[1]基于 EFL 中间件平台的嵌入式灵活安全系统

王小斐[1]　　杨青松[1]　　陈榕[2]

(1. 清华大学信息技术研究院操作系统与中间件技术研究中心，北京 100084)

(2. 清华大学深圳研究生院软件工程中心 深圳 518055）

摘要：

随着嵌入式系统的广泛应用，系统安全问题已经成为一个急需解决的问题。针对嵌入式系统功耗和性能均衡等方面的特点，基于 EFL 中间件平台实现了自适应的灵活安全中间件。该系统基于 EFL 中间件技术和和欣操作系统，提供了一个可灵活配置的构件化安全系统来解决嵌入式系统的安全问题。

关键词：

EFL 安全中间件，嵌入式系统，灵活，ezCOM

中图分类号：TP311.52　　TP302.1

An Scalable Security System For Embedded System Based On EFL Middleware

WANG Xiao-fei, YANG Qing-song

(Department of Computer Science and Technology, Graduate School of Shenzhen, Tsinghua University, Beijing 100084, China)

Abstract:

With the development of embedded systems, system security has become an important problem. The paper constructs a scalable security system--- EFL security middleware for embedded systems. The system is based on EFL middleware techniques, and provides a possibility of framework to settle embedded system's security problems.

Keyword:

EFL security middleware, embedded system , scalable, ezCOM

基金项目：国家高技术研究发展计划（863 计划）项目（编号 2003AA1Z2090）。

清华大学研发基金（编号 Jc2003021）。

收稿日期：2004-11-8

作者简介：王小斐（1980-），男，北京人，硕士研究生，主要研究方向为计算机应用技术，中间件技术，多媒体；杨青松，男，硕士研究生，主要研究方向为计算机应用技术；陈榕，男，清华大学信息技术研究院操作系统与中间件中心副主任，科泰世纪科技有限公司首席科学家，主要研究方向：网络操作系统，构件技术。

1 引言

随着 IT 行业的飞速发展，越来越多的移动式设备被广泛的使用在各个方面。由于空间和电源的限制，移动式设备大多使用嵌入式系统，硬件方面诸如 SOC 芯片，ARM 体系结构，MIPS 体系结构等等，软件方面的嵌入式操作系统：如 Wince、uclinux、ucosII, 等等。嵌入式系统硬件特点是价格便宜、体积小、功耗低；相对的性能也比较有限。由于 CPU 和内存等硬件资源的稀缺，嵌入式系统对运行在其上的操作系统和应用程序有严格的限制。必要时开发人员甚至要关注系统中每个字节的使用，对嵌入式系统进行优化是不可缺少，这样才能满足最基本的任务需求。同时在过去的大多数环境中，嵌入式系统都是被独立的使用，很少处于网络连接的环境中。在这种情况下，嵌入式操作系统的用户是单一的，而且不会存在通过网络的入侵，所以系统安全问题不需要很多的关注。

VLSI 技术的发展使嵌入式系统的硬件性能飞速提升，处理器性能更强、体积更小，内存则容量更大、速度更快，相对的硬件价格更便宜。这个嵌入式系统性能的提升放松了束缚在开发者身上的枷锁，使解决安全问题成为可能。在使用环境方面，随着移动电话、PDA、PVR 等设备的流行和普及，Internet 进入了嵌入式系统的领域。特别是使用手机上网已经是非常普遍的功能，无线接入用户飞速增加。嵌入式系统保存用户很多的隐私，也就提出了对安全问题的需求。确保嵌入式系统的安全性已经提上了议事日程，并且已经具备了一定的基础。

但是与普通的 PC 系统相比，嵌入式系统的性能还是处于相对低的水平。如果投入过多资源解决安全问题，系统的性能将会不可避免的受到很大的影响。这就决定了，嵌入式系统的安全解决方案不能照搬一般 PC 的解决方案，而是必须综合考虑性能和安全性之间的平衡问题。基于 EFL 组件技术的可灵活配置的构件化安全系统可以为嵌入式操作系统提供很好的安全问题解决方案和优秀的框架结构。

EFL 组件技术是和欣操作系统的 ezCOM 构件技术的发展，它吸取了 COM、CORBA 等组件技术的优点和有关组件的一些最新研究成果，在 Linux 操作系统上用 C 语言加以实现。ezCOM 构件技术是面向构件编程的编程模型，超越了面向对象编程技术，面向构件编程规定了一组构件间相互调用的标准，使得二进制构件能够自描述，能够在运行时动态链接。和欣构件操作系统是我国 863 计划中独立研发的嵌入式操作系统。它就是基于 ezCOM 构件技术的 32 位嵌入式操作系统，基于微内核，具有多进程、多线程、抢占式、基于线程的多优先级任务调度等特性。提供 FAT 兼容的文件系统，可以从软盘、硬盘、FLASH ROM 启动，也可以通过网络启动。和欣操作系统体积小，速度快，适合网络时代的绝大部分嵌入式信息设备。作为完全面向构件技术的操作系统。和欣操作系统提供的功能模块全部基于 ezCOM 构件技术，因此是可拆卸的构件，系统和应用程序可以按照需要剪裁组装，或在运行时动态加载必要的构件。这样的特点使得构件技术可以非常简单的灵活配置，实现很多传统操作系统不能完成的模块化配置功能。EFL 组件技术是 ezCOM 的发展，使用 EFL 组件技术可以使 ezCOM 编写的组件在 Windows 和 linux 平台上运行，具备优良的跨平台特性。这些特性使为嵌入式系统开发安全系统成为可能。图 1 所示是 EFL 构件平台的运行框架以及它与操作系统和硬件的关系。

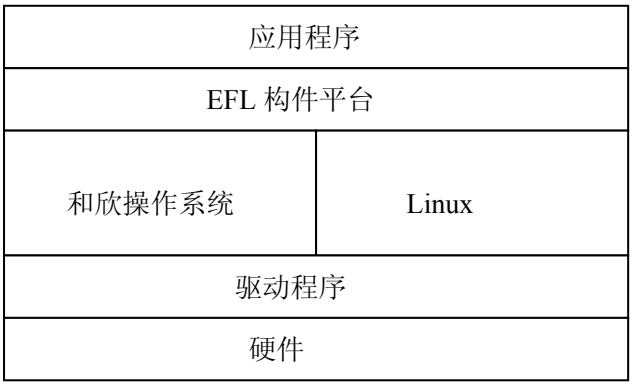

图 1.EFL 构件平台工作结构

2 EFL 安全中间件的框架结构

2.1EFL 中间件具备如下特点：

- 自描述

所谓自描述就是将元数据包含在组件代码中，和组件一同发布出来，这些元数据能够在组件的开发、调试和运行环境过程中利用，完善组件的功能。元数据（metadata）包括组件中使用到的类型声明、函数、接口声明、依赖环境等。元数据有如下的作用：对调用请求进行类型检查、检查接口继承关系的正确性、便于接口声明的发布和安装、为跨语言调用提供类型信息、在没有 IDL 的情况下生成代理代码。

- 动态加载、动态调用

在静态调用的情况下，组件开发者需要事先手动编写或者自动生成代理/存根代码，经过编译后，代理/存根和客户代码、组件代码被链接起来。动态调用无需实现代理和存根，客户程序无需和组件链接在一起，而是在客户程序运行的过程中，动态加载需要的组件，并进行类型检查，调用组件代码。EFL 组件有了自描述信息，组件运行平台就可以在任何时候获得组件的接口声明和类型声明，从而可以进行动态调用。

- 跨平台

EFL 组件主要运行于 Linux 系统之上，二进制代码为 elf 格式。EFL 的原型是 ezCOM，ezCOM 组件可以无缝的运行在 EFL 平台上。而 ezCOM 是微软 COM 技术的发展，这就确保了 EFL 可以兼容微软的组件，二进制兼容 windows。在代码加载器（类似于 wine ）的帮助下，组件可以很容易的移植到其他操作系统上，例如：UcosII、Vxworks、Nucleas 等等。

- 可运行于不同进程空间

EFL 组件遵从一套 API 规范，使得组件运行平台能够方便的将其加载到客户进程、新进程甚至内核中去。

- 高性能和低功耗

与 COM、Corba 等组件系统相比，EFL 组件系统可以在使用较少资源的情况下实现较高的性能，尤其适合嵌入式系统中的应用。

2.2EFL 安全中间件框架结构

EFL 安全中间件模型使用 EFL 中间件的特性使安全系统可以定制化的灵活配置，使用 EFL 的动态加载和调用特性均衡性能和安全需求。EFL 安全中间件的基本目的是更好的分配和使用资源，使得嵌入式系统在有限的资源下取得高安全性，而不影响性能。

EFL 安全中间件探测网络连接和本地节点的运行以及资源使用状态，维护一张资源占用

表。网络连接状态包括带宽、网络延迟、端口监听，等等。本地节点状态包括处理器、内存和总线设备的使用情况，等等。使用这些信息，配合控制策略，EFL 安全中间件的包括五个主要的模块：

- 状态监控模块
- 访问控制模块
- 管理权限分割模块
- TCP(可信路径) 模块
- 残留信息保护模块

图 2 是 EFL 安全中间件的框架结构，展示了 EFL 安全中间件中各个模块的基本关系。

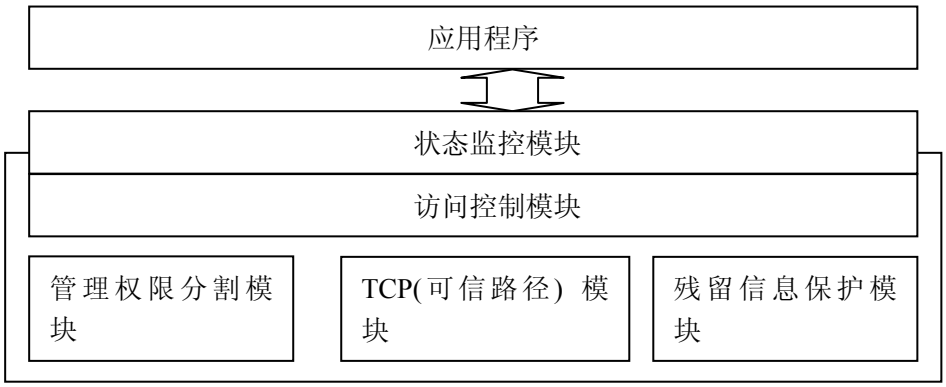

图 2 .EFL 安全中间件的框架结构

入侵就是指试图非法进入或者使用一个计算机系统的行为。入侵可能是盗取机密数据或者是使用你的系统产生垃圾邮件。EFL 安全中间件可以检如何测到这类入侵，并对之进行防范，避免损失。

从图 2 中可以看出，上面描述的五个模块中，状态监控模块和访问控制模块是 EFL 安全中间件框架结构中的核心部分，它们管理和控制了整个中间件，分别是眼睛和大脑。使用调度策略动态加载和运行其他的模块的组件。其他模块调用不同的组件来提供支持和不同的安全保障级别。

3 各模块详细描述

3.1 状态监控模块和访问控制模块

状态监控模块对网络环境和本地节点环境进行评估包括网络带宽、网络延迟、处理器和内存的使用状态。状态监控模块将侦测到的环境数据发送给访问控制模块。访问控制模块作为一个仲裁者，判断系统所处的入侵危险级别。现在的 EFL 安全中间件框架结构中简化入侵危险级别为 2 级：***普通级别*** 和***高危级别***。对于不同的入侵危险级别，EFL 安全中间件为系统加载不同的安全模块，或者为相同的安全模块加载不同的组件来谐调系统安全和系统性能的均衡。在***普通级别*** 状态下，EFL 安全中间件加载较少的模块和较简单的组件，减少安全系统的资源占用，维持整个嵌入式系统运行在高性能状态。在***高危级别*** 状态下，EFL 安全中间件以确保系统安全为第一要务，加载全部模块和较高级的组件，进行入侵检测；这时安全系统的资源占用率提高，系统整体性能降低，但是维护了整个信息系统的安全。使用 EFL 构件自适应和动态特性，EFL 安全中间件得以在系统安全和系统性能之间进行平衡决策，尽最大可能的保证了入侵检测和系统整体性能。

图 3 是两个主控模块的工作示意图，其中入侵模式库是一个独立的组件，不断更新，以

组件形式在网络上发布。

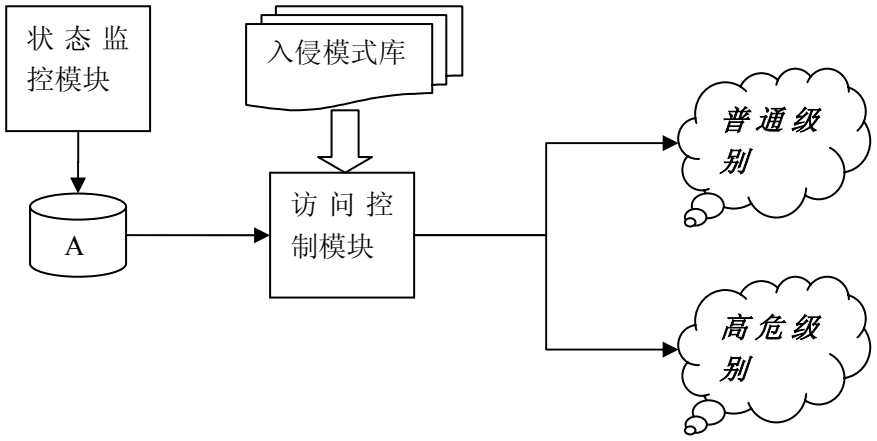

图 3.EFL 安全中间件监控模块的工作原理和主体框架

状态监控模块探测并向访问控制模块传递的数据 A 为一个四维向量，即 A（a0,a1,a2,a3,a4）。a1～a4 分别为网络带宽、网络延迟、处理器和内存的使用率，a0 为网络端口使用状态。

访问控制模块维持一个资源使用列表，记录正常使用状况下系统的资源使用模式。遵循以下规则判断是否入侵，并进行相映处理：

```
{
    如果本地资源使用出现异常，则根据资源列表信息判断是否符合过去的模式。
    如果不符合已有的使用模式，就认为可能是一次入侵，启用入侵模式库组件来扫描。
    IF（确认为入侵）
    {
        阻断网络连接，启动高危级别模式，中断现有工作防止信息丢失。
        同时更新入侵模式库组件，并向网络发布，避免同类入侵发生在其他系统中。
    }
    ELSE
    {
        确认普通级别，释放入侵模式库组件，减少安全系统的资源占用。
    }
}
```

EFL 安全中间件对网络数据包进行检测来判断是否处于一次入侵的状态，通过模式匹配来判断入侵模式。例如侦测到对不同的端口进行大量 TCP 连接请求，就意味着一次 TCP 端口扫描。EFL 安全中间件也可以监控日志和系统变化来探测针对访问权限的入侵，例如威胁文件系统的变化或者系统登陆的不寻常变化。

3.2 其余模块功能和机制

除了状态监控模块和访问控制模块，现在 EFL 安全中间件包括三个主要模块：

3.2.1 管理权限分割模块

该模块对系统的完整性进行检测。模块对入侵者可能造成的文件系统和系统登陆（后门入侵方式）的变化进行监控。如果运行的操作系统存在日志，该模块也负责对日志进行管理。本模块对系统的管理权限进行分割，使每个管理帐户都不能对系统的完整性进行颠覆性的完全破坏。EFL 安全中间件自动加载不同的组件实现 BLP 安全性模型和 Biba 完整性模型。目前，该模块将管理员权限分割给两个特权用户：管理员和安全员；或者分割给三个特权用户：管理员、安全员和审计员。权限划分和该模块的工作流程描述如下。

管理员：安装和使用任何程序在固定的域，不能破坏系统完整性，无法更改系统日志，无法获取系统级文件和信息。该特权帐号的权限被降低很多。

安全员：监控访问控制模块维护的资源分配列表，对跨越域的调用进行控制。维护系统的完整性，确保不会出现由于管理员帐户的入侵，使系统崩溃。如果出现越权的用户就判断为入侵，对其进行屏蔽，并写日志记录其入侵模式。

审计员：负责监控系统日志等文件记录和系统资源不合理分配，从中分析入侵方式，提交入侵模式库进行处理。

特权帐户的分割使每个特权用户都无法得到全部的系统控制权，并且互相监督互相制约，确保系统的完整性和安全性。

3.2.2 可信路径模块

该模块支持本地节点和服务器之间或者本地进程之间的可信路径通讯。该路径与其他通讯路径逻辑上完全独立，防止被监听和探测。通过此路径传递的信息可以确保不被监听丢失。系统安全最简单的方法就是在实现该路径的通讯时停止一切其他的并发线程。触发可信路径模块工作的可能有两个：系统本身保密信息的传输；访问控制模块确认系统处于***高危级别***时，自动使用该模块控制进程域间的通信，防止被监听。

可信路径模块的工作流程如下：

1.加载访问控制模块；

2.查找资源占用表；

3.屏蔽除将要传送的动作之外其他的所有进程的工作，冻结占用的资源。

4.根据描述组件的 xml 文档的设置，决定传输是否需要加密。

5.确立通讯，完成信息传输

6.释放占用资源和卸载的可信路径模块组件

3.2.3 残存信息保护模块

该模块对信息的授权进行监控。为每个要开始运行的构件创建一个伴随其生命周期的标识，来确保任何相关信息不会被后续的构件和进程监听，包括加密信息。实现残存信息保护的基本方法是在进程结束之后对其使用过的内存进行清零处理。在 EFL 安全中间件中使用资源占用表来解决这个问题。每个系统进程完成以后，根据资源占用表在释放资源之前对其进行清零操作。

上述三个模块的运行会很大程度的影响系统性能。所以 EFL 安全中间件通过访问控制模块来动态调节这些模块的使用来均衡性能与安全。

在***普通级别***，EFL 安全中间件加载很少的模块，而且在加载的模块中也使用较简单消耗资源较少的组件，这样系统工作在高性能状态。例如可信路径模块在系统不需要与外界通讯的时候就是不必要的；而当需要与服务器通讯传递需要保密的信息的时候，加载可信路径模块开始保密通讯，同时为防止并发进程的监听，系统的正常工作基本暂停。当状态监控模块探测到具备入侵特征的行为时，自动提升为***高危级别***状态。这时 EFL 安全中间件为已经加载的模块重新升级组件，添加高安全性组件来确保系统完整性和防范入侵可能造成的损害，访问控制模块会关闭危险的网络端口。入侵模式特性信息以组件方式发布，系统自动检测和更新。

即使入侵者成功的侵入系统，EFL 安全中间件可以把危害控制在最小：管理权限分割模块确保入侵不会是彻底的全部管理权限的丢失；可信路径模块确保现在发送的信息最大程度的被保密；残存信息保护模块使入侵者无法使用内存扫描的方式窃取有用信息。对日志文件的检测可以从中提取最近新出现的入侵模式，对入侵模式检测组件进行更新。从而防范新的入侵的发生。

4 组件的开发

4.1 组件开发

EFL 安全中间件组件的开发和 COM 很相似，步骤如下：

1）写组件的 IDL 文件，具体参考 EFL IDL 规范；

2）编译 IDL 文件，得到组件的头文件，组件的 C 语言代码实现框架以及 proxy 和 stub 的代码。如果是进程内组件的话，则不需要 proxy 和 stub 的代码；

3）在上一步生成的代码框架中填充实现代码；

4）编译组件代码，步骤如下：

- 改写组件源文件所在目录下的 Makefile.am 文件，加上生成 so 的指令：

```
lib_LTLIBRARIES = libtcpchannel.la                  //这个必须以.la 结尾
libtcpchannel_la_SOURCES = TCPChannel.c             //组件的所有实现文件名
                                                    //多个文件用空格分开
libtcpchannel_la_LIBADD = -lnsl                     //组件所用到的库
                                                    //多个文件用空格分开
INCLUDES= -I$(top_srcdir)/include $(all_includes)
```

  注意确认组件的源文件所在的目录的上一级目录中的 Makefile.am 文件中有 SUBDIRS 这一项，而且 SUBDIRS 的值里面有组件源文件所在的目录名。

- 执行 build；

4.2 组件自描述文件

组件自描述文件是 EFL 组件的一个组成部分，每个组件都应该有一个与之对应的自描述文件。自描述文件采用 xml 书写。组件自描述文件原则上可以放在任何位置，需要把实际存放位置相对于 EFL 代码所在目录的路径写到 config/info.txt 里面去，否则不能注册。

4.3 注册组件

所有组件在开发完成之后，使用之前，都需要进行注册，具体的过程：

```
cd EFL
cd lib/efl                                          //进入 EFL 代码所在的目录
./updatereg                                         //更新注册组件
```

完成以上步骤，新的组件就可以使用了。

5 结论和展望

EFL 安全中间件基于 EFL 组件技术，EFL 组件技术则源于和欣构件技术。与传统操作系统相比，和欣操作系统具备灵活的模块化结构，便于移植和剪裁等好处，而且是我国拥有自主知识产权的操作系统，其安全性是微软等外国产品无法比拟的。我国政府现在的采购趋势是放弃微软的 windows 转用开源的 linux 等操作系统，但是这些系统的问题是安全性不足。而 EFL 安全中间件构建和欣与 linux 之间的桥梁，强化其安全性。经过 EFL 安全中间件的

强化，安全级别可以提升到 TCSEC B2 级。

与国外同类型的系统相比较，EFL 安全中间件配合和欣或者 linux 的使用也有明显的特点和优势。国外同类型的安全强化系统包括：XT-300 STOP 5.2E；Linux Security Protection System；SELinux；等等。与这些项目相比，EFL 安全中间件基于灵活的构件化的和欣操作系统本身就具备了小、快、灵的特点，尤其适合嵌入式应用。EFL 安全中间件可以在现有的框架基础上很容易的通过下载新的组件进行升级，例如发布和下载入侵模式库组件。EFL 安全中间件不仅是一个可以灵活配置的中间件，而且还是一个模块化设计的安全系统框架。基于这个框架，可以添加或者修改模块来满足不断出现的安全要求，每个模块中的组件也可以灵活的替换来提高性能或者根据不同的应用环境更换算法。EFL 安全中间件为入侵检测研究提供了一个平台。随着信息技术的发展，入侵将会变的越来越复杂，可以灵活配置和扩展的嵌入式系统安全性平台在现在和将来都是必须的。

参考文献

[1] NCSC-TG-010,A Guide to understanding Security Modeling in Trusted System;

[2] Charles P. Pfleeger , Security in Computing , Printice-Hall International, Inc, 1989;

[3] Eric A. Fisch , Gregory B. White, Secure Computers and Network, Analysis, Design, and Implementation, CRC Press LLC 2000;

[4] DoD 5200.28-STD, Trusted Computer System Evaluation Criteria

[5] GB/T 18336.1-2001, GB/T 18336.2-2001, GB/T 18336.3-2001

[6] Intrusion Tolerant Web Servers via Network Layer Controls, Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)

[7] Open-Source PKI on SELinux, Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)

[8] 刑文训, 现代优化算法 [M], 清华大学出版社, 2002