

通用性接口模糊测试技术研究

目录

CONTENTS

01 研究内容介绍

02 模糊测试工具实现

03 实验分析

04 总结与展望

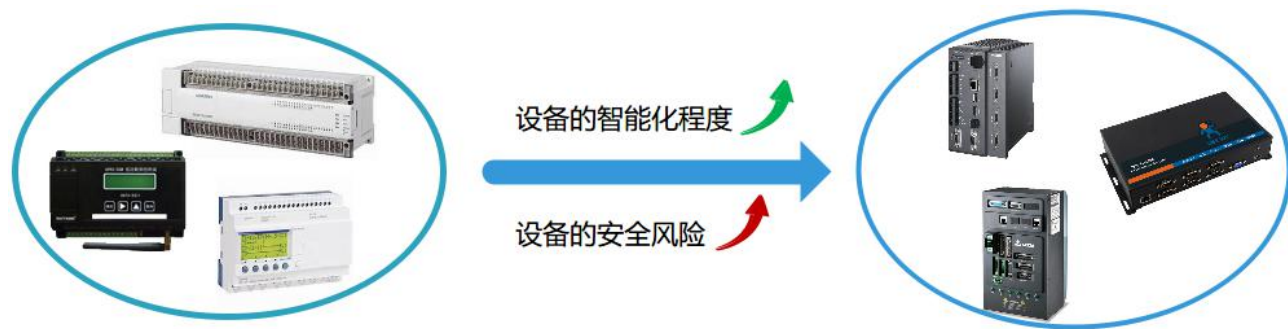
A thick, light blue curved line starts from the top left, curves around the left side of the slide, and then curves back towards the bottom right, creating a large, open shape that frames the central text.

01 研究内容介绍

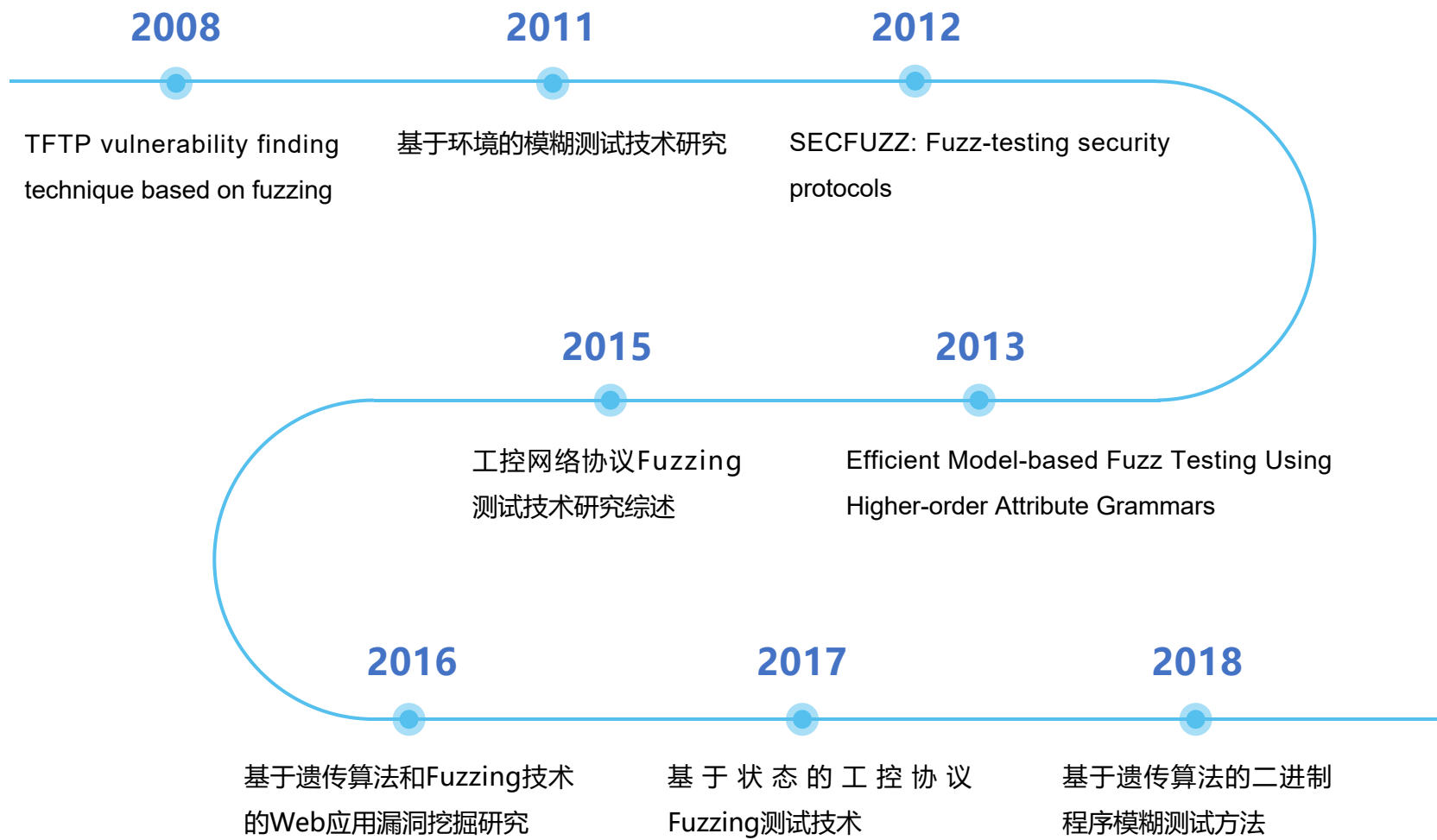
Introduction of Research

研究背景

- 工业控制设备变得越来越复杂，采用的通信接口越来越多，其面临的安全风险也越来越大。
- 而目前对非网口通信协议的模糊测试还没有一个比较好的方案。



相关研究



A thick, light blue curved line starts from the top left, curves around the left side of the slide, and then curves back towards the bottom right, creating a large, open shape on the left side of the slide.

02 模糊测试工具的实现

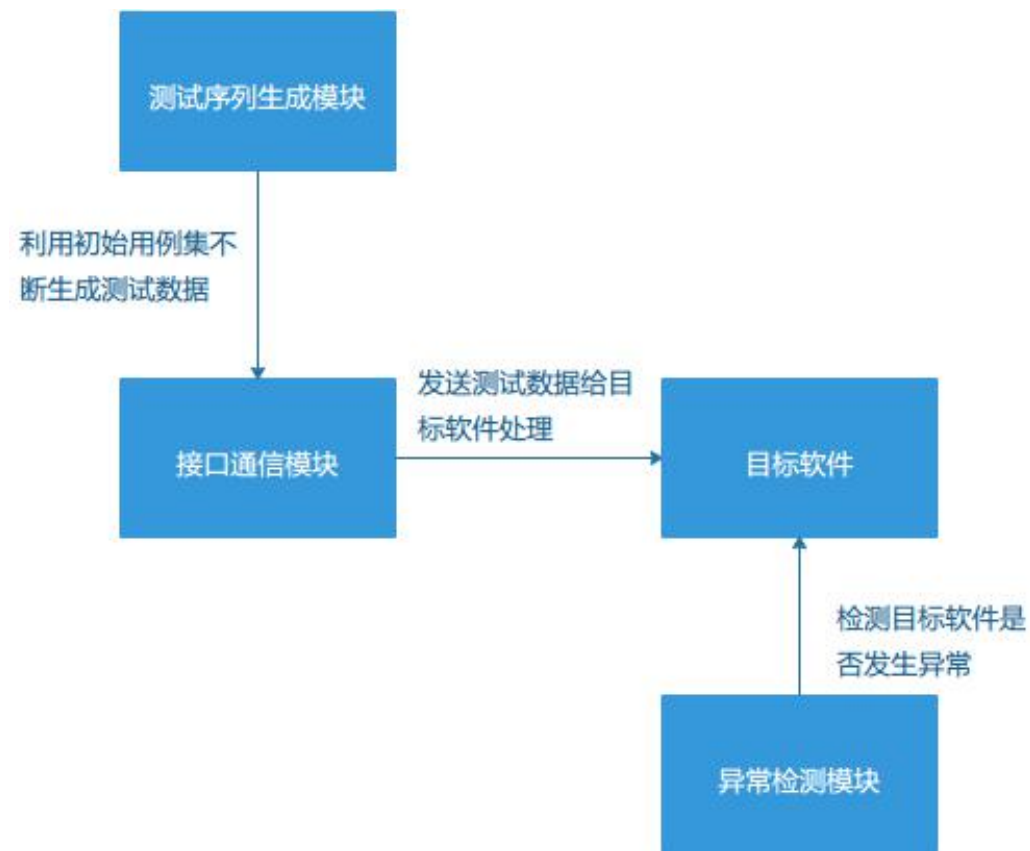
Implementation of Fuzzer

模糊测试工具设计

开发语言&运行平台



模糊测试工具的模块



接口通信模块

为了保证可扩展性，不同接口的通信模块均向上层提供一样的接口

```
def recv(self, size, timeout=2.0):
```

接收数据，最多接收 size 大小的数据

返回接收到的数据

```
def recv_until(self, need, timeout=2):
```

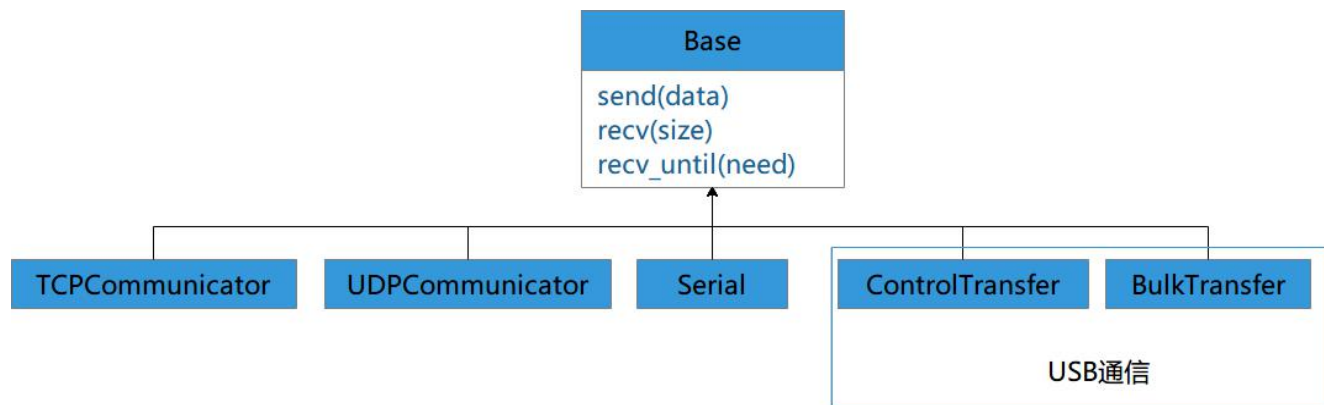
接收数据，直至收 need 字符串为止

返回收 need 字符串为止所有接收的数据

```
def send(self, data):
```

发送数据

返回发送的数据长度

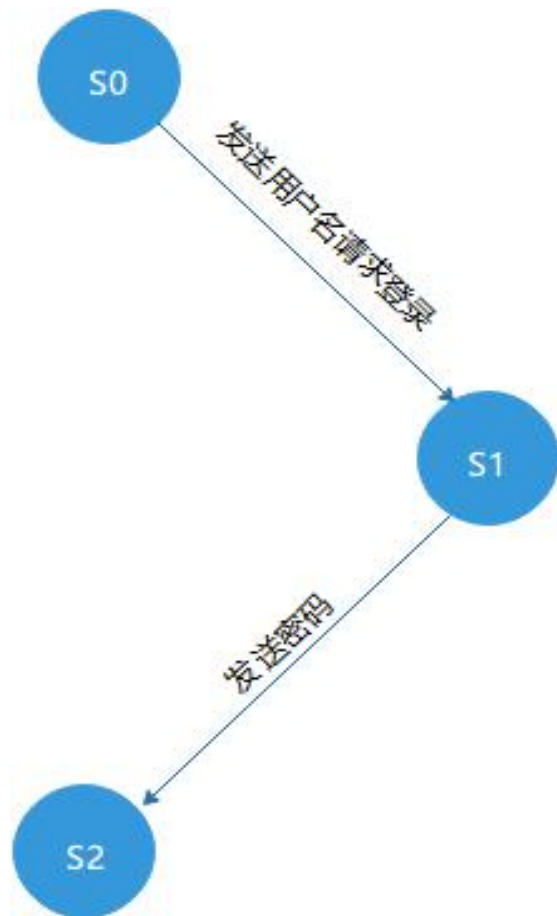


测试序列生成模块

协议的通用表示

通信协议本质上是一段或者多段二进制序列的组合。

$$\text{协议}P = \begin{cases} Q_s = \langle Q_{s1}, Q_{s2}, \dots, Q_{sn} \rangle, & \text{发送的序列} \\ Q_r = \langle Q_{r1}, Q_{r2}, \dots, Q_{rn} \rangle, & \text{接收的序列} \\ I = \langle (Q_{s1}, Q_{r1}), (Q_{s1}, Q_{r1}), \dots, (Q_{sn}, Q_{rn}) \rangle, & \text{协议的交互过程} \end{cases}$$

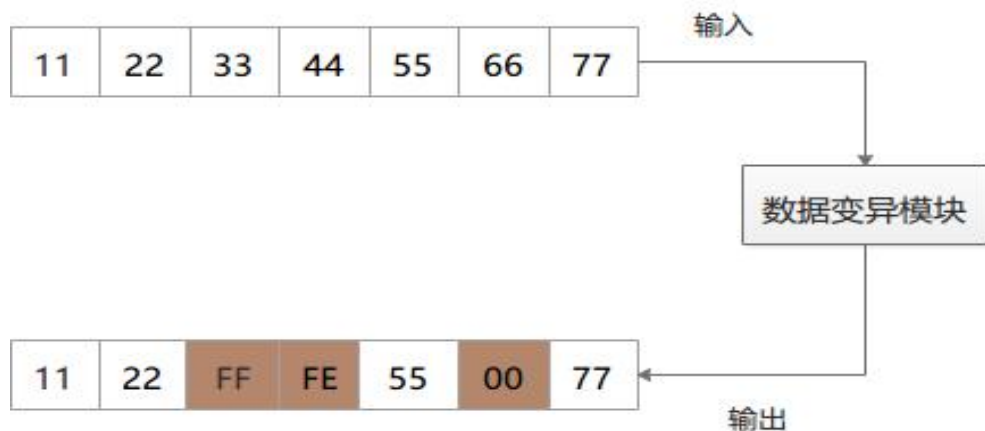


FTP协议的状态切换

测试序列生成模块

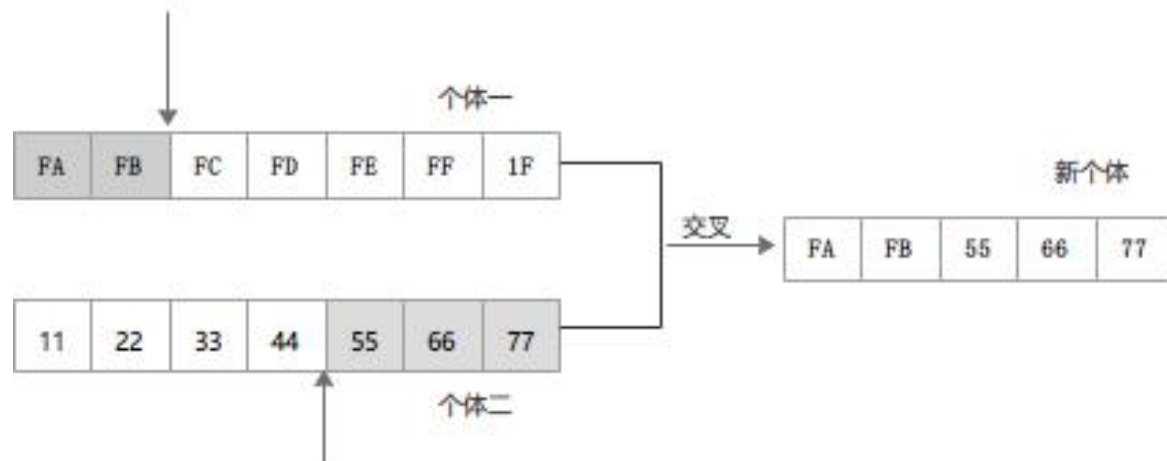
二进制序列变异策略

以二进制序列为变异对象可以保证变异策略的通用性



基于遗传算法的数据变异方案

由于触发漏洞的样本与正常样本之间往往差异不大，因此我们通过使用遗传算法来引导数据变异的过程，使得测试样本往与正常样本相似度大的方向去变异。

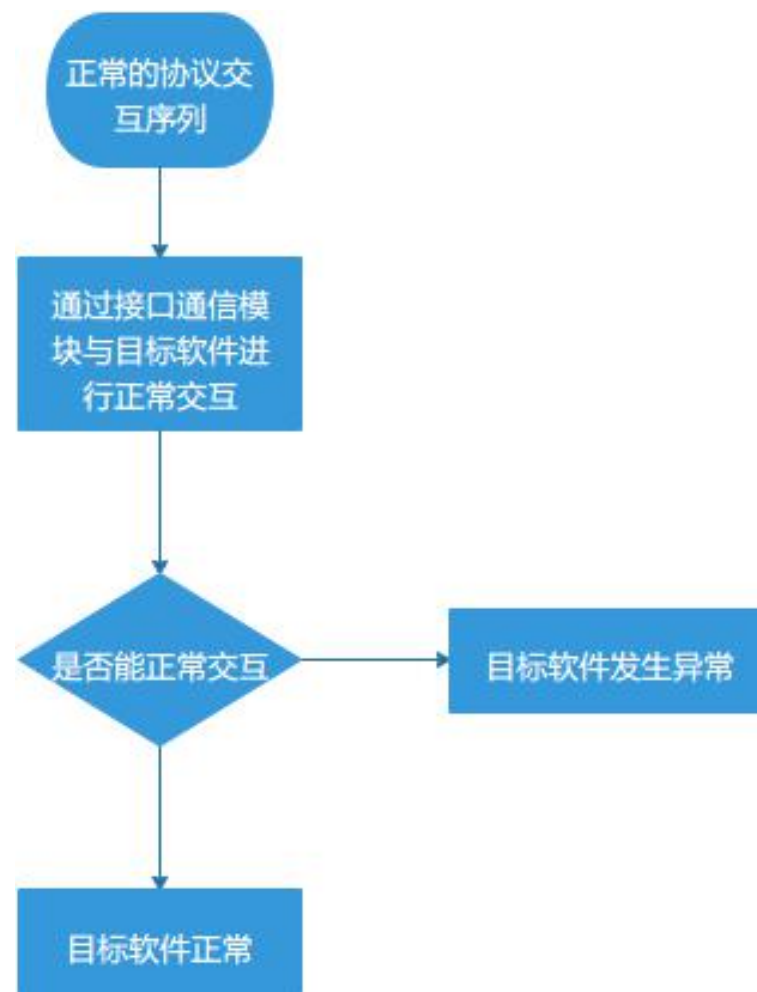
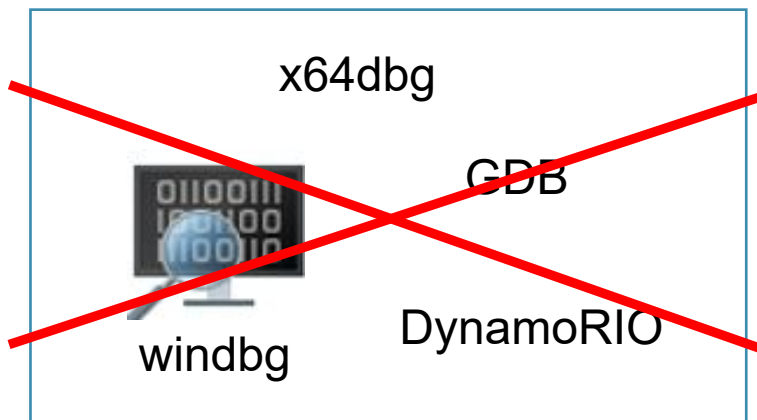


异常检测模块

对于常规平台的异常检测



对于工控设备的异常检测



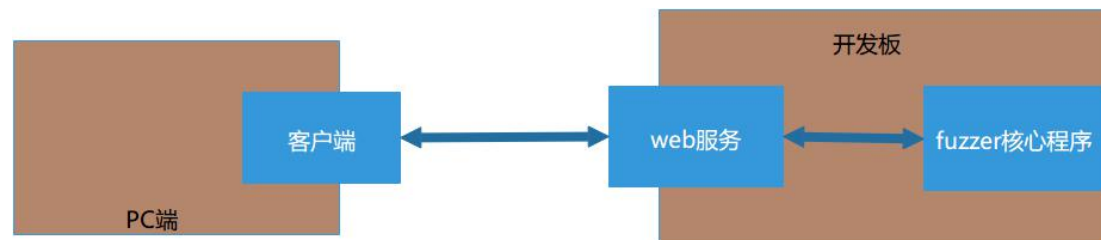
跨平台客户端的实现

采用的技术



设计

1. 首先将模糊测试工具组开发为一个命令行工具
2. 然后开发了一个Web服务与模糊测试工具交互
3. 客户端通过Web服务控制模糊测试工具的执行



创建模糊测试任务

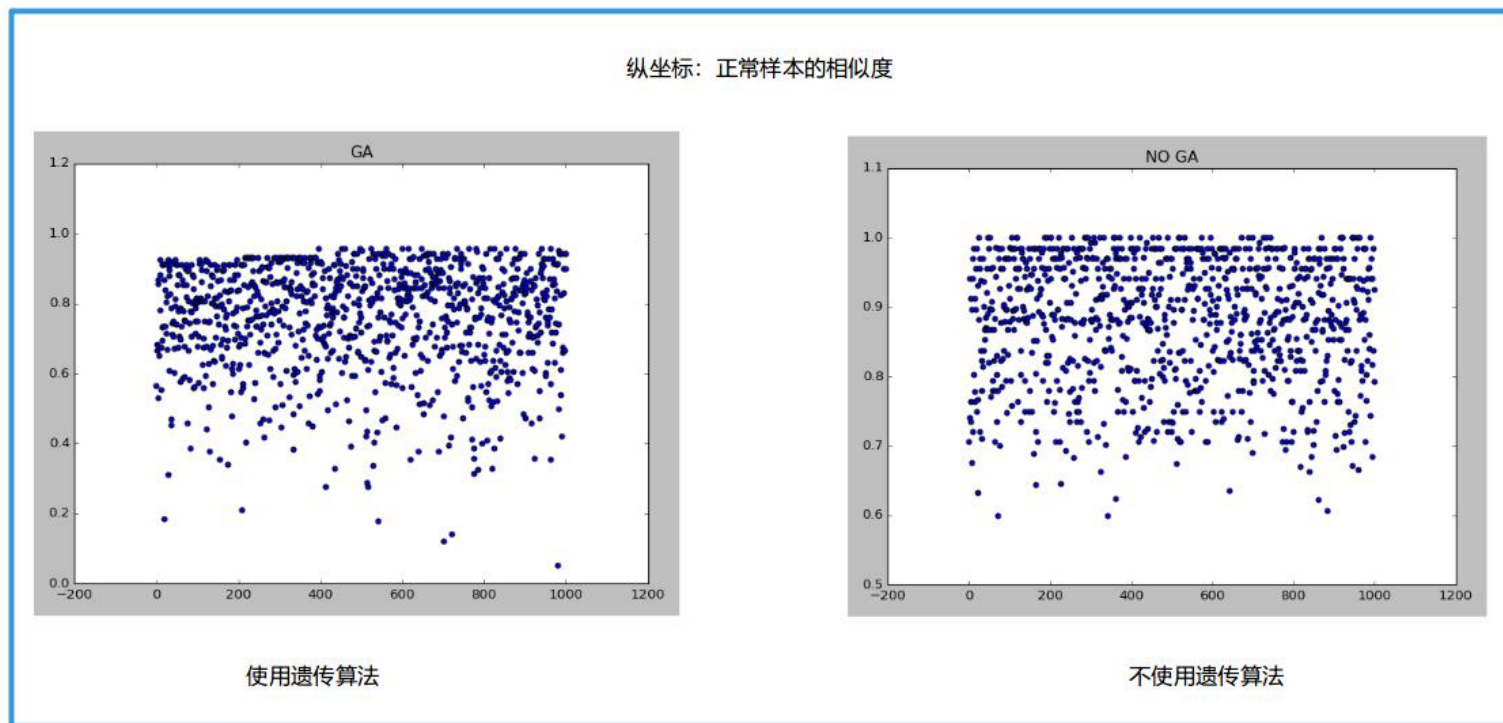
A thick, light blue curved line starts from the top left, curves around the left side of the slide, and then curves back towards the bottom right, creating a large, open shape that frames the central text.

03 实验分析

Experimental Analysis

实验分析

遗传算法对数据变异方案的优化



不采用遗传算法时所生成的变异数据中有很多分布在与原始样本相似度极高的区域（相似度接近1）并且样本的分布比较分散，而采用遗传算法优化后生成的变异数据的相似度主要分布在0.6~0.9之间，因此使用遗传算法对数据变异进行引导能够优化数据变异的效果。

实验分析



网口模糊测试验证

对 Float FTP 和 Schneider Electric Modbus Serial Driver 两款软件进行了测试，发现位于这两个软件中的历史漏洞。



串口模糊测试验证

为了测试实现了一个有漏洞的程序，程序采用串口通信，程序从串口获取数据后就会调用一个数据处理函数进行处理，在数据处理函数中存在漏洞，通过实验发现本文实现的模糊测试工具可以发现程序中的漏洞。



USB模糊测试验证

对一个闪迪的U盘进行模糊测试，发现了U盘的一个拒绝服务漏洞。


实验分析

测试结果如下所示：

模糊测试类型	能否挖掘出软件中的漏洞
网口模糊测试	能
串口模糊测试	能
USB模糊测试	能

通过实验可以得出：

- 遗传算法能够对数据变异策略起到优化的作用
- 本文实现的模糊测试工具能满足多接口模糊测试的需求，能够有效的挖掘出采用不同接口进行通信的软件中的潜在安全漏洞。

A thick, light blue curved line starts from the top left, curves around the left side of the slide, and then curves back towards the bottom right, creating a large, open shape that frames the central text.

04 总结与展望

Summary and Prospect

总结

主要研究内容

研究了通用的协议模糊测试方案

基于协议模糊测试的特点，提出了面向不同接口协议的通用模糊测试方案。

实现了模糊测试工具

基于 Orange Pi Prime 开发板实现了一个通用的协议模糊测试工具并为该工具开发了一个跨平台的 GUI 客户端。

验证实现的模糊测试工具

对实现的模糊测试工具进行测试，发现本论文设计并实现的模糊测试工具能有效地发现采用不同接口进行通信的软件中的潜在安全漏洞。

可能的改进

协议交互的自动化生成

01

02

设备出现异常后的自动重启

支持更多的通信接口

03

04

其他类型的漏洞检测



谢谢各位老师