Commandes et outils trafic Dns avec Wireshark

Pour générer du trafic DNS à analyser avec Wireshark, on peut utiliser plusieurs commandes et outils sur une machine Debian. Voici une série d'exemples pratiques pour générer du trafic DNS. Ces exemples couvrent des requêtes DNS basiques, avancées et des interactions entre un client et un serveur DNS.

Préparation

1. Assurez-vous que le serveur DNS est opérationnel:

   - Si tu utilises Bind9 sur Debian comme serveur DNS, il doit être en cours d'exécution :

     sudo systemctl status bind9

2. installer `dnsutils : Ce paquet contient des outils DNS comme `dig` et `nslookup` :

   sudo apt update

   sudo apt install dnsutils -y

Capturer le Trafic DNS avec Wireshark

1. Lancer Wireshark et sélectionner l'interface réseau appropriée.

2. Utiliser un filtre de capture pour le trafic HTTP.
Par exemple :

   dns

3. Démarre la capture et effectue les commandes DNS mentionnées.

4. Arrête la capture après avoir généré suffisamment de trafic et analyse les paquets HTTP capturés.

Exemples de Commandes DNS

Ces commandes permettent d'interroger les serveurs DNS pour obtenir diverses informations sur les noms de domaine et les adresses IP. Les commandes `dig` offrent des options avancées pour des requêtes détaillées, tandis que `nslookup` et `host` sont plus simples à utiliser pour des requêtes de base.

1. Requêtes DNS de Base

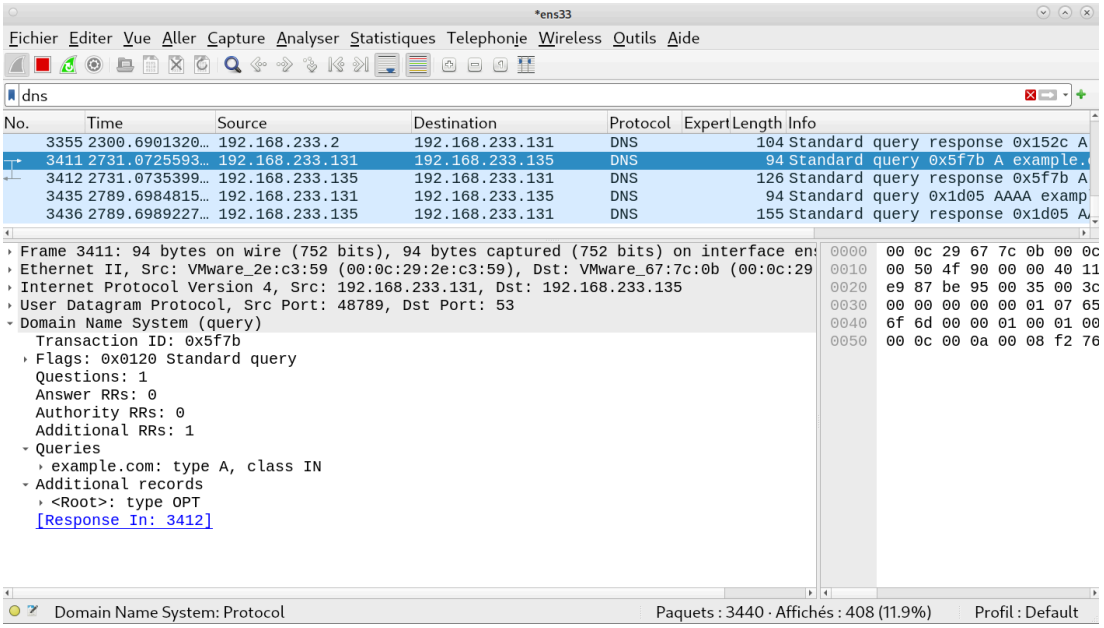
- Requête A (Adresse IPv4) avec `dig` :

dig example.com @192.168.233.135`

- ***`dig`*** *: Utilitaire pour interroger des serveurs DNS.*
- ***`example.com`*** *: Nom de domaine pour lequel on souhaite obtenir l'adresse IPv4.*
- ***`@192.168.233.135`*** *: Adresse IP du serveur DNS à interroger.*

Fonctionnement : Cette commande envoie une requête DNS pour obtenir l'enregistrement de type A (adresse IPv4) pour `example.com`*.*

```
dome@debian12AI:~$ sudo dig example.com @192.168.233.135

; <<>> DiG 9.18.24-1-Debian <<>> example.com @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24443
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: f27655c45a7c375b010000006671f7fe7b2920ce14893368 (good)
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            604800  IN      A       192.168.233.135

;; Query time: 0 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Tue Jun 18 23:11:26 CEST 2024
;; MSG SIZE  rcvd: 84

dome@debian12AI:~$
```



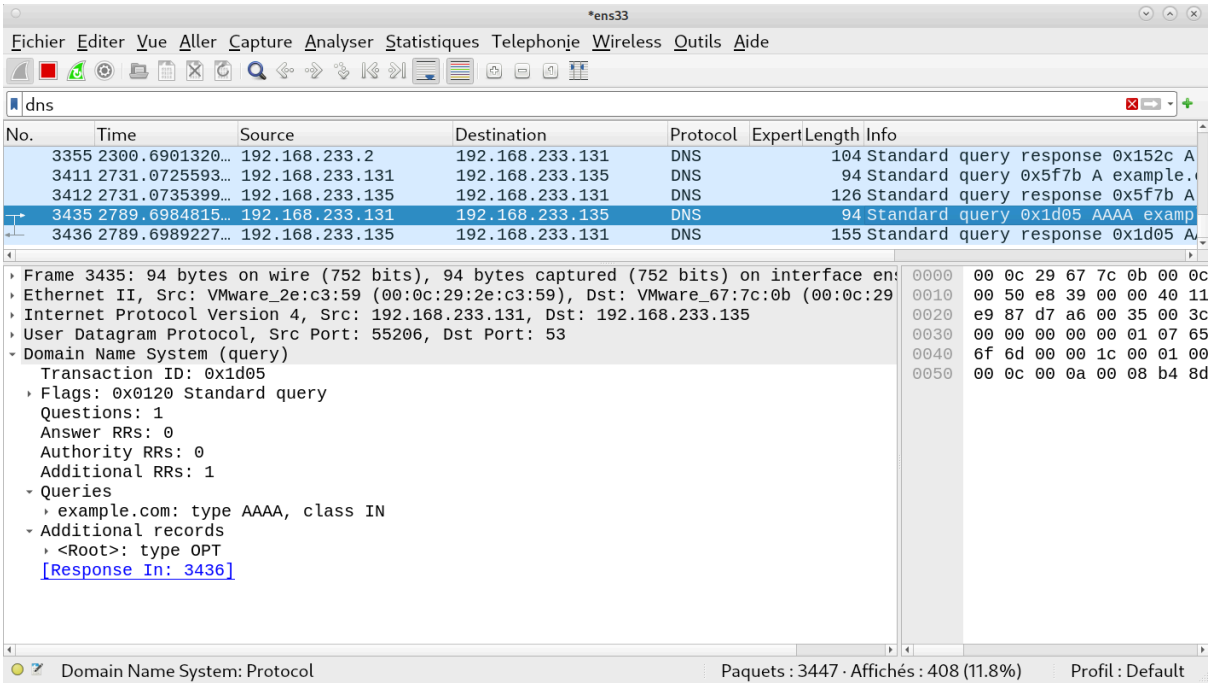
- Requête AAAA (Adresse IPv6) avec `dig` :

dig example.com AAAA @192.168.233.135

- AAAA : Spécifie que nous voulons une adresse IPv6.

Fonctionnement : Similaire à la requête A, mais pour une adresse IPv6.

```
dome@debian12AI:~$ dig example.com AAAA @192.168.233.135

; <<>> DiG 9.18.24-1-Debian <<>> example.com AAAA @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7429
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: b48deeca49ef5000010000006671f83948a17e9e0fbe2095 (good)
;; QUESTION SECTION:
;example.com.                   IN      AAAA

;; AUTHORITY SECTION:
example.com.            604800  IN      SOA     ns1.example.com. root.example.com. 2 6
04800 86400 2419200 604800

;; Query time: 0 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Tue Jun 18 23:12:25 CEST 2024
;; MSG SIZE  rcvd: 113
```



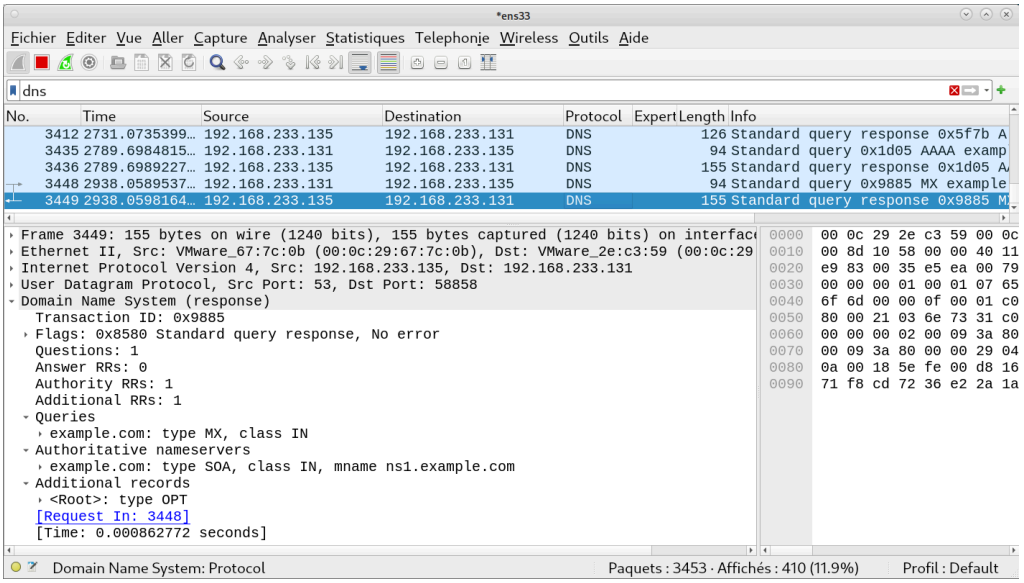
- Requête MX (Mail Exchange) avec `dig` :

dig example.com MX @192.168.233.135

- **`MX` : Spécifie une requête pour les enregistrements Mail Exchange.**

***Fonctionnement* : Retourne les serveurs de messagerie pour le domaine `example.com`.**

```
dome@debian12AI:~$ dig example.com MX @192.168.233.135

; <<>> DiG 9.18.24-1-Debian <<>> example.com MX @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39045
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 5efe00d8167369110100000006671f8cd7236e22a1a668a6a (good)
;; QUESTION SECTION:
;example.com.                   IN      MX

;; AUTHORITY SECTION:
example.com.            604800  IN      SOA     ns1.example.com. root.example.com. 2 6
04800 86400 2419200 604800

;; Query time: 0 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Tue Jun 18 23:14:53 CEST 2024
;; MSG SIZE  rcvd: 113
```



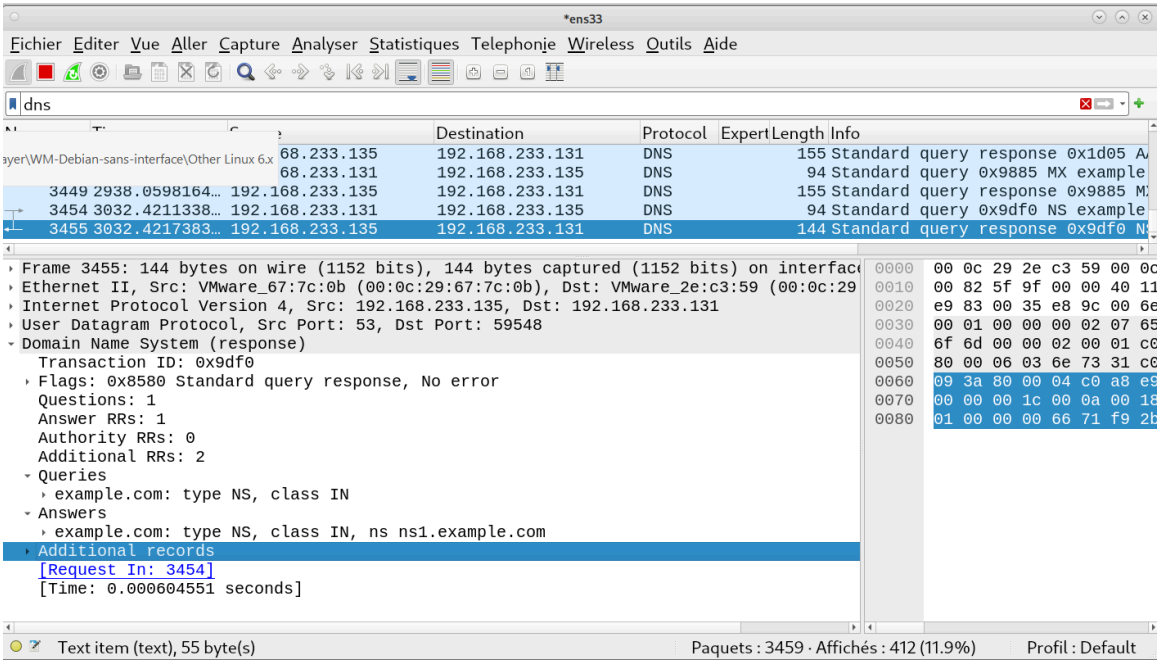
- Requête NS (Serveur de Noms) avec `dig` :

dig example.com NS @192.168.233.135

- NS : Spécifie une requête pour les enregistrements de serveurs de noms.

Fonctionnement : Retourne les serveurs DNS qui sont autoritaires pour `example.com`.

```
dome@debian12AI:~$ dig example.com NS @192.168.233.135

; <<>> DiG 9.18.24-1-Debian <<>> example.com NS @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40432
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: ae95d3b7e44746cc010000006671f92b43086f8ff8a31111 (good)
;; QUESTION SECTION:
;example.com.                   IN      NS

;; ANSWER SECTION:
example.com.            604800  IN      NS      ns1.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.        604800  IN      A       192.168.233.135

;; Query time: 3 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Tue Jun 18 23:16:27 CEST 2024
;; MSG SIZE  rcvd: 102

dome@debian12AI:~$
```



- Requête TXT (Texte) avec `dig` :

dig example.com TXT @192.168.233.135

- `TXT` : Spécifie une requête pour les enregistrements TXT.

Fonctionnement : Retourne les enregistrements de texte associés à `example.com`, souvent utilisés pour des informations sur le domaine ou des configurations SPF.

```
dome@debian12AI:~$ dig example.com TXT @192.168.233.135

; <<>> DiG 9.18.24-1-Debian <<>> example.com TXT @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24312
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 837fb1cac9df1d8d010000006671f9a79119190fff505f84 (good)
;; QUESTION SECTION:
;example.com.                   IN      TXT

;; AUTHORITY SECTION:
example.com.            604800  IN      SOA     ns1.example.com. root.example.com. 2 604800 86400 2419200 6
04800

;; Query time: 0 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Tue Jun 18 23:18:30 CEST 2024
;; MSG SIZE  rcvd: 113
```

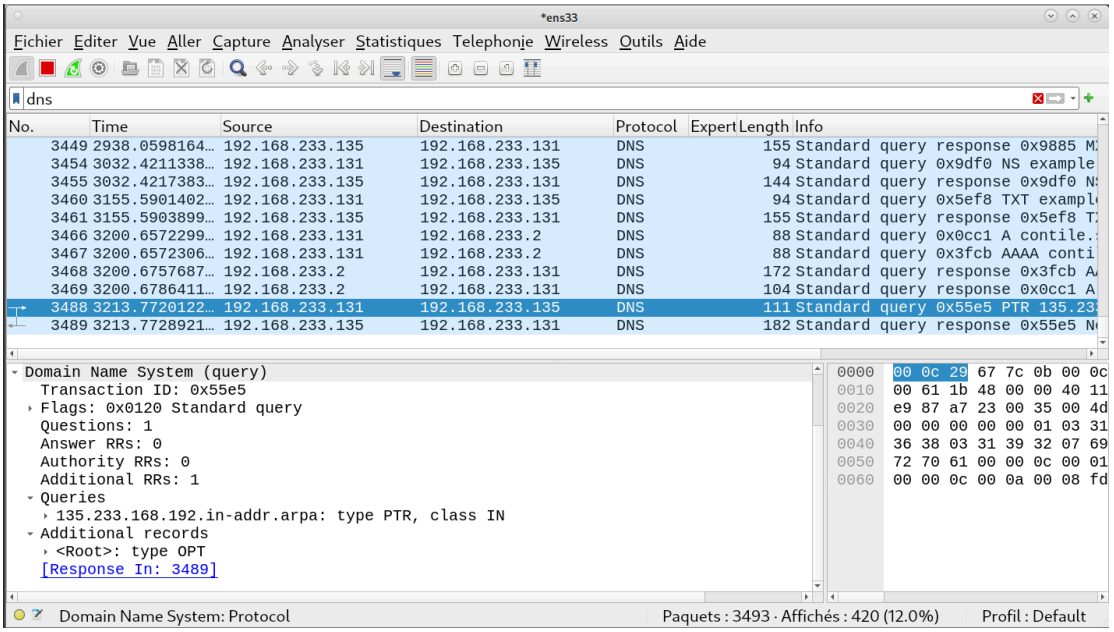
- Requête PTR (Réversée) avec `dig` :

dig -x 192.168.233.135 @192.168.233.135

- `-x` : Indique une requête inverse pour trouver le nom de domaine associé à une adresse IP.

Fonctionnement : Interroge le DNS pour savoir quel nom de domaine est associé à l'adresse IP `192.168.233.135`.

```
dome@debian12AI:~$ dig -x 192.168.233.135 @192.168.233.135

; <<>> DiG 9.18.24-1-Debian <<>> -x 192.168.233.135 @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 21989
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: fd995df9d785a3fd010000006671f9e19fb28915fb0705ff (good)
;; QUESTION SECTION:
;135.233.168.192.in-addr.arpa.  IN      PTR

;; AUTHORITY SECTION:
168.192.IN-ADDR.ARPA.   86400   IN      SOA     168.192.IN-ADDR.ARPA. . 0 28800 7200 604800 86400

;; Query time: 3 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Tue Jun 18 23:19:29 CEST 2024
;; MSG SIZE  rcvd: 140
```



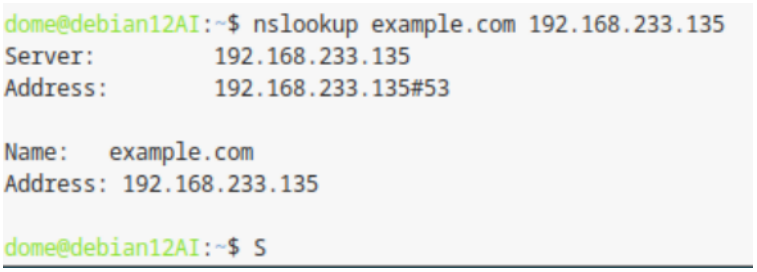
2. Utilisation de `nslookup`

- Requête de base avec `nslookup` :

nslookup example.com 192.168.233.135

- `nslookup` : Utilitaire pour effectuer des recherches DNS.
- `example.com` : Domaine à interroger.
- `192.168.233.135` : Serveur DNS à utiliser pour la requête.

Fonctionnement : Envoie une requête pour obtenir l'enregistrement de type A pour `example.com`.

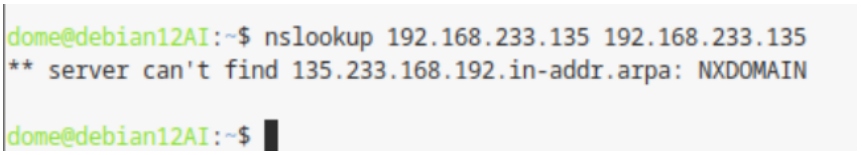


- Requête inverse avec `nslookup`:

nslookup 192.168.233.135 192.168.233.135

- `192.168.233.135` : IP pour laquelle nous voulons effectuer une recherche inverse.

Fonctionnement : Retourne le nom de domaine associé à l'adresse IP spécifiée.



- Requête avec serveur DNS alternatif :

nslookup example.com 8.8.8.8

- `8.8.8.8` : Utilise le serveur DNS de Google pour la requête.

Fonctionnement : Interroge le serveur DNS de Google pour obtenir l'enregistrement de type A pour `example.com`.

```
dome@debian12AI:~$ nslookup example.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   example.com
Address: 93.184.215.14
Name:   example.com
Address: 2606:2800:21f:cb07:6820:80da:af6b:8b2c

dome@debian12AI:~$
```

3. Requêtes avec `host`

- Requête A avec `host` :

host example.com 192.168.233.135

- `host` : Utilitaire simple pour rechercher des noms DNS.
- `example.com` : Domaine à interroger.
- `192.168.233.135` : Serveur DNS à utiliser.

Fonctionnement : Demande l'enregistrement de type A pour `example.com`.

```
dome@debian12AI:~$ host example.com 192.168.233.135
Using domain server:
Name: 192.168.233.135
Address: 192.168.233.135#53
Aliases:

example.com has address 192.168.233.135
```

- Requête MX avec `host :

host -t MX example.com 192.168.233.135

- `-t MX` : Spécifie que nous voulons les enregistrements de type Mail Exchange.

Fonctionnement : Retourne les enregistrements MX pour `example.com`.

```
dome@debian12AI:~$ host -t MX example.com 192.168.233.135
Using domain server:
Name: 192.168.233.135
Address: 192.168.233.135#53
Aliases:

example.com has no MX record
dome@debian12AI:~$
```

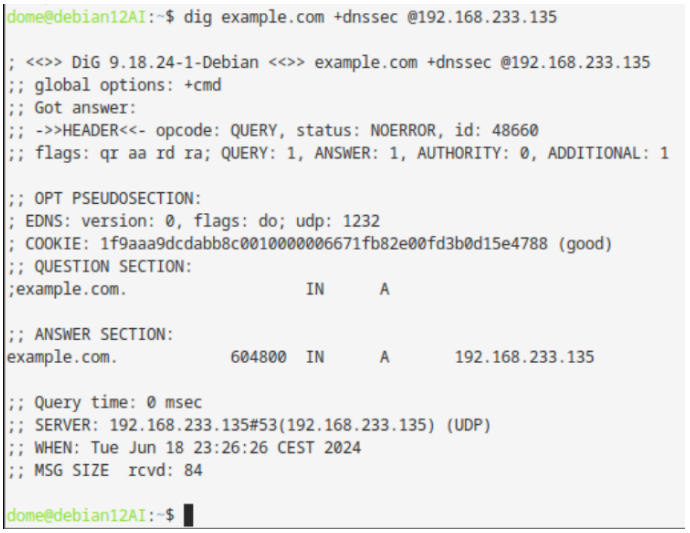
4. Requêtes Avancées avec `dig`

- Requête DNSSEC (vérifier les signatures DNS) :

dig example.com +dnssec @192.168.233.135

- **`+dnssec`** : Active la validation DNSSEC dans la requête.

Fonctionnement : Demande l'enregistrement A avec la vérification des signatures DNSSEC, fournissant des informations supplémentaires sur la sécurité DNS.



- Obtenir tous les enregistrements de zone (AXFR) :

dig example.com AXFR @192.168.233.135

- `AXFR` : Demande un transfert de zone.

Fonctionnement : Tente d'obtenir une copie complète de la zone DNS pour `example.com` (si le serveur le permet).

Note : Le transfert de zone (AXFR) est souvent restreint pour des raisons de sécurité.

et doit être activé et permis par le serveur DNS pour fonctionner.

```
dome@debian12AI:~$ dig example.com AXFR @192.168.233.135

; <<>> DiG 9.18.24-1-Debian <<>> example.com AXFR @192.168.233.135
;; global options: +cmd
example.com.            604800  IN      SOA     ns1.example.com. root.example.com. 2 604800 86400 2419200 6
04800
example.com.            604800  IN      NS      ns1.example.com.
example.com.            604800  IN      A       192.168.233.135
ns1.example.com.        604800  IN      A       192.168.233.135
www.example.com.        604800  IN      A       192.168.233.135
example.com.            604800  IN      SOA     ns1.example.com. root.example.com. 2 604800 86400 2419200 6
04800
;; Query time: 3 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (TCP)
;; WHEN: Tue Jun 18 23:27:09 CEST 2024
;; XFR size: 6 records (messages 1, bytes 215)

dome@debian12AI:~$
```

- Requête SOA (Start of Authority) :

dig example.com SOA @192.168.233.135

- `SOA` : Demande l'enregistrement de début d'autorité.

Fonctionnement : Retourne des informations sur le serveur principal pour la zone DNS de `example.com`, y compris le numéro de série de la zone et les informations de contact administratives.

```
dome@debian12AI:~$ dig example.com SOA @192.168.233.135

; <<>> DiG 9.18.24-1-Debian <<>> example.com SOA @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3192
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 8e383cff4acf412c010000006671fbec5e59148845671abe (good)
;; QUESTION SECTION:
;example.com.                   IN      SOA

;; ANSWER SECTION:
example.com.            604800  IN      SOA     ns1.example.com. root.example.com. 2 604800 86400 2419200 6
04800

;; Query time: 0 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Tue Jun 18 23:28:12 CEST 2024
;; MSG SIZE  rcvd: 113

dome@debian12AI:~$
```

5. Générer du Trafic via Navigateur

- Accéder à un site web via le navigateur:

Ouvre un navigateur sur la machine Debian et accède à une URL, par exemple `http://example.com`. Ceci générera du trafic DNS pour la résolution du nom de domaine.

- Changer le serveur DNS utilisé par le navigateur :

Modifier les paramètres réseau pour utiliser `192.168.233.135` comme serveur DNS préféré et accéder à des sites web pour générer du trafic DNS.

6. Utilisation de Scripts pour Générer du Trafic DNS

Tu peux aussi automatiser des requêtes DNS en utilisant des scripts Shell pour générer du trafic régulièrement.

#!/bin/bash

while true; do

dig example.com @192.168.233.135

sleep 5

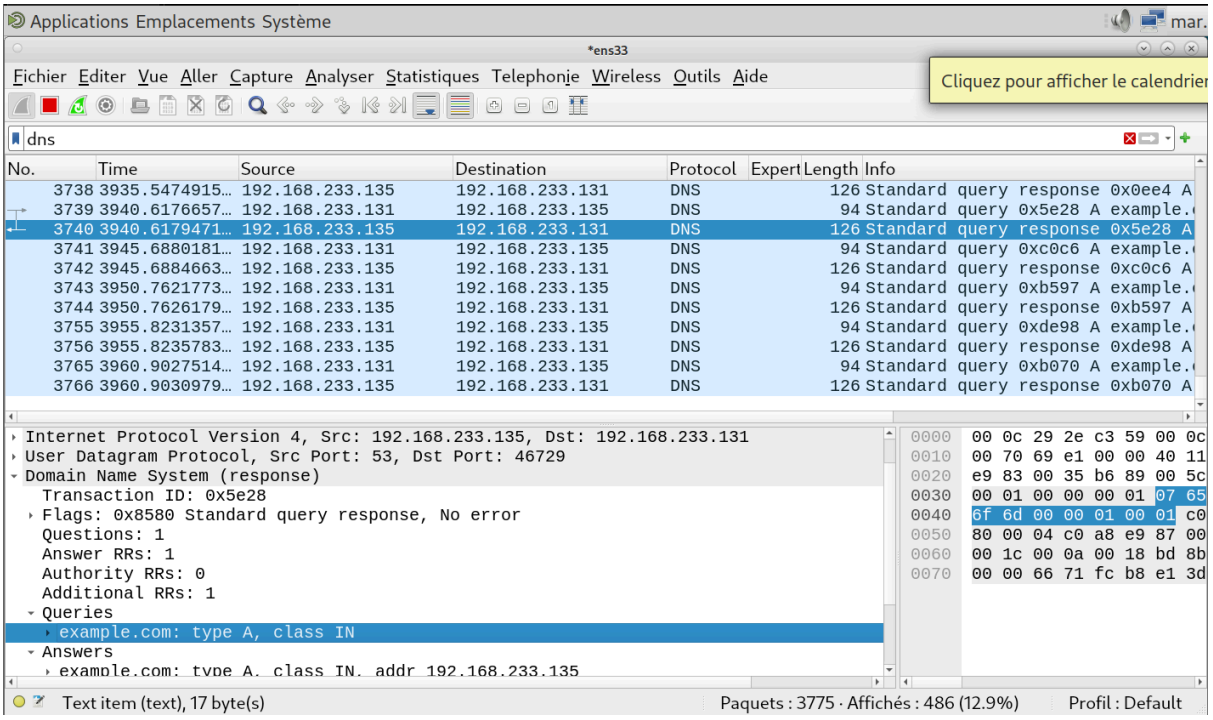
done

- Sauvegarde ce script dans un fichier (`dns_traffic.sh`), rends-le exécutable, et exécute-le :

chmod +x dns_traffic.sh

./dns_traffic.sh

```
dome@debian12AI:~/Bureau$ ./dns_traffic.sh

; <<>> DiG 9.18.24-1-Debian <<>> example.com @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30854
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 740cd659975af814010000006671fca30980a56d96af18ad (good)
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            604800  IN      A       192.168.233.135

;; Query time: 0 msec
;; SERVER: 192.168.233.135#53(192.168.233.135) (UDP)
;; WHEN: Tue Jun 18 23:31:15 CEST 2024
;; MSG SIZE  rcvd: 84


; <<>> DiG 9.18.24-1-Debian <<>> example.com @192.168.233.135
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62336
```


Applications Emplacements Système
mar.
*ens33
Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide
Cliquez pour afficher le calendrier
dns
No. Time Source Destination Protocol Expert Length Info
3738 3935.5474915… 192.168.233.135 192.168.233.131 DNS 126 Standard query response 0x0ee4 A
3739 3940.6176657… 192.168.233.131 192.168.233.135 DNS 94 Standard query 0x5e28 A example.
3740 3940.6179471… 192.168.233.135 192.168.233.131 DNS 126 Standard query response 0x5e28 A
3741 3945.6880181… 192.168.233.131 192.168.233.135 DNS 94 Standard query 0xc0c6 A example.
3742 3945.6884663… 192.168.233.135 192.168.233.131 DNS 126 Standard query response 0xc0c6 A
3743 3950.7621773… 192.168.233.131 192.168.233.135 DNS 94 Standard query 0xb597 A example.
3744 3950.7626179… 192.168.233.135 192.168.233.131 DNS 126 Standard query response 0xb597 A
3755 3955.8231357… 192.168.233.131 192.168.233.135 DNS 94 Standard query 0xde98 A example.
3756 3955.8235783… 192.168.233.135 192.168.233.131 DNS 126 Standard query response 0xde98 A
3765 3960.9027514… 192.168.233.131 192.168.233.135 DNS 94 Standard query 0xb070 A example.
3766 3960.9030979… 192.168.233.135 192.168.233.131 DNS 126 Standard query response 0xb070 A
Internet Protocol Version 4, Src: 192.168.233.135, Dst: 192.168.233.131
User Datagram Protocol, Src Port: 53, Dst Port: 46729
Domain Name System (response)
Transaction ID: 0x5e28
Flags: 0x8580 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 1
Queries
example.com: type A, class IN
Answers
example.com: type A, class IN, addr 192.168.233.135
0000 00 0c 29 2e c3 59 00 0c
0010 00 70 69 e1 00 00 40 11
0020 e9 83 00 35 b6 89 00 5c
0030 00 01 00 00 00 01 07 65
0040 6f 6d 00 00 01 00 01 c0
0050 80 00 04 c0 a8 e9 87 00
0060 00 1c 00 0a 00 18 bd 8b
0070 00 00 66 71 fc b8 e1 3d
Text item (text), 17 byte(s)
Paquets : 3775 · Affichés : 486 (12.9%)
Profil : Default