Task 4

Information Leakage

**Description:**The web application may reveal system data or debugging information by raising exceptions or generating error messages. Leakage of system data or debugging information through an output stream or logging function can allow attackers to gain knowledge about the application and craft specialized attacks on the it.

### Frame Injection

**Description:** Improper validation of input parameters could lead to attackers injecting frames to compromise confidential user information. Frame injection is a common method employed in [phishing](https://www.upguard.com/blog/phishing) attacks

### URL Redirection

**Description:**While it's common for web applications to redirect or forward users to other websites/pages, attackers commonly [exploit](https://www.upguard.com/blog/exploit) vulnerable applications without proper redirect validation in place. This can lead to malicious redirection to an untrusted page.

### Missing Session Timeout

**Description:** Attackers may gain unauthorized access to web applications if inactivity timeouts are not configured correctly.

### Session ID Cookies Not Marked Secure

**Description:** If session ID cookies for a web application are marked as secure, the browser will not transmit them over an unencrypted HTTP request. Not marking them as such allows cookies to be accessible and viewable in by attackers in clear text.

### Sensitive Information Cached

**Description:** Browsers typically store a copy of requested items in their caches: web pages, images, and more. This creates a security gap for applications that store, process, and display sensitive data, since attackers gaining access to the user's browser cache have access to any information contained therein.

### CRLF (Carriage Return and Line Feed) Injection

**Description:** CRLF exploits occur when malicious content is inserted into the browser's HTTP response headers after an unsuspecting user clicks on a malicious link. Hackers will typically inject malicious code into the user's browser through the web application/server, making casual detection difficult.