Internal Compliance Use Cases

Overview

The vast majority of IT and network teams have best practices or gold standards for network device configurations in order to maintain a stable and secure infrastructure when dealing with multiple devices spread across multiple locations. Since these are developed in-house, the term Internal Compliance (or Internal Standardization) is typically used. These standards typically focus on best practices such as:

- Logging server pointing to a specific IP address
- Access control to the device
- DHCP helper configuration pointing to a specific DHCP server
- Telnet disabled
- Standard VLAN numbers
- Standard routing protocol

Current Situation with Manual Processes

For many organizations, most teams attempt to maintain compliance via manual documentation and periodic spot checks. A major pain point is senior network staff is required to manually go device by device, standard by standard. And since your staff is likely overtaxed already, Internal Compliance often becomes an afterthought until something breaks or when a problem occurs. With each configuration change, the risk of non-compliance grows and drift eventually violates policy.

The challenge lies in the fact that inconsistent/non-standard networks tend to break more often. In fact, as Gartner notes, “80% of unplanned outages impacting mission-critical services will be caused by people and process issues, and more than 50% of those outages will be caused by change/configuration/release integration and hand-off issues.” (*Configuration Management for Virtual and Cloud Infrastructure* – Gartner)

Network Automation with Infoblox

Infoblox Network Automation helps ensure that your desired state actually matches your as-is state and does so continuously for each and every change made across the devices. Instead of manually digging through massive files one by one, the platform automates the process of comparing the current configuration to the internal standard and automatically notifies when violations occur. This ensures configuration parameters stay within your customizable, defined standards.

Best of all, the process is continuous. When a configuration change is made, the Network Automation platform detects the change and compares it to the standard. And if a violation is flagged, users can leverage the same platform and restore it to the proper configuration within 15 minutes.

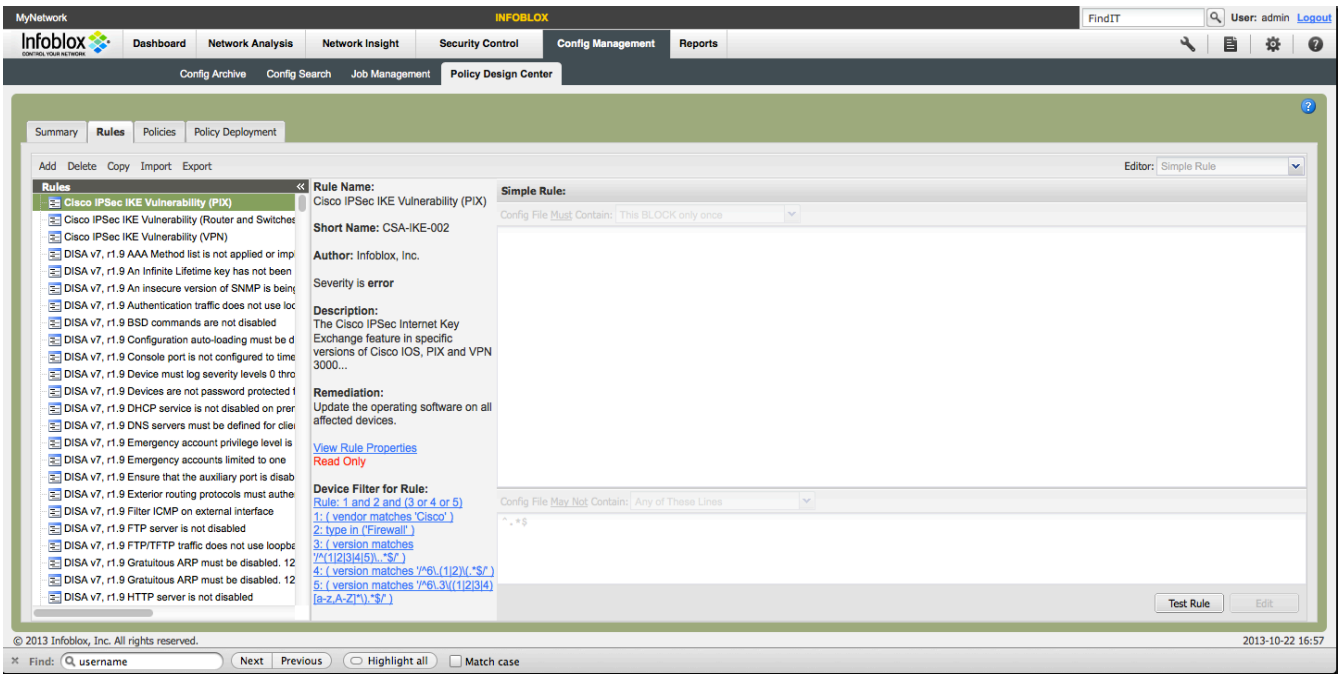
Use Case

This use case document provides a sample of how to create and monitor for Internal Compliance. By following the instructions, you can create your own rule and policy for your specific environment. More importantly, you can easily broaden this to include other parameters by using the templates and customization options available within Infoblox Network Automation.

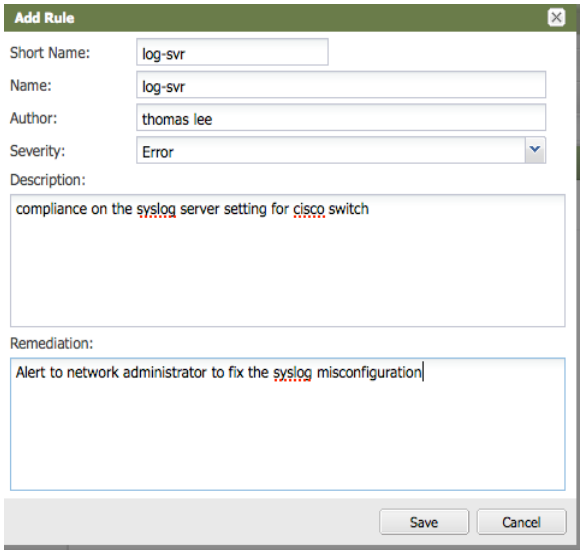
Instructions

Creating a simple rule and policy to ensure the logging server IP address is in compliance

1. Go to Config Management -> Policy Design Center -> Rules.



2. Click on the Add button to add a rule. A rule consists of one or more configuration file command matches and/or device attribute checks.

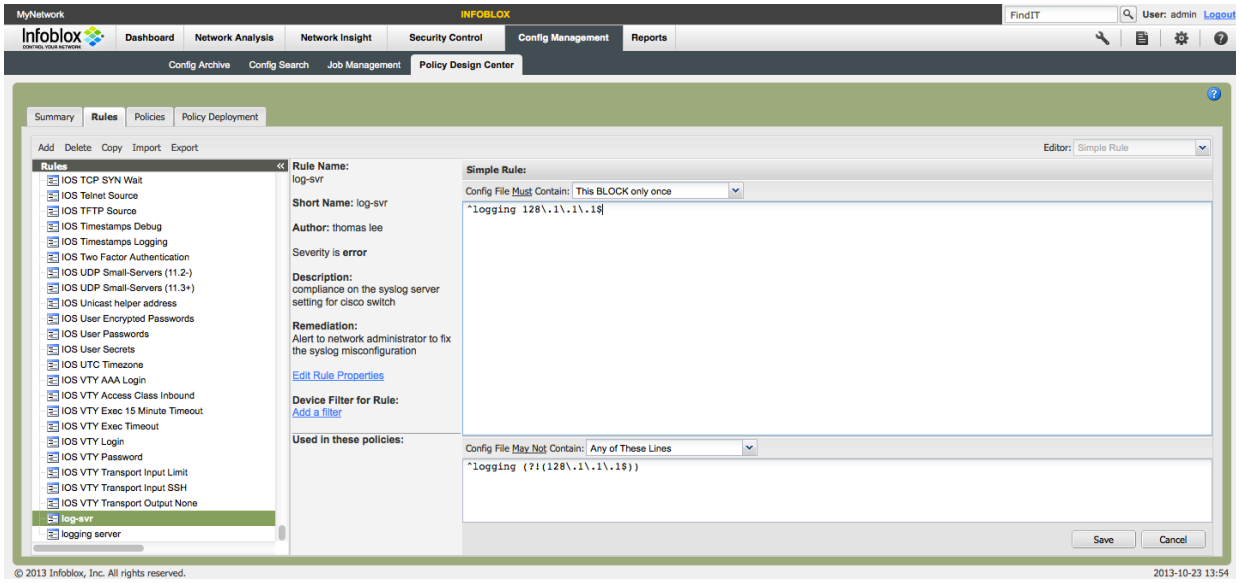


3. Enter a Short Name. The name is limited to 12 characters.
4. Enter author name.
5. Enter the severity. The choices are: Error, Info, or Warning.

A configuration Policy consists of one or more rules. Rules use different forms of regular expression pattern matching against configuration files—and tests of other data Network Automation has collected—to verify that the configuration of the device meets the rule(s). Each rule has a **severity** level, and may optionally define a device filter to limit the types of devices to which it applies. Rules may be freely re-used between

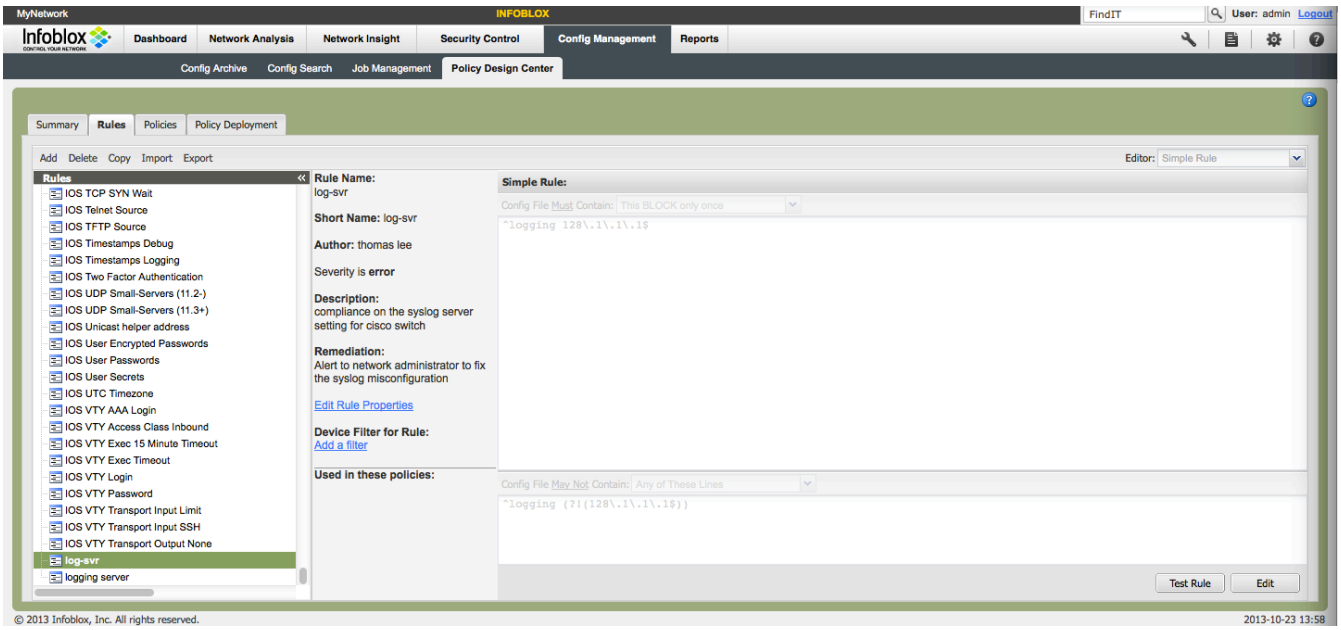
policies.

6. Enter a description of the rule.
7. Enter a remediation description. This text field describes what action should be taken when the policy fails.
8. Click Save to save the rule.

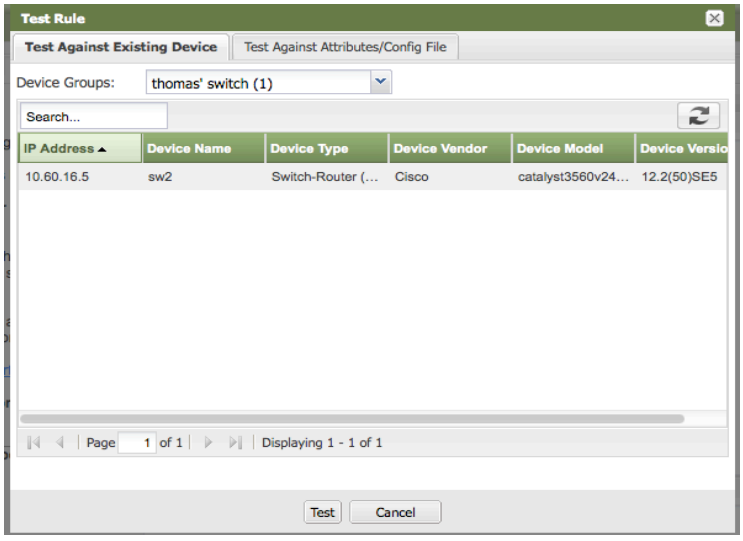


9. With the newly created rule highlighted on the left, select an editor on the upper right side. The choices are: simple editor, CPD, Rule Logic Builder, Raw XML. Refer to the Network Automation Administrator Guide for more information. For this example, simple editor is selected.
10. Enter the configuration line or block in the 'Config file must contain' section. In this case, it is ^logging 128\.1\.1\.1$. In the configuration file of the device, the command is 'logging 128.1.1.1'. Network Automation uses Ruby-style regular expressions. The reason for using the '^', '$', and the '\' characters is to ensure rules engine searches for this exact statement. The '^' character denotes the beginning of a line. The '$' denotes the end of a line. The '\' character denotes treating the subsequent character as a literal character instead of a regular expression. Network Automation rules use Ruby-like regular expressions.

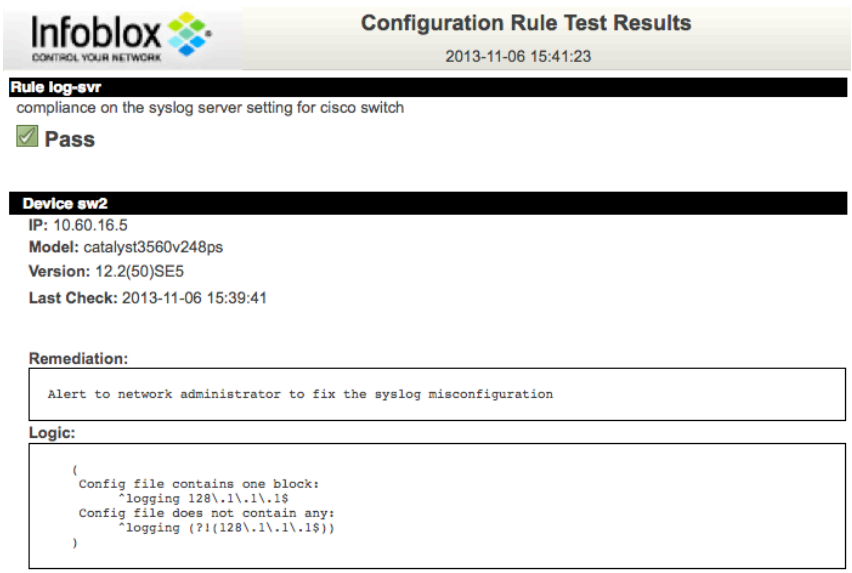
    In the 'Config file may not contain' section, input ^logging (?!(128\.1\.1\.1$)) . The '?!' means do not match 128.1.1.1. However, this section states must not contain the following statement. The overall effect is that if the logging statement IP address differs from 128.1.1.1, an error will be flagged. Refer to the following link on Ruby regex characters: http://www.ruby-doc.org/core-2.0.0/Regexp.html to get more information on regex characters.
11. Click Save. You can now test the rule to ensure the rule is working correctly.



1. Click on the Test Rule button to test the rule. By testing the rule before deploying, you can ensure the rule is written properly without having to deploy a policy.



2. Click ‘Test Against Existing Device’ tab.
3. Select and highlight the device from the device groups.
4. Click on the Test button.



Infoblox
CONTROL YOUR NETWORK

Configuration Rule Test Results

2013-11-06 15:41:23

Rule log-svr

compliance on the syslog server setting for cisco switch

Pass

Device sw2

IP: 10.60.16.5
Model: catalyst3560v248ps
Version: 12.2(50)SE5
Last Check: 2013-11-06 15:39:41

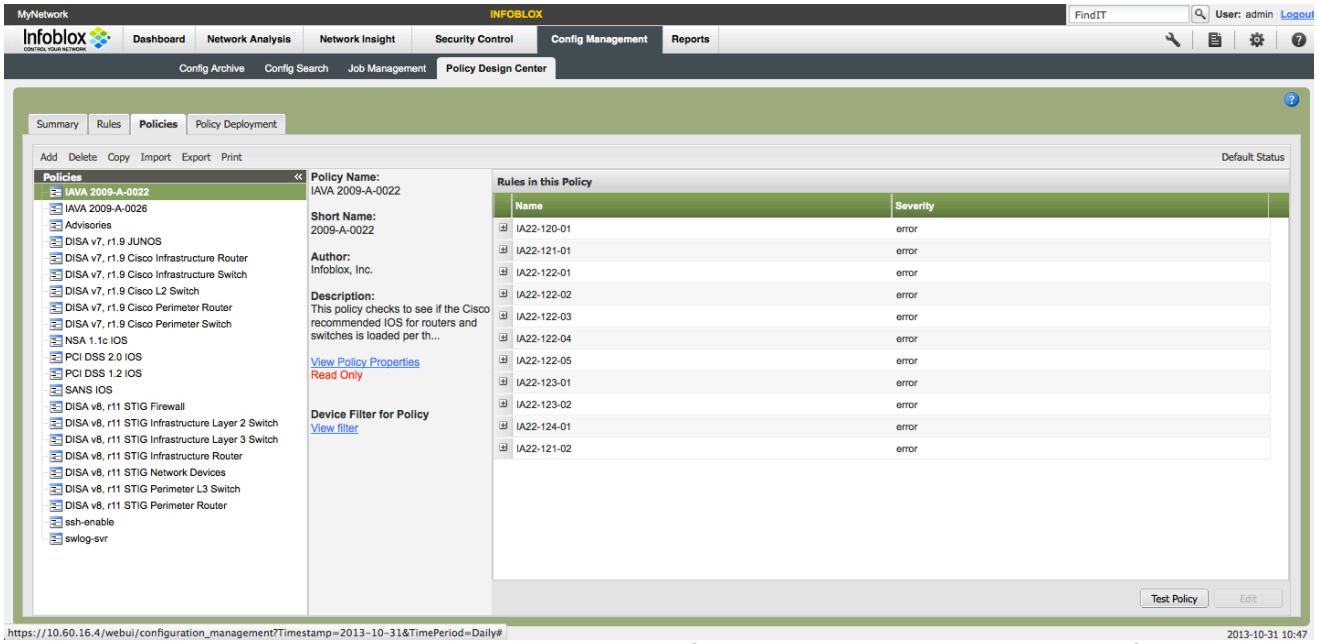
Remediation:

```
Alert to network administrator to fix the syslog misconfiguration
```

Logic:

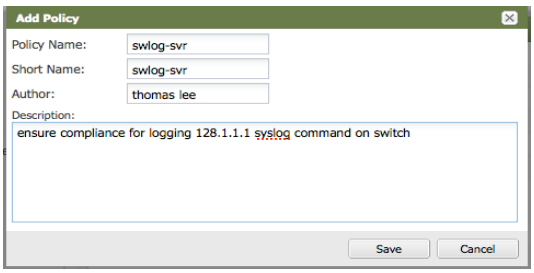
```
(
 Config file contains one block:
      ^logging 128\.1\.1\.1$
 Config file does not contain any:
      ^logging (?!(128\.1\.1\.1$))
)
```

5. It should come back as Pass if the configuration in the configuration file is correct.

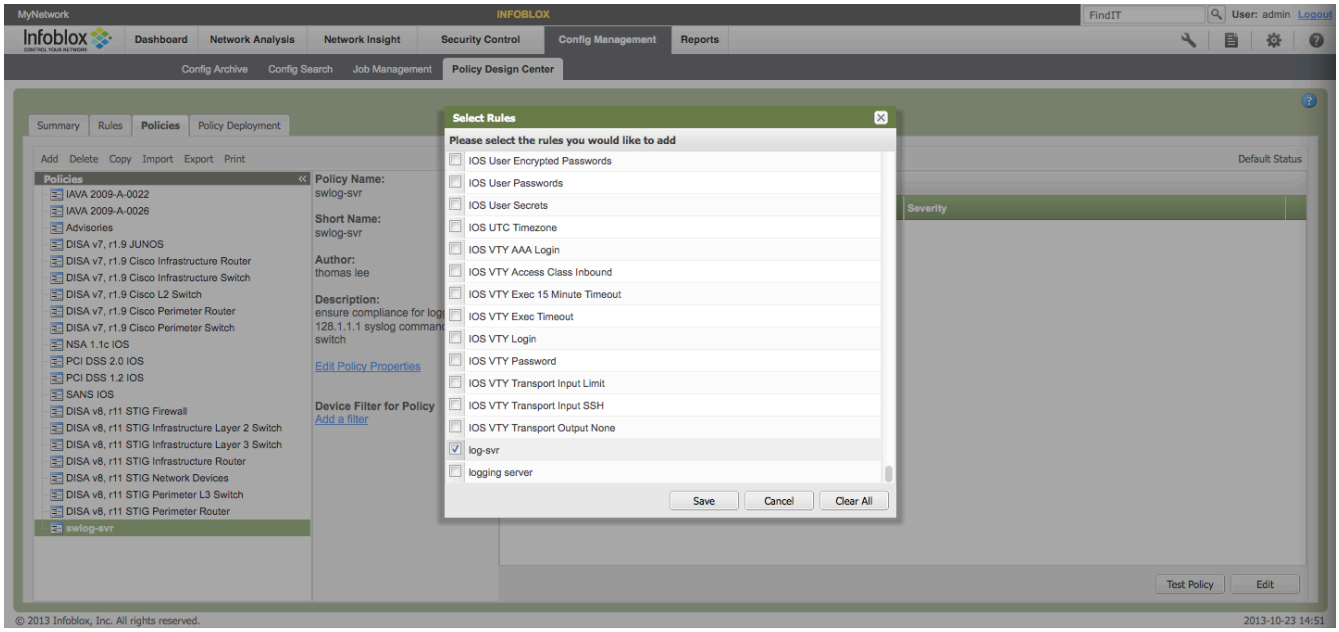


Now that you have created your rule and tested it successfully, you can automate the process of monitoring compliance by placing this rule into a policy and deploy the policy against a device or a group of devices. When a policy is deployed against a device or device group, any detect changes to the configuration file will be examined by this policy. If the change is against this policy, then an error/issue will be posted and if configured, relevant users will be notified.

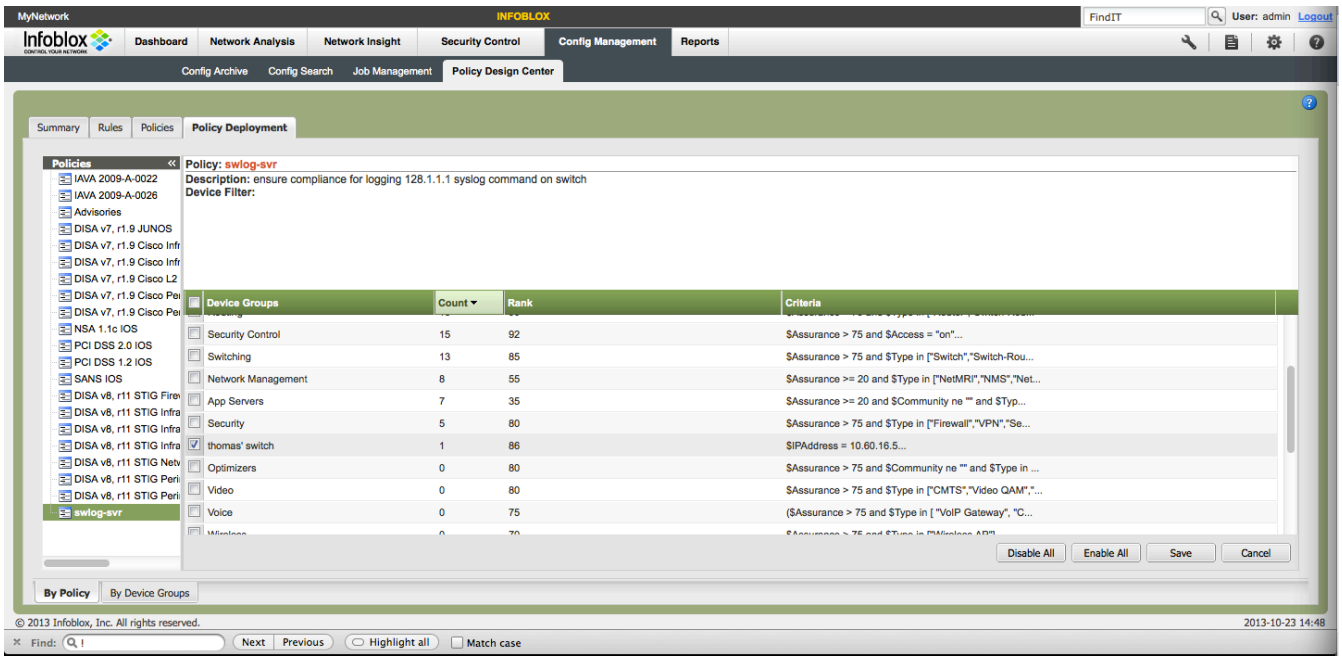
1. Click on the Polices Tab from the previous screen.
2. Click on the Add button to a policy.
3. Enter Policy Name, Short Name, Author, and Description.



4. Click on the Save button.

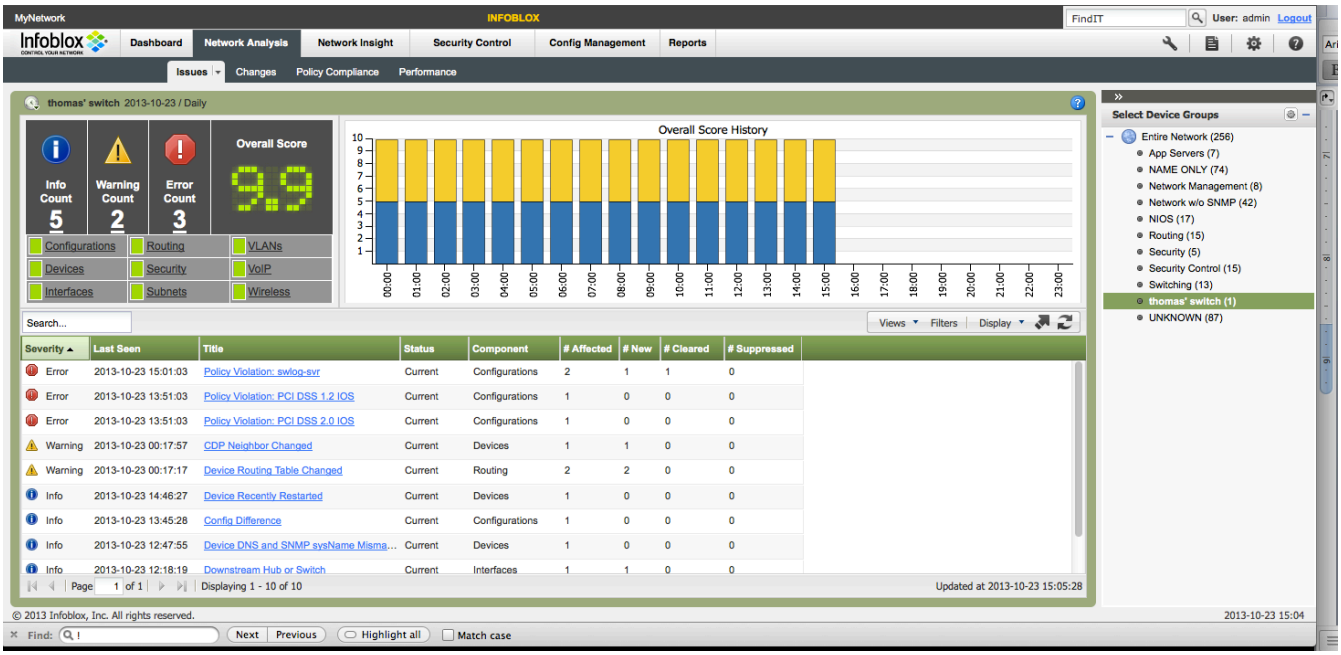


5. Highlight the policy was you just created and saved.
6. Click on the Edit button to add the rule that was created previously. In this example, it's the swlog-svr policy.
7. Click on the rule or rules to be placed into this policy. In this case, we click on the rule called log-svr.
8. Click the Save button.

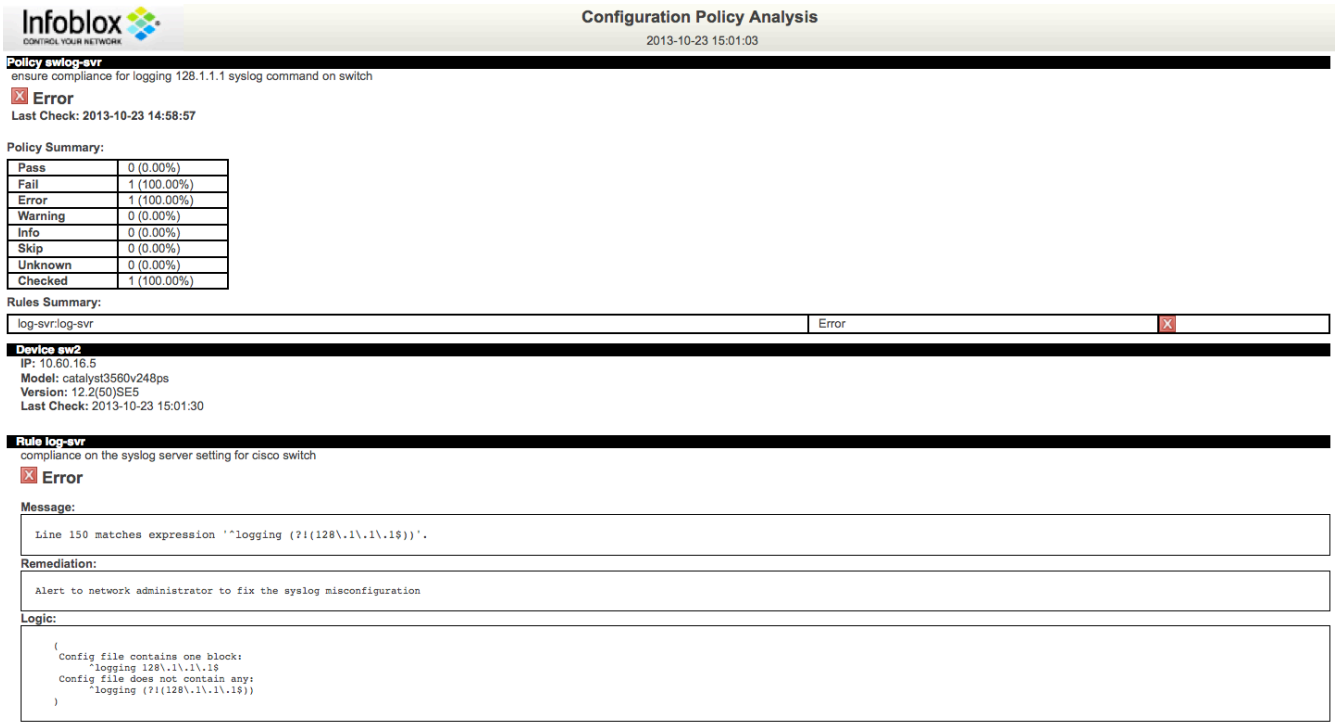


The last step is to deploy the policy. This means you are assigning the policy to the device or device group.

1. Click on the Policy Deployment Tab. Select the device group that you want the policy to enforce.
2. Click on the Save button.
3. Now the policy will examine the latest configuration file for this device or device group to ensure the logging statement is correct.



To test the policy, connect to the device and modify the configuration by adding another logging statement or modify the existing logging statement. Within 15 minutes, you should see an entry appear on the Network Analysis -> Issues screen like the first entry in the list above.



Configuration Policy Analysis
2013-10-23 15:01:03

Policy swlog-svr
ensure compliance for logging 128.1.1.1 syslog command on switch
Error
Last Check: 2013-10-23 14:58:57

Policy Summary:

Pass	0 (0.00%)
Fail	1 (100.00%)
Error	1 (100.00%)
Warning	0 (0.00%)
Info	0 (0.00%)
Skip	0 (0.00%)
Unknown	0 (0.00%)
Checked	1 (100.00%)

Rules Summary:

log-svr:log-svr	Error	X

Device sw2
IP: 10.60.16.5
Model: catalyst3560v248ps
Version: 12.2(50)SE5
Last Check: 2013-10-23 15:01:30

Rule log-svr
compliance on the syslog server setting for cisco switch
Error

Message:

```
Line 150 matches expression '^logging (?!(128\.1\.1\.1$))'.
```

Remediation:

```
Alert to network administrator to fix the syslog misconfiguration
```

Logic:

```
(
 Config file contains one block:
      ^logging 128\.1\.1\.1$
 Config file does not contain any:
      ^logging (?!(128\.1\.1\.1$))
)
```

You can then drill down and see the details of the error. In the screen shot above, the line number is given stating where the error occurred, the remediation to be taken, and the logic of the policy. Based upon this information, you can connect to that device, look at that line number in the configuration file, see the misconfiguration, and then correct the misconfiguration.

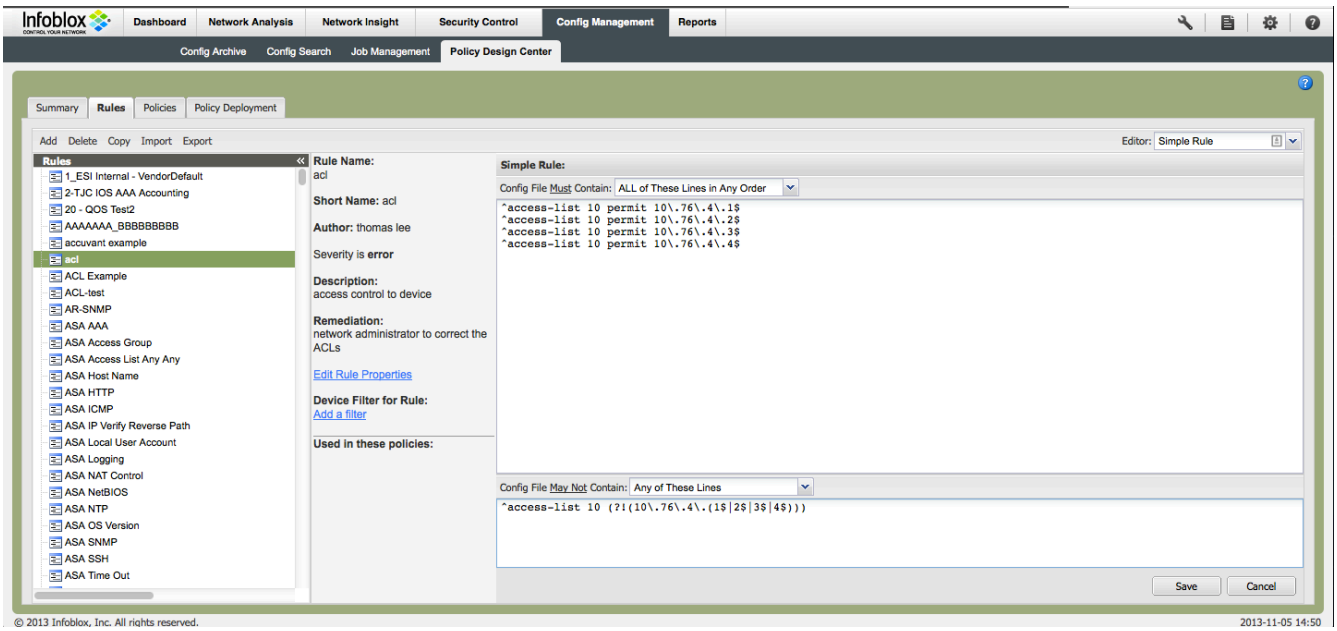
Below are some additional examples of rules and policies to deploy on a Cisco switch

Access control to the device

Access control to a device is done with access lists. For example, you may want to control which IP address(es) are authorized to access a device. You define which IP addresses are allowed to access this device. In this example, the following access lists are entered into the configuration file:

access-list 10 permit 10.76.4.1
access-list 10 permit 10.76.4.2
access-list 10 permit 10.76.4.3
access-list 10 permit 10.76.4.4

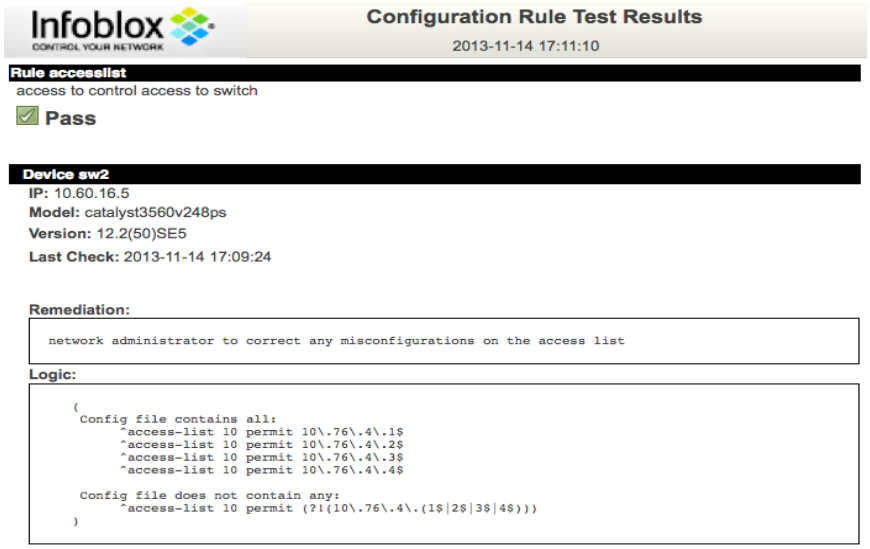
When assigned to the vty section, only IP addresses 10.76.4.1-4 will have access to this switch.



You would follow the same steps in the section for rules creation. The only difference would be the commands. For example, enter the strings below into the respective rules creation screen:

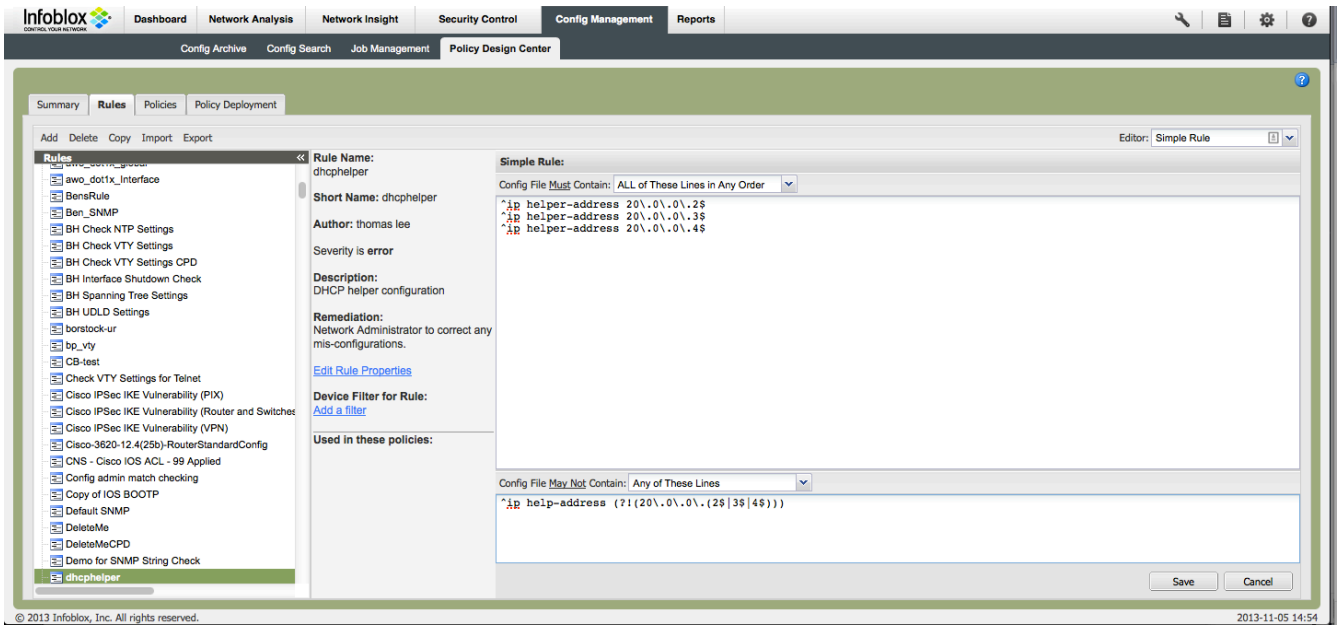
[Config File Must Contain]
^access-list 10 permit 10\.76\.4\.1$
^access-list 10 permit 10\.76\.4\.2$
^access-list 10 permit 10\.76\.4\.3$
^access-list 10 permit 10\.76\.4\.4$

[Config File May Not Contain]
^access-list 10 permit (?!(10\.76\.4\.(1$|2$|3$|4$)))



DHCP helper configuration pointing to a specific DHCP server on a Cisco switch

Configuring the right DHCP helper address(es) ensures the users in their assigned VLANs are given the correct IP subnets. If a computer gets the wrong address from the DHCP server, the computer will not be able to communicate.



You would follow the same steps in the section for rules creation. The only difference would be the commands. For example, enter the strings below into the respective rules creation screen:

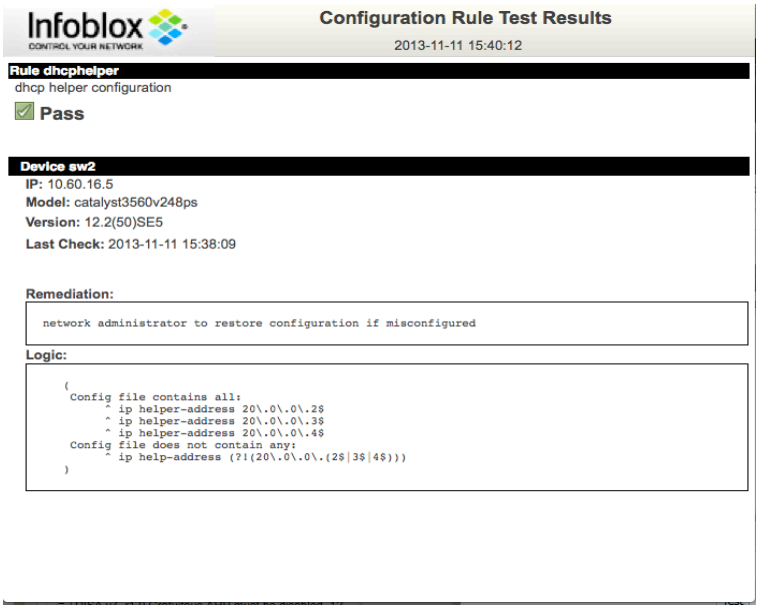
[Config File Must Contain]
^ip helper-address 20\.0\.0\.2$
^ip helper-address 20\.0\.0\.3$
^ip helper-address 20\.0\.0\.4$

[Config File May Not Contain]
^ip help-address (?!(20\.0\.0\.(2$|3$|4$)))

In the configuration file segment, the IP helper addresses are configured correctly:

interface Vlan1
 ip address 10.60.16.5 255.255.255.0
 ip helper-address 20.0.0.2
 ip helper-address 20.0.0.3
 ip helper-address 20.0.0.4

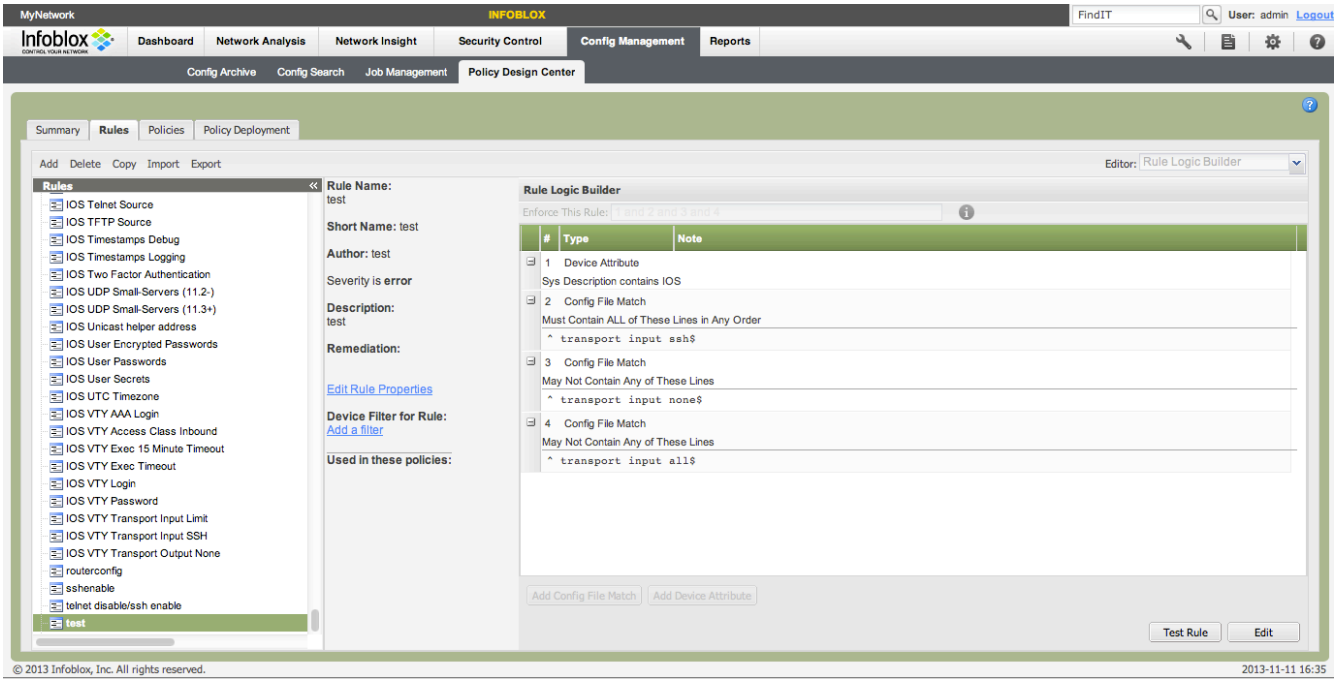
When the rule is tested, the following result is displayed:



This test passed as indicated in the upper left corner.

Telnet disabled on a Cisco switch

Disabling telnet access on devices is good security practice. In telnet, all characters typed and responses are sent in the clear. In SSH, all characters typed and responses are encrypted before transmission.



You would follow the same steps in the section for rules creation. The only difference would be the commands. For example, enter the strings below into the respective rules creation screen using the rule logic builder:

[Config File Must Contain]
^ transport input ssh$

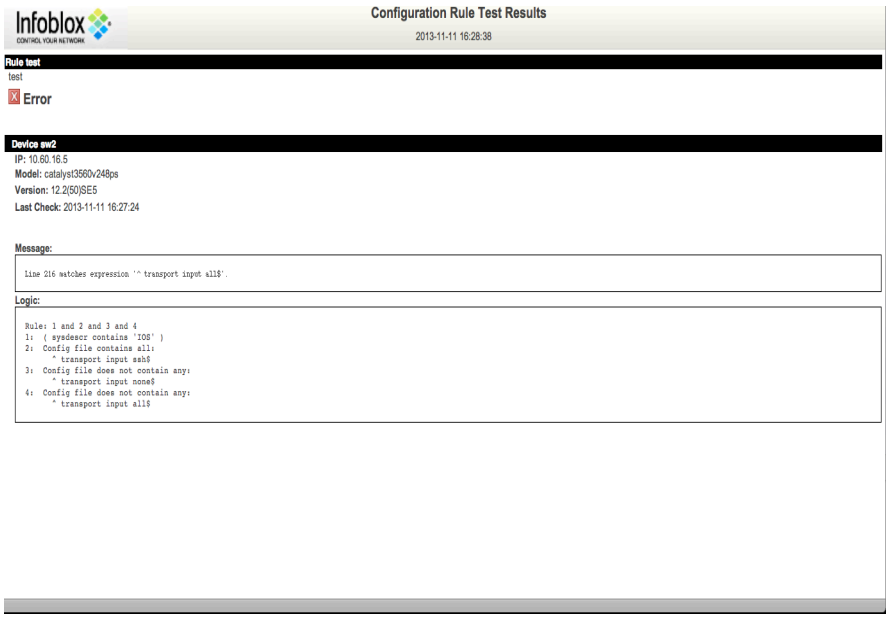
[May Not Contain Any of These Lines]
^ transport input all$

[May Not Contain Any of These Lines]
^ transport input none$

Line 216 below contains a misconfiguration based upon the rule above which is to ensure ‘transport input ssh’ is the only acceptable command

```
207 line vty 0 4
208  session-timeout 15
209  password infoblox
210  login local
211  transport input ssh
212 line vty 5 15
213  session-timeout 15
214  password infoblox
215  login local
216  transport input all
```

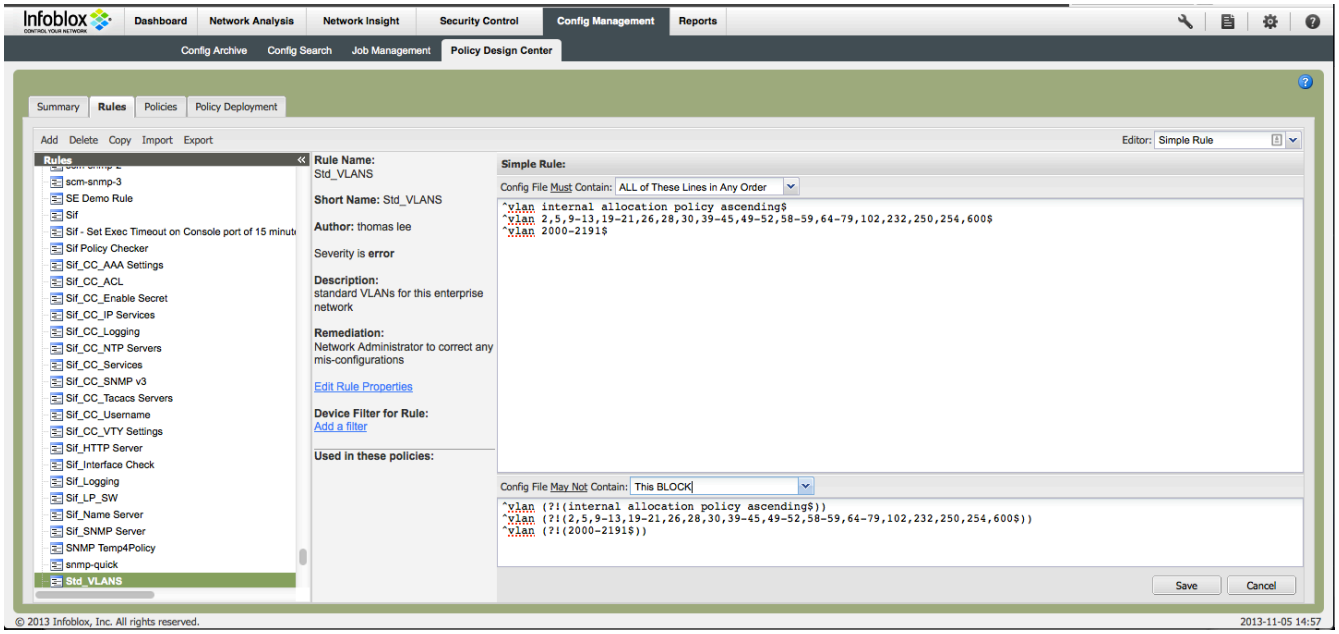
When the rule is tested against this configuration, the following screen will appear:



An error is indicated in the upper left portion of the screen and the message indicates an error on line 216.

Standard VLAN numbers

Standardizing VLAN numbers within your network is good network practice. This is to ensure configuration consistency.



You would follow the same steps in the section for rules creation. The only difference would be the commands. For example, enter the strings below into the respective rules creation screen:

[Config File Must Contain]
^vlan internal allocation policy ascending$
^vlan 2,5,9-13,19-21,26,28,30,39-45,49-52,58-59,64-79,102,232,250,254,600$
^vlan 2000-2191$

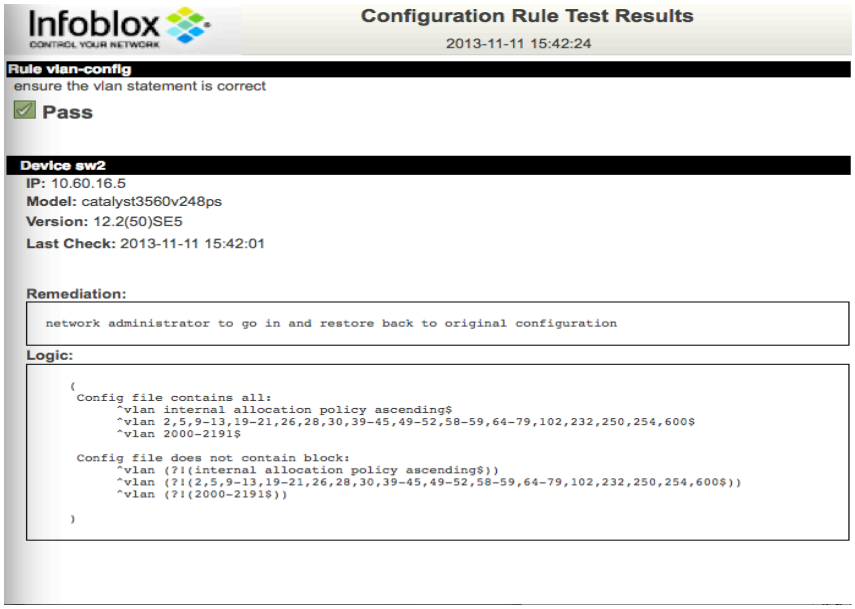
[Config File May Not Contain]
^vlan (?!(internal allocation policy ascending$))
^vlan (?!(2,5,9-13,19-21,26,28,30,39-45,49-52,58-59,64-79,102,232,250,254,600$))
^vlan (?!(2000-2191$))

Note: Set the 'Config File May Not Contain' pull down menu to 'This Block.'

In the configuration segment, the VLAN configuration is correct:

vlan internal allocation policy ascending
vlan 2,5,9-13,19-21,26,28,30,39-45,49-52,58-59,64-79,102,232,250,254,600
vlan 2000-2191

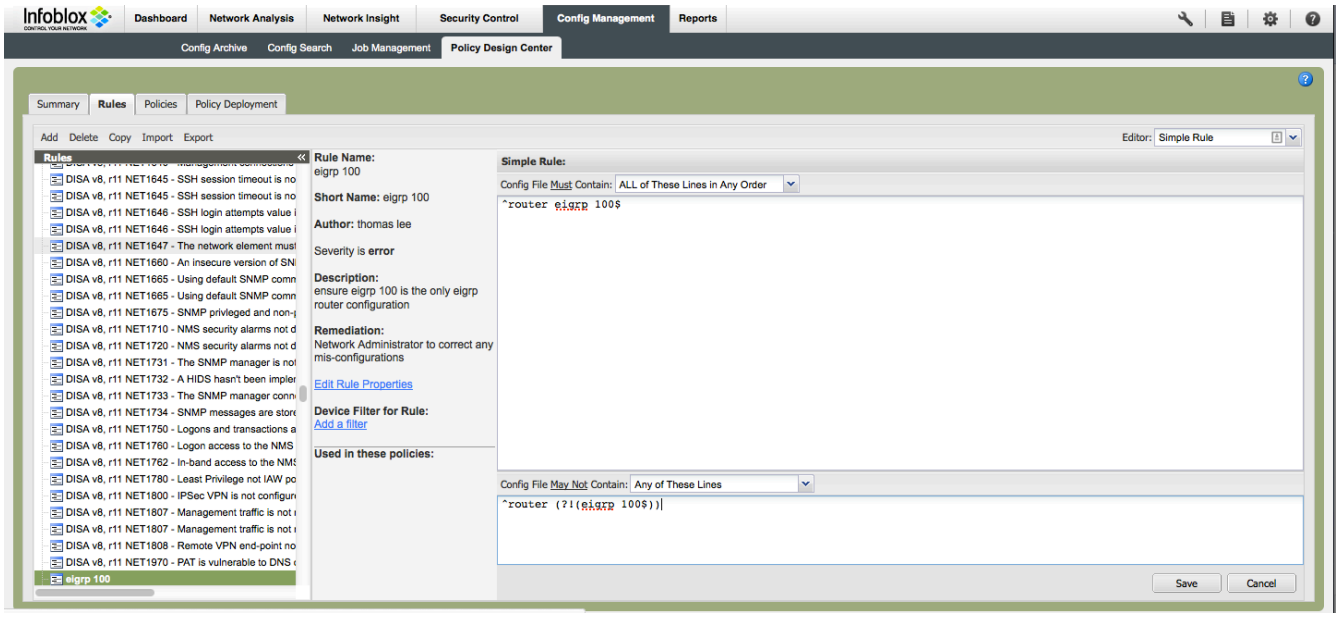
When the rule is test, the following result is displayed:



The rule passed as indicated in the upper left corner.

Standard routing protocol on a Cisco switch

Ensuring your routing protocol standards are correct is good network operations practice. This is to ensure memory and CPU cycles are not utilized for unauthorized purposes.



You would follow the same steps in the section for rules creation. The only difference would be the commands. For example, enter the strings below into the respective rules creation screen:

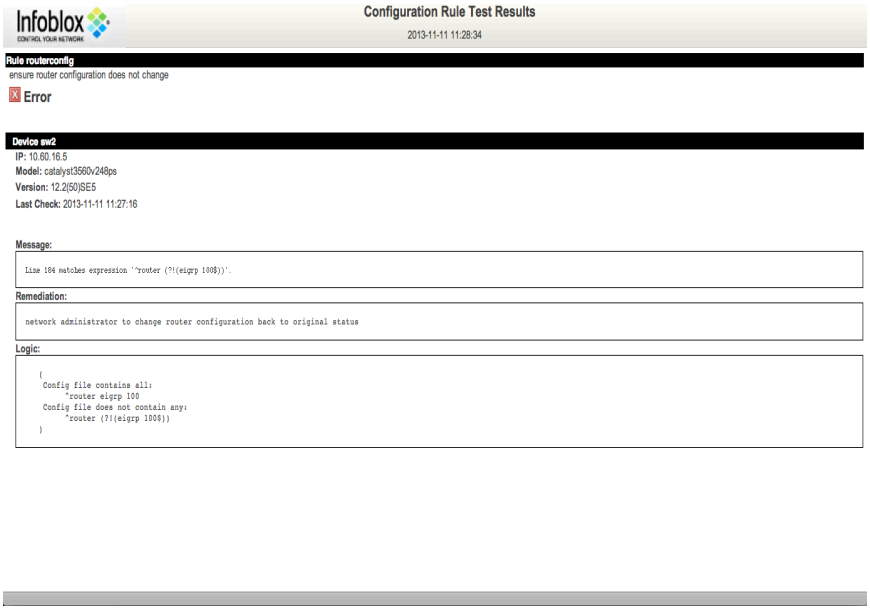
[Config File Must Contain]
^router eigrp 100$

[Config File May Not Contain]
^router (?!(eigrp 100$))

In the configuration segment below, you will see a misconfiguration:

180 router eigrp 100
181 eigrp stub connected summary
182 !
183 !
184 router eigrp 200
185 eigrp stub connected summary
186 !

Line 184 is the misconfiguration. When the rule above is tested, Network Automation returns the following result:



An error is indicated in the upper left corner. In the message section, line 184 is noted as the line in which the error was detected. You can now go to that line in the configuration file to resolve the error.