Weblogic WLS 组件漏洞

技术分析与防护方案

发布时间：2017 年 12 月 21 日

综述

近日，绿盟科技应急响应团队也陆续接到来自金融、运营商及互联网等多个行业的客户的安全事件的反馈，发现 Weblogic 主机被攻击者植入恶意程序，经分析，攻击者利用 Weblogic WLS 组件漏洞(CVE-2017-10271)，构造 payload 下载并执行虚拟币挖矿程序，对 Weblogic 中间件主机进行攻击。

受影响的版本

- Weblogic Server 10.3.6.0.0,
- Weblogic Server 12.1.3.0.0,
- Weblogic Server 12.2.1.1.0,
- Weblogic Server 12.2.1.2.0

以上均为 Weblogic 官方还在支持的版本

不受影响的版本

- Weblogic Server 12.2.1.3

用户自查

由于此次攻击主要目的为下载执行挖矿程序，从主机层面可通过监控主机系统资源或进程分析方式进行检测，从网络层面可对 C&C 地址及矿池相关域名/IP 进行监控，以发现其他受感染主机。

针对 linux 主机，首先查看/tmp 目录中是否存在属主为 WebLogic 运行账户的相关可疑文件，如：watch-smartd、Carbon、default。

```
[root@172-1-2-88 ~]# ls -ls /tmp/
total 2240
   4 drwxr-x--- 2 tomcat   tomcat     4096 Dec 20 02:48 hsperfdata_tomcat
   4 drwxr----- 2 weblogic bea        4096 Dec 20 02:48 hsperfdata_weblogic
2224 -rwxr--r-- 1 weblogic bea     2274080 Dec 19 21:13 watch-smartd
   4 drwxr-x--x 3 root     root       4096 Oct 31 23:47 wlstTemproot
   4 drwxr-x--x 3 weblogic bea        4096 Oct 31 09:31 wlstTempweblogic
[root@172-1-2-88 ~]# ps -ef |grep watch
root         6     2  0 02:47 ?        00:00:00 [watchdog/0]
weblogic  1832     1  0 17:37 ?        00:00:00 ./watch-smartd -B
root      1852  1814  0 17:43 pts/1    00:00:00 grep watch
[root@172-1-2-88 ~]#
```

同时通过进程及系统资源分析，是否存在启动用户为 WebLogic 运行账户的相关可疑进程。

```
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1099 weblogic  20   0 1550m 360m  40m S  0.0 19.2   1:37.93 java
 1100 weblogic  20   0 1559m 345m  44m S  1.3 18.5   2:20.81 java
 1062 tomcat    20   0 2435m  89m  13m S  0.0  4.8   1:09.17 java
 1753 root      20   0 99.2m 5700 3420 S  0.0  0.3   0:00.52 sshd
 1832 weblogic  20   0  325m 3476 1120 S  0.0  0.2   0:00.06 watch-smartd
 1758 root      20   0  105m 1912 1548 S  0.0  0.1   0:00.04 bash
 1786 weblogic  20   0  105m 1896 1552 S  0.0  0.1   0:00.02 bash
 1814 root      20   0  105m 1888 1528 S  0.0  0.1   0:00.01 bash
  916 root      20   0  243m 1668 1104 S  0.0  0.1   0:00.04 rsyslogd
 1785 root      20   0  141m 1580 1212 S  0.0  0.1   0:00.00 su
  948 weblogic  20   0  103m 1540 1228 S  0.0  0.1   0:00.03 startWebLogic.s
  934 weblogic  20   0  103m 1536 1228 S  0.0  0.1   0:00.04 startWebLogic.s
    1 root      20   0 19232 1508 1240 S  0.0  0.1   0:00.70 init
```

同时在网络层，通过防火墙或相关的入侵防御设备，对 C&C 地址及矿池相关域名/IP 进行监测，涉及域名及 IP 包括：

```
minergate.com
minexmr.com
78.46.91.134
104.25.208.15
104.25.209.15
136.243.102.167
136.243.102.154
94.130.143.162
88.99.142.163
72.11.140.178
```

官方修复方案

Oracle 官方对于 Weblogic WLS 组件漏洞(CVE-2017-10271)在 10 月份的更新补丁中已经进行了修复，建议用户及时下载更新包，升级至最新版本进行防护。

下载链接：

http://www.oracle.com/technetwork/middleware/weblogic/downloads/index.html

临时防护方案

根据攻击者利用 POC 分析发现所利用的为 wls-wsat 组件的 CoordinatorPortType 接口，若 Weblogic 服务器集群中未应用此组件，建议临时备份后将此组件删除。

1. 根据实际环境路径，删除 WebLogic wls-wsat 组件：

```
rm -f /home/WebLogic/Oracle/Middleware/wlserver_10.3/server/lib/wls-wsat.war

rm -f
/home/WebLogic/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/tmp/.internal/wls-wsat.war

rm -rf
/home/WebLogic/Oracle/Middleware/user_projects/domains/base_domain/servers/AdminServer/tmp/_WL_internal/wls-wsat
```

2. 重启 Weblogic 域控制器服务。

```
DOMAIN_NAME/bin/stopWeblogic.sh      #停止服务
DOMAIN_NAME/bin/startManagedWebLogic.sh      #启动服务
```

关于重启 Weblogic 服务的详细信息，可参考如下官方文档：

https://docs.oracle.com/cd/E13222_01/wls/docs90/server_start/overview.html

绿盟科技防护建议

绿盟科技检测类产品与服务

- 公网资产可使用绿盟云 紧急漏洞在线检测，检测地址如下：

https://cloud.nsfocus.com/#/krosa/views/initcdr/productandservice?page_id=12

- 内网资产可以使用绿盟科技的远程安全评估系统(RSAS V6/V5)或 Web 应用漏洞扫描系统（WVSS） 进行检测:
  - ◆ 远程安全评估系统（RSAS V6）

http://update.nsfocus.com/update/listRsasDetail/v/vulweb

  - ◆ 远程安全评估系统（RSAS V5）

http://update.nsfocus.com/update/listAurora/v/5

  - ◆ Web 应用漏洞扫描系统（WVSS）

http://update.nsfocus.com/update/listWvssDetail/v/6/t/plg

- 内网资产可以使用绿盟科技的入侵检测系统(IDS) 进行检测。
  入侵检测系统（IDS）

http://update.nsfocus.com/update/listIds

通过上述链接，升级至最新版本即可进行检测

使用绿盟科技防护类产品（IPS/NF/WAF）进行防护：

- 入侵防护系统（IPS）
  http://update.nsfocus.com/update/listIps
- 下一代防火墙系统（NF）
  http://update.nsfocus.com/update/listNf
- Web 应用防护系统（WAF）
  http://update.nsfocus.com/update/wafIndex

通过上述链接，升级至最新版本即可进行防护！

技术分析

首先来看下 weblogic 的历史补丁，四月份补丁通告：
http://www.oracle.com/technetwork/security-advisory/cpuapr2017-3236618.html

CVE-2017-3506	Oracle WebLogic Server	Web Services	HTTP	Yes	7.4	Network	High	None	None	Un-changed	High	High	None	10.3.6.0, 12.1.3.0, 12.2.1.0, 12.2.1.1, 12.2.1.2

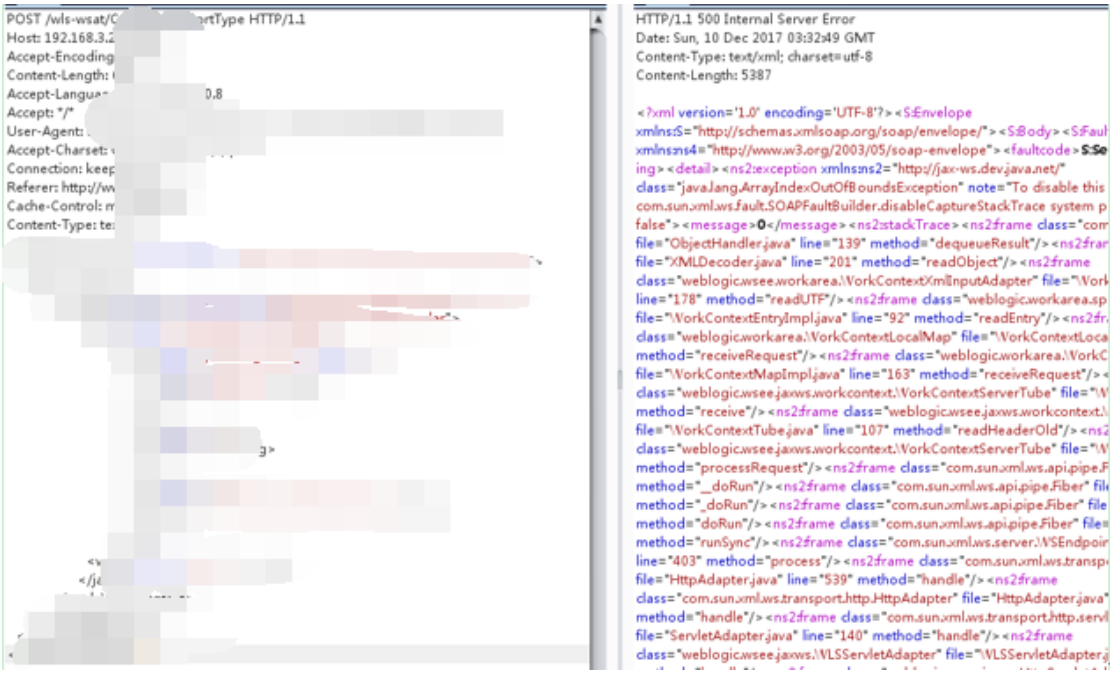
10 月份补丁通告：
https://www.oracle.com/technetwork/topics/security/cpuoct2017-3236626.html

CVE-2017-10271	Oracle WebLogic Server	WLS Security	HTTP	Yes	7.5	Network	Low	None	None	Un-changed	None	None	High	10.3.6.0.0, 12.1.3.0.0, 12.2.1.1.0, 12.2.1.2.0

“remote user can exploit a flaw in the Oracle WebLogic Server WLS Security component to gain elevated privileges [CVE-2017-10271]”

这是 CVE-2017-10271 的描述信息。

通过 PoC 测试，结果如下：



跟踪到底这还是 XMLDecoder 的漏洞，下面来分析补丁代码：首先来看 3506 的补丁的分析，在文件 weblogic/wsee/workarea/WorkContextXmlInputAdapter.java 中，添加了 validate 方法，方法的实现如下：

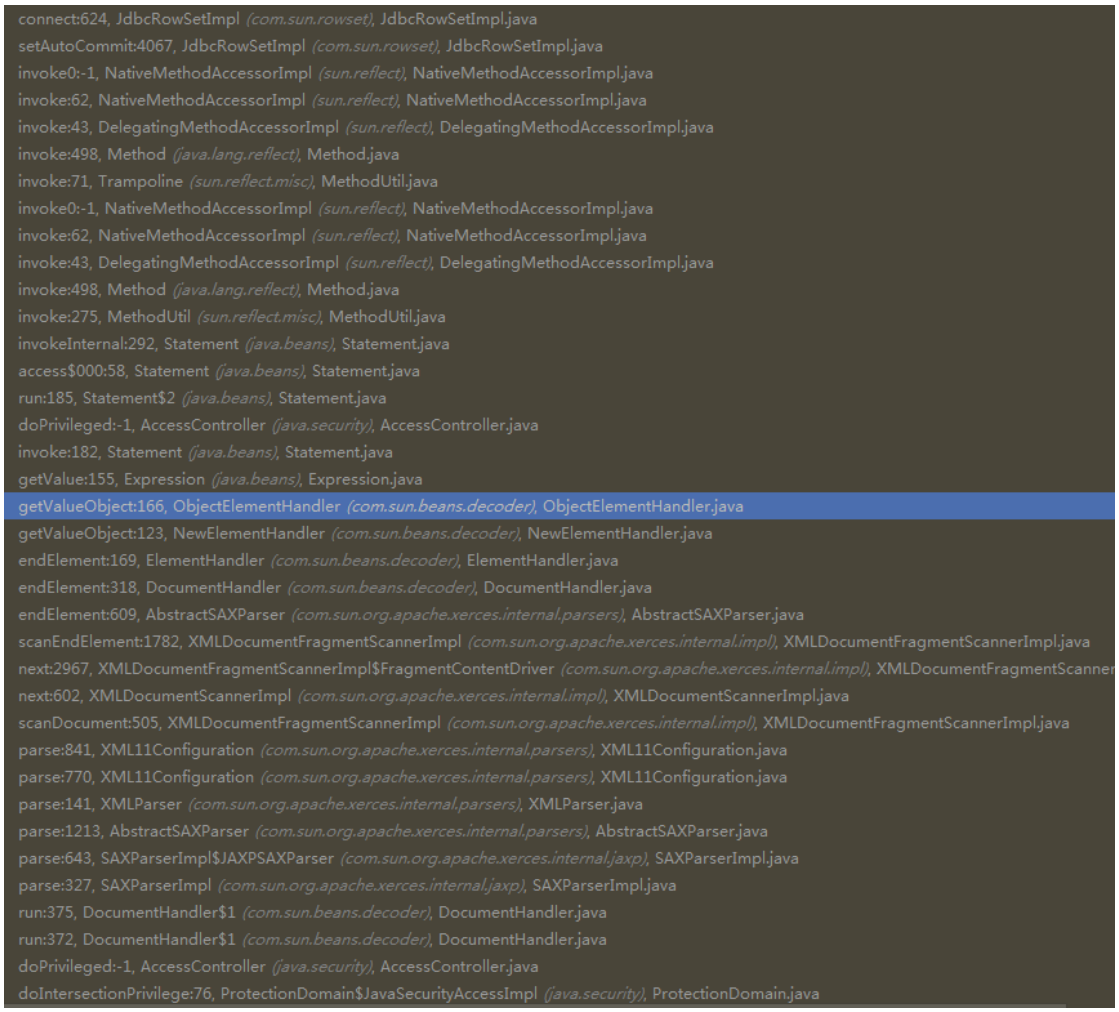
```
private void validate(InputStream is) {
        WebLogicSAXParserFactory factory = new
WebLogicSAXParserFactory();

        try {
            SAXParser parser = factory.newSAXParser();
            parser.parse(is, new DefaultHandler() {
                public void startElement(String uri, String
localName, String qName, Attributes attributes) throws SAXException {
                    if(qName.equalsIgnoreCase("object")) {
                        throw new IllegalStateException("Invalid
context type: object");
                    }
                }
            });
        } catch (ParserConfigurationException var5) {
            throw new IllegalStateException("Parser Exception",
var5);
        } catch (SAXException var6) {
            throw new IllegalStateException("Parser Exception",
var6);
```

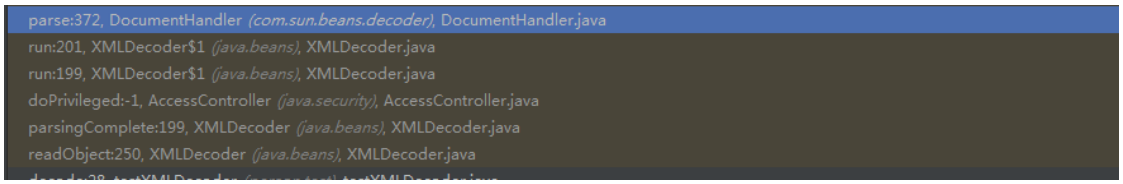
```
        } catch (IOException var7) {
            throw new IllegalStateException("Parser Exception",
var7);
        }
    }
```

简单来说就是在解析 xml 的过程中，如果 Element 字段值为 Object 就抛出异常，这修复显得有些业余，所以马上就有了 CVE-2017-10271。前段时间分析 Oracle Weblogic 十月份的补丁的时候看到 WorkContextXmlInputAdapter 相关代码时只关注了这块 dos 的漏洞，没看到 10271 添加的对 new，method，void 的去除。因为可以通过其他方式绕过，比如典型的就是将 object 改为 void，这是挖矿的人采用的 PoC，当然也还可以通过 new 关键字来创建反序列化执行的 PoC。

至于为什么 XMLDecoder 在解析的过程中能执行代码呢，下面我们来动态分析。以如下 poc 示例：

根据如上 PoC，首先生成 JdbcRowSetImpl 的实例，接着调用该实例的 set 方法来初始化该实例的属性，当调用完 setAutoCommit 接口的时候就会根据 dataSourceName 的值去远程加载一个类初始化。下图为调用栈的结果：





针对上面的 PoC，官方放出了 10271 的补丁，补丁如下：

```
private void validate(InputStream is) {
    WebLogicSAXParserFactory factory = new WebLogicSAXParserFactory();

    try {
        SAXParser parser = factory.newSAXParser();
        parser.parse(is, new DefaultHandler() {
            private int overallarraylength = 0;

            public void startElement(String uri, String localName, String qName, A
ttributes attributes) throws SAXException {
                if(qName.equalsIgnoreCase("object")) {
                    throw new IllegalStateException("Invalid element qName:object");
```

```
                } else if(qName.equalsIgnoreCase("new")) {
                    throw new IllegalStateException("Invalid element qName:new");
                } else if(qName.equalsIgnoreCase("method")) {
                    throw new IllegalStateException("Invalid element qName:method");
                } else {
                    if(qName.equalsIgnoreCase("void")) {
                        for(int attClass = 0; attClass < attributes.getLength(); ++
attClass) {
                            if(!"index".equalsIgnoreCase(attributes.getQName(attClass)))
 {
                                throw new IllegalStateException("Invalid attribute fo
r element void:" + attributes.getQName(attClass));
                            }
                        }
                    }

                    if(qName.equalsIgnoreCase("array")) {
                        String var9 = attributes.getValue("class");
                        if(var9 != null && !var9.equalsIgnoreCase("byte")) {
                            throw new IllegalStateException("The value of class att
ribute is not valid for array element.");
                        }
```

该补丁较为完整，通过限定 object，new,method,void，array 等字段来防止攻击者绕过。

声 明

本安全公告仅用来描述可能存在的安全问题，绿盟科技不为此安全公告提供任何保证或承诺。由于传播、利用此安全公告所提供的信息而造成的任何直接或者间接的后果及损失，均由使用者本人负责，绿盟科技以及安全公告作者不为此承担任何责任。绿盟科技拥有对此安全公告的修改和解释权。如欲转载或传播此安全公告，必须保证此安全公告的完整性，包括版权声明等全部内容。未经绿盟科技允许，不得任意修改或者增减此安全公告内容，不得以任何方式将其用于商业目的。

关于绿盟科技

北京神州绿盟信息安全科技股份有限公司（简称绿盟科技）成立于 2000 年 4 月，总部位于北京。在国内外设有 30 多个分支机构，为政府、运营商、金融、能源、互联网以及教育、医疗等行业用户，提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。

基于多年的安全攻防研究，绿盟科技在网络及终端安全、互联网基础安全、合规及安全管理等领域，为客户提供入侵检测/防护、抗拒绝服务攻击、远程安全评估以及 Web 安全防护等产品以及专业安全服务。

北京神州绿盟信息安全科技股份有限公司于 2014 年 1 月 29 日起在深圳证券交易所创业板上市交易，股票简称：绿盟科技，股票代码：300369。

绿盟科技官方微博二维码

绿盟科技官方微信二维码