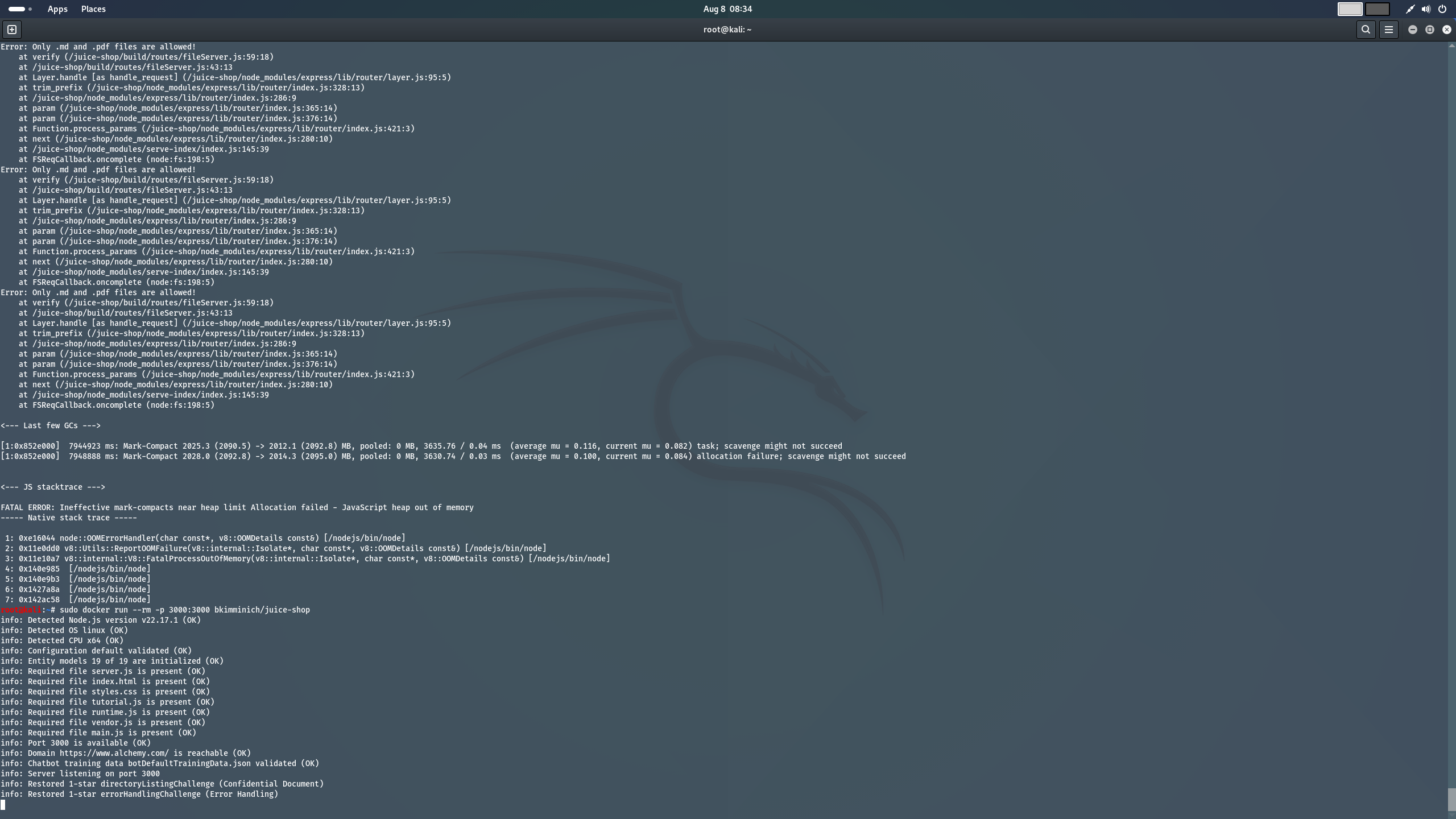
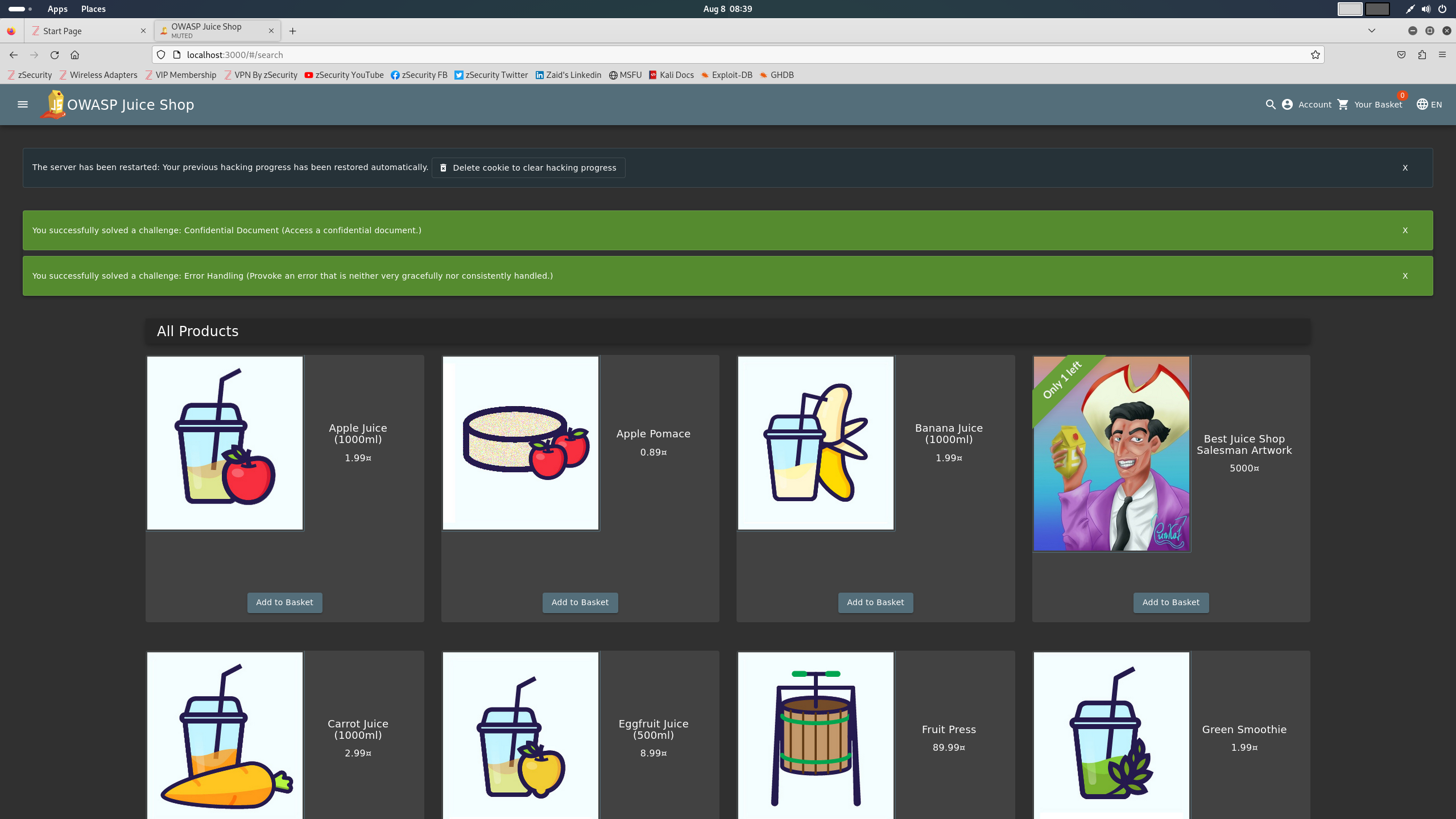
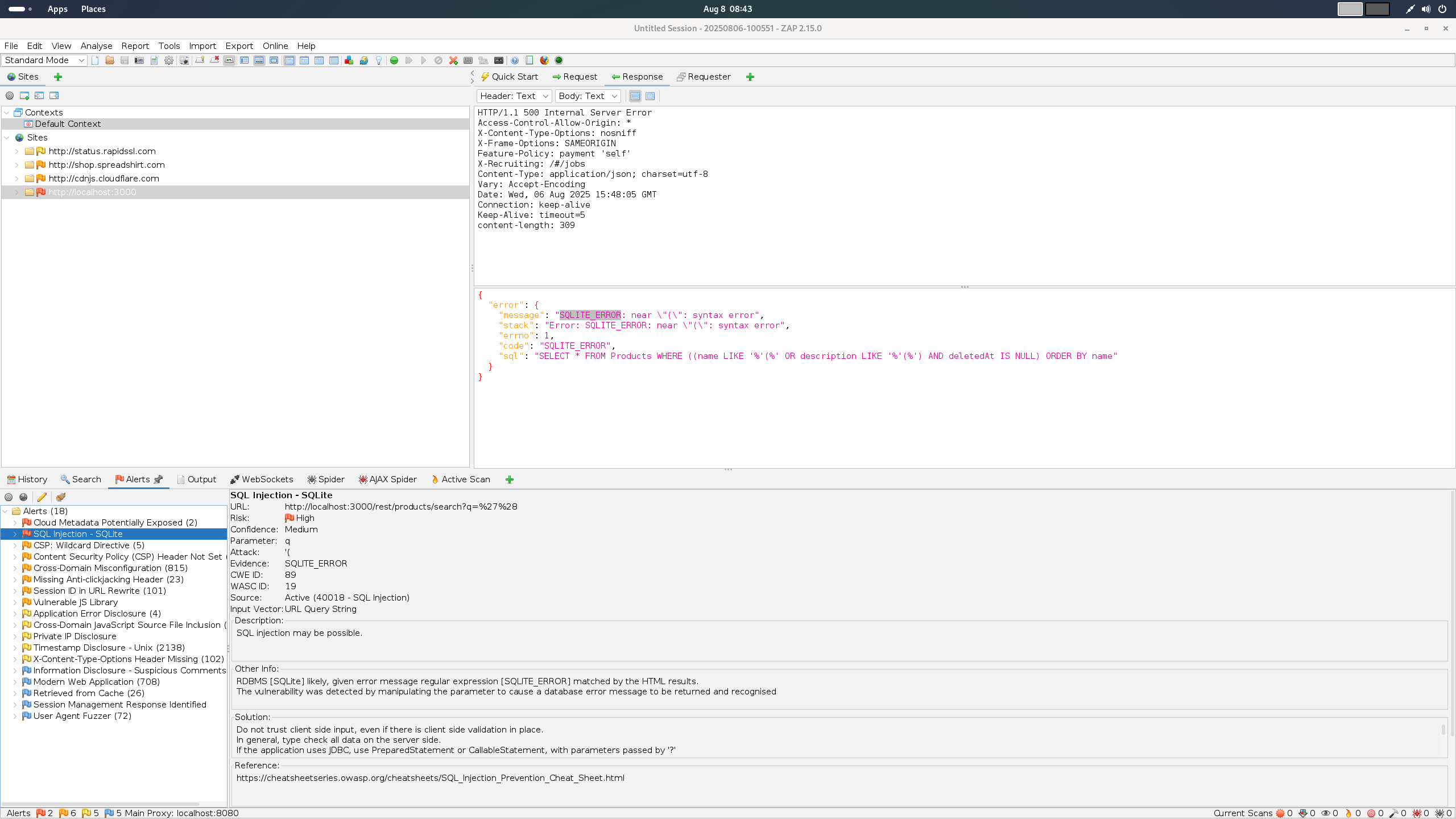
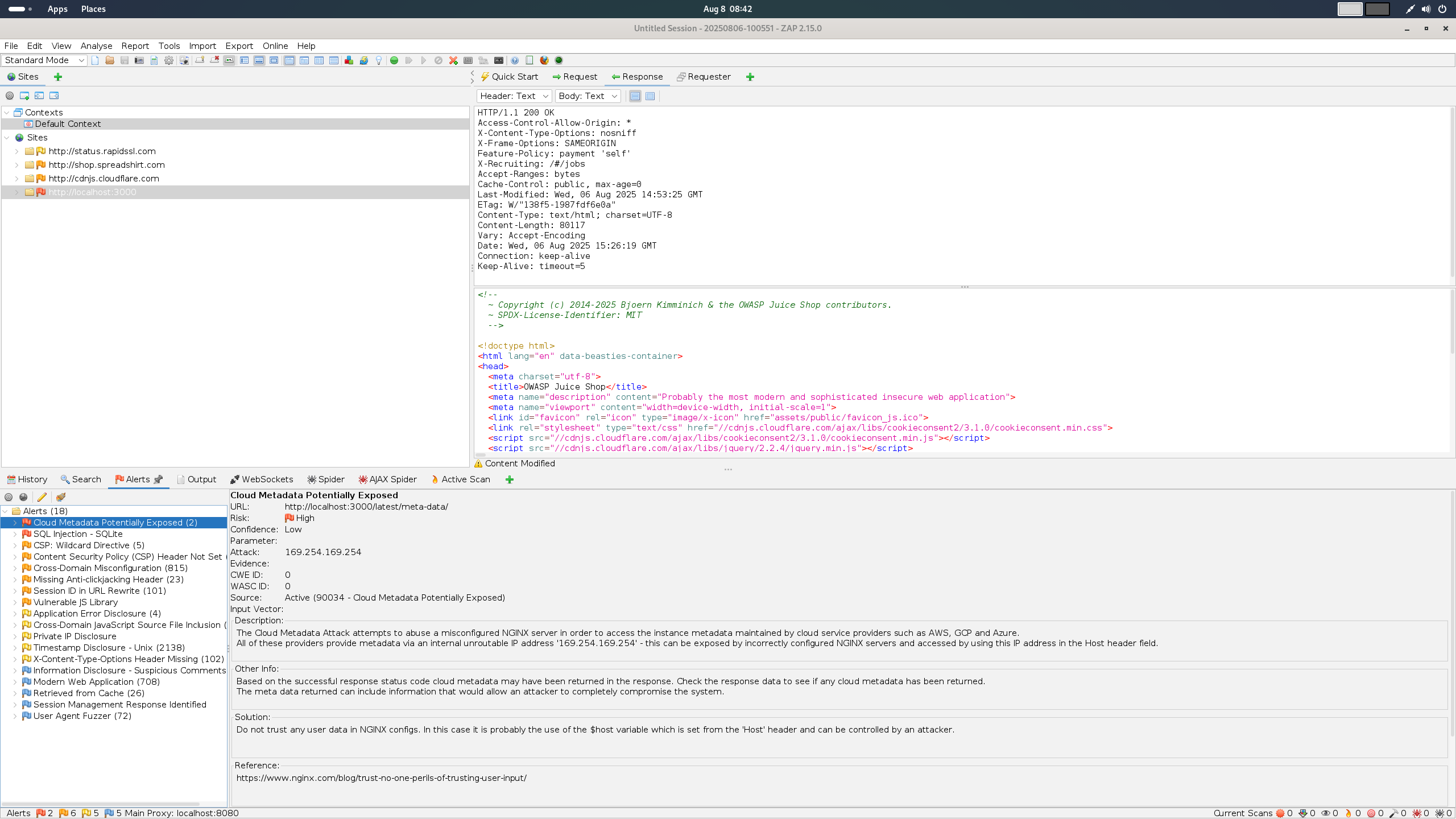
Executive Summary: Brief recap— “Scanned demo (Juice shop) app, found vulnerabilities, see attached details.”

Tools Used: Kali Linux, OWASP ZAP, Juice Shop.

Approach: “Set browser to proxy through ZAP, browsed app, ran automated scan.”

#### Key Findings Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vulnerability** | **Page/Parameter** | **Severity** | **Description** | **Recommendation** |
| Cloud Metadata Potentially Exposed | Metadata(Cloud) | High... | The Cloud Metadata Attack attempts to abuse a misconfigured NGINX server in order to access the instance metadata maintained by cloud service providers such as AWS, GCP and Azure.  All of these providers provide metadata via an internal unroutable IP address '169.254.169.254' - this can be exposed by incorrectly configured NGINX servers and accessed by using this IP address in the Host header field. | Do not trust any user data in NGINX configs. In this case it is probably the use of the $host variable which is set from the 'Host' header and can be controlled by an attacker |
| SQL injection may be possible | /search, /login... | High | RDBMS [SQLite] likely, given error message regular expression [SQLITE\_ERROR] matched by the HTML results.  The vulnerability was detected by manipulating the parameter to cause a database error message to be returned and recognised | Do not trust client side input, even if there is client side validation in place.  In general, type check all data on the server side.  If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' |

* **Screenshots**:
* 
* **Conclusion**: This demonstrates security testing & reporting process.