Suites de cifrado TLS/SSL (Parte I)

Por mikiminoru el 14 sept 2020 con 1 comentario

Meowy buenas! A la hora de configurar o analizar suites de cifrado en un servidor SSL/TLS nos encontramos diferentes parámetros a tener en cuenta, pero ¿sabemos qué significa cada parámetro o cuál es la solución más apropiada según nos aplique?, ¿cómo comprobar si nuestra solución realmente es segura o si convendría aplicar otras?, ¿entendemos las vulnerabilidades asociadas a estas malas configuraciones? Tanto si estamos en el equipo rojo o en el azul, puede ser interesante conocer estos detalles.

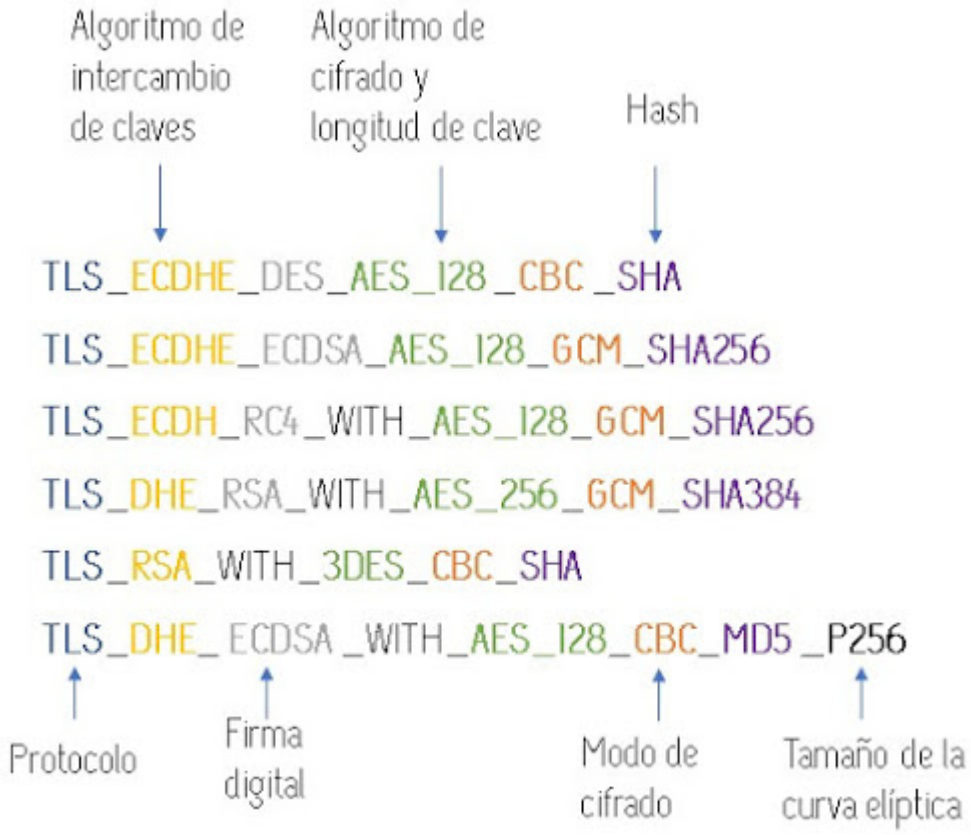
En este primer post nos vamos a centrar en los conceptos y aspectos a tener en cuenta para comprenderlo (que no pensemos *¿y esto cómo se comía?*).

Empecemos por el concepto clave, ¿qué es una suite de cifrado? Es un conjunto de algoritmos de cifrado, cada uno con una función determinada, y que en su conjunto permiten establecer comunicaciones cifradas entre un cliente y un servidor. Antes de poder realizar la conexión establece el SSL handshake, en el que:

1. El servidor deber tener configuradas un conjunto de suites de cifrado que acepta.
2. El cliente le envía aquellas versiones de TLS/SSL, suites de cifrado y métodos de compresión que soporta, en orden de preferencia (a.k.a. *ClientHello*).
3. El servidor escoge entre ellas la suite más favorable para comenzar a cifrar los datos (a.k.a. *ServerHello*).
4. El servidor envía su certificado.
5. Con la verificación del certificado (y, por tanto, la identidad del servidor), se negocia una clave secreta llamada *Master Secret*, teniendo en cuenta la suite de cifrado escogida.
6. El cliente envía un mensaje cifrado al servidor.
7. El servidor verifica que el MAC (ojo, en este caso las siglas significan *Message Authentication Code*, usado para la autenticación) es correcto y el mensaje

puede ser descifrado correctamente.
8. El servidor responde al mensaje, el cual es verificado también por el cliente.

Como hemos dicho previamente, una suite de cifrado está formada por varios algoritmos, de modo que el siguiente paso es descomponer una suite de cifrado. Para entenderlo mejor, vamos a tomar los ejemplos que vemos en la siguiente imagen (partiendo de TLS, ya que el uso de SSL lo consideramos obsoleto):



A continuación vamos a describir cada uno de los elementos con más detalle:

- **Protocolo**. Nos indica el tipo de protocolo a usar, siendo SSL (ya obsoleto) o TLS, con sus correspondientes versiones.
- **Algoritmo de intercambio de claves (Key Exchange)**. Algoritmo a emplear para compartir las claves simétricas con las que se cifrarán las comunicaciones.
- **Firma digital**. Verifica las identidades tanto del cliente y como del servidor durante la sesión. En este punto se debe hacer mención que el algoritmo RSA puede hacer función tanto de algoritmo de intercambio de claves como de firma digital.

- **Algoritmo de cifrado y la longitud de la clave**. Algoritmos simétricos usados para cifrar la comunicación (con la longitud correspondiente de cada algoritmo).
- **Modo de cifrado**. Se tratan de cifrados en bloque. Su uso depende del algoritmo de cifrado usado previamente.
- **Hash**. Algoritmo de cifrado irreversible que verifica la integridad de los mensajes.
- **Tamaño de la curva elíptica**. Esta opción no es obligatoria, y depende del algoritmo de intercambio de claves elegido previamente.

De forma práctica y esquematizando los conceptos anteriores, en la siguiente tabla se listan un conjunto de algoritmos asociados dentro de cada una de sus secciones correspondientes, indicando en color naranja aquellos que no se recomienda su uso (ya sea porque se encuentran obsoletos o porque se consideran inseguros), y en color verde la preferencia de su uso ante otros algoritmos.
Por otra parte, es importante recalcar que aquellos que se encuentran en color blanco dependen de la combinación que se realice con el resto de algoritmos, para que la suite resultante sea considerada insegura o no.

Protocolo	SSL2
	SSL3
	TLS1.0
	TLS1.1
	TLS1.2
	TLS1.3
Algoritmo de intercambio de claves	DH (Diffie-Hellman)
	DHE (Diffie-Hellman Ephimeral)
	ECDH (Elliptic curve Diffie-Hellman)
	ECDHE (Elliptic curve Diffie-Hellman Ephimeral)
	RSA (Rivest, Shamir y Adleman)
	DSA (Digital Signature Algorithm)
Firma digital	RSA
	ECDSA (Elliptic Curve Digital Signature Algorithm)
	DSS (Digital Signature Standard)
	AES (Advanced Encryption Standard – 128, 192, 256)
	RC2
	RC4
	RC5
	DES (Data Encryption Standard)
	3DES
	BLOWFISH
	CAMELLIA (128, 256)

	CAMELLIA (128, 256)
Modo de cifrado	CBC
	GCM
	EAX
	CCM (CBC-MAC)
	ECB
	PCBC
	CFB
	OFB
	CTR
Hash	SHA (Secure Hash Algorithm)
	SHA256
	SHA384
	AEAD
	HMAC
	MD2
	MD4
	MD5 (Message-Digest Algorithm 5)
Tamaño de la curva elíptica (opcional)	P256
	P384

Adicionalmente, tenemos que tener en cuenta que hay suites definidas para TLS1.2 que no deben ser usadas en TLS1.3, además de que hay algoritmos considerados legacy que se han eliminado para el uso de TLS1.3 (como SHA-1, RC4, DES, 3DES, MD5, AES-CBC... entre otros).

Si tenéis más curiosidad, podéis encontrar el RFC de TLS1.3 en este enlace :)

En la siguiente parte de este hilo veremos diferentes formas para comprobar las suites aceptadas por un servidor.

Muchos maullidos!

M

SSL , TLS

Otros artículos relacionados