❖ **Bug Bounty [YesWeHack Dojo]**

1. Summary

- **Vulnerability Type:** Command Injection
- **Target URL/Application:** https://dojo-yeswehack.com/challenge/edit/22731f6e-1def-4853-a939-040c9d668f6d
- **Severity Level:** Low
- **Impact:** Allows attackers to execute arbitrary commands on the server, leading to sensitive data exposure.

2. Description

The web application contains a command injection vulnerability in its service availability feature when the user is identified as dev. The custom sanitization method used for dev users fails to properly sanitize inputs, leading to command execution.

3. Steps to Reproduce

1) Set the token parameter to a value that corresponds to a dev user.
2) Set the cmd parameter to:
   bash
   127.0.0.1; cat /tmp/flag.txt
3) Submit the request to the server.
4) Observe that the contents of the /tmp/flag.txt file are displayed.

4. Proof of Concept (PoC)

- **Payload Used:** `127.0.0.1; cat /tmp/flag.txt`
- **Response Output:** The response from the server contained the contents of the flag file:
  Copy code
  `FLAG{154e8e64cb5bef39d550bdff7a99d3a2}`

5. Impact Analysis

- An attacker can execute arbitrary system commands on the server, potentially leading to data theft, privilege escalation, and full server compromise.

6. Mitigation Recommendations

- Use `shlex.quote()` or a similar secure function to sanitize user inputs uniformly, regardless of user type.
- Avoid using shell commands when interacting with user input; use language-specific functions where possible.

7. Timeline

- **Date of Discovery:** October 17, 2024
- **Date of Report Submission:** October 17, 2024

8. Additional Information

- **Platform:** [YesWeHack]
- **Disclosure Status:** Private disclosure to the target company.