Cristiano Pinheiro Machado
cmachado@cbpf.br

Alexandre Maia de Almeida
amaia@cbpf.br

Nilton Alves Jr.
naj@cbpf.br

A ferramenta Pancho para roteadores e *switches* Cisco

Resumo

O Pancho é uma ferramenta voltada para a administração remota de equipamentos de rede da marca Cisco. Esta ferramenta auxilia o administrador da rede na monitoração, manutenção e alteração das configurações desses equipamentos.

Nesta nota técnica serão descritas as funcionalidades desse aplicativo, assim como seu mecanismo de funcionamento e um guia para sua instalação.

Índice

Objetivo

Esta nota técnica tem como principal objetivo apresentar a ferramenta Pancho de administração remota de *switches* e roteadores da marca Cisco. Serão mostradas suas utilidades, seu mecanismo de funcionamento e um roteiro para sua instalação.

Para um bom entendimento desta nota técnica é necessário um conhecimento mínimo e básico a respeito de redes de computadores e de sistemas Linux.

É importante também ressaltar que a versão do Pancho abordada nesta nota técnica é a de 4.0, e que versões mais atualizadas são disponibilizadas constantemente no website do Pancho: http://pancho.lunarmedia.net pelo autor Charles J. Menezes.

Apresentação

Com o crescimento da importância das redes de computadores nos dias de hoje, torna-se indispensável que as redes estejam seguras e sujeitas a um mínimo de falhas. Nesse cenário torna-se imprescindível a necessidade de serem feitas cópias de segurança - backups das configurações dos equipamentos responsáveis pelo funcionamento das redes.

Nos equipamentos atuais, existem basicamente dois meios de se realizar a cópia das configurações dos equipamentos: ou através da própria CLI - Cisco IOS *Command Line Interface*, que é o meio principal de configurar, monitorar e manter dispositivos Cisco, ou através do protocolo de gerenciamento SNMP – *Simple Network Management Protocol*.

Dessa forma, temos que uma ferramenta bastante eficiente capaz de realizar este tipo de cópia de segurança é o Pancho. O Pancho consiste em um script feito em linguagem Perl projetado para funcionar em sistemas Linux que se utiliza dos recursos do protocolo SNMP para realizar a cópia remota das configurações dos equipamentos Cisco de uma rede.

Ele comunica-se com os dispositivos remotos através do protocolo SNMP presente nos roteadores e *switches* em questão. Dessa forma, além de suporte à Perl junto com o módulo Net::SNMP (e suas dependências) no computador que vai executar o Pancho, é necessário que os equipamentos estejam configurados com o suporte à SNMP e também que haja um servidor TFTP acessível na rede onde eles estão.

Os dispositivos Cisco permitem copiar arquivos de configuração - *running-config* e *startup-config* a partir da ou para a rede, através de TFTP ou RCP. Isso pode ser usado para salvar backups de configurações ou restaurar configurações arquivadas. Pode-se também querer que vários roteadores tenham configurações similares. Nesse caso, basta copiar a configuração de um deles para os outros e alterar somente as partes relevantes.

O Pancho é um aplicativo de distribuição gratuita e bastante interessante porque constitui uma interface simples, direta e flexível para esse recurso de manipulação das configurações dos equipamentos. Ele torna possível implementar uma política sistemática de backup e fazer alterações conjuntas numa lista de roteadores e *switches*, operando com IOS versão 12.x (preferencialmente) ou 11.x.

Gerenciamento SNMP

Visão Geral

O SNMP é um protocolo usado para gerenciamento de redes, usado pelos administradores para detectar problemas, controlar roteamentos, fazer o monitoramento e realizar tarefas de gereciamento em geral. O protocolo tem como base três conceitos distintos: gerentes, agentes e a MIB - *Management Information Base*.

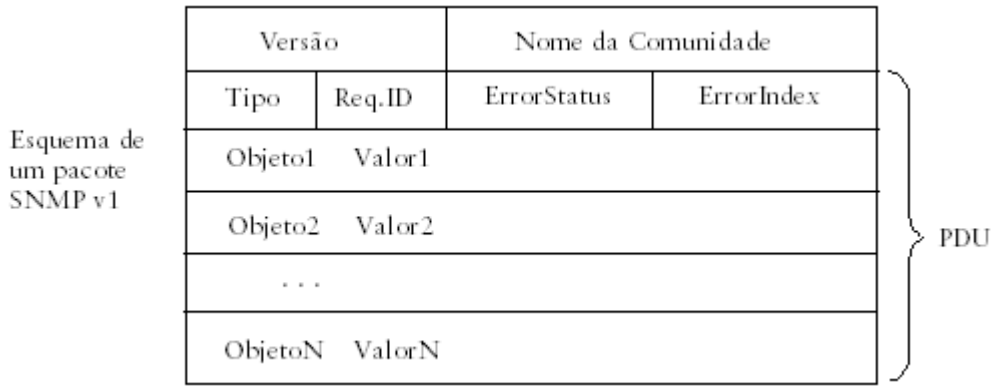
O SNMP utiliza o UDP como protocolo de transporte. O pacote pode ser dividido em duas partes, uma parte constituída pelo *wrapper* ou cabeçalho, que na versão 1 era o número da versão do SNMP seguido pelo nome de

comunidade, e que em versões posteriores do protocolo passou a incluir outros dados de autenticação e privacidade, e a outra parte constituída pelo PDU - *Protocol Data Unit*.

O PDU pode ser de vários tipos:

get-request
get-next-request
get-bulk
get-response
set-request
trap.

Um pacote do tipo set-request, por exemplo, contém um campo indicando seu tipo, campos de controle e uma sequência de identificadores de objetos e valores a serem atribuídos, conforme a figura:



MIB - Management Information Base

A MIB consiste numa árvore de objetos gerenciáveis cujos atributos podem ser lidos, atribuídos e modificados. Esses objetos podem representar entre outras coisas, informações do sistema, estatísticas de utilização, parâmetros de configuração, etc.

Além da MIB padrão, conhecida como MIB-II, existe espaço reservado para MIBs que vendedores específicos queiram implementar em seus dispositivos. Nessa nota técnica é de interesse particular uma extensão provida pela Cisco, que permite cópia de configurações.

Funcionamento

O Pancho funciona de forma bastante simples. Um uso típico do Pancho, por exemplo, seria:

```
$ pancho -d --server <ip-tftpd> --string <snmp community> --host <router> -f router-confg
```

Esse comando é equivalente ao seguinte, executado no roteador:

```
Router# copy system:running-config tftp://<ip>/router-confg
Write file router-confg on host <ip>? [confirm] y
Writing router-confg!!! [OK]
```

Observe que deve existir um servidor de TFTP acessível pelo roteador. Em algumas implementações de servidores TFTP, é necessário criar um arquivo vazio com permissões de leitura, escrita e execução antes de copiar o arquivo real por cima.

Quando executado com a opção *--upload*, a operação é equivalente ao comando

```
copy { ftp:| rcp: | tftp: } system:running-config.
```

Observe que ao se fazer o upload de uma configuração para o roteador, esta é salva em *running-config*. Desse modo a configuração existente anteriormente no roteador se mantém salva como *startup-config*. Logo, caso se reinicie o roteador, as configurações assumidas voltarão a ser aquelas especificadas antes como *startup-config*.

Para o caso de se desejar salvar as configurações em *startup-config* através do Pancho existem parâmetros de configuração que possibilitam isso.

A seguir será mostrado mais detalhadamente o mecanismo de funcionamento do Pancho, tomando como exemplo um pacote SNMP do tipo *set-request*.

Ao ser executado, o Pancho segue os seguinte passos:

(i) envia ao dispositivo um requerimento SNMP para o objeto cujo identificador é 1.3.6.1.2.1.1.1.0 (refere-se ao sysDescr, que inclui a descrição do hardware/sistema).

(ii) através do valor retornado, verifica a versão do IOS.
se for 11.x ou 10.x:
funcionará, porém a opção *–startup-config* não estará disponível

se for um *switch* Catalyst:
mesma restriçao acima

se for uma das versões 12.x:
envia um pacote do tipo *set-request*, que iniciará no dispositivo uma operação de cópia conforme os parâmetros passados.

(iii) envia *get-requests* para obter o status da operação. O dispositivo informará se ela já foi concluída ou ainda está em andamento.

(iv) permanece requisitando status até que a operação tenha falhado ou sido bem sucedida.

Veja abaixo o diagrama de um típico pacote SNMP enviado pelo Pancho:

Exemplo esquemático de um SNMP "Set-Request"

1	private_string		
8	xxxx	0	0
1.3.6.1.4.1.9.9.96.1.1.1.1.2		1	
1.3.6.1.4.1.9.9.96.1.1.1.1.3		4	
1.3.6.1.4.1.9.9.96.1.1.1.1.4		1	
1.3.6.1.4.1.9.9.96.1.1.1.1.5		10.0.1.2	
1.3.6.1.4.1.9.9.96.1.1.1.1.6		config/router1	
1.3.6.1.4.1.9.9.96.1.1.1.1.14		4	

O campo PDU type identifica que é um pacote do tipo *set-request*. Na parte de "*variable bindings*", existem seis pares de objetos gerenciados com seus respectivos valores. Os objetos estão referenciados pelo seu identificador numérico, eles estão na sub-árvore "Cisco Management" da MIB (ISO.ORG.DOD.Internet.Private.Enterprises.cisco.CiscoMgmt, ou 1.3.6.1.4.1.9.9).

Esta sub-árvore está definida num arquivo chamado CISCO-CONFIG-COPY-MIB, que pode ser obtido no site da Cisco (em ftp://ftp.cisco.com/pub/mibs/v1).

Esses objetos usados pelo Pancho possuem o seguinte significado:

"1.3.6.1.4.1.9.9.96.1.1.1.1.2"
ccCopyProtocol, pode possuir o valor 1 ou 3, indicando se será usado o protocolo TFTP ou RCP, respectivamente.

"1.3.6.1.4.1.9.9.96.1.1.1.1.3"
ccCopySourceFileType, indica o tipo de arquivo de origem. Pode ser 1 (*networkfile*), 3 (*startup-config*) ou 4 (*running-config*).

"1.3.6.1.4.1.9.9.96.1.1.1.1.4"
ccCopyDestFileType, mesmos valores que o anterior, mas agora indicando o arquivo de destino.

"1.3.6.1.4.1.9.9.96.1.1.1.1.5"
ccCopyServerAddress, o IP do servidor TFTP de onde (ou para onde) será copiado o arquivo.

"1.3.6.1.4.1.9.9.96.1.1.1.1.6"
ccCopyFileName, nome do arquivo (deve incluir o path, se for o caso).

"1.3.6.1.4.1.9.9.96.1.1.1.1.14"
ccEntryRowStatus, 4 – "createAndGo", para iniciar automaticamente, ou 5 – "createAndWait". Em outro contexto pode ser também 1 (active), 2 (notInService), 3 (notReady) ou 6 (destroy).

Ou seja, esse pacote requisita que se crie uma entrada (objeto do tipo ccCopyEntry) na tabela ccCopyTable com esses parâmetros. No exemplo, protocolo = TFTP, a cópia será feita do *running-config* para um arquivo da rede, o endereço do servidor TFTP 10.0.1.2 e o nome do arquivo a ser salvo config/router1).

Como resultado dessa operação de "set", será executada a cópia conforme os parâmetros indicados. Depois de iniciada, é necessário ler o valor de ccCopyFailCause (1.3.6.1.4.1.9.9.96.1.1.1.1.13) para confirmar que foi executada com sucesso.

É importante mencionar que as opções de política de segurança (firewall) da máquina que irá executar o Pancho permitam o tráfego de pacotes UDP, caso contrário mensagens de erro serão exibidas.

Instalação

Ativando suporte ao SNMP

O procedimento básico envolve a configuração de uma comunidade - SNMP *community string*. Para o Pancho é necessária uma comunidade RW (com permissão de leitura e escrita). É interessante também que a mesma comunidade seja configurada em todos os roteadores, pois no caso das operações em conjunto não se pode especificar comunidades diferentes para cada roteador.

Dessa forma, um comando suficiente, digitado em "global configuration mode" na CLI é:

```
snmp-server community <nome_comunidade> RW
```

Não serão discutidos aqui os detalhes da configuração, para isso consulte a documentação apropriada. Nem as consequências relativas a segurança decorrentes do uso de SNMP. Sabe-se que o SNMP versão 1 possui sérias limitações quanto a autenticação e privacidade, entretanto, como a SNMPv3 só está disponível em versões recentes do IOS (12.0.3) e o Pancho foi baseado na própria versão 1, nos restringiremos a essa configuração.

Observações

O servidor de TFTP é bastante simples de ser instalado. Apesar de não ser necessário que ele esteja no mesmo computador de onde será executado o Pancho, isso facilita pois o script para backup das configurações teria acesso direto ao diretório raiz do servidor. Um tutorial de instalação do servidor de TFTP pode ser encontrado no website do Pancho – http://pancho.lunarmedia.net/tftp.html

O Perl também precisa estar disponível, junto com o módulo Net::SNMP. O Perl é bastante comum nas plataformas Unix e geralmente é instalado junto com o sistema. Com o Perl funcionando, para instalar o módulo basta executar `perl -MCPAN -e 'install Net::SNMP',` como root. Neste caso a máquina em questão deve estar conectada na internet. Para mais informações, http://cpan.org ou http://perl.com

Adicionalmente, o autor do Pancho sugere que se crie um usuário pancho, e se ajuste permissões da seguinte forma:

```
%useradd -s /bin/false -d /dev/null pancho
%chown pancho.pancho pancho
%chmod 4755 pancho

%touch /var/log/pancho
%chown pancho.pancho /var/log/pancho
%chmod 600 /var/log/pancho
```

Opções

Ao executar o Pancho com o comando, `./pancho –help`, é exibida uma lista com todas as opções disponíveis:

```
$ ./pancho --help
     [ --upload | --download ]
     [ --start ]
     [ --commit ]
     [ --reload ]
     [ --list (filename) ]
     [ --host (hostname/ipaddress) ]
     [ --file (filename) ]
     [ --server (ip/fqdn/hostname) ]
     [ --string (rw snmp community) ]
     [ --path (path within TFTProot) ]
     [ --regex (regular expression) ]
     [ --fork ]
     [ --version ]
     [ --help ]
```

As configurações podem ser predefinidas editando-se o arquivo `pancho` ou colocadas na linha de comando. Se a a opção estiver na linha de comando, esta terá preferência sobre as que estiverem especificadas no arquivo.

Opções

-u, --upload
: indica que o arquivo será enviado ao(s) dispositivo(s) remoto(s)

-d, --download
: indica que arquivo(s) serão recebido(s) do(s) dispositivo(s) remoto(s)

--start
: dependendo do tipo de operação (upload ou download), executa "copy TFTP startup-config" ou "copy startup-config TFTP" nos dispositivos. Atenção que esta operação não esta disponível para versões do IOS inferiores a 12.x

-c, --commit
efetua uma operação de “write memory” ou “copy run start”

--reload
solicita um *reload* ao dispositivo. Essa opção só funcionará caso o parâmetro “snmp-server system shutdown” tenha sido posto no dispositivo.

--list
especifica nome de um arquivo que contenha, um por linha, nomes (ou ips) dos dispositivos onde serão requisitadas as operações.

--host
nome (ou ip) do dispositivo

-f, --filename
nome do arquivo a ser enviado ou copiado, no caso de uma operação de upload ou download, respectivamente.

--server
endereço do servidor TFTP

--string
nome da comunidade já configurada no roteador

-p, --path
path a partir do raiz do TFTP (onde se encontram os arquivos ou onde eles serão salvos)

--regex
flexibiliza a lista de dispositivos onde serão tomadas as ações. É possível indicar uma expressão regular, e somente as linhas do arquivo lista (especificado com o parâmetro --list) que “passarem” por essa regex serão consideradas.

--fork
essa opção pode ser usada caso hajam múltiplos dispositivos e se queira executar em todos paralelamente, diminuindo o tempo de execução.

--version
mostra o número de versão do Pancho

--help
exibe a listagem dos parâmetros

Existe também a opção de “logging”. Ela não está acessível através da linha de comando, para ativá-la é preciso editar o arquivo `pancho` e, logo no início, modificar as duas linhas seguintes:

```
my $logging = "0";      # set this to "1" to enable logging
my $logfile = "";       # full path to your logfile
```

Esse recurso é bastante útil quando não se executa o Pancho interativamente e sim através de procedimentos agendados. Dessa forma, os comandos executados ficam gravados nesse arquivo de log, dando ao administrador maior controle sobre erros que possam acontecer.

Exemplos

Como primeiro exemplo, para se testar se está tudo funcionando devidamente, é interessante se executar o script com todas as opções indicadas na linha de comando, contra um único roteador, como indicado:

```
%./pancho --download \
          --server <ip-tftpd> \
          --string <snmp community> \
          --host <router-name>
```

Com isso, deverá ser salvo um arquivo chamado *router-name*.cfg no raiz do servidor TFTP com as configurações do roteador.

Isto pode ser estendido para um número N de roteadores de forma bastante fácil. Cria-se um arquivo com os endereços dos roteadores, um por linha, e executa-se o script da seguinte forma:

```
%./pancho --download \
          --server <ip-tftpd> \
          --string <snmp community> \
          --list <nome_arquivo>
```

Também é possível, de maneira similar, enviar alterações comuns a um grupo indicado de roteadores. Por exemplo, os roteadores Cisco podem ser gerenciados remotamente através de telnet. Pode-se querer desativar esse recurso, com os seguintes comandos do IOS:

```
line vty 0 4
transport input none
```

então salva-se essas duas linhas num arquivo texto, dentro do servidor TFTP e envia-se para o roteador utilizando a opção de upload.

Suponha que o administrador queira enviar esses comandos de configuração para um subconjunto dos roteadores indicados no arquivo-lista do exemplo anterior. Pode-se filtrá-los da lista através de uma expressão regular e executar o Pancho da seguinte forma:

```
% ./pancho   --upload \
              --server <ip-tftpd>
              --string <private>
              --file <cmd_arquivo>
              --list <nome_arquivo>
              --regex <expressão_regular>
```

Ou então, suponha-se que queira que determinados roteadores usem a facilidade de syslog remoto, através do comando:

```
! Send syslog messages to the mgmt host and log with localtime
service timestamps log datetime localtime
logging syslog1_ip
logging syslog2_ip
```

Nesse caso, basta salvar esses comandos em arquivo e fazer *upload* como no exemplo anterior.

Backup remoto

Para nossos propósitos, será criado um *shell script* que utiliza o Pancho para fazer backup de uma lista de dispositivos e compará-los com backups anteriores, para notificação em caso de alterações. Isto permite um maior controle sobre essas mudanças nas configurações e possíveis anormalidades.

```
#!/bin/bash
#
#
#
#       Script para automatizar o Pancho (versão 4.0.1):
#
#       variáveis
#       ---------------------------------
#
#       TFTPDIR : diretório raiz do tftpd, onde os arquivos
#                 vindos do roteador serão gravados.
#
#       BACKDIR : diretório onde ficará a árvore com os subdiretórios
#                 de cada roteador e os arquivos de backup.
#
#       ARQUIVO_ROTEADORES : nome do arquivo que tem a lista
#                            de roteadores, 1 por linha.
#
#       SERVIDOR : o ip do tftpd
#
#       DATA : data atual
#       DATA_ONTEM : data do dia anterior
#


TFTPDIR="/tftpboot"
BACKDIR="/home/pancho/script/router_backup"
DIFFDIR="/home/pancho/script/router_backup"
ARQUIVO_ROTEADORES="/home/pancho/script/router_list"

SERVIDOR="<ip do servidor de TFTP>"
COMUNIDADE="<community string RW dos roteadores>"

DATA=$(date +%d%h%Y)
DATA_ONTEM=$(date  --date=yesterday +%d%h%Y)

CMD="perl -X /home/pancho/script/pancho -download \
    --fork --server $SERVIDOR --string $COMUNIDADE --list $ARQUIVO_ROTEADORES"

#       é preciso criar os arquivos no TFTPDIR inicialmente, com permissao 666
#       verificar também se os diretórios de cada roteador existem,
#       junto com o arquivo atual.cfg, de cada roteador
#
for i in $(cat $ARQUIVO_ROTEADORES)
do
        touch  $TFTPDIR/$i.cfg;
        chmod  666 $TFTPDIR/$i.cfg;

        if [ !  -d $BACKDIR/$i ]; then  mkdir $BACKDIR/$i; fi
        if [ !  -d $DIFFDIR/$i ]; then  mkdir $DIFFDIR/$i; fi

        touch $BACKDIR/$i/atual.cfg; # p/ garantir q existe um arquivo
        chmod 600 $BACKDIR/$i/atual.cfg
done

#       o arquivo router_list contem os nomes dos roteadores
#
```

```
$CMD

for i in $(cat $ARQUIVO_ROTEADORES)
do
        if [ ! -s $TFTPDIR/$i.cfg ]; then
                for c in 0 1 2 3 4
                do
                        if [ -n "$($CMD --regex "$i*" |grep SUCCESS)" ]; then break
                        fi
                done
        fi

        chmod 600 $TFTPDIR/$i.cfg
        diff --  $TFTPDIR/$i.cfg $BACKDIR/$i/atual.cfg > $DIFFDIR/$i/$DATA.diff

        if [ -s $DIFFDIR/$i/$DATA.diff ];
        then
                mail -s "ALTERAÇÃO $i !" <e-mail do adm> < $DIFFDIR/$i/$DATA.diff
                cp $TFTPDIR/$i.cfg $BACKDIR/$i/atual.cfg
                cp $TFTPDIR/$i.cfg $BACKDIR/$i/$DATA.cfg
        fi
        rm -f $DIFFDIR/$i/$DATA.diff
        rm -f $TFTPDIR/$i.cfg
done
#fim do arquivo
```

Associado ao script deverá haver uma linha no arquivo de crontab, que é o arquivo que agenda tarefas, indicando a periodicidade com que ele será executado.

Observe que este arquivo referencia um outro chamado router_list. Esse deverá ser a lista usada pelo Pancho. O arquivo tem a seguinte forma:

```
-----início do arquivo
<ip do primeiro roteador>
<ip do segundo roteador>
<ip do terceiro>
.
.
.
----fim do arquivo

Por exemplo :
```

```
10.0.1.1
10.1.1.1
10.0.2.3
```

Agendando Backups

O objetivo é automatizar o uso do Pancho, de forma que o backup das configurações dos dispositivos, por exemplo, não necessite de interação e seja executado regularmente. Ou mesmo para tarefas mais complexas, basta criar um “shell script” que execute os procedimentos desejados.

O cron é uma facilidade muito útil para se agendar e executar tarefas repetitivas. Elas podem ser executadas em datas específicas ou intervarlos regulares. Ele está presente na maioria das distribuições Unix e é de fácil configuração.

O conceito é simples, o servidor cron é iniciado durante o boot e deve permanecer ligado durante todo o tempo. Cada usuário tem um arquivo indicando as tarefas agendadas e o servidor cuida de executá-las nas horas indicadas.

Estando o servidor crond ativado, é necessaário verificar os arquivos /etc/cron.allow e /etc/cron.deny para certificar-se que o usuário desejado tem permissão para fazer uso do cron. Depois disso, basta que o usuário edite o arquivo com suas tarefas.

O arquivo indica os comandos e as datas/frequências em que ele deve ser executado.O comando para se manipular tal arquivo é o “crontab”.

“crontab <nome_do_arquivo>”, instala um arquivo crontab
“crontab –e”, edita o arquivo com as tarefas para esse usuário
“contrab –l”, mostra o arquivo
“contrab –r “, remove o arquivo

O formato do arquivo é o seguinte:
Cada linha contém 6 campos, os 5 primeiros indicam, respectivamente, minutos, horas, dias, meses e dias da semana. O último campo contém o comando a ser executado. Esses campos podem estar separados por tabulações ou espaços.

Os valores válidos para os 5 primeiros campos são:
0-59, para o campo dos minutos
0-23, das horas
1-31, dos dias
1-12, dos meses
0-6, para os dias da semana

Cada um desses campos pode conter um “*”, indicando todos os valores possíveis ou uma lista de elementos separados por vírgula. Um elemento consiste ou em um número ou dois números separados por hífen (significando todos os valores entre um e outro número, inclusive).

Por exemplo:

```
0       0       1,15    *       *       comando1
0       0        *      *       1       comando2
```

O comando1 seria sempre executado nos dias 1 e 15 as 0 horas, e devido a linha2 o comando2 seria executado todas as segundas-feiras às 0 horas.

Para mais informações sobre o cron, consulte a documentação apropriada ou o “man” do sistema com um dos comandos: “man 5 crontab”, “man 1 crontab”, “man cron”, “man at”, “man batch” ou “man 1 cron”.

Conclusão

Visto o mecanismo de funcionamento do Pancho, suas funcionalidades e sua simples forma de implementação, podemos dizer que esta é uma ferramenta de alta utilidade que torna bastante eficaz o monitoramento dos equipamentos Cisco de uma rede.

Na prática o Pancho torna fácil e rápida a recuperação de configurações que por ventura possam ser perdidas ou alteradas indevidamente, dando uma maior segurança aos administradores, principalmente em redes com um grande número de roteadores.

Bibliografia

Pancho – SNMP TFTP tool for Cisco Routers and Switches
http://pancho.lunarmedia.net/

RNP NewsGeneration Vol.6
http://www.rnp.br/newsgen/0203/pancho.shtml?pancho#first_hit

Cisco Systems, Inc
http://www.cisco.com/

Cisco – Moving Files and Images Between a Router and TFTP Server via SNMP
http://www.cisco.com/warp/public/477/SNMP/11_7910.shtml

The Simple Times
http://www.simple-times.org

www.SNMPLink.org – Simple Network Management Protocol
http://www.snmplink.org/