**Broadcom Corporation Technical Publication**

SONiC SQA Test Plan

Manageability Framework

Arlo(SONiC 2.0)

Document Status: Draft

Document Issue: 1.0  
Issue Date: 3/05/2019  
Security Status: Broadcom Confidential  
Author: Karthik Kumar Goud Battula

© 2011 Broadcom Corporation  
All rights reserved.

UNCONTROLLED COPY: The master of this document is stored on an electronic database and is “write protected”; it may be altered only by authorized persons. While copies may be printed, it is not recommended. Viewing of the master electronically ensures access to the current issue. Any hardcopies taken must be regarded as uncontrolled copies.

CONFIDENTIAL: The information contained in this document is the property of Broadcom Corporation Except as expressly authorized in writing by Broadcom Corporation. The holder shall keep all information contained herein confidential, shall not disclose to anyone outside Broadcom Corporation, including customers and partners, regardless of any non-disclosure agreement (NDA) that maybe in place, without prior written consent from the Quality Assurance Manager or Author. In the absence of written consent, the holder shall only disclose the information herein to authorize Broadcom Corporation employees with a need to know basis, and shall protect the information from disclosure and dissemination to third parties. Except as expressly authorized in writing by Broadcom Corporation, the holder is granted no rights to use the information contained herein.

Broadcom Corporation   
{enter site address of Author}

1030 Swabia Court

Suite 400

Durham, NC 27703

[www.broadcom.com](http://www.broadcom.com)

Broadcom®, the pulse logo, Connecting everything®, and the Connecting everything logo are among the trademarks of Broadcom Corporation and/or its affiliates in the United States, certain other countries and/or the EU. Any other trademarks or trade names mentioned are the property of their respective owners.



Proprietary and Confidential Information

# Test Strategy Plan Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Author** | **Change Description** |
| 0.01 |  |  | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# List of Approvers

This section identifies the participants required to approve this document. Enter function title, name and once approved at the meeting or by email trail, update the date field with the date approved.

|  |  |  |
| --- | --- | --- |
| Function | Name | Date Approved |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# List of Reviewers

Section identifies optional reviewers.

|  |  |
| --- | --- |
| Function | Name |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Abbreviations and Glossary

Add Abbreviations and Glossary terms here.

This section describes any words that the reader may not be familiar with or abbreviations used that the reader may not be familiar with. Do not include definitions or abbreviations that should be generally known by your audience.

**AT** Acceptance Testing

**CLI** Command Line Interface

**DOR** Dead Office Recovery

**FCT** Feature Confidence Test

**FT** Functional Testing

**GUI** Graphical User Interface

**IEEE** Institute of Electrical and Electronics Engineers

**IT** Integration test or Component Integration testing (CIT)

**IP**  Internet Protocol (implies version 4)

**IPv4** Internet Protocol version 4

**IPv6**  Internet Protocol version 6

**MIB**  Management Information Base

**NMS** Network Management System

**QA** Quality Assurance

RFT Regression Functional Test

**RIT** Release Integration Testing

**RoR** Release over Release

**SCT** System Confidence Test

**ST** System Test

**TA** Test Automation

TAM Test Automation Manager

**TFP** Test First Pass

**UI** User Interface

**UT** Unit Testing

**VLAN** Virtual Local Area Network

Add additional Abbreviations as it applies to this project.

**Table of Contents**

[MS Word Settings for Using this Template ii](#_Toc505181414)

[Test Strategy Plan Revision History 2](#_Toc505181415)

[List of Approvers 2](#_Toc505181416)

[List of Reviewers 2](#_Toc505181417)

[Abbreviations and Glossary 3](#_Toc505181418)

[*Page Intentionally Left Blank* 6](#_Toc505181419)

[1. Functional Project Overview 7](#_Toc505181420)

[1.1 Requirements Source and Test Traceability 7](#_Toc505181421)

[1.1.1 Source Document 7](#_Toc505181422)

[1.1.2 Test Traceability 7](#_Toc505181423)

[1.2 Results and Document Tracking Location 8](#_Toc505181424)

[1.3 Planned QA test cycles 8](#_Toc505181425)

[1.4 Test case distribution 8](#_Toc505181426)

[1.5 Feature test project plans 8](#_Toc505181427)

[1.5.1 Test Equipment Requirements 9](#_Toc505181428)

[1.5.2 Other Feature or co-feature dependencies 10](#_Toc505181429)

[1.6 Software Dependencies 10](#_Toc505181430)

[2. Test Focus Areas 11](#_Toc505181431)

[2.1 Configuration and Provisioning 11](#_Toc505181432)

[2.1.1 Scope of Testing 11](#_Toc505181433)

[2.1.2 Network Topology Design 11](#_Toc505181434)

[2.1.3 Limitations and Exceptions 11](#_Toc505181435)

[3. Test Case and Objectives 11](#_Toc505181436)

[3.1 <Functional Tests > 12](#_Toc505181437)

[3.1.1 L3VNI 12](#_Toc505181438)

[3.2 <FIT Tests > 13](#_Toc505181439)

[3.2.1 <Test Area > 13](#_Toc505181440)

[3.3 < CLI/RestAPI/SNMP Tests> 13](#_Toc505181441)

[3.3.1 <Test Area > 13](#_Toc505181442)

[3.4 < 1D Scaling Tests> 13](#_Toc505181443)

[3.4.1 <Test Area > 13](#_Toc505181444)

[3.5 < Negative Tests> 13](#_Toc505181445)

[3.5.1 <Test Area > 13](#_Toc505181446)

[3.6 < Longevity/Stress Tests> 14](#_Toc505181447)

[3.6.1 <Test Area > 14](#_Toc505181448)

[4. References 14](#_Toc505181449)

List of Tables

[Table 1 – Test Traceability Reference Table 10](#_Toc318118011)

[Table 2 – Test Case Development Plans 12](#_Toc318118012)

[Table 3 – Required DuT Equipment 13](#_Toc318118013)

[Table 4 – test Gear Requirements 14](#_Toc318118014)

# *Page Intentionally Left Blank*

# Functional Project Overview

This project…

* Provide a brief overview of the project. A cut-and-paste from the end-to-end Functional Design or Specification is acceptable. Also, add any references as needed regarding approved Requirements Specification in the References section..

## Requirements Source and Test Traceability

### Source Document

Provide URL location to FS Spec or document used here:

* Identify the main requirements tracking document (FS Spec or other documents that define actual requirements).
* This should typically be a Functional Requirements Specification. Supply a link to location of the document or list requirements records/table here

### Test Traceability

Provide URL to Excel worksheet, or fill out table below if plan only has a low number of requirements.

Table – Test Traceability Reference Table

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Test Case IDs | Automated  (Yes/No) | Script Details |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
|  |  |  |  |

* All test cases must trace back to a FS spec requirement or other approved document (including White papers, PRD,s or MRDs).

## Results and Document Tracking Location

All test cases and execution results will be stored in TAM. Test Cases for this functional plan are located in TestCase Manager at:

Include TAM folder paths and document locations for plans and test cases

**Test Selector Folder Name/Path:**

URL Location of this test plan:

* Identify at a high-level strategy what will be followed for verifying this feature. Items that should be included are:
  + Identification of test objectives and goals for this feature.
  + Identification of the different forms of testing that are needed for this feature, based on a detailed analysis of its contents.
    - Some examples:. Will interoperability and interworking testing be required?(think about 3rd party equipment and/or application?) If new hardware is part of project, will installation and commissioning documents be needed and who will verify them? Is there any hardware-specific testing required? Will error path and negative scenarios such as fault insertion testing be required? For each form of testing, identify the driving force that requires this testing to be done.
  + Define or identify any end to end solutions (for network based or function based product/application),i.e., to do POC (prove of concept) testing?
  + Identification of the testing groups responsible for doing the above testing. This would be a list of the groups or members that need to be involved. Describe what their involvement will be.

## Planned QA test cycles

CFT Planned Start Date:

RFT Planned Start Date:

* Note the planned QA cycles this feature will be validated in. In cases where the feature is split between cycles, add the additional cycles above. Make sure strategy section and CFT test cases identify each cycle as needed.

## Test case distribution

Note here the percentage (%) planned coverage by software product and platform. Note automation coverage, including % of coverage in section 4. Note lead platforms for each product below the table. Total % may be greater than 100% if we plan to retest specific test cases on multiple platforms.

Table – Test Case distribution

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | <pltfrm> | <pltfrm> | <pltfrm> | <pltfrm> | <pltfrm> | <pltfrm> | <pltfrm> | <pltfrm> |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

## Feature test project plans

* Insert references to the test document/project plans. Links are acceptable. Goal of this section is to identify when certain documentation activities will be planned. This document notes plans on design test activity and provides details on functional strategies, test case writing and execution. In most cases, the strategy should be defined and set to a small group of leads for comments. Once test cases are defined and strategy is updated, the document is sent out for final approval.
* Readers should be able to read this section and understand when test plan activities will occur and whom to contact for each test activity.

The following project plan covers the test activities for this feature:

Table – Test Case Development Plans

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Design CIT/RIT | Prime Name | Effort | Start | End |
| * TC Writing |  |  |  |  |
| * Reviewing |  |  |  |  |
| * Approved |  |  |  |  |
| * TC execution |  |  |  |  |
| * Complete |  |  |  |  |
| Feature Plan |  |  |  |  |
| * Plan writing |  |  |  |  |
| * Reviewing |  |  |  |  |
| * Approved |  |  |  |  |
| Feature Tests |  |  |  |  |
| * TC writing |  |  |  |  |
| * Reviewing |  |  |  |  |
| * Approved |  |  |  |  |
| * TC execution |  |  |  |  |
| * Complete |  |  |  |  |

## 

* Identify any special cable needs and quantity here. If standard fiber or Cat5 cable is required and is normally available in lab, just note no special requirements.

### Test Equipment Requirements

Table – test Gear Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Gear Type | Test Gear ports/Module Names | Model Number | HW Revision | Quantity |
| * Ixia 10G ports | * 8 Port Ixia module | * xxxxxx | * Rev x | * Nn |
| * Spirent Avalanche | * Avalanche Client, 2 ports | * Xxxxxx | * Rev xx | * Nn |

* Identify all test equipment requirements, the minimum port types version or revision required, and any test applications required.

#### Total test gear ports needed by type

* Identify any special port types, media types or speeds here. If a specific I/O module or port type is required, clearly note the requirement here.

#### Total port Optical interfaces/SFP requirements by type

* Identify all optical and pluggable copper interfaces required for each test gear port that will be used for feature validation. List all modules by type and quantity. If specific manufacture is required, list manufactures and model numbers here. Note only the Test Gear interfaces here. Test DuT interfaces are noted earlier in this section.

#### Other ports or services needed by type

* Identify any special interfaces such as data center NICs (CNA) or other traffic generation or services devices as needed.

### Other Feature or co-feature dependencies

* Identify all mandatory and co-feature dependencies required to validate the feature. Example: Data Center ETS feature requires DCBX, FIP Snooping required PFC, etc. A table list format is best for describing this requirement. Include features or dependencies that are not considered mandatory for successfully validation. A table list format is best for describing this requirement.

## Software Dependencies

* Identify all software requirements associated with each stage of testing. Note any external PC based, server based or test tool based software applications here.
* If any 3rd party export controlled software is in use, note it here and the export restrictions.
* If a requirement is unique to a particular stage then highlight that fact in your list

# Test Focus Areas

The following sections identify general functional test areas to be focused on. Focus areas are broad enough in scope that it should not be necessary to define additional areas.

The following sections should be filled out from a “STRATEGY” perspective. This means the test lead evaluates each section below and determines what they “feel” needs to be done to achieve the associated test discipline and deliver a quality product to the field. These sections are not intended to identify or define actual functional test steps or the actual test case. A strategy defines what to do, when it should be done, how it is done, and in some cases who should do it. Design and Test leads should collaborate to identify actual test needs to help define the initial strategy.

When filling in “Scope of Testing”, categorizing the feature items (defining “sub-areas”) is helpful in adding structure to the sections. When determining areas of verification and their associated “who/when” information, there should be an indication of the reasoning behind the decisions. This should be specified in the “Scope Of Testing” sections below.

Make sure there is enough detail in each of the sections to allow reviewers to understand the overall objectives, and especially if other members are writing test cases for this feature. It is important the success criteria/objective be clearly defined for each sub category. If the detail is, too high-level it will be difficult for reviewers and test cases writers to clearly understand the overall test intentions. The strategy and scope is an input to the actual tasks and test objectives so reviewers do not have to review actual test cases, and to help guide the test writers with the actual test step requirements to document.

## Configuration and Provisioning

### Scope of Testing

* Identify the overall configuration that this feature needs to be tested against. What is the purpose of the test, what are the goals and what conditions must be met before stating the feature meets all applicable release requirements.
* Define the strategy for verifying across the range of configurations to be defined to ensure the feature will work across all deployed configurations. In this section, note what nodes will be explicitly validated during the component functional test phase, and what will QA regress across other variants since code is normally considered generic.

### Network Topology Design

Server==>DUT==>client

* Provide the network topology design that will be used to validate this feature. Include all planned hardware configurations (to justify the number of required nodes), any partner type devices and any interop devices if applicable. Show were test points are and the test equipment that will be connected at these various test points. Be sure to identify any special media, ports, or services type connections. Clearly label LAGs, ECMP links and any other special port conditions applicable to the feature test objectives.
* It is not prudent to test every feature combination against every possible port or module configuration. Analysis needs to be done as to what configurations are most important to test against and why (e.g., customer configuration or complexity). All Network nodes affected by the code updates need to be considered in the topology design.

### Limitations and Exceptions

* What is the verification limitations associated with this testing? Is there anything that we are not able to do due to limitations that we currently have. Are there any mitigations to these limitations (if any)?

# Test Case and Objectives

* If using separate Test Case plan template, note Location of template and plans here
* The section below describes all test cases for each stage of testing and are stored in the Test Automation Manager (TAM) database. Every case must reference associated design activities/requirements, and test activities. A Test Objective is simply a preliminary view of the actual case. It should not have all the details of the test steps, but should contain requirement statements and pass criteria statement. **It is strongly suggested that test objectives be written first before writing the test step procedures and associated expectations.**

The following section describes each test case in detail. The top section header “X.X” defines a major test section, or may include a test focus area name. Every test should have a test case ID that follows the approved naming conventions for TAM, but a temporary unique ID number scheme may be used as needed

All test cases for the defined Test/Functional area must include all sections identified below the Test Case ID sub header “X.X.X”. Copy and paste this section as needed for each test case. Add new test/functional section headers when test focus area changes for the feature. All test cases require a Functional Spec or Test Plan reference that can be used as a cross reference to insure code coverage.

## <Functional Tests >

### TACACS+

#### Verify the functionality of failthrough mechanism by changing the login authentication order through REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify the functionality of failthrough mechanism by changing the login authentication order using REST | |
| Description | functionality of failthrough mechanism by changing the login authentication order using REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| Setup:  =========   1. Configure TACACS+ server with valid user credentials.   Procedure:  1. Perform the PUT operation to configure login authentication as local,tacacs+.  2. Perform the POST operation to Configure the IP address for the TACACS+ Server in the device under test with default configurations for timeout priority and L4 port.  3. Perform PUT operation to disable authentication failthrough.  4. Perform the GET operation to check the configurations are applied successfully or not.  5. Now do SSH/console login to DUT . try with user id/ password which is configured locally and tacacs+ server as well.  6. Perform the POST operation to configure login authentication as tacacs+,local. Do repeat the step-5.    Expected Result:  1. Verify the aaa authentication mode configuration is successful.  2. Verify that POST operation is successfull and the server address is configured and that the default parameters for the port timeout and priority are presented when viewing TACACS+ Server configurations.  3. Verify that PUT operation is successfull and failthrough authentication is disabled.  4. Verify that GET operation is successfull and details shown are as we configured.  5. Verify that client login successful via local user credentail.  6. Verify configuration is successful using POST operation.  Verify client login successful via tacacs+ server only and not locally. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify the default login via tacacs is successful using REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify default login via tacacs is successful using REST | |
| Description | Verify default login via tacacs is successful using REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| Setup:  =========  1) Configure TACACS+ server with valid user credentials.  Procedure:  1. Perform PUT Operation to configure AAA authentication login tacacs+  2. Perform the POST operation to configure the IP address L4 port timeout and priority for the TACACS+ Server in the device.  3. Perform GET operation to see the configations are applied successfully.  4. Attempt to open an SSH into the device and login using a credential set pointed at the TACACS+ Server for authentication.  5. Now, logout and login with invalid credentials or non-tacacs+ credentials.  Expected Result:  1. Verify the user and authentication mode configuration is successful or not.  2. Verify POST operation is successfull and the server address L4 port timeout and priority are configured by viewing TACACS+ Server configurations.  3. Verify that GET operation is successfull and details shown are same as we configured.  4. Verify that a login into the device under test successful via tacacs server only .  5. verify client authentication fails when tried to login via locally configured user credential. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify the functionality of failthrough mechanism when DUT have multiple server with default priority using REST.

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify the functionality of failthrough mechanism when DUT have multiple server with default priority using REST. | |
| Description | Functionality of failthrough mechanism when DUT have multiple server with default priority using REST. | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| PreSetup:  =========  1) Configure TACACS+ server with valid user credentials.  Procedure:  1. Perform the POST operation to configure login authentication as tacacs+,local.  2. Perform the POST operation to Configure the IP address for the TACACS+ Server in the device under test with default configurations for timeout priority and L4 port.  Make sure DUT have multiple TACACS+ Server configured  3. Perform the PUT operation to Enable authentication failthrough.  4. Perform the GET operation to check whether the configurations are applied successfull or not.  5. Now do SSH/console login to DUT . Try with user id/ password which is configured locally and tacacs+ server as well.  6. Disable authentication failthrough using POST operation. Try the step-5 again.  Expected Result:  1. Verify the aaa authentication mode configuration is successful.  2. Verify that POST operation is successfull and the server address is configured and that the default parameters for the port timeout and priority are presented when viewing TACACS+ Server configurations.  3. Verify failthrough is Enabled using POST operation.  4. Verify GET operation is successful and details are shown as we configured.  5. Verify that login to DUT is successful with both locally configured user and tacacs+ server user.  6. Verify that login to DUT should successful via TACACS+ server configured user only. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify that by default, no TACACS servers exist and that a single TACACS server can be added and used for authentication using REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify that by default, no TACACS servers exist and that a single TACACS server can be added and used for authentication using REST | |
| Description | By default, no TACACS servers exist and that a single TACACS server can be added and used for authentication using REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| PreSetup:  =========   1. Configure TACACS+ server with valid user credentials.   Procedure:  1. Performe GET operation to verify that no TACACS server is available by default  2. Perform POST operation to configure aaa authentication login as tacacs+  3. Perform POST configuration to configure the IP address for the TACACS+ Server in the device with default configurations for timeout, priority, and L4 port.  4. Perform the GET operation to check the configuration that are configured in step 2&3.  5. SSH to the device using a credential set pointed at the TACACS+ Server for authentication.  Expected Result:  1. Verify that the default configurations show no server IP address defined.  2. Verify that aaa authentication mode configuration is successful.  3. Verify that the server address is configured and also default parameters for the port, timeout, and priority  4. Verify that GET operation is successful and details are presented when viewing TACACS+ Server configuration.  5. Verify that a login into the device can be accomplished when authenticated against a TACACS+ Server. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify that user is able to execute the “TACACS-server key command successfully in appropriate mode via REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify that user is able to execute the “TACACS-server key command successfully in appropriate mode via REST | |
| Description | User is able to execute the “TACACS-server key command successfully in appropriate mode via REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| PreSetup:  =========   1. Configure TACACS+ server with valid user credentials.   Procedure:  ============  1. Bring up the DUT with default configuration.  2. Perform the POST operation to config “TACACS-server key" to eanble key TACPLUS global mode.  3. Perform the POST operation to config “TACACS-server key" to eanble key TACPLUS\_SERVER mode.  4. Perform the POST operation to modify the Key in TACPLUS global/ TACPLUS\_SERVER mode  5. Perform the DELETE operation to delete the added tacacs server.  6. Perform the GET operation to check the configurations that are configured in above steps.    Expected Results:  ================  1. Verify that DUT should come up with default configuration.  2. Verify that POST operation is successfully and TACACS server key is configured  3. verify the result same like above.  4. Verify that admin able to modify the key and key is showing properly using GET operation.  5. Verify that admin able to delete the server and all parameters under the server is deleted ( ex:- passkey, priority, port id etc).  6. Verify that GET operation is successfull and the details shown are correct as per our configuration. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify that Taccacs server key can be configured with more that 4 special characters using REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify that Taccacs server key can be configured with more that 4 special characters using REST | |
| Description | Taccacs server key can be configured with more that 4 special characters using REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| steps:  1) Bring up the DUT with default configuration.  2)Perform the POST operation to configure taccacs key with more that 4 special characters like "%"    Expected Results:  1)DUT is up and running.  2)Verify that password is configured more that 4 specila characters and woring fine using GET. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify that Maximum number of TACACS IPv4 servers can be configured on DUT using REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify that Maximum number of TACACS IPv4 servers can be configured on DUT using REST | |
| Description | Maximum number of TACACS IPv4 servers can be configured on DUT using REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| Procedure:  1.Bring up the DUT with factory defaults.  2.Perform the POST operation to configure maximum tacacs Ipv4 servers and try to add one more that max limit.  3.Perform the GET operation if maximum number of servers are created or not.  Expected Result:  1.Verify that DUT is Up with default config.  2.Verify that DUT allows user to config maximum limit only and throws error when try to configure more than max limit.  3.Verify that GET operation is successfull and shown the maximum number of servers created. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify that Maximum number of TACACS IPv6 servers can be configured on DUT using REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify that Maximum number of TACACS IPv6 servers can be configured on DUT using REST | |
| Description | Maximum number of TACACS IPv6 servers can be configured on DUT using REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| Procedure:  1.Bring up the DUT with factory defaults.  2.Perform the POST operation to configure maximum tacacs Ipv6 servers and try to add one more that max limit.  3.Perform the GET operation if maximum number of servers are created or not.  Expected Result:  1.Verify that DUT is Up with default config.  2.Verify that DUT allows user to config maximum limit only and throws error when try to configure more than max limit.  3.Verify that GET operation is successfull and shown the maximum number of servers created. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify the functionality of failthrough mechanism by changing the login authentication order using REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | Verify the functionality of failthrough mechanism by changing the login authentication order using REST | |
| Description | The functionality of failthrough mechanism by changing the login authentication order using REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| PreSetup:  =========   1. Configure TACACS+ server with valid user credentials.   Procedure:  1. Perform the POST operation to configure login authentication as local,tacacs+.  2. Perform the POST operation to configure the IP address for the TACACS+ Server in the device with default configurations for timeout priority and L4 port.  3. Perform the PUT operation to disable authentication failthrough.  4. Perform the GET operation to check the whether the configurations are applied or not.  5. Now do SSH/console login to DUT . try with user id/ password which is configured locally and tacacs+ server as well.  6. set the configure login authentication as tacacs+,local. Do repeat the step-5.    Expected Result:  1. Verify that aaa login authentication as local, tacacs+ successful.  2. Verify that the server address is configured and that the default parameters for the port timeout and priority are presented when viewing TACACS+ Server configurations.  3. Verify config successful.  4. Verify that GET operation is successful and configuration shown are correct.  5. Verify that client login successful via local user credentail.  6. Verify configuration is successful.  Verify client login successful via tacacs+ server only not locally. | | |
| Pass/Fail Criteria | | Above verifications should PASS |

#### Verify if DUT is configured with more than one tacacs server with priority configured client can login to dut via heightest priority tacacs server using REST

|  |  |  |
| --- | --- | --- |
| Test Case Details | | |
| Title / Test Name | verify if DUT is configured with more than one tacacs server with priority configured client can login to dut via heightest priority tacacs server using REST | |
| Description | if DUT is configured with more than one tacacs server with priority configured client can login to dut via heightest priority tacacs server using REST | |
| Test Setup | Topology 1 | |
| Manual Execution | Yes | |
| Automation Status | Yes | |
| Automation Priority | High | |
| Interface Mode | 1G, 10G, 40G, 100G, LAG | |
| Basic feature Sanity | Functional | |
| New in Release |  | |
| Platform Dependent | No | |
| PreSetup:  =========  1) Configure TACACS+ server with valid user credentials.  Procedure  ============  1. Perform the POST operation to configure the IP address for the TACACS+ Server in the device with priority configurations for timeout priority and L4 port.  Make sure DUT have multiple TACACS+ Server configured. one server have heighest priority configured.  3. Perform the PUT operation configure login authentication as tacacs+.  3. Perform the GET operation to check the configurations applied or not  4. login to DUT via SSH .  5. Perform the DELETE operation to delete the tacacs server which has heighest priority configured.  6. login to DUT via SSH.  Expected Result:  1,2,3. Verify the config is successful via GET  4. Verify client login successful via highest configured tacacs server.  5 Verify config is successful via GET operation.  6. Verify that clients auth success via another TACACS server (server who has 2nd heighest priority configured and user id /passwword aslo pereseent is server data base ) | | |
| Pass/Fail Criteria | | Above verifications should PASS |

## <FIT Tests >

### <Test Area >

#### <Test Case ID>

## < CLI/RestAPI/SNMP Tests>

### <Test Area >

#### <Test Case ID>

## < 1D Scaling Tests>

### <Test Area >

#### <Test Case ID>

## < Negative Tests>

### <Test Area >

#### <Test Case ID>

## < Longevity/Stress Tests>

### <Test Area >

#### <Test Case ID>

# References

* References are added as needed. Include the URL location and the document version as needed. This section is appropriate to reference materials such as relevant requirements; high level or detailed design documents, significant industry standards, testing methodology, etc..