Vulnerabilidades en el tratamiento de los datos de entrada

Una vulnerabilidad es algo que no está bien implementado o implementado de forma débil, de forma que esta debilidad pude ser aprovechada por un atacante para causar algún daño. La vulnerabilidad más habitual es que los datos de un usuario no se validen de forma adecuada. Para esto se pueden usar formularios o webs que se consultan de forma programática.

Inyección de código

En la comunicación entre aplicaciones y bases de datos suelen haber unos tokens como “;” que se pueden utilizar de forma que usando los datos del usuario se puede ejecutar una consulta en una base de datos que no se debería poder ejecutar normalmente. Esta vulnerabilidad era la más importante hasta que en 2021 paso a la tercera posición del OWASP. Existen varios tipos de inyección:

- SQL
- De Log
- Comandos de sistema operativo
- XML
- XPath

Inyección SQL

Se produce cuando los datos de entrada proporcionados por el usuario utilizan algunos de los tokens de SQL de forma que no se procesan de forma adecuada, resultando en que la consulta de validación no se ejecuta de la forma esperada. Por ejemplo, una consulta de login sql puede ser:

```
String q = "SELECT * FROM users Where email='" + email + "'AND password='" + password +"'";
```

Un ejemplo de ataque de inyección sería el siguiente:

```
"SELECT * FROM users Where email = 'bob@acme.com' OR '1' = '1' AND password = 'any'"
```

Al meter un OR de por medio puede engañar al sistema de validación para dejar loguear.

Otro ejemplo sería este:

```
SELECT email, password, full_name
FROM users
WHERE email  = 'any' OR full_name LIKE '%Bob%';
```

También se pueden realizar ataques destructivos metiendo un DROP en la consulta.

Inyección de JavaScript (XSS)

Entidades externas en documentos XML (XEE)

Deserialización y carga dinámica

From:
https://www.knoppia.net/ - **Knoppia**

Permanent link:
https://www.knoppia.net/doku.php?id=app:vemp&rev=1726680703

Last update: **2024/09/18 17:31**