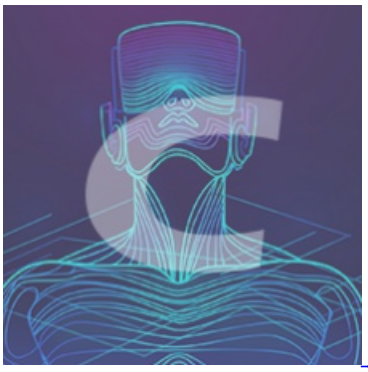
2019领航杯-恢复与解密

原创

天问_Herbert555 于 2019-11-27 20:30:04 发布 1779 收藏 1

分类专栏： # 比赛题目总结 文章标签： ctf 领航杯



恢复与解密

这道题做了很久都没做出来，下面是看别人的writeup写的，学到了很多知识。

题目描述：公安人员在犯罪分子的电脑中发现一些磁盘文件，但是发现关键信息已经被删除，现需要你对磁盘进行恢复，并对恢复出来的一些秘密文件里面的加密信息进行解密。 注意：通过strings获取到的yc4pl0fvjs2k1t7T为假flag，请尝试使用其他正确的做题方式获取flag。

一，附件是xty.img，在kali下进行挂载：

```
mount -o loop xty.img /mnt
```

第一步挂载就不知道是什么意思。。。百度一下。

- 命令解释。

-o： mount命令的一个参数，Options的首字母，后面跟着mount选项。

loop：用来把一个文件当成硬盘分区mount到目录。

/mnt： 挂载点。

- 什么是挂载？

Linux 系统中“一切皆文件”，所有文件都放置在以根目录为树根的树形目录结构中。在 Linux 看来，任何硬件设备也都是文件，它们各有自己的一套文件系统（文件目录结构）。

因此产生的问题是，当在 Linux 系统中使用这些硬件设备时，只有将Linux本身的文件目录与硬件设备的文件目录合二为一，硬件设备才能为我们所用。合二为一的过程称为“挂载”。

Linux中的根目录以外的文件要想被访问，需要将其“关联”到根目录下的某个目录来实现，这种关联操作就是“挂载”，这个目录就是“挂载点”，解除次关联关系的过程称之为“卸载”。

通俗理解就是把一个根目录之外的文件的顶级目录连接到Linux的一个根目录下，然后Linux就能直接对它进行操作了。

参考：什么是挂载？

```
root@thekali:~# mount -o loop xty.zip /mnt
root@thekali:~# cd /mnt
root@thekali:/mnt# ls -a
.  ..  .hide  lost+found  .ls
```

- ls -a 显示所有文件及目录。
- ls -A 同 -a ，但不列出 “.”(目前目录) 及 “..” (父目录)。

发现 .hide 目录，打开没有发现什么有用的。

二，然后要用到sstat 和 ext3grep这两个工具进行磁盘恢复。

用 cd --命令返回上一级目录。

首先用 fsstat 查看文件信息。

```
fsstat xty.img
```

```
root@thekali:~# fsstat xty.img
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: Ext3
Volume Name:
Volume ID: 809a1caaa174a48d5a4aeac6914601ca

Last Written at: 2019-11-27 18:21:20 (CST)
Last Checked at: 2014-02-07 09:28:31 (CST)

Last Mounted at: 2019-11-27 18:21:20 (CST)
Unmounted properly
Last mounted on: /root/CCSC

Source OS: Linux
Dynamic Structure
Compat Features: Journal, Ext Attributes, Resize Inode, Dir Index
InCompat Features: Filetype, Needs Recovery,
Read Only Compat Features: Sparse Super,

Journal ID: 00
Journal Inode: 8

METADATA INFORMATION
--------------------------------------------
Inode Range: 1 - 1281
Root Directory: 2
Free Inodes: 1266

CONTENT INFORMATION
--------------------------------------------
Block Range: 0 - 5119
Block Size: 1024
Reserved Blocks Before Block Groups: 1
Free Blocks: 3880

BLOCK GROUP INFORMATION
--------------------------------------------
```


这里的2是我们所需要的。

三，下载 ext3grep 工具。

```
apt -get install ext3grep
```

```
root@thekali:~# apt-get install ext3grep
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列【新】软件包将被安装：
  ext3grep
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 1171 个软件包未被升级。
需要下载 111 kB 的归档。
解压缩后会消耗 288 kB 的额外空间。
获取:1 http://mirrors.neusoft.edu.cn/kali kali-rolling/main amd64 ext3grep amd64 0.10.2-4 [111 kB]
已下载 111 kB，耗时 5秒 (22.3 kB/s)
正在选中未选择的软件包 ext3grep。
(正在读取数据库 ... 系统当前共安装有 353228 个文件和目录。)
准备解压 .../ext3grep_0.10.2-4_amd64.deb  ...
正在解压 ext3grep (0.10.2-4) ...
正在设置 ext3grep (0.10.2-4) ...
正在处理用于 man-db (2.8.6.1-1) 的触发器 ...
```

四，使用 ext3grep 工具，可以发现有secret文件。

```
ext3grep --inode 2 xty.img
```

```
root@thekali:~# ext3grep --inode 2 xty.img
Running ext3grep version 0.10.2
No --ls used; implying --print.

WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
WARNING: EXT3_FEATURE_INCOMPAT_RECOVER is set. This either means that
```

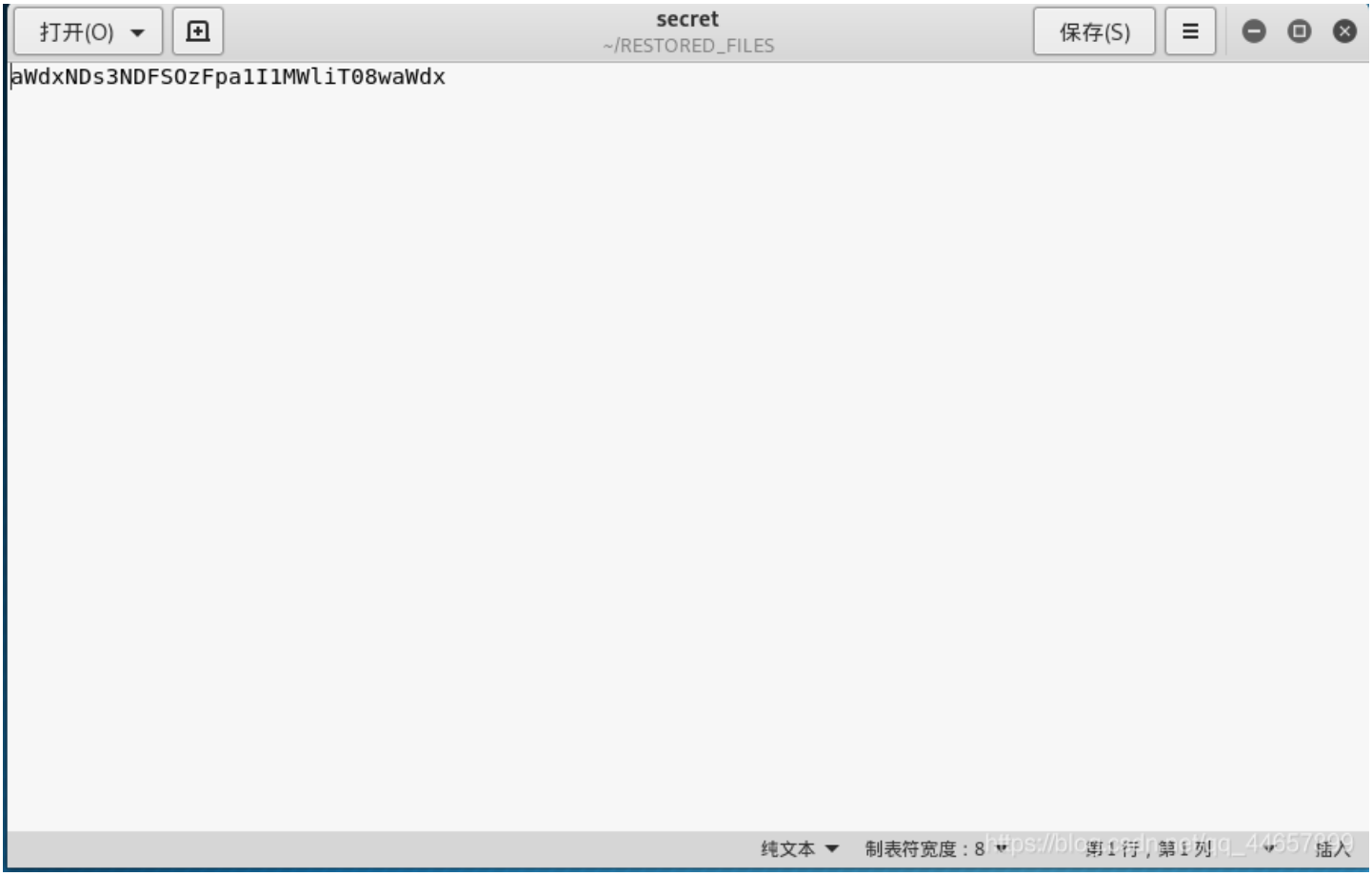
```
Directory block 184:
          .-- File type in dir_entry (r=regular file, d=directory, l=symlink)
          |          .-- D: Deleted ; R: Reallocated
 Indx Next |  Inode   | Deletion time                       Mode        File name
==========+==========+---------------data-from-inode------+-----------+=========
   0    1 d       2                                       drwxr-xr-x  .
   1    2 d       2                                       drwxr-xr-x  ..
   2    5 d      11                                       drwx------  lost+found
   3    5 r      12  D 1558974332 Tue May 28 00:25:32 2019  rrw-r--r--  0secret
   4    5 r      12  D 1558974332 Tue May 28 00:25:32 2019  rrw-r--r--  secret
   5    6 d      13                                       drwxr-xr-x  .hide
   6  end d      17                                       drwxr-xr-x  .ls
   7    8 r      16  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.1
   8    9 r      18  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.2
   9   10 r      19  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.3
  10   11 r      20  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.4
  11   12 r      21  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.5
  12   13 r      22  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.6
  13   14 r      23  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.7
  14   15 r      24  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.8
  15  end r      25  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.9
  16  end r      26  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.10
root@thekali:~# ^C
```

五，用命令把它恢复出来。

```
ext3grep --restore-file secret xty.img
```

```
root@thekali:~# ext3grep --restore-file secret xty.img
Running ext3grep version 0.10.2
WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
WARNING: EXT3_FEATURE_INCOMPAT_RECOVER is set. This either means that your partition is still mounted, and/or the file sys
Number of groups: 1
Minimum / maximum journal block: 198 / 1227
Loading journal descriptors... sorting... done
The oldest inode block that is still in the journal, appears to be from 1391737510 = Fri Feb  7 09:45:10 2014
Number of descriptors in journal: 244; min / max sequence numbers: 23 / 118
Loading xty.img.ext3grep.stage2... done
Restoring secret
```


打开可以看到恢复文件的内容。



六，base64解码一下,得到：

```
igq4;741R;1ikR51ibOO0igq
```

七，然后是异或解密：

```
import string

c = "igq4;741R;1ikR51ibOO0igq"

for i in range(0,200):
    p = ""
    for j in range(len(c)):
        p += chr(ord(c[j])^i)
    print p
```

```
>>> for i in range(0,200):
...     p = ""
...     for j in range(len(c)):
...         p += chr(ord(c[j])^i)
...     print p
...
igq4;741R;1ikR51ibOOOigq
hfp5:650S:0hjS40hcNN1hfp
kes69563P93kiP73k`MM2kes
jdr78472Q82jhQ62jaLL3jdr
mcu0?305V?5moV15mfKK4mcu
1bt1>214W>41nW041gJJ51bt
oaw2=127T=7omT37odII6oaw
n`v3<036U<6n1U26neHH7n`v
aoy<3?<9Z39acZ=9ajGG8aoy
```


拿到flag。