Report 08

URL: accounts.firefox.com

Weak Ciphers Enabled

MEDIUM

The Weak Ciphers Enabled vulnerability refers to a security issue where insecure or weak encryption algorithms (ciphers) are enabled for secure communication (SSL) on a web server. SSL is a cryptographic protocol used to establish secure connections between clients (such as web browsers) and servers to protect the confidentiality and integrity of data transmitted over the network. In this vulnerability, the web server allows the use of weak ciphers, which are encryption algorithms that have known security weaknesses or are considered outdated. These weak ciphers lack the strength and robustness needed to provide adequate protection against modern cryptographic attacks.

Affected URL: https://accounts.firefox.com/

List of Supported Weak Ciphers :

- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003D)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)

Affected Components

- Web Server: The vulnerability primarily affects the web server software responsible for establishing secure connections with clients. This includes popular web servers like Apache HTTP Server, Microsoft IIS (Internet Information Services), Nginx, or Lighttpd. The weak ciphers may be enabled in the SSL/TLS configurations of these servers, making them susceptible to exploitation.

- Secure Communication (SSL/TLS): The weak ciphers are specifically related to the SSL/TLS protocols used for secure communication between the web server and clients (such as web browsers). The vulnerability affects the encryption algorithms utilized during the SSL/TLS handshake process to establish a secure connection. The weakness lies in the ciphers allowed during this encryption process.

- Client-Server Interaction: The vulnerability affects the interaction between clients (web browsers, applications) and the web server. If weak ciphers are enabled on the server, clients connecting to it may unknowingly establish connections that use these weak encryption algorithms. This puts the confidentiality and integrity of the transmitted data at risk, potentially impacting the security of sensitive user information and communications.

- Secure Sessions: The weak ciphers impact the security of SSL/TLS sessions established between the web server and clients. These sessions are crucial for transmitting sensitive data securely, such as login credentials, financial information, or personal data. If weak ciphers are allowed, the confidentiality and integrity of these secure sessions may be compromised, leading to potential data breaches or unauthorized access.

Impact Assessment

- Data Exposure: The vulnerability allows attackers to potentially decrypt SSL traffic between the web server and clients. This means that sensitive information, such as login credentials, financial data, personal details, or any other confidential data transmitted during the SSL session, could be intercepted and exposed. The impact of data exposure can range from privacy breaches to financial loss or identity theft.

- Confidentiality Breach: Weak ciphers compromise the confidentiality of the transmitted data. Attackers may be able to exploit the weaknesses in the encryption algorithms to eavesdrop on the communication, gaining unauthorized access to sensitive information. This breach of confidentiality can have severe consequences, especially when dealing with sensitive data such as healthcare records, financial transactions, or trade secrets.

- Data Integrity Compromise: The vulnerability may also lead to a compromise in the integrity of the data. Attackers can manipulate the intercepted SSL traffic, modify the contents, or inject malicious payloads. This can result in unauthorized modifications, tampering, or even the introduction of malware, potentially leading to data corruption, system compromise, or the execution of unauthorized actions on the affected system.

- Compliance Violations: The use of weak ciphers may violate industry best practices and compliance requirements. Regulatory standards, such as the Payment Card Industry Data Security Standard (PCI DSS), require the use of strong encryption algorithms to protect sensitive data. Non-compliance with these standards can result in penalties, loss of customer trust, legal repercussions, and damage to the organization's reputation.

.

Steps to Reproduce

- Identify the target web server: Determine the specific web server that you want to test for the presence of weak ciphers. This could be a popular server like Apache HTTP Server, Microsoft IIS, Nginx, or Lighttpd.

- Identify the SSL/TLS configuration: Understand the SSL/TLS configuration settings of the target web server. This includes knowing where the configuration files are located and which settings control the allowed ciphers.

- Select a scanning or testing tool: Choose a suitable scanning or testing tool that can assess the SSL/TLS configuration of the target web server. Tools like OpenSSL, SSLyze, or Qualys SSL Labs Server Test can be used to perform vulnerability assessments.

- Configure the testing tool: Set up the testing tool to scan or test the target web server for weak cipher configurations. Configure the tool to focus on the SSL/TLS settings and cipher suites specifically.

- Initiate the vulnerability scan or test: Run the scanning or testing tool against the target web server. The tool will attempt to establish a connection with the server using various SSL/TLS protocols and cipher suites to identify which ciphers are allowed.

- Analyze the scan results: Review the results generated by the scanning or testing tool. The tool will provide information on the cipher suites supported by the server and indicate any weak or insecure ciphers that are enabled.

- Confirm the presence of weak ciphers: Validate the presence of weak ciphers by verifying the scan results. Pay attention to any weak cipher suites or deprecated encryption algorithms that are detected.

- Document the findings: Record the specific weak cipher configurations that are identified during the testing process. Take note of the SSL/TLS configuration settings and any recommendations provided by the scanning or testing tool.

Proposed Mitigation or Fix

- ✓ Update and Patch: Keep the web server software, operating system, and associated components up to date with the latest security patches. This ensures that any known vulnerabilities related to weak ciphers are addressed.

- ✓ Review SSL/TLS Configuration: Evaluate the SSL/TLS configuration of your web server to identify and disable weak ciphers. Modify the configuration settings to prioritize strong encryption algorithms and disable insecure or deprecated cipher suites.

- ✓ Disable Weak Ciphers: In the SSL/TLS configuration, explicitly disable weak ciphers by removing them from the allowed cipher suites list. This ensures that the web server only supports strong and secure ciphers for secure communication.

- ✓ Implement Strong Cipher Suites: Configure the web server to support only strong cipher suites that utilize robust encryption algorithms and provide better security. Choose cipher suites that comply with current industry standards and best practices.

- ✓ Regular Security Audits: Conduct periodic security audits and vulnerability assessments to identify any weak cipher configurations that may have been inadvertently introduced or missed during configuration changes. Regular audits help maintain the integrity and security of your SSL/TLS implementation.

- ✓ Test and Validate: Perform thorough testing and validation of the SSL/TLS configuration after making changes. Use testing tools and services to verify the security and integrity of your web server's secure communication.