educación IT
CODE YOUR FUTURE

Introducción a la Ciberseguridad

Módulo 2

Gestión de vulnerabilidades

Gestión de vulnerabilidades

Una *vulnerabilidad* es una debilidad o fallo en un sistema de información, que pone en riesgo la seguridad de la información; lo cual permite que un atacante pueda comprometer la integridad, disponibilidad, o confidencialidad de ésta.

Todas las tecnologías que utilizamos fueron creadas por personas y son **propensas a tener fallas**. Regularmente, distintos investigadores de seguridad encuentran vulnerabilidades y las reportan, para que sean solucionadas. Por ello, es crucial tener bien identificado el *hardware* y *software* que usamos, para conocer si pueden ser susceptibles de sufrir alguna vulnerabilidad con el tiempo. A toda persona que trabaje en seguridad le tomará una cuota considerable de tiempo estar informada sobre las nuevas fallas de seguridad que aparecen, y sobre todo, estar atenta a ver si representan un riesgo en la infraestructura que se controla.

Por dicho motivo, otra de las actividades de una estrategia de seguridad debe ser la de **mantener los sistemas operativos y aplicaciones actualizados/as**, esto significa que debe contar siempre con **la última versión instalada**.

Actualización del sistema o aplicación

Que un sistema o aplicación tenga su **última versión**, **asegura que se han corregido todos aquellos errores "descubiertos"** que pueden provocar que el sistema/aplicación no funcione correctamente. Esta tarea se vuelve esencial dentro de la vida de una organización, y en especial sobre los dispositivos tecnológicos, dado que **si un sistema operativo o aplicación no está con su última versión es vulnerable**, es decir, un atacante podría averiguar qué versión de sistema o aplicación está corriendo, notar que esa misma tiene una vulnerabilidad que no fue corregida y explotarla, es decir, utilizar esa falla para poder acceder y manipular al sistema en cuestión.

En tal sentido, **la actualización del S.O. / aplicación es una actividad que deberá realizarse de forma cíclica**, estableciendo cada cuánto se ejecutará, no sólo el control de las versiones, sino también su actualización en caso de corresponder, y buscar así tener asegurados todos los sistemas y las aplicaciones existentes. Ahora bien, hay que definir **cada cuánto se deberá realizar este control de las versiones**. Esta decisión debe ser establecida por quién tome las decisiones de seguridad, pero un buen parámetro para definirlo podría ser realizar el control **una vez por mes**.

Habitualmente notamos que en los dispositivos móviles, las **aplicaciones** que tenemos instaladas se actualizan para corregir errores e incluir nuevas funcionalidades si las hay. En el caso de los S.O. y aplicaciones de computadoras funciona de la misma forma.

Al instalar un **sistema**, por lo general, éste establece una conexión con Internet para corroborar si hay actualizaciones, descargarlas e instalarlas. No obstante, si en una organización, todos los equipos de la red realizan esta tarea, o sea, salir a Internet a ver si existe alguna actualización y descargarla, se produciría una **degradación de la capacidad de la red**, ya que cada dispositivo está descargando el mismo archivo.

Entonces, habitualmente lo que se hace es **no permitir a los dispositivos de la red que se actualicen.** Esta medida suele ser realizada al prohibir la salida a Internet por *Firewall*, al deshabilitar el servicio de actualización, etc.

La forma de solucionar este problema, es que **un equipo de la red sea el que descargue las actualizaciones correspondientes**, tarea suele realizar algún administrador; y luego durante un **horario en en el que no se realicen actividades laborales, se proceda a implementar las actualizaciones** en los equipos de la red que haya que actualizar.

Esto mejora la performance de la red ya que no se satura de la misma información y por otro lado, se hace en un momento del día que no moleste a los usuarios que dependen de ella.

Escáneres de vulnerabilidades

Una forma de conocer cuáles equipos de nuestra red tienen errores o si son vulnerables por falta de alguna actualización, es **mediante el uso de programas que descubran estas mismas fallas.** Hay varios que automatizan dicha tarea y resultan muy útiles para los analistas de seguridad, ya que facilita la tarea de relevamiento.

Los **sistemas de escaneo de vulnerabilidades** ayudan a una organización a identificar y remediar las vulnerabilidades dentro de su ambiente de tecnología, antes de que los ciberdelincuentes / atacantes puedan obtener acceso a modificar o destruir información confidencial. Existen varios muy conocidos, pero entre los más usados, encontramos los siguientes: ***Nessus***, ***GFI Languard***, ***Rapid7***, ***OpenVAS***, entre otros.

- **Nessus:** programa de escaneo de vulnerabilidades en diversos sistemas operativos. Con una versión de prueba de hasta 16 IPs, su mayor eficiencia se logra con la versión paga.

  Comprueba desde puertos abiertos, hasta *exploits* que pueden ser utilizados para atacar un equipo por ser vulnerable a estos. Posee una **consola** desde la cual se pueden ver los resultados del análisis efectuado y permite la exportación de los resultados en varios formatos. Es **muy robusto, pero quizá resulte algo complejo de utilizar** y de poder comprender los resultados obtenidos.

  Link para poder descargarlo: Nessus.

- **GFI Languard:** es un programa para realizar escaneos de vulnerabilidades multiplataforma que cumple con **los mismos requerimientos que *Nessus*, pero que suele utilizarse sobre plataformas Microsoft**. Tanto *Nessus* como *Languard* se utilizan en las organizaciones para automatizar esta tarea, cabe destacar que puede haber un grupo de analistas dedicados exclusivamente a llevar esto adelante, ya que desde el trabajo de descubrimiento, hasta la implementación de la corrección del error, lleva tiempo y esfuerzo, más si son muchos los equipos en los que se debe trabajar.

  Link de descarga: Gfi-Languard.

Si bien estos programas automatizan el descubrimiento, una vez llevada adelante esta tarea, se avecina la más ardua, la cual es **corregir lo relevado**, mediante la implementación en cada uno de los equipos vulnerables y sus respectivas correcciones. **Esta actividad suele realizarla el área de IT o Seguridad Informática**.

Escáneres de vulnerabilidades para uso personal

De la misma manera que podemos encontrar soluciones corporativas para grandes entornos de equipos, como *Nessus* u *OpenVAS*, también podemos utilizar software automatizado para encontrar en nuestros equipos vulnerabilidades de seguridad. Estos programas nos permitirán **descubrir qué S.O. o programas no están actualizados** a la última versión, o **qué parche de seguridad está faltando.**

Podemos encontrar entre los más conocidos a:

- **MSBA 2.3:** programa (descontinuado) **creado por *Microsoft* para relevar errores y malas configuraciones de seguridad, sobre cualquier producto de dicha empresa**, ya sea en servidores, bases de datos, *workstation*, etc. No arroja resultados sobre un producto que no haya sido creado por Microsoft.

  Permite realizar análisis de un equipo en particular, como de un conjunto en una red. Es gratuito y muy útil para realizar auditorías. Dejó de tener soporte oficial de Microsoft hace tiempo, pero resulta muy útil para fines didácticos de poder entender cómo funciona un escáner de vulnerabilidades de bajo nivel.

  Puede ser **reemplazado por el uso de *Nessus* en su versión gratuita** de hasta 16 IP.

  Link: Microsoft Baseline Security Analyzer 2.3 Download | TechSpot.

- **Sumo Updater:** para el resto de programas que estén instalados en los equipos podemos utilizar este *software* gratuito creado por ***KC software***, el cual permite **identificar qué aplicaciones están desactualizadas y nos brinda el enlace directo para poder actualizarlas**.

  Link: SUMo (Software Update Monitor).

De esta manera, tendremos ambos frentes cubiertos: por un lado, actualizado nuestro sistema operativo y por el otro, nuestros programas instalados en el equipo en cuestión.

Ambientes de prueba

Es clave tener en cuenta probar todo cambio o implementación nueva que queramos incluir en el ambiente de trabajo de una organización.

Explicado de una forma sencilla, existen **tres ambientes**:

- **Ambiente de desarrollo:** donde se **crea**.
- **Ambiente de prueba:** donde se **verifica** lo que se creó.
- **Ambiente de producción:** donde **funciona** lo que se creó.

Si nosotros hemos descubierto que somos vulnerables y debemos aplicar **una actualización** para solucionar esa falla, tenemos que considerar que **no se podrá instalar directamente sobre los equipos productivos** (los que están trabajando); ya que si no hemos verificado si esa actualización funciona bien, ésta podría provocar fallas que no estábamos esperando.

Es fundamental entender que, muchas veces, los desarrolladores de software hacen pruebas de sus productos y actualizaciones en un entorno de pruebas que no suele ser el mismo al que una organización pueda tener.

Estas diferencias podrían provocar que la actualización que nosotros queramos instalar genere una falla nueva y complique el desarrollo normal de las actividades. Por este motivo, **es necesario contar con un *ambiente de pruebas*** donde todo cambio, instalación o cosa que se tenga que probar pase primero por un **control de funcionamiento en este ambiente y si las pruebas son favorables, estará listo para llevar al ambiente de producción**.

¿Qué deberíamos tener en cuenta para crear un ambiente de pruebas?

- Los equipos que incluiremos dentro del ambiente deben ser **representativos del total**.
- Los equipos que incluiremos deben tener las **mismas características**: mismo S.O, aplicaciones instaladas y configuraciones.
- **No deberían utilizarse dispositivos fuera de servicio** o desactualizados ya que no son representativos de los que están en producción.
- Todos los **usuarios**, cuyos dispositivos formen parte de este ambiente de pruebas, serán **informados para poder recibir *feedback***.

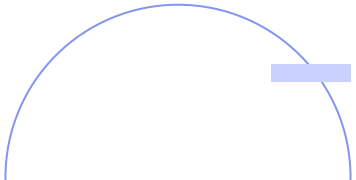
Si el resultado que arroja el test realizado en el ambiente de pruebas es **favorable**, es decir que esa instalación de *software* nueva o actualización o cambio en la configuración, etc. funcionó de forma esperada y no presentó problemas, significa que **está listo para llevar dicho cambio al ambiente productivo**.

Es muy importante tener en cuenta estas pruebas y no implementar nada directamente en producción, ya que podrían aparecer problemas no esperados, que sean más complicados que la vulnerabilidad que tratamos de solucionar.

Actualizaciones

La gran mayoría de las vulnerabilidades serán corregidas cuando implementemos los ***parches de actualización*** y otras se solucionarán con algunos **cambios de configuración**.

Es muy importante implementar los parches porque de no hacerlo, estamos provocando una dilatación en nuestra exposición, frente a una determinada vulnerabilidad.



Consideraciones a tener en cuenta al actualizar:

- **Instalar actualizaciones gradualmente** en los equipos, no todos juntos ni en el mismo día.
- Realizar la tarea **fuera del horario laboral**.
- **No permitir** que los equipos **se actualicen por sí solos**.
- **Centralizar la actualización** desde un equipo al resto (*WSUS*).
- Descargarlas desde **repositorios oficiales**.

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!