Network Assessment Project Report

Penetration Testing on Metasploitable 2 using OpenVAS

V1.0

November 2023

By: Washif Akhtar

Vulnerability Identification

1. PHP Multiple Vulnerabilities - 02 - Sep16 (Linux)
   - **Severity**: High
   - **Potential Impact**: These vulnerabilities could allow remote code execution, denial of service, or information disclosure.
   - **Recommended Mitigation Strategies**:
     - Update PHP to the latest version to patch these vulnerabilities.
     - Regularly apply security patches and updates to PHP to stay protected against new vulnerabilities.
     - Implement strong input validation to prevent code injection attacks.
2. ProFTPD Server SQL Injection Vulnerability
   - **Severity**: High
   - **Potential Impact**: This vulnerability could allow remote attackers to execute arbitrary SQL commands.
   - **Recommended Mitigation Strategies**:
     - Update ProFTPD to the latest version to patch this vulnerability.
     - Use prepared statements or parameterized queries to prevent SQL injection attacks.
     - Implement proper input validation and sanitization to mitigate the risk of SQL injection.
3. VNC Brute Force Login
   - **Severity**: High
   - **Potential Impact**: This weakness could allow remote attackers to guess the password and gain unauthorized access to the server.
   - **Recommended Mitigation Strategies**:
     - Implement account lockout policies to prevent brute force attacks.
     - Use strong, complex passwords for VNC authentication.
     - Consider using VPNs or other secure methods for remote access instead of exposing VNC directly to the internet.
4. MySQL 'sql_parse.cc' Multiple Format String Vulnerabilities
   - **Severity**: High
   - **Potential Impact**: These vulnerabilities could allow remote authenticated users to cause a denial of service or launch unspecified other attacks.
   - **Recommended Mitigation Strategies**:
     - Update MySQL to the latest version to patch these vulnerabilities.
     - Regularly apply security patches and updates to MySQL to stay protected against new vulnerabilities.
     - Implement strong access controls and user permissions to limit the impact of potential attacks.
5. MySQL Authenticated Access Restrictions Bypass Vulnerability (Linux)
   - **Severity**: High
   - **Potential Impact**: This vulnerability could allow remote authenticated users to bypass access restrictions and execute arbitrary commands.

- **Recommended Mitigation Strategies**:
    - Update MySQL to the latest version to patch this vulnerability.
    - Review and tighten access controls and permissions within MySQL.
    - Monitor and log MySQL activity to detect and respond to any unauthorized access attempts.

6. PostgreSQL Code Injection and Denial of Service Vulnerabilities (Linux)
    - **Severity**: High
    - **Potential Impact**: These vulnerabilities could allow remote attackers to inject code and cause the server to crash.
    - **Recommended Mitigation Strategies**:
        - Update PostgreSQL to the latest version to patch these vulnerabilities.
        - Regularly apply security patches and updates to PostgreSQL to stay protected against new vulnerabilities.
        - Implement strict input validation to prevent code injection attacks.
7. PHP 'imageRotate()' Memory Information Disclosure Vulnerability
    - **Severity**: Medium
    - **Potential Impact**: This flaw could allow remote attackers to disclose sensitive information from process memory.
    - **Recommended Mitigation Strategies**:
        - Update PHP to the latest version to patch this vulnerability.
        - Review and restrict access to sensitive PHP functions and libraries.
        - Monitor PHP logs for any signs of memory information disclosure.
8. PHP 'LibGD' Denial of Service Vulnerability
    - **Severity**: Medium
    - **Potential Impact**: This flaw could allow remote attackers to conduct denial of service attacks.
    - **Recommended Mitigation Strategies**:
        - Update PHP to the latest version to patch this vulnerability.
        - Implement rate limiting and request throttling to mitigate the impact of potential denial of service attacks.
        - Monitor server performance for signs of unusual activity that could indicate an ongoing attack.
9. TWiki Cross-Site Request Forgery Vulnerability - Sep10
    - **Severity**: Medium
    - **Potential Impact**: This flaw could allow remote attackers to perform unauthorized actions on behalf of a logged-in user.
    - **Recommended Mitigation Strategies**:
        - Update TWiki to the latest version to patch this vulnerability.
        - Implement CSRF protection mechanisms, such as anti-CSRF tokens, to prevent cross-site request forgery attacks.
        - Educate users about the risks of clicking on untrusted links or submitting forms from untrusted sources.
10. Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection Vulnerability
    - **Severity**: Medium
    - **Potential Impact**: This flaw could allow remote attackers to inject arbitrary commands.
    - **Recommended Mitigation Strategies**:

    - Update affected software to the latest versions that include patches for this vulnerability.
    - Disable plaintext communication where possible and enforce the use of encrypted channels.
    - Regularly monitor network traffic for signs of unauthorized commands or unusual behavior.

11. SSL/TLS: Certificate Expired
    - **Severity**: Low
    - **Potential Impact**: This weakness could allow remote attackers to impersonate the server or intercept communication.
    - **Recommended Mitigation Strategies**:
        - Renew the expired SSL/TLS certificate with a valid one.
        - Implement a process for regular certificate renewal and monitoring to avoid future expiration issues.
        - Consider using certificate management tools to automate the renewal process and ensure timely updates.
12. SSL/TLS: Certificate Signed Using A Weak Signature Algorithm
    - **Severity**: Low
    - **Potential Impact**: This weakness could allow remote attackers to forge the certificate or compromise communication integrity.
    - **Recommended Mitigation Strategies**:
        - Replace the weakly signed SSL/TLS certificate with one signed using a stronger algorithm.
        - Follow best practices for selecting and configuring SSL/TLS certificates to ensure security and compliance.
        - Monitor SSL/TLS certificate configurations for any signs of vulnerabilities or misconfigurations.
13. VNC Server Unencrypted Data Transmission
    - **Severity**: Low
    - **Potential Impact**: This weakness could allow remote attackers to uncover sensitive data by sniffing traffic to the server.
    - **Recommended Mitigation Strategies**:
        - Enable encryption for VNC server communications to protect data in transit.
        - Use strong encryption algorithms and protocols supported by the VNC server and client.
        - Educate users about the risks of using unencrypted VNC connections and encourage the use of secure alternatives.
14. ProFTPD mod_tls Module NULL Character CA SSL Certificate Validation Security Bypass Vulnerability
    - **Severity**: Low
    - **Potential Impact**: This flaw could allow remote attackers to bypass SSL certificate validation and spoof the server identity.
    - **Recommended Mitigation Strategies**:
        - Update ProFTPD to the latest version to patch this vulnerability.
        - Review and configure SSL/TLS settings in ProFTPD to ensure proper certificate validation.

    - Monitor SSL/TLS connections for any signs of unauthorized access or certificate spoofing attempts.

15. SSL/TLS: Report Weak Cipher Suites
    - **Severity**: Log
    - **Potential Impact**: This report identifies weak cipher suites that could compromise the security of SSL/TLS communications.
    - **Recommended Mitigation Strategies**:
        - Review and update SSL/TLS configurations to disable weak cipher suites.
        - Follow best practices for selecting and configuring cipher suites to ensure strong encryption and security.
        - Regularly audit SSL/TLS configurations for any signs of weak cipher suite usage.
16. SSL/TLS: Report Medium Cipher Suites
    - **Severity**: Log
    - **Potential Impact**: This report identifies medium-strength cipher suites that could potentially compromise the security of SSL/TLS communications.
    - **Recommended Mitigation Strategies**:
        - Review and update SSL/TLS configurations to prioritize strong cipher suites over medium-strength ones.
        - Disable medium-strength cipher suites that are not necessary for compatibility with older systems.
        - Regularly monitor SSL/TLS configurations for any signs of medium-strength cipher suite usage.
17. SSL/TLS: Certificate - Self-Signed Certificate Detection
    - **Severity**: Log
    - **Potential Impact**: This report identifies the use of self-signed certificates, which could pose security risks if not properly managed.
    - **Recommended Mitigation Strategies**:
        - Replace self-signed certificates with certificates signed by a trusted certificate authority (CA).
        - Implement proper certificate management practices, including regular renewal and monitoring.
        - Educate users about the importance of trusting only valid, signed certificates to avoid security risks.

Summary

In this part of the report, we have identified several vulnerabilities related to SSL/TLS certificates and cipher suites. While the severity of these vulnerabilities is relatively low, they still pose risks to the security and integrity of the system. It is important to address these issues by updating software, configuring SSL/TLS settings properly, and following best practices for certificate management and encryption.

Mitigation Plan for Identified Vulnerabilities

1. PHP Multiple Vulnerabilities - 02 - Sep16 (Linux)
   - **Mitigation Plan**:
     - Update PHP to the latest version.
     - Regularly apply security patches and updates to PHP.
     - Implement strong input validation to prevent code injection attacks.
2. ProFTPD Server SQL Injection Vulnerability
   - **Mitigation Plan**:
     - Update ProFTPD to the latest version.
     - Use prepared statements or parameterized queries to prevent SQL injection attacks.
     - Implement proper input validation and sanitization.
3. VNC Brute Force Login
   - **Mitigation Plan**:
     - Implement account lockout policies.
     - Use strong, complex passwords for VNC authentication.
     - Consider using VPNs or other secure methods for remote access.
4. MySQL 'sql_parse.cc' Multiple Format String Vulnerabilities
   - **Mitigation Plan**:
     - Update MySQL to the latest version.
     - Regularly apply security patches and updates to MySQL.
     - Implement strong access controls and user permissions.
5. MySQL Authenticated Access Restrictions Bypass Vulnerability (Linux)
   - **Mitigation Plan**:
     - Update MySQL to the latest version.
     - Review and tighten access controls and permissions.
     - Monitor and log MySQL activity.
6. PostgreSQL Code Injection and Denial of Service Vulnerabilities (Linux)
   - **Mitigation Plan**:
     - Update PostgreSQL to the latest version.
     - Regularly apply security patches and updates to PostgreSQL.
     - Implement strict input validation to prevent code injection attacks.
7. PHP 'imageRotate()' Memory Information Disclosure Vulnerability
   - **Mitigation Plan**:
     - Update PHP to the latest version.
     - Review and restrict access to sensitive PHP functions and libraries.
     - Monitor PHP logs for any signs of memory information disclosure.
8. PHP 'LibGD' Denial of Service Vulnerability
   - **Mitigation Plan**:
     - Update PHP to the latest version.
     - Implement rate limiting and request throttling.
     - Monitor server performance for signs of unusual activity.
9. TWiki Cross-Site Request Forgery Vulnerability - Sep10
   - **Mitigation Plan**:
     - Update TWiki to the latest version.
     - Implement CSRF protection mechanisms.

- Educate users about the risks of untrusted links and forms.

10. Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection Vulnerability
    - **Mitigation Plan**:
        - Update affected software to the latest versions.
        - Disable plaintext communication where possible.
        - Monitor network traffic for signs of unauthorized commands.
11. SSL/TLS: Certificate Expired
    - **Mitigation Plan**:
        - Renew the expired SSL/TLS certificate with a valid one.
        - Implement a process for regular certificate renewal and monitoring.
12. SSL/TLS: Certificate Signed Using A Weak Signature Algorithm
    - **Mitigation Plan**:
        - Replace the weakly signed SSL/TLS certificate with one signed using a stronger algorithm.
        - Follow best practices for selecting and configuring SSL/TLS certificates.
13. VNC Server Unencrypted Data Transmission
    - **Mitigation Plan**:
        - Enable encryption for VNC server communications.
        - Use strong encryption algorithms and protocols.
14. ProFTPD mod_tls Module NULL Character CA SSL Certificate Validation Security Bypass Vulnerability
    - **Mitigation Plan**:
        - Update ProFTPD to the latest version.
        - Review and configure SSL/TLS settings in ProFTPD.
15. SSL/TLS: Report Weak Cipher Suites
    - **Mitigation Plan**:
        - Review and update SSL/TLS configurations to disable weak cipher suites.
        - Regularly audit SSL/TLS configurations.
16. SSL/TLS: Report Medium Cipher Suites
    - **Mitigation Plan**:
        - Review and update SSL/TLS configurations to prioritize strong cipher suites.
        - Disable medium-strength cipher suites not necessary for compatibility.
17. SSL/TLS: Certificate - Self-Signed Certificate Detection
    - **Mitigation Plan**:
        - Replace self-signed certificates with ones signed by a trusted CA.
        - Implement proper certificate management practices.

Summary

In this part of the mitigation plan, we have outlined strategies to address vulnerabilities related to SSL/TLS certificates and cipher suites. By updating software, configuring SSL/TLS settings properly, and following best practices for certificate management and encryption, the identified weaknesses can be effectively mitigated to enhance the overall security posture of the system.

Additional Recommendations

In addition to the specific vulnerabilities identified and their corresponding mitigation plans, it is important to consider the following general security recommendations to further enhance the overall security posture of the system:

- **Regular Security Assessments**: Conduct regular vulnerability assessments and penetration tests to identify and address any new security vulnerabilities that may arise.
- **Employee Training**: Provide comprehensive security awareness training to all employees to educate them about common security threats, best practices for secure behavior, and how to recognize and report potential security incidents.
- **Incident Response Plan**: Develop and maintain an incident response plan that outlines the steps to be taken in the event of a security incident, including roles and responsibilities, communication protocols, and recovery procedures.
- **Data Encryption**: Implement strong encryption algorithms to protect sensitive data both at rest and in transit, ensuring that data is encrypted whenever possible to prevent unauthorized access.
- **Network Segmentation**: Implement network segmentation to isolate critical systems and sensitive data from the rest of the network, reducing the impact of potential security breaches.
- **Patch Management**: Establish a robust patch management process to ensure that all software and systems are regularly updated with the latest security patches and updates to address known vulnerabilities.
- **Access Control**: Enforce the principle of least privilege by ensuring that users are only granted the minimum level of access required to perform their job functions, reducing the risk of unauthorized access and privilege escalation.
- **Monitoring and Logging**: Implement comprehensive monitoring and logging mechanisms to detect and respond to security incidents in a timely manner, including monitoring for unusual activity, unauthorized access attempts, and system anomalies.
- **Security Policies and Procedures**: Develop and enforce comprehensive security policies and procedures that cover all aspects of security, including data protection, access control, incident response, and employee conduct.
- **Regular Security Audits**: Conduct regular security audits to assess the effectiveness of security controls, identify areas for improvement, and ensure compliance with security standards and regulations.

Conclusion

In conclusion, the vulnerability analysis of the Metasploitable2 machine has identified multiple critical vulnerabilities across various components such as PHP, ProFTPD, VNC, MySQL, PostgreSQL, and SSL/TLS. These vulnerabilities range from high to low severity and pose risks including remote code execution, denial of service, information disclosure, and unauthorized access.

The mitigation plan outlines specific steps to address each vulnerability, including updating software to the latest versions, implementing strong access controls, and configuring SSL/TLS settings securely. Additionally, general security recommendations such as regular security assessments, employee training, and incident response planning are provided to enhance the overall security posture of the system.

It is crucial to prioritize the remediation of high-severity vulnerabilities to mitigate the most significant risks to the system's security. Regular monitoring, patch management, and adherence to security best practices are essential for maintaining a secure environment.

By addressing the identified vulnerabilities and implementing the recommended security measures, the overall security of the system can be significantly improved, reducing the risk of exploitation and unauthorized access.