## Cobalt Strike: Using Known Private Keys To Decrypt Traffic – Part 1

blog.nviso.eu/2021/10/21/cobalt-strike-using-known-private-keys-to-decrypt-traffic-part-1/

October 21, 2021



Blogpost series: <u>Cobalt Strike: Decrypting Traffic</u> We found 6 private keys for rogue Cobalt Strike software, enabling C2 network traffic decryption.

The communication between a Cobalt Strike beacon (client) and a Cobalt Strike team server (C2) is encrypted with AES (even when it takes place over HTTPS). The AES key is generated by the beacon, and communicated to the C2 using an encrypted metadata blob (a cookie, by default).

RSA encryption is used to <u>encrypt this metadata</u>: the beacon has the public key of the C2, and the C2 has the private key.

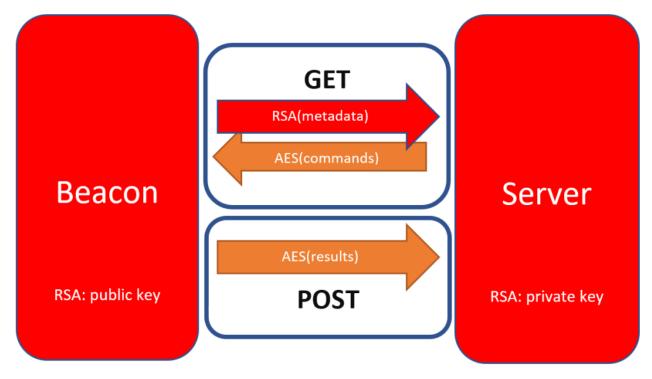


Figure 1: C2 traffic

Public and private keys are stored in file .cobaltstrike.beacon\_keys. These keys are generated when the Cobalt Strike team server software is used for the first time.

During our fingerprinting of Internet facing Cobalt Strike servers, we found public keys that are used by many different servers. This implies that they use the same private key, thus that their .cobaltstrike.beacon\_keys file is shared.

One possible explanation we verified: are there cracked versions of Cobalt Strike, used by malicious actors, that include a .cobaltstrike.beacon\_keys? This file is not part of a legitimate Cobalt Strike package, as it is generated at first time use.

Searching through VirusTotal, we found 10 cracked Cobalt Strike packages: ZIP files containing a file named .cobaltstrike.beacon\_keys. Out of these 10 packages, we extracted 6 unique RSA key pairs.

2 of these pairs are prevalent on the Internet: 25% of the Cobalt Strike servers we fingerprinted (1500+) use one of these 2 key pairs.

This key information is now included in tool <u>1768.py</u>, a tool developed by Didier Stevens to extract configurations of Cobalt Strike beacons.

Whenever a public key is extracted with known private key, the tool highlights this:

0x0002 port         0x0001 0x0002 8080           0x0003 sleeptime         0x0002 0x0004 60000           0x0004 maxgetsize         0x0002 0x0004 104857           0x0005 jitter         0x0001 0x0002 0           0x0007 publickey         0x0003 0x0100 308194           5f1fbblc18646c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae12205a         9523d71f08a56cae0315bfde3d6c8a16386b03b7a6551aa1336d50325a35           5f82361302030100010000000000000000000000000	<pre>^ lows-beacon_http-reverse_http</pre>
File: EbHm.vir         xorkey((chain): 0x94f45fff         length: 0x032e0033         Config found: xorkey b'.' 0x00000000 0x00002fef         0x0001 payload type       0x0001 0x0002 0 wind         0x0002 port       0x0001 0x0002 8080         0x0003 sleeptime       0x0002 0x0004 60000         0x0005 jitter       0x0001 0x0002 0x0004 104857         0x0007 publickey       0x0003 0x0100 308194         551fbb1c18646c377e03016b162b12ba72bdf7dc56b4cd2e4e9bae12205a         0523d71f08a56cae0315bfde3d6c8a16386b03b7a6551aa1336d50325a33         5582361302030100010000000000000000000000000	lows-beacon_http-reverse_http
xorkey(chain): 0x94f45fff         length: 0x032e0033         Config found: xorkey b'.' 0x00000000 0x00002fef         0x0001 payload type       0x0001 0x0002 0 wind         0x0002 port       0x0001 0x0002 8080         0x0003 sleeptime       0x0002 0x0004 60000         0x0005 jitter       0x0001 0x0002 0         0x0007 publickey       0x0001 0x0002 0         0x110846c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae122055         0x20008 server,get-uri       0x0003 0x0100 192.25         0x0008 server,get-uri       0x0003 0x0010 (NULL         0x0003 0x0010 (NULL       0x0003 0x0010 (NULL         0x0003 0x0010 (NULL       0x0003 0x0010 (XULL)	lows-beacon_http-reverse_http
Length:       0x0032e0033         Config found:       xorkey b'.'       0x0000000       0x00002fef         0x0001       payload type       0x0001       0x0001       0x0002 0       wind         0x0001       payload type       0x0001       0x0002 0       wind         0x0001       payload type       0x0001       0x0002 0       wind         0x0001       payload type       0x0001       0x0002 0       0x0002       0x0004       60000         0x0005       sleeptime       0x0002       0x0001       0x0002 0       00001       0x0002 0         0x0005       jitter       0x0001       0x0002 0       0x0003       0x1004       308194         5f1fbb1c18646c377e03016b162b12ba72bdf7dc36b4c42e4e9bae122055       0x0003       0x0100       308294         5f82361302030100010000000000000000000000000	lows-beacon_http-reverse_http
Config found: xorkey b'.' 0x00000000 0x00002fef           0x0001 payload type         0x0001 0x0002 0 wind           0x0002 port         0x0001 0x0002 0 wind           0x0003 sleeptime         0x0002 0x0004 60000           0x0005 jitter         0x0001 0x0002 0x0004 104857           0x0007 publickey         0x0001 0x0002 0           0x1f6846c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae122053         0x0003 0x0100 308194           0x523d71f08a56cae0315bfde3d6c8a16386b03b7a6551aa133d50325a33         0x0003 0x0100 12025a33           0x0008 server,get-uri         0x0003 0x0100 19225a33           0x0003 0x0010 19225a33         0x0003 0x0010 19225a33	lows-beacon_http-reverse_http
0x0001 payload type         0x0001 0x0002 0 wind           0x0002 port         0x0001 0x0002 8080           0x0003 sleeptime         0x0002 0x0004 60000           0x0005 jitter         0x0001 0x0002 0x0004 104857           0x0007 publickey         0x0003 0x0100 0x0002 0           0x0007 publickey         0x0003 0x0100 0x0100 20           0x23017803016b162b12ba72bdf7dc36b4cd224e9bae12205         0x2301768356cae0315bfde3d6c8a16386b03b7a6551aa1336d50325a33           0x0008 server,get-uri         0x0003 0x0100 '192.2           0x0008 server,get-uri         0x0003 0x0100 '192.2           0x0008 server,get.uri         0x0003 0x0100 '192.4           0x0003 0x0010 '192.2         0x0003 0x0010 '192.4	lows-beacon_http-reverse_http
0x0002 port         0x0001 0x0002 8080           0x0003 sleeptime         0x0002 0x0004 60000           0x0004 maxgetsize         0x0001 0x0002 0x0004 104857           0x0005 jitter         0x0001 0x0002 0           0x0007 publickey         0x0001 0x0003 08194           5511fbb1c18646c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae12205a         9523d71f08a56cae0315bfde3d6c8a16386b03b7a6551aa1336d50325a35           5f8236130203010e010e00e0e0e0e0e0e0e0e0e0e0e0e0e0e0	
ØX0003         sleeptime         ØX0002         ØX0004         60004           0x0004         maxgetsize         0x0002         0x0004         104857           0x0005         jitter         0x0001         0x0002         0           0x0007         publickey         0x0003         0x0100         308194           5f1fbb1c18646c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae122655         9523d71f08a56cae0315bfde3d6c8a16386b03b7a6551aa1336d50325a35         5f82361302030100010000000000000000000000000	
0x0004         maxgetsize         0x0002         0x0004         104857           0x0005         jitter         0x0001         0x0002         0           0x0007         publickey         0x0003         0x1003         308194           5f1fbb1c18646c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae122058         9523d71f08a56cae0315bfde3d6c8a16386b03b7a6551aa1336d50325a33         5f82361302030100010000000000000000000000000	
Øx0005 jitter         Øx0001 0x0002 0           0x0007 publickey         0x0003 0x0100 308194           5f1fbb1c18646c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae12205a         9523d7168a56cae0315bfde3d6c8a16386b03b7a6551aa133d5d5325a33           5f23d7168a56cae0315bfde3d6c8a16386b03b7a6551aa133d5d5325a33         9523d7168a56cae0315bfde3d6c8a16386b03b7a6551aa133d5d50325a33           5f82361302030100010000000000000000000000000	6
0x0007         0x0003         0x0100         308194           5f1fbb1c18646c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae12205a         39523d71f08a56cae0315bfde3d6c8a16386b03b7a6551aa1336d50325a35         356236130203010001000000000000000000000000	
5f1fbb1c18646c377e03016b162b12ba72bdf7dc36b4cd2e4e9bae12205a           9523d71f08a56cae0315bfde3d6c8a16386b03b7a6551aa1336d50325a3           5f82361302030100010000000000000000000000000	
5f82361302030100010000000000000000000000000	195c26170bf908105ad7fa4bbccfa798632261bed9870f975f20794e1fe49
000000000000000000000000000000000000	;00db27d78ad8fd13b6a73b9fb7c3fb4d7a088e323f07618656ecd83595fa
0x0008 server,get-uri         0x0003 0x0100 '192.2           0x000e SpawnTo         0x0003 0x0010 (NULL           0x001d spawnto_x86         0x0003 0x0040 '%wind	00000000000000000000000000000000000000
0x000e SpawnTo 0x0003 0x0010 (NULL 0x001d spawnto_x86 0x0003 0x0040 '%wind	00000000000000000000000000000000000000
0x001d spawnto_x86 0x0003 0x0040 '%wind	254.79.71,/ptj'
0x001e spawnto x64 0x0003 0x0040 '%wind	lir%\\syswow64\\rundll32.exe'
	lir%\\sysnative\\rundll32.exe'
0x001f CryptoScheme 0x0001 0x0002 0	
0x001a get-verb 0x0003 0x0010 'GET'	
0x001b post-verb 0x0003 0x0010 'POST'	
0x001c HttpPostChunk 0x0002 0x0004 0	
	al or pirated? - Stats uniques -≻ ips/hostnames: 380 publicke
ys: 218	
0x0026 bStageCleanup 0x0001 0x0002 0	
0x0027 bCFGCaution 0x0001 0x0002 0	
	la/5.0 (compatible; MSIE 9.0; Windows NT 6.0; WOW64; Trident
/5.0)'	de shat
0x000a post-uri 0x0003 0x0040 '/subm 0x000b Malleable C2 Instructions 0x0003 0x0100 '\x00\	
	X00 \X00 \X04
0x000c http_get_header 0x0003 0x0200 Cookie	
0x000d http_post_header 0x0003 0x0200	
&Content-Type: application/octet-stream	
acontencerype. apprication/occec-scream	

Figure 2: 1768.py extracting configuration from beacon

At minimum, this information is further confirmation that the sample came from a rogue Cobalt Strike server (and not a red team server).

Using option verbose, the private key is also displayed.

@NVISO_Labs			×
			_
@NVISO_Labs C:\Demo≻1768.py -V File: EbHm.vir	EDHM.VIr		
xorkey(chain): 0x94f45fff			
length: 0x032e0033			
Config found: xorkey b'.' 0x00	000000 0x00002fef		
0x0001 payload type	0x0001 0x0002 0 windows-beacon http-reverse http		
0x0002 port	0x0001 0x0002 8080		
x0003 sleeptime	0x0002 0x0004 60000		
x0004 maxgetsize	0x0002 0x0004 1048576		
0x0005 jitter	0x0001 0x0002 0		
0x0007 publickey	0x0003 0x0100 30819f300d06092a864886f70d010101050003818d0030818902818	100a738	cde7
f1fbb1c18646c377e03016b162b12	pa72bdf7dc36b4cd2e4e9bae12205a95c26170bf908105ad7fa4bbccfa798632261bed9870f975f	20794e1	fe49
523d71f08a56cae0315bfde3d6c8a	l6386b03b7a6551aa1336d50325a3500db27d78ad8fd13b6a73b9fb7c3fb4d7a088e323f076186	6ecd835	95fa
f8236130203010001000000000000	000000000000000000000000000000000000000	0000000	0000
	00000000000000000000000000000000000000		
	0500048202603082025c02010002818100a738cde75f1fbb1c18646c377e03016b162b12ba72bd		
	7fa4bbccfa798632261bed9870f975f20794e1fe499523d71f08a56cae0315bfde3d6c8a16386b6		
	73b9fb7c3fb4d7a088e323f07618656ecd83595fa5f823613020301000102818059eb70c54ee07		
	74b62e7ebf1e88b7692c0e7de44d8ce90bef60514c0b16ff5680c415aa026acaf80ab62f8f30698		
	0e20967435f74cac6703a201ec3431b86008e0d973fa775799bc7b8863037270a33829c081e6cba		
	06d054ec6e3b03f78f6bde89ef2d9e4b932b735fc264a5c7a3d2a7c66d00187c3dca51c3ef758		
	Ld5aca42fd808c1f7085c5ced215180360adabd6c553414fdf4ccd9b91d6995c35886636cb14e6f		
	7514bf16e29af7da2d33f1cf00eba1ee98afa3d6a7c858eaefa85b7f748b9da2ac2a2cd42db76e6		
	/102402d2e19e1e5ecebec773ea51717440af6eef7e9eda50889a3900303dabed7ab9939e4c62b8		
23471380948a7ec6e3a4672c4c42e 8bf6868fc3b4a5f8dd120f5e53813d	l3ea1824af3bab024100bd154581c5d399c838f476b75482886fcdf194d8419d6564a8dc8b9c074	e044359	519e
x0008 Server,get-uri	0x0003 0x0100 192.254.79.71,/µLj 0x0003 0x0010 (NULL)		
0x001d spawnto x86	0x0003 0x0040 '%windir%\\syswow64\\rundll32.exe'		
x001d spawnto_x80	0x0003 0x0040 '%windir%\\sysnative\\rundl132.exe'		
0x001f CryptoScheme	0x0001 0x0002 0		
0x001a get-verb	0x0003 0x0010 'GET'		
0x001b post-verb	0x0003 0x0010 'POST'		
0x001c HttpPostChunk	0x0002 0x0004 0		
0x0025 license-id	0x0002 0x0004 0 trial or pirated? - Stats uniques -> ips/hostnames: 3	80 publ	icke

Figure 3: using option verbose to display the private key

This can then be used to decrypt the metadata, and the C2 traffic (more on this later).

In International	_		×
@NVISO_Labs C:\Demo>strings.py -u -s cookie 2021-02-02-Hancitor-with-Ficker-Stealer-and-Cobalt-Strike-and	-NetSu	pport	-RAT
.pcap Cookie: d=0: n=UTC			
Cookie: KN9zfIq31DBBdLtF4JUjmrhm0lRKkC/I/zAiJ+Xxjz787h9yh35cRjEnXJAwQcWP4chXobXT/E5YrZjgreeGTrORnj//A5iZw	2TC1En	t++gL/	M∨MH
wgjsnvg9czGx6Ekpz0L1uEfkVoo4MpQ0/kJk9myZagRrPrFWdE9U7BwCz1E=		8	
Set-Cookie: ASPSESSIONIDQADBQTDR=AHHOAELAMBOOPFOANOAMCHFO; path=/			
	WP4chX	obXT/I	E5Yr
ZjgreeGTrORnj//A5iZw2TClEnt++gLMyMHwgjsnvg9czGx6Ekpz0L1uEfkVoo4MpQ0/kJk9myZagRrPrFWdE9U7BwCzlE=			
Encrypted metadata: KN9zfIq31DBBdLtF4JUjmrhm0lRKkC/I/zAiJ+Xxjz787h9yh35cRjEnXJAwQcWP4chXobXT/E5YrZjgreeGT	rORnj/	/A5iZı	v2TC
lEnt++gLMyMHwgjsnvg9czGx6Ekpz0L1uEfkVoo4MpQ0/kJk9myZagRrPrFWdE9U7BwCz1E=			
Decrypted:			
Header: 0000beef Datasize: 000005d			
Raw key: caeab4f452fe41182d504aa24966fbd0			
aeskey: 3342f45e6e2f7lf5975c998600b11471			
hmackey: 7142dd70f4ec320badac8ca246a9488f			
charset: 04e4 ANSI Latin 1; Western European (Windows)			
charset_oem: 01b5 OEM United States			
bid: 644d8e4 105175268			
pid: 1c7c 7292			
port: 0			
flags: 04 var1: 10			
var1. 10 var2: 0			
Var3: 19042			
var4: 0			
var5: 1988364896			
var6: 1988359504			
var7: 1862862346			
Field: b'DESKTOP-Q21RU7A			
Field: b'maxwell.carter'			
Field: b'svchost.exe'			
@NVISO Labs C:\Demo>			

Figure 4: decrypting metadata

In upcoming blog posts, we will show in detail how to use these private keys to decrypt metadata and decrypt C2 traffic.

## About the authors

Didier Stevens is a malware expert working for NVISO. Didier is a SANS Internet Storm Center senior handler and Microsoft MVP, and has developed numerous popular tools to assist with malware analysis. You can find Didier on <u>Twitter</u> and <u>LinkedIn</u>.

You can follow NVISO Labs on <u>Twitter</u> to stay up to date on all our future research and publications.

Series Navigation Cobalt Strike: Using Known Private Keys To Decrypt Traffic – Part 2 >>