**D1**

Bonjour à tous,
Nous savons tous que le https sécurise nos communications web, mais est-ce suffisant pour garantir l'intégrité des données en transit ?

**D2**

Un bon point de départ, pour comprendre la pertinence de la signature HTTP et justifier la nécessité d'explorer des mesures supplémentaires à https, sera :

- Une courte présentation des vulnérabilités de https.
- Puis voir ce qu'est la signature HTTP, son fonctionnement,
- Comparaison et complémentarité avec https.
- Voir aussi, la valeur ajoutée et comment la signature HTTP augmente la sécurité au-delà de ce que https offre.
- Enfin, terminer avec un exemple d'implémentation démo.

**D3**

Lorsqu' un utilisateur essaie d'accéder à une application mobile ou site, via le web, un attaquant peut intercepter et modifier les communications pour rédiger le trafic https vers http. Cela permet à l'attaquant de lire, modifier les données transmises.
• L'attaquant rétrograde la connexion HTTPS à une connexion HTTP non sécurisée.
via par exemple la décompilation / recompilation d'un apk de l'app mobile ;
• Ou si un attaquant réussit et ajoute son certificat ou dans la liste blanche.
En 2019 une vulnérabilité a été découverte dans l'app WhatsApp Android qui permettait de contourner le mécanisme SSL Pininng.

Le SSL Pinning permet d'abandonner la connexion si un certificat invalide est détecté.
Permet à l'application client de s'assurer que le serveur avec lequel elle communique est bien celui auquel elle s'attend et empêche les attaques man-in-the-middle: qui pourrait présenter un certificat frauduleux.
Son fonctionnement en quelques mots:
• le client (comme l'App Mobile) conserve une copie du certificat public du serveur ou un hash empreinte du certificat.
• Lors de l'établissement de la connexion SSL/TLS, le client compare le certificat présenté par le serveur avec celui qu'il a en mémoire.

mTLS
Le contrôle d'authentification mutuelle fait référence non seulement au client qui valide le certificat du serveur, mais aussi au serveur qui valide le certificat du client. Le serveur interrompt la connexion si le certificat du serveur ou du client n'est pas valide.
• Toutefois, le serveur doit avoir préalablement fourni à chaque client un certificat unique.

proxy SSL
Le processus de proxy SSL consiste à intercepter le trafic SSL/TLS, à le déchiffrer, puis à le chiffrer à nouveau avec un nouveau certificat avant de le transférer vers la destination prévue. Cela permet à l'outil d'inspecter le trafic à la recherche d'activités malveillantes et de le bloquer avant qu'il ne puisse causer des dommages. Cela permet également à l'outil de surveiller le trafic pour vérifier la conformité aux politiques de sécurité et d'identifier et de bloquer toute tentative de contournement des contrôles de sécurité

Q: POODLE
R: vulnérabilité découverte en 2014 par des chercheurs de google. Sur ssl 3.0

Q: Logiam Diffie-Hellman
R: liée à l'utilisation de suites de chiffrements obsolètes et faibles.

Q: Android Hooking
R: une technique pour intercepter, modifier ou remplacer des appels de méthodes et des comportements dans les app à l'exécution. Cette technique est souvent employée à des fin de développement, debug, mais peut être également être utilisée à des fin malveillantes

**D4**

Quels avantages en ajoutant la signature http à une archi déjà sécurisée par HTTPS ?
Contrairement à HTTPS qui assure la confidentialité des données, la signature assure l'authenticité, l'intégrité et la non-réputation des messages échangés.

• Confidentialité: signifie que les informations sont protégées contre toute divulgation non autorisée.
• Authenticité: assure que l'identité des parties impliquées dans une communication est confirmée, empêchant ainsi toute usurpation d'identité.
• Intégrité: cela implique que les informations reçues sont exactes et complètes telles qu'elles ont été envoyées par la source d'origine.
• Non-réputation: signifie qu'une partie ne peut pas nier avoir envoyé un message ou réalisé une transaction. Car il existe une preuve irréfutable de son implication.

**D5**

Le principe de fonctionnement de la signature HTTP est simple:

• Une signature cryptographique est ajoutée à chaque requête HTTP, souvent dans un en-tête HTTP.

• Cette signature est générée en utilisant une clé privée et peut être vérifiée par un serveur (ou la destination) à l'aide de la clé publique correspondante.

• Si la vérification est en erreur, il faut garder des logs de d'alerte sécurité et retourner un code d'erreur approprié.

**D6**

La signature HTTP est couramment utilisée pour sécuriser les APIs, les services web, et les environnements où il est crucial de s'assurer que les requêtes et les réponses n'ont pas été altérées.

Plusieurs risques sont associés à la non-utilisation d'une signature: corruption du contenu, corruption des informations de sécurité et des informations techniques.

Quelques avantages:

• Assurer que la requête ou la réponse est bien celle émise par le client ou le serveur légitime par la vérification de l'authenticité des requêtes et des réponses.

• Protéger contre les attaques par rejeu grâce à l'utilisation de nonces ou de timestamps.

Q: nonce
R: Number used Once, une valeur arbitraire qui est utilisée pour protéger contre le rejeu d'une même requête.

• Détecter toute tentative d'interception intermédiaire, d'attaque par altération et injection non autorisée des données.

**D7**

• le https, se concentrent sur la sécurisation du canal de communication entre le client et le serveur, assurant que la connexion est établie non pas avec un malveillant.

• Tandis que la signature HTTP, se concentre sur la sécurisation des données transmises via HTTP, en garantissant que chaque requête ou réponse est authentique et n'a pas été altérée en transit.

**D8**

- Facteurs à considérer pour une bonne implémentation : le niveau de sécurité et quels actifs à sécuriser, les performances, les exigences réglementaires, la comptabilité technologique…
- Asymétrique : pas besoin de partager la key secrète; plus sécurisé; un peu plus lent en calcul;
- Symétrique : rapide (moins de ressources); mais nécessité de partager la key secrète entre les parties.
- Pour assurer une intégrité, les mécanismes de signatures doivent a minima utiliser SHA256 et RSA (ig. HMAC SHA256; RSA SHA256)
- Une bonne pratique pour la gestion des clés de sécurité, est l'exposition d'un endpoint JWKS de clé publique ;

Q:JWKS
R: JSON Web Key Set, est un format de données utilisé pour représenter un ensemble de clés cryptographiques en utilisant JSON. Le endpoint JWKS c'est une URL ou le serveur expose son set de clés.

- Mettre en place un SDK commun (pour la génération et vérification de la signature par technologie).
- Fournir une gestion des erreurs appropriée,
- Inclusion des informations horodatées comme un nonce et une fois qu'un nonce a été vérifié, il doit être marqué comme utilisé pour éviter qu'il ne soit réutilisé dans une autre requête.

**D9**

La gestion efficace des certificats et clés de sécurité est cruciale.

• Mettre en place des procédures pour la rotation régulière des certificats afin de minimiser les risques de compromis,

• Les certificats doivent être distribués de manière sécurisée aux tiers: cela peut inclure des procédures de gestion des certificats pour l'émission, la révocation et le renouvellement des certificats.

• Les CA publient des listes de certificats révoqués (CRL), et offrent des services de validation en temps réel via le protocole OCSP (avec une liste blanche).

Q: CRL (Certificate Revocation List)
R: listes de certificats révoqués (par numéro de série des certificats), l'inconvénient de CRL est la : consultation de la liste complète, qui peut entraîner une perte sur les perf.

Q: OCSP (Online Certificate Status Protocol)
R: demander l'etat (valide ou révoqué) d'un certif auprès d'un serveur OCSP. Plus efficace que CRL, une réponse instantanée et sans recours au besoin de télécharger de longue liste de révocations.

Q: PKI
R: une infra PKI, public key infra est un cadre de technologies de politique et procédures nécessaire à la création, à la gestion, à la distribution, à l'utilisation, au stockage et révocation de certificats numériques. Composé principalement de CA (autorité de certificat CA et autorité d'enregistrement RA).

Q: CA/RA
R: une CA est une entité de confiance qui émit et gère les certificats, elle vérifie l'identité des entités avant de leur délivrer un certificat. Une RA, est responsable de l'enregistrement des utilisateurs et vérification de leurs identités avant que CA n'émette un certificat.

• Les fournisseurs de confiance QTSP
(Qualified Trust Service Provider)
Offrent des niveaux supplémentaires de garantie et de conformité réglementaires comme eIDAS. Cela les rend particulièrement adaptés pour des usages nécessitant une reconnaissance juridique et une conformité a titre d'exemple à l'échelle européenne.

**D10**

Dans cet exemple : l'objectif est de mettre en place un mécanisme demo de signature HTTP, qui permet de vérifier l'intégrité d'une requête initiée par l'App Mobile et d'une réponse retournée par le Backend.
Dans la pratique, l'App Mobile doit pouvoir signer une requête à destination du Backend. De même le Backend doit pouvoir vérifier l'authenticité et l'intégrité de la requête et signer la réponse associée.
Finalement, l'App Mobile vérifie à son tour l'authenticité de la réponse serveur Backend.

**D11**

- keyId; identifie la clé utilisée pour signer la demande. La clé secrète ne doit jamais être transmise dans une requête.
- algorithm: spécifie l'algorithme de signature utilisé, exemple "hmac-sha256" et "rsa-sha256" ...etc
- headers: éléments signés, une liste ordonnée des noms d'en-tete de demande HTTP qui doivent être inclus dans la signature, exemple "date", "nonce" et "digest"....etc. le digest, content-type et content-length utilisés pour les requêtes POST/PUT/PATCH.
- Le (request-target) est un pseudo-header, une combinaison de la méthode de requête HTTP (HTTP Method) et l'URI de la requête HTTP (HTTP Request URI path), utilisée seulement pour les requêtes.
- signature: la signature générée à partir des éléments signés en utilisant la clé privée, dans cet exemple signature="Base64 Encode(hmac-sha256(payload))”

**D12**

L'utilisation de la signature http, comme solution complémentaire à HTTPS renforce la sécurité en garantissant l'intégrité et l'authenticité des requêtes et des réponses, même si HTTPS protège déjà la confidentialité des données échangées.
La signature HTTP permet de détecter et de prévenir les manipulations de contenu. Ajoutant une couche de sécurité supplémentaire pour les communications web.

Q: TLS
R: transport layer security, est un protocol de sécurité, il succède le SSL (Secure Socket Layer).

Q:X.509
R: est une norme définit le format des certificats de clé pour l'auth et le chiffrement.