Sécurité Applicative

AppSec Web
Lu. 15 Oct. 2018 - PHELIZOT Yvan

Programme

- Rappels sur HTTP/HTML/Javascript
- Top 10 OWASP
  - OWASP WebGoat

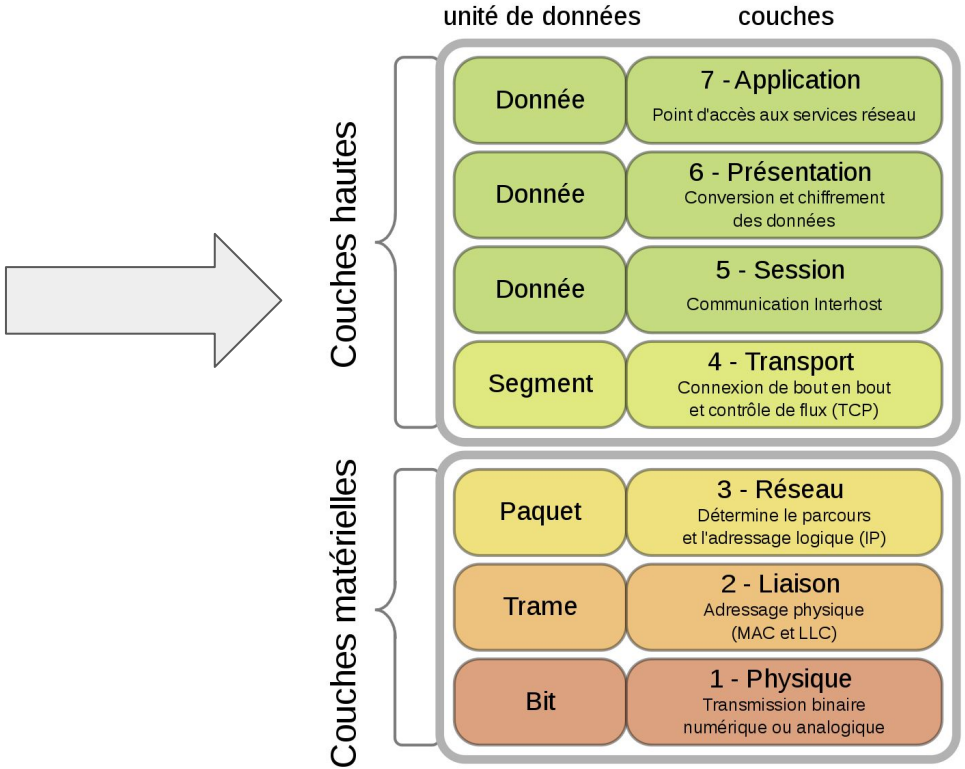
Rappels

HTTP

- HyperText Transfer Protocol
- Protocole sans état : requête auto-suffisante
- Crée en 1990
- Version actuelle : HTTP 1.1
- Version future : HTTP/2

Modèle OSI

HTTP



Source: https://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI

URL

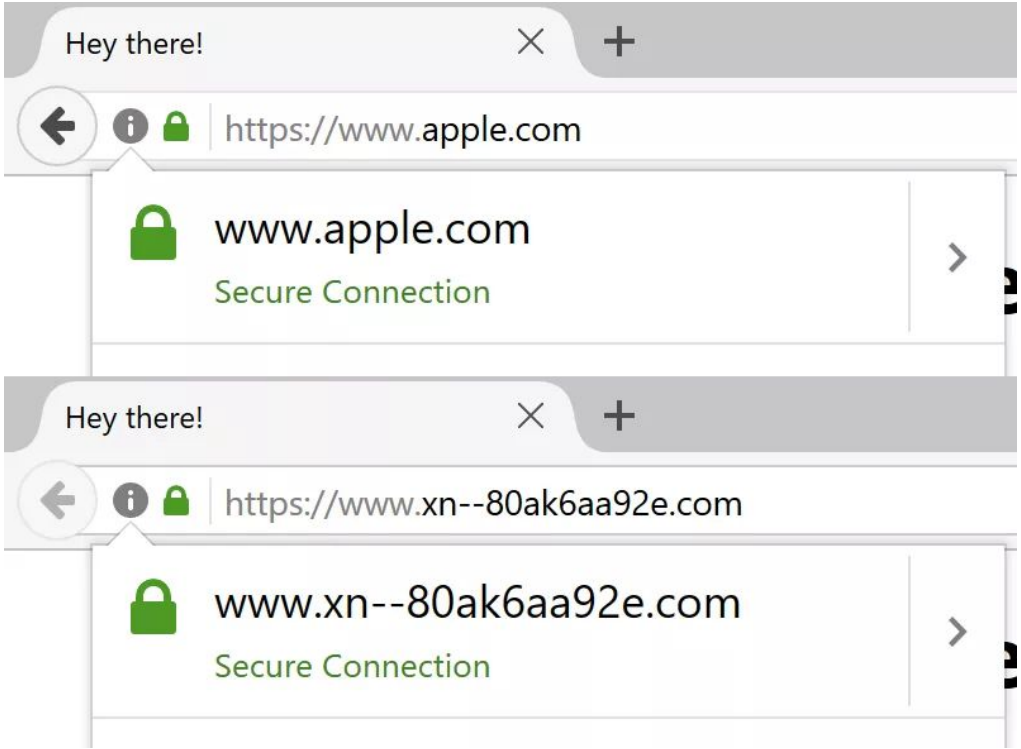
- Uniform Resource Locator
- Identifiant pour accéder à une ressource

http://login:pwd@www.here.com:8888/chemin/d/acc%C3%A8s.php?q=req&q2=req2#s

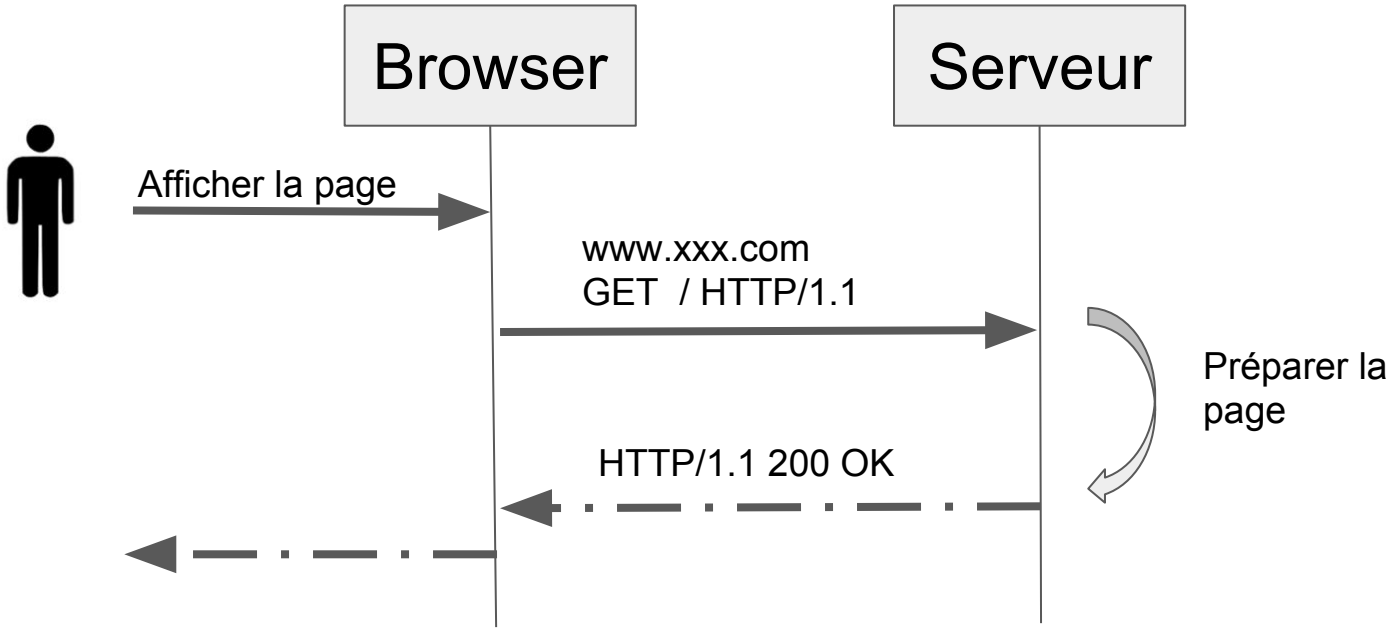
- http ⇒ protocole
- login:pwd ⇒ login & password
- www.here.com ⇒ sous-domaine
- 8888 ⇒ port
- Échappement des caractères : %C3
- q=req : paramètres

Homograph attack

- URL phising attack
- Exemple : https://www.xn--80ak6aa92e.com/ sous Firefox



Requêtes/Réponses



HTTP Request/Démo

curl -v http://www.lemonde.fr

> GET / HTTP/1.1

> Host: www.lemonde.fr

> User-Agent: curl/7.55.1

> Accept: */*

HTTP Response/Démo

< HTTP/1.1 301 Moved Permanently
< Location: https://www.lemonde.fr/
< Content-Length: 0
< Accept-Ranges: bytes
< Date: Sun, 30 Sep 2018 07:08:15 GMT
< Via: 1.1 varnish
< Age: 27
< Connection: keep-alive
< X-Served-By: cache-cdg20729-CDG
< Set-Cookie: prog-deploy=23; expires=Fri, 29 Mar 2019 07:08:15 GMT; path=/; domain=.lemonde.fr;
< Set-Cookie: prog-deploy2=78; expires=Fri, 29 Mar 2019 07:08:15 GMT; path=/; domain=.lemonde.fr;

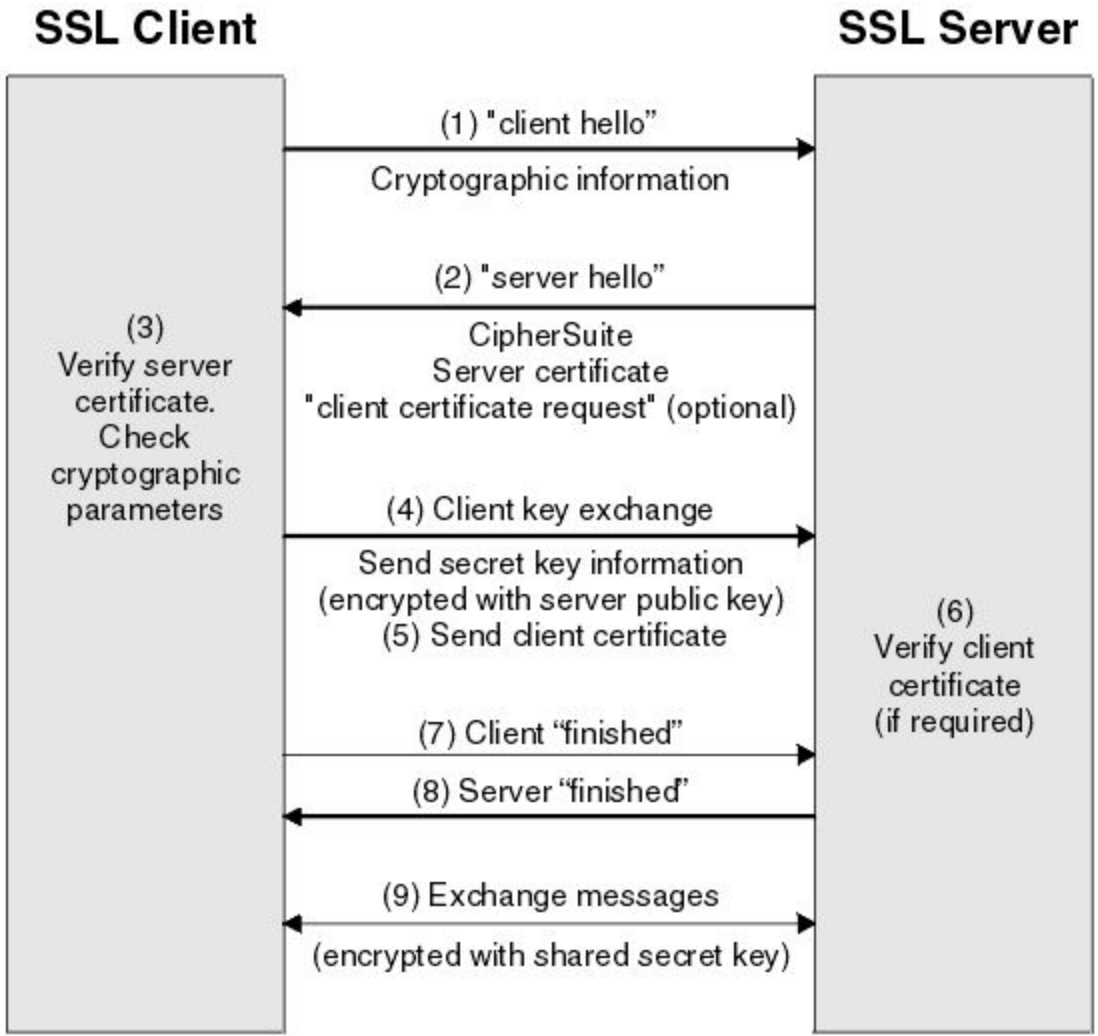
HTTPS

- HTTP for Secure communication
- Chiffrer les communications sur HTTP
- Histoire
  - 1994: SSL1.0
  - 1998 : TLS.10
  - SSL 3.0 (Deprecated)
  - TLS : 1.3 (Current)

HTTPS

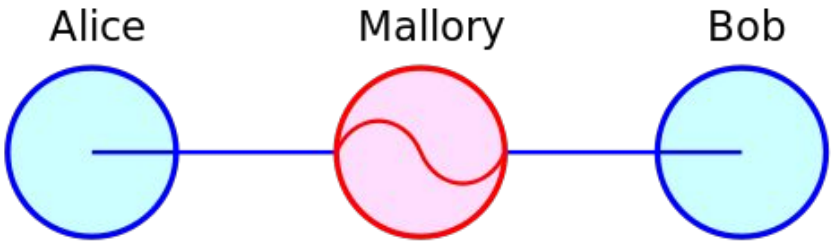
- Chiffrement & authentification du serveur ⇒ Certificat
- Chaine de validation
- Niveau de validation (Extended Validation Cert)
- Let’s Encrypt
- Google : importance

Regardons un certificat

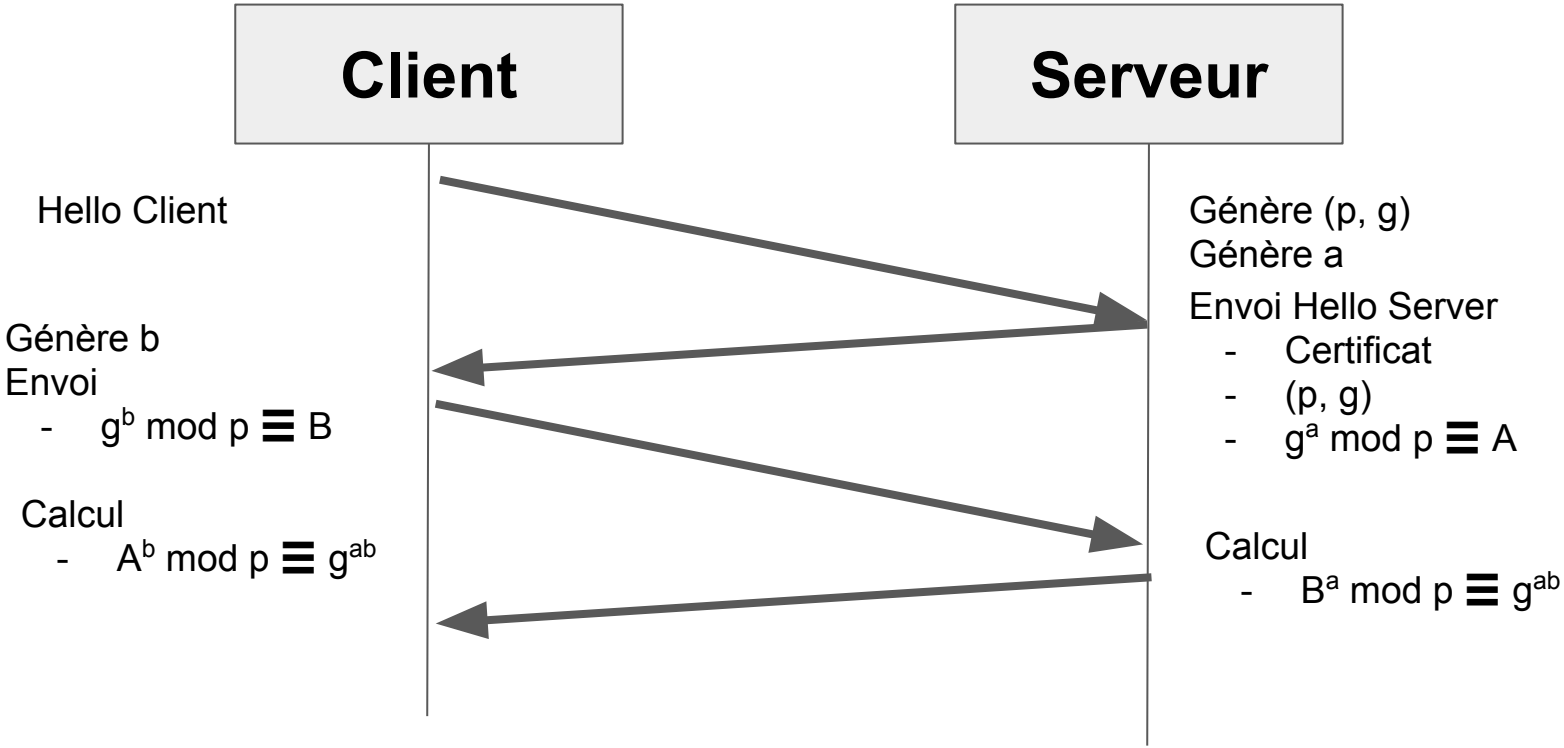

SSL Client
SSL Server
(1) "client hello"
Cryptographic information
(2) "server hello"
CipherSuite
Server certificate
"client certificate request" (optional)
(3)
Verify server
certificate.
Check
cryptographic
parameters
(4) Client key exchange
Send secret key information
(encrypted with server public key)
(5) Send client certificate
(6)
Verify client
certificate
(if required)
(7) Client "finished"
(8) Server "finished"
(9) Exchange messages
(encrypted with shared secret key)

Imaginons...

- Interception (ex: Man in the Middle) d'une communication HTTPS
- Deux ans plus tard, le certificat est volé
- Comment protéger la communication?



Perfect Forward Secrecy



Wireshark

HTTPS

Avantages

- Confiance
- Protection de la communication (privacy)

Désavantages

- Confiance
- Pas de cache
- Performances

Attaques HTTPS

- Protocol negotiation
- Brute-force
- Insecure conception
  - SSL 1.0
  - by design
    https://en.wikipedia.org/wiki/Dual_EC_DRBG
- Vol de certificat

Protection

- Redirect to HTTPS
- Durée de vie (Let’s encrypt)
- Révocation
- HSTS
- Certificate Transparency
- Public-key pinning (HPKP)

Session

- HTTP is stateless
- Solution : Session
- Permet à un utilisateur d’avoir accès à l’application sans devoir rentrer son mot de passe à chaque requête

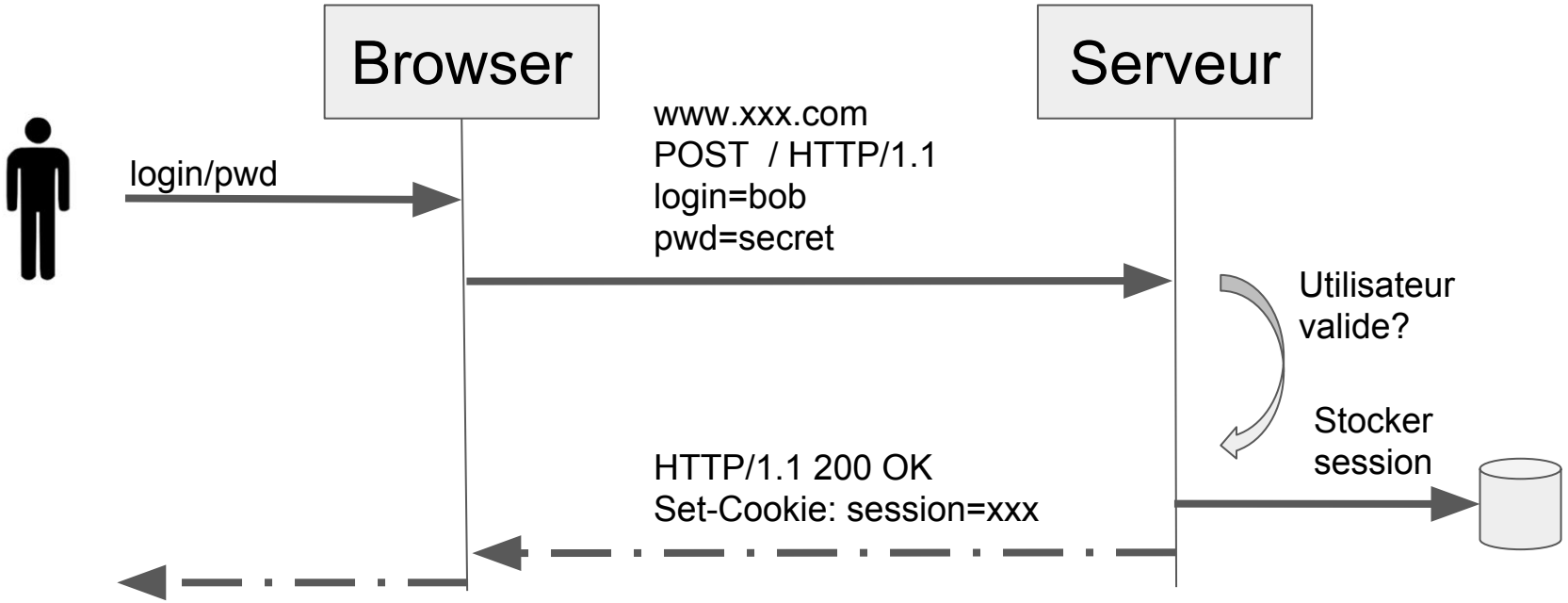
Où stocker le numéro de session?

http://www.monsite.com/page.php?id=45&PHPSESSID=0c6ca5b447035bbb274830f1ad7695bc

Solutions trouvées

- Cookie
  - Fichier texte stocké côté client
- Connexion : Token de session
- Informations dans les requêtes
  - Paramètres
  - Cookies

Connexion


Browser
Serveur
login/pwd
www.xxx.com
POST / HTTP/1.1
login=bob
pwd=secret
Utilisateur
valide?
Stocker
session
HTTP/1.1 200 OK
Set-Cookie: session=xxx

Lecture

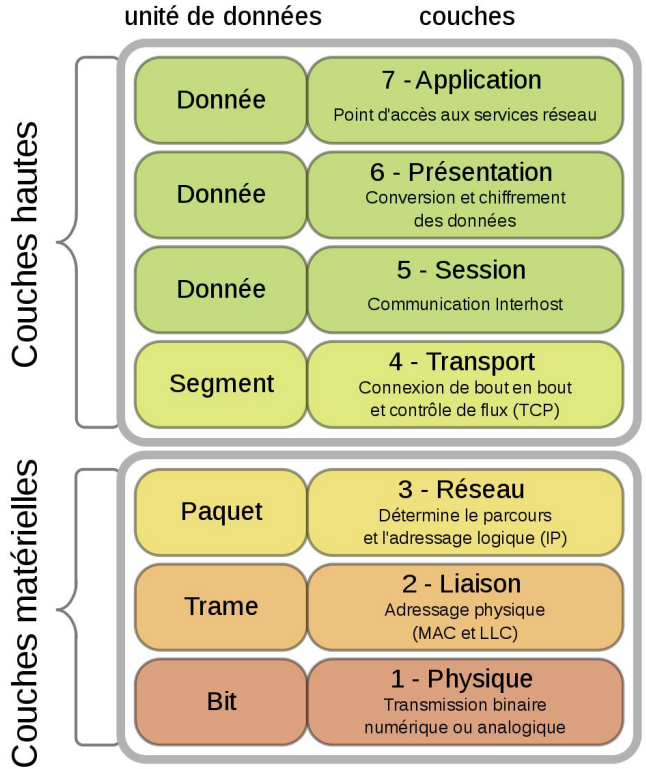

Browser
Serveur
Afficher page
www.xxx.com
GET / HTTP/1.1
Cookie: session=xxx
Récupérer
session
Session
valide?
HTTP/1.1 200 OK

Cookie

- Stocké côté client
  - Peut être changé
  - Chiffrement par le serveur
- Secure
- HttpOnly

Modèle OSI

HTML



Source: https://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI

HTML

- Frame
- Each frame of a page has an origin
  - Origin = protocol://host:port
  - Same Origin Policy : restreint la manière dont un document ou un script chargé depuis une origine peut interagir avec une autre ressource chargée depuis une autre origine.
- Frame can access data on frame with the same origin
  - Network access, Read/write DOM, Storage (cookies)
- Frame cannot access data associated with a different origin
- Same Origin Policy
  - Fichier javascript
  -

Javascript

- Langage de programmation
- Inventé par Netscape en 1995
- Pages web dynamiques
- Node

Cookie

Quid si j’arrive à mettre dans une page:

<img src=x
onerror="&#0000106&#0000097&#0000118&#0000097&#0000115&#0000099&#0000114&#0000105&#0000112&#0000116&#0000058&#0000097&#0000108&#0000101&#0000114&#0000116&#0000040&#0000039&#0000088&#0000083&#0000083&#0000039&#0000041">

- document.cookie

Session

- Interception: Token stealing/Session fixation
- Token prévisible
  - PRNG
  - Token ++

Protection

- httpOnly : cookie non accessible
- CSP

Top 10 OWASP

Top 10 OWASP

- OWASP : Open Web Application Security Project
- Top 10 : les 10 risques les plus courantes
  - Top 10 WEB, Top 10 Mobile
- Autre classement : SANS 25
- Existe depuis 2004, v5
- Propose de nombreux guides à destination des managers, développeurs, testeurs, ...

OWASP Top 10 de 2007 à nos jours

2007	2010	2013	2017
A1 - XSS	A1-Injection	A1-Injection	A1-Injection
A2 - Injection flaws	A2-Cross Site Scripting (XSS)	A2-Broken Authentication and Session Management	A2-Broken Authentication
A3 - Malicious File Execution	A3-Broken Authentication and Session Management	A3-Cross-Site Scripting (XSS)	A3-Sensitive Data Exposure
A4 - Insecure Direct Object Reference	A4-Insecure Direct Object References	A4-Insecure Direct Object References	A4-XML External Entities
A5 - CSRF	A5-Cross Site Request Forgery (CSRF)	A5-Security Misconfiguration	A5-Broken Access Control

OWASP Top 10 de 2007 à nos jours

2007	2010	2013	2017
A6 - Information Leakage and Improper Error Handling	A6-Security Misconfiguration	A6-Sensitive Data Exposure	A6-Security Misconfiguration
A7 - Broken Authentication and Session Management	A7-Insecure Cryptographic Storage	A7-Missing Function Level Access Control	A7-XSS
A8 - Insecure Cryptographic Storage	A8-Failure to Restrict URL Access	A8-Cross-Site Request Forgery (CSRF)	A8-Insecure Deserialization
A9 - Insecure Communications	A9-Insufficient Transport Layer Protection	A9-Using Components with Known Vulnerabilities	A9-Using Components with Known Vulnerabilities
A10 - Failure to Restrict URL Access	A10-Unvalidated Redirects and Forwards	A10-Unvalidated Redirects and Forwards	A10-Insufficient Logging & Monitoring

Format

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App Specific	Exploitability: 3	Prevalence: 2	Detectability: 2	Technical: 3	Business ?

A1 - Injection

1 - Injection

Definition : send untrusted data to an interpreter to change the course of normal execution

Three elements

- Untrusted source of data
- An interpreter
- Something which calls the interpreter and send data

Risks

- Modifying behavior of system

1 - Example : SQL Injection in Java

```
ResultSet res = st.executeQuery('SELECT * FROM
User where userId='' + user + '' and password=''
+ password+'');
```

What happens if password == XXX' or '1'='1?

Result :

```
SELECT * FROM  User where userId='admin' and
password='XXX' or '1'='1'
```

1 - Example : SQL Injection in Java

HQL : Take care!

```
session.createQuery("from User where userId='' +
user + '' and password='' + password+'' );
```

1 - Injection

- How to fix it?
  - Validate input
  - Escape input
  - Sanitize input
  - Limit rights (only allow a user to read certain tables)
  - Limit resources
  - SQL : Prepared Statements

Web Goat

A2 - Broken authentication

A2 - Broken Authentication

Définition : le processus d’authentification ne joue pas son rôle

- Authentification affaiblie
- Pas de MFA
- Mauvaise gestion des sessions

Risques

- Access to account

A2 - Broken Authentication

- How to fix it?
  - 2-FA, M-FA
  - Format du mot de passe
  - Interdire les mots de passe trop simples
  - Interdire les mots de passe trop courants
  - Limiter le nombre de tentatives, en fonction de l'IP, de l'heure (soir) ou de la date (week-end), ...
  - Respecter les guidelines pour les sessions

Web Goat

A3 - Sensitive Data Exposure

A3 - Sensitive Data Exposure

Définition : des informations sensibles sont affichées

- Chiffrement incorrect

Risques

- Confidentialité

A3 - Sensitive Data Exposure

- How to fix it?
  - Demander de l’aide à un expert
  - Utiliser des outils de vérification
  - Changer les secrets
  - Respecter les guidelines de sécurité

Web Goat

A4 - XXE

A4 - XXE

Définition : exploitation des moteurs XML

- Exécution de XML non sûr

Risques

- Lecture de fichier
- DoS

XXE example

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

A4 - XXE

- How to fix it?
  - Limiter les droits du moteur (processus indépendant, isolé, utilisateur avec des droits limités, quota linux, configuration…)
  - Echappement/Validation

Web Goat

A5 - Broken Access Control

A5 - Broken Access Control

Définition : Accéder à des informations sans avoir les autorisations

- Elevation of privilege
- CORS
- Insecure Direct Object References
- Missing Function Level Access Control

Risques

- Utiliser des fonctions non autorisées
- Manipuler à des données incorrectes

A5 - Broken Access Control

- How to fix it?
  - Complete mediation : vérifier tous les droits
  - RBAC/ACL
  - Deny by default

A5 - Broken Access Control

- Exemple avec Apache

```
Order Allow,Deny
Deny from 127.0.0.1
...
```

Web Goat

A6 - Security Misconfiguration

A6 - Security Misconfiguration

Définition : exploiter une faiblesse dans la configuration

- Pages d’installation laissées
- Droits insuffisants sur un dossier
- Stacktrace

Risques

- Configuration à la main de l’attaquant
- Accès à des informations du système

Google Hacking Database

- Essayons de trouver la page d’installation Wordpress d’un site

Stacktrace

```
org.h2.jdbc.JdbcSQLException: Column count does not match; SQL statement:
INSERT INTO USERS(NAME, PASSWORD) values('', '') [21002-196]
    at org.h2.message.DbException.getJdbcSQLException(DbException.java:345)
    at org.h2.message.DbException.get(DbException.java:179)
    at org.h2.message.DbException.get(DbException.java:155)
    at org.h2.message.DbException.get(DbException.java:144)
    at org.h2.command.dml.Insert.prepare(Insert.java:265)
    at org.h2.command.Parser.prepareCommand(Parser.java:263)
    at org.h2.engine.Session.prepareLocal(Session.java:578)
    at org.h2.engine.Session.prepareCommand(Session.java:519)
```

A6 - Security Misconfiguration

- How to fix it?
  - Utiliser un scanner de vulnérabilités
  - Suivre les guidelines
  - Filtrer les informations (détection du mot de passe dans un message de log ⇒ suppression)

Web Goat

A7 - XSS

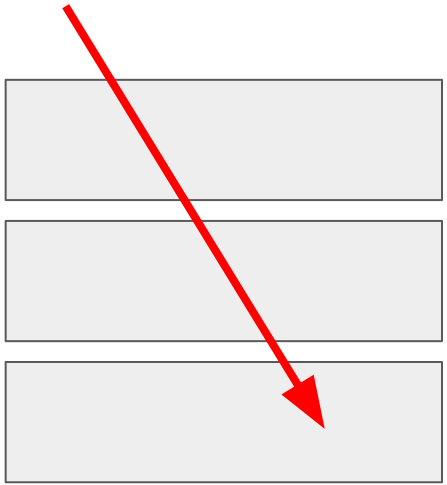
A7 - Cross-Site Scripting (XSS)

- A type of injection
- Front page interprets unsafe data

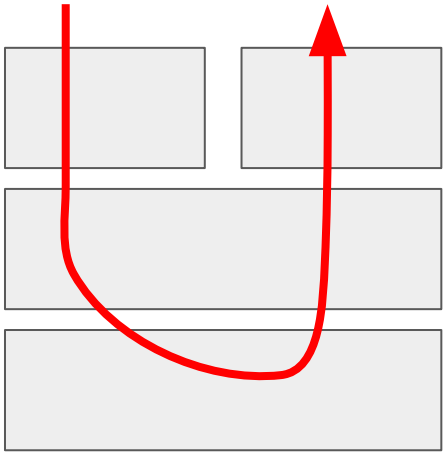
Risks:

- Site defacing
- Cookie theft
- Redirection
- Malicious operation on corrupted website

Injection vs XSS



Injection



XSS

Don’t Mix Code and Data (*Writing Secure Code*)

- Violation d’un principe de sécurité
- HTML DOM & Code Javascript
- Rend la page dynamique

Cross-Site Scripting (XSS)

- How to fix XSS?
  - Like others injections (sanitize, escape, …)
  - Use anti-xss frameworks (eg. Angular {{ }})
  - Use HTTP headers
    - X-XSS-Protection
    - CSP

Web Goat

A9 - Using Components with Known Vulnerabilities

A9 - Using Components with Known Vulnerabilities

- A type of injection
- Front page interprets unsafe data

Risks:

- Site defacing
- Cookie theft
- Redirection
- Malicious operation on corrupted website

A9 - Using Components with Known Vulnerabilities

- How to fix XSS?
  - Utiliser des libraires connues et reconnues
  - Utiliser un scanner de paquets vulnérables (OWASP Dependency Check, npm audit, SafeNuGet, …)

Demo
NPM Audit

Attention!

- Il existe beaucoup d’autres attaques/failles
  - Time attack
  - Race condition
  - Injection de fautes
  - Insecure Redirect
  - CRLF/Template/HTML Injection
- De nouvelles attaques apparaissent régulièrement

New solutions, new problems

- HTTP/2
- Worker
- WebSocket
-

New protections

- CSP
- SRI (Sub Resource Integrity)
-

Browser protection

- Sandboxing
- Depedance à l’OS : GDI Windows

CVE