漏洞挖掘技术

————基于FileFuzz工具

张晓力
T201089950

Fuzz由来

- Fuzz是一种软件安全漏洞挖掘技术
- Fuzz技术的思想就是利用“暴力”来实现对目标程序的自动化测试，然后见时间差其最后的结果，如果符合某种情况就认为程序可能存在漏洞或者问题，这里的暴力指的是利用不断地向目标程序发送或者传递不同格式的数据来测试目标程序的反应
- Fuzz技术属于典型的黑盒测设手段，进行Fuzz测试时，只要求对于目标程序有个大概的认识，就可以利用Fuzz技术来挖掘系统中可能存在的安全漏洞

Fuzzer类型

- 依据Fuzz原理开发出的安全测试程序被称之为Fuzzer
- Fuzzer按工作范围的不同分为

  文件类型Fuzzer

  专门用来挖掘文件处理型软件安全漏洞

  网络类型Fuzzer

  专门用来挖掘网络工作软件安全漏洞

  内存类型Fuzzer

  专门用来挖掘软件内部安全漏洞

FileFuzz介绍

- Filefuzz是一种针对文件处理软件进行Fuzz安全测试的工具软件
- 由于处理文件类型的分类，文件处理软件被区分为：

  图文处理软件——Microoffice Office

  媒体处理软件——音乐视频播放软件

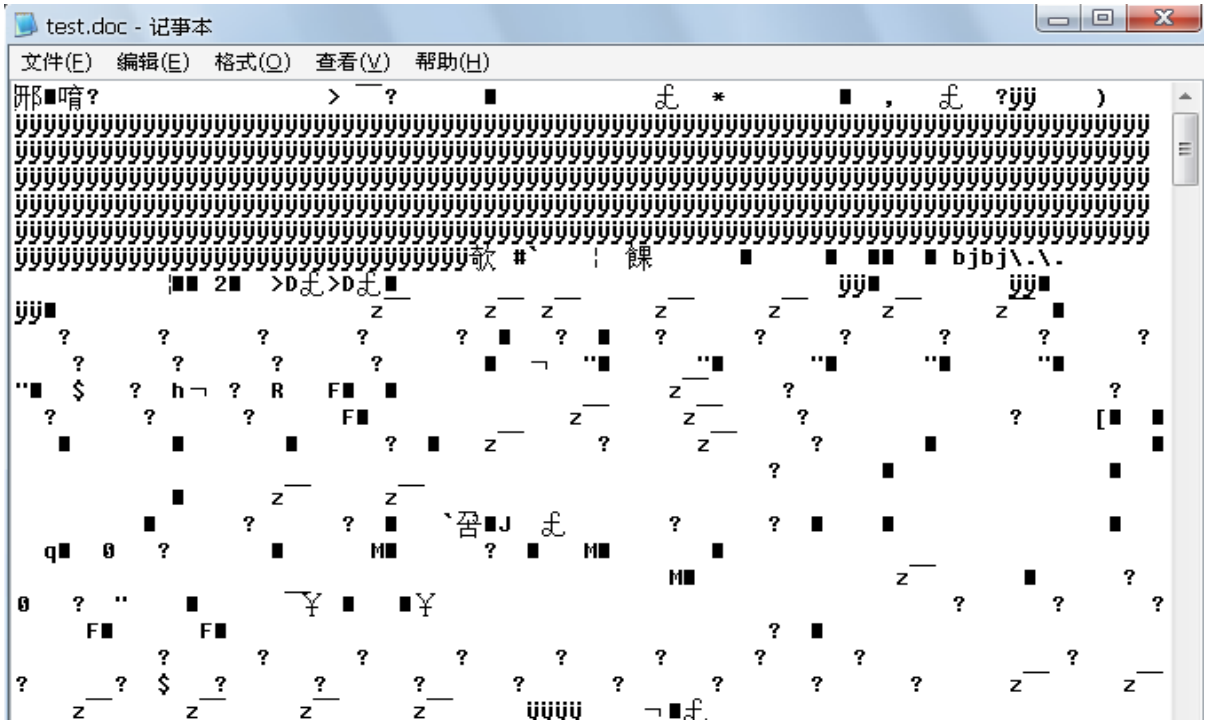
  其它文件处理型软件——专用文件处理软件如WinRAR

  无论哪种类型的文件处理软件，我们都将软件处理的文件类型称为文档文件，既是我们待处理的File

  例如：word程序处理得文档文件就是doc类型的文件，doc文档中包含各种各样的文字或者图片信息

FileFuzz原理

- 文档文件在保存数据信息时，往往采用特殊的编码格式，这些编码格式一般都采用非文本形式，即是说文档保存的数据不是明文格式，以doc文档文件为例，我们用记事本打开一个doc文档，会发现全部是乱码



FileFuzz原理

- 这种非明文形式编码格式目的确保用户只有利用与文档文件相匹配的文件处理软件，才能正确读取文字或图片信息，防止数据信息被任意泄露
- 这种非明文形式的编码意味只有文件处理的开发者知道如何解释这些编码格式，而作为漏洞挖掘者无从知晓，而文件处理软件的漏洞往往发生在软件处理文档文件过程中，例如文档文件中某个地方数据过长就可能造成软件发生溢出漏洞
- 由于我们不知道文档文件的具体编码格式，如果冒然修改就会破坏文档文件编码格式，文档处理软件可能发现这种错误的编码文档文件，从而拒绝打开，我们也就无法测试软件处理该文档有无安全漏洞

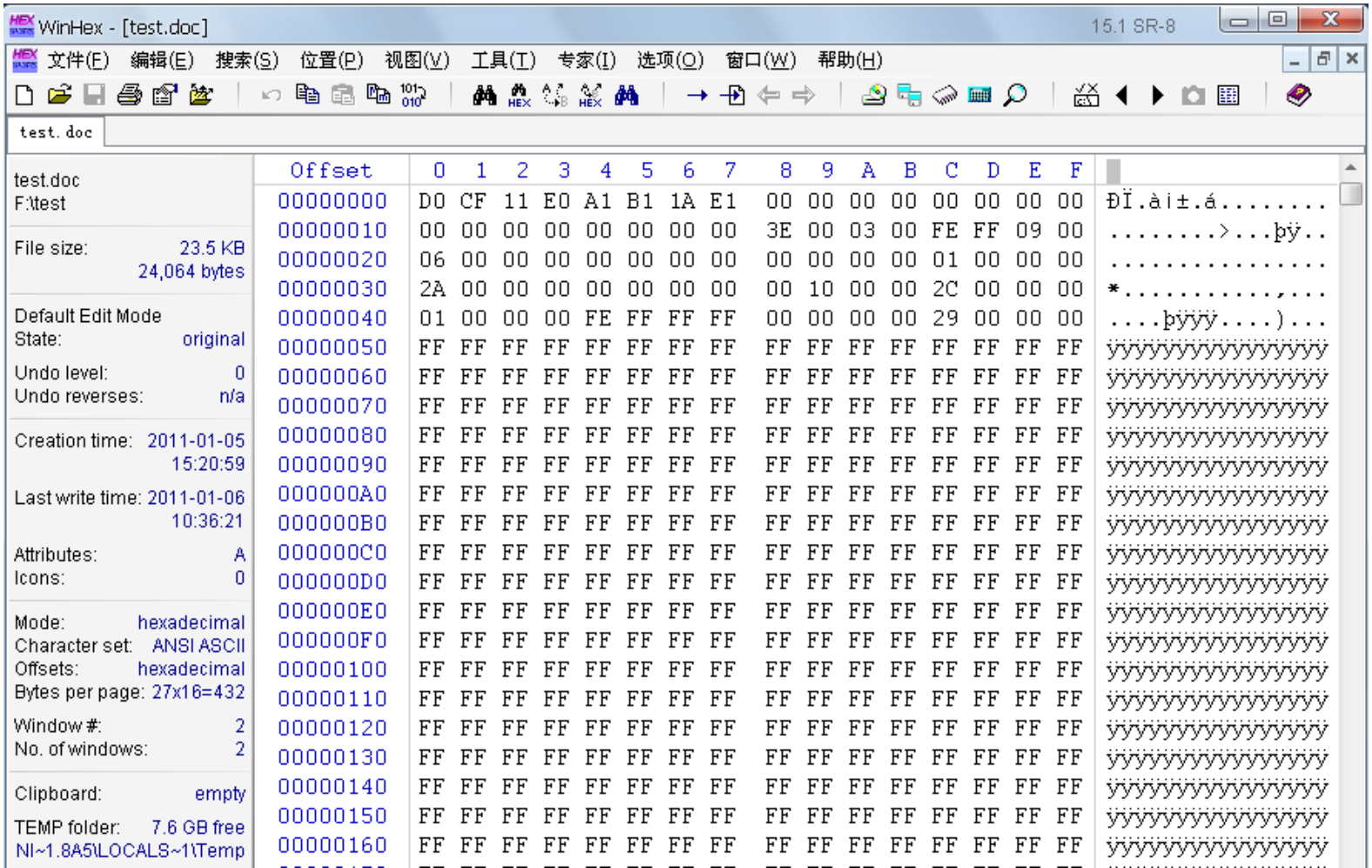
FileFuzz原理

- 虽然我们不知道文档文件的编码格式，但是修改文档文件中的某些数据是不会造成整个文档文件不能被软件打开，一旦软件能够打开被修改的文档文件，那么漏洞就有可能会被文档文件触发到
- 一般文档文件采用非明文形式编码，我们无法通过记事本这样的程序来修改，但是利用十六进制编辑软件，我们依旧可以修改这些非明文形式的文档文件

  例如：WinHex这款十六进制编辑软件

FileFuzz原理

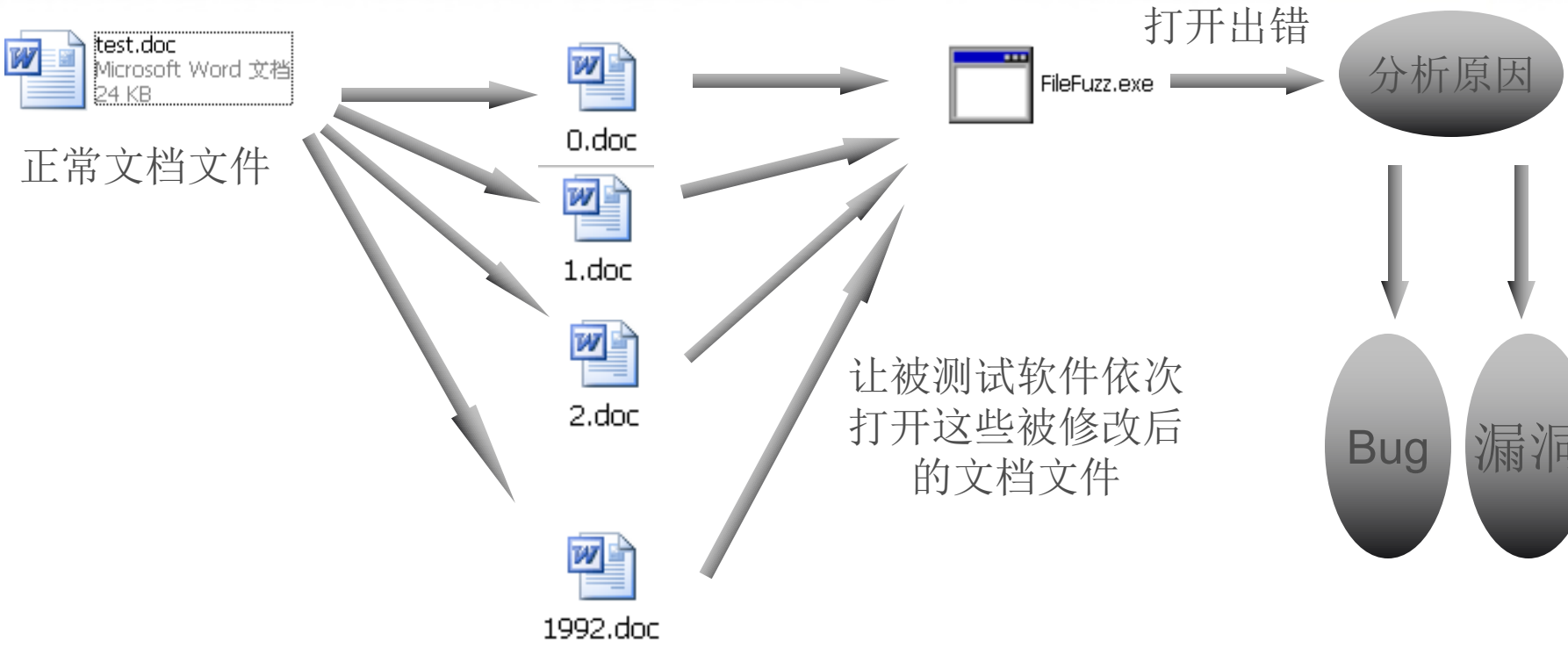
- 用winHex打开doc文档文件



FileFuzz原理

- 我们可以利用WinHex来修改非明文的文档文件
- 由于我们不知道编码的格式，因而不知道那些数据被修改可以触发安全漏洞，难道要我们一个一个字节的修改测试？这样太繁琐
- FileFuzz就是一款文档处理软件暴力化自动测试工具
- FileFuzz采用字节替换法批量生成待测试文档文件，软后将这些待测试文档文件逐一调用相对应文件处理软件打开，同时监视打开过程中发生的错误，并将错误结果记录下来，一边安全研究人员分析该错误是不是属于安全漏洞

FileFuzz原理

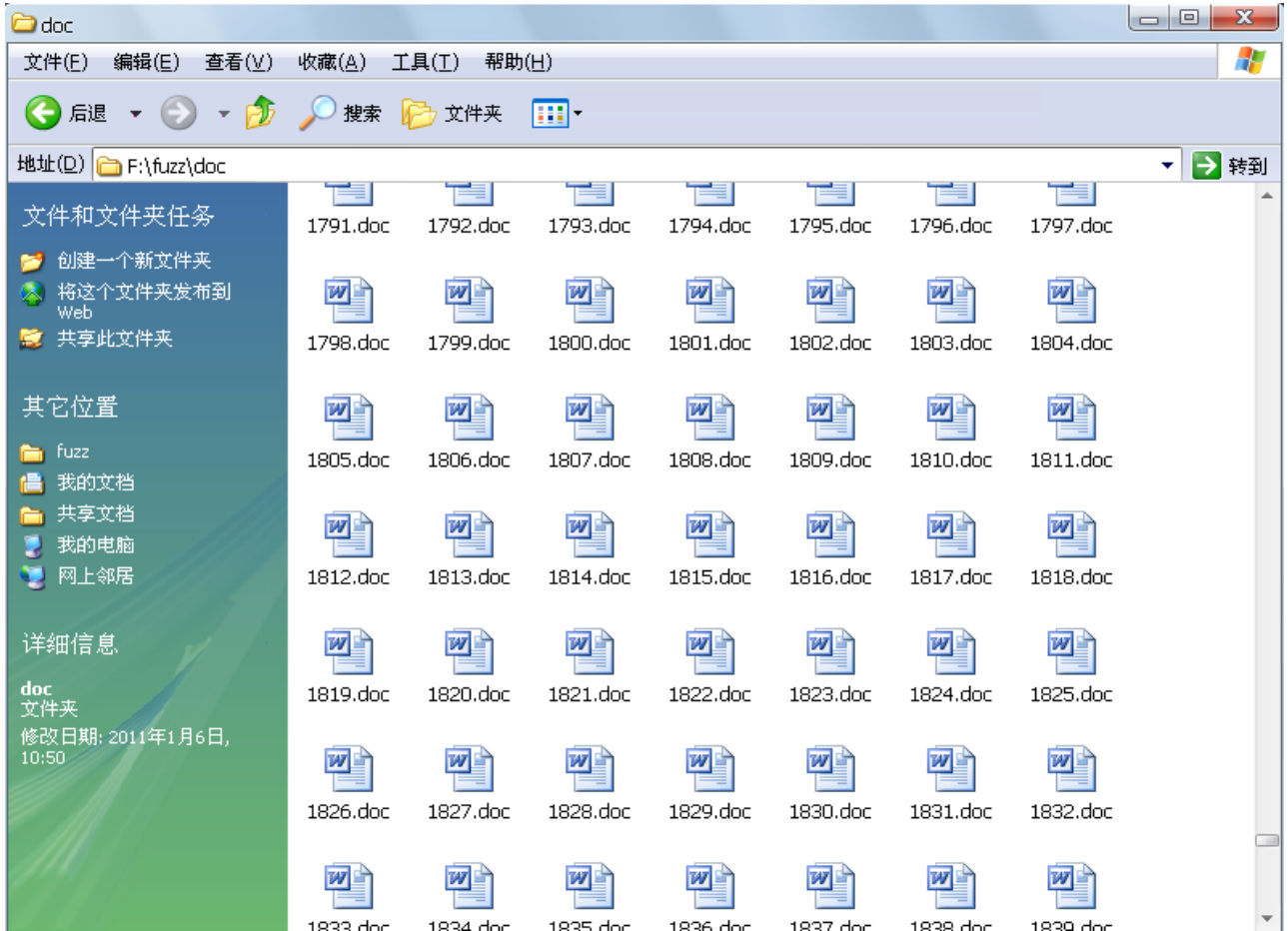

test.doc
Microsoft Word 文档
24 KB
正常文档文件
0.doc
1.doc
2.doc
1992.doc
被修改后的文档文件
让被测试软件依次
打开这些被修改后
的文档文件
FileFuzz.exe
打开出错
分析原因
Bug
漏洞

FileFuzz使用

```
<fuzz xmlns="http://www.idefense.com/filefuzz.xsd">
    <test>
                <name>doc - winword.exe</name> //用winhex打开逐字修改doc
                <file>
                <fileName>DOC</fileName>
        <fileDescription>Microsoft Office Word Document</fileDescription> //测试目标软件
                </file>
                <source>
                        <sourceFile>test.doc</sourceFile> //待测文件
                        <sourceDir>F:\test\</sourceDir> //待测文件地址
                </source>
                <app>
                        <appName>WINWORD.EXE</appName>
                        <appDescription>Microsoft Office Word</appDescription>
                        <appAction>open</appAction>
                        <appLaunch>"C:\Program Files\Microsoft
Office\OFFICE11\winword.exe"</appLaunch>
                        <appFlags>"{0}"</appFlags>
                </app>
                <target>
                        <targetDir>F:\fuzz\doc\</targetDir> //生成修改文件地址
                </target>
    </test>
</fuzz>
```

FileFuzz使用

- 正确配置好FileFuzz后，就可以生成待测文件了

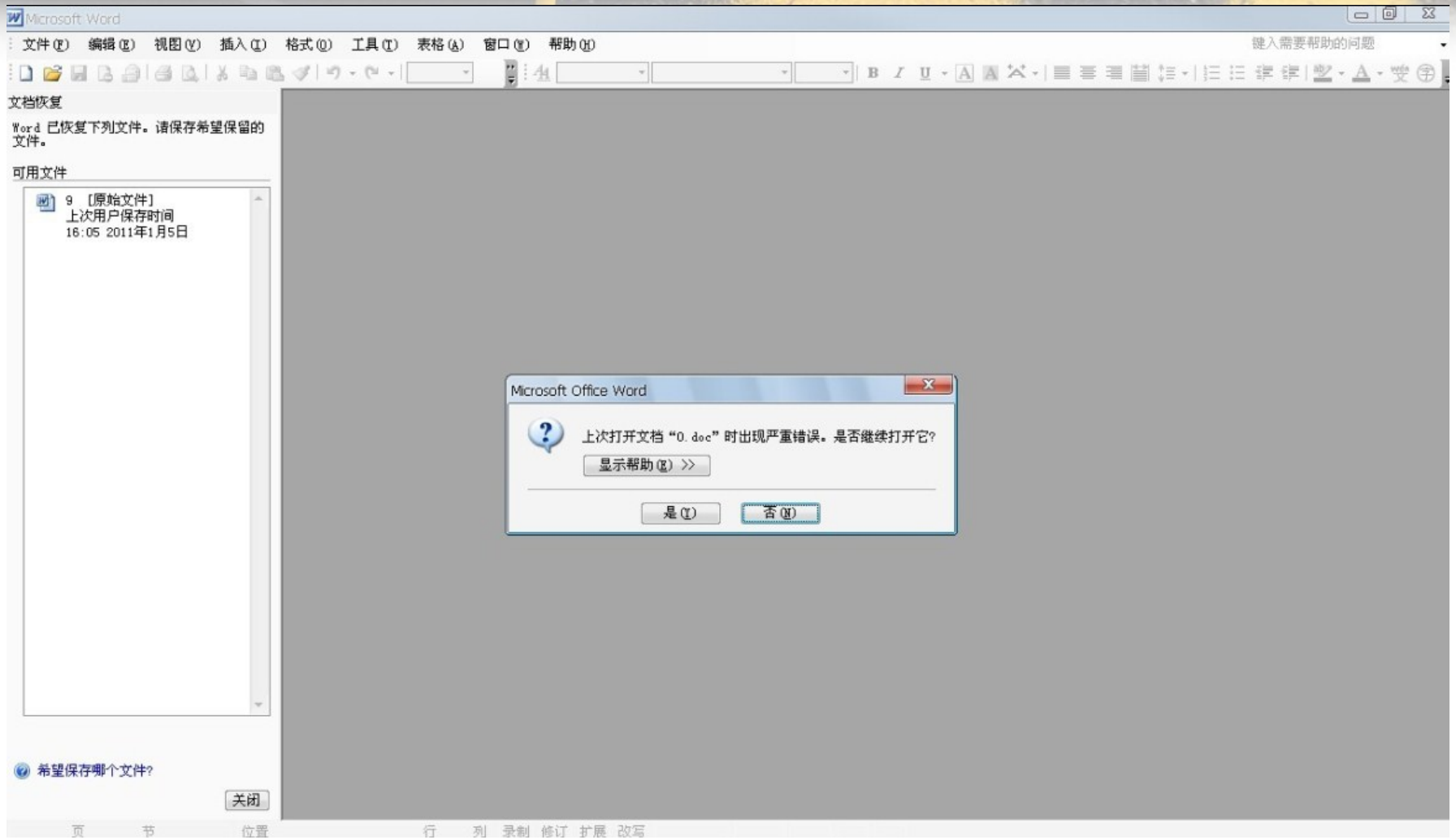


FileFuzz使用

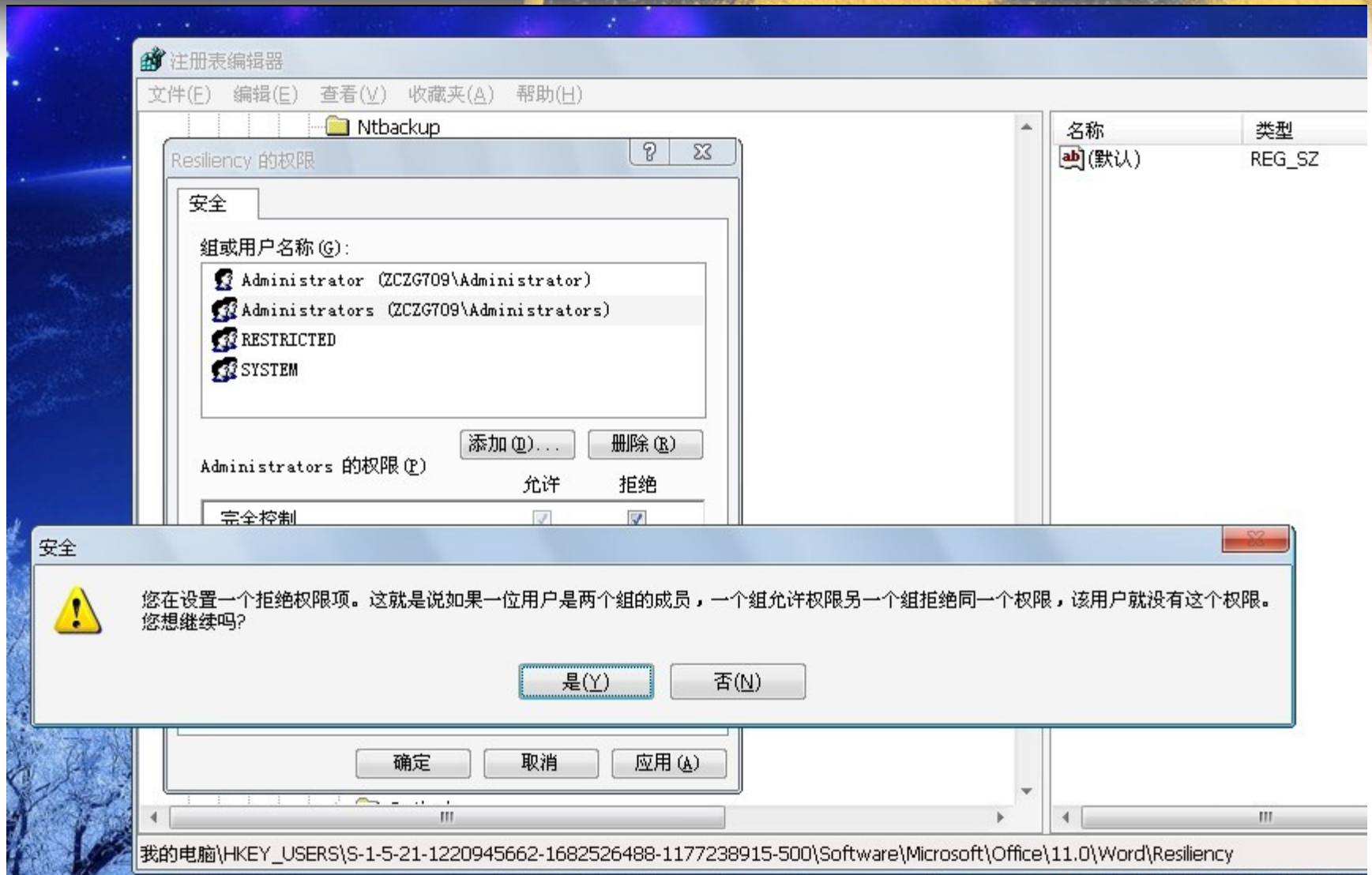
- 生成待测文件后，我们可以通过FileFuzz来自动测试所有修改的文档文件
- 但是Word程序自带有安全模式保护功能

- 若是每次手动选择，那会大大延误我们的测试速度，所以我们修改了Word程序的权限阻止安全模式的启动

FileFuzz使用


Microsoft Word
文件(F) 编辑(E) 视图(V) 插入(I) 格式(O) 工具(T) 表格(A) 窗口(W) 帮助(H)
键入需要帮助的问题
文档恢复
Word 已恢复下列文件。请保存希望保留的文件。
可用文件
9 [原始文件]
上次用户保存时间
16:05 2011年1月5日
希望保存哪个文件?
关闭
Microsoft Office Word
上次打开文档 "0.doc" 时出现严重错误。是否继续打开它?
显示帮助(E) >>
是(Y)
否(N)
页 节 位置 行 列 录制 修订 扩展 改写

FileFuzz使用


注册表编辑器
文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)
Ntbackup
Resiliency 的权限
安全
组或用户名称(G):
Administrator (ZCZG709\Administrator)
Administrators (ZCZG709\Administrators)
RESTRICTED
SYSTEM
添加(D)... 删除(R)
Administrators 的权限(P)
允许 拒绝
完全控制
确定 取消 应用(A)
名称 类型
(默认) REG_SZ
安全
您在设置一个拒绝权限项。这就是说如果一位用户是两个组的成员，一个组允许权限另一个组拒绝同一个权限，该用户就没有这个权限。
您想继续吗?
是(Y) 否(N)
我的电脑\HKEY_USERS\S-1-5-21-1220945662-1682526488-1177238915-500\Software\Microsoft\Office\11.0\Word\Resiliency

FileFuzz使用

- 修改完成以后我们就可以自动测试修改的文档文件来挖掘漏洞了

自动测试分析

- [*]“crash.exe” “C:\program files\mircrosoft office\office11\winword.exe”5000“F:\fuzz\doc\1586.doc”
- [*]Access Violation //非法访问内存地址
- [*]Exception caught at 301bd449 call[ecx+14] //在地址301bd449处调用指令 [ecx+14]发生异常
- [*]EAX：0ac50330 EBX：00000003 ECX：00e00003 EDX：00000000
- [*]ESI：00b00290 EDI：00000000 ESP:0012b89c EBP:ffffffff

自动测试分析

- 这个测试结果反映word2003在打开1586.doc测试文档时发生严重错误，错误指令是call[ecx+14]
- 这个错误让我们想到如果我们能够控制ecx寄存器让其指向一个充满shellcode的内存地址，那么就可以控制word2003在打开doc文档时运行我们的shellcode代码，从而实现在用户系统上运行木马病毒程序目的，用户一旦打开doc文档就会遭受恶意攻击
- 我们通过ollydbg发现ecx寄存器来自指令：mov ecx，dword ptr[eax],而eax指向的是doc文档本身，于是我们可以将doc文档相应的位置修改成一个有shellcode代码的位置，最终借助eax将这个数值传递给ecx，当

自动测试分析

- Cpu执行到call[ecx+14] 指令时，就会调用shellcode代码的内存地址，从而运行我们的shellcode
- 其实这个就是MS06-027漏洞

MS06-027漏洞

- MS06-027漏洞可以利用一个普通的word文档来实现借助Word程序执行任意代码，所以就有人开始利用doc文档中加入恶意木马病毒程序，然后将doc文档发送给受害者，受害者只要打开该文档就会感染木马病毒，甚至还有专门的木马捆绑工具：浩天Word2003捆绑器，实现病毒与doc文档的捆绑