# RemoteAccess-Linux：ssh+vnc+xfce4

```
59 ⊟    RemoteAccess分工（学号尾号%5）：
70          //使用vmware虚拟机、Xmanager、ssh等工具和环境，演示
71          0 linux：ssh+vnc+xfce4
72 ⊟        1 X/xdmcp/DISPLAY
73              如果是gdm3：
74              https://netsarang.atlassian.net/wiki/spaces/ENSUP/pages/14123
    +Based
75              如果是lightdm，创建/etc/lightdm.conf，内含两行：
76              [XDMCPServer]
77              enabled=true
78          2 rdp wrapper https://github.com/stascorp/rdpwrap
79          3 TeamViewer // 个人免费
80          4 windows RDP // 家庭版不行(服务端)，需要专业版，可以用vhdx安装
81
```

## 一、Linux：ssh+vnc+xfce4架构和原理

### 1. 理解各个组件的功能

#### a. SSH (Secure Shell)

- **功能**：SSH是一种加密的网络协议，用于安全地远程登录和管理Linux服务器。它通过加密通信防止中间人攻击，支持命令行操作、文件传输（SCP/SFTP）和端口转发。
- **原理**：基于客户端-服务器模型，使用非对称加密（如RSA）进行身份验证，对称加密（如AES）保护数据传输。

#### b. VNC (Virtual Network Computing)

- **功能**：VNC是一种图形化桌面共享协议，允许用户远程查看并控制另一台计算机的桌面环境。与SSH不同，VNC传输的是图形界面（像素数据）。
- **原理**：服务端（如 `tigervnc-server`）捕获屏幕图像并压缩传输到客户端（如 `Remmina`），客户端发送鼠标/键盘事件到服务端。默认使用RFB协议，通常不加密（需结合SSH隧道提升安全性）。

#### c. Xfce4

- **功能**：Xfce是一个轻量级Linux桌面环境，适合远程桌面场景。相比GNOME/KDE，它资源占用低，响应快。
- **原理**：基于X Window System（或Wayland），提供窗口管理器、面板、文件管理器等组件。在VNC中，Xfce作为服务端的桌面环境运行。

---

### 2. 组件间的协作关系

#### 架构流程

1. **用户层**：
   - 本地机器通过SSH连接到远程Linux服务器。
   - 通过SSH隧道安全转发VNC端口（如 `ssh -L 5901:localhost:5901 user@server`）。
2. **服务端层**：
   - 远程服务器启动VNC服务（如 `vncserver :1 -geometry 1920x1080 -depth 24`），指定使用Xfce4作为桌面环境（通过 `~/.vnc/xstartup` 配置）。
   - VNC服务监听端口（通常5900+显示编号），通过SSH隧道加密传输图形数据。
3. **协议层**：

> - SSH保障通信安全，VNC传输图形界面，Xfce4提供具体的桌面交互。

---

## 3. 整体功能与原理

### 实现的功能

- **安全的远程图形化访问**：用户通过SSH加密通道连接到远程服务器，再通过VNC在本地显示Xfce4桌面，实现完整的图形化操作（如运行GUI程序、文件管理等）。
- **轻量化解决方案**：Xfce4的低资源消耗使其适合服务器环境，尤其对带宽有限的远程连接更友好。

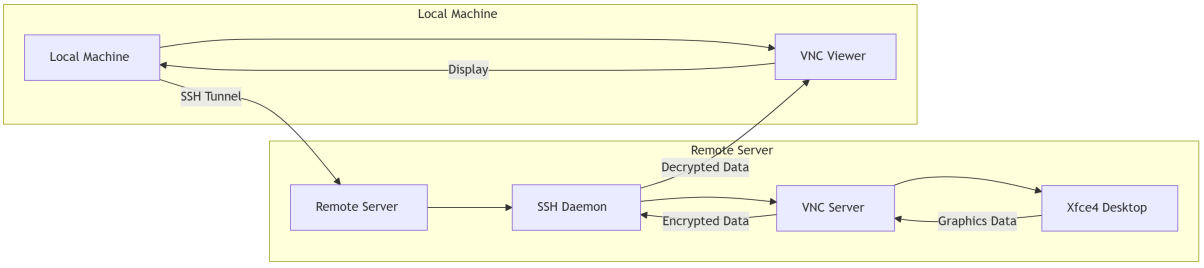
### 工作原理

- **加密通道**：SSH建立安全隧道，防止VNC的明文数据被窃听。
- **图形渲染**：Xfce4在服务端渲染桌面，VNC服务将渲染结果压缩后传输。
- **事件传递**：客户端的输入（鼠标/键盘）通过VNC协议传回服务端，Xfce4处理这些事件并更新界面。

---

- **SSH+VNC+Xfce4**构成了一套**安全的远程图形化桌面系统**：
  - **SSH**提供加密通信。
  - **VNC**传输图形界面。
  - **Xfce4**提供轻量级桌面环境。
- **适用场景**：远程服务器管理、低带宽环境、需要GUI操作的开发/测试。

通过这种组合，用户可以在任何地方高效、安全地访问完整的Linux桌面体验。




# 二、演示Linux：ssh+vnc+xfce4

---

## 1. 实验准备（Ubuntu 22.04 桌面版）

### 安装必要工具（VNC,SSH,WHIRESHARK，xfce4）

```
sudo apt update
sudo apt install -y tigervnc-standalone-server tigervnc-common# 安装 VNC

sudo apt install -y tigervnc-standalone-server wireshark# 安装 wireshark

sudo apt install -y openssh-server  # 安装 SSH 服务
sudo systemctl start ssh           # 启动服务
sudo systemctl enable ssh          # 设置开机自启

# 安装 Xfce4 桌面（轻量级，适合远程）
sudo apt install -y xfce4 xfce4-goodies

# 创建自定义 xstartup 配置
mkdir -p ~/.vnc
```

```
cat > ~/.vnc/xstartup <<EOF
#!/bin/bash
unset SESSION_MANAGER
exec startxfce4
EOF
chmod +x ~/.vnc/xstartup

# 启动 VNC（强制使用自定义配置）
vncserver :2 -geometry 1280x720 -xstartup ~/.vnc/xstartup
```

(Wireshark 用来抓包分析流量)

---

## 2. 测试方案

### 🔒 方案A：SSH加密隧道 + VNC（安全）

1. **启动VNC服务端**（默认监听本地`5901`）

```
vncserver :1 -localhost  # 限制只允许本地访问 -localhost可去掉
```

2. **从Windows建立SSH隧道**

```
ssh -L 5901:localhost:5901 ubuntu用户名@虚拟机IP
#这里我的代码是
ssh -L 5901:localhost:5901 yang@192.168.42.128
```

首先ping一下看网络是否通畅：

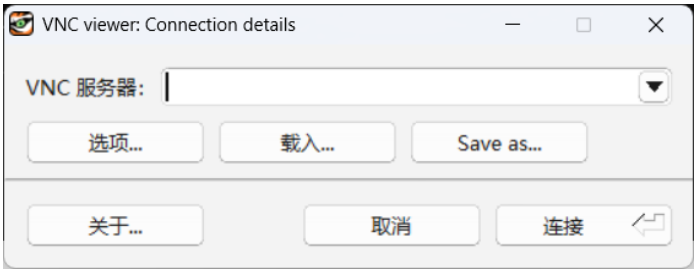
```
PS C:\Users\26283\Desktop> ping 192.168.42.128

正在 Ping 192.168.42.128 具有 32 字节的数据：
来自 192.168.42.128 的回复：字节=32 时间<1ms TTL=64
来自 192.168.42.128 的回复：字节=32 时间<1ms TTL=64
来自 192.168.42.128 的回复：字节=32 时间<1ms TTL=64
来自 192.168.42.128 的回复：字节=32 时间<1ms TTL=64

192.168.42.128 的 Ping 统计信息：
    数据包：已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位)：
    最短 = 0ms，最长 = 0ms，平均 = 0ms
```

尝试进行ssh连接：

```
PS C:\Users\26283\Desktop> ssh -L 5901:localhost:5901 yang@192.168.42.128
The authenticity of host '192.168.42.128 (192.168.42.128)' can't be established.
ED25519 key fingerprint is SHA256:kEYpXuxDc/G5z5J5/1+I8XaExbAHGZdoQ0SNa7YfpHY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.42.128' (ED25519) to the list of known hosts.
yang@192.168.42.128's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

扩展安全维护（ESM）Applications 未启用。

212 更新可以立即应用。
这些更新中有 204 个是标准安全更新。
要查看这些附加更新，请运行：apt list --upgradable

2 个额外的安全更新可以通过 ESM Apps 来获取安装。
可通过以下途径了解如何启用 ESM Apps：at https://ubuntu.com/esm


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

yang@yang-virtual-machine:~$
```

3. **VNC客户端连接**

- **下载 VNC 客户端**（如 TigerVNC Viewer 或 RealVNC）。
- **打开 VNC 客户端**




- 地址填 `localhost:5901`
- 流量全程通过SSH加密
- 在连接时发现无法连接：

  yang@yang-virtual-machine:~$ channel 3: open failed: connect failed: Connection refused

- 排查原因发现：从我的 `ps aux` 输出来看，**显示编号 `:1` 被 Ubuntu 默认的 GNOME 桌面环境占用了**（如 `gvfsd` 和 `gnome-shell` 相关进程）。这是导致 VNC 无法使用 `:1` 的根本原因。

```
root@yang-virtual-machine:/home/yang/桌面# ps aux | grep ':1' | grep -v grep
root           1  0.4  0.3 168204 13400 ?        Ss   10:24   0:12 /sbin/init au
to noprompt splash
```

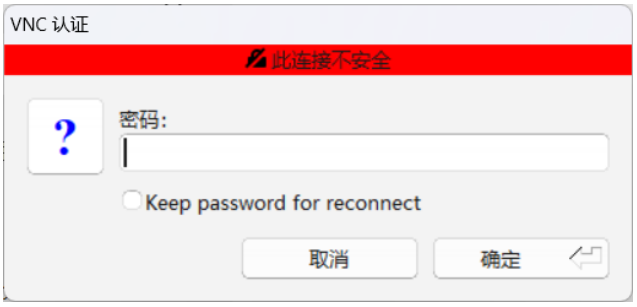
- **更换 VNC 显示编号，和ssh连接等一系列的1改为3**

```
# 直接使用 :3 或其他编号（避开系统占用的 :1）
vncserver :3 -geometry 1280x720

# 在 Windows 上同步修改 SSH 端口转发
ssh -L 5902:localhost:5902 yang@192.168.42.128
```

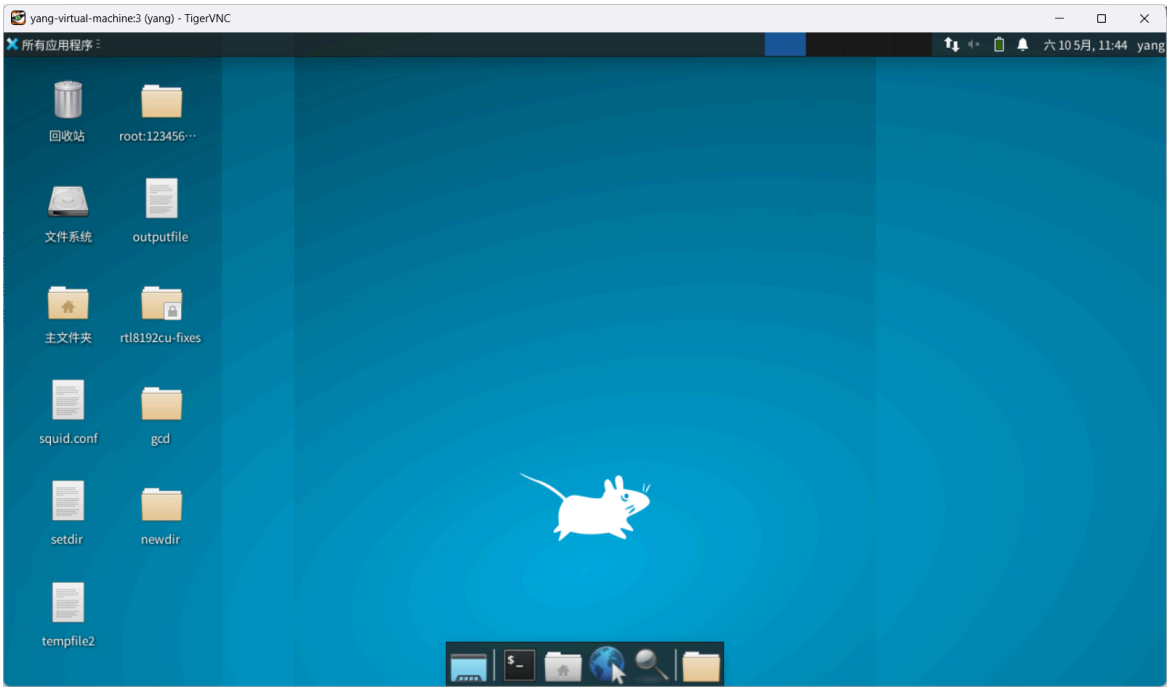
VNC成功启动

```
yang@yang-virtual-machine:~$ vncserver :3 -geometry 1280x720 -localhost yes -Sec
urityTypes VncAuth

New Xtigervnc server 'yang-virtual-machine:3 (yang)' on port 5903 for display :3
.
Use xtigervncviewer -SecurityTypes VncAuth -passwd /home/yang/.vnc/passwd :3 to
connect to the VNC server.
```

- 然后在 VNC 客户端连接 `localhost:5903`。
- 此时VNC客户端连接就会跳转到输入密码阶段




- 成功连接到有小老鼠的桌面!



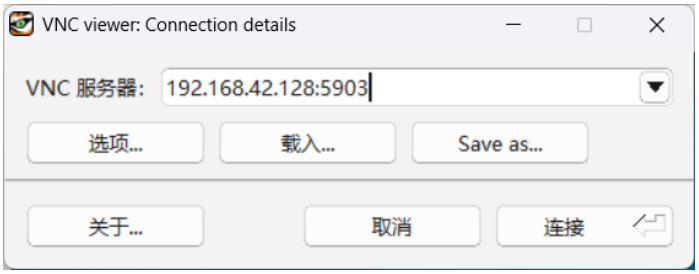

## ⚠️ 方案B：裸奔VNC

1. **启动VNC服务端（暴露端口）**

```
vncserver -kill :3
vncserver :3 -geometry 1280x720 -localhost no -SecurityTypes VncAuth
```

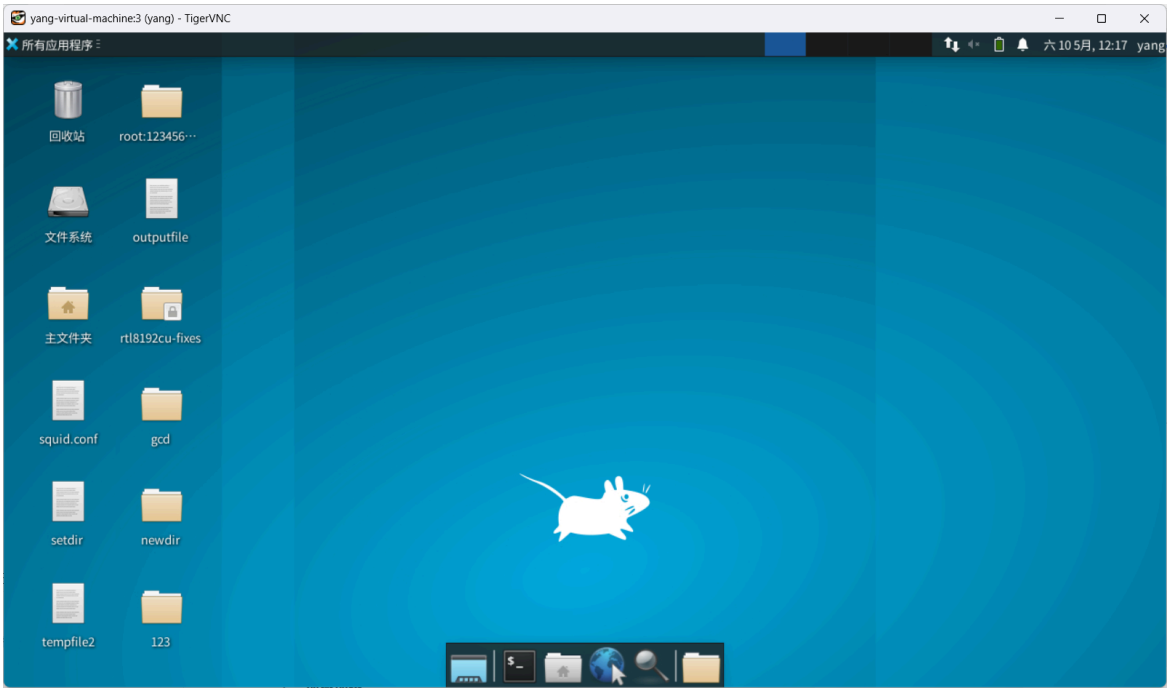
2. **直接连接VNC**

- 地址填 `虚拟机IP:5903`




- 流量明文传输

  也能够连接




---

tip：不能使用root端，必须使用用户端，要不然会导致权限问题

## 3. 抓包对比

### 用Wireshark验证加密效果

1. **在Ubuntu虚拟机抓包**

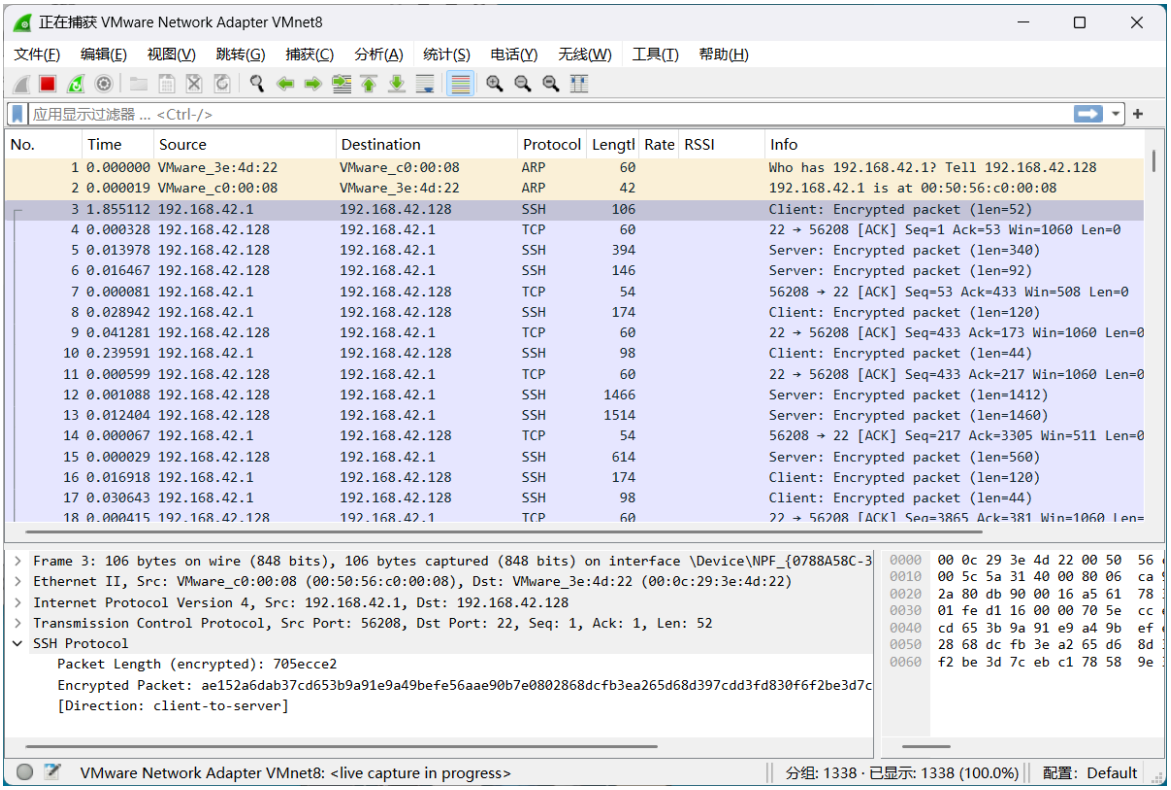
```
sudo wireshark  # 选择网卡，过滤`port 5901`
```

2. **观察现象**

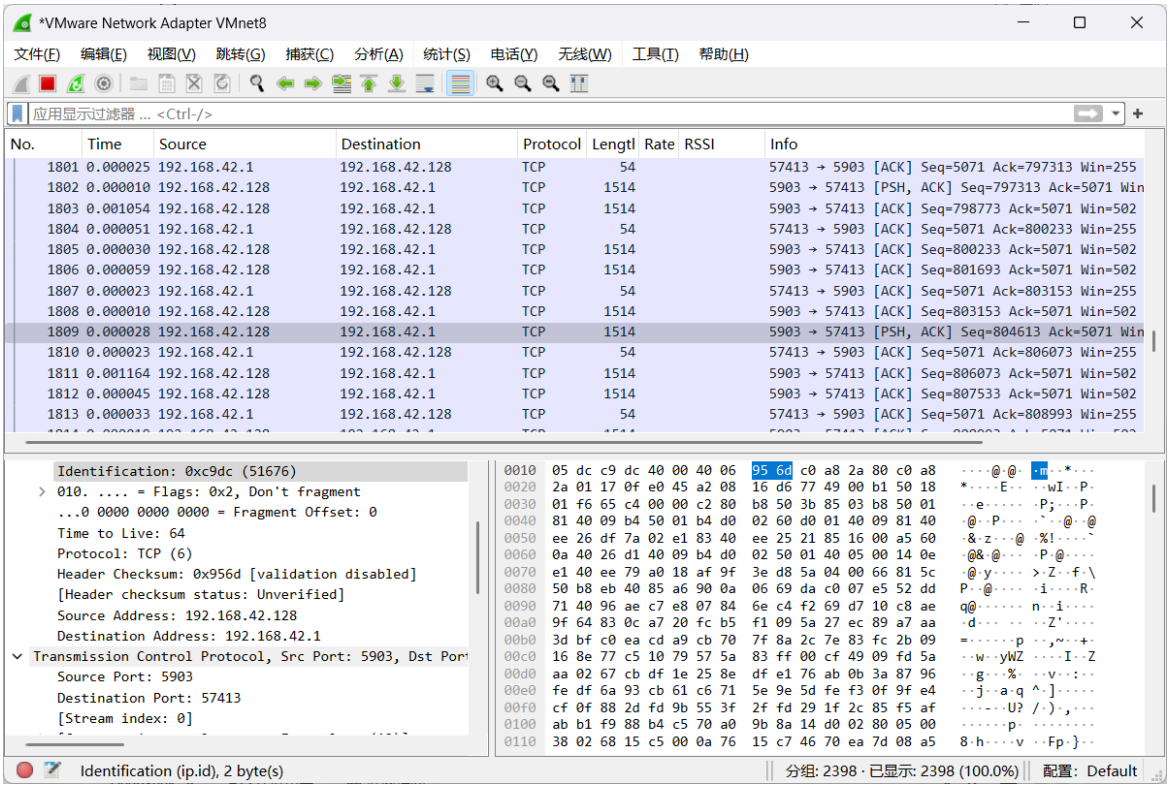
- **SSH+VNC**：只能看到加密的SSH流量（协议显示`TLSv1.3`）

  捕获的流量可以看到都是SSH加密的，而且我不选中vnc桌面没有流量，我的鼠标在远程连接的每动一下就产生大量的流量，说明

  - VNC 协议会实时传输桌面变化：
    - 鼠标移动/点击 → 触发坐标更新（每秒 10-30 个包）
    - 窗口拖动/内容变化 → 触发像素块传输（流量激增）
    - 静止时 → 仅保持心跳包（约 1 个包/秒）




- **裸奔VNC**：直接看到RFB协议明文（如密码、像素数据）




## 4. 快速命令总结

| 场景 | 服务端命令 | 客户端连接方式 |
| --- | --- | --- |
| SSH加密隧道 | `vncserver :1 -localhost` | `ssh -L 5901...` + VNC |

| 场景 | 服务端命令 | 客户端连接方式 |
| --- | --- | --- |
| 裸奔VNC（测试用） | `vncserver :3 -geometry 1280x720 -localhost no -SecurityTypes VncAuth` | 直连 `IP:5901` |

---

## 5. 安全建议

- **实验结束务必关闭裸奔VNC**：

```
vncserver -kill :3
sudo ufw deny 5901/tcp  # 防火墙阻止端口
```

- 长期使用**必须走SSH隧道**，避免密码和操作被嗅探！