Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Aplicaciones con Tecnología en Internet

METODOLOGÍA PARA LA AUDITORÍA DE SEGURIDAD DE APLICACIONES WEB

Trabajo Especial de Grado
presentado ante la excelentísima
Universidad Central de Venezuela
Por los Bachilleres
Agüero Villamizar, Robert Thaylor
Dorado Jiménez, Manuel Francisco
para optar al título de
Licenciado en Computación

Tutores
Prof. Sergio Rivas
Prof. Walter Hernández

Caracas, Mayo / 2008

Acta

Quienes suscriben, miembros del Jurado por el Consejo de Escuela de Computación de la Facultad de Ciencias, para examinar el Trabajo Especial de Grado presentado por los bachilleres Manuel F. Dorado J. C.I.: 16.672.912 y Robert T. Aguero V. C.I.: 16.971.434, con el título: "Metodología para la auditoría de seguridad de Aplicaciones Web.", a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 27 de Mayo de 2008, a las 12:00 m, para que sus autores lo defiendan en forma pública, lo que se hizo en el aula I de la Escuela de Computación, mediante la presentación oral de su contenido, luego de la cual respondieron las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlos.

En fe de lo cual se levanta la presente Acta, en Caracas al 27 de Mayo de 2007, dejándose también constancia que actuó como Coordinador del Jurado el Profesor Tutor Sergio Rivas.

Prof. Sergio Rivas Jurado Principal	Prof. Walter Hernández Jurado Principal
Prof. Antonio Leal Jurado Principal	Prof. Andrés Sanoja Jurado Principal

Agradecimientos y Dedicatoria

Quisiera agradecer a mis padres y a mis abuelos, de manera muy especial, a mi madre y a mi abuela, dos mujeres luchadoras que siempre le han dado un sentido a mi vida al enseñarme, apoyarme y educarme todos estos años y procurar que ante todos los inconvenientes, saliera adelante. Por su incondicional apoyo, y por darme lo que me merecía en el momento justo.

Agradezco a mi compañero de Tesis, Manuel F. Dorado J., por haberme dado la oportunidad de nutrirme de todos sus conocimientos, por sus críticas, su objetividad y su incansable amistad, Te quiero hermano.

Agradezco a nuestros tutores por ayudarnos y respaldarnos en todo momento durante el desarrollo del Trabajo Especial de Grado.

A mis hermanos, familiares y amigos, que me apoyaron y creyeron en mis capacidades, a pesar de no tener un contacto directo y constante con ellos, en especial a mi tio Gustavo Aguero que me ayudo enormemente.

A todos mis compañeros de trabajo del MPPPAT, en especial a Manuel Gómez, Peter Blanco y Lenin Caro, que me ayudaron en la elaboración de este TEG, y me dieron ánimos para que siguiera adelante.

Finalmente y no por eso menos importante a Dios, por darme fuerza, salud y bienestar.

Robert T. Aguero V.

Agradecimientos y Dedicatoria

Quisiera agradecer a Dios por ser mi guía en todos los momentos difíciles.

Quisiera agradecer a toda mi familia, en especial a mi mamá y a mi hermano Marco. Gracias por haber confiado en mí, por escucharme, por aconsejarme y por haberme apoyado en todas mis decisiones. Mejor madre imposible. A mi tío Luís Almeida quien siempre me ha apoyado e impulsado valiosamente.

Agradezco a mi mejor amigo y compañero de tesis, Robert Aguero, por su apoyo, trabajo y confianza, así como, por su gran amistad y lealtad incondicional.

Gracias a nuestros tutores, Sergio Rivas y Walter Hernández, por todo su apoyo y por siempre brindarnos la ayuda necesaria para solucionar los problemas, que se nos plantearon durante la realización de la investigación.

Finalmente quisiera agradecer a todos mis amigos, personal de la Escuela de Computación y de Control de Estudios por su compañía, consejos y ayuda en esta gran carrera. Por siempre creer en mí, apoyarme y por darme ánimos y hacer de estos años de estudio una experiencia que nunca olvidaré. Siempre los recordaré.

Manuel F. Dorado J.

Resumen

Las aplicaciones web con frecuencia resultan ser el punto de entrada más débil a una red corporativa. Generalmente el acceso a las mismas debe ser público y estar disponible desde cualquier parte del mundo, 24 horas al día, los 365 días del año.

Si la aplicación web no es segura, entonces toda la información sensible esta en grave peligro. Asegurar las aplicaciones web es crítico y no es una tarea fácil, debido a la inmensa cantidad de vulnerabilidades y amenazas a las que están expuestas.

Para las aplicaciones web existen vulnerabilidades en la red, el sistema operativo, servidores que las contienen, e incluso en las mismas aplicaciones web. Un atacante puede aprovechar estas vulnerabilidades y cometer ataques, con el fin de obtener información confidencial, conseguir accesos no autorizados o simplemente causar algún daño.

Sin embargo existen técnicas y buenas prácticas de desarrollo de software que pueden ayudar a proteger las aplicaciones web de estas vulnerabilidades. Siempre y cuando los desarrolladores, integradores y administradores de sistemas, estén concientes de la existencia de estas vulnerabilidades y de cómo prevenirlas o dificultar su explotación.

El Trabajo Especial de Grado aborda las aplicaciones web desde el punto de vista de la seguridad. Describe las principales características de las aplicaciones web, sus vulnerabilidades y amenazas, y las medidas de seguridad que permiten protegerlas. Se consideran aspectos de plataforma, red, aplicaciones e implementaciones de funcionalidades en las aplicaciones web. Se propone una metodología para la auditoría de seguridad en aplicaciones web. Esta metodología establece un conjunto de pasos bien definidos, los cuales permitirán auditar cada característica de una aplicación web y determinar la existencia de posibles vulnerabilidades y amenazas de seguridad. Para luego hacer referencia a las respectivas medidas de seguridad correctivas que pueden ser aplicadas. Finalmente para probar la eficacia de la metodología se realizaron tres (3) casos de estudio, con los cuales se auditaron tres (3) aplicaciones web desarrolladas con tecnologías diferentes.

Índice general

1. Introducción y Problema **2**
- 1.1. Propuesta . . . 3
- 1.2. Objetivo General . . . 3
- 1.3. Objetivos Específicos . . . 3
- 1.4. Alcance . . . 4

2. Introducción a las aplicaciones web y la seguridad **6**
- 2.1. Aplicaciones web . . . 6
  - 2.1.1. Protocolo HTTP y la red . . . 7
  - 2.1.2. El Cliente . . . 8
  - 2.1.3. Servidor . . . 8
- 2.2. Seguridad en las aplicaciones web . . . 10
  - 2.2.1. Seguridad de la información . . . 11
  - 2.2.2. Requerimientos básicos de seguridad de la información . . . 11
  - 2.2.3. Principios de seguridad de aplicaciones . . . 12
- 2.3. Roles y Capas de una aplicación web . . . 14

3. Amenazas y Vulnerabilidades **17**
- 3.1. Capa Física . . . 17
  - 3.1.1. Catástrofes naturales . . . 17
  - 3.1.2. Deficiente control de acceso a la sala de servidores . . . 17
  - 3.1.3. Deficiente control de acceso a las computadoras de trabajo pertenecientes a la red de la compañía . . . 18
  - 3.1.4. Deficiente protección de las copias de seguridad . . . 18
  - 3.1.5. Permitir utilizar cualquier tipo de dispositivo de almacenamiento en equipos con información importante . . . 18
- 3.2. Capa de Red . . . 18
  - 3.2.1. Deficiente protección de los datos en el tránsito . . . 18
  - 3.2.2. Acceso a recursos privados . . . 19
  - 3.2.3. Suplantación de IP (IP Spoofing) . . . 20
  - 3.2.4. Suplantación de DNS (DNS Spoofing) . . . 20
  - 3.2.5. Suplantación de una página web real . . . 21
  - 3.2.6. Denegación de servicio (DOS) . . . 21
- 3.3. Capa de Sistema Operativo y Servicios . . . 21
  - 3.3.1. Deficiente protección de directorios y archivos . . . 21
  - 3.3.2. Servicios ejecutándose con cuentas privilegiadas . . . 22
  - 3.3.3. Generar cuentas de administradores sin contraseña o con contraseña débiles . 22

3.3.4. Servicios en texto plano . . . 22
3.3.5. Utilizar versiones viejas e inseguras de los servicios . . . 23
3.3.6. Denegación de servicios del SO . . . 23
3.3.7. Instalación y Ejecución de servicios que no son útiles . . . 24
3.3.8. Configuraciones por defecto . . . 24
3.3.9. Capturadores de teclas – (Key Loggers) . . . 25
3.3.10. Uso de versiones "viejas" y/o inseguras . . . 25
3.3.11. Ataque huella digital HTTP (HTTP fingerprint) . . . 26
3.3.12. Servidor web en modo privilegiado . . . 26
3.3.13. Funcionalidades que no se necesitan habilitadas . . . 26
3.3.14. Exponer archivos sensibles en el directorio web raíz . . . 26
3.3.15. Exponer archivos fuentes en el directorio web raíz . . . 27
3.3.16. Exponer archivos de configuración . . . 28
3.3.17. Configuración incorrecta de permisos en directorios y archivos . . . 28
3.3.18. Inyección de código SQL . . . 29
3.3.19. Utilizar versiones inseguras de los manejadores de base de datos . . . 29
3.3.20. Vulnerabilidades en los driver conectores a base de datos . . . 29
3.3.21. Información sensible almacenada incorrectamente . . . 30
3.3.22. Exponer el servidor de base de datos . . . 30
3.3.23. Exponer una base de datos que sea crítica . . . 30
3.3.24. Usuarios con control total del repositorio de datos . . . 31
3.3.25. Uso de credenciales predecibles o por defecto . . . 31
3.3.26. Usar binarios corruptos . . . 31
3.4. Capa de Aplicación . . . 32
3.4.1. Ataques que afectan la autenticación . . . 32
3.4.2. Ataques de fuerza bruta . . . 32
3.4.3. Autenticación insuficiente . . . 32
3.4.4. Débil validación en la recuperación de contraseñas . . . 33
3.4.5. Ataques que afectan la autorización . . . 33
3.4.6. Predicción de credenciales/sesión (Session hijacking) . . . 33
3.4.7. Expiración de sesión insuficiente . . . 33
3.4.8. Fijación de sesión (Session fixation) . . . 34
3.4.9. Ataques en la parte cliente . . . 34
3.4.10. Suplantación de contenido . . . 34
3.4.11. Ejecución de código forzado (Cross-site scripting) . . . 35
3.4.12. Forjar peticiones de un sitio a otro (cross-site request forgery) . . . 35
3.4.13. Ataques que permiten ejecución de comandos . . . 35
3.4.14. Desbordamiento de buffer . . . 36
3.4.15. Ataques de formato de cadena . . . 36
3.4.16. Inyección LDAP . . . 36
3.4.17. Ejecución de comandos de sistema operativo . . . 36
3.4.18. Inyección de código SQL . . . 37
3.4.19. Inyección de código server side include SSI . . . 37
3.4.20. Inyección XPath . . . 37
3.4.21. Ataques de revelación de información . . . 38
3.4.22. Fuga de información . . . 38

3.4.23. Subida de directorios (path traversal) . . . 38
3.4.24. Localización de recursos predecibles . . . 39
3.4.25. Referencia directa de objetos . . . 39
3.4.26. Ataques lógicos . . . 39
3.4.27. Abuso de funcionalidad . . . 39
3.4.28. Ataques de denegación de servicios . . . 40
3.4.29. Anti-automatización insuficiente . . . 40
3.4.30. Validación de proceso insuficiente . . . 40

4. Medidas de Seguridad y buenas prácticas 41
4.1. Capa Física . . . 41
4.1.1. Establecer políticas de redundancia de la información sensible de la empresa 41
4.1.2. Detección y corrección de fallas a nivel físico . . . 41
4.1.3. Bloquear el acceso a la sala de servidores . . . 42
4.1.4. Establecer vigilancia . . . 42
4.1.5. Asegurar que los dispositivos más vulnerables se encuentran en un lugar con díficil acceso . . . 42
4.1.6. Utilizar rack para colocar los servidores . . . 42
4.1.7. Controlar el acceso a computadoras ociosas . . . 43
4.1.8. Proteger las copias de seguridad . . . 43
4.1.9. Desactivar unidades que permitan almacenamiento en equipos con información crítica . . . 43
4.2. Capa de Red . . . 44
4.2.1. Uso de Cortafuegos (Firewall) . . . 44
4.2.2. Uso de VPN . . . 45
4.2.3. Red Privada con direcciones internas . . . 46
4.2.4. Uso de sistemas de detección de intruso o ataques (IDS - Intrusion Detection) 46
4.2.5. Bloqueos de IP y limpiado de Buffer . . . 46
4.3. Capa de Sistemas Operativos y Servicios . . . 46
4.3.1. Uso de Antivirus . . . 46
4.3.2. Actualizaciones periódicas . . . 47
4.3.3. Permisos de ejecución y cuentas de usuario limitadas . . . 47
4.3.4. Eliminación de servicios y configuraciones innecesarias . . . 47
4.3.5. Implementar políticas para crear contraseñas seguras . . . 47
4.3.6. Implementar técnicas de encriptación para los datos manejados en servicios críticos . . . 47
4.3.7. Usar la última versión estable . . . 48
4.3.8. Deshabilitar todo lo que no sea necesario . . . 48
4.3.9. Revisar accesos por defecto o comunes . . . 48
4.3.10. Actualizaciones frecuentes y parches . . . 48
4.3.11. Blindar el modulo CGI o Server side scripting tecnology . . . 48
4.3.12. Colocar en el directorio web raíz sólo archivos públicos . . . 48
4.3.13. Uso de Directorios privados dentro del directorio web raíz . . . 49
4.3.14. Correcta configuración de permisos en los directorios y archivos . . . 49
4.3.15. Cifrar datos sensibles . . . 49
4.3.16. Realizar copias de seguridad de los datos . . . 49

4.3.17. Actualizar periódicamente los manejadores de bases de datos . . . 49
4.3.18. Limitar adecuadamente el acceso a los datos . . . 49
4.3.19. Proveer vistas del repositorio de los datos . . . 50
4.3.20. Chequeo de configuración desde cero . . . 50
4.3.21. Verificación de credenciales y permisos . . . 50
4.3.22. Forzar el uso de contraseñas seguras . . . 50
4.4. Capa de Aplicación . . . 50
4.4.1. Validar las entradas de datos . . . 50
4.4.2. Filtrar la entrada de datos (Filter Input) . . . 51
4.4.3. Escapado de salida (escape output) . . . 51
4.4.4. Uso de herramientas para distinguir autómatas de humanos (CAPTCHA) . . . 52
4.4.5. Detección de comportamiento irregular . . . 52
4.4.6. Control de ataques de fuerza bruta . . . 53
4.4.7. Inducir al usuario hacia la seguridad . . . 53
4.4.8. Uso de contraseñas seguras . . . 53
4.4.9. Caducidad de las contraseñas . . . 54
4.4.10. Proveer al usuario múltiples consejos de seguridad . . . 54
4.4.11. Eficiente uso de la manipulación de la información . . . 54
4.4.12. Eficiente protección en la transición de la información . . . 55
4.4.13. Eficiente almacenamiento de la información . . . 55
4.4.14. Garantizar la mejor forma de utilización de la información . . . 55
4.4.15. Verificar la autenticación de un usuario en todo momento . . . 55
4.4.16. Seguridad en profundidad para la recuperación en contraseñas . . . 56
4.4.17. Establecer controles de sesión en procesos críticos . . . 56

5. Características principales de las aplicaciones web y sus consideraciones de seguridad 58
5.1. Capa Física . . . 59
5.1.1. Hardware propio de la empresa . . . 59
5.1.2. Hardware arrendado a terceros . . . 59
5.1.3. Administración directa en el equipo . . . 60
5.2. Capa de red . . . 60
5.2.1. Equipos de acceso público a través de la red . . . 60
5.2.2. Equipos de acceso privado . . . 61
5.3. Capa de sistema operativo y servicios . . . 61
5.3.1. Sistema operativo . . . 61
5.3.2. Hospedaje compartido . . . 61
5.3.3. Administración del sistema en forma remota . . . 63
5.3.4. Actualizaciones remotas . . . 64
5.3.5. Actualizaciones frecuentes . . . 65
5.3.6. Servidores de aplicación o servidores web . . . 65
5.3.7. Servidores de correo . . . 66
5.3.8. Servidores de base de datos . . . 66
5.3.9. Servidores FTP . . . 67
5.3.10. Otros servicios . . . 68

5.3.11. Tecnologías del lado del servidor para contenido dinámico 68
5.4. Capa de Aplicación 69
5.4.1. Inicio de sesión de usuario vía formulario HTML 69
5.4.2. Recuperación de contraseña 70
5.4.3. Administradores remotos 71
5.4.4. Registro de usuarios en línea 72
5.4.5. Usuarios mantienen sesión 73
5.4.6. Usuarios realizan operaciones 74
5.4.7. Sitios web que se comunican con usuarios 75
5.4.8. Pase de parámetros accesibles al usuario 76
5.4.9. Manipulan información sensible 78
5.4.10. Uso de módulos o aplicaciones de terceros del lado del cliente 78
5.4.11. Uso de módulos o aplicaciones de terceros del lado del servidor 80

6. Metodología **82**

7. Conclusiones y Consideraciones **102**
7.1. Conclusiones 102
7.2. Consideraciones 104
7.3. Aportes del trabajo 104
7.4. Trabajos a Futuro 105

Índice de figuras

1.1. Diagrama de la solución (Elaboración propia) 4

2.1. Arquitectura cliente-servidor (Elaboración propia) 7
2.2. Componentes involucrados al realizar peticiones utilizando distintos protocolos (Elaboración propia) 8
2.3. Diagrama con la arquitectura de un cliente web y sus funciones (Elaboración propia) 8
2.4. Componentes de software principales en los servidores web (Elaboración propia) . . 9
2.5. Diagrama de peticiones lado cliente y lado servidor (Elaboración propia) 9
2.6. Modelo de capas desde el punto de vista de la seguridad (Elaboración propia) . . . 16

3.1. Información que muestra los datos capturados por un sniffer (Elaboración propia) . 19
3.2. Suplantación de la IP origen (Elaboración propia) 20
3.3. Servicios que contienen información sensitiva en texto plano (Elaboración propia) . 23
3.4. Exposición de archivos de configuración en el directorio raíz de un servidor web IIS (Elaboración propia) 28
3.5. Creación de un usuario en mysql con todos los privilegios (Elaboración propia) . . . 31
3.6. Ejemplo de fijación de sesión (session fixation) (Elaboración propia) 34
3.7. Ejemplo de vulnerabilidad XSS en You Tube (Imagen tomada de http://cordobo.com/wp-content/youtube-xss-cordobo.png) 35
3.8. Ejemplo de un intento de inyección de sql (Elaboración propia) 37
3.9. Fuga de información (Elaboración propia) 38

4.1. Ejemplo de rack para alojar servidores (Elaboración propia) 43
4.2. Ejemplo de una DMZ (Elaboración propia) 44
4.3. Ejemplo de implementación CAPTCHA (Elaboración propia) 52
4.4. Bloqueo de cuenta, medida que controla ataques de fuerza bruta (Imagen tomada de https://ws027.juntadeandalucia.es/profesionales/eatencion/ayuda/img/UsuBloq1dia.gif) 53
4.5. Ejemplo de inducir al uso de contraseñas seguras (Elaboración propia) 54
4.6. Ejemplo de Proveer al usuario múltiples Consejos de seguridad (Elaboración propia) 54
4.7. Ejemplo de Seguridad en profundidad para la recuperación en contraseñas (Elaboración propia) 56
4.8. Ejemplo de revalidación de credenciales de sesión (Elaboración propia) 57

5.1. Diferencia entre el protocolo http 1.0 y http1.1 (Elaboración propia) 62
5.2. Implementación de una VNC (Elaboración propia) 64
5.3. Un cliente de correo web, en este caso google mail o gmail (Elaboración propia) . . 69
5.4. Una aplicación de banca electrónica (banco Banesco) (Elaboración propia) 70
5.5. Recordar contraseña de la aplicación del Banco de Venezuela (Elaboración propia) . 71

5.6. Panel de administración remota de la página de bebek (Elaboración propia) 72
5.7. Registro de usuarios en línea de la página 123.com.ve (Elaboración propia) 73
5.8. Imagen de publicación de un anuncio (Elaboración propia) 74
5.9. Imagen de vista del anuncio publicado en la pagina web (Elaboración propia) 75
5.10. Email enviado por banesco a sus usuarios (Elaboración propia) 76
5.11. Error al cambiar manualmente un valor en la URL (Elaboración propia) 77
5.12. Uso de plugin de flash en una pagina web (Elaboración propia) 79

7.1. Disposición de la página (Elaboración propia) 107
7.2. Diagrama de la red 109
7.3. Registro de clientes nuevos (Elaboración propia) 114
7.4. Enviar información de contacto (Elaboración propia) 114
7.5. Subir fotos al servidor (Elaboración propia) 114
7.6. Llenar información de contenido de la página (Elaboración propia) 114
7.7. Recordar Contraseña (Elaboración propia) 115
7.8. Diagrama de caso de uso de nivel 0 (Elaboración propia) 115
7.9. Mensaje de error cuando el password de un usuario no corresponde (Elaboración propia) 122
7.10. Campo password guardado en base de datos en modo texto (Elaboración propia) . . 122
7.11. Ejemplo de la información mostrada al colocar la URL sin autenticar un usuario (Elaboración propia) 124
7.12. Vulnerabilidad XSS presente en la aplicación (Elaboración propia) 124
7.13. Implementación del captcha luego de realizar 2 intentos fallidos (Elaboración propia) 128
7.14. Código fuente de la página index.asp (Elaboración propia) 129
7.15. Fragmentos de código donde se unifica el mensaje (Elaboración propia) 129
7.16. Contraseña guardada en la base de datos aplicandole la función md5 (Elaboración propia) 130
7.17. Funciones para evitar una inyección de códigos o comando (Elaboración propia) . . 130
7.18. Implementación de la página que verifica la sesión de un usuario (Elaboración propia) 130
7.19. Código de panel/administrar_articulos.asp (Elaboración propia) 131
7.20. Resultado obtenido luego de filtrar datos de entrada y escapar datos de salida (E-laboración propia) 131
7.21. Disposición de la página (Elaboración propia) 132
7.22. Diagrama de la red 135
7.23. Registro de clientes nuevos. (Elaboración propia) 139
7.24. Llenar información de contenido de la página. (Elaboración propia) 140
7.25. Mensaje de error cuando el usuario no se encuentra en la base de datos (Elaboración propia) 146
7.26. Valores del campo password en la base de datos (Elaboración propia) 147
7.27. Uso de la función nativa de PHP pg_escape_string() (Elaboración propia) 147
7.28. Formulario para agregar usuarios (Elaboración propia) 148
7.29. Vulnerabilidad XSS presente en la aplicación (Elaboración propia) 149
7.30. Código fuente del archivo modelo/phpcaptcha.php (Elaboración propia) 152
7.31. Código fuente del archivo control/usuario/inicio_sesion.php (Elaboración propia) . . 152
7.32. Implementación de la página que verifica la sesión de un usuario (Elaboración propia) 153
7.33. Resultado de utilizar la función html_entities (Elaboración propia) 153
7.34. Diagrama de la red 156

7.35. Estructura de Directorio y recursos de la aplicación 160
7.36. Pantalla registro sin CAPTCHA ni protección SSL 166
7.37. Acceso a la aplicación a través de una dirección IP 166
7.38. Manejo de Excepciones inapropiado, error 500 167
7.39. Acceso a información sensible manipulando los parametros 167
7.40. Información de correo de confirmación 168
7.41. Código donde se solicita unicamente Nro Solicitud (`requesid`) 170
7.42. Incorporación de captcha en la aplicación (`requestid`) 172
7.43. Correo donde se envía Nro. de Solicitud y Nro. de confirmación(`requestid`) 173
7.44. Código donde se verifica tanto Nro. de Solicitud como el Nro. de confirmación(`requesid`)173
7.45. Parámetro con confirmación (`requestid`) 174
7.46. Manejo correcto de errores(`requesid`) 174
7.47. Formulario para recuperar contraseña, Onidex (Elaboración propia) 176
7.48. Email enviado por la aplicación web, especificando la nueva contraseña (Elaboración propia) 177
7.49. Enlace de ¿Olvido su clave? en Mercantil en línea (Elaboración propia) 177
7.50. Ventana emergente al presionar el enlace ¿Olvido su clave? (Elaboración propia) . . 178

Índice de cuadros

3.1. Tiempo requerido para un ataque de fuerza bruta. Computadora a una rata de 500.000 contrasenas por segundo 32

Capítulo 1

Introducción y Problema

Las actividades que se realizan en Internet tienen mayor importancia cada día. En todas partes del mundo las personas diariamente navegan en busca de información, revisan su email personal, compran en sitios de comercio electrónico, realizan consultas y transferencias bancarias. La seguridad de estas actividades es de vital importancia para el completo desarrollo de las mismas. Los usuarios deben sentirse confiados y seguros para realizar sus operaciones a través de Internet.

Muchas veces la seguridad se ve amenazada por vulnerabilidades presentes en las aplicaciones web. Estas vulnerabilidades son objeto de ataques por personas mal intencionadas que buscan hacerse con información confidencial, cometer estafas, fraudes o simplemente causar daño a una persona o empresa. La falta de prevención, el descuido y desconocimiento, son las principales razones por las cuales estas vulnerabilidades existen.

Es de vital importancia considerar la implementación de controles o medidas de seguridad en función de evitar o minimizar el riesgo de estos ataques en las aplicaciones web. En ocasiones durante el desarrollo e implantación de una aplicación web, el tópico de medidas de seguridad, no es tomado en cuenta en la planificación y desarrollo del proyecto, y se deja como un aspecto secundario o para cuando la aplicación y sus funcionalidades estén finalizadas. Por esto a veces es muy tarde cuando se requiere implementar una solución a una vulnerabilidad, ya habiendo pagado el precio de recibir un ataque.

En la actualidad existe una gran cantidad de aplicaciones web que presentan vulnerabilidades y posibilidades de ataques que pueden poner en riesgo a las empresas y sus clientes. Es necesario identificar a tiempo las vulnerabilidades y amenazas a las que están expuestas estas aplicaciones, e implementar los controles y medidas de seguridad para prevenirlas.

Las aplicaciones hechas a la medida tienden a ser más susceptibles a presentar vulnerabilidades, ya que en muchas ocasiones las mismas implican un menor grado de pruebas o exigencias de calidad, comparadas con otras aplicaciones empaquetadas por grandes casas de software para un mercado masivo, que necesariamente deben ser probadas miles de veces para poder ser lanzadas al mercado.

Es muy importante saber que la seguridad no es un evento de una sola vez. Ya que, constantemente se descubren nuevos ataques, nuevas vulnerabilidades en procesos existentes. Por ende, se tienen que evaluar constantemente los procesos de forma rigurosa para prevenir y descartar cualquier posibilidad de amenaza o ataque.

1.1. Propuesta

Se propone un estudio científico que tenga como objetivo identificar las características y funcionalidades más comunes presentes en aplicaciones web y determinar a cuales vulnerabilidades y amenazas de seguridad están expuestas. Elaborar una metodología para la auditoria de seguridad en las aplicaciones web, que permita identificar las características y funcionalidades que posee una determinada aplicación web, para verificar la existencia o no, de amenazas y vulnerabilidades, y así poder corregirlas. Luego aplicar la metodología en diferentes aplicaciones web, con el fin de identificar sus vulnerabilidades y amenazas e implementar las medidas de seguridad correctivas correspondientes.

1.2. Objetivo General

Investigar las características y funcionalidades más comunes presentes en aplicaciones web y determinar a cuales vulnerabilidades y amenazas de seguridad están expuestas. Elaborar una metodología para la auditoria de seguridad en aplicaciones web. Aplicar la metodología a diferentes aplicaciones web para determinar sus posibles vulnerabilidades y amenazas. Luego implementar las respectivas medidas de seguridad y correcciones necesarias en las mismas.

1.3. Objetivos Específicos

El trabajo a realizar cuenta con los siguientes objetivos específicos:

1. Identificar las características y funcionalidad más comunes presentes en aplicaciones web.
2. Para cada una de las características determinar implicaciones en cuanto a vulnerabilidades y amenazas de seguridad a las que puedan estar expuestas.
3. Elaborar un conjunto de pasos que ayuden a determinar las características presentes en una aplicación web y verificar sus vulnerabilidades, es decir, elaborar una metodología para la auditoria de seguridad en aplicaciones web.
4. Realizar casos de estudio para comprobar la utilidad de la metodología, específicamente, aplicar la metodología en diferentes aplicaciones web con el fin de identificar características y sus posibles amenazas de seguridad.
5. Implementar soluciones para las posibles vulnerabilidades que presenten las aplicaciones como complemento del caso de estudio.

1.4. Alcance

Identificar las características principales más comunes en aplicaciones web y determinar las posibles vulnerabilidades y amenazas de seguridad asociada a cada una de ellas. Establecer el conjunto de pasos necesarios para determinar las vulnerabilidades presentes en las aplicaciones web. En la actualidad el número de vulnerabilidades existentes es muy grande y está en constante aumento, por tanto es imposible considerar todas las posibilidades de vulnerabilidades y amenazas. Se tomaran las más importantes y comunes presentes en las aplicaciones web, sin embargo los conceptos y principios que comprenden a las mismas son perfectamente aplicables en la mayoría de las variantes y nuevas vulnerabilidades que van surgiendo con el tiempo.

Este trabajo toma en cuenta las aplicaciones web y su entorno. Considera aspectos de plataforma, red, sistemas operativos, aplicaciones como servidores web, servidores de bases de datos e incluso la forma cómo están desarrolladas las mismas. Se hace mayor énfasis en aspectos relacionados el código fuente, implementaciones de procesos y el manejo de la información en la aplicación web.

Figura 1.1: Diagrama de la solución (Elaboración propia)

Para cumplir estos objetivos se desarrollaron seis capítulos: En el primer capítulo se realiza una breve introducción del contenido del TEG, se describe el problema que generó este estudio. Para puntualizar el objetivo general y objetivos específicos, finalmente, se precisa el alcance que logra el TEG.

En el segundo capítulo se realiza un breve análisis de las aplicaciones web y su entorno. También se discuten los conceptos de seguridad relacionados con las aplicaciones web a la vez que se familiariza al lector con algunos términos que le facilitaran la comprensión del presente TEG.

En el tercer capítulo se describe las vulnerabilidades y amenazas de seguridad más comunes, presentes en aplicaciones web. ¿Cuándo? y ¿Cómo? se presentan. En cada caso se hace referencia a las medidas de seguridad que deben tomarse para evitarlas o prevenirlas.

En el cuarto capítulo se definen cada una las medidas de seguridad necesarias para corregir las vulnerabilidades y amenazas de seguridad mencionadas en el segundo capítulo. Consideraciones,

políticas y controles que deben implementarse en conjunto con las funcionalidades y características de una aplicación web.

En el quinto cuatro se identifican las características principales en las aplicaciones web y sus implicaciones de seguridad. Características que frecuentemente se pueden apreciar en distintas aplicaciones web, y que muchas veces implican la existencia de vulnerabilidades, y por tanto se deben tomar las medidas de seguridad necesarias.

El capítulo seis es la propuesta de una metodología de auditoria para la seguridad en aplicaciones web. Se mencionan las herramientas y conocimientos necesarios para una correcta aplicación de la misma y se describe paso a paso cada una de sus etapas.

Finalmente, se realizaron tres (3) casos de estudios en los cuales fueron auditadas, haciendo uso de la metodología, tres aplicaciones web diferentes. Para cada una de ellas se determinó la existencia de vulnerabilidades y amenazas de seguridad. Luego se implementaron las medidas o controles de seguridad necesaria para protegerlas.

Capítulo 2

Introducción a las aplicaciones web y la seguridad

En este capítulo se describe las aplicaciones web y su entorno. Un breve repaso a sus componentes de hardware y software más notables. También se discuten los conceptos de seguridad relacionados con las aplicaciones web a la vez que se familiariza al lector con algunos términos que le facilitaran la comprensión del presente Trabajo Especial de Grado (TEG).

2.1. Aplicaciones web

Una aplicación web es una aplicación que es accedida a través de una red como Internet o una intranet, generalmente haciendo uso de un navegador web (browser, en inglés). Los sitios web, también conocidos como páginas web, son ejemplos de aplicaciones web en las que generalmente hay información en textos e imágenes. Los navegadores son considerados clientes livianos, ya que no requieren la instalación de grandes paquetes de software, y en la actualidad están presentes en la mayoría de los computadores. Las aplicaciones web se ejecutan en servidores, y los clientes no necesitan instalar y/o actualizar ningún tipo de software para tener acceso a las mismas. Estas características de actualización y administración centralizada, evita tener que distribuir e instalar numerosas actualizaciones de software en múltiples clientes. Esto, junto con las capacidades de ubicuidad de los clientes, son las principales razones por las cuales se han vuelto tan populares las aplicaciones web.

Las aplicaciones web funcionan bajo una arquitectura cliente-servidor. Bajo esta arquitectura se define una relación entre dos computadores sobre una red, los clientes que realizan peticiones a un servidor, y estos que aceptan las peticiones, las procesan y responden con la información solicitada a los clientes. Bajo este modelo cada cliente tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura. El intercambio de información se realiza principalmente a través del protocolo HTTP.

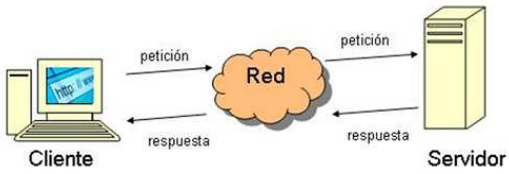


Figura 2.1: Arquitectura cliente-servidor (Elaboración propia)

2.1.1. Protocolo HTTP y la red

El protocolo HTTP (Hypertext transfer protocol), o Protocolo para transferencia de hipertextos permite enviar y recibir información como textos y archivos. Define las reglas de comunicación entre el cliente y el servidor. En el protocolo se definen diferentes métodos para las peticiones. Los métodos GET y POST son los más utilizados, y sirven para enviar información y solicitar recursos al servidor.

El formato preferido para el intercambio es código HTML (Hypertext markup language), o lenguaje de marcado de hipertextos, el cual es un lenguaje de marcado, que define un conjunto de etiquetas para marcar su contenido, las cuales le dicen al cliente web como interpretar y mostrar el mismo.

Generalmente los servidores tienen que ser de acceso público a través de una red como Internet. Su propósito es servir los recursos solicitados a través de una URL.[1]. Para que estos servidores sean alcanzables desde Internet, deben tener una dirección IP[2] pública, la cual muchas veces esta asociada a un nombre de dominio, porque resulta más fácil de recordar que una dirección IP numérica. Esto gracias a los sistemas de nombres de dominio o DNS (Domain Name System), los cuales permiten traducir un nombre de dominio a su correspondiente dirección IP.

Se ha convertido en un estándar que los servidores web, presten sus servicios a través del puerto 80 para conexiones HTTP regulares y el puerto 443 para conexiones HTTP seguras, mediante el protocolo HTTPS (HTTP sobre SSL/TLS[3]). Adicionalmente estos servidores se protegen haciendo uso de equipos de firewall o corta fuegos. Que evitan el acceso a servicios protegidos en el servidor, tales como un servidor de base de datos.

[1]URL: (Localizador Uniforme de Recurso) es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en la Internet. El termino URL ha sido incorporado dentro de un termino más general, el de URI, que significa Uniform Resource Identifier - Identificador Uniforme de Recurso.

[2]Dirección IP: es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol).

[3]SSL/TLS: son protocolos criptográficos que proporcionan comunicaciones seguras en Internet

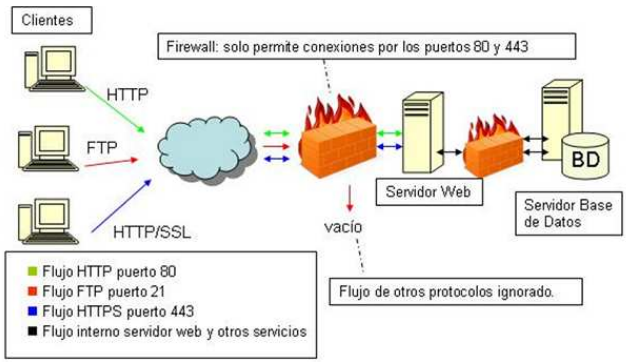


Figura 2.2: Componentes involucrados al realizar peticiones utilizando distintos protocolos (Elaboración propia)

2.1.2. El Cliente

En las aplicaciones web, el cliente puede ser cualquier computadora o dispositivo móvil con un navegador web. Los navegadores web son programas que muestran el código HTML de una forma legible para los usuarios. Tienen capacidades para mostrar textos, imágenes y vídeo. Adicionalmente poseen poder de cómputo, gracias a tecnologías como Javascript[4], que se encuentran disponibles por defecto al momento de instalar algún navegador, u otros elementos, que permiten que el contenido mostrado no sea del todo estático, una vez se haya recibido la respuesta del servidor.

Estas tecnologías son conocidas como tecnologías del lado del cliente, se presentan en forma de complementos y permiten, entre otras cosas, cambiar el contenido de la página dinámicamente, reproducir multimedia (audio y vídeo) y mostrar otros formatos como documentos pdf. Actúan como clientes del protocolo HTTP para el intercambio de información con servidores web. Algunos navegadores web más utilizados son Internet Explorer, Mozilla Firefox, Opera y Safari.

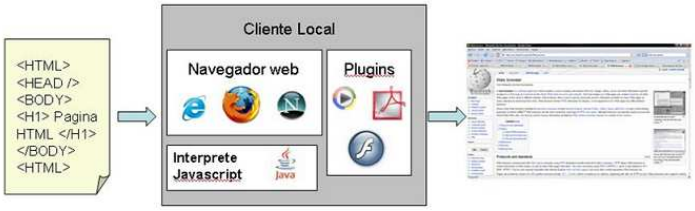


Figura 2.3: Diagrama con la arquitectura de un cliente web y sus funciones (Elaboración propia)

2.1.3. Servidor

Los servidores son dispositivos en los que se ejecutan programas destinados a recibir las peticiones de los clientes y servir las respuestas a los mismos. En las aplicaciones web encontramos principalmente servidores web, los cuales implementan el protocolo HTTP para enviar y recibir

[4]JavaScript: es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, y permite el comportamiento dinámico de las mismas.

respuestas. Aplicaciones como Apache, Tomcat, IIS, son ejemplos de servidores web.

En los equipos servidores también podemos encontrar programas como servidores de base de datos, servidores de correo, servidores de autenticación entre otros. Estos funcionan de forma similar a los servidores web, sólo que en lugar de implementar el protocolo HTTP, implementan otros protocolos como SMTP para servidores de correo, FTP para transferencia de archivos o protocolos propios, de acuerdo a lo que estén destinados a servir. Tal es el caso de los servidores con servicios manejadores de bases de datos como MySQL u Oracle.

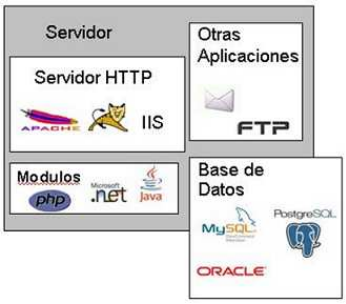


Figura 2.4: Componentes de software principales en los servidores web (Elaboración propia)

En sus inicios las páginas web eran por naturaleza estáticas. Únicamente consistían en un conjunto de textos e imágenes fijas, e iguales para todos los clientes que las solicitaban. Con el tiempo se desarrollo lo que hoy se conoce como tecnologías del lado del servidor. Las mismas permitieron procesar y generar en forma dinámica, para cada cliente, la información contenida en cada respuesta HTML. Haciendo uso del poder computacional en el servidor, en lenguajes como PHP, ASP, Java, entre otros.

Este poder en el lado del servidor le permite no solamente generar dinámicamente el contenido para cada cliente, sino también interactuar con servidores de bases de datos, servidores de correo e incluso otras aplicaciones.

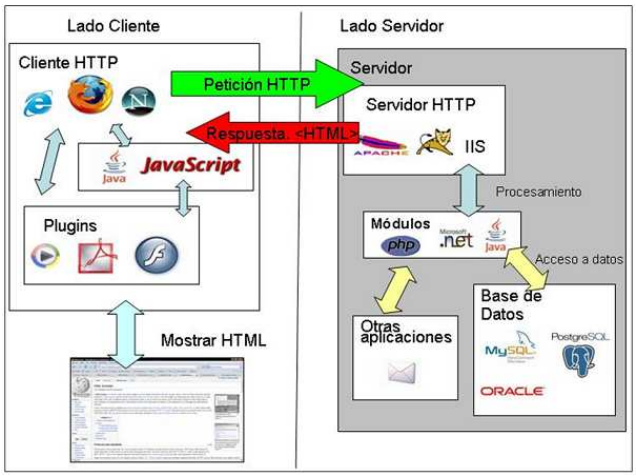


Figura 2.5: Diagrama de peticiones lado cliente y lado servidor (Elaboración propia)

Todo esto sumado le da grandes capacidades a las aplicaciones web. Hoy en día las empresas delegan responsabilidades importantes del negocio a las aplicaciones web, e incluso existen algunas empresas cuyo negocio depende exclusivamente de Internet. Pero con la creciente complejidad de las aplicaciones web y sus funcionalidades, también viene un crecimiento de la necesidad de protegerlas y garantizar la seguridad a los clientes.

2.2. Seguridad en las aplicaciones web

La seguridad en Aplicaciones Web es un aspecto de vital importancia, debido a la información sensible que manejan. Los sitios web y sus aplicaciones web relacionadas deben estar disponibles 24 x 7 para proveer un buen servicio a sus clientes, empleados y proveedores. Estas aplicaciones tienen acceso directo a bases de datos con información privada de clientes, productos o información financiera. Por tanto, controlan información muy valiosa y difícil de proteger, debido a la inmensa cantidad de vulnerabilidades y amenazas existentes.

La mayoría de estas vulnerabilidades y amenazas son producto del desconocimiento u omisión, en cuanto a medidas de seguridad y controles, por parte de administradores de sistema, personas responsable de la red o desarrolladores de software.

Existen vulnerabilidades en todos los niveles de una aplicación web. Desde los equipos físicos y la red, que permiten la ejecución de las aplicaciones y la comunicación entre clientes y servidores, hasta las aplicaciones web en sí mismas. En otras palabras, las aplicaciones web no sólo se ven expuestas a amenazas de seguridad en el tráfico de la información y vulnerabilidades del sistema operativo, sino que además, existen muchas vulnerabilidades provenientes de como están realizadas estas aplicaciones, la implementación de sus procesos y el lenguaje de programación utilizado en su desarrollo.

El código inseguro o codificación insegura, puede tener implicaciones de seguridad que ningún cortafuegos (firewall) o sistema de detección de intrusos (IDS) esta en capacidad de proteger. Estos elementos no pueden hacer nada contra malas prácticas o descuidos de programación en las aplicaciones web. Existen múltiples vulnerabilidades de seguridad producto de una codificación insegura y malas configuraciones. Y ningún lenguaje de programación o producto de software puede prevenir esto. Es completamente una tarea responsable del desarrollador.

El objetivo final de los atacantes a cambiado de simples actos de vandalismo como un defacement[5], cambiar la apariencia de un sitio web sin autorización, a el robo y obtención de datos corporativos, como números de tarjetas de crédito o información de clientes, por sus lucrativas ganancias de venta en el mercado negro. En otras palabras, los hackers prefieren tener acceso a data sensible ya que pueden vender esta información por grandes sumas de dinero.

[5]Defacement: Cambiar el aspecto de las páginas sin autorización, introduciendo elementos propios, modificando los existentes o eliminando archivos web.

2.2.1. Seguridad de la información

La información hoy en día, es uno de los más importantes activos no sólo para las empresas y organizaciones, sino para cada individuo. Por este motivo la misma requiere ser asegurada y protegida en forma apropiada. La seguridad de la información es el conjunto de metodologías, prácticas y procedimientos que buscan proteger la información como activo valioso, con el fin de minimizar las amenazas y riesgos continuos a los que esta expuesta, a efectos de asegurar la continuidad del negocio, minimizar los daños a la organización y maximizar el retorno de inversiones y las oportunidades del negocio. Y en el caso de cada individuo de proteger la identidad y la privacidad.

Garantizar la confidencialidad, integridad y disponibilidad de la información es uno de los temas más delicados en materia de seguridad.

2.2.2. Requerimientos básicos de seguridad de la información

En toda referencia de seguridad los tres requerimientos principales enunciados son siempre, confidencialidad, integridad y disponibilidad. Los mismos se definen a continuación:

1. **Confidencialidad:** La confidencialidad busca prevenir el acceso no autorizado de la información. La pérdida de la confidencialidad puede ocurrir de muchas maneras, como por ejemplo con la publicación intencional de información confidencial de la organización o el robo de la misma.
2. **Integridad:** El concepto de integridad busca asegurar tres aspectos claves:
   a) Que no se realicen modificaciones por personas no autorizadas a los datos, información o procesos.
   b) Que no se realicen modificaciones no autorizadas por personal autorizado a los datos, información o procesos.
   c) Que los datos o información sea consistente tanto interna como externamente.
3. **Disponibilidad:** La disponibilidad busca el acceso confiable y oportuno a los datos, información o recursos. Otros requerimientos importantes para la seguridad de la información son:
4. **Autenticidad:** Es la necesidad de asegurar que los datos, información y procesos son genuinos. Por ejemplo garantizar que los mismos no han sido forjados o fabricados por personas mal intencionadas.
5. **No repudiación:** Busca garantizar que en una transacción, ninguna de las partes puede negar haber recibido la transacción y tampoco su contraparte negar haber enviado la transacción.

Las actividades de comercio electrónico usan técnicas de certificados o firmas digitales, y encriptación para establecer los requerimientos de autenticidad y no repudiación necesarios para este tipo de operaciones.

2.2.3. Principios de seguridad de aplicaciones

Los principios de seguridad de aplicaciones son colecciones de propiedades deseables, comportamientos, diseños e implementaciones prácticas que intentan reducir la posibilidad de ocurrencia de una amenaza y su impacto. Los principios de seguridad son independientes del lenguaje o arquitectura de software, así como metodologías de construcción para diseño y construcción de aplicaciones.

Los principios son importantes porque ayudan a crear y tomar decisiones de seguridad ante nuevas situaciones con las mismas ideas básicas. Mediante la consideración de cada uno de estos principios, se pueden derivar requerimientos de seguridad, tomar decisión de arquitectura e implementación e identificar posibles debilidades en sistemas.

Lo más importante que hay que recordar es que para que sean útiles, los principios deben ser evaluados, interpretados y aplicados a un problema específico.

Algunos principios de seguridad son:

Aplicar defensa en profundidad - defense in depth

La seguridad en la información debe estar presente en toda la vida útil de la misma, desde su creación hasta su disposición final. Durante su tiempo de vida esta información puede pasar a través de múltiples sistemas de procesamiento de datos. En consecuencia, pueden existir un sinfín de amenazas diferentes que pueden ser vulneradas. Para lograr la seguridad de la información en todo su ciclo de vida cada uno de los componentes de los sistemas de procesamiento de datos que ella recorre debe tener sus propios mecanismos de protección. Aplicando la defensa en profundidad, podemos asegurar que si una medida defensiva fue violada, existen otras medidas resguardando nuestra información.

La defensa en profundidad puede aplicarse basándose en tres tipos de niveles críticos de control: controles administrativos, controles lógicos y controles físicos, los cuales van a permitir elaborar una eficiente estrategia. Con este planteamiento la defensa en profundidad puede verse conceptualizada en 3 capas solapadas, es decir, una encima de la otra.

Usar un modelo restrictivo (cerrado) de seguridad

Se basa en la decisión de proveer acceso en vez de negarlo. Este principio determina que por defecto se debe negar el acceso y sólo proveerlo en los casos que el sistema de protección determine que sean necesarios.

Un diseño conservador debe basarse en argumentos que determinan qué objetos deben ser accesibles, en lugar de buscar razones para no acceder a los mismos.
Por ejemplo a la hora de configurar un firewall, sus políticas deberían ser: negar todo y sólo permitir lo que se necesita dejar pasar, en lugar de, permitir todo y negar algunas cosas consideradas nocivas.

Establecer menos privilegios

En la seguridad de la información, este principio exige que en cada módulo, un proceso, un usuario o un programa sea capaz de acceder sólo a la información y los recursos que sean necesarios para cumplir su funcionalidad respectiva.

Este principio es una herramienta útil en la seguridad, pero no garantiza totalmente la seguridad en un sistema.

Cuando se aplica a los usuarios y a las aplicaciones que se ejecutan, este concepto implica que todos los usuarios y aplicaciones deben contar con el menor número posible de privilegios todo el tiempo.

Evitar la seguridad por oscuridad

La seguridad a través de la oscuridad representa un débil control de la seguridad. Sin embargo, esto no quiere decir que la idea sea completamente errada.

Por ejemplo, la seguridad de una aplicación no debe basarse en la idea de que el código fuente es secreto. La seguridad debe depender de otros factores, entre ellos implementar políticas de creación de contraseñas que cuenten con seguridad, la implementación de defensa en profundidad, establecer límites de transacción comercial, asegurar una sólida arquitectura de la red, tener la capacidad de detectar fraudes, realizar auditorías y generar controles.

Un ejemplo práctico de este principio es Linux. Linux es de código fuente abierto, es decir, esta a disposición de todos, sin embargo, Linux es un seguro, resistente y robusto sistema operativo.

Capacidad de detectar comportamientos irregulares

La detección de comportamientos irregulares es importante ya que un ataque perfecto cuenta con tiempo ilimitado. Si se implementa una eficiente detección de comportamientos irregulares, entonces un atacante sólo obtiene pocos intentos de ataque antes de que se detecte y se le impida lanzar más ataques. Hay que recordar que si se recibe una petición que un usuario legítimo no ha generado, seguramente se trata de un ataque por lo que se debe responder de forma adecuada.

La detección de comportamientos irregulares no depende de otras tecnologías. El código fuente de la aplicación es el único componente del sistema que tiene suficiente información para detectar ataques.

El inicio y verificación constante de las sesiones son factores importantes en la detección de comportamientos irregulares. Se debe registrar todos los datos pertinentes que permiten manejar las sesiones de los usuarios, y las acciones que estos realizan durante su estadía en el sistema. Así podemos detectar problemas que no fueron detectados en tiempo de ejecución, simplemente con la revisión de los registros (logs). En particular, el uso de todos los mecanismos de seguridad debe quedar completamente registrado, con suficiente información para ayudar a localizar al delin-

cuente en caso de sufrir un ataque.

Hay muchas otras medidas de seguridad que se pueden ejecutar al detectar comportamientos irregulares. Una de las mejores es desactivar la cuenta del usuario atacante. Esto hará que la aplicación sea más difícil de atacar.

No confiar en servicios internos y/o externos

Los servicios muchas veces implican el uso de un sistema, bien sea, interno o externo.

Muchas organizaciones utilizan la capacidad de procesamiento de terceros, que tienen diferentes políticas y implementaciones de seguridad. Es poco probable poder controlar cualquier influencia externa o de terceros.

Por lo tanto, Todos los sistemas externos deben ser tratados de manera similar. No se puede dar una confianza implícita a los servicios de terceros ya que no podemos asegurar la seguridad en los mismos.

2.3. Roles y Capas de una aplicación web

Para la implementación y puesta en marcha de una aplicación web, se requiere de un capital humano multidisciplinario. Por lo general los desarrolladores de software no son especialistas en redes, y tampoco los especialistas en redes y plataforma son programadores. Cada persona o grupo de trabajo tiene responsabilidades específicas, con respecto a la aplicación web, en su área.

Por ejemplo el personal de redes se encarga de tareas como instalar los enlaces de comunicación, asignarles direcciones IP a cada máquina en la red y configurar los firewalls. Las personas de plataforma se encargan de instalar y configurar el sistema operativo en los equipos, así como instalar las aplicaciones y servicios que se ejecutan en los mismos. Servicios como los servidores web, servidores de correo y servidores de bases de datos. Y por otra parte los desarrolladores se encargan de la programación e implementación de las funcionalidades o procesos en las aplicaciones. Se pueden identificar diferentes roles relacionados con la implementación de una aplicación web. Estos roles definen áreas de trabajo así como conocimientos deseables que debe tener una persona para desempeñarse en los mismos. Los mismos se describen a continuación:

Técnico en hardware: Relacionado con el área física de los equipos y sus componentes. Equipos como servidores, routers y switches. Debe tener conocimientos de las características y capacidades de los mismos, memoria RAM, capacidad de procesamiento, tipos de interfaces, cantidad de interfaces, entre otros. Alimentación de los equipos y su ubicación física dentro de las instalaciones.

Especialista en Redes: Relacionado con la interconexión de equipos en las redes. Definen los enlaces para la comunicación y las características de los mismos. Capacidad, velocidad de conexión, políticas de acceso, protocolos de red utilizados, etc. Requieren tener conocimientos del hardware

y software relacionado con redes. Configurar componentes como switches, routers y firewalls. Y tener amplios conocimientos en protocolos de redes y servicios, como servidores de DNS, Proxy, DHCP, entre otros. Principalmente sobre redes IP.

Especialista en Aplicaciones y Servicios: Relacionado con la instalación y configuración de software en los equipos. Desde el sistema operativo hasta los servicios que corren en el mismo. Instalación y configuración de servicios como servidores web, servidores de correo, servidores de base de datos, etc. Dominio de las capacidades y operaciones del sistema operativo. Conocer el sistema de archivos y sistema de permisología de acceso a datos y ejecución. Debe poseer conocimientos de los protocolos que usan las aplicaciones, así como aspectos de configuración de cada servicio. La memoria que necesitan, los puertos que utilizan, cantidad de conexiones concurrentes, etc. Para este rol se puede encontrar muchísimas ramas especializadas, entre las que se destacan, el personal especializado en instalación, configuración y mantenimiento de sistemas operativos (Sun Microsystems) y los especialistas para sistemas manejadores de bases de datos (Oracle).

Desarrollador de software: Relacionados con el código fuente de la aplicación y sus ejecutables. Conocimientos en lenguajes de programación, patrones de diseño y metodologías de desarrollo de software. Implementan los procesos y funcionalidades de una aplicación web. Manejan conceptos de persistencia, lógica del negocio y presentación de datos. Conocen los protocolos utilizados para la comunicación con servicios. Son los responsables de implementar la lógica de negocios de la empresa en la aplicación web.

Estos roles no son necesariamente excluyentes entre sí, y es común que una persona puede actuar o desempeñarse, al mismo tiempo en cualquier combinación de los mismos. Muchas veces los programadores son los encargados de configurar los servidores web donde se ejecutan sus aplicaciones, cuando lo mejor sería, que una persona especializa sea la responsable de esta tarea, para así garantizar los niveles de calidad y seguridad necesarios en estas aplicaciones.

En base a los roles definidos anteriormente, se propone un esquema de capas, resultado de la investigación realizada, para el análisis de las vulnerabilidades y amenazas en las aplicaciones web, con sus respectivas medidas de seguridad o controles necesarios para protegerlas. Este esquema tiene como finalidad separar cada tema en este TEG por capas, para ofrecer una mayor comprensión y entendimiento de los mismos, de acuerdo a cada área y roles relacionados.

Estas capas son:

Figura 2.6: Modelo de capas desde el punto de vista de la seguridad (Elaboración propia)

La capa de física: Comprende el hardware de los sistemas. Específicamente, las instalaciones físicas, el cableado, los equipos, servidores, switches y routers, etc. Estos son el corazón de la red. Permiten la comunicación y es donde se ejecutan las aplicaciones.

La capa de red: Comprende la red. Las aplicaciones web son accedidas a través de una red, bien sea Internet o una intranet. Se usan un conjunto de protocolos y reglas de acceso a las redes para el transporte de datos, y es en esta capa donde se realizan dichas funciones.

La capa del Sistema Operativo y Servicios: El sistema operativo es el programa que administra los recursos del equipo. El sistema operativo se encarga de tareas como asignación de memoria, administración de procesos, entrada y salida de dispositivos, conexiones de red y manejo de archivos. Los sistemas operativos también implementan ciertas funciones de seguridad. Seguridad en cuanto al acceso de recursos locales en disco, permisos de ejecución de programas, información personal de los usuarios y los servicios ofrecidos por programas corriendo en el sistema. Específicamente contiene los servicios en los que se ejecutan las aplicaciones web.

La capa de aplicación: Comprende las aplicaciones web en sí. Su desarrollo, código fuente, integración con servicios, comunicación entre usuarios y la aplicación, manejo de la información, funcionalidades y características de la aplicación web entre otros.

Capítulo 3

Amenazas y Vulnerabilidades

En este capítulo se describen las vulnerabilidades y amenazas de seguridad más comunes, presentes en aplicaciones web. ¿Cuándo? y ¿Cómo? se presentan. En cada caso se hace referencia a las medidas de seguridad que deben tomarse para evitarlas o prevenirlas.

Una vulnerabilidad es un agujero o debilidad en la aplicación, que puede ser una deficiencia de diseño o desperfecto de implementación, la cual permite a un atacante causar daño a cualquier parte de una aplicación web. Si en una aplicación web se conoce que existe determinadas vulnerabilidades y existen ataques para explotar o aprovechar dichas vulnerabilidades, entonces hemos identificado una amenaza, ya que se ha determinado la posibilidad de que ocurra algo perjudicial para la aplicación.

3.1. Capa Física

3.1.1. Catástrofes naturales

Nos referimos a todos aquellos acontecimientos catastróficos, con distintos tipos y grados de consecuencias dañinas, cuya fuente de producción se debe a un hecho natural, es decir a un fenómeno de la naturaleza y por tanto no han sido causados, directamente, por el hombre. Este tipo de fenómenos pueden causar enormes daños a la infraestructura y equipos de alguna empresa, y suelen ocurrir en cualquier momento inesperado.

Medidas de Seguridad:

Establecer políticas de redundancia de la información sensible de la empresa, proteger las copias de seguridad, detección y corrección de fallas a nivel físico.

3.1.2. Deficiente control de acceso a la sala de servidores

La sala de servidores es el corazón de toda la red física, y alguien con acceso físico a los servidores, conmutadores, enrutadores, cables y otros dispositivos puede dañarlos gravemente e incluso robarlos con gran facilidad. Por lo que, se debe crear una política de seguridad que bloquee el acceso a la sala, sólo a las personas debidamente autorizadas, por ejemplo colocando un sistema de seguridad que necesita de una llave de acceso y con guardias custodiando el lugar.

Medidas de Seguridad:

Bloquear el acceso a la sala de servidores, establecer vigilancia.

3.1.3. Deficiente control de acceso a las computadoras de trabajo pertenecientes a la red de la compañía

Los atacantes pueden utilizar cualquier computadora conectada a la red para accesar o eliminar información que es importante para la empresa. Los lugares de trabajo desocupados o los lugares con fácil acceso a cualquier persona, por ejemplo la computadora de la recepción, son particularmente vulnerables a este tipo de amenazas.

Medidas de Seguridad:

Establecer vigilancia, asegurar que los dispositivos más vulnerables se encuentran en un lugar con díficil acceso, controlar el acceso a computadoras ociosas.

3.1.4. Deficiente protección de las copias de seguridad

Las copias de seguridad son elementos importantes en la recuperación en caso de desastres, pero no hay que olvidar que estas copias pueden ser robadas y utilizadas por personas mal intencionadas. Muchos administradores suelen dejar las copias de seguridad del servidor en la sala de servidores sin ningún tipo de protección.

Medidas de Seguridad:

Establecer vigilancia, asegurar que los dispositivos más vulnerables se encuentran en un lugar con díficil acceso, proteger las copias de seguridad.

3.1.5. Permitir utilizar cualquier tipo de dispositivo de almacenamiento en equipos con información importante

Si la información almacenada en un equipo es de alta importancia no se debería permitir guardar información en dispositivos como Floopy, o Pen Drives, sino en los dispositivos especializados para hacer respaldos, por ejemplo, los discos magnéticos.

Medidas de Seguridad:

Establecer vigilancia, desactivar unidades que permitan almacenamiento en equipos con información crítica.

3.2. Capa de Red

3.2.1. Deficiente protección de los datos en el tránsito

Esta vulnerabilidad ocurre cuando se transportan datos sensibles en un medio público, sin ningún tipo de protección. Entiéndase por protección por ejemplo el hacer uso de mecanismos de

cifrado de la información. Un atacante puede con un sniffer[1] capturar las tramas que viajan en la red y con esto hacerse de información confidencial que le permitan el acceso a una cuenta de usuario o información personal.

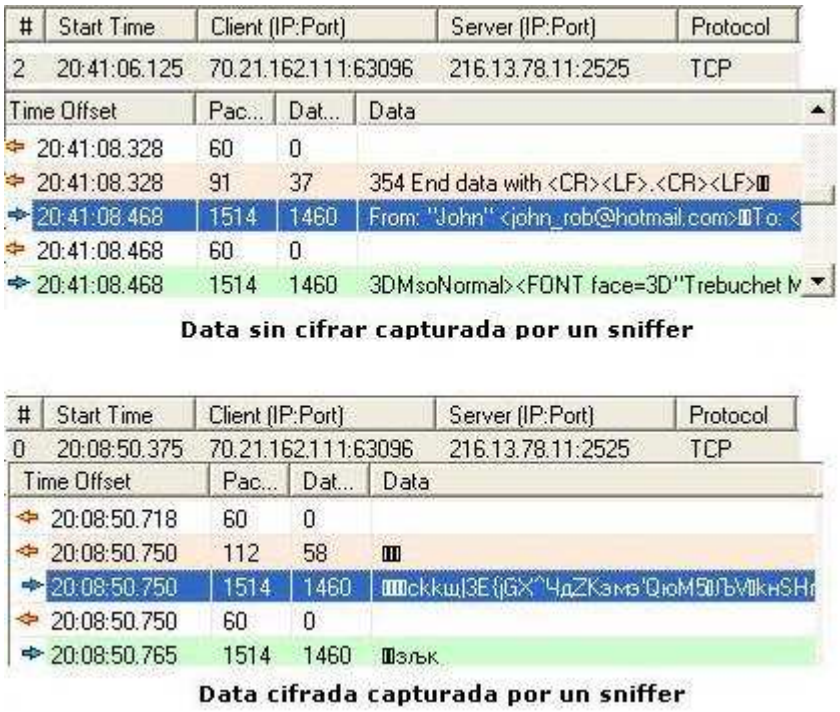


Figura 3.1: Información que muestra los datos capturados por un sniffer (Elaboración propia)

Medidas de Seguridad:

Uso de cortafuegos (firewall), uso de VPN.

3.2.2. Acceso a recursos privados

Esta vulnerabilidad ocurre cuando se puede tener acceso a recursos de la red, tales como un servidor de base de datos, acceso a servidores SSH, FTP, que no deberían ser públicos, incluso cuando algunas aplicaciones públicas tengan que accesar a dichos recursos, estos pudieran ser alcanzables a través de una red privada, o mediante una DMZ y no necesariamente a través de una IP publica, ni mucho menos exponiendo el puerto o punto de acceso al servicio de la misma.

Medidas de Seguridad:

Uso de cortafuegos (firewall), uso de VPN, uso de DMZ, uso de IDS, utilizar para recursos importantes una red privada con direcciones internas.

[1]Es un software o hardware que tiene la capacidad de interceptar y registrar el tráfico que pasa a través de una red completa o parte de ella. Un sniffer captura cada paquete, lo decodifica y analiza el contenido acorde al RFC especificado o a otra especificación.

3.2.3. Suplantación de IP (IP Spoofing)

La idea de este ataque es simular la identidad de una máquina de la red para conseguir accesos a recursos de un tercer sistema que confía plenamente en la IP suplantada. Como vemos hay tres actores importantes que entran en juego en este escenario: el atacante, el atacado y el suplantado, los objetivos del atacante son establecer comunicación con el sistema y evitar que el suplantado interfiera de alguna forma en el ataque. Es muy común ver este tipo de ataques en configuraciones de algunos manejadores de BD (postgreSQL, Oracle) que solamente dan acceso a las BD a una o varias direcciones IP que se consideran confiables.

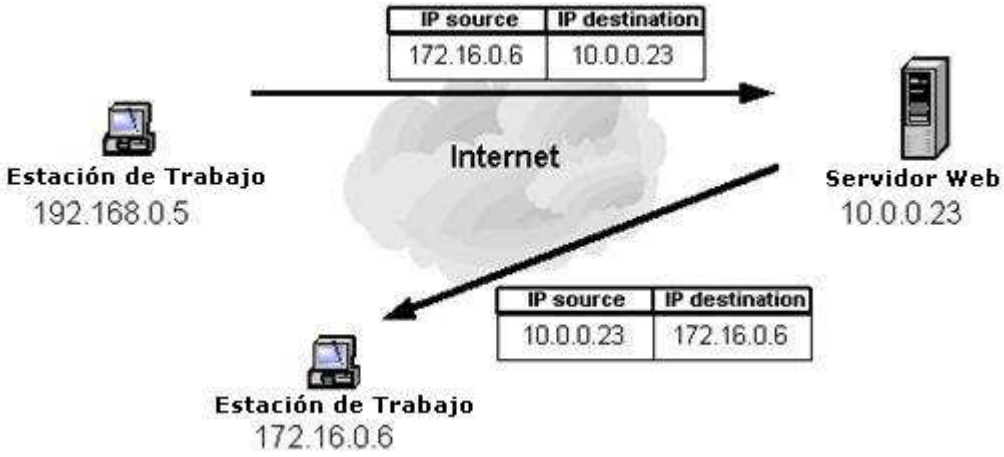


Figura 3.2: Suplantación de la IP origen (Elaboración propia)

Medidas de Seguridad:

Uso de cortafuegos (firewall), uso de VPN, uso de DMZ, uso de IDS.

3.2.4. Suplantación de DNS (DNS Spoofing)

Se trata de falsear la relación que existe entre la IP y la dirección de dominio cuando se solicita una resolución de nombre. Se logra vulnerando el servidor DNS para generar entradas que sean falsas asi se puede redirigir cualquier petición hacia otro sitio. Estas entradas falseadas del servidor DNS pueden envenenar el caché DNS de otro servidor diferente lo que es ampliamente conocido como DNS poisoning[2].

Medidas de Seguridad:

Uso de cortafuegos (firewall), uso de VPN, uso de DMZ, uso de IDS.

[2]DNS poisoning:Se trata del falseamiento de una relación "Nombre de dominio-IP.ªnte una consulta de resolución de nombre, es decir, resolver con una dirección IP falsa un cierto nombre DNS o viceversa. Esto se consigue falseando las entradas de la relación Nombre de dominio-IP de un servidor DNS, mediante alguna vulnerabilidad del servidor en concreto o por su confianza hacia servidores poco fiables. Las entradas falseadas de un servidor DNS son susceptibles de infectar (envenenar) el caché DNS de otro servidor diferente.

3.2.5. Suplantación de una página web real

Enruta la conexión de una víctima a través de una página web falsa hacia otras paginas web con el objetivo de obtener información de la víctima. La página web falseada funciona como un Proxy, es decir solicita información a cada servidor original que desee ser visitado por la víctima, generándole la respuesta correspondiente en cada petición.

Medidas de Seguridad:

Uso de cortafuegos (firewall), uso de VPN, uso de DMZ, uso de IDS.

3.2.6. Denegación de servicio (DOS)

Este tipo de ataques desactivan o saturan los recursos del sistema. Por ejemplo, un atacante puede consumir toda la memoria o espacio en disco disponible, así como enviar tanto tráfico a la red que nadie más pueda utilizarla.

Aquí el atacante satura el sistema con mensajes que requieren establecer conexión. Sin embargo, en vez de proveer la dirección IP del emisor, el mensaje contiene falsas direcciones IP usando Suplantación de IP. El sistema responde al mensaje, pero como no recibe respuesta, acumula buffers con información de las conexiones abiertas, no dejando lugar a las conexiones legítimas.

Medidas de Seguridad:

Uso de cortafuegos (firewall), uso de VPN, uso de DMZ, uso de IDS, bloqueos de IP y limpiado de buffer.

3.3. Capa de Sistema Operativo y Servicios

3.3.1. Deficiente protección de directorios y archivos

En general los sistemas operativos permiten funcionalidades para definir permisos específicos sobre cada directorio o archivo en el sistema. A través de estos mecanismos se establece que usuarios o grupos de usuarios tienen permisos, bien sea de lectura, escritura y/o ejecución, sobre los diferentes archivos del sistema.

Con respecto al sistema operativo y los servicios que corren en él, en nuestro caso los más relevantes, los servidores web, es importante tener bien definido y de forma estricta, la permisología de archivos del sistema y otros archivos sensibles. Si estos no tienen una definición de permisos adecuada, un servidor web ejecutándose con un usuario cualquiera puede tener acceso a los mismos, y bien sea ejecutarlos o servir estos archivos a un atacante.

Por ejemplo en los sistemas Unix existen archivos del sistema que almacenan cuentas de usuarios y sus contraseñas. Tal es el caso de archivos como “/etc/passwd”. El acceso a este archivo debe ser exclusivo para usuarios privilegiados (root o cuentas de administrador), en caso contrario el servidor web puede tener acceso al mismo al ser víctima de un ataque de path trasversal o

ejecución de comandos.

Otro ejemplo de comandos que pueden generar este tipo de vulnerabilidad:

1. chmod 777 –Rf /
2. chown nobody:nobody –Rf /etc /var
3. chmod 777 –Rf /logs

Medidas de Seguridad:

Permisos de ejecución y cuentas de usuario limitadas, eliminación de servicios y configuraciones innecesarias.

3.3.2. Servicios ejecutándose con cuentas privilegiadas

Esta vulnerabilidad también se presenta si el servidor web esta ejecutándose con una cuenta de usuario privilegiado, ya que esto puede permitir el acceso del servidor web a todo el sistema de archivos de la máquina y por tanto abrir la posibilidad a un ataque.
No solamente el servidor web sino también sus módulos, como por ejemplo el CGI, o el PHP.

Medidas de Seguridad:

Permisos de ejecución y cuentas de usuario limitadas.

3.3.3. Generar cuentas de administradores sin contraseña o con contraseña débiles

Una de las vulnerabilidades más explotadas a nivel de SO ocurre gracias a la creación de cuentas con privilegios valiosos, tipo administrador, que muchas veces por descuido y olvido no poseen contraseñas y si las tienen son muy inseguras, esto ocasiona, por ejemplo, que una contraseña insegura con un ataque de fuerza bruta pueda ser fácilmente descubierta. Si al generar a los usuarios se utiliza una política fuerte en la creación de las contraseñas como por ejemplo, caducar las contraseñas, no poder utilizar contraseñas anteriores, combinar letras, números y/o caracteres especiales, se va a producir un nivel extra de seguridad a la posibilidad de explotar esta vulnerabilidad.

Medidas de Seguridad:

Implementar políticas para crear contraseñas seguras.

3.3.4. Servicios en texto plano

Como se comento en la capa de red mediante un sniffer se puede capturar la información que es transportada como texto plano y fácilmente puede ser leída provocando una vulnerabilidad. Ahora bien, si a nivel de servicios de SO esta información que va a ser transportada por un medio que en

la mayoría de las veces no es seguro, a su vez es manejada ya sea como entrada o salida al servicio como un texto plano, se va a facilitar que esta vulnerabilidad sea igualmente explotada a nivel de SO. Para reducir el riesgo se deben considerar reemplazar servicios de texto plano por alternativas con cifrado de datos, por ejemplo, utilizar HTTPS por HTTP.

Otro ejemplo, viene a ser, los servidores de correo, los cuales se encargan de enviar mensajes de un usuario a otro independientemente de la red que cada uno este utilizando. SendMail es uno de los servidores más populares y por ende uno de los que tiene más alertas de seguridad. Para esta vulnerabilidad se tiende a agregar la capa SSL en el proceso de intercambio de información de los usuarios.

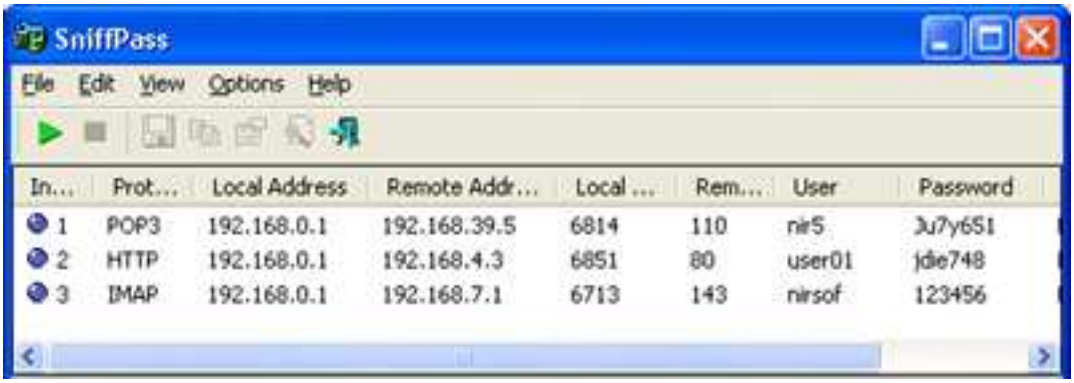


Figura 3.3: Servicios que contienen información sensitiva en texto plano (Elaboración propia)

Medidas de Seguridad:

Implementar técnicas de encriptación para los datos manejados en servicios críticos.

3.3.5. Utilizar versiones viejas e inseguras de los servicios

La mayoría de las veces las aplicaciones son explotadas y rápidamente sus desarrolladores crean parches, los documentan y liberan las versiones correspondientes que contrarrestan estas vulnerabilidades, la prueba más real de este tipo de comportamiento son los constantes service pack y actualizaciones que realiza la empresa Microsoft a sus sistemas operativos Windows, sin embargo, hay muchos sistemas que no se actualizan progresivamente y pueden ser amenazados no sólo por atacantes expertos, sino que también pueden atacantes inexpertos con sólo bajar documentación de la vulnerabilidad explotarla fácilmente. Lo recomendable sería crear políticas de actualización automáticas o constantes en períodos cortos de tiempo.

Medidas de Seguridad:

Actualizaciones periódicas.

3.3.6. Denegación de servicios del SO

Este tipo de vulnerabilidad es una de las que se presenta con más frecuencia a nivel de red y de SO, el objetivo principal de estas amenazas es consumir todos los recursos administrados por

el SO, provocando la suspensión temporal de uno o todos los servicios del mismo.

Entre los métodos de ataques de denegación de servicios de SO más conocidos tenemos: perturbar servicios a un sistema o persona en específico y saturar servicios con peticiones que son imposibles de manejar.

Por ejemplo, si múltiples peticiones son enviadas simultáneamente en un período muy corto de tiempo a nuestro servidor web se van a consumir rápidamente todos los recursos del SO provocando que este colapse y en muchos casos sea necesario reiniciarlo para poder reestablecer su correcto funcionamiento, esto provoca que los usuarios legítimos no puedan tener acceso a una aplicación web en un espacio de tiempo determinado.

3.3.7. Instalación y Ejecución de servicios que no son útiles

Aunque pudiera parecer muy trivial esta amenaza, es muy frecuente conseguir en los SO servicios en estado de ejecución que simplemente no se necesitan en el entorno de nuestra aplicación Web. Muchos de estos servicios pueden venir instalados por defecto y en SO como Windows pueden ejecutarse al encender la máquina.
Por ejemplo, si tenemos un servicio de FTP ejecutándose en una máquina y simplemente no lo utilizamos algún atacante puede hacer uso de este servicio para cargar y descargar archivos de manera no autorizada.
Este tipo de servicios deberían ser deshabilitados o desinstalados para mejorar la seguridad.

Medidas de Seguridad:

Eliminación de servicios y configuraciones innecesarias.

3.3.8. Configuraciones por defecto

Es común que por cuestiones de tiempo en múltiples máquinas de nuestra red sean instalados y configurados los SO con las opciones y servicios que vienen por defecto, muchas de estas configuraciones son ampliamente conocidas y de no ser revisadas y auditadas pueden presentar un riesgo de vulnerabilidad.

Las mismas pueden dar espacio a por ejemplo cuentas de administradores por defecto con contraseña en blanco o ampliamente conocida (caso de tomcat en versiones iniciales).

La existencia de paginas de inicio o secciones por defecto pueden alertar a un atacante sobre una potencial vulnerabilidad. Por ejemplo la pagina de inicio de apache muy conocida “It Works!”. O paginas de error por defecto como la clásica pagina no encontrada “404”.

Las configuraciones por defecto pueden llevar a otras configuraciones incorrectas o peligrosas como Listado de directorio, uso de librerías que no se necesitan y métodos HTTP en algunos casos peligrosos como PUT, DELETE.

Medidas de Seguridad:

Eliminación de servicios y configuraciones innecesarias, chequeo de configuración desde cero, verificación de credenciales y permisos..

3.3.9. Capturadores de teclas – (Key Loggers)

Es una herramienta de diagnostico utilizada en el desarrollo de software que se encarga de registrar las pulsaciones que se realizan sobre el teclado, para memorizarlas en un fichero o enviarlas a través de Internet. Este programa es instalado en la máquina de la víctima y el atacante posteriormente revisa los registros de pulsaciones del teclado y busca las secuencias de las teclas para dar con contraseñas, números de tarjetas de créditos, etc.
Son programas difíciles de detectar por el usuario ya que dejan sus registros en directorios ocultos.

Medidas de Seguridad:

Uso de Antivirus.

3.3.10. Uso de versiones "viejas" y/o inseguras

Se pueden encontrar distintas implementaciones de aplicaciones web. De diferentes casas y para usos particulares. Por ejemplo el servidor apache es ampliamente aceptado dentro de la comunidad de desarrolladores y empresas de servicios en Internet. Este puede soportar e integrarse con diferentes lenguajes y módulos que incrementan sus capacidades.

Los servidores web se pueden encontrar en diferentes versiones. A lo largo del tiempo estas aplicaciones se encuentran en un proceso continuo de mejoramiento e incremento de sus funcionalidades.

Usar versiones anteriores a la última versión estable puede significar una vulnerabilidad de seguridad. Ya que generalmente las últimas versiones llevan consigo soluciones y parches para defectos y vulnerabilidades conocidas en versiones anteriores.

Algunas versiones pueden ser consideradas inseguras bien sea porque existen vulnerabilidades ampliamente conocidas o porque son versiones beta o lo que se conoce como inestables. Encontrar servidores web públicos, en producción, con estas características representa una vulnerabilidad para la aplicación web y en algunos casos para la máquina servidor en sí. Si se esta trabajando con versiones "viejas"y que están suficientemente estables, o simplemente su actualización no es una tarea sencilla, se recomienda por lo menos estar al dia con los respectivos parches para las vulnerabilidades conocidas.

Medidas de Seguridad:

Actualización de versiones, aplicación de parches.

3.3.11. Ataque huella digital HTTP (HTTP fingerprint)

Se trata de un ataque que buscar información propia del servidor web y el sistema operativo. Consiste en enviar un conjunto de peticiones especializadas, y mediante el análisis de las respuestas del servidor web, determinar e identificar características tales como versión del sistema operativo y servidor web. Con esta información un atacante puede planificar mejor una estrategia de ataque contra el sistema y/o aplicación.

Medidas de Seguridad:

Revisar accesos por defecto o comunes, deshabilitar todo lo que no sea necesario.

3.3.12. Servidor web en modo privilegiado

Esta vulnerabilidad se presenta cuando el servidor web corre con permisos de un usuario administrador o usuario privilegiado (por ejemplo usuario root en sistemas unix). Este tipo de configuración puede permitirle acceso a archivos delicados del sistema. Por ejemplo, si el servidor es víctima de un ataque de ejecución de comandos, y el mismo tiene permisos de súper usuario, el un atacante pudiera causar daños graves al sistema, tales como borrar archivos, subir y ejecutar binarios, entre otras cosas.

Medidas de Seguridad:

Revisar accesos por defecto o comunes.

3.3.13. Funcionalidades que no se necesitan habilitadas

Tener funcionalidades o características que realmente no se necesitan para el normal y correcto funcionamiento del servidor web y sus aplicaciones, se puede degenerar en vulnerabilidades para el sistema. Estas funcionalidades pueden ofrecer puntos de entrada y/o características adicionales para un atacante. Lo que hoy parece inofensivo, mañana puede ser un problema.

Medidas de Seguridad:

Deshabilitar todo lo que no sea necesario.

3.3.14. Exponer archivos sensibles en el directorio web raíz

En las aplicaciones web existen distintos tipos de archivos. Cada uno de ellos con diferentes fines y propósitos. Algunos de carácter público como imágenes, las páginas de vistas de usuario, documentos para descargar, pdf, zip, doc, hojas de estilo css, entre otros. Otros de carácter privado, como archivos de configuración, logs, propiedades, incluso páginas o scripts que no están destinadas a ser accedidas directamente por los usuarios, sino como parte de la lógica de negocios o flujo de ejecución procesos internos en la aplicación (script para los llamados controles o modelos).

Colocar todos estos archivos sin distinción alguna dentro del directorio web raíz (o webroot en ingles), es una vulnerabilidad que resulta muy peligrosa. Si archivos de carácter privado se encuentran dentro del directorio web raíz, entonces existe la posibilidad de que puedan ser servidos

a los usuarios. Un atacante pudiera conseguir información importante al obtener archivos de logs o archivos de configuración de la aplicación.

Se puede tener un falso sentido de seguridad, al no tener indexado de alguna forma estos recursos en la página web. Se denomina seguridad a través de oscuridad. El hecho de que estos recursos no estén evidenciados en el contenido y forma de la página web, no quiere decir que estos no sean alcanzables. Bien sea por heurística o adivinación un atacante puede llegar a ellos.

Esta vulnerabilidad ocurre con bastante frecuencia. Sobre todo en entornos limitados como hosting compartido.

Existen mecanismo provistos por algunos servidores web o servidores de aplicaciones que permiten definir directorios privados aún dentro del directorio web raíz, por ejemplo el uso de archivos .htaccess en servidores apache o el directorio WEB-INF en servidores tomcat.

Medidas de Seguridad:

Colocar en el directorio web raíz sólo archivos públicos, uso de directorios privados dentro del directorio web raíz.

3.3.15. Exponer archivos fuentes en el directorio web raíz

Esta vulnerabilidad es un caso específico de Exponer archivos sensibles en el directorio web raíz, sin embargo es muy común que ocurra en el proceso de puesta en producción y actualización de versiones de las aplicaciones web.

Muchas veces las aplicaciones son llevadas a su destino final, en un ambiente de producción, con sus respectivos códigos fuentes. A veces por descuido o simplemente para facilitar el manejo de versiones (la ultima versión esta en producción y siempre se pueda recuperar). También es común que, por descuido, en un proceso de actualización de la aplicación se cuelen los archivos fuentes por error. Un atacante puede tener acceso a estos códigos fuentes y buscar vulnerabilidades en la implementación con facilidad.

Existen casos de lenguajes interpretados, ejemplo: PHP, ASP, Ruby on Rails; que generalmente por ser lenguajes interpretados, sus scripts son tanto el código fuente como el ejecutable. Sin embargo aplicaciones en estos lenguajes pueden acceder a otros componentes tales como archivos para inclusión (archivos .inc por ejemplo) y módulos desarrollados en lenguajes compilados como lenguaje C con sus fuentes, en los cuales se pueden exponer código fuente de la aplicación.

Otros lenguajes son compilados, como el caso de Java. En este caso el código fuente es expuesto si es subido al ambiente de producción con los respectivos archivos .java y los mismos se encuentran dentro del directorio web raíz.

Medidas de Seguridad:

Colocar en el directorio web raíz sólo archivos públicos, uso de directorios privados dentro del directorio web raíz.

3.3.16. Exponer archivos de configuración

Esta vulnerabilidad es un caso específico de Exponer archivos sensibles en el directorio web raíz, sin embargo, es una vulnerabilidad presente con mucha frecuencia en las aplicaciones web. Se colocan los archivos de configuración en el directorio raíz del servidor web para facilitar su acceso desde los script de la aplicación y/o por omisión de complicaciones para establecer políticas de acceso estricta a estos archivos privados. O simplemente por limitaciones en el entorno de producción tales como las presentes en algunos entornos compartidos como web hosting.

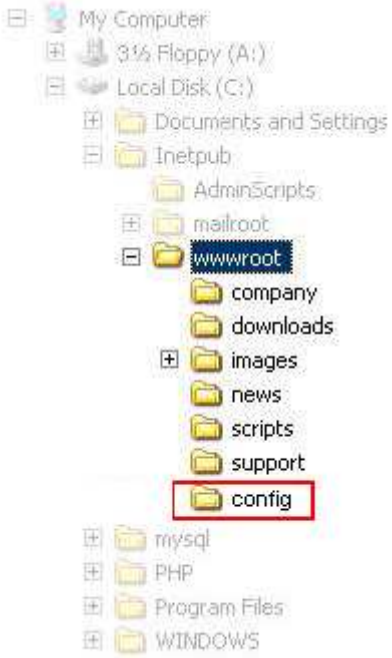


Figura 3.4: Exposición de archivos de configuración en el directorio raíz de un servidor web IIS (Elaboración propia)

Medidas de Seguridad:

Colocar en el directorio web raíz sólo archivos públicos, uso de directorios privados dentro del directorio web raíz.

3.3.17. Configuración incorrecta de permisos en directorios y archivos

Un sistema operativo multiusuario permite garantizar a los usuarios permisos distintos sobre un directorio o archivo. Cuando se crea un directorio o un archivo, se puede introducir una serie de permisos para ese directorio o archivo. Se debe garantizar que directorios y archivos claves en una aplicación no puedan ser modificados y/o eliminados por cualquier usuario.

Medidas de Seguridad:

Colocar en el directorio web raíz sólo archivos públicos, uso de directorios privados dentro del directorio web raíz, correcta configuración de permisos en los directorios y archivos.

3.3.18. Inyección de código SQL

Una inyección de SQL se produce cuando algún manejador utiliza datos mal formados proporcionados por algún usuario, que no han sido validados, para realizar alguna consulta.

Este tipo de errores puede permitir a usuarios malintencionados acceder a datos a los cuales desde otro modo no tendrán acceso. Consiguiendo en el peor de los casos lograr modificar el comportamiento de alguna aplicación.

La forma principal de inyección de código SQL consiste en la inserción directa de código en variables especificadas por el usuario que se concatenan con comandos SQL y se ejecutan. Existe un ataque menos directo que inyecta código dañino en cadenas que estén destinadas a almacenarse en una tabla o como metadatos. Cuando las cadenas almacenadas se concatenan posteriormente en un comando SQL dinámico, se ejecuta el código dañino.

Medidas de Seguridad:

Realizar copias de seguridad de los datos, filtrado de entradas.

3.3.19. Utilizar versiones inseguras de los manejadores de base de datos

La mayoría de las vulnerabilidades que comprometen directamente a los datos vienen dadas, por utilizar versiones de programas que presentan fuertes hoyos de seguridad, que son fuertemente explotados.

Este tipo de comportamiento es tan cotidiano que empresas como ORACLE realizan actualizaciones y parches de seguridad trimestralmente, donde se busca contrarrestar todos los problemas a nivel de seguridad que presentan las versiones anteriores.

Medidas de Seguridad:

Actualice periódicamente los manejadores de bases de datos, realizar copias de seguridad de los datos, proveer vistas del repositorio de los datos.

3.3.20. Vulnerabilidades en los driver conectores a base de datos

Existen muchos componentes que se encargan de proporcionar conectividad a bases de datos mediante múltiples lenguajes, por ejemplo, PHP, ASP o Python.

Este tipo de componentes son ampliamente usados por ser de capaces proveer un nivel de abstracción muy alto y rápido, para manejar la comunicación entre la aplicación y los datos.

La mayoría de los problemas encontrados en estos componentes facilitan enormemente la explotación de otras vulnerabilidades, por ejemplo: Inyección de código SQL, por consiguiente, la manipulación remota de los datos.

Medidas de Seguridad:

Actualice periódicamente los manejadores de bases de datos, realizar copias de seguridad de los datos, proveer vistas del repositorio de los datos.

3.3.21. Información sensible almacenada incorrectamente

Muchas veces la información es recogida y almacenada en archivos de texto plano. Este tipo de información la mayoría de las veces corresponde a contraseñas de usuarios que se deben autenticar en algún sistema. Al no cifrar el contenido de estos archivos cualquier usuario puede facilmente hacerse con los datos necesarios para autenticarse sobre algún sistema y lograr vulnerar el comportamiento del mismo.

Medidas de Seguridad:

Cifrar datos sensibles.

3.3.22. Exponer el servidor de base de datos

El servidor de base de datos es el repositorio principal de toda la información que posee nuestra empresa. Colocar este servidor en un equipo que puede ser accedido publicamente, representa uno de las mayores riesgos en el repositorio de nuestros datos.

Medidas de Seguridad:

Uso de cortafuegos, configurar DMZ.

3.3.23. Exponer una base de datos que sea crítica

Las bases de datos permiten organizar los datos en forma estructurada los cuales son accesibles en tiempo real por usuarios con diferentes necesidad de información, existen datos guardados en las bases de datos los cuales son cruciales para la empresa como por ejemplo, los datos de clientes de una empresa. Esta informacion es manejada directamente por los usuarios autenticados, lo que puede representar un peligro latente en el caso de que sea vulnerada la autenticidad de un usuario. La idea es preveer este tipo de escenarios proveyendo un mecanismo donde se manipule indirectamente la base de datos, por ejemplo, crear una base de datos secundaria y permitir que los usuarios manipulen esta base de datos y si todas las operaciones se han realizado con éxito durante un periodo determinado de tiempo sincronizarla con la base de datos original. Con esto se evita que en caso de haber un error los datos sean corruptos en el origen.

Medidas de Seguridad:

Proveer vistas del repositorio de los datos, verificación de credenciales y permisos, chequeo de configuración desde cero.

3.3.24. Usuarios con control total del repositorio de datos

Mantener usuarios que sean dioses de un repositorio de datos, no es una técnica muy fiable, se deben crear usuarios que tengan los permisos adecuados según la información y acción específica que cada uno requiere.

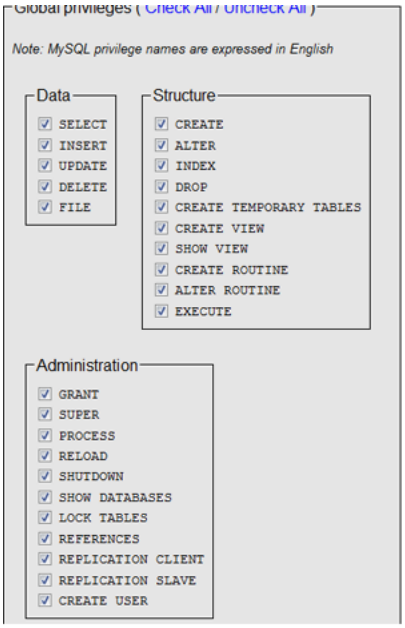


Figura 3.5: Creación de un usuario en mysql con todos los privilegios (Elaboración propia)

Medidas de Seguridad:

Verificación de credenciales y permisos, chequeo de configuración desde cero, forzar el uso de contraseñas seguras.

3.3.25. Uso de credenciales predecibles o por defecto

Los usuarios generalmente no tienen mucho cuidado a la hora de elegir una contraseña, y colocan contraseñas que cualquier persona puede predecir, Por descuido tambien es común encontrarnos en los accesos a bases de datos cuentas y contraseñas por defecto. Por ejemplo en SQL Server el usuario sa con la contraseña en blanco. Los atacantes deben haber accedido a millones de sistemas utilizando este tipo de vulnerabilidad.

Medidas de Seguridad:

Verificación de credenciales y permisos, chequeo de configuración desde cero.

3.3.26. Usar binarios corruptos

Se debe verificar la integridad de los binarios para instalar o actualizar las aplicaciones, herramientas y servicios. Los mismos pueden ser objeto de modificaciones. Casi siempre se provee conjuntamente con el binario descargable, su respectivo código de verificación Hash.

Medidas de Seguridad:

Verificación integridad de Herramientas y Servicios.

3.4. Capa de Aplicación

3.4.1. Ataques que afectan la autenticación

Son ataques cuyo objetivo es vulnerar la forma en la que un sitio web valida la identidad de un usuario, servicio o aplicación. Se da principalmente cuando el interesado posee su nombre de usuario conjunto a una contraseña que le permita el acceso al sitio web.

3.4.2. Ataques de fuerza bruta

Es la forma de recuperar una clave probando todas las combinaciones posibles de caracteres hasta encontrar aquella que permite el acceso. Si se llegan a probar todas las combinaciones posibles este ataque tendrá un seguro éxito, solo que por ser un método automatizado de prueba y error lleva costos de tiempo computacional inclusive hasta años, sobre todo cuando se trata de campos combinados como mayúsculas, minúsculas y dígitos.

Longitud Contrasena	**letras en minúsculas**	**Letras en minúsculas y dígitos**	**Letras en mayúsculas y minúsculas**	**Todos los caracteres ASCII**
$\leq$ **4**	instantáneo	instantáneo	instantáneo	2 min.
5	instantáneo	2 min.	12 min.	4 horas
6	10 min.	72 min.	10 horas	18 días
7	4 horas	43 horas	23 días	4 años
8	4 días	65 días	3 años	463 años
9	4 meses	6 años	178 años	44530 años

Cuadro 3.1: Tiempo requerido para un ataque de fuerza bruta. Computadora a una rata de 500.000 contrasenas por segundo

Medidas de seguridad:

Uso de captcha, control de ataques de fuerza bruta.

3.4.3. Autenticación insuficiente

Ocurre cuando un sitio web permite a un atacante acceder a contenido o funcionalidad sensible sin tener que autenticarse correctamente.

Medidas de seguridad:

Verificar la autenticación de un usuario en todo momento.

3.4.4. Débil validación en la recuperación de contraseñas

Es la que puede permitir a un atacante obtener, de forma ilegal, la contraseña de un usuario, ya que las formas de recuperar las contraseñas pérdidas u olvidadas es muy fácil de vulnerar. Para recuperar las contraseñas existe un método llamado hint o pista que reside en solicitar que el usuario suministre una *indicación* durante el proceso de registro para recordar su contraseña este ayuda a comprimir ampliamente el tamaño del diccionario de fuerza bruta.

Medidas de seguridad:

Uso de captcha, seguridad en profundidad para la recuperación en contraseñas.

3.4.5. Ataques que afectan la autorización

Son los ataques que tienen como objetivo vulnerar los métodos de los sitios web para determinar si un usuario, servicio o aplicación tiene los permisos necesarios para ejecutar una acción solicitada

3.4.6. Predicción de credenciales/sesión (Session hijacking)

El término es ampliamente conocido como "Session hijacking" y consiste en deducir o adivinar el valor único que identifica a un usuario o sesión en particular. Con el fin de secuestrar o suplantar la identidad de un usuario de un sitio web. Las consecuencias podrían permitir a los atacantes la capacidad de emitir peticiones al sitio web con los privilegios del usuario comprometido. Las aplicaciones web crearon un id único $identificador$. Mediante este id se establece un mecanismo para identificar los distintos usuarios que accedan a un sitio web. Si un atacante es capaz de predecir o adivinar la id de sesión de otro usuario, la actividad fraudulenta es posible, sin pasar por el proceso de autenticación.

Medidas de seguridad:

Establecer controles de sesión en procesos críticos.

3.4.7. Expiración de sesión insuficiente

Se produce cuando un sitio web permite a un atacante reutilizar unas credenciales de sesión antiguas o ids de sesión caducadas para la autorización. La expiración de sesión insuficiente puede permitir al agresor usar el botón atrás del navegador para permitir el acceso a las paginas vistas previamente por la víctima y su sesión aún puede estar en uso.

Medidas de seguridad:

Establecer controles de sesión en procesos críticos.

3.4.8. Fijación de sesión (Session fixation)

El término es ampliamente conocido como *Session fixation* y es una técnica de ataque que fuerza a un usuario a adoptar un id de sesión determinado. Este id de sesión lo conoce el atacante y es incorporado en la víctima. Luego, una vez que la víctima se autentica en el sitio con sus credenciales, el atacante, usurpa la identidad del usuario, usando el id de sesión previamente fijado. Si el usuario no tiene una protección activa contra la fijación de sesión, el ataque puede ser ajustado contra cualquier sitio web utilizando sesiones para igualar a los usuarios autenticados.

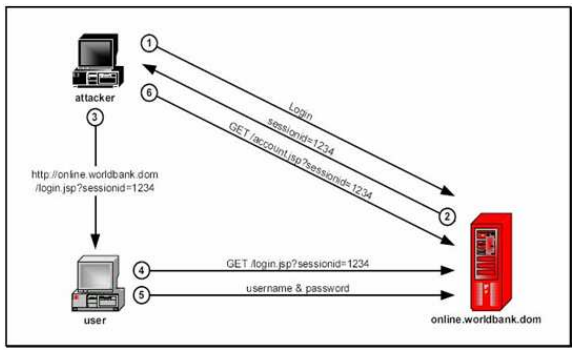


Figura 3.6: Ejemplo de fijación de sesión (session fixation) (Elaboración propia)

Medidas de seguridad:

Establecer controles de sesión en procesos críticos.

3.4.9. Ataques en la parte cliente

Estos ataques consiguen engañar al cliente haciéndole creer que el contenido en el sitio web es legítimo, o simplemente ocultando a la vista del cliente lo que realmente esta ocurriendo en el fondo. Es significativo que los sitios web respondan en todo momento que su contenido es integro y autentico ya que el usuario establece una relación de confianza de carácter tecnológico y psicológico.

3.4.10. Suplantación de contenido

Es una técnica de ataque usada para engañar a un usuario haciéndole creer que ciertamente el contenido que aparece en el sitio web es legítimo y no proviene de ninguna fuente externa. Un atacante puede ser capaz de reemplazar el valor del parámetro *fuente_frame* con "fuente_frame = http://atacante.dom/ataque.html" y de esta manera engañar al usuario ya que este pensara que el contenido es autentico porque que la barra de navegador muestra la dirección del sitio de su confianza, pero la información en el frame no lo es.

Medidas de seguridad:

Filtrar la entrada de datos.

3.4.11. Ejecución de código forzado (Cross-site scripting)

Es una técnica de ataque en la cual un sitio web es obligado o forzado a suministrar código ejecutable a un cliente. Cuando un atacante consigue que el navegador de un usuario ejecute su código, el código se ejecutará dentro del contexto de seguridad (o zona) del sitio web. Con esta técnica un agresor puede hacerse con el id de sesión de un usuario (Sesión hijacking), o hacer que el navegador del usuario sea redirigido a otra dirección, o mostrar contenido fraudulento entregado por el sitio web que está visitando. Estos ataques se pueden clasificar en persistentes y no persistentes, los no persistentes solicitan que un usuario visite un enlace especialmente diseñado que incluye código malicioso y los persistentes ocurren cuando el código malicioso es insertado en un sitio web y este es almacenado por el mismo, y luego servido a los clientes como código legítimo.

Figura 3.7: Ejemplo de vulnerabilidad XSS en You Tube (Imagen tomada de http://cordobo.com/wp-content/youtube-xss-cordobo.png)

Medidas de seguridad:

Filtrar la entrada de datos, escapar la salida de datos.

3.4.12. Forjar peticiones de un sitio a otro (cross-site request forgery)

Esta técnica permite a un atacante enviar peticiones HTTP desde un usuario víctima. El escenario típico es cuando el usuario ha establecido un nivel de privilegios con el sitio objetivo del ataque y esto permite que el atacante realice acciones no autorizadas por el pero autorizadas al usuario, este tipo de ataques explota la confianza que tiene un sitio en el usuario.

Medidas de seguridad:

Filtrar la entrada de datos, escapar la salida de datos.

3.4.13. Ataques que permiten ejecución de comandos

Estos son los ataques diseñados para ejecutar comandos remotos en el sitio web. Los datos facilitados por el usuario se usan para crear comandos para la elaboración de contenido dinámico, si este proceso no es seguro, el agresor altera la ejecución de los comandos.

3.4.14. Desbordamiento de buffer

Este tipo de ataque altera el flujo de una aplicación sobrescribiendo partes de la memoria. Cuando el buffer es desbordado, direcciones de memoria adyacentes son sobrescritas causando que el software falle o termine de manera inesperada. Un desbordamiento de buffer logra ser usado como un ataque de denegación de servicio cuando se obtiene corrumpir la memoria, debido a que el software puede presentar fallas y dejar de funcionar. Las vulnerabilidades de desbordamiento de buffer ocurren principalmente en lenguajes de programación como C y C++.

Medidas de seguridad:

Filtrar la entrada de datos.

3.4.15. Ataques de formato de cadena

Estos ataques alteran el flujo de una aplicación usando funcionalidades de librerías de formatos de cadena para acceder a otros espacios de memoria. Las vulnerabilidades se dan principalmente cuando los datos facilitados por el usuario son usados verdaderamente como parámetros de formato de cadenas de ciertas funciones de C/C++. Sus consecuencias podrían ser: ejecutar código arbitrario en el servidor, leer datos de la pila, leer cadenas de caracteres de la memoria del proceso, escribir un entero en posiciones de la memoria del proceso o causar fallos de segmentación.

3.4.16. Inyección LDAP

Esta es una técnica de ataque usada para explotar sitios web que construyen sentencias Lightweight Directory Access Protocol (LDAP) directamente desde datos facilitados por el usuario. Esta modalidad ha obtenido extensa aceptación como método de acceso a directorios desde Internet, iniciando estratégicamente con intranet corporativas. Cuando un atacante cambia una sentencia LDAP, el proceso se produce con los mismos permisos del componente que elaboró el comando, causando problemas de seguridad donde los permisos fundan derechos de consulta, modificación o eliminación de todos los elementos dentro del árbol LDAP.

Medidas de seguridad:

Filtrar la entrada de datos.

3.4.17. Ejecución de comandos de sistema operativo

Esta técnica permite ejecutar comandos del sistema operativo, a través de la manipulación de las entradas a la aplicación. Cuando no se realiza el filtrado de los datos dados por el usuario previamente de usarlos en el código de la aplicación se puede engañar a la misma para que ejecuten comandos del sistema operativo.

Medidas de seguridad:

Filtrar la entrada de datos.

3.4.18. Inyección de código SQL

Es un ataque en el cual sentencias SQL son inyectadas en las entradas de datos de los usuarios, causando la ejecución de comandos SQL predefinidos. Cuando la aplicación web no ejecuta una correcta filtración de los datos dados por el usuario, el atacante puede alterar la construcción de las sentencias SQL, a su vez el atacante puede modificar una sentencia SQL y el proceso se hará con los mismos permisos que el componente que elaboró el comando.

Figura 3.8: Ejemplo de un intento de inyección de sql (Elaboración propia)

Medidas de seguridad:

Filtrar la entrada de datos.

3.4.19. Inyección de código server side include SSI

Esta técnica permite a un atacante enviar código a la aplicación web, el cual será ejecutado localmente por el servidor. Si un atacante envía una sentencia SSI, puede poseer la capacidad de componer comandos arbitrarios de sistema operativo, o incluir el contenido de un archivo privado la próxima vez que la página sea ejecutada.

Medidas de seguridad:

Filtrar la entrada de datos.

3.4.20. Inyección XPath

Esta técnica busca explotar sitios web que construyen consultas XPATH a partir de datos suministrados por los usuarios. Si la aplicación edifica consultas XPath de carácter dinámico concatenando fundamentos inseguros dados por el usuario, un atacante puede inyectar datos en la consulta que admitan que sean desarrollados de forma diferente a la intención del programador y a fines de vulnerar el sistema.

Medidas de seguridad:

Filtrar la entrada de datos.

3.4.21. Ataques de revelación de información

Aborda los ataques diseñados para adquirir información específica del sistema sobre un sitio web. Hay que tomar en cuenta que mientras más información obtenga el atacante del sitio web, se hará mucho más fácil atacar y vulnerar el sistema.

3.4.22. Fuga de información

Se produce cuando un sitio web revela datos sensibles, como comentarios del desarrollador o mensajes de error, que pueden ayudar a un atacante a explotar el sistema. La información más vulnerable puede estar presente en comentarios HTML, mensajes de error, código fuente, o simplemente en una vista normal. Esta modalidad es aplicada sobre datos considerados confidenciales y que a su vez no son correctamente protegidos por el sitio web. Las categorías principales de fuga son los comentarios en el código, mensajes de error y datos confidenciales a simple vista.

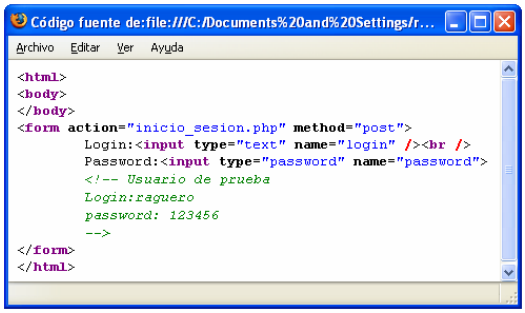


Figura 3.9: Fuga de información (Elaboración propia)

Medidas de seguridad:

Filtrar la entrada de datos, eficiente almacenamiento de la información, garantizar la mejor forma de utilización de la información.

3.4.23. Subida de directorios (path traversal)

Este ataque fuerza el acceso a archivos, directorios y comandos que se encuentran fuera del directorio raíz de la web. Cualquier dispositivo que expone una interfaz basada en HTTP es altamente vulnerable a path traversal. El principal ataque de path traversal utiliza la secuencia de caracteres especiales "../"para alterar el lugar del recurso solicitado en la URL. Como consecuencia de este ataque, el resultado puede revelar código fuente porque el archivo es efectuado como texto en lugar de un script ejecutable.

Medidas de seguridad:

Filtrar la entrada de datos.

3.4.24. Localización de recursos predecibles

A partir de suposiciones, el ataque es una búsqueda por fuerza bruta de contenido que no se espera ser visto públicamente. Estas búsquedas por fuerza bruta resultan fáciles ya que los archivos ocultos, de configuración, temporales y de ejemplo, generalmente suelen seguir convenciones de nombre comunes y residen en ubicaciones estándares.

Medidas de seguridad:

Control de ataques de fuerza bruta.

3.4.25. Referencia directa de objetos

Esta vulnerabilidad se presenta cuando un usuario puede acceder directamente a un objeto, sin necesidad de realizar las acciones propias de la aplicación, sino que, es llevado a cabo un comportamiento forzado por la manipulación directa del usuario. Es muy común encontrar aplicaciones web, que reciben en la URL parámetros para servir ciertos objetos a los usuarios, el usuario fácilmente puede cambiar la referencia a los objetos y acceder a los mismos sin ningún tipo de inconveniente.

Medidas de seguridad:

Establecer controles de sesión en procesos críticos.

3.4.26. Ataques lógicos

Este ataque hace uso del abuso o explotación del flujo lógico de una aplicación web. La lógica de la aplicación es el flujo de procedimientos esperados para realizar una acción determinada. Un atacante puede ser capaz de burlar o abusar de un proceso específico, o una serie de pasos para completar una acción particular, para dañar un sitio web.

3.4.27. Abuso de funcionalidad

Es una técnica de ataque que usa las propias características y funcionalidad de un sitio web para defraudar, burlar o violar los mecanismos de control de acceso. Cuando una parte de la funcionalidad se encuentra abierta al ataque, un atacante podría dañar a otros usuarios o quizás llevar a cabo un fraude sobre el sistema completo. Estas técnicas son principalmente entrelazadas con otras categorías de ataques de aplicación web, como la ejecución de un ataque de codificación para introducir una cadena de consulta que cambie una función de búsqueda web en un proxy web remoto.

Medidas de seguridad:

Establecer controles de sesión en procesos críticos, uso de captcha.

3.4.28. Ataques de denegación de servicios

Es una técnica de ataque que tiene la intención de impedir que un sitio web permita la interacción con los usuarios. Estos ataques son realizados en la capa de red, y son posibles en la capa de aplicación. Pueden ocurrir privando a un sistema de recursos críticos, explotando vulnerabilidades o mediante un abuso de funcionalidad. Mediente de ellos se intenta consumir todos los recursos disponibles del sistema (CPU, memoria, espacio de disco, etc.). Existen determinados tipos de ataques DoS: contra un usuario específico, contra un servidor de base de datos y contra un servidor web.

3.4.29. Anti-automatización insuficiente

Se produce cuando un sitio web permite a un atacante automatizar un proceso que sólo debería llevarse a cabo de forma manual. Ciertas funcionalidades de sitios web deberían ser protegidas contra ataques automáticos. Los robots automáticos (programas) o atacantes podrían reiteradamente intentar engañar o explotar el sistema.

Medidas de seguridad:

Uso de captcha, verificar la autenticación de un usuario en todo momento.

3.4.30. Validación de proceso insuficiente

Se produce cuando un sitio web permite a un atacante saltar o evadir el flujo de control previsto de una aplicación. Si el usuario no es verificado ni reforzado, el sitio web podría ser vulnerable a la explotación o el engaño. Si el usuario ejecuta ciertos pasos de forma incorrecta, puede que ocurra un error en la integridad de los datos. Los sitios web habitualmente indican el estado del usuario con el uso de cookies o campos ocultos de formulario. No obstante cuando el identificador es almacenado en la parte cliente dentro del navegador web, debe ser verificada la integridad de los datos, de lo contrario un atacante puede crear el flujo de tráfico esperado, alterando el estado actual.

Medidas de seguridad:

Establecer controles de sesión en procesos críticos, verificar la autenticación de un usuario en todo momento.

Capítulo 4

Medidas de Seguridad y buenas prácticas

En este capítulo se definen cada una las medidas de seguridad necesarias para corregir las vulnerabilidades y amenazas de seguridad mencionadas en el segundo capítulo. Consideraciones, políticas y controles que deben implementarse en conjunto con funcionalidades y características de una aplicación web.

Las medidas de seguridad, son acciones defensivas usadas para eliminar, reducir o mitigar los riesgos de ataques y vulnerabilidades. En otras palabras se traducen en tecnologías o módulos que podemos usar para prevenir, disminuir o no permitir ataques. Es necesario identificar e implementar medidas de seguridad basado en un análisis de amenazas, para asegurarnos que la aplicación está protegida.

4.1. Capa Física

4.1.1. Establecer políticas de redundancia de la información sensible de la empresa

Ante la posibilidad de la ocurrencia de alguna catástrofe natural, es imprescindible que la información realmente importante en una empresa se encuentre duplicada en otro espacio físico distinto al original. Esto a pesar de implicar una gran inversión a nivel monetario, permite establecer un proceso de recuperación de información.

4.1.2. Detección y corrección de fallas a nivel físico

El constante chequeo de las posibles fallas a nivel físico que pueden tener nuestros sistemas es una medida de seguridad que no es aplicada en todas las empresas, realizar simulaciones de fallas, permite detectar y corregir los errores presentados, estas simulaciones pueden ser de gran ayuda a la hora de que ocurra un evento real inesperado, que pueda vulnerar la seguridad de nuestros sistemas.

4.1.3. Bloquear el acceso a la sala de servidores

Hay que preveer que sólo las personas autorizadas tengan acceso físico a la sala de servidores, Por lo que, es necesario asegurar que hay buenas cerraduras en la puerta de la sala de servidores. Por supuesto, esta limitación no representa nada si no se utiliza, por lo que también es necesario exigir que las políticas de estas puertas cerradas se cumplan cuando la sala este desocupada, y estas políticas deberían establecer de la misma manera quién es la persona indicada para tener la llave o la herramienta de acceso a la misma.

4.1.4. Establecer vigilancia

Bloquear el acceso a la sala de servidores corresponde a un primer paso, sin embargo, este bloqueo puede ser vulnerado, o simplemente alguien con acceso autorizado puede hacer uso indebido de su autorización. De alguna manera se necesita saber quien entro o salio, en que fecha, a que hora. Un libro de registros donde se firmen las entradas y las salidas, es la forma más elemental de lograr este objetivo, pero existen soluciones más complejas y seguras por ejemplo, el uso de tarjetas inteligentes o siendo aún más complejos, el escaneo biometrico, para permitir el acceso.

Una cámara de vídeo de vigilancia, situada en un lugar donde se haga difícil manipularla o desactivarla, pero que de una buena vista de las personas que entran y salen debe complementar cualquier mecanismo de vigilancia adoptado.

4.1.5. Asegurar que los dispositivos más vulnerables se encuentran en un lugar con díficil acceso

Hay que recordar que no es sólo de los servidores que debemos de preocuparnos. Un atacante puede conectar una portátil en nuestra red y con el uso de un sniffer capturar el tráfico. Por ende, debemos asegurarnos que el mayor número de dispositivos de red tienen díficil acceso físico.

4.1.6. Utilizar rack para colocar los servidores

Los Racks[1] permiten juntar todos los equipos en un mismo espacio físico similar a un estante, por lo que, ocupan menos espacio en la sala de servidores. A su vez pueden ser atornillados al piso por lo que se hace casi imposible robar todos los equipos que ahi se colocan.

[1]Los racks son un simple armazón metálico con un ancho normalizado de 19 pulgadas, mientras que el alto y el fondo son variables para adaptarse a las distintas necesidades. El armazón cuenta con guías horizontales donde puede apoyarse el equipo, así como puntos de anclaje para los tornillos que fijan dicho equipo al armazón. En este sentido, un rack es muy parecido a una simple estantería.

Figura 4.1: Ejemplo de rack para alojar servidores (Elaboración propia)

4.1.7. Controlar el acceso a computadoras ociosas

Desconecte y/o elimine las computadoras que no están siendo utilizadas, bloquee las puertas de las oficinas vacías, incluidas las que están vacías temporalmente mientras que un empleado esta almorzando, o esta enfermo. Equipe a los equipos que deben permanecer en zonas abiertas con tarjetas inteligentes o lectores biométricos para que le sea más difícil a las personas no autorizadas usarlas.

4.1.8. Proteger las copias de seguridad

Se recomienda que las copias de seguridad sean encerradas en un cajón de seguro por lo menos. Idealmente, un conjunto de copias de seguridad deben mantenerse fuera del sitio físico de la empresa, pero de igual forma se debe garantizar que estan guardadas en un sitio muy seguro.

No hay que olvidar el hecho de que algunos trabajadores pueden realizar copias de seguridad de sus trabajos en disquetes, pen drives, o discos duros externos. Si esta práctica se permite o alienta, asegúrese de disponer de políticas que requieren que las copias de seguridad se guardan en todo momento.

4.1.9. Desactivar unidades que permitan almacenamiento en equipos con información crítica

Para no permitir que personas mal intencionadas copien la información de la empresa en medios extraíbles, puede desactivar o eliminar las unidades de disquetes, puertos USB, y otros medios de conexión de unidades externas.

4.2. Capa de Red

4.2.1. Uso de Cortafuegos (Firewall)

Como sabemos un firewall es un elemento de hardware o software utilizado en una red de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas según las políticas de red que haya definido la organización responsable de la red.

Esto nos va a permitir aislar la red privada de internet. Permitir el acceso sólo a unos servicios y prohibir el resto. Permitir el acceso sólo a los puertos especificados.

Unas de las principales ventajas que nos provee el uso de un cortafuegos es que permite configurar una Zona DesMilitarizada mejor conocida como una DMZ, con el objetivo de que las conexiones desde la red interna y la externa a la DMZ estén permitidas, mientras que las conexiones desde la DMZ sólo se permitan a la red externa los equipos (hosts) en la DMZ no pueden conectar con la red interna. Esto permite que los equipos (hosts) de la DMZ puedan dar servicios a la red externa a la vez que protegen la red interna en el caso de que intrusos comprometan la seguridad de los equipos (host) situados en la zona desmilitarizada. Para cualquiera de la red externa que quiera conectarse ilegalmente a la red interna, la zona desmilitarizada se convierte en un callejón sin salida.

La DMZ se usa habitualmente para ubicar servidores que es necesario que sean accedidos desde fuera, como servidores de e-mail, Web y DNS.

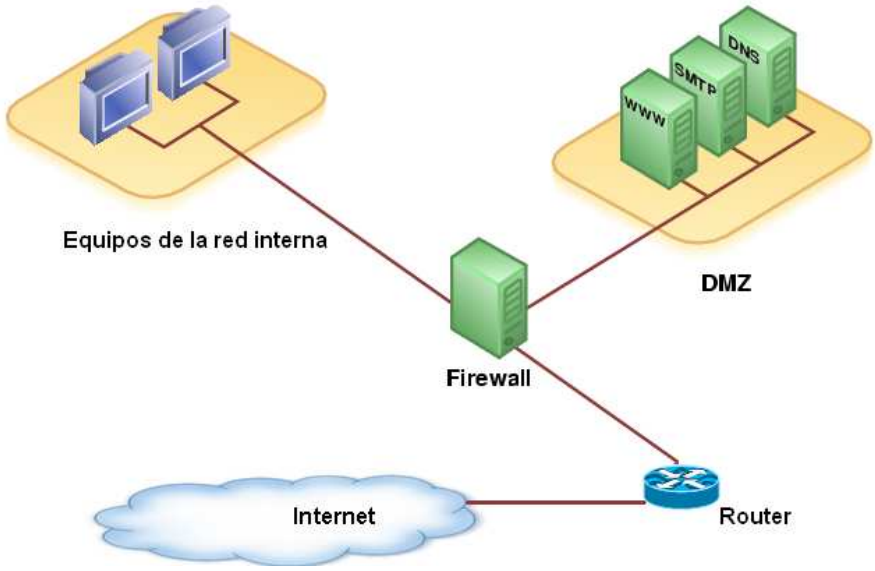


Figura 4.2: Ejemplo de una DMZ (Elaboración propia)

Las conexiones que se realizan desde la red externa hacia la DMZ se controlan generalmente utilizando port address translation (PAT).

Una DMZ se crea a menudo a través de las opciones de configuración del firewall, donde cada red se conecta a un puerto distinto de éste, esta configuración se llama firewall en trípode (three-legged firewall). Un planteamiento más seguro es usar dos cortafuegos, donde la DMZ se sitúa en

medio y se conecta a ambos cortafuegos, uno conectado a la red interna y el otro a la red externa. Esta configuración ayuda a prevenir configuraciones erróneas accidentales que permitan el acceso desde la red externa a la interna. Este tipo de configuración también es llamado firewall de subred monitoreada (screened-subnet firewall).

Otras ventajas que nos provee el uso de firewall son:

1. Protege de instrusos. El acceso a ciertos segmentos de la red de una organización, sólo se permite desde máquinas autorizadas de otros segmentos de la organización o de Internet.
2. Protección de información privada. Permite definir distintos niveles de acceso a la información de manera que en una organización cada grupo de usuarios definido tendrá acceso sólo a los servicios y la información que le son estrictamente necesarios.
3. Optimización de acceso.- Identifica los elementos de la red internos y optimiza que la comunicación entre ellos sea más directa. Esto ayuda a reconfigurar los parámetros de seguridad.

Las limitaciones serían:

1. No protege contra aquellos ataques cuyo tráfico no pase a través de el.
2. No protege de las amenazas por ataques internos o usuarios negligentes. No puede prohibir a espías corporativos copiar datos sensibles en medios físicos de almacenamiento (diskettes, memorias, etc).
3. No protege contra ataques de ingeniería social.
4. No protege los fallos de seguridad de los servicios y protocolos.
5. No entienden el significado de HTTP.
6. Dejan pasar el 100

4.2.2. Uso de VPN

La VPN nos va a permitir extender la red local sobre una red pública o no controlada, como por ejemplo Internet. Este tipo de tecnologías es muy usada cuando queremos por ejemplo: conectar dos o más sucursales de una empresa utilizando como vínculo Internet, o permitir a los miembros del equipo de soporte técnico la conexión desde su casa, o que un usuario pueda acceder a su equipo doméstico desde un sitio remoto, como por ejemplo un hotel. Todo ello utilizando la infraestructura de Internet.

Para hacerlo posible de manera segura es necesario proporcionar los medios para garantizar la autenticación, integridad y confidencialidad de toda la comunicación.

Entre las ventajas que tenemos al hacer uso de esta tecnología encontramos:

1. Integridad, confidencialidad y seguridad de datos.
2. Reducen costos y son sencillas de usar.

4.2.3. Red Privada con direcciones internas

Consiste en colocar los recursos importantes tras una red privada. A diferencia de las direcciones IP publicas las cuales pueden ser alcanzables desde cualquier parte. Las direcciones IP privadas solamente pueden ser alcanzadas si se esta dentro de la misma subred.

4.2.4. Uso de sistemas de detección de intruso o ataques (IDS - Intrusion Detection)

Este tipo de programas son usados para detectar accesos desautorizados a un computador o a una red. Estos accesos pueden ser ataques de habilidosos hackers, o de Script Kiddies que usan herramientas automáticas.

El IDS suele tener sensores virtuales (por ejemplo, un capturador de tramas (sniffer) de red) con los que el núcleo del IDS puede obtener datos externos (generalmente sobre el tráfico de red). El IDS detecta, gracias a dichos sensores, anomalías que pueden ser indicio de la presencia de ataques o falsas alarmas.

4.2.5. Bloqueos de IP y limpiado de Buffer

Para contrarrestar los ataques de DoS, es muy frecuente utilizar técnicas de bloqueo, con lo que se impide a una cuenta de usuario, o a una o a un rango de direcciones IP, realizar peticiones, es decir, todas las peticiones de recursos enviados desde una dirección IP o cuenta de usuario son ignoradas por completo. Otra medida es ignorar las conexiones TCP incompletas para no llenar el buffer que almacena las conexiones entrantes.

4.3. Capa de Sistemas Operativos y Servicios

4.3.1. Uso de Antivirus

Básicamente, es un programa cuya finalidad es prevenir y evitar la infección de virus, impidiendo también su propagación, cuenta con la capacidad para detectar y eliminar los virus y restaurar los archivos afectados por su infección.

La única forma de mantener su sistema seguro es mantener su antivirus actualizado y estar constantemente leyendo sobre los virus y las nuevas tecnologías. La función de un programa antivirus es detectar, de alguna manera, la presencia o el accionar de un virus informático en una computadora. Éste es el aspecto más importante de un antivirus, pero, las empresas deben buscar identificar también las características administrativas que el antivirus ofrece. La instalación y administración de un antivirus en una red es una función muy compleja si el producto no lo hace automáticamente. Es importante tener en claro la diferencia entre “detectar” e “identificar” un virus en una computadora. La detección es la determinación de la presencia de un virus, la identificación es la determinación de qué virus es. Aunque parezca contradictorio, lo mejor que debe tener un antivirus es su capacidad de detección, pues las capacidades de identificación están expuestas a muchos errores y sólo funcionan con virus conocidos.

4.3.2. Actualizaciones periódicas

En la mayoría de los casos las empresas que desarrollan software continuamente realizan actualizaciones o versiones que poseen caracteristicas mejoradas de algún funcionamiento en particular, o simplemente acomodan un hoyo de seguridad que posea el mismo. Implementar una política de continua actualización de todos los sistemas instalados en todas las computadoras de nuestra empresa es una medida de seguridad ampliamente recomendada.

4.3.3. Permisos de ejecución y cuentas de usuario limitadas

La mayoría de los usuarios de nuestro sistema operativo, no tienen porque tener privilegios de niveles altos, tipo administrador, debido a que ellos realizan operaciones que no requieren un nivel tan alto de privilegio. Por lo que es recomendable restringir adecuadamente las operaciones que realizan los usuarios y los permisos de ejecución que estos posean sobre ciertos servicios criticos del Sistema Operativo.

4.3.4. Eliminación de servicios y configuraciones innecesarias

La mayoría de los ataques a nivel de Sistemas Operativos son posibles gracias a que los usuarios de la red interna gozan de privilegios y poseen servicios que no son necesarios para que estos puedan realizar sus tareas diarias, Las personas encargadas de la seguridad informática deben preveer este tipo de situaciones y eliminar servicios a todos los usuarios que no los utilicen, de la misma manera, no se debe pensar en utilizar configuraciones por defecto a menos que estan hayan sido aprobadas como seguras.

4.3.5. Implementar políticas para crear contraseñas seguras

Se deben pensar e implementar políticas para que los usuarios con privilegios altos, creen contraseñas que posean un nivel de robustez necesario que no permita que sea facilmente vulnerada. Existen políticas muy utilizadas para tales fines, como: caducar contraseñas, no permitir el uso de contraseñas anteriores por un período de tiempo determinado, utilizar combinaciones de letras y caracteres con un tamaño mínimo específico. Es recomendable aplicar este tipo de métodos.

4.3.6. Implementar técnicas de encriptación para los datos manejados en servicios críticos

Toda encriptación se basa en un algoritmo que se encarga de codificar la información para que no sea legible a simple vista. Este método representa una de las alternativas más fiables de seguridad de los datos. Por lo que, resulta recomendable que a los datos manejados en servicios críticos se le apliquen cualquiera de las múltiples técnicas de encriptado según sea el caso, con esto podemos insertar una capa más de seguridad en la información manipulada.

4.3.7. Usar la última versión estable

Siempre la última versión estable trae consigo correcciones a problemas y vulnerabilidades anteriores.

4.3.8. Deshabilitar todo lo que no sea necesario

Sobre todo con instalaciones por defecto traen muchas características que no necesariamente se necesitan en la aplicación, puertos para Proxy, consolas de debug o de administración.

4.3.9. Revisar accesos por defecto o comunes

Buscar interfaces de administración que generalmente vienen por defecto. Cuantas de usuario por defecto. Cuentas de usuario comunes en estapas de desarrollo y que pueden ser arrastradas al ambiente de producción, test/test, tomcat/tomcat, admin./12345678 Puertos por defecto para administración remota. Consolas de debug. etc.

4.3.10. Actualizaciones frecuentes y parches

Revisar periódicamente actualizaciones, parches contra nuevas vulnerabilidades, suscribirse a listas de correo con notificaciones cuando se descubran nuevas vulnerabilidades y salgan nuevos parches.

4.3.11. Blindar el modulo CGI o Server side scripting tecnology

En el caso de algunas tecnologías como php o asp existen una lista de configuraciones o variables de entorno, permisos que pueden resultar peligrosos para la ejecución en un servidor y que se recomienda que en la medida de lo posible estos estén deshabilitados. Por ejemplo: register globals, Remote Server side include, Command execution.

4.3.12. Colocar en el directorio web raíz sólo archivos públicos

Los archivos de configuraciones, contraseñas, o que simplemente pertenecen al modelo de negocios no deberian de estar alojados en el directorio raíz de una aplicación web ya que pueden ser accedidos por cualquier usuario en internet. Es muy común que los desarrolladores piensen que porque no se haga un enlace directo a un archivo este no pueda ser accedido, sin embargo gran parte de los ataques resultan de solicitar un recurso al servidor que parecía invisible para los usuarios y que por encontrarse en el directorio raíz es servido. Mueva archivos básicos que no requieran acceso por sobre la raíz web.

4.3.13. Uso de Directorios privados dentro del directorio web raíz

Actualmente es posible en algunas aplicaciones tener directorios privados, cuyos recursos no pueden ser accedidos por los usuarios, dentro del directorio raíz, esta característica permite mantener seguros archivos que son importantes para el funcionamiento de la aplicación, pero que a su vez no deben ser manipulados por cualquier usuario o aplicación, este tipo de medida de seguridad lo podemos ver reflejado por ejemplo en el directorio WEB-INF de los servidores Tomcat.

4.3.14. Correcta configuración de permisos en los directorios y archivos

En numerosas ocasiones es necesario establecer permisos específicos a directorios que existen dentro de una ruta en la estructura de archivos, así como a todos los archivos almacenados en esos directorios. Por lo que se hace necesario establecer la lista de directorios a los cuales se les va a aplicar estos privilegios, ya que el tener todos los directorios con permisos incorrectos puede traer graves consecuencias de vulnerabilidad en los mismos, Por ejemplo un usuario mal intencionado puede eliminar archivos de configuración o de acceso, que son indispensables para una aplicación.

4.3.15. Cifrar datos sensibles

La información que es demasiado importante para una empresa, debería de almacenarse de modo que no sea comprensible para cualquier persona. Para tal fin existen múltiples métodos que van desde los más sencillos, hasta los más complicados, La idea es utilizar el método apropiado que requiere la empresa para mantener la seguridad en los mismos.

4.3.16. Realizar copias de seguridad de los datos

Hay que tener siempre presente que los desastres suceden y, si no ha guardado la información y los archivos importantes en un sistema de almacenamiento independiente, se pueden perder todos los datos vitales de las aplicaciones. Lo que implica millones de gastos en las empresas. Por lo tanto, Realice copias de seguridad de los datos periódicamente y guarde las copias fuera del lugar de trabajo.

4.3.17. Actualizar periódicamente los manejadores de bases de datos

Instale los últimos Service Packs, parches o actualizaciones de la base de datos. Esta es la mejor manera de asegurar que no posee vulnerabilidades que fueron descubiertas y para las cuales ya existe alguna solución.

4.3.18. Limitar adecuadamente el acceso a los datos

El aislamiento físico y lógico constituye la base de la seguridad de los datos. Los equipos que alojan una base de datos se deben encontrar en una ubicación protegida físicamente.

4.3.19. Proveer vistas del repositorio de los datos

Resulta muy frecuente que las aplicaciones web sólo realicen consultas sobre una base de datos, o sobre tablas especificas, es recomendable que se generen vistas de tablas donde un usuario solamente pueda consultar y manipular datos en especifico, ya en el caso de vulnerar esta vista la data transaccional, no sufre ningún tipo de inconveniente.

4.3.20. Chequeo de configuración desde cero

Para evitar vulnerabilidades por una configuración por defecto, o simplemente para verificar que todo este correcto, un usuario experto debe periodicamente revisar las configuraciones de los repositorios de datos. Este chequeo ayuda enormemente a corregir errores que se puedan mantener al instalar un repositorio con configuraciones por defecto, o mantener usuarios activos que ya no utilizan los sistemas.

4.3.21. Verificación de credenciales y permisos

Hay que tener siempre en mente que las vulnerabilidades pueden ser explotadas por usuarios que se consideran de çonfianzaz que en base a esto, se le otorgan privilegios que no corresponden a las funciones que estos realizan, Por lo que, el administrador de las bases de datos debe verificar que los usuarios creados cuenten con los privilegios suficientes para ejecutar todas sus funciones, sin posibilidad de afectar otros sistemas.

4.3.22. Forzar el uso de contraseñas seguras

Se debe implementar una política de seguridad que forzé a un usuario, o administrador, crear contraseñas que tengan un nivel mínimo que sea considerado como seguro para que no sea una tarea fácil el adivinar o predecir la contraseña de un usuario, y con esto lograr vulnerar los datos.

4.4. Capa de Aplicación

4.4.1. Validar las entradas de datos

Consiste en asegurar que las entradas de datos sean seguras antes de usarlas. La forma más segura de detectar las entradas dudosas es utilizando estrategias como la lista blanca *whitelist* y la negra *blacklist* y no ejecutando acciones con estas entradas sospechosas.

Lista blanca o lista negra ¿Cuál estrategia decidir?

La lista blanca (whitelist): Básicamente identifica las entradas buenas y las malas en una lista de buenas entradas conocidas.
La lista negra (blacklist): Se desarrolla identificando las malas entradas conocidas.

En su mayoría los profesionales prefieren utilizar la lista blanca ya que la negra puede arrojar una mala entrada, como segura.

4.4.2. Filtrar la entrada de datos (Filter Input)

El manejo de la entrada de datos es la definición de cómo una aplicación, sistema o servidor manipula las entradas facilitadas por los usuarios, redes, clientes u otros sistemas. Es importante desarrollar de manera segura el manejo de entradas para prevenir vulnerabilidades.

Consiste en probar la validez de los datos, afirmando que los mismos estén filtrados correctamente en la entrada y eliminar el riesgo de confianza y utilización de datos corrompidos. Esta estrategia requiere de tres pasos: identificar las entradas, filtrar las entradas y distinguir entre datos filtrados y corrompidos.

La ventaja del filtrado recae sobre los usuarios finales ya que el mecanismo de seguridad se admite de forma menos visible: la desventaja es que el filtrado es que es difícil de conseguir ya que muchas aplicaciones lo tienen aplicado en un lugar del código.

Ejemplo del filtrado por lista blanca

Se esperan que todos los caracteres pertenezcan a la lista A-Za-A que se utiliza para proteger una aplicación que este corriendo en entorno UNIX contra inyección en Shell. El atacante provee la entrada, se aplica el filtro en la entrada se tratan los caracteres que no están en la lista, el ls-l se guardan porque están en la lista y finalmente el intento de ataque falla debido a que la entrada es segura.

Ejemplo de filtrado por la lista negra

Cuando los caracteres en un sistema se conocen como maliciosos la entrada original se substituye. Esta estrategia presenta varios problemas: no protege contra amenazas desconocidas ni futuras.

4.4.3. Escapado de salida (escape output)

Es una estrategia similar a una aplicación, un servidor o un sistema maneja la salida. El manejo de la salida intenta prevenir las vulnerabilidades de Cross Side Scripting XSS, También puede ser importante en áreas semejantes.

Consiste en la práctica de caracteres especiales “outputescaping” o de codificación donde escapan caracteres para preservar su significado original. Con el filtrado de entrada cuando hay escape de salida se dan tres pasos diferentes: identificar la salida, escapar la salida y distinguir entre datos codificados (escapados) o no codificados (escapados).
La codificación permite elaborar el contenido que va a ser generado al usuario para manipular un

carácter peligroso.

Ejemplo de escapado de salida

La función de php htmlentities es un ejemplo de escapado de salida, esta función transforma todos los caracteres HTML a su respectiva entidad HTML, es decir, aplicar esta función al caracter (") comilla doble, produce la siguiente salida ", que corresponde a la entidad HMTL correspondiente a la comilla doble.

4.4.4. Uso de herramientas para distinguir autómatas de humanos (CAPTCHA)

Significa Completely Automated Public Turing Test To Tell Computers and Humas Apart (prueba de turing pública y automática para diferenciar a máquinas y humanos). Se basa en una prueba de carácter desafío respuesta utilizada para establecer cuando el usuario es un computador o en un humano, se da generalmente en una imagen distorsionada e pantalla con un conjunto de caracteres que el usuario debe introducir.
Esta prueba se utiliza para impedir que un robot o computador tenga acceso a ciertos servicios. Se han presentado problemas de accesibilidad con usuarios discapacitados, pero a su vez se han logrado otras alternativas como la vía auditiva.

Figura 4.3: Ejemplo de implementación CAPTCHA (Elaboración propia)

4.4.5. Detección de comportamiento irregular

Consiste en detectar cuando una aplicación o usuario esta actuando de manera irregular, este proceso requiere cierto grado de complejidad. Depende del tipo de usuario que actúa y el nivel de seguridad requerido. Cuando se detecta un patrón irregular se levanta una alerta y se toman medidas pertinentes según el caso, medidas que varían según la aplicación.

Ejemplo de Detección de comportamiento irregular

Por ejemplo múltiples peticiones desde la misma dirección IP en un período corto de tiempo, puede significar un ataque de DOS, ante esto, se puede bloquear las peticiones que se generen desde ciertas direcciones IP.

4.4.6. Control de ataques de fuerza bruta

Los ataques por fuerza bruta se determinan llevando el control de intentos fallidos en un tiempo determinado. Existen acciones contra ataques de fuerza bruta por bloqueo, entre ellas bloqueos de credenciales de acceso temporal e indefinido, de dirección IP indefinidamente y temporalmente y colocar un delay entre un intento fallido y otro.

Figura 4.4: Bloqueo de cuenta, medida que controla ataques de fuerza bruta (Imagen tomada de https://ws027.juntadeandalucia.es/profesionales/eatencion/ayuda/img/UsuBloq1dia.gif)

4.4.7. Inducir al usuario hacia la seguridad

Consiste en hacerles tomarles medidas de seguridad a los usuarios ya que de nada sirven todas las tecnologías de seguridad en los sitios web si el usuario actúa de manera descuidada. Aunque no se espera que todos los usuarios sean expertos informáticos y es por ello que a veces de forma inconciente se les obliga a tomar medidas de seguridad como la longitud mínima de caracteres en las contraseñas, caducidad en las mismas, certificados digitales en el cliente, informar que la aplicación web no envía correos solicitando contraseñas o números de tarjetas de crédito, etc.

4.4.8. Uso de contraseñas seguras

Las contraseñas más seguras y fuertes son aquellas que se constituyen con caracteres diferentes (letras, números, mayúsculas, minúsculas y caracteres extendidos) aunque por lo general es usuario escoge contraseñas débiles y vulnerables, generalmente cortas y constituidas con un solo conjunto de caracteres, ellos las utilizan para recordarlas y se forman en mayoría, cumpleaños, aniversarios, nombres, etc. Lo que representa que alguien que conoce al usuario o por ingeniería social puede

descubrir la contraseña.

Figura 4.5: Ejemplo de inducir al uso de contraseñas seguras (Elaboración propia)

4.4.9. Caducidad de las contraseñas

Consiste en sistemas que obligan ala usuario a cambiar la contraseña periódicamente creando tiempos de expiración y no permitiendo que la nueva contraseña sea igual a la anterior o anteriores. Esta medida de seguridad a veces incomoda a los usuarios aunque sea para su beneficio y es aconsejable para aplicaciones que contengan información personal delicada, transacciones financieras electrónicas, compras por Internet, etc.

4.4.10. Proveer al usuario múltiples consejos de seguridad

En la seguridad de las aplicaciones web, es primordial que los usuarios conozcan los riesgos a los que se exponen al igual que las medidas que deben tomar para minimizar y evitar los riesgos. Algunas aplicaciones aconsejan a los usuarios el uso de contraseñas fuertes a través de la combinación de caracteres, creación de acrónimos, la no utilización de passwords que contengan información personal, cambiar la contraseña periódicamente, igualmente que ser precavido al recibir correos que soliciten actualización de datos. Todas estas situaciones pueden hacer del usuario un blanco fácil para el atacante.

Figura 4.6: Ejemplo de Proveer al usuario múltiples Consejos de seguridad (Elaboración propia)

4.4.11. Eficiente uso de la manipulación de la información

Cuando la información es privada se dice que es sensible o delicada ya que el contenido compromete la integridad o nivel de seguridad de un sistema o persona. Las aplicaciones Web manejan

información de dos tipos pública y privada. La pública o no sensible se describe generalmente como noticias, publicaciones periódicas, chistes, entretenimiento, foros públicos, etc. y no necesariamente deben ser encriptadas, en cambio, la privada contiene generalmente datos sensibles como teléfonos, direcciones, correos electrónicos, números de tarjetas de créditos, esta información necesariamente debe ser encriptada para que no sea legible por cualquier persona que vulnere el sistema.

4.4.12. Eficiente protección en la transición de la información

El viaje de la información a través de Internet por defecto es en una forma no segura ya que el protocolo IP no define el cifrado a la hora de llevar la información de su origen al destino. Tampoco el protocolo de transporte TCP que se encarga de establecer una sesión lógica entre emisor y el receptor. El protocolo HTTP es de texto plano y cualquier persona con un capturador y analizador de tramas de red puede capturar la información que circule en red y así identificar passwords, claves de cuentas bancarias,etc. esta información debe ser enviada utilizando un medio seguro, que asegure que en caso de existir intersección y robo de la misma, no sea entendible para el atacante.

4.4.13. Eficiente almacenamiento de la información

Consiste en que toda la información debe ser almacenada de forma segura ya que si un servidor es comprometido toda la información que posea igualmente estará comprometida. Cuando la información es almacenada si ningún tipo de precaución el atacante puede acceder automáticamente a la misma y usarla de manera fraudulenta. Cuando se almacenan las contraseñas de un usuario de un sitio Web se puede hacer una función de HASH que es aquella que dado el conjunto de datos de entrada generan una única salida.
Cuando se necesita almacenar información que va a ser utilizada posteriormente, no cifrada y en su forma original, se utiliza una función criptográfica de doble vía, es decir, a la inversa. Esto es útil para el almacenamiento de números de tarjetas de créditos, documentos confidenciales, datos personales, correos electrónicos, etc.

4.4.14. Garantizar la mejor forma de utilización de la información

Indica el cuidado que deben tener los usuarios al utilizar la información ya que se hallan aplicaciones que exponen datos confidenciales a simple vista, de forma que cualquiera puede mirar el monitor y dar con información delicada. Para proteger la información se busca ofuscar u ocultarla en su despliegue en pantalla.
Se ve en los campos de contraseñas en un formulario HTML que muestran la información de forma enmascarada y así se evita que la misma quede en evidencia si alguien esta observando la pantalla.

4.4.15. Verificar la autenticación de un usuario en todo momento

A muchos desarrolladores se les olvida que un recurso puede ser accedido directamente colocando la URL del mismo, y creen que los usuarios solo van a acceder a ellos por el menú que les

proveen, este descuido es una de las vulnerabilidades más frecuentes a nivel de aplicaciones web. Por lo que se debe: realizar "siempre.ªntes de servir cualquier página de acceso privado, o ejecutar alguna acción de acceso privado, la verificación de autenticidad, esto se logra, accediendo a las variables de sesión que se crean en el servidor al momento en el cual un usuario inicia sesión.

4.4.16. Seguridad en profundidad para la recuperación en contraseñas

Normalmente la forma de recuperar una contraseña, es solicitando la respuesta a una pregunta secreta, previamente definida, pero este método a su vez, puede ser objeto de un ataque de fuerza bruta, la seguridad en profundidad implica colocar tantos niveles de seguridad como sea posible, si en vez de recuperar la contraseña usando este método, la misma es enviada a un correo electrónico proporcionado por un usuario al registrarse, el atacante no sólo va a tener que adivinar la respuesta secreta, sino que además debe conocer la contraseña y el correo electrónico de la víctima, esta técnica es altamente recomendada para tales fines.

Figura 4.7: Ejemplo de Seguridad en profundidad para la recuperación en contraseñas (Elaboración propia)

4.4.17. Establecer controles de sesión en procesos críticos

Existen procesos en una aplicación web que son altamente importantes, ya que permiten hacer operaciones a los usuarios autenticados que afectan directamente el modelo de negocios de nuestra empresa, por ejemplo, una página web de banca electrónica, te permite, no sólo consultar tus estados de cuentas, sino que se pueden realizar transferencias, pagos, con sólo hacer un clic. Por lo que resulta primordial, establecer controles de sesión que impliquen, duraciones cortas de sesión si un usuario esta inactivo, revalidación de las credenciales de sesión, no aceptar identificadores de sesión de variables GET y POST, utilizar SSL y TLS en identificadores de sesión, regenerar los identificadores de sesión en cada petición, aceptar sólo los identificadores de sesión generados por el servidor, destruir una sesión si el equipo que hace la petición es sospechoso.

Figura 4.8: Ejemplo de revalidación de credenciales de sesión (Elaboración propia)

Capítulo 5

Características principales de las aplicaciones web y sus consideraciones de seguridad

En la actualidad existen aplicaciones web para muchos fines: para comercio electrónico, portales informativos, páginas educativas, de entretenimiento, de servicios, entre otras. Cada una de estas aplicaciones desarrolladas para distintas plataformas, arquitecturas de software y por diferentes autores. A pesar de todas las diferencias que existen entre cada una de estas aplicaciones web, se pueden apreciar gran cantidad de características o funcionalidades comunes entre cada una de ellas. Estas características comunes se traducen en configuraciones típicas de los servidores, configuraciones de red, arquitecturas de software, uso de frameworks e incluso en acciones u operaciones que pueden realizar los usuarios.

Por ejemplo, es común encontrar distintas aplicaciones web alojadas en servidores compartidos o que las mismas interactúen con una base de datos. También es común encontrar aplicaciones que permiten a los usuarios ingresar información personal a través de formularios HTML o iniciar una sesión a través de estos formularios, sin importar la plataforma o el lenguaje en el que estén desarrolladas.

La implementación de estas características sin las medidas de seguridad o controles adecuados pueden implicar o exponer, en una aplicación web, alguna vulnerabilidad o amenaza poniendo en riesgo la seguridad de la misma.

Una muestra de ello es que muchas páginas web, poseen un administrador de contenido el cual es accedido, generalmente mediante un nombre de usuario y contraseña, a través de un formulario. Utilizando un ataque de fuerza bruta, probar todas las posibles combinaciones de nombre de usuario y contraseña hasta conseguir la correcta, un atacante puede entrar en dicho panel de administración, y de ahí obtener información confidencial de usuarios y/o colocar contenido inapropiado en la página. Por tanto es de suma importancia saber identificar estas características e implementar las medidas de seguridad apropiadas, para prevenir las posibles vulnerabilidades de seguridad asociadas.

El siguiente capítulo describe las características principales y más comunes, presentes en aplica-

ciones web, y sus posibles implicaciones en cuanto a vulnerabilidades y amenazas de seguridad. A su vez hace referencia a las medidas de seguridad o controles que se deben considerar para asegurar dichas funcionalidades.

5.1. Capa Física

5.1.1. Hardware propio de la empresa

La compañía se encarga de la adquisición y administración del hardware necesario para sus sistemas. Empresas grandes o con requerimientos específicos optan por esta opción, asumiendo los costos del hardware y su administración. Requieren espacio fisco adecuado.

Vulnerabilidades:

Catástrofes naturales, accidentes, accesos no autorizados, vandalismo, falla.

Medidas de Seguridad:

Establecer políticas de acceso, controlar el acceso físico al hardware, políticas y prácticas de respaldo, redundancia, planes de recuperación ante desastres, extintores de fuego, guardias de seguridad.

5.1.2. Hardware arrendado a terceros

Otra compañía se encarga de la adquisición y administración del hardware necesario para sus clientes. Empresas de hospedaje y servicios de outsourcing ofrecen a sus clientes este tipo de servicio. Librando al cliente de toda responsabilidad de adquisición de hardware y administración del mismo. Son esas empresas especializadas en este negocio quienes garantizan a sus clientes, los niveles de seguridad y disponibilidad que los mismos necesitan. Representan una buena opción para empresas cuya rama de negocio no esta directamente relacionada con el área tecnológica o no pueden adquirir o administrar la plataforma necesaria.

Sin embargo hay que considerar que se pudiera estar dejando en manos de un tercero información delicada. Información que nunca debería manejar una empresa diferente a la propietaria de la misma. Como es el caso de entidades bancarias o instituciones del gobierno, donde son ellos mismos quienes tienen que asumir la responsabilidad de resguardar la integridad y confidencialidad su data.

Vulnerabilidades:

Catástrofes naturales, accidentes, accesos no autorizados, vandalismo, falla.

Medidas de Seguridad:

Establecer políticas de acceso, controlar el acceso físico a los equipos, políticas y prácticas de respaldo, redundancia, planes de recuperación ante desastres, extintores de fuego, guardias de seguridad.

5.1.3. Administración directa en el equipo

Es una de las vías de administración más comunes. El equipo es administrado por una persona que tiene acceso físico al mismo. Esta situación generalmente se presenta cuando las empresas adquieren y administran el hardware por cuenta propia.

Vulnerabilidades:

Catástrofes naturales, accidentes, accesos no autorizados, vandalismo, falla.

Medidas de Seguridad:

Establecer políticas de acceso, controlar el acceso físico a los equipos, políticas y prácticas de respaldo, redundancia, planes de recuperación ante desastres, extintores de fuego, guardias de seguridad.

5.2. Capa de red

5.2.1. Equipos de acceso público a través de la red

Estos equipos deben ser alcanzables desde Internet. En la mayoría de los casos estos equipos poseen una dirección IP pública. Los servicios públicos, tales como un servidor web o un servidor de correo, se ejecutan en estos equipos. Son el punto de entrada de un atacante a la red, así que deben estar bien protegidos contra ataques y explotación de vulnerabilidades. A nivel de red sólo se debe habilitar el acceso a estos equipos en los puertos que estén destinados a ser públicos. A través de un firewall de borde de red y un firewall en el equipo se puede mitigar el riesgo de una gran cantidad de ataques conocidos. Las comunicaciones deben ser cifradas si se requiere privacidad.

Vulnerabilidades:

Acceso no autorizado, DoS, revelación de información, malware, ataques de hombre en el medio.

Medidas de Seguridad:

Balanceo de carga, cortafuegos (firewall), IDS, IPS, DMZ, filtrado por direcciones IP, VPN, honeypots, monitores de red, proxy, antivirus.

5.2.2. Equipos de acceso privado

Estos equipos no deben ser alcanzables desde Internet. Generalmente sólo son accedidas a través de la intranet. En ellos pueden haber servidores web y otros servicios, pero solamente destinados al uso interno, por parte de empleados u otras aplicaciones. Es común encontrar en estos equipos servidores de bases de datos, servidores LDAP, servidores de directorio e incluso otros servidores web. Se deben separar estos equipos de la red pública, impidiendo el acceso a ellos a través desde Internet. Configurar canales de comunicación seguro a estos equipos, por ejemplo a través de una VPN, puede permitir el acceso remoto seguro.

Vulnerabilidades:

Acceso no autorizado, DoS, revelación de información, malware.

Medidas de Seguridad:

Balanceo de carga, cortafuegos (firewall), IDS, IPS, DMZ, filtrado por direcciones IP, VPN, honeypots, monitores de red, proxy, antivirus.

5.3. Capa de sistema operativo y servicios

5.3.1. Sistema operativo

En el sistema operativo se ejecutan las aplicaciones y los servicios. Cada aplicación o servicio con sus respectivos permisos de ejecución y acceso a los recursos de la máquina. El sistema operativo controla todo en el sistema. Si un atacante compromete el sistema operativo, tiene acceso a todas las aplicaciones y sistemas en el mismo. Significa una de las peores catástrofes que puede ocurrir en seguridad. Por tanto es de suma importancia estar prevenido contra cualquier tipo de ataque o amenaza.

Vulnerabilidades:

Acceso no autorizado, DoS, malware, buffer overflow, intrusiones, deficiente autenticación, deficiente autorización, versiones viejas, falta de actualizaciones, configuraciones por defecto, exposición de servicios innecesarios, credenciales predecibles o por defecto.

Medidas de seguridad:

Cortafuegos (firewall), sistema de detección de intrusos (IDS), IPS, antivirus, actualizaciones automáticas, parches, chequeo de configurar desde cero, verificación de credenciales y permisos.

5.3.2. Hospedaje compartido

En un mismo servidor web se pueden servir diferentes dominios y cada uno de ellos ejecutando distintas aplicaciones. Una forma de conseguir esto es gracias a que el protocolo HTTP, desde su

versión HTTP/1.1, requiere el uso de la cabecera Host. Como resultado las compañías de hospedaje pueden alojar muchos dominios en un mismo servidor, sin necesidad de tener una dirección IP pública para cada dominio.

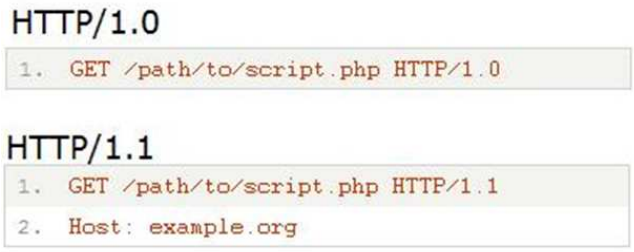


Figura 5.1: Diferencia entre el protocolo http 1.0 y http1.1 (Elaboración propia)

Las aplicaciones en estos servidores comparten el entorno del sistema operativo. Así como servicios en el mismo, servidores web, servidores de correo y servidores de bases de datos.

Las empresas de hospedaje ofrecen a sus clientes un espacio en un servidor público en Internet. A través de estas empresas de hospedaje una compañía o individuo puede hacer pública su página web o sistema. La complejidad de administración, configuración y adquisición de equipos y enlaces queda por parte de estas compañías de hospedaje y sus clientes sólo deben pagar tarifas mensuales, anuales o por uso.

Muchos de estos clientes optan por alquilar servidores de uso exclusivo. Hechos a la medida y con administración por parte de la empresa de hospedaje o el mismo cliente. Pero en gran parte de los casos, estamos hablando de un servidor compartido.

Otro de los servicios que ofrecen es el de hospedaje compartido. Consiste en ofrecer diferentes espacios en un mismo servidor para varios clientes. Los clientes son separados lógicamente dentro del servidor, pero físicamente sus datos están alojados en mismos recursos del servidor. A veces comparten el mismo servidor web pero con diferente documento raíz (www o wwwroot). También el mismo servidor de base de datos pero con diferentes esquemas de usuario.

El problema con los hospedajes compartidos es que promueven riesgo de seguridad que no existen cuando se esta en un ambiente de servidor dedicado. Compañías con requerimientos de confidencialidad y privacidad muy elevados no deberían considerar esta opción. Casos como el sector bancario o el sector gobierno ya que manejan información muy delicada. Sino se protegen bien los recursos de las aplicaciones, como códigos fuentes, ejecutables, archivos de configuración, librerías y la información privada, se pueden exponer gran cantidad de vulnerabilidades, y con ellas poner en riesgo la seguridad de todo el sistema.

Una de las preocupaciones más importantes en este tema, es el de resguardar las credenciales de acceso a la base de datos. Generalmente se encuentran en archivos de propiedades, archivos .inc, .properties, xml, entre otros. Este tipo de archivos se pueden ver expuestos, con facilidad, en un ambiente de hosting compartido que no considere configuración apropiada de privilegios de usuarios y aplicaciones.

Para un usuario en un equipo compartido, es poco probable que tenga acceso a archivos fuera de su directorio personal. Seguramente no podrá acceder al directorio raíz del sistema o el directorio personal de otros usuarios. Pero con un scripts en el servidor web, esto le pudiera ser posible. En este tipo de entornos se recomienda definir estrictos permisos sobre el sistema de archivos y en algunos casos separar los servidores web para que corran con diferentes usuarios y permisos.

Vulnerabilidades:

Débil autenticación, débil autorización, revelación de información, exposición de archivos delicados, fuentes, ejecutables, archivos de configuración, deficiente definición de permisos de usuario, datos delicados no cifrados.

Medidas de seguridad:

Cifrado de los datos físicos (BD, códigos fuentes), establecer permisos estrictos para las aplicaciones y usuarios, asegurar recursos álgidos (códigos fuentes, archivos de configuración).

5.3.3. Administración del sistema en forma remota

Muchos sistemas permiten la administración remota de equipos, específicamente del sistema operativo y sus servicios. Casos como el de compañías que requieren que sus empleados tengan acceso a los servidores desde cualquier parte del mundo o el de compañías que poseen servidores arrendados a terceros que requieren administración. Entre otras cosas la administración remota permite: revisar los registros (logs) del sistema, logs de aplicaciones, instalar o desinstalar aplicaciones, administrar un firewall, iniciar y detener servicios, administración de cuentas de usuario, permisos, etc.

Servicios como SSH y VNC son comúnmente utilizados para este fin. Escritorio remoto de Microsoft Windows también es usado ampliamente; incluso existen aplicaciones web que permiten administrar los recursos en un servidor. Estas aplicaciones de administración generalmente las proveen las compañías de hosting a sus clientes, y ellas se pueden configurar y administrar dominios (Administrar servidores DNS), cuentas FTP, servidores de bases de datos, instalar y desinstalar aplicaciones o módulos, cambiar valores de configuración de los servidores, etc.

Figura 5.2: Implementación de una VNC (Elaboración propia)

Sino se toman medidas de seguridad necesarias se puede comprometer todo el sistema, sus aplicaciones y la información en el mismo.

Vulnerabilidades:

Deficiente autenticación y autorización, configuraciones por defecto, versiones viejas, falta de actualización o parches, deficiencia en canales de comunicación.

Medidas de seguridad:

Revisar y aplicar actualizaciones y parches, configuración apropiada (blindaje), establecer canales de comunicación seguros (VPN, SSL), Cortafuegos (firewall) (lista blanca), filtro por direcciones IP.

5.3.4. Actualizaciones remotas

Muchas veces el mecanismo utilizado para este propósito es mediante transferencia de archivos vía FTP. Se observa sobre todo en casos en que la aplicación esta alojada en un servidor de una compañía de hospedaje. Y en ocasiones, por la naturaleza de la aplicación, es crítico proteger los recursos que se transfieren desde y hacia el servidor.

El protocolo FTP es un protocolo en texto plano. Por ejemplo si se realiza una actualización a través de una red pública, fácilmente se puede estar exponiendo información importante de la aplicación, tales como archivos fuentes, registros de logs, archivos de configuración, etc.

Es muy importante establecer un canal seguro para la actualización de archivos. Protocolos como SFTP y SSH son una opción más segura para este tipo de acciones. También se pueden establecer canales seguros, debidamente autenticados y autorizados, por ejemplo configurando una VPN.

Vulnerabilidades:

Débil autenticación, débil autorización, revelación de información, canal de comunicación inseguro, fuerza bruta.

Medidas de seguridad:

Uso de protocolos seguros (SSH, SFTP), establecer canales seguros (VPN), cortafuegos (firewall), filtrado de direcciones IP, políticas de contraseña.

5.3.5. Actualizaciones frecuentes

Cuando se tiene una nueva versión de una aplicación, bien sea para incorporar nuevas funcionalidades, mejoras o corregir errores, se deben actualizar físicamente los archivos y componentes de la aplicación. Actualizar una aplicación web no siempre es un proceso sencillo. Antes de que una nueva versión pase a reemplazar la actual, se debe garantizar que cumple con todos los estándares de calidad establecidos. Esto incluye verificar el correcto funcionamiento y seguridad de todas sus funcionalidades en el ambiente de producción. Para evitar errores por omisión o descuido es conveniente definir protocolos que aseguren la correcta realización de la operación de actualización. Una buena práctica es definir procesos de calidad de servicios, también conocidos como procesos de QA (Quality Assurance).

Vulnerabilidades:

Accidentes, revelación de información, autenticación insuficiente.

Medidas de seguridad:

establecer procesos de QA.

5.3.6. Servidores de aplicación o servidores web

Son las aplicaciones responsables de recibir y responder las peticiones http de los clientes. Ejecutar las aplicaciones que en ellos se encuentran y administrar sus recursos. Los servidores de aplicación se distinguen de los servidores web por su uso amplio de contenido dinámico del lado del servidor y fuerte integración con base de datos. Adicionalmente los servidores de aplicaciones proveen componentes de integración, mecanismo de comunicación entre múltiples aplicaciones, para ejecutar la parte de la lógica de negocio y acceso a datos. Algunos servidores de aplicación proveen API, lo que los hace independientes del sistema operativo. Las dos grandes alternativas de servidores de aplicación son los Java application servers y .NET Framework servers de Microsoft.

Estos servicios poseen módulos de administración, así como archivos de configuración. A través de los módulos de administración se pueden agregar, modificar y eliminar las aplicaciones que en ellos se ejecutan, administrar credenciales de accesos, se habilitan y deshabilitan funcionalidades, etc. Las configuraciones por defecto, permitir el acceso remoto a través de un canal no seguro, son ejemplos de vulnerabilidades que afectan este tipo de aplicaciones.

En los archivos de configuración se definen los parámetros de ejecución de estas aplicaciones. Los más relevantes son las credenciales de ejecución, los directorios en el sistema de archivos a los cuales tienen acceso, los puertos en los que se ejecutan y configuraciones como proxy o balanceo de carga.

No es deseable tener un servidor web público, ejecutándose con privilegios de súper usuario en el sistema. En caso de que el mismo sea comprometido, puede ocasionar daños inimaginables ya que no posee restricción alguna sobre el sistema. Por ejemplo si a través del servidor un atacante consigue ejecutar un comando arbitrario al sistema operativo.

Vulnerabilidades:

Abuso de funcionalidad, débil autenticación, configuración por defecto, versiones viejas, falta de actualizaciones o parches, usurpación.

Medidas de seguridad:

Cortafuegos (firewall), filtrado de direcciones IP, autenticación, revisar y aplicar actualizaciones y parches, configuración apropiada (blindaje).

5.3.7. Servidores de correo

Un servidor de correo se encarga del envió y recepción de mensajes a través del protocolo SMTP. Las aplicaciones web utilizan servidores de correo para enviar mensajes a sus clientes. Por ejemplo para envío de contraseñas, facturas, boletines entre otros. Si estos servidores no están debidamente configurados, no requieren autenticación y son alcanzables desde la red pública, los mismos pueden ser objeto de ataques como DoS, abuso de funcionalidad, usurpación, etc.

Vulnerabilidades:

Abuso de funcionalidad, débil autenticación, configuración por defecto, versiones viejas, falta de actualizaciones o parches, usurpación.

Medidas de seguridad:

Cortafuegos (firewall), filtrado de direcciones IP, autenticación, revisar y aplicar actualizaciones y parches, configuración apropiada (blindaje).

5.3.8. Servidores de base de datos

Los servidores de bases de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos y las aplicaciones que la utilizan. En ellos pueden existir gran

cantidad de bases de datos y diferentes cuentas de usuarios con sus permisos asociados. Una configuración común en estos servidores consiste en que tanto el servidor web como el servidor de base de datos residen en la misma máquina. Por tanto el acceso al servidor web puede estar restringido única y exclusivamente mediante una conexión localhost. El servidor de bases de datos no acepta conexiones desde máquinas remotas. También ofrecen opciones de configuración para que solamente determinados usuarios se puedan conectar desde una máquina X con IP (Filtro de lista blanca por direcciones IP). Permitir conexiones remotas a este tipo de servicio sin justificación, puede exponer al sistema a vulnerabilidades y amenazas.

Los servidores de bases de datos por lo general guardan datos muy importantes para las compañías. En ellos puede existir data de clientes, tarjetas de crédito, histórico de operaciones, datos de cuentas bancarias, información histórica del seguro social de las personas, etc. Es importante configurarlos correctamente y proteger la data que en ellos existe.

En algunos casos donde los niveles de seguridad que se requieren son "de paranoia", se protegen, mediante algún algoritmo de cifrado, los datos físicos del servidor de base de datos y las bases de datos. Esto previene que se pueda restaurar y obtener los datos con facilidad. Imagine todo lo que pudiera hacer un atacante al acceder a una base de datos con información bancaria de personas en el mundo, sin que los datos ahí contenidos estén como mínimo cifrados.

A pesar de todos los controles y medidas de seguridad, existen amenazas que van más allá de cualquier control y que pueden resultar en una catástrofe para los datos importantes de una empresa. Los desastres naturales son un ejemplo de este tipo de amenazas y sino no se toman medidas de prevención como políticas de respaldo oportunas, las consecuencias pueden ser irreversibles. Otro tipo de accidente es el borrado accidental de los datos, por un usuario correctamente autenticado y con los permisos adecuados.

Vulnerabilidades:

Abuso de funcionalidad, débil autenticación, configuración por defecto, versiones viejas, falta de actualizaciones o parches, deficiente respaldo, permisología inadecuada, canal de comunicación no seguro, exposición de recursos innecesaria.

Medidas de seguridad:

Cortafuegos (firewall), configuración de una DMZ, filtrado de direcciones IP, autenticación, revisar y aplicar actualizaciones y parches, configuración apropiada (Blindaje), permisos de cuentas de usuario.

5.3.9. Servidores FTP

Es el mecanismo por excelencia para la transferencia de archivos al servidor. El servicio FTP utiliza normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el

intercambio de información, desde el credenciales de acceso del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante lo tiene muy fácil para capturar este tráfico, acceder al servidor, o apropiarse de los archivos transferidos.

Para solucionar este problema son de gran utilidad aplicaciones como scp y sftp, incluidas en el paquete SSH, que permiten transferir archivos pero cifrando todo el tráfico.

Vulnerabilidades:

Abuso de funcionalidad, débil autenticación, configuración por defecto, versiones viejas, falta de actualizaciones o parches, permisología inadecuada, canal de comunicación no seguro, exposición de recursos innecesaria.

Medidas de seguridad:

Cortafuegos (firewall), configuración de una DMZ, filtrado de direcciones IP, autenticación, revisar y aplicar actualizaciones y parches, configuración apropiada (Blindaje), verificación de credenciales y permisos.

5.3.10. Otros servicios

Aparte de servidores web, servidores de bases de datos y correo, las apliaciones web pueden requerir interactuar con otros tipos de servicios. Tales como servidores de autenticación, servidores de mensajeria, servidores LDAP, servidores de directorio, servidores RPC.

Vulnerabilidades:

Abuso de funcionalidad, débil autenticación, configuración por defecto, versiones viejas, falta de actualizaciones o parches, permisología inadecuada, canal de comunicación no seguro, exposición de recursos innecesaria.

Medidas de seguridad:

Cortafuegos (firewall), configuración de una DMZ, filtrado de direcciones IP, autenticación, revisar y aplicar actualizaciones y parches, configuración apropiada (Blindaje), verificación de credenciales y permisos.

5.3.11. Tecnologías del lado del servidor para contenido dinámico

Son tecnologías del servidor de web en las cuales las peticiones de un usuario son completadas, luego de un procesamiento realizado directamente en el servidor de web para generar la respues-

ta dinámicamente. Por lo general son usados para permitir interacción en las páginas web con sistemas manejadores de bases de datos u otros servicios. Son diferentes a tecnologías del lado del cliente, donde la ejecución es controlada por el navegador de web. Por lo general JavaScript. Ofrece capacidades para personalizar la respuesta a los usuarios, llevar seguimientos de sus peticiones, realizar operaciones de búsqueda, agregar y modificar registros, interactuar con servidores de correos entre otros.

Tecnologías del lado del servidor ampliamente utilizadas son: PHP, ASP, JSP, Ruby, Java, C, etc.

5.4. Capa de Aplicación

5.4.1. Inicio de sesión de usuario vía formulario HTML

Actualmente es la forma más utilizada en el entorno web para autenticar usuarios en los sistemas. Existen otras formas como el uso de certificados digitales instalados en la máquina cliente, pero en lugar de utilizar estos mecanismos por sí solos, se usa en combinación con la técnica de autenticación a través de formularios web.

Figura 5.3: Un cliente de correo web, en este caso google mail o gmail (Elaboración propia)

Esta técnica consiste en proveer a través de un formulario web, como mínimo, dos campos de texto. Un campo para identificar la cuenta o el usuario (puede ser una cadena de texto alfanumérica, nombre de usuario, una dirección de correo electrónico, un numero de cuenta o tarjeta, etc.) y una clave o contraseña secreta (combinación de letras, números y/o caracteres especiales).

Figura 5.4: Una aplicación de banca electrónica (banco Banesco) (Elaboración propia)

Vulnerabilidades:

Ataques de fuerza bruta, revelación de información, inyección de código o comandos, phishing, denegación de servicios.

Medidas:

Uso de captcha, control de ataques de fuerza bruta, filtrar la entrada de datos, eficiente almacenamiento de la información, garantizar la mejor forma de utilización de la información.

5.4.2. Recuperación de contraseña

Los usuarios navegan en múltiples páginas web y es común que en cada una de estas páginas tengan distintos nombres de usuario y contraseña para poder acceder a las mismas. Se recomienda a los usuarios que las credenciales de acceso sean memorizadas y nunca anotadas físicamente en papel o almacenadas en documentos de texto o correos electrónicos. Sin embargo la memoria puede fallar y el usuario simplemente puede olvidar su contraseña o clave secreta y en muchos casos hasta el nombre de usuario o login.

Es común en las aplicaciones web ofrecer mecanismos automatizados para recuperar la clave secreta o contraseña de una cuenta de usuario en caso de olvido. Y existen muchas implementaciones diferentes para este fin. Las más comunes exigen a los usuarios al momento de crear su cuenta, especificar una dirección de correo, la cual servirá como medio para la recuperación de la clave anterior o entrega de una nueva clave. A veces los usuarios, en el registro u administración de su cuenta, deben definir un desafió tipo pregunta y respuesta secreta, la cual debe ser contestada previo al envió de su clave secreta. Otras implementaciones pueden exigir información específica del usuario como número de licencia de conducir, fecha de nacimiento, numero de libreta, etc.

En la siguiente imagen se muestra la aplicación de banca electrónica del Banco de Venezuela en la sección en la cual se permite recuperar la contraseña del usuario.

Figura 5.5: Recordar contraseña de la aplicación del Banco de Venezuela (Elaboración propia)

En estos sistemas de recuperación de clave no es conveniente de una vez entregar credenciales de acceso al instante, es bueno siempre enviar la clave a una dirección de correo especificada por el usuario. En caso de los sistemas denominados web mail en donde la cuenta de usuario que se quiere recuperar es el email en sí, seria conveniente especificar otra dirección de correo alternativa. Esto agrega otra capa de seguridad una vez se haya contestado con la respuesta correcta al desafió, teniendo también que conocer las credenciales de acceso al email alternativo de la víctima.

Hay que tomar en cuenta que responder a la pregunta secreta de un usuario puede resultar menos complicado que adivinar la clave secreta, ya que la pregunta secreta en sí, actúa como una pista o indicio, que puede servir para acotar por ejemplo un ataque de fuerza bruta.

Por ejemplo, si la pregunta a la respuesta secreta fuese “ciudad de nacimiento?”, un ataque de fuerza bruta, con un diccionario de palabras de las ciudades del mundo, no tardaría mucho en tener éxito.

Vulnerabilidades:

Ataques de fuerza bruta, revelación de información, inyección de código o comandos, phishing, abuso de funcionalidad, expiración de sesión insuficiente, débil validación en la recuperación de contraseñas.

Medidas:

Uso de captcha, control de ataques de fuerza bruta, filtrar la entrada de datos, eficiente almacenamiento de la información, garantizar la mejor forma de utilización de la información.

5.4.3. Administradores remotos

Muchos sitios web poseen administradores remotos. Estos administradores remotos son aplicaciones o parte de una aplicación que permite a una persona autorizada realizar cambios o modificaciones en el sitio web. Por ejemplo publicación de contenido como noticias, galerías, productos, aprobar transacciones, ver reportes, ver y modificar información de clientes, etc.

Estas herramientas son muy útiles para la administración de una pagina web, sin embargo, también pueden ser muy útiles para un atacante si consigue tener acceso a la misma.

Este tipo de aplicaciones de administración son objeto de ataques que tienen por objetivo vulnerar la autenticación y autorización del sistema. Estos sistemas deben contar con un buen sistema de autenticación. Si estas aplicaciones son accedidas a través URLs públicos, es un error pensar que serán inaccesibles a un atacante por el simple hecho de no aparecer públicamente como un enlace en el site, o porque solamente conoce la dirección y/o el puerto la persona administradora. El ejemplo típico es el usuario de URLs como www.mipagina.com/admin o www.mipagina.com/panel.

El otro gran flanco esta relacionado con la autorización, que consiste en verificar si en efecto un usuario tiene el permiso para ejecutar determinada acción. A veces un atacante no necesita usurpar la identidad de un usuario para cometer un ataque. Simplemente puede inducir a un usuario debidamente autenticado a cometer acciones sin que este se percate de lo que esta ocurriendo, mediante técnicas de XSS o XSRF.

Otro caso a considerar es, si sólo se autentica el usuario para acceder al panel de administración, pero no se lleva seguimiento de la sesión del usuario en cada página o acción dentro del administrador, pudiera permitir a un atacante ejecutar directamente una acción sin necesidad de autenticarse en la página.

Figura 5.6: Panel de administración remota de la página de bebek (Elaboración propia)

Vulnerabilidades:

Autenticación insuficiente, autorización insuficiente, robo de sesión, XSS, XSRF, fijación de sesión, revelación de información, localización de recursos predecibles, ábuso de funcionalidad.

Medidas:

Verificar la autenticación de un usuario en todo momento, establecer controles de sesión en procesos críticos, filtrar la entrada de datos, control de ataques de fuerza bruta.

Estas funcionalidades pueden ser víctimas de:

5.4.4. Registro de usuarios en línea

Hay páginas únicamente informativas o de consulta, en donde los distintos usuarios pueden pasar desapercibidos en cada visita. Por otra parte hay páginas en donde los usuarios son un

elemento activo dentro de la misma. Donde se quiere llevar registro de cada usuario y a su vez ofrecerle al usuario la forma de manejar su cuenta personal dentro de la página web. O simplemente la información de los usuarios es utilizada para llenar una base de datos que es utilizada para el envío de boletines informativos, propagandas, concursos, servicios, etc.

Para ello muchas páginas utilizan sistemas de registros en línea. Generalmente a través de formularios web que recogen información personal y de contacto de los usuarios. El tener muchos usuarios registrados en un sitio web se ha convertido en sinónimo de estatus en el Internet. Pero para que sea efectivo se requiere garantizar como mínimo que se trate de registros únicos y no de un mismo usuario con varios registros, registros reales y no de personas ficticias o datos corruptos.

Figura 5.7: Registro de usuarios en línea de la página 123.com.ve (Elaboración propia)

Una buena práctica es confirmar registros con direcciones de correo existentes, y siempre validar que no se pueda registrar más de un usuario con la mismas direcciones de correo. Evitar registros automatizados con el uso de captchas y de esta forma prevenir un abuso de funcionalidad. Así como la respectiva validación y filtrado de datos de entrada para evitar cualquier tipo de data corrupta o incompleta y/o la inyección de código no deseado dentro de la aplicación.

Vulnerabilidades:

revelación de información, abuso de funcionalidad, inyección de código SQL, phishing.

Medidas:

Filtrar la entrada de datos, eficiente almacenamiento de la información, garantizar la mejor forma de utilización de la información, establecer controles de sesión en procesos críticos, uso de captcha.

5.4.5. Usuarios mantienen sesión

Los usuarios son identificados en el sistema y una vez autenticados pueden tener acceso a información personal para simplemente verla y en el peor de los casos modificarla y hasta eliminarla, pueden realizar operaciones, compras en línea, publicar contenido entre otras cosas más. Los

usuarios pasan por un proceso de registro, como se explico anteriormente, generalmente a través de formularios web, en otros casos se trata de usuarios creados fuera de línea, por ejemplo entidades bancarias ofreciendo servicios de banca electrónica a clientes existentes.

Una vez que los usuarios se autentican ante un sitio web, se crea una relación de confianza entre el usuario y el sitio. Todas las actividades y operaciones realizadas durante esa sesión, son adjudicadas a esa cuenta de usuario que identifica a un comercio o persona. Por ejemplo: la compra de un producto, una transferencia electrónica, realizar un comentario o post, enviar un email entre otras.

Vulnerabilidades:

Predicción de credenciales/sesión, robo de sesión (session hijacking), fijación de sesión (session fixation), XSS, XSRF.

Medidas:

Establecer controles de sesión en procesos críticos, filtrar la entrada de datos.

5.4.6. Usuarios realizan operaciones

Las características que permiten a los usuarios realizar operaciones, son todas aquellas entradas del usuario donde de forma indirecta o directa interviene en el contenido de la página y/o lógica de negocio del sistema.

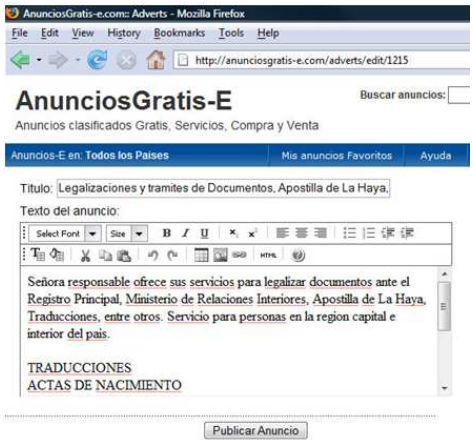


Figura 5.8: Imagen de publicación de un anuncio (Elaboración propia)

A diferencias de paginas absolutamente informativas, donde los usuarios solamente pueden consultar información, este tipo de características le ofrecen la posibilidad al usuario de tener presencia en la página, a través de foros, blogs, reviews, comentarios, etc. De igual forma pueden generar entradas en la lógica de negocio del sistema mediante operaciones de compra, realización de apuestas, realización de transferencia de saldo, envío de un correo electrónico, etc.

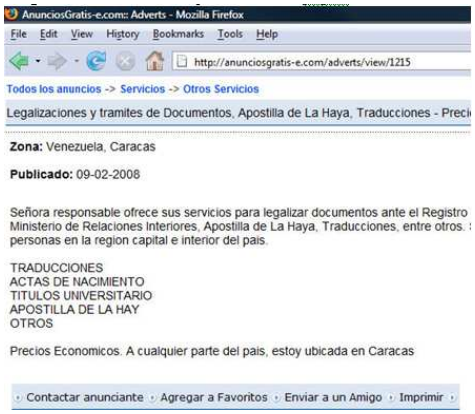


Figura 5.9: Imagen de vista del anuncio publicado en la pagina web (Elaboración propia)

Vulnerabilidades:

Inyección de código y comandos, XSS, fijación de sesión, suplantación de contenido.

Medidas:

Filtrar la entrada de datos, escapar la salida de datos.

5.4.7. Sitios web que se comunican con usuarios

Muchos sitios web cuentan con usuarios web registrados. En la mayoría de los casos, aparte de información de identificación, suelen tener información para contactarlos, tales como teléfono, celular y dirección. Pero el favorito en este medio de Internet es sin lugar a dudas, el correo electrónico. A través de este los sitios web pueden comunicarse con sus usuarios y enviarles información acerca de promociones y servicios, notificaciones de operaciones realizadas e incluso enviar información personal y/o confidencial a los mismos.

Los atacantes aprovechándose de esta característica, en ocasiones, buscan confundir a los usuarios, suplantando la identidad de su sitio web de confianza, y solicitando información personal de acceso a sus cuentas de usuario. Esta técnica es conocida como phishing y se ha convertido en una estafa muy popular entre los atacantes para comprometer usuarios de entidades bancarias y/o instituciones afines. En general cualquier aplicación, o mejor dicho, cualquier entidad o empresa son susceptibles a un ataque de phishing.

A pesar de las medidas de seguridad que se puedan tomar en todos los sistemas de contacto y mensajería hacia el usuario, simplemente no se puede evitar que un atacante envié este tipo de correos a través de sistemas externos a la aplicación con direcciones parecidas a la de un sitio auténtico. El no caer en este tipo de estafa depende mucho del conocimiento y sentido común del usuario (en este caso la víctima). Por tanto se han tomado medidas como educar a los usuarios en esta materia, y darles a conocer de alguna forma el tipo de mensajes que pueden recibir o no de su sitio web de confianza.

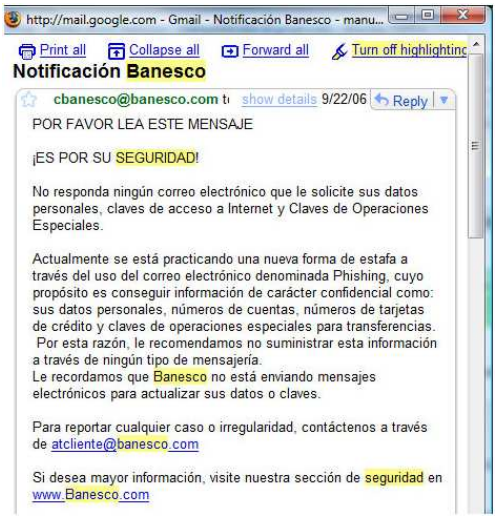


Figura 5.10: Email enviado por banesco a sus usuarios (Elaboración propia)

Cabe destacar que este tipo de información de contacto, en la mayoría de los casos, es también considerada información privada. Por tanto se le debe dar el tratamiento adecuado tanto en el tráfico, como en el flujo y almacenamiento de dicha información.

Vulnerabilidades:

Phising, revelación de información, XSS, XSRF.

Medidas:

Filtrar la entrada de datos, escapar la salida de datos, garantizar la mejor forma de utilización de la información.

5.4.8. Pase de parámetros accesibles al usuario

Básicamente toda aplicación web existente implementa esta característica de alguna forma. Puede parecer tonto, pero en ocasiones, se intenta proteger recursos mediante un mecanismo de oscuridad u ocultación. Por ejemplo ocultar enlaces, en menúes que solamente están activos o se visualizan al ser administrador, sin embargo, los accesos a dichas URL existen y si no se tienen un buen control de autorización, a través de un mecanismo de sesión para acceso a los mismos, puede provocarse un acceso no autorizado a contenido privilegiado.
Si un desarrollador expone, mediante algún parámetro de URL o formulario, referencias directas a objetos internos de la aplicación sin que haya un respectivo chequeo a nivel de aplicación, un atacante puede cometer un ataque de referencia directa de objeto.
Por ejemplo en una aplicación bancaria es común usar números de cuentas bancarias como claves primarias. Es tentador usar el número de cuenta en la interfaz de usuario para referenciar el objeto. Aún si el desarrollador implemento debidamente medidas de filtrado y correcta parametrización de SQL previendo algún tipo de ataque de SQL injection, si no se hacen chequeos adicionales que determinen que la cuenta es del usuario y que el mismo tiene autorización para verla, un atacante puede jugar con el parámetro numero de cuenta y así ver la información de todas las cuentas.

Este tipo de ataque ocurrió en Australia, en el sitio web de una oficina de impuesto llamada GST Start Up Assistance , en el año 2000. Donde un usuario legitimo pero hostil, simplemente cambiando el ABN (un numero de id de impuesto en ese país) presente en la URL, pudo obtener información financiera de aproximadamente 17000 empresas y las envío por correo a cada empresa, resultando en un embarazoso agravio para el gobierno de estas empresas.

Este tipo de vulnerabilidad es muy común en aplicaciones web y es muy poco probado por desarrolladores en la actualidad.

Ejemplo de algunas URL en las que se puede probar cambiando parámetros en la url y cambiar valores para tener acceso a diferentes objetos dentro de la aplicación.

http://anunciosgratis-e.com/adverts/view/1215, es de una pagina de anuncios y nos muestra el anuncio con id 1215 que es un perfecto y legitimo anuncio.

Cambiando el valor del view por valores bajos como 1, 2, se puede tener acceso a los primeros anuncios publicados, y los mismos resultan ser anuncios de pruebas que olvidaron borrar físicamente de la base de datos una vez puesta en producción. Muchas veces estos datos o cuentas de pruebas contienen información real, que puede develar información importante a un posible atacante.

http://anunciosgratis-e.com/adverts/view/2

y removiendo el numero de id, resulta en un error de ejecución código 500, el cual deja ver perfectamente la excepción ocurrida, en que línea, y con cual función.

Figura 5.11: Error al cambiar manualmente un valor en la URL (Elaboración propia)

Vulnerabilidades:

revelación de información, referencia directa de objetos, autorización insuficiente, inyección de código, XSS, XSRF, subida de directorios (path traversal).

Medidas:

filtrar la entrada de datos, escapar la salida de datos, establecer controles de sesión en procesos críticos.

5.4.9. Manipulan información sensible

Garantizar la confidencialidad, integridad y disponibilidad de la información es uno de los temas más delicados en materia de seguridad. Para conseguir esto se debe proteger la información de accesos no autorizados para uso, exposición, perturbación, copia, modificación o destrucción de la misma. Proteger la información confidencial es un requerimiento estricto de todo negocio, y en algunos casos es un requerimiento ético y legal.

Los gobiernos, instituciones financieras, hospitales, y negocios privados almacenan mucha información confidencial sobre sus empleados, clientes, productos, investigación, y su estado financiero. En algunas aplicaciones web, gran parte de esta información sensible es procesada y almacenada. Por tanto las mismas deben garantizar que dicha información este segura en todo momento, en el trafico, procesamiento y almacenamiento de la misma en el sistema.

Vulnerabilidades:

Inyección de código y comandos, fuga de información.

Medidas:

Filtrar la entrada de datos, escapar la salida de datos, eficiente almacenamiento de la información, garantizar la mejor forma de utilización de la información.

5.4.10. Uso de módulos o aplicaciones de terceros del lado del cliente

Muchas aplicaciones web incluyen módulos de software de terceros, para realizar alguna funcionalidad adicional o incorporar mejoras en los procesos existentes. Estos módulos pueden encontrarse tanto en tecnología del lado del cliente como en tecnología del lado del servidor.

Para el caso de tecnología del lado del cliente, se presentan los complementos o plugins del navegador. Estos permiten entre otras cosas, abrir formatos no conocidos por defecto para el navegador y ejecutar programas del lado del cliente para cambiar la apariencia de la página o incluso realizar actualizaciones y “mejoras” en el navegador. Algunos ejemplos de estos complementos son los reproductores de videos Windows Media Player y QuickTime, Adobe Acrobar Reader para leer documentos en formato PDF, el plugin de Flash para ejecutar multimedios interactivos con capacidades web, entre otros.

Figura 5.12: Uso de plugin de flash en una pagina web (Elaboración propia)

La mayoría de las amenazas para este tipo de software son producto de uso de versiones viejas o sin parchear. En Internet y a través de listas de correo, se consiguen detalles de las vulnerabilidades de software más conocidas y frecuentes para cada uno de estos complementos. El usuario puede protegerse instalando las últimas actualizaciones y parches directamente del fabricante de dichos módulos.

Los desarrolladores de aplicaciones no poseen una responsabilidad directa, para controlar o corregir las vulnerabilidades y amenazas de seguridad en el uso de estos complementos en el cliente, ya que es responsabilidad de terceros. Sin embargo, se pueden tomar medidas como educar y concienciar al usuario, de la existencia de los riesgos e indicarle como protegerse. En algunos casos se pueden establecer restricciones para que sólo se pueda servir y ejecutar los contenidos, relacionados a estos complementos, si el cliente cuenta con la versión más actualizada de los mismos.

Los ActiveX de Microsoft, también se encuentran en esta categoría de módulos o aplicaciones de terceros del lado del cliente. Son pequeños programas que se pueden incluir dentro de páginas web y sirven para realizar acciones de diversa índole. Por ejemplo hay controles ActiveX para mostrar un calendario, para implementar un sistema de FTP, etc. Son exclusivos para Internet Explorer y bajo plataforma Windows.

Los ActiveX, son un poco parecidos a los Applets de Java en su funcionamiento, aunque una diferencia fundamental es la seguridad, pues un Applet de Java no podrá tomar privilegios para realizar acciones malignas (como borrarnos el disco duro), ya que se ejecutan dentro de una máquina virtual de Java, en cambio a los controles ActiveX sí se le pueden otorgar permisos para hacer cualquier cosa.

La recomendación es evitar la instalación y ejecución de los ActiveX, sobre todo si no son provenientes del sitio oficial de Microsoft.

Vulnerabilidades:

Versiones viejas o sin parchear, deficiente autorización, accesos no autorizados.

Medidas:

Revisar y aplicar actualizaciones y parches, configuración apropiada (blindaje).

5.4.11. Uso de módulos o aplicaciones de terceros del lado del servidor

Las aplicaciones web, utilizan también complementos o módulos de terceros del lado del servidor. Estos complementos pueden ser librerías o componentes de software que implementan funcionalidades y aspectos de programación para la lógica de negocios, el acceso a datos, el flujo de datos y presentación o interfaces de usuario. Pueden ser piezas individuales de software con una función bien específica o suites completas con bibliotecas e incluso lineamientos de una arquitectura de software. Estos últimos se conocen como Frameworks. También pueden ser otras aplicaciones, incluidas como funcionalidades dentro de una aplicación más grande. Por ejemplo muchas páginas web incluyen sistemas de foros o RSS desarrollados por terceros.

Los Framework son considerados plugins o complementos. En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de programación, entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Algunos frameworks ampliamente utilizados en diferentes aplicaciones web son Struts, Spring, .NET Framework, Cakephp, Ruby on Rails, entre otros. La mayoría de estos frameworks incluyen bibliotecas para seguridad en aplicaciones web, y definen lineamientos de cómo se deben implementar las funcionalidades y procesos de una aplicación web, en forma segura. Por ejemplo ofrecen funcionalidades como, bibliotecas de acceso a datos que minimizan los riesgos de inyección de SQL, funciones para el filtrado de entradas, funciones para escapado de datos de salida y funciones para cifrado de datos.

Es importante que los desarrolladores hagan uso de las herramientas que ofrecen estos frameworks, y que lo hagan de forma correcta. Y en caso de que el framework utilizado no contenga una funcionalidad necesaria para la seguridad de la aplicación, se debe implementar la misma.

Los frameworks pueden presentar Bugs y con el tiempo se le pueden descubrir vulnerabilidades. Se deben actualizar o parchear según sea necesario. Y nunca se deben usar para ambientes de produccion, versiones Betas o inestables.

Vulnerabilidades:

Versiones viejas o sin parchear, deficiente autorización, accesos no autorizados.

Medidas:

Revisar y aplicar actualizaciones y parches, configuración apropiada (blindaje).

Capítulo 6

Metodología

Se define una metodología para la auditoría de seguridad en aplicaciones web. Basada en las principales características de las aplicaciones web, y dividida en las distintas capas propuestas en este trabajo especial de grado: capa física, capa de red, capa de sistema operativo y servicios, y capa de aplicación. La separación por capas permite identificar, durante el análisis y aplicación de la metodología, el área de estudio específica, así como el rol o persona especialista correspondiente en cada paso de la misma. En cada paso de la metodología se busca identificar las amenazas producto de la existencia o no de vulnerabilidades en las características, o por la omisión o deficiencia en cuanto a controles y medidas de seguridad.

Objetivos de la metodología

El objetivo principal de la metodología es determinar las vulnerabilidades y amenazas presentes en una aplicación web en sus diferentes capas, y ofrecer un reporte con las recomendaciones y medidas de seguridad que deben ser tomadas para protegerlas.

Confidencialidad

Los auditores deben mantener la confidencialidad de la información obtenida durante el curso del trabajo de auditoría, asegurándose que el acceso a la información esta controlado apropiadamente y que sólo será distribuido a personas autorizadas.

Herramientas y conocimientos necesarios

Es indispensable tener conocimientos en el área de aplicaciones web, para la correcta aplicación de esta metodología. Sus temas exigen un mínimo de comprensión en aspectos como plataforma, redes, sistemas operativos, servidores web, base de datos, programación, tecnologías del lado del servidor y del lado del cliente. Específicamente debe saber para qué sirven cada una de estas tecnologías y como se relacionan entre ellas, en el universo de las aplicaciones web.

Otro aspecto importante para la correcta aplicación de esta metodología, es el uso de herramientas o técnicas que ayuden a tener una mejor comprensión y análisis de los elementos en una aplicación web. El uso de Diagramas de Flujo para describir detalles algorítmicos de un proceso, realizar Diagramas UML (Lenguaje Unificado de Modelado) a conveniencia que ayuden a visualizar, especificar y documentar partes del sistema de software (por ejemplo diagramas de clases, diagramas de casos de uso, diagramas de secuencia), modelos de entidad relación (ER) para describir tablas y objetos en una base de datos, entre otros.

El uso de herramientas de software puede agilizar en gran medida el proceso de auditoría. Un escaneo de puertos rápidos con nmap puede revelar un servidor de escritorio remoto no protegido o un servidor de base de datos accesible desde la red pública sin contraseña. Un escáner de vulnerabilidades automatizado como NESSUS o RETINA puede ayudar a detectar muchas vulnerabilidades de los equipos en la red. Son ejemplos de herramientas en las que se puede apoyar el auditor.

Se insta al auditor a valerse de estas herramientas durante la aplicación de la metodología, pero no se es especifico con respecto a la estricta inclusión o uso de alguna de estas herramientas en los pasos de la metodología, ya que son una ayuda para el auditor y no un paso obligatorio. El auditor puede usar las herramientas que desee, incluso él puede desarrollar herramientas propias y hacer uso de ellas para realizar, de forma automática o más precisa, alguna de las verificaciones de vulnerabilidades en esta metodología.

Aspectos fundamentales

Pre-requisitos o condiciones para poder aplicar la metodología. Estos aspectos deben tenerse en cuenta durante todo el proceso de aplicación de la metodología y correcciones de errores. Estos postulados son:

La metodología dependerá, de esta forma, de los postulados que el investigador considere como válidos, de aquello que considere objeto de la ciencia y conocimiento científico, pues será a través de la acción metodológica como recolecte, ordene y analice la realidad estudiada.

1. Si la aplicación web no es segura, entonces toda la información sensible está en grave peligro.
2. Los sitios web y sus aplicaciones web relacionadas deben estar disponibles 24 x 7 para proveer un buen servicio a sus clientes, empleados y proveedores.
3. Los Cortafuegos y SSL no proveen protección contra vulnerabilidades o amenazas de aplicaciones.
4. Los hackers prefieren tener acceso a data sensible ya que pueden vender esta información por grandes sumas de dinero.
5. Ocultar objetos no garantiza su seguridad.

Etapas de la metodología

La metodología se divide en 4 etapas.

Planificación: La etapa de planificación y recolección de información esta diseñada para permitir a los autores realizar un análisis de riesgo y desarrollar un esquema de planificación y enfoque específico para la auditoría.

Ejecución de la auditoría: En esta etapa se ejecutan las entrevistas técnicas junto con las verificaciones de las políticas y procesos, así como los controles existentes.

Comunicación de resultados: Los resultados de la auditoría son presentados con informes que contienen los hallazgos, observaciones y recomendaciones. Incluye las medidas de seguridad necesarias para proteger la aplicación.

Verificación Posterior (Follow-up): Esta etapa establece un comportamiento iterativo en la metodología. Busca vulnerabilidades, amenazas existentes, producto de los nuevos controles y medidas de seguridad implementados a raíz del informe de auditoría. Verifica existencia o no, y deficiencia en las medidas y controles antes mencionadas. Este paso se repite tanta veces como sea necesario.

Etapa 1: Planificación

Comprensión del negocio y de su ambiente

1. Al planificar una auditoría, el auditor de sistemas debe tener una comprensión suficiente del ambiente total que se revisa. Debe incluir una comprensión general de las diversas prácticas comerciales, lógica del negocio, ambiente normativo y políticas de la empresa. Debe conocer a grosso modo aspectos técnicos de los sistemas, tales como, detalles de implementación, ambiente de ejecución, arquitectura de red, arquitectura de software, etc.

2. Los pasos que puede llevar a cabo un auditor para obtener una comprensión del negocio y sus sistemas son: navegar las aplicaciones web del negocio, lectura de material informativo o documentación, entrevistas a gerentes, empleados y de ser posible con desarrolladores, especialistas técnicos y administradores, estudio de los informes sobre normas o reglamentos, revisión de informes de auditorías anteriores.

3. Definir los objetivos y el alcance de la auditoría. No todas las auditorías son iguales. Se deben definir objetivos concretos como sólo identificar las vulnerabilidades de una parte de una aplicación web, o en su totalidad.

Etapa 2:Describir el sistema

En cada paso apoyarse en la documentación existente, pida el nombre de las personas de contacto, realice entrevistas, revise las políticas y procedimientos, y cualquier otra información que pueda ser relevante para la ejecución de la auditoría.

Pasos:

Capa Física

4. Describa lo mejor posible y con el mayor detalle la infraestructura física de los dispositivos y equipos en el sistema, así como las instalaciones físicas, edificio, cuarto de servidores, puertas, etc. Detalle de los equipos servidores, dispositivos de red, cableado, tipo de instalaciones, personal en el sitio, etc. Indique si dicha arquitectura es de administración propia o por parte de terceros. En el caso que sea administrada por terceros, por ejemplo a través de un servicio de hospedaje (hosting), solicitar al proveedor toda la información referente al mismo. Tomar nota de si es un servicio de hosting compartido o de uso exclusivo. Puntos de accesos a las instalaciones, controles de acceso (llaves, claves, cámaras de vigilancia, dispositivos biométricos, vigilantes, registro de novedades), personas autorizadas, vigilantes, otros.

   4.1. ¿Hardware propio de la empresa? Identifique y describa:

      4.1.1. Políticas de acceso

      4.1.2. Personal autorizado

      4.1.3. Instalaciones físicas - Edif., habitación, cuarto de servidores

      4.1.4. Los equipos y dispositivos

      Características en cuanto a:

      a) Servidores
      b) Cables para conexiones
      c) Alimentación (UPS)
      d) otros

   4.2. ¿Hardware arrendado a terceros? (hosting)

      Si, solicitar información a compañía de hospedaje:

      4.2.1. Políticas de acceso

      4.2.2. Personal autorizado

      4.2.3. Instalaciones físicas - Edif., habitación, cuarto de servidores.

      4.2.4. Los equipos y dispositivos.

      Características en cuanto a:

      a) Servidores
      b) Cables para conexiones
      c) Alimentación (UPS)
      d) otros

   4.3. Administración directa en el equipo

4.3.1. Políticas de acceso

4.3.2. Personal autorizado

4.3.3. Dispositivos de entrada y salida (CD/DVD, USB, Disco Extraible)

Capa de Red

5. Describa lo mejor posible y con el mayor detalle la arquitectura de red en donde reside la aplicación. Detalles de configuración de la red, identificar servidores, protocolos de red utilizados, existencia de enlaces privados o públicos (VPN, intranet o enlaces dedicados), presencia de Firewall y/o IDS, existencia o no de una DMZ, etc. Realice diagramas, esquemas y tablas.Indique si dicha arquitectura es de administración propia o por parte de terceros. En el caso que sea administrada por terceros, por ejemplo a través de un servicio de hospedaje (hosting), solicitar al proveedor toda la información referente al mismo. Usar herramientas de escaneo de vulnerabilidades automatizado (Nmap, nessus, retina, otros).

5.1. ¿Administración por terceros? (hosting)

Si, solicitar información a compañía de hospedaje

5.2. Identificar los equipos en la red

5.2.1. Direcciones IP privadas

5.2.2. Direcciones IP públicas

5.2.3. Equipos en la red local

5.2.4. Equipos remotos

5.3. Identificar dispositivos de firewall

5.3.1. Filtrado lista blanca o lista negra

5.3.2. Políticas de seguridad

5.3.3. Segmentación de redes

5.4. IDS/IPS.

5.5. Identificar protocolos de red usados

5.6. Esquematizar la arquitectura de red (hacer uso de técnicas de modelado)

5.6.1. Diagramas UML de componentes especializados para redes

5.6.2. Otros Diagramas

5.7. Servidores de DNS privados y públicos

Capa de Sistema Operativo y Servicios

6. Describa lo mejor posible y con el mayor detalle las características de los equipos en cuanto a, el sistema operativo que usan, el sistemas de archivos y esquemas de permisologías existentes, servicios disponibles, usuarios del sistema, ubicación de los recursos en los sistemas.

6.1. Sistema operativo (Windows, Unix, Solaris...)

6.1.1. Distribución del SO

6.1.2. Versión

6.1.3. Usuarios y Grupos

6.1.4. Usuarios y Grupos administradores (superusuarios)

6.2. Servicios

6.2.1. Servidores de aplicaciones (servidores web)

6.2.2. Servidores de bases de datos

6.2.3. Servidores de correo

6.2.4. Servidores LDAP

6.2.5. Servidores FTP

6.2.6. Otros servicios

6.3. Para cada servicio

6.3.1. Versión

6.3.2. Usuarios y permisos de ejecución

6.3.3. Archivos y directorios relacionados

6.3.4. Puertos que utilizan (puertos de servicio, puertos para administración)

6.3.5. Cuentas de usuarios del servicio

6.3.6. Archivos de configuración, propiedades

6.4. Sistema de archivos

6.4.1. Ubicación y contexto de las aplicaciones

6.4.2. Identifique recursos compartidos (/tmp, /lib)

6.4.3. Identifique directorios protegidos (C:/ProgramFiles, /)

6.4.4. Ruta de instalación de servicios (base de datos, servidor web, etc)

6.5. Administración remota

6.5.1. A través de red interna

6.5.2. VPN

6.5.3. SSH

6.5.4. Escritorios Remotos (VNC)

7. Identifique las aplicaciones o módulos presentes en la aplicación web. Para cada una de estos elementos determinar como mínimo el nombre de aplicación o módulo, versión, ubicación, permisos de ejecución relacionados (si aplica). Algunas tecnologías comúnmente relacionadas con las aplicaciones web son: servidores de aplicaciones o servidores web (Apache, IIS, Tomcat...), servidores de base de datos, tecnologías del lado del servidor (Java, ASP .Net, PHP, Ruby...), entre otras.

7.1. Identificar servidores de aplicaciones

7.1.1. Servidor Web (Geronimo, JBOSS, Tomcat, Apache, IIS...)

7.1.2. Versión

7.1.3. Permisos de ejecución

7.1.4. Accesos y permisos de directorios

7.1.5. Usuarios o cuentas de administración

7.1.6. Módulos instalados o librerías (mod_rewrite, fileupload)

7.2. Servidores de base de datos

7.2.1. Servidor de base de datos (ORACLE, MySQL, SQL server, MS access)

7.2.2. Versión

7.2.3. Ubicación

7.2.4. Permisos de ejecución

7.2.5. Accesos y permisos de directorios

7.2.6. Usuarios o cuentas de administración y usuarios

7.2.7. Acceso

a) Acceso Local

b) Acceso Remoto

7.2.8. Cifrado de archivos físicos

7.3. Servidores de Correo

7.3.1. Versión

7.3.2. Ubicación

7.3.3. Permisos de ejecución

7.4. Servidores LDAP

7.4.1. Versión

7.4.2. Ubicación

7.5. Tecnologías del lado del servidor

7.5.1. Identificar las tecnologías usadas. (Java, PHP, ASP .NET, Ruby)

7.5.2. Versión

7.5.3. Permisos de ejecución

7.5.4. Permisos de directorios

Capa de Aplicación

8. Describa lo mejor posible y con el mayor detalle las tecnologías usadas en las aplicaciones web. Identificar protocolos de comunicación (HTTP, HTTPS, FTP, LDAP...), uso de Frameworks (Struts, Spring, Cakephp, phpNuke, Ruby on Rails...) y los patrones de diseño (Composite view, MVC) que utilizan, tecnologías del lado del cliente (HTML, CSS, Javascript, AJAX, Flash, Plugins como Windows Media Player o Acrobat Reader, entre otros...), archivos y recursos de la aplicación, acceso a datos (servidores de base de datos, SAN, sistemas de archivos), servicios web.

8.1. Identificar protocolos de comunicación

8.1.1. HTTP

8.1.2. HTTPS (HTTP y SSL). Certificados digitales

8.1.3. SMTP

a) ¿Público o Privado?

b) Autenticado ¿Sí o No?

8.1.4. LDAP, FTP

a) ¿Público o Privado?
b) Autenticado ¿Sí o No?

8.1.5. DNS.

8.2. Determinar el uso de uno o más Frameworks. Conocer o investigar el Framework sobre el cual están hechas las aplicaciones ayuda a una mejor comprensión de sus funcionalidades y procesos, desde su estructura y organización de archivos y recursos hasta el código fuente e implementación.

8.2.1. Versión

8.2.2. Librerías y complementos (*.jar, *.dll, Hibernate, Ibatis, etc)

8.2.3. Patrones (MVC, Composite view)

8.2.4. Estructura de la aplicación

a) Ubicación de archivos fuentes
b) Ubicación de archivos de registro (logs)
c) Ubicación de archivos de configuración
d) Ubicación de archivos públicos (paginas, scripts, imágenes, css, multimedia, etc.)
e) Ubicación de archivos subidos al servidor

8.3. Tecnologías del lado del cliente involucradas

8.3.1. HTML (XML, CSS)

8.3.2. Javascript, VBscript. (Son requisitos de la página)

8.3.3. AJAX

8.3.4. Complementos o Plugins (Windows Media Player, Flash,Acrobat Reader)

8.3.5. Complementos propios (.exe, ActiveX, hotbars)

8.3.6. Cookies

8.4. Acceso a datos

8.4.1. Acceso a base de datos

a) Ubicación de los recursos
b) Almacenamiento de credenciales de acceso
c) Conexión segura
d) Cifrado de datos

8.4.2. Acceso a archivos

a) Permisología sobre los recursos
b) Recursos locales
c) Recursos remotos
d) Cifrado de archivos

8.5. Acceso a servicios.

8.5.1. Servicios Web (Web Services)

a) Internos
b) Externos

9. Determinar funcionalidades o características que existen en la aplicación web. Identificando características como registro de usuario, inicio de sesión, uso de certificado digital o no, foros, formas para recuperación de contraseñas, modulo de administración, administradores de contenido, capacidades para subir archivos al servidor, ejecución de comandos, inclusión de archivos en el servidor, accesos a recursos remotos tales como otras aplicaciones, servicios web o recursos como imágenes, hojas de estilos entre otros.

    9.1. En cada paso realizar:

        9.1.1. Diagramas de flujo de las funcionalidades y procesos

        9.1.2. Diagramas de secuencia UML

        9.1.3. Diagramas de clases

    9.2. Identificar procesos de autenticación

        9.2.1. Inicio de sesión con formularios web

        9.2.2. Certificados digitales

        9.2.3. Sistemas de recuperación de contraseña

    9.3. Identificar operaciones. Procesos de compra, transacciones, agregar contenido, registros, etc.

    9.4. Módulos de administración

        9.4.1. Administradores de contenido

        9.4.2. Administradores de consulta y operaciones

    9.5. Procesos con capacidades de:

        9.5.1. Subir archivos al servidor

        9.5.2. Entradas y salidas al usuario (parametros, archivos)

        9.5.3. Envió de mensajes (correos, posts, blogs, foros)

    9.6. Secciones o procesos que usan o exigen uso de javascript

    9.7. Secciones o procesos que usan o exigen uso de complementos

Etapa 2: Verificación y comprobación

10. Verificación y comprobación de seguridad de **Capa Física**:

    10.1. Hardware propio de la empresa

    Verificar existencia o no de vulnerabilidades en cuanto a:

        10.1.1. Accesos no autorizados

        10.1.2. Catástrofes naturales

        10.1.3. Accidentes

        10.1.4. Fallas

        10.1.5. Deficientes políticas de acceso

    Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

        10.1.6. Políticas de acceso estrictas

a) Registro de eventos y control de acceso
b) Guardias de seguridad
c) Libro de novedades
d) Puertas cerradas
e) Llaves o Claves
f) Dispositivos de seguridad biométricos
g) Cámaras

10.1.7. Continuidad del negocio

a) Redundancia
b) Corriente ininterrumpida (UPS)

10.1.8. Respaldo

10.1.9. Detección y Corrección de Fallas

a) Control de temperatura y humedad
b) Detectores y alarmas de: calor,humo y humedad
c) Sistema de extinción de incendios

10.2. Hardware arrendado a terceros

10.2.1. Obtener información de infraestructura y resguardo del mismo por parte tercero (Empresa de hospedaje, Outsourcing).

Verificar existencia o no de vulnerabilidades en cuanto a:

10.2.2. Accesos no autorizados

10.2.3. Catástrofes naturales

10.2.4. Accidentes

10.2.5. Fallas

10.2.6. Deficientes políticas de acceso

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.2.7. Políticas de acceso estrictas

a) Registro de eventos y control de acceso
b) Guardias de seguridad
c) Libro de novedades
d) Puertas cerradas
e) Llaves o Claves
f) Dispositivos de seguridad biométricos
g) Cámaras

10.2.8. Continuidad del negocio

a) Redundancia
b) Corriente ininterrumpida (UPS)

10.2.9. Respaldo

10.2.10. Detección y Corrección de Fallas

a) Control de temperatura y humedad
b) Detectores y alarmas de: calor,humo y humedad

c) Sistema de extinción de incendios

10.3. Administración directa en el equipo

Verificar existencia o no de vulnerabilidades en cuanto a:

10.3.1. Accesos no autorizados.

10.3.2. Copia de información a través de dispositivos removibles

10.3.3. Accidentes

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.3.1. Políticas de acceso estrictas

a) Registro de eventos y control de acceso
b) Guardias de seguridad
c) Libro de novedades
d) Puertas cerradas
e) Llaves o Claves
f) Dispositivos de seguridad biométricos
g) Cámaras

10.3.2. Continuidad del negocio

a) Redundancia
b) Corriente ininterrumpida (UPS)

10.3.3. Respaldo

10.3.4. Detección y Corrección de Fallas

a) Control de temperatura y humedad
b) Detectores y alarmas de: calor,humo y humedad
c) Sistema de extinción de incendios

11. Verificación y comprobación de seguridad de la **Capa de Red:**

11.1. Equipos de acceso público a través de la red

Verificar existencia o no de vulnerabilidades en cuanto a:

11.1.1. Deficiente protección de los datos en el tránsito

11.1.2. Suplantación de una página web real

11.1.3. Denegación de Servicio (DoS)

11.1.4. Malware (gusanos, virus)

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.1.5. Firewall

a) Modelo de seguridad positivo (whitelist)
b) Filtro por direcciones IP
c) Bloquear puertos innecesarios o privados

11.1.6. Antivirus

11.1.7. DMZ

11.1.8. IDS/IPS

11.1.9. VPN
11.1.10. SSL/TLS
11.1.11. Balanceo de Carga

11.2. Equipos de acceso privado

Verificar existencia o no de vulnerabilidades en cuanto a:

11.2.1. Deficiente protección de los datos en el tránsito
11.2.2. Suplantación de IP (IP spoofing)
11.2.3. Suplantación de una página web real
11.2.4. Denegación de Servicio (DoS)
11.2.5. Malware (gusanos, virus)

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.2.6. Firewall
a) Modelo de seguridad positivo (whitelist)
b) Filtro por direcciones IP
c) Bloquear puertos innecesarios o privados
11.2.7. Antivirus
11.2.8. DMZ
11.2.9. IDS/IPS
11.2.10. VPN
11.2.11. SSL/TLS
11.2.12. Balanceo de Carga

12. Verificación y comprobación de seguridad la **capa de Sistema Operativo y servicios:**

12.1. Sistema operativo

Verificar existencia o no de vulnerabilidades en cuanto a:

12.1.1. Acceso no autorizado
12.1.2. DoS
12.1.3. Malware
12.1.4. Buffer Overflow
12.1.5. Intrusiones
12.1.6. Versiones viejas
12.1.7. Falta de actualizaciones
12.1.8. Configuración por defecto
12.1.9. Exposición de servicios innecesarios
12.1.10. Credenciales predecibles o por defecto
12.1.11. Credenciales

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.1.12. Firewall
12.1.13. IDS, IPS

12.1.14. Antivirus
12.1.15. Versión reciente.
12.1.16. Actualizaciones y parches automáticas
12.1.17. Verificación de credenciales y permisos

12.2. ¿Se encuentra en Hospedaje Compartido?

Verificar existencia o no de vulnerabilidades en cuanto a:

12.2.1. Débil autenticación
12.2.2. Débil autorización
12.2.3. Revelación de información
12.2.4. Exposición de archivos delicados(fuentes, ejecutables, archivos configuración)
12.2.5. Configuración por defecto
12.2.6. Deficiente definición de permisos de usuario
12.2.7. Información delicada no cifrada

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.2.8. Verificación de credenciales y permisos
12.2.9. Cifrado de los datos físicos
12.2.10. Establecer permisos estrictos para las aplicaciones y usuarios.
12.2.11. Resguardar recursos álgidos (códigos fuentes, archivos de configuración)

12.3. ¿Administración del sistema en forma remota?

Verificar existencia o no de vulnerabilidades en cuanto a:

12.3.1. Deficiente autenticación y autorización
12.3.2. Configuraciones por defecto
12.3.3. Versiones viejas, falta de actualización o parches
12.3.4. Deficiencia en canales de comunicación, protocolos no seguros

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.3.5. Canal de comunicación seguro (SSH, SSL, VPN)
12.3.6. Políticas de vencimiento y cambio de credenciales
12.3.7. Chequeo de configuración desde cero
12.3.8. Firewall (lista blanca)
12.3.9. Verificación de credenciales y permisos

12.4. ¿Actualizaciones remotas?

Verificar existencia o no de vulnerabilidades en cuanto a:

12.4.1. Deficiente autenticación y autorización
12.4.2. Deficiencia en canales de comunicación, protocolos no seguros(FTP)
12.4.3. Accidentes

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.4.4. Canal de comunicación seguro (SSH, SFTP, VPN)
12.4.5. Políticas de vencimiento y cambio de credenciales

12.4.6. Firewall (lista blanca)

12.4.7. Verificación de credenciales y permisos

12.5. ¿Actualizaciones frecuentes?

Verificar existencia o no de vulnerabilidades en cuanto a:

12.5.1. Reemplazo de información

12.5.2. Pruebas insuficientes

12.5.3. Accidentes

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.5.4. Procesos de QA.

12.5.5. Organización y métodos para actualizaciones

12.5.6. Verificación de credenciales y permisos.

12.6. ¿Utilizan servidores de aplicación o servidores web?

Verificar existencia o no de vulnerabilidades en cuanto a:

12.6.1. Débil autenticación

12.6.2. Configuración por defecto

12.6.3. Versiones viejas

12.6.4. Falta de actualizaciones o parches

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.6.5. Versión reciente.

12.6.6. Actualizaciones y parches.

12.6.7. Chequeo de configuración desde cero.

12.6.8. Verificación de credenciales y permisos.

12.7. ¿Utilizan servidores de bases de datos?

Verificar existencia o no de vulnerabilidades en cuanto a:

12.7.1. Deficiente respaldo

12.7.2. Deficiente autorización

12.7.3. Débil autenticación

12.7.4. Configuración por defecto

12.7.5. Versiones viejas

12.7.6. Falta de actualizaciones o parches

12.7.7. Canal de comunicación no seguro

12.7.8. Exposición de recursos innecesaria

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.7.9. Versión reciente

12.7.10. Cifrado de datos

12.7.11. Actualizaciones y parches

12.7.12. Chequeo de configuración desde cero

12.7.13. Verificación de credenciales y permisos

12.8. Utilizan servidores FTP

Verificar existencia o no de vulnerabilidades en cuanto a:

12.8.1. Deficiente respaldo
12.8.2. Deficiente autorización
12.8.3. Débil autenticación
12.8.4. Configuración por defecto
12.8.5. Versiones viejas
12.8.6. Falta de actualizaciones o parches
12.8.7. Canal de comunicación no seguro
12.8.8. Exposición de recursos innecesaria

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.8.9. Versión reciente
12.8.10. Cifrado de datos
12.8.11. Canal comunicación seguro (SSL)
12.8.12. Actualizaciones y parches
12.8.13. Chequeo de configuración desde cero
12.8.14. Verificación de credenciales y permisos

12.9. Uso de tecnologías del lado del servidor para contenido dinámico

Verificar existencia o no de vulnerabilidades en cuanto a:

12.9.1. Abuso de funcionalidad
12.9.2. Ejecución super usuario
12.9.3. Deficiente autorización
12.9.4. Configuración por defecto
12.9.5. Versiones viejas
12.9.6. Falta de actualizaciones o parches

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.9.7. Chequeo de configuración desde cero
12.9.8. Versión reciente, Actualizaciones y parches
12.9.9. Verificación de credenciales y permisos de ejecución

13. Verificación y comprobación de seguridad de la **capa de aplicación:**

13.1. Inicio de Sesión de Usuario.

Verificar existencia o no de vulnerabilidades en cuanto a:

13.1.1. Ataques de Fuerza bruta
13.1.2. Abuso de funcionalidad
13.1.3. DoS
13.1.4. Revelación de información
13.1.5. Inyección de código o comandos
13.1.6. Phishing

13.1.7. inyección de códigos (SQL, HTML, LDAP, entre otros)

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.1.8. Captcha
13.1.9. Filtrado de entrada
13.1.10. Escapado de salida
13.1.11. Conexiones seguras
13.1.12. Manejo de sesión apropiado
13.1.13. Enmascarado de información sensible
13.1.14. Contador de intentos
13.1.15. Registro de conexiones
13.1.16. Revisión de comentarios en el código
13.1.17. Manejo de excepciones y mensajes apropiado
13.1.18. Contraseñas fuertes
13.1.19. Educación a usuarios

13.2. Recuperación de Contraseña

Verificar existencia o no de vulnerabilidades en cuanto a:

13.2.1. Débil Validación en la Recuperación de Contraseñas
13.2.2. Ataques de Fuerza bruta
13.2.3. Abuso de funcionalidad
13.2.4. DoS
13.2.5. Revelación de información
13.2.6. Inyección de código o comandos
13.2.7. Phishing
13.2.8. inyección de códigos (SQL, HTML, LDAP, entre otros)

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.2.9. Captcha
13.2.10. Filtrado de entrada
13.2.11. Conexiones seguras
13.2.12. Manejo de sesión apropiado
13.2.13. Enmascarado de información sensible
13.2.14. Revisión de comentarios en el código
13.2.15. Manejo de excepciones y mensajes apropiado

13.3. Administradores Remotos

Verificar existencia o no de vulnerabilidades en cuanto a:

13.3.1. Autenticación insuficiente
13.3.2. Autorización insuficiente
13.3.3. Robo de Sesión
13.3.4. XSS
13.3.5. XSRF

13.3.6. Fijación de Sesión
13.3.7. Fuga de Información
13.3.8. Localización de Recursos Predecibles
13.3.9. Abuso de Funcionalidad.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.3.10. Autenticación adecuada
13.3.11. Verificación de Autorización
13.3.12. Renovación de credenciales
13.3.13. Certificados digitales

13.4. Registro de usuarios en línea

Verificar existencia o no de vulnerabilidades en cuanto a:

13.4.1. Revelación de Información
13.4.2. Abuso de Funcionalidad
13.4.3. Inyección de código SQL
13.4.4. Phishing.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.4.5. Captcha
13.4.6. Filtrado de entrada
13.4.7. Escapado de salida
13.4.8. Conexiones seguras

13.5. Usuarios mantienen sesión

Verificar existencia o no de vulnerabilidades en cuanto a:

13.5.1. Predicción de credenciales/sesión
13.5.2. Robo de sesión (Session Hijacking)
13.5.3. Fijación de sesión (Session Fixation)
13.5.4. XSS
13.5.5. XSRF

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.5.6. Escapado de salida
13.5.7. Verificación de autorización
13.5.8. Renovación de credenciales
13.5.9. Caducidad y vencimiento de sesiones
13.5.10. Políticas de vencimiento y cambio de contraseñas

13.6. Usuarios realizan operaciones

Verificar existencia o no de vulnerabilidades en cuanto a:

13.6.1. Abuso de funcionalidad
13.6.2. Anti-automatización insuficiente
13.6.3. DoS

13.6.4. Repudiación de Operaciones
13.6.5. Fijación de sesión
13.6.6. Suplantación de contenido
13.6.7. Cross-Site Request Forgery

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.6.8. Filtrado de entrada
13.6.9. Escapado de salida
13.6.10. Token
13.6.11. Captcha
13.6.12. Renovación de credenciales
13.6.13. Verificación de autorización
13.6.14. Logs y registro de acciones
13.6.15. Notificaciones al usuario
13.6.16. Educación a usuarios

13.7. Sitios Web que se comunican con Usuarios.

Verificar existencia o no de vulnerabilidades en cuanto a:

13.7.1. Phishing
13.7.2. Revelación de información
13.7.3. XSS
13.7.4. XSRF

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.7.5. Educación a usuarios
13.7.6. Escapado de la salida

13.8. Pase de parámetros accesibles al usuario

Verificar existencia o no de vulnerabilidades en cuanto a:

13.8.1. Revelación de Información
13.8.2. Referencia Directa de objetos
13.8.3. Autorización insuficiente
13.8.4. Inyección de código
13.8.5. XSS
13.8.6. XSRF

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.8.7. Filtrado de entrada
13.8.8. Verificación de autorización

13.9. Manipulan información sensible

Verificar existencia o no de vulnerabilidades en cuanto a:

13.9.1.
13.9.2. Revelación de Información

13.9.3. Autorización insuficiente

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.9.4. Cifrado de datos

13.9.5. Verificación de autorización

13.9.6. Enmascarado de información sensible

13.9.7. Autenticidad

13.9.8. Certificado digital

13.9.9. SSL,VPN

13.10. Los usuarios suben archivos al servidor

Verificar existencia o no de vulnerabilidades en cuanto a:

13.10.1. Versiones viejas o sin parchear

13.10.2. deficiente autorización

13.10.3. accesos no autorizados y violación de privilegios

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.10.4. Antivirus

13.10.5. Filtrado de entrada

13.10.6. Captcha

13.11. La aplicación interactúa con base de datos

Verificar existencia o no de vulnerabilidades en cuanto a:

13.11.1. Inyección de comandos

13.11.2. Canal de comunicación no seguro

13.11.3. Revelación de información

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.11.4. Resguardo de credenciales

13.11.5. Filtrado de datos

13.11.6. Escapado de salida

13.11.7. Cifrado de datos

13.12. Uso de Plugins (Windows media player, Acrobat, propietarios, .exe, ActiveX, toolbars)

Verificar existencia o no de vulnerabilidades en cuanto a:

13.12.1. Versiones viejas o sin parchear

13.12.2. deficiente autorización

13.12.3. accesos no autorizados y violación de privilegios

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

13.12.4. Versiones actuales

13.12.5. Parches

13.12.6. Exigir últimas versiones

13.12.7. Token

13.12.8. Verificación de autorización

Etapa 3: Comunicación de Resultados

Los informes de auditoría son el producto final del trabajo del auditor, este informe es utilizado para indicar las observaciones y recomendaciones a la gerencia, desarrolladores o integradores de sistemas. Aquí también se expone la opinión y analisis sobre lo adecuado o lo inadecuado de los controles, medidas de seguridad y procedimientos revisados durante la auditoría. No existe un formato específico para exponer el informe de auditoría, pero generalmente se puede regir bajo la siguiente estructura:

- Introducción al informe, donde se reiteran los objetivos y el alcance de la auditoría. Se puede expresar un resumen breve de los resultados generales de la auditoria y especificar si se pudieron cumplir o no la totalidad de los objetivos.
- Observaciones detalladas y recomendaciones de auditoría con respecto a los objetivos cumplidos, los objetivos no cumplidos y en general para todas las vulnerabilidades y deficiencias de seguridad encontradas.
- Conclusión global del auditor expresando una opinión sobre los controles y procedimientos revisados.

De ser necesario, se pueden planificar exposiciones o reuniones conjuntamente con los informes de auditoría, a fin de exponer estos resultados al personal técnico, grupos de desarrollo y gerentes.

Etapa 4: Verificación Posterior (Follow-up):

El trabajo de auditoría es un proceso continuo. Se debe llevar seguimiento de que las observaciones producto de la auditoría han sido comprendidas y que las medidas de seguridad necesarias se han tomado correctamente. No serviría de nada el trabajo de auditoría si no se comprueba que las acciones correctivas, se fueron realizadas y que su implementación no supone nuevas vulnerabilidades y amenazas.

Revisar y verificar estos controles es de vital importancia, ya que con la implementación de los mismos, se puede estar incorporando nuevas vulnerabilidades y amenazas de seguridad. En otras palabras se deben certificar los controles y medidas de seguridad.

Es esta etapa se puede determinar la realización o no de una nueva auditoría. Parcial o total con respecto a las diferentes capas y características de la aplicación.

Capítulo 7

Conclusiones y Consideraciones

7.1. Conclusiones

La evolución de las tecnologías web esta apuntando en una dirección que permite a las aplicaciones web ser cada vez más eficientes, con mejores soluciones y más interactivas. Tales procesos, de cualquier forma, también incrementan las amenazas a los que los negocios, empresas y desarrolladores web enfrentan día a día.

A través de la investigación realizada en el tema de las vulnerabilidades y amenazas de seguridad en aplicaciones web y las medidas de seguridad necesarias a tomar en consideración para protegerlas, se logró documentar características principales en las aplicaciones web y sus consideraciones en materia de seguridad.

Se definió un esquema basado en capas desde el punto de vista de la seguridad, con el fin de organizar los contenidos de este trabajo bajo una estructura que permita el mayor entendimiento y comprensión de los temas tratados.

Con este trabajo especial de grado se logró crear una propuesta de metodología para la auditoría de seguridad en aplicaciones web, adaptándose a las necesidades de auditores, desarrolladores e integradores de software en materia de seguridad para aplicaciones web. Dándole un valor agregado a las empresas y compañías en cuanto a la seguridad de la información para sus proyectos en la web.

En cuanto a la ejecución de la metodología se notó que un aspecto clave y decisivo para el éxito en la ejecución de la misma, es que debe existir una total colaboración por parte de los diferentes equipos de trabajo relacionados con la aplicación web a auditar, ya que a partir de ellos es que se obtiene la mayor cantidad de información necesaria para la correcta aplicación de la metodología.

El auditor no necesariamente es experto en cada uno de los aspectos relacionados con las aplicaciones web. Debido a la inmensa cantidad de lenguajes de programación y diferentes arquitecturas posibles, conocer las mejores prácticas en cada una de estas resulta imposible. Pero sí se debe tener una comprensión general de cada uno de estos elementos.

En tal sentido se recomienda que en lugar de un auditor, esta metodología sea ejecutada por un grupo de auditores. Este grupo debe ser multidisciplinario, comprendido en todas la areas de

conocimiento que abarca esta metodología.

Una de las dificultades más notables encontradas durante la aplicación de la metodología, es para el caso en que las aplicaciones web se encuentran bajo una plataforma de terceros, es decir una empresa de hospedaje (hosting). Se hace difícil obtener y verificar información de aspectos como infraestructura física, equipos en la red y sus características, políticas de seguridad de la empresa de hospedaje, entre otros.

También el tiempo de aplicación de la metodología es un aspecto importante, ya que a pesar de que uno de los casos fue el de una compañía pequeña, llevó gran cantidad de tiempo levantar información, organizar, analizar los procesos, aprender la arquitectura de software, así como discutir y generar los reportes necesarios, evitando interrumpir lo menos posible en las actividades diarias de la organización y sin tocar datos sensibles de la empresa.

Por otro lado, del análisis de los resultados de la metodología se evidencia la importancia que tiene la correcta definición de políticas, junto con las prácticas y técnicas de implementación. Establecen un marco de conducta o comportamiento a la vez que regulan las acciones y las formas de los procesos dentro y fuera de la aplicación. No sirve de nada la mejor cerradura del mundo si no se establecen reglas (políticas) que indiquen cuándo debe permanecer abierta o cerrada.

El trabajo de auditoría es un proceso continuo. Debido a la constante aparición de nuevas vulnerabilidades y amenazas de seguridad, se requiere que la ejecución de las auditorías sea un proceso constante e iterativo. Se debe volver a aplicar cada cierto tiempo en busca de nuevas vulnerabilidades y amenazas presentes en la aplicación.

Las aplicaciones web deben ser seguras en cada una de sus capas con respecto a la seguridad. De nada vale una aplicación web programada de la forma más segura posible, si las instalaciones físicas donde reside no son seguras, o si el sistema operativo de la máquina donde corren no esta protegido. Es de vital importancia identificar las vulnerabilidades y amenazas presentes en cada una de las capas de la aplicación y tomar las medidas necesarias para prevenirlas o minimizar su impacto. Y es responsabilidad de cada rol específico en cada capa, realizar el trabajo de proteger las mismas.

La metodología se aplicó en tres (3) casos de estudio diferentes con resultados satisfactorios en todos ellos, a pesar de estar desarrolladas por distintos equipos de trabajo, bajo diferentes plataformas de hardware y software, en diferentes lenguajes y tecnologías del lado del servidor. Esto demuestra la versatilidad de la metodología y que no esta comprometida a una arquitectura o lenguaje de programación específico.

Los mecanismos de control y medidas de seguridad para las aplicaciones web son eficientes. Es realmente difícil o imposible, para un atacante, realizar un ataque exitoso si en una aplicación se considera un mínimo en cuanto a medidas de seguridad. Sin embargo existe un problema que ninguna implementación de software o hardware puede prevenir. El factor humano. Los atacantes adoptan otras estrategias, como la de persuadir o engañar a personas para que le den credenciales de acceso o información confidencial. Por tanto la seguridad debe partir de las personas hacia los sistemas, y no solo descansar toda la responsabilidad de resguardar la información en lo fuerte de las medidas de seguridad. Se debe educar a las personas en cuanto al manejo de la información, así como hacerlas conscientes de los riesgos y amenazas a los que está expuesta la misma.

7.2. Consideraciones

Se trata de una metodología no comercial, ya que en sus pasos se verifica adicionalmente a las vulnerabilidades y amenazas propias de la aplicación, la existencia o no de controles para minimizar los riesgos de las vulnerabilidades. También en los informes de auditoría se debe indicar los controles y medidas de seguridad a ser aplicados. Al aplicar esta metodología de forma comercial, el auditor deberá omitar en su informe las soluciones y medidas de seguridad necesaria, y limitarse a solo ofrecer un reporte de vulnerabilidades y amenazas existentes.

A pesar de la gran cantidad de información que existe en cuanto a vulnerabilidades, amenazas y ataques para las aplicaciones web, así como información sobre medidas de seguridad, es muy poca o inexistente la información sobre las características comunes de las aplicaciones web. Tal caracterización debió hacerse para tener un marco de referencia sobre el cual se realizarían las auditorías.

No siempre es posible constatar la información sobre la arquitectura física y de red para los casos de estudio con aplicaciones web bajo servicios de hospedaje. Se tuvo que confiar plenamente en la información del proveedor de servicio de hosting.

Existen limitaciones al aplicar la metodología asociadas al conocimiento y dominio en cada una de las áreas que comprende la misma. Por ejemplo, para ofrecer un informe preciso acerca de vulnerabilidades en la capa de aplicación, se debe revisar el código fuente de la misma, y puede que no sea una tarea fácil sino se posee conocimientos del lenguaje de programación en la que esta implementada o del framework de trabajo en el que se realizó. Incluso entender el código de otras personas, si estas no usan ningún tipo de estandar o buenas prácticas de programación, puede resultar una tarea imposible. La recomendación en este caso es apoyarse en los programadores y responsables del desarrollo del proyecto.

Al tiempo en que se entrega este trabajo, se descubren y aparecen nuevas vulnerabilidades y amenazas de seguridad. Debido a la inmensa cantidad de las mismas, con sus respectivos casos específicos y variantes, se hace imposible considerarlas todas. Sin embargo, los conceptos discutidos en este trabajo permiten la compresión e incluso la fácil adaptación de las medidas de seguridad contra las nuevas vulnerabilidades y amenazas que puedan encontrarse.

7.3. Aportes del trabajo

1. Definición de un esquema por capas desde el punto de vista de la seguridad para las aplicaciones web
2. Recopilación de vulnerabilidades y amenazas en las aplicaciones web
3. Recopilación de medidas de seguridad en las aplicaciones web
4. Definición y clasificación de las principales características en las aplicaciones web y sus implicaciones de seguridad

5. Metodología para la auditoría de seguridad de Aplicaciones Web
6. Aplicación de la metodología a tres (3) casos de estudio satisfactoriamente

7.4. Trabajos a Futuro

1. Profundizar aún más en el esquema por capas desde el punto de vista de la seguridad para las aplicaciones web
2. Extender y ampliar el conjunto de características comunes en las aplicaciones web
3. Recopilar las nuevas vulnerabilidades y amenazas en las aplicaciones web
4. Recopilar las nuevas medidas de seguridad en las aplicaciones web
5. Realizar adaptaciones para arquitecturas y lenguajes específicos de la metodología para la auditoría de seguridad de Aplicaciones Web
6. Definir lineamientos para el desarrollo de Aplicaciones Web seguras

Anexos

Caso de estudio # 1: www.guiamamaybebe.com

Etapa 1: Planificación

Describir el sistema

1. Al realizar la etapa previa de la metodología se puede concluir que Guía Mamá y Bebé .com es un portal Web Creado para alojar a todas las compañías, comercios e individuos que venden y/o prestan servicios para el ramo materno infantil en Venezuela. A su vez mantiene informado a todas las madres con gran cantidad de información de interés, que se publican diariamente en el portal.

   Es una guía que tienen las Madres en Internet para encontrar cualquier bien o servicio relacionado con la etapa prenatal y postnatal, hasta los 5 años de vida del bebé, todo lo que buscan en un mismo lugar y en un solo clic

   Esta página esta activa desde hace 5 años, y su ranking en www.alexa.com es de 225.810 para el día 23 de Abril de 2008

   La página se encuentra alojada en un servicio de hosting compartido que presta la empresa 1&1, y posee las siguientes características:

   La página a nivel de diseño presenta la siguiente disposición:

Figura 7.1: Disposición de la página (Elaboración propia)

Todas las páginas que visitan los usuarios, tienen:

a) En la parte superior un encabezado, donde del lado izquierdo esta el logo de la empresa, y del lado derecho anuncios de publicidad aleatorios.

b) En la parte izquierda hay en primer lugar la lista de los temas de los artículos, en segundo lugar unos anuncios de google, y en tercer lugar una sección denominada de interés, donde se colocan enlaces a varios temas de interés que puedan resultar resaltantes para un usuario.

c) En la parte central se muestra el contenido correspondiente a la sección de la página visitada.

d) En la parte derecha hay en primer lugar una lista de los artículos más leídos, en segundo lugar una lista de los artículos más comentados y en tercer lugar una lista de artículos recomendados por los administradores de la página.

Pasos:

Capa Física

2. Describa lo mejor posible y con el mayor detalle la infraestructura física de los dispositivos y equipos en el sistema, así como las instalaciones físicas, edificio, cuarto de servidores, puertas, etc. Detalle de los equipos servidores, dispositivos de red, cableado, tipo de instalaciones, personal en el sitio, etc. Indique si dicha arquitectura es de administración propia o por parte de terceros. En el caso que sea administrada por terceros, por ejemplo a través de un servicio de hospedaje (hosting), solicitar al proveedor toda la información referente al mismo. Tomar nota de si es un servicio de hosting compartido o de uso exclusivo. Puntos de accesos a las instalaciones, controles de acceso (llaves, claves, cámaras de vigilancia, dispositivos biométricos, vigilantes, registro de novedades), personas autorizadas, vigilantes, otros.

   2.1. ¿Hardware propio de la empresa?: No aplica.

   2.2. ¿Hardware arrendado a terceros? (hosting): Sí, esta página se encuentra alojada en un servicio de hosting que proporciona la empresa 1&1, la cual cuenta con aproximadamente 7.07 millones de clientes. Posee múltiples paquetes de dominio y planes de alojamiento web, servidores privados virtuales, servidores dedicados potentes y listos

para ejecutar las aplicaciones web, 1&1 ofrece una gama completa de servicios web para cada necesidad, cada nivel de destreza y cada presupuesto. Todos los productos 1&1 han sido diseñados para proporcionar la combinación ideal de características, a un precio accesible.

Información adquirida de la página web www.1and1.com.

2.2.1. Políticas de acceso: Posee un sistema de acceso que garantiza que sólo las personas autorizadas entren a la sala de servidores.

2.2.2. Personal autorizado: Información no disponible.

2.2.3. Instalaciones físicas - Edif., habitación, cuarto de servidores: Posee múltiples salas de servidores que albergan 40.000 servidores cada una con sistemas de enfriamiento, sistemas de protección contra fuego, sistema de autorización de acceso, redundantes UPS. Posee una sala de bombas la cual contiene todo el sistema mecánico que conecta la infraestructura de enfriamiento externa a esta sala con el sistema de tuberías que proporcionan enfriamiento al edificio. Posee un laboratorio donde los equipos son probados para asegurar interoperabilidad, estabilidad, seguridad y operación a lo largo del tiempo. Posee un cuarto de almacen donde los equipos nuevos son almacenados de manera rápida y eficiente. Posee múltiples salas de baterías. Posee múltiples salas electricas que se encargan de distribuir la energía, utilizando transformadores, interruptores de transferencia automática y sistemas de UPS, Posee 2 salas de red separadas que proveen un espacio de seguridad para sistemas de redundancia de enrutamiento que constantemente transportan datos entre los servidores y la conexión de Internet de 20 Gb, el flujo de datos entre los servidores es de 100 Mbps, estos se encuentran protegidos con sistemas de detección de intrusos producto de redundantes sistemas de cortafuegos (firewall). Posee una zona de plantas generadoras del continuo flujo de la mezcla de agua/glycol necesaria para los sistemas de enfriamiento. Finalmente, posee un área con 2 generadores que suministran cerca de 2250Kw y 2 independientes interruptores de transferencia automática (Automatic Transfer Switches-ATS) que monitorean la alimentación de energía y automáticamente prenden o apagan los generadores cuando sea necesario.

2.2.4. Los equipos y dispositivos.

Características en cuanto a:

a) Servidores: 5 salas de servidores proporcionan espacio para 47.000 servidores.

b) Cables para conexiones: Información no disponible.

c) Alimentación (UPS): Sistemas UPS redundantes, sistemas generadores de diesel, que garantizan que los servidores funcionen independientemente de las condiciones externas.

d) otros: Información no disponible.

2.3. Administración directa en el equipo: No aplica.

Capa de Red

3. Describa lo mejor posible y con el mayor detalle la arquitectura de red en donde reside la aplicación. Detalles de configuración de la red, identificar servidores, protocolos de red utilizados, existencia de enlaces privados o públicos (VPN, intranet o enlaces dedicados), presencia de Firewall y/o IDS, existencia o no de una DMZ, etc. Realice diagramas, esquemas

y tablas.Indique si dicha arquitectura es de administración propia o por parte de terceros. En el caso que sea administrada por terceros, por ejemplo a través de un servicio de hospedaje (hosting), solicitar al proveedor toda la información referente al mismo. Usar herramientas de escaneo de vulnerabilidades automatizado (Nmap, nessus, retina, otros).

3.1. ¿Administración por terceros? (hosting): Sí.

Si, solicitar información a compañía de hospedaje

3.2. Identificar los equipos en la red

3.2.1. Direcciones IP privadas: Información no disponible.

3.2.2. Direcciones IP públicas: Información no disponible.

3.2.3. Equipos en la red local: Posee 47.000 servidores instalados, Posee routers Cisco Serie 7600, para manejar el tráfico del sitio web.

3.2.4. Equipos remotos: Información no disponible.

3.3. Identificar dispositivos de firewall

3.3.1. Filtrado lista blanca o lista negra: Información no disponible.

3.3.2. Políticas de seguridad: Todos los servidores se encuentran protegidos con redundantes sistemas de cortafuegos (firewall) que permiten detectar intrusos.

3.3.3. Segmentación de redes: Información no disponible.

3.4. IDS/IPS: Poseen sistema de detección de intrusos.

3.5. Identificar protocolos de red usados: utilizan IP.

3.6. Esquematizar la arquitectura de red (hacer uso de técnicas de modelado)

3.6.1. Diagramas UML de componentes especializados para redes

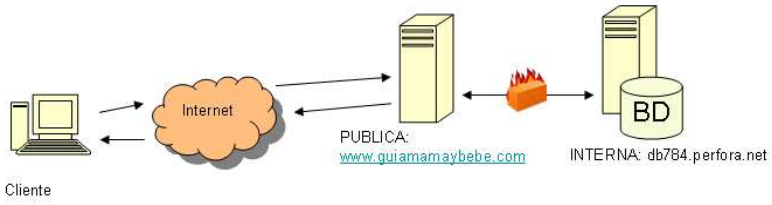


Figura 7.2: Diagrama de la red

3.7. Servidores de DNS privados y públicos: ns57.1and1.com y ns58.1and1.com.

Capa de Sistema Operativo y Servicios

4. Describa lo mejor posible y con el mayor detalle las características de los equipos en cuanto a, el sistema operativo que usan, el sistemas de archivos y esquemas de permisologías existentes, servicios disponibles, usuarios del sistema, ubicación de los recursos en los sistemas.

4.1. Sistema operativo: Windows.

4.1.1. Distribución del SO: Windows NT.

4.1.2. Versión: Windows NT 5.2 (Build 3790: Service Pack 2).

4.1.3. Usuarios y Grupos: Información no disponible.

4.1.4. Usuarios y Grupos administradores (superusuarios): Información no disponible.

4.2. Servicios

4.2.1. Servidores de aplicaciones (servidores web): IIS, Apache.

4.2.2. Servidores de bases de datos: MSQL Server, MySQL.

4.2.3. Servidores de correo: Servidor de entrada de mail (IMAP): imap.1and1.com.
Servidor de entrada de mail (POP): pop.1and1.com.
Servidor de salida de mail (SMTP): smtp.1and1.com.

4.2.4. Servidores LDAP: No aplica.

4.2.5. Servidores FTP: Disponible.

4.2.6. Otros servicios: Administración remota de permisos de directorios y archivos, Administración remota de sevicios de base de datos.

4.3. Para cada servicio

IIS.

4.3.1. Versión: 6.0.

4.3.2. Usuarios y permisos de ejecución: Información no disponible.

4.3.3. Archivos y directorios relacionados: Información no disponible.

4.3.4. Puertos que utilizan: puerto 80.

4.3.5. Cuentas de usuarios del servicio: Información no disponible.

4.3.6. Archivos de configuración, propiedades: Información no disponible.

Apache.

4.3.1. Versión: 2.

4.3.2. Usuarios y permisos de ejecución: Información no disponible.

4.3.3. Archivos y directorios relacionados: Información no disponible.

4.3.4. Puertos que utilizan: puerto 80.

4.3.5. Cuentas de usuarios del servicio: Información no disponible.

4.3.6. Archivos de configuración, propiedades: Información no disponible.

MSQL Server.

4.3.1. Versión: Microsoft SQL Server 2005 - 9.00.1399.06 (Intel X86) .

4.3.2. Usuarios y permisos de ejecución: Información no disponible.

4.3.3. Archivos y directorios relacionados: Información no disponible.

4.3.4. Puertos que utilizan: Información no disponible.

4.3.5. Cuentas de usuarios del servicio: Información no disponible.

4.3.6. Archivos de configuración, propiedades: Información no disponible.

MySQL.

4.3.1. Versión: 5.0.

4.3.2. Usuarios y permisos de ejecución: Información no disponible.

4.3.3. Archivos y directorios relacionados: Información no disponible.

4.3.4. Puertos que utilizan: Información no disponible.

4.3.5. Cuentas de usuarios del servicio: Información no disponible.

4.3.6. Archivos de configuración, propiedades: Información no disponible.

4.4. Sistema de archivos

4.4.1. Ubicación y contexto de las aplicaciones: Información no disponible.

4.4.2. Identifique recursos compartidos: Información no disponible.

4.4.3. Identifique directorios protegidos: Información no disponible.

4.4.4. Ruta de instalación de servicios: Información no disponible.

4.5. Administración remota

4.5.1. A través de red interna: Disponible.

4.5.2. VPN: Información no disponible.

4.5.3. SSH: Información no disponible.

4.5.4. Sistemas de Administración Remota: Disponible.

5. Identifique las aplicaciones o módulos presentes en la aplicación web. Para cada una de estos elementos determinar como mínimo el nombre de aplicación o módulo, versión, ubicación, permisos de ejecución relacionados (si aplica). Algunas tecnologías comúnmente relacionadas con las aplicaciones web son: servidores de aplicaciones o servidores web (Apache, IIS, Tomcat...), servidores de base de datos, tecnologías del lado del servidor (Java, ASP .Net, PHP, Ruby...), entre otras.

5.1. Identificar servidores de aplicaciones

5.1.1. Servidor Web: IIS.

5.1.2. Versión: 6.0.

5.1.3. Permisos de ejecución: Información no disponible.

5.1.4. Accesos y permisos de directorios: Información no disponible.

5.1.5. Usuarios o cuentas de administración: Información no disponible.

5.1.6. Módulos instalados o librerías: Posee 42 componentes instalados, entre los cuales destacan: ADODB, MSXML2, SMTPsvg.Mailer, SoftArtisans.FileUp, WScript.Shell.

5.2. Servidores de base de datos

5.2.1. Servidor de base de datos: MSQL Server.

5.2.2. Versión: Microsoft SQL Server 2005 - 9.00.1399.06 (Intel X86) .

5.2.3. Ubicación: Información no disponible.

5.2.4. Permisos de ejecución: Información no disponible.

5.2.5. Accesos y permisos de directorios: Información no disponible.

5.2.6. Usuarios o cuentas de administración y usuarios: Usuario: dbo242750724

5.2.7. Acceso

a) Acceso Local: Disponible.

b) Acceso Remoto: Se utiliza la siguiente aplicación web: myLittleAdmin (for SQL Server and MSDE) v.2.7 r.123,para acceder remotamente al servidor de base de datos.

5.2.8. Cifrado de archivos físicos: Disponible pero no por defecto.

5.3. Servidores de Correo

5.3.1. Versión: IMAP,POP,SMTP.

5.3.2. Ubicación: imap.1and1.com, pop.1and1.com, smtp.1and1.com.

5.3.3. Permisos de ejecución: Información no disponible.

5.4. Servidores LDAP: No disponible.

5.5. Tecnologías del lado del servidor

5.5.1. Identificar las tecnologías usadas: ASP.

5.5.2. Versión: 4.0

5.5.3. Permisos de ejecución: Información no disponible.

5.5.4. Permisos de directorios: Información no disponible.

Capa de Aplicación

6. Describa lo mejor posible y con el mayor detalle las tecnologías usadas en las aplicaciones web. Identificar protocolos de comunicación (HTTP, HTTPS, FTP, LDAP...), uso de Frameworks (Struts, Spring, Cakephp, phpNuke, Ruby on Rails...) y los patrones de diseño (Composite view, MVC) que utilizan, tecnologías del lado del cliente (HTML, CSS, Javascript, AJAX, Flash, Plugins como Windows Media Player o Acrobat Reader, entre otros...), archivos y recursos de la aplicación, acceso a datos (servidores de base de datos, SAN, sistemas de archivos), servicios web.

6.1. Identificar protocolos de comunicación

6.1.1. HTTP: Disponible.

6.1.2. HTTPS (HTTP y SSL). Certificados digitales: No disponible.

6.1.3. SMTP

a) ¿Público o Privado?: Privado.

b) Autenticado ¿Sí o No?: Sí.

6.1.4. DNS: Disponible.

6.2. Determinar el uso de uno o más Frameworks. Conocer o investigar el Framework sobre el cual están hechas las aplicaciones ayuda a una mejor comprensión de sus funcionalidades y procesos, desde su estructura y organización de archivos y recursos hasta el código fuente e implementación.

6.2.1. Versión: Framework propio del desarrollador.

6.2.2. Librerías y complementos: No disponible.

6.2.3. Patrones: Composite View.

6.2.4. Estructura de la aplicación

a) Ubicación de archivos fuentes: E:\kunden\homepages\9\d242014665.

b) Ubicación de archivos de registro (logs): E:\logs.

c) Ubicación de archivos de configuración: E:\kunden\homepages\9\d242014665\modelo\.

d) Ubicación de archivos públicos (paginas, scripts, imágenes, css, multimedia, etc.): E:\kunden\
homepages\9\d242014665\.

e) Ubicación de archivos subidos al servidor: E:\kunden\
homepages\9\d242014665\images\.

6.3. Tecnologías del lado del cliente involucradas

6.3.1. HTML (XML, CSS): Disponible HTML y CSS.

6.3.2. Javascript, VBscript: Disponible Javascript y VBscript que es requisito para que la página funcione.

6.3.3. AJAX: Disponible.

6.3.4. Complementos o Plugins: Flash

6.3.5. Complementos propios: No disponible.

6.3.6. Cookies: Disponible.

6.4. Acceso a datos

6.4.1. Acceso a base de datos

a) Ubicación de los recursos: Información no disponible.

b) Almacenamiento de credenciales de acceso: Disponible.

c) Conexión segura: Disponible.

d) Cifrado de datos: No Disponible.

6.4.2. Acceso a archivos

a) Permisología sobre los recursos: Sólo tiene permisos de escritura el directorio E:\kunden\
homepages\9\d242014665\images\.

b) Recursos locales: Todos los directorios y archivos encontrados en E:\kunden\
homepages\9\d242014665\.

c) Recursos remotos: No disponible.

d) Cifrado de archivos: No disponible.

6.5. Acceso a servicios.

6.5.1. Servicios Web: No aplica.

7. Determinar funcionalidades o características que existen en la aplicación web. Identificando características como registro de usuario, inicio de sesión, uso de certificado digital o no, foros, formas para recuperación de contraseñas, modulo de administración, administradores de contenido, capacidades para subir archivos al servidor, ejecución de comandos, inclusión de archivos en el servidor, accesos a recursos remotos tales como otras aplicaciones, servicios web o recursos como imágenes, hojas de estilos entre otros.

7.1. En cada paso realizar:

7.1.1. Diagramas de flujo de las funcionalidades y procesos

Registro de clientes nuevos.

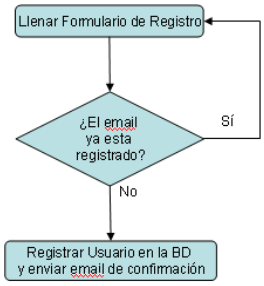


Figura 7.3: Registro de clientes nuevos (Elaboración propia)

Enviar información de contacto.

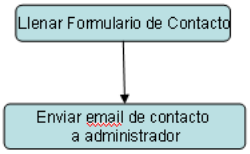


Figura 7.4: Enviar información de contacto (Elaboración propia)

Subir fotos al servidor.

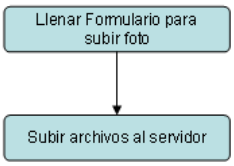


Figura 7.5: Subir fotos al servidor (Elaboración propia)

Llenar información de contenido de la página.

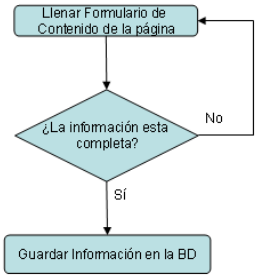


Figura 7.6: Llenar información de contenido de la página (Elaboración propia)

Recordar Contraseña.

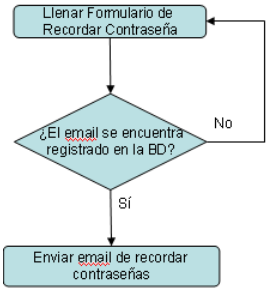


Figura 7.7: Recordar Contraseña (Elaboración propia)

7.1.2. Diagramas de secuencia UML

Diagrama de caso de uso de nivel 0.

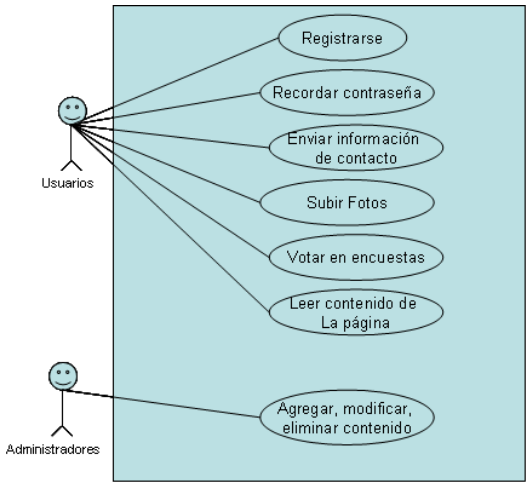


Figura 7.8: Diagrama de caso de uso de nivel 0 (Elaboración propia)

7.2. Identificar procesos de autenticación

7.2.1. Inicio de sesión con formularios web: Disponible.

7.2.2. Certificados digitales: No disponible.

7.2.3. Sistemas de recuperación de contraseña: Disponible

7.3. Identificar operaciones: Subir Fotos, Agregar, Modificar y eliminar Contenido de la página, Votar por encuesta.

7.4. Módulos de administración

7.4.1. Administradores de contenido: Disponible.

7.4.2. Administradores de consulta y operaciones: No disponible.

7.5. Procesos con capacidades de:

7.5.1. Subir archivos al servidor: Disponible.

7.5.2. Entradas de usuario: Disponible.

7.5.3. Envió de mensajes: Disponible, Se envían mensajes de contacto, al registrarse, boletines informativos, para recordar contraseñas.

7.6. Secciones o procesos que usan o exigen uso de javascript: Disponible: La validación de todos los formularios se hace usando javascript.

7.7. Secciones o procesos que usan o exigen uso de complementos: Para algunos anuncios publicitarios se necesita flash.

Etapa 2: Verificación y comprobación

8. Verificación y comprobación de seguridad de **Capa Física**:

8.1. Hardware propio de la empresa: No aplica.

8.2. Hardware arrendado a terceros

8.2.1. Obtener información de infraestructura y resguardo del mismo por parte tercero (Empresa de hospedaje, Outsourcing).

Verificar existencia o no de vulnerabilidades en cuanto a:

8.2.2. Accesos no autorizados: No presenta vulnerabilidades, ya que, el acceso sólo se permite a las personas autorizadas.

8.2.3. Catástrofes naturales: No presenta vulnerabilidades ,ya que, esta disponible la redundancia de información en otro edificio.

8.2.4. Accidentes: Información no disponible.

8.2.5. Fallas: Pueden presentarse.

8.2.6. Deficientes políticas de acceso: Puede presentarse.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

8.2.7. Políticas de acceso estrictas

a) Registro de eventos y control de acceso: Información no disponible.
b) Guardias de seguridad: Información no disponible.
c) Libro de novedades: Información no disponible.
d) Puertas cerradas: Información no disponible.
e) Llaves o Claves: Información no disponible.
f) Dispositivos de seguridad biométricos: Información no disponible.
g) Cámaras: Información no disponible.

8.2.8. Continuidad del negocio

a) Redundancia: Posee múltiples servidores donde se alojan los datos, Posee redundantes sistemas de cortafuegos, Posee redundantes sistemas de enfriamiento.
b) Corriente ininterrumpida (UPS): Posee redundancia de UPS.

8.2.9. Respaldo: Información No disponible.

8.2.10. Detección y Corrección de Fallas

a) Control de temperatura y humedad: Sistemas de enfriamiento.
b) Detectores y alarmas de: calor,humo y humedad: Sistema de detección temprana, con sistema doble de interlock que permite activar el sistema de rociadores.
c) Sistema de extinción de incendios: Sistema de protección de incendios.

8.3. Administración directa en el equipo: No aplica.

9. Verificación y comprobación de seguridad de la **Capa de Red:**

9.1. Equipos de acceso público a través de la red

Verificar existencia o no de vulnerabilidades en cuanto a:

9.1.1. Deficiente protección de los datos en el tránsito: Se presenta, ya que los datos viajan en texto plano.

9.1.2. Suplantación de IP (IP spoofing): No se presenta.

9.1.3. Suplantación de una página web real: No se presenta.

9.1.4. Denegación de Servicio (DoS): No se presenta, ya que se tiene redundancia en sistemas cortafuegos, que permiten detectar intrusos.

9.1.5. Malware (gusanos, virus): Información no disponible.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

9.1.6. Firewall

a) Modelo de seguridad positivo (whitelist): Se presenta como una opción a los administradores de los cortafuegos.

b) Filtro por direcciones IP: Se presenta como una opción a los administradores de los cortafuegos.

c) Bloquear puertos innecesarios o privados: Acceso sólo a los puertos 80/443, HTTP/S, 20/21(FTP), 25(SMTP, Mail) y 3389 para el escritorio remoto.

9.1.7. Antivirus: Información no disponible.

9.1.8. DMZ: Información no disponible.

9.1.9. IDS/IPS: Se presenta redundancia en sistemas de detección de intrusos.

9.1.10. VPN: Información no disponible.

9.1.11. SSL/TLS: Se presenta si es requerido, no viene por defecto, e implica costos adicionales.

9.1.12. Balanceo de Carga: Información no disponible.

9.2. Equipos de acceso privado

Verificar existencia o no de vulnerabilidades en cuanto a:

9.2.1. Deficiente protección de los datos en el transito: Información no disponible.

9.2.2. Suplantación de IP (IP spoofing): Información no disponible.

9.2.3. Suplantación de una página web real: Información no disponible.

9.2.4. Denegación de Servicio (DoS): Información no disponible.

9.2.5. Malware (gusanos, virus): Información no disponible.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

9.2.6. Firewall

a) Modelo de seguridad positivo (whitelist): Información no disponible.

b) Filtro por direcciones IP: Información no disponible.

c) Bloquear puertos innecesarios o privados: Información no disponible.

9.2.7. Antivirus: Información no disponible.

9.2.8. DMZ: Información no disponible.

9.2.9. IDS/IPS: Información no disponible.

9.2.10. VPN: Información no disponible.

9.2.11. SSL/TLS: Información no disponible.

9.2.12. Balanceo de Carga: Información no disponible.

10. Verificación y comprobación de seguridad la **capa de Sistema Operativo y servicios:**

10.1. Sistema operativo

Verificar existencia o no de vulnerabilidades en cuanto a:

10.1.1. Acceso no autorizado: Información no disponible.

10.1.2. DoS: Información no disponible.

10.1.3. Malware: Información no disponible.

10.1.4. Buffer Overflow: Información no disponible.

10.1.5. Intrusiones: Información no disponible.

10.1.6. Versiones viejas: No se presenta, ya que tiene la versión y el service pack actual del sistema operativo.

10.1.7. Falta de actualizaciones: No se presenta, ya que tiene la versión y el service pack actual del sistema operativo.

10.1.8. Configuración por defecto: Información no disponible.

10.1.9. Exposición de servicios innecesarios: Sólo se encuentran disponibles los servicios HTTP/S, FTP, SMTP y escritorio remoto.

10.1.10. Credenciales predecibles o por defecto: No se presenta.

10.1.11. Credenciales

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.1.12. Cortafuegos (Firewall): La configuración de los cortafuegos es administrada por cada cliente, lo que puede implicar un riesgo, pero la configuración por defecto es segura.

10.1.13. IDS, IPS: Redundancia en sistema de detección de intrusos, por lo que, no hay deficiencia.

10.1.14. Antivirus: Información no disponible.

10.1.15. Versión reciente: Cuenta con la última versión y el service pack actual del sistema operativo.

10.1.16. Actualizaciones y parches automáticas: Información no disponible.

10.1.17. Verificación de credenciales y permisos: Información no disponible.

10.2. ¿Se encuentra en Hospedaje Compartido?

Verificar existencia o no de vulnerabilidades en cuanto a:

10.2.1. Débil autenticación: No se presenta, se necesita usuario y contraseña para acceder a todos los servicios del sistema operativo.

10.2.2. Débil autorización: No se presenta, para cada procesos hay que estar debidamente autenticado.

10.2.3. Revelación de información: Se presenta, los archivos sensibles no son cifrados.

10.2.4. Exposición de archivos delicados (fuentes, ejecutables, archivos configuración): No se presenta.

10.2.5. Configuración por defecto: Información no disponible.

10.2.6. Deficiente definición de permisos de usuario: Información no disponible.

10.2.7. Información delicada no cifrada: Se presenta, los archivos sensibles no son cifrados.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.2.8. Verificación de credenciales y permisos: Existe, se verifica usuario y contraseña para acceder a todos los servicios del sistema operativo.

10.2.9. Cifrado de los datos físicos: No existe.

10.2.10. Establecer permisos estrictos para las aplicaciones y usuarios: Existe, se verifica usuario y contraseña para acceder a todos los servicios del sistema operativo.

10.2.11. Resguardar recursos álgidos: Información no disponible.

10.3. ¿Administración del sistema en forma remota?

Verificar existencia o no de vulnerabilidades en cuanto a:

10.3.1. Deficiente autenticación y autorización: No se presenta, se necesita usuario y contraseña para acceder a todos los servicios del sistema operativo.

10.3.2. Configuraciones por defecto: Información no disponible.

10.3.3. Versiones viejas, falta de actualización o parches: No se presenta, ya que tiene la versión y el service pack actual del sistema operativo.

10.3.4. Deficiencia en canales de comunicación, protocolos no seguros: Se presenta, Uso de protocolo HTTP, usos de protolo FTP, son servicios de texto plano.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.3.5. Canal de comunicación seguro (SSH, SSL, VPN): No existen por defecto hay que pagar extra por ellos.

10.3.6. Políticas de vencimiento y cambio de credenciales: Las credenciales no se vencen, ni se exigen cambios cada cierto tiempo.

10.3.7. Chequeo de configuración desde cero: Información no disponible.

10.3.8. Firewall (lista blanca): Depende de cada cliente, pero la configuración por defecto sólo permite acceso a los puertos antes mencionados.

10.3.9. Verificación de credenciales y permisos: Para utilizar todos los servicios necesitas estar autenticado.

10.4. ¿Actualizaciones remotas?

Verificar existencia o no de vulnerabilidades en cuanto a:

10.4.1. Deficiente autenticación y autorización: Información no disponible.

10.4.2. Deficiencia en canales de comunicación, protocolos no seguros(FTP): Información no disponible.

10.4.3. Accidentes: Información no disponible.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.4.4. Canal de comunicación seguro (SSH, SFTP, VPN): Información no disponible.

10.4.5. Políticas de vencimiento y cambio de credenciales: Información no disponible.

10.4.6. Firewall (lista blanca): Información no disponible.

10.4.7. Verificación de credenciales y permisos: Información no disponible.

10.5. ¿Actualizaciones frecuentes?

Verificar existencia o no de vulnerabilidades en cuanto a:

10.5.1. Reemplazo de información: Información no disponible.

10.5.2. Pruebas insuficientes: Información no disponible.

10.5.3. Accidentes: Información no disponible.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.5.4. Procesos de QA: Información no disponible.

10.5.5. Organización y métodos para actualizaciones: Información no disponible.

10.5.6. Verificación de credenciales y permisos: Si existe, todos los servicios son accedidos utilizando nombre y contraseña.

10.6. ¿Utilizan servidores de aplicación o servidores web?

Verificar existencia o no de vulnerabilidades en cuanto a:

10.6.1. Débil autenticación: Información no disponible.

10.6.2. Configuración por defecto: Información no disponible.

10.6.3. Versiones viejas: No se presenta, tiene la última versión del servidor web.

10.6.4. Falta de actualizaciones o parches: No se presenta, tiene la última versión del servidor web.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.6.5. Versión reciente: Existe.

10.6.6. Actualizaciones y parches: Existe.

10.6.7. Chequeo de configuración desde cero: Información no disponible.

10.6.8. Verificación de credenciales y permisos: Si existe, todos los servicios son accedidos utilizando nombre y contraseña.

10.7. ¿Utilizan servidores de bases de datos?

Verificar existencia o no de vulnerabilidades en cuanto a:

10.7.1. Deficiente respaldo: Información no disponible.

10.7.2. Deficiente autorización: No se presenta, para poder acceder y realizar cualquier acción en el servidor de base de datos se debe estar debidamente autorizado.

10.7.3. Débil autenticación: No se presenta, para poder acceder y realizar cualquier acción en el servidor de base de datos es necesario tener un inicio de sesión y contraseña.

10.7.4. Configuración por defecto: Información no disponible.

10.7.5. Versiones viejas: No se presenta, tiene la última versión del servidor de base de datos.

10.7.6. Falta de actualizaciones o parches: No se presenta, tiene la última versión del servidor de base de datos.

10.7.7. Canal de comunicación no seguro: Se presenta los datos viajan en servicios de texto plano.

10.7.8. Exposición de recursos innecesaria: No se presenta, solo esta disponible acceso a la base de datos de la aplicación.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.7.9. Versión reciente: Existe.

10.7.10. Cifrado de datos: No existe.

10.7.11. Actualizaciones y parches: Existe.

10.7.12. Chequeo de configuración desde cero: Información no disponible.

10.7.13. Verificación de credenciales y permisos: Existe, para poder acceder y realizar cualquier acción en el servidor de base de datos es necesario tener un inicio de sesión y contraseña.

10.8. Utilizan servidores FTP

Verificar existencia o no de vulnerabilidades en cuanto a:

10.8.1. Deficiente respaldo: Información no disponible.

10.8.2. Deficiente autorización: No se presenta, para poder acceder y realizar cualquier acción en el servidor de base de datos se debe estar debidamente autorizado.

10.8.3. Débil autenticación: No se presenta, para poder acceder y realizar cualquier acción en el servidor de base de datos es necesario tener un inicio de sesión y contraseña.

10.8.4. Configuración por defecto: Información no disponible.

10.8.5. Versiones viejas: No se presenta.

10.8.6. Falta de actualizaciones o parches: No se presenta.

10.8.7. Canal de comunicación no seguro: Se presenta, los datos viajan en texto plano.

10.8.8. Exposición de recursos innecesaria: No se presenta, sólo se tiene acceso a la carpeta raíz de la aplicación web.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.8.9. Versión reciente: Existe.

10.8.10. Cifrado de datos: No existe.

10.8.11. Canal comunicación seguro (SSL): No existe por defecto, hay que pagar un extra por este servicio.

10.8.12. Actualizaciones y parches: Existe.

10.8.13. Chequeo de configuración desde cero: Información no disponible.

10.8.14. Verificación de credenciales y permisos: Existe, para poder acceder al servicio es necesario contar con un usuario y su contraseña.

10.9. Uso de tecnologías del lado del servidor para contenido dinámico

Verificar existencia o no de vulnerabilidades en cuanto a:

10.9.1. Abuso de funcionalidad: No se presenta.

10.9.2. Ejecución super usuario: Información no disponible.

10.9.3. Deficiente autorización: Información no disponible.

10.9.4. Configuración por defecto: Información no disponible.

10.9.5. Versiones viejas: No se presenta, tienen la última versión del ASP.

10.9.6. Falta de actualizaciones o parches: No se presenta, tienen la última versión del ASP.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.9.7. Chequeo de configuración desde cero: Información no disponible.

10.9.8. Versión reciente, Actualizaciones y parches: Existe, tienen la última versión del ASP.

10.9.9. Verificación de credenciales y permisos de ejecución: Información no disponible.

11. Verificación y comprobación de seguridad de la **capa de aplicación:**

11.1. Inicio de Sesión de Usuario.

Verificar existencia o no de vulnerabilidades en cuanto a:

11.1.1. Ataques de Fuerza bruta: Se realizaron varios intentos con un login y varios password de forma manual y todas las peticiones las atendía sin ningún tipo de filtro. Si se hace un proceso automatizado en algún momento se podría encontrar un login y password validos para entrar al sistema.

11.1.2. Abuso de funcionalidad: Se podía implementar un robot que registrará múltiples usuarios en pocos segundos.

11.1.3. DoS: No se presenta.

11.1.4. Revelación de información: Al realizar los intentos fallidos de inicio de sesión se produjeron 2 mensajes claros de error.

a) El usuario no se encuentra registrado.

b) El password no coincide.

Por lo que se podría hacer fácilmente una base de datos de usuarios, facilitando aun más un ataque de fuerza bruta.

Figura 7.9: Mensaje de error cuando el password de un usuario no corresponde (Elaboración propia)

Tambien se pudo percibir revelación de información en la base de datos ya que el password se guarda en modo texto, por lo que al vulnerar el servidor de base de datos, fácilmente un atacante podría entrar al sistema con login y password válidos.

Figura 7.10: Campo password guardado en base de datos en modo texto (Elaboración propia)

11.1.5. Inyección de código o comandos: Pueden hacer uso de los formularios presentes en la aplicación para explotar la base de datos.

11.1.6. Phishing: Se presenta.

11.1.7. Inyección de códigos (SQL, HTML, LDAP, entre otros): Pueden hacer uso de los formularios presentes en la aplicación para inyectar todo tipo de códigos.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.1.1. Captcha: No existe.
11.1.2. Filtrado de entrada: No existe.
11.1.3. Escapado de salida: No existe.
11.1.4. Conexiones seguras: No existe.
11.1.5. Manejo de sesión apropiado: No existe.
11.1.6. Enmascarado de información sensible: No existe.
11.1.7. Contador de intentos: No existe.
11.1.8. Registro de conexiones: No existe.
11.1.9. Revisión de comentarios en el código: No existe.
11.1.10. Manejo de excepciones y mensajes apropiado: No existe.
11.1.11. Contraseñas fuertes: No existe.
11.1.12. Educación a usuarios: No existe.

11.2. Recuperación de Contraseña

Verificar existencia o no de vulnerabilidades en cuanto a:

11.2.1. Débil Validación en la Recuperación de Contraseñas: No se presenta ya que la información de inicio de sesión es enviada vía email.
11.2.2. Ataques de Fuerza bruta: Se presenta, la solución puede ser implementar un CAPTCHA.
11.2.3. Abuso de funcionalidad: Se presenta, la solución puede ser implementar un CAPTCHA.
11.2.4. DoS: No se presenta.
11.2.5. Revelación de información: No se presenta.
11.2.6. Inyección de código o comandos: Se presenta, la solución filtrar entrada de datos.
11.2.7. Phishing: Se presenta.
11.2.8. inyección de códigos (SQL, HTML, LDAP, entre otros): Se presenta, la solución filtrar entrada de datos.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.2.9. Captcha: No existe.
11.2.10. Filtrado de entrada: No existe.
11.2.11. Conexiones seguras: No existe.
11.2.12. Manejo de sesión apropiado: No existe.
11.2.13. Enmascarado de información sensible: No existe.
11.2.14. Revisión de comentarios en el código: No existe.
11.2.15. Manejo de excepciones y mensajes apropiado: No existe.

11.3. Administradores Remotos

Verificar existencia o no de vulnerabilidades en cuanto a:

11.3.1. Autenticación insuficiente: Se puedo evidenciar que una vez que un usuario se autenticaba utilizando el formulario, en las demás páginas pertenecientes al administrador remoto no se realizaba una verificación de autenticación del usuario.

Figura 7.11: Ejemplo de la información mostrada al colocar la URL sin autenticar un usuario (Elaboración propia)

Como vemos cualquier atacante podría sin necesidad de conocer un usuario acceder y modificar el contenido de la aplicación Web.

11.3.2. Autorización insuficiente: Se presenta una vez iniciado sesión no se revalidan credenciales.

11.3.3. Robo de Sesión: Se presenta.

11.3.4. XSS: Se pudo evidenciar que en los formularios que se utilizan para administrar el sitio Web no se filtran los datos de entrada ni se escapan los datos de salida, se realizo una prueba de agregar un nuevo origen de nombre colocando en el campo origen lo siguiente: <script >alert("XSS")</script> y este fue el resultado.

Figura 7.12: Vulnerabilidad XSS presente en la aplicación (Elaboración propia)

11.3.5. XSRF: Se presenta.

11.3.6. Fijación de Sesión: Se presenta.

11.3.7. Fuga de Información: No se presenta.

11.3.8. Localización de Recursos Predecibles. Se presenta.

11.3.9. Abuso de Funcionalidad: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.3.10. Autenticación adecuada: No existe.

11.3.11. Verificación de Autorización: No existe.

11.3.12. Renovación de credenciales: No existe.

11.3.13. Certificados digitales: No existe.

11.4. Registro de usuarios en línea

Verificar existencia o no de vulnerabilidades en cuanto a:

11.4.1. Revelación de Información: No se presenta.

11.4.2. Abuso de Funcionalidad: Se presenta, solución implementar CAPTCHA.

11.4.3. Inyección de código SQL: Se presenta, solución filtrar entrada de datos.

11.4.4. Phishing: No se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.4.5. Captcha: No existe.

11.4.6. Filtrado de entrada: No existe.

11.4.7. Escapado de salida: No existe.

11.4.8. Conexiones seguras: No existe.

11.5. Usuarios mantienen sesión

Verificar existencia o no de vulnerabilidades en cuanto a:

11.5.1. Predicción de credenciales/sesión: Se presenta, los identificadores de sesión no son lo suficientemente aleatorios.

11.5.2. Robo de sesión (Session Hijacking): Se presenta.

11.5.3. Fijación de sesión (Session Fixation): Se presenta.

11.5.4. XSS: Se presenta, solución filtrar entrada de datos.

11.5.5. XSRF: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.5.6. Escapado de salida: No existe.

11.5.7. Verificación de autorización: No existe.

11.5.8. Renovación de credenciales: No existe.

11.5.9. Caducidad y vencimiento de sesiones: Existe son configuradas en el servidor web.

11.5.10. Políticas de vencimiento y cambio de contraseñas: No existe.

11.6. Usuarios realizan operaciones

Verificar existencia o no de vulnerabilidades en cuanto a:

11.6.1. Abuso de funcionalidad: Se presenta.

11.6.2. Anti-automatización insuficiente: Se presenta.

11.6.3. DoS: No se presenta.

11.6.4. Repudiación de Operaciones: Se presenta.
11.6.5. Fijación de sesión: Se presenta.
11.6.6. Suplantación de contenido: No se presenta
11.6.7. XSRF: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.6.8. Filtrado de entrada: No existe.
11.6.9. Escapado de salida: No existe.
11.6.10. Token: No existe.
11.6.11. Captcha: No existe.
11.6.12. Renovación de credenciales: No existe.
11.6.13. Verificación de autorización: No existe.
11.6.14. Logs y registro de acciones: No existe.
11.6.15. Notificaciones al usuario: No existe.
11.6.16. Educación a usuarios: No existe.

11.7. Sitios Web que se comunican con Usuarios.

Verificar existencia o no de vulnerabilidades en cuanto a:

11.7.1. Phishing: Se presenta.
11.7.2. Revelación de información: No se presenta.
11.7.3. XSS: Se presenta.
11.7.4. XSRF: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.7.5. Educación a usuarios: No se presenta.
11.7.6. Escapado de la salida: No se presenta.

11.8. Pase de parámetros accesibles al usuario

Verificar existencia o no de vulnerabilidades en cuanto a:

11.8.1. Revelación de Información: No se presenta.
11.8.2. Referencia Directa de objetos: Se presenta.
11.8.3. Autorización insuficiente: No se presenta.
11.8.4. Inyección de código: Se presenta.
11.8.5. XSS: Se presenta.
11.8.6. XSRF: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.8.7. Filtrado de entrada: No existe.
11.8.8. Verificación de autorización: No existe.

11.9. Manipulan información sensible

Verificar existencia o no de vulnerabilidades en cuanto a:

11.9.1. Revelación de Información: Se presenta.
11.9.2. Autorización insuficiente: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.9.3. Cifrado de datos: No existe.

11.9.4. Verificación de autorización: No existe.

11.9.5. Enmascarado de información sensible: Existe.

11.9.6. Autenticidad: Existe.

11.9.7. Certificado digital: No existe.

11.9.8. SSL,VPN: No existe.

11.10. Los usuarios suben archivos al servidor

Verificar existencia o no de vulnerabilidades en cuanto a:

11.10.1. Versiones viejas o sin parchear: No se presenta.

11.10.2. Deficiente autorización: Se presenta.

11.10.3. Accesos no autorizados y violación de privilegios: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.10.4. Antivirus: Información no disponible.

11.10.5. Filtrado de entrada: No existe.

11.10.6. Captcha: No existe.

11.11. La aplicación interactúa con base de datos

Verificar existencia o no de vulnerabilidades en cuanto a:

11.11.1. Inyección de comandos: Se presenta.

11.11.2. Canal de comunicación no seguro: Se presenta.

11.11.3. Revelación de información: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.11.4. Resguardo de credenciales: Existe.

11.11.5. Filtrado de datos: No existe.

11.11.6. Escapado de salida: No existe.

11.11.7. Cifrado de datos: No existe.

11.12. Uso de Plugins (Windows media player, Acrobat, propietarios, .exe, ActiveX, toolbars)

Verificar existencia o no de vulnerabilidades en cuanto a:

11.12.1. Versiones viejas o sin parchear: No se presenta.

11.12.2. Deficiente autorización: Se presenta.

11.12.3. Accesos no autorizados y violación de privilegios: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.12.4. Versiones actuales: Existe.

11.12.5. Parches: Existe.

11.12.6. Exigir últimas versiones: No existe.

11.12.7. Token: No existe.

11.12.8. Verificación de autorización: No existe.

Etapa 3: Comunicación de Resultados

RESUMEN DE INFORME DE AUDITORIA El objetivo de esta auditoría fue determinar las vulnerabilidades que se presentan en la página web www.guiamamaybebe.com.

Durante la realización de esta auditoría estuvieron involucrados directamente el administrador y el desarrollador e indirectamente la empresa de servicio de hosting 1&1.

Esta auditoría tuvo la limitación de no poder obtener toda la información de infraestructura física donde se encuentra alojada la página, ya que la empresa de servicio de hosting no se encuentra residenciada en Venezuela, sino en los Estados Unidos, se tuvo que confiar plenamente en la inormación suministrada en la página web y en las llamadas a soporte.

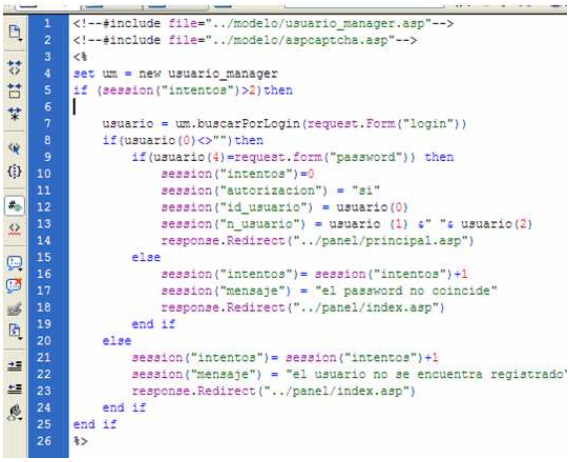
Se pudo determinar múltiples vulnerabilidades a nivel de la capa de aplicación que pueden afectar seriamente el funcionamiento de la aplicación web y la sensiblidad de nuestros datos, para contrarrestar estas vulnerabilidades se proveen las siguientes soluciones:

Para los ataques de fuerza bruta y abuso de funcionalidad:

Solución: Uso de un captcha. Se puede implementar un captcha cuando se producen tres intentos fallidos de inicio de sesión durante un misma sesión.

Figura 7.13: Implementación del captcha luego de realizar 2 intentos fallidos (Elaboración propia)

Para implementarlo se agrego a la aplicación Web un archivo denominado aspcaptcha.asp y se modificaron los siguientes archivos: panel/index.asp,
control/inicio_sesion.asp, global.asa, control/inicio_cliente.asp,
inicio_cliente.asp.



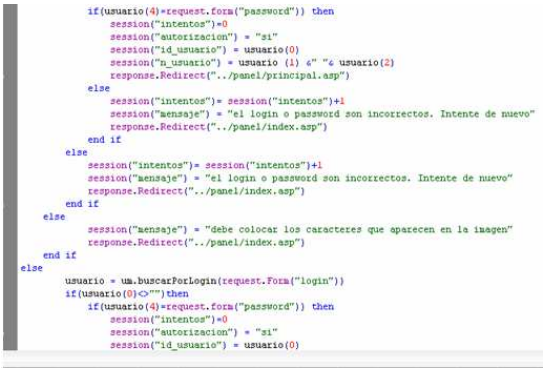
```
<!--#include file="../modelo/usuario_manager.asp"-->
<!--#include file="../modelo/aspcaptcha.asp"-->
<%
set um = new usuario_manager
if (session("intentos")>2)then

    usuario = um.buscarPorLogin(request.Form("login"))
    if(usuario(0)<>"")then
        if(usuario(4)=request.form("password")) then
            session("intentos")=0
            session("autorizacion") = "si"
            session("id_usuario") = usuario(0)
            session("n_usuario") = usuario (1) &" "& usuario(2)
            response.Redirect("../panel/principal.asp")
        else
            session("intentos")= session("intentos")+1
            session("mensaje") = "el password no coincide"
            response.Redirect("../panel/index.asp")
        end if
    else
        session("intentos")= session("intentos")+1
        session("mensaje") = "el usuario no se encuentra registrado'
        response.Redirect("../panel/index.asp")
    end if
end if
%>
```

Figura 7.14: Código fuente de la página index.asp (Elaboración propia)

Para la revelación de información:

Solución: Colocar un mensaje que no revele información o que confunda al atacante. El mensaje que se puede colocar es el siguiente: “el login o password son incorrectos. Intente de nuevo”. Con esta información el atacante no tendrá como diferenciar si es incorrecto el login o el password. Los archivos modificados fueron: control/inicio_sesion.asp, control/inicio_cliente.asp.



```
            if(usuario(4)=request.form("password")) then
                session("intentos")=0
                session("autorizacion") = "si"
                session("id_usuario") = usuario(0)
                session("n_usuario") = usuario (1) &" "& usuario(2)
                response.Redirect("../panel/principal.asp")
            else
                session("intentos")= session("intentos")+1
                session("mensaje") = "el login o password son incorrectos. Intente de nuevo"
                response.Redirect("../panel/index.asp")
            end if
        else
            session("intentos")= session("intentos")+1
            session("mensaje") = "el login o password son incorrectos. Intente de nuevo"
            response.Redirect("../panel/index.asp")
        end if
    else
            session("mensaje") = "debe colocar los caracteres que aparecen en la imagen"
            response.Redirect("../panel/index.asp")
    end if
else
        usuario = um.buscarPorLogin(request.Form("login"))
        if(usuario(0)<>"")then
            if(usuario(4)=request.form("password")) then
                session("intentos")=0
                session("autorizacion") = "si"
                session("id_usuario") = usuario(0)
```

Figura 7.15: Fragmentos de código donde se unifica el mensaje (Elaboración propia)

Solución: Se puede utilizar una función MD5 para enmascarar el campo password, a pesar de que esta función es enormemente utilizada para proceso de verificaciones, y no propiamente como función para encriptador datos ya que no resulta muy robusta para este objetivo, de igual forma proporciona un nivel de seguridad sencillo y que es indispensable para la aplicación. Los archivos modificados fueron:control/agregar_usuario.asp, control/modificar_usuario.asp, control/inicio_sesion.asp.

Figura 7.16: Contraseña guardada en la base de datos aplicandole la función md5 (Elaboración propia)

Para la inyección de códigos o comandos:

Solución: Filtrar las entradas que nos provee un usuario. Esta función es usada para limpiar todas y cada una de las entradas proporcionadas por un usuario cuando realiza una petición http utilizando el método de get o post.

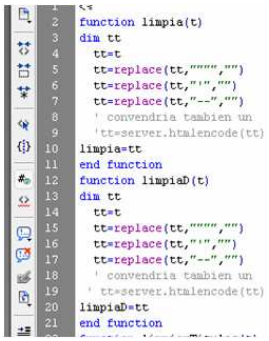


Figura 7.17: Funciones para evitar una inyección de códigos o comando (Elaboración propia)

Para la autenticación insuficiente

Solución: Se creo un archivo en control/sesion.asp que verifica si hay una variable de autorización seteada en "si", si esta variable no existe envía al usuario al panel/index.asp que le muestra el formulario de inicio de sesión.

Figura 7.18: Implementación de la página que verifica la sesión de un usuario (Elaboración propia)

Este archivo es incluido en todos los archivos que se despliegan al usuario administrador cuando el realiza una acción y que se encuentran en el directorio panel y en todos los archivos que ejecutan la acción correspondiente agregar, modificar, eliminar en la base de datos que se encuentran en el directorio control.

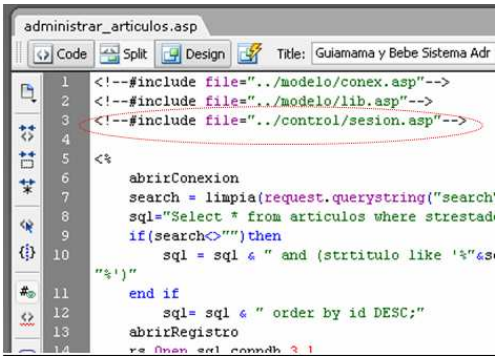


Figura 7.19: Código de panel/administrar_articulos.asp (Elaboración propia)

Para el Cross Site Scripting (XSS)

Solución: Filtrado de entrada de los datos y escapado de la salida de los datos. Se filtraron los datos utilizando la función antes mencionada que se llama limpia, que se encuentra en el archivo modelo/lib.asp
Se escaparon los datos de salida utilizando una función primitiva de ASP denominada server.htmlencode.

Figura 7.20: Resultado obtenido luego de filtrar datos de entrada y escapar datos de salida (Elaboración propia)

Finalmente se pudo constatar que a pesar de que el servicio de hosting tiene una plataforma robusta a nivel físico y de red, la aplicación web posee muchas vulnerabilidades que de no ser resueltas pueden causar graves daños a los datos que la misma maneja y el funcionamiento que esta provee. Es necesario cumplir con las recomendaciones expuestas en este informe, o buscar alternativas que permitan controlar y erradicar las vulnerabilidades. Este informe fue presentado al administrador y al desarrollador de la página para que estuvieran conscientes de los riesgos que actualmente sufre la aplciación.

Etapa 4: Verificación Posterior (Follow-up):

Despues de implementar las recomendaciones dadas en la etapa posterior, en la capa de aplicación, se verificó, que las amenazas y vulnerabilidades presentes por ejemplo, para la inyección de código SQL, fueron subsanadas utilizando la medida de seguridad de filtrado de datos, esta

medida de seguridad tambien sirvió para solucionar problemas de XSS.

De igual forma para que no fuera posible un ataque de fuerza bruta y un abuso de funcionalidad se implementó la recomendación de utilizar un CAPTCHA.

Caso de estudio # 2: www.mat.gob.ve/sectores_agricolas

Etapa 1: Planificación

Describir el sistema

1. El Ministerio del Poder Popular para la Agricultura y Tierras ha desarrollado una aplicación para la administración de los sectores que posee nuestro país, esta información permitirá redefinir la división política territorial de Venezuela la cual va a ser utilizada como base logístico en el desarrollo del VII Censo Nacional Agrícola.

   Cabe destacar que esta aplicación fue realizada para usos internos de este ministerio, y no puede ser accedida por cualquier usuario por lo cual no es ampliamente conocido por los usuarios comunes de Internet, sin embargo, es necesario reforzar la seguridad al tratarse de una aplicación fundamental para el desarrollo de un proyecto nacional.

   La página a nivel de diseño presenta la siguiente disposición:

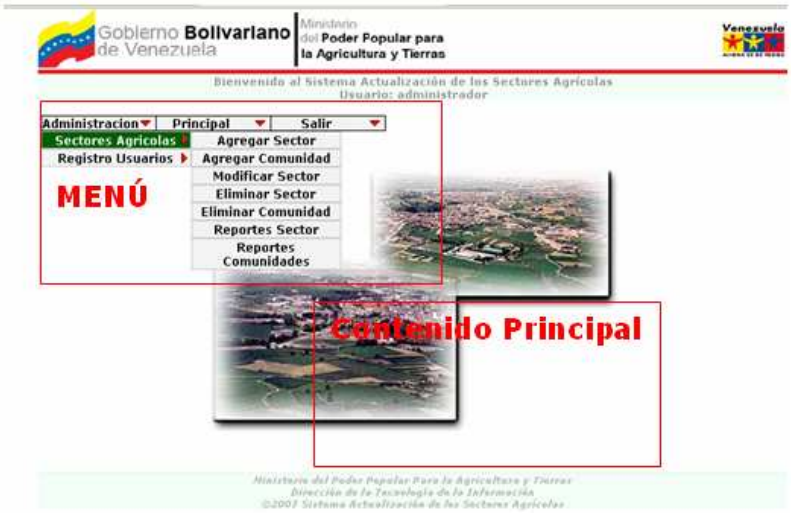


Figura 7.21: Disposición de la página (Elaboración propia)

Todas las páginas que visitan los usuarios, tienen:

a) En la parte superior un encabezado.

b) Seguidamente encontramos un menú con las opciones disponibles para los usuarios.

c) En la parte central se muestra el contenido correspondiente a la sección de la página visitada.

Pasos:

Capa Física

2. Describa lo mejor posible y con el mayor detalle la infraestructura física de los dispositivos y equipos en el sistema, así como las instalaciones físicas, edificio, cuarto de servidores, puertas, etc. Detalle de los equipos servidores, dispositivos de red, cableado, tipo de instalaciones, personal en el sitio, etc. Indique si dicha arquitectura es de administración propia o por parte de terceros. En el caso que sea administrada por terceros, por ejemplo a través de un servicio de hospedaje (hosting), solicitar al proveedor toda la información referente al mismo. Tomar nota de si es un servicio de hosting compartido o de uso exclusivo. Puntos de accesos a las instalaciones, controles de acceso (llaves, claves, cámaras de vigilancia, dispositivos biométricos, vigilantes, registro de novedades), personas autorizadas, vigilantes, otros.

   2.1. ¿Hardware propio de la empresa? Identifique y describa:

      2.1.1. Políticas de acceso: Si existen políticas de acceso, con sistemas de seguridad utilizando un carnet.

      2.1.2. Personal autorizado: A los servidores: Administrador de redes, Administrador de telecomunicaciones, a los demás equipos personal de soporte técnico y usuarios.

      2.1.3. Instalaciones físicas - Edif., habitación, cuarto de servidores: Edificio de 14 pisos, en el piso 6 se encuentra una sala de servidores, que aloja 15 servidores, en todos los pisos hay aproximadamente 90 estaciones de trabajo, 3 impresoras

      2.1.4. Los equipos y dispositivos

      Características en cuanto a:

      a) Servidores: La sala de servidores posee 15 servidores.

      b) Cables para conexiones: UTP nivel 6.

      c) Alimentación (UPS): Posee 2 UPS para todo los equipos de la sala de servidores. Y UPS individuales para equipos de telecomunicaciones y servidores.

   2.2. ¿Hardware arrendado a terceros? (hosting): No aplica.

   2.3. Administración directa en el equipo

      2.3.1. Políticas de acceso: Si existen políticas de acceso, con sistemas de seguridad utilizando un carnet.

      2.3.2. Personal autorizado: A los servidores: Administrador de redes, Administrador de telecomunicaciones, a los demás equipos personal de soporte técnico y usuarios.

      2.3.3. Dispositivos de entrada y salida (CD/DVD, USB, Disco Extraible): Permite respaldar información utilizando USB.

Capa de Red

3. Describa lo mejor posible y con el mayor detalle la arquitectura de red en donde reside la aplicación. Detalles de configuración de la red, identificar servidores, protocolos de red utilizados, existencia de enlaces privados o públicos (VPN, intranet o enlaces dedicados), presencia de Firewall y/o IDS, existencia o no de una DMZ, etc. Realice diagramas, esquemas y tablas.Indique si dicha arquitectura es de administración propia o por parte de terceros. En el caso que sea administrada por terceros, por ejemplo a través de un servicio de hospedaje (hosting), solicitar al proveedor toda la información referente al mismo. Usar herramientas de escaneo de vulnerabilidades automatizado (Nmap, nessus, retina, otros).

   3.1. ¿Administración por terceros? (hosting)

      Si, solicitar información a compañía de hospedaje

   3.2. Identificar los equipos en la red

      3.2.1. Direcciones IP privadas: 192.168.2.0/24 y 172.16.0.0/16.

      3.2.2. Direcciones IP públicas: 200.90.24.240/29, 200.44.191.252/30 y 200.44.42.192/30.

      3.2.3. Equipos en la red local: Posee 15 servidores instalados, Posee 1 router Cisco 3725, 2 routers Cisco 2821 y 1 Cisco 2800.

      3.2.4. Equipos remotos: 22 routers son de las UEMAT con 22 switches.

   3.3. Identificar dispositivos de firewall

      3.3.1. Filtrado lista blanca o lista negra: listas negras.

      3.3.2. Políticas de seguridad: No se puede utilizar tráfico P2P, ningún tipo de mensajería instántanea y páginas muy pesadas que no sean necesarias para las labores del empleado público.

      3.3.3. Segmentación de redes: Por cada piso hay un VLAN.

   3.4. IDS/IPS: Poseen sistema de detección de intrusos.

   3.5. Identificar protocolos de red usados: utilizan IP.

   3.6. Esquematizar la arquitectura de red (hacer uso de técnicas de modelado)

      3.6.1. Diagramas UML de componentes especializados para redes

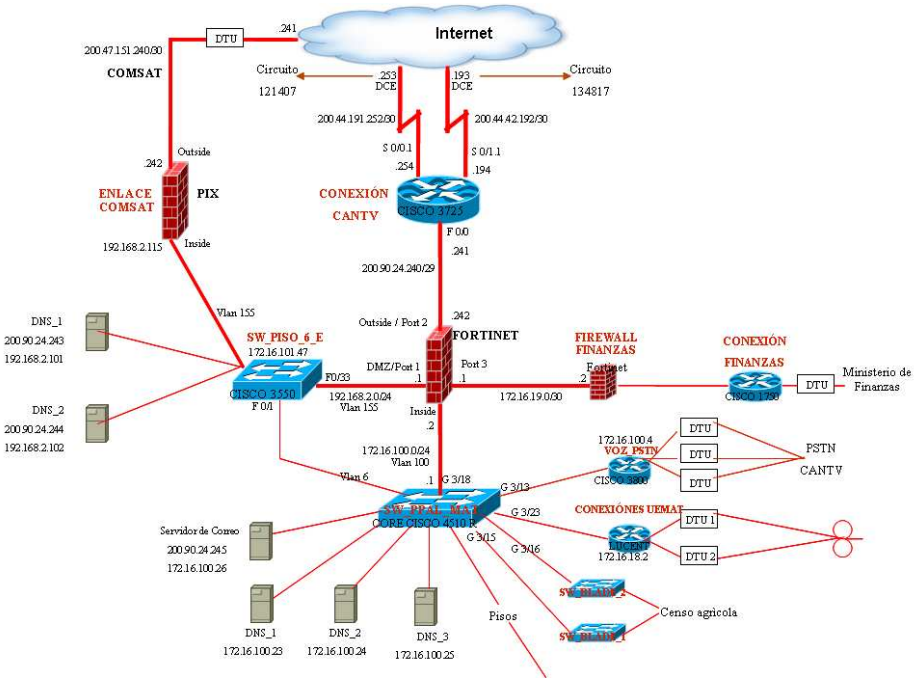


Figura 7.22: Diagrama de la red

3.7. Servidores de DNS privados y públicos: Públicos los de CANTV y privados 192.168.2.101 y la pública de la privada es 200.90.24.243.

Capa de Sistema Operativo y Servicios

4. Describa lo mejor posible y con el mayor detalle las características de los equipos en cuanto a, el sistema operativo que usan, el sistemas de archivos y esquemas de permisologías existentes, servicios disponibles, usuarios del sistema, ubicación de los recursos en los sistemas.

   4.1. Sistema operativo: Debian.

      4.1.1. Distribución del SO: etch y suse.

      4.1.2. Versión: 4.0 R3 y 10.0 versión para servidores.

      4.1.3. Usuarios y Grupos: root/root, www/www, postgre/postgre, apache/apache, tomcat/tomcat, oracle/oracle.

      4.1.4. Usuarios y Grupos administradores: root/root.

   4.2. Servicios

      4.2.1. Servidores de aplicaciones: Apache, Tomcat.

      4.2.2. Servidores de bases de datos: PostgreSQL, Oracle.

      4.2.3. Servidores de correo: Servidor de entrada de mail (IMAP): mail.mat.gob.ve.
      Servidor de entrada de mail (POP): mail.mat.gob.ve.
      Servidor de salida de mail (SMTP): mail.mat.gob.ve.

      4.2.4. Servidores LDAP: No disponible.

      4.2.5. Servidores FTP: Disponible.

      4.2.6. Otros servicios: SSH, FTP seguro, Administración remota.

   4.3. Para cada servicio

Apache.

4.3.1. Versión: 2.0.

4.3.2. Usuarios y permisos de ejecución: apache/apache para labores de mantenimiento.

4.3.3. Archivos y directorios relacionados: /etc/apache2/ archivos ejecutables, /etc/usr/bin/ archivos binarios.

4.3.4. Puertos que utilizan: puerto 80.

4.3.5. Cuentas de usuarios del servicio: apache.

4.3.6. Archivos de configuración, propiedades: Todos los archivos de configuración del apache están en /etc/apache2/.

Tomcat.

4.3.1. Versión: 5.5.15.

4.3.2. Usuarios y permisos de ejecución: tomcat/tomcat para labores de administración.

4.3.3. Archivos y directorios relacionados: /usr/local/apache/tomcat archivos ejecutables, /usr/local/apache/tomcat/bin/.

4.3.4. Puertos que utilizan: puerto 8080.

4.3.5. Cuentas de usuarios del servicio: tomcat, y cada programador puede crear una cuenta.

4.3.6. Archivos de configuración, propiedades: Todos los archivos de configuración del tomcat están en /usr/local/apache/tomcat.

Postgre SQL.

4.3.1. Versión: 8.3.

4.3.2. Usuarios y permisos de ejecución: postgre/postgre.

4.3.3. Archivos y directorios relacionados: /etc/postgre8.3 archivos ejecutables, /etc/usr/bin archivos binarios.

4.3.4. Puertos que utilizan: 5432.

4.3.5. Cuentas de usuarios del servicio: postgre.

4.3.6. Archivos de configuración, propiedades: Todos los archivos de configuración del postgre están en /etc/postgre8.3 .

Oracle.

4.3.1. Versión: 10g.

4.3.2. Usuarios y permisos de ejecución: oracle/oracle.

4.3.3. Archivos y directorios relacionados: tiene 10 directorios, en 3 directorios se encuentran los controlfile, en 2 se encuentran los REDO, en un directorio el datafile, un directorio para el swap, un directorio para las aplicaciones y 2 para indexación de los datos.

4.3.4. Puertos que utilizan: 1521.

4.3.5. Cuentas de usuarios del servicio: oracle.

4.3.6. Archivos de configuración, propiedades: Todos los archivos de configuración del oracle están en /opt/oracle/.

4.4. Sistema de archivos

4.4.1. Ubicación y contexto de las aplicaciones: Los archivos ejecutables de las aplicaciones se consiguen en /etc/.

4.4.2. Identifique recursos compartidos: No disponible.

4.4.3. Identifique directorios protegidos: No disponible los permisos son definidos por cada usuario, para cada directorio que el utilice.

4.4.4. Ruta de instalación de servicios: /etc/.

4.5. Administración remota

4.5.1. A través de red interna: Disponible.

4.5.2. VPN: Disponible.

4.5.3. SSH: Disponible.

4.5.4. Sistemas de Administración Remota: Disponible.

5. Identifique las aplicaciones o módulos presentes en la aplicación web. Para cada una de estos elementos determinar como mínimo el nombre de aplicación o módulo, versión, ubicación, permisos de ejecución relacionados (si aplica). Algunas tecnologías comúnmente relacionadas con las aplicaciones web son: servidores de aplicaciones o servidores web (Apache, IIS, Tomcat...), servidores de base de datos, tecnologías del lado del servidor (Java, ASP .Net, PHP, Ruby...), entre otras.

5.1. Identificar servidores de aplicaciones

5.1.1. Servidor Web: Apache.

5.1.2. Versión: 2.0.

5.1.3. Permisos de ejecución: Para usuario apache.

5.1.4. Accesos y permisos de directorios: usuario apache tiene permisos sobre /etc/apache2/.

5.1.5. Usuarios o cuentas de administración: root/root y apache/apache.

5.1.6. Módulos instalados o librerías: mod_rewrite.

5.2. Servidores de base de datos

5.2.1. Servidor de base de datos: PostgreSQL.

5.2.2. Versión: 8.3.

5.2.3. Ubicación: /etc/postgre8.3/

5.2.4. Permisos de ejecución: Para usuario postgre.

5.2.5. Accesos y permisos de directorios: postgre sólo a /etc/postgre8.3/.

5.2.6. Usuarios o cuentas de administración y usuarios: Usuario: postgre.

5.2.7. Acceso

a) Acceso Local: Disponible.

b) Acceso Remoto: No Disponible.

5.2.8. Cifrado de archivos físicos: No disponible.

5.3. Servidores de Correo: No disponible para la aplicación

5.4. Servidores LDAP: No disponible.

5.5. Tecnologías del lado del servidor

5.5.1. Identificar las tecnologías usadas: PHP.

5.5.2. Versión: 5.0

5.5.3. Permisos de ejecución: No disponible.

5.5.4. Permisos de directorios: permisos de lectura y escritura /etc/php.

Capa de Aplicación

6. Describa lo mejor posible y con el mayor detalle las tecnologías usadas en las aplicaciones web. Identificar protocolos de comunicación (HTTP, HTTPS, FTP, LDAP...), uso de Frameworks (Struts, Spring, Cakephp, phpNuke, Ruby on Rails...) y los patrones de diseño (Composite view, MVC) que utilizan, tecnologías del lado del cliente (HTML, CSS, Javascript, AJAX, Flash, Plugins como Windows Media Player o Acrobat Reader, entre otros...), archivos y recursos de la aplicación, acceso a datos (servidores de base de datos, SAN, sistemas de archivos), servicios web.

6.1. Identificar protocolos de comunicación

6.1.1. HTTP: Disponible.

6.1.2. HTTPS (HTTP y SSL). Certificados digitales: No disponible.

6.1.3. SMTP: No Disponible.

6.1.4. FTP: No disponible

6.1.5. DNS: Disponible.

6.2. Determinar el uso de uno o más Frameworks. Conocer o investigar el Framework sobre el cual están hechas las aplicaciones ayuda a una mejor comprensión de sus funcionalidades y procesos, desde su estructura y organización de archivos y recursos hasta el código fuente e implementación.

6.2.1. Versión: Framework propio del desarrollador.

6.2.2. Librerías y complementos: No disponible.

6.2.3. Patrones: MVC, Composite View.

6.2.4. Estructura de la aplicación

a) Ubicación de archivos fuentes: /var/www/sectores_agricolas/.

b) Ubicación de archivos de registro (logs): /var/logs/.

c) Ubicación de archivos de configuración: /var/www/sectores_agricolas/.

d) Ubicación de archivos públicos (paginas, scripts, imágenes, css, multimedia, etc.): /var/www/sectores_agricolas/.

e) Ubicación de archivos subidos al servidor: No Aplica.

6.3. Tecnologías del lado del cliente involucradas

6.3.1. HTML (XML, CSS): Disponible HTML y CSS.

6.3.2. Javascript, VBscript: Disponible Javascript.

6.3.3. AJAX: Disponible.

6.3.4. Complementos o Plugins: No disponible.

6.3.5. Complementos propios: No disponible.

6.3.6. Cookies: Disponible.

6.4. Acceso a datos

6.4.1. Acceso a base de datos

a) Ubicación de los recursos: /etc/postgre8.3/data/.
b) Almacenamiento de credenciales de acceso: No Disponible.
c) Conexión segura: Disponible.
d) Cifrado de datos: No Disponible.

6.4.2. Acceso a archivos

a) Permisología sobre los recursos: Usuario www/www tiene permisos de escritura, lectura y ejecución del directorio /var/www.
b) Recursos locales: /var/www/.
c) Recursos remotos: No disponible.
d) Cifrado de archivos: No disponible.

6.5. Acceso a servicios.

6.5.1. Servicios Web: No aplica.

7. Determinar funcionalidades o características que existen en la aplicación web. Identificando características como registro de usuario, inicio de sesión, uso de certificado digital o no, foros, formas para recuperación de contraseñas, modulo de administración, administradores de contenido, capacidades para subir archivos al servidor, ejecución de comandos, inclusión de archivos en el servidor, accesos a recursos remotos tales como otras aplicaciones, servicios web o recursos como imágenes, hojas de estilos entre otros.

7.1. En cada paso realizar:

Registro de clientes nuevos.

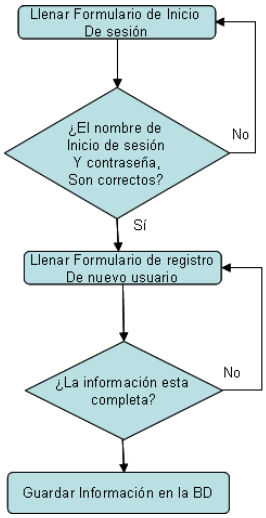


Figura 7.23: Registro de clientes nuevos. (Elaboración propia)

Llenar información de contenido de la página.

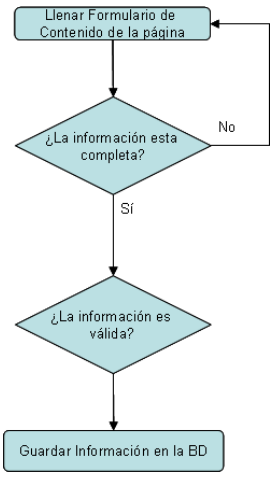


Figura 7.24: Llenar información de contenido de la página. (Elaboración propia)

7.2. Identificar procesos de autenticación

7.2.1. Inicio de sesión con formularios web: Disponible.

7.2.2. Certificados digitales: No disponible.

7.2.3. Sistemas de recuperación de contraseña: No Disponible.

7.3. Identificar operaciones. Procesos de compra, transacciones, agregar contenido, registros, etc.

7.4. Módulos de administración

7.4.1. Administradores de contenido: Disponible.

7.4.2. Administradores de consulta y operaciones: No disponible.

7.5. Procesos con capacidades de:

7.5.1. Subir archivos al servidor: No Disponible.

7.5.2. Entradas de usuario: Disponible.

7.5.3. Envió de mensajes: No Disponible.

7.6. Secciones o procesos que usan o exigen uso de javascript: Disponible: La validación de todos los formularios se hace usando javascript.

7.7. Secciones o procesos que usan o exigen uso de complementos: No disponible.

Etapa 2: Verificación y comprobación

8. Verificación y comprobación de seguridad de **Capa Física**:

8.1. Hardware propio de la empresa

Verificar existencia o no de vulnerabilidades en cuanto a:

8.1.1. Accesos no autorizados: A pesar de que el acceso a sala de servidores está restringido por el uso de una clave, cualquier persona que conozca al que concede los permisos, los puede tener.

8.1.2. Catástrofes naturales: Se pueden presentar, ya que no se realizan respaldos en otras localidades físicas, lo que representa un gran riesgo a nivel de recuperación de los datos.

8.1.3. Accidentes: Se pueden presentar ya que no hay un buen control del acceso a los mismos.

8.1.4. Fallas: No se presentan.

8.1.5. Deficientes políticas de acceso: Se presentan, no se lleva control riguroso de las personas que entran a los servidores, tienen mucha confianza en la buena bondad de los empleados.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

8.1.6. Políticas de acceso estrictas

a) Registro de eventos y control de acceso: Existe, hay un programa que genera estos reportes, pero los mismos no son utilizados.
b) Guardias de seguridad: No existe.
c) Libro de novedades: No existe.
d) Puertas cerradas: Existe, la puerta se abre con el uso del carnet de personas autorizadas.
e) Llaves o Claves: Existe, con carnet.
f) Dispositivos de seguridad biométricos: No existe.
g) Cámaras: Existe.

8.1.7. Continuidad del negocio

a) Redundancia: No existe.
b) Corriente ininterrumpida (UPS): Existe, Hay 2 UPS en la sala de servidor, y uno por cada servidor.

8.1.8. Respaldo: Se realizan respaldos diarios y semanales de todos los sistemas y todos los datos.

8.1.9. Detección y Corrección de Fallas

a) Control de temperatura y humedad: Existe, hay un termómetro que mide la temperatura.
b) Detectores y alarmas de: calor,humo y humedad: Existe, hay Sistema de detección de calor.
c) Sistema de extinción de incendios: Existe, hay un sistema que se activa al detectar el calor y hay bombonas para extinguir el fuego.

8.2. Hardware arrendado a terceros: No aplica.

8.3. Administración directa en el equipo

Verificar existencia o no de vulnerabilidades en cuanto a:

8.3.1. Accesos no autorizados: Se presenta, no hay riguroso contro sobre los accesos no autorizados.

8.3.2. Copia de información a través de dispositivos removibles: Se presenta la información de los equipos puede ser copiada a un pen drive vía USB.

8.3.3. Accidentes: Se presenta, al no controlar la entrada de las personas a la sala, cualquier accidente tipo desenchufar algún dispositivo puede suceder.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

8.3.1. Políticas de acceso estrictas

a) Registro de eventos y control de acceso: Existe, hay un programa que genera estos reportes, pero los mismos no son utilizados.

b) Guardias de seguridad: No existe.

c) Libro de novedades: No existe.

d) Puertas cerradas: Existe, todas las puertas de cada piso se abren con el uso de carnet, cada carnet tiene permisos para abrir solo ciertas puertas.

e) Llaves o Claves: Existe, todas los accesos de cada piso se abren con el uso de carnet, cada carnet tiene permisos para abrir solo ciertas puertas.

f) Dispositivos de seguridad biométricos: No existe.

g) Cámaras: Existe, hay cámaras de seguridad en los almacenes y en la sala de servidores.

8.3.2. Continuidad del negocio

a) Redundancia: No existe.

b) Corriente ininterrumpida (UPS): No existe.

8.3.3. Respaldo: El respaldo de la información depende de cada usuario.

8.3.4. Detección y Corrección de Fallas

a) Control de temperatura y humedad: No existe.

b) Detectores y alarmas de: calor,humo y humedad: Existe, hay Sistema de detección de calor.

c) Sistema de extinción de incendios: Existe, hay un sistema que se activa al detectar el calor y hay bombonas para extinguir el fuego.

9. Verificación y comprobación de seguridad de la **Capa de Red:**

9.1. Equipos de acceso público a través de la red

Verificar existencia o no de vulnerabilidades en cuanto a:

9.1.1. Deficiente protección de los datos en el tránsito: Se presenta, ya que los datos viajan en texto plano.

9.1.2. Suplantación de IP (IP spoofing): No se presenta.

9.1.3. Suplantación de una página web real: No se presenta.

9.1.4. Denegación de Servicio (DoS): No se presenta, ya que se tiene un cortafuegos a nivel de hardware, que permite detectar intrusos.

9.1.5. Malware (gusanos, virus): Existe, se comprobo que hasta uno de los servidores tenía un virus, y múltiples estaciones de trabajo también presentaban este problema.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

9.1.6. Firewall

a) Modelo de seguridad positivo (whitelist): Es una opción del firewall sin embargo se utiliza, un filtrado por lista negra.

b) Filtro por direcciones IP: Se presenta.

c) Bloquear puertos innecesarios o privados: Acceso sólo a los puertos permitidos para ejecutar las aplicaciones.

9.1.7. Antivirus: Se presenta, todos los equipos tienen McAfee instalado.

9.1.8. DMZ: Se presenta, esta configurado una DMZ.

9.1.9. IDS/IPS: Se presenta, el cortafuegos tiene Sistema de detección de intrusos.

9.1.10. VPN: Se presenta, hay una VPN para comunicarse con los organismos adscritos en cada estado.

9.1.11. SSL/TLS: Se presenta si es requerido, no viene por defecto.

9.1.12. Balanceo de Carga: Información no disponible.

9.2. Equipos de acceso privado

Verificar existencia o no de vulnerabilidades en cuanto a:

9.2.1. Deficiente protección de los datos en el tránsito: Se presenta, ya que los datos viajan en texto plano.

9.2.2. Suplantación de IP (IP spoofing): No se presenta.

9.2.3. Suplantación de una página web real: No se presenta.

9.2.4. Denegación de Servicio (DoS): No se presenta.

9.2.5. Malware (gusanos, virus): Información no disponible.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

9.2.6. Firewall

a) Modelo de seguridad positivo (whitelist): Es una opción del firewall sin embargo se utiliza, un filtrado por lista negra.

b) Filtro por direcciones IP: Se presenta.

c) Bloquear puertos innecesarios o privados: Acceso sólo a los puertos permitidos para ejecutar las aplicaciones.

9.2.7. Antivirus: Se presenta, todos los equipos tienen McAfee instalado.

9.2.8. DMZ: Se presenta, esta configurado una DMZ.

9.2.9. IDS/IPS: Se presenta, el cortafuegos tiene Sistema de detección de intrusos.

9.2.10. VPN: Se presenta, hay una VPN para comunicarse con los organismos adscritos en cada estado.

9.2.11. SSL/TLS: Se presenta si es requerido, no viene por defecto.

9.2.12. Balanceo de Carga: Información no disponible.

10. Verificación y comprobación de seguridad la **capa de Sistema Operativo y servicios:**

10.1. Sistema operativo

Verificar existencia o no de vulnerabilidades en cuanto a:

10.1.1. Acceso no autorizado: Sólo se permite el acceso al grupo de usuarios antes mencionado.

10.1.2. DoS: No se presenta.

10.1.3. Malware: No se presenta.

10.1.4. Buffer Overflow: No se presenta.

10.1.5. Intrusiones: No se presenta.

10.1.6. Versiones viejas: No se presenta, ya que tiene la última versión estable de esa distribución.

10.1.7. Falta de actualizaciones: No existe política de actualización.

10.1.8. Configuración por defecto: Se presenta, existen configuraciones por defecto por ejemplo del SSH.

10.1.9. Exposición de servicios innecesarios: Se presenta, hay servicios que no se utilizan y están instalados.

10.1.10. Credenciales predecibles o por defecto: No se presenta.

10.1.11. Credenciales

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.1.12. Cortafuegos (Firewall): Existe.

10.1.13. IDS, IPS: Existe.

10.1.14. Antivirus: Existe, pero hay deficiencias.

10.1.15. Versión reciente: No existe, no todos los sistemas tienen las últimas versiones.

10.1.16. Actualizaciones y parches automáticas: No existe, no todos los sistemas tienen las últimas versiones.

10.1.17. Verificación de credenciales y permisos: Existe, en todos los servicios.

10.2. ¿Se encuentra en Hospedaje Compartido?: No aplica.

10.3. ¿Administración del sistema en forma remota?

Verificar existencia o no de vulnerabilidades en cuanto a:

10.3.1. Deficiente autenticación y autorización: No se presenta, se necesita usuario y contraseña para acceder a todos los servicios del sistema operativo.

10.3.2. Configuraciones por defecto: No se presenta.

10.3.3. Versiones viejas, falta de actualización o parches: No se presenta, ya que tiene la última versión.

10.3.4. Deficiencia en canales de comunicación, protocolos no seguros: No se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.3.5. Canal de comunicación seguro (SSH, SSL, VPN): Existe, hay uso de SSL y VPN.

10.3.6. Políticas de vencimiento y cambio de credenciales: Existe, las contraseñas se deben cambiar cada cierto tiempo, y no se pueden reutilizar contraseñas anteriores.

10.3.7. Chequeo de configuración desde cero: No existe.

10.3.8. Firewall (lista blanca): Existe pero se hace filtrado por lista negra.

10.3.9. Verificación de credenciales y permisos: Para utilizar todos los servicios necesitas estar autenticado.

10.4. ¿Actualizaciones remotas?: No aplica

10.5. ¿Actualizaciones frecuentes?: No aplica

10.6. ¿Utilizan servidores de aplicación o servidores web?: Sí

Verificar existencia o no de vulnerabilidades en cuanto a:

10.6.1. Abuso de funcionalidad: No se presenta.

10.6.2. Débil autenticación: No se presenta.

10.6.3. Configuración por defecto: Se presenta.

10.6.4. Versiones viejas: No se presenta, se cuenta con la última versión.

10.6.5. Falta de actualizaciones o parches: No se presenta, tiene la última versión del servidor web.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.6.6. Versión reciente: Existe.

10.6.7. Actualizaciones y parches: Existe.

10.6.8. Chequeo de configuración desde cero: No existe.

10.6.9. Verificación de credenciales y permisos: Si existe, todos los servicios son accedidos utilizando nombre y contraseña.

10.7. ¿Utilizan servidores de bases de datos?: Sí

Verificar existencia o no de vulnerabilidades en cuanto a:

10.7.1. Abuso de funcionalidad: No se presenta.

10.7.2. Deficiente respaldo: No se presenta.

10.7.3. Deficiente autorización: No se presenta, para poder acceder y realizar cualquier acción en el servidor de base de datos se debe estar debidamente autorizado.

10.7.4. Débil autenticación: No se presenta, para poder acceder y realizar cualquier acción en el servidor de base de datos es necesario tener un inicio de sesión y contraseña.

10.7.5. Configuración por defecto: No se presenta.

10.7.6. Versiones viejas: No se presenta, tiene la última versión del servidor de base de datos.

10.7.7. Falta de actualizaciones o parches: No se presenta, tiene la última versión del servidor de base de datos.

10.7.8. Canal de comunicación no seguro: Se presenta los datos viajan en servicios de texto plano.

10.7.9. Exposición de recursos innecesaria: No se presenta, solo esta disponible acceso a la base de datos de la aplicación.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.7.10. Versión reciente: Existe.

10.7.11. Cifrado de datos: No existe.

10.7.12. Actualizaciones y parches: No Existe.

10.7.13. Chequeo de configuración desde cero: No existe.

10.7.14. Verificación de credenciales y permisos: Existe, para poder acceder y realizar cualquier acción en el servidor de base de datos es necesario tener un inicio de sesión y contraseña.

10.8. Utilizan servidores FTP? No

10.9. Uso de tecnologías del lado del servidor para contenido dinámico

Verificar existencia o no de vulnerabilidades en cuanto a:

10.9.1. Abuso de funcionalidad: No se presenta.

10.9.2. Ejecución super usuario: No se presenta.

10.9.3. Deficiente autorización: No se presenta.

10.9.4. Configuración por defecto: No se presenta.

10.9.5. Versiones viejas: No se presenta, tienen la última versión del PHP.

10.9.6. Falta de actualizaciones o parches: No se presenta, tienen la última versión del PHP.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.9.7. Chequeo de configuración desde cero: No se realiza.

10.9.8. Versión reciente, Actualizaciones y parches: Existe, tienen la última versión del PHP, sin embargo no existe política de actualizaciones.

10.9.9. Verificación de credenciales y permisos de ejecución: Existe.

11. Verificación y comprobación de seguridad de la **capa de aplicación:**

11.1. Inicio de Sesión de Usuario.

Verificar existencia o no de vulnerabilidades en cuanto a:

11.1.1. Ataques de Fuerza bruta: Se realizaron varios intentos con un login y varios password de forma manual y todas las peticiones las atendía sin ningún tipo de filtro. Si se hace un proceso automatizado en algún momento se podría encontrar un login y password validos para entrar al sistema.

11.1.2. Abuso de funcionalidad: Se podía implementar un robot que registrará múltiples usuarios en pocos segundos.

11.1.3. DoS: No se presenta.

11.1.4. Revelación de información: Al realizar los intentos fallidos de inicio de sesión se produjeron 2 mensajes claros de error.

a) El usuario no se encuentra registrado en la base de datos.

b) Contraseña incorrecta.

Por lo que se podría hacer fácilmente una base de datos de usuarios, facilitando aun más un ataque de fuerza bruta.

Figura 7.25: Mensaje de error cuando el usuario no se encuentra en la base de datos (Elaboración propia)

En la base de datos se percibe que al password se le esta aplicando un md5, por lo que al vulnerar la BD se complica la lectura de un password

password character varying(100)
e10adc3949ba59abbe56e057f20f883e
edd7939f717802e1ec2f75210cd82e6a
edd7939f717802e1ec2f75210cd82e6a
c33367701511b4f6020ec61ded352059
8bc531f0ae1e813230e34a78a36be8c2
c4661f6669f5f35823f39acb6d6be725

Figura 7.26: Valores del campo password en la base de datos (Elaboración propia)

11.1.5. Inyección de código o comandos: Este tipo de vulnerabilidad esta controlada por una función nativa de php denominada pg_escape_string, la cual se encarga de limpiar los valores introducidos por el usuario, esa función es utilizada en todos los sql que procesará el manejador de la base de datos.

```
function buscarPorLogin($login){
    $usuario = "";
    $bd = new bd_manager();
    $enlace = $bd->conectar();
    $sql = "SELECT * FROM usuarios WHERE activo='t' and login='".pg_escape_string($login)."'";
    $result = pg_query($enlace,$sql) or die("Error - buscarPorLogin");
    $bd->desconectar();
    if(pg_num_rows($result)!=0){
        $usuario = $this->construir($result);
    }
    return $usuario;
}
```

Figura 7.27: Uso de la función nativa de PHP pg_escape_string() (Elaboración propia)

11.1.6. Phishing: Se presenta.

11.1.7. Inyección de códigos (SQL, HTML, LDAP, entre otros): No se presenta por el uso de la nativa de PHP pg_escape_string.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.1.8. Captcha: No existe.

11.1.9. Filtrado de entrada: Existe.

11.1.10. Escapado de salida: No existe.

11.1.11. Conexiones seguras: Existe.

11.1.12. Manejo de sesión apropiado: No existe.

11.1.13. Enmascarado de información sensible: Existe.

11.1.14. Contador de intentos: No existe.

11.1.15. Registro de conexiones: No existe.

11.1.16. Revisión de comentarios en el código: No existe.

11.1.17. Manejo de excepciones y mensajes apropiado: No existe.

11.1.18. Contraseñas fuertes: No existe.

11.1.19. Educación a usuarios: No existe.

11.2. Recuperación de Contraseña: No aplica.

11.3. Administradores Remotos

Verificar existencia o no de vulnerabilidades en cuanto a:

11.3.1. Autenticación insuficiente: Se pudo evidenciar que no se estaba verificando la sesión, en todas las páginas pertenecientes a la aplicación. En la siguiente imagen observamos el formulario para agregar un nuevo usuario, colocando directamente el URL: http://www.mat.gob.ve/sectores_agricolas /vistas/usuarios/agregar_usuarios.php

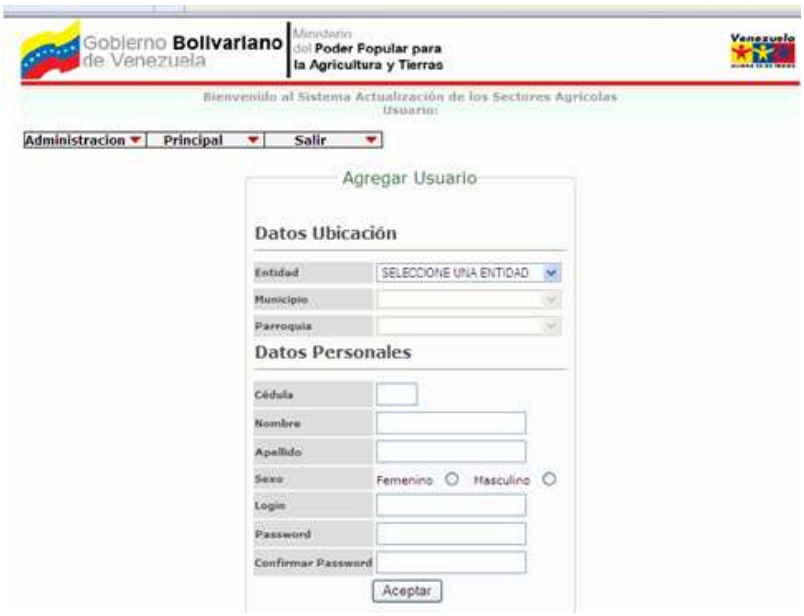


Figura 7.28: Formulario para agregar usuarios (Elaboración propia)

Un atacante fácilmente sin necesidad de iniciar sesión puede crear un usuario valido, y este tiene todos los privilegios para administrar toda la información de sectores.

11.3.2. Autorización insuficiente: Se presenta una vez iniciado sesión no se revalidan credenciales.

11.3.3. Robo de Sesión: Se presenta.

11.3.4. XSS: Como se comento anteriormente todos los datos que proporcionan los usuarios son limpiados mediante la función pg_escape_string. Sin embargo si la base de datos es vulnerada los datos que se muestran no estan siendo escapados esto se pudo evidenciar al colocar el siguiente nombre de usuario en la base de datos: <script>alert("XSS")</script>, cuando se recargo la página este fue el resultado.

Figura 7.29: Vulnerabilidad XSS presente en la aplicación (Elaboración propia)

11.3.5. XSRF: Se presenta.
11.3.6. Fijación de Sesión: Se presenta.
11.3.7. Fuga de Información: No se presenta.
11.3.8. Localización de Recursos Predecibles. Se presenta.
11.3.9. Abuso de Funcionalidad: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.3.10. Captcha: No existe.
11.3.11. Filtrado de entrada: Existe.
11.3.12. Conexiones seguras: No existe.
11.3.13. Manejo de sesión apropiado: No existe.
11.3.14. Enmascarado de información sensible: No existe.
11.3.15. Revisión de comentarios en el código: No existe.
11.3.16. Manejo de excepciones y mensajes apropiado: No existe.

11.4. Registro de usuarios en línea: No aplica.

11.5. Usuarios mantienen sesión

Verificar existencia o no de vulnerabilidades en cuanto a:

11.5.1. Predicción de credenciales/sesión: Se presenta, los identificadores de sesión no son lo suficientemente aleatorios.
11.5.2. Robo de sesión (Session Hijacking): Se presenta.
11.5.3. Fijación de sesión (Session Fixation): Se presenta.
11.5.4. XSS: Se presenta, solución filtrar entrada de datos.
11.5.5. XSRF: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.5.6. Escapado de salida: No existe.
11.5.7. Verificación de autorización: No existe.
11.5.8. Renovación de credenciales: No existe.
11.5.9. Caducidad y vencimiento de sesiones: Existe son configuradas en el servidor web.
11.5.10. Políticas de vencimiento y cambio de contraseñas: No existe.

11.6. Usuarios realizan operaciones

Verificar existencia o no de vulnerabilidades en cuanto a:

11.6.1. Abuso de funcionalidad: Se presenta.
11.6.2. Anti-automatización insuficiente: Se presenta.
11.6.3. DoS: No se presenta.
11.6.4. Repudiación de Operaciones: Se presenta.
11.6.5. Fijación de sesión: Se presenta.
11.6.6. Suplantación de contenido: No se presenta
11.6.7. XSRF: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.6.8. Filtrado de entrada: Existe.
11.6.9. Escapado de salida: No existe.
11.6.10. Token: No existe.
11.6.11. Captcha: No existe.
11.6.12. Renovación de credenciales: No existe.
11.6.13. Verificación de autorización: No existe.
11.6.14. Logs y registro de acciones: No existe.
11.6.15. Notificaciones al usuario: No existe.
11.6.16. Educación a usuarios: No existe.

11.7. Sitios Web que se comunican con Usuarios: No aplica.

11.8. Pase de parámetros accesibles al usuario

Verificar existencia o no de vulnerabilidades en cuanto a:

11.8.1. Revelación de Información: No se presenta.
11.8.2. Referencia Directa de objetos: Se presenta.
11.8.3. Autorización insuficiente: No se presenta.
11.8.4. Inyección de código: No se presenta.
11.8.5. XSS: Se presenta.
11.8.6. XSRF: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.8.7. Filtrado de entrada: Existe.
11.8.8. Verificación de autorización: No existe.

11.9. Manipulan información sensible

Verificar existencia o no de vulnerabilidades en cuanto a:

11.9.1. Revelación de Información: No se presenta.
11.9.2. Autorización insuficiente: Se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.9.3. Cifrado de datos: Existe.
11.9.4. Verificación de autorización: No existe.
11.9.5. Enmascarado de información sensible: Existe.

11.9.6. Autenticidad: Existe.

11.9.7. Certificado digital: No existe.

11.9.8. SSL,VPN: No existe.

11.10. Los usuarios suben archivos al servidor: No aplica.

11.11. La aplicación interactúa con base de datos

Verificar existencia o no de vulnerabilidades en cuanto a:

11.11.1. Inyección de comandos: No se presenta.

11.11.2. Canal de comunicación no seguro: Se presenta.

11.11.3. Revelación de información: No se presenta.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.11.4. Resguardo de credenciales: Existe.

11.11.5. Filtrado de datos: Existe.

11.11.6. Escapado de salida: No existe.

11.11.7. Cifrado de datos: Existe.

11.12. Uso de Plugins (Windows media player, Acrobat, propietarios, .exe, ActiveX, toolbars): No aplica.

Etapa 3: Comunicación de Resultados

RESUMEN DE INFORME DE AUDITORIA El objetivo de esta auditoría fue determinar las vulnerabilidades que se presentan en la página web www.mat.gob.ve, especifícamente el módulo sectores agrícolas, que puede ser accedido mediante la URL: www.mat.gob.ve/sectores_agricolas/

Durante la realización de esta auditoría estuvieron involucrados directamente el administrador del servidor web, el desarrollador, el administrador de la base de datos, el director del área y la coordinadora del área

En esta auditoría toda la información física y de red, pudo obtenerse de primera mano, y además se pudo verificar la veracidad de la misma.

Se pudieron determinar amenazas y vulnerabilidades importantes a nivel físico y de aplicación para las cuales de proveen las siguientes recomendaciones:

Para los accesos no autorizados:

Se recomienda crear una política de acceso, controlarla y verificar su cumplimiento, debido a que, existe protección al acceso a los servidores, sin embargo, cualquier persona puede entrar a la sala, hasta se realizan reuniones en la misma, es decir, es probable que cualquier persona pueda desconectar un equipo, dañarlo o robar información.

Para las catástrofes naturales:

Es recomendable que los respaldos que se realizan sean debidamente almacenados en otras localidades físicas, hay una política de respaldo diaria y semanal de la información pero los datos se mantienen en la misma sala de servidores, no se replica esta información en otra localidad.

Para los ataques de fuerza bruta:

Solución: Uso de un captcha. Se puede implementar un captcha cuando se producen tres intentos fallidos durante una misma sesión.

```php
<?php
session_start();
// archivo: captcha.php
function randomText($length,$type=0) {
switch ($type) {
case 0: $pattern = "1234567890abcdefghijklmnopqrstuvwxyz"; break;
case 1: $pattern = "abcdefghijklmnopqrstuvwxyz"; break;
case 2: $pattern = "1234567890"; break;
}
$max = strlen($pattern)-1;
for($i=0;$i < $length;$i++) $key .= $pattern(mt_rand(0,$max));
return $key;
}

$_SESSION['tmptxt'] = randomText(8);
$captcha = imagecreatefromgif("bgcaptcha.gif");
$colText = imagecolorallocate($captcha, 0, 0, 0);
imagestring($captcha, 5, 16, 7, $_SESSION['tmptxt'], $colText);
header("Content-type: image/gif");
imagegif($captcha);
?>
```

Figura 7.30: Código fuente del archivo modelo/phpcaptcha.php (Elaboración propia)

Para implementarlo se agrego a la aplicación Web un archivo denominado modelo/phpcaptcha.php, y se modificaron index.php,
control/usuarios/inicio_sesion.asp

```
            header("location:../../vistas/formulario_sa/index.php");
            }else{
            $_SESSION["mensaje"]="Contrase&ntilde;a incorrecta";
            header("location:../../index.php");
            exit();
            }

        }else{
        if(!isset($_SESSION["intentos"])){
            $_SESSION["intentos"]=0;
        }
        $_SESSION["intentos"] = $_SESSION["intentos"] + 1;
        $_SESSION["mensaje"]="El usuario no esta registrado en la base de datos";
        header("location:../../index.php");
        exit();
        }
}else{
    $_SESSION["mensaje"]="Los caracteres no coinciden de la imagen no coinciden";
    header("location:../../index.php");
```

Figura 7.31: Código fuente del archivo control/usuario/inicio_sesion.php (Elaboración propia)

Para la revelación de información:

Solución: Colocar un mensaje que no revele este tipo de información o que confunda al atacante. El mensaje que se va a colocar es el siguiente: .[el] login o password son incorrectos. Intente de nuevo". Con esta información el atacante no tendrá como diferenciar si es incorrecto el login o el password. Los archivos modificados fueron: control/usuario/inicio_sesion.php.

Para la autenticación insuficiente:

Solución: Se creo un archivo en control/usuarios/ llamado sesion.php que verifica si un usuario ha iniciado sesión o no, en caso de no haber iniciado lo redirecciona al index.php

Figura 7.32: Implementación de la página que verifica la sesión de un usuario (Elaboración propia)

Para el Cross Site Scripting (XSS):

Solución: Utilizar la función nativa de php htmlentities con la cual todos los caracteres que tengan una entidad equivalente en HTML serán cambiados a esas entidades.

Figura 7.33: Resultado de utilizar la función html_entities (Elaboración propia)

Etapa 4: Verificación Posterior (Follow-up):

Es preocupante que la recomendación realizada para los accesos no autorizados, no ha sido tomada en cuenta, se mantiene el riesgo que implica que cualquier persona entre a la sala de servidores, se siguen realizando reuniones en este sitio, que deberia tener control de acceso.

De igual forma, la recomendación para las catastrofes naturales, tampoco fue adoptada, se siguen manteniendo las copias de seguridad en el mismo sitio.

Las recomendaciones realizadas en la capa de aplicación fueron implementadas, entre las cuales se puede resaltar la implementación del archivo sesion.php para evitar la autenticación insuficiente, ahora para que un usuario vea X página del sistema debe estar debidamente autenticado.

De igual forma se verificó que toda la información mostrada al usuario fuera escapada utilizando la función nativa de php htmlentities.

Caso de estudio # 3: Aplicación de registro I.V.S.S.

Etapa 1: Planificación

Comprensión del negocio y de su ambiente

1. En el presente caso de estudio se va a aplicar la metodología de auditoría de seguridad para aplicaciones web, a una aplicación para el ingreso de solicitudes de registro. Está aplicación esta destinada a un conjunto de empresas del **Instituto Venezolano de los Seguros Sociales**. Él instituto se define en su página web *http://www.ivss.gov.ve/* de la siguiente manera :

   "El Instituto Venezolano de los Seguros Sociales es una institución pública, cuya razón de ser es brindar protección de la Seguridad Social a todos sus beneficiarios en las contingencias de maternidad, vejez, sobrevivencia, enfermedad, accidentes, invalidez, muerte, retiro y cesantía o paro forzoso, de manera oportuna y con calidad de excelencia en el servicio prestado, en atención al marco legal, bajo la inspiración de la justicia social y de la equidad, garantiza el cumplimiento de los principios y normas de la Seguridad Social a todos los habitantes del país, de manera oportuna y con calidad de excelencia en los servicios prestados."

   La aplicación a auditar tiene por finalidad recibir las solicitudes de registro, las cuales son el mecanismo para obtener las credenciales de acceso (Cuenta de usuario y contraseña) a otro sistema denominado: **Sistema de Gestión y Autoliquidación de Empresas**. La descripción y el analisis de está otra aplicación esta fuera del alcance de esta auditoría.

   El proceso de solicitud de registro consiste en 4 pasos.

   1) Un primer paso donde se crea una nueva solicitud y se establece un canal de comunicación directo entre el usuario y el I.V.S.S (cuenta de correo electrónico).

   2) Luego a través de un correo electrónico se envia un Número de Solicitud y Número de Confirmación para que el usuario pueda llenar los datos de la solicitud. Una vez completado todos los datos de la solicitud, el Sistema le dará la opción de imprimir la constancia de registro.

   3) En el tercer paso el usuario debe imprimir la constancia de solicitud y llevarla a la oficina administrativa correspondiente. Adicionalmente debe consignar ciertos requisitos, los cuales serán revisados por un funcionario del I.V.S.S., para verificar la autenticidad de los datos y proceder a aprobar dicha solicitud.

   4) Una vez aprobada, el sistema genera de forma automática las credenciales de acceso al sistema y las mismas se les hará llegar vía correo. A partir de ese momento el usuario podrá tener acceso al Sistema de Gestión y Autoliquidación de Empresas.

(información extraida directamente de la aplicación web en producción en `http://190.9.128.114/home.htm` *en fecha 21-04-2008)*

2. Se realizó entrevistas con las personas encargadas y responsables del proyecto de la Dirección General de Afiliación y Prestaciones en Dinero. En donde se determino:

   Se requiere colectar información de las empresas y su información de registro fiscal.

   Actualmente dicha información no existe o es poco confiable. Por tanto se va a crear un expediente desde cero con esta información, incluyendo dirección fiscal de la empresa, razón social, número de cédula del representante legal, entre otros.

   El proceso contempla una fase administrativa en donde los usuarios deben llevar las solicitudes impresas, junto con algunos recaudos, a las instalaciones físicas del seguro social, donde un funcionario verificara la autenticidad de los datos en la solicitud contra los archivos físicos como original y copia del registro de información fiscal, cédula de indentidad del representante legal y recibo de agua, luz o telefono. Esta parte del proceso esta fuera del alcance de esta auditoría, sin embargo la aplicación usada por el funcionario para aprobar dichas solicitudes, sí sera tomada en cuenta.

Etapa 2: Describir el sistema

Capa Física

3. Describa lo mejor posible y con el mayor detalle la infraestructura física de los dispositivos y equipos en el sistema, así como las instalaciones físicas, edificio, cuarto de servidores, puertas, etc. Detalle de los equipos servidores, dispositivos de red, cableado, tipo de instalaciones, personal en el sitio, etc. Indique si dicha arquitectura es de administración propia o por parte de terceros. En el caso que sea administrada por terceros, por ejemplo a través de un servicio de hospedaje (hosting), solicitar al proveedor toda la información referente al mismo. Tomar nota de si es un servicio de hosting compartido o de uso exclusivo. Puntos de accesos a las instalaciones, controles de acceso (llaves, claves, cámaras de vigilancia, dispositivos biométricos, vigilantes, registro de novedades), personas autorizadas, vigilantes, otros.

   3.1. ¿Hardware propio de la empresa? No.

   3.2. ¿Hardware arrendado a terceros? (hosting)

   Sí. por la empresa CANTV C.A., servidores dedicados para el I.V.S.S. El reguardo de los equipos es responsabilidad de la empresa proveedora. La siguiente información es producto de entrevistas con el personal técnico, quienes visitaron las instalaciones de CANTV donde residen los equipos, al momento de la contratación.

   3.2.1. Políticas de acceso. Sí, solo personal autorizado, politicas estrictas.

   3.2.2. Personal autorizado. Empleados de la compañia CANTV. No se dispone de nombres ni perfiles.

3.2.3. Instalaciones físicas - Edif. Propio de la empresa proveedora, cuarto de servidores adecuado, aire acondicionado entre otros.

3.2.4. Los equipos y dispositivos: Segun informe de compra de servicio:

a) 10 Servidores UNIX, 2 Procesadores, 100Gb especio en disco, 4Gb memoria RAM.

b) Conexión pública a través de internet.

3.3. Administración directa en el equipo. No aplica

Capa de Red

4. Describa lo mejor posible y con el mayor detalle la arquitectura de red en donde reside la aplicación. Detalles de configuración de la red, identificar servidores, protocolos de red utilizados, existencia de enlaces privados o públicos (VPN, intranet o enlaces dedicados), presencia de Firewall y/o IDS, existencia o no de una DMZ, etc. Realice diagramas, esquemas y tablas.Indique si dicha arquitectura es de administración propia o por parte de terceros. En el caso que sea administrada por terceros, por ejemplo a través de un servicio de hospedaje (hosting), solicitar al proveedor toda la información referente al mismo. Usar herramientas de escaneo de vulnerabilidades automatizado (Nmap, nessus, retina, otros).

4.1. ¿Administración por terceros?

Sí. La compañía proveedora configura todos los enlaces y conexiones de red de los equipos.

4.2. Identificar los equipos en la red

4.2.1. Direcciones IP privadas: Servidor de Base de datos 10.70.202.76, Servidor Web 10.70.202.75, Servidor de Correo 10.70.202.02

4.2.2. Direcciones IP públicas: 190.9.128.114 (única para servidor web).

4.2.3. Equipos en Clúster, centralizados por el equipo donde esta el servidor web. Son servidores espejo.

4.3. Identificar dispositivos de firewall. Sí, configuración por cuenta de la empresa de hospedaje.

4.4. IDS/IPS. Sí, configuración por cuenta de la empresa de hospedaje.

4.5. Identificar protocolos de red usados

4.6. Esquematizar la arquitectura de red (hacer uso de técnicas de modelado)

4.6.1. Diagramas UML de componentes especializados para redes

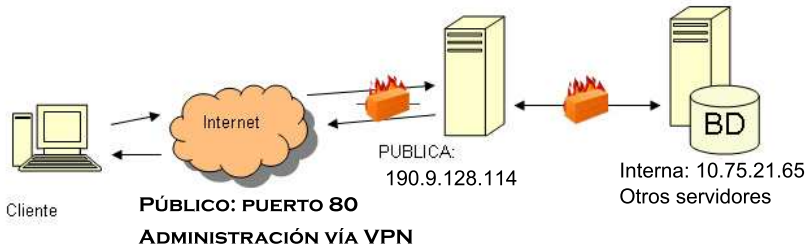


Figura 7.34: Diagrama de la red

4.7. Servidores de DNS privados. Sí, configuración por cuenta de la empresa de hospedaje.

Capa de Sistema Operativo y Servicios

5. Describa lo mejor posible y con el mayor detalle las características de los equipos en cuanto a, el sistema operativo que usan, el sistemas de archivos y esquemas de permisologías existentes, servicios disponibles, usuarios del sistema, ubicación de los recursos en los sistemas.

5.1. Sistema operativo Unix - Solaris

5.1.1. Distribución del SO: Solaris 10 Sparc

5.1.2. Usuarios y Grupos: `root, pacheco, tiuna, tomcat, mysql, oracle`

5.1.3. Usuarios y Grupos administradores: `root, pacheco`

5.2. Servicios. Con excepción del sistema operativo, la instalación y configuración de otras aplicaciones y servicios son responsabilidad del cliente.

identificar:

5.2.1. Servidores de aplicaciones: Tomcat

5.2.2. Servidores de bases de datos: Oracle

5.2.3. Servidores de correo: posfix

5.2.4. Servidores LDAP: No aplica

5.2.5. Servidores FTP: No aplica

5.3. Para cada servicio

Tomcat

5.3.1. Versión: 6.0.16

5.3.2. Usuarios y permisos de ejecución: `tomcat/tomcat` (Usuario limitado)

5.3.3. Archivos y directorios relacionados: `/etc/tomcat/ ; /etc/tomcat/bin/` archivos binarios

5.3.4. Puertos que utilizan: puerto 80.

5.3.5. Archivos de configuración, propiedades: Todos los archivos de configuración del tomcat están en `/etc/tomcat/conf`.

```
-rwxrwxrwx 1 tomcat tomcat 6592 2007-07-19 22:20 catalina.policy
-rwxrwxrwx 1 tomcat tomcat 2883 2007-07-19 22:20 catalina.properties
-rwxrwxrwx 1 tomcat tomcat 569 2007-07-19 22:20 context.xml
-rwxrwxrwx 1 tomcat tomcat 2882 2007-07-19 22:20 logging.properties
-rwxrwxrwx 1 tomcat tomcat 5635 2007-07-19 22:20 server.xml
-rwxrwxrwx 1 tomcat tomcat 70 2008-02-13 17:39 tomcat-users.xml
-rwxrwxrwx 1 tomcat tomcat 49663 2007-07-19 22:20 web.xml
```

Nota: Permisos no estrictos.

Oracle

5.3.1. Versión: Oracle 10 G.

5.3.2. Usuarios y permisos de ejecución: `oracle/oracle` (Usuario limitado).

5.3.3. Archivos y directorios relacionados: `/oracle/`.

5.3.4. Puertos que utilizan: puerto 1521 (por defecto).

5.3.5. Cuentas de usuarios del servicio: `tiuna, aldsira, register_request, sira`.

5.4. Sistema de archivos

5.4.1. Partición: ext3.

5.4.2. Identifique recursos compartidos: `/tmp` (archivos temporales del SO), `/etc/tomcat/work/ROOT` (archivos temporales)

5.4.3. Identifique directorios protegidos: `/, /var, /home, /etc/`,entre otros. Cualquier directorio fuera de `/etc/tomcat/webapp/ROOT` y `/tmp`

5.5. Administración remota

5.5.1. A través de red interna. No aplica

5.5.2. VPN. Sí (usuario ivss con politicas de vencimiento de contraseña)

5.5.3. SSH. Sí (usuario root y tiuna).

5.5.4. Escritorio Remoto. Nó

6. Identifique las aplicaciones o módulos presentes en la aplicación web. Para cada una de estos elementos determinar como mínimo el nombre de aplicación o módulo, versión, ubicación, permisos de ejecución relacionados (si aplica). Algunas tecnologías comúnmente relacionadas con las aplicaciones web son: servidores de aplicaciones o servidores web (Apache, IIS, Tomcat...), servidores de base de datos, tecnologías del lado del servidor (Java, ASP .Net, PHP, Ruby...), entre otras.

6.1. describir servidores de aplicaciones

6.1.1. Servidor Web: Tomcat

6.1.2. Versión: 6.0.16

6.1.3. Permisos de ejecución: usuario limitado tomcat, solo tiene permiso sobre directorio `/etc/tomcat` y `/tmp`.

6.1.4. Accesos y permisos de directorios:la aplicación esta ubicada dentro del servidor tomcat en el directorio `/etc/tomcat/webapp/ROOT`.

6.1.5. Cuentas de usuarios del servicio: archivo tomcat-user.xml; cuentas tomcat, manager y roles manager, tomcat, role1 (cuentas por defecto y otras)

6.1.6. Módulos instalados o librerías: `spring.jar, log4j.jar, oci.jar,activation.jar, mail.jar, commons*.jar`

6.2. Servidores de base de datos

6.2.1. Servidor de base de datos: ORACLE

6.2.2. Versión: Oracle 10.02 G.

6.2.3. Ubicación `/etc/oracle/`

6.2.4. Permisos de ejecución: usuario limitado oracle.

6.2.5. Accesos y permisos de directorios: `/etc/oracle`

6.2.6. Usuarios o cuentas de administración y usuarios: `oracle, tiuna, register_request`, entre otros usuarios/esquemas.

6.2.7. Acceso

a) Acceso Local: acceso únicamente desde la red local desde las direcciones IP 10.70.202.*

b) Acceso Remoto: No se permite acceso remoto, protegido tanto por firewall como por configuración de politicas en el manejador de base de datos.

6.2.8. Cifrado de archivos físicos: No se usa cifrado de datos fisicos.

6.3. Servidores LDAP: NA.

6.4. Tecnologías del lado del servidor

6.4.1. Identificar las tecnologías usadas: Java (jsp y servlets).

6.4.2. Versión: Java 6.

Capa de Aplicación

7. Describa lo mejor posible y con el mayor detalle las tecnologías usadas en las aplicaciones web. Identificar protocolos de comunicación (HTTP, HTTPS, FTP, LDAP...), uso de Frameworks (Struts, Spring, Cakephp, phpNuke, Ruby on Rails...) y los patrones de diseño (Composite view, MVC) que utilizan, tecnologías del lado del cliente (HTML, CSS, Javascript, AJAX, Flash, Plugins como Windows Media Player o Acrobat Reader, entre otros...), archivos y recursos de la aplicación, acceso a datos (servidores de base de datos, SAN, sistemas de archivos), servicios web.

7.1. Protocolos de comunicación

7.1.1. HTTP: Sí.

7.1.2. HTTPS: No.

7.1.3. SMTP

a) ¿Público o Privado?: público.

b) Autenticado: Sí.

7.1.4. SFTP

a) Privado: a través de la VPN.

b) Autenticado: Sí.

7.2. Determinar el uso de uno o más Frameworks. Conocer o investigar el Framework sobre el cual están hechas las aplicaciones ayuda a una mejor comprensión de sus funcionalidades y procesos, desde su estructura y organización de archivos y recursos hasta el código fuente e implementación.

7.2.1. Framework: Spring Framework.

7.2.2. Versión: 2.0

7.2.3. Librerías y complementos: Ibatis,Spring Web MVC, Jdbc Spring.

7.2.4. Patrones/paradigma: MVC, AOP (Aspect Oriented Programming).

7.2.5. Estructura de la aplicación:

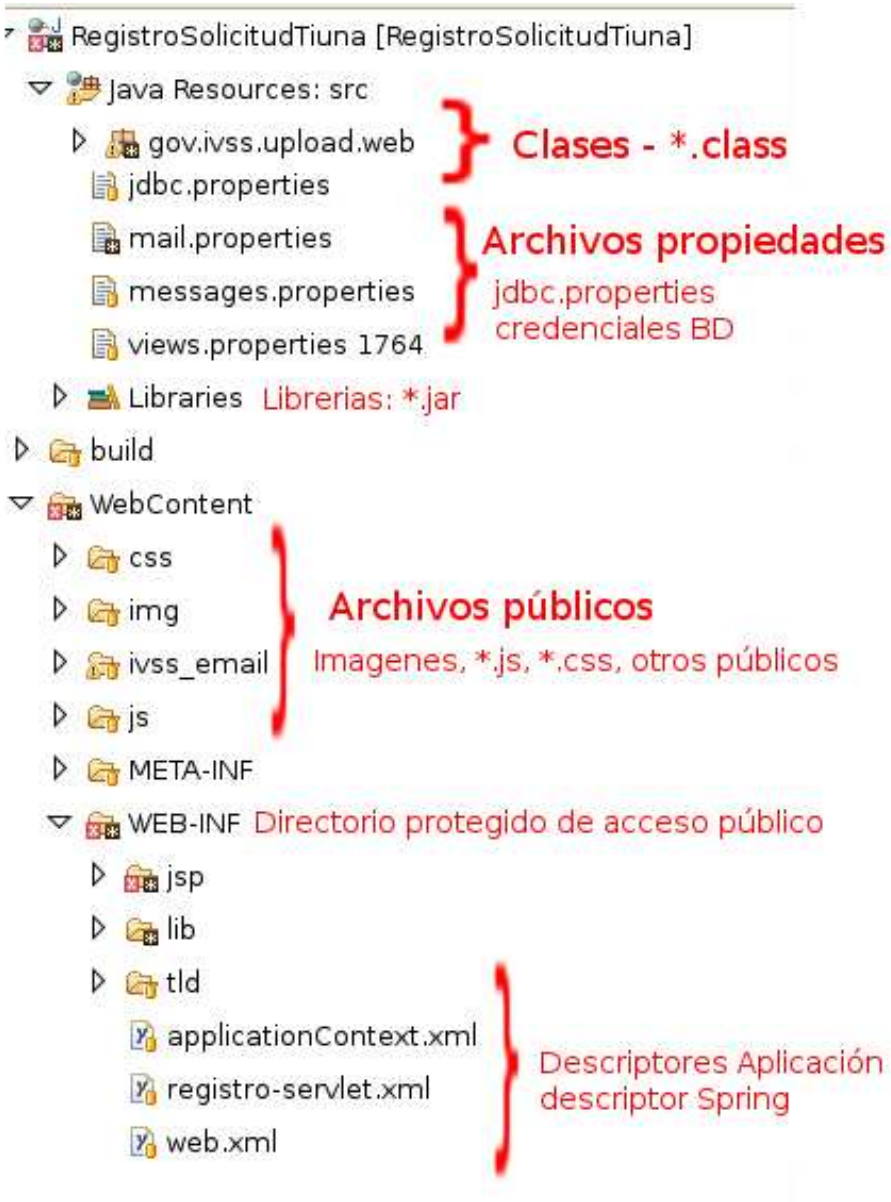


Figura 7.35: Estructura de Directorio y recursos de la aplicación

`/"` *representa el directorio más alto de la aplicación*

a) Ubicación de archivos fuentes: `/src`
b) Ubicación de archivos de registro (logs): No Existen.
c) Ubicación de archivos de configuración: `/src y /WEB-INF/classes`
d) Ubicación de archivos públicos: `/, /img , /css, /js, /ivss_email`
e) Ubicación de archivos subidos al servidor `/upload`

Lo que esta dentro del directorio `WEB-INF` *no es accesible públicamente. Archivos protegidos de acceso público.*

7.3. Tecnologías del lado del cliente involucradas

7.3.1. HTML Y CSS.

7.3.2. Javascript y AJAX para algunas pantallas, actualización de dirección via ajax.

7.3.3. Complementos o Plugins (Windows Media Player, Flash,Acrobat Reader): No aplica.

7.3.4. Complementos propios (.exe, ActiveX, hotbars): No aplica.

7.3.5. Cookies: utiliza cookies de sesión, el mecanismo por defecto provisto por el servidor tomcat.

7.4. Acceso a datos

7.4.1. Acceso a base de datos

a) Almacenamiento de credenciales de acceso: `/WEB-INF/classes/jdbc.properties`

b) Cifrado de datos: No disponible.

c) conexión: pool de conexiones via `jdbc`.

7.4.2. Acceso a archivos

a) Permisología sobre los recursos: Todos los archivos en el servidor poseen permisos totales (permisos unix: `-rwxrwxrwx ó 777`)

b) Cifrado de archivos: No poseé.

7.5. Acceso a servicios.

7.5.1. Servicios Web (Web Services)

a) Internos: No.

b) Externos: No.

8. Determinar funcionalidades o características que existen en la aplicación web. Identificando características como registro de usuario, inicio de sesión, uso de certificado digital o no, foros, formas para recuperación de contraseñas, modulo de administración, administradores de contenido, capacidades para subir archivos al servidor, ejecución de comandos, inclusión de archivos en el servidor, accesos a recursos remotos tales como otras aplicaciones, servicios web o recursos como imágenes, hojas de estilos entre otros.

8.1. Identificar procesos de autenticación

8.1.1. Inicio de sesión con formularios web: No aplica.

8.1.2. Certificados digitales: No.

8.1.3. Sistemas de recuperación de contraseña: No aplica.

8.2. Identificar operaciones: Creación de solicitudes. URL = `/crearSolicitud.htm`

8.3. Módulos de administración

8.3.1. Administradores de contenido: No aplica.

8.3.2. Administradores de consulta y operaciones: No aplica.

8.4. Procesos con capacidades de:

8.4.1. Subir archivos al servidor: No aplica.

8.4.2. Entradas y salidas al usuario:

crearSolicitud.htm (Nro Patronal, Cédula, correo, nro solicitud)

confirmarSolicitudManual.htm (nro Solicitud)

confirmarSolicitud.htm (RIF, razón social, telefonos)

actualizarDireccion.htm (parroquia, avenida o calle, edif o casa, piso nivel, apto, punto de referencia, codigo postal, otros)

8.4.3. Envió de mensajes (correos, posts, blogs, foros)

8.5. Secciones o procesos que usan o exigen uso de javascript o ajax: actualizarDirección.htm

8.6. Secciones o procesos que usan o exigen uso de complementos: No aplica.

8.7. Secciones o procesos que pasan parametros por URL:

crearSolicitud.htm, confirmarSolicitudManual.htm, confirmarSolicitud.htmm actualizarDireccion.htm

Etapa 2: Verificación y comprobación

9. Verificación y comprobación de seguridad de la Capa Física:

9.1. Hardware propio de la empresa: No aplica.

9.2. Hardware arrendado a terceros

Verificar existencia o no de vulnerabilidades en cuanto a:

9.2.1. Accesos no autorizados: Riesgo minimo, CANTV implementa estrictas politicas de acceso para personas externas y sus empleados.

9.2.2. Catástrofes naturales: Siempre existe este riesgo, sin embargo las instalaciónes son adecuadas.

9.2.3. Accidentes y Fallas: Las instalaciones son adecuadas, el monitereo de temperatura, humedad y otros esta presente.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

9.2.4. Políticas de acceso estrictas: Sí.

a) Registro de eventos y control de acceso: Sí.
b) Guardias de seguridad: Sí.
c) Libro de novedades: desconocido.
d) Puertas cerradas: Sí.
e) Llaves o Claves: Sí, si puerta con claves de seguridad.
f) Dispositivos de seguridad biométricos: desconocido.
g) Cámaras:Sí.

9.2.5. Continuidad del negocio

a) Redundancia: No se cuenta con una plataforma espejo. Se esta en proceso de configuración en una nueva sede.
b) Corriente ininterrumpida (UPS): Sí.

9.2.6. Detección y Corrección de Fallas

a) Control de temperatura y humedad: Sí.
b) Detectores y alarmas de calor,humo y humedad: Sí.
c) Sistema de extinción de incendios: Sí.

9.3. Administración directa en el equipo: No aplica.

10. Verificación y comprobación de seguridad de la Capa de Red

10.1. Equipos de acceso público a través de la red

La fuente de esta información es directamente el proveedor y no se posee completa o con exactitud o detalles específicos

Verificar existencia o no de vulnerabilidades en cuanto a:

10.1.1. Deficiente protección de los datos en el transito: Riesgo medio. Red pública internet, no se usa SSL. Para la administración con CANTV si esta configurada una VPN.

10.1.2. Suplantación de una página web real: Riesgo alto, No se posee certificado digital verificable, la aplicación es accedida a traves de una dirección IP pública y no un nombre de dominio.

10.1.3. Denegación de Servicio (DoS): riesgo controlado, CANTV provee multiples Firewalls, IDS/IPS, adicionalmente un gran ancho de banda del enlace así como balanceradores de carga.

10.1.4. Malware (gusanos, virus): riesgo mínimo. Existen antivirus, no se suben archivos al servidor.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

10.1.5. Firewall

a) Modelo de seguridad restrictivo: Sí, restricciones por lista blanca.
b) Filtro por direcciones IP: No existe.
c) Bloquear puertos innecesarios o privados: Sí, actualmente solo el puerto 80.

10.1.6. Antivirus: Sí.

10.1.7. DMZ: Sí, los servidores internos no son accesibles desde la red pública.

10.1.8. IDS/IPS: Sí.

10.1.9. VPN: Sí existe.

10.1.10. SSL/TLS: No se usa.

10.1.11. Balanceo de Carga: Sí existe.

11. Verificación y comprobación de seguridad de la capa de Sistema Operativo y Servicios:

11.1. Sistema Operativo

Verificar existencia o no de vulnerabilidades en cuanto a:

11.1.1. Acceso no autorizado: riesgo bajo. No hay claves por defecto, exige contraseñas fuertes, vencimiento de contraseñas, solo acceso a traves de la VPN, no se permite acceso remoto del super usuario root.

11.1.2. Malware: instalado antivirus.

11.1.3. Buffer Overflow: parches actualizados y recientes.

11.1.4. Versiones viejas: No existe. Versión estable de Solaris, administrado por outsourcing capacitado (SUN microsystem).

11.1.5. Configuración por defecto: No existe. La instalación es mínima, y las nuevas funcionalidades se habilitan contra demanda.

11.1.6. Exposición de servicios innecesarios: No existe, los servicios se habilitan contra demanda.

11.1.7. Credenciales predecibles o por defecto: No existe.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.1.8. Firewall: Sí.

11.1.9. IDS, IPS: Sí.

11.1.10. Antivirus: Sí.

11.1.11. Versión reciente: Sí.

11.1.12. Actualizaciones y parches automáticas: Sí.

11.1.13. Verificación de credenciales y permisos: Sí.

11.2. ¿Se encuentra en Hospedaje Compartido? No. Es un servidor dedicado.

11.3. ¿Administración del sistema en forma remota? Sí.

Verificar existencia o no de vulnerabilidades en cuanto a:

11.3.1. Deficiente autenticación y autorización: riesgo mínimo, usuario y contraseñas con politicas de vencimiento de contraseña y definición de contraseñas fuertes.

11.3.2. Deficiencia en canales de comunicación, protocolos no seguros: mínimo, Se usa una VPN correctamente configurada, actualizaciones via secure copy, y sftp.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.3.3. Canal de comunicación seguro (SSH, SSL, VPN): Sí se usan canal de comunicación seguro para la administración.

11.3.4. Políticas de vencimiento y cambio de credenciales: Sí.

11.3.5. Firewall: Sí, lista blanca. Solo las direcciones IP autorizadas.

11.4. ¿Actualizaciones remotas?

Verificar existencia o no de vulnerabilidades en cuanto a:

11.4.1. Deficiente autenticación y autorización: no existe. La autenticación es de forma correcta ya que se debe autenticar primero el canal con la VPN, y luego el protocolo de subida de archivos bien sea la conexión SSH o contra el servidor SFTP.

11.4.2. Deficiencia en canales de comunicación, protocolos no seguros(FTP): No existe.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.4.3. Canal de comunicación seguro (SSH, SFTP, VPN): Sí.

11.4.4. Políticas de vencimiento y cambio de credenciales: Sí.

11.5. ¿Utilizan servidores de aplicación o servidores web?

Verificar existencia o no de vulnerabilidades en cuanto a:

11.5.1. Configuración por defecto: No.

11.5.2. Versiones viejas: No.

11.5.3. Falta de actualizaciones o parches: No.

11.5.4. Archivos de configuración: Vulnerables, permisos totales sobre los archivos de configuración.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.5.5. Versión reciente: Sí.

11.5.6. Actualizaciones y parches: Sí.

11.5.7. Chequeo de configuración desde cero: Sí.

11.5.8. Verificación de credenciales y permisos: Deficiente, archivos de configuración, específicamente usuarios del tomcat.

11.6. ¿Utilizan servidores de bases de datos?

Verificar existencia o no de vulnerabilidades en cuanto a:

11.6.1. Deficiente respaldo: No existen politicas de respaldo.

11.6.2. Deficiente autorización y autenticación: el esquema de base de datos esta bien definido por esquemas y permisos sobre tablas del esquema.

11.6.3. Configuración por defecto: No existe, se contrato personal especializado de Oracle para la instalación de los servicios.

11.6.4. Versiones viejas: No.

11.6.5. Falta de actualizaciones o parches: No.

11.6.6. Canal de comunicación no seguro: No.

11.6.7. Exposición de recursos innecesaria: No.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.6.8. Versión reciente: Sí, correcto.

11.6.9. Cifrado de datos: No posee.

11.6.10. Actualizaciones y parches: Sí, correcto.

11.6.11. Chequeo de configuración desde cero: Sí, correcto.

11.6.12. Verificación de credenciales y permisos: Sí, correcto.

11.7. Uso de tecnologías del lado del servidor para contenido dinámico

Verificar existencia o no de vulnerabilidades en cuanto a:

11.7.1. Ejecución super usuario: No, en este caso la maquina virtual de Java usa la misma permisología del servidor que la contiene.

11.7.2. Deficiente autorización: No aplica. No esta habilitado el módulo de administración del tomcat, los *deploy* se hacen manualmente.

11.7.3. Configuración por defecto: No Vulnerable.

11.7.4. Versiones viejas: No Vulnerable.

11.7.5. Falta de actualizaciones o parches: No Vulnerable, esta actualizado.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

11.7.6. Verificación de credenciales y permisos de ejecución: Deficiente definición de usuarios del servidor web tomcat.

12. Verificación y comprobación de seguridad de la capa de aplicación

12.1. Inicio de Sesión de Usuario: NA.

12.2. Recuperación de Contraseña: NA.

12.3. Administradores Remotos: NA.

12.4. Registro de usuarios en línea: Sí.

/crearSolicitud.htm

Verificar existencia o no de vulnerabilidades en cuanto a:

12.4.1. Revelación de Información: Sí, el manejo de excepciones es inapropiado, Revela nombre de componentes, muestra nombres de campos y tablas de base de datos. El registro no esta sobre un canal seguro (SSL). Permite farming de las empresas que pertenecen o no pertenecen al conjunto de las empresas S.A.N.E.

12.4.2. Abuso de Funcionalidad: riesgo Alto. No existen restricciones, cualquier persona o programa puede usar el sistema. La única restricción es que el número de empresa pertenezca a un sub-conjunto S.A.N.E.

Figura 7.36: Pantalla registro sin CAPTCHA ni protección SSL

12.4.3. Inyección de código SQL: No, se usa jdbc y los queries estan bien construidos haciendo uso correcto del mismo.

12.4.4. Phishing: Riesgo alto, no se usa certificado digital, el URL de acceso es una dirección IP y no esta asociada a un nombre de dominio, No hay información con educación al usuario contra phishing.

Figura 7.37: Acceso a la aplicación a través de una dirección IP

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.4.5. Captcha: No existe, Vulnerable a abuso de funcionalidad.

12.4.6. Filtrado de entrada: Sí se validan los datos en forma.

12.4.7. Escapado de salida: Sí, se usan mecanismos provistos por el framework spring para el escapado de las salidas y los datos no se extraen directamente del request.

12.4.8. Conexiones seguras: No, actualmente vulnerable. No se usa SSL.

/confirmarSolicitud.htm y /actualizarDirección.htm

Verificar existencia o no de vulnerabilidades en cuanto a:

12.4.1. Revelación de Información: Sí, muy vulnerable.

El manejo de excepciones es inapropiado, revela nombre de componentes, muestra nombres de campos y tablas de base de datos. El registro no esta sobre un canal seguro (SSL).

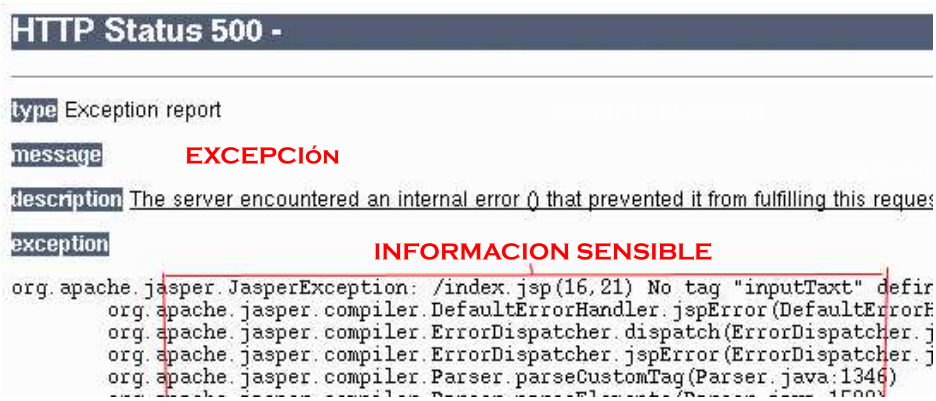


Figura 7.38: Manejo de Excepciones inapropiado, error 500

Vulnerable a exposición de objetos de dominio.

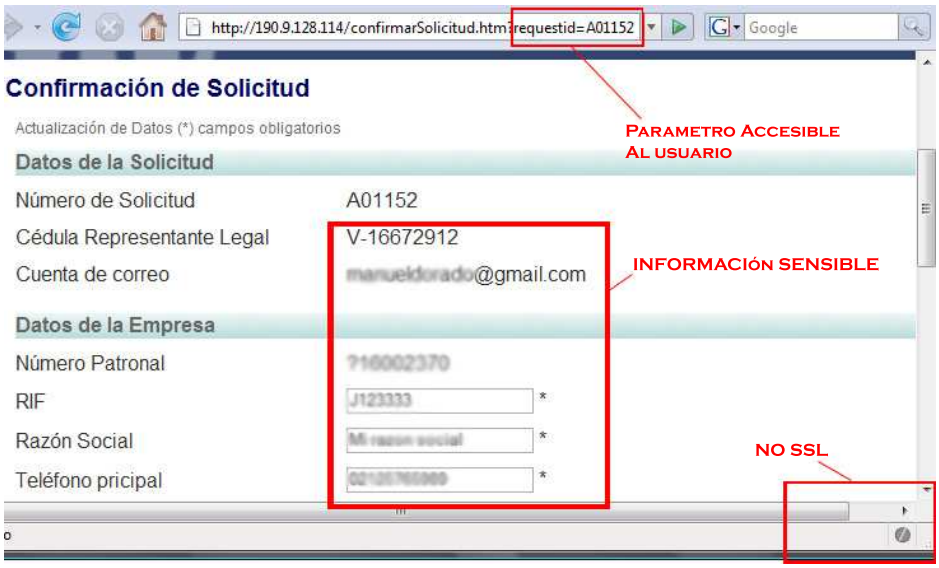


Figura 7.39: Acceso a información sensible manipulando los parametros

12.4.2. Abuso de Funcionalidad: Riesgo Alto, conjuntamente con la revelación de objetos de dominio, se pudiera estar modificando secuencialmente datos de todas las solicitudes.

12.4.3. Inyección de código SQL: No vulnerable, se usa jdbc y los queries estan bien construidos haciendo uso correcto del mismo.

12.4.4. Phishing: Riesgo alto, no se usa certificado digital, el URL de acceso es una dirección IP y no esta asociada a un nombre de dominio, No hay información con educación al usuario contra phishig.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.4.5. Captcha: No existe, Vulnerable a abuso de funcionalidad.

12.4.6. Filtrado de entrada: Sí se validan los datos en forma.

12.4.7. Escapado de salida: Sí, se usan mecanismos provistos por el framework spring para el escapado de las salidas y los datos no se extraen directamente del request.

12.4.8. Conexiones seguras: No, actualmente vulnerable.

12.5. Usuarios mantienen sesión: NA.

12.6. Usuarios realizan operaciones

/crearSolicitud.htm: es una operación que al terminar inserta un registro nuevo en la base de datos correspondiente a una nueva solicitd, y envia un correo a la dirección de email sumistrada por el usuario. Pudiera usarse esta operación para un envio de correo indiscriminado a cualquier dirección de email arbitraria y/o causar problemas de desempeño al manejador de base de datos.

Verificar existencia o no de vulnerabilidades en cuanto a:

12.6.1. Abuso de funcionalidad: Sí, acepta cualquier dirección y no hay controles contra automatización. Se pudiera tambien hacer un envio de correos indiscriminados a cualquier dirección de correo.

Figura 7.40: Información de correo de confirmación

12.6.2. Anti-automatización insuficiente: Vulnerable.

12.6.3. Repudiación de Operaciones: NA.

12.6.4. Fijación de sesión: NA.

12.6.5. Suplantación de contenido: NA.

12.6.6. Cross-Site Request Forgery:NA.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.6.7. Filtrado de entrada: Sí, correcto.

12.6.8. Escapado de salida: Sí, correcto.

12.6.9. Token: No, no hace falta.

12.6.10. Captcha: No, vulnerable, necesita control de automatización.

12.6.11. Verificación de autorización: Deficiente, cualquier puede modificar los datos de una solicitud con solo obtener su número.

12.6.12. Logs y registro de acciones: Esta correcto.Se registran las nuevas solicitudes y cuando son confirmadas.

12.6.13. Notificaciones al usuario: El usuario no es notificado, se le deberia notificar al realizar cambios en su solicitud.

12.6.14. Educación a usuarios: No existe, vulnerable. Se debe inducir al usuario a practicas que le ayuden a su seguridad como solo accesar a este módulo a través del enlace que existe en la página principal del I.V.S.S. www.ivss.gov.ve.

12.7. Sitios Web que se comunican con Usuarios

Verificar existencia o no de vulnerabilidades en cuanto a:

12.7.1. Phishing: Riesgo Alto.

12.7.2. Revelación de información: Riesgo Alto.

12.7.3. XSS: NA.

12.7.4. XSRF: NA.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.7.5. Educación a usuarios: Vulnerable no existe.

12.7.6. Escapado de la salida: Sí, correcto.

12.8. Pase de parámetros accesibles al usuario

/confirmarSolicitud.htm y /actualizarDirección.htm

Verificar existencia o no de vulnerabilidades en cuanto a:

12.8.1. Revelación de Información: No se usa SSL, los parametros estan disponibles via GET en la URL.

12.8.2. Referencia Directa de objetos:Vulnerable, El numero de solicitud es manipulable con facilidad y sin ningun mecanismo de control de autorización.

Vulnerable a Revelación de información sensible a través de una vulnerablidad deexposición de objetos de dominio, unicamente con el ID de la solicitud se puede obtener y modificar datos de la misma, se puede hacer *farming* con cambiar el número de solicitud. Ejemplo: `confirmarSolicitud.htm?request_id=A0000X`.

12.8.3. Autorización insuficiente: Vulnerable. Sólo se pide el número de solicitud, sin otro tipo de dato para garantizar que quién que actualiza la solicitud tiene autorización para hacerlo.

```
@Override
protected ModelAndView showForm(HttpServletRequest request,
        HttpServletResponse response, BindException errors)
        throws Exception {

        String requestId = (String)request.getParameter("requestid");

        RegisterRequest registerRequestForm =
                        registerRequestManager.getRegisterRequestByRequestID(requestId);
        if(registerRequestForm!=null ){

            registerRequestForm.setMetadata("ok");
            registerRequestManager.updateRegisterRequestId(registerRequestForm);

        }
    return super.showForm(request, response, errors);
}
```

Figura 7.41: Código donde se solicita unicamente Nro Solicitud (`requesid`)

12.8.4. Inyección de código: No vulnerable. Los datos estan siendo filtrados y escapados adecuadamente.

12.8.5. XSS: No vulnerable. Ningun entrada del usuario es usada como insumo para la ejecución de código.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.8.6. Filtrado de entrada: Correcto.

12.8.7. Filtrado de entrada: Escapado de Salida.

12.8.8. Verificación de autorización: Deficiente, No existe.

12.9. Los usuarios suben archivos al servidor: NA.

12.10. La aplicación interactúa con base de datos

Verificar existencia o no de vulnerabilidades en cuanto a:

12.10.1. Inyección de comandos: No vulnerable. Se esta usando correctamente filtrado via JDBC.

12.10.2. Canal de comunicación no seguro: Entre la aplicación y el servidor de base de datos la comunicación es segura. Red interna, y servidor de BD con politicas correctas.

12.10.3. Revelación de información: La información en la Base de datos se guarda absolutamente plana, riesgo moderado.

Verificar existencia o no y deficiencia en cuanto a políticas y controles de:

12.10.4. Resguardo de credenciales: Correcto, archivos .properties, dentro de un directorio no accesible publicamente.

12.10.5. Filtrado de datos: Correcto.

12.10.6. Escapado de salida: Correcto.

12.10.7. Cifrado de datos: No se usa. Riesgo moderado.

12.11. Uso de Plugins:NA.

Etapa 3: Comunicación de Resultados

Resumen de informe de auditoria y exposiciones

Los objetivos se cumplieron, se audito el módulo de solicitud de registro. La aplicación resulto ser bastante insegura, sobre todo a nivel de aplicación. A nivel de capa fisica y capa de red esta

bien ya que se la administración corre por una empresa especializada, con basta experiencia en telecomunicaciones y sistemas. En su mayoria las vulnerabilidades presente son producto de configuraciones por defecto y descuidos.

CAPA física: La empresa de hospedaje provee los requerimientos mínimos a nivel de capa física para el buen resguardo de los equipo. Cuenta con las instalaciones apropiadas para el alojamiento de servidores y cuenta con la seguridad necesaria. No se requiere redundancia de los sistemas para esta aplicación ya que no requiere alta disponibilidad y es no es considerada crítica para el negocio del Instituto Venezolano de los Seguros Sociales.

Capa de red: La empresa de hospedaje y arredamiento de enlaces tiene una arquitectura de red solida a nivel de telecomunicaciones, lo cual ofrece una gran ventaja para el Seguro Social al contar con este compañero de negocio para el soporte de su red, ya que el mismo posee el conocimiento y cuanta con los recursos para montar, rapidamente y de forma eficiente, enlaces de comunicación públicos protegidos adecuadamente contra los ataques más comunes de red. Por mercionar algunas capacidades: DMZ's, configurar acceso de administración seguros mediante una VPN, etc. El contrato es bien explícito en cuanto a los niveles de disponibilidad a los que estan sujetos y deben garantizar.

EL acceso a la aplicación es mediante una dirección IP y no a través de un nombre de dominio. Se recomienda adquirir un dominio para la aplicación o configurar un subdominio con el dominio raíz del Instituto: ivss.gov.ve. por ejempo, registro.ivss.gov.ve.(Esto se recomendo y en la ETAPA 4 de verificacion se coloca q no se realizo, ignoramos porque).

Capa de SO y servicios: El servidor web a pesar de correr con la permisologia apropiada (usuario tomcat) el directorio de archivos de configuracion no presentaba permisos estrictos.

Se recomienda cambiar

```
-rwxr-x--- 1 root tomcat 6592 2007-07-19 22:20 catalina.policy

-rwxr-x--- 1 root tomcat 2883 2007-07-19 22:20 catalina.properties

-rwxr-x--- 1 root tomcat 569 2007-07-19 22:20 context.xml

-rwxr-x--- 1 root tomcat 2882 2007-07-19 22:20 logging.properties

-rwxrwx--- 1 root tomcat 5635 2007-07-19 22:20 server.xml

-rwxr-x--- 1 root tomcat 70 2008-02-13 17:39 tomcat-users.xml

-rwxrwx--- 1 root tomcat 49663 2007-07-19 22:20 web.xml
```

de igual forma se deben configurar los permisos correctamente para los recursos como credenciales de base de datos /WEB-INF/jdbc.properties posee permisos totales 777 o -xrwxrwxrwx.

Los archivos fuentes son exportados al servidor de producción y no se encuentran en un directorio seguro. Están ubicados dentro de el documento raíz del servidor web. Se recomendo no subir la versión con los fuentes, en todo caso si esto se requiere, se deben proteger dentro en un directorio fuera del documento raíz del servidor o dentro de la carpeta protegida WEB-INF.

Por lo demás esta actualizado y bien configurado con su firewall.

No se posee certificado digital. No se realiza una autenticación a los usuarios por parte del instituto.

No se utiliza SSL y los usuarios suministran información sensible como datos personales y datos fiscales de la empresa, toda esta información viaja en texto plano.

Las políticas de respaldo de las bases de datos son deficientes, no se realizan respaldos en ningun momento. Si ocurre una eliminación de tablas de forma accidental o intencional se perderían todos los registros de solicitudes y la información de las empresas.

Se recomienda establecer políticas de respaldo periódicas con una frecuencia diaria. Realizarla en los mismos servidores de CANTV, si es posible descargar los datos a servidores locales del Instituto de los Seguros Sociales.

No existe cifrado físico de datos, pero por ser un servidor dedicado el riesgo de robo de los archivos es bajo, además que debe ser garantizado por CANTV.

Para el abuso de funcionalidad:

Solución: Se recomendó incorporar un mecanismo para diferenciar automatas de humanos (CAPTCHA).

Figura 7.42: Incorporación de captcha en la aplicación (`requestid`)

Para el abuso de funcionalidad:

Solución: Se recomienda añadirle un número de confirmación al correo que contiene el número de solicitud.

Figura 7.43: Correo donde se envía Nro. de Solicitud y Nro. de confirmación(`requestid`)

Para implementar esta recomendación el cambio a nivel de código es:

```
@Override
protected ModelAndView showForm(HttpServletRequest request,
        HttpServletResponse response, BindException errors)
        throws Exception {

String requestId = (String)request.getParameter("requestid");
String confirmationNumber = (String)request.getParameter("confirm");

RegisterRequest registerRequestForm = registerRequestManager.getRegisterRequestByRequestID(requestId);

if(registerRequestForm!=null && registerRequestForm.getConfirmationNumber().equals(confirmationNumber)){
    registerRequestForm.setMetadata("ok");
    registerRequestManager.updateRegisterRequestId(registerRequestForm);

}else{
    return new ModelAndView(new RedirectView("confirmarSolicitudManual.htm")
}
```

NUMERO CONFIRMACIÓN REQUERIDO

COMPARACIÓN NRO CONFIRMACIÓN

Figura 7.44: Código donde se verifica tanto Nro. de Solicitud como el Nro. de confirmación(`requesid`)

El numero de confirmación esta en la petición y no se puede cambiar simplemente la secuencia del número de solicitud, se requiere conocer también el número de confirmación que no es predecible.

Figura 7.45: Parámetro con confirmación (`requestid`)

Para la revelación de información:

Solución: Manejo correcto de errores que no revele campos de base de datos, ni nombre de objetos.

Figura 7.46: Manejo correcto de errores(`requesid`)

Para la revelación de información:

Solución: Se recomienda enmascarado de datos para solicitar el numero de solicitud y contraseña, y que los mismos sean pasados por post y no esten expuestos en la URL.

Para las vulnerabilidades en general:

Solución: Se recomienda educar al usuario en una sección informativa, contra amenazas y vulnerabilidades.

Etapa 4: Verificación Posterior (Follow-up):

En el proceso de verificación se pudo notar que las recomendaciones expuestas en el informe no fueron totalmente acatadas, lo que representa un problema de seguridad vigente para la aplicación web.

EL acceso a la aplicación es mediante una dirección IP y no a través de un nombre de dominio. Se recomendo adquirir un dominio para la aplicación o configurar un subdominio con el dominio raíz del Instituto: ivss.gov.ve. por ejempo, registro.ivss.gov.ve. No se tomo ninguna acción respecto a esto.

No se desarrollo una sección para educar al usuario, con información que le ayude a protegerse de ataques como phishing.

Sin embargo, funcionalidades como uso de CAPTCHA para no permitir la creación de múltiples solicitudes, o envío de correos masivos a cuentas registradas, fueron implementadas y probadas con éxito.

Otra funcionalidad importante, fue la implementación de un código de verificación para evitar el pharming.

Caso de estudio # 4:
SMTP tráfico en texto plano

El correo electrónico actualmente es una aplicación de uso masivo y completamente indispensable en Internet. El correo electrónico es una aplicación insegura y vulnerable que requiere introducir todas las medidas necesarias para garantizar el uso por parte de los usuarios, más allá de simplemente utilizar antivirus y antispam. También es una aplicación que permite una amplia interconexión mundial a través de intercambio de correo electrónico.

Los mensajes de correo electrónico son como una postal – cualquier çartero"los puede leer. Los antiguos protocolos en los que se sustentan el correo electrónico fueron tan confiados que concibieron la transmisión de todo tipo de datos a través de la red de forma transparente.

Cifrar mensajes de correo electrónico con PGP o S/MIME sería el equivalente electrónico de un sobre cerrado en el mundo del correo postal. Los mensajes de correo cifrados por los emisores viajarán seguros por la red hasta llegar a los destinatarios que serán los únicos que podrán acceder al contenido. Sin embargo, estos métodos individuales, si bien, son útiles han venido planteando varios tipos de problemas que no han permitido su utilización de forma habitual. Un problema es el legal, ya que algunos países niegan a sus ciudadanos el acceso a métodos de cifrado en aras

de la lucha contra el crimen. El segundo problema es de aprendizaje por parte del usuario debido a que estos métodos de cifrado no son transparentes al usuario y se requiere conocimientos informáticos e intervención. Un tercer problema que se plantea en una institución o empresa que implante una política de cifrado personal es ¿Qué ocurre si una tercera persona necesita acceder al correo posteriormente? ¿Qué ocurre con las claves utilizadas por una persona de la empresa que deja de trabajar en ella, o con su correspondencia cifrada? Como el cifrado normalmente sólo se utiliza para prevenir que datos sensibles crucen la red en forma plana, un sistema de encriptación que funcione en el servidor y que sea transparente para los usuarios, como SMTP/TLS (Transport Layer Security – Seguridad de la Capa de Transporte) podría se una buena solución. TLS nos permite crear túneles de cifrado entre servidores cuando la información viaja por las líneas de comunicaciones, sin embargo esta solución implica que los servicios de correo electrónico de otros proveedores comerciales y empresas implementen esta tecnología.

Por lo que pareciera que no existe una solución fácil al problema del tráfico plano en los correos electrónicos, una de las vulnerabilidades que suelen ser explotadas utilizando la funcionalidad de los sistemas de correo electrónico se refiere al proceso que tienen algunas aplicaciones a la hora de recordar la contraseña. La mayoría de las aplicaciones web actuales al recordar contraseña simplemente envían un correo electrónico con la nueva contraseña. Un ejemplo de este funcionamiento lo provee la pagina web de la onidex.

Al darle clic al enlace ¿Olvido su contraseña?, nos muestra lo siguiente:

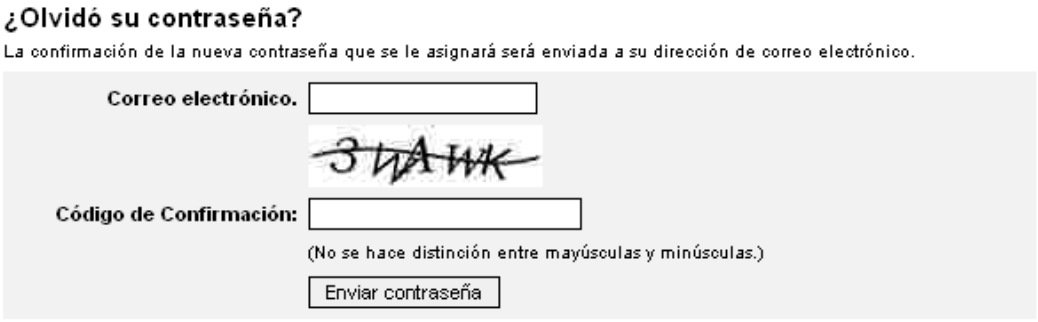


Figura 7.47: Formulario para recuperar contraseña, Onidex (Elaboración propia)

Al colocar la información correspondiente al correo electrónico y imagen captcha, nos envía un correo a la dirección especificada con la siguiente información:

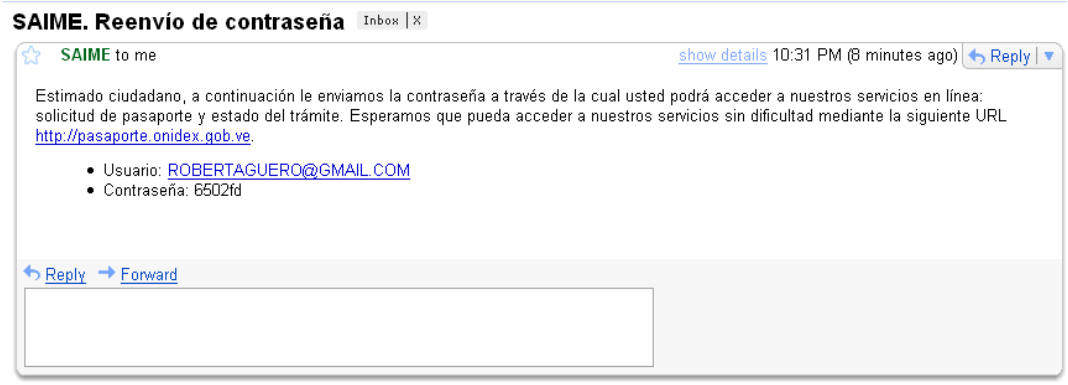


Figura 7.48: Email enviado por la aplicación web, especificando la nueva contraseña (Elaboración propia)

Como se evidencia la nueva contraseña es enviada a la dirección de correo electrónico del usuario, vía email, si existe un ataque de hombre en el medio, y mediante un sniffer se estan capturando los paquetes facilmente un atacante puede obtener la nueva contraseña del usuario y vulnerar la aplicación web.

Dependiendo de la aplicación web existen medidas de seguridad que pueden ser aplicadas, estas medidas pueden ser tan drásticas como: aplicaciones de banca electrónica, como mercantil en línea que al presionar al enlace de ¿Olvido su clave?

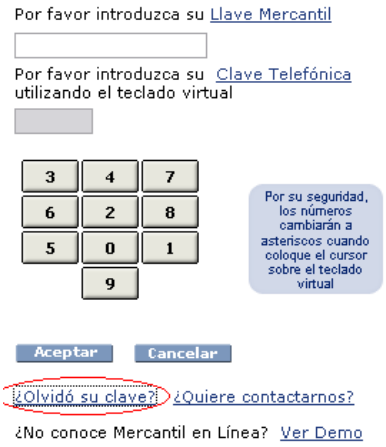


Figura 7.49: Enlace de ¿Olvido su clave? en Mercantil en línea (Elaboración propia)

Muestran la siguiente ventana emergente:

Figura 7.50: Ventana emergente al presionar el enlace ¿Olvido su clave? (Elaboración propia)

La medida de seguridad adoptada por esta aplicación web, es que para poder cambiar su contraseña, hay que dirigirse personalmente a una agencia del banco, aunque esta medida pueda resultar incomoda para un usuario, tiene un nivel de seguridad máximo que es indispensable en este tipo de aplicaciones, que permiten administrar virtualmente el dinero de cada cliente.

Una solución menos dramatica que puede funcionar al recordar la contraseña, sería establecer un canal seguro entre el servidor y el cliente utilizando HTTPS, al momento de que el cliente le de clic al enlace de recordar contraseña se le solicite el correo electrónico, se le envíe un mensaje con un código de confirmación, y se le genere en el servidor un código de confirmación adicional que es enviado al cliente por este canal seguro y mostrado por el navegador, para poder reestablecer la contraseña aparte de colocar el código de confirmación enviado a la dirección del correo electrónico, tambien, sería necesario colocar este código que fue generado por el servidor. Así estaríamos implementando una capa adicional de seguridad a este funcionalidad y no sería tan engorroso y drástico como la aplicación de banca electrónica.

Plantilla para Verificación y comprobación de seguridad de la capa de aplicación:

	Existe	
	Sí	No
Inicio de Sesión de Usuario		
Ataques de Fuerza bruta		
Abuso de funcionalidad		
DoS		
Revelación de información		
Inyección de código o comandos		
Phishing		
Inyección de códigos (SQL, HTML, LDAP, entre otros)		
Captcha		
Filtrado de entrada		
Escapado de salida		
Conexiones seguras		
Manejo de sesión apropiado		
Enmascarado de información sensible		
Contador de intentos		
Registro de conexiones		
Revisión de comentarios en el código		
Manejo de excepciones y mensajes apropiado		
Contraseñas fuertes		
Educación a usuarios		
Recuperación de Contraseña		
Débil Validación en la Recuperación de Contraseñas		
Ataques de Fuerza bruta		
Abuso de funcionalidad		
DoS		
Revelación de información		
Inyección de código o comandos		
Phishing		
Inyección de códigos (SQL, HTML, LDAP, entre otros)		
Captcha		
Filtrado de entrada		
Conexiones seguras		
Manejo de sesión apropiado		
Enmascarado de información sensible		
Revisión de comentarios en el código		
Administradores Remotos		
Autenticación insuficiente		
Autorización insuficiente		
Robo de Sesión		
XSS		
XSRF		
Fijación de Sesión		
Fuga de Información		

Localización de Recursos Predecibles		
Abuso de Funcionalidad		
Autenticación adecuada		
Verificación de Autorización		
Renovación de credenciales		
Certificados digitales		
Registro de usuarios en línea		
Revelación de Información		
Abuso de Funcionalidad		
Inyección de código SQL		
Phishing		
Captcha		
Filtrado de entrada		
Escapado de salida		
Conexiones seguras		
Usuarios mantienen sesión		
Predicción de credenciales/sesión		
Robo de sesión (Session Hijacking)		
Fijación de sesión (Session Fixation)		
XSS		
XSRF		
Escapado de salida		
Verificación de autorización		
Renovación de credenciales		
Caducidad y vencimiento de sesiones		
Políticas de vencimiento y cambio de contraseñas		
Usuarios realizan operaciones		
Abuso de funcionalidad		
Anti-automatización insuficiente		
DoS		
Repudiación de Operaciones		
Fijación de sesión		
Suplantación de contenido		
Cross-Site Request Forgery		
Filtrado de entrada		
Escapado de salida		
Token		
Captcha		
Renovación de credenciales		
Verificación de autorización		
Logs y registro de acciones		
Notificaciones al usuario		

Educación a usuarios		
Sitios Web que se comunican con Usuarios		
Phishing		
Revelación de información		
XSS		
XSRF		
Educación a usuarios		
Escapado de la salida		
Pase de parámetros accesibles al usuario		
Revelación de Información		
Referencia Directa de objetos		
Autorización insuficiente		
Inyección de código		
XSS		
XSRF		
Filtrado de entrada		
Verificación de autorización		
Manipulan información sensible		
Revelación de Información		
Autorización insuficiente		
Cifrado de datos		
Verificación de autorización		
Enmascarado de información sensible		
Autenticidad		
Certificado digital		
SSL,VPN		
Los usuarios suben archivos al servidor		
Versiones viejas o sin parchear		
Deficiente autorización		
Accesos no autorizados y violación de privilegios		
Antivirus		
Captcha		
La aplicación interactúa con base de datos		
Inyección de comandos		
Canal de comunicación no seguro		
Revelación de información		
Resguardo de credenciales		
Filtrado de datos		
Escapado de salida		
Cifrado de datos		
Uso de Plugins (Windows media player, Acrobat, propietarios, .exe, ActiveX, toolbars)		
Versiones viejas o sin parchear		
Deficiente autorización		
Accesos no autorizados y violación de privilegios		
Versiones actuales		
Parches		
Exigir últimas versiones		
Token		
Verificación de autorización		

Glosario

Agente de amenazas

El término es usado para referirse a un individuo o grupo que representa una amenaza. Es fundamental en identificar estos actores y determinar sus posibles intenciones y sus motivaciones. Se pueden clasificar de la siguiente manera:

1. Sin Objetivo Específico: Virus de computadoras, gusanos (Worms), Caballos de Troya, bombas lógicas..
2. Empleados: Personal en general, Contratistas, Operativos, de Mantenimiento e incluso encargados de la seguridad. Cualquiera relacionado directamente con la compañía.
3. Criminales: Estos buscan conseguir información de valor para ellos, como números de tarjetas de crédito, cuentas de banco o propiedad intelectual que pueda convertirse en dinero.
4. Corporaciones: Competidores ofensivos en la guerra de información donde estar mejor informado o conocer estrategias del compañero suponen ventajas competitivas.
5. Humanos No intencionales: Accidentales o sin cuidado.
6. Humanos intencionales: Adentro o Afuera, script kiddies, hackers.
7. Naturales: Fuego, Terremotos, Meteoritos, en fin cualquier tipo de desastre natural.

Alojamiento (Hosting)

Traducción de alojamiento u hospedaje. Las empresas de hosting, son empresas en las que sus clientes pueden alquilar un servidor en la web o un espacio dentro del mismo.

Amenazas

Se refiere a las posibilidades de peligro, en nuestro caso posibilidades de que uno o varios ataques tengan éxito. Si en una aplicación web se conoce que existe determinadas vulnerabilidades y existen ataques para explotar o aprovechar dichas vulnerabilidades, entonces hemos identificado una amenaza, ya que se ha determinado la posibilidad de que ocurra algo perjudicial para la aplicación. Las amenazas siempre van a existir aun y cuando se este protegido contra las mismas.

Aplicación Web

Una aplicación web es una aplicación que accede a una red, como Internet o la intranet,mediante un Navegador o Web Browser. Igualmente, es una aplicación de software que se codifica en un navegador para poder ser ejecutable, apoyándose de varias tecnologías (HTML, PHP, ASP, Perl, etc).

Ataque

Los ataques son las técnicas usadas por los atacantes para explotar o aprovechar vulnerabilidades en aplicaciones. Es importante distinguir entre ataque y vulnerabilidad, ambos términos frecuentemente se confunden. Ataque describe algo que puede realizar un atacante y no una debilidad presente en una aplicación.

Ejemplos de ataque:
Fuerza bruta
Cross Site Scripting (XSS)

Balanceo de carga

Es un concepto usado en informática que se refiere a la técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, discos u otros recursos. Está íntimamente ligado a los sistemas de multiprocesamiento, que hacen uso de más de una unidad de procesamiento para realizar labores útiles.

Capturador de Tramas (Sniffer)

Es un software o hardware que tiene la capacidad de interceptar y registrar el tráfico que pasa a través de una red completa o parte de ella. Un sniffer captura cada paquete, lo decodifica y analiza el contenido acorde al RFC especificado o a otra especificación.

Certificado Digital

Un Certificado Digital es un documento digital mediante el cual un tercero confiable (una autoridad de certificación) garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública. Si bien existen variados formatos para certificados digitales, los más comúnmente empleados se rigen por el estándar UIT-T X.509. El certificado contiene usualmente el nombre de la entidad certificada, un número serial, fecha de expiración, una copia de la clave pública del titular del certificado (utilizada para la verificación de su firma digital), y la firma digital de la autoridad emisora del certificado de forma que el receptor pueda verificar que esta última ha establecido realmente la asociación.

Criptografía

Es el arte o ciencia de cifrar y descifrar información utilizando técnicas que hagan posible el intercambio de mensajes de manera segura que sólo puedan ser leídos por las personas a quienes van dirigidos.

Con más precisión, cuando se habla de esta área de conocimiento como ciencia se debería hablar de criptología, que engloba tanto las técnicas de cifrado, la criptografía propiamente dicha, como sus técnicas complementarias: el criptoanálisis, que estudia los métodos que se utilizan para romper textos cifrados con objeto de recuperar la información original en ausencia de las claves.

Denegación de Servicios (Denial of Services)

En seguridad informática, un ataque de denegación de servicio, también llamado ataque DoS (de las siglas en inglés Denial of Service), es un ataque a un sistema de ordenadores o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima.

DHCP

Siglas en inglés de Dynamic Host Configuration Protocol. Es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van siendo demandadas y van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

DMZ

Una DMZ (del inglés Demilitarized zone) o Zona DesMilitarizada. En seguridad informática, una zona desmilitarizada (DMZ) o red perimetral es una red local (una subred) que se ubica entre la red interna de una organización y una red externa, generalmente Internet. El objetivo de una DMZ es que las conexiones desde la red interna y la externa a la DMZ estén permitidas, mientras que las conexiones desde la DMZ sólo se permitan a la red externa – los equipos (hosts) en la DMZ no pueden conectar con la red interna. Esto permite que los equipos (hosts) de la DMZ's puedan dar servicios a la red externa a la vez que protegen la red interna en el caso de que intrusos comprometan la seguridad de los equipos (host) situados en la zona desmilitarizada. Para cualquiera de la red externa que quiera conectarse ilegalmente a la red interna, la zona desmilitarizada se convierte en un callejón sin salida. La DMZ se usa habitualmente para ubicar servidores que es necesario que sean accedidos desde fuera, como servidores de e-mail, Web y DNS.

Envenenamiento de la caché DNS (DNS poisoning)

Se trata del falseamiento de una relación N̈ombre de dominio-IPänte una consulta de resolución de nombre, es decir, resolver con una dirección IP falsa un cierto nombre DNS o viceversa. Esto se consigue falseando las entradas de la relación Nombre de dominio-IP de un servidor DNS, mediante alguna vulnerabilidad del servidor en concreto o por su confianza hacia servidores poco fiables. Las entradas falseadas de un servidor DNS son susceptibles de infectar (envenenar) el caché DNS de otro servidor diferente.

Frame Relay

Es una técnica de comunicación mediante retransmisión de tramas, introducida por la ITU-T a partir de la recomendación I.122 de 1988. Consiste en una forma simplificada de tecnología de conmutación de paquetes que transmite una variedad de tamaños de tramas o marcos (frames) para datos, perfecto para la transmisión de grandes cantidades de datos. CIR (Commited Information Rate) EIR (Excess Information Rate) DLCI (Data Link Connection Identifier)

Funciones de Resumen o de una sola dirección (Hash)

Son procedimientos bien definidos o funciones matemáticas para convertir entradas de datos en enteros relativamente pequeños o resumen. Tiene la característica de que desde el resumen matemáticamente no se puede llegar al origen, sin embargo es vulnerable a ataques de fuerza bruta.

Hackers

Los hackers son personas relacionadas con la seguridad de sistemas, bien sea para protegerlos o atacarlos. Se especializan en descubrir vulnerabilidades en sistemas y en obtener o prevenir accesos no autorizados a los mismos, haciendo uso de habilidades, técnicas y conocimiento especializado.

Hash

Se refiere a una función o método para generar claves o llaves que representen de manera casi unívoca a un documento, registro, archivo, etc., resumir o identificar un dato a través de la probabilidad, utilizando una función hash o algoritmo hash. Un hash es el resultado de dicha función o algoritmo.

HTTP

El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido. También sirve el protocolo para enviar

información adicional en ambos sentidos, como formularios con campos de texto. HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Por esto se popularizaron las cookies, que son pequeños archivos guardados en la propia computadora que pueden leer un sitio web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente. Gracias a esta identificación, el sitio web puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio. La versión actual de HTTP es la 1.1, y su especificación está en el documento RFC 2616.

ISP

Un proveedor de servicios de Internet (o ISP, del inglés Internet Service Provider) es una empresa dedicada a conectar a Internet a los usuarios o las distintas redes que tengan, y dar el mantenimiento necesario para que el acceso funcione correctamente. También ofrecen servicios relacionados, como alojamiento web o registro de dominios entre otros.

LAN

Es la abreviatura de Local Area Network (Red de Área Local). Una red local es la interconexión de varios ordenadores y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de unos pocos kilómetros. Su aplicación más extendida es la interconexión de ordenadores personales y estaciones de trabajo en oficinas, fábricas, etc; para compartir recursos e intercambiar datos y aplicaciones. En definitiva, permite que dos o más máquinas se comuniquen. El término red local incluye tanto el hardware como el software necesario para la interconexión de los distintos dispositivos y el tratamiento de la información.

Medidas de seguridad (Countermeasure)

Son acciones defensivas usadas para eliminar, reducir o mitigar los riesgos de ataques y vulnerabilidades. En otras palabras se traducen en tecnologías o módulos que podemos usar para prevenir, disminuir o no permitir ataques. Es necesario identificar e implementar medidas de seguridad basado en un análisis de amenazas, para asegurarnos que la aplicación esta protegida contra tipos comunes de ataques que estas amenazas representen. La deficiencia o falta en medidas de seguridad puede resultar en una vulnerabilidad.

Phishing

Es un término informático que denomina el uso de un tipo de ingeniería social caracterizado por intentar adquirir información confidencial de forma fraudulenta (como puede ser una contraseña o información detallada sobre tarjetas de crédito u otra información bancaria). El estafador, mejor conocido como phisher, se hace pasar por una persona o empresa de confianza en una aparente comunicación oficial electrónica, por lo común un correo electrónico, algún sistema de mensajería instantánea o incluso utilizando también llamadas telefónicas. Dado el creciente

número de denuncias de incidentes relacionados con el phishing se requieren métodos adicionales de protección. Se han realizado intentos con leyes que castigan la práctica, campañas para prevenir a los usuarios y con la aplicación de medidas técnicas a los programas.

Sistema de Dominios de Nombres (DNS)

El DNS (del inglés Domain Name System) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio. La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Por ejemplo, si la dirección IP del sitio FTP de prox.ve es 200.64.128.4, la mayoría de la gente llega a este equipo especificando ftp.prox.ve y no la dirección IP. Además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre.

SSL

Secure Sockets Layer SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes. Los protocolos permiten a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas (eavesdropping), la falsificación de la identidad del remitente (phishing) y mantener la integridad del mensaje.

Suplantar Contenido (Defacement)

Es un ataque que consiste en suplantar el contenido de una página web o portal. Muchos atacantes hacen esto con la intención de ganar admiración y respeto dentro de su comunidad de hackers.

VPN

VPN (Virtual Private Network) es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet. El ejemplo más común es la posibilidad de conectar dos o más sucursales de una empresa utilizando como vínculo Internet, permitir a los miembros del equipo de soporte técnico la conexión desde su casa al centro de cómputo, o que un usuario pueda acceder a su equipo doméstico desde un sitio remoto, como por ejemplo un hotel. Todo esto utilizando la infraestructura de Internet.

Vulnerabilidad

Una vulnerabilidad es un agujero o debilidad en la aplicación, que puede ser una deficiencia de diseño o desperfecto de implementación, la cual permite a un atacante causar daño a cualquier parte de una aplicación web. Estas partes pueden ser los dueños de la aplicación, los usuarios de la aplicación o cualquier otra entidad que se relacione con la aplicación.

Ejemplos de vulnerabilidad:
Deficiencia o falta de validación de datos de entrada del usuario.
Deficiencia o falta de mecanismos de autenticación.

WAN

Una red de área amplia, WAN, acrónimo de la expresión en idioma inglés Wide Area Network, es un tipo de red de computadoras capaz de cubrir distancias desde unos 100 hasta unos 1000 km, dando el servicio a un país o un continente. Un ejemplo de este tipo de redes sería RedIRIS, Internet o cualquier red en la cual no estén en un mismo edificio todos sus miembros (sobre la distancia hay discusión posible). Muchas WAN son construidas por y para una organización o empresa particular y son de uso privado, otras son construidas por los proveedores de internet (ISP) para proveer de conexión a sus clientes.

Bibliografía

[1 & 1 , 2008] 1 & 1 The world's #1 web host: Data center
Consultado en http://http://order.1and1.com/xml/order/DataCenter

[Acunetix , 2008] Acunetix: Which Vulnerabilities does Acunetix WVS Check for?
Consultado en http://www.acunetix.com/support/vulnerability-checks.htm

[Acunetix , 2008] Acunetix: Google hacking.
Consultado en http://www.acunetix.com/websitesecurity/google-hacking.htm

[Help Net Security, 2008] Help Net Security: Top 10 vulnerabilities in Web applications in Q1 2007
Consultado en http://www.net-security.org/secworld.php?id=5184

[Chris Shiflett, 2005] Chris Shiflett, Octubre 2005. *Essential PHP Security*. O'Reilly

[Eric Freeman & Elisabeth Freeman, 2005] Eric Freeman & Elisabeth Freeman, Octubre 2005. *Head First Design Patterns*. O'Reilly

[Instituto seguridad Internet, 2008] Instituto seguridad Internet: Auditoría de aplicaciones web
Consultado en http://www.instisec.com/publico/auditoria.asp?id=2

[Introduction to Web Security, 2008] Google: Introduction to Web Security
Consultado en http://code.google.com/edu/submissions/web_security/listing.html

[Kevin Beaver, 2005] Kevin Beaver, May 2005. *Hacking For Dummies*. Wiley Publishing, Inc

[Mike Danseglio, 2005] Mike Danseglio, Noviembre 2005. *Securing Windows Server 2003*. O'Reilly

[Microsoft , 2008] Microsoft: Security Central
Consultado en http://www.microsoft.com/security/default.mspx

[Microsoft , 2008] Microsoft: Audita la seguridad de una aplicación Web ASP.NET o un servicio Web, 2008.
Consultado en http://support.microsoft.com/kb/815144/

[OWASP, 2008] OWASP: Página web de la comunidad abierta de seguridad en aplicaciones
Consultado en http://www.owasp.com

[Qualys, 2008] Qualys: Top ten vulnerabilities
Consultado en http://www.qualys.com/

[SANS, 2008] SANS: Top-20 2007 Security Risks FAQ
Consultado en http://www.sans.org/top20/

[Seguridad en la red, 2008] Seguridad en la red
Consultado en http://www.seguridadenlared.org/

[WASC, 2005] Web Application Security Consortium
Consultado en http://www.webappsec.org/

[WSAC, 2005] Web Application Security Consortium: Threat Clasification
Consultado en http://www.webappsec.org/projects/threat/

[Wikipedia] Wikipedia
Consultado en http://www.wikipedia.com/