

0X00 前言

由于在反弹shell的过程中有一些非常精简的语句，但是一直没有深入理解，只是作为一个伸手党/搬运工，于是下定决心要将其弄明白，而这里面最难的也就是文件描述符和

0X01 文件描述符

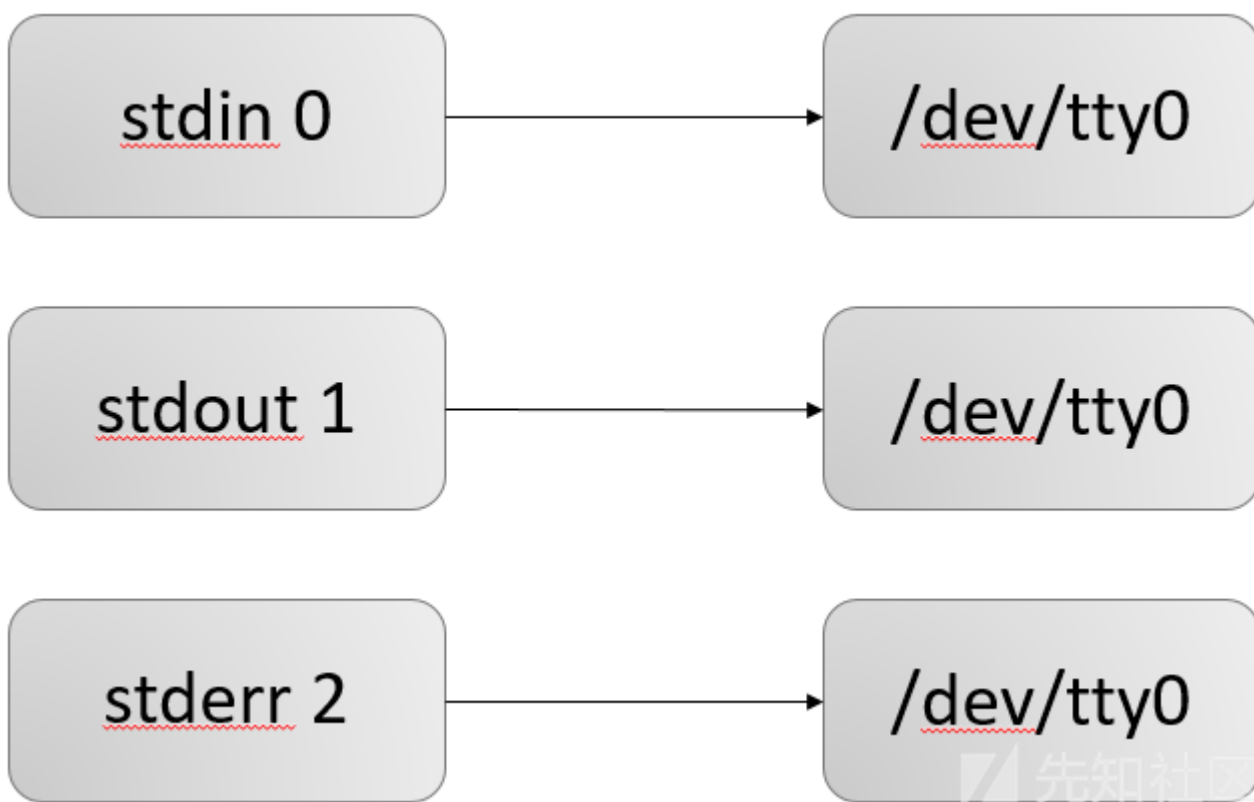
linux文件描述符：可以理解为linux跟踪打开文件，而分配的一个数字，这个数字有点类似c语言操作文件时候的句柄，通过句柄就可以实现文件的读写操作。

当Linux启动的时候会默认打开三个文件描述符，分别是：

标准输入standard input 0（默认设备键盘）

标准输出standard output 1（默认设备显示器）

错误输出：error output 2（默认设备显示器）



注意：

- （1）以后再打开文件，描述符可以依次增加
- （2）一条shell命令，都会继承其父进程的文件描述符，因此所有的shell命令，都会默认有三个文件描述符。

文件所有输入输出都是由该进程所有打开的文件描述符控制的。（Linux一切皆文件，就连键盘显示器设备都是文件，因此他们的输入输出也是由文件描述符控制）

一条命令执行以前先会按照默认的情况进行绑定（也就是上面所说的0,1,2），如果我们有时候需要让输出不显示在显示器上，而是输出到文件或者其他设备，那我们就需要重定向。

0X02 重定向

重定向主要分为两种(其他复杂的都是从这两种衍生而来的)：

- （1）输入重定向 < <<
- （2）输出重定向 > >>

重点：

1.bash

在执行一条指令的时候，首先会检查命令中存不存在重定向的符号，如果存在那么首先将文件描述符重定向（之前说过了，输入输出操作都是依赖文件描述符实现的，重定向

2.如果指令中存在多个重定向，那么不要随便改变顺序，因为重定向是从左向右解析的，改变顺序可能会带来完全不同的结果（这一点我们后面会展示）

3.< 是对标准输入 0 重定向，> 是对标准输出 1 重定向

4.再强调一下，重定向就是针对文件描述符的操作

1.输入重定向

格式：[n]< word（注意[n]与<之间没有空格）

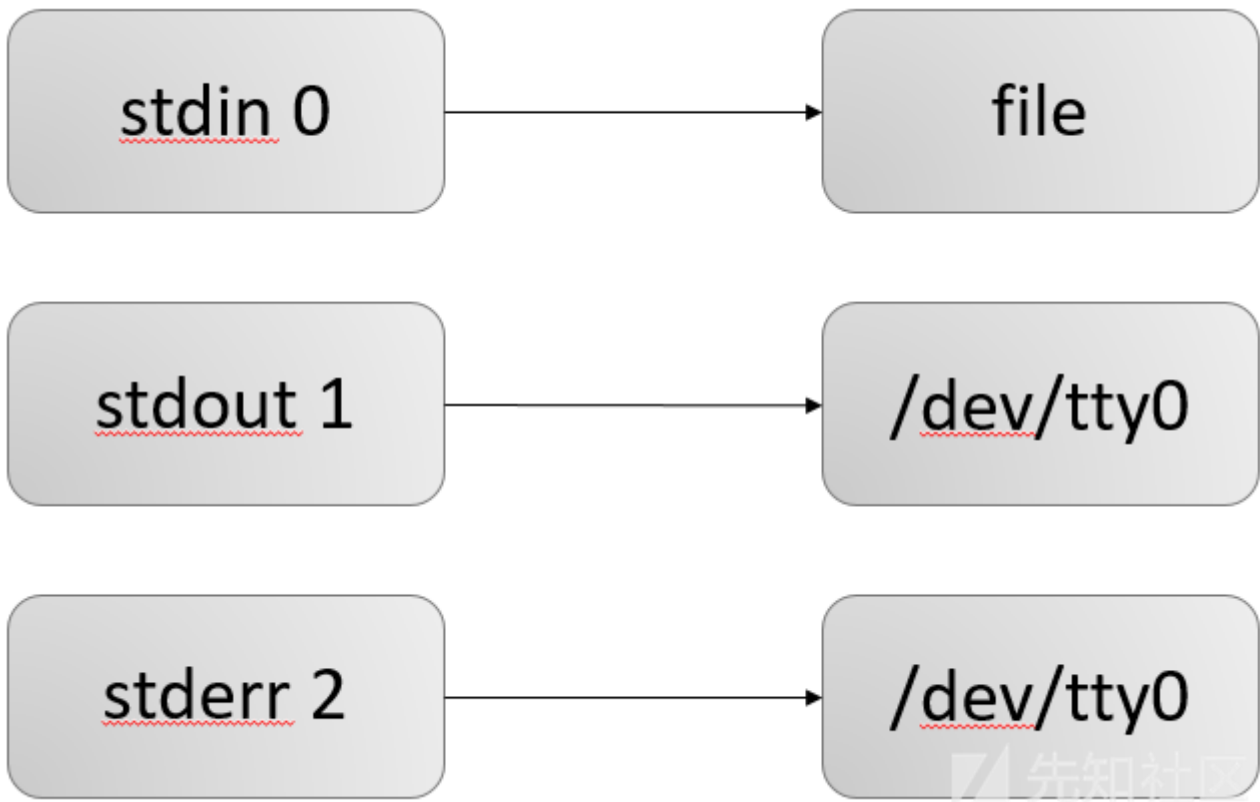
说明：将文件描述符 n 重定向到 word 指代的文件（以只读方式打开），如果n省略就是0（标准输入）

```
root@K0rz3n:~# cat 0< file
hello world
root@K0rz3n:~# cat < file
hello world
root@K0rz3n:~#
```

```
root@K0rz3n:~# 0< file cat
hello world
root@K0rz3n:~# < file cat
hello world
root@K0rz3n:~#
```

解释: 解析器解析到 "<" 以后会先处理重定向，将标准输入重定向到file，之后cat再从标准输入读取指令的时候，由于标准输入已经重定向到了file，于是cat就从file中读取指令了。（有没有觉得这个其实就是C语言中的指针或者文件句柄，就是将0这个指针指向了不同的地址，自然有不同的输入）

图示:



2.输出重定向

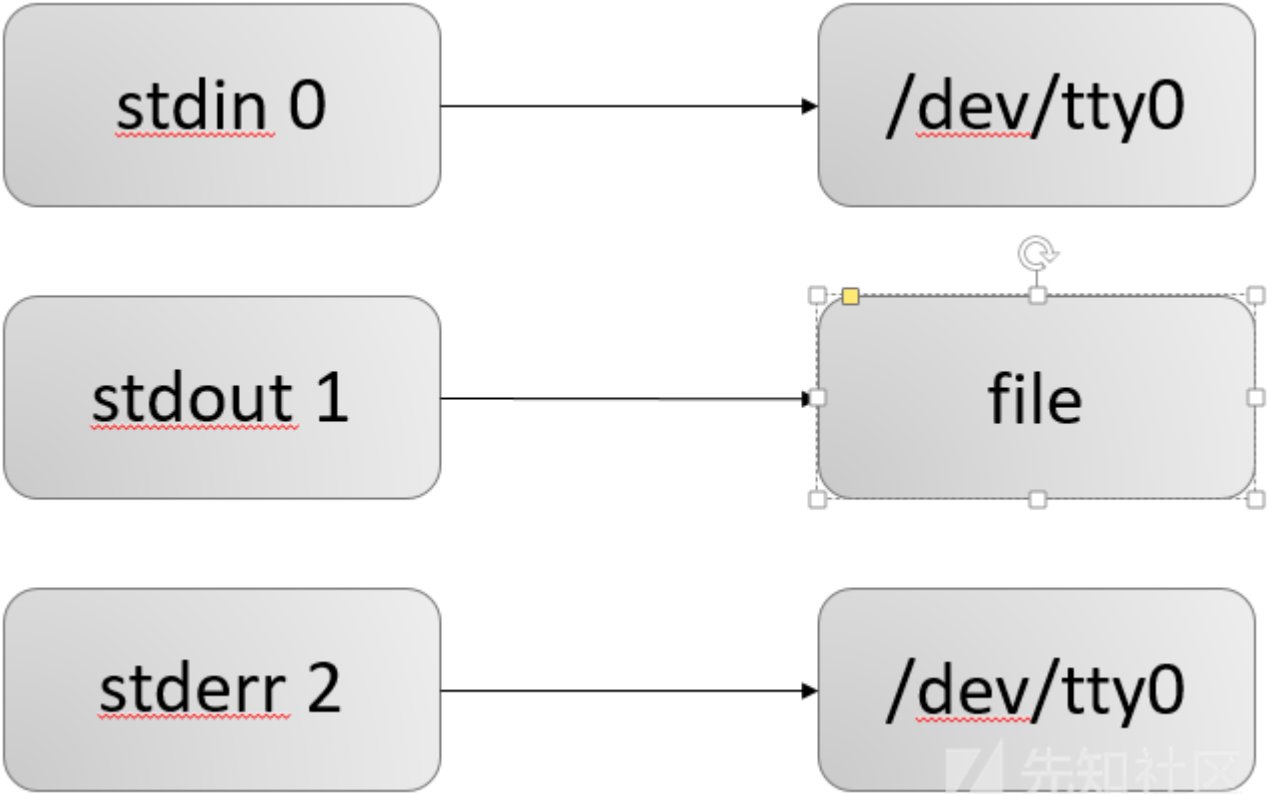
格式：[n]> word

```
root@K0rz3n:~# echo hello > file
root@K0rz3n:~# cat file
hello
root@K0rz3n:~# echo world 1> file
root@K0rz3n:~# cat file
world
root@K0rz3n:~#
```

```
root@K0rz3n:~# > file echo hello
root@K0rz3n:~# cat file
hello
root@K0rz3n:~# > file echo world
root@K0rz3n:~# cat file
world
root@K0rz3n:~#
```

说明： 将文件描述符 n 重定向到word 指代的文件（以写的方式打开），如果n 省略则默认就是 1（标准输出）

图示：



3.标准输出与标准错误输出重定向

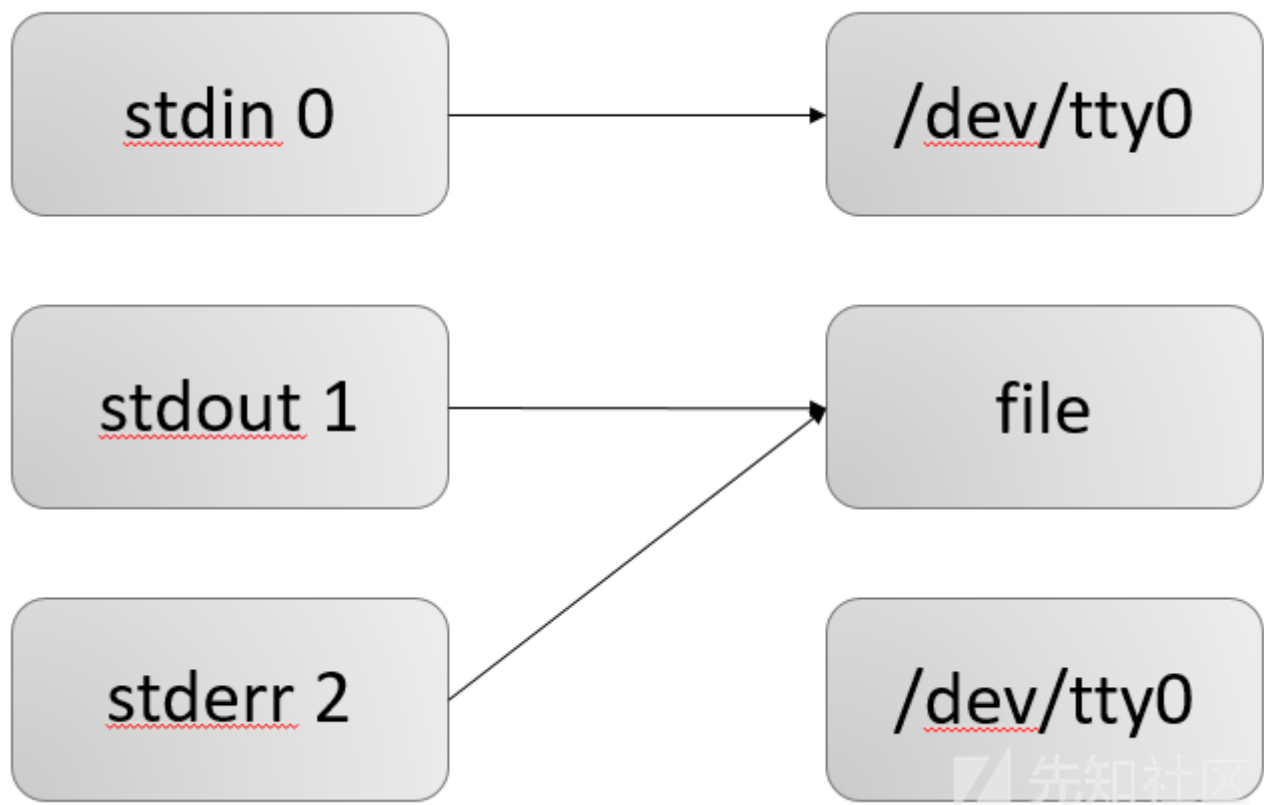
格式： &> word >& word

说明:将标准输出与标准错误输出都定向到word代表的文件（以写的方式打开），两种格式意义完全相同，这种格式完全等价于 > word 2>&1 (2>&1 是将标准错误输出复制到标准输出，&是为了区分文件1和文件描述符1的，详细的介绍后面会有)

```
root@K0rz3n:~# mkdir &> file
root@K0rz3n:~# cat file
mkdir: 缺少操作数
Try 'mkdir --help' for more information.
root@K0rz3n:~# ls &> file
root@K0rz3n:~# cat file
file
VMwareTools-10.1.6-5214329.tar.gz
vmware-tools-distrib
公共模板
视频
图片
文档
下载
音乐
桌面
root@K0rz3n:~#
```

解释：我们首先执行了一个错误的命令，可以看到错误提示被写入文件（正常情况下是会直接输出的），我们又执行了一条正确的指令，发现结果也输入到了文件，说明正确

图示：



4.文件描述符的复制

格式：[n]<&[m] / [n]>&[m] (这里所有字符之间不要有空格)

说明：

1) 这里两个都是将文件描述符 n 复制到 m，两者的区别是，前者是以只读的形式打开，后者是以写的形式打开

因此 0<&1 和 0>&1 是完全等价的（读/写方式打开对其没有任何影响）

2) 这里的& 目的是为了区分数字名字的文件和文件描述符，如果没有& 系统会认为是将文件描述符重定向到了一个数字作为文件名的文件，而不是一个文件描述符

这里就可以用上面的例子作为演示，将错误和正确的输出都输入到文件中

重点：

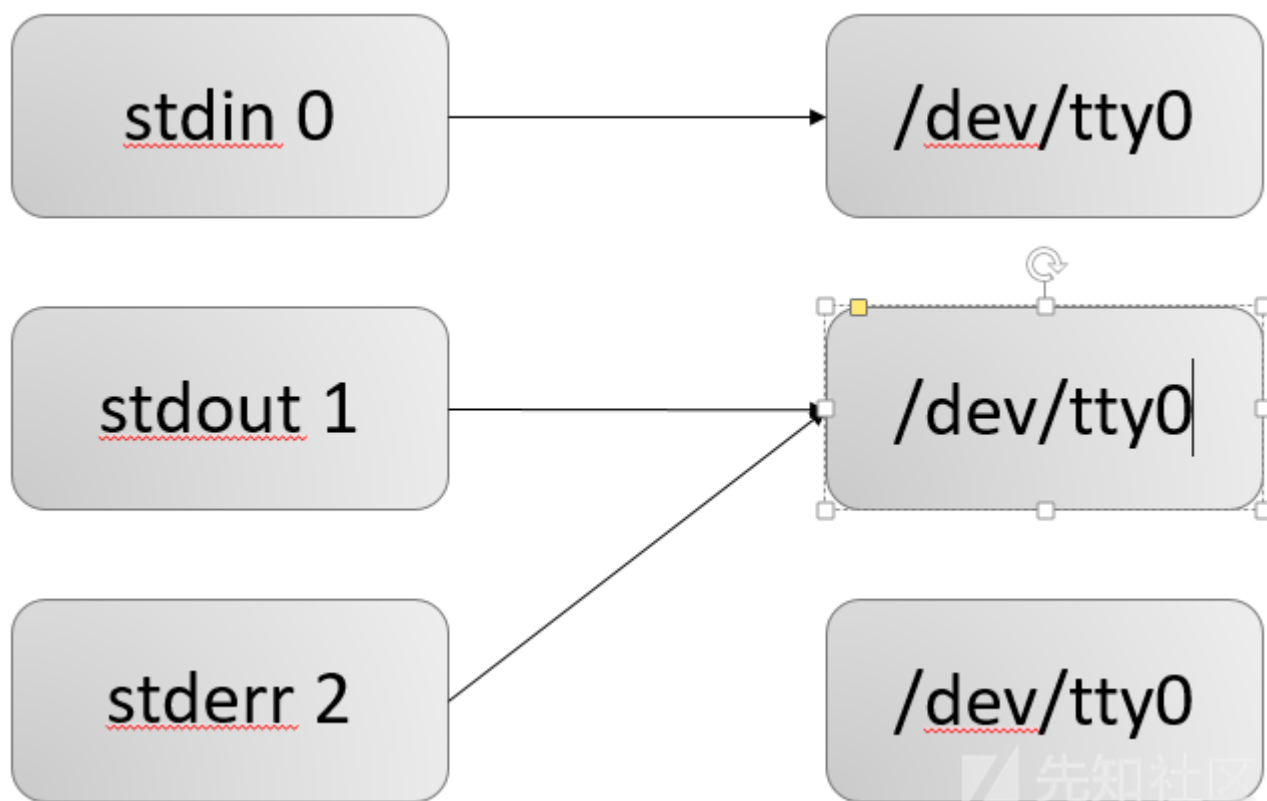
之前我们说过，重定向符号的顺序不能随便换，因为系统是从左到右执行的，我们下面就举一个例子

(1)cmd > file 2>&1

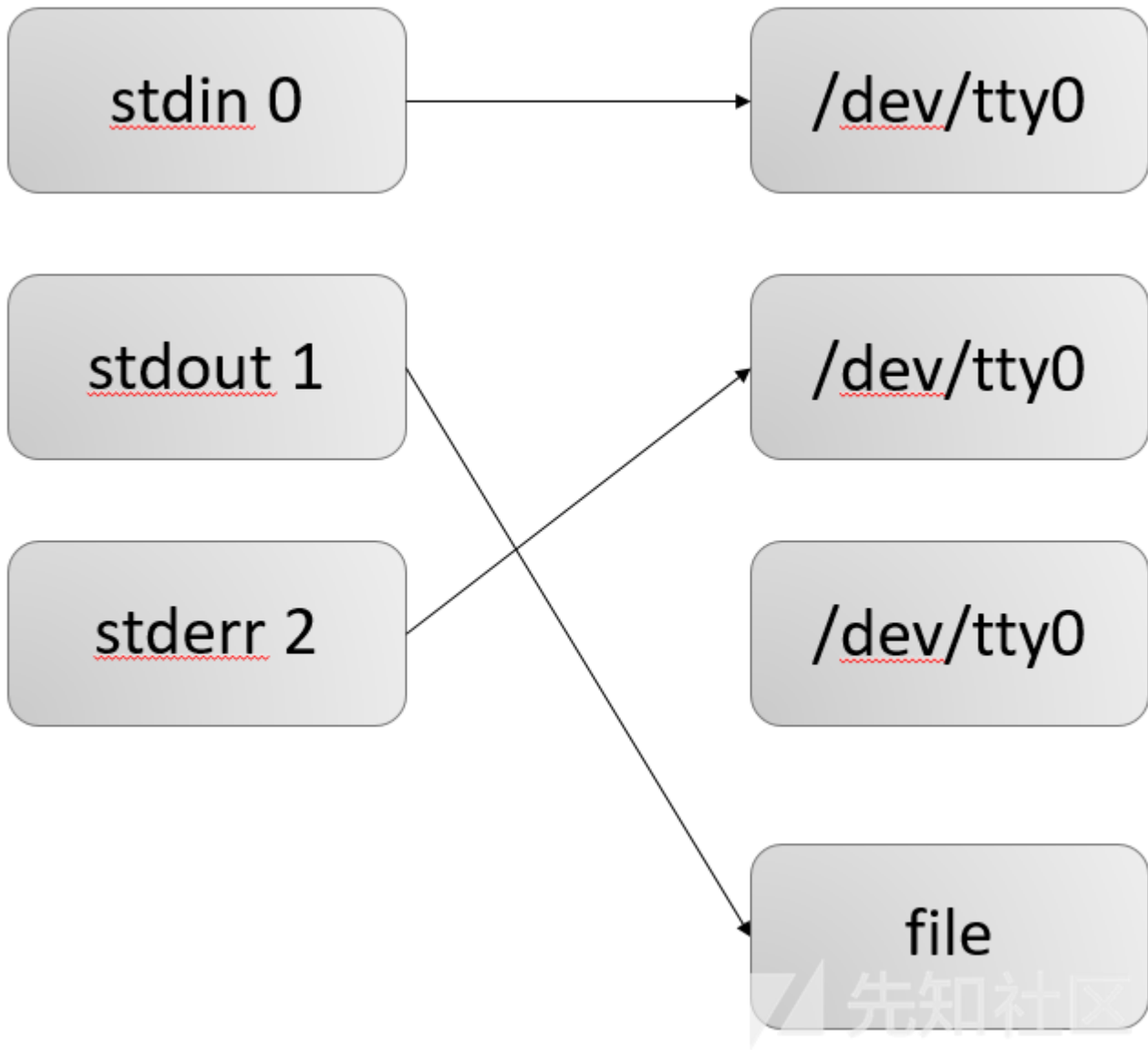
(2)cmd 2>&1 >file

与第一条指令类似的指令在上面我已经介绍过了，我们现在来看看第二条指令的执行过程

1.首先解析器解析到 2>&1



2.解析器再向后解析到 ">"



5.exec 绑定重定向

格式：exec [n] </> file/[n]

上面的输入输出重定向将输入和输出绑定文件或者设备以后只对当前的那条指令有效，如果需要接下来的指令都支持的话就需要使用 exec 指令

重点：

格式：[n]<>word

说明：以读写方式打开word指代的文件，并将n重定向到该文件。如果n不指定的话，默认为标准输入。

```

root@K0rz3n:~# exec 3<>file
root@K0rz3n:~# ls >&3
root@K0rz3n:~# cat file
file
VMwareTools-10.1.6-5214329.tar.gz
vmware-tools-distrib
公共
模板
视频
图片
文档
下载
音乐
桌面

```

格式：[n]<>word
说明：以读写方式打开word指代的文件，并将n重定向到该文件。如果n不指定的话，默认为标准输入。

先知社区

```

root@K0rz3n:~# exec 3<>file shell.sh
root@K0rz3n:~# cat <&3
file
VMwareTools-10.1.6-5214329.tar.gz
vmware-tools-distrib
公共
模板
视频
图片
文档
下载
音乐
桌面
root@K0rz3n:~#

```

关闭文件的输入输出绑定

```

exec 3>&-
exec 3<&-

```

先知社区

0X03 总结

文件描述符和重定向的作用巨大，很好的体现出了Linux中一切皆文件的特性，在反弹shell建立交互通道的过程中也起到了至关重要的作用。

个人博客：<http://www.k0rz3n.com>

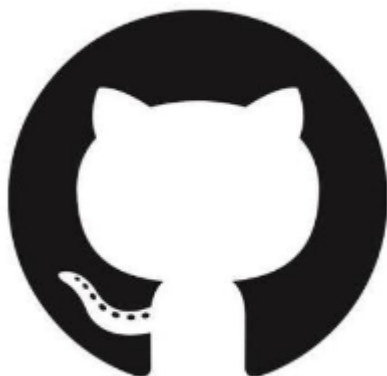
0X04 参考链接

<https://blog.csdn.net/ccwwff/article/details/48519119>
<http://www.cnblogs.com/chengmo/archive/2010/10/20/1855805.html>
<http://www.178linux.com/54471>

点击收藏 | 10 关注 | 1

[上一篇：BurpSuite Extende...](#) [下一篇：Linux 反弹shell \(二\) 反...](#)

1. 7 条回复



[chybeta](#) 2018-08-10 10:16:28

2 回复Ta





[枕边月亮](#) 2018-08-13 16:22:34

写的很好，来给圆圆顶一下

1 回复Ta



[54497****@qq.com](#) 2018-11-28 17:37:19

dup2(), 就看这个就好了

0 回复Ta



[峨眉峰独照](#) 2019-08-27 11:25:08

最后这个图是否有画错，标准输出和出错应该都到 file 去了。

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)