(20221Q) 72.44 - Criptografía y Seguridad - Comisión: S

Anuncios
Material Didáctico
Discusiones (Foros)
Grupos

E-mails
Sala Virtual

Estadísticas Alumno
Calificaciones

Ayuda

Revisar entrega de examen: Parcial 2

Usuario	AGUSTIN NASO RODRIGUEZ
Curso	(20221Q) 72.44 - Criptografía y Seguridad - Comisión: S
Examen	Parcial 2
Iniciado	23/06/22 16:03
Enviado	23/06/22 17:02
Estado	Completado
Puntuación del intento	65 de 100 puntos
Tiempo transcurrido	58 minutos de 1 hora y 30 minutos
Resultados mostrados	Todas las respuestas, Respuestas enviadas, Respuestas correctas, Comentarios, Preguntas respondidas incorrectamente

Pregunta 1

0 de 10 puntos

Describir cómo se podría implementar un esquema básico de autenticación en una aplicación web basada en HTTPS, y como implementar un esquema para evitar un ataque basado en CSRF.

Respuesta seleccionada: [No se ha dado ninguna]
Respuesta correcta: [None]
Comentarios para respuesta: [No se ha dado ninguna]

Pregunta 2

10 de 10 puntos

Considerar una política de contraseñas de 8 caracteres, entre ellas letras del alfabeto inglés en minúscula y números. ¿Cuál sería la probabilidad de que un atacante pueda obtener la contraseña en menos de un año si este contase con una tasa de pruebas de 23.000 pruebas por segundo y si se cuenta con 5 bits de salt?

Respuesta seleccionada: Si la longitud de la contraseña es de 8 mas 5 bits del salt. Esto da 2 a la 5 x 36 a la 8 (26 de letras + 10 de numeros)
P = (23.000 x 365x24x60x60)/(2 a la 5 x 36 a la 8)
P = 8,03 e-3

Respuesta correcta: [None]
Comentarios para respuesta: [No se ha dado ninguna]

Pregunta 3

3 de 10 puntos

Suponiendo el software del controlador de las barras de grafito de la central nuclear de Chernobyl, ¿Se puede asegurar que un sistema cómo este es seguro mediante algún mecanismo?

Respuesta seleccionada: No, no existe un mecanismo para garantizar la seguridad total de un sistema de software
Respuesta correcta: [None]
Comentarios para respuesta: La pregunta apunta a la pieza de software que implementa el controlador, y es justo el ejemplo típico donde es posible usar métodos formales para verificar que hace lo que dice que tiene que hacer y demostrarlo.

Pregunta 4

10 de 10 puntos

La manera más segura de evitar que un sistema de APIs interna de un Sistema Operativo sea vulnerado es mantener secreta la documentación.

Respuesta seleccionada: Falso
Respuestas: Verdadero
Falso
Comentarios para respuesta: Eso sería seguridad por ofuscación.

Pregunta 5

6 de 10 puntos

Enumerar y explicar algunas de las técnicas utilizadas en los sistemas operativos modernos para intentar mitigar ataques de buffer overflow.

Respuesta seleccionada: Un ataque de buffer overflow se produce cuando un programa arranca siempre en una direccion de memoria fija.
Si el atacante descubre esto, es posible que encuentre la forma de, mediante overflow en el buffer, llegar a partes de codigo especificas.
Para resolver esto, se aleatoriza la direccion de memoria donde comienza el progama, logrando asi que al no ser determinista el atacante no pueda saber que escibir en el buffer para obtener una direccion especifica.
Otra forma es agregando explicitamente un 0 al final del string de entrada, al final de una longitud predefinida.
Respuesta correcta: [None]
Comentarios para respuesta: El segundo es más complejo de forzarlo desde el sistema operativo.

Pregunta 6

10 de 10 puntos

La Autenticación de un usuario implica verificar que un usuario tiene una pieza de información que puede contastarse contra la que tiene un sistema y con eso decidir si la verificación fue correcta o incorrecta.

Respuesta seleccionada: Verdadero
Respuestas: Verdadero
Falso
Comentarios para respuesta: Si

Pregunta 7

0 de 10 puntos

Mencione las 5 etapas por las que se pasa durante un pentesting explicando brevemente de qué se trata cada una.

Respuesta seleccionada: [No se ha dado ninguna]
Respuesta correcta:
(Recolección = Recon, Scanning = Hipótesis, Prueba = Exploit, Generalización = Privesc/escalamiento vertical o lateral)
1. Recon, se realiza toda la investigación del objetivo.
2. Scanning. Se utiliza toda la información obtenida en la parte 1 y se verifica la existencia de vulnerabilidades especificas.
3. Exploit. Se realiza la explotación en sí misma de la vulnerabilidad.
4. Privesc. Se continúa el ataque desde otros ángulos, ya sean con mayores privilegios (esc vertical) o con similares (esc lateral).
5. Eliminarlo
Comentarios para respuesta: [No se ha dado ninguna]

Pregunta 8

6 de 10 puntos

Para un trabajo práctico de la universidad, con su grupo concuerdan realizar un "hosteador de CVs". La idea es que una persona suba su CV como un archivo HTML y su servicio simplemente hostearía el archivo, garantizando su disponibilidad y ofreciendo que esté bajo el nombre de dominio que usted elija. ¿Ve algún problema de seguridad en la idea? Argumente su respuesta.

Respuesta seleccionada: El probema de esto es que si simplemente hosteo el archivo HTML, este puede contener codigo JavaScript malicioso dentro de los tags script.
Si yo escribo casi identico el dominio del HTML con codigo malicioso al de un usuario x y el usuario por distraido accede a mi archivo, pensando que es el suyo, va a ser victima del objetivo del codigo malicioso de mi archivo.
Respuesta correcta:
Es una muy mala idea desde el punto de vista de seguridad porque un usuario podría subir un HTML con código malicioso y nosotros los estaríamos hosteando para que cualquier persona que entre, lo active. Habría que *mínimo* sanitizar la entrada y aún así, revisar todos los links que utilice.
Comentarios para respuesta: También XSS, CSRF, etc

Pregunta 9

10 de 10 puntos

A su equipo le asignaron realizar el aislamiento de ciertos programas para garantizar que no haya flujo de información. Hablando con sus compañeros de las tareas, le dicen que se puede hacer en un solo día debido a que para aislar un proceso solamente se necesita realizar un aislamiento de red y de espacio en disco. ¿Es correcto lo que plantean? Explicar

Respuesta seleccionada: No es correcto. No solo es necesario aislar red y espacio en disco sino que tambien habria que tratar de que no se comparta absolutamente nada entre los procesos.
Esto es bastante dificil ya que para aislar por completo habria que correrlos, por ejemplo, en sandoboxes separados
Si solo no comparto red y espacio en disco, y no tengo en cuenta otras variables como memoria o capacidad de proceso, puede haber flujo de informacion, por ejemplo, en cuanto a la duracion de la ejecucion del proceso (utilizando entropia condicional respecto de lo que procesa el programa y el tiempo de ejecucion).
Respuesta correcta:
El problema son los canales ocultos que pueden ocurrir dentro de la misma computadora, y los que puedan surgir de canales ocultos físicos como puede ser tiempo de uso de procesador, consumo, movimiento y velocidad de los disipadores.
Comentarios para respuesta: [No se ha dado ninguna]

Pregunta 10

10 de 10 puntos

En relación a las políticas de seguridad,

Respuesta seleccionada: Los ACLs y los C-Lists son ejemplos de DAC.
Respuestas: Los ACLs y los C-Lists son ejemplos de DAC.
Los ACLs son ejemplo de DAC, mientras que los CAPs son ejemplos de MACs
Biba y Bell-Lapadula son ejemplos de DACs.
Las lista de capacidades referencian los usuarios que tienen diferentes tipos de acceso a un recurso.

miércoles 7 de junio de 2023 09H32' ART

← Aceptar