Vulnerability Assessment / Penetration Test Report

**Report Title:** Vulnerability Assessment and Penetration Testing Report  
**Date:** April 21, 2024  
**Prepared by:** CyberGuardian Solutions

**Team Members** (Group 4)**:**   
Uju Anachebe, Richard Beiswanger, Masrura Tasnim, Gevorges Chemmala

**Recipient Team** (Group 3)**:**   
Jarien Tabor, Darin Sun, Christopher Fox, Christopher Nicolaus, Andrew Marks

# Executive Summary

CyberGuardian Solutions (Team 4)’s findings. During an in-depth analysis, we uncovered high-risk vulnerabilities that potentially lead to the malicious access of Akwaaba’s systems and data, highlighted by the following key findings:

• Misconfigured permissions enabling index directory browsing

• Weak authentication mechanisms and credentials (no https)

• Lack of input validation and sanitization controls

• Outdated software versions with known vulnerabilities (plugins and themes)

• Default content and configuration information

Until they are corrected, these vulnerabilities could cause a disruption to business operations for Akwaaba. It is recommended that these issues are mitigated by updating the problem configurations, implementing controls, and removing the default contents. We recommend continuous monitoring to prevent new vulnerabilities as well.

The following report contains our findings in more detail, specifying how we discovered them, what the options may be to correct the issues, and future recommendations to ensure Akwaaba’s WordPress web server is as secure as possible while remaining fully operational as a successful e-commerce website.

# 1. Test Environment

The vulnerability assessment was conducted on a server with the following specifications:

* **Operating System:** Red Hat Linux 8.4
* **Webserver:** Apache 2.4.37
* **Database:** MariaDB 10.6.4
* **Content Management System:** WordPress

The server has SSH access, and WordPress appears to have security enhanced with plugins such as Solid Security, All-in-One Security (AIOS), and Sucuri Security.

# 2. Vulnerability Analysis

### Vulnerability Analysis Summary

Several vulnerabilities of varying severity levels were identified:

1. **Medium Severity Vulnerabilities**
   * **Weak Key Exchange Algorithm(s) Supported (SSH):** Weak KEX algorithms, like diffie-hellman-group-exchange-sha1, can be exploited to break individual connections. Recommendation: Disable weak KEX algorithms and use more secure alternatives.
   * **Weak Encryption Algorithm(s) Supported (SSH):** Weak encryption algorithms such as aes128-cbc and aes256-cbc expose the system to security breaches. Recommendation: Disable these algorithms and use more secure options.
   * **Content Security Policy (CSP) Header Not Set:** This can lead to various injection attacks. Recommendation: Implement a CSP header to prevent such attacks.
   * **Lack of HTTPS:** The website does not support HTTPS, exposing data to interception. Recommendation: Implement SSL/TLS encryption and redirect HTTP traffic to HTTPS.
2. **Low Severity Vulnerabilities**
   * **Server Leaks Version Information via “Server” HTTP Response Header Field:** Attackers can use this information to exploit known vulnerabilities. Recommendation: Remove or customize the “Server” HTTP response header.
   * **Cross-Domain JavaScript Source File Inclusion:** Including JavaScript from different domains can lead to security risks. Recommendation: Ensure only trusted domains are allowed.
   * **Private IP Address Disclosure:** Disclosing internal IP addresses can help attackers map the internal network. Recommendation: Obfuscate or remove internal IP addresses from responses.
   * **Server Leaks Information via “X-Powered-By” HTTP Response Header Field:** This can reveal the underlying technology stack. Recommendation: Remove or customize the “X-Powered-By” header.
   * **X-Content-Type-Options Header Missing:** This can lead to cross-site scripting (XSS) attacks. Recommendation: Implement the header with the value “nosniff.”
3. **Informational Severity Vulnerabilities**
   * **Suspicious Comments in Source Code:** Comments can reveal sensitive information. Recommendation: Review and remove unnecessary comments.
   * **User Agent Fuzzer:** The application responds differently based on the user agent string, which can be exploited to bypass security controls. Recommendation: Ensure consistent application behavior regardless of the user agent.

### Vulnerability Assessment Details

### Vulnerability Name/ID: Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)

**Severity Rating:** Medium (CVSS: 5.3)

**Description:** The remote SSH server is configured to allow/support weak key exchange (KEX) algorithm(s), specifically diffie-hellman-group-exchange-sha1, which uses SHA-1.

**Proof of Concept (if applicable):** Not provided.

**Impact on System:** An attacker can quickly break individual connections due to the use of weak KEX algorithms.

**Recommendations for Mitigation:**

* Disable the reported weak KEX algorithm(s).
* For 1024-bit MODP group/prime KEX algorithms, alternatively use elliptic-curve Diffie-Hellman, e.g., Curve 25519.

### Vulnerability Name/ID: Weak Encryption Algorithm(s) Supported (SSH) OID: 1.3.6.1.4.1.25623.1.0.105611

**Severity Rating:** Medium (CVSS: 4.3)

**Description:** The remote SSH server is configured to allow/support weak encryption algorithm(s), which could potentially expose the system to security breaches.

**Proof of Concept (if applicable):** Not provided.

**Impact on System:** The use of weak encryption algorithms may allow an attacker to recover plaintext from a block of ciphertext, compromising the confidentiality of the information transmitted.

**Recommendations for Mitigation:**

* Disable the reported weak encryption algorithm(s).
* The remote SSH server should no longer support the following weak client-to-server and server-to-client encryption algorithms:
  + aes128-cbc
  + aes256-cbc

### Vulnerability Name/ID: Content Security Policy (CSP) Header Not Set

**Severity Rating:** Medium

**Description:** The CSP header is not set, which can expose the site to various injection attacks.

**Proof of Concept:**

* URL: http://10.96.32.116/sitemap.xml
* Method: GET

**Impact on System:** Without the CSP header, attackers can inject malicious content, potentially leading to data theft or site defacement.

**Recommendations for Mitigation**: Implement and configure the CSP header according to the guidelines.

### Vulnerability Name/ID: Server Leaks Version Information via “Server” HTTP Response Header Field

**Severity Rating:** Low

**Description:** The server discloses version information that can be used by an attacker to exploit known vulnerabilities.

**Proof of Concept:**

* URL: http://10.96.32.116/sitemap.xml
* Method: GET

**Impact on System**: Revealing version information can help an attacker tailor attacks to specific vulnerabilities of the server version.

**Recommendations for Mitigation**: Remove or customize the “Server” HTTP response header field to not disclose specific version details.

### Vulnerability Name/ID: Cross-Domain JavaScript Source File Inclusion

**Severity Rating:** Low

**Description:** JavaScript files are included from different domains, which can lead to security risks.

**Proof of Concept:**

* URL: http://10.96.32.116/home/
* Method: GET

**Impact on System:** This can lead to data leakage if malicious domains execute JavaScript in the browser context.

**Recommendations for Mitigation**: Ensure that only trusted domains are allowed to serve JavaScript files included in the web application.

### Vulnerability Name/ID: Private IP Disclosure

**Severity Rating:** Low

**Description:** The application discloses private IP addresses, which should not be publicly accessible.

**Proof of Concept:**

* URL: http://10.96.32.116/home/
* Method: GET

**Impact on System**: Disclosing internal IP addresses can aid an attacker in mapping the internal network.

**Recommendations for Mitigation**: Remove or obfuscate internal IP addresses from the application responses.

### Vulnerability Name/ID: Server Leaks Information via “X-Powered-By” HTTP Response Header Field(s)

**Severity Rating:** Low

**Description:** The server discloses information through the “X-Powered-By” header, which can reveal the underlying technology stack.

**Proof of Concept:**

* URL: http://10.96.32.116/home/
* Method: GET

**Impact on System**: This information can be used by attackers to identify potential weaknesses in the server’s technology stack.

**Recommendations for Mitigation**: Remove or customize the “X-Powered-By” HTTP response header field to not disclose specific technology details.

### Vulnerability Name/ID: X-Content-Type-Options Header Missing

**Severity Rating:** Low

**Description:** The X-Content-Type-Options header is not set, allowing browsers to sniff the MIME type, which can lead to security risks.

**Proof of Concept:**

* URL: http://10.96.32.116/home/
* Method: GET

**Impact on System**: Without this header, browsers can interpret the content type incorrectly, leading to XSS attacks.

**Recommendations for Mitigation**: Implement the X-Content-Type-Options header with the value “nosniff”.

### Vulnerability Name/ID: Information Disclosure - Suspicious Comments

**Severity Rating:** Informational

**Description:** Suspicious comments found in the source code can lead to information disclosure.

**Proof of Concept:**

* URL: http://10.96.32.116/home/
* Method: GET

**Impact on System:** Comments in the source code can reveal sensitive information or hints about the application’s structure.

**Recommendations for Mitigation**: Review and remove any unnecessary comments from the source code before deployment.

### Vulnerability Name/ID: User Agent Fuzzer

**Severity Rating:** Informational

**Description:** The application responds differently based on the user agent string, which can be indicative of underlying issues.

**Proof of Concept:**

* URL: http://10.96.32.116/home
* Method: GET

**Impact on System:** An inconsistent response based on user agent strings can be exploited to bypass security controls.

**Recommendations for Mitigation**: Ensure consistent application behavior regardless of the user agent string.

### Vulnerability Name/ID: Lack of HTTPS

**Severity Rating:** Medium

**Description:** The website ( http://10.96.32.116/) does not support HTTPS (SSL/TLS) encryption. This means that the data transmitted between the client and the server is unencrypted and can be intercepted by attackers.

**Proof of Concept (if applicable):** Accessing the website via a browser or a tool like curl shows no HTTPS support.

**Impact on System:** Without HTTPS, sensitive data such as login credentials, personal information, and other confidential data can be intercepted, read, and modified by attackers, leading to a breach of confidentiality and integrity.

**Recommendations for Mitigation:**

* Implement SSL/TLS encryption by acquiring and installing a valid SSL certificate.
* Redirect all HTTP traffic to HTTPS to ensure data is always encrypted during transit.
* Enforce strict transport security headers to prevent protocol downgrade attacks.

# 3. Penetration Testing

### Penetration Testing Summary

The penetration testing revealed additional security concerns:

* **Directory Browsing:** Unauthorized users can access sensitive files. Recommendation: Disable directory browsing.
* **WordPress Plugins and Themes:** Outdated plugins and themes pose security risks. Recommendation: Update them regularly.
* **Open Ports:** The server has open ports (22 & 2222 for SSH, and 80 for HTTP). Recommendation: Review and minimize open ports.
* **WordPress XML-RPC Enabled:** This can lead to brute force and DDoS attacks. Recommendation: Disable or secure XML-RPC.
* **Disabled Right-Click Context Menu:** This does not add security but hinders user accessibility. Recommendation: Enable the right-click context menu.

### Findings Details

1. **Index Directory Browsing**
   * The website at http://10.96.32.116/ allows index directory browsing, which exposes sensitive files and information to unauthorized users. This was observed in the following URL:
     + http://10.96.32.116/home/wp-content/uploads/
   * Additionally, critical files such as firewall-rules and better-wp-security.9.3.2.zip were accessible and downloadable.
2. **WordPress Plugins and Themes**
   * Several WordPress plugins were identified, including better-wp-security, aios, and sucuri. These plugins could contain vulnerabilities if not regularly updated.
   * The WordPress theme Dyad (version 1.0.10) was found and last updated in 2016. This outdated theme may pose security risks.
3. **Disabled Site Functionality: Right-Click Context Menu, Search & Comment box**
   * The website has disabled the right-click context menu, which could hinder user accessibility and does not provide additional security. Data can still be accessed through view-source and Developer Console.
   * The website search and comment features were disabled, reducing the user functionality for the site. This may not be the best balance between security and site operations.
4. **WordPress XML-RPC Enabled**
   * XML-RPC appears to be enabled on the website, which could expose the site to various attacks, such as brute force and DDoS attacks.
5. **Open Ports**
   * Nmap reported that ports 22 (SSH), 80 (HTTP) and 2222 (SSH) are enabled on the server. This configuration should be reviewed to ensure that only necessary ports are open.

### Testing Tools and Outputs

1. **SQLMap**
   * SQLMap output indicated connection issues and potential resets due to suspicious requests. SQL injection testing did not yield definitive results, but connection resets suggest protection mechanisms against SQL injection.
2. **XSSer**
   * XSSer tests indicated that an HTTP 403 Forbidden error occurred, suggesting that some protection mechanisms are in place. However, it also detected a potential XSS vulnerability in the author parameter.
3. **Gobuster**
   * Gobuster's directory enumeration found several default WordPress directories (wp-content, wp-includes, wp-admin), indicating that the website structure follows typical WordPress patterns.

### Recommendations

Based on these findings, the following actions are recommended to improve the security of the server:

1. **Disable Directory Browsing**
   * Configure the web server to prevent directory browsing. This reduces the risk of unauthorized access to sensitive files and information.
2. **Update WordPress Plugins and Themes**
   * Ensure all WordPress plugins and themes are up to date. Consider replacing outdated themes with more recent versions.
3. **Review Open Ports**
   * Conduct a comprehensive port audit to ensure that only necessary ports are open. Implement firewall rules to minimize attack surfaces.
4. **Implement Security Measures for XML-RPC**
   * Consider disabling XML-RPC if not required. If needed, implement additional security measures to prevent brute force and other attacks.
5. **Update Homepage Banner Picture**
   * During the Vulnerability assessment, Akwaaba’s restaurant e-commerce site had a banner image of monkeys around a computer. This banner is not fit for the company image and would cause a loss of customers and profits. Visitors would think the site had been hacked seeing that image on the front page.
6. **Forwarding of Default Admin Panel Page**
   * When searching for the admin page, the default location auto-forwarded off site to an unprofessional picture of a crying baby. Recommendation: forward to a specially crafted, professional page on-site that Akwaaba has full control over.

# 4. Conclusion

Based on our findings, the vulnerabilities to Akwaaba’s web server contain considerable risk to data and system integrity which can affect the functionality and business of their e-commerce website. CyberGuardian Solutions recommends specific actions to mitigate these vulnerabilities in this report. Due to the severity of these risks, a failure to act could lead to financial and reputational loss for the company. The key measures recommended include updating all software, themes, and plugins, utilizing more secure configurations, securing input controls correctly, and eliminating all default content.

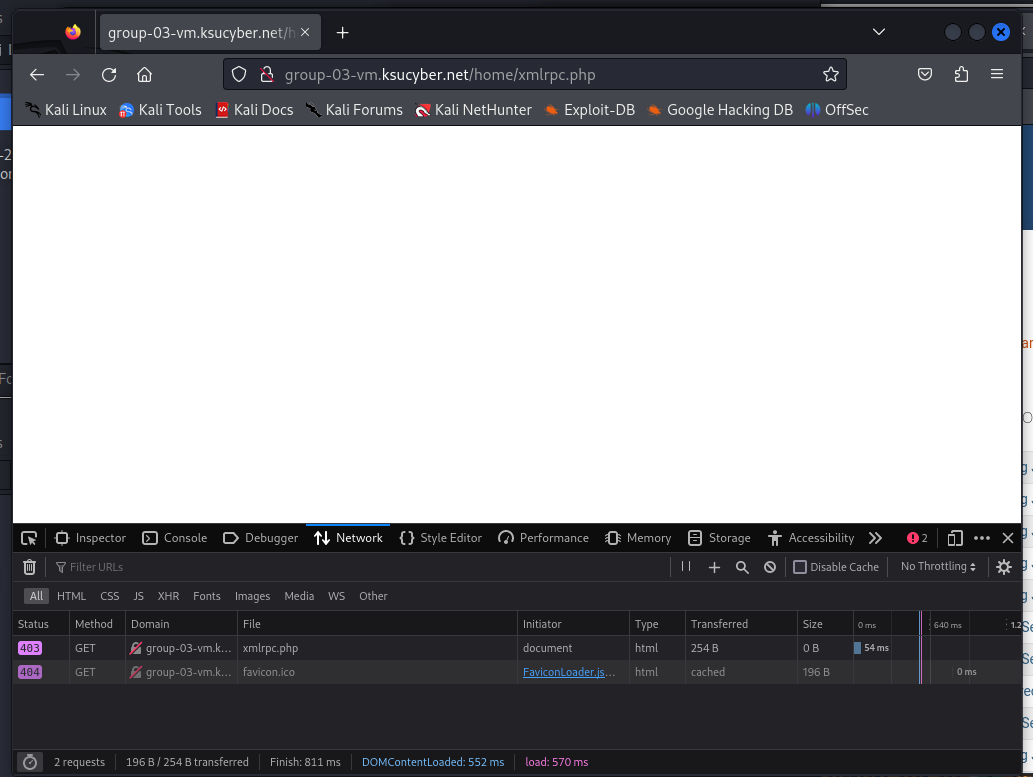
In addition to the key issues, all staff with access to the web server should engage in regular security training, and a continuous vulnerability monitoring program should be implemented. This service is something CyberGuardian Solutions can provide, as we are committed to supporting Akwaaba, so that your digital assets and business operations remain safe into the foreseeable future.

# Appendices

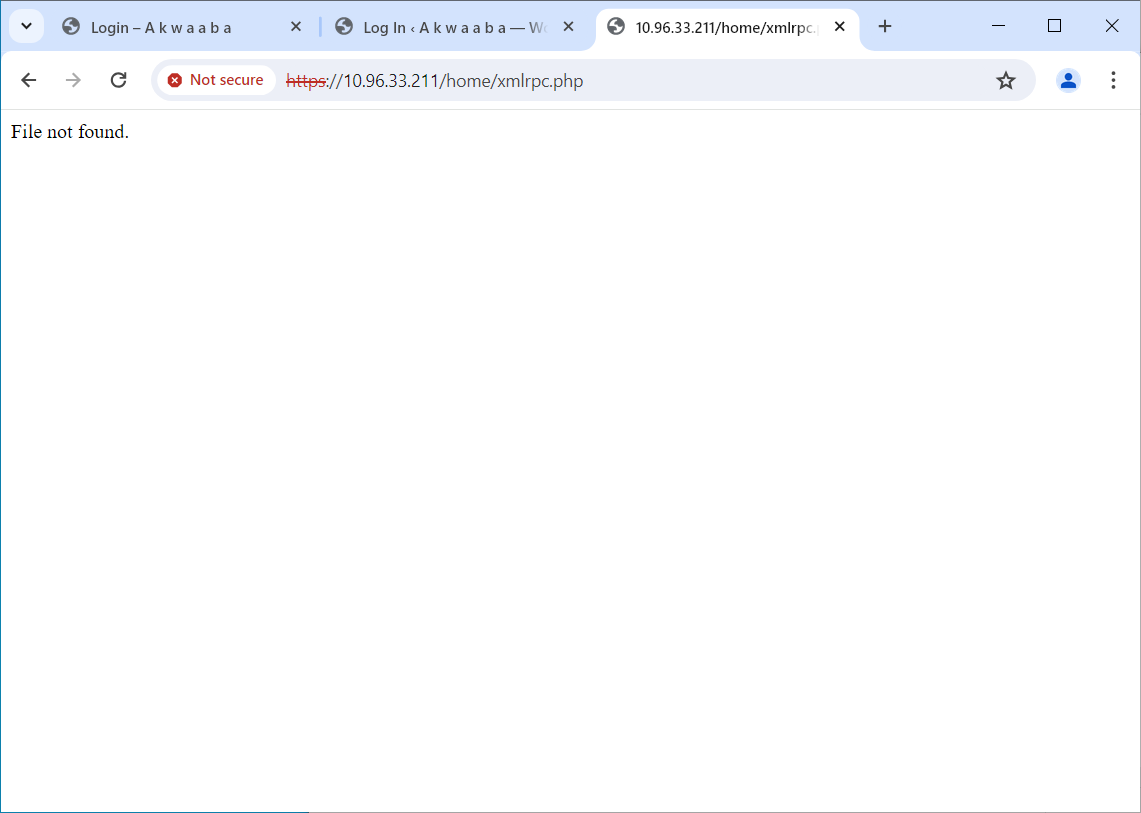
## Appendices 1 - XML-RPC

Group 3 has publicly accessible XML-RPC vulnerability file.

<https://10.96.32.116/home/xmlrpc.php>



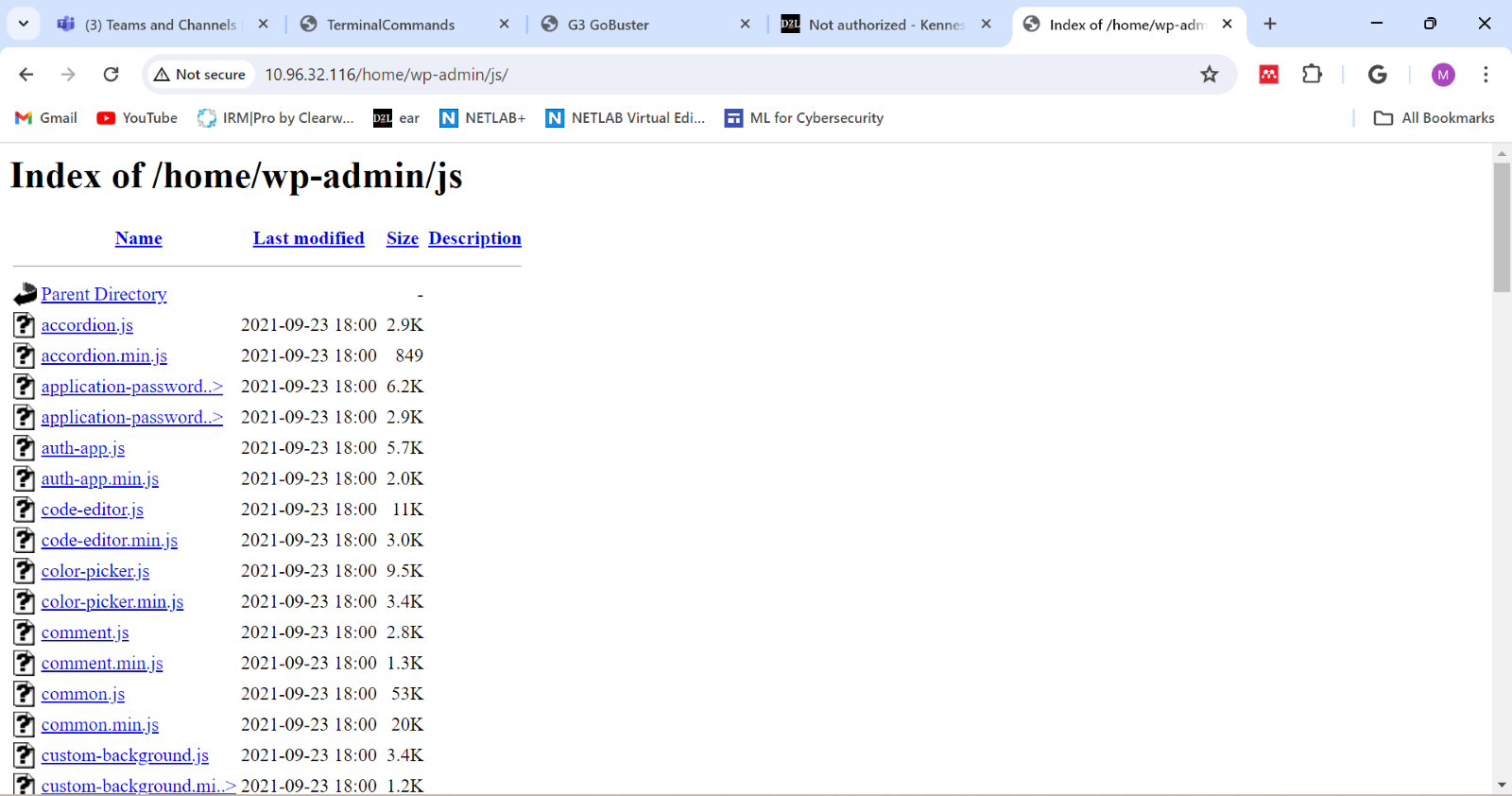
Group 4 XML-RPC <https://10.96.33.211/home/xmlrpc.php>



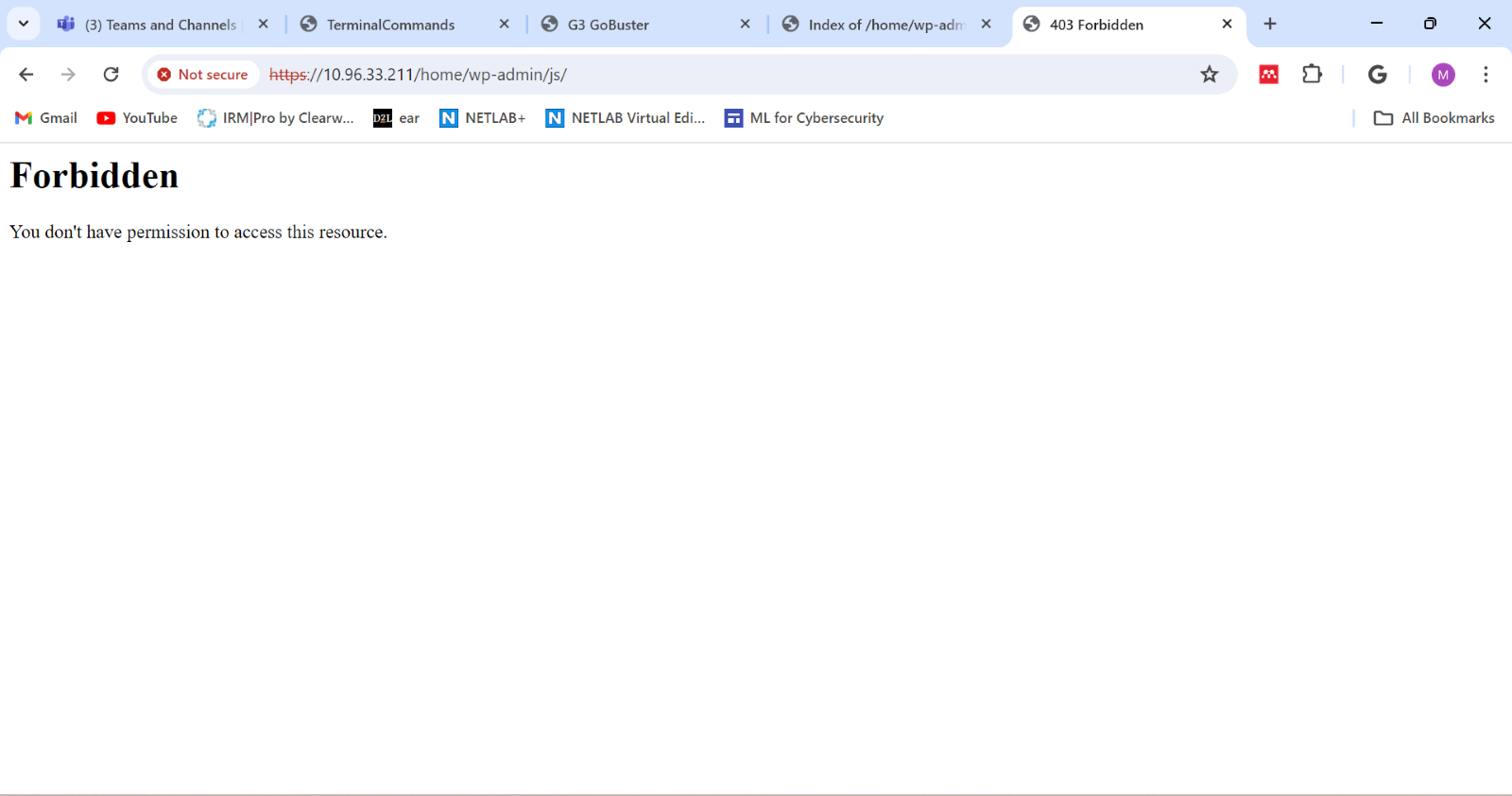
## Appendices 2 - Directory Index Browsing

Directory Index browsing is allowed on Group 3 VM (Virtual Machine). It should be made forbidden.

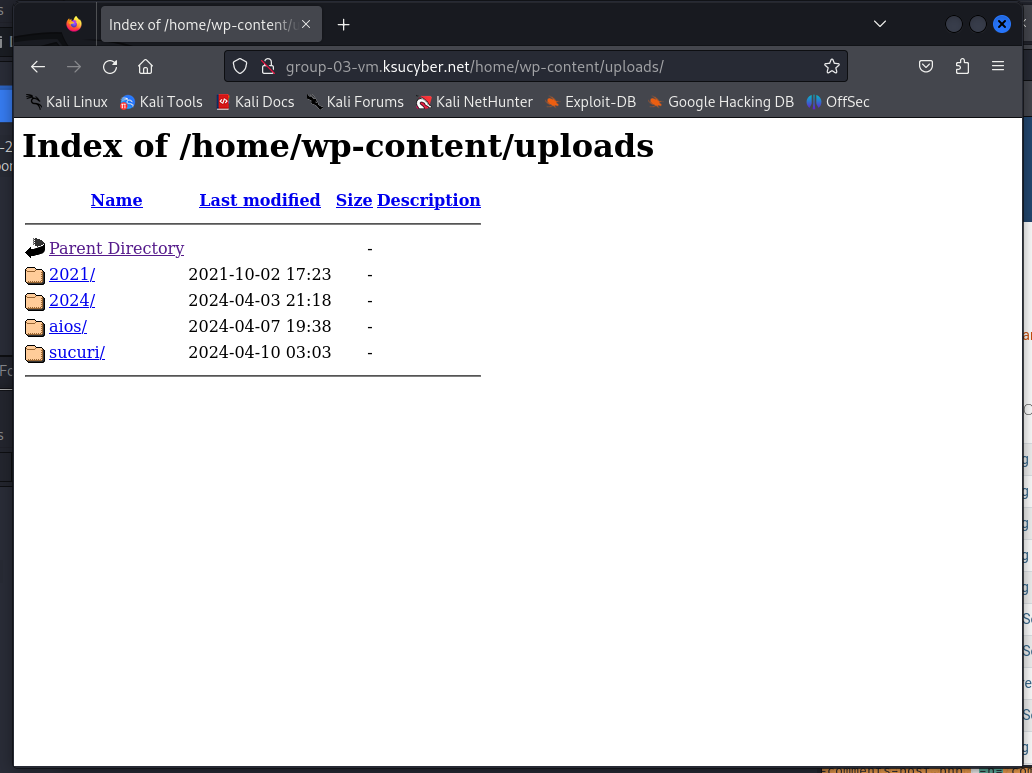
Group 3 - <http://10.96.32.116/home/wp-admin/js/>



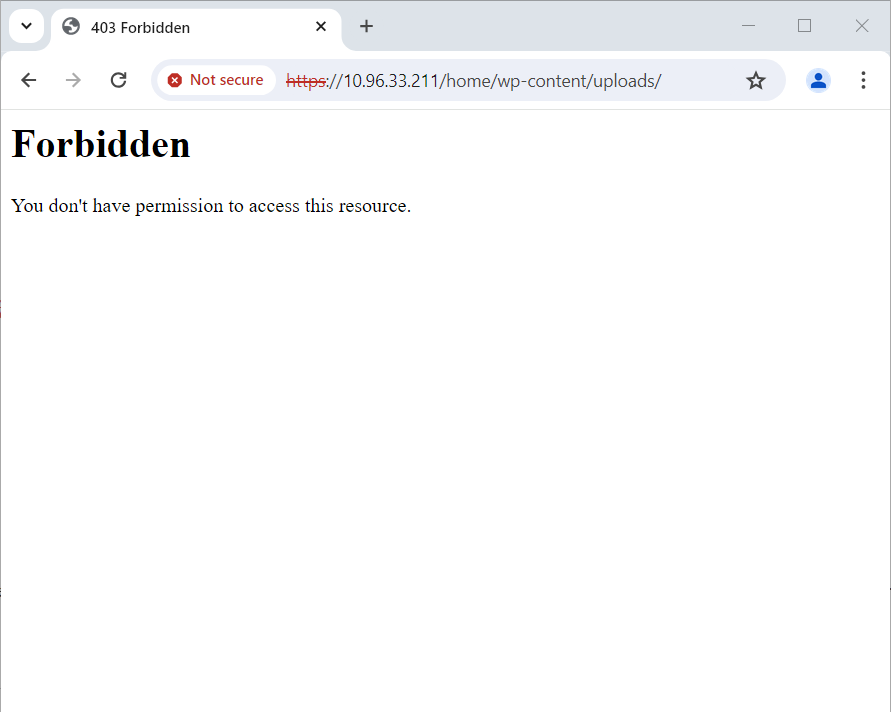
Group 4 - <https://10.96.33.211/home/wp-admin/js/>



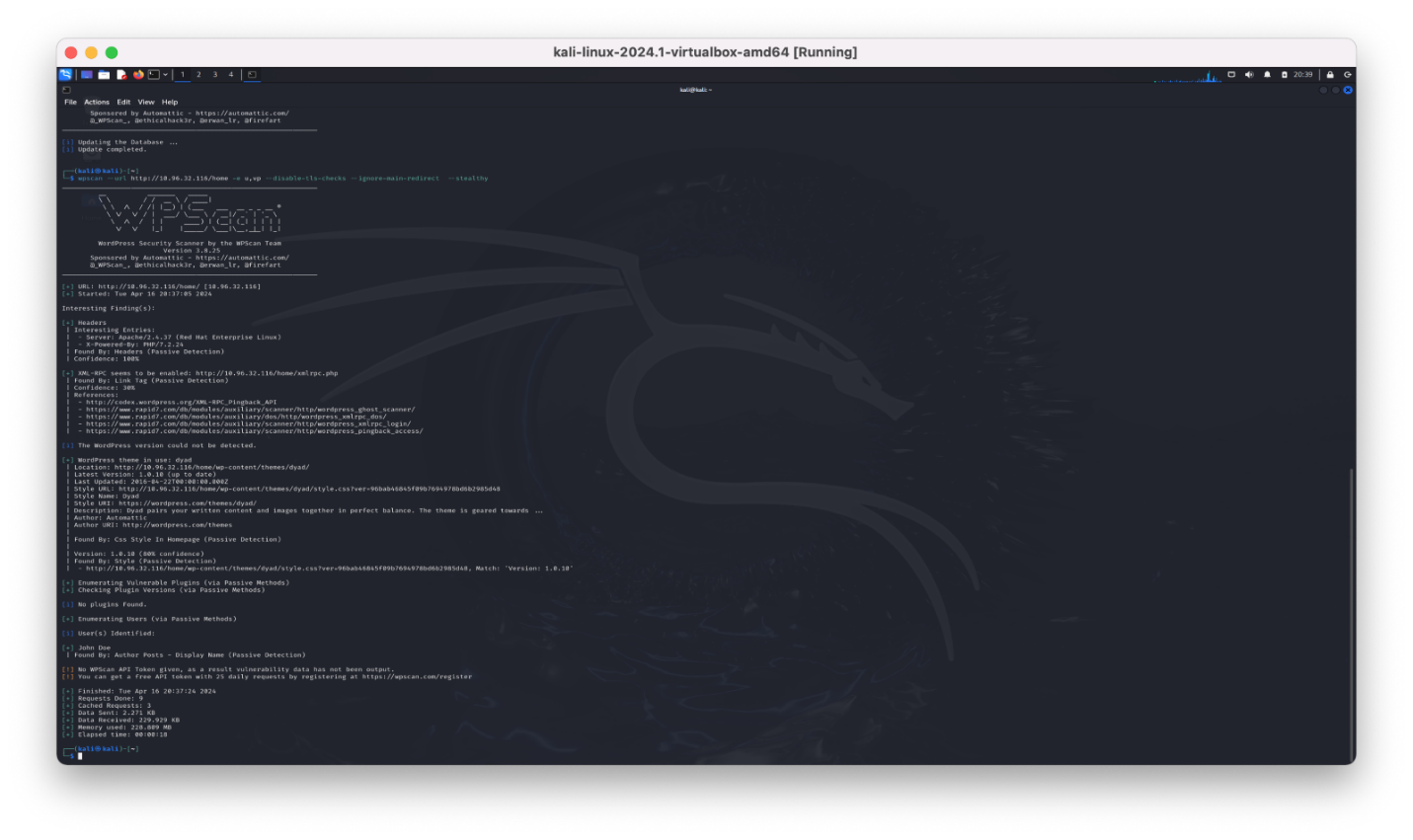
Group 3 <https://10.96.32.116/home/wp-content/uploads/>



Group 4 <https://10.96.33.211/home/wp-content/uploads/>



## Appendices 3 - WPScan Output



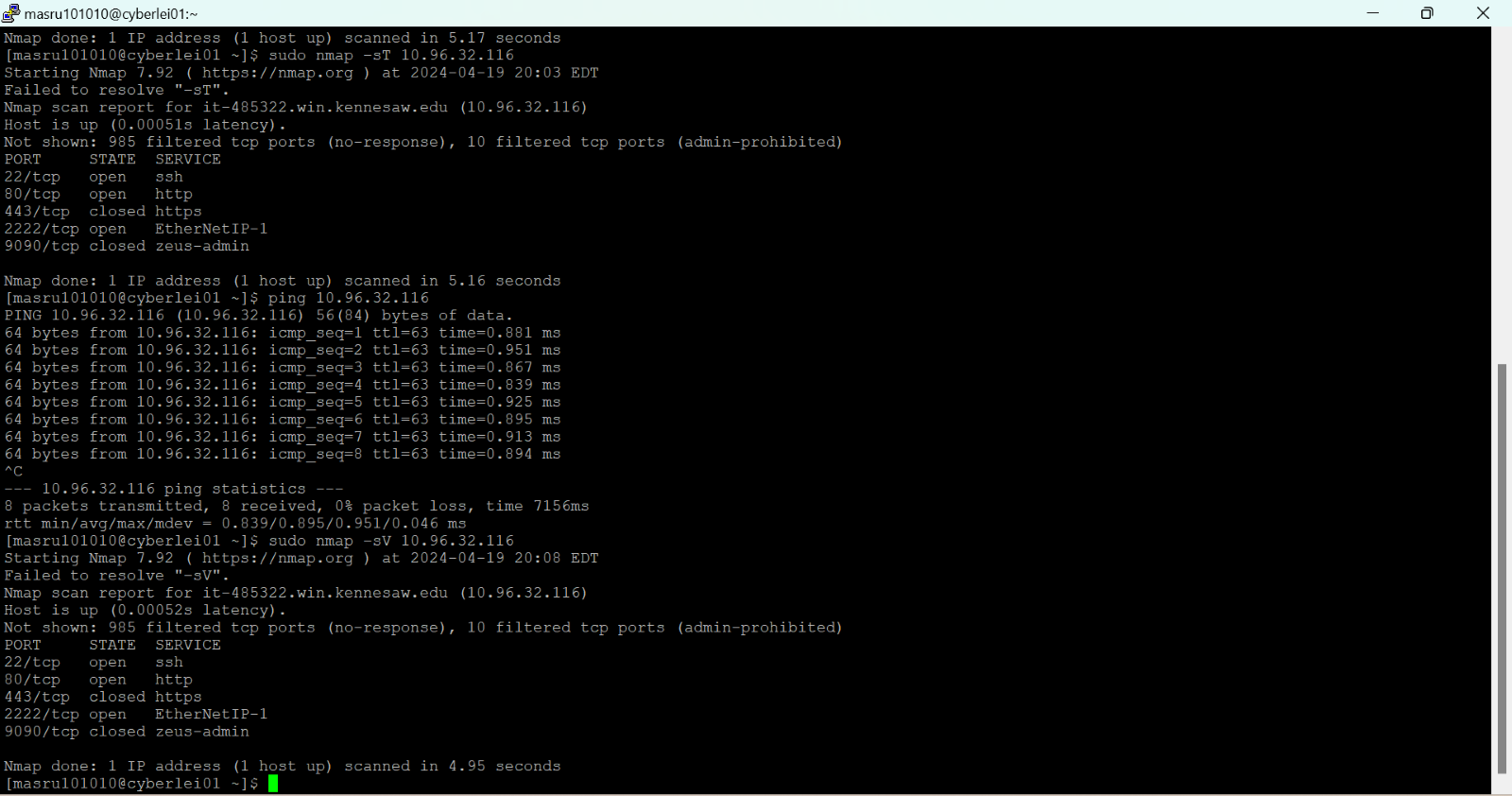
WPScan revealed several of the vulnerabilities mentioned in this report including headers, XML, Dyad theme, as well as a user: John Doe.

## Appendices 4 - Nmap Output

Nmap identified open port for http (80), ssh (22), and (2222).

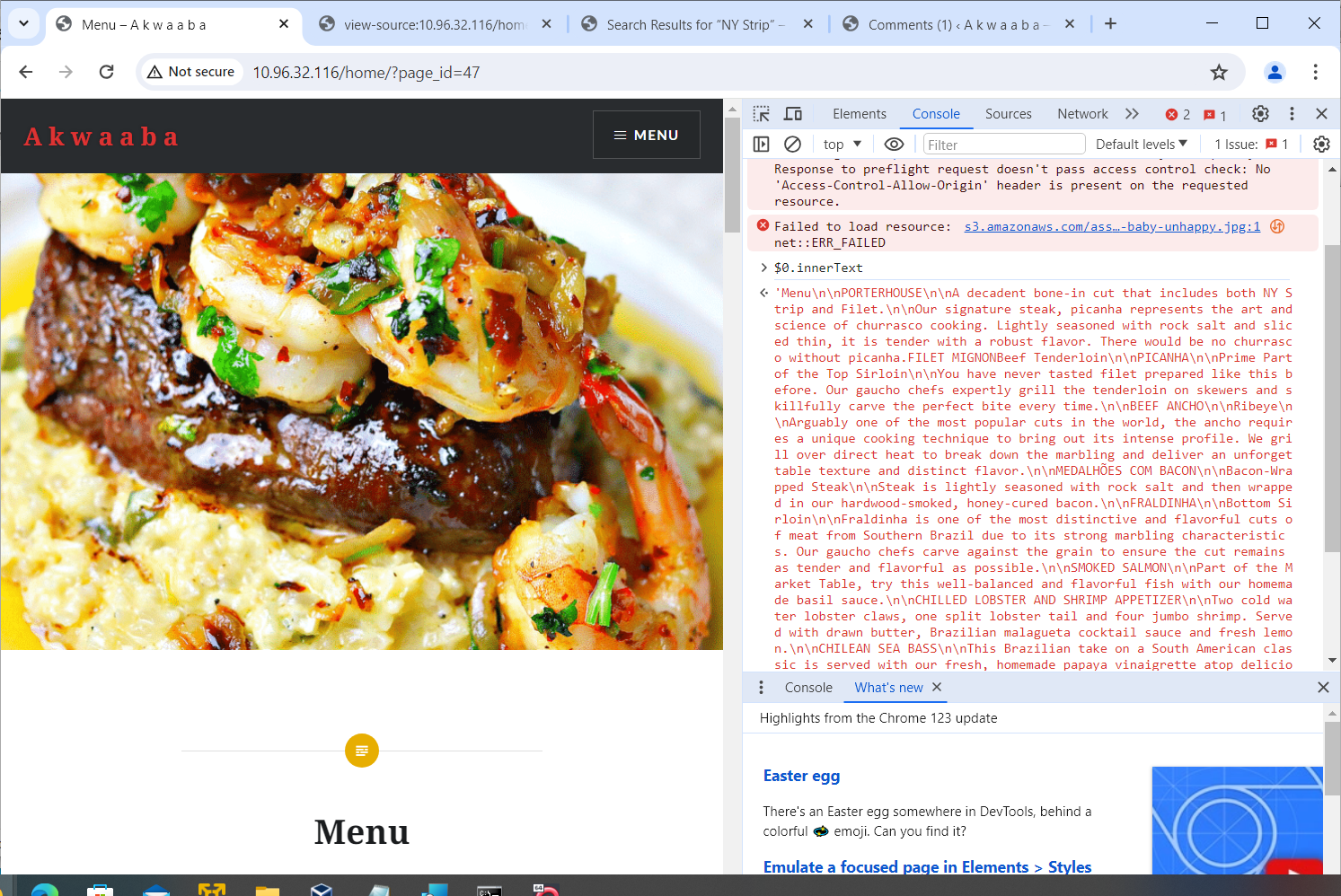
And closed ports for https (443) and (9090)

sudo nmap –sV 10.96.32.116

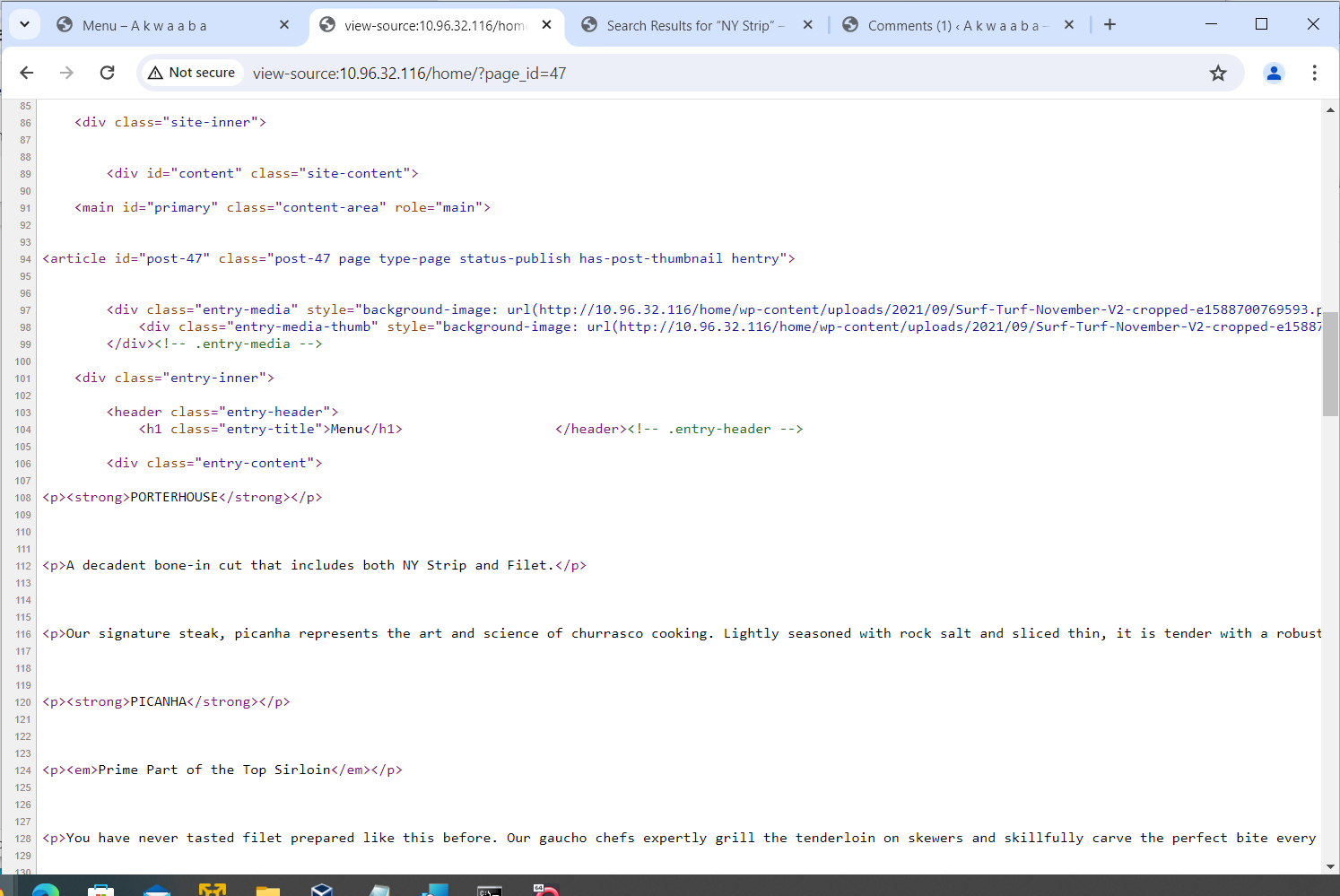


## Appendices 5 - Right Click Context Menu

Right click context menu is disabled affecting accessibility. Contents are still accessible through the developer console and view source.

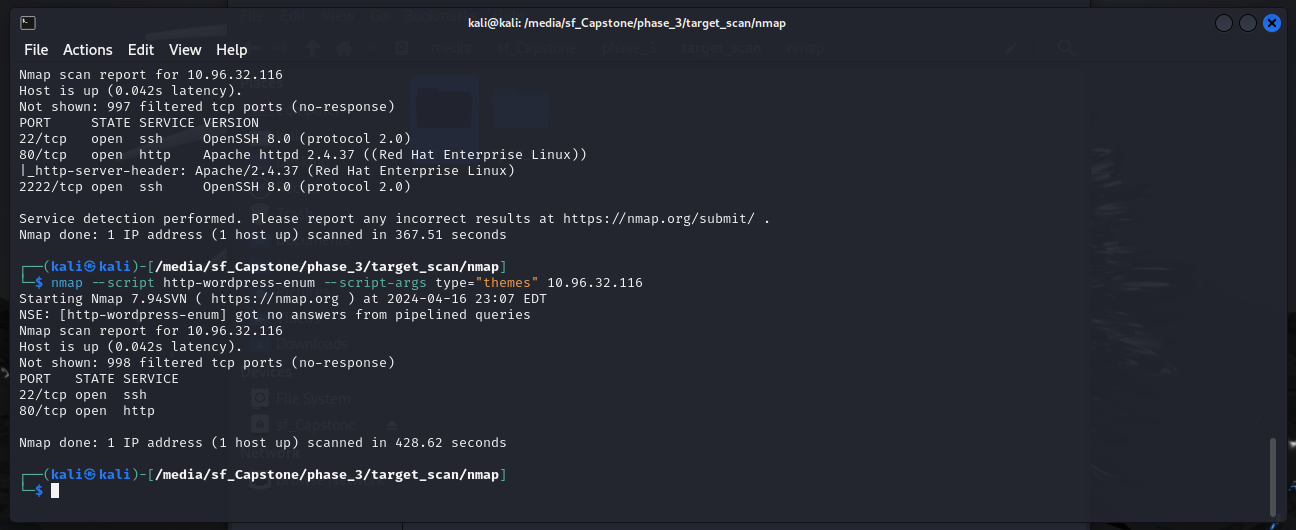
Developer Console shown on the right side of the picture.

View Source

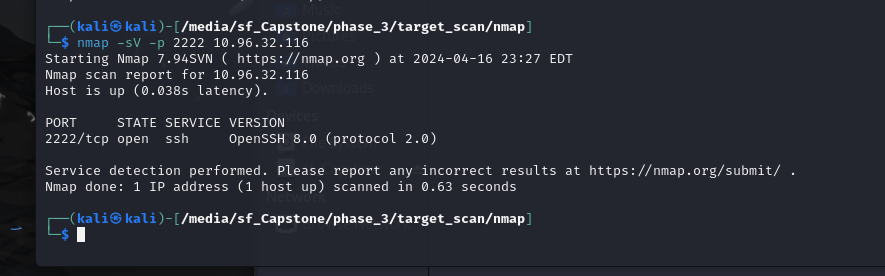
View Source reveals the content of the page.

## Appendices 6 - Nmap

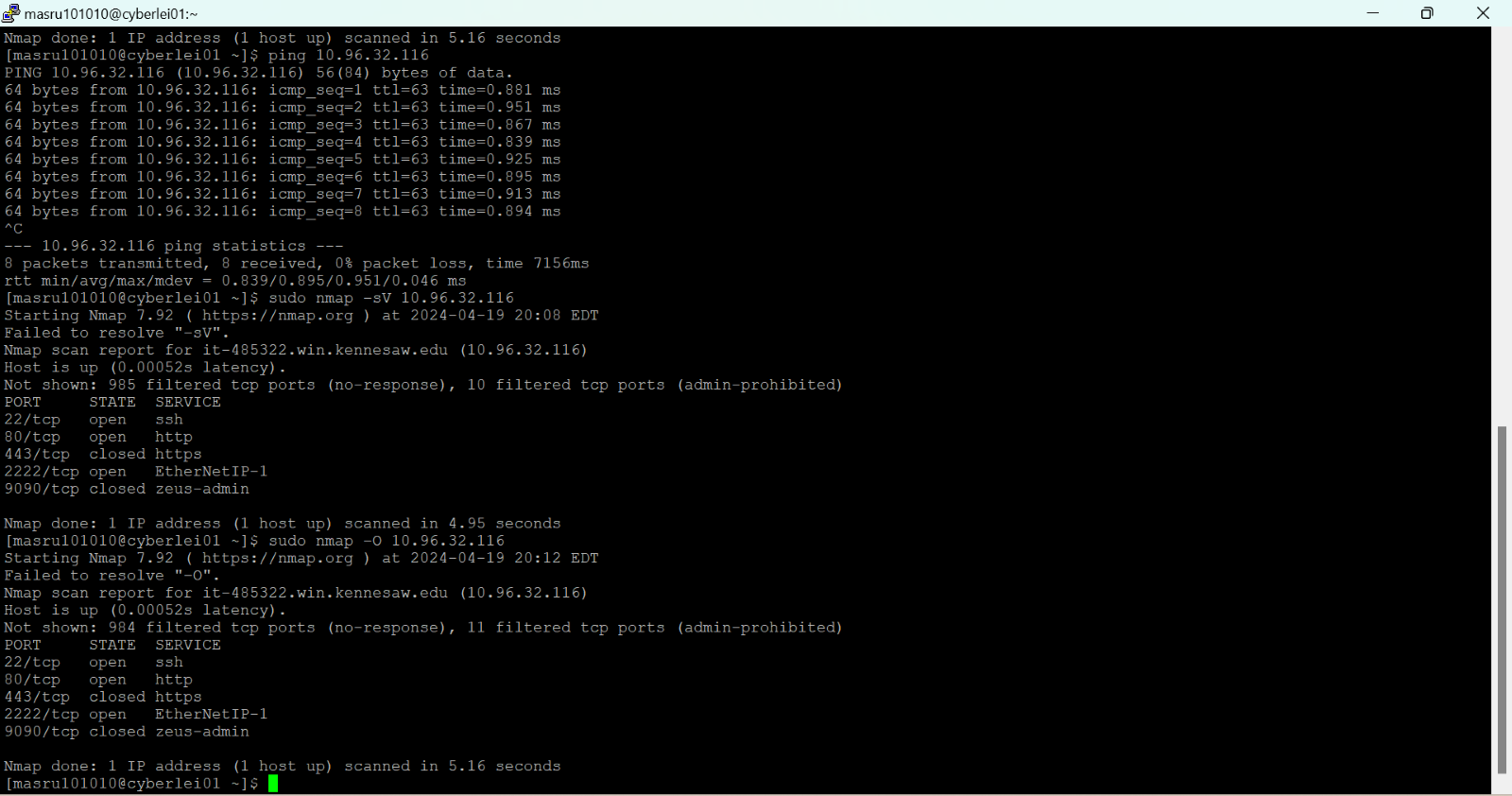
nmap --script http-wordpress-enum --script-args type="themes" 10.96.32.116



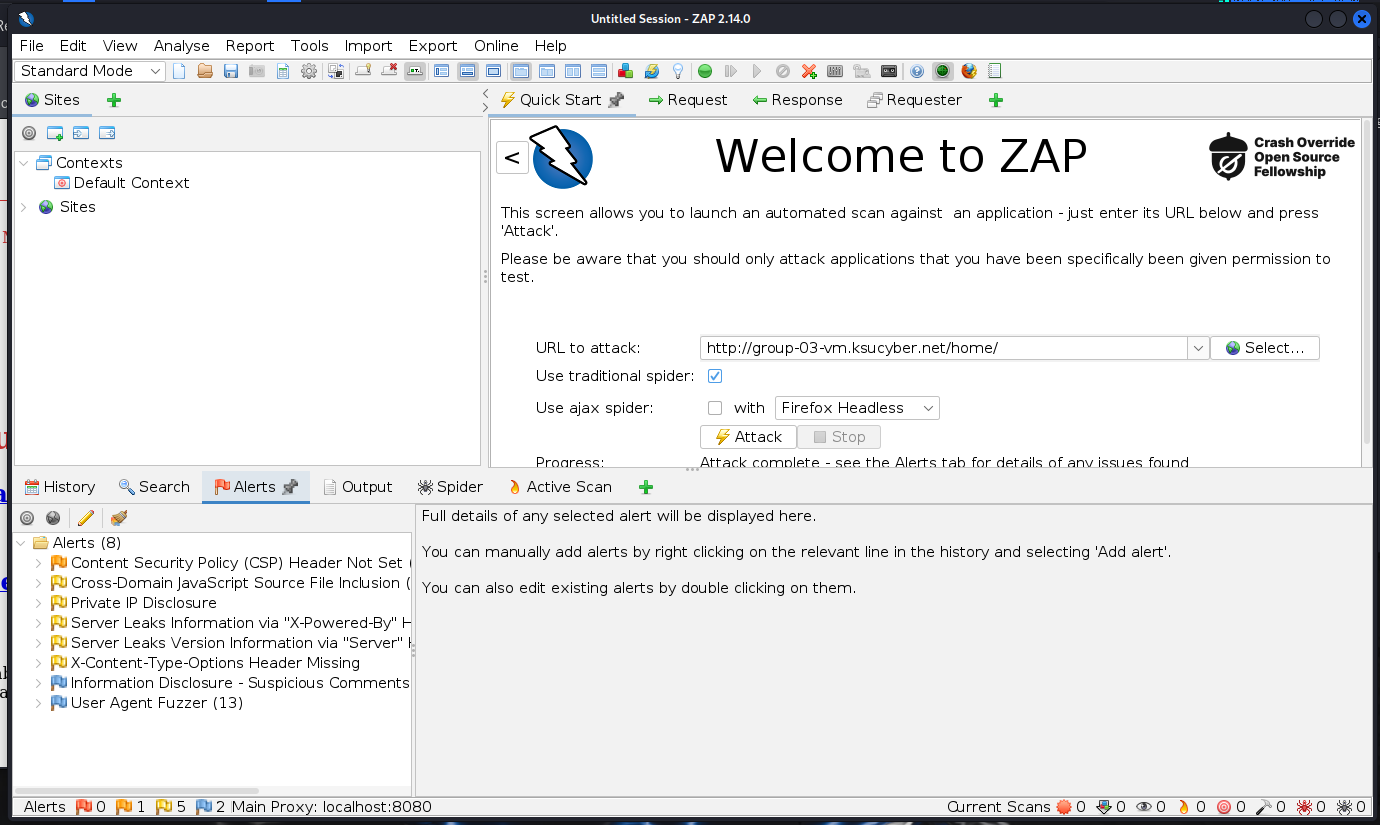
nmap -sV -p 2222 10.96.32.116



sudo nmap –O 10.96.32.116

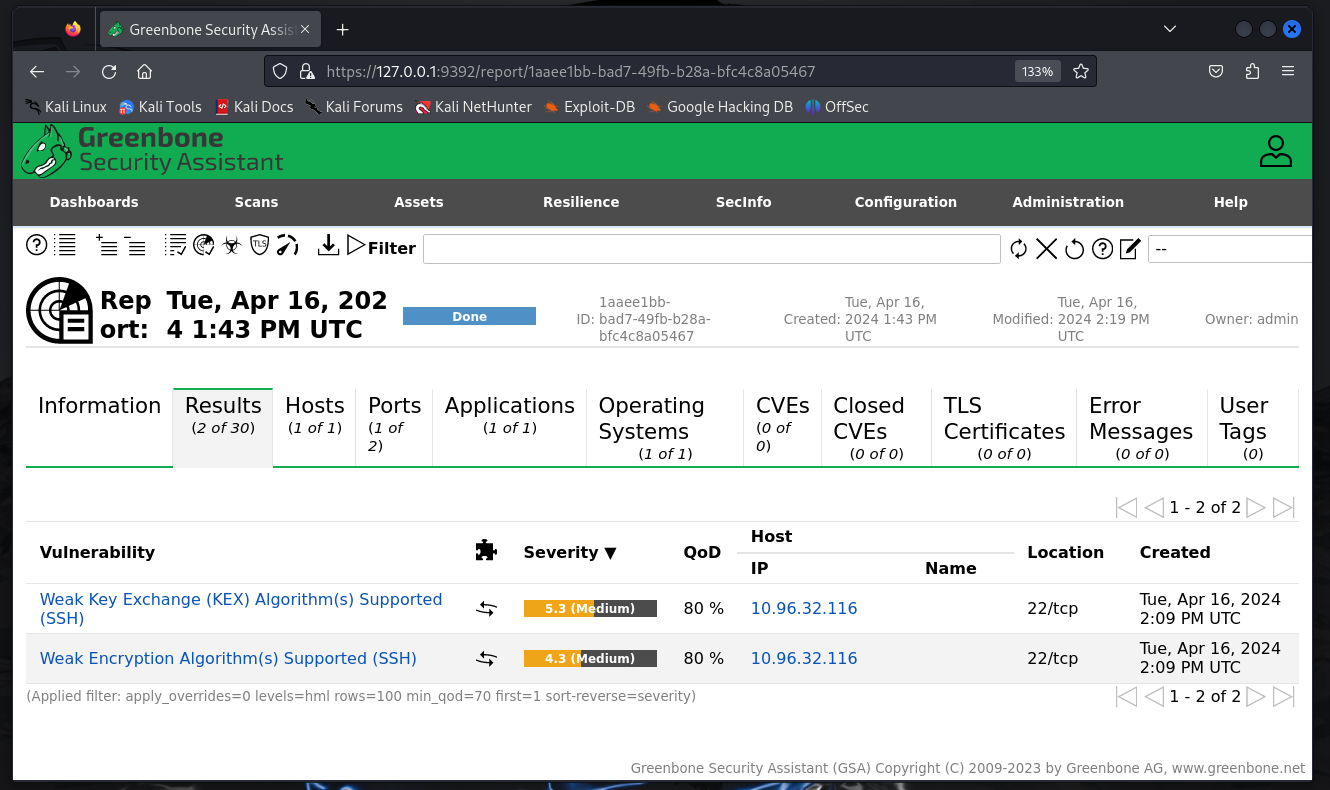


## Appendices 7 - ZAP (OWASP ZAP)

  
Issues identified by ZAP (formerly OWASP ZAP)

* Content Security Policy (CSP) Header Not Set
* Cross-Domain JavaScript Source File Inclusion
* Private IP Disclosure
* Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
* Server Leaks Version Information via "Server" HTTP Response Header Field
* X-Content-Type-Options Header Missing
* Information Disclosure - Suspicious Comments
* User Agent Fuzzer

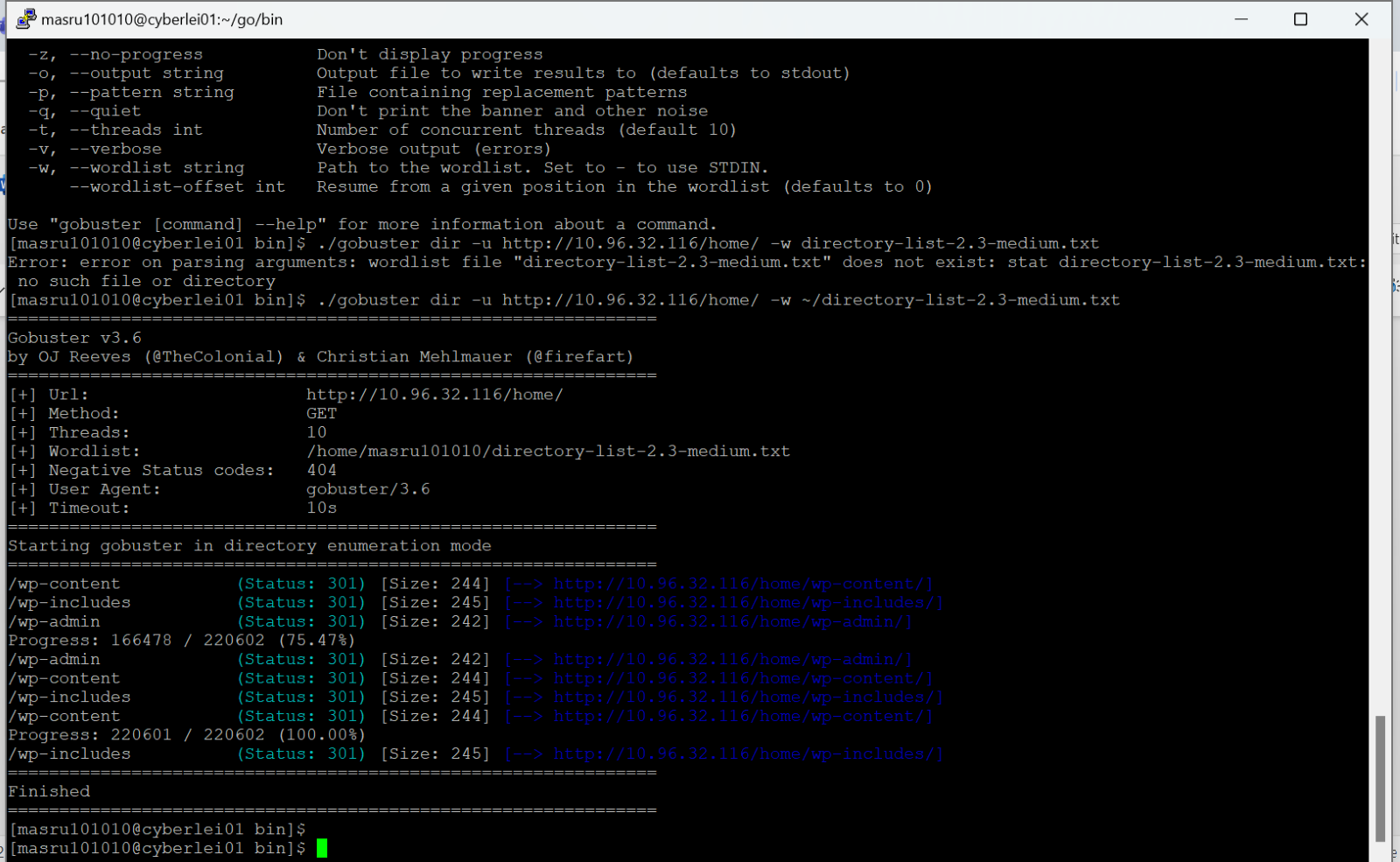
## Appendices 8 - Open VAS

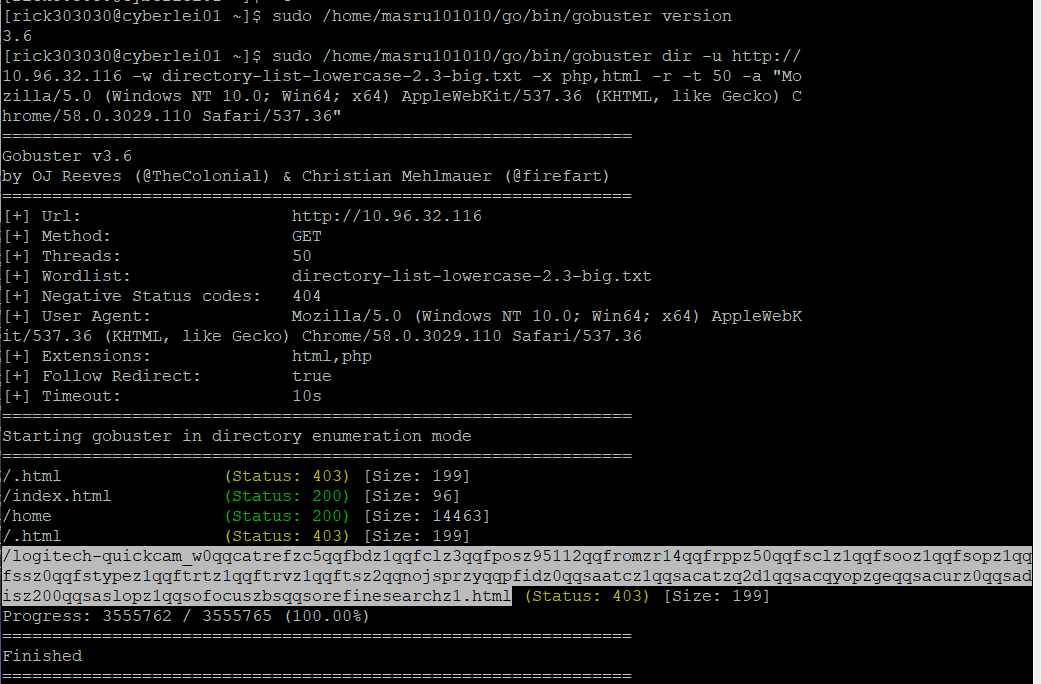
  
Open VAS (Greenbone Security Assistant) displaying issues at 10.96.32.116

* Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)
* Weak Encryption Algorithm(s) Supported (SSH)

## Appendices 9 - Gobuster

Gobuster executed against Group 3 VM

Directories identified on Group 3 VM by running Gobuster.

  
Files identified on Group 3 VM by running Gobuster.

## Appendices 10 - SQLMap

## SQLMap executed against Group 3 VM but the attempt was unsuccessful.

## Appendices 11 - XSSer

XSSer executed against Group 3 VM but the attempt was unsuccessful.

