UE L318 - Semaine 1

Hugo Lignères

22/02/2025

Table des matières

1 DNS

1.1 Le protocole DNS en détail

Le **Domain Name System** est un protocole qui permet de traduire le nom de domaine d'un site web en adresse IP qu'un serveur peut comprendre.

Son architecture est la suivante :

1. **Résolveur DNS** → Reçoit et gère les requêtes DNS qui proviennent de la machine côté client. Ensuite, il entre en contact avec les serveurs DNS pour traduire le nom de domaine en adresse IP ;
2. **Serveurs racines** → Points d'entrée d'une résolution DNS. Ces serveurs sont au nombre de 13. Ils redirigent les requêtes vers les serveurs TLD adéquats ;
3. **Serveurs TLD (Top-Level Domain** → Ils gèrent les noms de domaines comme `.com`, `.net`, `.org`, etc.
4. **Serveurs de noms** → Ces serveurs contiennent des enregistrements DNS pour des noms de domaine spécifiques.
5. **Enregistrements DNS** → Ce sont des entrées stockées dans une base de données DNS qui contiennent des informations sur des noms de domaine et leurs adresses IP correspondantes ;

1.2 Les détournements DNS

1.2.1 DNS Hijacking

Cette attaque consiste à modifier les enregistrements DNS d'un domaine, afin de rediriger le traffic vers des sites malveillants. Le plus souvent, ces attaques se produisent en corrompant les registres DNS d'un domaine, d'un serveur ou d'un routeur.

Tout simplement, un attaquant infiltre un ou plusieurs serveurs d'un site souvent visité, pour rediriger les utilisateurs vers des sites malveillants, souvent contrôlés par les attaquants.

1.2.2 Man-in-the-Middle

Un attaquant intercepte des requêtes et réponses DNS, pour altérer les informations contenues dans ces dernières, et renvoyer les utilisateurs vers d'autres sites, souvent malveillants.

Un exemple courant est de créer un faux point d'accès wi-fi gratuit dans un espace publique (aéroport, gare, restaurant, parc, etc), pour intercepter le traffic DNS des utilisateurs connectés. Il peut ensuite rediriger ces utilisateurs vers des sites qui lui permettent récupérer les données sensibles de ses victimes.

1.2.3 Empoisonnement du cache DNS (DNS Spoofing)

Un attaquant introduit de fausses informations DNS directement dans le cache d'un résolveur DNS. Cela a pour effet de retourner de fausses adresses IP, et d'ensuite renvoyer des utilisateurs vers des sites malveillants.

En 2008, Le Pakistan a tenté de censurer YouTube en redirigeant les utilisateurs vers une fausse adresse IP. Le résolveur DNS de nombreux FAI a mis en cache cette fausse information, affectant ainsi les utilisateurs hors du Pakistan

1.2.4 DNS Tunneling

Dans cette attaque, les requêtes et réponses DNS contiennent des données non-DNS (commandes, fichiers, etc). Cela permet à un attaquant d'outrepasser les pare-feus et filtrages.

Un attaquant peut utiliser un malware peut encapsuler les requêtes DNS et exfiltrer les données hors du réseau sécurisé.

1.2.5 Déni de service (DDOS)

Le but de cette attaque est d'inonder un serveur DNS ou un site web d'un nombre important de requêtes massives, rendant la cible inaccessible pour les utilisateurs.

En 2020, Cloudflare, un fournisseur majeur de services DNS, a subi une attaque DDoS massive ciblant ses serveurs DNS. L'attaque, dépassant 1,1 Tbps, exploitait des requêtes DNS amplifiées, où des attaquants envoyaient de petites requêtes à des résolveurs DNS mal configurés, qui renvoyaient des réponses volumineuses vers les serveurs de Cloudflare. Cela a entraîné des ralentissements massifs pour de nombreux sites utilisant leurs services.

1.2.6 Interception des paquets

Ici, un attaquant intercepte et modifie les paquets DNS en transit pour manipuler les réponses à des requêtes DNS.

Le gouvernement chinois a utilisé une technique d'interception DNS dans le cadre de son projet "Great Cannon". Lorsqu'un utilisateur accédait à des sites censurés (comme Google ou Facebook), ses requêtes DNS étaient interceptées et modifiées pour rediriger vers des pages d'avertissement ou vers des sites de propagande.

1.3 Solutions aux différents types d'attaques

- Utiliser un pare-feu DNS ;
- Implémenter des extensions DNS (DNSSEC) ;
- Exiger une authentification multifactorielle ;
- Surveiller le traffic DNS pour prévenir d'activités suspectes ;
- Isoler les parties critiques d'un réseau en le segmentant ;
- Effectuer régulièrement des mises à jour.

1.4 Exemple réel de détournement DNS

En août 2013, certains site internet d'entreprises Américaines ont été pris pour cible par des hackers Syriens, notamment Twitter, le Huffington Post et le site du New York Times. Trois types d'attaques on été utilisés :

— DNS Hijacking : Les enregistrement DNS des sites ciblés ont été modifié en accédant aux paramètres de configuration du registrar de Melbourne IT. Ainsi, Les noms de domaines légitimes des sites internet visés renvoyaient en fait vers des sites malveillants, choisis par les attaquants ;

— Interception des paquets DNS : En se servant d'une faille de sécurité d'un revendeur du registrar de Melbourne IT, les attaquants ont pu avoir accès aux identifiants d'accès aux

serveurs DNS. En conséquence, des emails ont été intercepté, et certaines pages des sites ciblés ont été effacés pour afficher à la place un message choisi par les attaquants ;

— Empoisonnement du cache DNS : L'attaque a corrompu le cache DNS des sites, donc la propagation des bons DNS a pris du temps, ce qui a fait durer l'attaque plus longtemps que si ça n'avait pas été le cas.

2 OpenSSL & Certification

2.1 Génération du certificat auto-signé

```
administrateur@pc-hugo ~/D/C/L/L/S/certificats> openssl genpkey -algorithm RSA -out private.key
.........+..............+...+..........+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++*............+......+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.
..+.........+.......+.............+..........+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++
.+....+.....+....+...........+...+...+............+.+.....+++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++*..........+...+..+......+.....................++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++*..+.+.....+.......+..+.............+.+...................
...+.....+...+.............................+..+.............+...+...+............+...+....+......
+.....+......+.........+.+...+..+...+......+.......+...+..+.........+...+.....................+...
......+.......+...+...+........+.+......+..+.+..+....+.....+.+.............+....+++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++
```

Étapê 1 - Génération de la clé privée

```
administrateur@pc-hugo ~/D/C/L/L/S/certificats [127]> ls
private.key
administrateur@pc-hugo ~/D/C/L/L/S/certificats> openssl req -new -key private.key -out csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Pays de la Loire
Locality Name (eg, city) []:Angers
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Université de Limoges
Organizational Unit Name (eg, section) []:Campus CVTIC
Common Name (e.g. server FQDN or YOUR name) []:Hugo Lignères
Email Address []:hugo.ligneres@orange.fr

Please enter the following 'extra' attributes
to be sent with your certificate request
```

Étape 2 - Génération de demande de signature du certificat

```
administrateur@pc-hugo ~/D/C/L/L/S/certificats> ls
csr.pem  private.key
administrateur@pc-hugo ~/D/C/L/L/S/certificats> openssl req -x509 -days 365 -key private.key
-in csr.pem -out certificat.crt
Warning: No -copy_extensions given; ignoring any extensions in the request
```

Étape 3 - Génération d'un certificat auto-signé

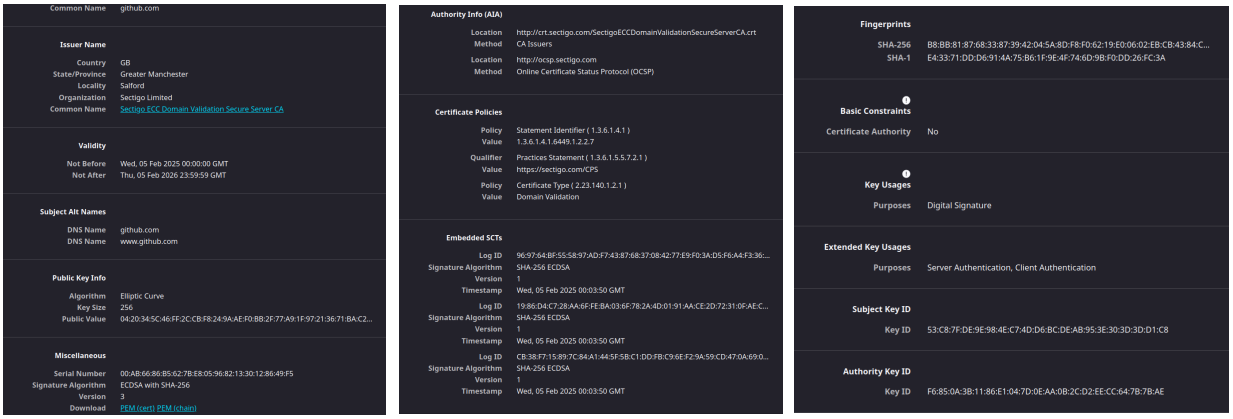
2.2 Détails du certificat

Explication des détails du certificat :

```
administrateur@pc-hugo ~/D/C/L/L/S/certificats> ls
certificat.crt  csr.pem  private.key
administrateur@pc-hugo ~/D/C/L/L/S/certificats> openssl x509 -in certificat.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            4e:bc:cb:a5:c3:e1:54:07:be:82:4e:db:54:ae:e9:0b:a3:1c:5e:21
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = FR, ST = Pays de la Loire, L = Angers, O = Universit\C3\83\C2\A9 de Limoges, OU = Campus CVTIC, CN = Hugo Lign\C3\83\C2\A8res, emailAddress = hugo.ligneres@orange.fr
        Validity
            Not Before: Feb 22 15:05:48 2025 GMT
            Not After : Feb 22 15:05:48 2026 GMT
        Subject: C = FR, ST = Pays de la Loire, L = Angers, O = Universit\C3\83\C2\A9 de Limoges, OU = Campus CVTIC, CN = Hugo Lign\C3\83\C2\A8res, emailAddress = hugo.ligneres@orange.fr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b2:2c:8a:5a:b4:96:a9:a8:51:e9:a6:5b:36:c9:
                    c2:7d:ba:0c:0d:3e:ac:78:a2:ad:43:ca:cb:c9:e1:
                    7e:5e:0d:96:71:f3:e2:ff:f8:52:2c:c4:34:fb:f0:
                    53:1a:b0:e8:b3:10:b0:59:93:7b:cf:1f:a2:cc:ec:
                    a8:7b:f9:31:02:13:52:dc:02:a8:26:5e:f9:e7:81:
                    aa:99:a4:1f:f9:6e:31:1a:c0:61:6b:04:aa:c0:5a:
                    b0:e7:cb:20:59:4d:ce:6f:30:38:a6:c3:2b:af:e5:
                    91:e1:7b:66:8b:6f:e2:6a:83:45:61:4b:d0:b5:1f:
                    d2:6a:32:fa:83:7b:1c:d9:fe:20:f4:9f:f3:79:17:
                    d7:b7:a3:68:8e:dd:03:0a:85:4e:0c:2d:18:94:ed:
                    47:dc:bf:d7:b4:53:e4:2d:ab:61:de:72:50:d1:a3:
                    eb:9d:53:41:e2:69:f3:67:53:2c:c8:8a:ec:f4:08:
                    dc:8d:04:8c:2f:05:c2:0c:39:3b:d1:ee:af:32:9f:
                    91:25:1c:e9:c8:cf:5c:4c:33:a4:2a:89:33:6c:7d:
                    4b:51:9b:df:c1:73:ce:ec:df:dd:58:63:40:2f:26:
                    49:b0:08:ac:6c:d9:f3:0c:e6:0e:ad:de:d4:e5:e1:
                    87:05:cb:0d:27:fc:24:32:6a:2b:58:d2:f2:0e:1c:
                    98:65
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                A9:1E:4D:7A:F8:7C:23:C2:E6:BF:B4:DA:1C:C1:29:D4:0F:E1:E3:EA
            X509v3 Authority Key Identifier:
                A9:1E:4D:7A:F8:7C:23:C2:E6:BF:B4:DA:1C:C1:29:D4:0F:E1:E3:EA
            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        a4:8c:9f:c4:fe:82:61:3c:95:4c:a7:6c:1f:b0:e5:35:65:1b:
        73:09:61:f9:2b:84:b7:86:2b:d4:ee:e2:06:a7:fd:dc:ab:2c:
        a5:01:ec:38:54:8b:97:5c:c4:9f:23:68:7a:6b:72:91:98:7d:
        3f:9b:22:17:e0:77:07:ca:ac:d3:14:7a:16:af:6f:c6:e1:92:
        00:0d:61:92:63:77:7c:23:b7:88:1e:38:31:c9:ba:d2:18:6f:
        ce:5b:ae:61:02:7f:7a:f8:b3:04:cb:a5:66:3b:0c:5a:5f:a2:
        d8:76:f2:de:ae:c9:b9:4e:4d:f1:9b:6b:f5:f5:2e:54:d6:e0:
        e1:89:09:19:85:95:28:ba:0a:1d:0d:cd:12:66:72:00:07:0a:
        f0:24:08:fa:1c:07:4b:c8:0e:38:36:dd:c6:f1:1c:e7:15:04:
        5c:f9:cd:c7:6a:e1:5b:75:03:ab:71:1a:56:09:b5:a4:cf:bc:
        c4:ed:b8:fc:d0:bc:fc:f6:1c:a6:81:ed:c4:41:75:2c:69:9c:
        bd:eb:4b:63:78:8a:8b:31:84:77:ad:cf:3e:0d:39:ce:39:20:
        17:41:b6:2f:24:9e:8c:a4:00:e6:a7:43:08:70:2e:e3:13:bd:
        ff:17:6f:05:9a:68:d9:a8:30:d1:23:9d:4c:5b:f1:9f:d2:12:
        ed:68:83:fa
```

— **Version : 3 (0x2)** →
— **Serial number** → numéro de série unique du certificat ;
— **Signature algorithm** → Algorithme de signature du certificat, qui est l'algorithme `SHA-256` avec RSA ;
— **Issuer** → L'émetteur du certificat. Les détails affichés correspondent aux informations entrées pendant la génération de demande de signature du certificat ;
— **Validity** → Date jusqu'à laquelle le certificat sera valide et accepté par des applications utilisant SSL/TLS ;
— **Subject** → C'est l'identité du propriétaire du certificat. Comme il est auto-signé, le sujet est identique à l'émetteur ;
— **Public Key Algorithm : rsaEncryption** → L'algorithme utilisé pour chiffrer la clé publique ;
— **Public-Key : (2048 bit)** → La taille de l'algorithme ;
— **Modulus** → Partie publique de la clé RSA du certificat
— **Exponent** → Exposant utilisé dans le chiffrement RSA ;
— **X509v3 Subject Key Identifier** → Identifiant unique de la clé du sujet ;
— **X509v3 Authority Key Identifier** → Identifiant unique de la clé d'autorité. Identique à la clé du sujet car c'est une certificat auto-signé ;
— **CA :TRUE** → Indique que le certificat est un certificat d'autorité de certification :
— **Signature Algorithm : sha256WithRSAEncryption** → Indique que le certificat est chiffré et signé avec SHA-256 et RSA ;
— **Signature Value** → Signature permettant de vérifier l'authenticité du certificat.

2.3 Comparaison entre deux certificats



Certificat du site github.com

Élément	**Certificat auto-signé**	**Certificat github.com**
Algorithme de signature	SHA-256 avec RSA	SHA-256 avec ECDSA
Émetteur = Sujet ?	Oui	Non
Algorithme clé publique	rsaEncryption 2048 bit	Elliptic Curve - 256 bit
Algorithme de signature	SHA-256 et RSA	SHA-256 ECDSA
Certificat d'autorité ?	Oui	Non
Version	3	3

Comparaison entre deux certificats

3 Recommandations ANSSI sur l'utilisation du protocole HTTPS

1. TLS :
   - Utiliser les versions 1.2 ou 1.3 ;
   - Éviter les connexions HTTP non-sécurisées en les redirigeant vers un HTTPS ;
   - Même si le site ne traite pas d'informations sensibles, toujours appliquer des recommandations de sécurité liées au TLS.

2. Mettre en œuvre le HTTP Strict Transport Security (HSTS) :
   - Activer l'HSTS pour forcer l'utilisation du HTTPS ;
   - Assurer que l'accès HTTPS est pérenne, car l'HSTS rend impossible l'accès en HTTP une fois activé ;

3. Surveiller les Certificate Transparency (CT) logs :
   - Mettre en place un processus de surveillance des CT logs pour détecter et révoquer les certificats illégitimes ;
   - Utiliser ces registres cryptographiquement sécurisés pour vérifier la légitimité des certificats émis par les autorités de certification.

3.1 Explication de certaines notions

3.2 L'HTTP Strict Transport Security

L'HSTS est un mécanisme qui force l'utilisation du protocole HTTPS sur un site, en empêchant les navigateurs d'y accéder via un protocole HTTP non sécurisé.
Lorsqu'un navigateur accède à un site pour la première fois en HTTPS, le serveur envoi un en-tête HSTS dans sa réponse HTTP. Cet en-tête indique que toutes les connexions futures à ce site devront se faire en HTTPS, sinon le site ne sera pas accessible, ou le redirigera en HTTPS.

3.3 Certificate Transparency Logs

C'est un protocole dont le but est d'améliorer la sécurité et la transparence des certificats HTTPS, en mettant en place une journalisation qui produit des logs publiques des certificats SSl/LTS, délivrés par les autorités de certification.
Lorsqu'une autorité de certification délivre un certificat, elle ajoute ce dernier dans un CT log. Chaque certificat ajouté aux logs est enregistré de manière sécurisée. Pourquoi c'est utile ? Parce que les navigateurs exigent que les sites en HTTPS utilisent des certificats enregistrés dans les logs. Également, un site peut surveiller ces logs pour s'assurer qu'il n'y a pas de faux certificats.

4 Ressources utilisées

Sources

— https://www.technologygee.com/what-is-the-domain-name-system-dns-protocol/
— https://next.ink/28083/82009-le-new-york-times-et-twitter-victimes-hier-soir-dun-
— https://www.frameip.com/dns/#61-8211-fragilite
— https://www.ibm.com/fr-fr/topics/dns-protocol
— https://fr.wikipedia.org/wiki/Domain_Name_System
— https://www.nameshield.com/ressources/lexique/dns-domain-name-system/
— https://www.paloaltonetworks.com/cyberpedia/what-is-a-dns-attack