**BUB BOUNTY**

A blue and orange logo

Description automatically generated

**IT NUMBER: IT22345332**

**NAME: G.P DINUJAYA THAMARA**

**WEEKEND BATCH**

**MALABE CAMPUS**

**Bug Bounty Platform – Hacker One**

**Bug Bounty Program - Booking.com**

**Scope**

**In Scope Assets**

For in Scope Assets please refer to the Scope tab

**Out-Of-Scope Applications** Any application whether owned by Booking.com or third-party vendor **not included as an in-scope asset** will be mentioned on the scope tab as out of scope.

For Out Of Scope Assets please refer to the Scope tab

**In-scope Vulnerabilities**

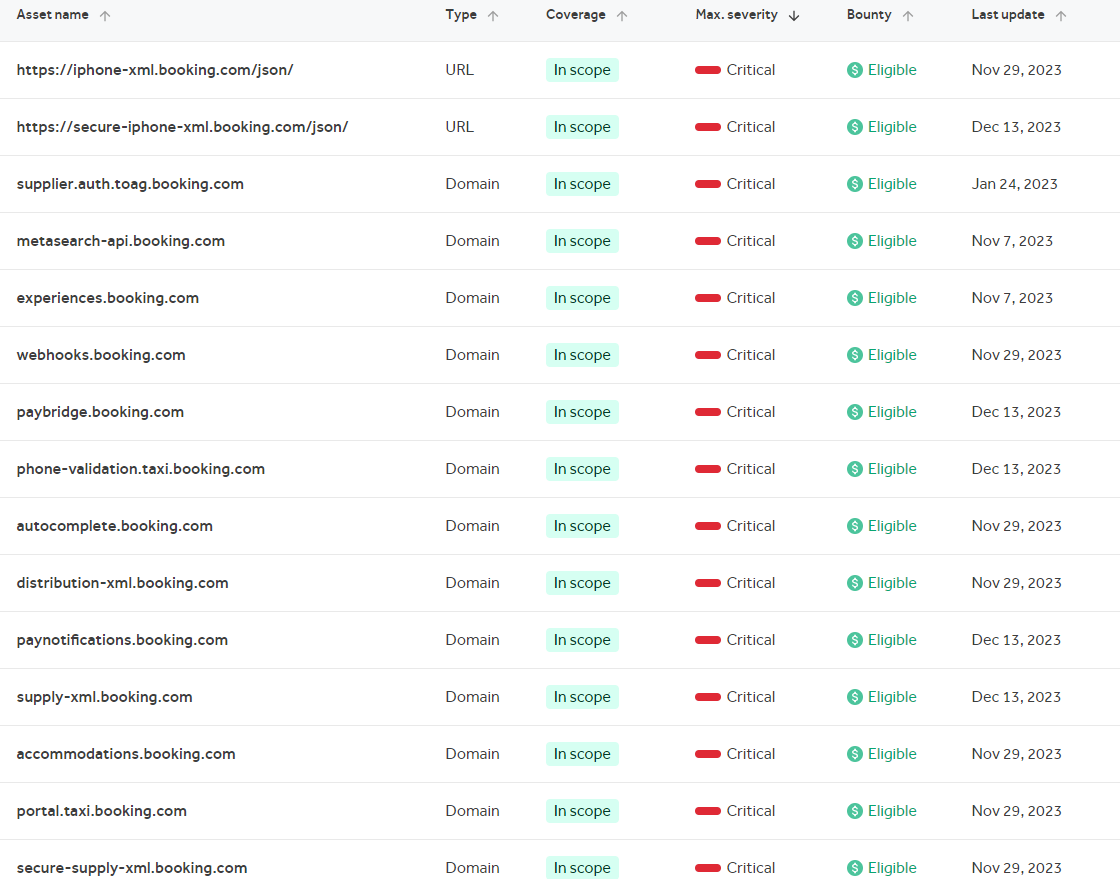
**Accepted, in-scope vulnerabilities include, but are not limited to:**

* Disclosure of sensitive or personally identifiable information
* Cross-Site Scripting (XSS) - Please note, for XSS if the same issue is reported for the different subdomains but with the same root cause, it will be considered duplicate
* Cross-Site Request Forgery (CSRF) for sensitive functions in a privileged context
* Remote code execution (RCE)
* Authentication or authorization flaws, including insecure direct object references and authentication bypass
* Injection vulnerabilities, including SQL and XML injection
* Directory traversal
* Significant security misconfiguration with a verifiable vulnerability
* Account takeover by exploiting a vulnerability
* SSRF
* XXE
* Subdomain takeover in \*.booking.com domains

**Out-Of-Scope Vulnerabilities** Depending on their impact, not all reported issues may qualify for a monetary reward. However, all reports are reviewed on a case-by-case basis and any report that results in a change being made will at a minimum receive recognition. Please note that our **program terms and rules of engagement** still apply.

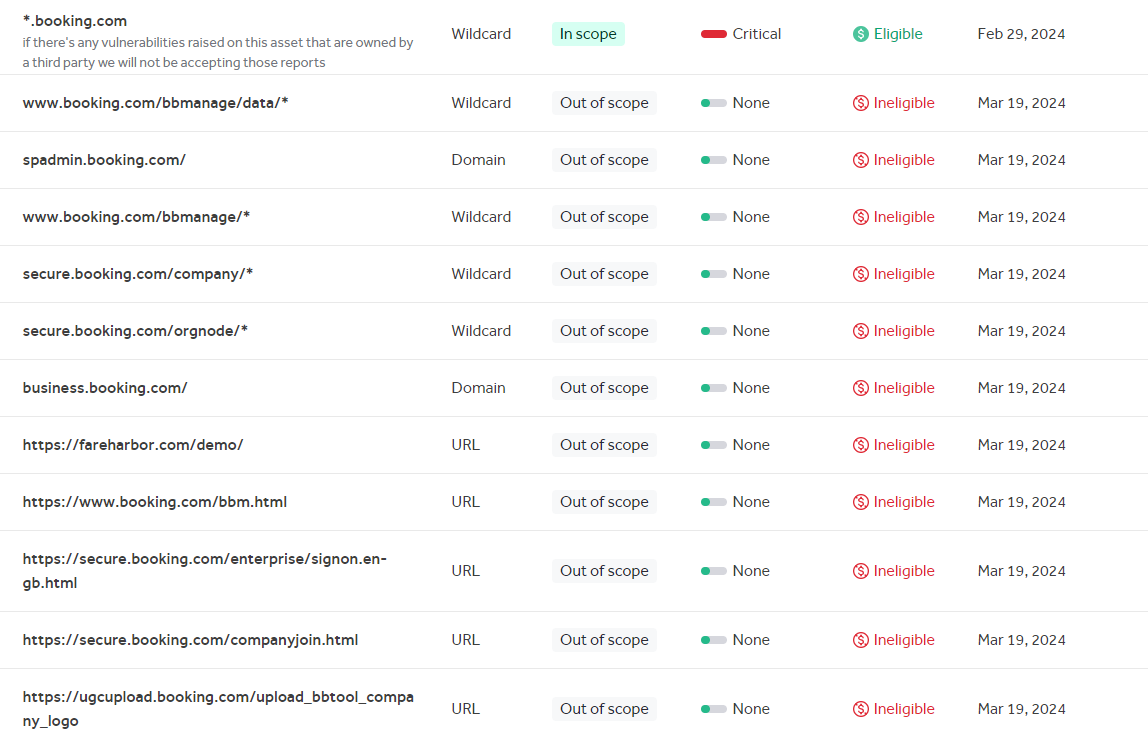
**The following issues are outside the scope of our vulnerability rewards program:**

* Any vulnerability which requires access to a compromised email account or Booking.com account for successful exploitation
* Vulnerabilities on Third Party Products
* Attacks requiring physical access to a user's device or network.
* Forms missing CSRF tokens (we require evidence of actual CSRF vulnerability)
* Login/Logout CSRF
* Missing security headers which do not lead directly to a vulnerability
* Use of a known-vulnerable library (without evidence of exploitability)
* Reports from automated tools or scans
* Social engineering of Booking staff or contractors
* Denial of Service attacks and/or reports on rate limiting issues
* Not enforcing certificate pinning
* Any issues that require a rooted or jailbroken device or a compromised device
* Clickjacking
* Improper session invalidation
* User enumeration
* Host header injections without a specific, demonstrable impact
* Self-XSS, which includes any payload entered by the victim
* Any vulnerabilities requiring significant and unlikely interaction by the victim, such as disabling browser controls
* Content spoofing without embedded HTML or JavaScript
* Hypothetical issues that do not have any practical impact
* Infrastructure vulnerabilities, including:
* Issues related to SSL certificates
* DNS configuration issues
* Server configuration issues (e.g. open ports, TLS versions, etc.)



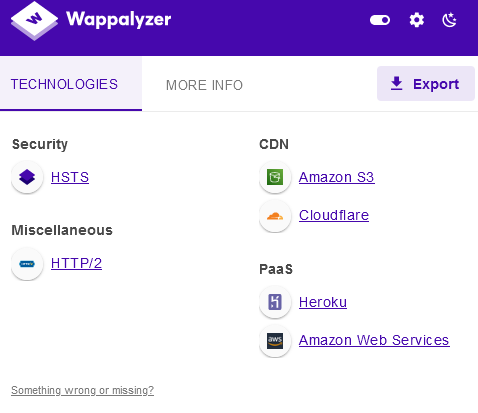
A screenshot of a computer

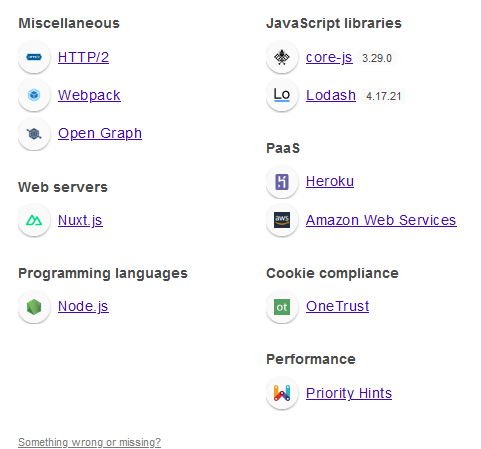
Description automatically generated



<https://careers.booking.com/>

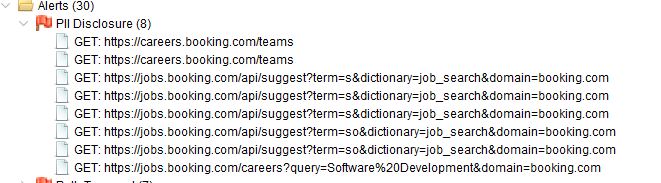
According the wappalyzer the following technologies are used by this subdomain.





After being tested through the OWSAP ZAP following vulnerabilities were found in the above subdomain.



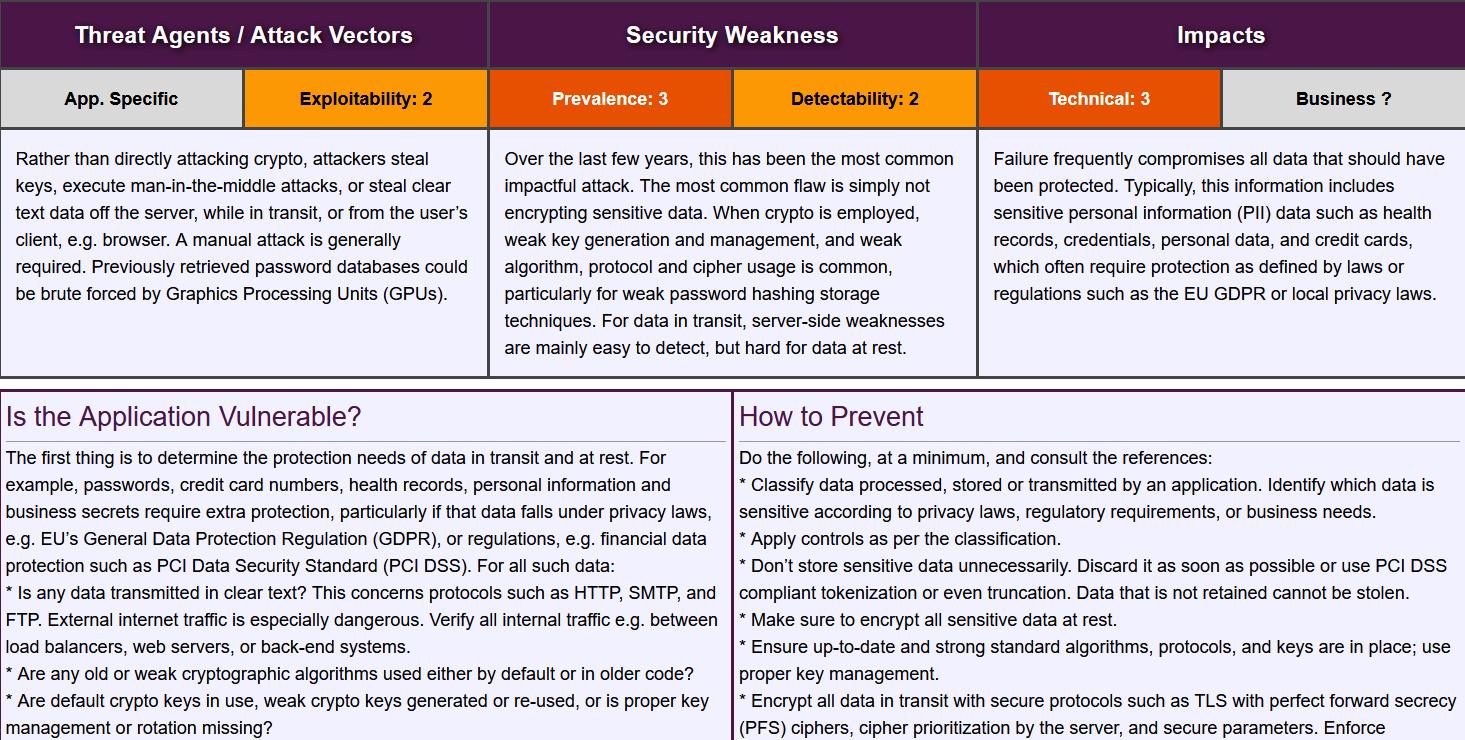


How to test for information disclosure vulnerabilities

Generally speaking, it is important not to develop "tunnel vision" during testing. In other words, you should avoid focussing too narrowly on a particular vulnerability. Sensitive data can be leaked in all kinds of places, so it is important not to miss anything that could be useful later. You will often find sensitive data while testing for something else. A key skill is being able to recognize interesting information whenever and wherever you come across it.

<https://portswigger.net/web-security/information-disclosure/exploiting>

<https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure>



Personally Identifiable Information (PII) [is formally defined as](https://www.dol.gov/general/ppii) "any representation of information that permits the identity of an individual to whom the information applies to be reasonably inferred by either direct or indirect means." This includes information that directly identifies an individual (e.g., name, address, Social Security number, or other identifying numbers or codes, telephone numbers, email addresses, etc.) or "by which an agency intends to identify specific individuals in conjunction with other data elements, i.e., indirect identification." These data elements may include a combination of gender, race, date of birth, geographic indicators, and other descriptors.

**Many types of PII can be described in the following ways:**

1. **Identity**: name, date of birth, signature, gender, race, familial situation
2. **Contact information**: address, phone number, email address
3. **Professional information**: job, company, position, date of hire, HR evaluation, salary
4. **Administrative documents:** ID, passport number, driver's license, vehicle identification number (VIN), Social Security number
5. **Healthcare**: biometric data, medical records
6. **IT related:** Internet Protocol (IP) address, password(s), cookies, logs

PII breaches or exposures can happen in a wide variety of ways which can make it difficult to prepare for data security threats and protect sensitive information. Generally, threats to PII fall under at least one of the three following categories:

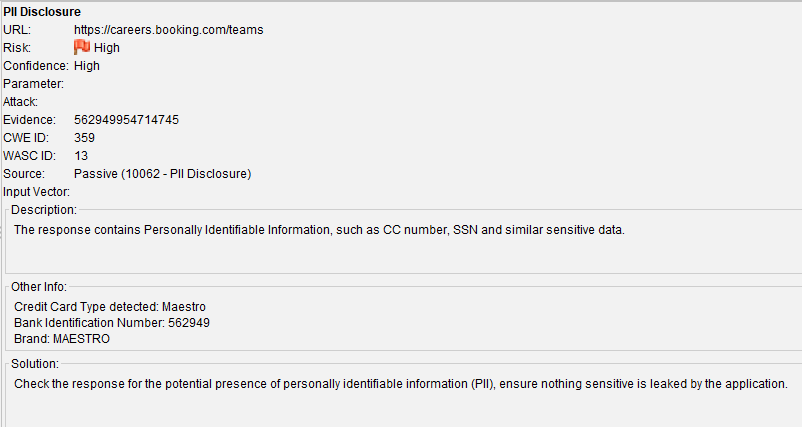
1. **Insider threat**. This results from someone from within your systems intentionally or unintentionally viewing or providing access to restricted data. Early in 2020, we talked about how work in a remote-first world could lead to insider threats becoming [an increased risk for organizations](https://nightfall.ai/resources/insider-threats-cybersecurity-threat-landscape-2020/).
2. **External threat**. This involves someone outside your organization deliberately tampering with or modifying systems in order to exfiltrate data.
3. **Security misconfigurations.** Thesegenerally occur within an application that captures user data or in the environment where the sensitive data is stored. Security misconfigurations can result in data exposures on their own or can be exploited by internal or external threat actors to exfiltrate data.

**4 ways to protect PII and other sensitive data**

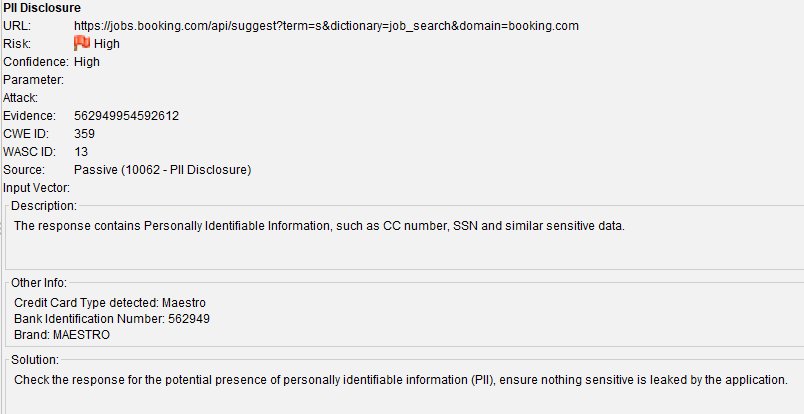
We want to wrap up this guide by discussing some important technologies you can use to protect PII and other types of sensitive data. Below is a non-exhaustive list of the technologies that can prevent unauthorized access to your systems and data.

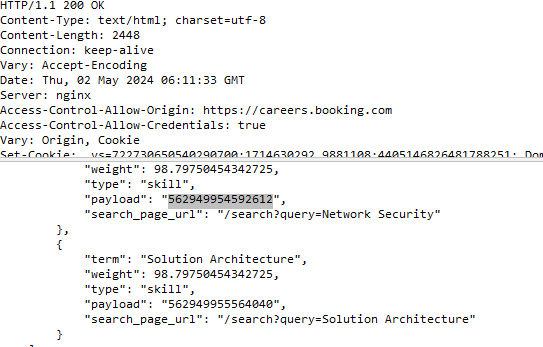
1. **Encryption**. Encryption is defined as the conversion of something to code or symbols so that its contents cannot be understood if intercepted. When sending a confidential email, you should use a program to encrypt its content. Additionally, encrypting data at rest is an important best practice for securing data at rest in databases and other sensitive systems.
2. **Identity and access management (IAM).** As the number of services and systems required by organizations and their employees increases, the value of identity and access management (IAM) has increased. IAM allows organizations to manage the accounts and permissions of employees at scale. With IAM, companies can enforce two factor authentication or multi-factor authentication and rapidly provision and deprovision accounts with ease. We talk in detail about IAM and why it matters in our [security playbook](https://nightfall.ai/download-2021-security-playbook-remote-first-orgs/).
3. **Endpoint management.** Endpoint managers provide a wide variety of functions, from managing device firmware and monitoring hardware activity to providing on-device security in the form of antivirus protection. We also talk in greater detail about endpoint management in our security playbook.
4. **Data Loss Prevention.** Data loss prevention, which we mentioned earlier in this guide, is a way to monitor the data flowing in and out of your environments. In our security playbook we make a case for leveraging cloud native data loss prevention to detect and classify business critical data in your SaaS applications and cloud infrastructure. Nightfall specifically uses machine learning detectors that are capable of object character recognition (OCR) and natural language processing (NLP) to classify strings, files and images that contain PII, PHI and a wide variety of sensitive data. From there we implement rules that let you monitor who has access to this data and control when to redact or remove it. Such a technology is invaluable for ensuring your data can live in the cloud safely.

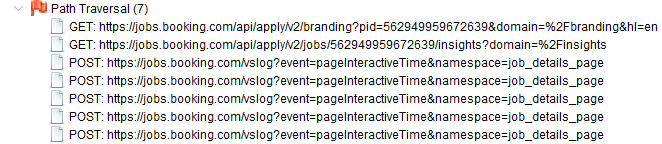
<https://github.com/nightfallai/pii-leak-prevention-guide>









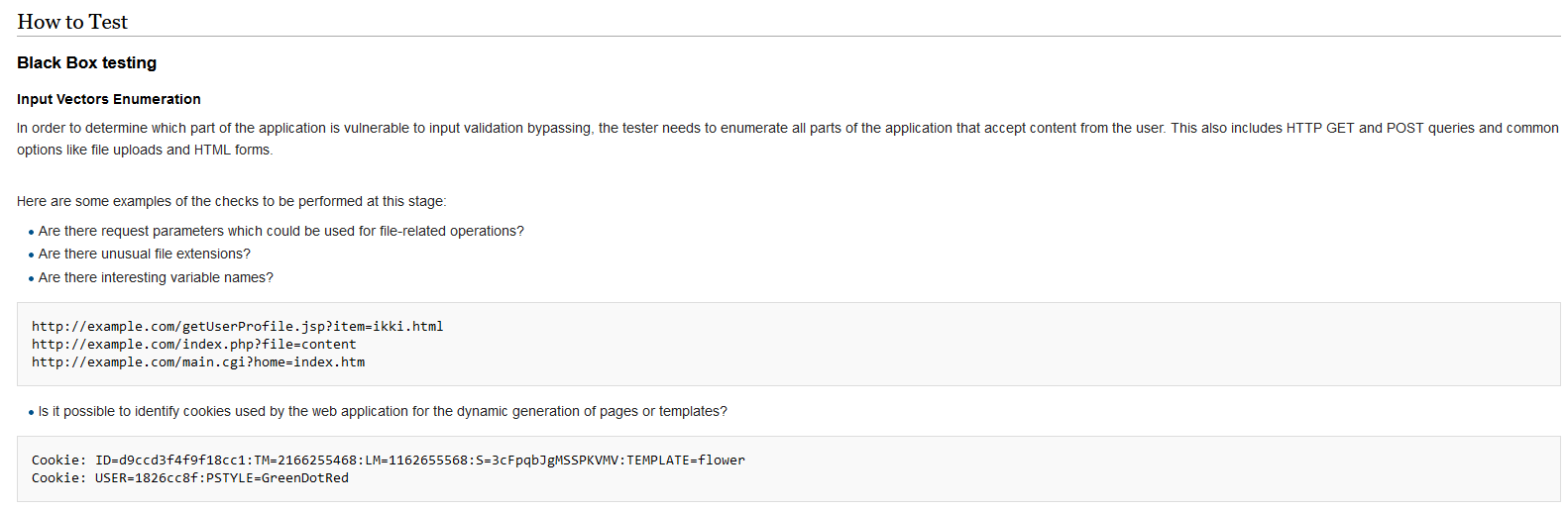


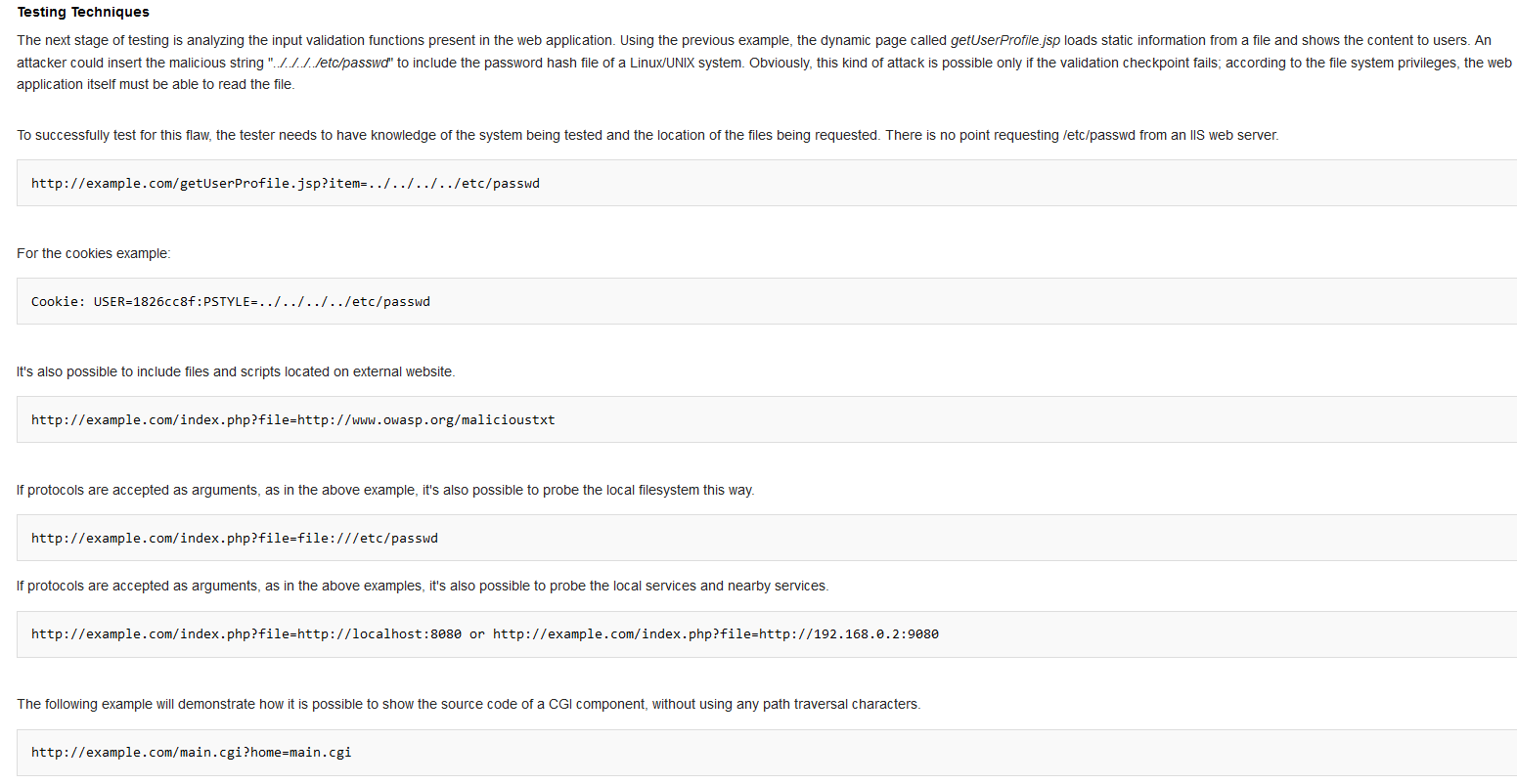
A path traversal attack (also known as directory traversal) aims to access files and directories that are stored outside the web root folder. By manipulating variables that reference files with “dot-dot-slash (../)” sequences and its variations or by using absolute file paths, it may be possible to access arbitrary files and directories stored on file system including application source code or configuration and critical system files. It should be noted that access to files is limited by system operational access control (such as in the case of locked or in-use files on the Microsoft Windows operating system).

This attack is also known as “dot-dot-slash”, “directory traversal”, “directory climbing” and “backtracking.

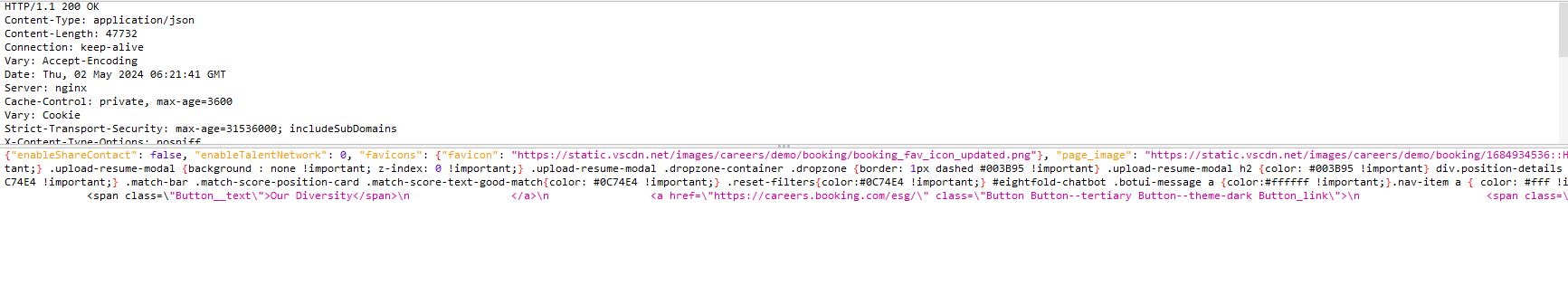


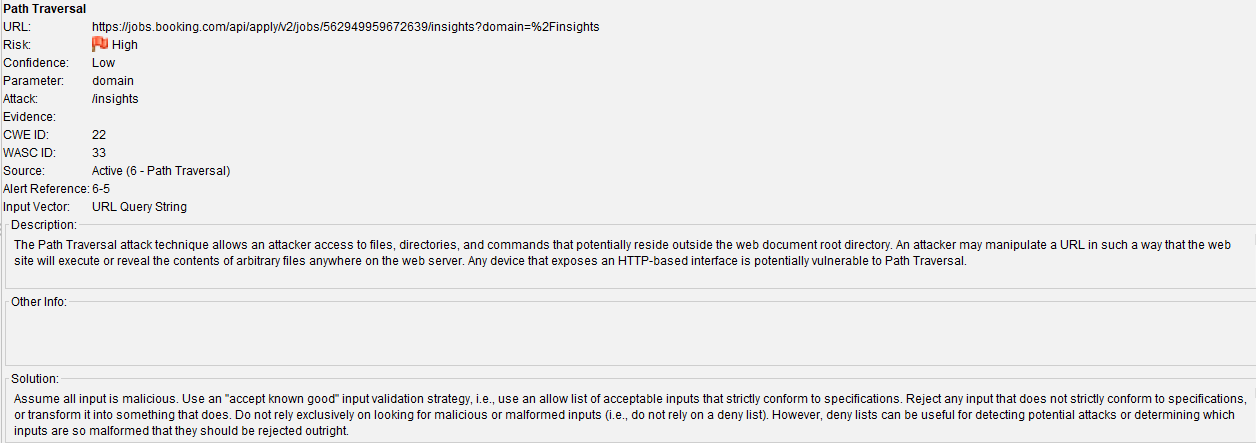
<https://wiki.owasp.org/index.php/Path_Traversal>



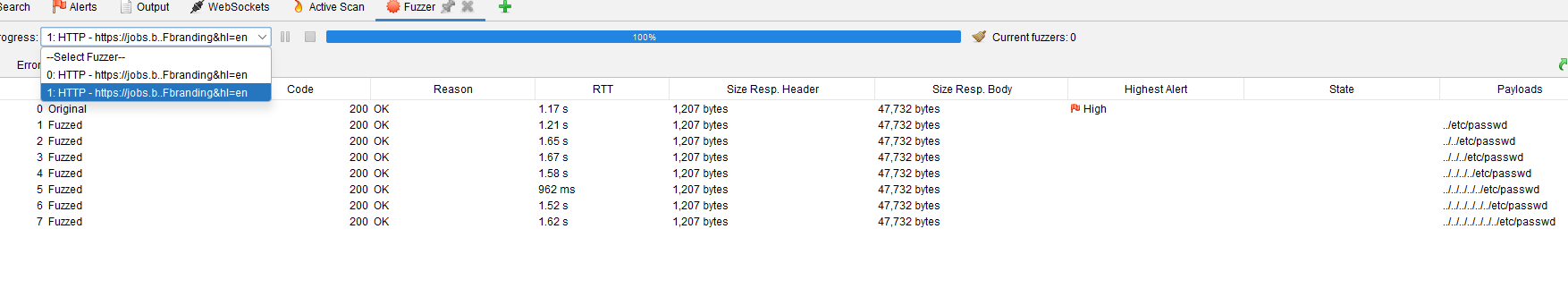


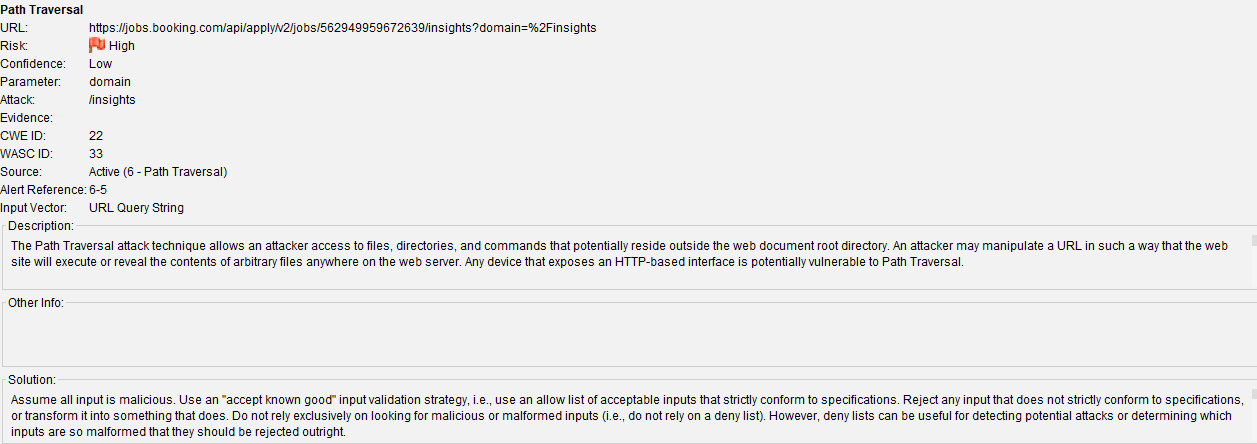






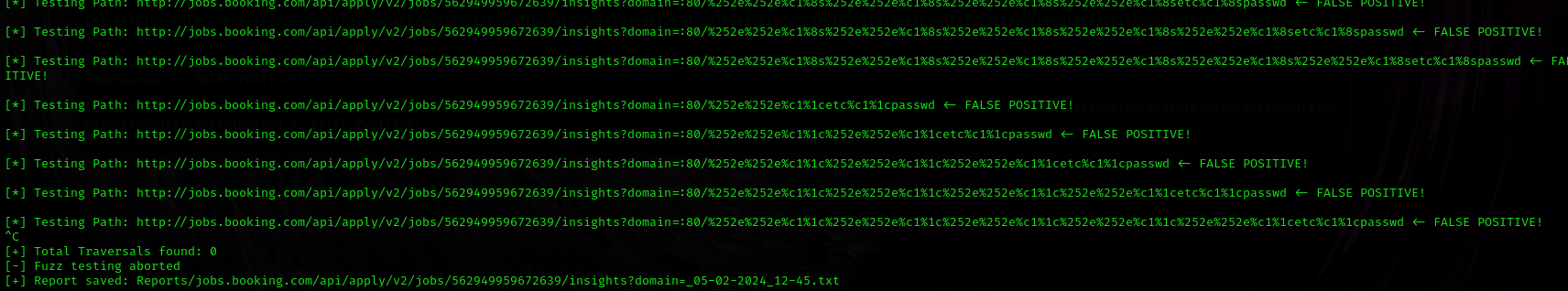
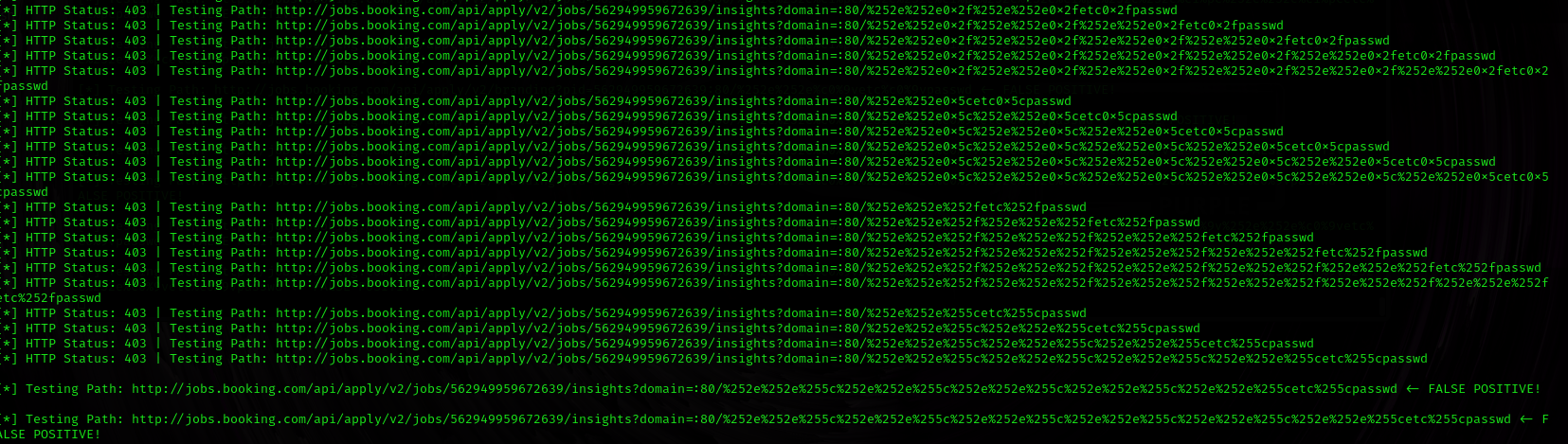
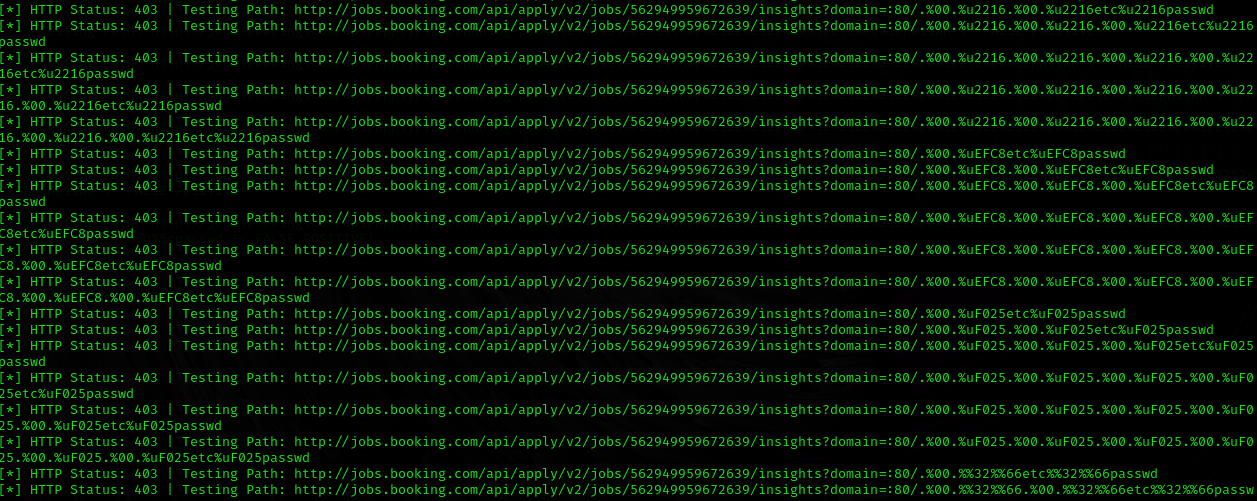
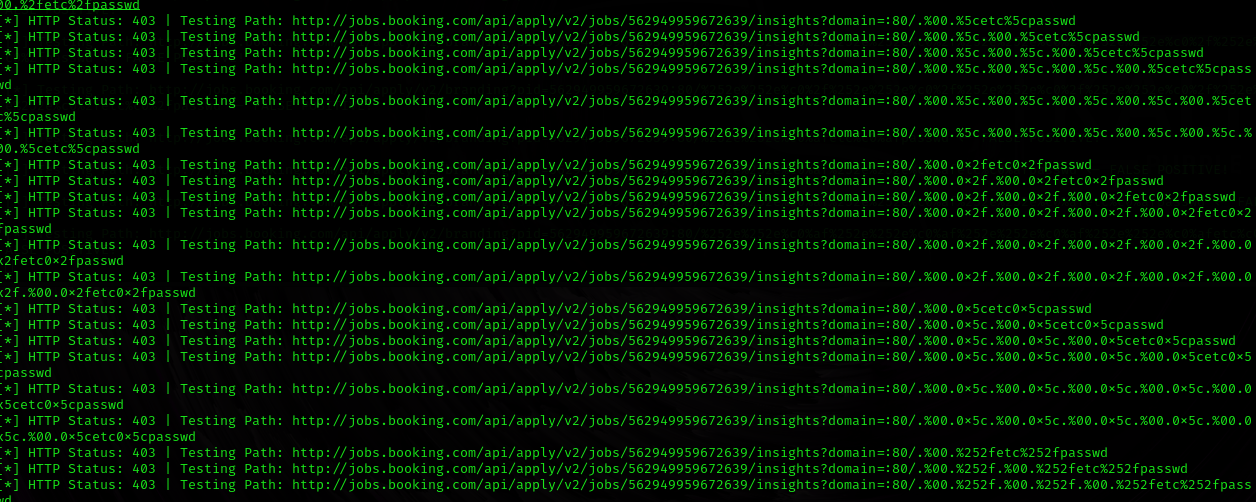
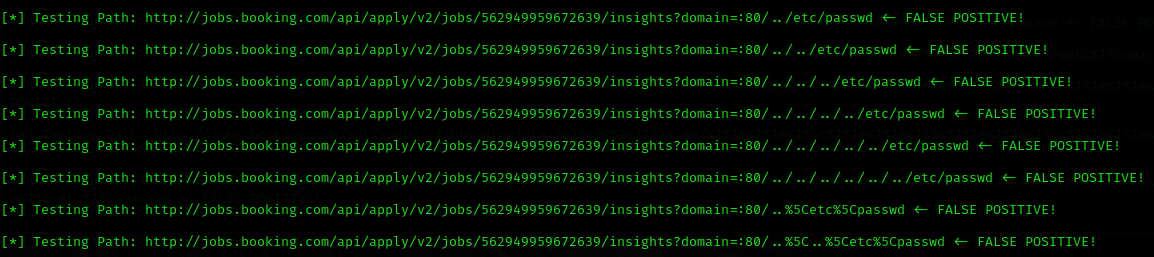
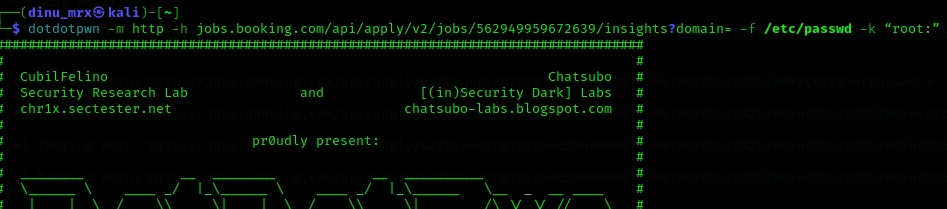
Below is a screen shot of Fuzzing using OWSAP ZAP .



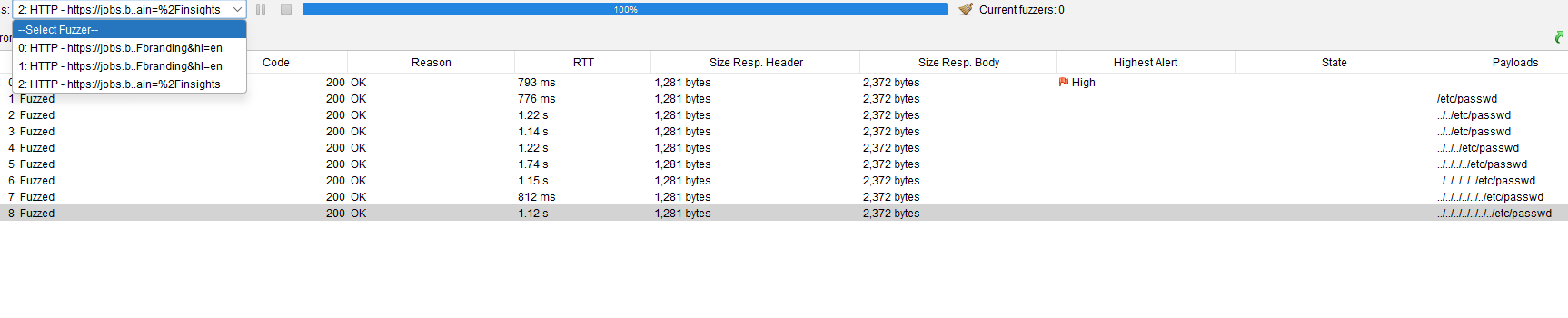


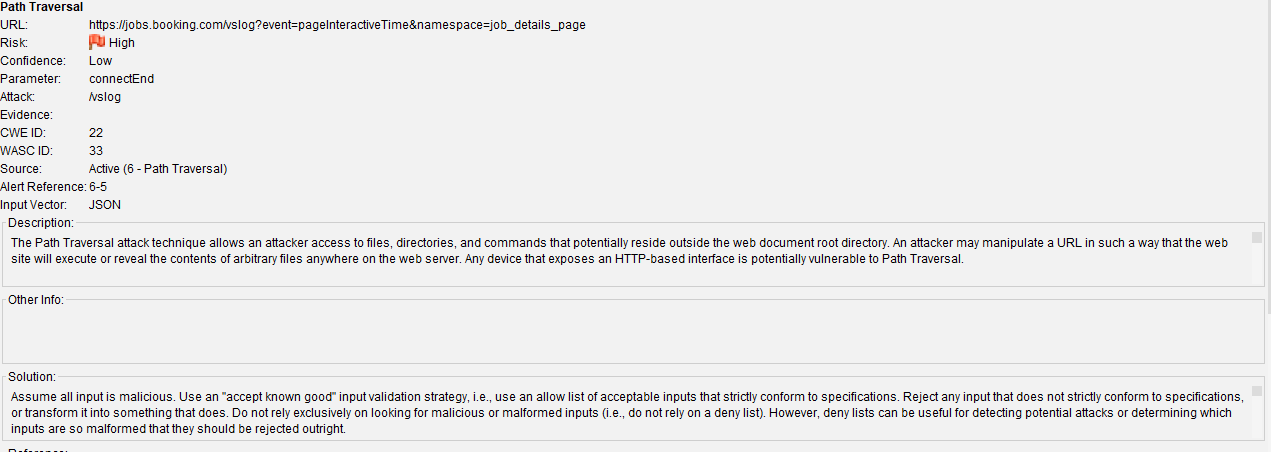


I fuzzed the directory traversal through the dotdotpwn tools in there most of them are false positive.

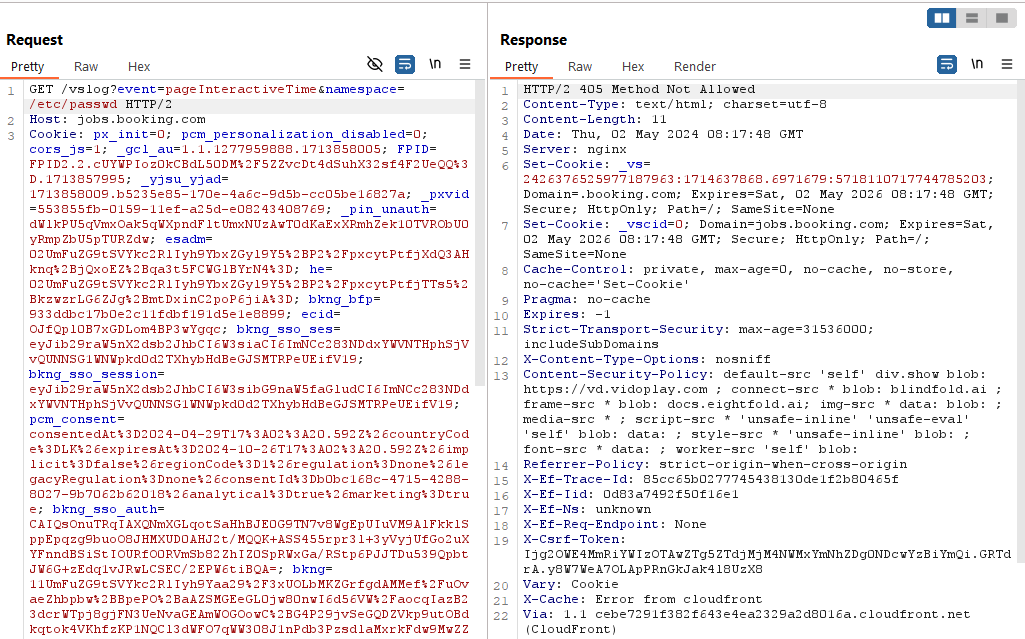


Below is a screen shot of Fuzzing using OWSAP ZAP

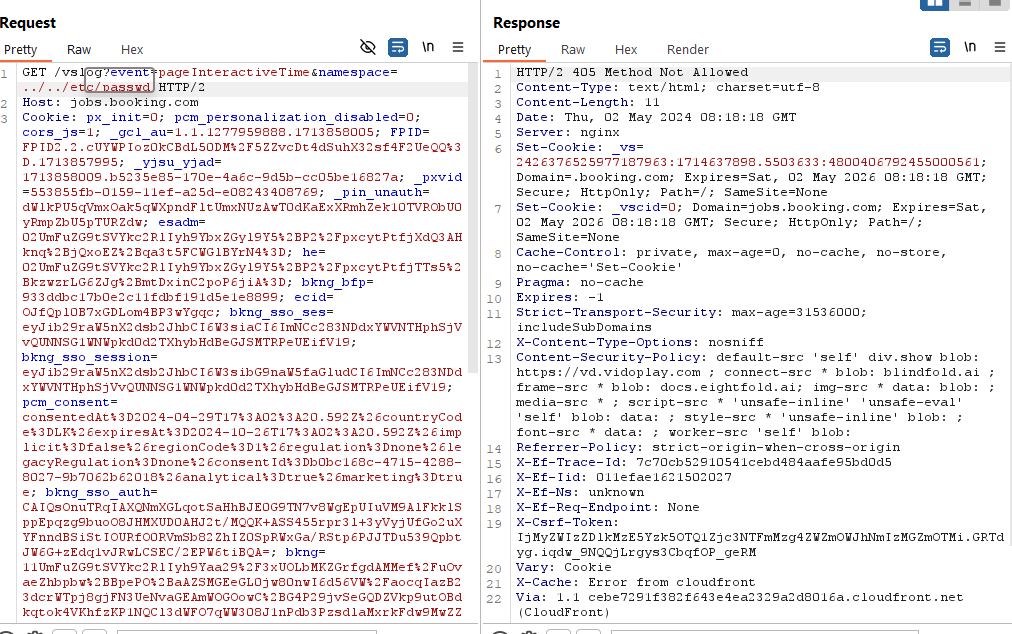


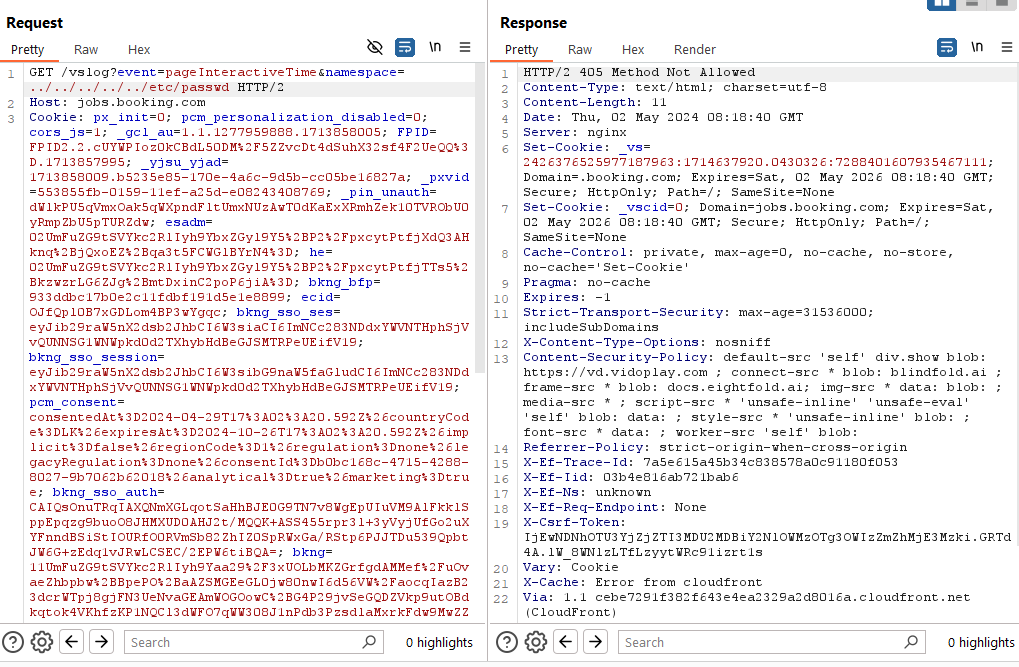


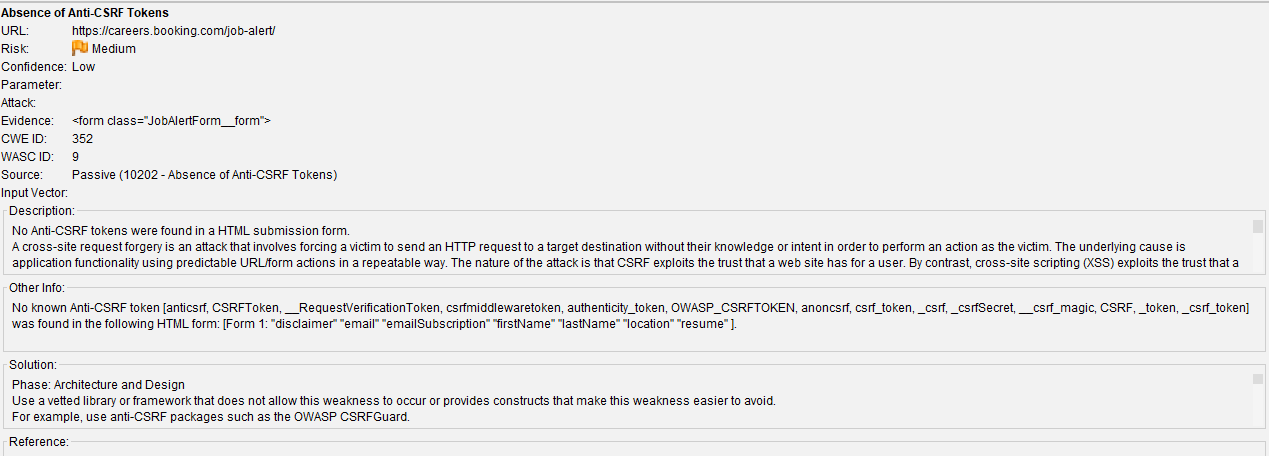
Below is a screen shot of manually testing the above mentioned directory traversal



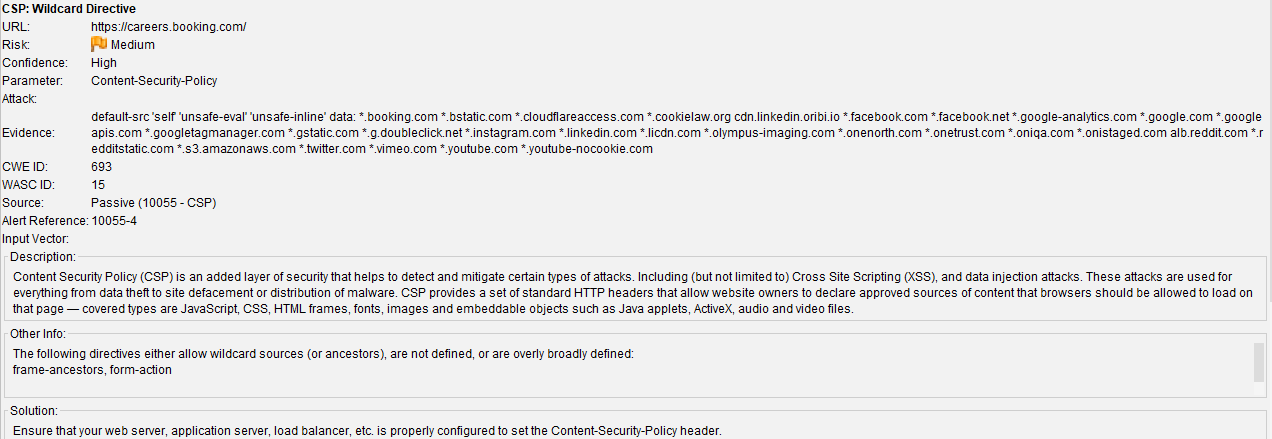
Repeatedly done the above process by moving one folder upward at each request





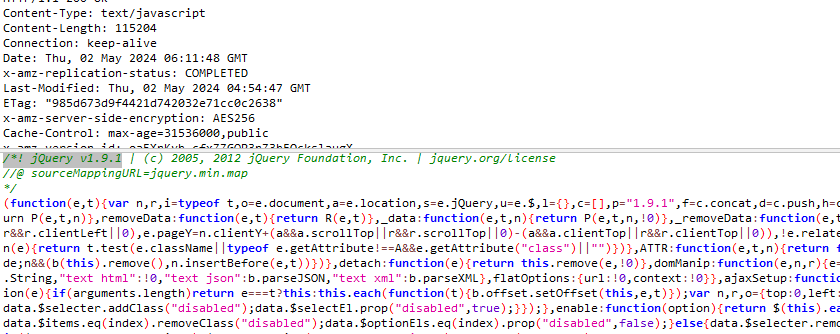




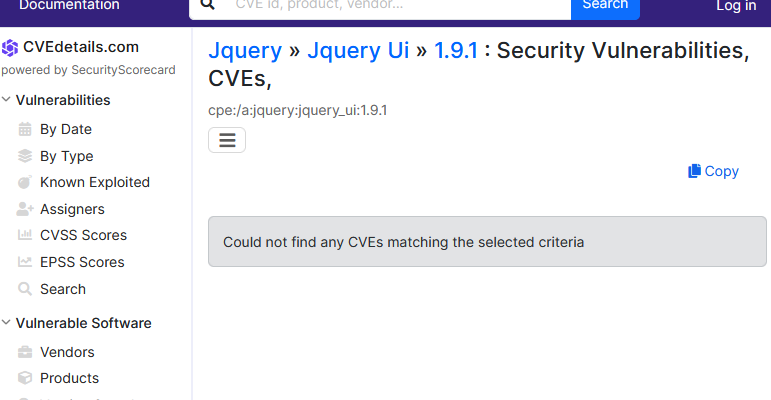








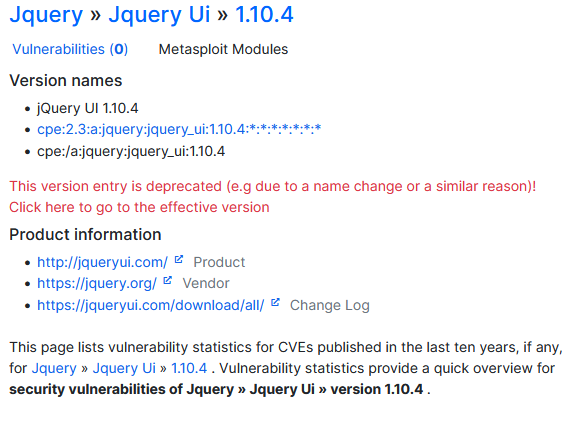
Vulnerabilities found in jQuery v1.9.1

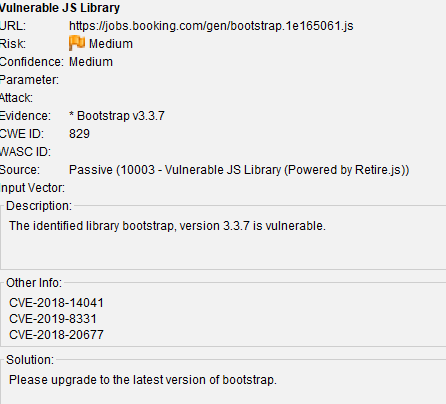


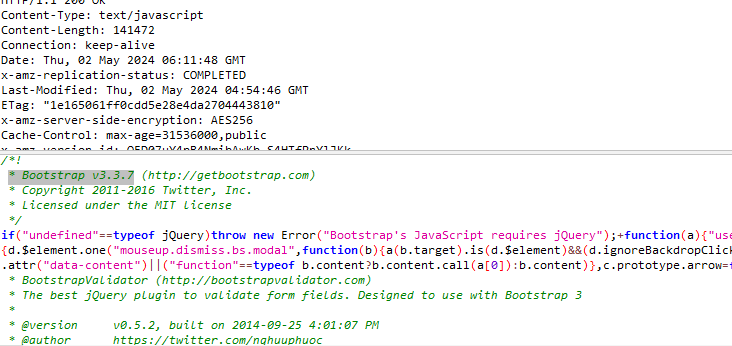




vulnerabilities found in jQuery UI v1.10.4

https://security.stackexchange.com/questions/215041/bootstrap-3-3-7-xss





<https://security.stackexchange.com/questions/215041/bootstrap-3-3-7-xss>



But this is not mentioned in the formally and there no such mention in the CVE details website also