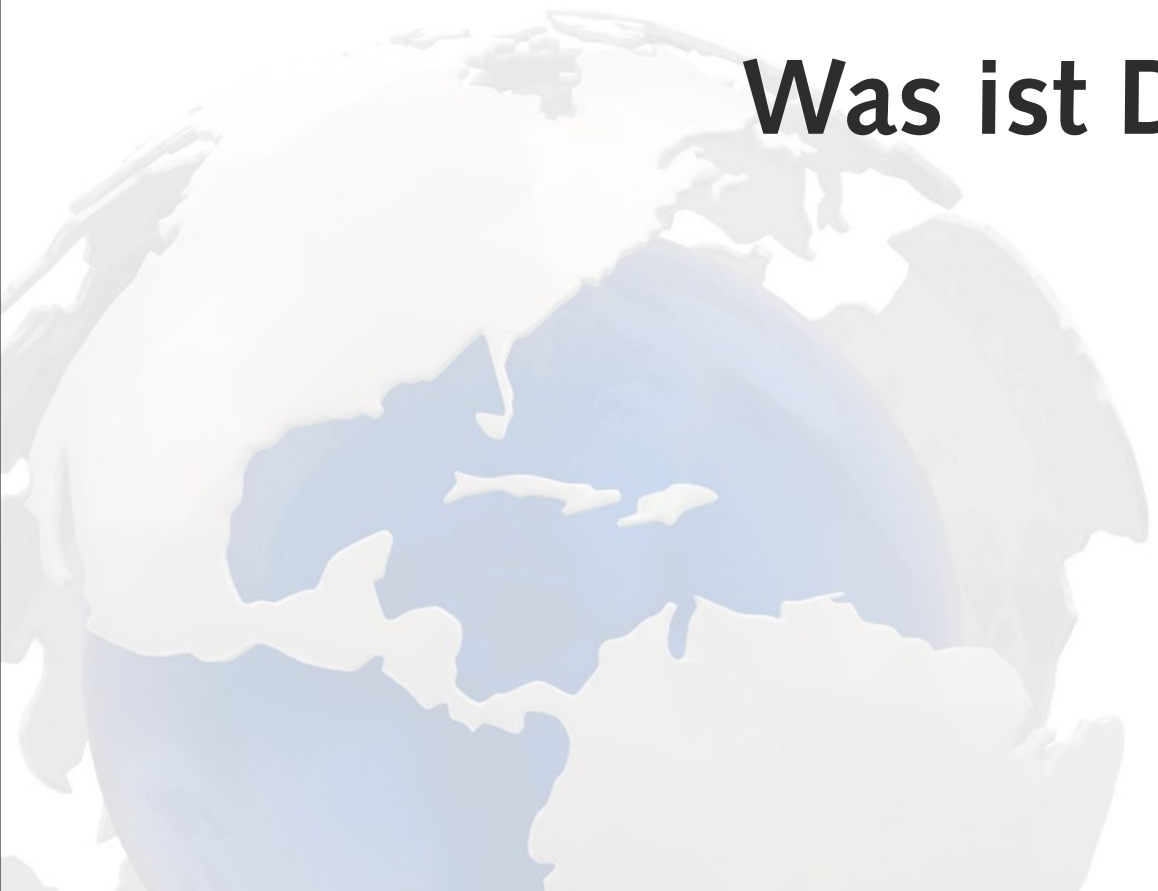
DNSSEC
Ein Erfahrungsbericht

Florian Obser – Hostserver GmbH

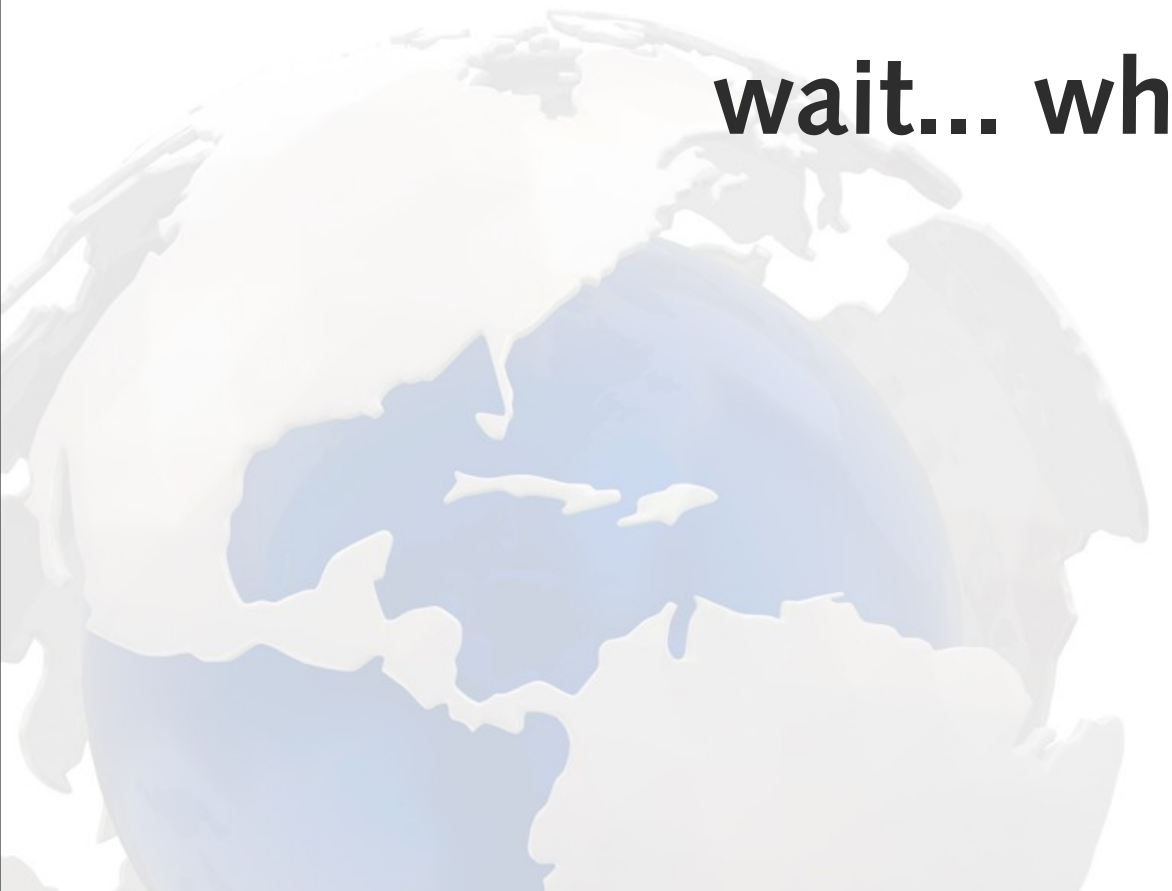
support@hostserver.de

Was ist DNSSEC?



Was ist DNS?

http/1.1



wait... what?!?


I TRY TO MAKE EVERYONE'S DAY A LITTLE MORE SURREAL.

Wer bin ich?

Jack of all trades

Layer 2 – 8 Support

Hostserver GmbH

Programmierer

Perl, Java, PHP, C, Erlang, Python, Smalltalk, Lisp...

Sysadmin

Linux / OpenBSD / ...

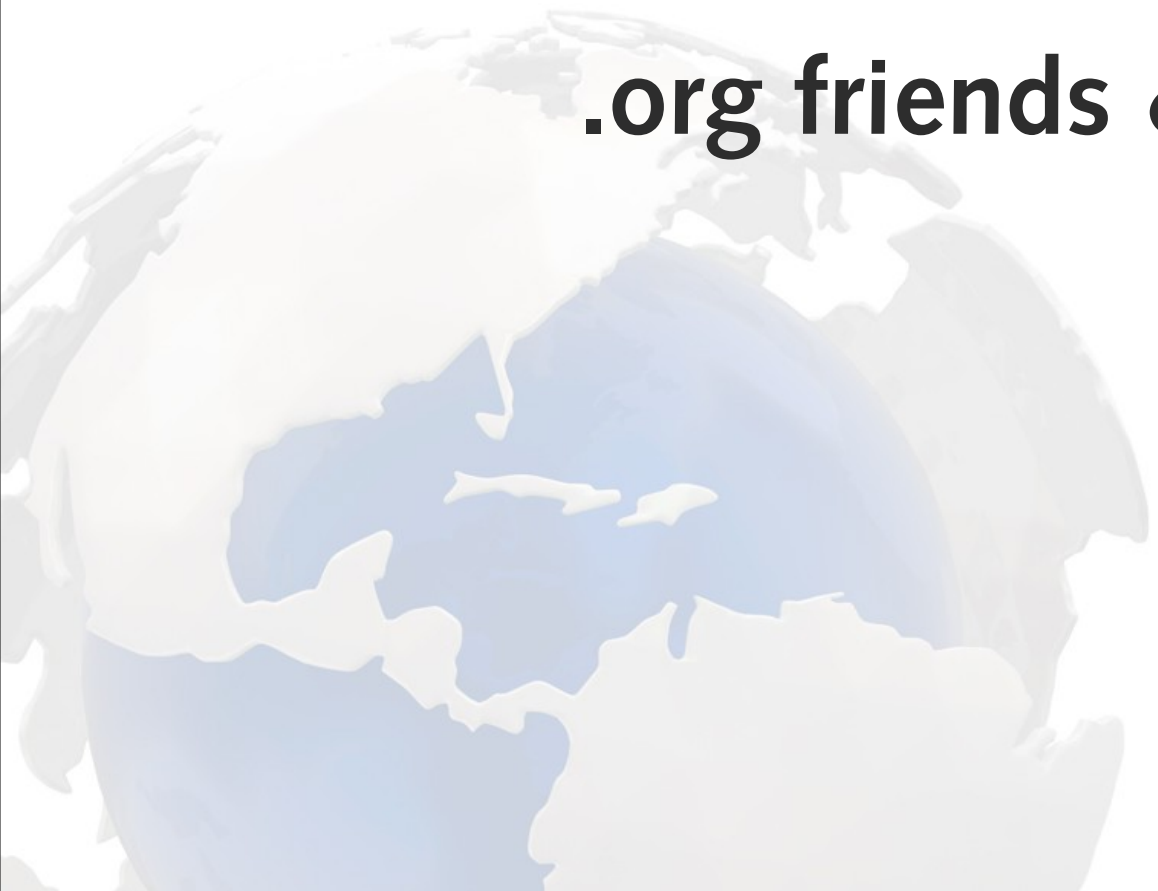
Once upon a time...
(2009-12-07)

IPv6

Testdomain

ipv6only.$TLD

$TLD?



.org friends & family

Guter Plan ™

ipv6only.org signieren

```
$ /opt/bind-9.6.1-p2/sbin/dnssec-signzone ipv6only.org.zone
```

FÍN

$bignum Zonen in pdns DB

Webinterface

ZSK online

automatisches resign

KSK „offline“

Support für ZSK/KSK roll

```
$ script.pl < pdns_db | dnssec-signzone > nsd
```

piggyback pdns-db?

```
mysql> describe records;
+-------------+--------------+------+-----+---------+
| Field       | Type         | Null | Key | Default |
+-------------+--------------+------+-----+---------+
| id          | int(11)      | NO   | PRI | NULL    |
| domain_id   | int(11)      | YES  | MUL | NULL    |
| name        | varchar(255) | YES  | MUL | NULL    |
| type        | varchar(6)   | YES  |     | NULL    |
| content     | varchar(300) | YES  |     | NULL    |
| ttl         | int(11)      | YES  |     | NULL    |
| prio        | int(11)      | YES  |     | NULL    |
| change_date | int(11)      | YES  |     | NULL    |
+-------------+--------------+------+-----+---------+
```

Constraints?!?

RETARDS

We all know one.

DNSSEC lernen
+
Postgres stored procedures

RFCs lesen

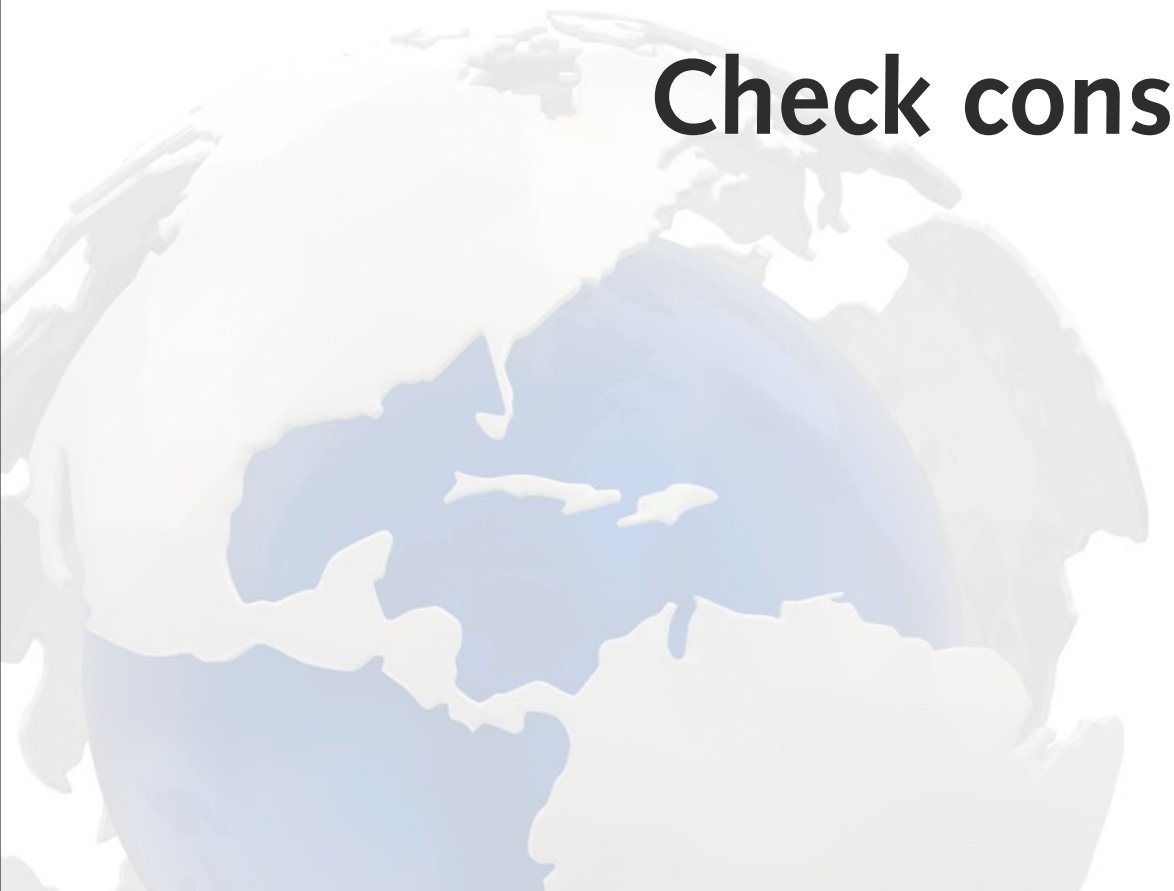
clortho

Vinz Clortho - Keymaster of Gozer

DB Schema

Tabelle für jeden RR

Spalten passend zum RR



Check constraints

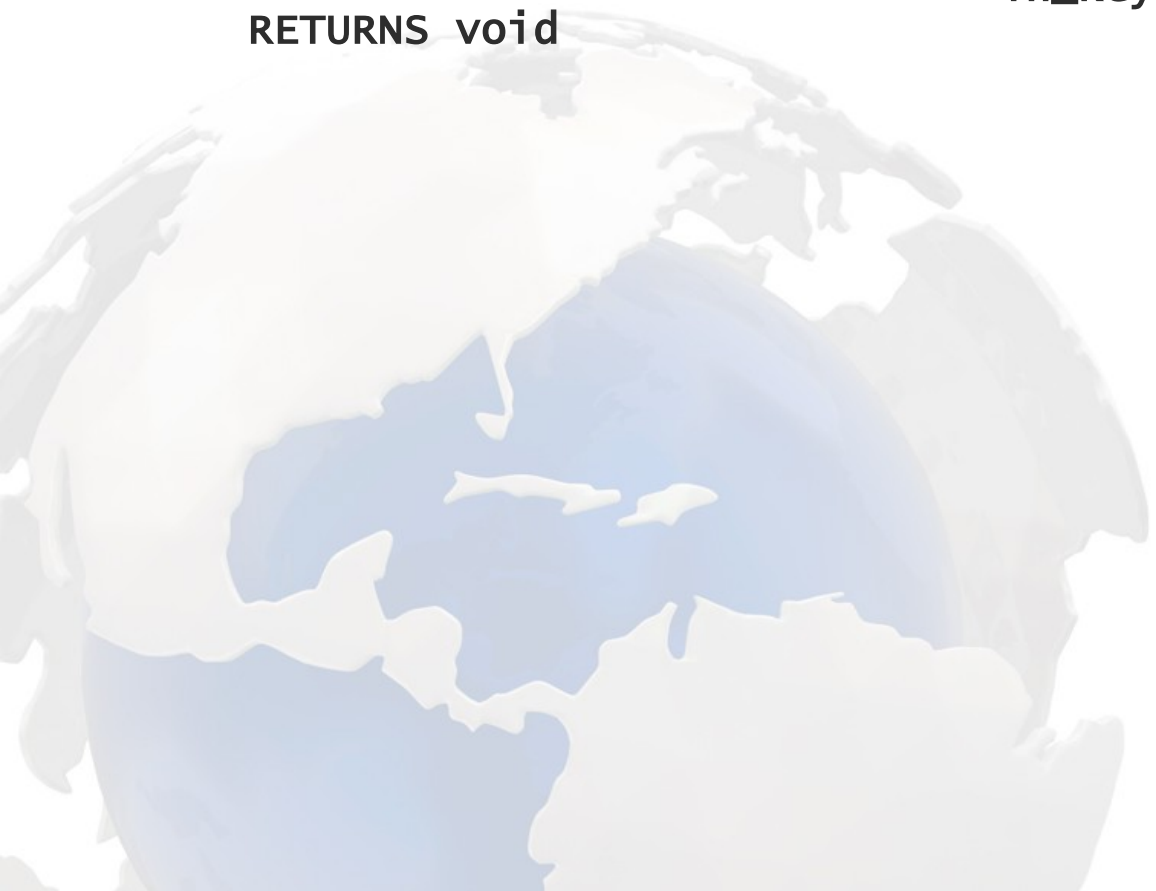
Obskure Features

auto reverse Zone

Bind-Zone-File-View

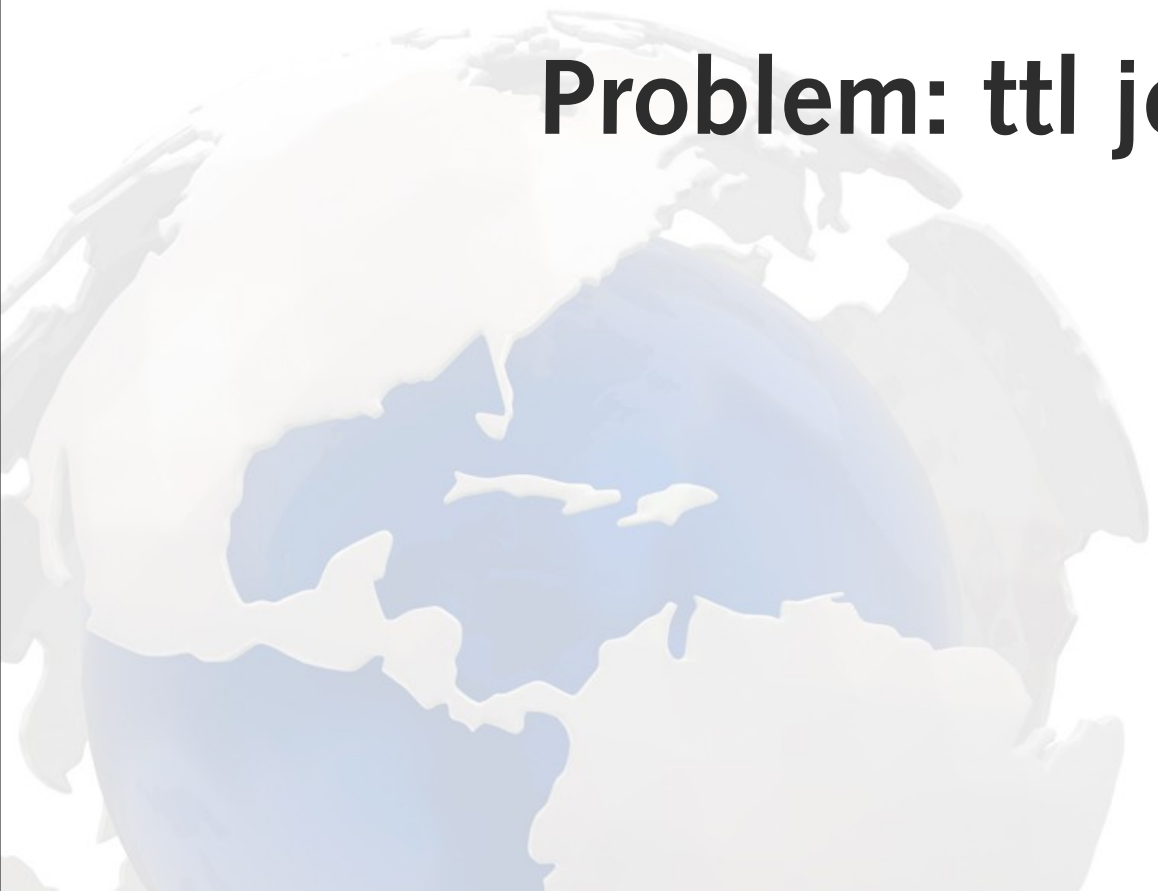
ZSK / KSK Roll FSM

```
zsk_roll_to_pre_publish_new_key(in_zone character varying,
                                in_flags integer,
                                in_algorithm integer,
                                in_publickey text,
                                in_keyid integer)
    RETURNS void
```



Delete / insert / update Trigger auf RR Tabellen

berechnet max expire Datum



Problem: ttl jetzt 3600

aber vor 10 min 86000

→

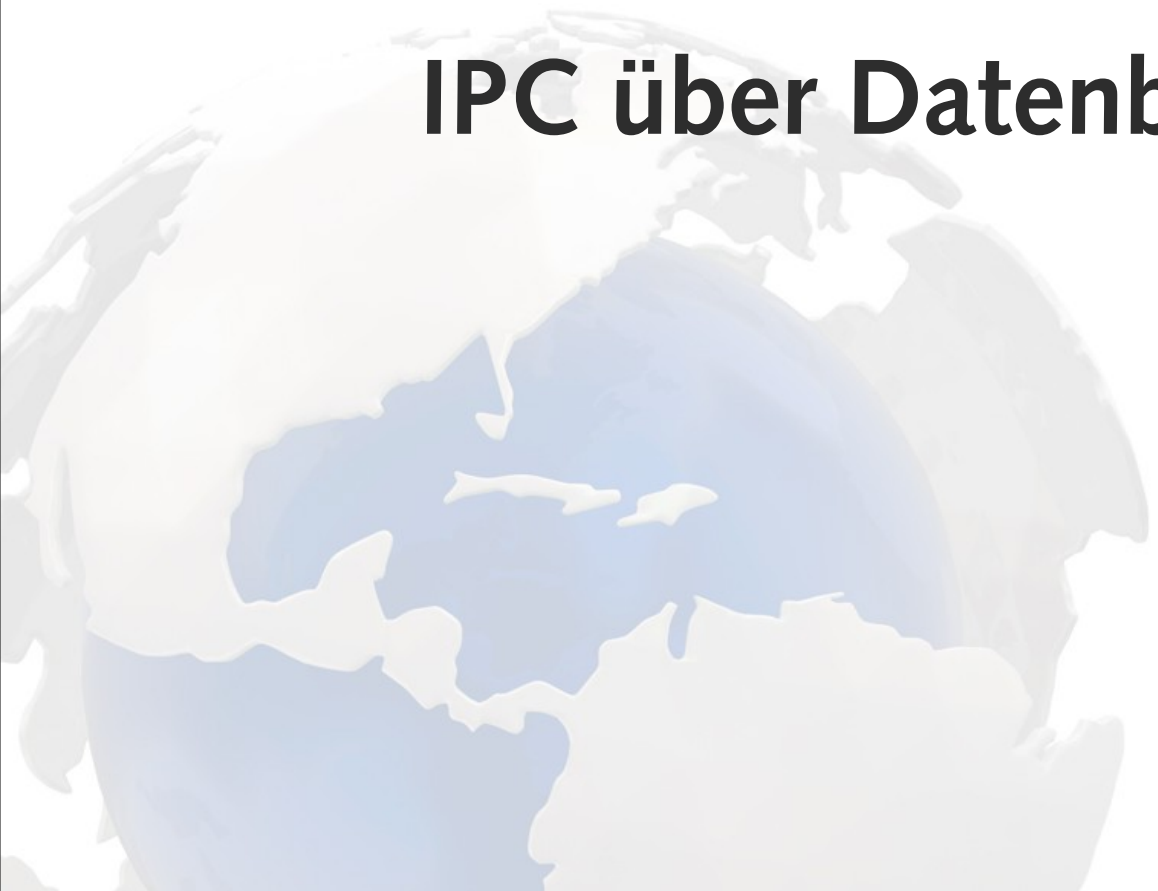
> 3600 sec. warten
bei Keyroll

CLI

```
$ /usr/local/bin/clortho --help
clortho init ZONE
clortho zsk-sign ZONE
clortho push ZONE
clortho reload ZONE
clortho sign-reload ZONE
clortho show-zsk-roll-state ZONE
clortho show-next-zsk-roll-state ZONE
clortho zsk-roll ZONE
clortho show-ksk-roll-state ZONE
clortho show-next-ksk-roll-state ZONE
clortho ksk-roll ZONE

    init      Initializes zone for dnssec.
[...]
```

2 Unixaccounts



IPC über Datenbank / rsync

keymaster

erzeugt KSK / ZSK Keys

(init, {zsk,ksk}-roll)

signiert minimal Zone (KSK)

SOA + DNSKEYs

DNSKEY RRSIGs → DB

```
$ rsync zsk.key zsk.private gatekeeper@...
```

gatekeeper

Signiert komplette Zone (ZSK)

zone.signed → master NS

Fehlendes Feature

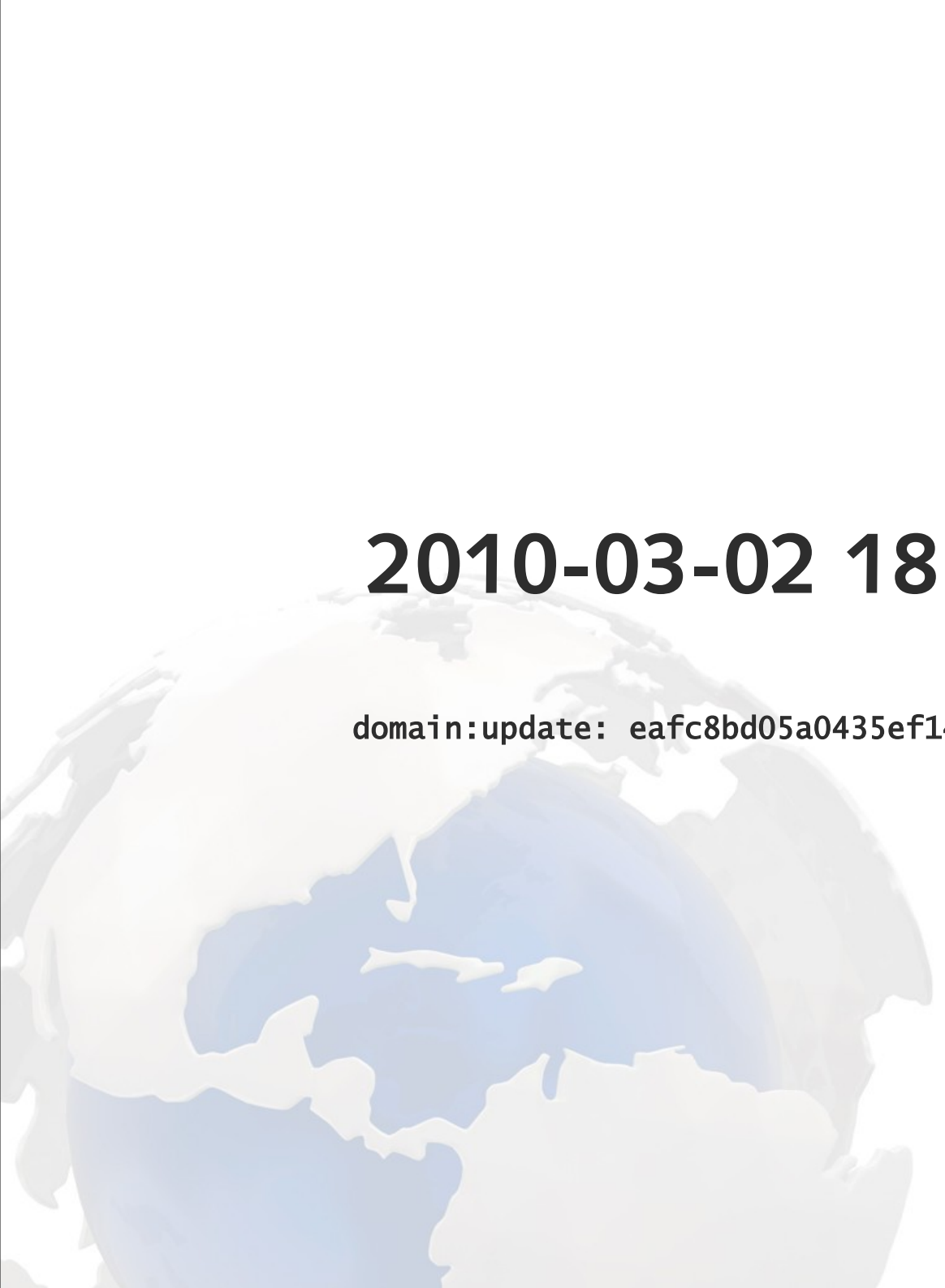
2 KSKs: FSM State: WAIT_FOR_EXTERNAL_EVENT

z.Z.: Webinterface klicken

Plan: Plugin für jede Registry

TCO

1 kaltes Winterwochenende



2010-03-02 18:05:44+00:

domain:update: eafc8bd05a0435ef14d31bbff29687dd7b71c3ca.de

2010-03-03 16:07:25+00:

flags: qr rd ra ad

Registry Backend: 5 min

Registry Webfrontend: 1-2h

2010-06-01 22:33:42+00:

KSK Roll


JOIN
THE
DARKSIDE
and get a free cookie...

[DJB] in 1999 pointed out that "the 16 bit transaction id in DNS... Yeah that's probably not enough"
[...]
The only reason I was able to get all the people in a room was because they have been dealing with Paul for a long time and when Paul says "Oh dear Lord, you remember that thing from DJB that we ignored? Uuuuh We probably shouldn't have done that."...

26C3: Dan Kaminsky –
Why were we so
vulnerable to the DNS vulnerablility?

„Breaking DNSSEC“

http://cr.yp.to/talks/2009.08.10/slides.pdf

(d)DOS amplifier

```
$ dig @a.nic.de -t ANY de
```

Ratio: 1:4

```
$ dig @auth-fra.dnssec.denic.de -t ANY de
```

Ratio: 1:25

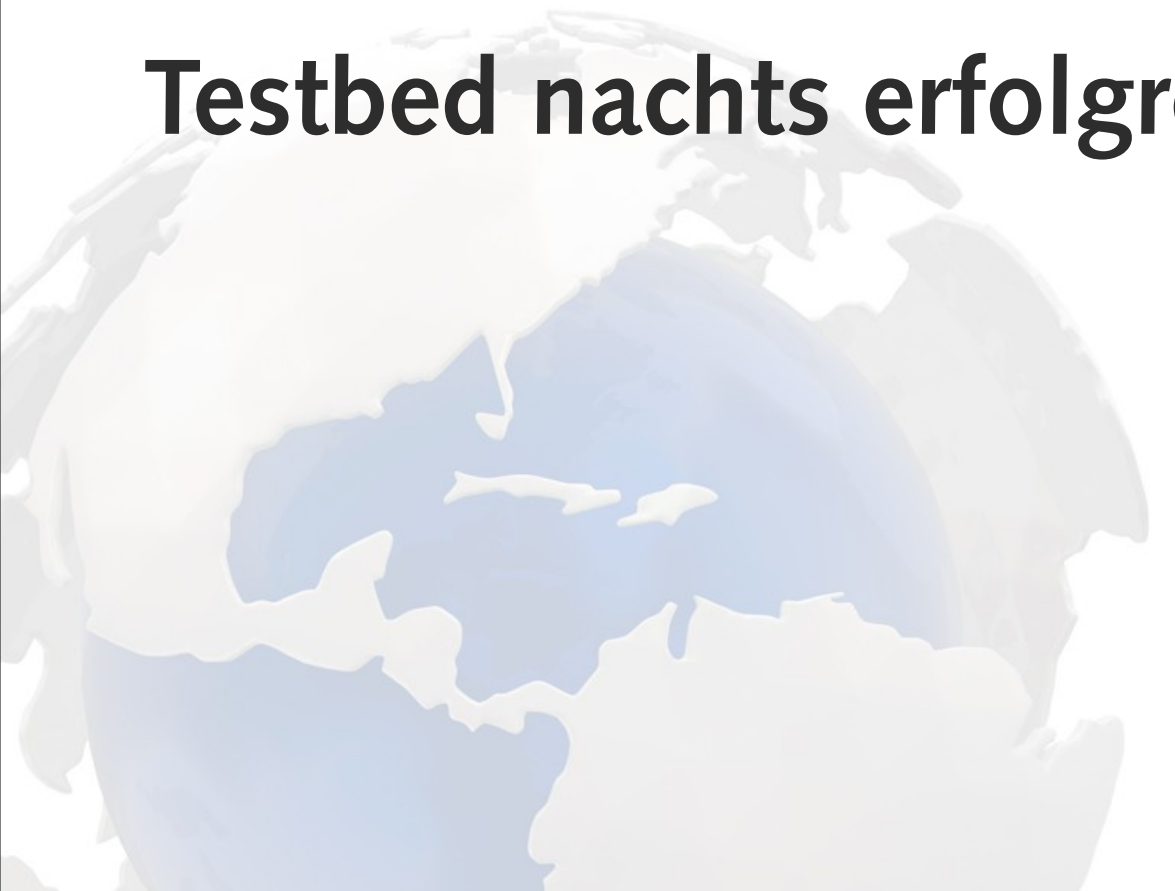
```
$ dig @a.ns.adns1.de +dnssec -t ANY
eafc8bd05a0435ef14d31bbff29687dd7b71c3ca.de
```

Ratio: 1:33

UDP

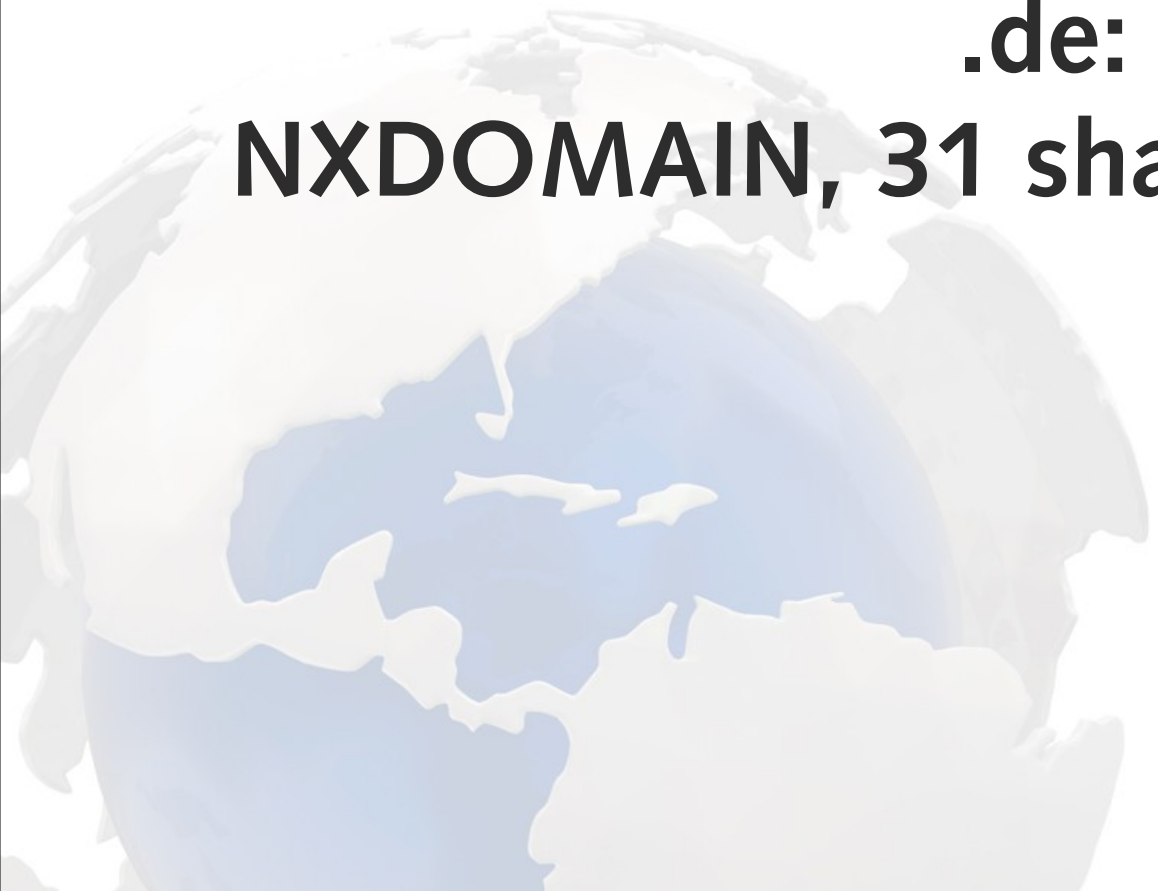
Wer hat sich in letzter Zeit unbeliebt gemacht?

Testbed nachts erfolgreich abgeschaltet



CPU-DOS

NSEC3



.de:
NXDOMAIN, 31 sha1 Operationen

→ Traffic DOS

Ideen, nicht ausprobiert

TCP?

window size = 0

oder RST schicken

States / Loadbalancer Probleme?

UDP

icmp port unreachable schicken?

edns: max. Antwort = 1 Byte (funktioniert nicht, min: 512)

non-issue

nsec Zonewalking

dig +dnssec barbaz.cz

barbatana.cz. 900 IN NSEC barbazar.cz. NS RRSIG NSEC

(barbatana.cz < barbaz.cz < barbazar.cz)

dig +dnssec barbaz.de

```
NSECF17OQCFVGUD9CQ7KCSUBFQSC7QI9D2CA.de. 7200
        IN NSEC3 1 1 31 DE15C001 F17QOL0QU5KECQUHT2JEMVJKQ9BSCG1T
```

Uff!

hm...

/etc/shadow, anyone?

Schritt 1: alle NSEC3 records finden

Disclaimer:

folgende NSEC3 Erklärung nicht vollständig, beinhaltet nur Teile die wir für Bruteforce brauchen

HASH1.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH2

HASH1.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH2

foo.domain.tld: NXDOMAIN iff

HASH1 < nsec3_hash(foo.domain.tld, $ITER, $SALT) < HASH2

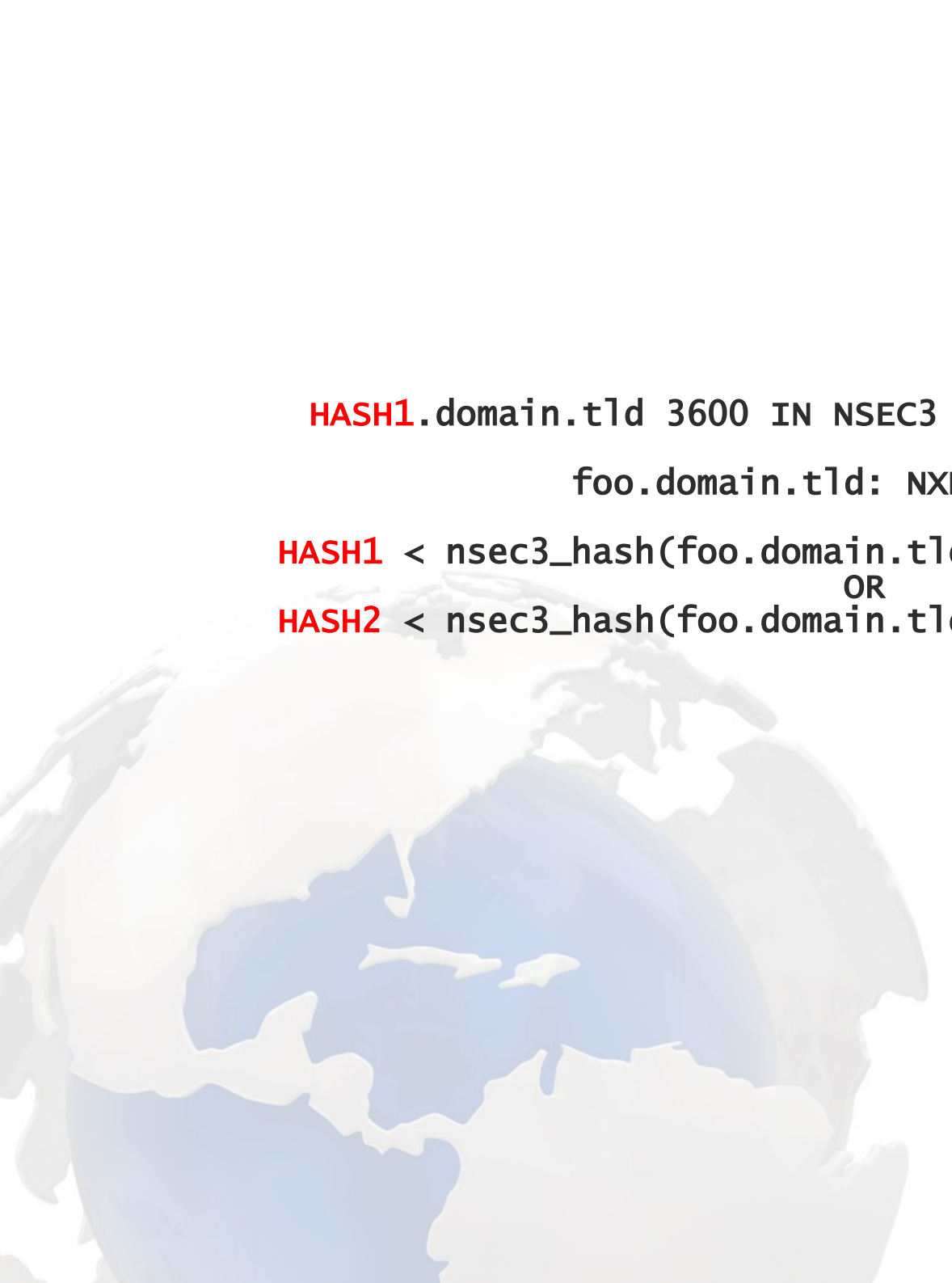
HASH1.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH2

foo.domain.tld: NXDOMAIN iff

HASH1 < nsec3_hash(foo.domain.tld, $ITER, $SALT) < HASH2

OR

HASH2 < nsec3_hash(foo.domain.tld, $ITER, $SALT) < HASH1



HASH1 < nsec3_hash(foo.domain.tld, $ITER, $SALT) < HASH2

```
while(true) {
    $random.domain.tld erzeugen
    nsec3_hash($random.domain.tld, $ITER, $SALT)
```

NSEC3 RR bekannt? → next

Nein:
DNS fragen,
NSEC3 RRs merken → next

Perl + Net::DNS::Sec

NSEC3 RRs speichern / abfragen

HASH1 < nsec3_hash($random.domain.tld, $ITER, $SALT) < HASH2

```
SELECT 1 FROM nsec3_rr
         WHERE hash1 < ? AND ? < hash2
         LIMIT 1;
```

```
CREATE TABLE nsec3_rr (
  hash1 char(32) NOT NULL,
  hash2 char(32) NOT NULL,
  PRIMARY KEY  (hash1,hash2))
```

cow-orker: WTF?!?

Ah! Deshalb dieses unintuitive Gefühl...

.de zone

(ungetuntes) mysql stirbt bei 200k rows

sqlite bei 300k rows

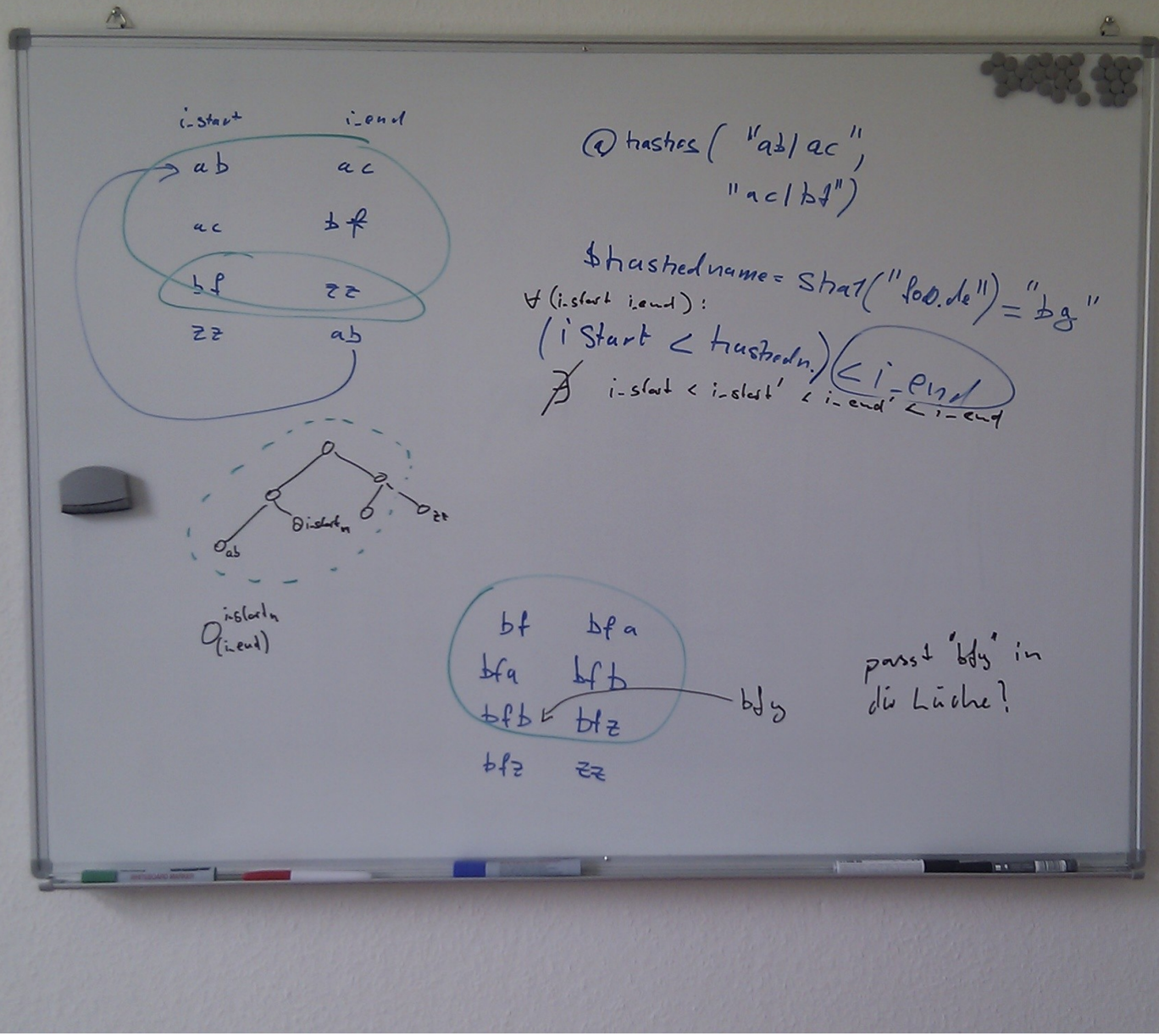
stirbt == 1 Query > 1sec

cow-orker: Du willst da einen Suchbaum

aber das geht nicht!1!11

Ich brauche: a < $x < b

Suchbaum: a < $x


i_start
i_end
ab
ac
ac
bf
bf
zz
zz
ab
@hashes("ab|ac",
"ac|bf")
$hashedname = sha1("foo.de") = "bg"
∀(i_start i_end):
(iStart < hashedn.) <i_end
i_start < i_start' < i_end' < i_end
bf
bfa
bfb
bfz
zz
bfy
passt 'bfy' in die Lücke?

HASH1.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH2

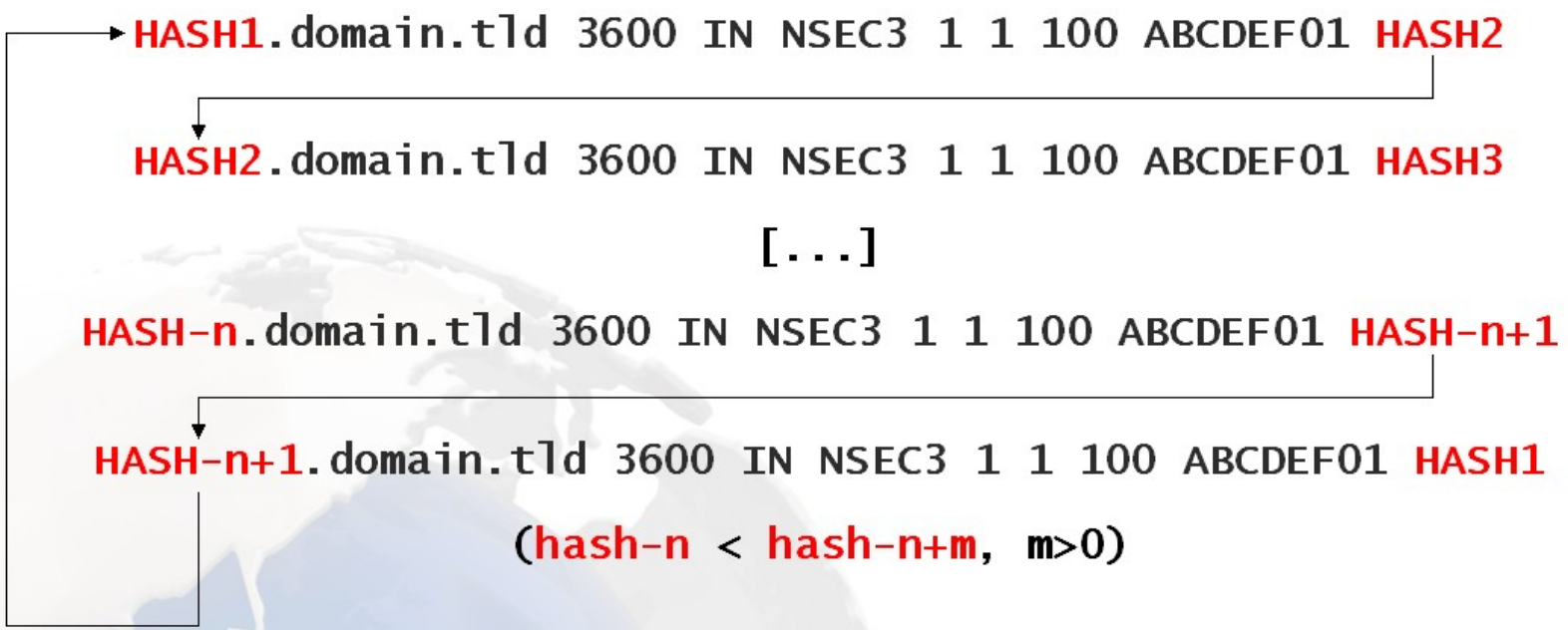
HASH2.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH3

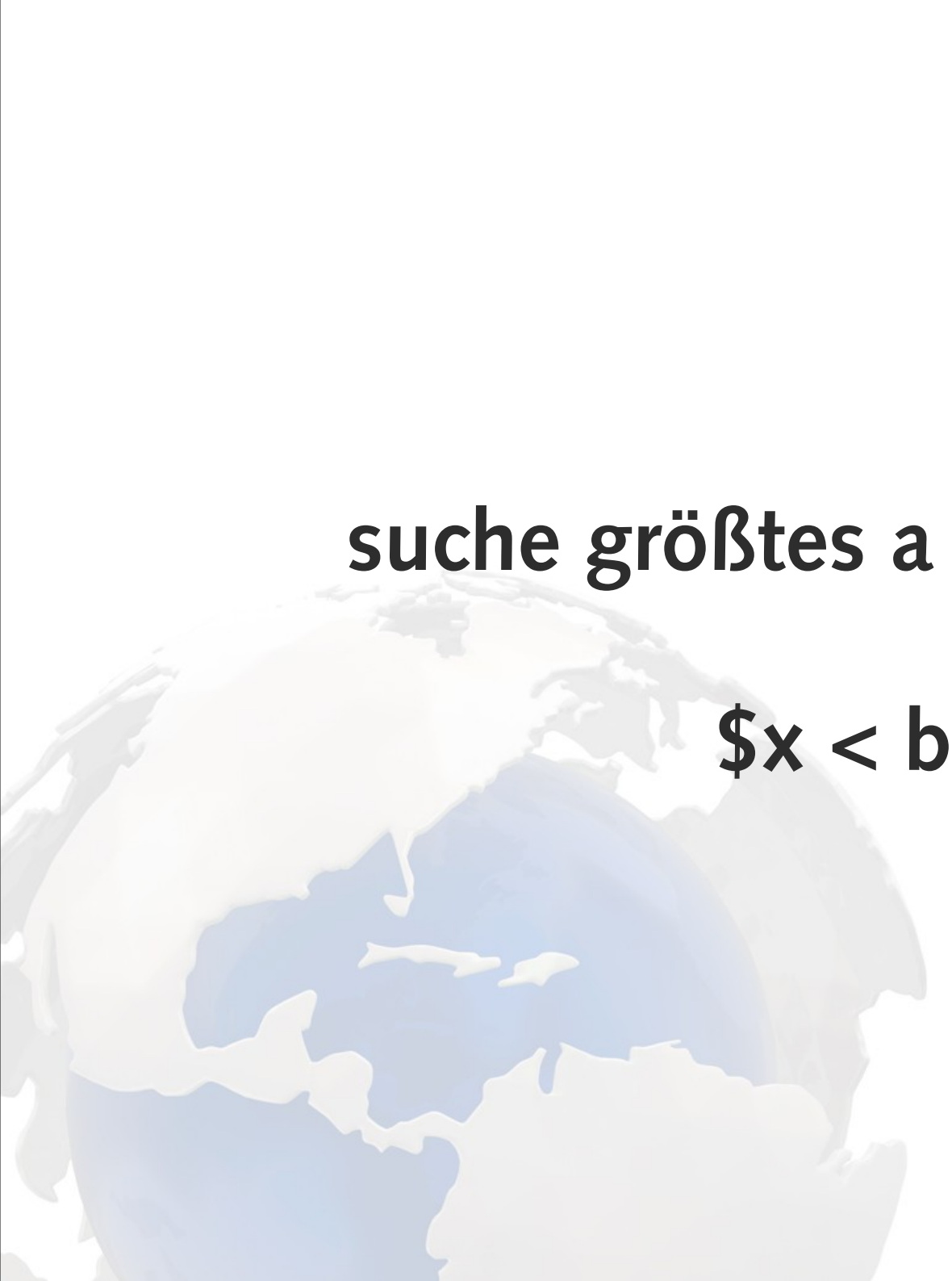
[...]

HASH-n.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH-n+1

HASH-n+1.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH1

(hash-n < hash-n+m, m>0)


HASH1.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH2
HASH2.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH3
[...]
HASH-n.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH-n+1
HASH-n+1.domain.tld 3600 IN NSEC3 1 1 100 ABCDEF01 HASH1
(hash-n < hash-n+m, m>0)



suche größtes a mit a < $x.

$x < b?

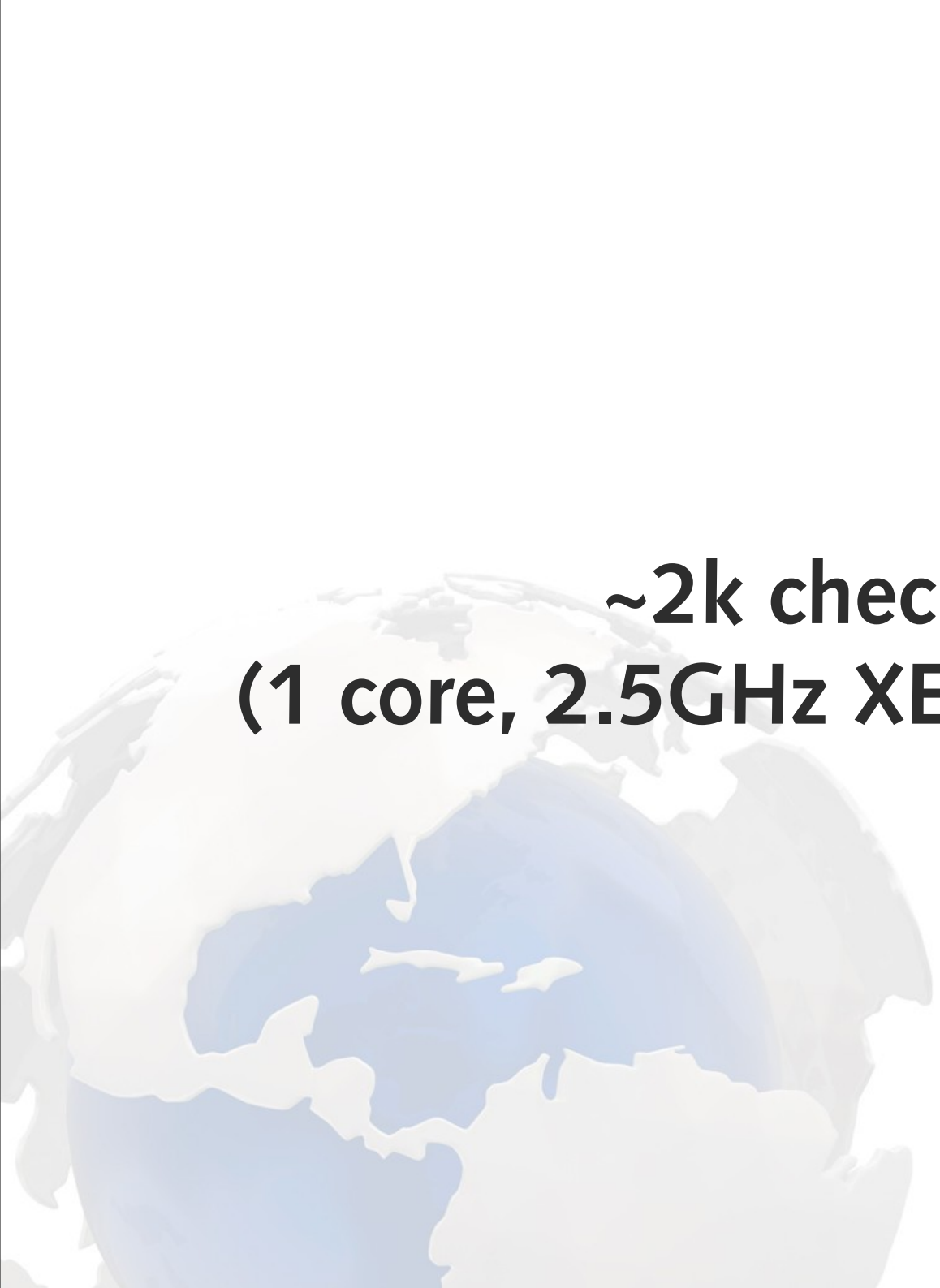
use Tree::RedBlack + custom search

Debugging...

```
utft2iokc8ljjuhi2r0n89sk4rmuo1jp|3ebbqejgnnpp69uudovuv672nl5hlv2j
( 9t6b7cte4t0do73miee9o5c1539ru39m|alh99tvdbd2155enhqoj84og8uabe27p
( 5gilipdo1icff7gj6sn966kqmtk76si7|6i4plaf05cmphboe5h7b68cl6ospmaah
( 3j447bpith6qt2815t98qutc2c2dpf9v|48hjivl94v00slt68a21opujea2nm7dt
( 3ebbqejgnnpp69uudovuv672nl5hlv2j|3j447bpith6qt2815t98qutc2c2dpf9v ### ###)
 ( 48hjivl94v00slt68a21opujea2nm7dt|5gilipdo1icff7gj6sn966kqmtk76si7 ### ###)
)
 ( 6i4plaf05cmphboe5h7b68cl6ospmaah|9t6b7cte4t0do73miee9o5c1539ru39m ### ###)
)
 ( dhrp6bmqcihohin8tlvk8lre8sll5mqj|geugtfk6n46888mgmvfuo7ajb1rcocpu
( b632ambi66vt0ujj9ak3noluj80i498l|bg4p2pe768dm8vqhrm5nvnoh1pi6v22t
( alh99tvdbd2155enhqoj84og8uabe27p|b632ambi66vt0ujj9ak3noluj80i498l ### ###)
 ( bg4p2pe768dm8vqhrm5nvnoh1pi6v22t|dhrp6bmqcihohin8tlvk8lre8sll5mqj ### ###)
)
 ( p0vlv7c4ohheep6ibmombopp2j1vrghu|r0tlp9jvu7humviil711vupjahfammv6
( ita5ak1embs9nld4kvdl871u5q14ilaf|j95u9bb6t6fd90l8s4mbtpjjsij5nj4a
( geugtfk6n46888mgmvfuo7ajb1rcocpu|ita5ak1embs9nld4kvdl871u5q14ilaf ### ###)
 ( j95u9bb6t6fd90l8s4mbtpjjsij5nj4a|p0vlv7c4ohheep6ibmombopp2j1vrghu ### ###)
)
 ( rk23q85mmf228fpa8lckghebbcta2moh|tskfk5iqalo64lnoueodccpft12md2ks
( r0tlp9jvu7humviil711vupjahfammv6|rk23q85mmf228fpa8lckghebbcta2moh ### ###)
 ( u2idlls97jsom8gae3t79vckjjums079|utft2iokc8ljjuhi2r0n89sk4rmuo1jp
( tskfk5iqalo64lnoueodccpft12md2ks|u2idlls97jsom8gae3t79vckjjums079 ### ###)
 ###)
)
)
)
)
```

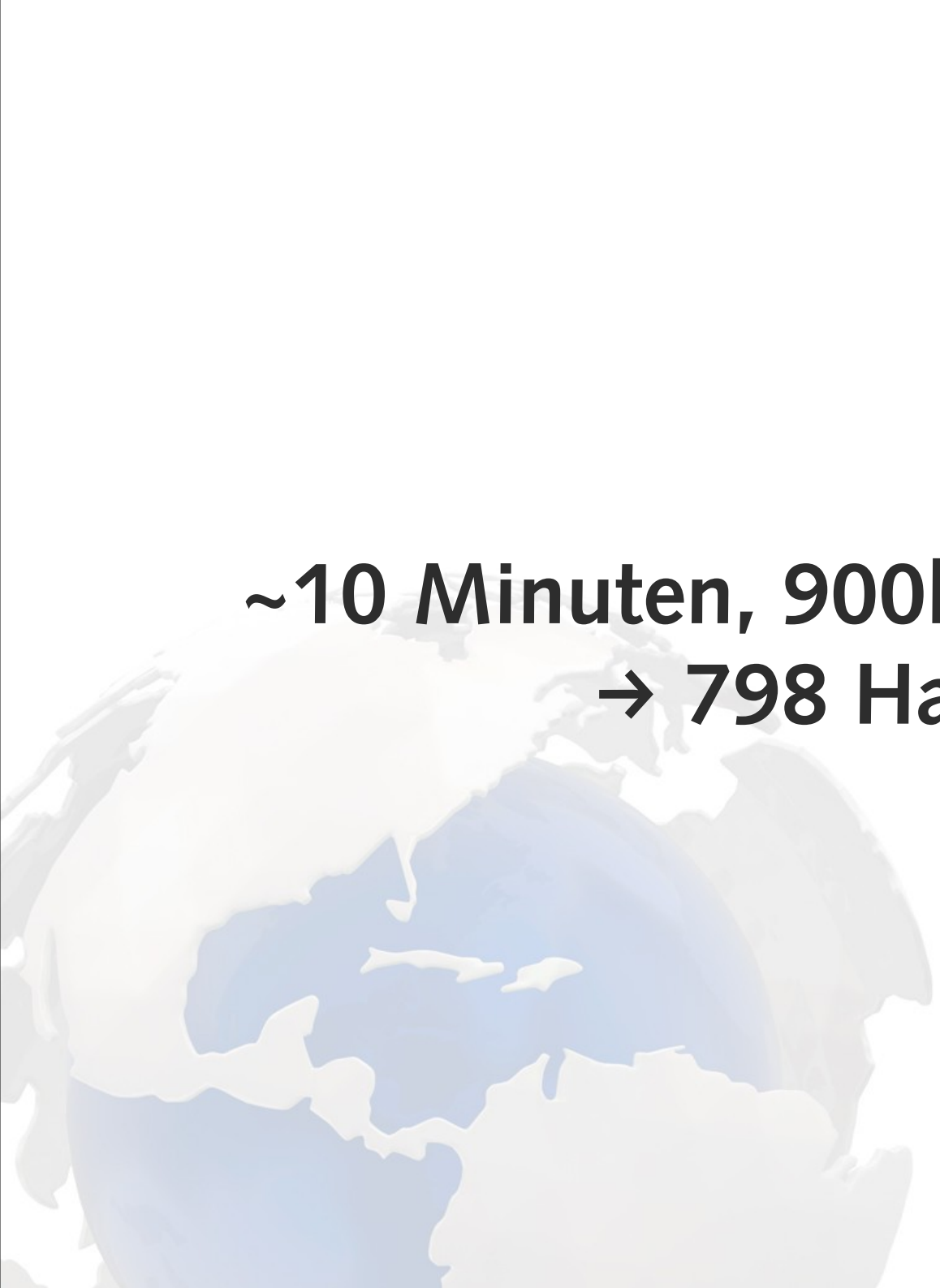

DANGER
Emacs

```
( 9t6b7cte4t0do73miee9o5c1539ru39m|alh99tvdbd2155enhqoj84og8uabe27p
  ( 5gilipdo1icff7gj6sn966kqmtk76si7|6i4plaf05cmphboe5h7b68cl6ospmaah
    ( 3j447bpith6qt2815t98qutc2c2dpf9v|48hjivl94v00slt68a21opujea2nm7dt
      ( 3ebbqejgnnpp69uudovuv672nl5hlv2j|3j447bpith6qt2815t98qutc2c2dpf9v ### ###)
      ( 48hjivl94v00slt68a21opujea2nm7dt|5gilipdo1icff7gj6sn966kqmtk76si7 ### ###))
    ( 6i4plaf05cmphboe5h7b68cl6ospmaah|9t6b7cte4t0do73miee9o5c1539ru39m ### ###))
  ( dhrp6bmqcihohin8tlvk8lre8sll5mqj|geugtfk6n46888mgmvfuo7ajb1rcocpu
    ( b632ambi66vt0ujj9ak3noluj80i498l|bg4p2pe768dm8vqhrm5nvnoh1pi6v22t
      ( alh99tvdbd2155enhqoj84og8uabe27p|b632ambi66vt0ujj9ak3noluj80i498l ### ###)
      ( bg4p2pe768dm8vqhrm5nvnoh1pi6v22t|dhrp6bmqcihohin8tlvk8lre8sll5mqj ### ###))
    ( p0vlv7c4ohheep6ibmombopp2j1vrghu|r0tlp9jvu7humviil711vupjahfammv6
      ( ita5ak1embs9nld4kvdl871u5q14ilaf|j95u9bb6t6fd90l8s4mbtpjjsij5nj4a
        ( geugtfk6n46888mgmvfuo7ajb1rcocpu|ita5ak1embs9nld4kvdl871u5q14ilaf ### ###)
        ( j95u9bb6t6fd90l8s4mbtpjjsij5nj4a|p0vlv7c4ohheep6ibmombopp2j1vrghu ### ###))
      ( rk23q85mmf228fpa8lckghebbcta2moh|tskfk5iqalo64lnoueodccpft12md2ks
        ( r0tlp9jvu7humviil711vupjahfammv6|rk23q85mmf228fpa8lckghebbcta2moh ### ###)
        ( u2idlls97jsom8gae3t79vckjjums079|utft2iokc8ljjuhi2r0n89sk4rmuo1jp
          ( tskfk5iqalo64lnoueodccpft12md2ks|u2idlls97jsom8gae3t79vckjjums079 ### ###)
          ###)))))
```



~2k checks/s
(1 core, 2.5GHz XEON quadcore)

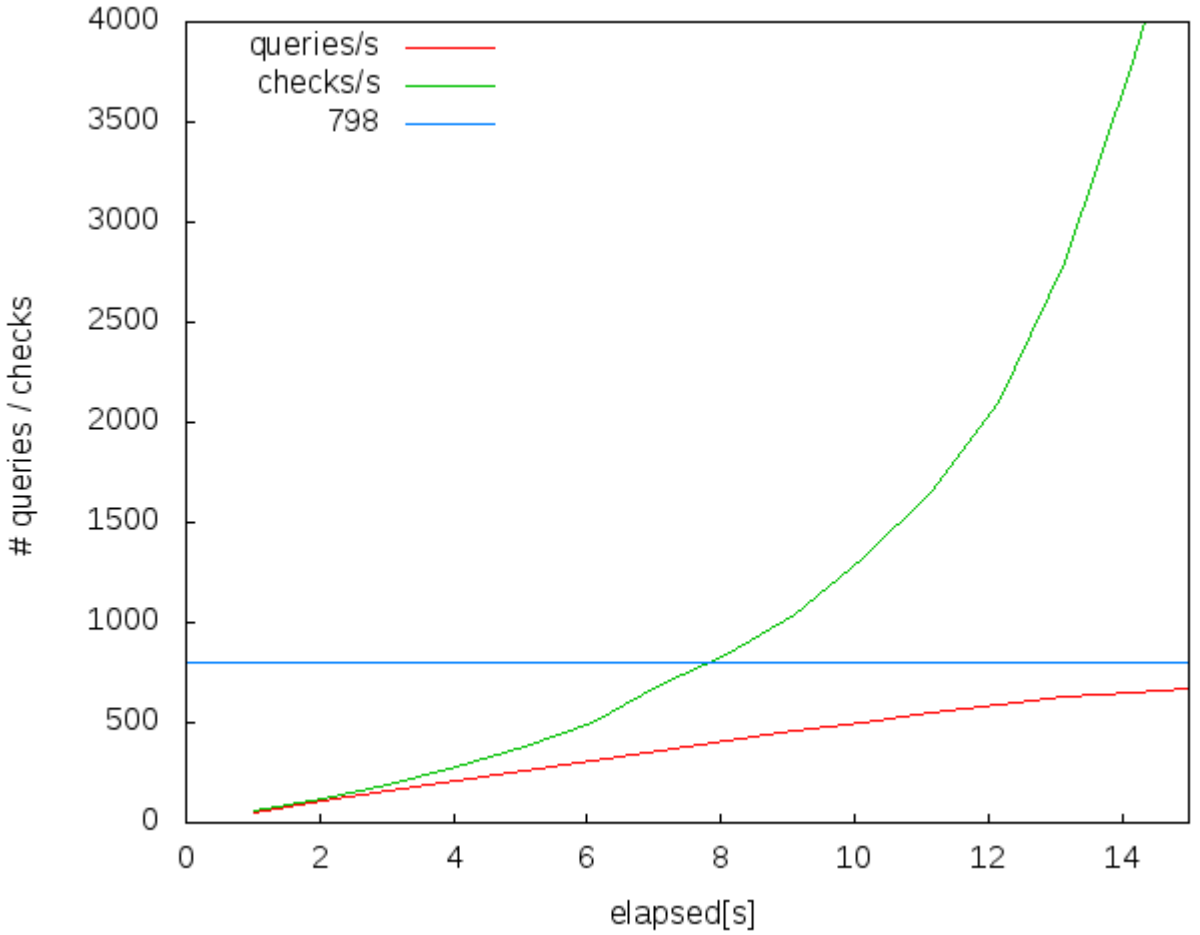
bund.de

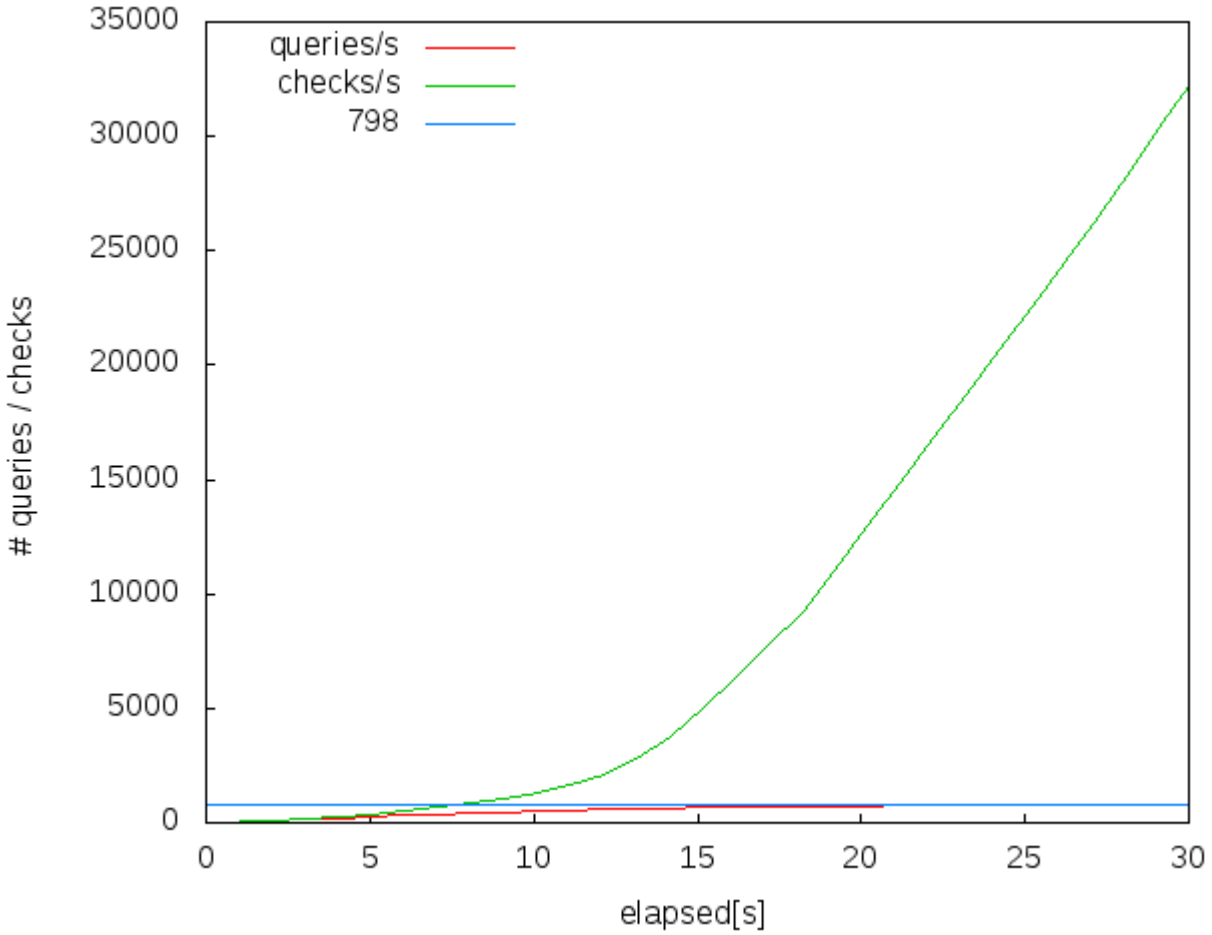


~10 Minuten, 900k-1.2M checks
→ 798 Hashes

NSEC3 garantiert:

max. 798 DNS queries


queries/s
checks/s
798
queries / checks
elapsed[s]
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14


queries/s
checks/s
798
queries / checks
elapsed[s]
35000
30000
25000
20000
15000
10000
5000
0
0
5
10
15
20
25
30

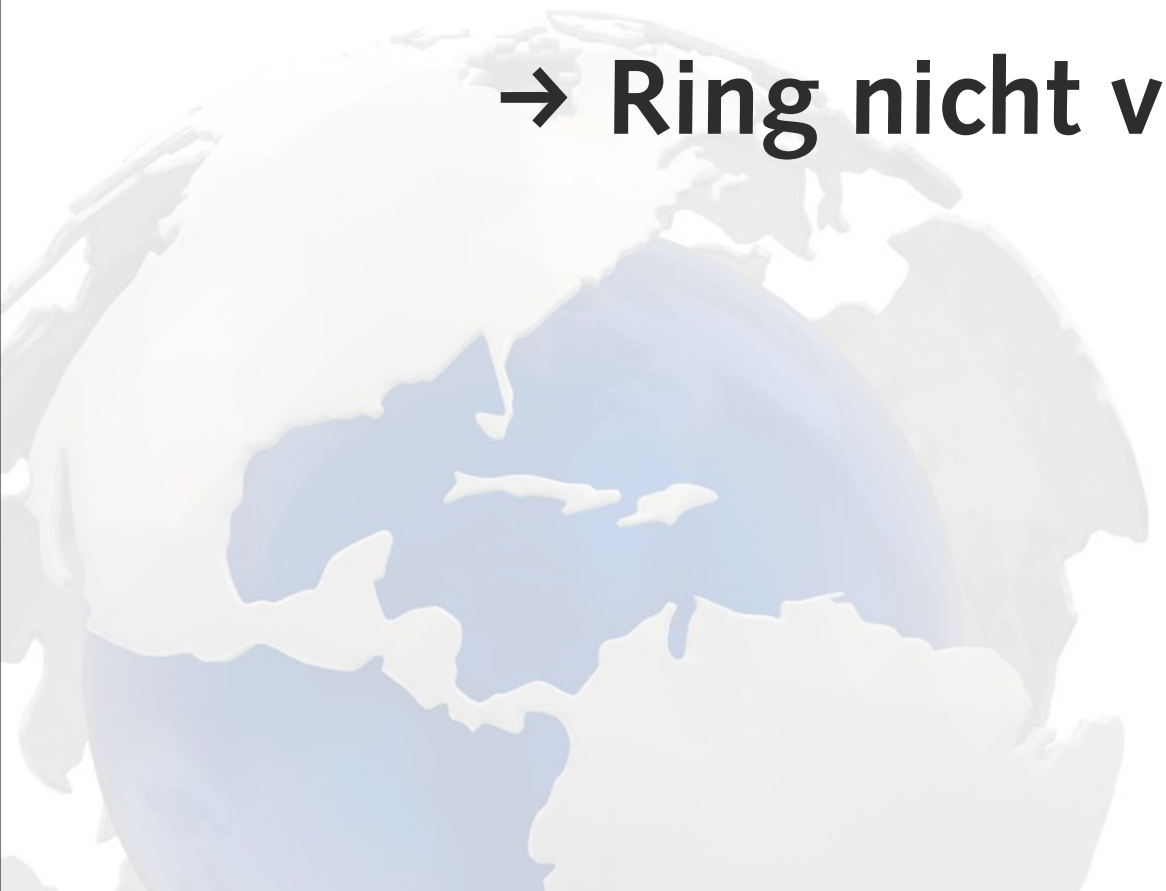
.de

200 Millionen checks

~ 1 Tag

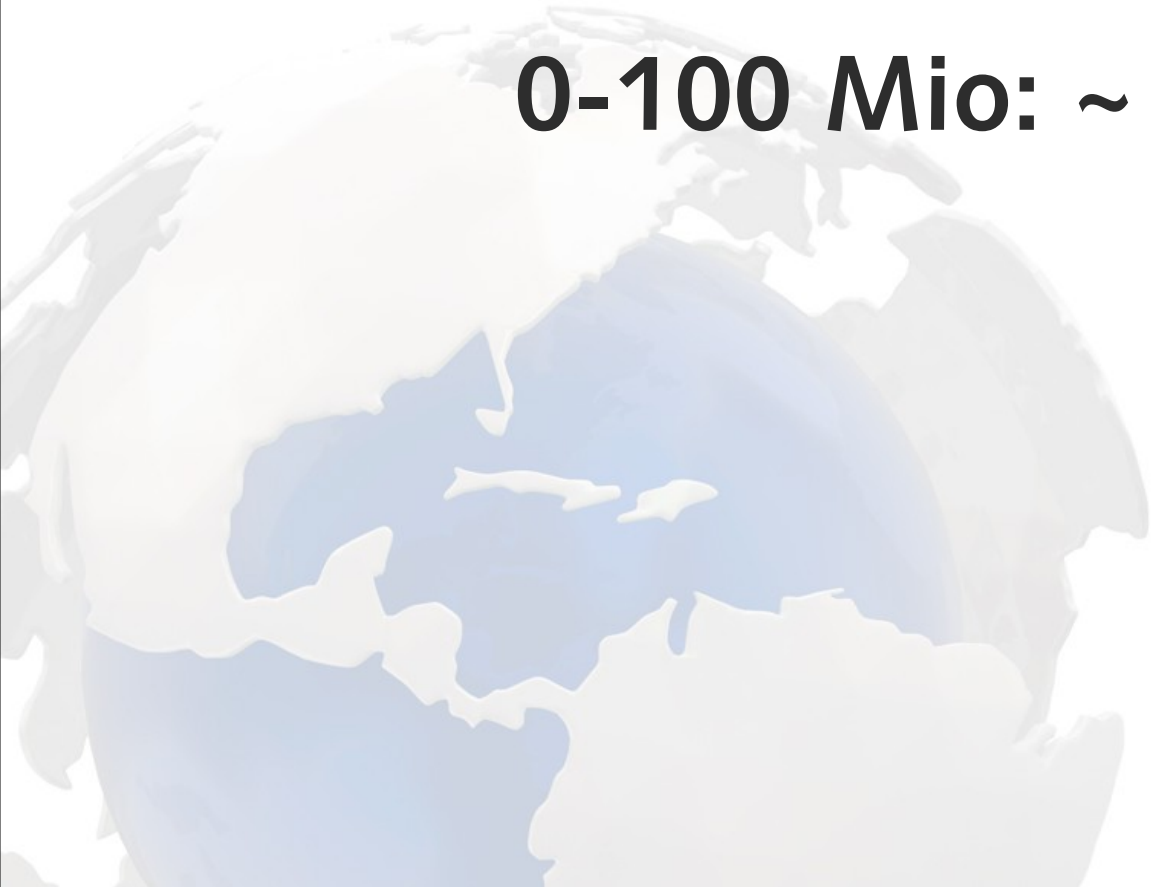
501.517 NSEC3 RRs

504.628 unique Hashs



→ Ring nicht vollständig

aber fast?



0-100 Mio: ~ 500k RRs

100-200 Mio: ~ 2k RRs

Schritt 2: dictionary attack

de.wikipedia.org XML Dump

7.745.060 „Wörter“

„stream-mode“ in ldns-nsec3-hash patchen

bund.de

409 / 798

Wikipedia Relevanz Kriterien?

7.500 checks/s

17 Minuten

5 stelligen Permutationen:

+151

.de

Wikipedia:

47.699 / 504.628

5 stellige Permutationen:

+7495

Bruteforce geht auch ohne DNSSEC!1!11!

hmja...

Bruteforce skaliert ~ linear

.de NSEC3 Antwort: ~ 1KByte

7.500 checks/s

58 Mbit/s

kurzes Suchen:
15 Quadcores / 5 Dualcores

Ich darf 4Gbit/s Traffic im Testbed / bei bund.de verursachen?

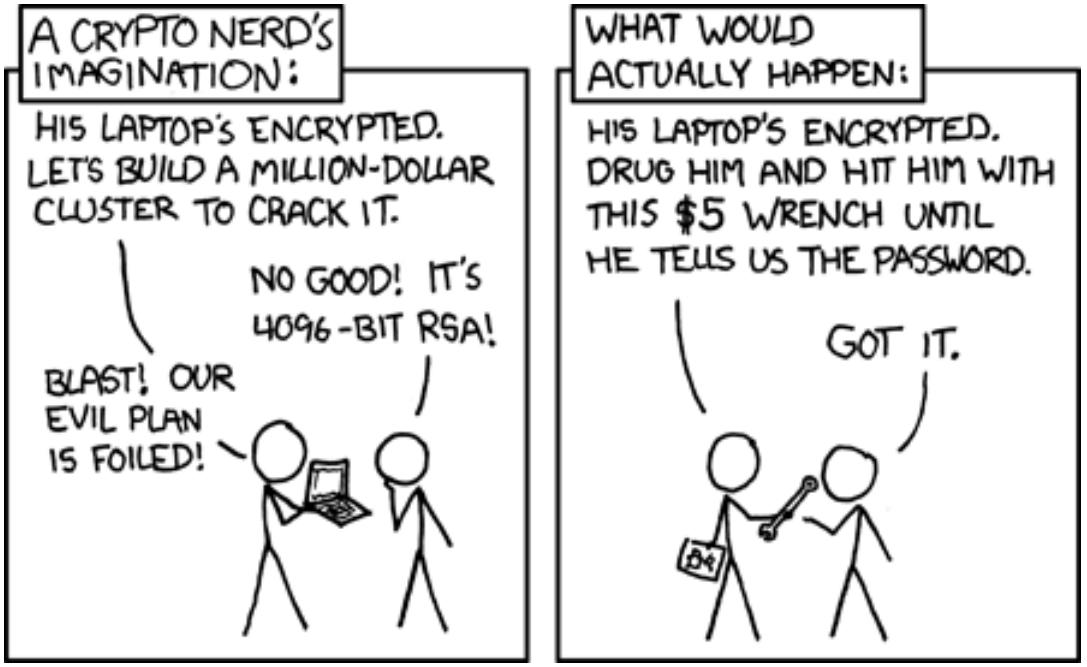
Bruteforce + NSEC3: deutlich einfacher

Bruteforce offline

Domains != Passwörter

Heuristik verbessern:
name24.de
name1-name2.de
name-shop.de
etc...

Plugin für john?


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.


Fragen?