Le principe de Master/Slave

Ramle, Eban

Le principe de Master/Slave

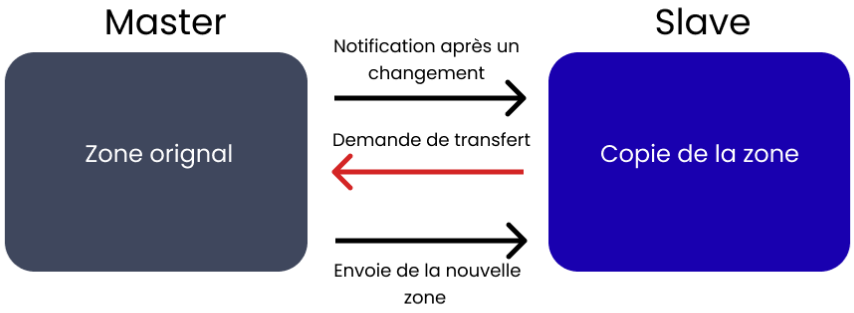
Le principe de master et de slave est assez important dans le monde du DNS et de l'informatique en général, il permet, dans le cadre du DNS, de redonder facilement une zone sur plusieurs serveurs autoritaires.

Le serveur dit `master` (maître) est celui qui contrôle la zone, il possède le fichier "original".

Dans certains contextes, comme pour éviter la corruption de tous les serveurs en cas d'erreur sur le maître, on utilise plusieurs serveurs masters, il n'y a plus une seule zone "original" mais plusieurs, le désavantage d'un tel système c'est qu'on est obligé de mettre à jour chaque serveurs master *manuellement*.

Le serveur dit `slave` (esclave) est celui qui reçoit la zone depuis un serveur master, ce transfert de zone DNS **utilise des protocoles comme AXFR ou IXFR.

Ce concept de master/slave permet de redonder une zone bien plus facilement que si on le faisait manuellement.



Source : https://blog.eban.bzh/today-i-learned/xfr-zone-transfer.html CC-BY-SA

Communication entre Master et Slave - Vulnérabilité transfert de zone

Eban, Ramle

2021-04-26

Communication entre Master et Slave - Vulnérabilité transfert de zone

Aujourd'hui, on parle des procotoles AXFR et IXFR ainsi que de la vulnérabilité que peut représenter un transfert de zone non autorisé !

AXFR est un protocole qui sert à faire des transferts de zone du `master` vers le `slave`, le `slave` vérifie périodiquement si le serial (=numéro de version, voir ici si vous ne vous rappelez plus de ce que c'est) de son SOA est inférieur à celui du `master`, si c'est le cas, il fait une demande de transfert de zone, qui peut être fait grâce au protocole AXFR. Concrètement le `master` envoie simplement une copie de sa zone DNS vers le `slave`.

Il existe aussi un autre protocole pour le même usage appelé IXFR, avec IXFR, si le `slave` remarque que son serial est inférieur à celui du `master`, il envoie le serial de la version de la zone qu'il détient au `master`, le `master` envoie ensuite le nouveau serial et deux listes, une pour les records à supprimer et une autre pour les records à ajouter/modifier. IXFR a pour avantage principal de nécessiter moins de bande passante qu'AXFR.

Un fois ces petites explications faites, passons à la partie la plus intéressante, l'exploitation d'un transfert de zone :p. Pour authentifier un slave on effectue souvent un filtrage par IP, néanmoins il arrive que ce filtrage ne soit pas mit en place, n'importe qui peut alors dumper (= avoir une copie) de la zone très facilement, voici un exemple que vous pouvez essayer de reproduire chez vous.

Nous allons dans un premier temps demander à notre serveur DNS résolveur la liste des serveurs DNS autoritaires pour le domaine `zonetransfer.me`. L'option `+short` que j'ai ajouté ici permet simplement d'avoir directement le résultat sans d'autres informations.

```
% dig zonetransfer.me NS +short
nsztm1.digi.ninja.
nsztm2.digi.ninja.
```

Nous avons donc ici les deux serveurs DNS autoritaires du nom de domaine `zonetransfer.me`. Essayons donc de faire une requête DNS de type AXFR sur un de ces deux serveurs.

```
% dig axfr @nsztm2.digi.ninja zonetransfer.me

; <<>> DiG 9.11.28-RedHat-9.11.28-1.fc33 <<>> axfr @nsztm2.digi.ninja zonetransfer.me
; (1 server found)
;; global options: +cmd
zonetransfer.me.    7200    IN  SOA nsztm1.digi.ninja. robin.digi.ninja. 2019100801 172800 900
↪  1209600 3600
zonetransfer.me.    300 IN  HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.    301 IN  TXT
↪  "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"
zonetransfer.me.    7200    IN  MX  0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200    IN  MX  10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200    IN  MX  10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200    IN  MX  20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.    7200    IN  MX  20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.    7200    IN  MX  20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.    7200    IN  MX  20 ASPMX5.GOOGLEMAIL.COM.
```

```
zonetransfer.me.    7200    IN  A   5.196.105.14
zonetransfer.me.    7200    IN  NS  nsztm1.digi.ninja.
zonetransfer.me.    7200    IN  NS  nsztm2.digi.ninja.
```

J'ai volontairement coupé une bonne partie de la réponse car elle serait trop longue sinon. Nous pouvons donc voir que nous arrivons à avoir une copie de la zone DNS complète, copie que nous ne devrions normalement pas avoir car elle peut comporter des informations sensibles comme par exemple des sous-domaines censés rester en interne ou des record PTR (les enregistrements PTR associent une IP à un nom de domaine) qui permettraient de trouver des IPs à analyser.

Et voilà concrètement ce que donne une requête AXFR sur le réseau :

```
10.2.0.2.36871 > 34.225.33.2.53: Flags [P.], cksum 0xa73b (correct), seq 1:59, ack 1, win 502,
↪  options [nop,nop,TS val 3406917213 ecr 943752624], length 58 38384 [1au] AXFR?
↪  zonetransfer.me. (56)

34.225.33.2.53 > 10.2.0.2.36871: Flags [.], seq 1:1441, ack 59, win 489, options [nop,nop,TS
↪  val 943753618 ecr 3406917375], length 1440 38384*- 51/0/1 zonetransfer.me. SOA
↪  nsztm1.digi.ninja. robin.digi.ninja. 2019100801 172800 900 1209600 3600, zonetransfer.me.
↪  HINFO, zonetransfer.me. TXT
↪  "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA", zonetransfer.me. MX
↪  ASPMX.L.GOOGLE.COM. 0, zonetransfer.me. MX ALT1.ASPMX.L.GOOGLE.COM. 10, zonetransfer.me. MX
↪  ALT2.ASPMX.L.GOOGLE.COM. 10, zonetransfer.me. MX ASPMX2.GOOGLEMAIL.COM. 20,
↪  zonetransfer.me. MX ASPMX3.GOOGLEMAIL.COM. 20, zonetransfer.me. MX ASPMX4.GOOGLEMAIL.COM.
↪  20, zonetransfer.me. MX ASPMX5.GOOGLEMAIL.COM. 20
```

On voit donc que les informations transitent **en clair** (= non chiffrées) sur le réseau.

source : https://blog.eban.bzh/today-i-learned/xfr-zone-transfer.html CC-BY-SA