系统安全课程报告

2020.12.30

计算机系统安全一直以来都是一个可以引发热烈讨论的话题，似乎每隔一段时间就会有一些公共安全问题的出现来引发人们对于安全问题的担忧，通过对本门课程的学习，伴随着老师的深入浅出的讲解，我对于系统安全的理解也在逐渐加深，这也为我未来的职业发展奠定了基础；从这个角度来说，我对于老师和这门课本身是非常感谢的。

事实上，系统安全一直都被高度重视，美国政府把确保信息安全列为国家安全战略最重要的组成部分，自 2000 年以来，多次颁布法令以发展期信息系统的安全；欧盟中国也是如此，并且规定了与系统安全相关的《中华人民共和国密码法》。

然而，无论来自国家政府层面的重视程度有多高，一个系统最不安全的因素往往来自使用者。事实上，一年中会发生无数起信息内容安全相关的事件，大部分与数据库的问题相关，还有一部分与权限问题相关；可以说，一部分原因来自于软件中的 Bug，另一大部分来源于错误的配置------也就是说，许多使用者（不管是程序员还是普通用户）的知识储备不够丰富，以及随着安全性提高而来的用户成本的升高，都是引发安全问题的因素。因此，我们知道必须遵守适当保护的原则，安全目标不是最大化安全，而是最大化实用性，限制风险的花费控制在可接受范围内，但同时也要遵守有效性原则，必须使用控制措施；控制措施要适当、有效；措施要充分、适合、容易使用，用户心理能接受。

从系统的角度来看，计算机系统可划分为多个层次，硬件、操作系统、系

统软件：数据库等、以及通过网络连接起来的计算机系统；但不要忘记，人也是其中的一个组成成分。因此，如果一个操作系统希望提供安全方面的特性，那么，首先他应该允许多用户安全的共享单机，也即进程、内存、文件设备等的分离与共享；其次，他应该实现在网络环境下的安全操作。

按照上文所述，我们先来谈谈内存的保护，即内存的访问控制，我们希望可以保证一个用户的进程不能访问其他人的内存空间，因此我们可以考虑使用权限控制的思路。众所周知的是，在 Linux 系统中，往往存在两类用户：普通用户和管理员用户。相对的，在运行时，存在着用户模式和内核模式；内核模式可以执行任意指令、访问任意内存地址、 硬件设备、中断操作、改变处理器特权状态、访问内存管理单元、修改寄存器；而用户模式则不然，他不能执行某些指令，不能停止中断，改变任意进程状态，访问内存管理单元等，这二者之间的转换必须通过系统调用 system call 来实现，system call 是一种陷阱，在 Linux 上起着至关重要的作用。

由于用户的重要性，我们同样必须设计一种关于用户的安全机制，这包括了鉴别、访问控制、记录日志和审计、入侵检测以及恢复。用户鉴别这部分实际上可以和之前学到的密码学联系起来，但总体来说从加密方法的设计到用户实际使用中需要注意的地方都有许多非常繁杂的地方，一直以来都是一个热点问题。访问控制同样也是一个非常重要的部分，参考监视器是其重点，检测所有与安全相关的操作，如创建进程、访问文件等，其中主体包括用户、进程；客体包括文件、进程、端口；访问包括读、写、执行。类似地，入侵检测检测和记录网络相关事件、计算机系统遭到入侵和攻击事件，主要是被动式 IDS 与主动式 IDS 的比较，以及基于主机 IDS vs. 基于网络 IDS 的入侵检测系统。

这门课的学习中，UNIX 访问控制对我的帮助是很大的，之前系统遇到问题我往往在网上查到解决办法后就结束了，从未尝试理解其中的原因，尤其是 chmod 777 到底是什么意思呢？在这个部分，本课程从文件的组织形式开始，介绍了在系统层面上，文件和目录之间的关系，这部分实际上和 CSAPP 中的内容是有异曲同工之妙的。一个很重要的部分就是文件的基本权限位，每种用户范围都具有不同的的 rwx 权限位，这就构成了三位 0-7 组成的三个数字。如将权限设置为 rwxr--r-x，对应二进制表示为 111 100 101，而当权限检查时依次检查拥有者、组、其它：当用户是文件的拥有者，运行时检查拥有者的 r/w/x 位，再检查组权限，查看组 r/w/x 权限位，最后，检查其它人的 r/w/x 权限位。类似地，对于文件的执行权限，二进制文件 vs. 脚本文件之间也同样存在类似的问题，这就告诉我们一定要合理设置权限，对于目录和目录中的文件，一定要谨慎处理，因为权限位不是直接授权用户操作某程序，而是授权给用户可以使用相应的系统调用，由于 UNIX 中，目录的权限不具有继承性，访问一个路径下的文件时，需要整个路径上的目录都有执行权限。

访问控制也是一个亟需加强的部分，根据本学期课程中所学到的内容，计算机感染恶意病毒往往是因为访问控制中存在的问题，配置因素也应该被纳入其中。这是因为 Unix 设计时只考虑了单机的用户共享系统，但并未太多考虑网络相关的问题，加上控制管理是粗粒度的，没能做到每个进程相独立，以及 root 后功能太过强大，导致访问控制存在问题。为了解决上述的问题，有三类方法被提出：虚拟化限制：限制进程在有限空间内，不影响其它进程，主要有 chroot 等技术；以及把 root 权限进行划分，从而使得用户不用 root 也能执行需要部分 root 权限才能完成的工作；最后，细粒

度的、基于进程的强制访问控制（Mandatory Access Control）也是需要被实现的，他的目的是更好的实现最小特权：分配一个程序其需要的权限，不同于按用户进行权限划分。Chroot 的系统调用改变当前进程和子进程到指定路径下的“根”目录，新的“根”目录受真正的 root 的文件系统约束。

一般的目录架构：	CHROOT的目录架构：
/	/hell/
/bin	/hell/bin
/sbin	/hell/sbin
/usr/bin	/hell/usr/bin
/home	/hell/home

Chroot 增加了系统的安全性，限制了用户的权力；建立了一个与原系统隔离的系统目录结构，方便用户的开发；切换系统的根目录位置，引导 Linux 系统启动以及急救系统等。

与 Chroot 类似，虚拟化限制方法还有 FreeBSD jail 这样的操作系统级虚拟化，以及虚拟机、虚拟指令集，但是后两者实际的研究与我们关系不大，是一个比较复杂的工程，同时在性能上有许多困难。

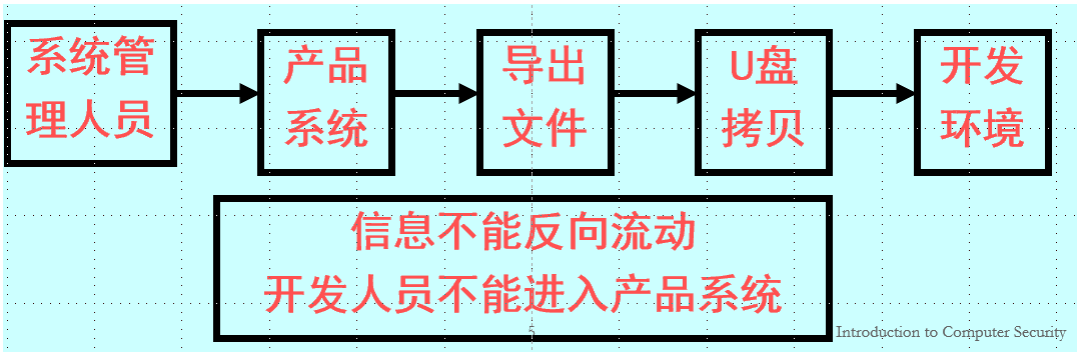
在这样的基础上，之后我们谈到了数据库安全的内容，正如上文谈到的，数据库是安全问题的重灾区。用户通过数据库管理系统(DBMS)和数据库交互，这就会在权限、管理、访问上都存在问题。用户通过 DBMS 和数据库交互，DBMS 提供查询、修改、添加、删除命令。数据库建立在文件系统之上，具有访问时共享、最小冗余、数据一致性、数据完整性、控制访问的特点，然而安全和性能是一对冲突。物理数据库、逻辑数据库、数据项都需要完整性的保证，当两个用户同时下单时，如何选择？类似这样的问题处理不好，都会变成重大的安全问题。此外，存在数据库中的数据本身---甚至是字段名本身，都会带来安全问题。存在敏感数据时，要设想

是否他人可以通过推理的方式发现某些重要信息---甚至不需要输入某些查询，解决的方法可以是不给出完全精确的数值。

除此之外，安全策略也是一个值得讨论的问题。对于不同的主体，安全策略的着重点也可能不同，军队会非常在乎保密性，而商业上会比较重视安全性，这也是当前存在的问题。当然，无论如何，在这之中有信任是作为基础的。

机密性模型同样也是一个重要的组成部分，Bell-LaPadula 模型（BLP）是一种状态机模型，用于在政府和军事应用中实施访问控制,用于规范美国国防部（DoD）的多级安全（MLS）策略。该模型是计算机安全策略的形式状态转换模型，它利用访问主体和访问对象的安全等级来描述一系列访问控制规则。安全等级的范围从最敏感（如“最高机密”）到最不敏感（如“未分类”或“公开”）。

完整性也是要非常重要的一部分,如同上面所说，商业上关注完整性，而不是机密性。因此，作为未来的执业人员，我认为确实有必要参考如此的工作流程（如下图）。



由此，我们可以得出几个通用操作规则：责任分离、功能分离、审计需求。

本门课后续还有 Biba 模型、clark-wilson 模型等内容，这都是系统安全

非常重要的组成部分。根据上文的论述，随着各行各业中应用计算机程度的不断加深，计算机在便捷我们工作、生活的同时，还需要更好的思考如何通过恰当的方式检测软件存在的漏洞和安全问题，不断提升计算机系统的安全性，这对更好的应用计算机具有较强的现实意义。

参考文献：

[1]张静.关联数据的访问控制技术进展探究[J].北京印刷学院学报,2020,28(07):154-156.
[2]何鼎权,胡辉,严家成.基于 RBAC 的通用权限管理系统[J].电脑知识与技术,2020,16(33):97-102.
[3]陆华.浅谈内网系统安全及防范对策[J].中国新通信,2020,22(22):121-122.
[4]张沛强.大数据背景下计算机网络安全防范与对策分析[J].中小企业管理与科技(下旬刊),2020(12):114-115.
[5]刘可道. 网络安全法治研究述评[N]. 中国社会科学报,2020-12-23(004).
[6].最容易“泄露”的密码支付方式 “人脸识别”正在兴起[J].中外管理,2019(11):80-82.