Un Peuple – Un But – Une Foi

REPUBLIQUE DU SENEGAL

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE ET DE L'INNOVATION

UNIVERSITE GASTON BERGER DE SAINT-LOUIS

U.F.R DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

CENTRE DE FORMATION PROFESSIONNELLE PLURIDISCIPLINAIRE

RAPPORT DE STAGE DE FIN D'ETUDES POUR L'OBTENTION DU

MASTER PROFESSIONNEL RESEAUX ET TELECOMMUNICATIONS

Sujet : Découverte de services basée sur DNS-SD en lien avec l'Internet des Objets et mécanismes de sécurité associés

Présenté le 10/03/2022 par :

AMADOU ABDOULAYE LY

Sous la direction de :

Dr. Chérif DIALLO

Jury :

Président :	M. Daouda SANGARE
Examinateur :	M. Abdou Karim MBODJI
Examinateur :	M. Mahamadou TRAORE
Encadrant :	M. Chérif DIALLO

Année académique 2020/2021

DÉDICACE

À la mémoire de mon cher père.

Je dédie ce travail à mon père décédé très tôt, pour exprimer l'amour, l'estime, le respect que j'ai toujours eu pour lui. Puisse Dieu le tout puissant l'avoir en sa sainte miséricorde.

À ma chère mère qui m'a soutenu et encouragé durant ces années d'études.

À mes frères et sœurs qui n'ont pas cessé de me conseiller, encourager et soutenir tout au long de mes études,

À tous mes enseignants,

À toute ma famille pour le soutient qu'elle m'a apporté tout au long de mon parcours universitaire.

Puisse Dieu vous donner santé, bonheur, courage et surtout réussite.

REMERCIEMENTS

Je tiens tous d'abord à remercier le Bon Dieu de m'avoir accordé la foi, le courage, de la force et de la patience tout au long de mon parcours.

J'exprime mes sincères remerciements et toute ma gratitude à mon encadrant professeur Cherif Diallo pour son excellente qualité d'encadrement et pour les orientations et les consignes pertinentes qu'il m'a accordées, qui m'ont été très utiles durant les différentes phases de réalisation de ce mémoire.

Je remercie vivement les honorables membres du jury qui ont accepté d'évaluer mon travail.

Je voudrais remercier mes chers parents et mes frères et sœurs pour leur accompagnement et leurs encouragements permanent. Ils n'ont préservé aucun effort pour m'aider à mener à succès mes études.

Je remercie sincèrement tous les gens, qui ont participé de proche ou de loin à la réussite de mon parcours scolaire

RÉSUME

L'environnement IoT présente des exigences difficiles pour la découverte de services, telles que : l'hétérogénéité des services, la mobilité, l'évolutivité la sécurité, la prise en charge de qualité de service et la gestion du contexte.

Différents protocoles ont été proposés pour faciliter la découverte de service, mais il est difficile d'évaluer dans quelle mesure ces protocoles répondent aux exigences de l'IoT.

Cet article présente une évaluation des protocoles de découverte de services couramment utiliser pour l'IoT à savoir le DNS-SD.

Mots clés : Internet des Objets, DNS-SD, sécurité

ABSTRACT

The IoT environment presents challenging requirements for service discovery, such as: service heterogeneity, mobility, scalability, security, QoS support and context management.

Various protocols have been proposed to facilitate service discovery, but it is difficult to evaluate how well these protocols meet the requirements of the IoT.

This paper presents an evaluation of service discovery protocols commonly used for IoT, namely DNS-SD.

Keywords: Internet of Things, DNS-SD, security

LISTES DES ABRÉVIATIONS

BOOTP : Bootstrap Protocol

DNS : Domain Name System

DNS-SD : DNS Service Discovery

DDoS : Distribute Denial os Service

DNSSEC : Domain Name System Security Extensions

HTTP : HyperText Transfert Protocol

IoT : Internet of Things

IoE : Internet of Everythings

ICMP : Internet Control Message Protocol

IETF : Internet Enginering Task Force

KSK : Key Signing Key

MDNS : multicast DNS

MiTM : Man in The Middle

MAC : Media Access Control

NSEC : Next SECure record

RFID : Radio Fréquency IDenfication

RCSF : Réseaux de Capteur Sans Fil

RRSIG : Ressource Record Signature

RRset : Ressource record set

SSDP : Simple Service Discovery Protocol

SOA : Start Of Authority

TTL : Time To Live

TLD : Top-Level Domain

UDP : User Datagram Protocol

UPnP : Universal Plug and Play

VoT : Virtualization of Things

WoT : Web of Things

WI-FI : Wireless Fidelity

XML : eXtensible Markup Language

Zeroconf : Zero configuration networking

Figure 1: Champs d'enregistrement DNS 4
Figure 2 : hiérarchie DNS (source) 6
Figure 3 : Zeroconf 8
Figure 4 : fonctionnement de DHCP 9
Figure 5 : datagramme du message DNS 10
Figure 6 : Structure d'enregistrement SRV 11
Figure 7: Application Bonjour d'Apple 13
Figure 8 : Avahi 14
Figure 9 : scénario de découverte 1 16
Figure 10 : scénario de découverte 2 17
Figure 11 : scénario de découverte 3 18
Figure 12 : Universal plug in Play 20
Figure 13 : Simple Service Discovery Protocol 21
Figure 14 : nœud capteur 24
Figure 15 : Objets d'identifications 25
Figure 16 : drone connecté 25
Figure 17 : Typologie des objets dans l'IoT 26
Figure 18 : Formes des étiquettes RFID 27
Figure 19 : Le domaine militaire et l'Internet des objets 29
Figure 20 : L'internet des objets dans le domaine médical 30
Figure 21 : IoT et l'Industrie 31
Figure 22 : L'Internet des objets et la domotique 32
Figure 23 : ville intelligente 32
Figure 24 : Attaque DDoS 36
Figure 25 : Attaque MiTM 37
Figure 26 : Attaque par DNS Cache Poisoning 39
Figure 27 : RRset Ensemble d'enregistrements de ressources 42
Figure 28 : RRSIG 42
Figure 29 : ZSK clé de signature de zone 43
Figure 30 : KSK clé de signature de clé 44
Figure 31: validation de la signature 45
Figure 32 : Signature de Délégation 46
Figure 33 : La chaine de confiance 47
Figure 34 : Raspberry Pi 50
Figure 35 : Téléchargement des paquets DNS 51
Figure 36 : Déclaration de zone DNS 51
Figure 37 : Définition de la zone de recherche 52
Figure 38 : Installation des paquets Avahi 52
Figure 39 : Configuration du service de lumière 53
Figure 40 : Configuration du service d'imprimérie 54
Figure 41 : Déclaration des services dans le domaine 54
Figure 42 : Découverte des services 55
Figure 43 : Configuration du serveur DHCP 55
Figure 44 : Câblage du système 56
Figure 45 : Programme 57
Figure 46 : Sniffing du réseau 58
Figure 47 : Activation de DNSSEC 59
Figure 48: Génération de ZSK 59
Figure 49 : Géneration de KSK 59
Figure 50 : La clé partagée 59
Figure 51 : Signature de la zone 60

Table des matières

I. INTRODUCTION GENERALE 1
II. PREMIERE PARTIE : ETAT DE L'ART 4
Chapitre 1 : Présentation du système DNS-SD 4
1. Introduction 4
2. Contexte 4
2.1 DNS 4
2.2 ZEROCONF 7
2.3 Protocole DHCP 8
3. Découverte de service basé sur DNS (DNS-SD) 10
3.1 Objectif du système DNS-SD 11
3.2 Fonctionnement du DNS-SD 11
3.3 Utilisation de DNS-SD 12
3.4 Types d'enregistrements du DNS-SD 13
4. Applications de découverte de services (DNS-SD) 13
4.1 Application Bonjour 13
4.2 Application Avahi 14
4.3 Application I-phone 15
5. Scénarios de découverte de services 15
5.1 Client Privé et Serveur public 15
5.2 Client privé et Serveur privé 16
5.3 Client et serveur portable 17
6. Différents protocoles de découverte de service 18
6.1 Protocole mDNS 19
6.2 Protocole DNS-SD 19
6.3 Protocole UPnP 19
6.4 Protocole SSDP 20
7. Conclusion 21
Chapitre 2 : Généralités sur l'Internet des objets 23
1. Introduction 23
2. Historique 23
3. Les types d'objets 24
3.1 Les nœuds capteurs 24
3.2 Les objets d'identification 24
3.3 Les drones 25

3.4 Les smartphones et tablettes 26
4. Technologie fondatrice d'IoT 26
4.1 L'identification par radio fréquence (RFID) 27
4.2 Les réseaux de capteurs sans fil (RCSFs) 28
5. Les applications de l'IoT 28
5.1 Application militaire 28
5.2 Application médicale 29
5.3 Applications industrielles (smart indistry) 30
5.4 Les maisons intelligentes (smart home) 31
5.5 Les villes intelligentes (smart city) 32
6. Les défis de l'IoT 33
6.1 La sécurité 33
6.2 La protection de la vie privée des utilisateurs 33
6.3 Limitations de ressources 33
6.4 La virtualisation 33
6.5 Le nombre croissant des objets connectés 33
7. Les avantages de l'internet des objets 33
8. Conclusion 34
III. DEUXIEME PARTIE : MÉCANISMES DE SECURITE DANS LE DNS-SD 35
Chapitre 3 : Étude de la sécurité dans DNS-SD 36
1. Introduction 36
2. Mécanisme d'attaques dans DNS 36
2.1 Attaque par inondation (DDoS) 36
2.2 Attaque MiTM (Man in The Middle) 37
2.3 Attaque par écoute illicite 37
2.4 Attaque sur mDNS 38
2.5 Résistance à l'usurpation d'identité de l'expéditeur 38
2.6 Résistance aux attaques de dictionnaires 38
2.7 Attaque sur la base de découverte (DNS) 39
3. La sécurité des données dans la découverte de service 40
3.1 Authenticité, intégrité et fraîcheur 40
3.2 Confidentialité 40
3.3 Mécanismes de contrôle d'accès 41
3.4 Refus de l'expéditeur (déni de responsabilité) 41
3.5 Extension de sécurité de nom de domaine (DNSSEC) 41
4. Conclusion 48

IV. TROISIEME PARTIE : MISE EN PLACE DU SOLUTION DNS-SD 49
Chapitre 4 : Mise en œuvre de DNS-SD 50
1. Introduction 50
2. Mise en place du système de découverte 50
2.1 Installation de bind9 51
2.2 Installation de Avahi 52
2.3 Installation du serveur DHCP 55
3. Implémentation de la lampe connectée 56
4. Attaque sur le système de découverte 57
5. Configuration de DNSSEC 58
6. Conclusion 61
V. CONCLUSION GENERALE 62
Webographie 64
ANNEXES 66

I. INTRODUCTION GENERALE

La place de plus en plus prépondérante qu'occupe l'Internet dans notre vie sociale est devenue incontestable.

L'internet des objets, ou IoT (Internet of Things), est un réseau d'objets tangibles connecté à Internet pour échanger des données, sans aucune interaction humaine. L'IoT a radicalement bouleversé notre manière de voir, de penser, de communiquer, mais surtout d'interagir avec notre environnement qui est devenu de plus en plus complexe, et pour cause la majorité des agents économiques ont dû s'adapter à cet avènement.

Ainsi, cette réalité implique une nouvelle phase de changement. La méthodologie de traitement de données nécessite la création et la mise à disposition de nouveaux outils de découvertes plus performants et plus précis pour gérer au mieux l'énorme flux de services dans l'Internet. L'importance de la phase de découverte dans le cadre de l'IoT est donc un des premiers postulats de la raison pratique auxquels on doit porter une attention particulière.

Le nombre énorme de dispositifs dans les applications IoT nécessite une technique efficace, dynamique et sans configuration pour la découverte de services.

Les protocoles Multicast DNS (mDNS) et DNS Service Discovery (DNS-SD) ont été conçus pour la découverte de services dans des dispositifs riches en ressources. Ce dernier favorise considérablement à rendre un service disponible et à être découvert dans un réseau local par d'autre appareils connecté sur ce réseau par le biais de son nom.

L'IoT peut communiquer, traiter, analyser et gérer des données, automatiquement et sans aucune intervention humaine, d'ailleurs c'est ce qui fait sans grande puissance dans le monde de la technologie. Cependant, les problèmes liés à la sécurité restent un frein majeur pour le développement de ce dernier. Les vols de d'information, l'usurpation d'identité, représente un grand danger pour ce système.

Notre approche se base sur comment fonctionne le système DNS-SD, de l'enregistrement jusqu'à la mise en disposition du service, et d'étudier le mécanisme de sécurité dans ce dernier.

Cet article est organisé comme suit :

Partie 1 : Les objectifs que nous avons présentés dans l'introduction nous ont conduits à étudier l'état de l'art des différents domaines de recherche définissant la portée du mémoire. Nous effectuerons une analyse détaillée du système DNS-SD (chapitre 1) et l'internet des objets (chapitre 2).

Partie 2 : nous présenterons les mécanismes de sécurité dans le système de découverte.

La partie 3 nous allons tenter de mettre en œuvre un système de découverte de service basé sur DNS (DNS-SD), puis nous allons terminer par une conclusion générale.

Objectifs de l'étude

Notre objectif primordial est de mettre en place un système de découverte de services basé sur DNS, qui sera en mesure de publier des services sur un réseau, tels que Avahi.

C'est dans cette optique que la mise en place du système s'effectuera via un serveur DNS.

Pour cela, de façon non exhaustive, ce document présentera l'état de l'art du système DNS-SD ainsi que l'Internet des objets, ensuite détaillera le mécanisme de sécurité associé et enfin nous implémenterons un système de découverte de services basé sur DNS pour valider notre travail.

À l'issu de ce travail, nous serons en mesure de faire la configuration de DNS-SD, des objets connectés, et d'enregistrer des services pour la découverte

II. PREMIERE PARTIE : ETAT DE L'ART

Chapitre 1 : Présentation du système DNS-SD

1. Introduction

La découverte de service basée sur DNS (DNS-SD), DNS multidiffusion (mDNS) consiste à mettre en place un système pour la découverte des services sans configuration. C'est très pratique pour les utilisateurs, mais cela nécessite l'exposition publique de l'identité des demandeurs ainsi que des informations sur les services proposés et demandés. Certaines parties des informations publiées peuvent porter gravement atteinte à la vie privée de l'utilisateur. Ces problèmes de confidentialité et les solutions potentielles sont discutés dans la partie 2. Alors que la nature multicast du mDNS rend ces risques évidents, la plupart des risques découlent de l'observabilité des transactions. Ces risques doivent également être atténués lors de l'utilisation des services basées sur le serveur de DNS-SD. Il existe des cas où des nœuds connectés à un réseau souhaitent fournir ou consommer des services sans exposer leur identité aux autres parties connectés au même réseau.

2. Contexte

2.1 DNS

Un système de noms de domaine, ou DNS, permet de résoudre les noms de domaine lisibles par l'homme (par exemple www.google.com) en adresses IP lisibles par une machine (par exemple, 192.1.0.10). Un enregistrement de ressource est une entrée dans un fichier de base de données DNS et se compose d'un nom, d'une durée de vie, d'un type et de données spécifiques au type. Ces enregistrements de ressources, dans une structure hiérarchique, constituent le système de noms de domaine (DNS).

Les champs d'enregistrement de ressource sont définis comme suit :

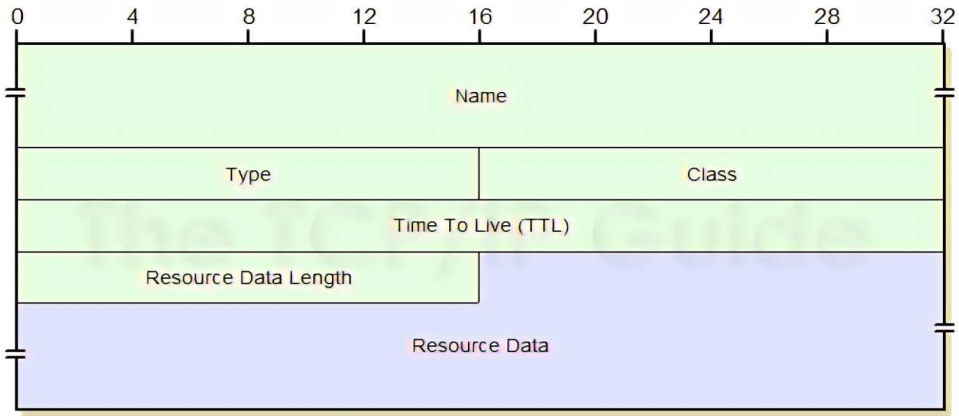


Figure 1: Champs d'enregistrement DNS

Source : http://www.tcpipguide.com/free

- **Name :** ce champ, est le nom de l'enregistrement de domaine et il doit toujours commencer dans la colonne 1. Le nom de domaine correspond à ce que l'internaute va taper dans son navigateur.

- **TTL :** Le champ **ttl** (durée de vie), est facultatif. Ce champ spécifie combien de temps l'enregistrement de ressource est stocké par le DNS. Si ce champ est laissé vide, la valeur TTL par défaut est spécifiée dans l'enregistrement de ressource de début d'autorité (SOA) (décrit plus loin dans ce chapitre).
- **Classe :** ce champ est la classe d'adresse. Actuellement, une seule classe est prise en charge : **IN**, pour les adresses Internet et autres informations Internet.
- **Type :** Le quatrième champ, **type d'enregistrement,** définit le type de cet enregistrement de ressource, tel qu'A, **NS** ou **CNAME**.
- Le champ **Resource Data Length** (**RDLENGTH**) spécifie le nombre d'octets contenus dans le champ
- **Resource Data** (**RDATA**). Le format de ces données dépend du type. Par exemple, les enregistrements A (type 1) ont une adresse IPv4 32 bits dans la zone RDATA. [1]

➢ Les principaux types d'enregistrements DNS

Un type d'enregistrement permet de préciser le type d'information qui est contenu dans l'enregistrement. En plus de la résolution des noms de domaines à partir des adresses IP, les enregistrements DNS ont d'autres fonctions.

- Enregistrement A
  Les enregistrements d'adresse, ou enregistrement A (aussi appelés « enregistrement d'hôte ») associent un domaine à l'adresse IP physique d'un ordinateur hébergeant les services de ce domaine.

- Enregistrement PTR
  Le PTR-Record (pointer) est un DNS record qui permet un Reverse-lookup (recherche inversée). Le serveur DNS peut ainsi donner des informations sur les noms des hôtes qui sont associés à une adresse IP spécifique. Chaque adresse IP utilisée dans un enregistrement de type A ou AAAA, est associée à un enregistrement PTR. L'adresse IP est alors structurée dans l'ordre inverse, avant d'être complétée par le nom d'une zone.

- Enregistrement CNAME
  Un enregistrement CNAME ou enregistrement de nom canonique associe un nom d'alias au véritable nom de domaine (nom de domaine canonique). Par exemple www.exemple.com peut être associé à exemple.com.

- Enregistrement AAAA
  Un enregistrement AAAA, aussi connu sous le nom de « quad-a », fonctionne comme un enregistrement A, sauf qu'il a recours à une adresse IPv6 au lieu d'une adresse IPv4 pour résoudre le nom du domaine. Une adresse IPv6 ayant une longueur de 128 bits ou 16 octets, on précise ici la longueur du champ de données. La désignation AAAA est une manière d'indiquer que cet enregistrement DNS est quatre fois plus long qu'un enregistrement A [1].

- Enregistrement TXT

Un enregistrement TXT est un enregistrement DNS qui fournit des informations au format texte à des sources extérieures à notre domaine. Ces informations peuvent servir différents objectifs. La valeur de l'enregistrement est un texte exploitable humainement ou par ordinateur.

- Enregistrement NS
  Les enregistrements NS (Name Server) déterminent quels serveurs communiquent les informations DNS d'un domaine. Généralement, vous disposez d'enregistrements NS principaux et secondaire pour votre domaine.

- Enregistrement MX
  Les enregistrements Mail Exchange (MX) dirige les e-mails d'un domaine vers les serveurs hébergeant les comptes utilisateur du domaine.

- Enregistrement SRV
  Avec un enregistrement SRV, vous pouvez informer un serveur d'autres services(SRV) proposés. Vous y inscrivez le service disponible, ainsi que le numéro de port. Vous pouvez aussi y entrer le protocole que vous souhaitez employer. Cette entrée DNS permet par exemple de donner à un client des informations portant sur des services LDAP ou XMPP. [2]

Le DNS est organisé sous la forme d'une inversé, avec une « racine » dont dépendent les différentes « branches ». Au premier niveau de l'arborescence se trouve les « Top-Level Domain » ou domaine de premier niveau, comme les .sn, .com, .fr, .mr etc. Au second niveau, nous avons les Domain Name ou les noms de domaine « classique » comme « google.com ». Fonctionnant comme une base de données distribuée sur des millions de machines, le DNS repose sur des interactions entre ces machines permettant d'identifier celle qui est la plus susceptible de pouvoir répondre à la requête d'un internaute.

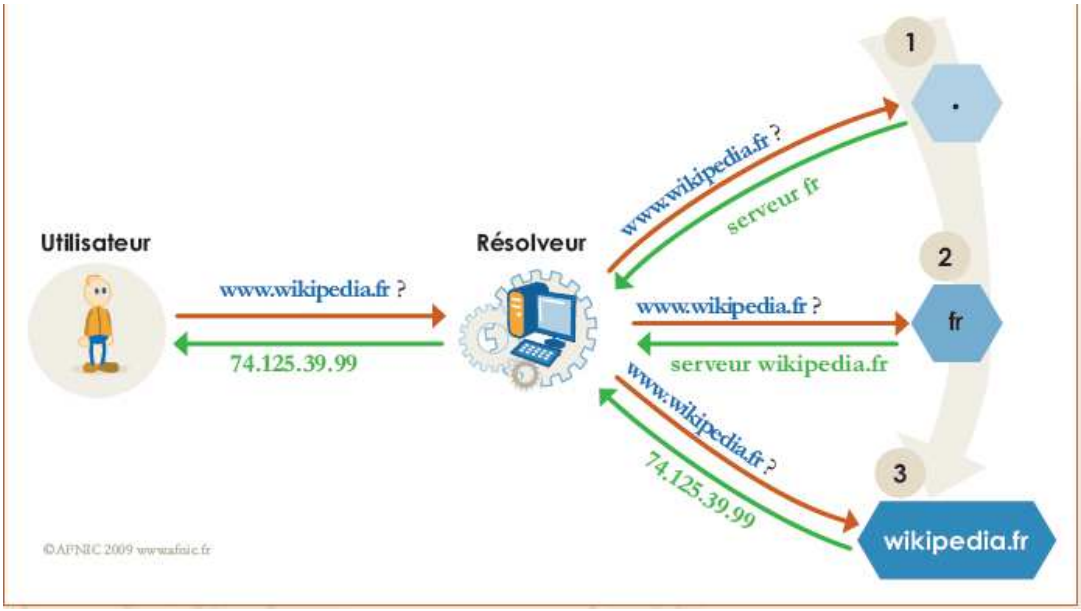


Figure 2 : hiérarchie DNS (source)

Dans l'exemple ci-dessus, l'utilisateur veut se connecter au site http://www.wikipedia.fr. Il envoie sa requête via son navigateur. Celle-ci est reçus par un serveur dit « résolveur » qui a pour première mission d'identifier la machine sur laquelle est installé le nom de domaine wikipedia.fr le résolveur s'adresse d'abord à la « racine » du DNS (matérialisé par un point), qui lui indique quels sont les serveurs faisant autorité (c'est-à-dire compétents) pour .fr puisque le nom de domaine est en.fr. Dans un second temps, les serveurs du .fr indiquent à leur tour au résolveur que le nom de domaine wikipedia.fr est hébergé sur tel serveur. Celui-ci est alors en mesure d'indiquer au navigateur l'adresse IP du serveur web hébergeant les contenus du site web www.wikipedia.fr [2].

- DNS-LLQ et DNS-UL

Une fois le serveur DNS configuré, les clients peuvent utiliser et/ou ajouter des services. Cependant, il est d'abord recommandé d'utiliser les extensions DNS-LLQ et DNS-UL présentées ci-dessous.

- DNS-LLQ (DNS Long-Lived Queries)

  L'extension DNS-LLQ permet aux clients d'être immédiatement notifiés lorsque de nouveaux services sont ajoutés ou enlevés du serveur. Sans cette fonctionnalité, les clients devraient périodiquement explorer les domaines DNS-SD pour connaitre la liste des services disponibles. En plus d'être incommode, cela ne permettrait pas d'avoir les informations en temps réel.

- DNS-UL (DNS Update-Leases)

  L'extension DNS-UL impose une durée limite de validité pour les mises à jour d'enregistrements. Donc un service sera automatiquement effacé s'il ne se réenregistre pas régulièrement. Cela permet d'être certain que les services devenus indisponibles ne soient plus annoncés [3].

  Utiliser DNS-LLQ sans DNS-UL conduira à ce que les services restent présents sur le serveur jusqu'à ce qu'ils soient effacés manuellement. On se retrouverait probablement avec un serveur qui annonce des services qui n'existent plus.

2.2 ZEROCONF

Zeroconf est l'ensemble de technologie permettant à plusieurs ordinateurs de communiquer sans configuration. Il est destiné aux petits réseaux locaux

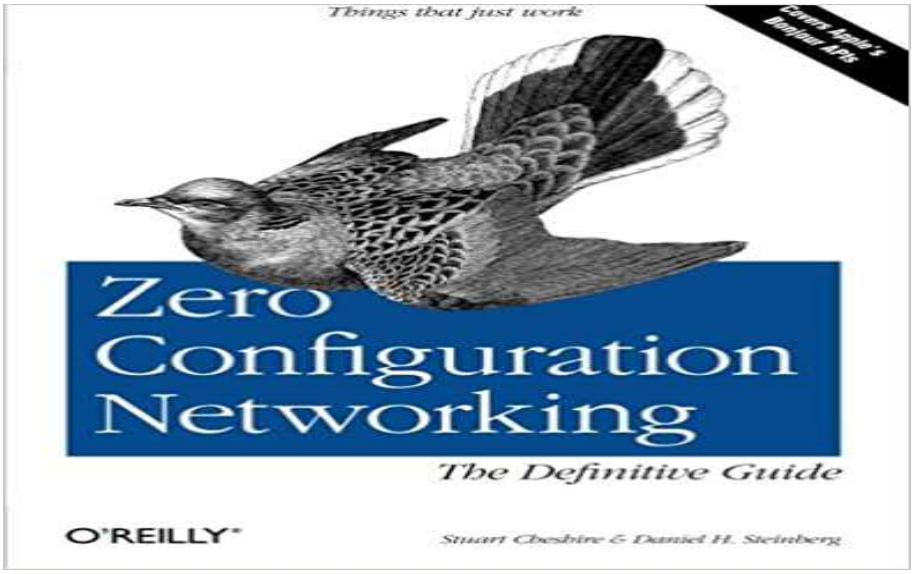


Figure 3 : Zeroconf

Source : https://www.bing.com/saves?&FORM=savim1

Traditionnellement la plupart de ce travail est réalisé par DHCP et DNS. Cependant, la conception décentralisée de Zeroconf est plus appropriée dans certaines situations, comme pour les réseaux ad-hoc. De plus, cela se fait sans aucune configuration (mise à part celle de l'installation de Zeroconf). Ce concept que nous allons étudier est né grâce à l'impulsion de Stuart Cheshire qui réfléchissait au remplacement d'Appletalk et avait eu l'idée d'un NBP/IP (AppleTalk Name Binding Protocol) en multicast. Un groupe de travail de l'IETF a été créé : Zéro Configuration Networking alias Zeroconf.

Zeroconf résout actuellement trois problèmes :

- Allocation dynamique d'adresse IP sans serveur DHCP
- Résolution de noms et adresse IP sans serveur DNS
- Recherche de services sans annuaire

2.3 Protocole DHCP

DHCP signifie Dynamic Host Configuration Protocol. Il s'agit d'un protocole qui permet à un ordinateur qui se connecte sur un réseau d'obtenir dynamiquement (c'est-à-dire sans intervention particulière) sa configuration (principalement, sa configuration réseau). Vous avez qu'à spécifier à l'ordinateur de se trouver une adresse IP tout seul par DHCP. Le but principal étant de la simplification de l'administration d'un réseau [4].

Le protocole DHCP sert principalement à distribuer des adresses IP sur un réseau, mais il a été conçu au départ comme complément au protocole BOOTP (Bootstrap Protocol) qui est utilisé par exemple lorsque l'on installe une machine à travers un réseau (BOOTP est utilisé en étroite collaboration avec un serveur TFTP sur lequel le client va trouver les fichiers à charger et à copier sur le disque dur). Un serveur DHCP peut renvoyer des paramètres BOOTP ou de configuration propre à un hôte donné.

➢ FONCTINNEMENT DE DHCP

Il faut dans un premier temps un serveur DHCP qui distribue des adresses IP. Cette machine va servir de base pour toutes les requêtes DHCP, aussi elle doit avoir une adresse IP fixe. Dans un réseau, on peut donc n'avoir qu'une seule machine avec adresse IP fixe, le serveur DHCP.

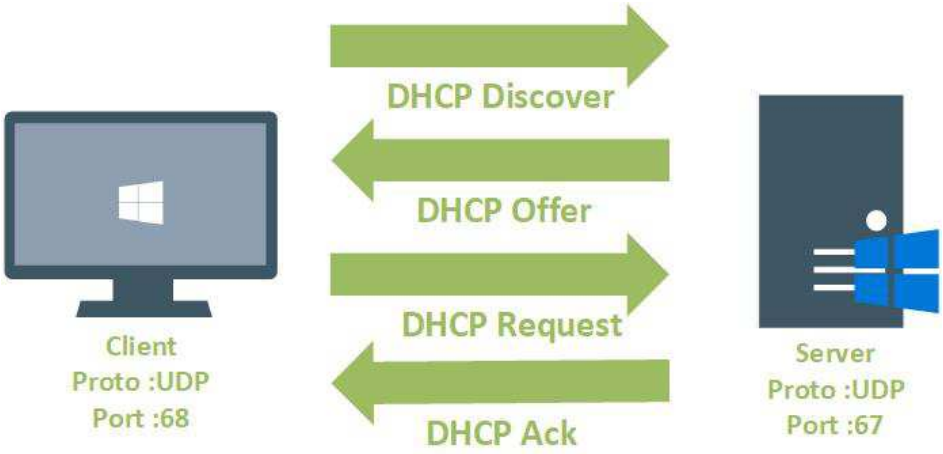


Figure 4 : fonctionnement de DHCP

Source : https://th.bing.com/th

Le mécanisme de base de la communication est BOOTP (avec trame UDP). Quand une machine est démarrée, elle n'a aucune information sur sa configuration réseau, et surtout, l'utilisateur ne doit rien faire de particulier pour trouver une adresse IP. Pour faire ça la technique utilisée est le broadcast (diffusion) : pour trouver et dialoguer avec un serveur DHCP, la machine va simplement émettre un paquet spécial de broadcast (broadcast sur 255.255.255.255 avec d'autres informations comme le type de requête, les ports de connexion…) sur le réseau local. Lorsque le serveur DHCP recevra le paquet de broadcast, il renverra un autre paquet de broadcast (n'oublier pas que le client n'a pas forcement son adresse IP est que donc il n'est pas joignable directement) contenant toutes les informations requises pour le client [5].

On pourrait croire qu'un seul paquet peut suffire à la bonne marche du protocole. En fait, il existe plusieurs types de paquets DHCP susceptible d'être émis soit par le client pour le ou les serveurs, soit par le serveur vers un client :

- **DHCPDISCOVER :** pour localiser les serveurs DHCP disponible.
- **DHCPOFFER :** réponse du serveur à un paquet DHCPDISCOVER, qui contient les premiers paramètres.
- **DHCPREQUEST :** requête diverse du client pour par exemple prolonger son bail.
- **DHCPACK :** réponse du serveur qui contient des paramètres et l'adresse IP du client.
- **DHCPNACK** : réponse du serveur pour signaler au client que son bail est échu ou si le client annonce une mauvaise configuration réseau.
- **DHCPDECLINE :** le client annonce au serveur que l'adresse est déjà utilisée.
- **DHCPRELEASE :** le client libère son adresse IP**.**
- **DHCPINFORM :** le client demande des paramètres locaux, il a déjà son adresse IP.

Le premier paquet émis par le client est un paquet de type DHCPDISCOVER. Le serveur répond par un paquet DHCPOFFER, en particulier pour soumettre une adresse IP au client. Le client établit sa configuration, puis fait un DHCPREQUEST pour valider son adresse IP (requête en broadcast car DHCPOFFER ne contient pas son adresse IP). Le serveur répond simplement par un DHCPACK avec l'adresse IP pour confirmation de l'attribution. [5]

3. Découverte de service basé sur DNS (DNS-SD)

DNS-SD (DNS Based Service Discovery) permet aux clients de découvrir une liste nommée d'instances de service et de résoudre ces services en noms d'hôtes à l'aide de requêtes DNS standard. La spécification est compatible avec le serveur DNS monodiffusion et le logiciel clients existants, mais fonctionne aussi bien avec mDNS dans un environnement sans configuration. Chaque instance de service est décrite à l'aide d'un enregistrement DNS SRV et DNS TXT. Un client découvre la liste des instances disponibles pour un type de service donné en interrogeant l'enregistrement DNS PTR du nom de ce type de service ; le serveur renvoie zéro ou plusieurs noms de la forme <Service>. <Domain>, chacun correspondant à une paire d'enregistrements SRV/TXT. Le dossier SRV se résout au nom de domaine fournissant l'instance, tandis que le TXT peut contenir des paramètres de configuration spécifiques au service. Un client peut alors résoudre l'enregistrement A/AAA du nom de domaine et se connecter au service. Les types de services sont fournis selon le principe du premier arrivé, premier servi.

Un paquet DNS est défini comme suite :

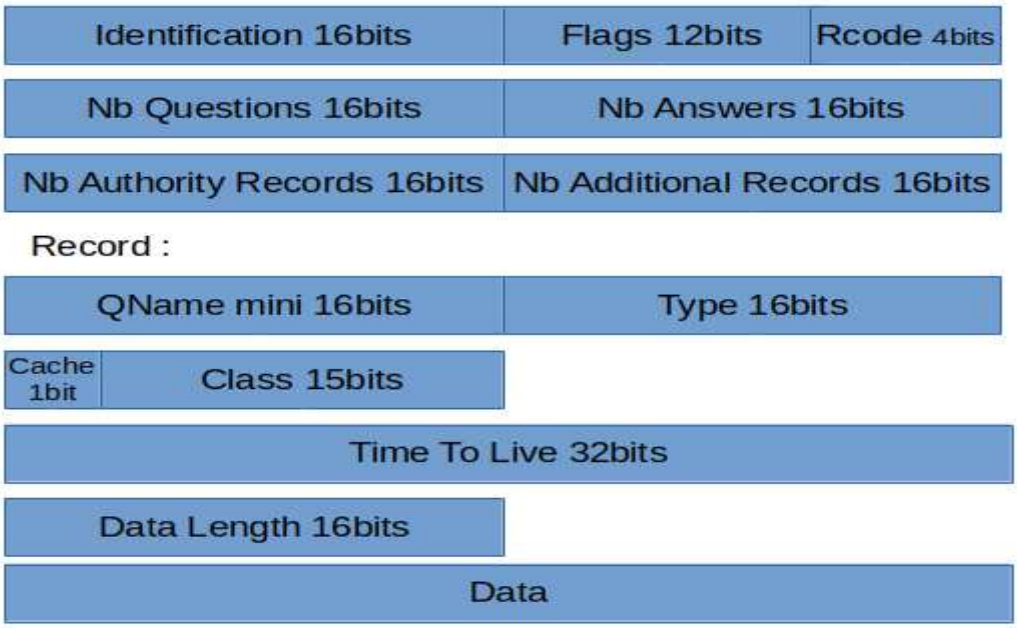


Figure 5 : datagramme du message DNS

C'est donc 4 listes d'enregistrements ayant tous la même structure :

- Nb Questions (qn) : pour les questions, sont les requêtes envoyées par une machine ;
- Nb Answers (Ans) : pour les réponses à des questions reçues ;
- Nb Authority Records (auth) : pour la gestion des autorités de résolution
- Nb Additional Records (add) : pour les informations additionnelles aux réponses.

3.1 Objectif du système DNS-SD

Le système de découverte de service basé sur DNS (DNS-SD) a été conçus pour des objectifs principaux. Parmi les nombreuses propriétés qu'un bon protocole de découverte de service doit-on, trois d'importance particulière sont :

- La possibilité de demander des services d'un certain type dans un certain domaine logique, et recevoir en réponse une liste de noms d'instances de ces services (navigation sur le réseau ou énumération des instances).
- Une fois que l'instance est obtenue, le système à la capacité de résoudre ce nom d'instance aux informations requises par un client qui doit réellement utiliser le service, c'est-à-dire au minimum une adresse IP et numéro de port (Résolution d'instance de service).
- Les noms d'instance doivent être relativement persistants. Si un utilisateur sélectionne son imprimante par défaut dans une liste de choix disponibles aujourd'hui, puis demain, il devrait encore pouvoir imprimer sur cette imprimante, même si l'adresse IP et/ou le numéro de port où les résidences du service -sans que l'utilisateur (ou son logiciel) devant répéter la première étape une seconde fois.

De plus le protocole devrait être si simple à mettre en œuvre que n'importe quel appareil capable de mettre en œuvre IP ne devrait pas avoir de difficulté à mettre en œuvre le logiciel de découverte de service également.

3.2 Fonctionnement du DNS-SD

Au contraire de SSDP, DNS-SD est un protocole poids plume. Il est basé sur le DNS, ce qui fait sa simplicité. Il peut s'appuyer indifféremment sur la version classique du DNS ou sur mDNS. Il n'utilise donc pas de port ni d'adresse IP en propre. Ce n'est pas un nouveau protocole circulant sur le réseau, aucune attribution IANA n'est demandée pour ce protocole.

DNS-SD reprends les concepts de la RFC 2782 (DNS SRV) permettant la découverte de service et y rajoute la découverte d'instance de service. L'utilisation d'enregistrement de type SRV n'est pas massivement utilisée dans les serveurs DNS.

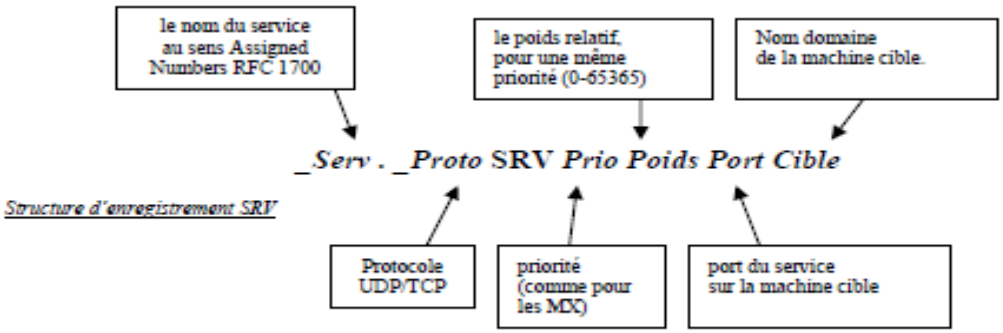


Figure 6 : Structure d'enregistrement SRV

Le poids relatif est un nouveau champ dans les enregistrements DNS qui permet de donner une probabilité d'accès entre les différents serveurs. Si par exemple deux machines de même priorité ont des poids de 1 et 3, alors la répartition sera de 25% et 75% respectivement.

1. Les clients demandent au serveur DNS un service (ou un protocole) et il leur retourne la liste des machines du domaine qui le proposent. L'idée principale de DNS-SD est d'ajouter un niveau d'indirection. Au lieu de faire une requête sur les enregistrements de type SRV sur le couple <service>. <Domaine>, le client fait une requête sur les enregistrements de type PTR. Le résultat de cette requête est une liste de zéro ou plusieurs enregistrements PTR qui contiennent des Noms d'Instance de Service sous la forme : <Instance>. <Service>. <Domaine>.
   - La partie <Instance> est un simple label DNS qui contient du texte encodé en UTF-8 suivant la RFC 3629. Il n'y a aucune restriction sur les caractères utilisés. La longueur des Instances est limitée à 63 octets.
   - La partie <Service> est identique à celle des enregistrements SRV du RFC 2782.
   - La partie <Domaine> est un nom de domaine du DNS.
2. Une fois que le client a un Nom d'Instance de Service et qu'il veut accéder à ce service, il fait une nouvelle requête DNS en demandant cette fois l'enregistrement SRV de ce nom.
3. Le résultat de cette requête est un (ou plusieurs) enregistrement(s) SRV contenant le port et la machine cible où l'on peut trouver ce service. Dans le cas exceptionnel où il y'a plusieurs enregistrements SRV retournés, le client doit interpréter les priorités et les poids. En général il y a un seul enregistrement SRV dont la priorité et le poids sont nuls.

Le protocole spécifie aussi des règles de format pour les enregistrements de type TXT qui vont permettre de passer des paramètres supplémentaires. [6]

3.3 Utilisation de DNS-SD

➢ Actions réalisées par l'utilisateur

Un utilisateur qui souhaite DNS-SD n'a pas besoin de connaitre toutes les spécificités du protocole. Il doit juste connaitre le nom du service et savoir utiliser un logiciel appelé client DNS-SD.

En général, le client connait le domaine à explorer pour découvrir les services. L'utilisateur n'a qu'à choisir le service désiré parmi une liste dans un menu déroulant. Une fois, la recherche de service effectuée, l'utilisateur visualise les instances disponibles et en sélectionne une. Le client peut ensuite utiliser le service.

➢ Actions réalisées en interne

Lors de la découverte du service, le client a fait une requête PTR. Il a reçu en réponse les enregistrements PTR correspondants, c'est-à-dire la liste des instances du service. Il s'en est suivi l'affichage de ces instances.

Lorsque l'utilisateur sélectionne l'instance qu'il souhaite utiliser, le client émet cette fois-ci une requête SRV et obtient ainsi les informations nécessaires pour utiliser le service. En réalité, le client reçoit en réponse les enregistrements SRV et TXT correspondants à l'instance.

Toutes les opérations effectuées en interne sont transparentes pour l'utilisateur.

L'utilisateur plus expérimenté pourra paramétrer manuellement le client DNS-SD s'il le souhaite. Il sélectionnera ainsi un service DNS-SD, le domaine à explorer, l'interface réseau à utiliser.

3.4 Types d'enregistrements du DNS-SD

Pour effectuer une requête de découverte de service, on doit demander plusieurs types d'enregistrements. Cependant DNS-SD utilise trois types d'enregistrements pour mettre en place la découverte de services.

- Les enregistrements PTR qui énumèrent les instances d'un service DNS
- Les enregistrements SRV et TXT qui contiennent les détails des instances de service, comme le nom d'hôte et le port.
- Les enregistrements A qui contiennent les adresses IP des noms d'hôte récupérés.

4. Applications de découverte de services (DNS-SD)

Les principales implémentations actuelles de Zeroconf sont résumées ci-dessous.

4.1 Application Bonjour



Figure 7: Application Bonjour d'Apple

Source : https://mcreative.tistory.com/199

Bonjour est la technologie réseau standard d'Apple conçu pour permettre aux appareils et aux services de se détecter mutuellement sur le même réseau. Par exemple les iPhones et les iPads utilisent Bonjour pour détecter des imprimantes AirPrint, et les iPhone et iPad et les ordinateurs Mac utilisent Bonjour pour découvrir des appareils compatibles avec AirPlay, tels que les Apple TV. Certaines applications utilisent Bonjour pour le partage et la collaboration pair à pair.

Bonjour fonctionne à l'aide d'un trafic en multidiffusion pour signaler la disponibilité des services. Le trafic en multidiffusion n'est généralement pas acheminé sur les sous-réseaux. Par conséquent, les Apple TV et les imprimantes AirPrint doivent être connectées aux mêmes sous-réseaux IP que les iPhone et iPad et les ordinateurs Mac qui les utilisent. Pour les réseaux de petite envergure, cette approche constitue la norme.

Bonjour sert principalement à détecter des services et des appareils sur les petits réseaux, mais vous pouvez tout de même le configurer pour des réseaux de plus grande envergure à l'aide de

passerelle Bonjour ou d'une technologie similaire. Les réseaux utilisant ce type de fonctionnalités pour AirPlay doivent être configurés de façon à signaler les services _airplay._tcp et _raop._tcp. Et même s'il est possible d'utiliser des configurations DNS avancées pour signaler des services Bonjour sur des réseaux d'entreprise. [6]

Les clients MacOs et Bonjour pour Windows peuvent également utiliser le DNS (Domain Name System) monodiffusion conventionnelle pour détecter des services annoncés dans des domaines accessibles partout dans le monde.

4.2 Application Avahi

Avahi est une bibliothèque de logicielle libre implémentant les protocoles définis par le projet Zeroconf : mDNS, DNS-SD, et IPv4ALL. Ses fonctionnalités sont sensiblement les mêmes que Bonjour, c'est-à-dire l'auto-configuration IP, la découverte et le partage de services sur un même réseau local.

Avahi est disponible sur les systèmes GNU/Linux et *BSD et reste totalement compatible avec Bonjour. Par ailleurs, le créateur des protocoles mDNS et DNS-SD (et initiateur du projet Zeroconf), Stuart Cheshire, a déclaré que la société pour laquelle il travaille, Apple, suivait de très près le travail des développeurs d'Avahi et que ce dernier pourrait, à terme, remplacer Bonjour. [6]

En revanche Avahi ne se limite pas forcément aux services du réseau local. En effet il est possible de le configurer pour propager l'information sur plusieurs réseaux d'une même passerelle, voire même sur des réseaux plus distants vu qu'il utilise la technologie de multidiffusion (multicast).



Figure 8 : Avahi

4.3 Application I-phone

Il nous a paru intéressant de voir que DNS-SD était tout récemment utilisé par des applications I-phone pour découvrir les services disponibles avec sa connexion WI-FI. Cela présage un bon avenir pour DNS-SD.

Netbrowser recherche dans votre réseau les services disponibles enregistrés dans le DNS-SD. Avec cette application pratique, vous pouvez savoir si une imprimante est disponible, si des serveurs de fichiers se cachent ou si un hôte accepte les connexions VNC. J'utilise cette application pour tester mes applications de services compatibles DNS-SD. Il suffit de lancer votre service activé par DNS-SD et NetBrowser le trouvera [7].

Au premier lancement, NetBrowser vous présentera une liste de domaines à parcourir. Habituellement, cette liste ne contient que le domaine local. Cependant, il y a des cas où d'autres domaines peuvent également être parcourus. Lorsque vous touchez le domaine, une liste de tous les services enregistrés sur ce domaine apparaît. Il suffit de toucher un service pour que la liste des hôtes qui fournissent ce service s'affiche. En sélectionnant l'hôte, toutes les informations le concernant s'affichent. Ces informations comprennent le nom de l'hôte, l'adresse, le port du service et toute autre information provenant de l'enregistrement du service. Si vous appuyez sur Annuler à tout moment, vous reviendrez à la liste des domaines.

5. Scénarios de découverte de services

Dans cette partie, nous passons en revue les scénarios de découverte de services courants et discuter des menaces à la vie privée et de leurs exigences en matière de confidentialité. Dans les trois de ces scénarios courants, l'attaquant est du type passif on-relier.

5.1 Client Privé et Serveur public

Le scenario de découverte privé le plus simple implique peut-être un seul client se connectant à un serveur public via un réseau public. Un exemple courant serait un voyageur utilisant une imprimante accessible au public dans un centre d'affaires, dans un hôtel ou dans un aéroport.

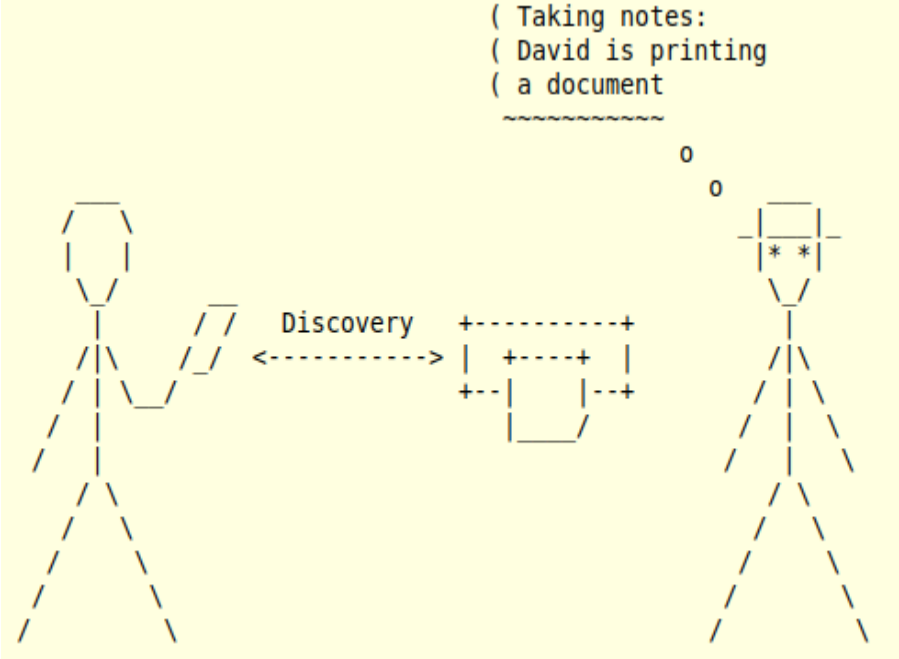


Figure 9 : scénario de découverte 1

Source : www.iett.org/

Dans ce scénario, le serveur est public et veut être découvert, mais le client est privé. L'adversaire sera à l'écoute du trafic réseau essayant d'identifier les appareils des visiteurs et leur activité. L'identification des dispositifs conduit à l'identification des personnes, soit pour la surveillance de ces individus dans le monde physique, soit comme « tape préalable à une cyberattaque ciblée [8].

L'exigence dans ce scénario est que l'activité de découverte ne doit pas divulguer l'identité du client.

5.2 Client privé et Serveur privé

Le deuxième scénario de découverte privée implique un client privé se connectant à un serveur privé. Un exemple courant serait deux personnes s'engageant dans une application collaborative dans un lieu public, comme le salon d'un aéroport.

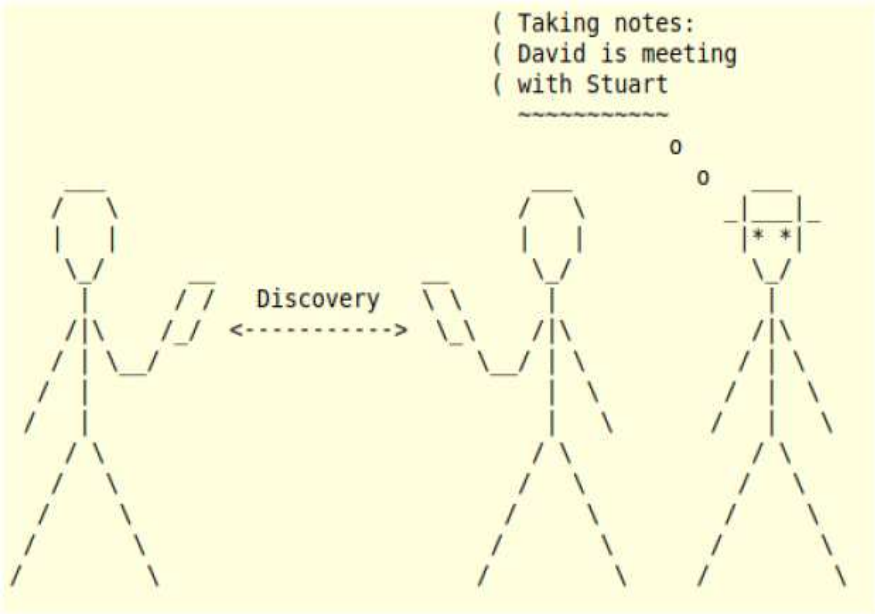


Figure 10 : scénario de découverte 2

Source : www.iett.org/

Dans ce scénario, l'application collaborative sur l'un des appareils agira en tant que serveur et l'application sur l'autre appareil agira en tant que client. Le serveur veut être découvert par le client mais n'a aucun désir d'être découvert par quelqu'un d'autre. L'adversaire écoutera le trafic réseau, tentera de découvrir l'identité des appareils comme dans le premier scénario et tentera également de découvrir les modèles de trafic, car ces modèles révèlent les interactions commerciales et sociales entre les propriétés des appareils [9].

L'exigence dans ce scénario est que l'activité de découverte ne doit pas divulguer l'identité du client ou du serveur ni révéler les interactions commerciales et sociales entre les propriétaires des appareils.

5.3 Client et serveur portable

Le troisième scénario de découverte privée implique des appareils portables. Un exemple typique serait la montre au poignet de quelqu'un qui se connecte au téléphone dans sa poche.

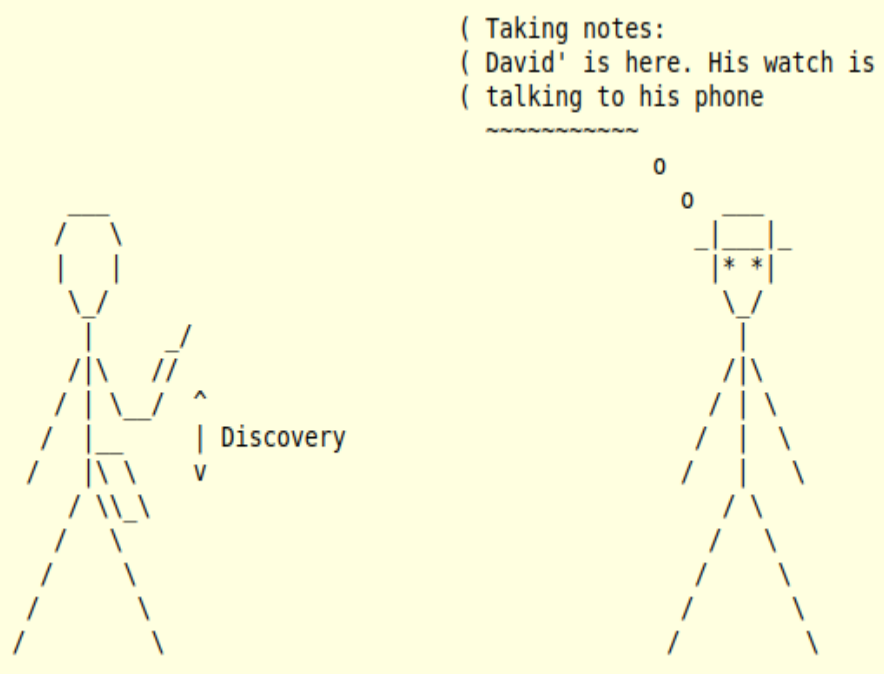


Figure 11 : scénario de découverte 3

Ce troisième scénario est à bien des égards similaires au deuxième scénario. Cela implique deux dispositifs, l'un agissant en tant que serveur et l'autre agissant en tant que client, et cela conduit ç la même exigence que le trafic de découverte ne révèle l'identité ni du client ni du serveur.

La principale différence est que les appareils sont gérés par un seul propriétaire, ce qui peut conduire à différentes méthodes pour établir des relations sécurisées entre les appareils. L'accent est également mis sur dissimilation du type d'appareils que la personne porte.

En plus de suivre l'identité du propriétaire des appareils, l'adversaire s'intéresse aux caractéristiques des appareils, telles que le type, la marque et le modèle. L'identification du type d'appareil peut conduire à d'autres attaques, du vol au piratage spécifique à l'appareil. La combinaison d'appareils portés par la même personne fournira également une « empreinte digitale » de la personne, risquant son identification. [13]

Ce scénario représente également le cas général de l'introduction d'appareils Internet des Objets (IoT) dans des lieux publics. Un appareil IoT portable peut agir comme un client DNS-SD/mDNS, ce qui permet aux attaquants de déduire des informations sur les propriétaires des appareils. Bien que l'attaquant puisse être une personne, comme dans l'exemple de figure, cela pourrait également être utilisé de manière abusive pour la collecte de données à grande échelle en installant des serveurs fixes de suivi des appareils IoT dans des lieux publics fréquentés.

Les problèmes décrits dans la section 5.2 tels que l'identification des personnes ou l'utilisation des informations pour des attaques ciblées, s'appliquent également ici.

6. Différents protocoles de découverte de service

Les protocoles de découverte de services fournissent des mécanismes pour la découverte spontanée et dynamique de services disponibles dans le réseau. Dans ce qui suit, nous donnons

un aperçu sur les protocoles les plus représentatifs et nous montrons comment chaque protocole assure les différentes fonctionnalités de la découverte de services.

6.1 Protocole mDNS

Multicast Domain Name System(mDNS) est un protocole de découverte de service de type DNS pour résoudre les noms d'hôtes en adresse IP dans un réseau local sans utiliser de serveur DNS de monodiffusion. Il peut être utilisé sans infrastructure supplémentaire ni serveur DNS dans le réseau. Le protocole fonctionne sur des paquets UDP multicast IP à travers lesquels un nœud du réseau local demande les noms de tous les autres nœuds. Le nœud client envoie un message de requête pour répondre par un nœud avec un nom spécifique. Lorsque le nœud portant le nom correspondant reçoit la requête, il répond par un message de réponse multicast contenant son adresse IP. Étant une réponse multicast, l'adresse IP et le nom du périphérique cible sont également enregistrés par tous les périphériques (nœuds) du réseau dans leurs caches locaux.

Dans un tel réseau local, les adresses IP peuvent changer à un moment donné, mais pas les noms. Ce protocole est vraiment utile car il ne nécessite aucune infrastructure supplémentaire (serveur DNS dans le réseau) et peut être utilisé pour gérer des appareils sans aucune configuration ou administration manuelle. Le protocole peut être mis en œuvre malgré les défaillances de l'infrastructure. Développé par l'IETF, le protocole mDNS est défini dans la RFC6762. [10]

6.2 Protocole DNS-SD

DNS Service Discovery (DNS-SD), cette pile de protocole utilise des messages DNS standard pour découvrir des services dans un réseau IoT. Basé sur mDNS, DNS-SD est utilisé pour résoudre les services disponibles dans un réseau. La découverte de service est implémentée en deux étapes : dans la première étape, les noms d'hôte des fournisseurs de services sont résolus et dans l'étape suivante, les adresses IP sont associées aux noms d'hôte à l'aide de mDNS. Il est important d'identifier les noms d'hôte car les adresses IP peuvent changer dans le réseau.

Comme dans mDNS, l'adresse IP et le nom d'hôte du périphérique cible sont communiqués sous forme de réponse multidiffusion et chaque nœud (périphérique IoT connecté au réseau) met à jour le nom d'hôte et l'adresse IP associée à la réception de la réponse. C'est le protocole qui maintient les noms d'hôtes constants dans le réseau malgré les appareils ou les nœuds ont des adresses IP dynamiques. Comme le nom d'hôte reste toujours constant dans le réseau, il est possible d'identifier les appareils de manière unique et fiable au sein du réseau IoT local.

Comme mDNS cette pile de protocoles ne nécessite aucune infrastructure supplémentaire (DNS Name Server dans le réseau) ni aucune configuration ou administration manuelle des appareils IoT connectés.

6.3 Protocole UPnP

Géré par Open Connectivity Foundation, Universal Plug and Play (UPnP) est une pile de protocoles qui permet aux périphériques d'un réseau de se découvrir et de découvrir leurs capacités ainsi que de configurer des fonctions réseau telles que le partage de données et la communication.



Figure 12 : Universal plug in Play

Source : https://cdn.freebiesupply.com/logos/large/2x/upnp-logo-png-transparent.png

Il existe trois éléments constitutifs de tout réseau UPnP : les appareils, les services et les points de contrôle. Les appareils sont les blocs de base du réseau ou chaque appareil fournit un ou plusieurs services spécifiques. Les services sont vus comme un ensemble d'actions qui peuvent être mises en œuvre par l'appareil. Les points de contrôle identifient les appareils par description d'appareil et de service et répondent au nœud client en appelant les services demandés. Dans ce protocole, la découverte des appareils et des services se fait automatiquement et de manière autonome une fois qu'un nœud entre dans le réseau. Ainsi, il n'est plus nécessaire de configurer manuellement les nœuds (périphériques IoT) pour découvrir les périphériques et services disponibles sur le réseau.

Ce protocole est basé sur le routage TCP/IP. Les appareils sont identifiés par leurs indicateurs de ressources universelles (URI) et les services sont découverts à l'aide du protocole http (HyperText Transfer Protocol). Un appareil connecté au réseau se configure automatiquement en acquérant une adresse TCP/IP et en indiquant les services disponibles via http. Les appareils utilisent alors XML pour communiquer entre eux et s'indiquer les capacités de chacun (services à offrir).

UPnP était auparavant une norme propriétaire de Microsoft, mais est maintenant une norme ouverte. Il est tout à fait adapté aux réseaux IoT locaux ou les appareils devraient communiquer entre eux sans aucune infra structure supplémentaire. Comme UPnP, les appareils mobiles peuvent être utilisés pour se connecter aux imprimantes et aux scanners dans un bureau automatisé. De même, il peut être utilisé pour transférer des photos d'un appareil photo vers des appareils mobiles ou un ordinateur portable [11].

6.4 Protocole SSDP

Simple Service Discovery Protocol (SSDP) – Le protocole de service de découverte simple est utilisé dans les réseaux UPnP pour découvrir les services disponibles.

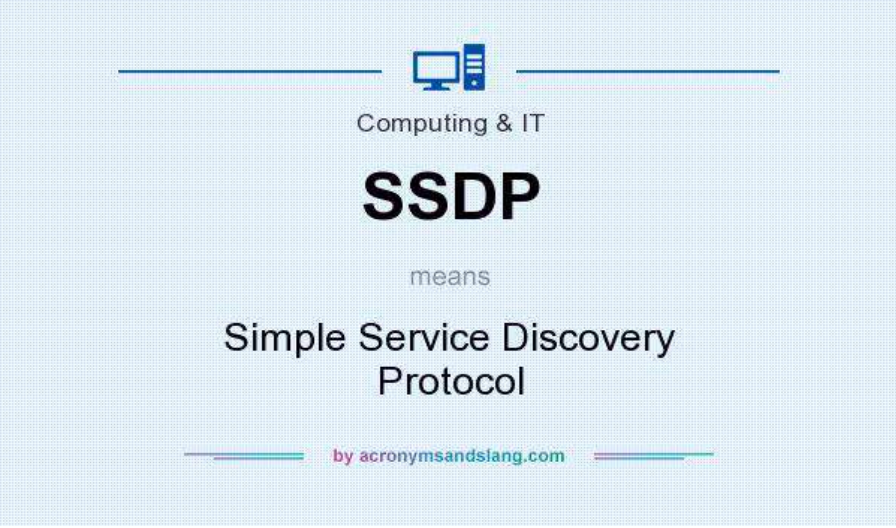


Figure 13 : Simple Service Discovery Protocol

Il est utilisé par les points de contrôle du réseau UPnP pour rechercher les appareils, les services qu'ils offrent et leur disponibilité à un moment donné. Le point de contrôle envoie une requête de recherche multicast à laquelle répond le dispositif offrant le service demandé. Les appareils et services qu'ils proposent sont identifiés respectivement par des descripteurs d'appareil et de service. Le point de contrôle utilise des appareils et des descriptions de service pour rechercher les services demandés dans le réseau.

Il existe également de nombreuses plates-formes de découverte de service public auxquelles les appareils IoT peuvent demander des services. [12]

7. Conclusion

Multicast DNS Service Discovery est une technique prévalent utilisée pour la distribution et la découverte de services. Elle utilise les deux couches supérieures de la pile Zeroconf, à savoir la découverte de services DNS construite sur le DNS multidiffusion et permet aux utilisateurs d'offrir et d'utiliser des services tels que la synchronisation de périphériques, le partage de fichiers et le chat, lorsqu'il rejoigne un réseau local sans aucune configuration manuelle.

Contrairement à de nombreux autres mécanismes de découverte de service où les clients doivent être configurés avec l'adresse du serveur de découverte de service approprié à utiliser (et sont limités à l'utilisation d'un seul serveur de découverte à la fois), un grand avantage de la découverte de service basée sur DNS est que les clients continuent d'utiliser leur résolveur récursif existant, inchangé, et ce résolveur continue de parler à de nombreux serveurs faisant autorité au nom du client. Tout cela n'est une invention nouvelle ; il s'agit simplement d'utiliser le protocole DNS et les mécanismes déjà en place.

Chapitre 2 : Généralités sur l'Internet des objets

1. Introduction

L'internet des objets ou IoT (Internet of Things), est un paradigme émergent dans le monde des réseaux informatiques. Il peut être défini comme une évolution et extension de l'internet de nos jours pour l'inclusion de tous les objets et les endroits dans notre entourage (réfrigérateurs, thermostat, maisons, véhicules, routes, etc.). Le concept promoteur de l'IoT va nous simplifier la vie, nous faire gagner du temps, décharger notre cerveau de la mémorisation de données logistiques (itinéraires, temps de prise de médicaments, etc.). Ainsi, l'accès omniprésent à différents types d'informations permettant la sophistication du mode de vie et une amélioration significative de la qualité des services dans différents domaines.

L'IoT qui est une nouvelle vague de l'internet, est en réalité une partie naissante de l'internet du futur, appelé l'internet de tous les objets ou IoE (Internet of Everythings), qui vise à interconnecter les gens, les données, et tous les objets, de telle sorte qu'il y'ait une fusion entre le monde réel (physique) et le monde numérique (virtuel) ; les objets du monde physique vont être incorporés dans le monde virtuel de l'internet. Cela fait appel à de nouvelles tendances et innovations que ce soit sur le plan architectures de communications ou sur le plan présentation et explication des services.

Ce chapitre est consacré à la présentation du domaine de l'internet des objets et les aspects qui s'y rapportent.

2. Historique

Entre les craintes et les promesses que soulève l'aventure commune, hier improbable, du web et des objets, un chemin, déjà presque une histoire !

Le concept de l'Internet des objets nait en 1999, au MIT (*Massachusetts Institute of Technology*). Le MIT Auto-ID laboratoire est dédié à la création d'objets connectés à l'aide de l'identification par radiofréquence et les réseaux de capteurs sans fil.

L'appellation « **internet des objets** », a vu le jour la même année. On la doit à Kevin Ashton, un collaborateur de Procter & Gamble qui voulait qualifier le lien entre la technologie RFID (*Radio Fréquency IDentification*) et internet. Autrement dit des objets embarquant des étiquettes RFID et internet. Les informations contenues dans la puce électronique d'un tag RFID peuvent être des identifiants et éventuellement d'autres données complémentaires.

À l'époque, les ordinateurs et Internet reposaient presque entièrement sur l'information fournie par les êtres humains. Également à cette époque, nous collections presque toutes les données et informations en les tapant sur diverses machines. Comme Kevin Ashton l'a dit le 22 juin 2009, dans le RFID Journal : « Le problème, c'est que les gens ont une attention, une précision et un temps limités, et ne sont donc pas pleinement capables de saisir la totalité des données échangées dans le monde physique. » Au XXe siècle, les ordinateurs étaient des cerveaux sans intelligence, puisqu'ils ne connaissaient que ce que nous leurs avions transmis, ceci représentant donc une importante limitation, puisqu'ils existent des milliards de fois plus d'informations qui

peuvent être capturées par des capteurs que ce que les êtres humains peuvent saisir via un clavier ou scanner avec un code à barres.

Aujourd'hui, grâce à l'Internet of Things et aux capteurs intelligents, les ordinateurs sont en mesure de fonctionner de manière quasi autonome. Un bon exemple est la détermination de la position par GPS : Le GPS civil a été approuvé pour la première fois par le Congrès américain en 2000, et les systèmes GPS des téléphones mobiles ont été testés pour la première fois en 2004. Aujourd'hui encore, un monde sans GPS est difficilement concevable, puisqu'il nous aide à nous diriger et nous retrouver (presque) partout où nous allons. [13]

3. Les types d'objets

Avec l'avènement de l'internet des objets, la connexion internet acquiert une troisième dimension ; en plus de la possibilité de se connecter n'importe quand et n'importe où, il est désormais possible d'être connecté avec n'importe quel objet. De plus, les objets connectés sont identifiés de façon unique et sont capable de récolter des informations environnementales (liées aux changements des paramètres de l'environnement, comme la température) ou comportementales (issues des variations d'état de l'objet lui-même ou des objets contextuels), de les traiter et de les communiquer sur Internet. D'où leur appellation par objets intelligents.

On distingue différents types de dispositifs connectés à l'IoT, ou qui font connecter d'autres objets à Internet, dont on cite principalement :

3.1 Les nœuds capteurs



Figure 14 : nœud capteur

Source : https://moodle.utc.fr/file.php/498/SupportWeb/res/telosb-verso.png

Les capteurs dans l'IoT permettant de récolter des informations contextuelles concernant les objets dans lesquels ils sont intégrés, ou les environnements sur lesquels ils sont déployés. Les capteurs communiquent les informations collectées sur internet d'une manière directe ou indirecte, tout dépend du modèle adopté pour l'intégration des réseaux de capteurs

3.2 Les objets d'identification

Codes barre, marqueur RFID et autres dispositifs miniaturisés qui servent à l'identification et la traçabilité des objets sur lesquels ils sont collé pouvant être collés sur les objets d'usage courant (exemple : vêtements, marchandises, livres, véhicules, etc.). Ce type de dispositifs nécessite qu'il y'ait un lecteur pour récupérer leurs données qui seront par la suite téléchargées

sur un serveur et deviennent alors accessibles via le système d'information d'une organisation ou directement sur internet.

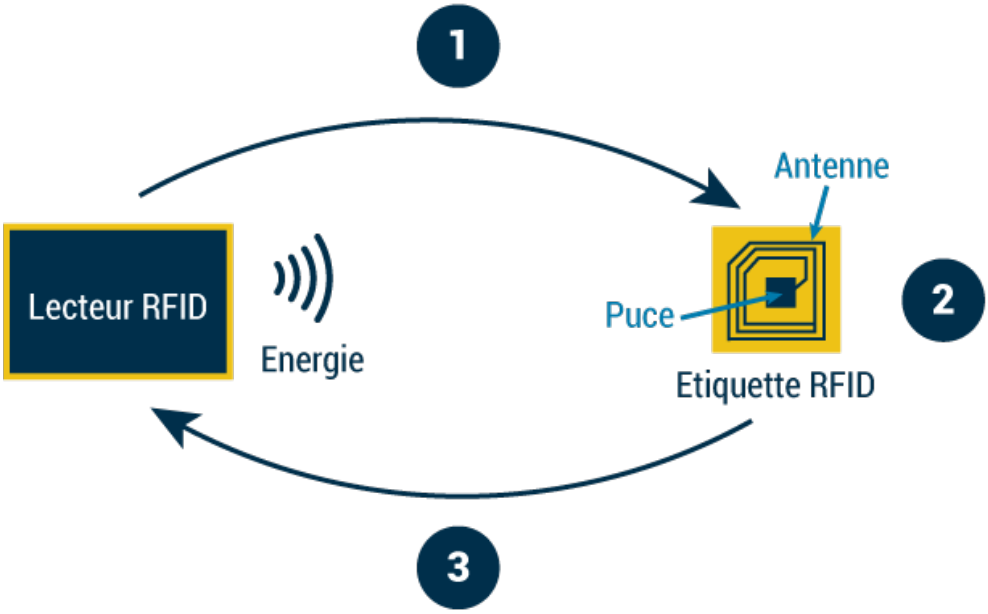


Figure 15 : Objets d'identifications

3.3 Les drones



Figure 16 : drone connecté

Un drone désigne un aéronef miniature sans pilote, pouvant porter des charges utiles, communiquer et exécuté des commandes en toute flexibilité. Les drones sont utilisés dans des applications civiles aussi bien que dans des applications militaires pour accomplir des missions bien déterminées. On entend parler de l'efficacité de l'utilisation des drones dans le domaine commercial pour par exemple, les livraisons à domicile des commandes faites sur Internet. Aussi des opérations de sauvetage, d'exploration et de surveillance sont réalisable par les drones dans le contexte des applications militaires. Bien que la technologie (ou bien son prototype) des drones en elle-même existait depuis bien longtemps, son exploitation idéale dans différentes applications demeure modeste. Récemment, les drones sont élus pour faire une importante part de l'internet de la future, soit en tant que objets intelligents terminaux rapportant des données de contrôle, soit en tant que routeurs particuliers (mobiles, volants) de données entre les parties connectées à internet. Comparés aux capteurs qui sont le plus souvent stationnaires ou dans certains cas mobiles mais dans tous les cas, manquent de l'aspect aérien, un drone parvient très

efficacement à donner une vision aérienne sur l'état de la zone à contrôler même dans les zones isolée et/ou inaccessibles (là où il est difficile d'installer une infrastructure terrestre avec des points d'accès et des stations de base).

3.4 Les smartphones et tablettes

Les smartphones et les tablettes qui sont déjà connectés à internet par le biais de diverses technologies (WIFI,3G 4G) permettant aux utilisateurs de communiquer à distances avec les autres types d'objets connectés dans l'IoT. Les objets intelligents peuvent rapporter en temps réel l'état actuel aux utilisateurs via internet. Dans ce cas, les utilisateurs reçoivent des e-mails ou simplement des messages d'alertes sur leurs Smartphones ou tablettes, tout dépend de l'application. Il est même possible que les utilisateurs supervisent ou ordonnent leurs objets connectés à distance, via leurs smartphones ou tablettes. [14]

La figure ci-dessous représente les principaux types d'objets dans l'IoT



Figure 17 : Typologie des objets dans l'IoT

4. Technologie fondatrice d'IoT

Bien que l'internet des objets soit une notion relativement nouvelle, les technologies qui la rendent possible existaient depuis quelques années déjà. On parle alors des réseaux de capteurs sans fil et de la technologie d'identification par radio fréquence. Les évolutions observées par les technologies sans fil et le domaine des réseaux de télécommunication d'une part, et l'internet de l'autre part, ont permis d'ouvrir de nouvelles perspectives pour ces technologies, qui ont pu s'instaurer efficacement dans notre vie quotidienne et qui sont devenues de plus en plus omniprésentes. Ainsi, de nouvelles facilités et de nouveaux modes d'exploitation des services peuvent être envisagés si les capteurs et les marqueurs d'identification intègre l'internet. Dans

cette partie nous présentons les technologies basiques et nous accentuons leurs rôles dans le contexte de l'internet des objets.

4.1 L'identification par radio fréquence (RFID)

Un système RFID est composé d'un ou plusieurs lecteurs et d'un ensemble d'étiquettes (appelée aussi tags, marqueurs, identifiants ou transpondeurs) à micro-puissances. Les étiquettes sont des dispositifs minuscules équipés d'une puce contenant des informations et une antenne pour la communication radio. Elles sont placées sur les éléments que l'on veut identifier d'une manière unique ou tracer. [15] Les étiquettes peuvent avoir différentes formes (fig. :18) et peuvent être passives ou actives. Les étiquettes actives sont équipées d'une batterie, elles diffusent des signaux automatiquement et d'une façon autonome, tandis que les étiquettes passives ne disposent d'aucune source d'énergie et attendent à ce qu'un signal électromagnétique leur arrive et munit de l'énergie pour pouvoir envoyer leurs propres signaux. Les étiquettes passives sont plus déployées que celles qui sont actives car leur usage est beaucoup plus flexible avec un cout nettement réduit (comparé au cout relatif aux étiquettes actives qui est nettement élevé). Une autre spécificité pas moins importante dans les étiquettes passives qui est la durée de vie. Par le fait d'être passive, la durée de vie de l'étiquette est importante (elle reste valable tant qu'elle garde son bon état), ce qui n'est pas le cas pour une étiquette active ou la durée de vie est restreinte (s'achève avec l'épuisement de la batterie).

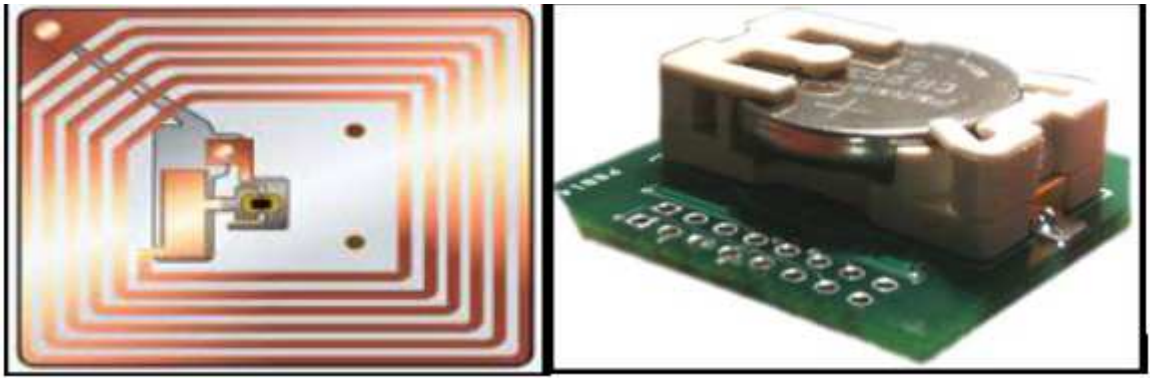


Figure 18 : Formes des étiquettes RFID

Le processus d'identification se réalise à travers un scénario bien déterminé. En effet, le lecteur active les étiquettes qui passent devant lui en leur envoyant un signal électromagnétique puissant. Les étiquettent s'active et réagissent en répondant par un signal transportant les identités. Contrairement aux systèmes d'identification par codes barre qui exigent que le lecteur et le code barre soient exactement opposés et très proches l'un de l'autre, dans un système RFID, il suffit juste que le lecteur et l'étiquette soient l'un dans la portée de communication de l'autre pour que l'interaction puisse avoir lieu. La portée de communication radio (appelée aussi la distance de lecture) dans un système RFID dépend du type de tag (passif ou actif) et de la gamme de fréquences utilisée. Par exemple, la portée avec les étiquettes actives est plus importante qu'avec celles qui sont passives.

Dans le contexte de l'Internet des objets, les objets intelligents ont besoin d'être identifiés de façon unique. À partir de là, l'adoption de la technologie RFID s'est avérée nécessaire.

4.2 Les réseaux de capteurs sans fil (RCSFs)

Les RCSFs jouent un rôle très intéressant dans l'internet des objets. En effet, les capteurs permettent la représentation des caractéristiques dynamiques (température, humidité, pression, mouvements, etc.) des objets et des endroits du monde réel dans le monde virtuel représenté par le réseau internet global. Ainsi, avec l'incorporation des réseaux de capteurs dans l'internet, les capteurs deviennent des serveurs (fournisseurs de services) dans ce que l'on désigne par le web des objets (dit WoT Web of Things) [29]. Ainsi, les services(applications) des RCSFs se rajoutent à l'ensemble des services et applications de l'internet.

5. Les applications de l'IoT

L'internet des objets ce n'est pas qu'un immense ensemble d'objets intelligents interconnectés et connectés à internet mais c'est également et plus considérablement, les applications qui sont en fait la raison d'être de cette nouvelle vague de connectivité sur Internet. L'existence des objets intelligents avec de nouvelles possibilités de communications automatiques et intelligentes vont sensiblement améliorer le mode de vie des gens ainsi que la qualité de services dans divers domaines à travers des degrés élevés d'autonomie et d'intelligence.

Les potentialités de l'internet des objets ont mené à ce que des modèles de nouvelles applications soient développées sur internet. Dans cette section, citons les applications en vedette de l'IoT.

5.1 Application militaire

L'Internet des objets est un domaine fertile tant pour les applications civiles que pour les applications militaires. Dans le domaine de défenses les capteurs et les nano-drones connectés à Internet permettent d'envisager des applications sophistiquées pour l'exploration, la surveillance des champs de batailles et des frontières, ainsi que la poursuite et la localisation géographique des objets connectés. Les forces militaires ont la tendance d'utiliser des infrastructures propriétaires pour la connectivité et les communications. En transitant vers l'Internet, il sera plutôt possible d'utiliser des infrastructures cloud, qui offrent une flexibilité opérationnelle très intéressante. Le soldat en mission peut lui-même être connecté à Internet à travers les capteurs connectés, intégrés dans sa tenue. Ces capteurs peuvent être par exemple des capteurs médicaux qui rapportent l'état de santé du soldat, ou des capteurs multimédias qui captent des images, une vidéo ou du son depuis la zone où il se trouve (le soldat) [16].

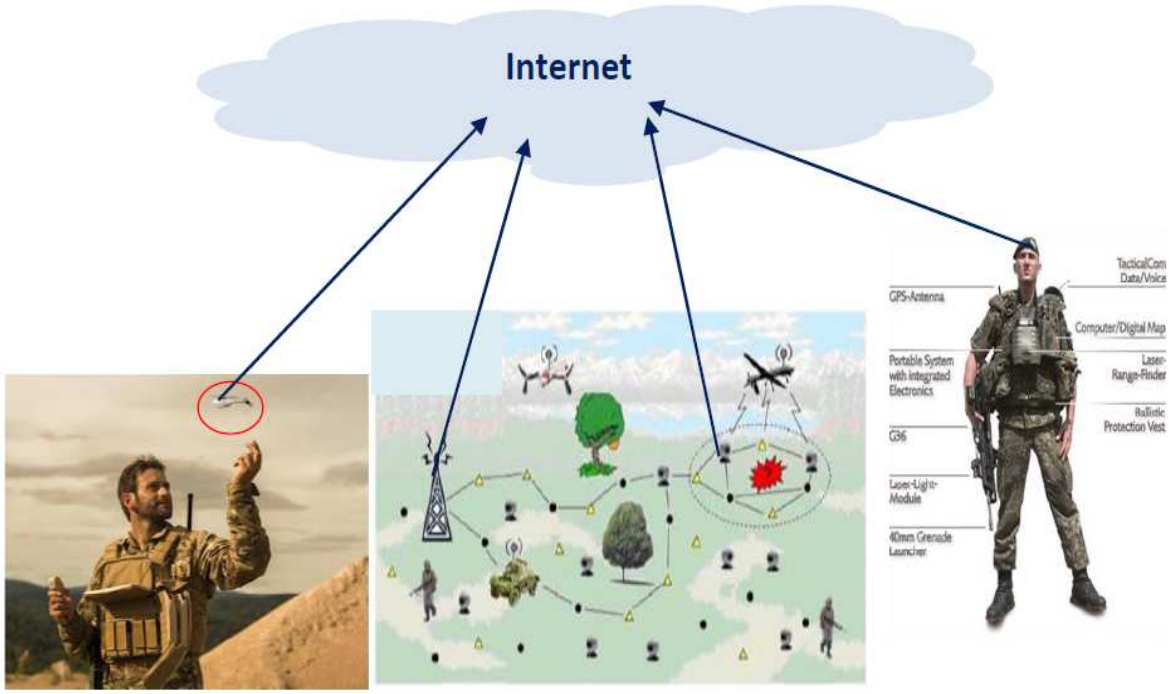


Figure 19 : Le domaine militaire et l'Internet des objets

5.2 Application médicale

L'IoT aura de nombreuses applications dans le secteur de la santé où l'objectif est d'arriver à prévenir des situations graves et suivre à distance des patients atteints des maladies chroniques et agir rapidement si cela s'est avéré nécessaire. Des capteurs corporels implantés dans le corps du patient récoltent des informations relatives aux paramètres médicaux, telles que la température, la glycémie, le rythme des battements du cœur ou encore même la tension artérielle. Ces informations seront stockées et traitées sur Internet (plus précisément sur un cloud) et mises à la disposition du médecin qui pourra les consulter n'importe quand et depuis n'importe quel dispositif connecté à Internet (ex : smartphone ou sa tablette). Le médecin est alerté en temps réel (en lui envoyant un mail ou un SMS) de tout changement brusque concernant l'état de son patient. Suivant le degré de gravité de la situation, le médecin réagit soit en se déplaçant chez le patient ou juste en le contactant et lui indiquant ce qu'il faut faire pour revenir à l'état normal. Imaginons par exemple un patient avec un rythme cardiaque irrégulier. Le capteur détectant tel événement déclenche une alerte au cardiologue s'occupant du patient. Le médecin peut également consulter à tout moment les rapports médicaux de ses patients ou bien interroger les capteurs pour avoir les valeurs actuelles.

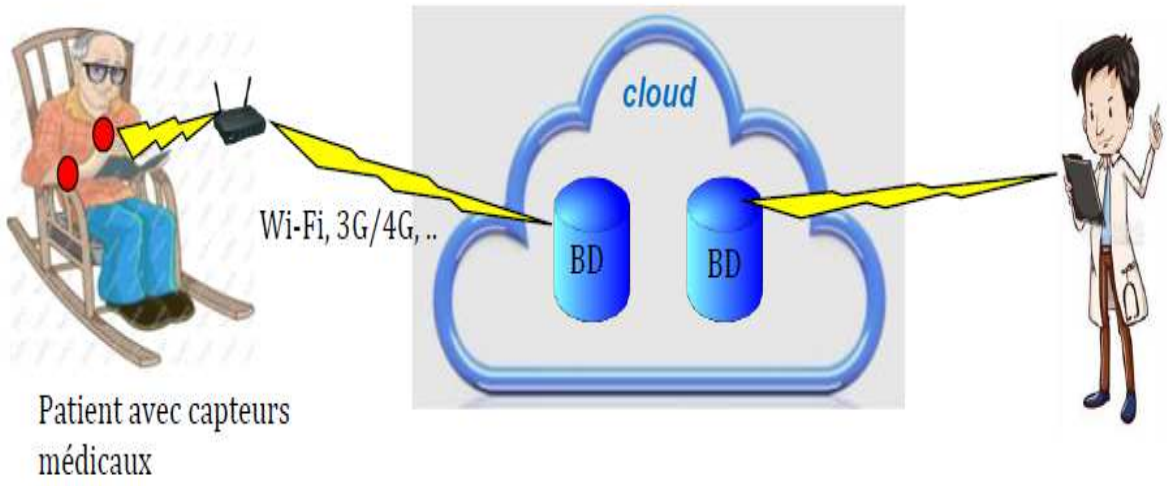


Figure 20 : L'internet des objets dans le domaine médical

5.3 Applications industrielles (smart indistry)

Une quantité considérable de capteurs, d'étiquettes RFID et les contrôleurs embarqués, s'accroit sensiblement dans les systèmes de production industrielle, sur la chaine logistique et même dans les produits. Les objets connectés aident les entreprises à améliorer la qualité de leurs processus de fabrication et à fournir un service après-vente plus concurrentiel. Ainsi, les usines connectées à Internet sont plus productives et efficaces que ceux qui ne le sont pas. D'une part, le producteur peut avoir une idée sur la commercialisation de ses produits à travers le monde, à l'aide des informations collectées auprès des différents points de ventes. D'autre part, les produits connectés seront capables de transmettre les avis(feedback) des clients aux producteurs pour faire un sondage sur le taux de satisfaction de la clientèle [17].

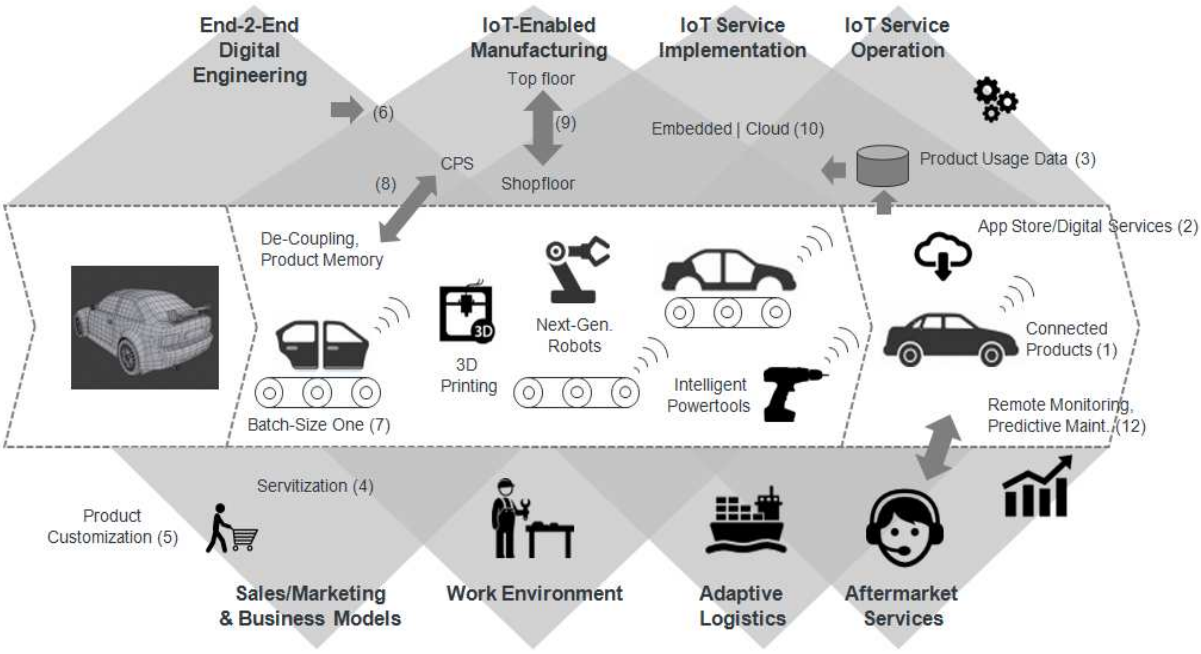


Figure 21 : IoT et l'Industrie

5.4 Les maisons intelligentes (smart home)

La maison du futur sera un objet connecté à internet accessible à distance par ses propriétaires via des Smartphones, tablettes ou ordinateurs connectés. La porte, la télévision, le thermostat, le réfrigérateur, les parapluies, les montres etc. de telle sorte qu'une porte connectée informe les parents par Internet de la rentrée de leurs enfants. La télévision qui était seulement un terminal récepteur, connectée à Internet, elle devient plutôt un dispositif émetteur/récepteur qui fournit à ses téléspectateurs la possibilité d'envoyer et recevoir des e-mails, faire des appels téléphoniques sur internet, ou autre. Un thermostat intelligent connecté au service WI-FI de la maison permet de contrôler facilement la température de celle-ci à partir de n'importe où, pour une amélioration du confort et une optimisation des économies énergétiques. Le réfrigérateur intelligent connecté à internet est muni d'un système RFID traque les produits élémentaires qui y sont stockés et enregistre des informations pertinentes leur concernant (comme la durée, du stockage et la date d'expiration). L'utilisateur peut l'interroger à distance pour savoir ce qui reste et ramener les produits manquant avant de rentrer à la maison.

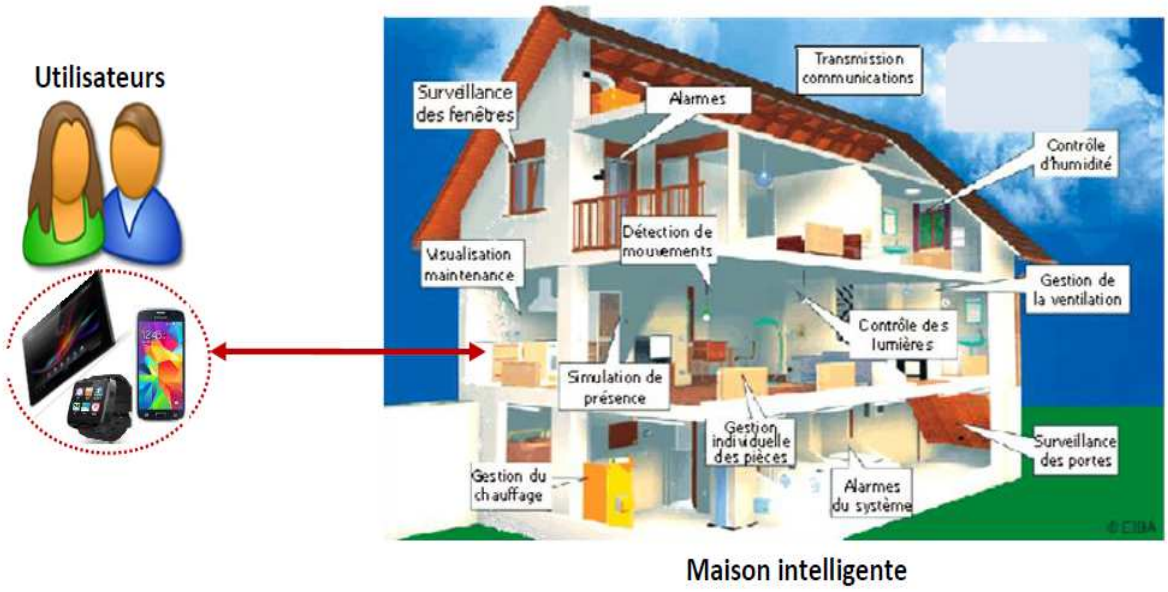


Figure 22 : L'Internet des objets et la domotique

5.5 Les villes intelligentes (smart city)

Les maisons, les routes, les bâtiments, les véhicules, les magasins, les parkings, les réseaux électriques (smart grid) etc., seront tous connectés à Internet. Le contrôle du trafic routier aide les citoyens (surtout les automobilistes) à gagner du temps en leur fournissant des informations pertinentes, en temps réel, sur son parcours (par exemple, le meilleur itinéraire selon le taux de trafic, le plus proche parking, hôtel, restaurant, hôpital etc.). D'une part, des informations d'ordre général sur la ville, comme la température, le taux d'humidité les niveaux de radiation, etc., seront disponible sur Internet pour tout le monde.

D'autre part, les autorités de la ville intelligente trouveront une facilité de réalisation des taches de contrôle de la pollution, l'éclairage urbain, etc. Néanmoins, une coexistence massive de multiples technologies est nécessaire pour la réalisation du projet de la ville intelligente, principalement les réseaux de capteurs.



Figure 23 : ville intelligente

6. Les défis de l'IoT

Bien que l'Internet des objets soit un concept qui est à la fois avantageux et promoteur. En compensation, l'IoT soulève certaines questions décisives, étroitement liées à sa maturité et son acceptabilité. On cite ci-après les enjeux les plus marquants.

6.1 La sécurité

La sécurité des personnes, des communications, des données, des services, des réseaux et des équipements était et continue à être un problème sévère observé par l'Internet habituel. Aujourd'hui avec la naissance de l'IoT, l'amplitude du problème va prendre un autre ordre de gravité. Des milliers d'objets contraints connectés en permanence à Internet et intégrés dans toute sorte d'objets dans notre vie quotidienne, vont porter le risque d'être ciblés par les menaces classique de l'Internet.

Il est même possible que de nouvelles générations d'attaques apparaissent. Donc, les objets intelligents dans l'IoT, la transmission et le stockage de leurs données sur Internet devraient être sécurisés [18].

6.2 La protection de la vie privée des utilisateurs

Un grand nombre de capteurs connectés à Internet et implantés dans des objets d'usage quotidien révèlent nos habitudes, notre état de santé, notre localisation géographique et autres types d'informations qui nous sont privés. Il devra strictement y avoir des mécanismes robustes qui peuvent assurer la confidentialité des données que l'utilisateur qualifie être sensibles.

6.3 Limitations de ressources

Les capteurs et les tags RFID sont très limités en ressources de calculs, de stockage mémoire et d'énergie. A cet effet, les solutions (protocoles de communications ou de sécurité, technologies de transmission, etc.) destinées à l'Internet des objets doivent prendre en considération tous ces contraintes et limitations.

6.4 La virtualisation

Plusieurs capteurs connectés peuvent représenter un seul capteur virtuel qui rapporte une mesure virtuelle résultant de l'agrégation de plusieurs états secondaires.

Par exemple un capteur virtuel qui nous dit si l'état de santé du patient est bon ou non. Cette information est une combinaison de plusieurs informations fournies par plusieurs capteurs médicaux [18].

Ainsi, un model générique de virtualisation des objets connectés à l'IoT, nommé VoT (Virtualization of Things) permet une représentation abstraite des objets et l'accumulation des données qui en proviennent, depuis différents endroits, pour faciliter leur contrôle.

6.5 Le nombre croissant des objets connectés

Il est prévu que le nombre d'objets intelligents qui vont peupler l'Internet du futur franchira les milliards. Avec cela, l'adoption de nouveaux mécanismes qui supportent efficacement l'évolutivité continue dans le nombre d'objets connectés, est vivement recommandée.

7. Les avantages de l'internet des objets

Voici en résumé des principaux avantages de l'IoT

- Amélioration de la qualité de service et de la télésurveillance dans différents domaines d'applications, à savoir le domaine industriel, médical, militaire etc. ;
- Possibilité d'exploitation des ressources géantes de l'Internet pour le stockage et traitement des données écoulées de l'IoT ;
- Accès omniprésent à l'information pour un monde plus intelligent et un mode de vie plus sophistiqué et confortable ;
- Améliorer la productivité et l'expérience-client : les objets connectés envoient des rapports à leurs constructeurs indiquant les préférences et les habitudes des clients aidant davantage les entreprises à adaptée la production selon la demande et les exigences de la clientèle ;
- Le gain du temps est un avantage de l'IoT, les déplacements inutiles sont dès lors remplacés par une simple navigation sur le web pour commander des produits, contrôler l'état des objets et/ou endroits connectés ;
- Dans certaines applications, l'IoT nous permet même de rationaliser os dépenses et faire des économies car on ne consomme qu'en cas de besoin, que ça soit pour les achats ou la consommation énergétique (nécessaire pour l'éclairage ou la climatisation) ou autre [19].

8. Conclusion

L'IoT comme étant une évolution de l'Internet actuel permet une amélioration considérable de notre mode de vie. Cette amélioration nous est donnée grâce à des objets intelligent qui interagissent entre eux et avec les utilisateurs de telles sortes que nos activités, notre état de santé, nos dépenses, etc., puissent être contrôler efficacement.

III. DEUXIEME PARTIE : MÉCANISMES DE SECURITE DANS LE DNS-SD

Chapitre 3 : Étude de la sécurité dans DNS-SD

1. Introduction

Même si le DNS est largement utilisé sur Internet, le système n'a pas été conçu à l'origine dans un souci de sécurité. La raison en est que l'Internet n'était pas accessible au public au moment du lancement du DNS. Après que l'Internet soit devenu accessible au public, des vulnérabilités ont été découvertes et exploitées. Parmi les exemples de vulnérabilités du DNS, citons les attaques par déni de service distribué (DDoS), dirigées à la fois vers les serveurs DNS et d'autres cibles spécifiques, et les attaques de type "man-in-the-middle". Les attaques DDoS ont augmenté en volume et en fréquence, il est donc important de garder à l'esprit que ces attaques existent et de s'y préparer. Même si la durée de ces attaques est plus courte qu'auparavant (l'hôte ciblé peut se rétablir en moins d'une heure), elles ont toujours un impact sur l'activité des entreprises ciblées.

2. Mécanisme d'attaques dans DNS

2.1 Attaque par inondation (DDoS)

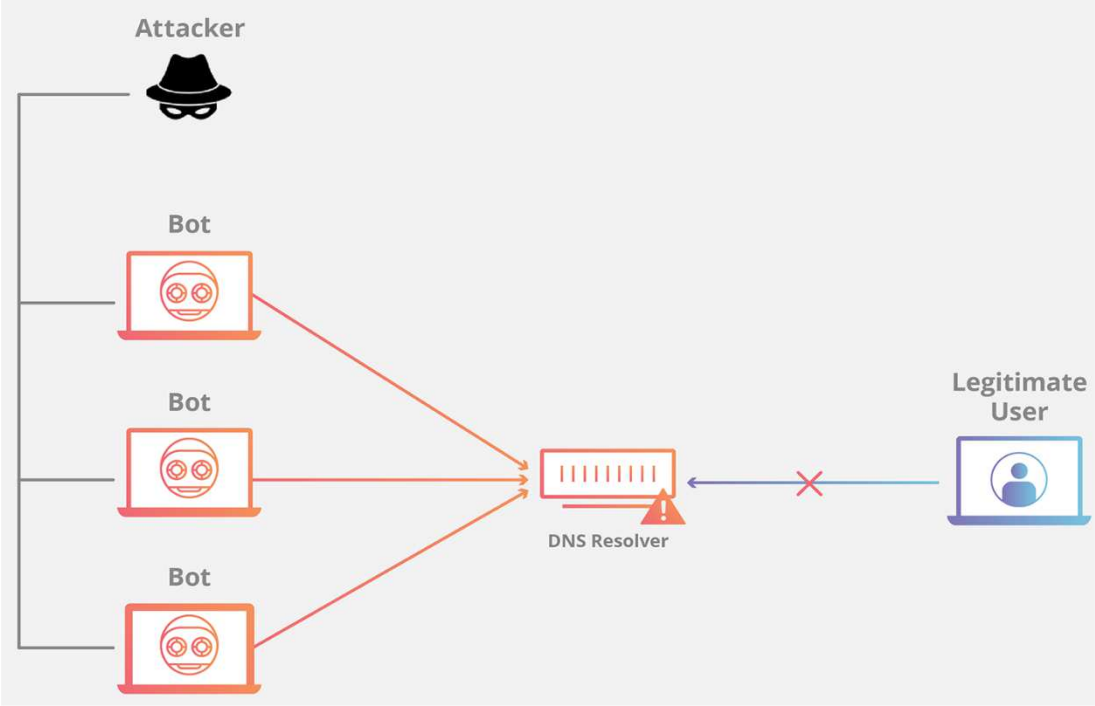


Figure 24 : Attaque DDoS

L'un des types d'attaque DDoS est l'attaque qui cible les serveurs racine. Ces attaques sont exécutées en inondant les serveurs racines de messages, généralement des messages Ping ICMP (Internet Control Message Protocol).

Un autre type d'attaque DDoS, plus susceptible d'être visible pour l'internaute moyen, est une attaque contre un hôte spécifique, tel qu'une société de jeux ou les serveurs de messagerie d'une université. Même si ces attaques sont dirigées vers une cible spécifique, elles peuvent avoir un impact sur d'autres hôtes plus essentiels, comme le serveur DNS d'un fournisseur de services Internet (FSI). [20]

2.2 Attaque MiTM (Man in The Middle)

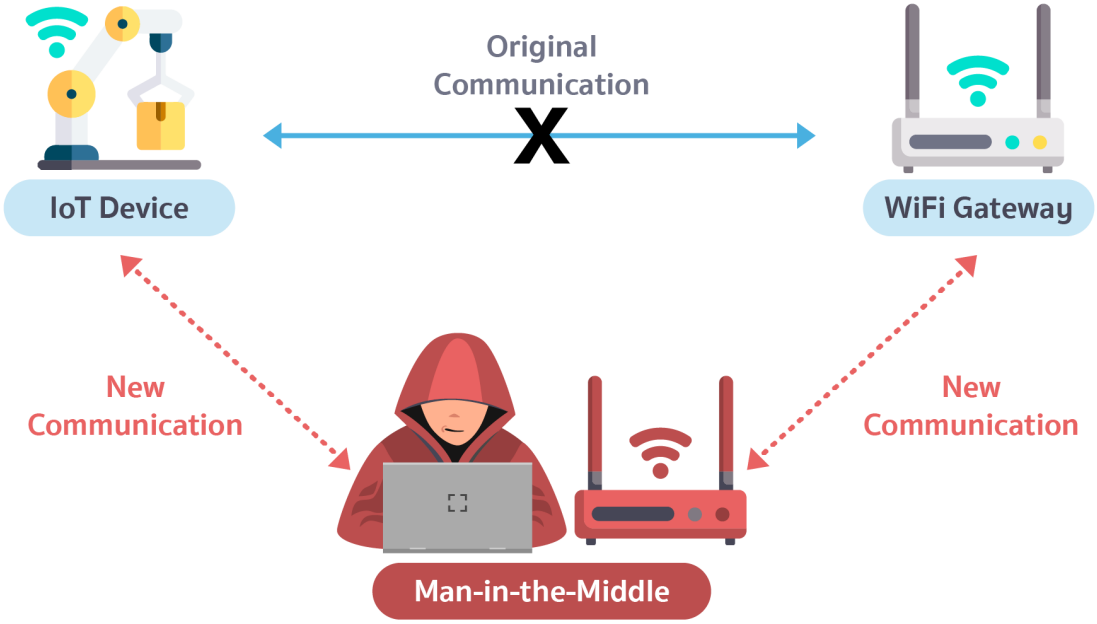


Figure 25 : Attaque MiTM

Source : https://www.bing.com/images/search?view=detailV2&ccid

Le DNS est également vulnérable aux attaques de type "man-in-the-middle". Ce type d'attaque est également connu sous le nom de DNS spoofing ou DNS cache poisoning. Lorsqu'un utilisateur envoie une requête DNS, il s'attend à ce que les informations qu'il reçoit des serveurs DNS soient correctes. Ce n'est pas toujours le cas. Lors d'une attaque de type man-in-the-middle, un attaquant peut se faire passer pour un serveur DNS et renvoyer une adresse IP erronée [21]. L'utilisateur se rendra alors sur le mauvais site web. Ce faux site peut ressembler au site original que l'utilisateur essayait d'atteindre. Ainsi, si l'utilisateur croit que le faux site Web est le vrai, il peut saisir son nom d'utilisateur et son mot de passe sur un faux site Web, donnant ainsi ces informations à l'attaquant. Ce problème peut être résolu grâce aux extensions de sécurité DNS (DNSSEC) que nous allons étudier par la suite.

2.3 Attaque par écoute illicite

Les écoutes clandestines sont le résultat d'une interception du trafic réseau. Elles permettent à un attaquant d'obtenir des mots de passe, des numéros de carte bancaire et d'autres informations confidentielles qu'un utilisateur envoie sur le réseau. Elles peuvent être passives ou actives :

- **Écoute clandestine passive** – Un pirate détecte des informations en écoutant la transmission de messages sur le réseau.
- **Écoute clandestine active** – Un pirate s'empare activement d'informations en se faisant passer pour une unité amie et en envoyant des requêtes aux transmetteurs. On appelle cela sonder, scanner ou saboter [22] .

Il est souvent plus important de détecter des écoutes passives que des écoutes actives, car ces dernières exigent de l'attaquant qu'il apprenne à connaître les unités amies en effectuant préalablement des écoutes passives.

Le chiffrement des données est la meilleure contre-mesure contre les écoutes clandestines.

2.4 Attaque sur mDNS

Le protocole mDNS est destiné à résoudre les noms d'hôte en adresses IP dans de petits réseaux qui ne possèdent pas de serveur DNS local. Le service mDNS peut être contacté à l'aide de requêtes UDP sur le port 5353. Si votre service mDNS est exposé à Internet, interroger le service peut permettre à des personnes malveillantes de collecter des informations sur votre serveur (telles que l'adresse MAC du périphérique ou les services s'exécutant sur la machine) qui pourraient être utilisées pour préparer une attaque.

Aussi, puisque mDNS est basé sur UDP, les requêtes mDNS peuvent être exploitées pour effectuer des attaques par amplification (l'attaquant peut falsifier l'adresse IP de sa cible pour la saturer avec les réponses mDNS de votre serveur).

Multicast DNS est conçu pour être accédé depuis un réseau local. Cela signifie que ce n'est généralement pas une bonne idée d'exposer ce service directement sur Internet ou, de façon générale, sur un environnement où des clients non-approuvés peuvent accéder directement au service [23].

Différentes options sont disponibles pour résoudre ce problème et protéger votre serveur :

- Désactivez le service mDNS (Apple Bonjour ou Avahi-daemon) si vous ne l'utilisez pas. Il s'agit de la solution la plus facile et la plus efficace.
- Configurez votre pare-feu pour filtrer les connexions entrantes vers le port UDP/5353 de votre serveur, et n'autorisez que les IP/réseaux de confiance qui doivent contacter votre service mDNS pour y accéder.

2.5 Résistance à l'usurpation d'identité de l'expéditeur

L'usurpation d'identité de l'expéditeur est une attaque dans laquelle des messages, tels que des offres de service, sont falsifiés par des entités qui ne possèdent pas le matériel de clé secrète correspondant. Ces attaques peuvent être utilisées pour connaître l'identité d'un interlocuteur, activement ou passivement.

2.6 Résistance aux attaques de dictionnaires

Dans tout protocole où le récepteur de messages doit effectuer des opérations cryptographiques sur ces messages, il existe un risque d'attaque par inondation par force brute provoquant une

dépense excessive de temps CPU par le récepteur et, le cas échéant, la puissance de la batterie pour le traitement et l'élimination de ceux-ci. En outre, les attaques d'amplification doivent être prises en considération. Les messages avec des charges utiles plus importantes ne doivent être envoyés qu'en réponse à une requête envoyée par un client vérifié.

2.7 Attaque sur la base de découverte (DNS)

- **Empoisonnement du cache DNS**
  L'empoisonnement du cache DNS (DNS Cache Poisoning) ou pollution de cache DNS est une technique permettant de trouver les serveurs DNS afin de leur faire croire qu'ils reçoivent une réponse valide à une requête qu'ils effectuent, alors qu'elle est frauduleuse. Pour mener à bien une attaque par DNS Cache Poisoning, l'attaquant exploite une vulnérabilité du serveur DNS qui accepte alors des informations incorrectes. Si le serveur ne valide pas les informations reçues et qu'il ne vérifie pas qu'elles proviennent d'une source fiable, alors il stockera dans son cache ces informations erronées. Il les transmettra par la suite aux utilisateurs qui effectuent la requête visée par l'attaque.

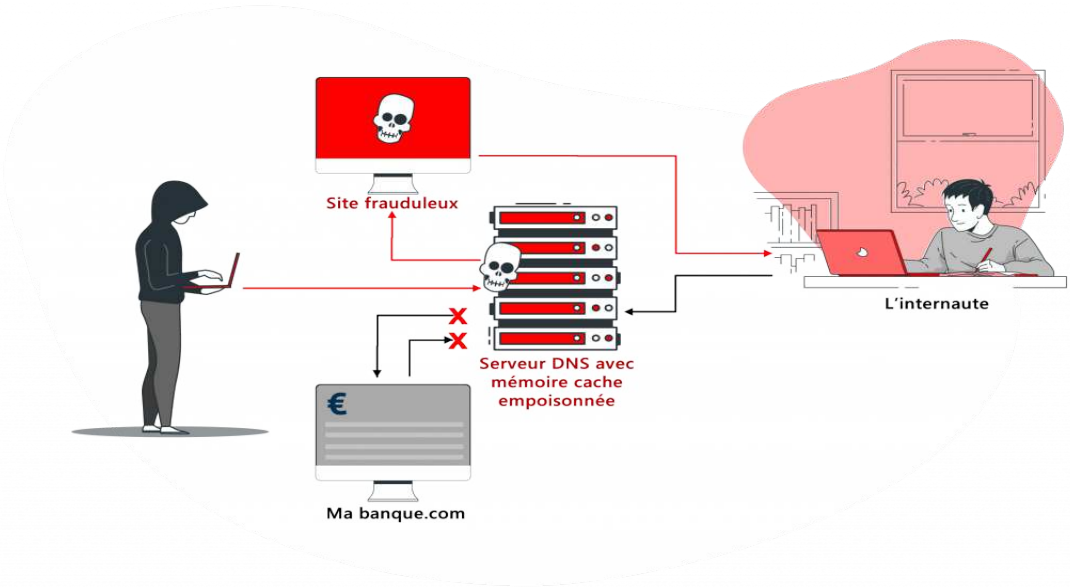


Figure 26 : Attaque par DNS Cache Poisoning

Ce type d'attaque par DNS Cache Poisoning permet, par exemple, d'envoyer un utilisateur vers un faux site dont le contenu peut servir à du phishing ou comme vecteur de virus et autres applications malveillantes. Un ordinateur présent sur Internet utilise normalement un serveur DNS géré par le fournisseur d'accès. Ce serveur DNS est la plupart du temps limité aux seuls utilisateurs du réseau du fournisseur d'accès et son cache contient une partie des informations rapatriées par le passé. Une attaque par empoisonnement sur un seul serveur DNS du fournisseur d'accès peut affecter l'ensemble de ses utilisateurs, soit directement ou indirectement si des serveurs esclaves s'occupent de propager l'information [24].

- Attaque sur le protocole DNS

DNS est un protocole « plein texte » que tout le monde peut consulter facilement et les malwares peuvent l'utiliser. Les entreprises doivent donc filtrer ce trafic et bloquer d'éventuelles exfiltrations de données. DNSSEC est un moyen de s'assurer que l'enregistrement que vous demandez au serveur DNS est authentique. La donnée est chiffrée et si un attaquant cherche à altérer la réponse du serveur, il y aura une erreur qui permettra de détecter la tentative d'attaque. DNSSEC représente un peu ce que HTTP/S a été pour HTTP, mais est beaucoup plus complexe à déployer. DNS est une architecture beaucoup plus distribuée, avec des registrar, des serveurs DNS chez les fournisseurs d'accès internet, dans les entreprises. Migrer vers DNSSEC sera une transition majeure. [25]

- Vol de domaines
  Chaque registrar propose la possibilité de transférer un domaine d'un autre registrar vers lui-même afin de le gérer. Le transfert est une opération délicate qui nécessite beaucoup de précautions. C'est au registrar demandant le transfert de les prendre en compte. A savoir: vérifier que le transfert est réellement demandé par les contacts du domaine.
  Le registrar qui doit perdre la gestion du domaine ne peut pas vérifier si le transfert a été demandé suivant cette règle et donc une fois que la demande de transfert est lancée, le transfert peut éventuellement aboutir (si par exemple vous êtes en vacances et que vous ne pouvez pas refuser le transfert)
  Les attaquants modifient l'enregistrement d'un nom de domaine et en deviennent propriétaires. Ils y parviennent généralement en accédant sans autorisation au compte du bureau d'enregistrement du domaine, par le vol de justificatifs d'identité ou par l'ingénierie sociale. [26]

3. La sécurité des données dans la découverte de service

3.1 Authenticité, intégrité et fraîcheur

Les appareils (serveurs et clients) peuvent-ils faire confiance aux informations qu'ils reçoivent ? A-t-il été modifié en vol par un adversaire ? Les appareils peuvent-ils faire confiance à la source de l'information ? La source d'information est-elle fraîche, c'est-à-dire non rejouée ? La fraîcheur peut être requise ou non selon que le processus de découverte est censé être en ligne. Dans certains cas, la publication d'informations de découverte dans un répertoire ou un registre partagé, plutôt que dans chaque destinataire en ligne via un canal de diffusion, peut suffire

3.2 Confidentialité

La confidentialité consiste à restreindre l'accès aux informations aux seules personnes autorisées. Idéalement, il ne devrait s'agir que des parties de confiance appropriées, bien qu'il puisse être difficile de définir qui sont "les parties de confiance appropriées". Dans certains cas d'utilisation, cela peut signifier que seuls les clients et serveurs mutuellement authentifiés et de confiance peuvent lire les messages envoyés les uns pour les autres. Le processus de découverte de service en particulier est souvent utilisé pour découvrir de nouvelles entités que l'appareil ne connaissait pas auparavant. Il peut être difficile de comprendre comment un appareil peut avoir une relation de confiance établie avec une nouvelle entité avec laquelle il n'a jamais communiqué auparavant [27].

3.3 Mécanismes de contrôle d'accès

Un mécanisme de contrôle d'accès est une procédure visant à déterminer quels clients ou applications auront accès au serveur. Seuls les clients disposant des droits appropriés pourront se connecter au serveur; tous les autres se verront refuser l'accès. Il existe deux types de mécanisme de contrôle d'accès, l'un étant basé sur l'utilisateur et l'autre sur la machine [28].

- Accès basé sur l'utilisateur : Le mécanisme de contrôle d'accès basé sur l'utilisateur, ou sur les droits, permet d'attribuer des droits explicites à un utilisateur donné sur une machine donnée. Le client de cet utilisateur transmet au serveur les données d'attribution de droits. Si ces données sont conformes à celles du serveur, l'utilisateur est autorisé à se connecter.
- Accès basé sur la machine : Le mécanisme de contrôle d'accès basé sur la machine est un mécanisme général qui donne accès à une machine spécifique, depuis laquelle tous les utilisateurs peuvent se connecter au serveur. C'est une forme de contrôle d'accès plus souple : si la machine a accès au serveur, tous ses utilisateurs sont également autorisés à s'y connecter.

3.4 Refus de l'expéditeur (déni de responsabilité)

Le refus de l'activité de l'expéditeur, par exemple la diffusion d'une demande de découverte, peut être souhaitable ou nécessaire dans certains cas d'utilisation. Cette propriété garantit que les espions ne peuvent pas prouver que les expéditeurs ont émis un message spécifique destiné à un ou plusieurs pairs.

3.5 Extension de sécurité de nom de domaine (DNSSEC)

Le DNSSEC crée un système de noms de domaine sécurisé en ajoutant des signatures cryptographiques aux enregistrements DNS existants. Ces signatures numériques sont stockées dans des serveurs de noms DNS parallèlement à des types d'enregistrements courants comme A, AAAA, MX, CNAME, etc. En vérifiant sa signature associée, vous pouvez vérifier qu'un enregistrement DNS demandé provident de son serveur de noms faisant autorité et qu'il n'a pas été modifié en chemin, contrairement à un enregistrement injecté dans une attaque de l'homme du milieu.

Pour faciliter la validation des signatures, DNSSEC ajoute quelques nouveaux types d'enregistrement DNS :

- RRSIG : contient une signature cryptographique
- DNSKEY : contient une clé de signature publique
- DS : contient le hachage d'un enregistrement DNSKEY
- NSEC et NSEC3 : pour le déni d'existence explicite d'un enregistrement DNS
- CDNSKEY CDS : pour une zone fille demandant des mises à jour d'un ou de plusieurs enregistrements DS dans la zone parent.

Dans cet article, nous allons aborder l'interaction entre les enregistrements RRSIG, DNSKEY et DS, ainsi que la façon dont ils ajoutent une couche de confiance en haut du DNS.

RRsets (Ensemble d'enregistrement de ressources) **:** La première étape pour sécuriser une zone avec DNSSEC consiste à regrouper tous les enregistrements de même type dans un ensemble d'enregistrement de ressources (RRset).

Par exemple si trois enregistrements AAAA dans une zone ont la même étiquette (c'est-à-dire label.exemple.com), ils seront tous regroupés en un seul RRset AAAA.

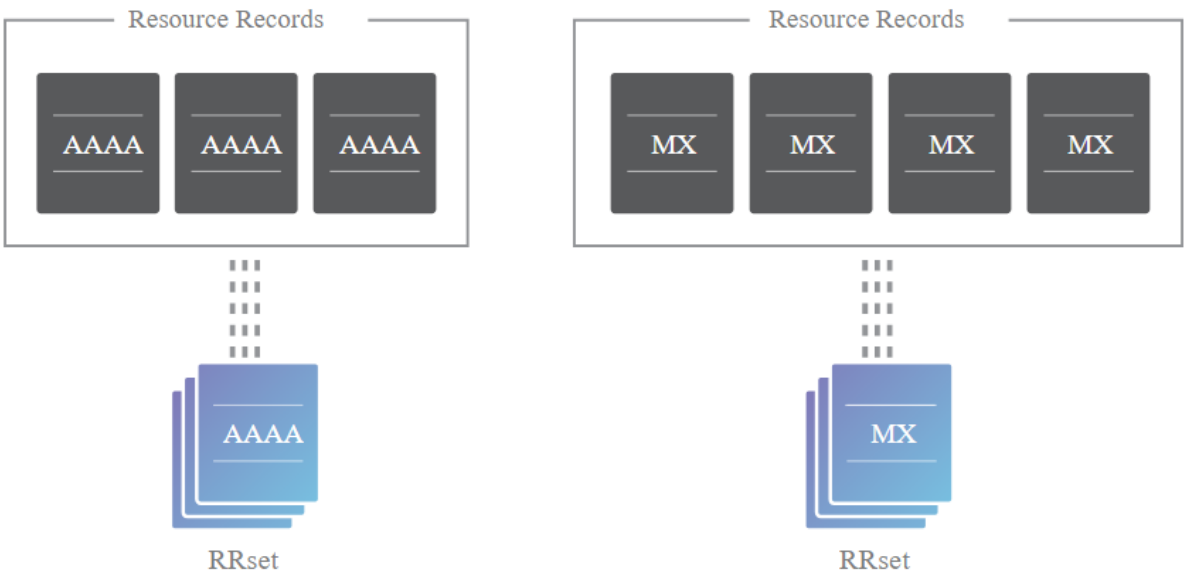


Figure 27 : RRset Ensemble d'enregistrements de ressources

C'est en fait ce RRset complet qui est signé numériquement, contrairement aux enregistrements DNS individuels. Bien sûr, cela signifie également que vous devez demander et valider tous les enregistrement AAAA d'une zone portant la même étiquette au lieu de ne valiser qu'un seul d'entre eux [29].

- **Clés de signature de zone (RRSIG)**

Chaque zone dans le DNSSEC possède une paire de clés de signature de zone (ZSKZ) : la partie privée de la clé signe numériquement chaque RRset de la zone, tandis que la partie publique vérifie la signature. Pour activer DNSSEC, un opérateur de zone crée des signatures numériques pour chaque RRset à l'aide de la ZSK privée et les stocke dans son serveur de noms en tant qu'enregistrements RRSIG.

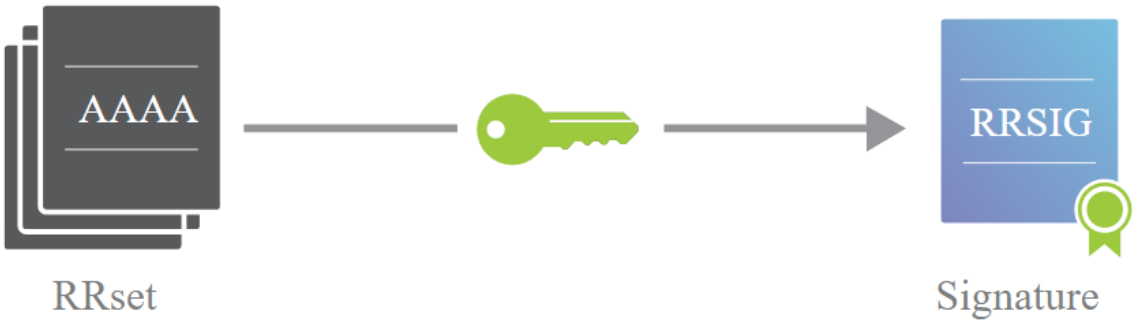


Figure 28 : RRSIG

Cependant, ces enregistrements RRSIG sont inutiles à moins que les résolveurs DNS puissent vérifier les signatures. L'opérateur de la zone doit également rendre sa ZSK publique disponible en l'ajoutant à son serveur de noms dans un enregistrement DNSKEY.

Lorsqu'un résolveur DNSSEC demande un type d'enregistrement particulier (par exemple AAAA), le serveur de noms renvoie également le RRSIG correspondant. Le résolveur peut alors extraire du serveur de noms l'enregistrement DNSKEY contenant la ZSK publique. Ensemble, le RRset, le RRSIG et la ZSK publique peuvent valider la réponse.

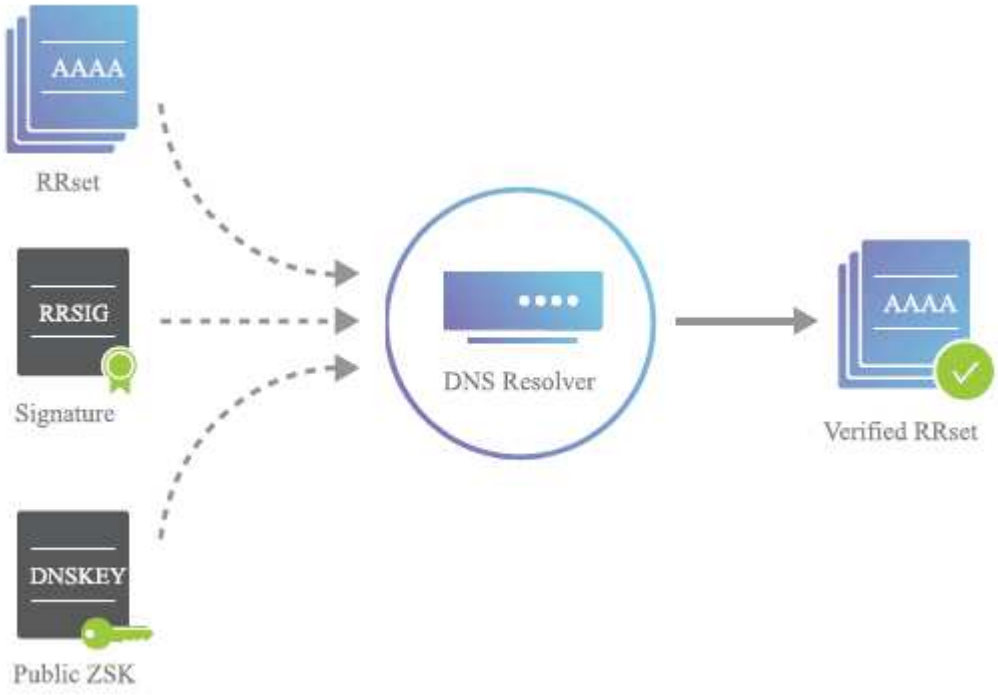


Figure 29 : ZSK clé de signature de zone

Si nous faisons confiance à la clé de signature de zone dans l'enregistrement DNSKEY, nous pouvons faire confiance à tous les enregistrements de la zone. Mais qu'en est-il si la clé de signature de zone est compromise ? Nous devons trouver un moyen de valider la ZSK publique [29].

- **Clés de signature de clé (KSK)**

En plus d'une clé de signature de zone, les serveurs de noms DNSSEC disposent également d'une clé de signature de clés (KSK). La KSK valide l'enregistrement DNSKEY exactement de la même manière que notre ZSK a sécurisé le reste de nos RRsets dans la section précédente : elle signe la ZSK publique (qui est stocké dans un enregistrement DNSKEY), créant un RRSIG pour le DNSKEY [29].

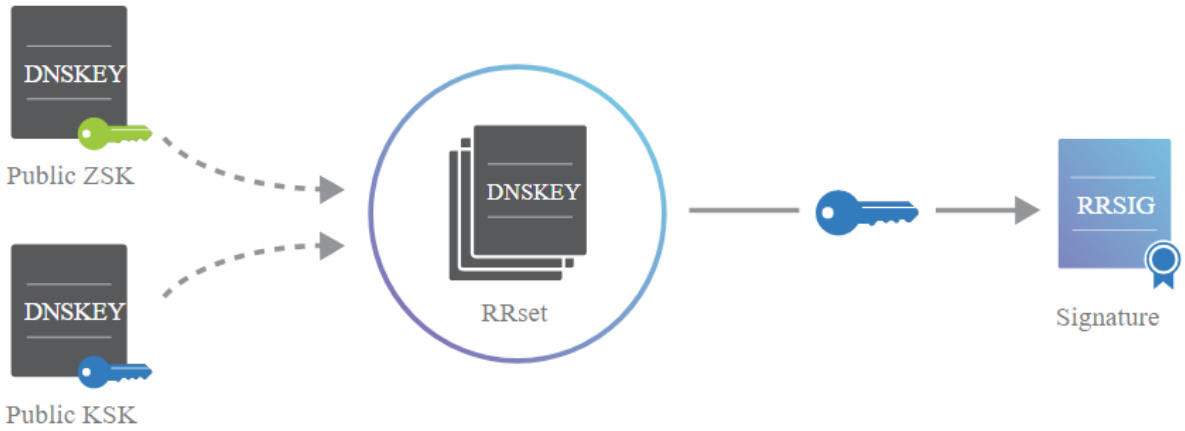


Figure 30 : KSK clé de signature de clé

Tout comme la ZSK publique, le serveur de noms publie la KSK publique dans un autre enregistrement DNSKEY, ce qui nous donne le RRset DNSKEY indiqué ci-dessus. La KSK publique et la ZSK publique sont toutes deux signés par la KSK privée. Les résolveurs peuvent ensuite utiliser la KSK publique pour valider la ZSK publique.

Pour les résolveurs, la validation ressemble maintenant à ce qui suit :

- ✓ Demande du RRset souhaité, qui renvoie également l'enregistrement RRSIG correspondant.
- ✓ Demande des enregistrements DNSKEY contenant la ZSK publique et la KSK publique, qui renvoie également le RRSIG pour le RRset DNSKEY.
- ✓ Vérification du RRSIG du RRset demandé à l'aide de la ZSK publique.
- ✓ Vérification du RRSIG du RRset DNSKEY à l'aide de la KSK publique.

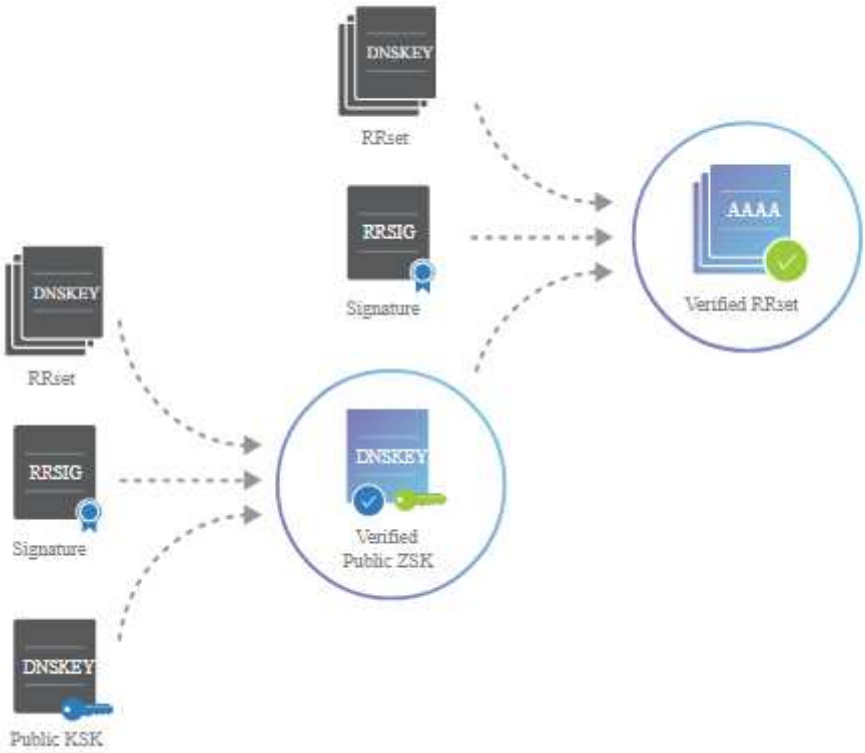


Figure 31: validation de la signature

Bien entendu, le RRset DNSKEY et les enregistrement RRSIG correspondants peuvent être mis en cache, ainsi les serveurs de noms DNS ne sont pas constamment bombardés de requêtes inutiles.

Pourquoi utilisons-nous des clés de signature de zone et de clé de signature de clés distinctes ? Comme nous le verrons dans la section suivante, il est difficile de remplacer une KSK obsolète ou compromise. Changer la ZSK, en revanche, est beaucoup plus facile. Nous pouvons ainsi utiliser une ZSK plus petite sans compromettre la sécurité du serveur, en réduisant la quantité de données que le serveur doit envoyer avec chaque réponse [30].

Nous avons maintenant établi la confiance au sein de notre zone, mais le DNS est un système hiérarchique et les zones fonctionnent rarement de manière indépendante. Pour compliquer encore les choses, la clé de signature de clé est signée par elle-même, ce qui n'apporte rien de plus à la confiance. Nous avons besoin d'un moyen de propager la confiance dans notre zone à sa zone parent [30].

- **Enregistrements signataires de délégation (DS)**

DNSSEC introduit un enregistrement Délégation Signer (DS) pour permettre la propagation de la confiance dans une zone parent à une enfant. Un opérateur de zone hache l'enregistrement DNSKEY contenant la KSK publique et le donne à la zone parent pour qu'elle le publie en tant qu'enregistrement DS.

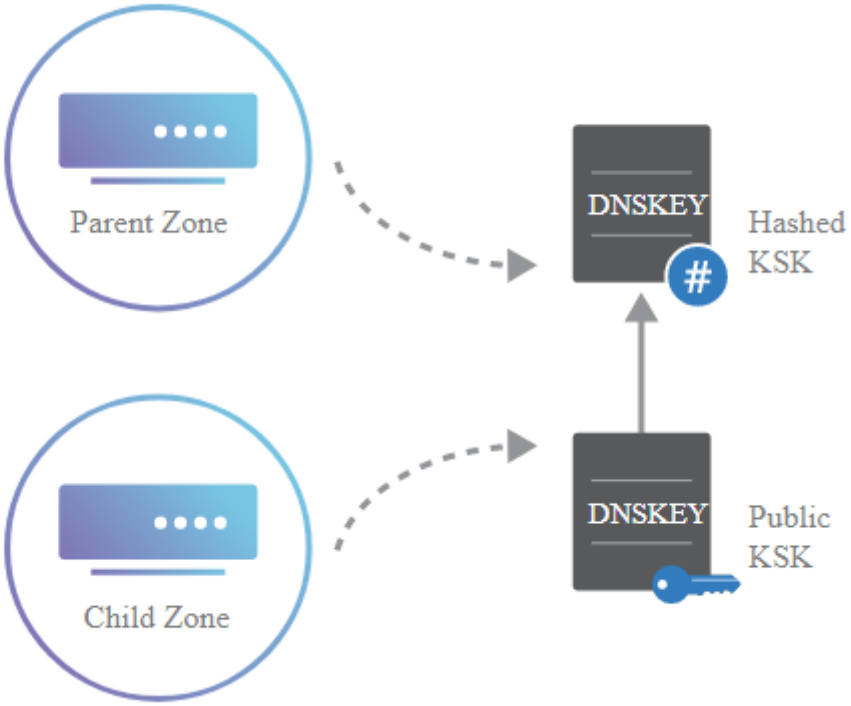


Figure 32 : Signature de Délégation

Chaque fois qu'un résolveur est dirigé vers une zone enfant, la zone parent fourni également u enregistrement DS. C'est grâce à cet enregistrement DS que les résolveurs savent que la zone enfant est compatible DNSSC. Pour vérifier la validité de la KSK publique de la zone enfant, le résolveur la hache et la compare à l'enregistrement DS du parent. En cas de correspondance, le résolveur peut supposer que la KSK publique n'a pas été falsifiée ce qui signifie qu'il peut faire confiance à tous les enregistrements de la zone enfant. C'est ainsi qu'une chaine de confiance est établie dans DNSSEC.

Notez que toute modification de la KSK nécessite de modifier l'enregistrement DS de la zone parent. La modification de l'enregistrement DS est un processus à plusieurs étapes qui peut finir par casser la zone si elle n'est pas effectuée correctement. Le parent doit commencer par ajouter le nouvel enregistrement DS, puis attendre que le TTL de l'enregistrement DS d'origine expire avant de le supprimer. C'est pourquoi il est beaucoup plus facile d'échanger des clés de signature de zone que des clés de signature de clé [31].

- **La chaine de confiance**

Nous avons maintenant un moyen d'établir la confiance au sein d'une zone et de la propager à sa zone parent, mais comment faisons-nous confiance à l'enregistrement DS ?

L'enregistrement DS est signé comme tout autre RRset, ce qui signifie qu'il a un RRSIG correspondant dans le parent. Tout le processus de validation se répète jusqu'à la KSK publique du parent. Pour procéder à la vérification, nous devons aller à l'enregistrement DS de ce parent, et remonter toute la chaine de confiance.

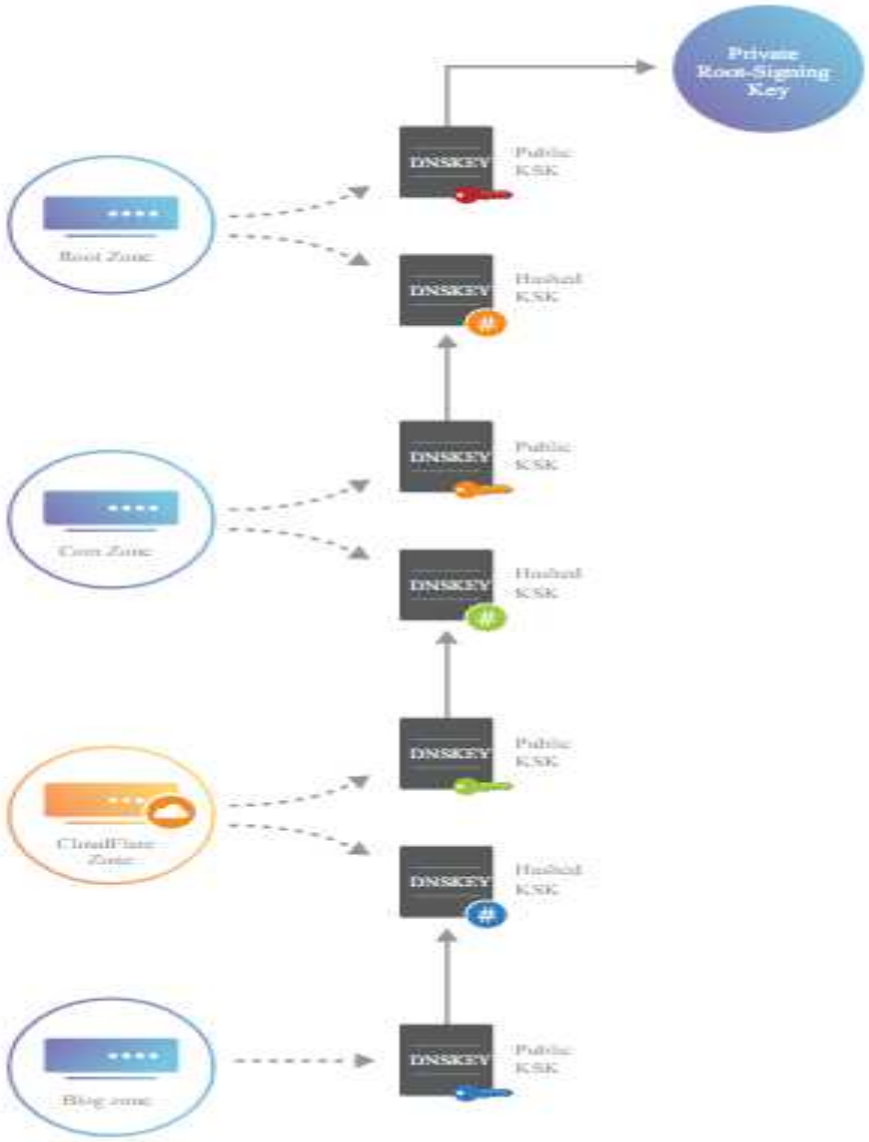


Figure 33 : La chaine de confiance

Cependant lorsque nous arrivons enfin à la zone DNS racine un problème se pose : il n'existe aucun enregistrement DS parent pour le valiser. C'est là qu'on découvre un aspect très internet mondial.

Lors de la Root Signing Ceremony, plusieurs personnes sélectionnées dans le monde entier se réunissent pour signer le RRset DNSKEY racine de manière très publique et hautement contrôlée. La cérémonie produit un enregistrement RRSIG qui peut être utilisé pour vérifier la KSK publique et la ZSK publique du serveur de noms racine. Au lieu de faire confiance à la KSK publique en raison de l'enregistrement DS du parent, nous supposons qu'il est valide par ce que nous faisons confiance aux procédures de sécurité entourant l'accès à la KSK privée.

La capacité à établir la confiance entre les zones parent et enfant fait partie intégrante de DNSSEC. Si une partie de la chaîne est rompue, nous ne pouvons pas faire confiance aux enregistrements que nous demandons, car un homme du milieu pourrait modifier les enregistrements et nous diriger vers l'adresse IP de son choix [32].

4. Conclusion

La sécurité permet de garantir la confidentialité, l'intégrité, l'authenticité des informations, cet aspect est fondamental quel que soit le domaine. Les pirates avec leurs nombreux techniques allant du vol d'informations, suppression des serveurs de noms faisant autorité pour refuser l'accès à un domaine, rediriger le trafic vers des destinations malveillantes, jusqu'à prendre le contrôle de l'infrastructure.

Toutefois, il existe une contre-mesure qui est le DNSSEC, ce dernier assure entre autres l'authenticité, l'intégrité, et la non falsification du DNS, et permet de procéder à une technique de sécurité sûre.

IV. TROISIEME PARTIE : MISE EN PLACE DU SOLUTION DNS-SD

Chapitre 4 : Mise en œuvre de DNS-SD

1. Introduction

Les principales implémentations de zeroconf se trouvent dans Bonjour d'Apple, Avahi, CUPS, SSDP et UPnP. Les protocoles zeroconf n'étant pas sécurisés par rapport aux protocoles de réseau configurés de manière standard, le développement de mécanismes de sécurité légers reste un domaine d'alimentation de la technologie Zeroconf.

Dans ce chapitre nous présentons un aperçu de la technologie Zeroconf, de ses architectures et de ses applications. Nous mettons en œuvre une implémentation Avahi pour la découverte de services (DNS-SD) fonctionnant sur le système d'exploitation Linux. Nous discutons des problèmes pratiques de déploiements de notre système et la manière dont il interagit avec l'environnement

2. Mise en place du système de découverte

L'installation notre système de découverte nécessite une configuration matérielle, des outils logiciels et des prérequis bien spécifiques. Nous avons choisi un Raspberry Pi comme notre serveur de noms et Gateway, avec les paramètres suivants :

- ✓ Raspberry Pi 3 model B+
- ✓ Système d'exploitation Raspbian
- ✓ Processeur quad-cor 64 bits 1.4 GHz
- ✓ 1 Go de mémoire SDRAM



Figure 34 : Raspberry Pi

Après avoir installé Raspbian sur notre matériel, on va installer les différentes dépendances pour le bon fonctionnement de notre serveur comme suite :

2.1 Installation de bind9

Pour offrir des services réseaux à l'utilisateur, DNS-SD utilise intensivement le protocole DNS. Nous verrons ici, comment configurer notre serveur DNS et ses extensions recommandées. Ensuite, nous expliquerons en quoi le protocole DHCP est un bon allié de DNS-SD.

- Installation des paquets

Ici nous allons télécharger les paquets nécessaires pour la configuration du serveur (bind9) de noms

```
pi@lyam0:~ $ sudo apt install bind9 bind9-utils bind9-doc bind9utils bind9-dnsutils
 bind9-host
```

Figure 35 : Téléchargement des paquets DNS

- Déclaration de la zone DNS

La première chose à faire est de choisir le domaine dans lequel les clients pourront explorer et enregistrer les services. Pour cela, il faut ajouter ces lignes dans le fichier de configuration :

```
//include "/etc/bind/zones.rfc1918";

zone "ugb.edu.sn" IN{
        type master;
        file "/etc/bind/db.ugb.edu.sn";
};

zone "1.168.192.in-addr.arpa" IN{
        type master;
        file "/etc/bind/db.ugb.edu.sn.rev";
};
```

Figure 36 : Déclaration de zone DNS

Nous définissons ici notre zone « ugb.edu.sn » nous indiquons ensuite que notre serveur DNS est le « maitre » de la zone. Nous précisons que les enregistrements se trouvent dans le fichier « db.ugb.edu.sn ». Puis nous activons les mises à jour dynamiques pour cette zone. Ainsi, les clients pourront eux-mêmes enregistrer leurs services sur le serveur.

- Définition de la zone de recherche

Ensuite, il faut créer et enregistrer le fichier db.<notre zone >, ici « db.ugb.edu.sn ». Voici ci-dessous une configuration type du fichier.

```
@       IN      SOA     lyam0.ugb.edu.sn. root.ugb.edu.sn. (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      lyam0.ugb.edu.sn.
lyam0   IN      A       192.168.1.1
@       IN      A       127.0.0.1
;@      IN      AAAA    ::1
```

Figure 37 : Définition de la zone de recherche

2.2 Installation de Avahi

Avahi est un système qui facilite la découverte de services sur un réseau local via la suite de protocoles mDNS/DNS-SD. Cela nous permet de brancher notre ordinateur portable ou un terminal personnel sur un réseau et de pouvoir instantanément voir d'autres personnes avec lesquelles on peut discuter, trouver des imprimantes sur lesquelles imprimer ou trouver des fichiers à partagés etc.

Avahi est principalement destiné aux systèmes Linux et est livré par défaut dans la plupart des distributions, raison pour laquelle nous l'avons choisi car on travaille sur un environnement Debian (Raspbian).

- Téléchargement et Installation des paquets Avahi

Avahi peut être installer avec yum et apt comme ci-dessous :

```
pi@lyam0:~ $ sudo apt install avahi-daemon avahi-utils avahi-discover
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
avahi-daemon is already the newest version (0.8-5).
avahi-discover is already the newest version (0.8-5).
avahi-utils is already the newest version (0.8-5).
The following packages were automatically installed and are no longer required:
  chromium-common chromium-sandbox libfuse2 libjsoncpp24 libminizip1 libre2-9 libu2f-udev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 33 not upgraded.
```

Figure 38 : Installation des paquets Avahi

- Configuration d'Avahi

Après avoir installé Avahi, comme nous le savons, Avahi fonctionne comme démon en général, nous allons démarrer le démon Avahi. En utilisant les scripts init, la commande suivant peut-être lancée. Avahi nécessite des privilèges root comme nous pouvons le voir dans la figure suivante, il demande le mot de passe root.

```
pi@lyam0:~ $ /etc/init.d/avahi-daemon start
Starting avahi-daemon (via systemctl): avahi-daemon.service==== AUTHENTICATING FOR org.freed
esktop.systemdl.manage-units ===
Authentication is required to start 'avahi-daemon.service'.
Authenticating as: ,,, (pi)
Password:
==== AUTHENTICATION COMPLETE ===
```

Les fichiers de configurations d'Avahi résident dans le répertoire racine du démon Avahi, pour ajouter un hôte au service DNS d'Avahi, c'est comme ajouter un fichier hôte Linux hosts. On ajoute la ligne suivante dans le fichier comme dans la figure ci-dessous :

192.168.1.1 laym0

```
# Examples:
#192.168.0.1 router.local
2001::81:1 lyam0.local
192.168.1.1 lyam0.local
```

- Enregistrement des services

Pour ajouter des services c'est un peu plus compliqué que l'ajout d'hôtes. Il un fichier de configuration XML utilisé pour décrire les services et ce fichier sera placé dans le répertoire de configuration de service, qui se trouve dans « /etc/avahi/services ». Comme nous pouvons l'observer sur les figures ci-dessous.

Le fichier de configuration de service suivant définit le fichier de notre lampe contrôler à distance qui est servi à partir du port tcp 80

```
<?xml version="1.0" standalone='no'?><!--*-nxml-*-->
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="no">lumière contoler à distance</name>
  <service>
    <type>_light._tcp</type>
    <host-name>lyam0.local</host-name>
    <port>80</port>
    <txt-record>url: http://192.168.1.13/</txt-record>
    <txt-record>foction :controler led à distace</txt-record>
    <txt-record> type_dispo : lampe </txt-record>
    <txt-record> util: bottom</txt-record>
    <txt-record> light</txt-record>
    <txt-record> type_ser : lampe connectée</txt-record>

  </service>
</service-group>
```

Figure 39 : Configuration du service de lumière

Ici nous avons configurer un service d'imprimerie avec les informations nécessaires pour l'utilisation du service par les utilisateurs.

```
<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name>imprimanteToto</name>
  <service>
   <type>_ipp._tcp</type>

   <subtype>_universal._sub._ipp._tcp</subtype>
   <port>631</port>
   <txt-record>txtver=1</txt-record>
   <txt-record>qtotal=1</txt-record>
   <txt-record>rp=priters/imprimmante1</txt-record>
   <txt-record>ty=imprimmante1</txt-record>
   <txt-record>adminurl=http://192.168.0.254/printers/imprimmante1</txt-record>
   <txt-record>note=Office Lasejet 4100n</txt-record>
   <txt-record>priority=0</txt-record>
   <txt-record>product=(GPL Ghostscript)</txt-record>
   <txt-record>printer-state=3</txt-record>
   <txt-record>printer-type=0x801046</txt-record>
   <txt-record>transparent=T</txt-record>
   <txt-record>Binary=T</txt-record>
   <txt-record>Fax=F</txt-record>
   <txt-record>Color=T</txt-record>
   <txt-record>Duplex=T</txt-record>
   <txt-record>Staple=F</txt-record>
   <txt-record>Copies=c</txt-record>    <txt-record>Bind=F</txt-record>
   <txt-record>Sort=F</txt-record>
   <txt-record>scan=F</txt-record>
   <txt-record>pdl=applicatio/octet-stream,applicatio/pdf,application/postscript,image/jpeg,image/p$
   <txt-record>URF=W8,SRG24,CP1,RS600</txt-record>
 </service>
</service-group>
```

Figure 40 : Configuration du service d'imprimérie

- Déclaration de services

Déclaration de nos services dans notre domaine « ugb.edu.sn » avec les enregistrements SRV, qui est une spécification de données dans le système de noms de domaine définissant l'emplacement, c'est-à-dire le nom d'hôte et le numéro de port, pour les services spécifiés, comme suite.

_service._proto.name. ttl IN SRV priority weight port Target.

```
;
@       IN      NS      lyam0.ugb.edu.sn.
lyam0   IN      A       192.168.1.1
_light._tcp.ugb.edu.sn.         1600 IN SRV      10      10      80      lyam0.ugb.edu.sn.

;_ftp._tcp.ugb.edu.sn.  600     IN      SRV     0       10      21      lyam0.ugb.edu.sn.
```

Figure 41 : Déclaration des services dans le domaine

- Découverte de services

Avahi-discover est un programme de ligne de commande qu'on peut utiliser pour rechercher toutes les diffusions DNS-SD sur le réseau et pour résoudre le nom d'hôte et l'adresse IP. Il suffit de taper avahi-discover et une application s'ouvre listant ainsi tous les services disponibles.

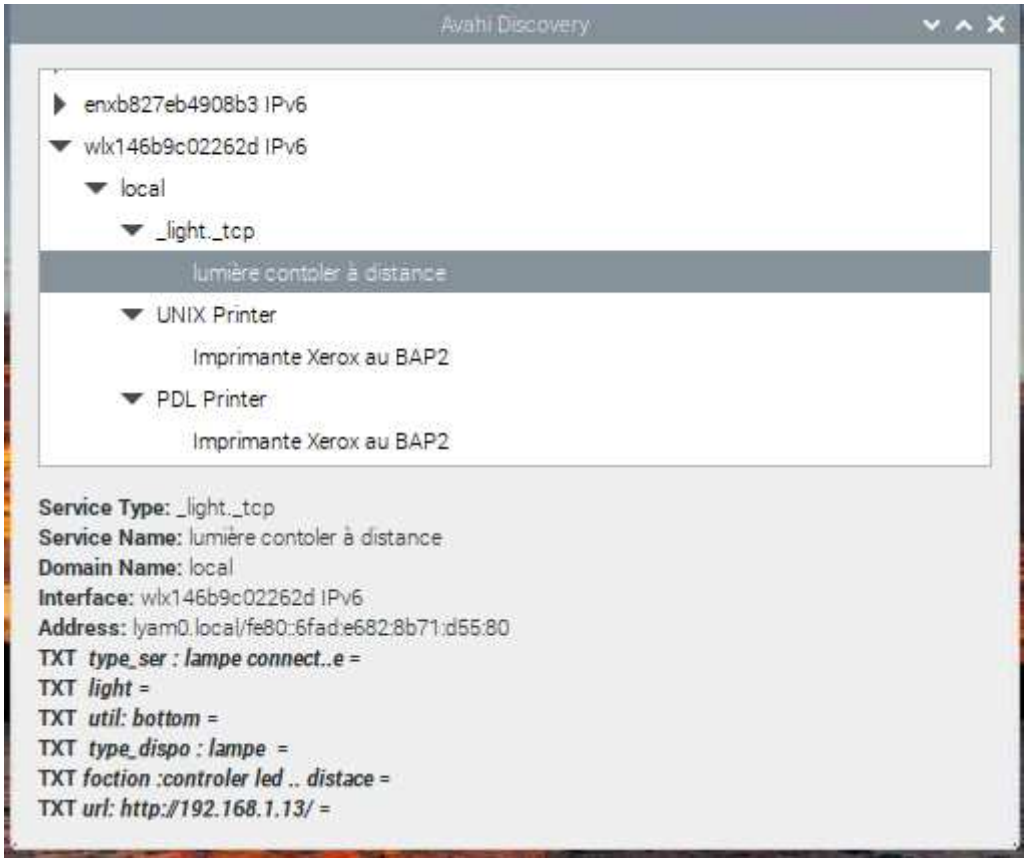


Figure 42 : Découverte des services

2.3 Installation du serveur DHCP

Pour utiliser DNS-SD, le client a besoin d'un configuration IP pour être relié au réseau, ainsi que de connaitre le domaine auquel il appartient. Une manière élégante d'obtenir ces informations est d'utiliser un serveur DHCP. Le client obtient ainsi la configuration IP (adresse IP, masque de sous-réseau, adresse broadcast, serveur DNS…).

- Pour installer un serveur DHCP sous Linux, il faut télécharger le paquet « isc-dhcp-server » avec la commande « **sudo apt-get install isc-dhcp-server** ».
- Configuration du serveur dans le fichier /etc/dhcp/dhcpd.conf

Pour les différentes étapes de la configuration voir annexes.

```
# A slightly different configuration for an internal subnet.
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.10 192.168.1.13;
  option domain-name-servers 192.168.1.1 ;
  option domain-name "ugb.edu.sn";
  option routers 192.168.1.1;
  option broadcast-address 192.168.1.255;
  default-lease-time 600;
  max-lease-time 7200;
}
```

Figure 43 : Configuration du serveur DHCP

3. Implémentation de la lampe connectée

Dans cette partie nous allons implémenter notre objet connecté, qui est ici une lampe qu'on peut contrôler à distance grâce à une liaison IP. Cette dernière sera reliée à un microcontrôleur qui va nous permettre de la contrôler via une page web.

Puis nous allons annoncer le service dans notre réseau local via le système de découverte de service.

- Câblage du système

Ici nous avons relié l'ensemble des circuits pour former notre objet connecté

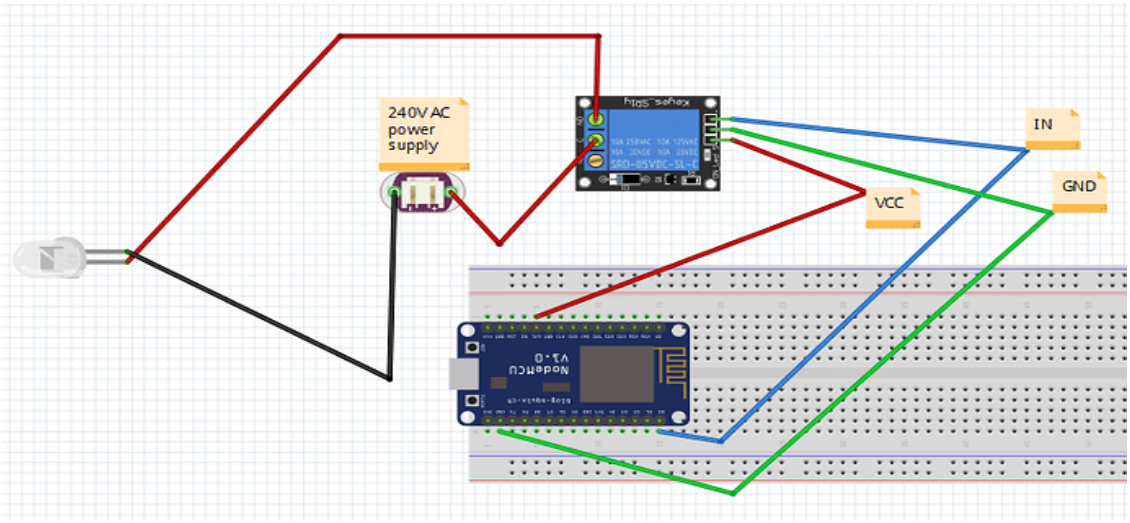


Figure 44 : Câblage du système

- Programme de contrôle

Pour faire fonctionner notre lampe à distance, nous avons réalisé un programme qui va nous générer une page web avec des fonctions « allumer » et « éteindre » pour le contrôle de la lampe.

Voici un extrait du programme, pour l'intégralité du code voire page annexe.

```
void loop()
{
WiFiClient client;


  // Vérification si le client est connecter.
  client = server.available();
  if (!client)
  {
    return;
  }

  // Attendre si le client envoie des données ...
  Serial.println("nouveau client");
  while(!client.available()){
    delay(1);
  }

  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  int value = LOW;
  if (request.indexOf("/LED=ON") != -1)  {
    digitalWrite(ledPin, HIGH); // allumer la led
    value = HIGH;
  }
  if (request.indexOf("/LED=OFF") != -1)  {
    digitalWrite(ledPin, LOW); // éteindre la led
    value = LOW;
  }

  // Réponse
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println("");
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");
```

Figure 45 : Programme

4. Attaque sur le système de découverte

Dans cette partie, nous allons essayer de simuler une attaque MiTM pour voir le comportement de notre système de découverte, pour cela nous avons utilisé une machine kali linux.

Kali Linux est une distribution GNU/Linux sortie en 2013, basé sur Debian. L'objectif de kali Linux est de fournir une distribution regroupant l'ensemble des outils nécessaires aux tests de sécurité d'un système d'information, notamment le test d'intrusion.

Nous avons utilisé Ettercap-graphical pour réaliser notre attaque, une fois qu'on lance le sniffing du réseau on peut lister l'ensemble des réseaux.



Figure 46 : Sniffing du réseau

Une fois qu'on a la liste des hôtes on peut sélectionner notre cible et passer à l'attaque, comme on peut voir ci-dessous nous avons lancer un dns spoofing.

```
Activating dns_spoof plugin...

dns_spoof: A [0.debian.pool.ntp.org] spoofed to [192.168.1.51] TTL [3600 s]
dns_spoof: A [0.debian.pool.ntp.org.ugb.edu.sn] spoofed to [192.168.1.51] TTL
 [3600 s]
dns_spoof: A [1.debian.pool.ntp.org] spoofed to [192.168.1.51] TTL [3600 s]
dns_spoof: A [1.debian.pool.ntp.org] spoofed to [192.168.1.51] TTL [3600 s]
dns_spoof: A [1.debian.pool.ntp.org.ugb.edu.sn] spoofed to [192.168.1.51] TTL
 [3600 s]
```

Dans notre, cas la machine attaquante et le serveur dns sont tous dans le même segment de réseau. Alors nous avons effectué un MiTM directement sur le serveur.

5. Configuration de DNSSEC

Les mises à jour du DNS avec la fonction DNS dynamique posent de gros problèmes de sécurité. Un individu malintentionné pourrait annoncer des services inexistants et l'utilisateur aurait du mal à trouver les services réellement disponibles. Pire, ce même utilisateur malintentionné pourrait mettre à disposition des utilisateurs de virus (par exemple via des pages web vérolés).

C'est pourquoi, il est plus que recommandé d'utiliser DNSSEC (DNS Sécurisé). DNSSEC permet de définir les personnes habilitées à effectuer des mises à jour dynamiques. Pour cela les échanges sont cryptés à l'aide d'une clé partagée.

- Activation de DNSSEC

Tout d'abord, il est nécessaire d'activer DNSSEC en ajoutant les directives de configuration suivantes à l'intérieur des options {} dans le fichier /etc/bind/named.conf.options comme suite :

```
        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-enable yes;
        dnssec-validation yes;
        dnssec-lookaside auto;
```

Figure 47 : Activation de DNSSEC

Puis il faut naviguer dans l'emplacement des fichiers de zone avec la commande cd /var/cache/bind.

Ensuite, on doit construire la clé de signature de zone qui est connu sous le nom de ZSK, avec la suite des commandes spécifiques suivantes :

sudo dnssec-keygen –a NSEC3RSASHA1 –b 2048 –n ZONE 'domain.com'

```
pi@lyam0:/var/cache/bind $ sudo dnssec-keygen -a NSEC3RSASHA1 -b 2048 -n ZONE ugb.edu.sn
Generating key pair.....................+++++ .....................+++++
Kugb.edu.sn.+007+00920
pi@lyam0:/var/cache/bind $
```

Figure 48: Génération de ZSK

Création de clé de signature de clé (KSK) avec la commande suivante :

Sudo dnssec-keygen –f KSK –a NSEC3RSASHA1 –b 4096 –n ZONE 'domaine.com'

```
pi@lyam0:/var/cache/bind $ sudo dnssec-keygen -f KSK -a NSEC3RSASHA1 -b 4096 -n ZONE ugb.edu.sn
Generating key pair.....................................................................................
...++++ .....................................................................................
...++++
Kugb.edu.sn.+007+14248
pi@lyam0:/var/cache/bind $
```

Figure 49 : Géneration de KSK

```
pi@lyam0:/var/cache/bind $ more Kugb.edu.sn.+007+14248.key
; This is a key-signing key, keyid 14248, for ugb.edu.sn.
; Created: 20220118000644 (Tue Jan 18 00:06:44 2022)
; Publish: 20220118000644 (Tue Jan 18 00:06:44 2022)
; Activate: 20220118000644 (Tue Jan 18 00:06:44 2022)
ugb.edu.sn. IN DNSKEY 257 3 7 AwEAAdWOfmySbHch3Aa4tFCb1CkHfmgviYEY9l0+VoM0+MawR+tRVuvj RU9OYCU1
2F5Zfs8+NjzkXTtZfsQYGcLkCyD/5tfyMmyawJdiL0xrL+tf px+UKKSB0zz3HaKIyySI0ByqK6T/AFa7xl4Kcki74xeYrf
3MvPelxIXd AbEJfw0zw4ZaJGGib6BbZCw756syddLWtryO23PwrYIZOZCMo2x9X0Ck 65tM7ijjcAwH5Tw3qB0/KH13T2N
TMsxh6gFkjLe+9Yjtzb7Jbb/Uyn8W ixvDKG85DRKpTl/KqnPqur/Ch/ADcuimX/vxPcmzGjU9PuYEVY+J5865 c7z3gWVj
kOwHeHxdEMqAHnSjNnPAEcrCEoNWNNW4GuixvNQDQ15YMWgU ZrF02argX3USvG/qCZP4lIuoSytXAbgqaJxRg9yCTqsQvs
ZPBzPP0HTQ RxT0JWQhIi1HgPNlg1msCjt5VIWV3AMtLrMYg43G1X2vysyAzW3qfIQA HRAoisvkJQbmW2nbM1GwFR8wF3h
MX6ta1aBLsoqTDyZHjrLMeNmNL9ug cXQXxb/BMO5MancNDms8Ph3Dg/i/zzIRSpqoRmyGZJaeRzsSS5ts8i7l xCPiCnPy
ItHS/rQzgT+v3e6XwWRpTavDaVMqrWzomVhv9exrOd+xRQpw rdtFxvxqK4Im+aSn
pi@lyam0:/var/cache/bind $
```

Figure 50 : La clé partagée

Il est nécessaire de signer la zone à l'aide de la commande dnssec-signzone suivante :

Sudo dnsssec-signzone -3 –A –N INCREMENT –o -t

Maintenant on doit remplacer par quelque chose d'aléatoire, on peut utiliser la chaine de 16 caractères pour entrer comme le sel avec la commande suivante :

head –c 1000 /dev/random | sha1sum | cut –b 1-16

```
pi@lyam0:/var/cache/bind $ sudo dnssec-signzone -o ugb.edu.sn -N INCREMENT -t -k Kugb.edu.sn.+0
07+14248.private /etc/bind/db.ugb.edu.sn Kugb.edu.sn.+007+00920.private
Verifying the zone using the following algorithms:
- NSEC3RSASHA1
Zone fully signed:
Algorithm: NSEC3RSASHA1: KSKs: 1 active, 0 stand-by, 0 revoked
                         ZSKs: 1 active, 0 stand-by, 0 revoked
/etc/bind/db.ugb.edu.sn.signed
Signatures generated:                      11
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                1.519
Signatures per second:                  7.236
Runtime in seconds:                     1.639
pi@lyam0:/var/cache/bind $
```

Figure 51 : Signature de la zone

On peut observer que notre zone a été bien signé avec la clé de signature de zone (ZSK), et cette dernière a été signée avec la clé de signature de clé (KSK), en vérifiant notre fichier de zone, puis on tape la commande dig pour vérifier le domaine comme suite :

dig DNSKEY ugb.edu.sn. @localhost +multiline

```
pi@lyam0:/etc/bind $ dig DNSKEY lyam0.ugb.edu.sn. @localhost +multiline

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> DNSKEY ugb.edu.sn. @localhost +multiline
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43986
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;ugb.edu.sn.            IN DNSKEY

;; ANSWER SECTION:
ugb.edu.sn.         86400 IN DNSKEY   256 3 7 (
                AwEAActPMYurNEyhUgHjPctbLCIlVuSj3xcjI8QFTpdM
                8k3cYrfwB/WlNKjnnjt98nPmHv6frnuvs2LKIvvGzz++
                kVwVc8uMLVyLOxVeKhygDurFQpLNNdPumuc2MMRvV9me
                fPrdKWtEEtOxq6Pce3DW2qRLjyElnloEq44gixn6hjgo
                sG2FzV4fTQdxdYCzlYjsaZwy0Kww4HpIaozGNjoDQVI/
                f3JtLpE1MYEb9DiUVMjkwVR5yH2UhJwZH6VVvDOZg6u6
                YPOSUDVvyofCGcICLqUOG+qITYVucyIWgZtHZUb49dpG
                aJTAdVKlOTbYV9sbmHNuMuGt+l/rc+StsjTPTHU=
                ) ; key id = 40400
ugb.edu.sn.         86400 IN DNSKEY   257 3 7 (
                AwEAAa2BE0dAvMs0pe2f+D6HaCyiFSHw47BA82YGs7Sj
                qSqH3MprNra9/4S0aV6SSqHM3iYZt5NRQNTNTRzkEl8e
                3j9AGV8JA+xbEow74n0eu33phoxq7rOpd/NlGpCrxUsG
                kK4PDkm+R0hhfufelZOSoiZUV7y8OVGFB+cmaVb7sYqB
                RxeWPilZ6Fjl/5oKwB6Zqbs7s7pmxl/GcjTvdQkMFtOQ
```

Maintenant on peut bien observer que notre zone a été signée avec la clé partagée générer précédemment.

Voici la clé partagée pour notre zone de recherche « ugb.edu.sn »

```
root@lyam0:/var/cache/bind# cat dsset-ugb.edu.sn.
ugb.edu.sn.             IN DS 14248 7 2 310CDAF6582D62BB70A7CA24F8060AF68B52EA6E
86AED86FFA249498 9965DA38
root@lyam0:/var/cache/bind#
```

6. Conclusion

Dans ce chapitre nous avons pu voir comment mettre en place un système de découverte de services, les différentes expérimentations techniques réalisées, à travers ce chapitre se focalisent sur comment le système DNS-SD fonctionne avec l'environnement IoT. Nous avons vu comment sécurisé le système de noms de domaine, avec le déploiement d'une contre-mesure qui permet une sécurité sur non seulement du système en entier, mais particulièrement des et cette dernière est le DNSSEC que nous avons développé dans le chapitre précédent.

V. CONCLUSION GENERALE

La grande évolutivité de l'IoT dans notre environnement nécessite un mécanisme de gestion des ressources capable d'enregistrer et de découvrir des ressources et des services de manière auto-configurée, efficace et dynamique. Les protocoles dominant dans ce domaine sont le DNS Service Discovery (DNS-SD) et le DNS multidiffusion qui peuvent découvrir les ressources et les services offerts par les appareils IoT. Bien que le protocole DNS-SD ait été conçu à l'origine pour les appareils riches en ressources, il existe des études de recherche qui ont adapté des versions légères de celui-ci aux environnements IoT. Grace à ce protocole, les clients peuvent découvrir un ensemble de services souhaités dans un réseau spécifique en utilisant des requêtes DNS standard. Essentiellement DNS-SD utilise le mDNS pour envoyer des paquets DNS à des adresses de multidiffusion spécifiques. L'IoT a besoin d'une sorte d'architecture sans dépendre d'un mécanisme de configuration. Dans une telle architecture, les appareils intelligents peuvent rejoindre la plate-forme ou la quitter sans affecter le comportement de l'ensemble du système, DNS-SD peut faciliter ce mode de développement. Le protocole le plus étendu pour la découverte de services dans l'Internet actuel est le DNS.

La réalisation du DNS-SD, en tant que projet, est encore dans ses débuts. Ainsi, le problème de sécurité dans le système est classé parmi les plus grands défis de cette réalisation. Pour cette raison, cet article analyse les exigences et les problèmes de sécurité pour appliquer cette technique de découverte dans l'Internet des objets, en terminant par un ensemble de recommandations et de méthodes pour construire une implémentation de DNS-SD pour la découverte des ressources.

En résumé voici des fonctionnalités que doit avoir un protocole de découverte de service et pourquoi s'appuyer sur le protocole DNS est souhaitable.

- La découverte de service requiert un serveur central pour regrouper les informations, qui est le serveur DNS
- La découverte de service requiert un protocole d'enregistrement de service.
- La découverte de service requiert un protocole de requête, qui est le DNS
- La découverte de service requiert des mécanismes de sécurité, le DNS en a déjà : DNSSEC
- La découverte de service en multicast pour les réseaux sur le lien local.

Ce mémoire a présenté une approche pour l'intégration transparente des objets intelligents dans l'internet avec le système de découverte, et inclus une description de la manière dont le protocole est mis en œuvre.

Pour conclure, il est plus raisonnable d'utiliser des systèmes déjà en place, qui sont utilisés sur tous les réseaux, plutôt que de déployer un système parallèle juste pour de la découverte de services. Raison pour laquelle, il est recommandé d'utiliser le système de découverte de service basé sur DNS (DNS-SD) en tant que protocole de découverte.

Webographie

[1] https://www.linformaticien.com/dossiers/58069-4dns-ce-protocole-qui-ne-vous-veut-pas-que-du-bien.html visité le 09 12 2021

[2] https://www.cigref.fr/archives/histoire-cigref/blog/histoire-des-objets-inanimes-aux- objets-connectes/ visité le 10 09 2021

[3] https ://ns1.com/dns-security visité le 03/01/2022

[4] https://www.akamai.com/fr/fr/resources/our-thinking/threatadvisories/Akamai zeroconf-network visité le 18 09 2021

[5] https ://docs.oracle.com/cd/E19620-01/805-1608/networkapp-5/index.html visité le 15 12 2021

[6] https://tools.ietf.org/id/draft-ietf-dnssd-prireq-00.html#rfc. section.4.6 visité le 13 09 2021

[7] https://support.google.com/a/answer/48090?hl=fr visité le 15 12 2021

[8] https ://tools.ietf.org/id/draft-ietf-dnssd-prireq-00.html#rfc.section.4.4 visité le 22/08/2021

[9] https ://tools.ietf.org/id/draft-ietf-dnssd-prireq-00.html#rfc.section.4.5 visité le 12/12/2021

[10] https://www.akamai.com/fr/fr/resources/our-thinking/threatadvisories/Akamai mDNS-reflection-ddos-threat-advisory.jsp visité le 30/12/2021

[11] https://www.engineersgarage.com/ Service Delivery Protocols -uPnP : IOT Part 10 visité le 08/01/2022

[12] https://www.engineersgarage.com/ Service Delivery Protocols -SSDP : IOT Part 10 visité le 20/12/2021

[13] http://Historique | les-objets-connectes (jeanchavanat.wixsite.com) visité le 09/12/2021

[14] Formation Objets connectées et Internet des Objets (capgemini.fr) visité le 12/1/2021

[15]https://www.fda.gov/radiation-emitting-products/electromagnetic-compatibility-emc/radio-frequency-identification-rfid visité le 08/10/2021

[16] https://wfiot2018.iot.ieee.org/sps2-military-applications-iot/ visité le 23/10/2021

[17] https://www.blackberry.com/fr/fr/solutions/iot-internet-of-things/iot-healthcare visité le 25/10/2021

[18] https://gladiacteur.com/iot-linternet-des-objets/ visité le 30/12/2021

[19] https://www.avast.com/fr-fr/c-iot-security visité le 10/12/2021

[20] https ://ns1.com/dns-security visité le 12/01/2022

[21] https ://ns1.com/dns-security/mitm visité le 12/01/2022

[22] https ://blog.netwrix.fr/2018/07/04/les-10-types-de-cyberattaquesles-plus-courants/ visité le 17/11/2021

[23] https ://vigilance.fr/vulnerabilites/Avahi-1009000visité le 18/11/2021

[24] https://www.nameshield.com/ressources/lexique/dns-cache-poisoning/ visité le 18/11/2021

[25] https ://www.cloudflare.com/fr-fr/dns/dnssec/how-dnssec-works/ visité le 30/12/2021

[26] https://www.nameshield.com/ressources/lexique/dns-spoofing/ visité le 19/11/2021

[27] https://tools.ietf.org/id/draft-ietf-dnssd-prireq-00.html#rfc.section.4.2 visité le 12/10/2021

[28] https://docs.oracle.com/cd/E19620-01/805-1608/networkapp-5/index.html visité le 12/01/2022

[29] https://www.cloudflare.com/fr-fr/dns/dnssec/how-dnssec-works/resource-record visité le 13/01/2022

[30] https://www.cloudflare.com/fr-fr/dns/dnssec/how-dnssec-works/rrsig visité le 13/01/2022

[31] https ://www.cloudflare.com/fr-fr/dns/dnssec/how-dnssec-works/delegation-signer visité le 13/01/2022

[32] https ://www.cloudflare.com/fr-fr/dns/dnssec/how-dnssec-works/ visité le 15/01/2022

ANNEXES

- Configuration du serveur DHCP

```
# A slightly different configuration for an internal subnet.
subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.50 192.168.0.200;
  option domain-name-servers lyamlocal.lyamlocal.sn;
  option domain-name "lyamlocal.sn";
  option subnet-mask 255.255.255.0;
  option routers 192.168.0.1;
  option broadcast-address 192.168.0.255;
  default-lease-time 600;
  max-lease-time 7200;
}
```

On déclare la plage d'adressage, le domaine de recherche et le Gateway.

- Déclaration de zone

Modification de l'option de fichier dans le cadre de l'emplacement de la zone.

```
zone "ugb.edu.sn" IN{
        type master;
//      file "/etc/bind/db.ugb.edu.sn";
        file "/etc/bind/db.ugb.edu.sn.signed";
        auto-dnssec maintain;
        inline-signing yes;

};
```

Configuration du domaine de résolution DNS

```
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     lyamlocal.lyamlocal.sn. root.lyamlocal.lyamlocal.sn. (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

@       IN      NS      lyamlocal.lyamlocal.sn.
lyamlocal       IN      A       192.168.0.254
client  IN      A       192.168.0.253
www     IN      CNAME   lyamlocal.lyamlocal.sn.
```

- Configuration de DNSSEC

Maintenant on enregistre le fichier et on recharge la liaison, puis on vérifie l'utilisation des enregistrements DNSKEY avec la commande suivante :

On peut observer que notre zone a été bien signé avec la clé de signature de zone (ZSK), et cette dernière a été signée avec la clé de signature de clé (KSK), en vérifiant notre fichier de zone.

```
_light._tcp.ugb.edu.sn.          1600 IN SRV      10      10      80      lyam0.u>

;_ftp._tcp.ugb.edu.sn.   600      IN       SRV      0       10      21      lyam0.u>

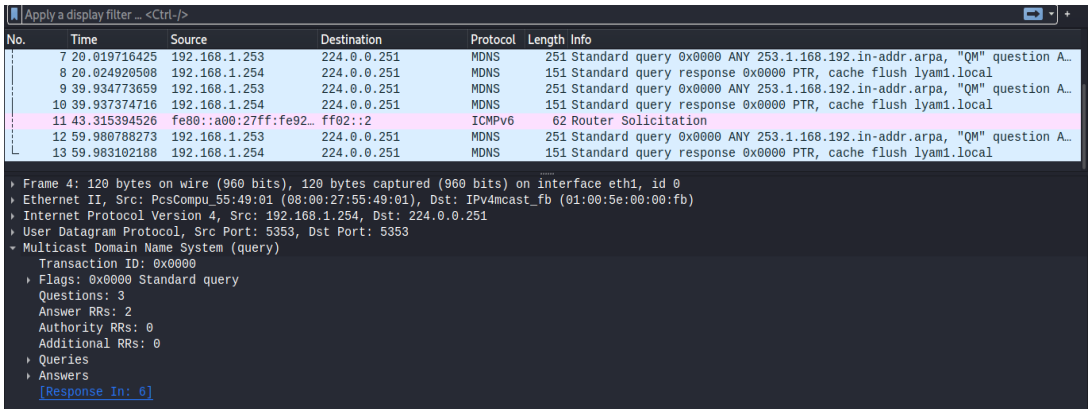
$INCLUDE "/var/cache/bind/Kugb.edu.sn.+007+00920.key"
$INCLUDE "/var/cache/bind/Kugb.edu.sn.+007+14248.key"


;@       IN       A        127.0.0.1
;@       IN       AAAA     ::1
; This is a zone-signing key, keyid 920, for ugb.edu.sn.
; Created: 20220118000408 (Tue Jan 18 00:04:08 2022)
; Publish: 20220118000408 (Tue Jan 18 00:04:08 2022)
; Activate: 20220118000408 (Tue Jan 18 00:04:08 2022)
ugb.edu.sn. IN DNSKEY 256 3 7 AwEAAbepfdfSsbp7ira/+RGQOfclSAvID0t0ViPqxjVuuMx24>
; This is a key-signing key, keyid 14248, for ugb.edu.sn.
; Created: 20220118000644 (Tue Jan 18 00:06:44 2022)
; Publish: 20220118000644 (Tue Jan 18 00:06:44 2022)
; Activate: 20220118000644 (Tue Jan 18 00:06:44 2022)
ugb.edu.sn. IN DNSKEY 257 3 7 AwEAAdWOfmySbHch3Aa4tFCblCkHfmgviYEY9l0+VoM0+MawR>
```

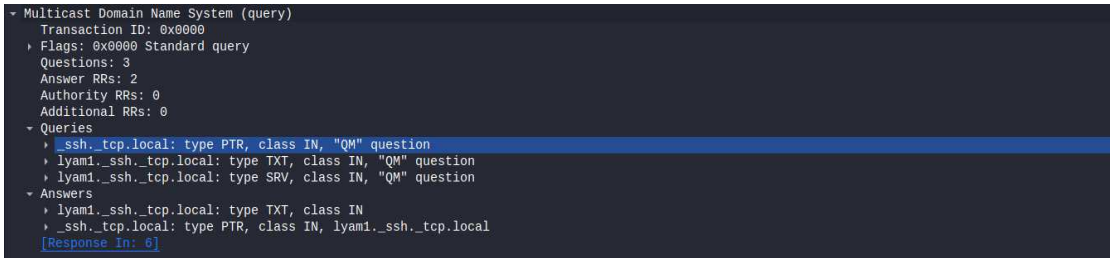
- La signature de zone « ugb.edu.sn »

```
; File written on Tue Dec 28 03:47:21 2021
; dnssec_signzone version 9.16.1-Ubuntu
ugb.edu.sn.             604800  IN SOA  ugb.edu.sn. root.lyam.ugb.edu.sn. (
                                        6           ; serial
                                        604800      ; refresh (1 week)
                                        86400       ; retry (1 day)
                                        2419200     ; expire (4 weeks)
                                        604800      ; minimum (1 week)
                                        )
                        604800  RRSIG   SOA 7 3 604800 (
                                        20220127024721 20211228024721 53589 ugb.edu.sn.
                                        c9gnELIBa+U6KzGf7UPKGpGmvQ6Ag5TkQUPm
                                        jwtaJyNOOmNgRNLR/Hum5S2wmVAzBpgqbwFI
                                        bawwRpIwUR52YgZ1hbhfbhTadq5yOsnWJvdN
                                        WHvuCH5GIzVDP77mFSMgrILjMkBH17HMSSul
                                        5L9xm2FbsDp3fJ6UDuWP8cXnpbRzLiRbCJpn
                                        k5p8xyHi+sjvxNrxvlkQmqrez8+Vww+DW5Yy
                                        tlUyNt34NvponSSp7UPcZt7FXfDOvGb/fijA
                                        KxBdETKVpar+62b/qsyb42tOkonuUiyX33Tf
                                        BHuk3JII4DZ+G9OOON318W4c2BWiciM+Bnpb
                                        kz1Tqj36mvtUlLWAjQ== )
                        604800  NS      lyam.ugb.edu.sn.
                        604800  RRSIG   NS 7 3 604800 (
                                        20220127024721 20211228024721 53589 ugb.edu.sn.
                                        n6dHCSF4qzUD1XbgVibXxM9C0bWjeEKlonoc
                                        JqZWLXPnuEY7dgAHsva55Gq/Lt/V/MoTCUf1
                                        PLJD2YZWgStduwX/NQyu1oVNla+O4JcL0mnE
                                        DiaJoXwrz/I43QOZ7Bl0n9cYrbKTnRvn9VQI
                                        keLlO8j2FPOAfTaNJ/7HONqEo/zc+bxiPV4q
                                        piSOmwHexbsrU5tRN7DMGQ+jb49+poxvEJGs
                                        aic6KvapkMO3MeDyNlUdzdmzG2S5clhgJ7d1
                                        1wqKfxkZT61qPkeZJWrOUOKIhfn8pNbrjoLS
                                        SHvyjPKyJcRiHMQytyJ9AOdOc8FoyfPFCfJf
```

- Utilisation de wireshark pour capturer les paquets de multicast DNS



Observe les services publier sur le réseau par notre système DNS-SD.



Dans cet exemple le client à effectuer trois requêtes (dans la section Queries) PTR et en a reçu deux comme on peut le voir dans la section Answers