Segundo Parcial – 1er cuatri 2020

Parcial Online

Ejercicios de Verdadero/Falso (sin justificación)

Ejercicio 1

En el modelo BIBA para objetos, cuando se usa la política de RING-POLICY se puede decir que "la información de los objetos se vulgariza". Tener en cuenta precisamente las características de cada política.

- Verdadero
- Falso

Respuesta correcta: Falso. Se puede tomar que cuando se opera con BIBA para objetos, cuando se usa "low watermarking", los objetos toman el valor mínimo del objeto o del sujeto, por lo que si un sujeto de menor nivel accede al objeto, le reduce su nivel de confiabilidad. (escrito por Rodrigo)

Ejercicio 2

Todos los ataques similares de inyección de código, como SQL Injection, se producen por tener datos de control mezclados con datos de usuario.

- Verdadero
- Falso

Respuesta correcta: Verdadero.

Ejercicio 3

El modelo de Muralla China consiste en tres reglas básicas. La primera es la propiedad de seguridad simple, la segunda la matriz de permisos discrecionales y la tercera las reglas de compartimientos.

- Verdadero
- Falso

*Respuesta correcta: Falso. Posiblemente sean solo 2: la condición de seguridad simple y la propiedad *.*

Ejercicio 4

La realización de pentesting permite certificar que un sistema es seguro.

- Verdadero
- Falso

Respuesta correcta: Falso. Puede determinar que no existen ciertas vulnerabilidades pero podría haber otras que no hayan sido probadas. No es una demostración teórica.

Ejercicios de respuesta escrita

Ejercicio 5

La empresa MSA* ideó lo que supuestamente es un algoritmo revolucionario de autenticación de claves que lo anuncia a diestra y siniestra. Algunos estudios preliminares parecieron demostrar que la ejecución de ese algoritmo demora de manera proporcional a la longitud de la clave que verifica.

a) Armar un ejemplo numérico de cómo se podría utilizar la variación en la entropía para demostrarle al creador del algoritmo que el mismo "filtra" información en el tiempo de procesamiento. Asumir todas las simplificaciones que se crean convenientes, por ejemplo, que la p.d.f. (i.e. la función de distribución de probabilidad) de la longitud de las claves posibles en este problema es uniforme así como también las cotas en las claves que necesiten.

b) Ante el sólido argumento que le presentan al dueño de MSA, él contraargumenta que conocer la longitud de la clave no vuelve el sistema más inseguro. ¿ Como le contestarían ?

(*) Este es un nombre de fantasía que no tiene ninguna connotación con ninguna empresa real y que si coincide con alguna es mera casualidad.

Comentario de respuesta:
Si lo plantean tal cual está detallado en las presentaciones teóricas surge bastante directo. Acá tienen una función de asociación de la longitud de la clave con el tiempo que demora. Por lo tanto la entropía de las longitudes de las claves (alta) debería reducirse al medir la entropía de esas claves condicionadas a los valores de tiempos que se conocen y se pueden medir (pueden armar una pequeña tabla con tres o cuatro casos). (Escrito por Rodrigo)

Ejercicio 6

Actualmente diversos bancos ofrecen a sus usuarios la selección de un nombre de usuario de un alfabeto de 26 letras que caduca a los 90 días (adicionalmente al password). José Cifrín, el CSO del Banco de Seguranza, dice que el modelo de amenazas que establecieron opera bajo el supuesto que un atacante puede probar 10^9 usuarios por segundo, y se desea que el atacante tenga una probabilidad de éxito de 1/1000 a lo largo de 365 días.

a) ¿ Cuál es la longitud mínima de un nombre de usuario que cumpla con lo pedido ?

b) Los usuarios encuentran esta tarea de cambio de usuario bastante desagradable. Sin embargo, el CSO descarta esta afirmación por considerarla que no está alineada a los principios de seguridad. ¿Es esto correcto?

c) ¿ Qué alternativa a este esquema de autenticación podría implementarse en este caso ?

Respuesta:
P = TG/N y en este caso P = 1/1000 (para 365 dias), N = 26^x y G = 10^9. T = 365 dias en segundos, osea T = 31536000. Despejando, vemos que x, osea la longitud de la clave (del nombre de usuario en este caso), cumple que 26^x = TG/P entonces, resolviendo, x ~= 14.

Comentario de respuesta:
La cuenta sale directa. La "Aceptación del Usuario" es también un principio de diseño de seguridad que tiene que tenerse en cuenta. El CSO está equivocado al respecto. En c) se puede ser creativo pero una opción es usar 2FA o similar. Habría que explicar ventajas y desventajas. (Escrito por Rodrigo)

Ejercicio 7

Facebook los contrata para diseñar un modelo de seguridad para el control de la información oficial que se publica en la plataforma. Teniendo en cuenta los modelos BIBA, Bell Lapadula y Muralla China, planteen cómo estructurarían la solución para evitarnla proliferación de "fake news". Aclarar por qué y cómo usarían uno y no otro.

Comentario de respuesta:
La respuesta correcta tienen que girar alrededor de la comparación entre los diferentes modelos enfatizando que un modelo BIBA estricto es la solución más adecuada a donde se necesita una verificación de integridad (veracidad, confiabilidad) de la información. (Escrito por Rodrigo)

Ejercicio 8

Hace algunos años atrás, dado un programa ejecutable para una plataforma particular, el SO cargaba el binario donde se encontraba el ejecutable en memoria y automáticamente ejecutaba una instrucción en assembler para arrancar de una posición fija dentro del mapa de memoria de ese programa. Los sistemas operativos actuales implementan una distribución de la asignación de memoria que se denomina ASLR, Address Space Layout Randomization, donde esa dirección de inicio, así como otras direcciones, se determinan al azar. Explique brevemente, como este esquema ayuda a evitar el impacto de un tipo de vulnerabilidad concreta, y cuál es esa vulnerabilidad.

Respuesta correcta:
La vulnerabilidad que se puede ver afectada es "buffer overflow". La razón es debido a que si en algún punto el código inyectado explota que las direcciones dentro del mapa de memoria del programa son fijas y PREDECIBLES, el código extra que se introduce después del fin de buffer (y el que provoca el overflow) puede realizar saltos a esas direcciones y explotar en mayor medida el impacto de esa vulnerabilidad. (Escrito por Rodrigo)

Ejercicio 9

Explicar brevemente las diferencias entre ACLs y CAP.

a) ¿ Cuáles son las variantes específicas a definir en los ACLs en relación a la administración de los posibles conflictos o la administración de los valores por defecto ?

b) ¿ En qué situación concreta es preferible la utilización de CAPs en vez de ACL ? Ejemplos.

Comentario de respuesta:
Los ACLs pueden ser GRANT-ALL o FIRST RULE para definir los conflictos, y OVERRIDE o AUGMENT para administrar las asignaciones de los valores por defecto. Los CAPs son más útiles para dar permisos a usuarios anónimos, como por ejemplo los links de permisos sobre documentos compartidos en google drive. (Escrito por Rodrigo)

Ejercicios de opción múltiple

Ejercicio 10

Un experto en seguridad afirma que la utilización de software de código abierto es un problema de seguridad, porque un atacante podría inspeccionar el código y explotar alguna vulnerabilidad existente en el mismo.

- La afirmación es incorrecta, ya que el "Diseño Abierto" es un principio de diseño de seguridad que enfatiza que lo único que debe permanecer secreto son las claves privadas o simétricas.

- La afirmación es correcta, tal como se hace en soluciones militares la seguridad de un sistema está en mantener secretos los algoritmos y el código fuente.

- La afirmación es correcta ya que muchas de las vulnerabilidades encontradas en los sistemas surgen de inspeccionar el código fuente al que se tiene acceso.

- La afirmación es incorrecta, ya que la utilización de código libre y abierto es una garantía de seguridad.

Respuesta correcta:
La afirmación es incorrecta, ya que el "Diseño Abierto" es un principio de diseño de seguridad que enfatiza que lo único que debe permanecer secreto son las claves privadas o simétricas.

Ejercicio 11

Angular y otros frameworks Web implementan un mecanismo de solución para vulnerabilidades del tipo CSRF, que consiste en la generación de un identificador oculto "fresco" que se le envía al usuario y que se usa como un dato adicional a todos los requests posteriores que vienen del usuario hacia el servidor, tal como por ejemplo la que se encuentra detallada en

https://medium.com/@d.silvas/how-to-implement-csrf-protection-on-a-jwt-based-app-node-csurf-angular-bb90af2a9efd

- Esta librería o similares se puede utilizar como está para evitar ataques de XSS, ya que la existencia del token permitiría verificar que todos los requests que llegan al servidor son válidos, y rechazaría las peticiones que tienen contenido javascript.

- Esta librería o similares podrían utilizarse también para mitigar escenarios de vulnerabilidades de XSS ya que el token de CSRF que se genera desde el servidor prohibiría a los usuarios embeber código javascript en las páginas.

- Esta librería o similares tendría poco efecto para mitigar vulnerabilidades como XSS ya que en ese caso el problema aparece cuando el contenido en código javascript que se registra en algún campo de usuario, aún generando un request válido, se permite almacenar como datos de usuario.aq

- Esta librería o similares puede también utilizarse para mitigar las vulnerabilidades provocadas por inyecciones de código SQL ya que los requests que llegan al servidor estarían protegidos de ser generados a mano.

Respuesta correcta:
Esta librería o similares tendría poco efecto para mitigar vulnerabilidades como XSS ya que en ese caso el problema aparece cuando el contenido en código javascript que se registra en algún campo de usuario, aún generando un request válido, se permite almacenar como datos de usuario.
La generación de un token que se genera desde el servidor hace que los requests generados posteriormente sean vivos (opera similar a un nonce). Así como está no mitiga para nada un ataque del estilo XSS, ya que el problema en ese caso es que se acepta como entrada de datos de usuario código javascript que posteriormente se embebe en una página web, sin escapear el código lo cual hace que se ejecute (traiciona la confianza que el usuario tiene en que el servidor siempre le manda si o solo si contenido que el programador del servidor generó). (Escrito por Rodrigo)

Ejercicios de subida de archivo

Ejercicio 12

Considerar un esquema de secreto compartido de Shamir (3,4) en Z_7, con el conjunto de 4 sombras C = {(1, 6), (2, 0), (3, 0), (4, 6)}.

a) Recuperar el secreto.

b) Indicar el polinomio utilizado para obtener el conjunto de sombras.

Dado el contexto, es importante detallar en la hoja los pasos para arribar a la solución. Completar el ejercicio en una sola carilla.

Ejercicio 1

FALSO

Vulgarizar implica que no hay integridad. La información de los objetos se vulgarizo en low watermark policy y no en ring policy. Esto pasa porque LWP baja de nivel cuando lees y llega a un punto en el que todos pierden integridad.

Ejercicio 2

VERDADERO

Ejercicio 3

FALSO

Las dos segundas reglas no aplican. Estas reglas remiten a Bell-Lapodula.

Ejercicio 4

FALSO

Determina si existen vulnerabilidades pero no determina que no las haya.

Ejercicio 5

a. Supongamos $X = [1, 8]$ con distribucion uniforme. Luego, supongamos $Y = (10, 20, 30, 40, 50, 60, 70, 80)$. Finalmente, definimos $P(x = x_i / y) = 1$ si $y = x_i . 10$ y 0 en otro caso.

- $H(x) = 3$
- $H(X/Y = y) = 0, \forall y \in Y \Rightarrow H(X/Y) = 0$

Conclusion: Como $H(X/Y) < H(X)$, hay traspaso de informacion.

Explicacion by rramele:

Sabemos que hay leakage de informacion siempre que $H(X/Y) < H(X)$. Esto es que la entropia del sistema de x = largo de password te cae si vos si vos sabes otra cosa y, como puede ser el tiempo de procesamiento. Es decir que es mas facil adivinar el largo de la password si sabes cuanto tardo en procesar algo. De esta manera, se puede demostrar mediante esto que hay perdida de informacion. Hay que suponer una funcion de probabilidad para ambos casos, o una relacion entre x e y, y con eso armar un ejemplo numerico.

- Dibujar $p(x)$ y una relacion entre x e y i.e dibujar $p(x/y)$
- Calcular $H(X)$
- Calcular $H(X/Y)$
- Verificar $H(X/Y) < H(X)$

b. $P \geq \frac{TG}{N} \Rightarrow \frac{PN}{G} \geq T$

$\Rightarrow$ si sabemos el largo de la clave, conocemos N y disminuye el tiempo para esperar a adivinar la clave (se reduce el tamaño del espacio de claves)

Ejercicio 6

$$P \geq \frac{IG}{N}$$

- $P = 0.001$
- $T = 90 * 3600 * 24$
- $G = 10^9$

a. $N = 26^L$

$$\Rightarrow N \geq \frac{IG}{P} = 7.776 \times 10^{18}$$

$$\Rightarrow 26^L \geq 7.776 \times 10^{18}$$

$$\Rightarrow \log_{26}(26^L) \geq \log_{26}(7.776 \times 10^{18})$$

$$\Rightarrow L \geq 13.5$$

Rta: La longitud minima es de 14 caracteres.

b. FALSO

La afirmacion esta alineada con el principio de diseño de aceptacion psicologica que dice que mientras mas seguro, menos usable.

c. Se puede hacer 2FA para evitar cambiar la contraseña pero esto no lo vuelve mas usable.

Ejercicio 7

- Bell - Lapadula: Este modelo garantiza confidencialidad, lo cual no aplica para este problema.
- Muralla China: Este modelo soluciona problemas de conflictos de interes, lo cual tampoco aplica.
- Biba: Este modelo garantiza integridad. El modelo que mas aplica es el estricto porque queremos que los usuarios hagan publicaciones en base a las publicaciones de mayor integridad que ellos.

Ejercicio 8

La vulnerabilidad es Buffer Overflow. En la solucion previa, un atacante puede corromper con la integridad de los datos i.e un atacante puede pisar los datos y modificarlos.

Ejercicio 8

a. Las variantes para la administracion de los posibles conflictos son Grant-All y First-Rule. Para la administracion de los valores por defecto, las variantes son Override y Augment.

b. Si Alice quiere que todos sus archivos sean leidos por todos excepto Berta, es conveniente C-Lists. C-Lists estan enfocadas en el objeto en si y no en los usuarios. Si queremos poner que "accedan todos", conviene darle algo a todos que represente que pueden acceder al objeto (excepto a Berta). Por ejemplo, los google docs compartidos funcionan asi (el link seria ese algo) y eso permite que hayan accesos anonimos, porque de haber un ACL, habria que listar uno por uno todas las personas que podrian acceder y no seria posible usar usuarios anonimos (o habria que generar una categoria generica "anonimo" y agregarlo al ACL). By Rodri

Ejercicio 10

La afirmacion es incorrecta, ya que el "Diseño Abierto" es un principio de diseño de seguridad que enfatiza que lo unico que debe permanecer secreto son las claves privadas o simetricas.

Ejercicio 11

Esta libreria o similares tendria poco efecto para mitigar vulnerabilidades como XSS ya que en ese caso el problema aparece cuando el contenido en codigo javascript que se registro en algun campo de usuario, aun generando un script valido, se permite almacenar como datos de usuario.

La generación de un token que se genera desde el servidor hace que los requests generados posteriormente sean vivos (opera similar a un nonce). Así como está no mitiga para nada un ataque del estilo XSS, ya que el problema en ese caso es que se acepta como entrada de datos de usuario código javascript que posteriormente se embebe en una página web, sin escapear el código lo cual hace que se ejecute (traiciona la confianza que el usuario tiene en que el servidor siempre le manda si o solo si contenido que el programador del servidor generó). (Escrito por Rodrigo)

Ejercicio 12

a. $P(x) = 6\frac{(x-2)(x-3)}{(1-2)(1-3)} = 3(x^2 - 5x + 6) = 3x^2 - 15x + 18 \mod 7$

Rta: $S = P(0) = 18 \mod 7 = 4$

b. $P(x)_7 = 3x^2 + 6x + 4$