Input Validation Testing

1. Introduction

Input validation testing is the process of verifying that the application correctly validates all input from users and other external sources. This is a critical aspect of web application security, as many of the most common and severe vulnerabilities, such as Cross-Site Scripting (XSS) and SQL Injection, are the result of improper input validation. This testing category covers a wide range of injection attacks, including XSS, SQLi, command injection, and more.

2. CVSS Score Summary

Test Case	Vulnerability	CVSS v3.1 Score	CVSS Vector	Severity
OTG-INPVAL-001	Reflected Cross-Site Scripting	6.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N	Medium
OTG-INPVAL-002	Stored Cross-Site Scripting	8.8	CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:H/I:H/A:H	High
OTG-INPVAL-003	HTTP Verb Tampering	5.3	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N	Medium
OTG-INPVAL-004	HTTP Parameter Pollution	4.3	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N	Low
OTG-INPVAL-005	SQL Injection	9.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	Critical
OTG-INPVAL-005-Blind	Blind SQL Injection	9.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	Critical
OTG-INPVAL-	LDAP Injection	0.0	N/A	Informational

006				
OTG-INPVAL-007	ORM Injection	0.0	N/A	Informational
OTG-INPVAL-008	XML Injection	0.0	N/A	Informational
OTG-INPVAL-009	SSI Injection	7.2	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:L	High
OTG-INPVAL-010	XPath Injection	0.0	N/A	Informational
OTG-INPVAL-011	IMAP/SMTP Injection	0.0	N/A	Informational
OTG-INPVAL-012	Code Injection (LFI/RFI)	7.5	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	High
OTG-INPVAL-013	Command Injection	9.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	Critical
OTG-INPVAL-014	Buffer Overflow	7.5	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	High
OTG-INPVAL-015	Incubated Vulnerability Testing	0.0	N/A	Informational
OTG-INPVAL-016	HTTP Splitting/Smuggling	0.0	N/A	Informational

3. Summary of Findings

The most critical vulnerabilities discovered in this assessment were in the input validation category. The application is highly vulnerable to SQL Injection, Blind SQL Injection, and Command Injection, all of which could allow an attacker to take complete control of the application and the underlying server.

Additionally, multiple Cross-Site Scripting (XSS) vulnerabilities were identified, which could be used to hijack user sessions and perform other malicious actions. A Local File Inclusion (LFI) vulnerability was also discovered, which could be used to read sensitive files from the server.

The primary challenge in this testing category was the sheer number of vulnerabilities discovered. The application appears to have very little in the way of input validation, making it an easy target for a wide range of attacks. The tests for LDAP, ORM, XML, XPath, and IMAP/SMTP injection also indicated potential vulnerabilities, but these would require further manual investigation to confirm. The buffer overflow test also indicated a potential Denial of Service (DoS) vulnerability, which should be investigated further.

4. General Recommendation/Remediation

- **Implement Input Validation and Sanitization:** All user-supplied input should be validated and sanitized on both the client-side and server-side. This includes using parameterized queries to prevent SQL injection, and encoding output to prevent XSS.
- **Use a Web Application Firewall (WAF):** A WAF can be used to detect and block common web application attacks, suchs as SQL injection and XSS.
- **Disable Unnecessary Features:** Features that are not required for the application to function, such as the `phpinfo()` page, should be disabled.
- **Implement the Principle of Least Privilege:** The application should be configured to run with the minimum level of privilege required to function. This will help to limit the damage that can be caused by a successful attack.
- **Regularly Patch and Update:** The application and all of its components should be regularly patched and updated to ensure that all known vulnerabilities are addressed.