Explicación OpenSSL y Qrencode

Ing. Jesús Antonio Ramírez Loza, Anterior Jefe de Departamento de Seguimiento y Evaluación
Dirección de Profesionalización, Subsecretaría del Ramo, SFA Michoacán

Morelia, Michoacán a 24 de febrero de 2022

Sellos digitales

OpenSSL

Se usa este programa de herramientas criptográficas debido a su comando **rand**, que genera una cadena o secuencia de caracteres aleatorios y da la posibilidad de elegir distintos tipos de codificación. Se usa de la siguiente forma:

$ openssl comando [opciones] [argumentos]

Algunos ejemplos:

Codificación	Comando	Resultado
Base64	$ openssl rand -base64 12	31C+rBXBA0M2Z/qc
Hexadecimal	$ openssl rand -hex 12	c47f81bd929feb95cc484c2a

Entonces usamos el programa openssl con su comando rand, que genera cadenas de caracteres aleatorias y en cada caso se establece el tipo de codificación (hex | base64) como opción y la longitud de cadena (12) como argumento. El argumento puede ser tan grande como se requiera; útil para usarse como identificación de operaciones únicas.

Codificación

- Hex está limitado a 16 posibilidades para cada carácter [0-9A-F]
  - Este arroja hexadecimales de dos caracteres, es por ello que observamos una cadena de 24 cuando el argumento fue 12. Indistintas mayúsculas y minúscula.
- Base64 tal como su nombre indica, tiene 64 posibilidades [0-9a-zA-Z+/](Relleno =)
  - El argumento de longitud se comporta de una forma distinta y se explica más adelante.

Los posibles resultados para ambos crecen exponencialmente conforme aumenta el tamaño de la cadena, por lo que generar resultados duplicados resulta muy improbable. Fabricar una cadena de 8 caracteres en Base64 puede tener más de 281 billones de resultados únicos (64^8 = 281,474,976,710,656).

Detalles sobre Base64

Este tipo de codificación crea cadenas de caracteres en múltiplos o **bloques de 4**. Cuando no se llena un bloque, se añaden caracteres de relleno **(=)**. Este comportamiento se ilustra en la siguiente tabla:

argumento	resultado	argumento	resultado	observación
1	/Q==	4	QotSmw==	2 (=)
2	Wvu=	5	2YF+bxs=	1 (=)
3	Oa/M	6	++kfvpLt	0 (=)

La recomendación es seleccionar argumentos de longitud que sean **múltiplos de 3**, lo que **maximiza la complejidad de la cadena**. Otros valores resultan en una cadena que mide (y pesa) lo mismo, pero de la cual su terminación ya está definida. Cuando la cantidad de posibles cadenas resultantes normalmente sería $\mathbf{64^n}$ (*n* = No. de caracteres), se convierte en:

$$64^{(u-r)} + r$$

Siendo *u* los caracteres únicos y *r* los de relleno (=).

Ecuaciones del argumento de longitud

El resultado final al crear estos sellos digitales Base64 tiene una conducta predecible que se puede explicar mediante ecuaciones.

Para determinar el tamaño de la cadena, por ejemplo: **longitud = (4 * ((argumento + 2) // 3))**

Nota: "//" es el operador división de piso, realiza la división normal y trunca todos los decimales que resulten, por lo que su resultado es un número entero.

- Para determinar cuántos caracteres de relleno (=) habrá al final:

rellenos = (2 - ((argumento + 2) % 3)) **# Opción 1**

rellenos = (2 - ((argumento - 1) % 3)) **# Opción 2**

Notas: Ambas ecuaciones arrojan el mismo resultado. "%" operador módulo, regresa el residuo de la división.

Repetir comandos

Fabricar los sellos digitales uno a la vez no es práctico cuando se tiene contemplado que miles de participantes acrediten cada año. La terminal tiene una manera sencilla de solucionar este problema y hacer escalable el proceso; cualquier comando se puede realizar un número determinado de veces y se hace de una de las siguientes formas:

$ for i in {1..10}; do *alguna_acción*; **done**

$ for i in `seq 10`; do *alguna_acción*; **done**

Notas: **seq** es un programa sencillo que sirve para imprimir una secuencia de números.
Importante usar el acento grave (`) alrededor del comando seq.
Entonces, para generar diez mil cadenas únicas de ocho caracteres su puede usar el comando:

usuario@máquina: constancias $ for i in `seq 10000`; do openssl rand -base64 6 >> sellos.txt; done

Aunque es improbable que en estos diez mil sellos haya duplicados, podemos cerciorarnos de ello al usar otros dos programas de la terminal:

$ sort sellos.txt | uniq -d

sort, que organiza las líneas del archivo en orden alfabético sin modificarlo.

uniq, que sirve para reportar líneas únicas; la opción (-d) cambia ese comportamiento para reportar solo las líneas duplicadas. Si no hay líneas repetidas no muestra ningún resultado.

Pasar sellos a Google Drive

Al fabricar sellos digitales, aproximadamente uno de cada 64 va a iniciar con un signo de más (+). Hay que tener esto muy en cuenta ya que tanto la base de datos de participantes como las hojas de cálculo de Excel/Calc, interpretan esto como una fórmula. La solución es poner un **apóstrofe** (‘) antes de todas las cadenas elaboradas antes de copiar y pegar en el control de participantes.

Abriendo el documento de (sellos.txt) en **neovim**, o a través de **ranger** si ya está configurado, esto se puede hacer muy sencillamente al teclear dos puntos (:) para abrir un cursor en la parte inferior de la terminal y teclear el comando:

:%s/^/’/g <Enter>

Se puede leer como:

- (%) en todas las líneas del archivo.
- (s) ejecuta el comando sustituir.
- (^) alias para el inicio de la línea antes del primer carácter. Este es el texto a sustituir.
- (‘) apóstrofe. Este es el texto con el que se sustituye.
- (g) global. Ejecuta la instrucción en todo el archivo, no te detengas después del primer resultado.
- (/) separador de los elementos/partes del comando de sustitución.

Por supuesto habrá que cerrar y guardar el archivo con (ZZ) para que se conserven los cambios.

Códigos QR

Qrencode

Programa que codifica texto en forma de Códigos QR (o Quick Response Code) que posteriormente pueden ser escaneados con cualquier dispositivo móvil para decodificarlo; ya sea la URL de un sitio web, un formulario digital o texto simple. Se usa de la siguiente forma:

$ qrencode [-t PNG] [-o archivo] [“texto”]

Nota: Las opciones entre “[]” no son obligatorias. Entre más grande es el texto de entrada, más complejo es el código QR y por consiguiente más grande (ocupa más espacio).

Opciones del comando:

- -t, tipo de imagen generada
  - Posibilidades: (PNG, PNG32, EPS, SVG, XPM, ANSI, ANSI256, ASCII, ASCIIi, UTF8, ANSIUTF8)
  - Se puede omitir y se asume a partir de la extensión del archivo resultante
- -o, archivo resultante (o ruta al archivo resultante)

Ejemplo de uso:

$ qrencode -o michoacan.png “https://michoacan.gob.mx/”

Qrencode tiene un comportamiento inconsistente al codificar texto con caracteres especiales como:

(ñ Ñ á Á é É í Í ó Ó ú Ú).

Al ser escaneado, el código QR muestra el texto con símbolos de otros idiomas; el problema solo se presenta si el primer caracter especial no es bien codificado; según mis observaciones esto sucede debido a que el caracter especial antecede a otro normal. Son casos muy particulares, como cuando una palabra tiene un acento intermedio: Dirección, Profesión, …

Una solución muy sencilla, que se puede implementar de forma global es insertar un carácter especial al inicio de cada cadena de texto, seguido de un espacio. Así se garantiza que este sea interpretado correctamente. Los siguientes tres comandos son ejemplos de cómo evitar una codificación incorrecta:

$ qrencode -o image.png '° Dirección de Profesionalización'
$ qrencode -o image.png 'ô Dirección de Profesionalización'
$ qrencode -o image.png 'í Dirección de Profesionalización'

Posibles usos

Los códigos QR pueden tener un gran número de funciones distintas. En el ejemplo anterior lo usamos para dirigir a la página oficial del Gobierno del Estado. En la Dirección de Profesionalización al día de hoy tiene dos funciones principales:

- Dirigir a formularios digitales con un Diagnóstico de Necesidades de Capacitación (DNC).
- Validar la autenticidad de las Constancias Digitales.

Incluir el código directamente en los oficios para llevar a cabo nuestro DNC permite recabar las respuestas en tiempo real, sin necesidad de regresar físicamente a la dependencia. Además de facilitar la participación de los interesados, ya que solo tienen que escanear el código y llenar el formulario.

DNC Plantilla	Validación de constancia

Estos códigos pueden tener muchas utilidades y llevar a páginas web, a registros, listas de asistencia, diagnósticos o texto simple. El impacto económico, ambiental y la disminución en los riesgos de salud (al disminuir el contacto de varios trabajadores con las mismas hojas de papel) es real, y sus aplicaciones están limitadas solo al conocimiento y creatividad de quien los implementa.