www.itsecgames.com	**GNU GPL**	**Bug report # 1**
software: bee-box/bWAPP	**© 2014**	**version: 2.2**
Report type (1-2): 2	**Severity (1-4): 1**	**Additions (y/n): y**
1 - problem	**1 - Critical**	**If yes, which ones?**
2 - solution	**2 - Major**	**screenshots**
	3 - Minor	**UML tests diagram**
	4 - Cosmetic	

FEATURE:

A1 HTML injection reflected (GET/POST) security level: low

PROBLEM:

Mismatch between actual behavior of the application and its expected behavior.

CAN YOU REPRODUCE THE PROBLEM SITUATION? (y/n):

Yes

PROBLEM DESCRIPTION:

Text input form does not support HTML tag validation.

TEST CASE:

1. Get the url: http://192.168.0.104/bWAPP/htmli_get.php
2. Fill in the form with data:

   First name: “<h1>Hacked</h1>”;

   Last name: “<p></p>”;
3. Submit the form;
4. Check the actual behavior - html tag h1 was added into the page source.

SUGGESTED CORRECTION (OPTIONAL):

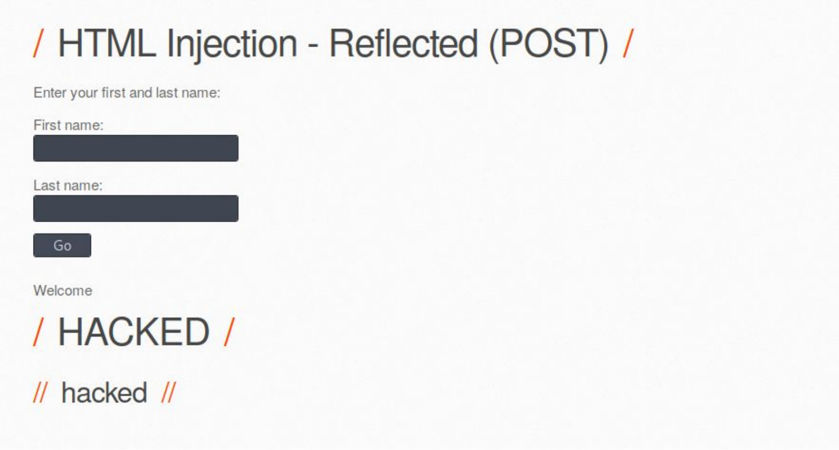
Data that has been entered into the form must be validate.

provided by: **Prytula Nick** **date:** **28.01.2019**

A1 injection: HTML injection - reflected(GET/POST)

Bug report #1 additions;

Actual behavior:



/ HTML Injection - Reflected (POST) /

Enter your first and last name:

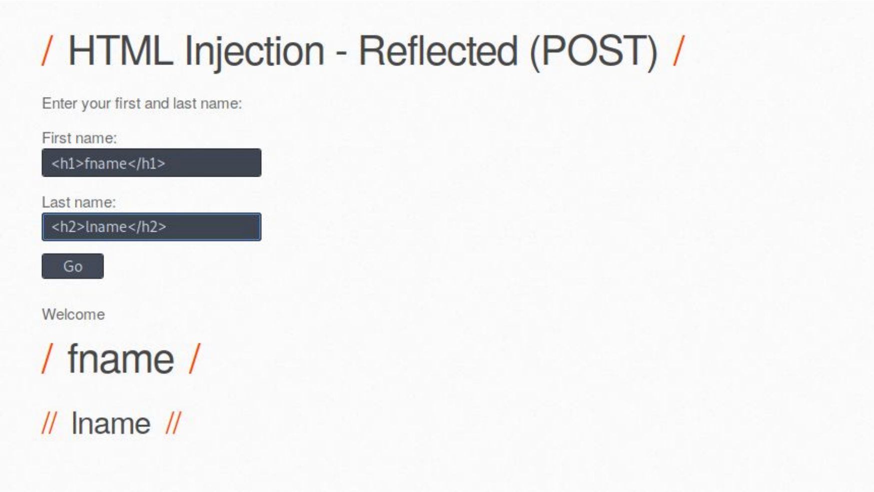
First name:

Last name:

Welcome

/ HACKED /

// hacked //



/ HTML Injection - Reflected (POST) /

Enter your first and last name:

First name: <h1>fname</h1>

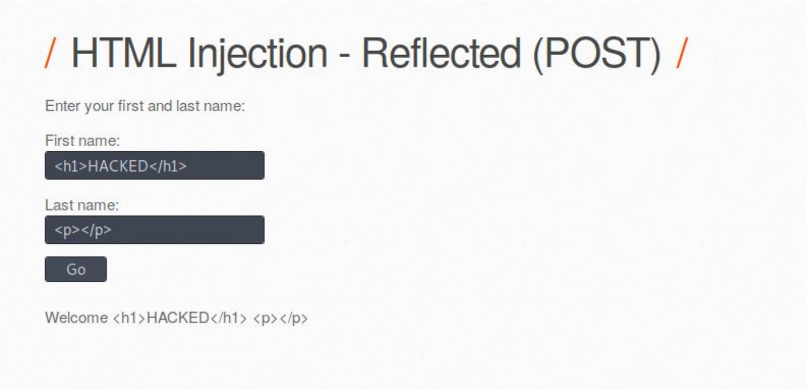
Last name: <h2>lname</h2>

Welcome

/ fname /

// lname //

Expected behavior:



/ HTML Injection - Reflected (POST) /

Enter your first and last name:

First name: <h1>HACKED</h1>

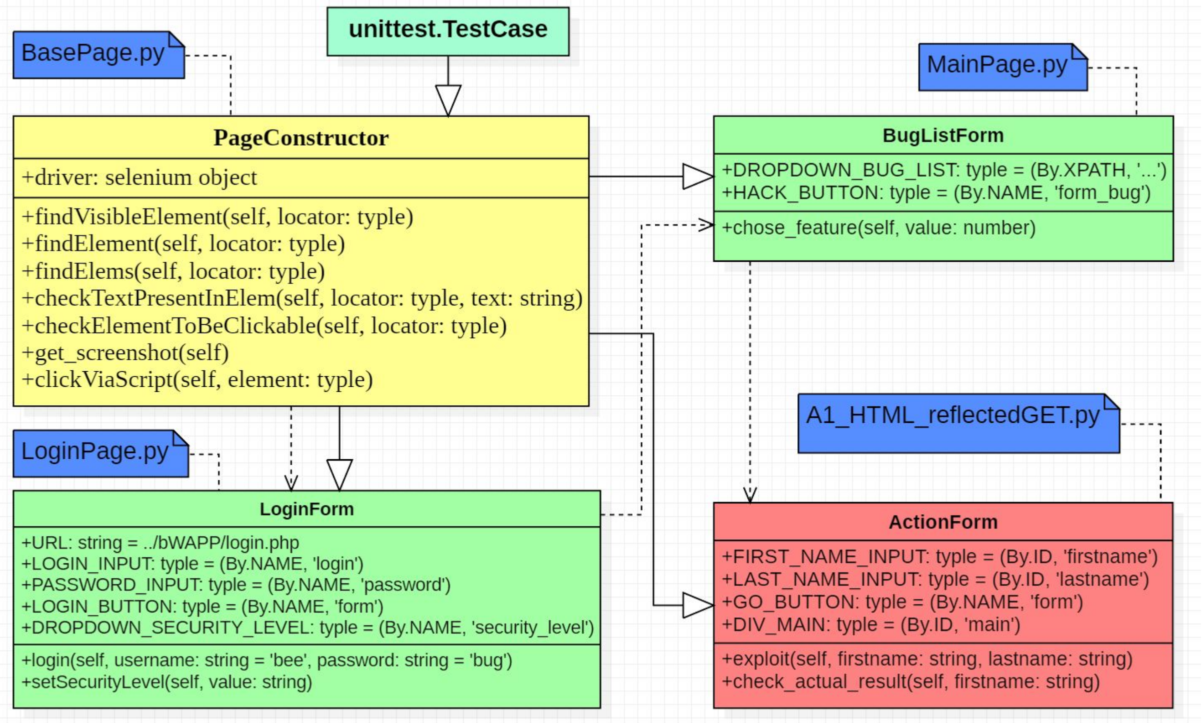
Last name: <p></p>

Welcome <h1>HACKED</h1> <p></p>

SUMMARY:

HTML injection is a type of injection issue that occurs when a user is able to control an input point and is able to inject arbitrary HTML code into a vulnerable web page. This vulnerability can have many consequences, like disclosure of a user's session cookies that could be used to impersonate the victim, or, more generally, it can allow the attacker to modify the page content seen by the victims.

OWASP top 10 A1 html injection; bWAPP automation tests, page element model.



BasePage.py

unittest.TestCase

MainPage.py

PageConstructor

+driver: selenium object

+findVisibleElement(self, locator: typle)
+findElement(self, locator: typle)
+findElems(self, locator: typle)
+checkTextPresentInElem(self, locator: typle, text: string)
+checkElementToBeClickable(self, locator: typle)
+get_screenshot(self)
+clickViaScript(self, element: typle)

BugListForm

+DROPDOWN_BUG_LIST: typle = (By.XPATH, '...')
+HACK_BUTTON: typle = (By.NAME, 'form_bug')

+chose_feature(self, value: number)

A1_HTML_reflectedGET.py

LoginPage.py

LoginForm

+URL: string = ../bWAPP/login.php
+LOGIN_INPUT: typle = (By.NAME, 'login')
+PASSWORD_INPUT: typle = (By.NAME, 'password')
+LOGIN_BUTTON: typle = (By.NAME, 'form')
+DROPDOWN_SECURITY_LEVEL: typle = (By.NAME, 'security_level')

+login(self, username: string = 'bee', password: string = 'bug')
+setSecurityLevel(self, value: string)

ActionForm

+FIRST_NAME_INPUT: typle = (By.ID, 'firstname')
+LAST_NAME_INPUT: typle = (By.ID, 'lastname')
+GO_BUTTON: typle = (By.NAME, 'form')
+DIV_MAIN: typle = (By.ID, 'main')

+exploit(self, firstname: string, lastname: string)
+check_actual_result(self, firstname: string)