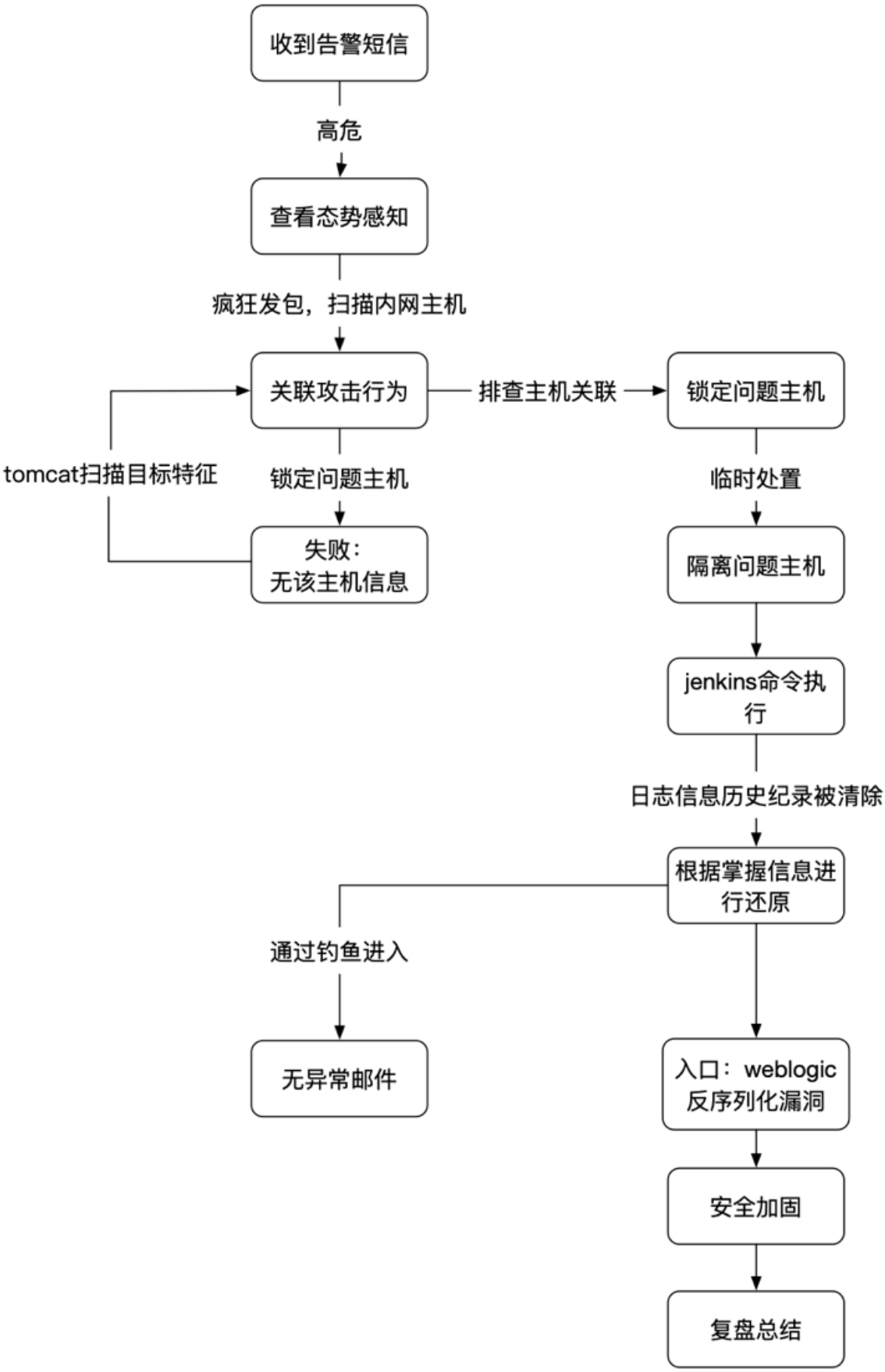
蓝队应急响应之“雄鸡夜鸣”

原创 队员编号044 酒仙桥六号部队 5天前

这是 **酒仙桥六号部队** 的第 **44** 篇文章。

全文共计3233个字，预计阅读时长12分钟。


收到告警短信
高危
查看态势感知
疯狂发包，扫描内网主机
关联攻击行为
排查主机关联
锁定问题主机
tomcat扫描目标特征
锁定问题主机
临时处置
失败：
无该主机信息
隔离问题主机
jenkins命令执行
日志信息历史纪录被清除
根据掌握信息进行还原
通过钓鱼进入
无异常邮件
入口：weblogic反序列化漏洞
安全加固
复盘总结

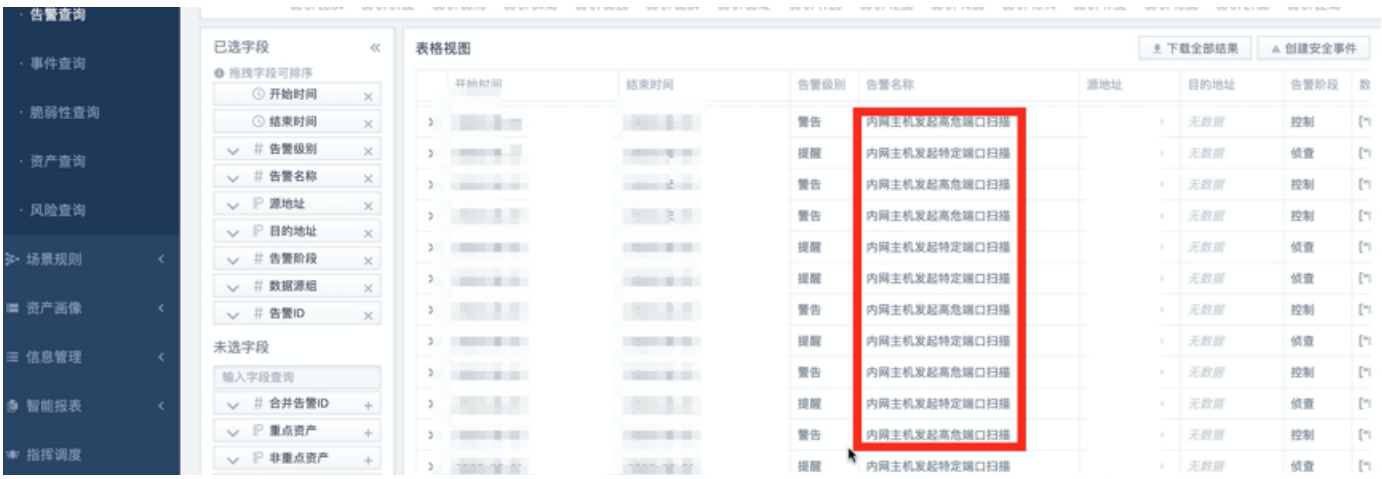
背景介绍

简单自我介绍一下，我是一个安全工程师，也是一个人的“安全部”。一个周五的晚上，我正在和小伙伴们在游戏中厮杀，还沉浸在躺赢快乐中的我突然接到紧急短信。

因为我重置过短信告警条件，只有行为出现漏洞扫描、入侵警告的达到触发值，才允许发送短信。所以我们接到短信后要立即回公司处理。

评估

登录态势感知平台，一台服务器在疯狂发包，对内网主机发起扫描。直觉告诉我，这种扫描行为与正常业务请求无关，因为这个时间没有业务提交过排期，所以很可能该`IP`已失陷。

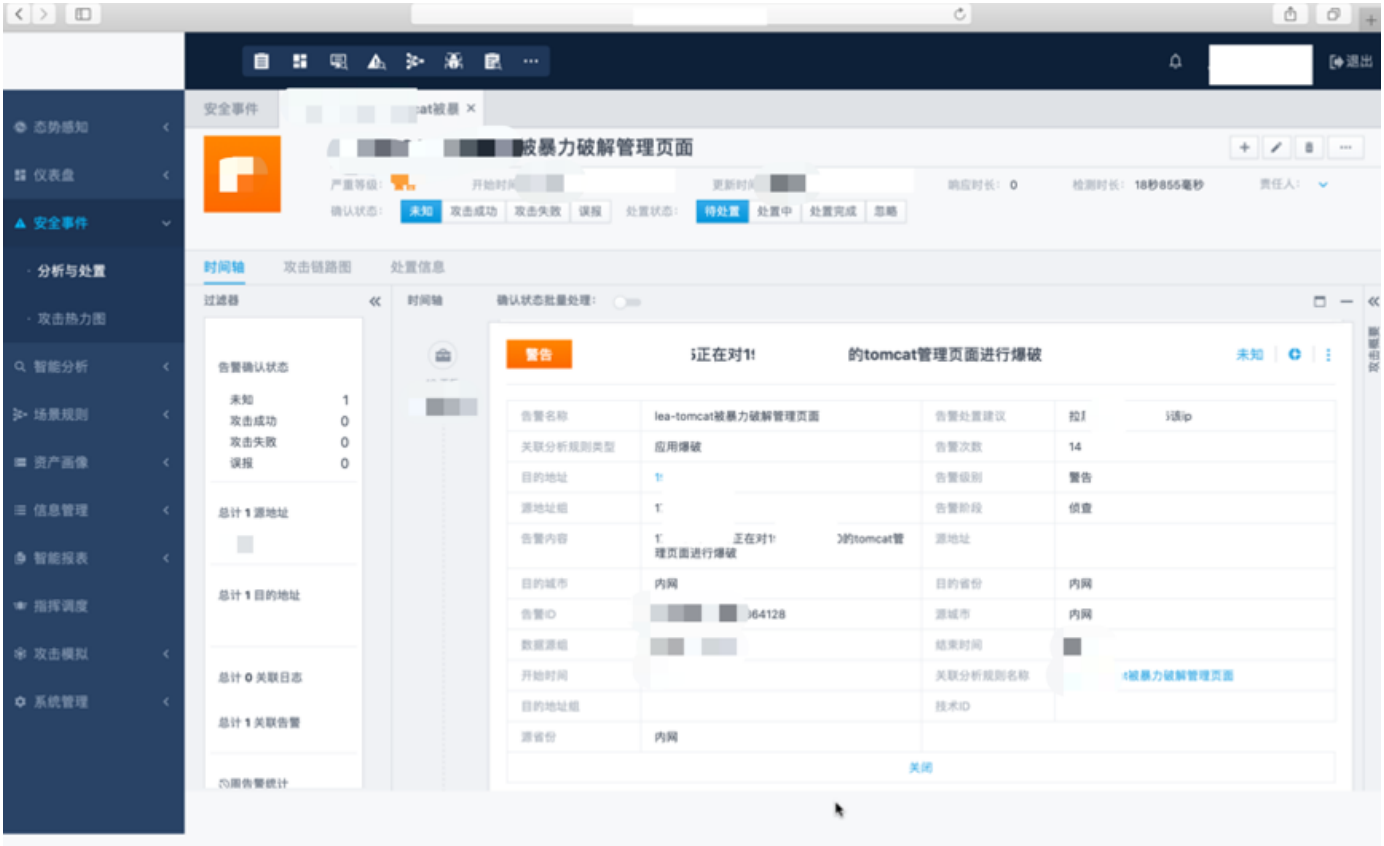


时间已入深夜，我匆忙的叫上运维同学来到公司定位失陷`IP`所关联的主机。可是奇怪的是这个`ip`经过第一眼的推测应该属于`dmz`区域，因为这个区域的地址段是`*.22`，为了更精准又去找的小伙伴一起确认。经过最终确认运维同学的记录中却并没有这个`IP`的记录。

（#*-*）！这给我们排查添了大麻烦，只能让运维同学找业务同学一起排查这个地址的归属。此时的我们是又累又困的。

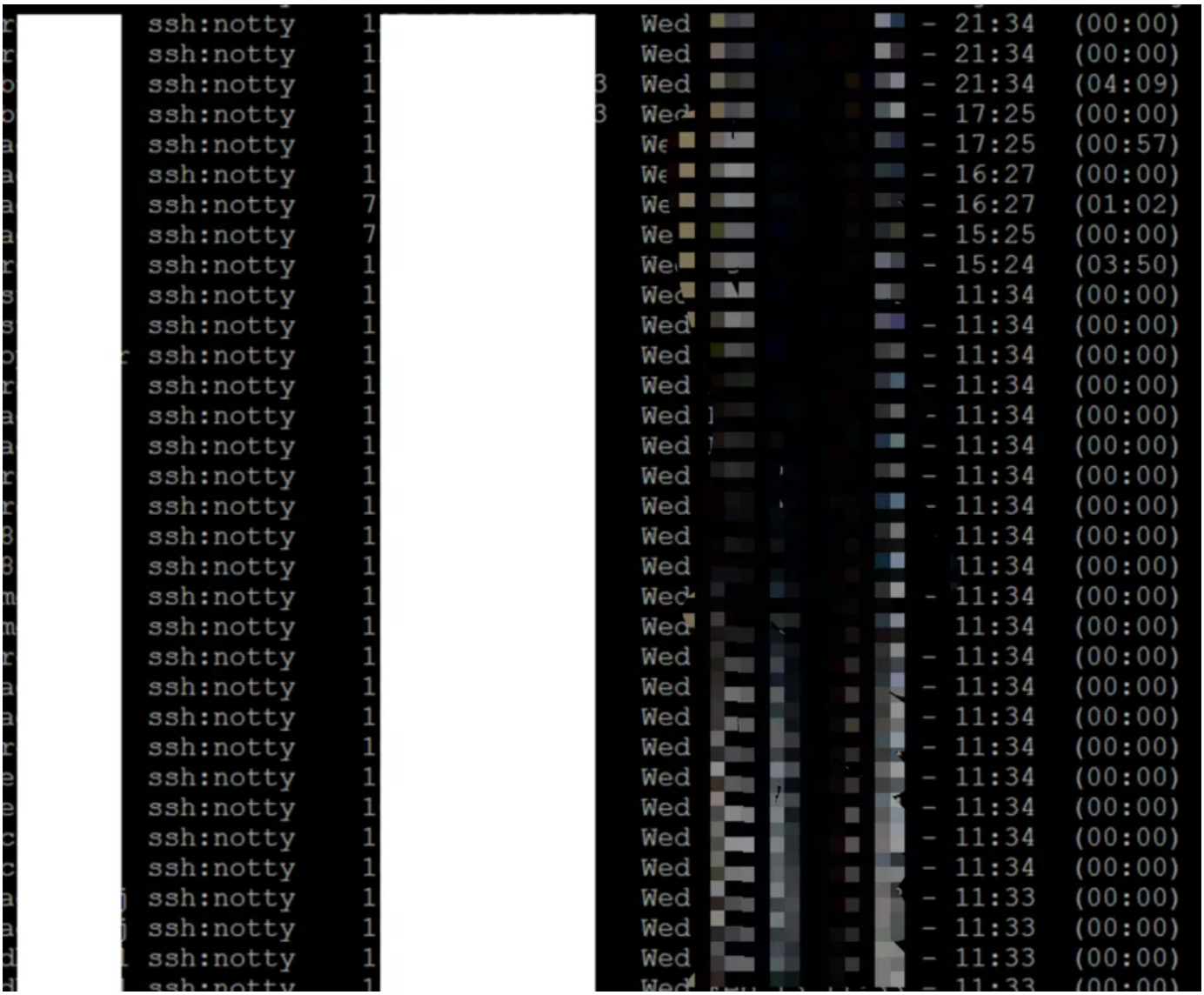
排查陷入了僵局，因为从资产管理档案中，无法寻找到该主机的相关人员的登记历史记录。所以只能回到态势感知系统上翻一翻态势感知的信息。

通过关联攻击IP行为，发现该IP地址存在对线上的一台tomcat暴力破解。做信息安全的我第一时间猜想，是不是和这台服务器上的业务有所关联？



叫上运维同学和业务同学来到机房，由运维同事进入`tomcat`服务器系统检查。首先检查一下`tomcat`日志，此文件默认存放`[tomcat home]\logs`目录中。

从上述记录可以看到，有用户在试图猜测`tomcat`口令，时间分别在某年某月某日凌晨`01：27`，`IP`地址来源分别是`*.*.*.1`。检查端口`21`，`22`的信息也存在猜测口令记录。



经过初步判断，内网机器疑似遭受黑客入侵。攻击者入侵后扫描了内网弱口令进行横向移动的尝试，从图中的暴力破解可以感知到。

这个ip地址不属于服务器范围。我询问同在旁边的业务同学是否能够确定是哪个在使用这一个IP，业务同学不能确定。此时有点慌~因为出现了类似的“网络灵异”事件。

几经波折，终于在运维和业务同学的配合下定位到这台机器。原来是一个外包项目组用的jenkins服务器。

Oops!

A problem occurred while processing the request. Please check our bug tracker to see if a similar problem has already been reported. If it is already reported, please vote and put a comment on it to let us gauge the impact of the problem. If you think this is a new issue, please file a new issue. When you file an issue, make sure to add the entire stack trace, along with the version of Jenkins and relevant plugins. The users list might be also useful in understanding what has happened.

Stack trace

```
java.lang.UnsupportedClassVersionError: Orange has been compiled by a more recent version of the Java Runtime (class
file version 55.0), this version of the Java Runtime only recognizes class file versions up to 52.0
        at java.lang.ClassLoader.defineClass1(Native Method)
```

随后辗转战场，我们在机房里找到了这台`Jenkins`主机。这台`jenkins`是由外包项目组实施项目的时候临时搭建，并未迁移到服务器网段，所以没有记录在我们的资产表中。由于外包人员不规范的操作，让我们错过了最宝贵的一段时间。

控制

当即通知业务将这台设备进行隔离，随后我立即把遇到的情况和处理结果通过邮件上报给领导。

消除

可是这台机器究竟出什么问题了呢？为了弄清真相，我们决定对这台系统进行取证分析。

检查进程

先用 `top` 命令查看系统进程列表，显示如下：

```
top - 00:57:30 up 14:59,  0 users,  load average: 0.02, 0.04, 0.00
Tasks:   3 total,   1 running,   2 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.5 sy,  0.0 ni, 99.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  2038904 total,   178292 free,   489832 used,  1370780 buff/cache
KiB Swap:  1048572 total,  1048572 free,        0 used.  1396456 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0 4081068 165308  20356 S   0.7  8.1   0:12.11 java
   59 root      20   0   19960   3652   3084 S   0.0  0.2   0:00.05 bash
  354 root      20   0   42824   3516   3008 R   0.0  0.2   0:00.00 top
```

只有3个系统进程正在运行，这很不正常。

通过 `rpm -Va` 命令列出目前系统上面所有可能被更改过的文件，如 `top`、`netstat`、`ps` 等命令已经被替换。

因为此前有过类似文件被篡改的经历，不想历史重演，所以在上线前运维同事就已经在`home`目录下加密保存了一些重要的系统文件的副本。

开放端口分析

`netstat -anp`

```
root@9e93782b9ac1:/# netstat -anp
Active Internet connections (servers and established)
Proto    Recv-Q     Send-Q    Local   Address  Foreign  Address   State  PID/Program name
tcp 0 0 0.0.0.0:50000        0.0.0.0:* LISTEN -
tcp 0 0 0.0.0.0:8080          0.0.0.0:* LISTEN 7231/java
```

只有`8080`，`50000`端口对外开放，网络连接层面正常。

账号分析

仅有一个用户具有root权限，未被注入超级权限。

`awk -F: '$3==0{print $1}' /etc/passwd`

```
root@9e93782b9ac1:/# awk -F: '$3==0{print $1}' /etc/passwd
root
```

可登录用户

只有两个用户可以使用SSH方式进行登录：

`root`、`jenkins` `cat /etc/passwd |grep -E “/bin/bash&”`

```
root@9e93782b9ac1:/$ cat /etc/passwd |grep -E '/bin/bash'
root:x:0:0:root:/root:/bin/bash
jenkins:x:1000:1000::/var/jenkins_home:/bin/bash
```

可以看出账号层面正常。

定时任务

通过分析系统未创建定时任务。

```
root@9e93782b9ac1:/$crontab -u root -l
no crontab for root
root@9e93782b9ac1:/$crontab -u jenkins -l
no crontab for jenkins
```

hosts 文件分析

`more /etc/hosts`

```
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

`hosts`文件正常。

启动项分析

启动项记录系统自启动的情况，若攻击者入侵植入木马或者后门，为了维持服务器控制，会在启动项中发现他的蛛丝马迹。

`systemctl list-unit-files |grep enabled`

启动项无异常。

历史命令分析：

History

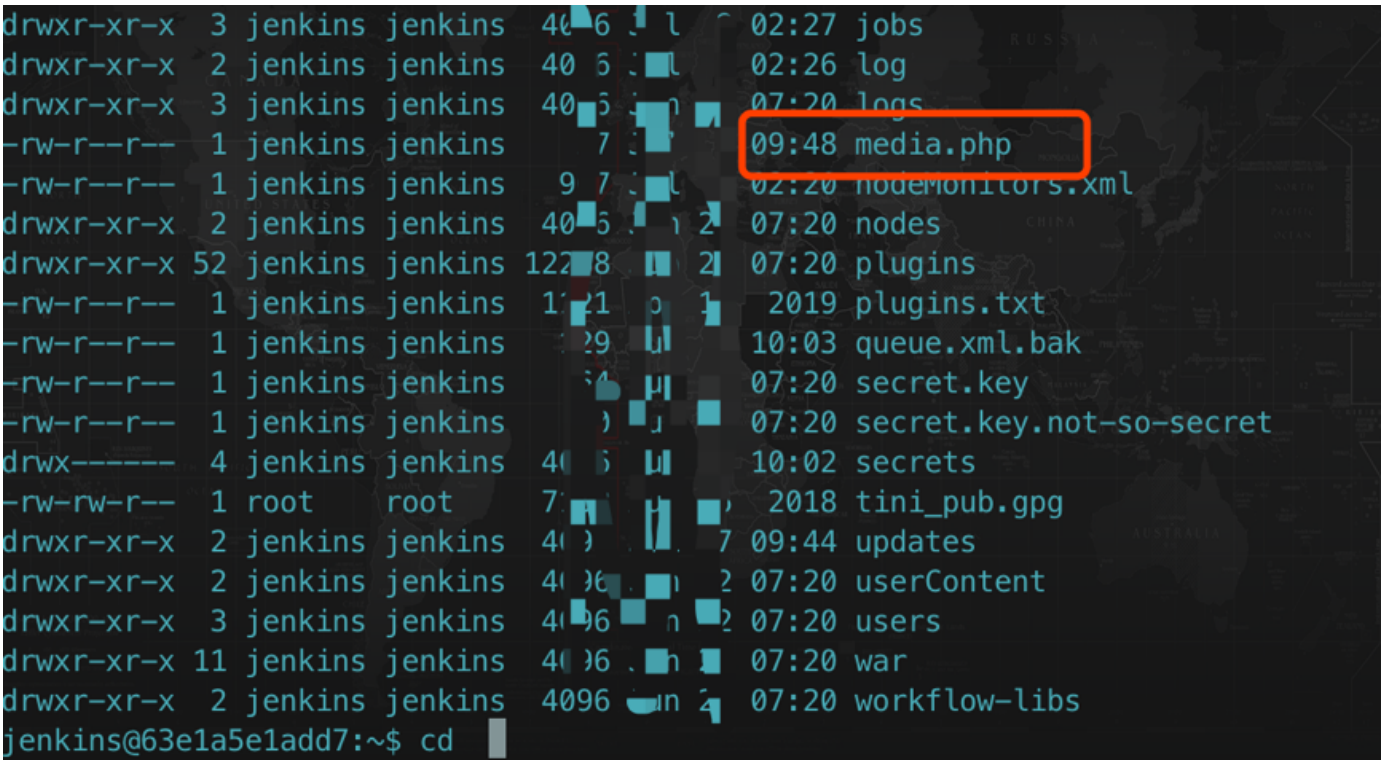
```
jenkins@63e1a5e1add7:/$ history 5
jenkins@63e1a5e1add7:/$ history 5
```

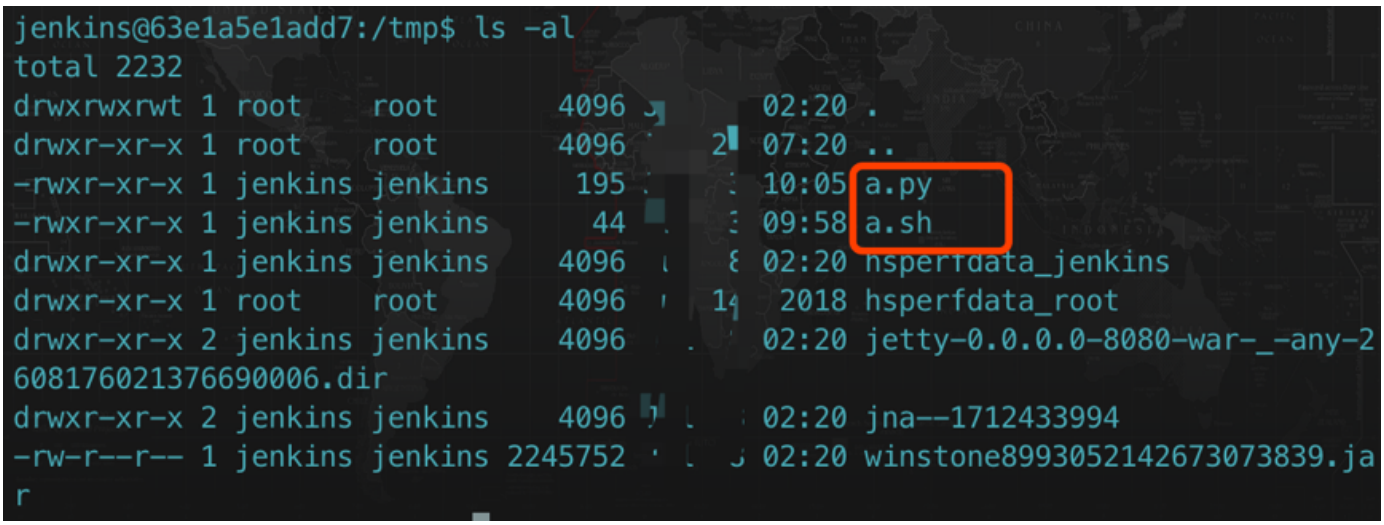
历史命令被清空，`top`等程序被替换攻击者对痕迹进行了清理，对系统排查似乎理不出什么头绪。我决定换一个思路，从`Jenkins`上尝试突破。从`Jenkins`日志排查。

Jenkins日志

```
AM INFO org.eclipse.jetty.server.handler.ContextHandler doStart
Started w.@5efa40fe{Jenkins v2.138,/,file:///var/jenkins_home/war/,AVAILABLE}
{/var/jenkins_home/war}
AM INFO org.eclipse.jetty.server.AbstractConnector doStart
Started ServerConnector@730d2164{HTTP/1.1,[http/1.1]}{0.0.0.0:8080}
46 AM INFO org.eclipse.jetty.server.Server doStart
Started @1868ms
6 AM INFO winstone.Logger logInternal
Winstone Servlet Engine v4.0 running: controlPort=disabled
AM INFO jenkins.InitReactorRunner$1 onAttained
Started initialization
AM INFO jenkins.InitReactorRunner$1 onAttained
Listed all plugins
AM INFO jenkins.InitReactorRunner$1 onAttained
Prepared all plugins
AM INFO jenkins.InitReactorRunner$1 onAttained
Started all plugins
INFO jenkins.InitReactorRunner$1 onAttained
Augmented all extensions
INFO jenkins.InitReactorRunner$1 onAttained
Loaded all jobs
INFO jenkins.util.groovy.GroovyHookScript execute
Executing /var/jenkins_home/init.groovy.d/init.groovy
INFO hudson.model.AsyncPeriodicWork$1 run
Started Download metadata
AM INFO hudson.model.AsyncPeriodicWork$1 run
Finished Download metadata. 39 ms
INFO org.springframework.context.support.AbstractApplicationContext prepareRefresh
Refreshing org.springframework.web.context.support.StaticWebApplicationContext@6b604d2b: display
name [Root WebApplicationContext]; startup date [Wed Jul 08 02:20:53 UTC 2020]; root of context
hierarchy
INFO org.springframework.context.support.AbstractApplicationContext obtainFreshBeanFactory
Bean factory for application context
[org.springframework.web.context.support.StaticWebApplicationContext@6b604d2b]:
org.springframework.beans.factory.support.DefaultListableBeanFactory@6ae9d1fc
```

异常文件





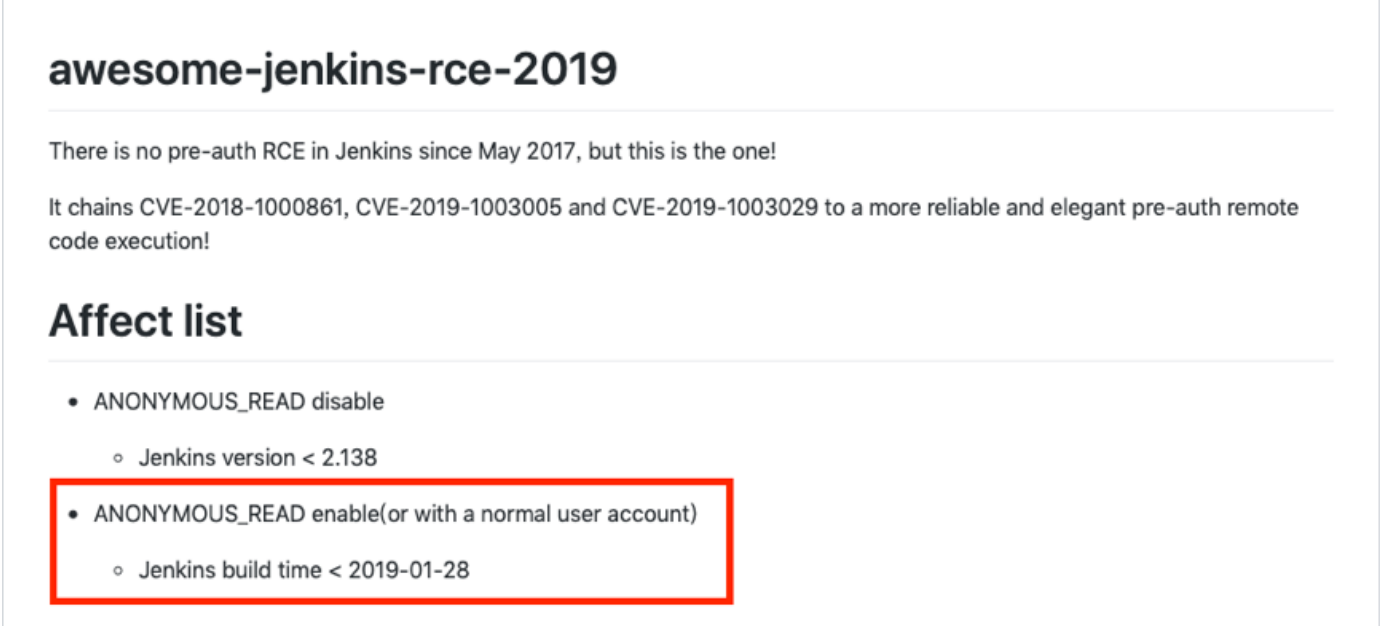
日志信息正常，但是我们找到了一些有趣的文件。通过分析这些文件，发现这是攻击者的使用的横向移动脚本，通过这些脚本进行横向移动。从这里可以确定这台`jenkins`已经失陷。

Jenkins简介

`JoudBees Jenkins（Hudson Labs）`是美国`CloudBees`公司的一套基于`Java`开发的持续集成工具。该产品主要用于监控持续的软件版本发布/测试项目和一些定时执行的任务。

```
jenkins@b7d14e9c5bd7:/$ cd var/jenkins_home/
jenkins@b7d14e9c5bd7:~$ ls
config.xml                                jobs              secret.key.not-so-secret
copy_reference_file.log                   logs              secrets
hudson.model.UpdateCenter.xml             nodeMonitors.xml  tini_pub.gpg
hudson.plugins.git.GitTool.xml            nodes             updates
identity.key.enc                          plugins           userContent
init.groovy.d                             plugins.txt       users
jenkins.install.InstallUtil.lastExecVersion  queue.xml.bak  war
jenkins.install.UpgradeWizard.state       secret.key        workflow-libs
jenkins@b7d14e9c5bd7:~$ cat config.xml|grep version
<?xml version='1.1' encoding='UTF-8'?>
  <version>2.138</version>
```

通过分析，系统使用`Jenkins`，查看`config.xml`配置文件分析，发现其`Jenkins`版本为`2.138`。



awesome-jenkins-rce-2019

There is no pre-auth RCE in Jenkins since May 2017, but this is the one!

It chains CVE-2018-1000861, CVE-2019-1003005 and CVE-2019-1003029 to a more reliable and elegant pre-auth remote code execution!

Affect list

- ANONYMOUS_READ disable
  - Jenkins version < 2.138
- ANONYMOUS_READ enable(or with a normal user account)
  - Jenkins build time < 2019-01-28

通过Google搜索响应漏洞，发现该版本重大漏洞，可利用CVE-2018-1000861，CVE-2019-1003005和CVE-2019-1003029组合getshell，拿到服务器权限。

Jenkins远程命令执行漏洞（CVE-2018-1000861）

Jenkins使用Stapler框架开发，其允许用户通过URL PATH来调用一次public方法。由于这个过程没有做限制，攻击者可以构造一些特殊的PATH来执行一些敏感的Java方法。

通过这个漏洞，我们可以找到很多可供利用的利用链。其中最严重的就是绕过Groovy沙盒导致未授权用户可执行任意命令：Jenkins在沙盒中执行Groovy前会先检查脚本是否有错误，检查操作是没有沙盒的，攻击者可以通过Meta-Programming的方式，在检查这个步骤时执行任意命令。参考链接：

```
http://blog.orange.tw/2019/01/hacking-jenkins-part-1-play-with-dynamic-r
http://blog.orange.tw/2019/02/abusing-meta-programming-for-unauthenticate
https://0xdf.gitlab.io/2019/02/27/playing-with-jenkins-rce-vulnerability.
```

使用 @orangetw 给出的poc，发送如下请求即可成功执行命令：

```
http://*.*.*.1:8080/securityRealm/user/admin/descriptorByName/org.jenkins
?sandbox=true
&value=public class x {
  public x(){
    "touch /tmp/tmp".execute()
  }
```

```
}
```

```
awesome-jenkins-rce git:(master) python exp.py http:/ .1:8080 "touch /tmp/tmp"
[*] ANONYMOUS_READ enable!
[*] Exploit success!(it should be :P)
```

poc地址：

```
https://github.com/orangetw/awesome-jenkins-rce-2019
```

迷雾重重

到现在为止已经可以确定这台机器被入侵，因为从态势感知上关联的攻击行为和历史记录、日志文件被清理的异常现象来看这台主机已经失陷。

但是`jenkins`是内网的系统，并未放给公网，攻击者通过什么途径入侵撕开到内网的入口呢？对此一筹莫展的我们决定下楼抽支烟顺便理一下思路。

由于`jenkins`放在`dmz`区域，最有可能的攻击路径有两条：

1. 通过钓鱼入侵：攻击者已经拿到办公网机器，入侵到`jenkins`。
2. 通过外部服务器：攻击者通过外部漏洞拿到应用服务器权限，横向移动入侵到`jenkins`。

云开雾释

当即我们兵分两路，对邮件服务器和`jenkins`下可以访问公网的服务器进行排查。还真让我们有了新的发现，发现一台`linux`服务器存在反连外部恶意域名的行为。排查并没有发现异常邮件，那么这一台`linux`服务器就有可能存在问题。

夜尽天明

登录这台linux服务器查看Web日志，发现尝试疑似木马后门上传，但根据日志中状态码反馈，均已405反馈失败，猜测这台服务器可能已经被攻击者攻破。随后我们对这台机器进行溯源。

攻击源IP统计如下：

```
[root@bogon 222]# awk '{print $1}' 2.txt |sort |uniq -c
      3 192.[redacted].137
     42 223.201.[redacted]
     21 223.201.[redacted]
     21 42.56.[redacted]
     33 47.90.[redacted]
```

攻击者IP

```
[root@bogon 222]# awk '{print $1}' asp2.txt |sort |uniq -c |sort -nr
   2728 223.201.
   1369 42.56.
   1369 223.201.
    516 121.58.
    387 23.235.
    358 192.154.
    330 218.93.
    308 132.232.
    267 115.47.
    258 156.235.
    258 121.58.
    258 103.51.
    257 154.85.
    250 23.235.
    204 218.60.
    144 211.159.
    129 45.199.
    129 45.199.
    129 45.199.
    129 43.248.
    129 219.235.
    129 156.235.
    129 156.232.
    129 121.58.
    129 121.58.
    101 140.143.
     94 119.29.
     94 116.196.1
     91 218.60.
     91 218.60.
     61 211.159.
     60 122.114.
```

上传木马失败日志：

```
185 - - [ 18:04:35 +0800] "POST //a.jsp HTTP/1.1" 405 43
.185 - - [ :18:04:36 +0800] "POST //11111.jsp HTTP/1.1" 405 4
.185 - - [ 18:04:36 +0800] "POST //11111.jsp HTTP/1.1" 405 4
.185 - - [ :18:04:36 +0800] "POST //11111.jsp HTTP/1.1" 405 4
185 - - [ :18:04:36 +0800] "POST //bak.jsp HTTP/1.1" 405 43
185 - - [ :18:04:36 +0800] "POST //bak.jsp HTTP/1.1" 405 43
185 - - [ :18:04:36 +0800] "POST //bak.jsp HTTP/1.1" 405 43
.115 - - [ 12:33:16 +0800] "POST //is/test.jsp HTTP/1.1" 405
.115 - - [ :12:33:16 +0800] "POST //is/test.jsp HTTP/1.1" 405
.115 - - [ 12:33:16 +0800] "POST //is/test.jsp HTTP/1.1" 405
.115 - - [ 12:33:17 +0800] "POST //is.test.jsp HTTP/1.1" 405
.115 - - [ :12:33:17 +0800] "POST //is.test.jsp HTTP/1.1" 405
.115 - - [ :12:33:17 +0800] "POST //is.test.jsp HTTP/1.1" 405
.115 - - [ :12:33:17 +0800] "POST //vaf/vaf.jsp HTTP/1.1" 405
.115 - - [ :12:33:17 +0800] "POST //vaf/vaf.jsp HTTP/1.1" 405
.115 - - [ :12:33:17 +0800] "POST //vaf/vaf.jsp HTTP/1.1" 405
.115 - - [ :12:33:17 +0800] "POST //1111.jsp HTTP/1.1" 405 43
.115 - - [ 12:33:17 +0800] "POST //1111.jsp HTTP/1.1" 405 43
.115 - - [ .12:33:17 +0800] "POST //1111.jsp HTTP/1.1" 405 43
.115 - - [ 12:33:18 +0800] "POST //a.jsp HTTP/1.1" 405 43
.115 - - [ :12:33:18 +0800] "POST //a.jsp HTTP/1.1" 405 43
.115 - - [ :12:33:18 +0800] "POST //a.jsp HTTP/1.1" 405 43
.115 - - [ :12:33:18 +0800] "POST //11111.jsp HTTP/1.1" 405 43
.115 - - [ :12:33:18 +0800] "POST //11111.jsp HTTP/1.1" 405 43
.115 - - [ :12:33:18 +0800] "POST //11111.jsp HTTP/1.1" 405 43
.115 - - [ :12:33:18 +0800] "POST //bak.jsp HTTP/1.1" 405 43
.115 - - [ :12:33:18 +0800] "POST //bak.jsp HTTP/1.1" 405 43
.115 - - [ 12:33:18 +0800] "POST //bak.jsp HTTP/1.1" 405 43
```

攻击者IP通过威胁情报反馈如下：

时间	情报内容	状态
16:53:37	可疑	有效
16:12:02	可疑	有效
2018-08-20 15:24:13	可疑	有效
2016-04-15 18:34:07	可疑	已过期
2015-11-26 04:36:17	可疑 劫持	已过期

读取访问日志，发现攻击者尝试痕迹。

```
[root@bogon 222]# more access.log00028 |grep jsp
.12 - - [ 17:08:18 +0800] "PUT /indexweb4.jsp/ HTTP/1.1" 405 42
.122 - - [ 15:46:11 +0800] "GET /uddi/st_KghHGpv3.jsp HTTP/1.1" 404 2075
.122 - - [ 15:46:11 +0800] "GET /uddiexplorer/st_KghHGpv3.jsp HTTP/1.1" 404 2075
.122 - - [ :15:46:21 +0800] "GET /uddi/st_SRiWWEo6.jsp HTTP/1.1" 404 2075
```

weblogic控制服务器日志：

分析weblogic控制服务器日志，发现利用反序列化漏洞经常出现的几个异常。

weblogic.rjvm.t3.MuxableSocketT3.dispatch类

```
####<[redacted] 4:19:08 PM CST> <Error> <RJVM> <webadm1> <AdminServer> <ExecuteThre
 org.apache.commons.collections.FunctorException: InvokerTransformer: The method 'new
org.apache.commons.collections.FunctorException: InvokerTransformer: The method 'newI
    at org.apache.commons.collections.functors.InvokerTransformer.transform(InvokerTr
    at org.apache.commons.collections.functors.ChainedTransformer.transform(ChainedTr
    at org.apache.commons.collections.map.TransformedMap.checkSetValue(TransformedMap
    at org.apache.commons.collections.map.AbstractInputCheckedMapDecorator$MapEntry.s
    at sun.reflect.annotation.AnnotationInvocationHandler.readObject(AnnotationInvoca
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
    at java.lang.reflect.Method.invoke(Method.java:606)
    at java.io.ObjectStreamClass.invokeReadObject(ObjectStreamClass.java:1017)
    at java.io.ObjectInputStream.readSerialData(ObjectInputStream.java:1893)
    at java.io.ObjectInputStream.readOrdinaryObject(ObjectInputStream.java:1798)
    at java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1350)
    at java.io.ObjectInputStream.readObject(ObjectInputStream.java:370)
    at weblogic.rjvm.InboundMsgAbbrev.readObject(InboundMsgAbbrev.java:67)
    at weblogic.rjvm.InboundMsgAbbrev.read(InboundMsgAbbrev.java:39)
    at weblogic.rjvm.MsgAbbrevJVMConnection.readMsgAbbrevs(MsgAbbrevJVMConnection.jav
    at weblogic.rjvm.MsgAbbrevInputStream.init(MsgAbbrevInputStream.java:212)
    at weblogic.rjvm.MsgAbbrevJVMConnection.dispatch(MsgAbbrevJVMConnection.java:507)
    at weblogic.rjvm.t3.MuxableSocketT3.dispatch(MuxableSocketT3.java:489)
    at weblogic.socket.BaseAbstractMuxableSocket.dispatch(BaseAbstractMuxableSocket.j
    at weblogic.socket.SocketMuxer.readReadySocketOnce(SocketMuxer.java:970)
    at weblogic.socket.SocketMuxer.readReadySocket(SocketMuxer.java:907)
    at weblogic.socket.NIOSocketMuxer.process(NIOSocketMuxer.java:495)
    at weblogic.socket.NIOSocketMuxer.processSockets(NIOSocketMuxer.java:461)
    at weblogic.socket.SocketReaderRequest.run(SocketReaderRequest.java:30)
    at weblogic.socket.SocketReaderRequest.execute(SocketReaderRequest.java:43)
    at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:147)
    at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:119)
Caused By: java.lang.reflect.InvocationTargetException
```

java.io.ObjectInputStream.readObject类

```
    at java.io.ObjectStreamClass.invokeReadObject(ObjectStream
    at java.io.ObjectInputStream.readSerialData(ObjectInputStr
    at java.io.ObjectInputStream.readOrdinaryObject(ObjectInpu
    at java.io.ObjectInputStream.readObject0(ObjectInputStream
    at java.io.ObjectInputStream.readObject(ObjectInputStream.
    at weblogic.rjvm.InboundMsgAbbrev.readObject(InboundMsgAbb
    at weblogic.rjvm.InboundMsgAbbrev.read(InboundMsgAbbrev.ja
    at weblogic.rjvm.MsgAbbrevJVMConnection.readMsgAbbrevs(Msg
```

反序列化缓存文件：

```
weblogic.rjvm.t3.MuxableSocketT3.dispatch(MuxableSocketT3.java:489)
weblogic.socket.BaseAbstractMuxableSocket.dispatch(BaseAbstractMuxableSoc
weblogic.socket.SocketMuxer.readReadySocketOnce(SocketMuxer.java:970)
weblogic.socket.SocketMuxer.readReadySocket(SocketMuxer.java:907)
weblogic.socket.NIOSocketMuxer.process(NIOSocketMuxer.java:495)
weblogic.socket.NIOSocketMuxer.processSockets(NIOSocketMuxer.java:461)
weblogic.socket.SocketReaderRequest.run(SocketReaderRequest.java:30)
weblogic.socket.SocketReaderRequest.execute(SocketReaderRequest.java:43)
weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:147)
weblogic.kernel.ExecuteThread.run(ExecuteThread.java:119)
By: java.io.FileNotFoundException: /c:/windows/temp/H3y5ec.tmp (No such f
java.io.FileOutputStream.open(Native Method)
java.io.FileOutputStream.<init>(FileOutputStream.java:221)
java.io.FileOutputStream.<init>(FileOutputStream.java:110)
sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
```

这个是网上的一个exp轮子，刚好入侵者是没有去修改这些特征。

附带链接：http://polaris-lab.com/index.php/archives/98/

网络weblogic反序列化exp文件。

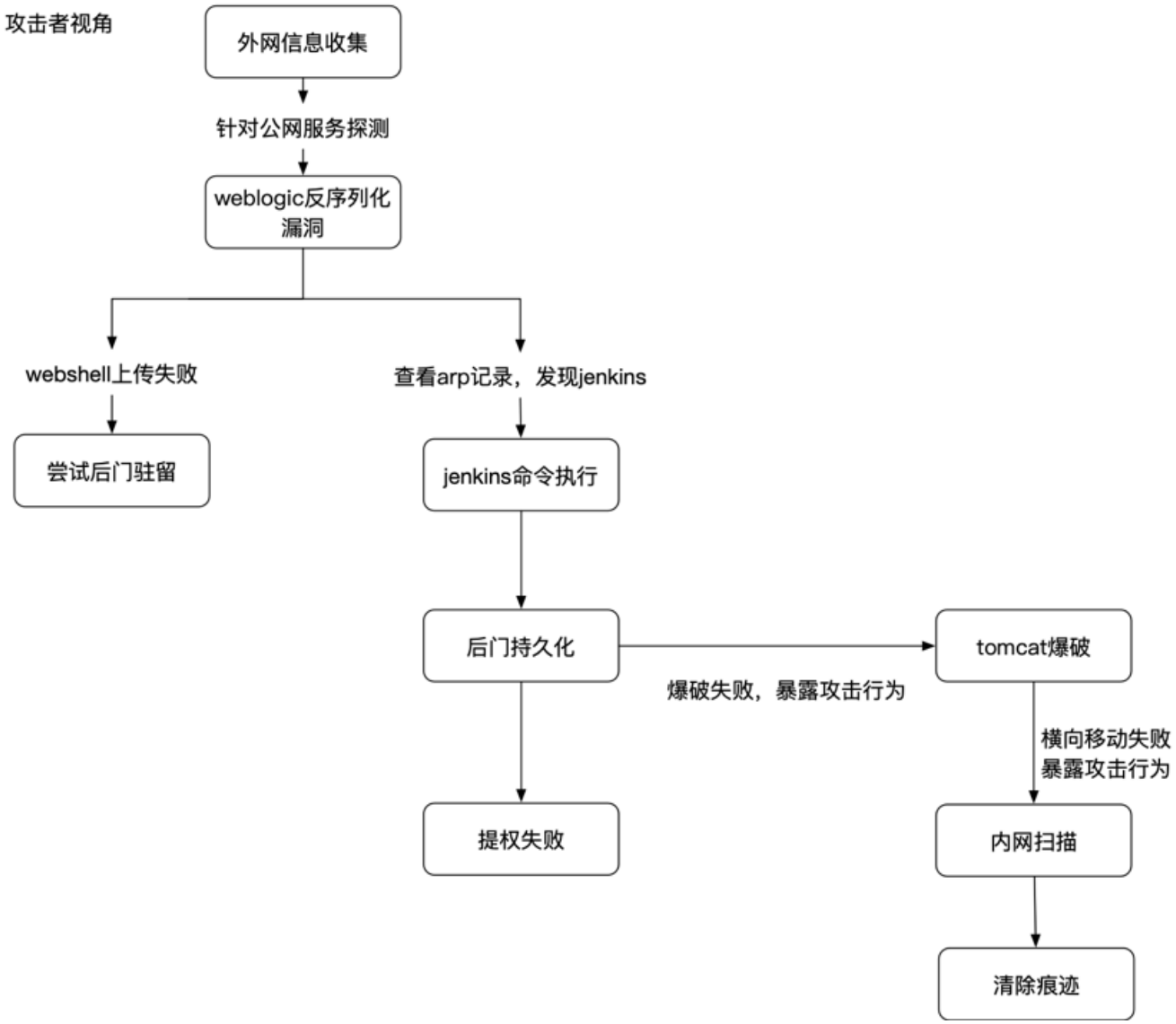
```
public class JavaExp {

    private static String remoteWindowsPath = "/c:/windows/temp/H3y5ec.tmp";
    private static String remoteLinuxPath = "/tmp/H3y5ec.tmp";
    private static String Notices = ""
            + "用法: \n"
            + "1.一句话命令执行:\n"
            + "例子: java exp.jar 127.0.0.1 7001 'net user'\n"
            + "2.文件上传\n"
            + "例子:java exp.jar 127.0.0.1 7001 upload '本地文件绝对路径' '远程目标文件绝对路
径'\n";
    private static RemoteObject remote;
    public static void main(String[] args) throws NamingException {
        if(args.length <3){
            System.out.println("参数错误！！\n");
            System.out.println(Notices);
            System.exit(0);
        }
            String host = args[0];
            String port = args[1];
            String order =  args[2];
```

通过检测weblogic控制服务器发现CVE-2018-2893漏洞。

```
PS C:\Users\d      \Desktop> python2 .\22.py 10.10.  .  7001
handshake successful
send request payload successful,recv length:1760
10.10.  :7001 is vul CVE-2018-2893
```

定位到的黑客攻击路径为：



通过一夜的努力我们终于可以大致梳理出攻击者的攻击路径，攻击者通过外网信息收集，探测到公网服务器存在`weblogic`反序列化漏洞。上传`shell`失败后，攻击者在尝试横向拓展驻留点过程中发现了我们未打补丁的`jenkins`服务，通过`jenkins`服务发起了内网扫描。

恢复：加固工作记录

首先将`Jenkins`升级到最新版本，对漏洞进行修复。再将访问权限设置为办公网有限权限访问。`weblogic`控制服务器，我们决定采用重装系统的方式。

总结分析

1. 此`weblogic`系统因为是临时启用，内网仅有jenkins服务访问权限，因为使用数据才临时启用一周，未经过安全审批，未及时同步信息。

2. weblogic服务对应内部IP为10.10.*.*，目前对互联网仅开放其80，8080，7001端口，22端口只能通过内部访问。
3. Jenkins服务部署的纯内网服务器，未更新安全补丁。
4. 系统账号正常。
5. 网络连接情况正常。
6. 历史命令曾存在手工清除痕迹。
7. 定时任务正常。
8. 启动项正常。
9. Jenkins日志正常
10. 系统层面黑客使用的脚本。
11. 结合IDS记录，通过入侵方向推演，攻击者通过信息收集发现公网服务器存在weblogic反序列化漏洞。以反序列化漏洞为入口，入侵后对Jenkins进行攻击，获得权限后尝试横向移动过程中触发告警。

复盘反思

1. 安全团队和业务团队之间有高效的沟通机制和较强的合作意识。
2. 提升内部安全意识，及时关注各补丁推送。

知其黑 守其白

分享知识盛宴，闲聊大院趣事，备好酒肉等你

长按二维码关注 酒仙桥六号部队