Cybersecurity Analyst Challenge 3: Server Security 2.0 V2

Testing server security is an important job for a cybersecurity analyst. It is their responsibility to conduct tests, discovering any and all weaknesses within their organization's servers; by exposing these weaknesses they can then recommend how best to protect their company's data. These tests need to be thorough and involve in-depth knowledge of various technologies.

Challenge Objective

Assess security of a WordPress cloud server via penetration testing and cloud monitoring.

Challenge Description

Continuing the previous server security Challenge, follow the instructions below to test the security of a WordPress cloud system. Install WordPress and configure a cloud alarm and a cloud watch before running your penetration test. Be sure to accurately synthesize and present your findings.

1. Setup AWS EC2 instance.
2. Install WordPress.
3. Configure cloud alarm.
4. Configure cloud watch.
5. Run the penetration test.
6. Extract the report.
7. Synthesize information.
8. Create a presentation highlighting key areas.
9. Record or screenshot your test completion notification.
10. Upload your test results to Files.
11. 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

Security Penetration Test Report: WordPress Installation on AWS EC2

https://docs.google.com/presentation/d/1zxTuBgb7XAaiNzbat5vkWP7g6iEq6BfKKn9O1vLhws0/edit?usp=sharing

Penetration Test Report

1. Introduction

This report summarizes the results of the penetration tests conducted on the WordPress website hosted at: **http://3.96.67.101/**

The tests aimed to assess the security posture of the website and identify potential vulnerabilities that could be exploited by attackers. Two tools were used for the assessment: **WPScan** and **Nikto**.

2. Tools Used

- **WPScan**: A WordPress security scanner that identifies vulnerabilities, versions, themes, plugins, and other WordPress-specific security issues.

- **Nikto**: A general-purpose web vulnerability scanner that checks for security misconfigurations, missing headers, and other web application weaknesses.

3. Summary of Findings

I. Server and Web Application Information

1. **WordPress Version**

    - WPScan identified that the website is running **WordPress version 6.7.1**, the latest version as of November 2024.

    - This indicates that the website is up to date and has the latest security patches.

2. **Tool Used**: WPScan

3. **Server Information**

    - Both WPScan and Nikto identified the web server as **Apache**.

4. **Tools Used**: WPScan and Nikto

II. Security Misconfigurations

1. **Missing Security Headers**

    - Nikto identified the absence of the following security headers:

        - **X-Frame-Options**: Helps prevent clickjacking attacks.

        - **X-Content-Type-Options**: Prevents MIME-type sniffing.

    - **Tool Used**: Nikto

    - **Recommendation**: Implement both headers to improve protection against clickjacking and MIME-type sniffing.

2. **Drupal Link Header Found**

    - Nikto found a **Drupal Link header** pointing to `wp-json`.

    - This may indicate a leftover from a previous Drupal installation or a server misconfiguration.

    - **Tool Used**: Nikto

    - **Recommendation**: Investigate the header to avoid confusion or exposure of unnecessary information. Ensure headers accurately reflect the current platform.

III. Site Configuration and User Authentication

1. **robots.txt File**

    - Both WPScan and Nikto found a `robots.txt` file with restricted indexing paths like `/wp-admin/` and `/wp-admin/admin-ajax.php`.

   - These are standard entries, but care should be taken to avoid exposing sensitive directories.
   - **Tool Used**: WPScan and Nikto

2. **WordPress Theme and Plugin Configuration**
   - WPScan identified the use of the **Twenty Twenty-Five** theme, which is up to date.
   - Nikto did not detect any plugins, suggesting either no plugins are installed or they are hidden from detection.
   - **Tool Used**: WPScan (theme); Nikto (plugins)
   - **Recommendation**: Regularly update plugins and themes. Use tools like Wordfence to monitor plugin security.

IV. Vulnerability and Backup Files

1. **No Configuration Backups Found**
   - Both tools confirmed that no config backups were accessible.
   - **Tool Used**: WPScan and Nikto
   - **Recommendation**: Continue to ensure that no sensitive backups are publicly accessible.
2. **WP-Cron Accessibility**
   - WPScan indicated that `wp-cron.php` is externally accessible. This could make the site vulnerable to DDoS or other misuse.
   - **Tool Used**: WPScan
   - **Recommendation**: Secure or disable `wp-cron.php` if not required.

V. Findings from Unusual Headers

1. **x-redirect-by Header**

    - Nikto detected the `x-redirect-by` header, indicating that redirections are being managed by WordPress.
    - **Tool Used**: Nikto
    - **Recommendation**: Ensure only necessary redirects are exposed and properly managed.

4. Conclusion

Both WPScan and Nikto provided valuable insights into the security configuration of the website. Key findings include:

- The site is running the latest WordPress version.
- Critical security headers such as **X-Frame-Options** and **X-Content-Type-Options** are missing and should be implemented.
- A **Drupal Link header** was found and should be investigated.
- The site has no exposed configuration backups, which is a positive sign.
- Ongoing monitoring and regular updates of themes, plugins, and server configurations are essential to maintaining security.

5. Recommendations

- Implement **X-Frame-Options** and **X-Content-Type-Options** headers.
- Investigate and remove any **Drupal Link headers** not intended to be exposed.
- Secure or disable **wp-cron.php** if unnecessary.
- Regularly review and update **WordPress plugins and themes**.
- Continue ensuring that **no configuration backups** are publicly accessible.

1. How did you approach the Challenge?

I started by setting up a WordPress website on an AWS EC2 instance. After the installation, I configured CloudWatch to monitor critical system metrics such as CPU utilization, disk space, and network traffic. Then, I set up CloudWatch alarms to be notified of any unusual activity. For the penetration testing, I used WPScan and Nikto to scan the website for vulnerabilities and security misconfigurations.

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

I needed skills in cloud computing (specifically with AWS EC2), WordPress installation and configuration, and a basic understanding of security best practices for web applications. I also had to be familiar with penetration testing tools like WPScan and Nikto, as well as CloudWatch for monitoring and alerts.

3. What tools and processes did you use?

Tools used:

- AWS EC2 for hosting the WordPress website
- CloudWatch for monitoring the website's performance and setting up alarms
- WPScan and Nikto for penetration testing
- WPScan API for deeper vulnerability analysis (optional)

Processes:

- Set up the website
- Configured monitoring
- Performed scans
- Analyzed the results to identify areas for security improvement

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

A roadblock I faced was ensuring that CloudWatch was properly configured to monitor the WordPress site and not just the EC2 instance itself. I also ran into some issues with the penetration testing, such as missing plugin detection in WPScan and needing more time to investigate the Drupal Link header found by Nikto.

If I had more time, I would have explored additional layers of security, such as firewall configurations and more in-depth testing of two-factor authentication (2FA) on WordPress.

5. What did you learn?

I learned how to set up WordPress in the AWS cloud environment and configure CloudWatch for performance monitoring and Cloud Alarms for proactive security alerts. I also gained hands-on experience in penetration testing using WPScan and Nikto, helping me identify potential vulnerabilities in a WordPress site.

6. How can you improve your skills?

To improve my skills, I can practice setting up more complex monitoring scenarios using CloudWatch and integrating it with other AWS services like Auto Scaling. I could also dive deeper into security testing, focusing on manual exploitation and advanced penetration testing strategies. Learning more about AWS security features and using additional security tools would also enhance my capabilities.