

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
INSTITUTO METRÓPOLE DIGITAL  
FORENSE COMPUTACIONAL

ATIVIDADE SEMANAL:  
BIND SHELL

Matheus Luna de Oliveira Dantas - 20190002737

Nesta atividade, iremos reproduzir um ataque de Backdoor Bind Shell. O termo shell se refere a uma classe geral de interpretadores de comandos baseados em texto, geralmente associados aos sistemas operacionais Unix e Linux.

- **Bind Shell:** instrui o computador-alvo a abrir um shell de comandos e ficar ouvindo uma porta local. Em outras palavras, a máquina do atacante atua como um cliente e a máquina da vítima atua como um servidor.
- **Reverse Shell:** força uma conexão de volta ao computador de ataque, ao invés de esperar uma conexão de entrada. Nesse caso, o nosso computador que funciona como um servidor.

Na atividade da semana, usaremos o Bind Shell, pois o nosso computador atua como cliente, e iremos obter os dados conforme formos dando os comandos.

A princípio, iremos identificar o processo que iremos analisar, com uma porta que não conseguimos reconhecer.

```
matheus@G3-3579:/tmp$ netstat -nlp
(Nem todos os processos puderam ser identificados, informações sobre processos
de outrem não serão mostrados, você deve ser root para vê-los todos.)
Conexões Internet Ativas (servidores e estabelecidas)
Proto Recv-Q Send-Q Endereço Local          Endereço Remoto      Estado      PID/Program name
tcp      0      0 127.0.0.53:53            0.0.0.0:*
tcp      0      0 0.0.0.0:22            0.0.0.0:*
tcp      0      0 127.0.0.1:631           0.0.0.0:*
tcp      0      0 127.0.0.1:5432           0.0.0.0:*
tcp      0      0 127.0.0.1:6463           0.0.0.0:*
tcp      0      0 0.0.0.0:41000          0.0.0.0:*
tcp      0      0 127.0.0.1:46289          0.0.0.0:*
tcp      0      0 192.168.1.9:52540          162.159.133.233:443 ESTABELECIDA 4398/Discord --type=
tcp      0      0 192.168.1.9:45108          151.101.92.133:443 ESTABELECIDA 3469/chrome --type=
tcp      0      0 192.168.1.9:44212          162.159.129.232:443 ESTABELECIDA 4398/Discord --type=
tcp      0      0 192.168.1.9:41168          209.95.56.146:444 ESTABELECIDA 3469/chrome --type=
tcp      0      0 192.168.1.9:56088          162.159.130.234:443 ESTABELECIDA 4398/Discord --type=
tcp      0      0 192.168.1.9:45982          162.159.134.232:443 ESTABELECIDA 4398/Discord --type=
tcp6     0      0 ::::22                  ::::*               OUÇA       -
tcp6     0      0 ::::1:631                ::::*               OUÇA       -
tcp6     0      0 ::::80                  ::::*               OUÇA       -
tcp6     0      0 2804:29b8:50de:59:57646 2600:9000:21e8:e000:443 ESTABELECIDA 3469/chrome --type=
tcp6     0      0 2804:29b8:50de:59:50716 2600:1901:1:e52:::443 ESTABELECIDA 4398/Discord --type=
tcp6     0      0 2804:29b8:50de:59:53510 2800:3f0:4003:c01:::5228 ESTABELECIDA 3469/chrome --type=
tcp6     0      0 2804:29b8:50de:59:34260 2600:9000:21e8:5600:443 ESTABELECIDA 3469/chrome --type=
tcp6     0      0 2804:29b8:50de:59:57658 2600:9000:21e8:e000:443 ESTABELECIDA 3469/chrome --type=
```

Aqui tivemos acesso aos processos que estão rodando, e iremos atrás do PID (Process ID) do programa “freedom”, que no caso foi 13714.

```
matheus@G3-3579:/tmp$ ls -al /proc/13714
total 0
dr-xr-xr-x  9 matheus matheus 0 out 20 16:19 .
dr-xr-xr-x 318 root      root    0 out 20 14:55 ..
-r--r--r--  1 matheus matheus 0 out 20 16:25 arch_status
dr-xr-xr-x  2 matheus matheus 0 out 20 16:20 attr
-rw-r--r--  1 matheus matheus 0 out 20 16:25 autogroup
-r-----  1 matheus matheus 0 out 20 16:25 auxv
-r--r--r--  1 matheus matheus 0 out 20 16:25 cgroup
--w-----  1 matheus matheus 0 out 20 16:25 clear_refs
-r--r--r--  1 matheus matheus 0 out 20 16:19 cmdline
-rw-r--r--  1 matheus matheus 0 out 20 16:25 comm
-rw-r--r--  1 matheus matheus 0 out 20 16:25 coredump_filter
-r--r--r--  1 matheus matheus 0 out 20 16:25 cpuset
lrwxrwxrwx  1 matheus matheus 0 out 20 16:22 cwd -> /tmp
-r-----  1 matheus matheus 0 out 20 16:25 environ
lrwxrwxrwx  1 matheus matheus 0 out 20 16:19 exe -> '/tmp/freedom (deleted)'
dr-x-----  2 matheus matheus 0 out 20 16:20 fd
dr-x-----  2 matheus matheus 0 out 20 16:25 fdinfo
-rw-r--r--  1 matheus matheus 0 out 20 16:25 gid_map
-r-----  1 matheus matheus 0 out 20 16:25 io
-r--r--r--  1 matheus matheus 0 out 20 16:25 limits
-rw-r--r--  1 matheus matheus 0 out 20 16:25 loginuid
dr-x-----  2 matheus matheus 0 out 20 16:25 map_files
-r--r--r--  1 matheus matheus 0 out 20 16:22 maps
-rw-----  1 matheus matheus 0 out 20 16:25 mem
-r--r--r--  1 matheus matheus 0 out 20 16:25 mountinfo
-r--r--r--  1 matheus matheus 0 out 20 16:25 mounts
-r-----  1 matheus matheus 0 out 20 16:25 mountstats
dr-xr-xr-x  5 matheus matheus 0 out 20 16:25 net
dr-xr-x--x  2 matheus matheus 0 out 20 16:25 ns
-r--r--r--  1 matheus matheus 0 out 20 16:25 numa_maps
-rw-r--r--  1 matheus matheus 0 out 20 16:25 oom_adj
-r--r--r--  1 matheus matheus 0 out 20 16:25 oom_score
-rw-r--r--  1 matheus matheus 0 out 20 16:25 oom_score_adj
-r-----  1 matheus matheus 0 out 20 16:25 pagemap
-r-----  1 matheus matheus 0 out 20 16:25 patch_state
-r-----  1 matheus matheus 0 out 20 16:25 personality
-rw-r--r--  1 matheus matheus 0 out 20 16:25 projid_map
lrwxrwxrwx  1 matheus matheus 0 out 20 16:22 root -> /
```

Aqui nós analisamos o processo desejado com o comando "ls -al /proc/13714", e observamos que o diretório em que o processo estava rodando era o /tmp, e que o binário estava em /tmp, mas foi deletado. Pelo processo estar em /tmp já é suspeito, pois é onde a maioria dos malwares jogam seus payloads para rodar lá.

Para recuperar o binário que foi deletado, iremos utilizar o comando "cp /proc/13714/exe /tmp/recovered\_bin". Após recuperar o binário, iremos rodar um hash no netcat e o binário recuperado para verificar se são os mesmos.

```
matheus@G3-3579:/tmp$ sha1sum /bin/nc
142391ab131af2520a0e4a1622643dbfd3057d52  /bin/nc

matheus@G3-3579:/tmp$ sha1sum /tmp/recovered_bin
142391ab131af2520a0e4a1622643dbfd3057d52  /tmp/recovered_bin
```

Podemos ver que ambos possuem de fato o mesmo hash, que é "142391ab131af2520a0e4a1622643dbfd3057d52".

Agora, iremos verificar as linhas de comando do malware com os comandos: "cat /proc/13714/comm" e "cat /proc/13714/cmdline".

```

matheus@G3-3579:/tmp$ cat /proc/13714/comm
freedom
matheus@G3-3579:/tmp$ cat /proc/13714/cmdline
./freedom-k-w1-l41000matheus@G3-3579:/tmp$ strings /proc/13714/environ
SHELL=/bin/bash
SESSION_MANAGER=local/G3-3579:@/tmp/.ICE-unix/1987,unix/G3-3579:/tmp/.ICE-unix/1987
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
rvm_prefix=/home/matheus
LANGUAGE=pt_BR:en
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
MY_RUBY_HOME=/home/matheus/.rvm/rubies/ruby-2.6.6
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1946
GTK_MODULES=galli:atk-bridge
RUBY_VERSION=ruby-2.6.6
PWD=/tmp
LOGNAME=matheus
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
rvm_version=1.29.10 (latest)
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
WINDOWPATH=2
HOME=/home/matheus
USERNAME=matheus
IM_CONFIG_PHASE=1
LANG=pt_BR.UTF-8
LS_COLORS=r=0;di=0;34;ln=0;36;mh=0;pi=40;33;so=0;35;do=0;35;bd=40;33;01;cd=40;33;01;r=40;31;01;ml=00;su=37;41;sg=30;43;ca=30;41;tw=30
1;*.jar=0;31;*.tar=0;31;*.lha=0;31;*.lz4=0;31;*.lzh=0;31;*.lzma=0;31;*.tlz=0;31;*.txz=0;31;*.tzo=0;31;*.tz=0;31;*.z=0
z=0;31;*.zst=0;31;*.tzst=0;31;*.bz=0;31;*.bz2=0;31;*.tbz=0;31;*.tbz2=0;31;*.tz=0;31;*.deb=0;31;*.rpm=0;31;*.jar=0;31;*.war=0;31;
=0;31;*.cpio=0;31;*.7z=0;31;*.rz=0;31;*.cab=0;31;*.wim=0;31;*.swm=0;31;*.dwm=0;31;*.esd=0;31;*.jpg=0;31;*.jpeg=0;31;*.mjpg=0;31;
pm=0;31;*.tga=0;31;*.xbm=0;31;*.xpm=0;31;*.tif=0;31;*.tiff=0;31;*.png=0;31;*.svg=0;31;*.svgs=0;31;*.mng=0;31;*.pcx=0;31;*.mov=0;31;
*.ogn=0;31;*.mp4=0;31;*.m4v=0;31;*.mp4v=0;31;*.vob=0;31;*.qt=0;31;*.nuv=0;31;*.wmv=0;31;*.asf=0;31;*.rm=0;31;*.rmvb=0;31;*.flc=0;31;
cf=0;31;*.xwd=0;31;*.yuv=0;31;*.cgm=0;31;*.emf=0;31;*.oqv=0;31;*.ogx=0;31;*.aac=0;36;*.au=0;36;*.flac=0;36;*.m4a=0;36;*.mid=0;36;
a=0;36;*.wav=0;36;*.oga=0;36;*.opus=0;36;*.spx=0;36;*.xspf=0;36;
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=0003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/e14cf4fe_e851_463d_8b05_0a52d85000aa
INVOCATION_ID=fcc916cef1efb4627bc583374177197e6
MANAGERPID=1678
rvm_bin_path=/home/matheus/.rvm/bin
GEM_PATH=/home/matheus/.rvm/gems/ruby-2.6.6@metasploit-framework:/home/matheus/.rvm/gems/ruby-2.6.6@global
GEM_HOME=/home/matheus/.rvm/gems/ruby-2.6.6@metasploit-framework
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color

```

Um malware provavelmente terá vários nomes diferentes. Foi utilizado também o comando “strings /proc/13714/environ”, para tentar identificar quem iniciou o processo, e quando foi iniciado.

```

matheus@G3-3579:/tmp$ ls -al /proc/13714/fd
total 0
dr-x----- 2 matheus matheus 0 out 20 16:20 .
dr-xr-xr-x 9 matheus matheus 0 out 20 16:19 ..
lrwx----- 1 matheus matheus 64 out 20 16:20 0 -> /dev/pts/0
l-wx----- 1 matheus matheus 64 out 20 16:20 1 -> /dev/null
lrwx----- 1 matheus matheus 64 out 20 16:20 2 -> /dev/pts/0
lrwx----- 1 matheus matheus 64 out 20 16:20 3 -> 'socket:[102305]'
lrwx----- 1 matheus matheus 64 out 20 16:20 6 -> /dev/pts/0

```

Utilizado o comando “ls -al /proc/13714/fd” para verificar os arquivos que o malware está utilizando.

```

matheus@G3-3579:/tmp$ cat /proc/13714/maps
55ac53502000-55ac53504000 r--p 00000000 08:06 657397
55ac53504000-55ac53509000 r-xp 00002000 08:06 657397
55ac53509000-55ac5350b000 r--p 00007000 08:06 657397
55ac5350c000-55ac5350d000 r--p 00009000 08:06 657397
55ac5350d000-55ac5350e000 rw-p 0000a000 08:06 657397
55ac5350e000-55ac5350e000 rw-p 00000000 00:00 0
55ac53bef000-55ac53c10000 rw-p 00000000 00:00 0
7f4bee747000-7f4bee74a000 rw-p 00000000 00:00 0
7f4bee74a000-7f4bee76f000 r--p 00000000 08:06 2099337
7f4bee76f000-7f4bee8e7000 r-xp 00025000 08:06 2099337
7f4bee8e7000-7f4bee931000 r--p 0019d000 08:06 2099337
7f4bee931000-7f4bee932000 ---p 001e7000 08:06 2099337
7f4bee932000-7f4bee935000 r--p 001e7000 08:06 2099337
7f4bee935000-7f4bee938000 rw-p 001ea000 08:06 2099337
7f4bee938000-7f4bee93c000 rw-p 00000000 00:00 0
7f4bee93c000-7f4bee940000 r--p 00000000 08:06 2099355
7f4bee940000-7f4bee950000 r--p 00004000 08:06 2099355
7f4bee950000-7f4bee953000 r--p 00014000 08:06 2099355
7f4bee953000-7f4bee954000 ---p 00017000 08:06 2099355
7f4bee954000-7f4bee955000 r--p 00017000 08:06 2099355
7f4bee955000-7f4bee956000 rw-p 00018000 08:06 2099355
7f4bee956000-7f4bee958000 rw-p 00000000 00:00 0
7f4bee958000-7f4bee95c000 r--p 00000000 08:06 2104535
7f4bee95c000-7f4bee96b000 r-xp 00004000 08:06 2104535
7f4bee96b000-7f4bee96e000 r--p 00013000 08:06 2104535
7f4bee96e000-7f4bee96f000 ---p 00016000 08:06 2104535
7f4bee96f000-7f4bee970000 r--p 00016000 08:06 2104535
7f4bee970000-7f4bee971000 rw-p 00017000 08:06 2104535
7f4bee971000-7f4bee974000 rw-p 00000000 00:00 0
7f4bee986000-7f4bee987000 r--p 00000000 08:06 2099333
7f4bee987000-7f4bee9aa000 r-xp 00001000 08:06 2099333
7f4bee9aa000-7f4bee9b2000 r--p 00024000 08:06 2099333
7f4bee9b3000-7f4bee9b4000 r--p 0002c000 08:06 2099333
7f4bee9b4000-7f4bee9b5000 rw-p 0002d000 08:06 2099333
7f4bee9b5000-7f4bee9b6000 rw-p 00000000 00:00 0
7ffe114ef000-7ffe11510000 rw-p 00000000 00:00 0
7ffe1158c000-7ffe1158f000 r--p 00000000 00:00 0
7ffe1158f000-7ffe11590000 r-xp 00000000 00:00 0
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0
matheus@G3-3579:/tmp$ cat /proc/13714/stack
cat: /proc/13714/stack: Permissão negada

```

Com o “cat /proc/13714/maps” podemos ver as bibliotecas que o malware está utilizando para sua execução. O “cat /proc/13714/stack” nos daria mais detalhes, só que deu permissão negada.

Por fim, iremos usar o “cat /proc/13714/status” para obter um detalhamento melhor do processo em si.

O intuito era saber identificar processos que poderiam ser maliciosos, para que não matemos um processo que seja importante erroneamente, por apenas acharmos que é um malware.