

[NT] Unsafe Functions in Office Web Components

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-08/0098.html>

From: support@securiteam.com

Date: 08/25/02

From: support@securiteam.com

To: list@securiteam.com

Date: Sun, 25 Aug 2002 22:35:08 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

Unsafe Functions in Office Web Components

SUMMARY

The Office Web Components (OWC) contain several ActiveX controls that give users limited functionality of Microsoft Office in a web browser without requiring that the user install the full Microsoft Office application. This allows users to utilize Microsoft Office applications in situations where installation of the full application is infeasible or undesirable.

The control contains three security vulnerabilities, each of which could be exploited either via a web site or an HTML mail. The vulnerabilities result because of implementation errors in the following methods and functions the controls expose:

* Host(). This function, by design, provides the caller with access to applications' object models on the user's system. By using the Host() function, an attacker could, for instance, open an Office application on the user's system and invoke commands there that would execute operating system commands as the user.

* LoadText(). This method allows a web page to load text into a browser window. The method does check that the source of the text is in the same domain as the window, and in theory should restrict the page to only loading text that it hosts itself. However, it is possible to circumvent this restriction by specifying a text source located within the web page's

Securiteam: [NT] Unsafe Functions in Office Web Components

domain, and then setting up a server-side redirect of that text to a file on the user's system. This would provide an attacker with a way to read any desired file on the user's system.

* Copy()/Paste(). These methods allow text to be copied and pasted. A security vulnerability results because the method does not respect the "disallow paste via script" security setting in IE. Thus, even if this setting had been selected, a web page could continue to access the copy buffer, and read any text that the user had copied or cut from within other applications.

The patch does not set "kill bit" on the control, for reasons discussed in the FAQ.

DETAILS

Affected Software:

- * Microsoft Office Web Components 2000
- * Microsoft Office Web Components 2002

Products which Include the Affected Software:

- * Microsoft BackOffice® Server 2000
- * Microsoft BizTalk® Server 2000
- * Microsoft BizTalk Server 2002
- * Microsoft Commerce Server 2000
- * Microsoft Commerce Server 2002
- * Microsoft Internet Security and Acceleration Server 2000
- * Microsoft Money 2002
- * Microsoft Money 2003
- * Microsoft Office 2000
- * Microsoft Office XP
- * Microsoft Project 2002
- * Microsoft Project Server 2002
- * Microsoft Small Business Server 2000

Mitigating factors:

Overall:

* In the case of the web-based attack, an attacker would need to force a user to visit the attacker's Web site. Users who exercise caution in visiting web sites could minimize their risk.

* In the web based attack, If ActiveX controls have been disabled in the zone in which the page were viewed, the vulnerability could not be exploited. Users who place untrusted sites in the Restricted Sites zone, which disables ActiveX by default, or have disabled ActiveX controls in the Internet zone could minimize their risk.

* In the case of HTML email based attacks, customers who read email in the Restricted Sites zone would be protected against attempts to exploit this vulnerability. Customers using Outlook 2002 and Outlook Express 6.0, as well as Outlook 2000 and Outlook 98 customers who have applied the

Securiteam: [NT] Unsafe Functions in Office Web Components

Outlook Email Security Update would thus be protected by default. In addition, Outlook Express 5.0 customers who have chosen to read mail in the Restricted Sites zone would be protected by default.

* In the HTML email based attack, Outlook 2002 customers who have enabled the "Read as Plain Text" option available in SP1 or later would also be protected.

Host() Vulnerability:

* The attacker's code would be limited by restrictions on the user's account. Users of non-privileged accounts would limit the potential damage from a successful attack.

LoadText():

* The attacker would need to know the full path and name of the file.

Copy()/Paste():

* The vulnerability could enable an attacker to access only to information in the Windows clipboard. The information in the clipboard is unpredictable and this vulnerability gives no means for an attacker to target and retrieve specific information. Further, it is possible for the clipboard to be empty, which would yield an attacker nothing.

* The security setting in question is not enabled by default. Thus, the vulnerability does not present a threat to the default installation.

Patch availability:

Download locations for this patch

* Microsoft recommends that users install the Office XP SP2 update using the Office Product Updates site.

* General Patch:

<<http://support.microsoft.com/default.aspx?scid=kb;%5bLN%5d:Q322382>>
<http://support.microsoft.com/default.aspx?scid=kb;%5bLN%5d:Q322382>

* Microsoft Project 2002:

<<http://office.microsoft.com/downloads/2002/prj1001.aspx>>
<http://office.microsoft.com/downloads/2002/prj1001.aspx>

* Microsoft Project Server 2002:

<<http://office.microsoft.com/downloads/2002/ps1001en.aspx>>
<http://office.microsoft.com/downloads/2002/ps1001en.aspx>

* Office Web Components Download:

<<http://office.microsoft.com/downloads/2002/owc10.aspx>>
<http://office.microsoft.com/downloads/2002/owc10.aspx>

What is does this patch do?

This patch eliminates three publicly disclosed vulnerabilities in the Office Web Components:

* A vulnerability which could be used by an attacker to run a program on the user's local system.

* A vulnerability that could allow an attacker to read the contents of a file on the local system

Securiteam: [NT] Unsafe Functions in Office Web Components

* A vulnerability that could allow an attacker to read the contents of the local clipboard.

In addition, it also changes the functionality of Office Web Components, based on an internal review of those functions. As such, it enhances the overall security of the product and eliminates additional potential vulnerabilities.

Are there any caveats associated with the patch?

Yes. Customers should be aware that although the vulnerabilities here involve an ActiveX control, the patch does not set the "Kill Bit".

What is an ActiveX control?

ActiveX controls are small, single-purpose programs that can be called by programs and web pages. ActiveX allows a programmer to write a piece of software one time, and make its functionality available to other programs that may need it.

What is the "Kill Bit"?

The Kill Bit is a method by which an ActiveX control can be prevented from ever being invoked via Internet Explorer, even if it is present on the system. (More information on the Kill Bit is available in Microsoft Knowledge Base article Q240797). Typically, when a security vulnerability involves an ActiveX control, the patch delivers a new control and sets the Kill Bit on the vulnerable control. However, it is not feasible to do so in this case.

Why is not it feasible to set the Kill Bit in this case?

The ActiveX control involved in these vulnerabilities is used in Web pages produced by Office applications to access data. Many applications, including third-party applications, contain hard-coded references to it; if the patch set the Kill Bit, the web pages would no longer function at all – even with the new, corrected version. As a result, the patch updates the control to remove the vulnerabilities, but does not provide a brand-new control and set the Kill Bit on the old one.

What the risk associated with taking this approach?

Because the ActiveX control at issue here has been digitally signed by Microsoft, and the signature is still valid, it could be possible under certain conditions for an attacker to re-introduce the old, vulnerable version of the control onto a system that had been patched, thereby making it vulnerable again. In order for this happen, though, the user would need to either visit a web site operated by a malicious person or open an HTML mail from one.

Why would an attacker be able to silently re-introduce the old version of the control? Should not there be a warning message?

A warning message is generated anytime there is an error associated with a digital signature (e.g., a bad signature or expired certificate) or the signer is not trusted. In this case, the digital signature on the old version of the control is still valid, and the signer is Microsoft – which

Securiteam: [NT] Unsafe Functions in Office Web Components

is a trusted publisher in most cases. Because of this, most users would not see a warning message of any kind if the old control were re-introduced.

If someone did try re-introducing the old version of the control, how difficult would it be?

Because of the size of the ActiveX control at issue here, it could be difficult for an attacker to successfully re-introduce the old control.

The Office Web Components control is over 7 MB in size. Even if an attacker did attempt to download the old, vulnerable control, there would be unmistakable signs – for instance, sustained heavy disk usage – and the download operation would take some time to complete. The latter would be especially true if the user were operating over a dial-up connection or other low-speed networking session.

Why not revoke the certificate that was used to sign the control?

The certificate that was used to sign the control is still valid – the problem lies in the control, not the certificate. In addition, a number of controls have been signed using the same certificate, and revoking the certificate would cause all of them to become invalid.

How can I prevent the control from being silently re-introduced onto my system?

The simplest way is to make sure you have no trusted publishers, including Microsoft. If you do that, any attempt by either a web page or an HTML mail to download an ActiveX control will generate a warning message. Here is how to empty the Trusted Publishers list:

- 1) In Internet Explorer, choose Tools, then Internet Options.
- 2) Select the Content tab. In the Certificates section of the page, click on Publishers.
- 3) In the Certificates dialog, click on the Trusted Publishers tab.
- 4) For each certificate in the list, click on the certificate and then select Remove. Confirm that you want to remove the entry.
- 5) When you have removed all entries from the list, select Close to close the Certificates dialog, then click on OK to close the Internet Options dialog.

After emptying the Trusted Publishers list, if I do see a warning saying that a web site or an HTML mail wants to download a control, how can I decide whether to let it proceed?

The best criterion to use is whether you trust the web site or the sender of the HTML mail. If you do not, cancel the download.

Will Microsoft eventually set the Kill Bit on this control?

Yes. Microsoft is developing a new technology that will enable it to set the Kill Bit on the vulnerable version of the control without forcing users to re-author web pages containing references to these controls. When the new technology is available, we will ensure that this fix uses it.

Securiteam: [NT] Unsafe Functions in Office Web Components

What are the Office Web Components (OWC)?

The OWC are a series of ActiveX components that provide limited Office functionality to users with a web browser, without requiring that the user have installed the full version of Office.

The user can then use this limited Office functionality within their browser to manipulate and analyze data. For example, the OWC can be used to show spreadsheet data in a web page to a user thus giving the user some of the key functionality of Excel, without requiring that it be installed.

How is the OWC distributed?

The OWC is distributed in two ways: either as a standalone downloads or included with other Microsoft products.

* It is available as a separate, independent download from the Microsoft Web site. Users can go to the Microsoft Web site, download the OWC and then install it.

* In the case where the OWC is included with other applications, the OWC are included in the installation routines of the other application. In some cases, the OWC will be called out in the installation options as a discrete entity that users can choose to install or not install. In other cases, the files that constitute the OWC application will be included in the broader installation routines and not explicitly identified.

How can I tell if I have the OWC if it's included with other applications?

As part of our investigation into this vulnerability, Microsoft made a search of currently supported applications to determine what applications ship the OWC. The list of supported applications that ship the versions of OWC that contain the vulnerability are:

- * Microsoft BackOffice Server 2000
- * Microsoft BizTalk Server 2000
- * Microsoft BizTalk Server 2002
- * Microsoft Commerce Server 2000
- * Microsoft Commerce Server 2002
- * Microsoft Internet Security and Acceleration Server 2000
- * Microsoft Money 2002
- * Microsoft Money 2003
- * Microsoft Office 2000
- * Microsoft Office XP
- * Microsoft Project 2002
- * Microsoft Project Server 2002
- * Microsoft Small Business Server 2000

No other applications that are currently supported ship with the OWC.

What do the patches do?

The patches address the vulnerabilities by eliminating those methods and functions that are unsafe for untrusted sites to access.

Host() Vulnerability:

What is the scope of first vulnerability?

Securiteam: [NT] Unsafe Functions in Office Web Components

This vulnerability could enable an attacker to run arbitrary commands on another user's system. By doing so, the attacker would be able to take any action that a user could take, including but not limited to loading and running programs, altering data on the system, reformatting the hard drive, or changing the security settings.

Any limitations on the user's account would also limit the attacker's actions. For example, if a user were prevented from changing the security settings on the system due to group policies, the attacker would be similarly restricted.

What causes the vulnerability?

The vulnerability results because the spreadsheet component in OWC exposes the Host() function.

What is the Host() function?

The Host() function is designed to provide scripts and programs which call it access to the object models of other application. Once the application's object model is available to the caller, it can access the full range of programmatic options that the application makes available. For instance, a script could use the Host() function to gain access to Excel's object model. It could then use that access to invoke any of the programmatic methods or functions that Excel makes available.

What's wrong with the Host() function?

While the OWC is marked as "safe for scripting", the functionality of this function is unsuitable for use by untrusted sites. As a result, it should not be exposed through a component marked "safe for scripting".

What could an attacker do via the vulnerability?

An attacker who was able to exploit this functionality could invoke the host() function in an HTML script and then issue commands against the user's system as if the user had chosen to issue them. The attacker could add, change, or delete any data that the user had access to. In addition, the attacker could change any settings, such as browser security settings, that the user had rights to change.

How might an attacker exploit the vulnerability?

To exploit the vulnerability, an attacker would need to create a web page that invokes the host method, and either host it on a web site or send it to another user as an HTML email. In either case, once the HTML page had been rendered, either in the browser or the email client, the attacker's script would execute and the vulnerabilities would be exploited.

Is there anything that mitigates my risk against the web-based attack?

Yes. A successful attack would require that a user choose to go to the attacker's site; however, the vulnerability provides no way for the attacker to force the user to their site automatically. Because of this, users can protect themselves if they exercise caution in their browsing habits. Specifically, if users avoid unknown and untrusted sites altogether, or place those sites in the Restricted Sites zone, which

Securiteam: [NT] Unsafe Functions in Office Web Components

disables ActiveX controls, users can mitigate and possibly eliminate all exposure to this vulnerability on the Web.

Is there anything that mitigates against the HTML email based attack?

Yes, mail clients that disable ActiveX controls can eliminate the HTML email based attack vector. There are two ways that customers can accomplish this with Microsoft products:

- * Customers can use a mail client that reads email in the Restricted Sites zone. By default, Outlook Express 6.0 and Outlook 2002 open HTML mails in the Restricted Sites Zone. In addition, Outlook 98 and 2000 open HTML mails in the Restricted Sites Zone if the Outlook Email Security Update has been installed. Outlook Express 5.0 will read HTML email in the Restricted Sits zone if it has been so configured.

- * Customers can use a mail client that reads HTML email as plain text. Customers who have enabled the "Read as Plan Text" feature introduced in Outlook 2002 SP1 would thus be protected against attempts to exploit this vulnerability.

LoadText() Vulnerability:

What is the scope of second vulnerability?

This is information disclosure vulnerability. The vulnerability could allow a malicious web site operator to view files on the computer of visiting user.

There a number of significant mitigating factors associated with this vulnerability:

- * It could only be used to read file – not create, change, delete, or execute them.
- * The attacker would need to know the name and location of the file on the user's computer

What causes the vulnerability?

The vulnerability results because a flaw in the spreadsheet component in OWC that exposes the LoadText() method.

What is the LoadText() Method?

The LoadText() method is a programming method that tells a spreadsheet to load and parse a text file into a worksheet.

What's wrong with the LoadText() Method?

By design, the LoadText() method is designed to allow text files to be loaded into spreadsheets. Those text files can be specified on a remote location by passing the function the location of the file. For instance, the LoadText() method can be instructed to retrieve a text file on a shared directory on a network server. When this method is implemented in the OWC, the data is loaded into the web page that hosts the spreadsheet.

It is possible, under the current design, for the web page to specify a file located on the user's own system rather than on a remote file share.

Securiteam: [NT] Unsafe Functions in Office Web Components

When this is done, and the file is read, the data is loaded into the web page.

Because of the nature of browser domain constraints, once this data is loaded into a web page, the hosting site can manipulate that data, including sending it back to the hosting site.

What could this enable an attacker to do?

This could enable an attacker to build an HTML page that uses the LoadText() method and reads files on the victims machine.

How might an attacker seek to exploit this vulnerability?

The attack vectors for this vulnerability are identical to those for the Host() function vulnerability: An attacker could seek to exploit this through the web or through HTML email. Users can protect themselves by taking the same steps as discussed in regards to the Host() function vulnerability.

Copy()/Paste() Vulnerability:

What is the scope of third vulnerability?

This is an information disclosure vulnerability. Specifically, it could enable a web site to programmatically read the contents of a user's clipboard, even when the user has enabled the setting to prevent sites from being able to do this.

The default setting for this option is to allow programmatic access to the clipboard. Therefore, the risk created by this vulnerability is no worse than the default setting for this feature. However, because it allows a security setting that controls the privacy of information to be bypassed, it does constitute a vulnerability.

What causes the vulnerability?

The vulnerability results because the Copy() and Paste() methods in the spreadsheet component in OWC fail to honor the "disabled" setting for Internet Explorer's "Allow paste operations via script".

What are the Copy() and Paste() Methods?

The Copy() and Paste() methods are programmatic equivalents to the act of copying and pasting in Windows. Just like when a user chooses to copy or paste, these commands send data to or remove it from the Windows clipboard.

What can this enable an attacker to do?

This can enable an attacker to perform copy and paste operations via script against the user's Windows clipboard, even when the user had explicitly disabled programmatic access to the clipboard. The attacker could then attempt to manipulate the user's clipboard through HTML scripting.

What kind of data would this give an attacker access to?

This would give an attacker access to whatever data was already present in

Securiteam: [NT] Unsafe Functions in Office Web Components

the Windows clipboard from copy and cut operations that the user had made. Because of the nature of the Windows clipboard and how it has used, this means that the data would be unpredictable and, in most cases, of little interest to an attacker.

Does this vulnerability give an attacker the means to specify data for retrieval?

No. The data must be in the Windows clipboard before the attack was mounted. The vulnerability gives no means for an attacker to specify what data would go into the Windows clipboard.

Where can I learn more about the "Allow paste operations via script" setting?

This feature is documented in Microsoft Knowledge Base Article Q224993.

How might an attacker seek to exploit this vulnerability?

The attack vectors for this vulnerability are identical to those for the Host() function vulnerability: An attacker could seek to exploit this through the web or through HTML email. Users can protect themselves by taking the same steps as discussed in regards to the Host() function vulnerability.

Remediation:

How can I eliminate the vulnerabilities?

The fix for these vulnerabilities is being made available in the following:

- * Office XP SP2
- * A patch for Microsoft Project 2002.
- * A patch for Microsoft Project Server 2002
- * A patch for all other products.
- * An updated version of the Office Web Components Download.

What should I do to eliminate the vulnerabilities?

The recommended way of addressing the vulnerabilities is going to depend on what product you are using.

* Microsoft Office XP: Microsoft Office XP customers should apply Office XP SP2, if at all possible. If they cannot apply SP2, they can apply the general patch. However, SP2 is the preferred means of addressing the vulnerabilities because it includes additional fixes beyond this particular issue. In general, Microsoft always recommends service packs over patches, for this reason.

* Microsoft Project 2002: Microsoft Project 2002 users must apply the Microsoft Project 2002 patch. If you are using Microsoft Project, this is the only way you can address these issues. Microsoft Project 2002 users cannot use the general patch.

* Microsoft Project Server 2002: Microsoft Project Server 2002 customers must apply the Microsoft Project Server 2002 patch, which is the only way you can address these issues. Microsoft Project Server 2002 users cannot

Securiteam: [NT] Unsafe Functions in Office Web Components

use the general patch. Please see Q328044 for more information.

* All Other Products: Customers using all other products can address the vulnerabilities one of two ways:

- * Installing the general patch
- * Installing the updated Office Web Components Download

In most cases, applying the general patch will be the preferred solution. However, both options will fully address the vulnerabilities. Note that customers using Office XP can apply the patch of the updated Office Web Components download. However, they are still urged to consider applying SP2 instead. Note that Microsoft Project 2002 customers must use the Microsoft Project 2002 patch: they cannot use the general patch or the updated Office Web Components download.

I am confused. I am running Microsoft Project 2002, what do I need to do? If you are running Microsoft Project 2002, you should apply the Microsoft Project 2002 patch.

What if I am running Microsoft Project 2002 and one of the other affected applications?

The only way to address Microsoft Project 2002 is to apply the Microsoft Project 2002 patch. This patch, however, only fixes Microsoft Project 2002. If you have any other affected applications on your system, you must address them separately.

For instance, if you have Office XP and Microsoft Project 2002, you should apply the Microsoft Project 2002 patch to fix Microsoft Project 2002. Then, you should apply Office XP SP2 or the general patch to address Office XP.

The net of this is that if you have Microsoft Project 2002, you should apply the Microsoft Project 2002 patch. If you have any other affected products in addition to Microsoft Project, you will have to address them separately.

I am running Microsoft Project Server 2002, what do I need to do? If you are running Microsoft Project Server 2002, you should apply the patch for Microsoft Project Server 2002.

Is that the same as applying the patch for Microsoft Project 2002? No. The patch for Microsoft Project Server 2002 is different from the patch for Microsoft Project 2002. The Microsoft Project Server patch is a server-side patch that need be applied only to the Microsoft Project Server. The Microsoft Project 2002 patch is a client-side patch that should be applied to all systems running Microsoft Project 2002.

I am running Office XP, what do I need to do? If you are running Office XP, you should apply Office XP SP2 if at all possible. In addition to addressing these issues, it includes many other important security and stability fixes.

Securiteam: [NT] Unsafe Functions in Office Web Components

If it is not possible to apply Office XP SP2, then you should apply the general patch or the updated Office Web Components Download.

I have installed OWC from the Web, what do I need to do?

If you have installed OWC from the web, you should use the general patch on your currently installed systems.

I am not running Microsoft Project 2002 or Office XP and I did not install OWC from the web, what I should do?

Customers using any of the other products listed should apply the general patch to address the vulnerabilities.

I want to go ahead and set the killbit for these controls, can I do that?

Yes, you can manually set the killbit for these controls. However, it is critical to keep in mind that doing so could cause the any of applications that use OWC to behave unexpectedly or fail to run entirely.

That said, if you wish to set the killbit for the OWC, follow the instructions contained in Q240797 and enter the following CLSID's:

OWC 2000:

* CLSID: {0002E530-0000-0000-C000-000000000046}

ProgID: OWC.DataSourceControl.9

* CLSID: {0002E510-0000-0000-C000-000000000046}

ProgID: OWC.Spreadsheet.9

* CLSID: {0002E500-0000-0000-C000-000000000046}

ProgID: OWC.Chart.9

* CLSID: {0002E520-0000-0000-C000-000000000046}

ProgID: OWC.PivotTable.9

* CLSID: {0002E531-0000-0000-C000-000000000046}

ProgID: OWC.RecordNavigationControl.9

OWC 2002:

* CLSID: {0002E553-0000-0000-C000-000000000046}

ProgID: OWC10.DataSourceControl.10

* CLSID: {0002E551-0000-0000-C000-000000000046}

ProgID: OWC10.Spreadsheet.10

* CLSID: {0002E556-0000-0000-C000-000000000046}

ProgID: OWC10.ChartSpace.10

* CLSID: {0002E552-0000-0000-C000-000000000046}

ProgID: OWC10.PivotTable.10

* CLSID: {0002E554-0000-0000-C000-000000000046}

ProgID: OWC10.RecordNavigationControl.10

Securiteam: [NT] Unsafe Functions in Office Web Components

ADDITIONAL INFORMATION

The information has been provided by

<mailto:0_35529_E51E4D7D-DECD-43AE-9A29-36080E8D4C3C_US@Newsletters.Microsoft.com>
Microsoft Product Security.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- **Previous message:** support@securiteam.com: "[NT] Accessing Remote and Local Content in IE"
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)