

## XSS Vulnerability in Zen Cart

**Title:** XSS Vulnerability in Zen Cart

**Release Date:** 2011/11/22

**Vulnerable Application:** Zen Cart (ver. 1.3.9h)

**Type:** Cross Site Scripting

**Level:** Low (Low/High/Critical)

**CVSS:** 4 (AV:N/AC:H/Au:N/C:P/I:P/A:N)

**Dognædis Ref.:** DGS-SEC-8

**CVE Ref.:** CVE-2011-4547

**Other Ref.:**

**Discover Credits:** CodeV - Code Analyzer

**Bulletin Author(s):** RPinto

**Contact:** irt@dognædis.com

**Nível de Acesso do Documento:** Público

A informação expressa neste documento é propriedade da Dognædis. Pode ser, no entanto, lida, copiada, distribuída, impressa ou acedida por qualquer pessoa ou entidade, desde que os créditos da Dognædis sejam respeitados.

**Document Access Level:** Public

The information expressed in this document is property of Dognædis. Although can be disclosed, distributed, copied, read, used, printed or accessed by anyone, since all the Dognædis credits are respected.

### Overview:

Zen Cart is an online store management system. It is PHP-based, using a MySQL database and HTML components. Support is provided for numerous languages and currencies, and it is freely available under the GNU General Public License.

### Scope:

**File:** /includes/templates/template\_default/templates/tpl\_gv\_send\_default.php

**Vulnerable Argument(s):** \$\_POST['message']

**Code:**

```
line 58: <div id="gvSendDefaultMessage" class="content"><?php echo stripslashes($_POST['message']);
?></div>
```

### Proof(s) of Concept:

GET: [https://<app\\_base>/index.php?main\\_page=gv\\_send&action=send](https://<app_base>/index.php?main_page=gv_send&action=send)

POST: message=</textarea><script>alert("XSS");</script><textarea>

### Description:

The referred vulnerability could be exploited through a XSS (Cross-Site-Scripting) attack.

Ultimately, the attacker could take complete control of the victims web-browser.

In a successful attack, the malicious script would be executed with the authenticated user permissions.

### Impact:

Generally, by exploiting this kind of vulnerability, it might be possible to achieve possible attack vectors to various kinds of attacks such as:

- Session/Cookie theft
- Account Hijacking
- Identity theft
- Accessing confidential resources
- Accessing pay content
- Account Denial of service

### Resolution:

Aiming a correct resolution of the identified vulnerability, the data obtained through the \$\_POST['message'] input argument should be properly sanitized for HTML and following ECMAS usage.

### Official Solution:

At the moment, there is no official solution for the reported vulnerabilities.

This vulnerability will be solved in the version 1.5, which is, according to the developer, due for release shortly.

**External References:**

[http://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](http://www.owasp.org/index.php/Cross-site_Scripting_(XSS))

[http://www.owasp.org/index.php/Cross-Site\\_Request\\_Forgery\\_\(CSRF\)](http://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF))

[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)

[http://en.wikipedia.org/wiki/Code\\_injection](http://en.wikipedia.org/wiki/Code_injection)

