操作系统加固技术（ROST）与安全服务器

网络安全现状以及ROST的意义

•现有的网络安全现状

前两年的网络安全市场中奉行产品为先的市场策略,因为那时客户还处于网络的基础建设阶段，各种复杂的应用系统都没有完善，随着这两年网络各层应用的兴起，安全问题已经不是简单的单一层面的产品，而应该是针对各层应用的网络安全整体解决方案，所以我们说，安全是伴随着应用的发展而发展的，安全和应用是相辅相成的，那么就引出了国家等级保护体系的概念。

•ROST在国家等级保护体系中的地位

•ROST的意义

ROST（Reinforcement Operating System Technique）是一项利用安全内核来提升操作系统安全等级的技术，这项技术的核心就是在操作系统的核心层重构操作系统的权限访问模型，实现真正的强制访问控制。使操作系统达到第三等级（B1 级）的安全技术要求，它是目前国家等级保护体系中系统层面的解决方案，意义在于“ 承上启下 ”，即可以很好的支持各种操作系统以及硬件平台，也能对操作系统上层的各种现有的大型应用有很好的安全支撑作用。

ROST知识背景

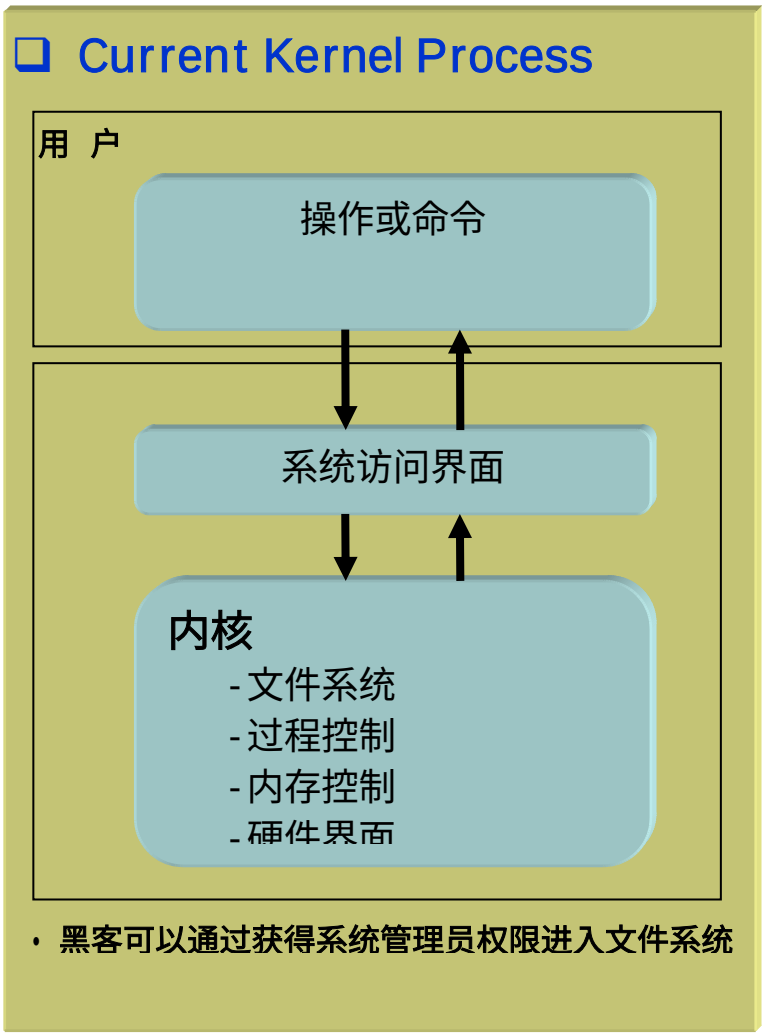
• 各种针对系统的黑客技术(本质描述)

- 病毒:是程序对程序非法的写操作。在我们的主流操作系统中有一个用户权限是至高无上的，它就是administrator/root，它的身份鉴别机制是密码，所以这是非可信的。一旦病毒获得了该权限,那么就意味着控制了整个系统，它可以往系统文件里写病毒的复制代码，这是病毒感染系统的基本原理。为什么病毒能有权限去写复制代码呢？是因为它已经拿到了凌驾整个操作系统的最高权限。
- 蠕虫:是自动的黑客，执行黑客的每个步骤，特定的攻击方式，主要是缓冲溢出攻击，蠕虫工作的重要依据需要攻陷目标机器，这个攻陷的标志就是拿到该机器的最高权限administrator/root，这和病毒的道理是一样的，因为这个最高权限变的非可信而导致的系统安全问题。

- 黑客攻击：综合以上的技术以及一些利用错误配置，密码等的入侵行为。

所以，现有的问题归根到底是二级的自主访问控制的操作系统的弱点所决定的。

一个操作系统的操作指令模型



重构操作系统的两个基本思路

- 重构操作系统源代码技术：通过改写 LINUX 源代码，从而达到三级的技术要求，优点是做的比较彻底，操作系统层的安全控制做的比较好。缺点是对上层应用兼容性不好，而且需要替换客户的现有应用，大大影响了客户的业务连续性。虽然它能够做的比较彻底，但它不能很好的“承上启下”，我们说没有应用就没有安全，所以这种做法是把自己和应用割裂开的做法。

- 内核模块技术：在驱动层（0层）加上安全内核模块，拦截所有的内核访问路径，从而达到三级的技术要求，达到的安全效果和重构操作系统源代码技术差不多，好处是不会影响客户的业务连续性，甚至不需要客户重启系统，对上层的所有应用都支持，对下层所有系统和机器都支持，而且能在操作系统粒度上保证上层应用的安全，是安

全服务器标准的技术基础。

三级的安全操作系统的基本要素

•信道：是指主体与客体的访问通道，在三级中是不考虑这个通道是否安全的，三级只关心访问结果。

•敏感标记：敏感标记是强制访问控制的依据，主客体都有，它存在形式无所谓，可能是整形的数字，也可能是字母，总之它表示主客体的安全级别。强制访问控制就是依据这个级别来决定主体以何种权限对客体进行操作，敏感标记是由强认证的安全管理员进行设置的。

•强制访问控制模型：是三级的重要标志，在强制访问控制模型中有一个安全管理员，他本身不参与操作系统行为，但他可以对主客体进行敏感标记的设置，当然安全管理员是具有强身份鉴别的，并且有审计管理员来监督他的行为，在强制访问控制模型下，无论主体是否是客体的主人，都遵循敏感标记的安全级别进行访问控制。

钩子技术

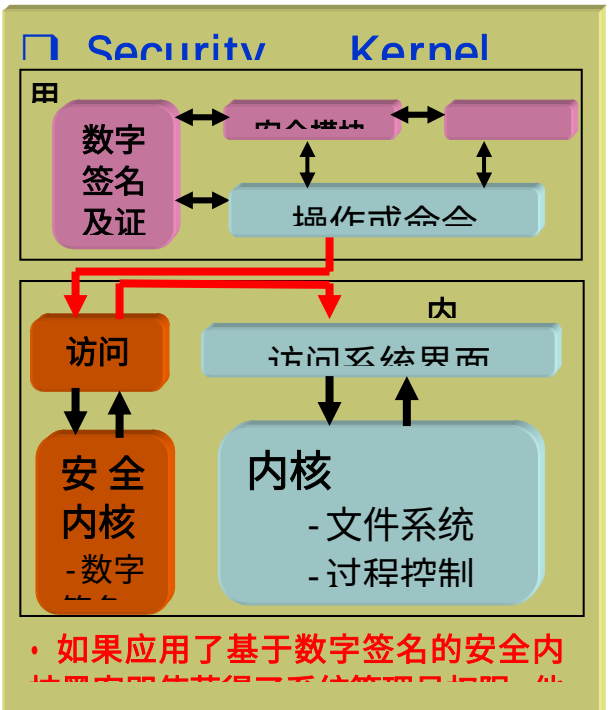
函数HOOK技术

HOOK技术的应用

内核函数的HOOK

我们在理解 ROST 技术核心的时候，内核钩子技术其实就是 ROST 的核心，钩子运用于很多网络安全技术中，比如反病毒技术，防火墙技术，IDS 技术等;病毒软件就是勾住了文件驱动，防火墙就是勾住了响应的网络层的函数等。

被勾过的内核函数访问流程



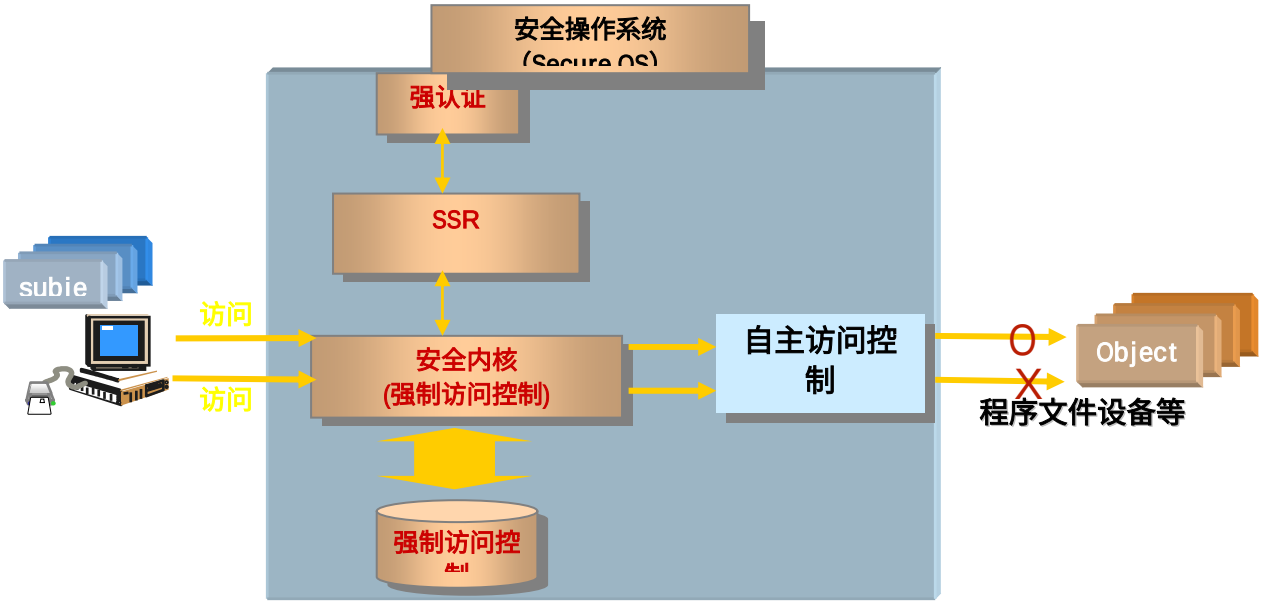
这是一个具有强制访问控制的安全内核模型图,在应用层存有我们强制访问控制的规则库，这个规则库是通过强身份鉴别的安全管理员去建立的，在 windows 内核启动的那一刻，这个规则库就被传递到安全内核里,形成访问控制。我们可以看到，这就是一个内核钩子的模型，在 0 层旁路了内核访问路径，在强制访问控制模型下，不论什么用户都要受安全规则库的规则限制,即使是最高权限用户也无法访问受保护的资源,这就是 ROST 的核心思想。

ROST原理

- **用钩子来控制系统的通讯信道**
- **强制访问控制模型**
- **规则的传递方式**

ROST 就是一个具有强制访问控制模型的安全内核

强制访问内核模型图例



基于ROST的安全操作系统

ROST技术的优势

•不影响客户的业务的连续性

对于一些大型客户来说，一套成熟系统的稳定性和连续性很需要保障，所以我们的安全提升一定要符合他们所应用的需要，ROST 的一个重要的技术优势就在于能够很好的支持客户现有的任何应用系统以及下层平台，让客户的业务能够不知不觉提升自己系统的安全等级。

•更好的支持各种操作系统平台

•更好的支持各种应用系统，并且在操作系统粒度下使其达到安全标准。

从 ROST 的技术优势可以看出，ROST 技术是最符合当前客户应用的系统级的安全解决方案，它不但能很好的和上层应用，对下层硬件进行兼容，而且能保证他们的安全性，比如对于一个数据库应用系统来说，它对于操作系统是两个要素，一个是数据库文件，一个是数据库进程等，ROST 能够在这几个方面很好的保障数据库的安全性，能够保障数据库文件以及数据库进程的安全，他们仍然是强制访问控制中受保护的客体。所以 ROST 也是安全服务器标准的技术基础。

ROST与各种平台操作系统的捆绑

•WINDOWS

•LINUX

•AIX,HP UNIX,SOLARIS… …

安全操作系统的概念

•安全操作系统＝ROST＋普通操作系统

这是基于 ROST 的安全操作系统的概念，当然安全操作系统还有其他的概念，象红旗的安全操作系统，他们就是通过重构操作系统源代码的技术来实现的安全操作系统。我们的安全操作系统的概念是普通的提供了各种应用的操作系统＋ROST 技术,ROST 技术只是一个安全模块，不会影响原有的操作系统的上层应用，这是我们这种安全操作系统和传统的安全操作系统概念的本质不同。

ROST对应用系统的安全支持

- MS SQL
- ORACLE
- MY SQL
- SYBASE
- DB2

当然除了数据库之外，我们还支持任何别的应用系统。

基于安全操作系统的安全服务器

安全服务器的要素

•安全的物理设备

是一个安全物理服务器设备，在物理层有一些安全措施，比如控制硬件的拔插，硬件各零件状态的管理检测等。

•安全的操作系统

这里的安全操作系统就是一个需要承上启下的系统，那么可以说就是我们的 ROST＋普通的操作系统

•安全的应用系统

在 ROST 的“ 承上 ”的特性安全支持下的应用系统。

•专业的管理系统

针对服务器的每个安全点的控制管理系统

安全服务器的概念

安全服务器＝安全操作系统＋安全的应用系统＋普通服务器

这里的安全应用系统是指在操作系统粒度上的安全应用系统，比如数据库在操作系统粒度上就表现为数据库文件和数据库进程,我们的 ROST 就是可以保证数据库文件和数据库进程的安全，从操作系统粒度上增强了数据库本身的安全性，当然对于数据库这个应用系统来说，它有它自己的一套安全标准，这个不在安全服务器的范畴。

需要注意的是，那些重构操作系统源代码技术的安全操作系统是不能很好的或者只能部分的去兼容现有应用系统，那么他们就不能构成一个很好的安全服务器，因为我们说没有应用就没有安全，牺牲了上层的应用还谈何安全呢？