

Resumen Semana 2

El reto de esta semana es conseguir la flag que habrá en el archivo comprimido que se enviará por Discord y enviársela al bot

Si no lo habéis hecho todavía, antes de nada, tenéis que enviarle a Merk por Discord un pantallazo de Kali y vuestro nick de telegram

Esta semana explicamos lo que son los **cifrados**:

- Procedimiento que utiliza un algoritmo de cifrado con cierta clave para transformar un mensaje y que sea incomprensible o difícil de comprender a toda persona que no tenga la clave secreta.

También lo que significa que un cifrado sea **simétrico**:

- El cifrado simétrico es un tipo de cifrado que utiliza una sola clave privada tanto para cifrar como para descifrar.

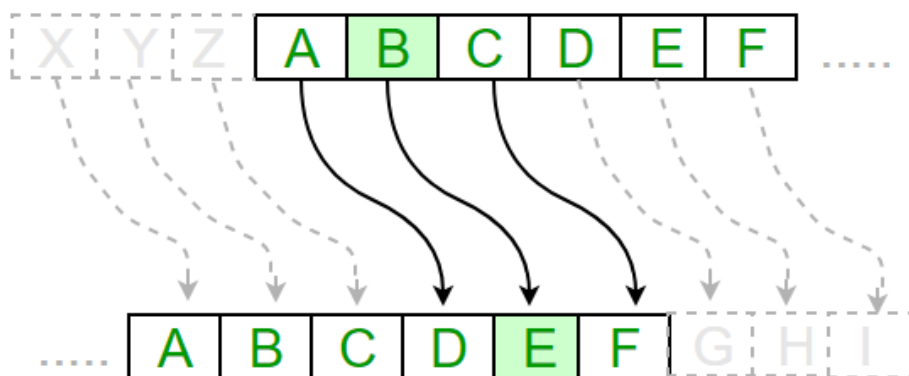
// Ya veremos más adelante como funciona el cifrado asimétrico

Vamos a ver por encima tres tipos de cifrado simétrico y un codificador muy importantes para ciberseguridad:

- Cifrado César
- Cifrado por sustitución
- Cifrado XOR
- Base64 (realmente no es un cifrado, es un codificador)

Cifrado César

El cifrado César es un cifrado por **sustitución fija**. Se elige una clave, por ejemplo (y originalmente) el número 3. Para cifrar la letra **A**, se le suma 3 y nos queda la **D**. La frase: “Ave cesar!” quedaría entonces como “Dyh fhvdu!”. Para descifrar, solo haría falta hacer la operación inversa (restar) con la misma clave (3). Este sistema tan rudimentario fue utilizado por Julio César para mandar mensajes militares.



Cifrado por sustitución

Para este cifrado se crea un nuevo abecedario con una sustitución tan arbitraria como se quiera. Como aparece en la imagen, por ejemplo, la **A** corresponde con la **M** y la **B** con la **R**.



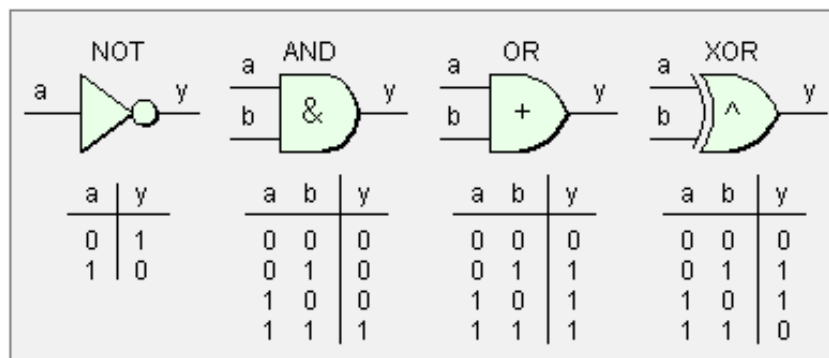
Curiosamente, el cifrado por sustitución es usado por desarrolladores de malware (virus) para ofuscar palabras o Strings.

Cifrado XOR

Para entender el cifrado XOR, hay que entender de dónde sale la operación XOR y los operadores lógicos. En la asignatura de lógica (y en algunas de computadores), conoceréis ciertas tablas que tienen esta pinta:

TABLAS DE VERDAD					
CONJUNCIÓN			DISYUNCIÓN		
p	q	$p \wedge q$	p	q	$p \vee q$
v	v	v	v	v	v
v	f	f	v	f	v
f	v	f	f	v	v
f	f	f	f	f	f

En cuanto a ordenadores y software respecta, estas tablas son MUY útiles y se utilizan con unos y ceros (bits) para entender cómo funcionan las puertas lógicas.



Con práctica en las asignaturas os acabaréis acordando de cada una de estas operaciones, pero las voy a explicar por encima (si las conocéis bien saltaros esto):

- **NOT**
 - Simplemente niega la entrada; le das un 1, te saca un 0 (y viceversa)
- **AND**
 - Solo es cierta si ambas (o todas) las condiciones son ciertas
 - Ex: si a y b son 1, saca un 1; en cualquier otro caso saca un 0
- **OR**
 - Es cierta siempre que alguna condición sea cierta
 - Ex: si a y b son 0, entonces saca un 0, pero con que una de las dos sea 1, saca un 1
- **XOR (Exclusive OR)**
 - Funciona similar a un OR pero
 - Si ambas condiciones son iguales, saca un 0; si son distintas, saca un 1
 - Es una condición lógica con propiedades MUY interesantes

Ahora que conocemos la condición XOR, y que se aplica a bits, veamos el cifrado.

Imaginemos que vamos a cifrar texto ASCII (asumimos que cada carácter son 8 bits), escogemos una clave secreta de 8 bits, por ejemplo **11110011**.

Ahora escojamos texto ASCII que queramos cifrar, como "Wiki", que es:

"01010111 01101001 01101011 01101001".

W i k i

Si utilizamos la tabla anterior, y hacemos la operación **XOR** de los bits de "Wiki" con la clave secreta repetida tantas veces haga falta:

```
01010111 01101001 01101011 01101001
⊕ 11110011 11110011 11110011 11110011
-----
= 10100100 10011010 10011000 10011010
```

Obtenemos este resultado (que quizá no sean ni caracteres ASCII). Así que hemos cifrado Wiki y nos ha salido un resultado que no podemos ni leer.

Pero lo interesante del cifrado XOR es que haciendo la misma operación sobre el resultado obtenido (y con la clave secreta), **volvemos a conseguir el texto original** "Wiki":

```
10100100 10011010 10011000 10011010
⊕ 11110011 11110011 11110011 11110011
-----
= 01010111 01101001 01101011 01101001
```

W i k i

El cifrado XOR se ha utilizado desde que existe el malware para cifrar partes importantes del proceso y hacernos a todos la vida difícil.

Base64

Por último (y muy importante), hablemos del codificador a Base64.

Base64 no es un cifrado, ya que no hay clave secreta. Es un algoritmo capaz de codificar datos binarios a texto ASCII.

- El texto
 - “Esto es base64”
- Se codifica como
 - “RXN0byBlcyBiYXNINjQ=”

Hoy en día, Base64 se utiliza para transferir cualquier tipo de datos binarios por la web.

Es importante darse cuenta de que, debido a la manera que Base64 codifica los caracteres, muchas veces deja un “padding” al final que corresponde con uno o dos símbolos de “=”.

test → dGVzdA==

testt → dGVzdHQ=

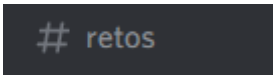
testtt → dGVzdHR0

Hay que tener algo de ojo y fijarse en cuando algo podría estar codificado con Base64.

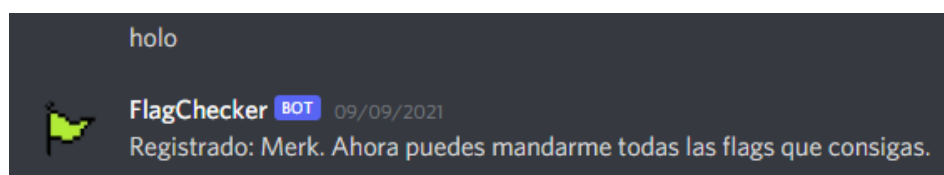
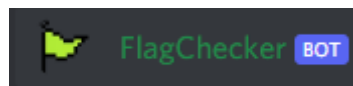
Para los retos de cifrados, recomiendo usar esta web:

[CyberChef \(gchq.github.io\)](https://gchq.github.io/CyberChef) (esta página es vuestra navaja suiza de los cifrados)

El reto de esta semana estará en Discord, en el apartado de retos. Tendréis que poner a prueba lo que habéis aprendido, y si os quedáis atascados, siempre podéis preguntar.

A dark grey rectangular button with a light grey hash symbol and the text "# retos" in a light grey font.

Además, tenéis que escribirle un mensaje al bot de Discord por privado cuando esté activo, y así os registrará.



A partir de aquí, podéis enviarle las flags (banderas) que consigáis todas las semanas. Recordad que tienen el aspecto “core{h0l4_bu3n4s}”.

Python

Por otro lado, en la presentación hemos dado lo fundamental de Python.

Antes de nada, os recomiendo tener instalado Visual Studio Code:

[Download Visual Studio Code - Mac, Linux, Windows](#)

Descargáis el archive “.deb”, vais con la terminal al directorio donde se haya instalado y ejecutamos el comando “`sudo dpkg -i nombre_del_fichero.deb`”. Después podréis abrir VSCode ejecutando el comando “`code`” en la consola.

Os dejo por aquí un par de links para que vayáis practicando un poco:

[Learn Python - Free Interactive Python Tutorial](#) (de aquí haced hasta el apartado de *Conditions*)

[TryHackMe | Python Basics](#) (de aquí haced hasta el apartado de *Introduction to If Statements*)

[CyberChef: BASE64/XOR Recipe - YouTube](#) (para aprender a usar alguna cosa de cyberchef, este tío está analizando una parte de un malware y lo usa)

Y si tenéis tiempo y os mola, haced todo lo que podáis 😊

Por ahora eso es todo, cualquier duda no dudéis en preguntar a cualquier compañero o a Llamas y Merk!

Hasta la semana que viene!