UNIVERSIDAD CATOLICA

DIPLOMADO
TESTIG DE SOFTWARE

MODULO 4

AUTOMATIZACION II MOBILE / REST API / VULNERABILITY

TAREA Nro. 3

Nombre: Atahuichi Mamani Glaucia Mariel

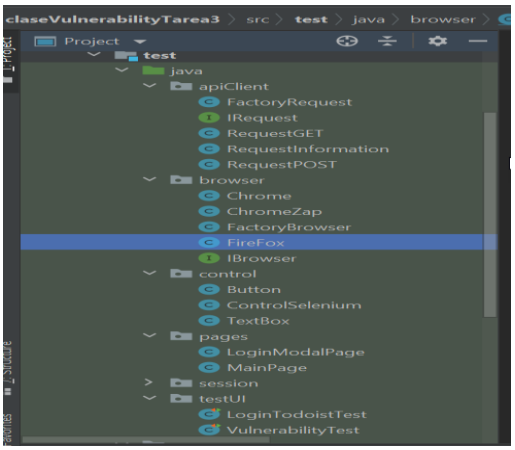
Tarea 3:

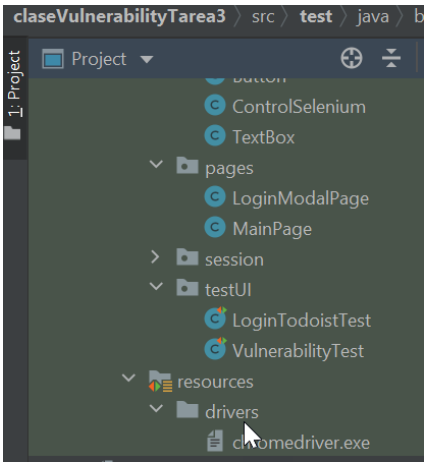
- Crear un documento PDF con capturas de pantalla realizando paso a paso la automatización de vulnerability test usando https://todoist.com/
- Subir el código a un repositorio de versionador de código y poner el link del repositorio en el documento pdf

nota: incluir datos del reporte generado en las capturas de pantalla

Link de repositorio:

1.El árbol que se estructurara para este ejercicio donde se realizara el test de login para la página https://todoist.com/ y el test de vulnerabilidad que se realizara.



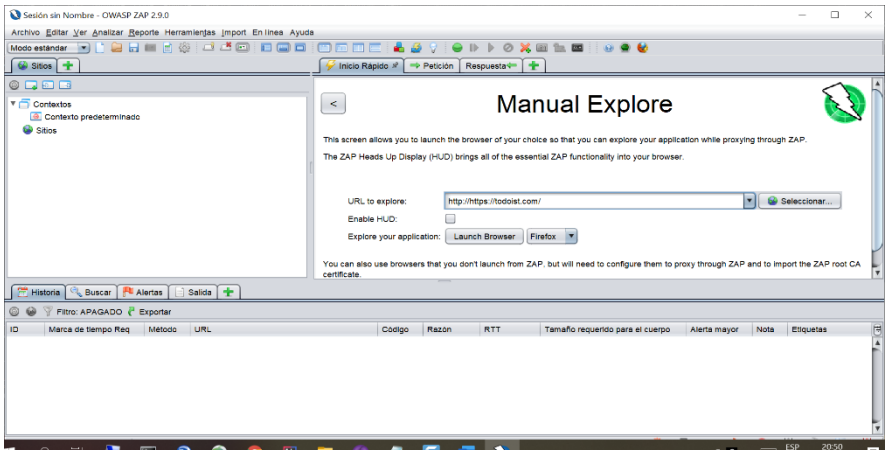


2.Se debe tomar en cuenta el PUERTO DE PROXY que se está usando en OWASP

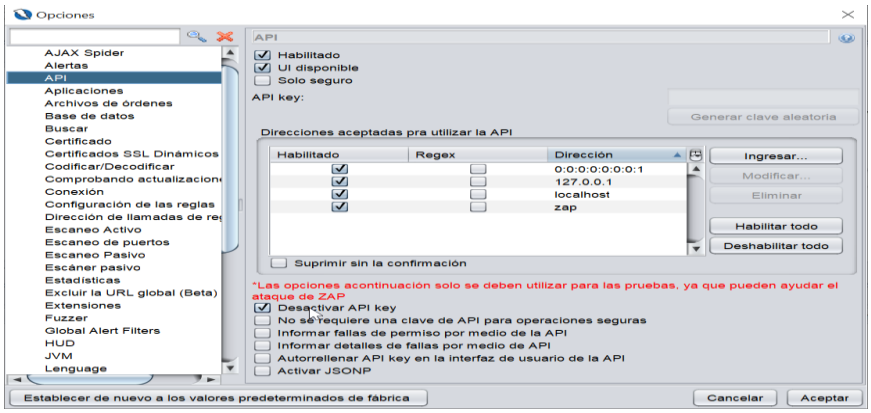
```
// 2 Proxy --- PORT ---> mismo del OWASP ZAP
Proxy proxy = new Proxy();
proxy.setHttpProxy("127.0.0.1:8888");
proxy.setSslProxy("127.0.0.1:8888");
```

3. configuración que se realiza en OWASP

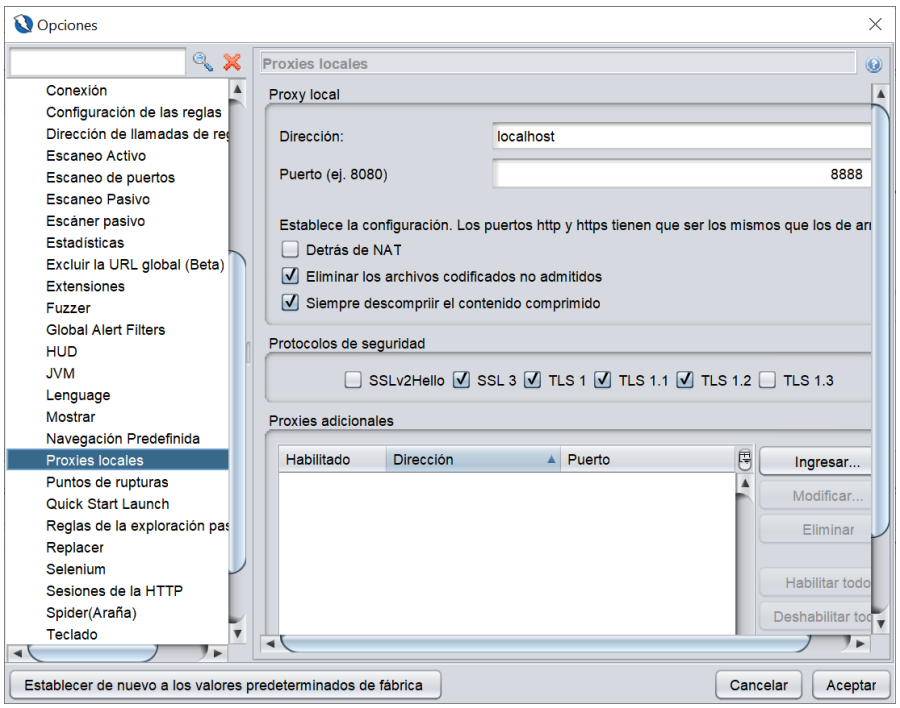
Se configura la página que se va analizar.



Seleccionar la opción desactivar API key en herramientas/opciones /API



Configurar el puerto que se usara 8888



4. package→ browser

Donde podremos agregar lo que usaremos en como el navegador y el link que se configurara para cromedriver.exe que estará en resuourses

```
package browser;
import org.openqa.selenium.WebDriver;
import
org.openqa.selenium.chrome.ChromeDriver;
import java.util.concurrent.TimeUnit;

public class Chrome implements  IBrowser {
    @Override
    public WebDriver create() {
        String
driverPath="C:\\Users\\MARIEL\\IdeaProjects
\\claseVulnerabilityTarea3\\src\\test\\reso
urces\\drivers\\chromedriver.exe";

System.setProperty("webdriver.chrome.driver
",driverPath);
        ChromeDriver driver = new
ChromeDriver();

driver.manage().timeouts().implicitlyWait(2
0, TimeUnit.SECONDS);
        return driver;
    }
}
```

```
package browser;
public class FactoryBrowser {
    public static IBrowser make(String
typeBrowser){
        IBrowser browser;
        switch (typeBrowser.toLowerCase()){
            case "chrome":
                browser=new Chrome();
                break;
            case "owasp":
                browser=new ChromeZap();
                break;
            case "firefox":
                browser=new FireFox();
                break;
            default:
                browser=new Chrome();
                break;
        }
        return browser;
    }
}
```

```
package browser;
import org.openqa.selenium.Proxy;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.remote.DesiredCapabilities;
import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.TimeUnit;
```

```
public class ChromeZap implements  IBrowser {
    @Override
    public WebDriver create() {
        String
driverPath="C:\\Users\\MARIEL\\IdeaProjects\\claseVulnerabilityTarea3\\src\\test\\resources
\\drivers\\chromedriver.exe";
        System.setProperty("webdriver.chrome.driver",driverPath);

        // informacion para que levante apuntando a un puerto (PROXY)

        // 1 Preferencias
        Map<String,Object> prefs = new HashMap<String,Object>();
        prefs.put("credential_enable_service",false);

        // 2 Proxy --- PORT ---> mismo del OWASP ZAP
        Proxy proxy = new Proxy();
        proxy.setHttpProxy("127.0.0.1:8888");
        proxy.setSslProxy("127.0.0.1:8888");
        // 3 Capabilities
        DesiredCapabilities capabilities = DesiredCapabilities.chrome();
        capabilities.setCapability("proxy",proxy);

        ChromeOptions options = new ChromeOptions();
        options.addArguments("ignore-certificate-errors");
        options.setExperimentalOption("prefs",prefs);

        capabilities.setCapability(ChromeOptions.CAPABILITY,options);
        ChromeDriver driver = new ChromeDriver(capabilities);
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
        return driver;
    }
}
```

5. package → session

Donde se controla la sesión que tenemos.

```
package session;

import browser.FactoryBrowser;
import org.openqa.selenium.WebDriver;
public class Session {
    // singleton
    private static Session session= null;
    private WebDriver driver;

    //constructor privado
    private  Session(){
        driver=
FactoryBrowser.make("owasp").create();
    }
    public static Session getInstance(){
        if (session==null)
            session=new Session();
        return session;
    }
    public void closeBrowser(){
        driver.close();
        session=null;
    }
    public WebDriver getDriver(){
        return driver;
    }
}
```

6. package→control

Creamos los controles de donde esta y como lo poder localizar, este también nos sirve para los reportes donde se realizó la modificación

```
package control;
import org.openqa.selenium.By;
public class Button extends  ControlSelenium {
    public Button(By locator, String myName) {
        super(locator, myName);     }
}
```

```
package control;
import org.openqa.selenium.By;
public class TextBox extends
ControlSelenium {
    public TextBox(By locator, String
myName) {
        super(locator, myName);
    }
}
```

```
package control;
import io.qameta.allure.Step;
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import session.Session;
public class ControlSelenium {
    protected WebElement webElement;
    protected By locator;
    protected String myName;
    public ControlSelenium(By locator,String myName){
        this.locator=locator;
        this.myName=myName;
    }
    @Step("{0}")
    public void allureStep(String action){
    }
    protected void findElement(){
        this.webElement= Session.getInstance().getDriver().findElement(this.locator);
    }

    public void click(){
        this.allureStep("Click on" +this.myName);
        this.findElement();
        this.webElement.click();
    }

    public void type(String value){
        this.allureStep("Type value: "+ value+ " on "+myName);
        this.findElement();
        this.webElement.sendKeys(value);
    }

    public boolean isDisplayOnePage() {
        try {
            this.allureStep("'"+myName+"' is displayed ? ");
            this.findElement();
            return this.webElement.isDisplayed();
        } catch (Exception e) {
            return false;
        }
    }
}
```

7. package→ pages

Los eventos o las páginas que recorrerá nuestra prueba, en este caso el HOME y el del INICIAR SESION.

```
package pages;
  import control.Button;
  import control.TextBox;
  import org.openqa.selenium.By;
public class LoginModalPage {
    public TextBox emailTextBox= new
TextBox(By.id("email"),"[email] textbox
on Login Modal Page");
    public TextBox passwordTextBox= new
TextBox(By.id("password"),"[password]
textbox on Login Modal Page");

    public Button signupButton= new
Button(By.id("//*[@id=\"login_form\"]/but
ton"),"[signup] textbox on Login Modal
Page");
```

```
package pages;
import control.Button;
import org.openqa.selenium.By;

public class MainPage {

    public Button singUpFreeButton= new
Button(By.xpath("//*[@id=\"__next\"]/div/main/d
iv[1]/header/nav/div/ul[2]/li[1]/a"),"[singUp
Free] Button on Main Page");
    public MainPage(){}
}
```

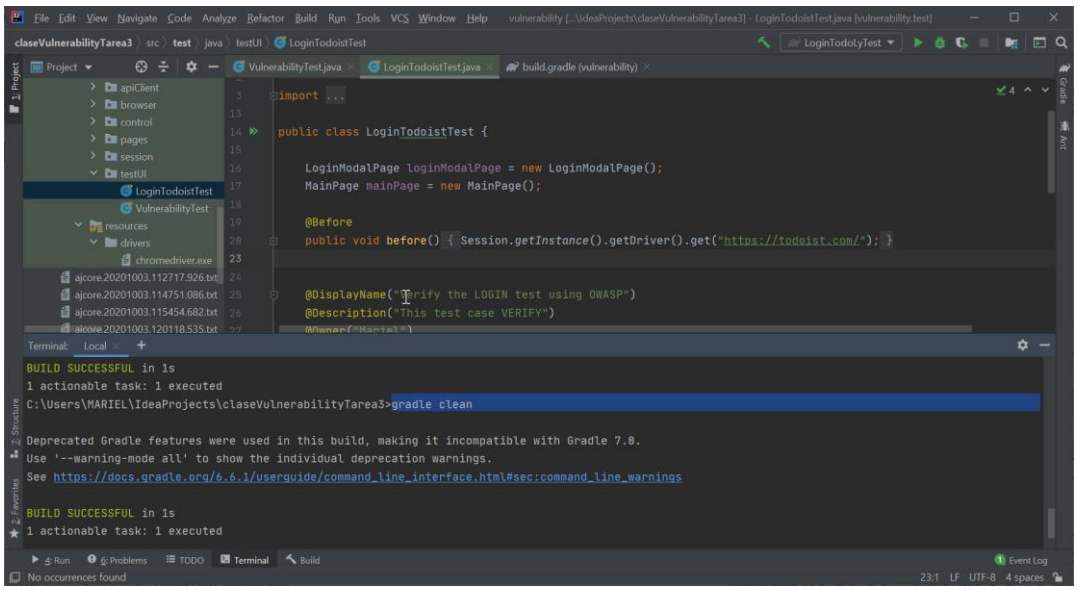
```
public TextBox verifyText = new
TextBox(By.xpath("/html/body/main/div/div
[2]/div/div[2]/div"),"mensaje de error");

    public LoginModalPage(){}

}
```

8. PACKAGE→ testUI

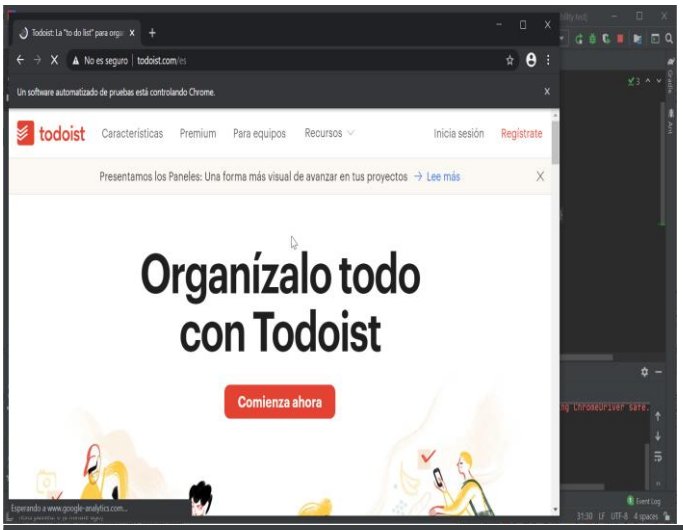
Donde registraremos los test a la pagina

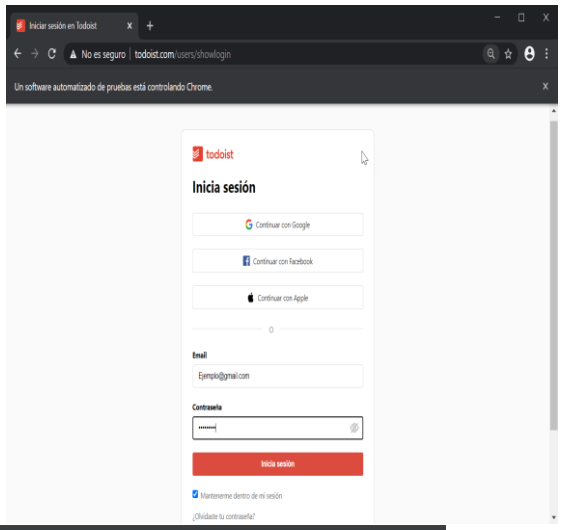
```
package testUI;
import io.qameta.allure.Description;
import io.qameta.allure.Owner;
import io.qameta.allure.junit4.DisplayName;
import org.junit.After;
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
import pages.LoginModalPage;
import pages.MainPage;
import session.Session;
public class LoginTodoistTest {
    LoginModalPage loginModalPage = new LoginModalPage();
    MainPage mainPage = new MainPage();
    @Before
    public void before(){
        Session.getInstance().getDriver().get("https://todoist.com/");
    }
    @DisplayName("Verify the LOGIN test using OWASP")
    @Description("This test case VERIFY")
    @Owner("Mariel")
    @Test
    public void verifyTheLoginUsingUserAndPassword() throws InterruptedException {
        mainPage.singUpFreeButton.click();
        loginModalPage.emailTextBox.type("Ejemplo@gmail.com");
        loginModalPage.passwordTextBox.type("Prueba123");
        loginModalPage.signupButton.click();
        // Verification
          Thread.sleep(500);Assert.assertEquals("ERROR, no se pudo realiza el
login!",false,loginModalPage.verifyText.isDisplayOnePage());     }
    @After
    public void after(){
       Session.getInstance().closeBrowser();
    }
}
```

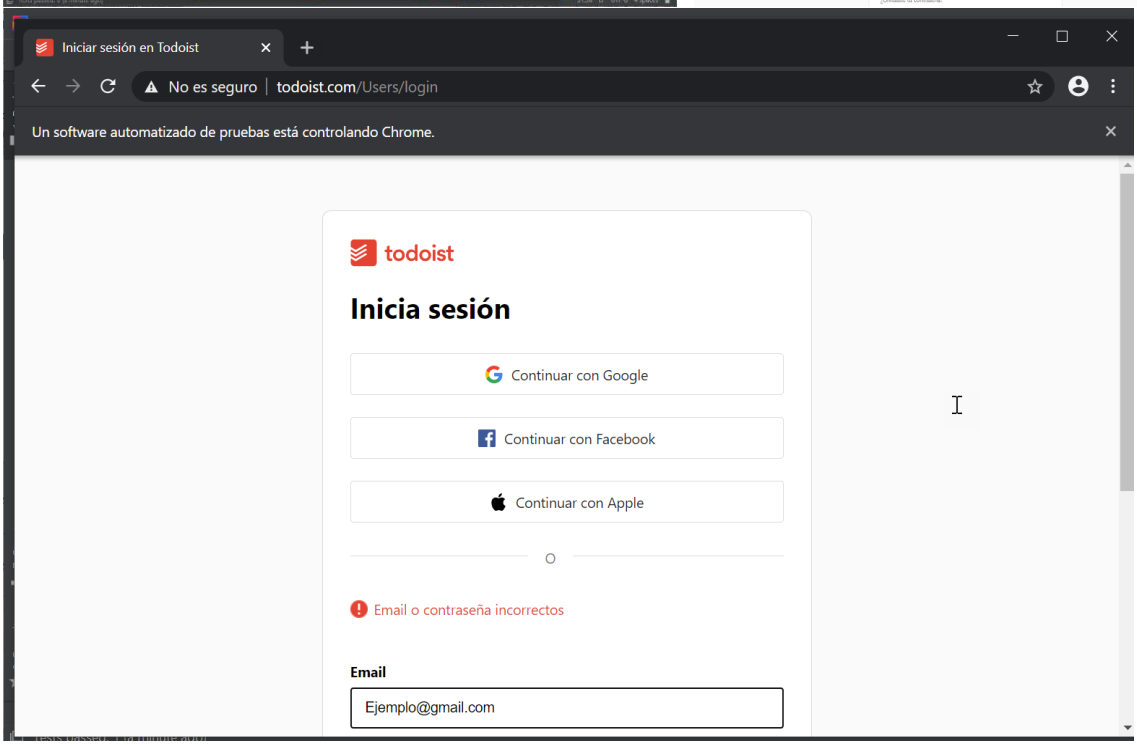
CORRIDA DEL ATAQUE SIMULO EN OWASSAP Y IJ



La corrida le LoginTodoistTest, donde se intentara loguear pero no ingresar por que el correo no esta registrado.







CODIGO PARA EL ANALISIS DE VULNIRABILIDAD

Creacion de package→ apiCliente

```
package apiClient;

import io.restassured.response.Response;

public interface IRequest {
    Response send(String url);
}
```

```
package apiClient;

import io.restassured.http.ContentType;
import io.restassured.response.Response;

import static
io.restassured.RestAssured.given;

public class RequestGET implements IRequest{
    @Override
```

```
    public Response send(String url) {
        Response response = given()

.contentType(ContentType.JSON)
                .when()
                .get(url);
        return response;
    }
}
```

```
package apiClient;

import io.restassured.http.ContentType;
import io.restassured.response.Response;

import static
io.restassured.RestAssured.given;

public class RequestPOST implements
IRequest{
    @Override
    public Response send(String url) {
        Response response = given()

.contentType(ContentType.JSON)
                .when()
                .post(url);
        return response;
    }
}
```

```
package apiClient;

public class FactoryRequest {
    public static IRequest make(String
type){
        IRequest request;

        switch (type.toLowerCase()){
            case "get":
                request= new RequestGET();
                break;
            case "post":
                request = new RequestPOST();
                break;
            default:
                request = new RequestGET();
                break;
        }
        return request;
    }
}
```

Package→testUI

Donde detallamos el progreso del scanner de vulnerabilidad

```
package testUI;
import apiClient.FactoryRequest;
import io.qameta.allure.Attachment;
import io.qameta.allure.Description;
import io.qameta.allure.Owner;
import io.qameta.allure.Step;
import io.qameta.allure.junit4.DisplayName;
import io.restassured.response.Response;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
public class VulnerabilityTest {
    String globalIdScan = "";
    @Before
    public void before(){
    }
    @Test
    @DisplayName("Verify the Vulnability test using OWASP")
    @Description("This test case is to verify the attack of vulnerability using owasp with the
last pluggins")
    @Owner("Mariel")
    public void verifyVulnerabilityScanTest() throws InterruptedException {
     String idScan = startScanningOWASPZAP();
     monitoringStateAttack(idScan);
    }
    @After
    public void after(){
        generateReportOWASPZAP();
    }
    @Attachment(value="{0}", type = "text/html")
    public static String attachHTMLFile(String name, String html){
        return html;
    }
    @Step("Start Vulnerability Test using OWASAP ZAP")
    public String startScanningOWASPZAP(){
        //INICIAR EL SCAN ----> obtener ID
        String
startScanUrl="http://127.0.0.1:8888/JSON/ascan/action/scan/?url=\"https://todoist.com/\"&recurse
=&inScopeOnly=&scanPolicyName=&method=&postData=&contextId=";
```

```
        Response response = (Response) FactoryRequest.make("get").send(startScanUrl);
        response.prettyPrint();
        String scanId = response.then().extract().path("scan");
        System.out.println("ID "+ scanId);
        return scanId;
    }
    @Step("Monitoring Scan of OWASP ZAP 100%")
    public void monitoringStateAttack(String scanId) throws InterruptedException {
        //preguntar si termino el scan y llego al 100%  ---> ID
        String getStateUrl="http://127.0.0.1:8888/JSON/ascan/view/status/?scanId="+scanId;
        String isComplete="";
        while (!isComplete.equals("100")){
            Thread.sleep(30000);
            Response responseStatus= FactoryRequest.make("get").send(getStateUrl);
            isComplete = responseStatus.then().extract().path("status");
            System.out.println("OWASP Status : "+ isComplete+ " %");
        }
    }
    public void generateReportOWASPZAP(){
        //recuperar el summary (Tipos de ataques)
        //html
        String getReportHTML="http://127.0.0.1:8888/OTHER/core/other/htmlreport/";
        Response responseReport=FactoryRequest.make("get").send(getReportHTML);
        String htmlReport=responseReport.body().asString();
        System.out.println("HMTL \n"+htmlReport);
        attachHTMLFile("OWASP Report Vulnerability Detail",htmlReport);
        String
getSummaryReportHTML="http://localhost:8888/HTML/ascan/view/scanProgress/?scanId="+globalIdScan;
        responseReport = FactoryRequest.make("get").send(getSummaryReportHTML);
        String htmlSumaryReport=responseReport.body().asString();
        attachHTMLFile("OWASP Summary Report ",htmlSumaryReport);
        //json
    }
}
```

El limpiado del build en la terminal con el comando > gradle clean

Se debe detener instalado allure y scoop se verifica en cmd

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.1082]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\MARIEL>allure --version
2.13.6
C:\Users\MARIEL>scoop --version
Current Scoop version:
fatal: not a git repository (or any of the parent directories): .git

C:\Users\MARIEL>scoop -version
Usage: scoop <command> [<args>]

Some useful commands are:

alias       Manage scoop aliases
bucket      Manage Scoop buckets
cache       Show or clear the download cache
checkup     Check for potential problems
cleanup     Cleanup apps by removing old versions
config      Get or set configuration values
create      Create a custom app manifest
depends     List dependencies for an app
export      Exports (an importable) list of installed apps
help        Show help for a command
hold        Hold an app to disable updates
home        Opens the app homepage
info        Display information about an app
install     Install apps
list        List installed apps
```

```
//INICIAR EL SCAN ----> obtener ID
String startScanUrl="http://127.0.0.1:8888/JSON/ascan/action/scan/?url=\"https://todoist.com/
Response response = (Response) FactoryRequest.make( type: "get").send(startScanUrl);

response.prettyPrint();
String scanId = response.then().extract().path( path: "scan");
System.out.println("ID "+ scanId);
```

```
Tests passed: 1 of 1 test – 6 m 12 s 497 ms
}
ID 2
OWASP Status : 8 %
OWASP Status : 15 %
OWASP Status : 22 %
OWASP Status : 26 %
OWASP Status : 34 %
OWASP Status : 38 %
OWASP Status : 38 %
OWASP Status : 39 %
OWASP Status : 39 %
OWASP Status : 39 %
OWASP Status : 47 %
OWASP Status : 100 %
HMTL
<html>
  <head>
```

Sesión sin Nombre - OWASP ZAP 2.9.0

https://todoist.com Progreso del escaneo

Fuerza	Progr...	Trans...	Re...	Ale...	E...
SQL Injection	Medio	00:07....	0	0	
Server Side Code Injection	Medio	00:07....	0	0	
Remote OS Command In...	Medio	00:07....	0	0	
Directory Browsing	Medio	00:21....	56	0	
Buffer Overflow	Medio	00:07....	0	0	
Error de formato de cade...	Medio	00:08....	0	0	
CRLF Injection	Medio	00:06....	0	0	
Parameter Tampering	Medio	00:09....	0	0	
Reglas de búsqueda acti...	Medio	00:00....	0	0	
Source Code Disclosure ...	Medio	00:11....	0	0	
Source Code Disclosure ...	Medio	00:06....	0	0	
Ejecución remota de cód...	Medio	00:06....	0	0	
Httpoxy - Proxy Header ...	Medio	02:29....	784	0	
Anti-CSRF Tokens Check	Medio	00:05....	1	0	
Desconfiguración de Do...	Medio	00:00....	2	0	
Vulnerabilidades de Ope...	Medio	00:01....	3	0	
Divulgación del código f...	Medio	00:04....	8	0	
Ejecución remota de cód...	Medio	00:13....	112	0	
Fijación de Sesión	Medio	00:03....	0	0	
Inyección SQL - MySQL	Medio		0	0	
Inyección SQL - SQL hip...	Medio		0	0	
Inyección SQL - Oráculo	Medio		0	0	
Inyección SQL - Postgre...	Medio		0	0	
SQL Injection - SQLite	Medio		0	0	

ID	Marca de tiempo Req	Marca de tiempo de Resp	Método	URL	Código	Razón	RTT
2.611	10/10/20 17:01:14	10/10/20 17:01:15	GET	https://todoist.com/_next/static	404	Not Found	257milis
2.612	10/10/20 17:01:15	10/10/20 17:01:15	GET	https://todoist.com/_next/static/chunks	404	Not Found	344milis
2.613	10/10/20 17:01:15	10/10/20 17:01:15	GET	https://todoist.com/_next/static/chunks/pag...	404	Not Found	248milis
2.614	10/10/20 17:01:17	10/10/20 17:01:17	GET	https://todoist.com/_next/static/css	404	Not Found	331milis
2.615	10/10/20 17:01:17	10/10/20 17:01:17	GET	https://todoist.com/_next/static/css/pages	404	Not Found	217milis
2.616	10/10/20 17:01:17	10/10/20 17:01:17	GET	https://todoist.com/_next/static/ETGqYAt-O...	404	Not Found	260milis
2.617	10/10/20 17:01:17	10/10/20 17:01:18	GET	https://todoist.com/_next/static/images	404	Not Found	280milis...

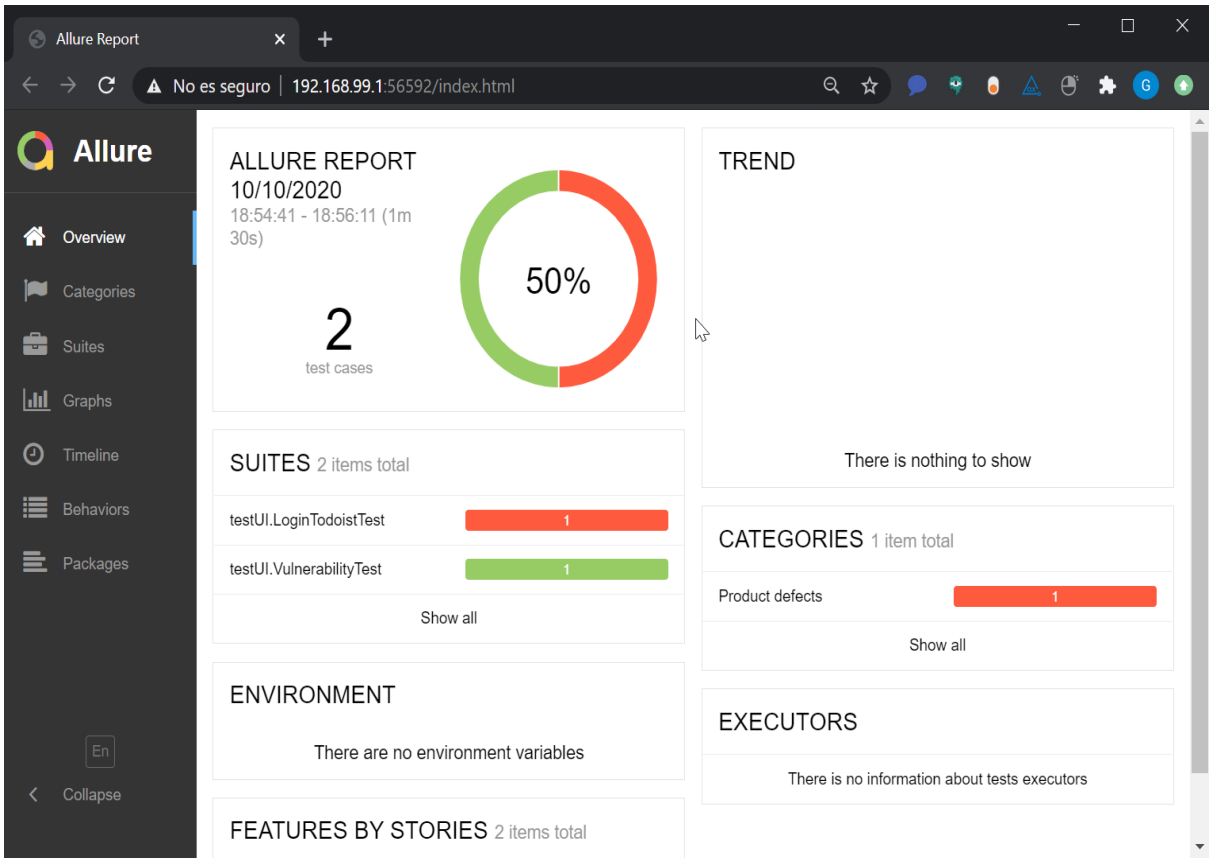
- gradle clean test

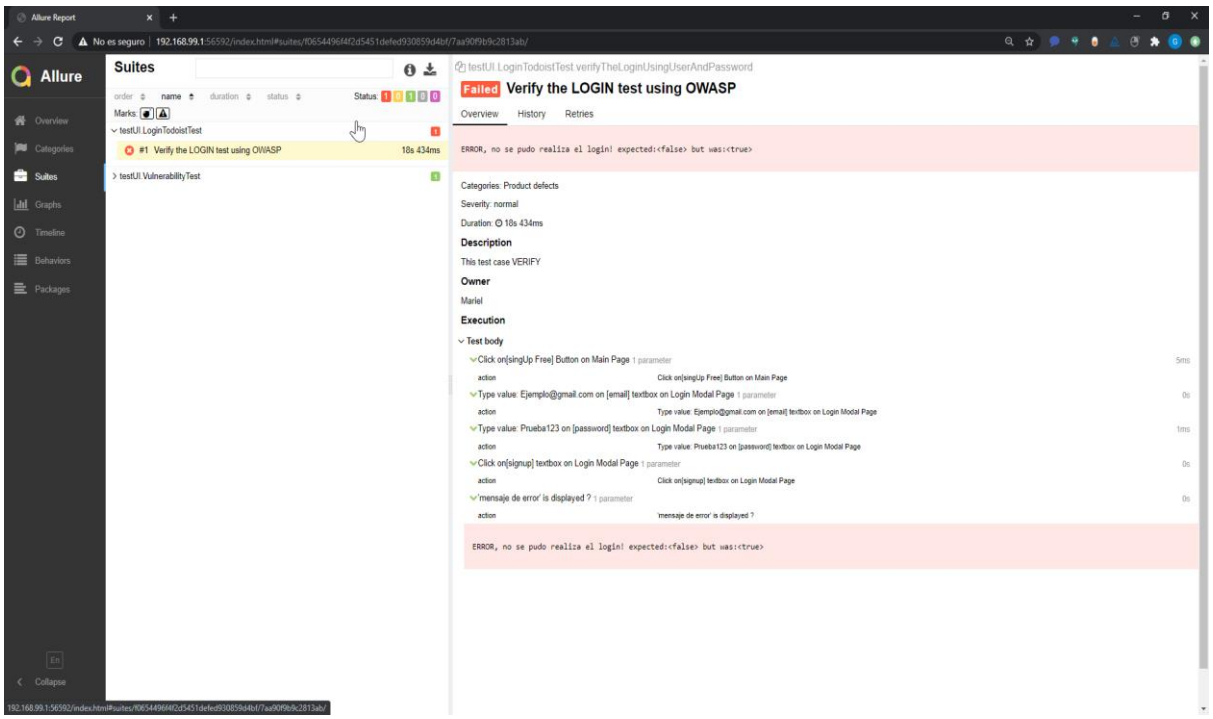
```
FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':test'.
> There were failing tests. See the report at: file:///C:/Users/MARIEL/IdeaProjects/claseVulnerabilityTarea3/build/reports/tests/test/index.html

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.6.1/userguide/command-line_interface.html#sec:command_line_warnings

BUILD FAILED in 1m 42s
4 actionable tasks: 4 executed
```

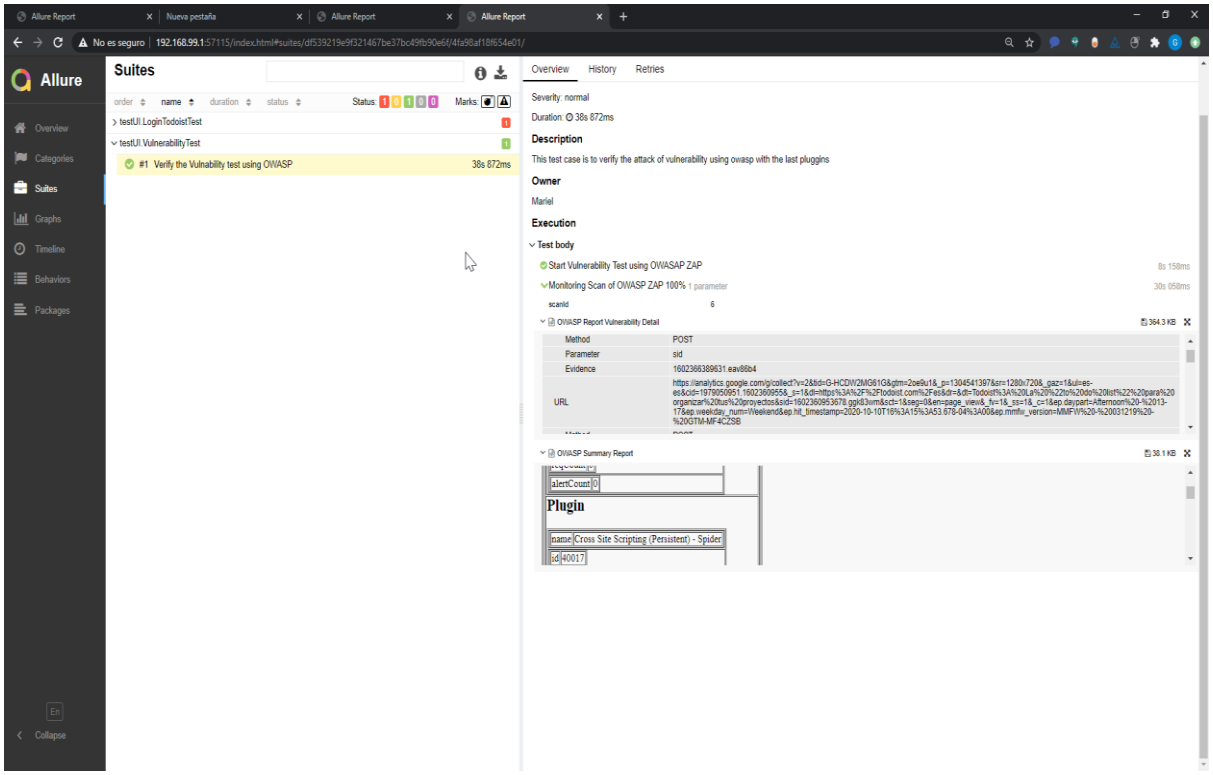
- allure serve build\allure-results

```
BUILD FAILED in 1m 42s
4 actionable tasks: 4 executed

C:\Users\MARIEL\IdeaProjects\claseVulnerabilityTarea3>allure serve build\allure-results
Generating report to temp directory...
Report successfully generated to C:\Users\MARIEL\AppData\Local\Temp\7664526082419891999\allure-report
Starting web server...
2020-10-10 19:00:17.569:INFO::main: Logging initialized @6459ms to org.eclipse.jetty.util.log.StdErrLog
Server started at <http://192.168.99.1:56592/>. Press <Ctrl+C> to exit
```

Resultado de la ejecución de vulnerabilidad y del test de login, allure es una herramienta que te permite ver de manera grafica los resultados obtenidos.






Allure Report
Nueva pestaña
Allure Report
Allure Report
No es seguro | 192.168.99.1:57115/index.html#suites/df539219e9f321467be37bc49fb90e6f/4fa98af18f654e01/
Allure
Overview
Categories
Suites
Graphs
Timeline
Behaviors
Packages
Collapse
Suites
order
name
duration
status
Status: 1 0 1 0 0
Marks:
testUI.LoginTodoistTest
testUI.VulnerabilityTest
#1 Verify the Vulnability test using OWASP
38s 872ms
Overview
History
Retries
Severity: normal
Duration: 38s 872ms
Description
This test case is to verify the attack of vulnerability using owasp with the last pluggins
Owner
Mariel
Execution
Test body
Start Vulnerability Test using OWASAP ZAP
8s 158ms
Monitoring Scan of OWASP ZAP 100% 1 parameter
30s 058ms
scanId
6
OWASP Report Vulnerability Detail
364.3 KB
Method
POST
Parameter
sid
Evidence
160236638963 1.eav66b4
URL
OWASP Summary Report
38.1 KB
alertCount 0
Plugin
name Cross Site Scripting (Persistent) - Spider
id 40017