一个基于 http 服务的环境因素检测系统(测试部分)

福建中学 2023-2024 届 5B 班 22 号 钟岱霖

1. 目錄

2. 測試計劃

2.1. 测试目的

前人曾经说过"不要相信用户的输入",我们应当视用户输入为安全性最低的数据,并对其进行深度验证后才输入系统内,再次以 stoi 函数为例,该函数仅能接受不包含字母符号的数字的 string 类型内容,否则将产生 std::invalid_argument 错误并在 error 未被 catch 的情况下强制关闭程序,因此,其中一种思路是对用户输入的数据进行一次验证之后再输入 stoi 函数,以此避免预期之外的错误

2.2. 測試設計

本项目由客户端,服务端组成,其中允许用户输入数据的地方仅有 config.txt 一项,因此,只需对读取 config.txt 时可能存在的问题进行测试即可

以下是 config.txt 的例子

```
serverPort = "1009",
SQLserverip = "192.168.1.103",
SQLserverPort = "3306",
username = "admin",
password = "admin",
table = "root"
```

可以注意到,config.txt 中, SQLserverip, username, password 和 table 是字符串,而 SQLserverPort 和 serverPort 是数字,输入需限制在这些类型中

2.2.1.1. 数据类型测试

上面提到,config.txt 的各项属性对数据类型是敏感的,因此,我们可以对数据类型进行测试

2.2.1.2. 极限数据测试

除去数据类型之外,我们还可以进行过大数据测试,我们知道,int 的最大值是 2^31-1,因此,在 serverPort 中输入 2^32.

2.2.1.3. 栈溢出测试

另一方面,我们都知道,栈空间的容量是存在上限的,当然,这一数字并非标准,而是取决于编译器,在 linux gcc 中,栈空间的容量是 8m,因此,我们建立一个非常大的文件,就有可能造成栈溢出错误

2.2.1.4. 空文件测试

空值是非常致命的因素,对此有必要进行测试

2.2.1.5. 端口号范围测试

端口的范围是 0 到 65535,因此 65536 或者-1 端口可能造成错误

2.2.1.6. 错误信息测试

输入错误的信息可能造成数据库连接失败,测试软件是否存在对此的有效反应是必要的

2.3. 測試方法

2.3.1. 数据类型测试

我们可以建立复数组数据,针对某可能存在的错误进行测试

2.3.1.1. 正常数据测试

```
serverPort = "1009",
SQLserverip = "192.168.1.103",
SQLserverPort = "3306",
username = "admin",
password = "admin",
table = "root"
```

2.3.1.2. 错误数据类型测试

```
serverPort = "a",
SQLserverip = "你好",
SQLserverPort = "c"",
username = "",
password = "////",
table = "0-"
```

2.3.1.3. 极限数据测试

```
serverPort = "2147483648",
SQLserverip = "192.168.1.103",
SQLserverPort = "3306",
username = "admin",
password = "admin",
table = "root"
```

2.3.1.4. 栈溢出测试

2.3.1.5. 空文件测试

2.3.1.6. 端口号范围测试

```
serverPort = "65536",
SQLserverip = "192.168.1.103",
SQLserverPort = "3306",
username = "admin",
password = "admin",
table = "root"
```

```
serverPort = "-1",
SQLserverip = "192.168.1.103",
SQLserverPort = "3306",
username = "admin",
password = "admin",
table = "root"
```

2.3.1.7. 错误信息测试

```
serverPort = "1009",
SQLserverip = "192.168.1.103",
```

```
SQLserverPort = "3306",
username = "admin",
password = "admin",
table = "root"
```

2.4. 測試結果

2.4.1.1. 正常数据

正常登入,连接

2.4.1.2. 错误数据类型测试

正常报错: config.txt not full fill

2.4.1.3. 过大数据测试

正常报错: serverPort out of range(0-2147483647)

2.4.1.4. 栈溢出测试

由于使用了 stl 容器,读取时使用堆空间进行存储,因此未引起栈溢出

2.4.1.5. 空文件测试

正常报错: config.txt is empty

2.4.1.6. 端口号范围测试

serverPort = "65536"

正常报错: database connect error

serverPort = "-1"

正常报错: serverPort out of range(0-2147483647)

2.4.1.7. 错误信息测试

正常报错: database connect error

3. 評估

3.1. 結果分析

从结果看来,效果相当不错,程序对预期内可能存在的错误都做好了应对,可以说,在可预期的范围内,该程序是没有恶劣 bug 的,可以说,它已经完全完成了系统最初的预期目标,即为用户提供高效的环境参数监测系统

4. 改進計劃

4.1. 改進設計

虽然该程序对文件读写做了充足的防御措施,但前人还说过"网络攻击往往来自你想不到的地方,而不是那个人人都知道去试着攻打的登录页面"本程序作为服务器软件,对于网络攻击方面并未提出一个足够的解决方案,当然,这是可以理解的,毕竟网络安全领域对于一个 sba 来说实在是过于复杂了,当然如果能做到自然是最好的了

4.2. 解決方案提議

Cpp-Httplib 提供了 https 协议组件,不过考虑到剃须刀理论因而在前面没有使用,如果要考虑网络安全因素,对网络协议进行加密显然是最方便快捷的一步,经过私钥加密后,http 服务可以使中间人劫持难以运作,从而提供高隐私性的通讯服务

与此同时,还可以在 html 中加入身份验证以进一步提高安全性,保证用户个人隐私不受未经授权的用户观看,有些人或许会认为自家的光照强度数据被别人看到是一件不太好的事情,因此,他们需要方方面面的身份验证