Ensuring Safe ECDH-ECDSA SSL Ciphers for Your Web Server

The Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS), are the backbone of secure web communication. These cryptographic protocols enable secure data exchange between a web server and its users' browsers. Choosing the right set of SSL/TLS ciphers is crucial to protect sensitive information and prevent security vulnerabilities. This article provides an in-depth guide on selecting safe ECDHE-ECDSA ciphers for your web server, along with an analysis of their strengths and weaknesses.

Understanding ECDHE-ECDSA Ciphers

ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) is an elliptic curve-based key exchange algorithm used in TLS to securely establish a shared secret key between the client and server. ECDSA (Elliptic Curve Digital Signature Algorithm) is a digital signature scheme that uses elliptic curves for key generation and signature validation.

The combination of ECDHE and ECDSA provides a robust and efficient approach for TLS server authentication and data encryption. ECDHE-ECDSA ciphers utilize a pair of keys: a private key for signature generation and a public key for encrypting data. This approach offers better performance compared to traditional RSA-based ciphers, while maintaining strong cryptographic security.

Recommended ECDHE-ECDSA Ciphers

Based on current industry best practices and security guidelines, the following ECDHE-ECDSA ciphers are strongly recommended for use on your web server:

1. ECDHE-ECDSA-AES128-GCM-SHA256
2. ECDHE-RSA-AES128-GCM-SHA256
3. ECDHE-ECDSA-AES256-GCM-SHA384
4. ECDHE-RSA-AES256-GCM-SHA384
5. ECDHE-ECDSA-CHACHA20-POLY1305
6. ECDHE-RSA-CHACHA20-POLY1305

These ciphers utilize strong cryptographic primitives, including:

- AES (Advanced Encryption Standard) for block cipher encryption
- GCM (Galois/Counter Mode) for authenticated encryption and integrity protection
- SHA256 and SHA384 for message authentication and key derivation
- Elliptic curves (secp256r1, secp384r1) for key generation and signature validation

Table 1: Recommended ECDHE-ECDSA Ciphers

Cipher	Description	Key Size	Block Cipher	AAD Length	Authenticator
ECDHE-ECDSA-AES128-GCM-SHA256	ECDSA signature, AES-128-GCM encryption	256	AES-128	12 bytes	SHA-256
ECDHE-RSA-AES128-GCM-SHA256	RSA signature, AES-128-GCM encryption	2048	AES-128	12 bytes	SHA-256
ECDHE-ECDSA-AES256-GCM-	ECDSA	256	AES-256	12 bytes	SHA-384

Cipher	Description	Key Size	Block Cipher	AAD Length	Authenticator
SHA384	signature, AES-256-GCM encryption				
ECDHE-RSA-AES256-GCM-SHA384	RSA signature, AES-256-GCM encryption	2048	AES-256	12 bytes	SHA-384
ECDHE-ECDSA-CHACHA20-POLY1305	ECDSA signature, CHACHA20-POLY1305 encryption	256	CHACHA20	-	POLY1305
ECDHE-RSA-CHACHA20-POLY1305	RSA signature, CHACHA20-POLY1305 encryption	2048	CHACHA20	-	POLY1305

Cipher Strengths and Weaknesses

1. Key Size: The recommended ciphers feature a key size of 256 bits, which is considered secure against current computational power and expected advancements.
2. Block Cipher: AES-128 and AES-256 are both strong block ciphers with robust security properties. AES-256 provides additional security, but the performance gain over AES-128 is mitigated by modern hardware acceleration.
3. Authenticated Encryption: GCM and CHACHA20-POLY1305 provide both confidentiality and authenticity through authenticated encryption, ensuring that data is properly encrypted and integrity is maintained.
4. Signature Scheme: ECDSA signature schemes offer faster key generation and signature verification compared to RSA, while maintaining comparable security levels.
5. Elliptic Curves: The use of secp256r1 and secp384r1 elliptic curves provides a suitable trade-off between security and computational efficiency.

In summary, the recommended ECDHE-ECDSA ciphers offer a robust security profile, combining strong key exchange, encryption, and authentication mechanisms. They are suitable for modern web servers and provide a high level of protection against various attacks, including:

- Man-in-the-middle (MITM) attacks
- Downgrade attacks
- BEAST-like attacks
- CRIME attacks (due to the absence of padding oracle attacks on GCM and CHACHA20-POLY1305)

Ciphers to Avoid

The following ciphers are weaker and should be avoided in favor of the recommended ECDHE-ECDSA ciphers:

- DHE-RSA-AES128-GCM-SHA256 (uses RSA instead of ECDSA)
- DHE-RSA-AES256-GCM-SHA384 (uses RSA instead of ECDSA)
- DHE-RSA-CHACHA20-POLY1305 (uses RSA instead of ECDSA)

These ciphers rely on DHE (Diffie-Hellman Ephemeral) key exchange, which is less efficient and more vulnerable to attacks compared to ECDHE. Additionally, they utilize RSA signatures, which can be slower and less secure than ECDSA in certain scenarios.

Configuring Your Web Server

To implement the recommended ECDHE-ECDSA ciphers on your web server, you'll need to configure your SSL/TLS settings according to the specific server software you're using. Here are general guidelines for popular server platforms:

- Apache: Use the `SSLCipherSuite` directive to specify the preferred cipher order. For example:

```
SSLCipherSuite ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305
```