CobaltStrike去除流量特征

CobaltStrike去除流量特征

普通CS没有做流量混淆会被防火墙拦住流量，所以偶尔会看到CS上线了机器但是进行任何操作都没有

参考网上的文章，大部分是两种方法，一种更改teamserver 里面与CS流量相关的内容，一种是利用k

我们需要做的修改大概为3个地方：

```
1. 修改默认端口
2. 去除store证书特征
3. 修改profile
```

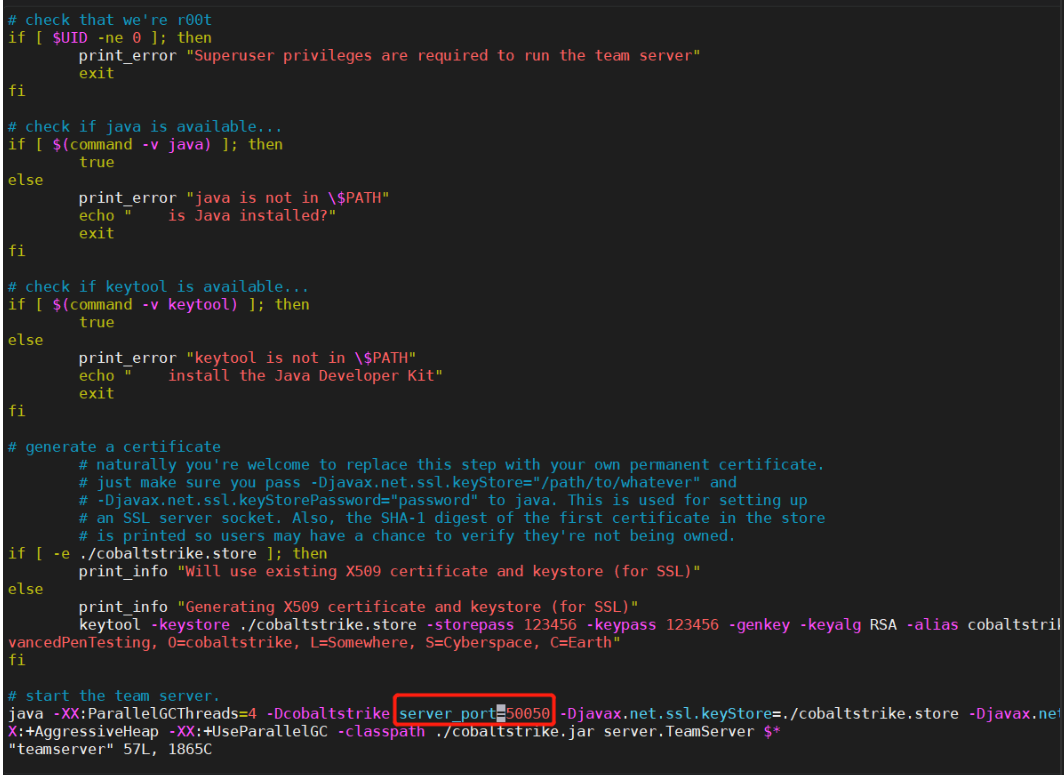
0x00 关闭后台运行的CS

```
ps -aux
找到CS相关的进程
kill -9 pid
```

0x01 修改默认端口

编辑teamserver文件，更改server port部分 50433

```
vim teamserver
```



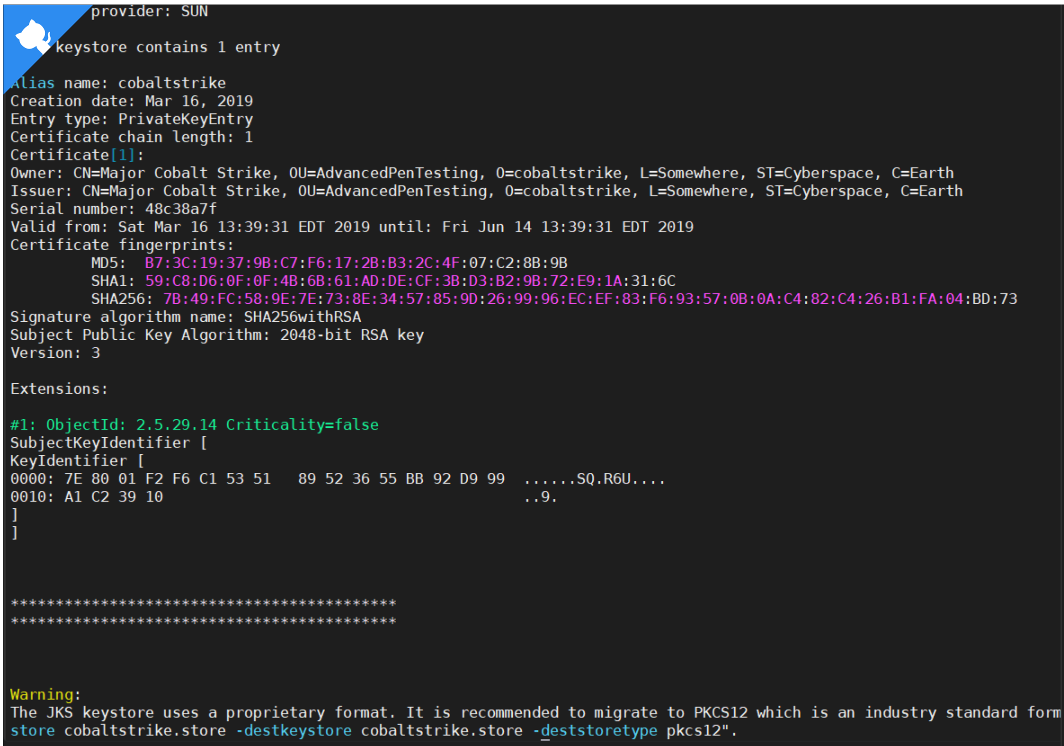
```
# check that we're r00t
if [ $UID -ne 0 ]; then
        print_error "Superuser privileges are required to run the team server"
        exit
fi

# check if java is available...
if [ $(command -v java) ]; then
        true
else
        print_error "java is not in \$PATH"
        echo "    is Java installed?"
        exit
fi

# check if keytool is available...
if [ $(command -v keytool) ]; then
        true
else
        print_error "keytool is not in \$PATH"
        echo "    install the Java Developer Kit"
        exit
fi

# generate a certificate
        # naturally you're welcome to replace this step with your own permanent certificate.
        # just make sure you pass -Djavax.net.ssl.keyStore="/path/to/whatever" and
        # -Djavax.net.ssl.keyStorePassword="password" to java. This is used for setting up
        # an SSL server socket. Also, the SHA-1 digest of the first certificate in the store
        # is printed so users may have a chance to verify they're not being owned.
if [ -e ./cobaltstrike.store ]; then
        print_info "Will use existing X509 certificate and keystore (for SSL)"
else
        print_info "Generating X509 certificate and keystore (for SSL)"
        keytool -keystore ./cobaltstrike.store -storepass 123456 -keypass 123456 -genkey -keyalg RSA -alias cobaltstrik
vancedPenTesting, O=cobaltstrike, L=Somewhere, S=Cyberspace, C=Earth"
fi

# start the team server.
java -XX:ParallelGCThreads=4 -Dcobaltstrike server_port=50050 -Djavax.net.ssl.keyStore=./cobaltstrike.store -Djavax.net
X:+AggressiveHeap -XX:+UseParallelGC -classpath ./cobaltstrike.jar server.TeamServer $*
"teamserver" 57L, 1865C
```

0x02 去除store证书特征

查看证书，默认密码123456

```
keytool -list -v -keystore cobaltstrike.store
```

可以看到未修改的证书还是有很明显的cs特征的，比如 `Alias name` `Owner` `Issuer` 字段



```
        provider: SUN

    keystore contains 1 entry

Alias name: cobaltstrike
Creation date: Mar 16, 2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Major Cobalt Strike, OU=AdvancedPenTesting, O=cobaltstrike, L=Somewhere, ST=Cyberspace, C=Earth
Issuer: CN=Major Cobalt Strike, OU=AdvancedPenTesting, O=cobaltstrike, L=Somewhere, ST=Cyberspace, C=Earth
Serial number: 48c38a7f
Valid from: Sat Mar 16 13:39:31 EDT 2019 until: Fri Jun 14 13:39:31 EDT 2019
Certificate fingerprints:
         MD5:  B7:3C:19:37:9B:C7:F6:17:2B:B3:2C:4F:07:C2:8B:9B
         SHA1: 59:C8:D6:0F:0F:4B:6B:61:AD:DE:CF:3B:D3:B2:9B:72:E9:1A:31:6C
         SHA256: 7B:49:FC:58:9E:7E:73:8E:34:57:85:9D:26:99:96:EC:EF:83:F6:93:57:0B:0A:C4:82:C4:26:B1:FA:04:BD:73
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 7E 80 01 F2 F6 C1 53 51   89 52 36 55 BB 92 D9 99  ......SQ.R6U....
0010: A1 C2 39 10                                        ..9.
]
]



*******************************************
*******************************************



Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard form
store cobaltstrike.store -destkeystore cobaltstrike.store -deststoretype pkcs12".
```

而Keytool是一个Java的证书管理工具，下面用Keytool生成一个store证书。

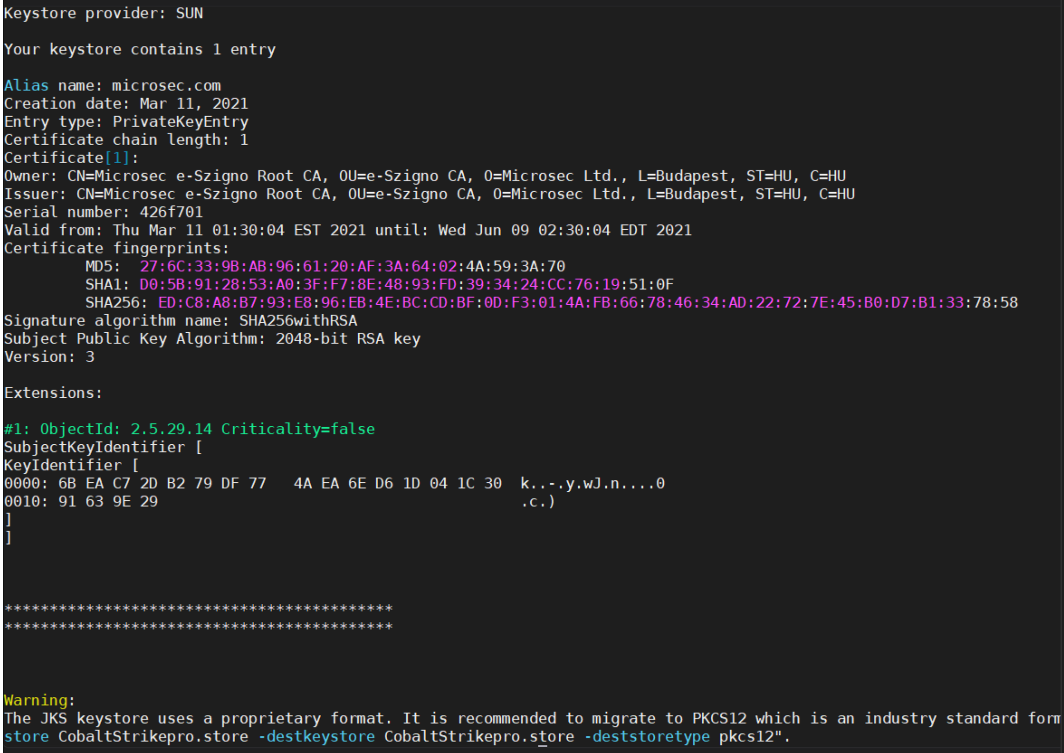
```
keytool -h
Illegal option:  -h
ey and Certificate Management Tool


Commands:

-certreq            Generates a certificate request

-changealias        Changes an entry's alias

-delete             Deletes an entry

-exportcert         Exports certificate

-genkeypair         Generates a key pair

-genseckey          Generates a secret key

-gencert            Generates certificate from a certificate request

-importcert         Imports a certificate or a certificate chain

-importpass         Imports a password

-importkeystore     Imports one or all entries from another keystore

-keypasswd          Changes the key password of an entry

-list               Lists entries in a keystore

-printcert          Prints the content of a certificate

-printcertreq       Prints the content of a certificate request

-printcrl           Prints the content of a CRL file
```

```
orepasswd          Changes the store password of a keystore
```

使用以下命令生成一个新的store证书，`-alias` 和 `-dname` 可以自由发挥，也可以用其他的来混淆

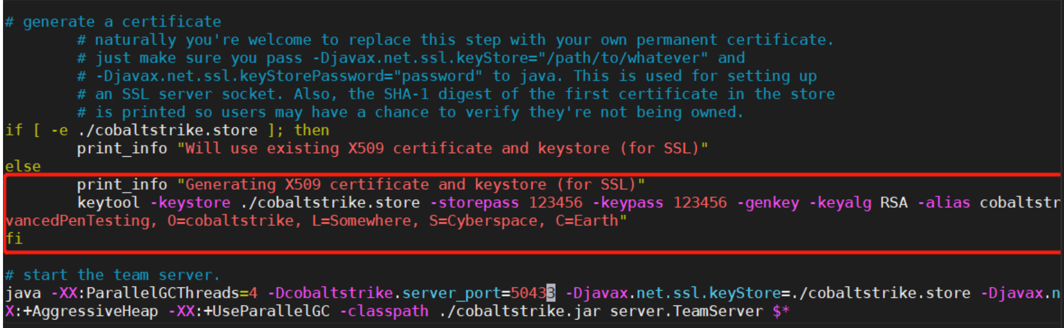
```
keytool -keystore CobaltStrikepro.store -storepass 123456 -keypass 123456 -genkey -k


参数                含义

`-alias`          指定别名
`-storepass`      指定更改密钥库的存储口令
`-keypass pass`   指定更改条目的密钥口令
`-keyalg`         指定算法
`-dname`          指定所有者信息
```

新生成的证书看着就很nice



```
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: microsec.com
Creation date: Mar 11, 2021
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Microsec e-Szigno Root CA, OU=e-Szigno CA, O=Microsec Ltd., L=Budapest, ST=HU, C=HU
Issuer: CN=Microsec e-Szigno Root CA, OU=e-Szigno CA, O=Microsec Ltd., L=Budapest, ST=HU, C=HU
Serial number: 426f701
Valid from: Thu Mar 11 01:30:04 EST 2021 until: Wed Jun 09 02:30:04 EDT 2021
Certificate fingerprints:
         MD5:  27:6C:33:9B:AB:96:61:20:AF:3A:64:02:4A:59:3A:70
         SHA1: D0:5B:91:28:53:A0:3F:F7:8E:48:93:FD:39:34:24:CC:76:19:51:0F
         SHA256: ED:C8:A8:B7:93:E8:96:EB:4E:BC:CD:BF:0D:F3:01:4A:FB:66:78:46:34:AD:22:72:7E:45:B0:D7:B1:33:78:58
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 6B EA C7 2D B2 79 DF 77   4A EA 6E D6 1D 04 1C 30  k..-.y.wJ.n....0
0010: 91 63 9E 29                                        .c.)
]
]



*******************************************
*******************************************



Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard form
store CobaltStrikepro.store -destkeystore CobaltStrikepro.store -deststoretype pkcs12".
```

当然也可以编辑 `teamserver` 文件来生成证书



```
# generate a certificate
        # naturally you're welcome to replace this step with your own permanent certificate.
        # just make sure you pass -Djavax.net.ssl.keyStore="/path/to/whatever" and
        # -Djavax.net.ssl.keyStorePassword="password" to java. This is used for setting up
        # an SSL server socket. Also, the SHA-1 digest of the first certificate in the store
        # is printed so users may have a chance to verify they're not being owned.
if [ -e ./cobaltstrike.store ]; then
        print_info "Will use existing X509 certificate and keystore (for SSL)"
else
        print_info "Generating X509 certificate and keystore (for SSL)"
        keytool -keystore ./cobaltstrike.store -storepass 123456 -keypass 123456 -genkey -keyalg RSA -alias cobaltstr
vancedPenTesting, O=cobaltstrike, L=Somewhere, S=Cyberspace, C=Earth"
fi

# start the team server.
java -XX:ParallelGCThreads=4 -Dcobaltstrike.server_port=50433 -Djavax.net.ssl.keyStore=./cobaltstrike.store -Djavax.n
X:+AggressiveHeap -XX:+UseParallelGC -classpath ./cobaltstrike.jar server.TeamServer $*
```

0x03 Malleable-C2-Profiles

因为现在很多WAF都能检测出CS的流量特征，而CS的流量由 `Malleable C2` 配置来掌控的，所以我们

`C2` 。

Beacon与teamserver端c2的通信逻辑

```
stager的beacon会先下载完整的payload执行
2.beacon进入睡眠状态，结束睡眠状态后用 http-get方式 发送一个metadata(具体发送细节可以在
3.如果存在待执行的任务，则teamserver上的c2会响应这个metadata发布命令。beacon将会收到具
4.执行完毕后beacon将回显数据与任务id用post方式发送回team server端的C2(细节可以在mallea
```

首先需要先下载profile文件

```
git clone https://github.com/rsmudge/Malleable-C2-Profiles.git
```

CS中集成了一个包含在Linux平台下的 `c2lint` 工具，可以检查profile代码是否有问题

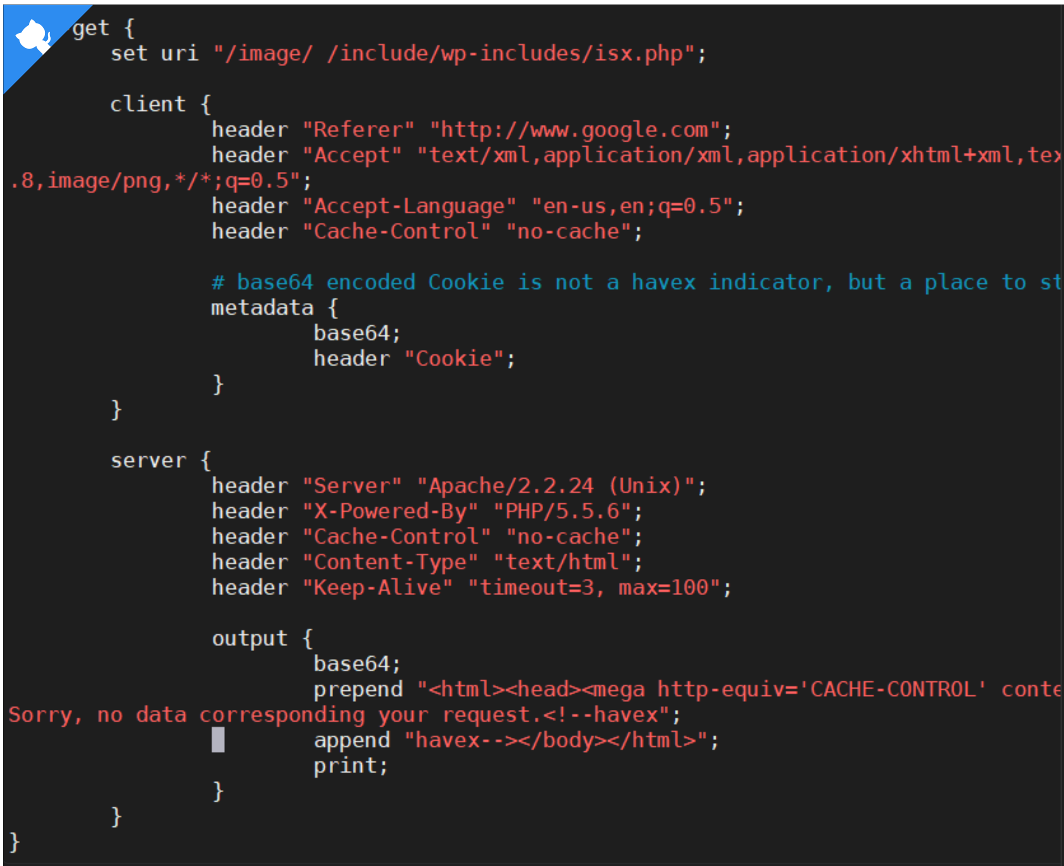
```
chmod 777 c2lint
./c2lint ./Malleable-C2-Profiles/APT/havex.profile
```

之后改一下profile的内容就好了网上有很多例子，我这里简单改了下。

因为0.0.0.0是Cobalt Strike DNS Beacon特征，可以在profile内加一段 `set dns_idle "8.8.8.8";`

```
set sample_name "google";
set dns_idle "8.8.8.8"
set sleeptime "30000";

set useragent "Mozilla/5.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/

set pipename "mypipe-f##";
set pipename_stager "mypipe-h##";

# Clone some header values (Sample from: https://malshare.com/sample.php?a
d5740b2f27964a)
# ./peclone c6e161a948f4474849d5740b2f27964a
stage {
        set checksum       "0";
        set compile_time   "30 Dec 2013 07:53:48";
        set entry_point    "134733";
        set image_size_x86 "348160";
        set image_size_x64 "348160";
        set name           "Tmprovider.dll";
        set rich_header    "\x63\x02\x25\x0f\x27\x63\x4b\x5c\x27\x63\x4b\x
```

http-get部分，包括uri和header都可以根据实战抓包进行修改。



```
get {
        set uri "/image/ /include/wp-includes/isx.php";

        client {
                header "Referer" "http://www.google.com";
                header "Accept" "text/xml,application/xml,application/xhtml+xml,tex
.8,image/png,*/*;q=0.5";
                header "Accept-Language" "en-us,en;q=0.5";
                header "Cache-Control" "no-cache";

                # base64 encoded Cookie is not a havex indicator, but a place to st
                metadata {
                        base64;
                        header "Cookie";
                }
        }

        server {
                header "Server" "Apache/2.2.24 (Unix)";
                header "X-Powered-By" "PHP/5.5.6";
                header "Cache-Control" "no-cache";
                header "Content-Type" "text/html";
                header "Keep-Alive" "timeout=3, max=100";

                output {
                        base64;
                        prepend "<html><head><mega http-equiv='CACHE-CONTROL' conte
Sorry, no data corresponding your request.<!--havex";
                        append "havex--></body></html>";
                        print;
                }
        }
}
```

Reference

https://www.adminxe.com/1489.html

https://www.chabug.org/web/832.html

https://paper.seebug.org/1349/

https://blog.csdn.net/shuteer_xu/article/details/110508415

> 作者：CoLoo
> 出处：https://www.cnblogs.com/CoLo/p/14518441.html
>
> 热爱技术

分类: 内网渗透

« 上一篇： Fastjson1.2.24RCE漏洞复现
» 下一篇： CVE-2020-1472 Zerologon

posted @ 2021-03-11 15:49 CoL