Attackers Exploiting Public Cobalt Strike Profiles

By Durgesh Sangvikar, Yanhui Jia, Chris Navarrete, Matthew Tennis

Published: 2024-06-26 · Archived: 2026-04-06 02:59:04 UTC

Executive Summary

In this article, Unit 42 researchers detail recent findings of malicious Cobalt Strike infrastructure. We also share examples of malicious Cobalt Strike samples that use Malleable C2 configuration profiles derived from the same profile hosted on a public code repository.

Cobalt Strike is a commercial software framework that enables security professionals like red team members to simulate attackers embedding themselves in a network environment. However, threat actors continue to use cracked versions of Cobalt Strike in real-world attacks. The post-exploitation payload called Beacon uses text-based profiles called Malleable C2 to change the characteristics of Beacon's web traffic in an attempt to avoid detection.

Despite its use in defensive cybersecurity assessments, threat actors continue to leverage Cobalt Strike for malicious purposes. Due to its malleable and evasive nature, Cobalt Strike remains a significant security threat to organizations.

Palo Alto Networks customers are better protected from Cobalt Strike Beacon and Team Server C2 communication in the following ways:

- The Next-Generation Firewall (NGFW) with an Advanced Threat Prevention subscription can help identify and block Cobalt Strike HTTP C2 requests generated by custom profiles and block Cobalt Strike HTTP C2 requests.
- WildFire, Cortex XDR and Prisma Cloud can help identify and block Cobalt Strike Beacon binaries, and XDR will report related exploitation attempts.
- Cortex XSOAR response pack and playbook can help automate the mitigation process.
- Malicious URLs and IPs related to this research have been added to Advanced URL Filtering.

If you think you might have been compromised or have an urgent matter, contact the Unit 42 Incident Response team.

Related Unit 42 Topics	**Cobalt Strike**, **Malleable C2 Profile**

From Server to Beacon to Profile

Unit 42 has multiple techniques to find Cobalt Strike servers hosted on the internet, some of which we have documented in a previous article about Cobalt Strike analysis. The traffic flow and detection in this article were triggered by our Advanced Threat Prevention (ATP) solution.

After finding these Cobalt Strike servers, we pivoted on this information to discover any associated Beacon files. Our investigation of these samples revealed Malleable C2 profiles, which are described in another previous article about Malleable C2 profiles.

Our research also revealed that these Malleable C2 profiles borrow heavily from a single example hosted on a publicly available software repository.

First Sample

This first Beacon sample borrows from a Malleable C2 profile named ocsp.profile hosted on a publicly available software repository. This profile itself is not malicious, and it is one of many hosted on publicly available repositories that attackers can copy and alter for their own malicious purposes.

First sample SHA256 hash:

- 1980becd2152f4c29dffbb9dc113524a78f8246d3ba57384caf1738142bb3a07

We downloaded this Beacon sample from one of the Cobalt Strike servers discovered by our ATP solution. Attackers typically retrieve Beacon instances from Cobalt Strike servers and load Beacon into memory through some other compromised process. Embedded in this Beacon binary are details from its Malleable C2 profile.

We used Didier Stevens' Python script 1768.py to extract the Malleable C2 profile details. These details are listed below in Table 1.

Profile Component Description	Details
GET Request to get the command to execute	Method: GET Cobalt Strike C2 domains: • msupdate.azurefd[.]net • o365updater.azureedge[.]net • gupdater.bbtecno[.]com • teamsupd.azurewebsites[.]net • msdn1357.centralus.cloudapp.azure[.]com • cupdater.bbtecno[.]com URI: /ocsp/ Header: User-Agent: Microsoft-CryptoAPI/7.0
Post Request to return the command execution result	Method: POST URI: /ocsp/a/

Table 1. Extracted network information from the profile of our first Beacon sample.

Below, Figure 1 shows part of the results from Stevens' Python script analysis of our first Beacon sample. This section contains information related to the sample's Malleable C2 profile configuration. As noted in the http_get_header section, metadata of the victim is encoded using lowercase NetBIOS encoding and appended to the request URI. This configuration also adds Accept: */* to the HTTP GET request header.

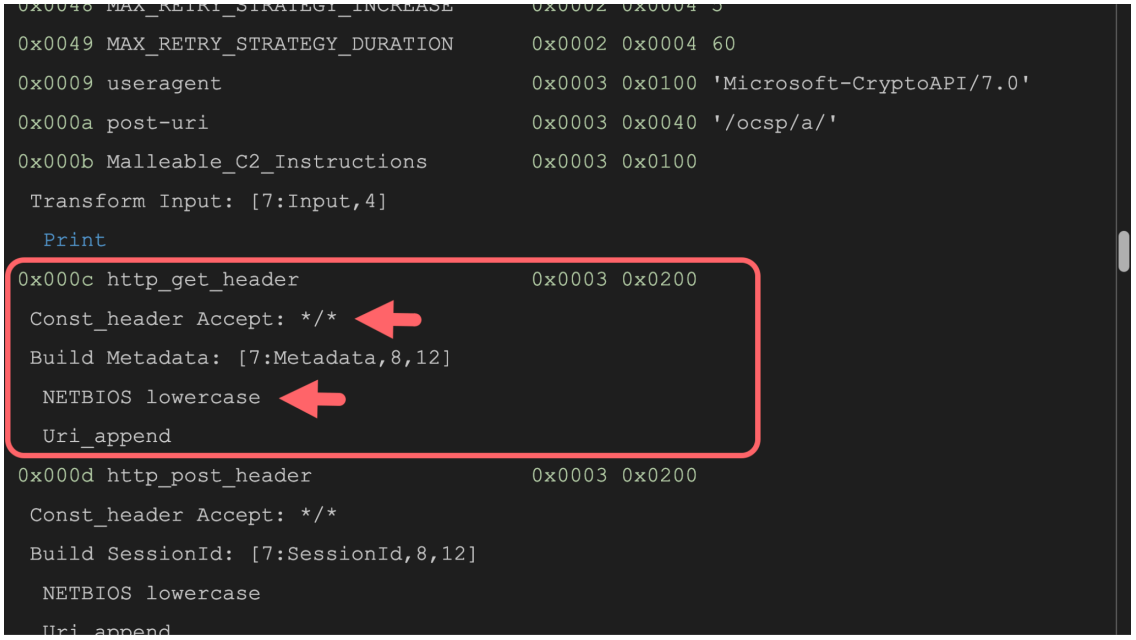

Figure 1. Output from running Stevens' 1768.py script on our first Beacon sample.

Figure 2 shows a TCP stream of the HTTP C2 traffic between this Beacon instance and the Cobalt Strike server. In it, we can see the lower-case NetBIOS encoding in the GET request as specified by the Malleable C2 profile.

```
GET /ocsp/idbiomddhpbjdgglmmefcdihhhmhapldplpepbcnjbfjenlpnddijlbfojmogpappb
kaghpbnbgdmnnibdoomdonpmonbdkcknlfepipnnhnfghngbibiohkcceleocaccfopenfgngipe
nnpckhkpgplhlkknpbkjhkeppfaaelhjcjcofndldnebmnpmhbpemlcenolhdfcjlomkaiiomcpi
pncmajdmfjjpbghpfnibgkbfhcfjimijmfohgc HTTP/1.1
Accept: */*
Cache-Control: no-cache
Host: [redacted]
Connection: Keep-Alive
User-Agent: Microsoft-CryptoAPI/7.0

HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 64

vv....=....Jc.....'$......Vc.@_.V..GP.....T.R..uc...AX.Cu.......
```

Figure 2. TCP stream of HTTP C2 traffic generated by our first Cobalt Strike Beacon sample.

This profile configuration appears to be based on the ocsp.profile from a publicly accessible software repository. The attackers merely replaced /oscp/ with /ocsp/ for both HTTP request methods and changed the User-Agent string from Microsoft-CryptoAPI/6.1 to Microsoft-CryptoAPI/7.0. Figure 3 indicates values from the original Malleable C2 profile that were altered for this Beacon sample. The rest of the profile used for this sample matches the original ocsp.profile content.

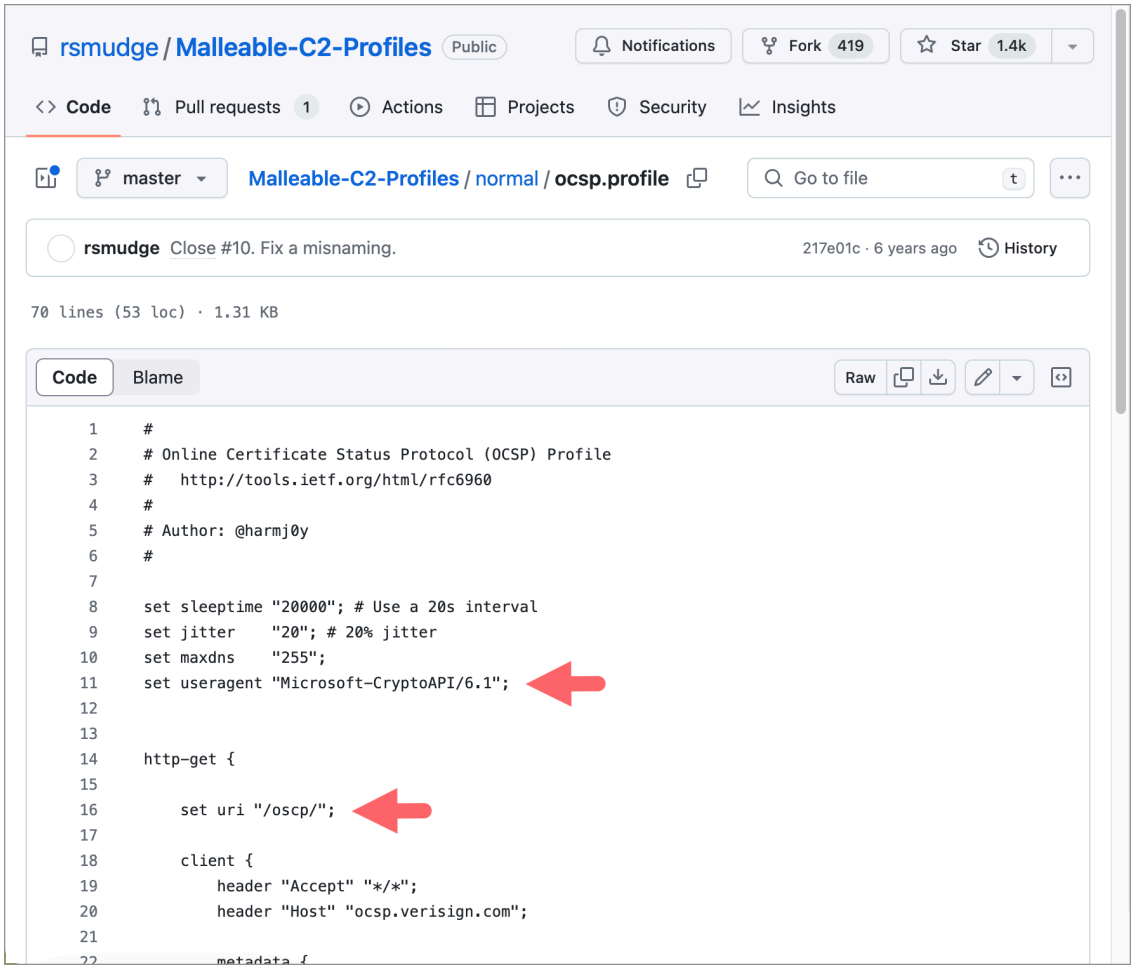

Figure 3. The original ocsp.profile, indicating the values updated in our first Beacon sample.

Second Sample

The Malleable C2 profile of our second Beacon sample borrows from the same ocsp.profile as our first sample.

Second sample SHA256 hash:

- b587e215ce8c0b3a1525f136fe38bfdc0232300e1a4f7e651e5dc6e86313e941

Like our first example, this Beacon sample is a staged binary hosted by a Cobalt Strike server that our ATP platform detected and downloaded. Following the same analysis procedure, we extracted the Malleable C2 profile information using 1768.py and compared the results with our repository of known profiles. Table 2 shows the network information we extracted from this profile.

Profile Component Description	Details
C2: GET Request to get the command to execute	Method: GET Cobalt Strike C2 domains: • msupdate.brazilsouth.cloudapp.azure[.]com • msdn1357.centralus.cloudapp.azure[.]com

	• update37.eastus.cloudapp.azure[.]com • update.westus.cloudapp.azure[.]com • 146.235.52[.]69 • 159.112.177[.]137 URI: /download/ Header: User-Agent: Microsoft-CryptoAPI/8.1
C2: Post Request to return the command execution result	Method: POST URI: /pkg/a/

Table 2. Extracted network information from the profile of our second Beacon sample.

In this Beacon sample, the attackers updated the URI path replacing the original ocsp.profile value of the HTTP GET request from /oscp/ to /download/. Attackers also replaced the original value of the HTTP POST request from /oscp/a/ to /pkg/a/. Finally they updated the User-Agent value from Microsoft-CryptoAPI/6.1 to Microsoft-CryptoAPI/8.1.

Figure 4 shows a TCP stream of the HTTP C2 traffic between this second Beacon instance and its Cobalt Strike server.

```
GET /download/ccanadfmfghnpnkijagflhaicnicjkmlanldhijkmbjekhlambcbpobjbjnodd
ohakhfhfmpiknc jmnnaaefeefgolhnfdpldaaffnoebmiaponikjekoohmgdcloeodkcakgpbccn
fijmncdekagchddeaokcdladjpcejjbpcjgjgemlkbmifdamcpmjgkcjkpodahkjljkmfkbmemoc
hmnakgpkmdbaddllpolgndfmalgplhooiecmpokodh HTTP/1.1
Accept: */*
Host: [redacted]
Connection: Keep-Alive
Cache-Control: no-cache
User-Agent: Microsoft-CryptoAPI/8.1

HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 64

..A.....J..vR..%1.'.t........~.....v.;1.p.7..,.....?./........w.
```

Figure 4. TCP stream of HTTP C2 traffic generated by our second Cobalt Strike Beacon sample.

Third Sample

The Malleable C2 profile of our third sample borrows from the same ocsp.profile as our first and second samples.

Third sample SHA256 hash:

- 38eeb82dbb5285ff6a2122a065cd1f820438b88a02057f4e31a1e1e5339feb2b

This third Cobalt Strike sample is a stageless 64-bit Windows executable file that uses the same ocsp.profile for its Malleable C2 profile, but with a twist. The domain for its C2 server contains a string in the leading subdomain that matches the FQDN of a well-known multinational technology company.

This FQDN for the Cobalt Strike C2 server is www.consumershop.lenovo.com.cn.d4e97cc6.cdnhwcggk22[.]com, however the parent domain is actually cdnhwcggk22[.]com.

Figure 5 shows an example of HTTP C2 traffic generated by this third sample, starting after a DNS query of the C2 domain resolves to the server's IP address.

No.	Time	Source	Destination	Protocol	Length	Info
…	32.245…	192.168.2.…	8.8.8.8	DNS	115	Standard query 0x478a A www.consumershop.lenovo.com.cn.d4e97cc6.cdnhwcggk22.com
…	33.247…	192.168.2.…	8.8.4.4	DNS	115	Standard query 0x478a A www.consumershop.lenovo.com.cn.d4e97cc6.cdnhwcggk22.com
…	33.485…	8.8.4.4	192.168.2.…	DNS	355	Standard query response 0x478a A www.consumershop.lenovo.com.cn.d4e97cc6.cdnhwcggk22.com A
…	33.580…	8.8.8.8	192.168.2.…	DNS	355	Standard query response 0x478a A www.consumershop.lenovo.com.cn.d4e97cc6.cdnhwcggk22.com A
…	33.692…	192.168.2.…	119.188.49…	HTTP	470	GET /oscp/bhlefpfmhpkecikhoacoglhhnjijidfblekojfhhelpjkgkonehfklmhpbljcjpljmompojbckhcojlhc
…	34.052…	119.188.49…	192.168.2.…	HTTP	599	HTTP/1.1 200 OK
…	34.216…	192.168.2.…	119.188.49…	HTTP	470	GET /oscp/bhlefpfmhpkecikhoacoglhhnjijidfblekojfhhelpjkgkonehfklmhpbljcjpljmompojbckhcojlhc
…	34.544…	119.188.49…	192.168.2.…	HTTP	598	HTTP/1.1 200 OK
…	34.654…	192.168.2.…	119.188.49…	HTTP	470	GET /oscp/bhlefpfmhpkecikhoacoglhhnjijidfblekojfhhelpjkgkonehfklmhpbljcjpljmompojbckhcojlhc
…	34.992…	119.188.49…	192.168.2.…	HTTP	596	HTTP/1.1 200 OK
…	35.107…	192.168.2.…	119.188.49…	HTTP	470	GET /oscp/bhlefpfmhpkecikhoacoglhhnjijidfblekojfhhelpjkgkonehfklmhpbljcjpljmompojbckhcojlhc
…	35.409…	119.188.49…	192.168.2.…	HTT…	678	HTTP/1.1 200 OK , JSON (application/json)
…	36.113…	192.168.2.…	119.188.49…	HTTP	6200	POST /oscp/a/dbdedadfdbdddadfddda HTTP/1.1
…	36.358…	119.188.49…	192.168.2.…	HTTP	602	HTTP/1.1 200 OK
…	38.998…	192.168.2.…	119.188.49…	HTTP	470	GET /oscp/bhlefpfmhpkecikhoacoglhhnjijidfblekojfhhelpjkgkonehfklmhpbljcjpljmompojbckhcojlhc
…	39.328…	119.188.49…	192.168.2.…	HTT…	651	HTTP/1.1 200 OK , JSON (application/json)
…	42.310…	192.168.2.…	119.188.49…	HTTP	470	GET /oscp/bhlefpfmhpkecikhoacoglhhnjijidfblekojfhhelpjkgkonehfklmhpbljcjpljmompojbckhcojlhc
…	43.113…	119.188.49…	192.168.2.…	HTT…	2351	HTTP/1.1 200 OK , JSON (application/json)
…	43.121…	192.168.2.…	119.188.49…	HTTP	345	POST /oscp/a/dbdedadfdbdddadfddda HTTP/1.1
…	43.400…	119.188.49…	192.168.2.…	HTTP	596	HTTP/1.1 200 OK

Figure 5. Filtered in Wireshark, C2 traffic generated by our third Cobalt Strike sample.

Borrowing From Public Malleable C2 Profiles

Detections for Cobalt Strike activity that depend on patterns in network traffic from the HTTP request headers patterns are of limited value, since any variation of these patterns can cause the detection to fail. Some workarounds such as regular expression patterns can temporarily alleviate this evasion. However, attackers can trivially modify the Malleable C2 profile, creating a detection arms race where attackers remain one step ahead of conventional network security solutions. In these cases, the cost is imposed more heavily on the defender than the attacker.

Furthermore, attackers do not need to create a Malleable C2 profile from scratch. They can easily copy publicly available examples and modify various values to fit their needs. Our research indicates that attackers use slight modifications of these publicly available profiles for their Cobalt Strike activity in an effort to evade detection.

Conclusion

In the ever-evolving landscape of cybersecurity, attackers persist in finding new methods, like leveraging publicly available Malleable C2 profiles. This strategy enables attackers to initiate Cobalt Strike C2 communications with flexibility, frequently altering profiles to evade detection and sustain malicious activity. Such tactics underscore the dynamic nature of cyberthreats and the continuous need for adaptive and forward-thinking defense mechanisms.

Machine-learning based solutions like ATP are the best type of defensive countermeasures available for preventing highly evasive attacks and C2 like Cobalt Strike. Heuristic detections cannot cover the huge amount of permutations that the Malleable C2 framework can so readily provide.

The cost for network security false positives is skewed heavily against the defender, which is a vulnerability in security operations that attackers exploit to their benefit.

Adopting a machine-learning network security platform like ATP provides detection capabilities to counter these types of threats.

This commitment to advancing our technologies in response to these threats reaffirms our dedication to cybersecurity excellence and the safety of the digital community.

Palo Alto Networks Protection and Mitigation

Palo Alto Networks customers are better protected from Cobalt Strike through the following products:

- The Next-Generation Firewall (NGFW) with an Advanced Threat Prevention subscription can identify and block Cobalt Strike HTTP C2 requests generated by custom profiles and block Cobalt Strike HTTP C2 requests.
- Advanced WildFire, Cortex XDR and Prisma Cloud can identify and block Cobalt Strike Beacon binaries, and XDR will report related exploitation attempts.
- Cortex XSOAR response pack and playbook can automate the mitigation process.
- Malicious URLs and IPs have been added to Advanced URL Filtering.

If you think you might have been compromised or have an urgent matter, get in touch with the Unit 42 Incident Response team or call:

- North America Toll-Free: 866.486.4842 (866.4.UNIT42)
- EMEA: +31.20.299.3130
- APAC: +65.6983.8730
- Japan: +81.50.1790.0200

Palo Alto Networks has shared these findings with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

Indicators of Compromise

SHA256 Hashes for Cobalt Strike Samples:

- 1980becd2152f4c29dffbb9dc113524a78f8246d3ba57384caf1738142bb3a07
- B587e215ce8c0b3a1525f136fe38bfdc0232300e1a4f7e651e5dc6e86313e941
- 38eeb82dbb5285ff6a2122a065cd1f820438b88a02057f4e31a1e1e5339feb2b

Domains and IP Addresses Used for Cobalt Strike C2:

- msupdate.azurefd[.]net
- o365updater.azureedge[.]net
- gupdater.bbtecno[.]com

- teamsupd.azurewebsites[.]net
- msdn1357.centralus.cloudapp.azure[.]com
- cupdater.bbtecno[.]com
- msupdate.brazilsouth.cloudapp.azure[.]com
- msdn1357.centralus.cloudapp.azure[.]com
- update37.eastus.cloudapp.azure[.]com
- update.westus.cloudapp.azure[.]com
- www.consumershop.lenovo.com.cn.d4e97cc6.cdnhwcggk22[.]com
- 146.235.52[.]69
- 159.112.177[.]137

Additional Resources

- Cobalt Strike Training Resources – Fortra
- Cobalt Strike User Guide: Malleable Command and Control – Fortra
- Cobalt Strike Analysis and Tutorial: How Malleable C2 Profiles Make Cobalt Strike Difficult to Detect – Palo Alto Networks, Unit 42
- Cobalt Strike Analysis and Tutorial: CS Metadata Encoding and Decoding – Palo Alto Networks, Unit 42
- Cobalt Strike Attack Detection & Defense Technology Overview – Palo Alto Networks, LIVEcommunity
- Detecting Popular Cobalt Strike Malleable C2 Profile Techniques – Palo Alto Networks, Unit 42
- Cobalt Strike: Using Known Private Keys To Decrypt Traffic, Part 1 – NVISO Labs

---