TAMPERING WITH INPUT

How hard is it for an attacker to modify the data they submit to your system? Can they break a trust boundary and modify the code which runs as part of your system? Tampering with input can allow attackers to do things they are not supposed to do.

An example of tampering with input is when an attacker submits a SQL injection attack via a web application and uses that action to delete all the data in a database table.

KEY CONCEPTS:

- Integrity
- Injection
- Validation
- Whitelisting
- Blacklisting

Lack of validation server-side

- Fails to prevent stored Cross Site Scripting (XSS)
- Fails to prevent reflected XSS
- Fails to prevent SQL, XML (XXE) or LDAP injection
- Fails to prevent shell injection
- Fails to prevent an open redirect
- Fails to prevent Cross-site request forgery (CSRF)
- Framework support for mass binding can be exploited
- Alternate character encodings can be used to circumvent protections
- It is possible for attacker to tamper with cookies
- File upload feature fails to block malware

Lack of validation in browser

- Fails to prevent DOM based XSS
- Relies on browser based business logic for validation
- Scripts to display advertising contain malicious code
- Code injection is possible via JSON responses received from server
- Transfers between DOM contexts are subject to code injection
- It is possible for attacker to tamper with cookies

And what else?

ThoughtWorks®

thoughtworks.com