

## Web Parameter Tampering

Watch 41 Star 93

Thank you for visiting OWASP.org. We recently migrated our community to a new web platform and regrettably the content for this page needed to be programmatically ported from its previous wiki page. There's still some work to be done.

### Description

The Web Parameter Tampering attack is based on the manipulation of parameters exchanged between client and server in order to modify application data, such as user credentials and permissions, price and quantity of products, etc. Usually, this information is stored in cookies, hidden form fields, or URL Query Strings, and is used to increase application functionality and control.

This attack can be performed by a malicious user who wants to exploit the application for their own benefit, or an attacker who wishes to attack a third-person using a [Man-in-the-middle attack](#). In both cases, tools like WebScarab and Paros proxy are mostly used.

The attack success depends on integrity and logic validation mechanism errors, and its exploitation can result in other consequences including [XSS](#), [SQL Injection](#), file inclusion, and path disclosure attacks.

For a short video clip describing the vulnerability, [click here](#) (Courtesy of [Checkmarx](#))

### Examples

#### Example 1

The parameter modification of form fields can be considered a typical example of Web Parameter Tampering attack.

**The OWASP Foundation** works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

### Important Community Links

[Community](#)  
[Attacks \(You are here\)](#)  
[Vulnerabilities](#)  
[Controls](#)

### Upcoming Global Events

[Global AppSec SF October 19th-23rd](#)  
[Global AppSec Dublin February 15-19th, 2021](#)

For example, consider a user who can select form field values (combo box, check box, etc.) on an application page. When these values are submitted by the user, they could be acquired and arbitrarily manipulated by an attacker.

### Example 2

When a web application uses hidden fields to store status information, a malicious user can tamper with the values stored on his browser and change the referred information. For example, an e-commerce shopping site uses hidden fields to refer to its items, as follows:

```
<input type="hidden" id="1008" name="cost" value="70.00">
```

In this example, an attacker can modify the "value" information of a specific item, thus lowering its cost.

### Example 3

An attacker can tamper with URL parameters directly. For example, consider a web application that permits a user to select his profile from a combo box and debit the account:

```
http://www.attackbank.com/default.asp?profile=741&debit=1000
```

In this case, an attacker could tamper with the URL, using other values for profile and debit:

```
http://www.attackbank.com/default.asp?profile=852&debit=2000
```

Other parameters can be changed including attribute parameters. In the following example, it's possible to tamper with the status variable and delete a page from the server:

```
http://www.attackbank.com/savepage.asp?nr=147&status=read
```

Modifying the status variable to delete the page:

```
http://www.attackbank.com/savepage.asp?nr=147&status=del
```

## Related Threat Agents

- :Category:Client-side Attacks
- :Category:Logical Attacks

## Related Attacks

- SQL Injection
- XSS Attacks
- Path Traversal

## Related Vulnerabilities

- :Category: Input Validation Vulnerability

## Related Controls

- :Category: Input Validation

## References

- <http://cwe.mitre.org/data/definitions/472.html> - Web Parameter Tampering
- [http://www.imperva.com/application\\_defense\\_center/glossary/parameter\\_tampering.html](http://www.imperva.com/application_defense_center/glossary/parameter_tampering.html)
  - Parameter Tampering Imperva - Application Defense Center
- <http://www.cgisecurity.com/owasp/html/ch11s04.html> - Parameter Manipulation - Chapter 11. Preventing Common Problems

Category:OWASP ASDR Project Category: Injection Category: Attack

 [Edit on Github](#)

## Spotlight: Avatao

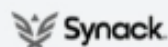
# avatao

Avatao is an online training platform for building secure software. It offers a rich library of hands-on IT security exercises for software engineers to teach secure programming from design to deployment in a fun and intuitive way. Educating for secure software can significantly improve software quality, increase trust in a company's brand, reduce development costs and reduce the risks of suffering significant losses from cyber-attacks. In today's world, secure software development practices are a basic requirement and we believe the solution should be integrated into the ground roots of software development teams. The platform can be used to provide security awareness to software engineers, to enrich existing IT security training, meetups or allow self-directed learning for continuous professional development. Avatao hackathons are also an excellent method to discover talented developers and engage and build brand awareness in the community.

## Corporate Supporters



detectify



ThoughtWorks



Open Web Application Security Project, OWASP, Global AppSec, AppSec Days, AppSec California, SnowFROC, LASCON, and the OWASP logo are trademarks of the OWASP Foundation. Unless otherwise specified, all content on the site is Creative Commons Attribution-ShareAlike v4.0 and provided without warranty of service or accuracy. For more information, please refer to our [General Disclaimer](#). Copyright 2020, OWASP Foundation, Inc.