Technicien Supérieur Gestionnaire
Exploitant de Ressources Informatiques
et Réseaux

DNSSEC


# UNBOUND

## Classic guide

O RLY?

Marchal Ludovic



# Table des matières

# Sécurité des requêtes DNS DNSSEC DNSSEC-TRIGGER Préambule :

## Faiblesse du DNS

Une des faiblesses de la sécurité de l'Internet est que le mécanisme d'annuaire principal, le DNS, n'est pas très sécurisé. Il est trop facile de tromper un serveur DNS en lui injectant de fausses informations (faille dite Kaminsky). Mais il y a pire : souvent, c'est le serveur DNS mis à la disposition de l'utilisateur, par le FAI ou par le service informatique local, qui le trompe.

Une solution technique existe, à ces deux problèmes : DNSSEC. Mais pour que DNSSEC protège M. Michu, les vérifications que permet ce protocole doivent être faites sur la machine de M. Michu, la seule en qui il peut avoir un peu confiance. C'est ce que permet le nouveau logiciel dnssec-trigger.

## RFC 3833: Threat Analysis of the Domain Name System (DNS)

Date de publication du RFC : Août 2004
Auteur(s) du RFC : D. Atkins (IHTFP Consulting), R. Austein (ISC)

note que les attaques suivantes sont possibles :

- Interception des paquets (questions ou réponses) et émission par l'attaquant d'un autre paquet. Tous les protocoles permettent cela mais c'est plus facile avec le DNS, où question et réponse tiennent souvent dans un seul paquet chacune. Cela nécessite en général que l'attaquant soit situé entre le client et le serveur.
- Fabrication d'une réponse vraisemblable. Cela peut se faire en devinant l'ID (un numéro choisi par le client DNS et qui lui permet de mettre en correspondance requêtes et réponses) d'une requête (et le nom demandé), ce que la faible taille des ID (16 bits) rend possible.
- Chainage de noms, une technique qui consiste à injecter des données dans le cache de la victime en utilisant une indirection. Par exemple, une réponse de type NS (*name server*) va entrainer des requêtes dans un domaine potentiellement complètement différent, pour résoudre les noms en partie droite de la réponse.
- Trahison par un serveur : si je branche mon portable sur un réseau que je ne contrôle pas, le serveur cache de ce réseau peut envoyer à ses clients des données mensongères. Même chose si un des secondaires d'un domaine décide de tricher, alors qu'il devrait n'envoyer que ce qu'a décidé son primaire.
- Déni de service

# RFC 4033: DNS Security Introduction and Requirements

Date de publication du RFC : Mars 2005
Auteur(s) du RFC : R. Arends (Telematica Instituut), R. Austein (ISC), M. Larson (VeriSign), D. Massey (Colorado State University), S. Rose (NIST)
Chemin des normes

la quasi-totalité des applications de l'Internet dépendent du DNS à un moment ou à un autre, cette vulnérabilité inquiète beaucoup de gens. Notre RFC, avec ses compagnons RFC 4034 et RFC 4035, propose une solution à certains de ces problèmes, solution nommée DNSSEC.

Ce RFC décrit le deuxième DNSSEC, le premier était présenté dans le RFC 2535 mais n'a jamais été un succès.

DNSSEC-bis propose une sécurisation des **données**, pas du **canal**.

La cryptographie permet  de sécuriser un canal de communication entre deux machines (ce que fait SSL) ou bien de sécuriser le message lui-même (ce que fait GPG et plus encore) , qui peut alors être authentifié et protégé quels que soient les messagers intermédiaires : c'est ce que fait DNSSEC.

Les parties **publiques** des clés utilisées par le **registre** de la zone pour la signature sont publiées dans des enregistrements de type DNSKEY. Pour "amorcer la pompe" de DNSSEC, on récupère typiquement ces clés via un canal non-DNS, par exemple un accès à un serveur Web authentifié.

## Solution

- Disposer de son résolveur
- Facilité de mise en place (ce que l'on va voir)
- Quasi-configuré avec une liste de serveur de confiance
    - (avec éventuellement une recherche supplémentaire)
- Gestion du DNS-trigger ( typiquement les DNS pourri des hall de gare , hôtel et autres zones WIFI daubé, c'est là que vous ferez les meilleurs pêche).

DNSSEC s'assure de l'authenticité d'un serveur de noms, c'est le « protocole de chiffrment » (pas tout à fait).

DENSSEC-TRIGGER s'assure de la fiabilité d'une zone c'est la partie qui s'occupe de la validation du chiffrement, cela répond à la question de savoir ou ça se passe.

# Presentation

## Unbound

### DNS Recursif, Cache, Authitaive

Développé aux NLnetLabs, partiellement financé par l'**AFNIC**, il va permettre de donner la puissance de DNSSEC.Contexte :

Resolveur en local, ou pour son LAN il y aura une étape supplémentaire pour le LAN

## Installation

```
apt-get install unbound && unboud-host && unbound-anchor && dnssec-trigger
```

## Organisation

```
>~ tree -A -L 3  /etc/unbound/
/etc/unbound/
├── [       332]  unbound.conf
├── [      4096]  unbound.conf.d
│   └── [       190]  root-auto-trust-anchor-file.conf
├── [      1277]  unbound_control.key
├── [       802]  unbound_control.pem
├── [      1277]  unbound_server.key
└── [       790]  unbound_server.pem

1 directory, 6 files
```

Unbound.conf

```
# The following line will configure unbound to perform cryptographic
# DNSSEC validation using the root trust anchor.
server:
    interface: 127.0.0.1
    auto-trust-anchor-file: "/var/lib/unbound/root.key"
```

/var/lib/root.key contient :

```
; autotrust trust anchor file
;;id: . 1
;;last_queried: 1465504617 ;;Thu Jun  9 22:36:57 2016
;;last_success: 1465504617 ;;Thu Jun  9 22:36:57 2016
;;next_probe_time: 1465546194 ;;Fri Jun 10 10:09:54 2016
;;query_failed: 0
;;query_interval: 43200
;;retry_time: 8640
.       172800  IN      DNSKEY  257 3 8
AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjFFVQUTf6v58fLjwBd0YI0Ezr
AcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkj
f5/Efucp2gaDX6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpzW5hOA2hzCTMjJ
PJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relSQageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLK
BP1dfwhYB4N7knNnulqQxA+Uk1ihz0= ;{id = 19036 (ksk), size = 2048b} ;;state=2 [  VALID  ]
;;count=0 ;;lastchange=1465490355 ;;Thu Jun  9 18:39:15 2016
```

Que l'on peuple avec cette commande

```
# unbound-anchor -a "/var/lib/unbound/root.key"
```

/etc/unbound/unbound.conf.d/root-auto-trust-anchor-file.conf

```
server:
    # The following line will configure unbound to perform cryptographic
    # DNSSEC validation using the root trust anchor.
    auto-trust-anchor-file: "/var/lib/unbound/root.key"
```

## Démarrer et tester le service

```
# /etc/init.d/unbound status
● unbound.service - (null)
   Loaded: loaded (/etc/init.d/unbound)
  Drop-In: /run/systemd/generator/unbound.service.d
           └─50-insserv.conf-$named.conf, 50-unbound-$named.conf
   Active: active (running) since sam. 2016-06-11 11:17:35 CEST; 10min ago
  Process: 22073 ExecStop=/etc/init.d/unbound stop (code=exited, status=0/SUCCESS)
  Process: 22082 ExecStart=/etc/init.d/unbound start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/unbound.service
           └─22094 /usr/sbin/unbound

juin 11 11:17:35 gally unbound-anchor[22090]: /var/lib/unbound/root.key has content
juin 11 11:17:35 gally unbound[22082]: Starting recursive DNS server: unbound.
juin 11 11:17:35 gally unbound[22094]: [22094:0] notice: init module 0: validator
juin 11 11:17:35 gally unbound[22094]: [22094:0] notice: init module 1: iterator
juin 11 11:17:35 gally unbound[22094]: [22094:0] info: start of service (unbound 1.4.22).
```

## *Test récursivité*

```
>~ dig @127.0.0.1 AAAA ordinatous.com

; <<>> DiG 9.9.5-9+deb8u6-Debian <<>> @127.0.0.1 AAAA ordinatous.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52593
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ordinatous.com.                        IN      AAAA

;; AUTHORITY SECTION:
ordinatous.com.         164     IN      SOA     dns108.ovh.net. tech.ovh.net. 2016060801 86400 3600
3600000 300

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Jun 09 23:00:29 CEST 2016
;; MSG SIZE  rcvd: 98
```

## *Bonus*

La configuration DNSSEC pour BIND

```
options {
        // N'écouter que sur l'interface locale. Autrement, faites
        // attention à interdire l'accès aux machines non-locales, pour
        // ne pas faire un résolveur ouvert.
      listen-on {127.0.0.1;};
      dnssec-enable yes;
      dnssec-validation yes;
};
trusted-keys {
        "."  (fichier bind.keys)
};
```

## *DNSSEC-TRIGGER*

Cas pour un usage nomade

Commande  de configuration

dnssec-trigger-control-setup

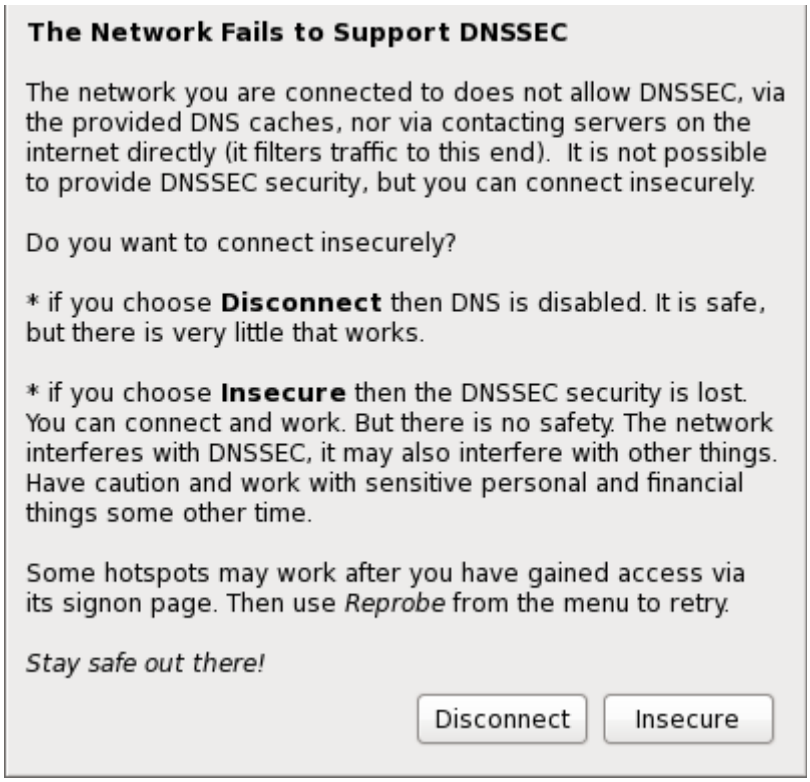
```
# dnssec-trigger-control-setup
setup in directory /etc/dnssec-trigger
dnssec_trigger_server.key exists
dnssec_trigger_control.key exists
create dnssec_trigger_server.pem (self signed certificate)
create dnssec_trigger_control.pem (signed client certificate)
Signature ok
subject=/CN=dnssec-trigger-control
Getting CA Private Key
Setup success. Certificates created.

run this script again with -i to:
        - enable remote-control in unbound.conf
        - start unbound-control-setup
        - add root trust anchor to unbound.conf
if you have not done this already
```

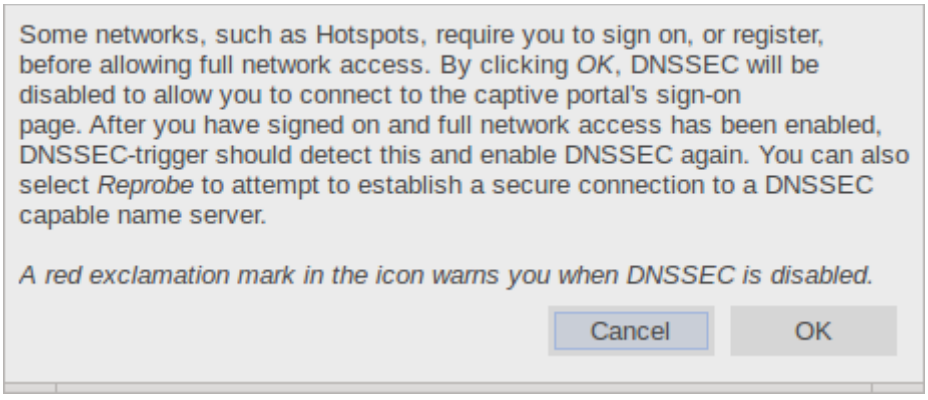
```
# dnssec-trigger-control-setup -i
setup in directory /etc/dnssec-trigger
unbound-checkconf: no errors in /etc/unbound/unbound.conf
checking if unbound-control needs to be enabled
checking if root trust anchor needs to be enabled
check for search path in resolv.conf and edit /etc/dnssec-trigger/dnssec-trigger.conf
check for domain in resolv.conf and edit /etc/dnssec-trigger/dnssec-trigger.conf
```

Démarrer l'interface graphique

dnssec-trigger-panel

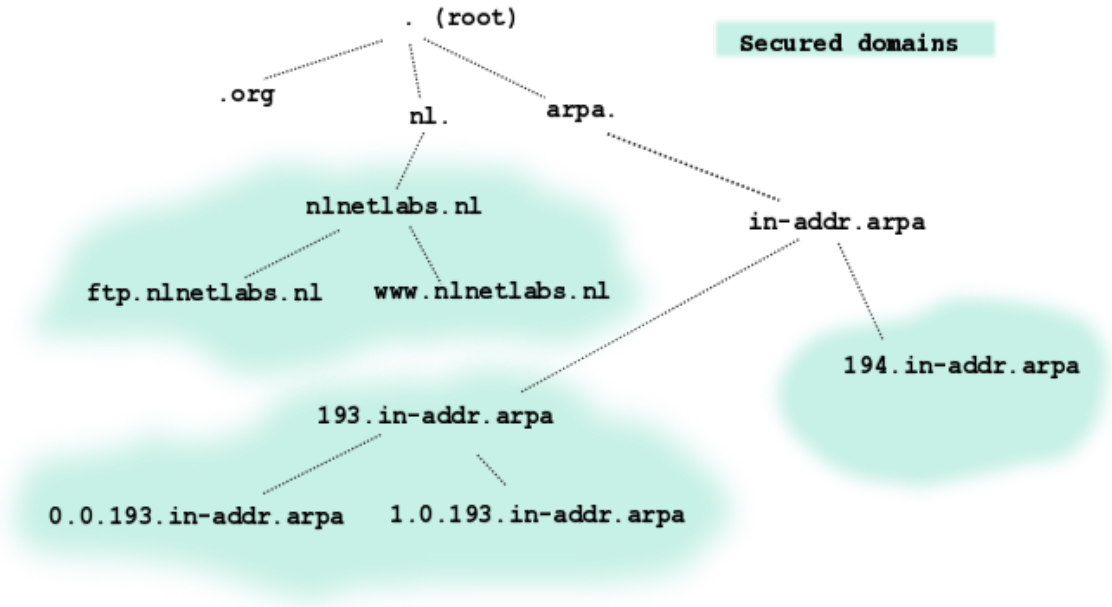
Le logiciel vous indiquera si la chaine de DNS est fiable ou non, et vous laisse le choix d'accepter ou non

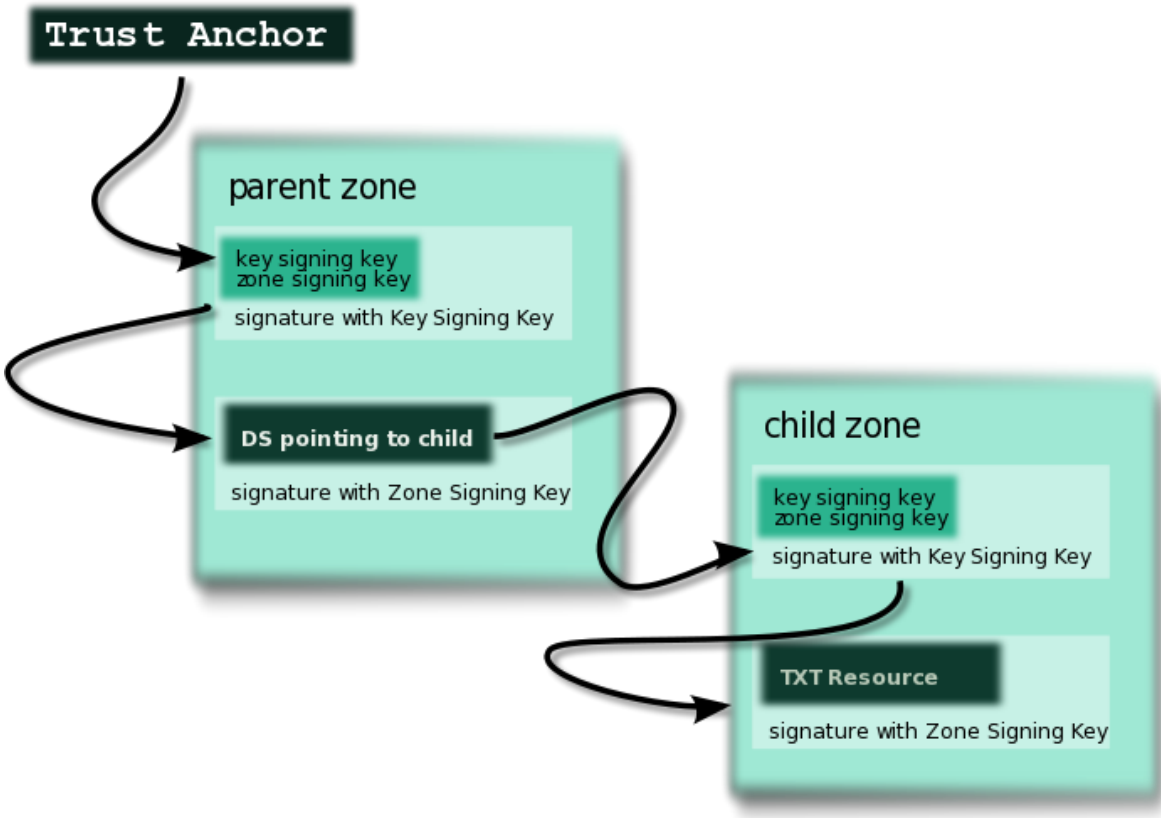


### *Anchor-trust*

Ce sont des zones qui garantissent leur fiabilité via l'échange de clé asymétrique, c'est une méthode censé être très fiable, très protocolaire. L'échange peut se faire de manière très confidentielle, avec un support amovible.



## *L'IANNA*

Fourni la liste ici : [Trust anchor & key]

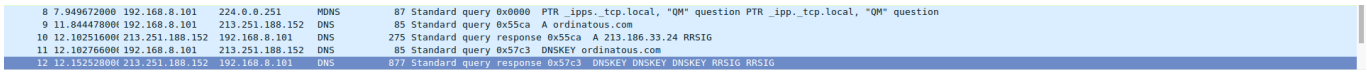
Ainsi que la synthaxe et les règles d'édition.

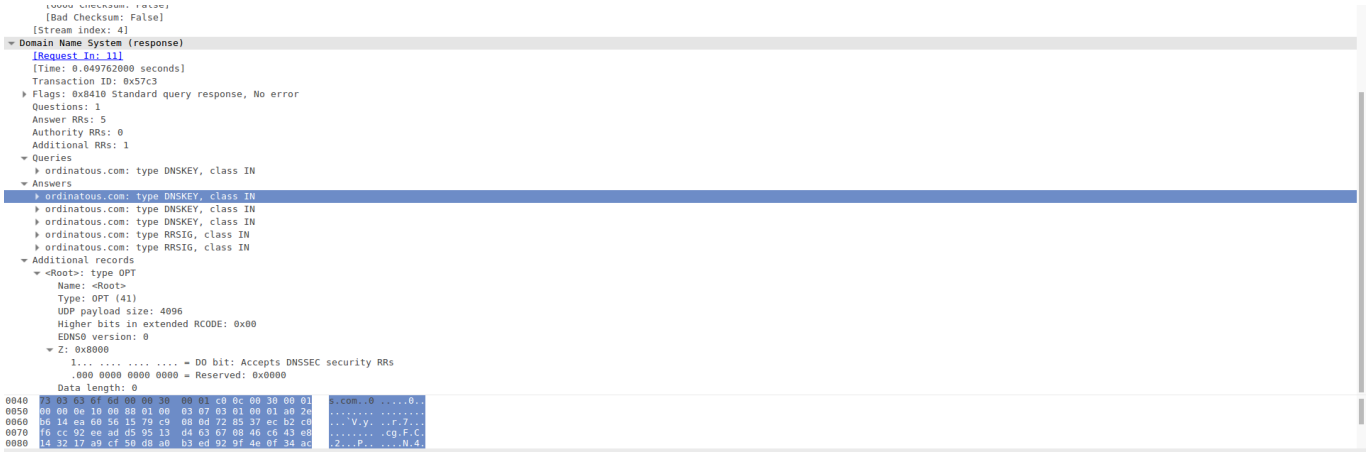
:https://data.iana.org/root-anchors/

## Observation wireshark

DNSSEC introduit de nouveau enregistrement comme :

- DS
- DNSKEY
- NSEC
- RRSIG




## Resolv.conf

Eventuellement il se pourrait que DHCP vienne ré écrire le fichier :

/etc/resolv.conf

Normalment DNSSEC-TRIGGER est là pour ça mais...

Dans ce cas installer resolvconf et modifier :

/etc/resolvconf/resolv.conf.d/head

Et indiquer ceci :

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
127.0.0.1
```

***Pour un LAN***

Unbound.conf

```
## Simple recursive caching DNS
## unbound.conf – chezmoi.lan
#
Server:
#repond sur toutes les interfaces
          Interface: 0.0.0.0
#plage d'adresse clients allow ou deny
 #       access-control: 10.0.0.0/16 allow
 #       access-control: 127.0.0.0/8 allow
        access-control: 192.168.1.0/24 allow
        verbosity: 1

#forward-zone:
        name: "."
        forward-addr :  80.67.169.12            #FDN
        forward-addr :  80.67.169.40            #FDN
#       forward-addr: 8.8.4.4                   # Google
#       forward-addr: 8.8.8.8                   # Google
#       forward-addr: 37.235.1.174              # FreeDNS
#       forward-addr: 37.235.1.177              # FreeDNS
#       forward-addr: 50.116.23.211             # OpenNIC
#       forward-addr: 64.6.64.6                 # Verisign
#       forward-addr: 64.6.65.6                 # Verisign
#       forward-addr: 74.82.42.42               # Hurricane Electric
#       forward-addr: 84.200.69.80              # DNS Watch
#       forward-addr: 84.200.70.40               # DNS Watch
#       forward-addr: 91.239.100.100            # censurfridns.dk
#       forward-addr: 109.69.8.51               # puntCAT
#       forward-addr: 208.67.222.220            # OpenDNS
#       forward-addr: 208.67.222.222            # OpenDNS
#       forward-addr: 216.146.35.35             # Dyn Public
#       forward-addr: 216.146.36.36             # Dyn Public
```

Un example est diponible ici :

/usr/share/doc/unbound/examples/unbound.conf

Le fichier est très grand, souvenez-vous ; utiliser un PAGER pour afficher du texte, il est là pour ça,

Un EDITOR pour editer, et cat est un outil pour assembler du texte.

## References

*Debian*

Debian_hand_book

Linuxsecurity

linux-administrator_guide

Manpages-fr

*Linux*

*Linux-france*

Wireshark tshark

*DNS*

Bind9

Unbound

ChaosComputerClub