R411 – Cyber
Sécurisation de services réseaux
(DNSSEC)

IUT Béziers, dépt. R&T © 2012-2023
http://www.borelly.net/
Christophe.BORELLY@umontpellier.fr

Généralités

- BIND : Berkeley Internet Name Daemon
- Début du projet dans les années 1980
- Actuellement développé par l'Internet Systems Consortium (http://www.isc.org/)
- **BIND 9 :** DNS Notify, Dynamic Update, IXFR (Incremental zone transfer), Split DNS (Views), TSIG (Transaction SIGnature), TKEY (Transaction KEY), DNSSEC (DNS Security Extensions), SIG (0) (DNS Request and Transaction Signatures), Dynamic Zones, Automatic Signing, Dynamic Trust Anchor Management, PKCS #11 (Cryptoki), IPv6, DoT, DoH...
- Dernière version 9.18.14 (19/04/2023)

Fichiers et outils

- `/etc/resolv.conf` (clients)
  - Ou bien `resolvctl` (systemd)
- `/etc/named.conf` (serveur)
- Outils de diagnostiques :
  - dig, delv, host et nslookup
- Outils coté serveur : named-checkconf, named-checkzone, rndc, ...
  - `/etc/rndc.conf` et/ou `/etc/rndc.key`
  - `/etc/bind.keys` (DNSSEC - https://www.isc.org/bind-keys)

/etc/resolv.conf

```
search borelly.net
nameserver 192.168.2.1
nameserver 192.168.2.2

$ dig google.fr
...
google.fr.      60553   IN  NS  ns4.google.com.
...
ns4.google.com.     81775   IN  A   216.239.38.10
...
$ dig @192.168.2.1 www.iutbeziers.fr +short
www.iutbeziers.univ-montp2.fr.
194.199.227.80
$ dig @192.168.2.1 borelly.net SOA +short
dns1.borelly.net. postmaster.borelly.net. 2011112001 21600 3600 3024000 86400
```

resolvctl (systemd)

- `resolvctl status`
- `resolvctl statistics`
- `resolvctl dns`
- `resolvctl dns 8.8.8.8`
- `resolvctl dns eth0 8.8.8.8`
- `...`
- Configuration par defaut :

`/etc/systemd/revolved.conf`

/usr/sbin/named

```
# named -V
BIND 9.7.4 built with '--prefix=/usr' … 'CFLAGS=-
O2 -march=i486 -mtune=i686'
```

- BIND en mode debug « foreground » :

```
# named -g (-d1 ou -d2)
```

/etc/named.conf (1)

```
options {
        directory "/var/named";
};

zone "." IN {
        type hint;
        file "zone.root";
};

zone "borelly.net" IN {
        type master;
        file "zone.borelly.net";
        allow-update { none; };
};
```

/etc/named.conf (2)

```
zone "localhost" IN {
   type master;
   file "zone.localhost";
   allow-update { none; };
};

zone "127.in-addr.arpa" IN {
   type master;
   file "zone.127.in-addr.arpa";
   allow-update { none; };
};
```

rndc

- Remote Name Daemon Control
- `rndc reload`
- `rndc reload zone`
- `rndc refresh zone`
- `rndc retransfer zone`
- `rndc status`
- `rndc stop`
- `...`

/etc/rndc.conf

```
# /etc/rndc.conf (generated by rndc-confgen)
key "rndc-key" {
   algorithm hmac-md5;
   secret "SgakZ1xY9zpL7WwJJ6pAew==";
};

options {
   default-key "rndc-key";
   default-server 127.0.0.1;
   default-port 953;
};
```

/etc/named.conf (3)

```
...

key "rndc-key" {
        algorithm hmac-md5;
        secret "SgakZ1xY9zpL7WwJJ6pAew==";
};
controls {
        inet 127.0.0.1 port 953
        allow { 127.0.0.1; } keys { "rndc-key"; };
};
```

Fichier de zone (1)

- RFC 1034
- Resource Record (RR)
- Éléments d'un RR : Nom, Classe , Type, TTL (Time-to-live), RDATA (Resource data)
- Classe : **IN** (Internet), **CH** (Chaosnet) ou **HS** (Hesiod)
- Type les plus courants : **A**, **AAAA**, **CNAME**, **MX**, **NS**, **PTR**, **SOA**, ...

```
www.borelly.net. IN A 192.168.2.1
```

Fichier de zone (2)

- **$TTL** directive globale à mettre en tout début
- @ représente l'origine courante (en début de fichier, le nom de la zone terminé par un .)
- **$ORIGIN** permet de compléter le nom des enregistrements « non qualifiés » :

```
$ORIGIN exemple.com.
www CNAME dns
```

  - Equivalent à :

```
www.exemple.com. CNAME dns.exemple.com.
```

Exemple

```
$TTL    1H
@ IN SOA dns1.borelly.net. postmaster.borelly.net. (
   2011112001 ; serial
   6H         ; refresh (6 hours - 21600)
   1H         ; retry   (1 hour  - 3600)
   5W         ; expire  (5 weeks - 3024000)
   1D         ; minimum (1 days  - 86400)
)
        IN NS      dnscb1.iutbeziers.fr.
        IN NS      dnscb2.iutbeziers.fr.
        IN MX    0 mail.borelly.fr.

dns1    A  192.168.2.1
dns2    A  192.168.2.2
mail    A  192.168.2.3
www     A  192.168.2.4
```

Résolution inverse

- Domaine : **in-addr.arpa** et/ou **ip6.arpa.**
- Écriture inversée de l'adresse IP :
  - `192.168.2.1` correspond à `1.2.168.192.in-addr.arpa.`
  - `2001:db8:0f00:12:34ff:fe56:789a` donne `a.9.8.7.6.5.e.f.f.f.4.3.2.1.0.0.0.0.0.0.0.0.f.0.8.b.d.0.1.0.0.2.ip6.arpa.`
- Utilisation du type **PTR** :

```
$ORIGIN 2.168.192.in-addr.arpa.
1 PTR dns1.borelly.net.
2 PTR dns2.borelly.net.
3 PTR mail.borelly.net.
```

Zone secondaire

- Sur le master (192.168.2.1) :

```
zone "borelly.net" {
  type master;
  file "zone.borelly.net";
  allow-transfer { 192.168.2.2; };
  allow-update { none; };
};
```

- Sur le slave (192.168.2.2) :

```
zone "borelly.net" {
  type slave;
  file "zone.borelly.net.bak";
  masters { 192.168.2.1; };
  allow-transfer { none; };
};
```

TSIG (Transaction SIGnature)

- RFC 2845
  - Secret Key Transaction Authentication for DNS
- Forme d'authentification pour les mises à jour dynamiques des bases de données DNS
- **Secret partagé** + fonction de hachage (HMAC)

TSIG - Sécurisation « master »

- Génération d'une clé :

~~dnssec-keygen -a HMAC-SHA1 -b 128 -n HOST myKey~~

```
tsig-keygen -a HMAC-SHA256 myKey
```

- Fichier de configuration :

```
...
key myKey {
  algorithm hmac-sha256;
  secret "MzEMMySfzMhkq1od9YFjEQSQRLv4KpfJkSGgNX4iBKM=";
};
zone "borelly.net" {
  type master;
  file "zone.borelly.net";
  allow-transfer { key myKey; };
  allow-update { none; };
};
```

TSIG - Sécurisation « slave »

```
...
key myKey {
  algorithm hmac-sha256;
  secret "MzEMMySfzMhkq1od9YFjEQSQRLv4KpfJkSGgNX4iBKM=";
};
server 192.168.2.1 {
  keys { myKey; };
};
...
```

TSIG - nsupdate

```
…
zone "borelly.net" {
  type master;
  file "zone.borelly.net";
  allow-update { key myKey; };
};
$ nsupdate -d -k KmyKey.private nsupdate.conf
...
;; TSIG PSEUDOSECTION:
myKey. 0 ANY  TSIG hmac-sha256. 1354851135 300 20
5jTN+d7yABGppwoq2EM618lN78Y= 55960 NOERROR 0
```

- Fichier nsupdate.conf (cf. man nsupdate) :

```
server 192.168.2.1
update add www.borelly.net 300 IN A 10.3.2.1
send
```

TKEY (Transaction KEY)

- RFC 2930
  - Secret Key Establishment for DNS (TKEY RR)
- Système de génération automatique d'un secret partagé
- Utilisation de algorithme d'échange de clés Diffie-Hellman

```
dnssec-keygen -a DH -b 128 -n HOST pccb
```

SIG (0)

- RFC 2931
- Protection des requêtes et des transactions DNS
- Authentification à clés publiques
- Génération de clés (**-C** : mode de compatibilité pour les anciennes version de bind = supprime les metadata)

```
$ dnssec-keygen -a RSASHA1 -b 1024 -T KEY -C \
                -n HOST pccb.borelly.net
Kpccb.borelly.net.+005+60169  (.key et .private)
$ cat Kpccb.borelly.net.+005+60169.key
pccb.borelly.net. IN KEY 512 3 5
AwEAAZb6Ih6HpXJ19VbF6GzxtyUoIS4Nu89FcjYGjXBttRQdizSg8KPs
vJ4OBb+85ua9UlMaGOWOuFR92j8liIPPlt0VMrP7k36h9pGiKSP9FInZ
+heiByCXScc0Vq3qYu8IzpXpT6Y2BXX7kWNp1gEbn+hL51vDpTKV+vFo BdFf3U7X
```

SIG (0) – Fichier de zone

```
...
$ORIGIN borelly.net
...
pccb    IN A 10.1.2.3
        IN AAAA 2001:db8::1
$INCLUDE keys/Kpccb.borelly.net.key
...
```

SIG (0) - « Master »

```
options {
   ...
   dnssec-enable yes;
};
zone "borelly.net" {
   type master;
   file "zone.borelly.net";
   update-policy {
     grant pccb.borelly.net subdomain borelly.net ANY;
   };
   //update-policy { grant * self * A AAAA; };
   ...
};
...
```

SIG (0) - nsupdate

```
$ nsupdate -d -k Kpccb.borelly.net.private \
           nsupdate.conf
...
;; SIG0 PSEUDOSECTION:
. 0 ANY SIG  0 5 0 0 20121207092511 20121207091511 60169
    pccb.borelly.net.
kKVStpw4sm9wQu5qAs4KwfBdcXy2R/eDMCnpnoLuY9vmEyOLomOUljHP
Iwid1pECJfAxyeu5ZyjVUQfHzLy8kzKFpMecNJf/Ac5qvspu1bfIgEEj
IKRZyunb2NscRuA0XCU55ghRzpL3n8MkbYQCNZpRdHHwe8VkxrdSldr1 Hjk=

Reply from update query:
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:  14553
;; flags: qr ra; ZONE: 1, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; ZONE SECTION:
;borelly.net.       IN SOA
```

DNSSEC (DNS Security Extensions)

- RFC 4033, RFC 4034 et RFC 4035
- Signature des enregistrements DNS (vérifiables)
- Nouveaux type RR (resource record) :
  - **RRSIG** (signature), **DNSKEY** (public key), **DS** (Delegation Signer), **NSEC** (Next secure record), **NSEC3** (NSEC v3), **NSEC3PARAM**
- Début 2010, environ une dizaine de TLD (Top Level Domain) sont signés

Clés DNSSEC

- Créer 2 clés asymétriques par sous domaine
  - **ZSK** – Zone Signing Key
  - **KSK** – Key Signing Key

```
options {
   ...
   dnssec-enable yes;
   dnssec-validation yes;
};
...
$include Kborelly.net.ksk.key
$include Kborelly.net.zsk.key
...
```

DNSSEC NSEC/NSEC3

- Génération des clés, algorithmes possibles :
  - ~~RSAMD5~~, **RSASHA1**, DSA, **NSEC3RSASHA1**, NSEC3DSA, RSASHA256, RSASHA512, ECCGOST, ECDSAP256SHA256 ou ECDSAP384SHA384
- Key Signing Key (KSK) :

```
$ dnssec-keygen -f KSK -a RSASHA1 -b 4096 -n ZONE borelly.net
Kborelly.net.+008+05644
$ dnssec-keygen -f KSK -a NSEC3RSASHA1 -b 4096 -n ZONE borelly.net
Kborelly.net.+005+60391
```

- Zone Signing Key (ZSK) :

```
$ dnssec-keygen -a RSASHA1 -b 2048 -n ZONE borelly.net
Kborelly.net.+008+06249
$ dnssec-keygen -a NSEC3RSASHA1 -b 2048 -n ZONE borelly.net
Kborelly.net.+005+06396
```

NSEC et NSEC3

- Ces 2 systèmes permettent de vérifier la preuve de la non existence d'un enregistrement (RFC 7129).
- NSEC - RFC 4034 - permet d'être sûr qu'un enregistrement n'existe pas, en indiquant qu'il n'y a rien entre 2 noms consécutifs.
- NSEC3 - RFC 5155 – utilise des noms hachés pour éviter le parcours de zone.

Signature NSEC/NSEC3

- Signature NSEC du fichier de zone :

```
dnssec-signzone -N increment \
                -o borelly.net zone.borelly.net
```

- Signature NSEC3 du fichier de zone :

```
SALT=$(head -c 1000 /dev/urandom | sha1sum | cut -b 1-16)
dnssec-signzone -3 $SALT -N increment \
                -o borelly.net zone.borelly.net
```

- Génère le fichier zone.borelly.net.signed
- Génère le fichier dsset.borelly.net. qui contient les enregistrements DS à indiquer dans la zone « parent ».

```
borelly.net. IN DS 5644 8 1 DF24...012A
borelly.net. IN DS 5644 8 2 831C...9070
```

Exemple NSEC

```
@ IN SOA @ postmaster (
      43 3H 15M 1W 1D
  )
  IN NS    borelly.net.
  IN A     127.0.0.1

a IN A     127.0.0.4
w IN A     127.0.0.7
```

```
borelly.net.        86400     IN SOA    borelly.net.
postmaster.borelly.net. (
 44         ; serial
 10800      ; refresh (3 hours)
 900        ; retry (15 minutes)
 604800     ; expire (1 week)
 86400      ; minimum (1 day)
)
86400     RRSIG     SOA 8 2 86400 (
  20180413111024 20180314111024 6249
  borelly.net. CaYs...xbc= )
86400     NS   borelly.net.
86400     RRSIG     NS 8 2 86400 (
  20180413111024 20180314111024 6249
  borelly.net. rWk/x02L...m7k= )
86400     A    127.0.0.1
86400     RRSIG     A 8 2 86400 (
  20180413111024 20180314111024 6249
  borelly.net. Y9Jm4lsN...gpY= )

borelly.net.   86400 NSEC a.borelly.net. A NS SOA RRSIG NSEC
DNSKEY
...
a.borelly.net. 86400 NSEC w.borelly.net. A RRSIG NSEC
...
w.borelly.net. 86400 NSEC borelly.net. A RRSIG NSEC
```

Exemple NSEC3

```
...
  0 NSEC3PARAM 1 0 10 4F823546E9BAC449
...
KQ4FPCF9TBQHE69N79QJNAB72VEDK6R3.borelly.net. 86400 IN NSEC3
 1 0 10 4F823546E9BAC449
 ( 7T0J0MUIFKLIGC01QQ9MOSKD3VRDOITE A NS SOA RRSIG DNSKEY
NSEC3PARAM )

7T0J0MUIFKLIGC01QQ9MOSKD3VRDOITE.borelly.net. 86400 IN NSEC3
 1 0 10 4F823546E9BAC449
 ( OH41PS70NVMNH04UE5MSNVV2GTNIEKNK A RRSIG )
...
OH41PS70NVMNH04UE5MSNVV2GTNIEKNK.borelly.net. 86400 IN NSEC3
 1 0 10 4F823546E9BAC449
 ( KQ4FPCF9TBQHE69N79QJNAB72VEDK6R3 A RRSIG )
...
```

delv

- DNS lookup and validation utility

```
cb@pccb$ delv @194.199.227.111 borelly.net
; fully validated
borelly.net.    3600   IN A  194.199.227.111
borelly.net.    3600   IN RRSIG  A 7 2 3600 20220410095552
20220110085552 9032 borelly.net.
LMYjixu0+Ea5J2wjdLiLOWbwxdZ92mhfTIu8hf+AAjatLKUOGUtC7ZwG
ETezPDQpLFvuBzqcgiUdN7o6hPO2w9mOvTqGWzh3izPudJf695ajSgo3
LQ/c35fUzh98OZXrX+SO4WAxS/KWyx21Nscf+pdOgFHE8XsvTBRt3/XL
V0gkf0sMs4FUQqyUkgToCxSCIaWkwQF9neoXF++S3aKDzZvnxGBPfnw+
DQX6KD6tULXc3j4c68TvWRp8+8P33kv5SxthtXHnaAvX7CP8hU5hJ4fI
IaXium2YQJnF/99dj35UpTutuQvqCKQ80WCIXLHjuV1FWT5DQXJSCS2R
YmdxeQ==
```

DoT et DoH

- DNS over TLS depuis 9.16.0 (port 853)
- DNS over HTTPS depuis 9.16.21 (port 443)

```
tls cbTLS {
  key-file "/path/to/priv_key.pem";
  cert-file "/path/to/cert_chain.pem";
};
http cbHTTPsrv {
  endpoints { "/dns-query"; };
};
options {
  ...
  #tls-port 8853;# Set default TLS port 853
  #http-port 8053;# Set default HTTP port 80
  #https-port 8443;# Set default HTTPS port 443
  listen-on port 8853 tls cbTLS { 4.3.2.1; }; # tls
  listen-on port 8053 http cbHTTPsrv { 9.8.7.6; }; # http
  listen-on port 8453 tls cbTLS http cbHTTPsrv { 8.7.6.5; }; # https
  ...
};
```

Clients DoT, DoH

- dig @127.0.0.1 google.com +tls
- https://github.com/dcid/ (PHP clients DoT et DoH)
- For DoT, you can use kdig tool provided by knot.

```
kdig -d @8.8.8.8 +tls-ca +tls-host=dns.google.com example.com
```

- curl has official DoH support since version 7.62.0

```
curl --doh-url https://cloudflare-dns.com/dns-query https://www.google.com
```

- https://github.com/ameshkov/dnslookup
- https://github.com/natesales/q

```
q --dnssec A AAAA example.com @https://dns.example.com
q --dnssec A AAAA example.com @tls://dns.example.com
```

- https://github.com/byu-imaal/dohjs/
- ...

DANE (1)

- DNS-Based Authentication of Named Entities
- RFC 6698
- Enregistrements **TLSA** ("TLSA" does not stand for anything; it is just the name of the RRtype)
  - 0 – AC PKIX
  - 1 – Certificat TLS spécifique + vérif. PKIX
  - 2 – AC personnelle
  - 3 – Certificat TLS spécifique sans vérif. PKIX

DANE (2)

- Exemple type 3 pour le certificat du serveur (0) en HASH-SHA256 (1)

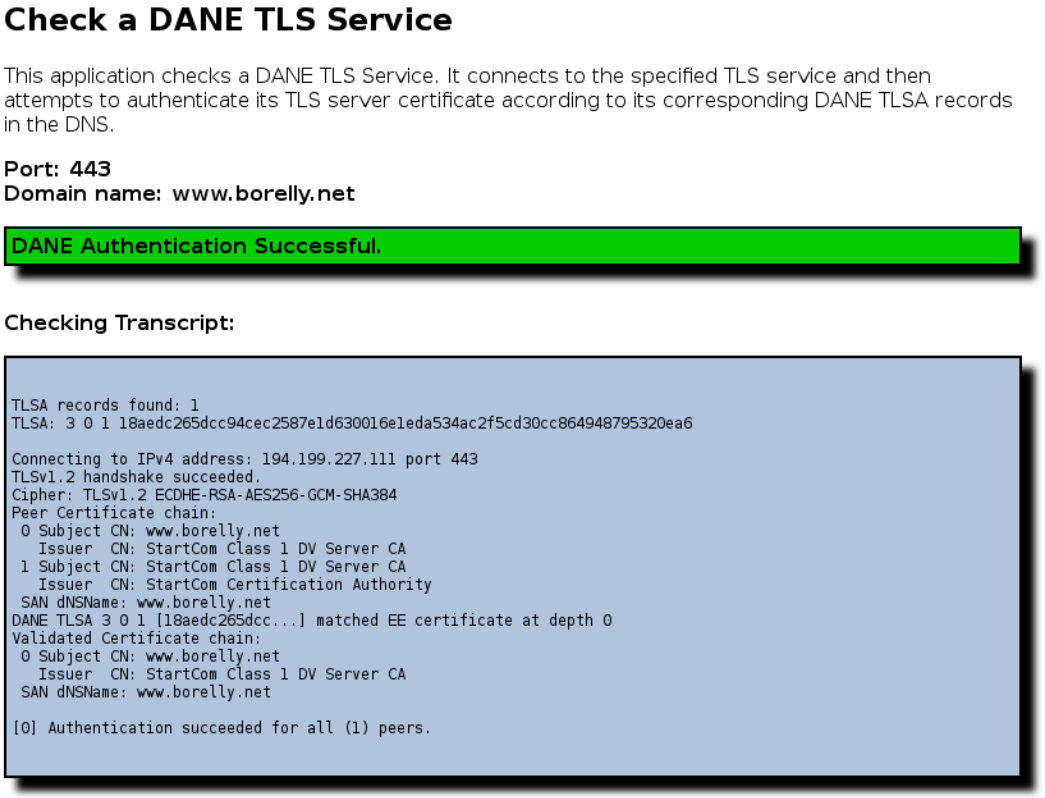
```
openssl x509 -in www.borelly.net.crt \
-outform DER | openssl sha256
(stdin)= 18aedc265dcc9...320ea6
```

- Dans le fichier de zone :

```
_443._tcp.www.borelly.net. IN TLSA 3 0 1
18aedc265dcc9...320ea6
```

Vérification DANE (3)

- https://www.huque.com/bin/danecheck
- Plugin pour Firefox "DNSSEC/TLSA Validator"



Check a DANE TLS Service

This application checks a DANE TLS Service. It connects to the specified TLS service and then attempts to authenticate its TLS server certificate according to its corresponding DANE TLSA records in the DNS.

Port: 443
Domain name: www.borelly.net

DANE Authentication Successful.

Checking Transcript:

```
TLSA records found: 1
TLSA: 3 0 1 18aedc265dcc94cec2587e1d630016e1eda534ac2f5cd30cc864948795320ea6

Connecting to IPv4 address: 194.199.227.111 port 443
TLSv1.2 handshake succeeded.
Cipher: TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384
Peer Certificate chain:
 0 Subject CN: www.borelly.net
   Issuer  CN: StartCom Class 1 DV Server CA
 1 Subject CN: StartCom Class 1 DV Server CA
   Issuer  CN: StartCom Certification Authority
 SAN dNSName: www.borelly.net
DANE TLSA 3 0 1 [18aedc265dcc...] matched EE certificate at depth 0
Validated Certificate chain:
 0 Subject CN: www.borelly.net
   Issuer  CN: StartCom Class 1 DV Server CA
 SAN dNSName: www.borelly.net

[0] Authentication succeeded for all (1) peers.
```

Références

- http://fr.wikipedia.org/wiki/BIND
- https://www.isc.org/software/bind/
- http://www.root-servers.org/
- http://www.root-dnssec.org/