Détournement de serveur DNS (Third-Man Attack)

PASCAL BONHEUR
PASCAL_BONHEUR@YAHOO.FR
4/07/2001

Introduction

Ce document traite de la possibilité d'exploiter le serveur DNS pour pirater certains sites et notamment détourner des transactions sécurisées entre un client et un serveur. Ce document présente quelques aspect du Domain Name Server, les possibilité de détourner les serveurs de noms et l'utilisation de cette technique dans le cadre d'un détournement de transaction sécurisée.

1 Domain Name Server

Une des caractéristique du DNS qui va nous intéresser est le transfert de zone. Cette opération a lieu entre un serveur primaire et un serveur secondaire d'un domaine dans le but de mettre a jour les enregistrements.

Il s'agit d'un transfert des maps de zone qui, contrairement aux opérations DNS classiques, s'effectue avec TCP en raison de la sécurité et de la fiabilité que nécessite cette opération. Même si a priori, ces opérations ne comportent pas de risque, elles peuvent être utilisées par un hacker pour réunir des informations ou injecter des informations éronnées.

La commande **nslookup** permet de réunir des informations sur le serveur dns d'un domaine :

```
% nslookup
Default serveur : local.dns.com
Adress : 257.257.257.257
> www.test.com
Server : local.dns.com
Adress : 257.257.257.257
```

2 Commandes DNS

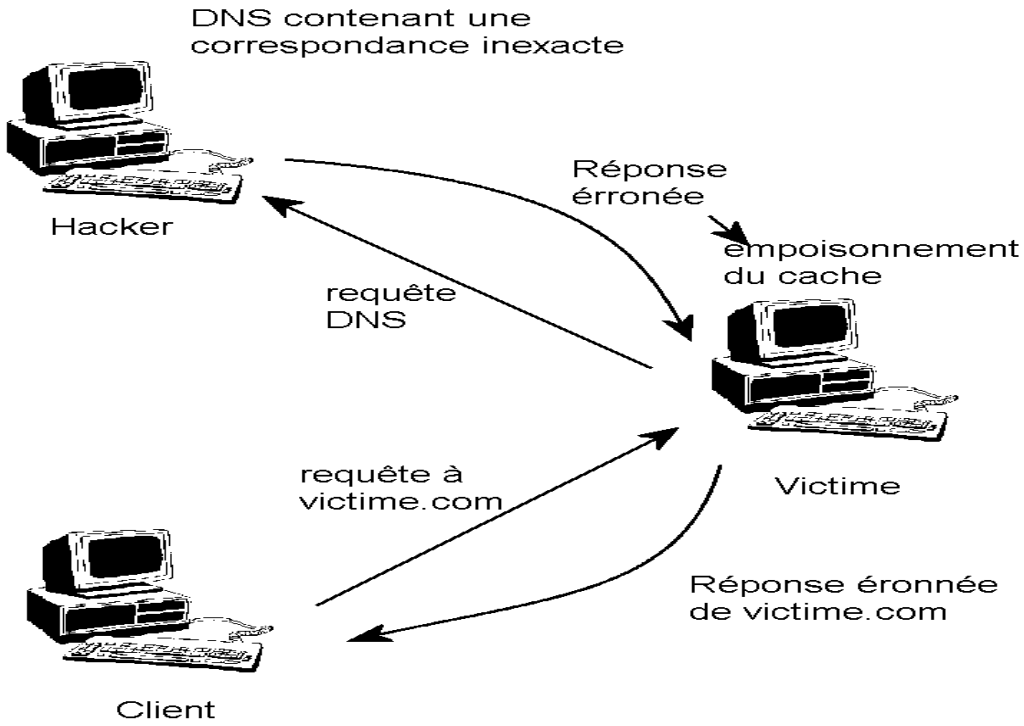
En utilisant la commande *set type=ns*, on peut déterminer le nom des serveurs d'un domaine. La commande *set type=hinfo* permet de réunir des informations sur les numéros de version des serveurs de nom. Ces informations ne peuvent pas toujours être obtenues : si l'administrateur réseau est vigilant, il aura pris soin de désactiver cette option.

Toutes ces informations peuvent être dangereuse si elles sont utilisées par un hacker : celui ci connaissant la version du serveur pourra tenter d'utiliser une bibliothéque d'exploit permettant de prendre le contrôle de ce serveur.

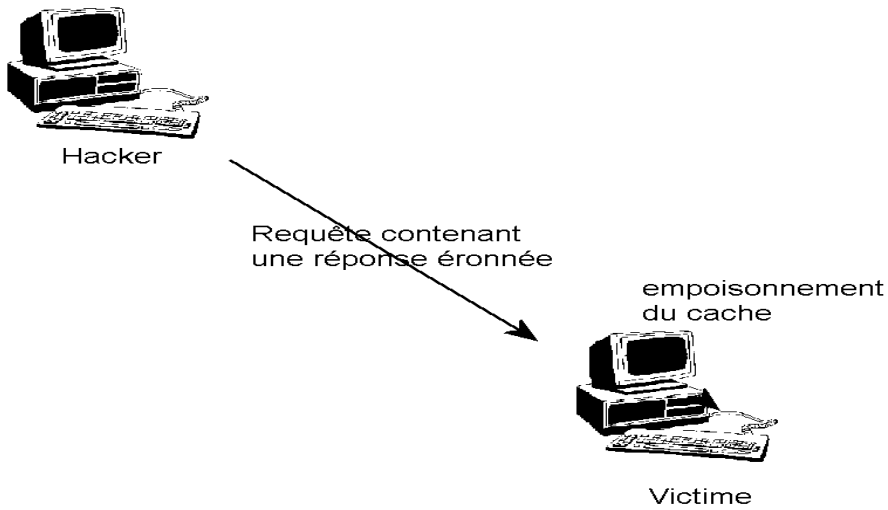
3 Hacking des serveurs DNS

L'avis CA-97.22 met en évidence une faille de sécurité sur les serveurs BIND dont la version est antérieure à la version 8.1.1 (d' où l'utilité pour un éventuel hacker de connaître la version du serveur de nom). Le problème résidait en la mise en mémoire tampon de données éronnées. Pour cela, il faut que le hacker dispose d'un

serveur DNS et qu'il force le serveur DNS qu'il veut hacker à lui envoyer une demande de résolution de nom. La réponse du serveur DNS du hacker contient des données empoisonnées qui vont infecter le cache du serveur DNS. On aura alors un détournement du serveur de nom.



Une autre technique pour hacker les serveurs DNS reposent sur un principe de fonctionnement de certaines version de BIND qui est un peut spécial : il n'y a pas de distinction entre les requêtes et les réponses (le flag QR permet de distinguer une question ou une réponse selon sa valeur : si QR = 0 c'est une question sinon QR = 1 et c'est une réponse). Ainsi, pour tromper un serveur de nom, on peut envoyer une requête qui contient une réponse : la réponse sera alors mise dans la mémoire cache du DNS, ce qui induira le serveur en erreur lors d'une future requête.

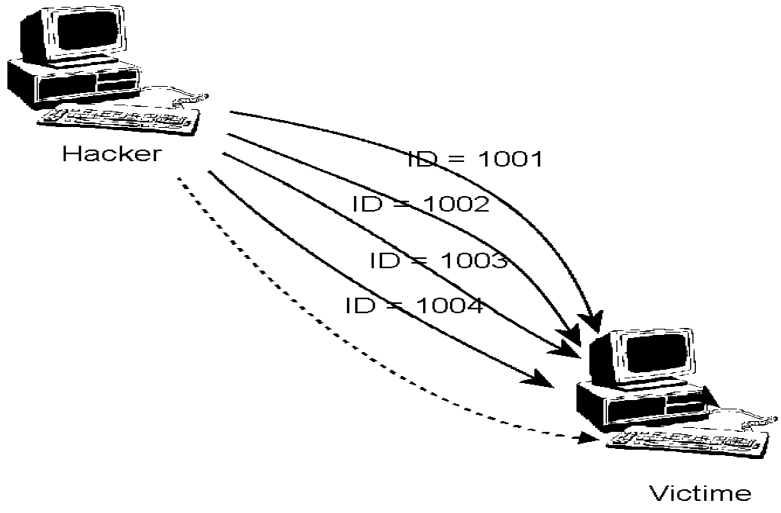


Un problème qui se pose lors de l'empoisenement de serveur DNS est l'identification des paquets. En effet, étant donné que toutes les communications DNS se font par l'intermédiaire du port 53, il est nécessaire d'identifier les différentes requêtes envoyées et les différentes réponses recues. Ainsi, si un hacker veut empoisonner un serveur DNS en envoyant une réponse éronnées, il devra (en plus de répondre plus vite que le serveur DNS à qui était déstiné cette requête) trouver l'ID de la requête DNS pour pouvoir tromper le serveur DNS.

Trouver l'ID peut être très difficile. Dans le cas d'un réseau local, un simple sniffer peut suffir pour trouver l'ID mais sur Internet, la chose se complique. Une des choses utiles est de ralentir le serveur a qui était déstinée la requête pour pouvoir éffectuer plusieurs tentatives : si on a une vague idée du numéro ID on pourra alors éssayer d'envoyer plusieurs paquets avec des IDs différents. L'inconvénient d'un tel procédés est qu'il peut être détecté par des IDS bien configurés : ceux ci permettent facilement de repérer un paquet qui est envoyé de nombreuses fois et dont seul l'ID change.

Pour éviter d'avertir les IDS, on peut essayer d'utiliser des prévisions d'ID qui permettent de trouver l'ID avec une probabilité beaucoup plus importante (De nombreux articles ont été écrits en particulier un traitant de la prédiction des ISN : *Strange Attractors and TCP/IP Sequence Number Analysis*) . Un exemple flagrant d'une telle technique est celui de windows 95 dont le numéro ID est 1 par défaut et 2 dans le cas de deux questions simultanées.

Un autre exemple est celui des failles découvertes dans BIND : le premier ID était bien tirer aléatoirement mais les IDs des requêtes suivantes étaient seulement obtenus en incrémentant l'ID précédent. Ainsi, un hacker peut exploiter cette faille de la facon suivante : en ayant un acces sur un serveur dns ns.exemple.com, le hacker place un sniffer. Ensuite, il envoie une requête sur le serveur qu'il souhaite hacker : ns.victim.com. Le sniffer installé sur ns.exemple.com va permettre de trouver l'ID de la requête envoyée par ns.victim.com a ns.exemple.com : par exemple ID = 1000. Le hacker n'aura alors plus qu'a envoyer une réponse en utilisant pour ID 1001 , 1002 . . . jusqu'a tomber sur le bon ID.



4 Problème des transactions sécurisées

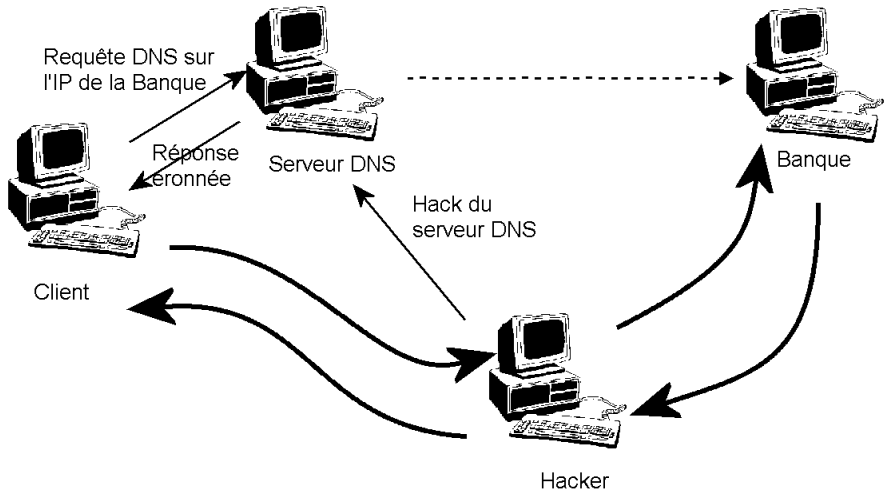
Les garanties de sécurité des transactions sur Internet repose sur des mécanismes de sécurité inviolable avec la puissance de calcul actuelle : ainsi, une transaction cryptée avec une clef de 128 bits est quasiment incassable.

Cependant, le problème peut se trouver autre part dans une transaction. On trouve aujourd'hui facilement sur Internet des milliers de numéro de carte de crédit qui peuvent être utilisés par des hackers. Ces numéros ont pour la plupart été récupéré par piratage de bases de données mais d'autres techniques existe. Parmi elles, une technique basée sur un détournement permet de pirater une transaction sécurisée.

Considérons le cas d'un hacker qui prend le controle du serveur DNS ns.exemple.com d'un client par l'une des techniques décrites précédemment. Le hacker peut modifier la table de ns.exemple.com pour changer l'adresse IP d'une banque www.banque.com pour la faire pointer vers son site. Ce site sera une paserelle entre le client et la banque.

Lorsque le client croira se connecter sur sa banque en ligne www.banque.com, il se connectera en réalité sur le site www.hacker.com. Celui-ci récupérera les informations envoyées par le client et les enverra vers le site de la banque. En récupérant les réponses de www.banque.com et en les renvoyant aux clients, le hacker créera l'illusion parfaite : le client ne verra pas de différence. Le hacker aura alors la possibilité de connaître toutes les informations d'identifications.

Le problème réside donc dans l'identification qui n'a lieu qu'a sens unique. La banque identifie le client mais le client n'identifie pas la banque. La solution serait donc une identification mutuelle. Ainsi, si le client et la banque disposaient chacun d'une clef publique, il n'y aurait pas de problème quant à l'usurpation d'identité.



5 Solutions aux problèmes des serveurs DNS

Certains problèmes concernant le piratage de serveurs DNS peuvent être résolus en prenant soin d'appliquer les patchs distribués pour des applications comme BIND. Ainsi, les versions récentes de BIND corrigent les problèmes d'empoisonnement du cache.
D'autres problèmes peuvent être résolus par une implémentation plus soigneuse. Ainsi, le choix des ID des paquets DNS doit être rendu le plus aléatoire possible : il faut dans la mesure du possible augmenter l'entropie et choisir des paramètres aléatoires comme les frappes aux claviers ou des événements concernant le temp.

Un autre problème quant aux détournements DNS réside dans le faible pourcentage d'attaques ayant lieu de cette manière : cela explique certaines négligences. On constate cependant les dêgats que peuvent engendrer un détournement DNS : l' exemple du site d'Hillary Clinton qui fut détourné vers le site de sa concurrente est assez révélateur.
Pour toutes ces raisons, un nouveau protocole (où plutôt une extension du protocole DNS) a été dévellopé et fait parti de versions récentes de serveurs DNS comme BIND (versions supérieures à 9).

6 Le protocole DNSSEC

Ce protocole dont les spécifications sont définis dans le RFC 2535 a pour but d'accroitre la sécurité des données lors des échanges DNS.
Les ajouts principaux concernent les Ressources Records (RR) (Toutes les données retournées par des serveurs de noms ont ce format). En effet, des RRs ont été ajoutés pour introduire une sécurité supplèmentaire. Avant de voir ces trois ajouts, nous allons brièvement voir la structure des RR classiques.

Name1	Name2	...	Name n
Type1	Type2	Class1	Class2
TTL1	TTL2	TTL3	TTL4
RDLength1	RDLength2	RData1	RData n

Comme le montre la figure, le champs nom est un champ de longueur variable et il est terminé par le caractère NULL. Après cela vient le champs TYPE de deux bits (qui peut par exemple être HINFO) puis la CLASS (qui sera souvent IN pour internet) . Ensuite, on retrouve le champ TTL qui est fournit pour les services de mise en mémoire cache.
L'avant dernier champ contient la taille des données dans le paquet et le dernier champ RDATA contient les données et il est de longueur variable.

Les trois principaux RR ajoutés sont KEY RR,SIG RR et NXT RR.

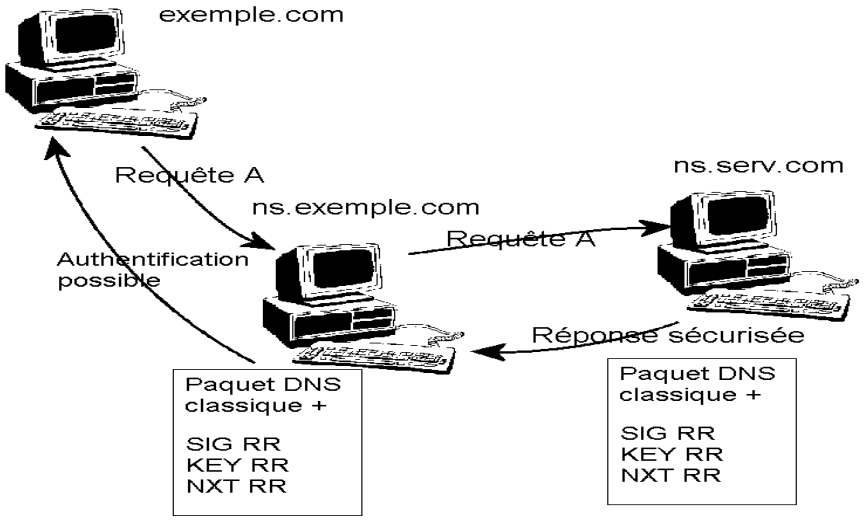
– KEY RR contient comme son nom l'indique une clé publique permettant de crypter. Cette clé est retournée dans le champ RDATA du RR. Ce RR fournit des informations de sécurité comme la clé, l'algorithme de cryptage (par exemple RSA/MD5,elliptic curves algorithm, DSA ...). Le champ concernant l'algorithme de cryptage étant un champ de 8 bits, on peut enregistrer jusqu'à 255 algirithmes de cryptages différents ce qui laisse entrevoir de nombreuses possibilités.

– SIG RR contient une signature qui permet d'authentifier le serveur qui envoie la réponse.

– NXT[1] RR permet de mettre en mémoire cache les réponses négatives (ie qui correspondent à des requêtes qui n'ont pues être résolues).Le protocole DNSSEC fournit des signatures pour ces NXT RR pour que ces réponses de non-existence puissent être authentifiées. Les NXT RR sont utilisés pour signifier que certains serveurs de noms sont indisponibles.

[1]NXT signifie nonexistent

Ainsi, ces RR permettent deux choses essentielles lors des transitions DNS : l'authentification et la vérification d'intégrité. La présence de la clé permet en effet d'identifier avec certitude le serveur DNS qui a répondu.
La figure ci dessous montre le procédé de la résolution de nom sécurisée.



Conclusion

Ce protocole sécurisé permettra vraisemblablement de réduire le nombre de piratage concernant les serveurs DNS.
Cependant l'augmentation du nombres de données transmises (certes faible pour une seule requête mais important pour des gros serveurs) entraîne une diminution de la bande passante et un temps de communication plus élevé.
Les premiers a être intéréssés par ce protocole sont les militaires (sites .mil) ainsi que les sites business to business.
Même si actuellement, les attaques les plus fréquentes sont des attaques directes basées sur l'utilisation d'exploits, il est fort vraisemblable que le développement des transactions sur internet donnera tout on sens au protocole DNSSEC.