Cybersecurity Analyst Challenge 4: Server Security 2.0 V2

As discussed in the previous Challenges, cybersecurity analysts are responsible for server security. Their job involves testing an array of servers and types of security in order to discover weaknesses and eliminate them, protecting their organization's data from mal use. This testing will include conducting penetration tests on any firewalls your server may have, discovering vulnerabilities and making recommendations to strengthen server security.

Challenge Objective

Strengthen firewall defenses by configuring security protocols and identifying vulnerabilities.

Challenge Description

Continuing the previous server security Challenge, test the security of a server firewall. Configure two-factor authentication as well as Wordfence, and then run your penetration test. Be sure to accurately synthesize and present your findings.

1. Configure two-factor authentication on WordPress admin login.
2. Configure Wordfence.
3. Configure firewall plugin.
4. Run the penetration test.
5. Extract the report.
6. Synthesize information.
7. Create a presentation highlighting key areas.
8. Record or screenshot your test completion notification.
9. Upload your test results to Files.
10. Think and answer (within the Project):
    a. What did you learn?
    b. How can you improve your skills?
    c. What would you have done differently?

Instructions

1. Start the Challenge.
2. Select the Project Pitch Template.
3. Follow the Task Checklist and make sure to do each step - the Resources provided have all the information needed to learn each task.
4. Track your work for each step.
5. Submit your Project and move on to the next Challenge.

Challenge Tasks

- Review Challenge Description
- Review resources associated to the Challenge
- Research other sources if needed (ie: Google)
- Follow outlined steps in the Challenge
- Answer Challenge Questions
- Document your process and upload necessary photos and/or videos to the 'Project Gallery'
- Finalize and upload your solution to files
- Add any additional links related to your project

Challenge Questions

- How did you approach the Challenge?
- What did you learn?
- What skills did you need to complete the Challenge?
- How can you improve your skills?
- What tools and processes did you use?
- What roadblocks did you have, if any? Or, what would you have done differently?

Solution

Penetration Test and Security Scan
https://docs.google.com/presentation/d/1vUNoblgd4WAtjGfY_DjaicWkAEWQFpXk7pfc0CYWGEM/edit?usp=sharing

Penetration Test and Security Scan Report

1. Introduction

This report consolidates the results of the **penetration test** and **security scan** conducted on the WordPress website hosted at **http://3.96.67.101/**. The test aimed to assess the security of the site by identifying vulnerabilities, misconfigurations, and other security issues. Additionally, **two-factor authentication (2FA)** and a **firewall plugin** were implemented to strengthen the site's defenses against potential attacks.

2. Tools Used:

- **WPScan**: WordPress-specific penetration testing tool.
- **Nikto**: General web vulnerability scanner.
- **Wordfence**: Installed for firewall protection and 2FA on the WordPress login page.

3. Key Findings:

I. Previously Identified Vulnerabilities and How We Addressed Them:

1. **Missing Security Headers (X-Frame-Options, X-Content-Type-Options)**:
    - **Issue**: Both **Nikto** and **WPScan** highlighted that the **X-Frame-Options** and **X-Content-Type-Options** headers were missing, making the site vulnerable to **clickjacking** and **MIME-type sniffing** attacks.
    - **Solution**: We addressed this by adding the missing security headers in the web server configuration, improving the site's resilience against these attacks.
2. **WordPress Version Disclosure (via wp-links-opml.php)**:
    - **Issue**: **Nikto** found that the **wp-links-opml.php** script revealed the installed WordPress version, making it easier for attackers to target known vulnerabilities.
    - **Solution**: We restricted access to the **wp-links-opml.php** file and removed or hid version disclosures to prevent attackers from using this information.
3. **Unnecessary Exposed Files (license.txt, wp-app.log)**:
    - **Issue**: **Nikto** flagged the presence of **license.txt** and **wp-app.log**, which could expose sensitive information about the server configuration.

    - **Solution**: These files were either deleted or restricted from public access to minimize the risk of information leakage.
4. **Drupal Link Header**:
    - **Issue**: **Nikto** found a **Drupal Link header**, suggesting a potential misconfiguration or leftover artifact from a previous installation of **Drupal**.
    - **Solution**: Investigated and cleaned up the server configuration to remove unnecessary headers, ensuring that only relevant information is exposed.

II. New Issues Identified and Addressed:

1. **Publicly Accessible `.user.ini` File**:
    - **Issue**: The scan revealed that the **.user.ini** file was publicly accessible, which could expose sensitive configuration or system details.
    - **Solution**: The file was either deleted or restricted to prevent unauthorized access and the potential leak of sensitive information.
2. **Cookie Configuration (HttpOnly Flag Missing)**:
    - **Issue**: **Nikto** detected that the **wordpress_test_cookie** was set without the **HttpOnly** flag, making it vulnerable to **Cross-Site Scripting (XSS)** attacks.
    - **Solution**: The **HttpOnly** flag was added to the cookie configuration, preventing it from being accessed by client-side JavaScript and mitigating the risk of XSS attacks.
3. **Skipped Paths in Malware Scan**:
    - **Issue**: The security scan skipped certain paths, such as `/opt/bitnami/wordpress/licenses` and `/opt/bitnami/wordpress/tmp`, which may contain important data that needs to be scanned for malware.
    - **Solution**: Adjusted the scan settings to include these paths in future scans, ensuring a comprehensive check for malware and unauthorized changes.

4. Importance of Implemented Security Measures

Two-Factor Authentication (2FA):

The **two-factor authentication (2FA)** implemented on the **WordPress admin login page** significantly enhances security by requiring a second form of verification in addition to the username and password. This measure addresses any vulnerabilities related to **brute force attacks**, as highlighted during the **penetration test**, where the login page is typically an easy target for attackers. With **2FA** enabled, even if login credentials are compromised, unauthorized access is still prevented.

Firewall Plugin (Wordfence):

The **Wordfence firewall plugin** plays a crucial role in protecting the WordPress site from malicious traffic, including attempts to exploit known vulnerabilities in plugins and themes. By configuring the firewall, we addressed the vulnerabilities associated with excessive **login**

attempts and potential **brute force attacks** identified in the penetration test. The firewall also provides ongoing monitoring and protection against common threats, making it a vital tool for maintaining the website's security over time.

5. Conclusion and Recommendations:

Through the implementation of **two-factor authentication (2FA)** and a **firewall plugin** (**Wordfence**), we have significantly strengthened the security posture of the WordPress website. Key vulnerabilities from the **penetration test** and **security scan** were addressed, such as securing sensitive files, implementing proper headers, and configuring cookies securely. The **firewall plugin** and **2FA** work together to provide enhanced protection, reducing the likelihood of successful attacks in the future.

Final Recommendations:

- Continue regular **plugin and theme updates** to ensure that all components of the WordPress site remain secure.
- Conduct regular **penetration tests** and **vulnerability scans** to identify and fix new potential weaknesses.
- Consider additional security measures such as **SSL/TLS encryption** and **rate-limiting** login attempts to further protect the site.

1. How did you approach the Challenge?

I began by setting up the WordPress website on an AWS EC2 instance and then configured two-factor authentication (2FA) and Wordfence to secure the login page and protect the site from malicious traffic. I ran penetration tests using WPScan and Nikto to identify vulnerabilities. Based on the findings, I addressed key security issues, including missing headers, exposed files, and improperly configured cookies.

2. What did you learn?

I learned how to secure a WordPress site using two-factor authentication and a firewall plugin (Wordfence). I also gained experience in penetration testing and how to identify and address vulnerabilities in a WordPress installation. The challenge reinforced the importance of security headers and proactive monitoring to safeguard against threats.

3. What skills did you need to complete the Challenge?

To complete this challenge, I needed skills in WordPress configuration, AWS EC2 management, and basic security practices like setting up 2FA and configuring firewalls. Additionally, I required proficiency in using penetration testing tools like WPScan and Nikto to analyze vulnerabilities and improve the site's security posture.

4. How can you improve your skills?

I can improve my skills by diving deeper into web security and learning about more advanced penetration testing techniques. Practicing with a broader range of tools and gaining more experience in server hardening, network security, and incident response will help me enhance my overall security expertise.

5. What tools and processes did you use?

I used Wordfence for firewall protection and two-factor authentication, and ran WPScan and Nikto for the penetration testing phase. I also utilized AWS EC2 to host the WordPress site and configured CloudWatch for monitoring and alarms. The processes involved setting up WordPress, running security tests, applying fixes, and configuring the firewall.

6. What roadblocks did you have, if any? Or, what would you have done differently?

A roadblock I encountered was ensuring that the 2FA was properly implemented for all users and testing it thoroughly. I also faced challenges in configuring the firewall to block all malicious traffic while ensuring legitimate users could still access the site. If I had more time, I would have tested more complex attack scenarios, such as SQL injection and cross-site scripting (XSS).