Dubbo 服务运行在 Docker 容器中，修复漏洞方法如下

结合你提供的截图信息，你的服务器 IP 是 `169.169.39.135`，且 Dubbo 服务运行在 Docker 容器中，端口映射为 `0.0.0.0:20817->20817`。

这意味着 **20817** 端口直接绑定在宿主机的所有网卡上（**0.0.0.0**）并暴露给了公网，这就是漏洞产生的原因。

以下是针对你当前环境（Docker + Linux）的具体修复步骤：

方法一：修改 Docker 端口绑定（最快修复，无需改代码）

既然 Dubbo 服务只需要内部调用，不需要从公网直接访问，我们可以修改 Docker 的端口映射，让 20817 端口只监听本地回环地址（127.0.0.1），或者直接不映射到宿主机。

操作步骤：

1. 找到你的 `docker-compose.yml` 文件。
2. 找到 `rurl-model-proxy-server` 这个服务的 `ports` 配置部分。
3. 将 `20817:20817` 修改为 `127.0.0.1:20817:20817`。

修改前：

```
ports:
  - "20817:20817"
  - "8041:8090"
```

修改后：

```
ports:
  - "127.0.0.1:20817:20817"  # 加上 127.0.0.1 限制
  - "8041:8090"
```

效果：修改后，外部网络（包括你的公网 IP `169.169.39.135`）将无法访问 20817 端口，只有服务器本机可以访问。

重启服务：

```
docker-compose down
docker-compose up -d
```

方法二：配置服务器防火墙（iptables）

如果你不想修改 Docker 配置，或者修改后仍希望特定内网 IP 能访问，可以使用防火墙策略。

目标：禁止公网访问 20817，但允许本机或特定内网访问。

执行命令（在宿主机上执行）：

1. 拒绝外部访问 **20817**：

   ```
   # 拒绝所有外部 IP 访问 TCP 20817 端口
   iptables -I INPUT -p tcp --dport 20817 ! -s 127.0.0.1 -j DROP
   ```

   （注意：如果你的应用服务器在其他机器上，比如 *192.168.x.x*，需要把 `! -s 127.0.0.1` 改为 `-s 192.168.x.x` 或者添加一条允许规则）
2. 保存规则（防止重启失效）：

   ```
   service iptables save
   # 或者根据你的系统
   iptables-save > /etc/iptables/rules.v4
   ```

方法三：Dubbo 内部配置（彻底禁用 Telnet）

这是最安全的做法，即使端口暴露了，黑客也无法通过 Telnet 执行命令。

操作步骤：

1. 进入容器内部或修改你的项目配置文件（`application.yml` 或 `dubbo.properties`）。
2. 禁用 **Telnet** 命令支持。

如果你使用的是 Spring Boot 配置（`application.yml`），在 `rurl-model-proxy-server` 的配置中添加：

```
dubbo:
  protocol:
    name: dubbo
    port: 20817
    # 关键点: 设置为 false 或空, 禁用 telnet
    telnet: false
```

或者在 `dubbo.properties` 中:

```
dubbo.protocol.telnet=false
```

修改后重启容器:

```
docker-compose restart rurl-model-proxy-server
```

验证修复结果

修复完成后, 请在你的本地电脑（非服务器内部）使用 telnet 测试:

```
telnet 169.169.39.135 20817
```

- 修复成功: 连接超时（Connection timed out）或连接被拒绝（Connection refused）。
- 修复失败: 屏幕变黑或出现 `Connected to...` 字样。

建议优先采用 “方法一”或“方法二”, 因为这是基础设施层面的防护, 最快生效。

Revision #1
Created 2026-04-24 02:08:59 UTC by Admin
Updated 2026-04-24 02:10:06 UTC by Admin