Vulnerability Report for Input Validation Issues – Part 3 in DIVA Application

- **Title:** Vulnerability Report for Input Validation Issues – Part 3 in DIVA Application

- **Severity:** Critical

- **Description:**

  The Diva application is susceptible to Denial of Service (DoS) attacks due to inadequate input validation. Attackers can exploit these vulnerabilities to overwhelm the application, causing service disruption or complete unavailability.

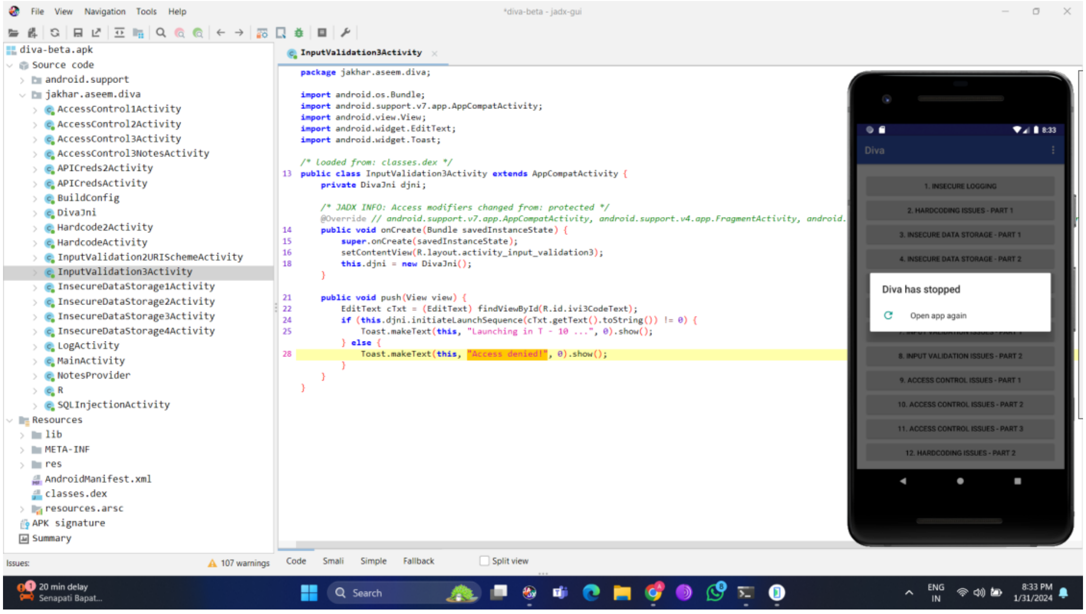
- **Impact:**

  1. Service Unavailability: Attackers can render the application inaccessible to legitimate users.
  2. Resource Exhaustion: Continuous exploitation may exhaust server resources, affecting overall system performance.
  3. Loss of Data Integrity: In some cases, a DoS attack may lead to data corruption or loss.

- **Steps to Reproduce:**

  1. Login to the application.
  2. Click on the “Input Validation Issues – Part 3” option.
  3. The application asks for a launch code.
  4. Enter a launch code,eg: 1234567890, and push the red button.
  5. If you get an error then try a longer code like:12345678901234567890.
  6. Keep increasing the size of the code until the application crashes.

- **<u>PoC (Proof of Concept)</u>:**



```java
package jakhar.aseem.diva;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

/* loaded from: classes.dex */
public class InputValidation3Activity extends AppCompatActivity {
    private DivaJni djni;

    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_input_validation3);
        this.djni = new DivaJni();
    }

    public void push(View view) {
        EditText cTxt = (EditText) findViewById(R.id.ivi3CodeText);
        if (this.djni.initiateLaunchSequence(cTxt.getText().toString()) != 0) {
            Toast.makeText(this, "Launching in T - 10 ...", 0).show();
        } else {
            Toast.makeText(this, "Access denied!", 0).show();
        }
    }
}
```

Diva has stopped

Open app again

- **<u>Remediation</u>:**

1. Implement Robust Input Validation: Apply proper input validation techniques to ensure that only valid and expected input is accepted.

2. Rate Limiting: Implement rate limiting mechanisms to restrict the number of requests from a single source within a specified time frame.

3. Use Web Application Firewall (WAF): Employ a WAF to filter and block malicious traffic, including DoS attempts.

4. Monitor and Alerting: Set up monitoring to detect abnormal traffic patterns and configure alerts for potential DoS attacks.

- **<u>CWE (Common Weakness Enumeration)</u>:**

1. CWE-770: Allocation of Resources Without Limits or Throttling
2. CWE-400: Uncontrolled Resource Consumption ('Resource Exhaustion')