DNSSEC... und dann?

Ray

4. April 2015

Übersicht

Übersicht

Übersicht

Übersicht

Übersicht

Übersicht

Abschnitt 1

Grundlagen

Warum DNSSEC

- Entwickelt zum Schutz vor DNS Spoofing
- Jetzt auch neue Recordtypen, die nur signiert sinnvoll sind

- Aber nicht um DNS Traffic zu verschlüsseln
- Auch nicht um sich gegen DoS zu schützen
- (eher das Gegenteil ist der Fall)

Was ist DNSSEC

- Hierarchisch signiertes DNS auf Basis von Public/Private Key Kryptographie
- Benötigt neue Recordtypen
- Ebenso Protokollerweiterungen (Flags, Queries)
- Offline-Signing tauglich (keine online Keys nötig) - es werden am Ende statische Records erzeugt die klassisch verteilt werden können
- In der Praxis aber meist dennoch einiges online

Vertrauen

- Streng Hierarchisch, eine Wurzel bei Root Domain "."
- Private Key der Root Zone in der Hand der ICANN, Signierung durch Verisign
- Trust Anchor im lokalen Resolver verweist auf passenden Public Key
- Inszenierte Key Signing Ceremony - ob's dem Vertrauen hilft?

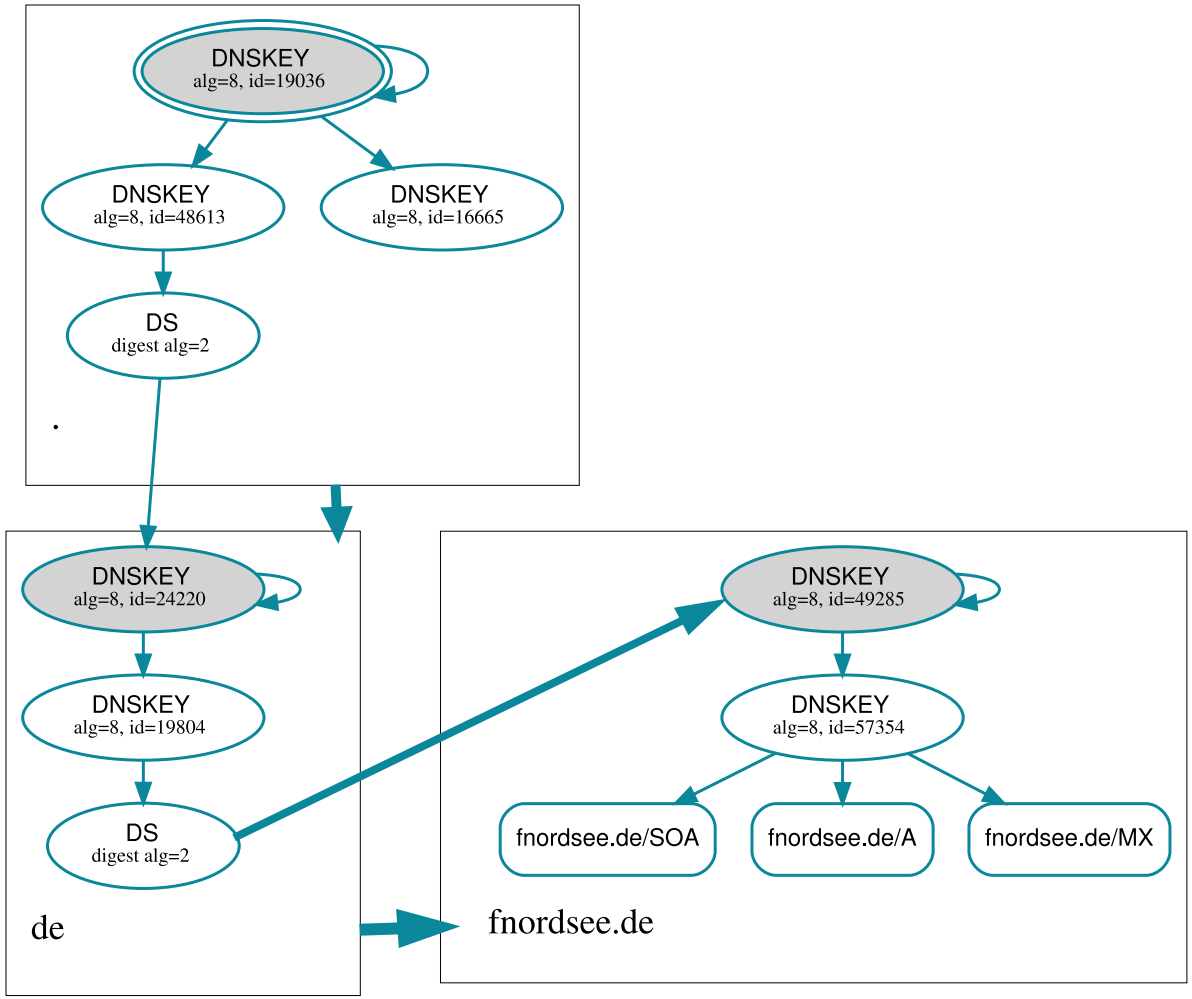
Delegation vom Vertrauen

- Keys der einzelnen TLDs in den Händen ihrer NICs, können nur Records unterhalb von sich signieren
- X.509 im Gegensatz dazu: unzählige CAs, die jeweils den gesamten Namensraum signieren können und auch noch Sub-CAs ausstellen dürfen
- Bei DNSSEC denkbar: Trust Anchor für bestimmte Subdomains eintragen, wenn man der Root Zone nur begrenzt traut
- Schlüsseltausch nach RFC5011 ermöglicht sogar automatische aktualisierung davon

Key Signing Keys/ Zone Signing Keys

- Technisch nicht notwendig, aber etablierte Praxis
- Key Signing Key mit grösserer Bitlänge (>=2048), der dazu dient den/die weiteren Keys der Zone zu signieren
- Kleinerer Zone Signing Key (>=1024) der alle weiteren Daten unterschreibt
- Die Elternzone verweist nur auf den Key Signing Key
- Key Signing Key wird nur selten benötigt, kann daher ggf. offline gelagert werden
- Durch kleineren ZSK insgesamt kleinere Zone / DNS Pakete
- ZSK kann getauscht werden, ohne Upstream Daten aktualisieren zu müssen
- Wichtig: bei Schlüsselaustausch sind zeitliche Abläufe genau zu beachten, um in allen cachenden Resolvern immer gültige Zonen zu haben! Stichwort Key Rollover

Struktur


DNSKEY
alg=8, id=19036
DNSKEY
alg=8, id=48613
DNSKEY
alg=8, id=16665
DS
digest alg=2
.
DNSKEY
alg=8, id=24220
DNSKEY
alg=8, id=19804
DS
digest alg=2
de
DNSKEY
alg=8, id=49285
DNSKEY
alg=8, id=57354
fnordsee.de/SOA
fnordsee.de/A
fnordsee.de/MX
fnordsee.de

Abschnitt 2

Technik

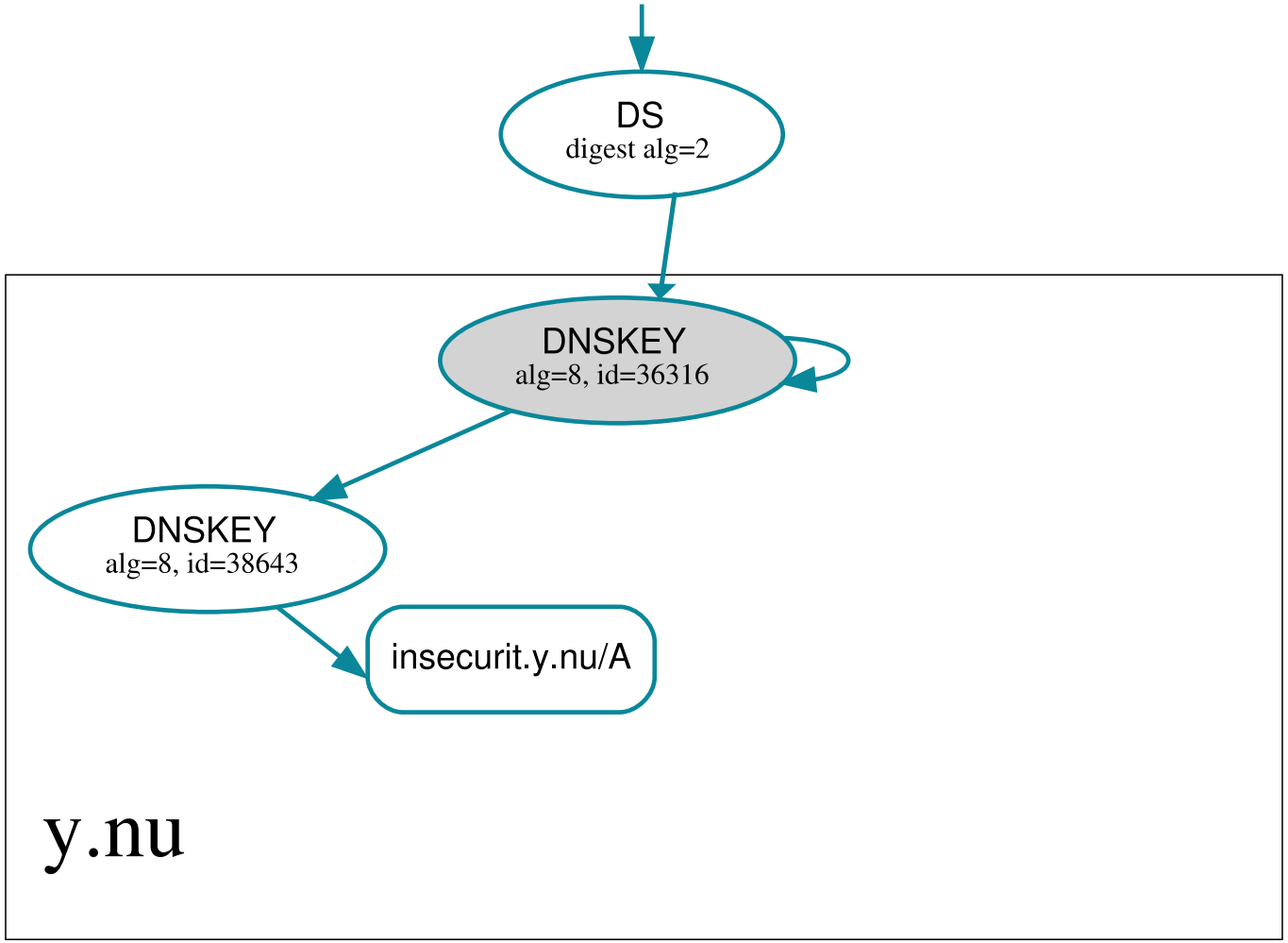
Neue Flags

- Benutzt EDNS0 (udp Antworten über 512 Bytes)
- DO: DNSSEC OK - signalisiert im Request, dass man DNSSEC versteht (dig +dnssec)
- CD: Checking disabled, Resolver soll nicht prüfen, sondern Daten weiter senden (dig +cdflag)
- AD: Authenticated Data: Resolver hat DNSSEC geprüft (bei Fehlschlag keine Antwort ohne AD, sondern SERVFAIL!)
- Achtung: DNS Pakete sind weiterhin nicht signiert oder ähnliches, man muss also sowohl seinem Resolver als auch dem Pfad dahin trauen können.

Neue Resource Records

- DS: Delegation Signer (Fingerprint eines Keys in der parent zone)
- DNSKEY: Public Keys (Flags + Kryptoverfahen + Keydaten)
- RRSIG: Signatur über ein RRSET (Sammlung von Records eines Types, z.B. alle MX-Records und v.a. alle DNSKEY-Records)
- NSEC: Next Secure (für denial of existence)
- NSEC3: NSEC gehasht

Signaturen einer Zone



DNSKEY/RRSIG

```
y.nu.     IN    DS     36316 8 2 2CF30B9C...BA713A
y.nu.     IN    RRSIG    DS 7 2 3600 20150417131642
                 20150403130502 4875 nu. WJ0....40=
.nu Zone                          |
----------------------------------|-----------------------
                                  v
y.nu.  IN    DNSKEY   257 3 8 AwEAAcDyQ...I4Xipi6tLWNv919
                               Hqb8WyBDioX/p+CpK+uBKzmsFF0=
y.nu.  IN    DNSKEY   256 3 8 AwEAAeVJM...ZTGRh7MnSPMcZ0R
                               IcxCw==
y.nu.  IN    RRSIG    DNSKEY 8 2 21600 20150424130636
                 20150403120636 36316 y.nu. KO/0u..91bchqpA==

securit.y.nu.   IN   A        172.16.23.42
securit.y.nu.   IN   RRSIG    A 8 3 21600 20150423153035
                 20150402150821 38643 y.nu. hI...zvUx
```

NSEC

- Signatur der nicht-existenz von Records
- Normalerweise NXDOMAIN error, das ist aber kein signierbarer Record
- Statt dessen: Bilden einer zyklischen verketteten Liste aller existierenden Records, und signieren dieser Liste
- Problem: dadurch Auslesen aller Records einer Domain möglich
- Nicht problematisch, wenn man ohnehin nur öffentliche Hosts in der Domain hat, z.B. www., mail. und ns. - dann ist NSEC durchaus eine sinnvolle Wahl

NSEC Beispiel

```
nasa.gov. IN  NSEC   3D-Printing.nasa.gov.
                     NS SOA MX TXT RRSIG NSEC DNSKEY
nasa.gov. IN  RRSIG NSEC 5 2 300 201504261  ...

3D-Printing.nasa.gov. IN NSEC _spf-ip4.nasa.gov.
                              A RRSIG NSEC
3D-Printing.nasa.gov. IN RRSIG NSEC 5 3 300 20150...

_spf-ip4.nasa.gov. IN NSEC _spf-ip6.nasa.gov.
                           NS RRSIG NSEC
_spf-ip4.nasa.gov. IN RRSIG NSEC 5 3 300 20150...

...
```

NSEC3

- U.a. auf Bestreben des DENIC entwickelt, um Datenschutz einzuhalten
- Konzept wie NSEC, aber es werden sortierte SHA1 Hashes der Hosts verkettet
- Gegen Brute Force Attacken Iterationen und SALT vorgesehen
- Etwas Verwirrend: um mit älteren Resolvern umzugehen, gibt es manche DNSKEY Algorithmen doppelt, einmal als NSEC3 getagged - zum Wechsel muss man daher auch den DNSKEY ersetzen
- Verfahren die neuer sind als NSEC3 gibt es hingegen nur in einer Fassung. RSASHA256 Keys z.B. können fuer NSEC und NSEC3 verwendet werden

NSEC3 Beispiel

```
P65U7HLLAG9ESBLLVH8HAIC6D138C19N.y.nu.  IN  NSEC3
  1  0  423  1337CAFE  PE404LSS31EVB8B2N4G9DJ19C359S9L2
  CNAME  RRSIG

PE404LSS31EVB8B2N4G9DJ19C359S9L2.y.nu.  IN  NSEC3
  1  0  423  1337CAFE  Q0G8N7R0O1A90C2BVAAI1HQDFMTQNO5T
  A  MX  RRSIG

Q0G8N7R0O1A90C2BVAAI1HQDFMTQNO5T.y.nu.  IN  NSEC3
  1  0  423  1337CAFE  QH30V19OUT9664KKOSSURT7MN9ELP83U
  A  MX  RRSIG
```

NSEC3 Probleme

- Hashing muss auf Resolvern passieren um Anfragen zu bearbeiten
- dadurch CPU-Exhaustion Denial of Service umso leichter, je mehr Iterationen man wählt
- Maximal zulässige Iterationen daher vom RFC begrenzt, abhängig von Bitlänge der eingesetzten Schlüssel (1024: 150, -2048: 500, mehr: 2500)
- Idee: Iterieren soll nicht mehr Rechenzeit benötigen, als die für RSA nötigen Berechnungen
- Ungleichgewicht: ernsthafte Brute-Force-Angreifer auf die Hashes haben SHA1-Hardwarebeschleunigung, Nameserver nicht
- Immerhin: Hash immer über Full Qualified Namen, kein Bilden von Rainbowtables für mehrere Domains möglich

Abschnitt 3

Verwendung

Als Client

- Lokalen Resolver installieren, z.B. unbound, validierung enablen und root trust-anchor hinterlegen (oft schon Default).
- resolv.conf anpassen:

```
nameserver 127.0.0.1
options edns0 (nicht zwingend)
```

- keine sonstigen NSe in resolv.conf!
- chattr +i /etc/resolv.conf (sicher ist sicher...)
- Tests:

```
dig +short test.dnssec-or-not.net TXT | tail -1
http://dnssec.vs.uni-due.de/
```

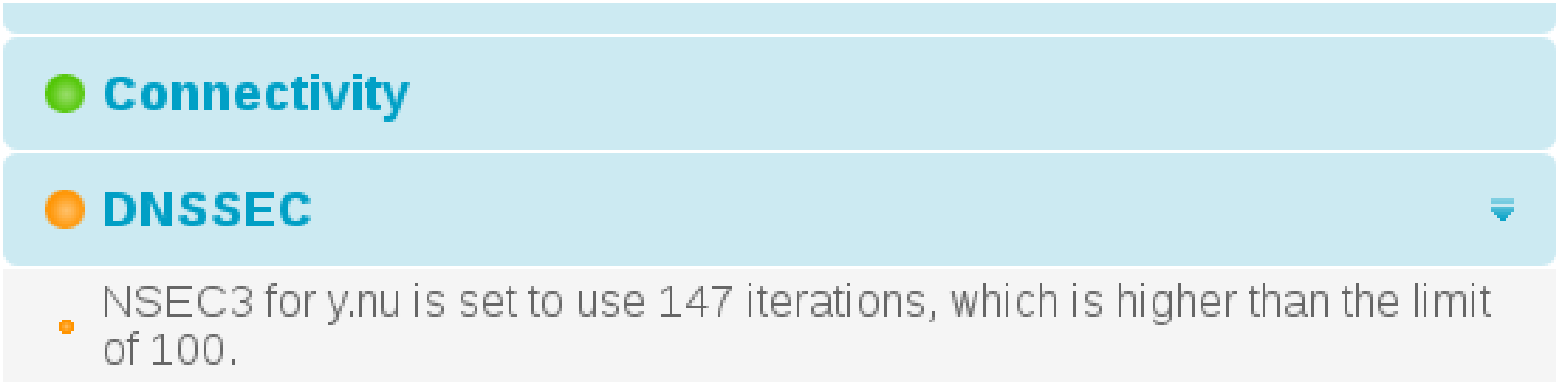
Als Zonenadmin

- NSEC3 fähige Zonenserver, möglichst Hidden Primary für Signaturerstellung (oder echtes offline signieren)
- Dokus lesen, Schlüsselkonzept planen
- Bind Inline Signing: leicht aufgesetzt, mehr Adminarbeit im Betrieb und bei Key-Rollover
- OpenDNSSEC hilft dabei, aber mehr Initialaufwand
- Wenn alles in Betrieb: DS Record über Registrar veröffentlichen lassen
- Wer mal testen will: vergebe gerne y.nu subdelegationen zum Spielen, siehe Workshop nachher

Abschnitt 4

Fun and Profit

Verifier FAIL



HBO FAIL

HBO's DNSSEC Issues

Posted by Peter Hagopian on March 12, 2015 in DNSSEC News

No doubt many of you heard HBO's recent announcement that they will be launching a streaming service in the near future. And as Comcast customers tried to visit HBO's site at order.hbonow.com, many were understandably confused and frustrated when the site wouldn't load for them properly.

Rest assured that this was not a conspiracy by Comcast to block access to the site. In fact, the HBO team had simply misconfigured DNSSEC on the order.hbonow.com site, making it appear that the site was invalid.

Kenya FAIL

[dns-operations] DNSSEC validation failures for .KE

Wouter Wijngaards just alerted me to validation failures for .KE (Kenya). I tried to call KENIC, but their phone numbers are all unreachable.

If anyone has local contacts in Kenya or nearby, please alert them!

http://dnsviz.net/d/ke/VRp4ag/dnssec/

Their current DS record points to a key that has the revoke bit set, but it is no longer signing the DNSKEY rrset.

NSEC Walking

- Standardtool ldns-walk aus Paket ldns-utils
- in vielen Zonen möglich
- „While trying to fix one problem (DNS spoofing) we introduced another”

```
$ ldns-walk nasa.gov | wc -l
1127
```

NSEC3 Walking

- Hackertool nsec3walker von dnscurve.org (surprise!)
- Verfahren: zufällige Hosts anfragen, bis man zwischen möglichst alle Hashes der verketteten Liste getroffen hat
- Dann offline analog zu Passwortknackern Wörter und Wortvariationen hashen und vergleichen
- Praxistauglichkeit: bedingt. Sehr ineffiziente Implementierung, schlechte Wortvariationen, v.a. nicht DNS spezifisch
- dennoch:

```
...
qpq1rfhth67h7au8qnk4qtfvkjp8jptl dodimages.mil.
found 190 private NSEC3 names (81%)
using 6.202.622.882 hash computations
```

NSEC3 Walking at home

- Wer mal selber unhashen will: http://insecurit.y.nu/dnssec/
- Hashes von 98% .de, 100% .fbi.gov, 99% .gov, .nl scannt noch
- Allerdings: teilweise nur die signierten Anteile, da NSEC3 ein opt-out Feature bietet
- .de: 290k Domains, .nl 2.4Mio Domains mit DNSSEC
- Gefundene Subdomains können durchaus dann trivial mit NSEC walkbar sein, vgl. nasa.gov
- Dank DNSSEC: endlich die maschinenlesbare Regierung!

Abschnitt 5

Neue Records

SSHFP

- ssh Fingerprints von Servern im DNS hinterlegen
- RFC4255 (SSHFP), RFC6594 (Erweiterung DSA/ECDSA)

```
$ ssh-keygen -r hostname
  -f /etc/ssh/ ssh_host_ecdsa_key.pub

hostname IN SSHFP 3 1 16c2...
hostname IN SSHFP 3 2 526462fe0ef7...
```

- Erstes Feld Keyformat: RSA/DSA/ECDSA/...
- Zweites Feld Hashverfahren des Records: SHA1/SHA256

SSHFP

- schon seit sehr langem in OpenSSH enthalten
- zuerst natürlich wie beschrieben lokalen Resolver einrichten
- Parameter VerifyHostkeyDNS in ssh_config oder commandline:

```
$ ssh -o "VerifyHostKeyDNS yes" test@insecurit.y.nu
test@insecurit.y.nu's password:
```

- Keine Rückfrage mehr, wenn Key passt und Resolver AD geliefert hat
- Alternative ask stat yes: Gibt Prüfergebnis an, manuelle Abfrage wie gewohnt

DANE/TLSA

- RFC6698 "DNS-based Authentication of Named Entities"
- Definiert das hinterlegen von X509 Zertifikaten im DNS
- Beispiel:

```
_25._tcp.<servername>.     IN  TLSA  3  0  1  \
      8cb0fc6c527506f4f1...738d11468dd953d7d6a3021f1
```

- Felder nach TLSA: Cert Usage, Selector, Matching Type
- RFC7218 definiert Namen dafür, weil zu verwirrend

DANE/TLSA für SMTP

- SMTP Aktuell primärer Einsatzzweck von DANE
- Insgesamt noch eher geringe Verbreitung
- Sichert die Ad-Hoc Verschlüsselung für Mail-Transit
- Seit ca. einem Jahr von ersten Mailprovidern genutzt
- Natürlich aber nicht von E-Mail Made in Germany - würde ja Weltweit funktionieren...

DANE/TLSA für HTTP

- Nahezu keine Verbreitung
- Über Browser Plugins nutzbar
- Google hält nichts davon, bevorzugt es das existierende CA System zu patchen
- Generell natürlich auch für alle möglichen weiteren Protokolle mit TLS nutzbar, z.B. Patch für irssi vorhanden

OPENPGPKEY

- PGP Keyring einer E-Mail im DNS
- IETF Draft, noch nicht final festgeschrieben, aber wohl grob stabil
- localpart der E-Mail SHA224 gehasht, zur verschleierung (echo -n ray | openssl dgst -sha224)

  90ffc2300bfbe8fbdddb57bc85db44fd0217b079b14e72e9ac98227._openpgpkey.y.nu. IN OPENPGPKEY mQFNBFMxZ4YBCgDVfzPNAS394VXUb8mC34gs5RZv...

- Kompletter Keyring mit Signaturen - also ggf. weitere Prüfung möglich
- auf jeden Fall authentifizierter als aktuelle Keyserver, zurückziehen wesentlich einfacher

SMIMEA

- Keys für S/MIME Mails im DNS
- IETF Draft, potentiell noch Veränderungen, Resource Record ID noch nicht zugeordnet
- Im Prinzip sehr analog zu TLSA Records
- Vorteile wie bei TLSA: nur eine Hierarchie, nicht CAs aus allen Ländern der Welt können für deutsche Domains signieren

Abschnitt 6

Abschluss

Links

- Client Test: http://dnssec.vs.uni-due.de/
- DNSSEC Check / Visualizer: http://dnsviz.net
- Mehr Links auf: http://insecurit.y.nu/dnssec/ (HowTos, DANE Validator, Videos, ...)

RFCs

- RFC4641 DNSSEC Operational Practices, Version 2
- RFC4034 DNSSEC Resource Records, RFC4035 Protocol
- RFC7218 DANE, RFC6394 Use Cases/Requirements
- RFC5011 Auto Trust Anchor Update

Workshop

- Um ca. 16:30 Workshopraum 3 (Gebäude Nebenan, 1. Stock im Gang)
- wer spielen will: irgendwo (rootserver o.ä.) BIND =>9.9 installieren und Zone-Server mit inline Signing bauen
- Mail mit DS Record für Wunsch-Subdomain unter y.nu an dnssec@y.nu - Bitte „Workshop” im Subject
- (auch für nicht-Teilnehmer des Workshops nach der Hegg machbar)

Fragen

- Kurze Fragen jetzt - ausführlicheres im Workshop
- Vielen Dank für die Aufmerksamkeit!