ISE 3.4 PACを使用しない、ISEとNAD間のTrustSec用の認証の設定

内容

はじめに

このドキュメントでは、TrustSec環境のデータをダウンロードするための、ISEとNADクライアント間のPACレス設定の初期設定について説明します。

前提条件

要件

- ネットワークセキュリティソリューションとしてのCisco TrustSecに精通していること。
- ネットワークセキュリティを管理するためのIdentity Services Engine(ISE)に関する知識
- 認証情報を転送するためのフレームワークとしての拡張可能認証プロトコル(EAP)の基本的な知識。

使用するコンポーネント

Identity Services Engine(ISE)リリース3.4.x

Cisco IOS® 17.15.1以降

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

情報

PACレスモードでは、TrustSecポリシーを実装する方が簡単です。これは、デバイスとIdentity Services Engine(ISE)間のセキュア通信に通常必要なProtected Access Credential(PAC)が不要であるためです。 このアプローチは、複数のISEノードがある環境で特に有効です。プライマリノードがオフラインになると、デバイスはクレデンシャルを再確立することなくバックアップに自動的に切り替えることができるため、中断が減少します。PACレス認証は、プロセスを簡素化し、拡張性と使いやすさを向上させ、ゼロトラスト原則に沿った最新のセキュリティ方式をサポートします。

このモードでは、デバイスはユーザ名とパスワードを含む要求を送信することで開始します。ISEがセキュアセッションを提案して応答します。このセッションを設定すると、ISEはセキュアな通信に必要な重要な情報を提供します。これには、セキュリティのキーと、サーバIDやタイミングなどの詳細が含まれます。この情報は、必要なポリシーとデータへの安全で継続的なアクセスを確保するために使用されます。

設定

コンフィギュレーション

スイッチの設定

このドキュメントでは、Cisco C9300スイッチを使用してPACレス認証の設定を行います。バージョン17.15.1以降を実行しているスイッチは、Identity Services Engine(ISE)を使用してPACなしの認証を実行できます。

ステップ1：スイッチのconfiguration terminalの下にあるスイッチで、RADIUSサーバとRADIUSグループを設定します。

RADIUS サーバ:

```
radius server

 address ipv4

        auth-port 1812 acct-port 1813

 key
```

RADIUSグループ:

```
aaa group server radius trustsec
 server name
```

ステップ2:PACレス認証のcts認証およびdot1xにRADIUSサーバグループをマッピングします。

CTSマッピング：

<#root>

```
cts authorization list
```

cts-mlist

```
   // cts-mlist is the name of the authorization list
```

Dot1x認証：

<#root>

```
aaa authentication dot1x default group
```

```
aaa authorization network
```

cts-mlist

```
 group
```

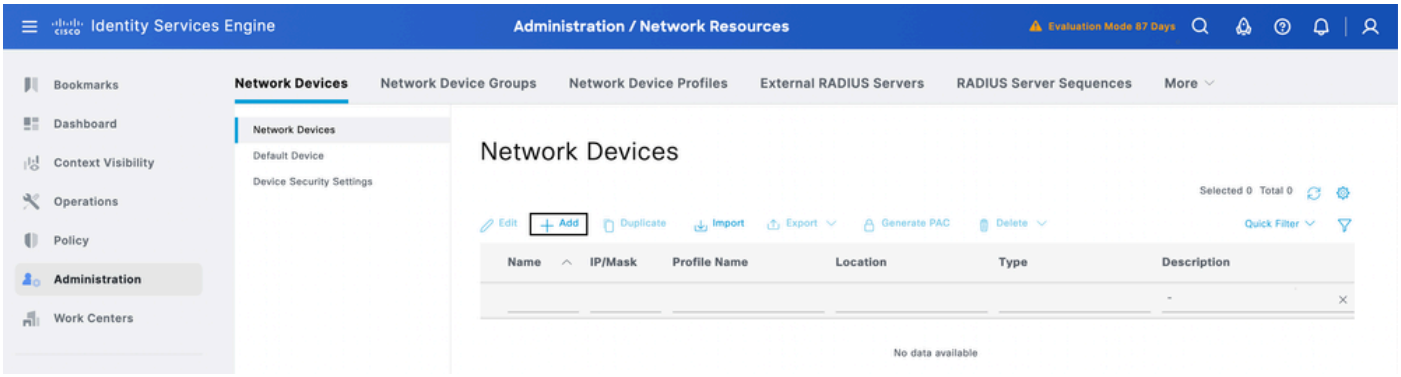
ステップ3：スイッチのイネーブルモードでCTS-IDとパスワードを設定します

```
cts credentials id

        password
```

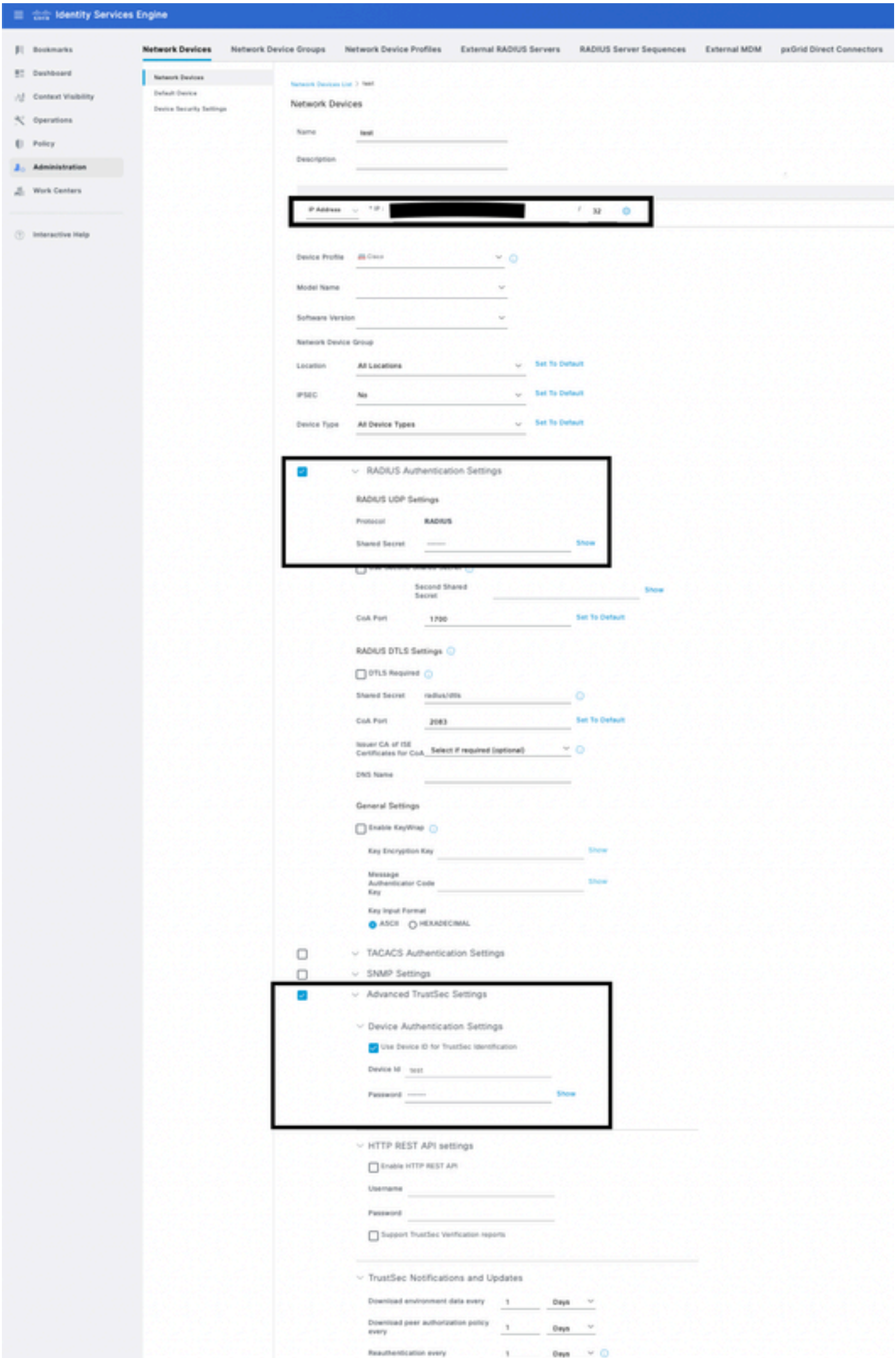
ISE 設定

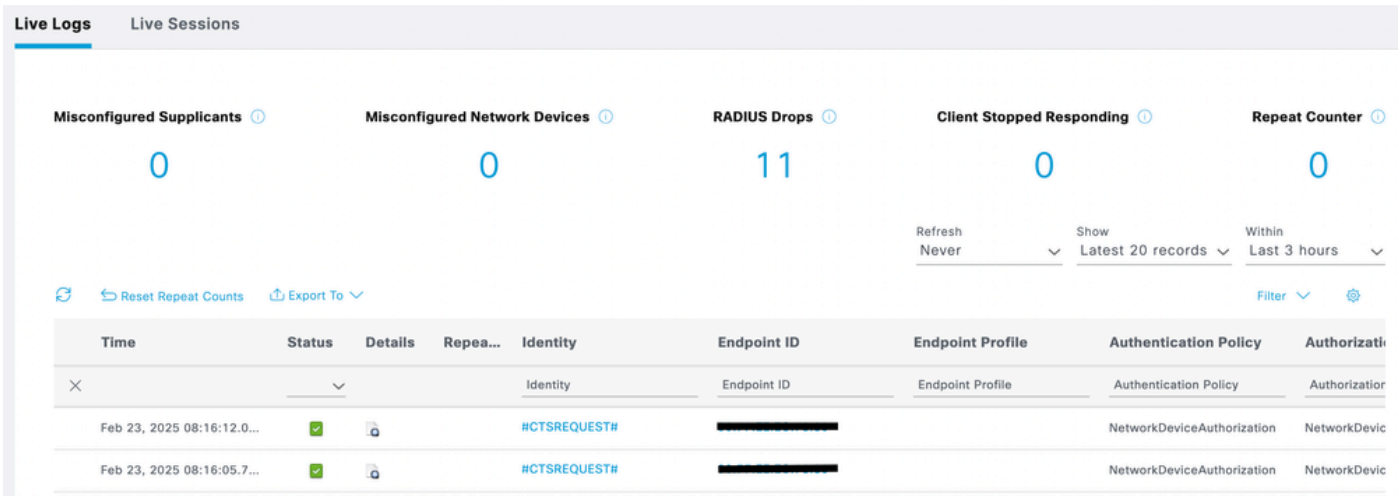
1. ISEで、Administration > Network Resources > Network Devices > Network Devicesの順に選択し、ネットワークデバイスを設定します。addをクリックして、スイッチをISEサーバに追加します。

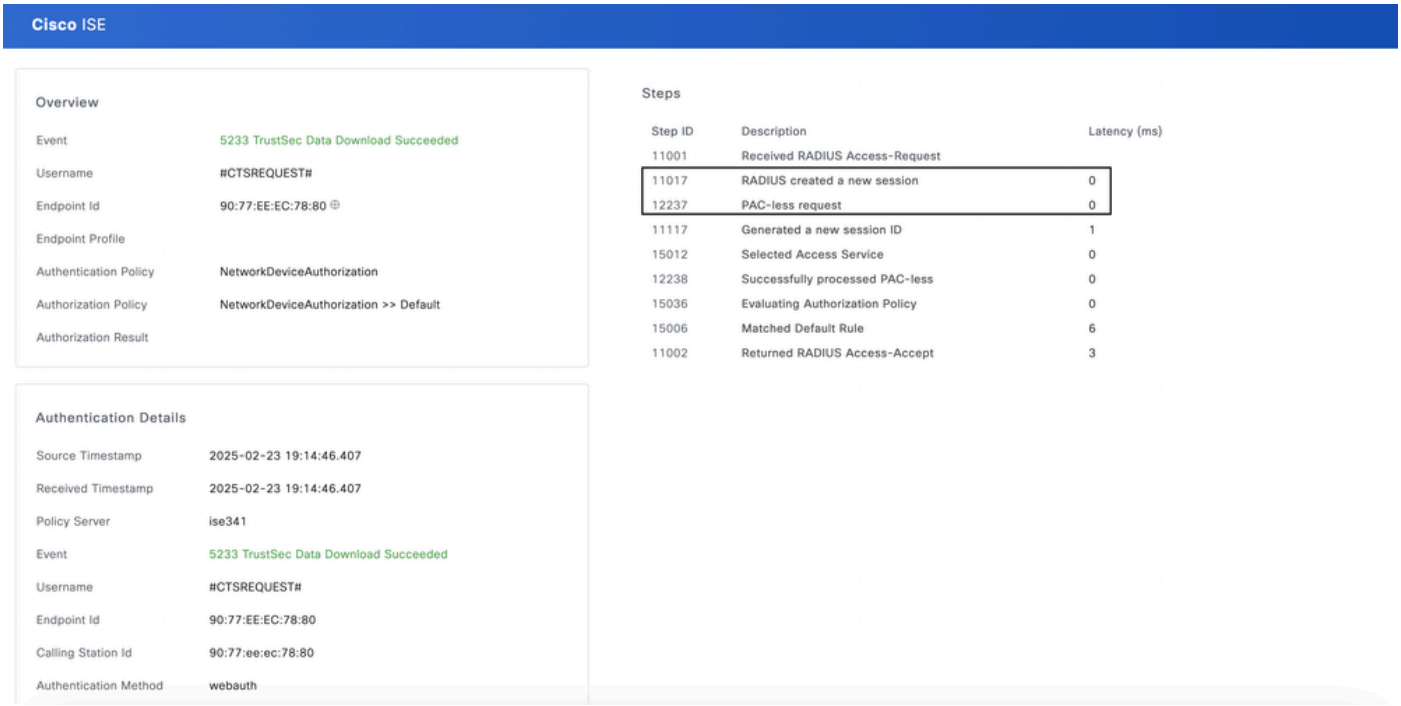


2. スイッチからのTrustSec認証のRADIUS要求を処理するには、ISEのIPアドレスフィールドにNAD IPアドレスを追加します。
3. NADクライアントのRADIUS認証設定を有効にし、RADIUS共有秘密キーを入力します。

4. Advanced Trustsec Settingsを有効にし、CTS-IDでデバイス名を更新し、コマンドでパスワード（ cts credentials id <CTS-ID> password <パスワード> ）を使用してパスワードフィールドを更新します。


Identity Services Engine
Bookmarks
Dashboard
Context Visibility
Operations
Policy
Administration
Work Centers
Interactive Help
Network Devices
Network Device Groups
Network Device Profiles
External RADIUS Servers
RADIUS Server Sequences
External MDM
pxGrid Direct Connectors
Network Devices
Default Device
Device Security Settings
Network Devices List > test
Network Devices
Name
test
Description
IP Address
[redacted]
32
Device Profile
Model Name
Software Version
Network Device Group
Location
All Locations
Set To Default
IPSEC
No
Set To Default
Device Type
All Device Types
Set To Default
RADIUS Authentication Settings
RADIUS UDP Settings
Protocol
RADIUS
Shared Secret
Show
Second Shared Secret
Show
CoA Port
1700
Set To Default
RADIUS DTLS Settings
DTLS Required
Shared Secret
radius/dtls
CoA Port
2083
Set To Default
Issuer CA of ISE Certificates for CoA
Select if required (optional)
DNS Name
General Settings
Enable KeyWrap
Key Encryption Key
Show
Message Authenticator Code Key
Show
Key Input Format
ASCII
HEXADECIMAL
TACACS Authentication Settings
SNMP Settings
Advanced TrustSec Settings
Device Authentication Settings
Use Device ID for TrustSec Identification
Device Id
test
Password
Show
HTTP REST API settings
Enable HTTP REST API
Username
Password
Support TrustSec Verification reports
TrustSec Notifications and Updates
Download environment data every
1
Days
Download peer authorization policy every
1
Days
Reauthentication every
1
Days





トラブルシュート

問題をトラブルシューティングするには、スイッチで次のデバッグを実行します。

```
Debug Command:

debug cts environment-data all
debug cts env
debug cts aaa
debug radius
debug cts ifc events


debug cts authentication details
```

```
debug cts authorization all debug
```

デバッグスニペット：

*Feb 23 14:48:14.974: CTS env-data: Force environment-data refresh bitmask 0x2

*Feb 23 14:48:14.974: CTS env-data: download transport-type = CTS_TRANSPORT_IP_UDP

*2月23日14:48:14.974:cts_env_data COMPLETE：状態env_data_completeの間、イベント0(env_data_request)を取得

*Feb 23 14:48:14.974: @@@ cts_env_data COMPLETE: env_data_complete -> env_data_waiting_rsp

*Feb 23 14:48:14.974:env_data_waiting_rsp_enter: state = WAITING_RESPONSE

*Feb 23 14:48:14.974: Secure Key is present on the device, proceed with pac-less env-data download // initiate the PAC-Less authentication from switch

*Feb 23 14:48:14.974: cts_aaa_is_fragmented: (CTS env-data SM)NOT-FRAG attr_q(0)

*Feb 23 14:48:14.974:env_data_request_action: state = WAITING_RESPONSE

*2月23日14:48:14.974:env_data_download_complete:

  status(FALSE)、req(x0)、rec(x0)

*Feb 23 14:48:14.974:status(FALSE)、req(x0)、rec(x0)、expect(x81)、

  wait_for_server_list(x85)、wait_for_multicast_SGT(xB5)、wait_for_SGName_mapping_tbl(x1485)、

  wait_for_SG-EPG_tbl(x18085)、wait_for_default_EPG_tbl(xC0085)、wait_for_default_SGT_tbl(x600085) wait_for_default_SERVICE_ENTRY_tbl(xC000085)

*Feb 23 14:48:14.974: env_data_request_action: state = WAITING_RESPONSE, received = 0x0要求= 0x0

*Feb 23 14:48:14.974: cts_env_data_aaa_req_setup : aaa_id = 15

*Feb 23 14:48:14.974: cts_aaa_req_setup: (CTS env-data SM)プライベートグループがDEADと表示され、パブリックグループを試行

*Feb 23 14:48:14.974: cts_aaa_attr_add: AAA要求(0x7AB57A6AA2C0)

*Feb 23 14:48:14.974: username = #CTSREQUEST#

*Feb 23 14:48:14.974: AAA Context Add Attribute: (CTS env-data SM)attr(test)

*Feb 23 14:48:14.974:cts-environment-data =テスト

*Feb 23 14:48:14.974: cts_aaa_attr_add: AAA要求(0x7AB57A6AA2C0)

*Feb 23 14:48:14.974: AAA Context Add Attribute: (CTS env-data SM)attr(env-data-fragment)

*Feb 23 14:48:14.974:cts-device-capability = env-data-fragment（データフラグメント）

*Feb 23 14:48:14.974: cts_aaa_attr_add: AAA要求(0x7AB57A6AA2C0)

*Feb 23 14:48:14.975: AAA Context Add Attribute: (CTS env-data SM)attr(multiple-server-ip-supported)

*Feb 23 14:48:14.975: cts-device-capability = multiple-server-ip-supported

*Feb 23 14:48:14.975: cts_aaa_attr_add: AAA要求(0x7AB57A6AA2C0)

*Feb 23 14:48:14.975: AAA Context Add属性： (CTS env-data SM)attr(wnlx)

*2月23日14:48:14.975:clid = wnlx

*Feb 23 14:48:14.975: cts_aaa_req_send: AAA req(0x7AB57A6AA2C0)がAAAに正常に送信されました。

*Feb 23 14:48:14.975: RADIUS/ENCODE(0000000F)：元のコンポーネントタイプ= CTS

*Feb 23 14:48:14.975: RADIUS(0000000F): Config NAS IP: 0.0.0.0

*Feb 23 14:48:14.975: vrfid: [65535] ipv6 tableid : [0]

*Feb 23 14:48:14.975: idb is NULL

*Feb 23 14:48:14.975: RADIUS(0000000F): Config NAS IPv6: ::

*Feb 23 14:48:14.975: RADIUS/ENCODE(0000000F): acct_session_id: 4003

*Feb 23 14:48:14.975: RADIUS(0000000F): sending

*Feb 23 14:48:14.975: RADIUS: PACなしモード、シークレットあり

*Feb 23 14:48:14.975: RADIUS: Successfully added CTS pacless attribute to the radius request

*Feb 23 14:48:14.975: RADIUS/ENCODE: Best Local IP-Address 10.127.196.234 for Radius-Server 10.127.196.169

*Feb 23 14:48:14.975: RADIUS: PACなしモード、シークレットあり

*Feb 23 14:48:14.975: RADIUS(0000000F): Send Access-Request to 10.127.196.169:1812 id 1645/11, len 249 //スイッチからのRadius Access Request

RADIUS:オーセンティケータ78 8A 70 5C E5 D3 DD F1 - B4 82 57 E2 1F 95 3B 92

*Feb 23 14:48:14.975: RADIUS: User-Name [1] 14 "#CTSREQUEST#"

*Feb 23 14:48:14.975: RADIUS: Vendor, Cisco [26] 33

*Feb 23 14:48:14.975: RADIUS: Cisco AVpair [1] 27 「cts-environment-data=test」

*Feb 23 14:48:14.975: RADIUS: Vendor, Cisco [26] 47

*Feb 23 14:48:14.975: RADIUS: Cisco AVpair [1] 41 「cts-device-capability=env-data-fragment」

*Feb 23 14:48:14.975: RADIUS: Vendor, Cisco [26] 58

*Feb 23 14:48:14.975: RADIUS: Cisco AVpair [1] 52 "cts-device-capability=multiple-server-ip-supported"

*Feb 23 14:48:14.975: RADIUS: User-Password [2] 18 *

*Feb 23 14:48:14.975: RADIUS: Calling-Station-Id [31] 8 「wnlx」

*Feb 23 14:48:14.975: RADIUS: Service-Type [6] 6 Outbound [5]

*Feb 23 14:48:14.975: RADIUS: NAS-IP-Address [4] 6 10.127.196.234

*Feb 23 14:48:14.975: RADIUS: Vendor, Cisco [26] 39

*Feb 23 14:48:14.975: RADIUS: Cisco AVpair [1] 33 「cts-pac-capability=cts-pac-less」 // CTS PAC Less cv-pair属性は、PACレス認証のパケットを処理するISEの要求に追加されます。

*Feb 23 14:48:14.975: RADIUS(0000000F): Sending a IPv4 Radius Packet

*Feb 23 14:48:14.975: RADIUS(0000000F): Started 5 sec timeout

*Feb 23 14:48:14.990: RADIUS: Received from id 1645/11 10.127.196.169:1812, Access-Accept, len 313. // Authentication success

RADIUS:オーセンティケータ92 4C 21 5C 99 28 64 8B - 23 06 4B 87 F6 FF 66 3C

*Feb 23 14:48:14.990: RADIUS: User-Name [1] 14 "#CTSREQUEST#"

*Feb 23 14:48:14.990: RADIUS: Class [25] 78

半径：43 41 43 53 3A 30 61 37 66 63 34 61 39 54 37 68 [CACS:0a7fc4a9T7h]

半径：39 79 44 42 70 2F 7A 6A 64 66 66 56 49 55 74 4D [9yDBp/zjdffVIUtM]

半径：78 34 68 63 50 4C 4A 45 49 76 75 79 51 62 4C 70 [x4hcPLJEIvuyQbLp]

半径：31 48 7A 35 50 45 39 38 3A 69 73 65 33 34 31 2F [1Hz5PE98:ise341/]

半径： 35 32 39 36 36 39 30 32 31 2F 32 31 [529669021/21]

*Feb 23 14:48:14.990: RADIUS: Vendor, Cisco [26] 39

*Feb 23 14:48:14.990: RADIUS: Cisco AVpair [1] 33 「cts-pac-capability=cts-pac-less」

*Feb 23 14:48:14.990: RADIUS: Vendor, Cisco [26] 43

*Feb 23 14:48:14.991: RADIUS: Cisco AVpair [1] 37 「cts:server-list=CTSServerList1-0001」

*Feb 23 14:48:14.991: RADIUS: Vendor, Cisco [26] 38

*Feb 23 14:48:14.991: RADIUS: Cisco AVpair [1] 32 「cts:security-group-tag=0002-00」

*Feb 23 14:48:14.991: RADIUS: Vendor, Cisco [26] 41

*Feb 23 14:48:14.991: RADIUS: Cisco AVpair [1] 35 「cts:environment-data-expiry=86400」

*Feb 23 14:48:14.991: RADIUS: Vendor, Cisco [26] 40

*Feb 23 14:48:14.991: RADIUS: Cisco AVpair [1] 34 "cts:security-group-table=0001-17"

*Feb 23 14:48:14.991: RADIUS: PACなしモード、シークレットあり

*Feb 23 14:48:14.991: RADIUS(0000000F): Received from id 1645/11

*Feb 23 14:48:14.991: cts_aaa_callback: (CTS env-data SM)AAA req(0x7AB57A6AA2C0)response success

*Feb 23 14:48:14.991: AAA CTX FRAG CLEAN:(CTS env-data SM)attr(test)

*Feb 23 14:48:14.991: AAA CTX FRAG CLEAN:(CTS env-data SM)attr(env-data-fragment)

*Feb 23 14:48:14.991: AAA CTX FRAG CLEAN: (CTS env-data SM)attr(multiple-server-ip-supported)

*Feb 23 14:48:14.991: AAA CTX FRAG CLEAN: (CTS env-data SM)attr(wnlx)

*Feb 23 14:48:14.991: AAA属性：不明なタイプ(450)。

*Feb 23 14:48:14.991: AAA属性：不明なタイプ(1324)。

*Feb 23 14:48:14.991: AAA属性： server-list = CTSServerList1-0001。

*Feb 23 14:48:14.991: Received SLIST name.cts_is_slist_send_to_binos_reqをFALSEに設定

*Feb 23 14:48:14.991: AAA属性： security-group-tag = 0002-00

*Feb 23 14:48:14.991: AAA属性： environment-data-expiry = 86400。

*Feb 23 14:48:14.991: AAA属性： security-group-table = 0001-17.CTS env-data: Receiving AAA attributes.　//環境データをダウンロードしています

CTS_AAA_SLIST（オプション）

1回目のAccess-Acceptで受信されたslist名(CTSServerList1)

slist名(CTSServerList1)が存在します

CTS_AAA_SECURITY_GROUP_TAG（オプション）

CTS_AAA_ENVIRONMENT_DATA_EXPIRY = 86400です。

CTS_AAA_SGT_NAME_リスト

テーブル(0001)を1回目のアクセス許可で受信しました

変更がないため、テーブル(0001)をインストール済みから受信済みにコピーします。

新しい名前(0001), gen(17)

CTS_AAA_DATA_END（オプション）

*2月23日14:48:14.991:cts_env_data WAITING_RESPONSE：状態env_data_waiting_rsp中、イベント1(env_data_received)を取得

*Feb 23 14:48:14.991: @@@ cts_env_data WAITING_RESPONSE: env_data_waiting_rsp -> env_data_assessing

*Feb 23 14:48:14.991: env_data_assessing_enter: state = ASSESSING

*Feb 23 14:48:14.991: cts_aaa_is_fragmented: (CTS env-data SM)NOT-FRAG attr_q(0)

*Feb 23 14:48:14.991: env_data_assessing_action: state = ASSESSING

*Feb 23 14:48:14.991: env_data_download_complete:

status(FALSE)、req(x81)、rec(xC87)

*Feb 23 14:48:14.991: Expect same as received

*Feb 23 14:48:14.991:ステータス(TRUE)、要求(x81)、rec(xC87)、予測(x81)、

wait_for_server_list(x85)、wait_for_multicast_SGT(xB5)、wait_for_SGName_mapping_tbl(x1485)、

wait_for_SG-EPG_tbl(x18085)、wait_for_default_EPG_tbl(xC0085)、wait_for_default_SGT_tbl(x600085) wait_for_default_SERVICE_ENTRY_tbl(xC000085)

*Feb 23 14:48:14.991: cts_env_data ASSESSING：状態中env_data_assessing、got event 4(env_data_complete)

*Feb 23 14:48:14.991: @@@ cts_env_data ASSESSING: env_data_assessing -> env_data_complete

*Feb 23 14:48:14.991:env_data_complete_enter: state = COMPLETE

*Feb 23 14:48:14.991: CTS-ifc-ev: env data reporting to core, result: Successful

*Feb 23 14:48:14.991: env_data_install_action: state = COMPLETE completed.types 0x0

*Feb 23 14:48:14.991: env_data_install_action: clean installed sgt<->sgname table

*Feb 23 14:48:14.991: Cleaning up installed sg-epg list

*Feb 23 14:48:14.991: Cleaning up installed default epg list

*Feb 23 14:48:14.991: env_data_install_action: mcast_sgtテーブルの更新

*Feb 23 14:48:14.991:Env data sync to standby status 2

*Feb 23 14:48:14.991: SLISTは前回の更新と同じです。BINOSに送信する必要はありません

*Feb 23 14:48:14.991: CTS-sg-epg-events:setting default_sg 0 to env data

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。