Preparación para el curso de Criptografía

Fernando Castaneda G.

Marzo 2020

1 Introdución

En un par de meses, si todo sale bien, nos estaremos viendo para este curso. Los siguientes ejercicios estan enfocados a reforzar conocimiento olvidado o no obtenido (entiendo las deficiencias que tiene nuestro sistema educativo).

Los ejercicios presentados no son obligatorios, sin embargo, facilitarán el curso y marcarán la diferencia entre sus habilidades profesionales y las de su competencía. La notación en la criptografía parece complicada pero es solo eso, notación.

Hasta entonces... Hack the Planet!
Dudas a: 6665726e616e646f@gmail.com

Figure 1: Foto actual de mi hogar, casualmente por defecto en la plantilla

2 Teoría de Conjuntos

- Esta sección los preparará para la noción de campos. Todo en criptografía (y en la vida) se puede representar con conjuntos

- Dados los siguientes conjuntos, describir en un lenguaje comprensible (español) el significado y desarrollar de ser posible. El resultado puede ser el conjunto vacio en caso de que no se cumpla la condición.
  - $\{x \mid x \in \mathbb{N}\}$
  - $\{K \mid K \subseteq \mathbb{Q}\}$
  - $\{(m,n) \mid m \in \mathbb{R}, n = 0\}$ (Graficar)
  - $\{n^2 \mid n \in \mathbb{N}\}$
  - $\{\mathbb{Z} \mid \mathbb{Z} \subseteq \mathbb{N}\}$
  - $\{n^2 \times m^2 \mid n, m \in \mathbb{N}\}$ (Graficar)
  - $\{A \mid A \in A, A \notin A\}$
  - $|\{\{\}\}|$
- Dados $F_1 = \{a, b, c\}, F_2 = \{d, e, f\}, F_3 = \{a, b, c, d, e, f\}$
  - $\bigcup_{i=\{1,2,3\}} F_i$
  - $\bigcap_{i=\{1,2,3\}} F_i$
  - $\bigcup_{i=1}^{2} F_i$
  - $\bigcup_{i=\mathbb{N}} [i, i+1]$ (Graficar)
  - $\bigcap_{i=-1}^{1} \mathbb{R} \times [i, i+3]$
  - $\{a - qb | q \in \mathbb{Z}, 0 \leq a - qb\}$ (Proponer n casos, dando valores a a,b. Si a = qb + r, que elemento del conjunto representa r para cada caso?)

3 Programación

- Tendrán que programar algunas cosas a nivel de bits y automatizar tareas con python durante el curso, esta sección los hará sentir mas comodos
- Realizar los siguientes programas en C (y python y powershell)
  1. FizzBuzz. Imprimir los numeros del 1 al 30, tomando como consideración: Multiplos de 3 imprimen Fizz, multiplos de 5 Buzz, multiplos de ambos imprimen FizzBuzz como sustitución del numero original.
  2. ascii2hex. Crear una función que reciba como parametro una cadena de caracteres, y devuelva su representación hexadecimal. Limitación: Usar unicamente memoria dinámica.
  3. notSoRandom. Empleando la llamada al sistema read(), obtener sizeof(short int) bytes, desde /dev/random y almacenarlos en una variable de tipo short int, imprimir el resultado. Ejecuta el programa 100000 veces empleando un script en python, almacena la salida y calcula el grado de aleatoreidad.

4. Strokes. Crear un keylogger empleando la llamada al sistema read() sobre dispositivo "event" del teclado localizado en /dev/, se deben leer $sizeof(struct\ input\_event)$ bytes por evento de tecla y almacenarse en una variable de tipo struct input_event, la cual se encuentra definida en input.h (Ejecuta este programa como root)
5. Crear un script con powershell, cuya finalidad sea ejecutar el binario nc.exe(netcat) sin ser detectado por el antivirus. Debe haber un puerto a la escucha en una maquina virtual con kali linux, y este debe recibir la salida de un comando de Windows(El que sea). Tips: [System.IO.File]::ReadAllBytes, IEX(), powercat
6. Implementar un WebService con Flask que brinde codificación en base64, base32, cifrado rot13, hashing con md5 y sha256.

   El web service debe recibir a través del metodo POST un archivo JSON con la siguiente estructura:
   {"method":"sha256","message":"test"}

   Y la respuesta debe ser en JSON con la estructura:
   {"Result":"9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08"}