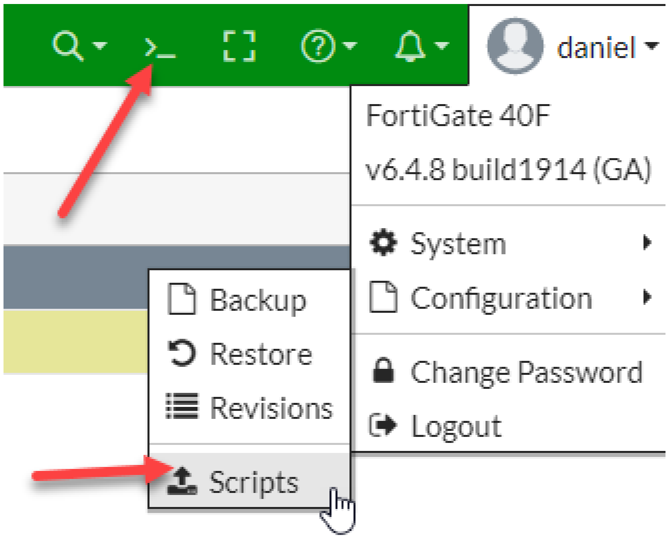
<u>CIS Firewall Alias Script Revised Documentation</u>

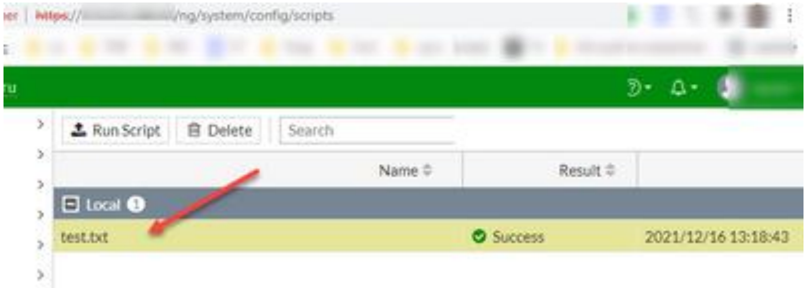
Step 1: Create Documentation

Create an alias from which the next can get a list of information

Part A – Load Script

Go to the Fortinet GUI and upload the script "cis_audit_script"





Name	Result	
test.txt	Success	2021/12/16 13:18:43

If this does not work, then simply paste the script into the CLI… this part must be done from an admin account

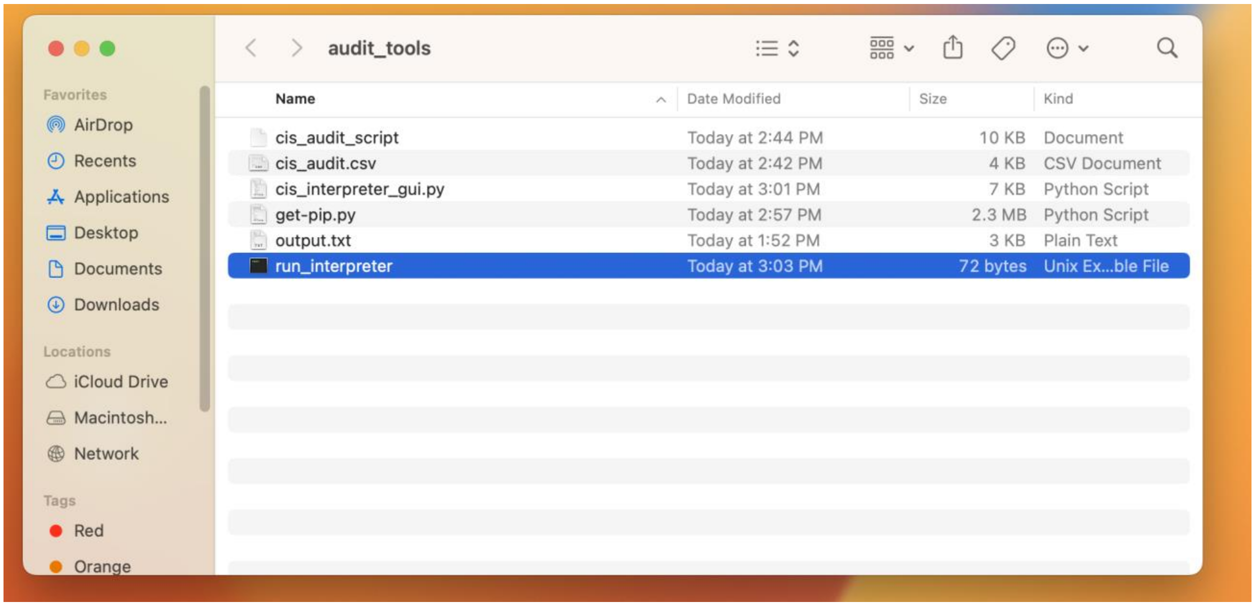
Part B – Run Script

Go into CLI and type "alias cis_audit"

Part C – Process the Output

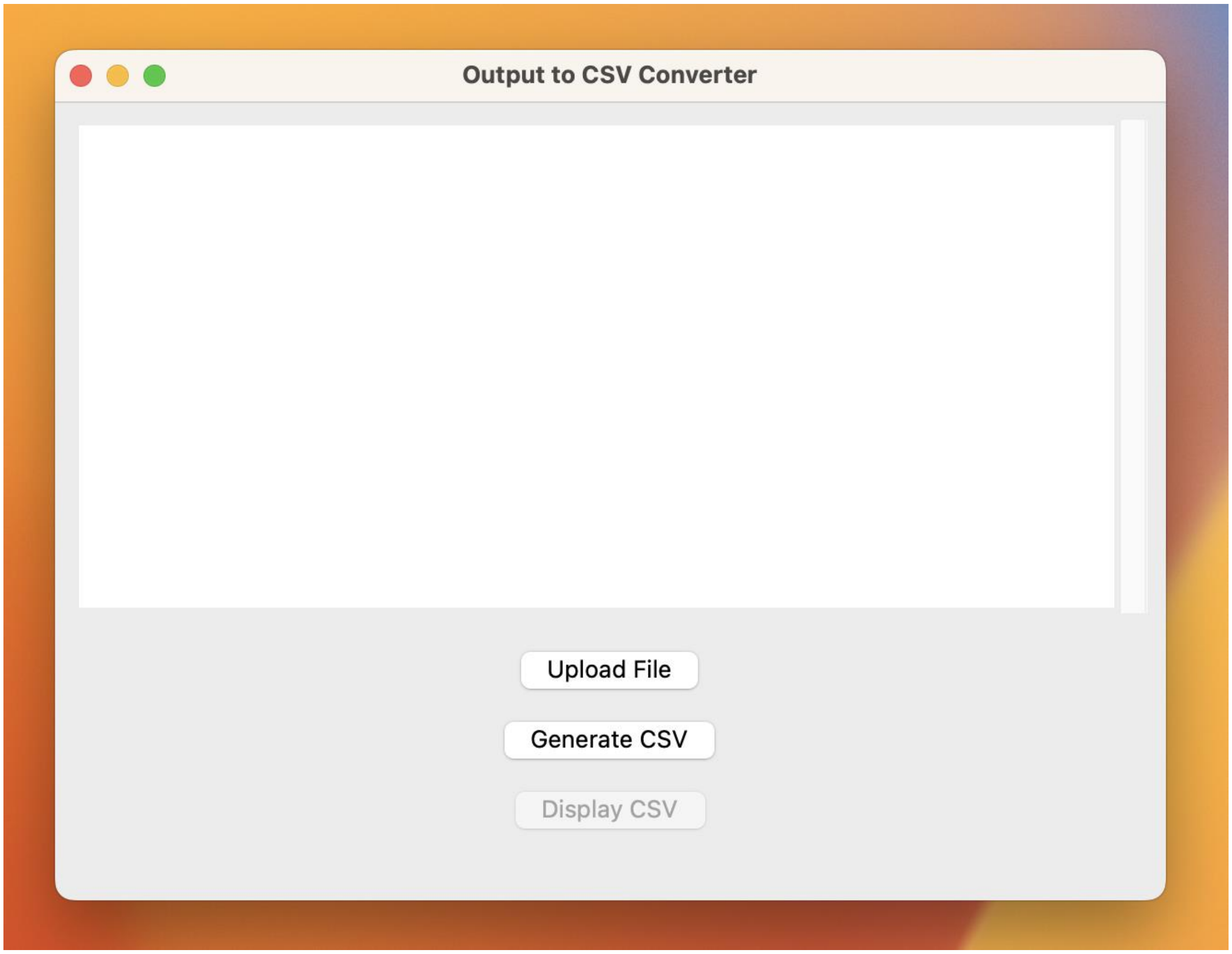
Save the output.

Double click the “run_interpreter” executable



Name	Date Modified	Size	Kind
cis_audit_script	Today at 2:44 PM	10 KB	Document
cis_audit.csv	Today at 2:42 PM	4 KB	CSV Document
cis_interpreter_gui.py	Today at 3:01 PM	7 KB	Python Script
get-pip.py	Today at 2:57 PM	2.3 MB	Python Script
output.txt	Today at 1:52 PM	3 KB	Plain Text
run_interpreter	Today at 3:03 PM	72 bytes	Unix Ex...ble File

Use either the text box or the “upload file” button to upload the script output, then click “Generate CSV”



Congrats! You’ve finished the audit. Use the CSV file to log.

[Continue to Page 2 for Technical Writeup]

Explanation:

Overview:

This is a script that creates many different alias commands on the fortinet CLI in order to automate the CIS auditing process. The script strategically utilizes grep in order to curate the quantity of returns for each section to a desired quantity. If there are too little or in some cases too many returns, then the conditions are not met to conclude that the benchmark has been met. However, if the quantity of returns follows the provided guidelines, then it has met the standard. The script causes it to become easy to discern which standards are met by this method. Furthermore, the key which is detailed in the appendix can be programmatically used to make the auditing process even easier.

Parts of the script:

Loading "Documentation":

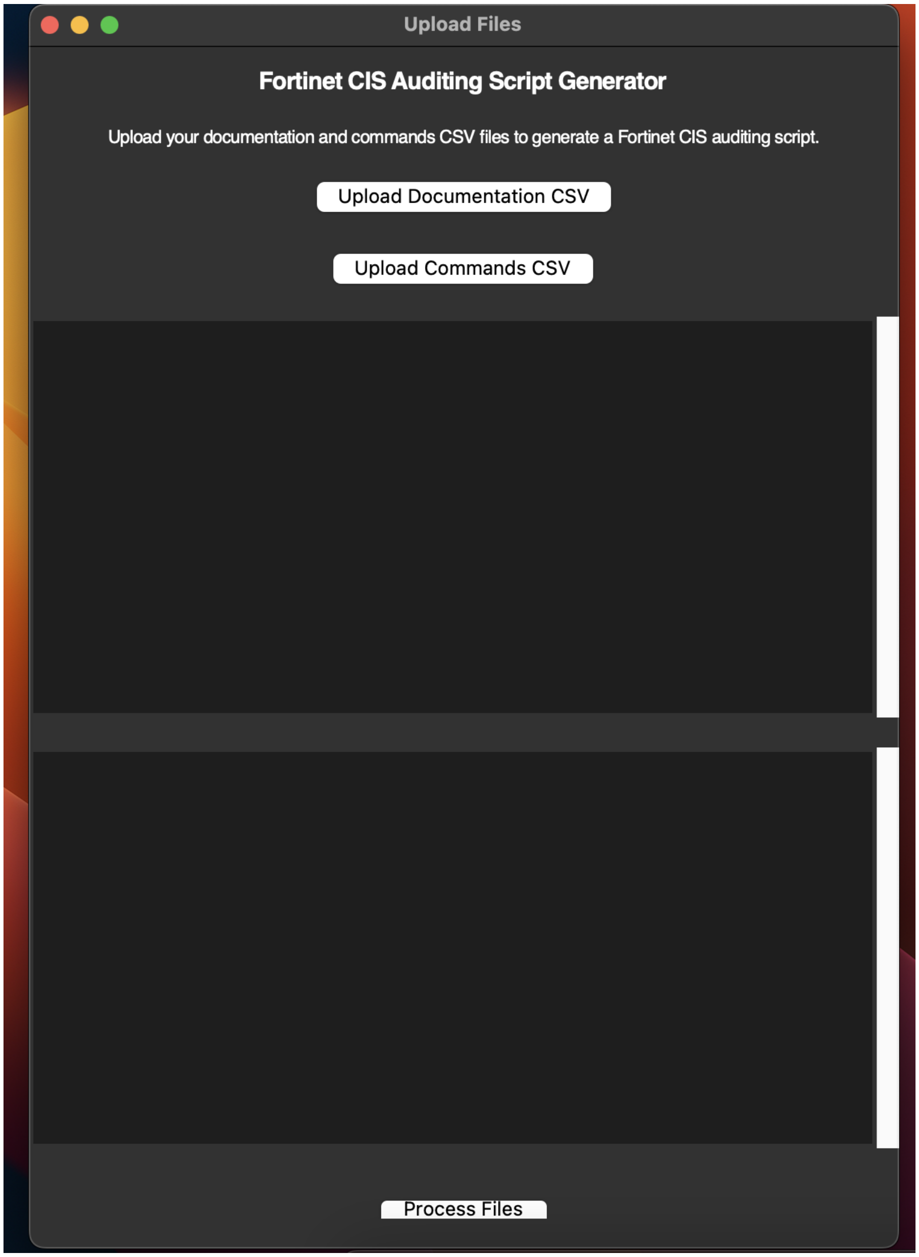
The first part of the script loads the documentation for each CIS standard. This is necessary since there is no print command in the Fortinet CLI. Furthermore, while storing this documentation in aliases may have made it more difficult to script, it allows the user to simply delete the script after two commands [config system alias, purge].

Running Alias Commands:

The second part of the script greps the documentation to get the standard number and display its requirements, then runs the respective commands for each standard to ensure that the benchmark is met. Creating the script using alias scripts had strategic tradeoffs because it made the script more difficult to write, but allowed it to be tested in sections, more easily debugged, and easily ran multiple times.

How to Add (Easy Way):

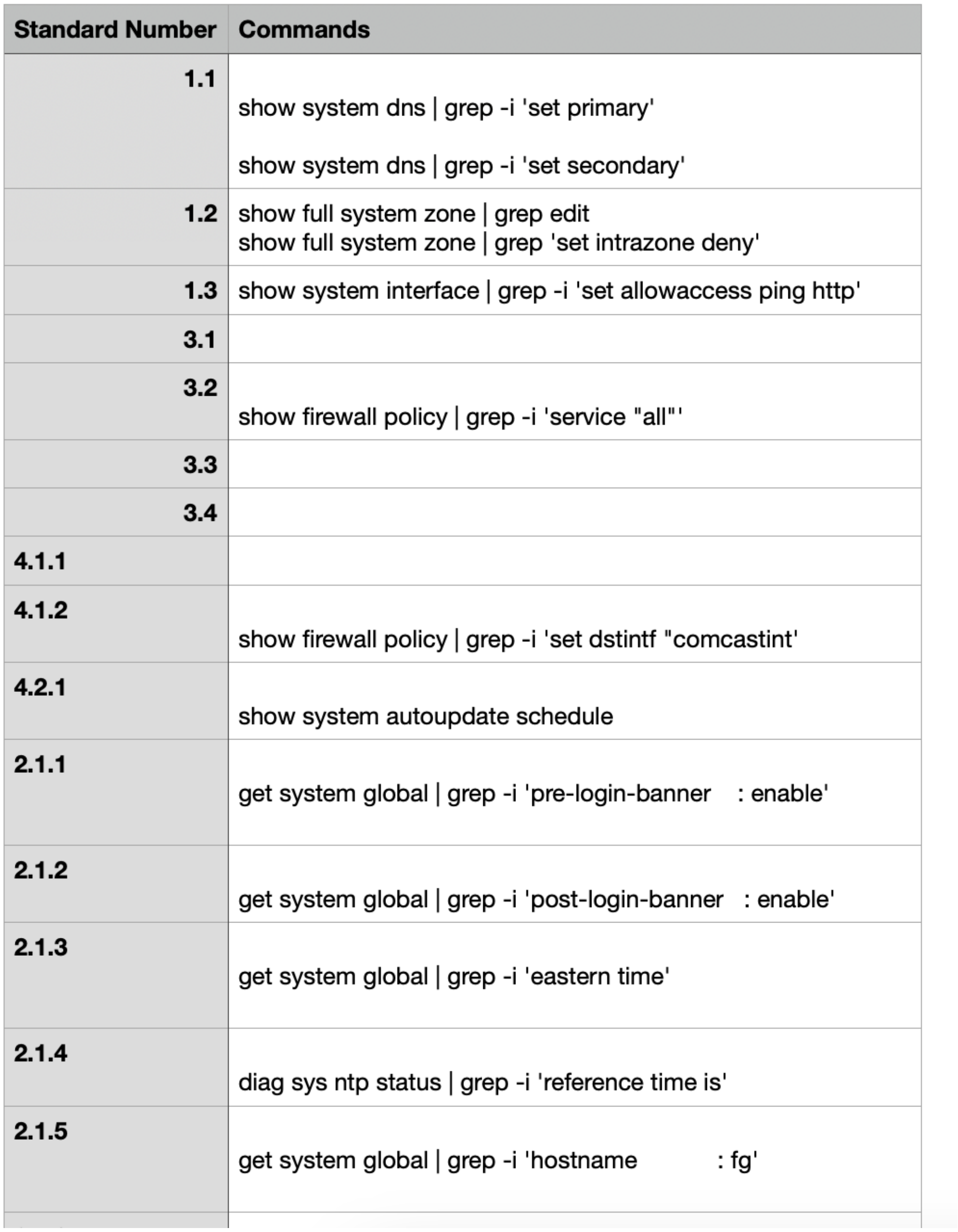
1. **Run the script update tool**



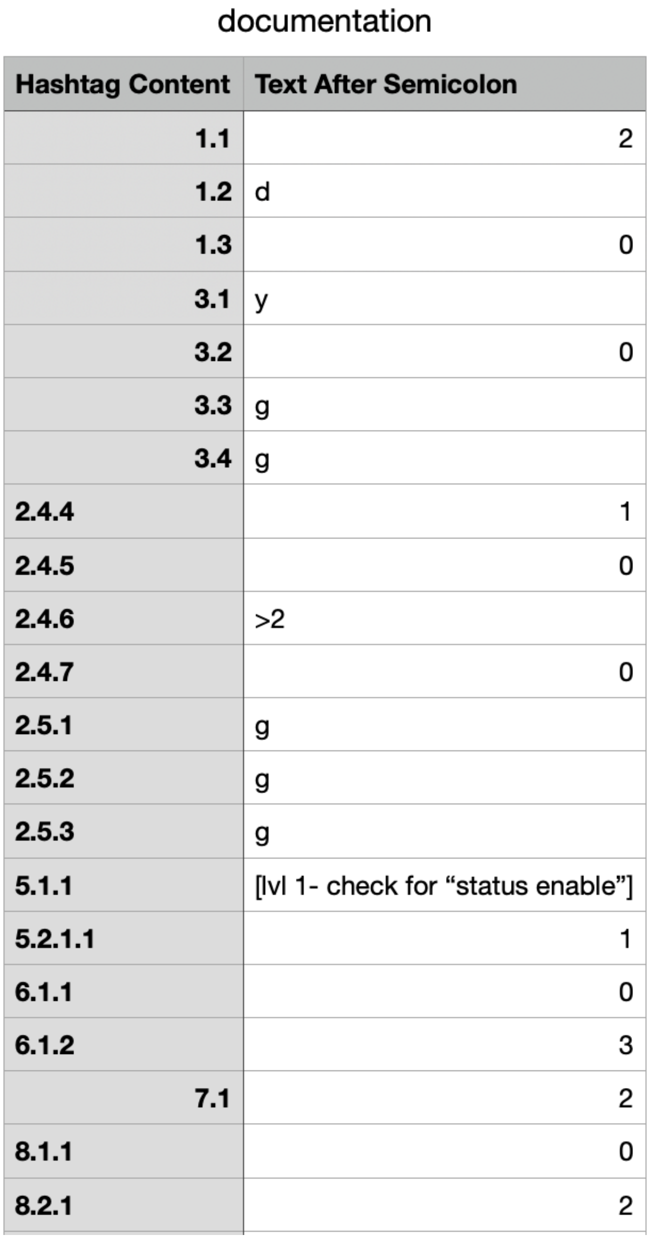
Fortinet CIS Auditing Script Generator

Upload your documentation and commands CSV files to generate a Fortinet CIS auditing script.

2. Assure commands excel sheet and documentation excel sheet are updated csv files
   To see how to make commands, see technical writeup sections 1 and 2



Standard Number	Commands
1.1	show system dns | grep -i 'set primary' show system dns | grep -i 'set secondary'
1.2	show full system zone | grep edit show full system zone | grep 'set intrazone deny'
1.3	show system interface | grep -i 'set allowaccess ping http'
3.1	
3.2	show firewall policy | grep -i 'service "all"'
3.3	
3.4	
4.1.1	
4.1.2	show firewall policy | grep -i 'set dstintf "comcastint'
4.2.1	show system autoupdate schedule
2.1.1	get system global | grep -i 'pre-login-banner : enable'
2.1.2	get system global | grep -i 'post-login-banner : enable'
2.1.3	get system global | grep -i 'eastern time'
2.1.4	diag sys ntp status | grep -i 'reference time is'
2.1.5	get system global | grep -i 'hostname : fg'



documentation

Hashtag Content	Text After Semicolon
1.1	2
1.2	d
1.3	0
3.1	y
3.2	0
3.3	g
3.4	g
2.4.4	1
2.4.5	0
2.4.6	>2
2.4.7	0
2.5.1	g
2.5.2	g
2.5.3	g
5.1.1	[lvl 1- check for “status enable”]
5.2.1.1	1
6.1.1	0
6.1.2	3
7.1	2
8.1.1	0
8.2.1	2

3. Upload the files and click “process files”

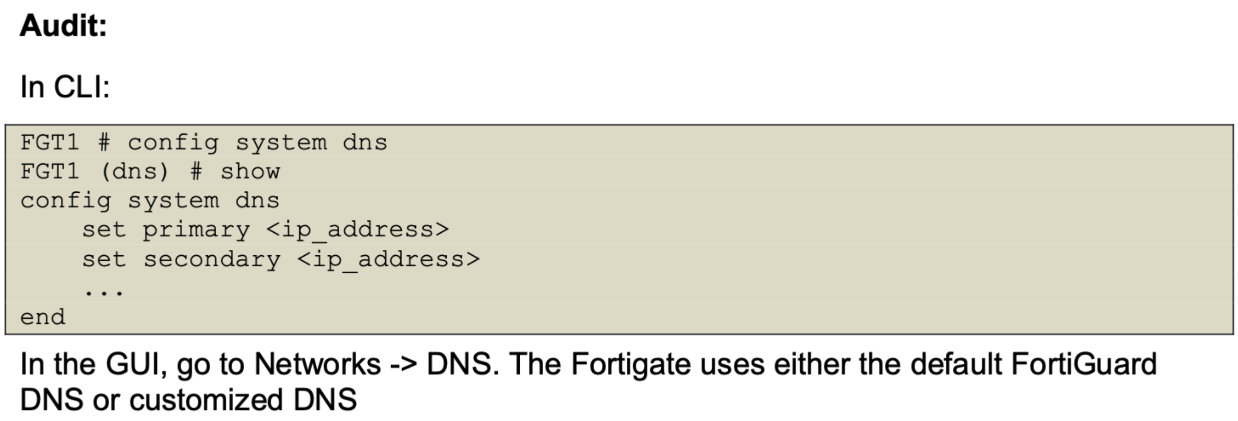
Congrats! You have successfully transformed your excel sheet into an audit script!

How to Add (Technical Writeup):

Identify the Commands Needed

Example: *Benchmark 1.1*

1. Review the commands you are being told to run, and what **specifically** is being looked for.



Audit:

In CLI:

```
FGT1 # config system dns
FGT1 (dns) # show
config system dns
    set primary <ip_address>
    set secondary <ip_address>
    ...
end
```

In the GUI, go to Networks -> DNS. The Fortigate uses either the default FortiGuard DNS or customized DNS

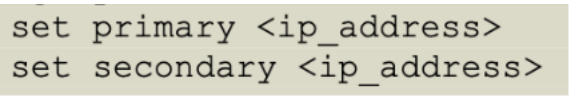
2. Rewrite this command into one or many commands so that the number of lines will always be set to a specific number, if the requirement is met.
    a. Rewrite the command not to use config. This can be used by replacing config with "show" or "show full" respectively.

In this case, we can compact the command into the following:

```
Show system dns
```

b. Use “grep” to systematically filter the output. We do this so that we can know whether or not the output has the text to which we are looking for.

In this case, we want to see if there are lines with the following text:



```
set primary <ip_address>
set secondary <ip_address>
```

The “grep” command line tool searches for a string and then prints out the line that the string exists. If the string does not exist, there is no line printed. We take advantage of this tool to detect whether or not there is a primary and secondary DNS server set.

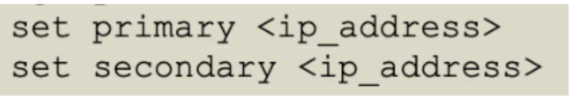
We want to isolate the output to **only** the lines that we want.

```
show system dns | grep –i ‘set primary’
show system dns | grep -i 'set secondary'
```

it is important* ***not* *to use double quotes in your command. If needed use the escape key, \”*

When run, these commands will **only** show the desired 2 lines if the requirement is met.

The output **should** look like this **if and only if the requirement is met**:



```
set primary <ip_address>
set secondary <ip_address>
```

c. Take note of the number of lines expected to be output.

In this case, we expect 2 lines.

3. Put the new command into the script
   a. Add your documentation to the script

Either create a new documentation alias, or add to an existing one; however, be aware that the **documentation files have a character limit of 255**. It will break if you go over. To make the documentation more in depth, we will create a new documentation file; however, you may not always want to create a new one, as they are organized numerically by sections.

Go into the cis_audit_script.txt and at the **top** of the file, type, out

```
config system alias

edit <cis_documentation_name_doc>

set command "
#<standard_number> ;<num_expec_outputs or letters from interpretation key>
"
end
```

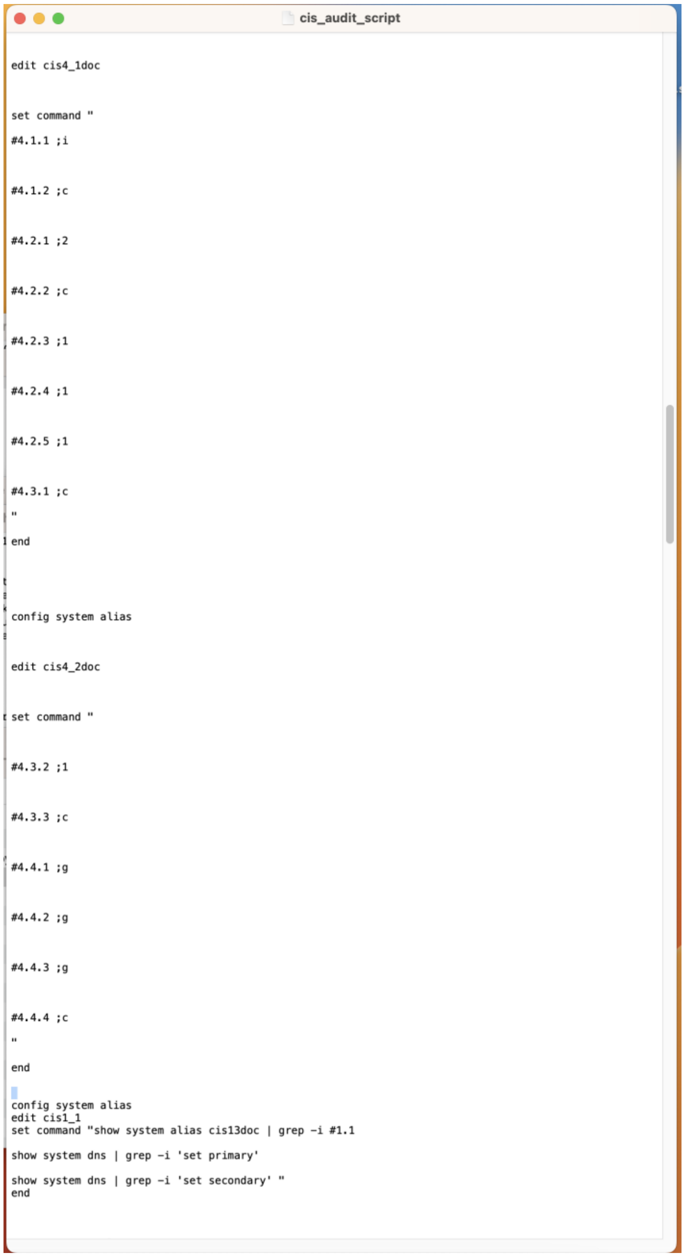
```
config system alias

edit cis1.1doc

set command "
#1.1 ;2
"
end
```

**It is more efficient to put many documentations in one alias*

b. Add your commands to the script

Either create a new command alias, or add to an existing one; however, be aware that the **alias commands have a character limit of 255**. It will break if you go over. To make the documentation more in depth, we will create a new command file; however, you may not always want to create a new one, as they are organized numerically by sections.

We will do this in the script file after the end of the documentation files.



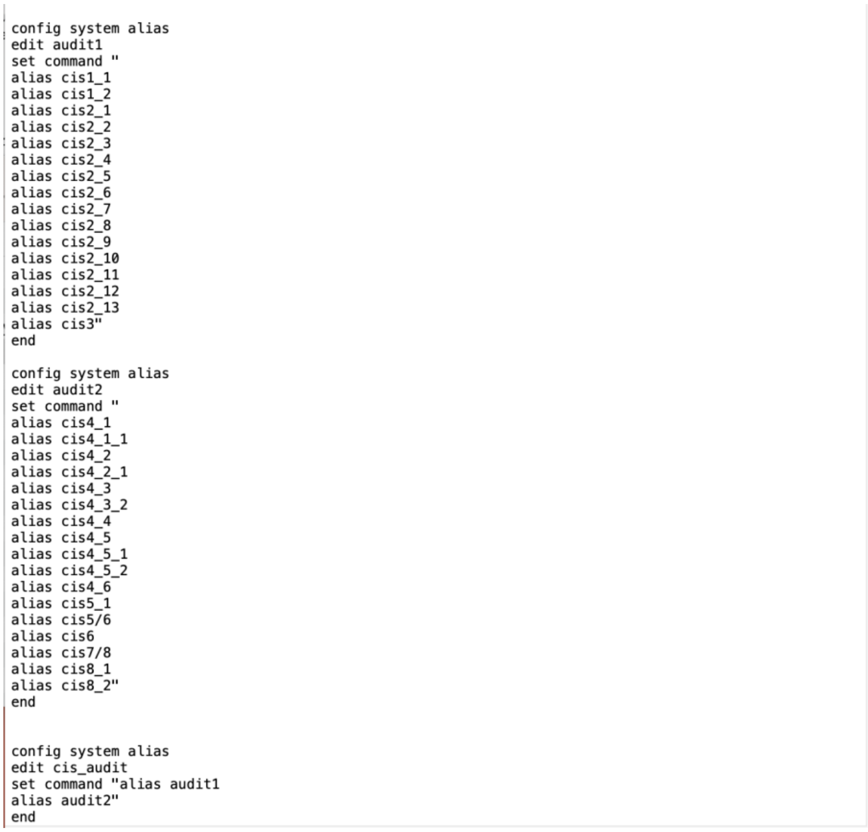
```
edit cis4_1doc


set command "
#4.1.1 ;i


#4.1.2 ;c


#4.2.1 ;2


#4.2.2 ;c


#4.2.3 ;1


#4.2.4 ;1


#4.2.5 ;1


#4.3.1 ;c
"
end



config system alias


edit cis4_2doc


set command "


#4.3.2 ;1


#4.3.3 ;c


#4.4.1 ;g


#4.4.2 ;g


#4.4.3 ;g


#4.4.4 ;c
"

end

config system alias
edit cis1_1
set command "show system alias cis13doc | grep -i #1.1

show system dns | grep -i 'set primary'

show system dns | grep -i 'set secondary' "
end
```

```
Config system alias
Edit <cis_command_alias_name>
Set command “
show system alias <cis_documentation_name_doc> |  grep #1.1
<your commands>
“
```

```
config system alias
edit cis1.1
set command "show system alias cis1.1doc | grep -i #1.1

show system dns | grep -i 'set primary'

show system dns | grep -i 'set secondary' "
end
```

c. Add your command to the master command list in its order

“Command f” to find either “audit1” or “audit2” or “cis_audit” and add your alias command to the final command list. Do this step in order, if you want the output to be in order.



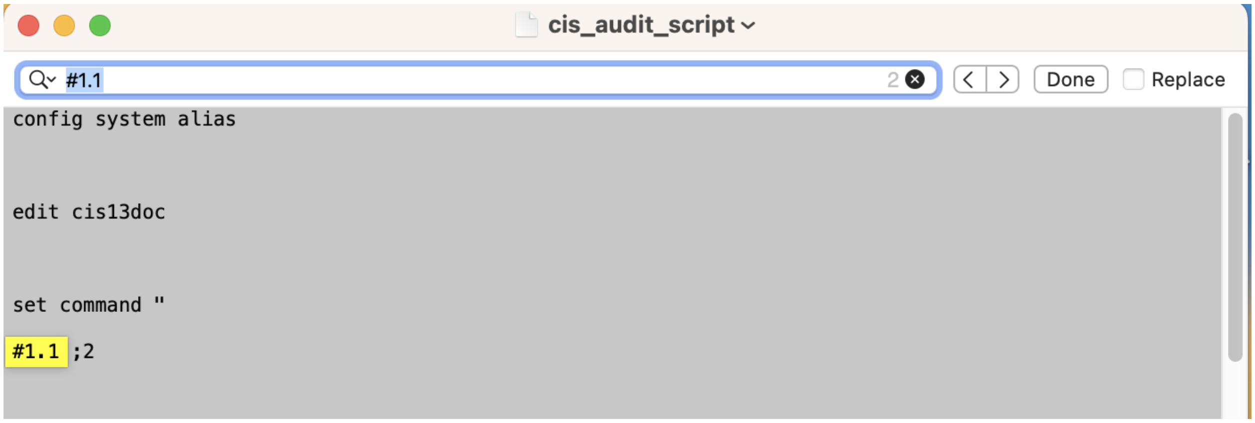
```
config system alias
edit audit1
set command "
alias cis1_1
alias cis1_2
alias cis2_1
alias cis2_2
alias cis2_3
alias cis2_4
alias cis2_5
alias cis2_6
alias cis2_7
alias cis2_8
alias cis2_9
alias cis2_10
alias cis2_11
alias cis2_12
alias cis2_13
alias cis3"
end

config system alias
edit audit2
set command "
alias cis4_1
alias cis4_1_1
alias cis4_2
alias cis4_2_1
alias cis4_3
alias cis4_3_2
alias cis4_4
alias cis4_5
alias cis4_5_1
alias cis4_5_2
alias cis4_6
alias cis5_1
alias cis5/6
alias cis6
alias cis7/8
alias cis8_1
alias cis8_2"
end


config system alias
edit cis_audit
set command "alias audit1
alias audit2"
end
```

In this case to either one of the alias commands, we will add, “alias cis1.1”

Congrats! You’ve successfully added to the CIS audit script.

How to Update:

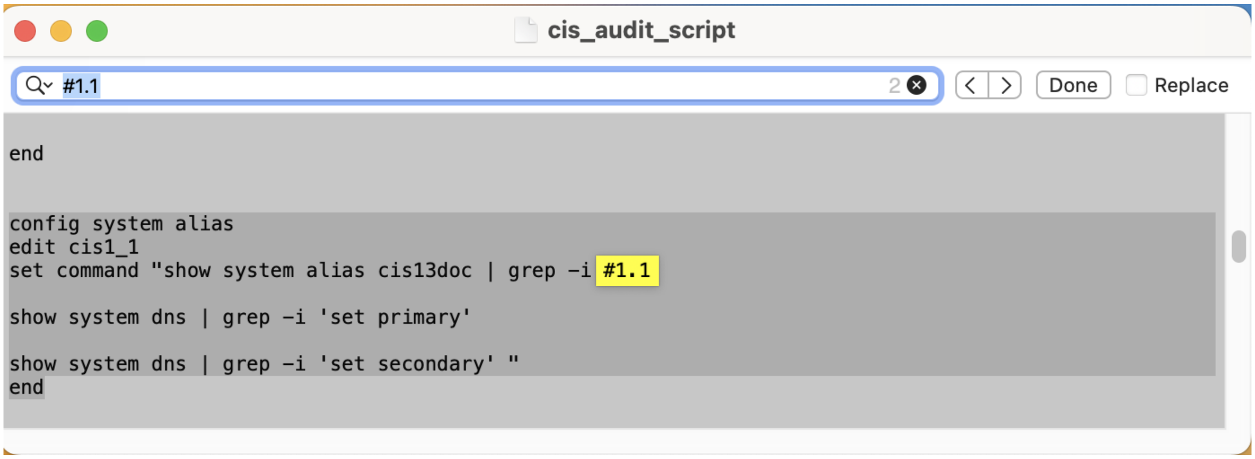
1. Go to cis_audit_script and "command f" to search the documentation number you wish to update preceded by a "#"



```
config system alias


edit cis13doc


set command "

#1.1 ;2
```

Update documentation here with respective key

2. Go to the next occurence of "#1.1"



```
end

config system alias
edit cis1_1
set command "show system alias cis13doc | grep -i #1.1

show system dns | grep -i 'set primary'

show system dns | grep -i 'set secondary' "
end
```

Change the commands underneath the command "show system alias cis13doc | grep -i #1.1"

Congrats! You've updated the cis_audit_script!

Appendix:

Cis script interpretation key:

;0 requires no return for yes

;1 requires 1 return for yes

;2 requires 2 returns for yes

;3 requires 3 returns for yes

;y always yes

;g need to use GUI (not automated)

;>2 requires at least 2 returns for yes

;m output needs manual review for yes

;==1 one and only one return for yes

;n always no and never yes

;c (comcast-1 outputs) == (command2 outputs) for yes

;d used to check dmz… make sure that the same amount of “edit” and “intrazone deny” exist in the output

;a the output should contain “ 2 “ and “ 6 “ this is used only in 4.4.1

;p make sure the output does not have the text ‘set category’ only used for 4.4.3

cis_audit_script:

```
config system alias




edit cis13doc
```

set command "

#1.1 ;2

#1.2 ;d

#1.3 ;0

#3.1 ;y

#3.2 ;0

#3.3 ;g

```
#3.4 ;g
"

end

config system alias

edit cis2_4_4doc

set command "

#2.4.4 ;1

#2.4.5 ;0

#2.4.6 ;>2
```

```
#2.4.7 ;0

#2.5.1 ;g

#2.5.2 ;g

#2.5.3 ;g
"

end

config system alias
```

```
edit cis56doc

set command "
#5.1.1 ;[lvl 1- check for “status enable”]

#5.2.1.1 ;1

#6.1.1 ;0

#6.1.2 ;3

"

end
```

```
config system alias

edit cis78doc

set command "
#7.1 ;2

#8.1.1 ;0

#8.2.1 ;2

#8.3.1 ;0
```

```
"

end

config system alias

edit cis2_1doc

set command "
#2.1.1 ;1

#2.1.2 ;1
```

```
#2.1.3 ;1

#2.1.4 ;m

#2.1.5 ;0

"

end

config system alias

edit cis2_2doc

set command "

#2.1.6 ;m
```

```
#2.1.7 ;2

#2.1.8 ;1

#2.1.9 ;1

#2.1.10 ;1

"

end


config system alias

edit cis2_2-_4_3doc
```

set command "

#2.2.1 ;1

#2.2.2 ;1

#2.3.1 ;1

#2.3.2 ;==1

#2.4.1 ;g

#2.4.2 ;1

```
#2.4.3 ;n

"

end

config system alias

edit cis4_1doc

set command "

#4.1.1 ;g

#4.1.2 ;c
```

#4.2.1 ;2

#4.2.2 ;c

#4.2.3 ;1

#4.2.4 ;1

#4.2.5 ;1

#4.3.1 ;c

"

```
end

config system alias

edit cis4_2doc

set command "

#4.3.2 ;1

#4.3.3 ;c
```

```
#4.4.1 ;g



#4.4.2 ;g



#4.4.3 ;g



#4.4.4 ;c

"

end


config system alias
edit cis1_1
set command "show system alias cis13doc | grep -i #1.1

show system dns | grep -i 'set primary'

show system dns | grep -i 'set secondary' "
```

```
end
```

```
config system alias
edit cis1_2
set command "show system alias cis13doc | grep -i #1.2
show full system zone | grep edit
show full system zone | grep 'set intrazone deny'
show system alias cis13doc | grep -i #1.3
show system interface | grep -i 'set allowaccess ping http' "
end
```

```
config system alias
edit cis3
set command "
show system alias cis13doc | grep -i #3.1

show system alias cis13doc | grep -i #3.2

show firewall policy | grep -i 'service \"all\"'

show system alias cis13doc | grep -i #3.3

show system alias cis13doc | grep -i #3.4"
```

```
end

config system alias
edit cis4_1
set command "show system alias cis4_1doc | grep -i #4.1.1

show system alias cis4_1doc | grep -i #4.1.2

show firewall policy | grep -i 'set dstintf \"comcastint'
show firewall policy | grep -i 'set dstintf \"internets\"'

show firewall policy | grep -i ips-sensor"

end


config system alias
edit cis4_1_1
set command "

show system alias cis4_1doc | grep -i #4.2.1

show system autoupdate schedule "
end
```

```
config system alias
edit cis2_1

set command "show system alias cis2_1doc | grep -i #2.1.1

get system global | grep -i 'pre-login-banner    : enable'

show system alias cis2_1doc | grep -i #2.1.2

get system global | grep -i 'post-login-banner   : enable'  "
end


config system alias
edit cis2_2
set command "show system alias cis2_1doc | grep -i #2.1.3

get system global | grep -i 'eastern time'

show system alias cis2_1doc | grep -i #2.1.4

diag sys ntp status | grep -i 'reference time is' "
end
```

```
config system alias
edit cis2_3

set command "show system alias cis2_1doc | grep -i #2.1.5

get system global | grep -i 'hostname          : fg'

show system alias cis2_2doc | grep -i #2.1.6

get system status | grep -i version:"

end
```

```
config system alias
edit cis2_4
set command "show system alias cis2_2doc | grep -i #2.1.7

get system auto-install | grep -i 'auto-install-config : disable'
get system auto-install | grep -i 'auto-install-image  : disable' "

end



config system alias

edit cis2_5


set command "show system alias cis2_2doc | grep -i #2.1.8


get system global | grep -i 'ssl-static-key-ciphers: disable'


show system alias cis2_2doc | grep -i #2.1.9"

end



config system alias

edit cis2_6

set command "get system global | grep -i 'strong-crypto      : enable'


show system alias cis2_2doc | grep -i #2.1.10


show system global | grep -i 'admin-https-ssl-versions tlsv1-3' "

end
```

```
config system alias
edit cis2_7

set command "show system alias cis2_2-_4_3doc | grep -i #2.2.1

get system password-policy | grep -i 'status            : enable'

show system alias cis2_2-_4_3doc | grep -i #2.2.2

get system global | grep -i 'admin-lockout-threshold: 3' "
end


config system alias
edit cis2_8
set command "get system global | grep -i 'admin-lockout-duration: 60'

show system alias cis2_2-_4_3doc | grep -i #2.3.1

show system snmp sysinfo | grep -i 'set status enable'"
end


config system alias
```

```
edit cis2_9

set command "show system alias cis2_2-_4_3doc | grep -i #2.3.2

show system snmp user | grep -i 'notify-hosts'

show system snmp user | grep -i 'notify-hosts 0.0.0.0'

show system alias cis2_2-_4_3doc | grep -i #2.4.1  "
end
```

```
config system alias
edit cis2_10
set command "
show system alias cis2_2-_4_3doc | grep -i #2.4.2

config system admin

edit admin

show | grep -i trusthost
```

```
end


show system alias cis2_2-_4_3doc | grep -i #2.4.3
"
end




config system alias
edit cis2_11


set command "show system alias cis2_4_4doc | grep -i #2.4.4


get system global | grep -i 'admintimeout       : 5'


show system alias cis2_4_4doc | grep -i #2.4.5


show system interface | grep -i ' http '


show system interface | grep -i ' telnet '
"
end


config system alias
edit cis2_12
```

```
set command "show system alias cis2_4_4doc | grep -i #2.4.6

show firewall local-in-policy

show system alias cis2_4_4doc | grep -i #2.4.7

show system global | grep -i 'set admin-port 80' "
end


config system alias
edit cis2_13

set command "show system alias cis2_4_4doc | grep -i #2.5.1

show system alias cis2_4_4doc | grep -i #2.5.2

show system alias cis2_4_4doc | grep -i #2.5.3"

end




config system alias
```

```
edit cis4_2
set command "show system alias cis4_1doc | grep -i #4.2.2

show firewall policy | grep -i 'set dstintf \"comcastint'
show firewall policy | grep -i 'set dstintf \"internets\"'

show firewall policy | grep -i av-profile"
end


config system alias
edit cis4_2_1
set command "

show system alias cis4_1doc | grep -i #4.2.3

show antivirus profile | grep -i 'set outbreak-prevention block'"
end



config system alias
edit cis4_3
set command "show system alias cis4_1doc | grep -i #4.2.4

show antivirus settings | grep -i 'set machine-learning-detection enable'
show system alias cis4_1doc | grep -i #4.2.5

show antivirus settings | grep -i 'set grayware enable' "
end


config system alias
edit cis4_3_2
set command "show system alias cis4_1doc | grep -i #4.3.1
show dnsfilter profile | grep -i 'set block-botnet enable'
show firewall policy | grep -i 'set dstintf \"comcastinternet\"'"
end


config system alias
edit cis4_4
set command "
show firewall policy | grep -i 'set dstintf \"internets\"'
show firewall policy | grep -i 'set dnsfilter-profile \"default\"'
show system alias cis4_2doc | grep -i #4.3.2
show dnsfilter profile | grep -i 'set log-all-domain enable' "
end
```

```
config system alias
edit cis4_5
set command "show system alias cis4_2doc | grep -i #4.3.3

show firewall policy | grep -i 'set dstintf \"comcastinternet\"'
show firewall policy | grep -i 'set dstintf \"internets\"'

show firewall policy | grep -i 'set dnsfilter-profile \"default\"'
"
end

config system alias
edit cis4_5_1
set command "
show system alias cis4_2doc | grep -i #4.4.1

show system alias cis4_2doc | grep -i #4.4.2"
end

config system alias
edit cis4_5_2
set command "
show system alias cis4_2doc | grep -i #4.4.3"
end
```

```
config system alias

edit cis4_6

set command "

show system alias cis4_2doc | grep -i #4.4.4


show firewall policy | grep -i 'set dstintf \"comcast'

show firewall policy | grep -i 'set dstintf \"internets\"'


show firewall policy | grep -i 'set application-list \"default\"'"


end
```

```
config system alias


edit cis5_1


set command "show system alias cis56doc | grep -i #5.1.1


show system automation-stitch | grep -i 'edit \"compromised host quarantine\"' "
```

```
end


config system alias
edit cis5/6
set command "show system alias cis56doc | grep -i #5.2.1.1
show system csf | grep -i 'set status enable'
show system alias cis56doc | grep -i #6.1.1
show vpn certificate local | grep -i default "
end


config system alias
edit cis6
set command "show system alias cis56doc | grep -i #6.1.2
get vpn ssl settings | grep -i 'ssl-max-proto-ver   : tls1-3'
get vpn ssl settings | grep -i 'ssl-min-proto-ver   : tls1-2'


get vpn ssl settings | grep -i 'algorithm          : high' "

end
```

```
config system alias


edit cis7/8


set command "show system alias cis78doc | grep -i #7.1


get user setting | grep -i 'auth-lockout-threshold: 5'


get user setting | grep -i 'auth-lockout-duration: 300'


show system alias cis78doc | grep -i #8.1.1


get log eventfilter | grep -i disable "

end
```

```
config system alias

edit cis8_1

set command "show system alias cis78doc | grep -i #8.2.1

get log fortianalyzer setting | grep -i enc-algorithm

get log fortianalyzer setting | grep -i high"
end



config system alias

edit cis8_2

set command "show system alias cis78doc | grep -i #8.3.1

get log fortianalyzer setting | grep -i 'status           : disable'"

end
```

```
config system alias
edit audit1
set command "
alias cis1_1
alias cis1_2
alias cis2_1
alias cis2_2
alias cis2_3
alias cis2_4
alias cis2_5
alias cis2_6
alias cis2_7
alias cis2_8
alias cis2_9
alias cis2_10
alias cis2_11
alias cis2_12
alias cis2_13
alias cis3"
end
```

```
config system alias

edit audit2

set command "

alias cis4_1

alias cis4_1_1

alias cis4_2

alias cis4_2_1

alias cis4_3

alias cis4_3_2

alias cis4_4

alias cis4_5

alias cis4_5_1

alias cis4_5_2

alias cis4_6

alias cis5_1

alias cis5/6

alias cis6

alias cis7/8

alias cis8_1

alias cis8_2"

end


config system alias

edit cis_audit
```

```
set command "alias audit1
alias audit2"
end
```

Example Return:

```
#1.1 ;2
  set primary 96.45.45.45
  set secondary 96.45.46.46
#1.2 ;d
  edit "Internets"
  edit "Trusted VLANs"
  edit "Untrusted VLANS"
    set intrazone deny
    set intrazone deny
    set intrazone deny
#1.3 ;0
    set allowaccess ping https fgfm
    set allowaccess ping https fgfm fabric
    set allowaccess ping https ssh fgfm fabric
    set allowaccess ping https ssh http fabric
    set allowaccess ping https
    set command "#2.1.1 ;1
#2.1.2 ;1
#2.1.3 ;1
timezone          : (GMT-5:00) Eastern Time (US & Canada)
#2.1.4 ;m
    reference time is ea393af8.74f72f6e -- UTC Wed Jul 10 16:53:12 2024
    reference time is ea393ae8.26f20033 -- UTC Wed Jul 10 16:52:56 2024
    reference time is ea393ae8.26f20033 -- UTC Wed Jul 10 16:52:56 2024
    reference time is ea393af8.74f72f6e -- UTC Wed Jul 10 16:53:12 2024
#2.1.5 ;0
#2.1.6 ;m
Version: FortiGate-60F v7.2.6,build1575,230926 (GA.F)
BIOS version: 05000030
#2.1.7 ;2
#2.1.8 ;1
#2.1.9 ;1
strong-crypto      : enable
#2.1.10 ;1
#2.2.1 ;1
#2.2.2 ;1
admin-lockout-threshold: 3
admin-lockout-duration: 60
#2.3.1 ;1
#2.3.2 ;==1
#2.4.1 ;g
#2.4.2 ;1
#2.4.3 ;n
#2.4.4 ;1
admintimeout       : 5
#2.4.5 ;0
    set allowaccess ping https ssh http fabric
#2.4.6 ;>2
config firewall local-in-policy
```

```
end
#2.4.7 ;0
#2.5.1 ;g
#2.5.2 ;g
#2.5.3 ;g
#3.1 ;y
#3.2 ;0
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
#3.3 ;g
#3.4 ;g"
#4.1.1 ;g
#4.1.2 ;c
    set dstintf "Internets"
    set dstintf "Internets"
    set dstintf "Internets"
    set ips-sensor "default"
#4.2.1 ;2
```

```
config system autoupdate schedule
end
#4.2.2 ;c
    set dstintf "Internets"
    set dstintf "Internets"
    set dstintf "Internets"
    set av-profile "default"
#4.2.3 ;1
#4.2.4 ;1
  set machine-learning-detection enable
#4.2.5 ;1
  set grayware enable
#4.3.1 ;c
    set block-botnet enable
    set dstintf "Internets"
    set dstintf "Internets"
    set dstintf "Internets"
    set dnsfilter-profile "default"
#4.3.2 ;1
#4.3.3 ;c
    set dstintf "Internets"
    set dstintf "Internets"
    set dstintf "Internets"
    set dnsfilter-profile "default"
#4.4.1 ;g
#4.4.2 ;g
```

```
#4.4.3 ;g

#4.4.4 ;c

    set dstintf "Internets"

    set dstintf "Internets"

    set dstintf "Internets"

    set application-list "default"

    set command "#5.1.1 ;[lvl 1- check for “status enable”]

#5.2.1.1 ;1

#6.1.1 ;0

    set comments "This is the default CA certificate the SSL Inspection will use when generating new server certificates."

    set comments "This is the default CA certificate the SSL Inspection will use when generating new server certificates."

    set comments "This is the default CA certificate the SSL Inspection will use when generating new server certificates."

#6.1.2 ;3

ssl-max-proto-ver   : tls1-3

ssl-min-proto-ver   : tls1-2

algorithm           : high

#7.1 ;2

#8.1.1 ;0

#8.2.1 ;2

#8.3.1 ;0

status              : disable
```

Python output interpreter:

```
import csv

import tkinter as tk

from tkinter import filedialog, scrolledtext, simpledialog

import os

from tkinter import ttk


def parse_output_to_csv(output, csv_filename):

    # Split the output by lines

    lines = output.strip().split('\n')


    # Process each part and extract text before the next "#"

    extracted_texts = []

    current_part = []

    current_id = None


    for line in lines:

        if '#' in line:

            if current_part:

                # Extract first line and append to extracted_texts

                first_line = current_part[0].strip() if current_part else ''

                line_count = len(current_part) - 1


                # Determine if first line count is a letter

                if first_line.isdigit():
```

```
    if int(first_line) == 0:
        is_equal = 0
    else:
        is_equal = 1 if int(first_line) <= line_count else 0
elif first_line == "c":
    internets_count = sum(1 for l in current_part if 'Internets' in l or 'omcast' in l)
    lines_without_internets = line_count - internets_count
    is_equal = 1 if internets_count - lines_without_internets == 1 else 0
elif first_line == "y":
    is_equal = 1
elif first_line == "n":
    is_equal = 0
elif first_line == ">2":
    is_equal = 1 if line_count > 2 else 0
elif first_line == "==1":
    is_equal = 1 if line_count == 1 else 0
elif first_line == "d":
    intrazone_count = sum(1 for l in current_part if 'intrazone' in l)
    lines_without_intrazone = line_count - intrazone_count
    is_equal = 1 if intrazone_count - lines_without_intrazone == 0 else 0
elif first_line == "i":
    enabled_count = sum(1 for l in current_part if 'scan-botnet-connections' in l)
    lines_without_enabled = line_count - enabled_count
    is_equal = 1 if enabled_count - lines_without_enabled == 0 else 0
elif first_line == "a":
    # Ensure the output contains " 2 " and " 6 "
```

```
                    contains_2_and_6 = all(any(f" {num} " in l for l in current_part) for num in [" 2 ", " 6
"])
                    is_equal = 1 if contains_2_and_6 else 0
                elif first_line == "p":
                    # Ensure the output does not contain 'set category'
                    does_not_contain_set_category = all('set category' not in l for l in current_part)
                    is_equal = 1 if does_not_contain_set_category else 0
                elif first_line == "m":
                    is_equal = 'Manual; Requires Review'
                else:
                    is_equal = '?'

                extracted_texts.append((current_id, '\n'.join(current_part).strip(), first_line,
line_count, is_equal))
                current_part = []

            # Extract identifier and text
            if ';' in line:
                parts = line.split(';', 1)
                identifier = parts[0].strip().replace('#', '')
                current_id = identifier
                line = parts[1]

            current_part.append(line.strip())
        else:
            current_part.append(line.strip())
```

```
    if current_part:
        first_line = current_part[0].strip() if current_part else ''
        line_count = len(current_part) - 1

        if first_line.isdigit():
            is_equal = 1 if first_line == str(line_count) else 0
        else:
            is_equal = '?'

        extracted_texts.append((current_id, '\n'.join(current_part).strip(), first_line, line_count, is_equal))

    # Write to CSV file
    with open(csv_filename, 'w', newline='') as csvfile:
        csvwriter = csv.writer(csvfile)
        csvwriter.writerow(['Identifier', 'Text', 'First Line', 'Line Count', 'Standard is Met'])

        for identifier, text, first_line, line_count, is_equal in extracted_texts:
            csvwriter.writerow([identifier, text, first_line, line_count, is_equal])

def upload_file():
    file_path = filedialog.askopenfilename(filetypes=[("Text files", "*.txt")])
    if file_path:
        with open(file_path, 'r') as file:
            content = file.read()
            text_box.delete('1.0', tk.END)
            text_box.insert(tk.END, content)


def prompt_for_filename():
    # Prompt the user for a CSV file name
    csv_filename = simpledialog.askstring("Save CSV", "Enter the CSV file name (with .csv extension):")
    return csv_filename


def generate_csv():
    csv_filename = prompt_for_filename()
    if csv_filename:
        output = text_box.get('1.0', tk.END)
        parse_output_to_csv(output, csv_filename)
        status_label.config(text=f"CSV generated successfully as {csv_filename}.")
        display_csv_button.config(state=tk.NORMAL)
    else:
        status_label.config(text="CSV generation canceled.")


def display_csv_file():
    csv_filename = 'cis_audit.csv'
    if os.path.exists(csv_filename):
        with open(csv_filename, 'r', newline='') as csvfile:
            csvreader = csv.reader(csvfile)
            data = list(csvreader)
        # Create a new window to display CSV content
        display_window = tk.Toplevel(root)
        display_window.title("CSV Content")

        # Create a Treeview widget with scrollbars
        tree = ttk.Treeview(display_window, columns=data[0], show="headings")
        tree.pack(padx=10, pady=10, fill=tk.BOTH, expand=True)

        # Add scrollbars
        vsb = ttk.Scrollbar(display_window, orient="vertical", command=tree.yview)
        vsb.pack(side=tk.RIGHT, fill=tk.Y)
        tree.configure(yscrollcommand=vsb.set)

        # Set column headings (first row of CSV)
        for col in data[0]:
            tree.heading(col, text=col)

        # Insert data rows into the Treeview
        for row in data[1:]:
            tree.insert("", tk.END, values=row)

    else:
        status_label.config(text="CSV file does not exist.")

# Create the main window
root = tk.Tk()
```

```
root.title("Output to CSV Converter")

# Create a text box for input/output
text_box = scrolledtext.ScrolledText(root, width=80, height=20)
text_box.pack(padx=10, pady=10)

# Button to upload file
upload_button = tk.Button(root, text="Upload File", command=upload_file)
upload_button.pack(pady=5)

# Button to generate CSV
generate_button = tk.Button(root, text="Generate CSV", command=generate_csv)
generate_button.pack(pady=5)

# Button to display CSV
display_csv_button = tk.Button(root, text="Display CSV", command=display_csv_file, state=tk.DISABLED)
display_csv_button.pack(pady=5)

# Status label
status_label = tk.Label(root, text="", fg="green")
status_label.pack(pady=5)

# Start the GUI main loop
root.mainloop()
```

```
# Example output


output = '''#1.1 ;2
  set primary 96.45.45.45
  set secondary 96.45.46.46
#1.2 ;d
  edit "Internets"
  edit "Trusted VLANs"
  edit "Untrusted VLANS"
    set intrazone deny
    set intrazone deny
    set intrazone deny
#1.3 ;0
    set allowaccess ping https fgfm
    set allowaccess ping https fgfm fabric
    set allowaccess ping https ssh fgfm fabric
    set allowaccess ping https ssh http fabric
    set allowaccess ping https
    set command "#2.1.1 ;1
#2.1.2 ;1
#2.1.3 ;1
timezone          : (GMT-5:00) Eastern Time (US & Canada)
#2.1.4 ;m
```

```
    reference time is ea393af8.74f72f6e -- UTC Wed Jul 10 16:53:12 2024
    reference time is ea393ae8.26f20033 -- UTC Wed Jul 10 16:52:56 2024
    reference time is ea393ae8.26f20033 -- UTC Wed Jul 10 16:52:56 2024
    reference time is ea393af8.74f72f6e -- UTC Wed Jul 10 16:53:12 2024
#2.1.5 ;0
#2.1.6 ;m
Version: FortiGate-60F v7.2.6,build1575,230926 (GA.F)
BIOS version: 05000030
#2.1.7 ;2
#2.1.8 ;1
#2.1.9 ;1
strong-crypto       : enable
#2.1.10 ;1
#2.2.1 ;1
#2.2.2 ;1
admin-lockout-threshold: 3
admin-lockout-duration: 60
#2.3.1 ;1
#2.3.2 ;==1
#2.4.1 ;g
#2.4.2 ;1
#2.4.3 ;n
#2.4.4 ;1
admintimeout        : 5
#2.4.5 ;0
    set allowaccess ping https ssh http fabric
```

```
#2.4.6 ;>2
config firewall local-in-policy
end
#2.4.7 ;0
#2.5.1 ;g
#2.5.2 ;g
#2.5.3 ;g
#3.1 ;y
#3.2 ;0
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
    set service "ALL"
#3.3 ;g
#3.4 ;g"
#4.1.1 ;g
#4.1.2 ;c
    set dstintf "Internets"
    set dstintf "Internets"
    set dstintf "Internets"
    set ips-sensor "default"
#4.2.1 ;2
config system autoupdate schedule
end
#4.2.2 ;c
    set dstintf "Internets"
    set dstintf "Internets"
    set dstintf "Internets"
    set av-profile "default"
#4.2.3 ;1
#4.2.4 ;1
  set machine-learning-detection enable
#4.2.5 ;1
  set grayware enable
#4.3.1 ;c
    set block-botnet enable
    set dstintf "Internets"
    set dstintf "Internets"
    set dstintf "Internets"
    set dnsfilter-profile "default"
#4.3.2 ;1
#4.3.3 ;c
    set dstintf "Internets"
    set dstintf "Internets"
    set dstintf "Internets"
    set dnsfilter-profile "default"
```

```
#4.4.1 ;g

#4.4.2 ;g

#4.4.3 ;g

#4.4.4 ;c

    set dstintf "Internets"

    set dstintf "Internets"

    set dstintf "Internets"

    set application-list "default"

    set command "#5.1.1 ;[lvl 1- check for “status enable”]

#5.2.1.1 ;1

#6.1.1 ;0

    set comments "This is the default CA certificate the SSL Inspection will use when generating new server certificates."

    set comments "This is the default CA certificate the SSL Inspection will use when generating new server certificates."

    set comments "This is the default CA certificate the SSL Inspection will use when generating new server certificates."

#6.1.2 ;3

ssl-max-proto-ver   : tls1-3

ssl-min-proto-ver   : tls1-2

algorithm           : high

#7.1 ;2

#8.1.1 ;0

#8.2.1 ;2

#8.3.1 ;0

status              : disable '"
```

```
# Call the function

parse_output_to_csv(output, 'cis_audit.csv')
```

CSV2Script tool:

```
import csv
import tkinter as tk
from tkinter import filedialog, scrolledtext

# Global variables for file paths
documentation_file = ""
commands_csv_file_path = ""

# Function to handle file upload for documentation CSV
def upload_documentation_file():
    global documentation_file
    documentation_file = filedialog.askopenfilename()
    output_text.insert(tk.END, f"Documentation File Path: {documentation_file}\n")

# Function to handle file upload for commands CSV
def upload_commands_file():
    global commands_csv_file_path
    commands_csv_file_path = filedialog.askopenfilename()
    output_text.insert(tk.END, f"Commands CSV File Path: {commands_csv_file_path}\n")

# Function to process documentation CSV and generate output
def process_documentation_csv(output_file):
    global documentation_file
    if documentation_file:
        with open(documentation_file, mode='r') as file:
            reader = csv.reader(file)
            data = list(reader)

        with open(output_file, mode='a') as output:
            for row in data:
                if len(row) >= 2:
                    standard = row[0]
                    documentation = row[1]
                    output.write("config system alias\n")
                    output.write(f"edit {standard}doc\n")
                    output.write('set command "\n')
                    output.write(f"#{standard} ;{documentation}\n")
                    output.write('"\n')
                    output.write("end\n\n")


                    output_text.insert(tk.END, f"Processed standard: {standard}\n")

# Placeholder function to collect standards from the CSV
def collect_standards_from_csv(csv_file):
    standards = []
    with open(csv_file, newline='') as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            standards.append(row["Standard Number"])
    return standards
```

```
# Placeholder function to prepend alias to standards
def prepend_alias_to_standards(standards):
    return [f"alias {standard}" for standard in standards]

# Function to chunk a list by length
def chunk_list_by_length(lst, max_length):
    chunks = []
    current_chunk = []
    current_length = 0

    for item in lst:
        item_length = len(item)
        if current_length + item_length > max_length:
            chunks.append(current_chunk)
            current_chunk = [item]
            current_length = item_length
        else:
            current_chunk.append(item)
            current_length += item_length

    if current_chunk:
        chunks.append(current_chunk)

    return chunks
```

```
# Function to process commands CSV and generate output commands
def process_commands_csv(output_file):
    global commands_csv_file_path
    if commands_csv_file_path:
        def print_commands_from_csv(csv_file, output):
            with open(csv_file, newline='') as csvfile:
                reader = csv.DictReader(csvfile)
                for row in reader:
                    standard_number = row["Standard Number"]
                    commands = row["Commands"]

                    if commands:
                        output.write("config system alias\n")
                        output.write(f"edit {standard_number}\n")
                        output.write('set command "\n')
                        output.write(f"show system alias {standard_number}doc | grep -i
#{standard_number}\n")
                        output.write(commands.strip() + "\n")
                        output.write('"\n')
                        output.write("end\n\n")

                        output_text.insert(tk.END, f"Processed commands for standard:
{standard_number}\n")

        standards = collect_standards_from_csv(commands_csv_file_path)
        prefixed_standards = prepend_alias_to_standards(standards)
```

```
with open(output_file, mode='a') as output:
    print_commands_from_csv(commands_csv_file_path, output)


    max_length = 250
    chunks = chunk_list_by_length(prefixed_standards, max_length)


    num_chunks = 0
    for num_chunks, chunk in enumerate(chunks, start=1):
        output.write("config system alias\n")
        output.write(f"edit a{num_chunks}\n")
        output.write('set command "\n')
        output.write("\n".join(chunk) + "\n")
        output.write('"\n')
        output.write("end\n\n")


    output.write("config system alias\n")
    output.write("edit cis_audit\n")
    output.write('set command "\n')
    for i in range(1, num_chunks + 1):
        output.write(f"alias a{i}\n")
    output.write('"\n')
    output.write("end\n")


    output_text.insert(tk.END, "Finished processing commands CSV\n")
```

```
# GUI setup
root = tk.Tk()
root.title("Upload Files")

# Title label
title_label = tk.Label(root, text="Fortinet CIS Auditing Script Generator", font=("Helvetica", 16, "bold"))
title_label.pack(pady=10)

# Description label
description_label = tk.Label(root, text="Upload your documentation and commands CSV files to generate a Fortinet CIS auditing script.", font=("Helvetica", 12))
description_label.pack(pady=5)

# Buttons for file uploads
btn_documentation = tk.Button(root, text="Upload Documentation CSV", command=upload_documentation_file)
btn_documentation.pack(pady=10)

btn_commands = tk.Button(root, text="Upload Commands CSV", command=upload_commands_file)
btn_commands.pack(pady=10)

# Text widget for displaying output
output_text = scrolledtext.ScrolledText(root, wrap=tk.WORD, width=80, height=20)
output_text.pack(pady=10)
```

```
# Text widget for displaying generated script
generated_text_area = scrolledtext.ScrolledText(root, wrap=tk.WORD, width=80,
height=20)
generated_text_area.pack(pady=10)

# Function to process files
def process_files():
    output_file_name = filedialog.asksaveasfilename(defaultextension=".txt",
filetypes=[("Text files", "*.txt")])
    if output_file_name:
        process_documentation_csv(output_file_name)
        process_commands_csv(output_file_name)
        with open(output_file_name, 'r') as file:
            generated_text = file.read()
            generated_text_area.delete('1.0', tk.END)  # Clear previous content
            generated_text_area.insert(tk.END, generated_text)
        output_text.insert(tk.END, "Done!\n")

btn_process = tk.Button(root, text="Process Files", command=process_files)
btn_process.pack(pady=20)

root.mainloop()
```

Script2CSVCommands:

```
import re
import csv

# Define the input text
input_text = """

"""

# Split the input text into lines
lines = input_text.strip().split('\n')

# Initialize a list to store data rows for CSV
csv_data = []

# Variables to track the indices of lines with "grep -i #"
indices = []

# Iterate through the lines to find "grep -i #" and capture the indices
for i, line in enumerate(lines):
    if 'grep -i #' in line:
        indices.append(i)

# Iterate through the indices to extract and format data for CSV
for i in range(len(indices)):
    standard_number = None
```

```
commands = []

# Extract the standard number from "grep -i #"
match = re.search(r"grep -i #(\d+(?:\.\d+)*)", lines[indices[i]])
if match:
  standard_number = match.group(1)

# Determine the range of lines to capture commands
if i < len(indices) - 1:
  start_index = indices[i] + 1
  end_index = indices[i + 1]
else:
  start_index = indices[i] + 1
  end_index = len(lines)

# Accumulate the lines with commands until the end of the command block
for line in lines[start_index:end_index]:
  line = line.strip()
  commands.append(line)
  # Check if the line contains the closing quote to stop capturing
  if '"' in line:
    break

# Append the data row to csv_data
csv_data.append([standard_number, '\n'.join(commands)])
```

```
# Define the CSV file path
csv_file = 'output_commands.csv'

# Write data to CSV file
with open(csv_file, mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['Standard Number', 'Commands'])
    writer.writerows(csv_data)

print(f"CSV file '{csv_file}' has been created successfully.")

# Input and output file names
input_file = 'output_commands.csv'
output_file = 'output_commands_processed.csv'

# Function to process the CSV file
def process_csv(input_file, output_file):
    with open(input_file, mode='r', newline='') as infile, \
         open(output_file, mode='w', newline='') as outfile:

        reader = csv.reader(infile)
        writer = csv.writer(outfile)

        for row in reader:
            if len(row) > 1 and row[1].endswith('"'):
                row[1] = row[1][:-1]  # Remove the last character
            writer.writerow(row)

# Process the CSV file
process_csv(input_file, output_file)
print(f"CSV file '{output_file}' has been processed successfully.")


import re
import csv

# Open the input file
input_file = '[INSERT_FILE_PATH_HERE]’
output_file = 'documentation.csv'

with open(input_file, 'r') as file:
    lines = file.readlines()

# Regular expression to match lines with both '#' and ';'
pattern = r'#([^;\n]*)\s*;\s*(.*)'

# List to store extracted data
extracted_data = []

# Process each line
for line in lines:
    match = re.search(pattern, line)
    if match:
        hashtag_content = match.group(1).strip()
        after_semicolon = match.group(2).strip()
        extracted_data.append([hashtag_content, after_semicolon])

# Write extracted data to CSV file
with open(output_file, 'w', newline='') as csvfile:
    csv_writer = csv.writer(csvfile)
    csv_writer.writerow(['Hashtag Content', 'Text After Semicolon'])
    csv_writer.writerows(extracted_data)

print(f"Data has been saved to {output_file}")
```

Script2CSVDoc

```
import re
import csv

# Open the input file
input_file = '[INSERT_SCRIPT_FILE_PATH_HERE]'
output_file = 'documentation.csv'

with open(input_file, 'r') as file:
    lines = file.readlines()

# Regular expression to match lines with both '#' and ';'
pattern = r'#([^;\n]*)\s*;\s*(.*)'
```

```
# List to store extracted data
extracted_data = []

# Process each line
for line in lines:
    match = re.search(pattern, line)
    if match:
        hashtag_content = match.group(1).strip()
        after_semicolon = match.group(2).strip()
        extracted_data.append([hashtag_content, after_semicolon])

# Write extracted data to CSV file
with open(output_file, 'w', newline='') as csvfile:
    csv_writer = csv.writer(csvfile)
    csv_writer.writerow(['Hashtag_Content', 'Text_After_Semicolon'])
    csv_writer.writerows(extracted_data)

print(f"Data has been saved to {output_file}")
```

Generated CIS Script Example:

```
config system alias
edit 1.1doc
set command "
#1.1 ;2
"
end

config system alias
edit 1.2doc
set command "
#1.2 ;d
"
end

config system alias
edit 1.3doc
set command "
#1.3 ;0
"
end

config system alias
edit 3.1doc
set command "
#3.1 ;y
```

```
"
end

config system alias
edit 3.2doc
set command "
#3.2 ;0
"
end

config system alias
edit 3.3doc
set command "
#3.3 ;1
"
end

config system alias
edit 3.4doc
set command "
#3.4 ;0
"
end

config system alias
edit 2.4.4doc
```

```
set command "
#2.4.4 ;1
"
end

config system alias
edit 2.4.5doc
set command "
#2.4.5 ;0
"
end

config system alias
edit 2.4.6doc
set command "
#2.4.6 ;>2
"
end

config system alias
edit 2.4.7doc
set command "
#2.4.7 ;0
"
end
```

```
config system alias

edit 2.5.1doc

set command "

#2.5.1 ;1

"

end

config system alias

edit 2.5.2doc

set command "

#2.5.2 ;1

"

end

config system alias

edit 2.5.3doc

set command "

#2.5.3 ;3

"

end

config system alias

edit 5.1.1doc

set command "

#5.1.1 ;1

"
```

```
end

config system alias
edit 5.2.1.1doc
set command "
#5.2.1.1 ;1
"
end

config system alias
edit 6.1.1doc
set command "
#6.1.1 ;0
"
end

config system alias
edit 6.1.2doc
set command "
#6.1.2 ;3
"
end

config system alias
edit 7.1doc
set command "
```

```
#7.1 ;2
"
end

config system alias
edit 8.1.1doc
set command "
#8.1.1 ;0
"
end

config system alias
edit 8.2.1doc
set command "
#8.2.1 ;2
"
end

config system alias
edit 8.3.1doc
set command "
#8.3.1 ;0
"
end

config system alias
```

```
edit 2.1.1doc
set command "
#2.1.1 ;1
"
end

config system alias
edit 2.1.2doc
set command "
#2.1.2 ;1
"
end

config system alias
edit 2.1.3doc
set command "
#2.1.3 ;1
"
end

config system alias
edit 2.1.4doc
set command "
#2.1.4 ;m
"
end
```

```
config system alias
edit 2.1.5doc
set command "
#2.1.5 ;0
"
end

config system alias
edit 2.1.6doc
set command "
#2.1.6 ;m
"
end

config system alias
edit 2.1.7doc
set command "
#2.1.7 ;2
"
end

config system alias
edit 2.1.8doc
set command "
#2.1.8 ;1
```

```
"
end

config system alias
edit 2.1.9doc
set command "
#2.1.9 ;1
"
end

config system alias
edit 2.1.10doc
set command "
#2.1.10 ;1
"
end

config system alias
edit 2.2.1doc
set command "
#2.2.1 ;1
"
end

config system alias
edit 2.2.2doc
```

```
set command "
#2.2.2 ;1
"
end

config system alias
edit 2.3.1doc
set command "
#2.3.1 ;1
"
end

config system alias
edit 2.3.2doc
set command "
#2.3.2 ;==1
"
end

config system alias
edit 2.4.1doc
set command "
#2.4.1 ;g
"
end
```

```
config system alias
edit 2.4.2doc
set command "
#2.4.2 ;1
"
end

config system alias
edit 2.4.3doc
set command "
#2.4.3 ;n
"
end

config system alias
edit 4.1.1doc
set command "
#4.1.1 ;m
"
end

config system alias
edit 4.1.2doc
set command "
#4.1.2 ;c
"
```

```
end

config system alias
edit 4.2.1doc
set command "
#4.2.1 ;2
"
end

config system alias
edit 4.2.2doc
set command "
#4.2.2 ;c
"
end

config system alias
edit 4.2.3doc
set command "
#4.2.3 ;1
"
end

config system alias
edit 4.2.4doc
set command "
```

```
#4.2.4 ;1
"
end

config system alias
edit 4.2.5doc
set command "
#4.2.5 ;1
"
end

config system alias
edit 4.3.1doc
set command "
#4.3.1 ;c
"
end

config system alias
edit 4.3.2doc
set command "
#4.3.2 ;1
"
end

config system alias
```

```
edit 4.3.3doc
set command "
#4.3.3 ;c
"
end

config system alias
edit 4.4.1doc
set command "
#4.4.1 ;a
"
end

config system alias
edit 4.4.2doc
set command "
#4.4.2 ;1
"
end

config system alias
edit 4.4.3doc
set command "
#4.4.3 ;p
"
end
```

```
config system alias
edit 4.4.4doc
set command "
#4.4.4 ;c
"
end

config system alias
edit 1.1
set command "
show system alias 1.1doc | grep -i #1.1
show system dns | grep -i 'set primary'

show system dns | grep -i 'set secondary'
"
end

config system alias
edit 1.2
set command "
show system alias 1.2doc | grep -i #1.2
show full system zone | grep edit
show full system zone | grep 'set intrazone deny'
"
end
```

```
config system alias
edit 1.3
set command "
show system alias 1.3doc | grep -i #1.3
show system interface | grep -i 'set allowaccess ping http'
"
end

config system alias
edit 3.2
set command "
show system alias 3.2doc | grep -i #3.2
show firewall policy | grep -i 'service "all"'
"
end

config system alias
edit 3.3
set command "
show system alias 3.3doc | grep -i #3.3
show full firewall policy | grep -i tor
"
end

config system alias
```

```
edit 3.4
set command "
show system alias 3.4doc | grep -i #3.4
show firewall policy | grep 'set log traffic disable'
"
end

config system alias
edit 4.1.1
set command "
show system alias 4.1.1doc | grep -i #4.1.1
show full firewall policy | grep 'ips-sensor "'
"
end

config system alias
edit 4.1.2
set command "
show system alias 4.1.2doc | grep -i #4.1.2
show firewall policy | grep -i 'set dstintf "comcastint'
"
end

config system alias
edit 4.2.1
set command "
```

```
show system alias 4.2.1doc | grep -i #4.2.1
show system autoupdate schedule | grep 'set status enable'
show system autoupdate schedule | grep 'set frequency automatic'
"
end

config system alias
edit 2.1.1
set command "
show system alias 2.1.1doc | grep -i #2.1.1
get system global | grep -i 'pre-login-banner    : enable'
"
end

config system alias
edit 2.1.2
set command "
show system alias 2.1.2doc | grep -i #2.1.2
get system global | grep -i 'post-login-banner   : enable'
"
end

config system alias
edit 2.1.3
set command "
show system alias 2.1.3doc | grep -i #2.1.3
```

```
get system global | grep -i 'eastern time'
"
end

config system alias
edit 2.1.4
set command "
show system alias 2.1.4doc | grep -i #2.1.4
diag sys ntp status | grep -i 'reference time is'
"
end

config system alias
edit 2.1.5
set command "
show system alias 2.1.5doc | grep -i #2.1.5
get system global | grep -i 'hostname           : fg'
"
end

config system alias
edit 2.1.6
set command "
show system alias 2.1.6doc | grep -i #2.1.6
get system status | grep -i version:
"
```

```
end

config system alias
edit 2.1.7
set command "
show system alias 2.1.7doc | grep -i #2.1.7
get system auto-install | grep -i 'auto-install-config : disable'

get system auto-install | grep -i 'auto-install-image  : disable'
"
end

config system alias
edit 2.1.8
set command "
show system alias 2.1.8doc | grep -i #2.1.8
get system global | grep -i 'ssl-static-key-ciphers: disable'
"
end

config system alias
edit 2.1.10
set command "
show system alias 2.1.10doc | grep -i #2.1.10
show system global | grep -i 'admin-https-ssl-versions tlsv1-3'
"
```

```
end

config system alias
edit 2.2.1
set command "
show system alias 2.2.1doc | grep -i #2.2.1
get system password-policy | grep -i 'status            : enable'
"
end

config system alias
edit 2.2.2
set command "
show system alias 2.2.2doc | grep -i #2.2.2
get system global | grep -i 'admin-lockout-threshold: 3'
"
end

config system alias
edit 2.3.1
set command "
show system alias 2.3.1doc | grep -i #2.3.1
show system snmp sysinfo | grep -i 'set status enable'
"
end
```

```
config system alias
edit 2.3.2
set command "
show system alias 2.3.2doc | grep -i #2.3.2
show system snmp user | grep -i 'notify-hosts'

show system snmp user | grep -i 'notify-hosts 0.0.0.0'
"
end

config system alias
edit 2.4.2
set command "
show system alias 2.4.2doc | grep -i #2.4.2
show full system admin | grep -i trusthost
"
end

config system alias
edit 2.4.4
set command "
show system alias 2.4.4doc | grep -i #2.4.4
get system global | grep -i 'admintimeout       : 5'
"
end
```

```
config system alias
edit 2.4.5
set command "
show system alias 2.4.5doc | grep -i #2.4.5
show system interface | grep -i ' http '

show system interface | grep -i ' telnet '
"
end

config system alias
edit 2.4.6
set command "
show system alias 2.4.6doc | grep -i #2.4.6
show firewall local-in-policy
"
end

config system alias
edit 2.4.7
set command "
show system alias 2.4.7doc | grep -i #2.4.7
show system global | grep -i 'set admin-port 80'
"
end
```

```
config system alias
edit 2.5.1
set command "
show system alias 2.5.1doc | grep -i #2.5.1
show system ha | grep 'set mode a-'
"
end
```

```
config system alias
edit 2.5.2
set command "
show system alias 2.5.2doc | grep -i #2.5.2
show system ha | grep 'set monitor'
"
end
```

```
config system alias
edit 2.5.3
set command "
show system alias 2.5.3doc | grep -i #2.5.3
show system ha | grep 'set ha-mgmt-status enable'
show system ha | grep 'set interface'
show system ha | grep 'set gateway'
"
end
```

```
config system alias
edit 4.2.2
set command "
show system alias 4.2.2doc | grep -i #4.2.2
show firewall policy | grep -i 'set dstintf "comcastint'
"
end

config system alias
edit 4.2.3
set command "
show system alias 4.2.3doc | grep -i #4.2.3
show antivirus profile | grep -i 'set outbreak-prevention block'
"
end

config system alias
edit 4.2.4
set command "
show system alias 4.2.4doc | grep -i #4.2.4
show antivirus settings | grep -i 'set machine-learning-detection enable'
"
end

config system alias
edit 4.2.5
```

```
set command "

show system alias 4.2.5doc | grep -i #4.2.5

show antivirus settings | grep -i 'set grayware enable'

"

end


config system alias

edit 4.3.1

set command "

show system alias 4.3.1doc | grep -i #4.3.1

show dnsfilter profile | grep -i 'set block-botnet enable'

show firewall policy | grep -i 'set dstintf "comcastinternet"'

"

end


config system alias

edit 4.3.2

set command "

show system alias 4.3.2doc | grep -i #4.3.2

show dnsfilter profile | grep -i 'set log-all-domain enable'

"

end


config system alias

edit 4.3.3

set command "
```

```
show system alias 4.3.3doc | grep -i #4.3.3

show firewall policy | grep -i 'set dstintf "comcastinternet"'

"

end


config system alias

edit 4.4.1

set command "

show system alias 4.4.1doc | grep -i #4.4.1

show application list default | grep -A1 'edit 1'

"

end


config system alias

edit 4.4.2

set command "

show system alias 4.4.2doc | grep -i #4.4.2

show application list default | grep 'set enforce-default-app-port enable'

"

end


config system alias

edit 4.4.3

set command "

show system alias 4.4.3doc | grep -i #4.4.3

show application list default | grep -A3 'edit 2'
```

```
"
end

config system alias
edit 4.4.4
set command "
show system alias 4.4.4doc | grep -i #4.4.4
show firewall policy | grep -i 'set dstintf "comcast'
"
end

config system alias
edit 5.1.1
set command "
show system alias 5.1.1doc | grep -i #5.1.1
show system automation-stitch "Compromised Host Quarantine" | grep -i 'set status enabled'
"
end

config system alias
edit 5.2.1.1
set command "
show system alias 5.2.1.1doc | grep -i #5.2.1.1
show system csf | grep -i 'set status enable'
"
```

```
end

config system alias
edit 6.1.1
set command "
show system alias 6.1.1doc | grep -i #6.1.1
show vpn certificate local | grep -i default
"
end

config system alias
edit 6.1.2
set command "
show system alias 6.1.2doc | grep -i #6.1.2
get vpn ssl settings | grep -i 'ssl-max-proto-ver   : tls1-3'

get vpn ssl settings | grep -i 'ssl-min-proto-ver   : tls1-2'

get vpn ssl settings | grep -i 'algorithm           : high'
"
end

config system alias
edit 7.1
set command "
show system alias 7.1doc | grep -i #7.1
```

```
get user setting | grep -i 'auth-lockout-threshold: 5'

get user setting | grep -i 'auth-lockout-duration: 300'
"
end

config system alias
edit 8.1.1
set command "
show system alias 8.1.1doc | grep -i #8.1.1
get log eventfilter | grep -i disable
"
end

config system alias
edit 8.2.1
set command "
show system alias 8.2.1doc | grep -i #8.2.1
get log fortianalyzer setting | grep -i enc-algorithm

get log fortianalyzer setting | grep -i high
"
end

config system alias
edit 8.3.1
```

```
set command "
show system alias 8.3.1doc | grep -i #8.3.1
get log fortianalyzer setting | grep -i 'status          : disable'
"
end

config system alias
edit a1
set command "
alias 1.1
alias 1.2
alias 1.3
alias 3.1
alias 3.2
alias 3.3
alias 3.4
alias 4.1.1
alias 4.1.2
alias 4.2.1
alias 2.1.1
alias 2.1.2
alias 2.1.3
alias 2.1.4
alias 2.1.5
alias 2.1.6
alias 2.1.7
alias 2.1.8
alias 2.1.9
alias 2.1.10
alias 2.2.1
alias 2.2.2
alias 2.3.1
"
end

config system alias
edit a2
set command "
alias 2.3.2
alias 2.4.1
alias 2.4.2
alias 2.4.3
alias 2.4.4
alias 2.4.5
alias 2.4.6
alias 2.4.7
alias 2.5.1
alias 2.5.2
alias 2.5.3
alias 4.2.2
alias 4.2.3
alias 4.2.4
alias 4.2.5
alias 4.3.1
alias 4.3.2
alias 4.3.3
alias 4.4.1
alias 4.4.2
alias 4.4.3
alias 4.4.4
"
end

config system alias
edit a3
set command "
alias 5.1.1
alias 5.2.1.1
alias 6.1.1
alias 6.1.2
alias 7.1
alias 8.1.1
alias 8.2.1
alias 8.3.1
"
end

config system alias
```

```
edit cis_audit

set command "

alias a1

alias a2

alias a3

"

end
```