Firmalice - Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware

JAN 12TH, 2016

论文下载：https://www.lastline.com/papers/2015_ndss15_firmalice-2.pdf

- 物联网(Internet of Things)发展迅速。智能电表，智能门锁，智能开关，智能手表，各种智能家居设备都连接在了因特网上。
- 这些所谓的“智能家居“，一旦发生了是发生了意外的错误，或者被有目的的攻击，将直接的影响到现实世界（用户显然不希望你的邻居能用他的手机打开你家的大门）
- 本文侧重于发现这些智能设备中的认证绕过和后门的问题。其主要思想是利用符号执行去分析设备固件中登陆认证相关的代码，得到可以进入特权状态的路径之后，判断这些路径的约束中是否有确定性的约束（只有一种解的约束），如果存在，就可以认定为是后门。

- 如何断定固件进入了特权状态？作者给出了他的分析引擎目前使用的四种策略：
    1. 程序输出类似”AUTHENTICATION SUCCEEDED”的静态字符串
    2. 系统访问/dev目录下的文件(智能锁通过访问/dev下的文件操纵电机进行开关)
    3. 嵌入式设备对特定内存的访问
    4. 由分析人员指定一些特殊的代码位置
- 为了减少符号执行分析时间，在符号执行之前，分析引擎会首先使用静态分析技术，从进入特权状态的点进行反向切片，再对切出的代码进行符号执行。
- 本文作者实现了自己的基于中间语言的符号执行引擎，与现有的符号执行引擎的区别的是采用了Symbolic Summaries和Lazy Initialization两种方法：

Symbolic Summaries:

- 作者使用函数名/一些测试用例 识别出 动态链接库/二进制文件中常用的 strncpy, strcpy, strcmp，memcpy等函数
- 在符号执行到这些函数点时，使用该函数的symbolic summary 直接替代执行过程

Lzay Initialization:

- 有些固件没有操作系统，直接运行在设备上，(我们不可能从固件的入口点直接开始分析)，我们选定的分析的开始的位置之前可能有一些初始化的函数没有运行，为此我们的策略就是如果符号执行引擎读取一个未初始化的内存，那么我们会尝试在整个固件中寻找对该内存进行写入的routine，然后引擎在该点分支成为两个一个无修改继续执行，另一个则先执行routine之后再恢复执行。

- (结合我自己对路由器分析的结果，这个方法应该是有效的：固件在初始化的时候确实会把某些函数和数据指针保存在一个特定的内存地址上，然后调用该函数的时候再读取出来)
- 再来看看作者给出对三个设备的分析结果：
    - Schneider ION 8600 智能电表: IOActive Labs的研究者在2012年BlackHat上指出该设备存在硬编码的后门，然而我们使用Firmalice穷尽了所有分支之后也发现该后门。经过手工分析确认该“后门”是用户合法登陆之后，使用该“后门“可以使用该设备更多的功能
    - 作者又进一步对 3S Vision N5072摄像头和 Dell 1130n打印机的分析，Firmalice成功检测出了这两个设备上的已被公开的后门。