INTERNSHIP REPORT

Name: Ravinshu Chauhan

Course: Bachelor's of Computer Application

Branch & Year: Information Technology (Graduate)

University: Maharshi Dayanand University, Rohtak (124001), Haryana

Internship Title: **Penetration Tester Intern**

Company Name: DeltaWare Solution Pvt. Ltd.

Duration: 13 August 2025 – 13 September 2025 (01 Month)

Location: Banda, Kanpur, Uttar Pradesh (220001)

Under the Guidance of: Mr. Anuj Kumar Dwivedi

(Founder & CEO at DeltaWare Solution Pvt.Ltd)

Submitted by:

Ravinshu Chauhan

Submitted To:

Deltaware Solution Pvt Ltd

Acknowledgement

I am grateful to **Deltaware Solution Pvt. Ltd.** for offering me the opportunity to work as a **Penetration Testing Intern**. The internship helped me gain hands-on experience in identifying vulnerabilities, performing security assessments, and applying industry best practices.

My sincere thanks to **Mr. Anuj Kumar Dwivedi** for his mentorship and guidance, I also appreciate the support of my colleagues and peers during this internship journey.

Abstract

This project explores the use of **Browser automation for penetration testing** using Python and Selenium. The objective is to demonstrate how automated scripts can interact with websites, detect login forms, attempt test credentials, and capture screenshots for evidence collection. This beginner-level project highlights the practical application of automation in cybersecurity by simulating real-world testing scenarios. The outcome emphasizes efficiency, reproducibility, and accuracy in web application security testing.

Company Profile

Deltaware Solution Pvt. Ltd. is a leading organization in the field of **cybersecurity and web development**, dedicated to delivering secure and innovative IT solutions. The company was established on **11 April 2025** in **Banda, Uttar Pradesh** (ROC: Kanpur) by **Mr. Anuj Kumar Dwivedi**, a seasoned cybersecurity professional with over four years of expertise, along with **Mr. Ashutosh Dwivedi** as the co-founder.

Legally registered under **CIN: U62099UP2025PTC221138**, Deltaware operates with a mission to strengthen the digital ecosystem through advanced security practices and modern web technologies. With a team of skilled professionals, the company offers a broad spectrum of services, including:

- Cybersecurity solutions
- Software and web development
- Network integration
- Cloud infrastructure management
- IT support and business intelligence solutions

Our Mission

To strengthen the digital landscape by providing top-notch security solutions and innovative web services, while maintaining integrity, innovation, and excellence.

Internship Objectives

The objectives of my Penetration Testing Internship at **Deltaware Solution Pvt. Ltd.** included:

- Acquiring **hands-on experience** in penetration testing and vulnerability assessment.
- Applying classroom learning in **ethical hacking, networking, and system security** to real-world cases.
- Building technical expertise in **identifying, exploiting, and mitigating security flaws**.
- Enhancing knowledge of **tools, frameworks, and methodologies** used in professional pentesting.
- Understanding the **practical workflow of cybersecurity engagements**, from reconnaissance to reporting.

Introduction

In modern penetration testing, automation plays a crucial role in streamlining repetitive tasks such as login testing, form submission, and screenshot capture. While professional tools like **Burp Suite** and **OWASP ZAP** offer advanced capabilities, custom automation provides flexibility and control.

The purpose of this project is to:

- Automate interaction with a target website.
- Detect and test login forms with dummy credentials.
- Collect and document evidence automatically.

This project bridges the gap between **manual testing** and **automated security assessments**, providing an entry point for cybersecurity students into automation-driven penetration testing.

Framework of Ideas

- **Burp Suite:** Widely used for manual and automated web testing. However, it may require licensing for advanced features.
- **OWASP ZAP:** Open-source tool offering automated scanning but limited in customization.
- **Selenium:** Primarily used for web application testing, but its ability to control browsers makes it a valuable tool for security testers who want to automate specific tasks.

This project builds on the concept of Selenium automation to create a simple, custom security testing tool.

Tools & Technologies Used

- **Python 3** – scripting language for automation.
- **Selenium** – browser automation framework.
- **Google Chrome + ChromeDriver** – web browser and driver for executing automation tasks.
- **Test Website:** http://testphp.vulnweb.com (vulnerable demo site for security testing).

Reason for Selection:

- Python offers simplicity and readability.
- Selenium supports interaction with dynamic web elements.
- The vulnerable demo site allows safe testing without legal concerns.

System Architecture / Workflow

Workflow of the Automation Script:

User → Python Script → Selenium → Chrome Browser → Target Website → Output (Screenshot + Logs)

Methodology

Step 1: Environment Setup

- Installed Python 3 and Selenium.
- Downloaded and configured Chrome Driver.

Step 2: Script Development

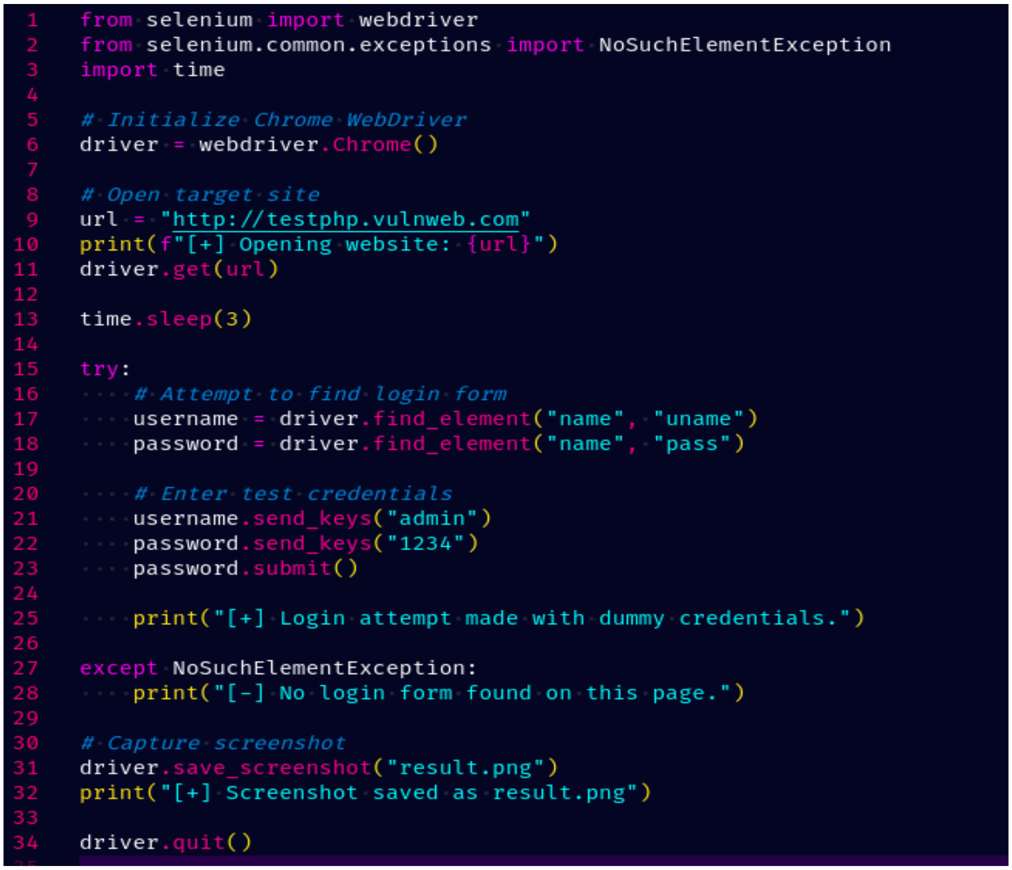
- The script was designed to:
    - Open the target website.
    - Search for a login form.
    - Attempt login with dummy credentials (admin: 1234).
    - Capture a screenshot of the result.

Step 3: Execution

- Executed on the target demo site testphp.vulnweb.com.
- The script successfully opened the site and attempted form interaction.

Code Reference

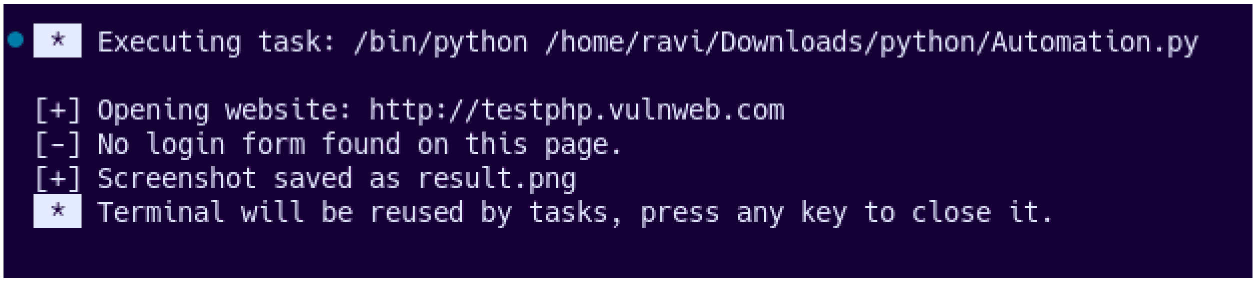
Python:



```python
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
import time

# Initialize Chrome WebDriver
driver = webdriver.Chrome()

# Open target site
url = "http://testphp.vulnweb.com"
print(f"[+] Opening website: {url}")
driver.get(url)

time.sleep(3)

try:
    # Attempt to find login form
    username = driver.find_element("name", "uname")
    password = driver.find_element("name", "pass")

    # Enter test credentials
    username.send_keys("admin")
    password.send_keys("1234")
    password.submit()

    print("[+] Login attempt made with dummy credentials.")

except NoSuchElementException:
    print("[-] No login form found on this page.")

# Capture screenshot
driver.save_screenshot("result.png")
print("[+] Screenshot saved as result.png")

driver.quit()
```

Results & Analysis

Terminal Output Example:



```
 *  Executing task: /bin/python /home/ravi/Downloads/python/Automation.py

[+] Opening website: http://testphp.vulnweb.com
[-] No login form found on this page.
[+] Screenshot saved as result.png
 *  Terminal will be reused by tasks, press any key to close it.
```

Generated Screenshot:

- The script created result.png, showing the page after execution.

Analysis:

- Demonstrates Selenium's capability to interact with websites.
- Can detect login forms and attempt credential testing.
- Provides evidence collection through automated screenshots.

Security Implications

- Automated login testing helps penetration testers quickly check for **weak login mechanisms**.
- Automation reduces manual effort and ensures consistency across multiple test cases.
- While basic, this project demonstrates how security tasks can be scaled for **larger assessments**.

Challenges Faced & Solutions

- **ChromeDriver Version Mismatch:** Fixed by downloading a driver matching the Chrome browser version.
- **Element Detection Issues:** Solved using Selenium locators (name, id, xpath).
- **Delays in Page Loading:** Addressed by adding time.sleep() waits.

Conclusion

This project demonstrates how **Python + Selenium** can be used in cybersecurity for browser automation. Although simple, it highlights the practical application of automation in penetration testing tasks such as login testing and evidence collection. Future improvements could include:

- Multi-site scanning.
- Advanced login brute-force testing.
- Integration with reporting tools like Burp Suite.

References

- OWASP Testing Guide: https://owasp.org
- Selenium Documentation: https://www.selenium.dev/documentation
- Test Website: http://testphp.vulnweb.com

Appendix

- Full Python script.
- Output screenshot files.