LBM（Linux eBPF Modules）是一个专门设计用于在Linux内核中保护外设的通用安全框架，利用eBPF（Extended Berkeley Packet Filter）技术对不同类型的外设进行数据包过滤和恶意行为防御。它的设计目标是提供一个高效、可扩展的框架，覆盖USB、蓝牙、NFC等外设协议，并允许用户通过灵活的规则语言对外设通信进行精确控制。以下是LBM的设计与实现细节：

### 1. ****架构设计****

LBM的核心设计理念是**统一管理不同外设的通信数据包**，通过一个通用的过滤框架来防御潜在的恶意攻击。该框架主要由以下几个部分组成：

* **核心过滤框架**：LBM的核心部分是一个基于eBPF的高性能包过滤系统。该系统负责从不同外设的协议栈中截获数据包，并将它们传递给过滤器进行处理。LBM支持对多种外设协议（如USB、蓝牙、NFC）的数据包进行过滤，并能扩展支持其他协议。
* **LBM模块**：每种外设（如USB、蓝牙）的协议都有相应的LBM模块，它们提供了钩子（hooks），用于在数据包传入或传出时拦截。每个模块可以独立扩展，方便未来的更新和新协议的支持。
* **eBPF程序**：LBM使用eBPF来编译和执行过滤规则。eBPF允许将用户空间的过滤规则转化为内核空间的字节码，直接在内核中执行。通过JIT（即时编译）技术，eBPF程序可以获得极高的执行效率。

### 2. ****数据包过滤机制****

LBM通过在协议栈的**TX**（传出）和**RX**（接收）路径上插入过滤器，确保所有数据包在被发送或接收之前都要经过LBM的过滤规则处理。具体步骤如下：

* **挂钩（Hook）机制**：对于每个外设协议，LBM在其协议栈的关键位置插入钩子。以USB为例，钩子插入到**主控制器设备驱动程序**的传输和接收路径上，这样可以确保所有USB数据包在传输或接收时都会被LBM拦截。
* **协议字段暴露**：为了使用户可以编写基于外设协议的过滤规则，LBM将协议相关的字段暴露给用户空间。以USB为例，LBM会暴露数据包中的**vendor ID**（厂商ID）、**product ID**（产品ID）等关键字段，用户可以基于这些字段编写过滤规则。
* **过滤器调用**：LBM通过eBPF程序执行过滤规则。每个数据包在被处理时，都会被传递给相应的eBPF过滤器。eBPF程序可以基于数据包的内容决定是否允许该数据包继续传输、丢弃或者修改。

### 3. ****高效的过滤规则编写和编译****

LBM设计了一种**高级过滤语言**，用户可以通过这种语言编写规则，指定如何处理不同的外设数据包。这些规则被编译成eBPF字节码，并在内核中运行。过滤规则的编写过程如下：

* **规则语言**：LBM的规则语言类似于Wireshark或tcpdump的语法，用户可以轻松编写规则。例如，用户可以写一个规则来指定只允许特定的USB设备通过：

usb.idVendor == 0x413c && usb.idProduct == 0x3010

这条规则表示只有厂商ID为0x413c且产品ID为0x3010的设备可以通过。

* **编译流程**：用户编写的过滤规则通过LBM的工具lbmtool编译为eBPF字节码。该编译器首先将规则转换为中间表示（IR），然后再生成eBPF指令。最后，生成的eBPF程序通过系统调用加载到内核中。

### 4. ****性能与扩展性****

LBM的设计目标之一是**高性能**和**可扩展性**：

* **JIT编译**：LBM通过eBPF的JIT（即时编译）技术，将过滤规则编译为接近于机器代码的字节码，在内核中以极高的效率执行。测试表明，LBM的包处理开销通常在1微秒以下。
* **扩展性**：LBM框架本身是高度模块化的，用户可以轻松扩展支持新的外设协议。文章展示了如何通过增加少量代码来支持NFC协议。对于每个新的外设协议，LBM需要定义相应的钩子位置、协议字段以及eBPF辅助函数，以便支持过滤操作。

### 5. ****实验与评估****

文章通过一系列实验验证了LBM的有效性和性能：

* **USB过滤器**：通过实验，LBM可以有效防御USB的BadUSB攻击和其他常见USB攻击，如阻止未授权的设备注入恶意数据包。
* **蓝牙保护**：LBM通过在蓝牙协议栈的HCI和L2CAP层插入过滤器，能够防御BlueBorne攻击等蓝牙漏洞攻击。
* **NFC支持**：LBM框架通过少量代码扩展，能够支持NFC协议，实现NFC相关的安全防护。

### 6. ****优势与不足****

**优势**：

* **通用性**：LBM框架支持多个外设协议，用户可以使用统一的接口管理不同协议的数据包。
* **高效性**：通过eBPF和JIT技术，LBM在保证安全性的同时，提供了高效的数据包处理能力。
* **灵活性**：用户可以通过高级语言灵活编写复杂的过滤规则，避免直接编写低级代码。

**不足**：

* **实时性挑战**：在面对复杂的攻击策略或大量外设数据时，LBM可能会遇到实时性问题。
* **依赖Linux内核**：LBM框架依赖于eBPF，适用于Linux内核，其他操作系统的实现可能需要进行较大改动。

### 总结

LBM框架通过eBPF技术实现了对Linux内核中外设的统一安全防护，为USB、蓝牙和NFC等协议提供了高效的过滤机制。它不仅能够防御常见的外设攻击，还能通过灵活的扩展支持未来的新协议。LBM的设计兼顾了性能和安全性，在实际应用中具有广泛的潜力。