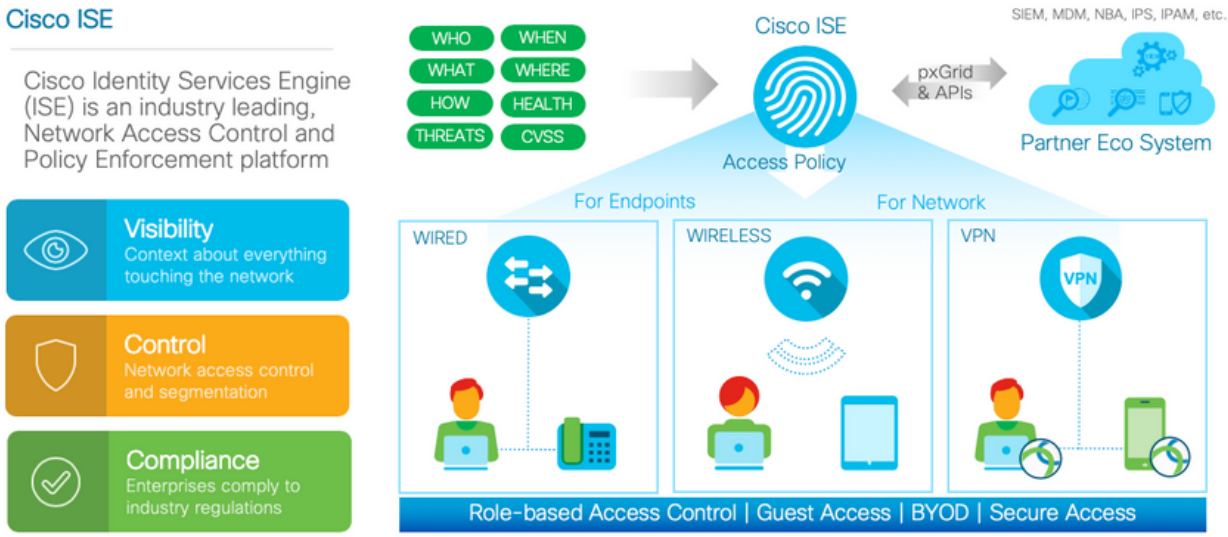
TACACS best practices

This document is intended to provide key details, information related to best practices, tips and tricks for implementation and running TACACS+ based Device Administration services on Cisco Identity Services Engine (ISE) software.

Authors:
Gennady Yakubovich
Satheeskumar Murugan
Chandrabose Sivagnanam

- Introduction
  - About Cisco Identity Service Engine (ISE)

Cisco Identity Services Engine (ISE) is a market leading, identity-based network access control and policy enforcement system. It's a common policy engine for controlling, endpoint access and network device administration for your enterprise. ISE allows an administrator to centrally control access policies for wired, wireless and VPN endpoints in the network.



- Figure1: Cisco Identity Services Engine

ISE builds context about the endpoints that include users and groups (Who), device-type (What), access-time (When), access-location (Where), access-type (Wired/Wireless/VPN) (how), threats and vulnerabilities. Cisco ISE implements Network Administration based on connection oriented TACACS+ protocol, allowing granular, secure and robust management of corporate network devices with extensive logging capabilities.

  - About this guide

This document provides partners and Cisco field engineers with a guide to troubleshoot and tune TACACS+ to its best performance.

- Understanding TACACS+

Unlike RADIUS that uses UDP - TACACS+ uses TCP. TCP offers several advantages over UDP. TCP offers a connection-oriented transport, while UDP offers best-effort delivery. RADIUS requires additional programmable variables such as re-transmit attempts and time-outs to compensate for best-effort transport, but it lacks the level of built-in support that a TCP transport offers:

  - TCP usage provides a separate acknowledgment that a request has been received, within (approximately) a network round-trip time (RTT), regardless of how loaded and slow the backend authentication mechanism (a TCP acknowledgment) might be.
  - TCP provides immediate indication of a crashed, or not running, server by a reset (RST). You can determine when a server crashes and returns to service if you use long-lived TCP connections. UDP cannot tell the difference between a server that is down, a slow server, and a non-existent server.
  - Using TCP keepalives, server crashes can be detected out-of-band with actual requests. Connections to multiple servers can be maintained simultaneously, and you only need to send messages to the ones that are known to be up and running.
  - TCP is more scalable and adapts to growing, as well as congested, networks.

- TACACS+ implementation in ISE - key factors

  TACACS+ in ISE uses TCP port 49 to implement Authentication, Authorization and Accounting. Being connection oriented that adds elevated requirements to the general network health – network noise such as packet loss, congestion, TCP packets reassembly errors will eventually affect TACACS+ communication.

  TACACS+ separates Authentication/Session Authorization from Command authorization. In fact, TACACS+ Authentication session can be directed to one specific PSN in the deployment and Command Authorization session(s) might be directed to other PSNs. Due to the protocol implementation - command Authorization does not require password, hence can be performed by any PSN with only lookup function.

  NOTE: If external ID store, such as RTS or SafeWord configured for Authentication with no lookup abilities - ISE implements Authentication caching to allow subsequent Command Authorization. In this case Load balance “stickiness” should be configured properly to ensure that NAD is pointing transactions to the same PSN during Authentication and Authorization.

- ISE deployment performance
  - TACACS+ AAA performance
    - Low Level (TCP) performance

      As TACACS+ communication based on TCP connection-oriented protocol it is highly important to keep TCP level on the network clean from typical connection errors. Issues such as dropped packets, congested network, packet assembly/reassembly errors will disrupt communication on TCP and, in turn on TACACS+ .

Multiple repetitive AAA transactions may potentially exhaust TCP sockets by quickly allocating connection sockets and relatively slowly releasing them. This is happened due to the fact that each socket being closed after TACACS+ transaction I still being held by the TCP stack in Linux and being transitioned from LISTENING state to TIME_WAIT and then to TIME_CLOSE for 2 minutes, before being finally released to the system. If ingress of connection requests is greater than sockets expiration time – socket exhaustion is likely to happen. In order to avoid socket exhaustion several things can be suggested:

- Reduce number of individual requests by configuring Single Connect
- Reduce overall number of requests by managing TACACS+ transactions - fix reauthentication attempts,

The way to assess TCP connections being accumulated is to check number of ESTABLISHED, TIME_WAIT and TIME_CLOSE connections for the port 49 from CLI:

```
tech netstat | inc  TIME_WAIT | inc :49
tech netstat | inc  TIME_CLOSE | inc :49
tech netstat | inc   ESTABLISH | inc :49
```

It is also available from root access (if needed for deeper troubleshooting) with command:

```
netstat -nat | awk '{print $6}' | uniq -c | sort -n
```

- High Level (Policy Engine) performance

ISE TACACS+ policy engine is powerful and flexible tool to create granular and secure Authorization decisions based on TACACS+ attributes. Will all that Policy Engine configuration need to be optimized in order to ensure maximum capacity and speed. Based on the fact that polices are being processed sequentially - order of policies plays important role - most used polices need to be configured first and least used polices need to be moved to the bottom of the list

In the picture above shown several polices that have most hits but they are resided in the lower part of the policy list. Contrarily - policies with zero hits are at the top of the list. Evidently, some optimization is required.

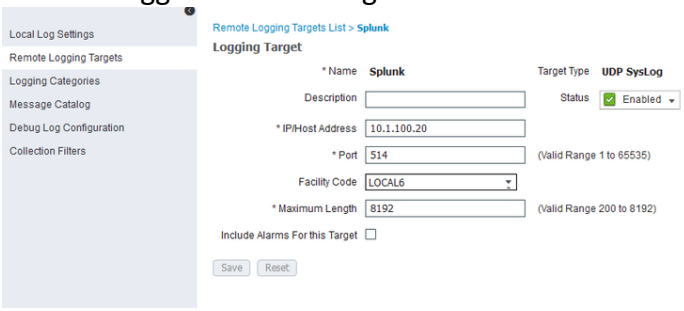
- TACACS+ in Logging performance

  Even though ISE TACACS+ AAA performance might be quite easily matched to the customer's traffic needs by deploying additional PSNs, meeting needs for logging traffic might be challenging sometimes. In fact, customer's TACACS+ infrastructure might be equipped with large number of scripts, device "Health Checks" and "Keep Alive" applications that creates a majority of the TACACS+ transactions. All those transactions are being logged into ISE MnT. ISE deployment providing one "logger" - MnT instance that will accept and process logging traffic from all PSNs in the deployment.

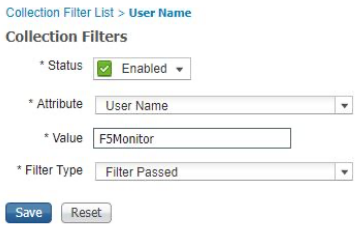
  NOTE: Secondary PSN is hot stand-by and not load balancing logging traffic in the deployment.

  In order to effectively process logging traffic in ISE deployment several things might be recommended

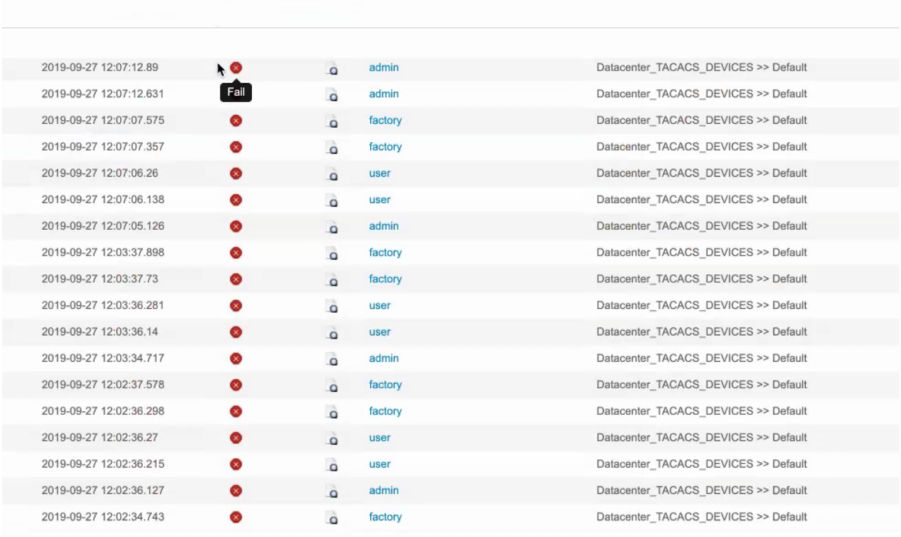
  - Splitting traffic between ISE MnT and external logger. For this purpose, external logger can be configured



  - Implementing Collection Filter in order to "mask" some of the repetitive traffic.

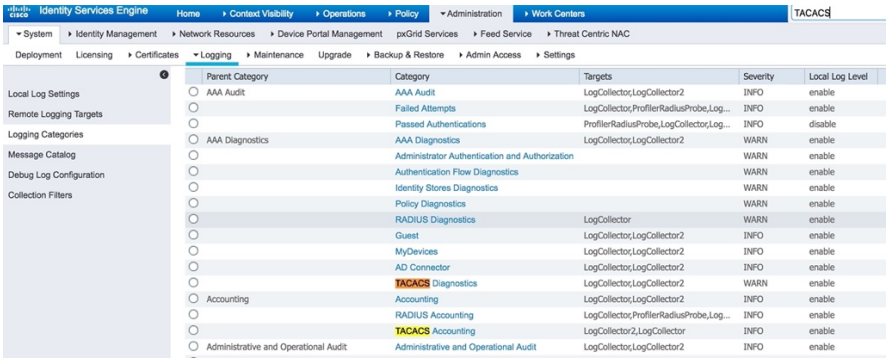


    These filters are especially effective to reduce logging traffic from service accounts that continuously sending AAA transactions and creating a lot of stress onto MnT. On the picture below it is shown multiple service accounts that creating majority of the traffic. They are good "candidates" for a filtering.



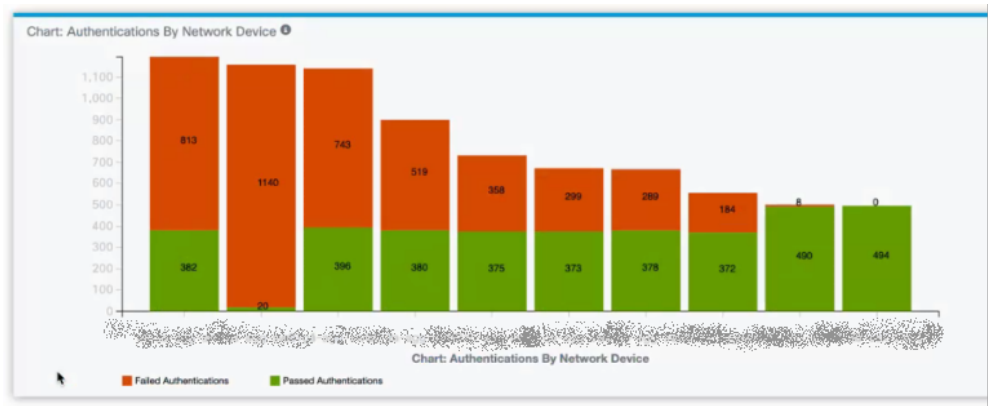
NOTE: Creating and using Collection Filter will not affect Authentication or Authorization for any clients; it will not collect logging data about it.

Accounting traffic – TACACS+ tends to have a lot of accounting traffic and that will create additional Logging load. It might be useful to move Accounting Logging traffic altogether to the external logger and reduce load on the MnT



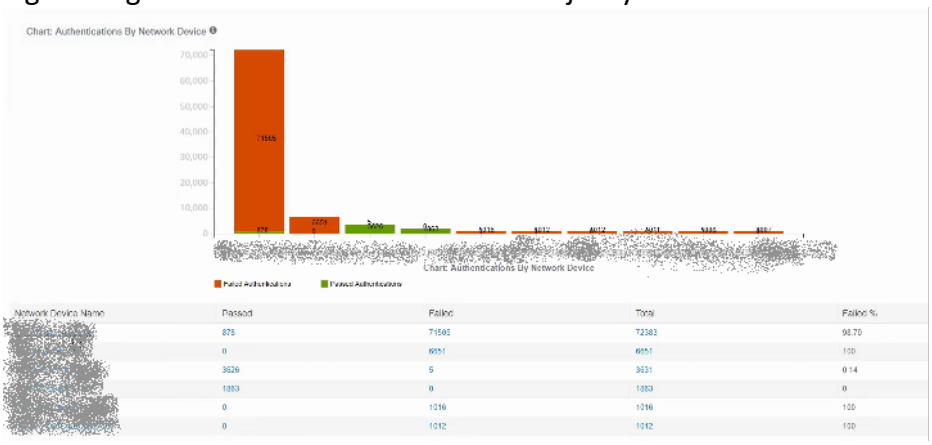
- Considering Single Connect

  In case of heavy traffic and multiple frequent TCP connections are present, it would be useful to enable Single Connect on specific set of NADs. In order to pick devices to work with Single Connect – Top-N most active devices report chart should be analysed. On the chart below, evidently, that first top handful of devices are most “chatty” and they are good candidates to enable Single Connect for them. Single Connect feature will optimize creating and dropping TCP connection by sustaining one single connection over several transactions.
  NOTE: Enable Single Connect for all devices has to be avoid as it will cause to TCP socket exhaustion when all NAD peers will try to maintain Single Connect.
  NOTE: In dynamic corporate environment Single Connect setting need to revised periodically to enable it for devices that are on Top-N chard and disable it for devices that became less “talkative”



Network Device Name	Passed	Failed	Total	Failed %
[redacted]	382	813	1195	68.03
[redacted]	20	1140	1160	98.28
[redacted]	396	743	1139	65.23
[redacted]	380	519	899	57.73
[redacted]	375	358	733	48.84
[redacted]	373	299	672	44.49

- Refraining of usage Default Network Device for majority of the traffic



Network Device Name	Passed	Failed	Total	Failed %
[redacted]	875	71503	72383	98.79
[redacted]	0	6851	6851	100
[redacted]	3626	5	3631	0.14
[redacted]	1883	0	1883	0
[redacted]	0	1216	1216	100
[redacted]	0	1012	1012	100

Using Default Network Device to accept large amount of traffic should be avoided due to next reasons:

- It is unsecure – using Default Network Device does not providing granularity and visibility to use it in the policies. As such it is poorly controlled.
- It is insecure (2) as using separate shared keys is not enforced.
- It is inefficient – it does not facilitate usage of Single Connect

- Deployment Considerations

It is generally not a good idea to deploy RADIUS and TACACS+ services on the same server. There may be a perceived advantage to consolidating these services because they are both AAA protocols, however, they are deployed for different purposes, they use resources differently, and the licensing can be unnecessarily expensive. By combining these services, you may be

increasing costs and reducing your network security. In an enterprise network, unprivileged remote users may be managed by a different operational group than privileged internal administrators. Combining these roles may violate the security principles of separation of duties and least privilege. TACACS+ servers should be deployed in a fully trusted internal network. There should not be any direct access from untrusted or semi-trusted networks. RADIUS is typically deployed in a semi-trusted network, and TACACS+ uses internal administrative logins, so combining these services on the same server could potentially compromise your network security. Deploying TACACS+ server in a semi-trusted network with a connection to your Windows Domain Controllers, you will have to open many ports for LDAP, SMB, Kerberos, etc. You may also need to open ports for DNS and NTP. If you keep your TACACS+ service within your trusted network, you only need to open one port, TCP 49. This is easier to manage and more secure.

Development and bugs that are noteworthy:

1. There is an enhancement request to have in ISE ability to track TCP sockets (in addition to memory and CPU). This enhancement request will allow to see TCP sockets consumption on-line and “predict” its exhaustion. This is very important to see if there TACACS+ is over-consuming sockets and causing to any internal issue.
2. CSCwa55866 ISE addResponse is not triggered. Issue is If the thread is already doing "handle_output" then the new addResponse is not triggered. This impacts when multiple threads are invoked in TACACS single-connect session. As a result, ISE doesn't send out the response packet, and failover is triggered on NAD.