STRCAT

The strcat() function is unsafe and should not be used.

Sean Barnum, Cigital, Inc. [vita[1]]

Copyright © 2007 Cigital, Inc.

2007-04-17

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6296 bytes

Attack Category	• Malicious Input
Vulnerability Category	• Buffer Overflow • Unconditional
Software Context	• String Management
Location	
Description	The strcat() function is unsafe and should not be used. The strcat() function will concatenate two strings by placing the second input on the end of the first. If the space in the first buffer is not capable of storing both strings, the data of the second string will overflow into the adjacent memory space.
APIs	

Function Name	Comments
_ftcscat	Windows
_mbscat	
_tcscat	Windows
lstrcat	Windows
lstrcatA	Windows
lstrcatW	Windows
mbscat	
olestrcat	ATL String cat macros.
strcat	
strccat	
StrCatA	Windows
StrCatW	Windows
wcscat	Windows

Method of Attack	strcat() and related functions do no bounds checking, providing an attacker with opportunity to overflow the buffer. Any time an attacker provides an input string greater than the length of the destination

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

	buffer, the buffer can be overrun, causing a buffer overflow.
Exception Criteria	
Solutions	

Solution Applicability	Solution Description	Solution Efficacy
Whenever strings need to be concatenated.	Never use strcat() or related functions. Replace the strcat(dest, src) call with a safer replacement that does bounds checking. There are many of these, depending on the platform and language used. Some replacement routines are better at handling this than others. 1) Generic: Replace strcat(dest, src) with strncat(dest, src, count), where count is the maximum number of characters to append. Care must be taken to ensure that the size of dest will in fact hold that many characters. For multiple calls to strncat(), the user must keep track of how many characters have been written and properly	Varies, depending on replacement function chosen.

		account for the shrinking available space remaining in dest. See the STRNCAT rule for more information on using strncat(). On some systems, replacements such as strlcat() have been implemented to deal with null termination and null strings in a more predictable fashion. 2) On Windows (SHLWAPI), replace strcat(dest, src) with StrCbCat(dest, sizeof(dest), src) or StrCchCat(dest, sizeof(dest), src). Check the error code for STRSAFE_E_INSUFFICIENT_BUFFER and ensure that the buffer is null terminated by inserting the statement dest[sizeof(dest)-1] = '\0'. 3) Replace strcat(dest, src) with snprintf(dest, sizeof(dest), "%s%s", dest, src). Ensure that the buffer is null terminated by inserting a statement dest[sizeof(dest)-1]='\0'.		

		See the SNPRINTF rule for issues about snprintf(), including the fact that it is non-standard and can return different results depending on the condition of the input strings.	

Signature Details	The strcat() function is called.

Examples of Incorrect Code	`char str1[10];`
	`strcpy(str1,"abc");`
	`char str2[]="123456789";`
	`strcat(str1,str2);`

Examples of Corrected Code	`/* The following works if one must use a standard replacement function. However, on many platforms there are better choices than strncat() */`
	`char str1[10];`
	`strcpy(str1,"abc");`
	`char str2[]="123456789";`
	`int count;`
	`if (sizeof(str1)>=strlen(str1)+1){`
	`strncat(str1,str2,sizeof(str1)-strlen(str1)-1); /* Specify buffer size to avoid overflow. Note truncation occurs. */`
	`str1[sizeof(str1)-1] = '\0'; /* Ensure null termination */`
	`}`
	`/* Note: using sizeof() works given local allocation, but if malloc is used then malloc size must be specified instead of relying on sizeof() */`

Source Reference	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winui/winui/windowsuserinterface/sec_winui.asp[2]
Recommended Resource	
Discriminant Set	**Operating System**: • Any **Languages**: • C • C++

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com[1].

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com