DEfensor:
基于 eBPF 的去中心化信息流控制系统

邓浚泉 仇文彬

摘要

去中心化信息流控制（DIFC）是一种访问控制方法，它允许程序编写者控制程序与其它信息实体之间的数据流。当应用于隐私性时，DIFC允许不受信任的程序处理私密数据，而受信任的程序控制该数据的传出。当应用于完整性时，DIFC允许受信任程序保护不受信任的程序免受意外的恶意输入。在上面两种情况下，只有受信任程序中的安全漏洞才可能导致安全问题，从而在普遍情况下做到了对信息流的安全控制。

我们提出了一个新的去中心化信息流控制系统DEfensor，它以进程为粒度控制信息流，以标准读写接口为粒度执行安全策略，简化了DIFC在现有应用中的使用，并解决了过往DIFC系统可移植性差、性能低下、引入内核风险的缺点。DEfensor在内核空间中使用**扩展的伯克利包过滤器**（eBPF）观测进程行为并执行安全策略，在用户空间中运行引用监视器管理信息流状态和安全规则。这一系统被认为是行之有效的。

信息流控制-标签体系

DEfensor使用标签跟踪系统中数据的流通。标签没有固有的含义，但进程通常将每个标记与私密性或完整性相关联。任意一个进程p都有两个标签集，私密标签集S_p和完整标签集I_p。

如果$t \in S_p$，系统就会认为p看到了带有t标签的私密数据。如果$t \in I_p$，则p的任意输入都可以对数据的完整性负责。

信息流控制-去中心化的能力

在传统的信息流控制中，只有受信任的安全管理员才能创建新标签，从私密标签集中移除标签（*降密，declassify*），或将标签添加到完整标签集（*背书，endorse*）。在DEfensor DIFC中，任何进程都可以创建新的标签，以及在一定范围内修改标签。

DEfensor使用两种能力表示权限。对于任意标签t，它拥有两种能力，t+和t−. 每个进程p都有一个能力集合O_p。

如果$t \in O_p$，则p拥有t+能力，它就有权将t添加到它的标签集中；如果$t - \in O_p$，则p拥有t−能力，它有权将t从它的标签集中移除。

信息流控制-安全性

以两个小例子来说明信息流控制安全的思想，严格的数学形式请阅读论文。

保护私密性 Bob希望保存一些私密文件，但恶意的文本编辑器会将私密文件上传到黑客的服务器。通过信息流控制，Bob可以限制编辑器的行为，对所有不可信的进程p运用以下规则：

- 如果p读了他的私密文件，那么$b \in S_p$
- 如果$b \in S_p$，那么p只能向符合$b \in S_q$的进程q或文件q传输信息。
- p不能将标签b从自己的私密集合中移除。
- 如果$b \in S_p$，p不能向不受控制的信道传输信息。（比如因特网）

如果所有这些条件都满足，编辑器就不能从系统中泄露Bob的私密数据。

保护完整性 Bob使用编辑器编辑敏感文件时，担心编辑器不会忠实地按照他的指令操作，而是偷偷地篡改文件内容。他同样可以使用信息流控制，对所有进程p运用以下规则：

- 如果p编辑敏感文件，那么$v \in I_p$.
- 如果$v \in I_p$，那么p不能从不符合$v \in I_q$的进程q或文件q中读取信息。
- p不能主动将v加到I_p中。
- 如果$v \in I_p$，p不能从不受控制的信道接收信息。

如果所有条件满足，Bob知道所有编辑敏感文件的编辑器都是未被污染的。

实现框架

DEfensor系统由一个运行在用户空间的引用监视器和运行在内核空间的eBPF程序组成。（如图1）

引用监视器包含一个标签管理模块，一个引用管理模块，以及与用户进程和系统内核交互的接口。

其中标签管理模块用于管理信息实体的标签，引用管理模块用于管理进程加入的信道以及属性。他们协同运作以保持系统状态。

内核接口接受和分析eBPF程序提交的进程行为，判断系统各进程间能否建立通信，以及进程能否读写某个文件；用户接口则是一个面向用户进程的服务器程序，程序可以通过调用用户接口提供的API函数，按需求创建、更改、分享标签。

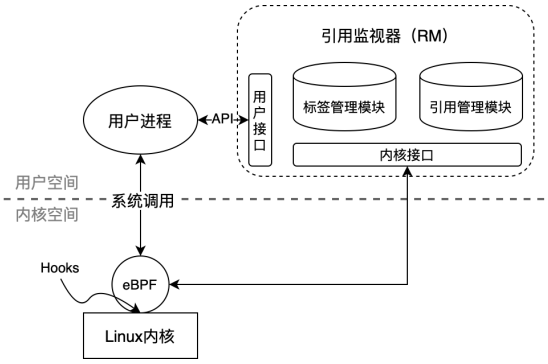

图1 系统框架

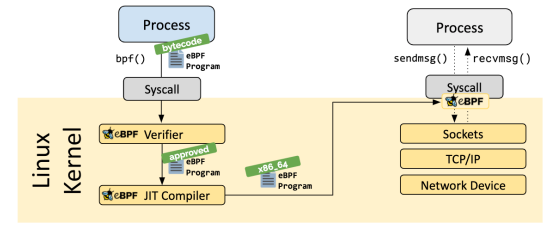

图2 eBPF框架

eBPF

eBPF程序运行在内核沙箱，以高性能提供系统观测和安全策略执行，并提供了良好的可移植性。

eBPF的核心构成是一个内核中的虚拟机、一个即时编译器和一个加载验证器。(图2)eBPF程序编写后，验证器会审查代码以确保其不会向内核引入风险。然后，即时编译器将eBPF程序翻译并运行。这种模型的好处是，开发者能以最小的修改内核的代价实现只有内核空间才有可能实现的种种功能。换而言之，eBPF在**编程限制与内核能力间找到了一个平衡**。

我们的**eBPF观测器**通过eBPF集成的Tracepoint、Kprobe等钩子，抓取系统中所有的进程周期过程，和系统中信道的创建和修改过程，以向引用监视器报告系统中实时的信息实体变动和信道变化过程。

在抓取系统进程行为的过程中，协同eBPF maps和perf_buffer对数据进行存储和打包，发送到用户空间的引用监视器并以异步方式进行解包。

我们的**eBPF策略执行器**的访问控制策略以（信道，访问权限）的二元组为基础。其中，信道由进程的pid和文件描述符fd唯一确定，对应读写情况下的四种可能权限。

eBPF策略执行器基于KRSI技术，挂载在file_permission、inode_permission、ipc_permission等LSM钩子上，在进程打开信道后、访问信道前执行。触发时eBPF策略执行器会检查该进程以及该进程请求访问的信道，查询maps中的访问控制策略是否禁止这次读/写访问，并根据相应规则予以拦截或放行。

联系方式

邓浚泉
武汉大学国家网络安全学院
Email: pvz122@whu.edu.cn

仇文彬
武汉大学国家网络安全学院
Email: 943871117@qq.com

POWERED BY eBPF