

目 录

第 1 部分 软件安全

第 1 章 Set-UID 特权程序原理及 攻击方法	3	1.6 最小特权原则	20
1.1 特权程序存在的必要性	3	1.7 总结	21
1.1.1 密码困境	3	第 2 章 通过环境变量实现攻击	22
1.1.2 不同类型的特权程序	4	2.1 环境变量	22
1.2 Set-UID 机制	5	2.1.1 如何访问环境变量	22
1.2.1 超人的故事	5	2.1.2 进程获取环境变量的方式	23
1.2.2 特权程序的工作原理	5	2.1.3 环境变量在内存中的位置	25
1.2.3 一个 Set-UID 程序的例子	7	2.1.4 shell 变量和环境变量	26
1.2.4 Set-UID 机制的安全性	8	2.2 环境变量带来的攻击面	29
1.2.5 Set-GID 机制	8	2.3 通过动态链接器的攻击	30
1.3 可能出现的问题: 超人的 遭遇	8	2.3.1 静态和动态链接	30
1.4 Set-UID 程序的攻击面	9	2.3.2 案例分析: LD_PRELOAD 和 LD_LIBRARY_PATH	32
1.4.1 用户输入: 显式输入	10	2.3.3 案例分析: OS X 动态链接器	34
1.4.2 系统输入	10	2.4 通过外部程序进行攻击	35
1.4.3 环境变量: 隐藏的输入	11	2.4.1 两种调用外部程序的典型方式	35
1.4.4 权限泄露	12	2.4.2 案例分析: PATH 环境变量	36
1.5 调用其他程序	14	2.4.3 减小攻击面	37
1.5.1 不安全的方式: 使用 system()	14	2.5 通过程序库攻击	38
1.5.2 安全的方式: 使用 execve()	17	2.6 通过程序本身的代码进行攻击	39
1.5.3 用其他语言调用外部命令	18	2.7 Set-UID 机制和服务机制的 比较	41
1.5.4 经验教训: 隔离的原则	19	2.8 总结	42

第 3 章 Shellshock 攻击	43	4.6 构造 shellcode	74
3.1 背景知识: shell 函数	43	4.6.1 用 C 语言编写一段恶意代码	75
3.2 Shellshock 漏洞	45	4.6.2 构造 shellcode 的核心方法	75
3.2.1 Shellshock 漏洞	46	4.6.3 一个 shellcode 示例的说明	76
3.2.2 Bash 源代码中的错误	46	4.7 防御措施概述	79
3.2.3 Shellshock 漏洞的利用	48	4.8 地址随机化	80
3.3 利用 Shellshock 攻击 Set-UID 程序	48	4.8.1 Linux 中的地址随机化	81
3.4 利用 Shellshock 攻击 CGI 程序	50	4.8.2 地址随机化的有效性	82
3.4.1 实验环境准备	50	4.9 StackGuard	84
3.4.2 Web 服务器调用 CGI 程序	51	4.9.1 观察到的现象和解决方法的提出 ..	84
3.4.3 攻击者向 Bash 发送数据	52	4.9.2 在函数中添加代码	84
3.4.4 实施 Shellshock 攻击	53	4.9.3 gcc 中的 StackGuard 实现	86
3.4.5 创建反向 shell	54	4.10 攻破 Bash 和 Dash 的保护 机制	89
3.5 针对 PHP 的远程攻击	57	4.11 总结	91
3.6 总结	58	第 5 章 return-to-libc 攻击	92
第 4 章 缓冲区溢出攻击	59	5.1 引言	92
4.1 程序的内存布局	59	5.2 攻击实验: 准备	94
4.2 栈与函数调用	60	5.3 发起 return-to-libc 攻击: 第一部分	96
4.2.1 栈的内存布局	60	5.3.1 任务 A: 找到 system() 函数的 地址	96
4.2.2 帧指针	61	5.3.2 任务 B: 找到字符串 “bin/sh 的 地址	97
4.3 栈的缓冲区溢出攻击	63	5.4 发起 return-to-libc 攻击: 第二部分	99
4.3.1 将数据复制到缓冲区	63	5.4.1 函数的序言	99
4.3.2 缓冲区溢出	64	5.4.2 函数的后记	100
4.3.3 利用缓冲区溢出漏洞	65	5.4.3 函数序言及后记示例	101
4.4 环境准备	67	5.4.4 任务 C	102
4.4.1 关闭地址空间随机化	67	5.4.5 构建恶意输入	103
4.4.2 有漏洞的程序	67	5.4.6 发起攻击	104
4.5 实施缓冲区溢出攻击	68	5.5 返回导向编程	106
4.5.1 寻找注入代码的内存地址	69	5.6 总结	106
4.5.2 提高猜测成功的概率	70		
4.5.3 通过调试程序找到地址	70		
4.5.4 构造输入文件	72		

第 6 章 格式化字符串漏洞	107	7.2 竞态条件漏洞	131
6.1 可变参数函数	107	7.3 实验准备	134
6.1.1 如何使用可变参数	108	7.4 利用竞态条件漏洞	135
6.1.2 printf() 函数如何访问可变 参数	109	7.4.1 选择一个目标文件	135
6.2 如果可变参数不够会出现什么 问题	110	7.4.2 发动攻击	136
6.3 漏洞程序以及实验准备	112	7.4.3 监控结果	137
6.4 利用格式化字符串漏洞	114	7.4.4 运行攻击代码	138
6.4.1 攻击一: 使程序崩溃	114	7.5 防御措施	139
6.4.2 攻击二: 输出栈中的数据	115	7.5.1 原子操作	139
6.4.3 攻击三: 修改内存中的程序 数据	115	7.5.2 重复检查和使用	141
6.4.4 攻击四: 修改程序数据为指 定值	117	7.5.3 粘滞符号链接保护	142
6.4.5 攻击四 (续): 更快的方法	117	7.5.4 最小权限原则	144
6.5 利用格式化字符串漏洞注入 恶意代码	120	7.6 总结	145
6.5.1 有漏洞的程序	120	第 8 章 脏牛竞态条件攻击	147
6.5.2 攻击策略	122	8.1 使用 mmap() 函数进行内存 映射	147
6.5.3 攻击程序	123	8.2 MAP_SHARED、MAP_ PRIVATE 和写时拷贝	149
6.5.4 减少格式化字符串的长度	125	8.3 抛弃复制的内存	150
6.6 防御措施	127	8.4 映射只读文件	151
6.6.1 开发者	127	8.5 脏牛漏洞	153
6.6.2 编译器	127	8.6 利用脏牛漏洞	154
6.6.3 地址随机化	129	8.6.1 选择 /etc/passwd 作为目标 文件	155
6.7 总结	129	8.6.2 设置内存映射和线程	155
第 7 章 竞态条件漏洞	130	8.6.3 写线程	157
7.1 一个常见的竞态条件漏洞	130	8.6.4 madvise 线程	157
		8.6.5 攻击结果	158
		8.7 总结	158

第 2 部分 Web 安全

第 9 章 跨站请求伪造	163	10.2.1 序幕: 注入 JavaScript 代码	181
9.1 跨站请求及其问题	163	10.2.2 通过 XSS 攻击成为他人的 好友	181
9.2 跨站请求伪造攻击	164	10.2.3 使用 XSS 攻击更改他人主页	185
9.3 针对 HTTP GET 服务的 CSRF 攻击	166	10.3 实现自我传播	187
9.3.1 HTTP GET 服务和 POST 服务	166	10.3.1 创建一个自我传播的 XSS 蠕虫: 通过 DOM 方法实现	188
9.3.2 CSRF 攻击的基本思路	166	10.3.2 创建一个自我传播的 XSS 蠕虫: 通过链接方法实现	191
9.3.3 对于 Elgg 的添加好友服务进行 攻击	167	10.4 抵御 XSS 攻击	191
9.4 针对 HTTP POST 服务的 CSRF 攻击	169	10.4.1 去除代码	192
9.4.1 使用 JavaScript 构建 POST 请求	169	10.4.2 用内容安全策略来抵御 XSS 攻击	193
9.4.2 针对 Elgg 的编辑个人资料服务的 攻击	170	10.5 总结	195
9.5 防御措施	173	第 11 章 SQL 注入攻击	177
9.5.1 使用 referer 头	173	11.1 SQL 简略教程	197
9.5.2 同站 cookie	174	11.1.1 登录 MySQL	197
9.5.3 秘密令牌	174	11.1.2 创建数据库	198
9.5.4 案例分析: Elgg 的应对措施	174	11.1.3 创建表	198
9.6 总结	175	11.1.4 插入行	199
第 10 章 跨站脚本攻击	177	11.1.5 SELECT 语句	199
10.1 跨站脚本攻击	177	11.1.6 WHERE 子句	200
10.1.1 反射型 XSS 攻击	178	11.1.7 UPDATE 语句	201
10.1.2 存储型 XSS 攻击	179	11.1.8 SQL 语句中的注释	201
10.1.3 XSS 能造成的破坏	180	11.2 在网络应用程序中与数据库 交互	202
10.2 XSS 攻击实战	180	11.2.1 获取用户数据	202
		11.2.2 从数据库获取数据	203
		11.3 发动 SQL 注入攻击	205

11.3.1 使用 cURL 进行攻击	206	11.5 防范措施	212
11.3.2 修改数据库	207	11.5.1 过滤掉代码和把代码变成数据 ..	212
11.3.3 多条 SQL 语句	208	11.5.2 预处理语句	213
11.4 根本原因	209	11.6 总结	215

第 3 部分 网络安全

第 12 章 数据包嗅探和伪造	219	13.1.3 数据传输的底层原理	250
12.1 数据包是如何被接收的	219	13.1.4 TCP 头部	251
12.1.1 网络接口卡	219	13.2 SYN 泛洪攻击	252
12.1.2 BSD 数据包过滤器	220	13.2.1 三次握手协议	253
12.2 数据包嗅探	222	13.2.2 SYN 泛洪攻击	253
12.2.1 使用 socket 接收数据包	222	13.2.3 进行 SYN 泛洪攻击	254
12.2.2 使用原始套接字进行数据包嗅探	223	13.2.4 编程进行 SYN 泛洪攻击	257
12.2.3 使用 pcap API 实现数据包嗅探	225	13.2.5 防御措施	259
12.2.4 处理捕获的数据包	227	13.3 TCP 复位攻击	259
12.3 数据包伪造	230	13.3.1 关闭 TCP 连接	260
12.3.1 使用 socket 发送正常数据包 ..	231	13.3.2 如何实现攻击	260
12.3.2 使用 raw socket 发送伪造数据包	232	13.3.3 发动 TCP 复位攻击前的准备 ..	261
12.3.3 构造 ICMP 数据包	234	13.3.4 telnet 连接中的 TCP 复位攻击	261
12.3.4 构造 UDP 数据包	236	攻击	263
12.4 嗅探与伪造	238	13.3.6 视频流连接中的 TCP 复位攻击	263
12.5 字节顺序	240	13.4 TCP 会话劫持攻击	265
12.6 计算校验和	242	13.4.1 TCP 会话和会话劫持	265
12.7 总结	244	13.4.2 发动 TCP 会话劫持攻击	266
第 13 章 对 TCP 的攻击	245	13.4.3 TCP 劫持攻击连接的过程	268
13.1 TCP 是如何工作的	245	13.4.4 造成更严重的后果	270
13.1.1 TCP 客户端程序	246	13.4.5 创建反向 shell	270
13.1.2 TCP 服务端程序	247	13.5 总结	272
		第 14 章 防火墙	273

14.1	介绍	273	15.1.2	DNS 区域	298
14.2	防火墙种类	274	15.1.3	权威域名服务器	299
14.2.1	数据包过滤器	275	15.1.4	因特网中区域的组织形式	300
14.2.2	状态防火墙	275	15.2	DNS 请求过程	301
14.2.3	应用防火墙	275	15.2.1	本地 DNS 文件	301
14.3	内核模块	276	15.2.2	本地 DNS 服务器和迭代查询过程	302
14.3.1	编写可加载内核模块	276	15.3	搭建 DNS 服务器以及实验环境	305
14.3.2	编译内核模块	277	15.3.1	配置用户机	305
14.3.3	安装内核模块	278	15.3.2	配置本地 DNS 服务器	306
14.4	使用 Netfilter 搭建一个简单的防火墙	278	15.3.3	在本地 DNS 服务器内配置区域	307
14.4.1	IPv4 中的 Netfilter 钩子函数	279	15.4	DNS 攻击概述	309
14.4.2	实现一个简单的数据包过滤器	280	15.5	本地 DNS 缓存中毒攻击	311
14.5	Linux 中的 iptables 防火墙	282	15.5.1	实现 DNS 缓存中毒攻击	311
14.5.1	iptables 防火墙的结构	283	15.5.2	针对授权部分的攻击	313
14.5.2	遍历链和规则匹配	283	15.6	远程 DNS 缓存中毒攻击	315
14.5.3	iptables 的扩展组件	285	15.6.1	Kaminsky 攻击	315
14.5.4	搭建一个简单的防火墙	286	15.6.2	构建 DNS 回复包的 IP 头和 UDP 头	318
14.6	基于连接跟踪的状态防火墙	289	15.6.3	构建 DNS 头和负载	319
14.6.1	状态防火墙	290	15.6.4	结果验证	320
14.6.2	Linux 的连接跟踪框架	290	15.7	来自恶意 DNS 服务器的回复伪造攻击	320
14.6.3	搭建一个状态防火墙	291	15.7.1	在附加部分伪造数据	320
14.7	应用防火墙及 Web 代理	291	15.7.2	在授权部分伪造数据	322
14.8	绕过防火墙	292	15.7.3	在附加部分伪造与授权部分相关的数据	323
14.8.1	使用 SSH 隧道绕过防火墙	293	15.7.4	在回复部分伪造数据针对 DNS 反向查找	325
14.8.2	动态端口转发	294	15.8	预防 DNS 缓存中毒攻击	327
14.8.3	使用 VPN 绕过防火墙	295	15.8.1	DNSSEC	327
14.9	总结	296	15.8.2	TLS/SSL 解决方案	328
第 15 章	针对域名系统的攻击	297			
15.1	DNS 层次结构、区域、服务器	297			
15.1.1	DNS 域层次结构	297			

15.9 对 DNS 服务器的拒绝服务攻击	329	16.6.2 telnet 测试	353
15.9.1 对 root 和 TLD 服务器的攻击 ..	329	16.7 用 VPN 绕过防火墙	354
15.9.2 对特定域的域名服务器的攻击 ..	330	16.7.1 网络配置	355
15.10 总结	331	16.7.2 搭建 VPN 绕过防火墙	356
第 16 章 虚拟专用网络	332	16.8 总结	357
16.1 虚拟专用网络概述	332	第 17 章 心脏滴血漏洞和攻击	359
16.1.1 虚拟专用网络	333	17.1 心跳协议	359
16.1.2 虚拟专用网络的工作原理	334	17.2 发动心脏滴血攻击	361
16.2 TLS/SSL VPN 原理概述	335	17.2.1 搭建攻击环境	362
16.2.1 建立一个 TLS/SSL 隧道	336	17.2.2 发动攻击	363
16.2.2 转发 IP 数据包	336	17.3 修复心脏滴血漏洞	365
16.2.3 传递 IP 数据包	337	17.4 总结	365
16.3 TLS/SSL VPN 原理的具体细节	338	第 18 章 公钥基础设施	367
16.3.1 虚拟网络接口	338	18.1 攻击公钥加密	367
16.3.2 建立 TUN 接口	339	18.1.1 中间人攻击	367
16.3.3 将数据包路由到 TUN 接口	341	18.1.2 防御中间人攻击	368
16.3.4 TUN 接口的读操作和写操作 ..	342	18.1.3 公钥基础设施	369
16.3.5 在隧道中转发数据包	343	18.2 公钥证书	369
16.3.6 数据包的回程	343	18.2.1 X.509 数字证书	370
16.4 开发 VPN 程序	344	18.2.2 从真实服务器获取证书	371
16.4.1 建立隧道	345	18.3 认证机构	372
16.4.2 监控文件描述符	346	18.3.1 成为认证机构	373
16.4.3 从 TUN 接口到隧道	347	18.3.2 从 CA 获取 X.509 数字证书 ..	374
16.4.4 从隧道到 TUN 接口	348	18.3.3 在网络服务器中部署公钥证书	377
16.4.5 代码汇总	348	18.3.4 使用 Apache 部署 HTTPS	378
16.5 VPN 的网络配置	350	18.4 根与中间 CA	379
16.5.1 实验环境的网络配置	350	18.4.1 根 CA 和自签名证书	379
16.5.2 配置 VPN 服务器	351	18.4.2 中间 CA 与信任链	380
16.5.3 配置 VPN 客户端	351	18.4.3 为中间 CA 制作证书	381
16.5.4 配置主机 V	352	18.4.4 Apache 服务器部署	382
16.6 测试 VPN	352	18.4.5 现实世界中的可信 CA	382
16.6.1 ping 测试	352	18.5 防御中间人攻击	383
		18.5.1 攻击者转发真实证书	383
		18.5.2 攻击者制作假证书	383

18.5.3 攻击者发送自己的证书	384	19.3.1 使用 TLS 记录协议发送数据 ...	399
18.5.4 中间人代理	385	19.3.2 使用 TLS 记录协议接收数据 ...	399
18.6 在公钥基础设施上实施		19.4 TLS 编程: 客户端程序	400
攻击	386	19.4.1 TLS 编程概述	401
18.6.1 对 CA 认证过程的攻击	387	19.4.2 TLS 初始化	401
18.6.2 对 CA 签名过程的攻击	387	19.4.3 TCP 连接设置	403
18.6.3 对算法的攻击	388	19.4.4 TLS 握手	403
18.6.4 对用户确认的攻击	389	19.4.5 应用数据传输	404
18.7 数字证书的类型	390	19.4.6 创建证书的文件夹	405
18.7.1 域名验证型证书	390	19.4.7 完整的 TLS 客户端程序	406
18.7.2 机构验证型证书	390	19.5 校验服务器的主机名	408
18.7.3 扩展验证型证书	391	19.5.1 修改过的客户端代码	408
18.8 总结	392	19.5.2 实验: 中间人攻击	410
第 19 章 传输层安全	393	19.5.3 主机名校验	412
19.1 TLS 概述	393	19.6 TLS 编程: 服务器程序	414
19.2 TLS 握手	395	19.6.1 建立 TLS	414
19.2.1 TLS 握手协议概述	395	19.6.2 TCP 的建立	415
19.2.2 证书验证	396	19.6.3 TLS 握手	416
19.2.3 密钥生成和交换	397	19.6.4 TLS 数据传输	418
19.3 TLS 数据传输	398	19.6.5 测试	419
		19.7 总结	420