

## 关于 **tt.pl** 的说明

[tt.pl](#) 及其替代版本 [tt2.pl](#)，是一个 perl 脚本。可以按照配置文件的内容，使用 ssh 协议模拟运维人员的登录过程，从 window NT 客户端上批量执行 linux 服务器的管理任务。[tt2.pl](#) 解决了 [tt.pl](#) 在执行 ssh 时的乱码问题。

下文用 [tt.pl](#) 指称 [tt.pl](#) 和 [tt2.pl](#)，所说内容对二者也都是正确的。需要指明是二者之一时会明确给出。

### 需求条件：

安装了较新 perl 环境的 Windows 2000 或者 XP 客户端，其他 windows NT 平台也可以。推荐使用 [activeperl](#) 的 perl 分发版，这也是目前唯一已经测试过的 perl 分发版。

需要安装 expect 程序，请从此处 [expect-5.21r1b1-setup.exe](#) 下载。

需要安装 [openssh](#) 的客户端。

如果使用 [tt2.pl](#)，还需要安装 [activetcl](#) 包。[activetcl](#) 安装完毕后，需要手工安装 Tcl 的 Expect 包。在 cmd 里，cd 到 Tcl 的安装目录下，cd 到其下的 bin 目录下，执行如下命令即可：

```
teacup install Expect
```

以上程序都是开源软件。在 google 里搜索可能找到这些程序的最新稳定版本。

链接失败的话，也请放 google 寻找搜索下载。

由于需要把服务器密码明文地存放在文本文件里，如果因安全原因，需要加密这些文件，请安装 [openssl](#) 或者 [gpg](#)，或者其他可以加密的软件，或者采取其他配套的安全措施。

被管理的服务器端支持 ssh 连接。

### **tt.pl** 的构成：

除了 [tt.pl](#) 外，还有 3 个 expect 脚本：[ssh.txt](#)、[sftp.txt](#) 和 [sussh.txt](#)。所有与服务器交互的动作，都是由这三个脚本处理。[tt.pl](#) 只是负责解析配置文件内容，并按需调用这三个脚本。

[tt2.pl](#) 与 [tt.pl](#) 类似。也有 4 个脚本作为辅助：[tclssh.tcl](#)、[tclsussh.tcl](#)、[tclssh\\_passwd.tcl](#) 和 [tclsussh\\_passwd.tcl](#)，都是 tcl 脚本。

### 如何使用：

[tt2](#) 的 man 页如下：

#### **NAME**

[tt2](#) - simple linuxes managing platform for win32

#### **SYNOPSIS**

```
tt2.pl [options] sftpcmd sftpfile sshcmd configofhost
```

```
tt2.pl --chgpaswd [options] passwd_cmd new_password configofhost
```

#### **DESCRIPTION**

[tt2](#) login remote linux host using ssh by a password to excute sftp/ssh commands. It supports simple regexs that can be used in selecting some hosts from a configure file. [tt2](#) have command mode and changing password mode. [tt2](#) works in command mode,except use --chgpaswd option.

-r	reverse regexes's match test
--user regex	select hosts which login name matches regex
--passwd regex	select hosts which login password matches regex
--rootpasswd regex	select hosts which root's password matche regex
--port regex	select hosts which ssh's port matches regex
--ip regex	select hosts which ip matches regex
--chgpaswd	select changing passwd mode

## REPORT BUG

Please report bug to [ningxibo@gmail.com](mailto:ningxibo@gmail.com)

可以不帶参数执行一下 tt.pl:

```
tt.pl
```

这样 tt.pl 会给出一行关于用法的简单说明。输出结果大致如下:

```
tt.pl sftpcmd sftpfile sshcmd configofhost.
```

sftpcmd 可以是 put, 或者 get, sftpfile 是文件名字。这两个参数告诉 tt.pl 要上传 (put) 或者下载 (get) 名字为 filename 的文件。sshcmd 是要在远程服务器上通过 ssh 执行的命令。configofhost 是配置文件, tt.pl 从这个文件中寻找 ssh 登录名、密码和主机 ip 及端口等等。它的具体格式见下面说明。

tt.pl 的四个参数, 可以分成三部分。一部分是 sftp 需要的两个参数: sftpcmd, 说明是上传 (put) 还是下载 (get); sftpfile, 说明上传下载的文件路径 (事实上此文件只能放在 tt.pl 所在目录下。已经测试过了, 在其他路径下会报错。所以, 准确的说, 应该这里应该说的是“sftpfile 说明上传下载的文件名”)。这两个参数必须按 sftpcmd sftpfile 的顺序一起给出。

另一部分是 ssh 需要的, 1 个参数, 说明需要在远程服务器上执行的命令。

最后一部分是配置文件, 1 个参数, 说明配置文件所在的路径。配置文件这部分的参数必须总是最后一个参数。配置文件的格式下面专有一小节予说明。

tt.pl 的参数, 除了配置文件外, 其他都可以为选。

参数里如果有空格或者特殊字符, 应该双引号引起来, 以防 perl 解释了特殊字符。另外需要注意的是, 如果 tt.pl 的参数里同时有 sftp 参数和 ssh 参数, tt.pl 总是按参数出现的次序执行相应的动作。就是说, 是先 sftp 文件还是 ssh 执行命令, 取决于 sftp 部分的参数和 ssh 部分的参数出现的先后次序。

下面是一些使用的例子。

让 tt.pl 只执行上传或者下载任务:

```
tt.pl sftpcmd sftpfile configofhost
```

让 tt.pl 只远程执行命令:

```
tt.pl sshcmd configofhost
```

让 tt.pl 只解析一下配置文件:

```
tt.pl configofhost
```

先执行上传任务再执行命令:

```
tt.pl get remotefile cmd configofhost
```

先执行命令再执行下载任务:

```
tt.pl cmd get remotefile configofhost
```

简单说, 就是 tt.pl 具有使用 sftp 上传下载, 并且同时可以在远程服务器上执行简单命令的功能。比如说:

```
tt.pl put ssh.txt "ls; rm -f ssh.txt" config.txt
```

就是按照 config.txt 里给出的用户名、密码和 ip, 逐一 sftp 登录, 上传 ssh.txt 文件至服务器, 而后, ssh 登录到服务器, 执行 rm -f ssh.txt 命令, 删除 ssh.txt 文件。

tt.pl 设计时, 考虑到典型的用法是, 把需要部署到服务器的文件和部署脚本, 打成一个包; 而在 sshcmd 参数里指定解包命令和执行部署脚本, 比如:

```
tt.pl put dist.tar.bz2 "tar xjf dist.tar.bz2 && chmod u+x dist.sh && ./dist.sh" config.txt
```

包 dist.tar.bz2 包含部署脚本和需要部署脚本处理的文件。tt.pl 把包 dist.tar.bz2 上传至服务器后，接着 ssh 到服务器，执行解包命令：tar xjf dist.tar.bz2。接着执行部署脚本：

```
chmod u+x dist.sh && ./dist.sh。
```

由 dist.sh 处理复杂的管理任务。自然，如果只需上传下载或者只需执行管理任务，只是典型用法的子集，更不用说远程执行几个简单命令了。

tt2 版本 1.3 兼容 tt.pl。单独执行 tt2.pl，输出如下：

```
tt2.pl [options] sftpcmd sftpfile sshcmd configfile
tt2.pl --chpasswd [options] passwd_cmd new_password configfile
```

可选参数（options）部分使得 tt2 支持使用简单的正则表达式（实际上是标准的 perl 正则表达式），来选择对配置文件里部分主机执行管理任务。配置文件里出现的所有配置项目都可以使用正则表达式进行匹配检查，还提供了对正则表达式的匹配结果取反的选项 -r。比如：

```
tt2 --ip "\.12$" ls configfile
```

在配置文件里所有 ip 匹配 \*.\*.12 模式的服务器上执行 ls 命令。而

```
tt2 -r --ip "\.12$" ls configfile
```

则是在所有 ip 不匹配 \*.\*.12 模式的服务器上执行 ls 命令。请参考手册页。

#### 注意：

正则表达式选项之间是 and 关系。-r 则是对所有正则表达式执行取反操作。

所有选项必须放在非选项参数的前面。

#### tt2 的密码更改模式

tt2 现在提供一个密码更改模式（不是正则表达式的模式 pattern，而是 mode）。用法如下：

```
tt2 --chpasswd "passwd" "dufi$%%1GH" configfile
```

#### 关于配置文件：

配置文件是 win 下的文本文件。其基本构成部分是配置段。配置段的格式如下：

```
user=string
passwd=string
port=string
rootpasswd=string
server_ip1
server_ip2
.....
```

看个简单点的配置文件：

```
user=ning
passwd=ningxibo
port=22
rootpasswd=123456
192.168.5.123
192.168.5.124
192.168.5.125
```

这个配置文件内容，指定 tt.pl 使用用户命 ning 和密码 ningxibo 通过 ssh 登录 linux 服务器，服务器的 ssh 端口是 22（如果是 ssh 默认端口 22，port=22 可以省略）。登录成功后，执行 su，使用 root 密码 123456 取得 root 权限。需要管理的服务器是 3 台：192.168.5.123~125。

注意，我们这里所说的权限改变，是指 ssh 到服务器远程执行命令命令时才会发生的。sftp 相关的动作总是在最低权限下执行；下文也需如此理解。  
我们看个复杂点的配置文件。

```
user=ning
passwd=ningxibo
port=22
rootpasswd=123456
192.168.5.123
192.168.5.124
192.168.5.125
user=liwei
passwd=liweiwj
port=8000
192.168.5.126
192.168.5.127
user=root
passwd=ming
port=23456
rootpasswd=
192.168.5.128
192.168.5.129
192.168.5.130
```

这个配置文件内容比起第一个来，复杂了一些。我们逐一说说。

对于 192.168.5.123~125 这 3 台服务器，仍然如上面解释所说的方式登录。192.168.5.126~127 这两台服务器，其 root 密码没有变，但是登录用的普通用户帐号却变了 liwei,这个用户的密码是 liweiwj。192.168.5.128~130 这三台服务器，则是直接用 root 账户登录。需要注意的是，不需要 su 转变权限的情况下，无需 rootpasswd 配置。root 账户和普通账户使用相同的配置项：user=?和 passwd=?。需要注意的是 192.168.5.128~130 的登录过程虽然不需要配置 rootpasswd，但是与之相关的配置段里，仍然需要有个“rootpasswd=”设置。这是因为上面其他服务器的配置段里，有 rootpasswd 相关的配置。配置段总是继承其上面段里的变量定义。所以 192.168.5.128~130 的配置段，要用“rootpasswd=”明显的取消上面配置端里 rootpasswd 的配置。

就是说，user、passwd、port、rootpasswd 在一个配置段里的设定，会影响其下的配置段，除非其下的配置段另有设定，否则这些变量值不变。还有一点就是，不需要 su 变化执行权限的话，只需在配置文件里设定 user 和 passwd 既可，无需设定 rootpasswd。直接以 root 登录的话，把 root 看成普通用户即可。

tt.pl 会忽略文件里的空行和所有以#开头的行。

所有的配置都必须占居一行并且此行不能有其他字符。

#### **需要注意：**

如果您安装了 cygwin，:( 这个软件的一些组件，与 expect 程序冲突。需要卸载。

tt.pl、ssh.txt、sussh.txt、sftp.txt、tt2.pl、telssh.tcl、tclsussh.tcl、telssh\_passwd.tcl 和 tclsussh\_passwd.tcl 应该放在同一目录下。上传下载的文件也应该置于此目录下；测试发现，

不如此会出错。但是配置文件可以放到任意目录。总之一句话，这个软件应该工作在简单的环境里，以得到足够的健壮性。

expect, activeperl、activetcl 和 openssh 安装完毕后，请检查相应的可执行程序所在目录是否加进了 PATH 环境变量。

tt.pl 没有使用选项处理的办法是因为，或许这样更容易与 windows 脚本配合。

Activetcl 的包，因 AMD 平台的 DEP 而会导致 spawn 进程失败；windows XP 的 dbghelp.dll 也会导致 spawn 失败。不知道这是不是无法用 activetcl 实现 spawn sftp 功能的原因呢？

#### **升级声明：**

版本 1.2 会在目录 log 下生成日志文件。日志文件名称与配置文件大致类似，一望便知。

版本 1.3 版本会在目录 log 下生成日志文件。日志文件名称与配置文件大致类似，一望便知。但是与版本 1.2 不同的是，现在的日志文件名称里加上了命令开始时的时间，精度到秒。这意味着不再存在因日志文件而导致的不能并行执行问题。同时也意味着，日志文件需要手工清除处理了。

宁希波  
20090803