REPÚBLICA DEL ECUADOR

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE POSGRADO

MAESTRÍA EN COMPUTACIÓN MENCIÓN SEGURIDAD INFORMÁTICA.

IMPLEMENTACIÓN DE MECANISMOS DE AUTENTICACIÓN 2FA Y QR PARA EL SISTEMA DE INFORMACIÓN DE LA JUNTA DE AGUA POTABLE Y SANEAMIENTO DE LA COMUNIDAD DE PIJAL ALTO – JAAPYS.

Trabajo de Titulación previo a la obtención del Título de Magíster en Computación con Mención en Seguridad Informática.

AUTOR:

Ing. Quilumbaqui Antamba Rodrigo

DIRECTOR:

MSc. Landeta López Pablo Andrés

Ibarra, noviembre 2024

UNIVERSIDAD TÉCNICA DEL NORTE
DIRECCIÓN DE BIBLIOTECA

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1002814430		
APELLIDOS Y NOMBRES:	QUILUMBAQUI ANTAMBA RODRIGO		
DIRECCIÓN:	OTAVALO/GONZALES SUAREZ		
EMAIL:	rquilumbaquia@utn.edu.ec		
TELÉFONO FIJO:	0988698793	**TELÉFONO MÓVIL:**	0968255549

DATOS DE LA OBRA	
TÍTULO:	IMPLEMENTACIÓN DE MECANISMOS DE AUTENTICACIÓN 2FA Y QR PARA EL SISTEMA DE INFORMACIÓN DE LA JUNTA DE AGUA POTABLE Y SANEAMIENTO DE LA COMUNIDAD DE PIJAL ALTO – JAAPYS.
AUTOR (ES):	QUILUMBAQUI ANTAMBA RODRIGO.
FECHA: DD/MM/AAAA	27/11/2024
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☐ **GRADO** ■ **POSGRADO**
TITULO POR EL QUE OPTA:	**MAESTRIA EN COMPUTACIÓN MENCION SEGURIDAD INFORMÁTICA.**
ASESOR /DIRECTOR:	**MSC. ORTIZ GARCES IVAN PATRICIO/ MSC. LANDETA LÓPEZ PABLO ANDRÉS**

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 27 días del mes de noviembre de 2024

EL AUTOR: Firmado electrónicamente por: RODRIGO QUILUMBAQUI ANTAMBA

(Firma)
Nombre: Quilumbaqui Antamba Rodrigo

REPÚBLICA DEL ECUADOR

APROBACIÓN DEL TUTOR.

Yo MSc. Landeta López Pablo Andrés en calidad de director del Trabajo de Grado : "IMPLEMENTACIÓN DE MECANISMOS DE AUTENTICACIÓN 2FA Y QR PARA EL SISTEMA DE INFORMACIÓN DE LA JUNTA DE AGUA POTABLE Y SANEAMIENTO DE LA COMUNIDAD DE PIJAL ALTO – JAAPYS" presentado por el Ing. Quilumbaqui Antamba Rodrigo, para optar por el grado de Magister en Computación con Mención en Seguridad Informática, doy fe de que dicho trabajo reúne los requisitos y méritos suficientes para ser sometidos a presentación y evaluación por parte del jurado examinador que se designe.

En la ciudad de Ibarra, a los 27 días del mes de noviembre del 2024

Lo Certifico

Firmado electrónicamente por:
PABLO ANDRES
LANDETA LOPEZ

MSc. Landeta López Pablo Andrés
TUTOR

Índice de contenidos.

CAPITULO I.....20
1. EL PROBLEMA.....20
1.1. Problema de investigación.....20
1.2. Interrogantes de la investigación.....21
1.3. Objetivos de la investigación.....21
1.3.1. Objetivo general.....21
1.3.2. Objetivos específicos.....22
1.4. Hipótesis de trabajo.....22
1.5. Hipótesis alternativa.....22
1.6. Categorización de variables.....22
1.7. Justificación.....23
CAPITULO II.....25
2. MARCO REFERENCIAL.....25
2.1. Antecedentes.....25
2.2. Marco Teórico.....26
2.2.1. Seguridad mediante mecanismos de autenticación.....26
2.2.1.1. Factores de autenticación.....26
2.2.1.1.1. Factores por conocimiento.....26
2.2.1.1.2. Factor por posesión.....26
2.2.1.1.3. Factor por característica personal.....27
2.2.2. Mecanismos de autenticación digital.....27
2.2.2.1. Identidad digital.....27
2.2.2.2. Inscripción y prueba de identidad.....28
2.2.2.3. Autenticación y gestión del ciclo de vida.....29
2.2.3. En que consiste QR.....29
2.2.3.1. Tipos de QR Code®.....30
2.2.3.1.1. Modelo 1 / Modelo 2.....30
2.2.3.1.2. Micro QR Code®.....30
2.2.3.1.3. SQRC.....31
2.2.3.1.4. rMQR Code.....31

2.2.3.1.5. Frame QR. 32
2.2.4. Autenticación de doble factor. 32
2.2.4.1. Segundos factores en la autenticación. 33
2.2.4.1.1. Verificación de autenticación por SMS. 33
2.2.4.1.2. Verificación por llamada telefónica. 33
2.2.4.1.3. Tarjetas coordenadas. 33
2.2.4.1.4. Correo electrónico. 34
2.2.4.1.5. Token. 34
2.2.4.1.6. Software token. 34
2.2.4.1.7. Hardware token. 35
2.2.4.1.8. Otras formas. 35
2.2.5. 2FA y QR. 36
2.2.6. Seguridades en sistemas de información. 36
2.2.6.1. Identificación, autenticación, autorización en sistemas de información. 36
2.2.6.2. Controles de acceso a los sistemas de información. 37
2.2.6.3. Aspectos sobre la autenticación en los sistemas de información. 38
2.2.6.4. OWASP, Web Security Test Guide Versión 4.2. 40
2.2.6.4.1. ¿Qué es una vulnerabilidad? 40
2.2.6.4.2. ¿Qué es una amenaza? 41
2.2.6.4.3. Que es una prueba. 41
2.2.6.4.4. Metodologías de pruebas OWASP. 41
2.3. Marco Legal. 42
CAPITULO III. 44
3. MARCO METODOLÓGICO. 44
3.1. Descripción del área de estudio/Descripción del grupo de estudio. 44
3.2. Enfoque y tipo de investigación. 44
3.3. Procedimiento de investigación. 45
3.3.1. Revisión de pruebas a la aplicación con OWASP, Web Security Test Guide. 45
3.3.1.1. Recopilación de información. 45

3.3.1.1.1. Fugas de información mediante motores de búsqueda. 45
3.3.1.1.2. Huellas dactilares del servidor web. 46
3.3.1.1.3. Identificar puntos de entradas y superficie de ataque. 46
3.3.1.1.4. Huellas del marco de la aplicación. 47
3.3.1.1.5. Mapa de la arquitectura de la aplicación. 47
3.3.1.2. Pruebas de autenticación. 48
3.3.1.2.1. Credenciales transportadas a través de un canal cifrado. 48
3.3.1.2.2. Credenciales predeterminadas. 48
3.3.1.2.3. Prueba de mecanismos de bloqueos débil. 48
3.3.1.2.4. Pruebas de omisión del esquema de autenticación. 50
3.3.1.2.5. Prueba de vulnerabilidad de recordar contraseñas. 50
3.3.1.2.6. Debilidades en prueba del cache del navegador. 50
3.3.1.2.7. Políticas de contraseñas débiles. 51
3.3.1.2.8. Respuestas débiles a preguntas de seguridad. 51
3.3.1.2.9. Funcionalidades, cambio y restablecimiento de contraseña. 52
3.3.1.2.10. Autenticación débil en un canal alternativo. 52
3.3.1.2.11. Prueba de autenticación multifactorial (MFA). 52
3.3.1.3. Pruebas de validación de entradas en la autenticación. 53
3.3.1.3.1. Pruebas de inyección Sql. 53
CAPITULO IV 54
4. EVALUACIÓN Y RESULTADOS. 54
4.1. Lista de verificación de pruebas (WSTG) en la autenticación. 54
4.2. Resumen de vulnerabilidades en la autenticación de la aplicación web. 58
CAPITULO V 61
5. PROPUESTAS, DESARROLLO DE SOLUCIONES Y PRUEBAS. 61
5.1. Propuesta. 61
5.2. Desarrollo e implementación de las soluciones. 62
5.2.1. Implementación de directivas HSTS(*Strict-Transport-Security*). 62
5.2.2. Implementación de políticas para generación de usuarios. 63
5.2.3. Políticas de generación de contraseña. 65
5.2.4. Hashing de contraseña. 65

5.2.5. Implementar restablecimiento y cambio de contraseñas. 66
5.2.6. Prevención de salto u omisión del esquema de autenticación. 69
5.2.7. Generación token’s de sesión seguros del lado del servidor. 71
5.2.8. Implementar mecanismos de autenticación 2FA. 71
5.2.8.1. 2FA por correo electrónico. 71
5.2.8.2. 2FA por QR. 73
5.2.9. Adaptación de lectura QR para la autenticación de usuarios. 75
5.2.10. Funciones de cifrado y descifrado de datos en QR. 77
5.2.11. Aplicar prevención contra inyección Sql. 78
5.3. Revisión de pruebas a la propuesta mediante OWASP Web Security Testing Guide. 80
5.3.1.1. Recopilación de Información de la propuesta. 80
5.3.1.1.1. Fugas de información mediante motores de búsqueda. 80
5.3.1.1.2. Huellas dactilares del servidor web. 80
5.3.1.1.3. Identificar puntos de entradas de la aplicación. 80
5.3.1.1.4. Huellas del marco de la aplicación. 81
5.3.1.2. Pruebas de Autenticación de la propuesta. 82
5.3.1.2.1. Credenciales transportadas a través de un canal cifrado. 82
5.3.1.2.2. Credenciales predeterminadas. 82
5.3.1.2.3. Prueba de mecanismos de bloqueos débil. 82
5.3.1.2.4. Pruebas de omisión del esquema de autenticación. 82
5.3.1.2.5. Prueba de vulnerabilidad de recordar contraseñas. 83
5.3.1.2.6. Debilidades en prueba del cache del navegador. 83
5.3.1.2.7. Políticas de contraseñas débiles. 83
5.3.1.2.8. Respuestas débiles a preguntas de seguridad. 83
5.3.1.2.9. Funcionalidades, cambio y restablecimiento de contraseña. 84
5.3.1.2.10. Autenticación débil en un canal alternativo. 84
5.3.1.2.11. Prueba de autenticación multifactorial (MFA). 84
5.3.1.3. Pruebas de validación de entradas en la autenticación. 84
5.3.1.3.1. Pruebas de inyección SQL. 84
5.4. Resultados de pruebas y evaluación de la propuesta. 85

5.4.1. Lista de verificación de pruebas (WSTG) a la autenticación. 85
5.4.2. Resumen y evaluación de vulnerabilidades de la propuesta..... 88
CAPITULO 6 91
6. CONCLUSIONES Y RECOMENDACIONES 91
6.1. Conclusiones 91
6.2. Recomendaciones. 92
ANEXOS..... 93
Anexo 1 Pruebas a la aplicación con OWASP, Web Security Testing Guide..... 93
Anexo 2 Parámetros de evaluación y factores de riesgos. 159
Anexo 3 Evaluación de riesgos del sistema de información..... 161
Anexo 4 Código fuente mecanismos de autenticación. 174
Anexos 5 Pruebas a la solución con OWASP, Web Security Testing Guide. 198
Anexo 6 Evaluación de riesgos de la solución del sistema de información. 261
BIBLIOGRAFIA. 274

Índice de tablas.

Tabla 1 WSTG-INFO-06 Superficie de ataque Owasp. 46
Tabla 2 Lista de verificación Owasp. 55
Tabla 3 Resumen de vulnerabilidades. 58
Tabla 4 Caracteres especiales, que permiten inyección Sql. 78
Tabla 5 WSTG-INFO-06 Superficie de ataque Owasp. 81
Tabla 6 Lista de verificación de la solución. 85
Tabla 7 Evaluación de vulnerabilidades, resumen de la propuesta. 88
Tabla 8 WSTG-INFO-06 Puntos de entrada. Autenticación. 99
Tabla 9 WSTG-INFO-06 Puntos de entrada, datos del sistema. 101
Tabla 10 WSTG-INFO-06 Puntos de entrada, usuarios de agua. 102
Tabla 11 WSTG-INFO-06 Puntos de entrada, rayas 104
Tabla 12 WSTG-INFO-06 Puntos de entrada, contabilidad. 105
Tabla 13 WSTG-INFO-06 Puntos de entrada, reportes. 107
Tabla 14 WSTG-INFO-06 Puntos de entrada, pagos. 108
Tabla 15 WSTG-INFO-06 Puntos de entrada, usuarios. 110
Tabla 16 WSTG-INFO-06 Superficie de ataque OWASP. 112
Tabla 17 WSTG-ATHN-02 Prueba de credenciales predeterminadas. 122
Tabla 18 WSTG-ATHN-02 Prueba de contraseñas predeterminadas. 123
Tabla 19 WSTG-ATHN-02 Prueba de caja gris, entrevista a personal de TI. 124
Tabla 20 WSTG-ATHN-07 Preguntas , políticas y contraseñas débiles. 143
Tabla 21 WSTG-ATHN-09 Comprobación previa. 146
Tabla 22 WSTG-INPV-05 Caracteres, ASCII Encondig permiten inyección Sql. 158
Tabla 23 WSTG-INFO-06 Puntos de entrada, autenticación para. 201
Tabla 24 WSTG-INFO-06 Puntos de entrada Verificación 2FA. 203
Tabla 25 WSTG-INFO-06 Puntos de entrada, verificación 2FA por QR. 205
Tabla 26 WSTG-INFO-06 Puntos de entrada, Cerrar sesión 206
Tabla 27 WSTG-INFO-06 Puntos de entrada autenticación por correo 208
Tabla 28 WSTG-INFO-06 Puntos de entrada 2FA por correo. 209
Tabla 29 WSTG-INFO-06 Puntos de entrada, comprobación por código. 211
Tabla 30 WSTG-INFO-06 Puntos de entrada página inicial del S.I. 212

Tabla 31 WSTG-INFO-06 Puntos de entrada, usuarios del sistema. 214
Tabla 32 WSTG-INFO-06 Superficie de ataque Owasp. 216
Tabla 33 WSTG-ATHN-02 Credenciales predeterminadas en aplicaciones. 223
Tabla 34 WSTG-ATHN-02 Prueba de contraseñas predeterminadas. 223
Tabla 35 WSTG-ATHN-02 Entrevistas a personal de T.I. 224
Tabla 36 WSTG-ATHN-07 Preguntas de políticas de contraseñas débiles. 243
Tabla 37 WSTG-ATHN-09 Comprobación previa. 245
Tabla 38 WSTG-ATHN-09 Preguntas de restablecimiento de contraseña. 245
Tabla 39 WSTG-ATHN-11 Preguntas previas, pruebas de gestión MFA 251
Tabla 40 Codificación ASCII para inyección MySql 259

Índice de figuras.

Figura 1 Variable independiente.....23
Figura 2 Variable dependiente.....23
Figura 3 QR Modelo /Modelos 2.....30
Figura 4 Micro QR Code.....30
Figura 5 SQRC.....31
Figura 6 rMOR Code.....31
Figura 7 Frame QR.....32
Figura 8 WSTG-INFO-01 Información personal de usuarios de agua.....45
Figura 9 Arquitectura de la aplicación.....47
Figura 10 Propuesta, aplicando HTTP Strict Transport Security (HSTS).....62
Figura 11 Propuesta, verificación HSTS.....63
Figura 12 Propuesta, política de generación de usuario.....64
Figura 13 Propuesta, fortaleza de contraseña.....65
Figura 14 Propuesta, verificación de requisitos de contraseña.....65
Figura 15 Propuesta Hashing de contraseña en base de datos del sistema.....66
Figura 16 Propuesta, recuperación o restablecimiento de contraseña.....67
Figura 17 Propuesta, recepción de correo electrónico.....67
Figura 18 Propuesta, código temporal para restablecimiento de contraseña.....68
Figura 19 Formulario cambio de contraseña.....68
Figura 20 Propuesta, restablecimiento de credenciales y QR de usuario.....69
Figura 21 Propuesta, ventana principal autenticación del sistema de información..70
Figura 22 Propuesta, inicio de sesión para 2FA.....72
Figura 23 Propuesta, correo electrónico con código temporal.....72
Figura 24 Propuesta, verificación 2FA.....72
Figura 25 Propuesta, sesión iniciada mediante 2FA.....73
Figura 26 Propuesta, inicio de sesión para QR.....74
Figura 27 Propuesta, verificación 2FA por QR.....74
Figura 28 Propuesta, generación QR de autenticación de usuario.....75
Figura 29 Propuesta, autenticación mediante QR.....76
Figura 30 Propuesta, 2FA después de autenticación QR.....77

Figura 31 Propuesta, Prevención de inyección Sql. 78
Figura 32 WSTG-INFO-01 site:comite-de-agua.webcindario.com/agua..... 94
Figura 33 WSTG-INFO-01 inurl:comite-de-agua.webcindario.com/agua..... 94
Figura 34 WSTG-INFO-01 intitle:comite-de-agua.webcindario.com/agua..... 95
Figura 35 WSTG-INFO-01 inbody:comite-de-agua.webcindario.com/agua 95
Figura 36 WSTG-INFO-01 comite-de-agua.webcindario.com/agua filetype:php .. 96
Figura 37 WSTG-INFO-02 Descubriendo el servidor a base de comandos Linux .. 97
Figura 38 WSTG-INFO-02 Descubriendo puertos en servidor en base a Nmap. 97
Figura 39 WSTG-INFO-02 Descubriendo host y puertos con Nmap. 98
Figura 40 WSTG-INFO-02 Descubriendo ip, nombres, sistema operativo. 98
Figura 41 WSTG-INFO-06 Puntos de entrada, ingreso de usuario al S.I. 100
Figura 42 WSTG-INFO-06 Puntos de entrada, datos del sistema de Información 102
Figura 43 WSTG-INFO-06 Puntos de entrada, usuarios de agua 103
Figura 44 WSTG-INFO-06 Puntos de entrada, rayas de usuarios. 105
Figura 45 WSTG-INFO-06 Puntos de entrada, contabilidad. 106
Figura 46 WSTG-INFO-06 Puntos de entrada, reportes. 108
Figura 47 WSTG-INFO-06 Puntos de entrada, pagos..... 109
Figura 48 WSTG-INFO-06 Puntos de entrada, usuarios del sistema..... 111
Figura 49 WSTG-INFO-08 Huellas del marco de aplicación mediante cookies. . 113
Figura 50 WSTG-INFO-08 Huellas del marco de aplicación mediante HTML. .. 113
Figura 51 WSTG-INFO-08 Huellas del marco de aplicación revisando carpetas. 113
Figura 52 WSTG-INFO-06 Huellas del marco de aplicación con Wappalizador.. 114
Figura 53 WSTG-INFO 10 Mapa de la arquitectura de la aplicación..... 115
Figura 54 WSTG-INFO-10 Mapa de la arquitectura de la aplicación. 115
Figura 55 WSTG-ATHN-01 Request, credenciales por canales no cifrados..... 116
Figura 56 WSTG-ATHN-01 Response, credenciales por canales no cifrados..... 117
Figura 57 WSTG-ATHN-01 Atrib. secure, credenciales por canales no cifrados. 117
Figura 58 WSTG-ATHN-01 Request, creación de cuentas, credenciales por canales no cifrados. 118
Figura 59 WSTG-ATHN-01 Response, creación de cuentas, credenciales por canales no cifrados..... 118

Figura 60 WSTG-ATHN-01 Petición, credenciales enviadas por de canales no cifrados. 119
Figura 61 WSTG-ATHN-01 Cambio de contraseña, credenciales por canales no cifrados 120
Figura 62 WSTG-ATHN-01 Cambio de contraseña, respuestas en canales no cifrados. 120
Figura 63 WSTG-ATHN-01 Sesiones inseguras, Credenciales por canales no cifrados 121
Figura 64 WSTG-ATHN-04 Prueba de caja negra solicitud de página directa. 128
Figura 65 WSTG-ATHN-04 Prueba de caja negra, modificación de parámetros.. 129
Figura 66 WSTG-ATHN-04 Predicción de ID de sesión. 129
Figura 67 WSTG-ATHN-04 Inyección Sql, vulnerabilidades con ZAP 130
Figura 68 WSTG-ATHN-04 Inyección Sql, atributo de vulnerabilidad en ZAP .. 130
Figura 69 WSTG-ATHN-04 Inyección Sql, Request autenticación por formulario HTML. 131
Figura 70 WSTG-ATHN-04 Inyección Sql, Response autenticación por formulario HTML. 131
Figura 71 WSTG-ATHN-04 Inyección Sql, Resultado de autenticación. 131
Figura 72 WSTG-ATHN-04 Prueba de caja gris, revisión de sentencia Sql. 132
Figura 73 WSTG-ATHN-04 Prueba de caja gris revisión de código fuente 132
Figura 74 WSTG-ATHN-05 Almacenar credenciales. 134
Figura 75 WSTG-ATHN-05 Almacenar credenciales, Local Storage 134
Figura 76 WSTG-ATHN-05 Almacenar credenciales, sesión storage. 134
Figura 77 WSTG-ATHN-05 Almacenar credenciales, cookies 135
Figura 78 WSTG-ATHN-05 Almacenar credenciales. cookies del lado cliente.... 135
Figura 79 WSTG-ATHN-05 Inyección de credenciales, Clickjacking. 137
Figura 80 WSTG-ATHN-06 Creación de usuario, historial de navegador. 139
Figura 81 WSTG-ATHN-06 Creación de usuario, historial de navegador, usuario creado..... 139
Figura 82 WSTG-ATHN-06 Creación de usuario, historial de navegador, <atrás. 139
Figura 83 WSTG-ATHN-06 Historial de navegador, cache. 140

Figura 84 WSTG-ATHN-06 Cache del navegador, no cache, no store. 141
Figura 85 WSTG-ATHN-06 Información almacenada en cache. 142
Figura 86 WSTG-ATHN-06 Información almacenada en cache móviles. 142
Figura 87 WSTG-ATHN-10 Nuevo usuario, comprensión mecanismo primario.. 148
Figura 88 WSTG-ATHN-10 Comprensión del mecanismo primario, usuario creado 149
Figura 89 WSTG-ATHN-10 Comprensión del mecanismo primario, modificar cuentas de usuario..... 149
Figura 90 WSTG-ATHN-11 Verificación derivaciones MFA, no existe método. 152
Figura 91 WSTG-INPV-05 Inyección Sql dentro de una petición POST. 153
Figura 92 WSTG-INPV-05 Consulta Sql por inyección GET-..... 154
Figura 93 WSTG-INPV-05 Resultado de inyección por el método GET. 154
Figura 94 WSTG-INPV-05 Inyección Sql, POST, descubriendo el servidor. 155
Figura 95 WSTG-INPV-05 Inyección Sql, consulta, descubriendo servidor..... 155
Figura 96 WSTG-INPV-05 Inyección Sql, POST, descubriendo usuarios de la BD. 156
Figura 97 WSTG-INPV-05 Inyección Sql, descubriendo usuarios, ejecución de consulta Sql..... 156
Figura 98 WSTG-INPV-05 Inyección Sql, nombre de la BD. 157
Figura 99 WSTG-INPV-05 Inyección Sql, nombre de BD, ejecución. 157
Figura 100 WSTG-INPV-05 Vulnerabilidades Mysql mediante ZAP..... 157
Figura 101 WSTG-INPV-05 Evidencia de vulnerabilidad Mysql. 158
Figura 102 Parámetros de evaluación de riesgo 1. 159
Figura 103 Parámetro de evaluación de riesgo 2. 159
Figura 104 Calculo de gravedad general de riesgo..... 160
Figura 105 Calculo de riesgo, credenciales enviadas por canales no cifrados. 161
Figura 106 Calculo de riegos Tokens y sesión. 162
Figura 107 Calculo de riesgos, credenciales por defecto 163
Figura 108 Calculo de riesgo, mecanismos de bloqueo y desbloqueo 164
Figura 109 Calculo de riesgo, acceso a información confidencial. 165
Figura 110 Cálculo de riesgo, inyección de credenciales..... 166

Figura 111 Cálculo de riesgo, sesiones sin tiempo de expiración. 167
Figura 112 Calculo de riesgo, políticas de fortalecimiento de credenciales 168
Figura 113 Cálculo de riesgo, preguntas de seguridad en credenciales 169
Figura 114 Cálculo de riesgo, funcionalidades de reinicio de credenciales. 170
Figura 115 Calculo de riesgo, canal alternativo de autenticación. 171
Figura 116 Calculo de riesgo, autentificación MFA 172
Figura 117 Calculo de riesgo, inyección Sql. 173
Figura 118 WSTG-INFO-01 site: juntadeaguapijal.webcindario.com/agua/ 199
Figura 119 WSTG-INFO-01 inurl: juntadeaguapijal.webcindario.com/agua/ 199
Figura 120 Descubriendo puertos en el servidor con Nmap. 200
Figura 121 WSTG-INFO-02 Descubriendo host y puertos con Nmap. 200
Figura 122 WSTG-INFO-02 Descubriendo S.O con Nmap..... 201
Figura 123 WSTG-INFO-06 Puntos de entrada, Response autenticación. 203
Figura 124 WSTG-INFO-06 Puntos de entrada, respuesta Verificación 2FA QR. 204
Figura 125 WSTG-INFO-06 Puntos de entrada respuesta verificación por QR. ... 206
Figura 126 WSTG-INFO-06 Puntos de entrada Respuesta Login. 209
Figura 127 WSTG-INFO-06 Punt. de entrada, respuesta verificación por correo. 210
Figura 128 WSTG-INFO-06 Punt. de entrada, respuesta verificación por código. 212
Figura 129 WSTG-INFO-06 Puntos de entrada, respuesta página inicial. 213
Figura 130 WSTG-INFO-06 Puntos de entrada, respuesta usuarios de sistema. ... 215
Figura 131 WSTG-INFO-08 Marco de aplicación con Wappalyzer 1..... 217
Figura 132 WSTG-INFO-08 Marco de aplicación con Wappalyzer 2..... 217
Figura 133 WSTG-ATHN-01 Prueba de acceso request, credenciales en canales cifrados. 218
Figura 134 WSTG-ATHN-01 Prueba de acceso response, credenciales por canales cifrados. 219
Figura 135 WSTG-ATHN-01 Creación de cuenta por HSTS. 220
Figura 136 WSTG-ATHN-01 Respuesta creación de cuenta por (HSTS). 220
Figura 137 WSTG-ATHN-01 Req. restablecimiento de contraseña por (HSTS) .. 221
Figura 138 WSTG-ATHN-01 Resp. restablecimiento de contraseña por (HSTS). 221
Figura 139 WSTG-ATHN-01 Acceder a recursos mientras está conectado. 222

Figura 140 WSTG-ATHN-01 Respuesta acceder a recursos estando conectado... 222
Figura 141WSTG-ATHN-03 Mecanismos de bloqueo..... 225
Figura 142 Solicitud página directa..... 226
Figura 143 WSTG-ATHN-04 Modificación de parámetros..... 227
Figura 144 WSTG-ATHN-04 Predicción de ID de sesión..... 228
Figura 145 WSTG-ATHN-04 Inyección Sql, parámetros de formulario..... 228
Figura 146 WSTG-ATHN-04 Inyección Sql, aplicación de prueba. 229
Figura 147 WSTG-ATHN-04 Respuesta Inyección Sql. 229
Figura 148 WSTG-ATHN-04 Pagina respuesta Inyección Sql..... 230
Figura 149 WSTG-ATHN-05 Almacenar credenciales sesión iniciada..... 233
Figura 150 WSTG-ATHN-05 Almacenar credenciales Local storage..... 233
Figura 151 WSTG-ATHN-05 Almacenar credenciales Session storage..... 233
Figura 152 WSTG-ATHN-05 Almacenar credenciales Key..... 234
Figura 153 WSTG-ATHN-05 Almacenar credenciales Cookies. 234
Figura 154 WSTG-ATHN-05 Inyección de credenciales Clickjacking..... 237
Figura 155 WSTG-ATHN-06 Prueba de historial del navegador. 238
Figura 156 WSTG-ATHN-06 Resultado de prueba historial de navegador. 239
Figura 157 WSTG-ATHN-06 Prueba historial de Cache de navegador. 239
Figura 158 WSTG-ATHN-06 Cache del navegador, no cache, no store, ... 240
Figura 159 WSTG-ATHN-06 Revisar información almacenada en cache. 241
Figura 160 WSTG-ATHN-06 Verificación de cache de navegadores móviles. 241
Figura 161 WSTG-ATHN-06 Cache previa autenticación de usuario..... 242
Figura 162 WSTG-ATHN-06 Revisión de cache no disponible..... 242
Figura 163 WSTG-ATHN-10 Creación de cuenta de usuario 246
Figura 164 WSTG-ATHN-10 Modificación de usuario..... 247
Figura 165 WSTG-ATHN-11 Verificación MFA autenticación..... 249
Figura 166 WSTG-ATHN-11 Verificación 2FA por correo electrónico. 250
Figura 167 WSTG-ATHN-11 Verificación 2FA con QR. 250
Figura 168 WSTG-ATHN-11 Recuperación de contraseña..... 251
Figura 169 WSTG-ATHN-11 Correo electrónico de recuperación. 251
Figura 170 WSTG-INPV-05 Inyección Sql para una petición POST. 252

Figura 171 WSTG-INPV-05 Inyección Sql en petición GET. 253
Figura 172 WSTG-INPV-05 Inyección Sql con verificación Sql. 254
Figura 173 WSTG-INPV-05 Inyección Sql descubrimiento de servidor. 254
Figura 174 WSTG-INPV-05 Inyección Sql, resultado, descubrimiendo servidor. 255
Figura 175 WSTG-INPV-05 Inyección Sql en formulario. 256
Figura 176 WSTG-INPV-05 Inyección SQL, descubriendo de usuarios BD. 256
Figura 177 WSTG-INPV-05 Inyección Sql en formulario. 257
Figura 178 WSTG-INPV-05 Inyección Sql, descubriendo base de datos. 257
Figura 179 WSTG-INPV-05 Verificación de vulnerabilidades Sql con ZAP. 258
Figura 180 WSTG-INPV-05 Resultado de inyección SQL con ASCCII %27. 259
Figura 181 WSTG-INPV-05 Inyección, enlistar todos los datos de una tabla. 259
Figura 182 Evaluación de riesgos, rastreo de credenciales(solución) 261
Figura 183 Evaluación de riesgos, aseguramiento de token y sesiones(solución). 262
Figura 184 Evaluación de riesgos, credenciales por defecto(solución). 263
Figura 185 Evaluación de riesgo, mecanismos de bloqueo y desbloqueo. 264
Figura 186 Evaluación de riesgo, acceso a información confidencial(solución). .. 265
Figura 187 Evaluación de riesgo, inyección automática de credenciales(s) 266
Figura 188 Evaluación de riesgos, expiración de sesiones(solución). 267
Figura 189 Evaluación de riesgos, de fortalecimiento de password(solución). 268
Figura 190 Evaluación de riesgos, preguntas de seguridad en credenciales. 269
Figura 191 Evaluación de riesgo, funcionalidades de reinicio(solución). 270
Figura 192 Evaluación de riesgo, canal alternativo de autenticación. 271
Figura 193 Evaluación de riesgo, autenticación MFA (solución). 272
Figura 194 Evaluación de riesgo, inyección SQL (solución). 273

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE POSGRADO

MAESTRÍA EN COMPUTACION MENCIÓN SEGURIDAD INFORMATICA

Autor: Quilumbaquí Antamba Rodrigo.

Tutor: Landeta López Pablo Andrés.

Año: 2024**.**

Resumen.

Este trabajo se desarrolló con el propósito de implementar mecanismos de autenticación 2FA y QR en un Sistema de Información existente de la Junta de Agua Potable y alcantarillado Pijál Alto, que por su época de desarrollo solo se había contemplado un mecanismo de autenticación basado en usuario y contraseña sin las seguridades necesarias para contrarrestar ataques y acceso no autorizado al sistema de información.

La guía metodológica de OWASP WSTG y sus pruebas WSTG-INFO-01 hasta WSTG-INFO-08, WSTG-ATHN-01 hasta WSTG-ATHN-11 y WSTG-INPV-05 nos ha permitido identificar vulnerabilidades en la autenticación y diseño de la aplicación web desarrollado en lenguaje PHP, entre las vulnerabilidades que podemos mencionar es: la aplicación transportaba sus credenciales sobre canales no cifrados http, debilidades en las credenciales por defecto, falta de mecanismos de bloqueo y desbloqueo, acceso directo a páginas confidenciales, inyección SQL, tokens y credenciales sin tiempo de expiración y sin seguridad, ausencia de políticas de generación de credenciales, credenciales sin cifrado, falta de un mecanismo de autenticación 2FA, etc. Descubierto estas deficiencias nos hemos visto en la necesidad de remedir estas vulnerabilidades y robustecer con un doble factor de autenticación que en nuestro caso es 2FA ya sea por correo electrónico o QR con Google Autenticator. Continuamos con el desarrollo de las soluciones con un nuevo Front-end que utiliza la misma base de datos que fue necesario usarla para la continuidad del negocio. Bibliotecas de desarrollo Bootstrap framework, Implementation QR 2-D, instascan, php Mailer, Php Google Autenticator 2-factor, JQuery-datatables, jquery, Popper, sweetaler2, fueron muy importantes para plasmar la solución.

Finalmente, realizamos una evaluación de los mismos puntos utilizando WSTG de OWASP pudiendo determinar mejor fortaleza en el mecanismo de autenticación del sistema.

Palabras clave: 2FA y QR, evaluación de autenticación, evaluación OWASP, evaluar aplicación web con OWASP.

Abstract.

This work was developed to implement 2FA and QR authentication mechanisms in an existing Information System of the “Pijál Alto” Drinking Water and Sewage Board, due to its time of development, it had only contemplated an authentication mechanism based on username and password. without the necessary security to counteract attacks and unauthorized access to the information system.

The OWASP WSTG methodological guide and its tests WSTG-INFO-01 to WSTG-INFO-08, WSTG-ATHN-01 to WSTG-ATHN-11 and WSTG-INPV-05 have allowed us to identify vulnerabilities in the authentication and design of the web application developed in PHP language, among the vulnerabilities that we can mention are: the application transported its credentials over unencrypted http channels, weaknesses in the default credentials, lack of locking and unlocking mechanisms, direct access to confidential pages, SQL injection, tokens and credentials with no expiration time and without security, absence of credential generation policies, unencrypted credentials, lack of a 2FA authentication mechanism, etc. Once these deficiencies were discovered, we found it necessary to remediate these vulnerabilities and strengthen them with a double authentication factor, 2FA either by email or QR with Google Authenticator. We continued with the development of the solutions with a new Front-end that uses the same database that was necessary for business continuity. Development libraries Bootstrap framework, QR 2-D Implementation, instascan, php Mailer, Php Google Authenticator 2-factor, JQuery-datatables, jquery, Popper, sweetaler2, were essential to shape the solution.

Finally, we evaluated the same points using OWASP WSTG, being able to determine better strength in the system's authentication mechanism.

Keywords: 2FA and QR, evaluation and authentication, OWASP evaluation, evaluate web application.

CAPITULO I

1. EL PROBLEMA.

1.1. Problema de investigación.

Como es de nuestro conocimiento en el proceso de administración de las organizaciones, la información se ha constituido uno de los activos muy importantes para todas las organizaciones, mismas que se debe asegurarlas cumpliendo los requisitos de C.I.A (Confidencialidad, Integridad, Disponibilidad) trilogías aplicadas en la seguridad de la información.

Aunque se procura tener un mayor grado en la seguridad de las aplicaciones, existe riesgos de que los métodos de autenticación sean vulnerados por lo que gran cantidad de datos del sistema de información puedan ser alterados, reemplazados, adulterados a partir de un acceso no autorizado como podemos relacionar lo que se dice:" el robo y fraude de identidad está estrechamente relacionado con la autenticación" (Xia et al., 2010, párr. Introduction).

Debemos aclarar que el acceso a los sistemas de información está basado principalmente en el uso de usuario y contraseña como credenciales de acceso, mismos que con el transcurso del tiempo se ha convertido más vulnerables a los ataques de acceso no autorizado, robo de identidad u otros como inyección SQL, dando como resultado los problemas mencionados anteriormente si el sistema es vulnerado.

Para fortalecer los métodos de autenticación de un sistema de información es necesario implementar medidas que permitan robustecer estos mecanismos, los que podríamos describir son: medidas de autenticación 2FA, medidas de autenticación basado en QR que incrementará el nivel de confianza para el acceso de un usuario a un sistema de información.

La Junta Administradora de Agua Potable y Alcantarillado; es una organización comunitaria con 470 usuarios jefes de familia, aproximadamente 2000 habitantes que conforman la población, requiere del servicio de alcantarillado y provisión de agua potable

para su subsistencia. Esta organización toma decisiones administrativas de manera colectiva mediante asambleas y consejo directivo, para actividades operacionales se ha dividido en unidades administrativas.

Para el caso de nuestro estudio la unidad financiera está encargada de la recaudación, contratación, compras y control del almacén con un sistema de información que es vital para controlar la recaudación, consumo de agua, registros contables que de ser vulnerados prácticamente paralizaría el trabajo financiero de la organización.

El sistema de información se ha desarrollado hace una década atrás y no tiene mecanismos robustos de autenticación, por lo que se expone a amenazas como son la suplantación de identidad, robo de información, entre otras.

1.2. Interrogantes de la investigación.

¿Cuáles son los principales inconvenientes en la autenticación; que los usuarios enfrentan con el mecanismo de autenticación basado en usuario y contraseña que esta implementado en el sistema de información actual?

¿Se puede robustecer los mecanismos de autenticación de un sistema de información mediante la aplicación de mecanismos 2FA y QR propuestos?

¿Estos nuevos sistemas de autenticación son suficientes para protegerse ante el intento de acceso no autorizado en este sistema de información?

¿Cuál es el nivel de mejora con la implementación de los nuevos mecanismos de autenticación?

1.3. Objetivos de la investigación.

1.3.1. Objetivo general.

Implementar mecanismos de Autenticación 2FA y QR que permitan mejorar la robustes de los controles de acceso al sistema de información de la Junta de agua Potable y Saneamiento de la comunidad de Pijál Alto – JAAPYS.

1.3.2. Objetivos específicos.

- Identificar las vulnerabilidades de autenticación actual del sistema de información de la JAAPYS.
- Implementar medidas de autenticación de 2FA y QR considerando las mejores alternativas aplicables al sistema de información del caso de estudio.
- Evaluar el impacto en la seguridad de acceso dentro de la organización al aplicar la autenticación de 2FA y QR.

1.4. Hipótesis de trabajo.

La implementación de mecanismos de autenticación 2FA y QR mejorarán la robustez de los controles de acceso al sistema de información de la Junta de agua Potable y Saneamiento de la comunidad de Pijál Alto – JAAPYS.

1.5. Hipótesis alternativa.

La implementación de mecanismos de autenticación 2FA y QR empeora la robustez de los controles de acceso al sistema de información de la Junta de agua Potable y Saneamiento de la comunidad de Pijál Alto – JAAPYS.

1.6. Categorización de variables.

- **Variable independiente.** Mecanismos de autenticación 2FA y QR**.**
- **Variable dependiente.** Los controles de acceso al sistema de información de la JAAPYS.

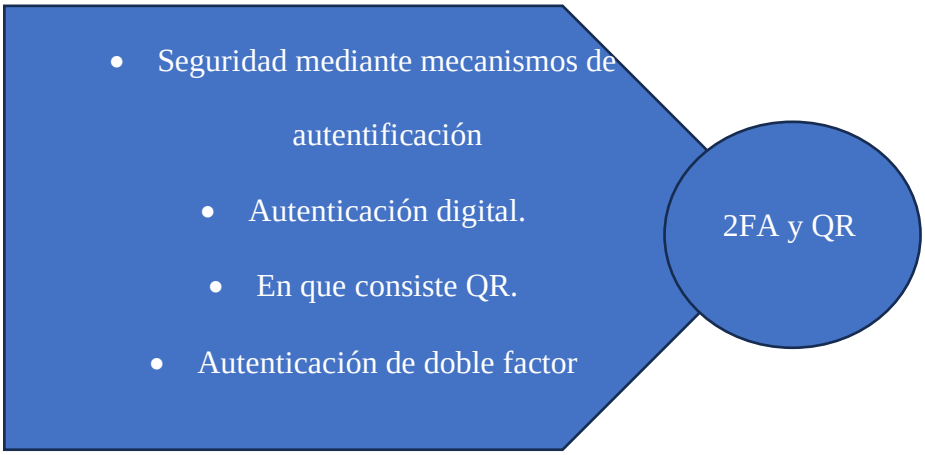


Figura 1 Variable independiente.

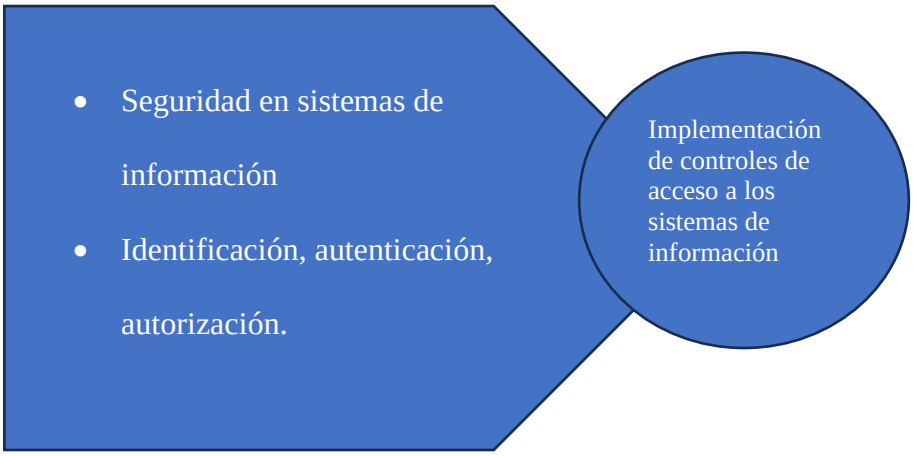


Figura 2 Variable dependiente.

1.7. Justificación.

El Sistema de Información acompaña al desempeño de las actividades operacionales de la Junta de Saneamiento y Agua potable, es muy importante para la administración y gestión de nuestra organización comunitaria.

Para aclarar estas interrogantes debemos revisar ciertas explicaciones que en el siguiente párrafo describe la importancia de una autenticación segura:

En ciberseguridad, son dos puntos en la **autenticación segura,** el control de acceso a las aplicaciones críticas, áreas restringidas y la gestión de claves de forma segura. Asegurar estos es fundamental para velar adecuadamente por la ciberseguridad de cualquier tipo de negocio, pues vulnerar estos incitaría a sucesos graves de seguridad(INCIBE, 2023).

Tomando en cuenta los factores mencionados, así como la necesidad de almacenar determinado tipo información en base a QR para crear un mecanismo seguro de autenticación, justificaría la aplicación de estos métodos al sistema de información de la junta Administradora de agua potable y alcantarillado para robustecer la autenticación del usuario y garantizar la confidencialidad, disponibilidad, e integridad de su información.

Por estas razones se indica la relevancia e importancia de la investigación. Esto implica considerar los aportes al área de conocimiento, al desarrollo regional o de una localidad como es la comunidad beneficiaria de la Junta administradora de Agua Potable y Saneamiento Pijál, el mejoramiento de la calidad de vida de una comunidad o grupo humano y perfeccionamiento de una técnica o proceso ya existente, entre otros.

Según el Plan Nacional de desarrollo y creación de oportunidades 2021-2025 con respecto a Gobernanza(G8), incentiva a: “Generar redes de conocimiento vinculados a la Educación Superior, que promuevan espacios territoriales de innovación adaptados a las necesidades de la sociedad y el sector productivo local “ (Plan de Creación de Oportunidades-2021-2025, 2021, p. 41).

Aclarando que la línea de acción que se mantendrá será la ciberseguridad a fin de proteger el sistema de información ante ataques maliciosos.

CAPITULO II.

2. MARCO REFERENCIAL.

2.1. Antecedentes.

En los tiempos la complejidad en la autenticación de los usuarios en los sistemas de información ha incrementado, ya sea por la forma de almacenar el usuario y contraseña, el nivel de acceso, privilegios o la forma de asegurarse en la comprobación de autenticidad de usuario, ahora podemos decir que muchas aplicaciones ya ofrecen en sus sistemas una autenticación de 2FA, que por ejemplo a parte de pedir un usuario y contraseña piden algún dato adicional para verificar la autenticidad de quien se dice ser. Por ejemplo, **algo que tienes** y a donde envían un código adicional para concretar la autenticación; un OTP enviado por correo, por el teléfono o SMS, etc.

Otras demostraciones de este tipo de autenticaciones los podemos mencionar el correo "Yahoo", que si no está seguro que es el dispositivo, pues pedirá el número telefónico que se registró anteriormente para enviar un código temporal. Así también entre otras aplicaciones podemos mencionar Google Authenticator, Microsoft Authenticator, etc.

Podemos indicar que existen diferentes trabajos relacionados y orientados a mitigar necesidades institucionales por ejemplo "Sistema de autenticación basado en códigos QR para acceso a servicios proporcionados por la UTN"(Rodríguez Vásquez, 2023).

Otro de trabajo similar que se puede mencionar es "Seguridad informática aplicando la autentificación por doble factor para la plataforma HomeOfi" (Espinoza Tinoco et al., 2022).

Lo expuesto anteriormente y visto la necesidad se puede decir que el trabajo actual lo realizamos para cubrir la demanda en el sistema de información de la Junta de Agua Potable y alcantarillado de la comunidad de Pijál alto "JAAPI".

2.2. Marco Teórico

2.2.1. Seguridad mediante mecanismos de autenticación.

2.2.1.1. Factores de autenticación.

Los factores de autenticación son procedimientos utilizados para verificar que el firmante realmente es el dueño de la firma o de una credencial, también para endurecer la seguridad a cualquier tipo de información al validar nuevamente la identificación de las personas autorizadas según se dice que "los usuarios se identifican por conocimiento (por ejemplo, un secreto como una contraseña o PIN), por posesión (por ejemplo, un token físico), por una característica personal (datos biométricos como una huella digital o geometría de la cara)" (De Soete, 2011, párr. Theory) .

2.2.1.1.1. Factores por conocimiento.

Según Mantovani (2019) es demostrar el conocimiento de un secreto que conoce puede ser una contraseña o un código, fecha de nacimiento, el nombre de una mascota o cosas que conozca, por lo general está cerca del usuario, muchas veces esperamos que el usuario memorice y evite que sean compartidas con otras personas. Por otra parte, se dice que es el tipo de "autenticación más frecuente se fundamenta en algo que los usuarios saben, normalmente una contraseña, este factor es que tiene requisitos bajos tanto para los desarrolladores como para el usuario final, no pide ningún hardware o combinación con otros servicios"(OWASP, 2024a, párr. Something You Know).

2.2.1.1.2. Factor por posesión.

Según Mantovani (2019) es algo que se posee o que parece a una llave de la casa que tiene un propietario, claro si es adquirida por otra persona puede ser vulnerado, entonces debe ser bien cuidado para su uso, entre las que podemos tener son: token, tarjeta de coordenadas, certificado digital, tarjeta de crédito, mientras que para OWASP (2024c) se cimenta en un elemento físico o digital para autenticarse que se utiliza junto con la contraseña.

2.2.1.1.3. Factor por característica personal.

Características únicas que son asociadas a un apersona y que se utiliza métodos que por el momento son de alta tecnología como los biométricos, lectores de huellas, de retina o reconocimiento de voz ya que "Se basa en atributos físicos del usuario, requiere que el usuario tenga un hardware específico, puede considerarse más invasiva en términos de privacidad utilizado comúnmente para la autenticación del sistema operativo y también se utiliza en algunas aplicaciones móviles"(OWASP, 2024a, párr. Something You Are).

2.2.2. Mecanismos de autenticación digital.

Entendemos a los mecanismos de autenticación digital las formas como se permite acceder a un servicio electrónico o aplicación, dentro de la aplicación debe haber alguna forma que permita conocer digitalmente una identidad en una red o un sistema de información, lo que se intenta mediante la red o sistemas de comunicación es identificar a un usuario o persona igual que en la vida real, pero de forma virtual.

La autenticación electrónica, es el proceso de confiar en la identidad del usuario electrónicamente a un sistema de información. Los sistemas pueden verificar la identidad y determinar si esa persona está autorizada a realizar una transacción electrónica, actividad o proceso llevando a cabo a través del Internet (Burr et al., 2013).

El Instituto Nacional Estadounidense de Estándares y Tecnología (NIST) ha creado un modelo genérico para la autenticación digital que describe los procesos que se utilizan para lograr una autenticación segura.

2.2.2.1. Identidad digital.

¿Una identidad digital podría ser un usuario de correo electrónico, o un nombre de usuario en un servicio web, nik, alias, etc, podría tener características, privilegios? Para despejar estas dudas pasamos a revisar el siguiente párrafo a continuación:

La identidad digital es la representación única de un sujeto que participa en una transacción en línea. Una identidad digital siempre es única en el contexto de un

servicio digital, pero no necesariamente necesita identificar de manera única al sujeto en todos los contextos. En otras palabras, acceder a un servicio digital puede no significar que se conozca la identidad real del sujeto. La comprobación de identidad establece que un sujeto es quien dice ser. La autenticación digital es el proceso de determinar la validez de uno o más autenticadores utilizados para reclamar una identidad digital. La autenticación establece que un sujeto que intenta acceder a un servicio digital tiene el control de las tecnologías utilizadas para autenticarse. Una autenticación exitosa proporciona garantías razonables basadas en el riesgo de que el sujeto que accede al servicio hoy es el mismo que accedió al servicio anteriormente. La identidad digital presenta un desafío técnico porque este proceso a menudo implica comprobar a las personas a través de una red abierta y, por lo general, implica la autenticación de sujetos individuales a través de una red abierta para acceder a los servicios gubernamentales digitales. Existen múltiples oportunidades para la suplantación de identidad y otros ataques que reclaman fraudulentamente la identidad digital de otro sujeto (Grassi, Fenton, et al., 2017, p. 2).

2.2.2.2. Inscripción y prueba de identidad.

Hay que diferenciar entre el solicitante de la identidad o suscriptor y el proveedor de credenciales (CSP), mientras que el suscriptor está en la responsabilidad de demostrar quien dice ser el proveedor debe tener los mecanismos necesarios para tener el control de los suscriptores según se aclara en el siguiente párrafo:

Una persona, denominada solicitante en esta etapa, opta por que un proveedor de credenciales (CSP) compruebe su identidad. Si el solicitante tiene éxito en la prueba, la persona se considera suscriptor. El proveedor de credenciales establece un mecanismo para identificar a cada suscriptor, registrar y rastrear los autenticadores

emitidos para ese suscriptor. El suscriptor puede recibir autenticadores en el momento de la inscripción, el proveedor de credenciales puede vincular los autenticadores que el suscriptor ya tiene o pueden generarse más adelante según sea necesario. Los suscriptores tienen el deber de mantener el control de sus autenticadores y cumplir con las políticas de CSP para mantener autenticadores activos. El CSP mantiene registros de inscripción para cada suscriptor para permitir la recuperación de los autenticadores, por ejemplo, en caso de pérdida o robo (Grassi, García, et al., 2017, p. 12).

2.2.2.3. Autenticación y gestión del ciclo de vida.

Puede ocurrir varios eventos durante el ciclo de vida del autenticador (CSP) y del suscriptor que afecta su uso, estos son: vinculación, pérdida, robo, duplicación no autorizada, caducidad y evocación.

El ciclo de vida según Grassi, Fenton, et al. (2017) es:

- Enlace del autenticador.
- Vinculación y matriculación.
- Vinculación posterior a la inscripción.
- Pérdida, robo, daño y duplicación no autorizada.
- Vencimiento.
- Revocación y terminación.

2.2.3. En que consiste QR.

Es una especie de imagen en 2D que nos permite almacenar información, necesariamente se tendrá que utilizar tecnología que permita leer esta imagen para decodificar su significado a altas velocidades y rápida un tamaño físico de una décima parte

más pequeño representa datos en forma horizontal y vertical, almacena números, letra, símbolos binarios y códigos de control (Denso Wave, 2023).

2.2.3.1. Tipos de QR Code®.

2.2.3.1.1. Modelo 1 / Modelo 2.

Figura 3 QR Modelo /Modelos 2.

Nota: Fuente What is a QR Code® (Denso Wave, 2023)

Para, Denso Wave (2023) existe dos modelos similares, pero claramente su capacidad es la diferencia; modelo uno 14(73x73 modules) hasta 1.167 dígitos y modelo dos 40 (177x177 módulos) hasta 7.089 dígitos.

2.2.3.1.2. Micro QR Code®.

Figura 4 Micro QR Code.

Nota: Fuente What is a QR Code® (Denso Wave, 2023)

La más larga versión de Micro QR Code es M4 (17x17 módulos), el cual es capaz de almacenar hasta 35 dígitos con un solo patrón de detección, puede ser impreso en una

pequeña área y el margen mínimo es con 2 módulos (el margen mínimo para normal QR código es 4 módulos) (Denso Wave, 2023, Figura Micro QR).

2.2.3.1.3. SQRC.

Figura 5 SQRC.

Nota: Fuente What is a QR Code® (Denso Wave, 2023)

"Tiene una función restringida de lectura de datos. SQRCs puede ser utilizado para administrar información personal, información interna y otra información confidencial. La apariencia de SQRCs es el mismo que el normal QR **Code®"** (Denso Wave, 2023, Figura SQRC)**.**

2.2.3.1.4. rMQR Code.

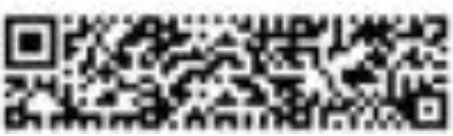


Figura 6 rMOR Code.

Nota: Fuente What is a QR Code® (Denso Wave, 2023)

Imagen horizontal: "Este es un QR Code rectangular fácil desplegar en espacios estrecho, grande y pequeño, los patrones de detección son puestos en cada localización para lograr fácil lectura y gran cantidad de información" (Denso Wave, 2023, Figura rMOR Code).

2.2.3.1.5. Frame QR.

Figura 7 Frame QR.

Nota: Fuente What is a QR Code® (Denso Wave, 2023)

Denso Wave (2023) manifiesta que es un marco de QR que tiene un campo de lienzo que puede ser usado libremente en el código. Letras e imágenes puede ser insertado en el campo del lienzo. El marco QR code puede ser usado para varios propósitos.

2.2.4. Autenticación de doble factor.

Según Reyes Riveros et al. (2023) es una capa adicional de seguridad dificultando a los pirata informáticos el acceso ya que si conocieren las credenciales de su víctima la autenticación se restringirá y evitara el acceso no autorizado así también "este término en la autenticación de usuario indica que dos factores independientes están envueltos en el proceso de establecer un nivel comprendido de confianza de que una entidad especifica o identidad reclamada es genuina" (De Soete, 2011, párr. Definition) finalmente, "la autenticación multifactorial (MFA) o la autenticación de dos factores (2FA) se produce cuando un usuario debe presentar más de un tipo de evidencia para autenticarse en un sistema"(OWASP, 2024a, párr. Introduction).

Entonces podemos entender que la autenticación (2FA) es un método de seguridad dos formas de identificación para obtener acceso a los recursos o datos, los pasos para cumplir este tipo de autenticación son:

- El probador ingresa usuario y contraseña.

- Se envía un código OTP (On Time Password) o de un solo uso mediante un email, SMS o llamada telefónica generalmente a un dispositivo móvil o correo.
- El probador ingresa el código recibido para completar el acceso a la cuenta.
- Este código tiene una ventana de tiempo en el cual puede funcionar, luego quedara inutilizado.

2.2.4.1. Segundos factores en la autenticación.

A continuación, vamos a listar segundos factores de autenticación que nos pueden ayudar en el desarrollo de nuestro trabajo.

2.2.4.1.1. Verificación de autenticación por SMS.

Es un código temporal generado por el autenticador y enviado a un dispositivo celular del usuario para ser utilizado en la autenticación y probar veracidad del autenticado.

Otra forma más evolucionada a este concepto es que; se ha empezado a utilizar la recepción de un mensaje de verificación por medio de otras plataformas de comunicación como por ejemplo Whatsapp o Telegram que progresivamente muchas aplicaciones han empezado a utilizar como medio de envió de un mensaje de verificación.

2.2.4.1.2. Verificación por llamada telefónica.

Muy parecido a SMS, puede utilizar teléfonos fijos y celulares, existen dos formas. la primera te deletrea un OTP para ser ingresado en el sistema autenticador y la segunda el autenticador nos pregunta si aceptamos o no el ingreso presionando botones opcionales del celular, etc.

2.2.4.1.3. Tarjetas coordenadas.

Mantovani (2019)manifiesta que la tarjeta de coordenadas es una tarjeta de plástico, que contiene una matriz de números impresos y ordenados en filas y columnas, las filas están tituladas con números ascendentes a partir del uno y las columnas con letras ascendentes alfabéticamente comenzando desde la A, en algunos casos el orden es

inverso. El sistema para poder validar la operación pide al probador que ingrese dos de los números que figuran en la tarjeta, como, por ejemplo, las que se ve en la imagen, las celdas C4 y D8. (p. 14)

2.2.4.1.4. Correo electrónico.

Muchas de las aplicaciones acostumbran a enviar un código OTP o un vínculo de verificación temporal al correo electrónico, el usuario deberá ser propietario y tener acceso al correo electrónico el cual fue registrado en el sistema de autenticación para su confirmación, según (OWASP, 2024a, párr. Something you have) existe cierto debate sobre si el correo electrónico constituye una forma de autenticación multifactor, ya que si el usuario no tiene la autenticación multifactor configurada en su cuenta de correo electrónico, simplemente se requiere conocer la contraseña de correo electrónico del usuario (que a menudo es la misma que la contraseña de su aplicación pudiendo ser una vulnerabilidad).

2.2.4.1.5. Token.

Según Mantovani (2019) nos muestra información de autenticación que genera, para que luego el usuario ingrese manualmente en la computadora o ventana de autenticación, contraseña de un solo uso ó OTP que es válida solo para verificación en el tiempo determinado, existen múltiples variaciones que van desde el uso de complejos algoritmos matemáticos, uso de tablas desafío-respuesta, al uso de sincronizaciones de tiempo utilizando el protocolo NTP, etc.

2.2.4.1.6. Software token.

2.2.4.1.6.1. TOTP (contraseña de un solo uso basada en el tiempo).

Según D. M'Raihi, (2011) RFC 6238 es una variante basada en el tiempo, este algoritmo TOTP proporciona valores OTP de corta duración ejemplo 30s, genera contraseñas de un solo uso que utiliza la hora actual como fuente de seguridad y es la

extensión del algoritmo de contraseña de un solo uso basado en HMAC-SHA-1 (HOTP), TOTP implementaciones podría usar funciones HMAC-SHA-256 o HMAC-SHA-512 en base a SHA-256 y SHA-512. (Sección. 1.2)

2.2.4.1.6.2. HOPT (contraseña de un solo uso basada en HMAC).

"RFC 4226, el algoritmo HOTP se basa en un valor de contador creciente y una clave simétrica estática conocida solo por el token y el servicio de validación. Para crear el valor HOTP, se utiliza el algoritmo HMAC-SHA-1" (D. M'Raihi, 2005).

Existen ciertos servicios que ofrecen sus propias aplicaciones y funcionan genéricamente entre los que podemos mencionar Google Authenticator, Microsoft Authenticator.

2.2.4.1.7. Hardware token.

"Los tokens OTP de hardware generan códigos numéricos que cambian constantemente y que deben enviarse para la autenticación. El más conocido de ellos es el RSA SecureID, que genera un número de seis dígitos que cambia cada 60 segundos" (OWASP, 2024a, párr. Hardware OTP Tokens) mientras que Mantovani (2019) manifiesta que al igual que el software generan códigos de un solo uso para validar una transacción los que pueden ser: USB, NFC, RFID con estándares FIDO2, WebAuth, etc.

2.2.4.1.8. Otras formas.

Es un tipo de autenticación que no tiene que ser los servicios del mismo software o autenticador, se llamaría e-autenticación. Por ejemplo: las aplicaciones pueden proporcionar registro de acceso y permitirle usar un inicio de sesión de red social; para iniciar sesión, en lugar de que el usuario cree una nueva identificación, puede iniciar sesión directamente si ya está conectado a la red social.

2.2.5. 2FA y QR.

El mecanismo de autenticación basado en contraseña proporciona la capacidad básica para evitar accesos no autorizados, los riesgos de seguridad son más inevitables para sistemas que dependen de un solo factor, por lo tanto, se ha visto adoptar varios esquemas para fortificar la autenticación, por ejemplo, tarjetas inteligentes, tokens sincronizados, SMS, e-mail, como método de doble factor, etc. a esto valdría añadir un mecanismo de utilización de codificación QR por sus características de almacenamiento de información y lectura de alta velocidad.

La aplicación de la tecnología de codificación QR se podría utilizar de diferentes formas para agilizar la autentificación de 2FA en el sistema de información, por lo que me permito citar los siguientes ejemplos:

1. Una vez que el usuario ingrese su credencial, el sistema informático genera una codificación de imagen 2D QR dentro del cual estará un código aleatorio temporal que el usuario ingresará al sistema informático aplicando el mecanismo de doble factor de autenticación.

2. Ya que QR almacena información se podría utilizar como mecanismo de identificación del usuario, en este caso el usuario presentaría al dispositivo del sistema informático(cámara) una imagen 2D QR mismo que luego de procesar identificaría al usuario, para seguir con el siguiente paso que es confirmar al usuario quien se dice ser ya sea mediante envió de un código temporal por mail, o nuevamente verificación de doble factor por QR.

2.2.6. Seguridades en sistemas de información.

2.2.6.1. Identificación, autenticación, autorización en sistemas de información.

Según IBM MQ (2023) define que la identificación es la capacidad de conocer exclusivamente al usuario de un sistema de información y la autenticación es la capacidad de

demostrar que un usuario o una aplicación es realmente quién se dice ser, entonces la Autentificación es el acto de probar la identidad de un usuario.

Así también en se dice que: “la autorización protege los recursos importantes de un sistema, ya que limita el acceso solamente a los usuarios autorizados y a sus correspondientes aplicaciones o privilegios e impide el uso de recursos que no le corresponden” (IBM, 2021, párr. Autorización).

Antes de dar acceso a un sistema, cada usuario debe tener la autorización formal por parte del custodio o una persona encargada de los datos que están en el sistema de información. Esta autorización respalda la identificación del usuario que, una vez concedida la autorización, el usuario será provisto de un identificador único. Los identificadores únicos y los mecanismos de autenticación son entregados al usuario de manera personal para garantizar que sólo es recibido por el usuario autorizado. Si la cuenta de un usuario deba desactivarse por cualquier razón, el identificador del usuario y mecanismos de autenticación también se debe desactivar, debemos tomar en cuenta que en muchos sistemas de información los tiempos o periodos de activación y renovación o expiración de credenciales son muy importantes para la aplicación.

2.2.6.2. Controles de acceso a los sistemas de información.

Según, Microsoft (2023) el control de acceso es un componente o elemento principal de una seguridad que nos indica quién tiene permiso y quien no tiene permiso para tener verificar o acceso a ciertos datos, aplicaciones, funciones o recursos y en qué condiciones.

Si hablamos de directivas de control de acceso, Microsoft (2023) menciona que estos dependen en gran medida de técnicas como la autenticación y la autorización, que nos permiten a las organizaciones verificar de forma precisa a los usuarios o quienes se dicen ser y verifican que tienen el privilegio adecuado de acceso en base a elementos contextuales como el dispositivo, la ubicación, el rol y mucho más.

Entonces el control de acceso impide que otros usuarios no autorizados tengan acceso a la información confidencial de un sistema de información, en los siguientes párrafos

tenemos los siguientes conceptos sobre controles de acceso los que describimos a continuación.

Según lo que dice Microsoft (2023) entendemos como un nivel jerárquico a lo que se refiere que el Control de acceso discrecional (DAC) es que todo objeto de un sistema protegido tiene un dueño, y solo los propietarios conceden acceso a los usuarios a su discreción. DAC proporciona control de recursos caso a caso u objeto por objeto, según se entiende que podría ser muy privado solo del propietario.

Mientras que en el Control de acceso obligatorio (MAC), los usuarios obtienen acceso en forma de autorización o permiso. Una autoridad centralizada regula los derechos de acceso y los ordena en niveles que se expanden de manera uniforme y equitativa dentro del ámbito de su competencia. Este modelo es muy habitual en contextos gubernamentales y militares (Microsoft, 2023).

El Control de acceso basado en roles (RBAC), se fundamenta en derechos de acceso se conceden de acuerdo con funciones empresariales definidas, no se basa en antigüedades como es más bien es en cargos que ostenta asignando solo datos o funciones que necesita para realizar el trabajo(Microsoft, 2023).

Finalmente, Microsoft(2023) indica que el Control de acceso basado en atributos (ABAC), se concede el acceso de manera combinada o flexible de atributos o funciones, condiciones del entorno, como la hora y la ubicación. ABAC es el modelo de control de acceso más granular y ayuda a reducir el número de asignaciones de roles, podría asignar a dos usuarios dos diferentes roles o iguales según se el caso.

2.2.6.3. Aspectos sobre la autenticación en los sistemas de información.

El termino C.I.A son tres grandiosos valores que está conformados por conceptos de Confidencialidad, Integridad y Disponibilidad.

- **Confidencialidad**, entendemos que es obligatorio que los sistemas seguros rechacen a quienes no deberían ver estos contenidos, la llave de la confidencialidad.

- **Integridad**, garantiza que no ha cambiado y que allí se lee es exactamente lo que se dijo y es confiable, podemos aceptarla como un hecho; con un acceso malicioso la información puede perder su integridad.
- **Disponibilidad**, se refiere a que alguien autorizado pueda acceder sin problema a la información y si es necesario, modificarla y utilizarla, durante un tiempo adecuado.

Por otra parte, se dice que "en los sistemas de información hay dos factores importantes: la usabilidad y la seguridad" (L. F. Cranor and S. Garfinkel, 2004a).

Entendiendo que cuanto más fácil de usar un sistema de seguridad menos seguro se vuelve, ya que en el diseño para buscar esa usabilidad se pierden mecanismos que aumentan la seguridad.

Es por eso debemos indagar ¿hasta qué nivel se debe bajar la seguridad de un sistema para aumentar la usabilidad?, Respondemos: depende de la sensibilidad de nuestra información que se vaya a guardar en el sistema de información.

Continuamos revisamos conceptos básicos que mejorarían un sistema información, según ISO 25010 (2022) y describimos a continuación.

- **La usabilidad,** se entiende por la capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario.
- **Recognoscibilidad de la adecuación.** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Aprendizabilidad.** Capacidad del producto que permite al usuario aprender su aplicación.
- **Operabilidad.** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario.** Capacidad del sistema para proteger a los usuarios de hacer errores.

- **Estética de la interfaz de usuario.** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- **Accesibilidad.** Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.
- **Seguridad,** es la **c**apacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.
- **Confidencialidad.** Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- **Integridad.** Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
- **No repudio.** Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- **Responsabilidad.** Capacidad de rastrear de forma inequívoca las acciones de una entidad.
- **Autenticidad.** Capacidad de demostrar la identidad de un sujeto o un recurso.

Son conceptos importantes que se debe tomar en cuenta para asegurar un sistema de información.

2.2.6.4. OWASP, Web Security Test Guide Versión 4.2.

"WSTG es una metodología de pruebas de seguridad específicamente para aplicación web, explica cómo realizar pruebas basado evidencia de vulnerabilidades dentro de la aplicación"(OWASP, 2024d, Capítulo 4).

2.2.6.4.1. ¿Qué es una vulnerabilidad?

"Una vulnerabilidad es una falla o debilidad en el diseño, implementación, operación o administración de un sistema que podría explotarse para comprometer los objetivos de seguridad del sistema" (OWASP, 2024d, Capítulo 4).

2.2.6.4.2. ¿Qué es una amenaza?

"Un usuario malicioso interno, externo, una inestabilidad del sistema, falla, etc. Puede dañar los activos de una aplicación, infraestructura, recursos de valor, como datos en una base de datos o en el sistema de archivos al explotar una vulnerabilidad"(OWASP, 2024d, Capítulo 4).

2.2.6.4.3. Que es una prueba.

La prueba es una acción para demostrar que una aplicación cumple con los requisitos de seguridad de sus partes interesadas(OWASP, 2024d)**.**

2.2.6.4.4. Metodologías de pruebas OWASP.

Menciona que **"**es una técnica apropiada para probar la seguridad de las aplicaciones web en determinadas circunstancias, basándose en un enfoque de caja negra y que consta de los siguientes participantes, Probador, Herramienta y aplicación**"**(OWASP, 2024d, Capítulo 4)**.**

2.2.6.4.4.1. Pruebas pasivas.

"Conocer la lógica de la aplicación y explorar como usuario, luego se utilizará herramientas para recopilar de información para comprender los puntos de acceso y funcionalidades" (OWASP, 2024d, Capítulo 4).

2.2.6.4.4.2. Pruebas activas.

Según OWASP (2024d) las pruebas activas son:

1. Recopilación de información.
2. Pruebas de gestión de configuración e implementación
3. Pruebas de gestión de identidad
4. Pruebas de autenticación
5. Prueba de autorización
6. Pruebas de gestión de sesiones

7. Pruebas de validación de entrada
8. Manejo de errores
9. Criptografía
10. Pruebas de lógica empresarial
11. Pruebas del lado del cliente
12. Pruebas API.

2.3. Marco Legal.

Según el (Decreto Legislativo 0, Constitución de la República, Art. 16, Literal 2, 2021) se garantiza el acceso universal a las tecnologías de información y comunicación, así también en la (Constitución de la República, Art. 96, 2021) que reconoce todas las formas de organización de la sociedad como expresión de soberanía popular para desarrollar procesos de autodeterminación e incidir en las decisiones y política públicas y en el control social de todos los niveles de gobierno, así como en las entidades públicas y de las privadas que presten servicios públicos.

En la ley de Comercio Electrónico tiene como objetivo regular el tráfico de información en los que establece lo siguiente:

Que, la confidencialidad y reserva, establecen los principios de confidencialidad y reserva para los mensajes de datos, cualquiera sea su forma, medio o intención. Toda violación a estos principios, principalmente aquellas referidas a la electrónica, transferencia ilegal de mensajes de datos o violación del secreto profesional, será sancionada conforme a lo dispuesto en esta ley y demás normas que rigen la materia (Ley No 67, Ley de Comercio Electrónico, firmas y mensajes de datos, Art. 5, 2002).

Que, la protección de datos, para la elaboración, transferencia o utilización de bases de datos, obtenidas directa o indirectamente del uso o transmisión de mensajes de datos, se requerirá el consentimiento expreso del titular de éstos, quien podrá seleccionar la información a compartirse con terceros. La recopilación y uso de datos personales responderá a los derechos de privacidad, intimidad y confidencialidad garantizados por la Constitución Política de la República y esta ley, los cuales podrán ser utilizados o transferidos únicamente

con autorización del titular u orden de autoridad competente. No será preciso el consentimiento para recopilar datos personales de fuentes accesibles (Ley No 67, Ley de Comercio Electrónico, firmas y mensajes de datos, Art. 9, 2002).

Así también los delitos contra la seguridad de los activos de información y comunicación, está destinado a sancionar las acciones ilícitas que realicen en contra de la seguridad de los sistemas de información y comunicación (Ley 0, Código Orgánico Integral Penal, Sección Tercera, Artículos 229-234, 2021).

Que se garantiza el ejercicio del derecho de protección de datos personales que incluye el acceso y decisión sobre información y datos de carácter personal, así como su correspondiente protección (Ley 0, Ley Orgánica de Protección de datos Personales, Artículo 1, 2021).

Con la enumeración de todos estos estamentos legales podemos decir que es un marco legal a fin de respaldar el aseguramiento de la información de los sistemas de información.

CAPITULO III.

3. MARCO METODOLÓGICO.

3.1. Descripción del área de estudio/Descripción del grupo de estudio.

El presente trabajo se enfoca en analizar y reforzar los mecanismos de autenticación para el sistema de información del Comité de Agua Potable y Alcantarillado de la comunidad de Pijál alto, mismo que se ubica en una comunidad rural del Cantón Otavalo de etnia Indígena Kayambi que es organizada con principios de administración comunitaria y para ejecutar los procesos de recaudación y finanzas posee ese sistema que fue desarrollado hace por lo menos 10 años en el que su mecanismo de autenticación no es robusta, se desarrollará e implementará los mecanismos 2FA y QR a fin de robustecer los mecanismos de autenticación.

Esta investigación se centra en estudiar de forma descriptiva y cualitativo los problemas de autenticación del sistema de información existente que utiliza un solo método de autenticación.

3.2. Enfoque y tipo de investigación.

Esta investigación se centró en estudiar de forma descriptiva y comprender cualitativamente los problemas de autenticación de usuario en el sistema de información existente que utiliza un solo método de autenticación.

Con el tipo de investigación descriptivo se buscó y verifico las características principales que posee el sistema de información con respecto a la autenticación.

El enfoque de investigación cualitativo nos ha permitido comprender la información recolectada, con la que se realizó el análisis minucioso e integral a fin desarrollar las soluciones para el sistema de información.

El diseño de investigación fue de campo pues se realizó un el espacio real, con el sistema de información, aplicando Owasp Test Security Guide v4.2; que es un proyecto de código abiertos para determinar y combatir las causas que hacen que el software sea inseguro.

3.3. Procedimiento de investigación.

3.3.1. Revisión de pruebas a la aplicación con OWASP, Web Security Test Guide.

El desarrollo de esta sección son conclusiones que hemos sacado del (**Anexo 1)** en donde está documentado las pruebas aplicadas al sistema de información, nos muestra toda una serie de actividades desarrolladas, para probar evidencia de vulnerabilidades en la aplicación web.

3.3.1.1. Recopilación de información.

3.3.1.1.1. Fugas de información mediante motores de búsqueda.

Con la prueba WSTG-INFO-01 se ha encontrado el acceso a la aplicación web mediante el motor de búsqueda Google por el que podemos acceder a la aplicación sin aún autentificarse a pesar de que no se puede realizar ninguna alteración de los datos de la aplicación, visualizamos la siguiente información personal de usuarios de agua potable y alcantarillado.

- Información personal de usuarios.
- Información de pagos de usuarios.
- Nombres de usuarios del sistema.

Figura 8 WSTG-INFO-01 Información personal de usuarios de agua.

3.3.1.1.2. Huellas dactilares del servidor web.

En la prueba WSTG-INFO-02 concluimos lo siguiente: como este procedimiento es de reconocimiento hemos descubierto que el servidor de nuestra aplicación es un hosting denominado “miarroba.com”, un servidor **nginx con un sistema operativo Hp P 2000 G3 NAS.** Por otra parte, los puertos que están activados para la aplicación web es 443(puerto seguro) y 80(puerto inseguro),

3.3.1.1.3. Identificar puntos de entradas y superficie de ataque.

En la prueba WSTG-INFO-06 podemos mencionar que, superficie de ataque son todos los puntos o mecanismos potenciales que un actor malicioso puede usar para comprometer un sistema y obtener un acceso no autorizado, necesariamente debemos identificarlos para realizar las pruebas a nuestro sistema de información.

En nuestra aplicación encontramos las siguientes superficies de ataque las que luego de una revisión concluimos que son muy importantes para la estabilidad del sistema de información.

Tabla 1 WSTG-INFO-06 Superficie de ataque Owasp.

DIRECCION	PARAMETROS	DEFINICION
https://comite-de-agua.webcindario.com/agua/	METODO POST: VARIABLES: login, password, enviar	Autenticación
https://comite-de-agua.webcindario.com/agua/busqueda_usuario.php?ordenar=apellidos	METODO: GET VARIABLES: ordenar	Ordenar usuarios.
https://comite-de-agua.webcindario.com/agua/busqueda_usuario.php	METODO: GET VARIABLES: GrupoOpciones1, campo, textfield	Búsqueda usuario

https://comite-de-agua.webcindario.com/agua/rayas.php	METODO: GET VARIABLES: ordenar, f_ejecucion, pageNum_lista_usuario, totalRows_lista_usuario, todo.	Ordenar lista de usuarios y paginación de usuario
https://comite-de-agua.webcindario.com/agua/usuario_acceso.php	METODO: GET	Creación de usuarios del sistema

3.3.1.1.4. Huellas del marco de la aplicación.

Según la prueba WSTG-INFO-08 del **(Anexo 1)**, podemos verificar que el servidor de la aplicación es Nginx, lenguaje del lado del servidor es PHP, desarrollado con el editor Dreamweaver.

3.3.1.1.5. Mapa de la arquitectura de la aplicación.

Según la prueba WSTG-INFO-10 y luego de la investigación realizada podemos concluir que es una aplicación pequeña, programado en lenguaje PHP, almacenado en un hosting miarroba.com.

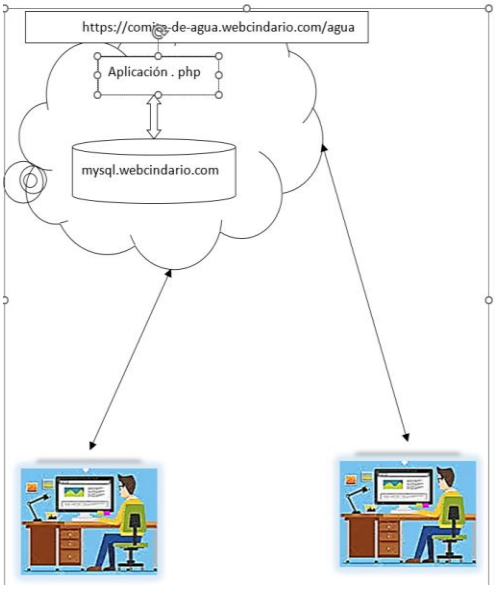


Figura 9 Arquitectura de la aplicación.

3.3.1.2. Pruebas de autenticación.

3.3.1.2.1. Credenciales transportadas a través de un canal cifrado.

Según la prueba WSTG-ATHN-01 hemos podido evidenciar que las credenciales pueden ser transportadas por canales cifrados Https (443) y canales no cifrados Http (80) lo que resulta una vulnerabilidad pues si el sistema de información transporta por un canal inseguro expone la confidencialidad de sus datos transportados a través de la red.

3.3.1.2.1.1. Remediación recomendada por Owasp.

Según (OWASP, 2024d) se debe utilizar HTTPS para tener las siguientes ventajas:

- Impide a los atacantes modifiques las interacciones con el servidor.
- Facilita la escritura de determinadas aplicaciones.

Se debe priorizar HTTPS para operaciones confidenciales y a largo plazo convertir toda la aplicación a HTTPS, los navegadores detectarán como un sitio inseguro si detectan HTTP.

3.3.1.2.2. Credenciales predeterminadas.

Una vez aplicado las pruebas WSTG-ATHN-02 documentadas en el (**Anexo 1**) existen usuarios por defecto y contraseñas predecibles, no existen funciones de bloqueo de cuentas, y podemos concluir que además de existir credenciales por defecto, no existe ninguna política de creación de usuario o política para fortalecer las contraseñas determinamos la vulnerabilidad de la aplicación en esta sección.

3.3.1.2.3. Prueba de mecanismos de bloqueos débil.

3.3.1.2.3.1. Remediación recomendada.

Según OWASP (2024d) se debe tomar las siguientes consideraciones en aplicar mecanismos de desbloqueo de cuentas, según el nivel de riesgo en orden de menor a mayor seguridad así también tomar en cuenta los siguientes puntos:

- Bloqueo y desbloqueo basados en el tiempo.
- Desbloqueo de autoservicio (envía un correo electrónico de desbloqueo a la dirección de correo electrónico registrada).
- Desbloqueo manual del administrador.
- Desbloqueo manual del administrador con identificación positiva del usuario.

Los factores para considerar según OWASP (2024d) para implementar un mecanismo de bloqueo de cuenta son:

1. ¿Cuál es el riesgo de adivinar la contraseña por fuerza bruta contra la aplicación?
2. ¿Es suficiente un CAPTCHA para mitigar este riesgo?
3. ¿Se utiliza algún mecanismo de bloqueo del lado del cliente (por ejemplo, JavaScript)? (Si es así, deshabilite el código del lado del cliente para realizar la prueba).

4. Número de intentos fallidos de inicio de sesión antes del bloqueo. Si el umbral de bloqueo es demasiado bajo, es posible que los usuarios válidos queden bloqueados con demasiada frecuencia. Si el umbral de bloqueo es demasiado alto, más intentos podrá hacer un atacante para forzar la cuenta antes de que se bloquee. Dependiendo del propósito de la aplicación, un umbral de bloqueo típico es un rango de 5 a 10 intentos fallidos.

Según OWASP (2024d)¿Cómo se desbloquearán las cuentas?

1. Manualmente por un administrador: este es el método de bloqueo más seguro, pero puede causar molestias a los usuarios y consumir el "valioso" tiempo del administrador.
    1. Tenga en cuenta que el administrador también debe tener un método de recuperación en caso de que su cuenta se bloquee.
    2. Este mecanismo de desbloqueo puede provocar un ataque de denegación de servicio si el objetivo del atacante es bloquear las cuentas de todos los usuarios de la aplicación web.

2. Después de un período de tiempo: ¿Cuál es la duración del bloqueo? ¿Es esto suficiente para proteger la aplicación? Por ejemplo, una duración de bloqueo de 5 a 30 minutos puede ser un buen equilibrio entre mitigar los ataques de fuerza bruta e incomodar a los usuarios válidos.
3. A través de un mecanismo de autoservicio: como se indicó anteriormente, este mecanismo de autoservicio debe ser lo suficientemente seguro como para evitar que el atacante pueda desbloquear las cuentas por sí mismo.

3.3.1.2.3.2. Evaluación.

Finalizado la prueba WSTG-ATHN-03, en esta sección podemos concluir que la aplicación es vulnerable ya que no posee mecanismos de bloqueo y desbloqueo o ninguna política para contrarrestar este tipo de ataques de fuerza bruta.

3.3.1.2.4. Pruebas de omisión del esquema de autenticación.

Según la prueba WSTG-ATHN-04, la aplicación no es vulnerable a omisiones de esquema de autenticación, pero podemos destacar que por motivos de diseño de la aplicación se permite el acceso a todas las páginas sin autenticación en modo de lectura por lo que consideramos vulnerable según la documentación del (Anexo 1) de pruebas realizadas a la aplicación el problema de la aplicación es el diseño por lo que tendremos que ver la posibilidad de diseñar un nuevo Front-end.

3.3.1.2.5. Prueba de vulnerabilidad de recordar contraseñas.

Según la prueba WSTG-ATHN-05, la aplicación en esta sección resulta vulnerable a token sin vida útil e inyección automática de credenciales ClickJacking y CSRF (falsificación de petición en sitios cruzados), también se debe a la cultura del usuario por la facilidad que busca al confiar sus contraseñas a funciones de navegador o programas que administran las credenciales del usuario.

3.3.1.2.6. Debilidades en prueba del cache del navegador.

La prueba WSTG-ATHN-06 muestra que al parecer en esta sección el sistema de información cumple con las expectativas esperadas solo que debemos recalcar que el diseño

del sistema de información en la sección de usuarios del sistema permite la identificación de usuarios del sistema.

Las directivas Cache-Control, no-cache, no-store y cache-control=must-revalidate existen por lo que la información no será almacenada en cache.

3.3.1.2.7. Políticas de contraseñas débiles.

3.3.1.2.7.1. Remediación propuesta.

Para debilitar el peligro de contraseñas fáciles de, existen opciones; autenticación de dos factores o introducir una política de contraseñas segura o la combinación de las dos alternativas anteriores. El más económico es una política de contraseñas segura, fortaleciendo la longitud, complejidad, reutilización y antigüedad de la contraseña(OWASP, 2024d).

3.3.1.2.7.2. Evaluación.

Según el (Anexo 1) correspondiente a la prueba WSTG-ATHN-07, en esta sección podemos concluir que en la generación de una contraseña para este sistema de información no se ha implementado ninguna política de fortalecimiento de contraseña, llegando a estar vulnerable, el sistema de información no controla la fortaleza de la contraseña, y almacenas credenciales sin cifrado.

3.3.1.2.8. Respuestas débiles a preguntas de seguridad.

La prueba WSTG-ATHN-08 aplicado al sistema de información muestra que el sistema de información no tiene aplicado este tipo de seguridades por lo que se puede considerar vulnerable no se puede determinar la complejidad ni la sencilles de las preguntas ni capacidades de fuerza bruta aplicables.

3.3.1.2.9. Funcionalidades, cambio y restablecimiento de contraseña.

3.3.1.2.9.1. Remediación recomendada.

Según como lo afirma OWASP (2024d) la función de cambio o restablecimiento de contraseña necesita defender con un mecanismo de protección, exigir a los usuarios que se vuelvan a autenticar o presentar métodos de confirmación durante la autenticación

3.3.1.2.9.2. Evaluación.

En la prueba WSTG-ATHN-09, el sistema de información no posee este tipo de funcionalidad por lo que se puede considerar que es vulnerable.

3.3.1.2.10. Autenticación débil en un canal alternativo.

3.3.1.2.10.1. Remediación.

Se puede decir que "se debe asegurarse de que se aplique una política de autenticación coherente en todos los canales para que sean igualmente seguros"(OWASP, 2024d, Sección 4.4.10).

3.3.1.2.10.2. Evaluación.

En la prueba WSTG-ATHN-10, podemos considerar que este sistema de información posee debilidad en la autenticación en el canal alternativo y **no existen canales alternativos que probar**

3.3.1.2.11. Prueba de autenticación multifactorial (MFA).

En la prueba WSTG-ATHN-11 hemos podido evidenciar que debido a la antigüedad de la aplicación y falta de conocimiento esta aplicación no posee mecanismos de autenticación MFA o no se ha podido aplicar la autenticación 2FA y consideramos que la aplicación es vulnerable y es necesario la integración de este mecanismo de seguridad para fortalecer los controles y acceso en el sistema de información.

3.3.1.3. Pruebas de validación de entradas en la autenticación.

3.3.1.3.1. Pruebas de inyección Sql.

En la prueba WSTG-INPV-05 aplicado al sistema de información podemos decir que la aplicación es **vulnerable** a este tipo de ataques, para contrarrestar este tipo de ataque se debe implementar en el código fuente “php” del lado de servidor, instrucciones y medidas que permitan sanear las llegadas de datos ya sea por peticiones POST o GET, impidiendo la introducción de caracteres como apostrofe comillas y otros caracteres especiales que facilitan la anidación de sentencias Sql comprometedoras así también del lado del cliente en los formularios HTML con codificación JavaScript , impidiendo la ejecución de sentencia por inyección.

CAPITULO IV

4. EVALUACIÓN Y RESULTADOS.

4.1. Lista de verificación de pruebas (WSTG) en la autenticación.

Luego de haber realizado las pruebas de acuerdo al manual Owasp WSTG 4.2, secciones 4 (Autentication Testing), que hemos documentado en el **(Anexo1)** de este trabajo; pasamos a presentar la lista de verificación de pruebas de vulnerabilidades seguido del resumen vulnerabilidades encontradas en el sistema de información.

Para mostrar esta lista de verificación y el resumen de vulnerabilidades debemos dar a conocer conceptos fundamentales reconocidos internacionalmente para identificar vulnerabilidades de software y riesgos a las aplicaciones web.

- **"CWE(Common Weakness Enumeration),** es un sistema de categorías para las debilidades y vulnerabilidades del software"(mitre, 2023).
- **"OWASP Top 10,** es un documento de concientización estándar para desarrolladores y seguridad de aplicaciones web que representa un amplio consenso sobre los riesgos de seguridad más críticos para las aplicaciones web"(OWASP, 2024b).

Tabla 2 Lista de verificación Owasp.

Pruebas de autenticación.								
ID	**WSTG-ID**	**Nombre de prueba**	**Descripción**	**Herramienta**	**OWASP Top 10**	**CWE**	**Resultado**	**Item Afectado**
4.1	WSTG-ATHN-01	Prueba de credenciales transportadas a través de un canal cifrado	Evaluar si algún caso de uso del sitio web o la aplicación hace que el servidor o el cliente intercambien credenciales sin cifrado.	Burpsuite/ZAP			No cumple	Autenticación, Creación de cuenta de usuario, Reinicio cambio o manipulación de password, Acceso a recursos mientras está conectado
4.2	WSTG-ATHN-02	Prueba de credenciales predeterminadas	Enumere las aplicaciones para las credenciales predeterminadas y valide si aún existen. Revise y evalúe las nuevas cuentas de usuario y si se crean con valores predeterminados o patrones identificables.	Browser Burpsuite/ZAP Hydra	A7	CWE-1392	No cumple	Credenciales predeterminadas, Prueba de contraseña por defecto en una nueva cuenta
4.3	WSTG-ATHN-03	Prueba de mecanismo de bloqueo débil	Evalúe la capacidad del mecanismo de bloqueo de cuentas para mitigar la adivinación de contraseñas por fuerza bruta. Evalúe la resistencia del mecanismo de desbloqueo al desbloqueo de cuentas no autorizado.	Browser Burpsuite/ZAP Hydra	A7	CWE-307	No cumple	Mecanismos de bloqueo, Mecanismos de desbloque

4.4	WSTG-ATHN-04	Prueba para omitir el esquema de autenticación	de que la autenticación se aplique en todos los servicios que la requieran.	/ZAP	A1 A7	CWE-287 CWE-302 CWE-425 CWE-200	No cumple	solicitud de página directa
4.5	WSTG-ATHN-05	Prueba de vulnerabilidad para recordar contraseña	Validar que la sesión generada se gestione de forma segura y no poner en peligro las credenciales del usuario.	Burpsuite/ZAP	A4	CWE-315 CWE-522 CWE-524 CWE-256	No cumple	Inyección automática de credenciales de usuario, Tokens debe ser analizado en términos de tiempo de vida.
4.6	WSTG-ATHN-06	Prueba de debilidades de la caché del navegador	Revise si la aplicación almacena información confidencial en el lado del cliente Revisar si el acceso puede ocurrir sin autorización.	Browser Burpsuite/ZAP	A4	CWE-525	Aprueba	
4.7	WSTG-ATHN-07	Prueba de política de contraseñas débiles	Determine la resistencia de la aplicación contra la adivinación de contraseñas por fuerza bruta utilizando diccionarios de contraseñas disponibles evaluando los requisitos de longitud, complejidad, reutilización y antigüedad de las contraseñas.	Burpsuite/ZAP Hydra	A7	CWE-521 CWE-1391	No cumple	No existe política de generación de password
4.8	WSTG-ATHN-08	Prueba de respuesta a preguntas de seguridad débiles	Determine la complejidad y cuán sencillas son las preguntas Evalúe las posibles respuestas de los usuarios y las capacidades de fuerza bruta.	Browser Burpsuite/ZAP	A7	CWE-640	No cumple	No existe mecanismos de preguntas de seguridad

4.9	WSTG-ATHN-09	Prueba de funcionalidades débiles de cambio o restablecimiento de contraseña	Determinar la resistencia de la aplicación a la subversión del proceso de cambio de cuenta permitiendo que alguien cambie la contraseña de una cuenta Determine la resistencia de la funcionalidad de restablecimiento de contraseñas contra adivinanzas o elusiones	Browser Burpsuite/ZAP	A7	CWE-620 CWE-640	No cumple	No existe funcionalidades de restablecimiento de contraseñas
4.10	WSTG-ATHN-10	Prueba de autenticación más débil en un canal alternativo	Identificar canales de autenticación alternativos. Evaluar las medidas de seguridad utilizadas y si existen omisiones en los canales alternativos.	Browser Burpsuite/ZAP	A7	CWE-288	No cumple	No existe un canal alternativo de autenticación
4.11	WSTG-ATHN-11	Prueba de autenticación Multi-Factor (MFA)	Identificar el tipo de MFA utilizado por la aplicación. Determinar si la implementación MFA es robusto y seguro. Intentar evadir MFA.	Browser Burpsuite/ZAP	A7	CWE-288 CWE-304 CWE-308	No cumple	No existe MFA autenticación.
Pruebas de Validación de entrada								
7.5	WSTG-INPV-05	Prueba de inyección SQL	Identificar puntos de inyección SQL. Evaluar la severidad de inyección y el nivel de acceso al que puede ser logrado	Burpsuite/ZAP SQLMap NoSQLMap	A3	CWE-89	No cumple	Consultas múltiples

En la tabla 2 listamos las pruebas de acuerdo a evidencias, especificamos detalladamente las pruebas aplicadas, herramientas utilizadas para probar el sistema de información y el resultado obtenido en cada una de las pruebas, los que hemos podido documentar en el **(Anexo 1), pruebas al sistema de información mediante Owasp**.

Como podemos ver en esta tabla la columna "WSTG-ID", nos indica el código de prueba especificado por Owasp para aplicar a nuestro sistema de información, y la columna "nombre de prueba" nos indica la prueba que se aplica, en las columnas siguientes nos describe las herramientas utilizadas y el resultado por cada prueba así como valoración CWE y Owasp top 10 reconocido en escala de riesgo, hemos podido evidenciar que en la mayoría de las pruebas aplicadas, el sistema de información no cumple con lo esperado, a excepción de: prueba de vulnerabilidades de cache de navegador.

Pasamos a la tabla 3 para continuar con el siguiente proceso de evaluación de resultados en el sistema de información.

4.2. Resumen de vulnerabilidades en la autenticación de la aplicación web.

Tabla 3 Resumen de vulnerabilidades.

No	WSTG-ID	Nombre del problema	Riesgo	Probabilidad	Impacto	Mapa OWASP Top 10	Observacion e implicación	Recomendación
1	WSTG-ATHN-01	Autenticación, (Rastreo de credenciales)	Critico	Alto	Alto	A2	Usuario y contraseña transportado sobre canales inseguros http	Usar HTTPS para todo el sitio
2	WSTG-ATHN-01	Creación de cuentas (Rastreo de credenciales)	Critico	Alto	Alto	A2	Exposición de detalles del nuevo usuario	Usar HTTPS para todo el sitio e implementar HSTS
3	WSTG-ATHN-01	Reinicio, Cambio o Manipulación de Password(Rastreo de credenciales)	Critico	Alto	Alto	A2	Exposición de detalles de reinicio, cambio y manipulación de contraseña.	Usar HTTPS para todo el sitio e implementar HSTS

4	WSTG-ATHN-01	Acceso a recursos mientras está conectado (Token de sesión sin cifrado)	Critico	Alto	Alto	A2	Token de sesión se envía sin cifrar mediante el uso de http	Usar HTTPS para todo el sitio e implementar HSTS
5	WSTG-ATHN-02	Prueba de credenciales por defecto	Critico	Alto	Alto	A7	Se ha encontrado credenciales por defecto	No implementar ninguna credencial predeterminada
6	WSTG-ATHN-02	Contraseñas por defecto en una nueva cuenta	Critico	Alto	Alto	A7	No se encontró ninguna política de generación de password	Adaptación de las políticas de longitud, complejidad y rotación de contraseñas a la directriz de NIST 800-3b en la sección 5.1.1
7	WSTG-ATHN-03	Sin mecanismos de bloqueo	Critico	Alto	Alto	A7	no se ha implementado un mecanismo de bloqueo	Implementar mecanismos de bloqueo en el sistema de información
8	WSTG-ATHN-03	Sin mecanismos de desbloqueo	Critico	Alto	Alto	A7	No se ha implementado un mecanismo de desbloqueo	Aplicar mecanismos bloqueo de cuenta dependiendo del nivel de riesgo
9	WSTG-ATHN-04	Solicitud de página directa (Acceso a información confidencial)	Alto	Alto	Moderado	A1 A4	Acceso a información confidencial sin autenticación	Limitar el acceso a información confidencial basado en autenticación
10	WSTG-ATHN-05	Inyección automática de credenciales de usuario	Critico	Alto	Alto	A4	Credenciales no debe ser almacenados de ninguna forma del lado del cliente	Tokens debe ser creado del lado del servidor
11	WSTG-ATHN-05	Token´s sin tiempo de expiración	Critico	Alto	Alto	A4	Token´s nunca expira.	Asegurarse se aplicación tiempo de expiración de la sesión
12	WSTG-ATHN-07	No existe políticas de fortalecimiento de contraseña.	Critico	Alto	Alto	A7	Solo existe password estático por usuario.	Debe ser aplicado políticas de fortalecimiento de password

13	WSTG-ATHN-08	No existe preguntas de seguridad	Moderado	Moderado	Moderado	A7	No existe preguntas de seguridad implementados en el sistema de información	Implementar preguntas de seguridad en recuperación de password
14	WSTG-ATHN-09	No existe funcionalidades de reinicio	Moderado	Moderado	Moderado	A7	No esta implementado funcionalidades de reinicio de password	Implementar funcionalidades de reinicio de password.
15	WSTG-ATHN-10	No existe canal alternativo de autenticación	Bajo	Bajo	Bajo	A7	No este implementado canal alternativos de autenticación	implementar canales alternativos de autenticación.
16	WSTG-ATHN-01	No existe autenticación MFA	Critico	Alto	Alto	A7	No esta implementada autenticación de doble factor	Implementar un mecanismo de autenticación de doble factor.
17	WSTG-INPV-05	Inyección Sql	Critico	Alto	Alto	A3	No está implementado funcionalidades de saneamiento de datos para caracteres especiales	Implementar funciones de saneamiento de datos en código php

La tabla 3 que hemos mostrado es el resumen de vulnerabilidades; describe específicamente que problemas tiene el sistema de información por cada una de las pruebas aplicadas anteriormente, para cada uno de los problemas se ha evaluado el riesgo, probabilidad e impacto en base parámetro sugeridos en el **(Anexo 2), parámetros de evaluación y factores de riesgo**; de este documento.

Mostramos la escala en riesgo (Owasp top 10) y se detalla las debidas observaciones e implicaciones que podríamos tener con estos problemas, también las recomendaciones necesarias para remediar estas vulnerabilidades.

CAPITULO V

5. PROPUESTAS, DESARROLLO DE SOLUCIONES Y PRUEBAS.

5.1. Propuesta.

Con resultados de la evaluación se ha determinado que el problema es la seguridad en la autenticación del usuario sus procesos de aseguramiento y transporte de credenciales por canales seguros, procesos de bloqueo y restablecimiento de credenciales, que son deficientes; por lo que debemos implementar funcionalidades adicionales para mitigar estas deficiencias en el sistema de información, así también aplicar factores de autenticación que permitan fortalecer la seguridad.

Recordando nuevamente según lo que dice: L. F. Cranor and S. Garfinkel (2004b) en los sistemas hay dos factores importantes a tomar en cuenta: la usabilidad y la seguridad; si es más fácil de usar el sistema puede volverse inseguro, mientras que si se procura la seguridad puede ser dificultoso para el usuario ya que el sistema pedirá otros requisitos en la autenticación.

Para aliviar este inconveniente de usabilidad frente a la seguridad hemos tenido que proponer el mecanismo de autenticación de código QR que es una funcionalidad de autenticación que permite a un dispositivo registrado escanear un código QR para autenticar al usuario, podría proporcionar un método alternativo al ingreso de un usuario y contraseña en un formulario de autenticación.

Entre las funcionalidades de autenticación y procedimientos de acceso en el sistema de información que se propuso mejorar son:

- Usar HTTPS para todo el sitio e implementar directivas HSTS(*Strict-Transport-Security*).
- Implementar políticas para generación de usuarios.
- Implementar políticas de generación de contraseñas con la entropía suficiente para dificultar el descifrado.

- Hashing de contraseñas.
- Implementar mecanismos de restablecimiento y cambio de contraseñas.
- Prevención de salto u omisión del esquema de autenticación (acceso sin autenticación a paginas).
- Generación tokens asegurados del lado del servidor.
- Implementar mecanismos de autenticación 2FA.
- Adaptación de lectura QR para la autenticación de usuarios.
- Cifrado y descifrado de datos en QR.
- Prevención de inyección Sql.

5.2. Desarrollo e implementación de las soluciones.

5.2.1. Implementación de directivas HSTS(*Strict-Transport-Security*).

Para la implementación de HSTS en nuestra aplicación sin habilitar para todo el sitio hemos configurado el encabezado HSTS mediante la asignación dentro del código “php” la siguiente directiva: **header("strict-transport-security:max-age=600"),** según (**Anexo 4) sección HSTS del código fuente**.

Una vez habilitado HSTS si realizamos un direccionamiento con Http, el navegador verificará esta directiva y cambiará a una navegación Https seguro.

- http://juntadeaguapijal.webcindario.com/agua/admin/login.php

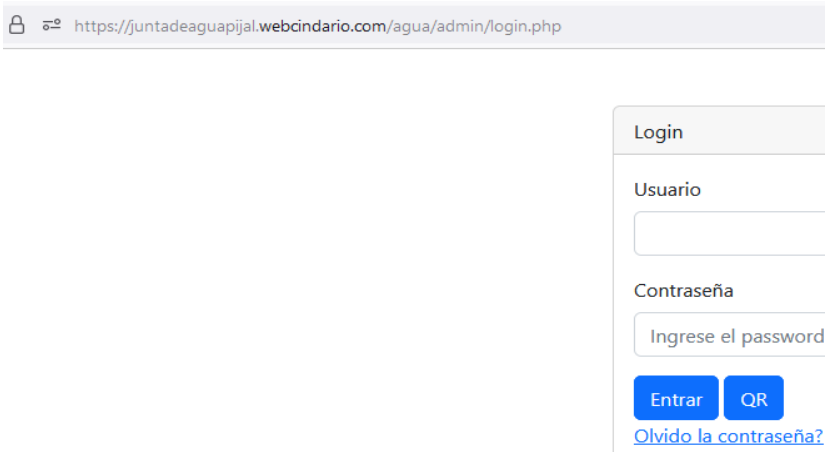


Figura 10 Propuesta, aplicando HTTP Strict Transport Security (HSTS).

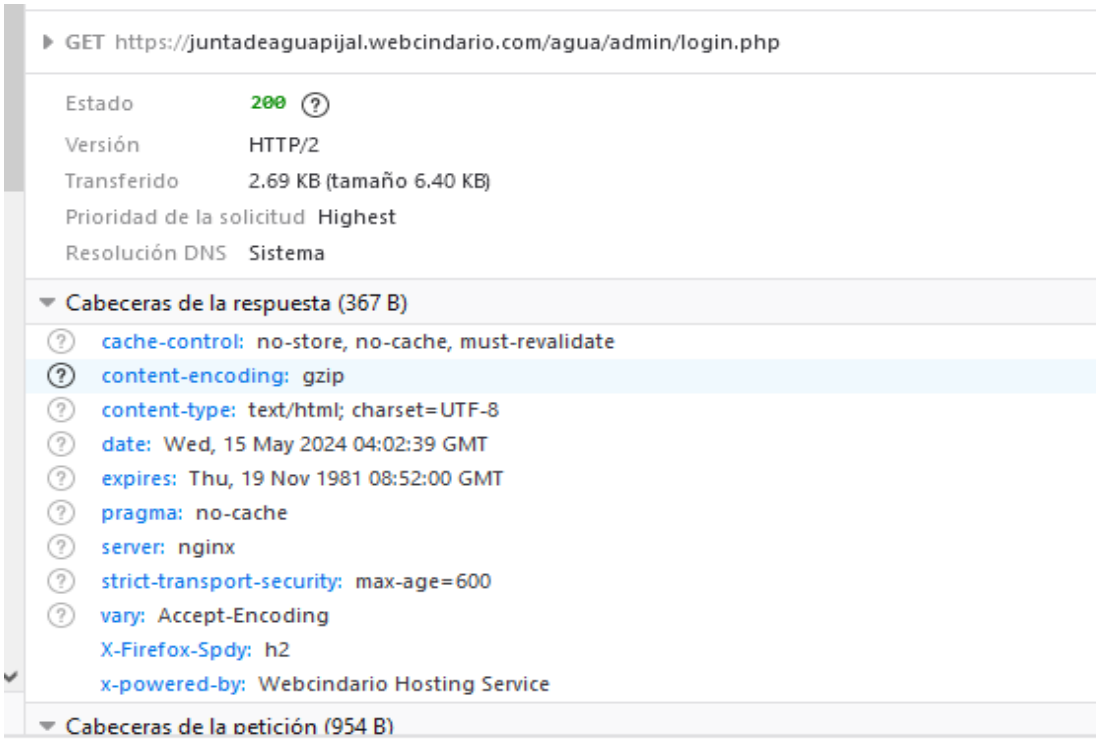


Figura 11 Propuesta, verificación HSTS.

5.2.2. Implementación de políticas para generación de usuarios.

Para la generación de usuarios hay que tomar en cuenta las siguientes normas y requerimientos:

- Los directivos son elegidos mediante elección en asamblea para un periodo económico de dos años por lo que para la administración del sistema de información se creará un usuario y contraseña con sus roles dentro del sistema de información.
- Para la creación del nombre de usuario dentro del sistema es necesario registrar los siguientes datos de la persona:
  - Año lectivo.
  - Login.
  - Password.
  - Nombre
  - Cargo.
  - Fecha de inicio del periodo.

- Fecha final del periodo.
- Estado de usuario activo/bloqueado.
- Correo.

- Los usuarios del sistema de información tienen periodo de inicio y final de acuerdo con el periodo económico para el que fueron elegidos.
- Los directivos pueden ser electos nuevamente por lo que para cada periodo fiscal se generará un nuevo usuario y contraseña correspondiente a la persona que en ese periodo va a ser directivo.
- Dado que un directivo puede ser reelecto y por cada periodo debe tener una nueva sesión de usuario y contraseña no se podrá utilizar el correo electrónico como recurso de autenticación.
- Para la generación de usuario el sistema de información revisara la existencia de el nombre de usuario dentro de su base de datos.
- Se ha recomendado crear usuario con el siguiente formato JAAPI+<año>+<nombre sugerido por el usuario>

Se ha tomado estos parámetros como norma para la creación de usuarios nuevos a partir del periodo de administración 2024 por lo que en el formulario de creación de usuario se sugiere este formato.

Figura 12 Propuesta, política de generación de usuario.

5.2.3. Políticas de generación de contraseña.

Para la política de generación de contraseña se ha tomado las siguientes directrices:

- Debe tener al menos 12 caracteres.
- La contraseña debe tener al menos una letra mayúscula.
- La contraseña debe tener al menos un número.
- La contraseña debe tener al menos un carácter especial.

Figura 13 Propuesta, fortaleza de contraseña.

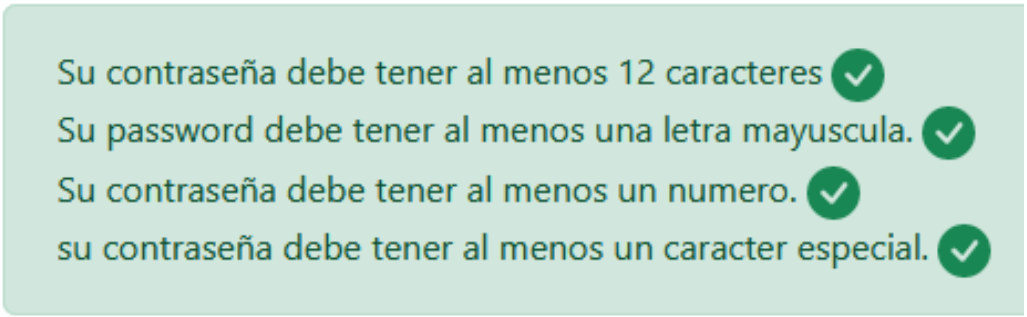


Figura 14 Propuesta, verificación de requisitos de contraseña.

5.2.4. Hashing de contraseña.

En el en sistema de información que hemos diagnosticado anteriormente hemos encontrado contraseñas de acceso sin cifrado en la base de datos, por lo que hemos procedido a utilizar funcionalidades PHP para almacenar las claves de acceso en forma de Hashe`s, según el código fuente que lo podemos encontrar en **(Anexo 4),** generación de hash de contraseña.

Ahora podemos visualizar en la tabla de contraseñas de la base de datos, las contraseñas cifradas.

ɔ	login	password	cargo
13	JAAPI2024PRUEBA1	$2y$09$ahpFC5yB2lUTYb2BlrwiaeYPgKmTFdfeh4L.Eh6/LMW...	presidente
13	jaapi2024	$2y$09$V7yucFQLYedolbOl8FLZa.U0ZlJjf8BM7qrUUXks3.L...	presidente
13	darwinp	$2y$10$hFganOuQqnWJ7mtGz.2gGeqPpL4CrmmWvp8uQl1.ocb...	presidente
13	rodrigo2024	$2y$10$3S6bFylcCKYjmdh1PFe/z.XzmODz6A9fYKo2l3XX3Pu...	tesorero
13	cecilia2024	$2y$09$U/gdZ7TiK1b5PuCBFeFRQOW1p1c1/7r8QsJ7S3ELtvN...	secretario
2	root	$2y$09$1xK8a.w746xVl8m59G2geewXEYGOuubLZX21R9oaDmr...	presidente
2	rodrigo	rqa123456	tesorero
3	2013	rqa123456	presidente
5	cooppijal	pijal1234	tesorero
5	root2016	rqa123456	presidente
6	lolita2017	lolita2017	presidente
6	pijalgonzal	pijalgonzal	tesorero
6	pijalrumina	runinahui12	tesorero

Figura 15 Propuesta Hashing de contraseña en base de datos del sistema.

Aquí se puede visualizar la tabla en donde se almacena los datos del usuario para el acceso al sistema de información, el campo **password** se almacena en forma de un hash, se debe aclarar que las contraseñas de periodos anteriores a 2024 están almacenadas en forma de texto visible.

5.2.5. Implementar restablecimiento y cambio de contraseñas.

El principal mecanismo para el restablecimiento o cambio de contraseñas en nuestro sistema de información es el vínculo ¿olvidó la contraseña? En la ventana de acceso al sistema; luego del cual pasamos a una función o página “recuperar.php” que le pide el correo electrónico al que está vinculado la cuenta de acceso, para poder enviar un mensaje de código secreto de verificación del usuario.

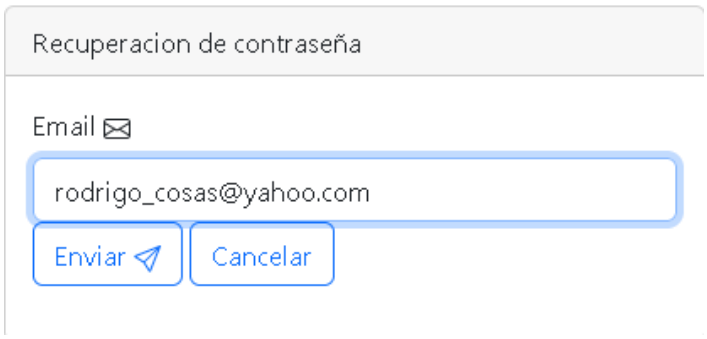


Figura 16 Propuesta, recuperación o restablecimiento de contraseña.

La generación de un código numérico aleatorio y el hash de este código de verificación lo podemos verificar en el (**Anexo 4**), **generación de código numérico hash aleatorio para restablecimiento de contraseña**.

Una vez inviado el mensaje al correo electrónico del usuario pasamos a la página de verificación "verificacion.php", donde el usuario debe ingresar el código temporal que previamente ha llegado al correo electrónico para poder hacer efectivo una ventana de restauración de contraseña.

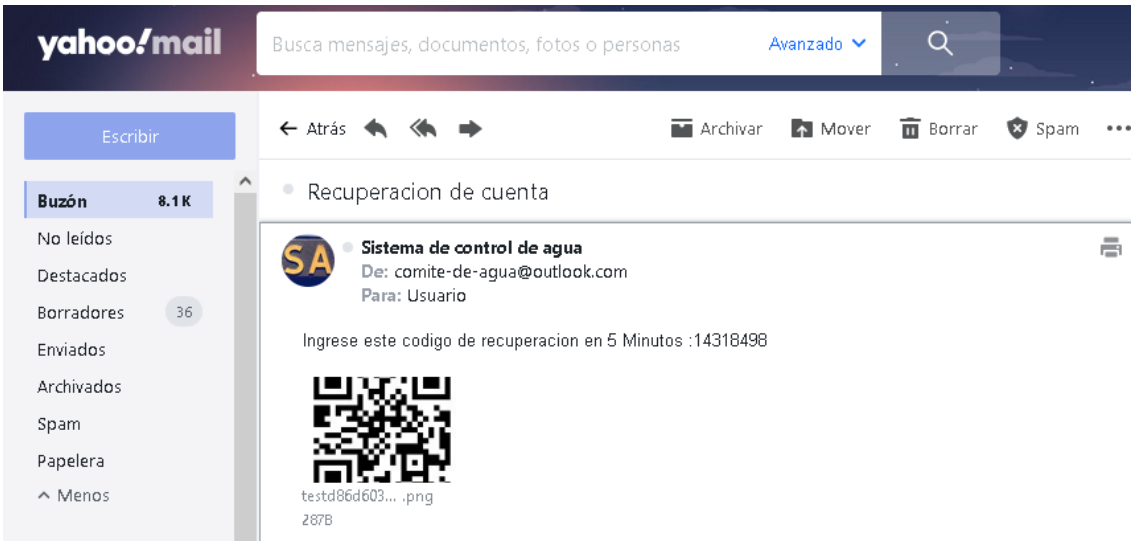


Figura 17 Propuesta, recepción de correo electrónico.

Para continuar con el proceso final de restauración de contraseña introducimos el código temporal llegado al correo electrónico en la misma página "recuperar.php", para verificar la autenticidad del usuario.

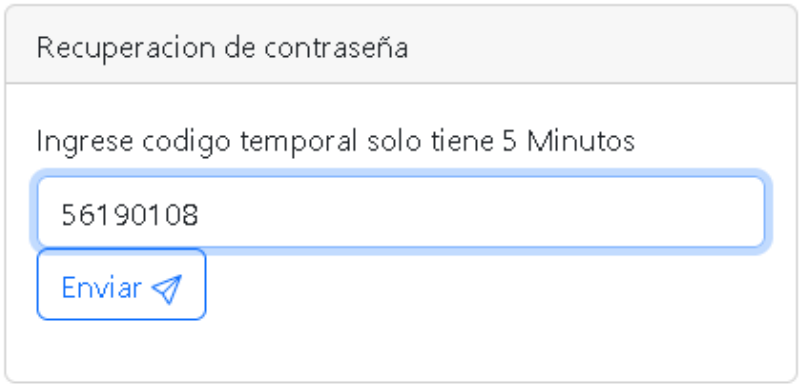


Figura 18 Propuesta, código temporal para restablecimiento de contraseña.

Este código numérico es enviado a la página "cambiopassword.php", se realiza el proceso de restablecimiento de contraseña del usuario, una de las medidas que se toma en este proceso es el de verificación de fortaleza de la contraseña.

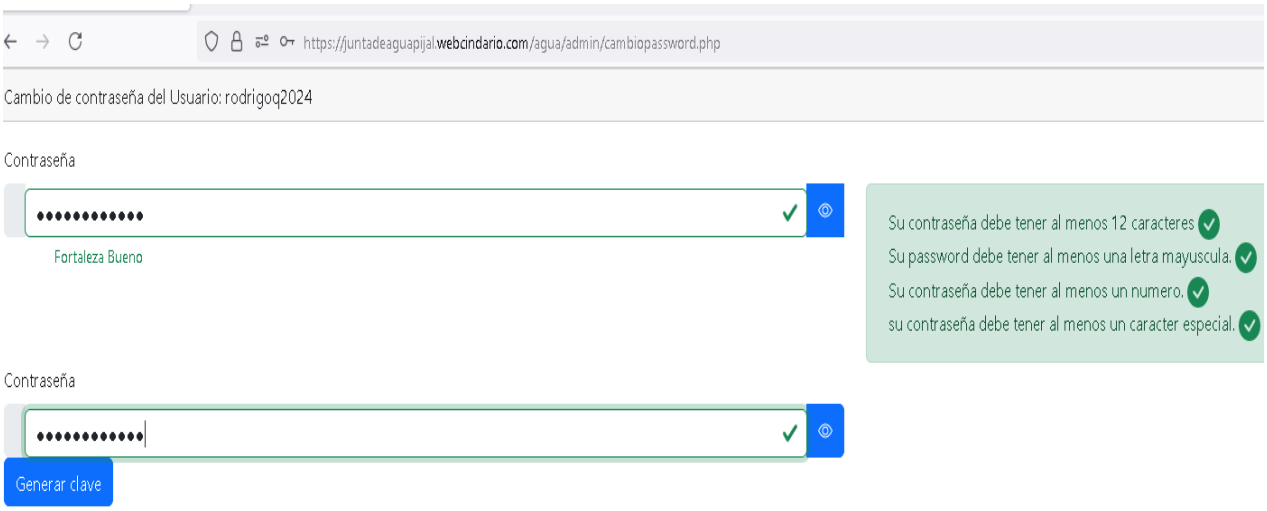


Figura 19 Formulario cambio de contraseña.

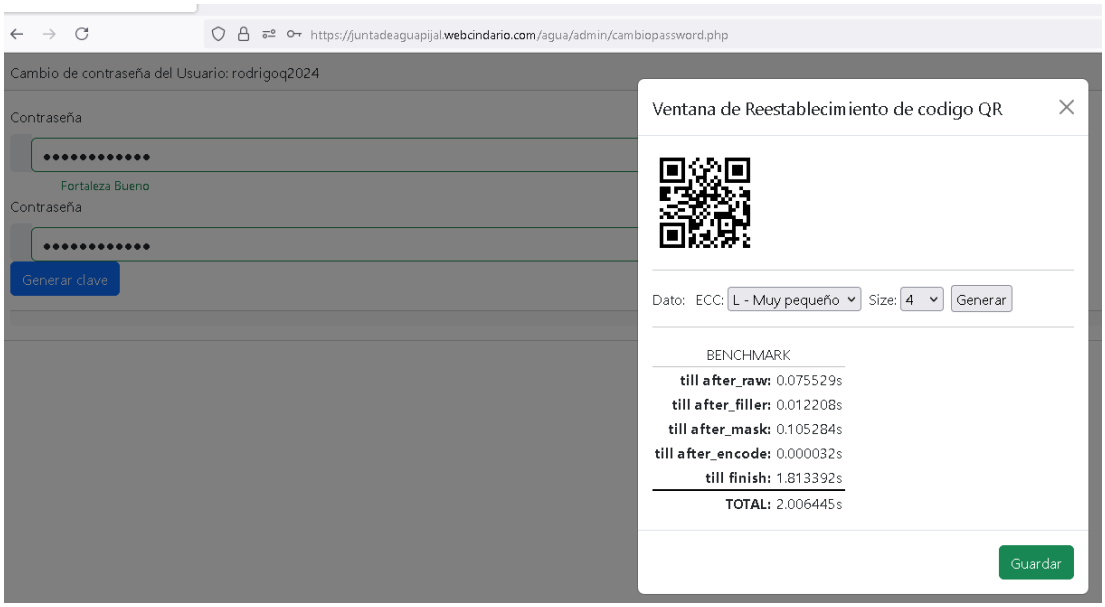


Figura 20 Propuesta, restablecimiento de credenciales y QR de usuario.

La imagen anterior nos muestra el último proceso en que generamos un código QR del usuario con la nueva contraseña.

La lógica para realizar el cambio de contraseña lo que hemos mostrado podemos encontrar en el (**Anexo 4**), **lógica del mecanismo de cambio de contraseña.**

5.2.6. Prevención de salto u omisión del esquema de autenticación.

En el anterior sistema de información, mientras se autenticaba para el acceso a un usuario esto controlaba solo los permisos de escritura dependiendo del rol en el sistema de información, se ha descubierto que sin la autenticación se puede acceder a todas las páginas del sitio, esto da como resultado la exposición de datos de consumidores de agua, usuarios del sistema, en general una exposición de datos confidenciales según las pruebas WSTG-INFO-01 y la prueba WSTG-ATHN-04 ; que hemos realizado a nuestra aplicación. Esto se debe a que el problema es el diseño de la aplicación web teniendo que trazar un nuevo Front-End de la aplicación y nos permita un control total del acceso a las páginas de la aplicación.

- https://juntadeaguapijal.webcindario.com/agua/admin/login.php

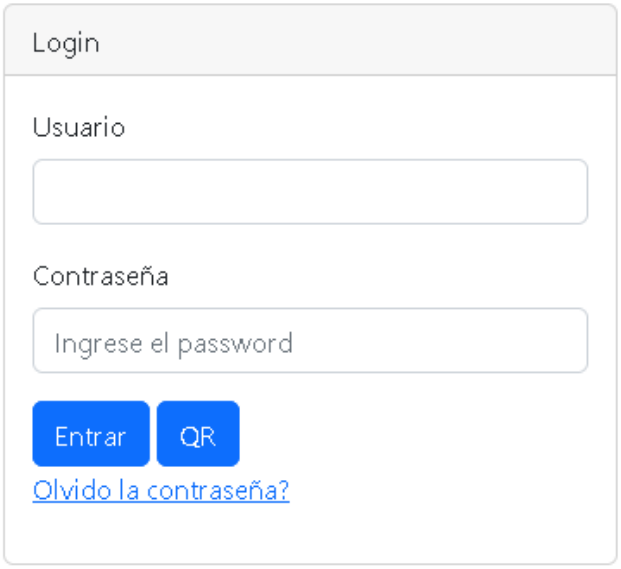


Figura 21 Propuesta, ventana principal autenticación del sistema de información.

El archivo “login.php” , es la ventana principal de autenticación en el que consta de una estructura principal para la autenticación que comprueba al usuario dentro de la base de datos, la lógica consta de una sentencia Sql para revisar la existencia del usuario y recuperación del hash de contraseña para realizar una comprobación , en el caso de que el usuario es correcto realizará algún procedimiento de verificación de doble factor redireccionándose a la página de “verificacion.php” , con el fin de aplicar autenticación de Doble Factor, según el código fuente **(Anexo 4), verificación de usuario y redirección.**.

Si la aplicación de doble factor es necesario se lo realiza en el archivo “verificacion.php”, en donde nuevamente realiza una comprobación ya sea por QR o por correo electrónico según sea el caso para, finalizar creando sesiones para el usuario del lado del servidor, mostramos el procedimiento de verificación, **(Anexo 4), verificación por QR**.

Una vez terminado de verificar en el procedimiento de doble factor se redirecciona a la página “index.php” y todas otras páginas de la aplicación que necesariamente en el encabezado poseen un código adicional de verificación de la existencia o no de una variable SESSION[‘usuario’], para permitir el despliegue, caso contrario nuevamente redireccionará a la página “login.php”, de petición de credenciales, **(Anexo4), comprobación de acceso directo a páginas.**

5.2.7. Generación token’s de sesión seguros del lado del servidor.

Las sesiones son valores creados del lado del servidor que son utilizados para mantener la relación entre el cliente navegador y el servidor, quien almacena esa sesión dentro de una carpeta temporal y el cliente se identifica mediante el valor de esa sesión. Los valores o características en la generación de estas sesiones son inseguros por defecto; por lo general amenazas como ataque de fijación de sesión y secuestro de sesión son muy frecuentes.

Para el desarrollo en nuestra aplicación hemos visto la necesidad de realizar tres funciones principales en la asignación de sesiones; asignar valores o inicializar el comportamiento de la sesión, regeneración de sesiones, iniciar la sesión.

Según el **(Anexo 4), generación de sesiones seguras**; en el código que mostramos podemos se distinguir 3 funciones que serán aplicadas para cada una de la página que utilizarán sesiones, correr_sesion(), regenerar_session_id, iniciar_sesion(), según podemos identificar realizan tareas para asegurar una sesión.

5.2.8. Implementar mecanismos de autenticación 2FA.

Para la autenticación 2FA se ha tomado en cuenta dos formas de autenticación ya sea por correo electrónico o verificación por QR utilizando la aplicación Google Autenticator.

5.2.8.1. 2FA por correo electrónico.

Una vez realizado proceso de autenticación, se envía un correo electrónico con un código aleatorio el cual tiene solo 5 minutos para poder ingresarlo, caso contrario el código se invalida.

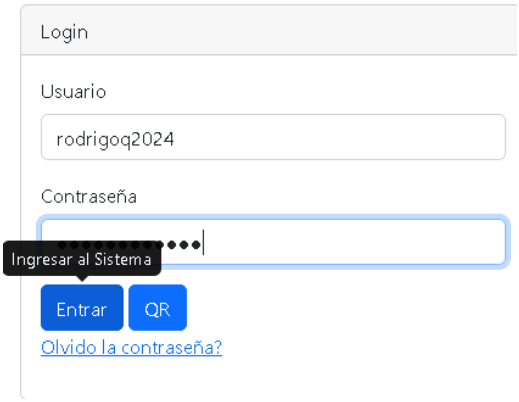


Figura 22 Propuesta, inicio de sesión para 2FA.

Figura 23 Propuesta, correo electrónico con código temporal.

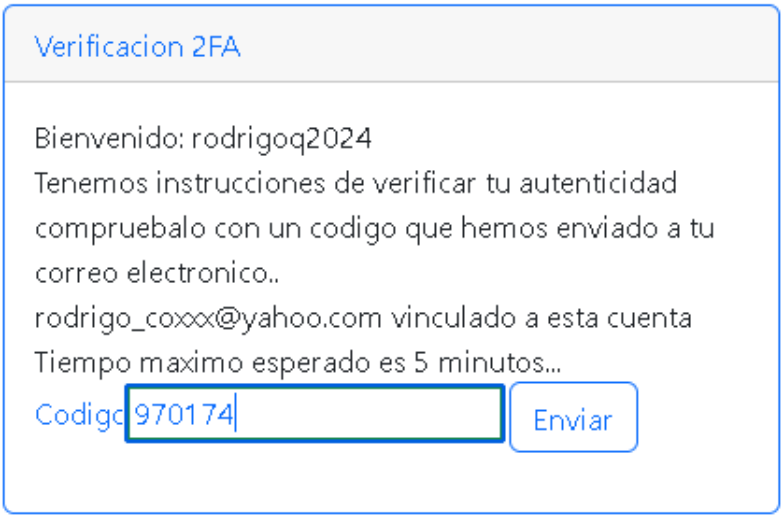


Figura 24 Propuesta, verificación 2FA.

Si por motivos de tiempo no se ingresa el código con anticipación se expirará el acceso, o si el código que ingrese es incorrecto se procederá nuevamente a pedir la autenticación; si el proceso es correcto la autenticación 2FA será exitosa.

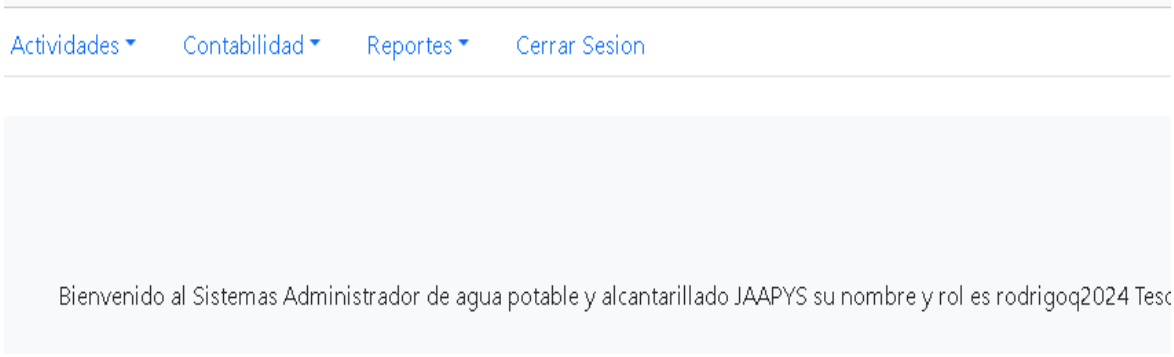


Figura 25 Propuesta, sesión iniciada mediante 2FA.

El código que mostramos en el **(Anexo 4)**, **generar y enviar código temporal por correo electrónico**, permite verificar la autenticación del usuario y como un segundo paso enviar por correo electrónico un código aleatorio de segundo factor para autenticación.

La página de verificación es el siguiente proceso, en el que verificamos el código aleatorio recibido por correo electrónico donde realizamos la comparación del código para permitir el acceso al usuario, (**Anexo 4), verificación 2FA con código temporal**.

5.2.8.2. 2FA por QR.

Este procedimiento de autenticación de doble factor se basa en la generación de un código secreto del cual se genera el QR para posteriormente decodificar en base a este un segundo código de comprobación mediante la aplicación Google Autenticator instalada en un dispositivo de celular, e ingresar para su comprobación.

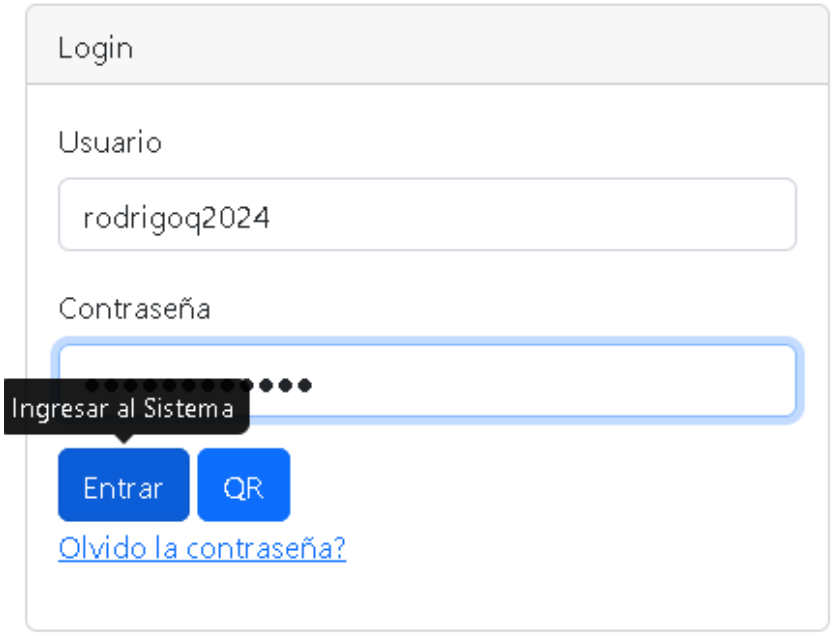


Figura 26 Propuesta, inicio de sesión para QR.

Figura 27 Propuesta, verificación 2FA por QR.

Se ingresa el código (Time-Based One time password) **225195** luego de decodificar mediante la aplicación Google Autenticator, se realiza una prueba de verificación para permitir el acceso al usuario.

Para la generación del código secreto se ha utilizado la librería PHP_GoogleAuthenticator, mediante el cual generamos el código QR un segmento de código ubicado en el archivo "verificacion.php" **(Anexo 4)**, **generación QR para 2FA.**

El procedimiento para verificar el código decodificado lo encontramos en el archivo "verificacion.php", demostramos en esta sección mediante la siguiente línea:

$checkResult = $ga->verifyCode($secret, $oneCode, 2), una vez verificado Pasamos a generar la sesión del usuario con los datos de autenticación recuperados de la base de datos (**Anexo 4),Verificación de QR y asignación de sesión**.

5.2.9. Adaptación de lectura QR para la autenticación de usuarios.

Cuando se crea un usuario en el sistema de información ya sea presidente, secretario, Tesorero, Operador, o Cooperativa, necesariamente en el sistema de información se guardar una imagen QR con la identificación del usuario y password, el que es entregado a la persona que es propietaria de la cuenta en el sistema de información, también si es necesario es posible enviar esta imagen QR al correo electrónico correspondiente a la cuenta.

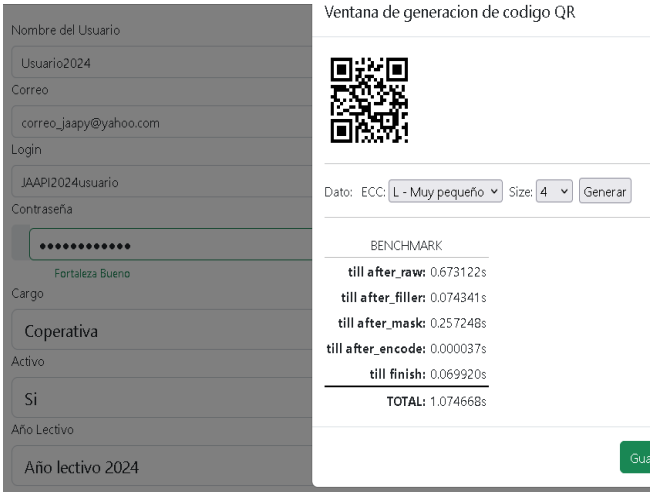


Figura 28 Propuesta, generación QR de autenticación de usuario.

En la ventana de autenticación existe el botón QR que permite activar la cámara de la Pc para indicar la imagen QR y autenticarse sin necesidad de tener que introducir la autenticación mediante el teclado.

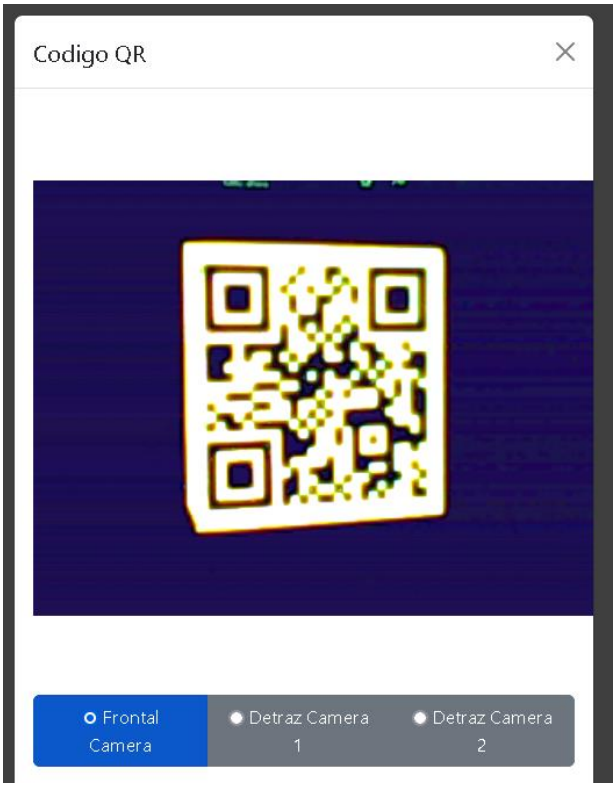


Figura 29 Propuesta, autenticación mediante QR.

Después de que la imagen es reconocida, el sistema de información continuará con el siguiente proceso de autentificación 2FA que ya hemos explicado en las secciones anteriores.

Figura 30 Propuesta, 2FA después de autenticación QR.

El código para activar la cámara de ordenador es incrustado de lado cliente de la aplicación un código java Script que activa la cámara y realiza una lectura del código QR utilizando una librería java script, (**Anexo 4), Activando cámara con Java Script.**

La librería adicional utilizada es instascan.js disponible para lectura de códigos QR del lado cliente navegador.

5.2.10.Funciones de cifrado y descifrado de datos en QR.

Puesto que una imagen QR puede ser decodificado con un lector, al momento de almacenar una credencial corre el riesgo de que las credenciales del usuario sean visibles si estos no son cifrados; en el (**Anexo 4**), **funciones de cifrado y descifrado de datos utilizados en QR:** encontramos el código utilizado para primero cifrarlas credenciales antes de ser almacenados en una imagen QR.

En el proceso de lectura de credenciales QR, solamente se verá una codificación cifrada que posteriormente el sistema de información pasará a un proceso de descifrado del lado del servidor para poder verificar la credencial del usuario.

5.2.11.Aplicar prevención contra inyección Sql.

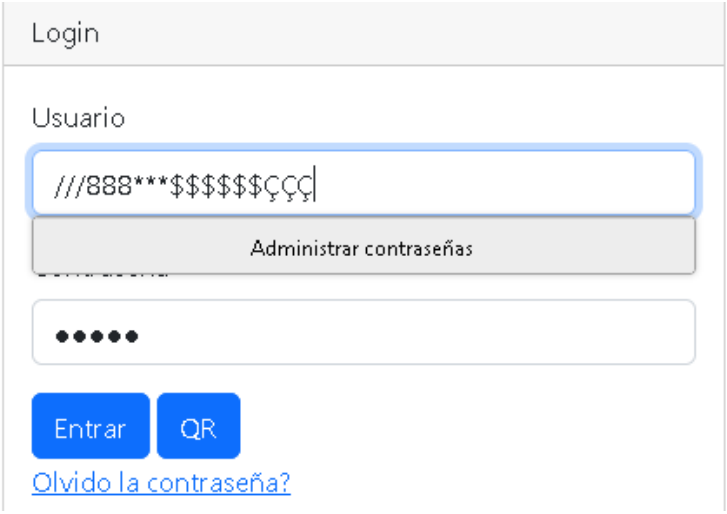


Figura 31 Propuesta, Prevención de inyección Sql.

Una de las principales preocupaciones para prevenir la inyección Sql, es el saneamiento de entradas o Input en los formularios de una aplicación web, el propósito del saneamiento es prevenir la entrada de caracteres específicamente reconocidas que pueden concatenar consultas adicionales a las entradas de datos, entre los caracteres especiales que debemos sanear para el método $_POST y $_GET son:

Tabla 4 Caracteres especiales, que permiten inyección Sql.

ASCII Encoding Reference	The HTML Coded Character Set	CARÁCTER	NOMBRE
%27	'	‘	Apostrofe
%28	(	(	Abrir paréntesis
%29	)	)	Cerrar paréntesis
		+	Más
		=	Igual

Para quitar peligrosidad estos valores basados en codificación html tendremos que impedir la digitación de los caracteres & y # quedando como resultado solo un numero entero adicionalmente tendremos que impedir la digitación de otros caracteres que consideramos de peligrosidad para la inyección SQL.

Este saneamiento de valores ingresados se debe realizarse en el lado del cliente o navegador con Java Script si el valor se ingresa en un formulario de envió de datos así

también del lado del servidor dentro de la codificación .php, luego de recibir el valor por el método $_POST o el método $_GET.

La codificación que presentamos en el (**Anexo 4), saneamiento de valores del lado del cliente**, está realizado en java script para impedir el ingreso, la inserción, o pegado de caracteres en la ventana de autenticación del sistema de información para el ingreso de login y password, es importante aclarar que para la ejecución de este script es necesario añadir en los encabezados la librería jquery-3.6.3.min.js.

La línea de código **this.value = this.value.replace(/[^0-9a-zA-ZñÑáéíóúÁÉÍÓÚ$%?¿Ç{}./::,@* _-]/g,'');** ordena al navegador permitir la entrada de valor numéricos de 0-9, a-z,A-Z, / y caracteres ñÑáéíóúÁÉÍÓÚ$%?¿Ç{}./::,@* _- , los otros valores diferentes a esta orden serán sustituidos por un espacio en blanco.

En la siguiente sección **(Anexo 4)**, **saneamiento de valores del lado del servidor;** pasamos a describir el código en el archivo “login.php” del lado del servidor que al momento de que se reciba ya sea por el método $_POST o $_GET, este código de la misma forma que en la sección anterior quitará los caracteres que hemos considerado generen peligro para una inyección SQL.

5.3. Revisión de pruebas a la propuesta mediante OWASP Web Security Testing Guide.

Luego de haber finalizado con el desarrollo de las medidas para remediar las vulnerabilidades de la aplicación, en la siguiente sección procedemos concluir las pruebas que por segunda vez hemos realizados a la aplicación mediante el mismos procedimiento y metodología OWASP, Web Security Testing Guide (WSTG); el documento se encuentra en el **(Anexo 5)**, de nuestro trabajo al que hacemos referencia en todas las conclusiones de este capítulo.

5.3.1.1. Recopilación de Información de la propuesta.

5.3.1.1.1. Fugas de información mediante motores de búsqueda.

Para esta evaluación WSTG-INFO-01 no se ha encontrado el acceso a la aplicación web mediante el motor de búsqueda Google u otros, por el que no podemos acceder a la aplicación directamente como fue con la anterior aplicación y no existe ningún rastro para el acceso a la aplicación, hasta el momento no encontramos fugas de información y por otra parte, no está publicado en ningún buscador, solo podremos acceder al sistema de información directamente si conocemos el vínculo.

- https://juntadeaguapijal.webcindario.com/agua/admin/login.php

5.3.1.1.2. Huellas dactilares del servidor web.

WSTG-INFO-02, Como este procedimiento es de reconocimiento hemos descubierto que el servidor de nuestra aplicación es **nginx con un sistema operativo Hp P 2000 G3 NAS,** así también los puertos de servicio del servidor hosting no ha cambiado como hemos podido distinguir los puertos 443 y 80 están abiertos**.**

5.3.1.1.3. Identificar puntos de entradas de la aplicación.

En la prueba WSTG-INFO-06 podemos mencionar que la superficie de ataque son todos los puntos o mecanismos potenciales que un actor malicioso puede usar para comprometer un sistema y obtener un acceso no autorizado.

Por motivos de que nuestro estudio está centrado en la autenticación revisaremos las páginas correspondientes a la autenticación.

Tabla 5 WSTG-INFO-06 Superficie de ataque Owasp.

DIRECCION	PARAMETROS	DEFINICION
https://juntadeaguapijal.webcindario.com/agua/admin/login.php	METODO POST: VARIABLES: usuario, password, entrar, options	Autenticación
https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	METODO: GET VARIABLES:	Verificación 2FA
https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	METODO: POST VARIABLES: id_acceso, secreto, código, entrar_QR	Comprobación 2FA por QR
https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	METODO: POST VARIABLES: id_acceso, codigo_recuperacion, entrar_codigo	Comprobación 2FA por código enviado al correo
https://juntadeaguapijal.webcindario.com/agua/admin/index.php	METODO: GET	Pagina inicial del sistema de información
https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/	METODO: GET	Administración de usuarios

5.3.1.1.4. Huellas del marco de la aplicación.

La prueba WSTG-INFO-08, con esta revisión podemos verificar el marco de aplicación y tecnología utilizado para nuestra aplicación son los siguientes:

- Servidor Nginx.
- Poxy reverso Nginx
- Lenguaje de programación php.
- UI Frameworks Bootstrap.

- Miscelanea Popper.
- Librerias Java Script Jquery 3.6.3,Core Js 2.4.0, SweetAlert2 11, DataTables 1.13.7.
- Framework Java Script AMP.
- Seguridad HSTS.

5.3.1.2. Pruebas de Autenticación de la propuesta.

5.3.1.2.1. Credenciales transportadas a través de un canal cifrado.

5.3.1.2.1.1. Evaluación.

- http://juntadeaguapijal.webcindario.com/agua/admin/login.php

En la prueba WSTG-ATHN-01, la aplicación utiliza Https, puerto seguro para enviar sus datos entre la red por lo que podemos considerar que no es vulnerable.

5.3.1.2.2. Credenciales predeterminadas.

Con la prueba WSTG-ATHN-02 podemos determinar que para la creación nuevos usuario se ha sugerido una nueva política de creación de usuario JAAPI<año><un nombre>, existe política de creación de usuario en el que el sistema información controla la fortaleza de la contraseña.

5.3.1.2.3. Prueba de mecanismos de bloqueos débil.

WSTG-ATHN-03, en esta sección podemos concluir que la aplicación posee mecanismos de bloqueo en mayor o igual a cuatro intentos y mecanismo de desbloqueo del usuario en 24 horas.

5.3.1.2.4. Pruebas de omisión del esquema de autenticación.

WSTG-ATHN-04, la aplicación no es vulnerable a omisiones de esquema de autenticación, ya que el principal acceso para la aplicación es la ventana principal de acceso del usuario en donde le piden las credenciales de usuario.

5.3.1.2.5. Prueba de vulnerabilidad de recordar contraseñas.

WSTG-ATHN-05, podemos decir que la aplicación no es vulnerable a esta sección de recordar contraseñas, hemos creado una sesión en el sistema de información para verificar los objetos SessionStorage, LocalStorage , cookies del lado del cliente los que podemos concluir que no exponen datos de sesiones creadas.

La inyección automática de credenciales clickJacking y CSRF está controlada no es accesible de un sitio malicioso y los tokens fueron asegurados en términos de tiempo de vida **(Anexo 4)**.

5.3.1.2.6. Debilidades en prueba del cache del navegador.

WSTG-ATHN-06, en esta sección el sistema de información cumple con las expectativas esperadas no es vulnerable a debilidades al cache del navegador, en la revisión del encabezado de la respuesta las directivas Cache-control son revalidadas, no almacenadas y aseguradas.

5.3.1.2.7. Políticas de contraseñas débiles.

WSTG-ATHN-07, en la generación de una contraseña para esta sección ya se ha implementado políticas de filtrado o purga de datos para prevenir inyección SQL, autenticación de 2 factores y una longitud mínima de contraseñas de 12 caracteres; podemos concluir que ya existe control y una política de creación de contraseña:

- **Al menos 12 caracteres.**
- **Al menos una letra mayúscula.**
- **Al menos un número.**
- **Al menos un carácter especial.**

5.3.1.2.8. Respuestas débiles a preguntas de seguridad.

WSTG-ATHN-08, el sistema de información no tiene aplicado este tipo de seguridad basado en preguntas, pero se ha aplicado un mecanismo de 2FA por su necesidad.

5.3.1.2.9. Funcionalidades, cambio y restablecimiento de contraseña.

WSTG-ATHN-09, el sistema de información posee los mecanismos de cambio mediante autenticación por el usuario y restablecimiento de contraseñas utilizando factores 2FA o un código temporal enviado al correo electrónico.

5.3.1.2.10. Autenticación débil en un canal alternativo.

WSTG-ATHN-10, el sistema de información no posee un canal alternativo.

5.3.1.2.11. Prueba de autenticación multifactorial (MFA).

WSTG-ATHN-11, se ha fortalecido la autenticación mediante la aplicación de estos mecanismos para nuestro sistema de información. Mecanismo de autenticación 2FA por código temporal enviado al correo electrónico y autenticación 2FA por QR mediante Google Autenticator instalado en un dispositivo celular.

5.3.1.3. Pruebas de validación de entradas en la autenticación.

5.3.1.3.1. Pruebas de inyección SQL.

Con la prueba WSTG-INPV-05, no es vulnerable a este tipo de ataques, para contrarrestar estos ataques hemos implementado en el código fuente .php instrucciones y medidas que permitan sanear las entradas de datos ya sea por peticiones POST o GET, impidiendo el paso de caracteres como apostrofe comillas y otros caracteres especiales que permitirían la construcción de una consulta SQL inadecuada vulnerando al sistema de información.

5.4. Resultados de pruebas y evaluación de la propuesta.

5.4.1. Lista de verificación de pruebas (WSTG) a la autenticación.

Tabla 6 Lista de verificación de la solución.

Pruebas de autenticación.								
ID	**WSTG-ID**	**Nombre de prueba**	**Descripción**	**Herramientas**	**OWASP Top 10**	**CWE**	**Resultado**	**Sección afectado**
4.1	WSTG-ATHN-01	Prueba de credenciales transportadas a través de un canal cifrado	Evaluar si algún caso de uso del sitio web o la aplicación hace que el servidor o el cliente intercambien credenciales sin cifrado.	Burpsuite/ZAP			Aprueba	Autenticación, Creación de cuenta de usuario, Reinicio cambio o manipulación de password, Acceso a recursos mientras está conectado
4.2	WSTG-ATHN-02	Prueba de credenciales predeterminadas	Enumere las aplicaciones para las credenciales predeterminadas y valide si aún existen. Revise y evalúe las nuevas cuentas de usuario y si se crean con valores predeterminados o patrones identificables.	Browser Burpsuite/ZAP Hydra	A7	CWE-1392	Aprueba	Credenciales predeterminadas, Prueba de contraseña por defecto en una nueva cuenta
4.3	WSTG-ATHN-03	Prueba de mecanismo de bloqueo débil	Evalúe la capacidad del mecanismo de bloqueo de cuentas para mitigar la adivinación de contraseñas por fuerza bruta. Evalúe la resistencia del mecanismo de desbloqueo al desbloqueo de cuentas no autorizado.	Browser Burpsuite/ZAP Hydra	A7	CWE-307	Aprueba	Mecanismos de bloqueo, Mecanismos de desbloqueo

4.4	WSTG-ATHN-04	Prueba para omitir el esquema de autenticación	de que la autenticación se aplique en todos los servicios que la requieran.	/ZAP	A1 A7	CWE-287 CWE-302 CWE-425 CWE-200	Aprueba	Solicitud de página directa
4.5	WSTG-ATHN-05	Prueba de vulnerabilidad para recordar contraseña	Validar que la sesión generada se gestione de forma segura y no poner en peligro las credenciales del usuario.	Burpsuite/ZAP	A4	CWE-315 CWE-522 CWE-524 CWE-256	Aprueba	Inyección automática de credenciales de usuario, Tokens debe ser analizado en términos de tiempo de vida.
4.6	WSTG-ATHN-06	Prueba de debilidades de la caché del navegador	Revise si la aplicación almacena información confidencial en el lado del cliente Revisar si el acceso puede ocurrir sin autorización.	Browser Burpsuite/ZAP	A4	CWE-525	Aprueba	
4.7	WSTG-ATHN-07	Prueba de política de contraseñas débiles	Determine la resistencia de la aplicación contra la adivinación de contraseñas por fuerza bruta utilizando diccionarios de contraseñas disponibles evaluando los requisitos de longitud, complejidad, reutilización y antigüedad de las contraseñas.	Burpsuite/ZAP Hydra	A7	CWE-521 CWE-1391	Aprueba	Políticas de generación y verificación de fortaleza de contraseña
4.8	WSTG-ATHN-08	Prueba de respuesta a preguntas de seguridad débiles	Determine la complejidad y cuán sencillas son las preguntas Evalúe las posibles respuestas de los usuarios y las capacidades de fuerza bruta.	Browser Burpsuite/ZAP	A7	CWE-640	N/A	No existe mecanismos de preguntas de seguridad

4.9	WSTG-ATHN-09	Prueba de funcionalidades débiles de cambio o restablecimiento de contraseña	Determinar la resistencia de la aplicación a la subversión del proceso de cambio de cuenta permitiendo que alguien cambie la contraseña de una cuenta Determine la resistencia de la funcionalidad de restablecimiento de contraseñas contra adivinanzas o elusiones	Browser Burpsuite/ZAP	A7	CWE-620 CWE-640	Aprueba	Se ha implementado funcionalidades de restablecimiento de contraseñas
4.10	WSTG-ATHN-10	Prueba de autenticación más débil en un canal alternativo	Identificar canales de autenticación alternativos. Evaluar las medidas de seguridad utilizadas y si existen omisiones en los canales alternativos.	Browser Burpsuite/ZAP	A7	CWE-288	N/A	No existe un canal alternativo de autenticación
4.11	WSTG-ATHN-11	Prueba de autenticación Multi-Factor (MFA)	Identificar el tipo de MFA utilizado por la aplicación. Determinar si la implementación MFA es robusto y seguro. Intentar evadir MFA.	Browser Burpsuite/ZAP	A7	CWE-288 CWE-304 CWE-308	Aprueba	Autenticación 2FA
Pruebas de Validación de entrada								
7.5	WSTG-INPV-05	Prueba de inyección SQL	Identificar puntos de inyección SQL. Evaluar la severidad de inyección y el nivel de acceso al que puede ser logrado	Burpsuite/ZAP SQLMap NoSQLMap	A3	CWE-89	Aprueba	Consultas múltiples

Presentamos la tabla 6 y nuevamente hemos aplicado las pruebas al sistema de información nuevo de acuerdo a evidencias, especificamos detalladamente las pruebas aplicadas, herramientas utilizadas para probar el sistema de información y el resultado obtenido en cada una de las pruebas según hemos podido documentar en el **(Anexo 5), pruebas de la solución mediante Owasp**.

Como podemos ver en esta tabla la columna "WSTG-ID", nos indica el código de prueba especificado por Owasp para aplicar a nuestro sistema de información, y la columna "nombre de prueba" nos indica la prueba que se aplica; en las columnas siguientes nos describe las herramientas utilizadas y el resultado por cada pruebas así como valoraciones CWE, Owasp top 10 reconocido en escala de riesgo, hemos podido evidenciar que en la mayoría de las pruebas aplicadas, el sistema de información ahora si cumple con las expectativas de seguridad, pues previamente hemos aplicado las seguridades recomendadas.

Pasamos a la tabla 7 una vez valorado las vulnerabilidades describimos las condiciones en que se encuentra el sistema de información.

5.4.2. Resumen y evaluación de vulnerabilidades de la propuesta.

Tabla 7 Evaluación de vulnerabilidades, resumen de la propuesta.

No	WSTG-ID	Nombre de problema	Riesgo	Probabilidad	Impacto	Mapa OWASP Top 10	Observacion e implicacion
1	WSTG-ATHN-01	Autenticación, (Rastreos de credenciales)	Bajo	Moderado	Bajo	A2	Se ha implementado HSTS para toda la aplicación
2	WSTG-ATHN-01	Creación de cuentas (Rastreo de credenciales)	Bajo	Moderado	Bajo	A2	Se ha implementado HSTS para toda la aplicación

3	WSTG-ATHN-01	Reinicio, Cambio o Manipulación de contraseña (Rastreo de credenciales)	Bajo	Moderado	Bajo	A2	Se ha implementado HSTS para toda la aplicación
4	WSTG-ATHN-01	Acceso a recursos mientras está conectado (Token de sesión sin asegurar)	Bajo	Moderado	Bajo	A2	Token de sesión están generados aleatoriamente y asegurados
5	WSTG-ATHN-02	Prueba de credenciales por defecto	Moderado	Moderado	Moderado	A7	Credenciales por defecto están fuera de uso
6	WSTG-ATHN-02	Contraseña por defecto en una nueva cuenta	Moderado	Moderado	Moderado	A7	Se aplican políticas de aseguramiento de contraseñas
7	WSTG-ATHN-03	Mecanismos de bloqueo	Moderado	Moderado	Moderado	A7	Se ha implementado mecanismos de bloqueo
8	WSTG-ATHN-03	Mecanismos de desbloqueo	Moderado	Moderado	Moderado	A7	Se ha implementado mecanismo de desbloqueo
9	WSTG-ATHN-04	Solicitud de página directa (Acceso a información confidencial)	Bajo	Bajo	Bajo	A1 A4	Se ha aplicado controles de acceso a paginas directas
10	WSTG-ATHN-05	Inyección automática de credenciales de usuario	Bajo	Moderado	Bajo	A4	Se a aplicado restablecimiento de credenciales y uso seguro de sesiones
11	WSTG-ATHN-05	Tokens tiempo de expiración	Bajo	Moderado	Bajo	A4	Aplicamos tiempo de vida para credenciales de usuario
12	WSTG-ATHN-07	Políticas de fortalecimiento de contraseña	Bajo	Moderado	Bajo	A7	Se aplicado políticas para generación de contraseña segura
13	WSTG-ATHN-08	No existe preguntas de seguridad	Moderado	Moderado	Moderado	A7	No necesario para el sistema de información

14	WSTG-ATHN-09	Funcionalidades de reinicio o restablecimiento de contraseñas	Bajo	Moderado	Bajo	A7	Se ha aplicado funcionalidades de restablecimiento de contraseñas
15	WSTG-ATHN-10	No existe canal alternativo de autenticación	Bajo	Bajo	Bajo	A7	No necesario para el sistema de información
16	WSTG-ATHN-01	Autenticación MFA	Bajo	Moderado	Bajo	A7	Se ha implementado factores de autenticación 2FA
17	WSTG-INPV-05	Inyección Sql	Bajo	Moderado	Bajo	A3	Se ha aplicado funcionalidades de saneamiento de valores ingresados por POST y GET

La tabla 7 que mostramos es el resumen de vulnerabilidades evaluadas de la tabla 6, describe específicamente que problemas se valoraron en el sistema de información por cada una de las pruebas aplicadas anteriormente, se ha evaluado el riesgo, probabilidad e impacto de las vulnerabilidades en base a cuantificaciones descritos en el **(Anexo 2),** "**parámetros de evaluación y factores de riesgo"**; que puede surgir en contra de la autenticación y los datos, se detalla las debidas observaciones, medidas que se aplicaron para contrarrestar esas vulnerabilidades, y hemos obtenido mejores resultados.

CAPITULO 6

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Al realizar este trabajo, hemos reflexionado que con el pasar de los tiempos aparecen técnicas para asegurar la autenticación, los datos que transitan en la red y almacenan en la base de datos, así también nuevas formas o métodos de vulnerar esas seguridades; nuestro sistema de información se desarrolló hace por lo menos 10 años atrás que para esa época estuvo para el momento pero en la actualidad resultó muy vulnerable, lo que hemos podido evidenciar mediante la metodología de prueba Owasp 4.2 y posteriormente implementar las soluciones necesarias para fortalecer la autenticación mediante la aplicación de mecanismos 2FA y QR .

En la identificación de vulnerabilidades, hemos descubierto ciertamente otras deficiencias como transporte de credenciales por canales no cifrados, falta de mecanismos de restablecimiento de credenciales, sin mecanismos de bloqueo, omisiones de esquema de autenticación, credenciales sin cifrado y sin políticas, amenazas de inyección Sql, falta de aseguramiento de sesiones; las que siguen junto a la inexistencia de un factor de autenticación fuerte debilitando totalmente la seguridad en la autenticación.

Debemos concluir que estas vulnerabilidades encontradas no solo son debilidades en la autenticación como se podría mencionar, más sin embargo todo el sistema de información estuvo expuesto a un acceso no autorizado a información personal o confidencial de la JAAPI.

Para la implementación de medidas de autenticación 2FA y QR, necesariamente, tuvimos realizar un nuevo Front-End con la misma base de datos para permitir la continuidad del negocio asegurando puntos como: transporte de credenciales por canales cifrados, mecanismos de restablecimiento de credenciales y bloqueos, aplicación de esquemas de autenticación, fortalecimiento de políticas ante credenciales, hashing de credenciales, aseguramiento de sesiones y saneamiento de entradas de autenticación pues si aplicamos solo 2FA y QR por si solos, prácticamente el sistema de información sigue siendo vulnerable.

Muchas de las medidas aplicadas por motivos del aseguramiento de la autenticación permitirán asegurar la información en todo el proceso del sistema de información entre la que podemos mencionar: el transporte de datos por canales cifrados (HSTS), aseguramiento de sesiones, saneamiento de entradas para evitar inyección SQL.

Terminado la implementación de estos nuevos mecanismos de autenticación y aseguramiento 2FA y QR, procedimos por segunda ocasión a evaluar al sistema de información obteniendo mejores resultados, según la lista de verificación Owasp de la solución, que nos muestra menor riesgo, probabilidad, impacto ante las vulnerabilidades planteadas.

Al aplicar los mecanismos de doble factor y mecanismo de bloqueo de credenciales se ha fortificado el acceso para contrarrestar ataques de fuerza bruta. También la confiabilidad en el usuario que se apoya en una segunda forma de verificar su credencial, mismo que está respaldado por el uso de QR como una forma más rápida para autenticarse en el sistema de información dando un mejor impacto en la facilidad de la autenticación.

6.2. Recomendaciones.

En este trabajo hemos utilizado WSTG, diseñado por OWASP, que es un conjunto procedimientos de pruebas estructurados según el tipo de vulnerabilidades que puede tener la aplicación web, en el caso de nuestro trabajo nos enfocamos en la **autenticación** del Sistema de Información de la "Junta de Agua y Alcantarillado Pijal Alto".

Para evaluar un sistema de Información se debe utilizar una metodología adecuada en nuestro caso fue WSTG la sección autenticación y la metodología de evaluación OWASP Check List, propuesta para nuestro tipo de aplicación, existen otras metodologías que por sus características no darían ningún resultado ya que están diseñadas para otro tipo de aplicación o infraestructura.

El equipo de desarrollo no solo debe verificar que la funcionalidad del sistema de información sea acorde al modelo de negocio, también la seguridad y privacidad de esos datos que son procesados, transportados, almacenados; son muy necesarios asegurar para el tratamiento de la información en todos sus procesos.

ANEXOS.

Anexo 1 Pruebas a la aplicación con OWASP, Web Security Testing Guide.

1. Recopilación de Información.

1.1. Reconocimiento en motores de búsqueda y detectar fugas de información.

IDENTIFICACIÓN

WSTG-INFO-01

1.1.1. Objetivo de la prueba.

"Identifique qué información confidencial de diseño y configuración de la aplicación, sistema u organización está expuesta directamente (en el sitio web de la organización) o indirectamente (a través de servicios de terceros)" (OWASP, 2024d, Sección 4.1.1).

1.1.2. Procedimientos en los motores de búsqueda.

Para realizar el reconocimiento de nuestra aplicación y buscar información potencialmente confidencial se realizó una revisión inicial en los motores de búsqueda más populares con los siguientes operadores.

- `site:` limitará la búsqueda al dominio proporcionado.
- `inurl:`solo devolverá resultados que incluyan la palabra clave en la URL.
- `intitle:`solo devolverá resultados que tengan la palabra clave en el título de la página.
- `intext:`o `inbody:` solo buscará la palabra clave en el cuerpo de las páginas.
- `filetype:php:` coincidirá sólo con un tipo de archivo específico, es decir, png o php.

1.1.3. Resultados encontrados en motor de búsqueda Google.

Dirección: https://www.google.com/

Instrucción: site:comite-de-agua.webcindario.com/agua

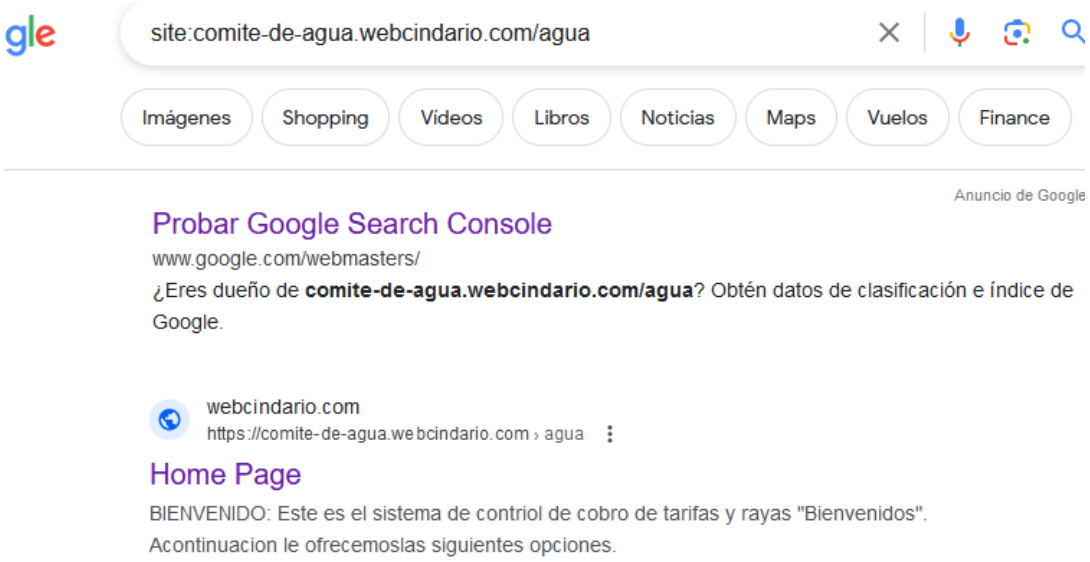


Figura 32 WSTG-INFO-01 site:comite-de-agua.webcindario.com/agua

Instrucción: inurl:comite-de-agua.webcindario.com/agua

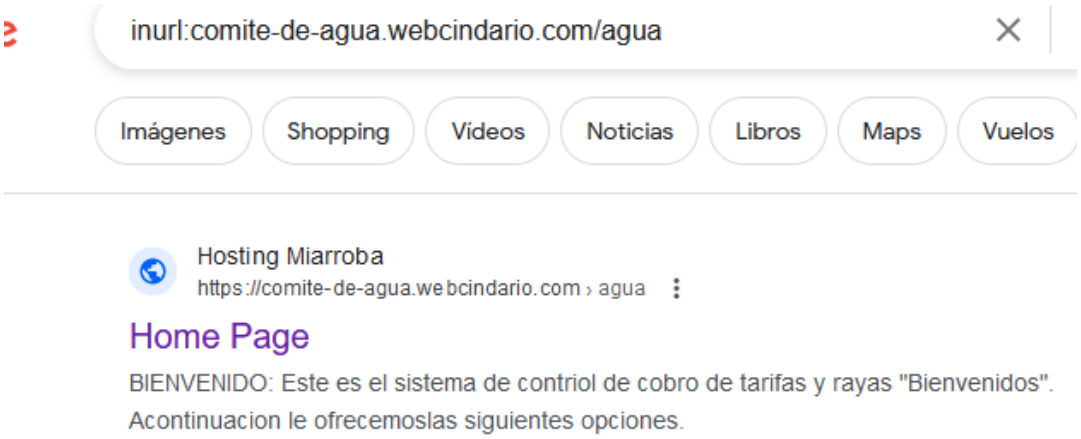


Figura 33 WSTG-INFO-01 inurl:comite-de-agua.webcindario.com/agua

Instrucción: intitle:comite-de-agua.webcindario.com/agua

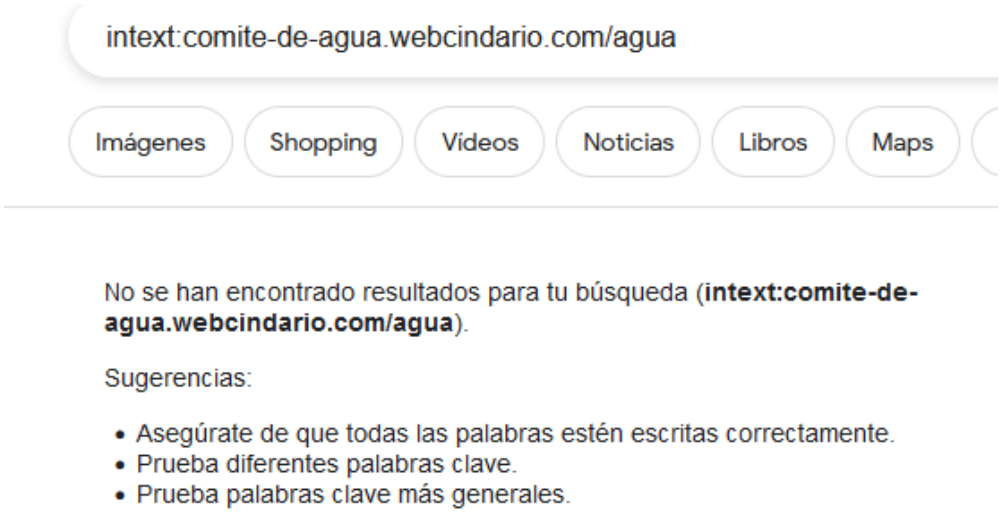


Figura 34 WSTG-INFO-01 intitle:comite-de-agua.webcindario.com/agua

Instrucción: inbody:comite-de-agua.webcindario.com/agua

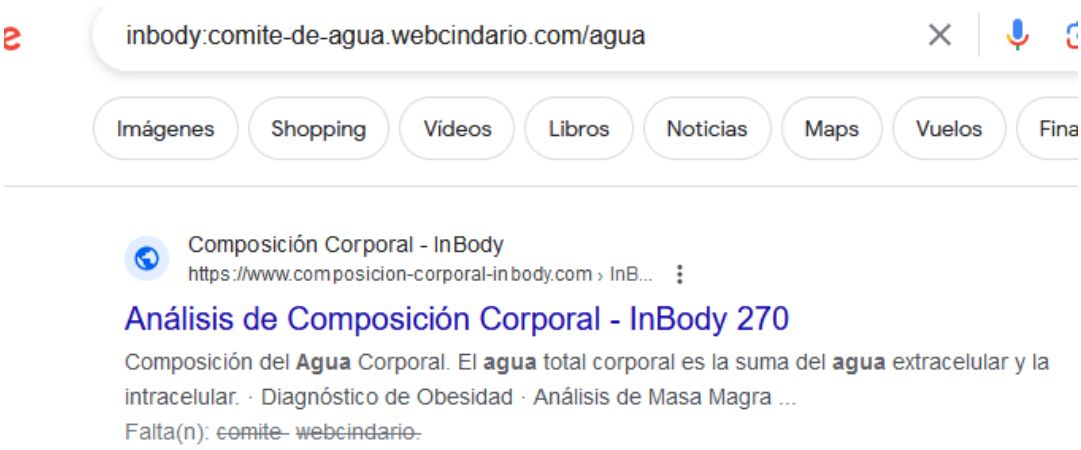


Figura 35 WSTG-INFO-01 inbody:comite-de-agua.webcindario.com/agua

***Resultado erróneo.**

Instrucción: comite-de-agua.webcindario.com/agua filetype:php

Figura 36 WSTG-INFO-01 comite-de-agua.webcindario.com/agua filetype:php

***Resultado erróneo.**

1.2. Huellas dactilares del servidor web.

IDENTIFICACIÓN
WSTG-INFO-02

1.2.1. Objetivo de la prueba.

"Determine la versión y el tipo de un servidor web en ejecución para permitir un mayor descubrimiento de cualquier vulnerabilidad conocida"(OWASP, 2024d, Sección 4.1.2).

1.2.2. Capturando banderas del servidor.

Para poder identificar nuestro servidor hemos procedido a enviar solicitudes Http en formato incorrecto para poder reconocer el servidor ya que en el servidor no se ha personalizado las páginas de error predeterminadas, utilizando una terminal de Kali linux.

1.2.3. Descubriendo el servidor a base de comandos en Kali Linux.

>telnet comite-de-agua.webcindario.com 80

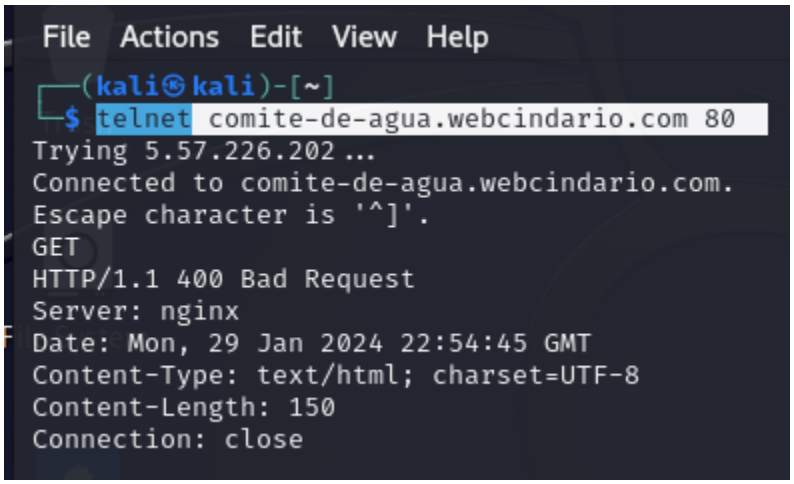


Figura 37 WSTG-INFO-02 Descubriendo el servidor a base de comandos Linux

1.2.4. Descubriendo el servidor en base a herramientas de escaneo.

En este procedimiento hemos utilizado la herramienta Nmap para descubrir al servidor web.

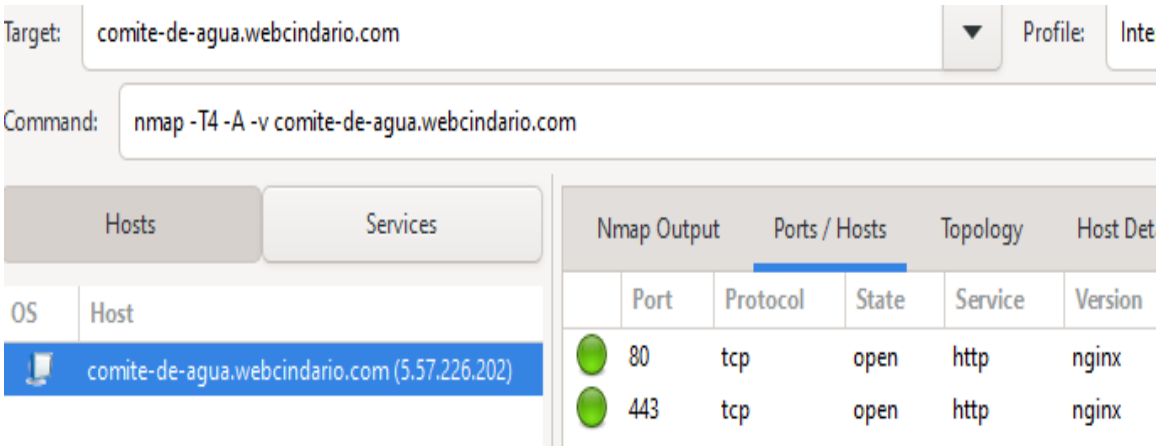


Figura 38 WSTG-INFO-02 Descubriendo puertos en servidor en base a Nmap.

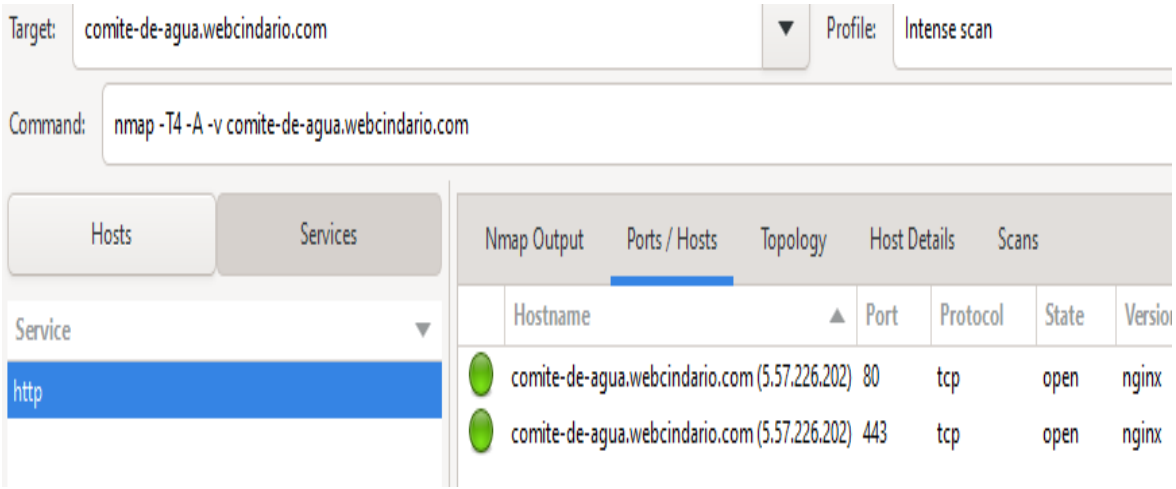


Figura 39 WSTG-INFO-02 D**escubriendo host y puertos con Nmap.**

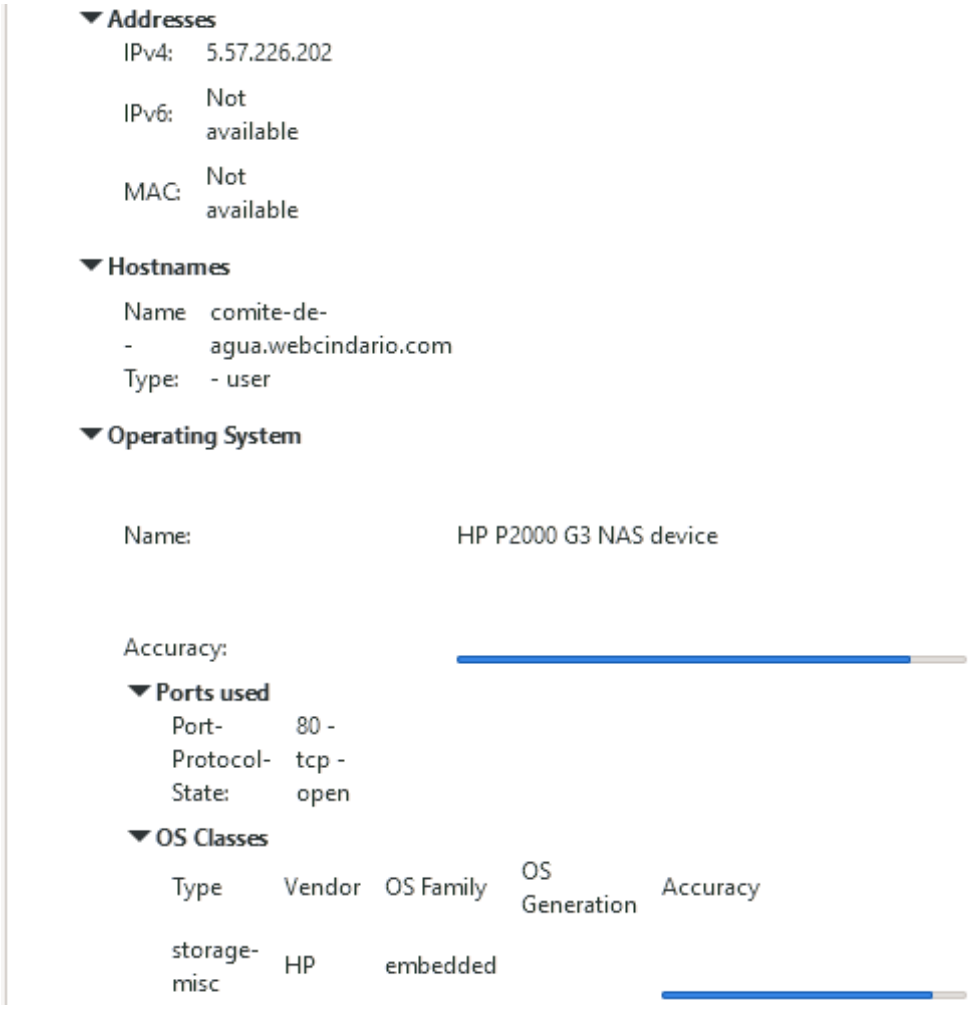


Figura 40 WSTG-INFO-02 Descubriendo ip, nombres, sistema operativo.

1.3. Identificar puntos de entradas de la aplicación.

IDENTIFICACIÓN
WSTG-INFO-06

Enumerar superficie de ataque es clave antes realizar cualquier prueba exhaustiva, ya que permite al evaluador identificar posibles debilidades y puntos a probar en el sistema de información(OWASP, 2024d).

1.3.1. Objetivo de la prueba.

"Identificar posibles puntos de entrada e inyección a través del análisis de solicitudes y respuestas" (OWASP, 2024d, Sección 4.1.6).

1.3.2. Prueba de caja Gris.

Tabla 8 WSTG-INFO-06 Puntos de entrada. Autenticación.

INGRESO DE USUARIOS AL SISTEMA	
Vinculo: https://comite-de-agua.webcindario.com/agua/inicio.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: POST** **PATH: /agua/inicio.php**
Request body parameters	**login=alex, password=2023, enviar=Enviar**
REQUEST	
POST /agua/inicio.php HTTP/2 **Host: comite-de-agua.webcindario.com** **Cookie: PHPSESSID=726dc84586976ada74942b07ec62e466;** **Content-Length: 38** **Cache-Control: max-age=0** **Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Upgrade-Insecure-Requests: 1** **Origin: https://comite-de-agua.webcindario.com** **Content-Type: application/x-www-form-urlencoded** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36**	

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://comite-de-agua.webcindario.com/agua/inicio.php
Accept-Encoding: gzip, deflate, br
Accept-Language: es-419,es;q=0.9
Priority: u=0, i
login=alex&password=2023&enviar=Enviar

RESPONSE

HTTP/2 302 Found
Server: nginx
Date: Thu, 08 Feb 2024 20:32:51 GMT
Content-Type: text/html; charset=iso-8859-1
Location: inicio.php
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Figura 41 WSTG-INFO-06 Puntos de entrada, ingreso de usuario al S.I.

Tabla 9 WSTG-INFO-06 Puntos de entrada, datos del sistema.

DATOS DEL SISTEMA	
Vinculo: https://comite-de-agua.webcindario.com/agua/datos.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/datos.php**
Request Query parameters	**doLogout=true**
REQUEST	
GET /agua/datos.php HTTP/2 **Host: comite-de-agua.webcindario.com** **Cookie: PHPSESSID=726dc84586976ada74942b07ec62e466;** **Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Upgrade-Insecure-Requests: 1** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36** **Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7** **Sec-Fetch-Site: same-origin** **Sec-Fetch-Mode: navigate** **Sec-Fetch-User: ?1** **Sec-Fetch-Dest: document** **Referer: https://comite-de-agua.webcindario.com/agua/inicio.php** **Accept-Encoding: gzip, deflate, br** **Accept-Language: es-419,es;q=0.9** **Priority: u=0, i**	
RESPONSE	

HTTP/2 200 OK
Server: nginx
Date: Fri, 09 Feb 2024 01:27:34 GMT
Content-Type: text/html; charset=iso-8859-1
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Figura 42 WSTG-INFO-06 Puntos de entrada, datos del sistema de Información

Tabla 10 WSTG-INFO-06 Puntos de entrada, usuarios de agua.

USUARIOS DE AGUA	
Vinculo: https://comite-de-agua.webcindario.com/agua/busqueda_usuario.php?ordenar=apellidos	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/busqueda_usuario.php**
Request Query parameters	**doLogout=true&ordenar=apellidos** **ordenar=apellidos** **ordenar=cedula** **ordenar=dirección** **ordenar=nombres** **ordenar=orden_lista** **pageNum_USUARIOS=1& totalRows_USUARIOS=378& ordenar=apellidos** **pageNum_USUARIOS=25& totalRows_USUARIOS=378& ordenar=apellidos**

REQUEST

GET /agua/busqueda_usuario.php?ordenar=apellidos HTTP/2
Host: comite-de-agua.webcindario.com
Cookie: PHPSESSID=726dc84586976ada74942b07ec62e466;
Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://comite-de-agua.webcindario.com/agua/index.php
Accept-Encoding: gzip, deflate, br
Accept-Language: es-419,es;q=0.9
Priority: u=0, i

RESPONSE

HTTP/2 200 OK
Server: nginx
Date: Thu, 08 Feb 2024 17:22:07 GMT
Content-Type: text/html; charset=iso-8859-1
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

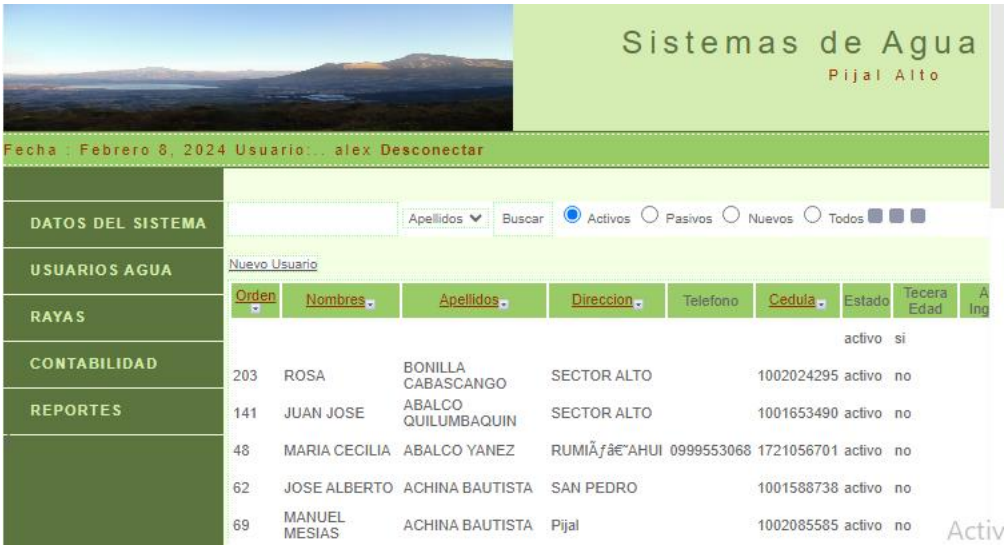


Figura 43 WSTG-INFO-06 Puntos de entrada, usuarios de agua

Tabla 11 WSTG-INFO-06 Puntos de entrada, rayas

RAYAS	
Vinculo: https://comite-de-agua.webcindario.com/agua/rayas.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/rayas.php**
Request query parameters	**ordenar=apellidos& f_ejecucion=2023-01-05** **ordenar=direccion& f_ejecucion=2023-01-05** **ordenar=nombres& f_ejecucion=2023-01-05** **pageNum_lista_usuario=0&totalRows_lista_usuario=378& todo=378** **pageNum_lista_usuario=1& totalRows_lista_usuario=378**
REQUEST	
GET /agua/rayas.php HTTP/2 **Host: comite-de-agua.webcindario.com** **Cookie: PHPSESSID=726dc84586976ada74942b07ec62e466;** **Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Upgrade-Insecure-Requests: 1** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36** **Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7** **Sec-Fetch-Site: same-origin** **Sec-Fetch-Mode: navigate** **Sec-Fetch-User: ?1** **Sec-Fetch-Dest: document** **Referer: https://comite-de-agua.webcindario.com/agua/datos.php** **Accept-Encoding: gzip, deflate, br** **Accept-Language: es-419,es;q=0.9** **Priority: u=0, i**	
RESPONSE	

HTTP/2 200 OK
Server: nginx
Date: Fri, 09 Feb 2024 03:15:57 GMT
Content-Type: text/html; charset=iso-8859-1
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Nombres	Apellidos	Sector	--	tercera edad	Yanahurco	Dia del agua Milan	Dia del agua Milan	CUMPLIDO
			-C-					0
ligia	Catucuago Gonza	Abajo de tanque Atahualpa	-C-	0	3	10	30	13
ROSA ISABEL	TUQUERRES PINCHAU	ATAHUALPA	-C-	0	3	10	30	13
MARIA JOSEFINA	ULLCO QUISHPE	ATAHUALPA	-C-	0	3	10	30	43
MARIA DIANA	CASTILLO BAUTISTA	ATAHUALPA	-C-	0	3	10	30	43
MARIA INES	CAIZALUISA	ATAHUALPA	-C-	0	3	10	30	0
JOSE ELIAS	QUILO CABASCANGO	ATAHUALPA	-C-	0	3	10	30	0
ALFREDO	BONILLA DE LA CRUZ	ATAHUALPA	-C-	0	3	10	30	0
SEGUNDO	PILATASIG	ATAHUALPA	-C-	0	3	10	30	0

Figura 44 WSTG-INFO-06 Puntos de entrada, rayas de usuarios.

Tabla 12 WSTG-INFO-06 Puntos de entrada, contabilidad.

CONTABILIDAD	
Vinculo: https://comite-de-agua.webcindario.com/agua/contabilidad.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/contabilidad.php**
Request query parameters	**doLogout=true**
REQUEST	
GET /agua/contabilidad.php HTTP/2 **Host: comite-de-agua.webcindario.com** **Cookie: PHPSESSID=726dc84586976ada74942b07ec62e466;** **Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Upgrade-Insecure-Requests: 1** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36**	

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://comite-de-agua.webcindario.com/agua/busqueda_usuario.php?ordenar=apellidos
Accept-Encoding: gzip, deflate, br
Accept-Language: es-419,es;q=0.9
Priority: u=0, i

RESPONSE

HTTP/2 200 OK
Server: nginx
Date: Fri, 09 Feb 2024 03:46:49 GMT
Content-Type: text/html; charset=iso-8859-1
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Figura 45 WSTG-INFO-06 Puntos de entrada, contabilidad.

Tabla 13 WSTG-INFO-06 Puntos de entrada, reportes.

REPORTES	
Vinculo: https://comite-de-agua.webcindario.com/agua/reportes.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/reportes.php**
Request query parameters	**doLogout=true**
REQUEST	
GET /agua/reportes.php HTTP/2 **Host: comite-de-agua.webcindario.com** **Cookie: PHPSESSID=726dc84586976ada74942b07ec62e466;** **Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Upgrade-Insecure-Requests: 1** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36** **Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7** **Sec-Fetch-Site: same-origin** **Sec-Fetch-Mode: navigate** **Sec-Fetch-User: ?1** **Sec-Fetch-Dest: document** **Referer: https://comite-de-agua.webcindario.com/agua/contabilidad.php** **Accept-Encoding: gzip, deflate, br** **Accept-Language: es-419,es;q=0.9** **Priority: u=0, i**	
RESPONSE	

HTTP/2 200 OK
Server: nginx
Date: Fri, 09 Feb 2024 03:55:54 GMT
Content-Type: text/html; charset=iso-8859-1
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Figura 46 WSTG-INFO-06 Puntos de entrada, reportes.

Tabla 14 WSTG-INFO-06 Puntos de entrada, pagos.

PAGOS	
Vinculo: https://comite-de-agua.webcindario.com/agua/pagos.php?id_usuario=203#actualizacion	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/pagos.php**
Request query parameters	**id_usuario=203**
REQUEST	
GET /agua/pagos.php?id_usuario=203 HTTP/2 **Host: comite-de-agua.webcindario.com** **Cookie: PHPSESSID=726dc84586976ada74942b07ec62e466;** **Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Upgrade-Insecure-Requests: 1**	

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

Sec-Fetch-Site: same-origin

Sec-Fetch-Mode: navigate

Sec-Fetch-User: ?1

Sec-Fetch-Dest: document

Referer: https://comite-de-agua.webcindario.com/agua/busqueda_usuario.php?ordenar=apellidos

Accept-Encoding: gzip, deflate, br

Accept-Language: es-419,es;q=0.9

Priority: u=0, i

RESPONSE

HTTP/2 200 OK

Server: nginx

Date: Fri, 09 Feb 2024 04:07:28 GMT

Content-Type: text/html; charset=iso-8859-1

Vary: Accept-Encoding

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Pragma: no-cache

X-Powered-By: Webcindario Hosting Service

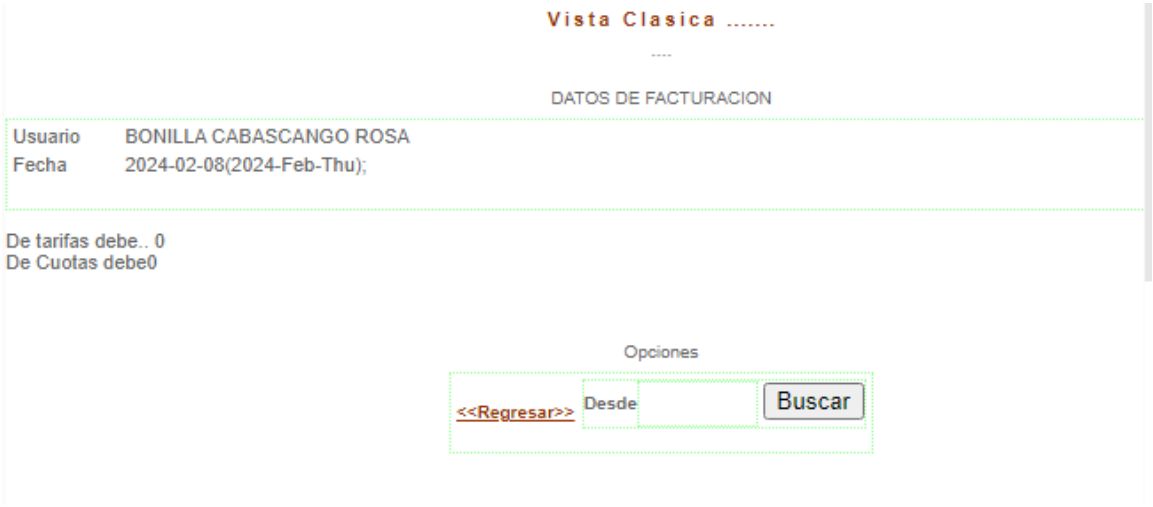


Figura 47 WSTG-INFO-06 Puntos de entrada, pagos.

Tabla 15 WSTG-INFO-06 Puntos de entrada, usuarios.

USUARIOS DEL SISTEMA	
Vinculo: https://comite-de-agua.webcindario.com/agua/usuario_acceso.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/usuario_acceso.php**
Request query parameters	**id_acceso=33**
REQUEST	
GET /agua/usuario_acceso.php?id_acceso=33 HTTP/2 **Host: comite-de-agua.webcindario.com** **Cookie: PHPSESSID=726dc84586976ada74942b07ec62e466;** **Cache-Control: max-age=0** **Upgrade-Insecure-Requests: 1** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36** **Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7** **Sec-Fetch-Site: same-origin** **Sec-Fetch-Mode: navigate** **Sec-Fetch-User: ?1** **Sec-Fetch-Dest: document** **Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Referer: https://comite-de-agua.webcindario.com/agua/modificac_usuario_acceso.php?id_acceso=33** **Accept-Encoding: gzip, deflate, br** **Accept-Language: es-419,es;q=0.9** **Priority: u=0, i**	
RESPONSE	

HTTP/2 200 OK
Server: nginx
Date: Wed, 14 Feb 2024 17:06:09 GMT
Content-Type: text/html; charset=iso-8859-1
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

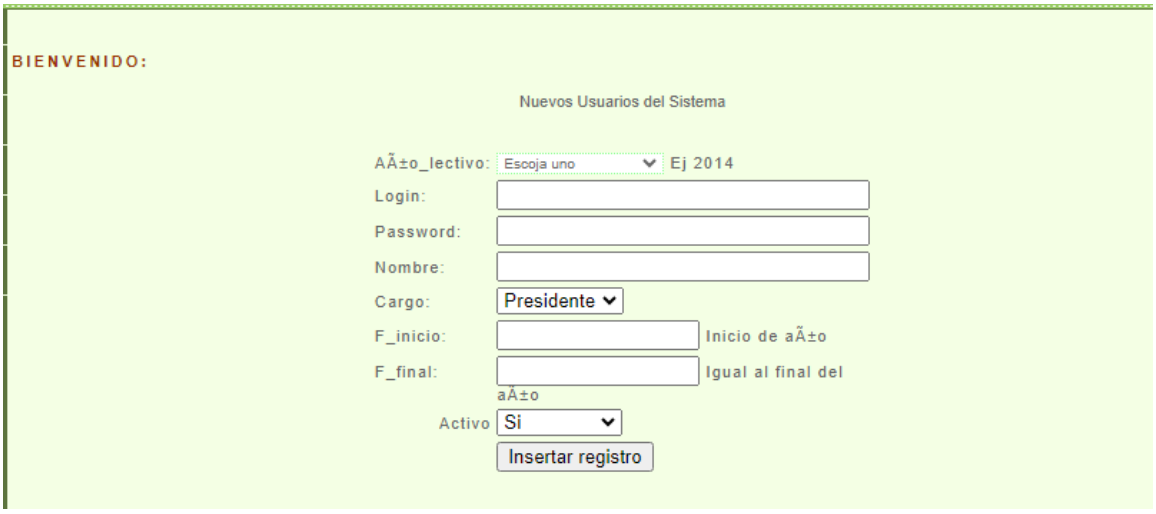


Figura 48 WSTG-INFO-06 Puntos de entrada, usuarios del sistema.

1.3.3. Detectar la superficie de ataque OWASP.

Podemos mencionar que superficie de ataque son todos los puntos o mecanismos potenciales que un actor malicioso puede usar para comprometer un sistema y obtener un acceso no autorizado.

En nuestra aplicación encontramos las siguientes superficies de ataque las que luego de una revisión concluimos que son muy vulnerables para la estabilidad del sistema de información.

Tabla 16 WSTG-INFO-06 Superficie de ataque OWASP.

DIRECCION	PARAMETROS	DEFINICION
https://comite-de-agua.webcindario.com/agua/	**METODO POST: VARIABLES: login, password, enviar**	Autenticación
https://comite-de-agua.webcindario.com/agua/busqueda_usuario.php?ordenar=apellidos	METODO: GET VARIABLES: ordenar	Ordenar usuarios.
https://comite-de-agua.webcindario.com/agua/busqueda_usuario.php	METODO: GET VARIABLES: GrupoOpciones1, campo, textfield	Búsqueda usuario
https://comite-de-agua.webcindario.com/agua/rayas.php	METODO :GET VARIABLES: **ordenar, f_ejecucion, pageNum_lista_usuario, totalRows_lista_usuario, todo.**	Ordenar lista de usuarios y paginación de usuario
https://comite-de-agua.webcindario.com/agua/usuario_acceso.php	METODO: GET	Creación de usuarios del sistema

1.4. Huellas del marco de la aplicación.

IDENTIFICACIÓN

WSTG-INFO-08

1.4.1. Objetivo de la prueba.

"Tomar huellas dactilares de los componentes que utilizan la aplicación web" (OWASP, 2024d, Sección 4.1.8).

1.4.2. Mediante Cookies.

```
POST https://comite-de-agua.webcindario.com/agua/index.php HTTP/1.1
host: comite-de-agua.webcindario.com
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Safari/537.36
pragma: no-cache
cache-control: no-cache
content-type: application/x-www-form-urlencoded
referer: https://comite-de-agua.webcindario.com/agua/
content-length: 36
Cookie: PHPSESSID=e4dae4a64ffae1332ebd1ff7d63959c5
```

Figura 49 WSTG-INFO-08 Huellas del marco de aplicación mediante cookies.

Al revisar esta imagen podemos encontrar el Cookie PHPSESSID que describe el marco de componentes en nuestro caso PHP.

1.4.3. Mediante condigo fuente HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template="/Templates/inicial.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<!-- DW6 -->
<head>
<!-- Copyright 2005 Macromedia, Inc. All rights reserved. -->
<!-- InstanceBeginEditable name="doctitle" -->
<title>Home Page</title>
<!-- InstanceEndEditable -->
```

Figura 50 WSTG-INFO-08 Huellas del marco de aplicación mediante HTML.

Podemos descubrir que para el desarrollo se ha utilizado DW6 de Macromedia.

1.4.4. Revisando archivos y carpetas específicas.

Host	Method	URL	...	Statu
https://comite-de-agua.we...	GET	/agua/		200
https://comite-de-agua.we...	GET	/agua/busqueda_usuario.php		
https://comite-de-agua.we...	GET	/agua/busqueda_usuario.php?ordenar=apellidos	✓	
https://comite-de-agua.we...	GET	/agua/contabilidad.php		
https://comite-de-agua.we...	GET	/agua/datos.php		
https://comite-de-agua.we...	GET	/agua/index.php		
https://comite-de-agua.we...	GET	/agua/index.php?doLogout=true	✓	
https://comite-de-agua.we...	GET	/agua/mm_health_nutr.css		
https://comite-de-agua.we...	GET	/agua/rayas.php		
https://comite-de-agua.we...	GET	/agua/reportes.php		

Figura 51 WSTG-INFO-08 Huellas del marco de aplicación revisando carpetas.

Podemos verificar que los archivos de la aplicación tienen extensión .php

1.4.5. Revisión de huellas del marco de aplicación mediante la herramienta Wappalizador.

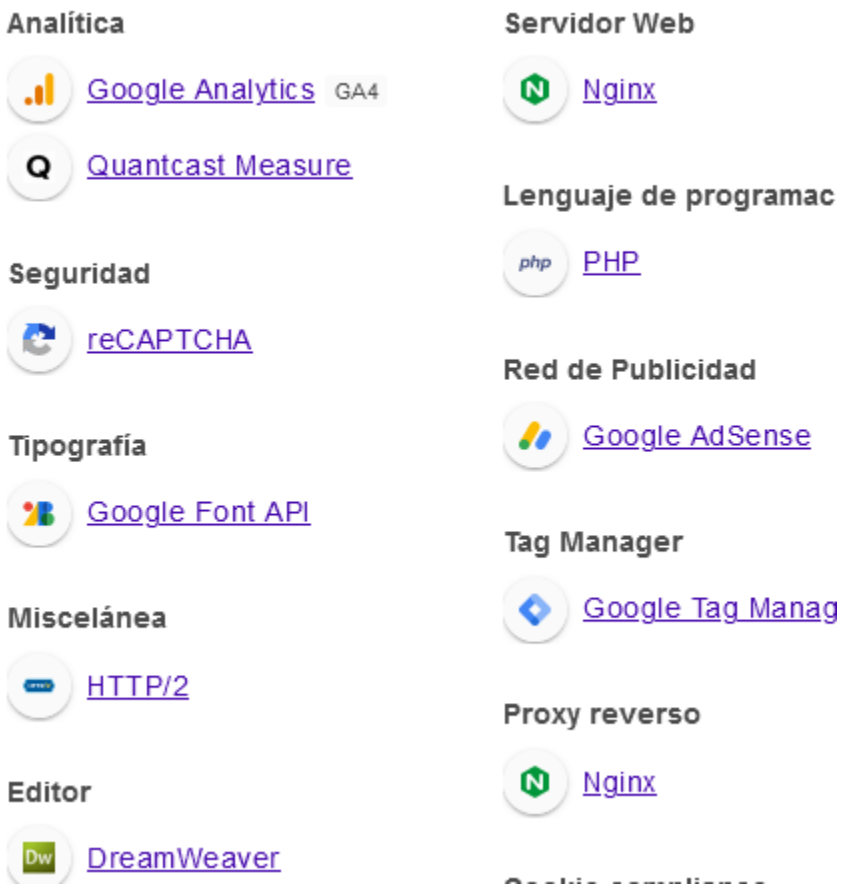


Figura 52 WSTG-INFO-06 Huellas del marco de aplicación con Wappalizador.

Podemos verificar que el servidor de la aplicación es Nginx, lenguaje de programación PHP y el editor o marco de desarrollo es Dreamweaver.

- Editor Dreamweaver.
- Servidor Nginx.
- Lenguaje de programación PHP.

1.5. Mapa de la arquitectura de la aplicación.

IDENTIFICACIÓN

WSTG-INFO-10

1.5.1. Objetivo de la prueba.

"Generar un mapa de la aplicación en cuestión basado en la investigación realizada"(OWASP, 2024d, Sección 4.1.10).

1.5.2. Mapear la arquitectura de la aplicación.

Luego de la investigación realizada podemos concluir que es una aplicación pequeña, programado en lenguaje PHP, almacenado en un hosting miarroba.com

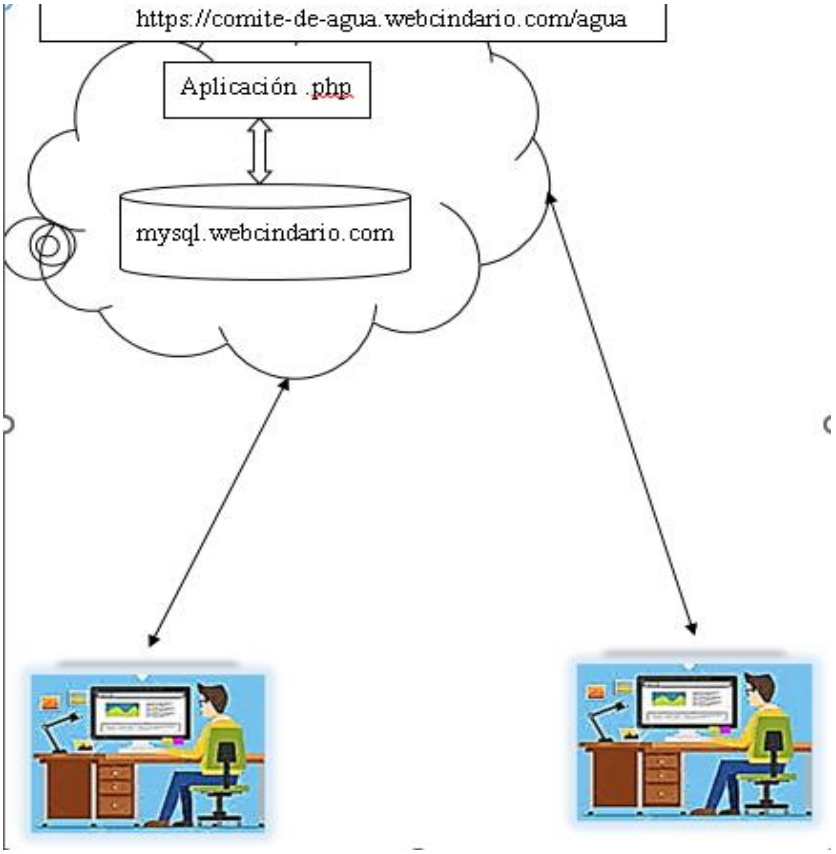


Figura 53 WSTG-INFO 10 Mapa de la arquitectura de la aplicación.

2. Pruebas de Autenticación del sistema de información.

2.1. Credenciales transportadas a través de un canal cifrado.

IDENTIFICACIÓN

WSTG-ATHN-01

Según OWASP (2024d) las aplicaciones web deben cifrar los datos de autenticación en tránsito, evita el rastreo del tráfico de red, a diferencia de Http, se utiliza Https para cifrar información de tránsito.

2.1.1. Objetivo de la prueba.

"Evaluar si algún caso de uso del sitio web o la aplicación hace que el servidor o el cliente intercambien credenciales sin cifrado"(OWASP, 2024d, Sección 4.4.1).

2.1.2. Prueba de acceso.

Intentamos realizar un inicio de sesión con una cuenta valida utilizando Http sin cifrar lo que nos da como resultado en: http://comite-de-agua.webcindario.com/agua/inicio.php

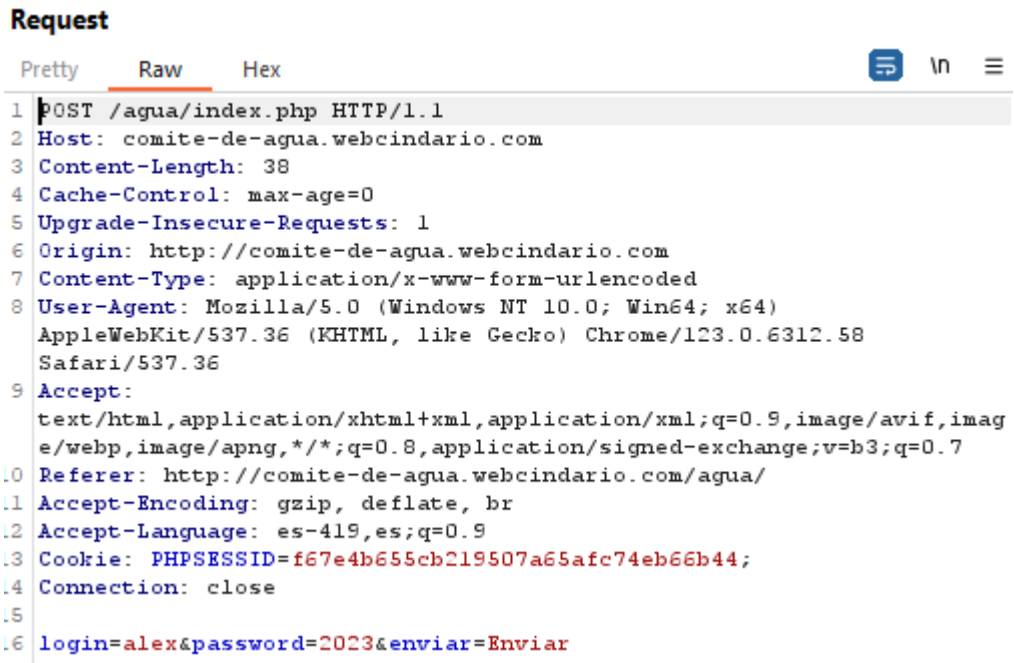


Figura 55 WSTG-ATHN-01 Request, credenciales por canales no cifrados.

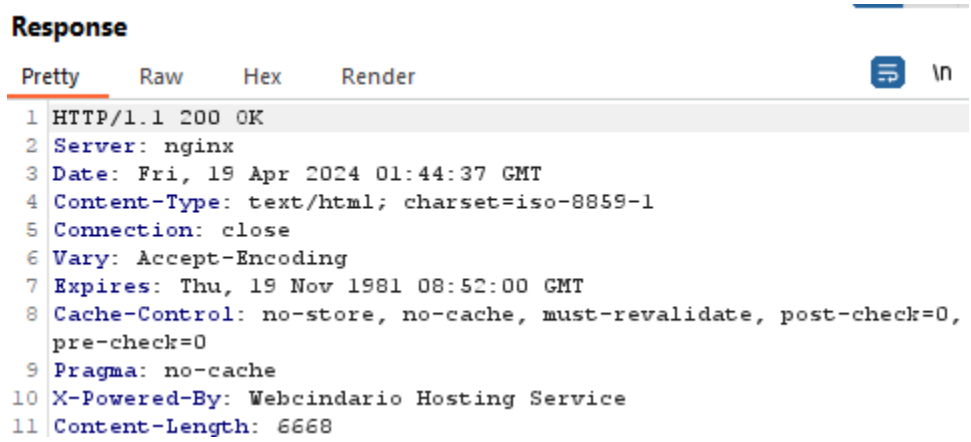


Figura 56 WSTG-ATHN-01 Response, credenciales por canales no cifrados

La imagen anterior muestra una solicitud de inicio de sesión en donde podemos visualizar las credenciales de acceso sin procesar una respuesta en donde no podemos visualizar por ninguna parte el tributo “Secure” en la imagen que presentamos a continuación capturado de la herramienta desarrollador del navegador podemos evidenciar que el atributo “Secure” de la sesión es false.

Figura 57 WSTG-ATHN-01 Atrib. secure, credenciales por canales no cifrados.

2.1.3. Creación de cuenta.

Para realizar la cuenta de un usuario sin cifrar, realizamos el procedimiento de creación de usuario con el siguiente procedimiento en:

- http://comite-de-agua.webcindario.com/agua/usuario_acceso.php

```
POST /agua/usuario_acceso.php? HTTP/1.1
Host: comite-de-agua.webcindario.com
Content-Length: 171
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://comite-de-agua.webcindario.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://comite-de-agua.webcindario.com/agua/usuario_acceso.php
Accept-Encoding: gzip, deflate, br
Accept-Language: es-419,es;q=0.9
Cookie: PHPSESSID=f67e4b655cb219507a65afc74eb66b44;
Connection: close

id_acceso=Automatico&anio_lectivo=13&login2=user456&password2=user456&
nombre=wstg-athn-01&cargo=presidente&f_inicio=2024-01-16&f_final=
2025-01-15&activo=si&MM_insert=form1
```

Figura 58 WSTG-ATHN-01 Request, creación de cuentas, credenciales por canales no cifrados.

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 302 Moved Temporarily
2  Server: nginx
3  Date: Fri, 03 May 2024 15:24:16 GMT
4  Content-Type: text/html; charset=iso-8859-1
5  Connection: close
6  Expires: Thu, 19 Nov 1981 08:52:00 GMT
7  Cache-Control: no-store, no-cache, must-revalidate,
   post-check=0, pre-check=0
8  Pragma: no-cache
9  Location: usuario_acceso.php?
10 X-Powered-By: Webcindario Hosting Service
11 Content-Length: 20309
```

Figura 59 WSTG-ATHN-01 Response, creación de cuentas, credenciales por canales no cifrados.

De la siguiente forma podemos visualizar las credenciales del nuevo usuario con esta prueba de creación de un nuevo usuario y de la misma forma no existe en ninguna parte de directiva **Set cookie = Secure.**

Para concluir de acuerdo con anteriores revisiones del servidor está activado los puertos 443,80 esto quiere decir que trabajara en cualquiera de esos puertos, pero no necesariamente cambia de Http a un puerto seguro Https pudiendo verificar el resultado de los datos inseguros así, pudiendo identificar la directiva **Upgrade-Insecur-Request,** como una petición insegura.

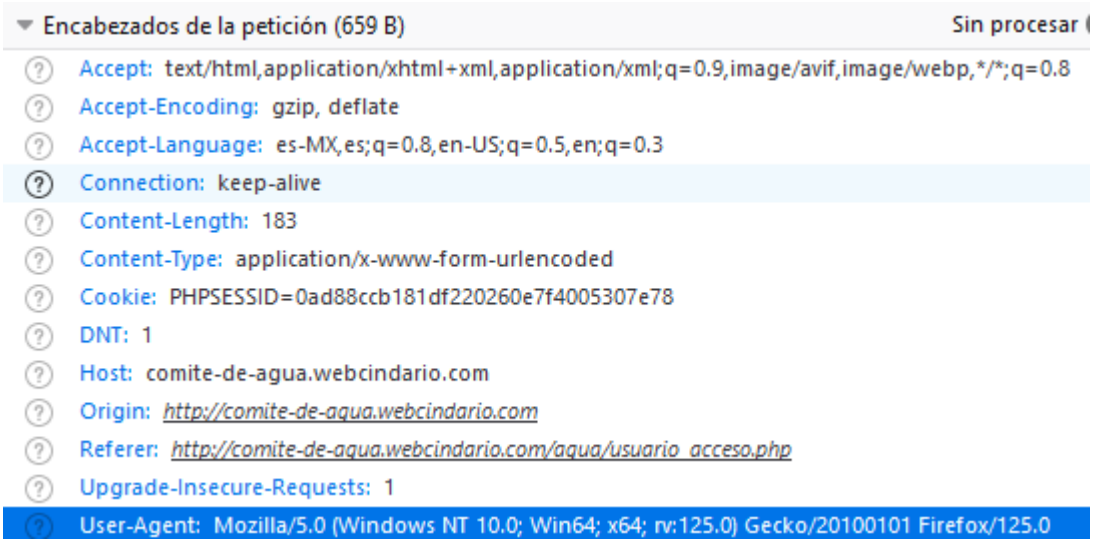


Figura 60 WSTG-ATHN-01 Petición, credenciales enviadas por de canales no cifrados.

2.1.4. Restablecimiento de contraseña, cambio o manipulación de contraseña.

Esta aplicación no posee una función de restablecimiento de contraseña, pero existe un formulario de modificación de datos del usuario incluido contraseña.

- https://comite-de-agua.webcindario.com/agua/modificac_usuario_acceso.php?id_acceso=36

Request

Pretty Raw Hex

```
1 POST /agua/modificac_usuario_acceso.php?id_acceso=36 HTTP/2
2 Host: comite-de-agua.webcindario.com
3 Cookie: _ga=GA1.3.1619521631.1707412730; __qca=P0-886665070-1707412752809; PHPSESSID=
  51f9aaedl8e650e253le41649512a2ba; _gid=GA1.3.519910366.1709403871; FCNEC=
  %5B%5B%22AKsRol-IGrLfgo2XcjzvWBwvBiRUWlbM-YlhX-k9pJ8a20vcJOcxRQ7wVvI-8bCZYQcujAMC-Zn2acUuDRIvZt
  i-yfwzUwccblMTovKGN5ZB7vaRfbY-yJZeDrGVIs_7AbHaSga07G7kKWHOKrs-F_0SiGPb8WAIBA%3D%3D%22%5D%5D
4 Content-Length: 187
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://comite-de-agua.webcindario.com
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.160 Safari/537.36
13 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.
  8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://comite-de-agua.webcindario.com/agua/modificac_usuario_acceso.php?id_acceso=36
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: es-419,es;q=0.9
21 Priority: u=0, i
22
23 anio_lectivo=13&login_acceso=prueba2024&password_acceso=prueba2024&nombre=rodrigoquilumbaqui&
  cargo=presidente&f_inicio=2024-01-16&f_final=2025-01-15&activo=si&MM_update=form1&id_acceso=36
```

Figura 61 WSTG-ATHN-01 Cambio de contraseña, credenciales por canales no cifrados

Response

Pretty Raw Hex Render

```
1 HTTP/2 302 Found
2 Server: nginx
3 Date: Sat, 02 Mar 2024 19:32:22 GMT
4 Content-Type: text/html; charset=iso-8859-1
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache,
  must-revalidate, post-check=0, pre-check=0
7 Pragma: no-cache
8 Location:
  /agua/usuario_acceso.php?id_acceso=36
9 X-Powered-By: Webcindario Hosting Service
```

Figura 62 WSTG-ATHN-01 Cambio de contraseña, respuestas en canales no cifrados.

De la misma forma podemos verificar que las credenciales de modificación del usuario son visibles e inseguras.

2.1.5. Acceder a recursos mientras está conectado.

Hemos accedido a todas las funciones de la aplicación del sitio web en protocolo Http las credenciales no se pierden ya que todas la paginas web mantienen la sesión, lo que se puede decir es que el Token de sesión no está asegurado por lo que esta prueba no cumple con las expectativas.

Figura 63 WSTG-ATHN-01 Sesiones inseguras, Credenciales por canales no cifrados

En la imagen anterior podemos ver la variable cookie, PHPSESSID que no está asegurado.

2.1.6. Remediación.

Según OWASP (2024d) al utilizar Https permite asegurar la aplicación:

- Impide a los atacantes modifiques las interacciones con el servidor.
- Facilita la escritura de determinadas aplicaciones.

Se debe priorizar HTTPS para operaciones confidenciales y a largo plazo convertir toda la aplicación a HTTPS, los navegadores detectarán como un sitio inseguro si detectan HTTP.

2.2. Credenciales predeterminadas.

IDENTIFICACIÓN

WSTG-ATHN-02

2.2.1. Objetivo de la prueba.

Según (OWASP, 2024d, Sección 4.4.2) los objetivos son:

- Enumere las aplicaciones para las credenciales predeterminadas y valide si aún existen.
- Revise y evalúe las nuevas cuentas de usuario y si se crean con valores predeterminados o patrones identificables.

2.2.2. Prueba de credenciales predeterminadas de aplicaciones comunes.

Tabla 17 WSTG-ATHN-02 Prueba de credenciales predeterminadas.

SUGERENCIA DE PRUEBA	RESULTADOS
Nombre de usuarios sugeridos: "admin", "administrador", "raíz", "sistema", "invitado", "operador", "super", "qa", "test", "test1", "testing", Contraseñas: "contraseña", "contraseña123", "contraseña123", "admin" o "invitado", contraseñas vacías y cuentas anteriores	Se encontró un usuario root
Los usuarios administrativos de la aplicación suelen recibir el nombre de la aplicación u organización.	No existen resultados
Nombres de contactos. usuarios, las direcciones de correo electrónico revelan la convención de nomenclatura de las cuentas de usuario	**Por lo general se utiliza abreviaturas de nombres y apellidos seguido del año en curso para generar usuarios**
Nombres de usuarios anteriores con contraseñas en blanco,	La debilidad de este sistema, sin autentificarse se puede acceder a nombres de usuarios del sistema a excepción de las contraseñas
Fuente de página y JavaScript en proxy o busque referencias a usuarios y contraseñas en la fuente. inicie sesión y vea cada solicitud y respuesta para un inicio de sesión, como parámetros ocultos adicionales, solicitudes GET interesantes (iniciar sesión = sí)	No existe resultado de encontrar inicia sesión=si
Nombres de cuentas y contraseñas en código fuente	No existe nombre de código y contraseñas en código fuente
Varios intentos de adivinar una contraseña con un nombre de usuario conocido pueden causar que la cuenta se bloquee	Función no habilitada en el sistema de información

Existen usuarios por defecto y contraseñas predecibles, no existen funciones de bloqueo de cuentas.

2.2.3. Prueba de contraseñas predeterminadas en cuentas nuevas.

Tabla 18 WSTG-ATHN-02 Prueba de contraseñas predeterminadas.

SUGERENCIA DE PRUEBA	RESULTADO
Política de bloque	No implementado en el sistema
Observar la página de Registro de usuarios, ayuda a determinar el formato esperado y la longitud mínima o máxima de los nombres de usuario y contraseñas de la aplicación. Si no existe una página de registro de usuario, determine si la organización utiliza una convención de nomenclatura estándar para nombres de usuario, como su dirección de correo electrónico o el nombre antes de en @el correo electrónico.	**Por lo general se utiliza abreviaturas de nombres y apellidos seguido del año en curso para generar usuarios además de utilizar la misma contraseña del usuario**
¿puede un usuario elegir su propio nombre de usuario o el sistema genera un nombre de cuenta para el usuario basándose en cierta información personal o utilizando una secuencia predecible? Si la aplicación genera los nombres de las cuentas en una secuencia predecible, como user7811, intente borrar todas las cuentas posibles de forma recursiva. Si puede identificar una respuesta diferente de la aplicación cuando usa un nombre de usuario válido y una contraseña incorrecta, entonces puede intentar un ataque de fuerza bruta en el nombre de usuario válido (o probar rápidamente cualquiera de las contraseñas comunes identificadas arriba o en la sección de referencia).	**El usuario puede generar su propio nombre de usuario siempre y cuando no exista un usuario similar registrado en el sistema de información**
Intente determinar si la contraseña generada por el sistema es predecible. Para hacer esto, cree muchas cuentas nuevas rápidamente una tras otra para poder comparar y determinar si las contraseñas son predecibles. Si es predecible, intente correlacionarlos con los nombres de usuario o cualquier cuenta enumerada y utilícelos como base para un ataque de fuerza bruta.	**Las contraseñas no son predecibles ya que el usuario de acuerdo a su criterio puede registrar su contraseña**
Si ha identificado la convención de nomenclatura correcta para el nombre de usuario, intente utilizar contraseñas de “fuerza bruta” con alguna secuencia predecible común, como por ejemplo fechas de nacimiento.	**No existe nomenclatura necesariamente para el nombre de usuario, no tiene política común de secuencia.**
Intente utilizar todos los nombres de usuario anteriores con contraseñas en blanco o utilizar el nombre de usuario también como valor de contraseña.	**Muchos usuarios pueden ingresar el mismo valor de contraseña**

2.2.4. Prueba de caja gris.

Para la prueba de caja gris hemos accedido a la entrevista de la persona encargada de TI para formula las preguntas.

Tabla 19 WSTG-ATHN-02 Prueba de caja gris, entrevista a personal de TI.

Preguntas	Respuestas
Hable con el personal de TI para determinar qué contraseñas utilizan para el acceso administrativo y cómo se lleva a cabo la administración de la aplicación.	**El acceso administrativo se realiza en base a Roles, presidente secretario, tesorero y Operador**
Pregunte al personal de TI si se cambian las contraseñas predeterminadas y si las cuentas de usuario predeterminadas están deshabilitadas.	**No existen contraseñas predeterminadas, pero nunca se cambian las contraseñas iniciales**
Examine la base de datos de usuarios en busca de credenciales predeterminadas como se describe en la sección de prueba de caja negra. También verifique si hay campos de contraseña vacíos.	**No se ha encontrado campos de contraseñas vacíos, pero se ha encontrados contraseñas muy débiles y fáciles de descifrar**
Examine el código en busca de nombres de usuario y contraseñas codificados.	**No se ha encontrado nombres de usuario y contraseña**
Busque archivos de configuración que contengan nombres de usuario y contraseñas.	**Existe un archivo de configuración para el acceso a la base de datos mysql**
Examine la política de contraseñas y, si la aplicación genera sus propias contraseñas para nuevos usuarios, verifique la política en uso para este procedimiento.	**La aplicación no genera sus propias contraseñas, no tiene política en uso**

2.3. Prueba de mecanismos de bloqueos débil.

IDENTIFICACIÓN

WSTG-ATHN-03

Para OWASP (2024d) los mecanismos de bloqueo de cuentas se utilizan para mitigar los siguientes ataques de fuerza bruta:

- Ataque para adivinar la contraseña de inicio de sesión o el nombre de usuario.
- Ataque de adivinar códigos sobre cualquier funcionalidad 2FA o preguntas de seguridad.

Los mecanismos de bloqueo de cuentas requieren un equilibrio entre proteger las cuentas del acceso no autorizado y proteger a los usuarios para que no se les niegue

el acceso autorizado. Las cuentas generalmente se bloquean después de 3 a 5 intentos fallidos y solo se pueden desbloquear después de un período de tiempo predeterminado, mediante un mecanismo de desbloqueo de autoservicio o la intervención de un administrador(OWASP, 2024d, Sección 4.4.3).

2.3.1. Objetivo de la prueba.

Según OWASP (2024d)son:

- Evalúe la capacidad del mecanismo de bloqueo de cuentas para mitigar la adivinación de contraseñas por fuerza bruta.
- Evalúe la resistencia del mecanismo de desbloqueo al desbloqueo de cuentas no autorizado.

2.3.2. Mecanismos de bloqueo.

Para probar la solidez de los mecanismos de bloqueo, necesitará acceso a una cuenta que esté dispuesto o pueda permitirse bloquear. Por lo que debido a que este sistema de información no posee un mecanismo de bloqueo no estamos en capacidad de realizar esta prueba.

2.3.3. Mecanismo de desbloqueo.

Para evaluar se debe iniciar el mecanismo de desbloqueo y buscar debilidades. Los mecanismos de desbloqueo pueden implicar preguntas secretas o un enlace de desbloqueo enviado por correo electrónico. El enlace de desbloqueo debe ser un enlace único. Otro punto se debe tomar en cuenta es que un mecanismo de desbloqueo sólo debe usarse para desbloquear cuentas. No es lo mismo recuperar la contraseña, pero podría seguir las mismas prácticas de seguridad (OWASP, 2024d).

De la misma forma que en el apartado anterior el sistema de información no posee un mecanismo de desbloqueo.

2.3.4. Remediación.

Según OWASP (2024d), se debe aplicar mecanismos de desbloqueo de cuentas, según el riesgo en orden de menor a mayor seguridad los que son:

- Bloqueo y desbloqueo basados en el tiempo.
- Desbloqueo de autoservicio (envía un correo electrónico de desbloqueo a la dirección de correo electrónico registrada).
- Desbloqueo manual del administrador.
- Desbloqueo manual del administrador con identificación positiva del usuario.

Los factores para considerar según OWASP (2024d) para implementar un mecanismo de bloqueo de cuenta son:

5. ¿Cuál es el riesgo de adivinar la contraseña por fuerza bruta contra la aplicación?
6. ¿Es suficiente un CAPTCHA para mitigar este riesgo?
7. ¿Se utiliza algún mecanismo de bloqueo del lado del cliente (por ejemplo, JavaScript)? (Si es así, deshabilite el código del lado del cliente para realizar la prueba).
8. Número de intentos fallidos de inicio de sesión antes del bloqueo. Si el umbral de bloqueo es demasiado bajo, es posible que los usuarios válidos queden bloqueados con demasiada frecuencia. Si el umbral de bloqueo es demasiado alto, más intentos podrá hacer un atacante para forzar la cuenta antes de que se bloquee. Dependiendo del propósito de la aplicación, un umbral de bloqueo típico es un rango de 5 a 10 intentos fallidos.

¿Cómo se desbloquearán las cuentas?

1. Manualmente por un administrador: este es el método de bloqueo más seguro, pero puede causar molestias a los usuarios y consumir el “valioso” tiempo del administrador.

1. Tenga en cuenta que el administrador también debe tener un método de recuperación en caso de que su cuenta se bloquee.
2. Este mecanismo de desbloqueo puede provocar un ataque de denegación de servicio si el objetivo del atacante es bloquear las cuentas de todos los usuarios de la aplicación web.

2. Después de un período de tiempo: ¿Cuál es la duración del bloqueo? ¿Es esto suficiente para proteger la aplicación? Por ejemplo, una duración de bloqueo de 5 a 30 minutos puede ser un buen equilibrio entre mitigar los ataques de fuerza bruta e incomodar a los usuarios válidos.
3. A través de un mecanismo de autoservicio: como se indicó anteriormente, este mecanismo de autoservicio debe ser lo suficientemente seguro como para evitar que el atacante pueda desbloquear las cuentas por sí mismo.

2.4. Pruebas de omisión del esquema de autenticación.

IDENTIFICACIÓN

WSTG-ATHN-04

2.4.1. Objetivo de la prueba.

"Asegurarse de que la autenticación se aplique en todos los servicios que la requieran" (OWASP, 2024a, Sección 4,4,4).

2.4.2. Prueba de caja negra.

2.4.2.1. Solicitud de página directa.

Por defecto se puede acceder a todos los sitios y páginas sin necesidad de autenticación con la única diferencia de que no se puede realizar ninguna actividad de modificación, edición, eliminación de registros en la base de datos, otras de las características de las páginas encontradas es que cada una de la página ofrece un mecanismo de autenticación que si se lo realiza permitirá privilegios de acuerdo a roles de: presidente, secretario, tesorero, cooperativa y operador.

- https://comite-de-agua.webcindario.com/agua/datos.php

Figura 64 WSTG-ATHN-04 Prueba de caja negra solicitud de página directa.

2.4.2.2. Modificación de parámetros.

Un esquema de autenticación mal diseñado verifica un inicio de sesión exitoso sobre parámetros de valor fijo. Se podría modificar estos parámetros para acceder a las áreas protegidas sin proporcionar credenciales válidas. por ejemplo, el parámetro "autenticado" se cambia a un valor de "sí", lo que permite al usuario obtener acceso(OWASP, 2024d).

Para la autenticación dentro de la aplicación los valores necesarios para tener acceso son, login y password , enviar.

Mediante el envió de una solicitud POST la aplicación verifica en la base de datos si existe el usuario.

Figura 65 WSTG-ATHN-04 Prueba de caja negra, modificación de parámetros.

Por lo que estos valores son evaluados dentro de la aplicación y no están basados sobre la base de un parámetro fijo.

2.4.2.3. Predicción de ID de sesión.

Date	Session
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	c172d71f7573594f42c9fafb8db9e999
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	553da9ebd0525903f2e305ee976b8522
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	6e0c16e43b3e5987aedd056807d21ac7
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	97a25f30618a766293a15eb7949ca50e
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	3567059b440a7732a39aa40a4abc1c17
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	19b8e0e19b413eeb9b132d811f8635c4
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	79ee0d6e7ddbb249d7e7bcf281717419
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	b1d1fe5709e0125bda9412269b003361
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	fcdc19a62c3c20830bb8e0b7e0776427
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	8c244ee98a120dbadd35fb8c3576f34c
Creación:"Mon, 04 Mar 2024 23:34:07 GMT"	32fa50b7a4cbef861c2645d92fc8c96e

Figura 66 WSTG-ATHN-04 Predicción de ID de sesión.

Hemos identificado en esta aplicación que la autenticación se gestiona mediante identificadores de sesiones ID, pero en la identificación de patrones para la generación no son parecidas por lo que no podemos establecer un patrón específico para poder establecer o adivinar fácilmente una sesión.

2.4.2.4. Inyección SQL(autenticación de formulario HTML).

Para esta prueba realizamos la verificación con la herramienta ZAP la que nos ha mostrado posibles debilidades dentro de nuestra aplicación.

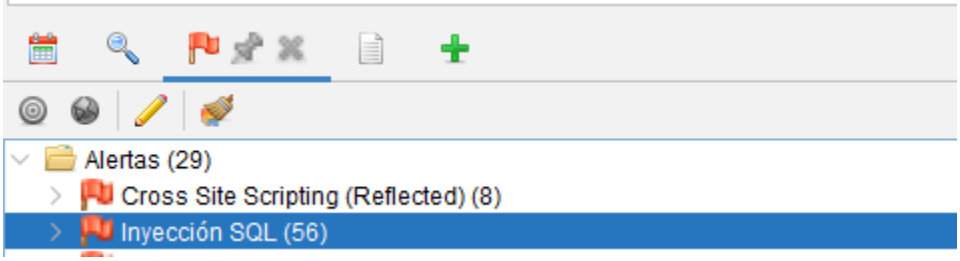


Figura 67 WSTG-ATHN-04 Inyección Sql, vulnerabilidades con ZAP

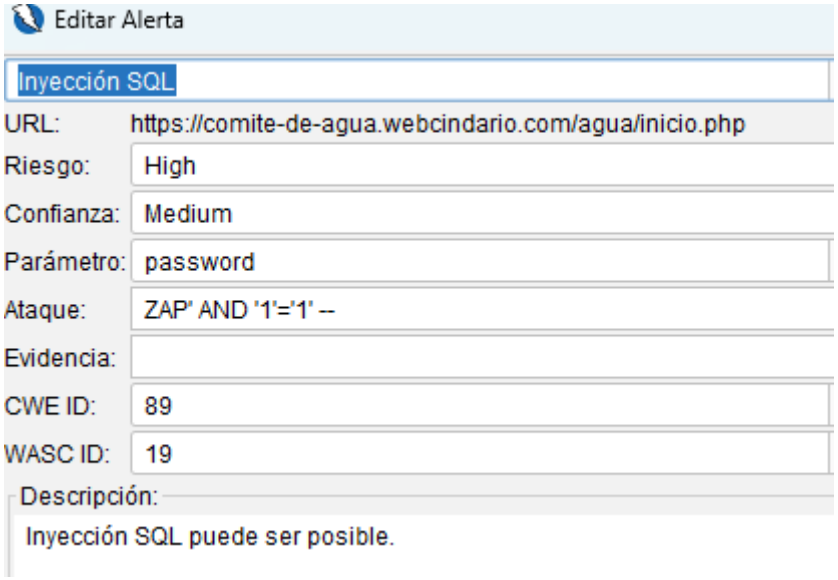


Figura 68 WSTG-ATHN-04 Inyección Sql, atributo de vulnerabilidad en ZAP .

En nuestro caso vamos a probar la vulnerabilidad de acceso por autenticación a la página “inicio.php” lo que nos da el siguiente resultado.

Figura 69 WSTG-ATHN-04 Inyección Sql, Request autenticación por formulario HTML.

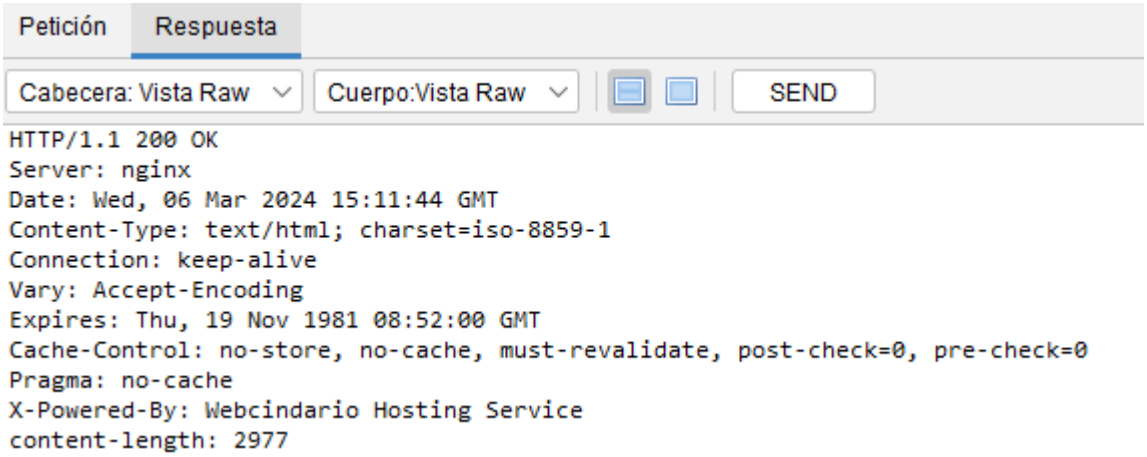


Figura 70 WSTG-ATHN-04 Inyección Sql, Response autenticación por formulario HTML.

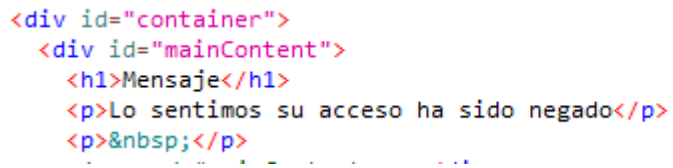


Figura 71 WSTG-ATHN-04 Inyección Sql, Resultado de autenticación.

Verificando los resultados podemos concluir que los datos han sido enviados y existe un resultado de página (200 OK) pero no necesariamente es un acceso, sino solo una respuesta de acceso negado ya que la inyección SQL no ha funcionado.

2.4.3. Pruebas de caja gris.

Figura 72 WSTG-ATHN-04 Prueba de caja gris, revisión de sentencia Sql.

Como hemos tenido acceso al código fuente, verificamos a fin de establecer una forma vulnerar el acceso por autenticación.

```php
if (isset($_POST['login'])) {
  $loginUsername=$_POST['login'];
  $password=$_POST['password'];
  $MM_fldUserAuthorization = "cargo";
  $MM_redirectLoginSuccess = "inicio.php";
  $MM_redirectLoginFailed = "denegado.php";
  $MM_redirecttoReferrer = false;
  mysql_select_db($database_acceso, $acceso);

  $LoginRS__query=sprintf("SELECT id_acceso, login, password, cargo FROM usuario_acceso WHERE login=%s AND password=%s"
   echo $LoginRS__query;    /* imprime la consulta Sql  */
   echo "<br>";
  $LoginRS = mysql_query($LoginRS__query, $acceso) or die(mysql_error());
  $loginFoundUser = mysql_num_rows($LoginRS);
  echo $loginFoundUser;  /* imprime valor si existe el usuario con 1  o 0 si no existe  */

  if ($loginFoundUser) {    /*  verificacion si existe el usuario¨*/

    $loginStrGroup  = mysql_result($LoginRS,0,'cargo');
   $loginacceso  = mysql_result($LoginRS,0,'id_acceso');

    //declare two session variables and assign them
  $_SESSION['id_acceso'] = $loginacceso;
    $_SESSION['MM_Username'] = $loginUsername;
    $_SESSION['MM_UserGroup'] = $loginStrGroup;

    if (isset($_SESSION['PrevUrl']) && false) {
      $MM_redirectLoginSuccess = $_SESSION['PrevUrl'];
    }
   header("Location: " . $MM_redirectLoginSuccess );
```

Figura 73 WSTG-ATHN-04 Prueba de caja gris revisión de código fuente

Si se podría modificar la variable $loginfoundUser=1 el acceso estaría permitido, así también existen 3 valores de sesión que son asignados luego de recuperarse de la base de datos y las que son:

- id_acceso, registro en la base de datos.

- MM_Username, login en la base de datos.
- MM_UserGroup, el cargo en este caso presidente o secretario, o tesorero etc.

2.5. Prueba de vulnerabilidad de recordar contraseñas.

IDENTIFICACIÓN.

WSTG-ATHN-05

Según OWASP (2024d) debido a tan amplio uso de las credenciales, login y password , han surgido múltiples tecnologías.

- Las aplicaciones proporcionan una función de recordar contraseña, sin volver a pedir contraseñas.
- Administradores de contraseñas, incluidos los administradores de contraseñas del navegador, que permiten al usuario almacenar sus credenciales de forma segura y luego inyectarlas en formularios de usuario sin ninguna intervención del usuario.

2.5.1. Objetivo de la prueba

"Validar que la sesión generada se gestione de forma segura y no poner en peligro las credenciales del usuario"(OWASP, 2024d, Sección 4.4.5).

2.5.2. Almacenar credenciales.

Procedemos a verificar credenciales almacenadas en mecanismos del navegador y como se menciona en el manual de las credenciales no deben ser almacenadas en la aplicación del lado del cliente y deben ser sustituidas por tokens generados del lado del servidor.

Procedemos a ingresar con el usuario y password a fin de crear una sesión.

- https://comite-de-agua.webcindario.com/agua/inicio.php

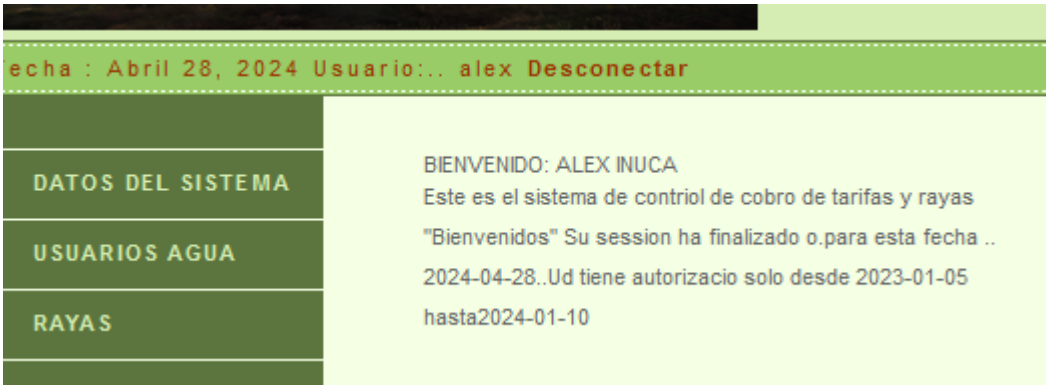


Figura 74 WSTG-ATHN-05 Almacenar credenciales.

Procedemos a probar SessionStorage y LocalStorage así también la cookie del usuario.

- **https://comite-de-agua.webcindario.com/agua/WSTG-ATHN-05.php**

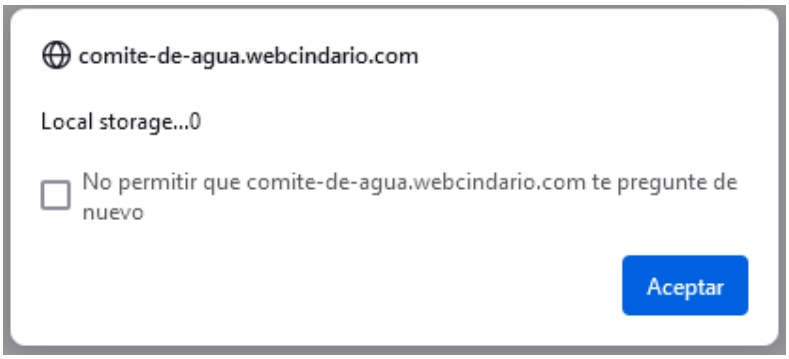


Figura 75 WSTG-ATHN-05 Almacenar credenciales, Local Storage

Figura 76 WSTG-ATHN-05 Almacenar credenciales, sesión storage.

Figura 77 WSTG-ATHN-05 Almacenar credenciales, cookies

Con esta prueba podemos verificar que no existe ningún valor para Sessionstorage y Localstorage objetos de almacenamiento API de sesione a diferencia de Cookies de sesión que queda almacenado en el lado del cliente a pesar de que realizamos el procedimiento de cierre de sesión la cookie queda vigente en el lado del cliente.

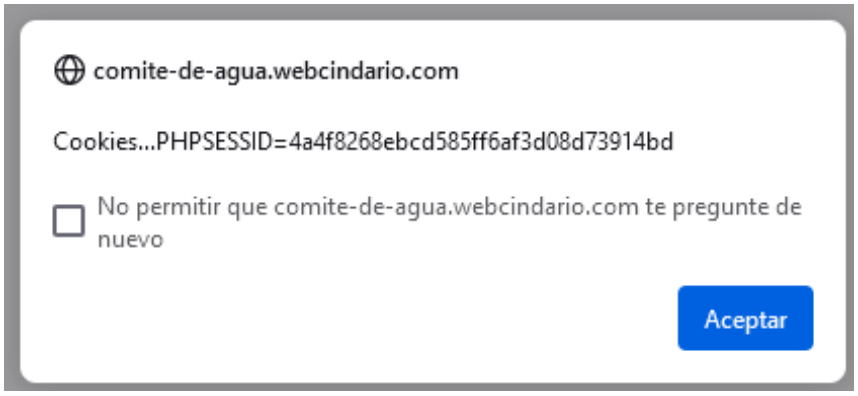


Figura 78 WSTG-ATHN-05 Almacenar credenciales. cookies del lado cliente.

El código fuente utilizado realizar esta prueba es el siguiente:

```
Prueba WSTG-ATHN-05 y WSTG-CLNT-12


<script>
alert("Local storage..."+localStorage.length);
for (let i = 0; i < localStorage.length; i++) {
  const key = localStorage.key(i);
  const value = localStorage.getItem(key);
  console.log(`${key}: ${value}`);
```

```
  alert("Key.."+key+" Value.. ."+value);
}
alert("Session storage..."+sessionStorage.length);
for (let i = 0; i < sessionStorage.length; i++) {
  const key = sessionStorage.key(i);
  const value = sessionStorage.getItem(key);
  console.log(`${key}: ${value}`);
  alert("Key.."+key+" Value.. ."+value);
}
alert("Cookies..."+window.document.cookie);
console.log(window.document.cookie);
alert("List All Entries on the Window Object");
(() => {

  // create an iframe and append to body to load a clean window
object
  const iframe = document.createElement('iframe');
  iframe.style.display = 'none';
  document.body.appendChild(iframe);
  // get the current list of properties on window
  const currentWindow = Object.getOwnPropertyNames(window);
  // filter the list against the properties that exist in the
clean window
  const results = currentWindow.filter(
    prop => !iframe.contentWindow.hasOwnProperty(prop)
  );
  // remove iframe
  document.body.removeChild(iframe);
  // log key-value entries that are different
  results.forEach(key => console.log(`${key}: ${window[key]}`));
})();
const dumpIndexedDB = dbName => {
  const DB_VERSION = 1;
  const req = indexedDB.open(dbName, DB_VERSION);
  req.onsuccess = function() {
    const db = req.result;
    const objectStoreNames = db.objectStoreNames || [];
    console.log(`[*] Database: ${dbName}`);
    Array.from(objectStoreNames).forEach(storeName => {
      const txn = db.transaction(storeName, 'readonly');
```

```
      const objectStore = txn.objectStore(storeName);
      console.log(`\t[+] ObjectStore: ${storeName}`);
      // Print all entries in objectStore with name `storeName`
      objectStore.getAll().onsuccess = event => {
        const items = event.target.result || [];
        items.forEach(item => console.log(`\t\t[-] `, item));
      };
    });
  };
};
indexedDB.databases().then(dbs => dbs.forEach(db =>
dumpIndexedDB(db.name)));
</script>
```

2.5.3. Inyección automática de credenciales.

La inyección automática depende del navegador cuando autocompleta el formulario del login y password si la aplicación es vulnerable al ataque de ClickJacking y CSRF, en nuestro caso la aplicación es vulnerable.

- https://sitio-malicioso.webcindario.com/Clickjacking.php

Figura 79 WSTG-ATHN-05 Inyección de credenciales, Clickjacking.

El sitio es accesible desde otro sitio malicioso resulta vulnerable.

2.5.4. Tokens debe ser analizado en términos de tiempo de vida.

Los tokens no son considerados en términos de vida útil en esta aplicación.

2.6. Debilidades en prueba del cache del navegador.

IDENTIFICACIÓN

WSTG-ATHN-06

2.6.1. Objetivo de la prueba.

Según OWASP (2024d)los objetivos son:

- Revisar si la aplicación almacena información confidencial en el lado del cliente.
- Revisar si el acceso puede ocurrir sin autorización.

2.6.2. Historial del navegador.

Según el botón ATRAZ del navegador es un historial y no un cache, lo que consiste en esta prueba es ingresar o verificar información sensible, cerrar la sesión y volver atrás para ver si se puede acceder a esta información confidencial

- https://comite-de-agua.webcindario.com/agua/usuario_acceso.php

Con acceso privilegiado hemos procedido a crear un usuario en el cual se puede ver la contraseña.

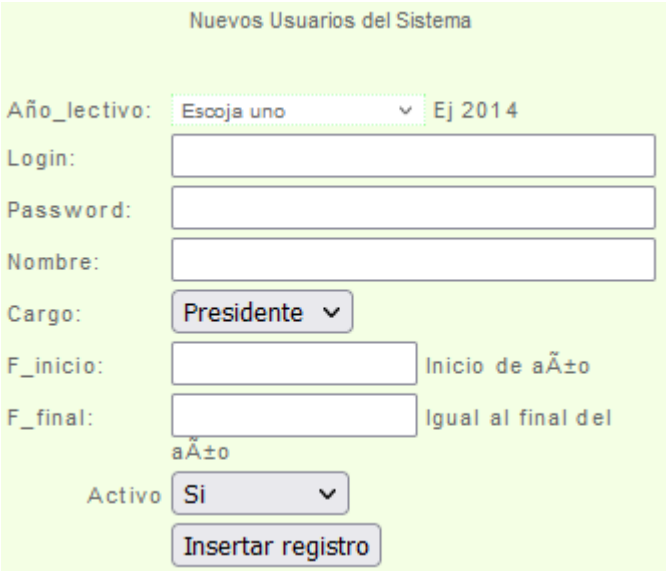
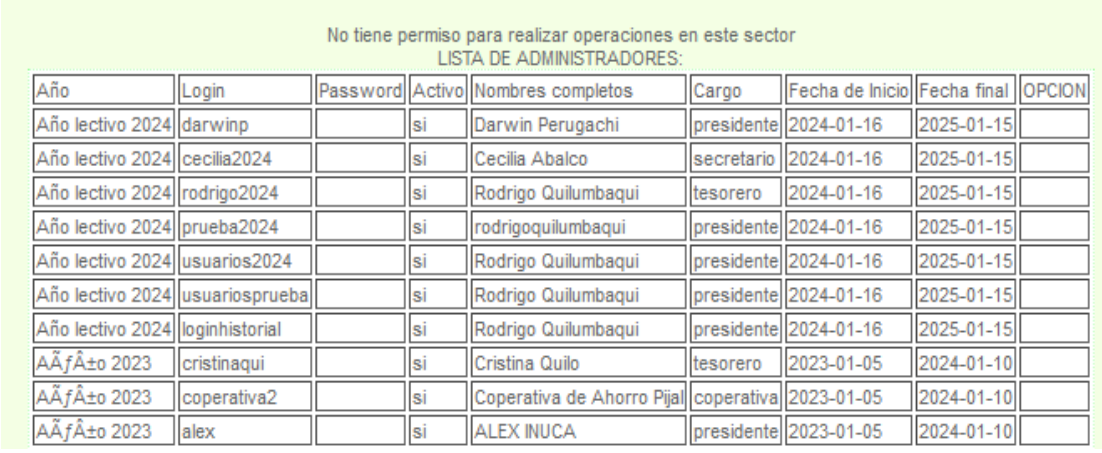


Figura 80 WSTG-ATHN-06 Creación de usuario, historial de navegador.

Figura 81 WSTG-ATHN-06 Creación de usuario, historial de navegador, usuario creado.

Cerramos la sesión y procedemos a verificar nuevamente esta información presionando los botones del navegador **atrás** hasta regresar al ala misma página de creación de usuarios**.**



Año	Login	Password	Activo	Nombres completos	Cargo	Fecha de Inicio	Fecha final	OPCION
Año lectivo 2024	darwinp		si	Darwin Perugachi	presidente	2024-01-16	2025-01-15	
Año lectivo 2024	cecilia2024		si	Cecilia Abalco	secretario	2024-01-16	2025-01-15	
Año lectivo 2024	rodrigo2024		si	Rodrigo Quilumbaqui	tesorero	2024-01-16	2025-01-15	
Año lectivo 2024	prueba2024		si	rodrigoquilumbaqui	presidente	2024-01-16	2025-01-15	
Año lectivo 2024	usuarios2024		si	Rodrigo Quilumbaqui	presidente	2024-01-16	2025-01-15	
Año lectivo 2024	usuariosprueba		si	Rodrigo Quilumbaqui	presidente	2024-01-16	2025-01-15	
Año lectivo 2024	loginhistorial		si	Rodrigo Quilumbaqui	presidente	2024-01-16	2025-01-15	
AÃƒÂ±o 2023	cristinaqui		si	Cristina Quilo	tesorero	2023-01-05	2024-01-10	
AÃƒÂ±o 2023	coperativa2		si	Coperativa de Ahorro Pijal	coperativa	2023-01-05	2024-01-10	
AÃƒÂ±o 2023	alex		si	ALEX INUCA	presidente	2023-01-05	2024-01-10	

Figura 82 WSTG-ATHN-06 Creación de usuario, historial de navegador, <atrás.

Al parecer hemos podido verificar que la sesión de usuarios se ha cerrado y la contraseña de los usuarios no se puede visualizar por lo que el usuario no está autentificado,

dando a relucir que el problema no es de autenticación sino en el diseño del sistema de información que permite la revisión de toda la información con privilegios de lectura.

- Se entrega la página mediante Https.
- Configuración Cache-Control: must-revalidate.

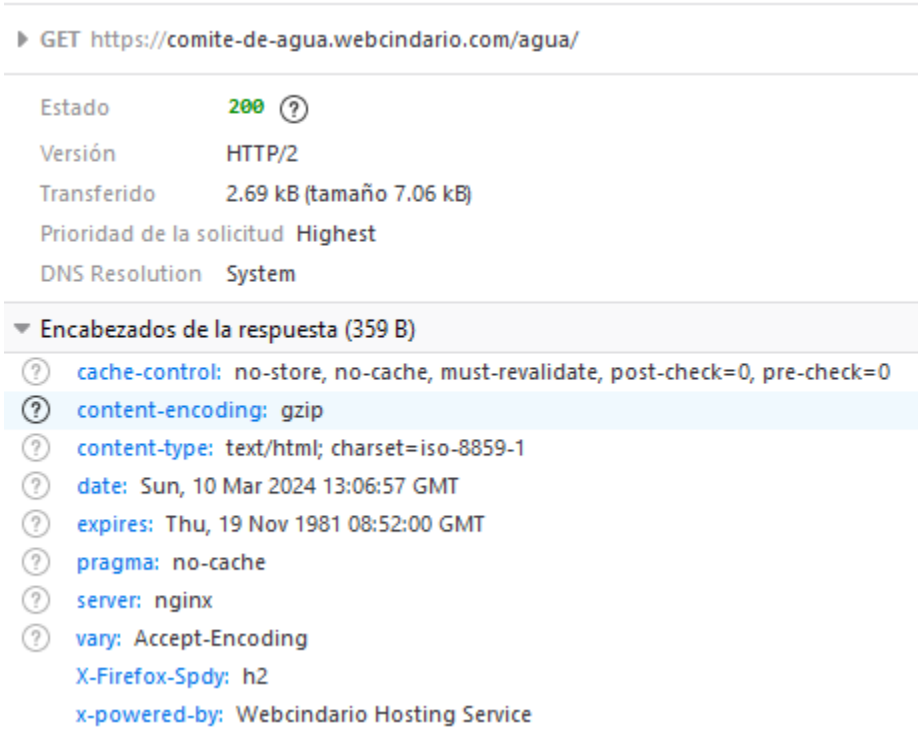


Figura 83 WSTG-ATHN-06 Historial de navegador, cache.

Como podemos ver en la imagen anterior la directiva **cache-control=must-revalidate,** existe**.**

2.6.3. Cache del navegador.

Se debe verificar que la aplicación no filtre ningún dato confidencial en el caché del navegador. Para hacer eso, pueden usar un proxy y buscar en las respuestas del servidor que pertenecen a la sesión, verificando para la página que contenga información confidencial el servidor le haya indicado al navegador que no almacene

en caché ningún dato. Dicha directiva se puede emitir en los encabezados de respuesta HTTP con las siguientes directivas (OWASP, 2024d, Sección 4.4.6).

- Cache-Control: no-cache, no-store
- Expires: 0
- Pragma: no-cache
- Cache-Control: must-revalidate, max-age=0, s-maxage=0

Según hemos revisado las directivas para nuestra aplicación los resultados podemos comprobar en su mayoría que si cumple con las expectativas.

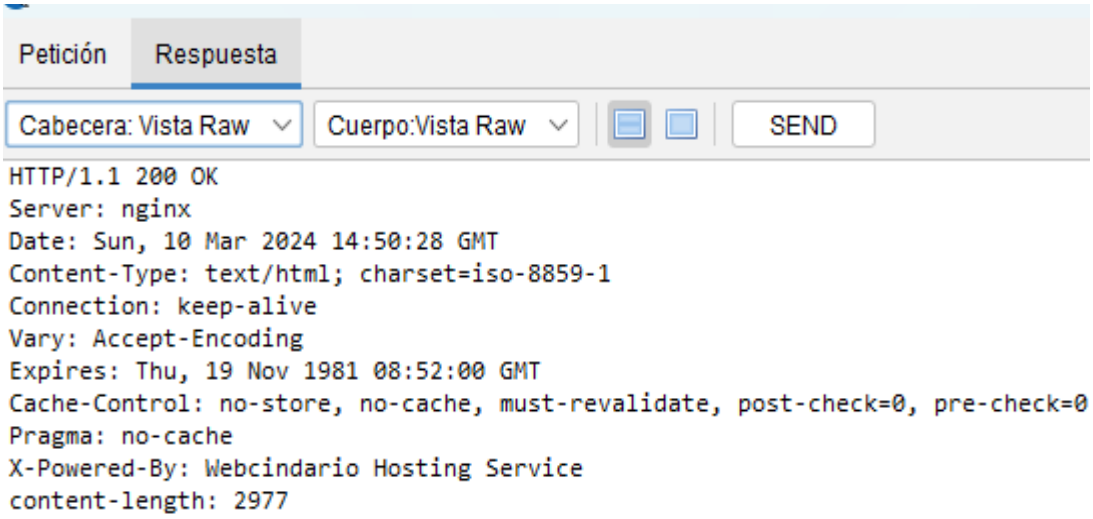


Figura 84 WSTG-ATHN-06 Cache del navegador, no cache, no store.

2.6.4. Revisar información almacenada en cache.

Para este paso hemos tenido que revisar la información almacenada en cache del navegador, y nos da como resultado que es un documento seguro.

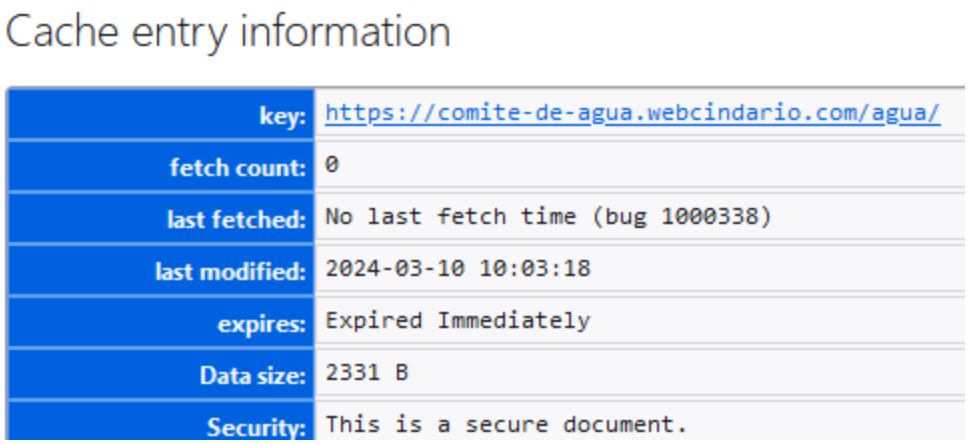


Figura 85 WSTG-ATHN-06 Información almacenada en cache.

2.6.5. Verificación de manejo para navegadores móviles.

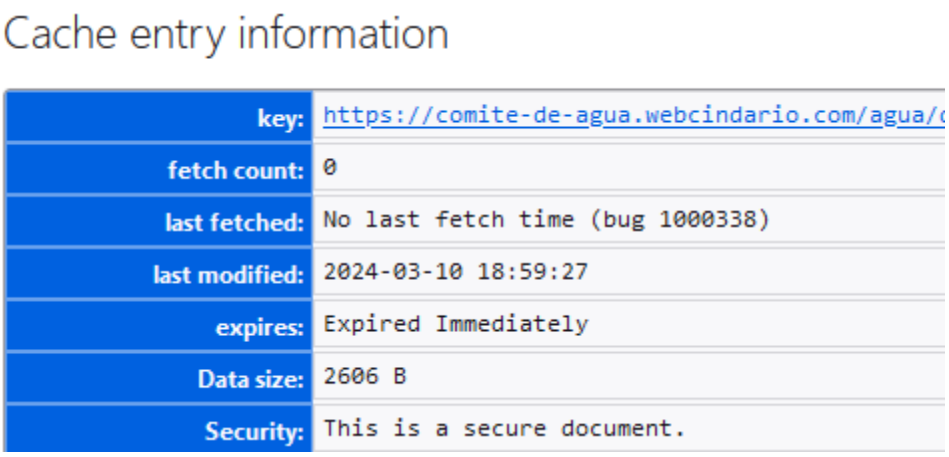


Figura 86 WSTG-ATHN-06 Información almacenada en cache móviles.

2.6.6. Prueba de caja gris.

Para esta prueba hemos procedido a tener acceso como un usuario con privilegios en este caso el usuario que tiene privilegios de presidente revisaremos el cache del navegador e historial de navegador obteniendo el mismo resultado de los procedimientos anteriores.

2.7. Políticas de contraseñas débiles.

IDENTIFICACIÓN

WSTG-ATHN-07

El mecanismo de autenticación más frecuente y más fácil de administrar es una contraseña estática. La contraseña representa las llaves del dominio, pero los usuarios a menudo lo sacrifican en nombre de la usabilidad. En cada uno de los recientes ataques de alto perfil que han revelado las credenciales de los usuarios, se lamenta que las contraseñas más comunes sigan siendo 123456: password y qwerty(OWASP, 2024d, Sección 4.4.7).

2.7.1. Objetivo de la prueba.

Determine la resistencia de la aplicación contra la adivinación de contraseñas por fuerza bruta utilizando diccionarios de contraseñas disponibles evaluando los requisitos de longitud, complejidad, reutilización y antigüedad de las contraseñas(OWASP, 2024d, Sección 4.4.7).

2.7.2. Prueba con preguntas.

Tabla 20 WSTG-ATHN-07 Preguntas , políticas y contraseñas débiles.

PREGUNTAS	RESPUESTAS
¿Qué caracteres están permitidos y prohibidos en una contraseña?	**NINGUNA**
¿Se requiere que el usuario utilice caracteres de diferentes conjuntos de caracteres, como letras minúsculas y mayúsculas, dígitos y símbolos especiales?	**NINGUNA**
¿Con qué frecuencia puede un usuario cambiar su contraseña?	**NINGUNA**
¿Qué tan rápido puede un usuario cambiar su contraseña después de un cambio anterior?	**NINGUNA**
Los usuarios pueden eludir los requisitos del historial de contraseñas cambiando su contraseña 5 veces seguidas para que después del último cambio de contraseña hayan configurado nuevamente su contraseña inicial.	**NINGUNA**
¿Cuándo debe un usuario cambiar su contraseña?	**NINGUNA**

○ Tanto NIST como NCSC recomiendan **no** forzar la caducidad regular de la contraseña, aunque puede ser requerido por estándares como PCI DSS.	
¿Con qué frecuencia un usuario puede reutilizar una contraseña?	**NINGUNA**
¿La aplicación mantiene un historial de las 8 contraseñas utilizadas anteriormente por el usuario?	**NINGUNA**
¿Qué tan diferente debe ser la siguiente contraseña de la última contraseña?	**NINGUNA**
¿Se impide al usuario utilizar su nombre de usuario u otra información de la cuenta (como el nombre o apellido) en la contraseña?	**NINGUNO**
¿Cuáles son las longitudes mínima y máxima de contraseña que se pueden establecer?	**0-300**
¿Son apropiadas para la sensibilidad de la cuenta y la aplicación?	**NO**
¿Es posible establecer contraseñas comunes como Password1o 123456?	**SI**

Al responder estas preguntas podemos concluir que no existe ningún control ni política para establecer contraseñas por lo que fue recomendable establecer políticas de implementación de contraseñas para el sistema de información.

2.7.3. Remediación.

Las contraseñas fáciles de adivinar facilitan el acceso no autorizado, existen dos soluciones: colocar autenticación de dos factores o introducir una política de contraseñas segura. El más fácil y económico es introducir una política de contraseñas segura que garantice longitud, complejidad, reutilización y antigüedad de la contraseña; lo ideal sería implementar ambos (OWASP, 2024d).

2.8. Respuestas débiles a preguntas de seguridad.

IDENTIFICACIÓN

WSTG-ATHN-08

“Las preguntas y respuestas de seguridad, a menudo denominadas preguntas y respuestas “secretas”, se utilizan a menudo para recuperar contraseñas olvidadas, o como seguridad adicional además de la contraseña” (OWASP, 2024d, Sección 4.4.8).

2.8.1. Objetivo de la prueba.

Los objetivos según OWASP (2024d) son:

- Determine la complejidad y cuán sencillas son las preguntas.
- Evalúe las posibles respuestas de los usuarios y las capacidades de fuerza bruta.

2.8.2. Prueba de preguntas débiles pre-generadas.

No se ha podido establecer ninguna evaluación de vulnerabilidad ya que el sistema de información solo maneja la autenticación mediante un usuario y contraseña, por lo que se podría determinar que sería necesario establecer preguntas de seguridad adicional para seguridad en la recuperación de una cuenta de autenticación.

2.8.3. Prueba de preguntas débiles autogeneradas.

De la misma forma no se puede obtener una evaluación de vulnerabilidad ya que en este sistema de información solo se autentica mediante un usuario y contraseña y no existe otro factor de verificación para recuperación de contraseña.

2.8.4. Pruebas para respuestas de fuerza-bruta.

De la misma forma no se puede obtener una evaluación de vulnerabilidad ya que en este sistema de información solo se autentica mediante un usuario y contraseña y no existe otro factor de verificación para recuperación de contraseña.

2.9. Probar funcionalidades de cambio y restablecimiento de contraseña.

IDENTIFICACIÓN

WSTG-ATHN-09

La función de cambio y restablecimiento de contraseña permite a los usuarios cambiar o restablecer rápidamente su contraseña sin el administrador. Cuando se cambian las contraseñas, normalmente se cambian en la aplicación. si se restablecen las contraseñas, se

muestran dentro de la aplicación o se envían por correo electrónico al usuario. Esto indica que las contraseñas están almacenadas en texto sin formato o en un formato descifrable (OWASP, 2024d).

2.9.1. Objetivo de la prueba.

- Determinar la resistencia de la aplicación a la subversión del proceso de cambio de cuenta permitiendo que alguien cambie la contraseña de una cuenta.
- Determine la resistencia de la funcionalidad de restablecimiento de contraseñas contra adivinanzas o elusiones.

2.9.2. Comprobación previa.

Según OWASP (2024d) es necesario preguntar lo siguiente para el cambio o el restablecimiento de contraseña.

Tabla 21 WSTG-ATHN-09 Comprobación previa.

Pregunta	Respuesta
Si los usuarios, distintos de los administradores, pueden cambiar o restablecer contraseñas de cuentas distintas a las suyas.	No se puede comprobar
si los usuarios pueden manipular o alterar el proceso de cambio o restablecimiento de contraseña para cambiar o restablecer la contraseña de otro usuario o administrador.	No se puede comprobar
si el proceso de cambio o restablecimiento de contraseña es vulnerable a CSRF **(**falsificación de solicitudes entre sitios**)**	No se puede comprobar

2.9.3. Prueba de restablecimiento de contraseña.

Según OWASP (2024d) además de las comprobaciones anteriores se debe dar respuestas a las siguientes preguntas adicionales:

¿Qué información se requiere para restablecer la contraseña?

¿Cómo se comunica al usuario el restablecimiento de contraseñas?

¿Las contraseñas de restablecimiento se generan aleatoriamente?

¿La función de restablecimiento de contraseña solicita confirmación antes de cambiar la contraseña?

***Como en nuestro sistema de información no posee funcionalidades de cambio o restablecimiento de contraseña no se ha procedido a la evaluación de esa funcionalidad.**

2.9.4. Prueba de cambio de contraseña.

Según OWASP (2024d) menciona que se debe solicitar la siguiente pregunta

¿Se solicita la contraseña anterior para completar el cambio?

2.9.5. Remediación.

Según la función de cambio o restablecimiento de contraseña es una función sensible y requiere alguna forma de protección, como exigir a los usuarios que se vuelvan a autenticar o presentar pantallas de confirmación durante el proceso (OWASP, 2024d).

2.10. Autenticación débil en un canal alternativo.

IDENTIFICACIÓN
WSTG-ATHN-10

Si no se existen vulnerabilidades de los mecanismos principales, podría existir vulnerabilidades de canales alternativos, pudiendo eludir el canal principal o exponer información que puede ayudar a un ataque contra el canal principal (OWASP, 2024d).

2.10.1. Objetivo de la prueba.

Según OWASP (2024b)los objetivos de la prueba son:

- Identificar canales de autenticación alternativos.
- Evaluar las medidas de seguridad utilizadas y si existen omisiones en los canales alternativos.

2.10.2. Comprensión del mecanismo Primario.

2.10.2.1. Creación de cuenta de usuario.

- https://comite-de-agua.webcindario.com/agua/usuario_acceso.php

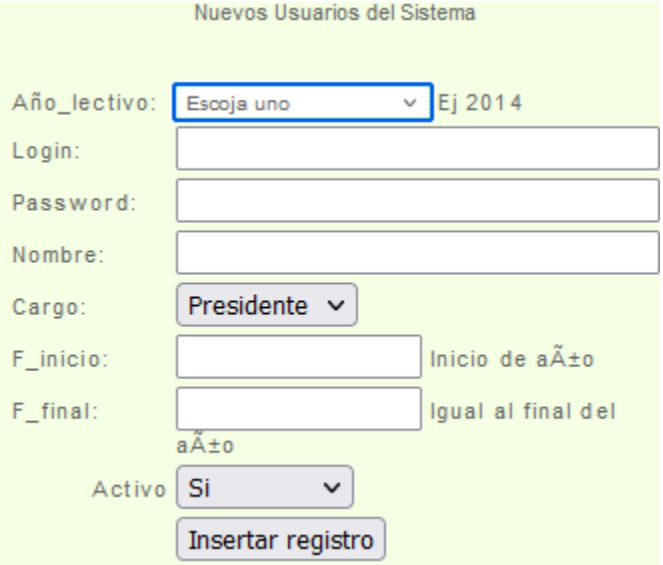


Figura 87 WSTG-ATHN-10 Nuevo usuario, comprensión mecanismo primario.

Una vez que se ha autentificado en el sistema de información con privilegios de **presidente** podemos acceder a este cuadro de dialogo para la creación de un nuevo usuario con los siguientes datos.

- Año lectivo, periodo económico en el que ejerce su actividad dentro del sistema de información.
- Login.
- Password.
- Nombre.
- Cargo(Rol), pudiendo adquirir el cargo de Presidente , Secretario, Tesorero, Coperativa, Operador.
- Fecha inicio y fecha final del periodo.
- Asctivo, **SI** el usuario está activo, **bloqueado,** si el usuario está bloqueado.

Año lectivo 2024	prueba2024	prueba2024	si	rodrigoquilumbaqui	presidente	2024-01-16	2025-01-15	ModificarEliminar
Año lectivo 2024	usuarios2024	usuario2024	si	Rodrigo Quilumbaqui	presidente	2024-01-16	2025-01-15	ModificarEliminar
Año lectivo 2024	usuariosprueba	usuario2024	si	Rodrigo Quilumbaqui	presidente	2024-01-16	2025-01-15	ModificarEliminar

Figura 88 WSTG-ATHN-10 Comprensión del mecanismo primario, usuario creado

2.10.2.2. Modificar cuentas de usuario.

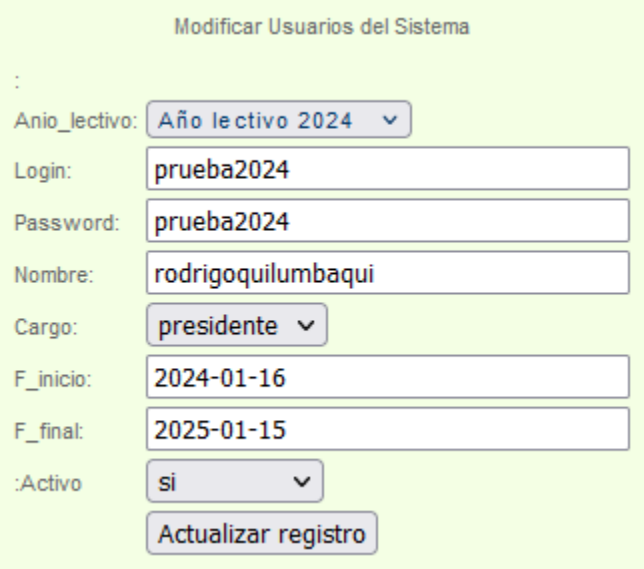


Figura 89 WSTG-ATHN-10 Comprensión del mecanismo primario, modificar cuentas de usuario.

Al ingresar al registro correspondiente al usuario se encuentra un vínculo de **modificar** que permite ingresar a una pantalla básica de modificación de usuario.

2.10.2.3. Eliminar cuentas de usuario.

En caso de que el registro de la cuenta en la base de datos se encuentre comprometido con otro registro por integridad referencial será imposible eliminar el usuario,

2.10.2.4. Identificar otros canales.

No existen canales alternativos para manipular registro de identificación de usuarios. Solo existe el canal principal.

2.10.2.5. Enumerar la funcionalidad de autenticación.

- Registro.
- Acceso.
- Modificación de datos del usuario.
- Eliminación.

2.10.2.6. Revisar y probar.

No existen canales alternativos que probar.

2.10.3. Remediación.

Según OWASP (2024d) se debe asegurarse de que se aplique una política de autenticación coherente en todos los canales para que sean igualmente seguros

2.11. Prueba de autenticación multifactorial (MFA).

ID

WSTG-ATHN-11

La autenticación multifactor (MFA) es una capa adicional de seguridad para proteger el proceso de autenticación. Conocido como autenticación (2FA) o verificación de dos pasos (2SV) (OWASP, 2024d).

2.11.1. Objetivo de la prueba.

Los objetivos según OWASP (2024d)son:

- Identifique el tipo de MFA utilizado por la aplicación.
- Determinar si la implementación de MFA es sólida y segura.
- Intente eludir el MFA.

2.11.2. Tipos de MFA utilizados.

Según OWASP (2024d) MFA significa que se requiere dos de los siguientes factores para la autenticación.

- **Algo que sabes.**
- **Algo que tienes.**
- **Algo que eres.**
- **Ubicación.**

2.11.3. Verificación derivaciones autenticación MFA.

Según OWASP (2024d) el primer paso es identificar todas las funciones de autenticación en la aplicación.

- La página principal de inicio de sesión.
- Funcionalidad crítica para la seguridad (como deshabilitar MFA o cambiar una contraseña).
- Proveedores de inicio de sesión federados.
- Puntos finales API (tanto desde la interfaz web principal como desde las aplicaciones móviles).
- Protocolos alternativos (no HTTP).
- Funcionalidad de prueba o depuración.

Para el caso de nuestro trabajo hemos podido acceder al método de autenticación por lógin y password, no existe ningún método o función adicional de autenticación.

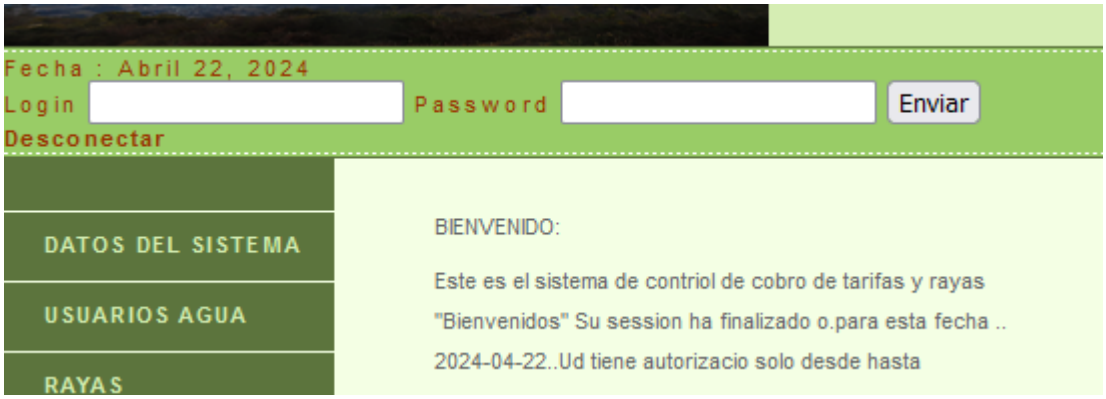


Figura 90 WSTG-ATHN-11 Verificación derivaciones MFA, no existe método.

2.11.4. Prueba de gestión MFA.

Según OWASP (2024d) las funcionalidades utilizadas para administrar MFA se debe probarse desde dentro de la cuenta para detectar vulnerabilidades.

- ¿Se requiere que el usuario se vuelva a autenticar para eliminar o cambiar la configuración de MFA?
- ¿La funcionalidad de gestión de MFA es vulnerable a la falsificación de solicitudes entre sitios ?
- ¿Se puede modificar la configuración de MFA de otros usuarios a través de las vulnerabilidades de IDOR ?

***No se ha podido evaluar.**

3. Pruebas de validación de entradas en la autenticación.

3.1. Pruebas de inyección SQl.

IDENTIFICACIÓN

WSTG-INPV-05

Según OWASP (2024d) un ataque de inyección SQL consiste en la inserción o "inyección" de una consulta SQL parcial o completa a través de la entrada de datos o transmitida desde el cliente (navegador) a la aplicación web.

3.1.1. Objetivo de la prueba.

Según OWASP (2024d) los objetivos son:

- Identificar puntos de inyección SQL.
- Valorar la gravedad de la inyección y el nivel de acceso que se puede conseguir mediante la misma.

3.1.2. Pruebas para Mysql.

Según OWASP (2024d) las vulnerabilidades de inyección SQL ocurren siempre que se utilizan entradas en la construcción de una consulta SQL sin estar adecuadamente restringidas o desinfectadas.

3.1.3. Comillas simples en la aplicación.

Para poder manipular dentro de nuestra aplicación es muy importante conocer como insertar las comillas simples y también concatenar la consulta o inyección Sql, dentro de un valor o variable GET o POST en nuestra aplicación.

3.1.4. Inyección Sql dentro de una petición POST.

Para inyectar Sql dentro de un valor enviado por el método POST se lo realiza de la siguiente manera.

' OR 1='1 => ' or 1='1'

Figura 91 WSTG-INPV-05 Inyección Sql dentro de una petición POST.

Tenemos que aclarar que el valor ' significa apostrofe para encerrar los valores de las variables login y password en nuestra aplicación como podemos ver en la sentencia SQl inyectada en la imagen anterior.

3.1.5. Inyección Sql dentro de una petición GET.

Dentro de una consulta mediante el método GET la inyección Sql se lo realiza de la siguiente forma.

+OR+1+=+1 => ' or 1='1'

http://localhost/agua/busqueda_usuario.php?ordenar=apellidos+OR+1+=+1

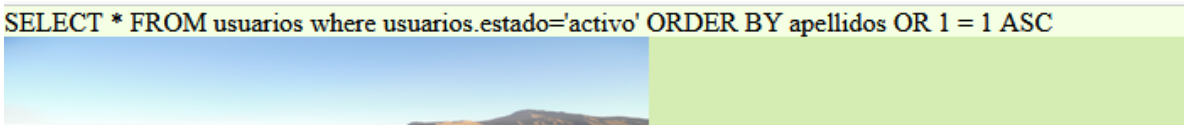


Figura 92 WSTG-INPV-05 Consulta Sql por inyección GET-

Como se puede ver la consulta lógicamente es incorrecta, pero se ejecuta.

Orden	Nombres	Apellidos	Direccion	Telefono	Cedula	Estado	Tecera Edad	Anio Ingreso	Opciones	Aportes
0	mYFuyLyV	CgYxZHlw	UFVJSJGD	rdHurnVJ	tgCnoVXQ	activo	si	0000-00-00		Pagos Llaves Rayas
12	DAVID	ULCUANGO	ATAHUALPA		1706255906	activo	no			Pagos Llaves Rayas
15	SEGUNDO ALBERTO	IMBAQUINGO ACERO	ATAHUALPA		1714955521	activo	no			Pagos Llaves Rayas
13	DIGNA LUCIA	CASCO CHICAIZA	Atahualpa		1713907291	activo	no			Pagos Llaves Rayas
13	OLGA	REINOSO TOCAGON	ATAHUALPA		1002008165	activo	no			Pagos Llaves Rayas

Figura 93 WSTG-INPV-05 Resultado de inyección por el método GET.

3.1.6. Huellas dactilares del servidor MySQL.

Empezamos la prueba en el siguiente vinculo.

- https://comite-de-agua.webcindario.com/agua/inicio.php

Para descubrir la versión del servidor vamos a añadir la siguiente sentencia a la consulta Sql de inicio de sesión: `1 AND 1=0 UNION SELECT @@version.`

Ingresamos en password la siguiente inyección sql:

```
1&#39; OR 1=1 UNION SELECT @@VERSION ,@@VERSION,@@VERSION,
1=1&#39;1
```

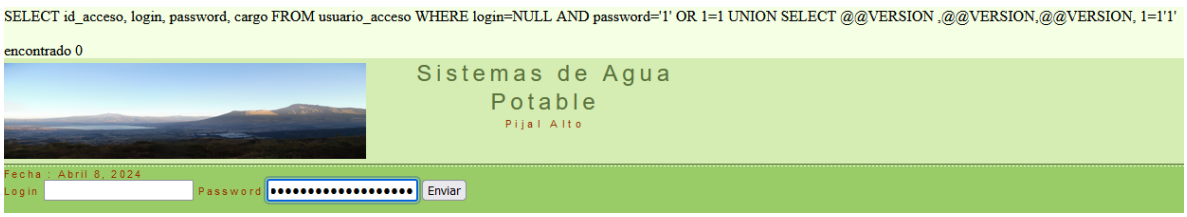


Figura 94 WSTG-INPV-05 Inyección Sql, POST, descubriendo el servidor.

Resultando una consulta Sql como la siguiente:

SELECT id_acceso, login, password, cargo FROM usuario_acceso WHERE login=NULL AND password='1' OR 1=1 UNION SELECT @@VERSION ,@@VERSION,@@VERSION, 1=1'1'

Como podemos ver hemos podido concatenar una inyección Sql pero para este caso de autenticación de usuario no surte ningún efecto, pues debe haber una salida de resultados o impresión del valor recuperado.

Si probamos esta consulta en un editor Sql para la base de datos podemos verificar el siguiente resultado.

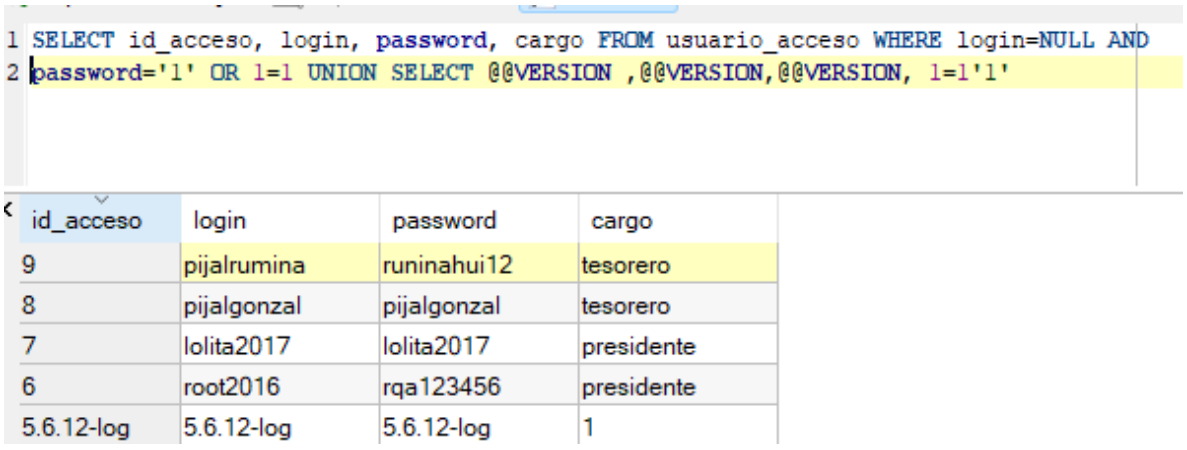


Figura 95 WSTG-INPV-05 Inyección Sql, consulta, descubriendo servidor

Comprobamos que la consulta si es válida y hemos podido recuperar la versión 5.6.12-log de servidor Mysql pero no dentro de la aplicación.

3.1.7. Iniciación de sesiones de usuario.

De la misma forma que en la sección anterior podemos inyectar Sql para revisar el usuario actual de la base de datos.

Ingresamos en password la siguiente inyección sql:

```
1&#39; OR 1=1 UNION SELECT USER() ,USER(),USER(), 1=&#39;1
```

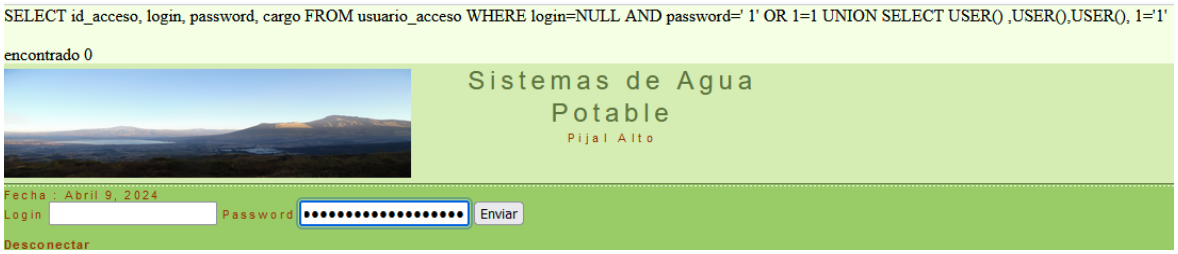


Figura 96 WSTG-INPV-05 Inyección Sql, POST, descubriendo usuarios de la BD.

Nuevamente probamos mediante un editor Sql para probar la validez de la consulta.

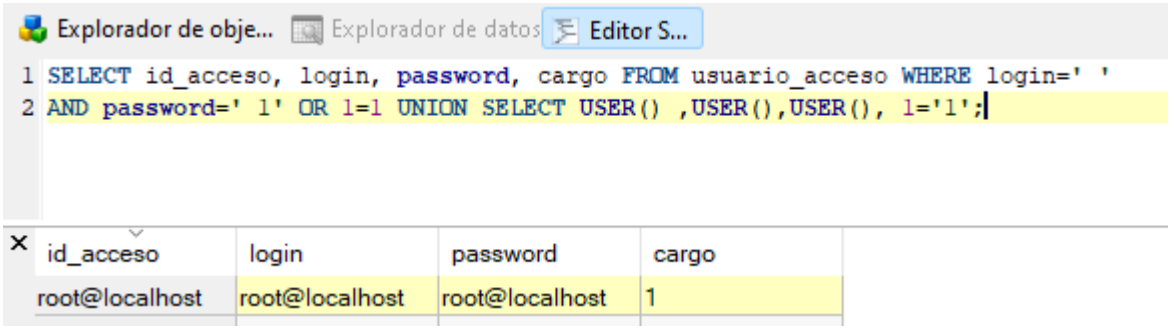


Figura 97 WSTG-INPV-05 Inyección Sql, descubriendo usuarios, ejecución de consulta Sql.

Podemos verificar el usuario de la base de datos, pero no dentro de la aplicación web.

3.1.8. Nombre de la base de datos.

Ingresamos en password la siguiente inyección sql:

```
1 1&#39; OR 1=1 UNION SELECT DATABASE() ,DATABASE(),DATABASE(),
1=&#39;1
```

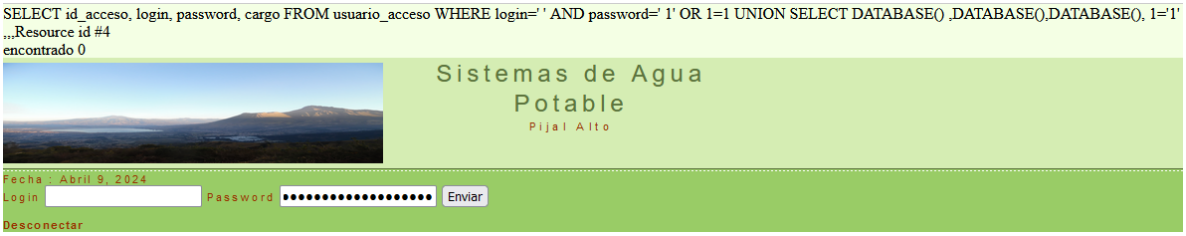


Figura 98 WSTG-INPV-05 Inyección Sql, nombre de la BD.

Mediante un editor Sql verificamos la validez de la consulta.

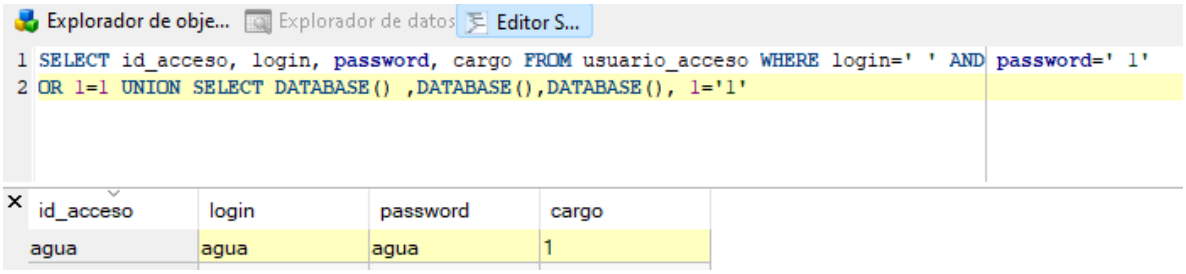


Figura 99 WSTG-INPV-05 Inyección Sql, nombre de BD, ejecución.

Podemos verificar la validez de la consulta, pero no dentro de la aplicación, en una inyección SQL estándar, puede mostrar los resultados directamente en una página como salida normal o como un error de MySQL, un buen ataque es conocer los resultados obligando a una función o procedimiento o al propio servidor a arrojar un error.

El siguiente proceso procedemos a verificar con la herramienta ZAP las vulnerabilidades ocurridas.

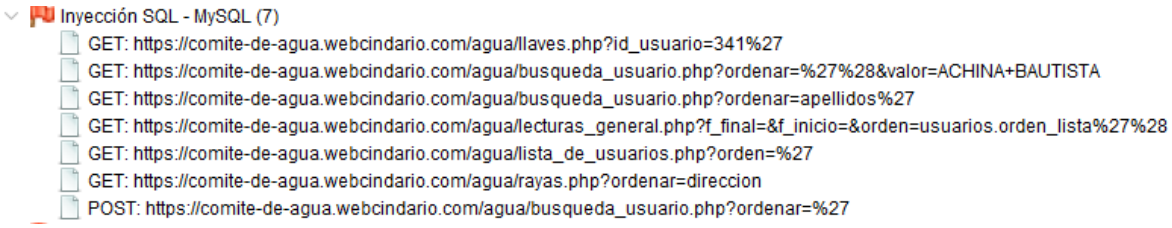


Figura 100 WSTG-INPV-05 Vulnerabilidades Mysql mediante ZAP.

Una forma de revisar que una página web es vulnerables, cuando se inserta ciertos parámetros, caracteres o valores a la consulta y resulta una respuesta del servidor mysql de

error de consulta, pudiendo darnos cuenta que podríamos adaptar una inyección Sql que dé resultados certeros.

- https://comite-de-agua.webcindario.com/agua/busqueda_usuario.php?ordenar=%27

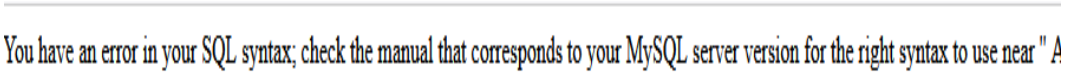


Figura 101 WSTG-INPV-05 Evidencia de vulnerabilidad Mysql.

Debemos recordar que para insertar sql en una consulta enviada por el método _GET debemos reconocer los siguientes valores ASCII Encoding Reference.

Tabla 22 WSTG-INPV-05 Caracteres, ASCII Encondig permiten inyección Sql.

ASCII Encoding Reference	**CARACTER**	**NOMBRE**
%27	‘	Apostrofe
%28	(	abrir paréntesis
%29	)	Cerrar paréntesis

Una de las vulnerabilidades que hemos encontrado es enlistar toda una lista de llaves de usuarios de consumo de agua:

https://comite-de-agua.webcindario.com/agua/llaves.php?id_usuario=426+OR+1+=+1

Por lo que podemos considerar que la aplicación es vulnerable a inyección Mysql.

Anexo 2 Parámetros de evaluación y factores de riesgos.

1. Parámetros de evaluación.

Habilidad requerida		**Motivo**		**Oportunidad**		**Tamaño de población**	
No Applicable [0]	0	No Applicable [0]	0	Se requiere acceso completo o recursos costosos [0]	0	No Applicable [0]	0
Sin habilidades tecnicas [1]	1	Recompensa baja o nula [1]	1	Acceso especial o recursos requeridos [4]	4	Administradores de sistemas [2]	2
Algunas habilidades técnicas [3]	3	Posible recompensa [4]	4	Se requiere algún acceso o recursos [7]	7	Usuarios de intranet [4]	4
Usuario avanzado de computadora [5]	5	Alta recompensa [9]	9	No se requiere acceso ni recursos [9]	9	Socios [5]	5
Habilidades de redes y programación. [6]	6					Usuarios autenticados [6]	6
Habilidades de penetración de seguridad. [9]	9					Usuarios de internet anónimos [9]	9
Perdida de confidencialidad		**Perdida de integridad**		**Perdida de disponibilidad**		**Perdida de responsabilidad**	
No Applicable [0]	0	No Applicable [0]	0	No Applicable [0]	0	No Applicable [0]	0
Se divulgan datos mínimos no confidenciales [2]	2	Datos mínimos ligeramente corrupto [1]	1	Servicios secundarios mínimos interrumpidos [1]	1	Ataque totalmente rastreable hasta el individuo [1]	1
Se divulgan numerosos datos no confidenciales [6]	6	Datos minimos seriamente corruptos [3]	3	Servicios primarios mínimos interrumpidos [5]	5	Ataque posiblemente rastreable hasta un individuo [7]	7
Se revelan numerosos datos críticos [7]	7	Amplios datos ligeramente corruptos [5]	5	Amplios servicios primarios interrumpidos [7]	7	Ataque completamente anónimo [9]	9
Todos los datos divulgados [9]	9	Amplios datos seriamente corruptos [7]	7	Todos los servicios completamente perdidos. [9]	9		
		Todos los datos totalmente corruptos. [9]	9				

Figura 102 Parámetros de evaluación de riesgo 1.

Facil de descubrir		**Facilidad de explotación**		**Conciencia**		**Deteción de intrusiones**	
No Applicable [0]	0	No Applicable [0]	0	No Applicable [0]	0	Not} Applicable [0]	0
Practicamente imposible [1]	1	Teórico [1]	1	Desconocido [1]	1	Detección activa en la aplicación. [1]	1
Dificil [3]	3	Dificil [3]	3	Oculto [4]	4	Registrado y revisado [3]	3
Facil [7]	7	Facil [5]	5	Obio [6]	6	Registrado sin revisión [8]	8
Herramientas automatizadas disponibles [9]	9	Herramientas automatizadas	9	Conocimiento público [9]	9	No registrado [9]	9
Daño financiero		**Daño de reputación**		**Incumplimiento**		**Violación de privacidad**	
No Applicable [0]	0	No Applicable [0]	0	No Applicable [0]	0	No Applicable [0]	0
Los daños cuestan menos que solucionar el problema [1]	1	Daño mínimo [1]	1	Violación menor [2]	2	un individuo [3]	3
Efecto menor en el beneficio anual [3]	3	Pérdida de cuentas importantes [4]	4	Clara violación [5]	5	Cientos de personas [5]	5
Efecto significativo en el beneficio anual [7]	7	Pérdida de buena voluntad [5]	5	Violación de alto perfil [7]	7	Miles de personas [7]	7
Bancarrota [9]	9	Marca de daño [9]	9			Millones de personas [9]	9

Figura 103 Parámetro de evaluación de riesgo 2.

2. **Clasificación de factores de riesgo.**

   - **Factores de riesgo por probabilidad según** OWASP-Testing-Checklist (2023)**.**
     - Factores del agente amenazante; habilidades requeridas, motivo, oportunidad, tamaño de población.
     - Factores de la vulnerabilidad; fácil de descubrir, fácil explotación, conciencia, detección de intrusiones.
   - **Factores de riesgo por Impacto según** OWASP-Testing-Checklist (2023)**.**
     - Factores de impacto técnico; perdida de confidencialidad, perdida de integridad, perdida de disponibilidad, perdida de responsabilidad.
     - Factores de impacto empresarial; daño financiero, daño a la reputación, incumplimiento, violación de privacidad.

3. **Cálculo de gravedad general de riesgo.**

Según OWASP-Testing-Checklist (2023) se calculará la gravedad general de riesgo en base a:

- ✓ **Puntuación de probabilidad es = p**romedio de lista de factores de riesgo por probabilidad.
- ✓ Si promedio (<3) es bajo, si (>=3 y < 6) es medio, si (>=6) es alto.
- ✓ **Puntuación de impacto=p**romedio de lista de factores de riesgo por impacto.
- ✓ Si promedio (<3) es bajo, si (>=3 y < 6) es medio, si (>=6) es alto.

Cálculo de gravedad general de riesgo

	Impacto		
Probabilidad	Bajo	Moderado	Alto
Bajo	Nada	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 104 Calculo de gravedad general de riesgo.

Anexo 3 Evaluación de riesgos del sistema de información.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Rastreo de credenciales de usuario

Factores de probabilidad	
Factores de la amenaza	
Habilidades requeridas	Algunas habilidades técnicas [3]
Motivo	Alta recompensa [9]
Oportunidad	Acceso especial o recursos requeridos [4]
Tamaño de poblacion	Usuarios de internet anónimos [9]

Factores de impacto		
Factores técnicos de impacto		
3	Pérdida de confidencialid	Todos los datos divulgados [9] 9
9	Pérdida de integridad	Amplios datos ligeramente corruptos [5] 5
4	Pérdida de disponibilidad	Amplios servicios primarios interrumpidos [7] 7
9	Pérdida de responsabilidad	Ataque posiblemente rastreable hasta un individuo [7] 7

Factores de la vulnerabilidad	
Fácil de descubrir	Facil [7]
Facilidad de explotación	Facil [5]
Conciencia	Conocimiento público [9]
Detección de intrusiones	No registrado [9]

Factores de impacto empresarial		
7	Daño financiero	Efecto significativo en el beneficio anual [7] 7
5	Daño a la reputación	Marca de daño [9] 9
9	Incumplimiento	Violación de alto perfil [7] 7
9	Violación de privacidad	Cientos de personas [5] 5

Puntuación de probabilidad:	**6,875**

Puntuación de impacto:	**7**

Gravedad general de riesgo:	**Critico**

Probabilidad	**Impacto**		
	Bajo	Moderado	**->Alto<-**
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	**->Critico<-**

Figura 105 Calculo de riesgo, credenciales enviadas por canales no cifrados.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema:Aseguramiento Token de sesion

Factores de probabilidad		
Factores de la amenaza		
Habilidades requeridas	Algunas habilidades técnicas [3]	3
Motivo	Alta recompensa [9]	9
Oportunidad	Acceso especial o recursos requeridos [4]	4
Tamaño de poblacion	Usuarios de internet anónimos [9]	9
Factores de la vulnerabilidad		
Fácil de descubrir	Facil [7]	7
Facilidad de explotación	Facil [5]	5
Conciencia	Obio [6]	6
Detección de intrusiones	Registrado sin revisión [8]	8

Factores de impacto		
Factores técnicos de impacto		
Pérdida de confidencialid	Todos los datos divulgados [9]	9
Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Pérdida de disponibilidad	No Applicable [0]	0
Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Factores de impacto empresarial		
Daño financiero	Efecto significativo en el beneficio anual [7]	7
Daño a la reputación	Marca de daño [9]	9
Incumplimiento	Violación de alto perfil [7]	7
Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**6,375**	**Puntuación de impacto:**	**6,375**

Gravedad general de riesgo :	**Critico**

Probabilidad	**Impacto**		
	Low	Moderado	->Alto<-
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	->Critico<-

Figura 106 Calculo de riegos Tokens y sesión.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Credenciales por defecto

Factores de probabilidad.		
Factores de la amenaza.		
Habilidades requeridas	Algunas habilidades técnicas [3]	3
Motivo	Alta recompensa [9]	9
Oportunidad	Acceso especial o recursos requeridos [4]	4
Tamaño de poblacion	Usuarios de internet anónimos [9]	9
Factores de la vulnerabilidad.		
Fácil de descubrir	Facil [7]	7
Facilidad de explotación	Facil [5]	5
Conciencia	Conocimiento público [9]	9
Detección de intrusiones	No registrado [9]	9

Factores de impacto.		
Factores tecnicos de impacto.		
Pérdida de confidencialic	Todos los datos divulgados [9]	9
Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Pérdida de disponibilidad	Amplios servicios primarios interrumpidos [7]	7
Pérdida de responsabilidad	Ataque posiblemente rastreable hasta un individuo [7]	7
Factores de impacto empresariál.		
Daño financiero	Efecto significativo en el beneficio anual [7]	7
Daño a la reputación	Marca de daño [9]	9
Incumplimiento	Violación de alto perfil [7]	7
Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**6,875**

Puntuación de impacto:	**7**

Gravedad general de riesgo :	**Critico**

Probabilidad	**Impacto**		
	Bajo	Moderado	->Alto<-
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	->Critico<-

Figura 107 Calculo de riesgos, credenciales por defecto

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: No existe mecanismos de bloqueo y desbloqueo

Factores de probabilidad.					
Factores de la amenaza.					
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialic	Todos los datos divulgados [9]	9
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	Amplios servicios primarios interrumpidos [7]	7
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	Ataque posiblemente rastreable hasta un individuo [7]	7
Factores de la vulnerabilidad.			***Factores de impacto empresarial.***		
Fácil de descubrir	Facil [7]	7	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Facil [5]	5	Daño a la reputación	Marca de daño [9]	9
Conciencia	Conocimiento público [9]	9	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**6,25**	**Puntuación de impacto:**	**7**

Gravedad general de riesgo :	**Critico**

Probabilidad	**Impacto**		
	Bajo	Moderado	->Alto<-
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	->Critico<-

Figura 108 Calculo de riesgo, mecanismos de bloqueo y desbloqueo

OWASP Calculadora de Evaluación de riesgo

Risk Assessment Calculator

Problema: Solicitud de pagina directa(Acceso a información confidencial)

Factores de probabilidad.			**Factores de impacto.**		
Factores de la amenaza.			***Factores tecnicos de impacto.***		
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialid	Todos los datos divulgados [9]	9
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	No se requiere acceso ni recursos [9]	9	Pérdida de disponibilidad	No Applicable [0]	0
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Factores de la vulnerabilidad.			***Factores de impacto empresarial.***		
Fácil de descubrir	Facil [7]	7	Daño financiero	No Applicable [0]	0
Facilidad de explotación	Facil [5]	5	Daño a la reputación	Daño mínimo [1]	1
Conciencia	Conocimiento público [9]	9	Incumplimiento	Violación menor [2]	2
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**6,875**	**Puntuación de impacto:**	**3,875**

Gravedad general de riesgo :	**Alto**

Probabilidad	**Impacto**		
	Bajo	->Moderado<-	Alto
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	->Alto<-	Critico

Figura 109 Calculo de riesgo, acceso a información confidencial.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Inyeccion automatica de credenciales

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza.			***Factores tecnicos de impacto***		
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialic	Todos los datos divulgados [9]	9
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	Amplios servicios primarios interrumpidos [7]	7
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	Ataque posiblemente rastreable hasta un individuo [7]	7
Factores de la vulnerabilidad			***Factores de impacto empresarial.***		
Fácil de descubrir	Facil [7]	7	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Facil [5]	5	Daño a la reputación	Marca de daño [9]	9
Conciencia	Conocimiento público [9]	9	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**6,25**	**Puntuación de impacto:**	**7**

Gravedad general de riesgo:	**Critico**

Probabilidad	**Impacto**		
	Bajo	Moderado	->Alto<-
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	->Critico<-

Figura 110 Cálculo de riesgo, inyección de credenciales

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Tokens sin tiempo de exíracion

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores tecnicos de impacto***		
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialic	Todos los datos divulgados [9]	9
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	Amplios servicios primarios interrumpidos [7]	7
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	Ataque posiblemente rastreable hasta un individuo [7]	7
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Herramientas automatizadas disponibles [9]	9	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Herramientas automatizadas disponibles [9]	9	Daño a la reputación	Marca de daño [9]	9
Conciencia	Obio [6]	6	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**6,625**	**Puntuación de impacto:**	**7**

Gravedad general de riesgo :	**Critico**

Probabilidad	**Impacto**		
	Bajo	Moderado	->Alto<-
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	->Critico<-

Figura 111 Cálculo de riesgo, sesiones sin tiempo de expiración.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: No existe politicas de fortalecmineto de password

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores tecnicos de impacto***		
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialid	Todos los datos divulgados [9]	9
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	No Applicable [0]	0
Tamaño de poblacion	Usuarios de internet anónimos [9]	9	Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Facil [7]	7	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Facil [5]	5	Daño a la reputación	Marca de daño [9]	9
Conciencia	Obio [6]	6	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**6,25**	**Puntuación de impacto:**	**6,375**

Gravedad general de riesgo :	**Critico**

Probabilidad	**Impacto**		
	Bajo	Moderado	->Alto<-
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	->Critico<-

Figura 112 Calculo de riesgo, políticas de fortalecimiento de credenciales

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: No existe preguntas de seguridad

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores tecnicos de impacto***		
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialic	Todos los datos divulgados [9]	9
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	No Applicable [0]	0
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	No Applicable [0]	0
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Facil [7]	7	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Facil [5]	5	Daño a la reputación	Marca de daño [9]	9
Conciencia	Obio [6]	6	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**5,875**	**Puntuación de impacto:**	**5,25**

Gravedad general de riesgo :	**MODERADO**

	Impacto		
Probabilidad	Bajo	->Moderado<-	alto
Bajo	Note	Bajo	Moderado
->Moderado<-	Bajo	->Moderado<-	Alto
Alto	Moderado	Alto	Critico

Figura 113 Cálculo de riesgo, preguntas de seguridad en credenciales

OWASP Calculadora de Evaluación de riesgo

Risk Assessment Calculator

Problema: No existe funcionalidades de reinicio de credenciales

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores tecnicos de impacto***		
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialic	Todos los datos divulgados [9]	9
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	Todos los servicios completamente perdidos. [9]	9
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	No Applicable [0]	0
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Facil [7]	7	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Facil [5]	5	Daño a la reputación	Pérdida de cuentas importantes [4]	4
Conciencia	Obio [6]	6	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**5,875**	**Puntuación de impacto:**	**5,125**

Gravedad general de riesgo :	**MODERADO**

	Impacto		
Probabilidad	Bajo	->Moderado<-	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	Bajo	->Moderado<-	Alto
Alto	Moderado	Alto	Critico

Figura 114 Cálculo de riesgo, funcionalidades de reinicio de credenciales.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema:No existe canal alternativo de autenticación

Factores de probabilidad					
Factores de la amenaza			***Factores tecnicos de impacto***		
Habilidades requeridas	No Applicable [0]	0	Pérdida de confidencialid	No Applicable [0]	0
Motivo	No Applicable [0]	0	Pérdida de integridad	No Applicable [0]	0
Oportunidad	Se requiere acceso completo o recursos costosos [0]	0	Pérdida de disponibilidad	Todos los servicios completamente perdidos. [9]	9
Tamaño de poblacion	No Applicable [0]	0	Pérdida de responsabilidad	No Applicable [0]	0
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	No Applicable [0]	0	Daño financiero	No Applicable [0]	0
Facilidad de explotación	No Applicable [0]	0	Daño a la reputación	Daño mínimo [1]	1
Conciencia	Obio [6]	6	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**1,875**	**Puntuación de impacto:**	**2,125**

Gravedad general de riesgo :	**Note**

	Impacto		
Probabilidad	->Bajo<-	Moderado	Alto
->Bajo<-	->Note<-	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 115 Calculo de riesgo, canal alternativo de autenticación.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: No tiene autenticacion MFA

Factores de probabilidad					
Factores de la amenaza			***Factores tecnicos de impacto***		
Habilidades requeridas	Habilidades de penetración de seguridad. [9]	9	Pérdida de confidencialid	Todos los datos divulgados [9]	9
Motivo	Posible recompensa [4]	4	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Se requiere algún acceso o recursos [7]	7	Pérdida de disponibilidad	Amplios servicios primarios interrumpidos [7]	7
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	Ataque posiblemente rastreable hasta un individuo [7]	7
Factores de la vulnerabilidad			***Factores de impacto empresarial.***		
Fácil de descubrir	Facil [7]	7	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Facil [5]	5	Daño a la reputación	Marca de daño [9]	9
Conciencia	Conocimiento público [9]	9	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**7**	**Puntuación de impacto:**	**7**

Gravedad general de riesgo :	**Critico**

Probabilidad	**Impacto**		
	Bajo	Moderado	**->Alto<-**
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	**->Critico<-**

Figura 116 Calculo de riesgo, autentificación MFA

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Inyeccion Sql

Factores de probabilidad					
Factores de la amenaza					
Habilidades requeridas	Habilidades de redes y programación. [6]	6	Pérdida de confidencialid	Todos los datos divulgados [9]	9
Motivo	Posible recompensa [4]	4	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Se requiere algún acceso o recursos [7]	7	Pérdida de disponibilidad	Amplios servicios primarios interrumpidos [7]	7
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	Ataque posiblemente rastreable hasta un individuo [7]	7
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Herramientas automatizadas disponibles [9]	9	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Herramientas automatizadas disponibles [9]	9	Daño a la reputación	Marca de daño [9]	9
Conciencia	Conocimiento público [9]	9	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**7,375**	**Puntuación de impacto:**	**7**

Gravedad general de riesgo	**Critico**

Probabilidad	**Impacto**		
	Bajo	Moderado	**->Alto<-**
Bajo	Note	Bajo	Moderado
Moderado	Bajo	Moderado	Alto
->Alto<-	Moderado	Alto	**->Critico<-**

Figura 117 Calculo de riesgo, inyección Sql.

Anexo 4 Código fuente mecanismos de autenticación.

El código fuente completo se encuentra en Github: https://github.com/rquilumbaquia/2FA-y-QR/blob/main/agua.zip

HSTS(Strict-transport-security).

```
Nota: Este código va en los encabezados de cada archivo php.
<?php
date_default_timezone_set("America/Bogota");
header("strict-transport-security:max-age=600");
header("X-Frame-Options: DENY");
header('X-Content-Type-Options: nosniff');
header("Content-Security-Policy: script-src 'self' 'unsafe-eval'
'unsafe-inline' ;style-src 'self';form-action 'self' ");
?>
```

Generación de hash de contraseña.

Archivo: \agua\admin\secciones\usuarios\guardar_usuario.php

A continuación, pasamos a describir parte del código fuente php principal para convertir un Hash.

```
<?php
    /* Buscando un coste apropiado para generación del Hash */
    $timeTarget = 0.05; // Tentativo para 50 milisegundos
    $coste = 8; //Coste tentativo
    do {
        $coste++;
        $inicio = microtime(true);
        password_hash("test", PASSWORD_BCRYPT, ["cost" => $coste]);
        $fin = microtime(true);
    } while (($fin - $inicio) < $timeTarget);
    echo "Coste apropiado encontrado: " . $coste . "\n";
```

```
    $opciones = ['cost' => $coste, ];
    /* realizando el Hash de la contraseña*/
    $password=password_hash($_POST['password-input'], PASSWORD_BCRYPT, $opciones);
?>
```

La función principal para la generación de un Hash en php es:

- **password_hash($_POST['password-input'], PASSWORD_BCRYPT, $opciones);**
- **$_POST['password-input'],** es el texto de la contraseña que ingresa por el método POST.
- **PASSWORD_BCRYPT,** permitirá crear un Hash de 60 caracteres.
- **$opciones,** Si se omite, se usará el valor predeterminado `10`. Este es un buen coste de referencia, pero se podría considerar aumentarlo dependiendo del hardware.

Generación de código numérico hash, aleatorio para restablecimiento de contraseña.

Archivo: \agua\admin\recuperar.php .

Lo siguiente es la porción de código php, que permite generar un hash del código numérico generado aleatoriamente para almacenar en la base de datos del usuario y enviar el código mediante un correo electrónico al usuario quien pretende restablecer la contraseña.

```
<?php
if((isset($_POST['correo']))&&($_SESSION['tocte']==$_POST['csrf_token']))
{
    /*Revisa si existe el usuario con el correo*/
    /*Proceso de saneamiento de variables y prevención de inyección Sql*/
    $caracteresNoPermitidos = array(" ","!",'"',"#","%","&","'","(",")","+","=",";");
    $correo=(isset($_POST['correo']))?$_POST['correo']:"";
```

```
   /*Proceso de saneamiento de variables y prevención de inyección Sql*/
   $caracteresNoPermitidos = array(" ","!",'"',"#","%","&","'","(",")","+","=",";");
   $correo = str_replace($caracteresNoPermitidos,'',$correo);
   $sentencia=$conexion->prepare("SELECT * FROM usuario_acceso WHERE correo=:txtcorreo AND activo='si'");
   $sentencia->bindParam(':txtcorreo',$correo);
   $sentencia->execute();
   if($sentencia->rowCount()>0)
   {
   $lista_usuario=$sentencia->fetch(PDO::FETCH_LAZY);
   $id_acceso=$lista_usuario['id_acceso'];
   $archivo=$lista_usuario['imagen_qr'];
   $codigo="";
   for($i=0;$i<3;$i++){$codigo=$codigo."".rand(1,200);}
   /*Hash del codigo aleatorio */
   $timeTarget = 0.05; // 50 milisegundos
   $coste = 8;
do {
   $coste++;
   $inicio = microtime(true);
   password_hash("test", PASSWORD_BCRYPT, ["cost" => $coste]);
   $fin = microtime(true);
   }while (($fin - $inicio) < $timeTarget);
   /*echo "Coste apropiado encontrado: " . $coste . "\n";*/
   $opciones = ['cost' => $coste,];
   $codigo_hash=password_hash($codigo, PASSWORD_BCRYPT, $opciones);
   /*fin del hash */
   /*almacena o actualiza el hash en base de datos*/
   $hora=date("H:i:s");
   $sentencia_update=$conexion->prepare("UPDATE usuario_acceso SET
   codigo_recuperacion=:codigo,hora_recuperacion=:txthora WHERE id_acceso=:idacceso");
   $sentencia_update->bindParam(':codigo',$codigo_hash);
   $sentencia_update->bindParam(':txthora',$hora);
```

```
    $sentencia_update->bindParam(':idacceso',$id_acceso);
    $sentencia_update->execute();
    /*Enviamos un correo electrónico*/
    include("../admin/libs/PHPMailer/mail.php");
    $_GET['autentificar']="si";
    $_GET['id_acceso']=$id_acceso;
   }else{
        $mensaje="No tiene permiso para recuperar cuentas";
        echo ".";
       }
}               ?>
```

Lógica del mecanismo de cambio de contraseña.

Archivo: \agua\admin\cambiopassword.php

La lógica para este mecanismo de cambio de contraseña consiste en 2 funciones o procesos, el primero consiste en verificar si existe un código de recuperación para verificar dentro de la base de datos y dar paso a la generación de una nueva contraseña y el segundo consiste en almacenar el Hash de la nueva contraseña en la base de datos para el usuario.

```
/* función para comprobación de tiempo de expiración y generación de un nuevo hash de contraseña
y almacenamiento del hash en la base de datos */

if((isset($_POST['password-input']))&&(isset($_POST['password-input2']))&&(isset($_POST['id_acceso'])))
{    /* Recupera los datos del usuario de la base de datos*/
     $id_acceso=$_POST['id_acceso'];
     $sentencia_revision=$conexion->prepare("SELECT * FROM usuario_acceso WHERE
     id_acceso=:txtidacceso AND codigo_recuperacion!='' ");
     $sentencia_revision->bindParam(':txtidacceso',$id_acceso);
```

```
$sentencia_revision->execute();
$lista_usuario=$sentencia_revision->fetch(PDO::FETCH_LAZY);
/* si existe un registro del usuario */
if($sentencia_revision->rowCount()>0)
 { /*Generamos el tiempo actual*/
   $hora_actual=date("His");
   $hora_recuperacion = new DateTime($lista_usuario['hora_recuperacion']);
   $hora_limite=($hora_recuperacion->format('His'))+300;
   /* comparamos el tiempo actual con el tiempo de expiracion */
if(($hora_actual>$hora_limite))
  {
   $mensaje= "El tiempo ha expirado realice un nuevo proceso de recuperacion";
   $id_acceso=$_POST['id_acceso'];
   $sentencia=$conexion->prepare("UPDATE usuario_acceso SET
   codigo_recuperacion='',hora_recuperacion='',intentos=''
   WHERE id_acceso=:txtidacceso ");
   $sentencia->bindParam(':txtidacceso',$id_acceso);
   $sentencia->execute();
 }else{
   /* Caso contrario comparamos si las contraseña nuevas son iguales password1 password 2 */
   if($_POST['password-input']==$_POST['password-input2'])
    {  /*Generamos un nuevo hash para la nueva contraseña */
       $timeTarget = 0.05; // 50 milisegundos
       $coste = 8;
    do {
       $coste++;
       $inicio = microtime(true);
       password_hash("test", PASSWORD_BCRYPT, ["cost" => $coste]);
       $fin = microtime(true);
       }  while (($fin - $inicio) < $timeTarget);
          /*echo "Coste apropiado encontrado: " . $coste . "\n";*/
          $opciones = ['cost' => $coste,];
```

```
            $password=password_hash($_POST['password-input'], PASSWORD_BCRYPT, $opciones);
            /* fin de genración del hash */
            $id_acceso=$_POST['id_acceso'];
            $imagen_qr=(isset($_POST['imagen_qr']))?$_POST['imagen_qr']:"";
            /*Eliminamos el archivo QR antiguo */
            If ($imagen_qr!=$lista_usuario['imagen_qr']) {
              unlink("./libs/generadorqr/temp/".$lista_usuario['imagen_qr']);
              echo "Archivo eliminado correctamente";
            } else {echo "No se elimina";}
            /*Actualizamos la base de datos del usuario con un nuevo hash de contraseña */
            $sentencia=$conexion->prepare("UPDATE usuario_acceso SET
            password=:txtpassword,imagen_qr=:txtimagen,codigo_recuperacion='',
            hora_recuperacion='',intentos='' WHERE id_acceso=:txtidacceso AND codigo_recuperacion<>''");
            $sentencia->bindParam(':txtpassword',$password);
            $sentencia->bindParam(':txtimagen',$imagen_qr);
            $sentencia->bindParam(':txtidacceso',$id_acceso);
            $sentencia->execute();
            $mensaje="El cambio de contraseña se ha realizado correctamente";
         }else{
            $mensaje="El Password no es igual";}
          }
       }else{$mensaje="el registro ya ha sido alterado(No se realiza ningún cambio)";}
}

/*funcion de recepción de codigo de recuperacion y despliegue de formulario de cambio de contraseña */
if((isset($_POST['id_acceso']))&&(isset($_POST['codigo_recuperacion'])))
{   /*Recuperacion del usuario de la base de datos con el codigo de recuperacion */
    $id_acceso=$_POST['id_acceso'];
    $codigo=$_POST['codigo_recuperacion'];
    $sentencia=$conexion->prepare("SELECT * FROM usuario_acceso WHERE id_acceso=:txtidacceso ");
    $sentencia->bindParam(':txtidacceso',$id_acceso);
    $sentencia->execute();
```

```
  $lista_usuario=$sentencia->fetch(PDO::FETCH_LAZY);
  /*Verificamos el Hash del codigo con el codigo ingresado*/
  if (password_verify($codigo, $lista_usuario['codigo_recuperacion']))
        {
          /*Generamos la hora actual */
          $hora_actual=date("His");
          $valor=$lista_usuario['hora_recuperacion'];
          $hora_recuperacion = new DateTime($valor);
          $hora_limite=($hora_recuperacion->format('His'))+300;
          /*Verificamos la expiración por tiempo del código de recuperación */
          if(($hora_actual>$hora_limite))
          {
          $mensaje= "El tiempo ha expirado realice un nuevo proceso de recuperación";
          $id_acceso=$_POST['id_acceso'];
          $sentencia=$conexion->prepare("UPDATE usuario_acceso SET
          codigo_recuperacion='',hora_recuperacion='' WHERE id_acceso=:txtidacceso ");
          $sentencia->bindParam(':txtidacceso',$id_acceso);
          $sentencia->execute();
          }else{
   /* si no existe inconveniente desplegamos el formulario de
   restablecimiento de contraseña incluyento
   La generación de un código QR */
 /*-------------------------------------------------------------------------------*/
} ?>
```

Verificación de usuario y redirección.

Archivo:\agua\admin\login.php

```
if (!isset($_SESSION)) {  session_start();}
if($_POST)
{ /*si existe el usuario*/
  $usuario=(isset($_POST['usuario']))?$_POST['usuario']:"";
  $password=(isset($_POST['password']))?$_POST['password']:"";
  $sentencia=$conexion->prepare("SELECT id_acceso,password,cargo,f_inicio,f_final,anio_lectivo,correo,intentos,
  fecha_caducidad,aplicar_2fa,activo,dia_hora_bloqueo,imagen_qr FROM usuario_acceso WHERE login=:txtusuario ");
  $sentencia->bindParam(':txtusuario',$usuario);
  $sentencia->execute();
  $lista_usuario=$sentencia->fetch(PDO::FETCH_LAZY);
  if (($sentencia->rowCount())<1)
  {
   $mensaje="El Usuario i/o Contraseña es incorrecto";
  }else
  {
  if (password_verify($password, $lista_usuario['password']))
  {
  /*Procedimiento Verificación 2FA por email si está activo */
        /* Procedimiento Verificación 2FA por QR */
        /* Procedimiento de verificación y desbloqueo de usuario */
        /* Procedimiento de autenticación directa sin aplicación de 2FA */
else
      {
        /* En caso de que la autenticación es incorrecta incrementa los intentos de acceso y
        si los intentos son mayores a 4 bloquea al acceso al usuario durante 24 hora
        */
        $mensaje="El Usuario i/o Contraseña es incorrecto";
     }
```

Verificación por QR.

Archivo: \agua\admin\verificacion.php

```
      /* Procedimiento de verificacion por QR */
if((isset($_POST['entrar_QR']))&&($_POST['entrar_QR']=="Verificar")&&($_SESSION['tocte']==$_POST['csrf_token']))
      {
      if((isset($_POST['id_acceso']))&&(isset($_POST['codigo']))&&(isset($_POST['secreto'])))
      {
        $secret=$_POST['secreto'];
        //$oneCode = $ga->getCode($secret);
        $oneCode =$_POST['codigo'];
        echo "Verificando Code '$oneCode' Y Secreto '$secret':";
        $checkResult = $ga->verifyCode($secret, $oneCode, 2);    // 2 = 2*30sec clock tolerance
        if ($checkResult) {
            $id_acceso=$_POST['id_acceso'];
            $sentencia=$conexion->prepare("SELECT * FROM usuario_acceso WHERE id_acceso=:txtidacceso ");
            $sentencia->bindParam(':txtidacceso',$id_acceso);
            $sentencia->execute();
            $lista_usuario=$sentencia->fetch(PDO::FETCH_LAZY);
            echo 'OK';
            $mensaje="Verificación correcta";
            $_SESSION['id_acceso']=$lista_usuario['id_acceso'];
            $_SESSION['usuario']=$lista_usuario['login'];
            $_SESSION['cargo']=$lista_usuario['cargo'];
            $_SESSION['logeado']=true;
            $_SESSION['f_inicio_usuario']=$lista_usuario['f_inicio'];
            $_SESSION['f_final_usuario']=$lista_usuario['f_final'];
            $fecha_caducidad=$lista_usuario['fecha_caducidad'];
            $anio_lectivo=$lista_usuario['anio_lectivo'];
            $sentencia_anio=$conexion->prepare("SELECT * FROM a_lectivo WHERE anio_lectivo=:txtanio ");
```

```
      $sentencia_anio->bindParam(':txtanio',$anio_lectivo);
      $sentencia_anio->execute();
      $lista_anio=$sentencia_anio->fetch(PDO::FETCH_LAZY);
      $_SESSION['anio_lectivo']=$lista_anio['anio_lectivo'];
      $_SESSION['f_inicio_periodo']=$lista_anio['f_inicio'];
      $_SESSION['f_final_periodo']=$lista_anio['f_final'];
      $id_acceso=$_POST['id_acceso'];
      $sentencia=$conexion->prepare("UPDATE usuario_acceso SET
      codigo_recuperacion='',hora_recuperacion='',intentos='' WHERE id_acceso=:txtidacceso ");
      $sentencia->bindParam(':txtidacceso',$id_acceso);
      $sentencia->execute();
      $mensaje="Succes";
      $MM_redirectLoginSuccess ="index.php";
      header("Location: " . $MM_redirectLoginSuccess );
  } else {
     $mensaje="La verificacion ha fallado";
     echo 'FAILED';
  }
}
}
```

Comprobación de acceso directo a páginas.

Nota: Estas sentencias se encuentran en los encabezados de todas las páginas.

```
/*encabezado de comprobación de usuario con acceso para todas la paginas*/
if(!isset($_SESSION['usuario']))
 {
 $MM_redirectLoginSuccess =$url_base."login.php";
 header("Location: " . $MM_redirectLoginSuccess );
 }
```

Generación de sesiones seguras.

Archivo:\agua\admin\sesion.php

```
<?php
// Función inicio de una nueva sessión
function correr_session() {
    session_name('mySession');
    session_start();
    // Destrucción de sesiones antiguas
    if (!empty($_SESSION['deleted_time']) && $_SESSION['deleted_time'] < time() - 600) {
        session_destroy();
        session_start();
    }
}
// funcion de regeneración de ID de sesión
function regenerar_session_id() {
    // Llame a session_create_id() mientras la sesión está activa
    // Se asegura de que existe colisiones
    if (session_status() != PHP_SESSION_ACTIVE) {
        session_start();
    }
    //Funcion para generación de id aleatrorio.
    $codigo="";
    for($i=0;$i<3;$i++){$codigo=$codigo."".rand(1,200);}
    //fin de función de generació ID aleatoria.
    // Assignación de prefijo de session
    $newid = session_create_id("-".$codigo);
    // Establecer marca de tiempo eliminada.
    $_SESSION['deleted_time'] = time();
    // Finalizar la session
    session_commit();
```

```
    // Asegúrese de aceptar el ID de sesión definido por el usuario
    // NOTA: Debe habilitar use_strict_mode para operaciones normales.
    ini_set('session.use_strict_mode', 0);
    // Establecer la nueva sesión
    session_id($newid);
    // Iniciar una session personalizada
    session_start();
}
// use_strict_mode seguridad.

function iniciar_sesion(){
ini_set('session.use_trans_sid', 0);
ini_set('session.use_strict_mode', 1);
ini_set('session.cookie_httponly', 1);
ini_set('session.cookie_secure', 1);
ini_set('session.use_cookies', 1);
ini_set("session.use_trans_sid", 0);
ini_set('session.use_only_cookies',1);
ini_set('session.hash_function',5);
ini_set('session.hash_bits_per_character',5);
ini_set('session.cookie_samesite', 'Strict');
}
?>
```

Generar y enviar código temporal por correo electrónico.

Archivo: \agua\admin\login.php

```
if (password_verify($password, $lista_usuario['password']))  {
             /*Verificacion 2FA por email si está activo */
        if(($lista_usuario['aplicar_2fa']=="email")&&($lista_usuario['activo']=="si"))
           {
             /*generacion de codigo aleatorio*/
             for($i=0;$i<3;$i++){$codigo=$codigo."".rand(1,200);}
             /*hash del codigo*/
             $timeTarget = 0.05; // 50 milisegundos
             $coste = 8;
          do {
             $coste++;
             $inicio = microtime(true);
             password_hash("test", PASSWORD_BCRYPT, ["cost" => $coste]);
             $fin = microtime(true);
             } while (($fin - $inicio) < $timeTarget);
               /*echo "Coste apropiado encontrado: " . $coste . "\n";*/
               $opciones = ['cost' => $coste,];
             $codigo_hash=password_hash($codigo, PASSWORD_BCRYPT, $opciones);
             /*fin del hash de codigo*/
             $correo=$lista_usuario['correo'];
             $archivo=$lista_usuario['imagen_qr'];
             $id_acceso=$lista_usuario['id_acceso'];
             $hora=date("H:i:s");
             $sentencia_update=$conexion->prepare("UPDATE usuario_acceso SET
             codigo_recuperacion=:codigo,hora_recuperacion=:txthora
             WHERE id_acceso=:idacceso");
             $sentencia_update->bindParam(':codigo',$codigo_hash);
             $sentencia_update->bindParam(':txthora',$hora);
```

```
                $sentencia_update->bindParam(':idacceso',$id_acceso);
                $sentencia_update->execute();
            /*enviamos un correo con  código de verificación generado y redirección a la página de verificación*/
                include("./libs/PHPMailer/mail_verificacion.php");
                $_SESSION['id_acceso']=$lista_usuario['id_acceso'];
                $_SESSION['usuario']=$usuario;
                $MM_redirectLoginSuccess ="verificacion.php";
                header("Location: " . $MM_redirectLoginSuccess );
              }
      }
```

Verificación 2FA con código temporal.

Archivo: \agua\admin\verificacion.php

```
      /* Procedimiento de verificacion 2FA por codigo */
if((isset($_POST['entrar_codigo']))&&($_POST['entrar_codigo']=="Enviar")&&($_SESSION['tocte']==$_POST['csrf_token']))
      {
      if((isset($_POST['id_acceso']))&&(isset($_POST['codigo_recuperacion'])))
        { $id_acceso=$_POST['id_acceso'];
          $codigo=$_POST['codigo_recuperacion'];
          $sentencia=$conexion->prepare("SELECT * FROM usuario_acceso WHERE id_acceso=:txtidacceso ");
          $sentencia->bindParam(':txtidacceso',$id_acceso);
          $sentencia->execute();
          $lista_usuario=$sentencia->fetch(PDO::FETCH_LAZY);
          if (password_verify($codigo, $lista_usuario['codigo_recuperacion']))
          {
                  $hora_actual=date("His");
                  $valor=$lista_usuario['hora_recuperacion'];
                  $hora_recuperacion = new DateTime($valor);
                  $hora_limite=($hora_recuperacion->format('His'))+300;
```

```
                    if(($hora_actual>$hora_limite))
                    {
                      $mensaje= "El tiempo ha expirado realice un nuevo proceso de recuperacion";
                      echo ".";
                      $id_acceso=$_POST['id_acceso'];
                      $sentencia=$conexion->prepare("UPDATE usuario_acceso SET
                      codigo_recuperacion='',hora_recuperacion='' WHERE id_acceso=:txtidacceso ");
                      $sentencia->bindParam(':txtidacceso',$id_acceso);
                      $sentencia->execute();
                      $MM_redirectLoginSuccess ="login.php";
                      header("Location: " . $MM_redirectLoginSuccess );
                      echo ".";
                    }else{
/* se autentica el usuario y se asignan los datos a las sesiones,finalmente se redirecciona a index.php */
                      $_SESSION['id_acceso']=$lista_usuario['id_acceso'];
                      $_SESSION['usuario']=$lista_usuario['login'];
                      $_SESSION['cargo']=$lista_usuario['cargo'];
                      $_SESSION['logeado']=true;
                      $_SESSION['f_inicio_usuario']=$lista_usuario['f_inicio'];
                      $_SESSION['f_final_usuario']=$lista_usuario['f_final'];
                      $fecha_caducidad=$lista_usuario['fecha_caducidad'];
                      $anio_lectivo=$lista_usuario['anio_lectivo'];
                      $sentencia_anio=$conexion->prepare("SELECT * FROM a_lectivo WHERE anio_lectivo=:txtanio ");
                      $sentencia_anio->bindParam(':txtanio',$anio_lectivo);
                      $sentencia_anio->execute();
                      $lista_anio=$sentencia_anio->fetch(PDO::FETCH_LAZY);
                      $_SESSION['anio_lectivo']=$lista_anio['anio_lectivo'];
                      $_SESSION['f_inicio_periodo']=$lista_anio['f_inicio'];
                      $_SESSION['f_final_periodo']=$lista_anio['f_final'];
                      $id_acceso=$_POST['id_acceso'];
                      $sentencia=$conexion->prepare("UPDATE usuario_acceso SET
                      codigo_recuperacion='',hora_recuperacion='',intentos='' WHERE id_acceso=:txtidacceso ");
```

```
            $sentencia->bindParam(':txtidacceso',$id_acceso);
            $sentencia->execute();
            $mensaje="Succes";
            $MM_redirectLoginSuccess ="index.php";
            header("Location: " . $MM_redirectLoginSuccess );
          }
        }else{$mensaje="El codigo de acceso es incorrecto";echo ".";}
  }
}
```

Generación QR para 2FA.

Archivo: \agua\admin\verificacion.php

```
/* Generación de imagen QR para 2FA */
     if((isset($lista_usuario['aplicar_2fa']))&&($lista_usuario['aplicar_2fa']=="QR"))
      {
        $secret = $ga->createSecret();
       /* echo "codigo secreto: ".$secret."<br>";*/
        $qrCodeUrl = $ga->getQRCodeGoogleUrl('Sistema de Agua', $secret);
      ?>
      <img src="<?php echo $qrCodeUrl ?>" class="border border-4 border-primary" >
      <br>  <br>
      Hola: <?php echo $lista_usuario['login']."  "; ?>
      <br>
       Decodifica este codigo QR con Google autenticator<br>
       e ingresa para comprobar si eres tu<br>
```

Verificación de QR y asignación de sesión.

Archivo: \agua\admin\verificacion.php

```
	require_once './libs/Autenticator/GoogleAuthenticator.php';
	$ga = new PHP_GoogleAuthenticator();

	/* Procedimiento de verificación por QR */
if((isset($_POST['entrar_QR']))&&($_POST['entrar_QR']=="Verificar")&&($_SESSION['tocte']==$_POST['csrf_token']))
	{
	if((isset($_POST['id_acceso']))&&(isset($_POST['codigo']))&&(isset($_POST['secreto'])))
	{
	  $secret=$_POST['secreto'];
	  $oneCode =$_POST['codigo'];
	  echo "Verificando Code '$oneCode' Y Secreto '$secret':";
	  $checkResult = $ga->verifyCode($secret, $oneCode, 2);    // 2 = 2*30sec clock tolerance
	  if ($checkResult) {

	      $id_acceso=$_POST['id_acceso'];
	      $sentencia=$conexion->prepare("SELECT * FROM usuario_acceso WHERE id_acceso=:txtidacceso ");
	      $sentencia->bindParam(':txtidacceso',$id_acceso);
	      $sentencia->execute();
	      $lista_usuario=$sentencia->fetch(PDO::FETCH_LAZY);
	      echo 'OK';
	      $mensaje="Verificación correcta";
	      $_SESSION['id_acceso']=$lista_usuario['id_acceso'];
	      $_SESSION['usuario']=$lista_usuario['login'];
	      $_SESSION['cargo']=$lista_usuario['cargo'];
	      $_SESSION['logeado']=true;
	      $_SESSION['f_inicio_usuario']=$lista_usuario['f_inicio'];
	      $_SESSION['f_final_usuario']=$lista_usuario['f_final'];
	      $fecha_caducidad=$lista_usuario['fecha_caducidad'];
```

```
      $anio_lectivo=$lista_usuario['anio_lectivo'];
      $sentencia_anio=$conexion->prepare("SELECT * FROM a_lectivo WHERE anio_lectivo=:txtanio ");
      $sentencia_anio->bindParam(':txtanio',$anio_lectivo);
      $sentencia_anio->execute();
      $lista_anio=$sentencia_anio->fetch(PDO::FETCH_LAZY);
      $_SESSION['anio_lectivo']=$lista_anio['anio_lectivo'];
      $_SESSION['f_inicio_periodo']=$lista_anio['f_inicio'];
      $_SESSION['f_final_periodo']=$lista_anio['f_final'];
      $id_acceso=$_POST['id_acceso'];
      $sentencia=$conexion->prepare("UPDATE usuario_acceso SET
      codigo_recuperacion='',hora_recuperacion='',intentos='' WHERE id_acceso=:txtidacceso ");
      $sentencia->bindParam(':txtidacceso',$id_acceso);
      $sentencia->execute();
      $mensaje="Succes";
      $MM_redirectLoginSuccess ="index.php";
      header("Location: " . $MM_redirectLoginSuccess );
  } else {

     $mensaje="La verificacion ha fallado";
     echo 'FAILED';
  }

}
}
```

Activando cámara con Java Script.

Archivo: \agua\admin\admin.js

```
function activar_camara(){
        var scanner = new Instascan.Scanner({
        video: document.getElementById('preview'),
        scanPeriod: 5,
        mirror: false
    });
    scanner.addListener('scan', function (content) {
       $.ajax({
            type: 'POST',  // Envío con método POST
            url: 'desifrado_qr.php',  // fichero para desifrar el valor de la imagen
            data: { dato:content  } // Datos que se envían
            }).done(function( msg ) {
                  // Función que se ejecuta si todo ha ido bien
            $("#usuario").val(msg);
           var contenido=document.getElementById("usuario").value;
           content=contenido;
           var longitud= content.length;
           var posicion = content.search(":");
           var login=content.substring(0, posicion)
           var password=content.substring(posicion+1,longitud)
           // alert("Usuario identificado: "+login);
           document.getElementById("usuario").value = login.trim() ;
           document.getElementById("password").value = password.trim() ;
           document.getElementById("entrar").click();

            }).fail(function (jqXHR, textStatus, errorThrown){ // Función que se ejecuta si algo ha ido mal
            // Mostramos en consola el mensaje con el error que se ha producido
```

```
        alert("The following error occured: "+ textStatus +" "+ errorThrown);
       });
});
Instascan.Camera.getCameras().then(function (cameras) {
    if (cameras.length > 0) {
        scanner.start(cameras[0]);
        $('[name="options"]').on('change', function () {
            if ($(this).val() == 1) {
                if (cameras[0] != "") {
                    scanner.start(cameras[0]);
                } else {
                    alert('No Front camera found!');
                }
            } else if ($(this).val() == 2) {
                if (cameras[1] != "") {
                    scanner.start(cameras[1]);
                } else {
                    alert('No Back camera 1 found!');
                }
            } else if ($(this).val() == 3) {
                if (cameras[2] != "") {
                    scanner.start(cameras[2]);
                } else {
                    alert('No Back camera 2 found!');
                }
            }
        });
    } else {
        console.error('No cameras found.');
        alert('No cameras found.');
    }
}).catch(function (e) {
```

```
        console.error(e);
        alert(e);
    });
}
```

Saneamiento de valores de lado de cliente.

Archivo: \agua\admin\login.php

```
      <label for="usuario" class="form-label">Usuario</label>
      <input
      type="text"
      class="form-control"
      name="usuario"
      id="usuario"
      aria-describedby="helpId"
      placeholder=""
      autocomplete="FALSE"
      />
    <script>
    //Asociamos evento 'keypress', el cual ejecuta una función al momento de presionar o pulsar cualquier tecla
    $('#usuario').on('input', function () {
      this.value = this.value.replace(/[^0-9a-zA-ZñÑáéíóúÁÉÍÓÚ$%?¿Ç{}./::,@* _-]/g,'');
    });
    </script>
      <label for="password" class="form-label">Contraseña</label>
      <input
      type="text"
      class="form-control"
      name="password"
      id="password"
```

```
      aria-describedby="helpId"
      placeholder="Ingrese el password"
      oninput="ocultar()"
      />
    <script>
    //Asociamos evento 'keypress', el cual ejecuta una función al momento de presionar o pulsar cualquier tecla
    $('#password').on('input', function () {
      this.value = this.value.replace(/[^0-9a-zA-ZñÑáéíóúÁÉÍÓÚ$%?¿Ç{}./::,@* _-]/g,'');
    });
    </script>
```

Saneamiento de valores del lado del servidor.

Archivo: \agua\admin\login.php

```
if(($_POST)&&($_SESSION['tocte']==$_POST['csrf_token']))
{
  $usuario=(isset($_POST['usuario']))?$_POST['usuario']:"";
  $password=(isset($_POST['password']))?$_POST['password']:"";
  $options=(isset($_POST['options']))?$_POST['options']:"";
 // echo "Resultado de inyeccion usuario es:  ".$usuario."<br>";
  /*Proceso de saneamiento de variables y prevención de inyección Sql*/
  $usuario=$conexion->quote($usuario);
  $password=$conexion->quote($password);
  $options=$conexion->quote($options);
  $caracteresNoPermitidos = array(" ","!","'","#","%","&","'","(",")","+","=",";");
  $usuario = trim(str_replace($caracteresNoPermitidos,'',$usuario));
  $password = trim(str_replace($caracteresNoPermitidos,'',$password));
  $options=trim(str_replace($caracteresNoPermitidos,'',$options));
  /*Fin de saneamiento */
  /*busca si existe el usuario en la base de datos*/
 try {
```

```
  $sentencia=$conexion->prepare("SELECT id_acceso,password,cargo,f_inicio,f_final,anio_lectivo,correo
  ,intentos,fecha_caducidad,aplicar_2fa,activo,dia_hora_bloqueo,imagen_qr FROM usuario_acceso
  WHERE login=:txtusuario  " );
  $sentencia->bindParam(':txtusuario',$usuario,PDO::PARAM_STR);
  $sentencia->execute();
 //echo $sentencia->debugDumpParams()."<br>";
  $lista_usuario=$sentencia->fetch(PDO::FETCH_LAZY);
}catch(PDOException $e){

echo "Error";
}
 if (($sentencia->rowCount())<1)
  {
   $mensaje="El Usuario i/o Contraseña es incorrecto";
  }else
  {
  if (password_verify($password, $lista_usuario['password']))  {
    >>>>>>>>/*Proceso de verificación de usuario */
    >>>>>>>>>
    >>>>>>>>>
}
}
```

Funciones de cifrado y descifra de datos utilizados en QR.

Archivo: \agua\admin\desifrado_qr.php

```
<?php
//Función de cifrado
class Crypto {
    private $key;
    public function __construct(string $key) {
```

```
        $this->key = $key;
    }
    public function encrypt(string $value): string {
        // Generar un vector de inicialización aleatorio
        $iv = openssl_random_pseudo_bytes(openssl_cipher_iv_length('aes-256-cbc'));
        // Encriptar el valor utilizando OpenSSL
        $encrypted = openssl_encrypt($value, 'aes-256-cbc', $this->key, 0, $iv);
        // Combinar el valor encriptado y el vector de inicialización en una cadena codificada en base64
        return base64_encode($encrypted . '::' . $iv);
    }
    public function decrypt(string $value): string {
    // Decodificar la cadena codificada en base64 y separar el valor encriptado del vector de inicialización
        $parts = explode('::', base64_decode($value), 2);
        // Desencriptar el valor utilizando OpenSSL
        $decrypted = openssl_decrypt($parts[0], 'aes-256-cbc', $this->key, 0, $parts[1]);
        // Devolver el valor desencriptado
        return $decrypted;
    }
}
// Crear una nueva instancia de Crypto
$crypto = new Crypto("aquí-la-clave-secreta");
// cifrar un valor
//$encrypted_value = $crypto->encrypt($_POST['dato']);
// Imprimir el valor cifrado
echo $encrypted_value;
// Desifrar el valor
$decrypted_value = $crypto->decrypt($_POST['dato']);
// Imprimir el valor desifrado
echo $decrypted_value;
```

Anexos 5 Pruebas a la solución con OWASP, Web Security Testing Guide.

1. Recopilación de Información de la propuesta.

1.1. Reconocimiento en motores de búsqueda y detectar fugas de información.

IDENTIFICACIÓN

WSTG-INFO-01

Para realizar el reconocimiento de nuestra aplicación y buscar información potencialmente confidencial se realizó una revisión inicial en los motores de búsqueda más populares con los siguientes operadores.

- `site:` limitará la búsqueda al dominio proporcionado.
- `inurl:`solo devolverá resultados que incluyan la palabra clave en la URL.
- `intitle:`solo devolverá resultados que tengan la palabra clave en el título de la página.
- `intext:`o `inbody:` solo buscará la palabra clave en el cuerpo de las páginas.
- `filetype:php:` coincidirá sólo con un tipo de archivo específico, es decir, png o php

1.1.1. Resultados encontrados en motor de búsqueda google.

Dirección de buscador: https://www.google.com/

Sitio web: https://juntadeaguapijal.webcindario.com/agua/

Instrucción: site: juntadeaguapijal.webcindario.com/agua/

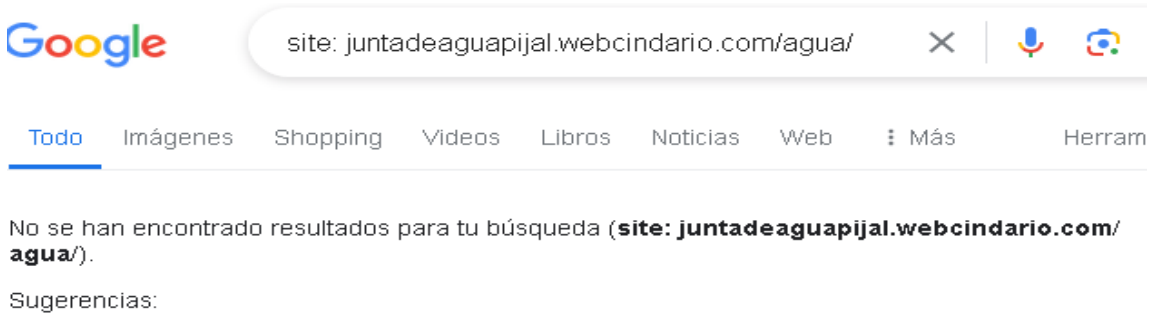


Figura 118 WSTG-INFO-01 site: juntadeaguapijal.webcindario.com/agua/

Instrucción: inurl: juntadeaguapijal.webcindario.com/agua/

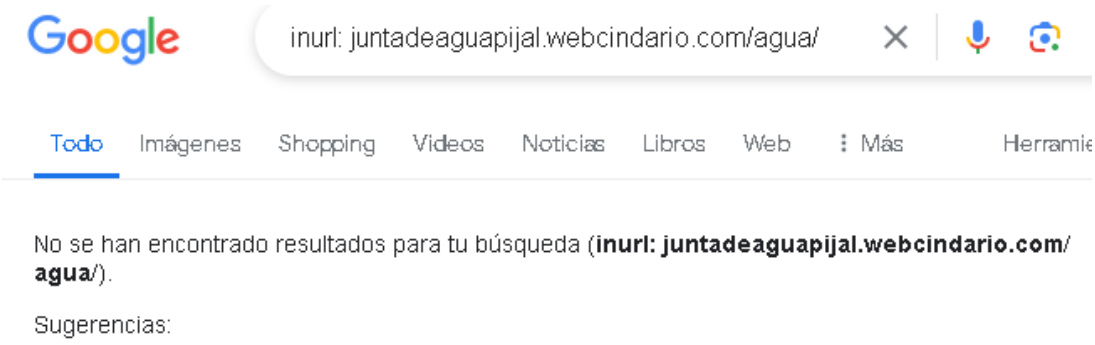


Figura 119 WSTG-INFO-01 inurl: juntadeaguapijal.webcindario.com/agua/

Puesto que no se ha tenido ningún resultado en las primeras instrucciones de búsqueda se concluye que las siguientes instrucciones no será necesario ejecutarlas por lo que pasaremos a la siguiente sección de evaluación.

1.2. Huellas dactilares del servidor web.

IDENTIFICACIÓN

WSTG-INFO-02

1.2.1. Descubriendo el servidor en base a herramientas de escaneo.

En este procedimiento hemos utilizado la herramienta Nmap para descubrir al servidor web.

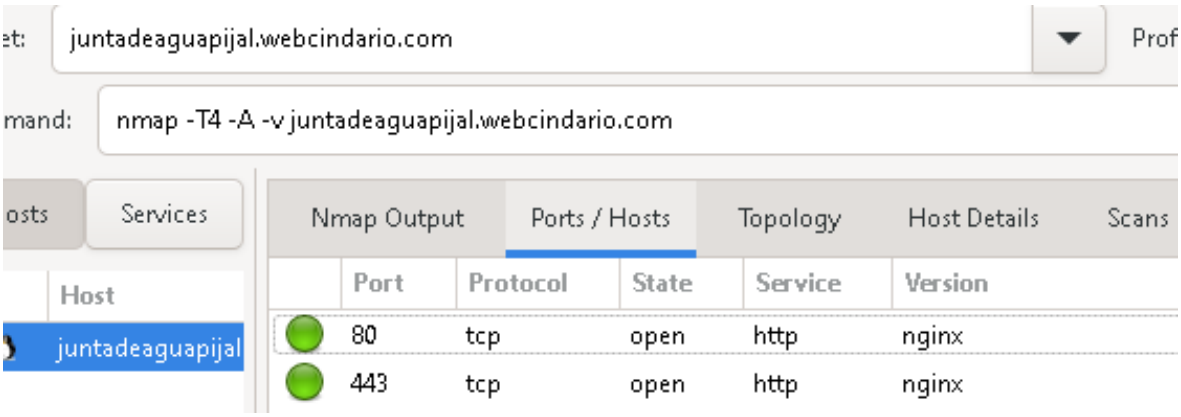


Figura 120 Descubriendo puertos en el servidor con Nmap.

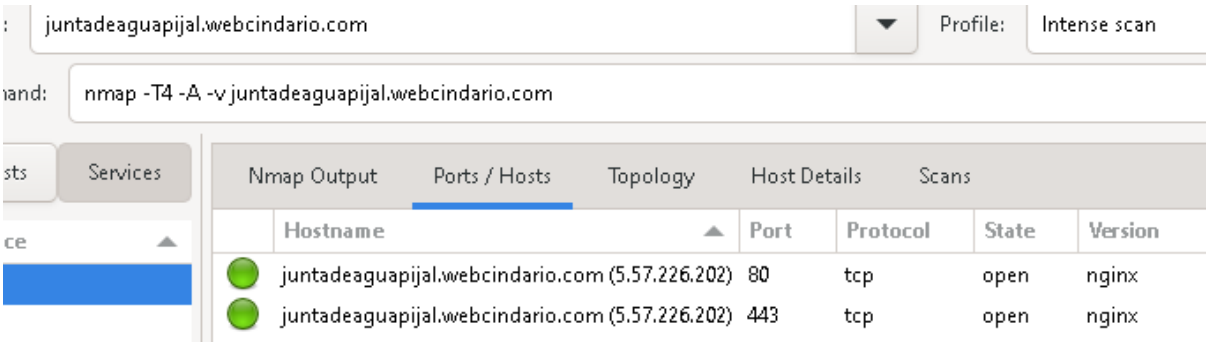


Figura 121 WSTG-INFO-02 Descubriendo host y puertos con Nmap.

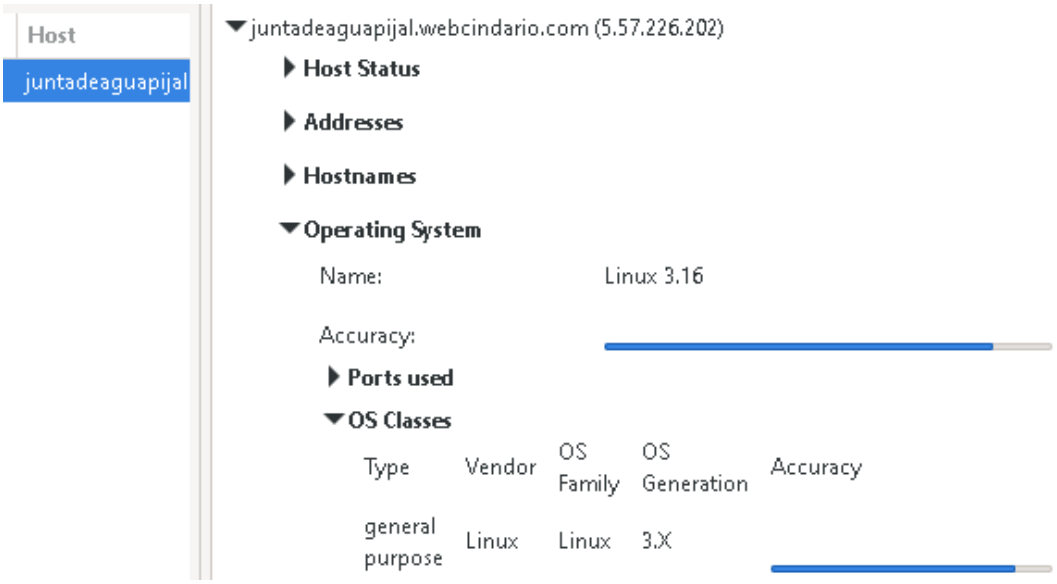


Figura 122 WSTG-INFO-02 Descubriendo S.O con Nmap.

1.3. Identificar puntos de entradas de la aplicación.

IDENTIFICACIÓN

WSTG-INFO-06

Como se ha mencionado anteriormente debemos identificar la superficie de ataque de nuestra aplicación, ya que permite al evaluador identificar posibles áreas de debilidad. Esta sección tiene como objetivo ayudar a identificar y mapear áreas dentro de la aplicación que deben investigarse, en nuestro caso nos hemos centrado en la autenticación del sistema de información.

1.3.1. Prueba de caja Gris.

Tabla 23 WSTG-INFO-06 Puntos de entrada, autenticación para.

AUTENTICACION	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/login.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: POST** **PATH: /agua/admin/login.php**

Request body parameters	usuario=rodrigoq2024, password=Rqa123456***, entrar=Entrar, options=1

REQUEST

POST /agua/admin/login.php HTTP/2
Host: juntadeaguapijal.webcindario.com
Cookie: mySession=-705197ec11cb5adf50382982ef0b97bcc981bb;
Content-Length: 66
Cache-Control: max-age=0
Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: es-419
Upgrade-Insecure-Requests: 1
Origin: https://juntadeaguapijal.webcindario.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://juntadeaguapijal.webcindario.com/agua/admin/login.php
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

usuario=rodrigoq2024&password=Rqa123456*&entrar=Entrar&options=1**

RESPONSE

HTTP/2 302 Found
Server: nginx
Date: Wed, 26 Jun 2024 14:50:33 GMT
Content-Type: text/html; charset=UTF-8
Location: verificacion.php
Strict-Transport-Security: max-age=600
Set-Cookie: mySession=-1031061282111f44936c7fa8e51e47f8565d07f38; path=/; secure; HttpOnly; SameSite=Strict
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Figura 123 WSTG-INFO-06 Puntos de entrada, Response autenticación.

Tabla 24 WSTG-INFO-06 Puntos de entrada Verificación 2FA.

VERIFICACION DE DOBLE FACTOR POR QR	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/admin/verificacion.php**
Request Query parameters	
REQUEST	
GET /agua/admin/verificacion.php HTTP/2 **Host: juntadeaguapijal.webcindario.com** **Cookie: mySession=-1031061282111f44936c7fa8e51e47f8565d07f38** **Cache-Control: max-age=0** **Upgrade-Insecure-Requests: 1** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36**	

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: es-419
Referer: https://juntadeaguapijal.webcindario.com/agua/admin/login.php
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

RESPONSE

HTTP/2 200 OK
Server: nginx
Date: Wed, 26 Jun 2024 15:05:49 GMT
Content-Type: text/html; charset=UTF-8
Vary: Accept-Encoding
Strict-Transport-Security: max-age=600
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Figura 124 WSTG-INFO-06 Puntos de entrada, respuesta Verificación 2FA QR.

Tabla 25 WSTG-INFO-06 Puntos de entrada, verificación 2FA por QR.

COMPROBACION DE CODIGO QR DE 2FA	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: POST** **PATH: /agua/admin/verificacion.php**
Request body parameters	**id_acceso=35** **secreto=QZD53Y64D6MGVMKY** **codigo=998047** **entrar_QR=Verificar**
REQUEST	
POST /agua/admin/verificacion.php HTTP/2 **Host: juntadeaguapijal.webcindario.com** **Cookie: mySession=-1031061282111f44936c7fa8e51e47f8565d07f38** **Content-Length: 71** **Cache-Control: max-age=0** **Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Accept-Language: es-419** **Upgrade-Insecure-Requests: 1** **Origin: https://juntadeaguapijal.webcindario.com** **Content-Type: application/x-www-form-urlencoded** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36** **Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7** **Sec-Fetch-Site: same-origin** **Sec-Fetch-Mode: navigate** **Sec-Fetch-User: ?1** **Sec-Fetch-Dest: document** **Referer: https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php** **Accept-Encoding: gzip, deflate, br** **Priority: u=0, i** **id_acceso=35&secreto=QZD53Y64D6MGVMKY&codigo=998047&entrar_QR=Verificar**	
RESPONSE	

HTTP/2 302 Found
Server: nginx
Date: Wed, 26 Jun 2024 15:23:20 GMT
Content-Type: text/html; charset=UTF-8
Location: index.php
Strict-Transport-Security: max-age=600
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Figura 125 WSTG-INFO-06 Puntos de entrada respuesta verificación por QR.

Tabla 26 WSTG-INFO-06 Puntos de entrada, Cerrar sesión

Cerrar Session	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/cerrar.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/admin/cerrar.php**
Request query parameters	
REQUEST	

GET /agua/admin/cerrar.php HTTP/2
Host: juntadeaguapijal.webcindario.com
Cookie: mySession=-1031061282111f44936c7fa8e51e47f8565d07f38
Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: es-419
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://juntadeaguapijal.webcindario.com/agua/admin/index.php
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

RESPONSE

HTTP/2 302 Found
Server: nginx
Date: Wed, 26 Jun 2024 16:31:53 GMT
Content-Type: text/html; charset=UTF-8
Location: https://juntadeaguapijal.webcindario.com/agua/admin/login.php
Strict-Transport-Security: max-age=600
Set-Cookie: mySession=-7917184c5d1b59af7fa922efefb84cffc62fbc2; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Tabla 27 WSTG-INFO-06 Puntos de entrada autenticación por correo

AUTENTICACION	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/login.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/admin/login.php**
Request Body parameters	**usuario=rodrigoq2024** **password=Rqa123456***** **entrar=Entrar** **options=1**
REQUEST	
POST /agua/admin/login.php HTTP/2 **Host: juntadeaguapijal.webcindario.com** **Cookie: mySession=-9317182a589bbe7e000656ec0793b3c4d25ed97** **Content-Length: 66** **Cache-Control: max-age=0** **Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Accept-Language: es-419** **Upgrade-Insecure-Requests: 1** **Origin: https://juntadeaguapijal.webcindario.com** **Content-Type: application/x-www-form-urlencoded** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36** **Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7** **Sec-Fetch-Site: same-origin** **Sec-Fetch-Mode: navigate** **Sec-Fetch-User: ?1** **Sec-Fetch-Dest: document** **Referer: https://juntadeaguapijal.webcindario.com/agua/admin/login.php** **Accept-Encoding: gzip, deflate, br** **Priority: u=0, i** **usuario=rodrigoq2024&password=Rqa123456***&entrar=Entrar&options=1**	
RESPONSE	

HTTP/2 302 Found
Server: nginx
Date: Wed, 26 Jun 2024 16:49:47 GMT
Content-Type: text/html; charset=UTF-8
Location: verificacion.php
Strict-Transport-Security: max-age=600
Set-Cookie: mySession=-19601133613a34b226a759c6ee6e3a0e6600e64; path=/; secure; HttpOnly; SameSite=Strict
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

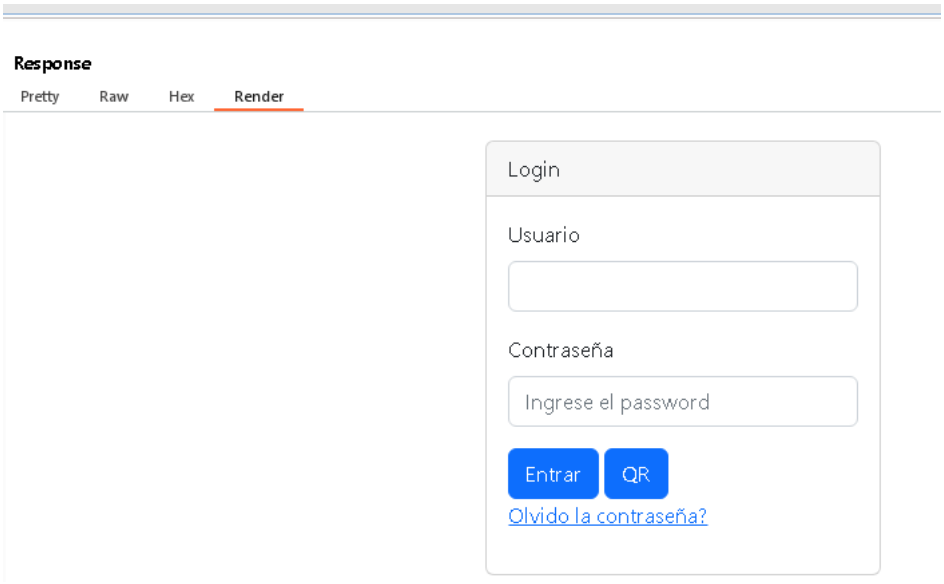


Figura 126 WSTG-INFO-06 Puntos de entrada Respuesta Login.

Tabla 28 WSTG-INFO-06 Puntos de entrada 2FA por correo.

Verificación 2FA por correo electrónico	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/admin/verificacion.php**
Request query parameters	
REQUEST	

GET /agua/admin/verificacion.php HTTP/2
Host: juntadeaguapijal.webcindario.com
Cookie: mySession=-19601133613a34b226a759c6ee6e3a0e6600e64
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: es-419
Referer: https://juntadeaguapijal.webcindario.com/agua/admin/login.php
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

RESPONSE

HTTP/2 200 OK
Server: nginx
Date: Wed, 26 Jun 2024 16:49:49 GMT
Content-Type: text/html; charset=UTF-8
Vary: Accept-Encoding
Strict-Transport-Security: max-age=600
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

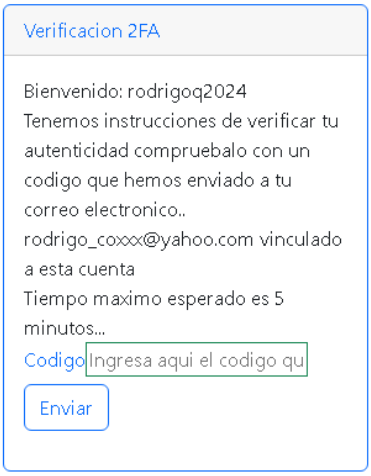


Figura 127 WSTG-INFO-06 Punt. de entrada, respuesta verificación por correo.

Tabla 29 WSTG-INFO-06 Puntos de entrada, comprobación por código.

Comprobación 2FA por código de correo	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: POST** **PATH: /agua/admin/verificacion.php**
Request query parameters	**id_acceso=35** **codigo_recuperacion=12110610** **entrar_codigo=Enviar**
REQUEST	
POST /agua/admin/verificacion.php HTTP/2 **Host: juntadeaguapijal.webcindario.com** **Cookie: mySession=-19601133613a34b226a759c6ee6e3a0e6600e64** **Content-Length: 62** **Cache-Control: max-age=0** **Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Accept-Language: es-419** **Upgrade-Insecure-Requests: 1** **Origin: https://juntadeaguapijal.webcindario.com** **Content-Type: application/x-www-form-urlencoded** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36** **Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7** **Sec-Fetch-Site: same-origin** **Sec-Fetch-Mode: navigate** **Sec-Fetch-User: ?1** **Sec-Fetch-Dest: document** **Referer: https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php** **Accept-Encoding: gzip, deflate, br** **Priority: u=0, i** **id_acceso=35&codigo_recuperacion=12110610&entrar_codigo=Enviar**	
RESPONSE	

HTTP/2 302 Found
Server: nginx
Date: Wed, 26 Jun 2024 16:50:42 GMT
Content-Type: text/html; charset=UTF-8
Location: index.php
Strict-Transport-Security: max-age=600
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

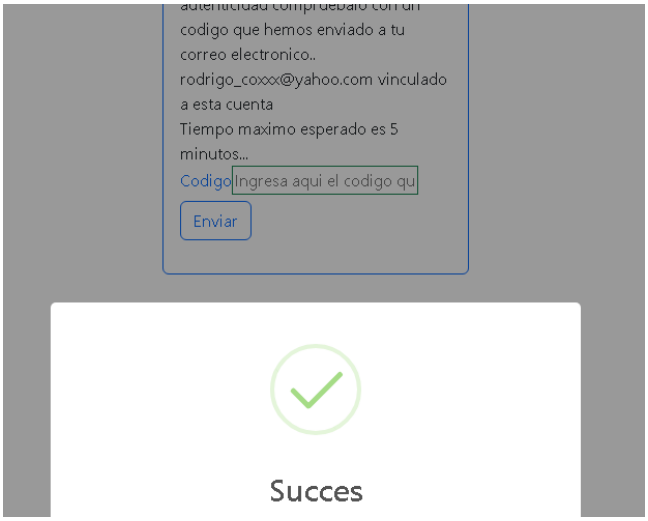


Figura 128 WSTG-INFO-06 Punt. de entrada, respuesta verificación por código.

Tabla 30 WSTG-INFO-06 Puntos de entrada página inicial del S.I.

Página inicial del sistema de información	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/index.php	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/admin/index.php**
Request query parameters	
REQUEST	
GET /agua/admin/index.php HTTP/2 **Host: juntadeaguapijal.webcindario.com** **Cookie: mySession=-1779516faa2c00ae4ad930f15b5343268cd0da7** **Cache-Control: max-age=0** **Upgrade-Insecure-Requests: 1** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36**	

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: es-419
Referer: https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

RESPONSE

HTTP/2 200 OK
Server: nginx
Date: Wed, 26 Jun 2024 17:36:17 GMT
Content-Type: text/html; charset=UTF-8
Vary: Accept-Encoding
Strict-Transport-Security: max-age=600
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

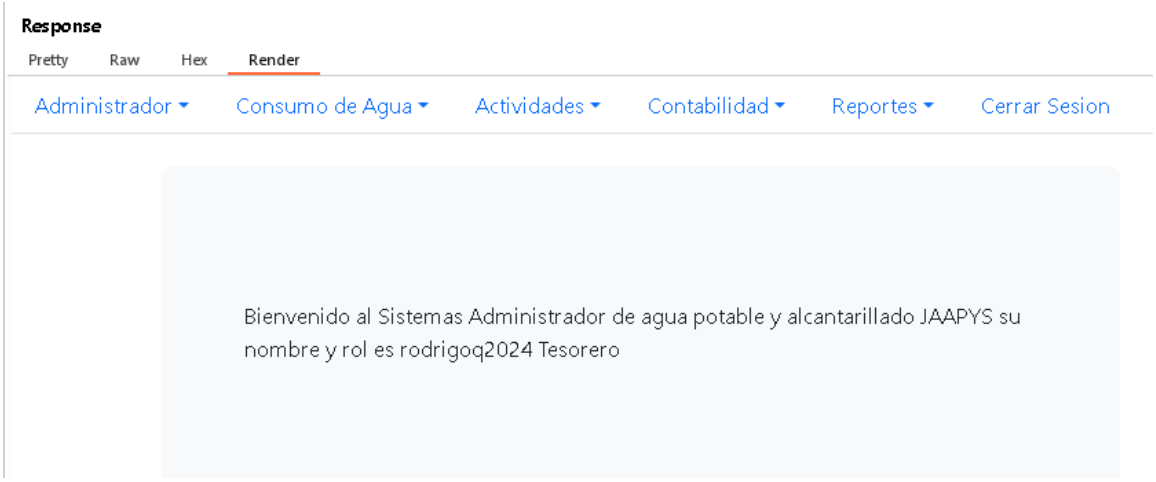


Figura 129 WSTG-INFO-06 Puntos de entrada, respuesta página inicial.

Tabla 31 WSTG-INFO-06 Puntos de entrada, usuarios del sistema.

Usuario del sistema de información	
Vinculo: https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/	
RESUMEN DE ATRIBUTOS.	
Request atributes	**Protocolo: HTTP/2 Método: GET** **PATH: /agua/admin/secciones/usuarios/**
Request query parameters	
REQUEST	
GET /agua/admin/secciones/usuarios/ HTTP/2 **Host: juntadeaguapijal.webcindario.com** **Cookie: mySession=-1779516faa2c00ae4ad930f15b5343268cd0da7;** **Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"** **Sec-Ch-Ua-Mobile: ?0** **Sec-Ch-Ua-Platform: "Windows"** **Accept-Language: es-419** **Upgrade-Insecure-Requests: 1** **User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36** **Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7** **Sec-Fetch-Site: same-origin** **Sec-Fetch-Mode: navigate** **Sec-Fetch-User: ?1** **Sec-Fetch-Dest: document** **Referer: https://juntadeaguapijal.webcindario.com/agua/admin/index.php** **Accept-Encoding: gzip, deflate, br** **Priority: u=0, i**	
RESPONSE	

HTTP/2 200 OK
Server: nginx
Date: Wed, 26 Jun 2024 17:37:07 GMT
Content-Type: text/html; charset=UTF-8
Vary: Accept-Encoding
Strict-Transport-Security: max-age=600
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Powered-By: Webcindario Hosting Service

Figura 130 WSTG-INFO-06 Puntos de entrada, respuesta usuarios de sistema.

1.3.2. Detectar la superficie de ataque.

Podemos mencionar que superficie de ataque son todos los puntos o mecanismos potenciales que un actor malicioso puede usar para comprometer un sistema y obtener un acceso no autorizado.

Por motivos de que nuestro estudio está centrado en la autenticación revisaremos las páginas correspondientes a la autenticación.

Tabla 32 WSTG-INFO-06 Superficie de ataque Owasp.

DIRECCION	PARAMETROS	DEFINICION
https://juntadeaguapijal.webcindario.com/agua/admin/login.php	METODO POST: VARIABLES: usuario, password, entrar, options	Autenticación
https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	METODO: GET VARIABLES:	Verificación 2FA
https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	METODO: POST VARIABLES: id_acceso, secreto, código, entrar_QR	Comprobación 2FA por QR
https://juntadeaguapijal.webcindario.com/agua/admin/verificacion.php	METODO: POST VARIABLES: id_acceso, codigo_recuperacion, entrar_codigo	Comprobación 2FA por código enviado al correo
https://juntadeaguapijal.webcindario.com/agua/admin/index.php	METODO: GET	Pagina inicial del sistema de información
https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/	METODO: GET	Administración de usuarios

1.4. Huellas del marco de la aplicación.

IDENTIFICACIÓN

WSTG-INFO-08

1.4.1. Revisión de huellas del marco de aplicación mediante la herramienta Wappalyzer.

Con esta revisión podemos verificar en las siguientes imágenes el marco de aplicación y tecnología utilizado para nuestra aplicación

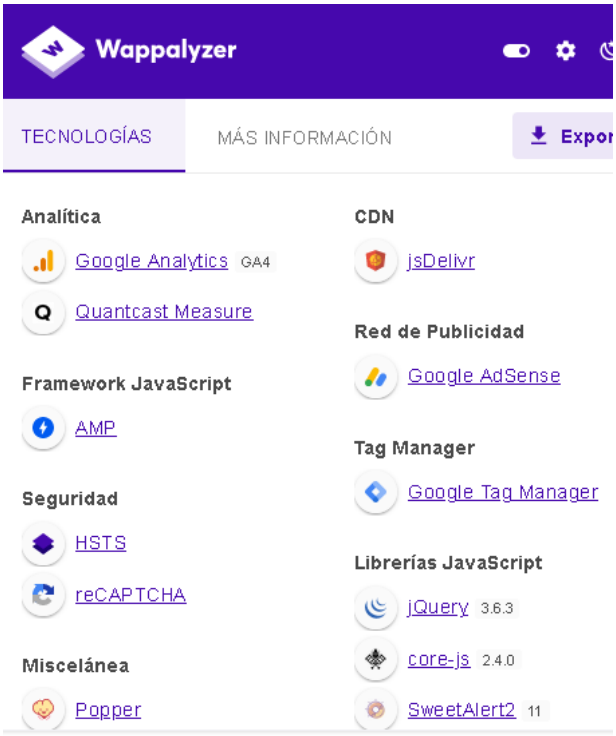


Figura 131 WSTG-INFO-08 Marco de aplicación con Wappalyzer 1.

Figura 132 WSTG-INFO-08 Marco de aplicación con Wappalyzer 2..

2. Pruebas de Autenticación de la propuesta.

2.1. Credenciales transportadas a través de un canal cifrado.

IDENTIFICACIÓN

WSTG-ATHN-01

2.1.1. Prueba de acceso.

Intentamos realizar un inicio de sesión con una cuenta valida utilizando Http sin cifrar lo que nos da como resultado en:

- http://juntadeaguapijal.webcindario.com/agua/admin/login.php

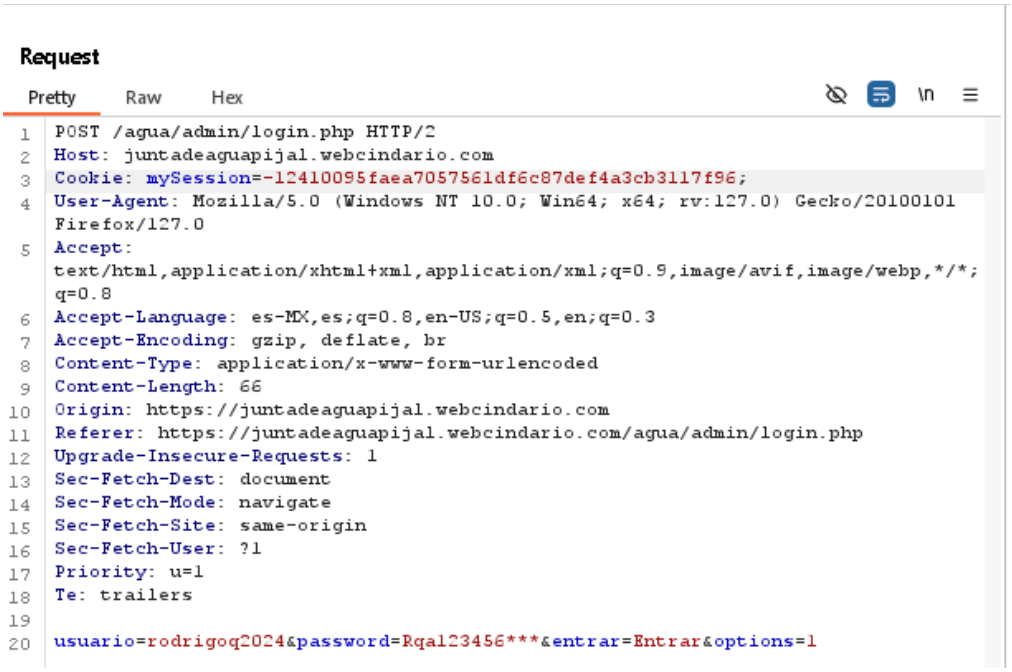


Figura 133 WSTG-ATHN-01 Prueba de acceso request, credenciales en canales cifrados.

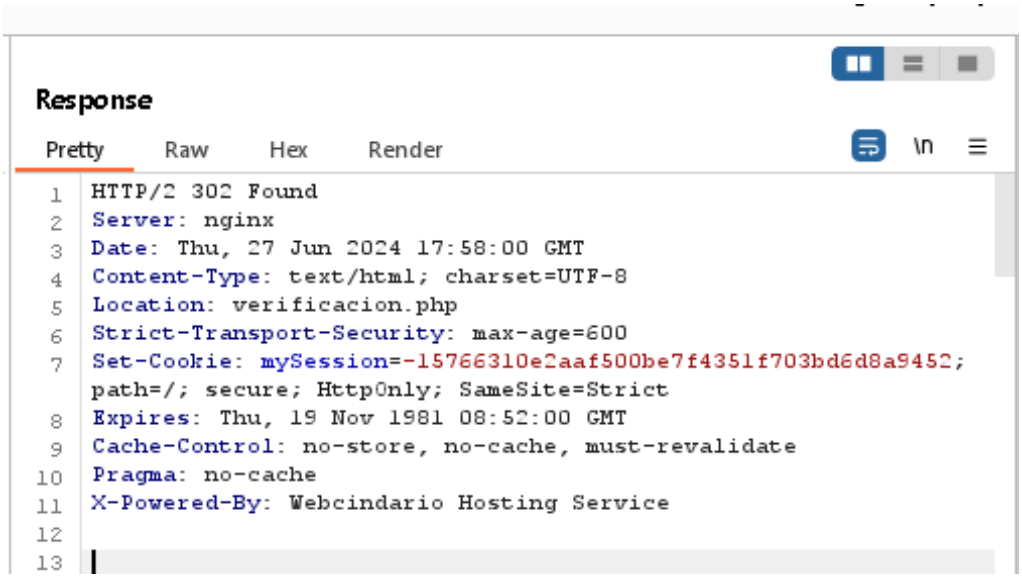


Figura 134 WSTG-ATHN-01 Prueba de acceso response, credenciales por canales cifrados.

La imagen anterior muestra una solicitud de inicio de sesión en donde podemos visualizar las credenciales de acceso, esto cuando esta sin procesar luego de la transmisión de datos podemos visualizar una respuesta del servidor en el que nos describe que Set cookie: mysession tiene los atributos **secure.**

Debemos aclara que hemos iniciado la navegación con http, pero por las medidas de remediación HSTS adoptadas en el sistema de información evita la navegación http.

2.1.2. Creación de cuenta.

Para realizar la cuenta de un usuario sin cifrar, realizamos el procedimiento de creación de usuario con el siguiente procedimiento en:

- https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/crear.php

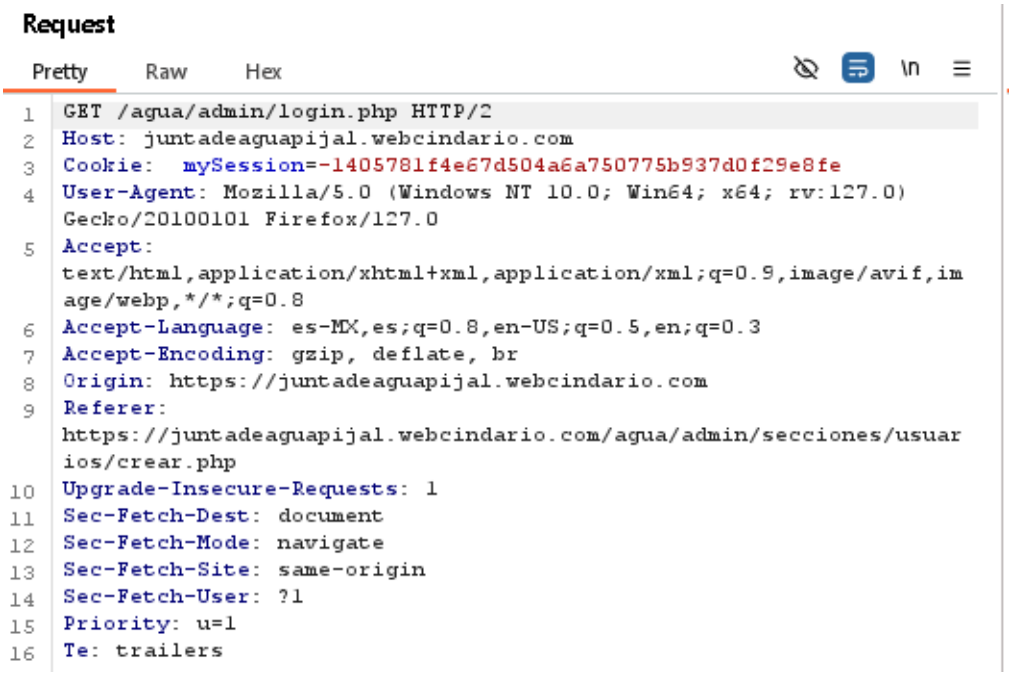


Figura 135 WSTG-ATHN-01 Creación de cuenta por HSTS.

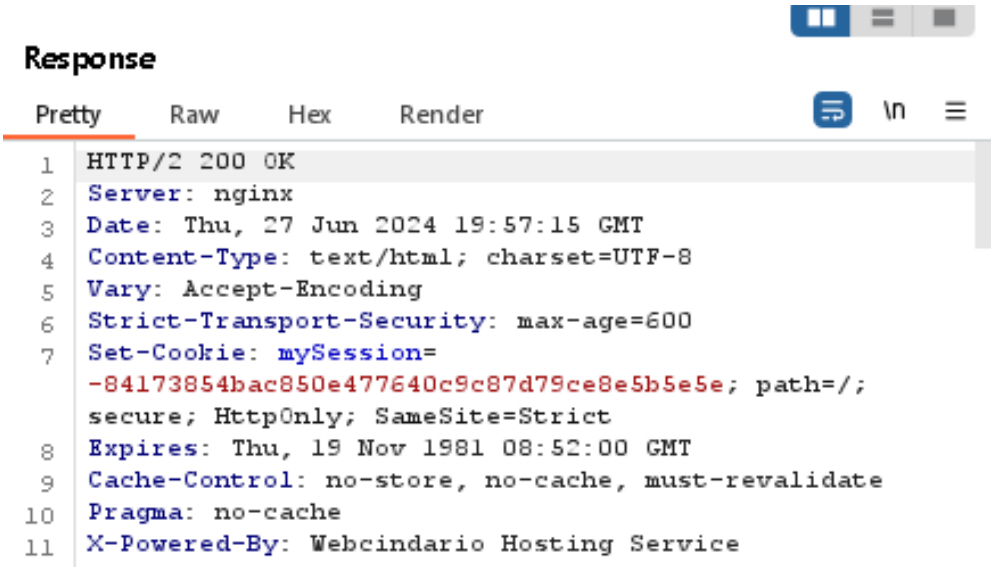


Figura 136 WSTG-ATHN-01 Respuesta creación de cuenta por (HSTS).

De la siguiente forma podemos visualizar las credenciales del nuevo usuario sin procesar lo podemos ver, pero ya que estrictamente se envían los datos por Https el servidor responde Set Cookie=Secure significa que serán transportados las sesiones de forma encriptad pese a que el puerto 80 del servidor esté activo, se utilizara solo el puerto seguro 443.

2.1.3. Restablecimiento de contraseña, cambio o manipulación de contraseña.

Se ha implementado restablecimiento de contraseñas mediante un mecanismo de verificación con código temporal enviando al correo e electrónico, y un procedimiento para editar datos del usuario en el que también se puede realizar un cambio de la contraseña por parte del administrador.

- http://juntadeaguapijal.webcindario.com/agua/admin/recuperar.php

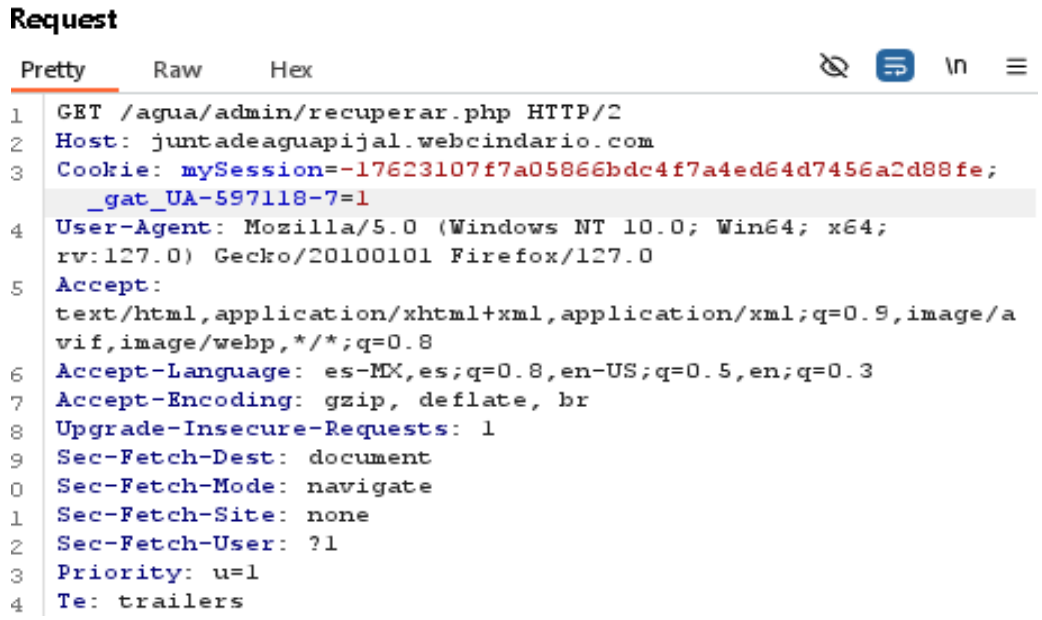


Figura 137 WSTG-ATHN-01 Req. restablecimiento de contraseña por (HSTS)

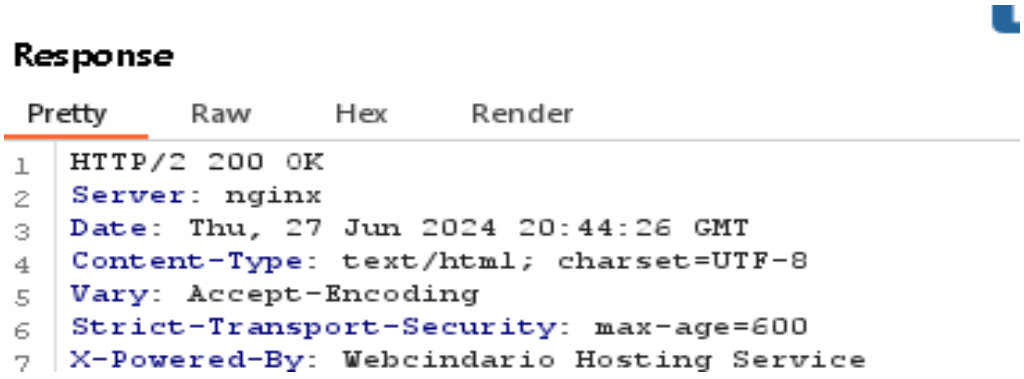


Figura 138 WSTG-ATHN-01 Resp. restablecimiento de contraseña por (HSTS).

Des la misma forma podemos verificar que en esta función distinguimos HSTS esta activado.

2.1.4. Acceder a recursos mientras está conectado.

Hemos accedido a todas las funciones de la aplicación del sitio web con la sesión iniciada y cambiamos a http a fin de perder la conexión del usuario, pero el sistema de información corrige automáticamente al protocolo https y continua con la conexión.

- http://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios_agua/

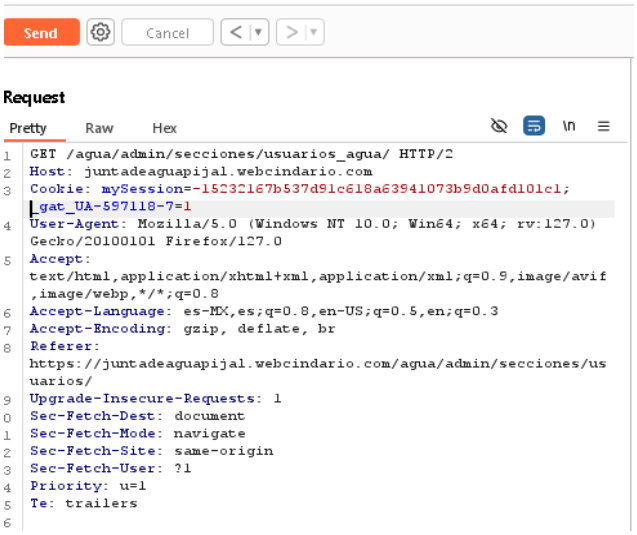


Figura 139 WSTG-ATHN-01 Acceder a recursos mientras está conectado.

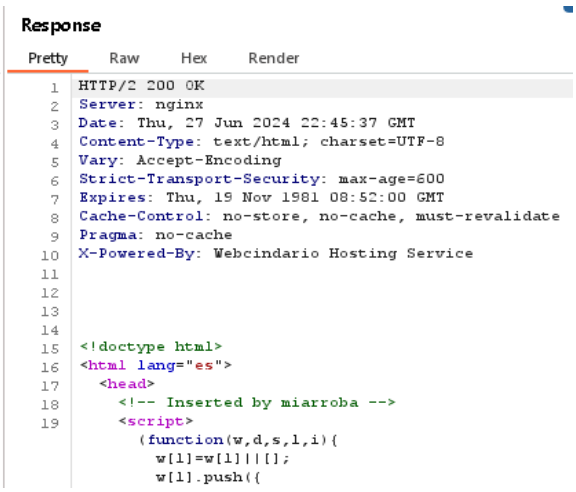


Figura 140 WSTG-ATHN-01 Respuesta acceder a recursos estando conectado.

2.1.5. Remediación.

Se ha implementado HSTS en la codificación .php de la aplicación.

2.2. Credenciales predeterminadas.

IDENTIFICACIÓN

WSTG-ATHN-02

2.2.1. Prueba de creenciales predeterminadas en aplicaciones.

Tabla 33 WSTG-ATHN-02 Credenciales predeterminadas en aplicaciones.

SUGERENCIA DE PRUEBA	RESULTADOS
Nombre de usuarios sugeridos: "admin", "administrador", "raíz", "sistema", "invitado", "operador", "super", "qa", "test", "test1", "testing", Contraseñas: "contraseña", "contraseña123", "contraseña123", "admin" o "invitado", contraseñas vacías y cuentas anteriores	Se encontró un usuario root pero esta deshabilitado.
Los usuarios administrativos de la aplicación suelen recibir el nombre de la aplicación u organización.	No
Nombres de contactos. usuarios, las direcciones de correo electrónico revelan la convención de nomenclatura de las cuentas de usuario	**Se ha sugerido utilizar nomenclatura para cuentas de usuario como la siguiente JAAAPI<año><algún nomre>**
Nombres de usuarios anteriores con contraseñas en blanco,	No se permiten contraseñas en blanco
Fuente de página y JavaScript en proxy o busque referencias a usuarios y contraseñas en la fuente. inicie sesión y vea cada solicitud y respuesta para un inicio de sesión, como parámetros ocultos adicionales, solicitudes GET interesantes (iniciar sesión = sí)	No existe resultado de encontrar inicia sesión=si
Nombres de cuentas y contraseñas en código fuente	No existe nombre de código y contraseñas en código fuente
Varios intentos de adivinar una contraseña con un nombre de usuario conocido pueden causar que la cuenta se bloquee	La función se ha habilitado en esta aplicación

2.2.2. Prueba de contraseñas predeterminadas en cuentas nuevas.

Tabla 34 WSTG-ATHN-02 Prueba de contraseñas predeterminadas.

SUGERENCIA DE PRUEBA	RESULTADO
Política de bloque	No implementado
Observar la página de Registro de usuarios, ayuda a determinar el formato esperado y la longitud mínima o máxima de los nombres de usuario y contraseñas de la aplicación. Si no existe una página de registro de usuario, determine si la organización utiliza una convención de nomenclatura estándar para nombres de usuario, como su dirección de	**Existe una página de registro de usuario en el que se pide datos del usuario y la nomenclatura para generación de usuario se sugiere una política de: JAAPI<año><nombre>**

correo electrónico o el nombre antes de en @el correo electrónico.	
¿Puede un usuario elegir su propio nombre de usuario o el sistema genera un nombre de cuenta para el usuario basándose en cierta información personal o utilizando una secuencia predecible? Si la aplicación genera los nombres de las cuentas en una secuencia predecible, como user7811, intente borrar todas las cuentas posibles de forma recursiva. Si puede identificar una respuesta diferente de la aplicación cuando usa un nombre de usuario válido y una contraseña incorrecta, entonces puede intentar un ataque de fuerza bruta en el nombre de usuario válido (o probar rápidamente cualquiera de las contraseñas comunes identificadas arriba o en la sección de referencia).	**El usuario puede generar su propio nombre de usuario con la política sugerida siempre y cuando no exista un usuario similar registrado en el sistema de información.** **Al intentar autentificarse la aplicación con un usuario valido y contraseña incorrecta da un mensaje general de "usuario i/o contraseña incorrecta"**
Intente determinar si la contraseña generada por el sistema es predecible. Para hacer esto, cree muchas cuentas nuevas rápidamente una tras otra para poder comparar y determinar si las contraseñas son predecibles. Si es predecible, intente correlacionarlos con los nombres de usuario o cualquier cuenta enumerada y utilícelos como base para un ataque de fuerza bruta.	**Las contraseñas no son predecibles, el momento de que se crea una contraseña existen requisitos para la creación de contraseñas**
Si ha identificado la convención de nomenclatura correcta para el nombre de usuario, intente utilizar contraseñas de "fuerza bruta" con alguna secuencia predecible común, como por ejemplo fechas de nacimiento.	**No se puede decir que exiiste una nomenclatura para la creación de usuario más, existe una política propuesta JAAPI<año><algún nombre>**
Intente utilizar todos los nombres de usuario anteriores con contraseñas en blanco o utilizar el nombre de usuario también como valor de contraseña.	**El sistema de información no permite utilizar contraseñas en blanco**

2.2.3. Prueba de caja gris.

Para la prueba de caja gris hemos accedido a la entrevista de la persona encargada de TI para formuladas según OWASP (2024d).

Tabla 35 WSTG-ATHN-02 Entrevistas a personal de T.I.

Preguntas	Respuestas
Hable con el personal de TI para determinar qué contraseñas utilizan para el acceso administrativo y cómo se lleva a cabo la administración de la aplicación.	**El acceso administrativo se realiza en base a Roles, presidente secretario, Tesorero, Operador y Cooperativa**
Pregunte al personal de TI si se cambian las contraseñas predeterminadas y si las cuentas de usuario predeterminadas están deshabilitadas.	**No existen contraseñas predeterminadas, y las contraseñas se pueden cambiar el momento de que el usuario ha olvidado la contraseña, existen usuario predeterminados esta desabilitado.**

Examine la base de datos de usuarios en busca de credenciales predeterminadas como se describe en la sección de prueba de caja negra. También verifique si hay campos de contraseña vacíos.	**No se ha encontrado campos de contraseñas vacíos, existe una credencial predeterminada root no está en uso.**
Examine el código en busca de nombres de usuario y contraseñas codificados.	**No se ha encontrado nombres de usuario y contraseña**
Busque archivos de configuración que contengan nombres de usuario y contraseñas.	**Existe un archivo de configuración para el acceso a la base de datos mysql**
Examine la política de contraseñas y, si la aplicación genera sus propias contraseñas para nuevos usuarios, verifique la política en uso para este procedimiento.	**La aplicación no genera sus propias contraseñas, pero verifica la fortaleza de la contraseña generada por el usuario**

2.3. Prueba de mecanismos de bloqueos débil.

IDENTIFICACIÓN

WSTG-ATHN-03

2.3.1. Mecanismos de bloqueo.

Se ha implementado un mecanismo de bloqueo de autenticación; si el usuario es correcto y la contraseña es incorrecta se incrementa el intento en más uno para la cuenta de autentificación del usuario en la base de datos, si ha realizado intentado mayor o igual a cuatro errores de autenticación se mostrará un mensaje de bloqueo de usuario e internamente se enviará un mensaje de bloqueo al correo electrónico indicando su estado por 24 horas.

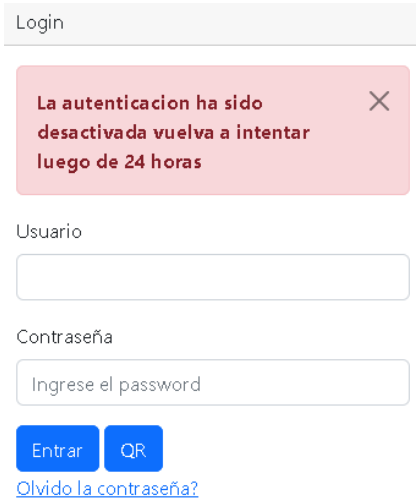


Figura 141WSTG-ATHN-03 Mecanismos de bloqueo

2.3.2. Mecanismo de desbloqueo.

Se ha implementado un mecanismo de desbloqueo con las siguientes características.

- Desbloqueo basado en tiempo, pasado 24 hora el usuario puede autentificarse nuevamente
- Desbloqueo manual por el administrador el número de intentos se pone en cero y el usuario se autentica con una nueva contraseña puesta por el administrador.
- Posteriormente el usuario podrá cambiar su contraseña.

2.4. Pruebas de omisión del esquema de autenticación.

IDENTIFICACIÓN

WSTG-ATHN-04

2.4.1. Prueba de caja negra.

2.4.1.1. Solicitud de página directa.

- https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/

Hemos realizado la solicitud de página directa de la sección usuarios del sistema de información, mientras el usuario no es autentificado no se puede tener acceso a esta sección teniendo que necesariamente autentificarse.

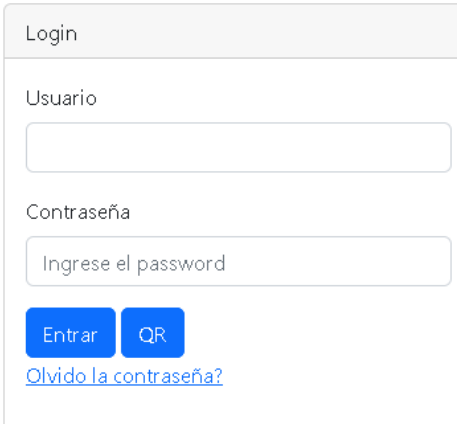


Figura 142 Solicitud página directa.

2.4.1.2. Modificación de parámetros.

Cuando la aplicación verifica un inicio de sesión exitoso sobre la base de parámetros de valor fijo. El usuario podría modificar los parámetros para acceder a las áreas protegidas sin proporcionar credenciales válidas. por ejemplo, el parámetro "autenticado" se cambia a un valor de "sí", permitiendo el acceso OWASP (2024d).

Para la autenticación dentro de la aplicación los valores necesarios para tener acceso son, usuario y password, mediante él envió de una solicitud POST la aplicación verifica en la base de datos si existe el usuario.

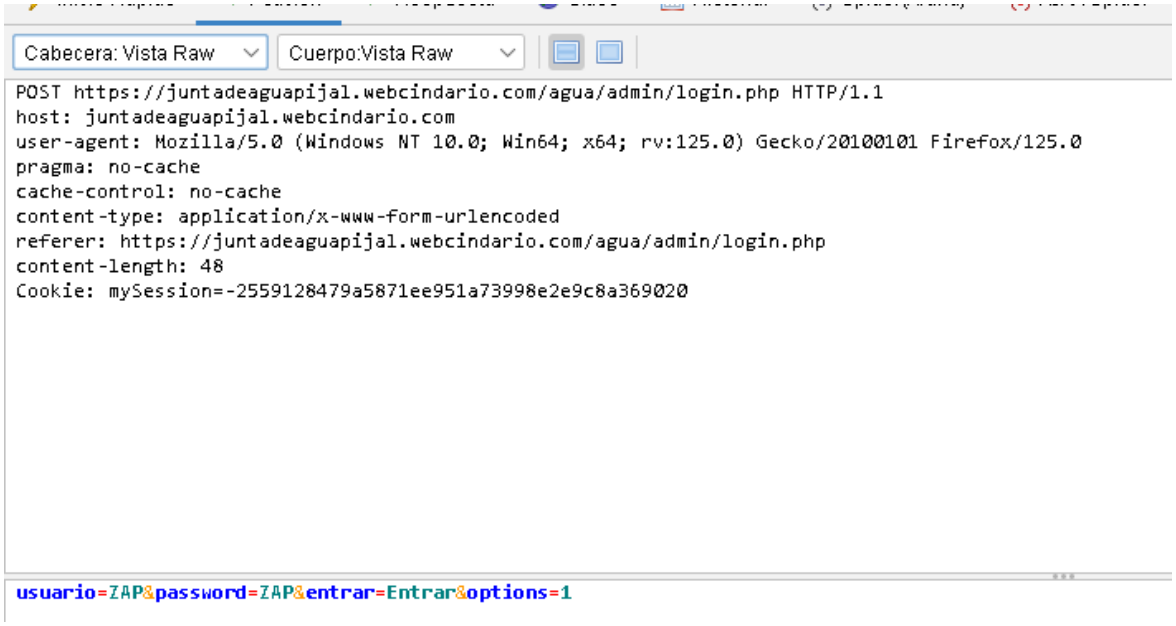


Figura 143 WSTG-ATHN-04 Modificación de parámetros.

Por lo que estos valores son evaluados dentro de la aplicación y no están basados sobre la base de un parámetro fijo ya que la autenticación se basa en usuario y contraseña almacenado en la base de datos entonces no es vulnerable.

2.4.1.3. Predicción de ID de sesión.

Fecha	Session
Mon, 01 Jul 2024 18:56:09 GMT	mySession=-5613467fbe302b5963da7c38a6f900089e201cd
Mon, 01 Jul 2024 18:58:21 GMT	mySession=-18158168b589eace79423ba19f282cfa1a467dee
Mon, 01 Jul 2024 19:01:23 GMT	mySession=-1014569b47e150d08c0c8ea077dfac3ccc5e2bf
Mon, 01 Jul 2024 19:02:05 GMT	mySession=-651821492a9e3afac814c3339eb2d51145fd84d5
Mon, 01 Jul 2024 19:02:53 GMT	mySession=-1873116230bf195da45b9de144818dc8646df5fc
Mon, 01 Jul 2024 19:03:32 GMT	mySession=-951401523635ff43de80f2aebfb3ce1bce0c29b3
Mon, 01 Jul 2024 19:03:59 GMT	mySession=-14093837cd45199757c4dc0d36f32128cd7b148
Mon, 01 Jul 2024 19:04:54 GMT	mySession=-519184b5f58f65cd7a0d94d596871fc0880f6;
Mon, 01 Jul 2024 19:05:25 GMT	mySession=-72121533144eedd76c3cdd9b3af68a91d91ecb2
Mon, 01 Jul 2024 19:18:12 GMT	mySession=-962426467643ceaf883a2ae269cb6ec3406d98;
Mon, 01 Jul 2024 19:18:57 GMT	mySession=-10414399e6a40a8413e4a82edc138326466f4b9d
Mon, 01 Jul 2024 19:19:42 GMT	mySession=-175947743fd98a21705bbb0862fd0b3b04e2e16

Figura 144 WSTG-ATHN-04 Predicción de ID de sesión.

Hemos identificado en esta aplicación que la autenticación se gestiona mediante identificadores de sesiones ID, pero en la identificación de patrones para la generación no son parecidas por lo que no podemos establecer un patrón específico para poder establecer o adivinar fácilmente una sesión, además para la generación de sesiones por parte de la aplicación se realiza una asignación aleatoria.

2.4.1.4. Inyección SQL (autenticación de formulario HTML).

Para esta prueba utilizamos la herramienta ZAP ya que anteriormente encontramos debilidades de inyección en el anterior sistema de información, utilizaremos la siguiente inyección.

Figura 145 WSTG-ATHN-04 Inyección Sql, parámetros de formulario.

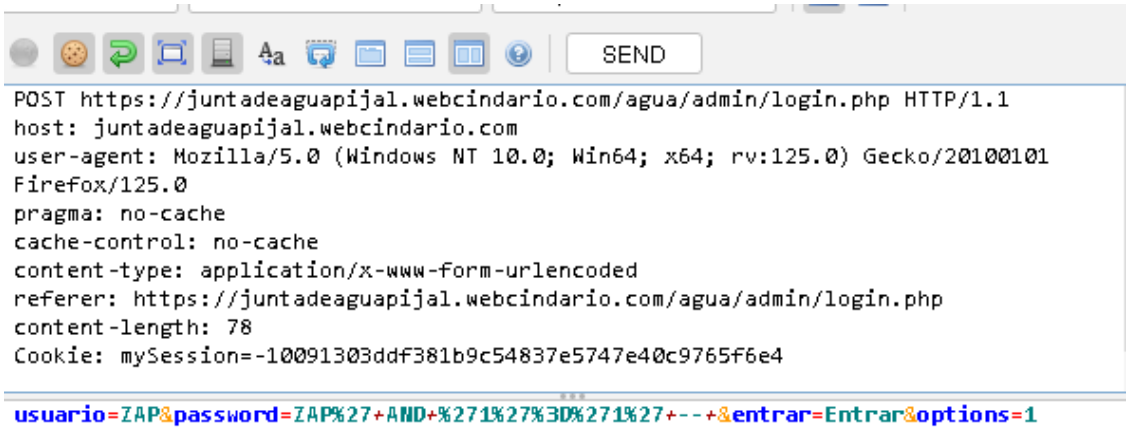


Figura 146 WSTG-ATHN-04 Inyección Sql, aplicación de prueba.

La prueba será aplicada a la sección de autenticación del sistema de información-

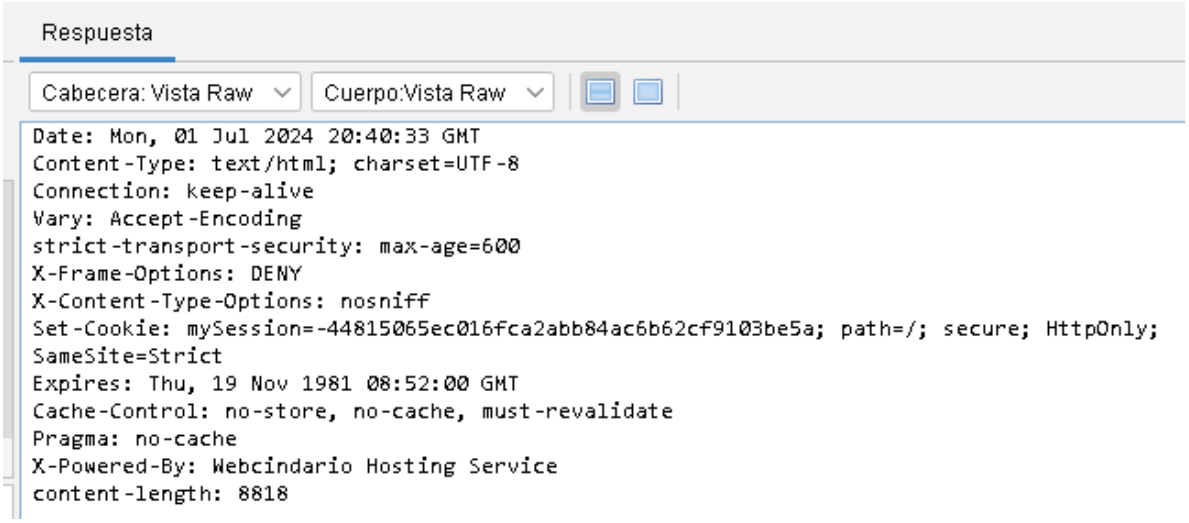


Figura 147 WSTG-ATHN-04 Respuesta Inyección Sql.

```
      <button
        type="button"
        class="btn-close"
        data-bs-dismiss="alert"
        aria-label="Close"
      ></button>
      <strong>El Usuario i/o Contraseña es incorrecto</strong>
    </div>
    <script>
      var alertList = document.querySelectorAll(".alert");
      alertList.forEach(function (alert) {
        new bootstrap.Alert(alert);
      });
    </script>
    <form action="" method="post" name="formulario" autocomplete="off">

iv class="mb-3">
```

Figura 148 WSTG-ATHN-04 Pagina respuesta Inyección Sql.

Verificamos el resultado y podemos concluir que la inyección Sql en formulario no ha surtido efecto, nos el mensaje de usuario y contraseña incorrecto.

2.4.2. Pruebas de caja gris.

Como hemos tenido acceso al código fuente en el archivo “login.php”, verificamos a fin de establecer una forma vulnerar el acceso por autenticación.

```
  $usuario=(isset($_POST['usuario']))?$_POST['usuario']:"";
  $password=(isset($_POST['password']))?$_POST['password']:"";
  $entrar=(isset($_POST['entrar']))?$_POST['entrar']:"";
  $options=(isset($_POST['options']))?$_POST['options']:"";
  /*Proceso de saneamiento de variables y prevención de inyección Sql*/
  $caracteresNoPermitidos = array("
","!",'"',"#","%","&","'","(",")","+","=",";");
  $usuario = str_replace($caracteresNoPermitidos,'',$usuario);
  $password = str_replace($caracteresNoPermitidos,'',$password);
  $entrar=str_replace($caracteresNoPermitidos,'',$entrar);
  $options=str_replace($caracteresNoPermitidos,'',$options);
  /*Fin de saneamiento */
  /*busca si existe el usuario en la base de datos*/
 try {
  $sentencia=$conexion->prepare("SELECT
id_acceso,password,cargo,f_inicio,f_final,anio_lectivo,correo,intentos,fecha
```

```
_caducidad,aplicar_2fa,activo,dia_hora_bloqueo,imagen_qr FROM usuario_acceso
WHERE login=:txtusuario  " );
  $sentencia->bindParam(':txtusuario',$usuario,PDO::PARAM_STR);
  $sentencia->execute();
  $lista_usuario=$sentencia->fetch(PDO::FETCH_LAZY);
}catch(PDOException $e){

echo "Error";
}

  if (($sentencia->rowCount())<1)
  {
   $mensaje="El Usuario i/o Contraseña es incorrecto";
  }else
  {
  if (password_verify($password, $lista_usuario['password']))  {

              $_SESSION['id_acceso']=$lista_usuario['id_acceso'];
              $_SESSION['usuario']=$usuario;
              $_SESSION['cargo']=$lista_usuario['cargo'];
              $_SESSION['logeado']=true;
              $_SESSION['f_inicio_usuario']=$lista_usuario['f_inicio'];
              $_SESSION['f_final_usuario']=$lista_usuario['f_final'];
              $fecha_caducidad=$lista_usuario['fecha_caducidad'];
              $anio_lectivo=$lista_usuario['anio_lectivo'];
  $sentencia_anio=$conexion->prepare("SELECT * FROM a_lectivo WHERE
anio_lectivo=:txtanio ");
              $sentencia_anio->bindParam(':txtanio',$anio_lectivo);
              $sentencia_anio->execute();
              $lista_anio=$sentencia_anio->fetch(PDO::FETCH_LAZY);
              $_SESSION['anio_lectivo']=$lista_anio['anio_lectivo'];
              $_SESSION['f_inicio_periodo']=$lista_anio['f_inicio'];
              $_SESSION['f_final_periodo']=$lista_anio['f_final'];
              $MM_redirectLoginSuccess ="index.php";
              header("Location: " . $MM_redirectLoginSuccess );

}
```

Una vez que se realiza el proceso de saneado de los valores $_POST de usuario y password, estos valores son consultados en la base de datos para buscar al usuario y sus datos; que luego serán almacenados en variables de sesión.

- id_acceso, registro en la base de datos.
- usuario,
- cargo.
- Logeado
- Año_lectivo
- Fechas de inicio y final para periodos

2.5. Prueba de vulnerabilidad de recordar contraseñas.

ID

WSTG-ATHN-05

2.5.1. Almacenar credenciales.

Procedemos a verificar credenciales almacenadas en mecanismos del navegador y "como se menciona en el manual de las credenciales no deben ser almacenadas en la aplicación del lado del cliente y deben ser sustituidas por tokens generados del lado del servidor".

Procedemos a ingresar con el usuario y password a fin de crear una sesión.

https://juntadeaguapijal.webcindario.com/agua/admin/login.php

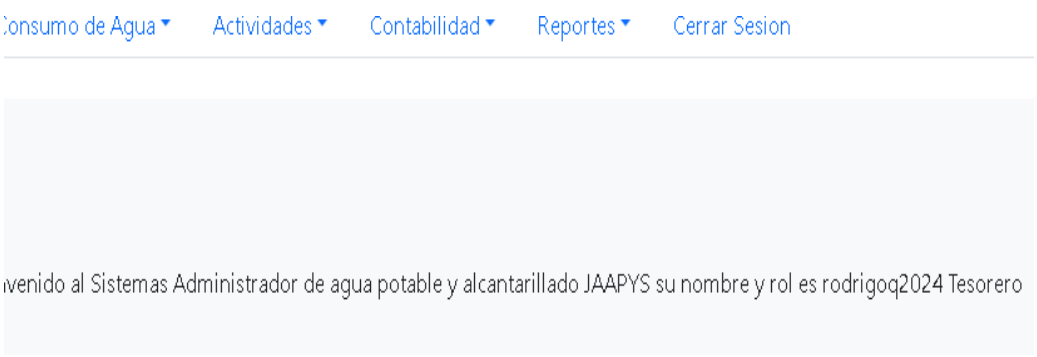


Figura 149 WSTG-ATHN-05 Almacenar credenciales sesión iniciada.

Procedemos a probar SessionStorage y LocalStorage así también la cookie del usuario.

https://juntadeaguapijal.webcindario.com/agua/admin/WSTG-ATHN-05.php

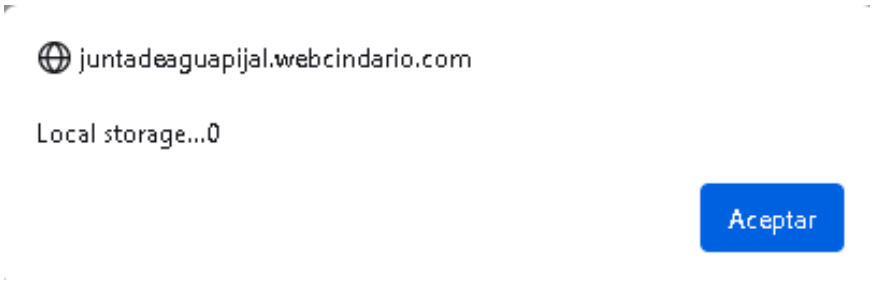


Figura 150 WSTG-ATHN-05 Almacenar credenciales Local storage.

Figura 151 WSTG-ATHN-05 Almacenar credenciales Session storage.

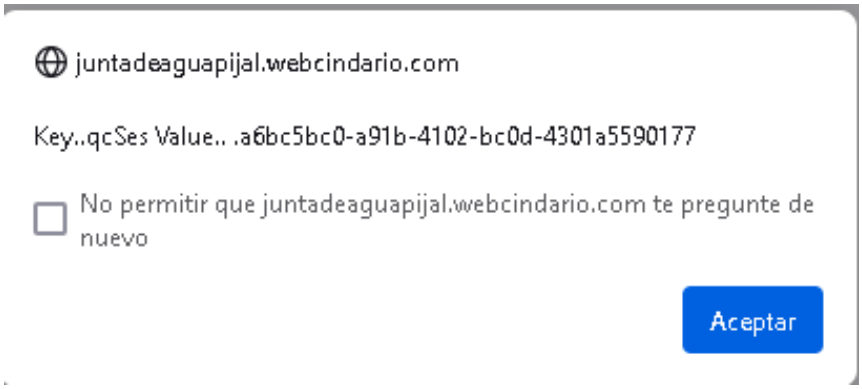


Figura 152 WSTG-ATHN-05 Almacenar credenciales Key.

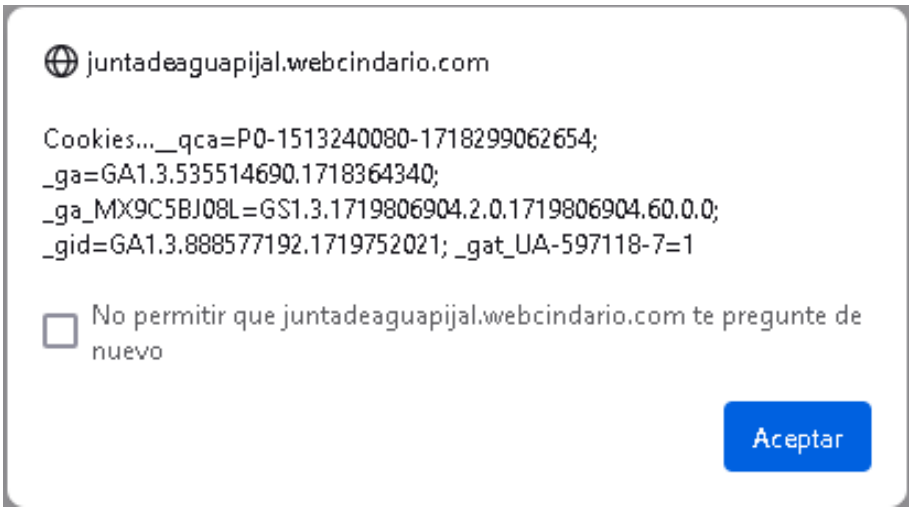


Figura 153 WSTG-ATHN-05 Almacenar credenciales Cookies.

Podemos visualizar la cookie del lado del cliente no podemos visualizar ninguna sesión parecida a nuestro sistema de información de debe ser similar a:

- `Cookie: mySession=-10091303ddf381b9c54837e5747e40c9765f6e4`

Con esta prueba podemos verificar que no existe ningún valor para Sessionstorage y Localstorage objetos de almacenamiento API de sesione a diferencia de Cookies de sesión de igual forma no podemos ver del lado de cliente con java script, y podemos deducir que la aplicación no es vulnerable a un ataque de robo de sesión por java script.

El código fuente utilizado realizar esta prueba es el siguiente:

Prueba WSTG-ATHN-05 y WSTG-CLNT-12

```
<script>
alert("Local storage..."+localStorage.length);
for (let i = 0; i < localStorage.length; i++) {
  const key = localStorage.key(i);
  const value = localStorage.getItem(key);
  console.log(`${key}: ${value}`);
  alert("Key.."+key+" Value.. ."+value);
}
alert("Session storage..."+sessionStorage.length);
for (let i = 0; i < sessionStorage.length; i++) {
  const key = sessionStorage.key(i);
  const value = sessionStorage.getItem(key);
  console.log(`${key}: ${value}`);
  alert("Key.."+key+" Value.. ."+value);
}
alert("Cookies..."+window.document.cookie);
console.log(window.document.cookie);
alert("List All Entries on the Window Object");
(() => {

  // create an iframe and append to body to load a clean window
object
  const iframe = document.createElement('iframe');
  iframe.style.display = 'none';
  document.body.appendChild(iframe);
  // get the current list of properties on window
  const currentWindow = Object.getOwnPropertyNames(window);
  // filter the list against the properties that exist in the
clean window
  const results = currentWindow.filter(
    prop => !iframe.contentWindow.hasOwnProperty(prop)
  );
  // remove iframe
  document.body.removeChild(iframe);
  // log key-value entries that are different
  results.forEach(key => console.log(`${key}: ${window[key]}`));
})();
```

```
const dumpIndexedDB = dbName => {
  const DB_VERSION = 1;
  const req = indexedDB.open(dbName, DB_VERSION);
  req.onsuccess = function() {
    const db = req.result;
    const objectStoreNames = db.objectStoreNames || [];
    console.log(`[*] Database: ${dbName}`);
    Array.from(objectStoreNames).forEach(storeName => {
      const txn = db.transaction(storeName, 'readonly');
      const objectStore = txn.objectStore(storeName);
      console.log(`\t[+] ObjectStore: ${storeName}`);
      // Print all entries in objectStore with name `storeName`
      objectStore.getAll().onsuccess = event => {
        const items = event.target.result || [];
        items.forEach(item => console.log(`\t\t[-] `, item));
      };
    });
  };
};
indexedDB.databases().then(dbs => dbs.forEach(db =>
dumpIndexedDB(db.name)));
</script>
```

2.5.2. Inyección automática de credenciales.

La inyección automática depende del navegador cuando autocompleta el formulario del usuario y password si la aplicación es vulnerable al ataque de ClickJacking y CSRF, en nuestro caso la aplicación es vulnerable.

https://sitio-malicioso.webcindario.com/Clickjacking_juntadeagua.php

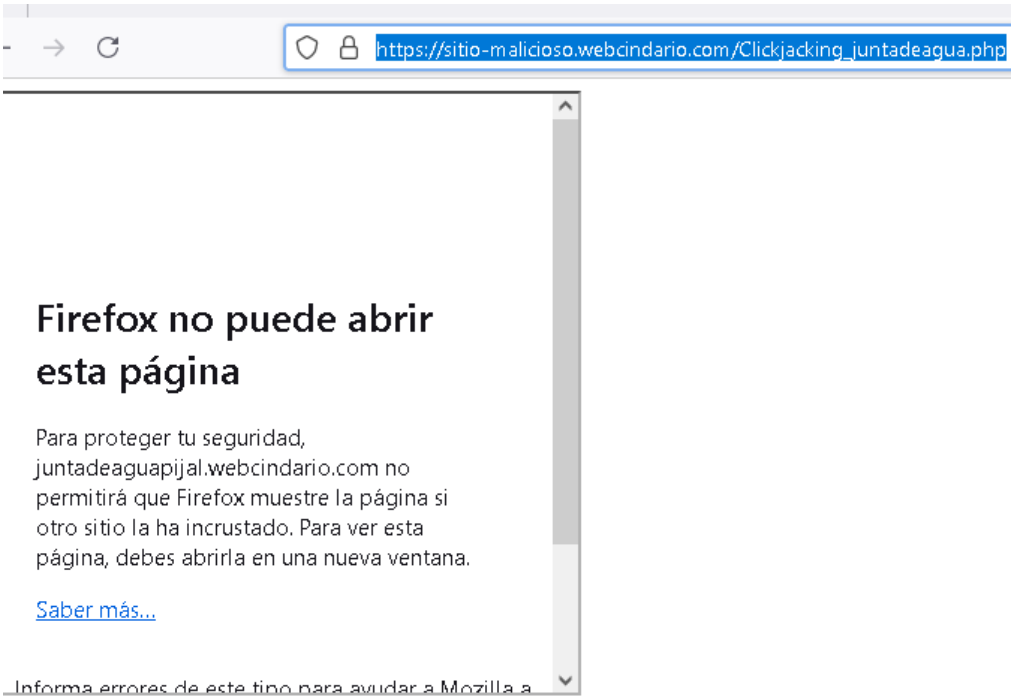


Figura 154 WSTG-ATHN-05 Inyección de credenciales Clickjacking.

El sitio no es accesible desde otro sitio malicioso, no es vulnerable.

2.5.3. Tokens debe ser analizado en términos de tiempo de vida.

Los tokens fueron programados en términos de tiempo de vida para que siempre este activo mientras el usuario realiza actividades, en caso de que el usuario esté inactivo por más de 3 minutos se cerrará la sesión del usuario y el token será destruido.

```
function correr_session() {
    session_name('mySession');
    session_start();
    // Destrucción de sesiones antiguas
    if (!empty($_SESSION['deleted_time']) && $_SESSION['deleted_time'] <
time() - 180) {
        session_destroy();
        session_start();
    }
}
```

2.6. Debilidades en prueba del cache del navegador.

IDENTIFICACIÓN

WSTG-ATHN-06

2.6.1. Historial del navegador.

El botón ATRAZ del navegador es un historial y no un cache, lo que consiste en esta prueba es ingresar o verificar información sensible, cerrar la sesión y volver atrás para ver si se puede acceder a esta información confidencial.

- https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/

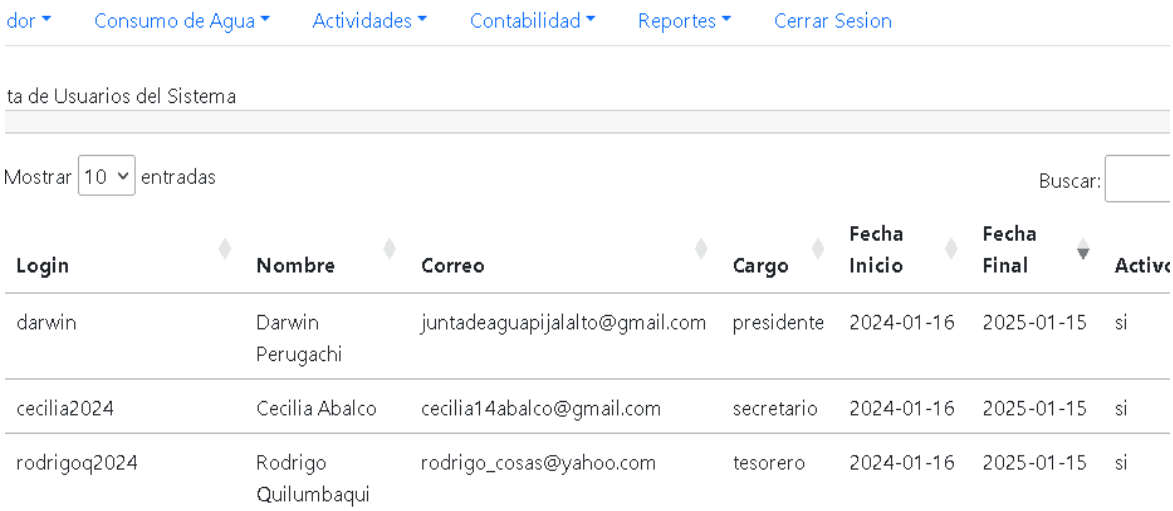


Figura 155 WSTG-ATHN-06 Prueba de historial del navegador.

Luego de visualizar información sensible procedemos a cerrar la sesión del usuario y verificamos con el botón atrás del navegador para ver si se puede visualizar nuevamente la información sensible.

En la imagen 156 podemos verificar que la sesión de usuarios se ha cerrado y la contraseña de los usuarios no se puede visualizar nuevamente y la información sensible no está expuesta.

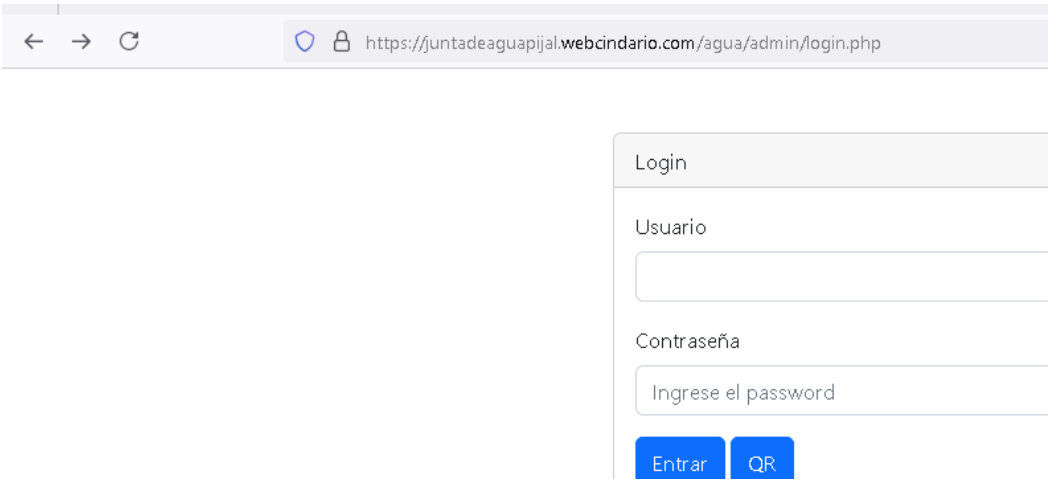


Figura 156 WSTG-ATHN-06 Resultado de prueba historial de navegador.

- Se entrega la página mediante Https.
- Configuración Cache-Control: must-revalidate.

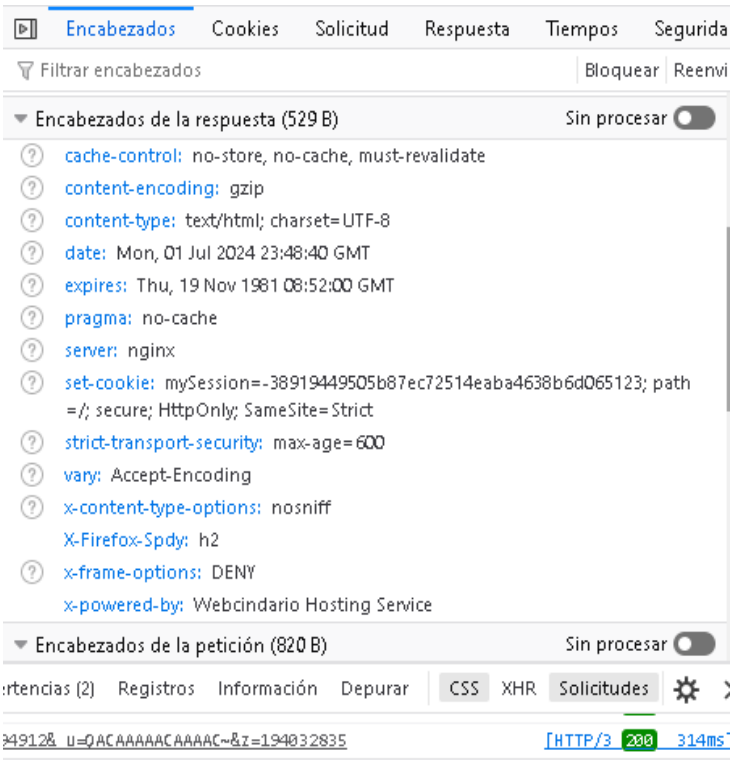


Figura 157 WSTG-ATHN-06 Prueba historial de Cache de navegador.

Como podemos ver en la imagen anterior la directiva **cache-control=must-revalidate** existe entonces resulta seguro**.**

2.6.2. Cache del navegador.

Se debe comprobar que la aplicación no filtre ningún dato confidencial en el caché del navegador; pueden usar un proxy y buscar en las respuestas del servidor que pertenecientes a la sesión, verificando la página si la página tiene información confidencial, el servidor ha indicado al navegador que no almacene en caché ningún dato. La directiva se puede emitir en los encabezados de respuesta HTTP con las siguientes directivas (OWASP, 2024d).

- Cache-Control: no-cache, no-store
- Expires: 0
- Pragma: no-cache
- Cache-Control: must-revalidate, max-age=0, s-maxage=0

Según hemos revisado las directivas para nuestra aplicación los resultados podemos comprobar en su mayoría que si cumple con los requisitos.

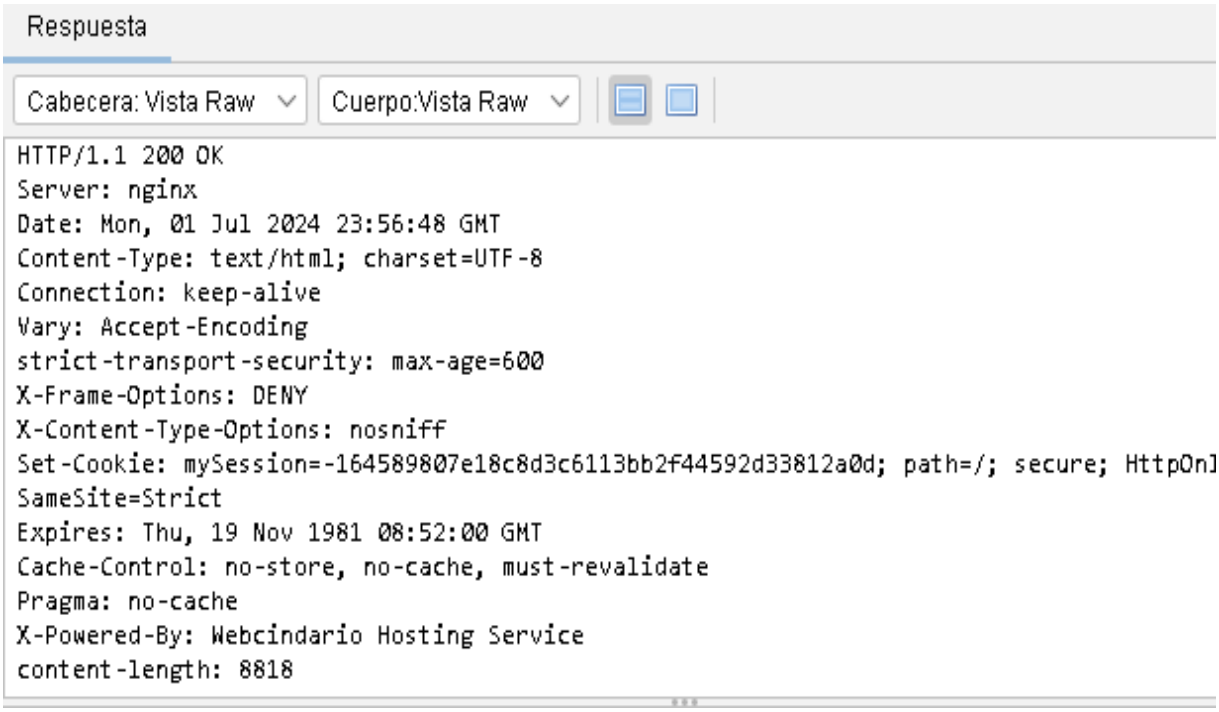


Figura 158 WSTG-ATHN-06 Cache del navegador, no cache, no store,

2.6.3. Revisar información almacenada en cache.

Para este paso hemos tenido que revisar la información almacenada en cache del navegador, y nos da como resultado que es un documento seguro.

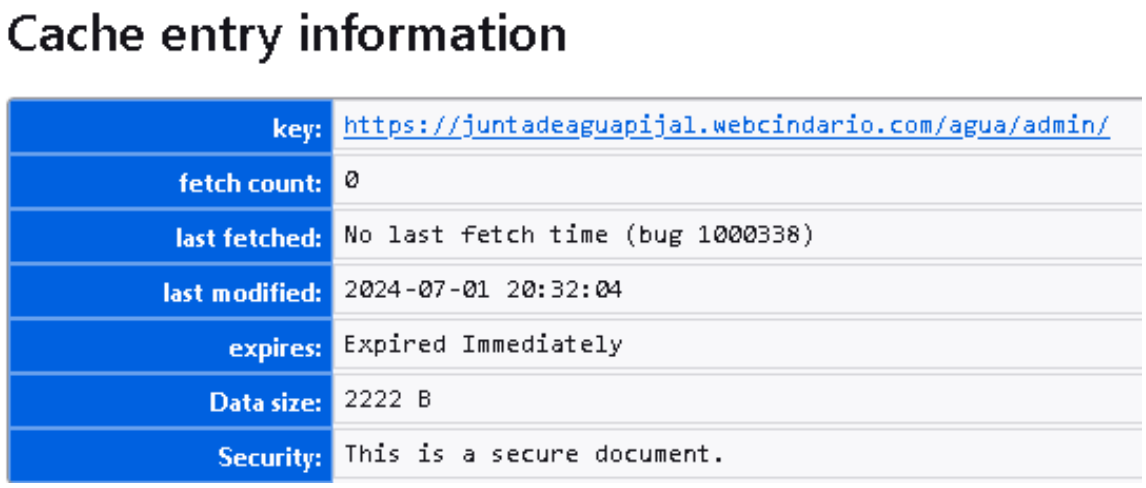


Figura 159 WSTG-ATHN-06 Revisar información almacenada en cache.

2.6.4. Verificación de manejo para navegadores móviles.

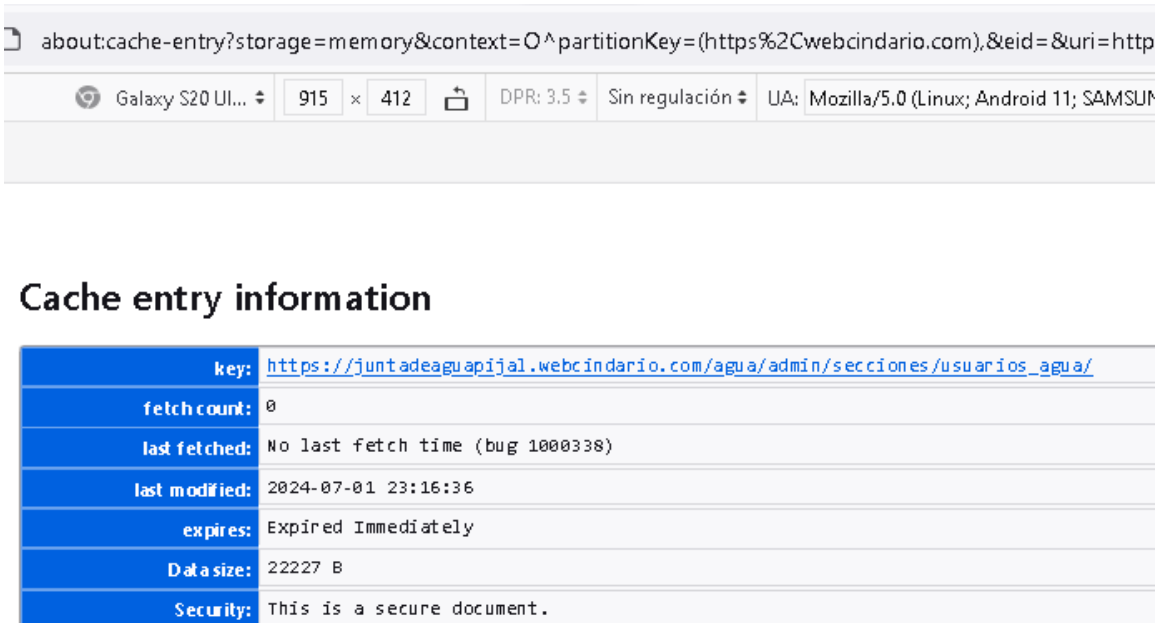


Figura 160 WSTG-ATHN-06 Verificación de cache de navegadores móviles.

2.6.5. Prueba de caja gris.

Para esta prueba hemos procedido a tener acceso como un usuario con privilegios en este caso el usuario que tiene privilegios de presidente revisaremos el cache del navegador e historial de navegador obteniendo el mismo resultado de los procedimientos anteriores.

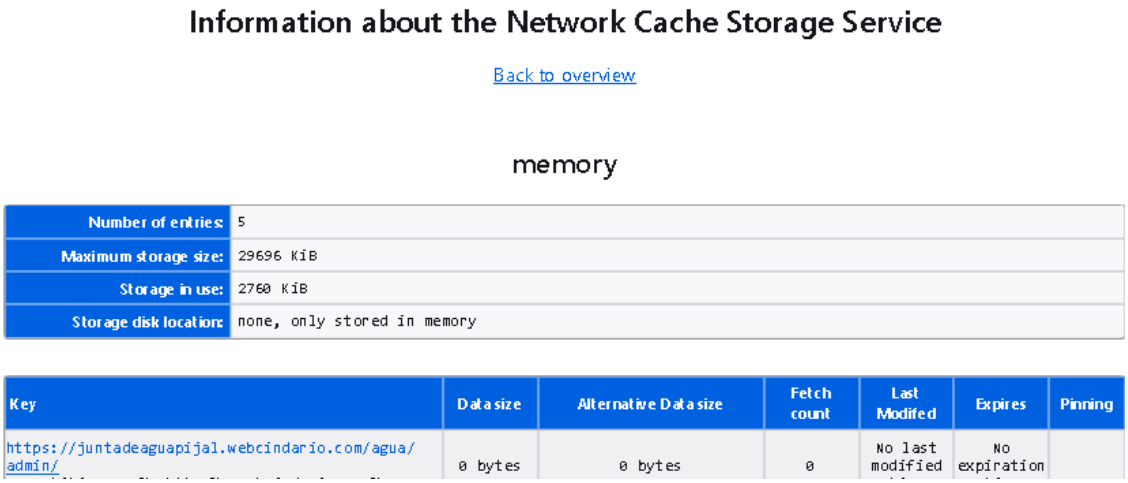


Figura 161 WSTG-ATHN-06 Cache previa autenticación de usuario.

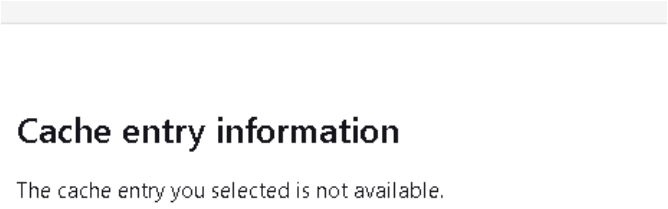


Figura 162 WSTG-ATHN-06 Revisión de cache no disponible.

2.7. Políticas de contraseñas débiles.

IDENTIFICACIÓN
WSTG-ATHN-07

El mecanismo de autenticación más frecuente es una contraseña estática. La contraseña representa las llaves del dominio, pero los usuarios a menudo lo sacrifican en nombre de la usabilidad, se sabe que las contraseñas más comunes son: 123456, password y qwerty (OWASP, 2024d).

2.7.1. Prueba con preguntas contraseñas débiles.

Tabla 36 WSTG-ATHN-07 Preguntas de políticas de contraseñas débiles.

PREGUNTAS	RESPUESTAS
¿Qué caracteres están permitidos y prohibidos en una contraseña?	**Caracteres permitidos: números, letras minúsculas, letras mayúsculas,letras con tilde, caracteres $ ¿?{}./::,@*_-** **Caracteres prohibidos: espacio, !,comillas simples, comillas dobles,#,%,&,(),+,=, punto y coma**
¿Se requiere que el usuario utilice caracteres de diferentes conjuntos de caracteres, como letras minúsculas y mayúsculas, dígitos y símbolos especiales?	**SI**
¿Con qué frecuencia puede un usuario cambiar su contraseña?	
¿Qué tan rápido puede un usuario cambiar su contraseña después de un cambio anterior?	**Después de una perdida u olvido del login**
Los usuarios pueden eludir los requisitos del historial de contraseñas cambiando su contraseña 5 veces seguidas para que después del último cambio de contraseña hayan configurado nuevamente su contraseña inicial.	**No está contemplado**
¿Cuándo debe un usuario cambiar su contraseña? o Tanto NIST como NCSC recomiendan **no** forzar la caducidad regular de la contraseña, aunque puede ser requerido por estándares como PCI DSS.	**No está contemplado**
¿Con qué frecuencia un usuario puede reutilizar una contraseña?	**En cualquiera frecuencia**
¿La aplicación mantiene un historial de las 8 contraseñas utilizadas anteriormente por el usuario?	**No**
¿Qué tan diferente debe ser la siguiente contraseña de la última contraseña?	**Ninguna**
¿Se impide al usuario utilizar su nombre de usuario u otra información de la cuenta (como el nombre o apellido) en la contraseña?	**NINGUNO**
¿Cuáles son las longitudes mínima y máxima de contraseña que se pueden establecer?	**>12 y <50**
¿Son apropiadas para la sensibilidad de la cuenta y la aplicación?	**SI**
¿Es posible establecer contraseñas comunes como Password1o 123456?	**SI**

Al responder estas preguntas podemos concluir que ya existe control y una política en la creación de la contraseña.

2.8. Respuestas débiles a preguntas de seguridad.

IDENTIFICACIÓN

WSTG-ATHN-08

Las preguntas y respuestas de seguridad, son denominadas preguntas y respuestas "secretas", se utilizan a menudo para recuperar contraseñas olvidadas, o como seguridad adicional además de la contraseña (OWASP, 2024d).

2.8.1. Prueba de preguntas débiles pre-generadas.

No se ha podido establecer ninguna evaluación de vulnerabilidad ya que el sistema de información solo maneja la autenticación mediante un usuario y contraseña, para la recuperación y desbloque de contraseña utiliza un mecanismo de 2FA

2.8.2. Prueba de preguntas débiles autogeneradas.

De la misma forma no se puede obtener una evaluación de vulnerabilidad ya que en este sistema de información solo se autentica mediante un usuario y contraseña a pesar de que existe un factor de 2FA para recuperación de usuario o desbloqueo.

2.9. Probar funcionalidades de cambio y restablecimiento de contraseña.

IDENTIFICACIÓN

WSTG-ATHN-09

2.9.1. Comprobación previa.

Para el cambio de contraseña como para el restablecimiento es importante responder a las siguientes preguntas planteadas por OWASP (2024d).

Tabla 37 WSTG-ATHN-09 Comprobación previa.

Pregunta	Respuesta
Si los usuarios, distintos de los administradores, pueden cambiar o restablecer contraseñas de cuentas distintas a las suyas.?	No
Si los usuarios pueden manipular o alterar el proceso de cambio o restablecimiento de contraseña para cambiar o restablecer la contraseña de otro usuario o administrador.	No
Si el proceso de cambio o restablecimiento de contraseña es vulnerable a CSRF **(**falsificación de solicitudes entre sitios**)**	No es vulnerable

2.9.2. Prueba de restablecimiento de contraseña.

Realizamos nuevamente el desarrollo de estas evaluaciones con el personal de T.I según lo recomendado OWASP (2024d).

Tabla 38 WSTG-ATHN-09 Preguntas de restablecimiento de contraseña.

Pregunta	**Respuesta**
¿Qué información se requiere para restablecer la contraseña?	El correo electrónico vinculado a la cuenta de usuario
¿Cómo se comunica al usuario el restablecimiento de contraseñas?	Mediante un mensaje de creo electrónico enviado al usuario
¿Las contraseñas de restablecimiento se generan aleatoriamente?	Token o código aleatorio de comprobación
¿La función de restablecimiento de contraseña solicita confirmación antes de cambiar la contraseña?	Mediante el mecanismo 2FA

2.9.3. Prueba de cambio de contraseña.

Según OWASP (2024d) otras de las preguntas más importantes para el cambio de contraseñas que debemos responder es:

¿Se solicita la contraseña anterior para completar el cambio?, Para cambio de la contraseña se ha implementado el mecanismo, para pedir la contraseña actual siempre y cuando el usuario esté autenticado.

2.10. Autenticación débil en un canal alternativo.

IDENTIFICACIÓN

WSTG-ATHN-10

Si no se encuentran vulnerabilidades de autenticación de los mecanismos principales, puede existir vulnerabilidades en canales alternativos, pudiendo eludir el canal principal o exponer información que puede ayudar a un ataque contra el canal principal(OWASP, 2024d).

2.10.1. Comprensión del mecanismo Primario.

2.10.1.1. Creación de cuenta de usuario.

https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/crear.php

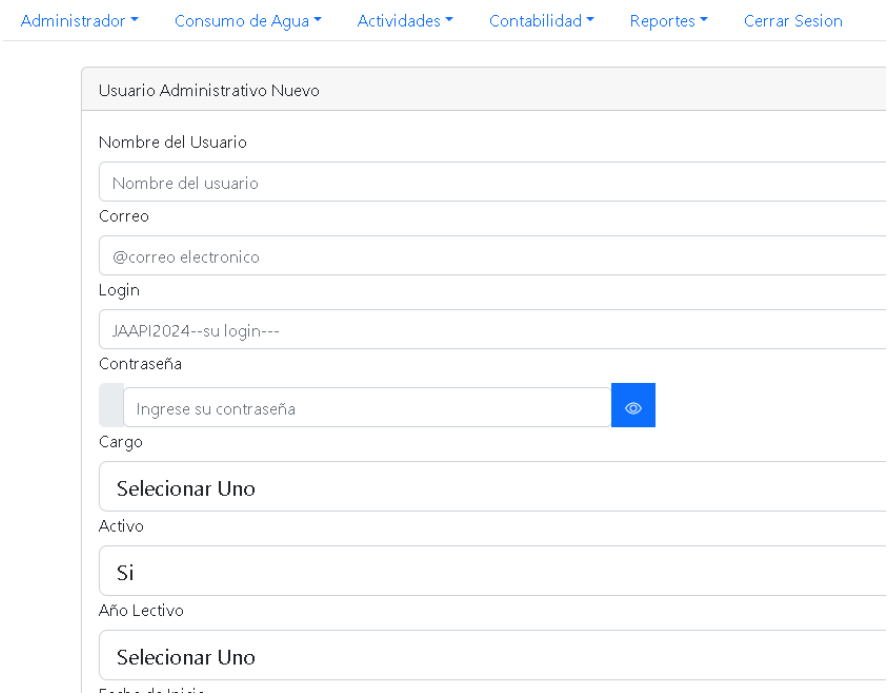


Figura 163 WSTG-ATHN-10 Creación de cuenta de usuario

Una vez que se ha autentificado en el sistema de información con privilegios de **presidente** podemos acceder a este cuadro de dialogo para la creación de un nuevo usuario con los siguientes datos.

- Nombre de usuario.
- Correo
- Login
- Contraseña
- Cargo
- Activo
- Año lectivo
- Fecha de inicio.
- Fecha final.
- Aplicar 2FA.

2.10.1.2. Modificar de cuenta de usuario.

https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/editar.php

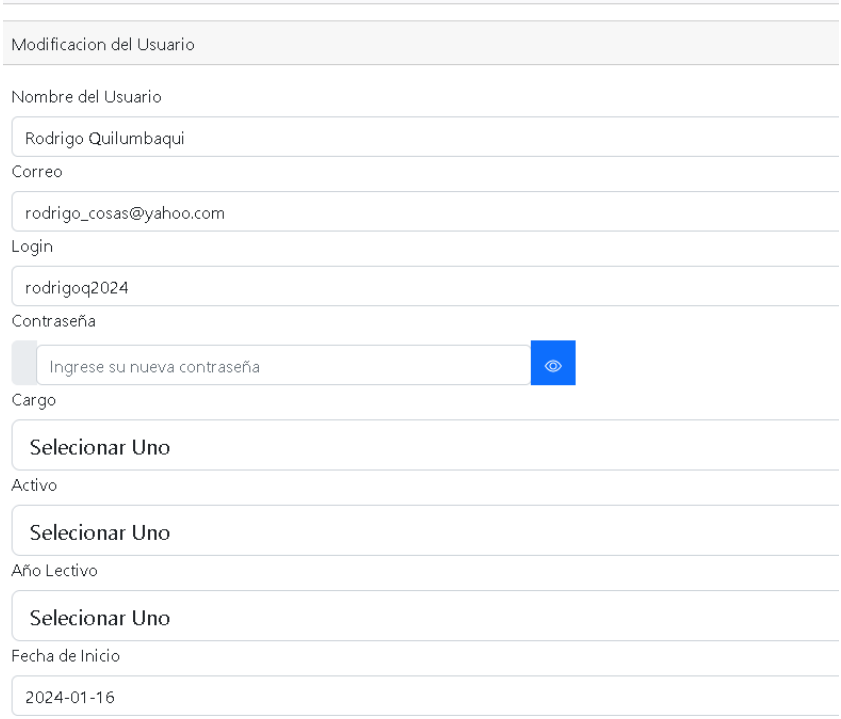


Figura 164 WSTG-ATHN-10 Modificación de usuario.

Al ingresar al registro correspondiente al usuario se encuentra un vínculo de **editar** que permite ingresar a una pantalla básica de modificación de usuario

2.10.1.3. Eliminar cuentas de usuario.

En caso de que el registro de la cuenta en la base de datos se encuentre comprometido con otro registro por integridad referencial será imposible eliminar el usuario ya que el usuario está relacionado contrastaciones echas en el periodo lectivo al que es asignado.

2.10.2. Identificar otros canales.

No existen canales alternativos para manipular registro de identificación de usuarios. Solo existe el canal principal.

2.10.3. Enumerar la funcionalidad de autenticación.

- Registro.
- Acceso.
- Modificación de datos del usuario.
- Recuperación de cuenta.
- Bloqueo de cuenta.
- Cambio de password.
- Desbloqueo de cuenta.

2.10.4. Revisar y probar.

No existen canales alternativos.

2.11. Prueba de autenticación multifactorial (MFA).

IDENTIFICACIÓN

WSTG-ATHN-11

Las aplicaciones implementan la autenticación multifactor (MFA) como una capa de seguridad para la autenticación, se conoce como autenticación de dos factores (2FA) o (2SV)(OWASP, 2024d).

2.11.1. Tipos de MFA utilizados.

Se requiere dos de los siguientes factores para la autenticación(OWASP, 2024d).

- **Algo que sabes.**
- **Algo que tienes.**
- **Algo que eres.**
- **Ubicación.**

Para nuestra aplicación se ha implementado los factores de autenticación basado en lo que posee una persona el esta caso es el correo electrónico a donde llega un código; por otra parte, el código QR de la misma forma poseemos un código para la verificación.

2.11.2. Verificación derivaciones autenticación MFA.

Para el caso de nuestro trabajo hemos podido acceder al método de autenticación 2FA para proteger los siguientes procesos en la autenticación.

Figura 165 WSTG-ATHN-11 Verificación MFA autenticación.

- Autenticación 2FA por correo electrónico.

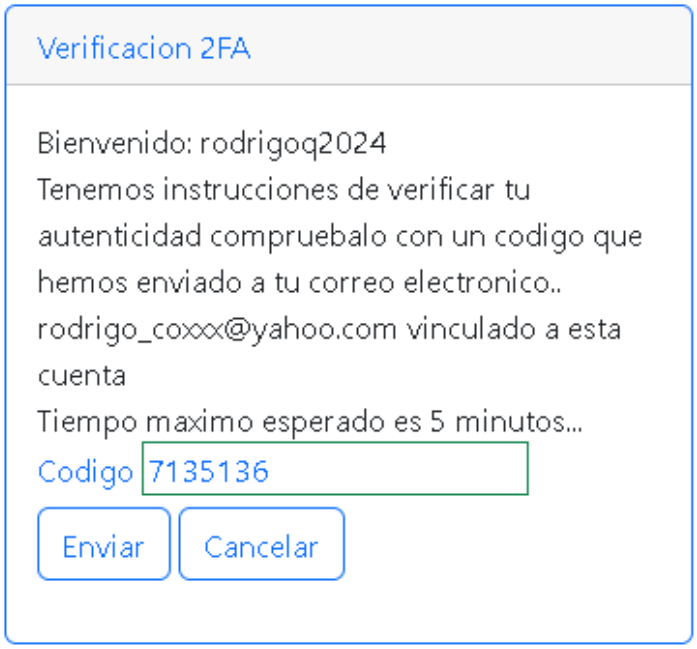


Figura 166 WSTG-ATHN-11 Verificación 2FA por correo electrónico.

- Autenticación 2FA por QR.

Figura 167 WSTG-ATHN-11 Verificación 2FA con QR.

- Olvido de contraseña y restablecimiento de contraseñas

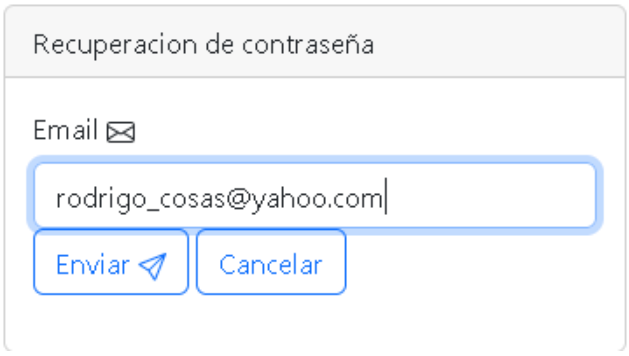


Figura 168 WSTG-ATHN-11 Recuperación de contraseña

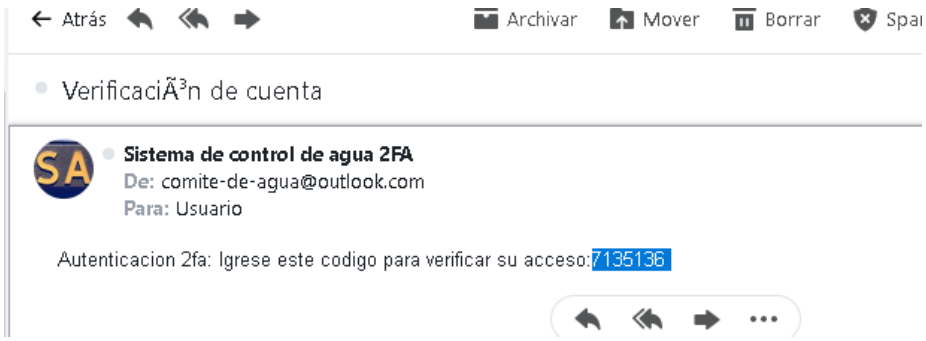


Figura 169 WSTG-ATHN-11 Correo electrónico de recuperación.

2.11.3. Prueba de gestión MFA.

Según OWASP (2024d) las funcionalidades utilizadas para administrar MFA se debe probarse desde dentro de la cuenta para detectar vulnerabilidades.

Tabla 39 WSTG-ATHN-11 Preguntas previas, pruebas de gestión MFA

Pregunta	**Respuesta.**
¿Se requiere que el usuario se vuelva a autenticar para eliminar o cambiar la configuración de MFA?	**No**
¿La funcionalidad de gestión de MFA es vulnerable a la falsificación de solicitudes entre sitios?	**No**
¿Se puede modificar la configuración de MFA de otros usuarios a través de las vulnerabilidades de IDOR (Insecure Direct Object References) ?	No

3. Pruebas de validación de entradas en la autenticación.

3.1. Pruebas de inyección SQL.

IDENTIFICACIÓN

WSTG-INPV-05

3.1.1. Pruebas para MySQL.

Según OWASP (2024d) las vulnerabilidades de inyección SQL ocurren siempre que se utilizan entradas en la construcción de una consulta SQL sin estar adecuadamente restringidas o desinfectadas.

Para poder inyectar Sql dentro de nuestra aplicación es muy importante conocer como insertar las comillas simples y también concatenar la consulta o inyección Sql, dentro de un valor o variable GET o POST en nuestra aplicación.

3.1.2. Inyección Sql dentro de una petición POST.

Para inyectar Sql dentro de un valor enviado por el método POST se lo realiza ingresando (**' OR 1='1**) en campos de usuario o contraseña de la ventana de autenticación del usuario.

' OR 1='1 => ' or 1='1'

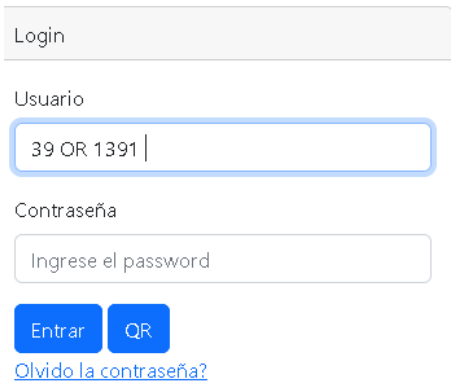


Figura 170 WSTG-INPV-05 Inyección Sql para una petición POST.

Podemos verificar que no se ha podido inserta completamente la cadena de inyección ya que existe un mecanismo de saneamiento en el que prohíbe el ingreso de los caracteres **& # ; =** quedando solamente valores saneados 39 OR 1391 debemos aclara que existe dos mecanismos de saneamiento, del lado del cliente con java Script y del lado del Servidor con una funcionalidad php; los valores de la autenticación son enviados por el método $_POST no es posible probar la autenticación por el método $_GET.

3.1.3. Inyección Sql dentro de una petición GET.

Dentro de una consulta mediante el método GET la inyección Sql se lo realiza de la siguiente forma.

+OR+1+=+1 => ' or 1='1'

- https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/editar.php?id_acceso=35+OR+1+=+1&csrf_token=c247c972874fa638a2a68399211ccfd7269d8c05e157de4e44fcfbf7d22a7ed3+OR+1+=+1

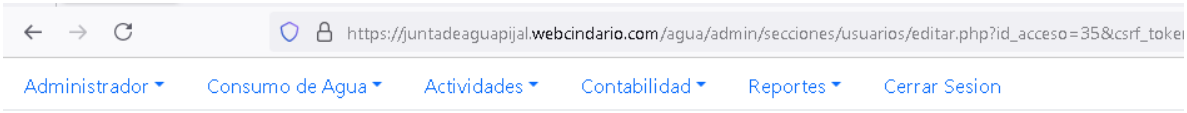


Figura 171 WSTG-INPV-05 Inyección Sql en petición GET.

- https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/editar.php?id_acceso=35+OR+1+=+1&csrf_token=75a1e070872498e2af82fe53abcd4dd0a15424af8bc085b952ecb7b294216950

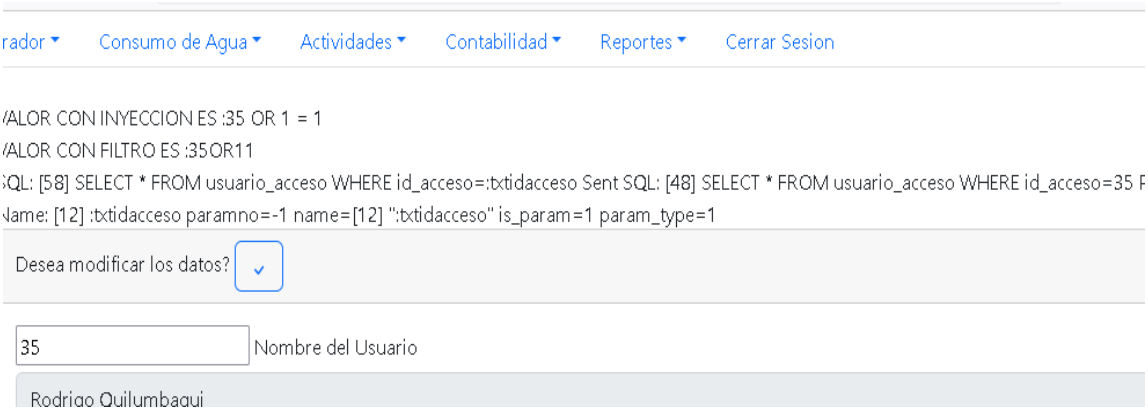


Figura 172 WSTG-INPV-05 Inyección Sql con verificación Sql.

- *SELECT * FROM usuario_acceso WHERE id_acceso=35*

Podemos verificar que la consulta SQL ha sido saneado y la inyección ha sido descartada.

3.1.4. Huellas dactilares del servidor MySQL.

Empezamos la prueba en el siguiente vinculo.

- https://comite-de-agua.webcindario.com/agua/inicio.php

Para descubrir la versión del servidor vamos a añadir la siguiente sentencia a la consulta Sql de inicio de sesión: `1 AND 1=0 UNION SELECT @@version.`

Ingresamos en login la siguiente inyección sql:

```
1&#39; OR 1=1 UNION SELECT @@VERSION ,@@VERSION,@@VERSION,
1=1&#39;1
```

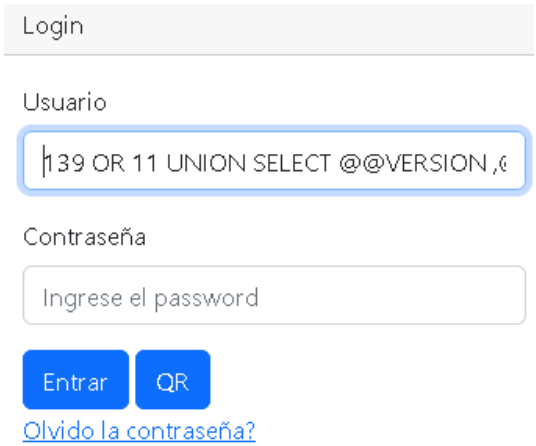


Figura 173 WSTG-INPV-05 Inyección Sql descubrimiento de servidor.

En primera instancia podemos visualizar que en la entrada Login ya la inyección ha sido filtrada de caracteres especiales y en la entrada del password no se podrá ingresar ninguna inyección ya que se lo verifica por medio de funcionalidades de hash y verify en php, resultando una consulta Sql como la siguiente:

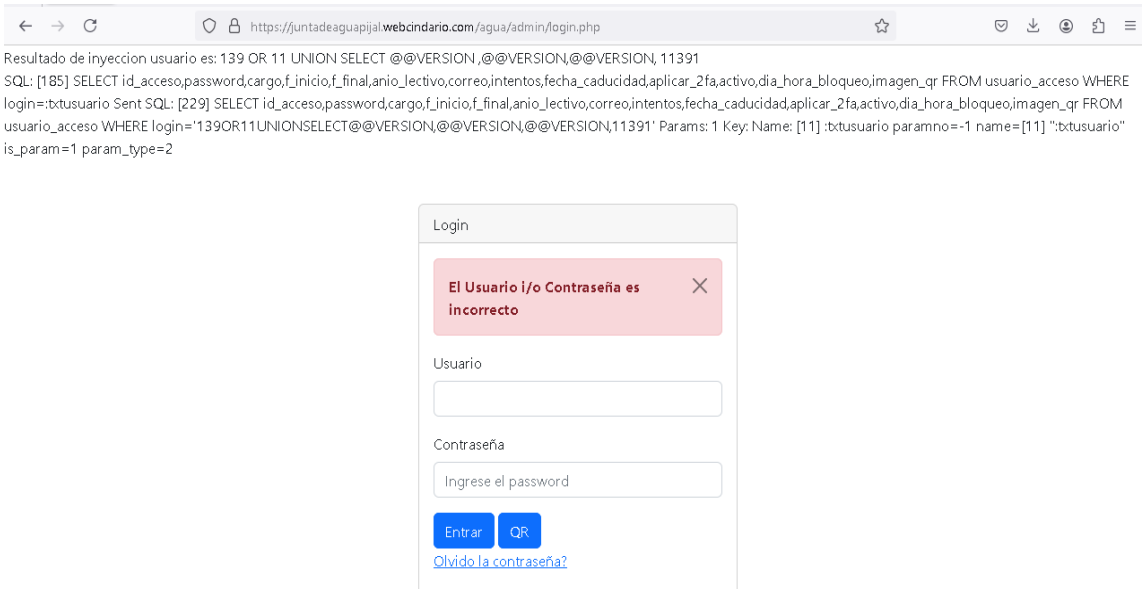


Figura 174 WSTG-INPV-05 Inyección Sql, resultado, descubrimiendo servidor.

Verificamos en el resultado de la consulta SQL y rescatamos parte de la sentencia:

SELECT id_acceso,password,cargo,f_inicio,f_final,anio_lectivo,correo,intentos,fecha_caducidad,aplicar_2fa,activo,dia_hora_bloqueo,imagen_qr FROM usuario_acceso WHERE login='139OR11UNIONSELECT@@VERSION,@@VERSION,@@VERSION,11391'

Ya que esta consulta está limpia no existe indicios de inyección Sql.

3.1.5. Descubrimiento de sesiones de usuario en servidor de base de datos.

De la misma forma que en la sección anterior podemos inyectar Sql para revisar el usuario actual de la base de datos.

Ingresamos en usuario la siguiente inyección sql:

```
1&#39; OR 1=1 UNION SELECT USER() ,USER(),USER(), 1=&#39;1
```

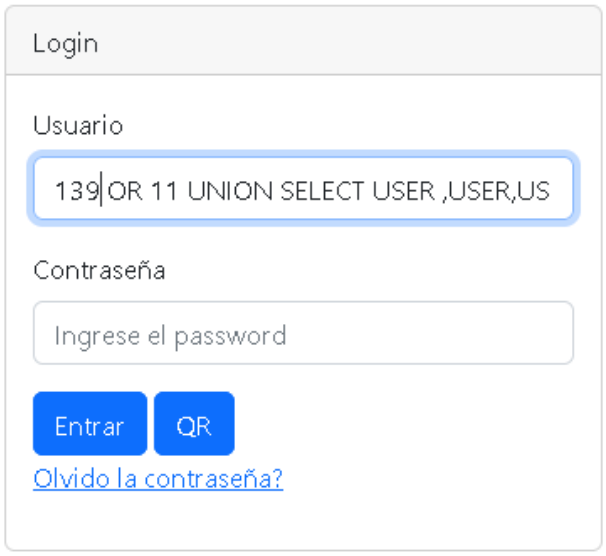


Figura 175 WSTG-INPV-05 Inyección Sql en formulario.

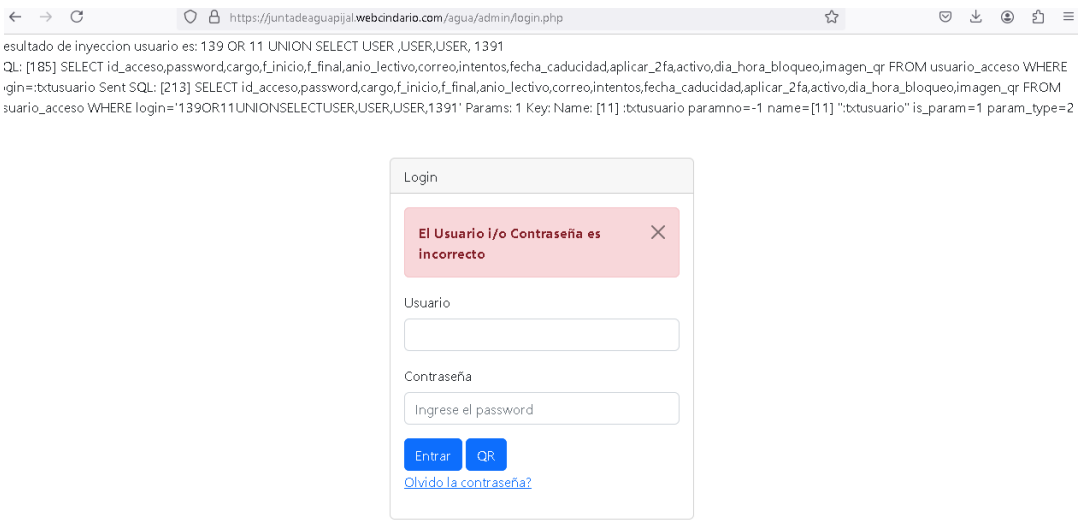


Figura 176 WSTG-INPV-05 Inyección SQL, descubriendo de usuarios BD.

El resultado de la consulta QSL es:

SELECT id_acceso,password,cargo,f_inicio,f_final,anio_lectivo,correo,intentos,fecha_caducidad,aplicar_2fa,activo,dia_hora_bloqueo,imagen_qr FROM usuario_acceso WHERE login='139OR11UNIONSELECTUSER,USER,USER,1391'

De la misma forma podemos concluir que la consulta Sql esta saneada y no puede ejecutarse como inyección Sql.

3.1.6. Nombre de la base de datos.

Ingresamos en contraseña la siguiente inyección sql:

```
1 1&#39; OR 1=1 UNION SELECT DATABASE() ,DATABASE(),DATABASE(), 1=&#39;1
```

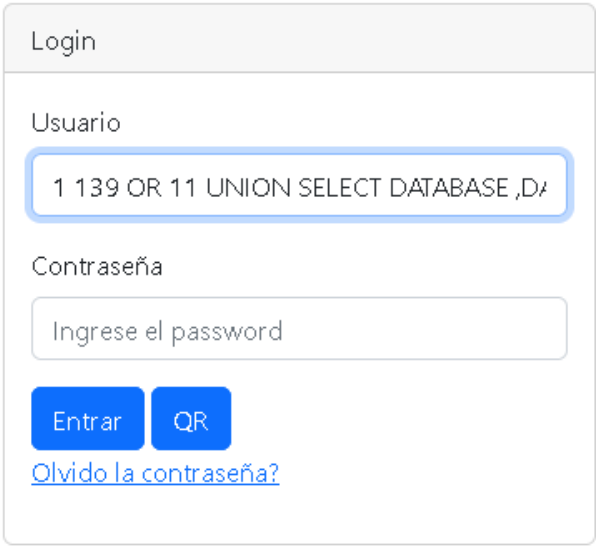


Figura 177 WSTG-INPV-05 Inyección Sql en formulario.

Podemos verificar la consulta generada.

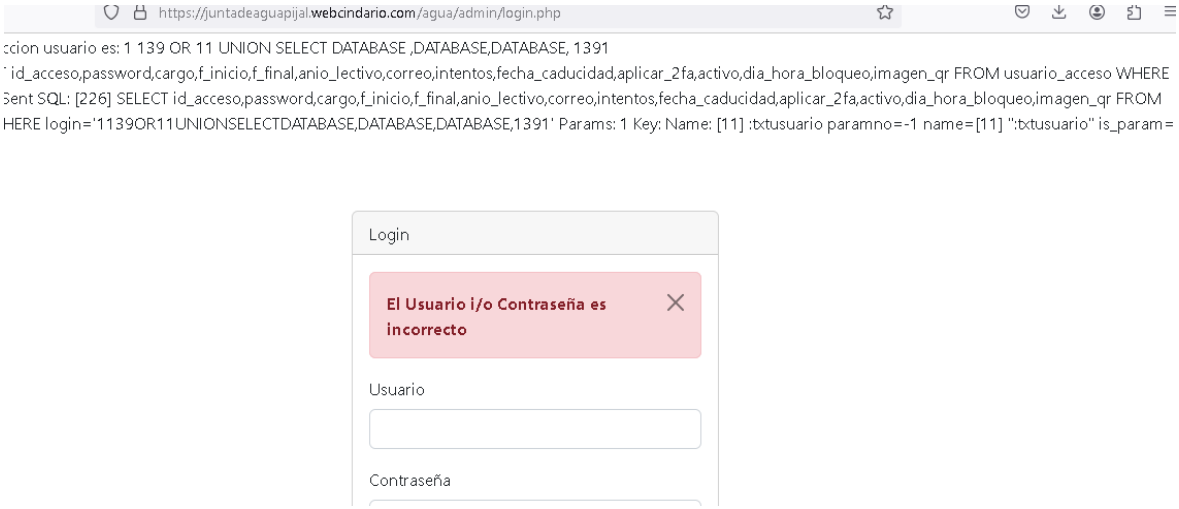


Figura 178 WSTG-INPV-05 Inyección Sql, descubriendo base de datos.

El resultado de la consulta es:

SELECT id_acceso,password,cargo,f_inicio,f_final,anio_lectivo,correo,intentos,fecha_caducidad,aplicar_2fa,activo,dia_hora_bloqueo,imagen_qr FROM usuario_acceso WHERE login='1139OR11UNIONSELECTDATABASE,DATABASE,DATABASE,1391'

La consulta no tiene indicios de inyección SQL ya que ha sido saneada; según OWASP (2024d) en una inyección SQL estándar, muestra resultados en una página como salida normal o error de MySQL, hay que reconocer los resultados obligando a una función/procedimiento o al servidor a lanzar un error.

Ahora procedemos a verificar vulnerabilidad MySQL según la herramienta ZAP.

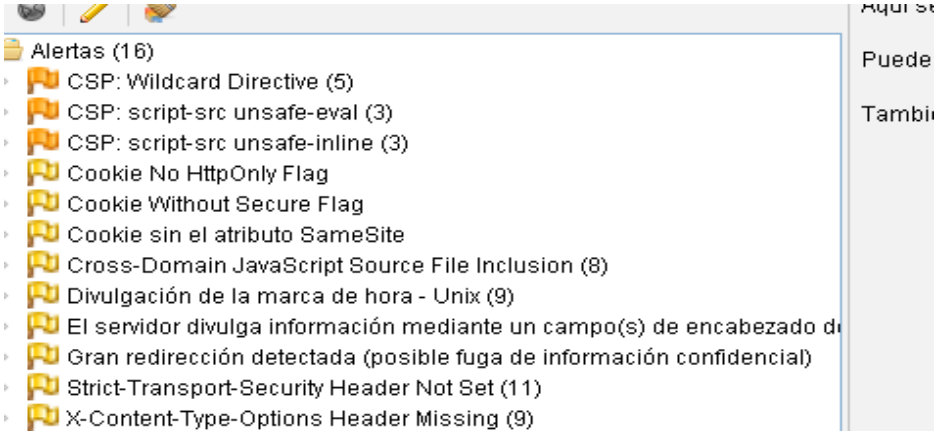


Figura 179 WSTG-INPV-05 Verificación de vulnerabilidades Sql con ZAP.

Para revisar si una página es vulnerable, se inserta ciertos parámetros, caracteres o valores a la consulta resultando con respuestas del servidor MySQL error de consulta, pudiendo estructurar una inyección Sql con resultados más certeros.

En las pruebas que aplicamos insertaremos el valor **%27 que equivale a comillas simples en ASCII Encoding Reference, html 5 UTF-8 a la variable id_acceso,** por el método $_GET.

Y lo podemos ver en la siguiente línea.

- https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/editar.php?**id_acceso=%27**&csrf_token=b49caefba1b41a1b41b5f8f9126ec0c2517414e6c0b56740008c4c05dc117f28

] SELECT * FROM usuario_acceso WHERE id_acceso=0 Params: 1 Key:
=1

Figura 180 WSTG-INPV-05 Resultado de inyección SQL con ASCCII %27.

La consulta es correcta y no da ningún error, entonces no existe riesgo de inyección Sql, finalmente los caracteres que debemos sanear antes de ejecutar una consulta Sql son:

Tabla 40 Codificación ASCII para inyección MySql

ASCII Encoding Reference	**CARÁCTER**	**NOMBRE**
%27	‘	Apostrofe
%28	(	abrir paréntesis
%29	)	Cerrar paréntesis

Otra de las vulnerabilidades que debemos probar es enlistar todos los datos de una tabla mediante la introducción **+OR+1+=+1** en la cadena de consulta $_GET de la variable **id_acceso.**

- https://juntadeaguapijal.webcindario.com/agua/admin/secciones/usuarios/editar.php?**id_acceso=35+OR+1**+=+1&csrf_token=a4974f28fffaf1f2b96ea16a768ec51340fc863037de8717abff2446f20012bf.

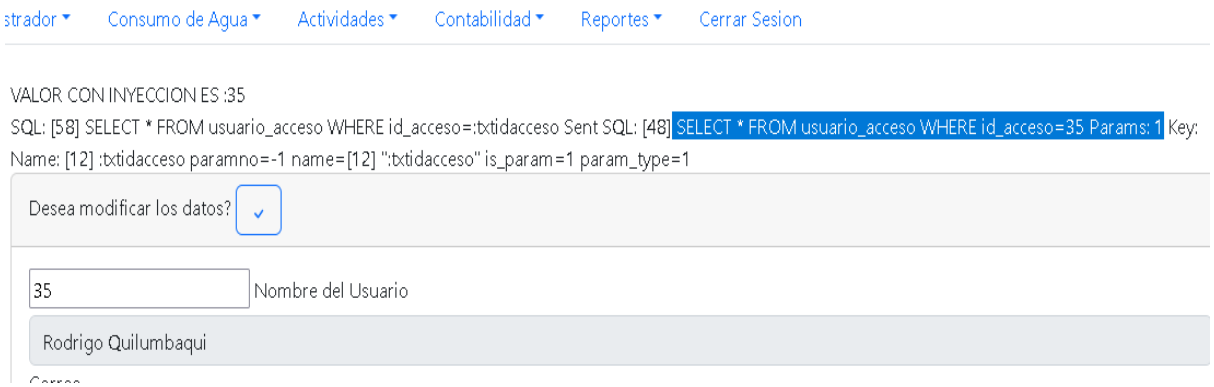


Figura 181 WSTG-INPV-05 Inyección, enlistar todos los datos de una tabla.

Podemos ver el resultado como la siguiente consulta: **SELECT * FROM usuario_acceso WHERE id_acceso=35** y concluimos que la consulta Sql no tiene ningún indicio de Inyección pues ha sido saneada y la aplicación no es vulnerable.

Anexo 6 Evaluación de riesgos de la solución del sistema de información.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Rastreo de credenciales de usuario

Factores de probabilidad				**Factores de impacto**		
Factores de la amenaza				***Factores ténicos de impacto***		
Habilidades requeridas	Habilidades de penetración de seguridad. [9]	9		Pérdida de confidencialic	Se divulgan datos mínimos no confidenciales [2]	2
Motivo	Posible recompensa [4]	4		Pérdida de integridad	Datos mínimos ligeramente corrupto [1]	1
Oportunidad	Se requiere algún acceso o recursos [7]	7		Pérdida de disponibilidad	Servicios secundarios mínimos interrumpidos [1]	1
Tamaño de poblacion	Usuarios de internet anónimos [9]	9		Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Facores de la vulnerabilidad				***Factores de impacto empresarial.***		
Fácil de descubrir	Practicamente imposible [1]	1		Daño financiero	Efecto menor en el beneficio anual [3]	3
Facilidad de explotación	Teórico [1]	1		Daño a la reputación	Daño mínimo [1]	1
Conciencia	Desconocido [1]	1		Incumplimiento	Violación menor [2]	2
Detección de intrusiones	No registrado [9]	9		Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**5,125**	**Puntuación de impacto:**	**2,375**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	**->Bajo<-**	Moderate	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	**->Bajo<-**	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 182 Evaluación de riesgos, rastreo de credenciales(solución)

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Aseguramiento de Token de sesion.

Factores de probabilidad		
Factores de la amenaza		
Habilidades requeridas	Habilidades de penetración de seguridad. [9]	9
Motivo	Posible recompensa [4]	4
Oportunidad	Se requiere algún acceso o recursos [7]	7
Tamaño de poblacion	Administradores de sistemas [2]	2

Factores de impacto		
Factores técnicos de impacto		
Pérdida de confidencialid	Se divulgan datos mínimos no confidenciales [2]	2
Pérdida de integridad	Datos mínimos ligeramente corrupto [1]	1
Pérdida de disponibilidad	Servicios secundarios mínimos interrumpidos [1]	1
Pérdida de responsabilidad	Ataque completamente anónimo [9]	9

Factores de la vulnerabilidad		
Fácil de descubrir	Dificil [3]	3
Facilidad de explotación	Teórico [1]	1
Conciencia	Oculto [4]	4
Detección de intrusiones	Registrado sin revisión [8]	8

Factores de impacto empresarial		
Daño financiero	Los daños cuestan menos que solucionar el problema [1]	1
Daño a la reputación	Daño mínimo [1]	1
Incumplimiento	Violación menor [2]	2
Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**4,75**

Puntuación de impacto:	**2,75**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	->Bajo<-	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	->Bajo<-	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 183 Evaluación de riesgos, aseguramiento de token y sesiones(solución).

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Credenciales por defecto

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores técnicos de impacto***		
Habilidades requeridas	Usuario avanzado de computadora [5]	5	Pérdida de confidencialic	Se divulgan datos mínimos no confidenciales [2]	2
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Datos mínimos ligeramente corrupto [1]	1
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	Servicios secundarios mínimos interrumpidos [1]	1
Tamaño de poblacion	Administradores de sistemas [2]	2	Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Dificil [3]	3	Daño financiero	Efecto menor en el beneficio anual [3]	3
Facilidad de explotación	Dificil [3]	3	Daño a la reputación	Daño mínimo [1]	1
Conciencia	Oculto [4]	4	Incumplimiento	Violación menor [2]	2
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**4,875**	**Puntuación de impacto:**	**3**

Gravedad general de riesgo :	**MODERADO**

Probabilidad	**Impacto**		
	Bajo	->Moderado<-	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	Bajo	->Moderado<-	Alto
Alto	Moderado	Alto	Critico

Figura 184 Evaluación de riesgos, credenciales por defecto(solución).

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Mecanismos de bloqueo y desbloqueo

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores técnicos de impacto***		
Habilidades requeridas	Habilidades de penetración de seguridad. [9]	9	Pérdida de confidencialid	No Applicable [0]	0
Motivo	Alta recompensa [9]	9	Pérdida de integridad	No Applicable [0]	0
Oportunidad	Se requiere acceso completo o recursos costosos [0]	0	Pérdida de disponibilidad	Servicios secundarios mínimos interrumpidos [1]	1
Tamaño de poblacion	Usuarios de internet anónimos [9]	9	Pérdida de responsabilidad	Ataque totalmente rastreable hasta el individuo [1]	1
Factores de la vulnerabilidad			***Factores de impacto empresarial.***		
Fácil de descubrir	Dificil [3]	3	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Teórico [1]	1	Daño a la reputación	Pérdida de cuentas importantes [4]	4
Conciencia	Desconocido [1]	1	Incumplimiento	No Applicable [0]	0
Detección de intrusiones	Not} Applicable [0]	0	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**4**	**Puntuación de impacto:**	**2,25**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	->Bajo<-	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	->Bajo<-	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 185 Evaluación de riesgo, mecanismos de bloqueo y desbloqueo.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Solicitud de pagina directa(Acceso a información confidencial)

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores técnicos de impacto***		
Habilidades requeridas	Habilidades de penetración de seguridad. [9]	9	Pérdida de confidencialic	Se divulgan datos mínimos no confidenciales [2]	2
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Datos mínimos ligeramente corrupto [1]	1
Oportunidad	Se requiere algún acceso o recursos [7]	7	Pérdida de disponibilidac	Servicios secundarios mínimos interrumpidos [1]	1
Tamaño de poblacion	Usuarios de internet anónimos [9]	9	Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Factores de la vulnerabilidad			***Factores de impacto empresarial.***		
Fácil de descubrir	Practicamente imposible [1]	1	Daño financiero	Efecto menor en el beneficio anual [3]	3
Facilidad de explotación	Teórico [1]	1	Daño a la reputación	Daño mínimo [1]	1
Conciencia	Desconocido [1]	1	Incumplimiento	No Applicable [0]	0
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**5,75**	**Puntuación de impacto:**	**2,125**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	->Bajo<-	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	->Bajo<-	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 186 Evaluación de riesgo, acceso a información confidencial(solución).

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Inyeccion automatica de credenciales

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores técnicos de impacto***		
Habilidades requeridas	Habilidades de penetración de seguridad. [9]	9	Pérdida de confidencialid	No Applicable [0]	0
Motivo	Alta recompensa [9]	9	Pérdida de integridad	No Applicable [0]	0
Oportunidad	Se requiere algún acceso o recursos [7]	7	Pérdida de disponibilidad	Servicios secundarios mínimos interrumpidos [1]	1
Tamaño de poblacion	Usuarios de internet anónimos [9]	9	Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Practicamente imposible [1]	1	Daño financiero	Los daños cuestan menos que solucionar el problema [1]	1
Facilidad de explotación	Teórico [1]	1	Daño a la reputación	No Applicable [0]	0
Conciencia	Desconocido [1]	1	Incumplimiento	No Applicable [0]	0
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**5,75**	**Puntuación de impacto:**	**1,375**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	->Bajo<-	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	->Bajo<-	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 187 Evaluación de riesgo, inyección automática de credenciales(s)

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Tokens y sesiones, tiempo de expíracion

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores técnicos de impacto***		
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialid	No Applicable [0]	0
Motivo	Alta recompensa [9]	9	Pérdida de integridad	No Applicable [0]	0
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	Servicios secundarios mínimos interrumpidos [1]	1
Tamaño de poblacion	Usuarios de internet anónimos [9]	9	Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Dificil [3]	3	Daño financiero	Los daños cuestan menos que solucionar el problema [1]	1
Facilidad de explotación	Dificil [3]	3	Daño a la reputación	Daño mínimo [1]	1
Conciencia	Obio [6]	6	Incumplimiento	Violación menor [2]	2
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**5,5**	**Puntuación de impacto:**	**1,75**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	->Bajo<-	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	->Bajo<-	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 188 Evaluación de riesgos, expiración de sesiones(solución).

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Politicas de fortalecimineto de password

Factores de probabilidad					
Factores de la amenaza					
Habilidades requeridas	Usuario avanzado de computadora [5]	5			
Motivo	Alta recompensa [9]	9			
Oportunidad	Se requiere algún acceso o recursos [7]	7			
Tamaño de poblacion	Usuarios de internet anónimos [9]	9			
Factores de vulnerabilidad					
Fácil de descubrir	Dificil [3]	3			
Facilidad de explotación	Dificil [3]	3			
Conciencia	Obio [6]	6			
Detección de intrusiones	Not} Applicable [0]	0			

Factores de impacto					
Factores técnicos de impacto					
Pérdida de confidencialic	Se divulgan datos mínimos no confidenciales [2]	2			
Pérdida de integridad	Datos mínimos ligeramente corrupto [1]	1			
Pérdida de disponibilidad	No Applicable [0]	0			
Pérdida de responsabilidad	Ataque completamente anónimo [9]	9			
Factores de impacto empresarial					
Daño financiero	Los daños cuestan menos que solucionar el problema [1]	1			
Daño a la reputación	Daño mínimo [1]	1			
Incumplimiento	Violación menor [2]	2			
Violación de privacidad	No Applicable [0]	0			

Puntuación de probabilidad:	**5,25**

Puntuación de impacto:	**2**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	->Bajo<-	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	->Bajo<-	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 189 Evaluación de riesgos, de fortalecimiento de password(solución).

OWASP Calculadora de Evaluación de riesgo

Risk Assessment Calculator

Problema: Preguntas de seguridad en autenticación y recuperación de credenciales

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores técnicos de impacto***		
Habilidades requeridas	Sin habilidades tecnicas [1]	1	Pérdida de confidencialic	Todos los datos divulgados [9]	9
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Amplios datos ligeramente corruptos [5]	5
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	No Applicable [0]	0
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	No Applicable [0]	0
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Facil [7]	7	Daño financiero	Efecto significativo en el beneficio anual [7]	7
Facilidad de explotación	Facil [5]	5	Daño a la reputación	Marca de daño [9]	9
Conciencia	Obio [6]	6	Incumplimiento	Violación de alto perfil [7]	7
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	Cientos de personas [5]	5

Puntuación de probabilidad:	**5,875**	**Puntuación de impacto:**	**5,25**

Gravedad general de riesgo :	**MODERADO**

	Impacto		
Probabilidad	Bajo	->Moderado<-	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	Bajo	->Moderado<-	Alto
Alto	Moderado	Alto	Critico

Figura 190 Evaluación de riesgos, preguntas de seguridad en credenciales.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Funcionalidades de reinicio

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores técnicos de impacto***		
Habilidades requeridas	Algunas habilidades técnicas [3]	3	Pérdida de confidencialid	No Applicable [0]	0
Motivo	Alta recompensa [9]	9	Pérdida de integridad	Datos mínimos ligeramente corrupto [1]	1
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidad	Servicios secundarios mínimos interrumpidos [1]	1
Tamaño de poblacion	Usuarios de internet anónimos [9]	9	Pérdida de responsabilidad	No Applicable [0]	0
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	No Applicable [0]	0	Daño financiero	Los daños cuestan menos que solucionar el problema [1]	1
Facilidad de explotación	No Applicable [0]	0	Daño a la reputación	Daño mínimo [1]	1
Conciencia	Obio [6]	6	Incumplimiento	Violación menor [2]	2
Detección de intrusiones	Not} Applicable [0]	0	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**3,875**	**Puntuación de impacto:**	**0,75**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	->Bajo<-	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	->Bajo<-	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 191 Evaluación de riesgo, funcionalidades de reinicio(solución).

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema:Canal alternativo de autenticación

Factortes de probabilidad	
Factores de la amenaza	
Habilidades requeridas	No Applicable [0]
Motivo	No Applicable [0]
Oportunidad	Se requiere acceso completo o recursos costosos [0]
Tamaño de poblacion	No Applicable [0]

	Factores de impacto		
	Factores técnicos de impacto		
0	Pérdida de confidencialid	No Applicable [0]	0
0	Pérdida de integridad	No Applicable [0]	0
0	Pérdida de disponibilidad	Todos los servicios completamente perdidos. [9]	9
0	Pérdida de responsabilidad	No Applicable [0]	0

Factores de la vulnerabilidad	
Fácil de descubrir	No Applicable [0]
Facilidad de explotación	No Applicable [0]
Conciencia	Obio [6]
Detección de intrusiones	No registrado [9]

	Factores de impacto empresarial		
0	Daño financiero	No Applicable [0]	0
0	Daño a la reputación	Daño mínimo [1]	1
6	Incumplimiento	Violación de alto perfil [7]	7
9	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**1,875**

Puntuación de impacto:	**2,125**

Gravedad general de riesgo :	**Note**

		Impacto		
Probabilidad		->Bajo<-	Moderado	Alto
	->Bajo<-	->Note<-	Bajo	Moderado
	Moderado	Bajo	Moderado	Alto
	Alto	Moderado	Alto	Critico

Figura 192 Evaluación de riesgo, canal alternativo de autenticación.

OWASP Calculadora de Evaluación de riesgo
Risk Assessment Calculator

Problema: Autenticacion MFA

Factores de probabilidad			**Factores de impacto**		
Factores de la amenaza			***Factores técnicos de impacto***		
Habilidades requeridas	Habilidades de penetración de seguridad. [9]	9	Pérdida de confidencialic	No Applicable [0]	0
Motivo	Posible recompensa [4]	4	Pérdida de integridad	No Applicable [0]	0
Oportunidad	Se requiere acceso completo o recursos costosos	0	Pérdida de disponibilidad	No Applicable [0]	0
Tamaño de poblacion	Usuarios autenticados [6]	6	Pérdida de responsabilidad	No Applicable [0]	0
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Practicamente imposible [1]	1	Daño financiero	Los daños cuestan menos que solucionar el problema [1]	1
Facilidad de explotación	Teórico [1]	1	Daño a la reputación	Daño mínimo [1]	1
Conciencia	Obio [6]	6	Incumplimiento	Violación menor [2]	2
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**4,5**	**Puntuación de impacto:**	**0,5**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	**->Bajo<-**	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	**->Bajo<-**	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 193 Evaluación de riesgo, autenticación MFA (solución).

OWASP Risk Assessment Calculator

Risk Assessment Calculator

Problema: Inyeccion Sql

Factores de probabilidad					
Factores de la amenaza					
Habilidades requeridas	Habilidades de redes y programación. [6]	6	Pérdida de confidencialic	No Applicable [0]	0
Motivo	Posible recompensa [4]	4	Pérdida de integridad	No Applicable [0]	0
Oportunidad	Acceso especial o recursos requeridos [4]	4	Pérdida de disponibilidac	No Applicable [0]	0
Tamaño de poblacion	Usuarios de internet anónimos [9]	9	Pérdida de responsabilidad	Ataque completamente anónimo [9]	9
Factores de la vulnerabilidad			***Factores de impacto empresarial***		
Fácil de descubrir	Dificil [3]	3	Daño financiero	No Applicable [0]	0
Facilidad de explotación	Dificil [3]	3	Daño a la reputación	No Applicable [0]	0
Conciencia	Oculto [4]	4	Incumplimiento	No Applicable [0]	0
Detección de intrusiones	No registrado [9]	9	Violación de privacidad	No Applicable [0]	0

Puntuación de probabilidad:	**5,25**	**Puntuación de impacto:**	**1,125**

Gravedad general de riesgo :	**Bajo**

	Impacto		
Probabilidad	**->Bajo<-**	Moderado	Alto
Bajo	Note	Bajo	Moderado
->Moderado<-	**->Bajo<-**	Moderado	Alto
Alto	Moderado	Alto	Critico

Figura 194 Evaluación de riesgo, inyección SQL (solución).

BIBLIOGRAFIA.

Burr, W. E., Dodson, D. F., Newton, E. M., Perlner, R. A., Polk, W. T., Gupta, S., & Nabbus, E. A. (2013). *Electronic Authentication Guideline*. https://doi.org/10.6028/NIST.SP.800-63-2

Constitución de la República, Art. 96, 20 de octubre de 2008 (2021). https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/02/Constitucion-de-la-Republica-del-Ecuador_act_ene-2021.pdf

D. M'Raihi. (2005). *RFC 4226 - HOTP: An HMAC-Based One-Time Password Algorithm*. https://datatracker.ietf.org/doc/html/rfc4226

D. M'Raihi. (2011). *RFC 6238 - TOTP: Time-Based One-Time Password Algorithm*. https://datatracker.ietf.org/doc/html/rfc6238

Decreto Legislativo 0, Constitución de la República, Art. 16, Literal 2, 20 de octubre de 2008 (2021). https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/02/Constitucion-de-la-Republica-del-Ecuador_act_ene-2021.pdf

Denso Wave. (2023). *What is a QR Code®?* https://www.denso-wave.com/en/system/qr/fundamental/qrcode/qrc/index.html

De Soete, M. (2011). Two-Factor Authentication. En H. C. A. van Tilborg & S. Jajodia (Eds.), *Encyclopedia of Cryptography and Security* (p. 1341). Springer US. https://doi.org/10.1007/978-1-4419-5906-5_310

Espinoza Tinoco, Lady, Barriga Rivera, B., Izurieta Navarrete, J., & Morales Alarcón, C. H. (2022). *Seguridad informática aplicando la Autentificación por Doble factor para la plataforma HomeOfi. 8*, 503-523. https://doi.org/10.23857/dc.v8i3

Grassi, P. A., Fenton, J. L., Newton, E. M., Perlner, R. A., Regenscheid, A. R., Burr, W. E., Richer, J. P., Lefkovitz, N. B., Danker, J. M., Choong, Y.-Y., Greene, K. K., & Theofanos, M. F. (2017). *Digital identity guidelines: authentication and lifecycle management*. https://doi.org/10.6028/NIST.SP.800-63b

Grassi, P. A., García, M. E., & Fenton, J. L. (2017). *Digital identity guidelines: revision 3*. https://doi.org/10.6028/NIST.SP.800-63-3

IBM. (2021). *Autorización - Documentación de IBM*. https://www.ibm.com/docs/es/ibm-mq/7.5?topic=ssfksj-7-5-0-com-ibm-mq-sec-doc-q009750--htm

IBM MQ. (2023). *Identificación y autenticación - Documentación de IBM*. https://www.ibm.com/docs/es/ibm-mq/9.0?topic=mechanisms-identification-authentication

INCIBE. (2023). *Tematicas Autenticacion Segura Y Donde Empiezo | Empresas | INCIBE*.

ISO 25010. (2022). *Calidad de Software y Datos*. https://iso25000.com/index.php/normas-iso-25000/iso-25010?start=3

L. F. Cranor and S. Garfinkel. (2004a). *Introduction: Secure or Usable? Security & Privacy*. http://lorrie.cranor.org/.

L. F. Cranor and S. Garfinkel. (2004b). *Introduction: Secure or Usable? Security & Privacy*. http://lorrie.cranor.org/.

Ley 0, Código Orgánico Integral Penal, Sección Tercera, Artículos 229-234, 10 de febrero de 2014 (2021). https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/03/COIP_act_feb-2021.pdf

Ley 0, Ley Orgánica de Protección de datos Personales, Artículo 1, 25 de mayo 2021 (2021). https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

Ley No 67, Ley de Comercio Electrónico, firmas y mensajes de datos, Art. 5, 16 de abril de 2002 (2002). https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2012/11/Ley-de-Comercio-Electronico-Firmas-y-Mensajes-de-Datos.pdf

Ley No 67, Ley de Comercio Electrónico, firmas y mensajes de datos, Art. 9, 17 de abril de 2002 (2002). https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2012/11/Ley-de-Comercio-Electronico-Firmas-y-Mensajes-de-Datos.pdf

Mantovani, V. (2019). *Autenticación de Múltiples factores (MFA)*. http://bibliotecadigital.econ.uba.ar/download/tpos/1502-1524_MantovaniV.pdf

Microsoft. (2023). *¿Qué es el control de acceso?* https://www.microsoft.com/es-es/security/business/security-101/what-is-access-control

mitre. (2023). *CWE - CWE Top 25 Most Dangerous Software Weaknesses*. https://cwe.mitre.org/top25/index.html

OWASP. (2024a). *Multifactor Authentication - OWASP Cheat Sheet Series*. https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html

OWASP. (2024b). *OWASP Top Ten | OWASP Foundation*. https://owasp.org/www-project-top-ten/

OWASP. (2024c). *Password Storage - OWASP Cheat Sheet Series*. https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

OWASP. (2024d). *WSTG - v4.2 Web Aplication Security Testing*. https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/

OWASP-Testing-Checklist. (2023). *GitHub - tanprathan/OWASP-Testing-Checklist: OWASP based Web Application Security Testing Checklist is an Excel based checklist which helps you to track the status of completed and pending test cases.* https://github.com/tanprathan/OWASP-Testing-Checklist

Plan de Creación de Oportunidades-2021-2025, 41 (2021). https://www.planificacion.gob.ec/wp-content/uploads/2021/09/Plan-de-Creacio%CC%81n-de-Oportunidades-2021-2025-Aprobado.pdf

Reyes Riveros, A. J., Salinas Meza, J. E., & Mendosa de los Santos, A. C. (2023). *Modelo de Autentificación de Doble Factor*. https://www.redalyc.org/articulo.oa?id=673874721006

Rodríguez Vásquez, J. C. (2023). *SISTEMA DE AUTENTICACIÓN BASADO EN CÓDIGOS QR PARA ACCESO A SERVICIOS PROPORCIONADOS POR LA UTN*.

Xia, L., Yang, Y., & Wang, Y. (2010). A method of evaluating authentication mechanisms. *Innovations and Advances in Computer Sciences and Engineering*, 179-184. https://doi.org/10.1007/978-90-481-3658-2_31