《openEuler内核编程》

课程讲稿

第十章 第3讲

AppArmor

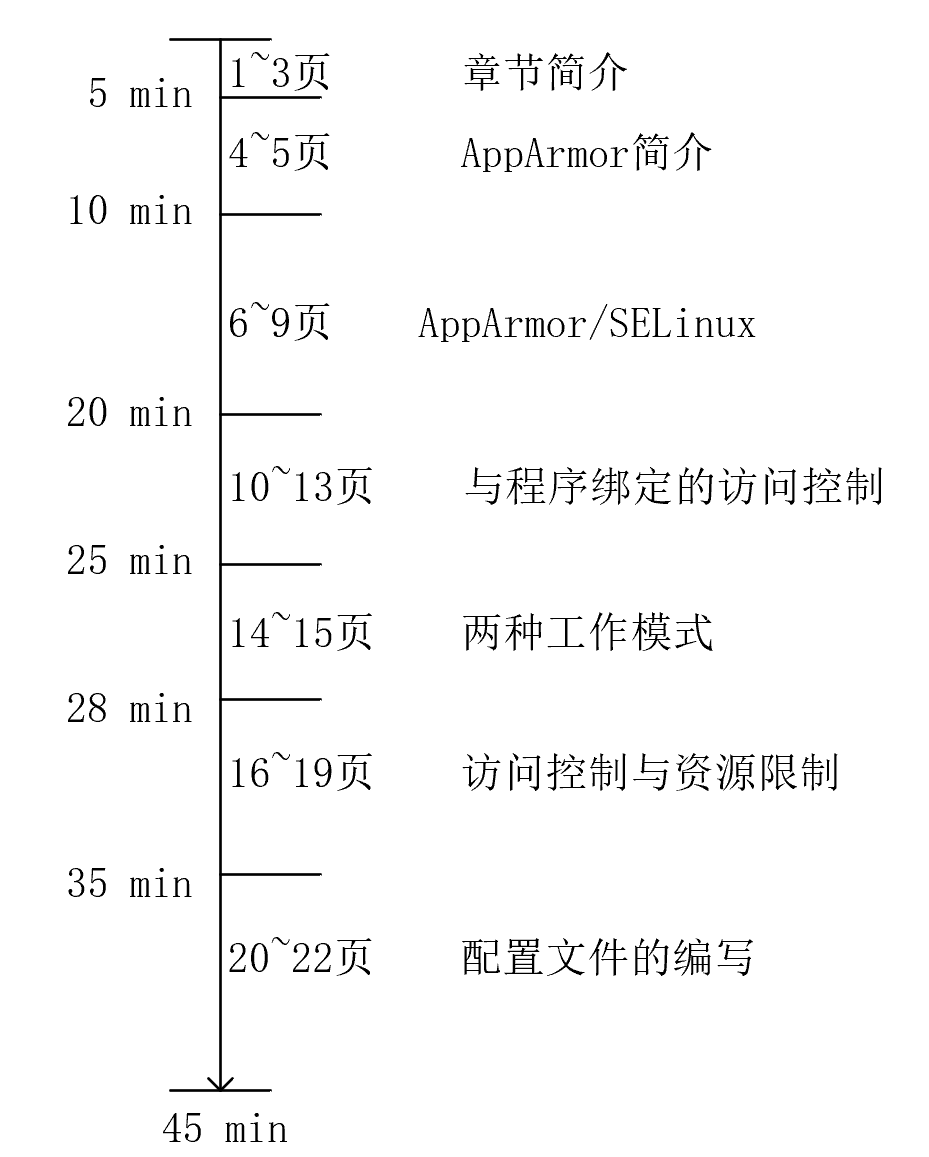
软件所制

第十章 第3讲 AppArmor

**学时：**1学时

**教学目的：**系统学习apparmor，深入了解apparmor原理

**课程时间线：**



**课外参考读物：**

<https://blog.csdn.net/zahuopuboss/article/details/53244872>

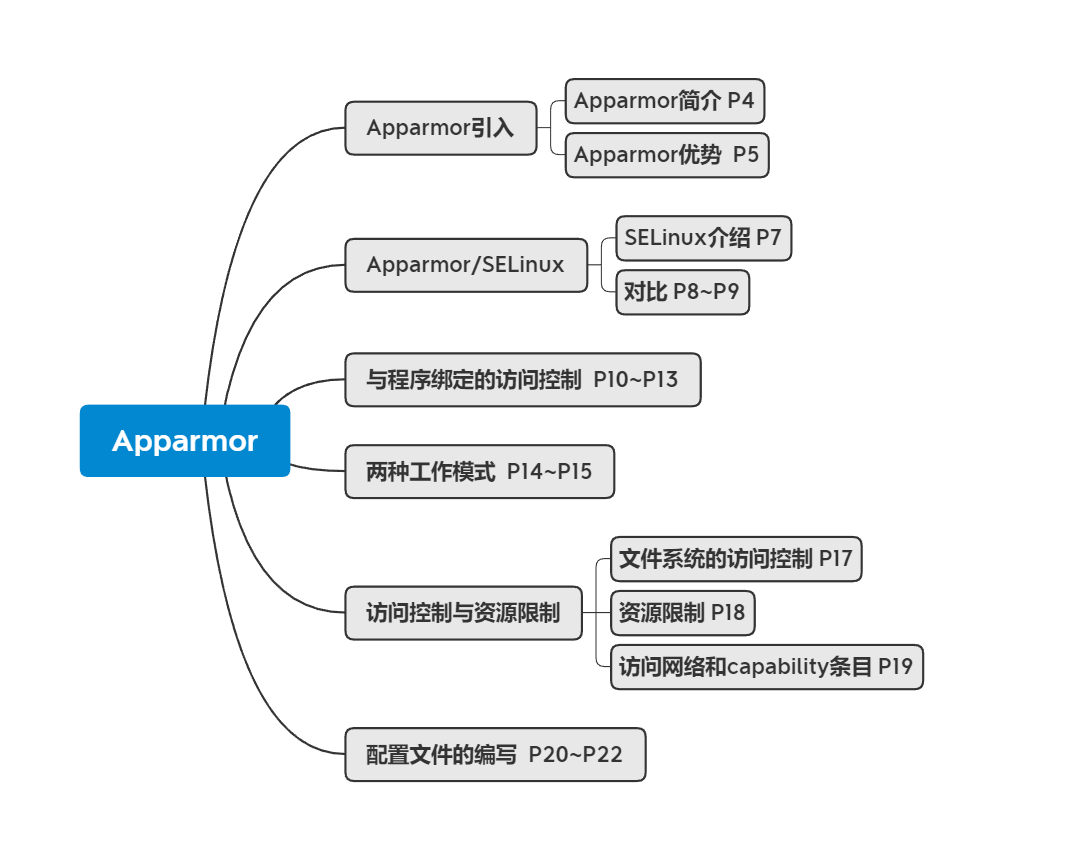
<https://blog.csdn.net/heluan123132/article/details/73874510>

<https://datahunter.org/apparmor>

<https://zh.wikipedia.org/wiki/AppArmor>

<https://apparmor.net/>

**知识框图：**



**PPT讲稿：**

1. 这是第几章操作系统课程，我们主要继续介绍Linux安全相关的知识。
2. 在这一讲，我们将课程分为6个部分。
3. 首先第一部分是Apparmor的引入。
4. apparmor 是一个类似于SELinux 的 LSM (Linux Security Module)，LSM是Linux内核负责管理安全的模块，主动保护操作系统和应用程序免受外部或者内部的威胁，而这一过程主要是通过强制访问控制机制。Apparmor诞生于19998年，在1998-2003年期间首先在immunix Linux中被使用，当时被称为subdomain，后来在2005年被novell收购并改名为apparmor。2007年4月的时候apparmor第一次被成功移植并打包进Ubuntu中，在2010年被整合到2.6.36版本的Linux内核中。
5. apparmor有很多优势，包括提升工作效率，通过apparmor在遇到问题时开发人员可以有计划的对系统做出及时的升级而不是仅仅做出反应；软件的可靠性得以保证，apparmor利用应用的配置文件规定应用程序有权利做什么，这种方式比开发一款无错误的软件要容易的多；提供高效的保护，可以有效的阻止位置的威胁和0-day攻击。
6. 与apparomor相似的是selinux。
7. Selinux是由美国国家安全局开发的，也是基于强制控制访问，最大限度的减少系统中服务进程可访问的资源，selinux和apparmor使用相同的框架，在使用中可以相互替换。而AppArmor是SELinux的替代产品，AppArmor 的开发初衷是因为人们认为 SELinux 太过复杂，不适合普通用户管理。
8. 比较apparmor和selinux，从安全的角度来看，apparmor是基于路径名的系统，需要标记文件系统，在创建配置文件时，无需修改其他配置文件，因为配置文件仅引用它们使用的路径名，路径名也易于理解和审核，而selinux需要将标签粘贴到所有文件和过程，标签标识了通信渠道，添加新的配置文件可能需要修改现有配置文件来拆分通信渠道，但是并非所有应用程序都保留标签。
9. 这是一个in.ftpd的配置文件，可以看出，相同的程序用apparmor的代码量要比selinux少4倍。
10. Apparmor是与程序绑定的访问控制。
11. 它的架构从下到上分别是kernel、Linux security module api、profile。LSM作为内核与apparmor的中间媒介，对各种类型的application使用配置文件这种白名单的方式，来授予对应用程序所需的文件/目录和POSIX功能的最小列表的访问权限。整个过程还包括自动化工作流，自动扫描、自动生成、自动学习模式和交互式优化器。
12. Apparmor的程序与配置文件绑定是以以下形式构成，假设有一个可执行文件的路径为/home/361way/demoexe，如果要用Apparmor对其进行访问控制的话，就要新建一个配置文件（后面我再讲怎么写这个配置文件），文件名为home.361way.demoexe，并把这个配置文件放到Apparmor专门放置配置文件的目录下（/etc/apparmor.d）。所以每一个可执行文件都是与一个配置文件绑定的，因此如果修改demoexe的文件名的话，配置文件将失效。
13. 以ntpd为例，这是ntpd的一个配置文件，其中会写名拥有权能capability和拥有的权限。
14. Apparmor有两种工作模式。
15. Enforcement和complain。再enforcement模式下，配置文件里列出的限制条件都会得到执行，并且对于违反这些限制条件的程序会进行日志记录。而在complain模式下，配置文件里的限制条件不会得到执行，AppArmor只是对程序的行为进行记录。例如程序可以写一个在配置文件里注明只读的文件，但AppArmor不会对程序的行为进行限制，只是进行记录。那既然complain不能限制程序，为什么还需要这种模式呢，因为——如果某个程序的行为不符合其配置文件的限制，可以将其行为记录到系统日志，并且可以根据程序的行为，将日志转换成配置文件。这样我们可以随时对配置文件进行修改，选择自己需要的模式。
16. 下来我们介绍访问控制和资源限制。
17. 首先是对文件系统的访问控制，apparmor可以对某一个文件，或者某一个目录下的文件进行访问控制，具体操作如下表，比如说如果需要对/tmp目录进行文件的读取时，可以表达为/tmp r。
18. 对资源的限制，Apparmor可以提供类似系统调用setrlimit一样的方式来限制程序可以使用的资源，使用语法为set rlimit [resource] <= [value]其中，resource代表某一种资源，value代表某一个值，如果要对程序可以使用的虚拟内存做限制时，set rlimit as<=1M, （可以使用的虚拟内存最大为1M）。这里值得注意的是Apparmor可以对程序要使用多种资源进行限制，如：fsize,d ata, stack, core, rss, as, memlock, msgqueue等，但暂不支持对程序可以使用CPU时间进行限制。
19. 访问网络，Apparmor设置程序是否可以访问网络进行限制：network [ [domain] [type] [protocol] ]，举例说明，比如要允许程序使用在ipv4下使用TCP协议：network inet tcp。最后还可以限制程序进行capability列表里的操作，在linux的手册页里面有一个capablities列表，apparmor可以限制程序是否可以进行列表里的操作，如：capability setgid。
20. 介绍配置文件的编写方法。
21. 前面提到，编写完配置文件后，要把文件放到/etc/apparmor.d这个目录下，其实有更方便的方法，直接在命令行里面用sudo genprof [filename]，比如sudo genprof '/home/361way/apparmor-helper/demoexe'。
22. 该命令建立的配置文件如下。