TP 1

Seguridad Informática

Farizano, Juan Ignacio

Ejercicio 3

Si las políticas de seguridad son muy restrictivas, con el e-mail por ejemplo, dificultaría la comunicación con direcciones ajenas a las internas de la empresa, incluso provocando que mails importantes no sean recibidos o terminen en la casilla de spam, entorpeciendo o evitando que se den actividades comerciales.

También si las politicas de acceso a los datos son muy firmes, podría dificultar el desarrollo de nuevos productos, por ejemplo si un sector de una empresa está desarrollando un nuevo producto que depende de una tecnología que está siendo a su vez desarrollada en un sector diferente, el intercambio de información limitado provocaría que se retrase la fecha de salida de este producto.

Ejercicio 9

Un ejemplo donde la ocultación de información no provee mayor seguridad es en el caso del software libre. Por ejemplo el kernel de Linux, sobre el cual corre la gran mayoría de servidores del mundo, el hecho de que cualquiera pueda ver el código permite que una gran desarrolladores de todo el mundo revise este mismo en busca de vulnerabilidades y sean corregidas rápidamente.

Un ejemplo donde sí sucede es cuando se oculta la ubicación de información confidencial. Por ejemplo, en una computadora se divide el disco en dos particiones, una con información pública y otra con información sensible. Un usuario con un nivel de seguridad bajo no debería conocer la ubicación, tamaño o ni siquiera la existencia de la partición con información de la que se necesita una nivel de seguridad mayor para acceder a ella, esto dificulta el acceso no autorizado a estos datos.

Ejercicio 15

Ver archivo **Ej15.hs** incluido

Ejercicios 21 y 22

Elijo el programa 4 para resolver los ejercicios 21 y 22.

Ejercicio 21

No hay flujo indebido de información ya que en ningún momento se lee una variable de nivel H, por lo que no puede haber ningún flujo indebido a algún canal de salida L. A pesar de que se pueda obtener información sobre el usuario y la contraseña, estos no están detrás de variables de nivel H, por lo que es más un problema de malas prácticas de programación.

Ejercicio 22

Bajo el sistema de seguridad por multi-ejecución, este programa funcionaría siempre de una sola manera por lo explicado en el inciso anterior. Al no haber ninguna lectura o escritura de nivel H, el proceso original nunca realiza un fork, por lo tanto la memoria nunca será diferente para usuarios con nivel L o con nivel H y la ejecución será la misma para ambos.

Ejercicio 23

Mi desafío elegido es el step 7.

En este desafío el lenguaje implementa un modelo simple de asignación dinámica de memoria, donde se le asigna un valor a una celda de memoria. El problema con este sistema es que revisa que no se escriban valores H en una referencia declarada como low, pero no revisa que no se escriba en una variable L haciendo una deferencia a una variable de referencia high.

Se puede explotar esta falla en el lenguaje, esta es mi solución para el problema:

```
declare ref x : high;
ref x = h;
l = deref(x);
```

Este es un ataque de filtrado de información con un flujo de información indirecto y explícito. Es indirecto porque el dato pasó por una variable intermedia (la variable de referencia x) para ir de la variable h a la variable l, y explícito porque el flujo de un dato hacia otro fue de manera explícita en el código, realizando asignaciones de variables.

Para impedir el ataque se deberían modificar las reglas de inferencia del sistema de tipos para que al hacer una asignación del estilo (x = deref(y)) se verifique que el nivel de la variable x sea mayor o igual al nivel de la variable de referencia y.