



# UE L318 - Sécurité des applications Web

*Introduction à la sécurité du web*

Semaine 1

**Ilaria Zappatore**

# Pourquoi sécuriser les web apps ?

Pourquoi la sécurité du web ?

*Le web :*

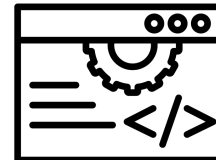
1. Permet **échange des données sensibles**



2. Est utilisé par les **gouvernements**



3. « **Facile** » de créer des pages/appli web



# Pourquoi sécuriser les web apps ?

Il est important « d'éduquer » les développeurs pour qu'ils prennent en **compte les bonnes normes** afin de rendre les sites moins vulnérables aux attaques.



<https://cyber.gouv.fr>

<https://cyber.gouv.fr/bonnes-pratiques-protegez-vous>

La photo provient du site <https://pixabay.com>

# Pourquoi sécuriser les web apps ?

Les sites et les applications web sont très exposés aux utilisateurs et aux utilisateurs malveillants

*Menaces les plus répandues :*

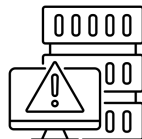
**1. Compromission des ressources**



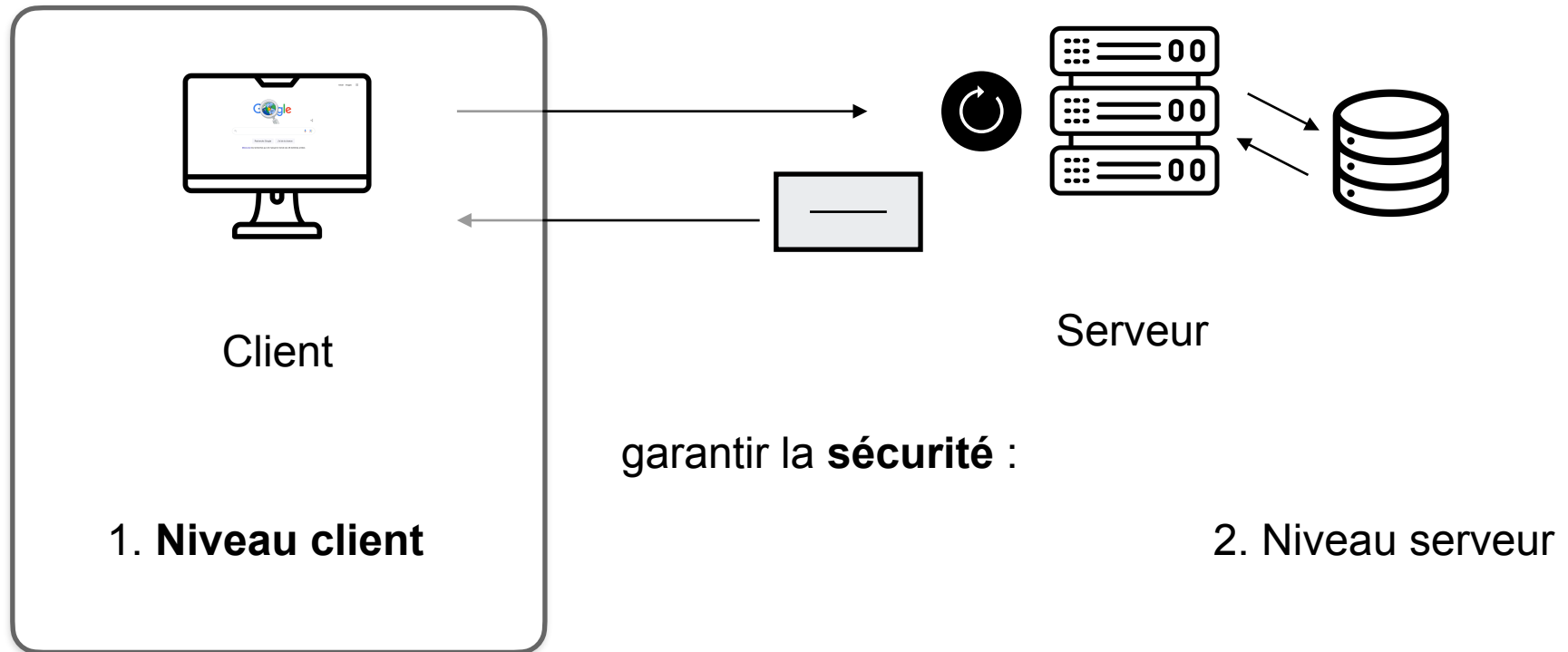
**2. Vol de données**



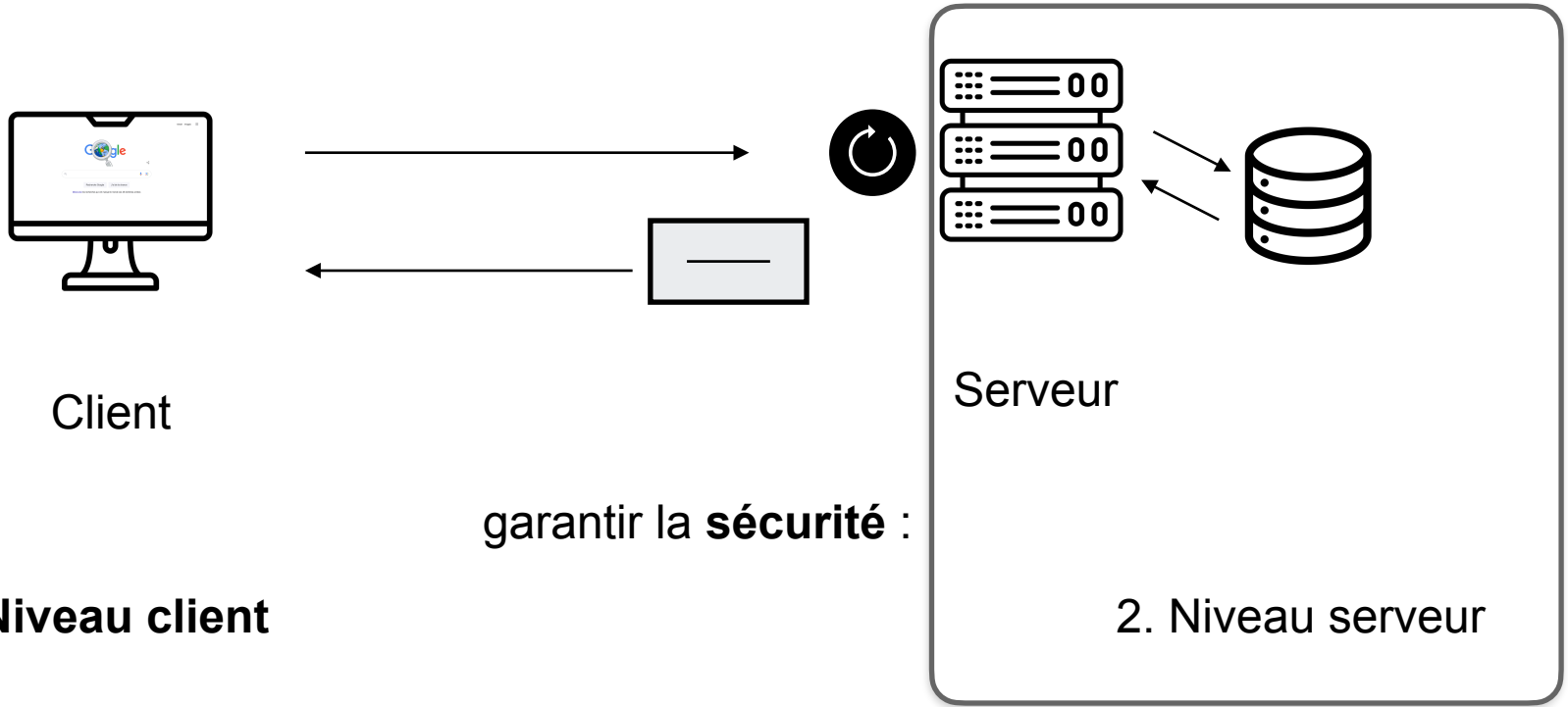
**3. Déni de service**



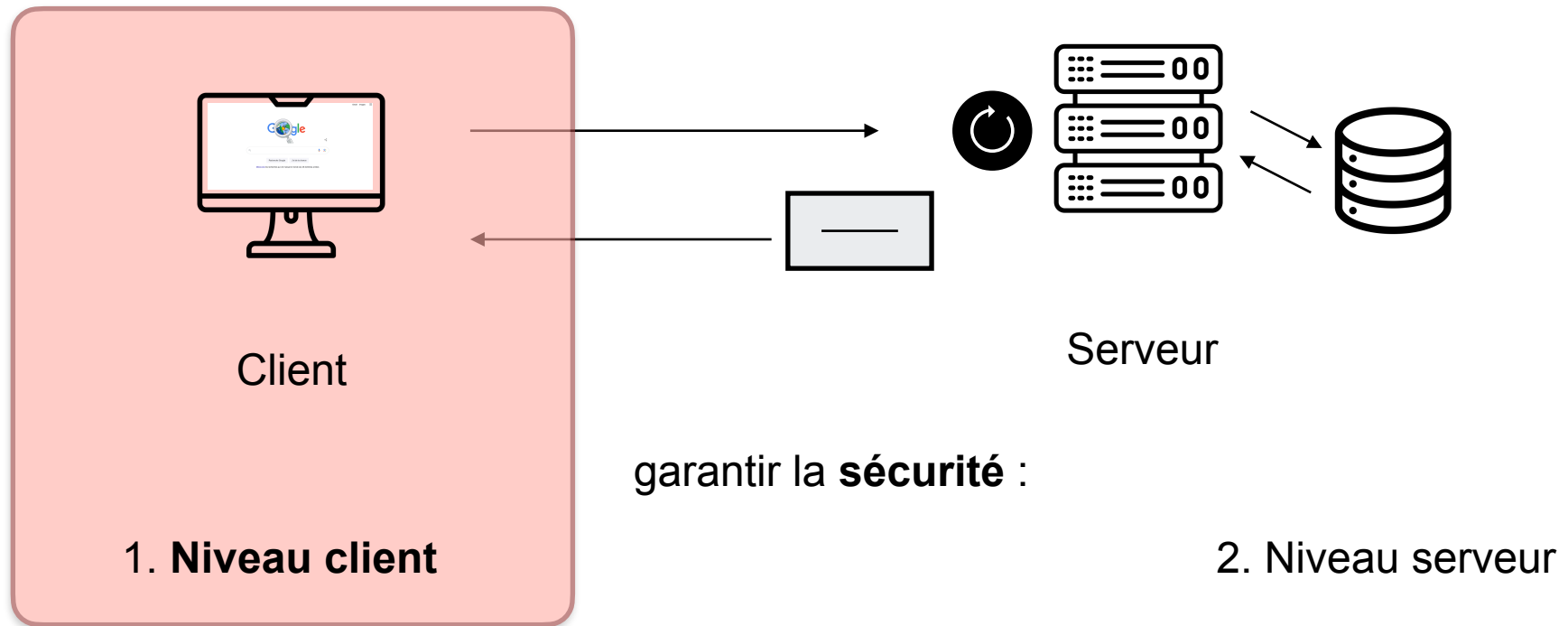
# Modèle du web



# Modèle du web



# Modèle du web



# Panorama des attaques

Voici quelques attaques récurrentes :

1. **XSS** (*Cross-Site Scripting*)
2. **CSRF** (*Cross-Site Request Forgery*)
3. **SSRF** (*Server-Site Request Forgery*)
4. **SQLi** (*SQL injection*)
5. **LFI/RFI** (*Local/Remote File Inclusion*)
6. **XXE** (*XML External Entity*)

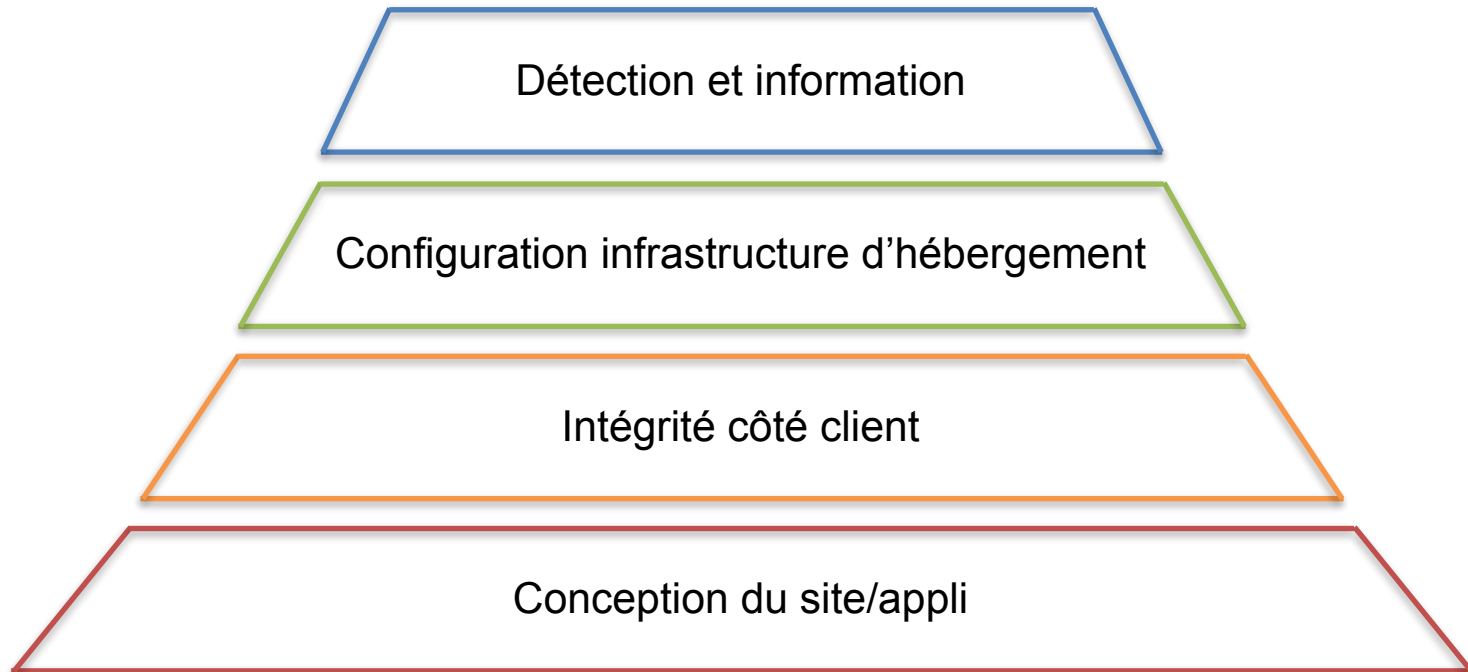


# Panorama des attaques

Voici quelques attaques récurrentes :

1. **XSS** (*Cross-Site Scripting*)
2. **CSRF** (*Cross-Site Request Forgery*)
3. **SSRF** (*Server-Site Request Forgery*)
4. **SQLi** (*SQL injection*)
5. **LFI/RFI** (*Local/Remote File Inclusion*)
6. **XXE** (*XML External Entity*)

# Règles d'hygiène



# URL, IP, DNS

URL (*Uniform Resource Locator*), adresse web

<https://www.google.com/>



Protocole de communication HTTP(S)



Nom du domaine

IP (*Internet Protocol*) : numéro unique assigné à chaque appareil connecté dans un réseau.

# URL, IP, DNS

URL (*Uniform Ressource Locator*), adresse web

<https://www.google.com/>



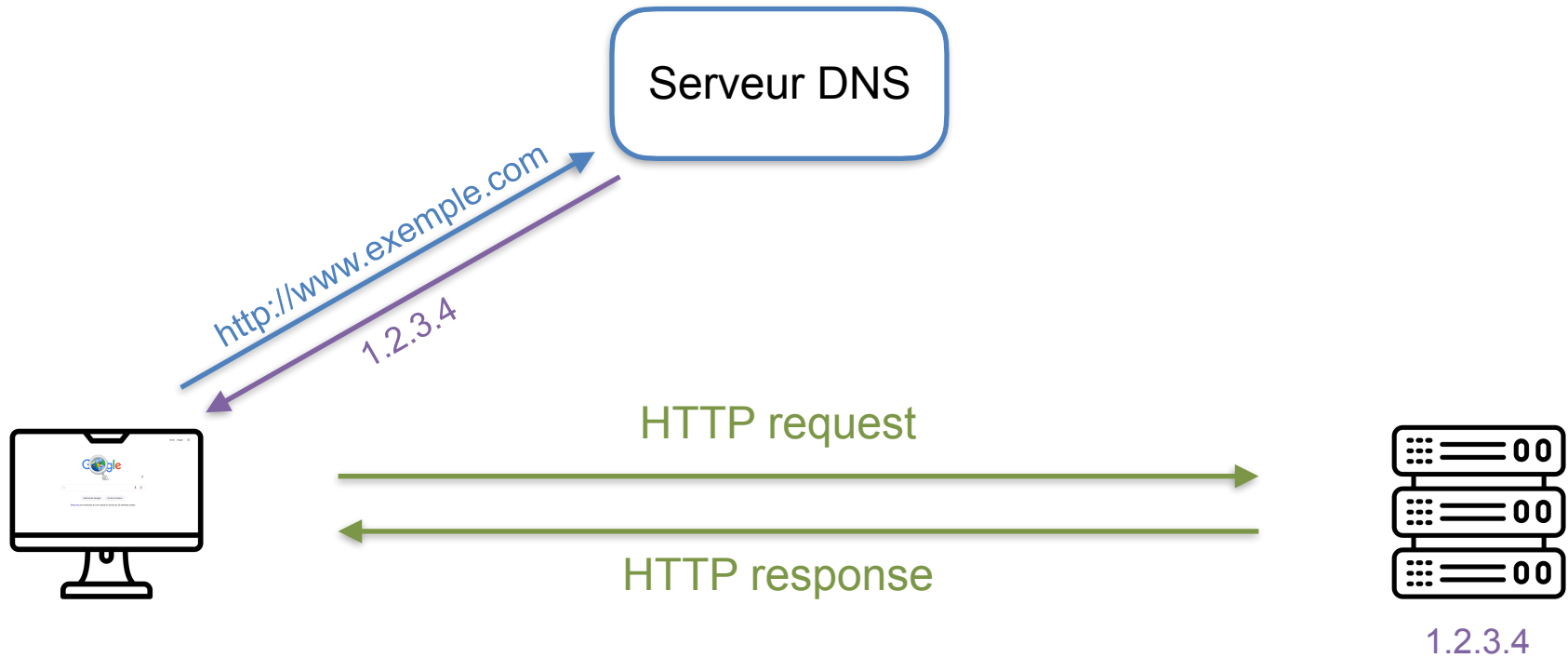
Protocole de communication HTTP(S)

Nom du domaine

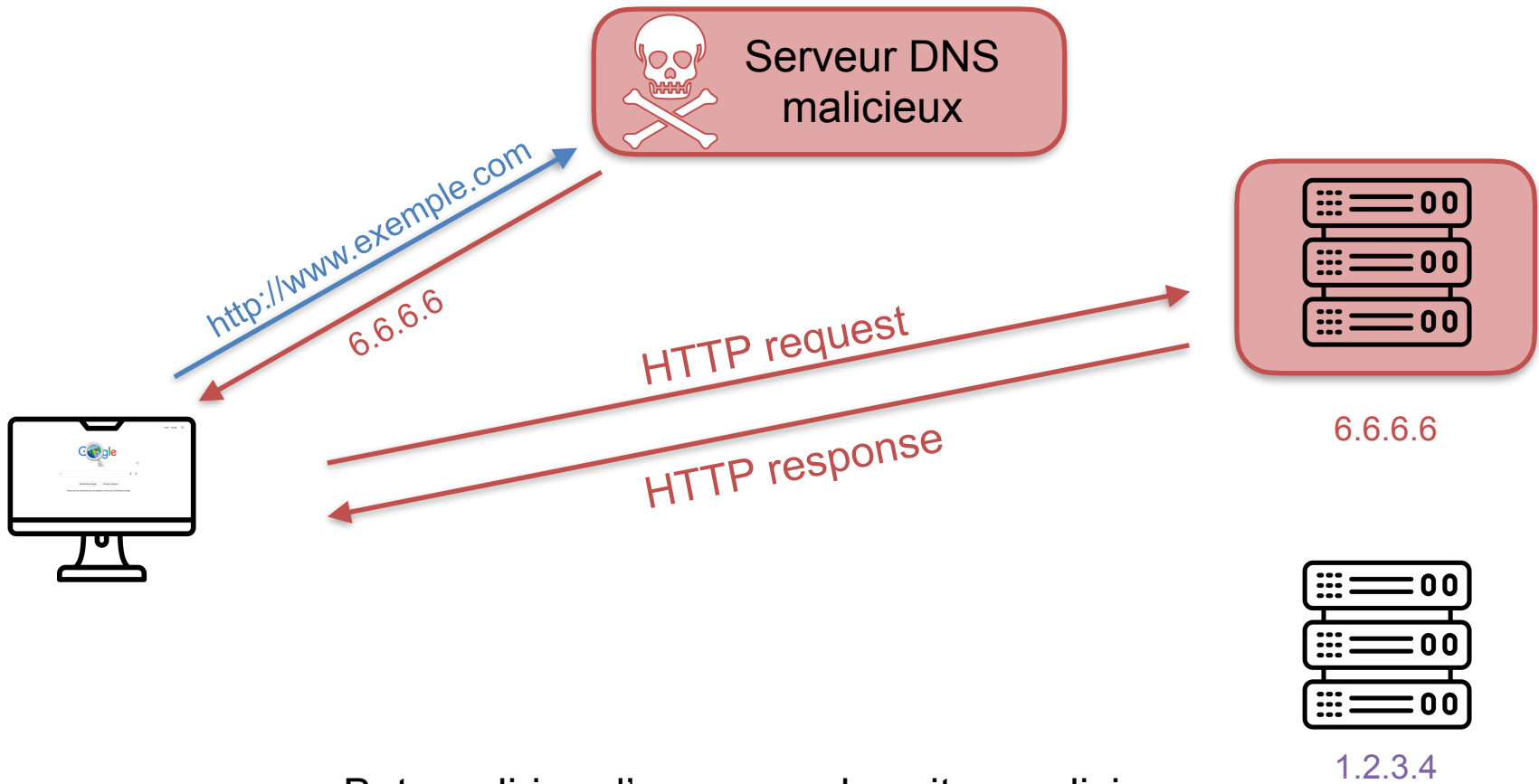
Serveur DNS (*Domain name system*)  
convertit URL en IP

IP (*Internet Protocol*) : numéro unique assigné à chaque appareil connecté dans un réseau.

# DNS







# DNS hijacking



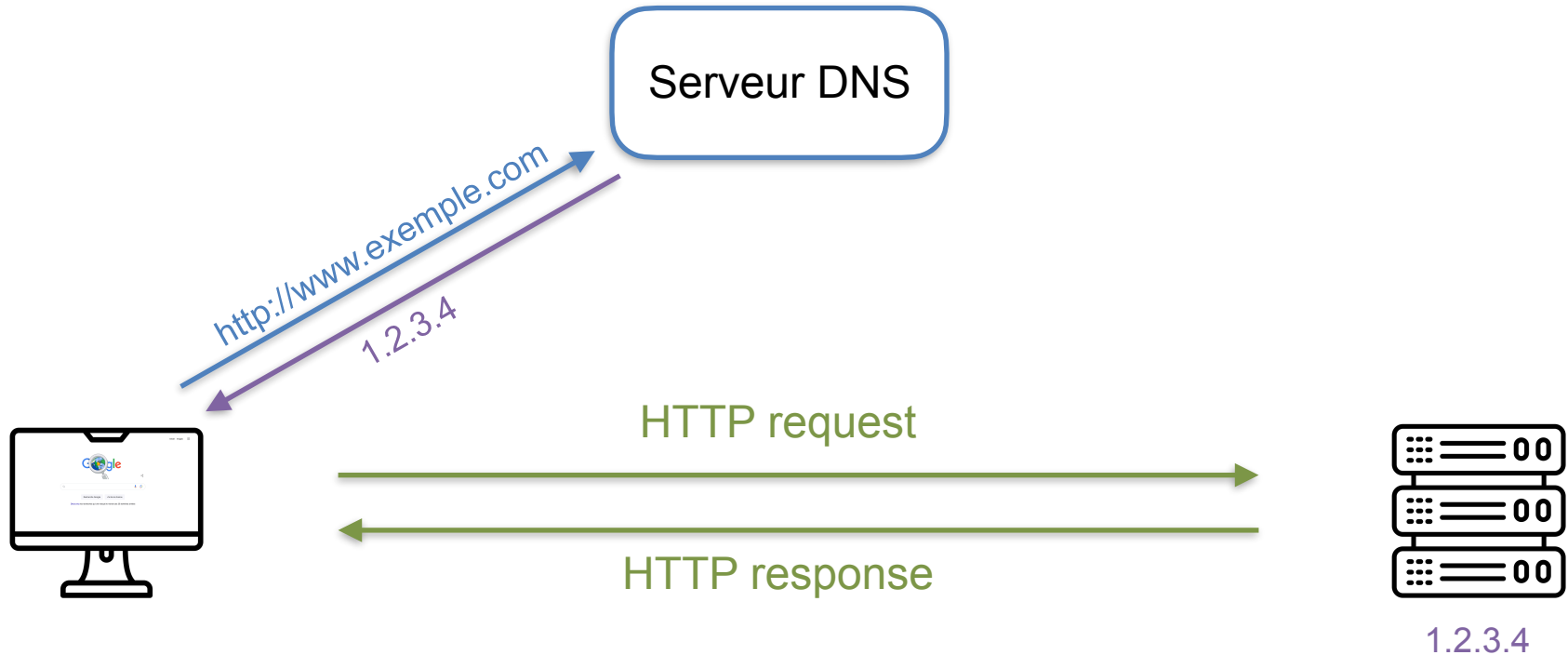
But : rediriger l'utilisateur vers des sites malicieux  
*Pharming, phishing*

# Travail de la semaine 1

## Partie 1

-  *Détaillez le protocole DNS.*
-  *Décrivez les différents types de détournement DNS. Illustrez chaque type par un exemple pratique.*
-  *Proposez des solutions aux différents types d'attaques.*
-  *Trouvez un exemple de détournement DNS qui s'est réellement produit et expliquez-le.*

# Protocole HTTP



Le protocole HTTP (*Hypertext transfer protocol*) : principal protocole de communication utilisé pour accéder au World Wide Web et est utilisé par toutes les applications web actuelles.



# Protocole HTTP

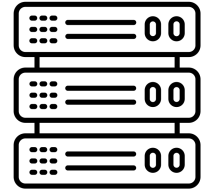


HTTP requête

HTTP request



HTTP response



HTTP request

```
GET [/http://www.exemple.com/] HTTP/1.1
```

```
Accept : text/html
```

```
User-Agent: Mozilla/5.0
```

# Protocole HTTP

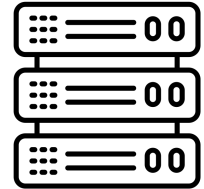


HTTP requête

HTTP request



HTTP response



méthode

URL

version du protocole

GET [/http://www.exemple.com/] HTTP/1.1

Accept : text/html

User-Agent: Mozilla/5.0

headers

# Protocole HTTP

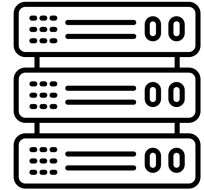


HTTP requête

HTTP request



HTTP response



HTTP response

HTTP/1.1 200 OK

Content-Type: text/html

Date: Tue, 28 Feb 2024 14:37:12 GMT

<!DOCTYPE html...

# Protocole HTTP

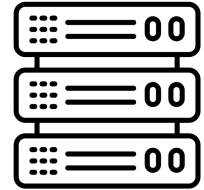


HTTP requête

HTTP request



HTTP response



HTTP/1.1 200 OK → code de réponse

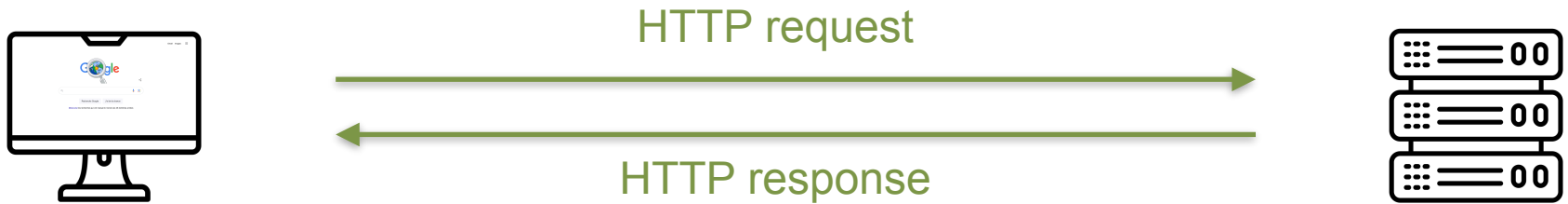
Content-Type: text/html

Date: Tue, 28 Feb 2024 14:37:12 GMT

headers

<!DOCTYPE html... → code html de la page

# Protocole HTTP



HTTP requête

Pour plus de détails sur les entêtes, les méthodes et les codes des messages

<https://www.commentcamarche.net/informatique/technologies/26181-protocole-http-principe-de-fonctionnement/>

# HTTP est...

1. **Simple** : facile à lire et comprendre
2. **Extensible** : on peut rajouter plus d'infos à travers les entêtes
3. **Stateless** : il n'y a pas de lien entre deux requêtes exécutées successivement sur la même connexion.

important d'ajouter les **cookies**



Données envoyées par le serveur au navigateur,  
pour tenir trace de la session

# Sécuriser HTTP : HTTPS

HTTP request

GET [/http://www.exemple.com/] HTTP/1.1

Accept : text/html

User-Agent: Mozilla/5.0



La requête est **chiffrée**

t8Fw6T8UV81pQfyhDkhebbz7+oiwldr1j2gHBB3L3RFTRsQCpaSnSBZ78Vme+DpDVJpVZdZUZHpzbbcqmSW1  
+3xXGsERHg9YDmpYk0VVDiRvw1H5miNieJeJ/FNUjgH0BmVRWII6+T4MnDwmCMZUI/  
orxP3HGwYCSiVyzS3MpmSe4iaWKC0HQ==

# Sécuriser HTTP : HTTPS

HTTPS = HTTP *Sécurisé*



Protocole SSL/TLS

Le protocole HTTPS assure :

1. **L'intégrité** : empêche aux intrus de falsifier les communications client/serveur
2. **La confidentialité** : seul le destinataire peut lire les infos échangés
3. **Authenticité** : de la communication avec le serveur

grâce à la CRYPTOGRAPHIE



# Cryptographie symétrique

discipline qui étudie comment protéger la transmission des données



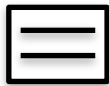
en assurant

*Confidentialité, Authenticité et Intégrité*

**Cryptographie à clé secrète (symétrique)**



clé secrète



# Cryptographie symétrique

discipline qui étudie comment protéger la transmission des données



en assurant

*Confidentialité, Authenticité et Intégrité*

## Cryptographie à clé secrète (symétrique)



# Cryptographie symétrique

discipline qui étudie comment protéger la transmission des données



en assurant

*Confidentialité, Authenticité et Intégrité*

**Cryptographie à clé secrète (symétrique)**



# Cryptographie symétrique

discipline qui étudie comment protéger la transmission des données



en assurant

*Confidentialité, Authenticité et Intégrité*

**Cryptographie à clé secrète (symétrique)**



Déchiffrement



clé secrète

# Cryptographie asymétrique

discipline qui étudie comment protéger la transmission des données



en assurant

*Confidentialité, Authenticité et Intégrité*

**Cryptographie à clé publique (asymétrique)**



clé publique



clé secrète

# Cryptographie asymétrique

discipline qui étudie comment protéger la transmission des données



en assurant

*Confidentialité, Authenticité et Intégrité*

## Cryptographie à clé publique (asymétrique)



# Cryptographie asymétrique

discipline qui étudie comment protéger la transmission des données



en assurant

*Confidentialité, Authenticité et Intégrité*

**Cryptographie à clé publique (asymétrique)**



# Cryptographie asymétrique

discipline qui étudie comment protéger la transmission des données



en assurant

*Confidentialité, Authenticité et Intégrité*

**Cryptographie à clé publique (asymétrique)**



*Déchiffrement*



clé secrète



# Crypto symétrique et asymétrique

discipline qui étudie comment protéger la transmission des données



en assurant

*Confidentialité, Authenticité et Intégrité*

Parfois il est utile **de combiner**  
**crypto symétrique et crypto asymétrique**

# Crypto symétrique et asymétrique

Parfois il est utile **de combiner**  
**crypto symétrique et crypto asymétrique**

**Échange des clés**  
**Crypto asymétrique**



clé secrète



clé publique



clé secrète

# Crypto symétrique et asymétrique

Parfois il est utile **de combiner**  
**crypto symétrique et crypto asymétrique**

**Échange des clés**  
**Crypto asymétrique**



# Crypto symétrique et asymétrique

Parfois il est utile **de combiner**  
**crypto symétrique et crypto asymétrique**

**Échange des clés**  
**Crypto asymétrique**



clé publique

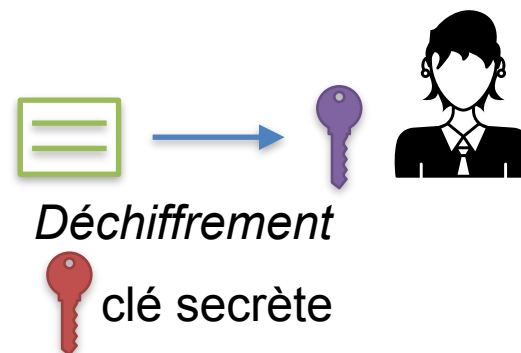


clé secrète

# Crypto symétrique et asymétrique

Parfois il est utile **de combiner**  
**crypto symétrique et crypto asymétrique**

**Échange des clés**  
**Crypto asymétrique**



# Crypto symétrique et asymétrique

Parfois il est utile **de combiner**  
**crypto symétrique et crypto asymétrique**

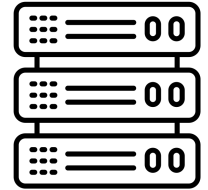
**Crypto symétrique**  
*Échanger des données*



# TLS/SSL simplifié



Établi une connexion sécurisée  
→  
méthodes de chiffrement



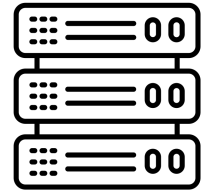
# TLS/SSL



certificat électronique  
délivré par une autorité



clé publique

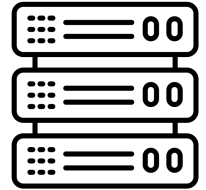




# TLS/SSL



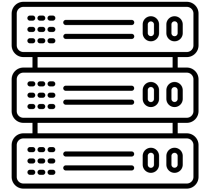
Vérifie l'identité du serveur



# TLS/SSL



Vérifie l'identité du serveur



Utilise la clé publique



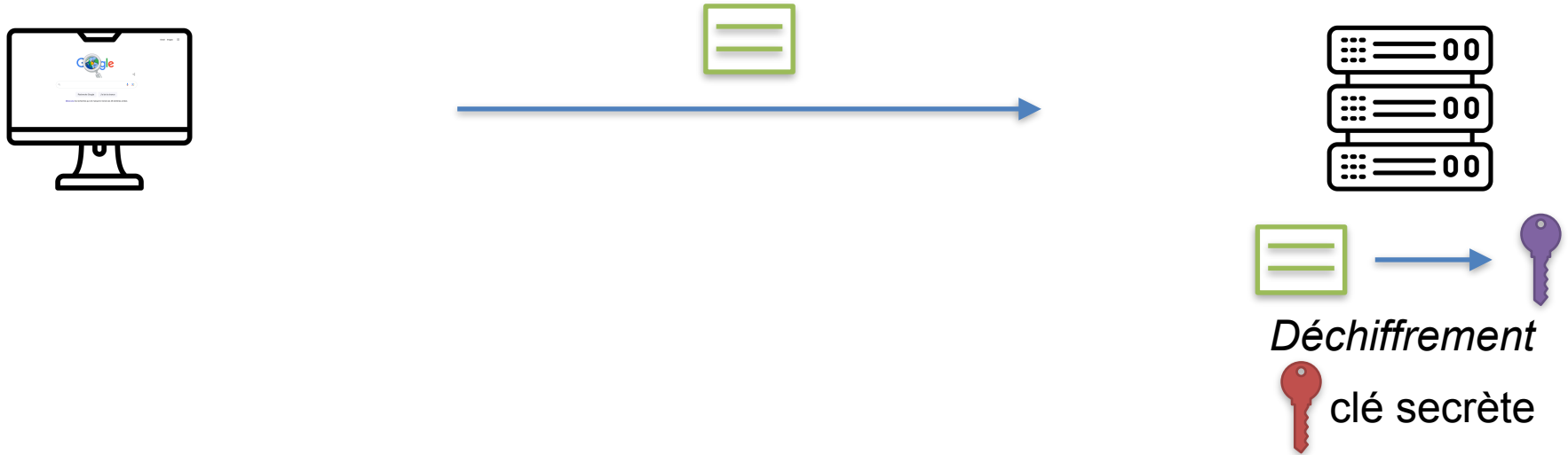
pour chiffrer une **clé de session** (secrète)



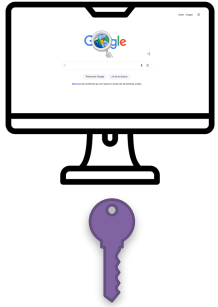
clé secrète



# TLS/SSL

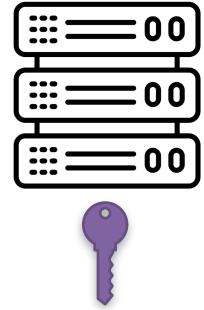


# TLS/SSL



Connexion établie

---



Communication HTTP chiffrée avec la clé de session  
Crypto symétrique

Connexion terminée - clé de session révoquée

# Travail de la semaine 1


## Partie 2

Téléchargez et installez OpenSSL sur votre machine.  
Faites des captures d'écran du terminal avec le résultat chaque fois que vous testez une requête.

- 📌 *Générez un certificat auto-signé à l'aide d'OpenSSL.*
- 📌 *Affichez les détails du certificat et expliquer leur signification.*
- 📌 *Ouvrez ensuite votre navigateur et accédez au certificat d'un site web de votre choix.*
- 📌 *Comparez les deux certificats obtenus.*

# Travail de la semaine 1

## Partie 3

 Expliquez les recommandations de l'ANSSI concernant l'utilisation du protocole HTTPS.

*Attention : tous et toutes les termes/notions qui n'ont pas été abordés dans le cadre de la CV doivent être expliqués.*

[https://cyber.gouv.fr/sites/default/files/2013/05/anssi-guide-recommandations\\_mise\\_en\\_oeuvre\\_site\\_web\\_maitriser\\_standards\\_securite\\_cote\\_navigateur-v2.0.pdf](https://cyber.gouv.fr/sites/default/files/2013/05/anssi-guide-recommandations_mise_en_oeuvre_site_web_maitriser_standards_securite_cote_navigateur-v2.0.pdf)