CTF1

```
flag{1_l0v3_54n17y_ch3ck_ch4ll5}
```

CTF2

```
COMPFEST13{aha_gotcha_9437e8f141}
```

Step 1: Open the pcapng file, find that it is a live stream using RTMP protocol.

Protocol	Length	Info
RTMP	1581	Handshake C0+C1
RTMP	1580	Handshake C2
RTMP	1580	Handshake S0+S1+S2
RTMP	60	Set Chunk Size 4096
RTMP	235	connect('live')
RTMP	60	Window Acknowledgement Size 5000000
RTMP	61	Set Peer Bandwidth 5000000,Dynamic
RTMP	60	Set Chunk Size 4096
RTMP	246	_result('NetConnection.Connect.Success')
RTMP	85	releaseStream('test')
RTMP	81	FCPublish('test')
RTMP	77	createStream()
RTMP	85	_result()
RTMP	90	publish('test')
RTMP	161	onStatus('NetStream.Publish.Start')
RTMP	455	@setDataFrame()
RTMP	63	Audio Data
RTMP	113	Video Data
RTMP	78	Video Data
RTMP	122	Video Data
RTMP	60	Audio Data

So the main problem is to extract rtmp stream from this package.

Step 2: use a tool called rtmp2flv to extract the video.

Tool is here:quo/rtmp2flv: Extract FLV video from unencrypted RTMP streams. (github.com)

Use tcpflow to extract the TCP streams:

```
tcpflow -T %T_%A%C%c.rtmp -r rtmp.pcap
```

```
xinyingzheng@master:~/Desktop/rtmp2flv-master$ tcpflow -T %T_%A%C%c.rtmp -r capture.pcapng
xinyingzheng@master:~/Desktop/rtmp2flv-master$ ./rtmp2flv.py *.rtmp
[INFO] Reading from '2021-04-11T11:13:56Z_192.168.018.010c1.rtmp'
[DEBUG] Server uptime: 0d 0h 36m 22.332s, version: 0.0.0.0
[DEBUG] New chunk stream 2
[INFO] Set chunk size 4096
[DEBUG] New chunk stream 3
[INFO] Stream 0 AMF0 command: ['_result', 1.0, {'fmsVer': 'FMS/3,0,1,123', 'capabilities': 31.0}, {'le
vel': 'status', 'code': 'NetConnection.Connect.Success', 'description': 'Connection succeeded.', 'obje
ctEncoding': 0.0}]
[INFO] Stream 0 AMF0 command: ['_result', 4.0, None, 1.0]
[DEBUG] New chunk stream 5
```

Then I get some ***.rtmp** file

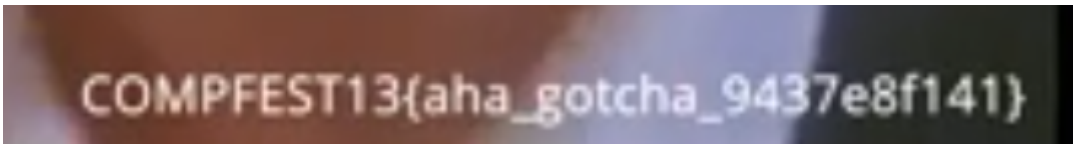
```
xinyingzheng@master:~/Desktop/rtmp2flv-master$ ls
2021-04-11T11:13:56Z_192.168.018.010c1.rtmp      capture.pcapng
2021-04-11T11:13:56Z_192.168.018.010.rtmp        README.md
2021-04-11T11:13:56Z_192.168.018.010.rtmp.1.flv  report.xml
2021-04-11T11:14:22Z_127.000.000.001c1.rtmp      rtmp2flv.py
2021-04-11T11:14:22Z_127.000.000.001.rtmp
```

Finally, convert the streams to FLV files:

```
./rtmp2flv.py *.rtmp
```

```
xinyingzheng@master:~/Desktop/rtmp2flv-master$ ls
2021-04-11T11:13:56Z_192.168.018.010c1.rtmp      capture.pcapng
2021-04-11T11:13:56Z_192.168.018.010.rtmp        README.md
2021-04-11T11:13:56Z_192.168.018.010.rtmp.1.flv  report.xml
2021-04-11T11:14:22Z_127.000.000.001c1.rtmp      rtmp2flv.py
2021-04-11T11:14:22Z_127.000.000.001.rtmp
```

Play it:



CTF3

```
flag{8bedfdbb-ba42-43d1-858c-c2a5-5012d309}
```

We get a memory file, so we first analyse it.

Step 1: Analyse the memory file.

```
sudo volatility -f memory imageinfo
```

```
xinyingzheng@master:~/Desktop$ sudo volatility -f memory imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/xinyingzheng/Desktop/memory)
                      PAE type : PAE
                           DTB : 0xa8f000L
                          KDBG : 0x80545ce0L
          Number of Processors : 1
     Image Type (Service Pack) : 2
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2021-08-06 16:43:57 UTC+0000
     Image local date and time : 2021-08-07 00:43:57 +0800
```

Step 2: check the cmd history.

```
sudo volatility -f memory --profile=WinXPSP2x86 cmdscan
```

```
**************************************************
CommandProcess: csrss.exe Pid: 620
CommandHistory: 0x3833938 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 16 LastAdded: 15 LastDisplayed: 15
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x404
Cmd #0 @ 0x3832110: It's useless to find so many things
Cmd #1 @ 0x3832ed0: ........................
Cmd #2 @ 0x52c778: what can i do about it
Cmd #3 @ 0x3833360: Heard that there is a one-click cleaning that is very useful
Cmd #4 @ 0x52b3c8: try it
Cmd #5 @ 0x52b7e8: "C:\Documents and Settings\Administrator\??\Oneclickcleanup.exe"
Cmd #6 @ 0x5224a0: what???
Cmd #7 @ 0x52d5c0: what happened??
Cmd #8 @ 0x52d410: who is 1cepeak?
Cmd #9 @ 0x3832de0: what's the meaning of hack?
Cmd #10 @ 0x3830e50: oh,no
Cmd #11 @ 0x52af40: holy shit
Cmd #12 @ 0x3830cf8: aaaaaa
Cmd #13 @ 0x522d28: Nonononononononononononono!!!!!!!!!!!!!!!
Cmd #14 @ 0x522d88: "C:\Documents and Settings\Administrator\??\Oneclickcleanup.exe"
Cmd #15 @ 0x5224b8: fuc
**************************************************
CommandProcess: csrss.exe Pid: 620
CommandHistory: 0x386b900 Application: DumpIt.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x520
Cmd #2 @ 0x700ff: ??????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
```

Find that user use a **Oneclickcleanup.exe** to clean his laptop.

Step 3: extract Oneclickcleanup.exe.

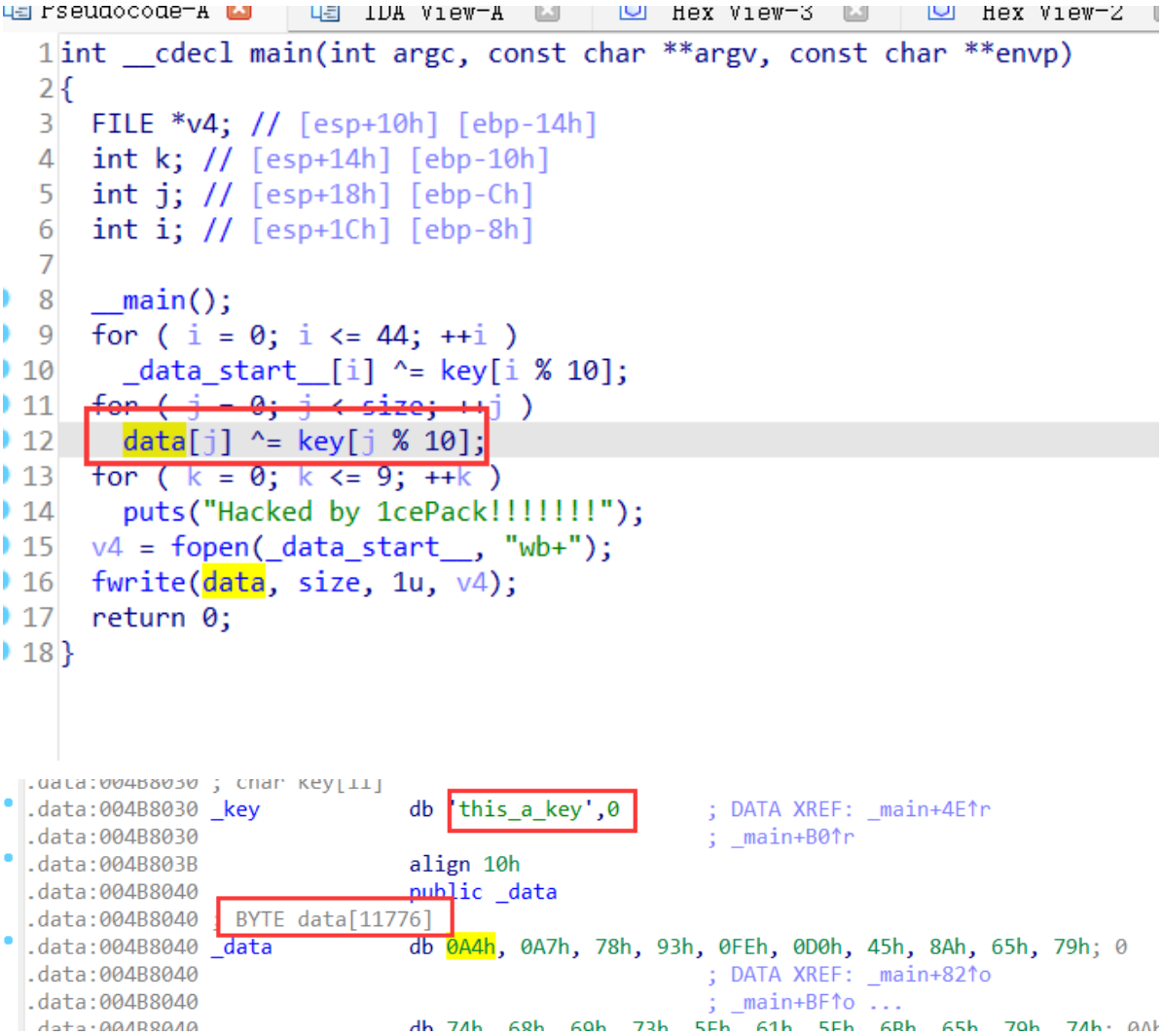
```
volatility -f memory --profile=WinXPSP2x86 filescan | grep Oneclickcleanup.exe
```

```
xinyingzheng@master:~/Desktop$ volatility -f memory --profile=WinXPSP2x86 filescan | grep Oneclickcleanup.exe
Volatility Foundation Volatility Framework 2.6
0x00000000017bcbc0      1      0 R--rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\Oneclickcleanup.exe
0x0000000000180c758      1      0 RW-rwd \Device\HarddiskVolume1\temp\Oneclickcleanup.exe
0x0000000001956d88      1      0 R--r-d \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\Oneclickcleanup.exe
xinyingzheng@master:~/Desktop$ sudo volatility -f memory --profile=WinXPSP2x86  dumpfiles -Q 0x00000000017bcbc0 -D ./
[sudo] password for xinyingzheng:
```

```
sudo volatility -f memory --profile=WinXPSP2x86  dumpfiles -Q 0x00000000017bcbc0
-D ./
```

```
xinyingzheng@master:~/Desktop$ sudo volatility -f memory --profile=WinXPSP2x86  dumpfiles -Q 0x00000000017bcbc0 -D ./
[sudo] password for xinyingzheng:
Volatility Foundation Volatility Framework 2.6
ImageSectionObject 0x017bcbc0   None   \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\Oneclickcleanup.exe
DataSectionObject 0x017bcbc0   None   \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\Oneclickcleanup.exe
xinyingzheng@master:~/Desktop$ ls
1688.dmp  a.txt  dire.txt  file.None.0x813bfdb0.img  file.None.0x81482a50.dat  memory  rtmp2flv-master  token.md
xinyingzheng@master:~/Desktop$ ^C
```

Step 4 : disassemble the .dat file



```
Pseudocode-A   IDA View-A   Hex View-3   Hex View-2

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3   FILE *v4; // [esp+10h] [ebp-14h]
4   int k; // [esp+14h] [ebp-10h]
5   int j; // [esp+18h] [ebp-Ch]
6   int i; // [esp+1Ch] [ebp-8h]
7
8   __main();
9   for ( i = 0; i <= 44; ++i )
10    _data_start__[i] ^= key[i % 10];
11  for ( j = 0; j < size; ++j )
12    data[j] ^= key[j % 10];
13  for ( k = 0; k <= 9; ++k )
14    puts("Hacked by 1cePack!!!!!!!");
15  v4 = fopen(_data_start__, "wb+");
16  fwrite(data, size, 1u, v4);
17  return 0;
18 }
```

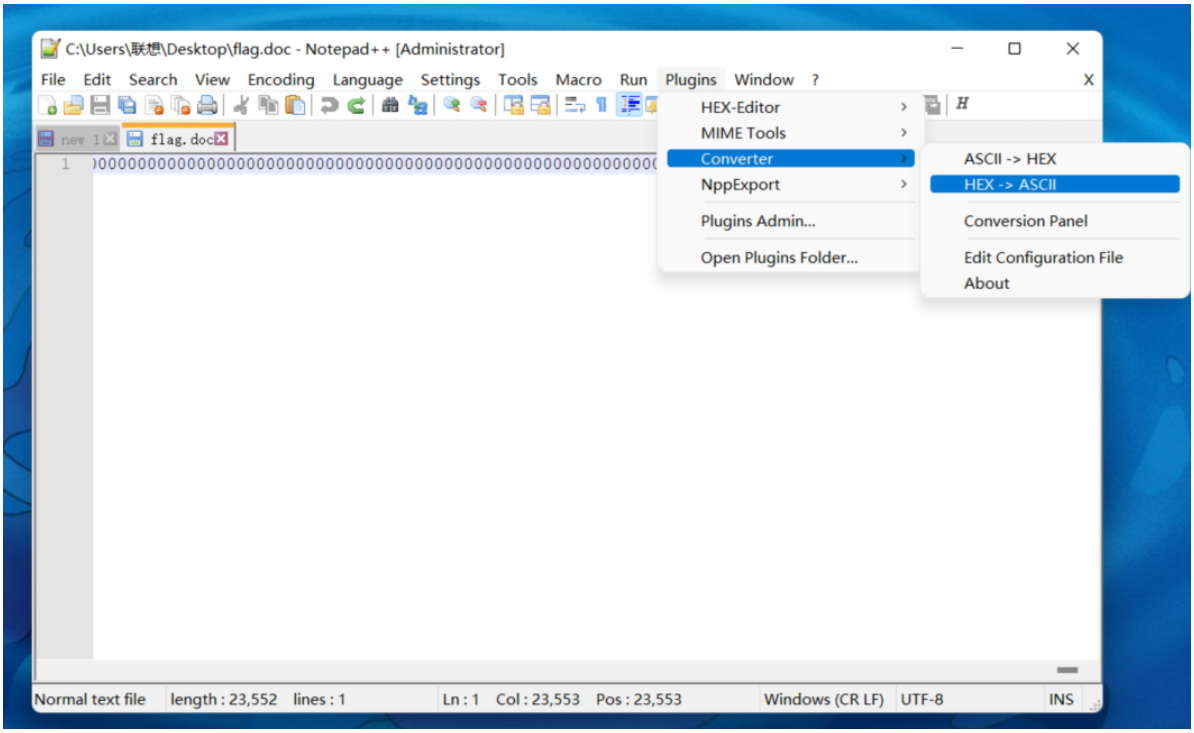
```
.data:004B8030 ; char key[11]
.data:004B8030 _key             db 'this_a_key',0       ; DATA XREF: _main+4E↑r
.data:004B8030                                          ; _main+B0↑r
.data:004B803B                  align 10h
.data:004B8040                  public _data
.data:004B8040 ; BYTE data[11776]
.data:004B8040 _data            db 0A4h, 0A7h, 78h, 93h, 0FEh, 0D0h, 45h, 8Ah, 65h, 79h; 0
.data:004B8040                                          ; DATA XREF: _main+82↑o
.data:004B8040                                          ; _main+BF↑o ...
.data:004B8040                  db 74h, 68h, 69h, 73h, 5Fh, 61h, 5Fh, 6Bh, 65h, 79h, 74h; 0Ah
```

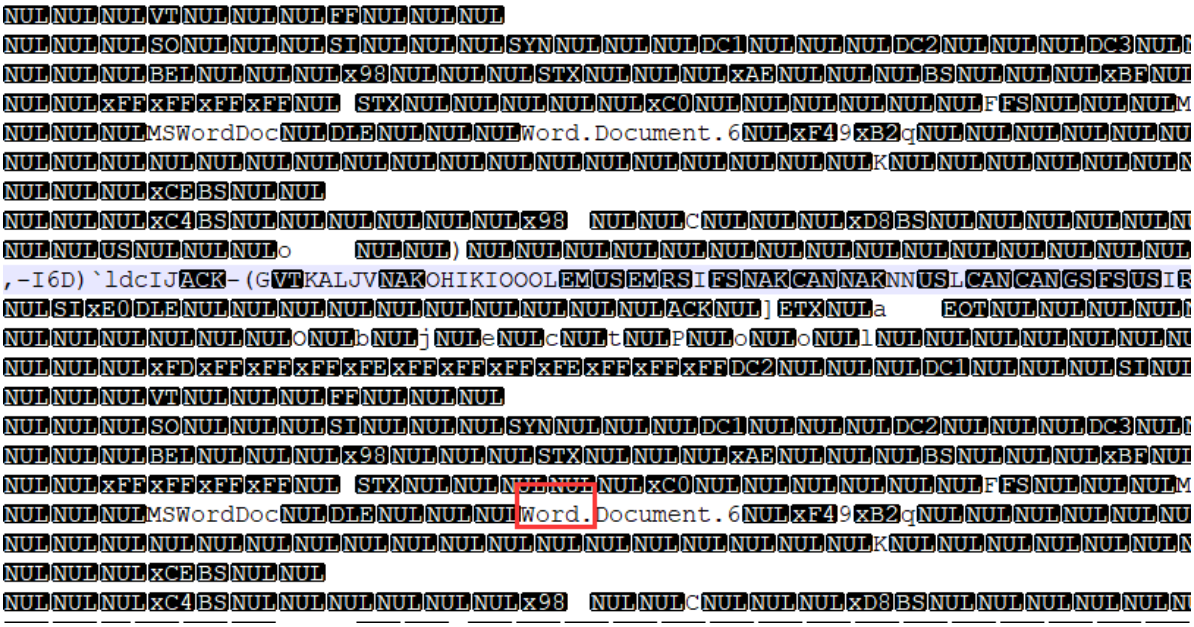
```
key='this_a_key'
data=[0A4h,0A7h.....]
```

Step 5: use a script tp manipulate it.

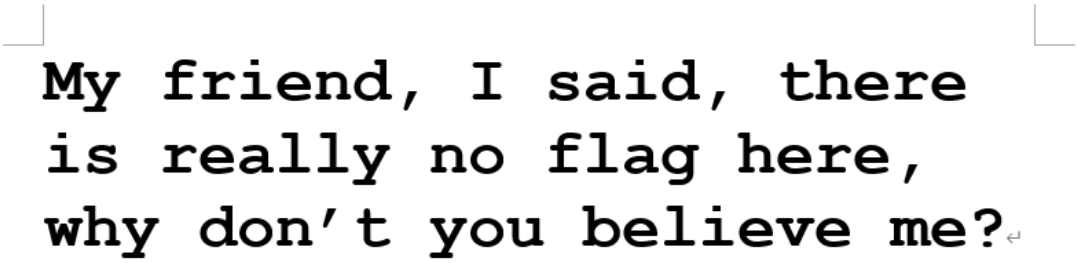
```
if __name__ == '__main__':
  s = 'this_a_key'
  flag = ''
  flags = [0xA4,0xA7,0x78,0x93,0xFE,0xD0,....,0x68,0x69,0x73,0x5F,0x61]
  for i in range(len(flags)):
      flag += str(hex(flags[i] ^ ord(s[i%10])))[2:].zfill(2)
  print(flag)
  f = open('flag.doc','wb')
  f.write(flag.encode())
```

It is a hex file, we use notepad++ to transfer it to ascii.





Then open use office word.



Step 6: according to the hint, do the xor brute directly.

```python
if __name__ == '__main__':
    f = open("flag.doc", "rb")
    for line in f:
        message=line
        for k in range(256):
            m3 = [x ^ k for x in message]
            m3 = bytes(m3)
            if (b'flag' in m3):
                print(m3.find(b'flag'))
                print("here")
                print(m3[m3.find(b'flag'):-1])
```

460
here
b'flag here, why don\x92t you believe me?\r,-I6D)`ldcIJ\x06-(G\x0bKALJV\x15OHIKIOOOL\x19\x1f\x1
514
here
b'flag{8bedfdbbba4243d1858cc2a55012d30-.--f.--\xab.---\xd3----------
354
here
b'flag{8bedfdbb-ba42-43d1-858c-c2a5-5012d309}\x06d\x1a\x1b----------
460
here
b'flag here, why don\x92t you believe me?\r,-I6D)`ldcIJ\x06-(G\x0bKALJV\x15OHIKIOOOL\x19\x1f\x1
514
here
b'flag{8bedfdbbba4243d1858cc2a55012d30-.--f.--\xab.--\xeb.---\xd3\xd3----------