考察知识点：从artifact内存dump 会话密钥。

```
File: artifact.exe_210821_020000.dmp
Position: 0x004f7fe6
Header: 0000beef
Datasize: 0000005e
Raw key:  2042cf56983791bf1b5ac9c93eaf9e72
 aeskey:  d832804430c3988daaa34bd1593d8db1
 hmackey: 138aad64ace14511b693282fab0f8f59
charset: 03a8 ANSI/OEM Simplified Chinese (PRC, Singapore); Chinese Simplified (GB2312)
charset_oem: 03a8 ANSI/OEM Simplified Chinese (PRC, Singapore); Chinese Simplified
(GB2312)
bid: 704726 7358246
pid: 0688 1672
port: 0
flags: 06
Field: b'6.2'
Field: b'192.168.144.1'
Field: b'LAPTOP-JG2LCT4M'
Field: b'\xb2\xbb\xba\xc3\xd2\xe2\xcb\xbc\xb3\xd0\xc8\xc3\xc1\xcb\xa3\xa1'
Field: b'artifact.exe'

AES key:
 Position: 0x004eb3c6

HMAC key:
 Position: 0x004eb3d6
```

```
'''
Beacon任务执行结果解密
'''
import hmac
import binascii
import base64
import struct
import hexdump
from Crypto.Cipher import AES

def compare_mac(mac, mac_verif):
  if mac == mac_verif:
    return True
  if len(mac) != len(mac_verif):
    print
    "invalid MAC size"
    return False

  result = 0

  for x, y in zip(mac, mac_verif):
    result |= x ^ y
```

```
  return result == 0

def decrypt(encrypted_data, iv_bytes, signature, shared_key, hmac_key):
  if not compare_mac(hmac.new(hmac_key, encrypted_data, digestmod="sha256").digest()
[0:16], signature):
    print("message authentication failed")
    return

  cypher = AES.new(shared_key, AES.MODE_CBC, iv_bytes)
  data = cypher.decrypt(encrypted_data)
  return data

#key源自Beacon_metadata_RSA_Decrypt.py
#keys =
b'\x83\x19\xba\x9a(>N\x9b37\x7f\x8f\x9ds\x01|N+\xc2\r@\xc5\xfd\xbe1\x16\xc1\x95|a_f'
# SHARED_KEY = keys[0:16]
# HMAC_KEY = keys[16:]

HMAC_KEY = binascii.unhexlify("138aad64ace14511b693282fab0f8f59")
SHARED_KEY = binascii.unhexlify("d832804430c3988daaa34bd1593d8db1")

#
encrypt_data="AAABUHLPP8OIRnEYEBQl2ihp3Iapmma17EpQxVQfsaXdnppNCC1ygXR22rRiEN/tB5xfyFjbt
X4OxQV6CmucYhobqC4wzeBAxX0IN7t1mBWGwz1/ICI9hi1/d1o8zW2F7s2VIj/GBzkNFcwjvEmLdmFcRiHBxllB
baJMeWP5+6Lyufk+/uN5Izo3Uze2abqcOFlcdU2LOkF9nGqy8gZW6yj6WScdPS3GhVMQAOVv7l60+U5s6DanZuM
2nm0tVtdp5m4ud9H089WMV2ELyO6Af6Xu/lxutprgpSGJXiDkZ+DEUFEkffK7j52L+psrInLs5lQDrRr22c7Fsn
0CbE0k6IMysZ3WGWXUFALtYNDgztbS3nuPe13zCh48ZDIJLqvDUSJh5uFfmULxbchOm+ylRahRo0dy7gDSL5u1C
H78dmos3Nf9CLdY2GmkXoBXVURinJnuZQ=="

encrypt_data=binascii.unhexlify("00000040867d84eab30db53600d3c9450cefe2ea582a912f61daec
080d1aff2a06bfd1054190e7addd95a6e84c1888dbdd0b0a0c501b9fc6240526b5e0759b28aa663028")
#encrypt_data=binascii.unhexlify("00000060f53d2d1e920a2116e92dc4774b7e0343a4f272da2111a
1509e79c0640a5b73f64f5c38960f3cc0e1a36ef9163a693c4d673f3361f409f59a3ac8900c2f935e923036
1ce9649b55362d88f29c71b943c6cb530f4d26560321853abd0728841bed")


encrypt_data_length=encrypt_data[0:4]


encrypt_data_length=int.from_bytes(encrypt_data_length, byteorder='big', signed=False)

encrypt_data_l = encrypt_data[4:len(encrypt_data)]

data1=encrypt_data_l[0:encrypt_data_length-16]
signature=encrypt_data_l[encrypt_data_length-16:encrypt_data_length]
iv_bytes = bytes("abcdefghijklmnop",'utf-8')

dec=decrypt(data1,iv_bytes,signature,SHARED_KEY,HMAC_KEY)


counter = dec[0:4]
counter=int.from_bytes(counter, byteorder='big', signed=False)
print("counter:{}".format(counter))
```

```
dec_length = dec[4:8]
dec_length=int.from_bytes(dec_length, byteorder='big', signed=False)
print("任务返回长度:{}".format(dec_length))

de_data= dec[8:len(dec)]
Task_type=de_data[0:4]
Task_type=int.from_bytes(Task_type, byteorder='big', signed=False)
print("任务输出类型:{}".format(Task_type))
print(de_data[4:dec_length])
#print(de_data[4:dec_length].decode('utf-8'))

print(hexdump.hexdump(dec))
```

```
'''
cobaltstrike任务解密
'''
import hmac
import binascii
import base64
import struct

import hexdump
from Crypto.Cipher import AES

def compare_mac(mac, mac_verif):
  if mac == mac_verif:
    return True
  if len(mac) != len(mac_verif):
    print
    "invalid MAC size"
    return False

  result = 0

  for x, y in zip(mac, mac_verif):
    result |= x ^ y

  return result == 0


def decrypt(encrypted_data, iv_bytes, signature, shared_key, hmac_key):
  if not compare_mac(hmac.new(hmac_key, encrypted_data, digestmod="sha256").digest()
[0:16], signature):
    print("message authentication failed")
    return

  cypher = AES.new(shared_key, AES.MODE_CBC, iv_bytes)
  data = cypher.decrypt(encrypted_data)
  return data


def readInt(buf):
  return struct.unpack('>L', buf[0:4])[0]
```

```
#接收到的任务数据
#shell_whoami="fHH4M/hP5Q3CerzcyfbtahSrpZXWncshJWQkz7FtfLa/hwWxzy22mOpEsl9o1qP2"
#shell_whoami="g71/a5LzIsANC3PLbEWtFVbfhrRKdk2eNyx72usEeZR/YoGZ813EBzqBGZI/heGT"
BeaconTask="66550b25df33a195d29ee37fcf34f92216e23730c6fa5b2cc5f38e1fa0323122d96632d4d0e
477ce76c8a43598a1fb54e73cd4c8b461c58ac554477d748b4bd3ba7b1121e07ce596ac13fcb0f88d7e51"


if __name__ == "__main__":
  #key源自Beacon_metadata_RSA_Decrypt.py
  # SHARED_KEY = binascii.unhexlify("")
  # HMAC_KEY = binascii.unhexlify("")
  # keys =
b'\x83\x19\xba\x9a(>N\x9b37\x7f\x8f\x9ds\x01|N+\xc2\r@\xc5\xfd\xbe1\x16\xc1\x95|a_f'
  # SHARED_KEY = keys[0:16]
  # HMAC_KEY = keys[16:]

  HMAC_KEY = binascii.unhexlify("138aad64ace14511b693282fab0f8f59")
  SHARED_KEY = binascii.unhexlify("d832804430c3988daaa34bd1593d8db1")

  enc_data = binascii.unhexlify(BeaconTask)
  print("数据总长度:{}".format(len(enc_data)))
  signature = enc_data[-16:]
  encrypted_data = enc_data[:-16]

  iv_bytes = bytes("abcdefghijklmnop",'utf-8')

  dec = decrypt(encrypted_data,iv_bytes,signature,SHARED_KEY,HMAC_KEY)
  print(dec)
  counter = readInt(dec)
  print("时间戳:{}".format(counter))

  decrypted_length = readInt(dec[4:])
  print("任务数据包长度:{}".format(decrypted_length))

  data = dec[8:len(dec)]
  print("任务Data")
  print(hexdump.hexdump(data))

  # 任务标志
  Task_Sign=data[0:4]
  print("Task_Sign:{}".format(Task_Sign))

  # 实际的任务数据长度
  Task_file_len = int.from_bytes(data[4:8], byteorder='big', signed=False)
  print("Task_file:{}".format(Task_file_len))

  with open('data.bin', 'wb') as f:
    f.write(data[8:Task_file_len])

  print(hexdump.hexdump(data[Task_file_len:]))
```

解密会话密钥：

```
数据总长度:80
b'a\x1f\xeeQ\x00\x00\x000\x00\x00\x00N\x00\x00\x00(\x00\x00\x00\t%COMSPEC%\x00\x00\x00\
x15 /C type password.txt\x00\x00AAAAAAAA'
时间戳:1629482577
任务数据包长度:48
任务Data
00000000: 00 00 00 4E 00 00 00 28  00 00 00 09 25 43 4F 4D  ...N...(....%COM
00000010: 53 50 45 43 25 00 00 00  15 20 2F 43 20 74 79 70  SPEC%.... /C typ
00000020: 65 20 70 61 73 73 77 6F  72 64 2E 74 78 74 00 00  e password.txt..
00000030: 41 41 41 41 41 41 41 41                           AAAAAAAA
None
Task_Sign:b'\x00\x00\x00N'
Task_file:40
00000000: 72 64 2E 74 78 74 00 00  41 41 41 41 41 41 41 41  rd.txt..AAAAAAAA
None
[Finished in 0.3s]
```

```
counter:2
任务返回长度:23
任务输出类型:30
b'password is DDASCTF'
00000000: 00 00 00 02 00 00 00 17  00 00 00 1E 70 61 73 73  ............pass
00000010: 77 6F 72 64 20 69 73 20  44 44 41 53 43 54 46 36  word is DDASCTF6
00000020: 36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36  6666666666666666
None
```

```
数据总长度:80
b'a\x1f\xee\x8d\x00\x00\x00.\x00\x00\x00N\x00\x00\x00&\x00\x00\x00\t%COMSPEC%\x00\x00\x
00\x13 /C type secret.txt\x00\x00AAAAAAAAAA'
时间戳:1629482637
任务数据包长度:46
任务Data
00000000: 00 00 00 4E 00 00 00 26  00 00 00 09 25 43 4F 4D  ...N...&....%COM
00000010: 53 50 45 43 25 00 00 00  13 20 2F 43 20 74 79 70  SPEC%.... /C typ
00000020: 65 20 73 65 63 72 65 74  2E 74 78 74 00 00 41 41  e secret.txt..AA
00000030: 41 41 41 41 41 41 41 41                           AAAAAAAA
None
Task_Sign:b'\x00\x00\x00N'
Task_file:38
00000000: 65 74 2E 74 78 74 00 00  41 41 41 41 41 41 41 41  et.txt..AAAAAAAA
00000010: 41 41                                             AA
None
[Finished in 0.3s]
```

```
counter:3
任务返回长度:66
任务输出类型:30
b'Kl Tzkl nv bomt yqdj! DSUVYI{25ha430s7e4uje53gct3d20v4i9f1g16}'
00000000: 00 00 00 03 00 00 00 42  00 00 00 1E 4B 6C 20 54  .......B....Kl T
00000010: 7A 6B 6C 20 6E 76 20 62  6F 6D 74 20 79 71 64 6A  zkl nv bomt yqdj
00000020: 21 20 44 53 55 56 59 49  7B 32 35 68 61 34 33 30  ! DSUVYI{25ha430
00000030: 73 37 65 34 75 6A 65 35  33 67 63 74 33 64 32 30  s7e4uje53gct3d20
00000040: 76 34 69 39 66 31 67 31  36 7D 6F 00 00 00 00 00  v4i9f1g16}o.....
None
[Finished in 0.4s]
```

https://www.qqxiuzi.cn/bianma/weijiniyamima.php

Kl Tzkl nv bomt yqdj! DSUVYI{25ha430s7e4uje53gct3d20v4i9f1g16}

密钥 DDASCTF 加密 解密

Hi This is your flag! DASCTF{25ea430a7c4beb53dcb3b20c4d9c1d16}