

# How to Hide Your CobaltStrike

## CobaltStrike Overview

Cobalt Strike is the most prevalent threat emulation software packages used by infosec red team. Often referred to as CS in the industry.

It has become an indispensable weapon in penetration testing and red teams. It has a variety of protocol hosted online methods, integrated privilege escalation, credential export, port forwarding, socket proxy, office attack, file bundling, phishing and other functions. At the same time, Cobalt Strike can also call other well-known tools such as Mimikatz, so it is widely loved by hackers.

Project official website: <https://www.cobaltstrike.com>

CS has become so widely used that most of its out-of-the box characteristics have been identified and even marked by WAF manufacturers. To learn more about this I would STRONGLY recommend the following blogs.

1. [Detecting Exposed Cobalt Strike DNS Redirectors](#) - by F-Secure
2. [Analyzing Cobalt Strike for Fun and Profit](#) - by Etienne "tek" Maynier

Think about it. If a WAF can spot CS traffic, and ban your IP then the source can eventually be traced and using a [simple BF script](#) your C2 is compromised as it will eventually become overwhelmed. We NEED to hide our CS.

### 1. Modify the default port

First things first, you have to change the teamserver port. If port 50050 is open for what's essentially the controller that's no good your C2 is too easy to identify. For example, any server using port 50050 that also provides an HTTP response unique to NanoHTTPD web servers is more likely a Cobalt Strike server than one found to only exhibit an HTTP response signature.

```
keytool -keystore ./cobaltstrike.store -storepass 123456 -keypass 123456 -genkey -keyalg RSA -alias cobaltstrike -dname CN=Major Cobalt Strike, o=AdvancedPenTesting, ou=cobaltstrike, L=Somewhere, S=Cyberspace, C=Earth"
fi

# start the team server.
java -XX:ParallelGCThreads=4 -Dcobaltstrike.server_port=50050 -Dcobaltstrike.server_bindto=0.0.0.0 -Djavax.net.ssl.keyStore=./cobaltstrike.store -Djavax.net.ssl.keyStorePassword=123456 -server -XX:+AggressiveHeap -XX:+UseParallelGC -classpath ./cobaltstrike.jar server.TeamServer $*
```

```
(root@boschko)~[~/Downloads/cobaltstrike]
# ./teamserver 192.168.87.40 boschko
[*] Will use existing X509 certificate and keystore (for SSL)
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Team server is up on 0.0.0.0:23921
[*] SHA256 hash of SSL cert is: 5f2d8152f59c4e4dc06b9643aa45f7baa423ea63a566f034bf0b828a73f582e9
[+] Listener: httpBeacon started!
```

2.

## 3. Remove the certificate feature

The default certificate of Cobalt Strike has been marked as bad by the waf manufacturer. We need to regenerate a new certificate. Here we use the keytool certificate tool that comes with the JDK to generate a new certificate.

You can use the keytool command in linux.

keytool

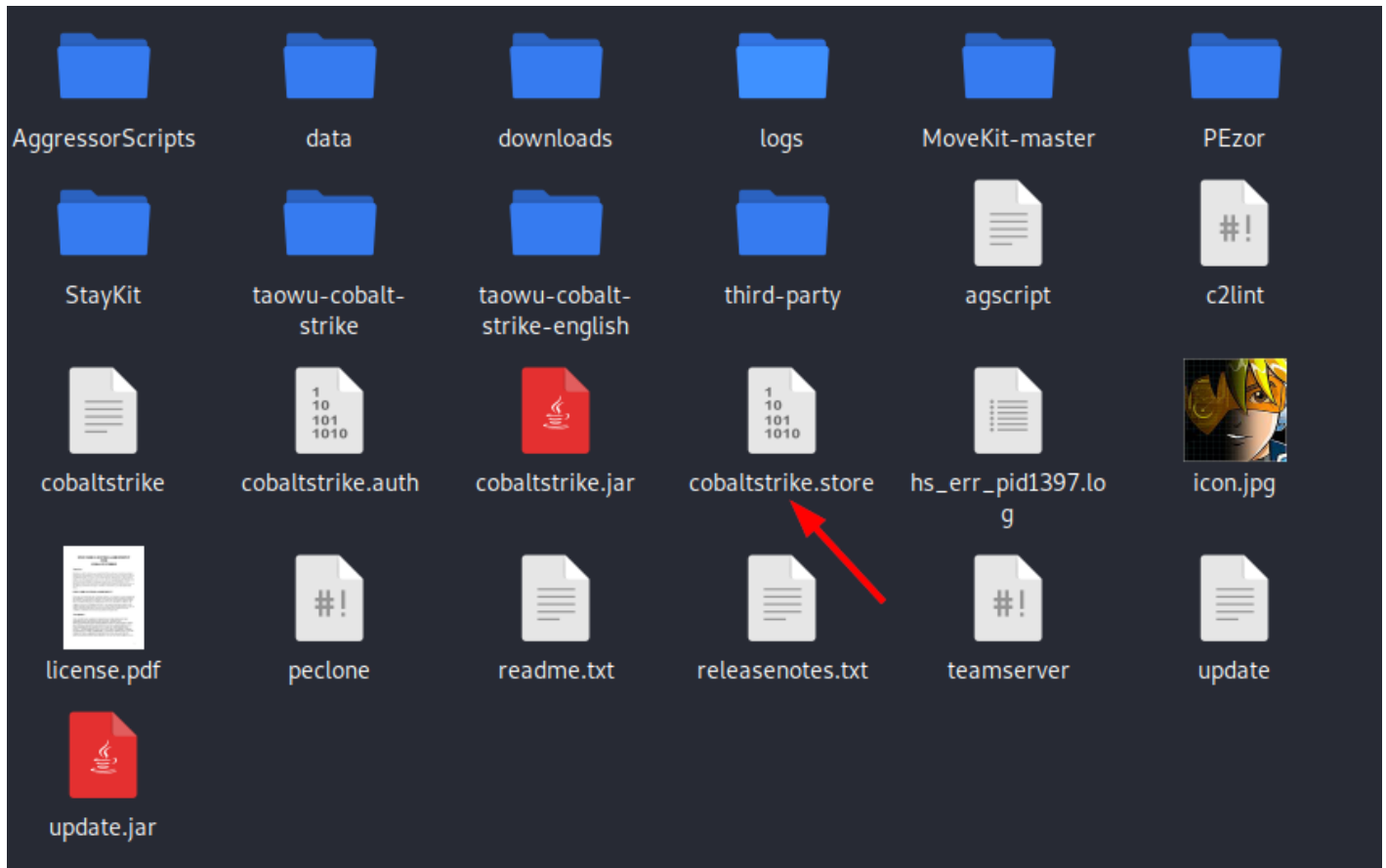
Key and certificate management tools

command:

- certreq Generate certificate request
- changealias change the alias of the entry
- delete delete entry
- exportcert export certificate
- genkeypair Generate key pair
- genseckey Generate key
- gencert generates a certificate based on the certificate request
- importcert import certificate or certificate chain
- importpass import password
- importkeystore imports one or all entries from other keystores
- keypasswd changes the key password of the entry
- list lists the entries in the keystore
- printcert print certificate content
- printcertreq print the content of the certificate request
- printcrl print the contents of the CRL file
- storepasswd change the storage password of the keystore

Use "keytool -command\_name -help" to get the usage of command\_name

Theres a bit of data in the key-store, but for the most part it contains: Key entity-secret key or private key and paired public key (using asymmetric encryption) Trusted certificate entries-only public key.



Check the default certificate of cs, the password is 123456

Revision #2

Created 27 June 2021 03:53:32 by Boschko

Updated 27 June 2021 06:46:08 by Boschko