基于配置文件的物联网设备风险识别

南京邮电大学

B21030823王锦程

目录

1 研究背景及意义

南京邮电大学
Nanjing University of Posts and Telecommunications

研究背景及意义

物联网设备大量激增

截至2024年底，全球已有188亿台联网的物联网设备，相较2023年增长了13%，预计到2030年，全球将有超过400亿台联网的物联网设备。

物理网安全形势严峻

最新数据显示，相比于2023年，2024年的物联网漏洞数量大幅增加了136%，已知漏洞的比例从14%上升到33%。因此，自动化的安全风险检测是必要的。

有关物联网设备的第三方组件的配置文件风险识别的研究仍处于空白

目前对物联网设备进行安全风险评估的工作主要聚焦于检测厂商引入的固件代码漏洞和通信协议漏洞。少量针对物联网第三方组件安全研究的工作也都只是对第三方组件中的代码漏洞进行检测。目前仍未有工作关注物联网设备中第三方组件的配置文件设置不当所引入的安全风险与性能损耗问题。

主要工作

任务1

对不同类型的物联网设备配置相关的安全问题进行总结分类。

任务2

通过动态、静态分析，从固件中提取配置文件。使用FirmAE、QEMU等工具，实现针对物联网固件配置文件进行提取和分析的系统。

任务3

搭建IoT配置文件相关的安全风险识别系统，利用自然语言处理、大模型分析方法，自动化识别其中的安全和性能风险。

2 研究思路与方法

南京邮电大学
Nanjing University of Posts and Telecommunications

系统架构

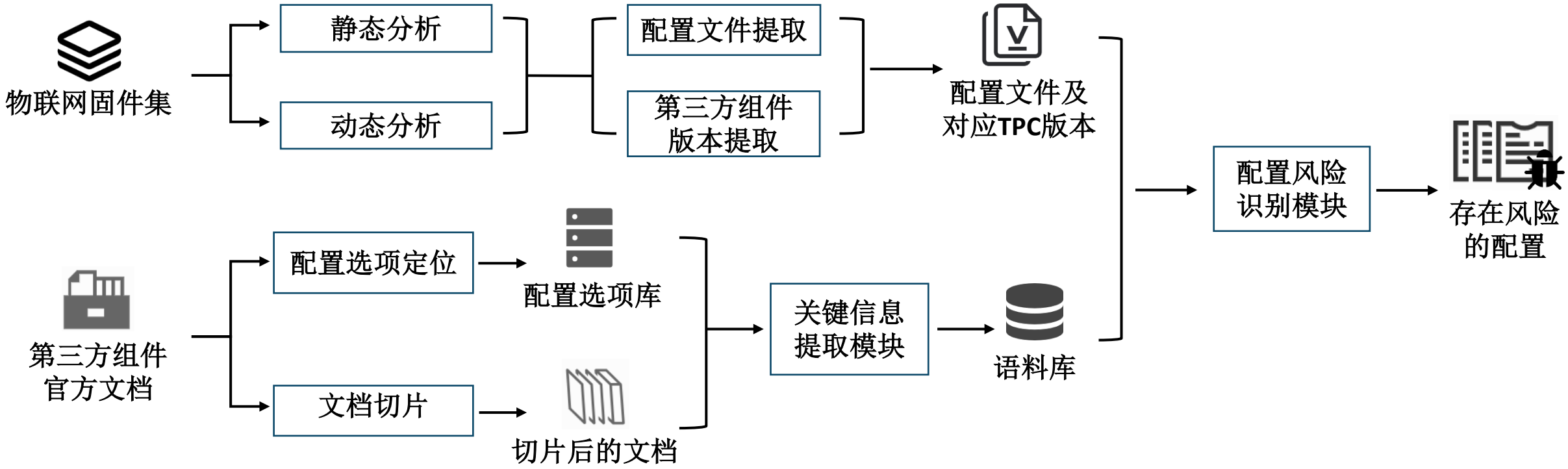

图1. 系统框架图

基于固件仿真的配置文件提取

物联网设备中**配置文件的全面提取**是本研究的前提。

静态分析：解包固件镜像文件，提取出文件系统，根据第三方组件规定的配置文件名，定位并提取出固件中的配置文件。

动态分析：通过设备自带的接口进入内部系统，从运行时的文件系统中根据第三方组件配置文件名称，定位并提取出配置文件。

从真实设备的运行环境中提取到的配置文件数量**多于**从固件中解包得到的配置文件数量。此外，部分从真实设备中提取出的配置文件的内容相比静态提取的配置文件也**发生了变化**。

↓

大量地采购真实设备进行动态分析是不合适的，**无法实现大规模、批量化的风险识别**。

↓

固件仿真：在虚拟环境中模拟物联网设备固件的运行，以实现对其动态行为的分析与测试。

基于固件仿真的配置文件提取

FirmAE是目前最先进的物联网固件全系统仿真工具，其目的是为了动态地验证和检测制造商在固件中引入的**代码漏洞**。然而，这些漏洞大多存在于 Web 服务器中。
因此，**FirmAE 只需要保证WEB 服务正常启用**（通常只有一个程序，例如 httpd），而不再关心其他程序，尤其是第三方组件是否正常运行，即**只完成了部分仿真**。

优化FirmAE，考虑固件的完整启动流程：内核在文件系统中找到初始化程序（例如 /sbin/init），并将其作为用户空间中的第一个程序启动。之后，执行一系列初始化操作以完成设备启动。但是，由于**仿真环境和物理环境的差异**，在缺乏硬件支持的情况下，**初始化程序可能无法成功执行**。

针对初始化程序中断，提出如下措施使固件启动流程得以继续进行：

（1）模拟初始化程序**解析 /etc/initab 文件**的过程，从中提取出初始化脚本，并按次序执行；

（2）遍历**/etc/rc.d目录**下的启动脚本，并按文件名的数字顺序调用对应脚本的 start 或 boot 参数执行；

（3）ubus是基于OpenWrt开发的物联网固件的核心组件，持续检测 **ubusd 进程**是否存活，若不存在则立即重启并挂起该进程。同时，利用同样的方法确保OpenWrt的**守护进程 procd** 保持运行状态。

第三方组件的版本提取

主流的第三方组件都会由其开发团队进行定期的维护及功能更新。因此，**第三方组件通常会存在不同的版本号**。由于不同版本的第三方组件功能存在差异，某些第三方组件**针对于不同的版本号会编写不同的官方文档**，或在同一个文档中描述某些配置选项适用的版本及不同版本间的用法变化等。

第三方组件是由固定的**二进制程序**启动的，其通常也会有固定的名称，故可以根据程序名称在固件文件系统中找到特定第三方组件所对应二进制程序的路径。

在第三方组件对应的二进制程序中，其内的字符串会包含第三方组件的版本信息。这些版本信息通常会**跟随在特定字符串（如version，第三方组件名称）之后**。

使用 **grep -i** 命令在第三方组件的二进制文件中**搜索字符串version和第三方组件的名称**。然后，通过正则表达式匹配并提取这些字符串后的第三方组件版本号。

```
$ strings nginx | grep -i version
SSL_get_version
SSLeay_version
libinjection_version
nginx version: nginx/1.12.2
  -v               : show version and exit
```

图2. 第三方组件版本提取示例

配置风险问题分类

从导致效果的角度，可分为： 安全威胁问题 和 性能损耗问题 两类

从产生原因的角度，可分为以下三类。

1. 使用了官方文档中不推荐或禁止使用的配置方式

It is **strongly recommended** that the ScoreboardFile path **not be** located on a networked filesystem.

ScoreboardFile /etc/service_conf/proftpd
存在**性能**风险

2. 配置选项的取值不在或部分超出文档规定的范围

You should **only limit FPM to .php extensions** to prevent malicious users to use other extensions to execute php code.

security.limit_extensions= **.html .htm** .php .php3 .php4 .php5 .php7　　存在**安全**风险

3. 配置文件中的存在依赖关系的选项组合所产生的风险

chown_uploads: If enabled, all anonymously uploaded files will have the ownership changed to the user specified in the setting **chown_username**. **Using "root" for uploaded files is not recommended**!

chown_uploads=YES
chown_username=root
存在**安全**风险

基于大模型的关键信息提取

应对大语言模型（LLM）的**幻觉问题**：引入**检索增强生成（RAG）**技术，收集不同第三方组件的官方文档，当LLM识别配置文件中的风险时，会查找相关文档内容进行判断，而不是仅仅依赖固有知识。

应对大模型的**Token限制（文本长度限制）**问题：对**文档切片**，将过长内容分为多个大小近似的分片。

即使将文档切片后，内容仍较为冗长，其中包含了大量非关键信息。因此，需要从中**进一步提取出与物联网配置风险相关的三类关键信息**。在后续风险识别模块，LLM能够根据最准确的信息作出判断。

表1. 配置风险及关键信息特征对照表

配置风险	关键信息特征
使用不推荐或禁止的配置方式	包含**语气强烈**的表达，包括强调词、大写表示和否定表达
选项设置的值与规定的范围不一致	描述**取值集合、取值范围**或某些不能取的值
多个选项之间的依赖关系未得到满足，或者组合产生风险	某些选项间有**设置顺序**，或某些选项要求其他选项需要**特定的设置**

文档中配置选项的定位

某些配置选项的名称为**常见的单词而非专有名词**，如密码字段password等。大模型在对文档切片进行关键信息提取的过程中，常常无法准确分辨出此类配置选项及对应描述。

因此，需要先在文档中定位配置选项，再**将提取出的配置选项名称与文档切片一起提供给大模型**，从而提高大模型进行关键信息提取的准确率。

被**下表的HTML标签包裹**并且纯文本位于**行首**位置的文本内容被认为是文档中的配置选项。

表2. 配置选项相关的HTML标签汇总表

标签类型	具体标签或位置
标题类标签	<h1>, <h2>, <h3>...
定义类标签	<dl>, <dt>, <dd>...
强调类标签	<b>, , <code>...
列表类标签	<p>下的第一个
表格类标签	<td>下的第一个<tr>

<DT><B>debug_ssl</B>

<DD>
If true, OpenSSL connection diagnostics are dumped to the vsftpd log file. (Added in v2.0.6).

debug_ssl
If true, OpenSSL connection diagnostics are dumped to the vsftpd log file. (Added in v2.0.6).

Default: NO

图3. 文档中配置选项示例图

基于大模型的关键信息提取

PROMPT	PROMPT
Role: You are an expert at identifying configuration file risks in IoT firmware. You will receive the selected configuration options and official specification for a certain IoT configuration file. <selected_configuration_options> {**selected options**} </selected_configuration_options> <official_specification> {**related specification**} </official_specification> You need to complete the following questions based only on the information given. 1. Please find **the description with a stronger tone** (e.g. emphasis words, capitalization, negative expression) in the given official specification. 2. Please find statements in the given official specification that describe a **range or set of values** for options, or the option **cannot take certain values**.	3. Please find statements in the given official specification that describe the **dependencies** between options. For example, some options have **a setting order**, or some options **restrict or require specific settings of others.** Please give each answer in the following format, one line per answer. **[configuration option name] [question category number] [related original text]** Last but not least, please abide by the following rules. 1. [question category number] can be only selected as [1] or [2] or [3] according to the question numbers above. 2. If [question category number] is [3], you need to give multiple options with dependencies in [configuration option name], separated by commas. 3. [related original text] can contain multiple sentences and must be **completely consistent** with the given specification.

图4. 关键信息提取模块的大模型提示词

基于大模型的物联网配置风险识别

PROMPT	PROMPT

Role: You are an expert at identifying configuration file risks in IoT firmware.
 	
You will be provided with the name and version of a third-party component, its configuration in the IoT device and the official configuration specifications related to it.
You need to identify **the security and performance risks** in the given configuration based on official specifications and its version.
 	
<name> {**tpc_name**} </name>
 	
<version> {**tpc_version**} </version>
 	
<configuration_content>
{**config_lines**}
</configuration_content>
 	
<official_specification>
{**critical_descriptions**}
</official_specification>
	Each official specification is given in the following format. For “category number”, [1] stands for ..... , [2] ..... , [3] .....
	(Serial number) [configuration option name] [category number] [related official specifications]
	
	If no risks are identified, output: "**Correct configuration.**“
	Otherwise, please provide each of final risk identification results in the following JSON format.
	{
	“Effect (Security or Performance)": "...",
	"Inappropriate configuration": "...",
	"Related official specification (**Serial number**)": "...",
	"Explanation": "...",
	"Corrective measures": "..."
	}
	
	Last but not least, please MUST abide by the following rules.
	1. Please output the configuration settings that are **DEFINITELY risky**, and do not give the ambiguous ones.
	2. ONLY judge based on the official specifications given, **NOT on your inherent knowledge**.

图5.配置风险识别模块的大模型提示词

基于大模型的物联网配置风险识别

自我反思机制：对LLM给出的风险识别结果进行初步检查，并让LLM针对性地修正其中错误。

（1）LLM给出的答案包含基本格式的错误，如不是JSON格式，或缺少项；
（2）LLM给出的不恰当的配置为空，或不存在于给定的配置内容中；
（3）LLM给出的序列号格式不正确，或不在给定范围内；
（4）LLM给出的答案，对于某条错误配置对应多条风险，即多个序列号；
（5）LLM给出的错误配置包含多行，但给出的规范不是类型3，即未违背选项之间存在依赖关系的规范；
（6）LLM给出的不恰当的配置中的配置选项与所提供的判断依据（给定的规范）中的配置选项不一致。

自我一致性策略：通过多次向LLM问询相同的问题，并选择最一致的答案来提高LLM回答的准确率。

- 自我反思机制第四点的检查：规定LLM给出的每个JSON格式的结果只能对应一条风险问题，即同一个不恰当的配置可能存在多个风险，需要将其输出为多个答案。这是为了方便自我一致性的判断。
- 对于LLM的每一个输出，当**不恰当的配置内容**与从相关第三方组件的官方文档中提取出的**关键信息**的组合一致时，则将其视为同一个风险问题，即计入一致的答案。
- 对 LLM 进行三次相同的问询，只有三次答案中都出现的风险识别结果才能作为最终的答案。

3 实验设计与评估

数据集及环境参数

物联网固件数据集：通过厂商的官网、论坛等公开途径，收集了来自D-Link、Netgear、Tenda等15个主流厂商的50个物联网固件，包含路由器、摄像头、中继器等5种类型的物联网设备。

第三方组件官方文档数据集：通过编写网络爬虫，从10个常见第三方组件的官网、Github仓库中收集了162个官方配置文档的网页文件和示例配置文件。其中，一个第三方组件可能包含多个不同版本的配置文档，每个版本的配置文档可能包含多个网页文件或示例配置文件。

本文实验在一台运行Ubuntu 24.04 LTS系统的机器上完成，配备有AMD Ryzen 5 7500F六核的CPU处理器、128G内存、一张NVIDIA 5060Ti的16G显存显卡。

通过大模型运行框架Ollama在本地部署了一个**deepseek-r1:14b**的大语言模型用于实验，且**温度参数设置为0.5，平衡了随机性与确定性**。

使用QEMU及Python搭建了固件仿真的框架，使用Python结合LangChain框架编写了基于大模型的配置文件风险识别模块的代码。

固件仿真测试结果评估

评估指标：

固件仿真成功率：仿真结束后，显示网络可达的固件占所有固件的比率；

存在配置文件动态变化的固件占比：产生动态变化的配置文件的固件占所有固件的比例；

动态变化的配置文件总数：数据集中所有固件在动态运行的过程中新增、修改或删除的配置文件总数

表3.本文的固件仿真框架与FirmAE的仿真效果对比表

	FirmAE	本文的固件仿真框架
网络可达的固件数量	11	44
固件仿真成功率	22%	88%
存在配置文件动态变化的固件数量	6	40
存在配置文件动态变化的固件占比	12%	80%
动态变化的配置文件总数	58	329

本文对FirmAE的优化方案在“提升固件仿真成功率”和“增加发现动态变化的配置文件”方面均有良好的效果。

配置文件风险识别结果评估

评估指标：记系统识别结果中正确的数量为TP，错误的数量为FP，遗漏的存在风险的配置数量为FN

精确率： $\text{Precision} = \frac{TP}{TP+FP}$

召回率： $\text{Recall} = \frac{TP}{TP+FN}$

F1分数： $\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

结合静态分析和动态分析，提取与所收集第三方组件相关的配置文件**475个**。

基于大模型从第三方组件官方文档中提取出**15509条**关键描述信息。

基于大模型的物联网配置文件风险识别系统共检测出**803个**存在风险的配置问题。

表4. 物联网配置风险识别系统的性能指标评估表

正确的配置风险数量（TP）	错误的配置风险数量（FP）	遗漏的配置风险数量（FN）
617	186	143
精确率（Precision）	召回率（Recall）	F1分数（F1-Score）
76.8%	81.2%	0.79

本系统具有较强风险感知能力，配置风险识别结果可靠。

统计与发现

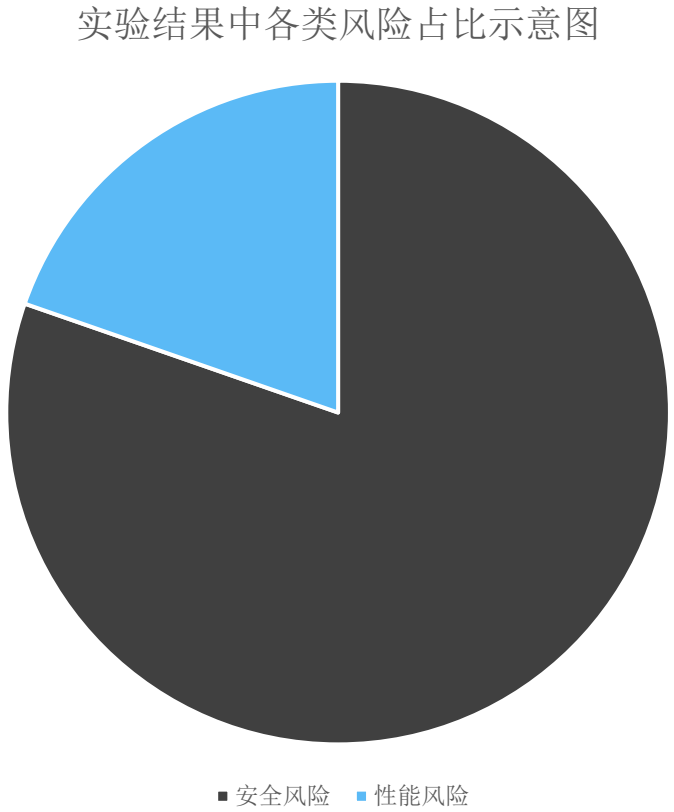

图6.实验结果中各类风险问题占比示意图

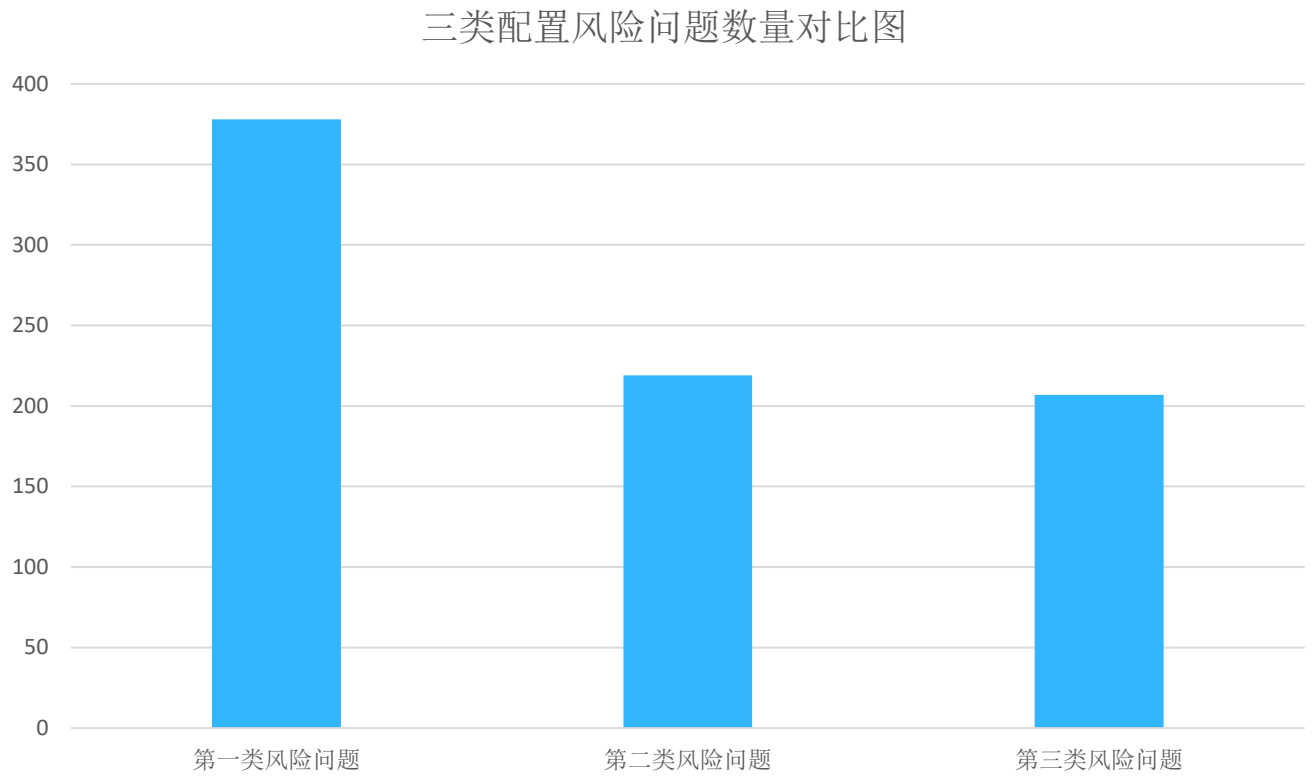

图7.实验结果中三类配置风险问题数量对比图

安全风险的占比远高于性能风险。
这表明大部分开发者在对第三方组件进行配置时，往往只会关注其是否能够正常运行，而忽视了可能带来的安全隐患。

第一类风险问题，即使用了第三方组件文档中强烈不推荐或禁止的配置方式而产生的风险问题数量最多。

4 讨论与未来展望

讨论与未来展望

系统及实验设计的局限性

- 在静态分析方面，无法使用binwalk解包厂商定制**加密或压缩**的固件。未来可以尝试总结部分常见的加密方式和压缩算法，尽可能解开更多的固件。
- 在动态分析方面，目前不支持一些**少见架构**的物联网固件仿真，如aarch64等架构。未来可以尝试定制相关架构的内核，从而支持仿真更多架构的物联网固件。
- 在基于大语言模型的物联网配置风险识别模块中，**大模型的幻觉、理解力不足**等开放性问题在一定程度上会影响风险感知结果的准确性与稳定性。未来可尝试更多先进的提示工程技术或微调模型，进一步优化。
- 在实验方面，目前只使用了deepseek-r1:14b大模型，未来可选用**更多先进的大模型**进行对比分析，从而选取出更适合本课题任务的大语言模型。

未来可能的研究方向

- 可扩展本研究的方法用于**其他主流开源软件、商业软件以及云服务器产品**中的配置文件进行风险识别。通过结合各自的领域知识，并选择合适的大语言模型，利用先进的提示工程技术或微调模型提高准确率。
- 未来研究可进一步探索**配置风险的自动化验证方法**。在本研究基础上，对识别出的物联网配置风险进行自动化地验证，并给出每条风险问题可实际触发的风险等级。

感谢各位专家批评指正！

南京邮电大学

B21030823王锦程