Examen transversal

Seguridad en sistemas computacionales 021D

Hombre parado junto a una camioneta

Descripción generada automáticamente con confianza media

Corporación Security LTDA

Caso cliente “Tu auto”

Integrantes:

* Fabian Muñoz – 20.146.798-5
* Matias Perez – 21.315.944-5
* Gabriel Soto – 20.532.134-9

Seccion: ASY6131\_021D

Profesor a cargo: Daniel Alonso Montero Valenzuela

Contenido

[1. Introducción 3](#_Toc153151476)

[2. Definiciones 4](#_Toc153151477)

[3. Propuesta de plan y política de ciberseguridad a la organización 7](#_Toc153151478)

[3.1. Levantamiento e identificación de activos de información y procesos de negocio 7](#_Toc153151479)

[3.2. Evaluación de rutas 7](#_Toc153151480)

[3.3. Análisis y evaluación de mecanismos y herramientas de seguridad organizacional 11](#_Toc153151481)

[3.4. Estructura de gobierno corporativo 11](#_Toc153151482)

[3.5. Políticas de desarrollo seguro (aplicación leasing automotriz) 12](#_Toc153151483)

[3.5.1. Protección de servicios de aplicación en redes pública 12](#_Toc153151484)

[3.5.2. Ambiente de desarrollo seguro 13](#_Toc153151485)

[3.5.3. Separación de entornos de desarrollo, prueba y operacionales 14](#_Toc153151486)

[3.5.4. Inventario de activos 14](#_Toc153151487)

[3.5.5. Devolución de activos 14](#_Toc153151488)

[3.5.6. Accesos a la red y servicios de la red 14](#_Toc153151489)

[3.5.7. Trabajos del evaluador PCI DSS 14](#_Toc153151490)

[3.5.8. Proteger datos de las tarjetas existentes con criptografía 15](#_Toc153151491)

[3.5.9. Desarrollar y mantener sistemas y software seguros 16](#_Toc153151492)

[3.6. Periodicidad de evaluación y revisión 16](#_Toc153151493)

[3.7. Difusión 16](#_Toc153151494)

[4. Gestión de vulnerabilidades 16](#_Toc153151495)

[4.1. Análisis de vulnerabilidades 17](#_Toc153151496)

[4.2. Clasificación de vulnerabilidades según su criticidad 21](#_Toc153151497)

[4.3. Explotación de vulnerabilidad 25](#_Toc153151498)

[4.4. Definir posibles soluciones 36](#_Toc153151499)

[5. Plan de mitigación 38](#_Toc153151500)

[5.1. Identificación, análisis y priorización de procesos y servicios críticos TI que forman parte del alcance 38](#_Toc153151501)

[5.1.1. Procesos de negocio críticos y los servicios – aplicaciones TI que lo soportan 39](#_Toc153151502)

[5.1.2. Identificación de dependencias de los servicios TI 39](#_Toc153151503)

[5.1.3. Documentación de arquitectura de sistemas y dependencias 40](#_Toc153151504)

[5.1.4. Tiempos y puntos de recuperación requeridos y factibles para sistemas y dependencias 41](#_Toc153151505)

[5.2. Estrategias de recuperación 42](#_Toc153151506)

[5.3. Procedimiento de alerta, escalamiento, activación y gobierno de la recuperación TI 42](#_Toc153151507)

[5.3.1. Diagramas de flujo 43](#_Toc153151508)

[5.3.2. Descripción de equipos y roles 43](#_Toc153151509)

[5.3.3. Descripción resumida del flujo de escalamiento 43](#_Toc153151510)

[5.4. Procedimientos de contingencia y recuperación 44](#_Toc153151511)

[5.4.1. Objetivo 44](#_Toc153151512)

[5.4.2. Alcance 44](#_Toc153151513)

[5.4.2.1. Procedimientos relacionados 44](#_Toc153151514)

[5.5. Desarrollo del procedimiento 45](#_Toc153151515)

[5.5.1. Fase de transición a operación en contingencia (failover) 45](#_Toc153151516)

[5.5.2. Fase de operación en contingencia 46](#_Toc153151517)

[5.5.3. Fase de transición a operación normal o vuelta atrás (failback) 46](#_Toc153151518)

[5.6. Detalle de tareas 46](#_Toc153151519)

[5.6.1 Ejemplo de restauración 46](#_Toc153151520)

[5.7. Anexo 58](#_Toc153151521)

[5.7.1. Nómina de personal crítico con datos de contacto 58](#_Toc153151522)

[5.7.2. Nómina de proveedores críticos con bases de datos 58](#_Toc153151523)

[6. Referencias 58](#_Toc153151524)

1. Introducción

La principal finalidad de este documento escrito por *Corporacion Security LTDA* es lograr asegurar la seguridad de la información de la empresa, buscando evitar la pérdida de información y fuga de datos, dando a conocer cuáles son los procesos más relevantes y que necesitan más prioridad dentro de la organización “Tu auto” y sabiendo como actuar en caso de algún tipo de desastre que afecte la integridad del servicio.

Por lo que mediante este escrito se documentara e instruirá cuáles serán las políticas de desarrollo para mantener un entorno seguro respecto a los futuros proyectos a ejecutar, se informaran y analizaran las vulnerabilidades presentes en todo el sistema, adicionalmente, se explicara el cómo estas fallas han sido explotadas y finalmente se hará un plan de procedimiento a ejecutar ante un desastre

1. Definiciones

* estándares
  + ISO: estándares con reconocimiento nacional y/o internacional, busca establecer un nivel de homogeneidad a un desarrollo, gestión o servicio
  + PCI-DSS: estándar global de seguridad para la industria de tarjetas de pago
  + OWASP: estándar de programación segura, busca promover buenas prácticas y técnicas
  + CVSS: Sistema de puntaje de vulnerabilidades comunes. Busca clasificar vulnerabilidades encontradas en escalas y darles una etiqueta con su nivel de peligrosidad, siendo el nivel crítico el más alto
  + POSIX: estándar creado a partir de las batallas legales de unix que busca permitir la compatibilidad entre sistemas basados en unix.
* Tecnicismos
  + Software: Conjunto de piezas de programación y Código que se distribuye
  + Hardware: Piezas y componentes físicos que en conjunto forman un servidor, computador o maquina
  + Firmware: Código de bajo nivel instalado en el hardware para que este funcione de una manera adecuada
  + BD – BBDD – DB: Siglas de base de datos
  + DBA: Siglas de data base administrator, es el que administra una base de datos
  + API: Siglas de Aplication programming interface, es la interfaz de aplicación de programación. Permite conectar dos sistemas independientes para hacer un software más complejo
  + Frontend: Parte de un software en la que está presente toda la interacción con el usuario, como por ejemplo menús, botones, imágenes, etc. Normalmente se ejecuta en la máquina del cliente
  + Backend: Parte de un software en la que está presente toda la interacción con la base de datos y el manejo de estos, también aquí se encuentra la mayoría de la lógica de un programa. Normalmente se ejecuta en un servidor
  + Honeypot: Es una forma de poder atraer a atacantes y hackers a un servidor especifico puesto con fines estratégicos o experimentales. Suele ser un servidor con vulnerabilidades puestas a propósito.
  + Distribución: En sistemas operativos basados en GNU/LINUX, es el conjunto de diversos programas que se encargan de hacer una sola tarea especifica bien hecha para hacer un sistema operativo completo.
  + Exploit: Literalmente se llama explotar, en este contexto es explotar una vulnerabilidad de un sistema informático para un fin propio o malicioso. En este caso, buscaremos explotar alguna vulnerabilidad para el fin propio de la empresa.
  + Payload: Literalmente se llama carga, en este contexto será la carga que tendrá el exploit para poder aprovechar la vulnerabilidad.
  + Leasing: Intercambio, en este contexto se refiere a una aplicación de intercambio automotriz.
  + Pentest: Significado de pruebas de penetración, en este contexto se refiere a probar si es posible penetrar un sistema haciendo pruebas con ciertas técnicas y softwares.
  + SQL injection: técnica que consiste en aprovechar la vulnerabilidad de un software para ejecutar consultas en las que se respeta la lógica y sintaxis del lenguaje SQL y poder obtener datos adicionales en donde no deberían estar, o incluso en algunos casos es posible inyectar Código en campos que no se debería poder.
  + IDS: Siglas de sistema de detección de intrusos.
  + IPS: Siglas de sistema de prevención de intrusos.
  + SSH: Siglas de Secure Shell. Es un protocolo de red en el que permite conectarse de manera remota a una terminal de un servidor o computador en funcionamiento.
  + IP: Siglas de Protocolo de internet, se encarga de brindar a diversos dispositivos un valor de dirección única en toda la red. Tiene dos versiones que se siguen usando en la actualidad (ipv4 y ipv6)
  + Firewall: Software y/o hardware hecho para proteger y filtrar conexiones externas de una máquina, bloqueando y permitiendo puertos y direcciones.
  + Hardening: técnicas hechas para poder mitigar, dificultad o denegar algún ataque informático, estas técnicas consisten en establecer configuraciones en un servidor como estableciendo políticas para los trabajadores tomen ciertos procedimientos ante las situaciones indicadas.
* Sistemas
  + Linux: también llamado GNU/LINUX Es una pieza de los sistemas operativos gratuitos basados en el estándar POSIX que permite la conexión entre el hardware y el software de una máquina.
  + IRC: Software que permite la comunicación en tiempo real entre varias personas en una sala de chat
  + Base de datos: Software hecho para administrar información, normalmente suelen trabajar en conjunto con el backend de un software.
  + Servidor: En este contexto, un servidor es un sistema físico que aloja los servicios necesarios para el funcionamiento del negocio, nuestro cliente aloja los servicios de su página web en un servidor vulnerable
  + CMS: Siglas de Content Managment System. Es un software que permite gestionar el contenido de un servicio o página web de una manera más amigable teniendo una conexión directa con la base de datos. Cumple la función de CRUD.
  + CRUD: Siglas de CREATE READ UPDATE DELETE. Concepto utilizado en el desarrollo de sistemas en el que se estandarizan las siglas definidas con anterioridad para poder manipular la información existente en la base de datos.
  + Teenable Nessus: Servicio que provee herramientas de análisis de vulnerabilidades de redes y servidores.
  + Metasploit-framework: Programa que facilita la explotación de vulnerabilidades mediante comandos como msfvenom o msfconsole
  + NMAP: Programa que permite analizar las redes y direcciones para obtener la lista de los puertos abiertos que tenga una máquina, con los servicios y protocolos que está ejecutando en los puertos.
  + Rsnapshot: Programa que permite la automatización de respaldos en servidores, proviene de la familia de programas de respaldos Rsync
  + NGINX:Servicio que permite utilizar los protocolos HTTP y HTTPS para levantar páginas webs o compartir directorios. Su competencia es Apache HTTP server project que cumple el mismo cometido
  + HTTP: Protocolo de red, sus siglas son Hypertext tansfer protocol, o protocolo de transferencia de hipertexto. Existe una variante de este protocolo que se llama HTTPS y trabaja de una manera que permite el uso de certificados SSL para habilitar una comunicación cifrada y segura entre cliente - servidor
  + PHP: Lenguaje de programación enfocada a backend y desarrollo de páginas web, en este contexto es este el lenguaje usado para la página de nuestro cliente.
  + SSL: Es un certificado digital en el que una organización especifica los puede crear. Pueden ser distribuidos comercial o gratuitamente mediante corporaciones para ser instalados en un servidor. Estos certificados tienen información acerca del servidor, dueño del servidor y fecha de expiración. El navegador de un cliente es el que tiene listada la información de empresas y organizaciones que emiten los certificados SSL y esta información es actualizada con frecuencia, por lo que, si un navegador no tiene en su lista de certificados SSL el certificado de una página, la conexión HTTPS será rechazada y el navegador saltará una Alerta. Del mismo modo, si el dueño de la página y del certificado ha hecho cosas maliciosas, este certificado puede revocarse tanto de parte de la empresa que emitió ese certificado como del navegador. Esto asegura que las páginas no se den chance a actos maliciosos, y en caso de que lo hagan, el navegador notificara al cliente.
  + SHA-2: Siglas de Secure Hash Algorithm,o algoritmo de Hash seguro. SHA-2 es una familia de los algoritmos SHA, son utilizados para firmar digitalmente un fichero o archivo, asegurando la integridad de que el archivo distribuido es genuinamente real y no una copia modificada. Tambien es utilizado para conexiones remotas como para validar llaves de SSH, asi como también se usa para validar la originalidad de las distribuciones de programas hechas por comunidades de usuarios en los diversos sistemas de paqueteria de los sistemas operativos basados en linux.

1. Propuesta de plan y política de ciberseguridad a la organización
   1. Levantamiento e identificación de activos de información y procesos de negocio

Nuestro cliente, automotoras “Tu auto” posee diversos servicios y proyectos a desarrollar, uno de estos servicios es una pagina web y otro de los proyectos a desarrollar es el de una aplicacíon en el que el objetivo sera el facilitar el prestamo de automoviles mientras un cliente pueda pagarlo hasta que sea el haya pagado el automovil por completo.

El Gerente ha manifestado que la pagina web ultimamente ha pasado por malos momentos respecto a lo que seria la seguridad informatica, debido a que han descubierto que han sufrido de fugas de datos y muchos de estos son información critica y sensible, por lo que han decidido poner manos en el asunto.

Nosotros, como corporacion de seguridad *Corporacion Security LTDA* Definiremos un plan con las tareas a ejecutar, en el que haremos diversas auditorias y planes para asegurar la integridad de los datos y de los servicios. Las tareas a ejecutar seran las siguientes:

* La redacción de las politicas de desarrollo seguro. Esta politica aplicara tanto para el desarrollo de la aplicación de prestamos de automoviles, basandose en las normalizaciones ISO 27001, ISO 27002 y OWASP
* Identificar las vulnerabilidades del servidor de nuestro cliente con las herramientas necesarias de analisis y pentesting.
* Una vez identificadas las vulnerabilidades del servidor, se redactara un plan de mitigación ante los casos encontrados, ademas de los posibles cassos que puedan pasar.
  1. Evaluación de rutas

La evaluación de rutas abordara los problemas de seguridad presentes en la pagina web y la apliación que se desarrollara para el proceso relacionado con el leasing automotriz.

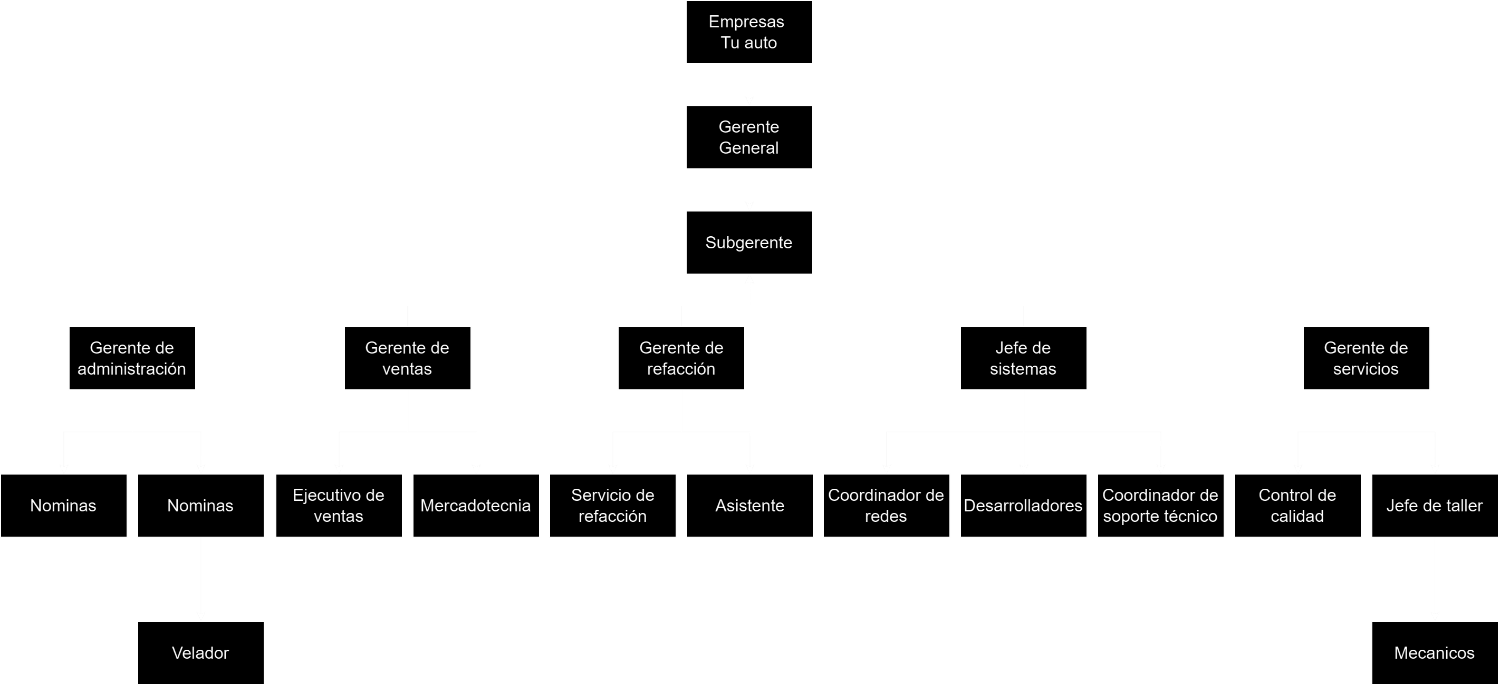
* Abordando los problemas de seguridad de la pagina web
  + Objetivos
    - Reconocer y clasificar los objetivos de seguridad para abordar los problemas en la pagina web
    - Definir los criterios de severidad y los procesos de notificación para los eventos de seguridad, o desastres que afecten la integridad de los sistemas
  + Análisis de incidentes actuales
    - Analizar y buscar en los incidentes de seguridad que han sucedido con anterioridad en la página web
    - Reconocer patrones, orígenes de los ataques, vectores de ataques y posibles vulnerabilidades explotadas
  + Investigación externa
    - Usar uso de motores de busquéda como Google u otras fuentes y tecnologías como inteligencia artificial (Chat-gpt Google bard, etc. Sin embargo, no fiarse de los resultados de esta última, solamente cuando puedan ser validados) para investigar tecnicas y procedimientos asociados a ataques similares
    - Analizar las mejores prácticas y recomendaciones de seguridad de la industria y lideres TI, de empresas como Meta, X, Alphabet, Google, Microsoft, etc.
  + Evaluación de infraestructura
    - Revisar la infraestructura del servidor y el servidor web nginx
    - Reconocer y dar soluciones a las vulnerabilidades conocidas en el sistema operativo y el software que se está utilizando.
  + Desarrollo de cirterios de severidad
    - Los equipos de desarrollo de “Tu auto” con el equipo de “Corporacion Security LTDA” deben trabajar en conjunto para establecer los criterios de severidad para los incidentes de seguridad.
    - Categorizar los umbrales y las acciones específicas basadas en la gravedad de los eventos.
  + Proceso de escalamiento
    - Establecer el proceso de escalamiento de forma clara para manejar los incidentes críticos.
    - Definir los roles y las responsabilidades para cada nivel.
  + Capacitación del personal
    - Capacitar a todo el personal de desarrollo y seguridad sobre las nuevas politicas y procedimientos
    - Asegurar que todos los miembros de la empresa relacionados con el proceso de negocio de la pagina web esten familiarizados con las formas de notifación y el escalamiento.
* Abordando el futuro desarrollo de la aplicación de leasing
  + Objetivos
    - Reconocer y definir los objetivos para el desarrollo de la nueva aplicación
    - Establecer un marco referencial para la codificación segura
  + Colaboración con el equipo de desarrollo
    - Los equipos de desarrollo de “Tu auto” con el equipo de “Corporacion Security LTDA” deberan trabajar en conjunto para establecer y comprender los requisitos del nuevo proyecto.
    - Reconocer los posibles reisgos de seguridad relacionados con el desarrollo del proyecto.
  + Desarrollo de marco referencial
    - El marco referencial debe incluir las practicas de codificación segura
    - Incluir estándares de seguridad establecidos por las normas ISO 27001, ISO 27002 y OWASP en el desarrollo.
  + Revisión de diseño de aplicación
    - Revisar el diseño de la aplicación desde una perspectiva de seguridad, una vez revisado determinar si el diseño esta aprobado, a modificar o rechazado
    - Identificar y corregir todas las posibles vulnerabilidades en las fases iniciales del desarrollo
  + Pruebas de seguridad
    - Aplicar pruebas de seguirdad regulares durante el ciclo de desarrollo.
  + Capacitación en desarrollo seguro
    - Dar formación al equipo de desarrollo sobre las mejores prácticas de desarrollo seguro (Si es posible ejercer una certificación, mejor)
    - Complementar la seguridad en cada etapa del ciclo de vida del proyecto.
  + Implementación de procesos de seguridad
    - Ejecutar medidas de seguridad proactivas, como un frecuente monitoreo y dar respuesta a incidentes.

Del mismo modo, tambien se necesitan definir los distintos requerimientos de seguridad en las distintas etapas del ciclo de desarrollo. Estas serian:

* Análisis
  + Requerimientos funcionales de seguridad
    - Reconocer y documentar los requisitos funcionales de seguridad
    - Añadir autenticación, autorización y adutoría de actividades críticas
  + Evaluaciones de riesgos
    - Realizar evaluaciones de reisgos para identificar las posibles amenazas y vulnerabilidades
    - Documentar los riesgos para abordarlos de forma adecuada
  + Definición de datos sensibles
    - Identificar y etiquetar los datos que la aplicación manejará
    - Las políticas seran claras para el manejo y la protección de la información
  + Requisitos de privacidad
    - Asegurar que la aplicación cumpla con los requsisitos de privacidad aplicables
    - Añadir sistema de control con la finalidad de proteger la informacion personal identificable
* Diseño del software
  + Arquitectura segura
    - La arquitectura debe incluir capas de seguridad.
    - Se deben implementar controles de seguridad en los componentes del sistema
  + Controles de acceso
    - Establecer políticas de control de acceso basadas en los roles y privilegios de grupos y usuarios
    - Establecer mecanismos de autentificación fuerte cuando sea necesario. Como por ejemplo Autentificación de dos pasos, validación por SMS, un tercer paso puede ser el uso de datos biometricos como la huella digital, escaneo de iris, etc.
  + Encriptación de datos
    - Especificar el uso de técnicas de encriptación adecuadas para la proteccion de datos almacenados y cuando se esten procesando.
    - Asegurarse de que los algoritmos y claves utilizadas sean robustos.
  + Auditorías y registro de eventos
    - Integrar funciones de auditoría para registrar eventos de seguridad
    - Asegurarse de que los registros tengan detalle, esten protegidos y sean revisados con frencuencia
  + Manejo de errores seguro
    - Implementar una regla de manejo de errores para que este no releve información sensible
    - Registrar los mensajes de error, de forma que no facilite la explotación de las vulnerabilidades
* Codificación del software
  + Validaciónes de entradas
    - Validar y limpiar todas las entrads del usuario para prevenir ataques de inyección
    - Añadir restricciones de formato y longitud en los campos de entrada
  + Prevencion de ataques de scripting
    - Utilizar funciones de seguridad en el backend y validar las entradas en el frontend.
  + Manejo seguro de las sesiones
    - Implementar prácticas seguras para el manejo de sesiones, como expiración de tokens y cookies y almacenamiento de credenciales
    - Las cookies deberan almacenar de forma encriptada usando algun algoritmos de la familia SHA-2 un poco de metadata del cliente, como por ejemplo una dirección mac de red. Esto con la finalidad de que en caso de que se secuestren la información de las cookies de un cliente y se usen en un equipo desconocido, el sistema checkee que la dirección mac de la sesión actual con la dirección mac indicada por la cookies sean las mismas. En caso de que el sistema determine que son diferentes, la sesión se cerrara automaticamente y se notificara al cliente que es probable de que haya sufrido de un secuestro de información.
  + Control detallado de errores
    - Ajustar la aplicación para que presente los mensajes de error genéricos al usuario mientras que se registren los detalles mas especificos en los registros internos.
    - Monitorear y observar regularmente los registros de errores
  + Pruebas de seguridad automatizadas
    - Añadir pruebas de seguridad automatizadas en el proceso de construcción continua mediante el uso de Cronjob
    - Hacer uso de herramientas de análisis estático y dinámico para identificar las vulnerabilidades en el código.
  1. Análisis y evaluación de mecanismos y herramientas de seguridad organizacional

Los mecanismos y herramientas destinados a la seguridad organizacional se pueden emplear para garantizar el adecuado estado y funcionamiento de los sistemas. Estos abarcan desde medidas mitigadoras y detectores de ataques informáticos hasta planes y procedimientos para hacer frente a situaciones de desastre, como terremotos, lluvias intensas o incendios, que inicialmente podrían escapar al control. En el caso del cliente "TU AUTO", se implementarán las siguientes herramientas y mecanismos:

* Establecimiento y mantenimiento de sistemas antivirus. En esta ocasión, se utilizará Norton para los equipos empleados por los operarios del sistema.
* Configuración de controles de acceso en cada servidor, ya sea para servidores SFTP, SSH u otros sistemas más complejos como un CMS. Se puede habilitar el gestor de contenido del framework de Django o incluso el que viene incorporado con Wordpress, pero se recomienda utilizar uno dedicado para servidores
* Implementación de Firewalls en cada punto de red de forma independiente, asegurándose de que cada servidor tenga su propio firewall. En esta situación, se sugiere la utilización de sistemas de FortiGate o Sophos.
* Realización periódica de copias de respaldo de todos los servidores, almacenándolas en un servidor ubicado físicamente en un lugar distinto. Se recomienda la utilización de Rsnapshot como gestor de respaldos del servidor.
  1. Estructura de gobierno corporativo



* 1. Políticas de desarrollo seguro (aplicación leasing automotriz)

(Control A.14.2.1 NCh ISO 27001:2013)

La Política de desarrollo seguro de Tu auto comprende las reglas para el desarrollo de software y sistemas dentro de la organización. Para esto, se establecen los siguientes lineamientos:

Se deberán utilizar técnicas de programación seguras tanto para los desarrollos nuevos como en las situaciones de reutilización de códigos donde es posible que no se conozcan las normas que se aplican al desarrollo o donde no sean coherentes con las buenas prácticas actuales. Lo anterior, tanto para el desarrollo interno como externo.

Se debe estandarizar el ciclo de vida del desarrollo de software en empresas Tu auto, logrando con ello los siguientes objetivos

Definir actividades a llevarse a cabo en un proyecto de desarrollo de software.

Unificar criterios en la organización para el desarrollo de software.

* Objetivos
  + El propósito de esta política es garantizar la seguridad de la información de los clientes y diferentes softwares desarrollados por” Tu auto” a través de un conjunto de estándares ya establecidos y certificados dentro del mercado. Para cumplir con este propósito, debemos considerar la seguridad de la información como parte integral del ciclo de vida de los diferentes tipos de software que se desarrollan en ”Tu auto”.
* Alcances
  + La presente política de desarrollo seguro se dicta en cumplimiento de las disposiciones legales vigentes, con el objetivo de gestionar adecuadamente la seguridad de la información de los clientes y los sistemas de “Tu auto”. Esta política es de aplicación obligatoria e inalienable para todos los funcionarios y proveedores de “Tu auto”, independiente de la jerarquía en la que se encuentre el empleado dentro de la compañía. De igual manera, la política debe ser conocida por todos los funcionarios de la empresa, sean éstos en calidad de planta, bomba, oficinas, etc. que cumplan sus funciones en dependencia de empresas “Tu Auto”, así como por los proveedores externos que tengan acceso mediante vínculos tareas de apoyo o asesoría externa a la empresa.
    1. Protección de servicios de aplicación en redes pública

(Control A.14.1.2 NCh ISO 27001:2013)

La información involucrada en los servicios de aplicación que pasan a través de redes públicas se deberá proteger contra la actividad fraudulenta, la disputa de contratos y la información/modificación no autorizada.

El software, datos y otra información que requiera un alto nivel de integridad y que estén accesibles públicamente se deben proteger por mecanismos adecuados. Los sistemas accesibles públicamente se deben probar contra debilidades y fallas antes que la información esté disponible.

Debe haber un proceso de aprobación formal antes que la información esté accesible públicamente. Además, toda la entrada proveniente del exterior al sistema deberá ser verificada y aprobada.

* Los sistemas electrónicos de validación de tarjetas, sobre todo aquellos que permiten la retroalimentación y el ingreso directo de información, se deben controlar con cuidado de modo que:
* La información se obtenga en cumplimiento con toda la legislación sobre protección de datos.
* El ingreso de la información a, y el procesamiento por, el sistema de edición será procesado completamente y con exactitud de manera oportuna.
* La información sensible será protegida durante la recolección, procesamiento y almacenamiento
* El acceso al sistema de tarjetas no debe permitir el acceso no planeado a redes las cuales se conecta al sistema.
* Se debe diferenciar, entre el encargado de celebrar y autorizar los contratos con terceros, de los que deben fiscalizar su cumplimiento
  + 1. Ambiente de desarrollo seguro

(Control A.14.2.6 NCh ISO 27001:2013)

El entorno de desarrollo seguro en “Tu Auto” considera los siguientes aspectos de seguridad de la información en:

* La fase de diseño de los proyectos de desarrollo de software y sistemas
* El entorno de desarrollo, identificado como el conjunto de procesos y herramientas que se utilizan para desarrollar un código de fuente o programa
* El ciclo d desarrollo de software, en particular:
  + Seguridad en la metodología de desarrollo de software.
  + Pautas de codificación segura para cada lenguaje de programación que se utiliza.
* El establecimiento de puntos de verificación de seguridad dentro de los hitos de los proyectos de desarrollo de software y sistemas
* Los repositorios de información asociados a los proyectos de desarrollo de software y sistemas
* El manejo del control de versiones de los proyectos de desarrollo de software y sistemas
* La capacidad del equipo de desarrollo para:
  + Conocer las condiciones de seguridad de las aplicaciones desarrolladas
  + Evitar, encontrar y resolver las vulnerabilidades de los desarrollos de software y sistemas
  + Los desarrollos ejecutados por personal externo y las condiciones contractuales con las empresas prestadoras de este servicio.
    1. Separación de entornos de desarrollo, prueba y operacionales

(Control A.12.1.4 NCh ISO 27001:2013)

Los ambientes de desarrollo, prueba y producción deben estar separados físicamente, de tal forma que se evite la contaminación de los ambientes de producción con código malicioso o con código que no ha sido probado.

Para el caso de los desarrollos externalizados, se definirán y se documentarán en los respectivos documentos las reglas y pasos para la transferencia del software codificado desde el ambiente de desarrollo al ambiente de prueba y desde el ambiente de prueba al ambiente de producción. Estas reglas deben considerar al menos los siguientes aspectos:

* Ambiente de desarrollo (Proveedor)
* Ambiente de prueba (Proveedor)
* Ambiente de preproducción (Tu auto)
* Ambiente de producción (Tu auto)
* Al final del proyecto, se debe hacer el traspaso del repositorio del proyecto A “Tu auto”
  + 1. Inventario de activos

(Control A.8.1.1 NCh ISO 27001:2013)

Se debe mantener un inventario de activos que contenga todos los activos que se usen para la ejecución de aplicaciones, proyectos, y procedimientos para la organización y que sean propiedad de la organización o que estén bajo su control. El inventario de activos debe incluir activos de información de propiedad de terceros que estén en posesión de la organización. El inventario deberá mantener un registro obligatorio de todos los objetos de valor de la organización, sobre todo aquellos que contengan información sensible de pago.

* + 1. Devolución de activos

Control A.8.1.3 NCh ISO 27001:2013)

En caso de que un activo sea devuelto a al inventario, se debe asegurar que la información contenida en el activo se gestione de la manera adecuada. Si necesita ser borrada, se debe hacer de forma segura. Si se necesita transferir a otro propietario, se debe hacer de forma segura. Si se necesita almacenar, se debe hacer de forma segura. del mismo modo, el estado del activo debe ser actualizado en el inventario.

* + 1. Accesos a la red y servicios de la red

(Control A.13.1.1 NCh ISO 27001:2013)

Los usuarios deben tener acceso directo a la red que se les asigna, y no a través de otras redes. De la misma forma, la red debe ser lo suficientemente segura para evitar que los usuarios no autorizados accedan a ella y la información pueda ser comprometida, robada o alterada.

* + 1. Trabajos del evaluador PCI DSS

Guía Rápida de Referencia de PCI DSS v4.0 - Los Profesionales de Asistencia con las Evaluaciones de PCI DSS

El evaluador debe cumplir con la tarea de generar una serie de informes, ya que estas son el medio por el cual se transmite el estatus del cumplimiento PCI DSS. Los informes que deberán entregar son los siguientes:

* Informe de cumplimiento
  + Informe detallado para que los asesores documenten los resultados de una evaluación de PCI DSS. EL ROC contiene información más detallada que los Cuestionarios de Autoevaluación, incluida la información sobre el entorno de la entidad, las muestras que el asesor ha seleccionado y cómo se evaluó y validó cada requisito. La Plantilla de ROC proporciona instrucciones de informes detallados para los asesores y es obligatorio para que utilicen los QSA en cualquier evaluación de PCI DSS que se documenta en el ROC.
* Cuestionario de autoevaluación
  + Los Cuestionarios de Autoevaluación (SAQ) proporcionan herramientas de validación alternativas para entidades que, de acuerdo con los programas de cumplimiento de las marcas de pago, son aptas para realizar autoevaluaciones que validen su cumplimiento de PCI DSS y que cumplen con el Criterio de Elegibilidad de SAQ especificado en cada SAQ. Distintos SAQ se encuentran disponibles para los diferentes entornos de los comerciantes, incluidos los entornos de comercio electrónico y el entorno de las soluciones de Cifrado de Punto a Punto (P2PE) de la lista de PCI. La mayoría de los SAQ incluyen un subgrupo de aquellos requisitos de PCI DSS que solo son aplicables a un entorno particular. Pueden encontrarse más detalles en el documento de Instrucciones y Directrices de SAQ y en la sección SAQ del sitio web de PCI SSC. Para determinar si usted es elegible para completar un SAQ y, en caso positivo, cuál sería el SAQ adecuado, contacte con las marcas de pago o su banco adquirente.
* Atestaciones de cumplimiento
  + Una Atestación de Cumplimiento (AOC) es una declaración de los resultados de una evaluación de PCI DSS, completada y firmada por la entidad que fue evaluada y el QSA de la compañía (si hubo). La AOC refleja los resultados de la evaluación de PCI DSS que se documentan en un ROC vinculado o SAQ.
    1. Proteger datos de las tarjetas existentes con criptografía

Para evitar que sean vulnerados, los Números de Cuenta Principal (PAN) deben estar cifrados durante la transmisión a través de redes a las que las personas maliciosas puedan acceder fácilmente, incluidas las redes públicas y no confiables. Las redes inalámbricas mal configuradas y las vulnerabilidades de los cifrados tradicionales y los protocolos de autenticación continúan siendo el objetivo de personas maliciosas que buscan aprovechar estas vulnerabilidades para obtener un acceso privilegiado a los entornos de los datos del Tarjetas habientes (CDE). Las transmisiones de los PAN pueden ser protegidas, cifrando los datos antes de que estos sean transmitidos o cifrando la sesión a través de la cual se transmiten los datos, o ambos.

* Los procesos y mecanismos para proteger los datos de los titulares de tarjetas con criptografía sólida durante la transmisión de redes públicas abiertas están definidos y documentados
* Los PAN están protegidos con criptografía sólida durante la transmisión
  + 1. Desarrollar y mantener sistemas y software seguros

Las vulnerabilidades de seguridad de los sistemas y de las aplicaciones pueden permitir a los criminales tener acceso a los datos de pago. Muchas de estas vulnerabilidades se eliminan con la instalación de parches de seguridad proporcionados por el proveedor, que realizan un trabajo de reparación rápida para una parte específica del código de programación. Todos los componentes del sistema deben contar con la instalación de los más recientes parches de seguridad para prevenir el aprovechamiento. Las entidades también deberán aplicar los parches a los sistemas menos críticos en un marco de tiempo correspondiente, en base a un análisis de riesgo formal. Las aplicaciones deben desarrollarse de acuerdo con un desarrollo seguro y a las prácticas de codificación, y los cambios a los sistemas en el entorno de los datos del tarjeta habientes deben cumplir con los procedimientos de control de cambios.

* Se definen y comprenden los procesos y mecanismos para desarrollar y mantener sistemas y software seguro
* El software a medida y personalziado se desarrolla de forma segura
* Las vulnerabilidades de seguridad se identifican y son abordadas.
* Las aplicaciones web públicas están protegidas contra ataques
* Los cambios en todos los componentes del sistema se gestionan de forma segura
  1. Periodicidad de evaluación y revisión

La presente política debe ser evaluada cada dos años como máximo y sus cambios deben ser aprobados por el secretario ejecutivo.

Su cumplimiento se debe revisar en forma anual, en reunión de Comité de Seguridad, con la finalidad de asegurar su cumplimiento e incorporación de todas las normas y procedimientos necesarios de implementar en el marco de Ciberseguridad y Seguridad de la Información.

* 1. Difusión

La presente Política entrará en vigor una vez aprobada por el secretario ejecutivo y será difundida mediante correo electrónico a todos los funcionarios de “Tu auto”.

1. Gestión de vulnerabilidades

La gestión de vulnerabilidades es un componente esencial en el panorama de la ciberseguridad, destinado a identificar, evaluar y abordar las debilidades potenciales en sistemas, redes y aplicaciones. En este contexto, se implementan estrategias proactivas para detectar vulnerabilidades, evaluando su impacto y riesgo asociado. La finalidad última de esta práctica es fortalecer la postura de seguridad de una organización, mitigar amenazas y garantizar la integridad, confidencialidad y disponibilidad de los activos de información. A continuación, analizaremos y explotaremos las vulnerabilidades que encontraremos en los servicios que se están ejecutando en el servidor web de “Tu auto”

* 1. Análisis de vulnerabilidades

Como mencionamos anteriormente, primero necesitaremos hacer un análisis de los servicios y procesos que se están ejecutando en la maquina y que están abiertos para que estos servicios puedan tener comunicación por red. Las herramientas que usaremos serán las siguientes:

* Sistema operativo ArchLinux (con repositorios de BlackArch añadidos. Sin embargo, se recomienda instalar KaliLinux)
* Npm para ejecutar un análisis en la red
* Metasploit-framework para poder ejecutar los exploits y las vulnerabilidades

Teniendo listas estas herramientas, deberemos hacer lo siguiente:

* Abrir nuestra terminal de preferencia, en nuestro caso utilizaremos kitty y ejecutar el comando ‘ip addr’ tal como se muestra en la imagen (también podemos utilizar ifconfig en otras distribuciones de Linux) Interfaz de usuario gráfica

  Descripción generada automáticamente con confianza baja
* Ahora que tenemos información de nuestra red, podemos ver el subhost en el que estamos. Asumiendo que estamos en la misma red local que nuestro servidor objetivo, pero sabemos cual es la IP especifica, deberemos ejecutar el siguiente comando: ‘nmap -sV 192.168.100.\*’ (La ip debe ser reemplazada por la que le arrojo a usted cuando ejecuto ip addr)
* Como podemos ver, hemos obtenido una lista de servicios de un servidor que está en nuestro mismo subhost. Este servidor contiene diversos servicios activos, y se encuentra en 192.168.100.194, sabiendo estos datos usaremos la herramienta Teenable Nessus para poder obtener más información acerca de vulnerabilidades en este servidor, sin embargo, ya podemos hacer uso de otros frameworks para poder atacar.

Texto

Descripción generada automáticamente

* Deberemos iniciar sesión en nuestro servicio Nessus, por defecto este servidor se encuentra en localhost:8834 (también puede ser 0.0.0.0:8834, 192.168.12.5:8834) e iniciar sesión con nuestra cuenta gratuita, en caso de no tener una cuenta se debe crear una cuenta en la página oficial de Nessus y seguir el procedimiento indicado. Una vez dentro, deberá crear un escaneo nuevo haciendo click en el botón “New Scan” indicado en la siguiente imagen Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

  Descripción generada automáticamente
* Ahora deberemos hacer click en el botón que dice “Basic network scan” y llenar el formulario tal como está indicado y ejecutar el escaneo. Las instrucciones estarán mostradas en las siguientes imágenes:
* :Interfaz de usuario gráfica, Aplicación

  Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación, Correo electrónico

  Descripción generada automáticamente \* el valor de target debe ser cambiado por la ip que le arrojo al momento de hacer el escaneo

Aplicación

Descripción generada automáticamente con confianza baja

* Una vez haya terminado el escaneo, podremos ver todas las vulnerabilidades encontradas en la pestaña que dice “Vulnerabilities”. En esta pestaña encontraremos todas las vulnerabilidades clasificadas por El nivel de severidad, su puntaje CVSS, su nombre, la familia de la vulnerabilidad, y la cantidad de veces que se ha encontrado. Tabla

  Descripción generada automáticamente
* Si hacemos click en una vulnerabilidad, podemos ver más información acerca de esta, en conjunto de su posible solución Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

  Descripción generada automáticamente
  1. Clasificación de vulnerabilidades según su criticidad

Nuestro cliente, solamente nos ha notificado que su página es la que está siendo vulnerada mediante inyecciones de Código. Sin embargo, mediante el proceso ejecutado con Nessus, nos hemos percatado que su máquina está haciendo una función muy parecida a la de un Honeypot, pero con la diferencia de que el sistema estaba en una producción real. Por lo que se puede intuir de que la mejor opción para nuestro cliente es tener que reinstalar todo el sistema. Idealmente tanto en hardware como software. A continuación, se presentará una imagen con todas las vulnerabilidades encontradas por Nessus. Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Aplicación, Tabla

Descripción generada automáticamente Tabla

Descripción generada automáticamente Tabla

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

En total, se han encontrado 73 Vulnerabilidades en el sistema, siendo el 70% avisos de configuración insegura, un 7% de vulnerabilidades críticas, 4% en altas, 13% en medias y un 4% en bajas.

* 1. Explotación de vulnerabilidad

Si bien, anteriormente dimos una posible solución que arreglaría todas las vulnerabilidades presentes de un solo tiro, nuestro verdadero propósito es explicar que es lo que está pasando en la página, ya que con esa finalidad se ha contratado Corporacion Security LTDA, por lo que a continuación se explicara que es realmente lo que sucede + otras dos vulnerabilidades bastante peligrosas que están presente en el servidor.

* Inyección SQL a página web
  + Para explotar la vulnerabilidad por la que nuestro cliente, empresas “Tu auto” nos ha contactado, deberemos ingresar al host analizado (en el caso de nosotros es 192.168.100.194/dvwa) e ingresar con las credenciales de ‘admin’ y ‘password’.
    - Pequeño hincapié, podemos notar que la contraseña usada para la cuenta de administración es bastante fácil de explotar, por lo que la primera recomendación es cambiar la contraseña por una más robusta

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

\*nota: Este ataque se puede ejecutar desde un PC común y corriente, sin la necesidad de un sistema operativo o herramientas especificas más allá de un navegador, por lo que por comodidad este ataque se rendirá desde una maquina con Windows

* + Una vez hayamos iniciado sesión, nos dirigiremos a este enlace <http://192.168.100.194/dvwa/vulnerabilities/sqli/>.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* + Se nos ha notificado que este es uno de los campos vulnerables a inyecciones, por lo que nuestra misión es el descubrir cómo. Como bien pudimos notar con anterioridad en nuestro escaneo de puertos, uno de los servicios que estaban ejecutándose era un servidor de MySQL, por lo que podremos probar ejecutando una inyección SQL enfocada a este motor, sin embargo, es posible ejecutar otras enfocadas a motores como SQL Server, Oracle SQL, Postgres, Sql lite, etc. El codigo utilizado es el siguiente:

1' or 1=1 union select 1, table\_name from information\_schema.tables #’ Imagen que contiene Patrón de fondo

Descripción generada automáticamenteImagen que contiene Texto

Descripción generada automáticamente

* + Se puede apreciar que existen varias columnas de sistema, y nosotros estamos buscando columnas que pueden tener información privilegiada y sensible, por lo que usaremos la tabla de usuarios como referencia para obtener datos haciendo la siguiente inyección:

1' or 1=1 union select 1, column\_name from information\_schema.columns where table\_name='users' #

Imagen que contiene Tabla

Descripción generada automáticamente

* + Como podemos ver, existen 6 columnas en la tabla de usuarios, y por la forma en la que estamos ejecutando la unión en la sentencia, adicionalmente nos está entregando información adicional con los usuarios de admin, gordon Brown, hack me, pablo Picasso y bob Smith. Nuestra misión ahora será armar la sentencia que presente todos los datos de todos los usuarios disponibles en la base de datos. Por lo que ejecutaremos la siguiente inyección:

%' AND 1=1 UNION SELECT NULL, CONCAT(USER\_ID,0x002E,FIRST\_NAME,0x0020,LAST\_NAME,0x007C,USER,0x007C,PASSWORD,0x007C,AVATAR) FROM users #Imagen que contiene Texto

Descripción generada automáticamente

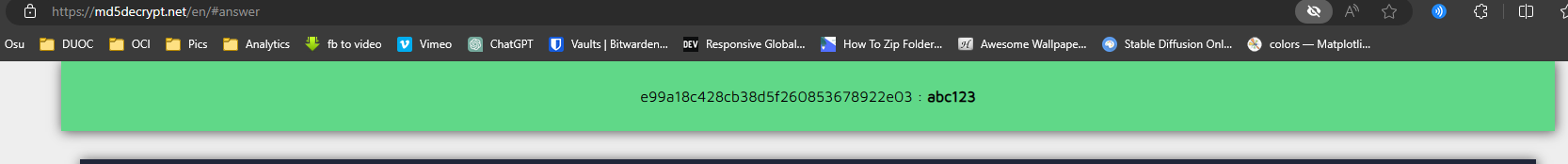
Los siguientes valores hexadecimales equivalen a lo siguiente:

* 0x002E = .
* 0x0020 = (espacio)
* 0x007C = |

Usamos valores hexadecimales para representar cadenas de texto, debido a que al estar ejecutando una inyección SQL, tenemos restricciones para escribir comillas, ya que rompería la lógica y sintaxis de la consulta a la base de datos, por lo que requerimos hacer uso de métodos mas arcaicos de computación, escribiendo la dirección que existe en la memoria para poder digitar ese signo.

* + Hemos logrado obtener todos los datos de la tabla de usuarios con éxito. Adicionalmente nos dimos cuenta quelas contraseñas usan un Hasheo MD5, el cual ya esta bastante obsoleto y las maquinas actuales pueden descubrir la firma de ese hash con bastante facilidad. Incluso sin usar programas, solamente con servicios de internet podemos obtener el valor de las contraseñas. Por ejemplo, tomando el hash de la contraseña de Gordon Brown podemos notar que su contraseña es abc123, tal como se indica en las siguientes imágenes: Interfaz de usuario gráfica

    Descripción generada automáticamente con confianza bajaCaptura de pantalla de un celular

    Descripción generada automáticamente
  + Si ingresamos con las credenciales de Gordon Brown también podemos ingresar al sistema

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Servidor IRC CVE-2010-2075
  + El servidor de nuestro cliente, esta usando un servicio llamado Unreal. Unreal es uno de los servidores IRC mas grandes dentro de su familia de servicios. Sin embargo, la versión que está ejecutándose es una versión que sufrió de una vulnerabilidad inyectada a propósito, en la que no se sabe quien fue el debido autor, por lo que el servicio estuvo mucho tiempo vulnerable. Efectivamente, este servidor posee esa vulnerabilidad, por lo que ahora deberemos explotarla. Primero que todo identificaremos el puerto en el cual está ejecutándose nuestro Servidor IRC

Texto

Descripción generada automáticamente

* + Podemos notar que se encuentra en el puerto 6667 bajo el protocolo TCP. Por lo que mediante metasploit-framework nos aprovecharemos de esto. Ejecutaremos el comando “msf-console” para abrir Imagen que contiene Mapa

    Descripción generada automáticamente
  + Una vez en este panel de comandos, deberemos ejecutar el comando de búsqueda para seleccionar el exploit del servidor IRC que queremos vulnerar. Escribiremos “search unreal” y seleccionaremos el exploit que nos sienta más cómodo con “use (id del exploit)”. En nuestro caso por comodidad seleccionaremos el exploit con el id numero 2Texto

    Descripción generada automáticamente
  + Con el comando “show info”, podremos ver todos los parámetros disponibles para la ejecución de este exploit Texto

    Descripción generada automáticamente
  + Podemos apreciar que este exploit requiere de un parámetro llamado RHOSTS (al igual que la mayoría de exploit en este framework). Deberemos llenarlo con la IP de nuestra víctima, por lo que el comando a ejecutar seria el siguiente: ”set RHOSTS 192.168.100.194”. Adicionalmente, en este caso también deberemos llenar el parámetro de LHOST, que significa que deberemos indicarle de donde viene la conexión (desde nosotros), pero en caso de que tengamos otra maquina como cebo, podemos usar la dirección de esa máquina. El comando para ejecutar es “set LHOST (ip de host)”
  + Gran parte de los exploits de metasploitable vienen con un payload preseleccionado, sin embargo, este exploit seleccionado no viene con uno. Por lo que debemos manualmente nosotros elegir un payload de la misma manera que buscamos un exploit. Primero debemos ejecutar “show payloads” y se nos desplegara una lista con los payloads compatibles con el exploit seleccionado. Seguido de eso deberemos elegir un payload a usar con “set PAYLOAD (id del payload a elegir)”, en nuestro caso usaremos el número 5 de la lista que nos proporcionó metasploit-framework Texto

    Descripción generada automáticamente
  + Cuando el exploit imprima el texto “Command Shell sesión ….” Quiere decir que hemos explotado la vulnerabilidad con éxito, y podemos comprobarlo ejecutando los siguientes comandos:
    - Whoami : para saber el usuario actual de la sesión. Si retorna root, quiere decir que entramos con los máximos permisos posibles
    - Pwd: retorna el directorio en el que nos encontramos
    - Ip addr o ifconfig: retorna la información de red de nuestra máquina. Podemos comprobar esto si nos retorna el ip local de nuestra victima
    - Ls: Ver los archivos y ficheros que se encuentran en la máquina de la victima

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* + Podemos notar que todos los comandos ejecutados aluden a información de nuestro objetivo. Incluso en la línea destacada el comando ip addr nos retorno la información de red que no es de nosotros, del mismo modo la dirección ip retornada es la que establecimos en el parámetro RHOSTS
  + Para poder salir de esta sesión podemos ejecutar el comando exit o la combinación de teclado CTRL+C
* Vulnerabilidad vsFTPd 2.3.4
  + El protocolo FTP es un protocolo de los años 70. Sus siglas significan “File transfer protocol” y trabaja bajo el protocolo TCP en el puerto 21. Interfaz de usuario gráfica, Texto

    Descripción generada automáticamente
  + Una vez obtenida esta información, ejecutaremos el mismo procedimiento que antes. Buscaremos el exploit y luego elegiremos el que mas nos convenga. En este caso nos da dos opciones, un exploit para denegación de servicios y otro para tomar control de la maquina mediante un backdoor. Por motivos prácticos, elegiremos la segunda opción en el id 1 con el comando “use 1”Pantalla de computadora con letras

    Descripción generada automáticamente con confianza media
  + Ahora, con el comando “show options” se nos desplegara la misma lista que en el exploit anterior, pero con parámetros enfocados a otro exploit. En este caso solamente nos pide llenar el parámetro de RHOSTS y de RPORT, siendo este ultimo con el valor por defecto. Estableceremos el parámetro RHOSTS tal como se muestra en la siguiente foto. Texto

    Descripción generada automáticamente
  + Ahora, si ejecutamos el exploit nos pedirá una contraseña. El procedimiento de este exploit consta de lo siguiente, cuando pida la contraseña no haremos absolutamente nada y dejaremos que de error por timeout, ya una vez de el error, ejecutaremos el exploit por segunda vez y el servicio nos dará pase automáticamente sin autentificar. Texto

    Descripción generada automáticamente
  + Cuando salga el mensaje “Command Shell sesión (id) opened ……..” quiere decir que estamos dentro del servidor con los máximos permisos, podemos validarlo con las instrucciones que probamos en el exploit anterior, tal como se indica en la siguiente imagen Texto

    Descripción generada automáticamente
  + Tal como antes, podremos ejecutar el comando exit o ejecutando la combinación CTRL+C para salir.
  1. Definir posibles soluciones

Considerando la cantidad de vulnerabilidades que fueron encontradas por Nessus, y solamente considerando que nosotros abarcamos las 3 más críticas, las soluciones más optimas son las siguientes:

* Cambiar hardware
  + Es posible que el servidor ya tenga un programa maligno instalado, y muchas veces puede ser mucho mas caro hacer un análisis forense que comprar equipamiento completamente nuevo y armarlo uno mismo en caso de que no se tenga un departamento de seguridad.
* [OBLIGATORIO] Cambiar sistema operativo por uno más nuevo
  + El sistema operativo que esta utilizando el servidor es muy antiguo y ya no recibe soporte, por lo que no es una opción segura mantener servicios en un servidor con tales versiones tan obsoletas de software. Lo ideal es reinstalar uno mas moderno, como por ejemplo Ubuntu Server en su versión LTS, Fedora WorkStation, Open Suse, Redhat, etc.
* Actualizar periódicamente las actualizaciones de seguridad tanto de software como de los servicios a utilizar
* Cerrar puertos y servicios que no estén en uso
  + Hemos notado que el servidor contiene muchos puertos y servicios innecesarios abiertos, considerando que solamente debería ser un servidor web, como mucho deberá contener 3 puertos y servicios abierto, puerto apache http server o nginx, SSH para permitir conexiones remotas, y mysql en caso de que se tenga la base de datos en el mismo servidor (No recomendado)
* Usar un firewall por servidor
* Distribuir los procesos de negocios y aplicaciones de la compañía “Tu auto” en un servidor independiente.
  + Por ejemplo, La página web debería ser un servidor, la base de datos para la página web debería ser otro, los programas utilizados para gestionar los inventarios deberían tener otro servidor, etc.
  + Esto se hace para que en caso de que se deshabilite un servidor, tenga el menor impacto posible en el resto de los sistemas
* Priorizar mantener todo el servidor en una subnet, y de eso que se conecte mediante un proxy o un tunnel hacia conexiones exteriores
* Los servidores SSH deben admitir solamente conexiones autorizadas y limitadas. Por ejemplo, si a un servidor se conectaran solamente 2 tipos de usuarios que serian el administrador y los desarrolladores se debería permitir las IPs de las personas involucradas
* Aplicar políticas de cambio de contraseñas a todos los usuarios que existan en el servido, las políticas deben considerar lo siguiente:
  + Cambio de contraseña semestral
  + Las contraseñas deben ser alfanuméricas por obligación, contener al menos 1 signo, y tener como mínimo una cantidad de 13 caracteres
* Aplicar hardening con el fin de evitar, molestar al atacante del servidor
* Usar sistemas de prevención y detección de intrusos
* Cambios en las políticas de documentos para evitar filtraciones de contraseñas
  + Usar contraseñas seguras, como las indicadas en el punto superior
  + Todas las contraseñas almacenadas deberán ser HASHEADAS con el algoritmo ARGON2 (Recomendado por la OWASP) o PBKDF2(Recomendado por NIST).
  + Todos los documentos que contengan información sensible a desechar

deben ser triturados por una trituradora de papel que tenga como mínimo un estándar de nivel 6, una vez el papel haya sido triturado se deberá quemar y las cenizas deberán ser re-destruidas según que tipo de información tengan, si el documento tiene un nivel de clasificación muy alta, lo ideal será desechar el papel a nivel químico. Caso contrario se puede aplicar agua o directamente ser botado en un traste común de oficina.

* Utilizar prácticas de Código seguro y limpio al momento de desarrollar
  + Los desarrolladores deben seguir los estándares instaurados por OWASASP
  + Los desarrolladores deberán estar certificados para poder desarrollar aplicaciones seguras (Idealmente la empresa deberá dar acceso a los certificados)
  + Evitar el uso de querys directas en el codigo, parametrizar todo antes de unir todo en la sentencia SQL para luego ejecutarla.
  + Añadir validadores tanto en el frontend como en el backend de inyecciones
  + Controlar las consultas mediante vistas y permisos de usuarios en el motor de SQL
  + Si es posible, usar modelos para controlar la base de datos. Ejemplos de estos pueden ser Prisma con Javascript.
  + Utilizar certificados SSL de forma obligatoria para establecer conexiones seguras entre cliente y servidor. Tanto en páginas webs, apis, servidores Backend y bases de datos

1. Plan de mitigación
   1. Identificación, análisis y priorización de procesos y servicios críticos TI que forman parte del alcance

Todos los procesos de negocios que tengan involucrados los siguientes contextos:

* Compradores
* Empleados
* Oficinas
* Órdenes
* Pagos
* Productos (Automóviles y refacciones)

Los principales procesos de negocios afectados por una caída o desaparición de la base de datos son:

**Reconocimiento de empleados en las cajas de venta “Tu auto”**: En caso de un desastre que afecte la base de datos, los empleados enfrentarán dificultades para ser identificados, lo que obstaculizara el proceso de ventas al no poder ingresar con sus credenciales, generando problemas operativos en las cajas y afectando la eficiencia del servicio al cliente.

**Reconocimiento de información de productos**: La falta de acceso a la base de datos hará imposible el reconocimiento de productos, ya que los códigos de barras no podrán cumplir su función. Esto podría resultar en interrupciones operativas, afectando la identificación precisa de productos, la facturación y la experiencia del cliente.

**Formas de Pago**: La base de datos almacena información crítica sobre las diversas formas de pago aceptadas en “TU AUTO”. Cualquier irregularidad o fallo en este sistema podría dar lugar a problemas significativos en los procesos de pago de servicios. La falta de acceso a la información pertinente podría resultar en dificultades para procesar transacciones, afectando la eficacia en el servicio y generando inconvenientes en la gestión financiera de la organización.

**Errores de Reabastecimiento locales “Tu Auto”:** La base de datos alberga información crucial sobre las necesidades de reabastecimiento de productos en las oficinas o establecimientos “TU AUTO”. Cualquier inconveniente con esta información podría desencadenar problemas significativos, potencialmente llevando a una compleja situación de desabastecimiento en los locales ”Tu auto”.

* + 1. Procesos de negocio críticos y los servicios – aplicaciones TI que lo soportan

**Proceso de Ventas en Cajas “TU AUTO”:**

Servicio de Reconocimiento de Empleados en Cajas (Sistema de Identificación de Empleados).

**Gestión de Inventario y Reabastecimiento:**

Sistema de Control de Inventario.

Aplicación de Registro de Necesidades de Reabastecimiento.

**Proceso de Pago de Servicios:**

Servicio de Formas de Pago “Tu auto” (Sistema de Gestión de Pagos).

**Identificación y Registro de Productos:**

Sistema de Reconocimiento de Información de Productos (Base de Datos de Productos y Códigos de Barras).

* + 1. Identificación de dependencias de los servicios TI

Las máquinas donde estarán alojados nuestros servidores web, backend, base de datos deberán tener las siguientes características tanto de hardware como dependencias de software

* Hardware
  + CPU: x2 AMD EPYC™ 9754
  + RAM: x24 128 GB RAM DDR5 5.5@GHz
  + Almacenamiento: Crucial T700 4 TB M.2 + (Raid 5) x5 Lenovo 4 TB (7XB7A00051)) (16 TB Utilizables)
  + Fuente 900W Redundante HP 745813-B21
  + x5 Cables interfaz SATA 3
  + Placa Madre Gigabyte MZ73-LM1
  + Teclado de oficina (opcional)
  + Monitor de oficina (opcional)
* Software
  + Sistema operativo Debian 12 minified
  + Bash
  + OpenSSH
  + MySql (MariaDB)
  + MySql-dump
  + Curl y/o Wget
  + Vim y/o Nano
  + tmux
  + htop
  + (nvm node version manager) NodeJs + npm
  + Python3 + pip
  + Certificado SSL para habilitar HTTPS
    1. Documentación de arquitectura de sistemas y dependencias

Diagrama

Descripción generada automáticamente

Todos los servidores se encontrarán en la misma subred a excepción del servidor de respaldo, este se encontrará físicamente en un lugar distinto para asegurar la integridad de los respaldos. Los equipos de desarrollo se encontrarán en una oficina externa a donde se encuentran los servidores, por lo que el sistema se estará ejecutando en un datacenter especializado en el mantenimiento de estos.

* + 1. Tiempos y puntos de recuperación requeridos y factibles para sistemas y dependencias
* **RPO**

El objetivo de este punto es documentar y ejecutar la recuperación de la base de datos MySql en un servidor Linux Debian. Primero lo que debemos hacer es revisar la integridad del servidor, ya que dependiendo del desastre variará ligeramente la recuperación a nivel de hardware.

Si el servidor fue víctima de un hackeo, solamente deberemos cambiar nuestra placa madre y disco duro, adicionalmente el dispositivo de almacenamiento recibirá un tratamiento de formateo profundo. Una vez el formateo se haya completado, el dispositivo será depositado a un almacén de hardware para donaciones. Nuestra placa madre recibirá un tratamiento de reprogramación de UEFI y cambio de pila de UEFI para también ser depositada al almacén de hardware para donaciones.

En caso de que el desastre sea causado por un incendio o lluvias extremas o un exceso de humedad el procedimiento consistirá en cambiar todo el hardware para evitar posibles fallas en producción, en el que los componentes antiguos serán llevados a su respectiva revisión técnica, en el que, si los componentes pasan las pruebas especificadas en la revisión, serán depositados para el almacén de hardware para donaciones

En caso de que el hardware sea causado por un terremoto, se deberán chequear todos los dispositivos de almacenamiento en revisión técnica, verificando que su funcionamiento sea el adecuado, luego se chequeará el resto del hardware. En caso de que el hardware no pase las pruebas, se aplicará el mismo procedimiento que en casos anteriores para que el hardware que se haya podido arreglar, se pueda depositar

Una vez se hayan chequeados todos los posibles desastres y si se requiere, procederemos a la restauración del sistema de las máquinas, en el que instalaremos Debian minimal, MySql + OpenSSH + tmux y luego los programas necesarios por servidor.

* **RTO**

La recuperación tendrá una variación de tiempo según el tipo de desastre, ya dependiendo del desastre, se podrán hacer ciertas tareas en paralelo, Sin embargo, estimando tiempos:

* 60 minutos- Ensamblar servidor
* 30 minutos - Instalar Debian
* 10 minutos- Instalar actualizaciones
* 1 minuto - Ejecutar script de recuperación de programas por servidor
* 60 minutos - Pruebas post recuperación

TOTAL = 2 horas y 41 minutos como máximo (161 minutos) [2.68 horas] estimados para volver a operar con normalidad

* 1. Estrategias de recuperación

Principalmente, se aplica el mismo filtro de desastres por lo que el procedimiento de instalación de hardware y sistema operativo siempre será el mismo para todos los servidores. Sin embargo, al momento de instalar el software varía el procedimiento, tanto como para servidores web, de base de datos y de apis.

Principalmente nos enfocaremos en la estrategia de recuperación de base de datos y la página web. Periódicamente se ejecutará una tarea en cronjob que se encargará de subir un script que tendrá todos los comandos ejecutables + otros scripts para poder recuperar nuestros sistemas. Este script se ejecutará 2 veces al día cada 12 horas, la primera vez a las 02:00 hrs y la segunda vez a las 14:00 hrs.

Cuando llegue el momento de restaurar los sistemas ante un desastre, instalaremos nuestro sistema operativo y lo primero que haremos será actualizarlo, seguido de eso instalaremos tmux, wget, curl, vim, y nano, una vez instalados los programas mencionados, ejecutaremos el comando wget seguido del archivo con el script para restaurar nuestra base de datos. Todo esto al ser de manera automatizada, nos permitirá volver a nuestro funcionamiento normal de una manera rápida y eficaz

* 1. Procedimiento de alerta, escalamiento, activación y gobierno de la recuperación TI

Apenas se inicie el estado de alerta, el gobierno sufrirá una mutación en la que el equipo del jefe de sistemas tomara el mando por sobre el subgerente.

El estado se accionará mediante un sistema de prevención y detección de intrusos informáticos o ante una obvia situación de alerta, esta alerta puede gatillarse por una causa natural como terremoto, lluvias extremas, frio extremo, etc o así también como por desastres provocados por accidentes, como por ejemplo incendios, explosiones, etc.

Si se activa por un incendio, lluvias extremas y humedad o terremoto se activará la alerta de desastre respectiva, en estos casos las escalas dentro del gobierno sufrirán un cambio en el que el Gerente general será el mandatario y el Jefe de sistemas el segundo al mando ambos puestos Diagrama

Descripción generada automáticamente

* + 1. Diagramas de flujo

Diagrama

Descripción generada automáticamente

\*Nota: Diagrama no actualizado. En vez de Servidor de base de datos esta en riesgo debería decir “Algún proceso de negocio está en riesgo”

* + 1. Descripción de equipos y roles

**Equipo de Seguridad**: [Nosotros, Corporación Security LTDA] Este equipo está encargado de salvaguardar la seguridad e integridad de servidores y procesos de negocios, así como de implementar medidas para proteger la información sensible. Sus roles incluyen la supervisión de accesos, la gestión de sistemas de vigilancia, la coordinación con autoridades locales en caso de cambios por posibles amenazas y vulnerabilidades.

**Equipo de Mantenimiento**: Este equipo se encarga de preservar la operatividad de los servidores físicos, distribuidos entre el equipo interno de empresas “Tu auto” y el externo que administra los servidores de nuestras instalaciones. Ambos equipos colaboran de manera coordinada para garantizar un óptimo funcionamiento.

**Equipo de desarrollo:** Este equipo esta encargado de desarrollar los servicios y el software ejecutado por empresas “Tu auto”, tendrán la misión de mantener un Código seguro, y estar al tanto con las normativas y estándares de OWASP y las ISO’s 27001 y 27002.

* + 1. Descripción resumida del flujo de escalamiento

El proceso de escalonamiento se activará al entrar en el estado de alerta, dando lugar a una redistribución de roles entre el comité de seguridad de la información y el equipo directivo. Esto resultará en una disminución de la influencia y autoridad del equipo directivo en las decisiones empresariales, durante el periodo en que se mantenga el estado de alerta por desastre. Este ajuste temporal se revertirá una vez que empresas “Tu auto” restablezca sus operaciones normales.

* 1. Procedimientos de contingencia y recuperación
     1. Objetivo

El objetivo de este punto es documentar y ejecutar la recuperación los servidores en un ambiente Linux Debian siguiendo los indicados en el diagrama del punto 5.3.1

* + 1. Alcance

En caso de que uno de los servidores haya sido víctima de un hackeo, solamente deberemos cambiar nuestra placa madre y disco duro de la maquina afectada, adicionalmente el dispositivo de almacenamiento recibirá un tratamiento de formateo profundo. Una vez el formateo se haya completado, el dispositivo será depositado a un almacén de hardware para donaciones. Nuestra placa madre recibirá un tratamiento de reprogramación de UEFI y cambio de pila de UEFI para también ser depositada al almacén de hardware para donaciones.

En caso de que el desastre sea causado por un incendio, lluvias extremas o un exceso de humedad el procedimiento consistirá en cambiar todo el hardware del servidor afectado para evitar posibles fallas en producción, en el que los componentes antiguos serán llevados a su respectiva revisión técnica, en el que, si los componentes pasan las pruebas especificadas en la revisión, serán depositados para el almacén de hardware para donaciones

En caso de que el desastre sea causado por un terremoto, se deberán chequear todos los dispositivos de almacenamiento en revisión técnica, verificando que su funcionamiento sea el adecuado, luego se chequeará el resto del hardware. En caso de que el hardware no pase las pruebas, se aplicará el mismo procedimiento que en casos anteriores para que el hardware que se haya podido arreglar, se pueda depositar

* + - 1. Procedimientos relacionados

Debido a que la situación de desastre es considerada una situación de suma urgencia, apenas se entre en este estado lo primero que se cambiarán serán las jerarquías de gobierno entre el equipo de comité de seguridad y el equipo ejecutivo sin necesidad de un documento de por medio. Una vez el equipo ejecutivo termine su trabajo, deberán volver a sus posiciones normales ambas entidades, y ambas entidades deberán generar un reporte de que es lo que se ha hecho durante y post desastre, adicionalmente el equipo ejecutivo deberá generar un reporte de que es lo que se estuvo haciendo pre desastre en todos sus departamentos que tengan a su disposición, esto con la finalidad de poder hacer un seguimiento a que fue lo que paso en caso de que haya sido un desastre por ataque informático.

* 1. Desarrollo del procedimiento

Diagrama

Descripción generada automáticamente

* + 1. Fase de transición a operación en contingencia (failover)

El estado de failover dará luz verde por el verde si es que es dado desde el por alguna entidad comoel Gerente General o el Subgerente, Este último deberá obtener un respaldo de alguna de las entidades antes mencionadas para comenzar el failover. Sin embargo, si existe un evidente ataque informático y ninguna entidad de las tres primeras entidades se manifiesta en 1 hora el comité de seguridad tendrá total facultad de tener control por sobre el equipo ejecutivo. Si es que algún integrante del comité de seguridad llegase a forzar esta situación de manera malintencionada o de forma que perjudique a “Tu auto”, el comité de ética por parte de Corporación Security LTDA contará con las herramientas para poder controlar esta situación, contando con un equipo externo especializado para investigación y análisis forense.

* + 1. Fase de operación en contingencia

Una vez llegado este procedimiento, se deberá determinar por qué comenzó el desastre, según el tipo de desastres se deberá actuar de manera diferente según lo indicado en el diagrama de flujo y en lo indicado en el RPO y el Alcance.

* + 1. Fase de transición a operación normal o vuelta atrás (failback)

Una vez se haya logrado identificar el problema, tomar las acciones necesarias y resolver a primera vista, el comité de seguridad en conjunto con el equipo desarrollador y el DBA deberán hacer un testeo de por lo menos 1 hora en los sistemas para asegurarse de que todo está funcionando con normalidad, en estos testeos estarán incluidos los cambios de credenciales, las aperturas de puertos específicas en los servidores y el levantamiento de servicios necesarios para el funcionamiento del sistema. Una vez pasadas las todas las pruebas el equipo ejecutivo y todos sus departamentos y empleados deberán reportar que es lo que estuvieron haciendo pre-desastre, durante el desastre y post-desastre, mientras que el comité de seguridad deberá reportar las actividades hechas durante el desastre y acciones a tomar post-desastre.

* 1. Detalle de tareas

5.6.1 Ejemplo de restauración

1. Booteamos el servidor desde un USB con la ISO del sistema operativo descargado, en este caso, debian 12 y seleccionamos Install

Interfaz de usuario gráfica

Descripción generada automáticamente

1. Elegimos el idioma que más nos acomode, en nuestro caso usaremos el inglés

Imagen que contiene Tabla

Descripción generada automáticamente

1. Como ubicación le diremos que vivimos en ‘Other’,‘South america’ y ‘Chile’’

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Luego daremos el formato de nuestro teclado. Si nuestro teclado físico tiene una Ñ lo ideal es que se seleccione el formato español en Latinoamérica, sin embargo, es probable que se tenga un teclado en inglés. En nuestro caso usaremos el último formato mencionado.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

1. Nos preguntará por el nombre del host (o nombre del servidor), podemos dejar este espacio con el valor que nos da por defecto o cambiarlo

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Una vez configurado el host, nos preguntará por la clave de nuestro usuario **root**, por ahora nuestra clave será ‘duoc2023’, luego nos pedirá nuevamente confirmar la contraseña, pero se pide encarecidamente que la clave se cambie apenas se termine de configurar la máquina.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Ahora daremos los datos de usuario para la cuenta para usos no administrativos, el cual será ‘mysql”Tu Auto”’ y luego daremos enter nuevamente para darle los permisos de ejecución de comandos de administrador. Luego de esto, nos pedirá una clave para nuestra cuenta recien creada, la cual pondremos la misma contraseña de nuestro usuario root, la cual nos pedirá que configuremos para asegurarnos que la clave

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Configuramos nuestra zona horaria en Santiago

Interfaz de usuario gráfica

Descripción generada automáticamente

1. Particionamos el disco, usaremos la primera opción (usar el disco completo) y luego seleccionamos nuestro dispositivo de almacenamiento.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Usaremos la opción que nos separa la partición

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Confirmamos los cambios y escribiremos los datos en el dispositivo de almacenamiento

Texto

Descripción generada automáticamente

1. Configuramos nuestro gestor de paquetes para que esté conectados a los servidores de nuestro país, luego seleccionamos un mirror y cuando nos pregunte por un proxy dejaremos el campo en blanco

Imagen que contiene Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. Solamente dejamos marcado ssh server y standard system utilities

Texto

Descripción generada automáticamente con confianza media

1. Instalaremos en nuestra partición primaria

Texto

Descripción generada automáticamente

1. Sacaremos nuestro pendrive de nuestro servidor y reiniciamos

Texto

Descripción generada automáticamente

1. Entraremos a nuestro grub y seleccionamos nuestro sistema operativo recién instalado

Texto

Descripción generada automáticamente

1. Una vez dentro, necesitaremos configurar nuestro archivo sudoers desde nuestra cuenta root, por lo que ingresamos con nuestras credenciales root para actualizar nuestro sistema y luego instalar sudo con el comando ‘apt-get update && apt-get upgrade && apt-get install sudo vim wget curl’

Texto

Descripción generada automáticamente

1. Ahora deberemos configurar el archivo sudoers que se encuentra /etc/sudoers usando el comando ‘vim /etc/sudoers’



1. Se nos abrirá un editor de texto, deberemos editar el archivo de la siguiente manera:

Texto

Descripción generada automáticamente

* + 1. Una vez hayamos añadido nuestro usuario a la lista de sudoers, deberemos apretar ‘:wq!’ para guardar y salir.

1. Usaremos el comando ‘ip addr && systemctl status sshd’ para revisar cual es nuestra ip local y revisar si nuestro servidor ssh está activo

Texto

Descripción generada automáticamente

* + 1. Si todo está en orden, podremos salir de nuestra sesión y conectarnos de forma remota. Si no está nuestro servidor ssh arriba, podemos ejecutar el comando ‘systemctl enable sshd && systemctl start sshd’

1. Desde nuestro computador local, usaremos el comando mysql”Tu Auto”@{ip local dada por ip addr}, escribimos ‘yes’ y nuestra clave para poder iniciar sesión en nuestro servidor de forma remota

Texto

Descripción generada automáticamente

1. ejecutaremos el comando ‘wget https://{ipservidorrespaldo} /Rsnapshot /Respaldos/MySqlbackup.sh && chmod +x MySqlbackup.sh’ para descargar nuestro script para recuperar nuestro backup y respaldar

Texto

Descripción generada automáticamente

1. Ahora ejecutaremos el comando ‘sudo ./MySqlbackup.sh’ seguido de nuestra clave para ejecutar nuestro script. Sabremos que el script comenzará a trabajar cuando aparezcan estos mensajes

Texto

Descripción generada automáticamente

1. El script nos preguntará por las credenciales de nuestro nuevo dba, podemos presionar enter para usar los valores por defecto (no recomendado).

Texto

Descripción generada automáticamente

(La clave por default es 2023duoc)

1. Cuando haya finalizado el script, nos avisará con este mensaje

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

1. Podemos comprobar que nuestro usuario está creado de manera local y remota. Para comprobarlo de manera local, podemos usar el comando ‘mysql -u dba -p’ y digitar nuestra clave

Texto

Descripción generada automáticamente

1. También podemos hacer este mismo procedimiento desde un cliente remoto con mysqlworkbench

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

1. Aprovecharemos de chequear la integridad de los datos desde nuestro cliente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Como podemos ver la base de datos está restaurada con sus respectivas tablas, sin embargo, no sabemos si la integridad de los datos se ha conservado. Podemos ejecutar alguna consulta para probar:

Tabla

Descripción generada automáticamente

1. Pudimos restaurar la base de datos con exitosamente
   1. Anexo
      1. Nómina de personal crítico con datos de contacto

* Alfredo Galdames
  + Líder en ciberseguridad informática DuocUC
  + Correo: [Alfredo.galdames@duocuc.cl](mailto:Alfredo.galdames@duocuc.cl)
  + Reemplazante: Ignacio Sepúlveda
    - Sublider ciberseguridad
    - Nachete@duocuc.cl
* Fernando Muñoz
  + Analista forense en radware
  + Correo: [munozfe@adware.cl](mailto:munozfe@adware.cl)
  + Reemplazante: Daniel Ortega
    - Ingeniero junior en ciberseguridad
    - [Ortegada@adware.cl](mailto:Ortegada@adware.cl)
* Jean Venegas
  + Decano en ciberseguridad de Universidad Católica de Chile
  + Correo: je.venegas@catolica.cl
  + Reemplazante: Paula Velásquez
    - Docente en universidad católica
    - Pau.velazques@catolica.cl
    1. Nómina de proveedores críticos con bases de datos
* DuocUC
  + www.duocuc.cl
  + Avenida equina blanca 501
  + (2) 2560 6900
* Radware Chile
  + [www.radware.com](http://www.radware.com)
  + +56 2 2595 2848
* Universidad Católica de Chile
  + https://educacionprofesional.ing.uc.cl/
  + +56 9 5504 4516 – +56 9 3353 0870
  + Avenida Vicuña Mackenna 4860 Macul, Santiago, Chile

1. Referencias

* Introducción a OWASP - https://owasp.org/www-pdf-archive/Introduccion\_a\_la\_OWASP.pdf
* Norma Chilena ISO 27001:2013 – https://trusttech.cl/docs/nch.27001.pdf
* Norma Chilena ISO 27002:2013 - <https://toaz.info/docdownloadv2-nch-iso-27002-2013-pr_23404e2022f2b5777b9320fe7186ef4c>
* Norma PCI DSS - <https://www.commerce.uwo.ca/pdf/PCI-DSS-v4_0.pdf>
* Debian ISO - <https://www.debian.org/distrib/index.es.html>
* GNU - <https://www.gnu.org/home.es.html>
* Linux - https://distrowatch.com/dwres.php?resource=glossary#linux
* POSIX - <https://distrowatch.com/dwres.php?resource=glossary#posix>
* SHA - <https://csrc.nist.gov/projects/hash-functions#approved-algorithms>
* Rnapshot - <https://rsnapshot.org/> y <http://www.mikerubel.org/computers/rsync_snapshots/>
* CSVV - <https://nvd.nist.gov/vuln-metrics/cvss#:~:text=The%20Common%20Vulnerability%20Scoring%20System,Base%2C%20Temporal%2C%20and%20Environmental>.
* Inyección SQL - <https://www.imperva.com/learn/application-security/sql-injection-sqli/#:~:text=SQL%20injection%2C%20also%20known%20as,lists%20or%20private%20customer%20details>.
* Imagen usada como Portada de documento: https://www.vhv.rs/viewpic/iRmihTT\_transparent-initial-d-png-initial-d-manga-cover/