MOPS-2010-002: Campsite TinyMCE Article Attachment SQL Injection Vulnerability

May 1st, 2010

A SQL Injection vulnerability was discovered in the TinyMCE custom article attachment plugin within Campsite that allows retrieving all data from the database.

Affected versions

Affected is Campsite <= 3.3.5

Credits

The vulnerability was discovered by Stefan Esser during the Month of PHP Security SQL Injecton Marathon.

About Campsite

Campsite is a multilingual web publishing system that can bring your newspaper or magazine content to the online world. It is often used by media organizations who also have a printed version of their publication, and enables them to increase their revenues with online subscriptions and ads. There are many systems that might seem similar to Campsite, but it is the only open source system designed to work in the same style of newspapers and magazines – for example, with multiple journalists, editor review, issue publishing, and subscription management.

Detailed information

This vulnerability was discovered during SQL Injection Marathon a PHP code auditing marathon performed by Stefan Esser. The basic idea of this initiative is to select random PHP applications and perform a short code audit on them. The maximum time spent on each application is 30 minutes and after the first found SQL injection usually the next application is audited.

During SQL Injection Marathon Campsite 3.3.5 was audited and a very simple SQL Injection could be found in a custom campsite article attachment plugin for the TinyMCE editor. The vulnerable code is very easy to spot inside the /javascript/tinymcs/plugins/campsiteattachment/attachments.php file.

```
// Get the list of files and directories
$list = $manager->getFiles($_REQUEST['article_id'], $languageSelected);
```

Here the getFiles() method is called directly with the unsanitized user input $_REQUEST['article_id'].

```
function getFiles($p_articleId, $p_languageId = null)
{
    $files = array();

if ($this->isValidBase() == false)
    return $files;

$articleAttachments = ArticleAttachment::GetAttachmentsByArticleNumber($p_articleId, $p_lang
```

The article_id is passed down to the internal method ArticleAttachment::GetAttachmentsByArticleNumber() where it is copied into an SQL query as is, which results in an easy exploitable SQL injection vulnerability.

```
public static function GetAttachmentsByArticleNumber($p_articleNumber, $p_languageId = null)
{
    ...
    $queryStr = 'SELECT '.$columnNames
            .' FROM Attachments, ArticleAttachments'
            .' WHERE ArticleAttachments.fk_article_number='.$p_articleNumber
            .' AND ArticleAttachments.fk_attachment_id=Attachments.id'
            ." AND (Attachments.fk_language_id IS NULL OR $langConstraint)"
            .' ORDER BY Attachments.time_created asc, Attachments.file_name asc';
    $rows = $g_ado_db->GetAll($queryStr);
```

Proof of concept, exploit or instructions to reproduce

The following proof of concept URL should be enough to test this vulnerability

http://campsite.audit/javascript/tinymce/plugins/campsiteattachment/attachments.php?article_id=0+U

Notes

The vendor already released a patch to fix this vulnerability at http://www.campware.org/en/camp/campsite_news/832/. The released patch ensures that the article_id parameter is numeric – the disclosed attack vector is therefore closed. However the released patch also introduces the call to a database escaping function into the internal function that has no effect because the parameter is not put within quotes.
