Chemnitzer Linux-Tage
13. und 14. März 2010

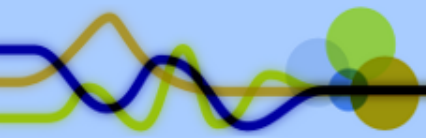
DNSSec-Workshop

Chemnitzer Linux-Tage 2010

Marcus Obst
Heiko Schlittermann

Inhalt

- Warum DNSSec / Motivation
- Bisherige Mechanismen (TSIG)
- DNSSec Grundlagen
- DNSSec-Server Setup (neue Zone)
- DNSSec-Resolver Setup (Validierung)
- Domain Lookaside Validation (DLV)
- Dynamische Updates

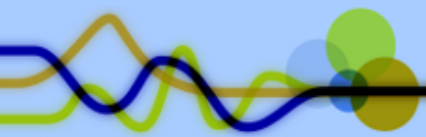


Voraussetzung/ Installation

- Netzzugang, WLAN
- Bitte installieren, wenn möglich!
  - Bind9 (Nameserver, Reslver)
  - Bind9utils (dnssec-keygen)
  - dnsutils (dig)
  - Unbound (leichtgewichtiger DNS resolver)
- Bitte nscd deaktivieren (killall nscd)

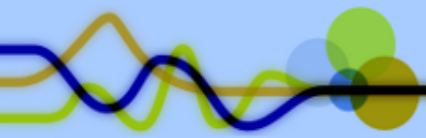
Warum DNSSec / Motivation

- DNS Cache Poisoning
  - Verschmutzung eines (Provider-) DNS-Servers
  - Erraten der ID/Portnummer (Zufall bekannt!)
  - Nächste Anfrage bekannt
- DNS-Filter, DNS-Proxies
  - Umleiten (aller) DNS-Anfragen
  - Eintragen eines „eigenen“ DNS-Server hilft nicht!



DNS-Cache Poisoning

- Angriff auf einen bestehenden DNS-Server
- Dieser sollte möglichst von vielen Clients genutzt werden

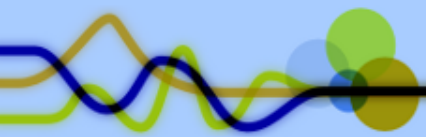


DNS-Proxy mit Bordmitteln (1)

- Beliebiger Linux-Rechner, der als Gateway, Router, Bridge arbeitet
- Umleiten alle DNS-Anfragen (UDP-Port 53)
- Präparierter Nameserver liefert eigenen Antworten
- Komponenten:
  - iptables (Port-Forwarding)
  - Dnsmasq (einfacher DNS-Forwarder)

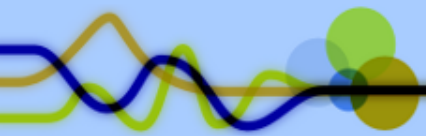
DNS-Proxy mit Bordmitteln (2)

- Setup: Client (VM), PC (Bridge)
- `pc# apt-get install dnsmasq`
- `pc# iptables ….`
- vm# host www.spiegel.de
- vm# host www.spiegel.de ….



DNS-Proxy mit Bordmitteln (3)

- → Keine Chance, außer
  - Verwendung eines anderen Ports (erfordert speziellen Nameserver)
  - VPN
  - TSIG



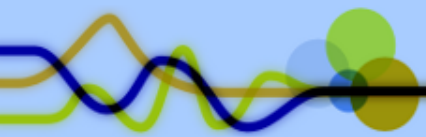
Bisherige Mechanismen: TSIG

TSIG: Transaction SIGnature

- Sicherstellung der Authentizität von Kommunikationspartnern durch **Shared Secret** (Prüfsumme über Daten und Zeit)
- Verwendung hauptsächlich für
  - Server2Server (Zonentransfer)
  - Client2Server (Dynamische Updates)
- Nachteil:
  - Skaliert sehr schlecht!!!

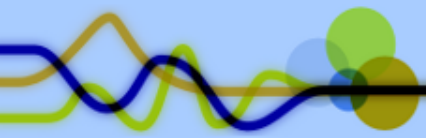
TSIG - Beispiel

- Shared Secret generieren:
  ```
  # dnssec-keygen -a hmac-md5 -b
  128 -n HOST ...
  ```
- Auf beteiligten Rechnern verteilen
- Query signieren:
  ```
  # dig -k K*private...
  ```



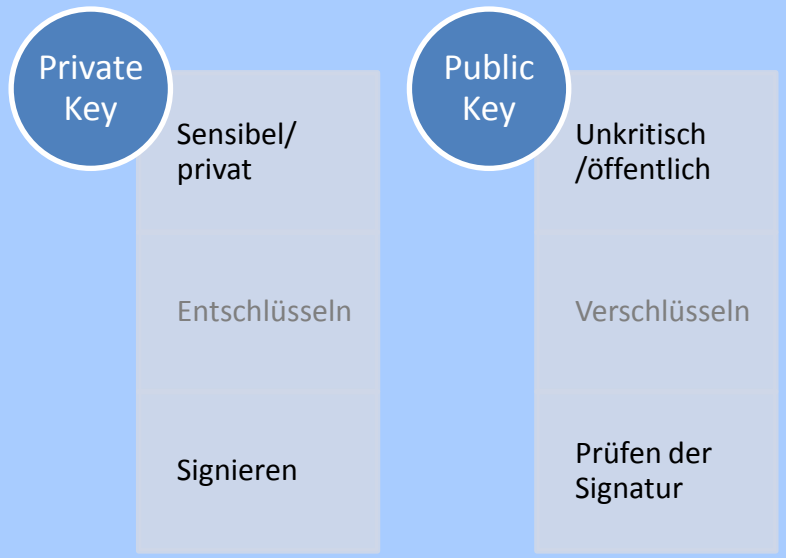
DNSSEEC

Public-Key-Kryptografie



Public Key Kryptografie

- DNSSec nutzt nur Signierung (Sicherung der Authentizität)!



Vorteile

- Einfaches Keymanagement
- Gute Skalierung

Probleme

- Vertrauenswürdigkeit des Public Key!!!

DNSSec – Einfach Abfrage

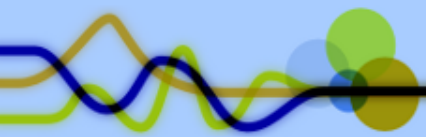
Abfragen einer bereits eingerichteten Zone

- `# dig +dnssec a.xxx.schlittermann.de @pu.schlittermann.de`
- Was sehen wir?
  - Eigentliche Antwort (A-Record)
  - Neuen Record: RRSIG
  - DO-Flag
  - EDNS

DNSSec – Neue Records

Folgende neue Resource Records sind in DNSSec definiert:

- RRSIG – Resource Record Signature
- DNSKEY – Public Key einer Signatur
- NSEC/NSEC3 – Next Secure
- DS – Domain Signer
- (DLV – Domain Lookaside Validation)



DNSSec – Validierung

- Bisher nur die Abfrage der Daten + (neue) Signatur
- Für Validierung muss der passende Public Key verfügbar sein
- → ebenfalls über DNSSEC
- `# dig +dnssec -t DNSKEY xxx.schlittermann.de @pu.schlittermann.de`

DNSSec – Validierung (sigchase)

- Dig kann die Validierung auch selbst vornehmen:
- `# dig +trusted-key=/dev/null +sigchase a.xxx.schlittermann.de @pu.schlittermann.de`
- → Validierung schlägt fehl, Authentizität des DNSKEY-Eintrages kann nicht überprüft werden!

DNSSec – Vollständige Validierung

- Bisher nur Signatur angezeigt und überprüft (in einer Zone!!!)
- Authentizität des öffentlichen Schlüssel nicht gesichert!
- Lösung:
  - Chain of Trust (über DS, schwierig in .DE)
  - Manuelle Prüfung (dig, bind)
  - Domain Lookaside Validation (dlv.isc.org)

Validierung durch lokalen Resolver

- Stub-Resolver kennt kein DNSSec!!!
- Bind9 kann auch also lokaler Resolver diene (ab 9.3)
- ```
  options { …
    dnssec-enable yes;
    dnssec-validation yes;
  };
  trusted-keys {
    xxx.schlittermann.de. 257 …
  };
  ```
- # dig +dnssec a.xxx.schlittermann.de
  ```

Validierung: Chain of Trust

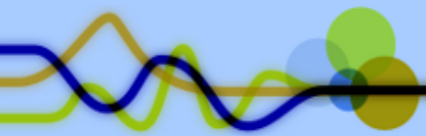
- Höchste Skalierbarkeit
- Hierarchische Idee des ursprünglichen DNS umgesetzt
- Parent-Zone stellt Authentizität über DS-Record (Domain Signer)
- Erforder komplette DNSSec-Umstellung, beginnen von der Top Level Domain (.DE)
- Wird hier nicht gezeigt!

Validierung: Domain Lookaside Validation (DLV)

- Stellt Mittelweg zwischen „Manueller Validierung“ und vollständiger Chain of Trust da
- Künstlicher Einstiegspunkt (muss nicht über TLD geschehen)
- Bind beherrscht diese Erweiterung!
- ```
  options {
    dnssec-lookaside . trust-anchor
  dlv.isc.org.;
  }
  trusted-keys {
    dlv.isc.org. 257 3 5 …
  };
  ```
  ```

Validierung: Domain Lookaside Validation (DLV)

- `# dig +dnssec soa cz. @localhost`
- Anfrage: DO-Flag gesetzt
- Antwort: AD-Flag gesetzt
- tcpdump → Type 32769 (DLV)



Validierung: Domain Lookaside Validation (DLV)

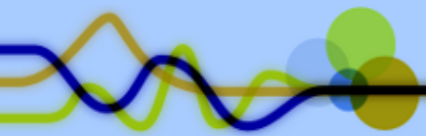
1. Account bei dlv.isc.org beantragen
2. Domain registrieren (sollte erreichbar sein, per TCP!)
3. DNSKEY hochladen
4. Angegebenen TXT Record für initiales Setup eintragen

→ fertig

Validierung: Typische Antworten

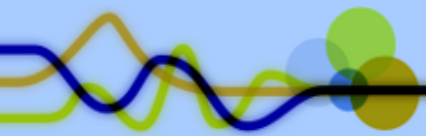
Im Kontext von DNSSec sind folgende Antworten möglich:

- NOERROR – Validierung erfolgreich
- NXDOMAIN – Eintrag nicht vorhanden + NSEC
- SERVFAIL – Signatur ungültig
  → kaputte Signatur sieht wie „Host not found“ aus



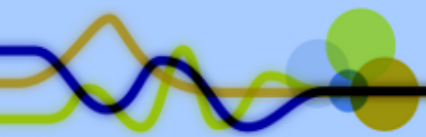
Was ist mit z.xxx.schlittermann.de?

- Signatur ist ungültig
- Adresse ist aber vorhanden!
- Man kann dig überreden (Checking Disabled)
- `# dig +dnssec +cdflag z.xxx.schlittermann.de`



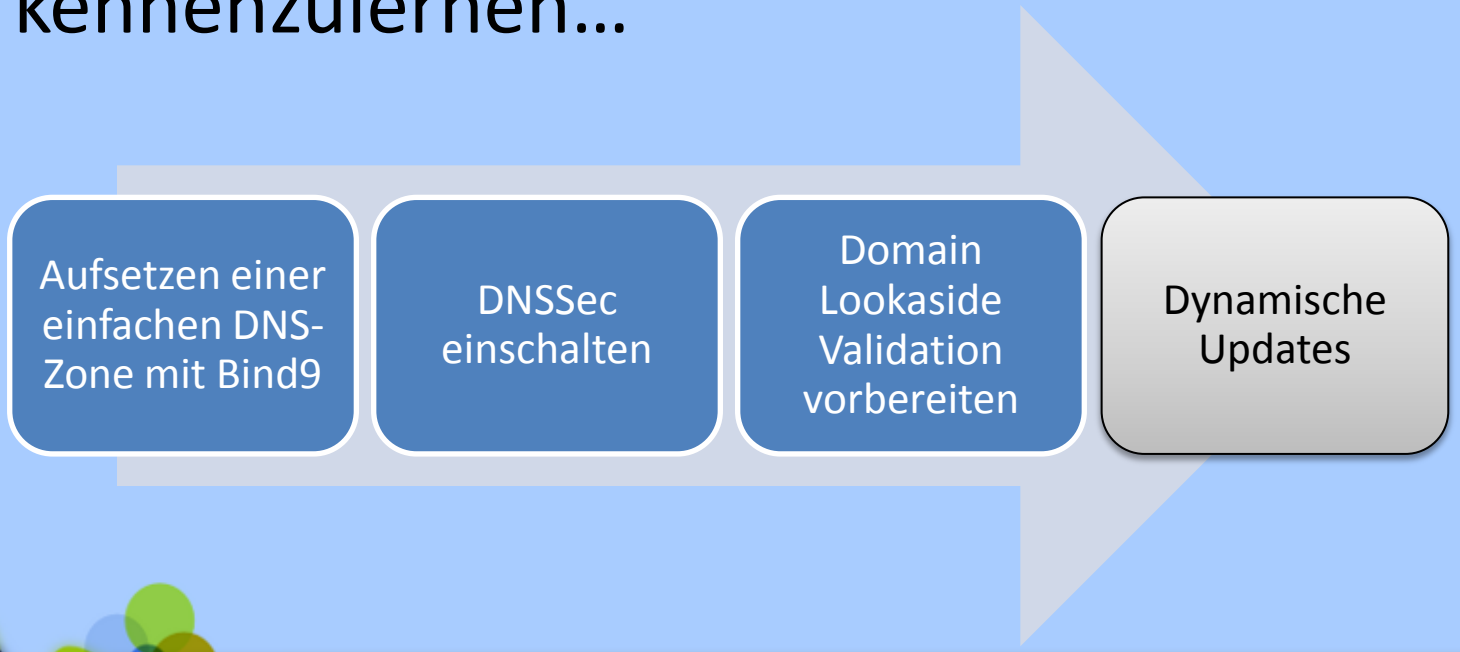
Unbound: alternativer Resolver

- Alternative zu Bind9 (als Resolver)
- DNSSec fähig
- Einfache Einrichtung
- Unterstützt DLV
- Schneller geht’s nicht!



Serverseite

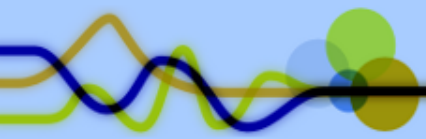
- Bisher haben wir uns nur an bestehender Infrastrukturen/Installationen versucht
- Es wird Zeit auch die andere Seite kennenzulernen…



Bind-Zone aufsetzen

Klassisch….

1. Lokalen Nameserver installieren
2. Leere/neue Zone anlegen (Forward reicht!)
3. Bind durchstarten
4. Testen



DNSSec vorbereiten: Keys erzeugen

1. Schlüsselpaar (ZSK & KSK) generieren

```
# dnssec-keygen -f KSK -a RSASHA1
   -b 1024 -n ZONE example.org.
# dnssec-keygen -a RSASHA1 -b 1024
   -n ZONE example.org.
```

2. Record für KSK & ZSK in Zone „example.org“ aufnehmen

```
# cat *key >> db.example.org
```

DNSSec vorbereiten: Zone signieren

1. Zoneendaten signieren

```
# dnssec-signzone -o example.org
   db.example.org
```

2. db.example.org.signed in Bind-Konfiguration eintragen!
3. Neusignieren bei jeder Änderung notwendig

```
# dnssec -signzone -o
  example.org -f
  db.example.org.signed.new
  db.example.org.signed
```

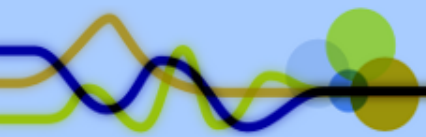
DNSSec vorbereiten: DLV Registration

DEMO….

DNSSec vorbereiten: Dynamische Updates

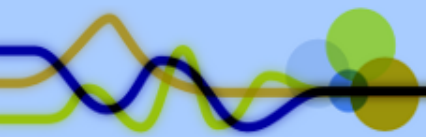
1. Bind kann auch „on-the-fly“ signieren
2. Private-Key muss vorhanden sein!
3. Dynamische Update für Zone erlauben:

```
zone … {
    allow-update { any; };
};
```



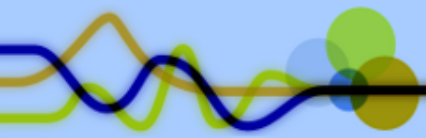
DNSSec vorbereiten: Dynamische Updates

- `# dig +dnssec dyn.example.org`
- Eintrag nicht vorhanden → NSEC
- ```
  # nsupdate
  update add dyn.example.org
  3600 IN 127.0.0.1
  send
  ```
- `# dig +dnssec dyn.example.org`


  ```

Weitere Tools

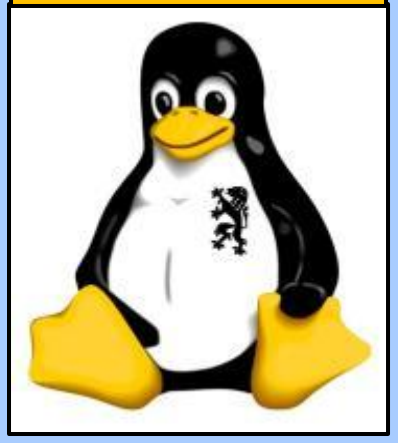
- Net::DNS (ist DNSSec fähig)
- DNSSEC-Tools



Danke!

- Fragen
- Vorschläge
- Ideen
- …

Linux-User-Group Dresden



schlittermann

internet & unix support
Heiko Schlittermann
Tannenstraße 2
D-01099 Dresden