Week 1: Security Assessment & Vulnerability Analysis

Firstly, this is the version of Ubuntu that the VM is using (Ubuntu 24.04.2 LTS):

* A screenshot of a computer

  AI-generated content may be incorrect.

The Long-Term Support version was chosen because it is designed for stability, receives support for 5 years, and the costs saved by infrequent migration are why it is also good for production and is enterprise grade.

Ubuntu is also better for general use, server environments and development, as opposed to Kali Linux that is heavily skewed towards pen-testing and security professionals. The tools I will use later are more compatible with Ubuntu and Ubuntu has less licensing issues than Red Hat Linux, these are some of the reasons I chose Ubuntu over RHL (I also am more familiar with Ubuntu).  
  
For installing OWASP ZAP I followed the tutorial here:

<https://www.youtube.com/watch?v=a6_TprVx7LE>

ran these commands for an update and Java installation   
“sudo apt update

sudo apt install openjdk-17-jdk -y” due to this erorr:

* A screenshot of a computer program

  AI-generated content may be incorrect.

Java version and run command of the shell script:

* A computer screen shot of a program

  AI-generated content may be incorrect.

Here is where it is installed in:

* A black background with white text

  AI-generated content may be incorrect.  
    
  Then I installed apache2 so there can be a web server to handle the HTTP requests (sudo systemctl status apache2, sudo apt install apache2, sudo systemctl start apache2):
* A screenshot of a computer

  AI-generated content may be incorrect.

Now I run ZAP again and do an automated scan:

* A screenshot of a computer

  AI-generated content may be incorrect.

Proof of my IP address:

* A screenshot of a computer

  AI-generated content may be incorrect.

For pentesting later I will put a fresh Kali machine and the CSE Ubuntu VM on the same network on adapter 2, with adapter 1 being the Bridged Adapter to enable access to the internet, hence if the IP address of the CSE VM changes that is why.   
  
As for the ZAP scan it showed 6 Alerts with none being High Priority alerts, 3 being medium, 2 being low and 1 being an informational priority alert. These are denoted by the red, orange, yellow and blue flag respectively.

* A screenshot of a computer

  AI-generated content may be incorrect.

For the first medium alert there are 13 instances where the CSP header is not set, the risk is medium and OWASP ZAP has a high confidence rating in its finding meaning it is unlikely to be a false positive. All my findings/screencaptures shall continue to be presented with bullet points.

* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.

A Content Security Policy as described by OWASP ZAP is an added layer of security that supports the detection and mitigation of some types of attacks, including Cross Site Scripting (XSS) and data injections.

XSS (abbreviated to XSS as CSS is usually used for Cascading Style Sheet) is a web security vulnerability that enables an attacker to compromise users’ interactions with a vulnerable application, ultimately leading to attackers pretending to be victim users, and carrying out any actions that user can perform, in the event that the compromised user’s account has admin privileges then the attacker could gain absolute control over all the application’s data and functionality (as described by <https://portswigger.net/web-security/cross-site-scripting>).

Injections, however, are an attacker’s attempt to send data to an application in a manner that will change the meaning of the commands that will be interpreted. One common example of a data injection attack is an SQL injection; this changes query logic by adding payloads that automatically default to true to do things like gain unauthorised logins (source: <https://owasp.org/www-community/Injection_Theory>).

Back to CSP headers, since they provide standard HTTP headers that allow website owners to declare which sources of content are approved and should be allowed to load on that webpage, then something like XSS happening is more unlikely as the owner can declare that only scripts from trusted sources are to be executed. Sourced from: <https://www.imperva.com/learn/application-security/content-security-policy-csp-header/#:~:text=A%20Content%20Security%20Policy%20(CSP,the%20source%20of%20the%20content>.

The solution to not having CSP headers set is ensuring that the web server (Apache2) is configured to set the CSP header.

Now for the second medium alert “Hidden File Found”:

* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.
* A close-up of a sign

  AI-generated content may be incorrect.

The apache\_server\_status file/page should be secured properly and only accessible to relevant parties as it provides information about the server’s current status and operations. For example, this is unsecure:

* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer program

  AI-generated content may be incorrect.

One of the vulnerabilities is that the VHost could show the path to more sensitive information via the URL, another is that the version of the server is displayed (this could lead to an attacker researching what known vulnerabilities there are in the current version of the server).

Read more here: <https://www.ultrared.ai/blog/apache-server-status-a-treasure-trove-for-penetration-testers#:~:text=Misconfigured%20or%20improperly%20secured%20Server,or%20fuzzer%20such%20as%20ffuf>.

Since I verified that this alert is correct and that it was found with a high degree of confidence the next step is the solution, one solution is to keep the mod\_status enabled and make it require an admin VM’s IP address to remotely view it (this could be a Kali VM) but I chose to just disable the mod\_status to remove the /server-status endpoint.

* A screenshot of a computer program

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.

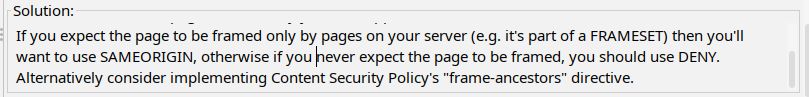
It still shows the server version which is not good.

Next an alert found with medium confidence, the missing anti-clickjacking header:

* A screenshot of a computer

  AI-generated content may be incorrect.
* A close-up of a message

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.
* 

Clickjacking is when an invisible, actionable webpage contains a hidden link or button, for example, within an iframe. The iframe would be laid on top of the user’s anticipated decoy web page content and could take them to another malicious site ([What is Clickjacking? Tutorial & Examples | Web Security Academy](https://portswigger.net/web-security/clickjacking)).

One solution is the X-Frame-Options HTTP headers, and if the server will use framesets then the syntax is “X-Frame-Options: SAMEORIGIN” otherwise it is “X-Frame-Options: DENY” ([X-Frame-Options header - HTTP | MDN](https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/X-Frame-Options)), ([Frameset - Glossary](https://www.devx.com/terms/frameset/#:~:text=A%20frameset%20is%20an%20HTML%20structure%20used%20to,multiple%20web%20pages%20within%20a%20single%20browser%20window.)). The frame ancestors CSP directive can also be used to specify which pages are allowed to embed what, this is more modern so this should be used.

Server still leaks version info at multiple places (14 instances) too:

* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.
* A close-up of a message

  AI-generated content may be incorrect.
* A close up of a text

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.

Now for the last low risk alert:

* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.
* A close up of a text

  AI-generated content may be incorrect.
* A white background with black text

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.

The solution to this issue is setting the “X-Content-Type-Options: nosniff” so that browsers do not guess an incorrect MIME Type. (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/X-Content-Type-Options>), (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/MIME_types#mime_sniffing>) .

The informational alert was just a false positive since the landing page had the word “bugs” in it.

* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.

Now I will install nmap.

Firstly I check to see if it is installed then I install it.

* A screenshot of a computer program

  AI-generated content may be incorrect.

Current IP addresses:

* A screenshot of a computer program

  AI-generated content may be incorrect.

Scan of common 1000 TCP ports, local IP address omitted:

* A computer screen with white text

  AI-generated content may be incorrect.

Other scans showing server version, available hosts:

* A screenshot of a computer program

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.

Also, since HTTP (port 80) is being used it is less secure than HTTPS (typically associated with port 88) which means attackers can read the traffic between the server and client since HTTP is unencrypted.

Now for security auditing with Lynis I will use the latest Kali machine, for the sake of OpSec, default credentials should be changed.

My Kali VMs IP addresses:

* A screenshot of a computer program

  AI-generated content may be incorrect.

Lynis install:

* A screenshot of a computer

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.

Installing openssh to remotely login from my Kali VM, the VMs need to be configured to only allow ssh logins using RSA keys or at the very least passwords.

* A screenshot of a computer screen

  AI-generated content may be incorrect.
* A computer screen shot of text

  AI-generated content may be incorrect.
* A computer screen shot of a computer screen

  AI-generated content may be incorrect.

Lynis is system specific so I installed it on Ubuntu

* A screenshot of a computer program

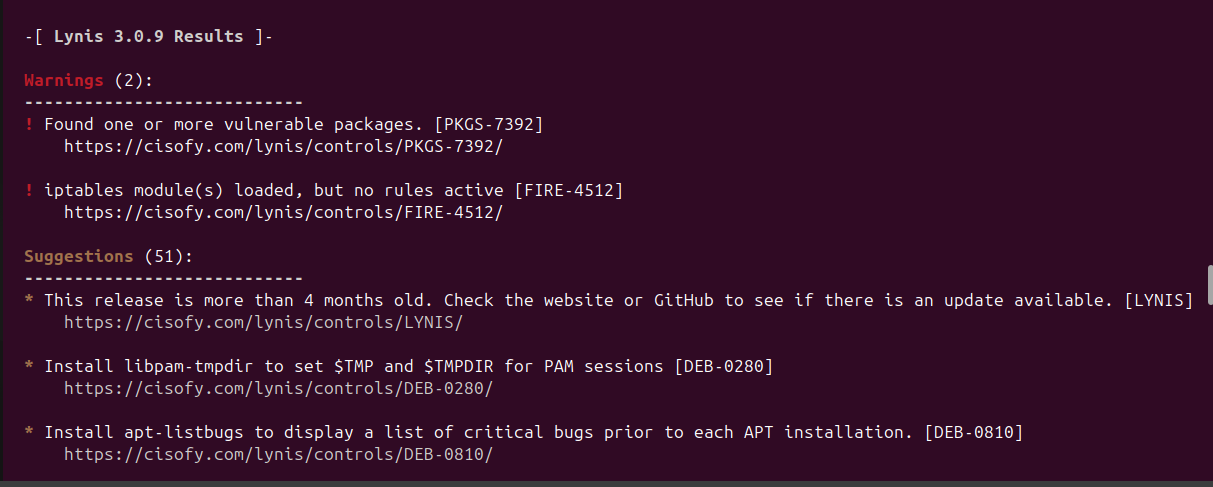
  AI-generated content may be incorrect.

PC crashed, redoing the scan

* A screenshot of a computer program

  AI-generated content may be incorrect.

There were 2 warnings and 51 suggestions given as there was a lot of things found. One thing that can be implemented to improve security is setting maximum password age so passwords reset as opposed to using a static one for a long time and running the risk of compromising that, it needs to be implemented for enterprise environments.

* 
* A screenshot of a computer program

  AI-generated content may be incorrect.
* A screenshot of a computer program

  AI-generated content may be incorrect.
* A screenshot of a computer program

  AI-generated content may be incorrect.

System hardening index should be improved until it scores 100 on the index for maximum security (according to Lynis).

Security assessments using risk matrices should be done.