# Parameter Tampering

## Introduction

This repository focuses on common web application vulnerabilities, including parameter tampering, Cross-Site Scripting (XSS), and SQL Injection. This section provides an overview of parameter tampering, examples of how it can be exploited, and guidance on how to prevent it.

## Parameter Tampering

### Explanation

Parameter tampering is a type of web application attack where an attacker manipulates parameters exchanged between the client and server to modify application data, such as user credentials, permissions, or other sensitive information.

### Common Techniques

- Modifying URL query parameters
- Altering form data
- Changing cookie values
- Manipulating hidden fields in HTML forms

### Example Scenarios

1. **URL Query Parameter Tampering**
    - Original URL: `http://example.com/profile?user_id=123`
    - Tampered URL: `http://example.com/profile?user_id=456`
    - Impact: An attacker can access another user's profile by changing the `user_id` parameter.
2. **Form Data Tampering**
    - Original Form Data: `price=100&quantity=1`
    - Tampered Form Data: `price=1&quantity=100`
    - Impact: An attacker can manipulate the price and quantity of items in an online store.

### Code Snippets

1. **URL Query Parameter Tampering**

```
# Original URL
url = "http://example.com/profile?user_id=123"

# Tampered URL
tampered_url = url.replace("user_id=123", "user_id=456")
```

2. **Form Data Tampering**

```html
<!-- Original Form -->
<form action="/purchase" method="POST">
    <input type="hidden" name="price" value="100">
    <input type="hidden" name="quantity" value="1">
    <button type="submit">Buy</button>
</form>

<!-- Tampered Form -->
<form action="/purchase" method="POST">
    <input type="hidden" name="price" value="1">
    <input type="hidden" name="quantity" value="100">
    <button type="submit">Buy</button>
</form>
```

### Prevention Techniques

- Validate and sanitize all user inputs on the server side.
- Use strong authentication and authorization mechanisms.
- Implement secure coding practices, such as using prepared statements and parameterized queries.
- Avoid relying solely on client-side validation.

## Conclusion

Parameter tampering is a serious security threat that can lead to unauthorized access and data manipulation. By understanding common techniques and implementing prevention measures, developers can protect their applications from such attacks.

For further reading, refer to the following resources: - OWASP: Parameter Tampering - OWASP: Input Validation