Dépannage de Dot1x sur les commutateurs Catalyst 9000

Table des matières

- Introduction
- Conditions préalables
  - Exigences
  - Composants utilisés
- Informations générales
- Configuration de base
- IBNS 2.0
- Configuration d'IBNS 2.0
- Vérification de la configuration et du fonctionnement
  - Présentation de la norme 802.1x
  - Configuration
  - Session d'authentification
  - Accessibilité au serveur d'authentification
- Dépannage
  - Méthode
  - Exemples de symptômes
  - Utilitaires spécifiques à la plate-forme
    - Exemples de suivi
- Additional Information
  - Paramètres par défaut
  - Paramètres facultatifs
  - Organigrammes
- Informations connexes

Introduction

Ce document décrit comment configurer, valider et dépanner le contrôle d'accès au réseau (NAC) 802.1x sur les commutateurs de la gamme Catalyst 9000.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes .

- Commutateurs Catalyst 9000
- Moteur du service de vérification des identités (ISE)

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Catalyst 9600
- Cisco IOS® XE 17.6.x et versions ultérieures
- ISE-VM-K9 version 3.0.0.458

Remarque : Consultez le guide de configuration approprié pour connaître les commandes utilisées afin d’activer ces fonctionnalités sur d’autres plateformes Cisco.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

La norme 802.1x définit un protocole d'authentification et de contrôle d'accès client-serveur qui empêche les clients non autorisés de se connecter à un réseau local via des ports accessibles au public, à moins qu'ils ne soient correctement authentifiés. Le serveur d’authentification authentifie chaque client connecté à un port de commutateur avant de rendre disponibles les services offerts par le commutateur ou le réseau local.

L'authentification 802.1x comporte trois composants distincts :

Demandeur - Client qui envoie les informations d'identification pour authentification
Authenticator : périphérique réseau qui fournit la connectivité réseau entre le client et le réseau et qui peut autoriser ou bloquer le trafic réseau.
Authentication Server : le serveur qui peut recevoir et répondre aux demandes d'accès réseau

indique à l'authentificateur si la connexion peut être autorisée et divers autres paramètres qui s'appliquent à la session d'authentification.

Ce document est destiné aux ingénieurs et au personnel d'assistance qui ne sont pas nécessairement axés sur la sécurité. Pour plus d'informations sur l'authentification 802.1x basée sur les ports et les composants tels que ISE, consultez le guide de configuration approprié.

Remarque : Consultez le guide de configuration correspondant à votre plate-forme et à votre version de code pour obtenir la configuration d'authentification 802.1x par défaut la plus précise.

Configuration de base

Cette section décrit la configuration de base requise pour implémenter l'authentification 802.1x basée sur les ports. Vous trouverez des explications supplémentaires sur les fonctionnalités dans l'onglet Addenda de ce document. Les normes de configuration varient légèrement d'une version à l'autre. Validez votre configuration par rapport à votre guide de configuration de version actuel.

L'authentification, l'autorisation et la comptabilité (AAA) doivent être activées avant de configurer l'authentification basée sur les ports 802.1x, et une liste de méthodes doit être établie.

- Les listes de méthodes décrivent la séquence et la méthode d'authentification à interroger pour authentifier un utilisateur.
- 802.1x doit également être activé globalement.

<#root>

```
C9300>

enable


C9300#

configure terminal


C9300(config)#

aaa new-model


C9300(config)#
```

```
aaa authentication dot1x default group radius

C9300(config)#
dot1x system-auth-control
```

Définissez un serveur RADIUS sur le commutateur.

```
<#root>
C9300(config)#
radius server RADIUS_SERVER_NAME

C9300(config-radius-server)#
address ipv4 10.0.1.12

C9300(config-radius-server)#
key rad123

C9300(config-radius-server)#
exit
```

Activez 802.1x sur l'interface client.

```
<#root>
C9300(config)#
interface TenGigabitEthernet 1/0/4

C9300(config-if)#
switchport mode access

C9300(config-if)#
```

```
authentication port-control auto


C9300(config-if)#

dot1x pae authenticator


C9300(config-if)#

end
```

IBNS 2.0

Les services IBNS (Identity-Based Networking Services) 2.0 reposent sur le cadre IBNS 1.0 traditionnel et le remplacent. Il résout les problèmes de complexité et de configuration d'IBNS 1.0 en introduisant un moteur de stratégie plus rationalisé et plus puissant appelé Access Session Manager.

Configuration d'IBNS 2.0

- Pour activer IBNS 2.0, vous devez exécuter la commande en mode privilégié sur votre commutateur Cisco :

```
#authentication display new-style
```

- Configurez switchport pour IBNS 2.0 avec les commandes suivantes :

```
access-session host-mode {single-host | multi-domain | multi-auth | multi-host}
access-session port-control auto
dot1x pae authenticator
{mab}
service-policy type control subscriber TEST
```

Ces commandes activent l'authentification dot1x et, en option, MAC Authentication Bypass (MAB) sur l'interface.
Lorsque vous utilisez la nouvelle syntaxe, vous utilisez des commandes qui commencent par access-session.
L'objectif de ces commandes est le même que pour les commandes qui utilisent l'ancienne

syntaxe (en commençant par le mot clé authentication).

Appliquez service-policy pour spécifier policy-map qui peut être utilisé pour l'interface.

- Le policy-map mentionné définit le comportement du commutateur (authentificateur) pendant l'authentification.

Par exemple, vous pouvez spécifier ce qui peut se produire en cas d'échec de l'authentification. Pour chaque événement, vous pouvez configurer plusieurs actions en fonction du type de l'événement correspondant dans le class-map configuré sous celui-ci.
Par exemple, observez la liste comme indiqué (policy-map TEST).
Si le point de terminaison dot1x, qui est connecté à l'interface où cette stratégie est appliquée, échoue, l'action définie dans DOT1X_FAILED est exécutée.
Si vous voulez spécifier le même comportement pour des classes comme MAB_FAILED et DOT1X_FAILED, alors vous pouvez toujours utiliser la classe par défaut - 'class-map'.

```
policy-map type control subscriber TEST
(...)
event authentication-failure match-first
10 class DOT1X_FAILED do-until-failure
10 terminate dot1x
(...)
40 class always do-until-failure
10 terminate mab
20 terminate dot1x
30 authentication-restart 60
(...)
```

- La carte de stratégie utilisée pour IBNS 2.0 doit toujours avoir un abonné de contrôle de type. Vous pouvez afficher la liste des événements disponibles de cette manière :

```
Switch(config-event-control-policymap)#event ?
aaa-available aaa-available event
absolute-timeout absolute timeout event
agent-found agent found event
authentication-failure authentication failure event
authentication-success authentication success event
authorization-failure authorization failure event
authorization-success authorization success event
identity-update identity update event
inactivity-timeout inactivity timeout event
remote-authentication-failure authentication failure event
remote-authentication-success authentication remote success event
remote-update update from remote device
session-disconnected session disconnected event
session-started session started event
tag-added tag to apply event
tag-removed tag to remove event
```

```
template-activated template activated event
template-activation-failed template activation failed event
template-deactivated template deactivated event
template-deactivation-failed template deactivation failed event
timer-expiry timer-expiry event
violation session violation event
```

- Dans la configuration des événements, vous avez la possibilité de définir comment les classes peuvent être évaluées :

```
Switch(config-event-control-policymap)#event authentication-failure ?
match-all Evaluate all the classes
match-first Evaluate the first class
```

- Vous pouvez définir des options similaires pour les class-maps, bien que vous spécifiiez ici comment les actions peuvent être exécutées en cas de correspondance de votre classe :

```
Switch(config-class-control-policymap)#10 class always ?
do-all Execute all the actions
do-until-failure Execute actions until one of them fails
do-until-success Execute actions until one of them is successful
```

- La dernière partie (facultative) de la configuration dans le nouveau style de dot1x est class-map. Il peut également taper un abonné de contrôle et il est utilisé pour faire correspondre un comportement ou un trafic spécifique.
Configurez les exigences pour l'évaluation de la condition class-map.
Vous pouvez spécifier que toutes les conditions ou n'importe quelle condition doivent correspondre, ou qu'aucune des conditions ne correspond.

```
Switch(config)#class-map type control subscriber ?
match-all TRUE if everything matches in the class-map
match-any TRUE if anything matches in the class-map
match-none TRUE if nothing matches in the class-map
```

- Voici un exemple de carte-classe utilisée pour la correspondance d'échec d'authentification dot1x :

```
class-map type control subscriber match-all DOT1X_FAILED
match method dot1x
match result-type method dot1x authoritative
```

- Pour plus d'informations sur IBMS 2.0, reportez-vous aux guides suivants :
Configuration d'IBNS 2.0 pour hôte unique et M

Dépannage de la mise en réseau identitaire

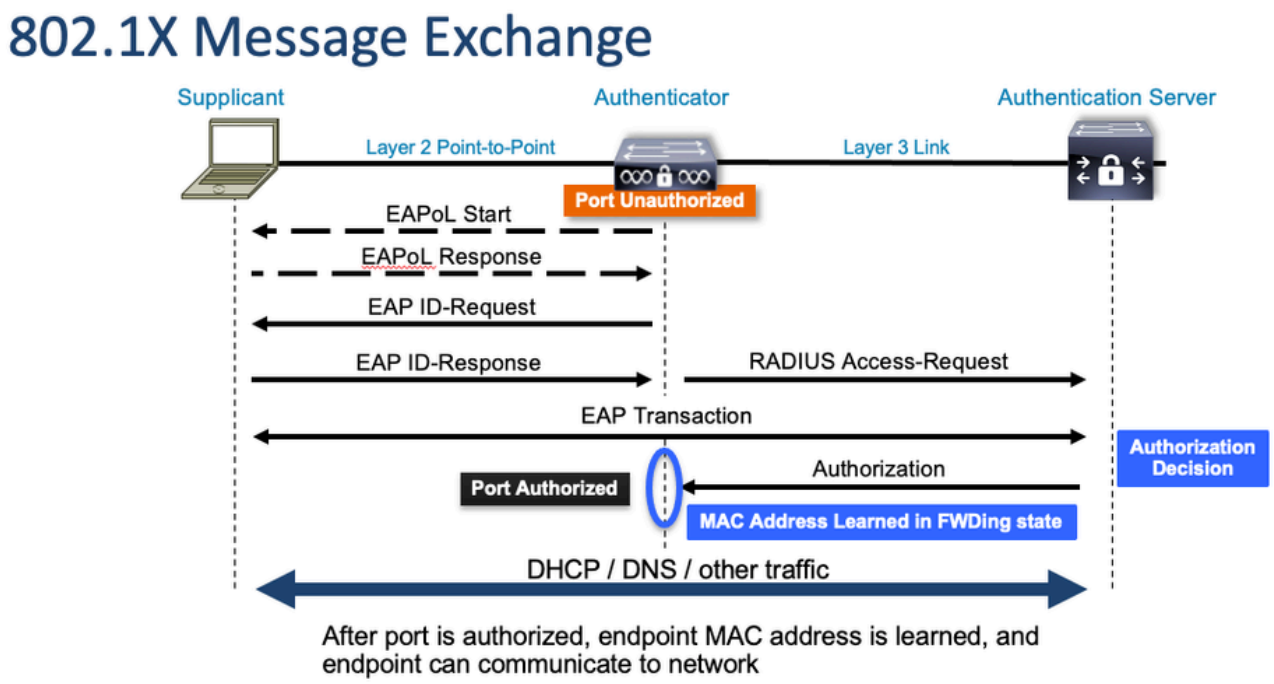
Vérification de la configuration et du fonctionnement

Cette section fournit des informations générales sur la norme 801.1x et explique comment vérifier la configuration et le fonctionnement.

Présentation de la norme 802.1x

La norme 802.1x implique deux types de trafic distincts : le trafic client-authentificateur (point-à-point) sur EAPoL (Extensible Authentication Protocol over LAN) et le trafic authentificateur-serveur d'authentification qui est encapsulé via RADIUS.

Ce diagramme représente le flux de données pour une transaction dot1x simple.



L'authentificateur (commutateur) et le serveur d'authentification (ISE, par exemple) sont souvent séparés par la couche 3. Le trafic RADIUS est acheminé sur le réseau entre l'authentificateur et le serveur. Le trafic EAPoL est échangé sur la liaison directe entre le demandeur (client) et l'authentificateur. L'apprentissage MAC se produit après authentification et autorisation.

Voici quelques questions à garder à l'esprit lorsque vous abordez un problème impliquant la norme 802.1x :

- Est-il correctement configuré ?
- Le serveur d'authentification est-il accessible ?
- Quel est l'état d'Authentication Manager ?
- Y a-t-il des problèmes de remise des paquets entre le client et l’authentificateur ou entre l’authentificateur et le serveur d’authentification ?

Configuration

Certaines configurations varient légèrement d'une version principale à l'autre. Reportez-vous au guide de configuration approprié pour obtenir des conseils spécifiques à la plate-forme/au code.

AAA doit être configuré pour utiliser l'authentification basée sur les ports 802.1x.

Une liste de méthodes d'authentification doit être établie pour « dot1x ». Il s'agit d'une configuration AAA courante dans laquelle 802.1X est activé.

```
<#root>
C9300#

show running-config | section aaa

aaa new-model

<-- This enables AAA.

aaa group server radius ISEGROUP

<-- This block establishes a RADIUS server group named "ISEGROUP".
 server name DOT1x

 ip radius source-interface Vlan1
aaa authentication dot1x default group ISEGROUP

<-- This line establishes the method list for 802.1X authentication. Group ISEGROUP is be used.
aaa authorization network default group ISEGROUP

aaa accounting update newinfo periodic 2880
aaa accounting dot1x default start-stop group ISEGROUP

C9300#

show running-config | section radius
```

```
aaa group server radius ISEGROUP
 server name DOT1x
 ip radius source-interface Vlan1

<-- Notice 'ip radius source-interface' configuration exists in both global configuration and the aaa se

ip radius source-interface Vlan1
radius server DOT1x
 address ipv4 10.122.141.228 auth-port 1812 acct-port 1813

<-- 1812 and 1813 are default auth-port and acct-port, respectively.

 key secretKey
```

Ceci est un exemple de configuration d'interface où 802.1x est activé. MAB (MAC Authentication Bypass) est une méthode de sauvegarde courante pour l'authentification des clients qui ne prennent pas en charge les demandeurs dot1x.

```
<#root>

C9300#

show running-config interface te1/0/4

Building configuration...

Current configuration : 148 bytes
!
interface TenGigabitEthernet1/0/4
 switchport access vlan 50
 switchport mode access
 authentication order dot1x mab

<-- Specifies authentication order, dot1x and then mab


 authentication priority dot1x mab

<-- Specifies authentication priority, dot1x and then mab


 authentication port-control auto

<-- Enables 802.1x dynamic authentication on the port


 mab

<-- Enables MAB
```

```
 dot1x pae authenticator

<-- Puts interface into "authenticator" mode.

end
```

Déterminez si une adresse MAC est apprise sur l'interface avec la commande show mac address-table interface <interface>. L'interface apprend une adresse MAC uniquement lorsqu'elle est authentifiée.

```
<#root>

C9300#

show mac address-table interface te1/0/4

          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  50    0800.2766.efc7    STATIC      Te1/0/4

<-- The "type" is STATIC and the MAC persists until the authentication session is cleared.

Total Mac Addresses for this criterion: 1
```

Session d'authentification

Les commandes show permettent de valider l'authentification 802.1x.

Utilisez show authentication sessions ou show authentication sessions <interface> afin d'afficher des informations sur les sessions d'authentification actuelles. Dans cet exemple, seule Te1/0/4 a une session d'authentification active établie.

```
<#root>

C9300#

show authentication sessions interface te1/0/4

Interface               MAC Address    Method  Domain  Status Fg  Session ID
---------------------------------------------------------------------------------------
Te1/0/4                 0800.2766.efc7 dot1x   DATA    Auth        13A37A0A0000011DC85C34C5
```

```
<-- "Method" and "Domain" in this example are dot1x and DATA, respectively. Multi-domain authentication


Key to Session Events Blocked Status Flags:

  A - Applying Policy (multi-line status for details)
  D - Awaiting Deletion
  F - Final Removal in progress
  I - Awaiting IIF ID allocation
  P - Pushed Session
  R - Removing User Profile (multi-line status for details)
  U - Applying User Profile (multi-line status for details)
  X - Unknown Blocker

Runnable methods list:
  Handle  Priority  Name
      13         5  dot1xSup
       1         5  dot1x
       2        10  webauth
      14        15  mab
```

Show authentication sessions interface <interface> details fournit des détails supplémentaires sur une session d'authentification d'interface spécifique.

<#root>

```
C9300#

show authentication session interface te1/0/4 details

            Interface:  TenGigabitEthernet1/0/4
               IIF-ID:  0x14D66776
          MAC Address:  0800.2766.efc7
         IPv6 Address:  Unknown
         IPv4 Address:  Unknown
            User-Name:  alice
               Status:  Authorized
               Domain:  DATA
      Oper host mode:  multi-auth
     Oper control dir:  both
      Session timeout:  N/A
  Acct update timeout:  172800s (local), Remaining: 152363s
    Common Session ID:  13A37A0A0000011DC85C34C5
      Acct Session ID:  0x00000002
               Handle:  0xe8000015
       Current Policy:  POLICY_Te1/0/4

<-- If a post-authentication ACL is applied, it is listed here.



Local Policies:
    Service Template: DEFAULT_LINKSEC_POLICY_SHOULD_SECURE (priority 150)
      Security Policy:  Should Secure
```

```
Server Policies:


Method status list:
       Method           State
        dot1x           Authc Success
```

<-- This example shows a successful 802.1x authentication session.

Si l'authentification est activée sur une interface mais qu'il n'y a pas de session active, la liste des méthodes exécutables s'affiche. Aucune session ne correspond aux critères fournis s'affiche également.

<#root>

```
C9300#
```

show authentication sessions interface te1/0/5

```
No sessions match supplied criteria.

Runnable methods list:
  Handle  Priority  Name
      13         5  dot1xSup
       1         5  dot1x
       2        10  webauth
      14        15  mab
```

Si aucune authentification n'est activée sur l'interface, aucune présence du Gestionnaire d'authentification n'est détectée sur l'interface. Aucune session ne correspond aux critères fournis s'affiche également.

<#root>

```
C9300#
```

show authentication sessions interface te1/0/6

```
No sessions match supplied criteria.
No Auth Manager presence on this interface
```

Accessibilité au serveur d'authentification

L'accessibilité au serveur d'authentification est une condition préalable à la réussite de

l'authentification 802.1x.

Utilisez la commande ping <server_ip> pour tester rapidement l’accessibilité. Assurez-vous que votre requête ping provient de l'interface source RADIUS.

<#root>

C9300#

ping 10.122.141.228 source vlan 1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.122.141.228, timeout is 2 seconds:
Packet sent with a source address of 10.122.163.19
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

La commande show aaa servers identifie l'état du serveur et fournit des statistiques sur les transactions avec tous les serveurs AAA configurés.

<#root>

C9300#

show aaa servers

```
RADIUS: id 3, priority 1, host 10.122.141.228, auth-port 1812, acct-port 1813, hostname DOT1x <-- Speci
     State: current UP, duration 84329s, previous duration 0s <-- Current State
     Dead: total time 0s, count 1
     Platform State from SMD: current UP, duration 24024s, previous duration 0s
     SMD Platform Dead: total time 0s, count 45
     Platform State from WNCD (1) : current UP
     Platform State from WNCD (2) : current UP
     Platform State from WNCD (3) : current UP
     Platform State from WNCD (4) : current UP
     Platform State from WNCD (5) : current UP
     Platform State from WNCD (6) : current UP
     Platform State from WNCD (7) : current UP
     Platform State from WNCD (8) : current UP, duration 0s, previous duration 0s
     Platform Dead: total time 0s, count 0UP
     Quarantined: No
```

Authen: request 510, timeouts 468, failover 0, retransmission 351 <-- Authentication Statistics

```
             Response: accept 2, reject 2, challenge 38
             Response: unexpected 0, server error 0, incorrect 12, time 21ms
             Transaction: success 42, failure 117
             Throttled: transaction 0, timeout 0, failure 0
             Malformed responses: 0
```

```
        Bad authenticators: 0
        Dot1x transactions:
        Response: total responses: 42, avg response time: 21ms
        Transaction: timeouts 114, failover 0
        Transaction: total 118, success 2, failure 116
        MAC auth transactions:
        Response: total responses: 0, avg response time: 0ms
        Transaction: timeouts 0, failover 0
        Transaction: total 0, success 0, failure 0
Author: request 0, timeouts 0, failover 0, retransmission 0
        Response: accept 0, reject 0, challenge 0
        Response: unexpected 0, server error 0, incorrect 0, time 0ms
        Transaction: success 0, failure 0
        Throttled: transaction 0, timeout 0, failure 0
        Malformed responses: 0
        Bad authenticators: 0
        MAC author transactions:
        Response: total responses: 0, avg response time: 0ms
        Transaction: timeouts 0, failover 0
        Transaction: total 0, success 0, failure 0
Account: request 3, timeouts 0, failover 0, retransmission 0
        Request: start 2, interim 0, stop 1
        Response: start 2, interim 0, stop 1
        Response: unexpected 0, server error 0, incorrect 0, time 11ms
        Transaction: success 3, failure 0
        Throttled: transaction 0, timeout 0, failure 0
        Malformed responses: 0
        Bad authenticators: 0
Elapsed time since counters last cleared: 1d3h4m
Estimated Outstanding Access Transactions: 0
Estimated Outstanding Accounting Transactions: 0
Estimated Throttled Access Transactions: 0
Estimated Throttled Accounting Transactions: 0
Maximum Throttled Transactions: access 0, accounting 0
Consecutive Response Failures: total 115
        SMD Platform : max 113, current 0 total 113
        WNCD Platform: max 0, current 0 total 0
        IOSD Platform : max 2, current 2 total 2
Consecutive Timeouts: total 466
        SMD Platform : max 455, current 0 total 455
        WNCD Platform: max 0, current 0 total 0
        IOSD Platform : max 11, current 11 total 11
Requests per minute past 24 hours:
        high - 23 hours, 25 minutes ago: 4
        low  - 3 hours, 4 minutes ago: 0
        average: 0
```

Utilisez l'utilitaire test aaa pour confirmer l'accessibilité du commutateur au serveur d'authentification. Cet utilitaire est déconseillé et n'est pas disponible indéfiniment.

<#root>

C9300#

debug radius <-- Classic Cisco IOS debugs are only useful in certain scenarios. See "Cisco IOS XE Debugs

```
C9300#

test aaa group ISE username password new-code <-- This sends a RADIUS test probe to the identified serve


User rejected

<-- This means that the RADIUS server received our test probe, but rejected our user. We can conclude th


*Jul 16 21:05:57.632: %PARSER-5-HIDDEN: Warning!!! ' test platform-aaa group server-group ISE user-name
*Jul 16 21:05:57.644: RADIUS/ENCODE(00000000):Orig. component type = Invalid
*Jul 16 21:05:57.644: RADIUS/ENCODE(00000000): dropping service type, "radius-server attribute 6 on-for
*Jul 16 21:05:57.644: RADIUS(00000000): Config NAS IP: 10.122.161.63
*Jul 16 21:05:57.644: vrfid: [65535] ipv6 tableid : [0]
*Jul 16 21:05:57.644: idb is NULL
*Jul 16 21:05:57.644: RADIUS(00000000): Config NAS IPv6: ::
*Jul 16 21:05:57.644: RADIUS(00000000): sending
*Jul 16 21:05:57.644: RADIUS/DECODE(00000000): There is no General DB. Want server details may not be s
*Jul 16 21:05:57.644: RADIUS(00000000): Send Access-Request to 10.122.141.199:1812 id 1645/8, len 50

<-- Sending Access-Request to RADIUS server


RADIUS: authenticator 3B 65 96 37 63 E3 32 41 - 3A 93 63 B6 6B 6A 5C 68
*Jul 16 21:05:57.644: RADIUS: User-Password [2] 18 *
*Jul 16 21:05:57.644: RADIUS: User-Name [1] 6 "username"
*Jul 16 21:05:57.644: RADIUS: NAS-IP-Address [4] 6 10.122.161.63
*Jul 16 21:05:57.644: RADIUS(00000000): Sending a IPv4 Radius Packet
*Jul 16 21:05:57.644: RADIUS(00000000): Started 5 sec timeout
*Jul 16 21:05:57.669: RADIUS: Received from id 1645/8 10.122.141.199:1812, Access-Reject, len 20

<-- Receiving the Access-Reject from RADIUS server


RADIUS: authenticator 1A 11 32 19 12 F9 C3 CC - 6A 83 54 DF 0F DB 00 B8
*Jul 16 21:05:57.670: RADIUS/DECODE(00000000): There is no General DB. Reply server details may not be
*Jul 16 21:05:57.670: RADIUS(00000000): Received from id 1645/8
```

Dépannage

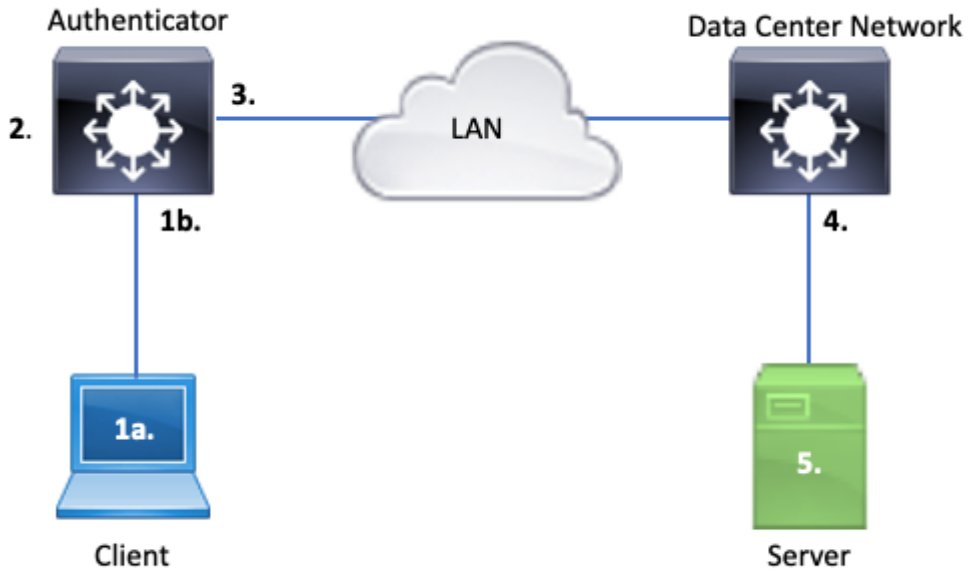
Cette section fournit des conseils sur la façon de dépanner la plupart des problèmes 802.1x sur un commutateur Catalyst.

Méthode

Abordez les problèmes impliquant 802.1x et l'authentification méthodiquement pour obtenir de meilleurs résultats. Voici quelques bonnes questions auxquelles il faut répondre :

- Le problème est-il isolé sur un seul commutateur ? Un seul port ? Un seul type de client ?
- La configuration a-t-elle été validée ? Le serveur d'authentification est-il accessible ?
- Le problème se produit-il à chaque fois ou est-il intermittent ? Se produit-il uniquement avec une nouvelle authentification ou un changement d'autorisation ?

Examiner de bout en bout une transaction unique ayant échoué si des problèmes persistent après que l'évidence a été écartée. L'ensemble de données le meilleur et le plus complet pour l'investigation d'une transaction 802.1x du client au serveur inclut :



1a. Capture sur le client et/ou

1 ter. Sur l'interface d'accès à laquelle le client se connecte

Ce point de référence est essentiel pour nous donner un aperçu des paquets EAPoL échangés entre le port d'accès où dot1x est activé et le client. La fonctionnalité SPAN est l'outil le plus fiable pour afficher le trafic entre le client et l'authentificateur.

2. Débogages sur l'authentificateur

Les débogages nous permettent de suivre la transaction à travers l'authentificateur.

- L'authentificateur doit envoyer les paquets EAPoL reçus et générer du trafic encapsulé RADIUS monodiffusion destiné au serveur d'authentification.

- Assurez-vous que les niveaux de débogage appropriés sont définis pour une efficacité maximale.

3. Capture adjacente à l'authentificateur

Cette capture nous permet de voir la conversation entre Authenticator et le serveur d'authentification.

- Cette capture affiche avec précision l'intégralité de la conversation du point de vue de l'authentificateur.
- Associé à la capture du point 4, vous pouvez déterminer s'il existe une perte entre le serveur d'authentification et l'authentificateur.

4. Capture adjacente au serveur d’authentification

Cette capture accompagne celle décrite au point 3.

- Cette capture fournit l'intégralité de la conversation du point de vue du serveur d'authentification.
- Associé à la capture du point 3, vous pouvez déterminer s'il existe une perte entre Authenticator et Authentication Server.

5. Capture, débogage, connexion au serveur d'authentification

La dernière pièce du puzzle, les débogages du serveur nous disent ce que le serveur sait de notre transaction.

- Avec cet ensemble de données de bout en bout, un ingénieur réseau peut déterminer où la transaction se casse et exclure les composants qui ne contribuent pas au problème.

Exemples de symptômes

Cette section fournit une liste des symptômes courants et des scénarios de problèmes.

- Aucune réponse du client

Si le trafic EAPoL généré par le commutateur ne provoque pas de réponse, ce syslog s'affiche :

```
Aug 23 11:23:46.387 EST: %DOT1X-5-FAIL: Switch 1 R0/0: sessmgrd: Authentication failed for client (aaaa
```

Le code raison, No Response from Client, indique que le commutateur a démarré le processus dot1x, mais qu'aucune réponse n'a été reçue du client dans le délai imparti.
Cela signifie que le client n'a pas reçu ou compris le trafic d'authentification envoyé par le port de commutateur, ou que la réponse du client n'a pas été reçue sur le port de commutateur.

- Le client abandonne la session

Si une session d'authentification est démarrée mais ne se termine pas, le serveur d'authentification (ISE par exemple) signale que le client a démarré une session, mais qu'il a abandonné la session avant de la terminer.
Souvent, cela signifie que le processus d'authentification ne peut être que partiellement terminé.

Assurez-vous que la transaction complète entre le commutateur d'authentification et le serveur d'authentification est livrée de bout en bout et correctement interprétée par le serveur d'authentification.

Si le trafic RADIUS est perdu sur le réseau ou acheminé d'une manière qui ne peut pas être correctement assemblé, la transaction est incomplète et le client relance l'authentification. Le serveur signale à son tour que le client a abandonné sa session.

- Le client MAB échoue DHCP/revient à APIPA

Le contournement d'authentification MAC (MAB) permet l'authentification basée sur l'adresse MAC. Souvent, les clients qui ne prennent pas en charge le logiciel demandeur s'authentifient via MAB.

Si MAB est utilisé comme méthode de secours pour l'authentification alors que dot1x est la méthode préférée et initiale qui s'exécute sur un port de commutateur, un scénario peut se produire lorsque le client ne peut pas exécuter DHCP.

Le problème se résume à l'ordre des opérations. Pendant l'exécution de dot1x, le port du commutateur consomme des paquets autres qu'EAPoL jusqu'à ce que l'authentification soit terminée ou que dot1x expire. Cependant, le client tente immédiatement d'obtenir une adresse IP et diffuse ses messages de détection DHCP. Ces messages de détection sont consommés par le port du commutateur jusqu'à ce que dot1x dépasse ses valeurs de délai d'attente configurées et que MAB puisse s'exécuter. Si le délai d'expiration DHCP du client est inférieur au délai d'expiration dot1x, le protocole DHCP échoue et le client revient à l'APIPA ou à toute autre stratégie de retour imposée par le client.

Ce problème est évité de plusieurs manières. Privilégiez MAB sur les interfaces où les clients authentifiés MAB se connectent. Si dot1x doit s'exécuter en premier, gardez à l'esprit le comportement DHCP du client et ajustez les valeurs de délai d'attente de manière appropriée.

Prenez garde au comportement du client lorsque dot1x et MAB sont utilisés. Une configuration valide peut entraîner un problème technique, comme décrit précédemment.

Utilitaires spécifiques à la plate-forme

Cette section présente de nombreux utilitaires spécifiques à la plate-forme disponibles sur la gamme de commutateurs Catalyst 9000, utiles pour résoudre les problèmes de dot1x.

- Analyseur de port de commutateur (SPAN)

La fonctionnalité SPAN permet à l'utilisateur de mettre en miroir le trafic d'un ou plusieurs ports vers un port de destination pour la capture et l'analyse. La fonctionnalité SPAN locale est l'utilitaire de capture le plus fiable.

- Capture de paquets intégrée (EPC)

EPC tire parti des ressources processeur et mémoire pour fournir une capacité de capture locale de paquets intégrée.

Il existe des limites à la CPE qui ont une incidence sur son efficacité dans l'examen de certains problèmes. Le débit EPC est limité à 1 000 paquets par seconde. EPC ne peut pas non plus capturer de manière fiable les paquets injectés par le processeur à la sortie des interfaces physiques. Ceci est important lorsque l'accent est mis sur la transaction RADIUS entre le commutateur d'authentificateur et le serveur d'authentification. Souvent, le débit de trafic sur l'interface qui fait face au serveur dépasse largement 1000 paquets par seconde. En outre, un EPC à la sortie de l'interface qui fait face au serveur ne peut pas capturer le trafic généré par le commutateur d'authentification.

Utilisez des listes d'accès bidirectionnelles pour filtrer l'EPC afin d'éviter l'impact de la limitation de 1 000 paquets par seconde. Si vous êtes intéressé par le trafic RADIUS entre l'authentificateur et le serveur, concentrez-vous sur le trafic entre l'adresse de l'interface source RADIUS de l'authentificateur et l'adresse du serveur.

Si le périphérique en amont suivant vers le serveur d'authentification est un commutateur Catalyst, utilisez un EPC filtré sur la liaison descendante vers le commutateur d'authentification pour obtenir les meilleurs résultats.

Reportez-vous à ce guide de configuration pour plus de détails sur la configuration et la mise en oeuvre.

- Débogages de Cisco IOS XE

Les modifications de l'architecture logicielle qui ont débuté avec la version 16.3.2 de Cisco IOS XE ont déplacé les composants AAA vers un démon Linux distinct. Les débogages familiers n'activent plus les débogages visualisables dans la mémoire tampon de journalisation.



Conseil : Les débogages AAA IOS traditionnels ne fournissent plus de sortie dans les journaux système pour l'authentification des ports du panneau avant dans la mémoire tampon syslog.

Au lieu de cela, ces débogages classiques de Cisco IOS pour dot1x et RADIUS n'activent plus les débogages visibles dans la mémoire tampon de journalisation du commutateur :

```
debug radius
debug access-session all
debug dot1x all
```

Les débogages des composants AAA sont désormais accessibles via la trace système sous le démon Session Manager (SMD).

- Comme les syslogs traditionnels, les rapports de suivi du système Catalyst à un niveau par défaut et doivent être chargés de collecter des journaux plus approfondis.
- Modifiez le niveau de suivi de routine pour le sous-composant souhaité à l'aide de la commande set platform software trace smd switch active r0 <component> debug.

<#root>

```
Switch#

set platform software trace smd switch active R0 auth-mgr debug

 <<<--- This sets the "auth-mgr" subcomponent to "debug" log level.
```

Ce tableau associe les débogages IOS traditionnels à leur équivalent de trace.

Commande Ancien style	Commande Nouveau style
#debug radius	#set plate-forme logicielle trace smd switch active R0 radius debug
#debug dot1x all	#set plate-forme logicielle trace smd switch active R0 dot1x-all debug
#debug access-session all	#set plate-forme logicielle trace smd switch active R0 auth-mgr-all debug
#debug epm all	#set plate-forme logicielle trace smd switch active R0 epm-all debug

Les débogages classiques activent toutes les traces de composants associées au niveau de débogage. Les commandes de plate-forme sont également utilisées pour activer des suivis spécifiques, le cas échéant.

Utilisez la commande show platform software trace level smd switch active R0 afin d'afficher le niveau de trace actuel pour les sous-composants SMD.

<#root>

```
Switch#

show platform software trace level smd switch active R0


Module Name                      Trace Level
----------------------------------------------
aaa

Notice



<--- Default level is "Notice"


aaa-acct                         Notice
aaa-admin                        Notice
aaa-api                          Notice
aaa-api-attr                     Notice
<snip>
auth-mgr

Debug <--- Subcomponent "auth-mgr" traces at "debug" level
```

```
auth-mgr-all                    Notice
<snip>
```

Le niveau de trace du sous-composant peut être restauré à sa valeur par défaut de deux manières.

- Utilisez undebug all ou set platform software trace smd switch active R0 <sous-composant> notice afin de restaurer.
- Si le périphérique se recharge, les niveaux de suivi reprennent également leurs valeurs par défaut.

<#root>

```
Switch#

undebug all


All possible debugging has been turned off

or

Switch#

set platform software trace smd switch active R0 auth-mgr notice



<--- Sets sub-component "auth-mgr" to trace level "Notice", the system default.
```

Les journaux de suivi des composants peuvent être affichés sur la console ou enregistrés dans des archives et affichés hors connexion. Les traces sont archivées dans des archives binaires compressées qui nécessitent un décodage. Contactez le TAC pour obtenir de l'aide au débogage lorsque vous traitez des traces archivées. Ce workflow explique comment afficher les traces dans l'interface de ligne de commande.

Show logging process est l'utilitaire mis à jour pour les suivis et la norme dans la version Cisco IOS XE 17.9.x et ultérieure.

<#root>

```
C9300#

show logging process smd ?
```

```
  <0-25>            instance number
  end               specify log filtering end location
  extract-pcap      Extract pcap data to a file
  filter            specify filter for logs
  fru               FRU specific commands
  internal          select all logs. (Without the internal keyword only
                    customer curated logs are displayed)
  level             select logs above specific level
  metadata          CLI to display metadata for every log message
  module            select logs for specific modules
  reverse           show logs in reverse chronological order
  start             specify log filtering start location
  switch            specify switch number
  to-file           decode files stored in disk and write output to file
  trace-on-failure  show the trace on failure summary
  |                 Output modifiers
```

Le processus Show logging offre les mêmes fonctionnalités que show platform software trace dans un format plus élégant et plus accessible.

<#root>

```
C9300#
```

clear auth sessions

```
C9300#
```

show logging process smd reverse

```
Logging display requested on 2023/05/02 16:44:04 (UTC) for Hostname: [C9300], Model: [C9300X-24HX], Ver

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis 1 ...
============================================================
 UTM [LUID NOT FOUND] ............... 0
 UTM [PCAP] ......................... 0
 UTM [MARKER] ....................... 0
 UTM [APP CONTEXT] .................. 0
 UTM [TDL TAN] ...................... 5
 UTM [MODULE ID] .................... 0
 UTM [DYN LIB] ...................... 0
 UTM [PLAIN TEXT] ................... 6
 UTM [ENCODED] ...................... 85839
 UTM [Skipped / Rendered / Total] .. 85128 / 722 / 85850
 Last UTM TimeStamp ................. 2023/05/02 16:44:03.775663010
 First UTM TimeStamp ................ 2023/05/02 15:52:18.763729918
============================================================
----------------- Decoder Output Information --------------
============================================================
 MRST Filter Rules ...... 1
 UTM Process Filter ..... smd
 Total UTM To Process ... 85850
 Total UTF To Process ... 1
 Num of Unique Streams .. 1
```

```
============================================================
---------------- Decoder Input Information --------------
============================================================
======= Unified Trace Decoder Information/Statistics ======
============================================================
2023/05/02 16:44:03.625123675 {smd_R0-0}{1}: [radius] [22624]: (ERR): Failed to mark Identifier for reu
2023/05/02 16:44:03.625123382 {smd_R0-0}{1}: [radius] [22624]: (ERR): RSPE-  Set Identifier Free for Re
2023/05/02 16:44:03.625116747 {smd_R0-0}{1}: [radius] [22624]: (info): Valid Response Packet, Free the
2023/05/02 16:44:03.625091040 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  authenticator 2b f4 ea
2023/05/02 16:44:03.625068520 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS: Received from id 1813/9
2023/05/02 16:44:03.610151863 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS: Started 5 sec timeout
2023/05/02 16:44:03.610097362 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Acct-Delay-Time     [41
2023/05/02 16:44:03.610090044 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Event-Timestamp     [55
2023/05/02 16:44:03.610085857 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Acct-Status-Type    [40
2023/05/02 16:44:03.610040912 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Class               [25
2023/05/02 16:44:03.610037444 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Acct-Authentic      [45
2023/05/02 16:44:03.610032802 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Acct-Session-Id     [44
2023/05/02 16:44:03.610028677 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Called-Station-Id   [30
2023/05/02 16:44:03.610024641 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Nas-Identifier      [32
2023/05/02 16:44:03.610020641 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Calling-Station-Id  [31
2023/05/02 16:44:03.610016809 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  NAS-Port            [5]
2023/05/02 16:44:03.610012487 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  NAS-Port-Type       [61
2023/05/02 16:44:03.610007504 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  NAS-Port-Id         [87
2023/05/02 16:44:03.610003581 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  NAS-IP-Address      [4]
2023/05/02 16:44:03.609998136 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Calling-Station-Id  [31
2023/05/02 16:44:03.609994109 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Called-Station-Id   [30
2023/05/02 16:44:03.609989329 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:   Cisco AVpair       [1]
2023/05/02 16:44:03.609985171 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Vendor, Cisco       [26
2023/05/02 16:44:03.609981606 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:   Cisco AVpair       [1]
2023/05/02 16:44:03.609976961 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  Vendor, Cisco       [26
2023/05/02 16:44:03.609969166 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  User-Name           [1]
2023/05/02 16:44:03.609963241 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS:  authenticator 0b 99 e3
2023/05/02 16:44:03.609953614 {smd_R0-0}{1}: [radius] [22624]: (info): RADIUS: Send Accounting-Request
2023/05/02 16:44:03.609863172 {smd_R0-0}{1}: [auth-mgr] [22624]: (info): [0800.2766.efc7:Te1/0/4] Handl
2023/05/02 16:44:03.609695649 {smd_R0-0}{1}: [dot1x] [22624]: (info): [0800.2766.efc7:Te1/0/4] EAPOL pa
2023/05/02 16:44:03.609689224 {smd_R0-0}{1}: [dot1x] [22624]: (info): [0000.0000.0000:unknown] Pkt body
2023/05/02 16:44:03.609686794 {smd_R0-0}{1}: [dot1x] [22624]: (info): [0800.2766.efc7:Te1/0/4] EAP Pack
2023/05/02 16:44:03.609683919 {smd_R0-0}{1}: [dot1x] [22624]: (info): [0800.2766.efc7:Te1/0/4] Sent EAP
2023/05/02 16:44:03.609334292 {smd_R0-0}{1}: [dot1x] [22624]: (info): [0000.0000.0000:Te1/0/4] Sending
2023/05/02 16:44:03.609332867 {smd_R0-0}{1}: [dot1x] [22624]: (info): [0000.0000.0000:Te1/0/4] Setting
2023/05/02 16:44:03.609310820 {smd_R0-0}{1}: [dot1x] [22624]: (info): [0800.2766.efc7:Te1/0/4] Posting
2023/05/02 16:44:03.609284841 {smd_R0-0}{1}: [auth-mgr] [22624]: (info): [0800.2766.efc7:Te1/0/4] Raisi
```

Exemples de suivi

Cette section inclut les suivis du gestionnaire de session pour les composants dot1x et radius pour une transaction complète ayant échoué (le serveur rejette les informations d'identification du client). L'objectif est de fournir une ligne directrice de base pour naviguer dans les traces système liées à l'authentification du panneau avant.

- Un client test tente de se connecter à GigabitEthernet1/0/2 et est rejeté.

Dans cet exemple, les traces de composant SMD sont définies sur « debug ».

<#root>

C9300#

set platform software trace smd sw active r0 dot1x-all

C9300#

set platform software trace smd sw active r0 radius debug

EAPoL : COMMENCER

```
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0000.0000.0000:unknown] Received EAPOL pkt (size=92) on l2 s
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] queuing an EAPOL pkt on Auth Q
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Dequeued pkt: CODE= 0,TYPE= 0,LEN= 0
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Received pkt saddr = 0040.E93E.0000 ,
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Couldn't find the supplicant in the l
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] New client detected, sending session
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: initialising
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: disconnected
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: entering restart
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Sending create new context event to E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: entering init state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering idle state
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Created a client entry (0x0A00000E)
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Dot1x authentication started for 0x0A
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting !EAP_RESTART on Client 0x0A0
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:enter connecting state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: restart connecting
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting RX_REQ on Client 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: authenticating state ent
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:connecting authenticating
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting AUTH_START for 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering request state
```

EAPoL : IDENTITÉ DE DEMANDE EAP

```
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Sending EAPOL packet
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Sending out EAPOL packet
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] EAPOL packet sent to client 0x0A00000
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:idle request action
```

EAPoL : RÉPONSE EAP

```
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0000.0000.0000:unknown] Received EAPOL pkt (size=92) on l2 s
```

```
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Queuing an EAPOL pkt on Authenticator
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Dequeued pkt: CODE= 2,TYPE= 1,LEN= 14
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Received pkt saddr = 0040.E93E.0000 ,
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting EAPOL_EAP for 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering response state
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Response sent to the server from 0x0A
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:request response action
[aaa-authen]: [16498]: UUID: 0, ra: 0 (debug): AAA/AUTHEN/8021X (00000000): Pick method list 'default'
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS/ENCODE: Best Local IP-Address 172.28.99.147 for Radiu
[radius-failover]: [16498]: UUID: 0, ra: 0 (debug): RADIUS/ENCODE: Nas-Identifier "C9300"
[aaa-author]: [16498]: UUID: 0, ra: 0 (debug): VALID SG handle
```

RADIUS: DEMANDE D'ACCÈS

```
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Send Access-Request to 172.28.99.252:1645
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: authenticator 59 c9 e0 be 4d b5 1c 11 - 02 cb 5b eb
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: User-Name [1] 11 "ixia_data"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Service-Type [6] 6 Framed [2]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 27
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 21 "service-type=Framed"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Framed-MTU [12] 6 1500
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Calling-Station-Id [31] 19 "00-40-E9-3E-00-00"
0e 01 69 78 69 61 5f 64 61 74 61 [ ixia_data]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Message [79] 16
69 87 3c 61 80 3a 31 a8 73 2b 55 76 f4 [ Ei<a:1s+Uv]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Message-Authenticator[80] 18
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Key-Name [102] 2 *
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 49
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 43 "audit-session-id=AC1C6393000000
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 20
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 14 "method=dot1x"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-IP-Address [4] 6 172.28.99.147
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Nas-Identifier [32] 8 "C9300"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port-Id [87] 22 "GigabitEthernet1/0/2"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port-Type [61] 6 Ethernet [15]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Sending a IPv4 Radius Packet
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Started 5 sec timeout
```

RADIUS: CONTESTATION D'ACCÈS

```
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Received from id 1645/82 172.28.99.252:0, Access-Cha
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port [5] 6 50014 RADIUS: authenticator 82 71 61
RADIUS: 33 37 43 50 4d 53 65 73 73 69 6f 6e 49 44 3d 41 [37CPMSessionID=A]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: State [24] 81
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 43 31 43 36 33 39 33 30 30 30 30 30 30 31 37 45 [C1C
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 35 33 43 36 45 38 45 3b 33 36 53 65 73 73 69 6f [53C
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 6e 49 44 3d 69 73 65 2d 70 6f 6c 2d 65 73 63 2f [nID=
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 32 34 30 31 39 38 34 32 39 2f 38 39 32 34 3b [ 24019
RADIUS: 01 f9 00 06 0d 20 [ ]
02/15 14:01:28.986 [radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Message [79] 8
RADIUS: 78 66 ec be 2c a4 af 79 5e ec c6 47 8b da 6a c2 [ xf,y^Gj]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Message-Authenticator[80] 18
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS:rad_code 11, suppress reject flag 0
```

```
[radius-authen]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Received from id 1645/82
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting EAP_REQ for 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:exiting response state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering request state###
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Sending EAPOL packet
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Sending out EAPOL packet
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] EAPOL packet sent to client 0x0A00000
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:response request action
```

EAPoL : RÉPONSE EAP

```
02/15 14:01:28.988 [dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0000.0000.0000:unknown] Received EAPOL pk
02/15 14:01:28.989 [dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Queuing an EAPOL p
02/15 14:01:28.989 [dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Dequeued pkt: CODE
02/15 14:01:28.989 [dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Received pkt saddr
02/15 14:01:28.989 [dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting EAPOL_EAP
02/15 14:01:28.989 [dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:enteri
02/15 14:01:28.989 [dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Response sent to t
02/15 14:01:28.989 [dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:reques
02/15 14:01:28.989 [aaa-authen]: [16498]: UUID: 0, ra: 0 (debug): AAA/AUTHEN/8021X (00000000): Pick met
02/15 14:01:28.990 [radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS/ENCODE: Best Local IP-Address 172.
02/15 14:01:28.990 [radius-failover]: [16498]: UUID: 0, ra: 0 (debug): RADIUS/ENCODE: Nas-Identifier "C
02/15 14:01:28.990 [aaa-author]: [16498]: UUID: 0, ra: 0 (debug): VALID SG handle
```

RADIUS: DEMANDE D'ACCÈS

```
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Send Access-Request to 172.28.99.252:1645
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: authenticator 3d 31 3f ee 14 b8 9d 63 - 7a 8b 52 90
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: User-Name [1] 11 "ixia_data"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Service-Type [6] 6 Framed [2]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 27
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 21 "service-type=Framed"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Framed-MTU [12] 6 1500
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Calling-Station-Id [31] 19 "00-40-E9-3E-00-00"
RADIUS: 02 f9 00 06 03 04
02/15 14:01:28.991 [radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Message [79] 8
RADIUS: 8b 2a 2e 75 90 a2 e1 c9 06 84 c9 fe f5 d0 98 39 [ *.u9]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Message-Authenticator[80] 18
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Key-Name [102] 2 *
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 49
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 43 "audit-session-id=AC1C6393000000
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 20
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 14 "method=dot1x"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-IP-Address [4] 6 172.28.99.147
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Nas-Identifier [32] 8 "C9300"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port-Id [87] 22 "GigabitEthernet1/0/2"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port-Type [61] 6 Ethernet [15]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port [5] 6 50014
RADIUS: 33 37 43 50 4d 53 65 73 73 69 6f 6e 49 44 3d 41 [37CPMSessionID=A]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: State [24] 81
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 43 31 43 36 33 39 33 30 30 30 30 30 30 31 37 45 [C1C
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 35 33 43 36 45 38 45 3b 33 36 53 65 73 73 69 6f [53C
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 6e 49 44 3d 69 73 65 2d 70 6f 6c 2d 65 73 63 2f [nID
```

```
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 32 34 30 31 39 38 34 32 39 2f 38 39 32 34 3b [ 24019
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Sending a IPv4 Radius Packet
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Started 5 sec timeout
```

RADIUS: CONTESTATION D'ACCÈS

```
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Received from id 1645/83 172.28.99.252:0, Access-Cha
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: authenticator 0c 8d 49 80 0f 51 89 fa - ba 22 2f 96
RADIUS: 33 37 43 50 4d 53 65 73 73 69 6f 6e 49 44 3d 41 [37CPMSessionID=A]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: State [24] 81
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 43 31 43 36 33 39 33 30 30 30 30 30 30 31 37 45 [C1C
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 35 33 43 36 45 38 45 3b 33 36 53 65 73 73 69 6f [53C
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 6e 49 44 3d 69 73 65 2d 70 6f 6c 2d 65 73 63 2f [nID=
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 32 34 30 31 39 38 34 32 39 2f 38 39 32 34 3b [ 24019
RADIUS: 01 fa 00 21 04 10 5b d0 b6 4e 68 37 6b ca 5e 6f 5a 65 78 04 77 bf 69 73 65 2d [![Nh7k^oZexwise-
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Message [79] 35
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 70 6f 6c 2d 65 73 63 [ pol-esc]
RADIUS: a3 0d b0 02 c8 32 85 2c 94 bd 03 b3 22 e6 71 1e [ 2,"q]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Message-Authenticator[80] 18
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS:rad_code 11, suppress reject flag 0
[radius-authen]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Received from id 1645/83
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting EAP_REQ for 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:exiting response state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering request state
```

EAPoL : DEMANDE EAP

```
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Sending EAPOL packet
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Sending out EAPOL packet
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] EAPOL packet sent to client 0x0A00000
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:response request action
```

EAPoL : RÉPONSE EAP

```
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0000.0000.0000:unknown] Received EAPOL pkt (size=92) on l2 s
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Queuing an EAPOL pkt on Authenticator
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Dequeued pkt: CODE= 2,TYPE= 4,LEN= 31
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Received pkt saddr = 0040.E93E.0000 ,
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting EAPOL_EAP for 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering response state
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Response sent to the server from 0x0A
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:request response action
[aaa-authen]: [16498]: UUID: 0, ra: 0 (debug): AAA/AUTHEN/8021X (00000000): Pick method list 'default'
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS/ENCODE: Best Local IP-Address 172.28.99.147 for Radiu
[radius-failover]: [16498]: UUID: 0, ra: 0 (debug): RADIUS/ENCODE: Nas-Identifier "C9300"
[aaa-author]: [16498]: UUID: 0, ra: 0 (debug): VALID SG handle
```

RADIUS: DEMANDE D'ACCÈS

```
radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Send Access-Request to 172.28.99.252:1645 i
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: authenticator 41 4d 76 8e 03 93 9f 05 - 5e fa f1 d6
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: User-Name [1] 11 "ixia_data"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Service-Type [6] 6 Framed [2]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 27
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 21 "service-type=Framed"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Framed-MTU [12] 6 1500
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Calling-Station-Id [31] 19 "00-40-E9-3E-00-00"
RADIUS: 02 fa 00 1f 04 10 02 b6 bc aa f4 91 2b d6 cf 9e 3b d5 44 96 78 d5 69 78 69 61 5f 64 61 74 61 [
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Message [79] 33
RADIUS: 3b 70 b1 dd 97 ac 47 ae 81 ca f8 78 5b a3 7b fe [ ;pGx[{]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Message-Authenticator[80] 18
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Key-Name [102] 2 *
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 49
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 43 "audit-session-id=AC1C6393000000
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Vendor, Cisco [26] 20
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Cisco AVpair [1] 14 "method=dot1x"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-IP-Address [4] 6 172.28.99.147
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Nas-Identifier [32] 8 "C9300"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port-Id [87] 22 "GigabitEthernet1/0/2"
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port-Type [61] 6 Ethernet [15]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: NAS-Port [5] 6 50014
RADIUS: 33 37 43 50 4d 53 65 73 73 69 6f 6e 49 44 3d 41 [37CPMSessionID=A]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: State [24] 81
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 43 31 43 36 33 39 33 30 30 30 30 30 30 31 37 45 [C1C
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 35 33 43 36 45 38 45 3b 33 36 53 65 73 73 69 6f [53C
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 6e 49 44 3d 69 73 65 2d 70 6f 6c 2d 65 73 63 2f [nID=
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: 32 34 30 31 39 38 34 32 39 2f 38 39 32 34 3b [ 24019
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Sending a IPv4 Radius Packet
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Started 5 sec timeout
```

RADIUS: REFUS D'ACCÈS

```
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Received from id 1645/84 172.28.99.252:0, Access-Rej
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: authenticator d1 a3 eb 43 11 45 6b 8f - 07 a7 34 dd
RADIUS: 04 fa 00 04
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: EAP-Message [79] 6
RADIUS: 80 77 07 f7 4d f8 a5 60 a6 b0 30 e4 67 85 ae ba [ wM`0g]
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS: Message-Authenticator[80] 18
[radius]: [16498]: UUID: 0, ra: 0 (debug): RADIUS:rad_code 3, suppress reject flag 0
[radius-authen]: [16498]: UUID: 0, ra: 0 (debug): RADIUS(00000000): Received from id 1645/84
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Received an EAP Fail
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting EAP_FAIL for 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:exiting response state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering fail state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:response fail action
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering idle state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting AUTH_FAIL on Client 0x0A0000
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:exiting authenticating st
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering authc result sta
[errmsg]: [16498]: UUID: 0, ra: 0 (note): %DOT1X-5-FAIL: Authentication failed for client (0040.E93E.00
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Added username in dot1x
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Dot1x did not receive any key data
```

```
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Received Authz fail (result: 2) for t
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting_AUTHZ_FAIL on Client 0x0A000
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: held
```

EAPoL : REJET EAP

```
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Sending EAPOL packet
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0000.0000.0000:Gi1/0/2] Sending out EAPOL packet
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] EAPOL packet sent to client 0x0A00000
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting FAILOVER_RETRY on Client 0x0A
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: exiting held state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: entering restart
[dot1x]: [16498]: UUID: 0, ra: 0 (info): [0040.E93E.0000:Gi1/0/2] Sending create new context event to EA
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:restart action called
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting RESTART on Client 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting !EAP_RESTART on Client 0x0A0
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:enter connecting state
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: restart connecting
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting RX_REQ on Client 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E: authenticating state ent
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:connecting authenticating
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] Posting AUTH_START for 0x0A00000E
[dot1x]: [16498]: UUID: 0, ra: 0 (debug): [0040.E93E.0000:Gi1/0/2] 0x0A00000E:entering request state
```

Additional Information

Paramètres par défaut

Fonctionnalité	Paramètres par défaut
État d'activation du commutateur 802.1x	Désactivé.
État d'activation 802.1x par port	Désactivé (autorisé de force). Le port envoie et reçoit le trafic normal sans authentification 802.1x du client.
AAA	Désactivé.
serveur RADIUS • Adresse IP	• Aucun spécifié. • 1645.

Fonctionnalité	Paramètres par défaut
• Port d'authentification UDP • Port de comptabilité par défaut • Key (Clé)	• 1646. • Aucun spécifié.
Mode hôte	Mode hôte unique.
Direction de contrôle	Contrôle bidirectionnel.
Réauthentification périodique	Désactivé.
Nombre de secondes entre les tentatives de réauthentification	3600 secondes.
Numéro de réauthentification	Deux fois (nombre de fois où le commutateur redémarre le processus d'authentification avant que le port ne passe à l'état non autorisé).
Période de silence	60 secondes (nombre de secondes pendant lesquelles le commutateur reste à l'état silencieux après un échec de l'échange d'authentification avec le client).
Temps de retransmission	30 secondes (nombre de secondes pendant lesquelles le commutateur attend une réponse à une requête EAP/trame d’identité du client avant de renvoyer la requête).
Nombre maximal de retransmissions	Deux fois (nombre de fois où le commutateur envoie une trame de demande/identité EAP avant de redémarrer le processus d’authentification).
Délai d'attente client	30 secondes (lors du relais d'une requête du serveur d'authentification au client, durée pendant laquelle le commutateur attend une réponse avant de renvoyer la requête au client).

Fonctionnalité	Paramètres par défaut
Délai d'attente du serveur d'authentification	30 secondes (lors du relais d'une réponse du client au serveur d'authentification, durée pendant laquelle le commutateur attend une réponse avant de renvoyer la réponse au serveur). Vous pouvez modifier ce délai d'attente à l'aide de la commande de configuration d'interface dot1x timeout server-timeout.
Délai d'inactivité	Désactivé.
VLAN invité	Aucun spécifié.
Contournement d'authentification inaccessible	Désactivé.
VLAN restreint	Aucun spécifié.
Mode Authenticator (commutateur)	Aucun spécifié.
Contournement d'authentification MAC	Désactivé.
Sécurité sensible à la voix	Désactivé.

Paramètres facultatifs

Réauthentification périodique :

Vous pouvez activer la réauthentification périodique du client 802.1x et spécifier la fréquence à laquelle elle se produit :

- authentication odic : active la réauthentification périodique du client.
- inactivity - Intervalle en secondes après lequel, si aucune activité n'est générée par le client, elle n'est pas autorisée

- reauthenticate - Durée en secondes après laquelle une tentative de réauthentification automatique est lancée.
- restart value : intervalle en secondes après lequel une tentative d'authentification d'un port non autorisé est effectuée.
- valeur non autorisée : intervalle en secondes après lequel une session non autorisée est supprimée.

```
authentication periodic
authentication timer {{[inactivity | reauthenticate | restart | unauthorized]} {value}}
```

Modes de violation :

Vous pouvez configurer un port 802.1x de sorte qu'il s'arrête, génère une erreur syslog ou rejette les paquets d'un nouveau périphérique lorsqu'un périphérique se connecte à un port 802.1x ou lorsque le nombre maximal autorisé d'informations sur les périphériques a été authentifié sur le port.

- shutdown - Erreur de désactivation du port.
- restrict - Générer une erreur syslog.
- protect - Supprime les paquets de tout nouveau périphérique qui envoie du trafic au port.
- replace - Supprime la session en cours et s'authentifie auprès du nouvel hôte.

```
authentication violation {shutdown | restrict | protect | replace}
```

Modification de la période de silence :

La commande de configuration d'interface authentication timer restart contrôle la période d'inactivité, qui dicte la période de temps définie où le commutateur reste inactif après qu'un commutateur ne peut pas authentifier le client. La plage de la valeur est comprise entre 1 et 65535 secondes.

```
authentication timer restart {seconds}
```

Modification du délai de retransmission du commutateur au client :

Le client répond à la trame de demande/identité EAP du commutateur par une trame de réponse/identité EAP. Si le commutateur ne reçoit pas cette réponse, il attend une période définie (appelée temps de retransmission), puis renvoie la trame.

```
authentication timer reauthenticate {seconds}
```

Définition du numéro de retransmission de trame du commutateur au client :

Vous pouvez modifier le nombre de fois que le commutateur envoie une trame de demande/identité EAP (en supposant qu'aucune réponse ne soit reçue) au client avant de redémarrer le processus d'authentification. La plage est comprise entre 1 et 10.

```
dot1x max-reauth-req {count}
```

Configuration du mode hôte :

Vous pouvez autoriser plusieurs hôtes (clients) sur un port 802.1x autorisé.

- multi-auth - Autorise plusieurs clients authentifiés à la fois sur le VLAN voix et le VLAN données.
- multi-host - Autorise plusieurs hôtes sur un port 802.1x autorisé après l'authentification d'un hôte unique.
- multidomaine - Permet à la fois à un hôte et à un périphérique vocal, tel qu'un téléphone IP (Cisco ou non Cisco), d'être authentifiés sur un port IEEE 802.1x autorisé.

```
authentication host-mode [multi-auth | multi-domain | multi-host | single-host]
```

Activation du déplacement MAC :

Le déplacement MAC permet à un hôte authentifié de passer d’un port du périphérique à un autre.

```
authentication mac-move permit
```

Activation du remplacement MAC :

Le remplacement MAC permet à un hôte de remplacer un hôte authentifié sur un port.

- protect - Le port abandonne les paquets avec des adresses MAC inattendues sans générer de message système.
- restrict - Les paquets en violation sont abandonnés par le processeur et un message système est généré.
- shutdown - Le port est désactivé en raison d'une erreur lorsqu'il reçoit une adresse MAC inattendue.

```
authentication violation {protect | replace | restrict | shutdown}
```

Définition du numéro de réauthentification :

Vous pouvez également modifier le nombre de redémarrages du processus d'authentification par le périphérique avant que le port ne passe à l'état non autorisé. La plage est comprise entre 0 et 10.

```
dot1x max-req {count}
```

Configuration d'un VLAN invité :

Lorsque vous configurez un VLAN invité, les clients qui ne sont pas compatibles 802.1x sont placés dans le VLAN invité lorsque le serveur ne reçoit pas de réponse à sa requête EAP/trame d'identité.

```
authentication event no-response action authorize vlan {vlan-id}
```

Configuration d'un VLAN restreint :

Lorsque vous configurez un VLAN restreint sur un périphérique, les clients conformes à la norme IEEE 802.1x sont déplacés vers le VLAN restreint lorsque le serveur d'authentification ne reçoit pas de nom d'utilisateur et de mot de passe valides.

```
authentication event fail action authorize vlan {vlan-id}
```

Configuration du nombre de tentatives d'authentification sur un VLAN restreint :

Vous pouvez configurer le nombre maximal de tentatives d'authentification autorisées avant qu'un utilisateur soit affecté au VLAN restreint en utilisant la commande de configuration d'interface authentication event fail retry retry count. La plage de tentatives d'authentification autorisées est comprise entre 1 et 3.

```
authentication event fail retry {retry count}
```

Configuration du contournement d'authentification 802.1x inaccessible avec le VLAN voix critique :

Vous pouvez configurer un VLAN voix critique sur un port et activer la fonctionnalité de contournement d'authentification inaccessible.

- allow - Déplace tous les nouveaux hôtes essayant de s'authentifier sur le VLAN critique spécifié par l'utilisateur.
- reinitialize - Déplace tous les hôtes autorisés sur le port vers le VLAN critique spécifié par l'utilisateur.

```
authentication event server dead action {authorize | reinitialize} vlan vlan-id]
authentication event server dead action authorize voice
```

Configuration de l'authentification 802.1x avec WoL :

Vous pouvez activer l'authentification 802.1x avec Wake on LAN (WOL).

```
authentication control-direction both
```

Configuration du contournement d'authentification MAC :

```
mab
```

Configuration de la commande d'authentification flexible :

```
authentication order [ dot1x | mab ] | {webauth}
authentication priority [ dot1x | mab ] | {webauth}
```

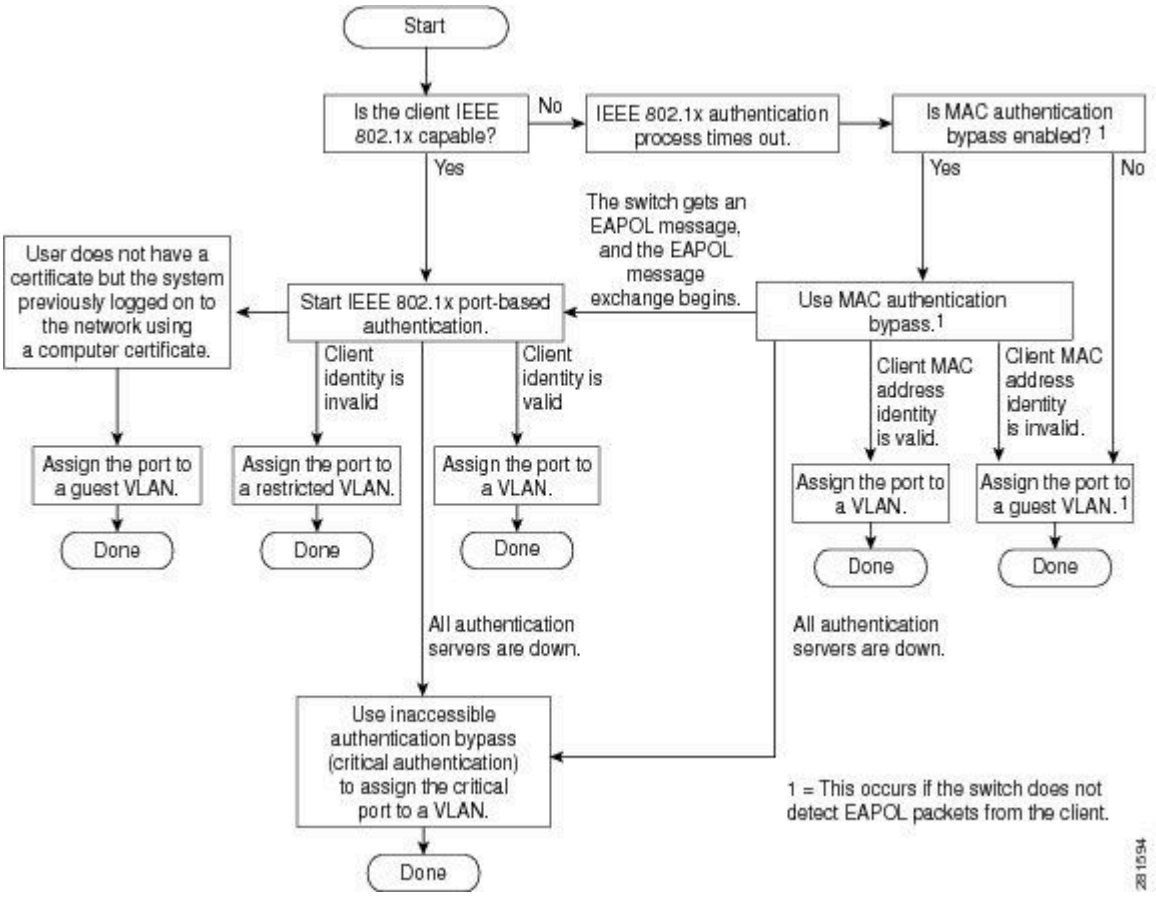
Configuration de la sécurité 802.1x sensible à la voix :

Vous utilisez la fonction de sécurité 802.1x sensible à la voix sur le périphérique pour désactiver uniquement le VLAN sur lequel une violation de sécurité se produit, qu'il s'agisse d'un VLAN de données ou d'un VLAN voix. Une violation de sécurité trouvée sur le VLAN de données entraîne l'arrêt du seul VLAN de données. Il s'agit d'une configuration globale.

```
errdisable detect cause security-violation shutdown vlan
errdisable recovery cause security-violation
```

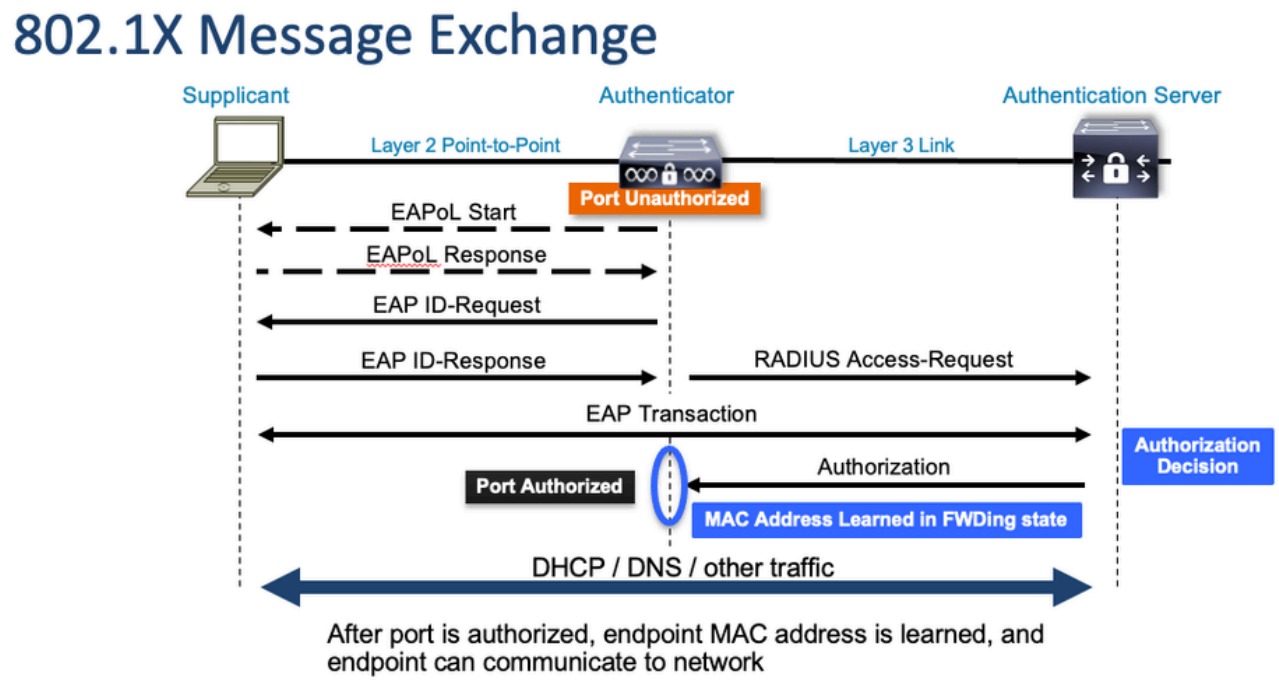
Organigrammes

Organigramme d'authentification :



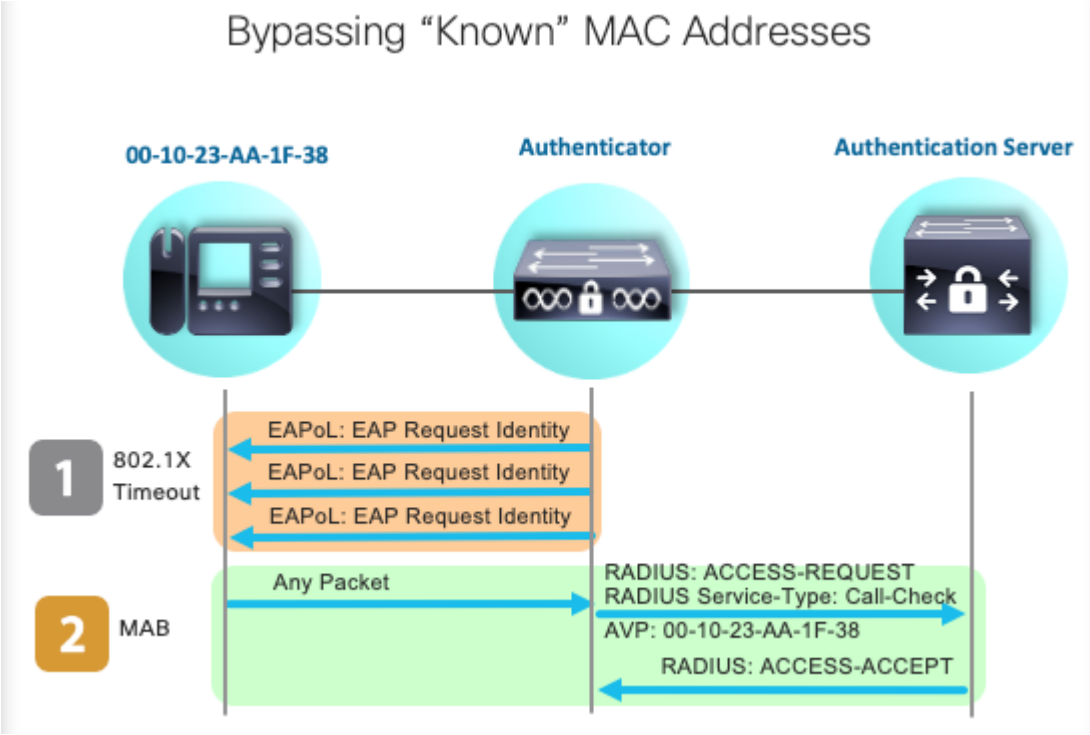
Initiation de l'authentification basée sur les ports et échange de messages :

Cette image montre le client initiant l'échange de messages avec le serveur RADIUS.



Initiation de l'authentification MAB et échange de messages :

Cette image montre l'échange de messages pendant le contournement d'authentification MAC (MAB).



Informations connexes

- Démystification des configurations de serveur RADIUS
- Guide de déploiement de contournement d'authentification MAC
- Guide de déploiement de la norme 802.1x filaire
- Guide de configuration SPAN du Catalyst 9300
- Guide de configuration EPC du Catalyst 9300
- Assistance technique de Cisco et téléchargements

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.