组　员	柴宇龙，林坤，霍南风，王春萍	指导教师	郭燕

一、课题的名称、来源

a.课题名称	**linux 系统下 rootkit 工具集的开发**
b.课题来源	☐生产 ■科研 ☐教学 ☐其他

二、前期调研和准备

1.摘要

rootkit 工具集通常包括一些常用木马程序套件，用来建立后门和隐藏行踪，包括为攻击者提供后门的特洛伊木马程序、隐藏攻击者的目录和进程的程序和一些日志清理工具等。攻击者通过使用 rootkit 工具集可以隐藏自己的踪迹并且保留 root 权限。本次工程实践提供了一种劫持系统调用的内核级 rootkit 实现方法，通过实现该 rootkit 可以更深刻地理解 Linux 系统调用机制和 rootkit 工作原理，为下一步检测和防范内核级 rootkit 打下基础。

关键词：Linux；内核级 rootkit；系统调用

2.选题依据

计算机技术的发展不仅给人们带来越来越多的便利，也使信息安全问题越显严峻，从操作系统漏洞到网银密码被窃取，计算机环境似乎越来越危险。人们不断研究新的信息安全防范机制和技术，然而，针对计算机系统和网络的攻击技术却日趋成熟、难以检测和防御。rootkit 工具集就是一种越来越具威胁性的攻击方式。所以为了能更好地检测和防范 rootkit 的攻击，研究其实现原理很有必要。

rootkit 最早出现于 20 世纪 90 年代初,是攻击者在攻击 Unix/ Linux 时用来隐藏自己的踪迹和保留 root 访问权限的工具。rootkit 的目的不是为了突破系统获得进入系统的权限,而是通过其他手段获得系统权限后对所得到的权限进行保存,同时隐藏攻击活动的痕迹,防止暴露，所以其攻击令人难以察觉。

rootkit 分为应用级 rootkit 和内核级 rootkit 两种类型(罗粮 等,2007)。应用级 rootkit 在操作系统应用层修改系统文件,比较容易预防和检测。内核级 rootkit 攻击操作系统内核,与应用级 Rootkit 相比功能更为强大,更难检测。因此通过对内核级 rootkit 的实现方式进行研究以便找到更有效的检测和防范机制。

3.课题内容与具体方法

3.1 课题内容

实现基于 LKM 的内核级 rootkit 工具集，以达到隐藏踪迹和保留 root 访问权限的目的。该 rootkit 工具集工作在 Linux 2.6.x 内核版本下。

LKM(Loadable Kernel Modules)是 Linux 内核为了扩展其功能所使用的可加载内核模块。LKM 的优点：动态加载，无须重新实现整个内核。基于此特性，LKM 常被用作特殊设备的驱动程序（或文件系统），如声卡的驱动程序等等。

该内核级 rootkit 利用 Linux 的 LKM 机制将模块挂在系统模块链上，成为内核代码运行于内核态，享有内核的所有特权。然后通过重定向或修改系统调用表实现将一些系

统调用函数重定向到 rootkit 提供的相应系统调用函数，从而达到隐藏自己的目的。

3.2 设计方案

3.2.1 模块划分

整个 rootkit 分为 8 个模块，如图 1：

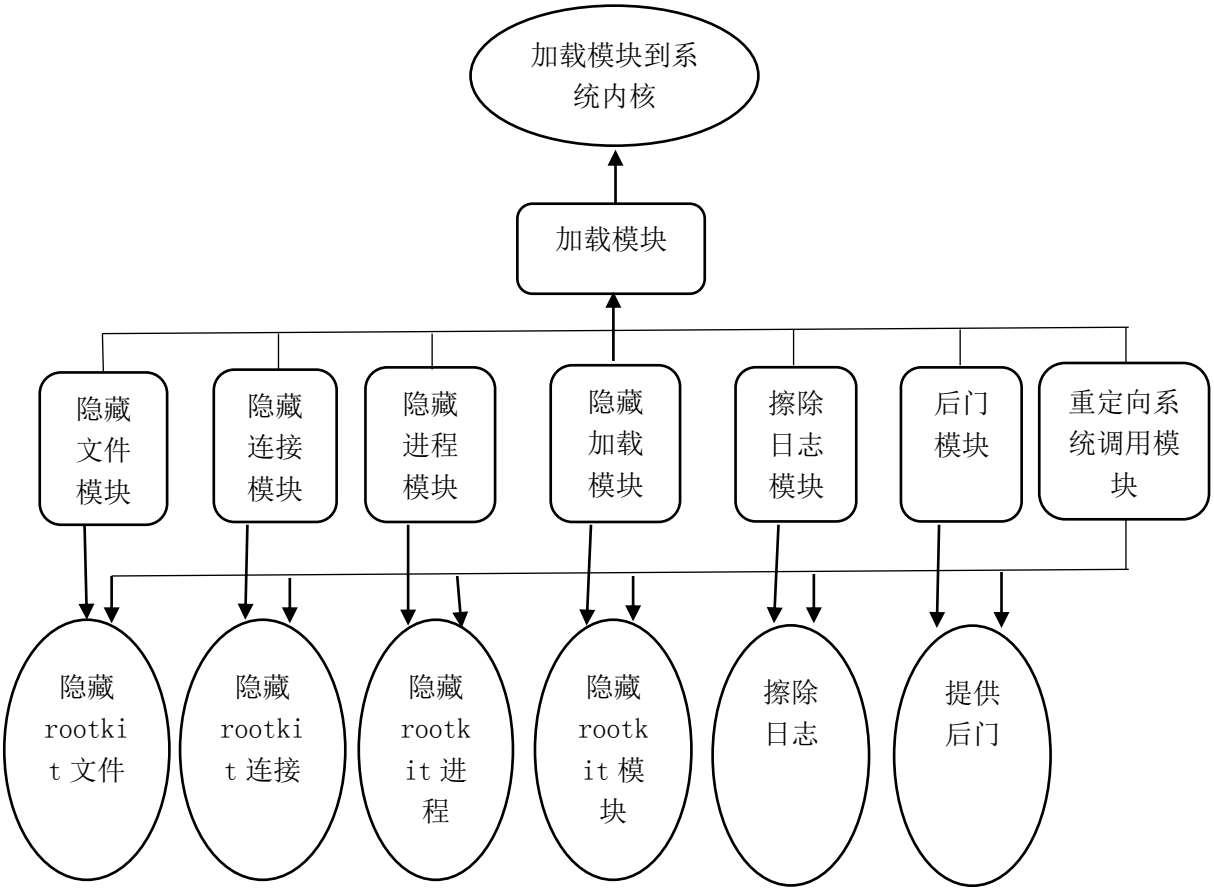

图 1. rootkit 各模块关系图

(1)加载模块

该模块是一个 Linux 下的动态加载模块，实现将 rootkit 加载到 Linux 内核中，加载成功后将创建一个隐藏目录，该目录下存放 rootkit 其他所有文件。

(2)隐藏文件模块

该模块在系统中隐藏 rootkit 所有文件,使被攻击者不能通过系统命令(如 ls)搜索发现 rootkit 文件。

(3)隐藏连接模块

该模块隐藏 rootkit 的网络连接,防止使用 netstat 命令看到 rootkit 的远程连接。

(4)隐藏进程模块

该模块隐藏 rootkit 运行时创建的进程，使被攻击者不能通过查看正在运行的进程发现 rootkit。

(5)隐藏加载的 rootkit 模块

当被攻击者使用 lsmod（list modules 的缩写，列出所有模块）查看系统模块时不能发现刚才加载进内核中的 rootkit 模块。

(6)擦除日志模块

该模块删除系统日志中记录的 rootkit 相关信息。

(7)后门模块

该模块提供本地的提权后门和网络的监听后门。

(8)重定向系统调用模块

该模块实现系统调用重定向。当被攻击者使用系统命令(如 ls)查看系统相关信息时，这些系统命令中的系统调用被重定向到 rootkit 提供的修改过的系统调用版本，在这些修改的系统调用版本中屏蔽 rootkit 相关信息，从而达到隐藏自己的目的。模块(2)到(7)通过调用该模块实现各自的隐藏功能。

3.2.2 方案分析

该方案提供了一种内核级 rootkit 的实现，其中的重定向系统调用模块涉及到 Linux 系统调用机制的工作原理，其他模块涉及到系统调用的实现，所以需要对 Linux 内核的有所了解。

这种内核级 rootkit 不修改系统文件，所以规避了文件完整性的检测。比较好的检测方法需要对比当前系统调用函数表和原始系统函数调用表或对系统内核进行监测，实时监测异常模块。由于内核级 rootkit 攻击方式的多样，很难提供高效统一的检测机制，所以从一开始就防止 root 权限的被攻击获取从而避免 rootkit 的安装可能是一种最好的防御机制。

3.3 技术路线

实现 rootkit 各模块的具体方法如下:

(1)加载模块

该模块按照 Linux 动态加载模块的结构编写，通过命令”insmod xxx”将模块 xxx 加载到内核空间中的模块链。

(2)隐藏文件模块

分析系统命令 ls，find，du 中查看文件和目录信息的系统调用函数，在被攻击者使用这些系统命令查看文件信息时重定向用到的系统调用函数(如 sys_read())到修改过的系统调用(如 my_sys_read())中，在 my_sys_read()中过滤掉 rootkit 文件，从而使系统命令(如 ls)显示不出 rootkit 相关文件信息。

(3)隐藏连接模块

分析系统命令 netstat 中查看网络连接信息和端口号的系统调用函数，然后通过模块(8)重定向到提供屏蔽 rootkit 网络连接信息的系统调用版本。

(4)隐藏进程模块

分析系统命令 ps，top 中查看进程信息的系统调用函数，提供屏蔽 rootkit 进程信息的系统调用版本。

(5)隐藏加载的 rootkit 模块

将加载的 rootkit 模块从内核模块链表中摘除。最新加载的模块总是在模块链表的表头，因此可以在加载完 rootkit 模块后再加载一个清除模块将 rootkit 模块信息从链表中删除，再退出清除模块。比较新的内核版本中也可以通过判断模块信息后直接 list_del。

(6)擦除日志模块

通过修改例如/var/log/security 等文件，清楚用户登陆信息。

(7)后门模块

提供具有后门的 login，chfn 和 chsh，可以使 rootkit 获得 root 权限并支持远程 root 登录。

(8)重定向系统调用模块

该模块利用以下方法获得系统调用表：

a.通过指令如 SIDT 得到中断描述符表寄存器 IDTR。

b.从中断描述符表寄存器 IDTR 得到中断描述符表 IDT 的地址。

c.从中断描述符表 IDT 中得到 system_call 的地址。

d.由于在 system_call 中会调用 system_call_table，所以可以遍历其指令得到 system_call_table 的地址。

e.直接将 system_call_table 中的系统调用重定向到相应的 rootkit 提供的系统调用。

最后修改系统调用表中相应系统调用函数为其他模块提供的修改版本的系统调用函数，实现系统调用的重定向。

3.4 技术关键和难点

（1）实现系统调用函数的替代版本。需要分析系统命令，如 ls，找到实现其功能对应的系统调用函数，系统调用函数涉及到 Linux 内核的一些操作机制，需要提前调研。

（2）实现重定向系统调用模块。由于需要对系统内核进行操作，一旦出错，可能造成系统崩溃，很难调试 bug。

3.5 备用方案

实现重定向系统调用不止一种方式，除了上面提到的方法，还可以通过修改系统调用入口函数或修改中断描述符表实现(石晶翔 等，2010)。为了确保重定向系统调用模块的顺利实现，下面提供一种备用方案，即修改系统调用入口函数实现系统调用重定向。

具体方法为：用户执行的系统调用通过中断指令陷入内核态，操作系统接收到中断请求后找到中断服务例程 system_call。system_call 根据系统调用号在系统调用表中找到相应的系统调用函数后执行。在 system_call 中执行指令 call *sys_call_table

前，增加对系统调用号的判断，如果是需要拦截的目标调用号的话，就转向执行 rootkit 指定的代码；如果不是，则继续执行原来的代码。

4.**预计成果**

研究成果的主要输出：

（1）前期调研形成的一些分析文档。

（2）每周在规定的时间进行项目讨论，以周记形式记录。

（3）形成其他文档，如开题报告、结题报告等。

（4）最终实现的 rootkit 工具集。

三、工作进度的大致安排

项目阶段	工作内容	时间区间
组队选题环节	完成组队、选题	第一学期 2~5 周
开题环节	输出开题报告、开题答辩 ppt	第一学期 5~12 周
详细设计环节	完成项目详细设计，输出对应文档，进入编码阶段	第一学期 12~19 周
中期检查环节	完成中期检查报告和中期答辩 ppt	第二学期 1~3 周
系统实现环节	完成系统开发、测试，输出相应文档	第二学期 4~11 周
结题环节	完成结题报告、结题答辩 ppt	第二学期 12~14 周

四、团队分工

姓名	学号	分工
柴宇龙	SA13226169	加载模块，重定向系统调用模块，隐藏文件模块
林坤	SA13226373	隐藏进程模块，隐藏连接模块
霍南风	SA13226115	隐藏加载模块，后门模块，Knark 的分析
王春萍	SA13226132	擦除日志模块，rootkit 的检测

五、主要参考文献

[1] 罗粮，周熙.2007.RootKit 在 Linux 下的工作原理及其检测[J].计算机安全 www.nsc.org.cn，2007.03：19-21.

[2] 石晶翔，陈蜀宇，黄晗辉. 2010.基于 Linux 系统调用的内核级 Rootkit 技术研究

计算机技术与发展，20(4)：175-178.
六、指导教师意见 指导教师签名：　　日期：　　年　　月　　日