UNIVERSITÉ
CÔTE D'AZUR

LP-SIGD IOTIA
Cours Sécurité sur le web

Marie-Agnès PERALDI-FRATI
Maître de conférences en informatique
www.i3s.unice.fr/~map
map@unice.fr

Description du module

- Objectifs
  - Sécuriser et protéger les sites web
  - Comprendre les menaces et vulnérabilités des applications et les infrastructures du web
  - Sécuriser les serveurs
  - Sécuriser les applications
- Prérequis
  - Notions réseau de base (modèles OSI, fonctionnement protocoles TCP/IP)
  - Notions système de base (système de fichiers, droits d'accès, OS et packages linux, déploiement service)
  - Programmation web, html, php, sql
- Compétences :
  - Notion d'infrastructure minimale sécurisée pour le web
  - Protection des serveurs contre les attaques client
  - Bonnes pratiques du développement web

Organisation du module

- Volume Horaire
  - 5h CM, 20hTD/TP
- Planning:
  - 6 Semaines de 4h C/TD/TP par semaine
  - DS 1h en fin de module
- Coeff /Notation 2
  - Séances (1)
  - Examen en fin de module (1)
- Enseignants :
  - Marie-Agnès Peraldi-Frati : Cours ,TD/TP **map@unice.fr**

Références

- **Site ANSSI** **A**gence **N**ationale de **S**écurité **des S**ystèmes d'**I**nformations
- https://legissa.ovh/internet-se-proteger-des-pirates-et-hackers.html
- https://docplayer.fr/4938763-Services-web-dan-vodislav-universite-de-cergy-pontoise-master-informatique-m1-cours-ied-plan.html
- ANSSI *Recommandations pour la sécurisation des sites web* https://www.ssi.gouv.fr/uploads/IMG/pdf/NP_Securite_Web_NoteTech.pdf
- ANSSI *Sécurité des applications Web et vulnérabilités de type « injection de données »* https://www.cert.ssi.gouv.fr/information/CERTA-2004-INF-001/
- Top 10 de l'OWASP : *les dix risques de sécurité applicatifs Web les plus critiques* **https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project**
- Magali Contensin - Kai Poutrain *PHP et la sécurité*, GNU/Linux Magazine HS n° 079 | juillet 2015 | http://cesar.resinfo.org/cours-prive/cours-contensin/SecuriteAppliWeb-contensin.pdf
- http://docplayer.fr/9996435-Securite-web-client-magali-contensin-anf-dev-web-asr-carry-le-rouet-25-octobre-2012.html#
- https://aresu.dsi.cnrs.fr/IMG/pdf/failles_de_securite_v1-3.pdf

Sommaire Cours

I. Introduction – problématique de la sécurité – sécurité web
II. Présentation des menaces – vulnérabilités
III. Standards de sécurité utiles pour le web
IV. Architecture du web (infrastructure et applicatif)
V. Sécurisation coté serveur
VI. Sécurisation des applications
VII. Sécurisation de l'infrastructure

Sommaire TD/TP

Pour bien se défendre il faut d'abord savoir attaquer

- TP1 Installation - Déploiement – sécurisation d'un serveur web
- TP2: + BD (aspect système droit d'accès, patch sécurité)
- TP3: Failles applicatives
- TP4: Sécurisation infrastructure

Sommaire Cours

I. Introduction – problématique de la sécurité – sécurité web
II. Présentation des menaces – vulnérabilités
III. Standards de sécurité utiles pour le web
IV. Architecture du web (infrastructure et applicatif)
V. Sécurisation coté serveur
VI. Sécurisation des applications
VII. Sécurisation de l'infrastructure

Introduction

- Un site web est un mélange complexe de plusieurs applications (base de données, code PHP, JavaScript, fichiers de configuration, configuration réseau, SSL, etc.)
- La sécurité informatique vise généralement cinq principaux objectifs :
  - *Confidentialité* :
    - seules les personnes autorisées ont accès aux ressources échangées ;
    - Information inintelligible aux personnes non concernées
  - *Authentification:*
    - Emetteur et destinataire doivent confirmer l'identité de chacun.
  - *Intégrité des messages :*
    - les données sont bien celles que l'on croit être
    - émetteur et destinataire peuvent s'assurer que le message n'est pas modifié
  - *Accès et disponibilité*:
    - Les services doivent être accessibles et disponibles aux utilisateurs
  - *non répudiation,*
    - permettant de garantir qu'une transaction ne peut être niée

Introduction

- Sites et services web: par nature très exposés par les systèmes d'information
- Menaces principales pour un site web*
  - Défiguration /destruction des serveurs web, internes et bases de données de l'entreprise
    - Modification du site : remplacement du contenu légitime, dénigrement entreprise, revendication
    - Collecte de données
  - Déni de service
    - Indisponibilité du service pour les utilisateurs
  - Porte d'entrée vers le système d'information
  - Hameçonnage d'utilisateurs
- Conséquences potentielles
  - Déficit d'image,
  - Manque à gagner pour l'entreprise
  - Création de back door vers l'intranet de l'entreprise et/ou l'hébergeur
  - Récupération de données d'utilisateurs du site

- **Extrait du livre blanc ANSSI https://www.ssi.gouv.fr/guide/recommandations-pour-la-securisation-des-sites-web/

Introduction

- La sécurité nécessite une approche globale => sécurité de toute la chaîne = sécurité du maillon le plus faible
- Actions de sécurisation préventives et offensives à mettre en œuvre face à ces menaces :
  - Infrastructure :
    - matériel et logiciel à la base de l'hébergement du site/service
  - Architecture :
    - Mise en place d'une défense en profondeur sur l'architecture
    - Veille sur les composants logiciels tiers
    - Mécanismes de protection de l'administration du site
    - Protection de la disponibilité du site
    - Filtrage réseau et utilisation de protocoles sécurisés client serveur
  - Code et gestion applicative du site
    - Veille et gestion des entrées client (injection de code, Cross Site Scripting, redirections, inclusion de fichiers)
    - Bonne gestion des rôles et privilèges et droits d'accès
    - Limitation des fuites d'informations
    - Gestion des sessions (cookies, vol de session, …)
  - Réactivité aux attaques
    - Détection des incidents, surveillance, journalisation des logs

**Extrait du livre blanc ANSSI https://www.ssi.gouv.fr/guide/recommandations-pour-la-securisation-des-sites-web/

Introduction

- Les causes de l'insécurité
  - État actif d'insécurité :
    - Non connaissance des fonctionnalités système (désactivation des services non utiles, …)
    - Non connaissance des bonnes pratiques en programmation (connaissance des failles des logiciels, …)
  - Etat passif d'insécurité
    - Méconnaissance des moyens de sécurité à mettre en place

Sommaire Cours

I. Introduction – problématique de la sécurité – sécurité web
II. Standards de sécurité utiles pour le web
III. Présentation des menaces – vulnérabilités
IV. Architecture du web (infrastructure et applicatif)
V. Sécurisation coté services
VI. Sécurisation des applications
VII. Sécurisation de l'infrastructure

Standards de sécurité utiles pour le web

WASC : Web Application Security Consortium - exhaustif

- Association d'experts internationaux, d'industriels et organisations du monde Open Source
- Publie des recueils de bonnes pratiques de sécurité

OWASP : Open Web Application Security Project: top 10

- communauté en ligne travaillant sur la sécurité des applications web
- publier des recommandations de sécurisation Web
- propose des méthodes et outils de référence permettant de contrôler le niveau de sécurisation de ses applications web.

ANSSI : Agence Nationale de Sécurité des Systèmes d'Informations

- agence rattachée au Secrétaire général de la défense et de la sécurité nationale
- met à disposition des guides sur la gestion des menaces informatiques
- Donne des recommandations et bonnes pratiques pour la sécurité des systèmes informatiques.

Sommaire Cours

I. Introduction – problématique de la sécurité – sécurité web
II. Standards de sécurité utiles pour le web
III. Présentation des menaces – vulnérabilités
IV. Architecture du web (infrastructure et applicatif)
V. Sécurisation coté serveur
VI. Sécurisation des applications
VII. Sécurisation de l'infrastructure

Vocabulaire de la sécurité informatique

- Menace : événements pouvant diminuer ou porter atteinte à l'intégrité du système et de son environnement, pendant toute la durée de l'activité du système,
- Vecteur de menace : chemin ou moyen par lequel un criminel peut accéder à un ordinateur ou à un réseau pour y introduire une menace
- Mitigation des risques : moyens et méthodes mis en place pour endiguer et réduire les menaces et les vecteurs de menace

Présentation des menaces

Les 5 principales attaques observées chez OVH* :

- DDoS basées sur l'IoT
- Vol de données personnelles
- Attaques sur les services Cloud
- Malwares ciblés sur les applications
- La fraude en ligne

Quantification des attaques :

- 1800 /jour
- Majoritairement entre 19h et 21h.
- Majoritairement services de jeux en ligne ou des plateformes de commerce

* OVH stat sur attaques web https://www.ovh.com/fr/blog/rapport-attaques-ddos-observees-par-ovh-en-2017/

Présentation des menaces

Schéma d'attaque classique basé sur l'utilisation des failles de sécurité

1) Collecte d'information sur le site à attaquer (requêtes indirectes) :
   - WHOIS ;
   - Google Hacking.

2) Repérage des lieux (requêtes directes) :
   - requête DNS ;
   - balayage de ports ;
   - reconnaissance des OS et services ;

3) Détermination des vulnérabilités

4) Attaques en déni ou attaques d'accès (intrusion), suivies de :
   - élévation de privilèges ;
   - collecte d'informations ;
   - repérage des lieux ;
   - détermination des vulnérabilités ;
   - effacement des traces.

Présentation des menaces – vulnérabilités et leurs évolution

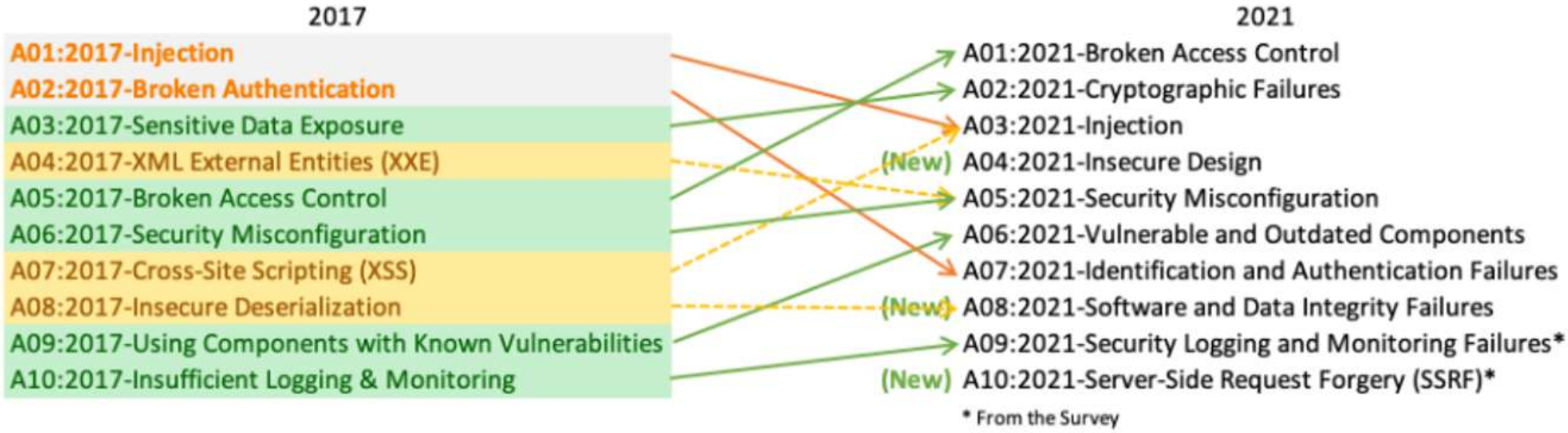
Le top 10 des menaces selon OWASP(Open Web Application Security Project)*

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

*https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

Présentation des menaces – vulnérabilités et leurs évolution

Le top 10 des menaces selon OWASP(Open Web Application Security Project)*



*https://owasp.org/Top10/

Présentation des menaces

Typologie des attaques

- Attaques applicatives : injection code,
- Attaques coté client : Cross Site Scripting (XSS), gestion de session, Phishing…
- Attaques coté serveur (authentification, autorisation, …)
- Attaques d'infrastructures :configuration standard, DDOS,

Classification et regroupement des menaces coté serveur et coté client

1. Injection de commandes
    a. Débordement de tampon (buffer overflow)
    b. Chaîne de format (format string)
    c. Injection LDAP
    d. Injection de commandes (OS Commanding)
    e. Injection SQL
    f. Injection SSI
    g. Injection XPath

2: Révélation d'informations
    a. Listing de répertoires (directory indexing)
    b. Fuite d'informations (information leakage)
    c. Traversée de chemin (path traversal)
    d. Prédiction de localisation de ressources (predictable resource location)

3: Authentification
    a. Force brute (brute force)
    b. Authentification insuffisante (insufficient authentication)
    c. Mauvais traitement des recouvrements de mot de passe (weak password recovery validation)

4: Autorisation
    a. Prédiction de session (credential/session prediction)
    b. Autorisation insuffisante (insufficient authaurization)
    c. Expiration de session insuffisante (insufficient session expiration)
    d. Fixation d'identifiant de session (session fixation)

5. Attaques côté client
    a. Usurpation de contenu (content spoofing)
    b. XSS (Cross Site Scripting)

6. Logiques
    a. Abus de fonctionnalité (abuse of functionality)
    b. Déni de service (denial of service)
    c. Anti-automatisation insuffisante (insufficient anti-automation)
    d. Validation insuffisante du flux logique de l'application

7. Autres
    a. HTTP Response Splitting / CR LF Injection
    b. Prise d'empreinte (Web Server/Application Fingerprinting)

* http://projects.webappsec.org/f/WASC-TC-v2_0.pdf

Sommaire Cours

I. Introduction – problématique de la sécurité – sécurité web
II. Présentation des menaces – vulnérabilités
III. Standards de sécurité utiles pour le web
IV. Architecture du web (infrastructure et applicatif)
V. Sécurisation coté serveur
VI. Sécurisation des applications
VII. Sécurisation de l'infrastructure

Architecture du web

En règle générale, une application web est découpée en 3 couches applicatives:

- La couche présentation ou IHM (Interface Homme/Machine) gère les interactions utilisateur/machine, la présentation des données
- La couche traitement:
  - Locaux :contrôles effectués au niveau du dialogue avec l'IHM
  - Globaux : L'application elle-même
- La couche données: Gère le stockage des données et l'accès à ces dernières

Architecture du web

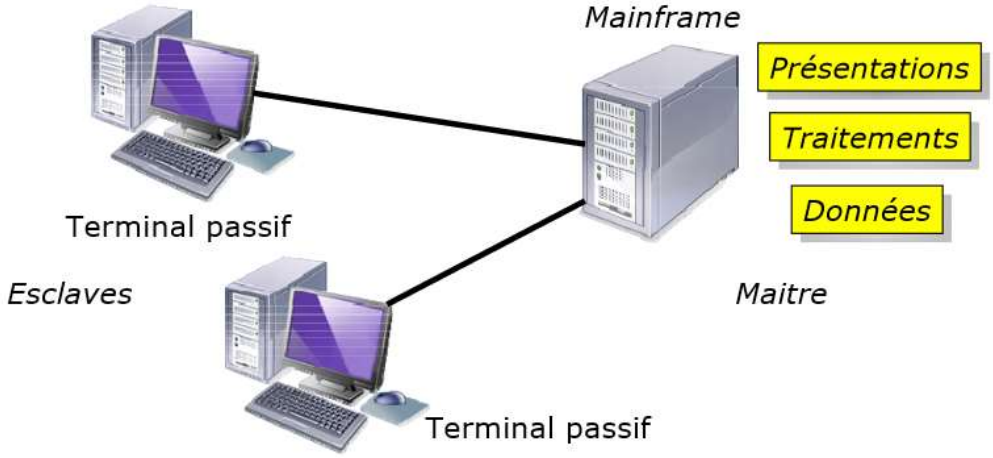
Ces 3 niveaux peuvent être imbriqués ou répartis de différentes manières:

Architecture 1-tiers : 3 couches sur la même machine, informatique centralisée

Architecture 2-tiers

Architecture 3-tiers

Architecture n-tiers

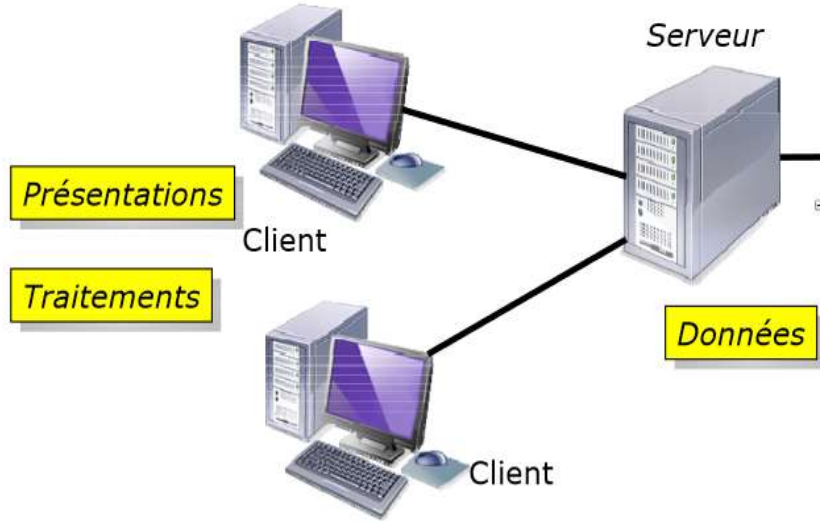


Architecture du web

Ces 3 niveaux peuvent être imbriqués ou répartis de différentes manières:

- Architecture 1-tiers
- Architecture 2-tiers : Présentation et traitements sur le client, les données sur le serveur, Contexte multi-utilisateurs avec accès aux données centralisées (middleware)

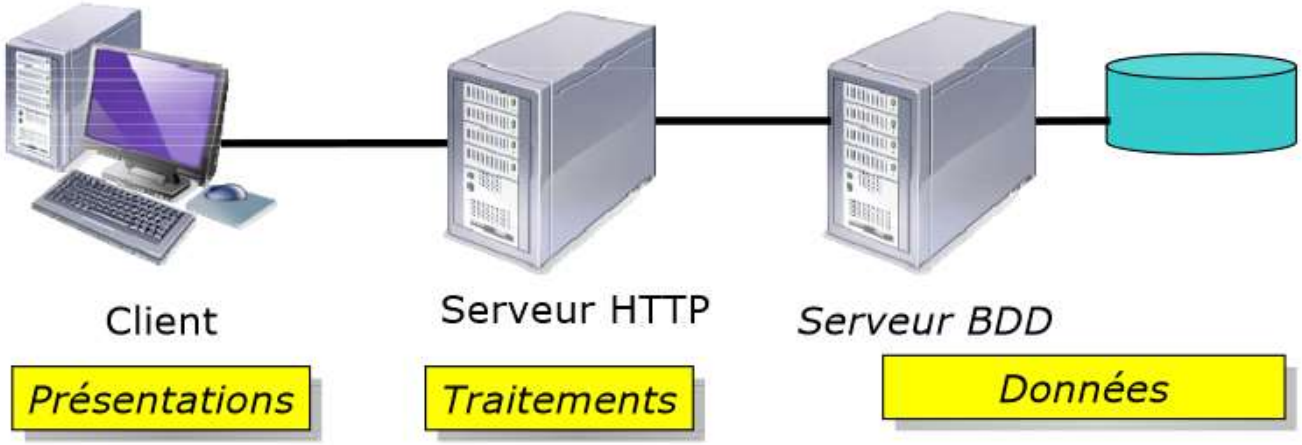
- Architecture 3-tiers
- Architecture n-tiers



Architecture du web

Ces 3 niveaux peuvent être imbriqués ou répartis de différentes manières:

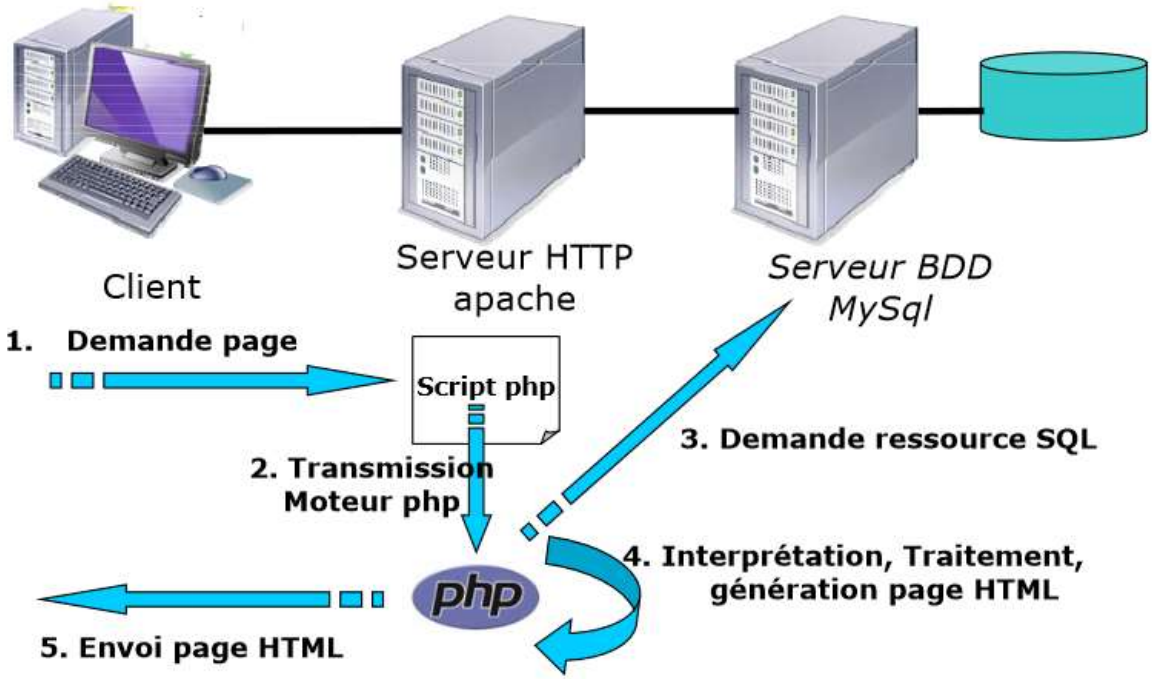
- Architecture 1-tiers
- Architecture 2-tiers
- Architecture 3-tiers : La présentation est sur le client, Les traitements sont pris par un serveur intermédiaire, Les données sont sur un serveur de données, Contexte multiutilisateur interne
- Architecture n-tiers



Architecture du web

Ces 3 niveaux peuvent être imbriqués ou répartis de différentes manières:

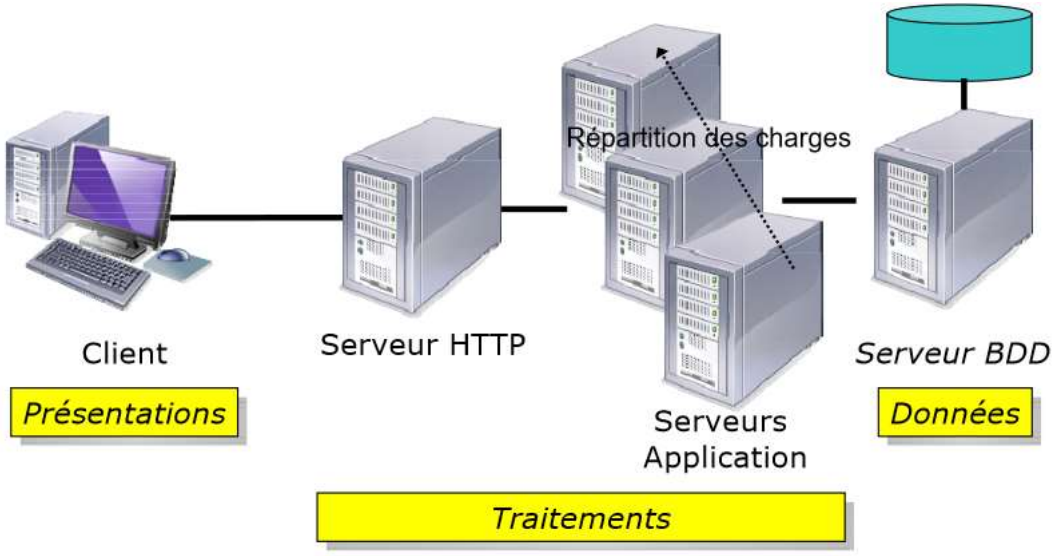
- Architecture 1-tiers
- Architecture 2-tiers
- Architecture 3-tiers : exemple PHP/MySql
- Architecture n-tiers:



Architecture du web

Ces 3 niveaux peuvent être imbriqués ou répartis de différentes manières:

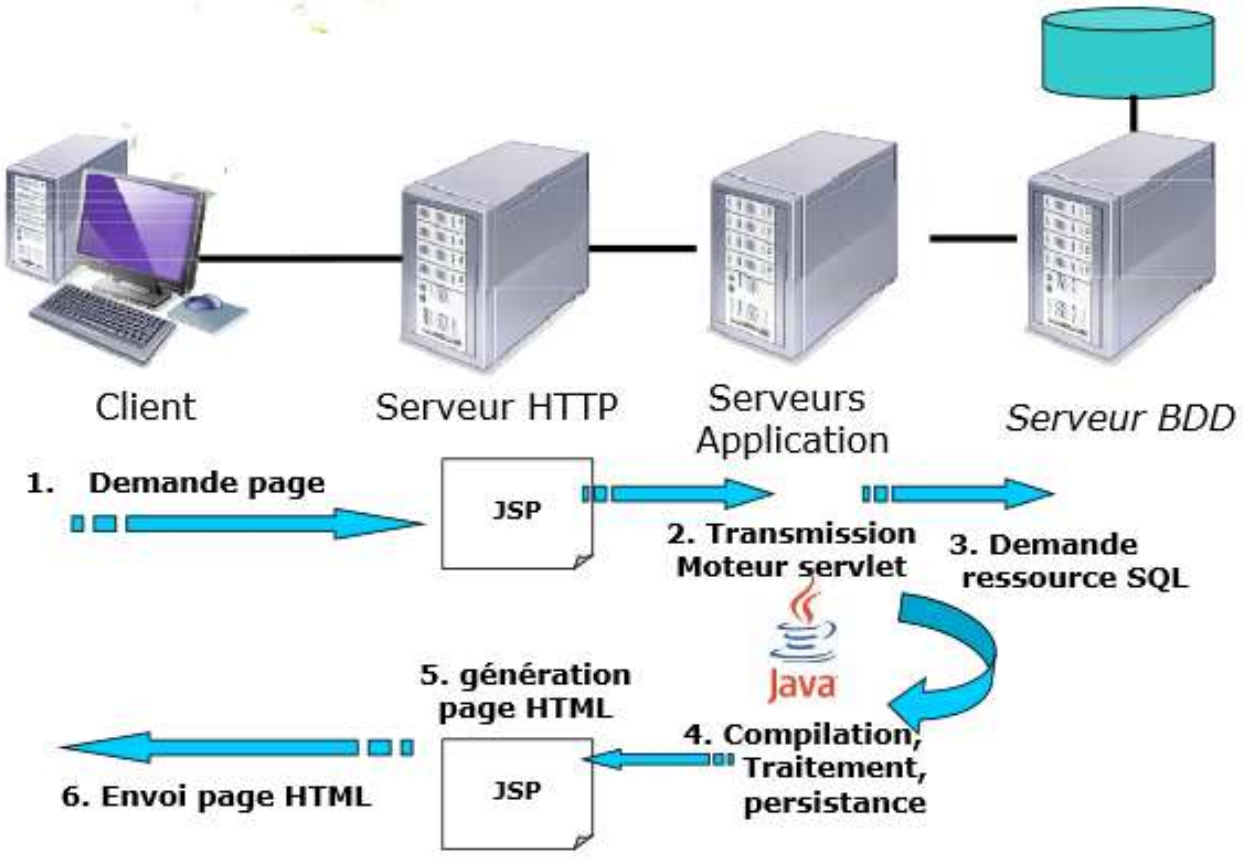
- Architecture 1-tiers
- Architecture 2-tiers
- Architecture 3-tiers
- Architecture n-tiers : La présentation est sur le client, les traitements sont pris par un serveur intermédiaire, les données sont sur un serveur de données. Contexte multi utilisateurs internet



Architecture du web

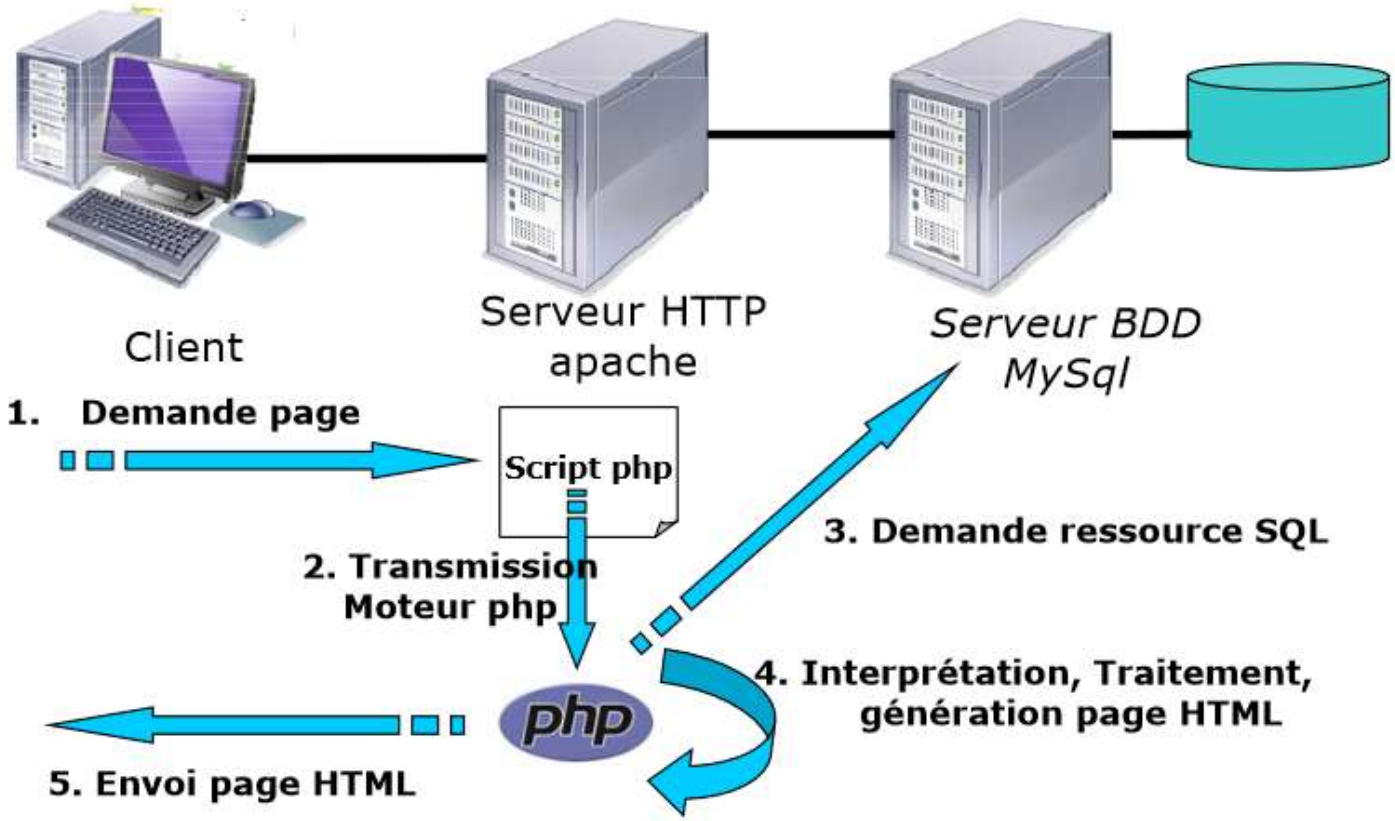
Ces 3 niveaux peuvent être imbriqués ou répartis de différentes manières:

- Architecture 1-tiers
- Architecture 2-tiers
- Architecture 3-tiers
- Architecture n-tiers : Exemple Java



Use case du module

- Service Apache
  - Installation package apache2
  - Structure
- PHP
- BD MySQL



Apache :

Caractéristiques techniques

- Serveur http gratuit
- Robuste, open source, bien documenté
- Modules SSL, Perl, PHP, Java, shtml
- Multi plateformes (Linux, Unixes OS\2, Windows)
- Virtual hosts (IP aliases, HTTP 1.1, etc)
- …

Les concepts

- HTTP vs HTML
  - HTTP (hyper-text transfert protocol) réalise une connexion telnet sur un port défini (par défault le port 80) avec une requête d'une page et recevant la page ou une erreur en retour.
  - HTML est le langage structurant les pages web. Son contenu n'est à priori pas traité par Apache.

- IP aliasing :
  - Il n’est plus nécessaire pour chaque site web différent d'avoir une machine pour lui tout seul (ni même plus simplement une carte réseau).
  - Sur le même port hardware on peut désormais avoir plusieurs adresses IP.
  - Une seule adresse IP suffit à des milliers de sites chacuns sur un différent domaine.

- Les transmissions sécurisées : SSL ou "https"

Apache : Configuration de base

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- La configuration par défaut est dans *apache2.conf*
- *Ports.conf* indique les ports sur lesquels le serveur écoute
- Les modules disponibles (php cgi, evasive…) sont sous */mods-available* , dans *mod-enabled* se trouvent les liens vers ceux qui sont actifs (`a2enmod` pour activer)
- Les fichiers de config des serveurs web sont sous *conf-available* , dans *conf-enabled* se trouvent les liens vers les configurations actives
- *sites-avalaible* contient vers la config des sites installés dans *sites-enabled* les liens vers ceux qui sont actifs (`a2ensite` pour activer)

En dehors d'Apache :

- /etc/hosts pour la résolution de nom de domaine
- /etc/nsswitch.conf pour donner l'ordre oú se fait la résolution de nom
-

Mise en ligne d'un serveur web

- Répertoires de stockage des sites web ainsi que les fichiers index.html : **/var/www/monsite**
- Configuration du site *monsite.com*
- virtual hosts: fichier **/etc/apache2/sites-available/monsite.conf**

```
<VirtualHost 10.1.10.1:80>
        ServerAdmin votre-mail@monsite.com
        ServerName monsite.com
        DocumentRoot /var/www/monsite
</VirtualHost>
```

- Mettre le serveur à l'écoute sur les ports « ports » **/etc/apache2/ports.conf**

```
NameVirtualHost 10.1.10.1:80
Listen 80
```

- Activer le site : lien symbolique entre *site-enabled* et *site-available*

```
$sudo a2dissite monsitewebport80.conf
```

- Redémarrer Apache

```
$sudo systemctl restart apache2 restart
```

Mise en ligne serveur web

- Ecoute sur le port
  - Définit dans **/etc/apache2/ports.conf**

```
NameVirtualHost *:80
Listen 80
```

- Résolution local de nom de domaine
  - En local dans **/etc/hosts** on indique les adresses ip associées aux sites

```
10.1.10.1  monsiteweb.com
```

  Ordre de la résolution à préciser dans **etc/hosts.conf**

```
# Valeurs possibles : hosts, bind, nis.
        order hosts, bind
# Autoriser plusieurs adresses par nom
        multi on
```

  - Pour l'extérieur on crée le domaine dans le serveur DNS

Modules : langages de scripts

- Langages de script => dynamicité des pages web html.
- Code interprété
  - coté serveur pour générer une page html -> des failles de sécuité peuvent apparaître lors de l'iterprétation.
  - Soit coté client pour faire de la vérification avant envoi des requêtes vers le serveur
- Interprèteurs sont intégrés sous forme de module dans apache (js, php, perl) et dans le client (applets) soit en natif (servlet, html, css, …)

Les services et langages coté serveur

- CGI-BIN
  - Code compilé (binaire) ou interprété (script) exécuté coté serveur
  - Génère un résultat envoyé au client
  - Temps d'exécution peut être lent
  - Permet de récupérer des données d'un formulaire, lire, écrire dans une base de donnée
- Exemple: affichage de la date du serveur sur une page :

```
#!/bin/sh
# date.cgi
echo "Content-type: text/html"
echo
#Creation du corps du document
echo
"<html><head><title>date.cgi</title></head>"
echo "<h1>Date sur le serveur</h1>"
echo </html>
```

Les services et langages coté serveur

- Javascript
  - Code inclut dans la page web et chargé avec la page html
  - Dynamise des sites « côté client » : Validation de formulaire, commande personnalisée
  - Met-à-jour les éléments de la page sans la recharger grâce à AJAX
  - Code visible de l'extérieur

Les services et langages coté serveur

- Exemple Javascript: validation de formulaire

HTML

```
<form>
  <div>
    <label for="name">Enter your name:</label>
    <input type="text" name="name" id="name">
  </div>
  <div>
    <label for="age">Enter your age:</label>
    <input type="number" name="age" id="age">
  </div>
  <div>
    <input type="submit">
  </div>
  <div></div>
</form>
<script src="validation.js"></script>
```

validation.js

```
var inputs = document.querySelectorAll('input');
var labels = document.querySelectorAll('label');
var form = document.querySelector('form');

var formItems = [];

var errorField = document.querySelector('.errors');
var errorList = document.querySelector('.errors ul');

for(var i = 0; i < inputs.length-1; i++) {
  var obj = {};
  obj.label = labels[i];
  obj.input = inputs[i];
  formItems.push(obj);
}

errorField.style.left = '-100%';

form.onsubmit = validate;

function validate(e) {
  errorList.innerHTML = '';
  for(var i = 0; i < formItems.length; i++) {
    var testItem = formItems[i];
    if(testItem.input.value === '') {
      errorField.style.left = '360px';
      createLink(testItem);
    }
  }
```

Les services et langages coté serveur

- ## Java applets
  - Une applet Java est un programme (bytecode java) exécuté par le client dans sa JVM.
  - Fourni des fonctionnalités interactives non présente dans HTML. Il peut communiquer avec le serveur
  - Sur le serveur c'est un simple fichier.

HTML

```
<BODY>
     <APPLET
         CODE="AppletSimple.class"
         WIDTH=200
         HEIGHT=100>
     </APPLET>
 </BODY>
```

AppletSimple.java

```
import java.applet.*;
public class AppletSimple extends Applet
{  public void init( ) {
     this.setBackground(Color.yellow);
    }
}
```

Les services et langages coté serveur

- Java servlets
  - Intégrées à Apache
  - Tourne en permanence, pas d'exécution/chargement
  - Permet de créer dynamiquement des données au sein d'un serveur HTTP : accès à des BD, transactions pour du commerce en ligne, génération de page web

Les services et langages coté serveur

- PHP
  - Purement destiné au web
  - Orienté objet depuis la V5
  - PHP est bien plus performant que JAVA.
  - La force de PHP c'est LAMP (Linux Apache MySql PHP) : 4 composants libres suffisant pour bénéficier d'un serveur.

```
<!DOCTYPE html>
<html>
<head>
<title>Ceci est une page de test</title>
        <?php
            $date = date("d-m-Y");
            $heure = date("H:i");
            Print("Nous sommes le $date et il est $heure");
        ?>
</head>
```

Les services et langages coté serveur

- Le shtml (server-side includes/SSI)
  - SSI rend des services simples : insertion de la date dans un pied de page, gestion de compteurs d'accès
  - Code SSI interprété par Apache
  - Pas de récupération de données en provenance du client
  - overhead
  - Invocation : <!--#include virtual="../gen/toolbar.html" -->

    <!--#echo var="date" -->

Sommaire Cours

I. Introduction – problématique de la sécurité – sécurité web
II. Présentation des menaces – vulnérabilités
III. Standards de sécurité utiles pour le web
IV. Architecture du web (infrastructure et applicatif)
V. Sécurisation des services
VI. Sécurisation des applications
VII. Sécurisation de l'infrastructure