

Cobalt Strike Analysis and Tutorial: How Malleable C2 Profiles Make Cobalt Strike Difficult to Detect

By Chris Navarrete, Durgesh Sangvikar, Andrew Guan, Yu Fu, Yanhui Jia, Siddhart Shibiraj

Published: 2022-03-16 · Archived: 2026-04-05 16:16:02 UTC

Executive Summary

Cobalt Strike is commercial threat emulation software that emulates a quiet, long-term embedded actor in a network. This actor, known as Beacon, communicates with an external team server to emulate command and control (C2) traffic. Due to its versatility, Cobalt Strike is commonly used as a legitimate tool by red teams – but is also widely used by threat actors for real-world attacks.

Cobalt Strike users control Beacon’s HTTP indicators through a profile, and can select either the default profile or a customizable Malleable C2 profile.

In this blog post, we will go through the concepts and definitions associated with these profiles, and explore differences between default and customized Malleable C2 profiles used in the Cobalt Strike framework as well as in some true attacks in the wild. In doing so, we demonstrate how the Malleable C2 profile lends versatility to Cobalt Strike, and why this versatility makes Cobalt Strike an effective emulator for which it is difficult to design traditional firewall defenses.

Palo Alto Networks customers receive protections against malicious uses of Cobalt Strike through Cortex XDR and the WildFire and Threat Prevention subscriptions for the Next-Generation Firewall.

Profile Options for Cobalt Strike

The Cobalt Strike tool’s primary configuration is specified using a profile file. The tool uses the values present in the profile to generate the Beacon payload, and users create the profile and set its values with a Malleable Command and Control (C2) profile language.

The profile specifies how the beacon will transform and store data in a transaction.

Within a profile, options are divided into global options and local options. Global options update the global Beacon settings, while local options are transaction-specific. Local option changes within one transaction do not affect the output from other transactions.

The profile is divided into multiple sections to specify the values for different parts of the C2 communications. An [example](#) of a generic structure of the profile is as follows:

this is a comment

```
set global_option "value";

protocol-transaction {

    set local_option "value";

    client {

        # customize client indicators

    }

    server {

        # customize server indicators

    }

}
```

Different parts of the profile are explained below.

Global Options

Global options are global to C2 communications. Options such as sleeptime and jitter define the frequency of Beacon's check-in with the team server. Here is a list of a few global options with example values:

```
set sample_name "Profile Name";

set sleeptime "30000";

set jitter "20";

set useragent "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/55.0.2883.87 Safari/537.36";

set host_stage "false";
```

If you are interested in a more comprehensive list of all the global options, refer to this [Cobalt Strike user guide](#).

Local Options

On the other hand, the scope for local options is per transaction only. The options for one transaction do not affect the other.

Examples of Local options:

```
set uri "URI_For HTTP transaction";  
  
set verb "POST";  
  
set uri_x86 "StagetURI_for_x86";  
  
set uri_x64 "StagetURI_for_x64";
```

In addition to these options, a profile can specify different protocol-transactions to carry out different actions. Below are example transactions, as well as brief explanations of their usage:

- **http-stager:** The Beacon is a staged payload. The stager downloads the file and injects it into memory. The values listed in this transaction are customizing the HTTP communication for downloading the beacon.
- **dns-beacon:** After Cobalt Strike v4.3, DNS options became part of the dns-beacon transaction. This transaction modifies the DNS C2 communication. If you are interested in a more comprehensive list of all the dns-beacon options, refer to this [Cobalt Strike user guide](#).
- **http-get:** The http-get transaction customizes the HTTP communication between the Beacon and the team server. The Beacon starts by sending the HTTP request with metadata about the compromised system. If the team server has tasks to execute, the server sends an HTTP response.
- **http-post:** Once the Beacon executes the tasks sent by the server, the output of the task is transferred in the http-post transaction. The values listed in this transaction affect the HTTP communication when the task output is sent over to the server.
- **https-certificate:** If the Beacon is tasked to communicate over HTTPS, The team server generates a self-signed certificate. The team server uses http-get and http-post transaction values to create actual HTTP requests and responses. This profile transaction can help to specify the different parameters for SSL certificates. If you are interested in a more comprehensive list of all the http-certificates options, refer to this [Cobalt Strike user guide](#).

```
# define indicators for an HTTP GET
http-get {
  # Beacon will randomly choose from this pool of URIs
  set uri "/ca /dpixel /__utm.gif /pixel.gif /g.pixel /dot.gif /updates.rss /fwlink /cm /cx /pixel /match

  client {
    # base64 encode session metadata and store it in the Cookie header.
    metadata {
      base64;
      header "Cookie";
    }
  }

  server {
    # server should send output with no changes
    header "Content-Type" "application/octet-stream";

    output {
      print;
    }
  }
}

# define indicators for an HTTP POST
http-post {
  # Same as above, Beacon will randomly choose from this pool of URIs [if multiple URIs are provided]
  set uri "/submit.php";

  client {
    header "Content-Type" "application/octet-stream";

    # transmit our session identifier as /submit.php?id=[identifier]
    id {
      parameter "id";
    }

    # post our output with no real changes
    output {
      print;
    }
  }

  # The server's response to our HTTP POST
  server {
    header "Content-Type" "text/html";

    # this will just print an empty string, meh...
    output {
      print;
    }
  }
}
```

Figure 1. Cobalt Strike default profile.

Cobalt Strike Default Profile

The default profile will be loaded if no other customized profiles are specified. Figure 1, above, is the specification of the default profile, and Figure 2, below, is an example of traffic capture from the default profile using the web drive-by-download option in a Cobalt Strike team server.

No.	Time	Source	Destination	Protocol	Length	Info
395	907.305207	10.3.228.11	10.3.228.192	HTTP	442	GET /j.ad HTTP/1.1
397	907.311682	10.3.228.192	10.3.228.11	HTTP	168	HTTP/1.1 200 OK
405	967.316716	10.3.228.11	10.3.228.192	HTTP	442	GET /j.ad HTTP/1.1
407	967.334143	10.3.228.192	10.3.228.11	HTTP	168	HTTP/1.1 200 OK
415	1027.345661	10.3.228.11	10.3.228.192	HTTP	442	GET /j.ad HTTP/1.1
418	1027.353083	10.3.228.192	10.3.228.11	HTTP	102	HTTP/1.1 200 OK
426	1027.371752	10.3.228.11	10.3.228.192	HTTP	1030	POST /submit.php?id=30067106 HTTP/1.1
430	1027.383516	10.3.228.192	10.3.228.11	HTTP	153	HTTP/1.1 200 OK
436	1087.393898	10.3.228.11	10.3.228.192	HTTP	442	GET /j.ad HTTP/1.1
438	1087.401324	10.3.228.192	10.3.228.11	HTTP	168	HTTP/1.1 200 OK
447	1147.403008	10.3.228.11	10.3.228.192	HTTP	442	GET /j.ad HTTP/1.1
450	1147.408793	10.3.228.192	10.3.228.11	HTTP	102	HTTP/1.1 200 OK
456	1147.409843	10.3.228.11	10.3.228.192	HTTP	375	POST /submit.php?id=30067106 HTTP/1.1
458	1147.412493	10.3.228.192	10.3.228.11	HTTP	153	HTTP/1.1 200 OK

Figure 2. An example traffic capture from the default profile.

From Figure 2, you can see that there are several HTTP transactions of GET and POST requests and responses.

- For GET requests, most of the request URIs are very short and have predefined patterns. The URIs are randomly chosen from the list of URIs specified under set uri in the default profile in Figure 1 (see Table 1 below for the complete list). Malicious attackers can easily modify the URI to arbitrary strings if they use a customized profile with set uri options inside the http-get section. This also explains why a pattern-based signature might catch the Cobalt Strike traffic using default profiles very well, but fail to capture any variations with customized profiles.
- For POST requests, there is a predefined pattern – /submit.php?id= – in the URI. The ID value is randomly generated. Similar to the possibilities for HTTP GET requests, malicious attackers can easily modify the URIs to arbitrary strings if they use customized profiles with set uri options inside the http-post section.

Index	URIs	Index	URIs	Index	URIs
1	/ca	8	/fwlink	15	/push
2	/dpxel	9	/cm	16	/ptj
3	/__utm.gif	10	/cx	17	/j.ad
4	/pixel.gif	11	/pixel	18	/ga.js
5	/g.pixel	12	/match	19	/en_US/all.js
6	/dot.gif	13	/visit.js	20	/activity
7	/updates.rss	14	/load	21	/IE9CompatViewList.xml

Table 1. Possible URIs specified in the Cobalt Strike default profile.

Customized Cobalt Strike Profiles

Public Malleable C2 profiles are available and can be downloaded in public repositories, such as from the official profiles examples on [GitHub](#). These profiles can be loaded by the team server and used as a Beacon download for C2 communications.

As an example, we walk through the [etumbot profile](#) to explain in more detail below.

1. Global Options.

- Sleeptime: The sleep time for the beacon callback is 5,000 milliseconds (5s).
- Jitter: The jitter to set % is 0. In this example, the Beacon will call back every 5s because of the jitter value 0.
- Maxdns: The maximum length of hostname is 255 when uploading data over DNS.

- UserAgent: Set the HTTP C2 request useragent as "Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/5.0)"

```
set sleeptime "5000";
set jitter "0";
set maxdns "255";
set useragent "Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/5.0)";
```

Figure 3. Global options in Etumbot profile.

2. Beacon check-In to get task from teamserver with HTTP GET request.

Below the global options, we find the following option configurations about HTTP request and response. Figures 4 and 5, below, show this configuration, which include URI, header and metadata information for both the client and the server.

```
http-get {
  set uri "/image/";
  client {
    header "Accept" "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8";
    header "Referer" "http://www.google.com";
    header "Pragma" "no-cache";
    header "Cache-Control" "no-cache";
    metadata {
      netbios;
      append "-.jpg";
      uri-append;
    }
  }
  server {
    header "Content-Type" "img/jpg";
    header "Server" "Microsoft-IIS/6.0";
    header "X-Powered-By" "ASP.NET";
    output {
      base64;
      print;
    }
  }
}
```

Figure 4. HTTP GET request options in Etumbot profile.



Figure 5. HTTP GET request in live traffic.

3. Beacon task execution result submission to teamserver with HTTP POST request.

We can find the following option configuration about HTTP response from Figure 6 below, as well as what the POST C2 traffic looks like in Figure 7.

```
http-post {
  set uri "/history/";
  client {
    header "Content-Type" "application/octet-stream";
    header "Referer" "http://www.google.com";
    header "Pragma" "no-cache";
    header "Cache-Control" "no-cache";

    id {
      netbiosu;
      append ".asp";
      uri-append;
    }

    output {
      base64;
      print;
    }
  }

  server {
    header "Content-Type" "img/jpg";
    header "Server" "Microsoft-IIS/6.0";
    header "X-Powered-By" "ASP.NET";

    output {
      base64;
      print;
    }
  }
}
```

Figure 6. HTTP POST request options in Etumbot profile.



Figure 7. HTTP POST request options in live traffic.

Cases in the Wild

The following sections show two different cases of Cobalt Strike payloads used in the wild: one using the default option (no profiles) and the other with a custom profile. Both samples have no trigger on VirusTotal at the time of this writing, but Palo Alto Networks identified them using static and dynamic analysis.

Default Profile Sample

SHA256 Hash: 6a6e5d2faeded086c3a97e14994d663e2ff768cb3ad1f5a1aa2a2b5fd344dde2

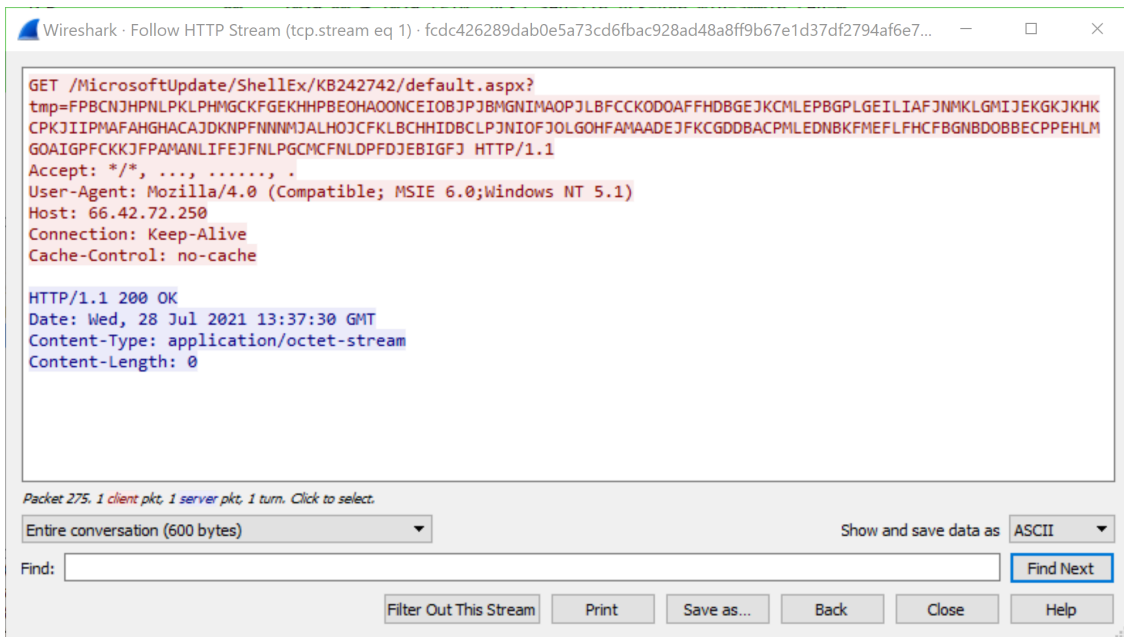


Figure 12. Cobalt Strike HTTP GET heartbeat request.

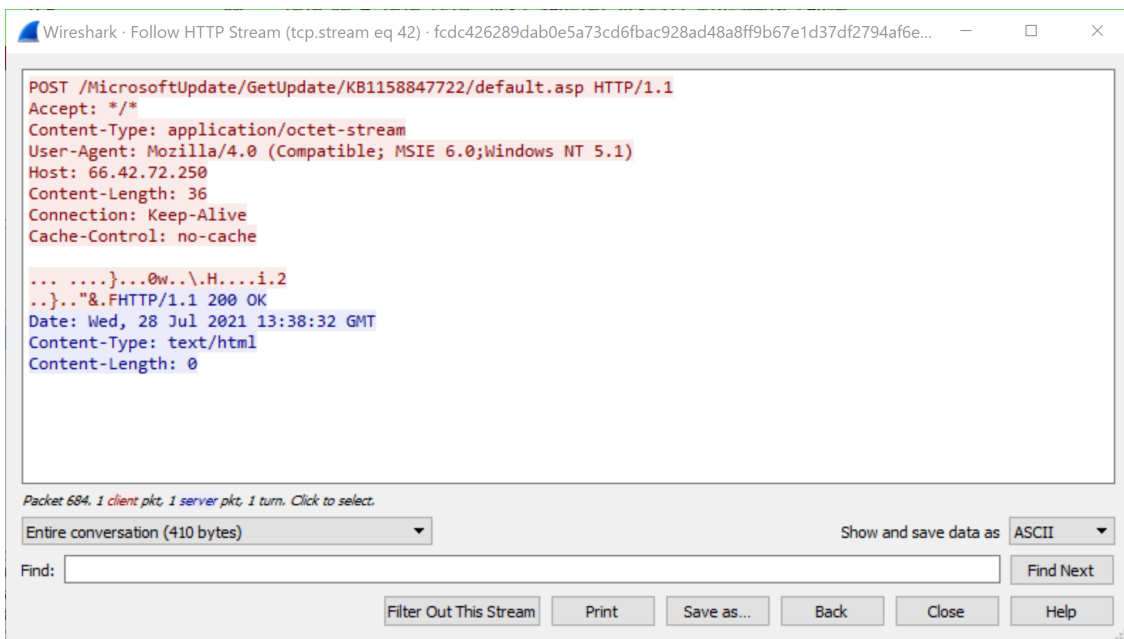


Figure 13. Cobalt Strike HTTP POST call-back request.

As we can see in Figures 12 and 13, the GET and POST request URIs have changed from the default profile. Both of these URIs are prepended with /MicrosoftUpdate in order to seem like a legitimate HTTP request to Microsoft servers for regular Windows updates – but are actually request and response traffic from C2 servers. This is how Cobalt Strike traffic from customized profiles can be so flexible and difficult to detect.

Cobalt Strike Beacon Configuration

In addition to the differences in GET and POST request parameters mentioned previously, Cobalt Strike Beacon configuration differs between default and custom profiles, and it contains useful metadata according to the settings in a Malleable C2 profile, which includes encoding types, blog submission mechanisms, instructions used to

<pre> application/octet-stream Build SessionId: [7:SessionId,5:id] Parameter id Build Output: [7:Output,4] Print </pre>	<pre> 0x000d http_post_header 0x0003 0x0100 Const_header Content-Type: application/octet-stream Const_header User-Agent: Mozilla/4.0 (Compatible; MSIE 6.0;Windows NT 5.1) Build SessionId: [7:SessionId,1:/default.asp,12] Append /default.asp Uri_append Build Output: [7:Output,4] Print </pre>
---	--

Table 2. Default profile vs custom Profile configuration meta-data.

Conclusion

Cobalt Strike is a potent post-exploitation adversary emulator. The Malleable C2 profile detailed above is elaborate and is designed to evade security detections. A single security appliance is not equipped to prevent a Cobalt Strike attack. Only a combination of security solutions – firewalls, sandboxes, endpoints and software to integrate all these components can help prevent this kind of attack.

Palo Alto Networks customers are protected from this kind of attack by the following:

1. [Next-Generation Firewalls](#) (NGFWs) with [Threat Prevention](#) signatures 86445 and 86446 identify HTTP C2 requests with default profiles.
2. [WildFire](#), an NGFW security subscription identifies and blocks Cobalt Strike Beacon.
3. [AutoFocus](#) users can track this activity using the [CobaltStrike](#) tags

Indicators of Compromise

CS Samples

- 6a6e5d2faeded086c3a97e14994d663e2ff768cb3ad1f5a1aa2a2b5fd344dde2
- fcdc426289dab0e5a73cd6fbac928ad48a8ff9b67e1d37df2794af6e7fa559e9

CS Beacon Samples

- /Iya9
 - 08e901d4ed0b43b46e632158f5ec5e900f16015e18995a875f62903a3c1eb1f9
- /api/1
 - d8b385d680bcdf7646f35df612712f7a3991f50a21cac8379630d05b3d2337ae

CS Team Server Domain

- www.symantecav[.]xyz

CS Team Server IP addresses

- 66.42.72[.]250
- 146.0.77[.]110

Source: <https://unit42.paloaltonetworks.com/cobalt-strike-malleable-c2-profile/>