Universidad del Valle de Guatemala
Ingeniería de Software 2


Tarea investigativa 5.

# Informe de Herramientas DevSecOps

Kristen Brandt - 171482
Estuardo Ureta - 17010
Sergio Marchena - 16387
Oliver Graf - 17190

# 1. Vulnerabilidades

## a. Las más comunes y los principales riesgos que pueden suceder en nuestro caso

Según la organización conocida como Open Web Application Security Project (OWASP) estas son algunas de las vulnerabilidades más comunes (y que consideramos nos afectarán más):

1. Inyección de código maligno
    a. Esto se refiere a cuando un hacker utiliza código ajeno al de la aplicación para manipular los métodos y la información de dicha aplicación. También son conocidos como SQL injection y pueden causar que mucha información sensible sea robada. Por lo general, estas vulnerabilidades se pueden evitar haciendo uso de “white lists”, que sirven para limitar el acceso para el uso de las bases de datos.
2. Autenticación rota
    a. En este caso, la seguridad de la aplicación se perjudica por tener una autenticación muy débil o expuesta. En plataformas como WordPress, sobre la cual un usuario puede crear y administrar sus propias páginas web, ocurre frecuentemente que el usuario administrador (o por lo menos la página donde el administrador ingresa sus datos para entrar) es accesible por cualquier persona que ingrese el URL en un navegador web. Este tipo de ataques pueden mitigarse haciendo uso de autenticación de múltiples factores/niveles. Además, es importante recordarle al usuario que utilice contraseñas más complejas, para que sea más difícil hackearlas con fuerza bruta.
3. Exponer información sensible
    a. Este tipo de vulnerabilidad es de las más comunes y suele pasar si el equipo de desarrollo se confía y no maneja de manera adecuada los datos del usuario. Para evitar exponer este tipo de información, es importante recordar qué información es la más importante ocultar y proteger. Algunos ejemplos de esta información son: números de tarjetas, credenciales, códigos de identificación, información privada, etc. Cabe mencionar que hay dos estados en los cuales se puede encontrar la información: almacenada y en tránsito. En ambos estados es posible que la información sea robada, por lo que es necesario tomar en cuenta ataques a la información almacenada (como lo son las inyecciones) y ataques a la información siendo enviada o recibida. Es importante estudiar más a fondo qué maneras hay para encriptar la información.

4. Monitoreo insuficiente
    a. Es de suma importancia no subestimar lo delicada e importante que es la seguridad de una aplicación. Claro que crear una aplicación que esté completamente sellada a ataques de hackers es prácticamente imposible, pero esto no quiere decir que podamos bajar la guardia en ningún momento. Se recomienda mantener bien monitoreadas todas las áreas de la aplicación, para que cuando se dé un ataque, éste sea identificado lo más rápido posible y que se pueda actuar adecuadamente. Algunas áreas que son importantes de monitorear son la integridad de los archivos, el log de movimiento de datos, procesos, etc.

## 2. Herramientas a utilizarse

Existen varias herramientas para ayudar a los desarrolladores a automatizar las pruebas de seguridad y estas son algunas:

Para monitoreo:

- evident.io
- metasploit
- splunk
- FireEye

Para desplegar código:
- Inspec
- Beaker
- Gitlab

Para Tests y scans:
- Contrast security
- evident.io
- Gauntlt

Para servidores CI:
- Splunk

Para código fuente:
- Gitrob
- Gitlab
- Checkmarx

En el caso de nuestra aplicación tenemos pensado emplear Gitlab para todo lo que es seguridad en el desarrollo. Pensamos que al utilizar solo una herramienta el desarrollo va a ser menos complejo y por lo tanto, más eficiente. También queremos hacer uso de evident.io, porque nos va a permitir monitorear la aplicación cuando ya esté en funcionamiento y también nos permite hacer pruebas unitarias de seguridad. Por el momento, nos enfocaremos en familiarizarnos bien con las herramientas que nos ofrece Gitlabs de DevSecOps, por un lado para aprender más sobre el tema y cómo se usan estas herramientas y, por otro lado, para determinar si vamos a necesitar alguna otra herramienta de las mencionadas anteriormente para asegurar más nuestra aplicación.

## 3. BDD y Gherkin

Estos son algunos de los ejemplos que consideramos que vale la pena retomar y aplicar los conceptos de seguridad.

```
@Login Correcto @AppFinca
Scenario: Funcionalidad de Login
Given: usuario entra a la sección de login de la aplicación
When: usuario ingresa usuario “K” y contraseña “12345” (registrados)
Then: el login es exitoso y la aplicación debe mostrar mensaje de bienvenida

@Login Incorrecto @AppFinca
Scenario: Funcionalidad de Login
Given: usuario entra a la sección de login de la aplicación
When: usuario ingresa usuario “USUARIO” y contraseña “CONTRASEÑA” (registrados)
Then: la aplicación debe mostrar mensaje de error

@Login Correcto @AppFinca
Scenario: Funcionalidad de Registro
Given: usuario entra a la sección de registro de la aplicación
When: usuario ingresa usuario “USUARIO” y contraseña “CONTRASEÑA” (no registrados)
Then: la aplicación debe mostrar mensaje de registro

@Login Incorrecto @AppFinca
Scenario: Funcionalidad de Registro
Given: usuario entra a la sección de registro de la aplicación
When: usuario ingresa usuario “USUARIO” y contraseña “CONTRASEÑA” (ya registrados)
Then: la aplicación debe mostrar mensaje de error

@Anadir nueva Venta @AppFinca
```

**Scenario:** Funcionalidad de ventas
**Given:** usuario entra a la sección de registro de la aplicación
**When:** usuario ingresa usuario “USUARIO” y contraseña “CONTRASEÑA” (ya registrados)
**Then:** la aplicación debe mostrar mensaje de error

Referencias:

1. National Institute of Standards and Technology, Framework for Improving Critical Infrastructure Cybersecurity, 2018.
2. Office of the DoD CIO, "DoD DevSecOps Playbook," 2019. [Online]. Available: https://www.milsuite.mil/book/groups/dod-enterprise-devsecops.
3. N. M. Chaillan, "DoD Enterprise DevSecOps Initiative Hardening Containers," DRAFT, 2019.
4. E. Ries, "The Lean Startup," [Online]. Available: http://theleanstartup.com/principles. [Accessed 09 Noviembre 2020].