

Première partie: Le DésastreCTF

Nous commençons avec une simple ip: 10.0.0.5

En premier lieu, nous allons scanner la machine avec nmap, en considérant que nous sommes en CTF, en red Team, faire un nmap en entier aussi brutalement sur une machine en local, c'est réveiller toutes les oies du capitole.

Nous trouvons trois ports ouverts:

```
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
2222/tcp  open  ssh          OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
6645/tcp  open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Sans identifiants, les ports SSH ne nous intéressent que très peu, d'autant que leurs version, facilement obtenables, ne semble pas vulnérables. Nous nous concentrons donc sur la troisième issue, qu'est le port 6645. En s'y connectant à travers un navigateur web, nous tombons sur une autre plateforme CTFd:

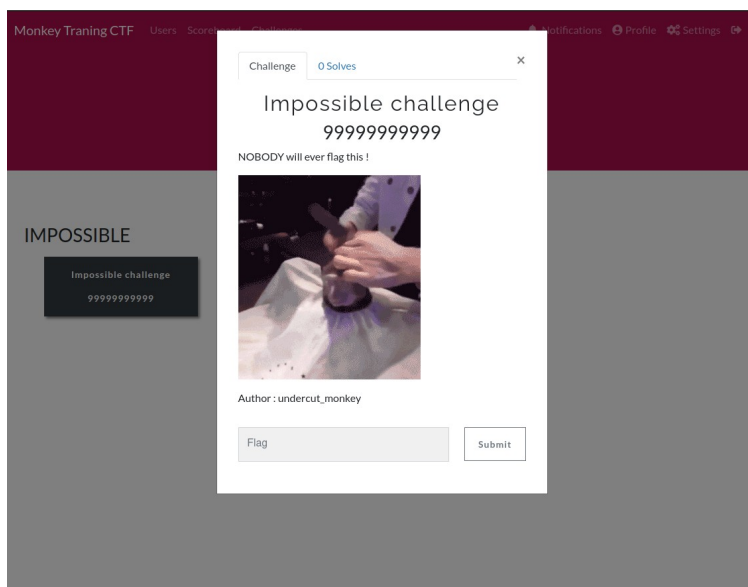
Monkey Traning CTF Users Scoreboard Challenges Register Login



Since the humans are trying to stop us,
have to get better too ...

Introducing Monkey Traning CTF, a platform
for all hacker monkeys to train and defeat
those damn humans !


Nous créons un compte banal et accédons ~~aux~~ AU challenge:



It's over 9000 !

Nous n'avons aucune indication sur le flag, mais nous avons tout de même le nom de l'auteur: **undercut_monkey**

N'ayant pas trop d'idées, nous avons consulté la liste des vulnérabilités existantes sur la plateforme CTFd: <https://vulmon.com/searchpage?q=CTFd>

 Vulmon

Recent Vulnerabilities

Research Posts

Trends

Blog

About

Contact

ctfd

☒ By Relevance ☐ By Risk Score ☐ By Publish Date ☐ By Recent Activity

ctfd vulnerabilities and exploits (subscribe to this query)

9.8

CVSSv3

CVE-2020-7245
Incorrect username validation in the registration process of CTFd v2.0.0 - v2.2.2 allows an attacker to take over an arbitrary account if the username is known and emails are enabled on the CTFd instance. To exploit the vulnerability, one must register with a username identical...CtfD CtfD  3 Github repositories available

6.5

CVSSv3

CVE-2020-5290
In RedpwnCTF before version 2.3, there is a session fixation vulnerability in exploitable through the '#token=\$ssid' hash when making a request to the '/verify' endpoint. An attacker team could potentially steal flags by, for example, exploiting a stored XSS payload in a CTF...

CtfD Rctf

La CVE à 9.8 semble bien alléchante. Nous allons tenter de l'utiliser. Pour cela, il faut créer un utilisateur avec le même nom qu'un autre, mais avec un espace avant ou après son nom. Par exemple, en remplaçant les espaces par des _ pour la visibilité, nous allons créer "_undercut_monkey", et ça fonctionne:

Settings

Profile

Access Tokens

User Name

undercut_monkey

Email

not_my_email@yopmail.com

Current Password

Maintenant, il faut nous déconnecter, puis demander à remettre à zéro le mot de passe à partir de la page de connection:

Login

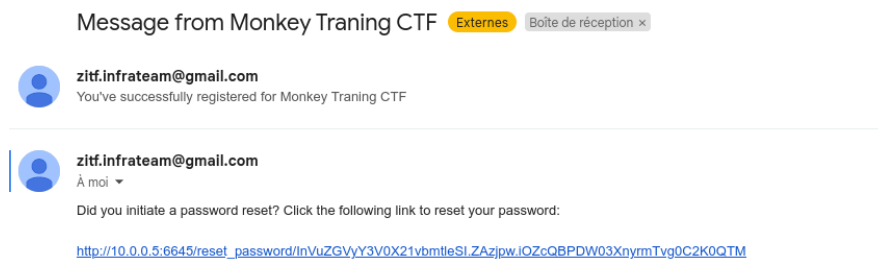
User Name or Email

Password

[Forgot your password?](#)

Forgot your password ?



Nous recevons un mail sur notre adresse mail, demandant le reset de mot de passe, et faisant suite au mail de confirmation de création de compte sur la plateforme:



Nous voila connectés en tant que l'utilisateur undercut_monkey, ayant les droits d'administrateur. Pour récupérer le drapeau, il faut résoudre le challenge impossible. Il suffit d'aller dans*

Admin Panel>Challenges>Impossible Challenge>Flags.

Le flag du challenge est le drapeau de la première partie de notre épreuve:

Type	Flag	Settings
static	ZiTF{95f2d3074260851411065100284dee8d}	 
<input type="button" value="Create Flag"/>		

Drapeau: ZiTF{95f2d3074260851411065100284dee8d}

Seconde partie: Mettre un pied à Terre:





Fort de notre compte d'administrateur, il est maintenant souhaitable de récupérer un terminal sur le serveur distant, c'est plus utile, et en plus c'est marrant !

Cependant, nous avons déjà épuisé notre stock de CVE...

En ce cas, promenons-nous sur la plateforme avec nos nouveaux pouvoirs d'administrateurs tout-puissant.

Nous trouvons un point où changer l'image du CTf, mais si il y avait une possibilité de faire un reverse shell avec ça, ça serait déjà une cve...

C'est en allant du côté des pages, servant à faire les pages de garde interactives au ctf, que nous allons trouver des informations importantes: **Pages>All Pages**

CTFd Statistics Notifications Pages Users Scoreboard Challenges Submissions Config						
Pages +						
Title	Route	Authentication	Hidden	Published	Settings	
None	index			Published	 	
Infra	infra_038e11b72f95e23c74b7708e143c9d56	Required	Hidden	Published	 	

Nous voyons qu'il y a un deuxième format de page de garde, apparemment réservé à l'infra. Ce dernier semble même être caché, nécessitant d'être authentifié pour y accéder. En cliquant sur *Preview*, nous pouvons voir qu'elle ne contient qu'un petit message et un lien vers un zip qui contient la clé privée ssh liée au compte undercut_monkey



To all admin monkeys,

In order to maintain administration best practices, password SSH access to our infrastructure will be forbidden.

We will now all have to use our new SSH private key for the undercut_monkey user, available [here](#)

Best of luck, and may we crush the humans.

Sans étonnement, en tentant de dézipper le fichier, nous nous apercevons qu'il est doté d'un mot de passe.

```
chelinka@RUSTYCOMPUTER:~/Downloads$ unzip ssh_key_647032702ede7e519b1eb0279ba0ef99572e92
Archive:  ssh_key_647032702ede7e519b1eb0279ba0ef99572e92.zip
[ssh_key_647032702ede7e519b1eb0279ba0ef99572e92.zip] monkey_ssh password:
```

Nous allons procéder à une attaque par dictionnaire sur le fichier en utilisant le binaire fcrackzip avec la liste rockyou. Le mot de passe tombe presque tout de suite:

```
chelinka@RUSTYCOMPUTER:~/Downloads$ fcrackzip -v -u -D -p rockyou.txt ssh_key_647032702ede7e519b1eb0279ba0ef99572e92.zip
found file 'monkey_ssh', (size cp/uc 1993/ 2610, flags 9, chk 964b)

PASSWORD FOUND!!!!: pw == bonobo
chelinka@RUSTYCOMPUTER:~/Downloads$
```

Nous pouvons désormais déchiffrer le fichier et récupérer le clé ssh d'undercut_monkey, avec laquelle nous allons nous connecter au serveur distant. À l'aide du nmap fait précédemment, nous savons qu'il y a deux serveurs ssh. Un en 22 (classique), et un en 2222 (un peu moins classique). Nous pouvons nous connecter au second. De plus, les permissions de la clé privée extraite doivent être mises sur 600 pour des questions de sécurité, et surtout pour que le serveur distant l'accepte.

```
chelinka@RUSTYCOMPUTER:~/Downloads$ ssh -i monkey_ssh undercut_monkey@10.0.0.5
undercut_monkey@10.0.0.5: Permission denied (publickey).
chelinka@RUSTYCOMPUTER:~/Downloads$ ssh -i monkey_ssh undercut_monkey@10.0.0.5 -p 2222
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0755 for 'monkey_ssh' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "monkey_ssh": bad permissions
undercut_monkey@10.0.0.5's password:

chelinka@RUSTYCOMPUTER:~/Downloads$ chmod 600 monkey_ssh
chelinka@RUSTYCOMPUTER:~/Downloads$ ssh -i monkey_ssh undercut_monkey@10.0.0.5 -p 2222
Linux 2928229b5f9b 5.15.0-1030-gcp #37-Ubuntu SMP Tue Feb 14 19:37:08 UTC 2023 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 11 16:38:29 2023 from 10.0.0.2
undercut_monkey@2928229b5f9b:~$
```

Nous sommes maintenant connectés en tant qu'undercut_monkey !
Le drapeau se trouve dans le fichier flag.txt juste dans votre home:

```
undercut_monkey@2928229b5f9b:~$ ls
'GCONV_PATH=.'  PwnKit  flag.txt  linpeas.sh  pspy64
undercut_monkey@2928229b5f9b:~$ cat flag.txt
ZiTF{e845bdb8696955f35c44db42775523e8}
undercut_monkey@2928229b5f9b:~$
```

Drapeau: ZiTF{e845bdb8696955f35c44db42775523e8}