

User guide

Cobalt Strike 4.7





blog

go

Personal blog

TOX

go

Tox for communication

FILE SHARE

go

file sharing service

PRIVATE NOTE

go

Private note service

MIRRORS

go

Mirrors of resources above

Content

Welcome to Cobalt Strike	8
Review	8
Installation and updates	9
C&C Startup Cobalt Strike Client	16
Startup	16
Distributed and command operations	18
Scripting in Cobalt Strike	20
Running the client on MacOS X	21
User interface	22
Review	22
Toolbar	23
Session and Goal Visualizations	24
Tabs	26
Consoles	27
tables	27
Hotkeys	28
Data management	29
Review	29
Goals	thirty
Services	thirty
Credentials	31
Maintenance	31
Managing Listeners and Infrastructure	32
Review	32
Listener control	32
Beacon payload Cobalt Strike	34
Staging payload	34
DNS Beacon	34
HTTP Beacon and HTTPS Beacon	38

SMB Beacon	42
TCP Beacon	44
External C2	46
Third Party Listeners	47
Consolidation of infrastructure	48
Payload security features	49
Initial Access	50
Client-side system profiler	50
Application Explorer	50
Cobalt Strike Web Services	51
Packets of user-driven attacks	51
File hosting	57
User-driven Web Drive-by attacks	57
Client side exploits	61
Website cloning	61
Spear Phishing	62
Payload artifacts and antivirus bypass	65
Artifact Kit	65
Veil Evasion framework	66
Java applet attacks	67
resource kit	68
Sleep Mask Kit	68
Post-exploitation	68
hidden beacon	68
Beacon Console	68
Beacon's menu	69
Asynchronous and Interactive Operations	70
Command execution	70
Session transfer	71
Alternative parent processes	72

Process Argument Substitution	72
DLL blocking in child processes	73
Uploading and downloading files	73
File browser	73
Windows Registry	74
Keystrokes and screenshots	75
Beacon job management	75
Process Explorer	75
Desktop management	76
Privilege Elevation	78
Mimikatz	80
Collecting credentials and hashes	81
Port scanning	81
Listing networks and hosts	82
Trusting relationship	83
Lateral movement	84
Lateral movement using the GUI	85
Other commands	86
Browser Pivoting	86
Review	86
Setting	87
Usage	88
How Browser Pivoting Works	89
Pivoting	89
What is Pivoting	89
SOCKS proxy	89
Reverse Port Forward	90
Spawn and tunneling	91
Pivot Listeners	92
Hidden VPN	93

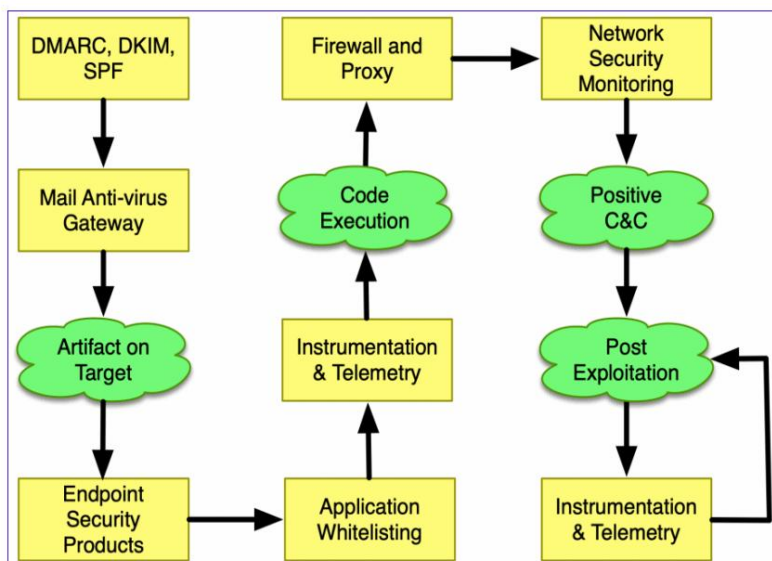
SSH Sessions.....	95
SSH client	95
Command execution	95
Uploading and downloading files	95
Peer C2	96
SOCKS Pivoting and Reverse Port Forward.....	96
Manage and control with Malleable	97
Review.....	97
Error checking	97
Profile Language.....	98
HTTP Staging.....	105
Beacon's HTTP transaction description	106
HTTP Server Configuration	107
Self-Signed SSL Certificates with SSL Beacon	108
Valid SSL Certificates with SSL Beacon	109
Profile Options	110
Developer Certificate	110
DNS Beacons	111
Malleable C2 Precautions	112
Malleable PE, Process Implementation and Post-Operation	113
Review.....	113
PE and memory indicators	113
Implementation in the process	117
Process Implementation Management.....	119
Post-operational management	122
User Defined Reflective DLL Loader	123
Beacon Object Files	127
What are the benefits of BOFs?	127
How do BOFs work?	127
What disadvantages do BOFs have?	128
How to create BOF?	128

Dynamic feature resolution	129
Aggressor Script and BOFs	129
BOF C API	131
Aggressor Script	134
What is Aggressor Script?	134
How to upload scripts	134
Script Console	135
Headless Cobalt Strike	136
A quick introduction to Sleep	136
User interaction	138
Cobalt Strike	139
Data Model	142
Listeners	143
beacon	145
SSH sessions	153
Other topics	155
Custom reports	157
Compatibility guide	159
Hooks	161
Events	176
Functions	189
Popup Hooks	334
Report-only features	335
Reports and logging	343
Logging	343
Reports	344
Custom logo in reports	350
Custom reports	351
Addition	351
Hotkeys	351
Beacon command behavior and OPSEC considerations	352
Unicode support	358

Welcome to Cobalt Strike

Cobalt Strike is a red team enemy simulation and operations platform. The product is intended for targeted attacks and imitation of post-exploitation actions of intruders. This section describes the attack process supported by the Cobalt Strike feature set. The rest of this guide discusses these features in detail.

Review



Assault problem set

A well-thought-out targeted attack begins with **reconnaissance**. Cobalt Strike's system profiler is a web-based application that displays the area for client side attacks on a target. The information gained from reconnaissance will help you understand which options are most likely to succeed against your goal.

Weaponization is the combination of a post-exploitation payload with a document or exploit to be executed on the target. Cobalt Strike contains options to turn ordinary documents into weaponized artifacts. Cobalt Strike also has the ability to export the post-operational payload, called the Beacon, to various formats for merging with artifacts outside of this toolbox. Use Cobalt Strike's spear phishing tool to **deliver** a weaponized document to

one or more people on the target network. Cobalt Strike's phishing tool turns saved emails into pixel perfect attack emails.

Control your target's network with Cobalt Strike's Beacons. This post-operational payload uses **the asynchronous "low and slow" communication pattern** that is common in advanced malware. Beacon will connect to the C&C via DNS, HTTP, or HTTPS. Beacon goes through common proxy configurations and connects to multiple hosts to resist blocking.

Experience the power of attribution and analysis of your targeted attacks with the Malleable Beacon Management and Control Language. Modify Beacon to **use network indicators that look like known malware** or merge with existing traffic. Pivot on a compromised network, discover hosts, and perform **lateral movement**

with useful Beacon automation and peer-to-peer interactions via named pipes and TCP sockets. Cobalt Strike is designed to capture trust relationships and enable lateral movement through obtained credentials, password hashes, access tokens, and Kerberos tickets.

Demonstrate meaningful business risks with Cobalt Strike's **user-exploitation** tools. Cobalt Strike's workflows make it easy to implement keystroke logging and screenshot capture tools on compromised systems. Use Browser Pivoting to access websites that your victim has logged into with Internet Explorer. This technique, developed only for Cobalt Strike, works with most sites and allows you to bypass two-factor authentication. Cobalt Strike's reporting features **will revolutionize the interaction** for your client. Provide

network administrators with an activity graph so they can find indicators of attacks on their systems. Cobalt Strike generates high-quality reports that you can present to your clients as stand-alone products or use as a complement to your written presentation.

In each of the above steps, you will need to study the target environment, its defenses, and think about how best to achieve your goals with the means at your disposal. This is evasion. Cobalt Strike is not aimed at providing evasion out of the box. Instead, the product provides flexibility, both in its potential configurations and offensive options, so you can tailor it to your environment and mission.

Installation and updates

HelpSystems LLC distributes Cobalt Strike packages as native archives for Windows, Linux and MacOS X.

Cobalt Strike uses a client/server model in which each component can be installed on one system, but is often deployed separately. The Cobalt Strike GUI is called "Cobalt Strike", "Cobalt Strike GUI" or the command used to launch the Cobalt Strike client. Cobalt Strike's server is called "Command Server" or the command used to start the C&C server. The basic installation process for Cobalt Strike involves downloading and

extracting the distribution to the operating system and running the update process to download the product.

Before starting work

Read this section before installing Cobalt Strike.

System Requirements The following components are required for any system hosting Cobalt Strike's client and/or server components.

Java

Cobalt Strike's GUI client and command and control server require one of the following Java environments:

- | Oracle Java 1.8
- | Oracle Java 11
- | OpenJDK 11. (See [Installing OpenJDK on page 10 for instructions](#))

NOTE:

If your organization does not have a commercial Java license from Oracle, we recommend that you use OpenJDK 11.

Supported Operating Systems Cobalt Strike C&C is supported on

Java-compliant Linux systems and has been tested on the following Debian-based Linux distributions (other versions may work but have not been tested):

- | Debian _
- | Ubuntu
- | Kali Linux

The Cobalt Strike client runs on the following systems:

- | Windows 7 and higher
- | MacOS X 10.13 and higher |
- | Linux GUI systems such as: Debian, Ubuntu and Kali Linux (other versions may work but have not been tested)

Hardware In addition to a valid operating system, the following minimum requirements must be met:

- | Processor 2GHz+ | 2GB
- | RAM |
- | Available disk space 500MB+

On Amazon EC2, use a High-CPU Medium instance (c1.medium, 1.7 GB).

Linux glibc Be

aware that some Linux distributions may not have or may not have the correct version of glibc. If you encounter this issue, please see [glibc Missing From Older Linux Distributions](#). on the HelpSystems portal.

Installing OpenJDK Cobalt Strike

has been tested with OpenJDK 11 and its launchers are compatible with a properly installed OpenJDK 11 environment.

Linux (Kali 2018.4, Ubuntu 18.04)

1. Update APT:
`sudo apt-get update`
2. Install OpenJDK 11 using APT:

```
sudo apt-get install openjdk-11-jdk
```

3. Install OpenJDK 11 by default: `sudo update-java-alternatives -s java-1.11.0-openjdk-amd64`

Linux (Other) 1.

Remove the current OpenJDK package(s). 2.

Download OpenJDK for Linux/x64 from: <https://jdk.java.net/archive/>. 3. Unzip the OpenJDK executable: `tar zxvf openjdk-11.0.1_linux-x64_bin.tar.gz`

4. Move the OpenJDK folder to **/usr/local**:

```
mv jdk-11.0.1 /usr/local
```

5. Add the following to **~/.bashrc**:

```
JAVA_HOME="/usr/local/jdk-11.0.1"
PATH=$PATH:$JAVA_HOME/bin
```

6. Update **~/.bashrc** source , the new environment variables to take effect:
`~/.bashrc`

macOS X

1. Download OpenJDK for macOS/x64 from: <https://jdk.java.net/archive/>. 2. Open the

Terminal and navigate to **the Downloads/** folder . 3. Unzip the

archive: `tar zxvf`

```
openjdk-11.0.1_osx-x64_bin.tar.gz
```

4. Move the extracted archive to the **/Library/Java/JavaVirtualMachines/** folder: `sudo mv jdk-11.0.1.jdk/ /Library/Java/JavaVirtualMachines/`

The java command on MacOS X will use the highest Java version in /Library/Java by default.

TIP: If you

see a **JRELoadError** message, this is because the JavaAppLauncher stub included with Cobalt Strike is loading a library from the given path to launch the JVM inside the stub process. To resolve this error, run the following command: `sudo ln -fs /Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk /Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin`

Replace **jdk-11.0.2.jdk** with your Java path. The next release of Cobalt Strike will use a more flexible Java application stub for MacOS X.

Windows

1. Download OpenJDK for Windows/x64 from: <https://jdk.java.net/archive/>. 2. Unzip the archive to **c:\program files\jdk-11.0.1**.

3. Add `c:\program files\jdk-11.0.\bin` to PATH environment variable

your user:

- a. Go to **Control Panel -> System -> Advanced System settings -> Environment Variables...**
- b. Highlight **Path** in **User Variables**.
- c. Click **Create**. d.

Enter: `c:\program files\jdk-11.0.1\bin`. e. Click **OK**

on all dialog boxes.

Wayland Desktop - not supported

[Wayland](#) is a modern replacement for the X Windows System. Wayland has made significant progress as a project, and some desktops use it as their default windowing system. However, don't let this statement fool you. Not all applications or software environments work 100% perfectly on Wayland. There are still bugs and issues to be resolved.

There are bugs in Java (or Wayland) that can cause a graphical Java application to crash in normal use when it is launched on the Wayland desktop. These bugs are for Cobalt Strike users. **HelpSystems does not support the use of Cobalt Strike on Wayland desktops.**

Do I use Wayland? Type echo

`$XDG_SESSION_TYPE` to see if you are using wayland or x11.

How to disable Wayland in Kali Linux Latest

version of Kali Linux 2017 Rolling uses desktop by default wayland. To change it back to X11: 1. Open the /

`etc/gdm3/daemon.conf` file with your favorite text editor.

2. Locate the **[daemon]** section.

3. Add **WaylandEnable=false** and reboot your system.

Installing Cobalt Strike

Follow these instructions to install Cobalt Strike.

NOTE:

The Cobalt Strike **distribution** (steps 1 and 3) contains the OS-specific Cobalt Strike launcher, support files, and an updater. It does not contain Cobalt Strike itself.

Running the updater (step 4) downloads the product Cobalt Strike and completes the final installation steps.

1. Download the Cobalt Strike distribution for a supported operating system.
(Download link will be emailed to you)

2. Install the recommended Java environment. (For instructions, see [Installation OpenJDK on page 10](#)).

3. Extract, mount or unzip the distribution. Do one of the following, depending on your operating system:
 - a. For Linux: i.
Unzip the file **cobaltstrike-dist.tgz**: `tar zxvf cobaltstrike-dist.tgz`
 - b. For macOS X:
 - i. Double click on the **cobaltstrike-dist.dmg** file to mount his.
 - ii. Drag the **Cobalt Strike** folder into the **Applications folder**.
 - c. For Windows:
 - i. Disable your antivirus before installing Cobalt Strike. ii. Use the zip tool of your choice to extract the file **cobaltstike.zip** to the installation location.
4. Run the updater to complete the installation. Depending on the opera system, do one of the following:
 - a. For Linux:
 - i. Enter the following commands:
`cd /path/to/cobaltstrike ./update`
 - b. For macOS X:
 - i. Go to the **Cobalt Strike folder**. ii.
Double click on **Update Cobalt Strike.command**.
 - c. For Windows:
 - i. Go to the **Cobalt Strike folder**. ii.
Double click on **update.bat**.

Make sure you update both the command and control server and the client software with the license key. Cobalt Strike is usually licensed for one user. The C&C does not require a separate license.

License authorization files

A valid authorization file is required to run the licensed version of Cobalt Strike. The authorization file is an encrypted file that contains information about your Cobalt Strike license. This information includes: your license key, license expiration date, and the ID associated with your license key.

How to get authorization file?

The built-in updater requests an authorization file from the Cobalt Strike update server on startup. The updater downloads the new authorization file even if your version of Cobalt Strike is up to date. This allows the authorization file to match the license expiration dates in the HelpSystems records.

What happens if my license expires?

Cobalt Strike will refuse to launch when the authorization file expires. If the authorization file expires while Cobalt Strike is running, it will not affect its operation in any way. The licensed Cobalt Strike only checks authorization files on startup.

When does my authorization file expire? Your authorization file will

expire when your Cobalt Strike license expires. When renewing your Cobalt Strike license, run **the built-in updater** to update the authorization file with the latest information.

Go to **Help -> System Information** to find out when your authorization file expires. Look for the "valid to" option in the **Other section**. Keep in mind that client information and C&C information can have different meanings (depending on which license key was used and when the authorization file was last updated).

Cobalt Strike will also warn you if the authorization file expires after 30 days.

How to add an authorization file to the sandbox? The authorization file is **cobaltstrike.auth**. The updater

always places this file along with cobaltstrike.jar. To use Cobalt Strike in a closed environment:

1. Download Cobalt Strike Trial Pack from <https://www.cobaltstrike.com/download>.
2. Update the Cobalt Strike trial package from an Internet connected system.
3. Copy the contents of the updated **cobaltstrike/** folder to your environment. The most important files are cobaltstrike.jar and cobaltstrike.auth.

Does Cobalt Strike have the ability to contact HelpSystems?

Cobalt Strike

does not "call home" to HelpSystems after the update process. The authorization file is generated during the update process.

How to use old version of Cobalt Strike with updated authorization file?

Cobalt Strike 3.8 and below does not check or require an authorization file.

Cobalt Strike 3.9 and later checks for the presence of the cobaltstrike.auth file located along with the cobaltstrike.jar file. Update Cobalt Strike from another folder and copy the new cobaltstrike.auth file to the folder containing the old version of Cobalt Strike. The authorization file is not tied to a specific product version.

What is meant by user ID? The user ID is a 4-byte number associated with the license key

Cobalt Strike. Cobalt Strike 3.9 and later embed this information into stagers and stages generated by Cobalt Strike.

How to find user id value in Cobalt Strike artifact? The user id value is the last 4 bytes of the stager payload in Cobalt Strike 3.9 and later.

This screenshot is an HTTP stager from a trial version. The trial version has a user id of 0. The last 4 bytes of this stager (0x0, 0x0, 0x0, 0x0) reflect this.

```

00000220 2d 54 45 53 54 2d 46 49 4c 45 21 24 48 2b 48 2a | -TEST-FILE!$H+H*
00000230 00 35 4f 21 50 25 40 41 50 5b 34 5c 50 5a 58 35 | .50!P%@AP[4\PZX5
00000240 34 28 50 5e 29 37 43 43 29 37 7d 24 45 49 43 41 | 4(P^)7CC)7}$EICA
00000250 52 2d 53 54 41 4e 44 41 52 44 2d 41 4e 54 49 56 | R-STANDARD-ANTIV
00000260 49 52 55 53 2d 54 45 53 54 2d 46 49 4c 45 21 24 | IRUS-TEST-FILE!$
00000270 48 2b 48 2a 00 35 4f 21 50 25 40 41 50 5b 34 5c | H+H*.50!P%@AP[4\
00000280 50 5a 58 35 34 28 50 5e 29 37 43 43 29 37 7d 24 | PZX54(P^)7CC)7}$
00000290 45 49 43 41 52 2d 53 54 41 4e 44 41 52 44 2d 41 | EICAR-STANDARD-A
000002a0 4e 54 49 56 49 52 55 53 2d 54 45 53 54 2d 46 49 | NTIVIRUS-TEST-FI
000002b0 4c 45 21 24 48 2b 48 2a 00 35 4f 21 50 25 40 41 | LE!$H+H*.50!P%@A
000002c0 50 5b 00 68 f0 b5 a2 56 ff d5 6a 40 68 00 10 00 | P[.h...V...j@h...
000002d0 00 68 00 00 40 00 57 68 58 a4 53 e5 ff d5 93 b9 | .h...@.WhX.S....
000002e0 00 00 00 00 01 d9 51 53 89 e7 57 68 00 20 00 00 | .....QS..Wh. ..
000002f0 53 56 68 12 96 89 e2 ff d5 85 c0 74 c6 8b 07 01 | SVh.....t....
00000300 c3 85 c0 75 e5 58 c3 e8 a9 fd ff ff 31 37 32 2e | ...u.X.....172.
00000310 31 36 2e 34 2e 31 33 34 00 00 00 00 00 00 00 | 16.4.134.....
0000031d

```

HTTP Payload Stager (trial version of Cobalt Strike)

The user id value also exists in the stage payload, but is harder to retrieve. Cobalt Strike does not use the Client ID value in network traffic or other parts of the tool.

How to protect different red team infrastructure from cross-identification with this id?

If you have a **unique** authorization file on each command server, then each command server and the artifacts that come from it will have a different ID. Cobalt Strike's update server generates a new

authorization file each time the update program is launched. Each authorization file has a unique identifier. Cobalt Strike distributes C&C ID only. It does not propagate the user ID from the GUI or headless client authorization file.

After you are done

Congratulations! Cobalt Strike is now installed. For more information and further steps, read the following:

Next Steps *Starting the C&C*

Server on page 16 Starting the Cobalt Strike Client on page 16

C&C launch

Cobalt Strike consists of a client and server component. The server, called the command and control server, is the dispatcher for the payload and the host for Cobalt Strike's social engineering functions. The C&C also stores the data collected by Cobalt Strike and manages the logging. Cobalt Strike C&C must run on a

supported Linux system. To start the Cobalt Strike C&C server, run the following command to run the script included in the Cobalt Strike Linux package:

```
root@kali:~/cobaltstrike# ./teamserver 192.168.1.4 password
[*] Generating X509 certificate and keystore (for SSL)
[+] Team server is up on 50050
[*] SHA1 hash of SSL cert is: 1d1edf9c258f3eca9534d5c911e23002f0b5a7e5
Offset is: 27006
[+] Listener: local - beacon http (windows/beacon_http/reverse_http) on port 80 started!
```

Figure 3. Launching the command and control server

`./teamserver <ip_address> <password> [<malleableC2profile> <destruction_date>]` The command server script uses the following two required and two optional parameters:

IP Address - (required) Enter the C&C server's externally accessible IP address. Cobalt Strike uses this value as the default host for its functions.

Password - (required) Enter the password that your team members will use to connect the Cobalt Strike client to the C&C.

Malleable C2 Profile - (Optional) Specify a valid Malleable C2 profile. For more information about this feature, see [*Malleable Command and Control on page 97*](#)

Kill Date - (Optional) Enter a date value in YYYY-MM-DD format. The C&C will insert this kill date into every Beacon stage it generates. Payload will refuse to run on or after this date, and will shut down if it works on or after this date.

When the C&C server starts, it publishes the SHA256 hash of the C&C SSL certificate. Distribute this hash to your team members. When your team members connect, their Cobalt Strike client will ask if they know this hash before authenticating to the C&C. This serves as an important defense against man-in-the-middle attacks.

Launching the Cobalt Strike client

Complete the following steps to connect the Cobalt Strike client to the C&C.

Steps

1. To launch the Cobalt Strike client, use the launcher, included with your platform package.
 - a. For Linux:
 - i. Enter the following commands: `./cobaltstrike`

- b. For MacOS X:
 - i. Go to the **Cobalt Strike folder**. ii.
 - Double click on **cobaltstrike**.
- c. For Windows:
 - i. Go to the **Cobalt Strike folder**. ii.
 - Double click **cobaltstrike.exe**.

The Connect dialog box will be displayed.

Connect Cobalt Strike Dialog Box

2. Cobalt Strike keeps track of the C&C servers you connect to and remembers information about them. Select one of the C&C profiles on the left side of the Connect dialog box to populate the box with its information. Use the **Alias Names** and **Host Names** buttons to toggle the display of the list of hosts. Active connections will be displayed in blue. You can control the initial display of the list of hosts, the color of the text of active connections and edit the list through **Cobalt Strike -> Preferences -> Team Servers**.

Parameters:

Alias - Enter an alias for the host, or use the default value. The alias cannot be empty, start with the same character as the active connection or use connection.

Host - Specify the address of your C&C server in the Host field. The hostname cannot be empty.

Port - Displays the default port for the C&C server (50050). It rarely changes. The port cannot be empty and must be numeric.

User - The User field is your C&C nickname. Change it to yours call sign, or made-up hacker name. The username cannot be empty.

Password - Enter the common password for the command and control server.

3. Click **Connect** to connect to Cobalt Strike's C&C server. If this is your first time connecting to this C&C server, Cobalt Strike will ask if you know the C&C SHA256 hash.



Checking the server's SSL certificate

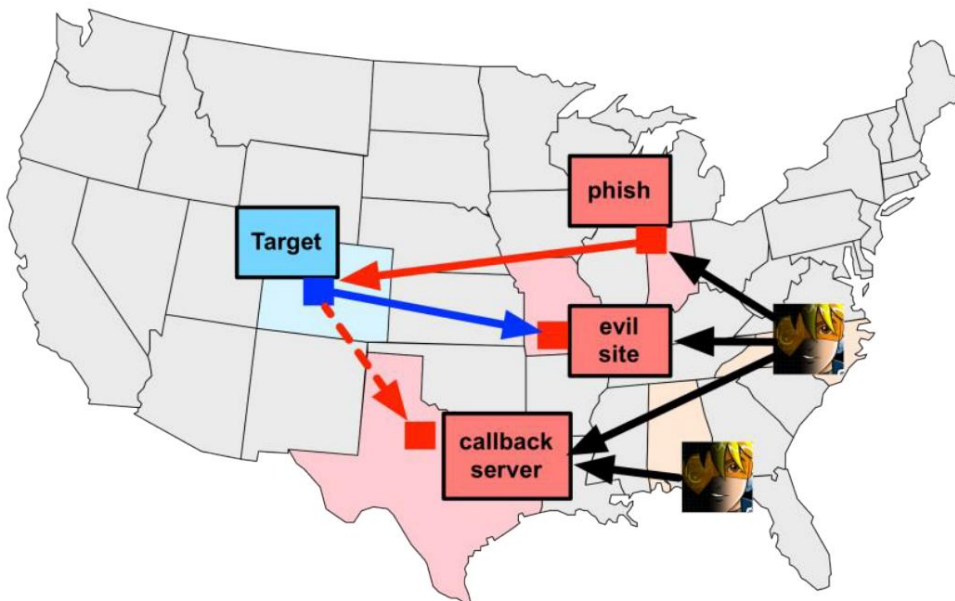
4. If you did, click **Yes** and the Cobalt Strike client will connect to the server and open the user interface.

NOTE:

Cobalt Strike will also remember this SHA256 hash for future connections. You can manage these hashes through **Cobalt Strike -> Preferences -> Fingerprints**.

Distributed and command operations

Use Cobalt Strike to coordinate red team distributed operations. Install Cobalt Strike on one or more remote hosts. Start command and control servers and ask team members to connect to them.



Distributed Operations with Cobalt Strike

After connecting to the command server, your command will be:

I Use the same sessions I Share hosts,
captured data, and downloaded files. I Communicate through the general event log.

The Cobalt Strike client can connect to multiple C&C servers. Go to **Cobalt Strike -> New Connection** to initiate a new connection. When connecting to multiple servers, a switch bar will appear at the bottom of the Cobalt Strike window.



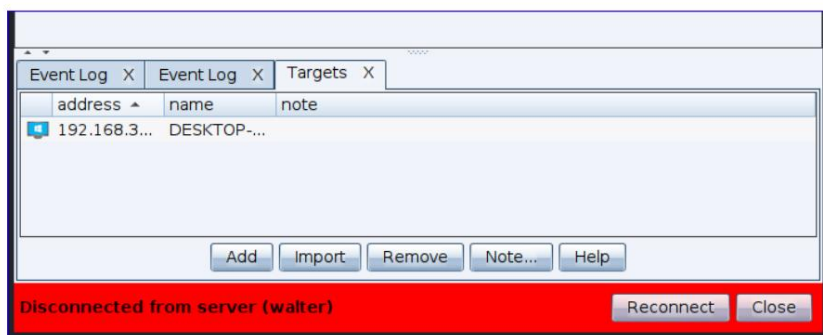
Server Switch Panel

This switch panel allows you to switch between active Cobalt Strike server instances. Each server has its own button. Right-click on it and select **Rename** so that the text of the button reflects the role of the server during your interaction with it. The server button will display the active button in bold and color based on the color preferences specified in **Cobalt Strike -> Preferences -> TeamServers** to better show which button is active. This button name will also identify the server in the Cobalt Strike activity report.

When connected to multiple servers, Cobalt Strike merges Listeners from all servers to which it is connected. This bundling allows you to send a phishing email from one server that points to a malicious website located on another server. At the end of your interaction, Cobalt Strike's reporting feature will query all the servers you are connected to and merge the data to create a single report.

Client reconnection

When a client disconnect is initiated by the user through the menu, toolbar, or switch server panel, a red banner is displayed with a **Reconnect** and **Close** button.



Click **Close** to close the window. Click **Reconnect** to reconnect to the C&C.

If the C&C server is unavailable, a dialog box will appear asking if you want to try again (Yes/No). If **Yes**, the connection will be attempted again (retry if necessary). If **No**, the window will close. When a disconnect is initiated by a C&C or other network interruption, a message appears on a red banner with a countdown to reconnect.

This will repeat until a connection to the command and control server is established or until the user clicks the **Close button**. In this case, the user can interact with other parts of the user interface. When the client connects, the red reconnect bar will disappear.

Scripting in Cobalt Strike

Cobalt Strike can be customized using the Aggressor Script language. Aggressor Script allows you to modify and extend the Cobalt Strike client.

Story

Aggressor Script is the spiritual successor to Cortana, the open source scripting engine in Armitage. Cortana appeared thanks to a contract under the DARPA Cyber Fast Track program. Cortana allows users to extend Armitage and manage the Metasploit® framework and its features through Armitage's command and control server. Cobalt Strike 3.0 is a completely redesigned version of Cobalt Strike without using Armitage as a base. This change made it possible to revisit Cobalt Strike's scripts and create something based on its functionality. The result of this work was Aggressor Script.

Aggressor Script is a scripting language for red team operations and enemy simulation inspired by IRC scripting clients and bots. Its purpose is twofold. You can create long-running bots that mimic virtual team members attacking side by side with you. You can also use it to extend and modify the Cobalt Strike client to suit your needs.

Loading scripts

Aggressor Script is built into the Cobalt Strike client. To manage scripts, go to **Cobalt Strike -> Script Manager** and click **Load**.



Scenario manager

The standard script inside Cobalt Strike defines all popup menus and formats the information displayed in Cobalt Strike's consoles. With Aggressor Script's mechanism, you can change these default settings and tailor Cobalt Strike to suit your preferences.

You can also use Aggressor Script to add new features to Beacon and to automate some tasks.

To learn more about scripting, see [Aggressor Script on page 134](#).

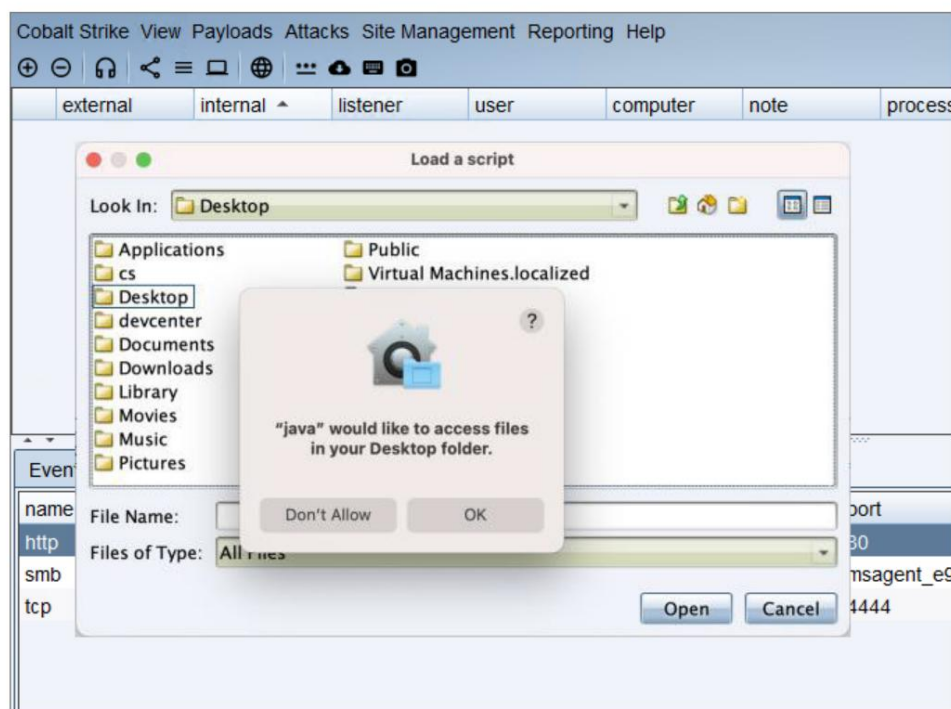
Running the client on MacOS X

The Cobalt Strike client may initially not display the contents of the Documents, Desktop and Downloads folders in the file browser. (for example, when loading scripts, downloading files, generating payloads, etc...) By

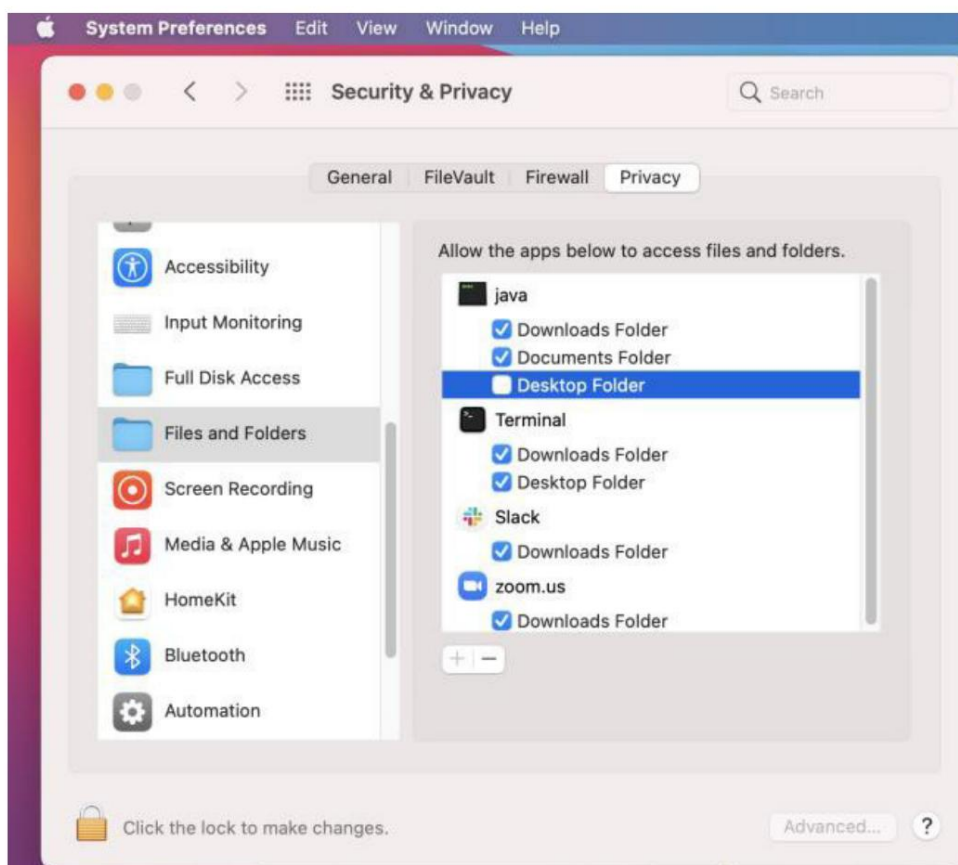
default, OSX restricts applications from accessing the Documents, Desktop, and Download folders. Applications need to explicitly grant access to these folders.

Since Cobalt Strike is a third party application, it will not be possible to simply grant access to the "Cobalt Strike" application. You may need to give the JRE running the Cobalt Strike client access to the file system. You can grant access to specific files and folders or full disk access.

You may be prompted to grant access to:



Or, if access was previously denied, you may need to change access in the OSX System Preferences / Security & Privacy / Privacy dialog:



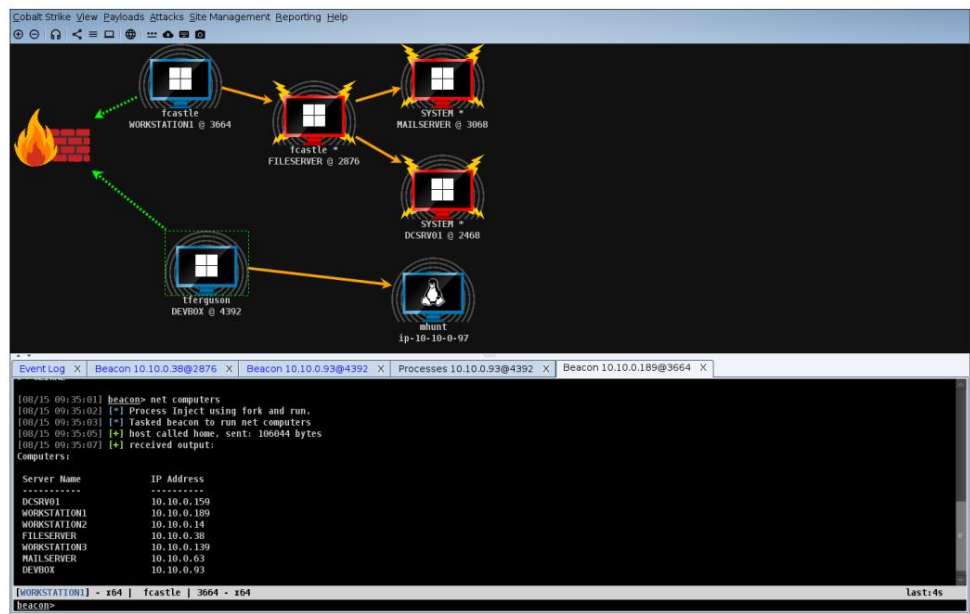
Keep in mind that other applications using the JRE will also have this access.

NOTE:

The same steps may need to be taken for '/bin/bash'.

User Interface Overview











The user interface of Cobalt Strike is divided into two parts. At the top of the interface, a visual representation of sessions or targets is displayed. The bottom of the interface displays tabs for each Cobalt Strike feature or session you are interacting with. You can click on the area between these two parts and resize them to your liking.



Cobalt Strike user interface

Toolbar

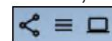
The toolbar at the top of Cobalt Strike provides quick access to common Cobalt Strike functions. Knowing the buttons on the toolbar will greatly speed up your work with Cobalt Strike.

	Connect to another C&C server
	Disconnect from current C&C
	Create and edit Listeners
	Show sessions in graph mode
	Show Sessions in Datasheet View
	Show Goals in Datasheet View
	Manage web server
	Show Credentials
	Show downloads
	Show Keystrokes

[Show screenshots](#)

Session and Goal Visualizations

Cobalt Strike has a number of renderers, each designed to help with a different aspect of your work. You can switch between visualizations using the buttons (Pivot Graph, Sessions Table, Targets Table) of the tools or the **Cobalt Strike -> Visualization menu**.



on the panel

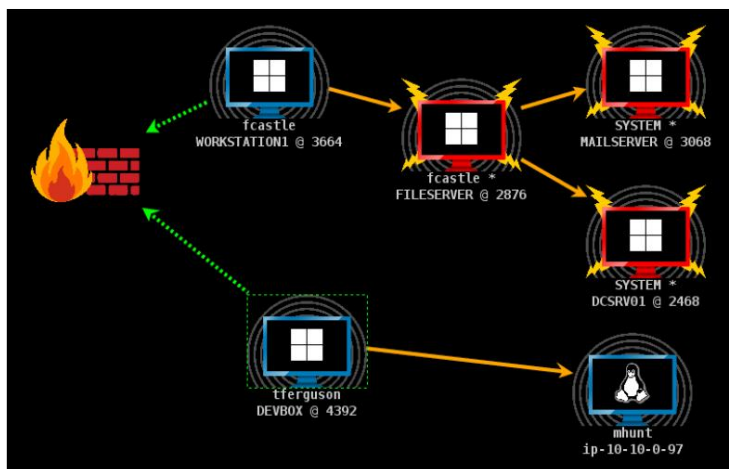
Pivot graph

Cobalt Strike has the ability to chain multiple Beacons. These linked Beacons receive their commands and send the results through the parent Beacon in the chain. This type of chain is useful for controlling which sessions exit the network and for imitating a legitimate user restricting communication paths within the network. This chain is one of the most powerful features of Cobalt Strike.

Cobalt Strike's workflows make chaining like this very easy. It's not uncommon for Cobalt Strike operators to line up Beacon chains four or five levels deep. Without visualization, it is very difficult to track and understand these chains. This is where the Pivot Graph comes in.

Pivot graph displays your chains of beacons in a visual way. Each Beacon session has an icon. As in the session table, the icon for each host indicates its operating system. If the icon is red with lightning bolts, then Beacon is running in a process with administrator privileges. A darker icon means that the Beacon's session was asked to end and he acknowledged it.

The firewall icon indicates the Beacon's exit point. **The green dotted line** indicates that the Beacon is using HTTP or HTTPS connections to exit the network. **The yellow dotted line** indicates that the DNS protocol is being used to leave the network.



[View Cobalt Strike Graph](#)

An arrow connecting one Beacon session to another represents the link between the two Beacons. Beacon uses Windows named pipes and TCP sockets to manage Beacons through a peer-to-peer connection.

The **orange arrow** is a named pipe (PIPE). SSH sessions also use an orange arrow. The **blue arrow** is the TCP socket channel. A **red** (named pipe) or **purple** (TCP) arrow indicates a disconnected connection. Click on a Beacon to select it. You can

select multiple Beacons by clicking and dragging the box above the desired hosts. Press Ctrl and Shift to select from an individual Beacon. , to select or remove

Right-click on a Beacon to bring up a menu with available options for post-exploitation.

Several **hotkeys are available in the Pivot graph.** I

Ctrl+Plus - Zoom in | **Ctrl+Minus** -

Zoom out | **Ctrl+0** - Reset zoom level

| **Ctrl+A** - Select all hosts | **Escape** - Clear selection

| **Ctrl+C** - Arrange hosts in a

circle | **Ctrl+S** - Arrange

stacked hosts | **Ctrl+H** - Arrange hosts in

a hierarchy

Right-click on a Pivot graph with no Beacons selected to adjust the position of that graph. This menu also has an option for unrelated sessions. Select **Hide** to hide unrelated sessions in the Pivot graph. Select **Show** to again by start unrelated sessions.

Session table

The session table shows which Beacons are accessing this instance of Cobalt Strike. Beacon is Cobalt Strike's payload for intruder simulation. Here you will see the external and internal IP address of each Beacon, its Listener, the time of the last connection to the server and other information. Next to each line is an icon indicating the operating system of the compromised target. If the icon is red with lightning bolts, Beacon is running in a process with administrator privileges. A darker icon means that the Beacon's session was asked to end and he acknowledged it.

external	internal	listener	user	computer	note	process	pid	arch	last
10.10.10.7	10.10.10.3	local - http	SYSTEM *	DC		rundll32.exe	2300	x64	30m
10.10.10.191	10.10.10.7	local - http	SYSTEM *	FILESERVER		rundll32.exe	2604	x64	6s
10.0.0.147	10.10.10.191	local - http	jim.stevens	WS1		jusched.exe	4844	x86	726ms
10.10.10.198	10.10.10.198	local - dns	SYSTEM *	DEVELOPERWS		rundll32.exe	3100	x86	6s
10.10.10.198	10.10.10.198	local - dns	Jamie.Grins	DEVELOPERWS		SecurityHealthSy...	5532	x86	12s
10.10.10.198	192.168.57.18	local - dns	jgrins	ubuntu					9m
10.10.10.7	192.168.58.3	local - http	SYSTEM *	POWERDC		rundll32.exe	3956	x64	3m
192.168.58.3	192.168.58.35		SYSTEM *	ENGINEER		rundll32.exe	1512	x86	3m

Beacon management tool

If you are using a DNS Listener, be aware that Cobalt Strike will not know anything about a host until it contacts it for the first time. If you see an entry with the time of the last call and nothing else, you need to give this Beacon the first task to see more information. Right click on one or more Beacons to see options for post-exploitation.

Goal table

The target table displays the targets in the Cobalt Strike's data model. The target table displays each target's IP address, its NetBIOS name, and a note that you or one of your team members assigned. The icon to the left of the target indicates its operating system. A red lightning icon indicates that a Beacon session is associated with the target.

address	name	note
10.10.10.1		
10.10.10.3	DC	domain controller for CORP
10.10.10.5	MAIL	
10.10.10.7	FILESERVER	
10.10.10.21		
10.10.10.50		
10.10.10.190	WS2	
10.10.10.191	WS1	
10.10.10.198	DEVELOPERWS	developer's system
192.168.57.18	ubuntu	
192.168.57.240	DEVELOPERWS	
192.168.58.3	POWERDC	
192.168.58.35	ENGINEER	SCADA HMI

View Cobalt Strike targets

Click on any table header to sort the hosts. Highlight a line and right-click on it to bring up a menu with options for that host. Press Ctrl and Alt to select or deselect individual hosts.

The target table is useful for lateral movement and understanding your target's network.

Tabs

Cobalt Strike opens every dialog, console and table in a tab. Press the X button to close the tab. Use **Ctrl+D** to close the active tab.

Ctrl+Shift+D will close all tabs except the active

one. Right-click X to open a tab in a window, take a screenshot of a tab, or close all tabs with the same name.

There are also hotkeys for these functions. Use **Ctrl+W** to open the active tab in its own window. Use **Ctrl+T** to quickly save a screenshot of the active tab.

Ctrl+B will send the current tab to the bottom of Cobalt Strike's window. This is useful for tabs that need to be constantly monitored. **Ctrl+E** will cancel this action and remove the tab at the bottom of the Cobalt Strike window. Hold

shift and press X to close all tabs with the same name. Hold shift + control and press X to open a tab in its own window. Use the **Ctrl+Left** and **Ctrl+Right** keys to quickly switch tabs. You can drag and drop tabs to change their order.

TIP: A

complete list of [default keyboard shortcuts](#) is available in the menu (**Help -> Default Keyboard Shortcuts**).

Consoles

Cobalt Strike provides a console to interact with Beacon sessions, scripts, and chat with teammates.

```

Event Log X Credentials X Beacon 192.168.58.20@2948 X Beacon 192.168.57.8@120 X Beacon 10.10.10.189@3344 X
192.168.58.20:139
192.168.58.20:135

[+] received output:
192.168.58.1:80
192.168.58.1:22 (SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7.1)
192.168.58.20:445 (platform: 500 version: 6.1 name: BILLING-POWER domain: CORP)
Scanner module is complete

[-] lost link to parent beacon: 10.10.10.4
[-] lost link to parent beacon: 10.10.10.4
[+] established link to parent beacon: 10.10.10.4
beacon> pwd
[*] Tasked beacon to print working directory
[+] host called home, sent: 8 bytes
[*] Current directory is C:\Windows\system32

[BILLING-POWER] SYSTEM */2948 last: 39s
beacon>

```

Console Tab

Consoles keep track of command history. Use **the up arrow** to navigate to previously typed commands. **The down arrow** allows you to return to the last command you typed.

The history command lists previously typed commands. Team ! allows you to re-execute previously typed commands.

NOTE:

The list of previously typed commands is not saved between different sessions. If you close the console window, then the next time you open it, it will start without previously typed commands.

Use the **Tab** key to complete commands and options.

Use **Ctrl+Plus** to increase the console font size, **Ctrl+Minus** to decrease it, and **Ctrl+0** to reset it. This change is local to the current console only. To change the font permanently go to **Cobalt Strike -> Preferences**.

Press **Ctrl+F** to display a panel that allows you to search for text in the console. Use **Ctrl+A** to select all text in the console buffer.

TIP: A

complete list of [default keyboard shortcuts](#) is available in the menu (**Help -> Default Keyboard Shortcuts**).

tables

Cobalt Strike uses tables to display sessions, credentials, targets, and other interaction information.

Most tables in Cobalt Strike have the ability to assign color highlighting to selected rows. These secretions are visible to other Cobalt Strike clients. Right click and find the **Color menu**. Press **Ctrl+F** on a table to display the

table search bar. This function allows you to filter the current table.

address ^	name	note
172.16.20.3	DC	
172.16.20.80	GRANITE	
172.16.20.81	COPPER	
172.16.20.128	metasploitable	
172.16.20.143	MARBLE	
172.16.20.163	QUARTZ	

Filter: ! 172.16.20.0/24 address 1 filter applied. Reset X

Add Import Remove Note... Help

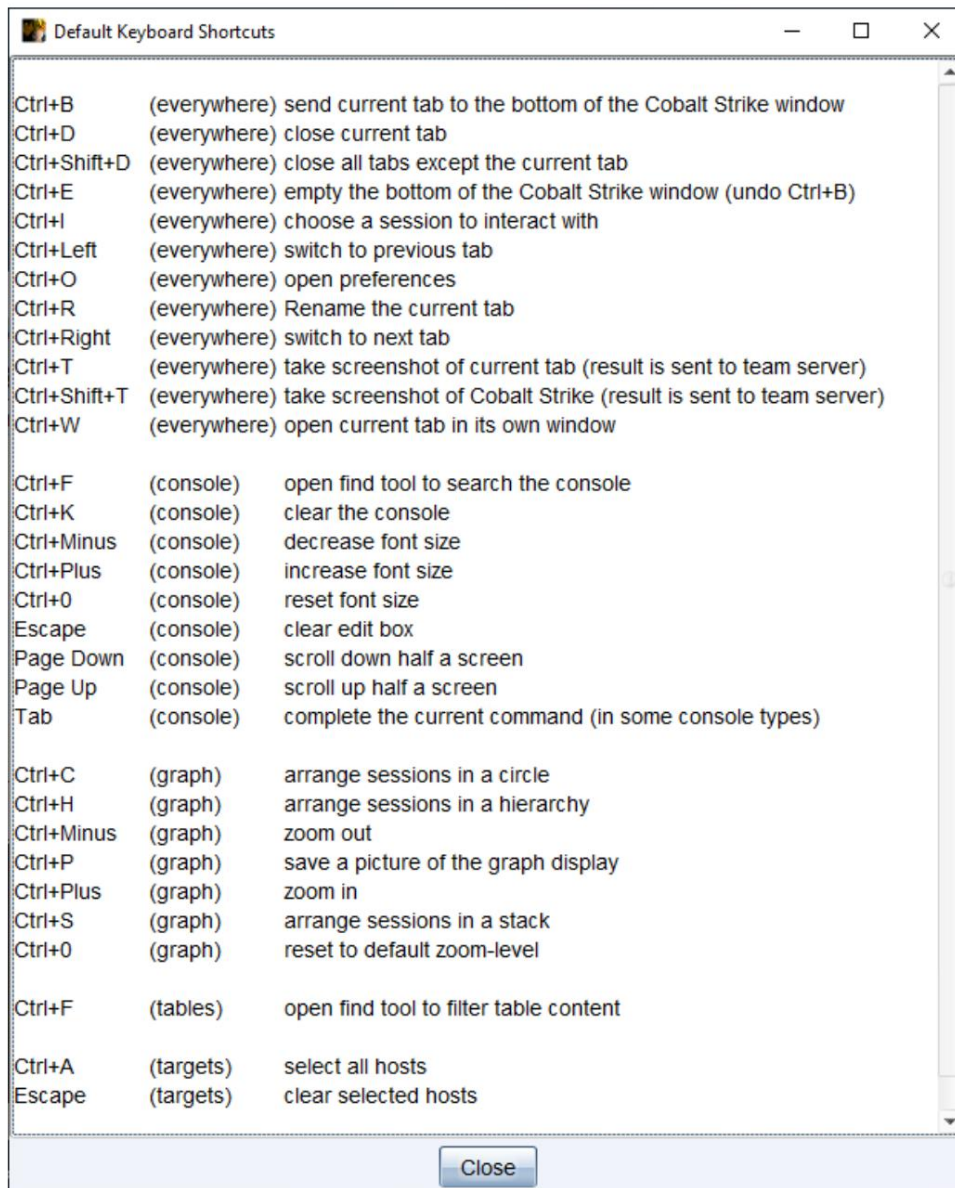
Table with search bar

Enter the filtering criteria in the text box. The format of the criteria depends on the column you choose to apply the filter to. Use CIDR notation (for example, 192.168.1.0/24) and host ranges (192.168.1-192.169.200) to filter columns containing addresses. Use numbers or ranges of numbers for columns containing numbers. Use wildcard characters (*, ?) to filter columns that contain strings.

button ! cancels the current criteria. Press **Enter** to apply the specified criteria to this table. You can combine as many criteria as you like. The **Reset** button will remove the filters applied to the current table.

Hotkeys

When working in the user interface, there are many default hotkeys available to you. Some of them can be used anywhere, others are designed for different areas of the user interface. Selecting **Help -> Default Keyboard Shortcuts** from the menu will open the following help window:



The Aggressor function [openDefaultShortcutsDialog](#), can also be used to open this list.

Data Management

Overview

Cobalt Strike C&C is a broker for information collected by Cobalt Strike while you are working. Cobalt Strike analyzes data received from Beacon to extract targets, services, and credentials. If you

want to export Cobalt Strike data, you can do so via **Reporting -> Export Data**. Cobalt Strike provides the ability to export data to TSV and XML files.

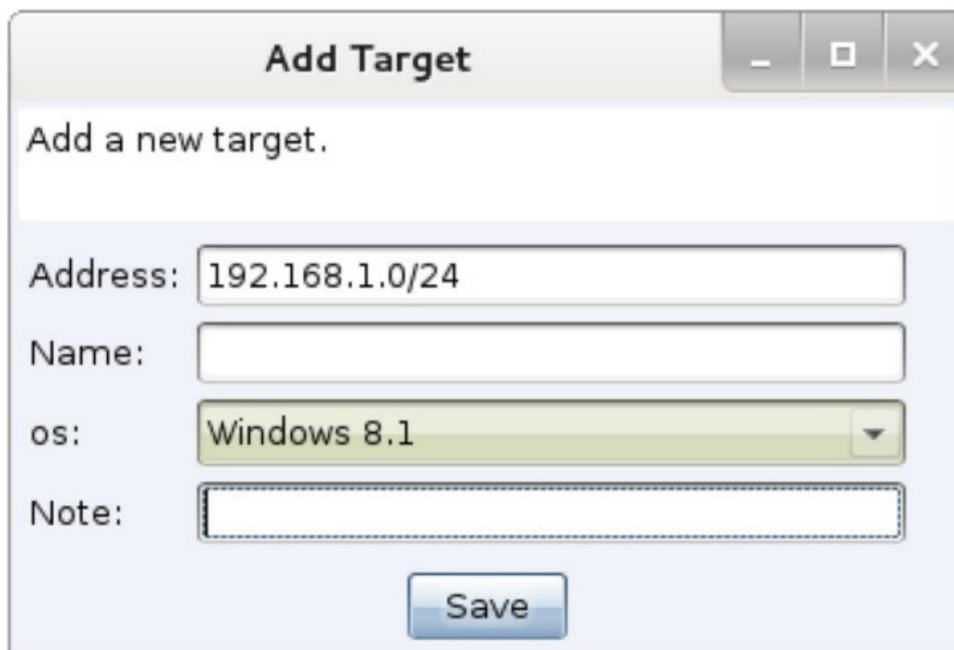
Cobalt Strike's client data export feature aggregates data from all C&C servers you are currently connected to and exports it to TSV and XML files with data from Cobalt Strike's data model.

Goals

You can interact with Cobalt Strike's target information via **View -> Targets**. This tab displays the same information as the goal visualization.

Click **Import** to import the target information file. Cobalt Strike accepts flat text files with one host per line. He also accepts XML files generated by Nmap (option -oX).

Click **Add** to add new targets to the Cobalt Strike data model.



Adding a target

This dialog allows you to add multiple hosts to the Cobalt Strike database. Specify a range of IP addresses, or use CIDR notation in the Address field to add multiple hosts at once. Hold down the shift key while clicking the Save button to add hosts to the data model and keep this dialog box open.

Select one or more hosts and right-click on them to bring up their menu. In this menu, you can change the description of hosts, set information about their operating system, or remove hosts from the data model.

Services

In the list of targets, right-click on the host and select **Services**. The Cobalt Strike Service Browser will open. Here you can view services, assign notes to them, and delete service entries.

address	port	banner	note
10.10.10.0	22	SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7.1	
10.10.10.21	22	SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7	
10.10.10.5	25	220 ACME Corporation Mail Server [hMailServer]	
10.10.10.1	53		
10.10.10.3	53		
10.10.10.0	80		
10.10.10.21	80		
10.10.10.1	81		
10.10.10.3	88		
10.10.10.5	110		
10.10.10.3	135		
10.10.10.4	135		

Services Dialog

Credentials

Go to **View -> Credentials** to interact with Cobalt Strike's credential model.

Click **Add** to add an entry to the credential model. Also, you can hold shift and hit **Save** to keep the window open and make it easier to add new credentials to the model.

Click **Copy** to copy the selected entries to the clipboard. Use **Export** to export credentials in PWDump format.

user	password	realm	note	source	host
Guest	31d6cfe0d16ae...	FILESERVER		hashdump	10.10.10.4
SUPPORT_3889...	5ace382672979...	FILESERVER		hashdump	10.10.10.4
Administrator	4d714387627d0...	FILESERVER		hashdump	10.10.10.4

Credential Model

Service

Cobalt Strike's data model stores its state and state metadata in the **data/ folder**. This folder is located in the folder from which you launched the Cobalt Strike C&C .

To clean up Cobalt Strike's data model: stop the C&C, delete the **data/ folder** and its contents. Cobalt Strike will re-create **the data/ folder** the next time the C&C is started.

If you want to archive the data model, stop the C&C and use your favorite program to save the **data/ folder** and its files somewhere else. To restore the data model, stop the C&C and restore the old content in the **data/ folder**.

Reporting -> Reset Data resets Cobalt Strike's data model without restarting the C& C.

Listener and infrastructure management

Review

The first step in any interaction is the creation of infrastructure. In the case of Cobalt Strike, the infrastructure consists of one or more command and control servers, redirectors, and DNS records that point to your command and control servers and redirectors. Once the C&C is up and running, you'll want to connect to it and configure it to accept connections from compromised systems. Listeners are Cobalt Strike's mechanism for doing this.

The Listener is both configuration information for the payload and an instruction for Cobalt Strike to create a server to accept connections from that payload.

The Listener consists of a user-defined name, a payload type, and a few payload-specific options.

Listener control

To manage Cobalt Strike's Listeners, go to **Cobalt Strike -> Listeners**. A tab will open with a list of all configured payloads and Listeners.

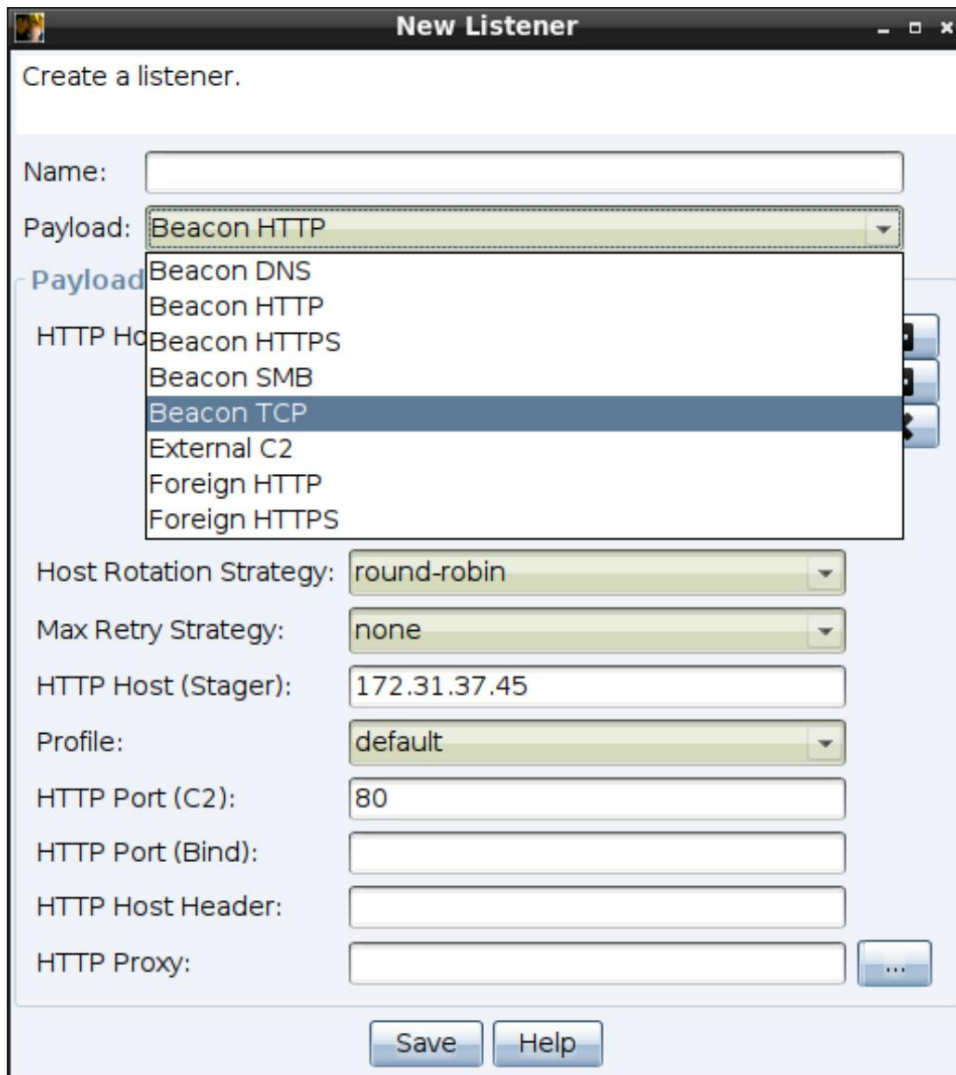


The screenshot shows the 'Listeners' tab in the Cobalt Strike interface. It contains a table with the following columns: name, payload, host, port, bindto, beacons, and profile. There are three rows of data. Below the table are buttons for 'Add', 'Edit', 'Remove', 'Restart', and 'Help'.

name	payload	host	port	bindto	beacons	profile
ec2 - HTTP Beacon	windows/beacon_http/reverse_http	secure.iosenolove.c...	80		secure.iosenolove.com	default
SMB score_pd233	windows/beacon_bind_pipe		score_pd233			
TCP 9002 (Local)	windows/beacon_bind_tcp		9002	127.0.0.1		

Figure 18. Listener Management tab

Click **Add** to create a new Listener. This will display the New Listener panel.



New Listener

Create a listener.

Name:

Payload: **Beacon HTTP**

- Beacon DNS
- Beacon HTTP
- Beacon HTTPS
- Beacon SMB
- Beacon TCP
- External C2
- Foreign HTTP
- Foreign HTTPS

Host Rotation Strategy: **round-robin**

Max Retry Strategy: **none**

HTTP Host (Stager):

Profile: **default**

HTTP Port (C2):

HTTP Port (Bind):

HTTP Host Header:

HTTP Proxy:

Use the **Payload** dropdown to select one of the available payload/Lestener types that you want to configure. Each of them has different options, which are described in the following sections:

[DNS Beacon on page 34](#)

[HTTP Beacon and HTTPS Beacon on page 38](#)

[SMB Beacon on page 42](#)

[TCP Beacon on page 44](#)

[External C2 on page 46](#)

[Foreign Listeners on page 47](#)

To edit a Lestener, select it and click **Edit**. To remove a Lestener, select it and click **Remove**.

Beacon payload Cobalt Strike

Most often, you set up Listeners for Cobalt Strike's Beacon. Beacon is a payload for simulating the actions of intruders. Use it to transfer data over the network via HTTP, HTTPS, or DNS. You can also set limits on which nodes go off the network by controlling peer beacons via Windows named pipes and TCP sockets.

Beacon is flexible and supports asynchronous and interactive communication. Asynchronous communication is low and slow ("low and slow" communication pattern).

Beacon will contact the command and control server, download its jobs, and terminate network activity. Interactive communication takes place in real time. Beacon's network

indicators can be changed. Redefine Beacon interaction with Cobalt Strike's Malleable C2 language. This will allow you to mask the Beacon's activity so that it looks like other malware or blends in with legitimate traffic. For more information, see [Manage and control Malleable on page 97](#).

Staging payload

There is one topic that deserves to be mentioned as a reference and that is the staging payload. Many attack frameworks separate the attack itself from what it does. What the attack does is known as the payload. Payload is often divided into two parts: stage and stager payload. A stager is usually a small hand-optimized assembly language program that loads the stage payload, injects it into memory, and passes execution to it. This process is known as staging.

Staging is required for some attacks. Many attacks have severe limits on how much data they can load into memory and execute after a successful exploit. This greatly limits post-exploitation options, unless you deliver your payload for post-exploitation in stages.

Cobalt Strike uses staging in its user-driven attacks. These are most of the items in **the Payload and Attack section**. The stagers used there depend on the payload that comes with the attack. For example, the HTTP Beacon has an HTTP stager. DNS Beacon has stager DNS TXT records. Not all payloads have stager options. A payload without a stager cannot be delivered using these attack options.

If you don't need staging payload, you can turn it off. Set the **host_stage** setting in your Malleable C2 profile to false. This will prevent Cobalt Strike from hosting payload stages on its web and DNS servers. This has a big advantage for OPSEC. With staging enabled, anyone can connect to your server, request a payload, and parse its contents to find information in your payload configuration. In Cobalt Strike 4.0 and later, post-exploitation and lateral movement actions do without stagers and

use the full payload where possible. When disabling payload staging, you shouldn't notice this when you're ready for post-production.

DNS Beacon

DNS Beacon is a favorite feature of Cobalt Strike. This payload uses DNS queries for communication. These DNS queries are lookups for domains for which your command and control server is authoritative.

The DNS response tells the Beacon to cease activity or connect to you to download jobs. The DNS response also tells the Beacon how to download jobs from your command and control server.

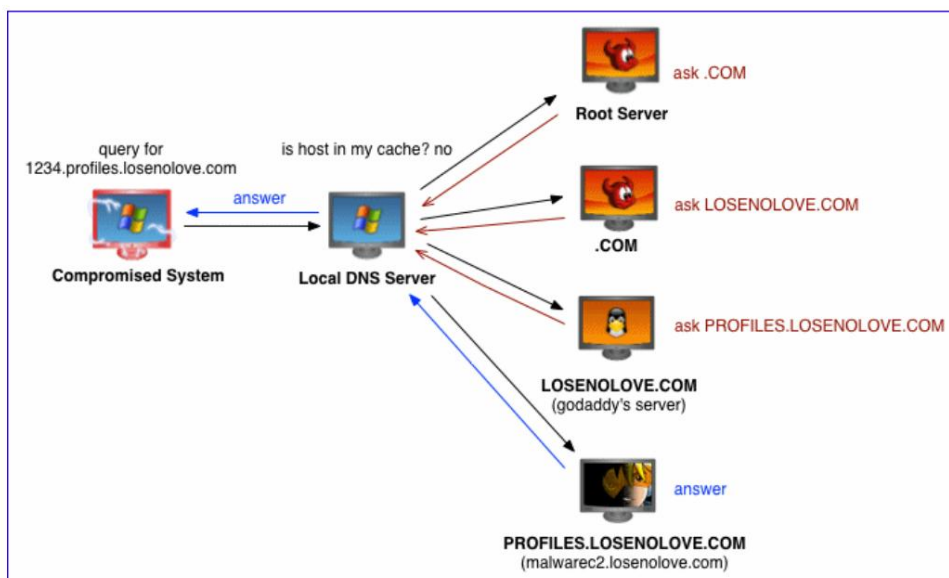


Figure 21. DNS Beacon in action

In Cobalt Strike 4.0 and later, the DNS Beacon is a DNS-only payload. There is no HTTP communication mode in this payload. This is different from previous versions of the product.

Data channels

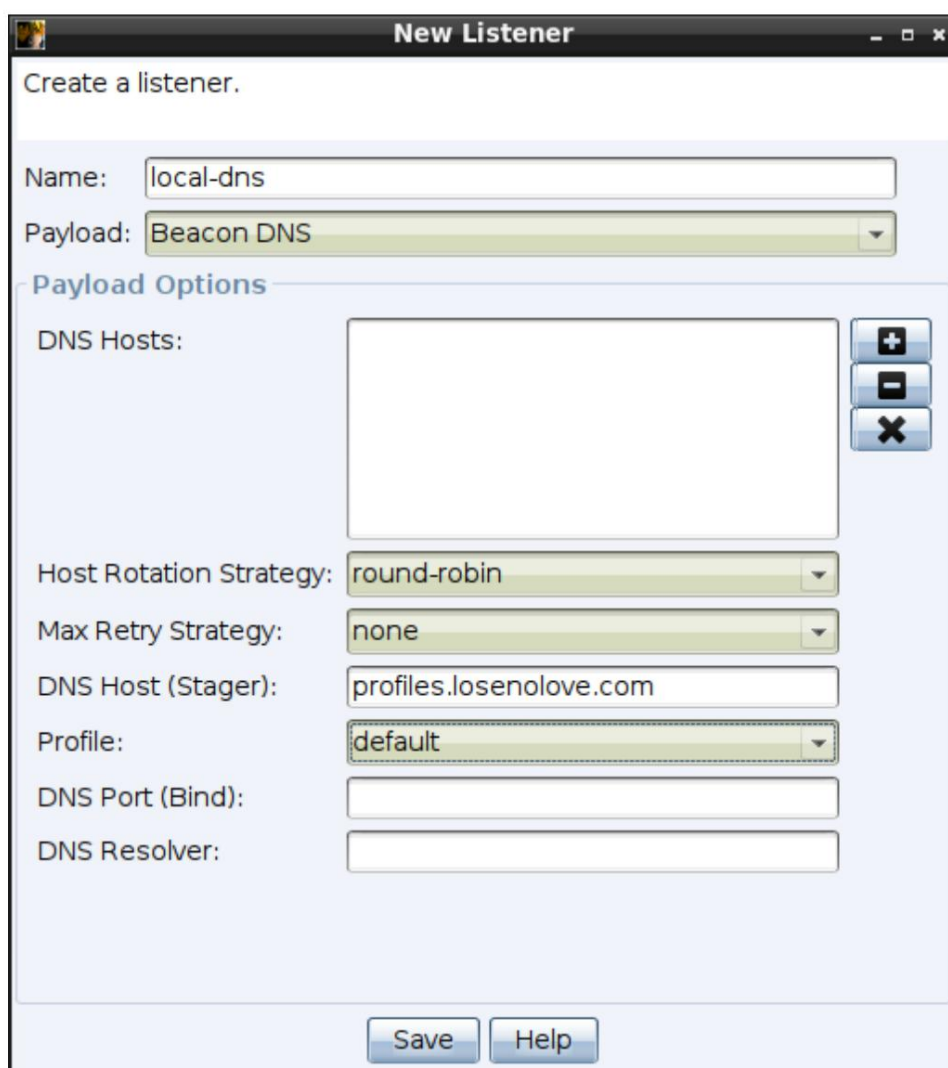
Today, the DNS Beacon can load jobs through DNS TXT, DNS AAAA, or A records. This payload can switch between these data channels while on target. Use the mode command to change its current data channel. **mode dns** - A-record data channel. **mode dns6** - AAAA record channel. And **the dns-txt mode** is the data feed of the TXT record. By default, the TXT record data channel is used.

Keep in mind that the DNS Beacon is not registered until the job is available. Use the **checkin** command to request that the DNS Beacon be registered the next time it hits the server.

Configuring DNS Listener

To create a DNS Beacon Listener, select **Cobalt Strike -> Listeners** from the main menu and click the **Add** button at the bottom of the Listeners tab. The

New Listener panel will be displayed.



New Listener

Create a listener.

Name:

Payload:

Payload Options

DNS Hosts:
+ - X

Host Rotation Strategy:

Max Retry Strategy:

DNS Host (Stager):

Profile:

DNS Port (Bind):

DNS Resolver:

Figure 22. DNS Beacon Options

Select **Beacon DNS** in the **Payload** field and give the Listener a **Name field**. Be sure to give the new Listener a memorable name, as this is the name you will use to refer to it through Cobalt Strike's commands and workflows.

Options

DNS Hosts - Click **[+]** to add one or more domains for the Beacon. Your Cobalt Strike C&C server must be authoritative for the domains you specify. Create a DNS A-record and point it to your Cobalt Strike command and control server. Use NS records to delegate multiple domains or subdomains to your C&C A record. The length of the Beacon's host list is limited to 255 characters. It includes a

randomly assigned URI for each host and a separator between each list item. If the length is exceeded, hosts will be dropped from the end of the list until it fits in the space provided. Dropped hosts will appear in the C&C logs.

Host Rotation Strategy - This value configures the behavior of Beacons when selecting a host(s) from a list to use when establishing a connection. Choose one of the following:

round-robin: Choose to cycle through a list of hostnames in the order in which they appear. Each host is used for one connection.

random: Select to randomly select a host name from the list each time a connection is attempted.

failover-xx: Select to use the working host as long as possible.
Use each host on the list until they reach the number of consecutive failures (x) or time period (m,h,d), then use the next host.

rotate-xx: Select to use each host for a specific period of time. Use each host in the list for the specified duration (m,h,d), then use the next host.

Max Retry Strategy - configures Beacon behavior to exit after several consecutive failed attempts to connect to the command and control server. There are several default options, or you can create your own list using the **LISTENER_MAX_RETRY_STRATEGIES** function. See [LISTENER_MAX_RETRY_STRATEGIES on page 167.](#)

none: Select to prevent the Beacon from exiting due to failed connection attempts.

exit-xxx: These options use the syntax **exit-[max_attempts]-[attempts_to_increase]-[duration][m,h,d]**. **max_retries** is the number of consecutive failed attempts after which the Beacon will exit. **retries_before_increment** is the number of consecutive failed attempts before the sleep time is increased. **duration** is the number of minutes, hours, or days for the new sleep duration.

The sleep time will not be updated if the current sleep time is greater than the new set value. Sleep time is affected by the current jitter value. On any successful connection, the failed attempts counter is reset to zero, and the sleep time returns to the previous value.

DNS Host (Stager) - This is where the stager of the DNS Beacon's TXT records is configured. This stager is only used with Cobalt Strike features that require it. Your C&C must be authoritative for this domain.

Profile - Allows you to customize the Beacon using the Malleable C2 profile.

DNS Port (Bind) - This field specifies the port that your Beacon's DNS server will bind to. This option is useful if you want to set up a redirector that changes the port, such as a redirector that accepts connections on port 53 but forwards them to your C&C on a different port.

DNS Resolver - Allows DNS to exit using a specific DNS resolver instead of using the default DNS resolver for the target server. Specify the IP address of the desired resolver, this DNS resolver is not used by Beacon's DNS stager.

Verification

To verify the DNS configuration, open a terminal and type **nslookup jibberish.beacon domain**. If you get an A-record of 0.0.0.0, then your DNS is set up correctly. If you don't get a response, then your DNS configuration is incorrect and DNS Beacon won't communicate with you.

Notes

- Make sure your DNS records point to the primary address of your network interface. The DNS server will always send responses from the primary address of your network interface. DNS resolvers typically discard responses when they request information from one server and receive a response from another.
 - If you are behind a NAT device, make sure you are using your public IP address for the NS record and have your firewall configured to forward UDP traffic on port 53 to your system. Cobalt Strike includes a DNS server to manage the Beacon
- To set up network traffic indicators for your DNS Beacons, see [DNS Beacons on page 111](#) in the Malleable C2 Help.

HTTP Beacon and HTTPS Beacon

HTTP and HTTPS Beacons download jobs using an HTTP GET request. Data Beacons send data back using an HTTP POST request. This option is the default. You have impressive control over the behavior and indicators in this payload with Malleable C2.

Configuring HTTP(S) Listener

To create an HTTP or HTTPS Beacon Listener, select **Cobalt Strike -> Listeners** from the main menu and click the **Add** button at the bottom of the Listeners tab screen. The New Listener panel opens.

New Listener

Create a listener.

Name:

Payload:

Payload Options

HTTP Hosts:

Host Rotation Strategy:

Max Retry Strategy:

HTTP Host (Stager):

Profile:

HTTP Port (C2):

HTTP Port (Bind):

HTTP Host Header:

HTTP Proxy:

Figure 19. HTTP Beacon parameters

Select **Beacon HTTP** or **Beacon HTTPS** in the **Payload** field and give the Listener a **Name field**. Be sure to give the new Listener a memorable name, as this is the name you will use to refer to it through Cobalt Strike's commands and workflows.

Options

HTTPS(S) Hosts - Click **[+]** to add one or more hosts to which will handle the HTTP Beacon. Press **[-]** to delete one or more hosts. Press **[X]** to clear the current hosts. If you have multiple hosts, you can paste in this window a list of hosts to access, separated by commas.

The length of the Beacon's host list is limited to 255 characters. It includes a randomly assigned URI for each host and a delimiter between each list item. If the length is exceeded, hosts will be dropped from the end of the list until they fit in the allotted space. Dropped hosts will appear in the C&C logs.

Host Rotation Strategy - This value configures the behavior of Beacons when selecting a host(s) from a list to use when establishing a connection. Choose one of the following:

round-robin: Choose to cycle through a list of hostnames in the order in which they appear. Each host is used for one connection.

random: Select to randomly select a host name from the list each time a connection is attempted.

failover-xx: Select to use the working host as long as possible.
Use each host on the list until they reach the number of consecutive failures (x) or time period (m,h,d), then use the next host.

rotate-xx: Select to use each host for a specific period of time. Use each host in the list for the specified duration (m,h,d), then use the next host.

Max Retry Strategy - configures Beacon behavior to exit after several consecutive failed attempts to connect to the command and control server. There are several default options, or you can create your own list using the **LISTENER_MAX_RETRY_STRATEGIES** function. See [LISTENER_MAX_RETRY_STRATEGIES on page 167.](#)

none: Select to prevent the Beacon from shutting down due to failed connection attempts.

exit-xxx: These options use the syntax **exit-[max_attempts]-[attempts_to_increase]-[duration][m,h,d]**. **max_retries** is the number of consecutive failed attempts after which the Beacon will exit. **retries_before_increment** is the number of consecutive failed attempts before the sleep time is increased. **duration** is the number of minutes, hours, or days for the new sleep duration.

The sleep time will not be updated if the current sleep time is greater than the new set value. Sleep time is affected by the current jitter value. On any successful connection, the failed attempts counter is reset to zero, and the sleep time returns to the previous value.

HTTP Host (Stager) - This setting controls the HTTP stager host for the HTTP Beacon. This value is only used if you are combining this payload with an attack that requires an explicit stager.

Profile - This field specifies the Malleable C2 profile variant. A variant is a way of specifying multiple profile variants in a single file. With options, each HTTP or HTTPS Listener you configure can have different network indicators.

HTTP Port (C2) - This field specifies the port that your HTTP Beacon will contact.

HTTP Port (Bind) - This field specifies the port to which your HTTP Beacon web server. This setting is useful if you want to set up a redirector that changes the port (for example, a redirector that accepts connections on port 80 or 443 but forwards them to your C&C on a different port).

HTTP Host Header - If specified, this value is passed to your HTTP stagers and through your HTTP communications. This option makes it easier to take advantage of domain fronting with Cobalt Strike.

HTTP Proxy - Click the ... button to specify a specific proxy configuration for this payload.

Manual HTTP Proxy Configuration

The **(Manual) Proxy Settings** dialog box offers several options for managing the proxy configuration for the Beacon's HTTP and HTTPS requests. The Beacon's default behavior is to use the Internet Explorer proxy configuration for the current process/user context.

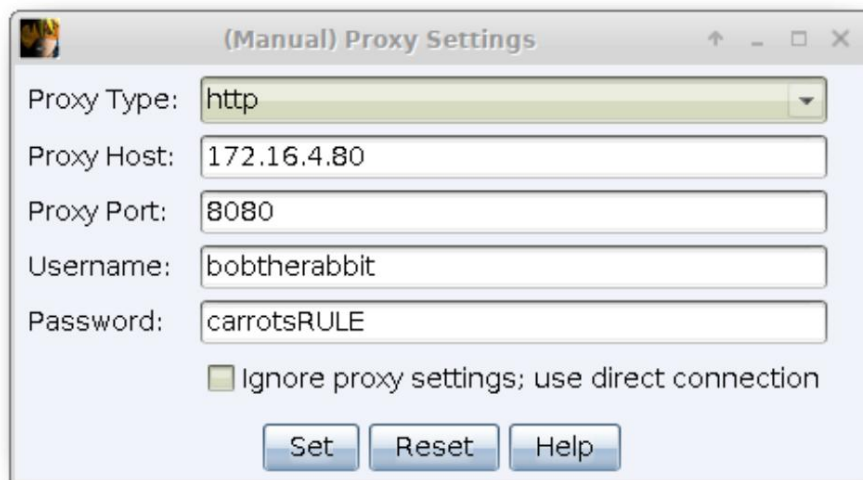


Figure 20. Proxy Server Setup Guide

The **Type** field sets the proxy type. The **Host** and **Port** fields tell the Beacon where the proxy is located. The **Username** and **Password** fields are optional. These fields specify the credentials that Beacon uses to authenticate with the proxy.

Select **Ignore proxy settings; use direct connection** to force Beacon to attempt HTTP and HTTPS requests without going through a proxy.

Click **Set** to update the Beacon's dialog box with the desired proxy settings. Click the **Reset** button to return the proxy configuration to its default behavior.

NOTE:

Manual proxy configuration only affects HTTP and HTTPS Beacon stage. It does not apply to stagers.

Redirectors

The redirector is the system that sits between your target's network and your C&C. Any connections coming into the redirector are forwarded to your C&C for processing. A redirector is a way to provide multiple hosts for your Beacon to access. The redirector also improves security, as it makes it harder to trace your C&C's true location.

Cobalt Strike's Listener controls support the use of redirectors. Just specify your redirectors when setting up an HTTP Listener or an HTTPS Beacon.

Cobalt Strike does not verify this information. If the host you specify is not associated with the current host, Cobalt Strike thinks it is a redirector. One easy way to turn a server into a redirector is to use socat. Here is the socat syntax to redirect all connections on port 80 to the C&C at

192.168.12.100 on port 80:

```
socat TCP4-LISTEN:80,fork TCP4:192.168.12.100:80
```

SMB Beacon

The SMB Beacon uses named pipes to communicate through the parent Beacon. This peer-to-peer communication works with Beacons on the same host. It also works over the web.

Windows encapsulates named pipe communication in the SMB protocol. Hence the name - SMB Beacon.

Configuring SMB Listener

To create an SMB Beacon Listener select **Cobalt Strike -> Listeners** from the main menu and click the **Add** button at the bottom of the Listeners tab screen. The New Listener panel opens.

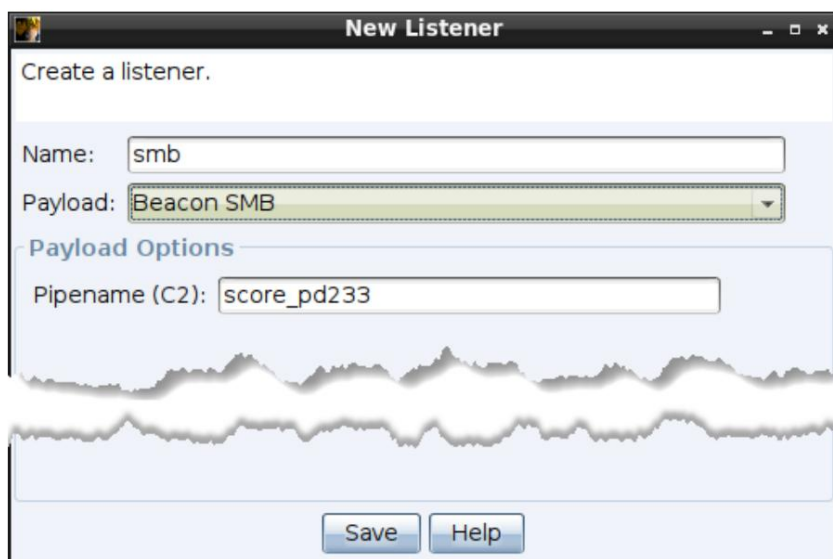


Figure 23. SMB Beacon

Select **Beacon SMB** in the **Payload** field and give the Listener a **Name field**. Be sure to give the new Listener a memorable name, as this is the name you will use to refer to it through Cobalt Strike's commands and workflows. The only parameter associated with SMB Beacon is

Pipename (C2). You can specify a specific named pipe or accept the default.

The SMB Beacon is compatible with most Cobalt Strike actions that generate payloads. The exception is user-driven attacks that require explicit stagers.

The Cobalt Strike's post-exploitation and lateral movement actions that spawn a payload will attempt to take control (communication) of the SMB Beacon for you. If you are starting the SMB Beacon manually, you will need to link it to the parent Beacon.

Linking and disconnecting

In the Beacon console, use the **link [host] [channel]** command to link the current Beacon to SMB Beacon that is waiting for a connection. When the current Beacon registers, its associated Beacons will also register. To merge with

normal traffic, linked Beacons use Windows named pipes to communicate. This traffic is encapsulated in the SMB protocol. There are a few caveats to this approach:

1. Hosts with SMB Beacon must accept connections on port 445.
2. You can only link beacons that are controlled by the same copy of Cobalt Strike.

If you get error 5 (Access Denied) after trying to bind a Beacon: Steal the domain user's token, or use **make_token DOMAIN\user password** to populate your current token with valid credentials for the target. Try linking Beacon again. To unlink a Beacon, use **unlink [ip] [session PID]** on the parent or child Beacon.

The [session PID] argument is the process ID of the Beacon to detach. This value allows you to specify a particular Beacon to detach when there are multiple child Beacons.

When you disconnect an SMB Beacon, it is not disabled or deleted. Instead, it goes into a state where it is waiting for a connection from another Beacon. You can use the link command to resume control of an SMB Beacon from another Beacon at a later date.

Hidden Peer-to-Peer Beacon Communication

It's hard to go unnoticed when many compromised systems go online. Use peer-to-peer communication to solve this problem. This feature allows you to link beacons to each other. Linked Beacons download jobs and send results via their parent Beacon.

Use **mode smb** to convert a Beacon to a peer that waits for another Beacon to connect. Use **link [ip address]** to link the

current Beacon to a peer that is waiting for a connection. When the current Beacon registers, its associated Beacon will also register.

In order to merge with normal traffic, connected Beacons use SMB channels for communication. There are a few caveats to this approach:

1. Hosts with a peer beacon must accept connections on port 445.
2. You can only link beacons that are controlled by the same copy of Cobalt Strike.

If you get error 5 (access denied) when trying to bind a Beacon: steal the domain user's token, or use **shell net use \host /U:DOMAIN\user password** to establish a session with the host. This does not require an administrator. Any valid domain user will do. Once you've established a session, try connecting to the Beacon again.

To unlink a Beacon , use the **unlink [ip address]** command on the parent or child Beacon. Later, you can re-link an unlinked Beacon (or link it to another Beacon).

Once a Beacon becomes a peer, there is no way to make it communicate over HTTP or DNS again. If you want to destroy a peer Beacon, use the exit command. If you want the Beacon to communicate over HTTP or DNS, ask the peer Beacon to give you another Beacon session.

Peer Beacon as payload Some systems cannot communicate with the Internet. In cases like this, it's good to have a way to deliver a ready-to-connect Beacon so you can connect to it. Use **[host] -> Login -> psexec** or **[host] -> Login -> psexec (psh) with a beacon (connect to target) Listener**. This will allow you to run a peer-to-peer Beacon on the host without the need for an internet connection for staging.

You can also set up a Listener to deliver a peer beacon. Create a Listener for windows/beacon_smb/reverse_tcp. This Listener will form your peer Beacon. After it is installed, you will still need to link it to another Beacon.

If staging is laborious, you can ask Cobalt Strike to export a fully-fledged peer-to-peer Beacon as an executable, DLL, PowerShell script, or raw shellcode block. Go to **Payloads -> Windows Stageless Payload** and select SMB Beacon.

TCP Beacon

The TCP Beacon uses a TCP socket to communicate through the parent Beacon. This peer-to-peer communication works with Beacons on the same host or across the network.

Configuring TCP Listener

To create a TCP Beacon Listener, select **Cobalt Strike -> Listeners** from the main menu and click the **Add** button at the bottom of the Listeners tab screen.

The New Listener panel opens.

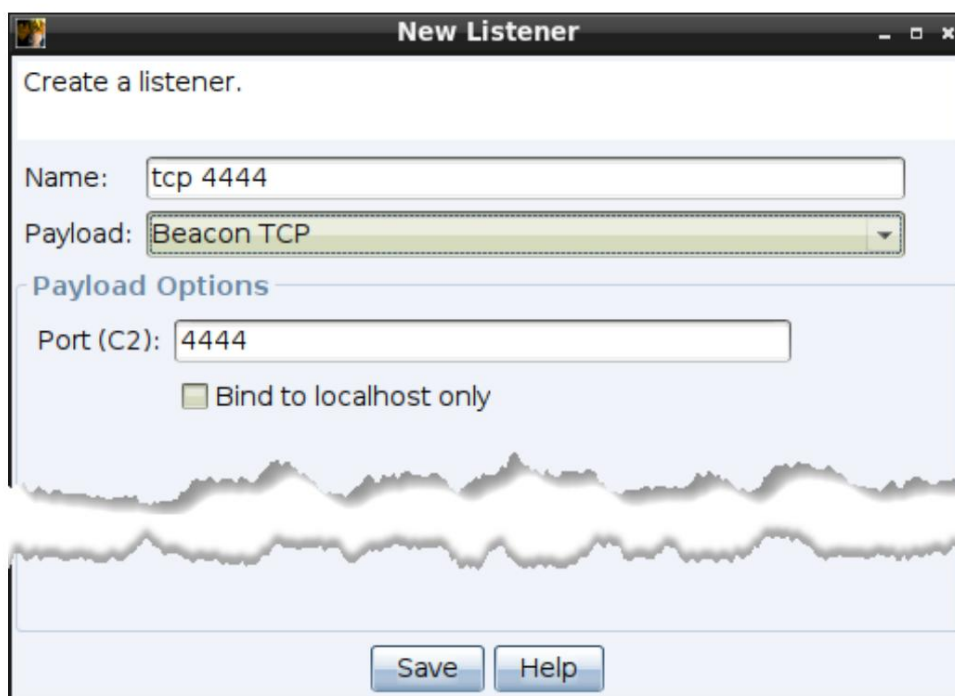


Figure 24. TCP Beacon

Select **Beacon TCP** in the **Payload** field and give the Listener a **Name field**. Be sure to give the new Listener a memorable name, as this is the name you will use to refer to it through Cobalt Strike's commands and workflows. A TCP Beacon configured in this way is a bind payload. The bind payload is the payload that is waiting for a connection from its control component (in this case, another Beacon session).

Options

Port (C2) - this setting controls the port on which the TCP Beacon will listen connections.

Bind to localhost only - Set to bind the TCP Beacon to 127.0.0.1 when it listens for a connection. This is a good option if you are using TCP Beacon for localhost only actions.

The TCP Beacon is compatible with most Cobalt Strike operations that generate payloads. The exception is, as with the SMB Beacon, user-driven attacks that require explicit stagers.

The Cobalt Strike's post-exploitation and lateral movement actions that generate the payload will attempt to take control (connect) to the SMB Beacon for you. If you are starting a TCP Beacon manually, you will need to connect to it from the parent Beacon.

Connection and disconnection

In the Beacon's console, use the **connect [ip address] [port]** command to connect the current session to the TCP Beacon that is waiting for a connection. When the current session is registered, its associated Beacons will also be registered.

To unlink a Beacon, use the **unlink [ip address] [session PID]** command in the console of the parent or child session. Later, you can reconnect to the TCP Beacon from the same host (or another host).

External C2

External C2 is a concept that allows third party programs to act as a communication layer for Cobalt Strike's payload. These third party programs connect to Cobalt Strike to read the data meant for them and write the results data from the payloads that are controlled in this way. The external C2 server is what these third party programs use to communicate with your Cobalt Strike command and control server.

Configuring the External C2 Listener

To create an Outer C2 Beacon Listener select **Cobalt Strike -> Listeners** from the main menu and click the **Add** button at the bottom of the Listeners tab screen.

The New Listener panel opens.

Go to **Cobalt Strike -> Listeners**, click **Add**, and select External C2 as the payload.

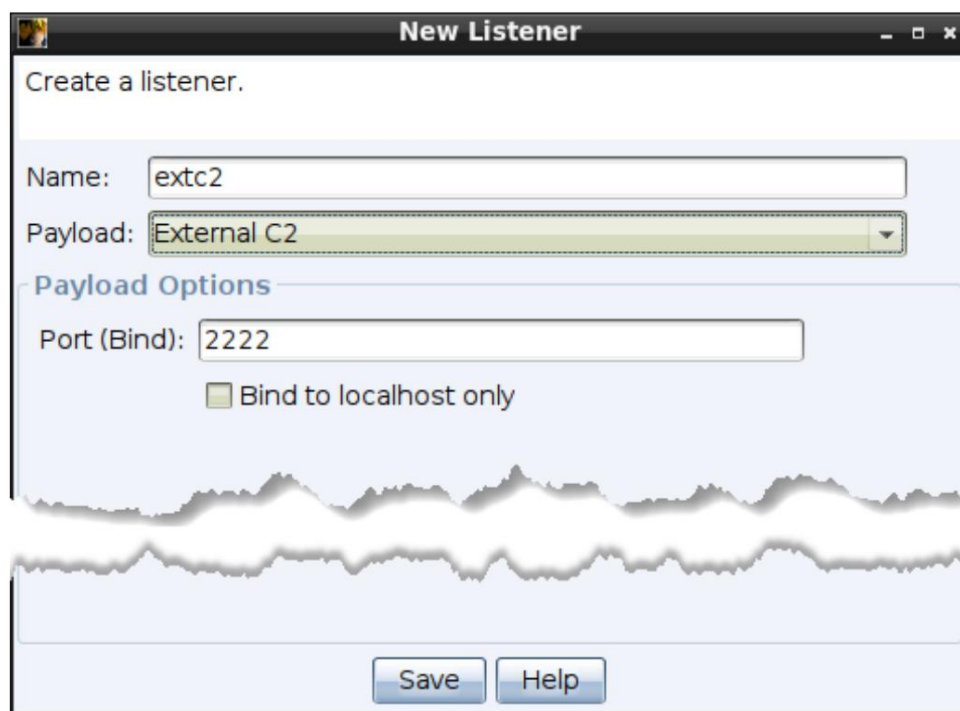


Figure 25 External C2

Select **External C2** in the **Payload** field and give the Listener a **Name** field. Be sure to give the new Listener a memorable name, as this is the name you will use to refer to it through Cobalt Strike's commands and workflows.

Port (Bind) - Specify the port on which the external C2 server listens for a connection.

Bind to localhost only - set to external C2 server to do only local.

NOTE:

External C2 Listeners are not like other Cobalt Strike Listeners . You cannot target them with Cobalt Strike's post-exploitation actions. This option is just an option to customize the interface itself.

Specification

The External C2 interface is described in the External C2 specification.

- | [External C2 Specification](#)
- | [extc2example.c](#)

If you want to adapt the example (Appendix B) from the spec for use in a third party C2, you can use the [3-clause BSD license](#) to the code contained in the [specification](#).

Third Party Content

Here is a list of third party projects and posts that reference, use or build on External C2:

- | [Custom Command and Control \(C3\)](#) from [F-Secure Labs](#). A framework for quickly prototyping custom C2 channels. [external_c2_framework](#)
- | [by Jonathan Echavarria](#). Python framework for creating external C2 clients and servers.
- | [ExternalC2 Library](#) by [Ryan Hanson](#). .NET library with Web API, WebSockets and direct socket. Includes unit tests and comments.
- | [Tasking Office 365 for Cobalt Strike C2](#) from [MWR Labs](#). Discussion and demonstration of Office 365 C2 for Cobalt Strike.
- | [Shared File C2](#) from [Outflank B.V.](#) POC to [use file/share for management and control](#).

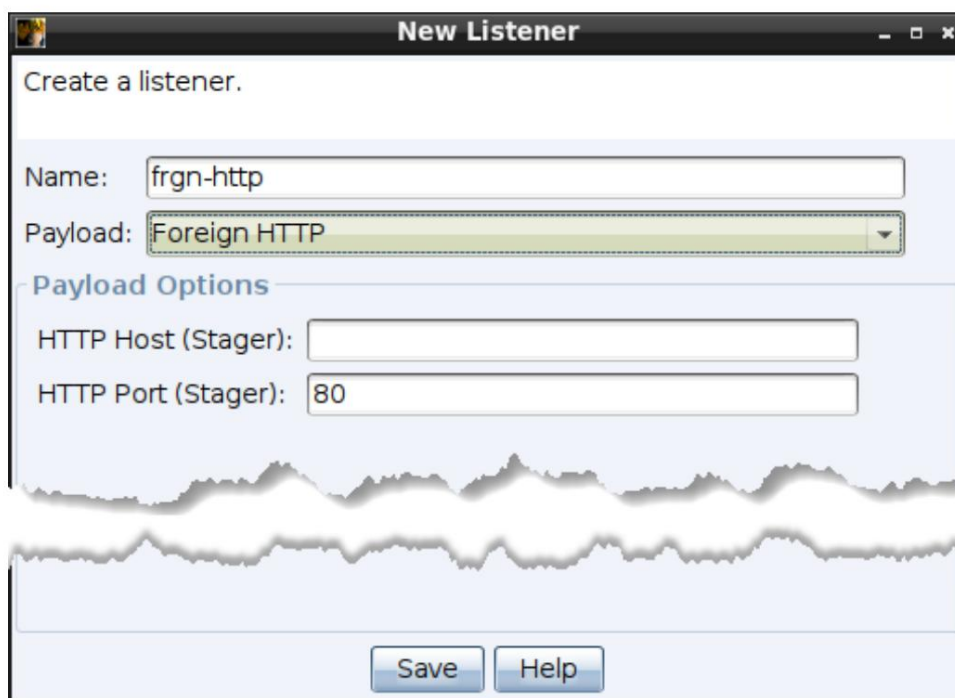
Third Party Listeners

Cobalt Strike supports the concept of third party Listeners. These are aliases for **x86 payload handlers** hosted in the Metasploit framework or other instances of Cobalt Strike. To transfer a Windows HTTPS Meterpreter session to another user using msfconsole, set up a Foreign HTTPS payload and set the Host and Port values to the appropriate handler. You can use third party Listeners anywhere you would use Cobalt Strike's x86 Listener.

Configuring a Third Party Listener

To create a Third Party Beacon Listener select **Cobalt Strike -> Listeners** and click the **Add** button at the bottom of the Listeners tab screen.

The New Listener panel opens.



Third Party HTTP

Select **Foreign HTTP** or **Foreign HTTPS** in the **Payload** field and set the Listener to **the Name field**. Be sure to give the new Listener a memorable name, as this is the name you will use to refer to it through Cobalt Strike's commands and workflows.

Options

HTTP(S) Host (Stager) - This field specifies the name of the server hosting the your third party Listener.

HTTP(S) Port (Stager) - This field specifies the port of the server where your a third party Listener is waiting for a connection.

Consolidation of infrastructure

Cobalt Strike's model for distributed operations is to create a separate C&C server for each phase of your operation. For example, it makes sense to separate the infrastructure for post-exploitation and persistence. If post-operational activities are detected and you do not want the callbacks to be removed when the infrastructure is restored, which will allow you to return to

net.

Some stages of the operation require the use of several redirectors and communication channels. Cobalt Strike 4.0 is friendly to this.

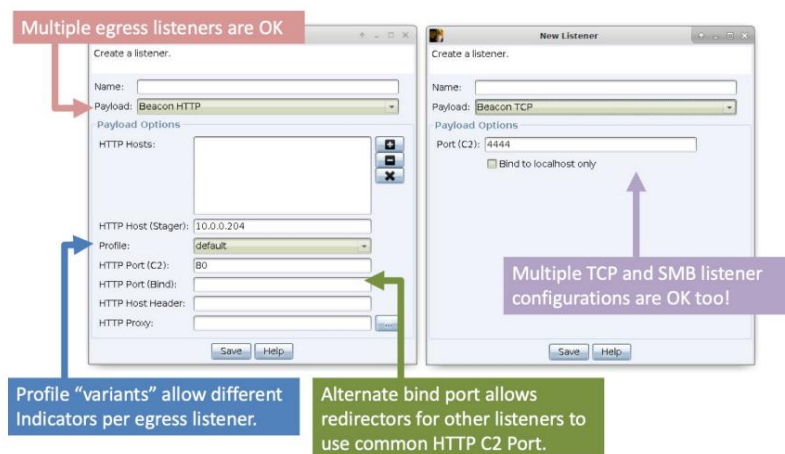


Figure 26. Features of infrastructure consolidation

You can bind multiple HTTP, HTTPS and DNS Listeners to one Cobalt Strike command and control server. Payloads also support port binding in their configuration. This allows you to use a common port for your channel (80, 443, or 53) in the redirector and C2 settings, but bind these Listeners to different ports to avoid port conflicts on your C&C.

To give variety to your network indicators, the Malleable C2 Cobalt Strike's profiles can contain several options. A variant is a way to add different versions of the current profile to the same profile file. You can specify a profile option when defining each HTTP Listener or HTTPS Beacon.

In addition, you can define multiple TCP and SMB beacons on the same command and control server, each with different channel and port configurations. Any outgoing Beacon from the same command and control server can manage any TCP or SMB Beacon after it has been deployed to the target environment.

payload security features

Cobalt Strike takes steps to secure Beacon communications and ensure that Beacon can only receive missions from their command and control server and send results to it.

When you first set up a Beacon, Cobalt Strike generates a public/private key pair that is unique to your C&C. The C&C public key is embedded in the Beacon stage payload. Beacon uses the C&C public key to encrypt the session metadata it sends to the C&C.

The Beacon must always send session metadata before the C&C can issue jobs and receive results from the Beacon session. This metadata contains a random session key generated by this Beacon. The C&C uses each Beacon's session key to encrypt jobs and decrypt results.

Each Beacon and data channel implementation uses the same schema. You have the same security when using an A-record data feed in a hybrid HTTP and DNS Beacon as you do with an HTTPS Beacon. Please note that all of the above

applies to the Beacon after it has been posted.

Payload stagers, due to their size, do not have built-in security features.

Initial Access

Cobalt Strike contains several features that help you to get a foothold on the target. This applies to both profiling potential targets and payload creation and delivery.

Client-side system profiler

The System Profiler is a reconnaissance tool for client-side attacks. This tool starts a local web server and fingerprints everyone who visits it. The system profiler provides a list of applications and plug-ins that it has found in the user's browser. The system profiler also attempts to discover the internal IP address of users who are behind a proxy server.

To start the system profiler, go to **Attacks -> System Profiler**. To launch the profiler, you must specify the URI to bind to and the port to launch the Cobalt Strike web server.

If you specify a redirect URL, Cobalt Strike will redirect visitors to that URL after their profile is retrieved. Click the **Launch** button to start the system profiler.

The system profiler uses an unsigned Java applet to reveal the target's internal IP address and determine which version of Java it is running. With Java's click-to-run security feature, this can raise suspicion. Uncheck **the Use Java Applet** when retrieving information box to remove the Java applet from the system profiler.

Check the **Enable SSL** checkbox to expose the system profiler over SSL. This field will be disabled if you do not provide a valid SSL certificate using Malleable C2. This is discussed in chapter 11.

Application Explorer

To view the results of the system profiler, go to the **View -> Applications menu**. The Applications tab opens with a table showing all information about the applications collected by the system profiler.

Analysis Tips

Application Explorer contains a lot of useful information for planning a targeted attack. The following describes how to get the most out of this information: The internal IP address field is compiled from data from a harmless, unsigned Java applet. If this field says "unknown", it means that the Java applet has most likely not been launched. If you see an IP address here, it means that an unsigned Java applet has been launched.

Internet Explorer reports the version installed by the user. When Internet Explorer receives updates, the version information does not change. Cobalt Strike uses the version of JScript.dll to evaluate the extent of Internet Explorer patches. Go to support.microsoft.com and find the build number of JScript.dll (third number in the version string) to match it with an Internet Explorer update.

***64** next to an app means it's an x64 app.

Cobalt Strike Web Services

Many of Cobalt Strike's features run on its own web server. Such services include the system profiler, HTTP Beacon, and web drive-by attacks. It is possible to host multiple Cobalt Strike functions on the same web server. To manage Cobalt Strike's web

services, go to **View -> Web Drive-by -> Manage**. Here you can either copy any Cobalt Strike URL to your clipboard or stop the Cobalt Strike web service. Use **View -> Web Log** to monitor visits to your Cobalt Strike web

services.

If the web server sees a request from a Lynx, Wget, or Curl browser; Cobalt Strike automatically returns a 404 page. It does this as a little defense against blue team surveillance. This can be configured with the Malleable C2 config `'.http-config.block_useragents'`.

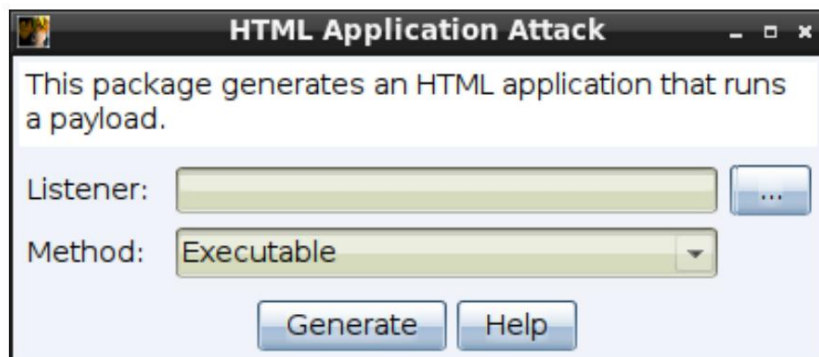
Packets of user-driven attacks

The best attacks are not exploits at all. Rather, the best attacks take advantage of regular functions to gain code execution. Cobalt Strike makes it easy to set up multiple user-driven attacks. These attacks take advantage of the Listeners you have already configured. Navigate to the menu item **Payloads** and select one of the following options.

HTML application

An HTML application is a Windows program written in HTML and supported by Internet Explorer scripting language. This package generates an HTML application that runs Cobalt Strike's Listener. Go to **Payloads**

-> **HTML Application**.



Options

Listener - Click the ... payload, to select the Listener for which you want to output button.

Method - Use the dropdown to select one of the following methods to launch the selected Listener:

Executable: This method writes an executable to disk and runs it.