## 一、漏洞信息

### 1. 漏洞简述

* 漏洞名称：MS06-055
* 漏洞编号：CVE-2006-4868
* 漏洞类型：栈溢出漏洞
* 漏洞影响：远程代码执行
* CVSS评分：（不知道）
* 利用难度：Medium
* 基础权限：不需要

### 2. 组件概述

此漏洞是IE浏览器在解析VML标记语言时存在基于栈的缓冲区溢出漏洞。矢量标记语言：IE从5.0在html中支持这种语言。在web应用中假如需要绘制的图形较为简单，就可以使用VML语言，用文本方式告诉客户端一些关键坐标的绘图坐标，汇览器按照VML语言的标识解析这些坐标后就能绘制出精确的图形，使用此方法可以用简单文本代替图形像素点的传输，以达到减小网络负载的目的。

### 3. 漏洞利用

此漏洞属于远程代码执行漏洞，在靶机进行漏洞利用网页的访问时，通过覆盖返回地址为0x0c0c0c0c，同时借助JavaScript申请大量内存，借助Heap Spray技术运行恶意代码，达到控制目标机器的目的

### 4. 漏洞影响

此漏洞在Windows 2000 sp4；Windows XP sp1；Windows XP xp2；Windows Server 2003上严重影响，在Windows Server 2003 sp1上为重要影响。

### 5. 解决方案

下载安装官方给出的补丁（KB925486）。

## 二、漏洞复现

### 1. 环境搭建

* 靶机环境

操作系统:Windows xp sp1 简体中文。

* 靶机配置：

无

* 攻击机环境

操作系统：kali linux 2021

* 攻击机配置

使用/etc/init.d/apache2 start运行apache2服务器

### 2. 复现过程

使攻击机与靶机处于同一局域网中，并将恶意html文件（index.html）置于攻击机/var/www/html文件夹中，使用ifconfig查询攻击机IP地址，使用靶机浏览器访问攻击机IP地址，在攻击机的控制台下输入telnet 靶机IP 6666进行正向连接，即可对靶机进行远程控制。

## 三、漏洞分析

### 1. 基本信息

* 漏洞文件：vgx.dll
* 漏洞函数：：\_IE5\_SHADETYPE\_TEXT::Text(unsigned short const \*, int)

### 2. 背景知识

Heap Spray 技术：在以往的程序进行缓冲区溢出攻击时，一般将返回地址置于堆栈中，之所以这样做是因为我们很难控制堆栈以外的空间，并将恶意代码置于指定位置，而在网页的的缓冲区溢出攻击中，我们可以借助JavaScript申请大量空间，并将申请的空间用0x90覆盖，只要申请的空间够大就可以覆盖0x0c0c0c0c地址处，这样就可以将溢出的返回地址用0x0c0c0c0c替代在函数返回时就可以跳转到0x0c0c0c0c处执行，只要在申请空间的末尾用恶意代码覆盖就可以执行恶意代码。

### 3. 详细分析

#### 1. 基础分析

本例中使用的是window xp sp简体中文版本的vgx.dll文件，大小为802304字节，位于C:\Program Files\Common Files\Microsoft Shared\VGX文件夹中，产生漏洞的函数为\_IE5\_SHADETYPE\_TEXT::Text，此函数有2个参数：

{

unsigned short const \*；

int

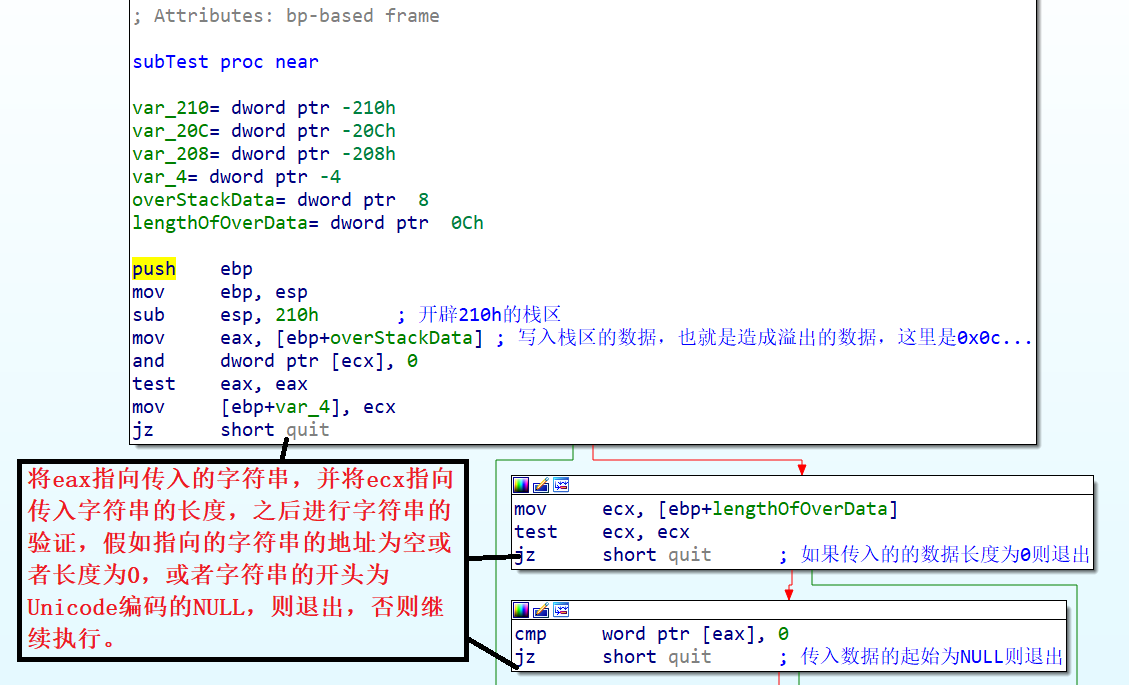
}

第一个参数指向字符串，此字符串将复制到栈区，第二个参数是此字符串的长度。

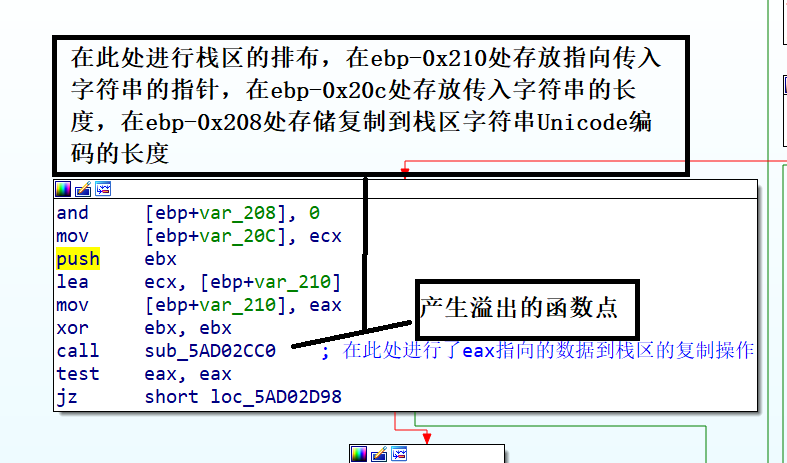
**（注意此函数在使用ida pro打开时并没有，需要在微软官网下载对应版本的符号文件）**

#### 2. 静态分析

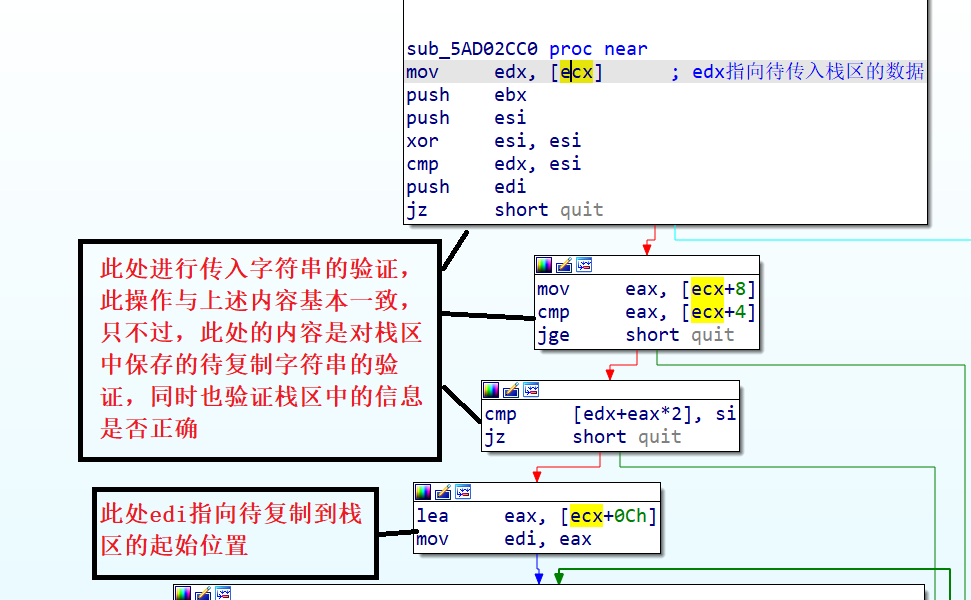
将存在漏洞的vgx.dll文件拖入IDA pro 32位版本中，定位到漏洞函数\_IE5\_SHADETYPE\_TEXT::Text的位置，此函数位于0x5AD02D1B处，并跳转到此函数的汇编代码中，进行分析。



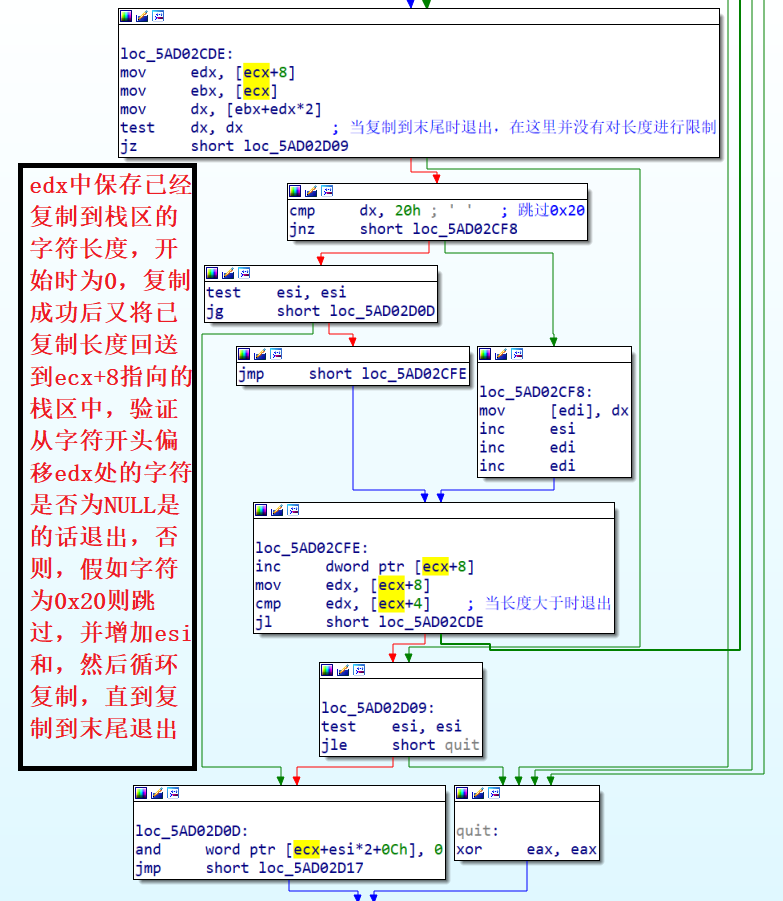
**图3.2.1 传入字符串的验证**

****

**图3.2.2 栈区内容的排布**

****

**图3.2.3 栈区内容验证**

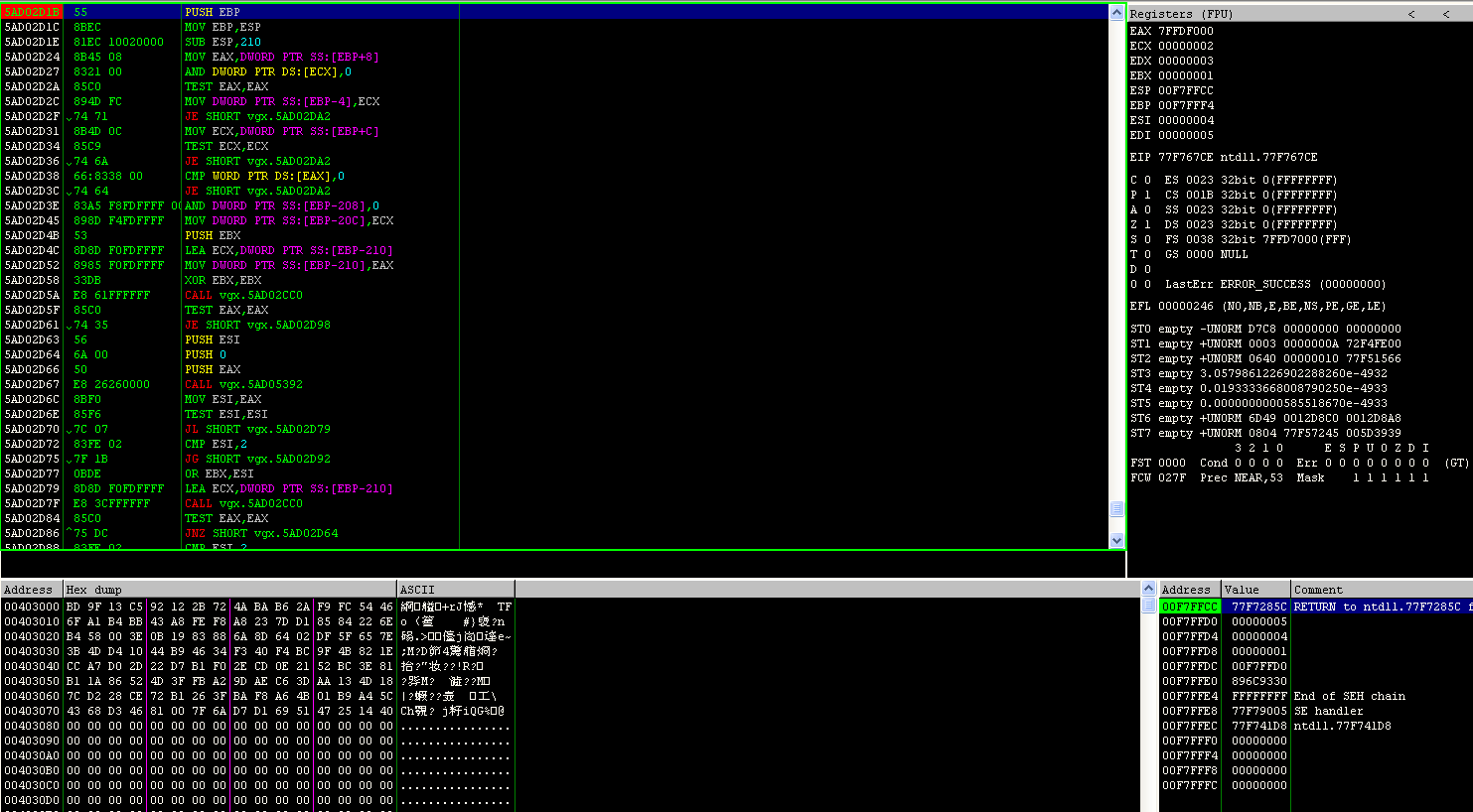
****

**图3.2.4 传入字符串复制到栈区**

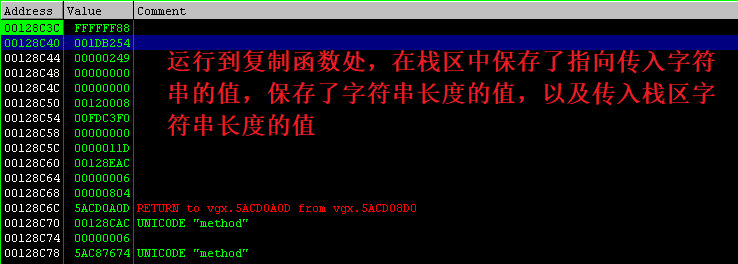
由于上述复制操作并未对长度进行验证，所以会产生溢出，并覆盖返回地址为0x0c0c0c0c。至此缓冲区的溢出已经完成。

#### 3. 动态分析

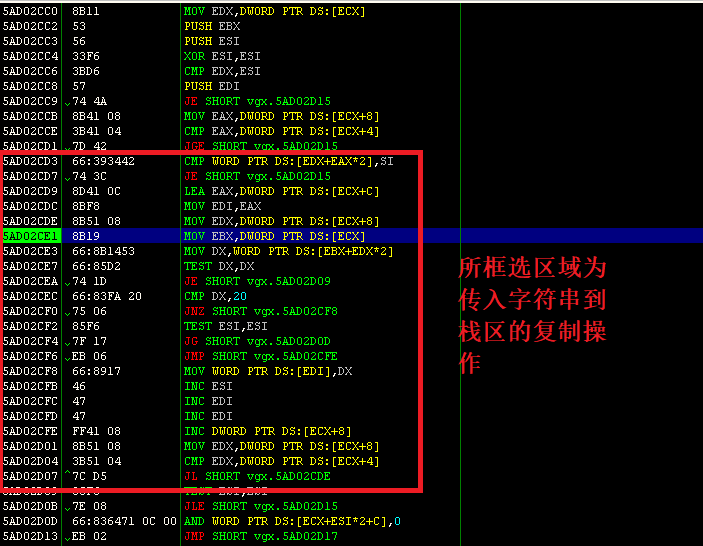
在靶机的IE浏览器中访问攻击机ip，然后使用Ollydbg attach到IE浏览器，使用ctrl+G定位到漏洞点0x5AD02D1B，并下断点。刷新页面，会在断点处停下。



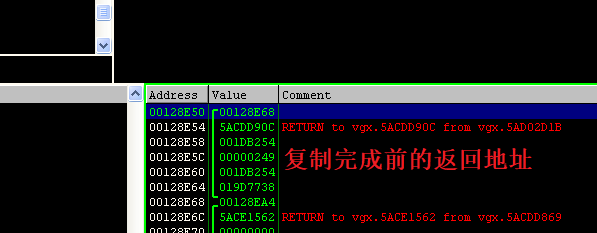
**图3.3.1 下断点处**

****

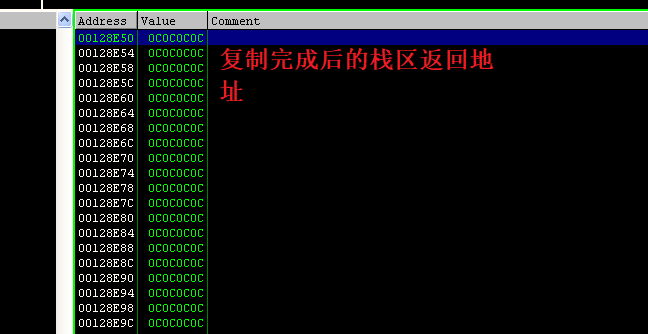
**图3.3.2 复制前栈区中的内容**

****

**图3.3.3 复制操做代码**

****

**图3.3.4 返回地址**

****

**图3.3.5 复制完成后的栈区返回地址**

至此一至运行到函数的返回，便可跳转0x0c0c0c0c执行，至此动态分析结束。

## 四、参考文献

0day安全：软件漏洞分析