SGSSI-20.InformeLaboratorio5.G13151719

Actividad 1.1
url al fichero resultante→

https://drive.google.com/file/d/1AtDusRKNxKwvbJUaF0n5jQiOSA9FunGB/view?usp=sharing

cadena prueba/relleno (ocho caracteres hexadecimales + codigo de grupo) →

`b0d948b2 G13151719`

Resumen MD5 resultante →

000038239c538160fb40b101f1420b92

Apartado sustancial del código:

```python
def AlteracionFicheroMD5(F):
    #abrimos el fichero original de texto
    text = open(F)
    #guardamos en una lista las lineas escritas
    lineas = text.readlines()

    #Establecemos los parametros de inicio
    start = time.time()
    end = time.time()
    max0 = 0

    while end - start < MAXTIME :
        #eliminamos el fichero del intento anterior
        try:
            os.remove("nuevoFichero.txt")
        except :
            pass
        #Generamos un string hexadecimal aleatorio
        s = uuid.uuid4().hex
        #Creamos nuestro nuevo fichero con permisos de escribit extra 'w+'
        newFile = open("nuevoFichero.txt","a")
        #Escribimos las lineas del otro fichero ademas de añadir los 8 primeros caracteres de s
        newFile.writelines(lineas)
        #Combertimos s a un string con un casteo str('')
        newFile.write( str(s)[:8]+ " G13151719" )
        #Cerramos el fichero para guardar los cambios
        newFile.close()
        #obtenemos su resumen md5 y contamos sus 0s
        brief = md5.md5(newFile.name)
        ceros = count0(str(brief))
        #Si tenemos mas 0s que el maximo anterior actualizamos los valores del resumen y el maximo de 0s
        if ceros > max0:
            max0 = ceros
            res = brief
            #Borramos el resultado anterior si existe
            try:
                os.remove("./ficheroResultado13.txt")
            except:
                pass
            #Renombramos el fichero para no eliminarlo en la siguiente rotacion
            os.rename(r"./nuevoFichero.txt",r"./ficheroResultado13.txt")
        #Calculamos el tiempo al terminar
        end = time.time()
        #Sacamos cuanto hemos tardado en segundos
        print(end- start)

    #Eliminar el fichero basura en caso de haberlo
    try:
        os.remove("nuevoFichero.txt")
    except :
        pass

    print(res)
```

Actividad 1.2
Python es el lenguaje que hemos utilizado. Las librerías son las siguientes:
Hashlib
os
time
uuid

Fragmento de Código sustancial

```
def AlteracionFicheroSHA(F):
    text = open(F)
    lineas = text.readlines()

    s = uuid.uuid4().hex

    start = time.time()
    end = time.time()
    max0 = 0

    while end - start < 60 :

        try:
            os.remove("nuevoFichero.txt")
        except :
            pass

        s = uuid.uuid4().hex

        newFile = open("nuevoFichero.txt","a")
        newFile.writelines(lineas)
        newFile.write( str(s)[:8]+ " G13151719" )
        newFile.close()

        brief = sha.sha(newFile.name)
        ceros = count0(str(brief))
        print(ceros)
        if ceros > max0:
            max0 = ceros
            res = brief
            try:
                os.remove("./ficheroResultadoSHA.txt")
            except:
                pass
            os.rename(r"./nuevoFichero.txt",r"./ficheroResultadoSHA.txt")
        end = time.time()
        print(end- start)

    try:
        os.remove("nuevoFichero.txt")
    except :
        pass
    print(res)
```

Pantallazo ejemplo de ejecución:

```
4.975124359130859
0
4.979126214981079
0
4.983125925064087
0
4.987127065658569
0
4.9911277294158936
0
4.995129346847534
0
4.9991302490234375
0
5.002130746841431
00c67b29bf99fde60fe7a86e46e32b6228a4cffb8b25b5696157e74a7fc6335d
```

Actividad 1.3
Lenguaje: Python

Librerías utilizadas:
md5

Fragmento de Código sustancial:

```python
def isFake(f_original_file, fake_file):

    #Por defecto lo ponemos a falso, es decir no es una falsificacion
    #abrimos ambos ficheros
    f_original = open(f_original_file)
    f_falso = open(fake_file)

    #Cogemos las lineas de los dos
    original_lineas = f_original.readlines()
    fake_lines = f_falso.readlines()



    #Si no coinciden en el contenido de las lineas es un fichero falso
    if not sameLines(original_lineas, fake_lines):

        return True
    #Si la ultima linea del fichero falso no es exadecimal es falso
    if not isHex(fake_lines, len(original_lineas)):

        return True

    #Calculamos el md5
    brief = md5.md5(f_falso.name)

    #Si su resumen MD5 no empieza por 0 es falso
    if not start0(brief):
        return True

    #Si ninguno de los anteriores se cumple es un fichero valido
    return False


#Si el fichero no es falso que no lo es, si lo es printeamos que no es
print("El fichero no es falso" if not isFake("SGSSI-20.CB.06.txt","ficheroResultado13.txt") else "El fichero es falso")
```

Pantallazo ejemplo de ejecución

```
U:\Documents\Curso 20-21\SGSSI\SGSSI.lab3.cod\SSGSI-20.Rep>"C:/Users/Jon Perez/AppData/Local/Programs/Python/Python35/python.exe"
"u:/Documents/Curso 20-21/SGSSI/SGSSI.lab3.cod/SSGSI-20.Rep/comprobacionFicheros.py"
El fichero no es falso
```

Acceso al repositorio con instrucciones y el código:

GitHub
Drive

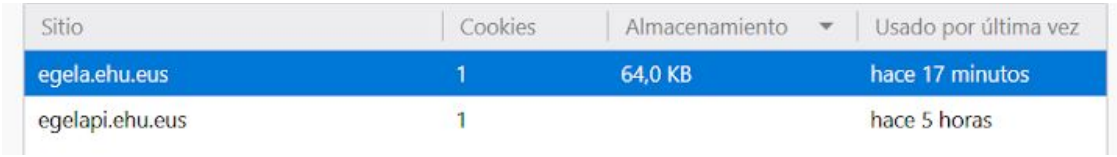
En primer lugar, acceder al sitio web de OWASP y mira información sobre *Broken Authentication* :

Autenticación rota . Las funciones de la aplicación relacionadas con la autenticación y la gestión de sesiones a menudo se implementan de manera incorrecta, lo que permite a los atacantes comprometer contraseñas, claves o tokens de sesión, o aprovechar otras fallas de implementación para asumir las identidades de otros usuarios de forma temporal o permanente.

Estos pasos han sido seguidos en el navegador firefox:

1. **Lo primero sería abrir una sesión del navegador escogido (preferiblemente uno de los utilizados en el Lab01) y comprobar si existe alguna cookie de egela (dependerá del caso).**

   -Si existían cookies



Sitio	Cookies	Almacenamiento	Usado por última vez
egela.ehu.eus	1	64,0 KB	hace 17 minutos
egelapi.ehu.eus	1		hace 5 horas

2. **Si es el caso, borrarlas.**

   -Borradas

3. **Conectarse con la página de entrada de egela (sin identificarse). Comprobar el valor de la cookie MoodleSessionegela y del resto de los atributos. Guardar aparte el valor de la cookie MoodleSessionegela.**

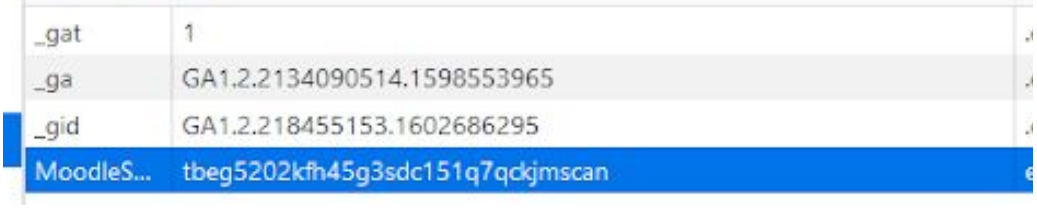
   Valor: q5b3c81bu2j7fqrdpmg2bg31mbnc5f82

4. **Identificarse/entrar en egela con tu usuario. Comprobar de nuevo el valor de la cookie MoodleSessionegela y del resto de los atributos. ¿Qué ha cambiado respecto a MoodleSessionegela?**

El valor ha sido modificado: 7djb9cd2u69ii8o52vs48h33kc5ookfm

5. **Salir de la sesión de egela. Volver a comprobar de nuevo el valor de la cookie MoodleSessionegela y del resto de los atributos. ¿Qué ha cambiado?**

   El valor vuelve a ser modificado: tbeg5202kfh45g3sdc151q7qckjmscan



6. **Cerrar (la sesión d)el navegador (no la pestaña, el navegador).**
   -Cerrado

7. **Volver a entrar en el navegador y comprobar si existe alguna cookie de egela. Si es el caso, analizarla antes de borrarla.**

   -0 cookies

8. **Conectarse con la página de entrada de egela e identificarse. Acceder al valor de la cookie MoodleSessionegela y copiarla en un fichero/recurso aparte.**

   Valor: if2l89b5ptl0hrtrfosgbr01a53rqb96

9. **Sin cerrar la sesión de egela, cerrar (la sesión d)el navegador.**

   -cerrado

10. **Volver a entrar en el navegador y conectarse con egela (normalmente debería ir a la página de identificación)**

    -vuelvo a entrar en el navegador y me quedo en la página de identificación.

11. **Localizar la cookie MoodleSessionegela y sustituir su valor actual por el que guardamos en el paso 8. Recargar la página de egela. ¿Qué sucede?**

    -Al cambiar el valor al valor de moodlesession anterior y refrescar la pagina entra en egela con el perfil que teníamos identificado en el paso 8.

describe los atributos de las cookies, intentando establecer relaciones con lo visto

Atributos:
-**Secure Attribute**
-**HttpOnly Attribute**
-**Domain Attribute**
-**Path Attribute**
-**Expires Attribute**
-**SameSite Attribute**

En cuanto a los atributos de las cookies que se comentan en OWASP, como por ejemplo Secure Attribute, se puede destacar que egela mantiene la comunicación a través de un canal seguro. HTTPS. Otra manera de darnos cuenta de esto es fijarnos en el candadito que figura en la parte izquierda del navegador.

Otro atributo a tener en cuenta es el de caducidad. En OWASP se comenta que este atributo es usado para:

- establecer cookies persistentes
- limitar la vida útil si una sesión dura demasiado
- eliminar una cookie con fuerza configurándola en una fecha pasada

En egela ese atributo figura con el siguiente valor:

Agentes de amenaza / Vectores de ataque y Ejemplos de escenarios de ataque
En la actividad 2.1.8-2.1.10 nos pedían salir del navegador sin cerrar sesión. Se puede observar que la cookie no ha caducado cuando volvemos a acceder a la página de inicio de egela, ya que no es necesario autentificarse otra vez, para acceder a los datos del usuario que ha estado conectado. Esto, puede suponer una amenaza ya que cualquiera podría acceder a ese egela si no ha pasado el tiempo de inactividad específico, para que caduque la cookie.

"Escenario n. ° 3: Los tiempos de espera de la sesión de la aplicación no están configurados correctamente. Un usuario usa una computadora pública para acceder a una aplicación. En lugar de seleccionar "cerrar sesión", el usuario simplemente cierra la pestaña del navegador y se marcha. Un atacante usa el mismo navegador una hora más tarde y el usuario aún está autenticado."

Actividad 2.2
Intenta identificar qué tipo de representación corresponde al valor de MoodleSessionegela, si no puedes asegurarlo, establece al menos alguna hipótesis.

https://moodle.com/es/politica-de-cookies/#cookie

Moodle	_revólver	Utilizado por Google Analytics para regular la tasa de solicitud.	Sesión
	_gid	Registra una identificación única que se utiliza para generar datos estadísticos sobre cómo el visitante utiliza el sitio web.	Sesión

El tipo de representación de esta cookie corresponde a una codificación alfanumérica, representada por 32 caracteres que pueden tomar valores entre el 0-9 y entre la a-z.

Combinaciones posibles : (36)^32 = 6,33e^49

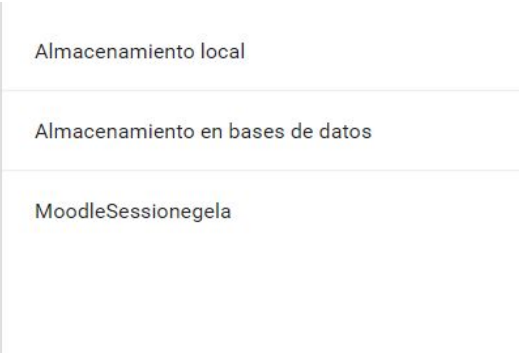
Conéctate a webposta.ehu.es. Intenta identificar cómo se gestiona el identificador de sesión. ¿Qué similitudes y diferencias encuentras respecto a egela?

Al conectarse al web posta e iniciar sesion no he encontrado ninguna moodlesession como en egela pero en cambio he encontrado estas 2 cookies:
En egela solo encontrábamos una cookie para la sesión en cambio en la web posta vemos 2 la horde y la horde secret key. (Autentificado)

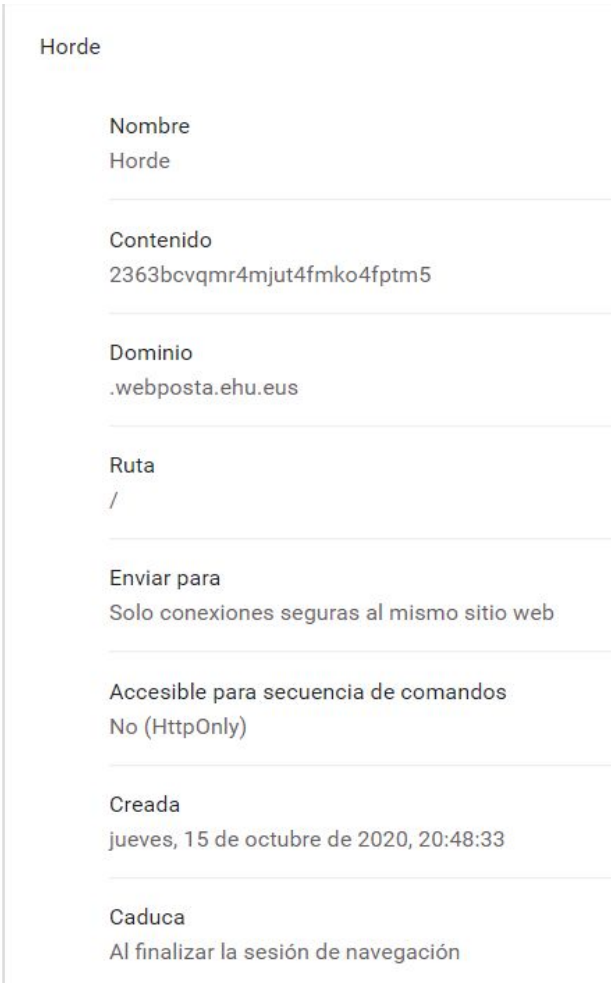
horde_secret_key	24hpc21lla0emise06c65mrnt3	.webposta.e...	/	Sesión	42	true
Horde	5vtvek68n0dfqm2odr82b4s543	.webposta.e...	/	Sesión	31	true

webposta.ehu.eus

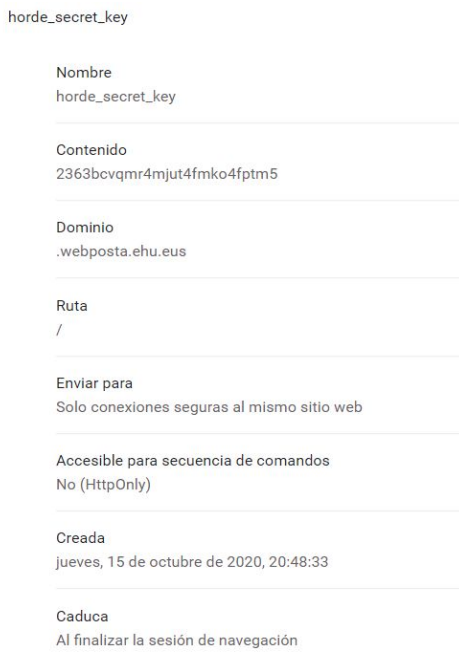
egela.ehu.eus



Horde = https://www.horde.org/apps/horde/docs/SECURITY . El contenido de la cookie cambia cuando te autentificas, antes era igual a Horde_secret_key .



Horde_secret_key = https://webcookies.org/cookie/http/horde_secret_key/61518
El valor del contenido de la cookie no cambia.



UqZBpD3n3iPIDwJU = Cookie no clasificada se almacena 10 años. Cookie Beleid
El contenido de la cookie es el mismo estando autenticado o no.