第 31 卷 第 21 期 计 算 机 工 程 2005 年 11 月
Vol.31 № 21 Computer Engineering November 2005
·安全技术· 文章编号：1000—3428(2005)21—0121—03 文献标识码：A 中图分类号：TP393

攻击技术分类研究

诸葛建伟，叶志远，邹 维
（北京大学计算机科学技术研究所，北京 100871）

摘 要：以系统化构建一个为入侵检测与防御提供知识基础的攻击知识库为目标，从攻击者角度出发对攻击技术进行了分类研究，提出了一种符合分类标准的攻击技术分类层次结构，并概述了每种不同的攻击技术。

关键词：网络安全；攻击；分类

Research on Classification of Attack Technologies

ZHUGE Jianwei, YE Zhiyuan, ZOU Wei
(Institute of Computer Science and Technology, Peking University, Beijing 100871)

【Abstract】For the goal to systemize an attack knowledge base which provides knowledge for intrusion detection and prevention, the classification of attack technologies is researched from the point of view of attackers, and a classification taxonomy which accords with the classification criterions is proposed, furthermore, each type of attack technology is introduced in summary.

【Key words】Network security; Attack; Classification

攻击技术作为攻击最为重要的一个内在属性，对其进行分析和分类研究，对了解攻击的本质，以更准确地对其进行检测和响应具有重要的意义。已提出的一些分类方法，包括我们在构建攻击知识库所采用的多维攻击分类体系，都已经把对攻击技术作为一个重要的维度，但已有的分类方法均是仅仅在概念层次上为其给出一个简单的分类列表，并不满足完整性、实用性等分类标准。本文从攻击者的角度出发，对现有的攻击技术进行了深入的研究和分析，提出了对攻击技术的层次化分类结构，并对每类攻击技术进行概述。

1 相关的工作

目前唯一得到广泛应用的公开攻击特征库为著名开源入侵检测系统 Snort 的规则库，而其采用的分类方法是根据经验进行罗列，出发的角度也不尽相同，有攻击结果、攻击技术、攻击目标等，从而导致分类结果十分混乱。

为了判断一个分类方法是否合理并满足实际应用的需求，Amoroso[2] 给出了一个攻击分类方法应满足的分类标准，即互斥性（分类类别不应重叠）、完备性（覆盖所有可能的攻击）、非二义性（类别划分清晰）、可重复性（对一个样本多次分类结果一致）、可接受性（符合逻辑和直觉）和实用性（可用于深入研究和调查）6 个特性。

不少研究工作致力于提出一个满足上述分类标准的攻击分类体系，提出的大部分攻击分类体系均采用多个维度对攻击进行划分，基本上都采纳了攻击技术这一维度，如 Lindqvist 等人[3]结合使用攻击技术和攻击后果的分类方法，MIT 的林肯实验室在开发 DARPA 入侵检测系统测试数据的过程中，提出的按初始权限、攻击方法、获得权限和攻击动作进行分类的方法[4]等，但它们在攻击技术这一维度上仍只简单罗列了一些常见的攻击技术，不具备完整性和实用性。

2 攻击技术分类

如图 1 所示，从攻击者的角度出发，攻击的步骤可分为探测（Probe）、攻击（Exploit）和隐藏（Conceal）。我们的攻击技术分类方法据此分为探测技术、攻击技术和隐藏技术 3 大类，并在每类中对各种不同的攻击技术进行细分。

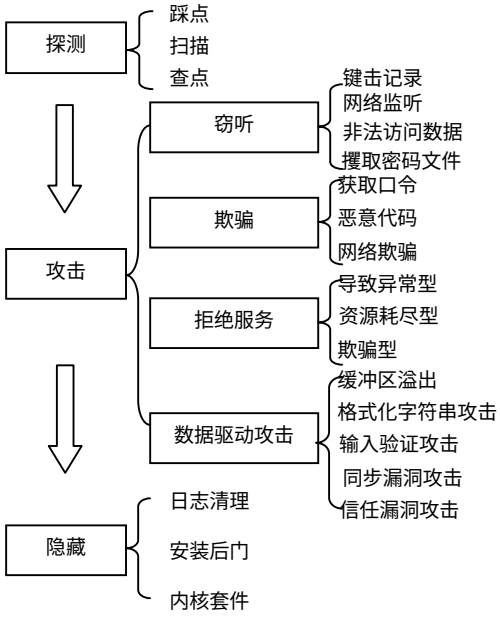

图 1 攻击技术分类层次结构

2.1 探测技术

探测是黑客在攻击开始前必需的情报收集工作，攻击者通过这个过程需要尽可能多地了解攻击目标安全相关的方方面面的信息，以便能够集中火力进行攻击。探测又可以分为 3 个基本步骤：踩点，扫描和查点。

踩点指攻击者结合各种工具和技巧，以正常合法的途径对攻击目标进行窥探，对其安全情况建立完整的剖析图。常

作者简介：诸葛建伟(1980—)，男，博士生，主研方向：攻防知识建模，入侵检测与防御；叶志远，高工；邹 维，教授

收稿日期：2004-10-05 **E-mail：**zhugejianwei@icst.pku.edu.cn

用的方法有通过搜索引擎对开放信息源进行搜索、域名查询、DNS 查询、网络勘察等。

扫描则是攻击者获取活动主机、开放服务、操作系统、安全漏洞等关键信息的重要技术。扫描技术包括 Ping 扫描（确定哪些主机正在活动）、端口扫描（确定有哪些开放服务）、操作系统辨识（确定目标主机的操作系统类型）和安全漏洞扫描（获得目标上存在着哪些可利用的安全漏洞）。

查点是攻击者常采用的从目标系统中抽取有效账号或导出资源名的技术，查点的信息类型大体可以归为网络资源和共享资源、用户和用户组和服务器程序及其旗标 3 类。

2.2 攻击（Exploit）技术

在攻击阶段，攻击者通过探测阶段掌握的有关攻击目标的安全情况会选择不同的攻击方法来达成其攻击目的。攻击方法层出不穷，但可以将其归为以下 4 类，即窃听技术、欺骗技术、拒绝服务和数据驱动攻击。

2.2.1 窃听技术

窃听技术指攻击者通过非法手段对系统活动的监视从而获得一些安全关键信息。目前属于窃听技术的流行攻击方法有键击记录器、网络监听、非法访问数据和攫取密码文件。

键击记录器是植入操作系统内核的隐蔽软件，通常实现为一个键盘设备驱动程序，能够把每次键击都记录下来，存放到攻击者指定的隐藏的本地文件中。著名的有 Win32 平台下适用的 IKS 等。

网络监听则是攻击者一旦在目标网络上获得一个立足点之后刺探网络情报的最有效方法，通过设置网卡的混杂（promiscuous）模式获得网络上所有的数据包，并从中抽取安全关键信息，如明文方式传输的口令。Unix 平台下提供了 libpcap 网络监听工具库和 tcpdump、dsniff 等著名监听工具，而在 Win32 平台下也拥有 WinPcap 监听工具库和 windump、dsniff for Win32、sniffer 等免费工具，另外还拥有大量简单易用的商业监听产品，如 Sniffer Pro。

非法访问数据指攻击者或内部人员违反安全策略对其访问权限之外的数据进行非法访问。

攫取密码文件是攻击者进行口令破解获取特权用户或其他用户口令的必要前提，关键的密码文件如 Win9x 下的 PWL 文件、Win NT/2000 下的 SAM 文件和 Unix 平台下的 /etc/password 和/etc/shadow。

2.2.2 欺骗技术

欺骗技术是攻击者通过冒充正常用户以获取对攻击目标访问权或获取关键信息的攻击方法，属于此类的有获取口令、恶意代码、网络欺骗等攻击手法。

获取口令的方式有通过缺省口令、口令猜测和口令破解 3 种途径。某些软件和网络设备在初始化时会设置缺省的用户名和密码，意在允许厂家有能力绕过被锁闭或遗忘的管理员账号，但这些缺省口令也给攻击者提供了最容易利用的脆弱点。口令猜测则是历史最为悠久的攻击手段，由于用户普遍缺乏安全意识，不设密码或使用弱密码的情况随处可见，这也为攻击者进行口令猜测提供了可能。口令破解技术则提供了进行口令猜测的自动化工具，通常需要攻击者首先获取密码文件，然后遍历字典或高频密码列表从而找到正确的口令。著名的工具有 John the Ripple、Crack 和适用于 Win32 平台的 L0phtcrack 等。

恶意代码包括特洛伊木马应用程序、邮件病毒、网页病毒等，通常冒充成有用的软件工具、重要的信息等，诱导用户下载运行或利用邮件客户端和浏览器的自动运行机制，在启动后暗地里安装邪恶的或破坏性软件的程序，通常为攻击者给出能够完全控制该主机的远程连接。

网络欺骗指攻击者通过向攻击目标发送冒充其信任主机的网络数据包，达到获取访问权或执行命令的攻击方法。具体的有 IP 欺骗、会话劫持、ARP（地址解析协议）重定向和 RIP（路由信息协议）路由欺骗等。

IP 欺骗是攻击者将其发送的网络数据包的源 IP 地址篡改为攻击目标所信任的某台主机的 IP 地址，从而骗取攻击目标信任的一种网络欺骗攻击方法。通用应用于攻击 Unix 平台下通过 IP 地址进行认证的一些远程服务如 rlogin、rsh 等，也常应用于穿透防火墙。

会话劫持指攻击者冒充网络正常会话中的某一方，从而欺骗另一方执行其所要的操作。目前较知名的如 TCP 会话劫持，通过监听和猜测 TCP 会话双方的 ACK，插入包含期待 ACK 的数据包，能够冒充会话一方达到在远程主机上执行命令的目的。支持 TCP 会话劫持的工具有最初的 Juggernaut 产品和著名的开源工具 Hunt。

ARP 提供将 IP 地址动态映射到 MAC 地址的机制，但 ARP 机制很容易被欺骗，攻击主机可以发送假冒的 ARP 回答给目标主机发起的 ARP 查询，从而使其错误地将网络数据包都发往攻击主机，导致拒绝服务或者中间人攻击。

RIP 由于其 v1 没有身份认证机制，v2 使用 16 字节的明文密码，因此攻击者很容易发送冒充的数据包欺骗 RIP 路由器，使之将网络流量路由到指定的主机而不是真正希望的主机，达到攻击的目标。

2.2.3 拒绝服务攻击

拒绝服务攻击指中断或者完全拒绝对合法用户、网络、系统和其他资源的服务的攻击方法，被认为是最邪恶的攻击，其意图就是彻底地破坏，而这往往比真正取得他们的访问权要容易得多，同时所需的工具在网络上也唾手可得。因此拒绝服务攻击，特别是分布式拒绝服务攻击对目前的互联网络构成了严重的威胁，造成的经济损失也极为庞大。

拒绝服务攻击的类型按其攻击形式划分包括导致异常型、资源耗尽型、欺骗型。另外分布式拒绝服务攻击（DDoS）采用资源耗尽型的攻击方式，但由于其特殊性，我们将其另归为一类。

导致异常型拒绝服务攻击利用软硬件实现上的编程缺陷，导致其出现异常，从而使其拒绝服务。如著名的 Ping of Death 攻击和利用 IP 协议栈对 IP 分片重叠处理异常的 Treadrop 攻击。

资源耗尽型拒绝服务攻击则通过大量消耗资源使得攻击目标由于资源耗尽不能提供正常的服务。视资源类型的不同可分为带宽耗尽和系统资源耗尽两类。带宽耗尽攻击的本质是攻击者通过放大等技巧消耗掉目标网络的所有可用带宽。著名的如 Smurf 攻击，冒充目标网络向多个广播地址发送 ping 包，造成数量庞大的 ping 响应湮没攻击目标网络。

系统资源耗尽攻击指对系统内存、CPU 或程序中的其它资源进行消耗，使其无法满足正常提供服务的需求。著名的 Syn Flood 攻击即是通过向目标服务发送大量的 syn 包造成服务的连接队列耗尽，无法再为其它正常的连接请求提供服务。

分布式拒绝服务攻击则是通过控制多台傀儡主机，利用它们的带宽资源集中向攻击目标发动总攻，从而耗尽其带宽或系统资源的攻击形式。如图 2 所示，DDoS 攻击的第一步

是瞄准并获得尽可能多的傀儡主机的系统管理员访问权，然后上传 DDoS 攻击并运行，大多数 DDoS 守护进程运行方式的监听发起攻击的指令，收到后并向指定的目标网络发动 flood 攻击。目前著名的 DDoS 工具有 TFN（Tribe Flood Network）、TFN2K、Trinoo、WinTrinoo 和 Stacheldraht 等。

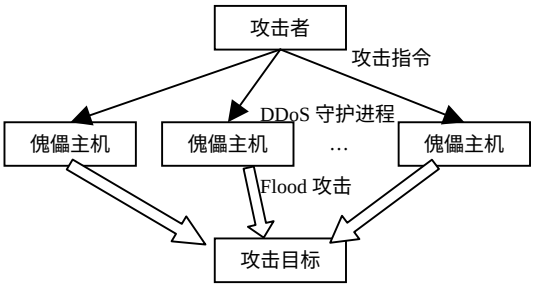

图 2 DDoS 攻击原理

2.2.4 数据驱动攻击

数据驱动攻击是通过向某个程序发送数据，以产生非预期结果的攻击，通常为攻击者给出访问目标系统的权限，数据驱动攻击分为缓冲区溢出攻击、格式化字符串攻击、输入验证攻击、同步漏洞攻击、信任漏洞攻击等。

缓冲区溢出攻击的原理是通过往程序的缓冲区写入超出其边界的内容，造成缓冲区的溢出，使得程序转而执行其他攻击者指定的代码，通常是为攻击者打开远程连接的 ShellCode，以达到攻击目标。最初的缓冲区溢出攻击仅限于 Unix 平台，大量缓冲区溢出漏洞被发掘并给出攻击脚本，著名的一些服务程序如 Apache、Wuftpd、Sendmail、OpenSSH 等都曾被发掘出缓冲区溢出漏洞。而 Win32 平台由于内存地址含有空字节造成缓冲区截断、操作系统会修改废弃的缓冲区等一些因素给缓冲区溢出攻击造成一些困难。但大量 Windows 平台下的缓冲区溢出漏洞也被利用，近年来著名的蠕虫如 Code-Red、SQL.Slammer、Blaster 和 Sasser 都是通过缓冲区溢出攻击获得系统管理员权限后进行传播。

格式化字符串攻击主要是利用由于格式化函数的微妙程序设计错误造成的安全漏洞，通过传递精心编制的含有格式化指令的文本字符串，以使目标程序执行任意命令。

输入验证攻击针对程序未能对输入进行有效验证的安全漏洞，使得攻击者能够让程序执行指定的命令。

同步漏洞攻击利用程序在处理同步操作时的缺陷，如竞争状态、信号处理等问题，以获取更高权限的访问。

发掘信任漏洞攻击则利用程序滥设的信任关系获取访问权的一种方法，著名的有 Win32 平台下互为映像的本地和域 Administrator 凭证、LSA 密码（Local Security Authority）和 Unix 平台下 SUID 权限的滥用和 X Window 系统的 xhost 认证机制等。

2.3 隐藏技术

攻击者在完成其攻击目标（如获得 root 权限）后，通常会采取隐藏技术来消除攻击留下的蛛丝马迹，避免被系统管理员发现，同时还会尽量保留隐蔽的通道，使其以后还能轻易地重新进入目标系统。隐藏技术主要包括日志清理、安装后门、内核套件等。

日志清理主要对系统日志中攻击者留下的访问记录进行清除，从而有效地抹除自己的行动踪迹。Unix 平台下较常用的日志清理工具包括 zap, wzap, wted 和 remove。攻击者通常在获得特权用户访问权后会安装一些后门工具，以便轻易地重新进入或远程控制该主机；攻击者还可以对系统程序进行特洛伊木马化，使其隐藏攻击者留下的程序、运行的服务等。内核套件则直接控制操作系统内核，提供给攻击者一个完整的隐藏自身的工具包。

2.4 各种攻击技术的融合

目前复杂的攻击工具往往融合了多种不同的攻击技术，特别是计算机蠕虫等能够自动运行、传播并造成破坏的 Agent。蠕虫指的是能在网络上完全地复制自身的独立可执行代码，而病毒则需要宿主程序通过某种方式将其激活。蠕虫技术融合了自复制技术、扫描技术以及缓冲区溢出等攻击技术。目前的病毒也逐渐融入将一些网络攻击的技术，如著名的 Nimda 病毒通过电子邮件、共享目录以及主动攻击 IIS 缓冲区溢出漏洞等形式达到广泛传播的效果。

3 实验分类结果

我们已经通过对 DARPA 资助的 1999 年入侵检测系统评测数据中所包含的 62 种攻击[4]进行分析，构建了一个基本的攻击知识库。在构建的同时，使用了上述的攻击技术分类方法对这些攻击进行分类，分类结果如表 1 所示。通过此分类结果可以说明我们提出的攻击技术分类方法满足 Amoroso[2] 提出的分类标准，即满足互斥性、完备性、非二义性、可重复性、可接受性和实用性 6 个特性。

表 1 对 DARPA 1999 IDS 评测数据中包含攻击的分类结果

攻击技术				攻击方法
探测	踩点			无
	扫描	Ping 扫描		ipsweep
		端口扫描		portsweep, resetscan
		操作系统辨识		queso
		漏洞扫描		satan, mscan
	查点			ls, ntinfoscan
攻击	窃听	键击记录		xsnoop
		网络监听		illegalsniffer
		非法访问数据		secret
		攫取密码		snmpget, ncftp
	欺骗	获取口令		guest, guessftp, guesstelnet, guesspop, dict
		恶意代码		sshtrojan, ppmacro, xlock, sechole, casesen, frames poofer
		网络欺骗		arppoison, httptunnel
	拒绝服务	资源耗尽型	带宽耗尽型	smurf, udpstorm
			系统资源耗尽型	mailbomb, processtable, sshprocesstable, neptune, apache2, back
		导致异常型		crassiis, land, treadrop, dosnuke, pod, syslogd, selfping, warezmaster, warez client
		欺骗型		tcpreset
	数据驱动攻击	缓冲区溢出攻击		sendmail, imap, named, eject, ffbconfig, fdformat, xterm,
		格式化字符串		loadmodule
		输入验证攻击		phf
		同步漏洞攻击		ps
		信任漏洞攻击		ftpwrite, yaga, ntfsdos, anypw, perl, sqlattack
隐藏	日志清理			无
	安装后门			netcat, netbus
	内核套件			无
各种技术的融合				无

（下转第 126 页）

件的方式，使其只作用于 EBD 设置了漏洞标志（e_ident[14]）的 ELF 文件所对应的进程。由于只需要在加载文件时增加一条简单的判断语句，因此对系统性能的影响很小。修改后的 PAX 不作用于正常的程序，所以对于正常程序不会有兼容性问题。

4 实验测试

我们对 EBD 进行了有效性测试，对 EBD + PAX 方案与原有的 PAX 方案进行了性能和兼容性的对比性测试。测试环境见表 2。

表 2 测试环境

硬件环境	Intel Celeron(R)1.7GHz CPU，256MB SDRAM，30GB IDE Disk
软件环境	RedHat 9.0 完全安装版

4.1 EBD 有效性实验

我们用 EBD 对 Linux 下几种常用的程序进行了漏洞检测，结果如表 3 所示（Y: yes, N: no; P: pass, A: alert）。

表 3 EBD 有效性测试结果

程序名称、版本	是否存在漏洞	检测结果
apache-1.3.29	N	P
wu-ftpd-2.6.1	N	A
gawk-3.1.2	Y	A
grep-2.5.1	N	P
hypermail-2.1.5	Y	A

4.2 EBD + PAX 性能测试

PAX 影响系统性能的主要因素是内存操作的速度，因此，我们对原 Linux 系统、增加 PAX 补丁的 Linux 系统和增加改进后的 PAX 补丁的 Linux 系统进行了内存操作的压力对比测试。测试程序如下：

```
#include <stdio.h>
#define TIMES 10000
#define OVERTLB 257
   int main(int argc, char *argv[])   {
char *buf;
int i, j, assined;
assined = atoi(argv[1]);
buf = (char *) malloc(assined * OVERTLB);
for(i = 0; i < TIMES; i++)
for(j = 0; j < OVERTLB; j++)
buf[j * assined] = 'a';
return 0; }
```

性能对比测试结果示意图见图 2，X 代表变量 assigned 的值；Y 代表系统执行时间(ms)。测试结果表明，与 PAX 相比，EBD＋PAX 对系统性能影响很小。

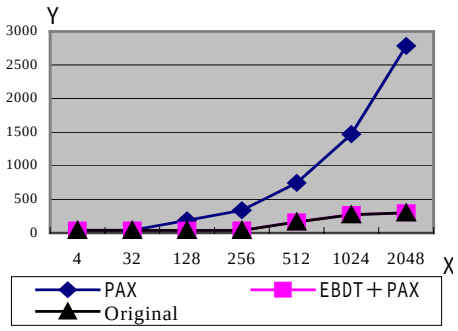

图 2 性能对比测试结果示意图

4.3 EBD + PAX 兼容性测试

系统打上 PAX 的补丁后，XFree86 服务因无法进行信号处理不能启动。改进 PAX 后，XFree86 可以启动，正常运行。

5 结论及展望

本文提出了一种针对 ELF 文件潜在缓冲区溢出漏洞的检测技术——EBD。测试结果表明 EBD 可以有效检测出 ELF 文件中对不安全的动态链接库函数的调用，从而可以根据定义的规则判断出程序中是否存在潜在的缓冲区溢出漏洞。作为 EBD 的应用实例，EBD+PAX 在很大程度上解决了 PAX 的兼容性问题，大大减小 PAX 对系统性能的影响，从而证明了 EBD 的实用价值。只依靠库函数调用作为判断依据的检测粒度较粗，所以 EBD 对一些虽然调用了不安全函数但没有缓冲区溢出漏洞的程序会产生错误判断。因此，需要对缓冲区溢出漏洞继续深入研究，引入其他检测规则作为补充，这是我们进一步研究的方向。

参考文献

1 Ruwase O, Lam M. A Practical Dynamic Buffer Overflow Detector. In: Proceedings of the 11th Annual Network and Distributed System Security Symposium, 2004-02

2 Wilander J, Kamkar M. A Comparison of Publicly Available Tools for Static Intrusion Prevention. In: Proceedings of the 7th Nordic Workshop on Secure IT Systems, 2002-11

3 Wilander J, Kamkar M. A Comparison of Publicly Available Tools for Dynamic Buffer Overflow Prevention. In: Proceedings of the 10th Network and Distributed System Security Symposium, 2003-02

4 PAX On Line Documentation. http://pax.grsecurity.net, 2004-03

5 何先波, 唐宁九, 吕　方等. ELF 文件格式及应用. 计算机应用研究, 2001,18 (11): 144-150

（上接第 123 页）

4 结论

本文在深入研究和分析各种不同的攻击技术的前提下，提出了一个符合分类标准的攻击技术分类层次结构，并对每种攻击技术进行了概述。这一分类方法已经实际应用于系统化构建一个完整的、实用的攻击知识库，为整体维护网络安全的入侵检测与防御系统提供知识基础。

参考文献

1 诸葛建伟, 徐　辉, 潘爱民. 基于面向对象方法的攻击知识模型. 计算机研究与发展, 2004, 41(7)

2 Amoroso E G. Fundamentals of Computer Security Technology. Englewood Cliffs (New Jersey): Prentice Hall, 1994

3 Lindqvist U, Jonsson E. How to Systematically Classify Computer Security Intrusions. In: Proceedings of the 1997 IEEE Symposium on Security and Privacy, Oakland, CA, IEEE Computer Society Press, 1997: 154-163

4 Kendall K. A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems[Master Thesis]. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1999

5 One A. Smashing the Stack for Fun and Profit. Phrack Magazine, 1996, 7(49)

6 Spyrit D, Jack A B. Win32 Buffer Overflows (Location, Exploitation and Prevention). Phrack Magazine, 2000, 55(15)