缓冲区溢出漏洞挖掘分析及利用的研究

史飞悦 傅德胜
(南京信息工程大学 南京 210044)

摘 要 当前,软件安全漏洞问题日趋严重,缓冲区溢出漏洞仍然是影响当前网络与分布式系统安全的主要问题之一。对缓冲区溢出漏洞进行挖掘分析以及利用的研究对于系统软件安全有着重要的意义。首先对缓冲区溢出原理以及漏洞挖掘分析与利用技术进行分析总结。然后提出了一种动静态分析相结合的漏洞挖掘分析方法,并采用此方法对微软Office漏洞进行挖掘分析,呈现了一个完整的漏洞挖掘分析过程。最后在理论与技术的基础上,在Windows平台下实现了漏洞挖掘分析系统VulAs,用于辅助漏洞挖掘分析工作,并验证了系统的准确性与有效性。

关键词 漏洞,缓冲区溢出,漏洞挖掘分析,Shellcode,Vulas
中图法分类号 TP393 **文献标识码** A

Research of Buffer Overflow Vulnerability Discovering Analysis and Exploiting

SHI Fei-yue FU De-sheng
(Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract Currently, the problem of software security vulnerability becomes worse, and buffer overflow vulnerability still affects the current network and distributed system security. So it is very important to research the buffer overflow vulnerability discovering analysis and exploiting for the security of system software. In the paper, first of all, principle of buffer overflow and vulnerability discovering analysis and utilization techniques were discussed. Then one method of static analysis combined with the dynamic analysis of vulnerability discovering analysis was proposed, and a complete vulnerability discovering analysis process was presented, and the availability and effectiveness of the method were verified by actual Microsoft Office vulnerability. Finally, on the basis of the theory and technology, vulnerability discovering analysis system-VulAs was designed and realized under the Windows platform to assist the discovering and analysis of vulnerability, and the effectiveness of the tool was verified.

Keywords Vulnerability, Buffer overflow, Vulnerability discovering analysis, Shellcode, Vulas

1 引言

早在20多年前缓冲区溢出漏洞就被用来进行蠕虫病毒的扩散,可是当时人们并没有对缓冲区溢出问题加以重视。直到1996年Aleph One详细描述了如何在Linux系统中利用栈溢出漏洞[1],这时缓冲区溢出漏洞利用技术才真正浮现于世人眼前。经过无数专家和黑客们针锋相对的研究和实战,缓冲区溢出漏洞利用技术已经普遍在多种操作系统和编译环境下得到了实践,同时已经慢慢趋向于完善。

在21世纪的今天,缓冲区溢出漏洞仍然是影响当前网络与分布式系统安全的主要问题之一[2-5]。根据CNCERT报告,在2011年,国家信息安全漏洞共享平台CNVD共收集整理并公开发布信息安全漏洞5547个。在所有漏洞中,涉及各种应用程序的比例为62.6%,涉及各类网站系统的漏洞比例为22.7%,而涉及各种操作系统的漏洞则占了8.8%。在这些安全漏洞中明确是缓冲区溢出漏洞的数量就有近500个[6]。因此对缓冲区溢出漏洞挖掘分析和利用的研究对于互联网以及系统软件的安全具有重要的意义。

2 漏洞挖掘分析及利用技术

影响计算机和网络安全的罪魁祸首是系统软件中存在的安全漏洞,不法分子或者黑客利用漏洞可以很容易地对目标主机和网络系统进行恶意或者破坏性攻击。只有在对漏洞挖掘分析和利用技术进行深入研究和分析、充分了解漏洞后才能进行漏洞的有效挖掘分析。

2.1 缓冲区溢出

计算机内部,输入数据通常被存放在一个临时空间内,这个临时存放空间就被称为缓冲区,而缓冲区的长度事先已经被程序或者操作系统定义好了。缓冲区犹如一个杯子,可以用来装水,但是大小是固定的。向缓冲区内填充数据,如同向杯子里倒入水,如果数据很大很长,超过了缓冲区(那个杯子)本身的容量,那么结果就如同水一样,四处溢出,数据也会溢

到稿日期:2013-01-17 返修日期:2013-04-16 本文受江苏省自然科学研究计划基金(11KJB520011)资助。
史飞悦(1987-),男,硕士生,主要研究方向为信息安全,E-mail:284629513@qq.com;傅德胜(1950-),男,教授,博士生导师,主要研究方向为信息安全、图像处理与模式识别。

出存储空间，这些装不下的数据会覆盖在合法数据上，这就是缓冲区溢出。在理想的情况下，程序会检查每个数据的长度，并且不允许其超过缓冲区的长度大小，就像在倒水的时候，水快要溢出时就会停止倒入。可是有些程序会假设数据长度总是与所分配的存储空间相匹配，而不作检查，从而为缓冲区溢出留下了隐患。

2.2 漏洞挖掘分析技术

安全性漏洞往往不会对系统软件本身造成功能的影响，因此很难被 QA 工程师的功能性测试发现，而安全性漏洞却有着极高的利用价值，一方面由于它的隐蔽性；同时又由于它的普遍性，因为即使再严格的软件开发流程也会产生低质量的软件，没有百分百安全的软件[8]。而单单想通过挖掘技术了解漏洞的明细是不可能的，必须结合一定的分析手段，所以常常讲的漏洞挖掘技术其实是漏洞的挖掘和分析相结合的技术。

漏洞挖掘分析技术是指对未知漏洞的探索，综合应用各种技术和工具，尽可能地找出软件中的潜在漏洞，然后对已发现漏洞的细节进行深入分析，为漏洞利用、补救等处理措施作铺垫。依据不同的划分标准，漏洞挖掘分析方法可以被划分为不同类型。根据漏洞挖掘分析的自动化程度，可分为手工分析、半自动或自动化分析；根据软件源代码的开放性，可分为白盒分析、黑盒分析和灰盒分析 3 类；根据目标软件的运行状态，又可分为静态分析和动态分析。而对系统软件进行漏洞的挖掘其实是一个多种漏洞挖掘分析技术相互结合、共同使用和优势互补的过程。

2.3 漏洞利用技术

漏洞的利用有很多技术，主要包括覆盖栈利用技术、栈指针覆盖技术、返回到 C 库的利用技术、指针托词利用技术以及堆溢出漏洞利用技术等。不管是什么技术，基本原理都是找到系统软件的脆弱点，通过编写合适的 shellcode，利用内存和寄存器中所需的数据来触发相应的漏洞，从而对电脑进行攻击、破坏，以及获取主机权限。

3 基于静、动态结合的漏洞挖掘分析方法

3.1 基于静、动态结合的漏洞挖掘分析方法

单纯的静态或者动态的漏洞挖掘分析，总是不可避免地存在一些缺陷和问题。但是动态分析和静态分析又有着各自的优势，所以本文设想可以通过将静态与动态分析方法相结合来进行漏洞挖掘分析，以更好地弥补各自的缺陷，发挥各自的优势，达到更好的漏洞挖掘分析效果。

下面以缓冲区溢出漏洞的挖掘分析为例，详细阐述这种方法的处理流程，具体方法为：

(1)首先，对于有源代码的软件程序，将目标程序通过源代码扫描软件进行分析，查找目标程序中存在的容易发生缓冲区溢出的函数，比如 strcpy()、sprintf()等，发现后直接进行修改直至没有。然后将目标程序编译成为可以执行的二进制文件，进而转为执行第(2)步的漏洞挖掘分析。

(2)对于没有源代码的二进制文件，通过反汇编软件获得反汇编代码，然后根据 strcpy()、sprintf()等函数的特征代码在目标软件的反汇编代码中搜索 strcpy()、sprintf()等容易发生缓冲区溢出的函数。再结合 IDA 或者 Ollydbg 动态加载目标软件，在所有搜索到的关键函数处设置断点，通过改变输入数据不断观察调试器的异常行为，最后确认脆弱点。

当然在实际的运用中，缓冲区溢出漏洞只是一种漏洞类型，在应对其他漏洞时，虽然查找监测的漏洞函数不一样，但是主体的思想与方法是一致的。下一节将对实际的缓冲区溢出漏洞进行漏洞的挖掘分析，从而验证该方法的可行性和有效性。

3.2 缓冲区漏洞挖掘分析实例

缓冲区溢出漏洞挖掘分析实例是编号为 MS12-027 的漏洞，该漏洞属于 MSCOMCTL ActiveX 缓冲区溢出漏洞。根据上一小节中缓冲区溢出漏洞的处理流程，对该漏洞采用静动态相结合的漏洞挖掘分析方法，并运用静动态相关分析工具对该漏洞进行详细剖析、挖掘，力图呈现出一个完整的漏洞挖掘分析过程，为漏洞挖掘分析工作提供一个较为清晰的参考模板。同时通过实际的漏洞挖掘分析发现了该漏洞许多未公布的信息。整个实验的环境是 Windows xp sp3，主要的工具是 WinDbg，OllyDbg，WinHex 以及 POC 文件，用于重现漏洞触发现场。

3.2.1 shellcode 定位

通过 WinDbg 附加到 WINWORD. EXE 进程，输入命令参数得到 WinExec 函数返回的地址，此地址已经位于 shellcode 中，反汇编地址 0012ae62 得到 eb226a01 这样的机器码，用 Winhex 反汇编 POC 文件，搜索查找 eb226a01，再继续往上翻阅可以看到 jmp esp 跳转方式的万能跳转地址的逆序 1245fa7f，以及之后的一段 nop 指令 0x90。如图 1(a)、图 1(b)所示。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00001800	30	66	66	31	30	65	62	32	32	36	61	30	31	36	61	30	0ff10eb226a016a0
00001810	30	36	61	30	30	66	66	37	35	66	34	36	61	30	30	65	06a00ff75f46a00e
00001820	38	31	34	66	65	66	66	66	66	30	35	35	35	30	30	30	814fefffff0555000
00001830	30	30	30	35	30	65	38	30	39	66	65	66	66	66	66	30	00050e809feffff0

(a) 机器码 eb226a01

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00001320	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	0000000000000000
00001330	30	30	30	30	30	30	30	30	30	31	32	34	35	66	61	37	000000000001245fa7
00001340	66	39	30	39	30	39	30	39	30	39	30	39	30	39	30	39	f9090909090909
00001350	30	38	62	63	34	30	35	31	30	30	31	30	30	30	30	63	08bc40510010000c

(b) jmp esp 跳转地址

图 1

3.2.2 漏洞成因分析

POC 文件在执行中先要通过漏洞触发函数触发漏洞，再在某一恰当的时刻将位置 A 中的内存的值修改为万能跳转地址 0x7afa4512，然后执行并跳转到 shellcode，进而执行安排好的 shellcode。因此，要知道该存在漏洞触发的函数位置，就需要先找到程序何时对内存位置 A 进行了修改，修改前的值是多少，之后就能确定存在漏洞的函数。

通过设置内存写入断点和内存执行断点，得到漏洞函数

地址，单步执行，进入函数可以看到漏洞函数的具体信息，如图 2 所示。

```
275c8b4e  55            push    ebp
275c8b4f  8bec          mov     ebp,esp
275c8b51  83ec14        sub     esp,14h
.....
275c8b61  e88efdffff    call    MSCOMCTL!DllGetClassObject+0x3a8bb (275c88f4)
.....
275c8b7a  837df408      cmp     dword ptr [ebp-0Ch],8
275c8b7c  0f82efa20000  jb      MSCOMCTL!DllGetClassObject+0x44e3a (275d2e73)
275c8b8c  e863fdffff    call    MSCOMCTL!DllGetClassObject+0x3a8bb (275c88f4)
```

图 2　漏洞函数的详细信息

可以看到该函数只开辟了 20 字节的空间，而两次调用了函数：MSCOMCTL! DllGetClassObject ＋ 0x3a8bb（275c88f4）。该函数第一次调用使用了 12 字节空间；在第二次调用之前对参数大小进行了检查，该大小与 8 比较，本该是大于等于 8 时直接跳转结束，但程序中由于程序员的疏忽变成了小于等于 8 时结束，而当输入参数大于 8 时则继续调用 MSCOMCTL!DllGetClassObject＋0x3a8bb（275c88f4）。于是当 POC 文件中直接调用写入了 0x8282 大小的数据时，同样进入到该函数进行调用执行，这就直接导致栈发生了溢出，使得调用函数无法正常返回，然后通过将此处返回地址精心覆盖成跳转地址 0x7ffa4512，直接导致了恶意代码的执行，这是一个典型的通过缓冲区溢出来执行其他代码的例子。

4　漏洞挖掘分析系统设计与实现

学术界一直热衷于使用静态分析的方法寻找源代码中的漏洞；而工业界普遍采用的漏洞挖掘方法是动态分析技术 fuzz，实际上它是一种黑盒测试。现在市面上，由于 Linux 的开源性，在 Linux 下缓冲区溢出漏洞挖掘分析技术已经颇为成熟，已经出现了许多工具比如 SPIKE[9]，Autodafe[10]，PROTOS[11]，beSTORM[12]，Google Bunny，Dfuz 等。这些工具都是基于 fuzz 的测试工具，通过构造半有效的带有攻击性的畸形数据输，用以触发各种类型的漏洞，它们都各有优势，也取得了不错的成绩。在国内，刘奇旭等[13]针对 TFTP 协议漏洞挖掘设计了 tftpServerFuzzer，李伟明等[14]对网络协议的自动化模糊测试漏洞挖掘方法进行了深入研究，杨丁宁等[15]对 ActiveX 控件漏洞挖掘设计实现了 ActiveX-Fuzzer。

然而对于 Windows 系统下的漏洞挖掘工具并没有比较通用且效果较好的，比较典型的是针对缓冲区溢出漏洞的 BugScam，但是由于作者 Cheers Halvar 在一开始并没有意识到自己的作品价值，没有对它进行进一步的设计开发，它也就成为了一个开始[16]。但是这种采用 IDA pro 和 IDC 脚本结合的方式给了漏洞挖掘分析工具研究人员很大的启发，本文的 VulAs 半自动化漏洞挖掘分析工具也是按照这个思路进行设计开发的。

4.1　控制程序设计

整个系统设计思路为：使用 IDA 对需要分析的关键文件（可以是 dll 文件、exe 文件等）进行反汇编，通过编写符合特定要求的 IDC 脚本文件对反汇编结果进行分析，而漏洞模型的程序实现也是在 IDC 脚本中。对分析的结果以 html 报告的方式进行显示，以便分析人员查看和存储。系统的逻辑结构如图 3 所示，程序运行界面如图 4 所示。

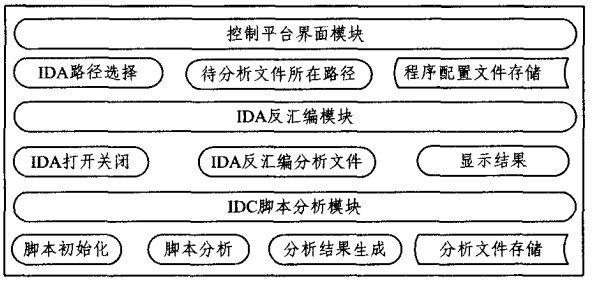

图 3　系统逻辑结构

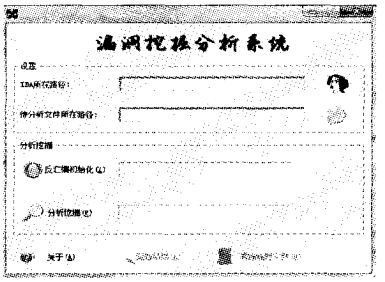

图 4　程序运行界面

4.2　IDA 反汇编模块

当程序路径配置成功之后，通过构造调用 IDA 执行反汇编分析命令，基本构造方式为：IDApath idag. exe-A-c-SVulas. idc，这条命令表示：以不显示 IDA 对话框的自动和批处理模式执行 IDA，并自动运行 Vulas. idc 脚本。

4.3　脚本分析模块

脚本分析模块包括调控模块以及分析模块，而漏洞挖掘分析系统的核心模块就是脚本分析模块，脚本分析模块主要就是将漏洞模型进行程序的实现。图 5 显示脚本分析调控模块的流程。而分析模块则根据漏洞模型找出可能存在问题的地方。比如对于 strcpy 函数的检测，首先判断 strcpy，获取函数地址，然后获取参数缓冲区空间 destminsize、destmaxsize、srcminsize、srcmaxsize 的缓冲区大小；最后对 4 个空间大小进行比较判断，确定是否存在溢出风险，如果存在风险，则将存在风险的地址写入分析报告文件。

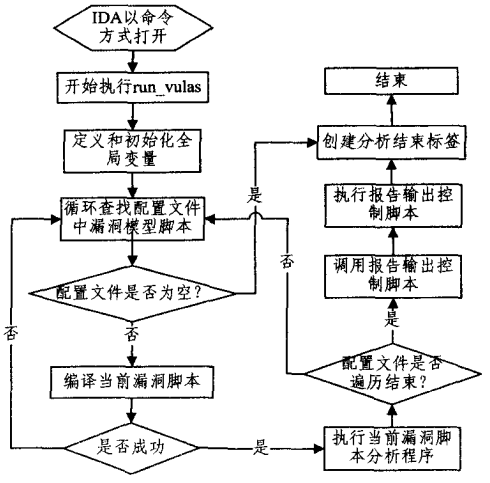

图 5　脚本分析调控模块的流程图

4.4 实验及结果分析

为了验证漏洞挖掘分析系统 VulAs 的有效性，对含有缓冲区溢出的漏洞程序 strcpy_overflow. exe 进行漏洞的实际挖掘分析。表 1 显示了相应的调用地址、严重程度以及相应的描述。

表 1 程序分析结果的主要内容

Address	Severity	Description
401054	2	UNKNOWN_SOURCE_SIZE: investigated manually
4017f5	8	target buffer is smaller than the source buffer
40182d	2	UNKNOWN_SOURCE_SIZE: investigated manually
4018c8	8	target buffer is smaller than the source buffer
4049c4	2	UNKNOWN_SOURCE_SIZE: investigated manually

由表 1 可见，在 IDA 中查看相应地址发现确实都存在 strcpy()函数的调用，其中在地址 004017F5 调用 strcpy()函数之前，直接将目标 DstBuf 地址传送到 eax 寄存器，但是并没有检查传送内容的长度就直接调用 strcpy()函数，这就为缓冲区溢出造就可能。通过分析说明，漏洞挖掘分析系统对于缓冲区溢出漏洞的检测具有较高的准确性与有效性。

结束语 缓冲区溢出漏洞挖掘分析、利用的研究已经成为当今国内外安全研究的热点，如何解决由缓冲区溢出漏洞所引起的安全问题也成为了安全研究人员的必修课题。本文对缓冲区溢出原理以及漏洞挖掘分析与利用技术进行分析总结，提出了一种基于动静态相结合的漏洞挖掘分析方法，并设计实现了漏洞挖掘分析系统 Vulas，最后通过实验验证了系统的有效性与准确性。

本文的半自动化漏洞挖掘分析系统虽然可以在 Windows 平台下对漏洞挖掘起到一定的作用，但是对于其他平台比如 Linux，还没有进行验证，需要进行下一步的验证。同时对于漏洞挖掘分析系统 Vulas 中核心的漏洞模型库的设计还有待于完善、完整，而对更加合理高效的漏洞特征描述语言的探索则需要继续改进与研究。

参 考 文 献

[1] Aleph One. Smashing The Stack For Fun And Profit [J]. Phrack, 1996, 7(49)

[2] 邓爽. 缓冲区溢出攻击分析及防范策略研究[D]. 济南：山东大学，2009

[3] 李毅超，刘丹，韩宏，等. 缓冲区溢出漏洞研究与进展[J]. 计算机科学，2008，35(1)：87-89，125

[4] 林志强，夏耐，茅兵，等. 缓冲区溢出研究综述[J]. 计算机科学，2004，31(9)：110-113，160

[5] 王业君，倪惜珍，文伟平，等. 缓冲区溢出攻击原理与防范的研究[J]. 计算机应用研究，2005，22(10)：101-104

[6] 2011 年我国互联网网络安全态势综述[EB/OL]. http://www.cert.org.cn/UserFiles/File/20120319201lannualreport.pdf

[7] 彭青白. 缓冲区溢出漏洞的挖掘与利用方法研究[D]. 武汉：华中科技大学，2009

[8] Voas J M, McGraw G. Software Fault Injection: Inoculating Programs Against Errors[M]. John Wiley and Sons, New York, 1998

[9] DaveAitel. TheAdvantages of Block-Based ProtocolAnalysis for Security Testing[R]. Immunity, Inc., 2003

[10] AutoDafe [EB/OL]. http://autodafe.sourceforge.net, http://autodafe.sourceforge.net/docs/autodafe.pdf

[11] Oulu University Secure Programming Group. PROTOS Test-Suite: c06-snmpv1[R]. University of Oulu, Electrical and Information Engineering, 2002

[12] BeyondSecurity. beStrom[EB/OL]. http://www.beyondsecurity.com/bestorm_whitepaper.html

[13] 刘奇旭，张玉清. 基于 Fuzzing 的 TFTP 漏洞挖掘技术[J]. 计算机工程，2007，33(20)：142-147

[14] 李伟明，张爱芳，刘建财，等. 网络协议的自动化模糊测试漏洞挖掘方法[J]. 计算机学报，2011，34(2)：242-255

[15] 杨丁宁，肖晖，张玉清. 基于 Fuzzing 的 ActiveX 控件漏洞挖掘技术研究[J]. 2012，49(7)：1525-1532

[16] Kkqq, bugscam Analysis[J]. 绿盟安全月刊，2004(46)

(上接第 142 页)

结束语 基于虚拟化技术，提出并设计了一种支持行业数据应用托管及数据隐私保护的数据库即服务系统，其允许用户在公共的 IT 基础设施之上利用虚拟机建立具有数据、性能隔离、可靠性保障的独立数据库及相关数据的应用。在数据隐私保护方面，采用 CryptDB 系统对数据进行加密，同时利用多种加密策略以及可动态调整的加密策略技术，解决了基于加密数据执行 SQL 查询的问题。下一步工作将研究在保护数据隐私的同时如何进行数据的优化处理以进一步提高数据处理效率，减少系统开销，提高用户体验。

参 考 文 献

[1] Ashraf A. Deploying database appliances in the cloud[J]. IEEE Data Engineering Bullentin, 2009, 32(1): 13-20

[2] Popa R A, Redfield C M S, Zeldovich N, et al. CryptDB: Protecting Confidentiality with Encrypted Query Processing[C]// Proceedings of the 23rd ACM Symposium on Operating Systems Principles(SOSP 2011). Cascais, Portugal, October 2011

[3] Boldyreva A, Chenette N, Lee Y, et al. Order preserving symmetric encryption[C]// Proceedings of the 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Cologne, Germany, April 2009

[4] Paillier P. Public-key cryptosystems based on composite degree residuosity classes[C]// Proceedings of the 18th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Prague, Czech Republic, May 1999

[5] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[C]// Proceedings of the 21st IEEE Symposium on Security and Privacy. Oakland, CA, May 2000

[6] 王卓昊，王希诚. 面向托管的数据库即服务系统及资源优化技术[J]. 计算机工程与应用，2011，47(27)：19-23