文章编号: 2095-6134(2015)03-0384-07

基于模拟攻击的内核提权漏洞自动利用系统*

李晓琦†, 刘奇旭, 张玉清

(中国科学院大学 国家计算机网络入侵防范中心, 北京 101408)

(2014 年 7 月 28 收稿; 2014 年 10 月 13 日收修改稿)

Li X Q, Liu Q X, Zhang Y Q. Automatically exploiting system of kernel privilege escalation vulnerabilities based on imitating attack [J]. Journal of University of Chinese Academy of Sciences, 2015, 32(3): 384-390.

摘 要 针对 Linux 下的内核级提权漏洞,基于模拟攻击的漏洞检测思想,设计并开发漏洞自动利用系统 KernelPET,揭示典型提权漏洞的利用过程,从而为漏洞防御提供支持. KernelPET 系统与主流漏洞库 exploit-db、securityfocus 等衔接,模拟攻击测试近百个提权漏洞,挑选 30 个经典的 Linux 内核提权漏洞载入 KernelPET 漏洞代码库,并基于不同内核、不同发行版的 Linux 平台测试. 实验结果表明,KernelPET 在多类发行版 Linux 系统下具有较好的效果.

关键词 Linux 内核; 提权漏洞; 漏洞利用; 系统安全

中图分类号: TP393 **文献标志码:** A **doi:** 10.7523/j.issn.2095-6134.2015.03.014

Automatically exploiting system of kernel privilege escalation vulnerabilities based on imitating attack

LI Xiaoqi, LIU Qixu, ZHANG Yuqing

(*National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing* 101408, *China*)

Abstract This paper focuses on the Linux kernel-level privilege escalation vulnerabilities. Based on vulnerability detection thoughts of imitating attack, we design and develop an automated privilege escalation vulnerabilities exploiting system KernelPET. It reveals the typical process of exploiting privilege escalation vulnerabilities, and provides support to vulnerabilities defense. KernelPET is developed with today's mainstream vulnerability databases: exploit-db, securityfocus, etc. We test nearly one hundred of privilege escalation vulnerabilities by simulated attack, select 30 classic Linux kernel privilege escalation vulnerabilities, and load them into KernelPET exploiting code libraries. The system is tested on different cores and releases of the Linux platform. Experimental results show that KernelPET runs in multi-class releases of Linux system with good results.

Key words Linux kernel; privilege escalation vulnerabilities; exploits; system security

安全漏洞,是指计算机信息系统在需求、设 计、实现、配置、运行等过程中,有意或无意产生的

* 国家自然科学基金(61272481,61303239)、北京市自然科学基金(4122089)和国家发改委信息安全专项(发改办高技[2012]1424)资助

† 通信作者,E-mail: csu2009@163.com

缺陷[1]. 通过对 2005—2014 上半年间公布的漏洞类型统计(如图 1[2])，其中，获取普通用户权限和获取管理用户权限分别占 45% 和 12%，权限漏洞总数超过所有其他漏洞总数. 从数量变化情况(如图 2[2])看，近年来权限漏洞数量总体上呈明显增长趋势. 鉴于此，本文针对 Linux 下的权限漏洞展开研究. 重点研究 Linux 内核权限提升漏洞，简称“提权漏洞”. 系统内核是保护应用型系统安全的基础，例如以 Linux 为架构的 Android 系统[3]. 这类漏洞属高危级别，攻击者通过利用此类漏洞，可以将自己从 user 级别的用户，提升到拥有 root 权限级别的用户，从而达到完全操控目标主机的目的. 随着当今 Linux 类系统应用日益广泛，Linux 内核提权漏洞的影响力日益显现.

本文主要创新点包括：1) 针对提权漏洞利用的安全检测工具，在国内外多表现为针对特定漏洞的恶意小程序，完整性利用系统研究很少，本次研究的 KernelPET 系统，完整性揭露了 Linux 下漏洞利用流程，从而为漏洞检测和防御提供支持；2) 目前，有个别基于 Metasploit[4] 开发的扩展性插件，多为半自动化工具，本次研究的 KernelPET 系统，实现了无须人工干预的自动化的漏洞利用系统，且系统具有可扩展性；3) 综合性漏洞检测工具，例如 Nmap[5]、Nessus[6]，大都是基于扫描式的，存在一定局限性，本次研究的 KernelPET 系统，基于模拟攻击的漏洞检测思想，漏洞利用更加具有针对性，系统漏洞检测更高效.

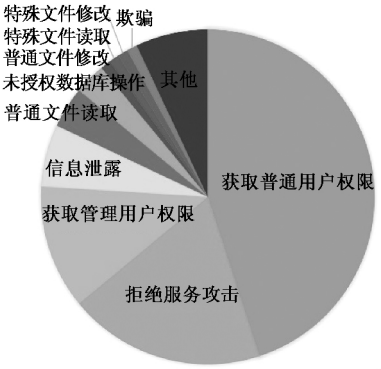

图 1　漏洞类型统计分析

Fig. 1　Statistical analysis of vulnerability types

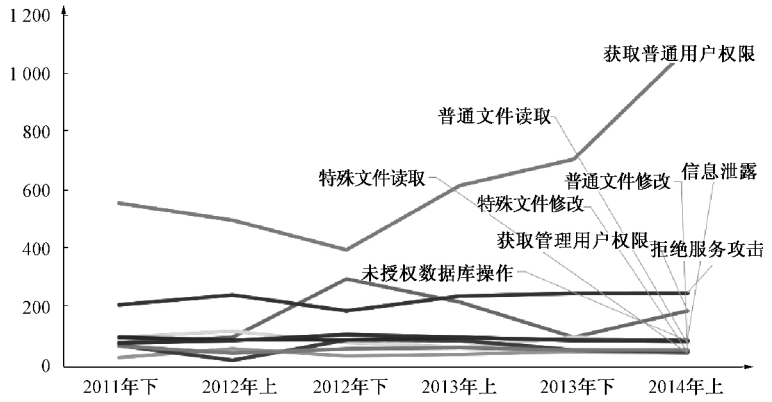

图 2　漏洞数量统计分析

Fig. 2　Statistical analysis of vulnerability quantity

1　研究现状与相关技术

1.1　研究现状

目前，针对 Linux 内核级安全的研究，大多从防御角度出发，主要有：

• 直接修改内核. 由于 Linux 内核开源，可根据特定需求，重写 Linux 内核模块，编译生成新的内核[7].

• 系统内核升级. Linux 内核源码更新速度较快，可以从 Linux kernel 官网 www. kernel. org 下载并编译新内核. 另外，Linux 各个发行版也会有相关补丁发布，及时升级有利于对最新内核级漏洞的防御.

• 系统权限安全管理. 例如通过监控 uid 权限标志位，管理应用程序的运行时权限，防止恶意代码对系统权限实施攻击[8].

• 其他. 例如特权分离技术，使得软件的不同组件运行于不同权限之下[9]；系统函数返回地址随机化处理；关联物理设备的权限改进系统技术；恶意程序检测隔离技术；设置白名单等.

而对于非授权提权攻击技术，主要有以下2种方式:

• 密码获取. 攻击者通过社会工程或暴力破解等手段，获取系统用户名和密码实施提权[8].

• 漏洞利用. 攻击者利用系统安全漏洞，编写漏洞利用代码发起提权攻击[8].

1.2 Metasploit 框架

Metasploit 是一个安全漏洞测试与利用工具，在安全会议 Black Hat 上首次发布. 它最大的特点就是简易，非专业安全人员可以熟练使用这个工具. Metasploit 不仅是一个安全工具，它更创新了一种框架. 它是一个完全开源的框架，任何人都可以开发自己的模块加至其中. 该框架主要用于渗透测试和漏洞测试[10]. 目前，很多国际知名漏洞库，都全面兼容 Metasploit 接口，数据库中的漏洞 exploit 代码可以直接与 Metasploit 衔接，进行特定安全漏洞的测试. 本次系统开发，在 Metasploit 框架漏洞利用思想基础上，加以创新和改进.

1.3 漏洞库

本次设计所分析的漏洞类型，主要来源于 exploit-db 漏洞库. exploit-db 漏洞库除具备 CVE 编码之外，还有自己的 EDB 编码体系，便于查询与检索，也便于国际化标准的统一. exploit-db 主要面向对象是安全漏洞研究人员和网络安全爱好者. 此外，为了漏洞信息的详尽性与准确性，我们用到多个其他漏洞库，如表1所示.

表1 本文涉及漏洞库列表

Table 1 Vulnerability databases involved in this paper

漏洞库名称	地址
国家安全漏洞库[11]	www. nipc. org. cn
exploit-db 漏洞库[12]	www. exploit-db. com
中国信息安全国家漏洞库[13]	www. cnnvd. org. cn
security 漏洞数据库[14]	www. security-database. com
securityfocus 漏洞平台[15]	www. securityfocus. com
绿盟漏洞数据库[16]	www. nsfocus. net/vulndb
freebuf 安全社区[17]	www. freebuf. com
美国国家漏洞数据库[18]	web. nvd. nist. gov
国家信息安全漏洞共享平台[19]	www. cnvd. org. cn

2 KernelPET 设计思路

2.1 基于模拟攻击的漏洞检测思想

传统的扫描式漏洞检测工具，试图通过搜集系统信息发现漏洞，但由于系统补丁等原因，往往难以准确对应相关漏洞，成功率不高[20]. 本文采用基于模拟攻击的漏洞检测思想，思想流程如图3所示，假若模拟攻击成功，则将系统与该漏洞匹配，可以准确判断目标系统是否存在某漏洞.

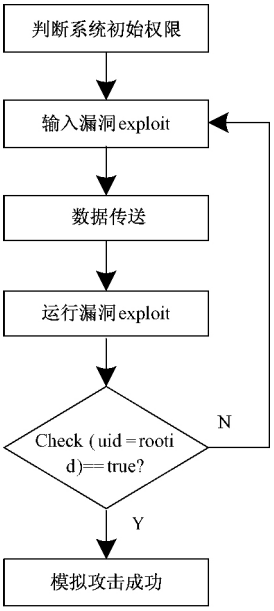

图3 模拟攻击思想流程

Fig. 3 Process of imitating attack thought

具体算法框架实现如下所示:

```
Process process;
process = Runtime. getRuntime ( ). exec ( "whoami"); //判断初始权限
InputStreamReader ir = new InputStreamReader(process. getInputStream( ) ); //建立输入流
LineNumberReader input = new LineNumberReader( ir); //数据传送流
String su = input. readLine( );
int num = jComboBox. getItemCount( );
for ( int i = 1; i < num; i + + ) {
jComboBox. setSelectedIndex( i);
jButton. doClick( ); //运行特定 exploit 代码
if ( su. equals( "root") ) {
break; //提权成功后退出} }
```

2.2 功能实现描述

KernelPET 系统采用 MVC 架构模式，细化成4个包文件 KernelPETController、ExploitSrc、KernelExecshell、KernelPETView.

• KernelPETController 包下主要分为4个基

类：ExploitTest 类接受用户的漏洞选择，通过处理反馈给其他类，从而调用后台漏洞 exploit 库，调取用户所需的 exploit 代码. ExportEpinfo 类响应一切输出请求，包括不同模块的输出信息，例如利用流程显示、漏洞关键信息提示等. InstantRoot 类在一键式提权请求下响应，通过综合策略式读取 exploit code 库，通过模拟攻击测试，将最终反馈给用户一个 root/user shell. ReportTopdf 类则接受用户的报告请求，接下来会执行一系列的 shell 交互式操作，获取本次漏洞利用的一系列综合信息，从而生成漏洞利用分析报告.

• ExploitSrc 包主要任务是与后台 exploit code 数据库交互.

• KernelExecshell 包主要负责与 Linux Shell 的一些数据交互工作.

• KernelPETView 包下包含了各种 Form 类，负责系统的 GUI，这其中运用了 JavaSwing 类库组件，使界面更人性化，增强了用户体验.

系统整体功能实现如图 4 所示. Exploit 实验成功再现的 30 个利用代码组成系统漏洞 exploit code 库. 漏洞测试模块响应用户需求，进行 exploit 代码的选择性读取，而一键提权模块则进行自动化的策略式读取，快速定位针对当前平台的可利用攻击代码. 然后，KernelPET 定位到漏洞编号，针对读取的 exploit 代码进行模拟攻击测试，进而判断终态 uid 参数，确定是否成功提权. 成功之后，系统将返回用户 RootShell，继而用户可以完全性操控本地操作系统. 随后，系统通过 Linux Shell 交互器完成本次提权的综合信息提取，包括判断 OS 类型、判断内核版本、判断终态 uid 参数、加工利用流程等，最后经报告生成器处理生成漏洞利用报告.

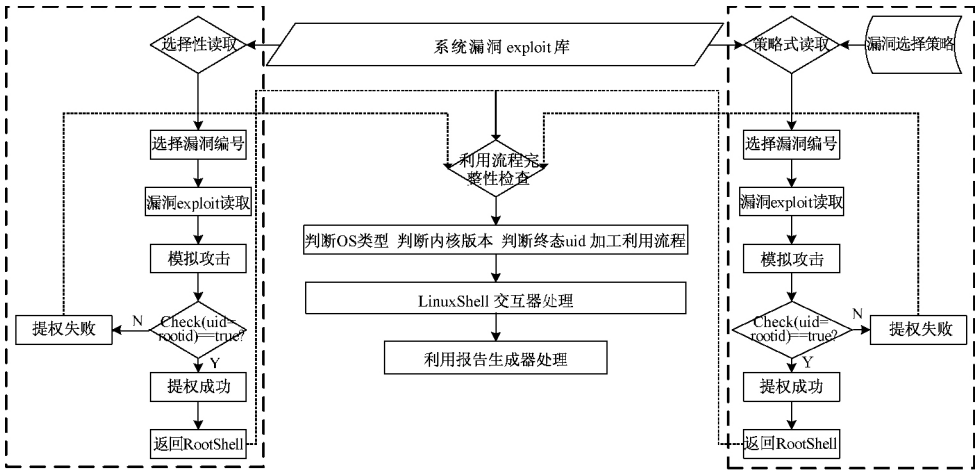

图 4　系统整体功能实现图

Fig. 4　Diagram of overall function of the system

2.3　KernelPET 核心模块描述

1) KernelPET GUI

该系统 GUI 设计采用 JavaSwing 可编程渲染模型，为保证一键提权的连续性与简洁性，即尽量保证无需中间性干预式提权，因此设计了单层次为主的 GUI. 系统采用 Panel、ToolBar、MenuBar、MenuItem、TextField、TextArea、TextPane 等多种屏幕显示元素，在保证功能的前提下，尽量使系统简单易用. 系统整体 GUI 主要分以下部分：

• 漏洞测试模块，包括测试模块 Label、漏洞选择 ToolBar、漏洞执行 Button.

• 漏洞利用报告生成模块，包括漏洞报告生成 Item、漏洞报告查看 Item.

• 漏洞利用代码执行流程显示模块，实时显示利用执行过程的运行时态.

• 漏洞代码执行信息显示模块，包括漏洞 CVE/EDB Item、系统内核版本 Item、系统发行版本 Item、当前用户态信息 Item、其他信息 TextArea.

• 漏洞选择模块，该模块与漏洞库信息同步衔接，便于系统扩展.

• 漏洞执行模块，该模块与 Linux shell 接口交互，通过 shell 命令解释器执行漏洞利用代码.

• 一键提权功能模块，该模块综合利用其他模块，实现模拟攻击等系统核心功能.

系统的模块功能组织如图 5 所示.

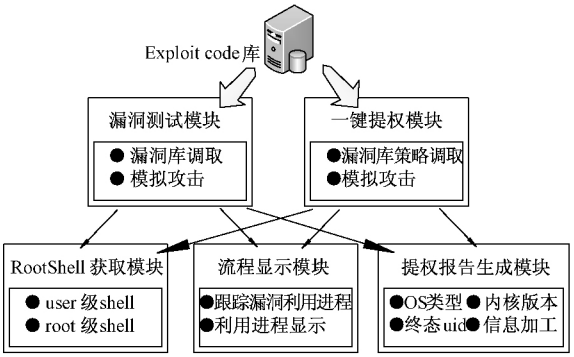

图5 模块功能组织

Fig. 5 Module functional organization

2) **漏洞测试模块**

漏洞模拟攻击测试模块实现与后台 exploit code **库的同步，针对当前的漏洞库，用户可以选择特定漏洞利用执行代码. 该模块包括测试模块** Label、**漏洞选择** ToolBar、**漏洞执行** Button. **用户通过** ToolBar **选择漏洞** CVE/EDB **编码. 其中各个漏洞** exploit **代码均统一接口，全部按照** CVE **和** exploit-db **标准编号进行组织，在接口统一、易扩展的同时，与国际漏洞描述标准接轨，方便用户在网络漏洞库平台进行漏洞信息检索.**

3) RootShell **获取模块**

RootShell **获取模块的设计原则是: 给用户最简洁的场景，最大化的空间. 本着这个原则该系统并没有设计特定的功能扩展，而是直接返回用户一个** RootShell **窗口，让用户在提权之后尽情发挥. 众所周知，当用户获取一个** root **级别的** shell **后，**shell **命令解释器可谓是万能的.** KernelPET **正是如此，当漏洞模拟攻击成功之后，会返回用户一个** root **级别的** shell **解释器窗口，供用户自由发挥.**

4) **一键提权模块**

一键提权模块综合利用其他模块，实现系统的核心功能: 一键式权限提升. 该模块设计的初衷是自动化，使得用户可以抛开复杂因素，只需简单一次按键即可实现权限提升. 该模块通过一定的流程算法，综合利用 漏洞选择模块、漏洞执行模块、漏洞信息显示模块、漏洞利用流程信息显示模块，通过在后台执行多步操作，实现一键式权限提升.

3 KernelPET 评估实验

3.1 总体运行评估

一键提权功能的实现，实际上是对 exploit code **库中的漏洞利用代码，进行策略式模拟攻击测试，直到发现可以提权成功的漏洞，亦或由于测试完所有代码而以失败告终. 由于每一个漏洞利用代码均有其统一的接口和完整的利用测试过程，每进行一次模拟攻击测试后，均会进一步生成当前用户态信息、系统内核信息、系统发行版信息、漏洞编号等，根据这些信息可以判断该漏洞的利用价值，评测当前系统安全性. 当得到一个** root **用户态** shell **后，程序自动退出漏洞利用代码的测试，用户可以在** root shell **下进一步拓展自己的工作. 为增强系统健壮性，设置漏洞利用失败响应机制，进一步显示提权失败信息. 无论一次成功的漏洞利用，还是一次失败的利用，均有其价值，这对于系统安全性测试将有益处. 系统运行效果如图** 6 **所示.**

系统测试报告的生成，采用 Java iText **组件技术，在** KernelPET **运行到任何一个系统间隔时，用户可以选择生成测试报告. 此时，系统就会将当前运行测试的模拟攻击信息，通过** PdfWriter IO **流写入到后台** PDF **文档中，用来记录系统测试综合信息. 该系统设计为单一文档书写器机制，系统设置了一个** PDF **白板，系统可以反复读写该白板，若用户需要长时间保存报告，可以选择加载功能. 这样，既减少数据冗余，精简系统复杂度，同时最大程度方便用户使用.**

为了进一步测试系统的健壮性和跨平台稳定性，将该系统放到不同内核、不同发行版的 Linux **平台下测试. 其中，涉及** Ubuntu、Redhat、Fedora、CentOS、OpenSUSE **等众多发行版** Linux. **而在同一发行版** Ubuntu **下，通过不同内核版本的转换，同样进行了测试. 总体测试效果如表** 2 **所示.**

表2 KernelPET 跨平台测试

Table 2 Cross-platform test of KernelPET

序号	测试环境	root
1	Ubuntu9. 04 + Kernel2. 6. 28	✓
2	Ubuntu13. 04 + Kernel3. 8. 0	✓
3	Ubuntu10. 10 + Kernel2. 6. 35	✓
4	Ubuntu12. 10 + Kernel3. 2. 28	✓
5	Redhat5 + Kernel2. 6. 18	✓
6	Fedora16 + Kernel3. 1. 0	✓
7	CentOS5. 6 + Kernel2. 6. 38	✓
8	OpenSUSE12. 1 + Kernel3. 1. 0	✓

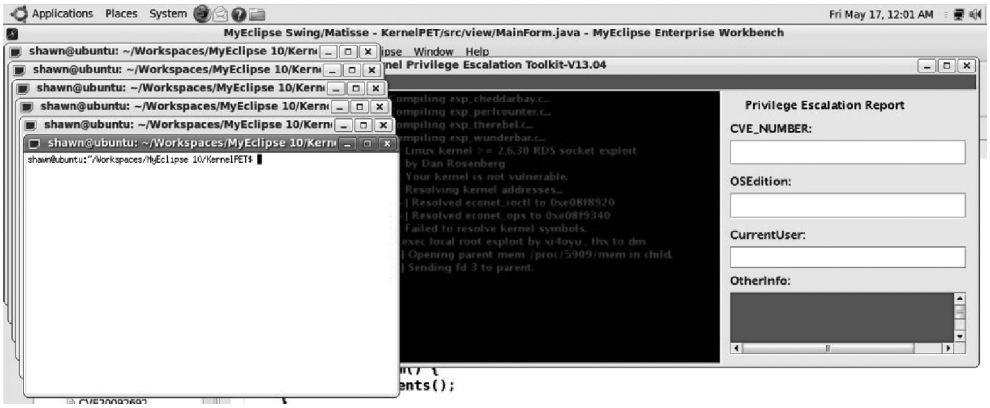

图 6　一键提权运行效果

Fig. 6　Effects of automatical privilege escalation

3.2　基于 KernelPET 的系统安全测试

KernelPET 的实现,对于 Linux 平台安全性测试有一定价值,有利于检测 Linux 下存在的一些典型漏洞,进而对这些漏洞进行跟踪分析,可以了解它们的特性以避免在系统中再次产生类似漏洞,通过对这些类型的漏洞进行预测挖掘,使我们能积极地防御黑客的攻击破坏.

1) 测试概述

根据 Linux 内核版本与发行版本的关系,内核低层级漏洞与发行版本关系不大,主要侧重内核版本. 不过内核应用级漏洞则与发行版存在着必然的联系. 综合考虑,我们采取以内核版本为主线,兼顾不同发行版之间的差异,进行全面的安全测试.

测试实验采用 VMware Workstation 10.0 虚拟平台,操作系统以 Ubuntu 发行版为主线,版本跨度从 Ubuntu5.10 到 Ubuntu14.04. 同时,兼顾考虑 32 位与 64 位的差异,保证全面性. 测试在以 Ubuntu 发行版为主的同时,涉及 CentOS、Fedora、OpenSUSE、Red Hat、Ubutnukylin 等众多发行版本,共涉及 27 种不同的系统平台,进一步展现了 Linux 内核级漏洞的广泛性. 部分检测结果如表 3 所示. 下面将介绍一个典型漏洞的检测情况.

表 3　成功检测漏洞列表(部分)

Table 3　List of successfully detected vulnerabilities (in part)

序号	检测平台	发现漏洞编号	漏洞信息	root
1	Ubuntu9.10	CVE-2010-4258	Linux 内核本地地址限制覆盖安全弱点漏洞	✓
2	Ubuntu9.04	CVE-2009-2692	sock_sendpage( ) 模块空指针引用漏洞	✓
3	Red Hat Server5.0	CVE-2009-3547	pipe.c 函数空指针引用权限提升漏洞	✓
4	Ubuntu10.04	CVE-2012-0056	/proc/ $ pid/mem 错误本地权限提升漏洞	✓
5	Ubuntu10.10	EDB-15916	内核 CAP_SYS_ADMIN 模块提权漏洞	✓
6	Ubuntu12.10	CVE-2013-1763	Linux 内核本地权限提升漏洞	✓
7	Ubuntu13.04	CVE-2011-1485	守护进程当地竞争条件漏洞	✓
8	Fedora16	CVE-2012-0809	标准格式串设计错误权限提升漏洞	✓
9	CentOS6.3	CVE-2013-2094	边界条件错误本地权限提升漏洞	✓

2) CVE-2010-4258 漏洞检测

①概述

在 Linux 下,用户经常须传递一个指针给内核,内核正常情况下读写该指针,进行下一步操作. 其中系统内核函数 access_ok 会对用户指针进行安全检查[15]. 但是,由于 Linux 内核庞大且功能复杂,在某些时候,内核必须停止 access_ok 的这种检查行为,通常由 get_fs( ) 和 set_fs( ) 来完成停止功能,使得函数 access_ok( ) 失效. 当进程运行出错,调用内核 BUG( ) 函数,亦或类似的其他操作时,内核将调用 do_exit( ) 函数结束当前进程,此时 set_fs( ) 将被覆盖. clear_child_tid( ) 函数将特定地址写零,以便通知各线程失效处理. 此过程中,将执行如下操作:

$$put\_user(0, tsk->clear\_child\_tid)$$

当内核执行 get_fs( ) = = KERNEL_DS 操作

时，将允许指向内核地址的读写请求，达到写入内核态内存的攻击目标.

②KernelPET 检测分析

由于 CentOS 默认不支持 Econet 协议，所以测试没有通过，利用流程信息如下所示.

[*] *Failed to open file descriptors*

在支持 Econet 协议的 Ubuntu9.10 上测试通过，并返回用户 RootShell，利用流程信息如下所示.

[*] *Resolving kernel addresses*…

[+] *Resolved econet_ops to* 0*xe*0883360

[+] *Resolved commit_creds to* 0*xc*01626*b*0

[+] *Resolved prepare_kernel_cred to* 0*xc*01628*b*0

[*] *Calculating target*…

[*] *Triggering payload*…

[*] *Got root*!

成功提权之后，有必要生成漏洞分析报告，进一步严谨分析该漏洞的表现范围和修复方案. 表4 列出了 CVE-2010-4258 的影响系统与主要修复方案.

表4 影响系统与修复方案

Table 4 Affected system and program repair

影响系统	所有 Kernel < =2.6.37 且支持 Econet 协议的 Linux 操作系统
应急方法	当计算资源有限时，简单把支持网卡 Econet 协议的内核模块删除或改名即可
裁剪内核	重新配置和编译内核
内核修补	下载并安装最新版本内核

其中，裁剪内核方案需要耗费相当多的计算资源，对于独立主机比较适合. 对于内核提权这类比较特殊的漏洞，最佳方案是内核修补更新方案，这样更有利于系统稳定性，同时避免二次漏洞暴露.

4 结束语

Linux 的权限控制机制，是其一切安全策略的根基. 通过分析典型的内核漏洞可以看到 Linux 并不是完美的，还有很多地方需要完善. 有些漏洞极大地影响了 Linux 的推广和使用，极易被攻击者进行权限攻击. 本文通过进行 Linux 下漏洞 exploit 实验，成功再现了 30 个可以实现权限提升的内核级漏洞. 进而设计并开发了减少人工干预的一键式提权系统 KernelPET.

参考文献

[1] 刘奇旭，张玉清，宫亚峰，等. 安全漏洞标识与描述规范的研究[J]. 信息网络安全，2011，7：4-6.

[2] 绿盟公司. 绿盟漏洞威胁态势报告[R]，北京：绿盟科技，2013[2014-07-23]. http://www.nsfocus.com.cn/4_research/4_6.html.

[3] Chen H G，Mao Y D，Wang X，et al. Linux kernel vulnerabilities：State-of-the-art defenses and open problems[C]. Proceedings of the Second Asia-Pacific Workshop on Systems，ACM，2011.

[4] HD M，Spoon M，James L，et al. Metasploit[CP/OL]. 2014[2014-07-23]. http://www.metasploit.com.

[5] Gordon L. Nmap[CP/OL]. 2014[2014-07-23]. http://nmap.org.

[6] Renaud D，Ron G. Nessus[CP/OL]. 2014[2014-07-23]. http://www.tenable.com/products/nessus.

[7] Nimbalkar R，Patel P，Meshram B. Advanced linux security[J]. Editorial Board，2013，2(3)：7-12.

[8] Treaster M，Koenig G A，Meng X，et al. Detection of privilege escalation for linux cluster security[C]. Proceedings of the 6th LCI International Conference on Linux Clusters. 2005.

[9] Provos，Markus F，Peter H，et al. Preventing privilege escalation[C]. Proceedings of the 12th USENIX Security Symposium，2003.

[10] O´Gorman J，Kearns D，Aharoni M. Metasploit：The Penetration Tester´s Guide[M]. San Francisco：No Starch Press，2011.

[11] 国家计算机网络入侵防范中心. 国家安全漏洞库[DB/OL]. 2014[2014-07-23]. http://www.nipc.org.cn/.

[12] Offensive Security Team. Exploit database[DB/OL]. 2014[2014-07-23]. http://www.exploit-db.com.

[13] 中国信息安全测评中心. 中国信息安全国家漏洞库[DB/OL]. 2014[2014-07-23]. http://www.cnnvd.org.cn.

[14] Security-Database Company. Security vulnerability database[DB/OL]. 2014[2014-07-23]. http://www.security-database.com.

[15] Symantec Company. Securityfocus vulnerability database[DB/OL]. 2014[2014-07-23]. http://www.securityfocus.com/bid.

[16] 绿盟公司. 绿盟漏洞数据库[DB/OL]. 2014[2014-07-23]. http://www.nsfocus.net/vulndb.

[17] Freebuf Team. Freebuf 安全社区[EB/OL]. 2014[2014-07-23]. http://www.freebuf.com.

[18] National Institute of Standards and Technology. American national vulnerability database[DB/OL]. 2014[2014-07-23]. http://web.nvd.nist.gov.

[19] 国家计算机网络应急技术处理协调中心. 中国国家信息安全漏洞共享平台[DB/OL]. 2014[2014-07-23]. http://www.cnvd.org.cn.

[20] Cobb C，Cobb S，Kabay M，et al. Penetrating computer systems and networks[M]. Computer Security Handbook，2012.