# 1. SGSSI-20.InformeLaboratorio4 (Jon Dorronsoro)

**Actividad 1.1**
url al fichero resultante:
*https://cryptpad.fr/file/#/2/file/CBRUYvCwPvmh8B+Ku5GsccFh/*

cadena prueba/relleno (ocho caracteres hexadecimales+identificador):
**30f7c0fdG161835**

Resumen MD5 resultante
**000b4a7d9690c19f2ae194c2f7ca6226**

**Actividad 1.2**

Lenguaje y librería utilizados: Python, hashlib, random, string, os, shutil, time

Fragmento de Código sustancial

```
ident = "G161835"
while(chivato):
    contador+=1
    print("\nintento numero:"+str(contador))
    #copiar el fichero con los mismos contenidos
    shutil.copyfile(file_path, 'destination.txt')
    #añadir linea con una secuencia de 8 caracteres hexadecimales
    Fuente = (random.choice(string.ascii_letters).encode('utf-8'))
    sha256_hash.update(Fuente)
    caracteres = sha256_hash.hexdigest()
    caracteres = caracteres[:8]

    #escribir en la copia los caracteres
    d_file = open('destination.txt', "a")
    d_file.write('\n'+caracteres+ident)
    d_file.close()

    #calcular resumen sha256 del destino
    d_file = open('destination.txt', "rb")
    content_d = d_file.read()
    d_file.close()
    sha256_hashD = hashlib.sha256()
    sha256_hashD.update(content_d)
    digestD = sha256_hashD.hexdigest()

    #printea los primeros X caracteres de cada resumen probado
    print("primeros "+str(secMax)+" caracteres: "+digestD[:secMax])
    if (digestD[:secMax] == '0'*secMax):#si hemos obtenido la secuencia maxima que buscamos
        #aumentar el tamaño de la secuencia maxima que buscamos
        secMax+=1
        #guardar el digest y los 8 caracteres
        digestDefinitivo = digestD
        caracteresDefinitivos = caracteres+ident
        #chivato = False

        print ("resumen destino: "+digestD)
    else:#sino eliminar y probar de nuevo
        d_file.close()
        os.remove("destination.txt")
    if(round(time.clock()-start_time, 0)==60):#si ha pasado un minuto
        chivato=False#finalizar ejecucion
    else:
        time.sleep(0.00001)#esto es para realentizar la ejecucion
        #que sino a mi portatil no le da la vida

#Al finalizar quedarnos con la secuencia mas larga
d_file.close()

shutil.copyfile(file_path, 'destination.txt')
d_file = open('destination.txt', "a")
d_file.write('\n'+caracteresDefinitivos)
d_file.close()
print ("\nresumen definitivo: "+digestDefinitivo)
print ("secuencia de caracteres: "+caracteresDefinitivos)
```

Pantallazo ejemplo de ejecución

```
intento numero:2851
primeros 3 caracteres: c71

intento numero:2852
primeros 3 caracteres: 7e0

intento numero:2853
primeros 3 caracteres: f89

intento numero:2854
primeros 3 caracteres: b9c

intento numero:2855
primeros 3 caracteres: 837

intento numero:2856
primeros 3 caracteres: 417

resumen definitivo:
00d8e685fb1fd94def901ec4844ab24b4effc5727a3998e045a85035affe8c34
secuencia de caracteres: 92228251G161835
```

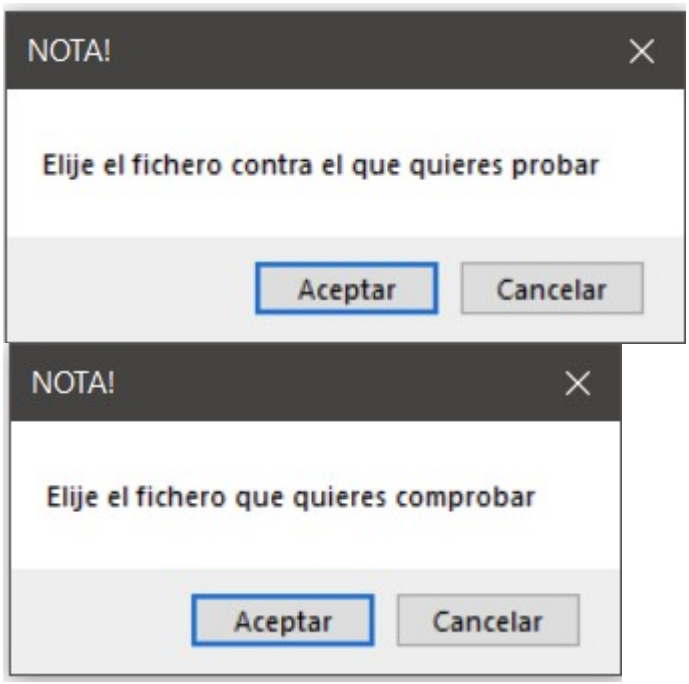
**Actividad 1.3**

Lenguaje y librería utilizados: Python, hashlib, random, string, os, shutil, time

Fragmento de Código sustancial

```
# Importar la libreria del md5
import hashlib
# Importar la libreria del seleccionador de archivos (para facilitar
# la seleccion)
from tkinter import filedialog as fd
import ctypes  # An included library with Python install.



#Aqui inicia el programa
resultado = True
md5_hash = hashlib.md5()
#aviso


ctypes.windll.user32.MessageBoxW(0, " Elije el fichero contra el que quieres probar ", "NOTA!", 1)
#pedir el archivo de texto 2
file_path2 = fd.askopenfilename()#fichero contra el que comprobar

ctypes.windll.user32.MessageBoxW(0, " Elije el fichero que quieres comprobar ", "NOTA!", 1)
#pedir el archivo de texto 1
file_path1 = fd.askopenfilename()#fichero a comprobar

#primero comprobar que el resumen md5 empiece por 0
    #leer el fichero
d_file = open(file_path1, 'br')
content = d_file.read()
    #actualizar el contenido del hash md5 en base al texto
md5_hash.update(content)
    #obtener el digest (resumen) del contenido del fichero
digestD = md5_hash.hexdigest()

if(digestD[:1]!='0'):#si no empieza por 0 no cumple la condicion
    resultado=False
    print("el resumen no empieza por 0")
else:
    print("Bien, el resumen empieza por 0")
    d_file.close()
    #comprobar que comienza por los mismos contenidos que el primero
    with open(file_path1, 'r') as file1:#leer fichero origen
        with open(file_path2, 'r') as file2:#leer fichero a comprobat
            difference = set(file1).difference(file2)

    #pasarlo a array para quitarle la ultima diferencia (para solucionar un error en el que)
    #me toma la ultima linea como diferencia
    diferenciaDef=[]
    diferenciaDef=list(difference)
    diferenciaDef.pop()
    #para mostrar las diferencias en la terminal
    for line in diferenciaDef:
        print("\nla linea diferente es: " + line)
        print("la longitud de la linea diferente es: "+str(len(line)))

    if(len(diferenciaDef)!=1):
        #si hay mas de una linea de diferencia
        print("\nHay mas de una linea de diferencia")
        resultado = False
    else:
        print("\nBien, solo hay una linea de diferencia")

        if(len(diferenciaDef[0])!=15):#si la unica linea de diferencia tiene una longitud diferente a 8 caracteres
            resultado = False
        else:
            print("\nBien, la longitud de la linea es de 8 caracteres")

print("\n\nResultado de la prueba: "+str(resultado))
```

Pantallazo ejemplo de ejecución




Si supera la prueba:

```
In [1]: runfile('A:/Desktop/4.Urtea/SGSSI/labo04_Actividad_1.3.py',
wdir='A:/Desktop/4.Urtea/SGSSI')
Bien, el resumen empieza por 0

la linea diferente es: 8d8cd475G161835
la longitud de la linea diferente es: 15

Bien, solo hay una linea de diferencia

Bien, la longitud de la linea es de 8 caracteres


Resultado de la prueba: True
```

Si no lo supera:

```
la longitud de la linea diferente es: 33

la linea diferente es: 7ac1b83f149c982fd28c6963095d1464

la longitud de la linea diferente es: 33

la linea diferente es: f9694bd9d36d5610681ebaf6961948e2

la longitud de la linea diferente es: 33

la linea diferente es: 2f07ebeb370c0f4ddce28aab55346e8e

la longitud de la linea diferente es: 33

la linea diferente es: b2530c58f735336bbb93e2037db1df51

la longitud de la linea diferente es: 33

Hay mas de una linea de diferencia


Resultado de la prueba: False
```

**Actividad 1.4 (opcional)**

Link de descarga:

https://cryptpad.fr/file/#/2/file/c7C6KJNbrhZxNm7ZRqfcudZM/

Para descargar el programa entre en el enlace y presione descargar.

Para ejecutar el programa, primero debe tener instalado python, despues acceder al terminal y escribir "python Labo04_secuenciaMasLarga.py"
El resumen MD5 del programa es el siguiente:
a75d14e1a63f98b4c1947e28ef08a29b

**Actividad 1.5 (opcional)**

Link de descarga:

https://cryptpad.fr/file/#/2/file/RykOtFPT+v6ZO2Ky5uFXkqOe/

Para descargar el programa entre en el enlace y presione descargar.

Para ejecutar el programa, primero debe tener instalado python, despues acceder al terminal y escribir "python labo04_Actividad_1.3.py"
El resumen MD5 del programa es el siguiente:
0b379ebcd43d8c0acf2e165fa2ea9e07

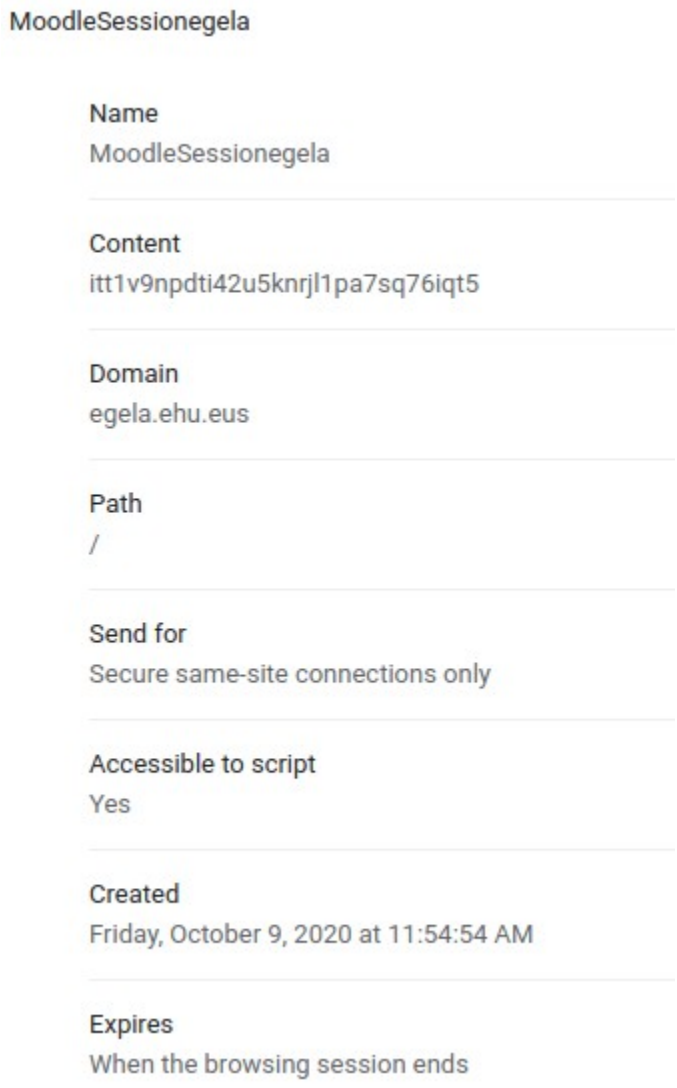
**Actividad 2.**

**Actividad 2.1 Cookies, sesión del navegador y sesión(es) en el servidor**

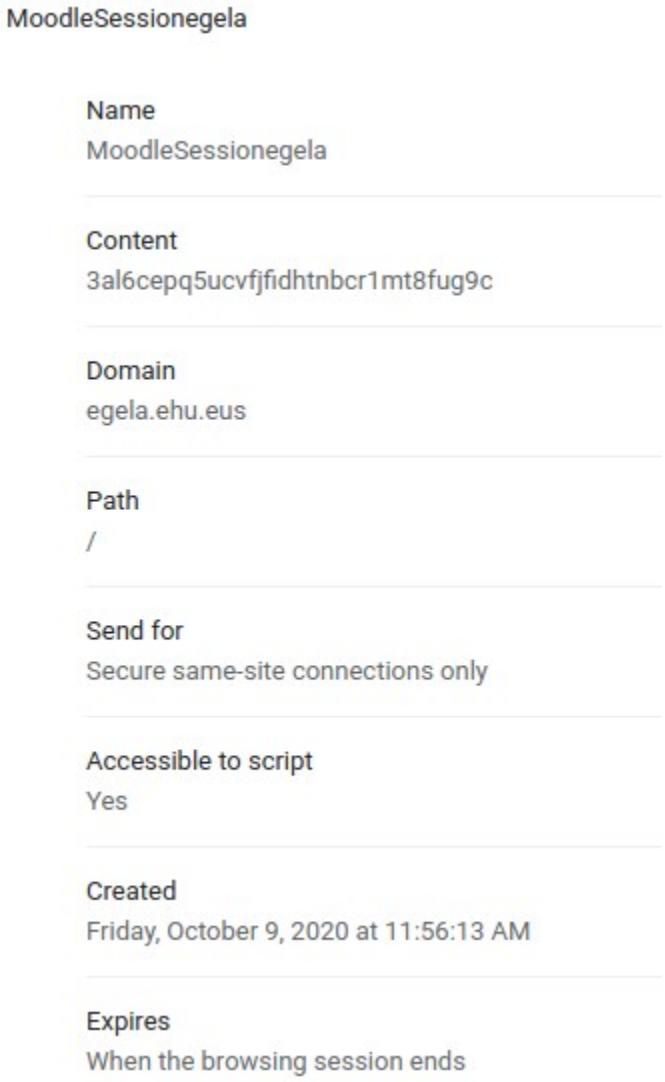
Descripción de lo observado siguiendo los pasos establecidos y conclusiones/reflexiones realizada.

Descripción de las relaciones establecidas con las páginas del sitio OWASP señaladas

*Egela (sin identificarse).* Valor: *itt1v9npdti42u5knrjl1pa7sq76iqt5*

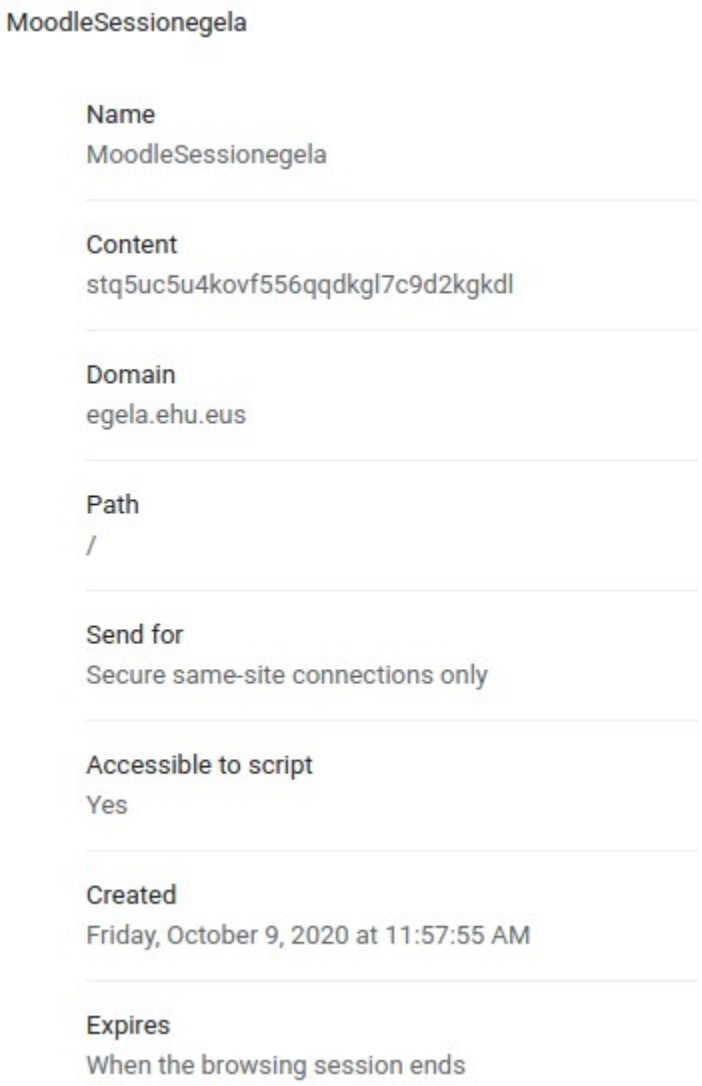


*Egela (identificada).* Valor: *3al6cepq5ucvfjfidhtnbcr1mt8fug9c*



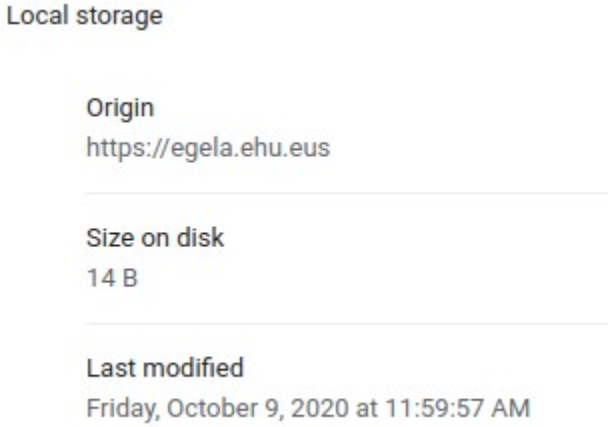

Ha cambiado completamente de valor y no solo eso, si no que la fecha de creación también distinta por lo que ha creado una nueva “instancia”.

*Egela (cerrada la sesión).* Valor: *stq5uc5u4kovf556qqdkgl7c9d2kgkdl*




Se vuelve ha generar una tercera “instancia”

Cookie de Egela despues de cerrar la sesion del navegador:



Valor de la cookie antes de cerrar el navegador sin cerrar la sesion de Egela:

*8baicit20pf50235ug0bl5alciad4dfq*

Al volver a colocar el valor anterior nos sigue accediendo a la sesion de Egela, por lo que se deduce que cerrar el navegador NO cierra la sesion de Egela.

Egela al realizar un correcto cerrado de sesion, aplica los mecanismos necesarios para que sea seguro, en cambio no los realiza cuando se cierra el navegador sin cerrar la sesion.

Por otro lado no queda ninguna ID registrada en la URL, ademas de implementar HTTPS "capando" la posibilidad de que se pueda optener mayor longitud de la URL que egela.ehu.eus.

Finalmente sí que acepta contraseñas debiles puesto que solo exige una longitud minima y alguna mayuscula, numero o caracter extraño

**Actividad 2.2 (Opcional)**

Hipótesis o conclusión sobre el tipo de representación correspondiente al valor de MoodleSessionegela

Hipótesis o conclusión sobre cómo se gestiona el identificador de sesión en webposta.ehu.eus.

Diferencias y similitudes en la gestión de las sesiones en egela y en webposta.