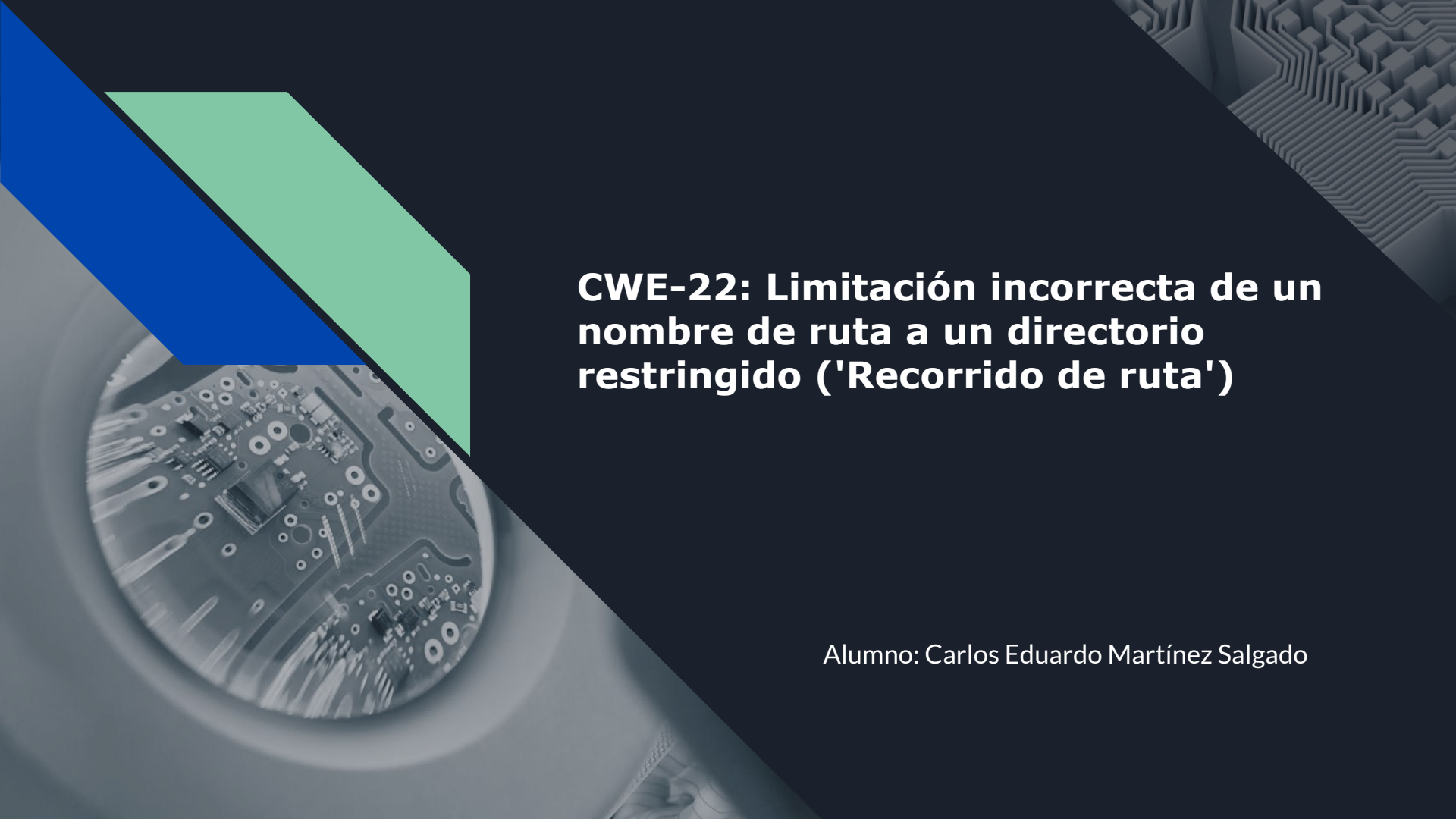

CWE-22: Limitación incorrecta de un nombre de ruta a un directorio restringido ('Recorrido de ruta')

Alumno: Carlos Eduardo Martínez Salgado

Descripción

El software utiliza una entrada externa para construir un nombre de ruta destinado a identificar un archivo o directorio ubicado debajo de un directorio principal restringido, pero el software no neutraliza adecuadamente elementos especiales dentro del nombre de ruta que pueden hacer que el nombre de ruta se resuelva en una ubicación que está fuera del directorio restringido.

Descripción extendida

Se pretende que muchas operaciones de archivo tengan lugar dentro de un directorio restringido. Mediante el uso de elementos especiales como ".." y "/" separadores, los atacantes pueden escapar fuera de la ubicación restringida para acceder a archivos o directorios que están en otras partes del sistema. Uno de los elementos especiales más comunes es la secuencia "../", que en la mayoría de los sistemas operativos modernos se interpreta como el directorio principal de la ubicación actual. Esto se conoce como recorrido relativo de la ruta. El recorrido de ruta también cubre el uso de nombres de ruta absolutos como "/ usr / local / bin", que también pueden ser útiles para acceder a archivos inesperados. Esto se conoce como recorrido absoluto del camino.

Relaciones

La (s) tabla (s) a continuación muestra las debilidades y las categorías de alto nivel relacionadas con esta debilidad. Estas relaciones se definen como ChildOf, ParentOf, MemberOf y brindan información sobre elementos similares que pueden existir en niveles de abstracción cada vez más altos. Además, las relaciones como PeerOf y CanAlsoBe se definen para mostrar debilidades similares que el usuario puede desear explorar.

-Relevante para la vista "Conceptos de investigación" (CWE-1000)

-Relevante para la vista "Debilidades para la asignación simplificada de -vulnerabilidades publicadas" (CWE-1003)

Relevante para la vista "Conceptos de desarrollo" (CWE-699)

Plataformas aplicables

Los listados a continuación muestran posibles áreas para las que podría aparecer la debilidad dada. Estos pueden ser para idiomas específicos nombrados, sistemas operativos, arquitecturas, paradigmas, tecnologías o una clase de tales plataformas. La plataforma se enumera junto con la frecuencia con la que aparece la debilidad dada para esa instancia.

Consecuencias comunes

La siguiente tabla especifica diferentes consecuencias individuales asociadas con la debilidad. El Alcance identifica el área de seguridad de la aplicación que se viola, mientras que el Impacto describe el impacto técnico negativo que surge si un adversario logra explotar esta debilidad. La probabilidad proporciona información sobre la probabilidad de que se vea la consecuencia específica en relación con las otras consecuencias en la lista. Por ejemplo, puede haber una alta probabilidad de que se explote una debilidad para lograr un cierto impacto, pero una baja probabilidad de que se explote para lograr un impacto diferente.

Alcance	Impacto	Probabilidad
Integridad Confidencialidad Disponibilidad	**Impacto técnico:** *ejecutar código o comandos no autorizados* El atacante puede crear o sobrescribir archivos críticos que se utilizan para ejecutar código, como programas o bibliotecas.	
Integridad	**Impacto técnico:** *modificar archivos o directorios* El atacante puede sobrescribir o crear archivos críticos, como programas, bibliotecas o datos importantes. Si el archivo de destino se usa para un mecanismo de seguridad, entonces el atacante puede evitar ese mecanismo. Por ejemplo, agregar una nueva cuenta al final de un archivo de contraseña puede permitir que un atacante omita la autenticación.	
Confidencialidad	**Impacto técnico:** *leer archivos o directorios* El atacante puede leer el contenido de archivos inesperados y exponer datos confidenciales. Si el archivo de destino se usa para un mecanismo de seguridad, entonces el atacante puede evitar ese mecanismo. Por ejemplo, al leer un archivo de contraseña, el atacante podría realizar ataques de adivinación de contraseña de fuerza bruta para entrar en una cuenta en el sistema.	
Disponibilidad	**Impacto técnico:** *DoS: bloqueo, salida o reinicio* El atacante puede sobrescribir, eliminar o corromper archivos críticos inesperados, como programas, bibliotecas o datos importantes. Esto puede evitar que el software funcione y, en el caso de mecanismos de protección como la autenticación, tiene el potencial de bloquear a todos los usuarios del software.	

EJEMPLO

El siguiente código podría ser para una aplicación de red social en la que la información de perfil de cada usuario se almacena en un archivo separado. Todos los archivos se almacenan en un solo directorio.

Lenguaje de ejemplo: **Perl** *(código incorrecto)*

```
my $ dataPath = "/ users / cwe / profiles";
my $ username = param ("usuario");
my $ profilePath = $ dataPath. "/". $ nombre de usuario;

abierto (mi $ fh, "<$ profilePath") || ExitError ("error de lectura de perfil: $ profilePath");
imprime "<ul> \ n";
while (<$ fh>) {
  print "<li> $ _ </li> \ n";
}
print "</ul> \ n";
```

Si bien el programador intenta acceder a archivos como "/ users / cwe / profiles / alice" o "/ users / cwe / profiles / bob", no hay verificación del parámetro de usuario entrante. Un atacante podría proporcionar una cadena como:

(código de ataque)

```
../../../etc/passwd
```

El programa generaría un nombre de ruta de perfil como este:

(resultado)

```
/users/cwe/profiles/../../../etc/passwd
```

Cuando se abre el archivo, el sistema operativo resuelve "../" durante la canonicalización de la ruta y realmente accede a este archivo:

(resultado)

```
/ etc / passwd
```

Como resultado, el atacante podría leer el texto completo del archivo de contraseña.