如何增强 Linux 系统的安全性，第一部分: Linux 安全模块（LSM）简介

未来的标准：**Linux**内核的通用安全支持框架

Linux安全模块（LSM）是Linux内核的一个轻量级通用访问控制框架。本文介绍Linux安全模块（LSM）的相关背景，设计思想，实现方法；并说明如何使用Linux安全模块（LSM）来增强Linux系统的安全性：一方面是供内核开发人员和安全研究人员使用的接口，另一方面是供普通用户使用的模块，以及具体的使用方法。如果读者具有Linux内核和安全的相关背景知识，可以有助于对本文的理解；如果不具有，可以先阅读本文最后参考资料中列出的IBM dW上的三篇文章。

赵亮，南京大学计算机系硕士研究生，研究方向：安全操作系统。 电子邮箱： zhao_liang@myway.com，欢迎讨论Linux内核和安全的相关问题。

2003 年 7 月 26 日

1．相关背景介绍：为什么和是什么

近年来Linux系统由于其出色的性能和稳定性，开放源代码特性带来的灵活性和可扩展性，以及较低廉的成本，而受到计算机工业界的广泛关注和应用。但在安全性方面，Linux内核只提供了经典的UNIX自主访问控制（root用户，用户ID，模式位安全机制），以及部分的支持了POSIX.1e标准草案中的capabilities安全机制，这对于Linux系统的安全性是不足够的，影响了Linux系统的进一步发展和更广泛的应用。

有很多安全访问控制模型和框架已经被研究和开发出来，用以增强Linux系统的安全性，比较知名的有安全增强Linux（SELinux），域和类型增强（DTE），以及Linux入侵检测系统（LIDS）等等。但是由于没有一个系统能够获得统治性的地位而进入Linux内核成为标准；并且这些系统都大多以各种不同的内核补丁的形式提供，使用这些系统需要有编译和定制内核的能力，对于没有内核开发经验的普通用户，获得并使用这些系统是有难度的。在2001年的Linux内核峰会上，美国国家安全局（NSA）介绍了他们关于安全增强Linux（SELinux）的工作，这是一个灵活的访问控制体系Flask在Linux中的实现，当时Linux内核的创始人Linus Torvalds同意Linux内核确实需要一个通用的安全访问控制框架，但他指出最好是通过可加载内核模块的方法，这样可以支持现存的各种不同的安全访问控制系统。因此，Linux安全模块（LSM）应运而生。

Linux安全模块（LSM）是Linux内核的一个轻量级通用访问控制框架。它使得各种不同的安全访问控制模型能够以Linux可加载内核模块的形式实现出来，用户可以根据其需求选择适合的安全模块加载到Linux内核中，从而大大提高了Linux安全访问控制机制的灵活性和易用性。目前已经有很多著名的增强访问控制系统移植到Linux安全模块（LSM）上实现，包括POSIX.1e capabilities，安全增强Linux（SELinux），域和类型增强（DTE），以及Linux入侵检测系统（LIDS）等等。虽然目前Linux安全模块（LSM）仍然是作为一个Linux内核补丁的形式提供，但是其同时提供Linux 2.4稳定版本的系列和Linux 2.5开发版本的系列，并且很有希望进入Linux 2.6稳定版本，进而实现其目标：被Linux内核接受成为Linux内核安全机制的标准，在各个Linux发行版中提供给用户使用。

2．设计思想介绍：得让两方面都满意

Linux安全模块（LSM）的设计必须尽量满足两方面人的要求：让不需要它的人尽可能少的因此得到麻烦；

同时让需要它的人因此得到有用和高效的功能。

以Linus Torvalds为代表的内核开发人员对Linux安全模块（LSM）提出了三点要求：

真正的通用，当使用一个不同的安全模型的时候，只需要加载一个不同的内核模块

概念上简单，对Linux内核影响最小，高效，并且

能够支持现存的POSIX.1e capabilities逻辑，作为一个可选的安全模块

另一方面，各种不同的Linux安全增强系统对Linux安全模块（LSM）提出的要求是：能够允许他们以可加载内核模块的形式重新实现其安全功能，并且不会在安全性方面带来明显的损失，也不会带来额外的系统开销。

为了满足这些设计目标，Linux安全模块（LSM）采用了通过在内核源代码中放置钩子的方法，来仲裁对内核内部对象进行的访问，这些对象有：任务，inode结点，打开的文件等等。用户进程执行系统调用，首先游历Linux内核原有的逻辑找到并分配资源，进行错误检查，并经过经典的UNIX自主访问控制，恰好就在Linux内核试图对内部对象进行访问之前，一个Linux安全模块（LSM）的钩子对安全模块所必须提供的函数进行一个调用，从而对安全模块提出这样的问题"是否允许访问执行？"，安全模块根据其安全策略进行决策，作出回答：允许，或者拒绝进而返回一个错误。

另一方面，为了满足大多数现存Linux安全增强系统的需要，Linux安全模块（LSM）采取了简化设计的决策。Linux安全模块（LSM）现在主要支持大多数现存安全增强系统的核心功能：访问控制；而对一些安全增强系统要求的其他安全功能，比如安全审计，只提供了的少量的支持。Linux安全模块（LSM）现在主要支持"限制型"的访问控制决策：当Linux内核给予访问权限时，Linux安全模块（LSM）可能会拒绝，而当Linux内核拒绝访问时，就直接跳过Linux安全模块（LSM）；而对于相反的"允许型"的访问控制决策只提供了少量的支持。对于模块功能合成，Linux安全模块（LSM）允许模块堆栈，但是把主要的工作留给了模块自身：由第一个加载的模块进行模块功能合成的最终决策。所有这些设计决策可能暂时影响了Linux安全模块（LSM）的功能和灵活性，但是大大降低了Linux安全模块（LSM）实现的复杂性，减少了对Linux内核的修改和影响，使得其进入Linux内核成为安全机制标准的可能性大大提高；等成为标准后，可以改变决策，增加功能和灵活性。

3．实现方法介绍：对Linux内核的修改

Linux安全模块（LSM）目前作为一个Linux内核补丁的形式实现。其本身不提供任何具体的安全策略，而是提供了一个通用的基础体系给安全模块，由安全模块来实现具体的安全策略。其主要在五个方面对Linux内核进行了修改：

在特定的内核数据结构中加入了安全域

在内核源代码中不同的关键点插入了对安全钩子函数的调用

加入了一个通用的安全系统调用

提供了函数允许内核模块注册为安全模块或者注销

将capabilities逻辑的大部分移植为一个可选的安全模块

下面对这五个方面的修改逐个做简要的介绍。

安全域是一个void*类型的指针，它使得安全模块把安全信息和内核内部对象联系起来。下面列出被修改加入了安全域的内核数据结构，以及各自所代表的内核内部对象：

task_struct结构：代表任务（进程）

linux_binprm结构：代表程序

super_block结构：代表文件系统

inode结构：代表管道，文件，或者Socket套接字

file结构：代表打开的文件

sk_buff结构：代表网络缓冲区（包）

net_device结构：代表网络设备

kern_ipc_perm结构：代表Semaphore信号，共享内存段，或者消息队列

msg_msg：代表单个的消息

另外，msg_msg结构，msg_queue结构，shmid_kernel结构被移到include/linux/msg.h和include/linux/shm.h这两个头文件中，使得安全模块可以使用这些定义。

Linux安全模块（LSM）提供了两类对安全钩子函数的调用：一类管理内核对象的安全域，另一类仲裁对这些内核对象的访问。对安全钩子函数的调用通过钩子来实现，钩子是全局表security_ops中的函数指针，这个全局表的类型是security_operations结构，这个结构定义在include/linux/security.h这个头文件中，这个结构中包含了按照内核对象或内核子系统分组的钩子组成的子结构，以及一些用于系统操作的顶层钩子。在内核源代码中很容易找到对钩子函数的调用：其前缀是security_ops->。对钩子函数的详细说明留到后面。

Linux安全模块（LSM）提供了一个通用的安全系统调用，允许安全模块为安全相关的应用编写新的系统调用，其风格类似于原有的Linux系统调用socketcall()，是一个多路的系统调用。这个系统调用为security()，其参数为(unsigned int id, unsigned int call, unsigned long *args)，其中id代表模块描述符，call代表调用描述符，args代表参数列表。这个系统调用缺省的提供了一个sys_security()入口函数：其简单的以参数调用sys_security()钩子函数。如果安全模块不提供新的系统调用，就可以定义返回-ENOSYS的sys_security()钩子函数，但是大多数安全模块都可以自己定义这个系统调用的实现。

在内核引导的过程中，Linux安全模块（LSM）框架被初始化为一系列的虚拟钩子函数，以实现传统的UNIX超级用户机制。当加载一个安全模块时，必须使用register_security()函数向Linux安全模块（LSM）框架注册这个安全模块：这个函数将设置全局表security_ops，使其指向这个安全模块的钩子函数指针，从而使内核向这个安全模块询问访问控制决策。一旦一个安全模块被加载，就成为系统的安全策略决策中心，而不会被后面的register_security()函数覆盖，直到这个安全模块被使用unregister_security()函数向框架注销：这简单的将钩子函数替换为缺省值，系统回到UNIX超级用户机制。另外，Linux安全模块（LSM）框架还提供了函数mod_reg_security()和函数mod_unreg_security()，使其后的安全模块可以向已经第一个注册的主模块注册和注销，但其策略实现由主模块决定：是提供某种策略来实现模块堆栈从而支持模块功能合成，还是简单的返回错误值以忽略其后的安全模块。这些函数都提供在内核源代码文件security/security.c中。

Linux内核现在对POSIX.1e capabilities的一个子集提供支持。Linux安全模块（LSM）设计的一个需求就是把这个功能移植为一个可选的安全模块。POSIX.1e capabilities提供了划分传统超级用户特权并赋给特定的进程的功能。Linux安全模块（LSM）保留了用来在内核中执行capability检查的现存的capable()接口，但把capable()函数简化为一个Linux安全模块（LSM）钩子函数的包装，从而允许在安全模块中实现任何需要的逻辑。Linux安全模块（LSM）还保留了task_struck结构中的进程capability集（一个简单的位向量），而并没有把它移到安全域中去。Linux内核对capabilities的支持还包括两个系统调用：capset()和capget()。Linux安全模块（LSM）同样保留了这些系统调用但将其替换为对钩子函数的调用，使其基本上可以通过security()系统调用来重新实现。Linux安全模块（LSM）已经开发并且移植了相当部分的capabilities逻辑到一个capabilities安全模块中，但内核中仍然保留了很多原有capabilities的残余。这些实现方法都最大程度的减少了对Linux内核的修改影响，并且最大程度保留了对原有使用capabilities的应用程序的支持，同时满足了设计的功能需求。以后要使capabilities模块完全独立，剩下要做的主要步骤是：把位向量移到task_struct结构中合适的安全域中，以及重新定位系统调用接口。

4．接口说明：给内核开发人员和安全研究人员使用的钩子

Linux安全模块（LSM）对于内核开发人员和安全研究人员的价值就在于：可以使用其提供的接口将现存的

安全增强系统移植到这一框架上，从而能够以可加载内核模块的形式提供给用户使用；或者甚至可以直接编写适合自己需要的安全模块。Linux安全模块（LSM）提供的接口就是钩子，其初始化时所指向的虚拟函数实现了缺省的传统UNIX超级用户机制，模块编写者必须重新实现这些钩子函数来满足自己的安全策略。下面简要介绍Linux安全模块（LSM）提供的钩子，详细情况请参考源代码，特别是include/linux/security.h头文件中security_operations结构的定义。至于具体如何根据自己需要的安全策略编写安全模块，可以参考SELinux，DTE，LIDS等系统的安全模块实现。

首先是任务钩子，Linux安全模块（LSM）提供了一系列的任务钩子使得安全模块可以管理进程的安全信息并且控制进程的操作。模块可以使用task_struct结构中的安全域来维护进程安全信息；任务钩子提供了控制进程间通信的钩子，例如kill()；还提供了控制对当前进程进行特权操作的钩子,例如setuid()；还提供了对资源管理操作进行细粒度控制的钩子，例如setrlimit()和nice()。

其次是程序装载钩子。很多安全模块，包括Linux capabilities，SELinux，DTE都需要有在一个新程序执行时改变特权的能力。因此Linux安全模块（LSM）提供了一系列程序装载钩子，用在一个execve()操作执行过程的关键点上。linux_binprm结构中的安全域允许安全模块维护程序装载过程中的安全信息；提供了钩子用于允许安全模块在装载程序前初始化安全信息和执行访问控制；还提供了钩子允许模块在新程序成功装载后更新任务的安全信息；还提供了钩子用来控制程序执行过程中的状态继承，例如确认打开的文件描述符。

再次是进程间通信IPC钩子。安全模块可以使用进程间通信IPC钩子来对System V IPC的安全信息进行管理，以及执行访问控制。IPC对象数据结构共享一个子结构kern_ipc_perm，并且这个子结构中只有一个指针传给现存的ipcperms()函数进行权限检查，因此Linux安全模块（LSM）在这个共享子结构中加入了一个安全域。为了支持单个消息的安全信息，Linux安全模块（LSM）还在msg_msg结构中加入了一个安全域。Linux安全模块（LSM）在现存的ipcperms()函数中插入了一个钩子，使得安全模块可以对每个现存的Linux IPC权限执行检查。由于对于某些安全模块，这样的检查是不足够的，Linux安全模块（LSM）也在单个的IPC操作中插入了钩子。另外还有钩子支持对通过System V消息队列发送的单个消息进行细粒度的访问控制。

下面是文件系统钩子。对于文件操作，定义了三种钩子：文件系统钩子，inode结点钩子，以及文件钩子。Linux安全模块（LSM）在对应的三个内核数据结构中加入了安全域，分别是：super_block结构，inode结构，file结构。超级块文件系统钩子使得安全模块能够控制对整个文件系统进行的操作，例如挂载，卸载，还有statfs()。Linux安全模块（LSM）在permission()函数中插入了钩子，从而保留了这个函数，但是也提供了很多其他inode结点钩子来对单个inode结点操作进行细粒度访问控制。文件钩子中的一些允许安全模块对read()和write()这样的文件操作进行额外的检查；还有文件钩子允许安全模块控制通过socket IPC接收打开文件描述符；其他的文件钩子对像fcntl()和ioctl()这样的操作提供细粒度访问控制。

接下来是网络钩子。对网络的应用层访问使用一系列的socket套接字钩子来进行仲裁，这些钩子基本覆盖了所有基于socket套接字的协议。由于每个激活的用户socket套接字有伴随有一个inode结构，所以在socket结构或是更底层的sock结构中都没有加入安全域。socket套接字钩子对有关进程的网络访问提供了一个通用的仲裁，从而显著扩展了内核的网络访问控制框架（这在网络层是已经由Linux内核防火墙netfilter进行处理的）。例如sock_rcv_skb钩子允许在进入内核的包排队到相应的用户空间socket套接字之前，按照其目的应用来对其进行仲裁。另外Linux安全模块（LSM）也为IPv4，UNIX域，以及Netlink协议实现了细粒度的钩子，以后还可能实现其他协议的钩子。网络数据以包的形式被封装在sk_buff结构（socket套接字缓冲区）中游历协议栈，Linux安全模块（LSM）在sk_buff结构中加入了一个安全域，使得能够在包的层次上对通过网络层的数据的安全信息进行管理，并提供了一系列的sk_buff钩子用于维护这个安全域的整个生命周期。硬件和软件网络设备被封装在一个net_device结构中，一个安全域被加到这个结构中，使得能够在设备的层次上维护安全信息。

最后是其他的钩子。Linux安全模块（LSM）提供了两种其他系列的钩子：模块钩子和顶层的系统钩子。模块钩子用来控制创建，初始化，清除内核模块的内核操作。系统钩子用来控制系统操作，例如设置主机名，访问I/O端口，以及配置进程记帐。虽然现在的Linux内核通过使用capability检查对这些系统操作提供了一些支持，但是这些检查对于不同操作差别很大并且没有提供任何参数信息。

5．模块说明：给普通用户使用的现成的安全功能

Linux安全模块（LSM）对于普通用户的价值就在于：可以提供各种安全模块，由用户选择适合自己需要加载到内核，满足特定的安全功能。Linux安全模块（LSM）本身只提供增强访问控制策略的机制，而由各个安全模块实现具体特定的安全策略。下面简要介绍一些已经实现的安全模块。

SELinux。这是一个Flask灵活访问控制体系在Linux上的实现，并且提供了类型增强，基于角色的访问控制，以及可选的多级安全策略。SELinux原来是作为一个内核补丁实现的，现在已经使用Linux安全模块（LSM）重新实现为一个安全模块。SELinux可以被用来限制进程为最小特权，保护进程和数据的完整性和机密性，并且支持应用安全需求。

DTE Linux。这是一个域和类型增强在Linux上的实现。就像SELinux一样，DTE Linux原来是作为一个内核补丁实现的，现在已经使用Linux安全模块（LSM）重新实现为一个安全模块。当这个安全模块被加载到内核上时，类型被赋给对象，域被赋给进程。DTE策略限制域之间和从域到类型的访问。

Openwall 内核补丁的LSM移植。Openwall内核补丁提供了一系列的安全特性集合来保护系统免受例如缓冲区溢出和临时文件竞争这样的攻击。有安全模块正在被开发出来以支持Openwall补丁的一个子集。

POSIX.1e capabilities。Linux内核中已经存在有POSIX.1e capabilities逻辑，但是Linux安全模块（LSM）把这个逻辑划分到了一个安全模块中。这样的修改使得不需要的用户可以从他们的内核中把这个功能略去；也使得capabilities逻辑的开发可以脱离内核开发获得更大的独立性。

LIDS。这是中国人谢华刚发起的项目。开始时作为一个入侵检测系统开发，后来逐渐演变为使用访问控制系统的形式来进行入侵预防，它通过描述一个给定的程序可以访问哪些文件来进行访问控制。同样的，LIDS原来是作为一个内核补丁实现的并附带了一些管理工具，现在已经使用Linux安全模块（LSM）重新实现为一个安全模块。

当然还有缺省的传统超级用户机制。这个安全模块是Linux安全模块（LSM）缺省的，实现了传统的UNIX超级用户特权机制。

6．具体使用方法说明：**step by step**

Linux安全模块（LSM）目前作为一个Linux内核补丁的形式实现，在GPL许可证下发布供用户自由使用。

首先用户可以在http://lsm.immunix.org/lsm_download.html下载到对应于Linux 2.4稳定版本和Linux 2.5开发版本的LSM补丁，放在某个目录下，例如是目录/path/to/linux-2.4.x，通过执行下面的命令来使LSM补丁作用在Linux内核上：

```
# cd /path/to/linux-2.4.x
# zcat /path/to/patch-2.4.x-lsm.gz | patch -p1
```

然后用户可以在**http://lsm.immunix.org/lsm_modules.html**连接到已经实现安全模块的站点，下载到所需要的安全模块，将安全模块加载到**Linux**内核中，这样用户需要的安全策略就可以起作用了，从而增强了系统的安全性。具体安全模块的安装方法这里就略过了，各个安全模块都会提供详细的安装说明文件，用户可以参考这些文件，例如**SELinux**的安全模块安装说明文件在：**http://www.nsa.gov/selinux/doc/readme.html**，又如**LIDS**的安全模块安装说明文件在：**http://www.lids.org/install.html**。

如果用户有**Linux**内核和安全的相关背景知识和开发经验，想根据自己需要的安全策略编写安全模块。可以在**http://lsm.immunix.org/lsm_bk.html**跟踪查看**Linux**安全模块（**LSM**）的源代码和现有安全模块的源代码，参考其实现方法编写自己的安全模块。这样在满足自己安全需求的同时，也可以为**Linux**安全模块（**LSM**）的发展作出一些贡献，使其早日被**Linux**内核接受成为**Linux**内核安全机制的标准，使更多的用户得益。

7．结束语：未来的标准

Linux安全模块（LSM）的起因是：一方面Linux内核现有的安全机制是不足够的；另一方面现存的安全增强系统又各自为战并且难以使用。Linux安全模块（LSM）比较好的解决了这个问题：一方面补丁比较小，对内核源代码的修改影响不多，所带来的负载也不大；另一方面对现存的安全增强系统提供了比较好的接口支持，并已经有不少很好的安全模块可以使用。Linux安全模块（LSM）目前仍然是作为一个Linux内核补丁的形式提供，但是其同时提供Linux 2.4稳定版本的系列和Linux 2.5开发版本的系列，并且很有希望进入Linux 2.6稳定版本。我们期待着那一天：Linux安全模块（LSM）被Linux内核接受成为Linux内核安全机制的标准，在各个Linux发行版中提供给越来越多的用户使用。

参考资料

Linux安全模块（LSM）的源代码管理站点在： http://lsm.bkbits.net/

Linux内核的主站点在： http://www.kernel.org/

安全增强Linux（SELinux）的主站点在： http://www.nsa.gov/selinux/

域和类型增强（DTE）的主站点在： http://www.cs.wm.edu/~hallyn/dte/

Linux入侵检测系统（LIDS）的主站点在： http://www.lids.org/

Linux内核防火墙netfilter/iptables的主站点在： http://www.netfilter.org/

如果读者不具有Linux内核和安全的相关背景知识，可以先阅读 developerWorks 上的这三篇文章：
"揭开 SE Linux 的秘密"： http://www.ibm.com/developerworks/cn/linux/s-selinux/index.shtml、 http://www.ibm.com/developerworks/cn/linux/s-selinux2/index.shtml
"FreeBSD 5.0中强制访问控制机制的使用与源代码分析"： http://www.ibm.com/developerworks/cn/security/se-fbsdsec/part1/index.shtml 、 http://www.ibm.com/developerworks/cn/security/se-fbsdsec/part2/index.shtml
"netfilter：Linux 防火墙在内核中的实现"： http://www.ibm.com/developerworks/cn/linux/network/l-netip/index.shtml

static.content.url=http://www.ibm.com/developerworks/js/artrating/
SITE_ID=10
Zone=Linux
ArticleID=21120
ArticleTitle=如何增强 Linux 系统的安全性，第一部分: Linux 安全模块（LSM）简介
publish-date=07262003
url=http://www.ibm.com/developerworks/cn/linux/l-lsm/part1/index.html