**Cybersecurity 401**

**Module 7 - Threat Hunting**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Ops 35 - Web Vulnerability Scanning with Nmap**

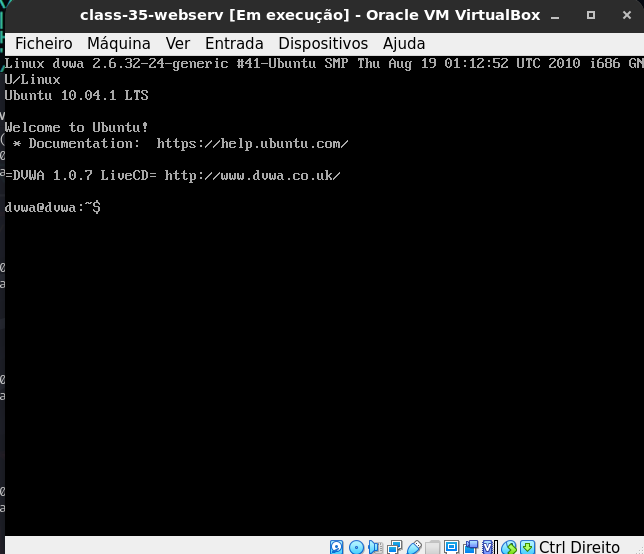
## 

**| Rodrigo Brasil 12/2023 |**

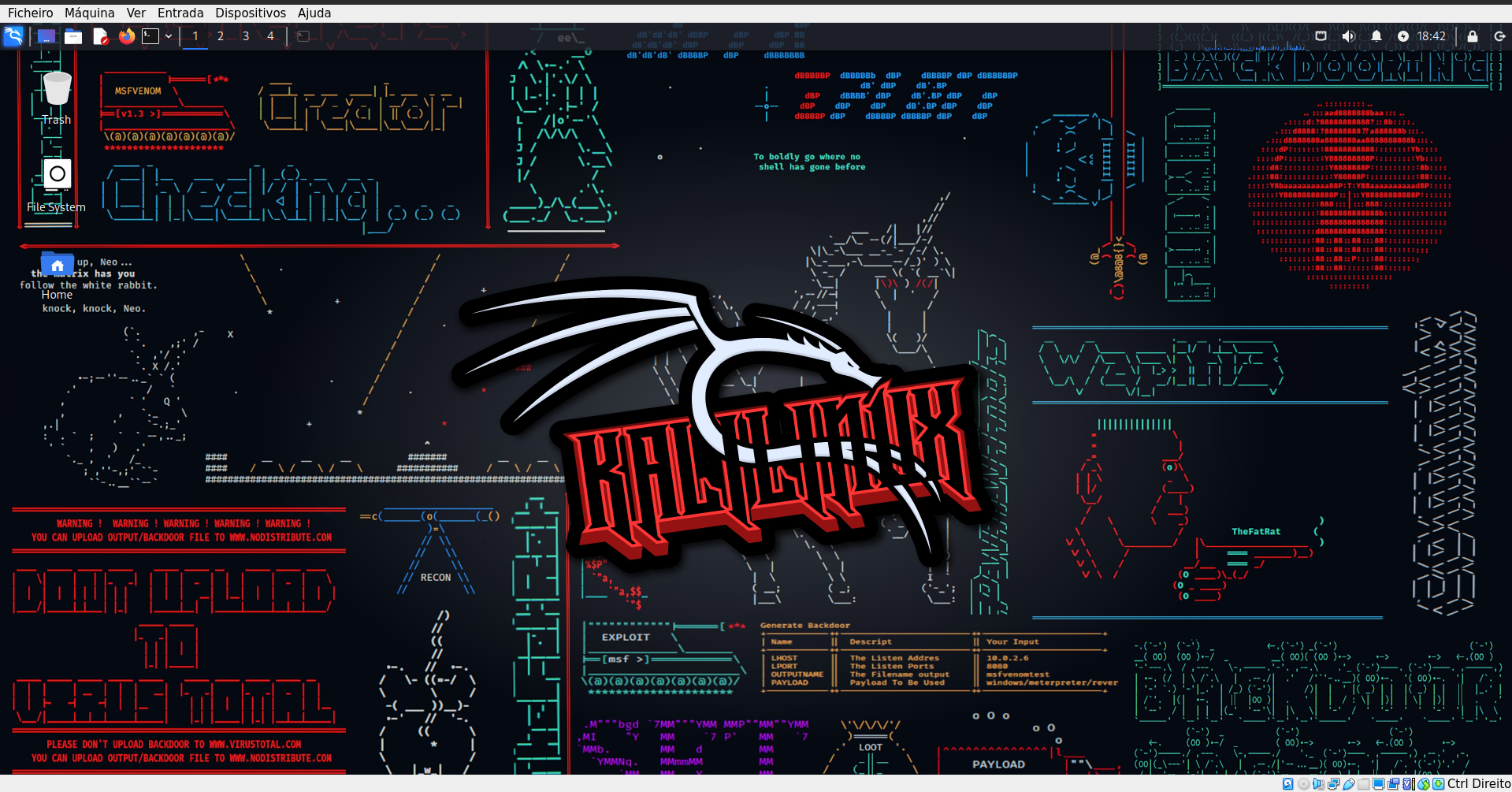
## **Part 1: Staging**

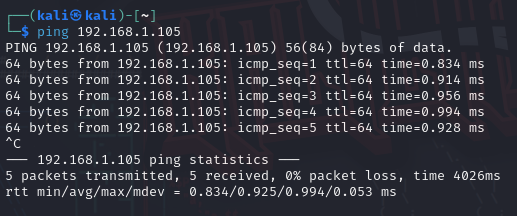
Your instructor will provide the class an OVA of the target system.

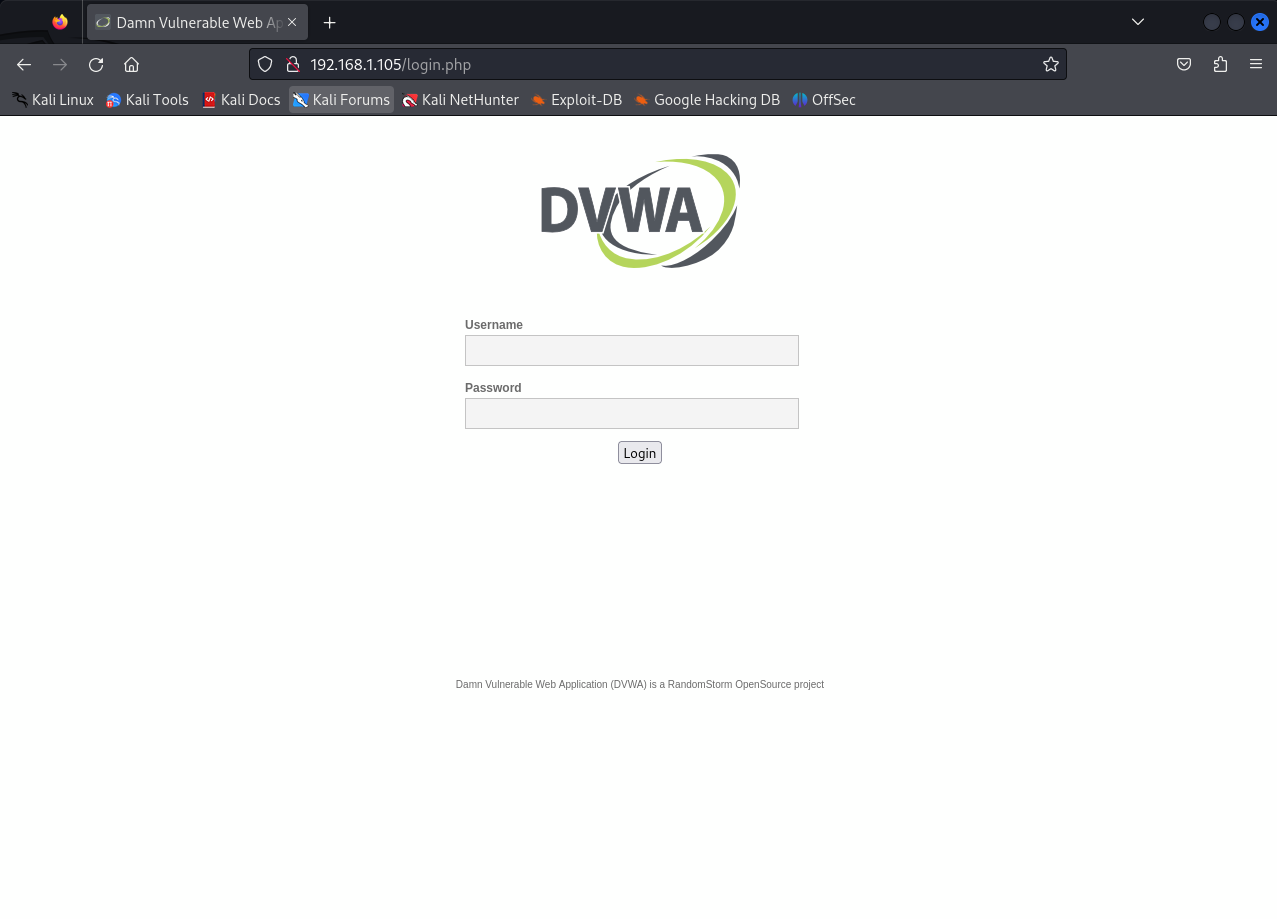
* Download and import the “class-35-webserv” OVA, insert the DVWA ISO, then boot from the ISO. Set it to NAT Network.



* Startup Kali Linux VM on the same NAT Network. Check that you can ping the web server and view its hosted page via browser.





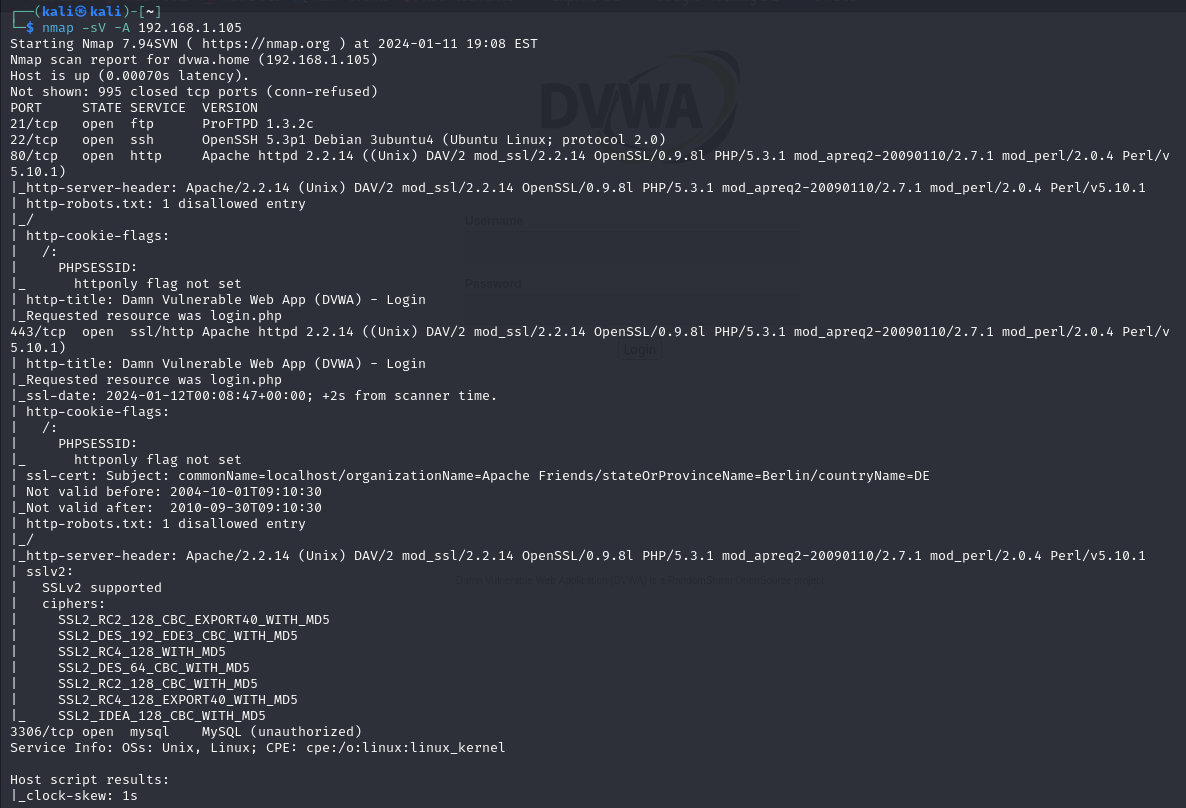


## 

## **Part 2: Basic Nmap Scans**

Let’s first practice the fundamental Nmap commands from the terminal against the target web server.

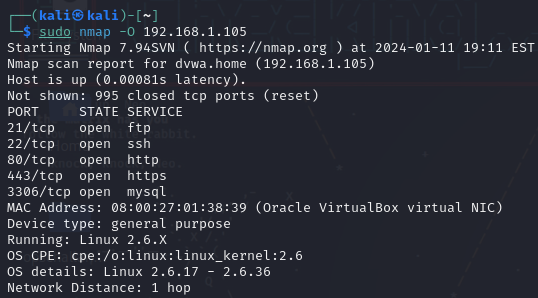
* Perform the highest intensity service and version scan against the target. What command did you use, and what was the outcome?



I used the command “nmap -sV -A 192.168.1.105” to make the aggressive service scan.

It gave me more details about the services running compared to a more passive scan

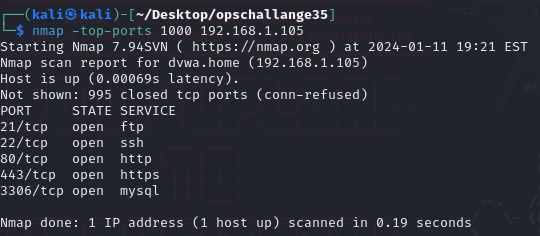
* Attempt to detect the operating system of the target. Do you think the findings are correct?





It did identify it is using a linux OS using the kernel version 2.6

* Have Nmap perform a port scan of only the well-known port numbers. What command did you use?



I used the command “nmap -top-ports 1000 192.168.1.105”

Now that you can perform basic Nmap scans without the aid of Zenmap, let’s move on to something much more powerful: NSE.

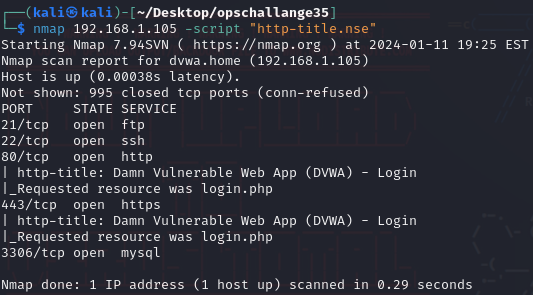
## 

## **Part 3: Scanning with the Nmap Scripting Engine (NSE)**

The NSE allows us to extend Nmap’s capabilities far beyond a mere port scan. Much like how YARA rules allow for customized detection rules, the NSE makes Nmap an extensible cyber scanning swiss army knife, capable of quickly finding vulnerabilities on a web application.

The http-title.nse script lets us grab the title of a web page from the web server.

* Using Nmap in NSE mode, execute http-title.nse against the target.



* What did you learn about the target web server?
* I learned that the http title is “Damn Vulnerable Web App (DVWA) - Login” and the resource came from login.php

The script “HTTP enumerator” aggressively attempts to view the files and directories on the target web server.

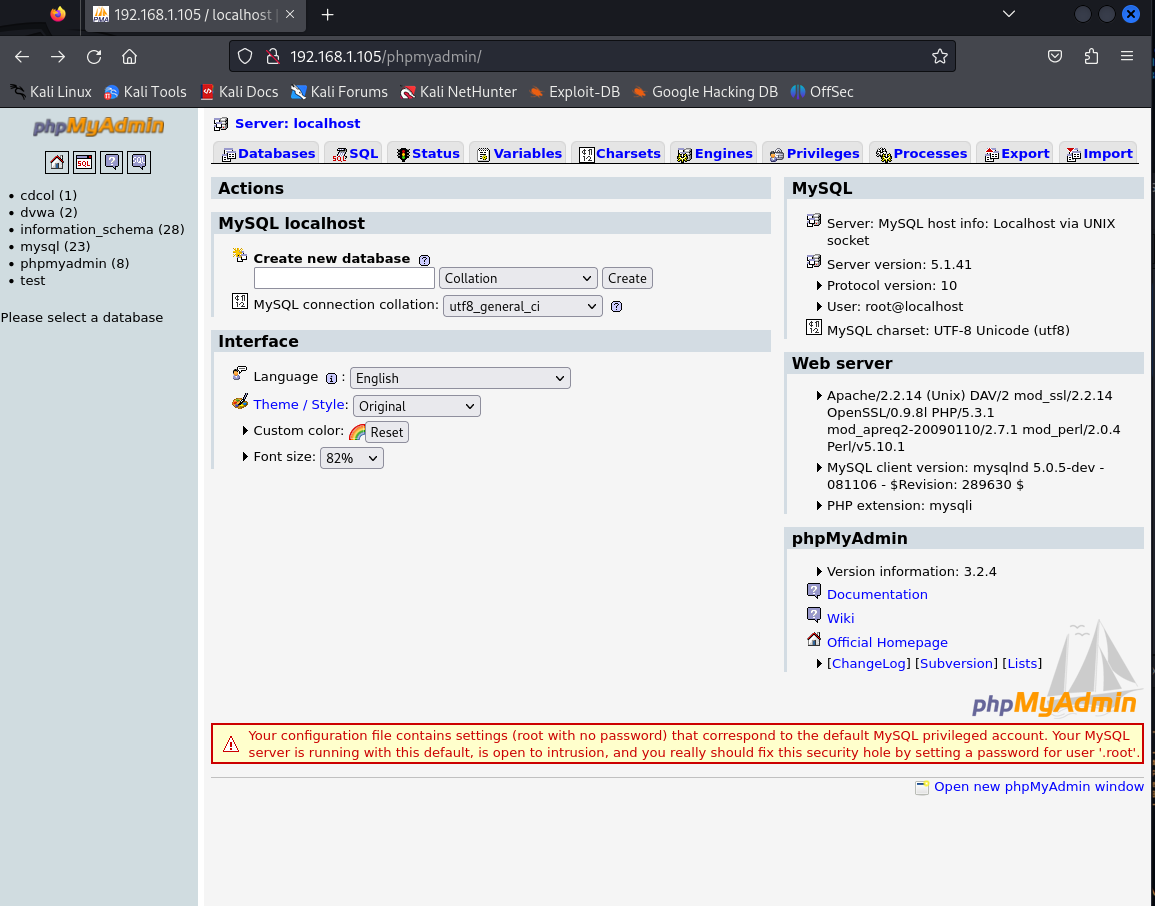
* Using Nmap in NSE mode, execute http-enum.nse against the target. Post the output in your submission doc.

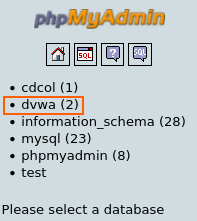


* From Mozilla Firefox in Kali Linux, access some of the enumerated files. What did you find out about this web server?

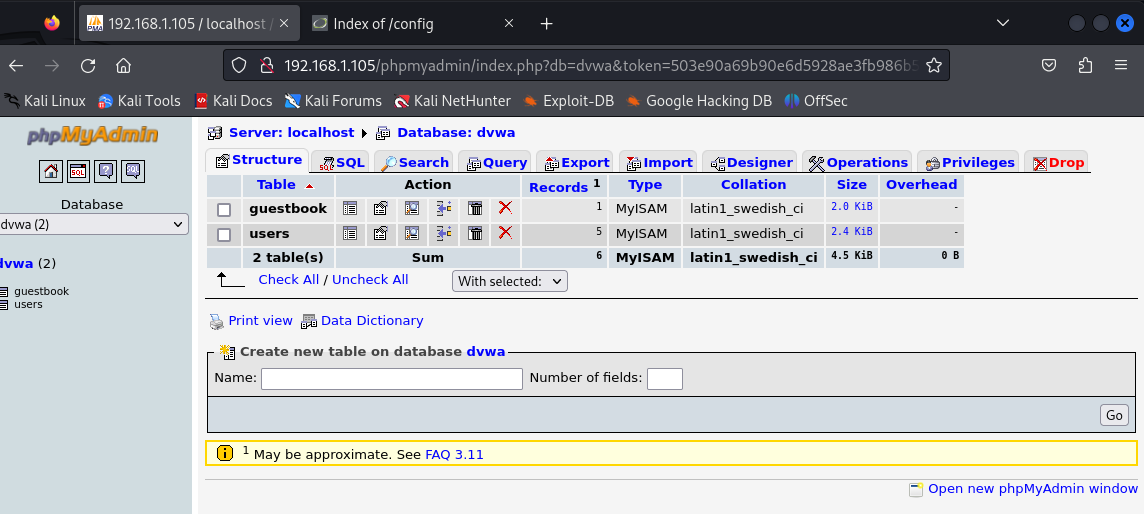
#### /phpmyadmin

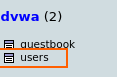
the admin php site. Inside it we can search the database for users



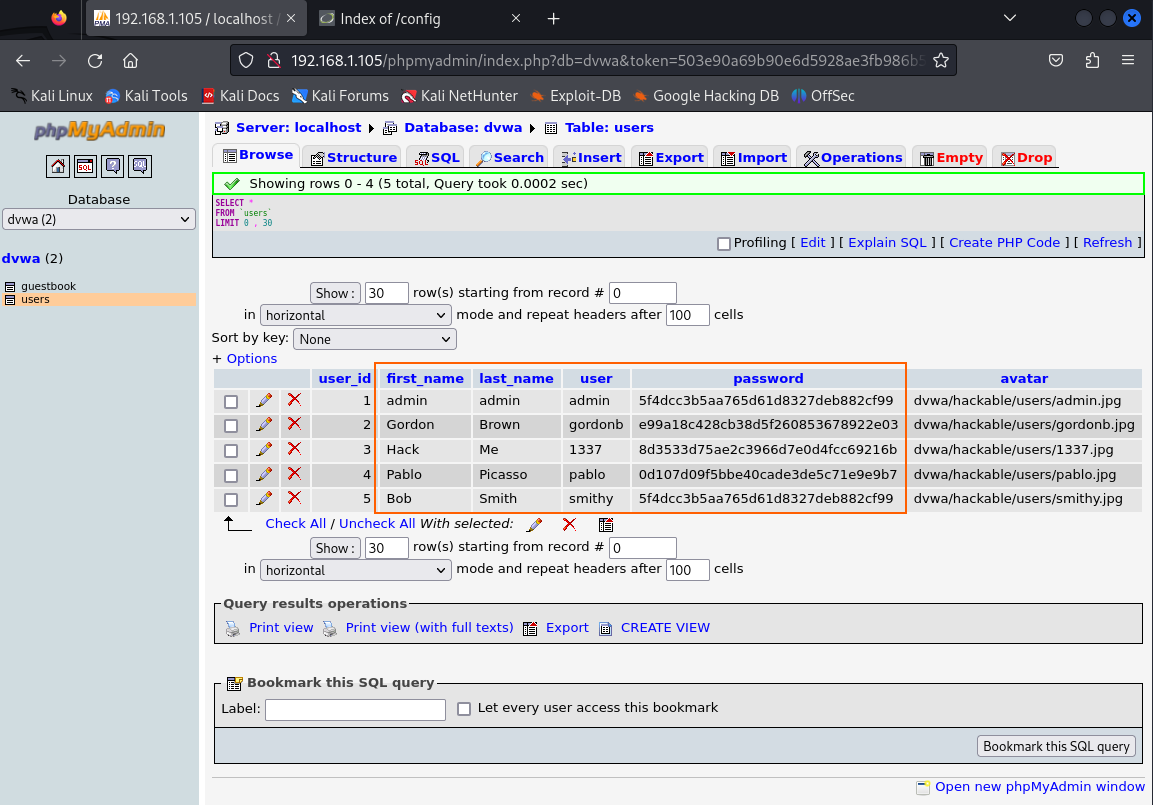


because our site is called dvwa we can assume this is the dvwa database

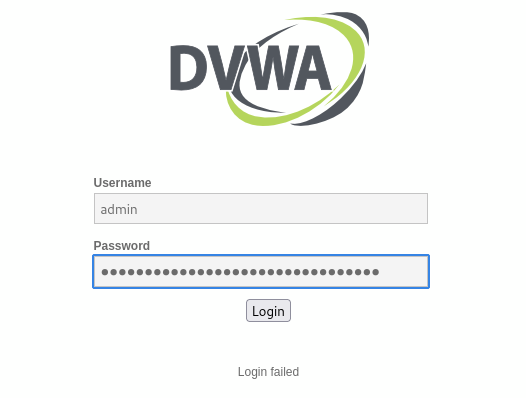




Inside it we can see there is users



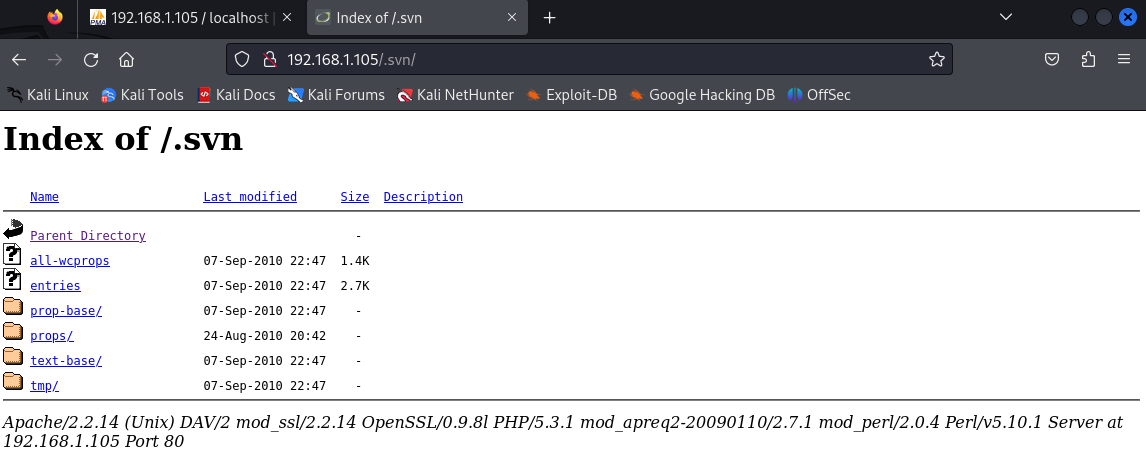
Inside it we can see the usernames but the passwords look like they are hashed



Trying to login using the admin username and the password gives a login fail because it is the password hashed, not in clear text

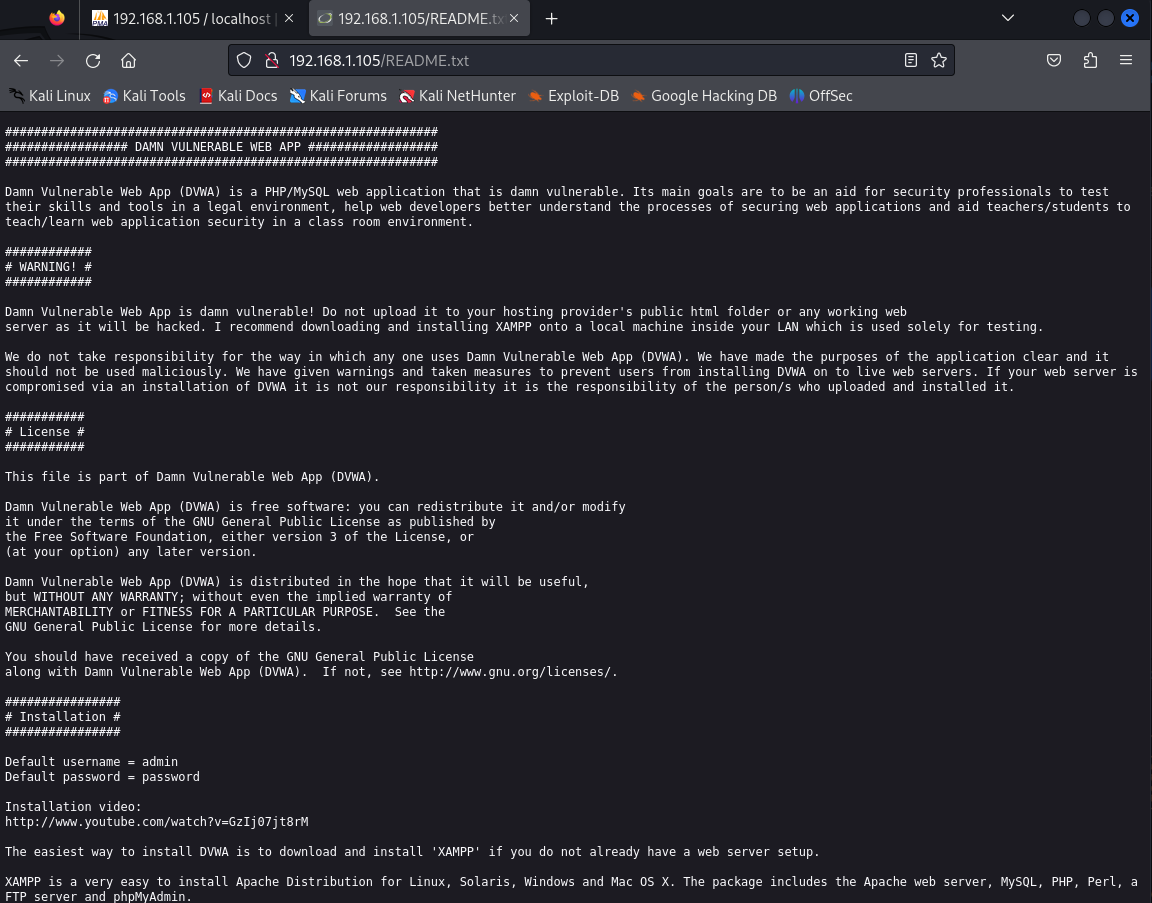
#### /.svn

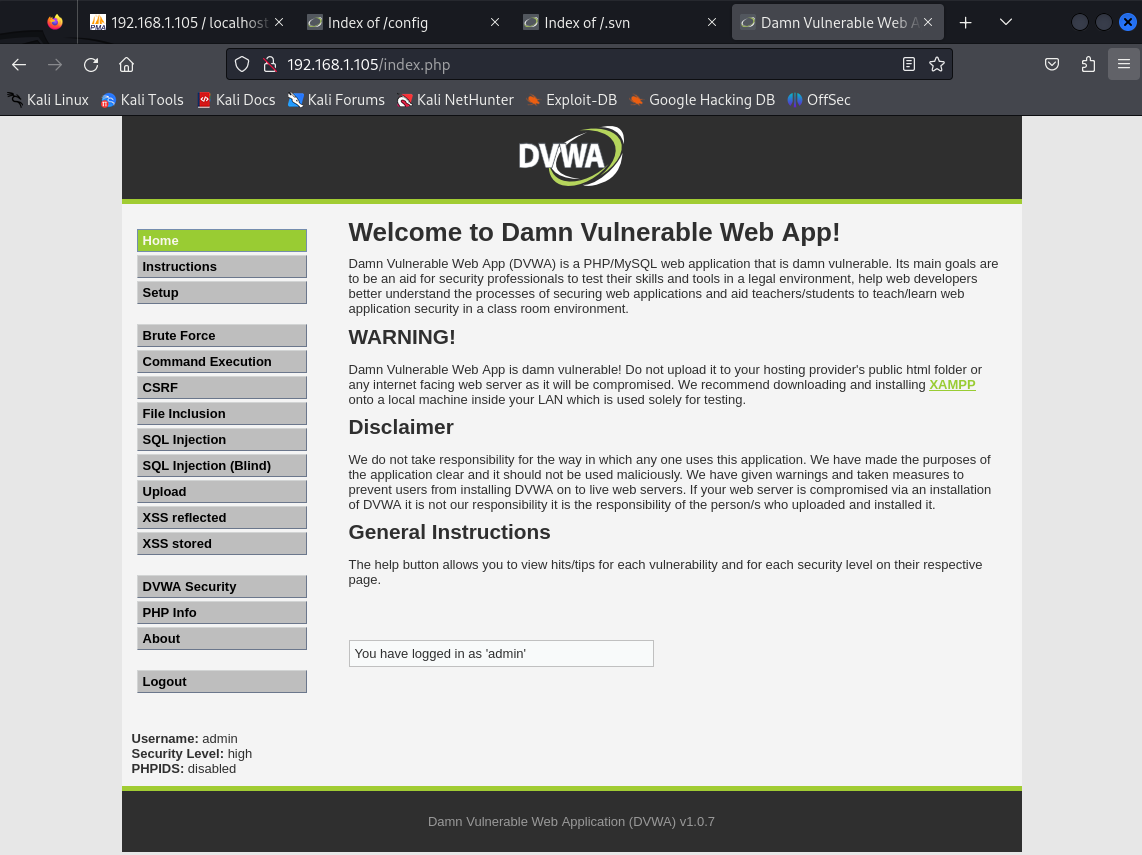
looks like where some files are stored, but I dug in a bit and didn't find anything useful, most of them show nothing.



#### /.README.txt

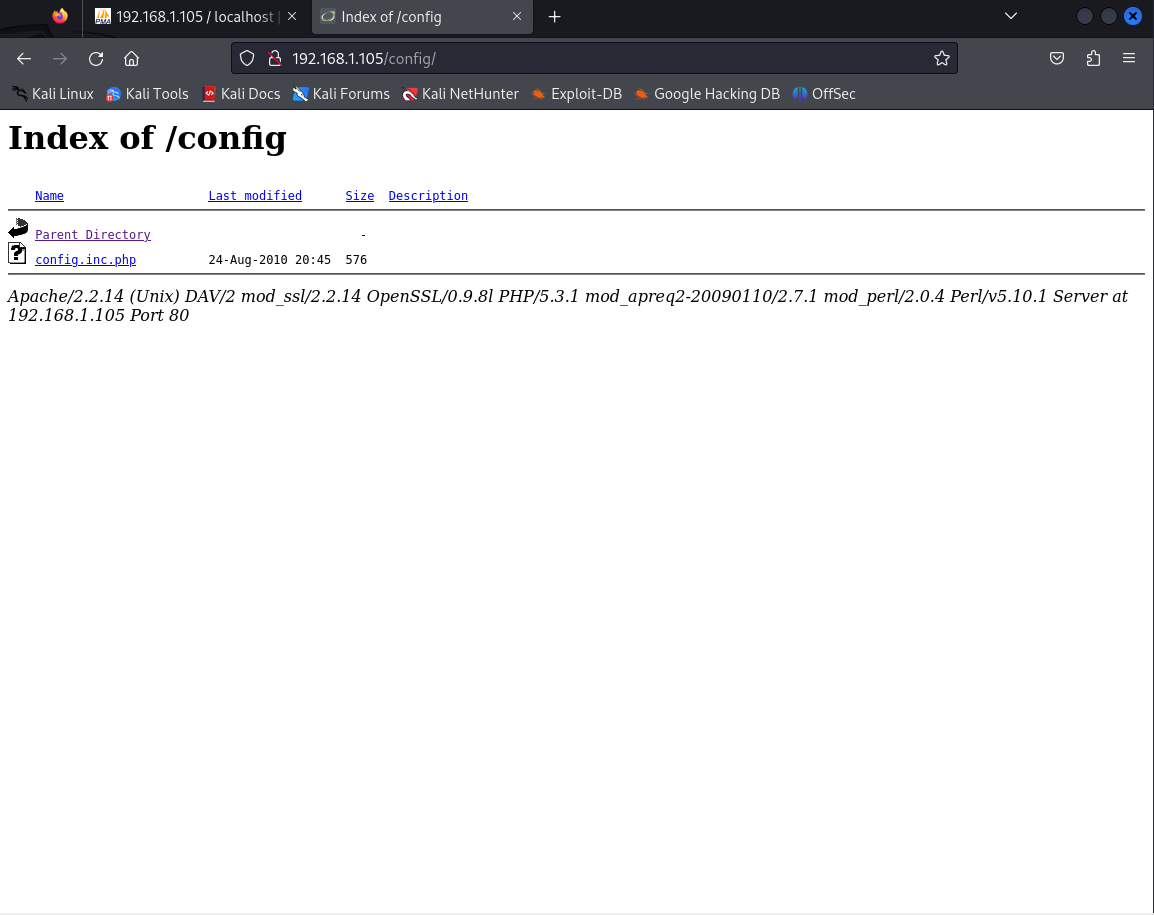
inside it was talking about the purpose of the web server. Also had the username and password in clear text which made it possible to login into page

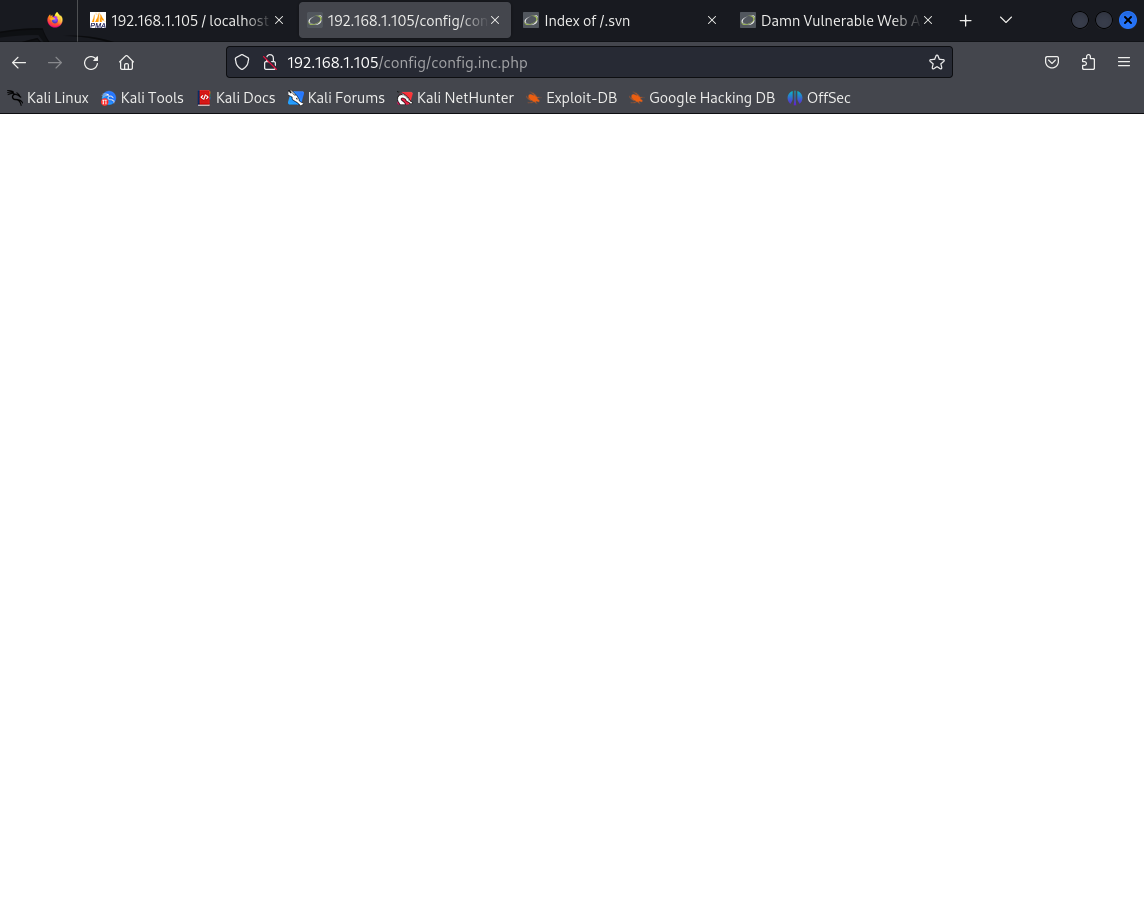




#### /config

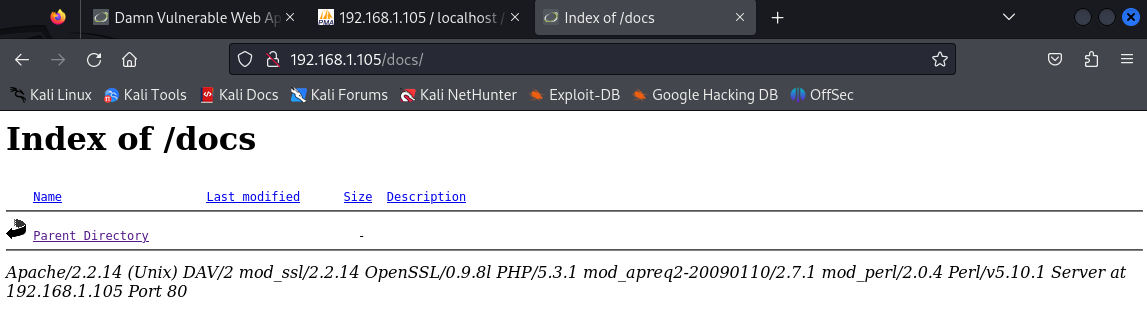
had one file in it and when opening showed a blank page





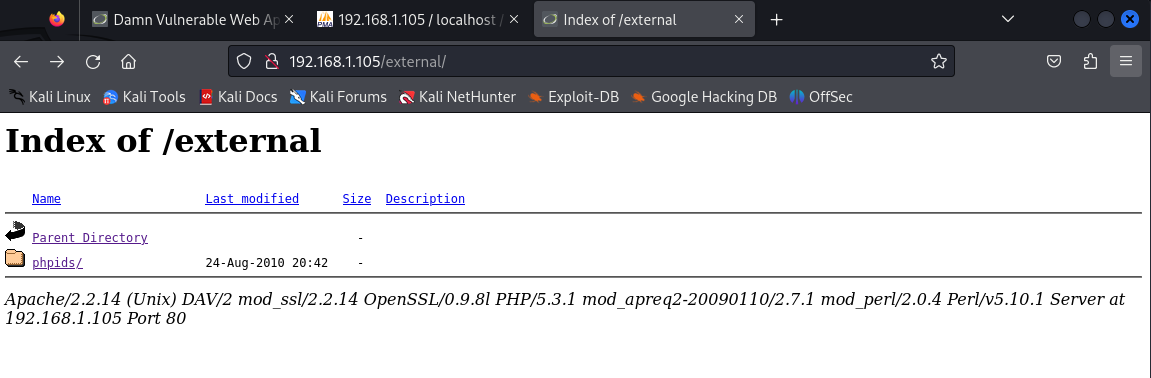
#### /docs

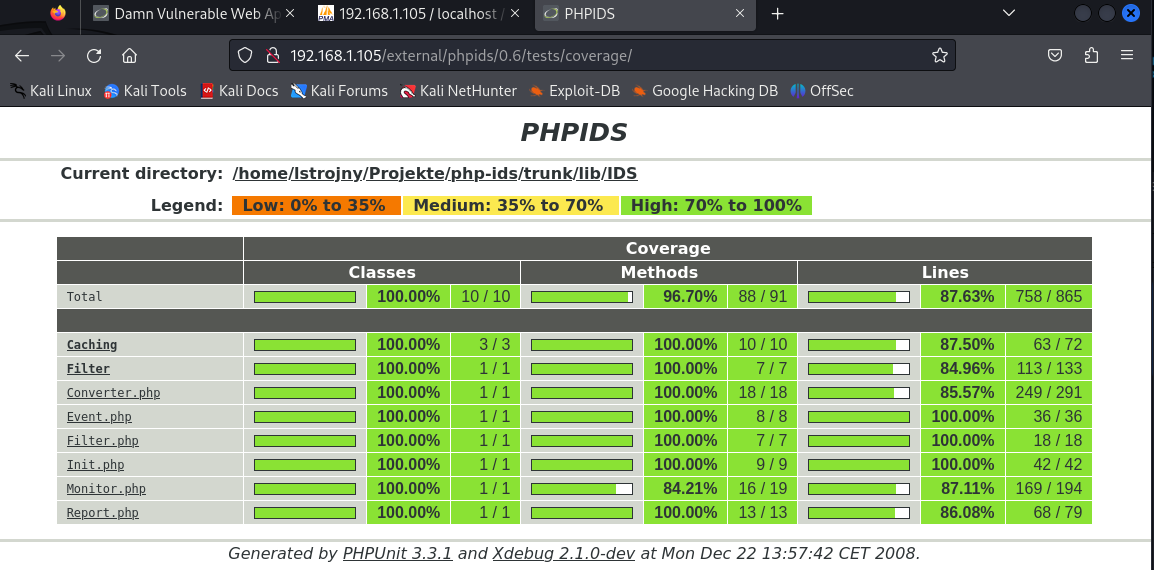
had nothing inside it



#### /externals

not sure what this is, assuming something about php because found a chart called PHPIDS





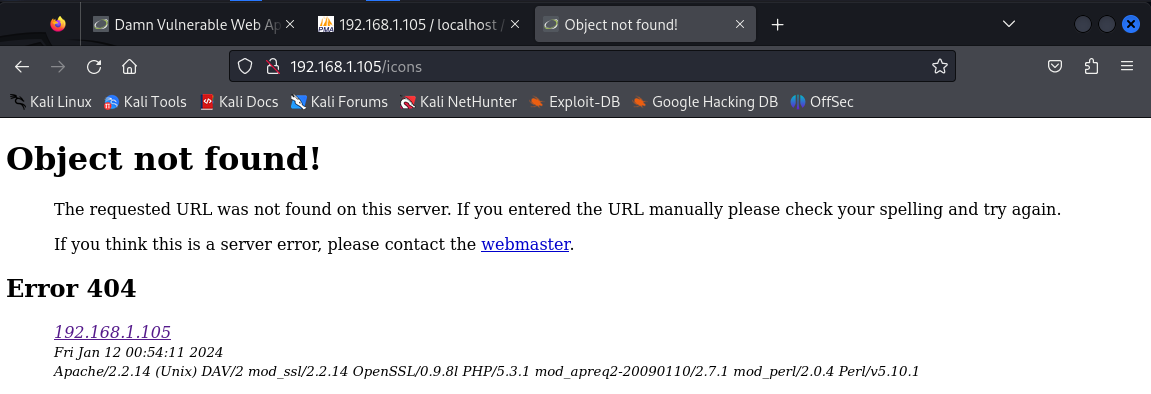
#### 

#### 

#### 

#### /icons

gave me an error 404, couldn't find the page



#### 

#### 

#### 

#### 

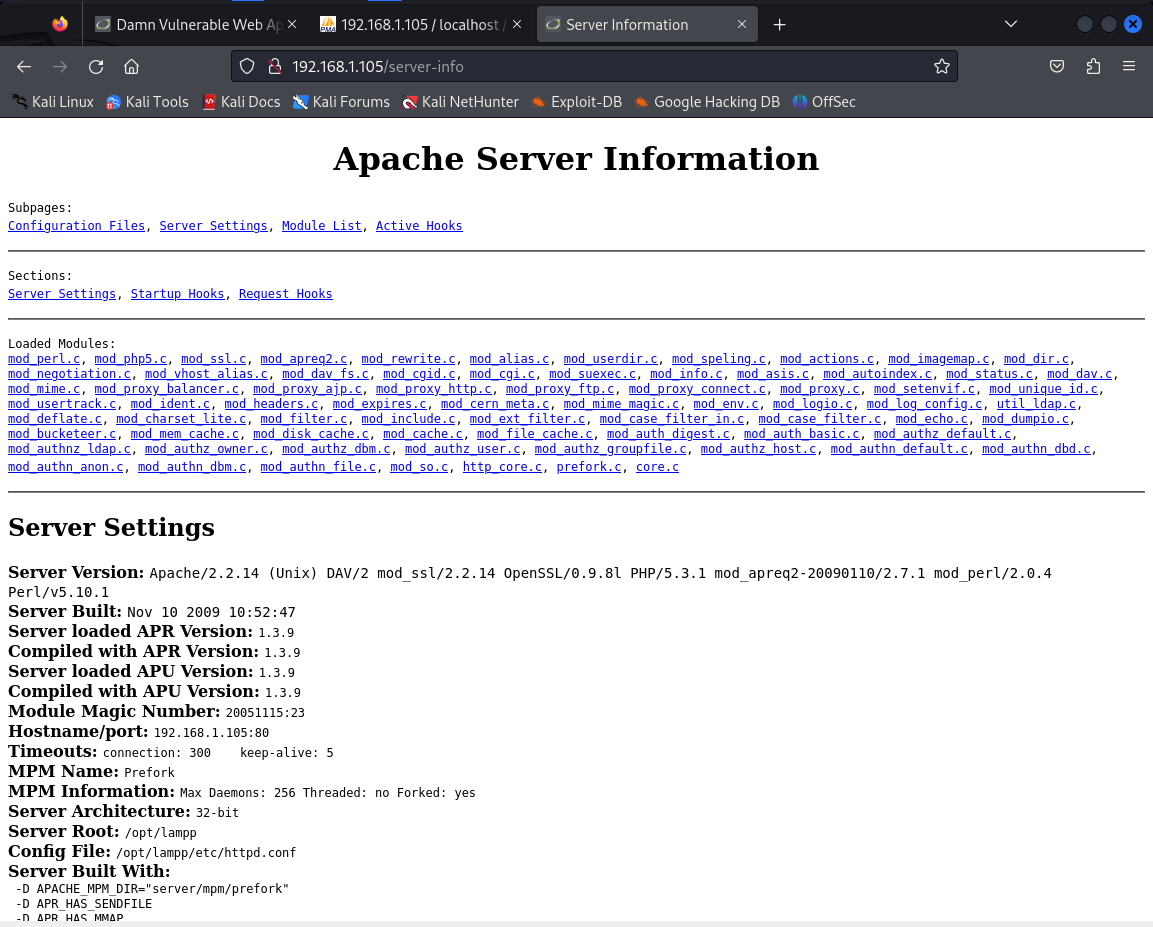
#### 

#### 

#### 

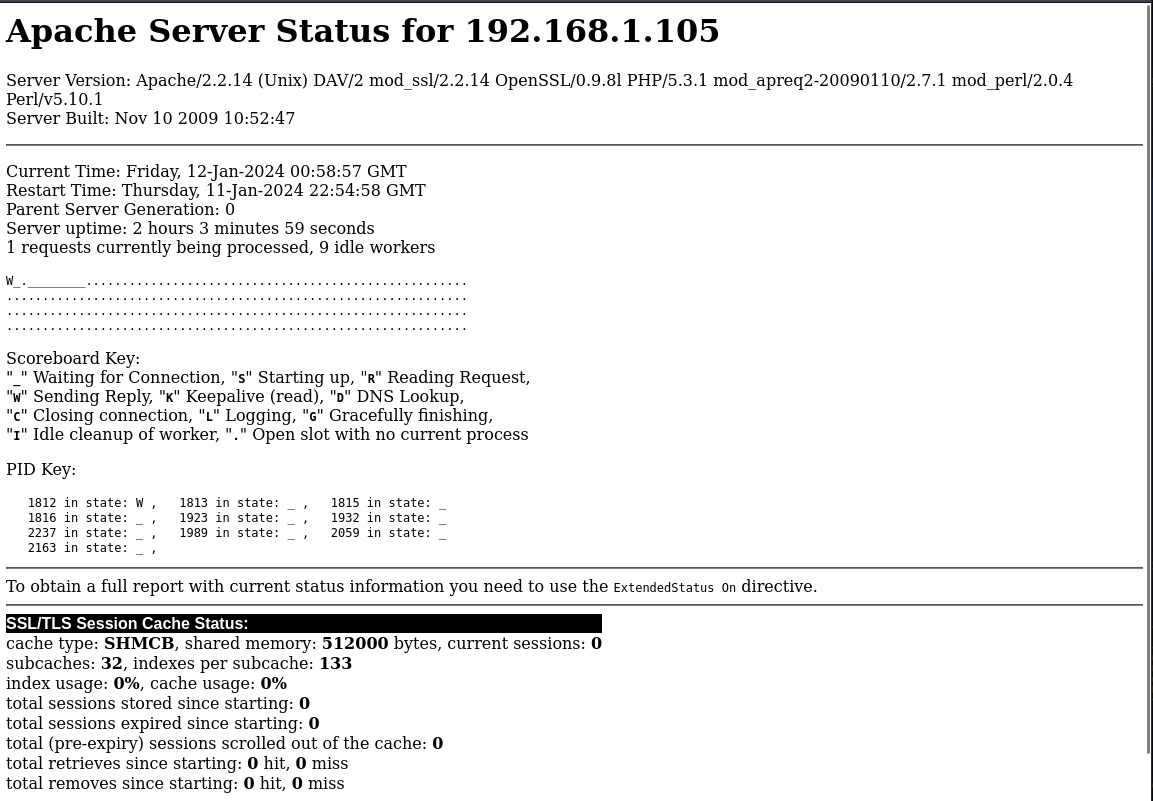
#### /server-info

is very interesting page, has a lot of information about the server running, like the apache version, the machine ip and what port its running on, when it was built, etc



#### /server-status

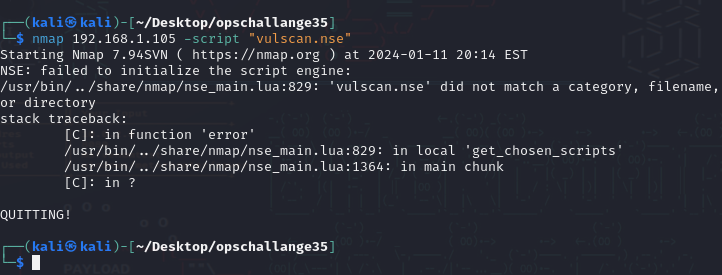
shows the status of the server, the date it was built, how long it has been up, the timezone it is on, the last time it was restarted, etc



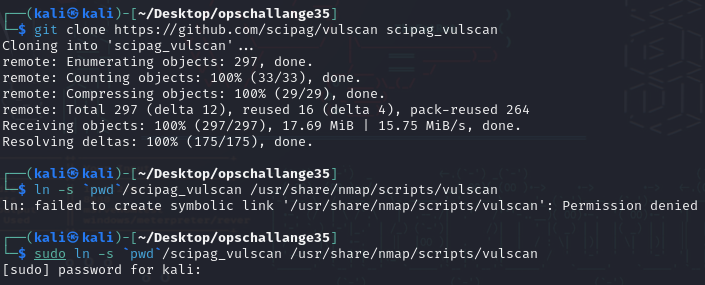
*You’d be surprised how much can be learned by merely browsing the files in web server directories! Many tools exist to enumerate web server directories, including Dirbuster and Burp Suite to name a few. Keep this technique in your pocket the next time you’re playing HackTheBox. A useful list of web enumeration tools can be found at* [*HrushikeshK’s blog*](https://hkh4cks.com/blog/2018/01/22/common-enumeration-tools/)*.*

This is a great start, but let’s also try detecting documented vulnerabilities. This next technique can be used defensively to determine how secure your web app is by checking for known vulnerabilities in communal databases (think VirusTotal, but for vulnerabilities). “Vulscan” queries CVE databases to produce a very thorough list of issues.

* Using Nmap in NSE mode, execute vulscan.nse against the target web server and save the output to a file for easy viewing.



gave an error, forgot to download and install it from the github repo <https://scipag.github.io/vulscan/>



Installation



command used



Output

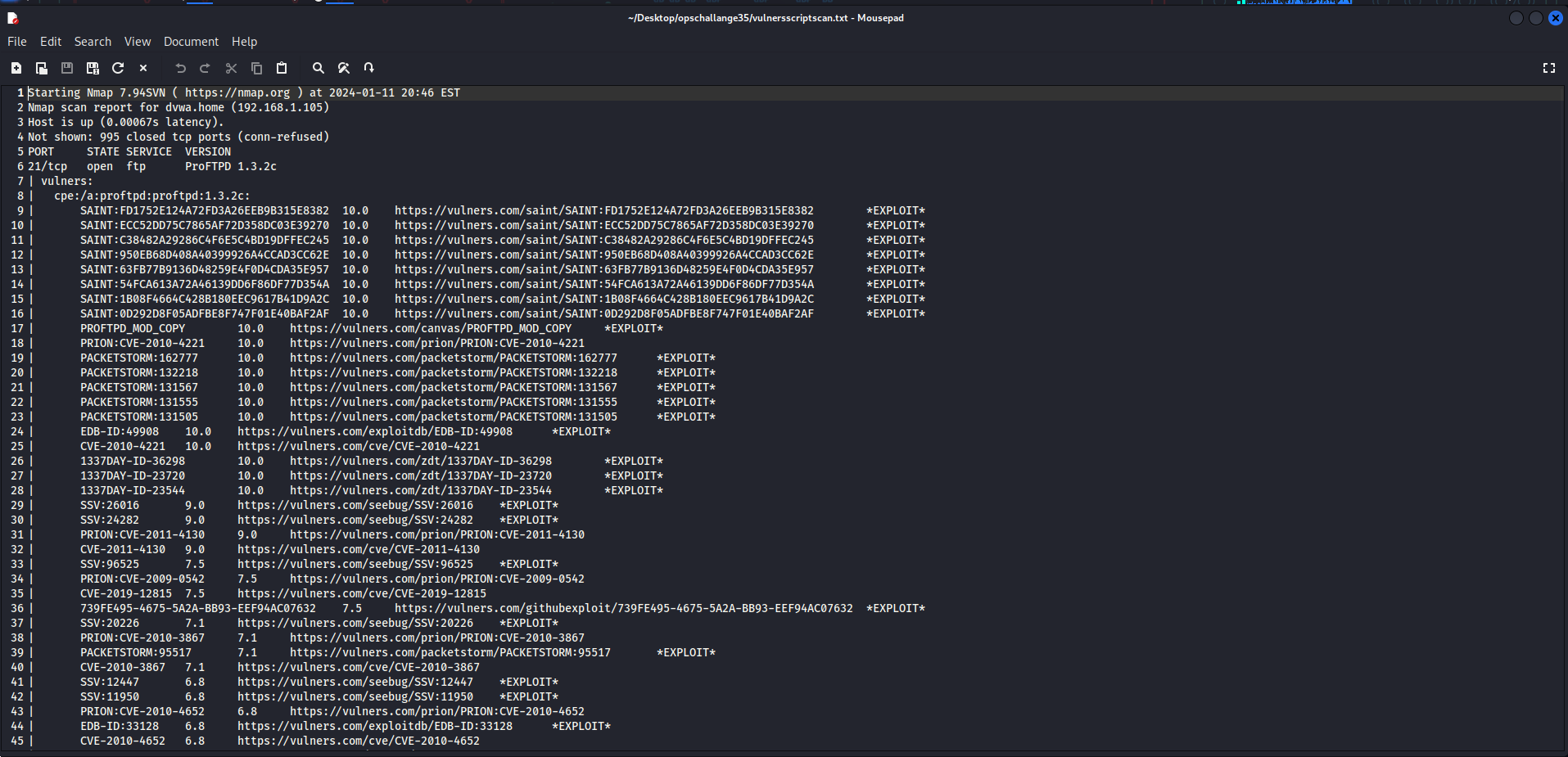
* Review the output for a CVE entry. Navigate your browser to the [CVE MITRE site](https://cve.mitre.org/) and look up the CVE. Paste a screenshot of this in your submission doc and explain the vulnerability in your own words.



DOS (Denial Of Service) attack to the server by connecting multiple times to the SSH server, waiting for the password prompt, then disconnecting.

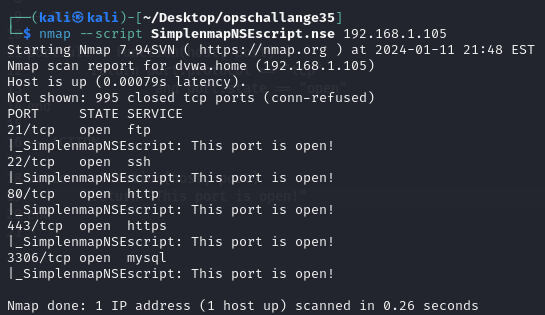
*Another script that lets Nmap gather vulnerabilities is called “Vulners.” Check it out!*

**



By now you probably have a good idea of how poorly configured this web server is. While this concludes our investigation, what if we wanted to write our own NSE rule?

* In Kali Linux, write your own .nse script that gathers some kind of information about the target.



* Post your LUA-language .nse script to your public GitHub repository. Share the link to it in your submission doc today.

<https://github.com/Rbrasil72/LUA-Scripts/blob/main/nmapscripts/SimplenmapNSEscript.nse>

## **Part 4: Reporting**

Answer the below discussion prompts in your own words:

* What are some of the biggest security concerns on the web server?

Php admin page was easily accessible without any type of protection, were there are users databases that show information about them like their name and password hashes

Alot of vulnerabilities and exploits are available because of how outdated the server is

Files can be easily accessible by just changing the URL

* Why might a security professional choose to use Nmap NSE for evaluating a web app target?

A security professional might choose to use nmap NSE for evaluating a web app target for the quick vulnerability assessment.

* In your opinion, how can adopting scripting and extensible defenses change the way security professionals handle operations?

They can up their defenses by knowing what vulnerabilities are on the web server.