SCR.3.2 TP 04 ⊥ :

Dynamic DNS

RFC 2136

bind9/Debian/IMUNES

https://wiki.debian.org/DDNS

Objectif. Mise à jour de zones DNS par les transactions transmises par un serveur DHCPv4.

1. Créer la topologie donnée en fin d'énoncé. Elle a déjà été créée l'an dernier sous le nom `dns1.imn` (`TP22.SCR.2.2`). On peut donc partir d'une copie de `dns1.imn` qu'on placera dans le répertoire `~/SCR.3.2/TP04/` sous le nom `ddns.imn`. La zone est `tp.scr`.
2. Dans ce TP, *osisris* va être lancé comme serveur DNS, et *isis* comme serveur DHCP. Les machines autres que *isis* et *osiris* qui n'ont pas d'adresse vont l'obtenir par *isis*.

   Copier les fichiers utilisés pour configurer *osiris* (`TP22.SCR.2.2`), dans le répertoire `~/SCR.3.2/TP04/`, en faisant préfixer leur nom par "`osiris`".

   Ne laisser dans les fichiers zones que les RRs concernant *osiris*. Ajouter les RRs pour *isis*. Tous les autres RRs de la zone vont être transmis par *isis*.
3. Configurer *isis* en serveur DHCPv4. Il alloue les adresses à partir de `192.168.1.11`. Il doit communiquer aux clients l'adresse du serveur DNS, ainsi que le nom du domaine. Placer les fichiers correspondants dans `~/SCR.3.2/TP04/`, en faisant préfixer leur nom par "`isis`".

I. Permissions.

Puisque les fichiers zones doivent être modifiés dès qu'un client DHCP obtient ou libère un bail, il faut regarder les permissions des différents répertoires.

1. Lancer le service DNS directement par le nom du dæmon, puis par `ps`, vérifier sous le nom de quel utilisateur le service est en train de tourner.
2. Arrêter, puis lancer le service DNS par `/etc/init.d/bind9 start` ou par `service bind9 start`, puis par `ps`, vérifier sous le nom de quel utilisateur le service est en train de tourner. Confirmer par la consultation du fichier `/etc/default/bind9` ou encore par `.../DOC/README.Debian` d'un nœud virtuel de imunes.
3. *Debian* place presque toute la configuration du service DNS dans `/etc/bind/`
   C'est donc un répertoire sensible sur lequel il faut restreindre les droits de modification. Consulter les droits de `/etc/bind/` et conclure ce qui se passera si le processus qui tourne le service cherche à écrire dans `/etc/bind/`
4. Lorsque *osiris* recevra de *isis* une première transaction concernant une zone, `named` créera un journal correspondant à la zone. C'est dans ce journal que toutes les transactions pour la zone vont être enregistrées par `named`. Par exemple, si le fichier zone en question est `db.tp.scr`, `named` créera un fichier `db.tp.scr.jnl`, dans le même répertoire où se trouve le fichier zone. Le répertoire en question doit donc être modifiable par l'utilisateur au nom duquel `named` tourne. Consulter alors `.../DOC/README.Debian` pour avoir la recommandation *debian* sur l'endroit où placer les fichiers zones lorsqu'on utilise DDNS. Noter le nom de ce répertoire.

II. Sécurité des transactions.

Afin de garantir l'authenticité de la transaction reçue par le DNS, RFC 2845 recommande l'utilisation de TSIG, *Transaction SIGnature*. C'est un mécanisme MAC : *Message Authentification Code* (une fonction hash à clé) utilisant une clé secrète partagée. Les données constituant la transaction ne sont pas chiffrées. L'émetteur utilise la clé pour produire un condensé du message, appelé alors *signature*. Le récepteur reçoit le message et la signature. Il applique la clé sur le message. Le message est considéré authentique si et seulement si le condensé calculé est le même que le condensé reçu. RFC 2845 ne fournit pas de recommandation sur le protocole à utiliser pour distribuer la clé secrète. Dans ce TP, comme les deux serveurs sont sur le même segment, sous l'administration d'une même personne, on considère qu'il suffit que `root` place la clé manuellement chez l'un et chez l'autre.

1. On va utiliser `tsig-keygen` en utilisant l'algorithme `hmac-md5`. La commande est en principe disponible sur la machine hôte. On est placé dans le répertoire `TP04/`, et on fait `tsig-keygen -a md5`, puis `tsig-keygen -a md5 tp.scr-key`, et on observe la différence (`man tsig-keygen`).
2. Relancer la dernière commande en envoyant le résultat dans le fichier de nom `ddns.key` Ce fichier ira dans `osiris:/etc/bind/` et dans `isis:/etc/dhcp/` mais pas tout de suite. Faisons durer le suspense.

III. Ajouter dans la configuration de *osiris* les directives relatives au DDNS.

Travailler la copie de `named.conf.local` qui est sur la machine hôte, répertoire `TP04/`
Si on a bien suivi les instructions au début, le nom de ce fichier commence par "`osiris`".

1. Placer au début une directive `include` afin d'y inclure le fichier qui contient la clé en indiquant le chemin complet. C'est le fichier créé en **II.** 2.
2. Dans chacune des deux déclarations *zone*, modifier le chemin vers le fichier zone en fonction de ce qui a été déduit de **I.** 4.
3. Dans chacune des deux déclarations *zone*, ajouter la clause `allow-update` en y indiquant le nom de la clé créée en **II.** 2.
4. Lancer l'exécution de `ddns.imn`, et compléter les fichiers correspondants dans *osiris* (`sudo hcp` ou copier-coller depuis les copies sur la machine hôte), sans oublier le fichier `ddns.key`.
5. Vérifier les permissions sur le fichier qui contient la clé. Ce fichier doit avoir comme propriétaire `root`, comme groupe le même nom que l'utilisateur trouvé en **I.** 2. Il doit être `rw` par `root`, seulement lisible par le groupe, et aucun droit pour les autres.
6. Tester par `named-checkconf` et `named-checkzone`.

IV. Ajouter dans la configuration de *isis* les directives relatives au DDNS.

Travailler la copie de `dhcpd.conf` qui est sur la machine hôte, répertoire `TP04/`
Si on a bien suivi les instructions au début, le nom de ce fichier commence par "`isis.`"

1. Placer au début une directive `include` afin d'y inclure le fichier qui contient la clé en indiquant le chemin complet. C'est le fichier créé en **II.** 2.
2. Le `ddns-update-style` est `standard` (au lieu de `none`).
   https://kb.isc.org/docs/aa-01091
3. Ajouter pour chacune des deux zones, une déclaration *zone*, en y incluant les clauses `primary` (qui donne l'adresse de *osiris*) et `key` qui donne le nom de la clé secrète partagée.
4. Ajouter dans la déclaration *subnet* ce qu'il faut pour communiquer aux clients DHCP, le nom du domaine et l'adresse de leur serveur DNS.

5. Vérifier que `isis` a bien une déclaration `host` avec son adresse fixe ; l'ajouter sinon.
6. Envoyer les fichiers `ddns.key` et `dhcpd.conf` à leur emplacement dans *isis* (`sudo hcp` ou copier-coller depuis les copies sur la machine hôte).
7. Vérifier les permissions sur le fichier qui contient la clé. Ce fichier doit avoir `root` comme propriétaire et groupe. Il doit être `rw` par `root`, seulement lisible par le groupe, et aucun droit pour les autres.
8. Tester le fichier de configuration (`dhcpd -t`).

V. Lancements des services.

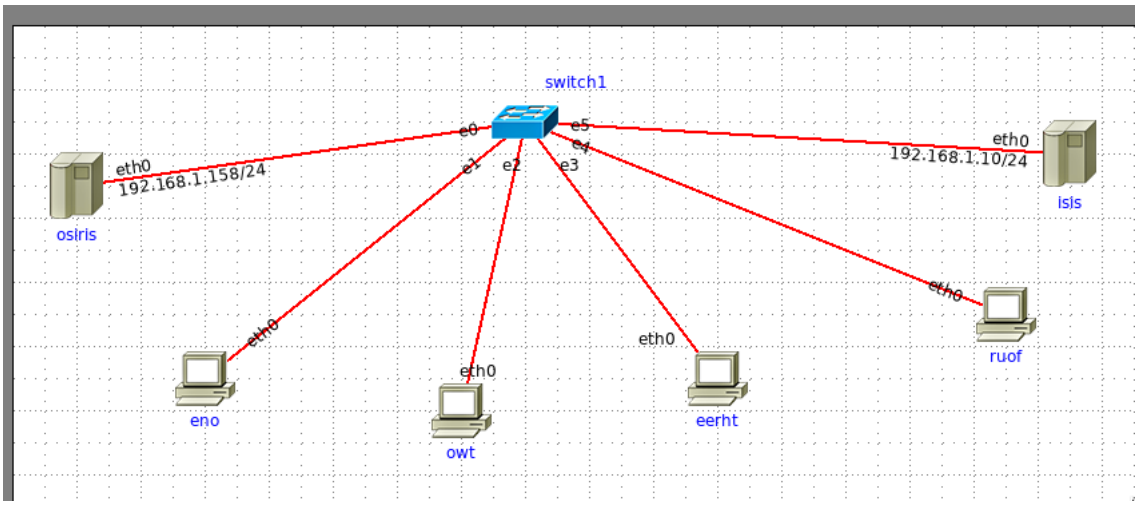
On ne va pas lancer par `/etc/init/... start`, parce qu'on ne verra pas l'activité des processus. On lancera donc directement par `named` et `dhcpd` respectivement. On s'assurera alors, en regardant les logs, que les chemins vers les fichiers de configuration sont corrects. Ceci parce qu'en général, ces chemins sont précisés dans les scripts de lancement des services mais ici, on ne veut pas utiliser de scripts de lancement.

1. Par man `dhcpd`, trouver quelle option donner à `dhcpd` pour qu'il envoie les logs sur `stderr` (tout en tournant au premier-plan). Lancer `dhcpd` avec l'option trouvée, et s'assurer dans le log affiché, que c'est bien le fichier de configuration `/etc/dhcp/dhcpd.conf` qui est utilisé. Si le processus réclame le fichier des baux, le créer par `touch`.
2. Trouver l'option de `named` qui permet de le lancer au premier-plan tout en envoyant tous les logs vers `stderr`. Il faudra la combiner avec l'option qui lance le processus sous le nom de l'utilisateur trouvé en **I.** 2.
   Lancer `named` avec les deux options, et s'assurer dans le log affiché, que c'est bien le fichier de configuration `/etc/bind/named.conf` qui est utilisé.
3. On s'apprête à lancer `dhclient` sur `eno`, **mais avant de le faire**, consulter le contenu du répertoire de *osiris* dont on a noté le nom en **I.** 4.
4. Sur `eno`, consulter le contenu du fichier `/etc/resolv.conf`, puis lancer la commande `dhclient -v eth0`, et consulter de nouveau le fichier `/etc/resolv.conf` de `eno`, ainsi que le contenu du répertoire dont il est question au point précédent. Expliquer.
5. Jeter un œil côté logs fournis par *osiris* et ceux fournis par *isis*. Les fichiers de zones ont-ils été mis à jour par `osiris` ? Expliquer.
   https://bind9.readthedocs.io/en/latest/chapter5.html
6. Sur `eno`, demander par `dig` l'adresse de `owt`. Expliquer.
7. Passer alors `dhclient -v eth0` sur `owt`, puis refaire le `dig` précédent.
8. Faire libérer son bail par `owt` et voir que la transaction correspondante est également réalisée, en refaisant, par exemple, un `dig` sur la machine `owt` devenue sans adresse.

La section suivante donne une illustration des résultats attendus agrémentée d'informations utiles.

Remarque. Si on veut dumper le journal sans attendre, on pourra utiliser `rndc sync <zone_name>`

Si tout fonctionne correctement, alors mettre à jour -si nécessaire- les fichiers sur la machine hôte, répertoire `TP04/`, en fonction des changements éventuels apportés aux fichiers sur les nœuds virtuels.



VI. Une illustration incrustée de quelques informations complémentaires.

```
root@eno:/# ip a
.......
eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc netem state UP ....
    link/ether 42:00:aa:00:00:01 brd ff:ff:ff:ff:ff:ff link-netnsid 0

root@eno:/# file /etc/resolv.conf
/etc/resolv.conf: empty

root@eno:/# dhclient eth0
root@eno:/# ip a

eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc netem state UP ....
    link/ether 42:00:aa:00:00:01 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.1.11/24 brd 192.168.1.255 scope global eth0

root@eno:/# cat /etc/resolv.conf
domain tp.scr  #Le domaine local est tp.scr
search tp.scr  #Les noms non totalement qualifiés seront recherchés dans le domaine tp.scr
               #Dans certaines installations, il peut être utile d'avoir une liste
               #de domaines ici. Le premier domaine de la liste est considéré comme
               #le domaine local. Les directives domain et search sont donc mutuellemnt
               #exclusives : c'est celle déclaré en second qui prédomine.
               #Ici, c'était rempli automatiquement suite à l'acquisition du bail.
nameserver 192.168.1.158
root@eno:/#

Les logs côté isis montrent :
DHCPDISCOVER from 42:00:aa:00:00:01 via eth0
DHCPOFFER on 192.168.1.11 to 42:00:aa:00:00:01 (eno) via eth0
DHCPREQUEST for 192.168.1.11 (192.168.1.10) from 42:00:aa:00:00:01 (eno) via eth0
DHCPACK on 192.168.1.11 to 42:00:aa:00:00:01 (eno) via eth0
Added new forward map from eno.tp.scr to 192.168.1.11
Added reverse map from 11.1.168.192.in-addr.arpa. to eno.tp.scr

Les logs côté osiris montrent :
23-May-2022 15:07:10.280 client 192.168.1.10#14893/key tp.scr-key: signer "tp.scr-key" approved
23-May-2022 15:07:10.280 client 192.168.1.10#14893/key tp.scr-key: updating zone 'tp.scr/IN': adding an RR at 'eno.tp.scr' A 192.168.1.11
23-May-2022 15:07:10.280 client 192.168.1.10#14893/key tp.scr-key: updating zone 'tp.scr/IN': adding an RR at 'eno.tp.scr'
                                                                    DHCID AAABcy4AjqW5ZWjIgMkVuDHBtx3RDcXvciZGU0WIgF81hnY=
23-May-2022 15:07:10.300 client 192.168.1.10#14893/key tp.scr-key: signer "tp.scr-key" approved
23-May-2022 15:07:10.300 client 192.168.1.10#14893/key tp.scr-key: updating zone '1.168.192.in-addr.arpa/IN': deleting rrset
                                                                                                  at '11.1.168.192.in-addr.arpa' PTR
23-May-2022 15:07:10.300 client 192.168.1.10#14893/key tp.scr-key: updating zone '1.168.192.in-addr.arpa/IN': adding an RR
                                                                                              at '11.1.168.192.in-addr.arpa' PTR eno.tp.scr.

root@owt:/# dhclient eth0

logs côté osiris montrent :
23-May-2022 15:10:25.114 client 192.168.1.10#14893/key tp.scr-key: signer "tp.scr-key" approved
23-May-2022 15:10:25.115 client 192.168.1.10#14893/key tp.scr-key: updating zone 'tp.scr/IN': adding an RR at 'owt.tp.scr' A 192.168.1.12
23-May-2022 15:10:25.115 client 192.168.1.10#14893/key tp.scr-key: updating zone 'tp.scr/IN': adding an RR at 'owt.tp.scr'
                                                                    DHCID AAABewHnvr1ZWVySqhPRDNxHocb7sftkuRPzm4YKDWomlF4=
23-May-2022 15:10:25.144 client 192.168.1.10#14893/key tp.scr-key: signer "tp.scr-key" approved
23-May-2022 15:10:25.144 client 192.168.1.10#14893/key tp.scr-key: updating zone '1.168.192.in-addr.arpa/IN': deleting rrset
                                                                                                  at '12.1.168.192.in-addr.arpa' PTR
23-May-2022 15:10:25.144 client 192.168.1.10#14893/key tp.scr-key: updating zone '1.168.192.in-addr.arpa/IN': adding an RR
                                                                                              at '12.1.168.192.in-addr.arpa' PTR owt.tp.scr.

root@osiris:/# less ..../db.tp.scr
; FORWARD tp.scr ZONE FILE

$TTL    4h

@       IN      SOA     osiris.tp.scr. root.tp.scr. (
                              2         ; Serial
```

```
                        604800         ; Refresh
                         86400         ; Retry
                       2419200         ; Expire
                        604800 )       ; Negative Cache TTL

@       IN      NS      osiris.tp.scr.

osiris  IN      A       192.168.1.158
isis    IN      A       192.168.1.10
db.tp.scr (END)
```

Pourtant le dig suivant fourni le résultat attendu :

```
root@eno:/# dig +search owt  #Par défaut, dig soumet le nom sans utiliser la liste search
                             #qui figure dans /etc/resolv.conf
                             #On peut faire dig owt.tp.scr ou dig +search owt
; <<>> DiG 9.10.3-P4-Debian <<>> owt.tp.scr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47791
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;owt.tp.scr.                    IN      A

;; ANSWER SECTION:
owt.tp.scr.             300     IN      A       192.168.1.12

;; AUTHORITY SECTION:
tp.scr.                 14400   IN      NS      osiris.tp.scr.


;; ADDITIONAL SECTION:
osiris.tp.scr.          14400   IN      A       192.168.1.158

;; Query time: 1 msec
;; SERVER: 192.168.1.158#53(192.168.1.158)
;; WHEN: Mon May 23 15:12:48 UTC 2022
;; MSG SIZE  rcvd: 92
```

Comme je doute de tout, j'arrête isis, je lui change la clé et je vois si je passe quand même :

```
root@isis:/# cd /etc/dhcp
root@isis:/etc/dhcp# mv ddns.key good.ddns.key

oot@isis:/etc/dhcp# tsig-keygen -a md5 tp.scr-key > ddns.key
root@isis:/etc/dhcp# chmod o-r ddns.key
```

Je relance isis, puis :

```
root@eerht:/# dhclient eth0


Les logs côté isis montrent :
DHCPDISCOVER from 42:00:aa:00:00:03 via eth0
DHCPOFFER on 192.168.1.13 to 42:00:aa:00:00:03 (eerht) via eth0
DHCPREQUEST for 192.168.1.13 (192.168.1.10) from 42:00:aa:00:00:03 (eerht) via eth0
DHCPACK on 192.168.1.13 to 42:00:aa:00:00:03 (eerht) via eth0
Unable to add forward map from eerht.tp.scr to 192.168.1.13: tsig indicates error

23-May-2022 15:20:00.303 client 192.168.1.10#37057: request has invalid signature: TSIG tp.scr-key: tsig verify failure (BADSIG)
```

eerht a donc bien acquis un bail mais la transaction avec osiris concernant eerht a echoué.

OK. C'est rassurant, mais j'insiste avec le dig suivant :

```
root@eno:/# dig eerht.tp.scr

; <<>> DiG 9.10.3-P4-Debian <<>> eerht.tp.scr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 56520
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;eerht.tp.scr.                  IN      A

;; AUTHORITY SECTION:
tp.scr.                 14400   IN      SOA     osiris.tp.scr. root.tp.scr. 4 604800 86400 2419200 604800

;; Query time: 1 msec
;; SERVER: 192.168.1.158#53(192.168.1.158)
;; WHEN: Mon May 23 15:23:51 UTC 2022
;; MSG SIZE  rcvd: 89
```

C'est cohérent : rien dans le dns concernant eerht.
eerht libère le bail, j'arrêtre isis, je lui remets la bonne clé et je le relance.

```
root@eerht:/# dhclient -r eth0
Killed old client process

root@isis:/etc/dhcp# mv good.ddns.key ddns.key
```

```
.....
root@eerht:/# dhclient eth0

Les logs côté isis montrent :
HCPDISCOVER from 42:00:aa:00:00:03 via eth0
DHCPOFFER on 192.168.1.13 to 42:00:aa:00:00:03 (eerht) via eth0
DHCPREQUEST for 192.168.1.13 (192.168.1.10) from 42:00:aa:00:00:03 (eerht) via eth0
DHCPACK on 192.168.1.13 to 42:00:aa:00:00:03 (eerht) via eth0
Added new forward map from eerht.tp.scr to 192.168.1.13
Added reverse map from 13.1.168.192.in-addr.arpa. to eerht.tp.scr

Les logs côté osiris montrent :
23-May-2022 15:27:22.651 client 192.168.1.10#7617/key tp.scr-key: signer "tp.scr-key" approved
23-May-2022 15:27:22.652 client 192.168.1.10#7617/key tp.scr-key: updating zone 'tp.scr/IN': adding an RR at 'eerht.tp.scr' A 192.168.1.13
23-May-2022 15:27:22.652 client 192.168.1.10#7617/key tp.scr-key: updating zone 'tp.scr/IN': adding an RR
                                                                  at 'eerht.tp.scr' DHCID AAABuntBRTqal2iKfJ4vKOMQV0CaSL88z9gSC32gkKwJ49o=
23-May-2022 15:27:22.700 client 192.168.1.10#7617/key tp.scr-key: signer "tp.scr-key" approved
23-May-2022 15:27:22.700 client 192.168.1.10#7617/key tp.scr-key: updating zone '1.168.192.in-addr.arpa/IN': deleting rrset
                                                                                                  at '13.1.168.192.in-addr.arpa' PTR
23-May-2022 15:27:22.701 client 192.168.1.10#7617/key tp.scr-key: updating zone '1.168.192.in-addr.arpa/IN': adding an RR
                                                                                                        at '13.1.168.192.in-addr.arpa' PTR eerht.tp.scr.


root@eno:/# dig +search eerht

; <<>> DiG 9.10.3-P4-Debian <<>> eerht.tp.scr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36731
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;eerht.tp.scr.                  IN      A

;; ANSWER SECTION:
eerht.tp.scr.           300     IN      A       192.168.1.13

;; AUTHORITY SECTION:
tp.scr.                 14400   IN      NS      osiris.tp.scr.

Entre-temps, des choses supplémentaires ont été dumpées du journal vers le fichier zone (pas par moi mais par le processus):

root@osiris:/# less /var/lib/bind/db.tp.scr
$ORIGIN .
$TTL 14400      ; 4 hours
tp.scr                  IN SOA  osiris.tp.scr. root.tp.scr. (
                                4          ; serial
                                604800     ; refresh (1 week)
                                86400      ; retry (1 day)
                                2419200    ; expire (4 weeks)
                                604800     ; minimum (1 week)
                                )
                        NS      osiris.tp.scr.
$ORIGIN tp.scr.
$TTL 300        ; 5 minutes
eno                     A       192.168.1.11
                        DHCID   ( AAABcy4AjqW5ZWjIgMkVuDHBtx3RDcXvciZGU0WIgF81
                                hnY= ) ; 48752 192 32
$TTL 14400      ; 4 hours
isis                    A       192.168.1.10
osiris                  A       192.168.1.158
$TTL 300        ; 5 minutes
owt                     A       192.168.1.12
                        DHCID   ( AAABewHnvr1ZWVySqhPRDNxHocb7sftkuRPzm4YKDWom
                                lF4= ) ; 48720 209 32
db.tp.scr (END)
```