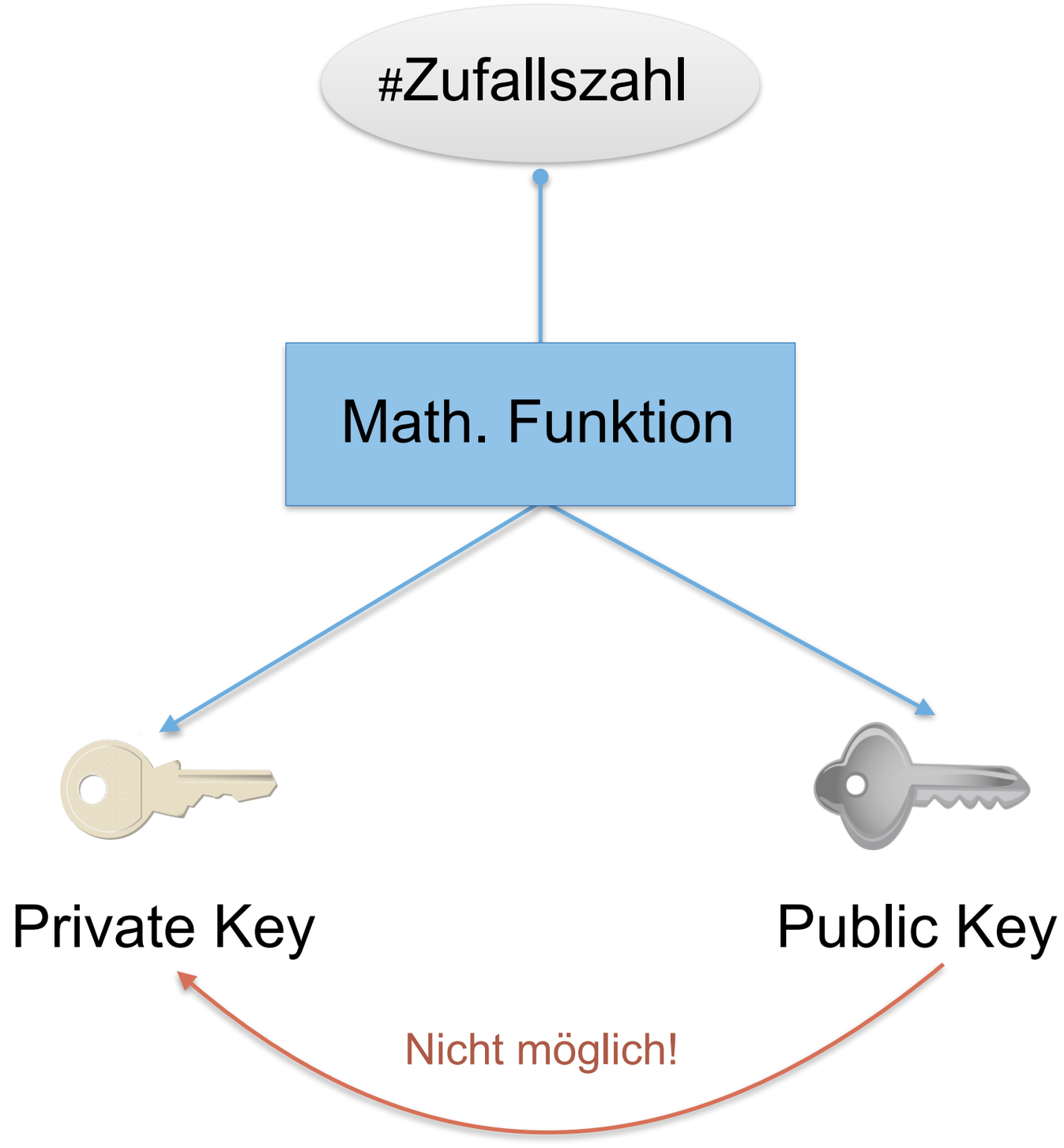


lrz

Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Public Key Kryptographie & DNSSEC

Public Key Kryptographie - Schlüssel erzeugen



Public Key Kryptographie - Hash erzeugen

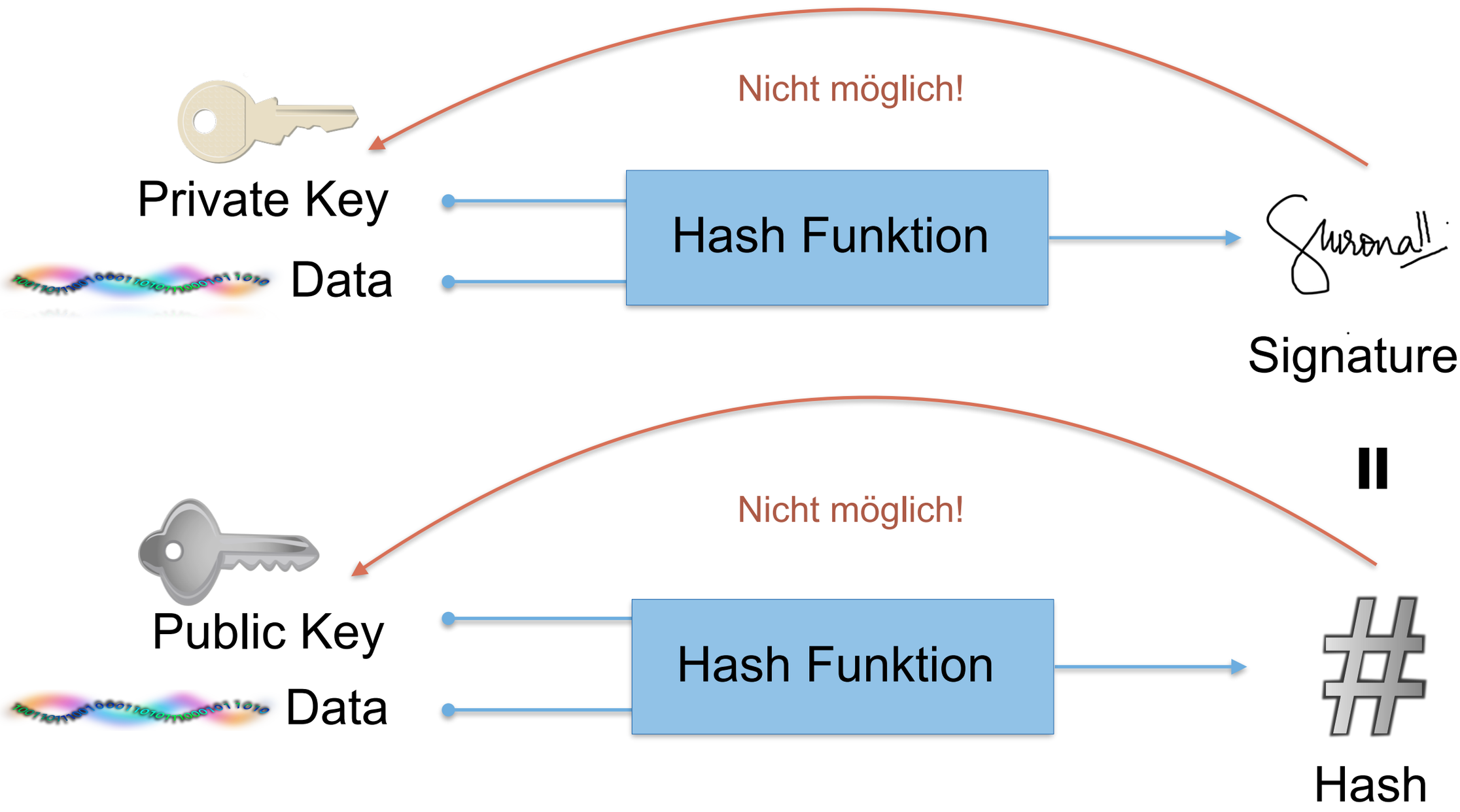
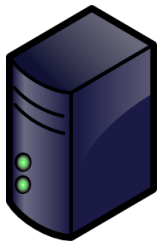


Bild: RIPE NCC

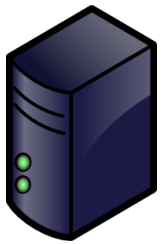
DNSSEC - Funktionsweise

Authoritative Nameserver

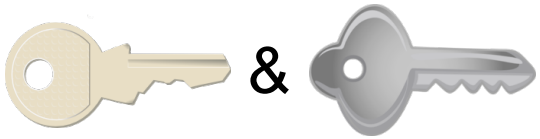


DNSSEC - Funktionsweise

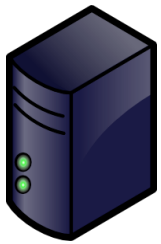
Authoritative Nameserver



- PKI basiert: **privater** und **öffentlicher** Schlüssel



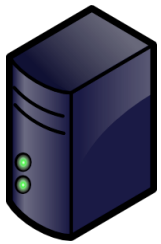
DNSSEC - Funktionsweise

Authoritative Nameserver 

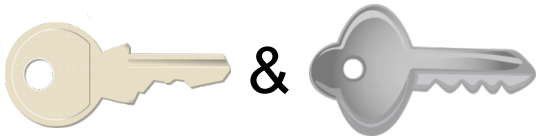
- PKI basiert: **privater** und **öffentlicher** Schlüssel  & 
- Antworten sind NICHT verschlüsselt 

DNSSEC - Funktionsweise

Authoritative Nameserver

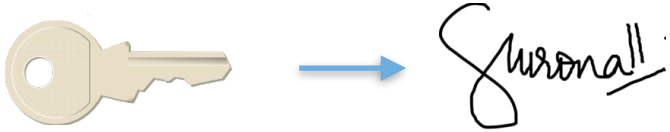


- PKI basiert: **privater** und **öffentlicher** Schlüssel



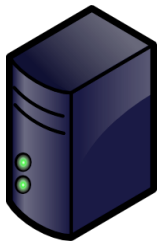
- Antworten sind NICHT verschlüsselt

- Antworten mit **geheimen** Schlüssel signiert

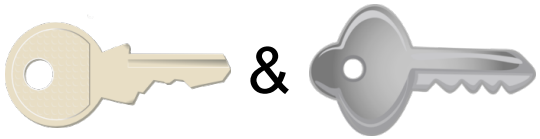


DNSSEC - Funktionsweise

Authoritative Nameserver

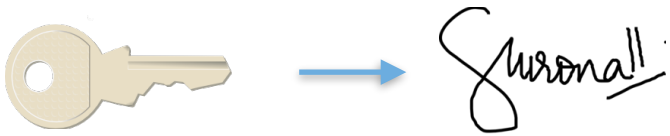


- PKI basiert: **privater** und **öffentlicher** Schlüssel



- Antworten sind NICHT verschlüsselt

- Antworten mit **geheimen** Schlüssel signiert

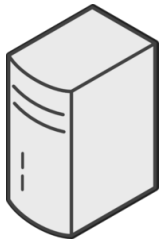


- **Signatur** wird mit DNS Antwort übertragen

IP +

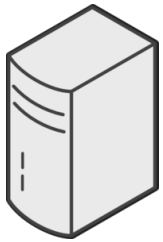
DNSSEC - Funktionsweise

Resolving Nameserver



DNSSEC - Funktionsweise

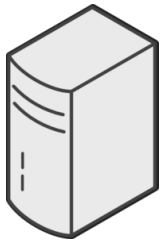
Resolving Nameserver



- Resolving NS überprüfen Signatur anhand des **Hashes** (errechnet aus Daten + **öffentlichem** Schlüssel)

DNSSEC - Funktionsweise

Resolving Nameserver

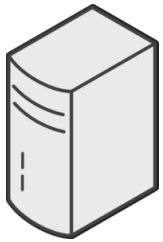


- Resolving NS überprüfen Signatur anhand des **Hashes** (errechnet aus Daten + **öffentlichem** Schlüssel)

- Errechneter Hash = Signatur der DNS Antwort?

DNSSEC - Funktionsweise

Resolving Nameserver

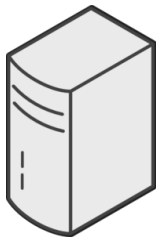


- Resolving NS überprüfen Signatur anhand des **Hashes** (errechnet aus Daten + **öffentlichem** Schlüssel)

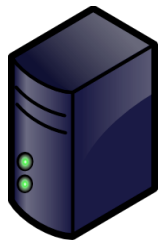
- Errechneter Hash = Signatur der DNS Antwort?

Ja ! → DNS Antwort authentisch von diesem DNS-Server

DNSSEC-Abfrage im Detail



resolving
nameserver



authoritative
nameserver

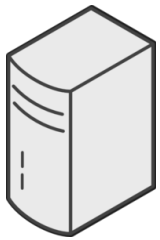
Zone lrz.de.

129.187.4.40

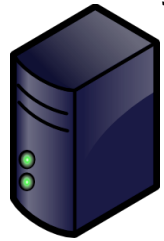
DNSSEC-Abfrage im Detail

1. author. NS erzeugt Keys

public key



resolving nameserver



authoritative nameserver

Zone lrz.de.

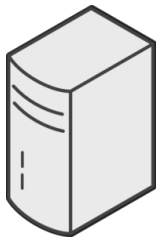
129.187.4.40

private key

DNSSEC-Abfrage im Detail

1. author. NS erzeugt Keys
2. author. NS signiert Zone

public key



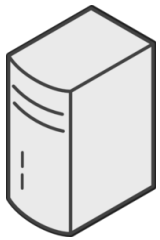
resolving nameserver

authoritative nameserver

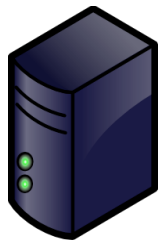
private key

DNSSEC-Abfrage im Detail

1. author. NS erzeugt Keys
2. author. NS signiert Zone
3. resolv. NS empfängt PK



resolving nameserver



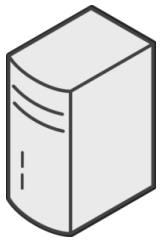
authoritative nameserver

public key

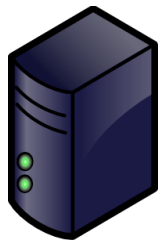
private key

DNSSEC-Abfrage im Detail

1. author. NS erzeugt Keys
2. author. NS signiert Zone
3. resolv. NS empfängt PK
4. confluence.lrz.de?



resolving nameserver



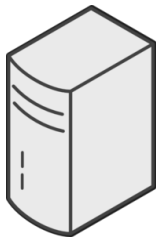
authoritative nameserver

public key

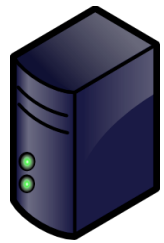
private key

DNSSEC-Abfrage im Detail

1. author. NS erzeugt Keys
2. author. NS signiert Zone
3. resolv. NS empfängt PK
4. confluence.lrz.de?
5. resolv. NS empfängt DNSSEC Paket



resolving nameserver



authoritative nameserver

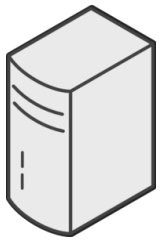
IP

public key

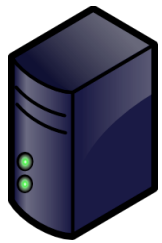
private key

DNSSEC-Abfrage im Detail

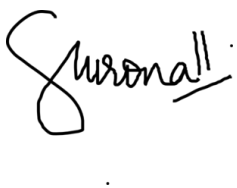
1. author. NS erzeugt Keys
2. author. NS signiert Zone
3. resolv. NS empfängt PK
4. confluence.lrz.de?
5. resolv. NS empfängt DNSSEC Paket
6. resolv. NS errechnet Hash aus public key



resolving nameserver



authoritative nameserver



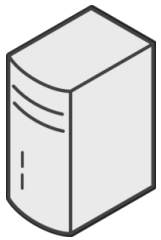
public key

IP

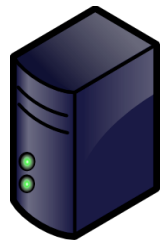
private key

DNSSEC-Abfrage im Detail

1. author. NS erzeugt Keys
2. author. NS signiert Zone
3. resolv. NS empfängt PK
4. confluence.lrz.de?
5. resolv. NS empfängt DNSSEC Paket
6. resolv. NS errechnet Hash aus public key
7. Hash = RRSIG?



resolving nameserver



authoritative nameserver

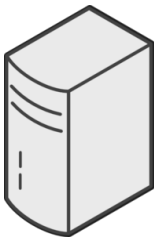
public key

IP

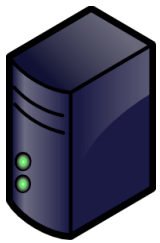
private key

DNSSEC-Abfrage im Detail

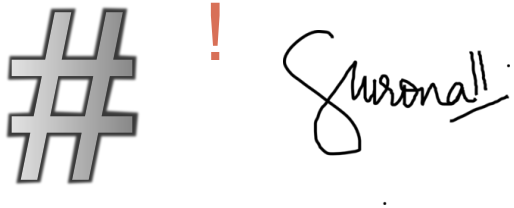
1. author. NS erzeugt Keys
2. author. NS signiert Zone
3. resolv. NS empfängt PK
4. confluence.lrz.de?
5. resolv. NS empfängt DNSSEC Paket
6. resolv. NS errechnet Hash aus public key
7. Hash = RRSIG?
8. IP von authoritative NS ist authentisch!



resolving nameserver



authoritative nameserver



public key

private key

IP

authentisch


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
DNSSEC Records und Zusammenhänge

Neue Resource Records für DNSSEC

- DNSSKEY - DNSSEC Public Key
- RRSIG - Signatur über RRSet
- DS - Delegated Signer, sichere Delegation
- NSEC - Next Secure, nächster sicherer Eintrag
- NSEC3 - Next Secure rehashed

DNSSEC - Schlüssel

- Schlüssel wird benutzt, um DNS Einträge zu signieren
- Jeder einzelne Eintrag in der Zone wird signiert
- Ohne Delegating Signing Authority müsste der Resolver Millionen an Schlüsseln speichern
- Mit Delegated Signing muß nur ein Schlüssel vorgehalten werden: root Schlüssel

DNSKEY - Algorithmen

- mögliche Algorithmen
  - RSAMD5
  - DH
  - RSASHA1
  - DSA-NSEC3-SHA1
  - RSASHA1-NSEC3-SHA1
  - RSASHA256
  - RSASHA512
  - ECC-GOSTECDSAP256SHA256
  - ECC-GOSTECDSAP256SHA384

DNSKEY - Algorithmen

- mögliche Algorithmen
  - RSAMD5
  - DH
  - RSASHA1
  - DSA-NSEC3-SHA1
  - RSASHA1-NSEC3-SHA1
  - RSASHA256
  - RSASHA512
  - ECC-GOSTECDSAP256SHA256
  - ECC-GOSTECDSAP256SHA384

unsicher!

DNSKEY - Algorithmen

- mögliche Algorithmen
  - RSAMD5
  - DH
  - RSASHA1
  - DSA-NSEC3-SHA1
  - RSASHA1-NSEC3-SHA1
  - RSASHA256
  - RSASHA512
  - ECC-GOSTECDSAP256SHA256
  - ECC-GOSTECDSAP256SHA384

meist verwendet

DNSKEY - Algorithmen

- mögliche Algorithmen
  - RSAMD5
  - DH
  - RSASHA1
  - DSA-NSEC3-SHA1
  - RSASHA1-NSEC3-SHA1
  - RSASHA256
  - RSASHA512
  - ECC-GOSTECDSAP256SHA256
  - ECC-GOSTECDSAP256SHA384

← kürzere Schlüssellängen

DNSKEY - Algorithmen

- mögliche Algorithmen
  - RSAMD5
  - DH
  - RSASHA1
  - DSA-NSEC3-SHA1
  - RSASHA1-NSEC3-SHA1
  - RSASHA256
  - RSASHA512
  - ECC-GOSTECDSAP256SHA256
  - ECC-GOSTECDSAP256SHA384

DNSSEC - Schlüssel

- DNS besteht aus Zonen, die Vertrauen garantieren, mit Delegation an andere dieser Zonen

- “Elternteil” (“Parent”) muss auf die Schlüssel der Kinder (“children keys”) verweisen
  - um diese zu signieren
  - DS Records bewerkstelligen dies

- Notwendige Interaktion zwischen Parent und Child sollte minimal sein

DS Records - Delegation of Authority

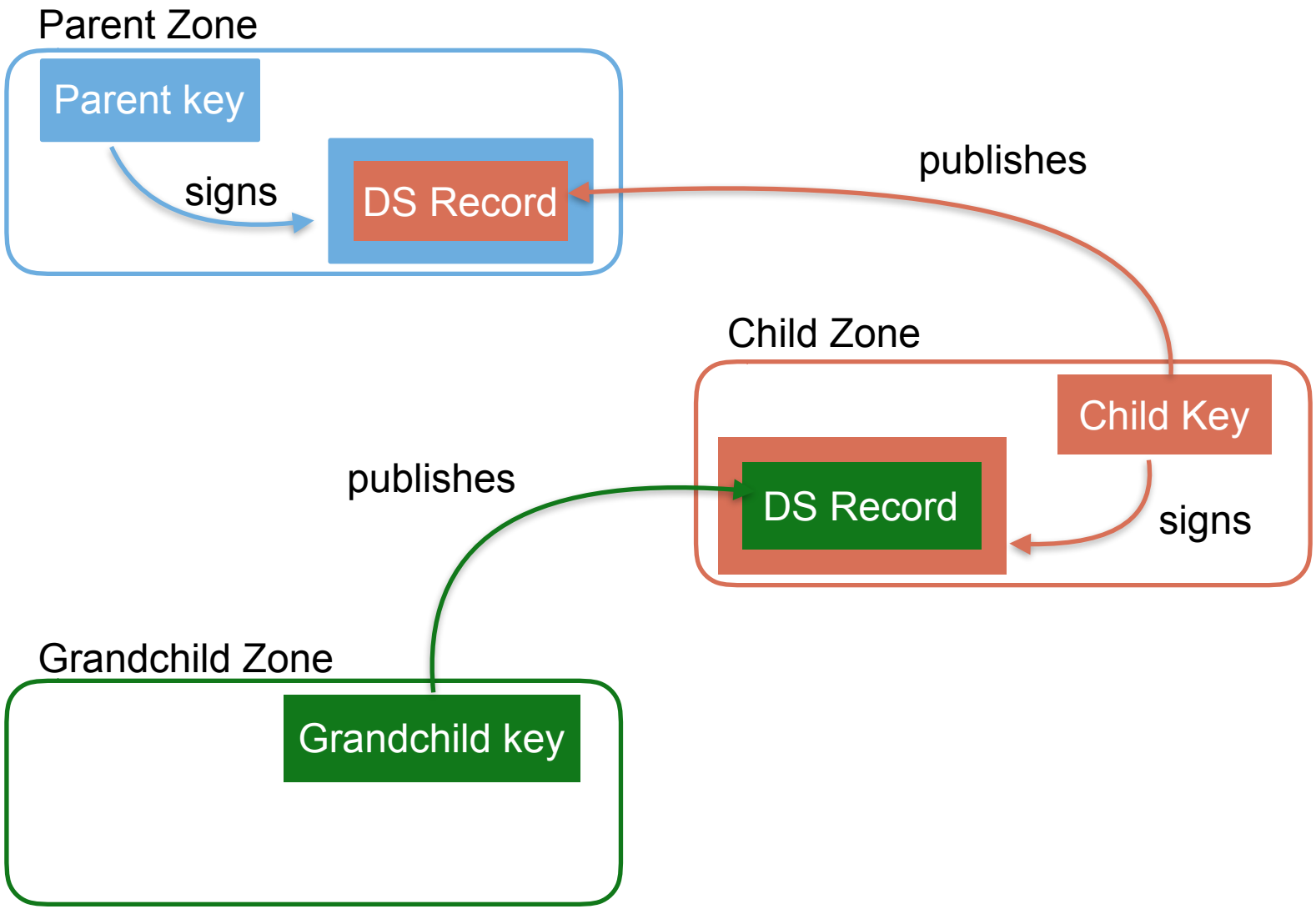


Bild: RIPE NCC

„Chain-of-trust“

So entsteht eine durchgehende Kette des Vertrauens.

Einer signierten DNS-Antwort, die aus Antworten aller beteiligten Zonen besteht,…

… kann vollständig vertraut werden.

Die Antwort ist damit **authentisch**.

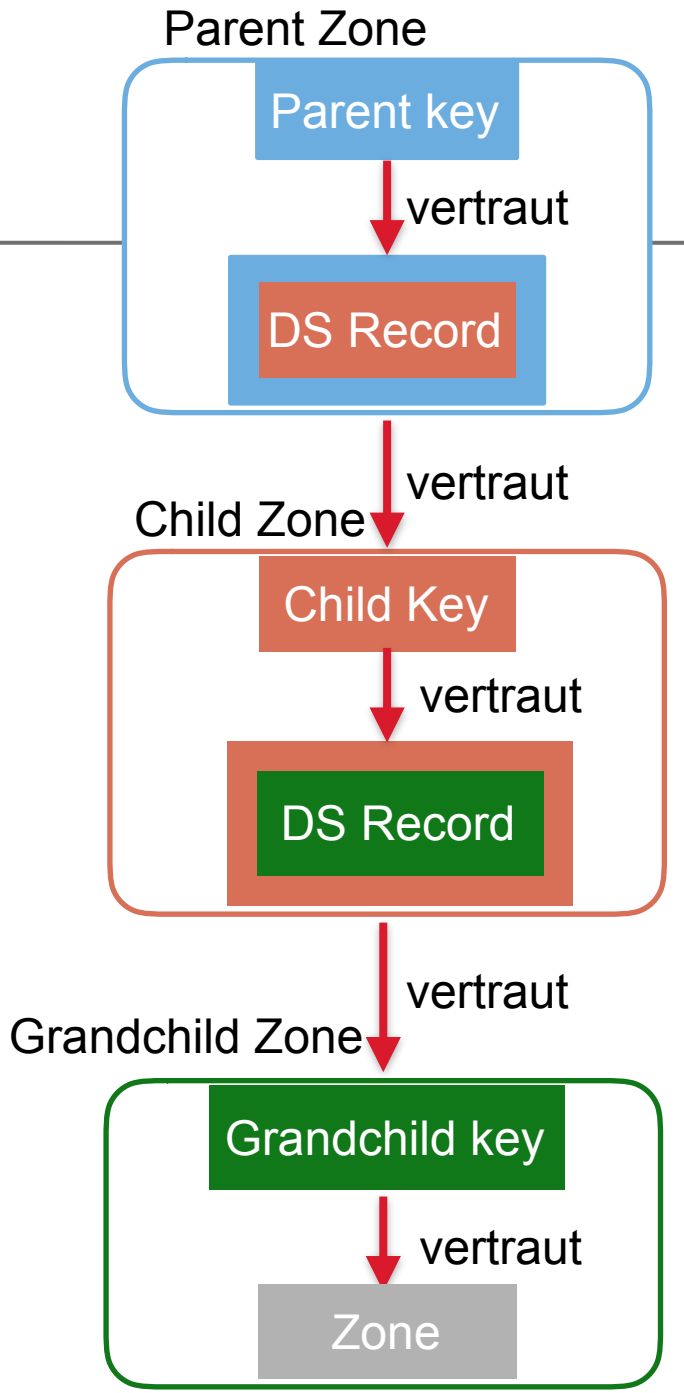


Bild: RIPE NCC

Schlüssel-Probleme

- Administration der Schlüsselhandhabung mit Parent zone aufwändig
  - Durchführung nur wenn nötig
  - Längere Schlüssel sind sicherer
- Signieren der Zonen soll schnell sein
  - Speicherlimitierung
  - Disk space und Zeitbedarf
  - Kürzere Schlüssel mit kurzen Lebensdauern sind besser

Schlüssel - funktionale Anforderungen

- Lange Schlüssel sind sicherer

  - können länger verwendet werden

  - lange Signaturen große Zone-Dateien

  - Signieren und Verifizierung rechenintensiv

- Kurze Schlüssel sind schneller

  - kurze Signaturen

  - Signieren und Verifizierung weniger aufwändig

  - kurze Lebensdauer

Lösung: Mehr als ein Schlüssel!

- Key Signing Key (KSK) - signiert nur DNSKEY RRset
- Zone Signing Key (ZSK) - signiert alle RRsets in der Zone

- RRsets werden signiert, nicht RRs
- DS verweist auf KSK des Kindes

  - Parent ZSK signiert DS

  - Signatur überträgt Vertrauen von Parent auf den Child key

ZSK und KSK in den DS Records

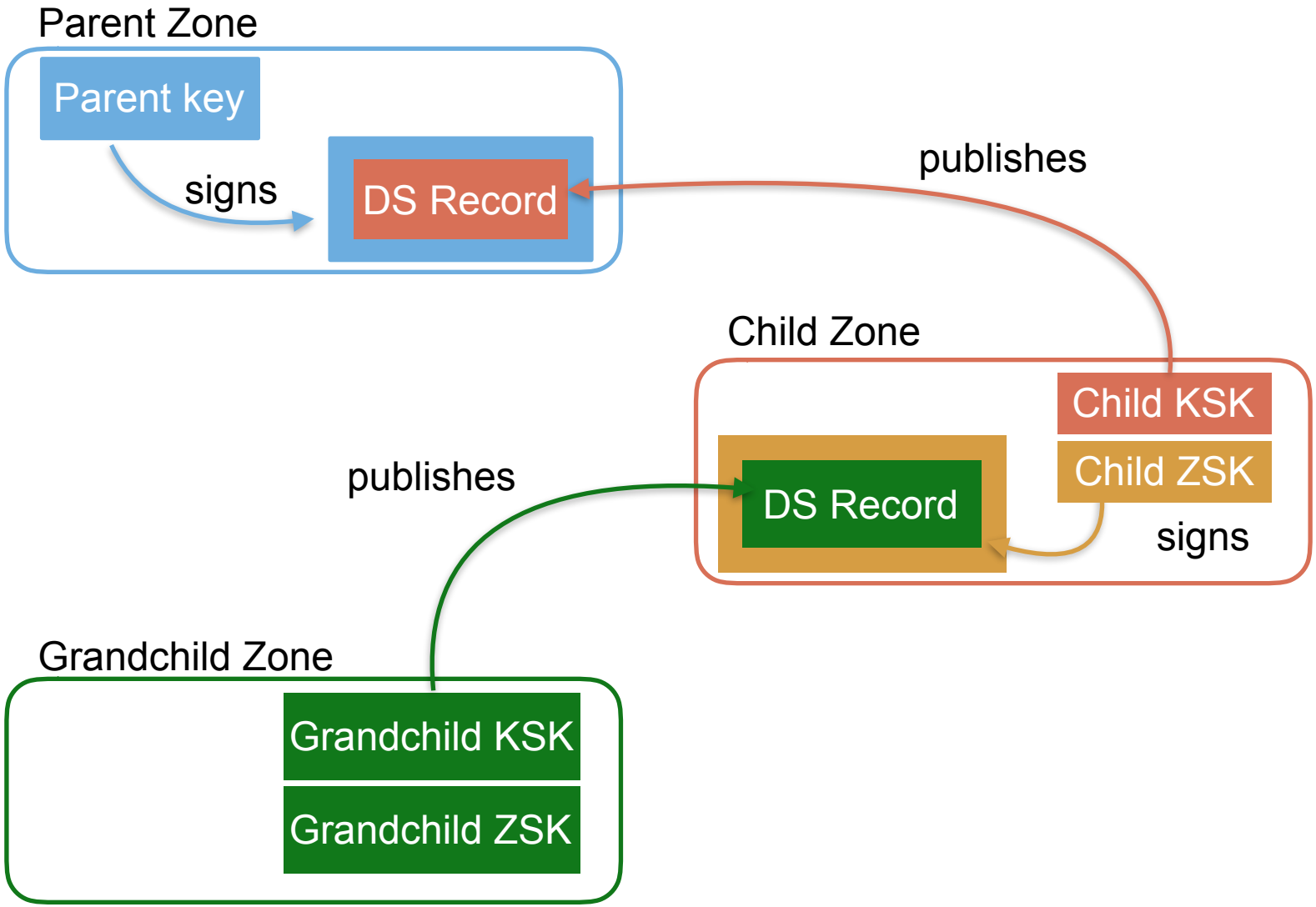


Bild: RIPE NCC

DNSSEC - Schlüssel: Zone Signing Key

Zone Signing Key (ZSK)

2 Files, public and private key:

zsk.key

zsk.pub

- **signiert Zonen (RRSets)**
- signiert jeden einzelnen Eintrag in einer Zone
- im Allgemeinen 1024 Bit RSA Key (LRZ 2048 Bit)
- Schlüssel wird kurze Zeit genutzt (~ 3 Monate)
  (Gültigkeit in den Meta-Daten)

DNSSEC - Schlüssel: Key Signing Key

Key Signing Key (KSK)

- **signiert den Zone Signing Key**
- public KSK wird veröffentlicht via DS Einträge
- DS-Nameserver nehmen dann diesen Public Key auf
- im Allgemeinen 2048 Bit RSA Key (LRZ 2048 Bit)
- Schlüssel wird nur begrenzte Zeit genutzt (~ 2 Jahre)
  (Gültigkeit in den Meta-Daten)

2 Files, public and private key:

ksk.key

ksk.pub

Zone Signing - Delegated Signer / Trust anchor

- DS (Delegated Signer) enthält den Public KSK des folgenden Nameservers im Baum („Kind“)
- DS von Großvater 

Vater 

Kind (**„chain of trust“**)
- . root-Nameserver ist „trust anchor“
- DNS-Abfrage ist nur dann sicher, wenn alle Nameserver mit DNSSEC ihre Zonen signiert haben

Delegated Signer - Initialer Schlüsseltausch

- Child muß Key Singing Keys (KSKs) an Parent schicken
- Parent muss:

  - in der Zone des Child DNSKEY und RRSIGs überprüfen

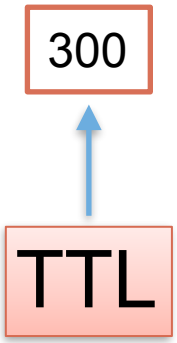
  - Verifizieren, dass dem Schlüssel getraut werden kann

  - DS RR erzeugen

DNSKEY Resource Record (ZSK)

```
300     DNSKEY          256 3 8 (
                        AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
                        5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
                        xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
                        i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
                        eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
                        eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
                        QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
                        cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
                        CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
                        wKZud8q7SFukpslVKcFLrZc=
                        ) ; ZSK; alg = RSASHA256; key id = 56961
```

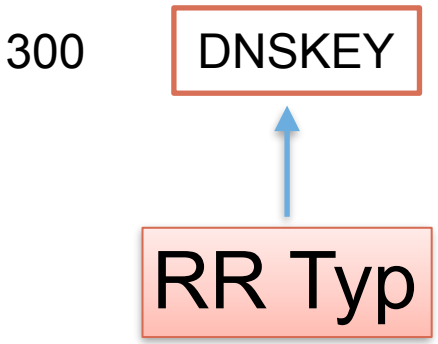
DNSKEY Resource Record (ZSK)



DNSKEY 256 3 8 (

AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
wKZud8q7SFukpslVKcFLrZc=
) ; ZSK; alg = RSASHA256; key id = 56961

DNSKEY Resource Record (ZSK)

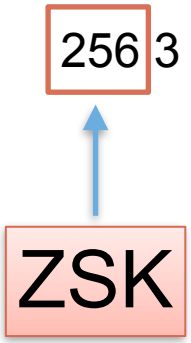


256 3 8 (

AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
wKZud8q7SFukpslVKcFLrZc=
) ; ZSK; alg = RSASHA256; key id = 56961

DNSKEY Resource Record (ZSK)

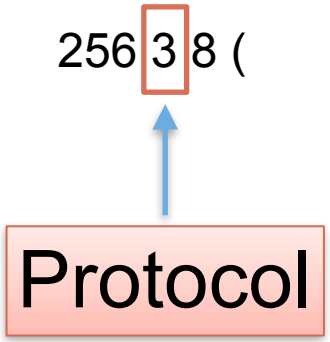
300 DNSKEY 256 3 8 (



AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
wKZud8q7SFukpslVKcFLrZc=
) ; ZSK; alg = RSASHA256; key id = 56961

DNSKEY Resource Record (ZSK)

300 DNSKEY 256 3 8 (



AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
wKZud8q7SFukpslVKcFLrZc=
) ; ZSK; alg = RSASHA256; key id = 56961

DNSKEY Resource Record (ZSK)

```
300     DNSKEY          256 3 8 (
AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
wKZud8q7SFukpslVKcFLrZc=
) ; ZSK; alg = RSASHA256; key id = 56961
```

Algorithm: RSA

DNSKEY Resource Record (ZSK)

300 DNSKEY 256 3 8 (

AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
wKZud8q7SFukpslVKcFLrZc=
) ; ZSK; alg = RSASHA256; key id = 56961

Signatur Hash

DNSKEY Resource Record (ZSK)

```
300     DNSKEY      256 3 8 (
                    AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
                    5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
                    xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
                    i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
                    eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
                    eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
                    QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
                    cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
                    CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
                    wKZud8q7SFukpslVKcFLrZc=
                    ) ; ZSK; alg = RSASHA256; key id = 56961
```

Kommentar mit Typ,Alg und key id!

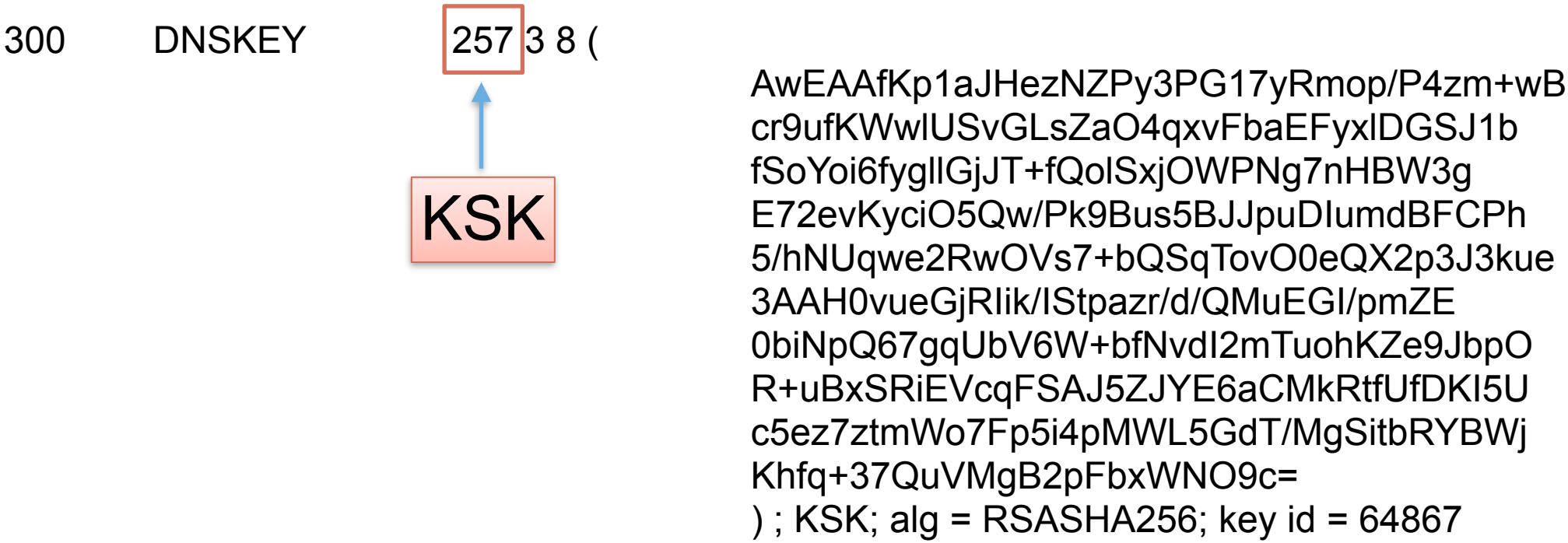
DNSKEY Resource Record (ZSK)

```
300     DNSKEY          256 3 8 (
                        AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z
                        5F8ArvpzOdx4o+/ef03DNon3pZt855P47fcY
                        xX3vIrsd1Jl+au1CIGaxwIAspWBoIyGqKofR
                        i01DhJeTWaZbgeipLoJmz/TjSM8cgJtDmgUO
                        eb9tLv25XNuktrq5q82809QINlSvFc+dr8tI
                        eoLuwBG3uPd/wgVzSLo9an6WDeOr1v6NtYKP
                        QzlTY7Hsyu0mitIz6OAn8z5yaB+KAcNkz6p1
                        cFXX7XJlFE0tnfvIIjjAV2Rrp3gCylDIc2QL
                        CYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTc
                        wKZud8q7SFukpslVKcFLrZc=
                        ) ; ZSK; alg = RSASHA256; key id = 56961
```

DNSKEY Resource Record (KSK)

```
300     DNSKEY          257 3 8 (
                        AwEAAfKp1aJHezNZPy3PG17yRmop/P4zm+wB
                        cr9ufKWwlUSvGLsZaO4qxvFbaEFyxlDGSJ1b
                        fSoYoi6fyglIGjJT+fQolSxjOWPNg7nHBW3g
                        E72evKyciO5Qw/Pk9Bus5BJJpuDIumdBFCPh
                        5/hNUqwe2RwOVs7+bQSqTovO0eQX2p3J3kue
                        3AAH0vueGjRIik/IStpazr/d/QMuEGI/pmZE
                        0biNpQ67gqUbV6W+bfNvdI2mTuohKZe9JbpO
                        R+uBxSRiEVcqFSAJ5ZJYE6aCMkRtfUfDKI5U
                        c5ez7ztmWo7Fp5i4pMWL5GdT/MgSitbRYBWj
                        Khfq+37QuVMgB2pFbxWNO9c=
                        ) ; KSK; alg = RSASHA256; key id = 64867
```

DNSKEY Resource Record (KSK)



DNSKEY Resource Record (KSK)

```
300     DNSKEY      257 3 8 (
                    AwEAAfKp1aJHezNZPy3PG17yRmop/P4zm+wB
                    cr9ufKWwlUSvGLsZaO4qxvFbaEFyxlDGSJ1b
                    fSoYoi6fyglIGjJT+fQolSxjOWPNg7nHBW3g
                    E72evKyciO5Qw/Pk9Bus5BJJpuDIumdBFCPh
                    5/hNUqwe2RwOVs7+bQSqTovO0eQX2p3J3kue
                    3AAH0vueGjRIik/IStpazr/d/QMuEGI/pmZE
                    0biNpQ67gqUbV6W+bfNvdI2mTuohKZe9JbpO
                    R+uBxSRiEVcqFSAJ5ZJYE6aCMkRtfUfDKI5U
                    c5ez7ztmWo7Fp5i4pMWL5GdT/MgSitbRYBWj
                    Khfq+37QuVMgB2pFbxWNO9c=
                    ) ; KSK; alg = RSASHA256; key id = 64867
```

sonst wie ZSK, key id: 64867!

DNSKEY Resource Record (KSK)

```
300     DNSKEY          257 3 8 (
                        AwEAAfKp1aJHezNZPy3PG17yRmop/P4zm+wB
                        cr9ufKWwlUSvGLsZaO4qxvFbaEFyxlDGSJ1b
                        fSoYoi6fyglIGjJT+fQolSxjOWPNg7nHBW3g
                        E72evKyciO5Qw/Pk9Bus5BJJpuDIumdBFCPh
                        5/hNUqwe2RwOVs7+bQSqTovO0eQX2p3J3kue
                        3AAH0vueGjRIik/IStpazr/d/QMuEGI/pmZE
                        0biNpQ67gqUbV6W+bfNvdI2mTuohKZe9JbpO
                        R+uBxSRiEVcqFSAJ5ZJYE6aCMkRtfUfDKI5U
                        c5ez7ztmWo7Fp5i4pMWL5GdT/MgSitbRYBWj
                        Khfq+37QuVMgB2pFbxWNO9c=
                        ) ; KSK; alg = RSASHA256; key id = 64867
```

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

```
300     RRSIG   A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
                          kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
                          cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
                          dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
                          gVJRElwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
                          IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
                          PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
                          kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
                          /6HaAmuaQ5R4jEUKUs1TS3C4dDsUlEf2WDly
                          a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
                          S7QUT0ZB/Haw19hdng== )
```

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

Name des Eintrags

```
300     RRSIG   A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
                          kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
                          cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
                          dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
                          gVJRElwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
                          IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
                          PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
                          kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
                          /6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
                          a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
                          S7QUT0ZB/Haw19hdng== )
```

A Resource Record mit RRSIG (Signatur)

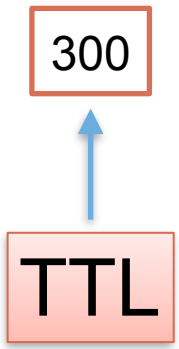
dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

wie gewohnt: TTL A-RR IPv4

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJREIwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

A Resource Record mit RRSIG (Signatur)

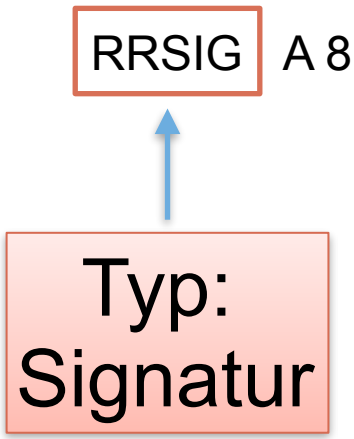
dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206



RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJRElwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUlEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206



300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJRElwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUlEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJREIwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

Signatur über A RR

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

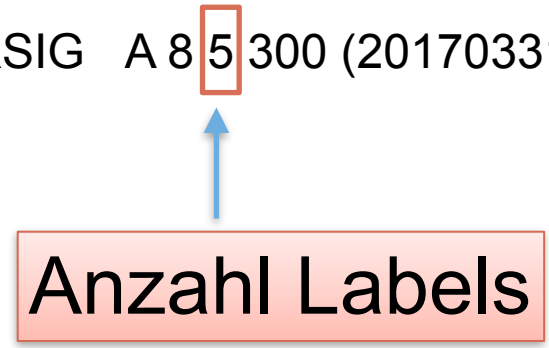
300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJRElwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

Schlüssel-Algo-rithmus: RSA

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJREIwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)



A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJRElwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

ursprüngliche TTL

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
AM47e/fLd5SR9vkcGRX4cWgY
XY5McZvF/vc6of9G1ofESTaDt
1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

Signatur Ende der Gültigkeit Datum/Zeit

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqqDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0Q3d5G2OPuD2fwxIt4
kcGRX4cWgY
9G1ofESTaDt
UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

Signatur Start der Gültigkeit Datum/Zeit

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJREIwaJOg7A SR9vkcGRX4cWgY
IM+pmGDO2F5X c6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

Key id

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJREIwaJOg7AM47e
IM+pmGDO2F5XY5Mc
PYdqIhouAEDc1/gL41z
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUIEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

Zone, in der die Signatur gilt

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

300 RRSIG A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
gVJRElwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
/6HaAmuaQ5R4jEUKUs1TS3C4dDsUlEf2WDly
a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
S7QUT0ZB/Haw19hdng==)

Hash der Signatur

A Resource Record mit RRSIG (Signatur)

dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206

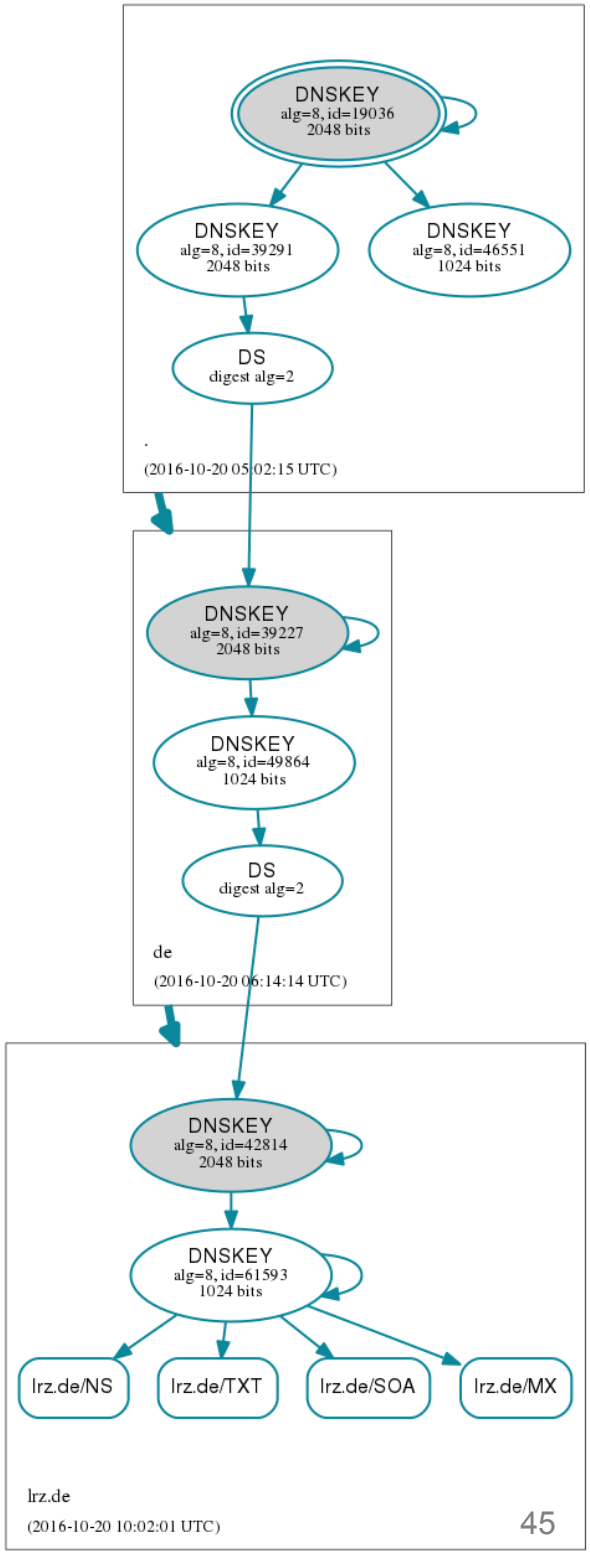
```
300     RRSIG   A 8 5 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
                          kXvoUzzqgDNuST0cxsFG9Vqmndb/MsWZQTS7
                          cmt6Z8o8wk3VwDqWucJhWsFLhLsZbJmsFFOY
                          dPuArIGmDBZawnwksc0O3d5G2OPuD2fwxIt4
                          gVJRElwaJOg7AM47e/fLd5SR9vkcGRX4cWgY
                          IM+pmGDO2F5XY5McZvF/vc6of9G1ofESTaDt
                          PYdqIhouAEDc1/gL41zLBNorb76UneUKtc9T
                          kvFK0vQGC2b/4Vy1XyGQwfQKII3zQJBRQMB1
                          /6HaAmuaQ5R4jEUKUs1TS3C4dDsUlEf2WDly
                          a6sHmGDC0yik1pbg0nfS4a2/HZqfiKs3xF5e
                          S7QUT0ZB/Haw19hdng== )
```


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
DNSSEC-Test mit DNSViz & Debugger

DNSViz Tool - dnsviz.net

- Web tool zur Überprüfung von DNSSEC
- DNSKEYs und RRsets
- Durchlauf der „chain of trust“
- Delegated Signer graphische Darstellung

Demonstration in der folgenden Übung



DNSSEC Debugger

- Tabellarische Information welcher Teil fehlschlägt:

  http://dnssec-debugger.verisignlabs.com/

- Erlaubt Fehleranalyse für pro Zone in „chain-of-trust“

  DNSKey (DNS Zone signing key)

  DS (Delegated Signer)

  RRSIG (Resource Record Signature)

- zeigt nicht welcher Schlüssel welche RRSets signiert

- unübersichtlich bei mehreren Schlüsseln

Beispiel:

Analyzing DNSSEC problems for mpe.mpg.de

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-1 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
de	✓ Found 1 DS records for de in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=39291 and DNSKEY=39291 verifies the DS RRset ✓ Found 3 DNSKEY records for de ✓ DS=39227/SHA-256 verifies DNSKEY=39227/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=39227 and DNSKEY=39227/SEP verifies the DNSKEY RRset
mpg.de	✓ Found 2 DS records for mpg.de in the de zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=44919 and DNSKEY=44919 verifies the DS RRset ✓ Found 2 DNSKEY records for mpg.de ✓ DS=40326/SHA-256 verifies DNSKEY=40326/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=13895 and DNSKEY=13895 verifies the DNSKEY RRset
mpe.mpg.de	✗ No DS records found for mpe.mpg.de in the mpg.de zone ✗ No DNSKEY records found ✓ mpe.mpg.de A RR has value 134.76.31.205 ✗ No RRSIGs found

Beispiel:

.root DNSSEC verifiziert

Analyzing DNSSEC problems for mpe.mpg.de

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-1 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
de	✓ Found 1 DS records for de in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=39291 and DNSKEY=39291 verifies the DS RRset ✓ Found 3 DNSKEY records for de ✓ DS=39227/SHA-256 verifies DNSKEY=39227/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=39227 and DNSKEY=39227/SEP verifies the DNSKEY RRset
mpg.de	✓ Found 2 DS records for mpg.de in the de zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=44919 and DNSKEY=44919 verifies the DS RRset ✓ Found 2 DNSKEY records for mpg.de ✓ DS=40326/SHA-256 verifies DNSKEY=40326/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=13895 and DNSKEY=13895 verifies the DNSKEY RRset
mpe.mpg.de	✗ No DS records found for mpe.mpg.de in the mpg.de zone ✗ No DNSKEY records found ✓ mpe.mpg.de A RR has value 134.76.31.205 ✗ No RRSIGs found

Beispiel:

.root DNSSEC verifiziert

Analyzing DNSSEC problems for mpe.mpg.de

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-1 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
de	✓ Found 1 DS records for de in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=39291 and DNSKEY=39291 verifies the DS RRset ✓ Found 3 DNSKEY records for de ✓ DS=39227/SHA-256 verifies DNSKEY=39227/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=39227 and DNSKEY=39227/SEP verifies the DNSKEY RRset
mpg.de	✓ Found 2 DS records for mpg.de in the de zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=44919 and DNSKEY=44919 verifies the DS RRset ✓ Found 2 DNSKEY records for mpg.de ✓ DS=40326/SHA-256 verifies DNSKEY=40326/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=13895 and DNSKEY=13895 verifies the DNSKEY RRset
mpe.mpg.de	✗ No DS records found for mpe.mpg.de in the mpg.de zone ✗ No DNSKEY records found ✓ mpe.mpg.de A RR has value 134.76.31.205 ✗ No RRSIGs found

Beispiel:

.root DNSSEC verifiziert

.de TLD Domäne ist korrekt signiert und authentifiziert. DNSKEY, DS und RRSIG verifiziert:

Analyzing DNSSEC problems for mpe.mpg.de

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-1 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
de	✓ Found 1 DS records for de in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=39291 and DNSKEY=39291 verifies the DS RRset ✓ Found 3 DNSKEY records for de ✓ DS=39227/SHA-256 verifies DNSKEY=39227/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=39227 and DNSKEY=39227/SEP verifies the DNSKEY RRset
mpg.de	✓ Found 2 DS records for mpg.de in the de zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=44919 and DNSKEY=44919 verifies the DS RRset ✓ Found 2 DNSKEY records for mpg.de ✓ DS=40326/SHA-256 verifies DNSKEY=40326/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=13895 and DNSKEY=13895 verifies the DNSKEY RRset
mpe.mpg.de	✗ No DS records found for mpe.mpg.de in the mpg.de zone ✗ No DNSKEY records found ✓ mpe.mpg.de A RR has value 134.76.31.205 ✗ No RRSIGs found

Beispiel:

.root DNSSEC verifiziert

Analyzing DNSSEC problems for mpe.mpg.de

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-1 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
de	✓ Found 1 DS records for de in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=39291 and DNSKEY=39291 verifies the DS RRset ✓ Found 3 DNSKEY records for de ✓ DS=39227/SHA-256 verifies DNSKEY=39227/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=39227 and DNSKEY=39227/SEP verifies the DNSKEY RRset
mpg.de	✓ Found 2 DS records for mpg.de in the de zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=44919 and DNSKEY=44919 verifies the DS RRset ✓ Found 2 DNSKEY records for mpg.de ✓ DS=40326/SHA-256 verifies DNSKEY=40326/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=13895 and DNSKEY=13895 verifies the DNSKEY RRset
mpe.mpg.de	✗ No DS records found for mpe.mpg.de in the mpg.de zone ✗ No DNSKEY records found ✓ mpe.mpg.de A RR has value 134.76.31.205 ✗ No RRSIGs found

Beispiel:

.root DNSSEC verifiziert

.de TLD Domäne ist korrekt signiert und authentifiziert. DNSKEY, DS und RRSIG verifiziert:

Max-Planck Gesellschaft Domäne ist korrekt signiert und authentifiziert.

Analyzing DNSSEC problems for mpe.mpg.de

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-1 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
de	✓ Found 1 DS records for de in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=39291 and DNSKEY=39291 verifies the DS RRset ✓ Found 3 DNSKEY records for de ✓ DS=39227/SHA-256 verifies DNSKEY=39227/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=39227 and DNSKEY=39227/SEP verifies the DNSKEY RRset
mpg.de	✓ Found 2 DS records for mpg.de in the de zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=44919 and DNSKEY=44919 verifies the DS RRset ✓ Found 2 DNSKEY records for mpg.de ✓ DS=40326/SHA-256 verifies DNSKEY=40326/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=13895 and DNSKEY=13895 verifies the DNSKEY RRset
mpe.mpg.de	✗ No DS records found for mpe.mpg.de in the mpg.de zone ✗ No DNSKEY records found ✓ mpe.mpg.de A RR has value 134.76.31.205 ✗ No RRSIGs found

Beispiel:

.root DNSSEC verifiziert

Analyzing DNSSEC problems for mpe.mpg.de

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-1 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
de	✓ Found 1 DS records for de in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=39291 and DNSKEY=39291 verifies the DS RRset ✓ Found 3 DNSKEY records for de ✓ DS=39227/SHA-256 verifies DNSKEY=39227/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=39227 and DNSKEY=39227/SEP verifies the DNSKEY RRset
mpg.de	✓ Found 2 DS records for mpg.de in the de zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=44919 and DNSKEY=44919 verifies the DS RRset ✓ Found 2 DNSKEY records for mpg.de ✓ DS=40326/SHA-256 verifies DNSKEY=40326/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=13895 and DNSKEY=13895 verifies the DNSKEY RRset
mpe.mpg.de	✗ No DS records found for mpe.mpg.de in the mpg.de zone ✗ No DNSKEY records found ✓ mpe.mpg.de A RR has value 134.76.31.205 ✗ No RRSIGs found

MPE-Domäne (Max-Planck Institut für Extraterrestrische Physik) hat keinen DNSKey oder DS, damit auch keine RRSIGs.

Beispiel:

.root DNSSEC verifiziert

.de TLD Domäne ist korrekt signiert und authentifiziert. DNSKEY, DS und RRSIG verifiziert:

Max-Planck Gesellschaft Domäne ist korrekt signiert und authentifiziert.

MPE-Domäne (Max-Planck Institut für Extraterrestrische Physik) hat keinen DNSKey oder DS, damit auch keine RRSIGs.

Analyzing DNSSEC problems for mpe.mpg.de

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-1 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
de	✓ Found 1 DS records for de in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=39291 and DNSKEY=39291 verifies the DS RRset ✓ Found 3 DNSKEY records for de ✓ DS=39227/SHA-256 verifies DNSKEY=39227/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=39227 and DNSKEY=39227/SEP verifies the DNSKEY RRset
mpg.de	✓ Found 2 DS records for mpg.de in the de zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=44919 and DNSKEY=44919 verifies the DS RRset ✓ Found 2 DNSKEY records for mpg.de ✓ DS=40326/SHA-256 verifies DNSKEY=40326/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=13895 and DNSKEY=13895 verifies the DNSKEY RRset
mpe.mpg.de	✗ No DS records found for mpe.mpg.de in the mpg.de zone ✗ No DNSKEY records found ✓ mpe.mpg.de A RR has value 134.76.31.205 ✗ No RRSIGs found

Übung - DNSViz/Debugger DNSSEC-Überprüfung

Testen Sie die folgenden Domains

Mit dnsviz.net und finden Sie mit dnssec-debugger.verisign.com heraus, wo DNSSEC gegebenenfalls fehl schlägt:

- tu-muenchen.de
- physik.uni-muenchen.de
- mpa-garching.mpg.de
- oder eigene Zone


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
DNSSEC am Beispiel von Bind 9.9

DNSSEC mit BIND - Schritt für Schritt

1. Zone-Signing Key erzeugen (ZSK)
2. Key-Signing Key erzeugen (KSK)
3. Zone signieren
4. DS Record im Parent setzen
5. Zone veröffentlichen

Zone Signing - Zone Signing Key

Zone Signing - Zone Signing Key

Zone Signing Key erzeugen

Zone Signing - Zone Signing Key

Zone Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256   -b 2048    -K /var/named/keys    wsXX.ws.dnssec.bayern
```

Zone Signing - Zone Signing Key

Zone Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -K /var/named/keys wsXX.ws.dnssec.bayern
```

Typ: RSA

Zone Signing - Zone Signing Key

Zone Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -K /var/named/keys wsXX.ws.dnssec.bayern
```

No. Bit

Zone Signing - Zone Signing Key

Zone Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256   -b 2048   -K /var/named/keys   wsXX.ws.dnssec.bayern
```

Verzeichnis für Keys

Zone Signing - Zone Signing Key

Zone Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256   -b 2048    -K /var/named/keys   wsXX.ws.dnssec.bayern
```

Name der Zone

Zone Signing - Zone Signing Key

Zone Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256   -b 2048    -K /var/named/keys    wsXX.ws.dnssec.bayern
```

Zone Signing - Key Signing Key

Zone Signing - Key Signing Key

Key Signing Key erzeugen

Zone Signing - Key Signing Key

Key Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -f KSK   -K /var/named/keys    wsXX.ws.dnssec.bayern
```

Zone Signing - Key Signing Key

Key Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -f KSK   -K /var/named/keys    wsXX.ws.dnssec.bayern
```

Typ: RSA

Zone Signing - Key Signing Key

Key Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -f KSK   -K /var/named/keys    wsXX.ws.dnssec.bayern
```

No. Bit

Zone Signing - Key Signing Key

Key Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -f KSK  -K /var/named/keys    wsXX.ws.dnssec.bayern
```

KSK!

Zone Signing - Key Signing Key

Key Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -f KSK -K /var/named/keys wsXX.ws.dnssec.bayern
```

Verzeichnis für Keys

Zone Signing - Key Signing Key

Key Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -f KSK   -K /var/named/keys   wsXX.ws.dnssec.bayern
```

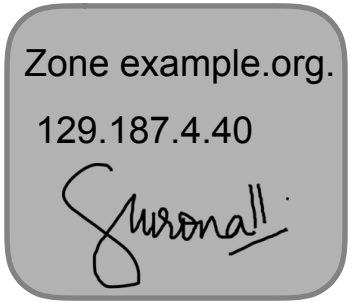
Name der Zone

Zone Signing - Key Signing Key

Key Signing Key erzeugen

```
$ dnssec-keygen -a RSASHA256 -b 2048 -f KSK   -K /var/named/keys    wsXX.ws.dnssec.bayern
```

Zone signieren



Zone signieren, signierte Zone wird als wsXX.ws.dnssec.bayern.signed gespeichert

dnssec-signzone -K /var/named/keys -S -o wsXX.ws.dnssec.bayern \ /etc/bind/wsXX.ws.dnssec.bayern.zone

Wichtige Parameter im Zusammenhang mit dem Schlüssel-Management

-K directory
Key repository: Verzeichnis, in dem nach Schlüsseln gesucht werden soll, wenn keines angegeben wird, wird im lokalen Verzeichnis gesucht

-o origin
The zone origin. Wenn keiner angegeben wird, wird der Dateiname der Zonen-Datei als origin angenommen.

-S
Smart signing: dnssec-signzone sucht selbst nach zur Zone (und Datum) passenden Schlüsseln, und fügt sie in die Zone ein

DS Record - dsset-wsXX.ws.dnssec.bayern.

NB:
Mit -S smart signing werden diese von dnssec-signzone automatisch erzeugt.

Sonst muss man den

DS-Record aus dem öffentlichen Key-Signing-Key erstellen

DS Record - dsset-wsXX.ws.dnssec.bayern.

NB:
Mit -S smart signing werden diese von dnssec-signzone automatisch erzeugt.

Sonst muss man den

DS-Record aus dem öffentlichen Key-Signing-Key erstellen

```
$ dnssec-dsfromkey /var/named/keys/Kdsnsec-wsXX.ws.dnssec.bayern.+008.+40924.key   \
> dsset-example.org
```

DS Record - dsset-wsXX.ws.dnssec.bayern.

NB:
Mit -S smart signing werden diese von dnssec-signzone automatisch erzeugt.

Sonst muss man den

DS-Record aus dem öffentlichen Key-Signing-Key erstellen

```
$ dnssec-dsfromkey /var/named/keys/Kdsnsec-wsXX.ws.dnssec.bayern.+008.+40924.key  \
> dsset-example.org
```

KSK Key-Datei

DS Record - dsset-wsXX.ws.dnssec.bayern.

NB:
Mit -S smart signing werden diese von dnssec-signzone automatisch erzeugt.

Sonst muss man den

DS-Record aus dem öffentlichen Key-Signing-Key erstellen

```
$ dnssec-dsfromkey /var/named/keys/Kdsnsec-wsXX.ws.dnssec.bayern.+008.+40924.key   \
> dsset-example.org
```

Ausgabedatei mit DS RRSets

DS Record - dsset-wsXX.ws.dnssec.bayern.

NB:
Mit -S smart signing werden diese von dnssec-signzone automatisch erzeugt.

Sonst muss man den

DS-Record aus dem öffentlichen Key-Signing-Key erstellen

```
$ dnssec-dsfromkey /var/named/keys/Kdsnsec-wsXX.ws.dnssec.bayern.+008.+40924.key   \
> dsset-example.org
```

DS Record im Parent setzen

nsupdate wird verwendet, um den DS RR im Parent zu setzen:

NB: TTL des Parent-Nameservers muss abgewartet werden, bis die DS-Einträge aktualisiert sind.

DS Record im Parent setzen

nsupdate wird verwendet, um den DS RR im Parent zu setzen:

```
$ nsupdate <ENTER>
> server 10.156.8.23
> zone ws.dnssec.bayern
> update add wsXX.ws.dnssec.bayern. 300 IN DS <DS RECORD RSA-1>
> update add wsXX.ws.dnssec.bayern. 300 IN DS <DS RECORD RSA-2>
> send
```

NB: TTL des Parent-Nameservers muss abgewartet werden, bis die DS-Einträge aktualisiert sind.

Zone veröffentlichen - BIND reload/Neustart

dnssec-signzone erzeugt eine neue Zonen-Datei „.signed“ als Ausgabe.

Darum muss **/etc/bind/named.conf** muss auf die **signierte Zone** verweisen!

```
options {
        directory "/var/cache/bind";
        dnssec-validation auto;
        listen-on-v6 { any; };
};

zone wsXX.ws.dnssec.bayern {
        type master;
        file "/etc/bind/wsXX.ws.dnssec.bayern.zone.signed";
};
```

Dann die Zone veröffentlichen, indem die Zonen-Konfiguration neu geladen wird.

rndc reload

server reload successful

Übung - DNSSEC mit Bind 9.9

Virtual Machine - dnssec-wsXX.dnssec.bayern

- Für jeden Teilnehmer existiert eine Virtuelle Maschine **dnssec-wsXX.dnssec.bayern** (**XX** die Teilnehmernummer **01-25**)

- Login auf jeder VM mit
  Benutzer: ***dnssecadmin***
  Passwort: ***DNSSEC@bayern***

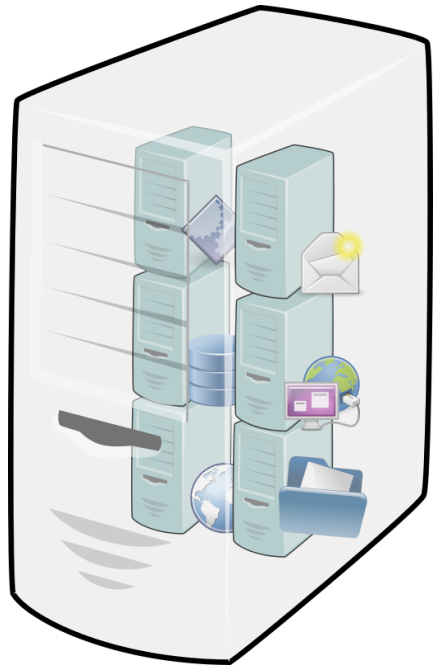
  13:42 [root@dnssec-ws01 ~]# ***sudo -s***

- BIND 9.9.5 installiert, einfaches Zonen-Beispiel in

  /etc/bind/wsXX.ws.dnssec.bayern.zone

- VMs sind aus dem WLAN/Eduroam erreichbar

  ssh -Y dnssecadmin@dnssec-wsXX.dnssec.bayern

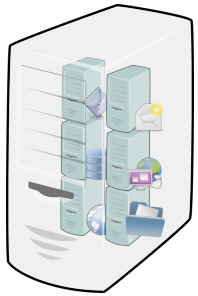


Zone Datei auf dnssec-wsXX.dnssec.bayern

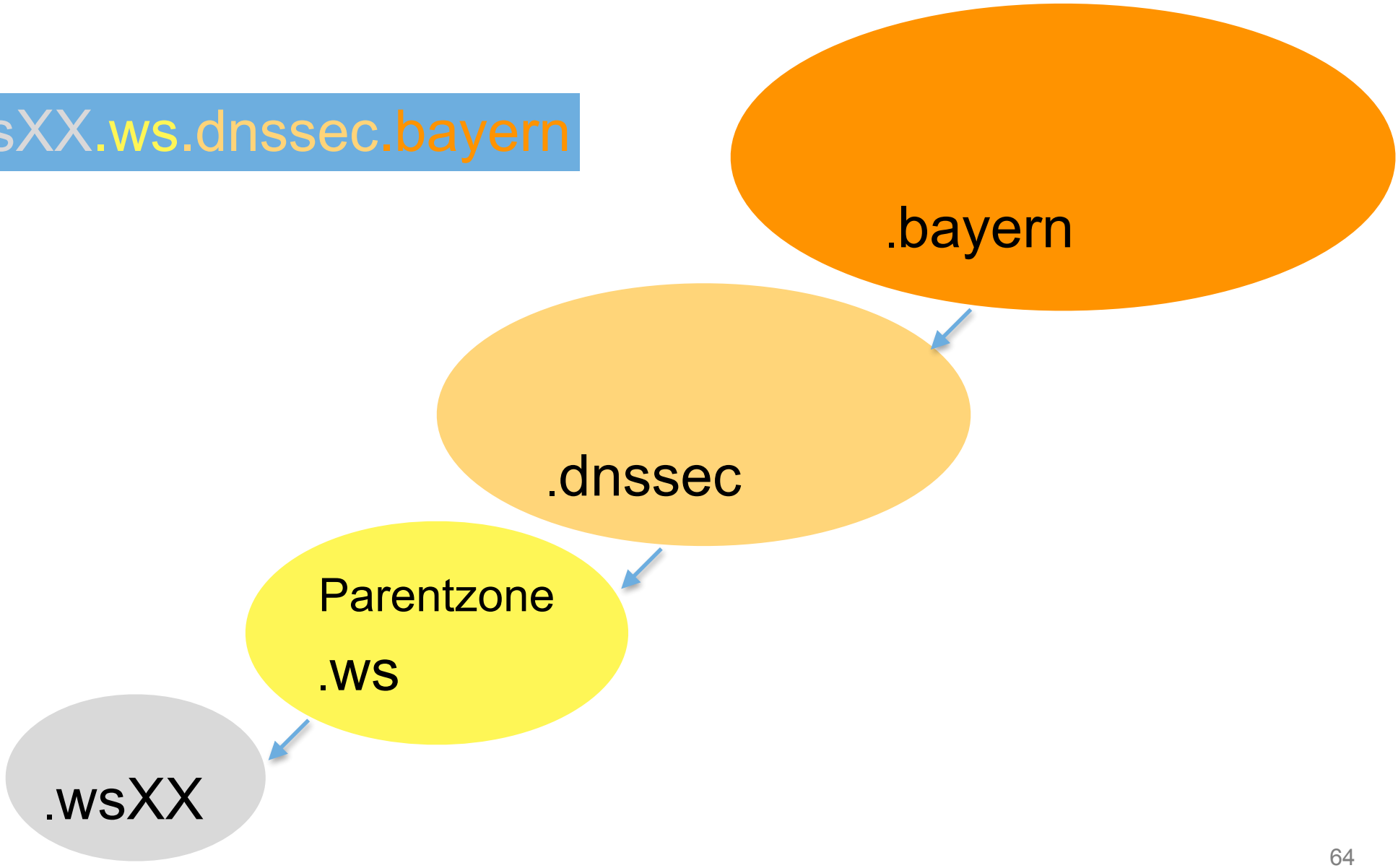
Eine einfache Zone ist in der Datei */etc/bind/ws01.ws.dnssec.bayern.zone* definiert:

```
$TTL 300
$ORIGIN ws01.ws.dnssec.bayern.

@    IN    SOA      dnssec-ws01.dnssec.bayern. root.dnssec-ws01.dnssec.bayern. (
      1     ; serial
      4h    ; refresh
      1h    ; retry
      2w    ; expire
      300 ; negative TTL
      )

      IN    NS  dnssec-ws01.dnssec.bayern.

dnssec-ws01 IN    A     138.246.99.206
dnssec-ws01 IN    AAAA     2001:4ca0:800:2:250:56ff:fe8f:5584
```

Hierarchie der Workshop-Zonen



wsXX.ws.dnssec.bayern

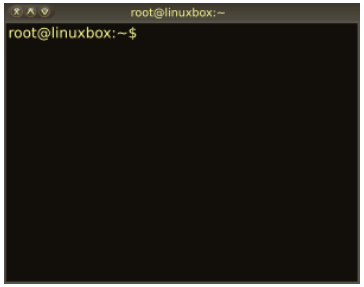


DNSSEC – Schritt für Schritt mit BIND 9.9

Siehe handout „Übung - DNSSEC mit BIND 9.9“

Übung - Debugging mit CLI-Tools

Debugging mit CLI Tools



Siehe Hand-out „Debugging mit CLI Tools“

Übung - Debugging mit Online-Tools

Debugging mit Online Tools

Siehe Hand-out „Debugging mit Online Tools“


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
Voraussetzungen für DNSSEC

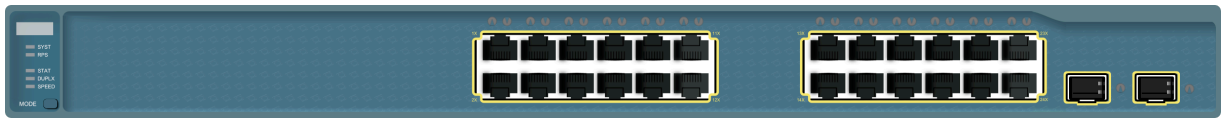
Voraussetzungen für DNSSEC

- Software
- Netzwerk
- Hardware
- Sicherheitsanforderungen

Voraussetzungen für DNSSEC - Software

- BIND Version mindestens 9.7 (ab 9.8 managed key rollover)
- OpenDNSSEC
- Unbound (1.4): resolving Nameserver mit DNSSEC Unterstützung
- Windows Server 2012 (empfohlen) ab Windows Server 2008 R2 limitierter Support

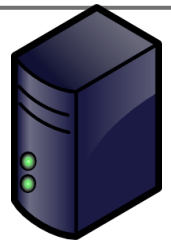
Voraussetzungen für DNSSEC - Netzwerk



- Switche/Router müssen Pakete >512 Byte unterstützen
- MTU-Discovery freigeschaltet
- Firewalls TCP Port 53 offen
- Firewalls DNSSEC-Pakete >512 Byte durchlassen

Voraussetzungen für DNSSEC - Hardware

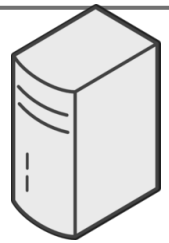
Autoritative Nameserver



- Public key encryption ist rechenintensiv aber mit moderner Serverhardware kein Problem
- Zone-Dateien mindestens 3x so groß (-> RAM-Anforderung größer)

Voraussetzungen für DNSSEC - Hardware

Resolving Nameserver



- Public key encryption ist rechenintensiv aber mit moderner Serverhardware (>2005) kein Problem
- Zone-Dateien mindestens 3x so groß (-> RAM-Anforderung größer)

Voraussetzungen für DNSSEC - Sicherheit

- ZSK/KSK Paar pro Zone ist zu empfehlen
- … oder zumindest für wichtige Zonen
- Private Key Signing Keys müssen an sicherem Ort aufbewahrt werden:

  Hardware Security Module **oder** Datenträger im Tresor

- Konsequentes Key management / rollover (Wer ist verantwortlich?)


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
NSEC/NSEC3 - Nichtexistenz von Objekten

NSEC - Next Secure Record

- DNSSEC sichert die Authentizität von DNS-Einträgen mit Signaturen
- NSEC gibt den nächsten Eintrag einer Zone und welche Typen für einen erfragten Namen existieren
- Authentische Nicht-Existenz von Einträgen
- Dynamisches Erstellen von „xyz.lrz.de existiert nicht“ ist zu rechenaufwändig und verstößt gegen Sicherheitsrichtlinien
- NSEC stellt sicher, dass sich zwischen zwei Einträgen kein weiterer befindet

NSEC - Beispiel 1

Zone-Datei

ant.lrz.de	NSEC	baby.lrz.de	A	AAAA	NSEC	RRSIG
baby.lrz.de	NSEC	cat.lrz.de	A	AAAA	NSEC	RRSIG
cat.lrz.de	NSEC	dodo.lrz.de	A	AAAA	NSEC	RRSIG
dodo.lrz.de	NSEC	mouse.lrz.de	A	AAAA	NSEC	RRSIG
mouse.lrz.de	NSEC	parrot.lrz.de	A	AAAA	NSEC	RRSIG
parrot.lrz.de	NSEC	www.lrz.de	A	AAAA	NSEC	RRSIG
www.lrz.de	NSEC	ant.lrz.de	A	AAAA	NSEC	RRSIG

A für fruit.lrz.de?

Existiert nicht! Kein Eintrag zwischen dodo und mouse

dodo.lrz.de	NSEC	mouse.lrz.de	A	AAAA	NSEC	RRSIG

Signatur über NSEC Eintrag

NSEC - Beispiel 2

Zone-Datei

ant.lrz.de	NSEC baby.lrz.de	A	AAAA	NSEC	RRSIG
baby.lrz.de	NSEC cat.lrz.de	A		NSEC	RRSIG
cat.lrz.de	NSEC dodo.lrz.de	A	AAAA	NSEC	RRSIG
dodo.lrz.de	NSEC mouse.lrz.de	A	AAAA	NSEC	RRSIG
mouse.lrz.de	NSEC parrot.lrz.de	A	AAAA	NSEC	RRSIG
parrot.lrz.de	NSEC www.lrz.de	A	AAAA	NSEC	RRSIG
www.lrz.de	NSEC ant.lrz.de	A	AAAA	NSEC	RRSIG

AAAA für baby.lrz.de?

Existiert nicht! Es ist nicht in der Liste im NSEC Record

baby.lrz.de NSEC cat.lrz.de A NSEC RRSIG

Signatur über NSEC Eintrag

NSEC Record - Detail und Zone Walking

- Zeigt auf den nächsten Eintrag in der Zone
- Zeigt auch alle existierenden RRs für einen Domain-Besitzer

```
www.lrz.de. 3600 IN    NSEC ant.lrz.de.    A        RRSIG NSEC
```

NSEC Record - Detail und Zone Walking

- Zeigt auf den nächsten Eintrag in der Zone
- Zeigt auch alle existierenden RRs für einen Domain-Besitzer

Besitzer

```
www.lrz.de.  3600 IN  NSEC ant.lrz.de.  A  RRSIG NSEC
```

NSEC Record - Detail und Zone Walking

- Zeigt auf den nächsten Eintrag in der Zone
- Zeigt auch alle existierenden RRs für einen Domain-Besitzer

NSEC Record - Detail und Zone Walking

- Zeigt auf den nächsten Eintrag in der Zone
- Zeigt auch alle existierenden RRs für einen Domain-Besitzer

Existierende Resource Record Typen für www.lrz.de

```
www.lrz.de. 3600 IN    NSEC ant.lrz.de.    A        RRSIG NSEC
```

NSEC Record - Detail und Zone Walking

- Zeigt auf den nächsten Eintrag in der Zone
- Zeigt auch alle existierenden RRs für einen Domain-Besitzer

```
www.lrz.de. 3600 IN    NSEC ant.lrz.de.    A        RRSIG NSEC
```

NSEC Record - Detail und Zone Walking

- Letzter NSEC Eintrag zeigt wieder auf den ersten (“Ring”)
- Erlaubt immer noch Entdecken der Zone (“Zone walking”)

NSEC - Problem “Zone Walking”

- Erlaubt die Rekonstruktion eines Zone-Files
- Datenschutz-Probleme
- Informationen können für Angriffe verwertet werden

Lösung: NSEC3 Resource Record

- Wie NSEC
- Aber die Namen sind hashed, um Zone-Entdeckung zu verhindern
- Hashed Namen sind geordnet
- Letzter Eintrag verweist auf ersten (“Ring”)

NSEC3 - Beispiel

Zone-Datei

```
df67wer9x1    NSEC3 ed5g8rt69v     A  AAAA    NSEC3 RRSIG
ed5g8rt69v    NSEC3 ftyro47f75     A          NSEC3 RRSIG
ftyro47f75    NSEC3 h3aq475y76q    A  AAAA    NSEC3 RRSIG
h3aq475y76q   NSEC3 iz45wt6P3d     A          NSEC3 RRSIG
iz45wt6P3d    NSEC3  jf8r8yt64j    A  AAAA    NSEC3 RRSIG
jf8r8yt64j    NSEC3 kt8y0gur9a     A  AAAA MX NSEC3 RRSIG
kt8y0gur9a    NSEC3 df67wer9x1     A  AAAA    NSEC3 RRSIG
```

A für fruit.lrz.de?

Existiert nicht! Es gibt keinen Eintrag zwischen h3aq475y76q und iz45wt6P3d!

```
h3aq475y76q NSEC3 iz45wt6P3d    A          NSEC3 RRSIG
```

RRSIG über NSEC

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QlImhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTlvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

TTL

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

Typ

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QlImhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

nächster sicherer Eintrag

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

NSEC Resource Record mit Signatur

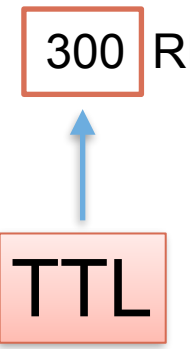
300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

existierende RRs

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY



RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QlImhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTlvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

Typ: Signatur

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

Signatur über NSEC RR

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

Schlüssel-Algo-rithmus: RSA

NSEC Resource Record mit Signatur

```
300   NSEC   dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300  RRSIG NSEC 8 4 300 ( 20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
                          QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
                          oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
                          OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
                          4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
                          Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
                          JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
                          7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
                          jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
                          XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
                          tnEaU6cGK3hBga3P4g== )
```

Anzahl Labels

NSEC Resource Record mit Signatur

```
300   NSEC   dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300  RRSIG NSEC 8 4 300 ( 20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
                          QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
                          oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
                          OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
                          4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
                          Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
                          JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
                          7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
                          jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
                          XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
                          tnEaU6cGK3hBga3P4g== )
```

ursprüngliche TTL

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
DK+1nq2x+M4/Kj36RRLWpX0r
hRuZv7uVF6+Z/6UqEW5rPsyZn
NvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUsIqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

Signatur Ende der Gültigkeit Datum/Zeit

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pnm8AZw/6WeQgi0ROVNYMXLwDzPw4O
6RRLWpX0r
UqEW5rPsyZn
4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTlvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

Signatur Start der Gültigkeit Datum/Zeit

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wQ7oyHjBNAfcfxEg28l
oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeQqi0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK 4/Kj36RRLWpX0r
Dw87wXkW/nimR +Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

Key id

NSEC Resource Record mit Signatur

```
300   NSEC   dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300  RRSIG NSEC 8 4 300 ( 20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
                          QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
                          oY3Zyt4QIlmhOwsCM+3tMfsIwklapoh6y1D8
                          OBQnr0pmm8AZw/6WeQqi0ROVNYMXLwDzPw4O
                          4ZTBUmtlDf0JOK+1nc
                          Dw87wXkW/nimRuZv7
                          JBAEb2AbcoyaNvobB
                          7AUWLGM0EKAEtU0WYmkIA2QC1ivro/Ifr1mY
                          jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
                          XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
                          tnEaU6cGK3hBga3P4g== )
```

Zone, in der die Signatur gilt

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QlImhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4C
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTIvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

Hash der Signatur

NSEC Resource Record mit Signatur

300 NSEC dnssec-ws01.ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY

300 RRSIG NSEC 8 4 300 (20170331153153 20170301153153 56961 ws01.ws.dnssec.bayern.
QDSjpAPvfCrfUJpL/wO7oyHjBNAfcfxEg28l
oY3Zyt4QlImhOwsCM+3tMfsIwklapoh6y1D8
OBQnr0pmm8AZw/6WeOgj0ROVNYMXLwDzPw4O
4ZTBUmtlDf0JOK+1nq2x+M4/Kj36RRLWpX0r
Dw87wXkW/nimRuZv7uVF6+Z/6UqEW5rPsyZn
JBAEb2AbcoyaNvobBId/OnJtJpo4WhPAN2F1
7AUWLGM0EKAEtU0WYmkIA2QCTlvro/lfr1mY
jHVzpTUpSWBA8ZaOxHki3J/fUr6ZCUslqsNU
XcXvdtal/8vnlKKkKozwDHh360XUehkWclVB
tnEaU6cGK3hBga3P4g==)

Übung - NSEC3-Signieren der Zone

Zone-Datei um NSEC3-Einträge erweitern

dnssec-signzone erstellt immer NSEC Einträge.

Um NSEC3 zu verwenden, damit „Zone Walking“ verhindert wird, muss beim signing „-3“ angegeben werden:

dnssec-signzone -3 <hexsalt> -S -o wsXX.ws.dnssec.bayern /etc/bind/wsXX.ws.dnssec.bayern.zone

wobei <hexsalt> ein hexadezimaler Randomseed für die Hashgenerierung ist, z.B: 1A2B3C4D5E6F

Der salt-Wert ist am besten per Skript aus /dev/random zu generieren:

$ ***head -c 512 /dev/random | sha1sum | cut -b 1-16***
e49d307b05a3a172

ldns-nsec3-hash und ldns-walk

DNSSEC erzeugt immer NSEC Einträge.

```
root@dnssec-ws01:/etc/bind# ldns-walk ws01.ws.dnssec.bayern.
ws01.ws.dnssec.bayern.    ws01.ws.dnssec.bayern. NS SOA MX RRSIG NSEC DNSKEY
TYPE65534
dnssec-ws01.ws01.ws.dnssec.bayern. A AAAA RRSIG NSEC
test.ws01.ws.dnssec.bayern. A RRSIG NSEC
*.wildcard.ws01.ws.dnssec.bayern. A RRSIG NSEC
```

ldns-nsec3-hash kann **zu Demonstrationszwecken** dazu verwendet werden, um einen einzelnen Domainnamen auf der Kommandozeile zu hashen:

ldns-nsec3-hash -t 10 -s 1A2B3C4D5E6F dnssec-ws01
mtifjr5uqkulo46r98dsam8lv20acb6n.

Auslesen der Zone mit einem Zonentransfer

- ldns-walk erkennt von selbst, dass es sich um eine NSEC3-gehashte Zone handelt

  ldns-walk wsXX.ws.dnssec.bayern

  ws01.ws.dnssec.bayern. Zone does not seem to be DNSSEC secured,or it uses NSEC3..

- Ein Zonen-Transfer mit dig ist aber möglich

  dig axfr wsXX.ws.dnssec.bayern @127.0.0.1

  liefert… (siehe folgende Folie)

ldns-nsec3 und dig

dnssec-ws01.dnssec.bayern. root.dnssec-ws01.dnssec.bayern. 1 14400 3600 1209600 300
SOA 8 4 300 20170329101013 20170227101013 56961 ws01.ws.dnssec.bayern. vi+SjFCu87y/vqfq8iw/
aHAVXk2ovhsuNiXmRhO5XhnXmqbVqWzi71pf R0/
qdW2iONePjEXjDGYZr+xFmkvjWfQ2i+DvaaGSSgB9STV3rKv1eKxW 0/qr568blOOhG65fAzJPAQDM/
RjJyL1rgP1AurEMlGgFhMZzEY8Pa3lG b8WbREVAZN1p36f562MSigZ+E6v7l9IGYtzSpOEahDTUNb4Bspga8CMl
or97JqpsGfwb2ZT79xPQA3Y+Qrd4zH3GlqFUIKV1TpmSq/atuGeZXu4b
zPxjP6DxaQcE8Rd8atJddxxX0pnpFc6f1vdSBw9cckVfJS0MTOFz/fiU 5PjCVQ==
dnssec-ws01.dnssec.bayern.
NS 8 4 300 20170329101013 20170227101013 56961 ws01.ws.dnssec.bayern.
Xm5L0HkZY5YgqgJ7uQUymd0hH+9PR+kr0bOmqD2L5fg+m5i/kxIhcG+8
ggnLnQpj5lqI7UXOadhFcvOywG86BZ3jQSzj/vJs5NFFGaHaGJ2Y649z
KkTvTWEoNCi0bzVmgfm4XXV62AOwrxCW+RRAqDvkbZfD9bfNxsxqCmpi
OJwdQPs2esHSB+3YdtY32ILDFZ3VfSEvYAvpUUTRjVvr7rx6HCnFaJIE DHNAIexhvzic1p+u6S1nLOxu8+v/7v/
TKsQTyW4cX56t2ShHhWOc+IYn jfBjwOjT8xb9Igg0MDpeRiQ1iudoGrUc4sSMtHdMmKXiMQTgF1pBaOj/ Jli2Qw==
100 dnssec-ws01.dnssec.bayern.
MX 8 4 300 20170329101013 20170227101013 56961 ws01.ws.dnssec.bayern.
nhV7bjn7tgmp1jAdGvRQrXeFkb432tShNma1pL89DMb4hGZMtp3rIjO0 IJ49/
iZmRzuRLs0IBNIcdeWQfbEHzeRUPUDF4LeVp2Gp8bzgOMcMq639 aCzfK2Wz+MNvBO87+JUxBDfu/
30kjn0DbOLvBh0jvoYKFcODYeiCnpca eJolVaCwIq++jLBnwPK9NZvRRY12PPjNZ4pE786ujgRAIHFnXrYXbXW5
kwQBjsaEo0Cz7y33/glWWQpigdFVAJqcpZayMCPogwMiYZbPB2CAqtt1 /luVau6DLw70/cPrAuW7pD6z/
1CZwVO4tDffS/powoPEuIEmtoiLRm98 ObTgfg==
256 3 8 AwEAAc3IHgEHpu5srb3fG1B3YOwNWtP2Sy0z5F8ArvpzOdx4o+/ef03D
Non3pZt855P47fcYxX3vlrsd1Jl+au1CIGaxwIAspWBoIyGqKofRi01D hJeTWaZbgeipLoJmz/
TjSM8cgJtDmgUOeb9tLv25XNuktrq5q82809QI NlSvFc+dr8tIeoLuwBG3uPd/
wgVzSLo9an6WDeOr1v6NtYKPQzlTY7Hs yu0mitIz6OAn8z5yaB+KAcNkz6p1cFXX7XJlFE0tnfvIIjjAV2Rrp3gC
ylDIc2QLCYPqQJCtpYKQ9VH4CKPIBilopRzv2BpRzgTcwKZud8q7SFuk pslVKcFLrZc=……..


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
Erfahrungen aus der DNSSEC-Praxis

DNSSEC am LRZ

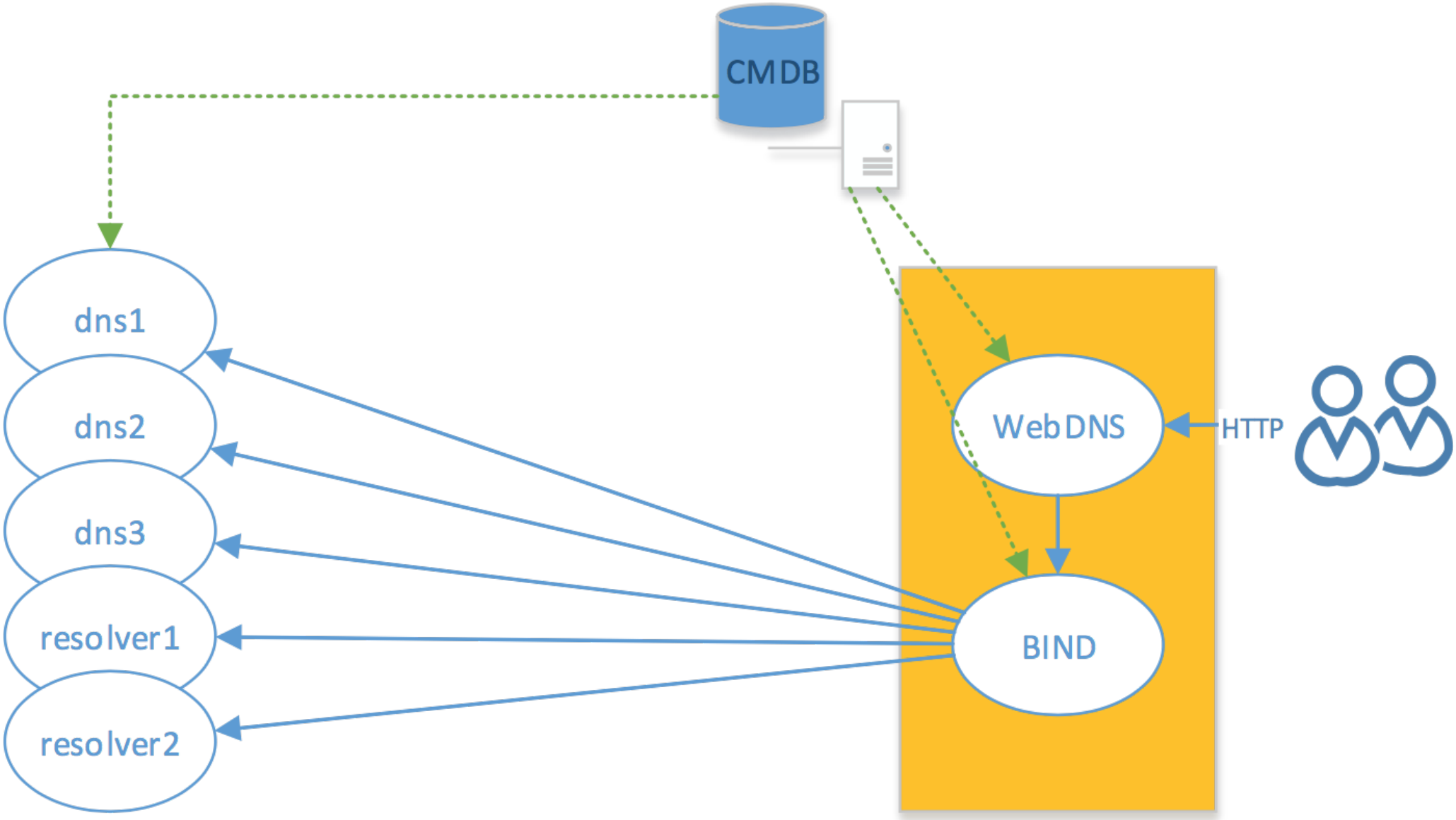
- **resolver1.lrz.de** validierend seit **2008**
- Teilnahme am DENIC-DNSSEC-Testbed, DLV, IANA ITAR
- **resolver2.lrz.de** validierend seit **März 2014**
- seitdem keine Auflösung von Domains mit kaputtem DNSSEC mehr!
- <5 Incidents die periphär durch DNSSEC hervorgerufen wurde

DNSSEC am LRZ

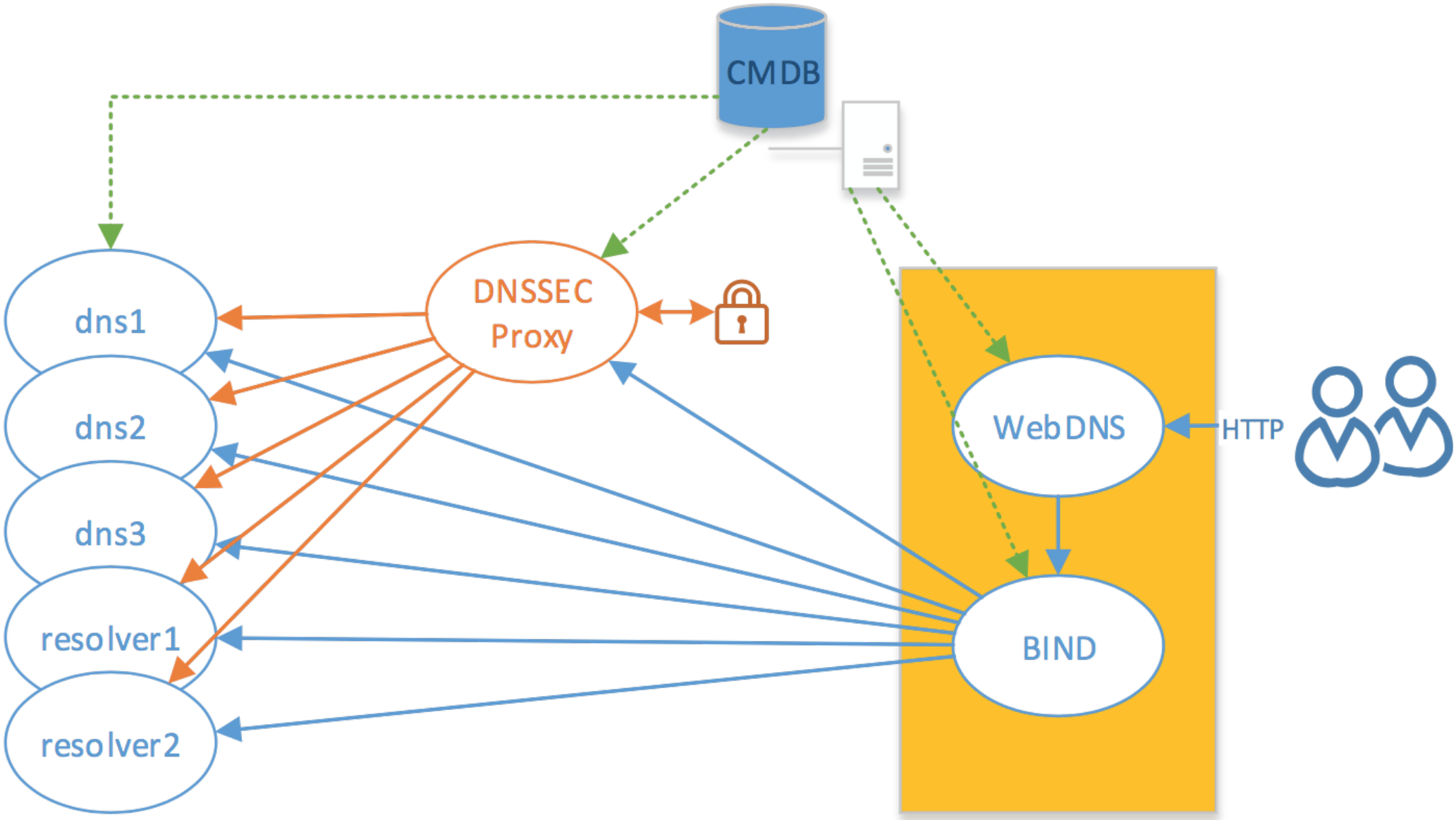
Autoritativer Nameserver

- Implementierung am LRZ als Signing-Proxy
- BIND 9.9 Inline-Signing auf Debian-VM
- alle autoritativen Server (auch Slaves) müssen DNSSEC-fähig sein
- kein Eingriff in bestehende Infrastruktur für !DNSSEC-Zonen

Autoritative Seite am LRZ



Autoritative Seite am LRZ

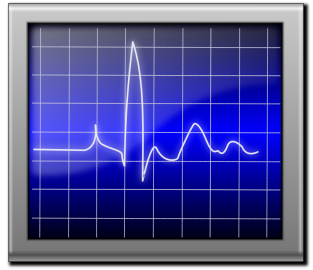


Autoritative Seite am LRZ

- RSASHA256 mit 2048 Bit KSK/ZSK
- DNSSEC-fähige Registrars DFN und InternetX
- Nebenschauplatz: Aufteilung der Nameserver auf drei TLDs

- Große Hauptdomains mit Infrastruktur signiert (wenn's kracht, dann richtig)
  - lrz.de - 28.10.2014
  - tum.de - 10.12.2014
  - lmu.de - 12.01.2015
- einige kleinere Domains ebenfalls, zum Teil schon länger
  - badw-muenchen.de - Dezember 2010

Monitoring am LRZ



- DNSSEC-Signierungsfehler sind tödlich
- fallen je nach lokaler Resolverstruktur gar nicht auf

- Tägliche Prüfung am LRZ für jede DNSSEC-Zone
  - Prüfen der signierten Zone durch ldns-verify-zone (ldns)
    - prüft NSEC-Chaining, Keys, RSSIG-Validity (>30 Tage)
  - Prüfen der signierten Zone durch dnssec-verify (BIND)
    - prüft NSEC-Chaining, Keys
  - Abfrage des SOA-Records bei Google DNS, DNS-OARC
    - prüft sichere Delegation (ad-Flag)
    - prüft Funktionsfähigkeit aus Nutzersicht
- Nutzung lokaler validierender Resolver auf wichtigen Servern


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
Key Rollovers und Schlüsselmanagement

Schlüssel sollten getauscht werden

- Schlüssel veralten bald
  - Neue exploits werden jeden Tag offen gelegt
  - “brute force” wird zunehmend machbar
- Schlüssel können gestohlen oder kompromittiert werden
- Man braucht einen Plan

Key rollover Methoden

- Vor-Veröffentlichung (“pre-publish”)
- Doppelte Singnaturen (“double signature”)
- Für ZSK und KSK
  - Einen KSK zu tauschen bedeutet DS records zu verändern
- “Rollover”-Zeiten hängen von TTL und der Methode ab
- Schlüssel zu (alten) RRSigs müssen vorhanden sein

Pre-publish Methode

- Ein neuer DNSKEY record with mit dem neuen Schlüssel eingeführt
  - allerdings noch nicht zum Signieren verwendet

- Nachdem die TTL abgelaufen ist, werden RRSIGs mit den neuen DNSKEY erzeugt
  - alter DNSKEY wird weiterhin veröffentlicht

- Nachdem die TTL erneut abgelaufen ist, wird der alte DNSKEY entfernt

- DNSKEY / RRSIGs müssen immer in der Zone auffindbar sein (Key_A zu RRSIG_Key_A darf nicht fehlen), TTL beachten

Pre-publish Methode

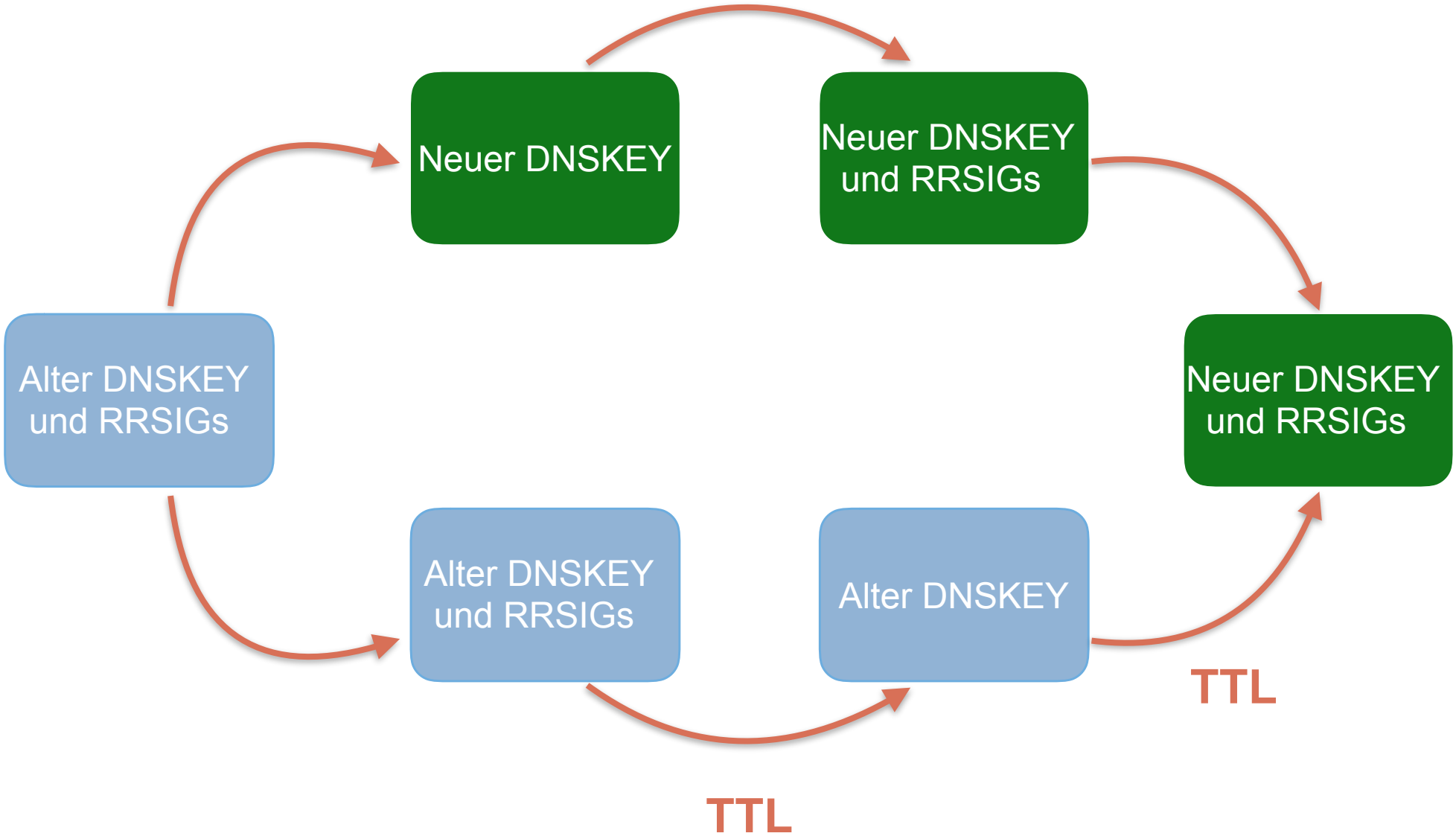
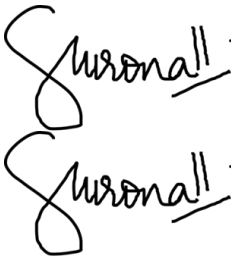


Bild: RIPE NCC

Double Signature Methode



- Ein neuer DNSKEY wird eingeführt und sofort zum Signieren der Records verwendet
- Es gibt zwei RRSIGs für jeden Record, mit Signaturen von beiden DNSKEYs → Zone-Dateien **doppelt** so groß
- Nachdem die TTL abgelaufen ist, wird der alte DNSKEY entfernt, und Records werden wieder nur einmal signiert

Double Signature Methode

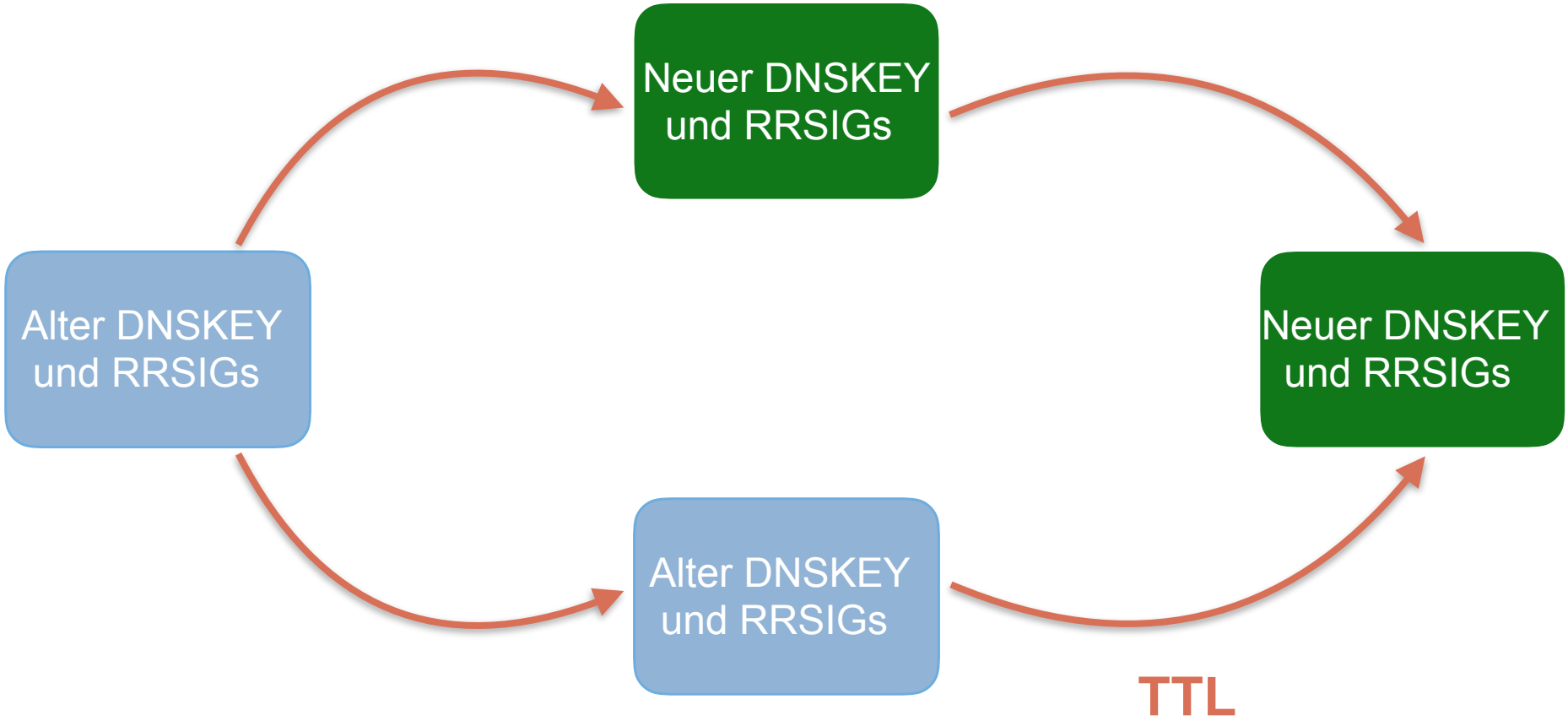
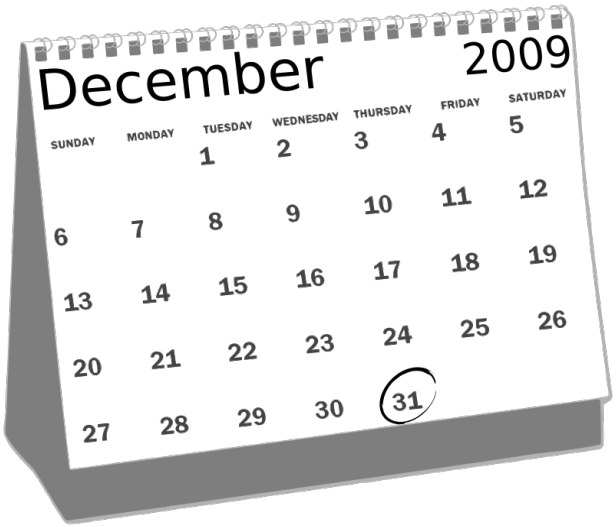


Bild: RIPE NCC

Muss ich an den Rollover denken?

- Nein, er kann vorbereitet werden (z.B. BIND >9.8)
  - in der Konfiguration
  - inklusive des Zeitplans
- Durchführung hängt aber vom DNS-Admin ab
- DNSSEC keys für den nächsten rollover müssen rechtzeitig bereit liegen

Schlüssel in der Praxis



- Ein Schlüssel besitzt 5 wichtige Daten:
  - Veröffentlichung
  - Aktivierung
  - Deaktivierung
  - Zurückziehung
  - Löschung
- korrespondierende **dnssec-keygen** Optionen:
  -P publication date
  -A activation date
  -R revocation date
  -I retIrement date
  -D deletion date
- BIND mit **auto-dnssec** verwendet die Schlüssel im Rahmen dieser konfigurierten Zeiten, erzeugt aber **keine neuen**

Ratschläge zur Schlüsselverwaltung

- Verwende pre-publishing für ZSK

  - insbesondere bei großen Zonen

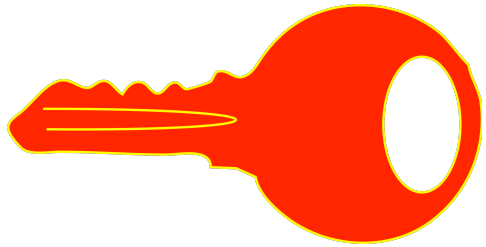
- Verwende double signature für KSK
  - KSK signiert DNSKEY doppelt, nicht die Zone

- Für KSK rollovers, DS records updaten („out-of-band“ Kommunikation)

Ratschläge zur Schlüsselverwaltung

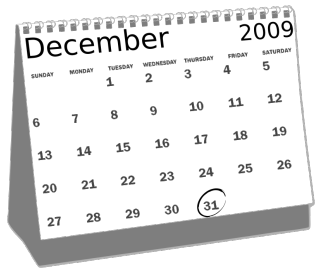
- Schlüssel regelmäßig wechseln!
- Rotation ~ alle 2 Jahre durchführen
- Schlüssel können 2 bis 4 Jahre im Voraus auf Vorrat erzeugt werden
- Das verschiebt aber nur das Problem, an neue Schlüssel muss (irgendwann) gedacht werden

Wechsel des Root-Keys kritisch



- Am **11.Oktober 2017** wird ICANN den DNSSEC .root-Key wechseln
- .root-Key wird bei der Einrichtung des DNSSEC-fähigen Nameserver herunter geladen (BIND >9.7 automatisch)
- **Bis** zum Zeitpunkt des .root-Key-Wechsels muss der neue auf dem Nameserver vorhanden sein
- Sonst schlagen alle DNSSEC-Validierungen fehl und alle DNSSEC-authentifizierten Zonen sind nicht erreichbar

Zeitplan des root KSK Rollovers



- 27. Oktober 2016 : neuer KSK wurde erzeugt
- Februar 2017: Veröffentlichung auf http://data.iana.org/root-anchors/
- 11. Juli 2017: neuer KSK wird im DNS veröffentlicht
- 11. Oktober 2017: neuer KSK wird zum Signieren verwendet
- Januar 2018: Rücknahme des alten KSK
- März 2018: Sichere Vernichtung des alten KSK und Abschluss des Key-rollover Prozesses


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
DNSSEC - Zusammenfassung

DNSSEC - Vorteile für DNS

- Vor Cache impersonation Angriffen geschützt
- DNS Spoofing / Cache-poisoning Angriffe werden verhindert
- Zone-Veränderung auf Slave-Nameservern nicht mehr möglich
- Authentifizierung von allen DNS-Inhalten einer signierten Zone
- zusätzliche RRTYPEs mit Zusatznutzen durch DNSSEC

DNSSEC - Nachteile der Nutzung

- Konfiguration etwas aufwändiger und komplexer
  → Gefahr von Fehlern größer
- Key rollover muß gemanaged werden
- Zonen und DNS-Antworten werden größer, damit werden DDOS-Attacken verstärkt durch die größeren Pakete
- (Aggressive NSEC3 caching bietet (bald) Entlastung von DDOS auf autoritative NS)

dnssec-keymgr (aus BIND 9.11)

Fallstricke des Schlüssel-Managements

- gültige Schlüssel essentiell für DNSSEC
- Schlüsselmanagement ist aufwändig und lästig
- Gefahr von Fehlern bei key rollovers
- BIND <= 9.10 signiert nach Zeitplan,
  erzeugt aber keine neuen Schlüssel
- große Anzahl an Schlüsseln bei vielen Zonen

BIND 9.11 führt dnssec-keymgr ein

- Policy mit Richtlinien zur Schlüsselerzeugung
- Allgemein und pro Zone definierbar
- Algorithmus, Bit-Länge, TTL
- Rollperiod, Prepublish und Postpublish-Zeiten definierbar
- Stand-by keys
- Coverage legt Vorhaltezeitraum fest, für den Schlüssel erzeugt werden

dnssec-keymgr mit BIND 9.9.x

- dnssec-keymgr ist rein in Python implementiert
- Aufruf dient nur der Erzeugung von Schlüsseln, anhand der Meta-Daten vorhandener Schlüssel
- unabhängig von kompilierter BIND-Instanz (funktioniert in 9.9.x)
- regelmäßiger Aufruf mit cron-job erzeugt Schlüssel nur bei Bedarf
- damit immer gültige ZSK ohne Admin-Interaktion
- **nicht für KSK wegen „out-of-band“ Kommunikation**

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

Policies definieren Parametersets, die auf Zonen angewendet werden können

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

z.B. algorithm und key-directory

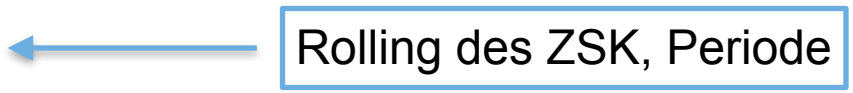
Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

Schlüssel TTL in der Zone und
Schlüssellängen in Bit für KSK und ZSK

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```



Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

Anzahl Standby-keys für KSK und ZSK

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

pre- und postpublish Zeiten für ZSK

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

pre- und postpublish Zeiten für KSK

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

Zeitraum, für den Schlüssel vorgehalten werden

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

wende policy „default“ für diese Zone an

Beispiel Policy-Datei: /etc/dnssec-policy.conf

```
policy default {

        algorithm RSASHA256;
        directory "/var/bind/keys";

        keyttl 10d;
        key-size ksk 2048;
        key-size zsk 2048;

        roll-period zsk 6mo;
        standby ksk 1;
        standby zsk 1;

        pre-publish zsk 20d;
        post-publish zsk 20d;
        pre-publish ksk 60d;
        post-publish ksk 60d;

        coverage 2y;
};

zone ws01.ws.dnssec.bayern {
        policy default;
};
```

dnssec-keymgr-Aufruf

- dnssec-keymgr muss immer noch regelmäßig aufgerufen werden
- überprüft vorhandene Schlüssel und deren Gültigkeit, erzeugt neue, wenn nötig

  dnssec-keymgr [Zonenname]

- Ohne Zonennamen, Schlüssel für alle Zonen in Policy-File erzeugen
- Kann leicht als cron-job laufen

  crontab –e

  0 */1 * * * /usr/local/sbin/dnssec-keymgr > /var/log/keymgr.log

Übung - dnssec-keymgr mit BIND 9.9

dnssec-keymgr mit BIND 9.9

Siehe hand-out
„Übung – dnssec-keymgr mit BIND 9.9 “

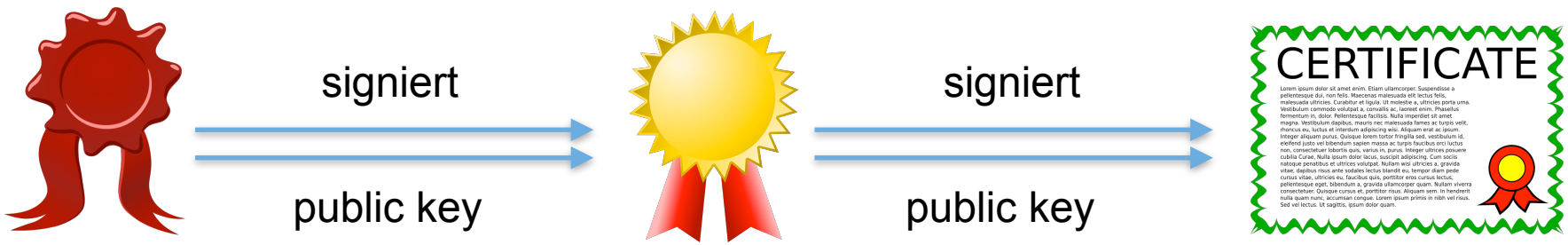

lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
DANE - Grundlagen

DANE - Was ist das?

- “Domain name based authenticated named entity”
- Einem Objekt kann ein Zertifikat authentisch zugeordnet werden
- DNSSEC garantiert Authentizität für einen Eintrag auf DNS
- TLSA = TLS certificate association (RFC6698), neuer RR Eintrag
- Public Key erlaubt Verifizierung dieses Eintrags über den DNS

Warum DANE? - Zertifikat-Probleme

- Basiert auf chain-of-trust
- root-Certificate Authority als root-Anchor
- Kompromittierte CAs z.B. Comodo (OCR-Fehler, 19.Okt. 2016) oder DigiNotar BV, Wosign, Startcom



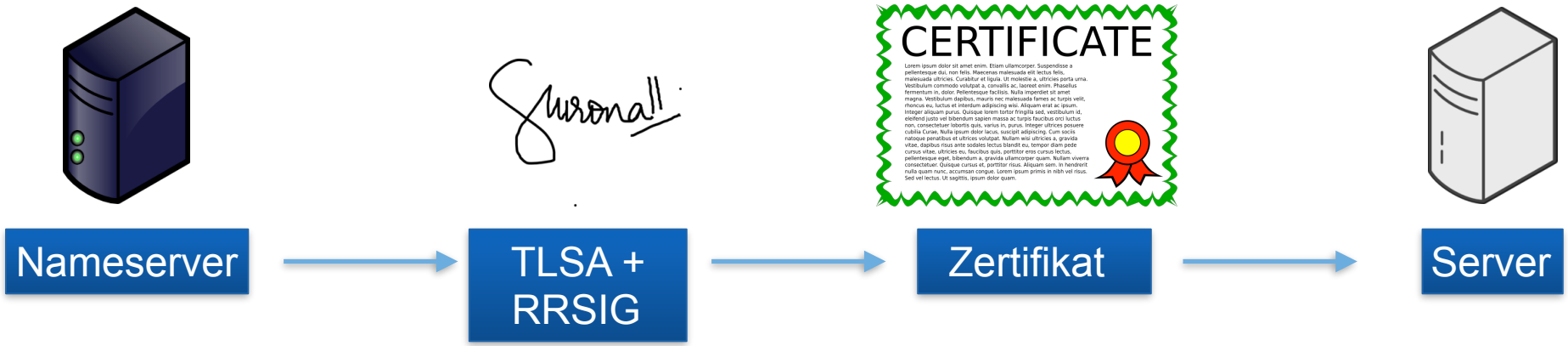
Root CA

CA (DFN)

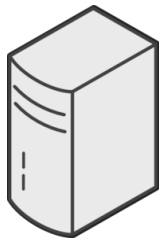
LRZ

DNSSEC - garantiert Authentizität ohne CA

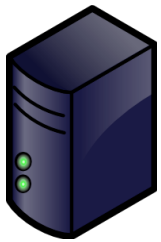
- DNS Name und Dienst wird mit Zertifikat assoziiert (“pinning”)
- Unabhängig von CAs, kann selbst-signiertes Zertifikat sein
- Vertrauenskette



DANE - DNS-based authenticated named Entity

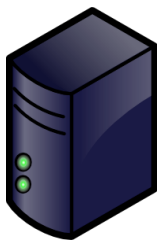


Resolving
Nameserver



Autoritativer
Nameserver

Client

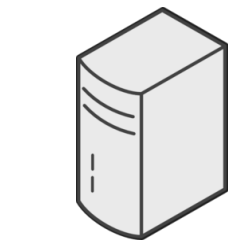


Server

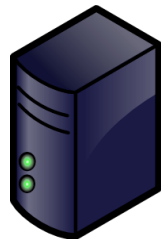
Dienst auf Port N

DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an



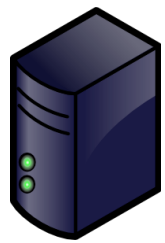
Resolving Nameserver



Autoritativer Nameserver

IP Server?

Client

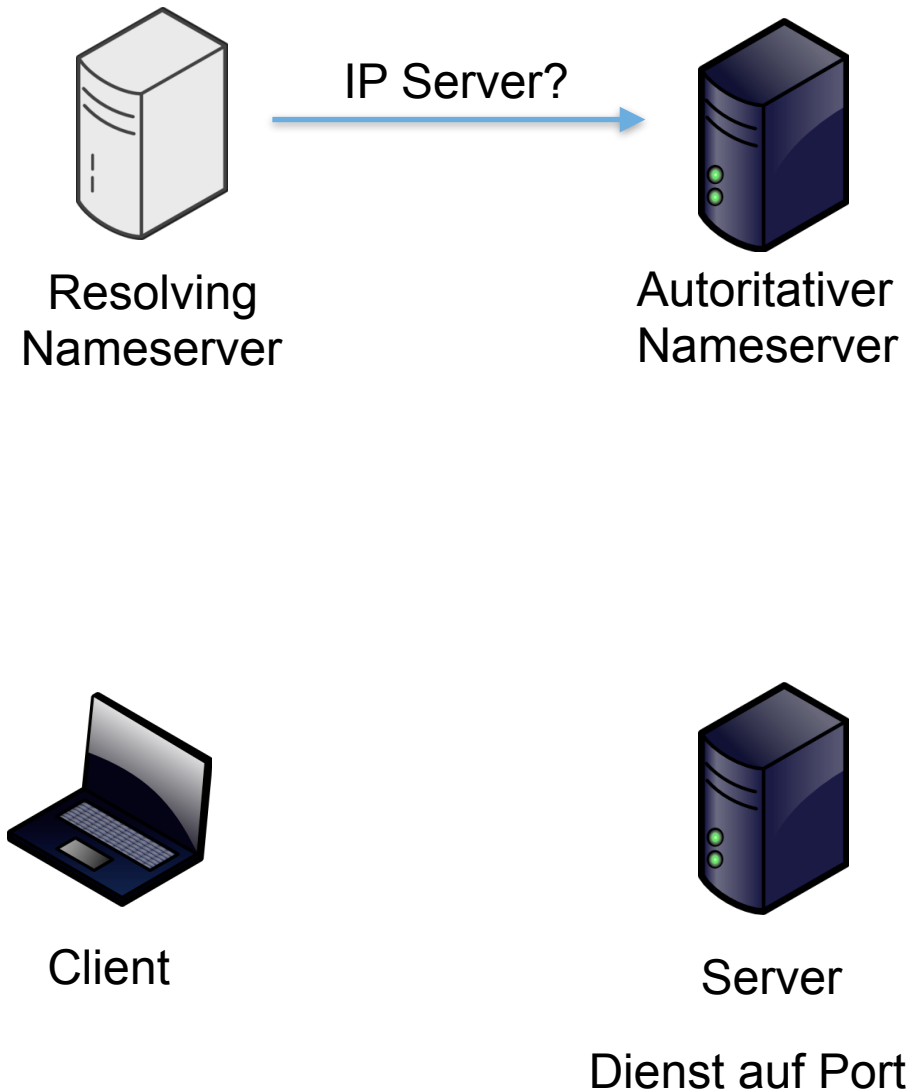


Server

Dienst auf Port N

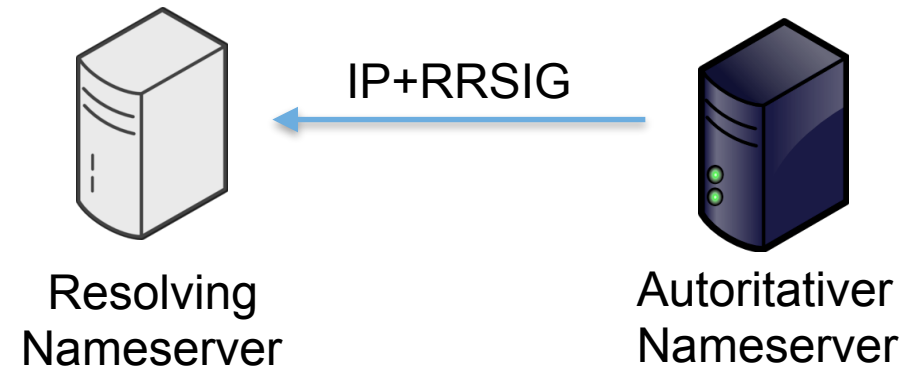
DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N
   auf Server an
2. Resolver fragt Autoritativen Nameserver
   nach IP Server



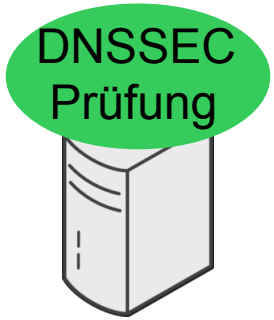
DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG

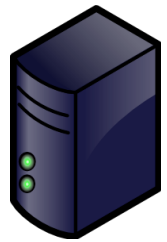


DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP

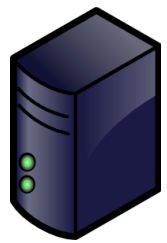


Resolving Nameserver



Autoritativer Nameserver

Client

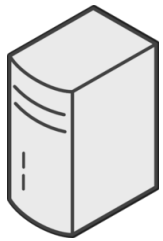


Server

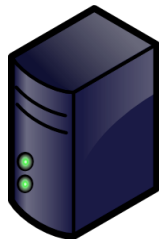
Dienst auf Port N

DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client



Resolving Nameserver

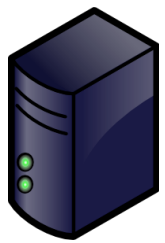


Autoritativer Nameserver

IP+RRSIG

AD Flag

Client

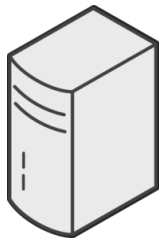


Server

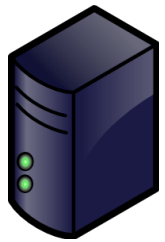
Dienst auf Port N

DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client
6. Client vertraut DNSSEC-AD

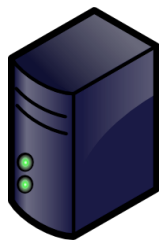


Resolving Nameserver



Autoritativer Nameserver

Client



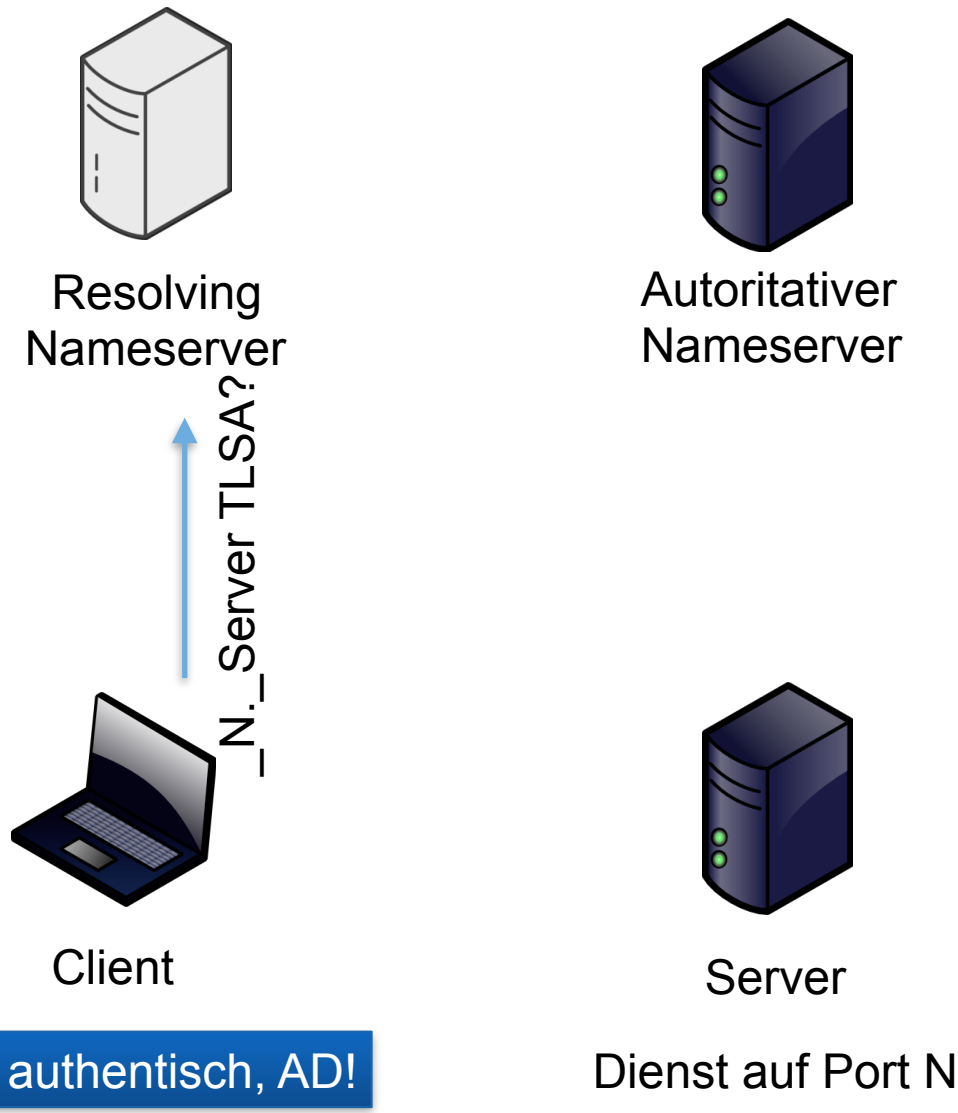
Server

IP authentisch, AD!

Dienst auf Port N

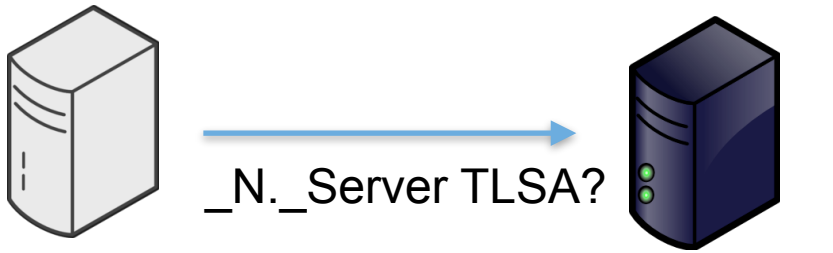
DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client
6. Client vertraut DNSSEC-AD
7. _N._Server TLSA (für Dienst auf Server)?



DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client
6. Client vertraut DNSSEC-AD
7. _N._Server TLSA (für Dienst auf Server)?
8. Resolver fragt Autoritativen NS nach _N._Server TLSA Record

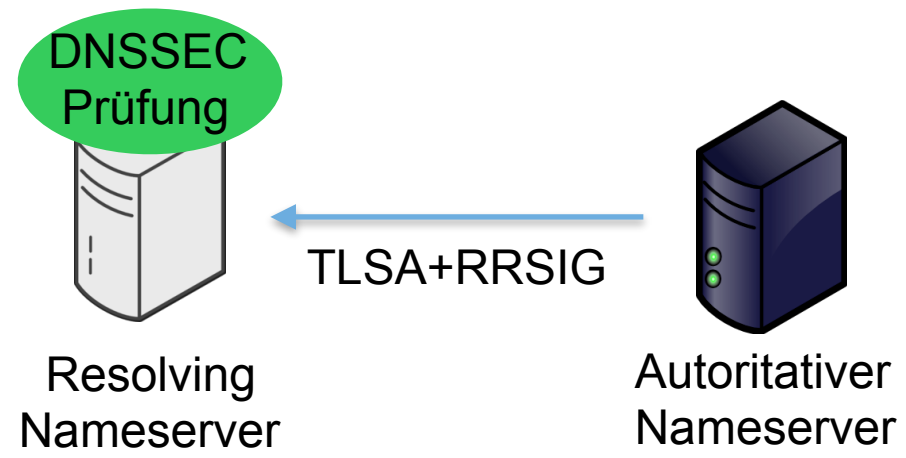


IP authentisch, AD!

Dienst auf Port N

DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client
6. Client vertraut DNSSEC-AD
7. _N._Server TLSA (für Dienst auf Server)?
8. Resolver fragt Autoritativen NS nach _N._Server TLSA Record
9. Aut. NS sendet TSLA+RRSIG-Eintrag für Port N

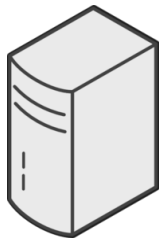


IP authentisch, AD!

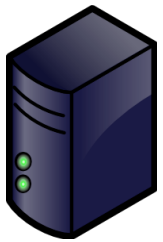
Dienst auf Port N

DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client
6. Client vertraut DNSSEC-AD
7. _N._Server TLSA (für Dienst auf Server)?
8. Resolver fragt Autoritativen NS nach _N._Server TLSA Record
9. Aut. NS sendet TSLA+RRSIG-Eintrag für Port N
10. Resolver sendet TLSA-Signatur an Client



Resolving Nameserver

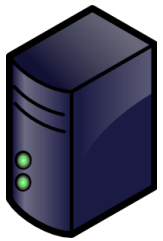


Autoritativer Nameserver

AD Flag

TSLA+RRSIG

Client

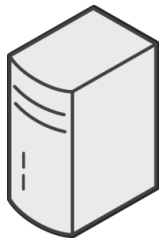


Server

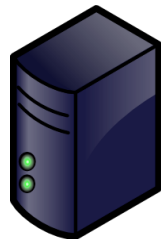
Dienst auf Port N

DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client
6. Client vertraut DNSSEC-AD
7. _N._Server TLSA (für Dienst auf Server)?
8. Resolver fragt Autoritativen NS nach _N._Server TLSA Record
9. Aut. NS sendet TSLA+RRSIG-Eintrag für Port N
10. Resolver sendet TLSA-Signatur an Client
11. Zertifikat empfangen



Resolving Nameserver

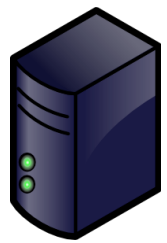


Autoritativer Nameserver

AD Flag

TSLA+RRSIG

Client

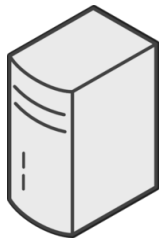


Server

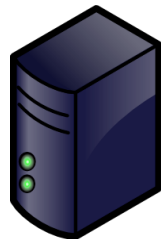
Dienst auf Port N

DANE - DNS-based authenticated named Entity

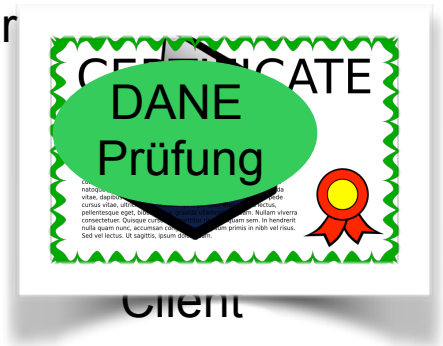
1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client
6. Client vertraut DNSSEC-AD
7. _N._Server TLSA (für Dienst auf Server)?
8. Resolver fragt Autoritativen NS nach _N._Server TLSA Record
9. Aut. NS sendet TSLA+RRSIG-Eintrag für Port N
10. Resolver sendet TLSA-Signatur an Client
11. Zertifikat empfangen
12. Client überprüft TSLA-Signatur anhand Public Key-Hash



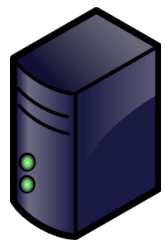
Resolving Nameserver



Autoritativer Nameserver



Client

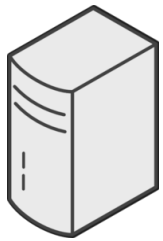


Server

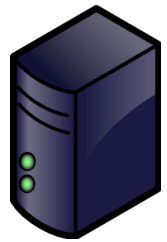
Dienst auf Port N

DANE - DNS-based authenticated named Entity

1. Client fragt Dienst auf Port N auf Server an
2. Resolver fragt Autoritativen Nameserver nach IP Server
3. Aut. NS sendet IP+RRSIG
4. Resolving Nameserver prüft IP mit DNSSEC Hash und RRSIG für IP
5. Resolver sendet DNSSEC-AD Antwort an Client
6. Client vertraut DNSSEC-AD
7. _N._Server TLSA (für Dienst auf Server)?
8. Resolver fragt Autoritativen NS nach _N._Server TLSA Record
9. Aut. NS sendet TSLA+RRSIG-Eintrag für Port N
10. Resolver sendet TLSA-Signatur an Client
11. Zertifikat empfangen
12. Client überprüft TSLA-Signatur anhand Public Key-Hash
13. Zertifikat gepinnt

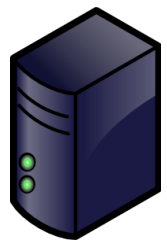


Resolving Nameserver



Autoritativer Nameserver

Client



Server

Dienst auf Port N

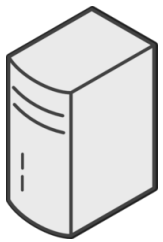
DANE - Beispiel Absicherung eines Emailservers

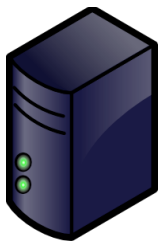
TLS-verschlüsselte SMTP-Verbindung

- STARTTLS-Erweiterung für SMTP wurde erst 2002 spezifiziert,
- 20 Jahre nach RFC821
  - signalisiert durch STARTTLS-Keyword EHLO (unauthentifizierte Plaintext-Session!)
  - Triviale Downgrade-Attacke auf unverschlüsselte Verbindung
  - Opportunistische Verschlüsselung ohne Zertifikatsprüfung (70% gültiges Zertifikat, 20% selbstsigniert, 10% kein TLS)
- Schützt lediglich gegen passive Lauscher, aber nicht gegen aktive Angriffe (MitM)

TLS-verschlüsselte SMTP Verbindung

- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen

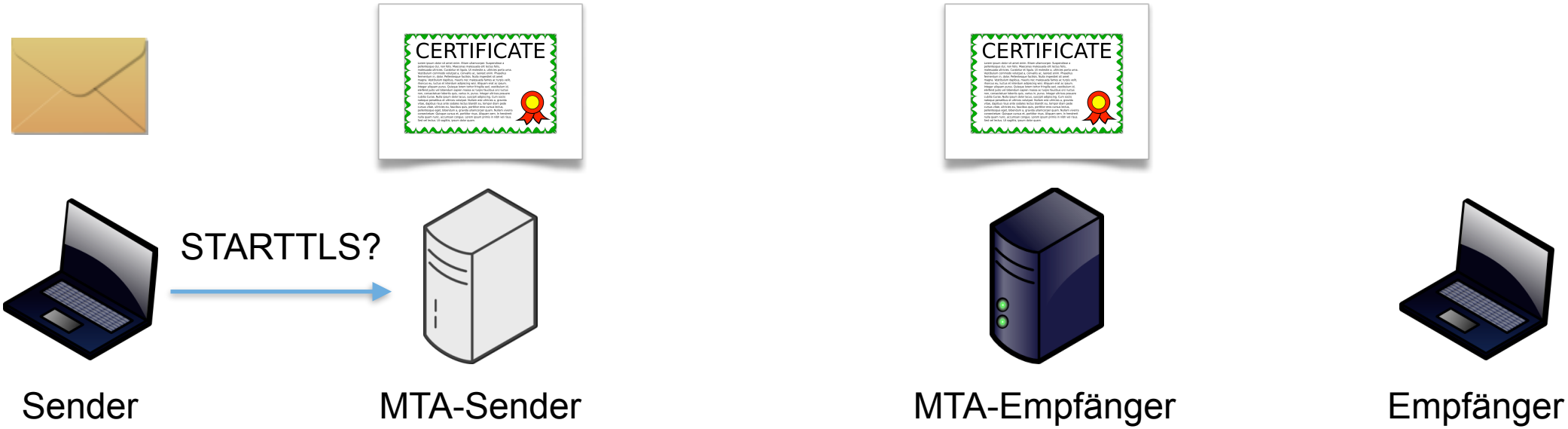




Sender | MTA-Sender | MTA-Empfänger | Empfänger

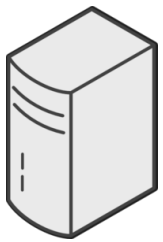
TLS-verschlüsselte SMTP Verbindung

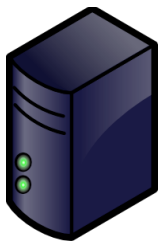
- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen



TLS-verschlüsselte SMTP Verbindung

- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen





Sender

MTA-Sender

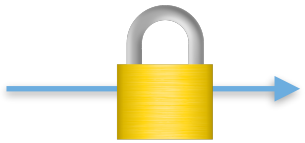
MTA-Empfänger

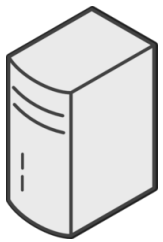
Empfänger

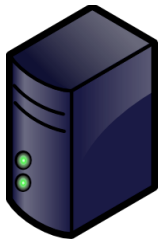
Vertrauen?

TLS-verschlüsselte SMTP Verbindung

- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen







Sender

MTA-Sender

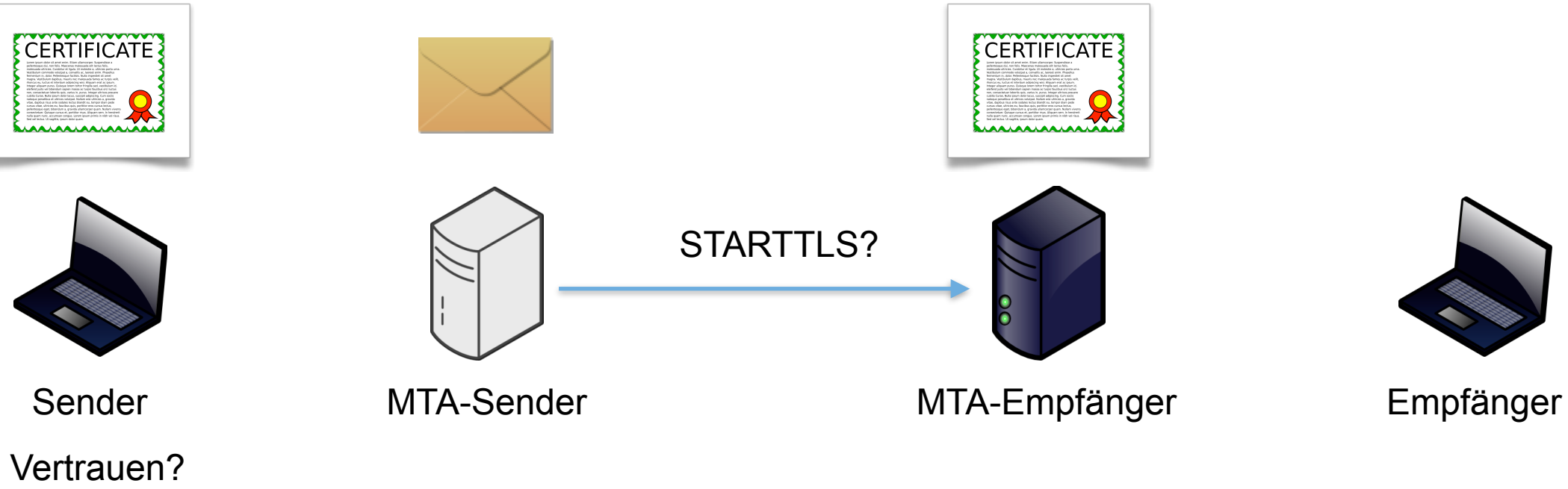
MTA-Empfänger

Empfänger

Vertrauen?

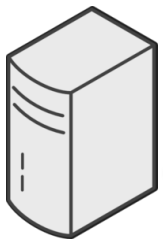
TLS-verschlüsselte SMTP Verbindung

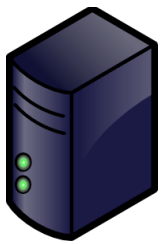
- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen



TLS-verschlüsselte SMTP Verbindung

- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen





TLS-Zertifikat

Sender

MTA-Sender

MTA-Empfänger

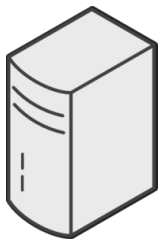
Empfänger

Vertrauen?

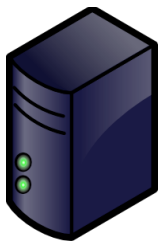
TLS-verschlüsselte SMTP Verbindung

- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen

Sender



MTA-Sender

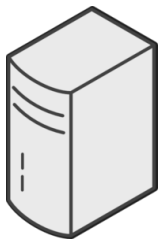


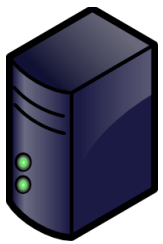
MTA-Empfänger

Empfänger

TLS-verschlüsselte SMTP Verbindung

- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen

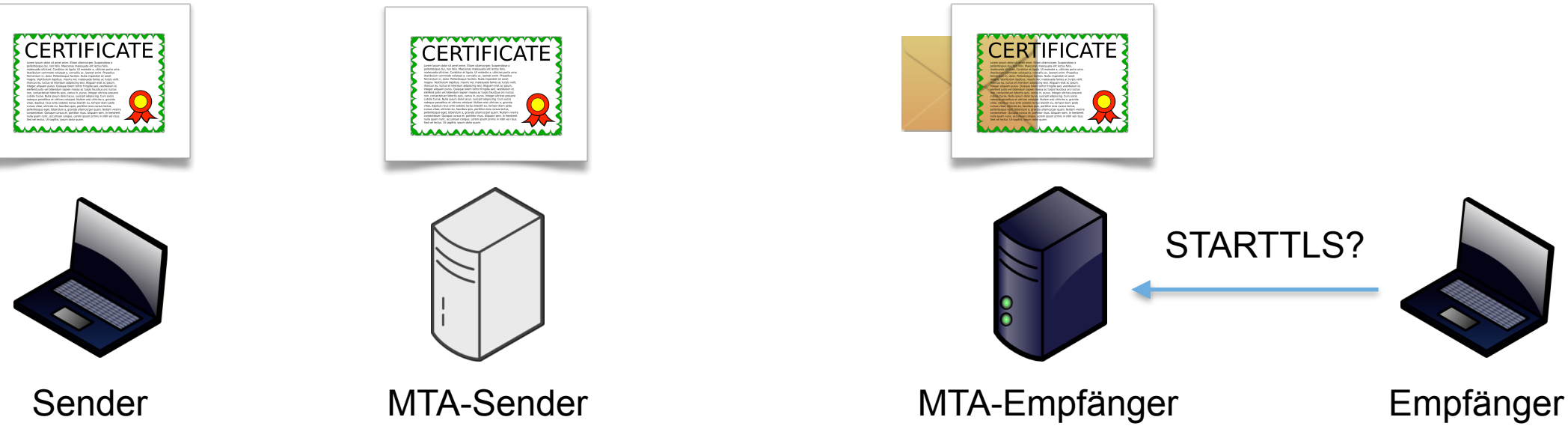




Sender MTA-Sender MTA-Empfänger Empfänger

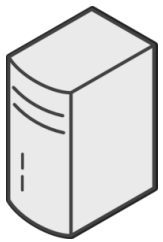
TLS-verschlüsselte SMTP Verbindung

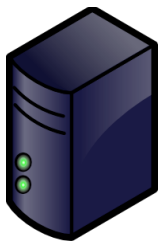
- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen



TLS-verschlüsselte SMTP Verbindung

- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen





Sender

MTA-Sender

MTA-Empfänger

Empfänger

TLS-verschlüsselte SMTP Verbindung

- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen

TLS-verschlüsselte SMTP Verbindung

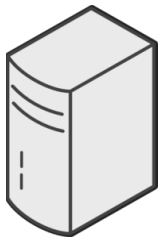
- Mailtransport via SMTP, TLS-verschlüsselt, Zertifikat-Vertrauen

TLS-Verschlüsselte SMTP-Verbindung mit DANE

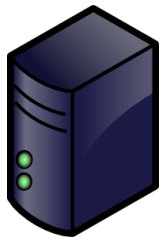
- DANE für SMTP (RFC 7672) erlaubt dem Empfänger über DNSSEC die **sichere** Signalisierung von
  - „Ich spreche STARTTLS“
  - „Ich habe ein Zertifikat mit bestimmten Eigenschaften“
- DANE-fähiger Sender kann diese Hinweise beachten und **SMTP-Session bei einem Fehler terminieren**

TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



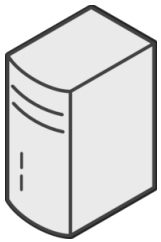
DNS Resolver



autoritativer Nameserver

Sender

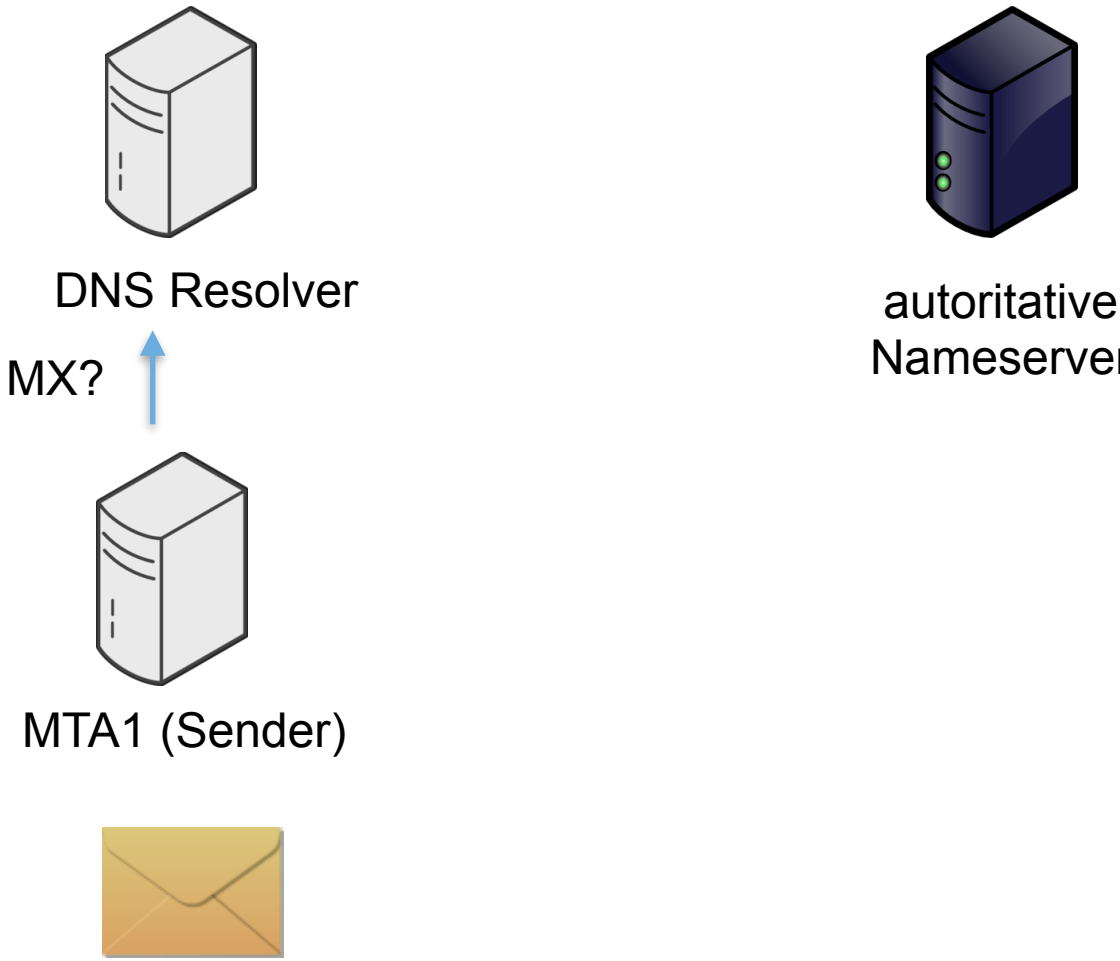
Empfänger



MTA1 (Sender)

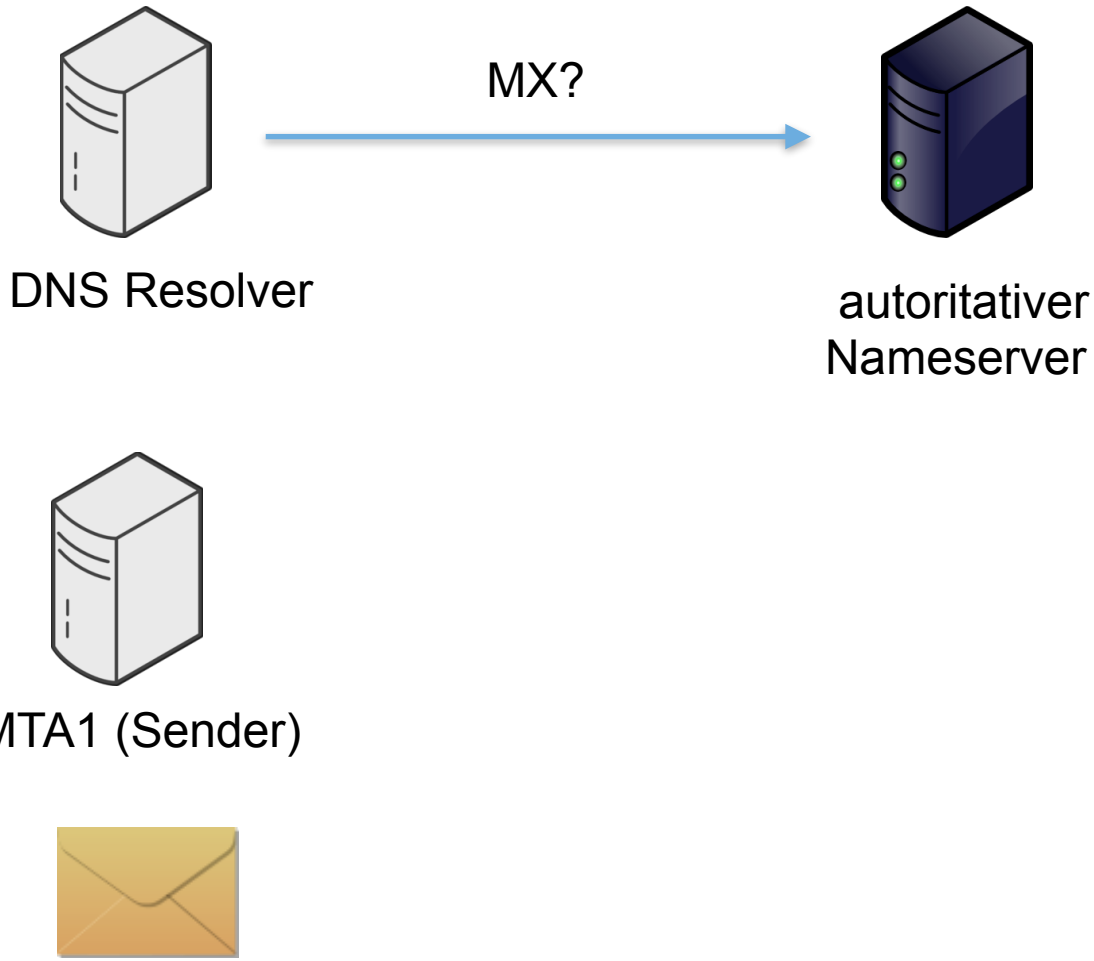
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



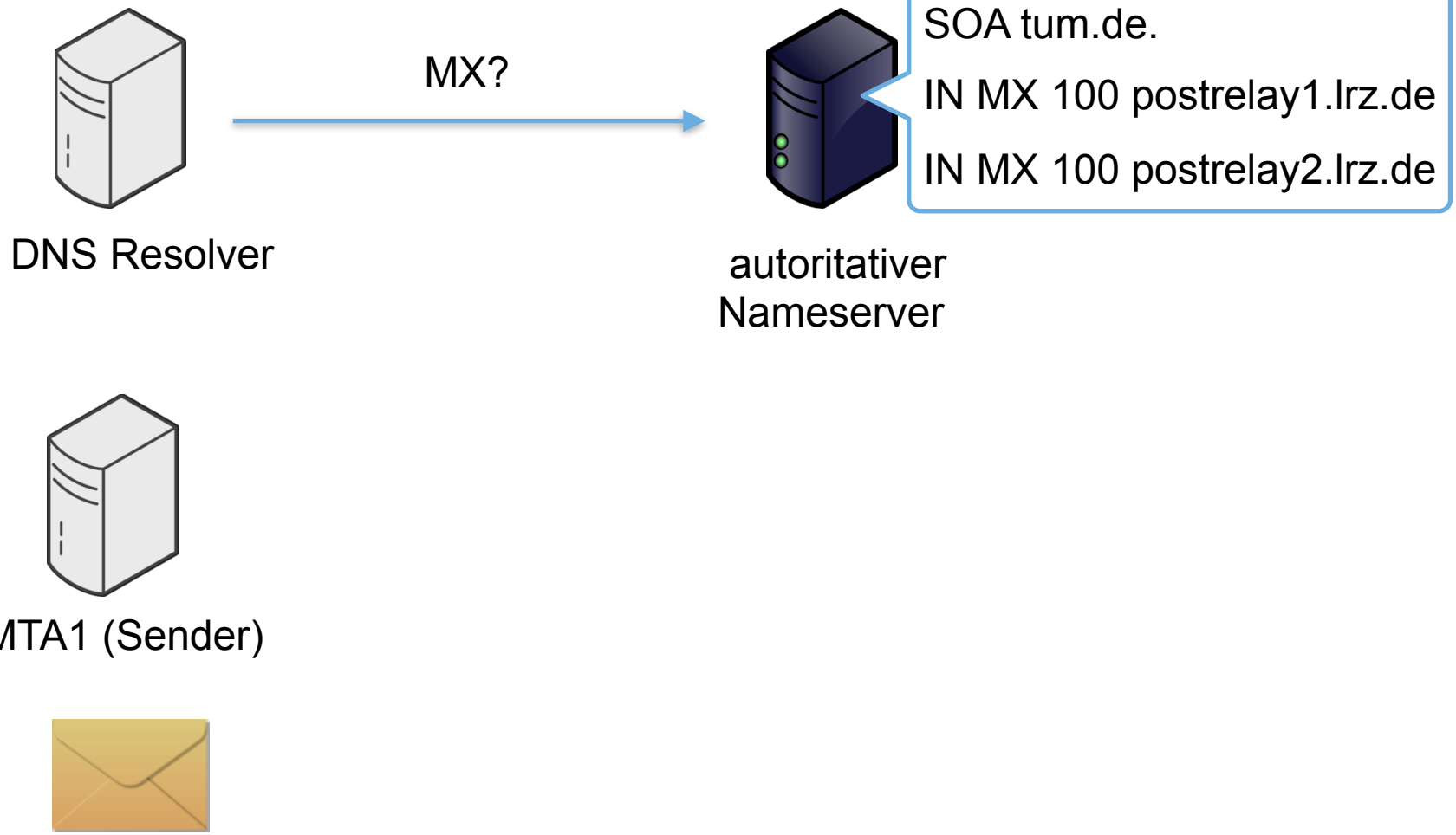
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



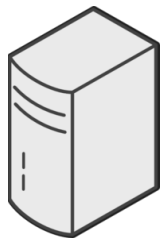
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



DNS Resolver

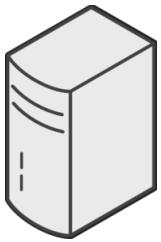
MX | RRSIG

SOA tum.de.

IN MX 100 postrelay1.lrz.de

IN MX 100 postrelay2.lrz.de

autoritativer Nameserver

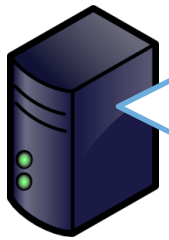


MTA1 (Sender)

TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

DNS Resolver

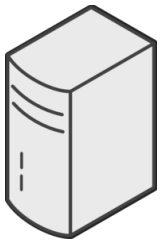


SOA tum.de.

IN MX 100 postrelay1.lrz.de

IN MX 100 postrelay2.lrz.de

autoritativer
Nameserver



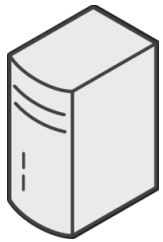
MTA1 (Sender)

TLS-SMTP Verbindung mit DANE

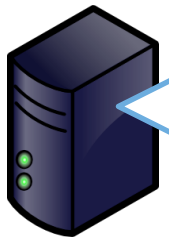
- Mailtransport via SMTP, TLS-verschlüsselt und DANE

DNS Resolver

MX | **AD** FLAG



MTA1 (Sender)



SOA tum.de.

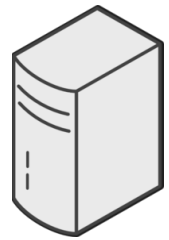
IN MX 100 postrelay1.lrz.de

IN MX 100 postrelay2.lrz.de

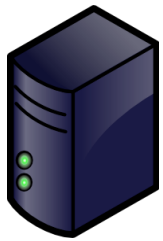
autoritativer
Nameserver

TLS-SMTP Verbindung mit DANE

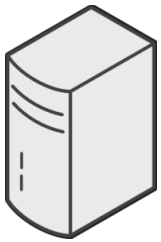
- Mailtransport via SMTP, TLS-verschlüsselt und DANE



DNS Resolver



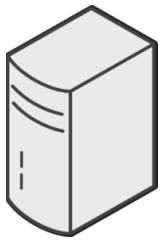
LRZ autoritativer Nameserver



MTA1 (Sender)

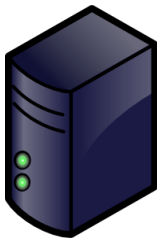
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

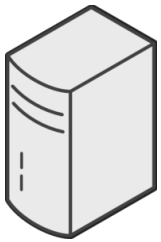


DNS Resolver

IP postrelay1.lrz.de?



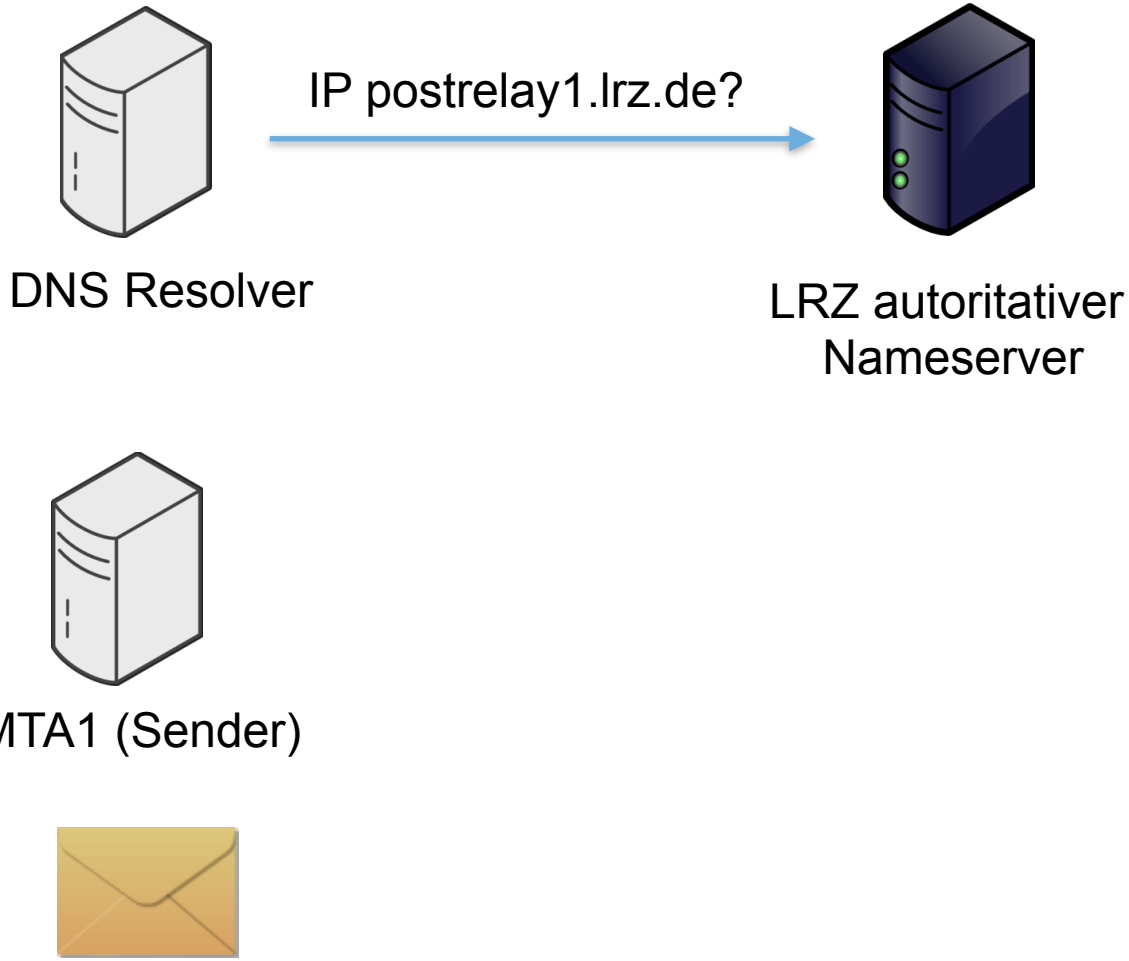
LRZ autoritativer Nameserver



MTA1 (Sender)

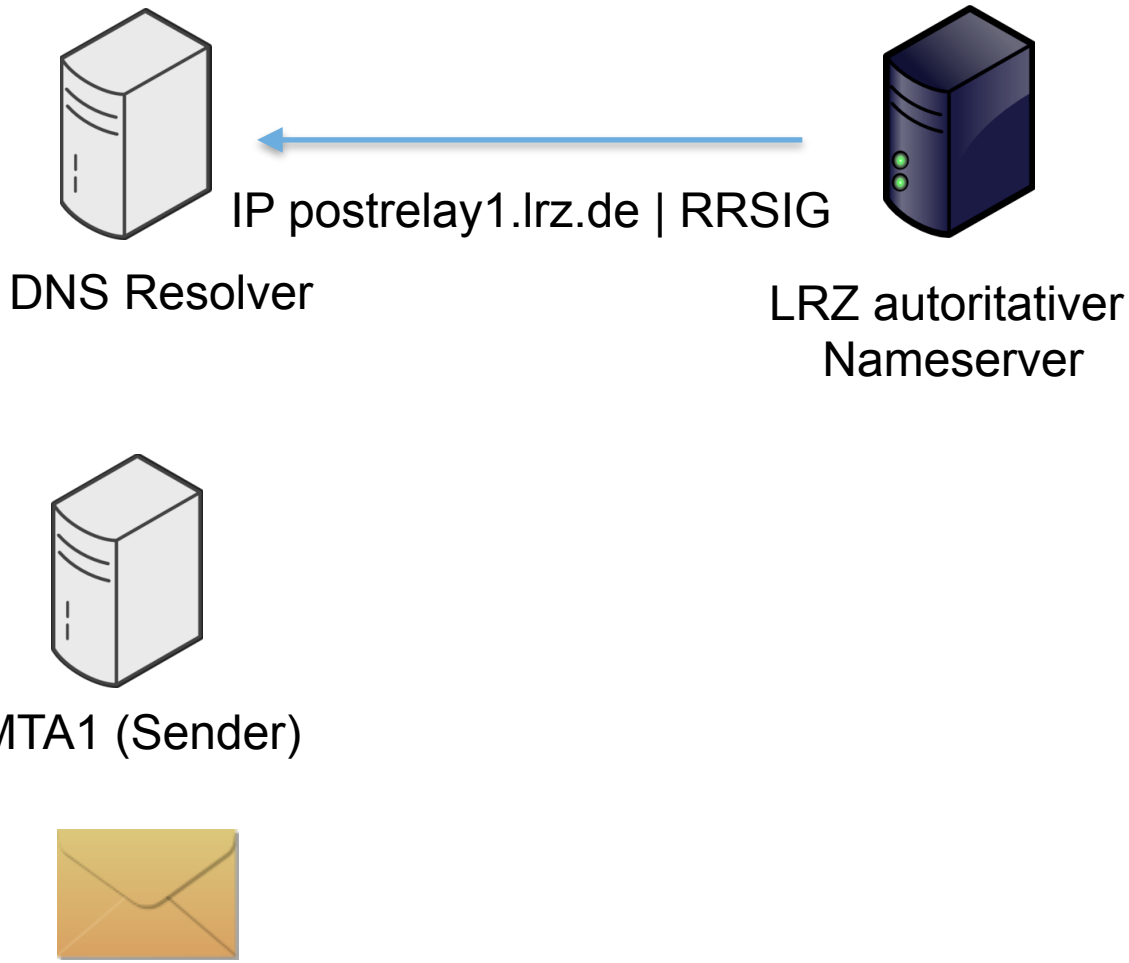
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



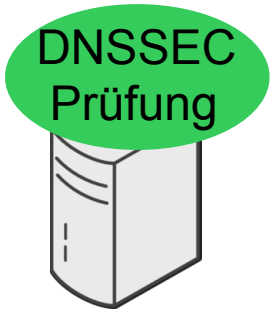
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

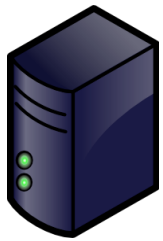


TLS-SMTP Verbindung mit DANE

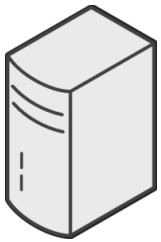
- Mailtransport via SMTP, TLS-verschlüsselt und DANE



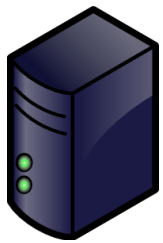
DNS Resolver



LRZ autoritativer Nameserver



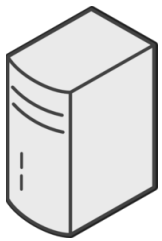
MTA1 (Sender)



postrelay1.lrz.de

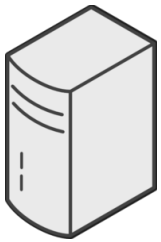
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

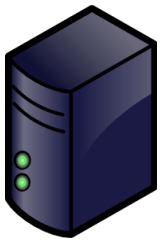


DNS Resolver

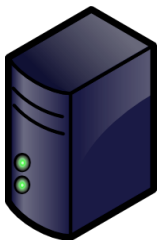
IP | **AD** FLAG



MTA1 (Sender)



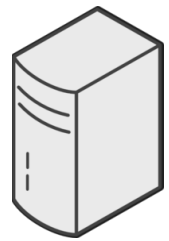
LRZ autoritativer Nameserver



postrelay1.lrz.de

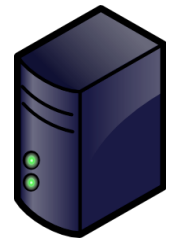
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

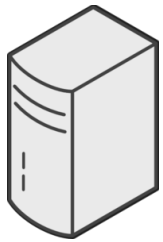


DNS Resolver

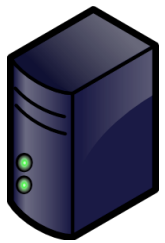
_25._.postrelay1.lrz.de?



LRZ autoritativer Nameserver



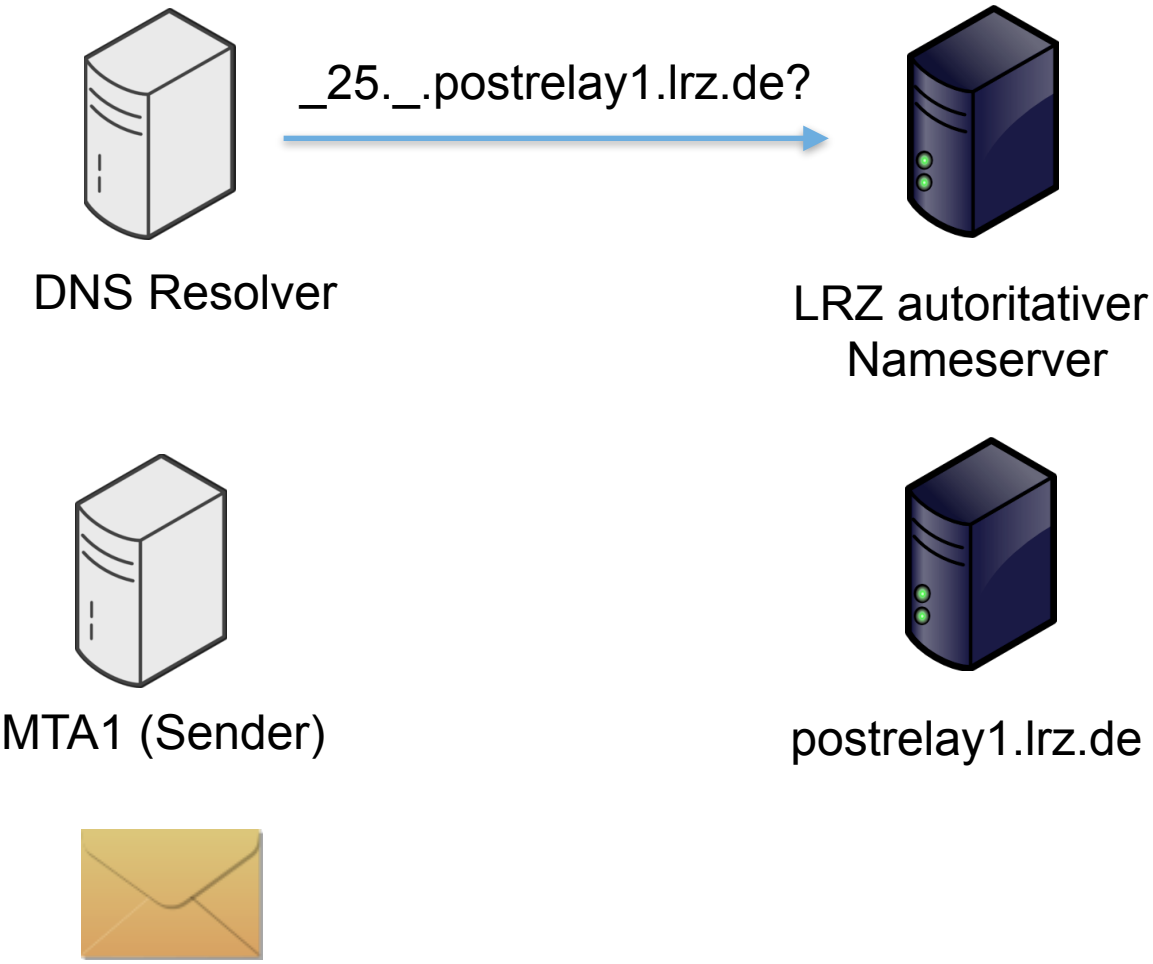
MTA1 (Sender)



postrelay1.lrz.de

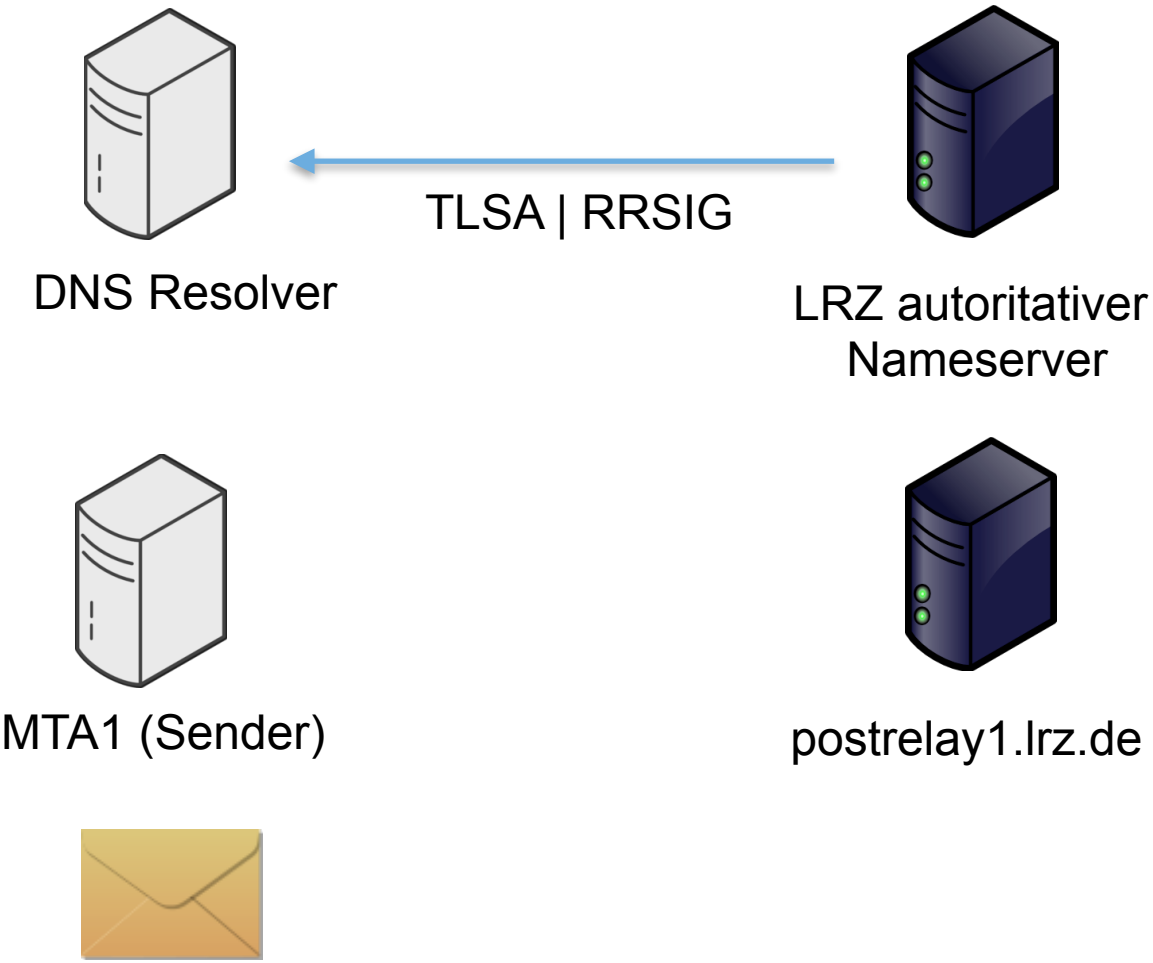
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



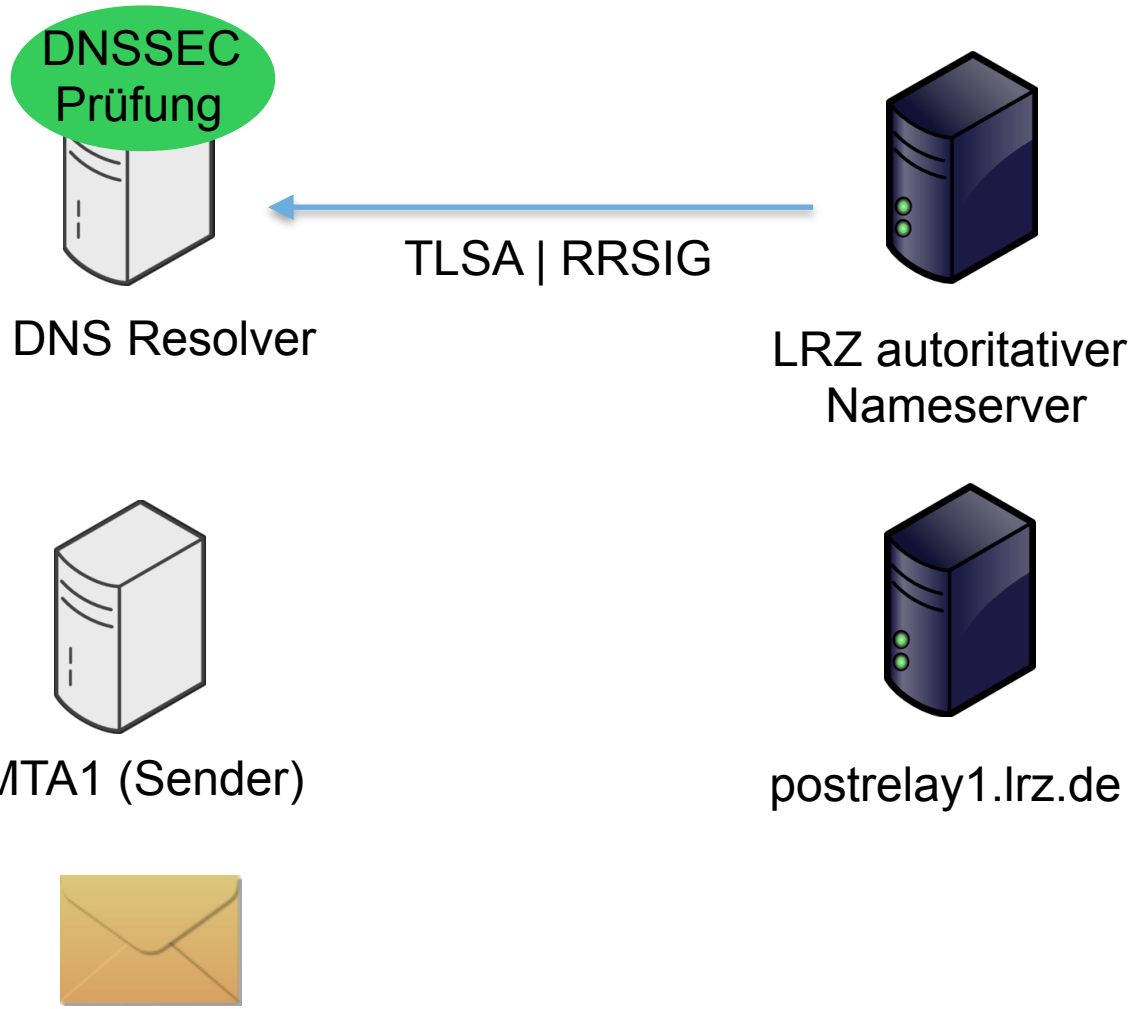
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



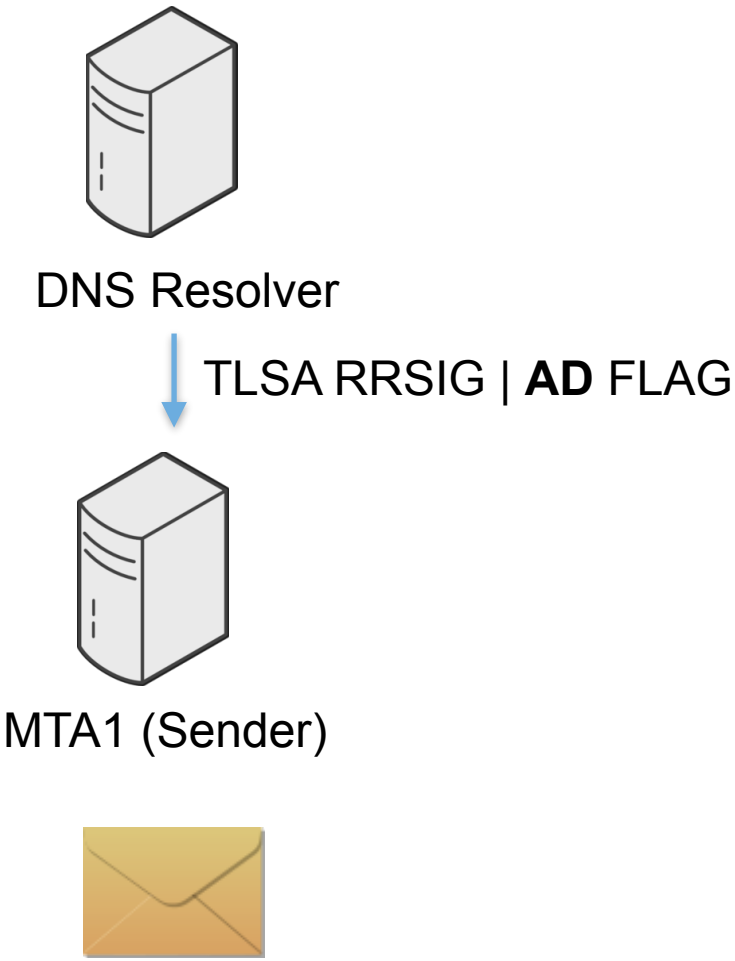
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



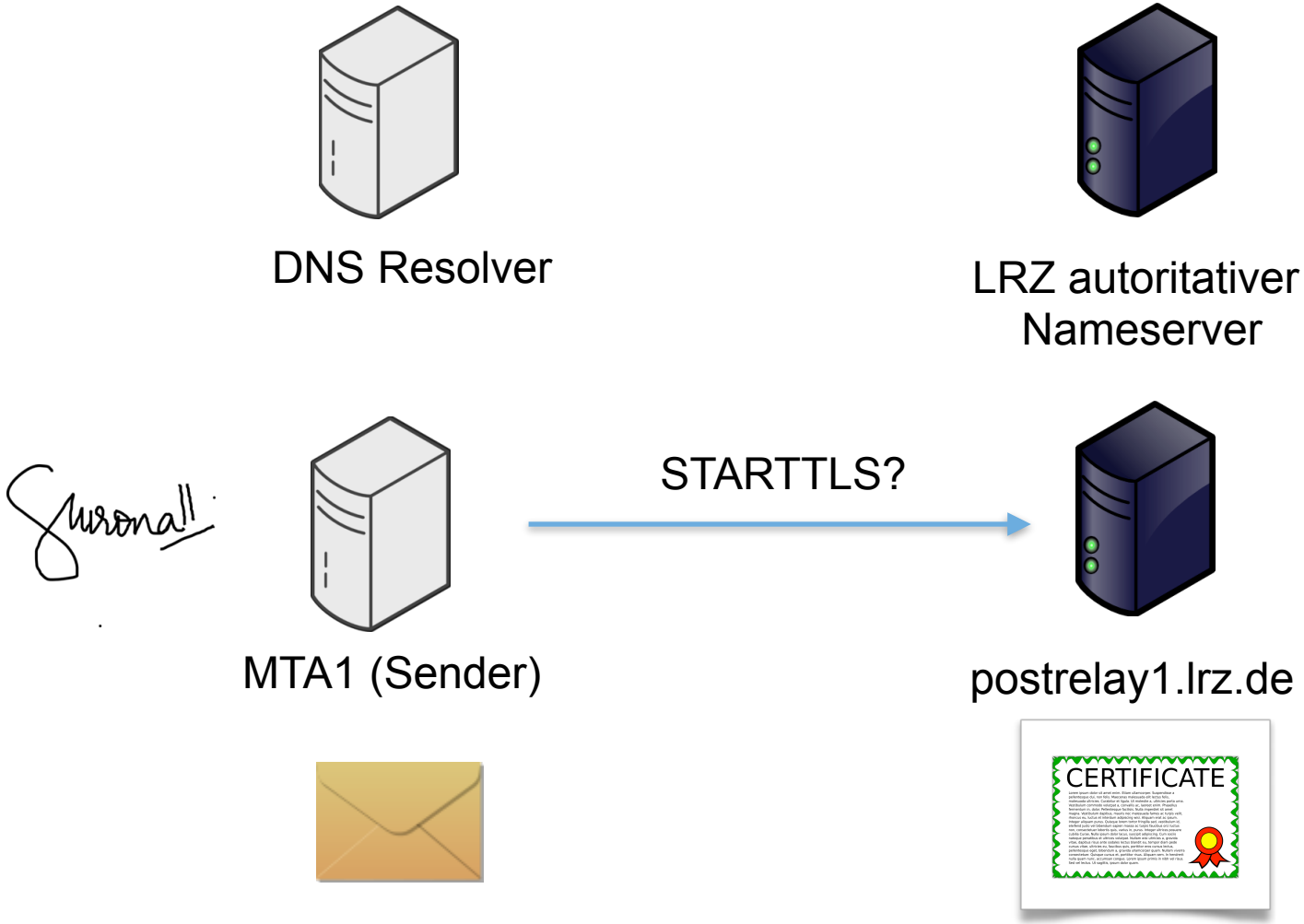
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



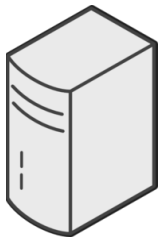
TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

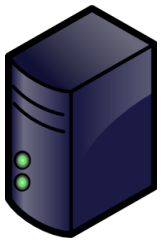


TLS-SMTP Verbindung mit DANE

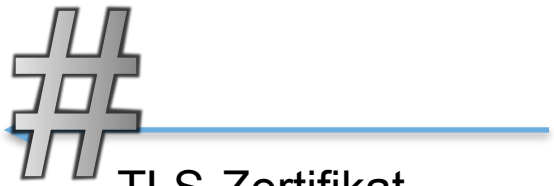
- Mailtransport via SMTP, TLS-verschlüsselt und DANE



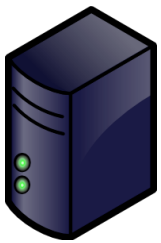
DNS Resolver



LRZ autoritativer Nameserver



TLS-Zertifikat

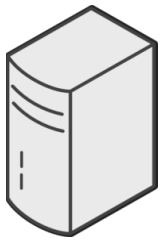


MTA1 (Sender)

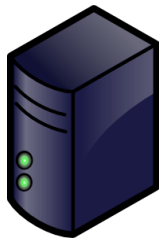
postrelay1.lrz.de

TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

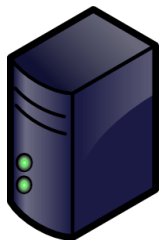


DNS Resolver



LRZ autoritativer Nameserver

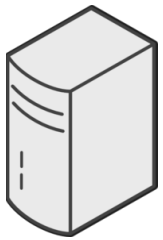
MTA1 (Sender)



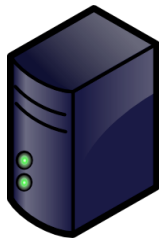
postrelay1.lrz.de

TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

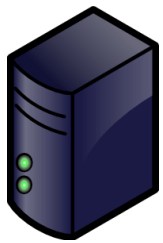


DNS Resolver



LRZ autoritativer Nameserver

MTA1 (Sender)

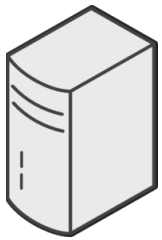


postrelay1.lrz.de

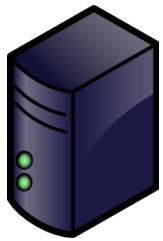
Hash = TSLA RRSIG?

TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE

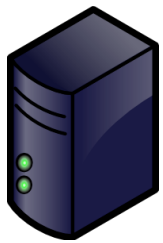


DNS Resolver



LRZ autoritativer Nameserver

MTA1 (Sender)

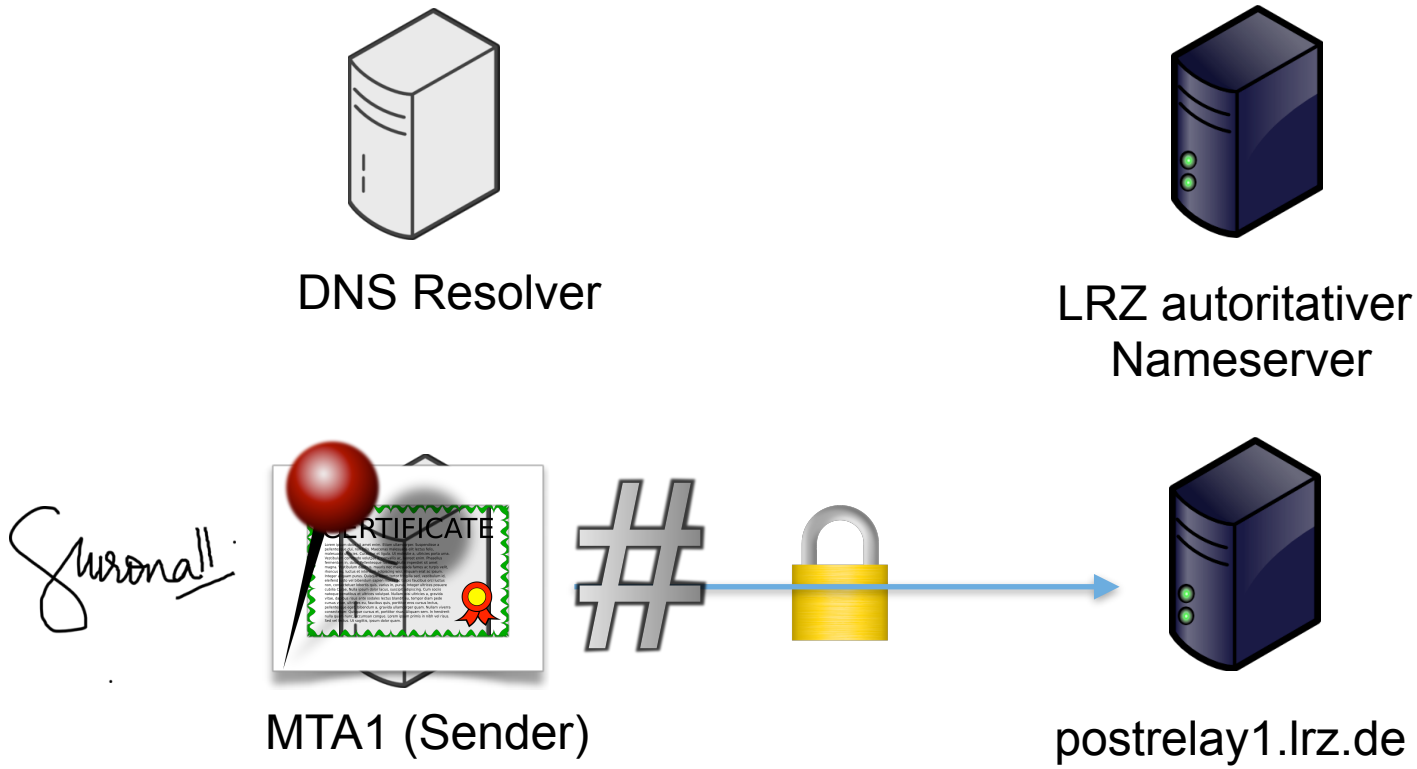


postrelay1.lrz.de

Zertifikat DANE verifiziert.

TLS-SMTP Verbindung mit DANE

- Mailtransport via SMTP, TLS-verschlüsselt und DANE



TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Port

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Protokoll

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Name des Mailservers

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Typ TLSA

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Zertifikat Nutzung

- → 0 = PKIX-TA: CA-Zertifikat, das in der Validierungskette auftauchen muss
- → 1 = PKIX-EE: CA-Root-Zertifikat, gegen das die CA-Kette validiert
- → 2 = DANE-TA: CA-Zertifikat mit dem der Key unterschrieben sein muß
- → 3 = DANE-EE: Konkreter exakter Public Key des Servers (self signed!)

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Zertifikat Nutzung

- → 0 = ~~PKIX-TA: CA-Zertifikat, das in der Validierungskette auftauchen muss~~
- → 1 = ~~PKIX-EE: CA-Root-Zertifikat, gegen das die CA-Kette validiert~~
- → 2 = DANE-TA: CA-Zertifikat mit dem der Key unterschrieben sein muß
- → 3 = DANE-EE: Konkreter exakter Public Key des Servers (self signed!)

0 und 1 sollten für SMTP nicht benutzt werden,
dieselben Probleme wie Zertifikate!

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Selector

→ 0 = Full Certificate
→ 1 = SubjectPublicKeyInfo

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Matching

→ 0 = Keinen Hash verwenden
→ 1 = SHA-256
→ 2 = SHA-512

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

Signatur

TLSA Ressource Record im Detail (RFC6698)

TLSA pinning kann verschieden interpretiert werden

_25._tcp.<servername>. IN TLSA 3 0 1 8cb0fc6c527506a053f4f1…

DANE ausgehend

- Outbound DANE benötigt Support in Software

POSTFIX

- die folgenden MTA unterstützen DANE ausgehend
  - Postfix >2.11
  - Exim 4.85 (EXPERIMENTAL_DANE)
  - Halon > 3.4-rocky-r2[(kommerziell)

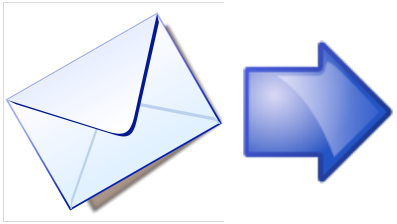
exim

- „Big player“ unterstützen noch kein DANE ausgehend
  - sendmail unterstützt DANE nicht ausgehend
  - Microsoft Exchange Server nicht ausgehend

sendmail.org

Microsoft® Exchange

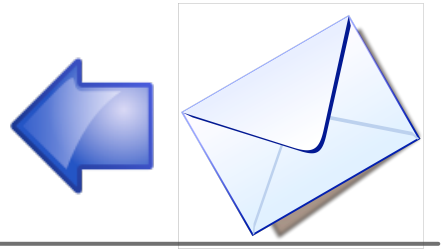
DANE ausgehend - Statistik am LRZ



- Postfix 2.11

- Statistik am LRZ (12.2. - 19.2.):
  - 2% der ausgehenden TLS-Verbindungen mit DANE gesichert
  - unter anderem zu bund.de, bayern.de, uni-kl.de, med.uni-rostock.de, posteo.de, mailbox.org, ...

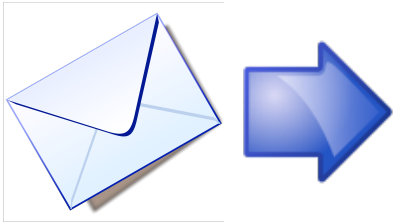
DANE eingehend

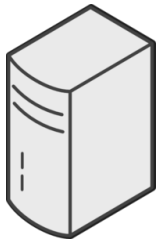


- kein Support der lokalen MTA nötig, STARTTLS genügt
- Zonen DNSSEC-signieren und Zertifikat im TLSA-Record hinterlegen
- keine Signalisierung der TLSA-Nutzung durch den Sender
  → keine Statistik


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
Konfiguration in Postfix 2.11

DANE ausgehender Mailserver





- DNSSEC-validierender Resolver, dem vertraut wird

- Postfix >= 2.11, gebaut gegen OpenSSL 1.0.0+
  - smtp_tls_security_level = dane
  - smtp_dns_support_level = dnssec
  - smtp_tls_loglevel = 1

Postfix log - Überprüfen der DANE-Verbindung

Postfix unterscheidet zwischen folgenden Verbindungen:

- Anonymous TLS Verbindung
  Anonymer Cipher, kein Zertifikat übertragen (i.A. Postfix vs. Postfix)
- Untrusted TLS Verbindung
  Zertifikat nicht verifizierbar, z.B. self-signed
- Trusted TLS Verbindung
  Zertifikat von einer vertrauenswürdigen CA
- Verified TLS Verbindung
  Zertifikat matcht DANE oder
  manuell konfigurierten Trust-Anchor

Verified TLSA-Eintrag gefunden

Postfix log:

postfix/smtp: Verified TLS connection established to dane.lhns.org[2001:db8:b51d:e5

:510::2]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)

postfix/smtp: DD8748072B: to=<danetest@lhns.org>, relay=dane.lhns.org[2001:db8:b51d:e5

:510::2]:25, delay=0.55, delays=0.09/0.02/0.25/0.1, dsn=2.0.0, status=sent (250

2.0.0 Ok: queued as 84A6B3FD38)

TLSA-Eintrag gefunden, und Signatur verifiziert

Message wird versendet

Invalid TLSA-Eintrag - Opportunistic TLS

Postfix log:

postfix/smtp: Trusted TLS connection established to dane.lhns.org[2001:db8:b51d:e5

:510::2]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)

postfix/smtp: 730D78072B: to=<danetest@lhns.org>, relay=dane.lhns.org[2001:db8:b51d:e5

:510::2]:25, delay=0.21, delays=0.09/0/0.12/0, dsn=4.7.5, status=deferred (Server certificate not verified)

Dem Zertifikat würde nach normalen „chain-of-trust“ vertraut.

Aber der Signatur-Eintrag im TLSA entspricht nicht dem im Zertifikat.

E-Mail deferred, möglicherweise MTM-Attacke.

Opportunistic TLS - Kein TLSA-Eintrag

Postfix log:

postfix/smtp: Trusted TLS connection established to dane.lhns.org[2001:db8:b51d:e5

:510::2]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)

postfix/smtp: DD8748072B: to=<danetest@lhns.org>, relay=dane.lhns.org[2001:db8:b51d:e5

:510::2]:25, delay=0.55, delays=0.09/0.02/0.25/0.1, dsn=2.0.0, status=sent (250

2.0.0 Ok: queued as 84A6B3FD38)

Es wird kein TLSA-Eintrag gefunden, und SMTP fällt auf “opportunistic TLS” zurück, Message sent

smtp tls security level = may

Kein TLSA-Eintrag - TLS verpflichtend

Postfix log:

postfix/smtp: warning: TLS policy lookup for lhns.org/dane.lhns.org: no TLSA records

found

postfix/smtp: E9BCC8072B: to=<danetest@lhns.org>, relay=none, delay=0.15, delays

=0.13/0.02/0/0, dsn=4.7.5, status=deferred (no TLSA records found)

Es wird kein TLSA-Eintrag gefunden, Postfix verschiebt das Versenden (“deferred”)

echo "lhns.org dane-only" >> /etc/postfix/tls_policy
postmap /etc/postfix/tls_policy

Postfix TLS Security Levels

- In /etc/postfix/main.cf wird der globale Security-Level gesetzt (smtp_tls_security_level)
  - none — kein TLS
  - may — opportunistisches TLS
  - dane — opportunistisches TLS + DANE
  - dane-only — DANE verpflichtend
  - verify/secure — verifiziertes TLS verpflichtend

- Kann über smtp_tls_policy_maps pro Domain überschrieben werden, bspw. um fehlerhafte TLSA-Einträge zu übergehen oder bekannte Ziele auch ohne DANE zu pinnen

Übung - DANE/TLSA mit Postfix 2.11

DANE - Schritt für Schritt mit Postfix 2.11

1. TLSA-Eintrag erstellen
2. TLSA-Eintrag überprüfen
3. Outbound Mailserver Konfiguration
4. TLSA-Verification testen

TLSA-Eintrag aus Zertifikat erstellen

Workshop-VMs haben bereits ein selbstsigniertes Zertifikat

/etc/postfix/ssl/certs/server.pem (Zertifikat)
/etc/postfix/ssl/certs/server.key (Schlüssel, mit dem signiert wurde)

Mailserver MX Resource Record in Zone:

wsXX.ws.dnssec.bayern. 300 IN MX 100 dnssec-wsXX.dnssec.bayern

TLSA-Eintrag aus Zertifikat erzeugen

- Eintrag kann direkt mit openssl erzeugt werden - nur manchmal intuitiv (für 3 0 1)

  openssl x509 -in /etc/postfix/ssl/certs/server.pem -outform DER | openssl sha256 | cut d'=' -f2 | awk '{printf "IN TLSA 3 0 1 %s\n", $NF}'
  IN TLSA 3 0 1 b80b5de3c513eccd84e2cfdbf6e4b4e2b05e513140a27d05b4a76d40b46b7590

- Online-Dienste für die Erstellung von beliebigen TLSA-Records
  https://www.huque.com/bin/gen_tsla
  https://de.ssl-tools.net/tsla-generator

- CLI-Tools, z.B. /usr/bin/tlsa aus hash-slinger oder ldns-dane
  tlsa --create --port 25 --certificate /etc/postfix/ssl/certs/server.pem --selector 0 dnssec-wsXX.wsXX.ws.dnssec.bayern
  _25._tcp. dnssec-ws01.ws01.ws.dnssec.bayern. IN TLSA 3 0 1 850e901d6c0989d4421f13953653c601e70fe69afdaa513274965f0f6e61471d

- **ldns-dane create dnssec-ws01.ws01.ws.dnssec.bayern 25 **
  -c /etc/postfix/ssl/certs/server.pem 3 0 1
  letzteres ist zu empfehlen

TLSA-RR mit SSL-Tools-net erstellen

TLSA Eintrag generieren

Benutze dieses Tool, um einen **TLSA-Eintrag** nach RFC 6698 für deine Domain zu erzeugen. TLS-Einträge werden für **DANE** benötigt (DNS-Based Authentication of Named Entities).

Usage

- ○ PKIX-TA: CA Constraint
- ○ PKIX-EE: Service Certificate Constraint
- ○ DANE-TA: Trust Anchor Assertion
- ○ DANE-EE: Domain Issued Certificate

Selector

- ○ Use full certificate
- ◉ Use subject public key

Matching Type

- ○ Full: No Hash
- ◉ SHA-256 Hash
- ○ SHA-512 Hash

Certificate

-----BEGIN CERTIFICATE-----

Port 443

Protocol tcp

Domain

Generieren

Das Zertifikat muss per copy-and-paste in die Webseite eingefügt werden.

TLSA-Eintrag mit dig überprüfen

TLSA-Eintrag mit dig aus der Zone auslesen:

dig -t tlsa _25._tcp.dnssec-wsXX.wsXX.ws.dnssec.bayern

;; ANSWER SECTION:
_25._tcp.dnssec-ws01.dnssec.bayern. 300 IN TLSA 3 0 1
3ED7344F7051A4B6C28A6445E5BC94FB57F0E25CBCFADB10903958CB
6E6C63CD

;; AUTHORITY SECTION:
ws01.ws.dnssec.bayern. 300 IN NS dnssec-ws01.dnssec.bayern.

dig -t tlsa _25._tcp.dnssec-wsXX.wsXX.ws.dnssec.bayern +short [@127.0.0.1]
Liefert nur den nackten TLSA-Typ und Fingerprint ohne Server und Port

3 0 1 3ED7344F7051A4B6C28A6445E5BC94FB57F0E25CBCFADB10903958CB
6E6C63CD

TLSA/DANE mit posttls-finger überprüfen

posttls-finger wsXX.ws.dnssec.bayern

- **Ohne TLSA:**

posttls-finger: dnssec-ws01.ws01.ws.dnssec.bayern[2001:4ca0:800:2:250:56ff:fe8f:5584]:25 CommonName debian.srv.lrz.de
*posttls-finger: **certificate verification failed** for dnssec-ws01.ws01.ws.dnssec.bayern[2001:4ca0:800:2:250:56ff:fe8f:5584]:25: self-signed certificate*
posttls-finger: dnssec-ws01.ws01.ws.dnssec.bayern[2001:4ca0:800:2:250:56ff:fe8f:5584]:25: subject_CN=debian.srv.lrz.de, issuer_CN=debian.srv.lrz.de, fingerprint=82:DA:1B:BB:48:07:D1:CD:6C:22:63:32:77:27:8C:24:2F:11:53:F2, pkey_fingerprin t=39:CD:0F:39:3C:11:90:60:7C:1F:99:29:7F:24:42:2F:D3:CE:5C:54
*posttls-finger: **Untrusted TLS connection established** to dnssec-ws01.ws01.ws.dnssec.bayern[2001:4ca0:800:2:250:56ff:fe8f:5584]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)*

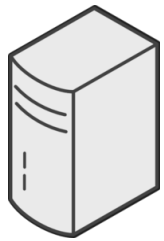
- **Mit TLSA:**

*posttls-finger: **using DANE RR**: _25._tcp.dnssec-ws01.ws01.ws.dnssec.bayern IN TLSA 3 0 1 A5:2E:5C:88:04:4A:0A:65:C9:2E:FD:13:3E:D9:09:19:DD:9A:11:81:2E:EB:2D:2D:A2:0E:E7:61:26:86:AA:6F*
posttls-finger: dnssec-ws01.ws01.ws.dnssec.bayern[138.246.99.206]:25: depth=0 **matched end entity certificate sha256 digest** A5:2E:5C:88:04:4A:0A:65:C9:2E:FD:13:3E:D9:09:19:DD:9A:11:81:2E:EB:2D:2D:A2:0E:E7:61:26:86:AA:6F
posttls-finger: dnssec-ws01.ws01.ws.dnssec.bayern[138.246.99.206]:25 CommonName debian.srv.lrz.de
posttls-finger: dnssec-ws01.ws01.ws.dnssec.bayern[138.246.99.206]:25: subject_CN=debian.srv.lrz.de, issuer_CN=debian.srv.lrz.de, fingerprint=82:DA:1B:BB:48:07:D1:CD:6C:22:63:32:77:27:8C:24:2F:11:53:F2, pkey_fingerprint=39:CD:0F:39:3C:11:90:60:7C:1F:99:29:7F:24:42:2F:D3:CE:5C:54
*posttls-finger: **Verified TLS connection established** to dnssec-ws01.ws01.ws.dnssec.bayern[138.246.99.206]:25: TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)*

Outbound Mailserver Postfix Konfiguration

- DNSSEC-validierender Resolver, dem man vertraut
- Postfix 2.11 Konfiguration

  smtp_dns_support_level = dnssec
  smtp_tls_security_level = dane



Resolving Nameserver

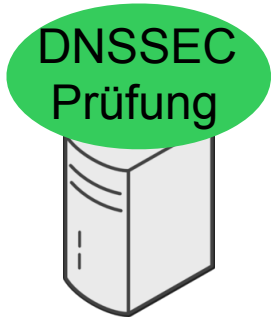
- In **/etc/postfix/main.cf** ändern, **postfix reload** eingeben
- Testmail an DANE-fähigen Admin-Server (wsadm) schicken
  echo „Test“ | sendmail test@wsadm.ws.dnssec.bayern
- Logdatei /var/log/mail.long ansehen

Jan 10 16:29:42 dnssec-ws01 postfix/pickup[36260]: BC94D2F725: uid=0 from=<root>
Jan 10 16:29:42 dnssec-ws01 postfix/cleanup[36517]: BC94D2F725: message-id=<20170110152942.BC94D2F725@dnssec-ws01.dnssec.bayern>
Jan 10 16:29:42 dnssec-ws01 postfix/qmgr[42797]: BC94D2F725: from=<root+dnssec-ws01@srv.mwn.de>, size=282, nrcpt=1 (queue active)
Jan 10 16:29:42 dnssec-ws01 postfix/discard[36519]: BC94D2F725: to=<test@ws01.ws.dnssec.bayern>, relay=none, delay=0.03, delays=0.01/0.01/0/0, dsn=2.0.0, **status=sent** (ws01.ws.dnssec.bayern)

Outbound Mailserver Postfix Konfiguration

- DNSSEC-validierender Resolver, dem man vertraut
- Postfix 2.11 Konfiguration

  smtp_dns_support_level = dnssec
  smtp_tls_security_level = dane



Resolving
Nameserver

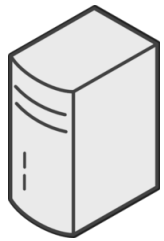
- In **/etc/postfix/main.cf** ändern, **postfix reload** eingeben
- Testmail an DANE-fähigen Admin-Server (wsadm) schicken
  echo „Test“ | sendmail test@wsadm.ws.dnssec.bayern
- Logdatei /var/log/mail.long ansehen

Jan 10 16:29:42 dnssec-ws01 postfix/pickup[36260]: BC94D2F725: uid=0 from=<root>
Jan 10 16:29:42 dnssec-ws01 postfix/cleanup[36517]: BC94D2F725: message-id=<20170110152942.BC94D2F725@dnssec-ws01.dnssec.bayern>
Jan 10 16:29:42 dnssec-ws01 postfix/qmgr[42797]: BC94D2F725: from=<root+dnssec-ws01@srv.mwn.de>, size=282, nrcpt=1 (queue active)
Jan 10 16:29:42 dnssec-ws01 postfix/discard[36519]: BC94D2F725: to=<test@ws01.ws.dnssec.bayern>, relay=none, delay=0.03, delays=0.01/0.01/0/0, dsn=2.0.0, **status=sent** (ws01.ws.dnssec.bayern)

Outbound Mailserver Postfix Konfiguration

- DNSSEC-validierender Resolver, dem man vertraut
- Postfix 2.11 Konfiguration

  smtp_dns_support_level = dnssec
  smtp_tls_security_level = dane



Resolving Nameserver

- In **/etc/postfix/main.cf** ändern, **postfix reload** eingeben
- Testmail an DANE-fähigen Admin-Server (wsadm) schicken
  echo „Test“ | sendmail test@wsadm.ws.dnssec.bayern
- Logdatei /var/log/mail.long ansehen

Jan 10 16:29:42 dnssec-ws01 postfix/pickup[36260]: BC94D2F725: uid=0 from=<root>
Jan 10 16:29:42 dnssec-ws01 postfix/cleanup[36517]: BC94D2F725: message-id=<20170110152942.BC94D2F725@dnssec-ws01.dnssec.bayern>
Jan 10 16:29:42 dnssec-ws01 postfix/qmgr[42797]: BC94D2F725: from=<root+dnssec-ws01@srv.mwn.de>, size=282, nrcpt=1 (queue active)
Jan 10 16:29:42 dnssec-ws01 postfix/discard[36519]: BC94D2F725: to=<test@ws01.ws.dnssec.bayern>, relay=none, delay=0.03, delays=0.01/0.01/0/0, dsn=2.0.0, **status=sent** (ws01.ws.dnssec.bayern)

https://dane.sys4.de

dnssec-ws01.ws01.ws.dnssec.bayern

DNSSEC ✓ TLSA ✓ SMTP !

The domain lists the following MX entries:

0 dnssec-ws01.ws01.ws.dnssec.bayern

DNSSEC ✓ TLSA ✓ SMTP ! Show Details

138.246.99.206: Connection timed out

IP Addresses

138.246.99.206

2001:4ca0:800:2:250:56ff:fe8f:5584

Usable TLSA Records

```
3, 0, 1 004c16b9a73d3b38[...]eb4d4173bcdb97f9
```

DANE - online mit web service testen

https://de.ssl-tools.net/mailservers/

- erlaubt die Angabe einer Zone, wsXX.ws.dnssec.bayern
- Liest die MX-Einträge aus
- Prüft die Mailserver auf TLS/TLSA
- Vergleich des TLSA-Eintrags mit dem Zertifikat-Fingerprint

DANE

DNS-based Authentication of Named Entities (DANE) is a protocol to allow X.509 certificates to be bound to DNS using TLSA records and DNSSEC.

Name	Options	DNSSEC	Matches
_25._tcp.dnssec-ws01.ws01.ws.dnssec.bayern	DANE-EE: Domain Issued Certificate Use full certificate SHA-256 Hash	✔ valid	✔ valid

Selbst-signiertes Zertifikat (keine offizielle CA)

Certificates

dnssec-ws01.dnssec.bayern First seen at: 4 days ago

Certificate chain

dnssec-ws01.dnssec.bayern (Certificate is self-signed.)
1020 days remaining 2048 bit sha256WithRSAEncryption
⚠ Unknown Authority

Subject

Country (C)	DE
State (ST)	Bavaria
Locality (L)	Garching
Organization (O)	BADW
Organizational Unit (OU)	Leibniz-Rechenzentrum
Common Name (CN)	dnssec-ws01.dnssec.bayern

Issuer

Certificate is self-signed.

validity period

Not valid before	2017-02-09
Not valid after	2019-11-30

Fingerprints

SHA256	3E:D7:34:4F:70:51:A4:B6:C2:8A:64:45:E5:BC:94:FB:57:F0:E2:5C:BC:FA:DB:10:90:39:58:CB:6E:6C:63:CD
SHA1	01:30:27:A9:5A:04:39:29:17:A6:8B:48:F9:62:0D:BC:78:78:3A:C6

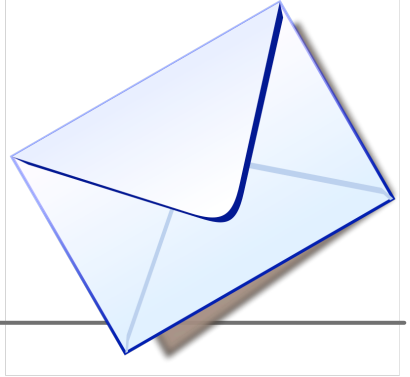
− less details


Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
DNSSEC/DANE - Unterstützung in Clients

Webbrowsers

- Kein Browser unterstützt „out-of-the-box“ DNSSEC/DANE
- Internet Explorer
  https://labs.nic.cz/en/dnssec-ie.html (pre-release)
- Javascript-Plugin kann DNSSEC-Verifikation nachrüsten
  - Google Chrome
    http://www.internetsociety.org/deploy360/resources/how-to-add-dnssec-support-to-google-chrome/
  - Mozilla Firefox
    http://www.internetsociety.org/deploy360/resources/how-to-add-dnssec-support-to-mozilla-firefox/

Email-Clients



- Bisher nur sehr geringe Unterstützung
- Keine „out-of-the-box“-Lösung
- Zusatztools und Plugins erlauben TLSA Verification
- Thunderbird-Startskript mit DANE-Test

  (https://www.privacy-handbuch.de/download/thunderbird_mit_danetest.sh)


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
DNSSEC für weitere abgesicherte Dienste

SSH Host Key Verification

DNSSEC kann auch dazu verwendet werden, SSH Host Keys zu authentifizieren.

- SSH Host key wird in der DNS Zone gespeichert
- SSHFP record
- /etc/ssh/ssh_config oder ~/.ssh/config

VerifyHostKeyDNS yes

- ssh -o "VerifyHostKeyDNS=yes" dnssec-ws01 (fallweise)
- DNSSEC SSHFP Überprüfung als zwingende Voraussetzung
  /etc/ssh/ssh_config

```
Host *
    StrictHostKeyChecking yes
```

ssh-Verbindung mit DNS Hostkey verification

```
ssh -o VerifyHostKeyDNS=yes root@dnssec-ws01.dnssec.bayern -i ~/.ssh/id_rsa
The authenticity of host 'dnssec-ws01.dnssec.bayern (2001:4ca0:800:2:250:56ff:fe8f:5584)' can't be established.
ECDSA key fingerprint is SHA256:X0zVrd7dqEO4z4dmgxkpoKrFbZLgO7MPG+v+vOPkeMY.
Matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)? yes
```

Unterstützung in ssh-Clients

- OpenSSH on Linux
- macOS 10.12 „Sierra“ (bug bis macOS 10.11)
- Putty - auf der Wunschliste seit 2007

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1   CF239AE438FF149185378D9735BE42B519416D0F
```

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1   CF239AE438FF149185378D9735BE42B519416D0F
```

Zielhost

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1   CF239AE438FF149185378D9735BE42B519416D0F
```

Protrokollart

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern      IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern      IN  SSHFP  2 1  CF239AE438FF149185378D9735BE42B519416D0F
```

RR-Typ

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1  CF239AE438FF149185378D9735BE42B519416D0F
```

Host-Key Algorithmus

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1  CF239AE438FF149185378D9735BE42B519416D0F
```

Hash-Art des FP

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639F2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1  CF239AE438FF149185378D9735BE42B519416D0F
```

Fingerprint

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1   CF239AE438FF149185378D9735BE42B519416D0F
```

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1   CF239AE438FF149185378D9735BE42B519416D0F
```

- sshfp erstellt RR Eintrag für SSH Hostkey Fingerprint (-s scan for public key)

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1   CF239AE438FF149185378D9735BE42B519416D0F
```

- sshfp erstellt RR Eintrag für SSH Hostkey Fingerprint (-s scan for public key)
- Unterstützte Algorithmen:
  1 ssh-rsa
  2 ssh-dsa
  3 ecdsa
  4 ed25519

SSHFP RR in der Zone


```
$ sshfp -s dnssec-ws01.dnssec.bayern
dnssec-ws01.dnssec.bayern     IN  SSHFP  1 1  57D79129FA85BCC0D9E15CE25C96E639E2DB3316
dnssec-ws01.dnssec.bayern     IN  SSHFP  2 1   CF239AE438FF149185378D9735BE42B519416D0F
```

- sshfp erstellt RR Eintrag für SSH Hostkey Fingerprint (-s scan for public key)
- Unterstützte Algorithmen:
  1 ssh-rsa
  2 ssh-dsa
  3 ecdsa
  4 ed25519
- SSHFP RR record dann in der Zone eintragen

DNS SSHFP-Einträge überprüfen

DNS SSHFP-Einträge überprüfen

SSHFP RR records mit dig abrufen

DNS SSHFP-Einträge überprüfen

SSHFP RR records mit dig abrufen

$ **dig +short +dnssec dnssec-ws01.dnssec.bayern IN SSHFP**
4 2 29AEEA1299BBB0E9EAF699AAC433EEFC44B30AA6E866B5D7ECF7E6F5 F5471B11
1 2 16323945829B3CBD3735409F1E79D1D981CE2DE4261DB74566872901 59766A08
3 2 5F4CD5ADDEDDA843B8CF8766831929A0AAC56D92E03BB30F1BEBFEBC E3E478C6
2 2 B17CCCCD54B29312743AE3B86168A5381C596F0D4CE4F2B8AED0DC00 43DE9F88
SSHFP 8 3 3600 20170126200838 20161103190838 36675 dnssec.bayern. yBwxi5qNwXKh2xJe0/
xV8e9UtNxt7nUdGHBwdVzn9W9JPV2PXwtACy/U PJij/
Br2jmxIdx9a+xoF8WoUB7ktX9gmP6ibNj9Yr35bbeeID1xXedTF
qtcxRcAWPeUScG+lwnmjpdZcnyryUq4ldpbFczfokCB+GCT+SWibkNab 7cH8vE0s4QtBxDl5Ug0VnPgZlv/
XkYxDHUA0GU2TKCeyx4o2qQccu Gu/1Y4R+gB1dUolYcxZy/
TzR2hOuo+NuS9GMY5VOz4ZVcd8MyNQOEXgZ
WSKWeQsmCuyoLfcXPCZ0XupfqsisL0gC3ZVIcn0j5dV21JmbNoU5a8kc yZ6HRg==

DNS SSHFP-Einträge überprüfen

SSHFP RR records mit dig abrufen

$ **dig +short +dnssec dnssec-ws01.dnssec.bayern IN SSHFP**
4 2 29AEEA1299BBB0E9EAF699AAC433EEFC44B30AA6E866B5D7ECF7E6F5 F5471B11
1 2 16323945829B3CBD3735409F1E79D1D981CE2DE4261DB74566872901 59766A08
3 2 5F4CD5ADDEDDA843B8CF8766831929A0AAC56D92E03BB30F1BEBFEBC E3E478C6
2 2 B17CCCCD54B29312743AE3B86168A5381C596F0D4CE4F2B8AED0DC00 43DE9F88
SSHFP 8 3 3600 20170126200838 20161103190838 36675 dnssec.bayern. yBwxi5qNwXKh2xJe0/
xV8e9UtNxt7nUdGHBwdVzn9W9JPV2PXwtACy/U PJij/
Br2jmxIdx9a+xoF8WoUB7ktX9gmP6ibNj9Yr35bbeeID1xXedTF
qtcxRcAWPeUScG+lwnmjpdZcnyryUq4ldpbFczfokCB+GCT+SWibkNab 7cH8vE0s4QtBxDl5Ug0VnPgZlv/
XkYxDHUA0GU2TKCeyx4o2qQccu Gu/1Y4R+gB1dUolYcxZy/
TzR2hOuo+NuS9GMY5VOz4ZVcd8MyNQOEXgZ
WSKWeQsmCuyoLfcXPCZ0XupfqsisL0gC3ZVIcn0j5dV21JmbNoU5a8kc yZ6HRg==

DNSSEC mit dig auf Vollständigkeit und Sicherheit überprüfen

ad Flag!

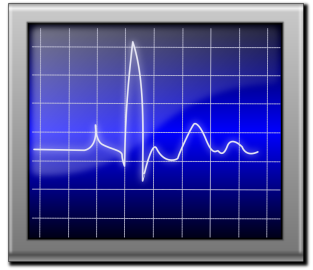
DNSSEC für andere Anwendungen

DNSSEC läßt sich zur Absicherung von kryptographischen Informationen in anderen Anwendungen verwenden

- XMPP/Jabber
- SIP
- SSHFP — SSH Host key fingerprint
- IPSec Key — IPsec keys in DNS
- OpenPGP Key — PGP keys in DNS
- *S/MIME* — Zertifikate Benutzern zuordnen
- …


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
Monitoring DNSSEC/DANE - mit Icinga2

DNSSEC Monitoring



Eine **korrekte DNSSEC-Absicherung** einer Zone/Domäne ist für die Erreichbarkeit der Zone/Domäne **essentiell** - sonst erhalten validierende Nameserver ein SERVAIL und die Zone ist vom Netz.

Die folgenden Einstellungen müssen richtig sein und können mit CLI-Tools überprüft werden:

- MTU Größe
- TCP Port 53
- NTP
- named-checkconf überprüft BIND Konfiguration
- named-checkzone überprüft Zonen-Einträge
- ldns-verify-zone check nach DNSSEC-Fehlern
- DANE-TLSA-Eintrag / Vergleich mit Zertifikatsfingerprint

Check-Skripte

- teilweise mit DNSSEC CLI tools selbst geschrieben
- Rückgabe-Werte „Nagios“-konform
  0=OK, 1=WARNING, 2=CRITICAL, 3=UNKNOWN
- Bash, Perl, Python… nutzbar
- Einbindung hängt vom Monitoring-System ab
- Icinga2:
  check_script → check_command → service
- Nagios ähnlich, service-Syntax differiert, kein Templating

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```


ldns-verify-zone ausführen

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Aufruf von ldns-verify-zone schlug fehl

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Failed-Nachricht + Stdout/stderr

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Rückgabewert: UNKNOWN

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

ldns-verify-zone liefert Zone=OK

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```


OK + Stdout mitgeben

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Rückgabewert: OK

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Signaturen nur noch 20 Tage gültig

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Rückgabewert: WARNING

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

sonst: kritischer Fehler!

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Stdout/stderr mitgeben

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Rückgabewert: CRITICAL

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Exit mit Rückgabewert

Beispiel: check_dnssec-ldns Skript

```
# Do the work using ldns-verify-zone
OUTPUT=`ldns-verify-zone -e $EXPIRATION $ZONEFILE 2>&1`

if [ "$?" -eq 1 ]
then
  echo "ldns-verify-zone failed.|$OUTPUT"
  RETVAL=$UNKNOWN
fi

if [ "$OUTPUT" == "Zone is verified and complete" ]
then
  echo "ldns-verify-zone OK.|$OUTPUT"
  RETVAL=$OK
elif [[ "$OUTPUT" == *"signature will expire"* ]]
then
  echo "$OUTPUT"
  RETVAL=$WARN
else
  echo "$OUTPUT"
  RETVAL=$CRIT
fi

exit $RETVAL
```

Einbindung in Icinga2: CheckCommand

- Folgt weitestgehend Icinga2-Standard-Schema
- Kommandoparameter-Übergabe mit „arguments“
- keine Interaktive Definition, aber dynamisches Reload möglich

```
# Check DNSSEC with ldns-verify-zone
object CheckCommand "check_dnssec-ldns" {
        import "plugin-check-command"
        command = [PluginDir + "/check_dnssec-ldns"]
        arguments = {
                "-z" = "/etc/bind/ws01.ws.dnssec.bayern.zone.signed"
                "-e" = "P10D"
        }
}
```

Einbindung in Icinga2: service definition

- Service-Definition macht CheckCommand zum Monitoring
- bestimmt auch, welche Hosts diesen Service-Check zugeordnet bekommen
- Check-Interval, Admin-Benachrichtigung usw. service-spezifisch bestimmbar

```
apply Service "DNSSEC-ldns" {
  import „generic-service"
  check_interval = 30s

  check_command = "check_dnssec-ldns"

  assign where host.address6
}
```

Icinga2 Service Detail

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: BIND configuration

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: BIND process „named“

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: Zone configuration

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	**OK**	**02-27-2017 17:37:17**	**12d 3h 22m 15s**	**1/5**	**Zone configuration ws01.ws.dnssec.bayern is ok.**	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: Zones loaded

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	**OK**	**02-27-2017 17:37:17**	**11d 0h 19m 29s**	**1/5**	**zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)**	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: DANE SMTP check

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: DS record KSK-ID?

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: KSKs und ZSKs

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: ldns-verify-zone?

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: Postfix master?

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	**OK**	**02-27-2017 17:37:19**	**5d 0h 35m 37s**	**1/5**	**PROCS OK: 1 process with command name 'master'**	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: Postfix queue Ok?

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	**OK**	**02-27-2017 17:37:18**	**5d 0h 17m 51s**	**1/5**	**Mailqueue OK - 0 messages on queue**	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: dig Nameserver?

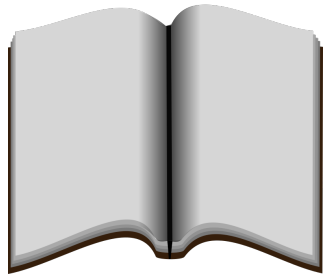
localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	**OK**	**02-27-2017 17:37:18**	**0d 7h 7m 43s**	**1/5**	**DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)**	
	dns	OK	02-27-2017 17:37:18	12d 21h 42m 25s	1/5	DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	

Icinga2 Service Detail: DNS Resolver?

localhost	BIND Conf	OK	02-27-2017 17:37:18	12d 4h 9m 52s	1/5	/usr/sbin/named configuration is ok.	
	BIND Running	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	PROCS OK: 1 process with command name 'named'	
	BIND Zone	OK	02-27-2017 17:37:17	12d 3h 22m 15s	1/5	Zone configuration ws01.ws.dnssec.bayern is ok.	
	BIND Zones	OK	02-27-2017 17:37:17	11d 0h 19m 29s	1/5	zone ws01.ws.dnssec.bayern/IN: loaded serial 1 (DNSSEC signed)	
	DANE SMTP	OK	02-27-2017 17:37:19	0d 5h 23m 2s	1/5	DANE OK - dnssec-ws01.ws01.ws.dnssec.bayern:25 cert matches TLSA record	
	DNSSEC DS 2	OK	02-27-2017 17:37:17	0d 7h 7m 40s	1/5	KSK: 64867 found in DS RR.	
	DNSSEC Keys	OK	02-27-2017 17:37:18	11d 1h 0m 29s	1/5	KSK: 64867 ZSK: 56961	
	DNSSEC-ldns	OK	02-27-2017 17:37:19	4d 5h 0m 6s	1/5	ldns-verify-zone OK.	
	Postfix	OK	02-27-2017 17:37:19	5d 0h 35m 37s	1/5	PROCS OK: 1 process with command name 'master'	
	Postfix Queue	OK	02-27-2017 17:37:18	5d 0h 17m 51s	1/5	Mailqueue OK - 0 messages on queue	
	dig	OK	02-27-2017 17:37:18	0d 7h 7m 43s	1/5	DNS OK - 0.008 seconds response time (dnssec-ws01.ws01.ws.dnssec.bayern. 300 IN A 138.246.99.206)	
	dns	**OK**	**02-27-2017 17:37:18**	**12d 21h 42m 25s**	**1/5**	**DNS OK: 0.008 seconds response time. lrz.de returns 129.187.255.234**	
	http	OK	02-27-2017 17:37:18	14d 4h 10m 53s	1/5	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time	
	load	OK	02-15-2017 10:22:48	14d 4h 10m 49s	1/5	OK - load average: 0.00, 0.01, 0.00	
	mtu	OK	02-27-2017 17:37:18	0d 9h 47m 28s	1/5	MTU size is 4064 Bytes	
	ntp	OK	02-27-2017 17:37:18	12d 21h 42m 26s	1/5	NTP OK: Offset 0.001079142094 secs	
	ping4	OK	02-27-2017 17:37:22	0d 0h 0m 19s	1/5	PING OK - Packet loss = 0%, RTA = 0.05 ms	
	ping6	OK	02-27-2017 17:37:23	0d 0h 0m 18s	1/5	PING OK - Packet loss = 0%, RTA = 0.07 ms	
	ssh	OK	02-15-2017 10:22:43	14d 4h 10m 55s	1/5	SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)	
	tcp port 53	OK	02-27-2017 17:37:18	0d 7h 7m 34s	1/5	TCP OK - 0.000 second response time on port 53	


lrz
Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
Weitere Informationsquellen & Fragen

Nützliche Ressourcen - Anleitungen

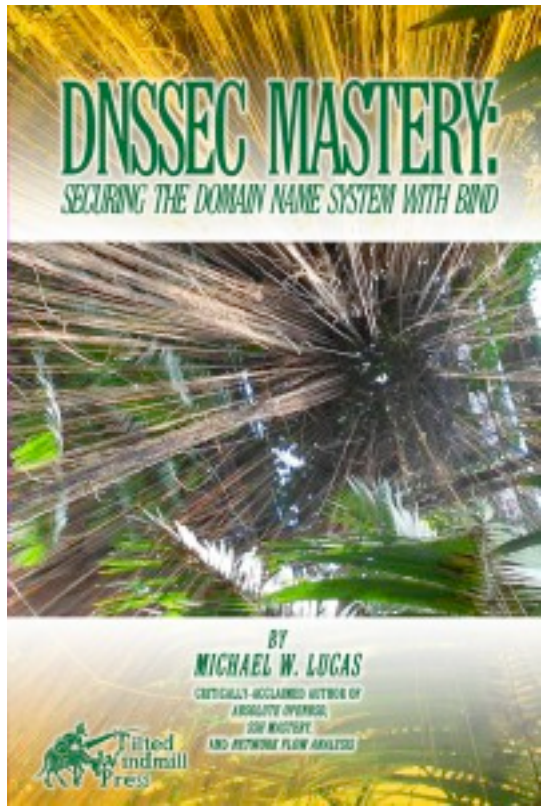


- DNSSEC HowTo - A tutorial in disguise
  https://www.nlnetlabs.nl/publications/dnssec_howto/dnssec_howto.pdf
- BIND DNSSEC Guide
  https://users.isc.org/~jreed/dnssec-guide/dnssec-guide.html
- BIND Automatic Signing
  http://www.average.org/dnssec/dnssec-configuring-auto-signed-dynamic-zones.txt
- White paper Deploying DNSSEC
  https://www.surf.nl/binaries/content/assets/surf/en/knowledgebase/2012/rapport_Deploying_DNSSEC_v20.pdf
- Heise Artikel http://www.heise.de/netze/artikel/Transitschutz-DNSSEC-und-DANE-auf-Linux-Servern-konfigurieren-2636175.html

Nützliche Ressourcen - Youtube Videos/Buch

- Auf Youtube finden sich viele kurze oder auch ausführliche Präsentationen zu DNSSEC/DANE

- Buch „DNSSEC Mastery“ (Michael W.Lucas) (enthält auch einen kurzen DANE-Teil) Einrichtung mit BIND 9.9



Vielen Dank für Ihre Aufmerksamkeit! Fragen?