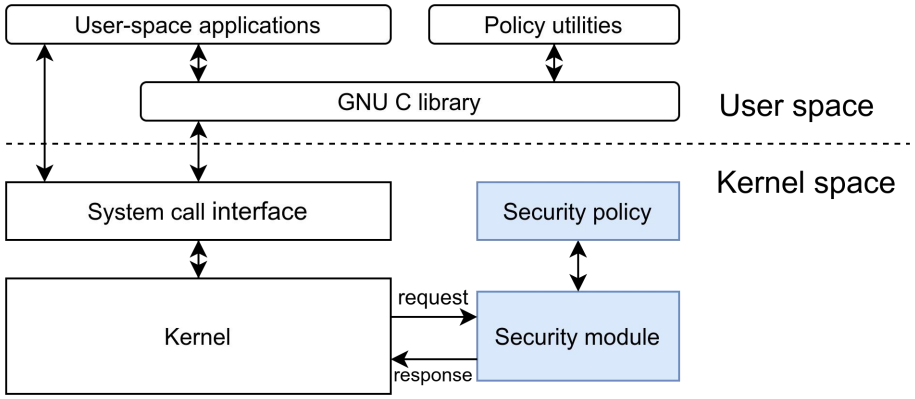
LINUX KERNEL SECURITY FRAMEWORK

Linux内核安全框架

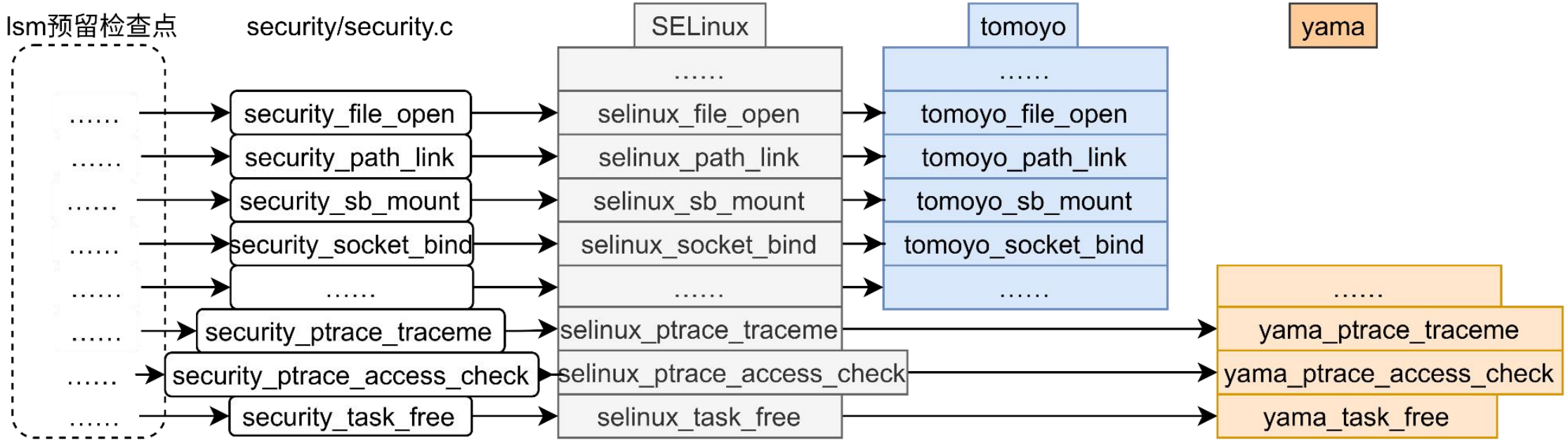
王昱力

LSM安全框架



- Linux内核提供了一套底层的通用内核框架来支持内核安全模块的开发，名为Linux Security Module，缩写为LSM。
- LSM本身不提供安全功能，仅提供基础设施，通过hook技术对一些进程的执行增加一层访问控制策略，开发者可通过加载一个或多个不同的访问策略来实现安全管控。
- 左图为Linux内核安全架构

LSM基础架构


lsm预留检查点
security/security.c
SELinux
tomoyo
yama
……
security_file_open
security_path_link
security_sb_mount
security_socket_bind
security_ptrace_traceme
security_ptrace_access_check
security_task_free
selinux_file_open
selinux_path_link
selinux_sb_mount
selinux_socket_bind
selinux_ptrace_traceme
selinux_ptrace_access_check
selinux_task_free
tomoyo_file_open
tomoyo_path_link
tomoyo_sb_mount
tomoyo_socket_bind
yama_ptrace_traceme
yama_ptrace_access_check
yama_task_free

lockdown内核锁定

- 内核锁定功能旨在防止间接访问正在运行的内核映像，防止对内核映像进行未经授权的修改，以及阻止访问内存中的安全和加密数据，同时不影响驱动程序模块加载。
- 64位x86和arm，如果UEFI开启了安全启动，则lockdown自动开启。

smack强制内存访问控制

- Simplified Mandatory Access Control Kernel
- smack利用LSM安全域将Linux内核中所有主体与客体都打上安全标签，并规定安全策略，只有符合安全策略的访问方式才被容许。
- 主体是指Linux内核进程。
- 客体是指Linux内核客体对象，如文件、消息队列、套接字、共享内存、信号量等，客体也可以是Linux进程或者IPC。
- smack的实现方式与selinux类似，都是通过文件打标签来实现的，但是smack更轻量，性能损耗更小，而功能也更少。

IMA完整性校验

- Integrity Measurement Architecture
- 对正在打开的文件、正在执行的程序、正在执行的共享库和正在加载中的kernel模块和固件进行完整性评估。
- 完整性评估指的是对内核对文件客体在执行特定的内核操作时，会主动对文件的内容进行完整性检查。
- IMA通过读取文件系统的扩展属性检查完整性，而该扩展属性需要提前部署。如果读取后校验结果不一致，会禁止执行。
- IMA支持将度量值写入TPM芯片，是实现可信启动的一条可选路径。
- IMA不能防止文件内容被篡改，只能在执行时检查它是否已经被篡改。

loadpin

- loadpin可确保内核加载的所有文件（内核模块，固件，kexec映像，安全策略）都来自同一文件系统，并期望这样的文件系统由只读设备支持。这旨在简化嵌入式系统（不需要复杂的签名机制），如果系统被配置为从只读设备引导，那么嵌入式系统不需要任何内核模块签名基础设施/检查。
- 安全敏感文件(包括内核模块)保存在只读存储器中，在系统引导之前，对该存储进行整体验证。因此从只读分区加载模块是安全的，无论它们是否已经签名，同时拒绝从其他地方加载模块。
- 当后续文件系统需要加载文件进内核时，都会与引导后第一次加载操作中使用的文件系统进行比较，如何二者不同则操作会被拦截。
- 如果固定加载的文件系统消失，则将完全禁用文件的加载。

tomoyo

- 与基于inode的访问控制系统，如selinux和smack，不同的是，tomoyo是基于名称的访问控制系统。
- tomoyo允许每个进程声明实现其目的所需的特性和资源。在不提供现成的策略时然后自动生成策略文件。

基于inode和基于路径，哪个更安全 ？

- 基于inode的文件安全属性与文件路径无关。文件可以在不同目录间移动，不管它怎么移动，它的安全属性都没有变化。
- 基于inode时文件可以有多个链接，从不同链接访问文件，其安全属性总是一样的。
- 基于inode要求文件系统必须支持扩展属性，并且挂载文件系统时必须使用扩展属性。
- 基于inode时，删除文件时，文件的安全属性会随之消失。再在原先的路径处创建同名文件，并不能保证新文件和老文件的安全属性相同。
- 基于inode时，安装软件和升级软件需要保证系统中新的文件具有正确的安全属性。新文件来自软件包，新的安全属性自然也应该来自软件包。于是有了下一个要求：众多软件包格式也需要支持文件的扩展属性，比如 tar、 cpio 等。

基于inode和基于路径，哪个更安全 ？

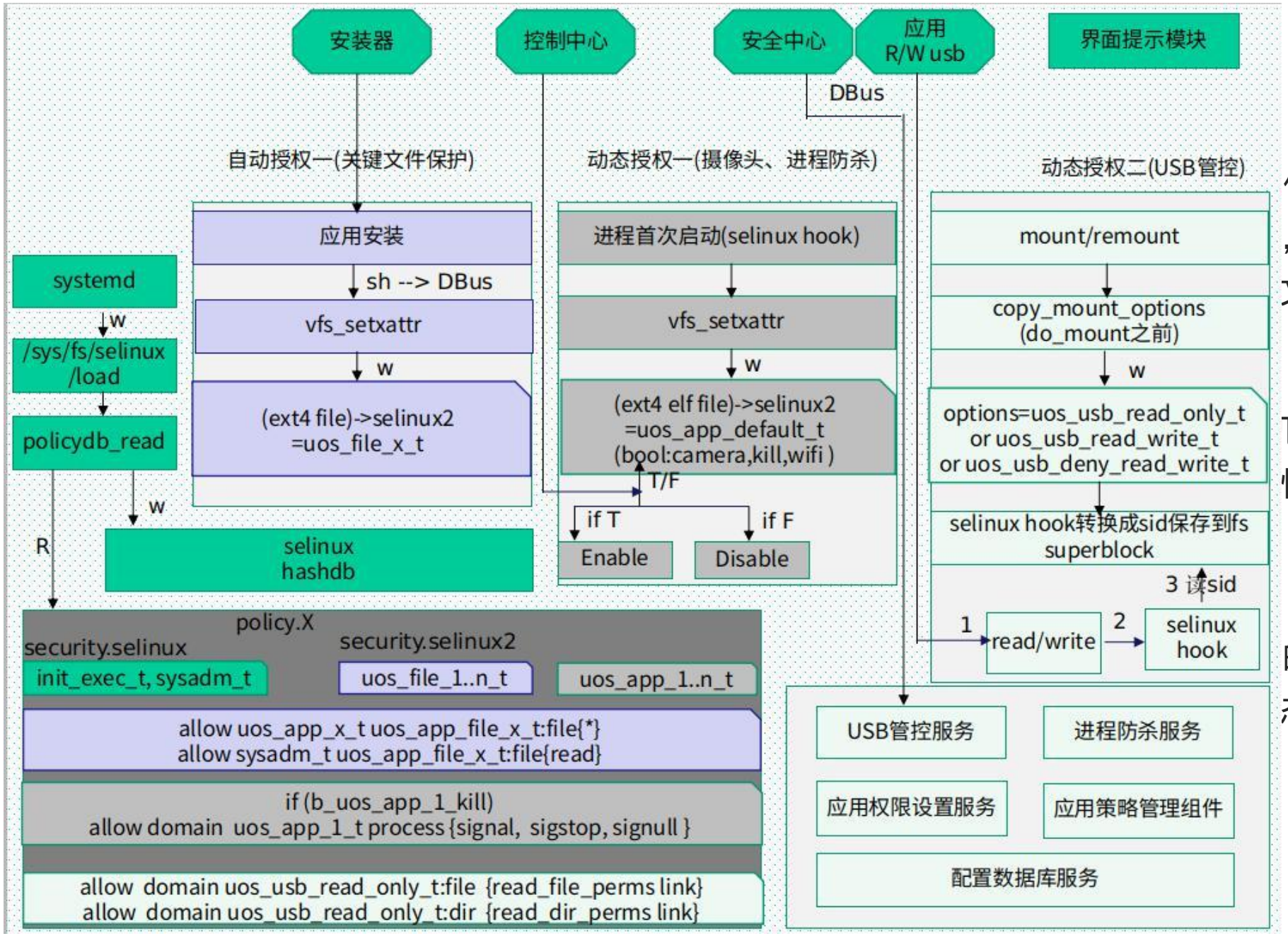
- 基于路径的访问控制，不把这些安全属性存储在文件的扩展属性中，而是在系统内部维护一张表。
- 基于路径时，不需要文件系统有额外支持。
- 基于路径时，不怕文件更新，对打包格式也没有额外要求。用户甚至可以为还不存在的文件定义安全属性。
- 基于路径时，同一个文件可能有多个安全属性，简单地创建链接就可能让文件拥有另一个安全属性。

yama

- yama的目的是对ptrace进行访问控制。
- ptrace是一个系统调用，它提供了一种方法来让父进程可以观察和控制其它进程的执行，检查和改变其核心映像以及寄存器。 主要用来实现断点调试和系统调用跟踪。利用ptrace函数，不仅可以劫持另一个进程的调用，修改系统函数调用和改变返回值，而且可以向另一个函数注入代码，修改eip，进入自己的逻辑。这个函数广泛用于调试和信号跟踪工具。

SELinux

- Security-Enhanced Linux
- SELinux 主要作用就是最大限度地减小系统中服务进程可访问的资源（最小权限原则）。
- 在缺省的 enforcing 情况下，一切均被拒绝，接着有一系列例外的策略来允许系统的每个元素（服务、程序、用户）运作时所需的访问权。当一项服务、程序或用户尝试访问或修改一个它不须用的文件或资源时，它的请求会遭拒绝，而这个行动会被记录下来。



保护、文件共享、IO，最核心的支撑就是文件打标签的基础函

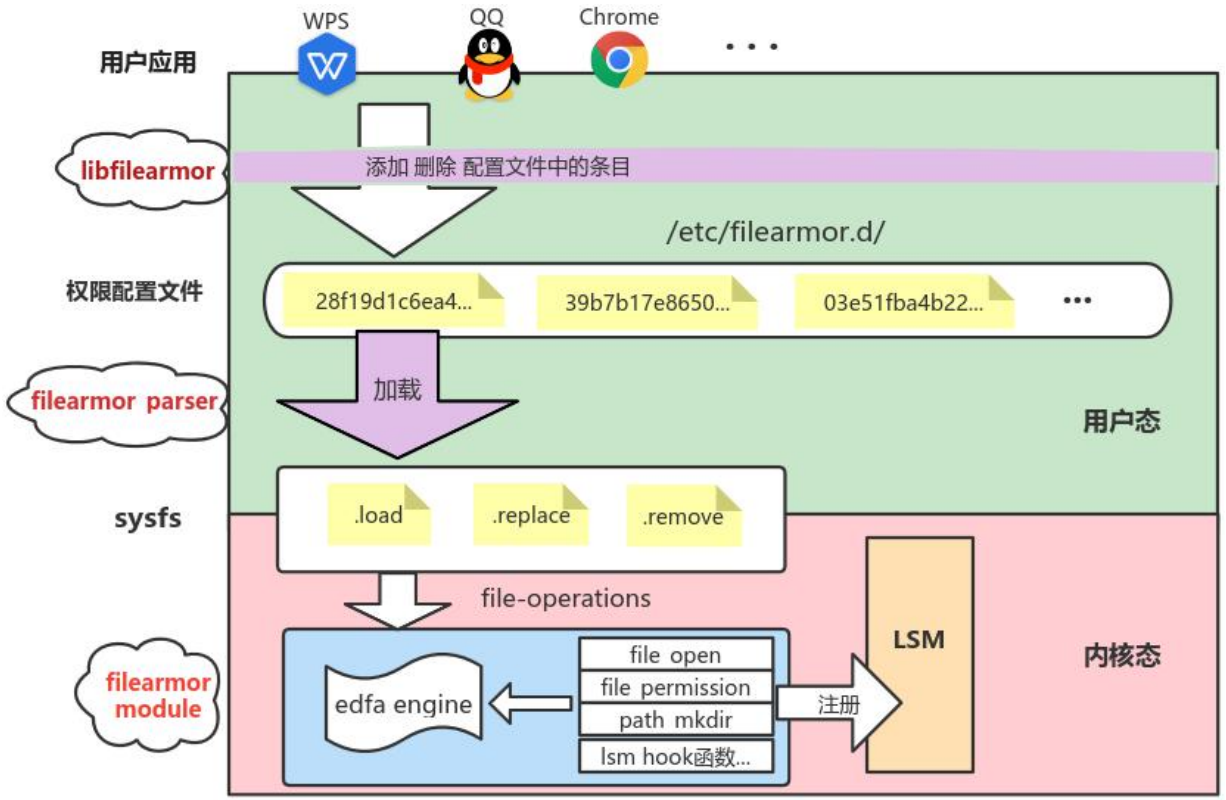
l制，性能影响极大。性能影响极小。

的域写入扩展属性，态申请。

Apparmor

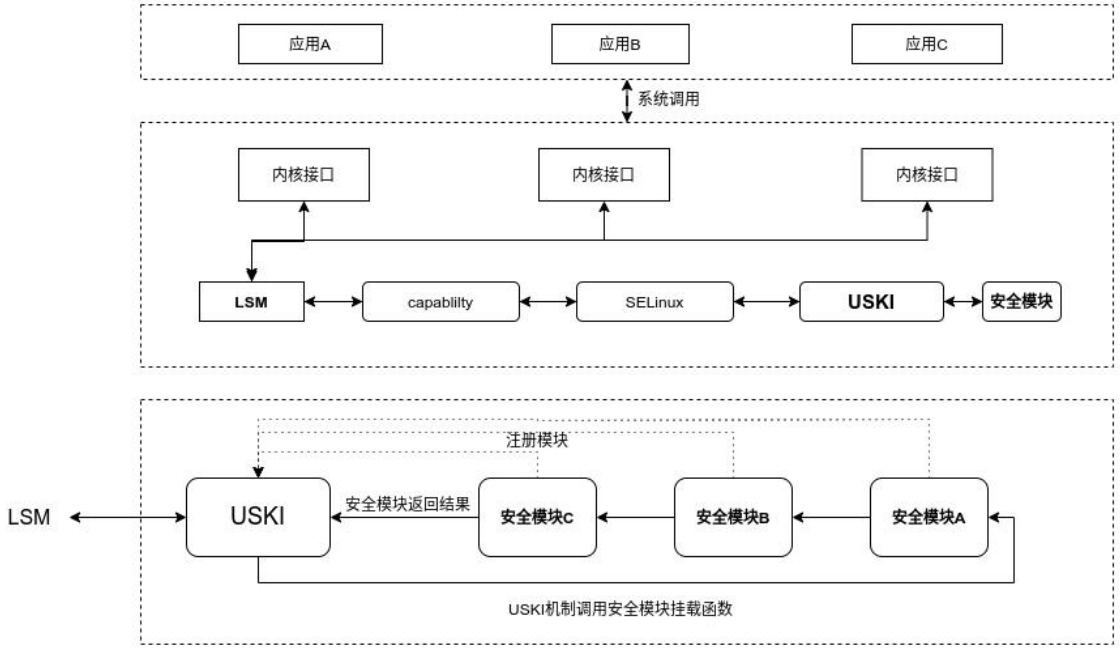
- AppArmor(Application Armor)是Linux内核的一个安全模块，AppArmor允许系统管理员将每个程序与一个安全配置文件关联，从而限制程序的功能。
- 简单的说，AppArmor是与SELinux类似的一个访问控制系统，通过它你可以指定程序可以读、写或运行哪些文件，是否可以打开网络端口等。

统信自研：Filearmor



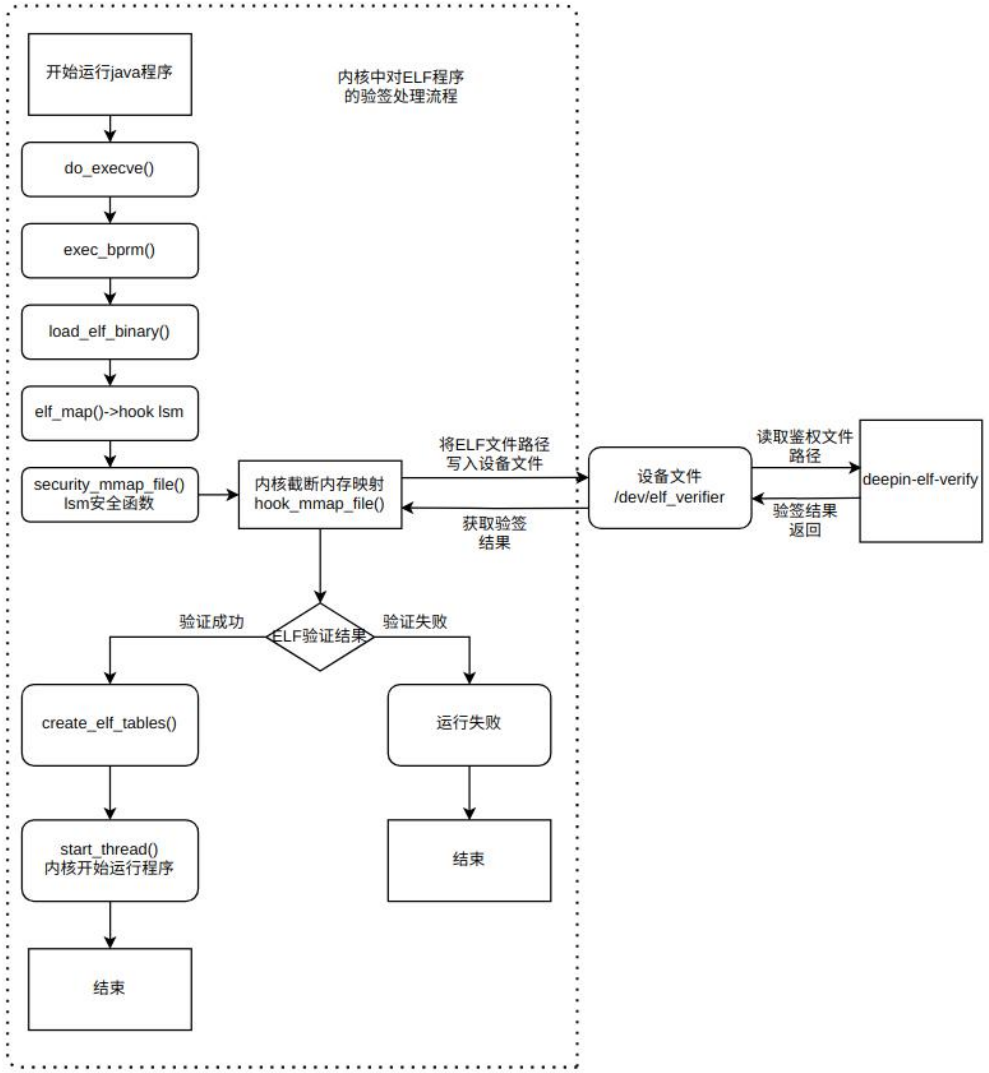
- FileArmor(文件防护)是UOS系统中强制访问控制(MAC)安全系统，用来控制程序和用户对文件的访问。
- 一般在系统启动时，将文件的访问权限配置文件加载到内核。当对文件发生读写等操作时，通过LSM的hook检查文件配置，允许或拒绝应用和用户的操作。
- filearmor兼容selinux等安全模块。
- 同其他安全模块(AppArmor)相比，性能无明显落后。用户使用上不出现明显卡顿。

统信自研：USKI



- UOS Security Kernel Interface
- USKI是统信软件自研的内核态安全接口，提供以ko的形式动态注册第三方安全模块的接口。
- 沿用LSM安全接口确保接口多样性以及稳定性
- 封装LSM调用机制确保LSM变更对开发者无感

统信自研：elfverify



- 用户在操作系统中运行应用软件，内核载入运行的ELF文件，在执行前对其签名信息进行检查，并通知签名验证后台服务对内核中指定的ELF文件进行可信验证，时间戳验证，数字证书链完整性验证，通过之后返回内核态继续执行应用软件，否则阻止应用软件运行。

- 用户在操作系统中安装应用软件的deb包需要通过deb包安装器来执行，deb包安装器会校验用户软件包中的签名信息来判断是否正常安装

THANK YOU

王昱力