脚本除错 启动环境

mktemp 命令，trap 命令

网道（WangDoc.com），互联网文档计划

Bash 脚本有时需要创建临时文件或临时目录。常见的做法是，在 `/tmp` 目录里面创建文件或目录，这样做有很多弊端，使用 `mktemp` 命令是最安全的做法。

目录 [隐藏]

1. 临时文件的安全问题

直接创建临时文件，尤其在 `/tmp` 目录里面，往往会导致安全问题。

首先，`/tmp` 目录是所有人可读写的，任何用户都可以往该目录里面写文件。创建的临时文件也是所有人可读的。

```
$ touch /tmp/info.txt
$ ls -l /tmp/info.txt
```

```
-rw-r--r-- 1 ruanyf ruanyf 0 12月 28 17:12 /tmp/int
```

上面命令在 `/tmp` 目录直接创建文件，该文件默认是所有人可读的。

其次，如果攻击者知道临时文件的文件名，他可以创建符号链接，链接到临时文件，可能导致系统运行异常。攻击者也可能向脚本提供一些恶意数据。因此，临时文件最好使用不可预测、每次都不一样的文件名，防止被利用。

最后，临时文件使用完毕，应该删除。但是，脚本意外退出时，往往会忽略清理临时文件。

生成临时文件应该遵循下面的规则。

> - 创建前检查文件是否已经存在。
> - 确保临时文件已成功创建。
> - 临时文件必须有权限的限制。
> - 临时文件要使用不可预测的文件名。
> - 脚本退出时，要删除临时文件（使用 `trap` 命令）。

2. mktemp 命令的用法

`mktemp` 命令就是为安全创建临时文件而设计的。虽然在创建临时文件之前，它不会检查临时文件是否存在，但是它支持唯一文件名和清除机制，因此可以减轻安全攻击的风险。

直接运行 `mktemp` 命令，就能生成一个临时文件。

```
$ mktemp
/tmp/tmp.4GcsWSG4vj

$ ls -l /tmp/tmp.4GcsWSG4vj
-rw------- 1 ruanyf ruanyf 0 12月 28 12:49 /tmp/tmp
```


链接

本文源码

代码仓库

反馈

上面命令中， `mktemp` 命令生成的临时文件名是随机的，而且权限是只有用户本人可读写。

Bash 脚本使用 `mktemp` 命令的用法如下。

```
#!/bin/bash

TMPFILE=$(mktemp)
echo "Our temp file is $TMPFILE"
```

为了确保临时文件创建成功， `mktemp` 命令后面最好使用 OR 运算符（ `||` ），保证创建失败时退出脚本。

```
#!/bin/bash

TMPFILE=$(mktemp) || exit 1
echo "Our temp file is $TMPFILE"
```

为了保证脚本退出时临时文件被删除，可以使用 `trap` 命令指定退出时的清除操作。

```
#!/bin/bash

trap 'rm -f "$TMPFILE"' EXIT

TMPFILE=$(mktemp) || exit 1
echo "Our temp file is $TMPFILE"
```

3. mktemp 命令的参数

`-d` 参数可以创建一个临时目录。

```
$ mktemp -d
/tmp/tmp.Wcau5UjmN6
```

`-p` 参数可以指定临时文件所在的目录。默认是使用 `$TMPDIR` 环境变量指定的目录，如果这个变量没设置，那么使用 `/tmp` 目录。

```
$ mktemp -p /home/ruanyf/
/home/ruanyf/tmp.FOKEtvs2H3
```

`-t` 参数可以指定临时文件的文件名模板，模板的末尾必须至少包含三个连续的 `X` 字符，表示随机字符，建议至少使用六个 `X` 。默认的文件名模板是 `tmp.` 后接十个随机字符。

```
$ mktemp -t mytemp.XXXXXXX
/tmp/mytemp.yZ1HgZV
```

4. trap 命令

`trap` 命令用来在 Bash 脚本中响应系统信号。

最常见的系统信号就是 SIGINT（中断），即按 Ctrl + C 所产生的信号。 `trap` 命令的 `-l` 参数，可以列出所有的系统信号。

```
$ trap -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SI
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SI
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SI
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SI
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SI
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SI
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SI
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SI
43) SIGRTMIN+9  44) SIGRTMIN+10        45) SIGRTMIN+1
48) SIGRTMIN+14        49) SIGRTMIN+15        50) SI
53) SIGRTMAX-11        54) SIGRTMAX-10        55) SI
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SI
63) SIGRTMAX-1  64) SIGRTMAX
```

`trap` 的命令格式如下。

```
$ trap [动作] [信号1] [信号2] ...
```

上面代码中，“动作”是一个 Bash 命令，“信号”常用的有以下几个。

> - HUP：编号1，脚本与所在的终端脱离联系。
> - INT：编号2，用户按下 Ctrl + C，意图让脚本终止运行。
> - QUIT：编号3，用户按下 Ctrl + 斜杠，意图退出脚本。
> - KILL：编号9，该信号用于杀死进程。
> - TERM：编号15，这是 `kill` 命令发出的默认信号。
> - EXIT：编号0，这不是系统信号，而是 Bash 脚本特有的信号，不管什么情况，只要退出脚本就会产生。

`trap` 命令响应 `EXIT` 信号的写法如下。

```
$ trap 'rm -f "$TMPFILE"' EXIT
```

上面命令中，脚本遇到 `EXIT` 信号时，就会执行 `rm -f "$TMPFILE"` 。

trap 命令的常见使用场景，就是在 Bash 脚本中指定退出时执行的清理命令。

```
#!/bin/bash

trap 'rm -f "$TMPFILE"' EXIT

TMPFILE=$(mktemp) || exit 1
ls /etc > $TMPFILE
if grep -qi "kernel" $TMPFILE; then
  echo 'find'
fi
```

上面代码中，不管是脚本正常执行结束，还是用户按 Ctrl + C 终止，都会产生 `EXIT` 信号，从而触发删除临时文件。

注意，`trap` 命令必须放在脚本的开头。否则，它上方的任何命令导致脚本退出，都不会被它捕获。

如果 `trap` 需要触发多条命令，可以封装一个 Bash 函数。

```
function egress {
  command1
  command2
  command3
}

trap egress EXIT
```

5. 参考链接

- Working with Temporary Files and Directories in Shell Scripts, Steven Vona
- Using Trap to Exit Bash Scripts Cleanly
- Sending and Trapping Signals

脚本除错　　启动环境

联系：contact@wangdoc.com