Service Réseau : DNSSEC

Contexte

Ce travail pratique s'inscrit dans le cadre du module Services Réseaux Avancés. Il s'est déroulé de manière individuelle, encadré par un enseignant, sur une durée totale d'environ 6 heures réparties sur plusieurs séances.

Le TP avait pour but d'approfondir la compréhension du fonctionnement du système de noms de domaine (DNS) et de sa sécurisation à l'aide du protocole DNSSEC.

Les manipulations ont été réalisées sur des machines Linux (Debian), avec deux serveurs DNS (ns1 et ns2) et un client configuré pour interagir avec ces serveurs. L'analyse des échanges réseau a été effectuée à l'aide de Wireshark.

Objectif

L'objectif principal de ce TP était de :

- Comprendre le fonctionnement du DNS et les différents types de requêtes (A, NS, MX, CNAME, PTR, etc.).
- Installer et configurer un serveur DNS principal (Bind9) et un serveur secondaire.
- Mettre en place des zones directes et inverses et vérifier leur cohérence.
- Configurer un transfert de zone sécurisé à l'aide d'une clé TSIG.
- Implémenter la signature des zones DNS avec DNSSEC, incluant les clés ZSK et KSK.
- Mettre en place un relais de zone sécurisé pour assurer la validation cryptographique des réponses DNS.

Travail réalisé

1. **Tests de différents types de requêtes DNS**

J'ai d'abord exploré le fonctionnement du DNS en ligne à l'aide des outils *Google Dig* et *Dig Web Interface*. En effectuant une requête sur le domaine uca.fr, j'ai pu obtenir l'adresse IP 193.49.117.66.

En activant l'option « no recursive », aucune IP n'était retournée, car ce mode exige que l'utilisateur spécifie le serveur DNS possédant l'information (serveur autoritaire).

En utilisant dig directement depuis ma machine sur le domaine google.fr, j'ai observé les réponses des serveurs DNS selon qu'ils étaient interrogés en mode récursif ou non. Les analyses Wireshark ont montré les échanges de requêtes et réponses DNS entre le client et les serveurs, confirmant le rôle du résolveur et du serveur autoritaire.

2. Installation du serveur DNS Bind9

J'ai ensuite installé le service Bind9 sur le serveur principal (ns1) à l'aide des commandes apt update et apt install bind9.

Après vérification du statut du service avec systemctl status bind9, j'ai confirmé qu'il écoutait bien sur le port 53.

Le répertoire */etc/bind* contient les fichiers de configuration essentiels : *named.conf, named.conf.local, named.conf.options et named.conf.root-hints*. J'ai testé la commande *rndc dumpdb*, qui permet de sauvegarder le cache DNS après des requêtes locales avec dig.

3. Configuration du serveur DNS primaire

La zone locale créée est secu.rt. J'ai ajouté la configuration correspondante dans */etc/bind/named.conf.local*.

J'ai ensuite créé le fichier de zone */etc/bind/db.secu* contenant les enregistrements de type NS, A, MX et CNAME pour les serveurs ns1, ns2, mail, client et leurs alias.

Après vérification syntaxique avec *named-checkzone* et redémarrage de Bind9, toutes les ressources étaient correctement chargées.

4. Création d'une zone inverse

J'ai ajouté une zone inverse *0.168.192.in-addr.arpa* pour associer les IP aux noms de domaine.

Le fichier */etc/bind/db.192.168.0* contient les enregistrements PTR pour chaque hôte. La commande *dig -x 192.168.0.40* a permis de vérifier la résolution inverse du serveur mail.secu.rt.

5. Configuration du client DNS

Sur le client, j'ai modifié le fichier */etc/resolv.conf* pour spécifier le serveur DNS par défaut. Après cela, les requêtes *dig @172.25.0.62 client.secu.rt* retournaient bien les bonnes adresses.

Le ping vers client.secu.rt a fonctionné après ajustement de l'adresse IP du client.

Les captures Wireshark ont montré les échanges typiques (résolution DNS suivie des paquets ICMP) entre le client et le serveur.

6. Mise en place d'un serveur secondaire et transfert de zone sécurisé

J'ai installé Bind9 sur un second serveur (ns2) et configuré la zone secu.rt en type *slave* (secondaire), reliée au serveur maître ns1.

Sur le serveur primaire, j'ai ajouté les directives *allow-transfer* et *also-notify* pour autoriser le transfert.

Pour sécuriser ce transfert, j'ai généré une clé TSIG avec *tsig-keygen -a HMAC-SHA512*. La clé a été déclarée dans les deux fichiers *named.conf.local* pour assurer l'authentification mutuelle. Le protocole TSIG garantit l'intégrité et l'authenticité des données échangées entre les deux serveurs.

7. Mise en œuvre de DNSSEC et signature de la zone

J'ai généré deux paires de clés : une ZSK (*Zone Signing Key*) et une KSK (*Key Signing Key*). Les fichiers de clés ont été inclus dans la zone via les directives $INCLUDE.

La signature de la zone a été effectuée par la *commande dnssec-signzone -o secu.rt -x db.secu*. Le fichier signé *db.secu.signed* a été intégré dans la configuration et testé avec *dig +dnssec*.

Ensuite, j'ai signé la zone avec NSEC3 pour éviter l'énumération de zones, puis validé la présence du paramètre NSEC3PARAM.

8. Mise en place d'un relais de zone

Sur le serveur secondaire, la zone a été configurée en type *forward* pour transférer les requêtes vers le maître. Une ancre de confiance a été ajoutée pour la validation DNSSEC.

Les requêtes *dig +dnssec +multi* ont montré les drapeaux (flags) QR, RD, RA et AD, indiquant que la validation cryptographique fonctionnait correctement.

Résultat

À l'issue de ce TP, j'ai mis en place une infrastructure DNS complète et sécurisée comprenant :

- Un serveur primaire et un serveur secondaire synchronisés.
- Des zones directes et inverses correctement configurées.
- Un transfert de zone sécurisé par TSIG.
- Une signature DNSSEC garantissant l'intégrité et l'authenticité des données.
- Un relais de zone capable de valider cryptographiquement les requêtes.

Ce TP m'a permis de comprendre en profondeur le fonctionnement interne du DNS, la configuration de Bind9 et les mécanismes de sécurisation mis en œuvre avec DNSSEC.