1. SGSSI-20.InformeLaboratorio4 (Jon Dorronsoro)

Actividad 0 (suprimir si ya se había realizado antes de comenzar el Lab04 y se había incluído en el informe del Lab03)

Lenguaje y librería utilizados: Java, librerias io, math y security

Fragmentos de Código sustancial:

```
for (int i = 0x0000000; i<0xFFFFFFF; i++) {
    String hex = Integer.toHexString(i);
    String lag = s + hex;
    if(getMd5(lag).startsWith("00000")) {
        System.out.println("Resumen MD5: " + getMd5(lag) + "\tHex: " + hex);
    }
```

```
public static String getMd5 (String input) {
    try {
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] messageDigest = md.digest(input.getBytes());
        BigInteger no = new BigInteger(1, messageDigest);
        String hashtext = no.toString(16);
        while (hashtext.length() < 32) {
            hashtext = "0" + hashtext;
        }
        return hashtext;
    } catch (NoSuchAlgorithmException e) {
        throw new RuntimeException(e);
    }
}
```

No hemos conseguido que se sean cadenas solo de 8 caracteres

Pantallazos ejemplo de ejecución

```
<terminated> Lab04 [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (2020 urr. 9 11:14:02 – 11:14:04)
Resumen MD5: 005b06b73fb7faadbd0d8787c158e39a   Hex: 34ba5
Resumen MD5: 004e0d2438e1aa8c0ec1ec4992ee122e   Hex: 34bd1
Resumen MD5: 00813fae68db0741464c3347afbaac99   Hex: 350af
Resumen MD5: 0089daf86ee898f6fb9fde0e2061c588   Hex: 350d2
Resumen MD5: 00960b852207c123a470e94111688ade   Hex: 35115
Resumen MD5: 002fb915107e3bf45fecdd762ffd024e   Hex: 3512b
```

Actividad 0.2 Realizar un programa que, tomando como entrada un fichero de texto, obtenga como salida una versión modificada

Lenguaje y librería utilizados: Java, la libreria io en su conjunto.

Fragmentos de Código sustancial:

```
    public static void main (String args[]) throws IOException {
        String s = "";
        FileReader f = new FileReader("src/Lab04/proba.txt");
        BufferedReader b = new BufferedReader(f);
        BufferedWriter bw = new BufferedWriter(new FileWriter("src/Lab04/proba1.txt", true));
        try {
            StringBuilder sb = new StringBuilder();
            String line = b.readLine();
            while (line!=null) {
                sb.append(line);
                sb.append(System.lineSeparator());
                line = b.readLine();
            }
            s = sb.toString();
            s = s + "proba bat da";
            bw.write(s);
        } catch (IOException e) {
        }finally {
            b.close();
            bw.close();
        }
```

Pantallazos ejemplo de ejecución:

```
<terminated> ariketa02 [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (2020 urr. 9 11:21:11 – 11:21:11)
```

Actividad 0.3 Ejecutar el programa desarrollado en la actividad anterior tomando como entrada el fichero [SGSSI-20.CB.03.txt]

Pantallazos ejemplo de ejecución:

```
24 bb2e6288f9c7f53e5246948a638f8212
25 bd739f627509f473b969234089fd7a08
26 c1d91ca80070fc4c7f89fe85042aaa4b
27 c34e154c68ed8a62c75b710da4bb45e2
28 c44263ba7b14259cc111efe4a2269c62
29 cac31bef6a84dd6e27292e37b3d3887a
30 d835c99bc9e251a1ed9714492855106b
31 db6c19ed4944c3a4edde4c5d6e6386ae
32 dbea40c0a3ebf8ab9f7ad2686a63b539
33 dbee6604099505b0f6d3b3266235c9f7
34 e17562c191c2de135fb2419f2f4471d1
35 e5e4f01f1630e39b66d74e0598a9fd4d
36 e71257f387d9f31de74876246e33b365
37 e9410755ac6825d92b93202dedd20d5a
38 f72d089843371654cb13cd4581cd34da
39 proba bat da
```

Actividad 1.1

url al fichero resultante

No hemos conseguido que se sean cadenas solo de 8 caracteres

```
<terminated> Lab04 [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (2020 urr. 9 11:12:27 – 11:12:30)
Resumen MD5: d00b2c4019049b1fb9617e916c4e3f19   Hex: 6a660
Resumen MD5: a00421a344768284c4ec3e4a96364ab0   Hex: 6a7ec
Resumen MD5: b005be7f5a37dfff2835efac4dafd296   Hex: 6a802
Resumen MD5: 30041f6d434dec771e2fe4a8e2a7c5a9   Hex: 6a86c
Resumen MD5: 200bdccd2ce61e459d79d226a4ed9184   Hex: 6a91e
Resumen MD5: b00825e87182e42d5af3882aca98e884   Hex: 6aaa3
```

Actividad 1.2

Lenguaje y librería utilizados: Java, ibrerias io, math y security

Fragmento de Código sustancial

```
for (int i = 0x0000000; i<0xFFFFFFF; i++) {
    String hex = Integer.toHexString(i);
    String lag = s + hex;
    String md5 = getMd5(lag);
    lag = md5.substring(1);
    if(lag.startsWith("00")) {
        System.out.println("Resumen MD5: " + md5 + "\tHex: " + hex);
    }
```

Pantallazo ejemplo de ejecución

```
Resumen MD5: e008f00b400d916f14c33a746285a843   Hex: 41a85
Resumen MD5: a0054d5e64630f2fc4b4e3ada71ecf19   Hex: 41b0d
Resumen MD5: 600cc1152576c434e6f0fa07162b80b5   Hex: 41c57
Resumen MD5: f00250e2cea1feada62aaf0018eaf12f   Hex: 41f6f
Resumen MD5: 200ca89d8f145c7fe59fb3f378395789   Hex: 42066
Resumen MD5: 80025345ce85cb1df36ebbd0df3eb8b6   Hex: 420f8
Resumen MD5: c0057897e0a19956f04ec09f8bf62268   Hex: 421d7
Resumen MD5: a004148a8a34db386d2f7b31ff4b670b   Hex: 42399
Resumen MD5: e00484ab20cacf896e935b4434670d18   Hex: 4244c
Resumen MD5: 70003af370cef83c2aa299c3f81db230   Hex: 4244e
Resumen MD5: 300f7034b20da9b9976371d69d3fbdec   Hex: 424e3
```

Actividad 1.3

Lenguaje y librería utilizados: Java, ibrerias io, math y security

Fragmento de Código sustancial

```java
for (int i = 0x0000000; i<0xFFFFFFF; i++) {
    String hex = Integer.toHexString(i);
    String lag = s + hex;
    String sha = getSha(lag);
    lag = sha.substring(1);
    //String md5 = getMd5(lag);
    //lag = md5.substring(1);
    if(lag.startsWith("00")) {
        System.out.println("Resumen SHA-256: " + sha + "\tHex: " + hex);
        //System.out.println("Resumen MD5: " + md5 + "\tHex: " + hex);
    }
    /*if(getMd5(lag).startsWith("00")) {
        System.out.println("Resumen MD5: " + getMd5(lag) + "\tHex: " + hex);
    }*/
}
```

Pantallazo ejemplo de ejecución

```
Resumen SHA-256: 300d7860239b16b815cb766c6f54472b5b171a368faa5d81433ed37dc40dac12      Hex: 246c4
Resumen SHA-256: 800576e64bfbd000654e8039440df9c66498666f2bd46700cf590dc864185ea8      Hex: 248bc
Resumen SHA-256: d00a5cc724436607fe26faf7d7435fa2c0fe7f61940bab913064f2c67144cf39      Hex: 24955
Resumen SHA-256: d00d815e24bb2632eec2c3043c88d21c781cf35036e7ef4fef31c649e7b4b2a1      Hex: 24962
Resumen SHA-256: d005bed9ce64d123657bcd41e9a89e9727f1f8ab41ca80c4944f1f979dcdcf2a      Hex: 24a34
Resumen SHA-256: f00466c8e32ba192a71823bb76b82af399005c9e746529c465a515203aaa2812      Hex: 24a97
Resumen SHA-256: 90088ced1793b069c50fac3c8fc9b21f8a9fbbaeb96ae8fe8cea9a3a19c4d635      Hex: 24d6c
Resumen SHA-256: c000bc61510367f28ff777777cdf539bc99c29748976f4d1dd67c411bd48d796      Hex: 24e09
Resumen SHA-256: 2004e3e1d8eef277906049bee52e894a2ab60e2d31feddb7c51564873cd93866      Hex: 24e5b
Resumen SHA-256: a00f8c1d496780086304b62a19e10045f0f036d92c03f2c7e861f480505e826f      Hex: 24e6c
```

Actividad 1.4 (opcional)

Enlace al repositorio o al fichero con el código fuente desarrollado en la actividad 1.1.

Instrucciones para descargar, compilar (si fuera necesario) y ejecutar el código fuente anterior.

Resumen MD5 del fichero que contiene el código fuente.

Licencia que establezca las limitaciones sobre el uso del código en cuestión.

Actividad 1.5 (opcional)

Enlace al repositorio o al fichero con el código fuente desarrollado en la actividad 1.3.

Instrucciones para descargar, compilar (si fuera necesario) y ejecutar el código fuente anterior.

Resumen MD5 del fichero que contiene el código fuente.

Licencia que establezca las limitaciones sobre el uso del código en cuestión.

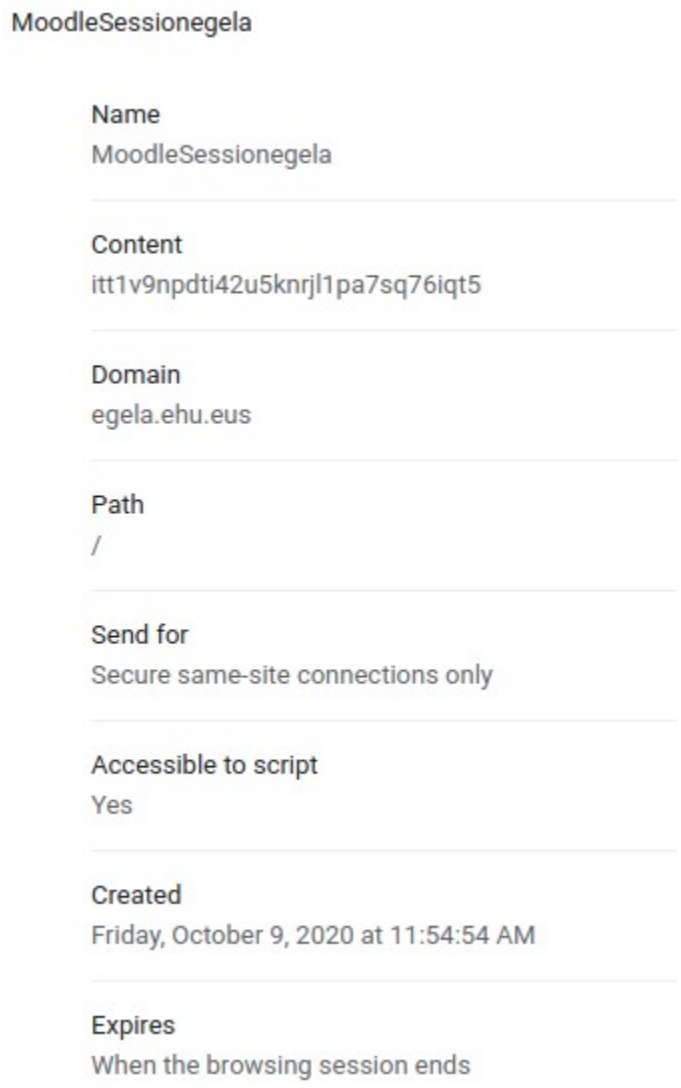
Actividad 2.

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

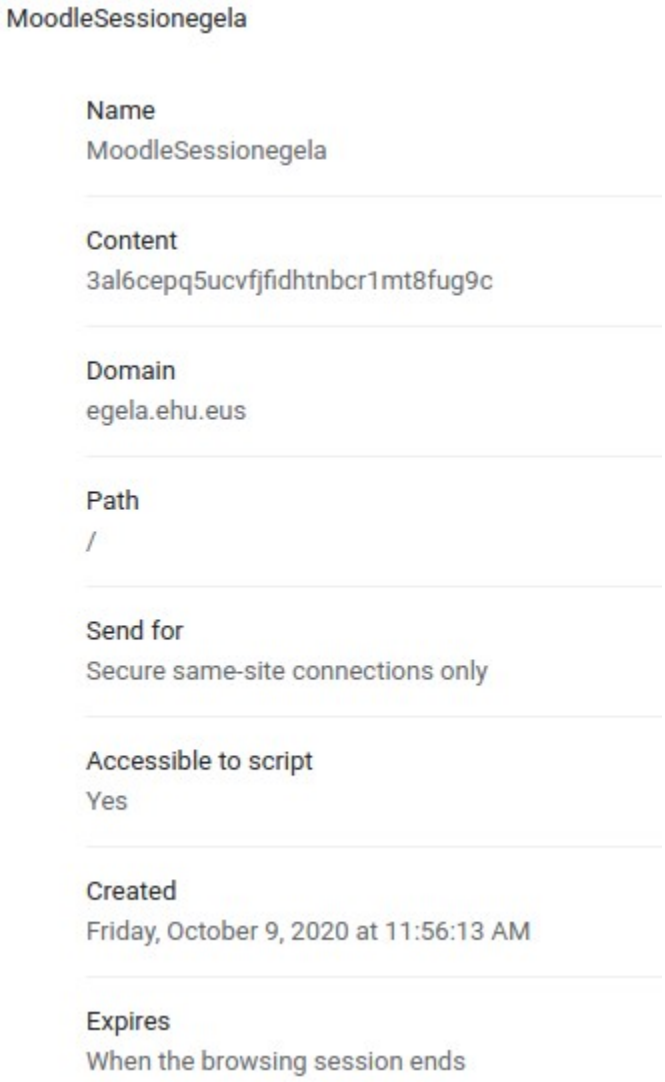
Descripción de lo observado siguiendo los pasos establecidos y conclusiones/reflexiones realizada.

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

Egela (sin identificarse). Valor: *itt1v9npdti42u5knrjl1pa7sq76iqt5*

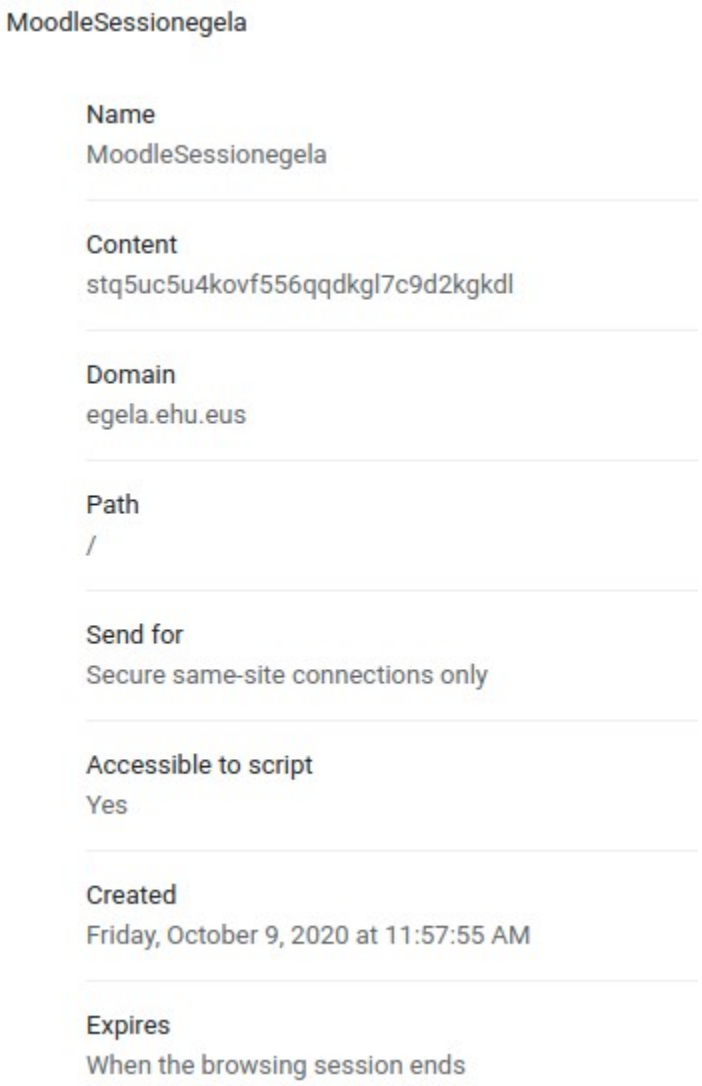


Egela (identificada). Valor: *3al6cepq5ucvfjfidhtnbcr1mt8fug9c*



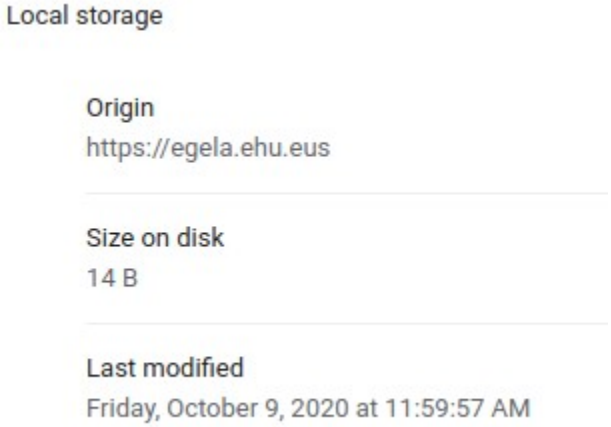
Ha cambiado completamente de valor y no solo eso, si no que la fecha de creación también distinta por lo que ha creado una nueva “instancia”.

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.