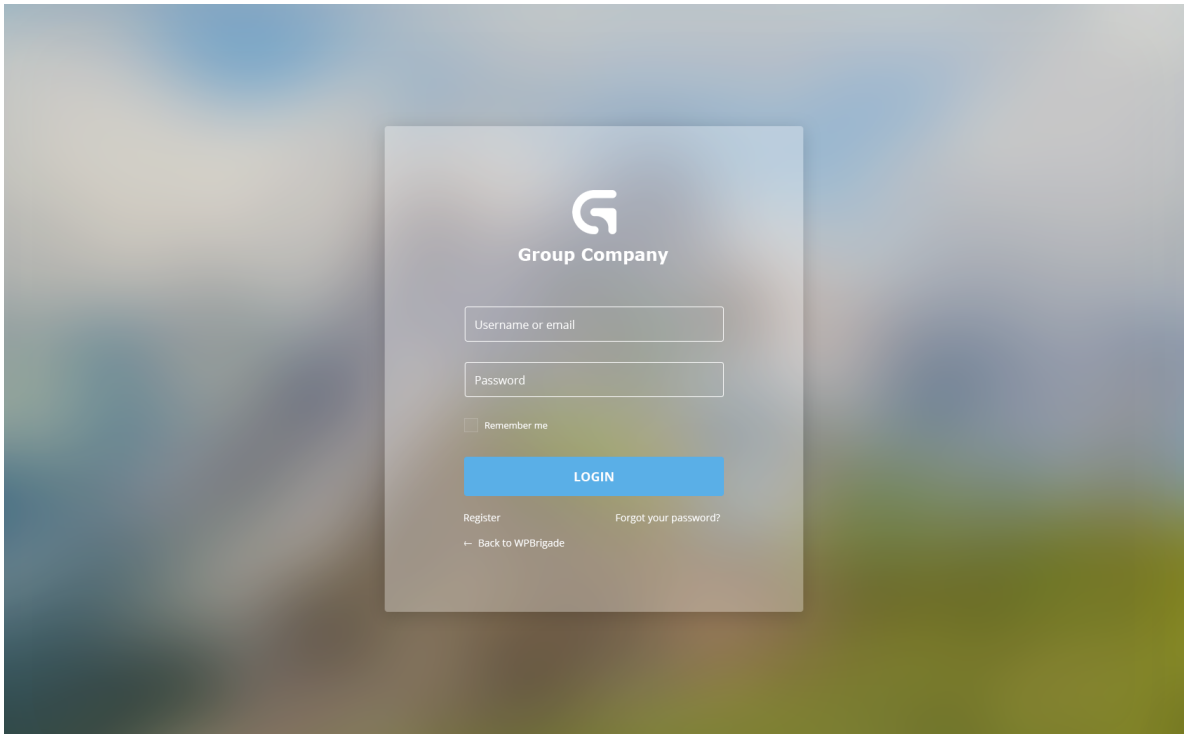
SQL Injection

Blind base

L'injection de type aveugle (ou 'Blind base') est une injection où ne connait rien de la base de donnée.



Prenons cette page de connexion classique. Ici nous avons aucune information sur ce qu'il peut avoir dans la base de donnée. Le but de cette attaque est de récuperé (et non de se connecter simplement) le mot de passe de l'administrateur. Dans une grande majorité des cas le nom de compte sera **admin**, on reprendra donc cette forte probabilité pour n'avoir que le mot de passe à trouver. On peut supposé aussi que le nom de la table sera users.

On va donc commence à tester une à une les lettres de l'alphabete, majuscule,miniscule ainsi que les chiffres et quelque caractères speciaux, **attention** certain caractère son interdit, ils ont une signification spéciale suivant la base de donnée ainsi que sa version employé.

La majorité des cas que vous pourrez voir utiliseront substr, ici pour changer un peu, nous allons utilisé ***left***(chaine,taille) qui permet de ressortir ***taille*** caractères de ***chaine***.

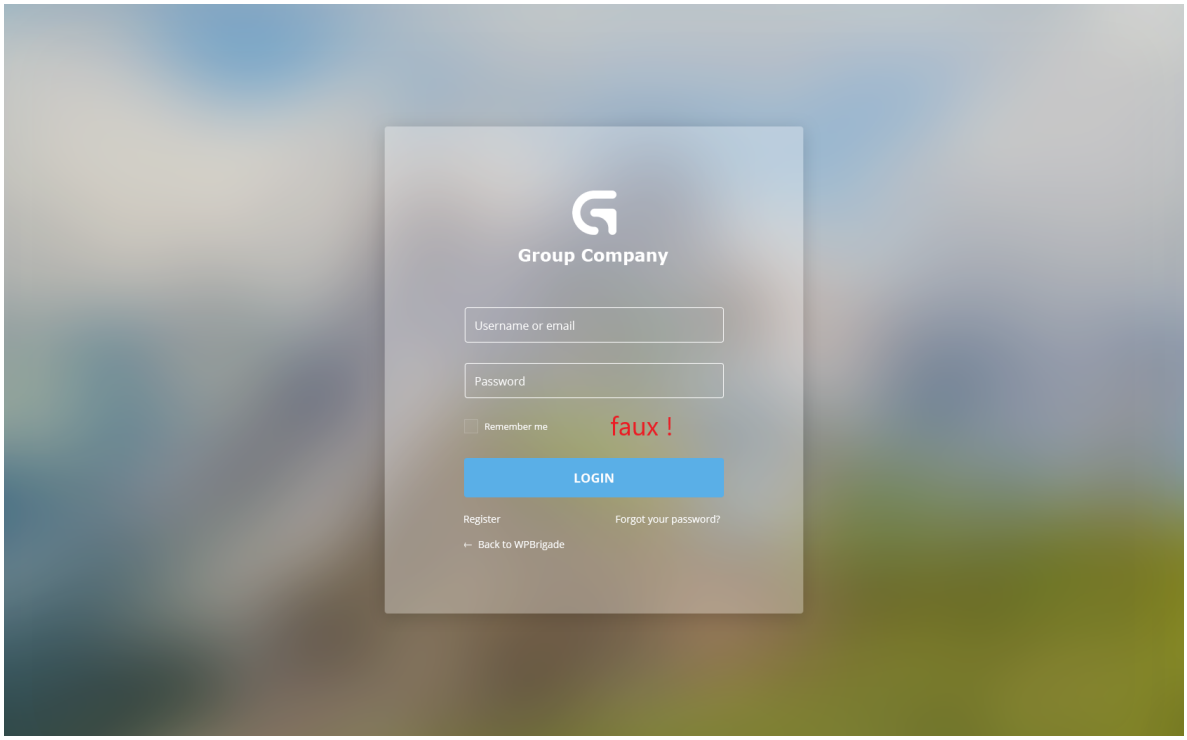
ex : left("Salut",2) => 'Sa'

on a donc USERNAME à remplir ainsi que PASSWORD

```
username="admin' and (select left(password,1)) from users where
name='admin')='a' -- -
password="salut"
```

Nous remplirons les champs ***USERNAME*** et ***PASSWORD*** comme décrit au dessus pour un premier essais

Supposons que la page nous renvois :

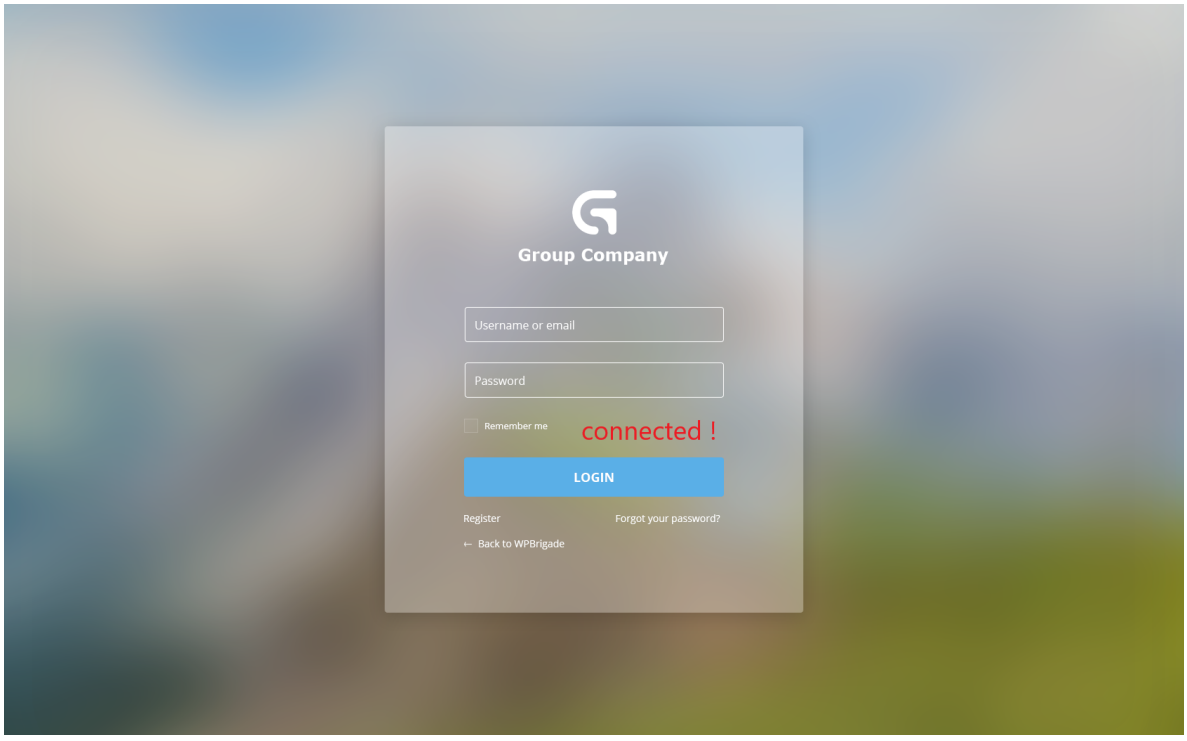


On en conclus donc que le mot de passe ne commence pas par la lettre **a**

Après avoir testé plusieurs cas, avec la commande suivante :

```
USERNAME = "admin' and (select left(password,1)) from users where
name='admin')='3' -- - "
```

Il nous apparait ceci :



On en conclut donc que le premier caractère de notre mots de passe est *'**3**'*

On peut donc passé au deuxieme comme ceci :

```
USERNAME = "and (select left(password,2)) from users where name='admin')='3a'
-- -"
```

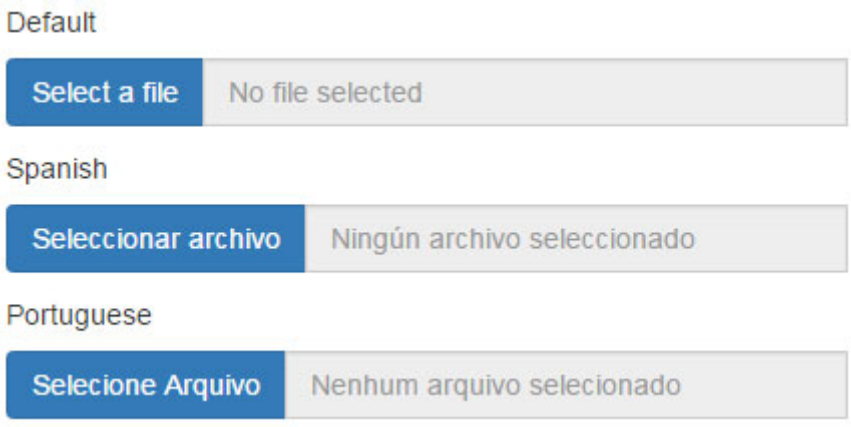
Et ainsi de suite jusqu'a trouvé toute les lettres

Voila vous comprenais maintenant l'injection SQL de type aveugle !

XXE

Une XXE de son nom XML External Entity, est une injection par du langage XML.

Prenons un site lambda permettant d'inserer du code XML, par exemple pour donner une police d'écriture spéciale, une taille d'écriture spéciale etc.



La base d'un code XML ressemble à ceci :

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
    <channel>
        <title>Mon site</title>
        <description>Ceci est un exemple de flux RSS 2.0</description>
        <lastBuildDate>Sat, 07 Sep 2002 00:00:01 GMT</lastBuildDate>
        <link>http://www.example.org</link>
        <item>
            <title>Actualité N°1</title>
            <description>Ceci est ma première actualité</description>
            <pubDate>Sat, 07 Sep 2002 00:00:01 GMT</pubDate>
            <link>http://www.example.org/actu1</link>
        </item>
        <item>
            <title>Actualité N°2</title>
            <description>Ceci est ma seconde actualité</description>
            <pubDate>Sat, 07 Sep 2002 00:00:01 GMT</pubDate>
            <link>http://www.example.org/actu2</link>
        </item>
    </channel>
</rss>
```

La méthode pour notre attaque qui va être d'injecté du code que le serveur executera, sera d'ajouté dans le header du fichier une comande dans ce genre la :

```
<!DOCTYPE HELLO [
<!ENTITY XXE_REKT SYSTEM "SOMETHING" >
]><title>&XXE_REKT;</title>
```

Ici il faut ajouté dans **SOMETHING** quelque chose que le serveur pourra traité. On pourrai choisir d'affiché par exemple le code source d'une page de login par exemple :

```
<!DOCTYPE HELLO [
<!ENTITY XXE_REKT SYSTEM "file:///PATH/TO/FILE/login.php" >
]><title>&XXE_REKT;</title>
```

On pourrai notamment poussé l'exploitation plus loins en outrepassant des filtres basique en utilisant des **WRAPPER** php qui sont des fonction interne a PHP.

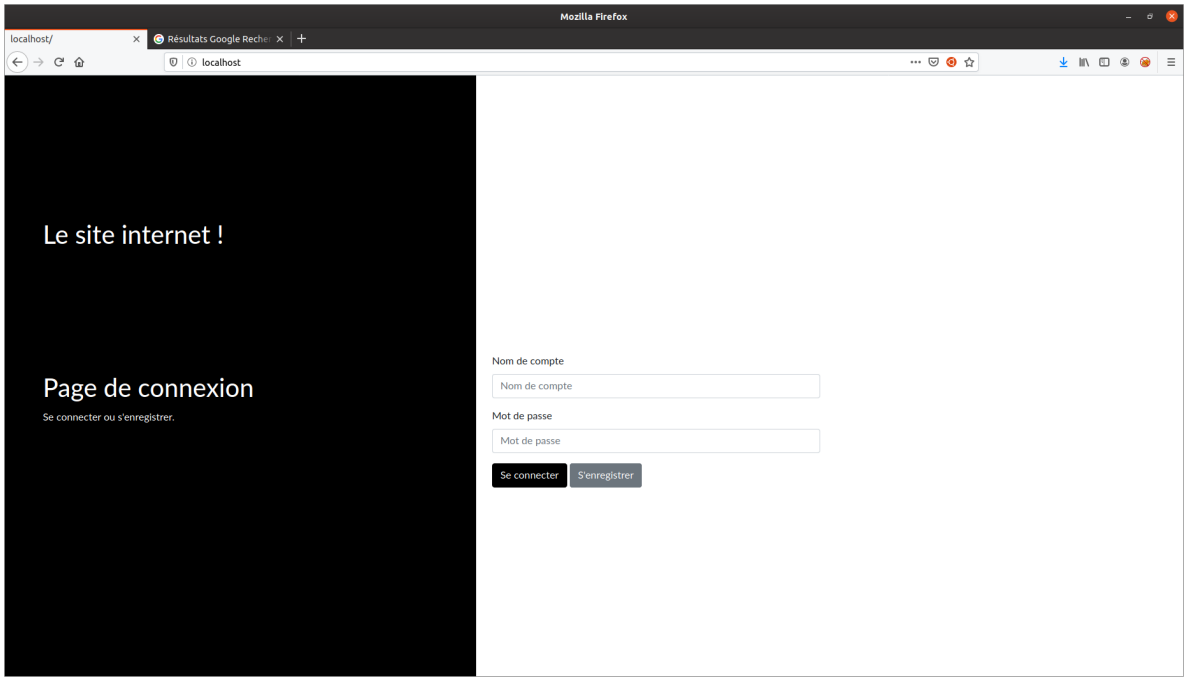
```
<!DOCTYPE HELLO [
<!ENTITY XXE_REKT SYSTEM "php://filter/read=convert.base64-
encode=PATH/TO/FILE" >
]><title>&XXE_REKT;</title>
```

Ceci nous retournerais le code sources du fichier demandé en base64, il nous resterait uniquement à décoder ce fichier pour avoir le code en clair !

Security Misconfiguration

Mauvaise configuration

Pour notre exemple prenons un site basique que nous pouvons faire nous même.

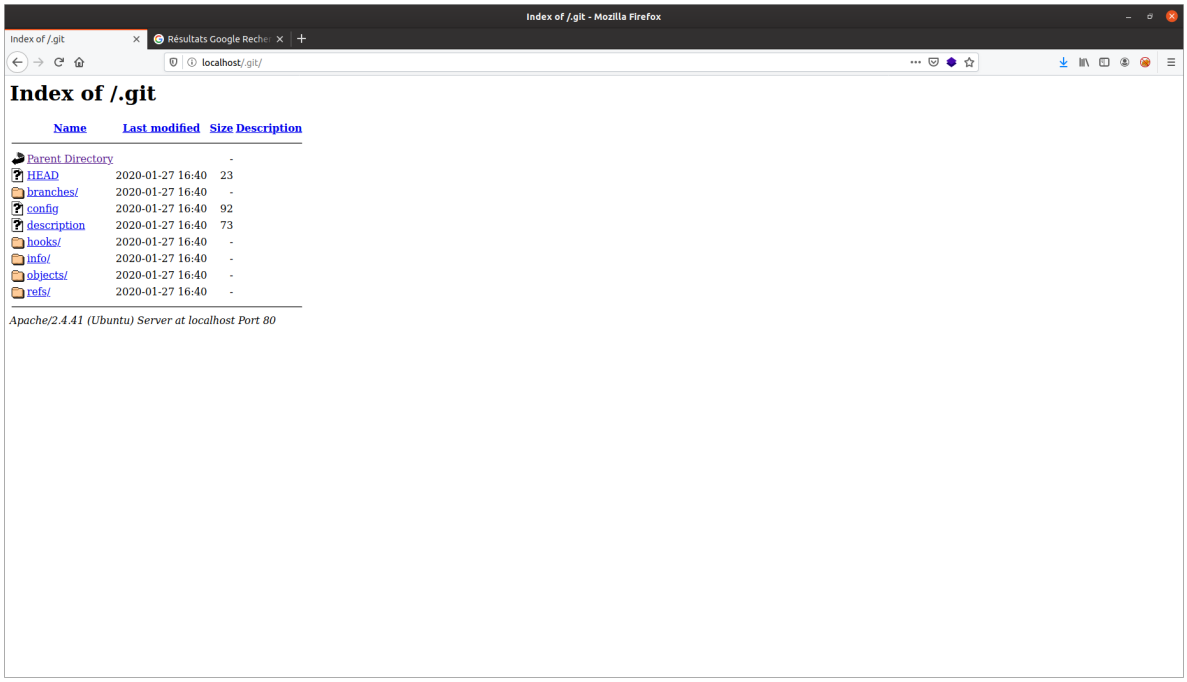


Chaque site web a des fichiers de configuration, des fichiers de gestions et surtout un dossier de production quand il est en développement. Pour le développement, ll y'a des logiciels qui permettent de facilité l'acces au code a tous les developpeur ainsi que la gestion des differente partis d'un projets, ici nous allons prendre **GIT**, c'est un logiciel de gestion descentralisé, les developpeurs peuvent déposer leurs bout de code ainsi que récuperé celui des autres personne, c'est une définition très basique de cet outils très complets.

Un dossier **git** à la création, crée un dossier nommé **'.git'** qui se trouve a la racine du dossier d'un projet et qui contient tous ce qui se trouve dans le projet, et notamment les differentes **branches** qui se trouve être les partis séparé du projet; (où la branche **master** est la partis principale d'un projet); comme par exemple la partis ***Authentification*** d'un site. Imaginons pour la suite que dans la partis authentification, les developpeur aurait mis dans le code sources, non chiffré :

```
SI ( nomDeCompte == "admin"  et  motDePasse == "Hello_je_suis_l_admin") {
```

Supposons maintenant que les developpeurs ont modifiés cette page authentification en la donnant à la branche principale **master** en enlevant la ligne de code d'au dessus, donc on ne la verrai pas. Cependant les developpeurs n'ont pas interdit l'acces au dossier **.git** à des utilisateur lambda. On tomberai sur cette situation :



On pourrai penser que ce n'est pas grave, il n'y a que la version finale ici **MAIS** ce n'est pas vrai, en telechargeant tous ce dossier, on peut recuperer chaque partis (branche) independament et donc recuperé le nom d'utilisateur et le mot de passe de l'administrateur du site. Ce qui revient a être très dangereux suivant les intentions de l'attaquant.