



## Harford Technology Corporation

Cisco Catalyst IOS Version 6.X  
Standard Security  
Configuration Version 0.86

### Abstract

As companies strive to become part of the e-Commerce industry, the Network and Systems are often the enabling technology for the business plan. Therefore, solid measures must be taken to protect the infrastructure from known ( and unknown ) threats present in the industry.

To aid in this endeavor, Harford Technology Corporation has assembled the attached security profile for Cisco's Catalyst IOS. While realizing that security is rarely a 'one size fits all' solution this document is intended to provide a framework of information to secure the device, while balancing operational concerns and maintaining the necessary scalability.

As new threats and techniques evolve, this document will be updated to reflect the new information.





© 2003 Harford Technology Corporation. All right reserved.

The information contained in this document represents the current view of Harford Technology Corporation on the issues discussed as of the date of publication. Because of the dynamic nature of business computing environments, Harford Technology Corporation cannot guarantee the accuracy of any information presented after the date of publication and/or delivery.

Harford Technology Corporation makes no warranties, express or implied, in this document.

Harford Technology Corporation is a registered trademark of Harford Technology Corporation. Other product or organizational names mentioned herein may be the trademarks of their respective owners.



---

## Table of Contents

Distribution.....	7
Document Update Process .....	7
Contributors.....	7
Conventions .....	7
Configuration Information.....	9
Catalyst OS Services .....	9
Logging.....	10
Authentication/Authorization/Accounting.....	11
tacacs+ .....	11
Radius .....	13
In-Band Management .....	15
Legal .....	16
Enterprise Management .....	17
General Practices .....	19
General Considerations.....	21
Configuration Archival.....	21
References.....	21
Appendices	
A. Template ( Cut & Paste ).....	22
B. Perl Script -- iosSanitize.pl.....	24
C. Expect Script -- cisco-cat-config.exp .....	26



## Distribution

This document can be released to **SPECIFIC** clients with approval from the Account Manager and Operations Manager.

## Document Update Process

The mechanics of an update process are TBD. However, this will require collaboration among many within Harford Technology Corporation and our clients. We will make certain that credit is given to the folks that help us keep the document up to date and aware of the latest issues

## Contributors

- 
- 
- 
- 

## Conventions

As we elaborate on the concepts and necessary IOS commands to realize the concepts, we will present the information in the following manner

**Bold** is used to display commands and options.

*Italic* is used to display arguments and options. These should be replaced locally defined values.

[ ] used to surround optional elements in the description of syntax

| used to separate optional arguments when illustrating a list where only one alternative may be chosen at a time.

< > used to illustrate one or more required parameters



---

## Configuration Information

The following sections illustrate what we believe to be the necessary configuration concepts to adequately protect the Catalyst OS device and maintain the necessary operational and administrative balance.

### Catalyst OS Services

Although many of the services present in Cisco's IOS can be tweaked, the Catalyst OS has been slow to adopt many of the capabilities due to many differences in the architecture and mission of the device.

Many of the following commands have appeared, as security has become a greater concern in the 21st Century

#### **set password set enablepass**

- The password is encrypted by default on the Catalyst 4XXX and 65XX
- Uses the MD5 Algorithm as opposed to the legacy encryption which was VERY easy to compromise ( even on the palm pilot )

<http://www.alcrypto.co.uk/cisco/>  
[http://www.boson.com/promo/utilities/getpass/getpass\\_utility.htm](http://www.boson.com/promo/utilities/getpass/getpass_utility.htm)  
<http://www.lopht.com/~kingpin/cisco.zip>

- ALL configurations should be sanitized prior to distribution!!! A small but capable Perl Utility is included in Appendix B at the end of the document.

#### **set cdp < enable | disable >**

- CDP is useful for some features in the NON-public side of a network, most notably, some of Cisco's VoIP products. However, as CDP distributes a variety of information regarding the network devices, this is rarely needed or advised in the public side of a network.
- Set this "on purpose" not by accident

#### **set logging timestamp < enable | disable >**

- It is generally advisable to have all devices logging in the same format and using the same time server/source ( if possible ). By doing so, if there is some sort of 'event', all device logs will be synchronized.

---

## Logging

It is generally advisable in most cases to enable logging. However, without proper planning, the information can be overwhelming.

### **set logging console < enable | disable >**

- This sends log information to the console or possibly a terminal server based on the site design. Although, sometime it can be **VERY** annoying, it can also be **VERY** informative.

### **set logging history < 1..500 >**

- logs information locally for diagnostic purposes

### **set logging server < enable | disable >**

- Enable/Disable system message logging to configured syslog servers

### **set logging server < ip address >**

- Send log information to a central syslog server. This can/will be an overwhelming amount of information.

### **set logging level all 4 default**

- Log all events of Severity Level 4 or greater.

### **set logging server *facility***

- Sets the facility for the syslog server. Generally set by the system administrator, or the Enterprise Management Team.

There are a variety of freeware tools available to manage the overwhelming amount of information that logging may produce. Below we have listed several tools that are currently available in the public domain/freeware realm.

Swatch –

<http://www.oit.ucsb.edu/~eta/swatch/>

Also available on the site is a white paper on the use of Swatch. This paper is available at <http://www.oit.ucsb.edu/~eta/swatch/lisa93.html>.

newsyslog –

<http://www.weird.com/~woods/projects/newsyslog.html>

Newsyslog is a highly configurable program for managing and archiving log files. The site listed above has a great deal of information regarding the installation and use of newsyslog.

---

## Authentication/Authorization/Accounting

This is generally related to tacacs+ and/or radius. In the Cisco realm, it is referred to as AAA, the Cisco published acronym for Authentication, Authorization, and Accounting.

### **tacacs+**

**set tacacs server a.b.c.d [ primary ]**  
**set tacacs server a.b.c.e**

- tacacs+ server addresses
- There may be more than one sever specified

**set authentication login tacacs enable console primary**  
**set authentication login tacacs enable telnet primary**  
**set authentication login tacacs enable http primary**  
**set authentication enable tacacs enable console primary**  
**set authentication enable tacacs enable telnet primary**  
**set authentication enable tacacs enable telnet primary**

- Enables tacacs+ authentication for login ( ie; console ) and enable mode. However, this model will fail to Cisco's legacy password system if the tacacs+ server is unreachable and/or unavailable.

**set tacacs key <key>**

- Encryption key for the 'back end' communications. Somewhat of a misnomer as the client traffic ( ie; telnet session ) is unencrypted.

**set tacacs timeout < seconds >**

- The default is 5 seconds which should be sufficient

**set tacacs attempts < number >**

- The suggested default for this parameter will be 3 attempts

**set tacacs directedrequest < enable | disable >**

- tacacs+ Directed Request may or may not be necessary based on the tacacs+ server deployment and design strategy. Until a need for Directed Request is identified it should be set to disabled

---

**set authorization exec enable tacacs+ none console**  
**set authorization exec enable tacacs+ none telnet**

- Authorization to start an exec session is arbitrated by the tacacs+ server and the user's configuration

**set authorization enable enable tacacs+ if-authenticated console**  
**set authorization enable enable tacacs+ if-authenticated telnet**

- Authorization to enter enable mode is arbitrated by the tacacs+ server and the user's configuration. However, this model will fail to Cisco's legacy password system if the tacacs+ server is unreachable and/or unavailable.

**set authorization commands enable all tacacs+ if-authenticated both**

- Authorization on a command level basis arbitrated by the tacacs+ server and the user's configuration. In the event the tacacs+ server is unreachable and/or unavailable the catalyst command set will be governed by cisco's legacy environment.

**set accounting connect enable start-stop tacacs+**

- Enable accounting for connection events

**set accounting exec enable start-stop tacacs+**

- Enable accounting for exec mode events

**set accounting system enable start-stop tacacs+**

- Enable accounting for system events

**set accounting commands enable all stop-only tacacs+**

- Enable accounting of configuration commands

**set accounting update new-info**

- Configure accounting to be updated as new information is available

---

## Radius

**set radius server a.b.c.d [ auth-port <port-number> ] [ primary ]**

- Radius Server Address
- There may be more than one Server Specified

**set radius key <key>**

- Encryption key for the 'back end' communications. Somewhat of a misnomer as the client traffic ( ie; telnet session ) is unencrypted.

**set authentication login tacacs enable [ console | telnet | both ]**

**set authentication enable tacacs enable [ console | telnet | both ]**

- Enables radius authentication for login and enable mode

**set radius timeout < seconds >**

- Sets the radius timeout interval
- The default is 5 Seconds

**set radius retransmit < count >**

- Set the number of times the switch will attempt to contact a radius server before attempting the next configured server
- The default is 2 tries

**set radius deadtime < minutes >**

- sets the radius server dead-time interval. If a server does not respond to a query, it is marked as dead and ignored for the specified number of minutes
- Ignored if only one radius server is configured

---

This represents the FULL tacacs+/radius model for the Catalyst OS. Please note that we have NO concept of a privilege level with the Catalyst prior to Version 6.. Therefore, the enable password will still be shared but controlled from the tacacs+/radius server.

In the latest versions of the Catalyst OS, we do have limited ability to set a privilege level, however, it is far less robust than the IOS.

One last point, notice that the tacacs+/radius model defaults to Cisco's legacy password system ( ie; Console & Enable ) if the tacacs+/radius server is unavailable. However, this is a **!!!VERY!!! SLOW** process.

At this time due to various issues within the Catalyst OS and the Radius specifications, tacacs+ is the only system that will provide command level Authorization and logging on a per user basis. Therefore, in a Cisco network we must recommend tacacs+ for the AAA system.

Future editions of this document will continue to provide information relating to radius such that we can all remain current in the developments of the AAA mechanics within the IOS.

---

## In-Band Management

### **set ip http server disable**

- Disable the capability to manage the device via http and a web interface

### **set ip permit enable**

- Enable the IP permit list

**set ip permit a.b.c.0 255.255.255.0 [ *telnet* | *ssh* | *snmp* ]**

**set ip permit d.e.f.0 255.255.255.0 [ *telnet* | *ssh* | *snmp* ]**

- This creates a list of IP addresses/subnets that are authorized to access the device. This is similar to a simple access list in the IOS.
- The functionality of the IP Permit list can also be achieved using VACLs. VACLs are handled by hardware (PFC – if present) and the processing is considerably faster under most circumstances. However, the introduction of VACLs strictly for this purpose may result in unnecessary complexity.

### **set snmp trap enable ippermit**

- Generate traps for unauthorized access attempts

### **set logging level ip 4 default**

- Generate syslog events for unauthorized access attempts

As ssh ( currently ssh V1 ) becomes available on the Catalyst platform, it is our belief that ssh should be the primary communications mechanism for interaction with the device.

---

## Legal

In general it is very useful to have a login banner such as the one below. In many cases that have resulted in litigation, the words in the login/banner messages have had great influence on the merit of the litigation.

Any banner message used across an enterprise of particularly a DMZ should be reviewed by Corporate Counsel and give away little or NO information regarding the name of the company, site, etc.

**set banner motd #**

**This system is for the use of authorized users only. Individuals using this system without authority, or in excess of their authority, are subject to having all of their activities on this system monitored and recorded by system personnel.**

**In the course of monitoring individuals improperly using this system, or in the course of system maintenance, the activities of unauthorized users may also be monitored.**

**Anyone using this system expressly consents to such monitoring and is advised that if such monitoring reveals possible evidence of criminal activity, system personnel may provide the evidence of such monitoring to law enforcement officials.**

**#**

---

## Enterprise Management

There are many variables to consider in the architecture of a Network Management System. However, the following should be sufficient as a starting point for the Catalyst OS devices in such architectures.

### **set snmp community read-only <RO\_String>**

- This sets up the community strings for the snmp process. In this case it sets up a READ-ONLY snmp community. If the IP Permit list is enabled it can be leveraged such that ANY snmp request must pass the VACL before being processed.

### **set snmp community read-write <RW\_String>**

- This sets up the community strings for the snmp process. In this case it sets up a READ-WRITE snmp community. If the IP Permit list is enabled it can be leveraged such that ANY snmp request must pass the VACL before being processed.

### **set snmp community read-write-all < RWA\_String >**

- This sets up the community strings for the snmp process. In this case it sets up a READ-WRITE-ALL snmp community. If the IP Permit list is enabled it can be leveraged such that ANY snmp request must pass the VACL before being processed.

### **set snmp view { NAME } { OID String } [ *included* | *excluded* ] [ *volatile* | *nonvolatile* ]**

- This sets up a view within the device's MIB. By utilizing a view within the Catalyst OS one can limit the visibility of an snmp request to a particular OID as opposed to the entire MIB. The included option is useful if the desired option is to grant access to a 'branch' within the MIB.

One note. In many cases with the proper planning, the READ-WRITE and READ-WRITE-ALL communities can be eliminated completely. This reduces many vulnerabilities, however, in some cases it may not be possible to eliminate the READ-WRITE community entirely. If presented with such circumstances, it is likely that after understanding the requirements, the snmp view can be leveraged in conjunction with the IP Permit List to mitigate some, if not significant risk.

---

**set snmp trap [ enable | disable ] [ all | auth | bridge | chassis | config | entity | ippermit | module | repeater | stpx | syslog | vmps | vtp ]**

- Enables and/or disables the different snmp traps on the system

**set snmp trap < receiver\_Address > < receiver\_Community >**

- Specifies the address of the trap receiver and the community to use when sending authentication traps

---

## General Practices

### **set system name < name string >**

- This configures a name for the system
- Also, if used in this fashion, it will set the prompt to the system name

### **set prompt < prompt string >**

- This configures the prompt for the CLI. Unnecessary if the **set system name** command has been used.

If we create a convention such that ALL system names are lower ( or upper ) case and ALL prompts are the system name, the administration of large networks becomes easier in that we can automate a variety of tasks.

### **set protocolfilter [ *enable* | *disable* ]**

- Activates ( or Deactivates ) protocol filtering on Ethernet VLANs and nonTrunking Ethernet, Fast Ethernet, and Gigabit Ethernet ports.

### **set port protocol *mod/port* < ip | ipx | group > < on | off | auto >**

- *set port protocol mod/port ip on*
- *set port protocol mod/port ipx off*
- Enables ( or Disables ) protocol membership on the specified port
- For Instance, why would we want IPX flood traffic sent to the port of a mission critical Unix Server? Admittedly this may be overkill in some areas, however, in mission critical or high security areas it is advisable.

### **set port disable *mod/port***

- Disable ANY unused ports. Definitely an aid to Physical Security.
- This can be very effective in high security applications such as a DMZ.

### **set port name *mod/port* [ *port\_name* ]**

- Configures a name for a port. Extremely informative, if correct. This is one area that some effort can go a long way. Particularly in high profile and high security areas.

**set port security mod\_num/port\_num enable < mac\_address >**

- In high security AND static configuration this is a significant aid to physical security in that only the mac address specified will function in the port.
- This may be overkill across the enterprise but should be useful in many areas with high profile/static configurations such as a high security area or DMZ.

## General Considerations

The information and concepts listed above are merely a starting point. No amount of policy and/or procedures can substitute for due diligence.

For instance, Cisco typically releases known security problems with work – arounds between Versions. It is the administrator's duty to determine the need for these patches. This may be a balance between Operational Issues as well as Security Needs.

Also, there are MANY mailing lists such as SAFER, Cisco, and SANS, that distribute a great deal of information for Cisco's IOS and the related threats, new attacks, and new methods of security.

## Configuration Archival

A configuration archive should be maintained for all devices in the Network Infrastructure. Also, this archive can be leveraged toward change control, as it can be

the authoritative archive for back-out plans and configurations. This is not only related to security, but to 'General Operations Best Practices' as well.

A simple but capable Expect Script is included in Appendix E at the end of the document. Expect is available at <http://expect.nist.gov>.

## References

Solaris Security, v 1.0, Sans Institute, <http://www.sans.org>  
<http://www.cisco.com/univercd/home/home.htm>  
<http://www.cisco.com/warp/customer/707/21.html>

## Appendix A

### *Configuration Template –*

In the event that this document is provided in electronic format the following is a collection of the concepts and commands introduced earlier. However, this is set up such that one can 'cut and paste' the individual sections into their editor of choice.

```
set password
set enablepass
set cdp < enable | disable >
set logging timestamp < enable | disable >
set logging console < enable | disable >
set logging history < 1..500 >
set logging server <ip address>
set logging level all 4 default
set logging server facility

set tacacs server a.b.c.d primary
set tacacs server a.b.c.e

set authentication login tacacs enable console primary
set authentication login tacacs enable telnet primary
set authentication login tacacs enable http primary
set authentication enable tacacs enable console primary
set authentication enable tacacs enable telnet primary
set authentication enable tacacs enable telnet primary

set tacacs key something@somewhere.com

set tacacs timeout 5
set tacacs attempts 3

set tacacs directedrequest disable

set authorization exec enable tacacs+ none console
set authorization exec enable tacacs+ none telnet

set authorization enable enable tacacs+ if-authenticated console
set authorization enable enable tacacs+ if-authenticated telnet

set authorization commands enable all tacacs+ if-authenticated both

set accounting connect enable start-stop tacacs+
set accounting exec enable start-stop tacacs+
set accounting system enable start-stop tacacs+
set accounting commands enable all stop-only tacacs+

set accounting update new-info

set ip http server disable
```



---

```
set ip permit enable
set ip permit a.b.c.0 255.255.255.0
set ip permit d.e.f.0 255.255.255.0

set snmp trap enable ippermit
set logging level ip 4 default

set banner motd #
This system is for the use of authorized users only. Individuals using
this system without authority, or in excess of their authority, are
subject to having all of their activities on this system monitored and
recorded by system personnel.

In the course of monitoring individuals improperly using this system, or
in the course of system maintenance, the activities of unauthorized users
may also be monitored.

Anyone using this system expressly consents to such monitoring and is
advised that if such monitoring reveals possible evidence of criminal
activity, system personnel may provide the evidence of such monitoring to
law enforcement officials.

#
set snmp community read-only <RO_String>
set snmp community read-write <RW_String>
set snmp community read-write-all <RWA_String>
set snmp view < name > < OID String > [ included | excluded ] [ volatile | non-volatile ]
set snmp trap [ enable | disable ] [ all | auth | bridge | chassis | config | entity |
  ippermit | module | repeater | stpx | syslog | vmps | vtp ]
set snmp trap < a.b.c.d > < receiver_Community >

set system name < string >
set prompt < prompt string >
set protocolfilter [ enable | disable ]

set port protocol mod/port < ip | ipx | group > < on | off | auto >
set port protocol mod/port ip on
set port protocol mod/port ipx off

set port disable mod/port
set port name mod/port [ port_name ]
set port security mod_num/port_num enable < mac_address >
```

## Appendix B

### *Perl Script: iosSanitize.pl –*

The following Perl script takes a normal configuration and sanitizes it by removing all ( type 5, type 7, and unencrypted ) of the passwords and replacing them with a pattern of repeating XxXxXx. The script has only one parameter, the name of the original configuration file. For example, if the original configuration is named router-confg, the result will be a file with .distribution as a suffix ( i.e.; switch-confg.distribution ).

```
#!/opt/local/bin/perl -w
#
# CatSanitize.pl -- remove Cisco type 7 passwords from IOS configs
#
# usage:
#
# CatSanitize.pl <config file name>
#
# created on Monday, May 19, 2003 by Mark Leighty ( mleighty@harfordtechnology.com )
#
# Copyright (2003) Harford Technology Corporation. With the exception of commercial
# resale, lease, license or other commercial transactions, permission is granted to use,
# copy, modify, and distribute this software. By exercising this permission you agree, that
# this Notice will accompany this software at all times.
#
# Harford Technology Corporation MAKES NO REPRESENTATIONS OR WARRANTIES OF
# ANY KIND CONCERNING THIS SOFTWARE OR USE THEREOF.
#
#----- Revision Log -----#
#$Version = 0.89;                # mleighty@harfordtechnology.com
# Currently "pre-production" however as features get added and we get
# some data it should be functional.
#----- End Revision Log -----#

open( ORIGINAL, "$ARGV[0]" ) or die "can't open $ARGV[0]";
open( SANITIZED, ">$ARGV[0].distribution" ) or die "can't open $ARGV[0].distribution";

while (<ORIGINAL> ) {
    if (/^set password/) {
        print SANITIZED "set password XxXxXxXxXxXxXxXxXxXxXxXx\n";
    } elsif (/^set enablepass/) {
        print SANITIZED "set enablepass XxXxXxXxXxXxXxXxXxXxXxXx\n";
    } elsif (/^set tacacs server/) {
        print SANITIZED "set tacacs server a.b.c.d\n";
    } elsif (/^set tacacs key/) {
        print SANITIZED "set tacacs key XxXxXxXxXxXxXxXxXxXxXxXx\n";
    } elsif (/^set snmp community/) {
        print SANITIZED "set snmp community read_XXX xXxXxXxXxXxXxXxXxXxXxXx\n";
    }
}
```

---

```
} elseif (/^set snmp trap .*able/){  
    print SANITIZED $_  
} elseif (/^set snmp trap/){  
    print SANITIZED "set snmp trap a.b.c.d xxxxxx\n";  
} else {  
    print SANITIZED $_  
}  
}  
close ORIGINAL;  
close SANITIZED;  
exit
```

## Appendix C

### *Expect Script: cisco-cat-config.exp*

The expect script below uses the Expect scripting language and a telnet session to capture the Catalyst's configuration and store it in a/the specified directory.

```
#!/opt/local/bin/expect --
#
# cisco-cat-config-notftpatm.exp -- configuration download for Cisco Catalyst Switches
#
# usage:
#
# cisco-cat-config-notftpatm.exp tacacs+ID tacacs+passwd current.vty current.enable mail
#
# created on Monday, May 19, 2003 by Mark Leighty ( mleighty@harfordtechnology.com )
#
# Special thanks to Don Libes and NIST for the creation of Expect!!!!
#
# Copyright (2003) Harford Technology Corporation. With the exception of commercial
# resale, lease, license or other commercial transactions, permission is granted to
# use, copy, modify, and distribute this software. By exercising this permission you
# agree, that this Notice will accompany this software at all times.
#
# Harford Technology Corporation.  MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND
# CONCERNING THIS SOFTWARE OR USE THEREOF.
#
# Variable Definitions
#
# argv[0] --> tacacs+ user ID
# argv[1] --> tacacs+ Password
# argv[2] --> current vty password ( ie; console password )
# argv[3] --> The current enable password
# argv[4] --> email for error notifications
#
# ----- #
# Revision Log #
# ----- #
set Version 0.96 ;# mleighty@harfordtechnology.com
# ----- #
# End Revision Log #
# ----- #

#
# Check usage ... if argv < 3 exit with usage display
#
if { $argc < 3 } {
    send_error "\n"
    send_error "\n"
    send_error "Usage:\n"
```



```
send_error "\n"
send_error "cisco-cat5-config-notftpatm.exp tacacs+ID tacacs+passwd vty enable mail"
send_error "\n"

send_error "\n"
send_error "Example\n"
send_error "\n"
send_error "cat5-config /home/cisco/switches tacacs+ID tacacs+passwd vty enable \
/home/cisco/switches/config/08may97" jsmith@somewhere.com
send_error "\n\n"
}

#
# Set up variables passed via the command line in argv
#

set tacacsID          [ lindex $argv 0 ]      ;# tacacs+ user ID
set tacacsPasswd     [ lindex $argv 1 ]      ;# tacacs+ password
set vty               [ lindex $argv 2 ]      ;# current vty ( console ) pasword
set enable            [ lindex $argv 3 ]      ;# current enable password
set MAIL              [ lindex $argv 4 ]      ;# email account for error reporting

set console_prompt   "> $"                   ;# pattern match for console prompt
set enable_prompt    "\\(enable\\) $"        ;# pattern match for enable prompt

set the_date         [ timestamp -format %d%b%Y ] ;# set date for archival purposes
set file             [ open "/opt/Config/etc/switches" "r" ] ;# Open file specified to have list of
switches

set timeout -1                               ;# handle switch timing issues

while { [ gets $file cat5 ] > 0 } {           ;# Set up looping structure to process each switch in the file
    if [ catch { spawn telnet $cat5 } msg ] { ;# Start telnet process to the specified switch
        exec echo "Subject: Config Backup Failure - $cat5\n$cat5\n$expect_out(buffer)" | \
            /usr/lib/sendmail $MAIL
        continue
    }
}

set OK               1                       ;# set up logic to break out of while loop and continue. This is
;# unfortunately necessary as NOTHING is consistent in terms of
;# prompting or the existence of tacacs+. Also timeout values for
;# tacacs+ are a source of contention causing AV pairs to fail
;# necessitating a looping structure ( in the event tacacs+ is
;# present ).

#
# look for a response ...
#
# if it's "enter password: " we'll go forward
# if it's something like "Could not ..." we'll send a messages to STD_ERROR
# and bail out or move on to the next router.
#

while 1 {
    expect {
        -re "Username: $"          { send "$tacacsID\r" }
        -re "Password: $"         { send "$tacacsPasswd\r" }
        -re "Enter password: $"   { send "$vty\r" }
        -re "Enter Password: $"   { send "$vty\r" }
```



---

```
-re "TACACS.*Enter Password: $"      { send "$vty\r" }
-re "Enter password: $"              { send "$vty\r" }
-re $console_prompt                  { break }
default                               { set OK 0; break }
}
}
if !$OK {
  exec echo "Subject: Config Backup Failure - $cat5\n$cat5\n$expect_out(buffer)" | \
    /usr/lib/sendmail $MAIL
  continue
}
match_max 200000                      ;# raise the matching buffer... capture the entire conf
send "enable\r"                       ;# change to privileged mode
expect -re "Password: $|Enter password: $" ;# wait for a prompt
send "$enable\r"                      ;# send the enable password
expect -re $enable_prompt              ;# expect the enable prompt
set config [ open "/tftpboot/$cat5-config.$the_date" "w" ] ;# set up output file host-config.ddmmy
send "set length 0\r"                  ;# Adjust terminal settings ... eliminate --More--
expect -re $enable_prompt              ;# Wait for the enable prompt
send "write term\r"                   ;# display the running config
expect -re $enable_prompt              ;# Wait for the enable prompt
regexp "(begin.*end)" $expect_out(buffer) test ;# get rid of all of the extras
puts $config $test                    ;# send Output of "wr t" to the designated file
flush $config                          ;# flush the I/O channel just in case
close $config                          ;# close switch-config output for this switch
close
wait                                   ;# make sure the child process is dead so we dont have pty problems
}                                       ;# end of while loop
close $file                             ;# Close the input file ( list of switches )
exit                                    ;# exit the expect shell ... we're done
```

