Criptografía y seguridad

TRABAJO PRÁCTICO ESPECIAL

Los 25 errores de Software más frecuentes  
*2 de Junio de 2014*

Alan Pomerantz

Juan José Marinelli

Agustin Marseillan

1. **Abstract**

En el presente informe se expone la investigación de cuatro de los veinticinco errores presentes en el software más frecuentes, dentro del ámbito informático. Se procede a introducir el tema que contempla a dichos errores, y luego se brinda una descripción técnica del error, junto a ejemplos de código, métodos de detección, niveles de vulnerabilidad y consecuencias sobre un posible exploit sobre dicho error, y por último, formas de mitigar dichos errores. Los errores seleccionados fueron el 2 (dos) “Improper Neutralization of Special Elements used in an OS command (‘OS Command Injection’), 7 (siete) “Use of Hard-coded Credentials” , 12 (doce) “Cross-Site Request Forgery (CSRF)” y 17 (diecisiete) “Incorrect Permission Assignment for Critical Resource”.

1. **Introducciòn**

Existe una lista ordenada bajo un ranking, conteniendo los errores más frecuentes dentro del ámbito informático, de los que uno como programador debe tener cuidado de no generar, ya que un atacante puede sacar provecho de la situación, generando escenarios no deseables dentro del flujo general dentro de un programa. La lista, elaborada en el año 2011, se puede encontrar online en la siguiente direccion: *http://cwe.mitre.org/top25/* [[1]](#footnote-1)

El presente informe selecciona 4 de estos 25 errores más frecuentes y propone un análisis detallado del mismo.

1. **Errores seleccionados**

Los errores seleccionados fueron elegidos por la cátedra para que los analice el grupo AliceBobMallory.

* 1. **OS Command Injection**
     1. **Descripción del error**

El error OS Command Injection convive en la idea de que un atacante tome control del input con el cual un programa puede llamar a otro, dentro del mismo SO (Sistema Operativo). Un atacante entonces, podría ejecutar sus propios comandos .

Facilidad de Detección del error: Sencilla

Consecuencias: Ejecución de código no deseado

Frecuencia del ataque : Alta

Costo de reparaciòn: Medio

Weakness Prevalence: Medio

Attacker Awareness: Alto

* + 1. **Detalles técnicos del error**

El escenario generalmente sucede en web, donde un atacante no tiene acceso directo al sistema operativo, y puede ejecutar comandos que pueden ser no deseados, o en sistemas que tienen jerarquía de permisos, y donde un atacante puede ejecutar comandos para los cuales de otra forma no tendría permisos. Tambien es conocido como Shell Injection, ya que se aplica sobre terminales de comandos. El error es independiente del lenguaje.

Existen dos tipos de inyección de comandos:

* Una aplicación intenta ejecutar un programa del cual tiene control. El mismo utiliza un input que es externo, y el atacante logra interponerse para ejecutar un programa de su interès.
* Una aplicaciòn acepta un input externo, con el cual decide que programa va a correr luego, asi como tambien què comando utilizarà, de esta manera, la aplicaciòn retransmite el input externo, directamente al sistema operativo.
  + 1. **Ejemplos de código**

Un ejemplo de código sería el siguiente:

system("nslookup [HOSTNAME]")

En donde el atacante, respetando la sintaxis de la terminal, podrìa poner en la variable HOSTNAME un comando de su propiedad,

entonces luego de la ejecucion del nslookup, se ejecutarìa su comando.

Otros ejemplos:

(Lenguaje PHP)

*$userName = $\_POST["user"];*

*$command = 'ls -l /home/' . $userName;*

*system($command);*

Nada previene que un usuario malicioso coloque en $userName algun comando linux como “;rm –rf/”. Lo que conduciria a que el comando final quede como: ls –l/home/;rm –rf/

* + 1. **Métodos de detección**

Para poder detectar este error, se utilizan distintas técnicas. Entre algunas de ellas se destacan:

* Automated Static Analysis:
* Automated Dynamic Analysis:
* Manual Static Analysis:
  + 1. **Nivel de vulnerabilidad y consecuencias posibles de un exploit sobre ese error**
    2. **Formas de mitigar y/o evitar el error**
  1. **Use of Hard-Coded Credentials**

Facilidad de Detección del error: Sencilla

Consecuencias: Ejecución de código no deseado

Frecuencia del ataque : Alta

Costo de reparaciòn: Medio

Weakness Prevalence: Medio

Attacker Awareness: Alto

* + 1. **Descripción del error**
    2. **Detalles técnicos del error**
    3. **Ejemplos de código**
    4. **Métodos de detección**
    5. **Nivel de vulnerabilidad y consecuencias posibles de un exploit sobre ese error**
    6. **Formas de mitigar y/o evitar el error**
  1. **Cross-Site Request Forgery (CSRF)**
     1. **Descripción del error**

Facilidad de Detección del error: Sencilla

Consecuencias: Ejecución de código no deseado

Frecuencia del ataque : Alta

Costo de reparaciòn: Medio

Weakness Prevalence: Medio

Attacker Awareness: Alto

* + 1. **Detalles técnicos del error**
    2. **Ejemplos de código**
    3. **Métodos de detección**
    4. **Nivel de vulnerabilidad y consecuencias posibles de un exploit sobre ese error**
    5. **Formas de mitigar y/o evitar el error**
  1. **Incorrect Permission Assigment for Critical Resource**
     1. **Descripción del error**

Facilidad de Detección del error: Sencilla

Consecuencias: Ejecución de código no deseado

Frecuencia del ataque : Alta

Costo de reparaciòn: Medio

Weakness Prevalence: Medio

Attacker Awareness: Alto

* + 1. **Detalles técnicos del error**
    2. **Ejemplos de código**
    3. **Métodos de detección**
    4. **Nivel de vulnerabilidad y consecuencias posibles de un exploit sobre ese error**
    5. **Formas de mitigar y/o evitar el error**

1. Sitio web perteneciente a la CWE (Community Weakness Enumeration) [↑](#footnote-ref-1)