

MWR InfoSecurity Advisory

Interwoven Worksite –
ActiveX Control Remote
Code Execution

10th March 2008

MWR  INFOSECURITY

Contents

| | | |
|----------|-------------------------------------------------|----------|
| 1 | Detailed Vulnerability Description | 5 |
| 1.1 | Introduction | 5 |
| 1.2 | Technical Background | 5 |
| 1.3 | Vulnerability Details | 5 |
| 1.4 | Vulnerability Impact | 6 |
| 1.5 | Exploit Information | 6 |
| 1.6 | Dependencies | 7 |
| 2 | Recommendations | 8 |
| 3 | Further Information..... | 8 |

Interwoven WorkSite – ActiveX Control Remote Code Execution

| | |
|---------------------------|----------------------------------------------------------------------------------------------------|
| Package Name: | Interwoven Worksite |
| Date: | 7 th November 2007 |
| Affected Versions: | Web TransferCtrl Class 8,2,1,4 (within iManFile.cab) CLSID:4BECECDE-E494-4f69-A3DE-DA0B77726307 |

| | |
|---------------------------------|------------------------------------------------------------------------------|
| CVE Reference | CVE-2008-1617 |
| Author | J Fitzpatrick |
| Severity | High Risk |
| Local/Remote | Remote |
| Vulnerability Class | Remote Code execution |
| Vendor Response | Interwoven have addressed this in their latest service pack. |
| Exploit Details Included | Yes (although exploit code has been omitted) |
| Versions Affected | Web TransferCtrl Class 8,2,1,4 CLSID:4BECECDE-E494-4f69-A3DE-DA0B77726307 |
| Affected OS | All supported Operating Systems |

Timeline:

| | |
|-----------------------------------------|------------|
| Vulnerability Reported to vendor | 2007-11-07 |
| Vendor Patch Released | 2008-01-29 |
| Advisory Released | 2008-03-10 |

Overview:

Worksite is a document management and email management solution from Interwoven Inc (Interwoven). Some of the functionality of the application is made available through ActiveX controls which are distributed within the iManFile.cab file. The ActiveX controls were found to be unsafe and permit code to be executed remotely by an attacker who is able to direct a user to a website containing exploit code.

Impact:

The most serious of these vulnerabilities could enable an attacker to execute arbitrary code on a user's computer remotely. This code would be executed with the permissions of the user logged into the system. However, other vulnerabilities are present. For more information refer to the Additional Vulnerability information section.

Cause:



The remote code execution arises due to the ActiveX control incorrectly maintaining a reference to a JavaScript variable. This results in a double free, which can be exploited in order to execute arbitrary code.

Solution:

Interwoven have incorporated a fix for these issues into their latest service pack. Further information can be found in section 2.

Additional Vulnerability information:

A DOS vulnerability was also been identified within the control with the CLSID as follows:

`C209367E-F2F1-497f-B990-08761195DAF1`

This was found to be vulnerable to a DOS through resource consumption. The SendNrLink opens an email composition window which can easily be called multiple times resulting in all available resources being consumed.

1 Detailed Vulnerability Description

1.1 Introduction

"Interwoven WorkSite provides a platform that supports geographically dispersed teams with project pages where they can create content, collaborate, and coordinate business-critical activities."

Source: <http://www.interwoven.com/components/page.jsp?topic=WORKSITE::PROFSERV>

Worksite provides web capabilities, integrates with Microsoft Office applications, and provides email management and other facilities providing a comprehensive collaborative platform. The vulnerabilities identified within this report were identified through use of the web interface only.

1.2 Technical Background

Some aspects of WorkSite require an ActiveX control to be installed in the user's web browser. On using the applications web capabilities the user is prompted to install a control which increases functionality. This is distributed within the file iManFile.cab.

The CSLIDs of the vulnerable control is listed here:

4BECECDE-E494-4f69-A3DE-DA0B77726307

Against Microsoft's guidance on designing secure ActiveX controls¹ these have been registered as safe for scripting and so are accessible by any website visited by a user with the control installed. These vulnerabilities could therefore be exploited by any website posing a very high risk to any users of the WorkSite application.

1.3 Vulnerability Details

This control incorrectly uses a JavaScript variable rather than creating its own copy which later results in a double free which could be exploited.

The vulnerability was found to be present in the ActiveX objects Server method which can be called with the following JavaScript:

```
<object id='target' classid="CLSID:4BECECDE-E494-4f69-A3DE-DA0B77726307"></object>
<script>
  var obj = document.getElementById('target');
  obj.Server = "value";
</script>
```

In order to exploit this vulnerability heap spraying could be used in order to gain some control over the heap into which shellcode can then be placed. The unlink operation which

¹ <http://msdn2.microsoft.com/en-us/library/aa752035.aspx>

occurs on a call to 'free()' could then be exploited in order to gain a four byte write and used to overwrite the return address directing the path of execution to the shellcode.

1.4 Vulnerability Impact

The successful exploitation of this vulnerability would result in remote code execution or a denial of service condition for the user. In the worst case an attacker could exploit this vulnerability in order to take full control of a user's computer through exploiting this vulnerability.

1.5 Exploit Information

The exploit developed by MWR InfoSecurity uses heap spraying in order to gain some control over the heap and then manipulates the unlinking operation which occurs when memory is freed in order to direct the path of execution to our shellcode.

By filling up the heap with data we eventually reach some valid memory where our shellcode can sit and execute. We perform this action in the JavaScript below by filling the heap with data and then filling the higher memory with our shellcode and a nop sled in order to increase chances of hitting the shellcode. A sample of the code to do this is given here:-

```
var ar = new Array();
function spray() {
  while(data.length<100000) data+=data;
  for(var x=0; x<512; x++) ar.push(data+"x")
  CollectGarbage();
  while(nop.length<99900) nop+=nop;
  for(var x=0; x<150; x++) ar.push(nop+shellcode)
  CollectGarbage();
}
spray();
```

The next step is to actually exploit the double free. To do this we need to allocate the memory on the heap. A cache of freed memory is maintained and allocations will come from here in the first instance. In order to maintain control over the memory allocations it is important that the allocations are made on the heap and not from the cache of freed memory. A technique for achieving this can be found in Alexander Sotirov's paper entitled "Heap Feng Shui in JavaScript". Also it is important that we don't simply allocate a new string literal as this does not create a copy of the string. Concatenation ensures that strings are allocated on the heap:-

```
var str1=str+s;
```

The affected method within this ActiveX control is the 'Server' method.

```
str1=str+s;
obj.Server=str1;
str1=obj.Server;
str1=0;
```

We set this to one of the strings and obj.Server now maintains a reference to some memory containing our data. By then assigning 'str1' the value 0 the location in memory holding the data str1 originally referenced becomes freed however obj.Server still maintains its reference to this memory.

In order to exploit the double free we first create two strings allocated side by side on the heap and give the ActiveX control a reference to them through the 'Server' method. Next we free these objects from memory; however, the ActiveX control continues to maintain a reference to these now free memory locations. Next we need to write our own data to these memory locations which have just been freed but are still referenced by the ActiveX object. By making a new heap allocation we can get an allocation at these addresses and so place data in memory on top of the previous first allocation and such that it overwrites the control structure of the second allocation. Now when the ActiveX object loses its reference to the already freed string it gets freed a second time. However, in this second free the data in the control structure for this memory has been overwritten allowing us to manipulate the unlinking process and obtain a 4 byte write. Using this 4 byte write we can direct the path of execution towards our shell code.

Using this technique MWR InfoSecurity have constructed working proof of concept code for this vulnerability. However, the code will not be released into the public domain at present. The decision to release such code in the future will be taken based on MWR InfoSecurity's obligations to protect its customers and Critical National Infrastructure (CNI) whilst also enabling the security community to accurately assess the vulnerability of systems running the software.

1.6 Dependencies

In order for an attacker to exploit these vulnerabilities they must direct a user to a website containing malicious code, this could be done through a phishing link, XSS or other means.

2 Recommendations

Interwoven have addressed this issue in their latest Service Pack:

WorkSite Web 8.2 SP1 P2

This is available through the Interwoven support site located at:

<http://worksitesupport.interwoven.com>

Additional Information

It should be noted that this vulnerability affects an ActiveX control which is installed on an end user's PC when using the Worksite application rather than the servers hosting the application. As a result, any system with this control installed is vulnerable whether or not WorkSite is being used at the time. It is therefore important that the appropriate vendor supplied patches are applied to all PCs which may have this control installed.

If access to WorkSite support is not available in order to apply these updates then it is strongly recommended that this control is removed. On a test system running windows XP with Internet Explorer 6.x. the affected control was located in the "C:\WINDOWS\Downloaded Program Files" directory with the CLSID {9F51E426-6EED-11D3-80B8-00C04F610DBB}. It was possible to remove the control right clicking and selecting remove.

3 Further Information

Further information on removing ActiveX controls can be found at the following location:-
<http://support.microsoft.com/kb/154850>

Further information on designing secure ActiveX controls can be found at the following location:-
<http://msdn2.microsoft.com/en-us/library/aa752035.aspx>

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