**A blue and orange logo

Description automatically generated**

**Web Security- IE2062**

**Journal**

Ashan G Punchihewa

IT22327680

Contents

[2 Tools and technologies used 6](#_Toc165560822)

[2.1 ShodanX 6](#_Toc165560823)

[2.2 Nikto 7](#_Toc165560824)

[2.3 Whatweb 8](#_Toc165560825)

[2.4 Dirb 9](#_Toc165560826)

[2.5 Wafw00f 10](#_Toc165560827)

[2.6 Xss\_vibes 11](#_Toc165560828)

[2.7 Katana 12](#_Toc165560829)

[2.8 Burp Suite 13](#_Toc165560830)

[2.9 4-ZERO-3 Bypass 14](#_Toc165560831)

[2.10 Webster 15](#_Toc165560832)

[2.11 ParamSpider 16](#_Toc165560833)

[2.12 OpenRedirex 17](#_Toc165560834)

[2.13 SSRFmap 18](#_Toc165560835)

[2.14 JS2PDFInjector 19](#_Toc165560836)

[3 Cross-Site Scripting 20](#_Toc165560837)

[3.1 Date: 04/25/2024 20](#_Toc165560838)

[3.2 Summary of the day's activities: 20](#_Toc165560839)

[3.3 Vulnerabilities discovered or explored: 20](#_Toc165560840)

[3.4 Challenges faced and how they were overcome: 20](#_Toc165560841)

[3.5 New tools, techniques, or concepts learned: 20](#_Toc165560842)

[3.6 Reflections and takeaways: 21](#_Toc165560843)

[4 Web Cache Poisoning 22](#_Toc165560844)

[4.1 Date: 04/26/2024 22](#_Toc165560845)

[4.2 Summary of the day’s activities: 22](#_Toc165560846)

[4.3 Vulnerabilities discovered: 22](#_Toc165560847)

[4.4 Challenges faced and how they were overcome: 22](#_Toc165560848)

[4.5 New tools, techniques and concepts learned: 22](#_Toc165560849)

[4.6 Reflections and Takeaways: 22](#_Toc165560850)

[5 Response Manipulation & Abusing Password Reset Functionality 23](#_Toc165560851)

[5.1 Date: 04/27/2024 23](#_Toc165560852)

[5.2 Summary of the Day’s Activities: 23](#_Toc165560853)

[5.3 Vulnerabilities Discovered: 23](#_Toc165560854)

[5.4 Challenges Faced and How They Were Overcome: 23](#_Toc165560855)

[5.5 New Tools, Techniques, or Concepts Learned: 23](#_Toc165560856)

[5.6 Reflections and Takeaways: 24](#_Toc165560857)

[6 SQL Injection 25](#_Toc165560858)

[6.1 Date: 04/27/2024 25](#_Toc165560859)

[6.2 Summary of the Day’s Activities: 25](#_Toc165560860)

[6.3 Vulnerabilities Discovered: 25](#_Toc165560861)

[6.4 Challenges faced and how they were overcome: 25](#_Toc165560862)

[6.5 New Tools, Techniques, or Concepts Learned: 25](#_Toc165560863)

[6.6 Reflections and Takeaways: 25](#_Toc165560864)

[7 Open Redirection 26](#_Toc165560865)

[7.1 Date: 04/28/2024 26](#_Toc165560866)

[7.2 Summary of the Day's Activities: 26](#_Toc165560867)

[7.3 Vulnerabilities Discovered: 26](#_Toc165560868)

[7.4 Challenges Faced and How They Were Overcome: 26](#_Toc165560869)

[7.5 New Tools, Techniques, or Concepts Learned: 26](#_Toc165560870)

[7.6 Reflections and Takeaways: 26](#_Toc165560871)

[8 Cross Site Request Forgery (CSRF) 27](#_Toc165560872)

[8.1 Date: 04/29/2024 27](#_Toc165560873)

[8.2 Summary of the Day’s Activities: 27](#_Toc165560874)

[8.3 Vulnerabilities Discovered: 27](#_Toc165560875)

[8.4 Challenges Faced and how they were overcome: 27](#_Toc165560876)

[8.5 New Tools, Techniques, or Concepts Learned: 27](#_Toc165560877)

[8.6 Reflections and Takeaways: 28](#_Toc165560878)

[9 Race Conditions 29](#_Toc165560879)

[9.1 Date: 04/30/2024 29](#_Toc165560880)

[9.2 Summary of the Day’s Activities: 29](#_Toc165560881)

[9.3 Vulnerabilities Discovered: 29](#_Toc165560882)

[9.4 Challenges Faced and how they were overcome: 29](#_Toc165560883)

[9.5 New Tools, Techniques, or Concepts Learned: 29](#_Toc165560884)

[9.6 Reflections and Takeaways: 29](#_Toc165560885)

[10 Server-Side Request Forgery (SSRF) 30](#_Toc165560886)

[10.1 Date: 05/01/2024 30](#_Toc165560887)

[10.2 Summary of the Day's Activities: 30](#_Toc165560888)

[10.3 Vulnerabilities Discovered: 30](#_Toc165560889)

[10.4 Challenges Faced and How They Were Overcome: 30](#_Toc165560890)

[10.5 New Tools, Techniques, or Concepts Learned: 30](#_Toc165560891)

[10.6 Reflections and Takeaways: 30](#_Toc165560892)

[11 Directory Traversal 31](#_Toc165560893)

[11.1 Date: 05/01/2024 31](#_Toc165560894)

[11.2 Summary of the Day’s Activities: 31](#_Toc165560895)

[11.3 Vulnerabilities Discovered: 31](#_Toc165560896)

[11.4 Challenges Faced and How They Were Overcome: 31](#_Toc165560897)

[11.5 New Tools, Techniques or Concepts Learned: 31](#_Toc165560898)

[11.6 Reflections and Takeaways: 32](#_Toc165560899)

[12 File Upload leading to JavaScript Code Execution 33](#_Toc165560900)

[12.1 Date: 05/01/2024 33](#_Toc165560901)

[12.2 Summary of the Day’s Activities: 33](#_Toc165560902)

[12.3 Vulnerabilities Discovered: 33](#_Toc165560903)

[12.4 Challenges Faced and how they were overcome: 33](#_Toc165560904)

[12.5 New tools, techniques, or concepts learned: 33](#_Toc165560905)

[12.6 Reflections and takeaways: 33](#_Toc165560906)

# Tools and technologies used

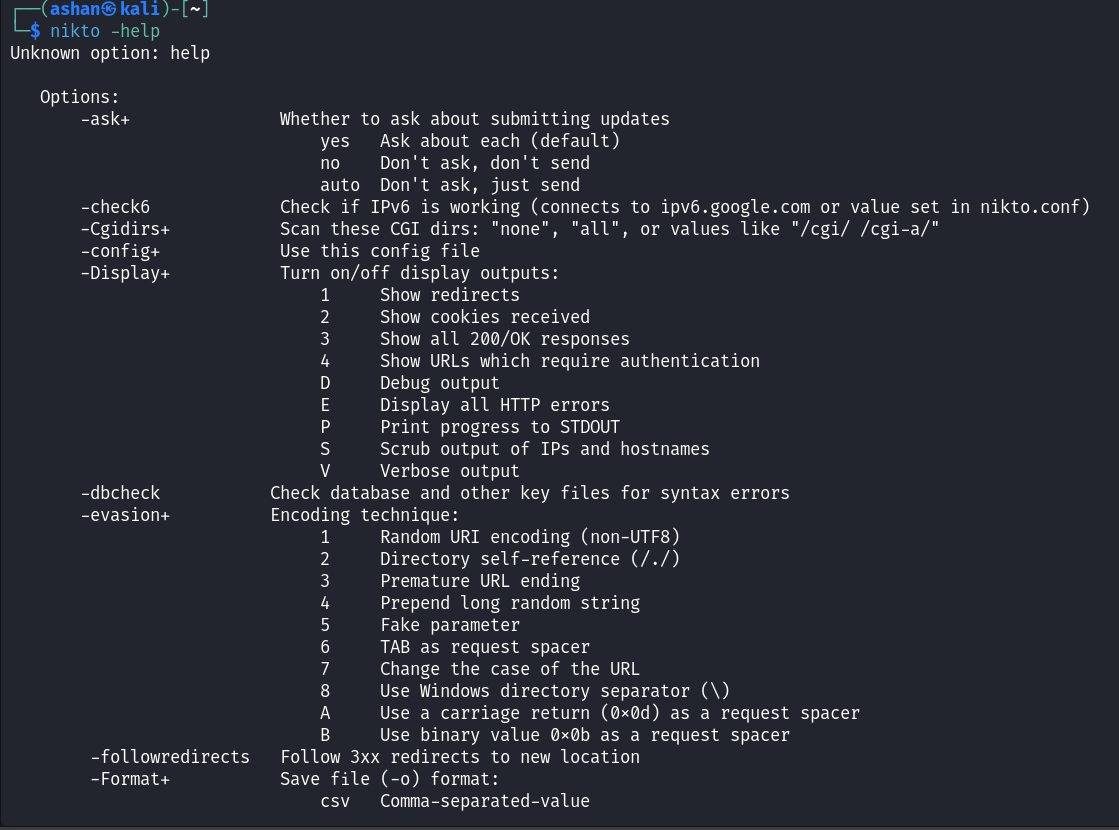
## ShodanX

ShodanX is a tool, for gathering information that taps into Shodan’s database. It offers modes and flexible queries to uncover insights for security assessments, reconnaissance and threat intelligence. With its output and user-friendly commands ShodanX enables users to efficiently collect and analyze data from Shodan’s different facets boosting their cybersecurity efforts.

Compared to Shodan, ShodanX is more user friendly as it doesn't require paid API keys. This means that anyone can access Shodan’s database of internet connected devices without any cost involved. It essentially provides the benefits of Shodan of charge making it accessible to an audience. Additionally, ShodanX delivers query results in time allowing users to obtain the up-to-date information quickly and effortlessly. In essence it presents a budget friendly option, for individuals interested in network reconnaissance and security analysis.

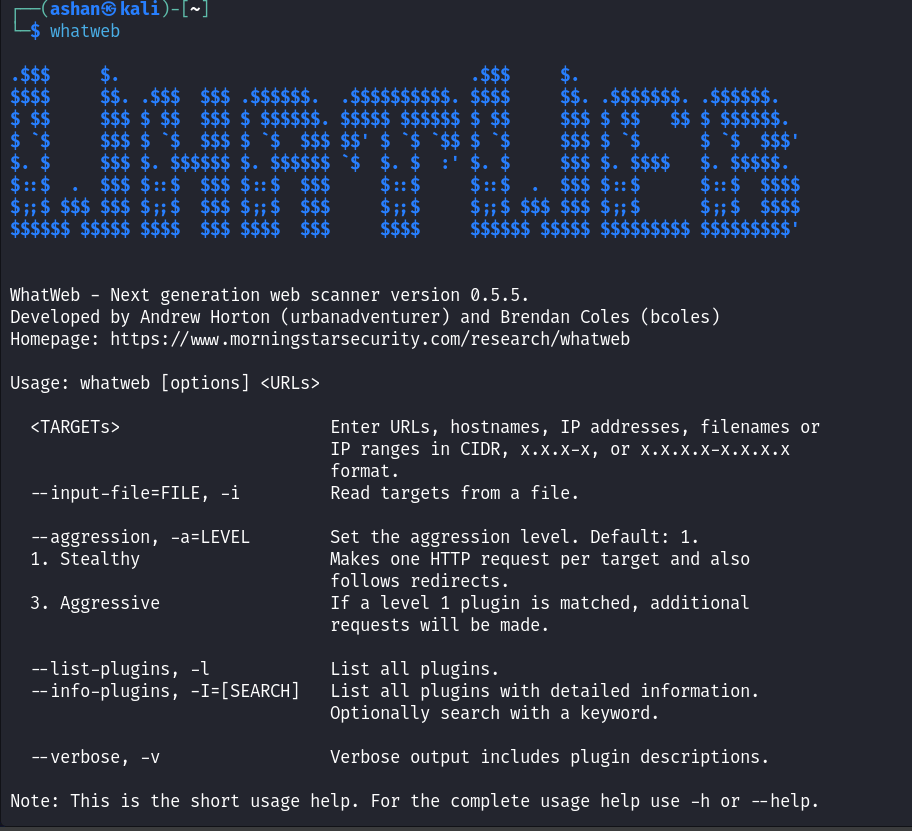


## Nikto

Nikto, which is freely available, under the GPL license is a tool used for scanning web servers to detect vulnerabilities. It checks for security risks such, as outdated software versions and potentially harmful files or programs.

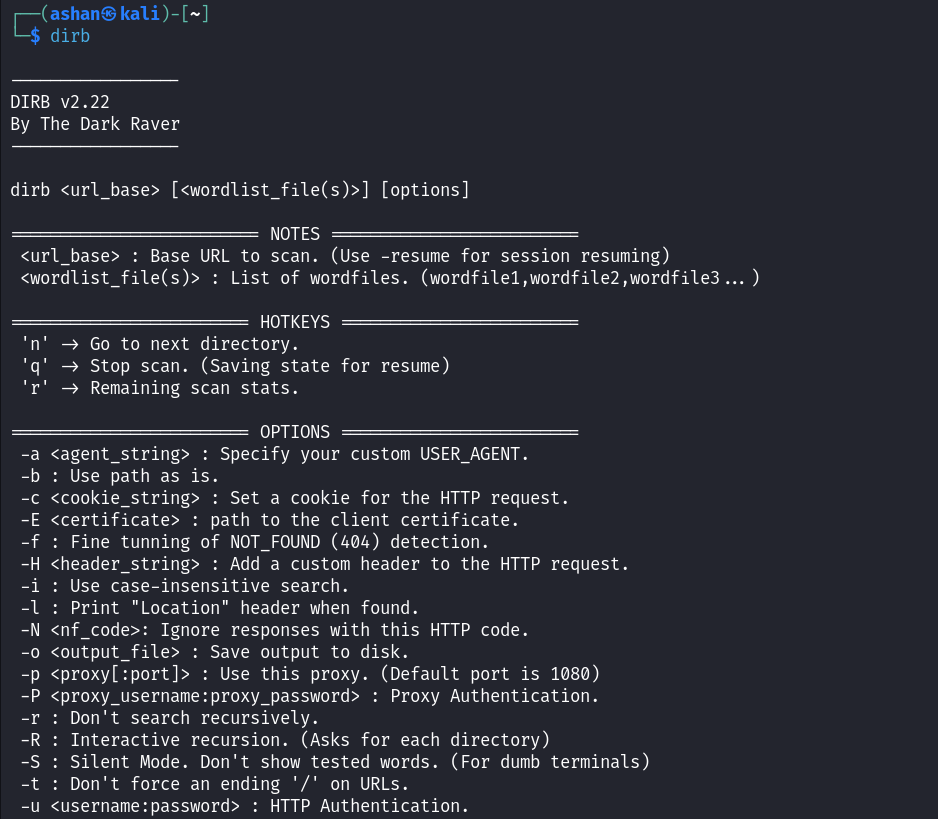
## Whatweb

The WhatWeb open-source tool is designed to identify websites and assess their vulnerability. It is utilized to analyze a websites fingerprint by detecting its applications, web servers and other technologies. This tool can inspect a webpage. Examine the HTTP headers and HTML source of a webpage to recognize the technologies employed on a website.



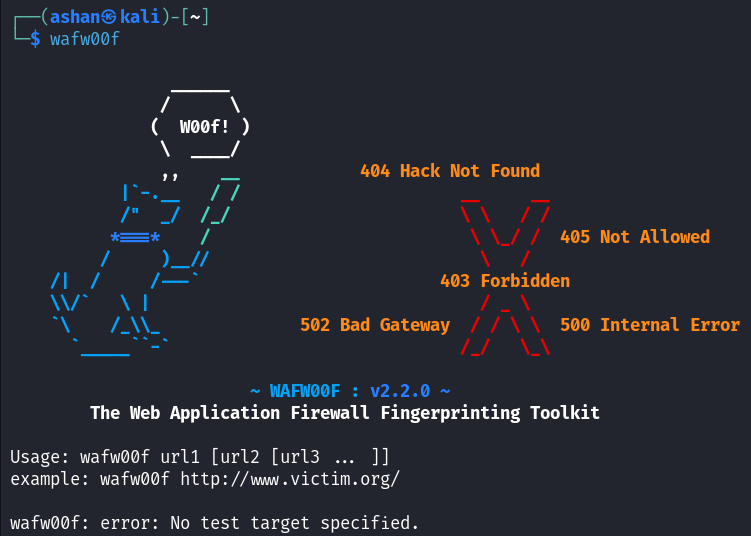
## Dirb

Dirb is a tool used to scan directories searching for concealed files, directories and pages, on web servers. It's available for free, as part of Kali Linux, an open source software suite.

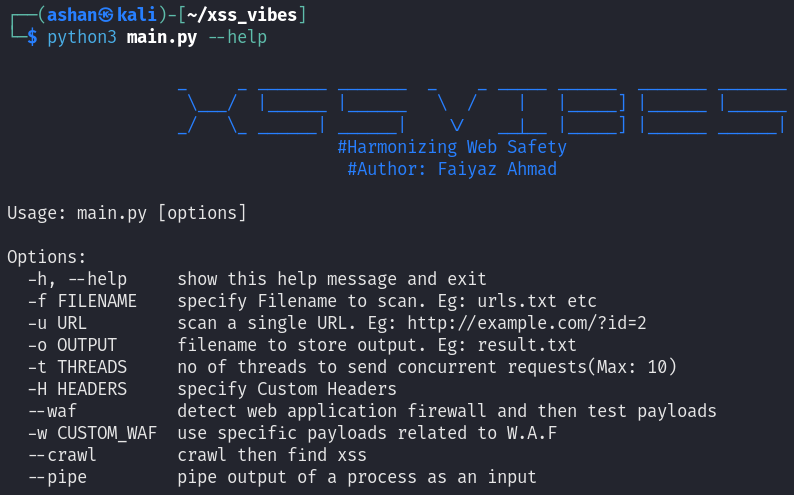


## Wafw00f

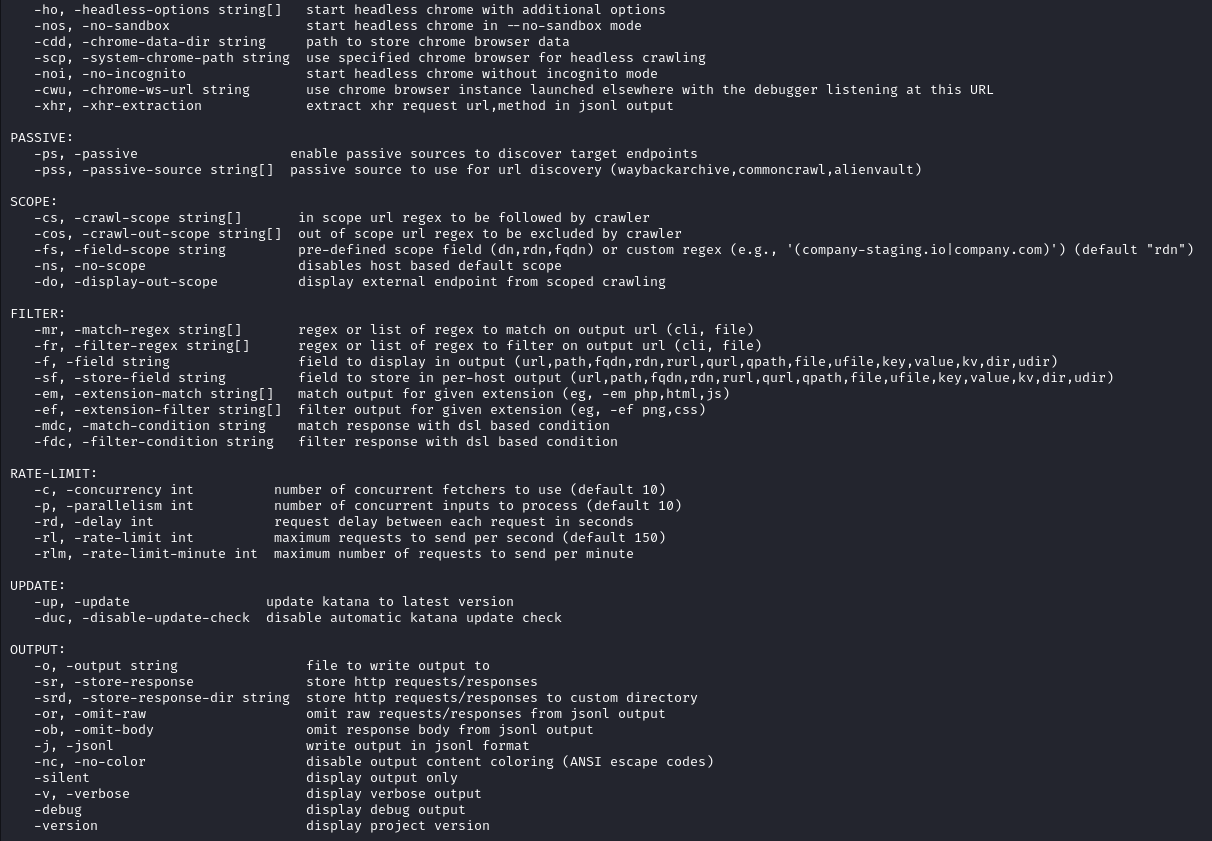
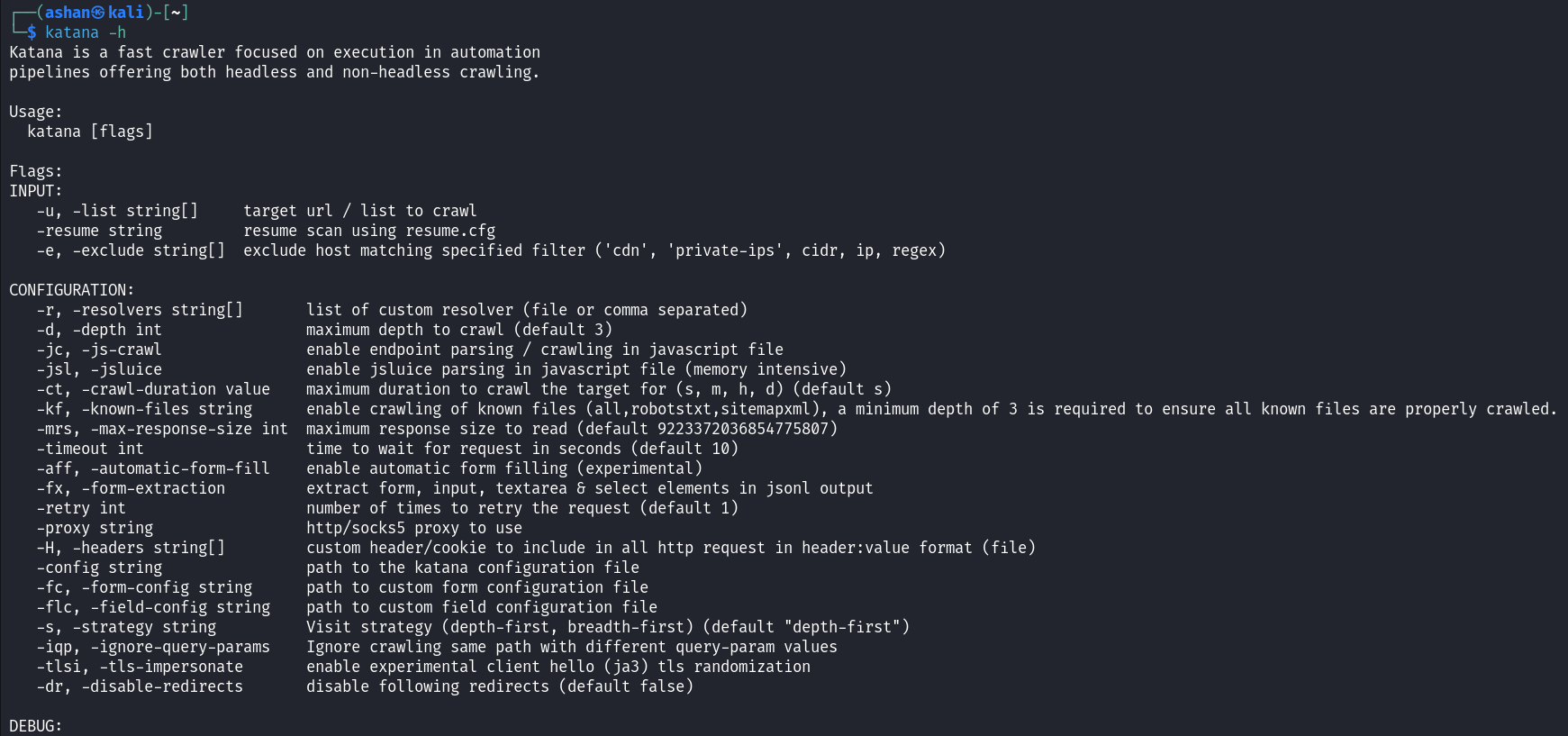
Wafw00f is a Python tool that helps detect and classify the web application firewall (WAF) in place. It conducts tests to identify the presence of a WAF and determine its type.



## Xss\_vibes

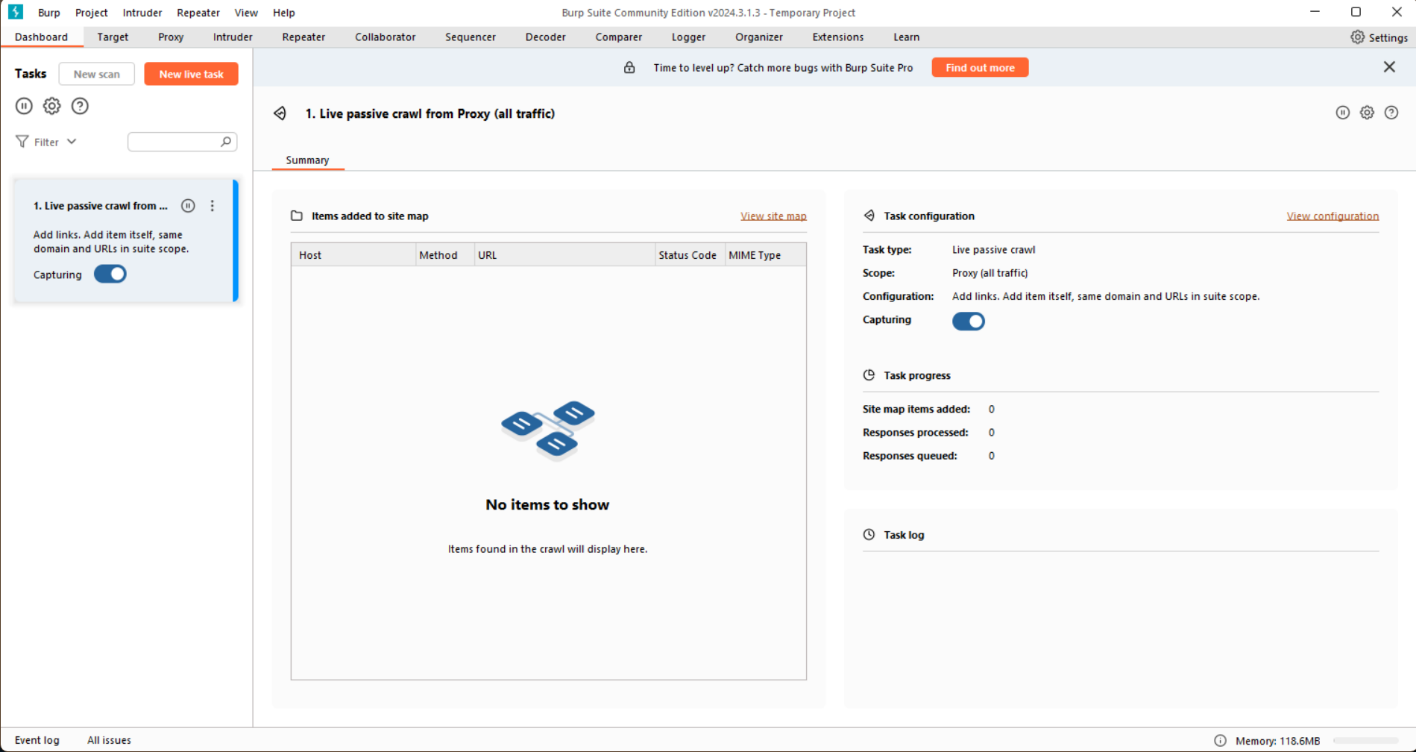
A modern tool that written in python which helps us to automate our XSS findings.

## Katana

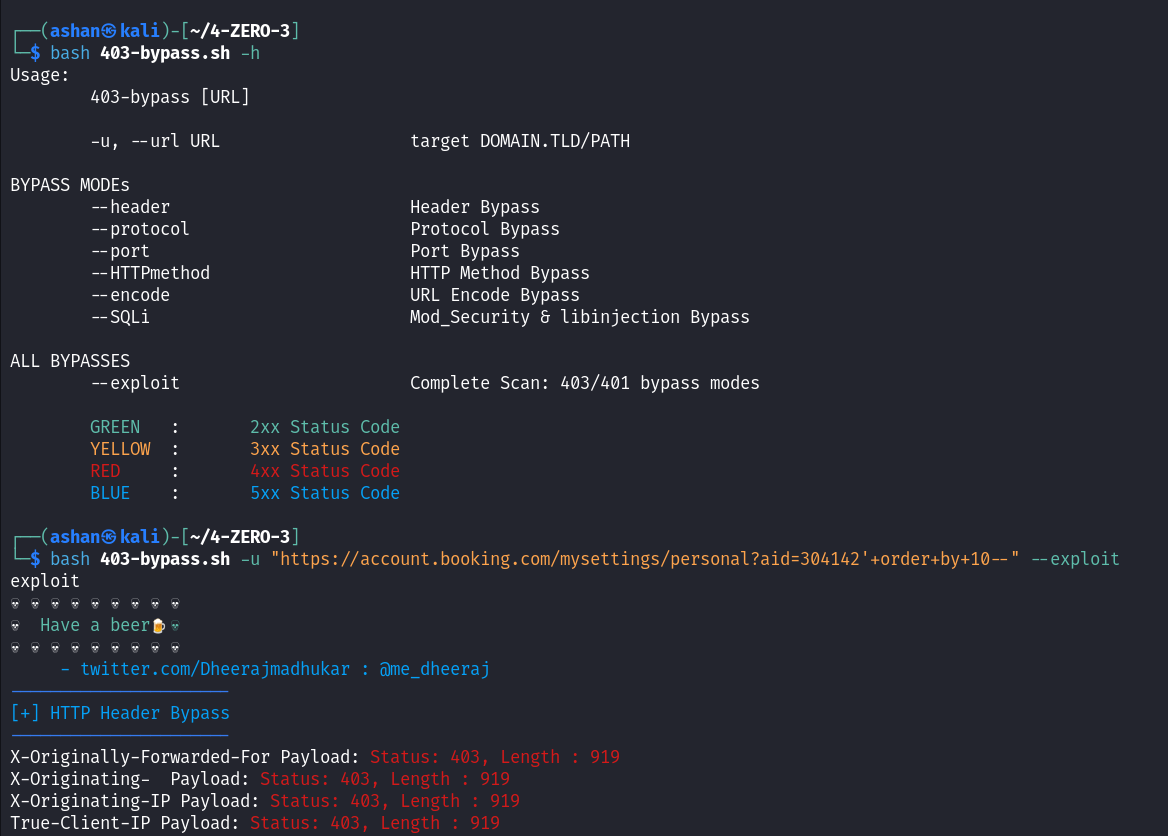
The software is crafted to be quick, effective and user friendly making it a favored option, for security experts and developers. It utilizes Go, a coding language recognized for its speed and ability to handle tasks simultaneously enhancing Katanas effectiveness, in scanning web applications.

## Burp Suite

Burp Suite is a set of tools used for penetration testing of web applications. It is developed by the company named Portswigger.



## 4-ZERO-3 Bypass

This tool is used for 403 forbidden status bypassing purposes.

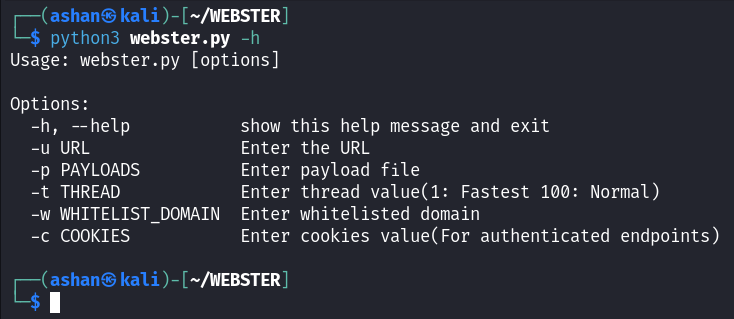
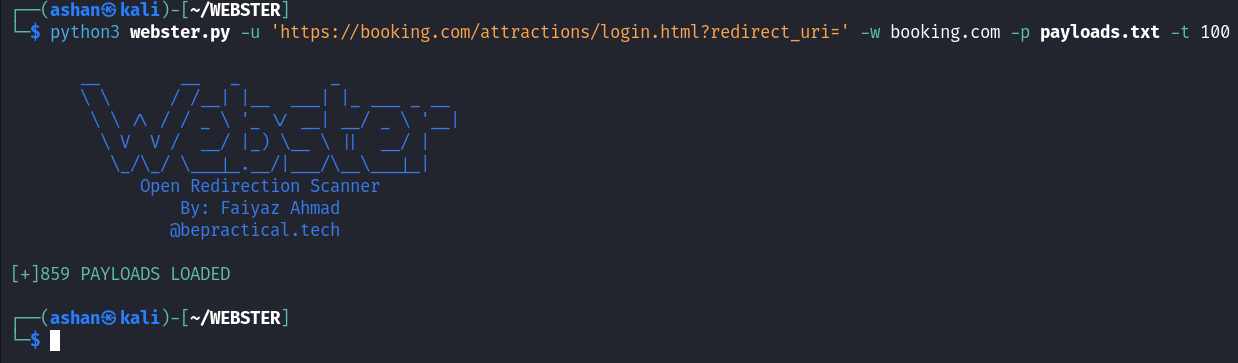
## Webster

Find Open Redirection on authenticated points: Most of the time, the redirection occurs after authentication. This is where most of the automation tool fails. With this tool you can test for open redirection on those authenticated pages by just specifying the cookies. The rest of the job will be done by this tool.

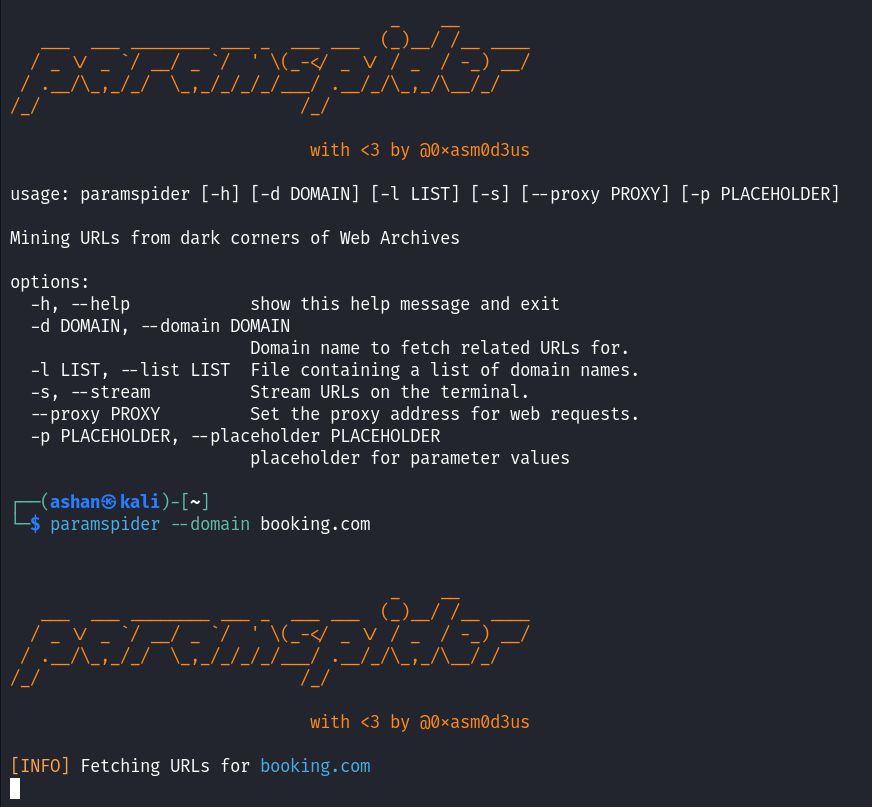
Dynamic Payload Generator: Suppose there is a target that allows only a specific domain to redirect. With this tool, you can just specify the whitelist domain and it will generate special payloads for the whitelist domain in it to bypass open redirection protection

Fast: This tool has a tremendous speed! You can test over 800+ payloads within 3-4 seconds.

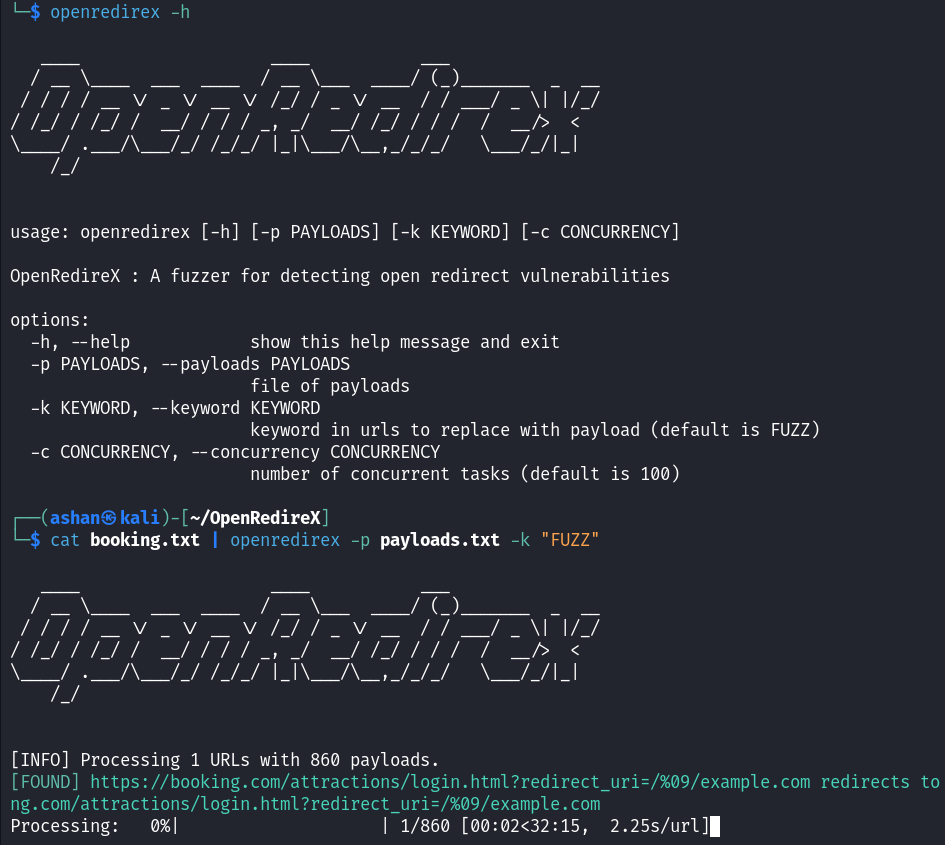
Easy to use: Everyone can use this tool with ease! No programming knowledge is required for it

Customizable: You can use your own payload to increase the probability of getting bugs!

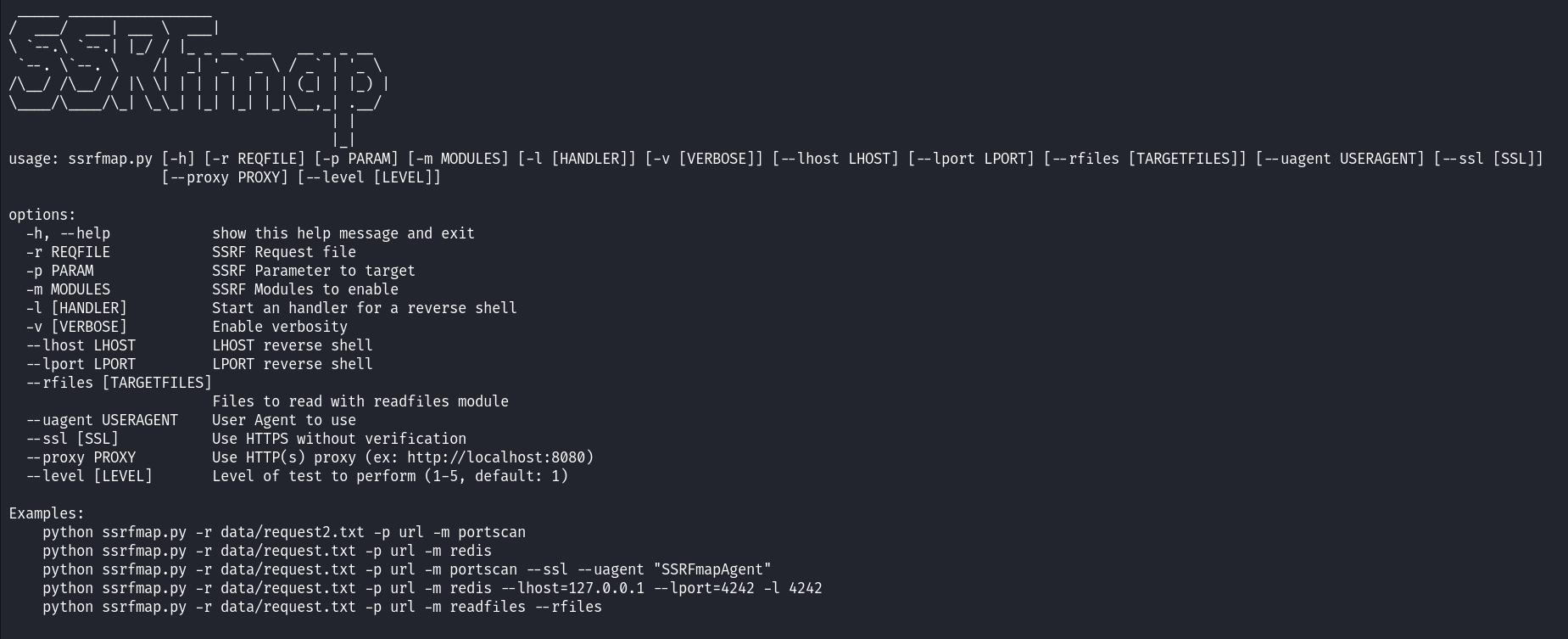
## ParamSpider

paramspider allows you to fetch URLs related to any domain or a list of domains from Wayback achives. It filters out "boring" URLs, allowing you to focus on the ones that matter the most.

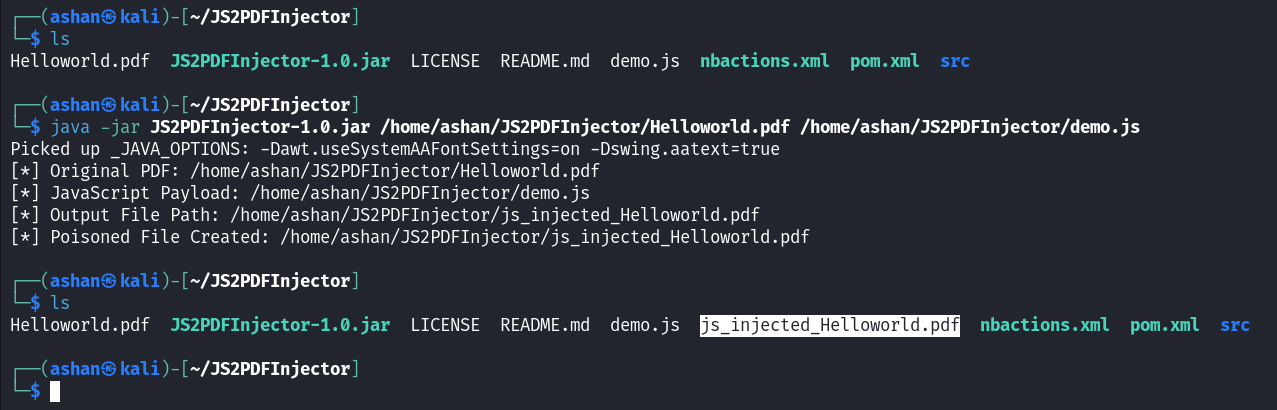
## OpenRedirex

A fuzzer for detecting open redirect vulnerabilities.

## SSRFmap

SSRF are often used to leverage actions on other services, this framework aims to find and exploit these services easily. SSRFmap takes a Burp request file as input and a parameter to fuzz.

## JS2PDFInjector

This tool is used for injecting JavaScript code into pdf file.

# Cross-Site Scripting

## Date: 04/25/2024

## Summary of the day's activities:

Today I spent my time focusing on conducting a security check, for XSS vulnerabilities on the account.booking.com webpage, which is featured on the hackerone platform.

## Vulnerabilities discovered or explored:

During the assessment my main goal was to search for XSS vulnerabilities within Booking.com’s website. I used methods like scanning with XSSVIBES manual testing with BurpSuite Intruder and URL enumeration with Katana. Despite conducting an examination no exploitable XSS weaknesses were found.

## Challenges faced and how they were overcome:

One significant challenge faced during the assessment was the inability to detect any XSS vulnerabilities despite employing tools and techniques and the time which takes to scan for vulnerabilities. However, I persisted through this challenge by staying focused and ensuring that each tool was utilized effectively.

## New tools, techniques, or concepts learned:

This assessment provided me with experience using tools like XSSVIBES for scanning and BurpSuite Intruder for manual testing. Although no exploitable vulnerabilities were identified it shed light on the intricacies of XSS testing. Emphasized the significance of security practices.

## Reflections and takeaways:

My activities today highlighted how security testing can be unpredictable, at times and underscored the importance of being thorough and resilient when tackling challenges in this field.

Although not finding any vulnerabilities might appear as a lack of advancement it's crucial to acknowledge the significance of verifying their absence. This incident highlighted the necessity, for education and adjustment in the field of cybersecurity along with the role of strong security measures, in minimizing potential threats.

# Web Cache Poisoning

## Date: 04/26/2024

## Summary of the day’s activities:

Today I spent my time focusing on conducting a security check, for web cache poisoning vulnerabilities on the voi.com website, which is featured on the intigriti platform.

## Vulnerabilities discovered:

During the assessment my main focus was on investigating web cache poisoning of the voi.com website. Despite testing I didn't find any vulnerabilities in <https://www.voi.com/blog/city-symposium> webpage.

## Challenges faced and how they were overcome:

One challenge faced was in attempting to find reflections of injected payload in the responses. Despite multiple attempts and manipulations of headers, no reflections of the payload were found.

## New tools, techniques and concepts learned:

I gained a deeper understanding of web cache poisoning and the various techniques used to detect and exploit it.

## Reflections and Takeaways:

Reflecting on todays activities it's evident that identifying Web Cache Poisoning vulnerabilities, in web applications can be quite complex.

Although the evaluation didn't uncover any vulnerabilities to exploit it did offer insights, for learning. Emphasized the significance of testing and ongoing enhancements in cybersecurity procedures. Looking ahead I am dedicated to enhancing my expertise in vulnerability assessment methods to enhance protection, against risks.

# Response Manipulation & Abusing Password Reset Functionality

## Date: 04/27/2024

## Summary of the Day’s Activities:

Today I spent my time focusing on conducting a response manipulation and abusing the password reset functionality in the account.booking.com webpage, which is featured on the hackerone platform. The main goal was to check for broken access control vulnerabilities within the web application.

## Vulnerabilities Discovered:

1. Response Manipulation – Analyzed the user authentication flow by intercepting and capturing the both incorrect and correct login attempts and checked for JSON coding with user authentication details.
2. Abusing the Password Reset Functionality – Analyzed how password reset function works. Captured and modified the password reset request on the JSON body by adding the attacker’s email address to get the victim’s password reset link.

## Challenges Faced and How They Were Overcome:

It was difficult to understand the encrypted values within the authentication and password reset process. I was confused by the complexity of the encrypted values to manipulate responses. To overcome this challenge, I watched YouTube videos related to this vulnerability and got some ideas from Medium platform also.

## New Tools, Techniques, or Concepts Learned:

This assessment provided me with experience using tools like BurpSuite for manual testing. Although no exploitable vulnerabilities were identified it shed light on the intricacies of Broken Access Control testing. Emphasized the significance of security practices.

## Reflections and Takeaways:

My activities today highlighted importance of understanding the work flow of authentication and password reset functionality at times. The challenges I encountered are the lack of understanding of the encrypted data.

Although not finding any vulnerabilities might appear as a lack of advancement it's crucial to acknowledge the significance of verifying their absence. This incident highlighted the necessity, for education and adjustment in the field of cybersecurity along with the role of strong security measures, in minimizing potential threats.

# SQL Injection

## Date: 04/27/2024

## Summary of the Day’s Activities:

Today I spent my day examining SQL Injection vulnerabilities, in the account.booking.com webpage. I used a mix of automated testing methods to test the target parameter.

## Vulnerabilities Discovered:

During the assessment my main focus was on investigating the "aid" parameter in the URL path of the account.booking.com webpage. Despite testing I didn't find any SQL Injection vulnerabilities in this parameter.

## Challenges faced and how they were overcome:

One major challenge I faced was the time consuming nature of running sqlmap scans. The tool took longer than expected to complete its scan causing delays in the assessment process and some connection issues also due to the bad weather.

## New Tools, Techniques, or Concepts Learned:

Through this assessment I learned lessons about conducting SQL Injection vulnerability tests. Realized the significance of using a mix of manual and automated testing approaches. Additionally I picked up tips on optimizing sqlmap commands and using testing methods for efficiency.

## Reflections and Takeaways:

Reflecting on todays activities it's evident that identifying SQL Injection vulnerabilities, in web applications can be quite complex.

Although the evaluation didn't uncover any vulnerabilities to exploit it did offer insights, for learning. Emphasized the significance of testing and ongoing enhancements in cybersecurity procedures. Looking ahead I am dedicated to enhancing my expertise in vulnerability assessment methods to enhance protection, against risks.

# Open Redirection

## Date: 04/28/2024

## Summary of the Day's Activities:

Today I focused on assessing Open Redirection vulnerabilities, in the Booking.com website. I used tools and techniques to test URL parameters and search for any redirection vulnerabilities.

## Vulnerabilities Discovered:

After testing I did not find any Open Redirection vulnerabilities in the Booking.com website.

## Challenges Faced and How They Were Overcome:

During the assessment one major challenge was the time it took to complete the testing process. I had to collect URLs with parameters test them for vulnerabilities and check for firewall restrictions. To address this issue I increased the Thread count to make the scanning more faster.

## New Tools, Techniques, or Concepts Learned:

This assessment helped me understand the complexities of finding and addressing Open Redirection vulnerabilities in web applications. I also got familiar with tools like Paramspider, OpenRedireX and Webster that were crucial for conducting vulnerability assessments.

## Reflections and Takeaways:

From todays activities I learned how essential it is to have an approach to vulnerability assessments in uncovering security risks, within web applications. Although no significant vulnerabilities were found during the evaluation it was a learning experience that highlighted the importance of improving cybersecurity measures. I plan to focus on enhancing my expertise in vulnerability assessment methods to enhance protection, against risks, in the future.

# Cross Site Request Forgery (CSRF)

## Date: 04/29/2024

## Summary of the Day’s Activities:

Today I focused on checking for Cross Site Request Forgery (CSRF) vulnerabilities, in the update personal details feature of the account.booking.com webpage. My assessment included analyzing how requests were made capturing them with Burp Suite and manually testing for CSRF vulnerabilities.

## Vulnerabilities Discovered:

I didn't find any CSRF vulnerabilities in the account.booking.com webpage despite conducting tests on personal details fields. No unauthorized actions were detected during our testing.

## Challenges Faced and how they were overcome:

I faced challenges during the assessment. Initially I encountered coding errors in the CSRF proof of concept (PoC) which required debugging and fixing. Furthermore power outages caused disruptions in my testing process resulting in some delays. I managed to overcome these challenges by examining and correcting the PoC code and continuing my tests after resolving the power issues.

## New Tools, Techniques, or Concepts Learned:

This assessment provided me with insights into how to test for CSRF vulnerabilities, in web applications. Furthermore I improved my understanding of creating CSRF Proof of Concepts (PoCs) and conducting tests, on personal information fields.

## Reflections and Takeaways:

The events of today underscored the significance of testing and meticulous attention to detail in pinpointing security weaknesses in web applications. Despite not discovering any CSRF vulnerabilities during the assessment it offered learning experiences and emphasized the importance of continually enhancing cybersecurity practices. Going forward I am dedicated to honing my expertise in vulnerability assessment methods to enhance protection, against risks.

# Race Conditions

## Date: 04/30/2024

## Summary of the Day’s Activities:

Today I focused on assessing vulnerabilities related to Race Conditions in the feature that allows adding travelers, on the account.booking.com webpage. I used the Burp Suite Turbo Intruder tool to send requests and examine how the server responded.

## Vulnerabilities Discovered:

After testing I did not find any Race Conditions vulnerabilities in this part of the account.booking.com webpage.

## Challenges Faced and how they were overcome:

During the assessment I faced challenges like encountering script errors while making changes to the Turbo Intruder script. To overcome these issues I searched for solutions on platforms like Stack Overflow. Made adjustments to the script.

## New Tools, Techniques, or Concepts Learned:

From this experience I learned more about testing for Race Conditions vulnerabilities in web applications. Using tools like Turbo Intruder for sending requests at once and analyzing server responses.

## Reflections and Takeaways:

This activity emphasized the importance of testing. Being vigilant in detecting security flaws, in web applications. Although no vulnerabilities were found this time it showed me how crucial it is to enhance cybersecurity practices through learning and improvement.

In the future I plan to improve my abilities and understanding, in vulnerability assessment methods to enhance protection, against risks.

# Server-Side Request Forgery (SSRF)

## Date: 05/01/2024

## Summary of the Day's Activities:

Today my main focus was to assess and pinpoint any vulnerabilities related to Server Side Request Forgery on the compass.com website. This assessment included an analysis of parameters to uncover any SSRF vulnerabilities utilizing both manual and automated testing methods.

## Vulnerabilities Discovered:

After examination no SSRF vulnerabilities were found within the compass.com site. I tested parameters that're commonly susceptible, to SSRF attacks, such as next, redirect, page, path, return, url and view parameters.

## Challenges Faced and How They Were Overcome:

Throughout the assessment process I faced challenges. One significant hurdle involved configuring and using testing tools like SSRFMap for effective automation.

## New Tools, Techniques, or Concepts Learned:

Through this assessment I gained knowledge about the tools functionality in automating testing, for SSRF vulnerabilities by offering a straightforward method to test various parameters efficiently.

This tool was really helpful, in making the testing process smoother and uncovering weaknesses.

## Reflections and Takeaways:

The tasks today emphasized how crucial it is to conduct testing and employ both automated methods to spot potential security issues in web applications. Even though the evaluation didn't uncover any vulnerabilities on the compass.com site, the experience. Using SSRFMap provided valuable insights into the testing procedure. Going ahead I am dedicated to enhancing my skills and understanding vulnerability assessment techniques to enhance protection, against risks.

# Directory Traversal

## Date: 05/01/2024

## Summary of the Day’s Activities:

Today I spent my time focusing on conducting a security check, for Directory traversal on the veriff.com web application, which is featured on the hackerone platform.

## Vulnerabilities Discovered:

Despite testing no instances of Directory Traversal vulnerabilities were found on the veriff.com website. Various tests were carried out on parameters, such, as the "image" parameter. All efforts were thwarted by the Web Application Firewall (WAF) resulting in a 403 Forbidden response.

## Challenges Faced and How They Were Overcome:

One challenge faced during the assessment was pinpointing a parameter for Directory Traversal. This demanded analysis and experimentation with parameters until the "image" parameter was identified as a potential target. Moreover overcoming the WAFs blocking proved challenging as it hindered exploitation of the vulnerability. These obstacles were tackled through perseverance, experimentation and employing testing methods.

## New Tools, Techniques or Concepts Learned:

Throughout the assessment process I familiarized myself with an array of tools and techniques for assessing Directory Traversal vulnerabilities. This included using Paramspider to collect URLs Intruder for scanning tasks and directory traversal payload sheets for payload testing. Additionally I gained insights into the significance of input validation and security measures, in mitigating Directory Traversal vulnerabilities.

## Reflections and Takeaways:

Todays activities offered perspectives to identifying and examining Directory Traversal vulnerabilities within web applications.

No vulnerabilities were found on veriff.com. The assessment emphasized the need, for testing, perseverance and using a mix of manual and automated methods during vulnerability assessments. I plan to focus on improving my expertise in web application security to recognize and address risks effectively in the future.

# File Upload leading to JavaScript Code Execution

## Date: 05/01/2024

## Summary of the Day’s Activities:

Today I spent my time focusing on conducting a security check, for File Upload vulnerabilities on the [www.peoplecert.org](http://www.peoplecert.org) web application, which is featured on the Intigriti platform.

## Vulnerabilities Discovered:

A file upload vulnerability allowing attackers to execute JavaScript code was identified within the profile picture upload functionality of the [www.peoplecert.org](http://www.peoplecert.org/) website. Which allows an attacker to upload and execute pdf file injected with malicious JavaScript code.

## Challenges Faced and how they were overcome:

No challenges were faced during the exploitation.

## New tools, techniques, or concepts learned:

This assessment provided me with insights into how to test for File Upload vulnerabilities, in web applications. Furthermore I improved my understanding of injecting JavaScript codes into a PDF file using JS2PDFInjector tool.

## Reflections and takeaways:

My activities today highlighted how importance of implementing security measures, input validation, content type checking and file extension inspection to protect against file upload vulnerabilities. By implementing the mentioned mitigation techniques, the security of the web server can be enhanced.