Auditoria Informática

Alumno: Santiago Vietto

Docente: Alfredo Luis Pardo

DNI: 42654882

Institución: UCC

Año: 2021

La informática como herramienta del auditor

Objetivo de la unidad: entender que es la auditoria a través de los elementos que componen sus concepto, diferenciar lo que es consultoría de auditoría, identificar las funciones de control interno y auditoria informática y poder definir cuáles son sus similitudes y diferencias.

Concepto de Auditoria

_ Cuando hablamos de auditoria hablamos de una actividad que consiste en la emisión de una opinión profesional sobre si un objeto sometido a análisis presenta adecuadamente la realidad que pretende reflejar o cumple ciertas condiciones que le han sido dispuestas. Entonces los elementos que componen este concepto son:

Contenido: que es dar o emitir una opinión.

Condición: esta opinión debe cumplir con la condición de ser profesional.

Justificación: la justificación de esa opinión está sustentada en determinados procedimientos.

Objeto: el objeto bajo estudio o sobre el cual se quiere auditar es una información obtenida de un cierto soporte que puede ser físico o digital.

Finalidad: la finalidad que tiene la auditoria es determinar la fiabilidad.

Momento: el momento en el que se realiza esto es posterior, y esto quiere decir por ejemplo si estamos auditando un sistema operativo, no lo auditamos cuando el sistema operativo se esté instalando sino cuando este en ejecución. Cuando auditamos una aplicación lo mismo se audita una vez que la aplicación está en ejecución para ver si cumple con determinadas condiciones.

Clases de auditoria

_ El objeto el cual sometemos a estudio sea cual sea su soporte y la finalidad con la que se hace ese estudio definen el tipo de auditoría del cual vamos a estar hablando. Como vemos el contenido siempre es una opinión, en las siguientes:

- Auditoria financiera: tenemos que en la auditoria financiera nos interesa ver las cuentas anuales y la finalidad es ver que tan bien presentan la realidad de esas cuentas.
- Auditoria informática: en la auditoria informática auditamos sistemas de aplicación, recursos informáticos, planes de contingencia (plan de continuidad de

negocio por ejemplo), etc. Entonces el objetivo que tenemos es ver si ese plan es eficiente y si se adecua a las normas establecidas para dicho plan.

- Auditoria de gestión: auditamos la dirección de la empresa, con la finalidad de ver que tan eficaz, eficiente y cuál es el rendimiento económico de la empresa que es en definitiva el objetivo que tienen todas las empresas.
- Auditoria de cumplimiento: en esta vamos a auditar cuales son las normas establecidas y que todas las operaciones de una organización, todos sus sistemas, estén alineadas a estas normas. Por ejemplo tenemos una norma referida al uso de las tarjetas de crédito, después tenemos normas que están referidas a la industria de la salud, etc.

Procedimientos

_ Los procedimientos son fundamentalmente para fundamentar y justificar la opinión del auditor, esto es lo que permiten los procedimientos. Entonces cuando vamos a emitir una opinión debemos tener un procedimiento que sea una guía paso a paso y que nos lleve a poder explicar porque llegamos a la conclusión que llegamos.

_ Cada clase o tipo de auditoria tienen sus propios procedimientos y en base a estos procedimientos se define el alcance de la auditoria. Es decir, cuándo vamos a auditar una empresa vemos que es lo critico que tiene esa empresa, cuáles son sus activos, en base a eso definimos los objetivos, que es lo que se quiere controlar sobre esos objetivos y los procedimientos nos van a ayudar a implementar esos controles.

_ Los procedimientos buscan garantizar que:

- Básicamente el trabajo se va a planificar y se va a supervisar adecuadamente.
- Se va a estudiar y evaluar el sistema de control interno. El control interno es cuales son las cosas que se proponen desde el interior de la organización para poder cumplir con las normativas y procedimientos establecidos.
- La evidencia que se va a obtener sea suficiente (cantidad necesaria para justificar esa opinión) y adecuada (cumple con los requisitos de calidad para esa justificación).

Riesgos de los procedimientos: por ahí tenemos la posibilidad de que con los métodos tradicionales que tenemos para auditar no sea posible verificar la totalidad de operaciones existiendo el riesgo de que existan algunas irregularidades que hayan escapado de la atención del auditor. Entonces el auditor debe mantener estos tipos de riesgos dentro de limites tolerables. Este riesgo tiene principalmente dos componentes:

- Primero, los errores de importancia que pueden producirse durante el proceso en sí.
- Por otro lado, la falta de detección de irregularidades cuando el auditor hace su análisis o examinación.

_ Entonces para mitigar estos riesgos el auditor por un lado debe:

- Confiar en el control interno de la entidad auditada, en donde el control interno como dijimos es el que va a llevar ese registro día a día de cada una de las implementaciones de controles que siguen a los activos críticos para asegurarnos de que cumplen con las condiciones que les fueron dispuestas.
- Y por otra parte confiar en las pruebas de detalle que realiza el auditor y en los procedimientos que mientras más detallados y eficientes sean mejor.

Variación del objeto de auditoria - CAATs

_ Con la introducción de la informática a todos los procedimientos de auditoria en su momento en sus diferentes áreas se realiza lo que se llama técnicas de auditoria asistidas por computadora (CAATs), entonces estas técnicas que son automatizadas tienen ciertas ventajas con respecto a los procedimientos manuales como ser:

- Bajo costo de puesta en marcha, porque por lo general ya se reutilizan soluciones propuestas por otros profesionales o por otras empresas.
- Bajo costo de operación, porque por lo general es prácticamente instalar y correr herramientas.
- Un aumento del rendimiento continuo, porque en el caso de que estas herramientas nos sirvan, vamos a ir aprendiéndolas y estos procedimientos van a ser cada vez más eficientes, vamos a tener menos costo y por lo tanto vamos a tener un mejor rendimiento de las inversiones.
- Excelente consistencia, en el sentido de que cuando uno corre una actividad manualmente puede estar sujeta a error humano y cuando corremos algo automatizado ante las mismas entradas obtenemos las mismas salidas.

_ Los procedimientos manuales tienen además algunas ventajas por sobre los automatizados, que están más relacionadas al factor humano, y son:

- Mejor capacidad de reacción ante lo inesperado, porque por lo general las herramientas automatizadas vienen parametrizadas y no permiten desviarse cuando aparece un factor inesperado.
- Incorporación del sentido común, no podemos esperar que una herramienta automatizada pueda tomar una decisión en un contexto difícil y por ahí es la experiencia de alguien que pueda manualmente autorizar o aprobar cierta operación.
- Mejor lenguaje para comunicar, especialmente para el caso en donde hay irregularidades o hay desviaciones al proceso original. Ya confiamos en el factor humano para poder comunicar cuales fueron esos hallazgos.

Concepto de consultoría

_ La consultoría consiste en “dar asesoramiento o consejo sobre lo que se debe hacer o cómo llevar adecuadamente una determinada actividad para obtener los fines deseados”. Los elementos de la consultoría podrían resumirse en:

Contenido: consiste en dar asesoramiento o consejo, entonces el contenido ya es diferente, antes en la auditoria teníamos que emitir una opinión y ahora es asesorar y aconsejar.

Condición: la condición es que este consejo es de carácter especializado, es decir, alguien que tiene respaldo ya sea con una carrera o certificación.

Justificación: este asesoramiento se justifica en base a un examen o análisis como vimos antes, la auditoria era basada en procedimientos, y acá estamos examinando el funcionamiento de algo.

Objeto: el objeto bajo estudio es una actividad o es una cuestión sometida a consideración, por ejemplo una aplicación, un entorno, un sistema operativo, etc.

Finalidad: la finalidad es establecer la manera de llevarla a cabo adecuadamente, es decir, que oportunidades de mejora encontramos sobre esto que hemos analizado.

Momento: el momento es a priori. Acá estamos viendo cómo se está desarrollando una aplicación y no venimos cuando la aplicación finalizo para verificar cosas como en la auditoria, sino que estamos viendo cómo se organiza por ejemplo un equipo de desarrollo y podemos hacer sugerencias en cuanto a planificación, a mejoras prácticas y demás.

Control interno y auditoria informática

Control interno

_ Como mencionamos, control interno es el área que esta internamente en una organización y que va preparando todo el terreno para lo que es una auditoria. Para reforzar el control interno, las organizaciones pueden recurrir a algunas iniciativas como lo que es:

- Reingeniería de procesos, con el objetivo de si un proceso no se está ejecutando de manera adecuada como hacemos para que de las salidas correctas, y nos proporciones la evidencia que un auditor va a necesitar.
- Gestión de la calidad total, porque sin duda en las auditorias también se analizan factores como la disponibilidad y para esto es importante que cada uno de los procesos de la empresa cumpla con ciertas funciones de calidad en cuanto a tiempos de respuestas sobre todo cuando hablamos de tecnología.

- Redimensionamiento del tamaño (downsizing, upsizing), aquí puede haber cambios organizacionales como reducción de personal o agregar personal en el caso de que ciertas funciones no se estén cumpliendo de manera adecuada.
- En el caso de que no podamos contratar directamente este personal, recurrimos a lo que es tercerización (outsourcing).
- Descentralización, apunta a que por lo general en las organizaciones en las grandes empresas no todo está centralizado en un solo edificio sino que se busca la eficiencia a través de ubicar en diferentes sedes diferentes procesos, y puede ser útil cuando la organización que estamos asistiendo tiene cuestiones como logísticas involucradas y puede facilitar esas tareas.

Control interno informático

_ Ya metiéndonos un poco más en profundidad se ocupa de:

- Que todas las actividades de sistemas de información sean realizadas cumpliendo los procedimientos , estándares y normas fijados por la dirección de la organización o la dirección de informática, así como los requerimientos legales. Acá es importante tener en cuenta que la empresa puede tener sus propios valores pero cuando está obligada a cumplir con alguna regulación debe adaptar sus procedimientos, estándares y normas internas a esta regulación.
- Asesorar sobre el cumplimiento de las normas, es decir va concientizando cada una de las áreas, especialmente aquellas que están directamente relacionadas a la normativa que se quiere cumplir.
- Colaborar con los esfuerzos de auditoria informática y otras auditorías externas que pueden ser no solo cuestiones informáticas sino también otro tipo de certificación. Por lo general la auditoria informática es ejecutada por un ente externo a la organización, donde ellos preparan todo el terreno.
- Ayudar a definir, establecer y ejecutar mecanismos y controles para comprobar el logro de los grados adecuados del servicio informático. Básicamente que nuestra área informática esté funcionando de acuerdo a lo que planificamos y que esa planificación este acorde a los objetivos organizacionales.

_ El control interno informático se realiza, en los diferentes sistemas (hablando de sedes centrales, departamentales, redes locales, equipos o PCs puntuales de los usuarios, etc) y entornos informáticos diversos (como lo son producción, desarrollo o pruebas), el control de las diferentes actividades operativas sobre:

- El cumplimiento de procedimiento, normas y controles dictados, especialmente sobre el control de cambios y versiones de software. Ver si tenemos un sistema de change management donde vamos registrando cuales son los cambios que queremos hacer en producción y si esos cambios están aprobados efectivamente por el dueño del recurso y por ejemplo si estamos teniendo en cuenta que va a

haber usuarios que van a quedar sin disponibilidad por un cambio que puede tener una ventana de mantenimiento.

- Controles sobre la producción diaria, es decir, que tan bien se va avanzando en el día a día con los objetivos propuestos.
- Controles sobre la calidad y eficiencia del desarrollo y mantenimiento del software y del servicio informático, es decir, que tan bien ven el resto de las áreas de la empresa a la ejecución del área informática.
- Controles en las redes de comunicaciones, básicamente monitorear y asegurarnos de que los recursos de comunicación se están usando para los fines para los cuales fueron planificados.
- Controles sobre el software de base, puntualmente sistemas operativos, hardware, middleware que este soportando la organización.
- Controles en los sistemas microinformáticos. Acá más relacionado a todo lo que es hardware que soporta el software de la organización.
- La seguridad informática, o sea los usuarios, responsables y perfiles de uso de archivos y bases de datos. Asegurarnos de que tenemos los mínimos privilegios implementados en cada uno de los recursos críticos en la organización. Que se establezcan normas de seguridad básicas y especificas también para diferentes tipos de activos. Y el control de la información clasificada que también nos va a ayudar a delimitar ese acceso y esa segmentación de perfiles de usuarios, acá hablamos de información confidencial, privada y pública, y eso nos va a ayudar por un lado a definir y a clasificar la información, y por otro lado segmentar los permisos.
- Licencias y relaciones contractuales con terceros, cuando incluimos o incorporamos un tercero también es importante ver si ellos consideran la seguridad tanto como lo hace nuestra organización.
- Asesorar y transmitir cultura sobre el riesgo informático a las diferentes áreas , es decir, esto también incluye la concientización interna en la organización.

_ Una definición entonces del control interno informático es cualquier actividad o acción realizada manual y/o automáticamente para prevenir, corregir errores o irregularidades que puedan afectar al funcionamiento de un sistema para conseguir sus objetivos. Debe definir cuáles son los diferentes controles periódicos a realizar en cada una de las funciones informáticas, conforme a los objetivos de negocio y dentro del marco legal aplicable de acuerdo a la normativa que hayamos seleccionado. Además debe realizar periódicamente la revisión de estos controles establecidos, informando las desviaciones a la dirección de informática, que es la que va a estar de alguna manera supervisando esto, y preparando el campo para la auditoria, y sugiriendo cuantos cambios serán convenientes en los controles, así como transmitir constantemente a toda la organización la cultura y política del riesgo informático, es decir, concientización y acompañar a la implementación de medidas de seguridad en diferentes niveles de la organización.

Auditoria informática

_ Es el proceso de recoger, agrupar y evaluar evidencias (el control interno va a ir preparando todo lo que es evidencia) para determinar si un sistema informatizado protege los activos, mantiene la integridad de los datos y lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos. Acá es importante mencionar la triada de seguridad que es confidencialidad, disponibilidad e integridad. Luego la auditoria informática sustenta y confirma que esos objetivos se consiguieron efectivamente.

_ El auditor evalúa y comprueba en determinados momentos del tiempo los controles y procedimientos informáticos más complejos. Hay algunas auditorias que ocurren semestralmente, otras una vez al año, entonces son en determinados momentos del tiempo que queremos revisar que tan bien se fueron implementando estos controles a lo largo del año o cual fue el último reporte de los controles que ejecutamos, que tan bien nos dieron y que acciones se están tomando con las irregularidades.

_ El auditor es responsable de revisar e informar a la dirección de la organización sobre el diseño y funcionamiento de los controles implementados y sobre la fiabilidad de la información suministrada. Entonces en un proceso de auditoría, por lo general el auditor es alguien imparcial externo a la organización, hace su evaluación, prepara un informe y le informa a la dirección de la organización que tan bien están funcionando esos controles y que tan confiable somos hacia el exterior.

_ Se pueden establecer tres grupos de funciones a realizar por un auditor informático:

- Por una parte participar en las revisiones durante y después del diseño, realización, implementación y uso de aplicaciones informáticas así como en las fases análogas de realización de cambios importantes, es decir, en esta etapa el auditor se interioriza de que sería interesante auditar en un futuro.
- Revisar y juzgar los controles implementados en los sistemas informáticos para verificar su adecuación a las ordenes e instrucciones de la dirección, requisitos legales, protección de confidencialidad y cobertura ante errores y fraudes, es decir, no solamente consideramos los valores de los objetivos organizacionales sino también que nos imponen las leyes y las normativas con las cuales queremos regirnos.
- Revisar y juzgar el nivel de eficacia, utilidad, fiabilidad y seguridad de los equipos e información. Básicamente que tan bien estamos implementando buenas prácticas en nuestros equipos, que tan bien están funcionando y que tan eficientes son cuando intentan respaldar el negocio.

Control interno y auditoria informática: campos análogos

_ A continuación vemos cuales son las similitudes y diferencias entre control interno y auditoria informática:

Similitudes:

- En el control interno siempre podemos contar con personal interno. En la auditoria informática el personal interno facilita la tarea pero no esta tan directamente involucrado.
- Ambas requieren conocimientos especializados en tecnología de la información. Esto por supuesto de acuerdo a la organización que estemos evaluando va a variar, van a ser diferentes tipos de hardware y software sobre los cuales deberíamos esta capacitados.
- Verificación del cumplimiento de controles internos, normativa y procedimientos establecidos por la dirección de informática y la dirección general para los sistemas de información. Por ejemplo que definió la organización internamente y por supuesto apoyado en diferentes normativas.

Diferencias:

_ Con respecto a el control interno informático:

- Es un área que esta todo el año dentro de la organización, hace controles sobre las actividades en el día a día.
- Informa solamente la dirección del departamento de informática, es decir, se ocupa de validar cuestiones informáticas y su información se limita solamente a este departamento.
- Cuenta solamente con personal interno.
- El alcance de sus funciones cubre únicamente lo que se propone desde el departamento de informática o todas las acciones relacionadas con este departamento.

_ Con respecto a la auditoria informática:

- Es una análisis en un momento determinado en el tiempo, puede ser una vez al año, trimestral, de acuerdo a la normativa que estemos persiguiendo.
- Informa a la dirección general de la organización, o sea ya es un tema que involucra a toda la organización.
- Tiene personal tanto interno como externo, por lo general las auditorias que son certificaciones importantes es personal externo o sea otras organizaciones que envían sus auditores y ellos preparan un informe y nos dan en definitiva la certificación final.

- Tiene cobertura sobre todos los componentes de los sistemas de información de la organización, es decir, no solamente el funcionamiento del departamento de informática sino todos los que sean sistemas informáticos en cada una de las diferentes áreas.

Sistema de control interno informático

_ Los objetivos de los controles informáticos se han clasificado en las siguientes categorías:

Controles preventivos: son los que se proponen con la idea de anticiparse a incidentes.

Controles detectivos: intentan alertar lo más temprano posible cuando ocurre una irregularidad o un incidente.

Controles correctivos: cuyo objetivo es reestablecer un servicio lo antes posible, que haya sufrido un tipo de incidente.

_ Para conocer las configuraciones de cada uno de los sistemas que queremos auditar es necesario documentar los detalles de la red, así como los distintos niveles de control y elementos relacionados:

- Entorno de red, para entender cómo funcionan las comunicaciones entre los diferentes componentes.
- Configuración de los equipos, para ver como implementamos las mejores prácticas y restringimos las configuraciones para que sean optimas del punto de vista de protección de información.
- Entorno de aplicaciones: para entender si únicamente nos estamos comunicando a una base de datos interna o si estamos interactuando con otros sistemas externos.
- Productos y herramientas que estamos utilizando, porque por supuesto cuando incorporamos productos de terceros también podemos introducir potenciales vectores de ataque a la organización.
- Seguridad de los equipos, tanto física como lógica, que medidas se están tomando en cada uno de ellos. Y además que tan bien concientizamos a los usuarios porque podemos tener todas las medidas de seguridad implementadas pero si el usuario no está preparado ataques de ingeniería social por ejemplo, aun así un atacante puede obtener acceso a información crítica.

_ Para establecer un sistema de controles internos informáticos habrá que definir:

- Como vamos a realizar la gestión de sistemas de información a nivel general en la organización. Como van a ejecutar todos los sistemas que estén interrelacionados
- Como se realiza la administración de sistemas a nivel individual, que medidas se están tomando.

- Cuáles son las implementaciones de seguridad que tenemos a nivel general y a nivel particular.
- Como se está gestionando el cambio, es decir, que medidas se están tomando cuando ocurre un cambio a nivel aplicaciones , por ejemplo se están agregando nuevos controles, validando nuevamente la seguridad, etc.

_ Cada función de la organización juega un papel importante en las fases de establecimiento de una política y cultura sobre la seguridad:

- Por un lado tenemos la dirección de negocio o dirección de sistemas información (SI), que deben estar comunicadas. Es importante que desde la dirección de negocio bajen líneas de que la seguridad es un factor importante con lo cual control interno informático tiene libertad de acción para implementar las diferentes medidas que requiere.
- Por otra parte la dirección de informática proponiendo las mejores prácticas para el resto de la organización.

Auditor interno/externo informático

_ Debe revisar los diferentes controles internos definidos en cada una de las funciones informáticas y el cumplimiento de normativa interna como externa de acuerdo a los niveles de riesgo, conforme a los objetivos definidos por la dirección del negocio y la dirección informática. Y por otra parte va a informar a la alta dirección de los hechos observados, es decir, irregularidades y al detectarse deficiencias o ausencias de controles va a recomendar acciones que minimicen los riesgos que pueden originarse.

Controles internos para sistema de información

_ Tenemos:

- Controles generales a nivel organizativo impuestos desde la alta dirección.
- Controles específicos de desarrollo, adquisición y mantenimiento de sistemas de información. Es importante tener bien delimitado lo que es desarrollo interno y como son las relaciones con terceras partes que nos proveen sistemas aplicativos o de hardware.
- Controles de uso de los sistemas de información, tener una política de uso aceptable entre nuestros usuarios y poder también delimitar de manera restrictiva el acceso y la ejecución de determinadas acciones de acuerdo a las necesidades que tengan con el fin de preservar el negocio.
- Controles en las aplicaciones, tanto a nivel seguridad como a que se eviten cuestiones como fraude.
- Controles específicos en ciertas tecnologías, ya que por ahí hay ciertas tecnologías que no vienen del todo pensadas para proteger la confidencialidad y es necesario implementar controles compensatorios.

Herramientas y etapas de la auditoria informática

Objetivo de la unidad: identificar cada una de las herramientas utilizadas en el proceso de auditoría, y escribir brevemente las etapas del proceso de auditoría para una aplicación informática. Algunas herramientas no se usan en la actualidad, dependiendo del tipo de organización que auditemos y el tipo de regulación.

Herramientas en auditoria de una aplicación informática

Entrevistas: debemos pensarlas para ser ejecutadas a las personas de la organización que creamos que más van a aportar al propósito pretendido o a lo que queremos evaluar. Puede ser personal de la dirección o alguien del último nivel que esta al día a día de la operación. Se requiere conocer lo suficiente sobre la aplicación para conducirla y obtener la mayor cantidad de información evitando preguntas cerradas, con lo cual es importante que previo a la ejecución de esta auditoria contemos con documentación de la aplicación que vamos a auditar o al menos con una demo que nos dé una idea de cómo funciona la aplicación (software).

_ Debe coordinarse con antelación suficiente, especificando los temas a tratar, duración aproximada y solicitando la información o documentación necesaria para ese momento de la entrevista, obviamente deberíamos contar con información previa para también saber sobre que vamos a indagar. Los supervisores de los entrevistadores deben estar informados de cada una de estas reuniones que se van a establecer, por supuesto que tiempo vamos a usar de ellos y deberían ser ellos quienes comuniquen a estos entrevistados y que expliquen por un lado cual es la necesidad de auditoría y que destinen el tiempo necesario o los recursos necesarios para que podamos llevarla a cabo.

_ En el momento de la entrevista el auditor va a tomar las anotaciones correspondientes y las va a revisar con el interlocutor para asegurarse que sus aportes estén reflejados correctamente y estén interpretados.

Encuestas: a diferencia de las entrevistas apuntan a una mayor cantidad de usuarios. Se utilizan para determinar el alcance y objetivos de la auditoria y para medir el nivel de satisfacción de los usuarios, por ejemplo con algún proceso o alguna transacción especifica de la aplicación. Se aplican la mayoría de los requisitos enunciados para las entrevistas, en el sentido de planificar adecuadamente, requerir documentación previa, la comunicación con el supervisor, etc.

_ Consiste en preparar un cuestionario que pueda ser respondido rápidamente a base de marcar entre las respuestas posibles. Y además de las respuestas, poder también establecer algunos campos de observaciones donde el que está respondiendo la encuesta

pueda explayarse en el caso que desee hacerlo y que esto nos permita indagar en el caso de encontrar alguna respuesta interesante poder volver a esa persona y profundizar. La encuesta por lo general busca ser anónima, aunque puede estar la opción de que digamos el nombre o dejemos que sea opcional para poder contactar a esa persona para enriquecer asuntos de interés en los que se quiera profundizar.

Observación del trabajo de los usuarios: acá nos metemos en el día a día de las operaciones y de lo que queremos auditar y nos ponemos a ver como efectivamente se hace uso de esta aplicación, como se ejecutan las transacciones obviamente priorizando por volumen o por riesgo de lo que queremos auditar. Esto ayuda a identificar irregularidades e ineficiencias, es decir, nos ayuda a ver que tan bien están usando la aplicación o si la están usando de manera inadecuada o haciendo transacciones que no se deberían hacer.

_ Debe aprovecharse esta oportunidad para probar también la efectividad de los controles de las transacciones (tanto nosotros como auditores como los que tiene la organización a nivel interno para prevenir fraude por ejemplo), solicitando la simulación de situaciones previsibles de error.

Pruebas de conformidad: se orientan a comprobar que determinados procedimientos, normas o controles internos se cumplen o funcionan de acuerdo con lo previsto y esperado, según lo documentado que respalda dicha aplicación. La comprobación debe llevar a la evidencia a través de los resultados producidos, es decir, debemos tener registros, documentos, etc, y/o observaciones del funcionamiento de un control ante pruebas específicas de comportamiento. Se debe elaborar informes de excepción cuando existe alguna irregularidad o evidencia de incumplimiento de lo que se ha establecido.

_ Los testimonios de incumplimiento aislados no implican evidencia, pero cuando son obtenidos de varias personas pueden derivar en posibles recomendaciones, o sea cuando varios usuarios informan de una irregularidad, ahorrando esfuerzos en obtener evidencias documentadas.

Pruebas sustantivas o de validación: buscan detectar la presencia o ausencia de errores o irregularidades en procesos, actividades, transacciones o controles internos integrados en ellos. Se buscan irregularidades externas que puedan afectar a las transacciones. Múltiples recursos pueden ser utilizados, muchos de ellos se apoyan en la utilización de herramientas informáticas. Otros recursos clásicos utilizados para la detección de errores o sus indicios son de ejecución manual. Normalmente se aplican sobre muestras, estadísticas y no estadísticas.

_ En definitiva acá buscamos ver si tenemos controles internos para lo que es evitar errores o irregularidades, es decir, no vemos tanto que está documentado sino que vamos a otro nivel y vamos a validar cuestiones que pueden no estar documentadas o la aplicación puede no estar preparada, con el fin de poder informar esas irregularidades.

Uso de equipos informáticos: con el avance del tiempo, las actividades comenzaron siendo manuales y moviéndose a soluciones más automatizadas. Existen en el mercado infinidad de productos de software concebidos para facilitar la tarea del auditor (sobre todo en entornos de nube), aunque se pueden obtener resultados similares haciendo uso de herramientas disponibles en la organización como lo son las bases de datos.

_ Los registros de auditoría de que este provista la aplicación deben constituir un apoyo importante a la hora de utilizar equipos para detectar situaciones o indicios de posible error. Es importante que nos aseguremos que la aplicación que estamos auditando provea información suficiente.

_ También hay que considerar la posibilidad de utilizar la propia aplicación, aplicando juegos de ensayo o transacciones físicas preparadas por los auditores, para verificar la eficacia de los controles implementados. Pruebas de validación que apuntan a descubrir estas irregularidades.

Etapas de la auditoria de una aplicación informática

Obtención de información y documentación sobre la aplicación: esto es lo que mencionábamos como fundamental previo a la auditoria. Se necesita tener un conocimiento básico de la aplicación y su entorno, para identificar puntos débiles y riesgos. Hacemos énfasis en su entorno, no solamente nos interesa la aplicación en si sino también que es lo que la rodea y como está protegido lo que rodea a la aplicación.

_ Se inicia el proceso con entrevistas tanto a los usuarios de la aplicación como a los responsables de los sistemas de información, hasta poder establecer los objetivos de la auditoria. En definitiva de las entrevistas queremos sacar lineamientos para establecer el alcance de la auditoria. Es necesario tener una guía que establezca pautas y documentos a solicitar, es decir, todo lo que ayude a:

- Adquirir una primera visión global del sistema.
- Conocer la organización y los procedimientos de los servicios que utilizan la aplicación.
- Describir el entorno en el que se desarrolla la aplicación.
- Entender el entorno de software básico de la aplicación, cuáles son los componentes de software, cual es el software de base, el sistema operativo, la base de datos, en que framework esta desarrollada, si se aplican parches de seguridad, etc.
- Asimilar la arquitectura y características lógicas de la aplicación, por ejemplo si tenemos acceso a toda la información o tenemos restringido el acceso a ciertos datos.
- Conocer las condiciones de utilización de la aplicación y los riesgos que se pueden dar, es decir, prevenir el uso inadecuado de la aplicación que se esté usando para

la cual fue definida y no para otro fin, y analizar los riesgos desde diferentes puntos de vista.
- Conocer las condiciones de seguridad de que dispone la aplicación, es decir, estamos usando mejores prácticas, testing, etc.
- Disponer de información estadística, métrica, indicadores, informes, etc, que nos ayuden a ver que tan bien están los tiempos de respuesta, monitorear la actividad, etc.

_ Es decir, no solamente consideramos cual es el entorno informático que rodea a esa aplicación sino en que contexto esta, si están los usuarios concientizados y capacitados, si tenemos medidas de seguridad física para la organización.

Determinación de los objetivos y alcance de la auditoria: las observaciones tras el examen preliminar, la identificación de los puntos débiles y las funciones críticas, deben permitirle al auditor establecer su propuesta de objetivos de la auditoria de la aplicación y un plan detallado del trabajo a realizar. De todas las actividades que hicimos previamente, por ahí definimos cual es el alcance y los objetivos de la auditoria y armamos un plan de trabajo.

_ En la preparación del plan de trabajo trataremos de incluir:

- La planificación de los trabajos y el tiempo a emplear.
- Las herramientas y métodos.
- El programa de trabajo detallado.
- Pruebas de confirmación, pruebas sobre los datos y los resultados que vamos a ejecutar.

Planificación de la auditoria: la auditoria de una aplicación informática debe tener una planificación cuidadosa. No es conveniente que la planificación coincida con el momento de la implementación de la aplicación. Por lo general debe ser en un momento posterior. No debe tener un retraso excesivo que prolongue el periodo de exposición a riesgos.

_ Hay que establecer el ámbito de actuación priorizando lugares donde estén la mayor cantidad de pruebas a realizar, por ahí funciones más críticas de la aplicación, que haya representatividad de usuarios y los costos sean razonables.

_ El personal de auditoria debe contar con las autorizaciones necesarias para acceder a la aplicación y a las herramientas de usuario. Es decir que desde la dirección de la organización deberíamos tener el apoyo suficiente para que nos permitan por un lado acceder a la aplicación y por otro comunicarnos con usuarios para que ellos puedan mostrarnos como hacen uso de la aplicación y nosotros poder sacar conclusiones.

Trabajo de campo, informe e implementación de mejoras: el trabajo de campo es la actividad en donde efectivamente se ejecuta todo el plan que hicimos con anterioridad. Consiste en la ejecución del programa de trabajo establecido.

_ En la etapa de redacción del informe de auditoría, se reunirán las características del trabajo realizado y sus conclusiones y recomendaciones o propuestas de mejora.

_ En la etapa de implementación de las mejoras identificadas en la auditoria, la situación óptima de alcanzar es conseguir que la organización auditada asuma las propuestas de actuación para implementar las recomendaciones como objetivos de la organización. Es decir que no solamente solucionen el problema que informamos sino que esto se incluya a largo plazo como un objetivo organizacional.

Principios de auditoría, evaluación de Riesgos y reportes efectivos

Objetivo de la unidad: los objetivos en esta unidad son identificar términos básicos de auditoría, comprender el rol de la auditoría en la mitigación de riesgos, integrar herramientas de identificación de riesgos y causa raíz, y reconocer los pasos del proceso de auditoría.

Metodología general

_ Nosotros como auditores nos preguntamos cómo auditar la seguridad en una empresa, y por lo general la respuesta a esta pregunta es empezar por auditar algo pequeño, no tratar de abarcar un todo sino tratar de separar el problema en partes, áreas, procesos, y empezar a ver a que se tienen que alinear esos procesos y definir los controles que estén por un lado alineados a la política de la organización y por otro lado al estándar que queremos evaluar. Entonces en la metodología general hablamos de un proceso de auditoria basado en riesgos donde por lo general pensamos que vamos a auditar la organización desde afuera hacia adentro porque cuando estamos dentro de la organización asumimos que tenemos menos riesgos con la gente que tenemos en nuestra organización (empleados, colaboradores), que hacia afuera donde puede haber otro tipo de intereses y otro tipo de atacantes. Cuando hablamos también de basado en riesgos, vamos a descomponer un gran sistema en cada uno de sus componentes y vamos a tratar de identificar los riesgos sobre cada uno de esos componentes y podemos analizar pieza por pieza e ir viendo si se ajustan a las políticas y a los controles definidos.

Términos de auditoria

Auditoria: de alguna manera significa medir algo contra un estándar, sea un estándar de desarrollo u otro tipo de documento que nos dé lineamientos de que es esperable de un sistema aplicación o dispositivo. Los 3 lugares más comunes para aplicar auditoría en tecnología de la información y seguridad de la información son:

- Nivel de política: es decir, la política nos dice en general que se espera que haga un determinado proceso.
- Nivel de procedimiento: sería un paso de un control por ejemplo que queremos ejecutar.
- Nivel de sistema (o nivel de aplicación): es decir, que esperamos que cumpla determinado sistema o determinado aplicativo dentro de la organización.

Tipos de auditoria: tenemos diferentes tipos:

- Auditoria de conformidad: se encarga de verificar que también un sistema o proceso está en conformidad con las políticas o procedimientos que han sido definidos en la organización.
- Auditoria de seguridad: se encarga de añadir políticas, procedimientos, o auditorias en sí mismas contra mejores prácticas de la industria.
- Auditoria de financiera: se encarga de verificar los estados contables y que los costos e ingresos de la compañía realmente reflejen lo que es la realidad.

Preguntas de auditoria

_ Para verificar si la política de seguridad esta implementada primero tenemos que ver si la política existe por supuesto y luego analizar si se está cumpliendo y si es efectiva. Asumimos que tenemos como ejemplo un nuevo firewall (o sistema de seguridad) que implementamos en nuestra organización, como sabemos si estamos realmente protegiéndonos ?, o si por ejemplo un administrador de la compañía nos dice que los sistemas están totalmente parchados y asegurados, vamos a creer ciegamente en esa persona ?, que preguntas nos realizamos en cada caso ?.

Objetivo principal de la auditoria

_ La función principal del auditor es actuar como una herramienta o intermediario de la Dirección y el resto de la compañía para medir y reportar sobre riesgos. Y como función secundaria una vez que el auditor haya identificado los riesgos, es reducir los mismos a través de la concientización, para que pueda mitigarlos de la manera adecuada.

Términos de auditoria (parte 2)

Evaluación: son medidas más arbitrarias o subjetivas si se quiere. Típicamente vamos a usarlas para medir que tan exitosa fue la ejecución de una auditoria, que tan bien asegurado esta un sistema, o que pudo salir mal, o sea que en definitiva nos sirve para medir o estimar:

- Riesgos
- Amenazas
- Vulnerabilidades

- Costo de exposición, o costo de que explote determinada vulnerabilidad.

Alcance: que es lo que estamos auditando o evaluando, también conocido como entidad auditable (departamento, aplicación, sistema, microservicio, base de datos, o llegar hasta el nivel que se nos ocurra). Esto define un área de autoridad, es decir, donde nosotros como auditores vamos a poder pedir información o hacer recomendaciones de acuerdo a un estándar. Y además define un área de responsabilidad, es decir, hasta donde podemos hacer sugerencias, que es lo que podemos hacer, por eso decimos que el alcance es definir el "que" puntualmente.

¿Quién define el alcance?

_ El alcance esta por un lado definido por el solicitante de la auditoria inicialmente, no estamos diciendo el alcance definitivo, o también por la dirección de la empresa, en el caso de que sea una empresa grande puede ser que el solicitante no sea un miembro de la dirección. El auditor de alguna manera podría definir en parte el alcance para una empresa que no sepa por donde iniciar. Pero es importante que ese alcance quede bien definido desde el inicio, que se definan bien cuales van a ser los procedimientos que van a definir el numero fino de tareas, tiempo, etc, para no arrastrar el alcance. Y tener en cuenta sobre todo que el auditor no exceda su autoridad que le fue concedida para esta auditoría, es decir, que respete los puntos que se le solicitaron, que defina los procedimientos, que evite irse fuera del alcance que definió inicialmente, y sobre todo involucrarse en temas que no son los que se especificaron para la auditoria.

El "que" versus el "como"

_ Como dijimos antes el "que" determina el alcance, o el alcance es el "que" necesitamos hacer. En principio no necesitamos considerar el "como" en la etapa inicial de la definición de la auditoria pero si necesitamos considerarlo en una etapa posterior.

¿Cómo hacer una validación técnica de un firewall?: una forma es que si tenemos una computadora protegida por un firewall, intento atacar a la misma y vemos que si mi ataque funciona es malo el firewall, y si no funciona es bueno.

Validación Técnica de Firewall

_ Si nosotros solamente revisamos las reglas del firewall, solamente sabe lo que las reglas dicen y que es lo que se está configurando en realidad, pero no vemos si efectivamente está cumpliendo bien con lo que se propone. Entonces el firewall como dijimos antes debe ser testeado, y lo que hacemos es disparar paquetes al firewall mismo para ver cómo responde, disparamos paquetes a diferentes dispositivos a través del firewall para ver si el mismo esta efectivamente está filtrando o permitiendo el tráfico. Y en base a eso también podemos hacer un diagrama de como viajan esos paquetes, o sea entendiendo cómo la información debe moverse a través del firewall y ver si las reglas que esperamos que se

ejecuten se están ejecutando. Entonces este “como” es bastante más complejo que leer solamente lo que está escrito en la configuración del firewall.

El “como”

_ Considerando el “como” de manera temprana. Si no sabemos el cómo implementar algo, lo que hacemos es ajustarnos a los objetivos o al alcance de la auditoria, es decir, tomamos como base lo que la empresa nos solicitó, lo que la dirección pide, lo que nosotros como auditores conocemos, o nos basamos en algún estándar que nos permita inicialmente definir ese cómo. Si cambiamos el alcance, que habíamos definido en el “que”, estamos limitando o modificando los riesgos que queríamos medir originalmente.

Términos de auditoria (parte 3)

Objetivo: que en definitiva es la meta o fin que tiene una política o procedimiento, o que tiene como objetivo la auditoria o la evaluación que quiera realizarse.

Objetivos de la política

_ Para saber los objetivos de una política primero deberíamos preguntarnos porque existe esta política, como se relaciona a la misión de la organización, una política que no esté alineada con los objetivos de negocio no tiene sentido, y se busca ver como protege a esa misión de la organización. Si tenemos el objetivo de que "todos los usuarios deben autenticarse con un usuario y contraseña", el objetivo organizacional que sugerimos es proteger la confidencialidad de la información de la empresa y tener por supuesto trazabilidad de quienes son los usuarios que están intentando acceder a nuestro sistema, si contamos con una cuenta genérica compartida por muchos usuarios va a ser difícil de identificar un atacante si ocurre un incidente, entonces de esta manera identificando unívocamente un usuario sabemos quién fue la persona que ejecuto determinada acción y por supuesto la contraseña debería ser única y no podría compartirse.

Términos de auditoria (parte 4)

Control: indica como alcanzamos nuestros objetivos. Responde la pregunta "Cómo sabemos que...", o sea como sabemos efectivamente que algo se está haciendo de la manera correcta. Un ejemplo podemos tener como objetivo la autenticación de usuario requerida, pedimos sí o sí que para entrar al sistema alguien se autentique, y como control tenemos por un lado un controlador de dominio Active Directory basado en Windows donde nosotros podemos definir diferentes reglas por ejemplo que el usuario este en el sistema hasta una cierta hora, y por otro lado tenemos un registro de eventos (logon/logoff/fallas de autenticación), cada vez que un usuario inicio sesión, cerro sesión o cuando intento por ejemplo cuando alguien intento hacer inicio de sesión con una contraseña registramos los intentos fallidos y podemos tomar acciones como bloquear esa cuenta temporalmente.

Excepción de auditoria: es el elemento que falla el cumplimiento del criterio de auditoría, o un control que falla para cumplir su objetivo. Es cuando queremos comprobar algo y vemos que efectivamente ese control no pasa exitosamente.

Remediación: es decir tenemos un incidente o irregularidad, nos preguntamos qué hacemos para corregirlo, por ende tenemos recomendaciones basadas en mejores prácticas, por suerte existen algunos documentos que ya nos dan indicios y no tenemos que pensar esas mejores prácticas, tenemos también recomendaciones basadas en políticas, ya la respuesta al problema ya puede estar escrita en la política de la organización, y recomendaciones basadas en procedimientos.

Mitigación: hace referencia a que hacemos para reducir perdidas o daños, aplica las mismas recomendaciones que para la remediación, es decir, recomendaciones basadas en mejores prácticas, basadas en políticas, y basadas en procedimientos.

Causa raíz: es lo que realmente salió mal, que no necesariamente está conectado de manera directa con la excepción de auditoría o la irregularidad sino que puede tener otro origen. Si las auditorías realizadas periódicamente continúan produciendo las mismas excepciones, no se está identificando la causa raíz, entonces podemos tener un auditor que nos está identificando una irregularidad, pero si el auditor no sugiere cual es la causa raíz o los colaboradores mismos no entienden cuál es esa causa , esta irregularidad va a seguir existiendo y va a seguir apareciendo como excepción en futuros informes de auditoría.

Líneas de Base

_ Por lo general las líneas de base es un método muy utilizado para analizar configuraciones de software, de sistemas operativos, de bases de datos, entonces aplicamos las mejores prácticas a una base de datos por ejemplo y sacamos una foto de esa configuración, las documentamos y decimos que las próximas bases de datos deberían cumplir mínimamente con este conjunto de configuraciones.

_ Medición de un sistema en un estado bueno conocido. Es decir las líneas de base son usadas para medir el estado actual de un sistema, es decir, tomamos esa base de configuraciones que cumple con las mejores prácticas, y cuando vamos a instalar la siguiente nos aseguramos que esa nueva base de datos cumpla con todas las configuraciones que definimos anteriormente. Es una de las mejores herramientas/métodos de Auditoría porque al tener esa base de configuraciones podemos de alguna manera automatizarla y no tener que hacer esos controles de manera manual. Y también las líneas de base son usadas para describir la configuración de un sistema.

Usando líneas de base: las líneas de base son de gran ayuda para la automatización porque con estas va a ser mucho más rápido el trabajo de lo queremos auditar. Permiten

auditar procesos en lugar de configuraciones. Para este trabajo, la línea de base debe ser confiable, también debe ser útil (quiere decir que replicada por prácticamente cualquier usuario), y también ser creativo en lo que se usa como punto de referencia y tener en claro el por qué. No necesariamente hay que ser creativo en que hacer sino en automatizarlo o en hacerlo de manera más eficiente a ese proceso.

_ Por ejemplo en un gráfico de línea de base de trafico de red, donde nosotros vamos capturando paquetes y podemos tener un comportamiento normal de nuestra red y cuando vemos que hay alguna anomalía disparamos alguna alarma. De alguna manera de eso sirve nuestra línea de base, no necesariamente es una configuración que se documenta, sino que también puede ser que vamos guardando información y hoy por hoy con machine learning este trabajo se hace mucho más fácil.

Seguridad basada en tiempo (TBS)

_ Es una manera simple y reproducible de que tan bien un sistema puede resistir un ataque a lo largo del tiempo. Nos sirve para auditar o evaluar a través del tiempo como una dimensión primaria. La Auditoría típicamente mide en qué medida uno sabe, en cuanto a que tan bien debe funcionar un proceso, cual es el tiempo de respuesta esperado y demás. También mide qué tan bien/cuánto tiempo y puede medir pérdidas potenciales/reales.

Ejemplo Air Canadá y WestJet: tenemos una compañía aérea WestJet que obtiene cierta "inteligencia competitiva" . Un antiguo empleado de Air Canadá trabaja ahora para WestJet , donde Air Canadá le suministró acceso a un sitio interno que contenía Información de carga de pasajeros, etc, y se usaba para la reserva de vuelos para empleados. El antiguo empleado le permite a WestJet usar su cuenta, entonces WestJet extrae información masivamente de 200.000 transacciones durante más de tres meses lo cual causa un problema a nivel competitivo porque pueden trabajar con ofertas y demás sabiendo las reservas que tenía hecho la competencia. Entonces, si nos preguntamos si esto se puede detectar este proceso, la respuesta es sí, las transacciones tenían logs, y si este usuario tenía un identificador único podíamos ver que era un trabajador que ya no pertenecía a la compañía y que durante más de tres meses estuvo usando su cuenta. Pero la pregunta real es qué tan rápido y qué tan bien, cuantos controles tenemos implementados por un lado para detectar estas transacciones que se hacen en un sistema y que tan bien se ejecutó el proceso de baja del usuario cuando ese usuario abandono la empresa, en donde en este ejemplo evidentemente la empresa no tenía un proceso bien preparado.

COBIT

_ Es uno de los estándares para lo que es seguridad en tecnología de la información, donde las siglas significan Objetivos de Control para Tecnologías de la Información (Control OBjectives for Information and related Technology). Este estándar provee un

marco de trabajo en diferentes secciones y prácticas de control en cada una de esas secciones. Podemos ver más información en el sitio.

Estándares de seguridad

_ Tenemos otros estándares de seguridad pero ahora, ¿se puede decir que se repiten?, esta es una pregunta interesante porque tenemos: PCI (industria tarjetas de crédito), FIPS, FISMA, DIACAP, GLB, HIPAA (industria de salid), C&A, ISO-27001 (se da mucha importancia en Europa), SARBOX, ALLIANCE (para la nube), etc. Entonces de alguna manera todos los estándares convergen a lo mismo, algunos tienen mayor nivel de detalle, otros más o menos requerimientos, pero por lo general se apunta a que los usuarios estén identificados, a cifrar la información en tránsito y en reposo, a que el acceso a red este delimitado controlado por un firewall. Muchos controles que aplican para un estándar también aplican para muchos otros.

_ Para saber cuál o como implementarlo, o que pasa si estamos sujetos a más de uno, como vimos en muchos casos hay coincidencia entonces es importante tener esos controles de manera genérica. Y la otra forma de saber cómo implementarlo es la evaluación de riesgos, es decir, por ejemplo tenemos la ISO-27001 donde no necesariamente vamos a cubrir el 100% pero vamos a ver cuáles son los componentes críticos de nuestra organización, a que riesgos están expuestos y en base a esos riesgos vamos a definir controles.

Estándares y Listas de control

Estándares: estos proveen una buena base para las listas de control, esto quiere decir por ejemplo que si un estándar nos dice el firewall no debe permitir acceso desde el exterior, debe restringir la comunicación entre puertos, esto ya nos da items que podemos incluir en una check list donde luego ejecutar esto sea a través de pasos más simples y no tener que leer un estándar completo. Tenemos:

- Listas de Control propias de los estándares.
- Listas de Control para comprobar controles.
- Listas de Control que buscan medir la conformidad, por ejemplo que un proceso nos devuelva un número determinado.

Listas de control: una buena lista de control debería incluir:

- Una declaración de propósito/alcance, es decir, para que tenemos esta lista y a donde queremos llegar con la obtención de esta información.
- Mejor práctica, que queremos evaluar.
- Que vamos a comprobar y medir.
- Como vamos a comprobar y medir.

Política y auditoria

_ Una buena política es necesaria para una buena auditoría. Entonces:

- La política principalmente responde quién, qué y por qué. Como dijimos al principio el alcance está definido por el "que", y la política va a ir con un poco más de profundidad y va a decir quién sería la persona responsable de ejecutar ese "que", y cuál es el fundamento o motivo por el que lo hace.
- El Procedimiento dice quién hace qué, cuándo y cómo, entra un poco más a nivel de detalle y es más un paso a paso de como cumplir con una política.
- La Auditoría mide el rendimiento de la organización con respecto a la política y el procedimiento, es decir, que tan bien la organización se está ajustando a esos dos componentes.
- El Manejo de incidentes y la auditoría pueden también servir como herramientas de evaluación de políticas/procedimientos, por ejemplo si tenemos un incidente de seguridad, que tan bien preparados estamos, si tenemos documentado el paso a paso de lo que deberíamos hacer, etc, y en el caso de que estemos trabajando con tranquilidad podemos ver, evaluando o entrevistando a las personas, que tan bien preparadas están de acuerdo a estas policías y procedimientos.

Política -> Procedimiento -> Auditoria

_ Yendo desde política hacia auditoria:

Política: tenemos una política que dice "todos los equipos de escritorio **deben** tener software antivirus con las firmas más recientes cuando son desplegados. Adicionalmente, todos los sistemas **deben** ser configurados para actualizar automáticamente las firmas de virus semanalmente". El deben esta resaltado para evitar ambigüedades, porque no es algo que nos parece que deberían hacer, sino que efectivamente todos los equipos deben tener ese antivirus instalado y sus firmas deben actualizarse semanalmente, si o si se debe cumplir.

Procedimiento de TI: tenemos por un lado un procedimiento para tecnología de la información donde "el administrador se asegurará que todos los nuevos sistemas desplegados tengan instalado el software antivirus XXXX (que la organización defina). El sistema debe ser configurado para obtener nuevas firmas desde xxxx.xxxx.com (sitio del fabricante o vendedor) cada semana, 1 hora después de que el sistema es iniciado". Entonces el procedimiento entra más en detalle, en este caso el antivirus tal, de la firma tal, en un tiempo determinado.

Procedimiento de la organización: ya no del administrador, "los empleados no deberían deshabilitar el software antivirus instalado por el Departamento de TI. Si el software reporta la detección de un virus, llamar al número xxxx (o mandar un mail a tal correo, etc) y hablar con xxxx para reportar el incidente".

Prueba de “Límite de Velocidad”

_ Esto es para hacer una analogía de lo que estuvimos hablando. Como objetivo tenemos que queremos incrementar la seguridad vial, reducir accidentes fatales, entonces queremos controlar que los autos que están circulando cumplan con ese límite de velocidad. Nos preguntamos si es suficiente con tener estas señales de tráfico, y la respuesta es no, nadie va a llamar a este número si usted está yendo rápido, entonces para asegurarnos que efectivamente ese control se está cumpliendo tenemos los radares de la policía caminera que intentan corregir las desviaciones a ese proceso.

Amenazas: Identificación y Evaluación

Vectores de Amenazas: tenemos los internos, que pueden ser:

- Intencional: no necesariamente debemos confiar en todo el personal interno de la organización, puede haber alguien mal intencionado que quiere robar información o dañar algún sistema.
- Accidental: cuando un usuario ejecuta una acción con un fin determinado y esa acción resulta perjudicial para la compañía no habiendo sido la intención de este usuario hacer ese daño.

_ Luego a nivel externo:

- Intención de pérdida o daño: ya si podemos contar con alguien que tenga la intención de causar perdidas a la organización o dañar algún sistema o la misma reputación de la empresa
- Accidental, como dijimos antes alguien que ejecuta determinada acción y el resultado termina siendo perjudicial no habiendo tenido intención de hacerlo.

Amenazas internas

_ Diferencias entre las accidentales y las intencionales:

Intencional:

- Eliminación de un sistema de archivos
- Exposición de propiedad intelectual
- Lanzamiento de “malware”
- Uso inapropiado de activos

Accidental:

- Eliminación de un sistema de archivos
- Exposición de propiedad intelectual
- Lanzamiento de “malware”

- Uso inapropiado de activos

_ Como vemos son exactamente las mismas, lo que varía es la intensión original del usuario. Cuando uno inicia como auditor o consultor de seguridad, uno de los principales temas que hay que abordar es el tema de la concientización, si un empleado no recibió concientización previa, es difícil pensar que el ataque o error fuera intencional, pero si se hizo una concientización, se evaluó, se aseguró que esa persona comprendió que estaba haciendo un daño si hacia tal cosa y aun así pasa entonces puede ser que sea intencional o accidental pero en definitiva hay que confiar en la persona, por eso debemos limitar a la persona de los mayores privilegios posibles para ejecutar algo en nuestro entorno. Podemos tener sistemas de detección, entre otras medidas.

Ejemplo:

- Amenaza (toma 1): el administrador recibe un correo indicando que un archivo adjunto infectado ha sido puesto en cuarentena en el servidor de correos corporativo. Esa es la amenaza.
- Amenaza (toma 2): el administrador está logueado como administrador de dominio aunque en realidad no está realizando tareas que requieran acceso administrativo. Es decir, se le reporto un incidente, se conectó al servidor donde está el archivo en cuarentena, pero lo está haciendo con altos privilegios lo cual ya de por si levanta una alarma.
- Amenaza (toma 3): el administrador hace doble click en el adjunto infectado para ver qué tiene adentro. Recordamos que estaba con permiso de administrador de dominio.
- Vulnerabilidad: como administrador de dominio, todos los controles de acceso sobre archivos se vuelven ineficaces. Aquí ocurre la vulnerabilidad, tenemos permisos absolutos, estamos ejecutando algo y por supuesto el sistema nos permite ejecutarlo porque somos administradores.
- Exposición: el virus corre contra el controlador de dominio, los recursos compartidos que están montados y otros dispositivos montados, por ejemplo directorios compartidos en máquinas de otros usuarios en otros servidores.

_ Entonces en este caso a lo mejor el administrador no tenía intención de que el malware se esparciera a través de la red, pero evidentemente fallaron muchos controles. La causa raíz pueden ser muchas, desde la falta de concientización del administrador, que el procedimiento no este definido para cuando un incidente de este tipo ocurra. Y obviamente no se cumplió el objetivo de proteger la red y la integridad de los sistemas de la organización. Un empleado puede ser malicioso debido diferentes motivos:

- Castigo
- Despido
- Rebeldía

Evaluación practica de riesgos

Objetivo del apartado: entender la aplicación de controles y estándares, entender porque lo que hacemos hoy no funciona a nivel de controles. Comprender los resultados de la evaluación de riesgos, poder seleccionar una estrategia. Y definir una evaluación de riesgos que funciona y que pueda utilizarse para especificar controles.

Controles y estándares

_ Las organizaciones requieren cumplir con estándares. El método típico que se utiliza para este cumplimiento es:

- Crear o definir un equipo con diferentes roles.
- Revisar el estándar y asignar tareas.
- Progresivamente aplicar los controles a la organización.

Las organizaciones fallan en las auditorias

_ ¿Por qué fallamos luego de seguir este método? porque los controles:

- Han sido aplicados sin analizar previamente riesgos. Simplemente tomamos un estándar, lo distribuimos entre nuestros equipos e intentamos ejecutarlo para determinada fecha tener la totalidad de esos controles abordados.
- Son aplicados sin una consideración real de su propósito. Los aplicamos porque están escritos en un documento sin pensar si son efectivos y si realmente agregan valor a nuestra organización.
- Son aplicados solo por aplicarlos.
- Son aplicados a la porción incorrecta de un proceso.
- No controlan causas raíz de las excepciones. Es decir, no estamos analizando cuales son los problemas que pueden tener los activos en nuestra organización sino que simplemente tomamos una lista de controles y los ejecutamos.

La hipótesis

_ La hipótesis de la que partimos es que las organizaciones aplican los controles de forma incorrecta. Sin mucho análisis y siguiendo un documento. El proceso correcto a seguir está a continuación.

El proceso correcto

_ Primero deberíamos identificar un estándar, como dijimos si la organización no está ligada a una industria en particular podríamos tomar ISO-27001 que es bastante genérico, y estos estándares ya tienen la definición de controles hecha por nosotros, en principio ya tenemos una lista de controles. Luego de esto deberíamos identificar los procesos críticos

para nuestra organización identificando la misión organizacional, cuáles son los procesos de negocio centrales. Realizamos evaluaciones de riesgo progresivas sobre la triada de seguridad (Confidencialidad, Integridad y Disponibilidad), empezamos desde alto nivel e ir descendiendo, desde el exterior al interior de la organización porque "asumimos que en el interior no tenemos tantos riesgos como en el exterior". La salida de la evaluación de riesgos debería identificar causas reales de riesgo, para reducir los riesgos, seleccionar controles desde el estándar que controlen los riesgos.

_ Es decir, primero identificamos activos críticos, identificamos cuales son los riesgos que pueden identificar a esos activos desde el punto de vista de la triada de seguridad, y luego vemos que controles nos ayudan a remediar o mitigar esos riesgos.

Resultados de la evaluación de riesgos

_ Para que esto funcione, la evaluación de riesgos debe:

- Ser usable y repetible, no nos sirve un procedimiento que sea extremadamente complejo y no podamos implementarlo.
- Identificar riesgos inaceptables, es decir, definir la criticidad y que es lo que la empresa estaría dispuesta a aceptar o no, invertir o no.
- Identificar causas subyacentes de estos riesgos.
- Establecer puntos de control para el riesgo evidente. En aquel riesgo que es inaceptable, ver cuáles son los pasos que podemos ir realizando para mitigarlo o controlarlo.

Selección de la estrategia

_ Lo que necesitamos es:

- Identificar riesgos inaceptables (con respecto a la triada de seguridad), cuáles son las fallas de controles críticos.
- Identificar causas subyacentes de fallas. Esto nos muestran dónde son necesarios los controles y nos ayudan a identificar la causa raíz de estas fallas.

Sobre la causa raíz

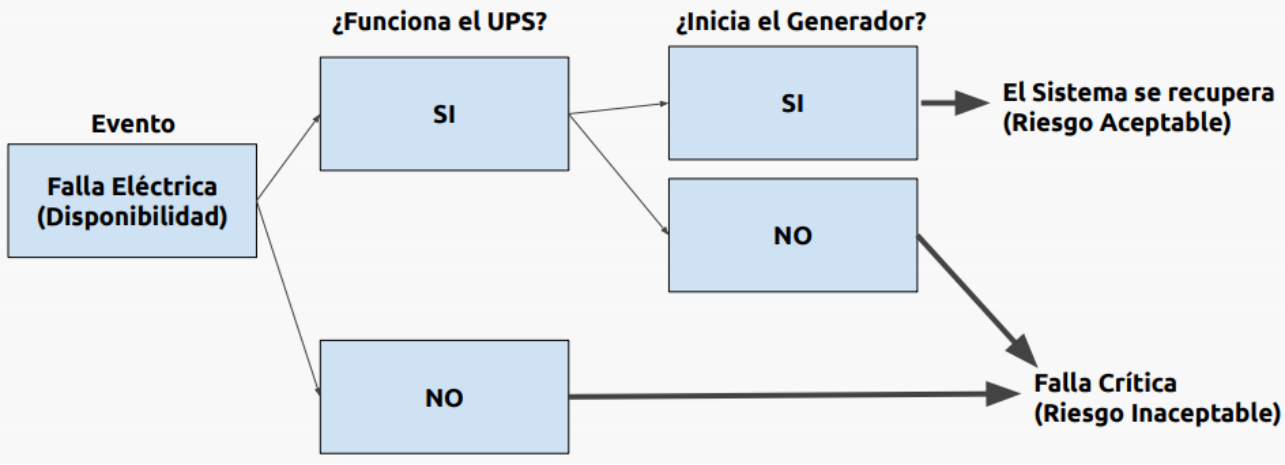
_ Por lo general cuando hay algún tipo de incidente o irregularidad hay una tentación real de culpar a la gente. A lo mejor la gente no tuvo la capacitación adecuada, a lo mejor le estamos dando más permisos de los que deberían, entonces puede haber fallas que están más allá de buscar que una cabeza ruede por un incidente sino que a lo mejor hay procedimientos que no están siendo ejecutados debidamente por alguien. Hay que evitar culpar a la gente. Y aún si la gente fuera la causa obvia, debemos estar perdiendo controles (detectivos, preventivos, correctivos) ya que la gente tuvo permitido comportarse como lo hizo.

Primer tipo de evaluación

Árbol de eventos: cuando hablamos de análisis de riesgos hablamos del árbol de eventos que consiste en analizar un sistema para identificar que controles detectivos podemos identificar y que controles reactivos que puedan restaurar el estado original del sistema. Tiene como objetivo identificar fallas críticas.

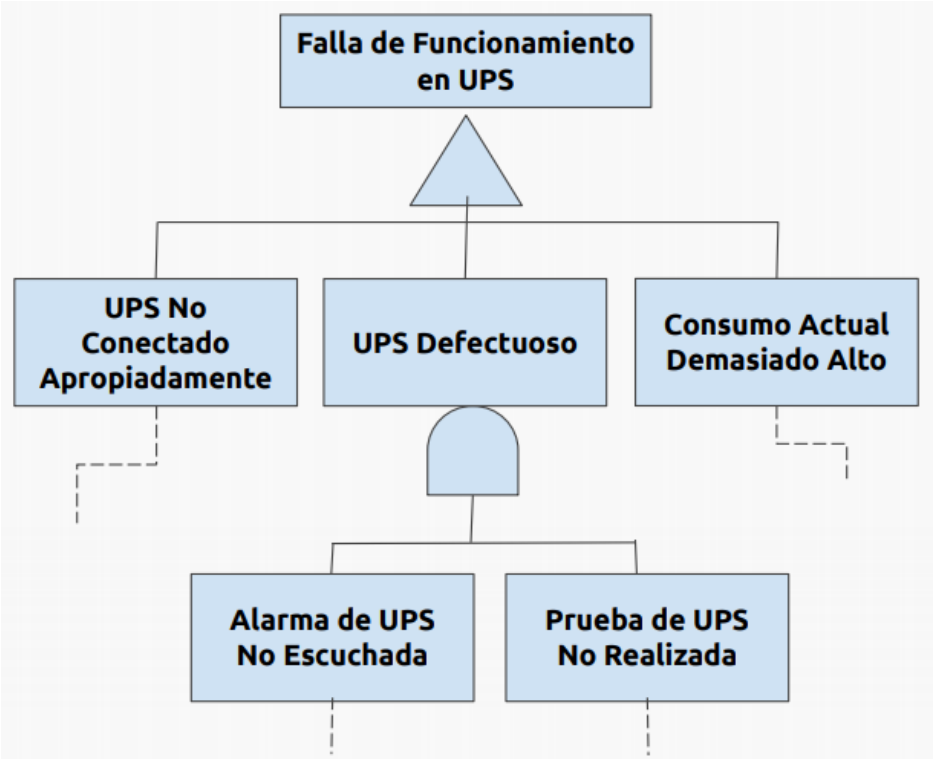
_ Para hacer este análisis lo que se hace es diagramar un escenario de fallas posible. Esto se asume que para los auditores es fácil y se asume también que si hay una excepción de auditoría, no significa eso que un control ha fallado, no significa esto que tenemos una falla crítica, ya que podemos encontrar que un control no cumple pero no es una falla critica. Vamos a identificar controles detectivos y correctivos rodeando al sistema, deben existir más de uno, porque no vamos a pensar en un solo escenario de falla sino que vamos a pensar en varios, y se busca encontrar todos y evaluar si detectan y reaccionan o no lo suficientemente temprano para prevenir una falla crítica.

_ Por ejemplo, un árbol de eventos debería mostrar lo siguiente:



_ Los árboles de eventos no son suficientes, pueden mostrarnos donde nuestro punto de falla crítica estaba y algunas veces la falla crítica real ocurrió más temprano de lo que los controles detectaron o reaccionaron, ¿no significaría esto que los controles están en el lugar incorrecto en el proceso?, y aún, en este ejemplo, no nos dice por qué falló.

Árbol de fallas: analizan una falla específica y determinan qué hechos subyacentes deben ser verdaderos. El resultado de la evaluación serán todas las causas (razonables) de la falla. Para hacer este análisis identificamos el evento crítico, y para el evento nos preguntamos qué causa subyacente debe ser verdadera. Repetir esto para cada causa subyacente hasta encontrar algo que está fuera de su control, por eso hablamos de causas razonables. En el siguiente ejemplo podemos tener diferentes fallas:



Uniendo las partes

_ Vamos a hablar de un análisis de causa consecuencia (CCA) que está compuesto por los métodos que vimos:

- Los árboles de eventos analizan las consecuencias de una falla, particularmente fallas críticas.
- Los árboles de fallas analizan las causas o fallas subyacentes, identificando idealmente la causa raíz.

_ Una vez que se identifican las causas subyacentes, se pueden escribir planes de tratamiento para controlar los problemas subyacentes.

¿Qué pasa con los auditores?

_ Estamos en una gran posición para encontrar fallas en los controles, estamos del lado bueno porque fuimos autorizados por la dirección a entrar en la organización y poder pedir explicaciones y documentación. Pero también como auditores es importante que podamos identificar esas causas raíces y poder, más allá de informar un problema, sugerir como remediarlo para que no aparezca en futuros informes de auditoría, o sea mejor que escribir el mismo problema cada seis meses es recomendar una manera de repararlo. Entonces las remediaciones que nosotros proponemos requieren también la evaluación de riesgos de la propuesta que estamos realizando.

Evaluación de riesgo

_ Por lo general se nos propone una fórmula que es amenaza x vulnerabilidad x probabilidad es lindo pero muy básico y no logra cubrir todas las necesidades y posibilidades para identificar riesgos. Lo único que hace es darnos un número que nos dice que algo es realmente arriesgado, y nos ayuda a identificar donde necesitamos controles pero no nos ayuda a definir en qué nivel deben estar implementados esos controles. Muchos métodos de evaluación de riesgos son similares cuando se trata de controles, pero no incorpora los dos métodos que vimos antes que por ahí son más específicos para problemas puntuales.

¿Es necesario hacer una evaluación de riesgos para cada excepción?: no, no es necesario hacerla, por eso nos enfocamos en elegir las fallas más críticas, donde identificamos primero los activos críticos para la organización y después cuales son las fallas que pueden causar mayor impacto o que tienen mayor probabilidad de ocurrencia. Por ejemplo listas de "Top 10" funcionan bastante bien y nos ayudan a enfocarnos en lo que realmente da más valor a la organización.

¿Es necesario realmente tener nuevos controles de seguridad?: no, como dijimos las chances son que el problema ya esté controlado a través de política implementada o a través de un marco de referencia de alto nivel al que la organización necesita estar adherido y alineado a esos controles. Debemos aplicar esos controles que ya tenemos de alguna manera definidos por la política o por el marco de referencia que estemos utilizando.

El proceso de auditoria

Objetivo del apartado: entender los objetivos del auditor y por otra parte comprender el proceso de auditoría de 6 pasos.

Objetivo principal del auditor

_ El objetivo principal del Auditor es medir y reportar sobre riesgos a la organización. Y lo va a hacer midiendo y reportando efectivamente sobre qué tan bien un sistema o proceso mide contra una "Mejor Práctica" o Política Corporativa definida por la organización. El objetivo secundario es influir sobre los demás en reducir riesgos mediante la concientización, para que los miembros de la organización a su vez puedan reducir los riesgos.

_ Puede ser pensado como una especie de Jurado, pero cuando se expresa debe ser "políticamente correcto", o sea debe ser cuidadoso cuando informa las irregularidades que encuentra. Entonces debe ser por un lado imparcial e integro.

El proceso de los 6 pasos

_ Tenemos diferentes etapas:

1) Planificación de la auditoria: reunión inicial con el que solicita la auditoria o entender las necesidades de la organización. Se realizan algunas actividades preauditoria como:

- Investigación
- Definición inicial del alcance, donde la definición final termina siendo dada por los procedimientos.
- Definición de la estrategia de auditoría, es decir, de qué manera se va a ejecutar.
- Creación de la lista de verificación, o listas.
- Formulación de los procedimientos de auditoría, que nos dan el alcance definitivo.

_ Si no construimos una buena fundación, vamos a perder la batalla en la auditoría. Es decir que esta etapa es la más importante, donde es importante definir adecuadamente el alcance y mantenerlo. Si estamos confiando en la auditoría como una parte integral de nuestra estrategia de defensa en profundidad, o sea que estamos analizando cada uno de los componentes de seguridad de la información de la organización, una planificación pobre de la auditoría significa que hemos perdido antes que alguien nos ataque.

_ La investigación consiste en investigar que políticas tiene la organización, cuáles son las mejores prácticas de la industria y cuáles son los marcos de referencia de auditoría. Una excelente referencia es el Center for Internet Security, que provee documentos con mejores prácticas y procedimientos paso a paso para asegurar ciertos sistemas. Estos documentos pueden ser transformados fácilmente en listas de verificación de auditoría y procedimientos.

_ Para planificar el alcance, la gerencia puede tener una idea, el auditor puede tener experiencia, pero el alcance se va a terminar acordando entre ambas partes basados en los procedimientos. El propósito de la auditoria debe caer dentro del alcance, o sea que sistemas queremos evaluar puntualmente y que queremos obtener de esa evaluación. Tratar de definir el alcance/propósito primero en las etapas previas. Y se intenta no volver más de una vez, con esto decimos que es importante invertir el tiempo necesario al principio, hacer la planificación adecuada y resolver todas las dudas para no tener que volver a definir el alcance. Una vez que se define el proceso de auditoria no hay vuelta atrás.

_ Para saber cuál es la estrategia, dijimos que la Investigación respondió el “Qué” y la Estrategia responde el “Cómo” se va a ejecutar la auditoria.

2) Conferencia de entrada: cuando se comunica a nivel general a todos los involucrados cual va a ser el proceso de la auditoria. Tiene varias preguntas principales:

¿Quién debería venir? es importante tener:

- Representante de la Dirección, para que el resto de la compañía pueda visibilizar la importancia que tiene esta actividad e imponga respeto dentro de la organización.
- Administradores de Sistemas, o de los diferentes componentes que se van a auditar, son los que van a ser afectados directamente por las evaluaciones y recomendaciones.
- Usuarios de Sistemas (algunas veces), que pueden aportar datos adicionales, por ejemplo por medio de entrevistas.
- Representante de Seguridad de Sistemas, que sin duda va a estar afectado o involucrado.

¿Qué debería ser cubierto? es importante que cada una de las partes comprendan:

- Alcance/Objetivos de la Auditoría.
- El Rol del Auditor.
- El Rol de los otros participantes (colaboradores como directivos).
- El Proceso de Auditoría (pasos o agenda).
- El Período de Tiempo, que involucra al trabajo de campo.

¿Qué NO debería hacerse/transmitirse? El auditor debe tener mucho cuidado en esta reunión, ya que no se quiere dar un mensaje de autoritarismo como por ejemplo:

- "Yo estoy a cargo aquí".
- "Yo estoy aquí para ver qué están haciendo mal", para intimidar.
- "Mi informe a la gerencia reflejará qué tan bien es su desempeño", tratar de desprestigiar a un colaborador.

_ Nos orientamos a políticas, procedimientos y controles, y evitar especificar demasiado en personas.

3) Trabajo de campo: cuando se ejecutan cada una de las actividades planificadas. Debería ejecutarse con integridad informando lo que se encuentra por más que no sea de total agrado para la persona o empresa, se debe hacer con profesionalismo, enfocado respetando el alcance que se definió, y razonable que pone en duda lo enfocado porque puede que una irregularidad que escapa levemente del alcance pero lo podemos incluir como un ítem a ser observado con más detalle en el futuro, entonces debemos ser también flexibles para ver más allá de lo que nos propusimos originalmente sin escaparnos del foco que decidimos para el alcance.

_ Entonces lo razonable consiste en informar lo que se encuentra, analizar el por qué se encontró, y explicar por qué es o no es una amenaza.

_ El trabajo de campo no es solamente ejecutado por el auditor sino que es un trabajo en equipo. Es importante que el auditor tenga la capacidad de movilizar sus fuerzas, teniendo buenas habilidades de comunicación y liderazgo, para dirigir apropiadamente a los

colaboradores, y confiar en las fortalezas de otros ya que el auditor no es experto en el 100% de los temas (humildad).

4) Preparación del reporte: no hablamos solamente de documentar lo que se encontró sino que también de ir intercambiando con cada uno de los involucrados, ir pidiendo feedback y estando de acuerdo en lo que se va a informar. El reporte va a tener diferentes secciones, pero es importante que el destinatario final pueda identificar en el resumen ejecutivo cuales fueron los hallazgos principales y puntos más importantes de la auditoria. El reporte debe ser claro y conciso, debe tener un desarrollo lógico en cuanto a organización. Buenas habilidades de lenguaje son obligatorias, como la redacción.

_ El resumen ejecutivo debería ser lo último en ser escrito porque en definitiva intentamos sumarizar cuál fue el propósito de la auditoria, el alcance, los puntos importantes de los hallazgos, y cuáles fueron los Riesgos y el Impacto a la organización acompañada por las recomendaciones.

5) Conferencia de salida: se realiza previamente al informe porque acá hacemos una presentación a la organización y buscamos también obtener feedback y realizar algún ajuste adicional. Esta tiene las mismas preguntas que la de entrada:

¿Quién debería venir? prácticamente los mismos integrantes que en la conferencia de entrada:

- Representante de la Gerencia, que esta para dar soporte.
- Administradores de Sistemas
- Usuarios de Sistemas (a veces)
- Representante de Seguridad de Sistemas, que puede dar sus aportes de acuerdo a sus hallazgos.

¿Qué debería ser cubierto? tenemos:

- Alcance/Objetivos de la Auditoría
- El Rol del Auditor
- El Rol de los otros participantes
- El Proceso de Auditoría
- Los Resultados de la Auditoría, que se encontró, que hicimos bien o mal, que controles se cumplen o fallan, siempre hablando de la organización.

¿Qué NO debería hacerse/transmitirse? Tratar de comunicar en un tono donde acusemos a la organización por no hacer las cosas bien, sino trata de informar que control no se cumple y porque, dar recomendaciones de manera profesional, y no involucrarnos con fallas de comunicación entre empleados.

- “Aquí está lo que ustedes están haciendo mal”
- “Muchos administradores saben mejor que nadie que… “

- Inquietudes de los Administradores hacia la Gerencia.

6) Informe a la gerencia: donde finalmente se hace el informe. El reporte a la gerencia debe ser claro y conciso, es un resumen ejecutivo y se hace en una presentación digital. Se resaltan los puntos más importantes de los hallazgos. Para hacerlo por lo general se prepara una presentación de 60 minutos, esta presentación es llamada por el ejecutivo de más alto nivel, y se agenda una reunión de aproximadamente 2 horas. La reunión se inicia con el ejecutivo para dar autoridad al auditor, se hace una presentación de 30 a 45 minutos, y se da un descanso sin hacer un cierre de la reunión con el fin de que se abra la discusión de los items presentados, donde las opiniones pueden cambiar el curso de la presentación. Luego de la discusión se finaliza la presentación, se hace la convocatoria de análisis/preguntas y se hace un cierre.

CIS Benchmarks

_ Center for Internet Security (CIS), estos documentos se usan principalmente para implementar mejores prácticas en un montón de entornos, en este caso vimos el CIS Amazon Web Services Foundation Benchmarks, pero también tenemos de sistemas operativos, bases de datos, etc. El documento de Amazon cubre los servicios fundamentales, donde dentro de este vemos que cada documento tiene apartados específicos de configuración. Estos documentos tienen una check list para ver detalles.

Auditoría de Bases de Datos

Metodologías para la Auditoría de Bases de Datos

_ Para la auditoria de bases de datos tenemos básicamente dos metodologías:

Metodología tradicional: se basa en básicamente una check list.

Metodología de evaluación de riesgos: es la que propone ISACA que es una organización internacional, que entre sus certificaciones da la de auditor certificado en sistemas. Esta metodología se basa en primero definir los riesgos, luego definir objetivos de control que van a ayudarnos a controlar esos riesgos, y estos objetivos de control se implementan a través de técnicas, donde esas técnicas pueden ser detectivas, correctivas o preventivas. Y además tenemos por un lado pruebas de conformidad, es decir, ver si se cumple o no el control, y también pruebas sustantivas que miden el impacto y van orientadas a ver si hay irregularidades que escapan a esos controles y como impacta en la organización.

Objetivos de control en el ciclo de vida de una base de datos

_ Tenemos las siguientes etapas:

Estudio previo y plan de trabajo: lo primero que realizamos es un análisis de viabilidad sobre alternativas para alcanzar objetivos específicos sobre la base de datos. Entonces tenemos por un lado análisis costo-beneficio, tenemos la posibilidad de seleccionar entre diferentes motores de base de datos, ver como cubren nuestras necesidades. Y podemos tomar algunas decisiones desde no realizar el proyecto, hasta desarrollar vs comprar una base de datos. La decisión seguir adelante o no con el proyecto recae en la dirección, alguien especializado presentara los pros y los contra y facilitara la toma de esta decisión. El auditor debe comprobar que los informes de viabilidad se revisan previamente. En COBIT se enfatiza la importancia de llevar a cabo una gestión de riesgos, esto consiste en la valoración, identificación, medida, plan de acción y aceptación, donde cada una de estas etapas deberían ejecutarse para un adecuado análisis de riesgos. De llevarse a cabo el proyecto, debe establecerse un "Plan director" alineado a los procedimientos de la organización, es decir, si vamos a elegir un motor de base de datos primero deberíamos tener en cuenta cuales son los objetivos de la organización, para que queremos esta base de datos y como la integramos a toda la infraestructura existente. Aprobación de estructura del proyecto y responsables de gestión y control de la base de datos, o sea por un lado estamos autorizando que se implemente y por otro lado queremos designar quien va a ser la persona a cargo de la base de datos, y en el caso de una auditoria quien va a responder por las configuraciones y por los datos que están almacenados. Algunos objetivos de control:

- Responsabilidades para la planificación, organización, plantillas y control de los activos de datos de la organización. La figura de administrador de datos era muy común en el pasado, hoy por hoy tiende a desaparecer y está prácticamente mezclado con el administrador de base de datos.
- Responsabilidad de la administración del entorno de la base de datos, no solamente debe configurar los campos índices, claves primarias, roles, usuarios, etc, sino también ocuparse de donde esta almacenada esa base de datos. Esta responsabilidad es del administrador de base de datos.
- Se define el posicionamiento en el organigrama lo suficientemente alto para asegurar su independencia.

Concepción de la base de datos y selección del motor: consiste en hacer el diseño de la base de datos, es decir que datos vamos a almacenar, con los modelos y las técnicas definidos en la metodología de desarrollo de sistemas de la empresa. Por un lado la empresa ya debería tener una metodología de desarrollo definida que nos ayude a seguir esas prácticas para modelar los datos que van a ir a nuestra base. Esta metodología debería también emplearse para especificar el código fuente, los mecanismos de control, las características de seguridad y las pistas de auditoría a incluir en el sistema, es decir,

nos va a proporcionar más información para ver como configuramos por ejemplo el login en la base de datos, donde almacenamos las contraseñas, etc, para cuidar como se accede externamente a la base de datos. El auditor debe analizar la metodología de diseño con el fin de determinar si es o no aceptable, y luego comprobar su correcta utilización. Como mínimo una metodología de diseño de BD debería contemplar fases de diseño lógico y diseño físico.

_ La definición de la arquitectura de la información, contempla cuatro objetivos de control relativos a:

- Modelo de arquitectura de información y su actualización: acá vemos cual es el flujo de datos entre las aplicaciones y la base de datos que va a soportar estos datos y cuál va a ser el plan cuando necesitemos realizar alguna actualización.
- Datos y diccionario de datos corporativo: esto prácticamente está en desuso, se establecía por lo general un diccionario donde se ponían los campos y se daba una definición de que objetivo tenía cada uno de esos campos, prácticamente no se invierte tiempo en eso hoy en día.
- Esquema de clasificación de datos en cuanto a seguridad: es muy importante que cuando vayamos a diseñar la base de datos, ver si los campos deberían ocultarse de los usuarios, por ejemplo el administrador de la base que no debería ver datos como las contraseñas, etc, por eso es importante que se haga la identificación de los campos, ver si ese dato debería ser accesible por cualquier usuario, por un administrador o incluso por nadie, con el fin de definir que se va a almacenar específicamente en ese campo.
- Niveles de seguridad para cada anterior clasificación de datos: por ejemplo si vamos a almacenar datos sensibles, quizás debamos cifrarlos o aplicar un hash.

_ La selección del motor deberá realizarse utilizando un procedimiento riguroso en el que se consideren:

- Las necesidades de la empresa (debidamente ponderadas).
- Las prestaciones que ofrecen los distintos SGBD (sistemas de gestión de bases de datos) candidatos (puntuados de manera oportuna).

<u>Diseño y carga:</u> acá se llevarán a cabo los diseños lógico y físico de la base de datos, por lo que el auditor determinará si la definición de los datos contempla además de su estructura, las asociaciones y restricciones oportunas, así como las especificaciones de almacenamiento de datos y cuestiones relativas a la seguridad. Entonces acá se llevan los diseños que hicimos a la implementación, tenemos que ver si se contemplan las restricciones oportunas o sea se hace una análisis más allá del primero que se defino, vemos si contamos con las restricciones adecuadas para esta implementación y si se hizo algún cálculo de capacidad que nos permita ver si vamos a necesitar un servidor más grande, que disco va a tener, y otras cuestiones relativas a la seguridad. Este diseño debe

estar aprobado por la dirección del departamento de informática, los usuarios e incluso la alta dirección, según corresponda si son cuestiones técnicas en las cuales la alta dirección no interviene serán solamente aprobadas por el área, de lo contrario si hay datos o información sensible quizás la dirección quiera contar con el asesoramiento de una tercera parte independiente y en base a eso aprobar este plan. Una vez diseñada la BD, se procederá a su carga. Las migraciones o conversiones de sistemas (por ejemplo pasar de una base de datos vieja a una nueva) deberán estar claramente planificadas para evitar pérdida de información y la transmisión al nuevo sistema de datos erróneos, o sea deberíamos tener de alguna manera pensado un mecanismo para validar que lo que intentamos migrar se está migrando efectivamente. También se deberán realizar pruebas en paralelo, mediante entornos de prueba.

_ En lo que respecta a la entrada manual de datos (data entry), hay que establecer un conjunto de controles que aseguren la integridad de los mismos. Las declaraciones escritas de procedimientos de la organización referentes a la entrega de datos a ser procesados deben asegurar que los datos se autorizan, recopilan, preparan, transmiten, y se comprueba su integridad de forma apropiada. Para el tratamiento de datos de entrada erróneos, deben cuidarse con atención los procedimientos de reintroducción de forma que no disminuyan los controles; lo ideal es que los datos se validen y corrijan tan cerca del punto de origen como sea posible, es decir, si tenemos una migración que pasa por diferentes capas o puntos hasta llegar a nuestra base de datos, tratar de implementar controles a lo largo de ese camino para tratar de evitar que se pierda información o que llegue de forma incorrecta.

Uso y mantenimiento: el sistema se pondrá en marcha, mediante las correspondientes autorizaciones y siguiendo los procedimientos establecidos, como dijimos puede venir esta autorización solamente del departamento de informática, o de la dirección o de algún grupo de usuarios que hace uso de la base de datos. Se debe comprobar que los datos se tratan de forma congruente y exacta y que el contenido de los sistemas sólo se modifica mediante la autorización adecuada, es decir, que establezcamos los mecanismos para restringir que si se hacen modificaciones a la base de datos ocurran en una aplicación o servicio para que veamos que se haga de esa forma y no por ejemplo por un IP fuera de la organización. COBIT especifica objetivos de control para la gestión de datos. El auditor debería llevar a cabo también una auditoría sobre el rendimiento de la BD, comprobando si se lleva a cabo un proceso de ajuste (tuning) y optimización adecuados.

Revisión post-implementación: se deberá efectuar una revisión post-implementación con el fin de evaluar si:

- Se han conseguido los resultados esperados, aquellos que se hayan planificado originalmente.
- Se satisfacen las necesidades de los usuarios, quienes van a beneficiarse con este motor de base de datos.

- Los costos y beneficios coinciden con los previstos.

Otros procesos auxiliares: se deberá controlar la formación que precisan tanto usuarios informáticos como no informáticos, ya que la formación es una de las claves para minimizar el riesgo en la implementación de la base de datos, es decir si vamos a incorporar un motor nuevo desconocido como parte o evaluación de este servicio, deberíamos ver que tan bien escrita esta la documentación, si tenemos entrenamiento disponible, en principio apuntando a nuestra área de informática es probable que usuarios finales quieran hacer consultas personalizadas a la base de datos que no estén contempladas en nuestro sistema y que necesiten también esa capacitación. Usuarios poco formados constituyen uno de los peligros más importantes de un sistema. Esta formación no debería limitarse al área de las bases de datos, sino que tendría que ser complementada con formación relativa a los conceptos de control y seguridad, como dijimos el entorno de la base de datos es importante. Además el auditor tendrá que revisar la documentación que se produce a lo largo de todo el proceso, desde que planificamos hasta que implementamos, para verificar si es suficiente y si se ajusta a los estándares establecidos por la metodología adoptada en la empresa. Lo ideal sería que en la propia empresa existiera un grupo de calidad que se encargara, entre otras cosas, de asegurar la calidad de los diseños de bases de datos, y este grupo por lo general se pretende que sea independiente con el fin de que los controles que hagan no estén de alguna manera influenciados por cómo fue realizada la implementación.

Auditoría y control interno en un entorno de bases de datos

SGBD y su entorno

_ El entorno es todo lo que rodea a la base de datos, o en que entorno esta hosteada esta base de datos. Tenemos diferentes componentes:

Software de auditoría: aquellas aplicaciones que ayudan específicamente a auditar tanto conexiones de usuarios como configuraciones.

Sistema de monitorización y ajuste (tuning): básicamente que tenemos para monitorizar el rendimiento, las conexiones de los usuarios, consultas, modificaciones, que tan bien está funcionando la performance y que podemos modificar para que esa performance mejore.

Auditoría del sistema operativo (SO): tenemos que ver cuáles son los mecanismos de seguridad implementados en ese sistema operativo donde tengamos cargado nuestro motor de base de datos.

Protocolos y sistemas distribuidos: para el caso de que tengamos nuestra base de datos replicada, y se pueda consultar información en ambas y controlar el tránsito.

Paquete de seguridad: orientados específicamente al motor de base de datos o bien al sistema operativo o lo que este hosteando la base de datos en sí.

Técnicas para control de bases de datos en entornos complejos

_ Existen muchos elementos del entorno del SGBD que influyen en la confidencialidad e integridad de los datos, algunos de ellos interdependientes. En cuanto a Seguridad, se deben fijar claramente las responsabilidades sobre los diferentes componentes, utilizar informes de excepción efectivos que permitan monitorear los controles, establecer procedimientos adecuados, implementar una gestión rigurosa de la configuración del sistema, etc, es decir, tener en cuenta todos aquellos mecanismos para primero ejecutar los controles, si no se cumplen saber cuál es el flujo de trabajo que tenemos para las remediaciones y a su vez cada cuanto revisamos esa configuración del sistema. El auditor puede emplear dos técnicas de control:

Matrices de control: sirven para identificar los conjuntos de datos del SI (sistema de información) junto con los controles de confidencialidad o integridad implementados sobre los mismos, es decir, que objeto vamos a almacenar nuestra base de datos, que campo o que atributos tienen esos objetos, y como esos campos o atributos se ven afectados en cuestiones de confidencialidad o integridad. Los controles se clasifican en:

- Detectivos: para darnos cuenta cuando un incidente o algo está sucediendo.
- Preventivos: para evitar que eso suceda.
- Correctivos: para restaurar la situación al estado esperado.

Análisis de los caminos de acceso: con esta técnica se documentan el flujo, almacenamiento y procesamiento de los datos, identificando los componentes del sistema que atraviesan y los controles asociados, es decir, por donde viaja la información, como se procesa. El auditor puede identificar las debilidades que exponen los datos a riesgos de integridad, confidencialidad, y disponibilidad, las distintas interfaces entre componentes y la compleción de los controles, es decir, que tan efectivos son en lo que queremos medir.

Consejos para auditoría y evaluación de bases de datos

No Alejarse del Objetivo

- Hacer que la base de datos sea un componente crítico de cualquier auditoría IT, es decir, que no solamente estemos analizando la red y los SO, sino que las base de datos es uno de los objetivos más importantes para un atacante.
- Incluir las bases de datos como parte de auditorías frecuentes de aplicaciones.

- No dejar que la falta de conocimiento de bases de datos sea un problema. Si no tenemos personal interno capacitado, debemos contratar un consultor o tener documentación y entrenamiento disponible para enfrentar esa falta de conocimiento.
- No dejar que el DBA sea una piedra en el camino. Si vamos a auditar una base de datos debemos hacerle entender a la persona responsable de ese motor que no queremos interferir con su trabajo sino simplemente buscar si las mejores prácticas están implementadas, si los datos están protegidos, y si el rendimiento de la base de datos es el correcto.

Planificar en consecuencia

- Entender el alcance, lo mencionamos a nivel de auditoría general, en base de datos es lo mismo. O sea que tipos de bases de datos, versiones y sistemas operativos vamos a utilizar, qué aplicaciones soporta la base de datos, como fluyen los datos desde y hacia la base de datos.
- Asociarse con el DBA (administrador de base de datos), para hacer que se sientan cómodos con el proceso, y asegurarles que los procesos no van a interferir con el rendimiento de la base de datos.
- Asignar los recursos apropiados, o sea cuales son los controles a auditar, cual es nuestra línea de base (CIS Benchmarks, Programas de ISACA, ITIL, otros). Si lo podemos hacer con personal interno de la empresa, hay un experto interno, o contratamos a otro experto, y también si podemos usar una herramienta que nos facilite esta tarea.

Automatizar lo que se pueda

- Por lo general una auditoria puede ser muy leve y alguien puede simplemente hacer preguntas. Hacer entrevistas solamente basadas en procesos no producirán resultados significativos, ya que simplemente nos van a dar respuestas y no vamos a poder medir la realidad.
- Ver si tenemos la habilidad de proveer scripts independientes al DBA para que nos de la información correcta para revisar, y ver si tenemos la habilidad de revisar los datos en bruto arrojados por esas consultas.
- Si elegimos una herramienta o si invertimos en una que sea automatizada y además elegir la mejor posible en el caso de invertir.

Ir más allá de la auditoría tradicional

_ La entrevista a lo mejor no es suficiente porque no nos deja meternos en la base de datos o ver como el DBA extra la información que necesitamos. La auditoría tradicional nos puede hacer perder de algo crítico. Las bases de datos tienen configuraciones

complejas y pueden variar, hay tantas configuraciones como motores. Entonces para hacer la comparativa:

Auditoria:

- Esta dirigida por control de procesos, vamos específicamente a controlar procesos.
- Es más general, no se mete en tanto nivel de detalle.
- Es un conjunto de controles más pequeño a revisar, porque tratamos de orientarlo a riesgos entonces por ahí estamos trabajando con un subconjunto de esos controles.
- Se hacen movimientos sólo para cumplir. Por ejemplo cuando una empresa quiere certificar cierto estándar no se preocupa por la seguridad de la información sino por pasar esa auditoria.

Evaluación de la seguridad:

- Es más específica a la tecnología. Acá intentamos comprender que nos ofrece la tecnología desde el punto de vista de seguridad.
- Necesita responder: ¿estamos seguros? y ¿cuál es el riesgo?
- Y en base a las preguntas saber cómo afecta esto la confidencialidad, integridad y disponibilidad.

_ Para llevar esto a un ejemplo seria: si la auditoria está en check, pasamos la misma de manera exitosa, y en el caso de la evaluación de seguridad vamos más allá de cumplir con ese check e intentamos ver si realmente estamos protegidos.

_ Evaluación de seguridad no es lo mismo que evaluación de riesgos.

Auditoría	Evaluación de Seguridad
✓ Necesito examinar si la longitud de la contraseña está establecida en 8 caracteres	✓ Necesito examinar si los intentos de inicio de sesión fallidos son controlados y cada cuantos se bloquea la cuenta ✓ Necesito examinar si la complejidad de contraseña está habilitada ✓ Necesito examinar si la longitud de la contraseña está establecida en 8 caracteres

Informar métricas significativas

_ Si estamos hablando a nivel gerencial queremos dar un informe ejecutivo con una lista de observaciones y detalles que sean interesantes. Una forma útil de compilar esta información es mediante una tabla marcando items como severidad, área de prueba

(interna, externa), breve descripción, sistemas y aplicaciones afectados, riesgo técnico, pasos para resolver la vulnerabilidad, nivel estimado de esfuerzo para resolver la vulnerabilidad (alto/medio/bajo).

#	Severidad	Área de Prueba (Interna, Externa)	Breve Descripción	Sistemas y Aplicaciones Afectados	Riesgo Técnico	Pasos para Resolver la Vulnerabilidad	Nivel Estimado de Esfuerzo para Resolver la Vulnerabilidad (Alto/Medio/Bajo)
...	...	...	...	...	...	...	...

Auditoría de MySQL basada en Benchmarks CIS

Auditoría de MySQL (controles a nivel general)

1) Configuración a nivel del sistema operativo: la base de datos debe estar en una partición diferente a la del sistema, con el objetivo de que si el sistema sufre en si algún inconveniente, nuestros datos queden a salvo.

1.1_ Cuando corremos el query se nos informa donde esta almacenada la base de datos y en este caso esta almacenada en la misma partición que el sistema operativo con lo cual no estría cumpliendo con ese control, o sea se descubre el directorio de datos ejecutando:

```
MariaDB [(none)]> show variables where variable_name = 'datadir';
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| datadir       | /var/lib/mysql/ |
+---------------+-----------------+
```

_ La base de datos debe estar en una partición diferente a la del sistema. Luego en el sistema operativo, sobre el directorio de datos obtenido, se ejecuta:

```
[root /]$ df -h /var/lib/mysql/
Filesystem      Size  Used Avail Use% Mounted on
overlay         801G  121G  640G  16% /
```

_ El comando df –h nos muestra que estamos montados en el root ya que está en la misma partición que el sistema operativo. Se valida que la partición no sea una del sistema.

1.2_ Luego deberíamos utilizar una cuenta dedicada con mínimos privilegios para el servicio MySQL. Si tenemos un usuario root que tiene acceso directo a la base de datos,

no tenemos más inconvenientes para acceder a los datos y así protegemos la confidencialidad. Ejecutar el siguiente comando para verificar, donde vemos que ejecutamos el proceso de mysql:

```
[root /]$ ps -ef | egrep "^mysql.*$"
mysql        183      37  0 10:26 pts/0    00:00:01 /usr/sbin/mysqld --basedir=/usr
--datadir=/var/lib/mysql --plugin-dir=/usr/lib/x86_64-linux-gnu/mariadb18/plugin --u
ser=mysql --skip-log-error --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/run/m
ysqld/mysqld.sock --port=3306
```

_ Si no se retorna ninguna línea, es un hallazgo, en este caso si tenemos retorno entonces vemos que si se está usando la cuenta MySQL para lo que pretendemos evaluar. Entonces se asume que el usuario corriendo MySQL es mysql.

1.3_ Deshabilitar el historial de comandos de MySQL. Ejecutar el siguiente comando para verificar, donde en este caso nos fijamos en el home si encontramos el mysql history, y también en root donde si lo tenemos:

```
[root /]$ find /home -name ".mysql_history"
[root /]$ find /root -name ".mysql_history"
/root/.mysql_history
[root /]$ ls -las /root/.mysql_history
4 -rw------- 1 root root 48 Mar 30 10:37 /root/.mysql_history
[root /]$
```

_ Para cada archivo retornado, determinar si el archivo está enlazado simbólicamente en /dev/null, lo que quiere decir que ese histórico se está enviando a la basura y que no está quedando almacenados. En este caso cada uno de los comandos que vamos corriendo a la base estarían quedando en este histórico con lo cual no se estaría cumpliendo.

1.4_ Deshabilitar el inicio de sesión interactivo. Ejecutar el siguiente comando para verificar:

```
[root /]$ getent passwd mysql | egrep "^.*[\/bin\/false|\/sbin\/nologin]$"
mysql:x:101:101:MySQL Server,,,:/nonexistent:/bin/false
```

_ No obtener ningún resultado implica un hallazgo. O sea, lo que buscamos acá específicamente es que la cuenta mysql no puede iniciar sesión desde afuera del servidor, entonces lo que buscamos es que tenga un false como es en este caso.

1.5_ Verificar que 'MYSQL_PWD' no está establecida en los perfiles de usuario. Ejecutar los siguientes comandos para verificar:

```
[root /]$ grep MYSQL_PWD /home/*/.{bashrc,profile,bash_profile}
grep: /home/*/.bashrc: No such file or directory
grep: /home/*/.profile: No such file or directory
grep: /home/*/.bash_profile: No such file or directory
[root /]$ grep MYSQL_PWD /root/.{bashrc,profile,bash_profile}
grep: /root/.bash_profile: No such file or directory
```

_ Buscamos específicamente el parámetro “MYSQL_PWD”, que no esté como una variable de entorno en cada uno de esos archivos que son los de sesión de usuario. Y decimos que no se deberían obtener resultados de estos comandos, y en este caso vemos que no se encontró ningún resultado.

2) Instalación y planificación:

2.1_Copias de Seguridad y Recuperación ante Desastres: buscamos específicamente ver la política de copia de seguridad establecida, si tenemos algún mecanismo para hacer backup de nuestra base de datos. Lo que propone este Benchmark es verificar si existen copias programadas con crontab que es un servicio que programa tareas, asegurarnos que esa base de datos se esté copiando a algún lugar con cierta frecuencia:

```
[root /]$ crontab -l
no crontab for root
```

_ En este caso no había ningún proceso de crontab para root. Entonces Si no hay resultados, no hay tareas programadas de backup.

3) Permisos en sistemas de archivos:

3.1_ Tenemos un directorio de datos y hay que asegurarse que ‘datadir’ tiene permisos apropiados. Primero se descubre el directorio de datos ejecutando:

```
MariaDB [(none)]> show variables where variable_name = 'datadir';
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| datadir       | /var/lib/mysql/ |
+---------------+-----------------+
1 row in set (0.01 sec)
```

_ Y luego vemos en el sistema operativo, sobre el directorio de datos obtenido, cuáles son los permisos que tiene:

```
[root /]$ ls -l /var/lib/mysql/.. | grep mysql
drwxr-xr-x 1 mysql mysql 4096 Mar 30 10:26 mysql
```

_ Y verificamos que los permisos correspondan al usuario mysql, que en este caso responde.

3.2_ Asegurar que los archivos ‘log_bin_basename’ (archivos de log binarios) tienen permisos apropiados. De nuevo se descubre el directorio de datos ejecutando:

```
MariaDB [(none)]> show variables like 'log_bin_basename';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| log_bin_basename |       |
+------------------+-------+
1 row in set (0.00 sec)
```

_ En este caso no obtuvimos ningún resultado pero de existir archivos, se debe verificar que los permisos de cada uno sean 660 (rw,rw,-) para mysql:mysql, donde rw es read-write.

3.3_ Asegurar que los archivos 'log_error' (archivos de log de errores) tienen permisos apropiados. Se descubre el directorio de datos ejecutando:

```
MariaDB [(none)]> show variables like 'log_error';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_error     |       |
+---------------+-------+
1 row in set (0.00 sec)
```

_ De nuevo esta variable no nos arroja ningún log con lo cual no podemos hacer la validación en este caso pero si existiera, se debe verificar que los permisos de cada uno sean 660 (rw,rw,-) para mysql:mysql.

3.4_ Asegurar que los archivos 'slow_query_log' (queries que toman +10" para correr) tienen permisos apropiados. Se descubre el directorio de datos ejecutando:

```
MariaDB [(none)]> show variables like 'slow_query_log_file';
+---------------------+------------------------+
| Variable_name       | Value                  |
+---------------------+------------------------+
| slow_query_log_file | 40034d206d78-slow.log |
+---------------------+------------------------+
1 row in set (0.01 sec)
```

_ Vemos el archivo que debería estar almacenando esos resultados. Cuando hicimos la verificación el archivo era inexistente y en el caso de existir archivos, se debe verificar que los permisos de cada uno sean 660 (rw,rw,-) para mysql:mysql.

3.5_ Asegurar que los archivos 'relay_log_basename' (set de logs creados por un esclavo durante la replicación) tienen permisos apropiados. Se descubre el directorio de datos ejecutando:

```
MariaDB [(none)]> show variables like 'relay_log_basename';
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| relay_log_basename |       |
+--------------------+-------+
1 row in set (0.00 sec)
```

_ Acá tampoco tenemos archivos de logs para este control. De existir archivos, se debe verificar que los permisos de cada uno sean 660 (rw,rw,-) para mysql:mysql.

3.6_ Asegurar que el archivo 'general_log_file' (log general de lo que es realizado por mysqld), que básicamente registra cada una de las acciones realizadas por el servicio tengan los permisos apropiados. Se descubre el directorio de datos ejecutando:

```
MariaDB [(none)]> show variables like 'general_log_file';
+------------------+--------------------+
| Variable_name    | Value              |
+------------------+--------------------+
| general_log_file | 40034d206d78.log |
+------------------+--------------------+
1 row in set (0.00 sec)
```

_ De nuevo en este caso no estamos registrando nada. De existir archivos, se debe verificar que los permisos de cada uno sean 660 (rw,rw,-) para mysql:mysql.

4) Controles generales:

4.1_ Acá aseguramos que las últimas actualizaciones de seguridad fueron aplicadas, o sea que estamos usando la última versión porque si estamos usando una versión vieja podemos estar corriendo riesgos de que esa versión tenga alguna vulnerabilidad conocida y si el atacante la identifica le puede tomar minutos. Y podemos ver si la versión es la última comparando con el sitio oficial. Se descubre la versión ejecutando:

```
MariaDB [(none)]> show variables where variable_name like 'version';
+---------------+--------------------------+
| Variable_name | Value                    |
+---------------+--------------------------+
| version       | 10.1.26-MariaDB-0+deb9u1 |
+---------------+--------------------------+
1 row in set (0.00 sec)
```

4.2_ Asegurar que la base de datos 'test' no está instalada. Se descubre el directorio de datos ejecutando:

```
MariaDB [(none)]> show databases like 'test';
Empty set (0.02 sec)
```

_ Entonces acá vemos si existe esa base de datos, y en este caso no existe. El fin de esto es, podemos instalar un motor de base de datos que tenga bases de datos de ejemplos que tenga esa base de datos con algún rol con permisos excesivos y que nos permitan acceder de manera indirecta a los datos que nos interesa proteger, entonces tratamos de que cuando instalamos un motor de eliminar todo lo que sea accesorio y que no sea el

objetivo principal como lo son las bases de datos de ejemplo. Si la base de datos existiera, deberá ser removida.

4.3_Asegurar que 'allow-suspicious-udfs' (adjuntar funciones de librería compartidas) está establecido en 'FALSE', es decir, que estén deshabilitadas. Se descubre ejecutando el comando:

```
[root /]$ ps -ef | grep mysqld
root          37       1  0 10:26 pts/0    00:00:00 /bin/bash /usr/bin/mysqld_safe
mysql        183      37  0 10:26 pts/0    00:00:32 /usr/sbin/mysqld --basedir=/usr --datadir
=/var/lib/mysql --plugin-dir=/usr/lib/x86_64-linux-gnu/mariadb18/plugin --user=mysql --skip-l
og-error --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/run/mysqld/mysqld.sock --port=33
06
```

_ Asegurar que —allow-suspicious-udfs no está especificado en el comando de inicio mysqld que es el servicio básicamente.

4.4_Asegurar que 'local_infile' (determina si los archivos ubicados en el cliente MySQL se puede cargar o seleccionar a través de LOAD DATA INFILE o SELECT local_file) está deshabilitado. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> show variables where variable_name = 'local_infile';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| local_infile  | ON    |
+---------------+-------+
1 row in set (0.00 sec)
```

_ Si el resultado fuera "ON", deberá cambiarse local-infile en el archivo de configuración de MySQL. Vamos a encontrar que estos archivos de configuración vienen con prácticamente nada establecido entonces cada uno de estos chequeos deberían ir agregándose a ese archivo de configuración.

4.5_ Asegurar que mysqld no se inicia con '—skip-grant-tables' (esta opción inicia mysql sin usar el sistema de privilegios), o sea que no toma en cuenta las configuraciones de roles que establecimos. Se descubre editando el archivo de configuración de mysql y validando la configuración:

```
# vim /etc/mysql/my.cnf
#
[client-server]

# Import all .cnf files from configuration directory
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mariadb.conf.d/
```

_ Buscar "skip-grant-tables" y asegurarse que esté establecido en "FALSE". En este caso no encontramos la configuración por lo tanto está en FALSE.

5) Permisos de MySQL:

5.1_ Asegurar que solamente los usuarios administrativos tienen acceso completo a la base de datos. Esto ira de acuerdo a como la organización haya definido ese esquema de accesos. Si no queremos que el root acceda a los datos esto sería totalmente incorrecto pero en este caso si se descubre ejecutando las consultas, nos dice cuáles son los usuarios que pueden acceder y va validando diferentes privilegios:

```
MariaDB [(none)]> select user, host from mysql.user where (select_priv = 'y') or (insert_priv
 = 'y') or (update_priv = 'y') or (delete_priv = 'y') or (create_priv = 'y') or (drop_priv =
'y');
+------+-----------+
| user | host      |
+------+-----------+
| root | localhost |
+------+-----------+
1 row in set (0.00 sec)

MariaDB [(none)]> select user, host from mysql.db where db = 'mysql' and ((select_priv = 'y')
 or (insert_priv = 'y') or (update_priv = 'y') or (delete_priv = 'y') or (create_priv = 'y')
or (drop_priv = 'y'));
Empty set (0.00 sec)
```

_ Todos los usuarios retornados deben ser usuarios administrativos. En este caso es root pero puede darse el caso que tenemos un administrador del SO que no debería tener influencia en la base de datos y queremos que este root solamente o el usuario con DBA debería tener ese acceso.

5.2_Asegurar que 'file_priv' no está establecida en 'Y' para usuarios no administrativos. Es importante seguir el sentido común a la hora de leer estos documentos. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> select user, host from mysql.user where file_priv = 'y';
+------+-----------+
| user | host      |
+------+-----------+
| root | localhost |
+------+-----------+
1 row in set (0.00 sec)
```

_ Todos los usuarios retornados deben ser usuarios administrativos para tener privilegios sobre los archivos.

5.3_Asegurar que 'process_priv' no está establecida en 'Y' para usuarios no administrativos. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> select user, host from mysql.user where process_priv = 'y';
+------+-----------+
| user | host      |
+------+-----------+
| root | localhost |
+------+-----------+
1 row in set (0.01 sec)
```

_ Lo mismo pasa con los procesos, y vemos que todos los usuarios retornados deben ser usuarios administrativos.

5.4_Asegurar que 'super_priv' no está establecida en 'Y' para usuarios no administrativos. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> select user, host from mysql.user where super_priv = 'y';
+------+-----------+
| user | host      |
+------+-----------+
| root | localhost |
+------+-----------+
1 row in set (0.00 sec)
```

_ En este caso root tiene un super privilegio para la base de datos. Todos los usuarios retornados deben ser usuarios administrativos.

6) Auditoria y registros:

6.1_ Sería interesante más que nada a nivel performance asegurar que 'log_error' no está vacío tanto para performance de la aplicación que este accediendo a la base de datos como está en sí. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> show variables like 'log_error';
+----------------+-------+
| Variable_name | Value |
+----------------+-------+
| log_error     |       |
+----------------+-------+
1 row in set (0.00 sec)
```

_ Entonces deberíamos asegurarnos de que este configurado es log de errores. En este caso no lo está. Asegurar que el valor retornado no está vacío.

6.2_ Asegurar que los archivos de registro son almacenados en una partición diferente a la del sistema. Si pasa algo con nuestro SO, tratar de que toda la información que nos sirva para auditoria de la base de datos o el almacenamiento en si estén en una partición diferente. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> select @@global.log_bin_basename;
+----------------------------+
| @@global.log_bin_basename |
+----------------------------+
| NULL                       |
+----------------------------+
1 row in set (0.00 sec)
```

_ En este caso no tenemos nada . Y debemos asegurar que el valor retornado no indique root (“/”), /var o /usr.

6.3_ Asegurar que ‘log_warnings’ está establecida en ‘2’. Nos interesa considerar alguna advertencia para hacer cambios en configuraciones o tener en cuenta alguna situación en particular. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> show global variables like 'log_warnings';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_warnings  | 1     |
+---------------+-------+
1 row in set (0.00 sec)
```

_ En este caso el nivel que nos pide el Benchmark que tengamos es 2 y nos devuelve 1 con lo cual deberíamos cambiar esa configuración para que quede establecida en 2. Esto es cuanto nivel de detalle nos ofrece esa advertencia.

6.4_ Asegurar el Registro de Auditoría está Habilitado. No tenemos registro en este ejemplo.

7) Autenticación:

7.1_ Asegurar que ‘old_passwords’ está establecida en ‘1’ u ‘ON’. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> show variables where variable_name = 'old_passwords';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| old_passwords | OFF   |
+---------------+-------+
1 row in set (0.00 sec)
```

_ En este caso no está validando eso ya que está en OFF, y debemos Asegurar que el valor no sea igual a 1 u “ON”.

7.2_Asegurar que ‘secure_auth’ está establecida en ‘ON’. Esto hace referencia a usar cifrado en tránsito. Se descubre ejecutando la consulta:

```
MariaDB [(none)]> show variables where variable_name = 'secure_auth';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| secure_auth   | ON    |
+---------------+-------+
1 row in set (0.00 sec)
```

_ Asegurar que el valor sea igual a “ON”. En este caso lo está.

7.3_Asegurar que las contraseñas no se almacenan en la Configuración Global . Se descubre abriendo el archivo de configuración my.cnf, yendo a la sección [client] y asegurándose que no se está especificando password:

```
#
[client-server]

# Import all .cnf files from configuration directory
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mariadb.conf.d/
```

_ En este caso no tenemos ninguna contraseña almacenada en el archivo lo cual es positivo.

7.4_Asegurar que ‘sql_mode’ contiene ‘NO_AUTO_CREATE_USER’ para que no se creen automáticamente los usuarios. Se descubre ejecutando las siguientes consultas:

```
MariaDB [(none)]> select @@global.sql_mode;
+--------------------------------------------+
| @@global.sql_mode                          |
+--------------------------------------------+
| NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION |
+--------------------------------------------+
1 row in set (0.00 sec)

MariaDB [(none)]> select @@session.sql_mode;
+--------------------------------------------+
| @@session.sql_mode                         |
+--------------------------------------------+
| NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION |
+--------------------------------------------+
1 row in set (0.00 sec)
```

_ Vemos que esta ese parámetro. Debemos asegurar que cada resultado contiene ‘NO_AUTO_CREATE_USER’.

7.5_Asegurar todas las cuentas MySQL tienen contraseña especificada. Se descubre ejecutando la siguiente consulta:

```
MariaDB [(none)]> select user, host from mysql.user where (plugin in('mysql_native_password',
 'mysql_old_password','') and (length(password) = 0 or password is null)) or (plugin='sha256_
password' and length(authentication_string) = 0);
Empty set (0.00 sec)
```

_ Hay muchas bases de datos que vienen por defecto sin autentificación lo cual es grave. En este caso no obtuvimos ningún resultado por lo que todas las cuentas si tenían una contraseña especificada . O sea no debería ser retornada ninguna cuenta si todas las cuentas tienen una contraseña especificada.

7.6_Asegurar que la política de contraseñas está especificada, lo que hablamos de complejidad. Se descubre ejecutando la siguiente consulta:

```
MariaDB [(none)]> show variables like 'validate_password%';
Empty set (0.01 sec)
```

_ Podemos ver si existen variables que tienen el parámetro validate_passsword donde se especifican los diferentes parámetros que deberían cumplirse. En este caso no nos devolvió nada con lo cual no tenemos:

- validate_password_length (14+)
- validate_password_mixed_case_count (1+)
- validate_password_number_count (1+)
- validate_password_special_char_count (1+)
- validate_password_policy (MEDIUM/STRONG)

_ Las siguientes líneas deberían estar presentes en la configuración global. En este caso no están ninguno de estos dos parámetros, por lo tanto deberíamos agregarlos:

```
plugin-load=validate_password.so

validate-password=FORCE_PLUS_PERMANENT
```

```
#
[client-server]

# Import all .cnf files from configuration directory
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mariadb.conf.d/
```

8) Red:

8.1_Asegurar que 'have_ssl' está establecida en 'YES', es decir, que estamos utilizando certificados para todas las comunicaciones en las bases de datos. En este caso vemos todas las conexiones que ofrezca el motor usen certificados. Se descubre ejecutando las siguientes consultas:

```
MariaDB [(none)]> show variables where variable_name = 'have_ssl';
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_ssl      | DISABLED |
+---------------+----------+
1 row in set (0.00 sec)
```

_ En este caso esta deshabilitado. Acá debemos asegurar que el valor retornado es "YES" pero probablemente sea "ENABLE".

8.2_ Asegurar que ‘ssl_type’ está establecido en ‘ANY’, ‘X509’ o ‘SPECIFIED’ para todos los usuarios remotos, es decir, que mecanismos de validación vamos a tener con respecto a ese SSL.

```
MariaDB [(none)]> select user, host, ssl_type from mysql.user where not host in ('::1', '127.
0.0.1', 'localhost');
Empty set (0.01 sec)
```

_ En este caso no tenemos ningún resultado. Deberíamos tener uno de estos o sea asegurar que el valor retornado de ssl_type para cada usuario es ‘ANY’, ‘X509’ o ‘SPECIFIED’.

9) Replicación:

_ Involucra controles para instancias con alta disponibilidad como:

- Asegurar que el tráfico de replicación se realiza por medios seguros (VPN, SSL/TLS, SSH).
- Donde se almacenan los registros maestros y esclavos, es decir, si tenemos una arquitectura definida y tenemos alguna protección adicional para el maestro y cuantos esclavos definimos para tener alta disponibilidad.
- Forzar que en el contexto esclavo se verifique el certificado SSL maestro, es decir, que no permitamos que alguien nos haga un ataque en el medio intercambiando certificados y usando uno invalido y aun así pueda acceder a nuestra información.
- Asegurar que los usuarios de replicación no tienen privilegios excesivos. Digamos tenemos nuestra base de datos maestra, las instancias de replicación, deberían tener copias de la información y no modificar la información de la base de datos principal si ese fuera el esquema.
- Asegurar que los Usuarios de Replicación no usan comodines para los nombres de equipo (‘’@’%’), es decir, restringir todo lo que se pueda el acceso a los equipos, ser muy específicos con cuales son los servidores esclavos y no dar lugar a que desde cualquier parte del mundo alguien se conecte a la base maestra y pueda tomar información o modificarla en el caso de que tuvieran privilegios para hacerlo.

Auditoría de Sistemas Unix

Objetivo del apartado: definir estándares para copos de nieve, vamos a dar un puntapié inicial en lo que es scripting básico, y vamos a describir algunos items importantes en la auditoria de sistemas basados en Unix.

Enfoque

_ Nos vamos a basar en buenos controles de configuración, que esto nos va a ayudar por un lado a tener sistemas seguros, y esto también nos brinda facilidad para auditar por lo que toda la investigación está hecha por nosotros, y partir de una certificación de seguridad a una auditoría es mucho más simple porque ya tenemos algo previo y no empezamos de cero.

Especialmente con Unix

Copos de nieve: hablamos de copos de nieve, porque cada servidor que tenemos corriendo en los entornos por lo general es único (cada copo de nieve es único, por ende los sistemas únicos son como copos de nieve), tienen un administrador dedicado primario y secundario que cubre esta persona, y tiene configuraciones o aplicaciones muy específicas, pero por lo general los sistemas basados en Unix tienden a estar descentralizados en cuanto a su configuración. Entonces decimos que los sistemas Unix tienden a ser copos de nieve porque habitualmente son servidores y tienen poca gestión y administración descentralizada.

_ Los copos de nieve son todos diferentes, caen en diferentes lugares, horizontales, verticales, entonces comparamos esto a los sistemas Unix porque por lo general no vamos a encontrar un sistema que este configurado igual que otro, porque por lo general corren servicios diferentes, tienen administradores diferentes, con toques y optimizaciones diferentes. Esto no aplica a todos los casos, pero puede suceder en una pyme por ejemplo donde todos los sistemas pueden ser diferentes.

Creación de listas de control

_ Debemos identificar fuentes de buenas prácticas:

- Podríamos basarnos en los Benchmarks que ya están listos por nosotros.
- Podríamos establecer mecanismos de certificación interna.
- Crear formularios que nos ayuden a esta tarea.
- Tener listas de control de seguridad para temas puntuales, y no necesariamente recorrer todo el benchmark cada vez que vamos a auditar un sistema.
- Tener políticas que de alguna manera establezcan las bases para cada uno de los controles o de las acciones que vamos a realizar.

_ Debemos identificar objetivos (apuntados a reducir riesgos) y buscar controles (que cumplan esos objetivos):

- Considerar el uso de una especie de “Permiso de Edificación”, es decir, que cuando un administrador crea un servidor, él va a documentar todo lo que fue configurando en el servidor, nosotros revisamos esas configuraciones y si desde el punto de seguridad están correctas, le damos el ok para que finalmente pueda lanzar ese servidor a producción. Desde el punto de vista de auditoría, es mucho más fácil cuando uno tiene que auditar agarrar ese permiso de edificación y validar con lo que está actualmente, y eso debería coincidir.

Scripting Básico

Scripting

_ El Scripting por ahí no es tan difícil como uno piensa. Se llama Script porque en español quiere decir “guion” y es lo que se suele usar en cine (instrucciones que un actor debe seguir para la actuación), y ocurre lo mismo para esta secuencia de comandos, tenemos un montón de líneas de código que se van a ir ejecutando secuencialmente.
Especialmente vemos scripts “Batch” para Unix. Para conseguir estos scripts que van a automatizar nuestros controles tenemos:

- Python
- Bash
- Cshell
- Perl

Conceptos básicos de scripting

_ Vamos a hablar de “Shell Scripting” usando Bash (o sh). Bash viene de Bourne Again Shell que era una de las primeras versiones que tuvimos en este shell y que fue evolucionando en el tiempo, es una versión gratuita de Bourne Shell original, tiene las mismas características de este y la programación es idéntica, es decir, se tomó fuertemente la base del shell original agregando funcionalidades y ventajas.

ABC del Scripting

_ Los scripts simplemente se pueden concatenar comandos como una secuencia de comandos por lote. En este ejemplo tenemos cuatro líneas de código, siempre en la primera línea se especifica el shell que se va a utilizar y que debería coincidir con el sistema operativo que estamos utilizando, un SO por lo general soporta uno o más shells, en este caso deberíamos usar uno de los que esta soportado, y luego vamos agregando los comandos en la secuencia que necesitamos que se ejecuten, por ejemplo en este caso queremos listar que tiene el directorio etc y enviarlo a un archivo que sea

resultados_auditoria dentro del directorio tmp (el signo > solo, si existiera el archivo resultados_auditoria va a pisar todo el contenido con la salida del comando ls). En el segundo caso vamos a listar procesos, y como tenemos dos signos mayores (>>) en lugar de pisarlo va a agregarlo a lo que ejecutamos previamente. Tenemos algunos otros comandos que nos sirven para evaluar los permisos en un sistema de archivos, permisos puntuales que permiten que un usuario común o que un usuario asignado a un grupo con pocos privilegios pueda ejecutar comandos importantes del SO como si fuera el usuario administrador que en este caso también los enviamos al archivo resultados_auditoria. Y el ultimo comando puntualmente evalúa los inicios de sesión y los envía a este archivo:

```
#!/bin/sh
ls /etc > /tmp/resultados_auditoria
ps -xa >> /tmp/resultados_auditoria
find / -perm 04000 >> /tmp/resultados_auditoria
last >> /tmp/resultados_auditoria
```

Variables

_ También pueden usarse “Variables” que pueden utilizarse para simplificar y generalizar nuestro código. A continuación tenemos el mismo ejemplo que el anterior, solo que en este caso estamos guardando la ubicación de ese archivo de salida en una variable que llamamos RESULTADOS_AUDITORIA. Las variables por lo general se crean de esta forma, dándoles un nombre, como buena práctica se escriben en mayúsculas, y cuando hacemos uso de esa variable se usa el signo $:

```
#!/bin/sh
RESULTADOS_AUDITORIA=/tmp/resultados_auditoria
ls /etc > $RESULTADOS_AUDITORIA
ps -xa >> $RESULTADOS_AUDITORIA
find / -perm 04000 >> $RESULTADOS_AUDITORIA
last >> $RESULTADOS_AUDITORIA
```

Echo

_ Se pueden agregar comentarios a la salida con el comando “echo”, para que nos ayude a separar la información en estos archivos. En el ejemplo puntualmente ponemos una especie de título que dice Resultados de Auditoria y lo estamos agregando al archivo, y agregamos también líneas de separación que nos permiten tener un poco más organizada la información:

```
#!/bin/sh
RESULTADOS_AUDITORIA=/tmp/resultados_auditoria
echo Resultados de Auditoría >> $RESULTADOS_AUDITORIA
ls /etc > $RESULTADOS_AUDITORIA
```

```
echo ---------- >> $RESULTADOS_AUDITORIA
ps -xa >> $RESULTADOS_AUDITORIA
```

If/Then y Corchetes

_ Es posible probar o agregar algunas condiciones a estos archivos. Tenemos el if/then que es muy común utilizado en diferentes lenguajes de programación y los corchetes, esto nos ayuda a probar condiciones, comparar resultados y reportar variaciones. En este ejemplo el script lo que está haciendo es obtener el estado de las conexiones de red, los está enviando a un archivo y lo está comparando con un archivo anterior, en este caso pregunta si el archivo tiene datos, y si los tiene lo envía por correo electrónico a un administrador:

```
#!/bin/sh
netstat -an > /tmp/netstat.obs
diff netstat.base /tmp/netstat.obs > /tmp/ns.diff
if [ -s /tmp/ns.diff ];
        then mail administrador@sitio.com < /tmp/ns.diff
fi
```

_ Es importante saber que los corchetes van separados de la condición, no van pegados porque los corchetes puntualmente se utilizan para establecer rangos en sistemas basados en Unix.

Test

_ El comando 'test' y los corchetes son equivalentes, es decir, funcionan de forma similar. En lugar de poner los corchetes se pone el comando test, y utilizamos validaciones o parámetros puntualmente sobre algún archivo especifico. A continuación vemos un ejemplo:

```
if test -z filename; then ls; fi
if [ -z filename ]; then ls; fi
```

_ Algunos de los test que tenemos disponibles son:

- -b Dispositivo de bloques
- -c Dispositivo de caracteres
- -d Directorio
- -e Existe
- -F Archivo normal
- -g Set GID está configurado, estos parámetros sirven para ver si están establecidos los parámetros que dan permisos especiales sobre los archivos.
- -G Propiedad de EGID
- -k Sticky esta seteado

- -L Enlace simbólico
- -n Cadena no nula
- -O Propiedad de EUID
- -P Es una tubería FIFO
- -r Archivo legible
- -s Archivo no vacío
- -S Es un socket
- -t Es una terminal
- -u Set UID está configurado
- -w Archivo escribible, es decir que tienen permisos de escritura.
- -x Bit de ejecución está configurado
- -z Cadena vacía

Otras verificaciones útiles

_ Tenemos la posibilidad de comparar archivos con los siguientes parámetros:

- A -nt B Si el archivo A es más nuevo que B
- A -ot B Si el archivo A es más antiguo que B
- A -ef B Si el archivo A está enlazado con B, es decir que son enlaces simbólicos esos archivos.
- A = B Si la cadena A es igual a la cadena B
- A -eq B Si la expr. A es igual a la expr. B
- -gt mayor que
- -le menor o igual que
- -ge mayor o igual que
- -lt menor que
- -ne distinto que

Argumentos de línea de comandos

_ También podemos pasarle argumentos a cada uno de estos archivos, esto nos permite la generalización de los scripts, es decir, no poner parámetros en duro dentro de los archivos sino pasarle un parámetro que pueda modificarse y que no nos haga falta modificar el script. Quizás especificar el archivo de salida para los resultados.

_ En este caso, hacemos de cuenta que audit_script es el nombre de nuestro script y le estamos pasando el archivo de salida donde queremos almacenar los resultados entonces cada uno de los parámetros se identifican con el signo $1 en el caso de que sea un parámetro, en el caso de tener más de uno usamos el 2 y así sucesivamente. Entonces pasamos el parámetro y ese resultado se debería enviar al archivo que especificamos como entrada:

```
#!/bin/sh
if [ -z $1 ]; then
  echo Debe especificar un archivo de salida!
  exit 1
fi
echo Enviando resultados a: $1

$./audit_script /tmp/results
```

Aceptando Entrada

_ También podemos en el momento de la ejecución de un script, permitirle al que está ejecutando que ingrese alguna entrada que nosotros podamos validar. Permite una auditoría repetible pero personalizada. Por ejemplo podríamos estar corriendo un script que entre tantas líneas tenga esta sección donde nosotros estamos confirmando sí o no para cada una de esas secciones que se están ejecutando y en este caso podemos preguntarle si desea chequear los puertos abiertos, si responde que si o "s" ejecutamos el comando netstat y lo agregamos a ese archivo, en caso de responder no, esto se saltea, pero esa entrada fue leída en el momento de ejecución y no como parámetro, y es otra alternativa que tenemos también para la ejecución de scripts:

```
#!/bin/sh
echo -n ¿Chequear puertos abiertos [s/n]?
read RESPUESTA
if [ $RESPUESTA = "s" ] || [ $RESPUESTA = "si" ]; then
        netstat -an > /tmp/resultados_auditoria
fi
```

¿Por qué usar Scripts?

_ Básicamente porque simplificamos las tareas repetitivas, podemos auditar un sistema en muy poco tiempo a diferencia de si lo hacemos manualmente y tenemos que ejecutar comandos uno a uno e interpretarlos:

- La auditoría es conducida exactamente de la misma manera cada vez.
- Resultados e informes pueden ser automatizados.
- Simplifica el análisis, porque además de tener lo en nuestro servidor que evaluamos actualmente, podemos centralizarlo y que sea mucho más fácil identificar las configuraciones que no coinciden con nuestra línea de base

Otras herramientas útiles de scripting

_ Estas son algunas utilidades independientes para rebanar y cortar, con esto nos referimos puntualmente a cortar horizontalmente datos, o sea filtrar por línea, o cortar verticalmente, y algunas de estas son: Grep/Egrep, Cut, Sed y Awk.

Grep / Egrep

_ Estos días es mejor usar egrep. La expresión original es Grep (Get Regular ExPression) y traía soporte limitado de expresiones regulares, lo vemos muy utilizado en lo que es administración de Linux para buscar por ejemplo un archivo dentro de un directorio, etc. Egrep es una versión evolucionada, "extendido" de Grep, que tiene un soporte completo de expresiones regulares, nos da mucha más flexibilidad en cuanto a búsqueda, se recomienda usar egrep exclusivamente y no los dos, y considerar poner alias de egrep a grep.

Expresiones regulares: estas son básicamente "meta caracteres" que se usan para describir lo que se quiere encontrar, por ejemplo un "Comodín" (*) que es uno de los más conocidos. Puede ser mucho más complicado. Las expresiones regulares, Regex, van a coincidir tan pronto como sea posible y con todo lo que sea posible, es decir, no necesariamente nos va a dar el resultado más específico a la consulta que estamos haciendo sino que nos va a devolver todo y lo más rápido. Es muy bueno para análisis de Logs, y puede ser también usado para buscar. Entonces, cuando queremos identificar por ejemplo los intentos de inicio de sesión fallidos en un log, podemos usar Grep por cadenas especiales que nos indiquen ese incidente. Algunos de los meta caracteres que se utilizan son:

- * Coincide cero o más de lo previo, o sea de lo que se escribe antes de este carácter.
- [ ] Describe un set.
- ^ Coincide con el principio de una línea.
- $ Coincide con el fin de la línea.
- ? Coincide con exactamente uno de lo previo.
- + Coincide con uno o más de lo previo.
- . Coincide con cualquier carácter.

Cut

_ Este comando sirve para extraer una columna de un conjunto de datos, pero no debemos confundirlo con Col que es un comando que básicamente toma la salida en bruto que tenemos y la reformatea estableciendo espacios en blanco y líneas recibidas.

_ En este ejemplo tratamos de ver la información de la memoria, con col como podemos ver la salida se ve con el mismo formato, donde simplemente lo que hizo col fue

reformatear espacios para que sean trabajables por otro comando. Y en este caso cut agarra la separación establecida, lo corta y toma el segundo campo:

```
[~]$ free -m | grep Mem
Mem:           31827       24286         836        2930        6703        4170
[~]$ free -m | grep Mem | col
Mem:           31827       24294         848        2910        6684        4181
[~]$ free -m | grep Mem | col | cut -f 2
      31827
```

_ Cut es especial para extraer información específica rápidamente, extraer solamente las columnas que necesitamos, y la -f = fields.

Sed

_ Sed viene de Stream Editor. Sirve para rebanar y cortar el texto mientras pasa. Ademas nos sirve para remover texto no deseado, esta es una funcionalidad muy simple y se puede hacer más, nos permite convertir el texto a algo más, es decir, buscar y reemplazar algo que necesitemos reemplazar en un texto, o reformatear el texto a algo que otra herramienta pueda manejar.

_ Por ejemplo cuando recorremos un archivo, en este caso tomamos un script, y decimos que nos corte desde la línea 12 hasta la 16 entonces nos da esa salida solamente:

```
[~]$ sed -n 12,16p nodesource_setup.sh
#   or
# wget -qO- https://deb.nodesource.com/setup_14.x | bash -
#
# CONTRIBUTIONS TO THIS SCRIPT
#
```

Awk

_ AWK viene del apellido de sus creadores Aho, Weinberger y Kernighan. Este busca una coincidencia de patrones y lenguaje de procesamiento de texto. Es rápido y fácil, coincide y reemplaza. Un ejemplo común de lo que hace AWK, es podemos filtrar por ejemplo una memoria libre:

free | awk '/Mem/ { print $2; }'

_ También se puede especificar el separador de campos, en este caso vemos los archivos de usuario de Linux donde cada uno de los campos vienen separados por los dos puntos, y en este caso nos va a imprimir el identificador del usuario que es el primer parámetro:

awk -F: '{print $1;}' /etc/passwd

_ Y si vemos este comando es mucho más simple y eficiente que utilizar sed que es más complejo para este mismo ejemplo:

sed -e 's/:/ /g' /etc/passwd | awk '{print $1;}'

Recetas: con todo esto podemos armar una especie de planilla que se llama receta. Por ejemplo tenemos un comando que se lo pasamos a AWK con un término que queremos buscar, y con print decimos que numero de columna queremos ver:

comando | awk '/termino_busqueda/ {print $<#_columna>; }'

_ Entonces con esa receta podemos armar diferentes platos:

- Memoria física: **free | awk '/Mem/ { print $2; }'**
- Espacio libre en disco: **df | awk '/\/$/ { print $4; }'**
- Direcciones MAC: **ifconfig -a | awk '/ether/ { print $2; }**

Auditando Unix

Auditoría de sistemas: el "Cómo"

_ Cuando hablamos de auditoría de sistemas nos interesa saber el cómo, entonces vamos a empezar a describir algunas tareas para ver cómo podemos llevarlas a cabo en entornos basados en Unix.

_ La tarea que tenemos en este caso es evaluar la seguridad de un sistema desconocido, y tenemos un problema que es si podemos confiar realmente en lo que encontramos cuando usamos herramientas locales, es decir, acá mencionamos los rootkits porque puede ser que en un sistema basado en Unix haya habido alguien malicioso que ingreso e instalo un rootkit que es un set de herramientas que modifican el comportamiento del sistema operativo con el fin de ocultar información o de cambiar el comportamiento.

Caso de estudio: Lrk5 (Linux Rootkit v5)

_ Este rootkit puntualmente oculta archivos, procesos y conexiones de red, borra y edita logs, limpia registros, nos deja un acceso Backdoor, es decir, nos da la posibilidad de volver a ese sistema sin la necesidad de usar un usuario del SO en sí, y escucha en la red. Algunos de los binarios remplazados por lrk5 son:

- chfn
- killall
- chsh
- login
- crontab
- ls
- du
- netstat (2)
- find
- passwd

- ifconfig
- ps (1)
- inetd
- rshd
- tcpd
- syslogd (3)
- pidof
- top (1)

_ Supongamos que nosotros tuvimos algún ataque a algún equipo basado en Unix, lo primero que intentaríamos hacer es ver que procesos tenemos corriendo para ver si son los que habitualmente corren en el sistema operativo y lo haríamos utilizando el comando ps o top, y en este caso estos dos comandos ya fueron modificados por ese rootkit con lo cual no nos estarían dando información confiable. Luego intentaríamos ver las conexiones de redes, en donde al estar netstat afectado nos estaría dando alguna información y quizás ocultando alguna conexión de red especifica, y si quisiéramos ver los logs también está modificando el demonio de los logs syslogd con lo cual la información también podría estar totalmente alterada.

El "Cómo" en un sistema no confiable

_ Tenemos el problema de examinar un sistema no confiable, donde la solución es crear un USB/CD con herramientas, esto aplica a Windows también, pero un inconveniente con esto es que necesitaríamos obviamente no evaluar el sistema en ejecución sino reiniciarlo y quizás montar todos los servicios y sistemas de archivos con esta herramienta. Es una de las alternativas que se proponen para evitar la posibilidad de que el sistema este infectado con un root.

Fuentes de CDs de herramientas

_ Tenemos algunos sistemas operativos que son booteables como Knoppix y Ubuntu, pero estas no son ideales ya que no fueron diseñadas para ser usadas en un sistema corriendo. Luego tenemos una alternativa que es Helix que tiene kits de respuesta en vivo, tienen contenido digital de forensia y es extremadamente caro, y por lo general no cubre todas las posibilidades de evaluación del sistema operativo. Con versiones específicas como Solaris, HP-UX, AIX, no tenemos por lo general kits de respuesta en vivo de estos, y hoy por hoy estos sistemas operativos tienden a desaparecer y se están estandarizando más fuertes en versiones como Ubuntu, Centos, etc, pero en el caso de que tuviéramos que auditar este tipo de sistemas específicamente deberíamos construir nuestro propio CD o USB para estas versiones que nos permita al menos bootear y que tenga scripts específicos para estas versiones.

¿Hacer uno propio?: como dijimos para las opciones anteriores conviene hacer un CD o USB propio:

- Personalizado a nuestros sistemas
- Puede crear USB/CD para cualquier sistema Unix: posiblemente un USB/CD de Auditoría Universal
- Actualización instantánea
- Agregar nuestros propios scripts de auditoría

Lista de Shopping del USB/CD de auditoría: vamos a encontrar muchos de los comandos que mencionamos en el rootkit:

- Librerías compartidas
- who, w, finger
- Librerías estáticas de sistema
- find
- netstat
- df, du
- lsof
- cp
- diff
- script
- ps
- dd
- ls
- sh/bash/csh
- md5
- [/test
- fdisk/cfdisk
- awk
- egrep/grep
- more/less

¿Cómo usar el Set de Herramientas?: montamos el USB/CD como sistema de archivos, obtenemos una shell "limpia" o sea que no esté afectada por un rootkit, y configurar los paths generales y paths para carga de librerías para saber qué librerías y binarios se están usando. A continuación tenemos el paso a paso de los comandos, donde hacemos el montaje de un CD ROM por ejemplo, ejecutamos la shell de bash que esta específicamente almacenada en ese CD, establecemos el path para que ejecute todos los binarios que están en nuestro CD y no en el sistema operativo en sí, y también damos

información sobre cuál es el path de las librerías que va a hacer uso cada uno de esos comandos y exportamos esas variables.

 # mount /mnt/cdrom
 # /mnt/cdrom/bin/bash
 # PATH="/mnt/cdrom/bin"
 # LD_LIBRARY_PATH="/mnt/cdrom/lib"
 # export PATH
 # export LD_LIBRARY_PATH

_ Entonces una vez que ejecutamos todos estos pasos ya tenemos un alto grado de probabilidad que los comandos que ejecutemos sean confiables y no los que estén afectados en el sistema operativo.

Objetivos y actividades de la auditoría Unix

Primer objetivo de auditoría

_ El primer objetivo es la información del sistema, donde vamos a tratar de identificar qué tipo de sistema estamos utilizando, identificar el nivel de actualizaciones (parches) que tenemos, y otra información general del sistema que nos sea útil para encontrar vulnerabilidades. Las actividades de auditoría o comandos que nos van a ser útiles en este caso son:

- Uname: nos permite identificar que versión de sistema operativo estamos utilizando, que plataforma y demás.
- Patchdiag (Sun): este es específico para Solaris, que nos permite ver el estado general de los parches de actualización.
- Etc

Versión del sistema operativo

_ Se obtiene con `uname -a`, nos da información de la arquitectura y sistema operativo, y este comando está disponible universalmente en todas las versiones de Unix.

```
→  ~ uname -a
Linux SyAI-2020 5.3.0-46-generic #38~18.04.1-Ubuntu SMP Tue Mar 31 04:17:56 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
→  ~
```

Sistemas de archivos

_ Si queremos ver puntualmente el sistema de archivos tenemos el comando 'mount', que nos permite ver los sistemas de archivos actualmente montados y los tipos de sistemas de archivos. Nos dice que tenemos montado y visible en este momento en el SO. Podemos ver a continuación en que ubicación se montaron, con que permisos y configuraciones:

```
→ ~ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=9800616k,nr_inodes=2450154,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=1964992k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
```

_ Podemos usar alternativamente el comando 'fdisk -l', que valida lo montado versus real. Nos muestra las particiones que físicamente tenemos en el equipo, donde podemos tener particiones que estén en el equipo pero que no estén montadas.

```
→ ~ fdisk -l /dev/sda
Disk /dev/sda: 238,5 GiB, 256060514304 bytes, 500118192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xc55b1469

Device     Boot Start       End   Sectors   Size Id Type
/dev/sda1  *     2048 500117503 500115456 238,5G 83 Linux
```

Información general del sistema

_ Con respecto a información general del sistema el comando 'free' nos da información de utilización de memoria. Esto nos puede servir para ver cuál es la memoria total que tiene un equipo, si esa memoria está siendo altamente utilizada, si vemos que tenemos un alto uso de la memoria podríamos tener un script malicioso corriendo que nos esté consumiendo esa memoria o nos intente llenar el disco y lo mismo con la memoria swap. Deberíamos estar atento a que tan normal es el uso de memoria en el servidor lo cual deberíamos tener una línea de base y saber cuál es el uso estándar que tiene el servidor que estamos analizando.

```
→ ~ free
              total        used        free      shared  buff/cache   available
Mem:       19649912    13056220      430944     1820352     6162748     4436276
Swap:       2097148      950700     1146448
→ ~
```

Parches

_ Para determinar el nivel de parches, según el sistema operativo deberíamos revisar “Avisos de Seguridad” en el sitio web de soporte del fabricante. Es una de las cosas más difíciles de alcanzar en sistemas Unix porque todo depende de cómo el software fue instalado, ya que estos sistemas tienden a estar descentralizados en muchas compañías, otras ofrecen mejores prácticas, etc.

```
→ ~ sudo cat /var/lib/update-notifier/updates-available

0 packages can be updated.
0 updates are security updates.
```

Segundo objetivo de auditoría

_ El segundo objetivo de la auditoria es determinar el perfil operativo de un servidor, es decir, identificar servicios de red, identificar servicios locales, e identificar comportamiento de red. Esto lo hacemos con el fin de armar una línea de base de lo que pretenderíamos encontrar en el servidor. Y las actividades de auditoría que tenemos son los comandos:

- Netstat
- Lsof (Open files)
- Ps (Process file)
- Top (Table of processes)

Identificando servicios de red

_ Uno de los comandos más útiles que tenemos es `netstat` que nos ayuda a identificar conexiones activas y cuáles son los puertos que están escuchando. Algunas versiones son capaces de relacionar esto a información de procesos, es decir, que tenemos por un lado conexiones establecidas y nos da un identificador de procesos que podemos linkear a un proceso especifico, y ese proceso nos puede dar información de que archivo lo inicio con lo cual podemos tener un panorama completo de si estamos conectándonos a un sitio malicioso de ver cuál es el archivo que inicio el proceso que estableció esa conexión.

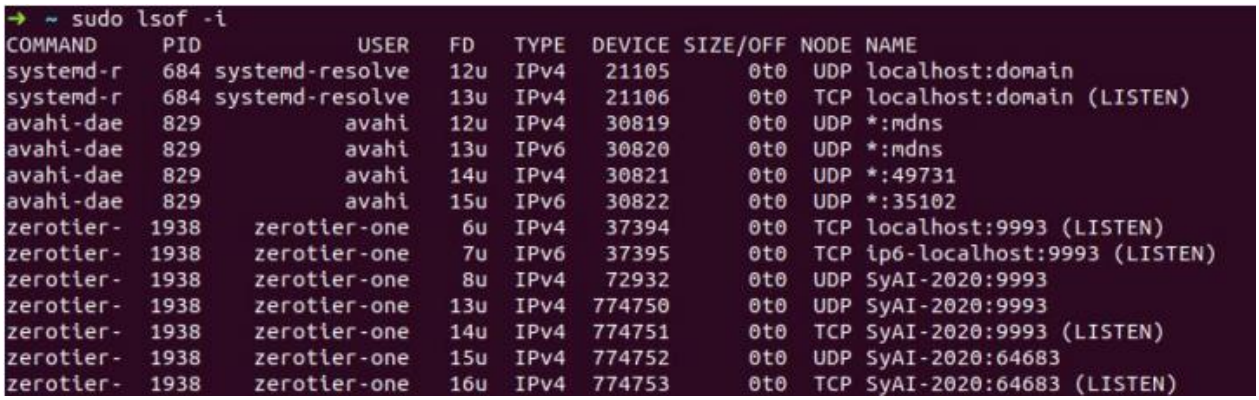
_ En este ejemplo vemos un proceso de salida de netstat, donde podemos ver el par dirección puerto y dirección destino, tenemos el estado de la conexión, y tenemos por un lado el process ID y cuál es el nombre del programa que inicio el proceso y está estableciendo la conexión:

```
→ ~ netstat -ntap
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:41271         0.0.0.0:*               LISTEN      12903/java
tcp        0      0 127.0.0.1:9788          0.0.0.0:*               LISTEN      16589/node
tcp        0      0 127.0.0.1:41734         0.0.0.0:*               LISTEN      13917/kbfsfuse
tcp        0      0 172.22.0.1:9993         0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.0.108:9993      0.0.0.0:*               LISTEN      -
tcp        0      0 172.21.0.1:9993         0.0.0.0:*               LISTEN      -
```

¿Qué es Lsof?

_ Dijimos que era para identificar archivos abiertos, por lo general está instalado por defecto. Se utiliza principalmente para procesos, archivos e investigaciones de estado de red, es decir, ver cuáles son los archivos abiertos que iniciaron un proceso y que establecieron una conexión. Y puede producir salida capaz de ser consumida por otros programas.

_ Acá tenemos un ejemplo para identificar conexiones de red, donde vemos que al final están los datos de la conexión, enfocada desde el punto de vista de archivos abiertos y procesos:



```
➜  ~ sudo lsof -i
COMMAND     PID            USER   FD   TYPE  DEVICE SIZE/OFF NODE NAME
systemd-r   684 systemd-resolve   12u  IPv4   21105      0t0  UDP localhost:domain
systemd-r   684 systemd-resolve   13u  IPv4   21106      0t0  TCP localhost:domain (LISTEN)
avahi-dae   829           avahi   12u  IPv4   30819      0t0  UDP *:mdns
avahi-dae   829           avahi   13u  IPv6   30820      0t0  UDP *:mdns
avahi-dae   829           avahi   14u  IPv4   30821      0t0  UDP *:49731
avahi-dae   829           avahi   15u  IPv6   30822      0t0  UDP *:35102
zerotier-  1938    zerotier-one    6u  IPv4   37394      0t0  TCP localhost:9993 (LISTEN)
zerotier-  1938    zerotier-one    7u  IPv6   37395      0t0  TCP ip6-localhost:9993 (LISTEN)
zerotier-  1938    zerotier-one    8u  IPv4   72932      0t0  UDP SyAI-2020:9993
zerotier-  1938    zerotier-one   13u  IPv4  774750      0t0  UDP SyAI-2020:9993
zerotier-  1938    zerotier-one   14u  IPv4  774751      0t0  TCP SyAI-2020:9993 (LISTEN)
zerotier-  1938    zerotier-one   15u  IPv4  774752      0t0  UDP SyAI-2020:64683
zerotier-  1938    zerotier-one   16u  IPv4  774753      0t0  TCP SyAI-2020:64683 (LISTEN)
```

¿Cómo son iniciados los servicios?

_ Por lo general en los SO basados en Unix tenemos diferentes demonios que son los que hacen que un servicio sea persistente ante los reinicios:

Inetd: originalmente teníamos este que era el “Super Daemon” original y no tenía control de acceso incorporado.

Xinetd: versión moderna y mejorada de inetd y si tiene control de acceso incorporado.

_ Esto puntualmente nos permite establecer cuáles son los Scripts de Inicio en el sistema operativo.

Scripts de inicio

_ En estos sistemas basados en Unix tenemos diferente scripts de inicio, que no necesariamente están vinculados a servicios. Algunos son:

- /etc/rc.d
- /etc/init.d: /etc/rc*.d
- /etc/rc.local
- /etc/init

El arranque de Unix es “determinístico”

_ Lo que queremos decir con determinístico es que siempre sigue la misma secuencia, cuando arrancamos el sistema operativo los process ID siempre deberían estar vinculados a los mismos comandos o procesos, es decir, están ordenados, y si encontráramos que ese orden esta invertido o alterado nos puede servir como alerta para identificar que está sucediendo un incidente y que de alguna manera el SO puede estar afectado.

Comportamiento de red

_ Las clase de opciones de red que están disponibles que nos importen son:

- Enrutamiento de Origen
- Enrutamiento en General
- Pueden las tablas de enrutamiento ser reconfiguradas a través de redirecciones.
- Alguna protección de denegación de servicio para los servidores.

Tercer objetivo de auditoría

_ El tercer objetivo de auditoria es poder identificar el acceso no autorizado, en este caso examinarnos el control de accesos a nivel de red para equipos. Y las actividades de auditoría consisten en revisar estos dos archivos Hosts.allow y Hosts.deny, por lo general se busca que el deny contenga una opción como denegar absolutamente todos y el allow lo usemos para especificar los equipos de confianza en nuestro entorno, es decir, solo permitimos el acceso a un grupo muy específico y preciso de equipos y al resto se le niega el acceso.

Listas de control de acceso (ACLs)

_ Lo que las ACLs dicen es que /etc/hosts.deny debería incluir ALL:ALL, y /etc/hosts.allow debería listar equipos individuales (confiables) servicio por servicio. Por defecto, tcpd (el demonio que valida estos archivos) está instalado, pero el archivo hosts. deny está vacío por lo general, entonces si tenemos que tomar esta configuración deberíamos asegurarnos que los archivos contienen información específica de lo que necesitamos, denegar todo lo que está en allow puntualmente.

Cuarto objetivo de auditoría

_ Otro objetivo es la administración/acceso de usuarios, asegurar cuentas de usuario únicas, identificar usuarios autorizados, examinar configuraciones de contraseñas y asegurar que se están usando contraseñas fuertes. Como actividades de auditoría tenemos examinar /etc/passwd, /etc/shadow y la herramienta John the Ripper que se utiliza para hacer ataques de fuerza bruta a los archivos de contraseña con el fin de identificar si tenemos contraseñas débiles.

Seguridad en el momento de arranque

_ Los sistemas Unix son particularmente vulnerables con acceso físico, es posible cambiar/borrar contraseñas si podemos reiniciar el sistema. Para mitigar estos riesgos debemos buscar contraseñas para el momento de arranque, y restringir teclas de acceso rápido para reinicio como es ctrl alt delete.

El “cómo hacerlo” varía

_ Dependiendo del sistema operativo, puntualmente en Linux tenemos dos cargadores de inicio principales: LILO/Grub (hoy se usa mas Grub). Grub permite protección por contraseña del proceso de inicio, por defecto la contraseña es almacenada en texto plano en /etc/grub.conf. El comando /sbin/grub-md5-crypt permite crear un hash cifrado. Y deberíamos verificar que la línea de contraseña en /etc/grub.conf es correcta, es decir, que en lo posible este configurado y no en texto plano.

Deshabilitando teclas de acceso rápido

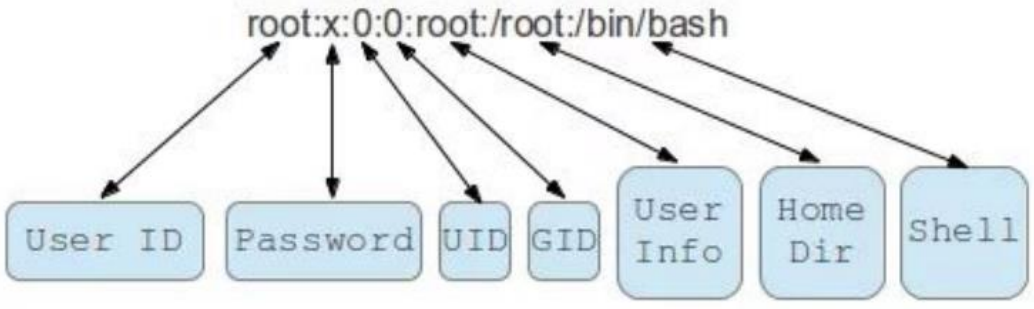
_ Muchos sistemas Unix x86 mapean Control-Alt-Delete para un reinicio automático, esto se puede controlar a través del archivo /etc/inittab. Y podemos comentar la línea para prevenir el comportamiento (agregar un “#” al inicio de la línea para comentarla).

Limitando el Acceso Remoto

_ Acá hacemos referencia al usuario root, deberíamos evaluar si alguien debería poder loguearse como root desde otra ubicación, por lo general la mejor practica es que alguien se loguee con su cuenta desde un servidor remoto y si tiene privilegios para cambiar a la cuenta root lo haga desde el mismo servidor. Muchos sistemas Unix permiten restringir esto usando el archivo de configuración /etc/securetty donde se especifican puntualmente si están permitidas estas conexiones. Y se puede verificar que solamente terminales conectadas localmente/físicamente permiten iniciar sesión directamente con root.

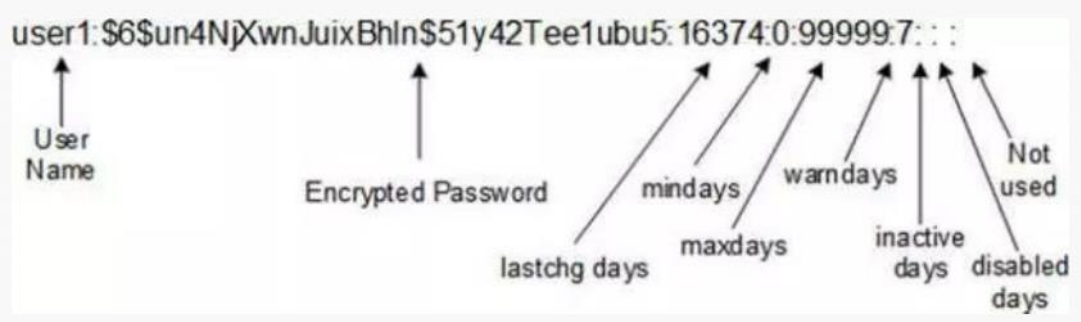
Archivo ‘passwd’

_ Este tiene información de los usuarios. Es una ubicación tradicional de información de autenticación. A continuación vemos los campos de este archivo, tenemos por un lado el user ID que nos permite sacar un listado de usuario, el campo password no almacena la password en sí, el UID hace referencia al grupo principal o primario del usuario, GID es el identificador de grupo, el user info puede ser una descripción más detallada del usuario, home directory, y tenemos la shell.



Archivo ‘shadow’

_ A diferencia del archivo passwd que puede ser leído por todos los usuarios, a este solamente root puede leerlo.



_ Revisión de los campos:

- Nombre de Usuario
- Hash de la Contraseña
- Días desde 1/1/1970 que la contraseña fue cambiada.
- Días que deben pasar para que la contraseña pueda ser cambiada ○ Días después que la contraseña debe ser cambiada.
- Días antes de la expiración que el usuario es advertido.
- Días después de la expiración que la cuenta es deshabilitada.
- Días desde 1/1/1970 que la cuenta fue deshabilitada

_ Generalmente configurado por /etc/default/useradd para cada una de las opciones de arriba.

Herramientas de evaluación de contraseñas

John the Ripper: que es una herramienta que nos permite hacer:

- Cracking distribuido
- Corre en Windows y Unix
- Contraseñas del estilo BSD
- Contraseñas basadas en DES
- Contraseñas basadas en Twofish
- Hashes NTLM

Quinto objetivo de auditoría

_ El quinto objetivo es identificar el acceso no autorizado, asegurar que solamente los archivos necesarios tienen los bits set-user o set-group configurados (permisos que permiten a un usuario no privilegiado, usar comandos privilegiados), e identificar archivos modificados recientemente. Entonces como actividades de auditoría usamos dos comandos comunes que son Find y Ls.

¿Qué Buscamos?

- Los archivos que tengan los permisos SUID y SGID (privilegios a nivel de usuarios o grupo).
- Binarios recientemente modificados, por ejemplo comandos de SO que intentaron ser afectados.
- Archivos ocultos, que no estén visibles a nivel del sistema de archivos pero si podemos ingresar a los procesos podemos verlo.
- Entradas extra o incorrectas en /etc/passwd.
- Cualquier cosa “fuera de lo ordinario”.

Comando ‘find’

_ Este trabaja dada una expresión, ‘find’ busca el árbol de directorios y realiza alguna acción en los archivos cuyos atributos coinciden. Puede buscar creación, modificación y fechas de acceso, patrones de nombre, tipo de archivo, tamaño, permisos, dueño, grupo.

Encontrar archivos SUID: obtener una lista de todos los archivos suid y sgid, encontrar archivos SUID debería ser parte de la línea de base, y usar el comando find.

Localizar binarios modificados recientemente

_ Creamos por ejemplo un archivo que tenga un tstamp en una carpeta temporal y luego buscamos archivos que sean más nuevos que el archivo que creamos y que sean del tipo f

 # touch -m 04072021 /tmp/tstamp
 # find / -newer /tmp/tstamp -type f

_ Entonces esto encuentra, imprime y ordena por fecha:

- Todos los archivos regulares.
- Más nuevos que 7 de abril de 2021 (fecha especificada acá).

Auditoría de Seguridad

Objeticos del apartado: comprender las implicancias de la auditoría de seguridad, diferenciar entre componentes auditoría de seguridad física y lógica, conocer las consideraciones de seguridad a incluir en el informe de auditoría, identificar las vulnerabilidades más comunes en redes, diferenciar entre elementos de auditoría de redes física y lógica.

Auditoría de seguridad

_ Debe evaluarse en la auditoría si los modelos de seguridad están en consonancia con las nuevas arquitecturas, las distintas plataformas y las posibilidades de las comunicaciones. Es decir, evaluamos esta auditoria basándonos en algunos modelos de seguridad que deben estar en consonancia con las nuevas arquitecturas que estén siendo utilizadas, además de las distintas plataformas que estemos analizando (por ejemplo un data center que este hosteado dentro de la empresa) y también las posibilidades de las comunicaciones. Los grandes grupos de controles, que además de poderlos dividir en manuales y automáticos, o en generales y de aplicación, son los siguientes:

Controles directivos: hacen referencia a aquellos objetivos que bajan a desde la dirección de la empresa y que deben ser controlados y que están relacionados a normativas y estándares.

Controles preventivos: están enfocados a prevenir incidentes o accesos indebidos puntualmente.

Controles detectivos: estos están alineados a detectar de manera lo más pronto posible cualquier ingreso indebido a los sistemas.

Controles correctivos: intentan recuperar un problema en particular, por ejemplo se elimina un archivo puntualmente.

Controles de recuperación: hablamos de un conjunto de controles correctivos, es decir, son muchas acciones que nos llevan a recuperar un sistema de manera global.

_ El sistema de control interno debe basarse en políticas, y se implementa con apoyo de herramientas. Cuando existe un sistema de control interno adecuado, los procesos de auditoría, especialmente si son periódicos, son revisiones necesarias pero más rápidas, con informes más breves. Si el control interno se hizo de manera adecuada y se fue documentando, cuando venga una auditoria va a ser mucho más fácil y rápida la comprensión de esa auditoria.

Evaluación de riesgos

_ Se trata de identificar y cuantificar la probabilidad e impacto de los riesgos y analizar medidas que disminuyan la probabilidad de que ocurran los hechos o mitiguen el impacto. Cuando hablamos de riesgo, nunca decimos que un riesgo puede eliminarse por completo sino que puntualmente hablamos de mitigarlo. Para evaluarlos hay que considerar:

- El tipo de información almacenada, procesada y transmitida.
- La criticidad de las aplicaciones.
- La tecnología usada, el marco legal aplicable, el sector de la entidad, la entidad misma y el momento.

_ Con todos estos factores lo que tratamos de definir es cual es el impacto real que tiene algún riesgo puntual en la organización. Es necesario revisar si se han considerado las amenazas y, además, errores y negligencias en general que pueden traducirse en daños, en algunos casos irreversibles.

_ Es necesario evaluar las vulnerabilidades que existen, ya que la cadena de protección se podrá romper con mayor probabilidad por los eslabones más débiles. Esto apunta a que los atacantes no siempre van a buscar el método más difícil para acceder a la infraestructura sino tratan de ver cuál es el punto más débil que no siempre está basado en la tecnología. El factor humano es el principal a considerar, salvo en algunas situaciones de protección física muy automatizados.

_ Es conveniente que haya cláusulas adecuadas en los contratos, sean de trabajo o de otro tipo, especialmente para quienes están en funciones más críticas. Es necesaria una separación de funciones, esto tiene que ver con que una persona no sea quien inicie y termine un proceso con el fin de evitar fraude, entonces mientras más separadas tengamos las funciones y más personas participen de un proceso es menos probable que ocurra un fraude.

_ Además de reducirse, se pueden transferir los riesgos contratando seguros. Otra posibilidad es asumir los riesgos, en donde planificamos o estimamos cuanto nos costaría sufrir un determinado ataque y ver si la inversión que necesitamos hacer para mitigar ese riesgo justifica ese costo.

_ En la auditoría externa se trata de saber si la entidad ha evaluado de forma adecuada los riesgos, si los informes han llegado a los destinatarios correspondientes y si se están tomando las medidas pertinentes, así como si el proceso se realiza con la frecuencia necesaria y no ha constituido un hecho aislado.

_ En estos casos se debe considerar la metodología que se sigue para evaluar los riesgos más que las herramientas, si se han considerado todos los riesgos y si se han medido bien.

_ Es necesaria la designación de propietarios de los activos, esto se hace con el fin por ejemplo de un recurso que almacena información sensible, tenga a alguien que este velando por la confidencialidad de ese activo, entonces va a exigir que se estén aplicando las mejores prácticas de configuración, entre otras cosas.

Triada de seguridad

_ Al hablar de seguridad siempre se habla de sus tres dimensiones clásicas y los controles buscan garantizar alguna de estas características:

Confidencialidad: solo la persona que tenga los permisos adecuados debería acceder a la información, y que tenga una necesidad de negocio concreta.

Integridad: se refiere al tema de tener premisos para modificar, alterar o eliminar la información, y quien lo haga tenga los permisos adecuados y que sus objetivos sean con un beneficio para la empresa.

Disponibilidad: aquellas personas que necesiten acceder a la información la tengan de alguna manera oportuna.

Autenticidad: debe además existir autenticidad, es decir, asegurarnos de que la información a la que accedemos es por un lado autentica y por otro lado accedida por usuarios auténticos, o sea que el identificador que utilizan se corresponda a ellos y no estén suplantando a otra persona.

Fases de la auditoría de seguridad

- Definición de los objetivos, alcance y profundidad de la auditoría.
- Análisis de posibles fuentes y recopilación de información.
- Determinación del plan de trabajo y comunicación a la entidad
- Adaptación de cuestionarios y consideración de herramientas o perfiles de especialistas necesarios, sobre todo en la auditoría externa.
- Realización de entrevistas y pruebas.
- Análisis de resultados y valoración de riesgos.
- Presentación y discusión del informe provisional.
- Informe definitivo.

Auditoría de la seguridad física

_ En este caso se evaluarán las protecciones físicas de datos, programas, instalaciones, equipos, redes y soportes, y personas. Puntualmente todo lo que rodea a los contenedores de información desde el punto de vista físico. Las amenazas pueden ser muy diversas, como sabotaje, vandalismo, terrorismo, accidentes de distinto tipo, incendios, inundaciones, averías importantes, derrumbamientos, explosiones, etc. Desde la perspectiva de las protecciones físicas algunos aspectos a considerar son:

- Ubicación del centro de datos, servidores locales y cualquier elemento a proteger.
- Protección de computadoras portátiles, incluso fuera de las oficinas: aeropuertos, automóviles, restaurantes, etc.
- Estructura, diseño, construcción y distribución de los edificios y de sus plantas.
- Riesgos a los que están expuestos por agentes externos como por accesos físicos no controlados.
- Amenazas de fuego; riesgos por agua; por accidentes atmosféricos o por averías en las conducciones; problemas en el suministro eléctrico, tanto por caídas como por perturbaciones. Controles ambientales
- Controles tanto preventivos como de detección relacionados con los puntos anteriores.
- Además debe controlarse el contenido de carteras, paquetes, bolsos o cajas, ya que podrían contener explosivos, así como lo que se quiere sacar del edificio, para evitar sustituciones o sustracción de equipos, componentes, soportes magnéticos, documentación u otros activos.
- Protección de soportes magnéticos (acceso, almacenamiento y posible transporte).
- Protección de documentos impresos y de cualquier tipo de documentación clasificada.

_ Todos los puntos anteriores pueden estar además cubiertos por seguros.

_ Es decir, cuando auditamos seguridad física, consideraríamos evaluar la ubicación del centro de datos, sala de servidores y cualquier elemento a proteger, y la protección de documentos impresos y de cualquier tipo de documentación clasificada.

Auditoría de la seguridad lógica

_ En este caso hablamos de los controles desde el punto de vista virtual. Es necesario verificar que cada usuario sólo pueda acceder a los recursos a los que le autorice el propietario según su función, y con las posibilidades que el propietario haya fijado, donde por lo general tenemos los permisos de lectura, modificación, borrado, ejecución, etc, lo que representaríamos en una matriz de accesos en la que figurarían los sujetos en las filas, y en las columnas los objetos, que puedan ser accedidos con mayor o menor granularidad y las posibilidades que se le otorgan. Desde el punto de vista de la auditoría es necesario revisar cómo se identifican y sobre todo autentican los usuarios, cómo han sido autorizados y por quién, y qué ocurre cuando se producen transgresiones o intentos: quién se entera, cuándo y qué se hace.

_ En cuanto a la autenticación el método más usado es la contraseña, cuyas características serán acordes con las normas y estándares de la entidad. Algunos de los aspectos a evaluar respecto a las contraseñas pueden ser:

- Quién asigna la contraseña.

- Longitud mínima y composición de caracteres (complejidad de la contraseña).
- Vigencia de la misma.
- Control para no asignar las “x” últimas (Historial).
- Número de intentos fallidos que se permiten al usuario.
- Si las contraseñas están cifradas, y bajo qué sistema.
- Protección o cambio de contraseñas iniciales que llegan en los sistemas, y que a menudo aparecen en los propios manuales.
- Controles existentes para evitar y detectar Troyanos. Instalar un antivirus.
- La no-cesión (no compartir nuestra cuenta), y el uso individual y responsable de cada usuario, a partir de la normativa.
- Promover el uso de diferentes contraseñas para diferentes sistemas.
- La solución más adecuada por ahora puede consistir en utilizar sistemas de identificación únicos (single sign-on).
- Verificar que el proceso de altas, variaciones y bajas de usuarios se realiza según la normativa en vigor. Debería estar previsto bloquear a un usuario que no accediera por un período determinado.
- Examinar situaciones de bloqueo por la existencia de un sólo administrador. Deberíamos tener más de un administrador.

Técnicas, métodos y herramientas

_ En cada proceso de auditoría, se fijan los objetivos, ámbito y profundidad, lo que sirve para la planificación y para la consideración de las fuentes, según los objetivos, así como de las técnicas, métodos y herramientas más adecuados. Como métodos y técnicas podemos considerar los cuestionarios, las entrevistas, la observación, los muestreos, las CAAT (Técnicas de Auditoría Asistidas por Computadora), las utilidades y programas, los paquetes específicos, las pruebas y la simulación en paralelo con datos reales.

Consideraciones respecto al informe

- Se harán constar los antecedentes y los objetivos, qué metodología de evaluación de riesgos y estándares se ha utilizado, y una breve descripción de los entornos revisados.
- Debe incluirse un resumen para la Dirección en términos no técnicos.
- Dependiendo de los casos, será preferible agrupar aspectos similares: seguridad física, seguridad lógica, etc, o bien clasificar los puntos por centros o redes.
- En cada punto que se incluya debe explicarse por qué es un incumplimiento o una debilidad, así como alguna recomendación, a veces abarcando varios puntos.
- El informe debe ser necesariamente revisado por los auditados, así como discutido si es necesario antes de emitir el definitivo.
- En muchos casos se recogen las respuestas de los auditados, sobre todo cuando la auditoría es interna.

- La entidad decide qué acciones tomar a partir del informe, y en el caso de los auditores internos éstos suelen hacer también un seguimiento de las implementaciones.
- En algunos casos los informes se han usado para comparar la seguridad de diferentes delegaciones, sucursales, o empresas de un mismo grupo, o bien filiales de una multinacional, pero si los entornos no son homogéneos las comparaciones pueden no ser útiles y llegar a distorsionar.
- Es necesario diferenciar puntos muy graves, graves, memorables, u otra clasificación, en definitiva establecer algunas métricas de seguridad y clasificar los puntos según su importancia y prioridad.
- Es importante que se delimiten las responsabilidades y los entregables que son objeto de auditoría externa en el contrato o propuesta.
- Algunos de los puntos importantes que pueden llegar a estar en los informes respecto a seguridad pueden ser la ausencia de:
  - Copias de activos críticos en cuanto a la continuidad, en lugar diferente y distante.
  - Cumplimiento de la legislación aplicable así como de las políticas y normas internas.
  - Diferenciación de entornos de desarrollo y producción. ○ Involucramiento de la Alta Dirección.
  - Motivación de los empleados y directivos en relación con la seguridad. ○ Evaluación periódica y adecuada de riesgos.
  - Segregación de funciones.
- Es frecuente también que quienes han pedido la auditoría quieran conocer después en qué medida se han resuelto los problemas, conocer la evolución de la situación en el tiempo.

Contratación de auditoría externa

_ Si no se sigue un proceso de selección adecuado de auditores externos, no se pueden garantizar los resultados. La empresa debe tener una clara definición de cómo va a seleccionar a la entidad que va a gestionar la auditoria. Algunas consideraciones pueden ser:

- La entidad auditora debe ser independiente de la auditada en el caso de una auditoría externa.
- Las personas que vayan a realizar el trabajo deben ser independientes y competentes, según el objetivo.
- La auditoría debe encargarse a un nivel suficiente alto, normalmente dirección general o consejero delegado.

- Puede ser necesario dar o mostrar a los auditores todo lo que necesiten para realizar su trabajo, pero nada más, e incluso lo que se les muestre o a lo que se les permita acceder puede ser con restricciones.

Fundamentos de auditoría de redes

Vulnerabilidades en redes

_ En las redes de comunicaciones, por causas propias de la tecnología, pueden producirse básicamente tres tipos de incidencias (estas hacen referencia a las redes cableadas aunque también podemos encontrar algunas de estas en las redes wireless):

Alteración de bits: una trama puede sufrir variación en parte de su contenido. Se agrega un sufijo a la trama con un código de redundancia cíclico (CRC) que detecte cualquier error y permita corregir errores que afecten hasta unos pocos bits.

Ausencia de tramas: alguna trama puede desaparecer en el camino del emisor al receptor. Se suele atajar este riesgo dando un número de secuencia a las tramas.

Alteración de secuencia: el orden en el que se envían y se reciben las tramas no coincide. Unas tramas han adelantado a otras. En el receptor, mediante el número de secuencia, se reconstruye el orden original.

Riesgos en redes

_ Teniendo en cuenta que es físicamente posible interceptar la información, los tres mayores riesgos a atacar son:

Indagación: un mensaje puede ser leído por un tercero, obteniendo la información que contenga, especialmente en entornos donde la información se transmita en texto claro.

Suplantación: un tercero puede introducir un mensaje adulterado que el receptor cree proveniente del emisor legítimo.

Modificación: un tercero puede alterar el contenido de un mensaje.

_ La diferencia entre suplantación y modificación es que en el caso de suplantación el mensaje es generado directamente por el tercero, y en el caso de modificación, el atacante intercepta el paquete del emisor, lo modifica y lo transmite.

_ Para este tipo de actuaciones, la única medida prácticamente efectiva en redes MAN y WAN (cuando la información sale del edificio) es la criptografía, o sea básicamente usar un canal cifrado de información.

Cableado de planta

_ El cableado que va desde el armario distribuidor a cada uno de los potenciales puestos, suele llamarse de “planta” suele ser de cobre y es propenso a escuchas (“pinchazos”) que pueden no dejar rastro. Por ejemplo el cable coaxial. Este tiene poco alcance.

Cableado troncal

_ El cableado troncal (conexión entre armarios y salas de equipos) y el de ruta (conexión desde sala de equipos hacia los transportistas de datos) se tienden frecuentemente mediante fibra óptica, que son muy difíciles de interceptar, debido a que no provocan radiación electromagnética y a que la conexión física a una fibra óptica requiere una tecnología delicada y compleja, pero son de mayor alcance.

_ En el propio puesto de trabajo puede haber peligros, como grabar/retransmitir la imagen que se ve en la pantalla, teclados que memorizan el orden en que se han pulsado las teclas, o directamente que las contraseñas estén escritas en papeles a la vista. Dentro de las redes locales, el mayor peligro es que alguien instale una “escucha” no autorizada. Al viajar en texto plano la información dentro de la red local, es imprescindible tener una organización que controle estrictamente los equipos de escucha, bien sean estos físicos (“sniffers”) o lógicos (“traceadores”).

Auditando la red física

- Deben comprobarse que efectivamente los accesos físicos provenientes del exterior han sido debidamente registrados y que desde el interior del edificio no se intercepta físicamente el cableado (“pinchazo”).
- En caso de desastre, debe comprobarse cuál es la parte del cableado que queda en condiciones de funcionar y qué operatividad puede soportar.
- Como objetivos de control, se debe marcar la existencia de:
  - Áreas controladas para los equipos de comunicaciones.
  - Protección y tendido adecuado de cables y líneas de comunicaciones.
  - Controles de utilización de los equipos de pruebas de comunicaciones, usados para monitorear la red y su tráfico.
  - Atención específica a la recuperación de los sistemas de comunicación de datos en el plan de recuperación de desastres en sistemas de información.
  - Controles específicos en caso de que se utilicen líneas telefónicas normales con acceso a la red de datos para prevenir accesos no autorizados al sistema o a la red.

_ Podemos trasladar estas medidas a redes wireless.

Auditando la red lógica

_ Se debe controlar que un equipo no pueda enviar indiscriminadamente mensajes ya que puede bloquear la red completa. Es necesario monitorear la red, revisar los errores o situaciones anómalas que se producen y tener establecidos los procedimientos para detectar y aislar equipos en situación anómala. Una solución totalmente efectiva es el cifrado de las comunicaciones. Como objetivos de control, se debe marcar la existencia de:

- Contraseñas y otros procedimientos para limitar y detectar cualquier intento de acceso no autorizado a la red de comunicaciones.
- Facilidades para detectar errores de transmisión y establecer las retransmisiones apropiadas.
- Controles para asegurar que las transmisiones van solamente a usuarios autorizados.
- Registro de la actividad de la red.
- Técnicas de cifrado de datos donde haya riesgos de accesos impropios a transmisiones sensibles.
- Controles adecuados que cubran la importación o exportación de datos a otros sistemas informáticos.

Auditoría de redes

Factibilidad de administración de redes

_ Con la administración de redes, viene alguien a la compañía, hace un escaneo, nos da un reporte muy extenso, y ese reporte queda sin accionar. Entonces el enfoque típico es realizar un escaneo, obtener un reporte de 1000 páginas por ejemplo y que el entorno de red permanezca sin cambios. La organización y la gestión de las mitigaciones es clave. Y es conveniente realizar una evaluación de riesgos para priorizar los componentes de red, y hacer la auditoría por partes.

Metodología general

_ La metodología general que se utiliza es:

1. Determinar áreas de responsabilidad.

2. Investigar riesgos y vulnerabilidades.

3. Asegurar el perímetro.

4. Asegurar la DMZ (zona desmilitarizada) y sistemas críticos.

5. Eliminar vulnerabilidades accesibles externamente.

6. Eliminar vulnerabilidades accesibles internamente.

7. Buscar malware.

Personalización de la metodología: puede consistir en identificar sistemas/dispositivos clave, puede haber sistemas adicionales que necesiten ser auditados antes en el proceso de auditoría de redes de acuerdo a nuestras prioridades. Saber dónde están las joyas de la corona en la red, es decir, el activo critico más importante al que un atacante quiere tener acceso. Mantener una lista organizada de qué componentes de red existen y cuándo será auditado cada uno, es decir, tener un inventario, fecha de ultima auditoria y con qué frecuencia se va a auditar. Auditar redes por funciones, ya que además muchos dispositivos cumplen hoy múltiples funciones y roles.

Routers

Preparación de la auditoría

_ La etapa de preparación de la auditoria consiste en definir el alcance y realizar la investigación (¿qué está siendo protegido?, ¿qué riesgos existen?, ¿cómo está configurado el router?, ¿cuál es la arquitectura? y ¿qué procesos existen?).

Fuentes para investigación

_ Algunas fuentes para investigación de los routers, puntualmente en su función en la organización, tenemos por un lado:

- Entrevistas: que se las podemos hacer a el equipo de auditoría, administradores de sistemas, administradores de red, equipo de políticas, y seguridad de la información.
- Documentación del router: por un lado tener la definición funcional del router, y podría ser complementada con diagramas de red.
- Fuentes externas: alertas y boletines del fabricante del sistema, alertas de vulnerabilidades de sans, grupos de usuarios/grupos de discusión, y fuentes de “mejores prácticas”.

Arquitectura

_ La arquitectura de los routers debe soportar el flujo de información, es decir, si nos comunicamos con empresas externas, con usuarios finales o hacia el interior de la empresa, entonces deberíamos ver qué información está siendo protegida, qué sistema operativo y nivel de parches está siendo usado, cuál es el rol del router, un router de borde (que filtra tráfico desde el exterior) donde tenemos opciones de arquitectura como router como única línea de defensa y router trabajando con un firewall; luego tenemos router interior, y un router backbone (hace conexión entre los de interiores y bordes).

Procesos de Prueba

_ Tenemos diferentes procesos como:

- Control de cambios, es decir, cuando cambiamos la configuración, si está quedando registrado quien autoriza ese cambio.
- Copias de seguridad, nos interesa tener puntualmente de la configuración.
- Administración de usuarios, es decir, quien accede a este router para configurarlo.
- Política de Contraseñas, al igual que en los SO.
- Actualizaciones de Parches, al estar expuestos a vulnerabilidades.
- Construcciones seguras y estandarizadas para plataformas de routers, es decir, tratar de implementar mejores prácticas al momento de instalar por primera vez el router.

_ También conducimos entrevistas, revisamos documentación, y realizar simulacros para ver que tan bien está configurado el router y que tan bien está cumpliendo la función que le fue encomendada.

Verificación del proceso de simulacros

_ Acá tenemos opciones de mostrar una alerta reciente, entender el procedimiento utilizado actualmente por los administradores para abordar las alertas, obtener evidencia de auditoría de que el proceso está en funcionamiento a través de entrevista u observación, y sugerir mejoras al proceso

¿Por qué Routers Cisco?

_ Hablamos de estos puntualmente porque tiene la mayor cuota de mercado de routers de internet. Los conceptos pueden ser aplicados a cualquier Router, ya que más allá de los comandos, lo importante son los conceptos detrás de esto. Al ser basados en línea de comandos, pueden ser más difícil de administrar y aprender a auditarlos posibilitará aplicar este conocimiento a routers de otros vendedores.

Filtrado estático de paquetes

_ Es el control de tráfico implementado en la mayoría de los routers. Funciona dividiendo y midiendo. Si le decimos a un router de forma explícita que permita tráfico hacia tal destino, el mismo evalúa, según los datos de la red, en los que se puede aceptar o rechazar el tráfico si es que cumple o no con las condiciones.

Filtrado con estado

_ La mayoría de las conexiones de red están basadas en estímulos de respuesta, esto significa que tenemos que permitir que las respuestas entren a nuestra red. Lo que hace el filtrado con estado es guardar el estado de la conexión que se inició y permitir el reingreso

de la respuesta a esa conexión inicial. El problema que tiene esto es que los paquetes pueden ser manipulados para parecer respuestas inofensivas. El filtrado con estado "recuerda" el tráfico saliente entonces solo las respuestas legítimas son permitidas para ingresar.

Cuando usar filtrado estático o con estado

_ Como dijimos, el estático era el explicito, y el estado es cuando hacemos un ping a algún paquete del cual esperamos tener respuesta. Usar estático para decisiones absolutas:

- Bloquear tráfico originado desde una dirección ip privada.
- Bloquear todo el tráfico direccionado a los puertos snmp (simple Network Management Protocol).
- Bloquear todos los echo-requests (ping) entrantes.

_ Usar con estado para decisiones condicionales:

- Por ejemplo, no filtrar nada, o para todo lo demás aplicar una condición especifica.
- El despliegue del Router es dependiente en la configuración del perímetro, es decir, debemos analizar en que contexto esta esté router para ver cuál va a ser la configuración que va a tener y cuáles van a ser las características dentro de la organización.

Listas de control de acceso (ACLs)

_ Las ACLs controlan el tráfico hacia y a través del router. Cisco tiene múltiples tipos de ACLs. Las ACLs pueden ser estáticas o con estado dependiendo del fabricante y de las capacidades del router. Para usar una ACL, el router necesita saber dónde debería ser aplicada, es decir, qué interfaz debería usar y en qué dirección (Entrante/Saliente).

Accediendo a los routers

_ Tenemos métodos de administración y configuración:

- Local: se puede hacer dentro de la misma red, es el mejor, pero no siempre es realista.
- Remoto: tenemos algunas opciones como Telnet, SSH, HTTP, SNMP, TFTP.

_ El acceso seguro seria:

- O bien a través de una red de administración (servidor de salto).
- A través de una comunicación cifrada usando SSH, IPSec.

Auditando métodos de acceso

_ Deshabilitar acceso administrativo que no es necesario, donde el acceso remoto (directo) debería ser deshabilitado si no es necesario. Luego el acceso cifrado (por

ejemplo, SSH, IPSec) es preferido sobre telnet para acceso remoto, además de controlar el acceso remoto al router a través de ACLs. Usar timeouts para inactividad de la sesión.

Autenticación

_ Usar cuentas individuales por Administrador para control de accesos. Administrar Autenticación, Autorización y Responsabilidad (Accountability), esto se suele encontrar como la triple A. Las cuentas centralizadas pueden ser usadas a través de RADIUS, TACACS, etc, estos protocolos son directorios de usuarios donde podemos conectar diferentes sistemas y tener centralizado lo que es la administración de usuarios.

SNMP

_ Es recomendable deshabilitar SNMP completamente porque transmite mucha información, especialmente a través de estas “community strings” que en algunos casos son de lectura-escritura y nos permiten administrar la red de manera indebida. Por defecto deberíamos prohibir las “community strings” por defecto tanto las públicas como las privadas. Y acotar el acceso a direcciones autorizadas con ACLs (controlan el tráfico hacia y a través del router).

Deshabilitar los servicios de administración innecesarios

_ Por ejemplo Finger, que provee información de un usuario en un sistema. Identd identifica el propietario de una conexión entre un cliente y un servidor. Y con HTTP podríamos tener una administración a través de un servidor web, en donde por ahí es necesario deshabilitarlo.

Cifrado de Contraseñas

_ Verificar como están almacenadas las contraseñas, que algoritmos de cifrado/hash están siendo utilizados, verificar quién tiene acceso a los hashes, y verificar cómo es realizada la autenticación a lo largo de la red.

AAA (Autenticación, Autorización y Responsabilidad)

_ Es decir, aseguramos que la actividad de los usuarios está siendo auditada. Para saber que debería ser auditado, deberíamos revisar la política y como mínimo el acceso al sistema (exitoso y fallido), actividad administrativa, auditar las fallas y auditar los cambios a la configuración.

Firewalls

Asegurando el perímetro ¿dónde está el fin del perímetro?

_ Hay una gran cantidad de potenciales puntos de acceso hacia nuestras redes, tenemos:

- VPNs / Módems (B2B), orientadas a usuarios finales y conexiones entre negocios.
- Wireless
- Router de Borde
- Firewall de Perímetro

Defensa en profundidad (DiD)

_ Hace referencia a como aplicamos controles de seguridad a las diferentes capas de protección de la organización. Las “Capas” deben ser incorporadas en la seguridad:

- Firewalls de Perímetro
- Firewalls internos
- Sistemas de Detección de Intrusiones (IDS)
- Routers de Borde
- Routers Internos
- Políticas y Procedimientos
- Auditorías

_ Múltiples controles deben estar presentes y ser evaluados. En un entorno donde se utiliza un esquema de Defensa en Profundidad (DiD), no es obligatorio que los Routers y Firewalls tengan reglas diferentes, ya que pueden tener reglas similares incluso repetidas para el caso donde uno de los dispositivos falle o sea comprometido.

¿Por qué realizar auditoría de perímetro?

_ Primero para ver si las reglas de filtrado están funcionando ya que son complejas y queremos asegurarnos de que sean efectivas. Muchos cocineros echan a perder la sopa, donde esto hace referencia a que grandes organizaciones tienen varios administradores y uno puede echar a perder lo que hizo el otro. Y también errores en el código del fabricante, la auditoría es una capa de defensa en profundidad porque nos permite ver si puntualmente el equipo está produciendo errores.

Filtros de firewall vs filtros de router

_ Los conceptos son los mismos, o sea poder permitir o rechazar el acceso de acuerdo a donde se originó el acceso, y difieren en base a las expectativas de la política, es decir, que uso quiere darle la organización a eso. Deben complementarse entre ellos , no necesariamente tener reglas iguales pero pueden tenerlas. Aprovechar las fortalezas de

cada uno. Múltiples filtros en servicios críticos. Podemos hacer pruebas de salud del entorno.

Conceptos claves de auditoría de firewall

_ Por un lado la política de seguridad debería ser un documento escrito y en caso de no tenerlo podemos comenzar desde la configuración del firewall a documentar esa política. Establecer que debería ser permitido por defecto y rechazado por defecto. Determinar grupos, ya sea de equipos, redes o servicios similares que van a ser incluidos en las reglas. Y establecer zonas de seguridad por ejemplo grupos de equipos y/o redes, y además establecer la criticidad similar y requerimientos de acceso.

Temas fundacionales de firewall

Filtrado de paquetes: por lo general es el método más rápido y de baja seguridad.

Inspección con estado: rendimiento medio y seguridad media, esto es más difícil de controlar.

Proxy o gateway de aplicaciones: este método es más lento, y la seguridad es más alta.

Inspección profunda de paquetes: combina la inspección con estado y la tecnología IDS o el protocolo de detección de anomalías.

NAT (Network Address Translation): es lo que permite desde el exterior el uso de direcciones privadas en la Intranet (RFC1918), tenemos algunas variaciones como reenvío o redirección de puertos, relaciones muchos a uno (NAT oculto), relaciones uno a uno (NAT estático), y un pool de direcciones NAT.

Preparación de la auditoría

Política

_ Antes de comenzar la auditoría, debe definirse el propósito del firewall, es decir, lo que se espera que haga el firewall, y este debe estar basado en la política de seguridad. Si no hay política de seguridad, debe iniciarse una conversación con la dirección, donde un gran comienzo puede ser comenzar a escribir las reglas en el lenguaje del firewall, es decir, lo que vamos configurando lo plasmamos en la política.

Cuestiones a Definir

- Que información está protegiendo el Firewall.
- Cuáles son las expectativas del Firewall.
- Que riesgos está dispuesta a aceptar la organización.
- Que acciones son autorizadas.

Procedimientos

_ Similares a los del router:

- Control de cambios
- Copias de seguridad
- Administración de usuarios
- Política de contraseñas
- Actualizaciones de parches
- Construcciones seguras y estandarizadas de plataformas de firewall

Arquitectura de firewall

Arquitectura de Firewall

_ Debemos revisar la arquitectura de firewall, y preguntarnos como es la política de seguridad es soportada por la arquitectura, si utilizamos hubs, switches, etc. Debemos definir cómo la información debería fluir, qué flujo de datos está y no está autorizado. Los diseñadores de firewall y perímetro tienden a usar diagramas físicos, el auditor deber ser capaz de deducir el flujo de información y posiblemente un diagrama lógico desde un diagrama físico, que por lo general el lógico es inexistente.

_ La Arquitectura debe soportar la política de seguridad. Si la arquitectura del firewall está mal hecha, es poco lo que la base de reglas del firewall podrá hacer.

Diagrama lógico

_ El propósito de un diagrama lógico es mostrar el flujo de información. Permite definir qué información puede fluir hacia dónde y la política de seguridad define que y que no está autorizado. El propósito del firewall es controlar el flujo de información. Por ejemplo: arquitectura de un sitio de e-commerce necesita una red dedicada separada para el comercio b2b, sin embargo, el sitio de e-commerce debe ser capaz de comunicarse con lo corporativo (bases de datos).

Arquitectura y Entornos B2B

- Se confía también en los controles del otro negocio.
- Se debe mantener la documentación adecuada.
- Ver su política de seguridad.
- Se debe firmar un acuerdo.
- Política de acceso y controles de autenticación.
- Aplicaciones Propietarias: Métodos y Requerimientos de Seguridad, Controles de Autorización, Cifrado, Logging.
- Planes de Respuesta ante Incidentes.

- Arquitectura.
- Cifrado punto a punto.

Probando el Firewall

Plataforma - ¿Dispositivo (Appliance) o Sistema Operativo?

Dispositivo (Appliance): hardware que ya viene preconfigurado:

_ Ventajas:

- Normalmente viene completamente asegurada, no debemos preocuparnos del patch y demás.
- Diseñadas desde cero como dispositivos firewall.

_ Desventajas:

- Muchas son cerradas y propietarias, por lo que no podemos hacer muchos cambios en cuestiones más generales.
- Se debe confiar la seguridad al vendedor, ya que son ellos los que van a gestionar las actualizaciones y demás.

Sistema Operativo:

_ Ventajas:

- Mayor control sobre el aseguramiento del sistema.
- Muchos proveen el código fuente, por lo que podemos ver que se está haciendo en ese firewall.

_ Desventajas:

- Se debe tener mayor control sobre el aseguramiento del sistema.
- Grandes oportunidades de cometer errores.

Específico de la plataforma de firewall

_ Considerar el firewall específico que se está auditando, primero ver si hay opciones de configuración que se desvían de la base de reglas, y luego si hay características de seguridad que son específicas a la plataforma que se está auditando. Y los recursos que tenemos es revisar la documentación del fabricante.

Probando la base de reglas del firewall

Validación manual de la base de reglas

_ Podemos hacer una validación manual de las reglas, comenzando por revisando la base de reglas manualmente, podemos eliminar cualquier regla innecesaria, y armar equipo con el gerente de seguridad, administrador de firewall y arquitecto de redes, podemos combinar las reglas repetitivas, identificar cualquier regla no autorizada, y finalizar con la menor cantidad de reglas posible con el fin de que sea fácil su mantenimiento y prueba.

Consejos sobre la base de reglas

- El ordenamiento de las reglas debe ser mantenido tan simple como sea posible.
- Verificar reglas pasadas por alto o implícitas.
- Verificar qué reglas tienen el logging habilitado, sólo debe loguearse lo que es necesario.
- Todas las reglas deberían estar documentadas, para qué existe, quién la autorizó y cuando fue cambiada por última vez.

Auditar reglas de filtrado

_ Ver si estas reglas cumplen las reglas de filtrado la política y/o mejores prácticas, si están autorizadas y optimizadas, y recomendar cambios como sea necesario, donde siempre explicar la razón para el cambio y el beneficio del cambio.

Recomendaciones de reglas de base de firewall

_ Debería existir una política de “rechazado por defecto”, es decir, que es lo que no vamos a permitir para poder documentarlo en el firewall. Las reglas deberían ser específicas, no deben superponerse o duplicarse entre sí, no deberían contradecir a otras reglas y deberían ser utilizadas (aplicadas). Los servicios deberían estar configurados de manera segura. El logging debería ocurrir para las reglas donde sea necesario. Y todas las reglas deberían tener una justificación de negocio.

Validación técnica de la base de reglas

_ Validar la base de reglas del firewall desde el nivel de red a través del escaneo, o sea escanear a través del firewall y determinar que paquetes el firewall permite que pase. Escanear cada red desde cada interfaz, una laptop puede reemplazar sistemas en la red de servicio y escanear la red interna para simular un compromiso.

¿Qué herramientas utilizar?

_ Cualquier conjunto de herramientas debería, como mínimo, proveer las siguientes tres capacidades:

- Herramientas de mapeo de red: hping, nemesis, y nmap que nos permite identificar los dispositivos que tenemos en la red a la cual estamos conectados.
- Análisis de vulnerabilidades pasivo: es decir, aquellas que escuchan el tráfico de red, como wireshark, tcpdump, windump.
- Análisis de vulnerabilidades activo: nessus, openvas, que analizan vulnerabilidades de sistemas operativos basándose en los puertos.

Permiso de ejecución

_ La diferencia entre un hacker malicioso y un analista de seguridad es el permiso de ejecución. Siempre se necesita por escrito y por individuos autorizados.

Alertas y logging

Revisión de logs

_ Durante la ejecución de la auditoría, se produjo mucho "ruido" y habría que ver si fue detectado por los controles, es decir si:

- Fueron detectadas estas exploraciones.
- Fue alertada la gente apropiada.
- Se está registrando información adecuada.
- Se están perdiendo entradas en los logs.
- Se revisan con frecuencia las entradas en los logs.

_ Deberíamos estudiar y aprender las firmas (signatures) en los logs para poder mejorar las reglas del firewall.

NAC, detección de intrusiones y prevención de intrusiones

Sobre NAC (network access control)

_ Este controla los endpoints o los dispositivos finales, toma decisiones antes de permitir a los sistemas conectarse a la red, y las políticas controlan el acceso. Centraliza la administración de tecnologías como antivirus, prevención de intrusiones de equipos, autenticación, etc. Y el objetivo de NAC es:

- Controlar los ataques “zero-day”, es decir, aquellos que aparecen y no tienen un parche disponible.
- Permitir a los administradores la definición de políticas.
- Autenticar identidades de usuario.

Pasos de verificación de detección/prevención de intrusiones (NIDS/NIPS)

- Podemos usar nmap para verificar la detección de escaneo de puertos, probar múltiples velocidades.
- Usar un analizador de vulnerabilidades como nessus para verificar la detección de payloads.
- Usar fragrouter para probar la fragmentación de paquetes.
- Y combinar con un sniffer para verificar precisión.

Auditoría de IDS/IPS

_ Tendríamos que entender cuál es la arquitectura, ya sea los basados en Red versus los basados en Host. Ver si detectó el IDS/IPS la mayoría de los ataques, y si está la base de firmas actualizada. Ver si es utilizable el sistema de alertas, si envía mensajes o agrega registros a un archivo que nadie lee. Y si tiene sentido su ubicación en la red, saber si está conectado al puerto de un switch, aislado, donde está ubicado para ver si cumple con su función.

El informe de auditoría

Las normas

_ Las normas se usan como base para todas las cuestiones de auditoría. Hay dos tendencias legislativas en las que se basan las normas en la actualidad:

- La anglosajona, con pocas leyes y jurisprudencia relevante. Este modelo es el más apropiado para los cambios tecnológicos porque tiene pocas leyes y puede ir evolucionando rápidamente en el tiempo.
- La latina, basada en el derecho Romano, de legislación muy detallada.

_ El uso de los principios generalmente aceptados hace posible la adaptación suficiente de las normas a la realidad de cada época. Los organismos de normalización, homologación, acreditación y certificación tendrán que funcionar a un ritmo más acorde con las necesidades cambiantes, es decir que deben prever mecanismos para adaptarse rápidamente a los cambios, sobre todo en el mundo tecnológico donde los cambios son exponenciales.

La evidencia

_ La evidencia es la base razonable de la opinión del auditor y tiene una serie de calificativos:

- La evidencia relevante, cuando esta tiene una relación lógica con los objetivos de la auditoría, es decir, que acompaña y da sentido a lo que vamos a emitir como opinión.
- La evidencia fiable, quiere decir que es válida y objetiva. Esto quiere decir que no depende de quien la mire, siempre vamos a llegar a la misma conclusión.
- La evidencia suficiente, quiere decir que es de tipo cuantitativo para soportar la opinión profesional del auditor. Con cuantitativo decimos que tiene la cantidad suficiente para respaldar esa opinión, donde con menos no es válida y con más seria en exceso.
- La evidencia adecuada, que es de tipo cualitativo para afectar a las conclusiones del auditor. La calidad de información con la que contamos nos sirve de soporte a la opinión que estamos brindando.

Las irregularidades

_ En los organismos y las empresas, la dirección tiene la responsabilidad principal y primaria de la detección de irregularidades, fraudes y errores. Se debe establecer los mecanismos para detectar cualquier tipo de incidentes. Es necesario diseñar pruebas antifraude, que lógicamente incrementarán el costo de la auditoría, previo análisis de riesgos (amenazas, importancia relativa, etc). En el caso de detectar fraude durante el proceso de auditoría procede actuar en consecuencia, con la debida prudencia, sobre todo si afecta a los administradores de la organización objeto de auditoría.

La documentación

_ Hablamos de los papeles de trabajo, que son todos los documentos preparados o recibidos por el auditor que reúnen la información utilizada y los resultados de las pruebas efectuadas en la ejecución de su trabajo, junto con las decisiones tomadas para llegar a su opinión. Es básicamente lo que constituye a la evidencia. El Informe de auditoría tiene que estar basado en la documentación o papeles de trabajo. La documentación es además fuente en algunos casos en los que la corporación profesional puede realizar un control de calidad, o hacerlo a través de algún organismo oficial. Los papeles de trabajo pueden llegar a tener valor en los tribunales de justicia y acá es importante tener en cuenta las fechas.

_ Esto es justamente lo que define la característica registral del informe, es decir, que tanto en su parte cronológica como en la organizativa, con procedimientos de archivo, búsqueda, custodia y conservación de su documentación, cumpliendo toda la normativa

vigente, legal y profesional, acá hablamos de la cadena de custodia. Además, se incluirán el contrato cliente/auditor informático, declaraciones de la dirección, contratos que afecten al sistema de información, informes de asesorías jurídicas del cliente, informes sobre terceros vinculados y conocimiento de la actividad del cliente.

El informe de auditoría informática

_ Es la comunicación del auditor informático al cliente tanto del alcance de la auditoría (objetivos, período de cobertura, naturaleza y extensión del trabajo realizado) como de los resultados y las conclusiones. No existe un formato predeterminado, pero sí algunas recomendaciones sobre estructura y contenido. Se debe decidir previamente si el informe será largo o corto, con otros informes sobre aspectos más detallados y más concretos. El informe deberá ser claro, adecuado, suficiente y comprensible, con la utilización conveniente del lenguaje informático. Los puntos esenciales, genéricos y mínimos del informe de auditoría informática son:

- Identificación del informe.
- Identificación del cliente.
- Identificación de la entidad auditada.
- Objetivos de la auditoría informática.
- Normativa aplicada y excepciones.
- Alcance de la auditoría.
- Conclusiones: opinión favorable, opinión con salvedades, opinión desfavorable, opinión denegada.
- Resultados: informe largo y otros informes.
- Informes previos.
- Fecha del informe.
- Identificación y firma del auditor.
- Distribución del informe.

Organización del departamento de auditoría informática

Perfiles profesionales de la función de auditoría informática

_ El auditor informático debe ser una persona con alto grado de calificación técnica y al mismo tiempo estar integrado en las corrientes organizativas empresariales, es decir, tiene que estar conectado con las diferentes áreas y con la dirección de la empresa. Dentro de la función de auditoría informática, se deben contemplar las siguientes características para mantener un perfil profesional adecuado y actualizado:

- Las personas que integren esta función deben contemplar en su formación básica una mezcla de conocimientos de auditoría y de informática general como desarrollo informático, gestión del departamento de sistemas, análisis de riesgos en un entorno informático, sistemas operativos, telecomunicaciones, gestión de bases de datos, redes locales, seguridad física, etc.

_ A estos conocimientos básicos se deben añadir especializaciones en función de la importancia de los distintos componentes en el entorno empresarial. En la realidad actual, los sistemas de información requieren cada vez mayor control, se hace necesario para el auditor informático conocer técnicas de gestión empresarial, y sobre todo gestión del cambio, poder interpretar como los cambios que se van realizando en estos entornos afectan a los sistemas de producción y que nuevos riesgos se pueden introducir a través de estos cambios. El auditor informático debe tener siempre el concepto de calidad total, es decir, tener medidas a mano para ejercitar y asegurarse de que todos los cambios que se están efectuando no producen impactos negativos en los sistemas de la organización.

Funciones a desarrollar por auditoría informática

_ El auditor informático debe revisar la seguridad, el control interno, la efectividad, la gestión del cambio y la integridad de la información. La función de auditoría informática debe realizar, entre otras actividades:

- Verificación del control interno, tanto de las aplicaciones como de los sistemas informáticos, centrales y periféricos.
- Análisis de la gestión de los sistemas de información desde un punto de vista de riesgo de seguridad, de gestión y de efectividad de la gestión.
- Análisis de la integridad, fiabilidad y certeza de la información a través del análisis de las aplicaciones.
- Auditoría del riesgo operativo de los circuitos de información.
- Análisis de la gestión de los riesgos de la información y de la seguridad implícita.
- Verificación del nivel de continuidad de las operaciones.
- Análisis del estado del arte tecnológico de la instalación revisada y de las consecuencias empresariales que un desfase tecnológico pueda acarrear.
- Diagnóstico sobre el grado de cobertura que dan las aplicaciones a las necesidades estratégicas y operativas de información de la organización.

Organización de la función de auditoría informática

_ El auditor informático pasa a ser auditor y consultor del ente empresarial, en el que va a ser analista, auditor y asesor en materias de: seguridad, control interno operativo, eficiencia y eficacia, tecnología informática, continuidad de operaciones y gestión de riesgos, como dijimos antes este es una persona que va a interactuar tanto con la parte técnica de la empresa, como con la parte gerencial. Todo esto no solamente aplicado a los

sistemas informáticos (objeto de su estudio), sino de las relaciones e implicancias operativas que esos sistemas tienen en el contexto empresarial. La organización típica de auditoría informática, debe contemplar los siguientes principios:

- Su localización puede estar ligada a la localización de la auditoría interna operativa, pero con independencia de objetivos, de planes de formación y de presupuestos.
- La organización operativa típica debe ser la de un grupo independiente del de auditoría interna, con accesibilidad total a los sistemas informáticos y de información, e idealmente dependiendo de la misma persona en la empresa que el de auditoría interna, que debería ser el director general o consejero delegado.
- Los recursos humanos con los que debe contar el departamento deben contener una mezcla entre personas con formación en auditoría y organización, y personas con perfil informático.
- Este personal debe contemplar entre su titulación la de CISA (certified information systems auditor) como un elemento básico para comenzar su carrera como auditor informático.
- La organización interna de la función podría ser: jefe del departamento, gerente o supervisor de auditoría informática y auditor informático.
- El tamaño del área sólo se puede precisar en función de los objetivos de la función, pero se debería cubrir con especialistas en el entorno informático a auditar, gestión de bases de datos, comunicaciones y/o redes, riesgos y aplicaciones, y auditoría de sistemas de información.

Deontología del auditor informático y códigos éticos

_ Los principios a continuación son de alguna manera acuerdos comunes entre auditores:

Principio de beneficio del auditado

_ El auditor deberá conseguir la máxima eficacia y rentabilidad de los medios informáticos de la empresa auditada. Es decir, evaluar que se tiene instalado, que tan bien está funcionando y evitar hacer recomendaciones que impliquen demasiados costos y que no den beneficios reales a la empresa.

_ Cualquier actitud que anteponga intereses personales del auditor a los del auditado deberá considerarse como no ética. El auditor deberá evitar estar ligado en cualquier forma, a intereses de determinadas marcas, productos o equipos compatibles con los de su cliente. El auditor deberá establecer los requisitos mínimos, aconsejables y óptimos para la adecuación del sistema informático a la finalidad para la que ha sido diseñado, determinando en cada caso su adaptabilidad, fiabilidad, limitaciones, posibles mejoras y costos de las mismas.

Principio de calidad

_ El auditor deberá prestar sus servicios conforme a las posibilidades de la ciencia y medios a su alcance en condiciones técnicas adecuadas para el idóneo cumplimiento de su labor. Es decir, que tenga la capacitación suficiente, y que este aplicando todo dentro del mayor marco de calidad posible.

_ En los casos en los que la precariedad de medios puestos a su disposición impida o dificulten seriamente la realización de la auditoría, deberá negarse a realizarla hasta que se le garantice un mínimo de condiciones técnicas que no comprometan la calidad de sus servicios o dictámenes. Cuando durante la ejecución de la auditoría, el auditor considerase conveniente recabar el informe de otros técnicos más calificados sobre algún aspecto o incidencia que superase su capacitación profesional para analizarlo en idóneas condiciones, deberá remitir el mismo a un especialista en la materia o recabar su dictamen para reforzar la calidad y fiabilidad global de la auditoría.

Principio de capacidad

_ El auditor debe estar plenamente capacitado para la realización de la auditoría encomendada, teniendo en cuenta que, dada su especialización, a los auditados en algunos casos les puede ser extremadamente difícil verificar sus recomendaciones y evaluar correctamente la precisión de las mismas. Está relacionado con el principio de calidad. El auditor debe saber cómo transmitir lo que se evaluó a alguien que quizás no tiene la misma capacidad de comprensión.

_ Debe ser plenamente consciente del alcance de sus conocimientos y de su capacidad y aptitud para desarrollar la auditoría evitando que una sobreestimación personal pudiera provocar el incumplimiento parcial o total de la misma.

Principio de cautela

_ El auditor debe en todo momento ser consciente de que sus recomendaciones deben estar basadas en la experiencia que tiene adquirida, evitando que el auditado se embarque en proyectos de futuro fundamentados en simples intuiciones sobre la posible evolución de las nuevas tecnologías de la información. Es decir, el auditor no puede proponer una tecnología porque le pareció buena, y sugerirla a la empresa, sino que debe estar fundamentada su recomendación.

_ El auditor debe actuar con un cierto grado de humildad, evitando dar la impresión de estar al corriente de información privilegiada sobre el estado real de la evolución de los proyectos sobre nuevas tecnologías y ponderar las dudas que le surjan en el transcurso de la auditoría a fin de poner de manifiesto las diferentes posibles líneas de actuación en función de previsiones reales y porcentajes de riesgo calculados de las mismas, debidamente fundamentadas.

Principio de comportamiento profesional

_ El auditor deberá actuar conforme a las normas, implícitas o explícitas, de dignidad de la profesión y de corrección en el trato personal. Estas son cuestiones básicas de sentido común de cómo actuar profesionalmente en una organización, por ejemplo cuidar la forma de expresarse sobre lo que va observando para no atemorizar al cliente, y tratar de ser preciso.

_ Para ello deberá cuidar la moderación en la exposición de sus juicios u opiniones evitando caer en exageraciones o atemorizaciones innecesarias procurando transmitir una imagen de precisión y exactitud en sus comentarios que avalen su comportamiento profesional e infundan una mayor seguridad y confianza a sus clientes.

Principio de concentración en el trabajo

_ El auditor deberá evitar que un exceso de trabajo supere sus posibilidades de concentración y precisión en cada una de las tareas encomendadas, ya que la saturación y dispersión de trabajos suele a menudo provocar la conclusión de los mismos sin las debidas medidas de seguridad. Esto también está emparentado con el principio de calidad.

Principio de confianza

_ El auditor deberá facilitar e incrementar la confianza del auditado en base a una actuación de transparencia en su actividad profesional sin alardes científico-técnicos que puedan restar credibilidad a los resultados obtenidos y a las directrices aconsejadas de actuación. Relacionado con el principio de cautela.

_ Este principio requiere asimismo el mantener una confianza en las indicaciones del auditado aceptándolas sin reservas como válidas, a no ser que observe datos que las contradigan y previa confirmación personal de la inequívoca veracidad de los mismos.

Principio de criterio propio

_ El auditor durante la ejecución de la auditoría deberá actuar con criterio propio y no permitir que este esté subordinado al de otros profesionales, aún de reconocido prestigio, que no coincidan con el mismo. Tenemos las evidencias que nos respaldan y no debemos dejarnos influenciar por las opiniones de los demás que pueden ser contrarias.

_ En los casos en que aprecie divergencias de criterio con dichos profesionales sobre aspectos puntuales de su trabajo, deberá reflejar dichas divergencias dejando plenamente de manifiesto su propio criterio e indicando esa circunstancia.

Principio de discreción

_ El auditor deberá en todo momento mantener cierta discreción en la divulgación de datos, aparentemente inofensivos, que se le hayan puesto de manifiesto durante la ejecución de la auditoría. Se limita a entender cuál es el problema, plasmarlo en el informe y que la información no salga de ese contexto.

_ En los casos en que aprecie divergencias de criterio con dichos profesionales sobre aspectos puntuales de su trabajo, deberá reflejar dichas divergencias dejando plenamente de manifiesto su propio criterio e indicando esa circunstancia.

Principio de formación continua

_ Este principio impone a los auditores el deber y la responsabilidad de mantener una permanente actualización de sus conocimientos y métodos a fin de adecuarlos a las necesidades de la demanda y a las exigencias de la competencia de la oferta.

Principio de fortalecimiento y respeto a la profesión

_ La defensa de los auditados pasa por el fortalecimiento de la profesión de los auditores informáticos, lo que exige un respeto por el ejercicio de la actividad desarrollada por los mismos y un comportamiento acorde con los requisitos exigibles para el idóneo cumplimiento de la finalidad de las auditorías. El auditor debe hacer respetar su función dentro de la organización, pero también debe tener un comportamiento a ese respeto que exige.

_ En consonancia con el principio de defensa de la profesión de los auditores, estos deberán cuidar del reconocimiento del valor de su trabajo y de la correcta valoración de la importancia de los resultados obtenidos con el mismo.

Principio de no injerencia

_ El auditor deberá evitar injerencias en los trabajos de otros profesionales, respetar su labor y eludir hacer comentarios que pudieran interpretarse como despreciativos de la misma o provocar un cierto desprestigio de su calificación profesional, a no ser que, por necesidades de la auditoría, tuviera que explicitar determinadas idoneidades que pudieran afectar a las conclusiones o el resultado de su dictamen. Por ejemplo podemos encontrarnos con trabajos realizados por otros previamente y que no estaban bien, pero no podemos hacer comentarios despectivos sobre quien los realizo.

Principio de responsabilidad

_ El auditor deberá responsabilizarse de lo que haga, diga o aconseje, sirviendo esta forma de actuar como limitación de injerencias extraprofesionales. Va de la mano con el principio de cautela.

_ Es conveniente impulsar la formalización y suscripción de seguros, adaptados a las características de su actividad, que cubran la responsabilidad civil de los auditores con una suficiente cobertura a fin de acrecentar la confianza y solvencia de su actuación profesional. La responsabilidad del auditor conlleva la obligación de resarcimiento de los daños o perjuicios que pudieran derivarse de una actuación negligente o culposa, siendo aconsejable estipular a priori un tope máximo de responsabilidad sobre los posibles daños acorde con la remuneración acordada como contraprestación por la realización de la auditoría.

Principio de secreto profesional

_ El auditor tiene la obligación de guardar en secreto los hechos e informaciones que conozca en el ejercicio de su actividad profesional. Solamente por imperativo legal podrá decaer esta obligación. Esto no hace referencia solo a lo a la documentación o a lo que el auditor observe, sino también a lo que escuche.

_ Este principio obliga al auditor a no difundir a terceras personas ningún dato que haya visto, oído, o deducido durante el desarrollo de su trabajo que pudiera perjudicar a su cliente, siendo nulos cualesquiera pactos contractuales que pretendieran excluir dicha obligación. El mantenimiento del secreto profesional sobre la información obtenida durante la auditoría se extiende a aquellas personas que, bajo la potestad organizadora del auditor, colaboren con él en cualesquiera de las actividades relacionadas con la misma.

Auditoría de sistemas Windows

Línea de base de un sistema

_ Comenzamos en definir lo que constituye un sistema "seguro" y ver si el sistema analizado cumple con el estándar. Una vez que se tiene el sistema configurado, se toma una "línea de base" de esta configuración, por ejemplo podemos agarrar un equipo desde cero que por lo general viene con configuraciones por defecto que no son las más seguras, hacemos una configuración quizás basándonos en un CIS Benchmark, y una vez que tenemos la configuración, establecemos la línea de base, es decir, cada nuevo sistema que se incorpore debería adaptarse a esa configuración. Periódicamente se restablece la línea de base y se buscan cambios. Y esto permite monitorear problemas de seguridad a lo largo del tiempo.

Auditando en un dominio Windows

_ Muchas organizaciones tienen un entorno de dominio de algún tamaño, por lo general se usa el entorno de dominio para centralizar lo que es administración de usuarios y dispositivos y de permisos. Las buenas noticias del impacto de esto en la auditoria son:

- Muchas herramientas funcionan igualmente bien en un sistema "single standalone" o un dominio de 10.000 equipos. Un sistema single standalone puede ser un equipo de escritorio por ejemplo o un servidor aislado que no forma parte de ningún dominio y ahí es donde vamos a establecer algunas diferencias.

_ Sin embargo, hay diferencias entre "standalone" y entornos de dominio, ya que intentar consolidar y revisar los logs del event viewer de 10.000 equipos individuales Windows es una tarea desalentadora, e intentar auditar apropiadamente permisos de archivos en un entorno grande con muchos grupos anidados puede ser desafiante.

Equipo standalone: en estos toda la información está en el propio equipo, es decir, tenemos todos los usuarios, todos los archivos y permisos están definidos en el mismo archivo y los servicios también se controlan ahí mismo. Es más simple para obtener la información sobre el sistema individual porque tenemos todo en un solo lugar, pero es más difícil obtener información sobre un gran número de sistemas individuales, aunque es aún posible pero puede no ser practico.

Entorno de dominio: en estos la seguridad de un equipo está integrada con la seguridad del dominio, donde los usuarios que acceden a este equipo se autentican contra una entidad central que es el dominio. Lo que es más fácil en un dominio, es que mucha información está ahora centralizada a nivel de dominio, usuarios, grupos, política de contraseñas. Lo que es más difícil, la seguridad del cliente impactada por las configuraciones de dominio, por ejemplo políticas de grupo (group policies), seguridad aplicada en "capas" pueden complicar la auditoría (y la solución de problemas), problemas de confianza de dominio en grandes entornos multi-dominio, y la seguridad del sistema local debe aún ser abordada (por ejemplo, cuentas de dominio estándar como administradores locales).

Problemas no técnicos

- Separación de tareas, es decir cómo vamos dividiendo los perfiles y roles de los servidores y los administradores.
- Principio de mínimo privilegio, es decir, como controlamos quienes están en los grupos de administradores.
- Cuestiones de procedimiento, por ejemplo configuración de nuevas cuentas, cambios de contraseña, política de copias de seguridad, gestión de las configuraciones.

Esquema de auditoría

_ Similar a lo que vimos en Unix, seria tener:

- La información básica del sistema.
- Servicios en uso a nivel de red y a nivel local.

- Usuarios, grupos y contraseñas.
- Protección de datos.
- Seguridad del sistema operativo y aplicaciones.
- Auditoría y logging.

WMIC (Windows Management Instrumentation Command-Line)

_ WMIC es una herramienta que la encontramos básicamente en todas las versiones de Windows.

Donde funciona: corre desde Windows XP/2003 en adelante hasta el día de hoy (abrir una terminal de comandos, tipear “wmic” y presionar enter). Y tiene modos interactivo y no-interactivo, pero nos interesa el no-interactivo para automatizar los procesos de auditoría, porque tiramos el comando completo y nos tira la salida y podemos seguir avanzando con el scripting o con más comandos.

Que permite ver: casi todo, esencialmente expone casi cualquier configuración con el fin de solucionar problemas y automatizar. Por ejemplo configuración de red (NIC), configuraciones de escritorio, usuarios y grupos, estado de bloqueo de contraseñas, información de configuración del sistema, logs de eventos, etc.

Reportes: los hacemos esencialmente basados en texto, se pone el comando, el objeto y el verbo, no siempre devuelve una salida elegante, por lo general, inferior a 80 columnas y se muestra en más espacio que el de la pantalla. Existen otras opciones.

Auditoría avanzada de sistemas Windows

Objetivo de auditoría: identificar el sistema

Objetivo: obtener información básica del equipo siendo auditado, por ejemplo tipo de sistema operativo, versión (número de compilación, nivel de service pack, etc), información del sistema (tiempo de actividad, usuario/compañía registrado, etc), hardware básico (CPU, memoria, disco), las particiones deberían ser NTFS.

Propósito: es identificar aspectos clave del equipo auditado.

Preguntas a hacer

_ Si el sistema operativo en uso está al día, si el service pack (conjunto de paquetes de parches de seguridad) instalado es el último, si el disco está formateado con NTFS (para permitir el cifrado y control de registros), cuando fue parchado por última vez el equipo, si hay alguna aplicación no autorizada instalada, etc.

Información básica del sistema

Herramienta de línea de comando:

- Ver
- Systeminfo

Herramienta con interfaz gráfica:

- Winver
- msinfo32

Herramientas de terceros:

- PsInfo de Sysinternals

WMIC – Discos

_ Puntualmente nos interesa encontrar todos los dispositivos físicos, donde podría haber más dispositivos que los que están montados. Ademas tenemos que encontrar todas las particiones lógicas y verificar el formato, no hay buenas razones para FAT32. La información puede ser exportada en otro formato también.

Parches/actualizaciones del sistema operativo

Objetivo: asegurar que el sistema está actualizado con los parches de seguridad críticos.

Propósito: las actualizaciones de sistema y parches de seguridad protegen al equipo de ser comprometido. Muchos compromisos ocurren a través de vulnerabilidades conocidas que nunca fueron arregladas. El parchado es una de las maneras más fáciles de abordar este problema. Y el software no soportado puede ser “no parchable” y vulnerable por defecto.

_ Muchos ataques hoy en día a sistemas operativos vulnerables ocurren por no tener un sistema de parchado periódico y apropiado, y por otra parte si tenemos corriendo alguna aplicación que no tiene parches disponibles y sabemos que es vulnerable, deberíamos tener algún tipo de control compensatorio para poder cubrir esa vulnerabilidad.

Herramientas de auditoria: estado de parches

_ Tenemos algunas herramientas de auditoría para lo que es el estado de parches, como Microsoft Baseline Security Analyzer (MBSA), y las herramientas de administración de parches como Microsoft SUS / WUS / SMS, y herramientas de administración de parches de terceros que son pagas.

Tipos de parches de Windows

Service Packs: actualizaciones importantes que reúnen parches previos de seguridad y generales. Pueden incluir nuevas características.

Actualizaciones Críticas o Hotfixes: reparación para un problema crítico simple que afecta la seguridad o estabilidad del sistema.

Reparaciones QFE (Quick Fix Engineering): reparación para un problema específico simple, generalmente disponible solamente a través de Microsoft Support.

Componentes/servicios innecesarios

_ Deberíamos asegurar que solamente las características necesarias del sistema operativo están instaladas/corriendo. Algunas herramientas de auditoría son nmap, fport/openports, psservice/scquery/tasklist.

Componentes/Servicios: la instalación por defecto de cualquier SO es generalmente insegura. El vendedor (o administrador) puede instalar características para facilitar el uso, pero por lo general viene con servicios que no van a ser necesarios para nuestro uso. Los componentes seleccionados/no seleccionados durante o luego de la instalación, pueden ser instalados/desinstalados. Los servicios pueden ser instalados como parte de componentes o ser construidos dentro del SO, solamente algunos pueden ser desinstalados (remover componente), pueden ser iniciados, detenidos y deshabilitados

Servicios innecesarios: muchos servicios instalados por defecto no son requeridos para la operación. Los servicios pueden contener vulnerabilidades, mientras menos servicios tengamos corriendo mejor. Lo servicios no utilizados probablemente no serán parchados y deberían ser removidos o deshabilitados. Falsos servicios pueden indicar infección de malware.

Verificación de servicios: por un lado con servicios escuchando (puertos abiertos), desde afuera, desde un equipo externo (escaneo de puertos) o desde adentro, desde el mismo equipo (netstat, openports). Es bueno hacer ambos y correlacionar resultados ○ El escaneo desde afuera puede ser más confiable. Los servicios locales pueden no ser visibles para los externos. Un servicio es WMIC (service list brief).

Problemas con usuarios y grupos

- Solamente usuarios válidos deberían estar en el sistema.
- Los grupos tienen membresías apropiadas, es decir, que no todos estén en el grupo de administradores.
- Sin contraseñas en blanco.
- Política de contraseñas apropiada.
- Contraseñas “fuertes” en uso.

- Cuentas locales vs. cuentas de dominio.
- Uso de fecha de expiración.
- Limitar horarios de login.
- Considerar cuentas especiales.

Usuarios válidos

Objetivo: es tener usuarios autorizados. Asegurar que solamente cuentas de usuario válidas y activas están presentes.

Actividades de auditoría: tenemos algunas como:

- Net user
- Addusers
- DumpSec (Somarsoft)
- Consultas a Active Directory

Cuentas de usuario "Huérfanas": son cuentas no utilizadas que permanecen en el sistema. por ejemplo si el usuario deja la organización, o si el usuario tiene una cuenta que nunca usa.

Cuentas de servicio

_ Los servicios deben correr en el contexto de una cuenta de usuario. Si el servicio es comprometido, el atacante tiene privilegios del servicio. Muchos servicios corren como SYSTEM/LocalSystem o Local Service/Network Service. Muchas aplicaciones corren con acceso administrador o administrador de dominio por defecto lo cual es muy peligroso. Deberían correr con mínimos privilegios requeridos. Y verificar cuentas usadas por varios servicios.

Grupos apropiados

Objetivo: verificar las membresías de grupo, que sean apropiadas y hacen cumplir el mínimo privilegio.

Actividades de Auditoría:

- Dsquery
- DumpSec (Somarsoft)

Membresías de grupo: los Grupos se utilizan para conceder permisos y privilegios. Las membresías restringen (o conceden) acceso a recursos. Los grupos sensibles deberían ser monitoreados. Y los grupos de administrador deberían tener muy poca gente.

Contraseñas fuertes

Objetivo: es que las contraseñas son administradas de manera apropiada. Las contraseñas que están en uso (no están en blanco), son “fuertes”, están administradas con una buena política, están protegidas por cifrado fuerte.

Actividades de auditoría:

- net accounts, DumpSec.
- Herramientas de evaluación de contraseñas.
- Verificación de registro/configuraciones.

Requerimientos de autenticación: el acceso a sistemas debería ser restringido/controlado. Windows requiere usuario/contraseña, pero la contraseña puede estar vacía o ser débil. Problemas clave de auditoría son, la existencia de la contraseñas, cambios de contraseña frecuente requeridos, cumplimiento del uso de Contraseñas “fuertes”, utilización de buen cifrado para proteger contraseñas.

Auditoría de la Dirección de Informática

Proceso de dirección informática

_ Es necesaria la evaluación independiente de la función que gestiona las tecnologías de la información, es decir, tengamos un organismo independiente que pueda auditar cómo funcionan estas actividades dentro de la dirección de informática. Las actividades básicas del proceso de dirección son:

- Planificar
- Organizar
- Dirigir
- Controlar

Planificar

_ Cuando hablamos de planificar, es principalmente ver como se cumplen los objetivos para prever la utilización de tecnologías de información. Esta actividad se basa fundamentalmente en un plan, que es el Plan Estratégico de Sistemas de Información, este es el plan base para todas las actividades que va a hacer la organización y debe estar alineado con los objetivos de la empresa, no es responsabilidad exclusiva de la dirección de informática, aunque ésta debe impulsar una planificación adecuada y a tiempo. El entorno de la empresa define el plazo del plan. Por lo general, el plazo es de 3 a 5 años.

_ Ademas de este plan tenemos otros planes relacionados como el plan operativo anual, que cuenta con los sistemas de información a desarrollar, cambios tecnológicos previstos,

recursos y plazos necesarios, además tenemos el plan de dirección tecnológica, también el plan de arquitectura de la información, y el plan de recuperación ante desastres.

Organizar y dirigir

_ Esto se organiza para estructurar los recursos, flujos de información y controles para alcanzar los objetivos propuestos en la planificación. Para esta actividad se establece lo que es el Comité de Informática, impulsado por la dirección de informática, como lugar de encuentro entre los informáticos, los usuarios, y gerentes. Por lo general está presidido por el director con mayor experiencia dentro de la empresa, las grandes áreas usuarias deberían estar representadas al nivel de sus directores, y el director de auditoría Interna debería ser miembro del comité porque es uno de los principales interesados.

Funciones del comité de informática: la principal es la aprobación del Plan Estratégico de Sistemas de Información, es decir esta es quien aprueba el plan, no lo crea. También la aprobación de las grandes inversiones en tecnología de la información, fijación de prioridades entre los grandes proyectos informáticos, es un vehículo de discusión entre la Informática y sus usuarios donde los usuarios plantean sus inquietudes desde el punto de vista de los sistemas informáticos, y vigila y realiza el seguimiento de la actividad del departamento de informática. En definitiva el departamento de informática debería rendirle cuentas a este comité y no a la dirección general, porque en definitiva este comité es el que hace de intermediario.

Posición del departamento de informática en la empresa: debería estar lo suficientemente alto en la jerarquía y contar con masa crítica suficiente. Suele depender de la dirección general. En grandes organizaciones, el director de informática es miembro del comité de dirección. El auditor evalúa la equidad de trato a los diferentes departamentos de la empresa.

Descripción de funciones y responsabilidades del departamento de informática

Segregación de funciones: que las funciones estén descritas y sus responsabilidades claramente delimitadas y documentadas, con el fin de que cada persona que pertenece al equipo sepa cuál es su tarea y que se espera de él. Por otro lado evita que un individuo pueda trastornar un proceso crítico.

Aseguramiento de la calidad: el departamento debería tener una función organizativa de aseguramiento de la calidad independiente. Este ente responde directamente a la dirección de informática. Es decir, es una rama de la dirección de informática que audita el funcionamiento del resto de los componentes del departamento de informática.

Estándares de funcionamiento y procedimientos

_ Deberíamos contar con la descripción de los puestos de trabajo, que estaba asociado a lo de segregación de funciones. Deben existir estándares de funcionamiento y procedimientos que gobiernen la actividad del departamento de informática y sus relaciones con los departamentos usuarios. Deben estar documentados, actualizados y ser comunicados adecuadamente a todos los departamentos afectados. Y deben existir descripciones documentadas de los puestos de trabajo dentro de informática.

Gestión Económica

Presupuestación: el departamento de informática debe tener un presupuesto económico, normalmente en base anual. Esto debería estar ligado al plan operativo anual.

Adquisición de bienes y servicios: los procedimientos que el departamento de informática siga para adquirir los bienes y servicios deben estar documentados y alineados con los procedimientos de compras del resto de la empresa.

Medida y reparto de costos: la dirección de informática debe en todo momento gestionar los costos asociados con la utilización de los recursos informáticos: humanos y tecnológicos. El departamento de informática no debería favorecer a un área en particular sino que debería asegurarse que los costos estén repartidos adecuadamente entre todas las áreas de la organización.

Controlar

_ Esta se ocupa principalmente en:

Control y seguimiento: la dirección tiene la obligación de controlar y efectuar un seguimiento permanente de las distintas actividades del departamento.

Cumplimiento de la normativa legal: la dirección de informática debe controlar que la realización de sus actividades se lleva a cabo dentro del respeto a la normativa legal aplicable.

Auditoría del Área de Desarrollo

Funciones del área de desarrollo

_ Las funciones son principalmente:

- La planificación del área y participación en la elaboración del plan estratégico de informática. Es decir, tienen parte en ese plan a largo plazo.
- Desarrollo de nuevos sistemas.

- Estudio de nuevos lenguajes, técnicas, metodologías, estándares, herramientas, etc, relacionados con el desarrollo y adopción de los mismos. Es decir, como incorporamos tecnologías nuevas y evitamos la obsolescencia de los sistemas que estamos soportando.
- Establecimiento de un plan de formación para el personal asignado al área. El personal se tiene que adaptar al cambio tecnológico e involucrarlos en los que son mejores prácticas de desarrollo incluyendo temas como seguridad y calidad desde el principio.
- Establecimiento de normas y controles para todas las actividades que se realizan en el área y comprobación de su cumplimiento. Esto es a nivel directivo, donde el área tiene que tener reglas claras definidas para sus funcionamientos.

Planteamiento y metodología

_ Se abordará la auditoría del área en dos grandes apartados:

- Auditoría de la organización y gestión del área de desarrollo.
- Auditoría de proyectos de desarrollo de sistemas de información.

_ Se aplicará la metodología propuesta por ISACA basada en la evaluación de riesgos. A través de las pruebas, se determinará cuáles son los riesgos no cubiertos, en qué medida lo son y qué consecuencias se pueden derivar de esa situación. En definitiva hacemos referencia a las pruebas que se aplican para los controles.

Auditoría de proyectos de desarrollo de S.I. (Fases)

_ La auditoría de un proyecto de desarrollo se puede hacer en dos momentos distintos, a medida que avanza el proyecto y una vez concluido. Las fases a auditar serán:

- Aprobación, planificación y gestión del proyecto
- Auditoría de la fase de análisis
- Auditoría de la fase de diseño
- Auditoría de la fase de construcción
- Auditoría de la fase de implementación

Auditoría de aplicaciones Web

Realidad actual

_ Las aplicaciones web vulnerables son consistentemente uno de los 10 problemas más importantes en la actualidad. Las pruebas pueden ser peligrosas causando problemas inesperados en la aplicación o infraestructura. La tecnología es aún "nueva" porque tenemos por un lado que http y html tienen muchos años en el mercado, son conceptos maduros, y por otro lado tenemos nuevos avances y técnicas (frameworks) que continúan evolucionando para apalancar estas tecnologías. Las aplicaciones que vemos en el mercado, por lo general comienzan resolviendo algunos problemas internos en las organizaciones y luego terminan publicándose en el exterior para resolver problemas de manera masiva. Es decir, las aplicaciones externas típicamente comienzan su vida como "hacks" internos. Con Google Hacking Database podemos ver información de vulnerabilidades en sitios o aplicaciones (vulnerabilidades conocidas, contraseñas inyección SQL, números de tarjetas de crédito, información de clientes, explotable remotamente (puntos de apoyo)).

Funcionamiento Básico y Entrenamiento

_ En este punto hablamos puntualmente de como los desarrolladores aprenden su oficio. Por lo general pueden utilizar técnicas o lugares como la universidad, aunque este no siempre es el caso, también tenemos los seminarios de entrenamiento que son hoy por hoy lo más común así como también los cursos online en diferentes plataformas. Ademas tenemos que saber la fuente de información que se utiliza para capacitarse como códigos de ejemplo o aplicaciones de terceros, en donde no sería de extrañar tener aplicaciones web defectuosas.

Buenas prácticas

_ Acciones simples que podemos hacer para crear aplicaciones seguras:

Validación de entradas: verificar todo lo que ingresa a la aplicación para comprobar que eso conforma lo que la aplicación está esperando (tipos de datos, caracteres, etc). No confiar en nada, desde el cliente, desde la base de datos, y cualquier cosa externa a la aplicación.

Desinfección: identificar todas las formas de entrada que serán necesarias para que la aplicación funcione (letras, números, etc.). Todo debe ser desinfectado a través de un filtro estándar, es decir, deberíamos intentar que estos métodos de filtrado estén en un solo lugar para poder reducir la posibilidad de errores y que no se esté copiando y pegando a lo largo de los diferentes módulos Descartar todo lo que no identificamos previamente, todo lo que no cumplió on los filtros.

Control de errores: deberíamos apuntar a manejar todas las condiciones de error, es decir, cosas que se pueden predecir que van a suceder por ejemplo errores de base de datos, errores de red, y además obtener y manejar lo inesperado también.

Administración robusta de sesiones: definimos antes que es una sesión, esta es una instancia única de un usuario a través del curso de su interacción con la aplicación, es decir, el usuario inicia sesión con la aplicación donde se le da un identificador único que lo identifica y que por lo general esta acotado en el tiempo (variable). Los IDs de sesión generan trazabilidad, es decir, están asociados unívocamente a un usuario. Y deberíamos generar IDs de sesión fuertes, aleatorios (para que el atacante no lo identifique), fuerte asociación con el cliente y considerar que sean a prueba de falsificaciones.

Mediación completa: hace referencia a que hay solamente una manera de ingresar y salir de una aplicación, es decir, que todo el tráfico o flujo pasa a través de un único punto de control reduciendo las posibilidades de que un usuario acceda a la información de otro. Consideramos un punto único de ingreso y que nada más sea accesible en el sitio sin pasar a través de este punto de acceso, y esto nos permite la reutilización de código confiable.

Solución de múltiples capas: para tener seguridad robusta, se recomienda usar tres capas que son la capa de presentación relacionada al front end, aplicación (o negocio) relacionada al back end, y la capa de persistencia que sería la base de datos u otro medio de almacenamiento. La seguridad es más baja con dos o menos capas (por ejemplo node.js que tiene la capa de presentación y aplicación juntas), donde la complejidad y el costo puede restringir las implementaciones a dos capas.

Dos capas vs tres capas

_ En una solución de dos capas, la capas de presentación y aplicación trabajan juntas, y nos preguntamos si se almacena información sensible en la capa de persistencia, es decir, la base de datos, como contraseñas o información del cliente. La pregunta a hacerse es dónde están las credenciales de base de datos, es decir, si están expuestas de alguna manera en la capa de aplicación en una solución de tres capas o entre las capas de presentación y aplicación en una solución de dos capas. Lo correcto sería que las credenciales estén almacenadas en algo externo y evitar de que estén en el código. Si se cifran los datos en la base de datos, tenemos que ver dónde están las claves de cifrado y hacer que no estén expuestas.

Conceptos básicos de aplicaciones Web

HTTP: es un protocolo, es sólo texto, viene de HyperText Transfer Protocol, donde los datos binarios pueden ser envueltos en HTTP como por ejemplo ejecutables, imágenes, etc. Es además un método de comunicación, los clientes solicitan páginas de servidores usando HTTP y los servidores responden con HTML (lenguaje de marcado y no es un protocolo) envuelto en encabezados HTTP, y en definitiva así como podemos enviar datos binarios, podemos enviar texto que es interpretado por los navegadores a través de este protocolo.

Formularios Web: para estos habitualmente se utiliza la etiqueta <FORM> para agrupar entradas. El método define como los formularios son enviados. Cuando hablamos de front end por lo general tenemos GET para obtener información o POST para enviar información pero se pueden utilizar ambos para ambos propósitos. La acción define que hacer cuando es enviado. Las etiquetas <INPUT> incluyen varios tipos como campos de texto, contraseñas, áreas de texto, botones de envió, etc. También tenemos cuadros de selección, listas desplegables, etc.

_ HTTP viene en una cantidad de sabores como TRACE, HEAD, GET, POST, PUT, DELETE, etc. Pero front end utiliza estos dos principalmente:

- GET: lo que hace es enviar los parámetros en la URL. Tiene un límite de 255 caracteres. Toda la entrada es incluida en la URL. En aplicaciones vulnerables, en el login, utilizando GET exponemos el usuario y la contraseña en la URL lo cual es una mala práctica.
- POST: sin límite duro (configurable), es decir, no tiene un límite de caracteres. Por otro lado toda la entrada es incluida en el cuerpo de la petición HTTP con lo cual si estamos usando certificados es prácticamente imposible ver el cuerpo del mensaje. Ademas tenemos que la URL que puede contener parámetros (información), encabezados HTTP (por ejemplo de autenticación), el identificador del Navegador (ya sea en la respuesta o en la petición), poder saber quién es el referente (saber de dónde viene la petición) y también poder enviar cookies.

Métodos REST (REpresentational State Transfer): las aplicaciones RESTful usan CRUD (modificaciones altas y bajas más las consultas) implícitamente, y a continuación vemos la relación de los métodos REST con los métodos HTTP:

- Create: POST
- Read: GET
- Update: PUT
- Delete: DELETE

_ Los servidores web no los implementan (en front end). Emulan CRUD a través de elementos ocultos. Solamente usan GET y POST en el front end.

Cookies: las cookies son simplemente texto que se utilizan para almacenar información del usuario actual. El servidor envía un nombre de cookie con un valor arbitrario al cliente que en este caso sería un navegador u otra aplicación, el cliente mismo lo almacena y luego envía la cookie con cada petición, es decir, comparte una especie de secreto al servidor y se lo da al cliente para que el servidor sepa que todas las peticiones que vienen con esa cookie son de un cliente en particular. Y es posible enviar más de una cookie dentro de una petición HTTP. Estas son configuradas con el encabezado Set-Cookie para las peticiones, se les asigna un nombre, un dominio, un path, Secure (nos obliga a usar SSL) y expires que nos da la fecha de expiración de la cookie. Las cookies son solo piezas de texto arbitrario. El flujo de una cookie es el siguiente:

- El navegador envía una petición.
- El servidor establece y devuelve una cookie.
- El navegador envía la cookie de vuelta en cada petición.

Los contenidos de la cookie pueden ser rastreados si no está marcada, si no usamos SSL. Tenemos que tener en cuenta también el dominio si es de un equipo especifico o un dominio entero, y también el path si por ejemplo es de un servidor compartido o una aplicación en particular.

- Cookies persistentes: son las que se almacenan en el cliente, este es el comportamiento típico, y se almacenan en disco para uso futuro de peticiones, esto permite también que sean de fácil manipulación.
- Cookies no persistentes: no pensadas para ser almacenadas en disco. Los desarrolladores piensan que no pueden ser cambiadas pero si pueden serlo porque pueden ser interceptadas por un proxy. La manipulación es más difícil, pero no imposible.

SSL/TLS: Secure Sockets Layer y Transport Layer Security (más enfocado en la capa de transporte). Es la tunelización para HTTP y nos asegura que el tráfico sea seguro. Solamente cifrado en tránsito, donde desde el origen al destino viaja por un canal cifrado y no puede ser visto por terceras partes. Llega en texto plano a los endpoints. Este provee cifrado no seguridad.

AJAX: significa Asynchronous JavaScript and XML. Google maps es un ejemplo perfecto de esto, las páginas cargan usando HTTP, JavaScript pide información de manera asíncrona, típicamente usando XML como transporte, y la página se actualiza sin recargar.

CSS: Cascading Style Sheets, es lo que se usa para estandarizar el formato visual de la página. Es un estándar Web, que permite un control sobre el diseño, aplica estilos a elementos en páginas (definimos clases con colores, tamaños de fuente, etc), esto nos permite consistencia y permite a los usuarios navegar usando sus propios estilos. No es importante para analizar vulnerabilidades porque simplemente define estilos.

OWASP: Open Web Application Security Project, esta es la fuente de referencia para lo que es seguridades de aplicaciones Web. En si sitio encontramos cosas como guías de desarrollo, recursos con mejores prácticas, base de datos de vulnerabilidades web, herramientas de aprendizaje, herramienta de auditoría OWASP ZAP.

- OWASP ZAP: esencialmente un "Hombre en el medio" entre el navegador web y el servidor. Permite ver información históricamente, podemos analizar y almacenar las peticiones, interceptar y modificar información, soporta SSL y mucho más.

BURP Suite: es el mismo concepto que OWASP ZAP, es un proxy que identifica peticiones. Tiene algunas características mejor y tiene versiones comerciales.

Seguridad de Servidores

Indexación de directorios: muchos de los servidores webs que utilizamos tienen la indexación de directorios. Esto no es una vulnerabilidad si no que es una forma de que el atacante pueda ver más información de lo que debería ver. Es el acceso a un directorio en lugar de una página web. Deberíamos comprobar si la configuración coincide con la postura de seguridad y el propósito que esperamos nosotros que tenga ese servidor. Entonces es más divulgación de información que una vulnerabilidad. Este elemento es cuando se expone el contenido de un directorio accedido a través de HTTP.

Encabezados: siguen siendo una exposición, son ideales para identificar exploits, encontrar sitios donde el mantenimiento no es frecuente, también para encontrar complementos o plugins.

Robots.txt: es un archivo que sea agrega comúnmente en los servidores web con el fin de que los spiders de los buscadores no indexen los directorios o páginas que se especifican. Se usan para configurar restricciones pero regalan información extra. Elemento destinado a controlar la indexación. Método para restringir el acceso. Se busca que esto este oculto con un meta tag.

Reuniendo Elementos: tenemos elementos de la lista de verificación o check list:

- Contenido por defecto (ejemplos de código).
- Indexación de directorios que este deshabilitada.
- Configuración base segura, examinar contenido oculto.
- Sistema operativo seguro.
- Despliegue de red seguro.

Pruebas de Configuración

¿Cómo funcionan los analizadores?

_ La estrategia típica es identificar primero el servidor, esto se hace a través de una huella dactilar (fingerprint) y nos ayuda a identificar sistema operativo, servicio web, los complementos. Y podemos hacer prueba por problemas del servidor basado en los resultados, esto acelera las cosas. Podemos hacer que el que me ataque vea información incorrecta.

Analizadores de propósito general

_ Tenemos analizadores como Nessus y similares aún son útiles para chequear vulnerabilidades de sistemas operativos considerando puertos, fingerprints, etc. Son buenos para comprobar la seguridad en general de un servidor. Van a encontrar material básico por defecto, por ejemplo si está expuesto el puerto 80, etc. Pero es como usar un martillo para un tornillo, es decir, es una herramienta para una aplicación web que a lo mejor tiene muchas particularidades. Entonces los analizadores web generalmente realizan muchas más pruebas con mayor precisión.

Advertencia automatizada

_ El análisis automatizado es muy bueno, rápido y fácil, muchas herramientas para apuntar y hacer click, y tenemos reportes fáciles de leer. Sin embargo pueden encontrarse solamente vulnerabilidades conocidas, pueden usarse solamente técnicas conocidas y buscar por patrones conocidos. Aún no hay mejor tester de aplicaciones web que una persona bien entrenada que vaya paso a paso testeando diferentes controles y no basándose en un herramienta que mira algo muy superficialmente.

Fuzzers

_ Fuzzing es una táctica antigua con nuevas herramientas. Es básicamente arrojar diferentes entradas aleatorias a una aplicación y así vamos midiendo los resultados que nos dan las mismas y ver si encontramos alguna vulnerabilidad. Encuentra todos los orificios de entrada que tiene una aplicación. Automáticamente pega todo tipo de basura en todos ellos al mismo tiempo. Algunas herramientas hacen esto son Nessus o OWASP ZAP. Aún no hay nada como un ser humano para algunas tareas.

Introducción a la autenticación

Objetivos de auditoría: determinar si el mecanismo de autenticación es seguro.

Controles: tenemos algunos controles como ver que la implementación sea adecuada, el cifrado y las credenciales fuertes.

Métodos comunes de autenticación

Autenticación básica HTTP: está prácticamente en desuso. Tiene como fortalezas que es fácil de implementar. Es prácticamente enviar un encabezado en una petición. Tiene como debilidades que no es fácilmente borrable desde el navegador, no tiene cifrado, es fácil para fuerza bruta y actúa como ID de sesión, confunde el secreto de largo plazo con secreto de corto plazo. Las mejores prácticas en el caso de que estemos obligados a usar esto, es cifrar todo el tráfico durante y luego de la autenticación.

Autenticación basada en formularios: tiene como fortalezas que es fácil de Implementar, es un balance razonable entre seguridad y conveniencia para los usuarios. Como debilidades tiene potencial de fuerza bruta, mala configuración conduce a exponer credenciales, tiene las mismas debilidades que nombre de usuario/contraseña, no viene cifrado por defecto (va por HTTP pero no acompañado por SSL). Mejores prácticas en esto serian usar HTTP POST para enviar credenciales de usuario porque van en el cuerpo del mensaje y no en la URL, también que el envío de credenciales de usuario vía cifrado (ej. SSL), en los campos de contraseña usar type=password donde la contraseña no sea visible al usuario, y considerar el uso de Tokens (por ejemplo, Google Authenticator) para tener un doble factor de autenticación.

Certificados de cliente: como fortalezas es ampliamente seguro, permite el "no repudio", es decir, si nosotros tenemos un certificado y se envía una petición a nombre nuestro, no podemos decir que nosotros no fuimos porque ese certificado se nos dio directamente a nosotros, también tenemos la confidencialidad porque hace una autenticación mutua justamente garantizando que entre el cliente y el servidor estuvo únicamente compartida esa información. Como debilidades tenemos la movilidad y la interoperabilidad limitada ya que es costoso el intercambio de certificados, y otra debilidad es la administración, por ejemplo si le estamos dando esto a usuarios finales debemos anular los certificados así como remover las cuentas que usan en el entorno. Como mejores prácticas podemos mencionar que es la mejor forma de autenticación para B2B (negocio a negocio) y necesidades de alta seguridad, y también usar token de hardware (por ejemplo, tarjeta inteligente) para incrementar movilidad.

Autenticación NTLM: se usa principalmente en sistemas Windows. Es una solución de Single Sign-On (SSO) para aplicaciones ASP.NET en IIS (Internet Information Server). Solamente útil para aplicaciones de Intranet, requiere el uso de IIS, no es un estándar, aprovecha la autenticación basada en hashes, es vulnerable con lo cual no es muy recomendable, y es segura cuando el ID de sesión está protegido. Se recomienda que cuando usemos esto que el servidor este fuera del dominio.

Carteles de advertencia

_ Si quisiéramos hacer un ataque de fuerza bruta, este cartel nos debería desalentar porque si no vamos a tener muchos ojos encima nuestro esta es una medida más, no es absolutamente preventiva pero puede reducir bastante el nivel de ataque.

Recolección de nombres de usuario

- Amenaza: un tercero malicioso podría recolectar nombres de usuario válidos. A veces se revela demasiada información, por ejemplo si tenemos un login fallido y nos sale un mensaje diciendo que la contraseña no es válida, el mensaje no sería lo más apropiado, por ejemplo en la creación de cuenta (Gmail) si ponemos un usuario existente y nos dice el mensaje justamente que existe esa cuenta entonces podríamos intentar hackear esa cuenta, también es otro riesgo el restablecimiento de contraseña.
- Recomendación: Indicar a través de un solo mensaje que la autenticación es incorrecta. Solamente autenticación fallida y no que es lo que fallo en definitiva.

Técnica de auditoría para recolección de nombres de usuario: probar en fallar intencionalmente intentos de inicio de sesión, cubriendo cada escenario posible, ser cuidadoso de no bloquear cuentas, eso ocurre después, también si es posible, probar cada lenguaje posible (por ejemplo español e inglés). Podemos tener un documento registrando todos los escenarios.

Ataques de contraseña por fuerza bruta

- Amenaza: algunos sitios permiten un número ilimitado de intentos fallidos de inicio de sesión (no bloquean la cuenta), es decir, nunca bloquean la cuenta.
- Impacto: es en la autorización. Cuentas comprometidas a través de ataques de fuerza bruta contra la contraseña.
- Recomendación: bloquear las cuentas (cuidadosamente).

DoS (denegación de servicio) por fuerza bruta - bloqueos de cuentas: algunos sitios hacen cumplir el bloqueo de cuentas luego de un número específico de intentos de inicio de sesión fallidos.

- Amenaza: un gran número de cuentas de usuario pueden ser bloqueadas usando herramientas automatizadas, la autenticación básica HTTP y la basada en formularios son fácilmente atacadas (aún con SSL).
- Impacto: es en la disponibilidad donde nos hacen un ataque DoS.
- Recomendación: usar bloqueos de velocidad (iniciar sesión 30 segundos después).

_ THC Hydra es una herramienta multiplataforma que tenemos para hacer ataques de fuerza bruta con el propósito validar la autenticación web. No es bueno usar estas herramientas sin permiso.