# TACACS+ per-command authorization&accounting Test Plan

# Table of Contents

* [Table of Contents](#table-of-contents)
* [Overview](#overview)
* [1 Test per-command authorization accept/deny user command](#test-per-command-authorization-accept-d)
  + [1.1 Test per-command authorization set to 'tacacs+'](#11-test-per-command-authorization-set-t)
  + [1.2 Test per-command authorization set to 'tacacs+ local'](#12-test-per-command-authorization-set-t)
  + [1.3 Test per-command authorization set to 'local'](#13-test-per-command-authorization-set-t)
* [2 Test per-command authorization when tacacs server accessible or not](#test-per-command-authorization-when-tac)
  + [2.1 Test per-command authorization set to 'tacacs+'](#21-test-per-command-authorization-set-t)
  + [2.2 Test per-command authorization set to 'tacacs+ local'](#22-test-per-command-authorization-set-t)
* [3 Test per-command accounting](#test-per-command-accounting)
  + [3.1 Test per-command accounting set to 'tacacs+'](#31-test-per-command-accounting-set-to-t)
  + [3.2 Test per-command accounting set to 'tacacs+ local'](#32-test-per-command-accounting-set-to-t)
  + [3.3 Test per-command accounting set to 'local'](#33-test-per-command-accounting-set-to-l)
* [4 References](#references)

# Overview

The purpose of this test plan is to validate the SONiC per-command authorization/accounting feature can match the Azure security requirement.

The test cases can be organized to 3 groups:

* Test per-command authorization accept/deny user command:
  + User will login to test device as GME domain user.
  + The test tacacs server-side will setup whitelist to allow user run 'show' command and deny 'config' command.
  + In prod environment tacacs server, the GME RO and RW user will have different command whitelist config, which not cover by these test cases, there will be another configuration proposal and test plan cover the prod environment tacacs setting.

Following table show the test case matrix:

column header: tacacs authorization setting

row header: domain user operation

cell: expected result.

|  | **‘tacacs+'** | **'tacacs+ local'** | **'local'** |
| --- | --- | --- | --- |
| Run command in whitelist | Command success | Command success | Command success |
| Run command not in whitelist | Command denied | Command denied | Command denied when user does not have local permission. |

* Test per-command authorization when tacacs server accessible or not:
  + According to Azure security requirement, when with tacacs per-command authorization enabled, domain user can't run any command when tacacs server is not accessible.
  + These test cases will check if the tacacs feature can match the Azure security requirements.

Following table show the test case matrix:

column header: tacacs authorization setting

row header: tacacs server status

cell: expected result.

|  | **‘tacacs+'** | **'tacacs+ local'** | **'local'** |
| --- | --- | --- | --- |
| Tacacs server accessible | Command success | Command success | N/A |
| Tacacs server not accessible | Can't run any command | Can't run any command | N/A |

* Test per-command accounting:
  + These test cases will check if accounting log can send to tacacs server correctly.
  + For security reason, sonic will replace password with '\*' from accounting log, these test cases also will check if the password been replaced correctly.

Following table show the test case matrix:

column header: tacacs accounting setting

row header: domain user operation

cell: expected result.

|  | **‘tacacs+'** | **'tacacs+ local'** | **'local'** |
| --- | --- | --- | --- |
| Run command | Accounting logs send to server-side log | Accounting logs exist in both server-side log and local log | Accounting logs exist in local log |
| Run command with password | Password been replaced with \* in server-side log | Password been replaced with \* in both server-side log and local log | Password been replaced with \* in local log |

# 1 Test per-command authorization accept/deny user command

## 1.1 Test per-command authorization set to 'tacacs+'

Prepare steps:

Set device per-command authorization to 'tacacs+' with following command:  
  sudo config aaa authorization tacacs+

### 1.1.1 Test domain user can run command in tacacs server whitelist

* Test domain user can run command in tacacs server whitelist. Steps:

Domain user login to device.  
Run following command: 'show aaa'

Expected result:

Command success, and output AAA setting in console.

### 1.1.2 Test domain user can't run command not in tacacs server whitelist

* Test domain user can't run command not in tacacs server whitelist. Steps:

Domain user login to device.  
Run following command: 'config aaa'

Expected result:

Command denied with following message:  
  '/usr/bin/config authorize failed by TACACS+ with given arguments, not executing'

## 1.2 Test per-command authorization set to 'tacacs+ local'

Prepare steps:

Set device per-command authorization to 'tacacs+ local' with following command:  
  sudo config aaa authorization 'tacacs+ local'

### 1.2.1 Test domain user can run command in tacacs server whitelist

* Test domain user can run command in tacacs server whitelist. Steps:

Domain user login to device.  
Run following command: 'show aaa'

Expected result:

Command success, and output AAA setting in console.

### 1.2.2 Test domain user can't run command not in tacacs server whitelist

* Test domain user can't run command not in tacacs server whitelist. Steps:

Domain user login to device.  
Run following command: 'config aaa'

Expected result:

Command denied with following message:  
  '/usr/bin/config authorize failed by TACACS+ with given arguments, not executing'

## 1.3 Test per-command authorization set to 'local'

Prepare steps:

Set device per-command authorization to 'local' with following command:  
  sudo config aaa authorization local

### 1.3.1 Test domain user can run command with local permission

* Test domain user can run command when user have local permission to run the command. Steps:

Domain user login to device.  
Run following command: 'show aaa'

Expected result:

Command success, and output AAA setting in console.

### 1.3.2 Test domain user can't run command when user not have local permission

* Test domain user can't run command when user not have local permission to run the command. Steps:

Domain user login to device.  
Run following command: 'config aaa'

Expected result:

Command failed with following message:  
  'Root privileges are required for this operation'

# 2 Test per-command authorization when tacacs server accessible or not

## 2.1 Test per-command authorization set to 'tacacs+'

Prepare steps:

Set device per-command authorization to 'tacacs+' with following command:  
  sudo config aaa authorization tacacs+

### 2.1.1 Test domain user can run command when tacacs server accessible

* Test domain user can run command in tacacs server whitelist. Steps:

Domain user login to device.  
Run following command: 'show aaa'

Expected result:

Command success, and output AAA setting in console.

### 2.1.2 Test domain user can't run any command when tacacs server not accessible

* Test domain user can't run command not in tacacs server whitelist. Steps:

Domain user login to device.  
Shutdown mgmt interface, so tacacs server not accessible.  
Run following command: 'show aaa'

Expected result:

Command denied with following message:  
  '/usr/bin/config not authorized by TACACS+ with given arguments, not executing'

## 2.2 Test per-command authorization set to 'tacacs+ local'

Prepare steps:

Set device per-command authorization to 'tacacs+ local' with following command:  
  sudo config aaa authorization 'tacacs+ local'

### 2.2.1 Test domain user can run command when tacacs server accessible

* Test domain user can run command in tacacs server whitelist. Steps:

Domain user login to device.  
Run following command: 'show aaa'

Expected result:

Command success, and output AAA setting in console.

### 2.2.2 Test domain user can't run any command when tacacs server not accessible

* Test domain user can't run command not in tacacs server whitelist. Steps:

Domain user login to device.  
Shutdown mgmt interface, so tacacs server not accessible.  
Run following command: 'show aaa'

Expected result:

Command denied with following message:  
  '/usr/bin/config not authorized by TACACS+ with given arguments, not executing'

# 3 Test per-command accounting

## 3.1 Test per-command accounting set to 'tacacs+'

Prepare steps:

Set device per-command accounting to 'tacacs+' with following command:  
  sudo config aaa accounting tacacs+

### 3.1.1 Test command log send to tacacs server

* Test command log can send so tacacs server. Steps:

Domain user login to device.  
Run following command: 'show aaa'

Expected result:

'show aaa' command exist in tacacs server side log.

### 3.1.2 Test password been removed from command log

* Test password been replaced with '\*' in tacacs server side log. Steps:

Domain user login to device.  
Run following command: 'chpasswd newpassword'

Expected result:

In tacacs server side log, new password been replaced with '\*'.

## 3.2 Test per-command accounting set to 'tacacs+ local'

Prepare steps:

Set device per-command accounting to 'tacacs+ local' with following command:  
  sudo config aaa accounting 'tacacs+ local'

### 3.2.1 Test command log send to tacacs server

* Test command log can send so tacacs server, also exist in local log. Steps:

Domain user login to device.  
Run following command: 'show aaa'

Expected result:

'show aaa' command exist in tacacs server side log.  
'show aaa' command exist in local log.

### 3.2.2 Test password been removed from command log

* Test password been replaced with '\*' in tacacs server side log and local log. Steps:

Domain user login to device.  
Run following command: 'chpasswd newpassword'

Expected result:

In tacacs server side log, new password been replaced with '\*'.  
In local log, new password been replaced with '\*'.

## 3.3 Test per-command accounting set to 'local'

Prepare steps:

Set device per-command accounting to 'local' with following command:  
  sudo config aaa accounting local

### 3.3.1 Test command log exist in local log

* Test command log exist in local log. Steps:

Domain user login to device.  
Run following command: 'show aaa'

Expected result:

'show aaa' command exist in local log.

### 3.3.2 Test password been removed from command log

* Test password been replaced with '\*' in local log. Steps:

Domain user login to device.  
Run following command: 'chpasswd newpassword'

Expected result:

In local log, new password been replaced with '\*'.

# 4 References

## SONiC TACACS+ per-command authorization&Accounting design document

<https://github.com/Azure/SONiC/blob/master/doc/aaa/TACACS%2B%20Design.md>