



February 20 - 24, 2017 • Berlin



Your Time Is Now

Configuration Compliance Management with Prime Infrastructure

Gilles Clugnac – Technical Leader

BRKNMS 2036

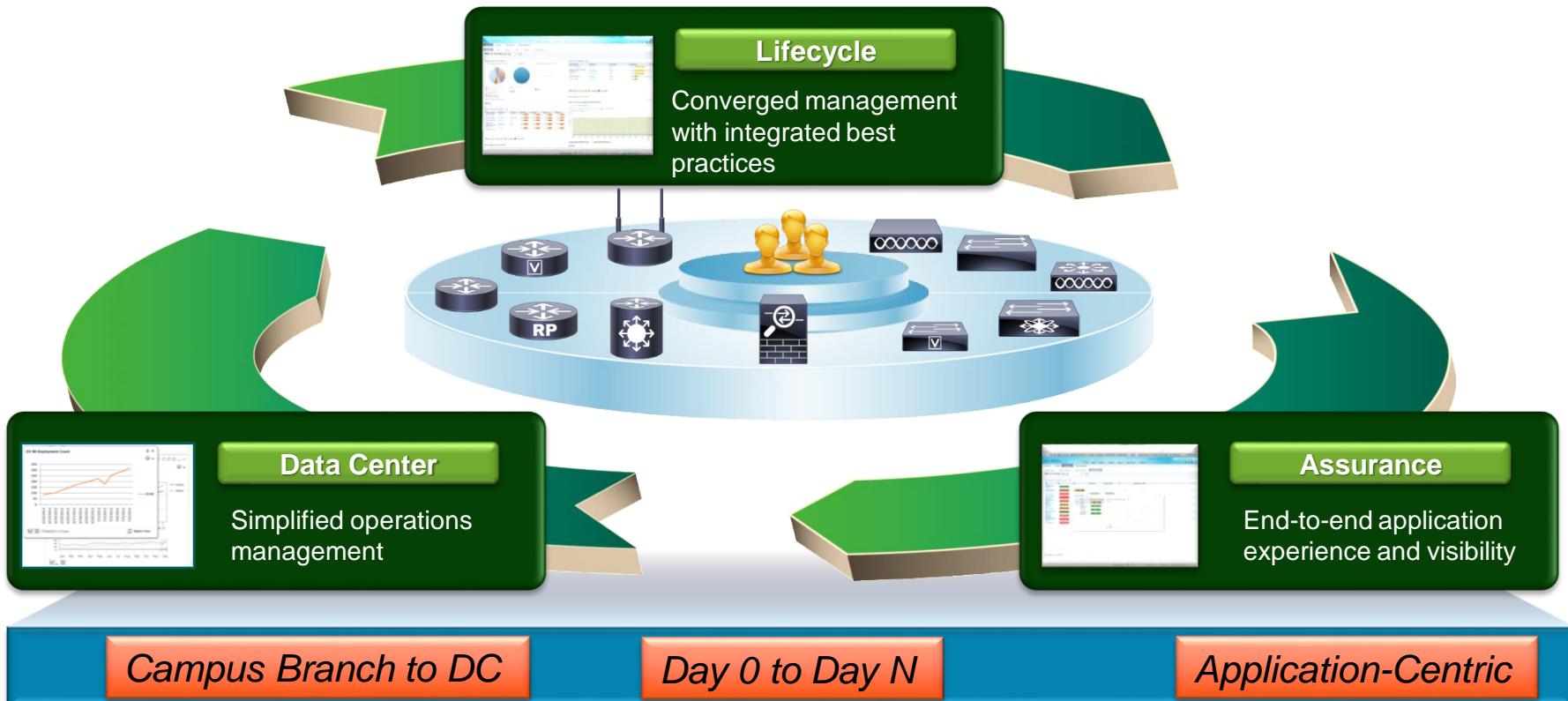
Agenda

- Introduction
- Prime Infrastructure Compliance Engine Concepts
- Getting started with compliance
- More realistic use cases
- Conclusion

Introduction

Cisco Prime Infrastructure

Realizing the Vision of One Management



Enterprise Management (1 of End-to-End Lifecycle Management)

- Centralized lifecycle management - discovery, inventory, configuration, SWIM, and proactive/reactive monitoring

- Compliance Baseline - Audit device configurations

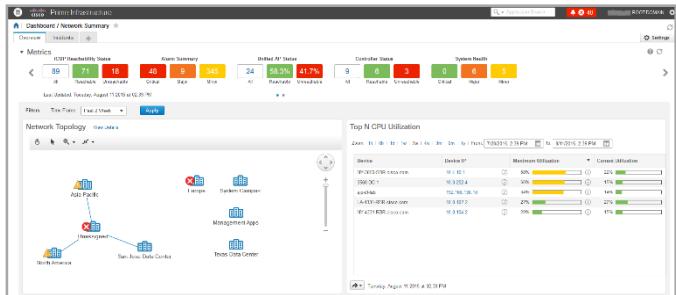
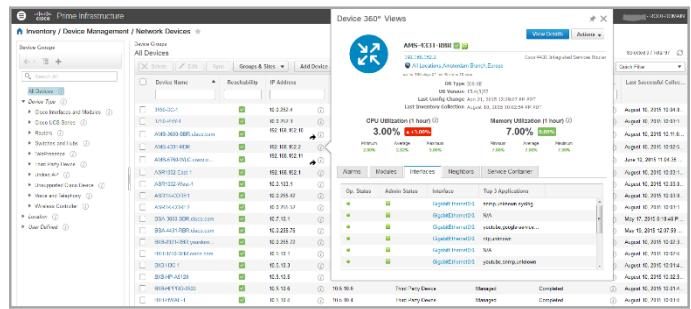
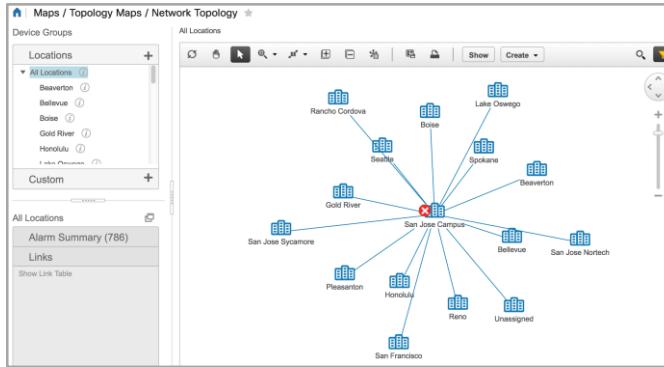
- Advanced troubleshooting of wired and wireless infrastructure issues

- Rapid device support through Device Packs for new Cisco® devices, routers, switches, controllers, access points, Nexus® technology, and more

- Customizable configuration templates based on Cisco validated designs and guided workflows, including IWAN support

Cisco Unified Access™ management and client tracking

- Seamless integration with Cisco Identity Services Engine (ISE) for simplified troubleshooting
- Seamless integration with Cisco Mobility Services Engine (MSE) for location-based services, rogue detection, etc.

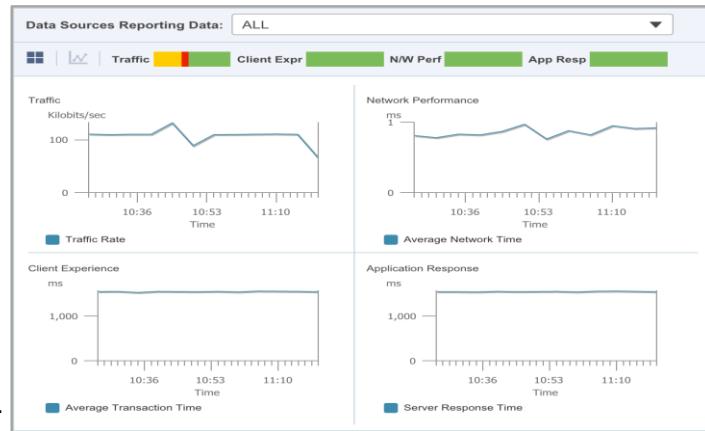


Assurance

Enterprise Management (2 of 2)

Application Experience and End User Experience

- **End-to-end visibility** for service-aware networking by applications, services, and end users
- **Out-of-the-box support** for Cisco® advanced technologies, including AVC , NetFlow, Flexible NetFlow, NBAR2, Performance Agent, Medianet, and more
- **Service health dashboard** allows quick health check on your business-critical applications
- **Simplified troubleshooting** of applications and client access issues
- **QoS Configuration / Monitoring** applied to and class-based traffic patterns
- **Multi-NAM management**
 - Traffic analysis
 - Application response time metrics
 - Packet capture and decode



Site	telnet	http	snmp	rtp	cisco-jabber-control	cisco-jabber-audio
Asia Pacific	✓	✓	✗	✓	✓	✓
Europe	✓	✓	✓	✓	✓	✓
Management Apps	⚠	✗	✗	✓	✓	✓
North America	✗	⚠	✗	✗	✓	✓
San Jose Data Center	⚠	✓	✓	✓	✓	✓
Texas Data Center	✓	✗	✗	✗	✓	✓
Unassigned	✓	✗	✗	✗	✓	✗

Compliance

- Audit the configuration of devices to assess compliance with baseline policies
- Predefined library of policies for best practices audit
- User defined policies



Configuration Baseline Compliance

The screenshot shows the 'Compliance Policy Selector' interface. On the left, under 'Title', the 'Trap Destination' policy is selected. On the right, the 'Select Rules and Inputs for the Policy: Trap Destination' dialog is open, showing the configuration for a trap destination. The 'Destination' field is set to '172.20.117.149' and the 'Community String' field is set to 'public'. A note indicates support for IOS, IOS-XE, and IOS-XR.

The screenshot shows the 'Condition Match Criteria' section for the 'Trap Destination' policy. It defines a condition where the 'Operator' is 'Contains the string' and the 'Value' is 'snmp-server host <_Destination> version 2c <_Community_String>'.

- Introduced in Prime Infrastructure 3.0
- Define configuration baseline policies
- Perform compliance audits
- View compliance audit violations
- Option to fix violations
- Support for IOS, IOS-XE, IOS-XR, NXOS, **AireOS*** and ASA devices
- Gen2 Appliance or **Standard**** OVA required

Prime Infrastructure Compliance Engine Concepts

Compliance Conceptual Model

Policies

Define granular “per-feature” level compliance rules

Profiles

Aggregate multiple compliance policies into larger sets of policies

Used when performing compliance **audits**

Jobs

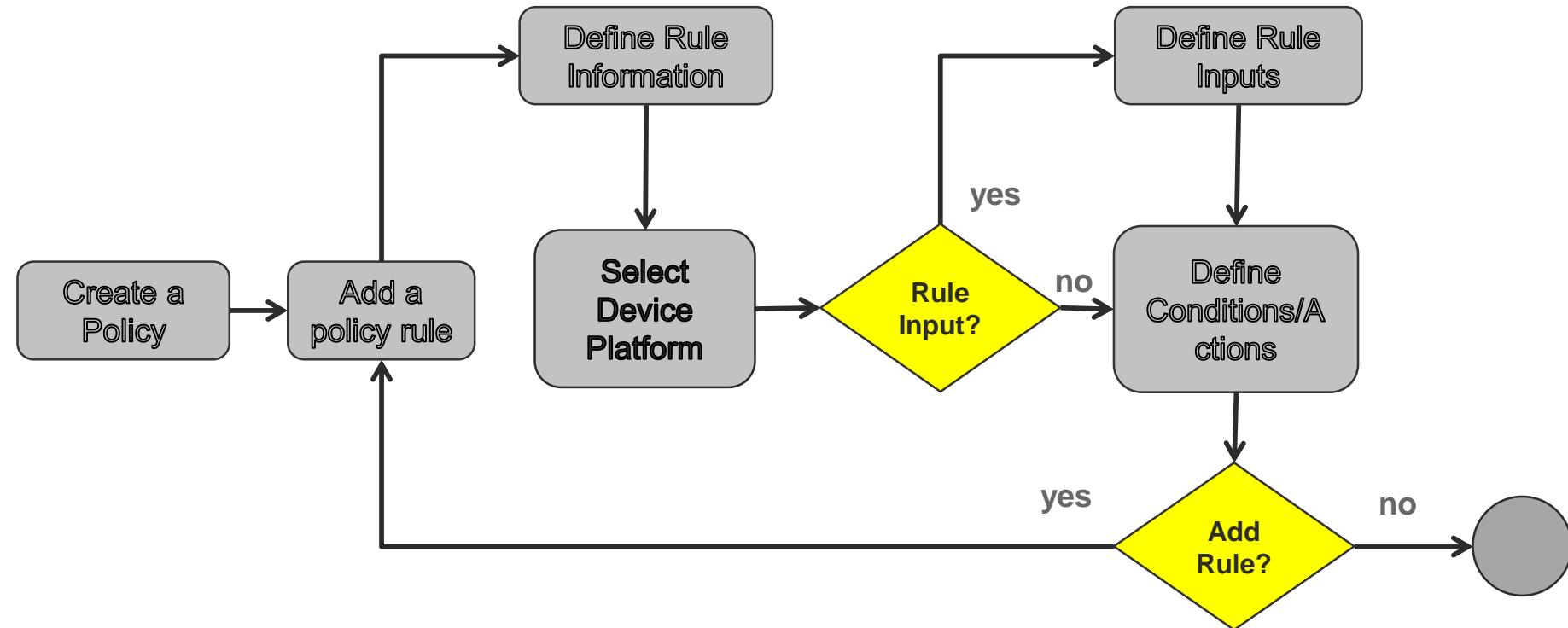
Maps a specific profile against a specific set of network devices

Perform compliance audits to detect compliance **violations**

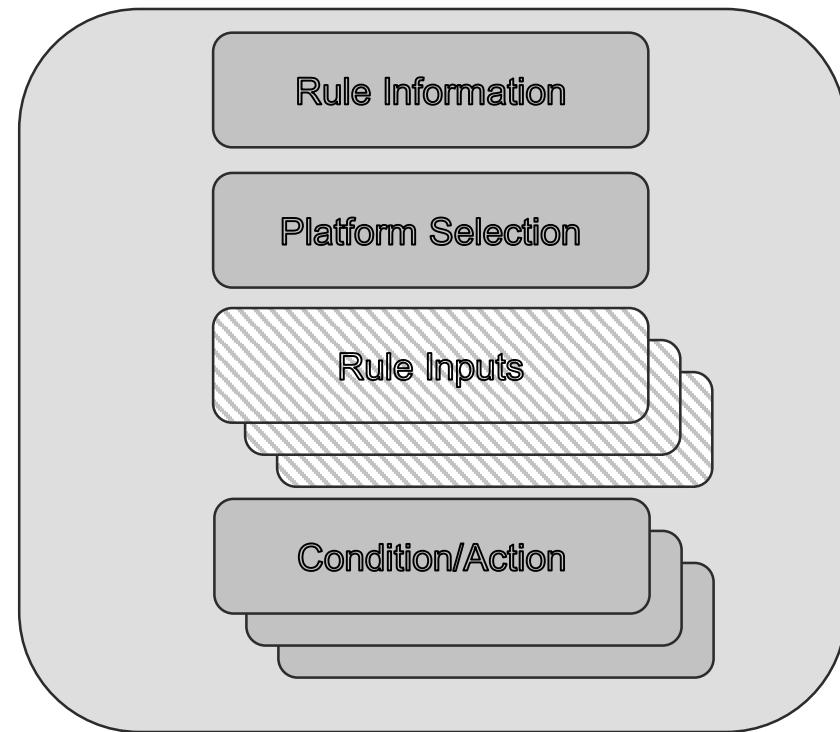
Compliance Process Overview

1. Configure Compliance Policies which contain Policy Rules
2. Configure a Compliance Profile which contains system provided and/or user defined Policies
3. Run Compliance Audit
4. Evaluate audit result
5. Based on results, make correction by running a fix job
6. Re-run the audit to validate

Summary of a Compliance Policy definition flow

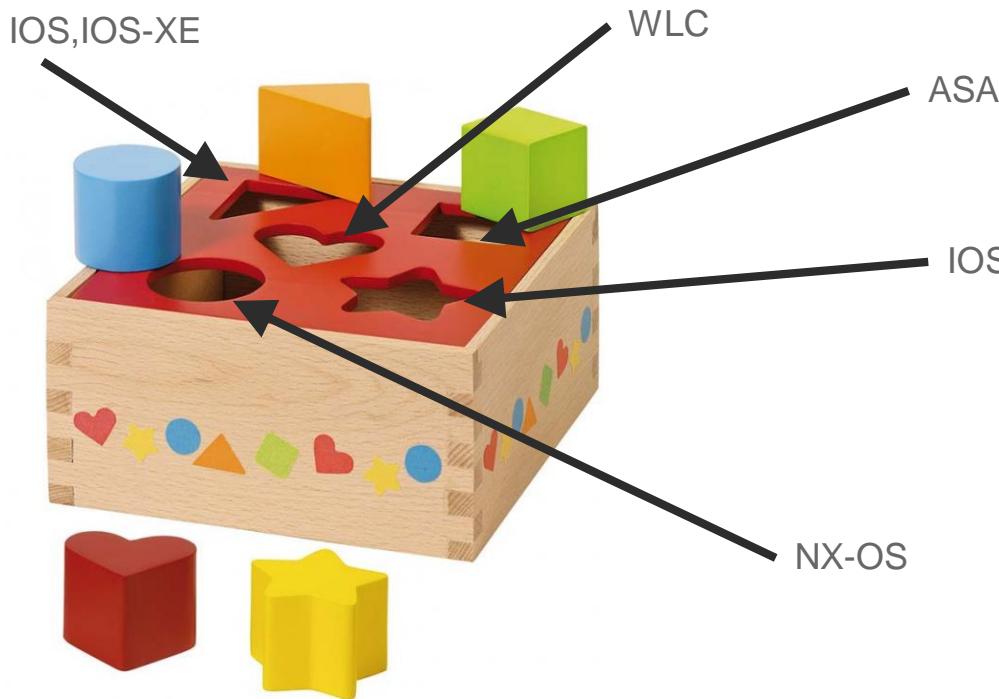


contains



- **Rule Information** – Name, Description, Impact, Suggested Fix
- **Platform Selection** – IOS, IOS-XE, IOS-XR, NXOS, ASA, AireOS
- **Rule Inputs** (optional) – string, IP address, boolean, etc
- **Conditions and Actions** – 1 or more (ordered list)

Platform Selection



A single Policy with multiple rules for different platforms

Devices will be audited only with the policies matching their platform type

Rule Inputs

Parameters provided to a rule

New Rule Input



*Title

*Identifier

Description

Scope

Data Type

Input Required

Accept Multiple Values

Default Value

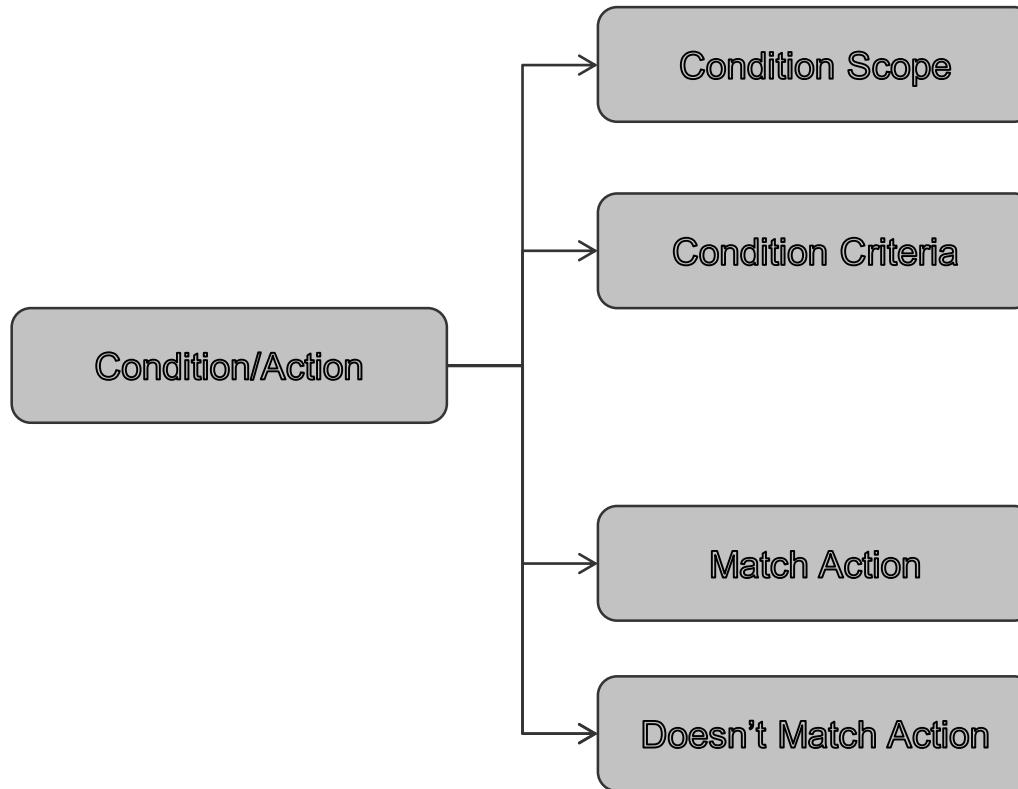
Preview

OK

Cancel

- Identifier used in :
 - Condition match
 - Message
 - Fix CLI
- Scope can be:
 - Execution
 - Fix

A rule contains one or more Condition/Action



Regex survival kit : the minimum to understand this presentation (1/2)



- . : matches any character except newline
- * : any repetition. Example .* : matches any character, any repetition
- ^ : starts with
- \$: ends with
- (exp) : when matches, creates a **capture a group** which can be reused in a **back reference**
- <x.y> : uses a *back reference*: *x = condition number of the capture group, y = capture group number in the condition*

Regex survival kit : the minimum to understand this presentation (2/2)

- (st1|st2) : match of st1 or st2 with capture
- \n : new line
- \s : space or tab
- \S : any character except space, tab or new line
- \d : digit



Regex survival kit : the minimum to understand this presentation (2/2)



PCRE

```
/^s*(((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{1,4}|:))|(((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{6}:[0-9A-Fa-f]{1,4}|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])){3}|:))|(((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{5}(((0-9A-Fa-f){1,4}{1,2}|:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])){3}|:))|(((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{4}(((0-9A-Fa-f){1,4}{1,3}|:((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{1,4}|?((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])){3}|:))|(((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{3}(((0-9A-Fa-f){1,4}{1,4}|:((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{1,4}{1,4})|((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{0,2}|:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])){3}|:))|(((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{2}(((0-9A-Fa-f){1,4}{1,5}|:((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{0,3}|:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])){3}|:))|(((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{1,6}|:((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{0,4}|:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])){3}|:))|(((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{1,7}|:((0-9A-Fa-f){1,4}:[0-9A-Fa-f]{0,5}|:((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])){3}|:))|((%+.)*$
```

[open regex in editor](#) ↗

Regex survival kit : the minimum to understand this presentation (2/2)



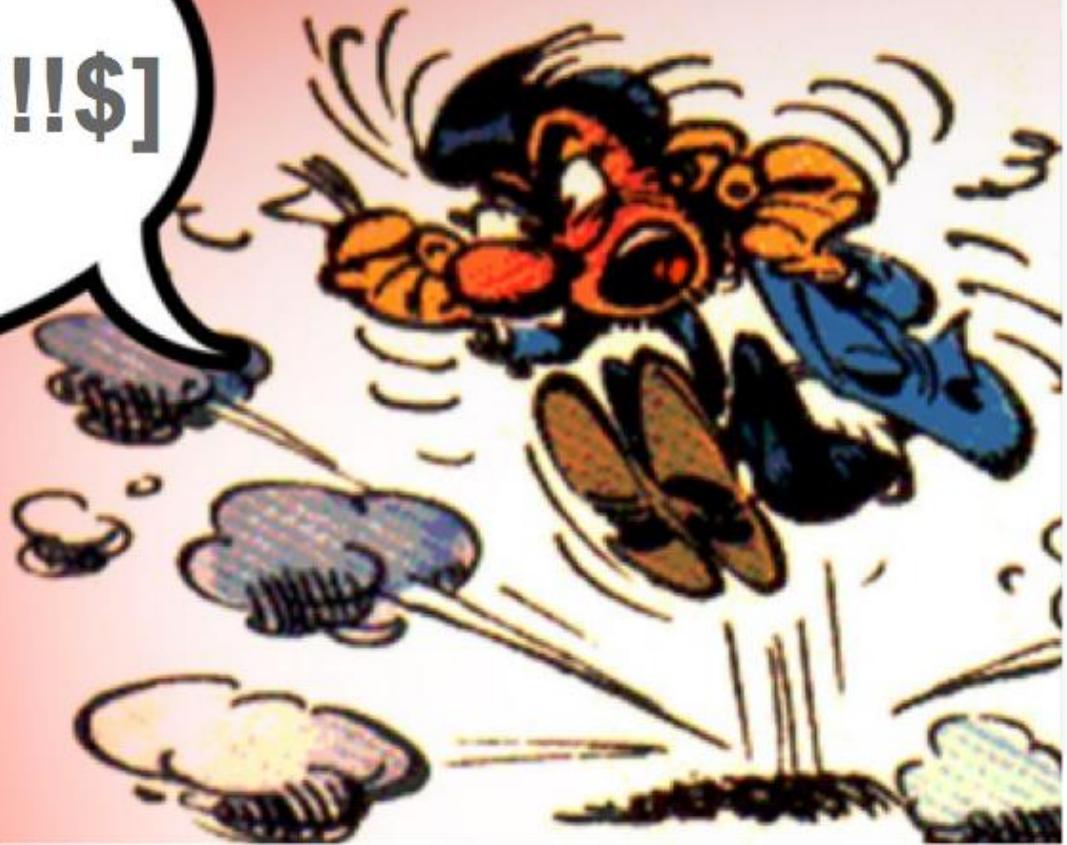
IPV6

PCRE

/^\s*(((0-9A-Fa-f){1,4}:){7}|((0-9A-Fa-f){1,4}I:)|(((0-9A-Fa-f){1,4}:){6}|([0-9A-Fa-f]{1,4}|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|(\.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9]))|{3}|I:)|(((0-9A-Fa-f){1,4}:){5}|(((0-9A-Fa-f){1,4}:){1,2}|I:|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9]))|{3}|I:)|(((0-9A-Fa-f){1,4}:){4}|(((0-9A-Fa-f){1,4}:){1,3}|I:|(((0-9A-Fa-f){1,4}):|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9]))|{3}|I:)|(((0-9A-Fa-f){1,4}:){3}|(((0-9A-Fa-f){1,4}:){1,4}|I:|(((0-9A-Fa-f){1,4}:){0,2}:|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|(\.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9]))|{3}|I:)|(((0-9A-Fa-f){1,4}:){2}|(((0-9A-Fa-f){1,4}:){1,5}|I:|(((0-9A-Fa-f){1,4}:){0,3}:|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|(\.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9]))|{3}|I:)|(((0-9A-Fa-f){1,4}:){1}|(((0-9A-Fa-f){1,4}:){1,6}|I:|(((0-9A-Fa-f){1,4}:){0,4}:|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|(\.\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9]))|{3}|I:)|(((0-9A-Fa-f){1,4}:){1,7}|I:|(((0-9A-Fa-f){1,4}:){0,5}:|((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[1-9]?[0-9])|{3}|I:))|(.+)?\s*/\$

[open regex in editor](#) ↗

[^\?:(#)\.*@\|S!!\$]



Tip: use Online Regex tester like <https://regex101.com/>

Screenshot of the regex101.com website showing a regex test.

REGULAR EXPRESSION: /ntp server ((?:[0-9]{1,3}\.){3}[0-9]{1,3}\$) / gmixXsuUAJ

TEST STRING: ntp server 10.1.1.1

EXPLANATION:

- /ntp server ((?:[0-9]{1,3}\.){3}[0-9]{1,3}\$) /
ntp server matches the characters **ntp server** literally (case sensitive)
- 1st Capturing group ((?:[0-9]{1,3}\.){3}) Non-capturing group
Quantifier: {3} Exactly 3 times
 - [0-9]{1,3} match a single character present in the list below
Quantifier: {1,3} Between 1 and 3 times, as many times as possible, giving back as needed [greedy]
 - 0-9 a single character in the range between 0 and 9
 - . matches the character . literally
 - [0-9]{1,3} match a single character present in the list below
Quantifier: {1,3} Between 1 and 3 times, as many times as possible, giving back as needed [greedy]

MATCH INFORMATION:

Match	Range	Value
1.	[11-19]	`10.1.1.1`



Rule Condition Scope & Block Options

Condition Scope Details

Condition Scope	Configuration
Device Property	Configuration
Show Commands	Device Command Outputs Device Properties Previously Matched Blocks

Scope controls what information is checked

- Configuration**
- Command Outputs**
Show commands, etc
- Device Properties**
Device Name, IP Address, OS Name, OS Version
- Previously Matched Block**

Block Options

<input checked="" type="checkbox"/> Parse as Blocks	
Block Start Expression	^interface .
Block End Expression	

Block Options

Check inside config sub-mode blocks

Typical uses:

Interface

Router

Conditions and Actions

Condition operations

String compare (contains / does not contain)

Regular Expressions (match / doesn't match)

Evaluate Expression

Execute Function

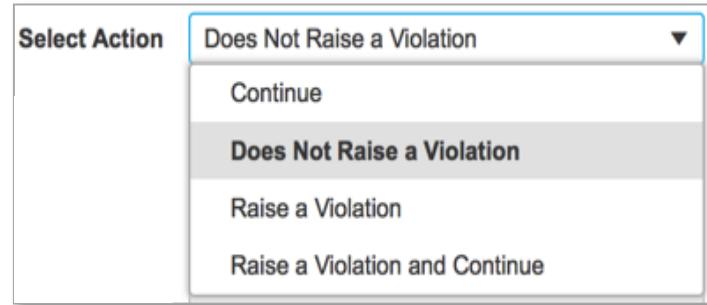
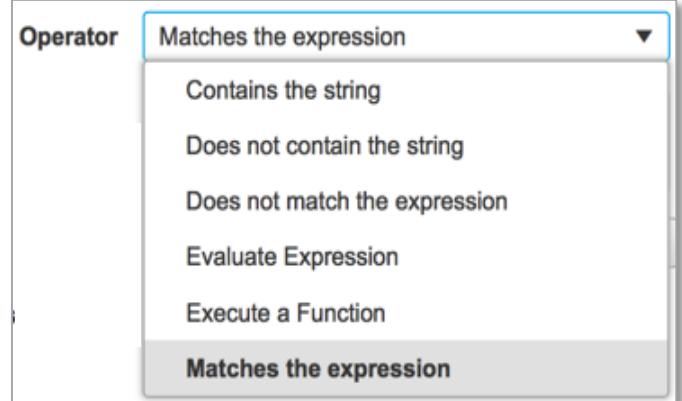
Actions

Continue – keep checking, go on to succeeding Condition

Does Not Raise a Violation – stop checking, all is good, no more checking needed

Raise a Violation – raise a violation and stop checking

Raise a Violation and Continue – raise a violation and keep checking, go on to succeeding Condition



Conditions: String and Expression Matching

The image shows three examples of configuration conditions:

- String Compare:** Operator: Contains the string. Value: snmp-server host <_Destination> version 2c<_Community_String>. A red box highlights the placeholder <_Destination>.
- Regular Expression Support:** Operator: Matches the expression. Value: ip access-group (.*). A blue box highlights the regular expression pattern (.*).
- Regular Expression Support:** Operator: Matches the expression. Value: ^ip access-list (standard|extended) <4.1>\$. A blue box highlights the regular expression pattern <4.1>\$.

A purple arrow points from the highlighted <_Destination> placeholder in the first example down to the highlighted <4.1>\$ placeholder in the third example, indicating they represent the same concept of a back-referenced value.

String Compare

Checks that line contains string

Rule Inputs can be inserted

Regular Expression Support

Single line regular expression

Parenthesis capture values

Back Reference previously captured values
<condition#.value#>

Actions: Violation Handling

Either “Match” or “Does Not Match” Condition could be a Violation

The screenshot shows a configuration interface with the following fields:

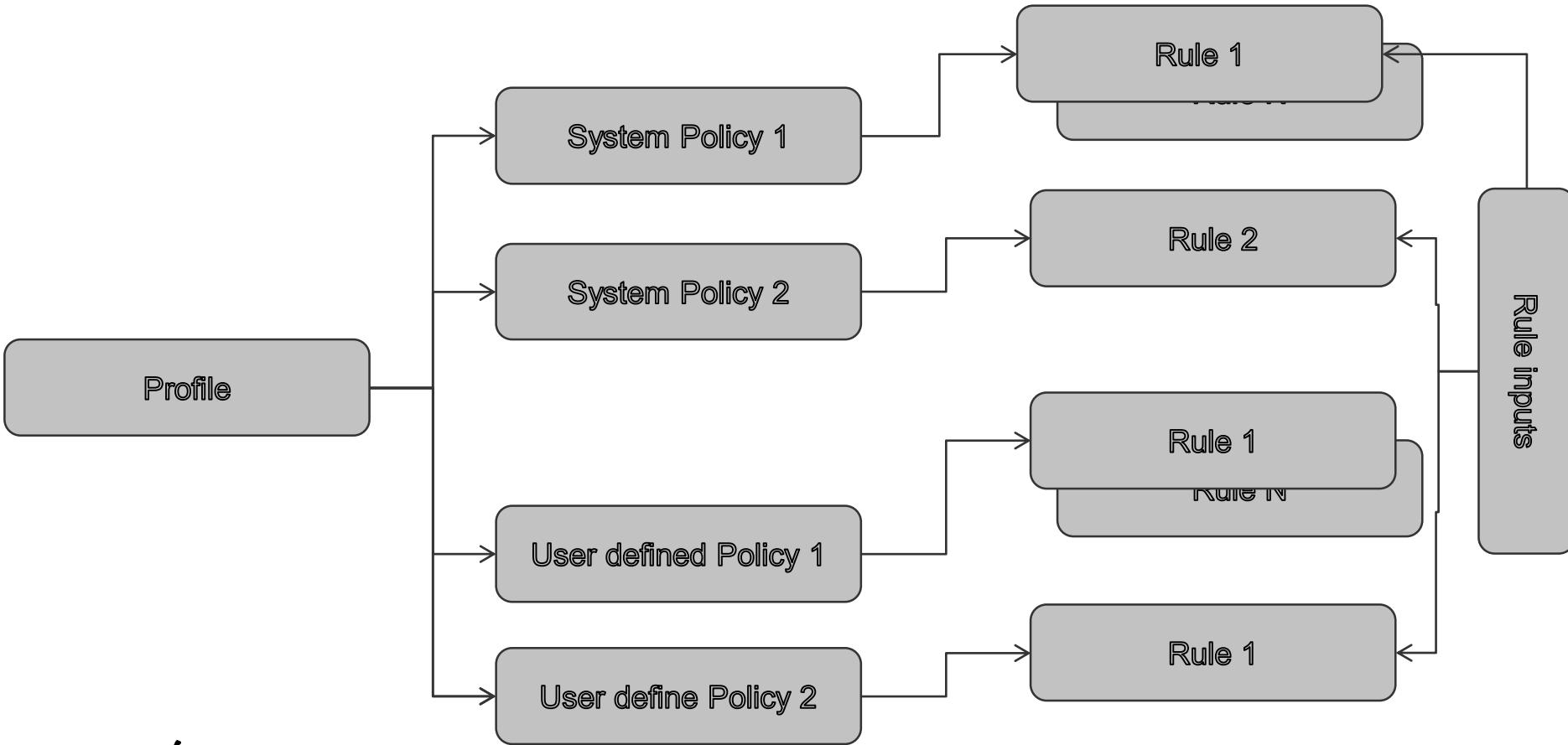
- Violation Severity:** Minor
- Violation Message Type:** User defined Violation Message
- Violation Message Id:** (empty)
- Violation Message:** Trap Destination is not configured. 'snmp-server h
- Fix CLI:** snmp-server host <_Destination> version 2c<_Community_String>

Annotations with green arrows point to specific fields:

- An arrow points from the text "User definable Severity" to the "Violation Severity" dropdown.
- An arrow points from the text "Default or User Defined Message Type" to the "Violation Message Type" dropdown.
- An arrow points from the text "User Defined Violation Message option enables additional fields:" to the "Violation Message" field.
- An arrow points from the text "Violation Message text" to the "Violation Message" field.
- An arrow points from the text "Fix CLI (optional)" to the "Fix CLI" field.

Fix CLI can be invoked from Audit Job Result (to generate Fix Job)

Profile, Policies and Rules



Compliance Jobs

Audit Jobs, Fix Jobs

Audit Jobs perform audit

The screenshot shows the Cisco Prime Infrastructure Job Dashboard. At the top, there are six status boxes: User Job Status (0 Scheduled, 36 Failed, 0 Suspended), Poller Job Status (0 Scheduled, 0 Failed, 0 Suspended), System Job Status (12 Scheduled, 2 Failed, 12 Suspended), In Progress Jobs (0 User, 4 System, 12 Poller), and My Jobs (0 Scheduled, 36 Failed, 0 Suspended). Below these are two tabs: 'Jobs' and 'User Jobs'. The 'User Jobs' tab is selected, displaying a table titled 'Compliance Jobs' with the following data:

Name	Job Type	Status	Last Run Status	Last Start...	Duration...	Next Start Time	Owner
Job_Compliance Audit Job_4_2...	Compliance Audit Job	Completed	Success	2017-02-22 1...	00:00:33		root
Job_Compliance Fix Job_4_24...	Compliance Fix Job	Completed	Success	2017-02-22 1...	00:00:05		root
Job_Compliance Audit Job_4_2...	Compliance Audit Job	Completed	Failure	2017-02-22 1...	00:00:34		root
Job_Compliance Audit Job_4_0...	Compliance Audit Job	Completed	Failure	2017-02-22 1...	00:00:34		root
Job_Compliance Audit Job_2_4...	Compliance Audit Job	Completed	Failure	2017-02-22 1...	00:00:33		root
Job_Compliance Audit Job_10_...	Compliance Audit Job	Completed	Failure	2017-02-22 1...	00:00:34		root
Job_Compliance Audit Job_10_...	Compliance Audit Job	Completed	Failure	2017-02-22 1...	00:00:33		root
Job_Compliance Audit Job_7_3...	Compliance Audit Job	Completed	Failure	2017-02-22 0...	00:00:32		root
Job_Compliance Audit Job_7_2...	Compliance Audit Job	Completed	Failure	2017-02-22 0...	00:00:36		root
Job_Compliance Fix Job_6_35...	Compliance Fix Job	Completed	Success	2017-02-21 1...	00:00:01		root

Select Profile

Select Devices

Perform Audit

Results show violations

Fix Jobs apply Fix CLI

Generate from Audit Job

Preview Fix CLI commands

Schedule Fix Job

Getting started with Compliance

Example 1: A profile with predefined policies

What did you learn in the previous example ?

- How to create a compliance profile
- How to use predefined policies in a profile
- How to send an audit job
- How to visualize the result of an audit job

Example 2

Testing SNMP community: A basic policy using string match

What did you learn in the previous example ?

- How to create a user defined compliance policy
- How to use string match in a condition
- How to generate a custom violation message and customize the severity

Example 3

Testing SNMP community: Define the community as a parameter

What did you learn in the previous example ?

- How to create a parameter (rule input) in a rule
- How to use a rule input in a condition and in a message

Example 4

Testing and fixing SNMP community A basic policy with fix CLI

Objective

- Reuse the previous policy to check another community called “dummy”
- If this community exist, apply cli to remove it

```
|SW2-acc-demo#show run | inc community  
snmp-server community private RW  
snmp-server community public RO  
snmp-server community cl17-acl RO 11  
snmp-server community cl17 ro RO  
snmp-server community dummy RO  
SW2-acc-demo#
```

What did you learn in the previous example ?

- How to clone a policy
- How to edit a policy
- How to create fix cli
- How to change the value of a parameter in a profile
- How to execute a profile on the device running configuration and what happens in reality

Example 5: Testing SNMP community and ACL: A basic policy with simple regex and back reference

```
[SW2-acc-demo#sho run | inc snmp-server
snmp-server community private RW 10 ← No violation: ACL 10 exists
snmp-server community public RO 10 ←
snmp-server community cl17-acl RO 11 ← Violation: ACL 11
snmp-server community cl17_ro RO ← violation:
snmp-server location Berlin
No ACL
SW2-acc-demo#]
```

```
SW2-acc-demo#sho run | inc access-list 10
access-list 10 permit 172.0.0.0 0.255.255.255
access-list 10 permit 192.0.0.0 0.255.255.255
access-list 10 permit 10.0.0.0 0.255.255.255
SW2-acc-demo#sho run | inc access-list 11
SW2-acc-demo#
```

What did you learn in the previous example ?

- How to use multiple conditions/actions in a rule
- How to test compliance using regex
- How to capture elements in regex and how to use them with back reference

Example 6

All interfaces “no shutdown” must have a description

Each interface block must be analyzed

```
!
interface GigabitEthernet0/0/0
| description to SP network #WAN#10M#SPP:SPP-CL17#
| ip vrf forwarding SP-demo
| ip address 10.1.1.1 255.255.255.0
| negotiation auto
| service-policy input prm-MARKING_IN
| service-policy output prm-dscp#EQ_SPP-CL17#shape#10.0
!
interface GigabitEthernet0/0/1
| ip address 172.18.253.1 255.255.255.252
| negotiation auto
| service-policy input prm-MARKING_IN
| service-policy output prm-dscp#QUEUEING_OUT
!
interface GigabitEthernet0/0/2
| ip address 172.18.254.1 255.255.255.0
| negotiation auto
| service-policy input prm-MARKING_IN
| service-policy output prm-dscp#QUEUEING_OUT
!
interface GigabitEthernet0
| vrf forwarding Mgmt-intf
| no ip address
| shutdown
| negotiation auto
```

No violation: description exists

violation

violation

No violation: interface is shutdown

What did you learn in the previous example ?

- How to parse a configuration per block

More advanced use cases

Example 7

Report Ethernet interfaces with status Admin Up / Oper Down

Policy should apply to the output of a show command

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	NVRAM	up	up
Vlan100	172.18.0.250	YES	NVRAM	up	up
GigabitEthernet0	unassigned	YES	NVRAM	down	down
GigabitEthernet1/0/1	unassigned	YES	unset	down	down
GigabitEthernet1/0/2	unassigned	YES	unset	down	down
GigabitEthernet1/0/3	unassigned	YES	unset	down	down
GigabitEthernet1/0/4	unassigned	YES	unset	administratively down	down
GigabitEthernet1/0/5	unassigned	YES	unset	administratively down	down
GigabitEthernet1/0/6	unassigned	YES	unset	administratively down	down
GigabitEthernet1/0/7	unassigned	YES	unset	administratively down	down
GigabitEthernet1/0/8	unassigned	YES	unset	administratively down	down
GigabitEthernet1/0/9	unassigned	YES	unset	down	down
GigabitEthernet1/0/10	unassigned	YES	unset	down	down
GigabitEthernet1/0/11	unassigned	YES	unset	down	down
GigabitEthernet1/0/12	unassigned	YES	unset	down	down
GigabitEthernet1/0/13	unassigned	YES	unset	up	up
GigabitEthernet1/0/14	unassigned	YES	unset	up	up
GigabitEthernet1/0/15	unassigned	YES	unset	down	down
GigabitEthernet1/0/16	unassigned	YES	unset	down	down
GigabitEthernet1/0/17	unassigned	YES	unset	down	down
GigabitEthernet1/0/18	unassigned	YES	unset	down	down
GigabitEthernet1/0/19	unassigned	YES	unset	up	up
GigabitEthernet1/0/20	unassigned	YES	unset	down	down
GigabitEthernet1/0/21	unassigned	YES	unset	down	down

What did you learn in the previous example ?

- How to analyze the output of a “show command”
- How to evaluate a regular expression in a condition

Example 8

Cleaning a configuration

Cleaning configuration

- For example, you want to verify that:
 - community cl17_ro is defined as Read Only
 - Community cl17_rw is defined as Read Write
 - No other community exists

```
[SW1-acc-demo#sho run | inc snmp-server
snmp-server community private RW
snmp-server community public RO
snmp-server community cl17 R0
snmp-server community cl17-acl RO 10
snmp-server location Paris
```

6 violations:

- 4 unexpected communities
- cl17_ro and cl17_rw missing

```
[SW2-acc-demo#sho run | inc community
snmp-server community private RW 10
snmp-server community public RO 10
snmp-server community cl17-acl RO 11
snmp-server community cl17_ro RO
snmp-server community cl17_rw RW
```

3 violations:

- 3 unexpected communities

How to do that ? 1 policy, with 2 rules and 3 conditions in each. (below the first rule)

- Define Rule Input for the expected snmp community RO <_RO>
- Condition 1: Parse the configuration
 - Match string which contains the expected community RO
 - action if no match: raise an error and fix
 - Fix : add the snmp community : snmp-server community <_RO> RO
- Condition 2: Parse the configuration as **blocks (snmp-server community as block of 1 line)**
 - Matches Read Only and capture the community name : <2.1>
- Condition 3: Evaluate the Expression :
 - Does the captured community in condition 2 matches the rule input (expected community)
 - action if no match: raise an error and fix
 - Fix: remove the non valid snmp community

What did you learn in the previous example ?

- How to audit "multiple commands" like snmp-server community, ntp server, name server, logging host
- How to create a policy with multiple rules
- How to clone a rule
- How to re-run an audit job after fixing

Example 9

Check if ports connected to Cisco IP
Phones are correctly configured

You want to check if interfaces where an IP Phone is connected have a voice vlan

```
[SW1-acc-demo#]
[SW1-acc-demo#show cdp nei
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
D - Remote, C - CVTA, M - Two-port Mac Relay
Device ID      Local Intrfce     Holdtme   Capability Platform Port ID
AP1-site1-demo Gig 1/0/13        127        T B I  AIR-CAP27 Gig 0
AP2-site1-demo Gig 1/0/14        129        T B I  AIR-CAP17 Gig 0
SW1-demo       Gig 1/0/25        162        R S T  WS-C3650- Tpe 1/1/1
SEP00115C4076C3 Gig 1/0/19        141        H P M  IP Phone Port 1
005050900040  Gig 1/0/20        100        H Cisco-VM- eth0

Total cdp entries displayed : 5
```

```
[SW1-acc-demo#sho run inter g1/0/19
Building configuration...
```

```
Current configuration : 270 bytes
!
interface GigabitEthernet1/0/19
description IP Phone
switchport access vlan 16
switchport voice vlan 18
ip device tracking maximum 10
srr-queue bandwidth share 1 30 35 5
priority-queue out
spanning-tree portfast edge
service-policy input prm-APIC_QOS_IN
end
```

```
SW1-acc-demo#
```

```
[SW2-acc-demo#show cdp nei
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
D - Remote, C - CVTA, M - Two-port Mac Relay
Device ID      Local Intrfce     Holdtme   Capability Platform Port ID
SW2-demo       Gig 1/0/25        161        R S I  WS-C3650- Gig 1/1/2
AP-site2-demo Gig 1/0/13        158        T B T  AIR-CAP26 Gig 0
SEP00115C0E5254 Gig 1/0/19        164        H P M  IP Phone Port 1
```

```
Total cdp entries displayed : 3
[SW2-acc-demo#sho run inter gig1/0/19
Building configuration...
```

```
Current configuration : 222 bytes
!
interface GigabitEthernet1/0/19
switchport access vlan 18
ip device tracking maximum 10
srr-queue bandwidth share 1 30 35 5
priority-queue out
spanning-tree portfast edge
service-policy input prm-APIC_QOS_IN
end
```

no voice vlan

How to do that ? 1 policy with 1 rule and 3 conditions

- Condition 1: Parse the output of "show cdp neighbors"
 - Matches lines which contains “IP Phone” at the appropriate expression , and capture the interface name and number (using Regex)
- Condition 2: Parse the configuration as **blocks (interface block in this case)**
 - Matches the interface name and number found in condition 1 only
- Condition 3: Parse **“inside the previously matched block”**
 - Matches the string “switchport voice vlan”

Condition 1 "show cdp neighbors"

Condition Details Action Details

Condition Scope Details

Condition Scope: Device Command Outputs
Device Property:
Show Commands: show cdp neighbors

Block Options

Parse as Blocks

Block Start Expression: ^\S\s*(Gig|Fas)\s*.*

Block End Expression:

Advanced Block Options

Condition Match Criteria

Operator: Matches the expression
Value: ^\S\s*(Gig|Fas)\s*(\S*)\s*.*IP Phone.*

Advanced Regular Expression Options Test Regular Expression

```
SW2-acc-demo#show cdp nei
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
D - Remote, C - CVTA, M - Two-port Mac Relay
Device ID      Local Intrfce     Holdtime   Capability Platform Port ID
SW2-demo       Gig 1/0/25        126        R S I WS-C3650- Gig 1/1/2
AP-site2-demo  Gig 1/0/13        154        T R T ATR-CAP36 Gig 0
SEP00115C0E5254 Gig 1/0/19        142        H P M IP Phone Port 1
```

regular expressions <https://regex101.com>

REGULAR EXPRESSION: /^\S*\s*(Gig|Fas)\s*(\S*)\s*.*IP Phone.*/

TEST STRING: SEP00115C0E5254 Gig 1/0/19 142 H P M IP Phone Port 1

EXPLANATION:

- ^ asserts position at start of the string
- \S* matches any non-whitespace character (equal to [^\r\n\t\f\v])
- Quantifier — Matches between zero and unlimited times, as many times as possible, giving back as needed (greedy)
- \s* matches any whitespace character (equal to [\r\n\t\f\v])

MATCH INFORMATION:

Match 1
Full match 0-74 'SEP00115C0E5254 Gig 1/0/19 142 H P M IP Phone Port 1'
Group 1. 17-20 'Gig'
Group 2. 21-27 '1/0/19'

Condition 2: find the appropriate interface blocks in configuration

Edit Conditions And Actions

X

Condition Details Action Details

Condition Scope Details

Condition Scope: Configuration

Device Property:

Show Commands:

Parse the configuration

Block Options

Parse as Blocks

* Block Start Expression: ^interface *

Block End Expression:

Advanced Block Options

As interface blocks

Condition Match Criteria

Operator: Matches the expression

* Value: interface (<1.1>Ethernet|<1.1>abitEthernet)<1.2>\$

Advanced Regular Expression Options Test Regular Expression

Match the interfaces previously found in “show cdp neighbors”

Condition Scope Details

Condition Scope: Configuration

Block Options

Parse as Blocks

Block Start Expression: ^interface *

Condition Match Criteria

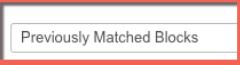
Operator: Matches the expression

Value: interface (<1.1>Ethernet|<1.1>abitEthernet)<1.2>\$

Condition 3: match “switchport voice vlan “ in the interface block

Condition Details Action Details

Condition Scope Details

Condition Scope: Previously Matched Blocks  Stay in the interface block

Device Property:

Show Commands:

Block Options

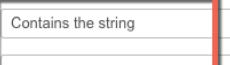
Parse as Blocks

*Block Start Expression:

Block End Expression:

Advanced Block Options

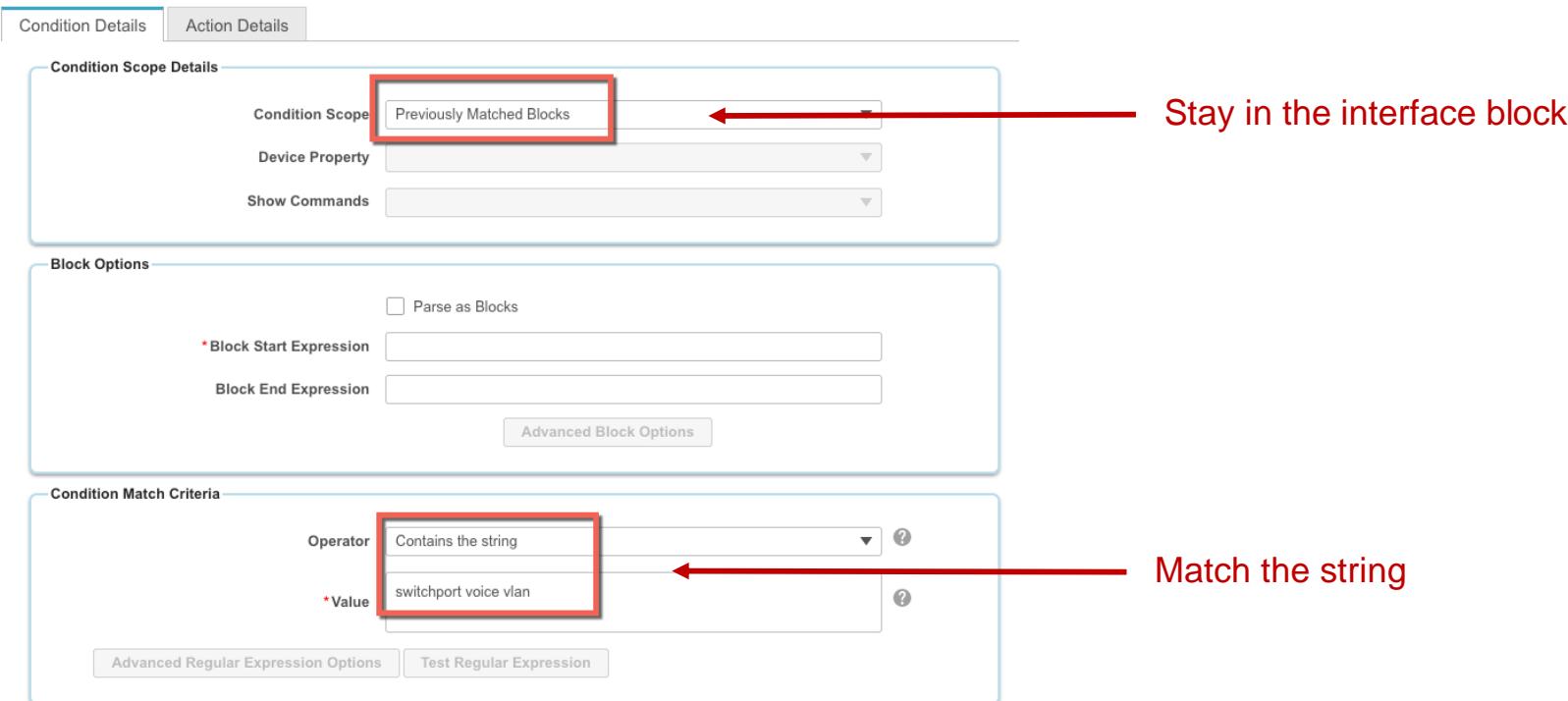
Condition Match Criteria

Operator: Contains the string 

*Value: switchport voice vlan 

Match the string

Advanced Regular Expression Options Test Regular Expression



Condition 3: match “switchport voice vlan “ in the interface block

Select Does not Match Action

Select Action	<input type="button" value="Raise a Violation"/>
Condition Number	
Violation Severity	Warning
Violation Message Type	User defined Violation Message
Violation Message Id	
*Violation Message	IP phone attached to port <2.1> <1.2> without voice vlan
Fix CLI	

Cisco Prime Infrastructure - L Cisco Prime Infrastructure - 172.16.1.1 Cisco Prime Infrastructure - 192.168.193.11 https://192.168.193.11/webacs/loginAction.do?action=login&product=wcs&selectedCategory=en#pageId=compliance_audit_policy_output_page&forceLoad=true

Non sécurisé https://192.168.193.11/webacs/loginAction.do?action=login&product=wcs&selectedCategory=en#pageId=compliance_audit_policy_output_page&forceLoad=true

Gilles

Prime Infrastructure Application Search 13 root - ROOT-DOMAIN

Compliance Policies

+ New Edit Duplicate Delete

Selected 1 / Total 1

Show All

Search All

Example - All interfaces should restrict traffic ⓘ

Example - Block incoming telnets using un-authorized protocols ⓘ

Example - Check DNS Servers are configured ⓘ

Example - NTP Server redundancy ⓘ

Example - OSPF MD5 Check ⓘ

Example - SMU verification on ASR ⓘ

Example - SNMP prohibit well known community strings ⓘ

Example - Trap Destination ⓘ

PHONES ⓘ

STM access port ⓘ

STM access STP ⓘ

STM dhcp snooping trust ⓘ

STM etherchannel ⓘ

STM HSRP ⓘ

STM no ip routing ⓘ

STM NTP ⓘ

STM Power redundancy ⓘ

STM STP root ⓘ

PHONES : Rules

Title	Description
voice vlan	ip phones must have a voice vlan

PHONES.xml Tout afficher

Other Monitoring compliance operations available Prime Infrastructure

- Monitor PSIRT
 - ensure that your network is not exposed to security threats:
- Monitor EoX
 - avoid obsolescence, plan for network upgrade:

Run Periodically EoX report

Reports / Reports / PSIRT and EOX ★

Schedule Job View Job Details

Device PSIRT Device Hardware EOX Device Software EOX Field Notice

Device HW EOX CSV Go

Device Name	Device Type	IP Address	Model Name	End of Life	End of Support	End of Sale	End of Engineering	End of Contract Renewal
Traffic-Gen-West	Cisco 2851 Integrated Services Router	10.14.89.210	CISCO2851	Yes	Oct 31, 2016	Nov 01, 2011	Oct 31, 2014	Jan 30, 2016
SW-POD99-W-Acce...	Cisco Catalyst 3560V2-24PS Switch	10.14.89.200	WS-C3560V2-24P...	Yes	May 31, 2021	May 14, 2016	May 13, 2018	Aug 12, 2020
SW-SP-West	Cisco Catalyst 3850 24P 10/100/1000 ...	10.0.255.6	WS-C3850-24P-E	No				
DC-2	Cisco Nexus 5548UP Switch	10.0.255.8	Fabric Extender M...	No				
SW-SP-East	Cisco Catalyst 3850 24P 10/100/1000 ...	10.0.255.5	WS-C3850-24P-E	No				
DC-1.prime.ciscofra...	Cisco Nexus 5548UP Switch	10.0.255.7	Fabric Extender M...	No				

Run periodically PSIRT report

Prime Infrastructure Application Search 13 root - ROOT-DOMAIN

Reports / Reports / PSIRT and EOX ★

Schedule Job View Job Details Last Run Time: Thu Dec 15 00:46:34 CET 2016

Device PSIRT Device Hardware EOX Device Software EOX Field Notice Total 46

Device PSIRT CSV Go Show All

Device Name	Device Type	IP Address	OS Type	OS Version	PSIRT Title	Based On Vers...	Based On Config
N7K-TEST	Cisco Nexus 7000 9-Slot Switch	10.0.255.11	NXOS	6.2(2)	GNU glibc gethostbyname Function Buffer Overflow Vulnerability- NX-OS (CVE...)	Vulnerable	Not Vulnerable
N7K-TEST	Cisco Nexus 7000 9-Slot Switch	10.0.255.11	NXOS	6.2(2)	GNU Bash Environment Variable Command Injection Vulnerability-NXOS	Vulnerable	Not Vulnerable
N7K	Cisco Nexus 7000 9-Slot Switch	10.0.255.14	NXOS	6.2(2)	GNU glibc gethostbyname Function Buffer Overflow Vulnerability- NX-OS (CVE...)	Vulnerable	Not Vulnerable
N7K	Cisco Nexus 7000 9-Slot Switch	10.0.255.14	NXOS	6.2(2)	GNU Bash Environment Variable Command Injection Vulnerability-NXOS	Vulnerable	Not Vulnerable
6880-VSS.prime.cisc...	Cisco Catalyst 68xx Virtual Switch	10.0.255.10	IOS	15.1(2)SY2	SSL Padding Oracle On Downgraded Legacy Encryption (POODLE) Vulnerabilit...	Vulnerable	Not Vulnerable
6880-VSS.prime.cisc...	Cisco Catalyst 68xx Virtual Switch	10.0.255.10	IOS	15.1(2)SY2	Multiple Vulnerabilities in OpenSSL Affecting Cisco Products - IOS	Vulnerable	Not Vulnerable
6880-VSS.prime.cisc...	Cisco Catalyst 68xx Virtual Switch	10.0.255.10	IOS	15.1(2)SY2	Multiple Vulnerabilities in OpenSSL (March 2015) Affecting Cisco Products - IOS	Vulnerable	Not Vulnerable
6880-VSS.prime.cisc...	Cisco Catalyst 68xx Virtual Switch	10.0.255.10	IOS	15.1(2)SY2	Cisco IOS Software and IOS XE Software TCP Packet Memory Leak Vulnerabilit...	Vulnerable	Not Vulnerable
N7K-CORE1	Cisco Nexus 7000 9-Slot Switch	10.0.255.1	NXOS	6.2(2)	GNU glibc gethostbyname Function Buffer Overflow Vulnerability- NX-OS (CVE...)	Vulnerable	Not Vulnerable
N7K-CORE1	Cisco Nexus 7000 9-Slot Switch	10.0.255.1	NXOS	6.2(2)	GNU Bash Environment Variable Command Injection Vulnerability-NXOS	Vulnerable	Not Vulnerable
N7K-CORE2	Cisco Nexus 7000 9-Slot Switch	10.0.255.2	NXOS	6.2(2)	GNU glibc gethostbyname Function Buffer Overflow Vulnerability- NX-OS (CVE...)	Vulnerable	Not Vulnerable
N7K-CORE2	Cisco Nexus 7000 9-Slot Switch	10.0.255.2	NXOS	6.2(2)	GNU Bash Environment Variable Command Injection Vulnerability-NXOS	Vulnerable	Not Vulnerable
RTR-POD6.prime.cisc...	Cisco 4331 Integrated Services Router	10.255.0.136	IOS-XE	15.5(3)S3	No	Not Vulnerable	Not Vulnerable
RTR-POD7.prime.cisc...	Cisco 4331 Integrated Services Router	10.255.0.137	IOS-XE	15.5(3)S3	No	Not Vulnerable	Not Vulnerable

Run periodically PSIRT report

Prime Infrastructure

Reports / Reports / PSIRT and EOX

Schedule Job View Job Details

Last Run Time: Thu Dec 15 00:46:34 CET 2016

Device PSIRT Device Hardware EOX Device Software EOX Field Notice

Device PSIRT CSV Go Show All Total 46

PSIRT Title	Based On Version	Based On Configuration
GNU glibc gethostbyname Function Buffer Overflow Vulnerability- NX-OS (CVE-2016-1010)	Vulnerable	Not Vulnerable
GNU Bash Environment Variable Command Injection Vulnerability-NXOS	Vulnerable	Not Vulnerable
GNU glibc gethostbyname Function Buffer Overflow Vulnerability- NX-OS (CVE-2016-1010)	Vulnerable	Not Vulnerable

RTR-POD7.prime.cis... Cisco 4331 Integrated Services Router 10.255.0.137 IOS-XE 15.5(3)S3 No Not Vulnerable Not Vulnerable

Conclusion/Take away

Conclusion/Takeway

- Powerful compliance engine
- 100+ pre-defined policies
- Supports IOS, IOS-XE, IOS-XR, NX-OS, **AireOS***, NX-OS
- One policy can have rules for different OS
- Works on **Standard OVA****, PRO OVA and GEN2 appliance
- Policy can be exported/imported (XML format)
- Available through **API***
 - Since 3.1
 - Since 3.1 MR4

Complete Your Online Session Evaluation

- Please complete your Online Session Evaluations after each session
- Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt
- All surveys can be completed via the Cisco Live Mobile App or the Communication Stations



Don't forget: Cisco Live sessions will be available for viewing on-demand after the event at
CiscoLive.com/Online

Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Lunch & Learn
- Meet the Engineer 1:1 meetings
- Related sessions

Q & A

Thank You



February 20 - 24, 2017 • Berlin



Your Time Is Now