Understand EAP-FAST and Chaining Implementations on AnyConnect NAM and ISE

Contents

Introduction

This document describes details regarding EAP-FAST implementation on Cisco AnyConnect Network Access Manager (NAM) and Identity Services Engine (ISE).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Basic knowledge of EAP framework and EAP-FAST methods
- Basic knowledge of Identity Services Engine (ISE)
- Basic knowledge of AnyConnect NAM and Profile Editor
- Basic knowledge of Cisco Catalyst configuration for 802.1x services

Components Used

The information in this document is based on these software versions:

- Windows 7 with Cisco AnyConnect Secure Mobility Client, Release 3.1 and 4.0
- Cisco Catalyst 3750X switch with software 15.2.1 and later
- Cisco ISE, Release 1.4

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. If your network is live, ensure that you understand the potential impact of any command.

Theory

Phases

EAP-FAST is a flexible EAP method which allows mutual authentication of a supplicant and a server. It is similar to EAP-PEAP, but typically does not require the use of client or even server certificates. One advantage of EAP-FAST is the ability to chain multiple authentications (using multiple inner methods) and bind it cryptographically together (EAP Chaining). Cisco implementations use this for user and machine authentications.

EAP-FAST utilizes Protected Access Credentials (PAC) in order to quickly establish the TLS tunnel (session resume) or to authorize the user/machine (skip inner method for authentication).

There are 3 phases for EAP-FAST:

- phase 0 (PAC provisioning)
- phase 1 (TLS tunnel establishment)
- phase 2 (Authentication)

EAP-FAST supports PAC-less and PAC-based conversation. PAC-based consists of PAC provisioning and PAC-based authentication. PAC provisioning can be based on anonymous or authenticated TLS session.

PAC

PAC is Protected Access Credentials generated by the server and provided to client. It consists of:

- PAC key (random secret value, used to derive TLS master and session keys)
- PAC opaque (PAC key + user identity - all encrypted by EAP-FAST server master key)
- PAC info (server identity, TTL timers)

The server issuing the PAC encrypts the PAC key and identity using the EAP-FAST server master key (that is PAC opaque) and sends the whole PAC to the client. It does not keep/store any other information (except master key which is the same for all PACs) .

Once the PAC opaque is received, it is decrypted using the EAP-FAST server master key and validated. The PAC key is used to derive the TLS master and session keys for an abbreviated TLS tunnel.

New EAP-FAST server master keys are generated when the previous master key expires. In some cases, a master key can be revoked.

There are a few types of PACs being used currently:

- Tunnel PAC: used for TLS tunnel establishment (without the need of client or server certificate). Sent in TLS Client Hello
- Machine PAC: used for TLS tunnel establishment and immediate machine authorization. Sent in TLS Client Hello
- User Authorization PAC: used for immediate user authentication (skip inner method) if allowed by server. Sent inside TLS tunnel using TLV.
- Machine Authorization PAC: used for immediate machine authentication (skip inner method) if allowed by server. Sent inside TLS tunnel using TLV.
- Trustsec PAC: used for authorization when performing environmental or policy refresh.

All of those PACs are usually delivered automatically in phase 0. Some of the PACs (Tunnel, Machine, Trustsec) can be also delivered manually.

When PACs are generated

- Tunnel PAC: provisioned after a successful authentication (inner method) if not used previously.
- Authorization PAC: provisioned after successful authentication (inner method) if not used previously.
- Machine PAC: provisioned after successful machine authentication (inner method) if not used previously and when an Authorization PAC is not used. It is provisioned when the Tunnel PAC expires; however, not when the Authorization PAC expires. It is provisioned when EAP-Chaining is enabled or disabled.

Note:

Each PAC provisioning requires successful authentication except in the use case: authorized user asks for the Machine PAC for a machine that does not have an AD account.

This table summarizes provisioning and proactive update functionality:

PAC Type	Tunnel v1/v1a/CTS	Machine	Authorization
Provide PAC on request on provisioning	yes	only on authenticated provisioning	only on authenticated provisioning and if Tunnel PAC is requested also
Provide PAC on request on authentication	yes	yes	only if it was not used in this authentication

Proactive update	yes	no	no
When falling back to PAC provisioning after failed PAC-based authentication (for example, when PAC is expired)	reject and do not provide the new one	reject and do not provide the new one	reject and do not provide the new one
Support ACS 4.x PACs	for Tunnel PAC v1/v1a	yes	no

EAP-FAST Server Master Key ACS 4.x vs ACS 5x and ISE

There is a slight difference in Master key handling when comparing ACS 4.x and ISE

Feature	ACS 4.1.2	ACS 5.x / ISE
Master Key	Master key has TTL, can be active, retired or expired	Master key is automatically generated from seed at every configured period of time. Specific Master Key is always accessible and then never expired
PAC Refresh	PAC update is sent by server when PAC is expired, unless Master Key used for PAC encryption is expired	PAC update is sent by server after first successful authentication that is performed in specific configurable period of time before PAC expiration moment.

In other words, ISE keeps all old master keys and generate a new one by default once per week. As the Master Key cannot expire, only the PAC TTL is validated.

The ISE Master Key generation period is configured from *Administration -> Settings -> Protocol -> EAP-FAST -> EAP-FAST Settings.*

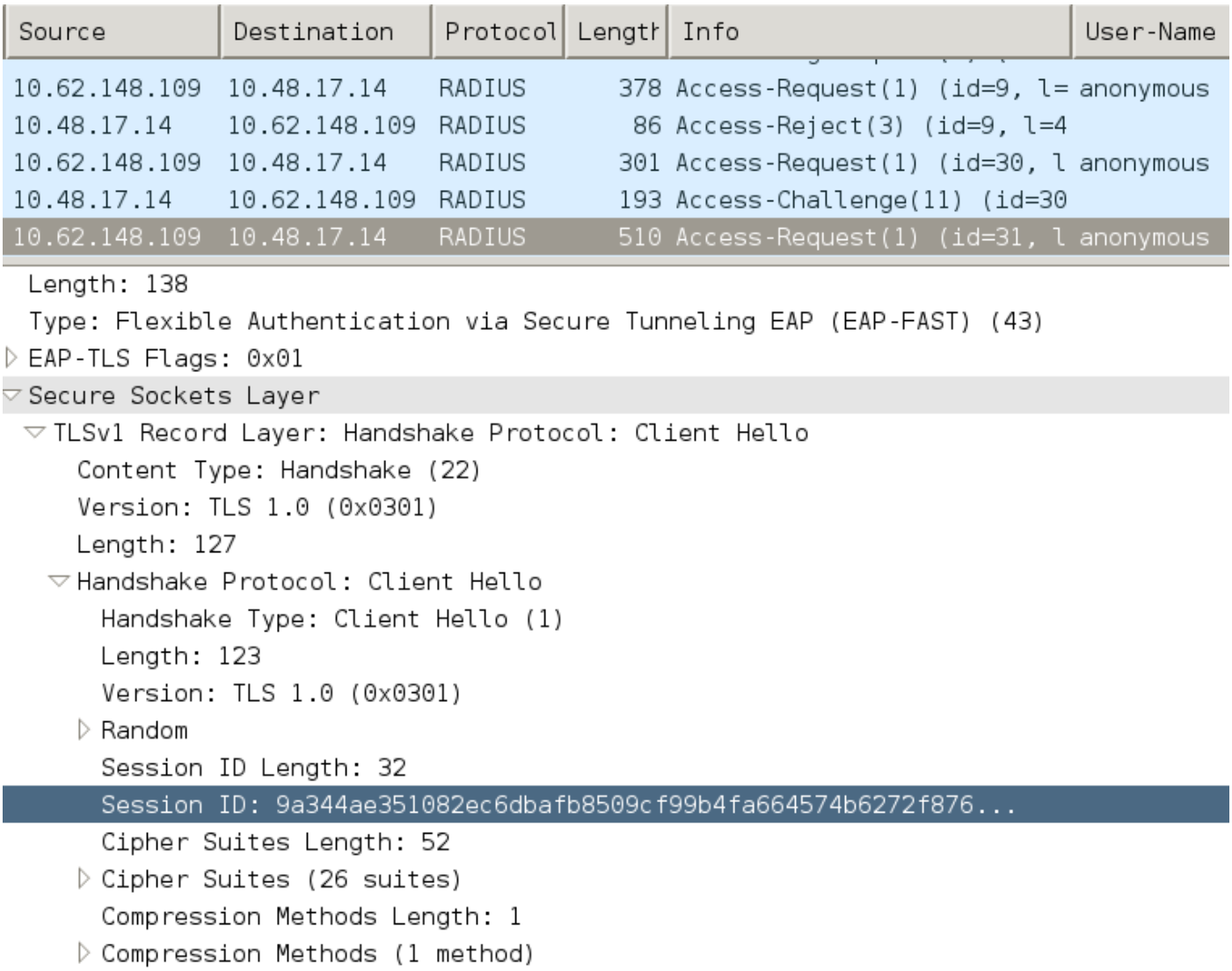
Session Resume

This is an important component allowing for Tunnel PAC usage. It allows for TLS tunnel renegotiation without usage of certificates.

There are two session resume types for EAP-FAST: Server state based and stateless (PAC based).

Server State

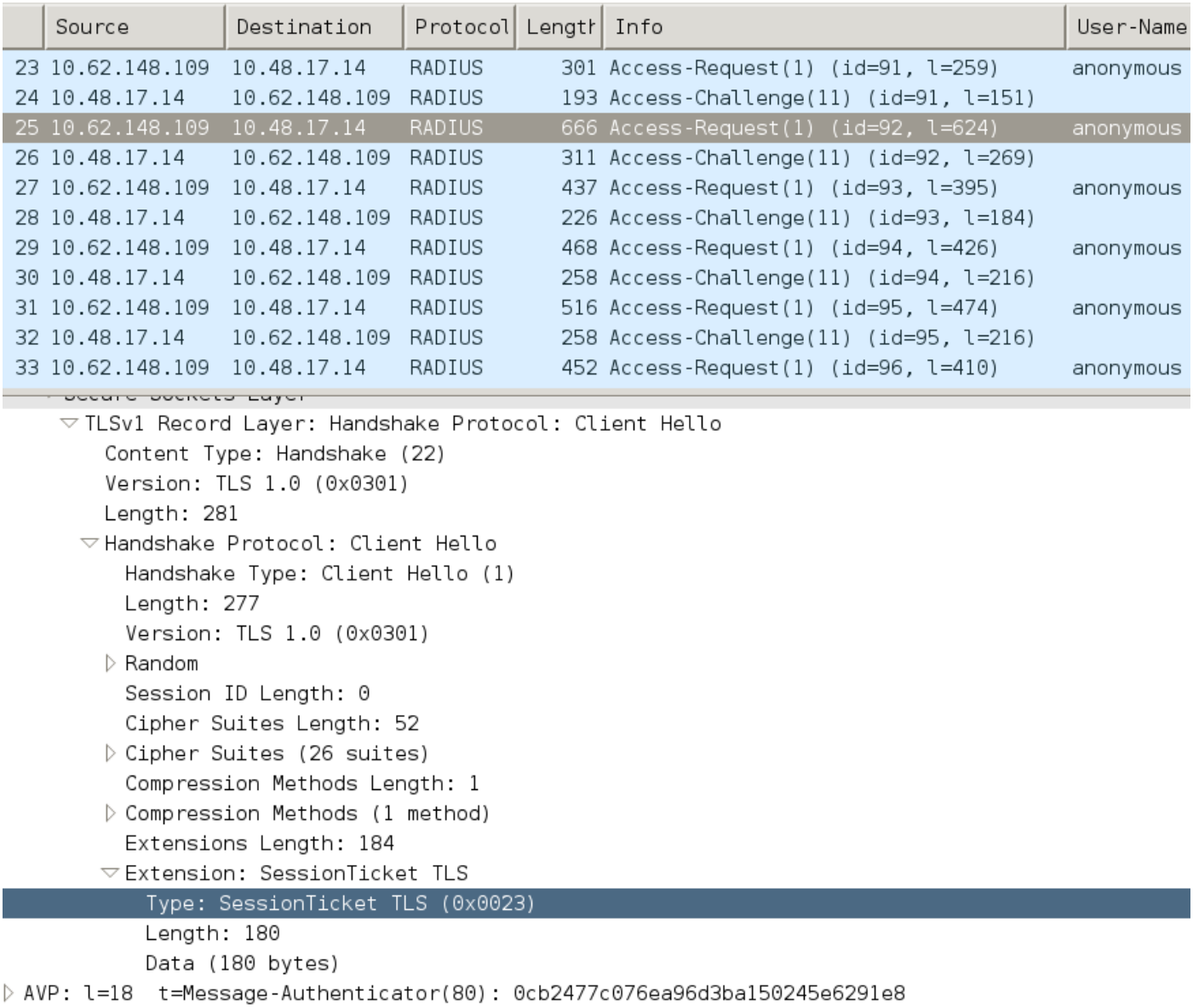
Standard TLS based method is based on the TLS SessionID cached on the server. The client sending the TLS Client Hello attaches the SessionID in order to resume the session. The session is only used for PAC provisioning when using an anonymous TLS tunnel:



Stateless (PAC based)

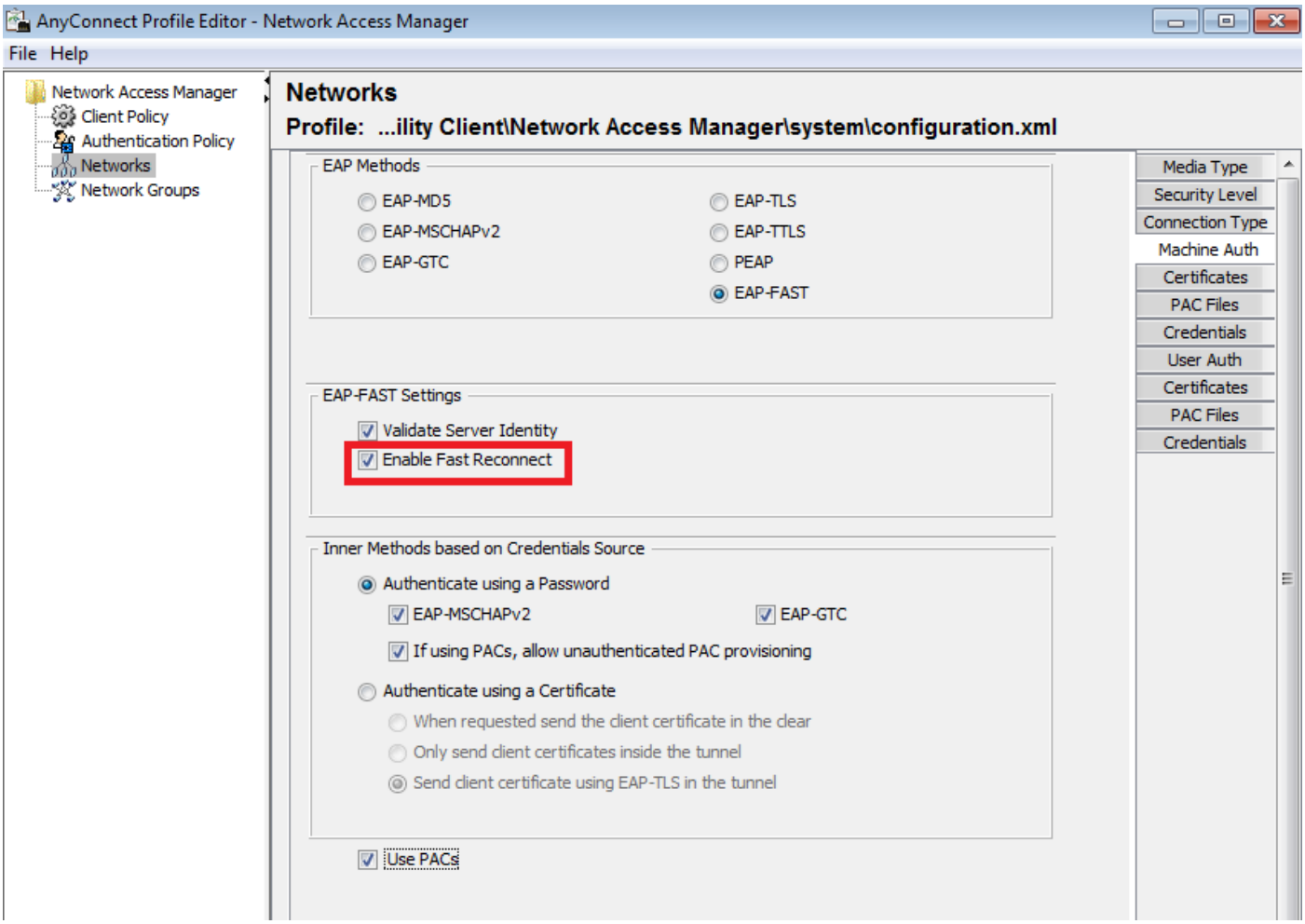
User/Machine Authorization PAC is used to store the previous authentication and authorization states for the peer.

Client side resume is based on RFC 4507. The server does not need to cache any data; instead the client attaches the PAC in the TLS Client Hello SessionTicket extension. In turn, the PAC is validated by the server. Example based on Tunnel PAC delivered to the server:



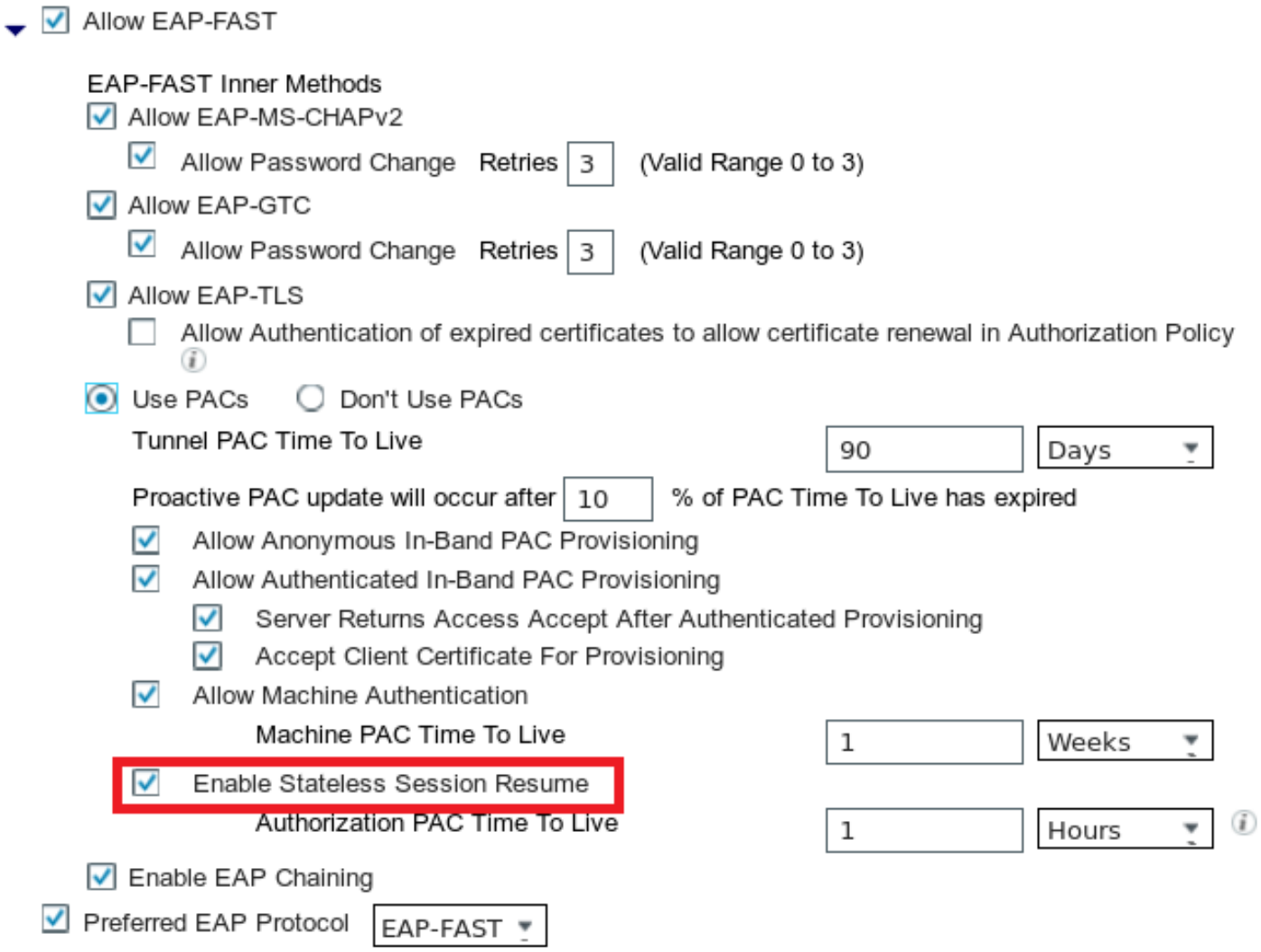
AnyConnect NAM implementation

It is enabled on client side (AnyConnect NAM) via Fast Reconnect - but it is used to control only authorization PAC usage.



With the setting disabled, NAM still uses the tunnel PAC to build the TLS tunnel (no certificates needed). However, this does not use authorization PACs in order to perform immediate user and machine authorization. As a result, phase 2 with the inner method is always required.

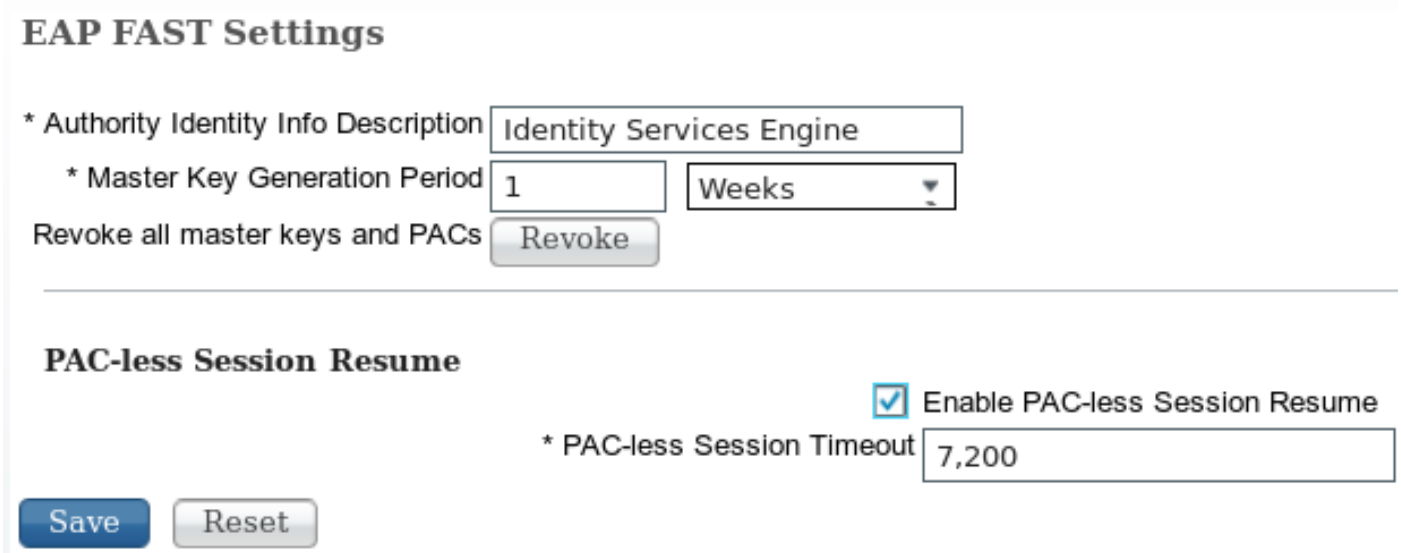
ISE has an option to enable Stateless Session Resume. And as on NAM it is just for Authorization PAC. Tunnel PAC usage is controlled with options "Use PACs".



NAM tries to use PACs if the option is enabled. If "Don't Use PACs" is configured in ISE and ISE receives a Tunnel PAC in the TLS extension the "insert here" error is reported and an EAP Failure is returned:

insert here

In ISE, it is also necessary to enable session resume based on TLS SessionID (from Global EAP-FAST settings). it is disabled by default:



Please keep in mind that only one type of session resume can be used. SessionID based is used only for PAC-less deployments, RFC 4507 based is used only for PAC deployments.

PAC provisioning (phase 0)

PACs can be automatically provisioned in phase0. Phase 0 consists of:

- TLS tunnel establishment
- Authentication (inner method)

PACs are delivered after a successful authentication inside the TLS tunnel via PAC TLV (and PAC TLV Acknowledgement)

Anonymous TLS tunnel

For deployments without a PKI infrastructure, it is possible to use an anonymous TLS tunnel. The anonymous TLS tunnel is built using the Diffie Hellman cipher suite - without the need of a server or client certificate. This approach is prone to Man in the Middle attacks (impersonation).

To use this option, NAM requires this configured option:

"If using PACs allow for unauthenticated PAC provisioning" (that makes sense only for password-based inner method because without PKI infrastructure it is not possible to use certificate-based inner method).

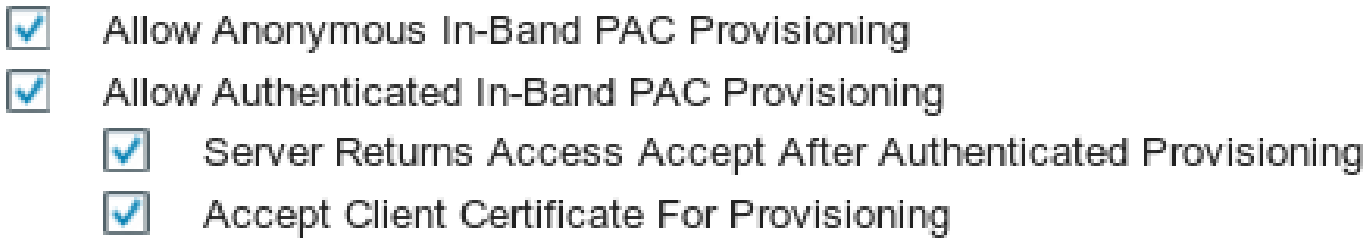
Also, ISE needs "Allow Anonymous In-band PAC Provisioning" configuration under the Authentication Allowed Protocols.

Anonymous in-band PAC provisioning is being used in TrustSec NDAC deployments (EAP-FAST session negotiated between network devices).

Authenticated TLS tunnel

This is the most secure and recommended option. The TLS tunnel is built based on the server certificate which is validated by the supplicant. This requires a PKI infrastructure on the server side only, which is required for ISE (on NAM it is possible to disable option "Validate Server Identity".

For ISE there are two additional options:



Normally, after PAC provisioning, an Access-Reject is sent forcing the supplicant to reauthenticate using PACs. But because PACs were delivered in the TLS tunnel with authentication, it is possible to shorten the whole process and return Access-Accept immediately after PAC provisioning.

The second option builds the TLS tunnel based on client certificate (this requires PKI deployment on the endpoints). This allows the TLS tunnel to be built with mutual authentication, which skips the inner method and goes directly to the PAC provisioning phase. it is important to be careful here - sometimes the supplicant presents a certificate which is not trusted by ISE (intended for other purposes) and the session

fails.

EAP-Chaining

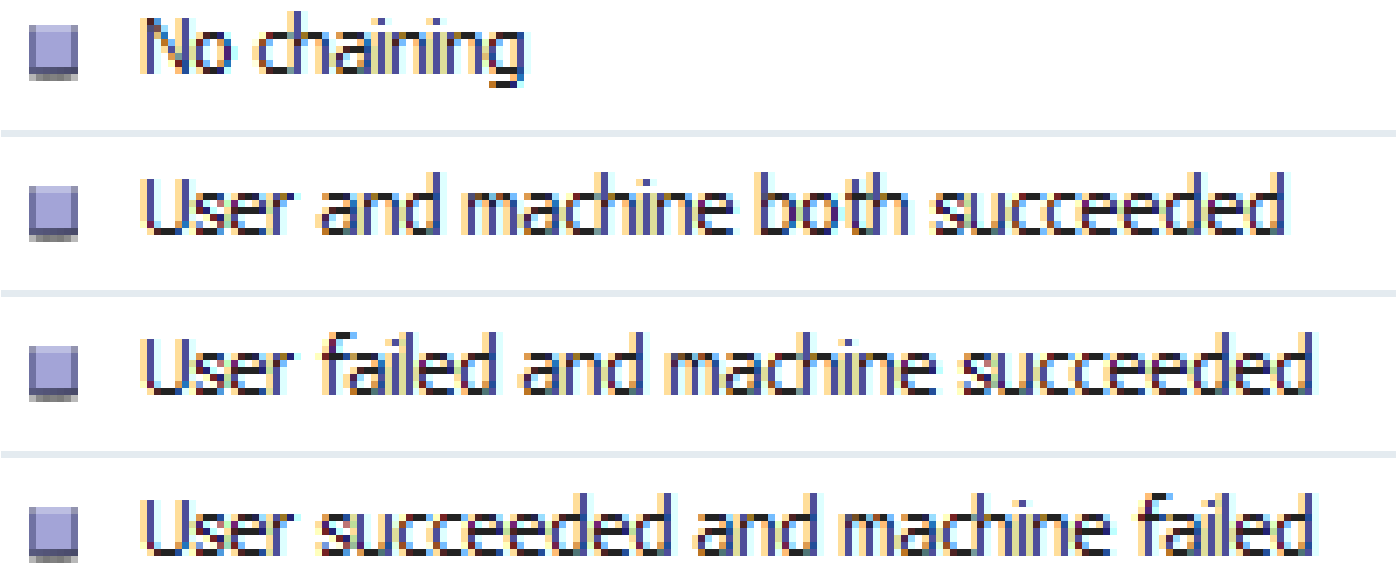
Allows user and machine authentication within one Radius/EAP session. Multiple EAP methods can be chained together. After the first authentication (typically machine) has finished successfully, the server sends an Intermediate-Result TLV (inside TLS tunnel) indicating success. That TLV must be accompanied by a Crypto-Binding TLV Request. Cryptobinding is used to prove that both the server and peer have participated in the specific sequence of authentications. The Cryptobinding process uses the keying material from phase 1 and phase 2. Additionally, one more TLV is attached: EAP-Payload - this is initiating the new session (typically for the user). Once the radius server (ISE) receives the Crypto-Binding TLV Response and validates it, this displays in the log and the next EAP method is tried (typically for user authentication):

<#root>

12126

EAP-FAST cryptobinding verification passed

If cryptobinding validation fails, the whole EAP session fails. If one of the authentications within failed then it is still fine - as a result, ISE allows an administrator to configure multiple chaining results based on Authorization Condition NetworkAccess:EapChainingResult:

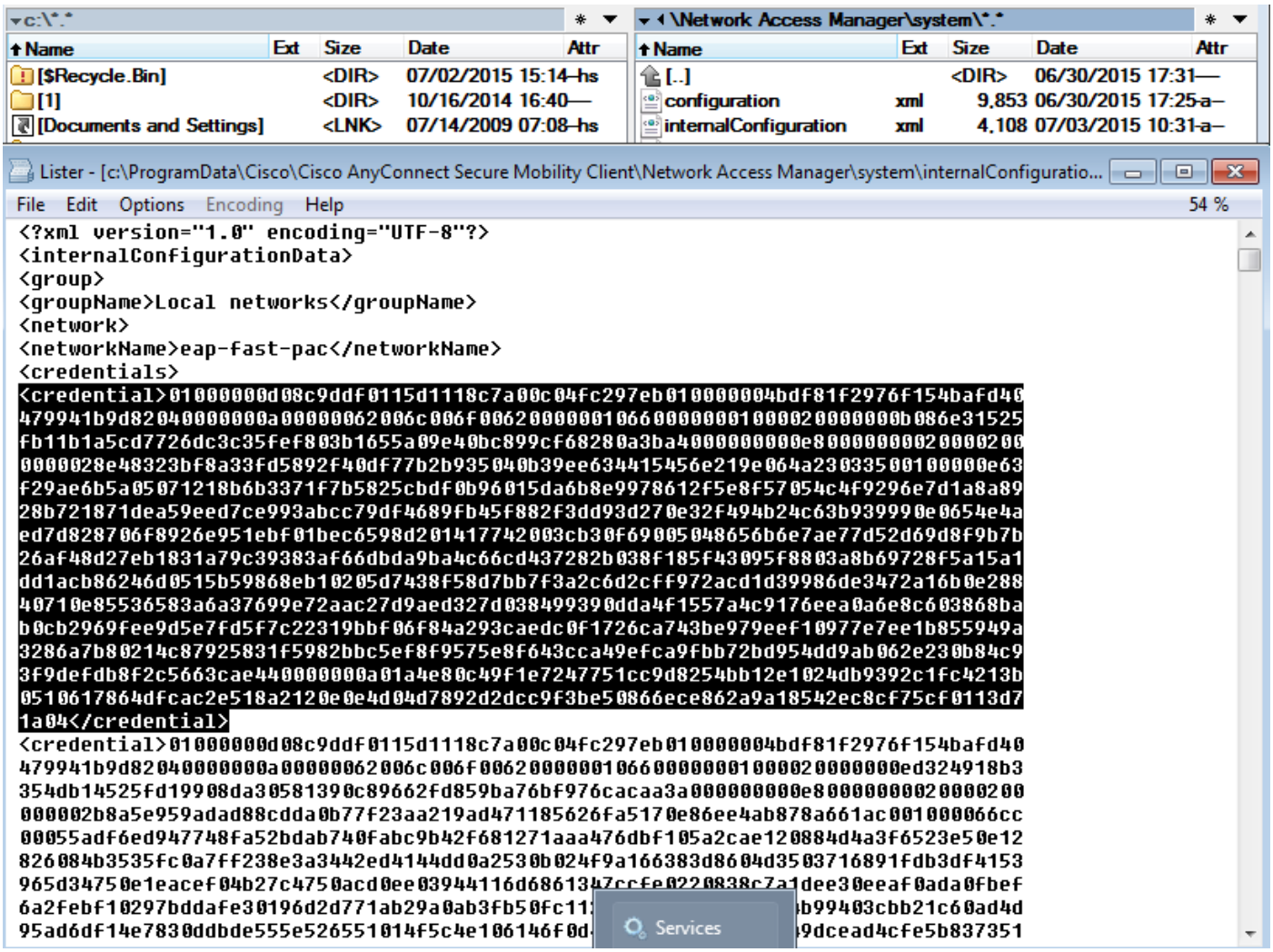


EAP-Chaining is enabled on NAM automatically when EAP-FAST user and machine authentication is enabled.

EAP-Chaining must be configured in ISE.

Where PAC files are stored

By default, Tunnel and Machine PACs are stored in C:\ProgramData\Cisco\Cisco AnyConnect Secure Mobility Client\Network Access Manager\system\internalConfiguration.xml in sections <credential>. Those are stored in encrypted form.

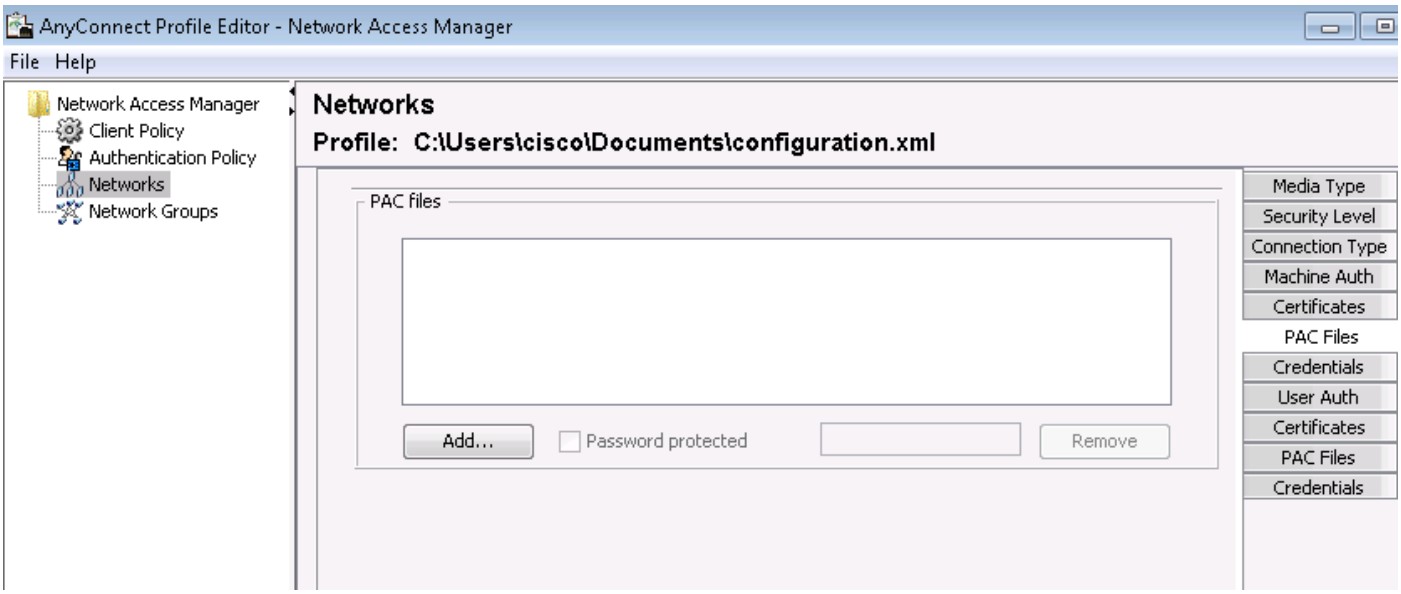


Authorization PACs are stored only in memory and are removed after reboot or NAM service restart.

A service restart is required to remove the Tunnel or Machine PAC.

AnyConnect NAM 3.1 vs 4.0

AnyConnect 3.x NAM profile editor allowed the administrator to configure PACs manually. This feature has been removed from AnyConnect 4.x NAM profile editor.

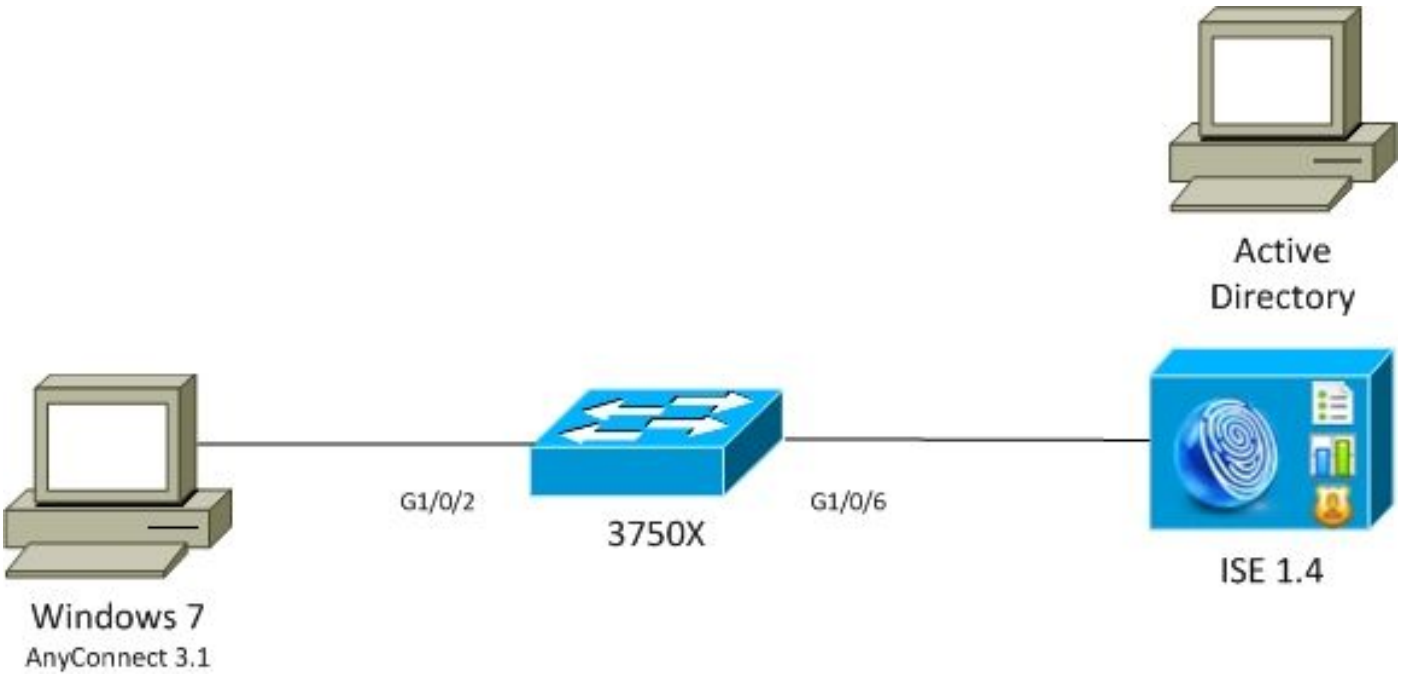


The decision to remove that functionality is based on Cisco bug ID CSCuf31422 and Cisco bug ID CSCua13140 .

Examples

Network Diagram

All the examples were tested using this network topology. The same applies also when using wireless.



EAP-Fast without EAP chaining with user and machine PAC

By default, EAP_chaining is disabled on ISE. However, all other options are enabled including Machine and Authorization PACs. The supplicant already has a valid Machine and Tunnel PAC. In this flow, there are two separate authentications - one for the machine and one for the user - with separate logs on ISE. The main steps as logged by ISE. First authentication (machine):

- Supplicant sends TLS Client Hello with Machine PAC.
- Server validates the Machine PAC and builds the TLS tunnel (no certificates used).
- Server validates the Machine PAC and performs the account lookup in Active Directory and skips the inner method.

<#root>

```
12102 Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negotiate
```

12800 Extracted first TLS record; TLS handshake started

12174 Received Machine PAC

```
12805 Extracted TLS ClientHello message
12806 Prepared TLS ServerHello message
12801 Prepared TLS ChangeCipherSpec message
```

12816 TLS handshake succeeded

12132 EAP-FAST built PAC-based tunnel for purpose of authentication

24351 Account validation succeeded

24420 User's Attributes retrieval from Active Directory succeeded - example . com

22037 Authentication Passed

12124 EAP-FAST inner method skipped

```
11503 Prepared EAP-Success
11002 Returned RADIUS Access-Accept
```

The second authentication (user):

- Supplicant sends the TLS Client Hello with Tunnel PAC.
- Server validates the PAC and builds the TLS tunnel (no certificates used).
- As supplicant does not have any Authorization PAC, the inner method (EAP-MSCHAP) is used for authentication.

<#root>

```
12102 Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negotiate
```

12800 Extracted first TLS record; TLS handshake started

12175 Received Tunnel PAC

12805 Extracted TLS ClientHello message
12806 Prepared TLS ServerHello message
12801 Prepared TLS ChangeCipherSpec message

12816 TLS handshake succeeded

12132 EAP-FAST built PAC-based tunnel for purpose of authentication

12125 EAP-FAST inner method started

11806 Prepared EAP-Request for inner method proposing

EAP-MSCHAP

with challenge

24402 User authentication against Active Directory succeeded - example . com

22037 Authentication Passed

11503 Prepared EAP-Success
11002 Returned RADIUS Access-Accept

In the "Other Attributes" section of the detailed report in ISE, this is noted for both user and machine authentications:

<#root>

EapChainingResult:

No chaining

EAP-Fast with EAP chaining with PAC Fast Reconnect

In this flow, the supplicant already has a valid Tunnel PAC along with the User and Machine Authorization PACs:

- Supplicant sends the TLS Client Hello with Tunnel PAC.
- Server validates the PAC and builds the TLS tunnel (no certificates used).
- ISE starts EAP Chaining, supplicant attaches Authorization PACs for user and Machine using TLV inside the TLS tunnel.
- ISE validates the Authorization PACs (no inner method needed), verifies that accounts exist in Active Directory (no additional authentication), returns success.

<#root>

12102 Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negotia

12800 Extracted first TLS record; TLS handshake started

12175 Received Tunnel PAC

12805 Extracted TLS ClientHello message
12806 Prepared TLS ServerHello message

12801 Prepared TLS ChangeCipherSpec message

12816 TLS handshake succeeded

12132 EAP-FAST built PAC-based tunnel for purpose of authentication

12209 Starting EAP chaining

12210 Received User Authorization PAC

12211 Received Machine Authorization PAC

24420 User's Attributes retrieval from Active Directory succeeded - example .com

22037 Authentication Passed

24439 Machine Attributes retrieval from Active Directory succeeded - example .com

22037 Authentication Passed

11503 Prepared EAP-Success
11002 Returned RADIUS Access-Accept

In the "Other Attributes" section of the detailed report in ISE, this result is noted:

<#root>

EapChainingResult:

EAP Chaining

Additionally, both user and machine credentials are included in the same log as seen here:

```
Username: cisco,host/mgarcarz-PC
```

EAP-Fast with EAP chaining without PAC

In this flow, NAM is configured to not use a PAC, ISE is also configured to not use PAC (but with EAP Chaining)

- Supplicant sends TLS Client Hello without Tunnel PAC.
- Server responds with the TLS Certificate and Certificate Request payloads.
- Supplicant must trust server certificate, does not send any client certificate (certificate payload is zero), TLS tunnel is built.
- ISE send a TLV request for the client certificate inside the TLS tunnel, but supplicant does not (it is not necessary to have it in order to continue).
- Starts EAP Chaining for user, using inner method with MSCHAPv2 authentication.
- Continues with machine authentication, using inner method with MSCHAPv2 authentication.
- No PACs are being provisioned.

<#root>

```
12102     Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negot
```

12800

Extracted first TLS record; TLS handshake started

```
12805     Extracted TLS ClientHello message
```

12806   Prepared TLS ServerHello message

12807   Prepared TLS Certificate message

```
12809     Prepared TLS CertificateRequest message
12811     Extracted TLS Certificate message containing client certificate
12812     Extracted TLS ClientKeyExchange message
```

12816   TLS handshake succeeded

12207   Client certificate was requested but not received during tunnel establishment. Will renegotiat

12226   Started renegotiated TLS handshake

```
12104     Extracted EAP-Response containing EAP-FAST challenge-response
12811     Extracted TLS Certificate message containing client certificate
```

```
12812     Extracted TLS ClientKeyExchange message
12804     Extracted TLS Finished message
12801     Prepared TLS ChangeCipherSpec message
12802     Prepared TLS Finished message
12226     Started renegotiated TLS handshake
12205     Client certificate was requested but not received inside the tunnel. Will continue with inner
12176     EAP-FAST PAC-less full handshake finished successfully
12209     Starting EAP chaining
12218     Selected identity type 'User'
11806     Prepared EAP-Request for inner method proposing EAP-MSCHAP with challenge
24402     User authentication against Active Directory succeeded - example .com
22037     Authentication Passed
12219     Selected identity type 'Machine'
11806     Prepared EAP-Request for inner method proposing EAP-MSCHAP with challenge
24470     Machine authentication against Active Directory is successful - example .com
22037     Authentication Passed
11503     Prepared EAP-Success
11002     Returned RADIUS Access-Accept
```

EAP-Fast with EAP chaining authorization PAC expiration

In this flow, the Supplicant has a valid Tunnel PAC but has expired Authorization PACs:

- Supplicant sends the TLS Client Hello with Tunnel PAC.
- Server validates the PAC and builds the TLS tunnel (no certificates used).
- ISE starts EAP Chaining, supplicant attaches Authorization PACs for User and Machine using TLV inside the TLS tunnel.
- As the PACs are expired, the inner method for both user and machine is started (EAP-MSCHAP).
- Once both authentications are successful, both user and machine Authorization PACs are provisioned.

<#root>

12102 Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negotia

12800 Extracted first TLS record; TLS handshake started

12175 Received Tunnel PAC

12805 Extracted TLS ClientHello message
12806 Prepared TLS ServerHello message
12801 Prepared TLS ChangeCipherSpec message

12816 TLS handshake succeeded

12132 EAP-FAST built PAC-based tunnel for purpose of authentication

12209 Starting EAP chaining

12227 User Authorization PAC has expired - will run inner method

12228 Machine Authorization PAC has expired - will run inner method

12218 Selected identity type 'User'

11806 Prepared EAP-Request for inner method proposing EAP-MSCHAP with challenge

24402 User authentication against Active Directory succeeded - example .com

22037 Authentication Passed

12219 Selected identity type 'Machine'

24470 Machine authentication against Active Directory is successful - example .com

22037 Authentication Passed

12171 Successfully finished EAP-FAST user authorization PAC provisioning/update

12179 Successfully finished EAP-FAST machine authorization PAC provisioning/update

11503 Prepared EAP-Success
11002 Returned RADIUS Access-Accept

EAP-Fast with EAP chaining tunnel PAC expired

In this flow when no valid tunnel PAC exists, full TLS negotiation with inner phase occurs.

- Supplicant sends the TLS Client Hello without Tunnel PAC.
- Server responds with the TLS Certificate and Certificate Request payloads.
- Supplicant must trust server certificate, does not send client certificate (certificate payload is zero), TLS tunnel built.
- ISE sends TLV request for the client certificate inside the TLS tunnel, but supplicant does not (it is not necessary to have it in order to continue).
- Starts EAP Chaining for user, using inner method with MSCHAPv2 authentication.
- Continues with machine authentication, using inner method with MSCHAPv2 authentication.
- Successfully provisioned all PACs (enabled in ISE config).

<#root>

12102 Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negotia
12800 Extracted first TLS record; TLS handshake started
12805 Extracted TLS ClientHello message
12806 Prepared TLS ServerHello message

12807

Prepared TLS Certificate message

12809

Prepared TLS CertificateRequest message

12105 Prepared EAP-Request with another EAP-FAST challenge
11006 Returned RADIUS Access-Challenge
11001 Received RADIUS Access-Request

12816 TLS handshake succeeded

12207

Client certificate was requested but not received during tunnel establishment. Will renegotiate and requ

12226 Started renegotiated TLS handshake

12104 Extracted EAP-Response containing EAP-FAST challenge-response

12811 Extracted TLS Certificate message containing client certificate

12812 Extracted TLS ClientKeyExchange message
12804 Extracted TLS Finished message
12801 Prepared TLS ChangeCipherSpec message
12802 Prepared TLS Finished message
12226 Started renegotiated TLS handshake

12205 Client certificate was requested but not received inside the tunnel. Will continue with inner me

12149 EAP-FAST built authenticated tunnel for purpose of PAC provisioning

12105 Prepared EAP-Request with another EAP-FAST challenge
11006 Returned RADIUS Access-Challenge
11001 Received RADIUS Access-Request
11018 RADIUS is re-using an existing session
12104 Extracted EAP-Response containing EAP-FAST challenge-response

12209 Starting EAP chaining

12218 Selected identity type 'User'

11806 Prepared EAP-Request for inner method proposing EAP-MSCHAP with challenge

24402 User authentication against Active Directory succeeded - example .com

22037 Authentication Passed

12126 EAP-FAST cryptobinding verification passed

12200 Approved EAP-FAST client Tunnel PAC request
12202 Approved EAP-FAST client Authorization PAC request

12219 Selected identity type 'Machine'

11806 Prepared EAP-Request for inner method proposing EAP-MSCHAP with challenge

24470 Machine authentication against Active Directory is successful - example .com

22037 Authentication Passed

12169 Successfully finished EAP-FAST tunnel PAC provisioning/update

12171 Successfully finished EAP-FAST user authorization PAC provisioning/update

12170 Successfully finished EAP-FAST machine PAC provisioning/update

12179 Successfully finished EAP-FAST machine authorization PAC provisioning/update

11503 Prepared EAP-Success
11002 Returned RADIUS Access-Accept

EAP-Fast with EAP chaining and anonymous TLS tunnel PAC provisioning

In this flow, ISE and NAM anonymous TLS tunnel is configured for PAC provisioning (ISE authenticated TLS tunnel for PAC provisioning is disabled) PAC provisioning request looks like:

- Supplicant sends TLS Client Hello without multiple ciphersuites.
- Server responds with the TLS Server Hello and TLS anonymous Diffie Hellman ciphers (for example TLS_DH_anon_WITH_AES_128_CBC_SHA).
- Supplicant accepts it and the anonymous TLS tunnel is built (no certificates exchanged).
- Starts EAP Chaining for user, using inner method with MSCHAPv2 authentication.
- Continues with machine authentication, using inner method with MSCHAPv2 authentication.
- Because the anonymous TLS tunnel is being built Authorization PACs are not allowed.
- Radius Reject is returned to force supplicant to reauthenticate (using provisioned PAC).

<#root>

12102 Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negot

12800 Extracted first TLS record; TLS handshake started

12805 Extracted TLS ClientHello message

12806 Prepared TLS ServerHello message

12808 Prepared TLS ServerKeyExchange message

12810 Prepared TLS ServerDone message

12812 Extracted TLS ClientKeyExchange message
12804 Extracted TLS Finished message
12801 Prepared TLS ChangeCipherSpec message

12802 Prepared TLS Finished message

12816 TLS handshake succeeded

12131 EAP-FAST built anonymous tunnel for purpose of PAC provisioning

12209 Starting EAP chaining

12218 Selected identity type 'User'

11806 Prepared EAP-Request for inner method proposing EAP-MSCHAP with challenge

24402 User authentication against Active Directory succeeded - example .com

22037 Authentication Passed

12162 Cannot provision Authorization PAC on anonymous provisioning. Authorization PAC can be provisi

12200 Approved EAP-FAST client Tunnel PAC request
12219 Selected identity type 'Machine'

24470 Machine authentication against Active Directory is successful - example .com

22037 Authentication Passed

12162 Cannot provision Authorization PAC on anonymous provisioning. Authorization PAC can be provisi

12169 Successfully finished EAP-FAST tunnel PAC provisioning/update

12170 Successfully finished EAP-FAST machine PAC provisioning/update

11504 Prepared EAP-Failure

11003 Returned RADIUS Access-Reject

Wireshark packet captures for anonymous TLS tunnel negotiation:

Source	Destination	Protocol	Length	Info	User-Name
10.62.148.109	10.48.17.14	RADIUS	301	Access-Request(1) (id=190,	anonymous
10.48.17.14	10.62.148.109	RADIUS	193	Access-Challenge(11) (id=19	
10.62.148.109	10.48.17.14	RADIUS	498	Access-Request(1) (id=191,	anonymous
10.48.17.14	10.62.148.109	RADIUS	793	Access-Challenge(11) (id=19	
10.62.148.109	10.48.17.14	RADIUS	706	Access-Request(1) (id=192,	anonymous
10.48.17.14	10.62.148.109	RADIUS	232	Access-Challenge(11) (id=19	
10.62.148.109	10.48.17.14	RADIUS	378	Access-Request(1) (id=193,	anonymous
10.48.17.14	10.62.148.109	RADIUS	226	Access-Challenge(11) (id=19	
10.62.148.109	10.48.17.14	RADIUS	468	Access-Request(1) (id=194,	anonymous
10.48.17.14	10.62.148.109	RADIUS	258	Access-Challenge(11) (id=19	

```
Code: Request (1)
Id: 161
Length: 622
Type: Flexible Authentication via Secure Tunneling EAP (EAP-FAST) (43)
▷ EAP-TLS Flags: 0x01
▽ Secure Sockets Layer
  ▽ TLSv1 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 74
    ▽ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 70
        Version: TLS 1.0 (0x0301)
      ▷ Random
        Session ID Length: 32
        Session ID: 41aee5db065f48165c56144aa9dccdc93f67167fbae96393...
        Cipher Suite: TLS_DH_anon_WITH_AES_128_CBC_SHA (0x0034)
        Compression Method: null (0)
  ▽ TLSv1 Record Layer: Handshake Protocol: Server Key Exchange
      Content Type: Handshake (22)
```

EAP-Fast with EAP chaining user authentication only

In this flow, AnyConnect NAM with EAP-FAST and User (EAP-TLS) and Machine authentication (EAP-TLS) is configured. The Windows PC is booted but user credentials are not provided. Switch initiates 802.1x session, NAM must respond however, user credentials are not provided, (no access to user store and certificate yet) therefore. user authentication fails while the machine is successful - ISE authz condition "Network Access:EapChainingResult EQUALS User failed and machine succeeded" is satisfied. Later, the user logs in and another authentication starts, both user and machine succeeds.

- Supplicant sends TLS Client Hello with Machine PAC.
- Server responds with the TLS Change Cipher Spec - TLS tunnel is immediately build based on that PAC.
- ISE initiates EAP Chaining and asking for user identity.
- Supplicant provides the machine identity instead (user not yet ready), finishes EAP-TLS inner method.

- ISE asks for user identity again, supplicant can not provide it.
- ISE sends TLV with intermediate result = failure (for user authentication).
- ISE returns the final EAP success message, ISE condition Network Access:EapChainingResult EQUALS User failed and machine succeeded is satisfied.

<#root>

```
12102   Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negotia
12800   Extracted first TLS record; TLS handshake started

12174   Received Machine PAC



12805   Extracted TLS ClientHello message
12806   Prepared TLS ServerHello message

12801   Prepared TLS ChangeCipherSpec message


12802   Prepared TLS Finished message


12816   TLS handshake succeeded



12132   EAP-FAST built PAC-based tunnel for purpose of authentication



12209   Starting EAP chaining


12218   Selected identity type 'User'



12213   Identity type provided by client is not equal to requested type


12215   Client suggested 'Machine' identity type instead



12104   Extracted EAP-Response containing EAP-FAST challenge-response
12523   Extracted EAP-Response/NAK for inner method

requesting to use EAP-TLS instead



12805   Extracted TLS ClientHello message
12806   Prepared TLS ServerHello message
12807   Prepared TLS Certificate message
12809   Prepared TLS CertificateRequest message


12816   TLS handshake succeeded
```

```
12509   EAP-TLS full handshake finished successfully

22070   Identity name is taken from certificate attribute
15013   Selected Identity Source - Test-AD

24323   Identity resolution detected single matching account


22037   Authentication Passed


12202   Approved EAP-FAST client Authorization PAC request

12218   Selected identity type 'User'


12213   Identity type provided by client is not equal to requested type

12216   Identity type provided by client was already used for authentication


12967   Sent EAP Intermediate Result TLV indicating failure


12179   Successfully finished EAP-FAST machine authorization PAC provisioning/update


12106   EAP-FAST authentication phase finished successfully
11503   Prepared EAP-Success
11002   Returned RADIUS Access-Accept
```

EAP-Fast with EAP chaining and inconsistent anonymous TLS tunnel settings

In this flow, ISE is configured for PAC provisioning only via anonymous TLS tunnel, but NAM is using an authenticated TLS tunnel, this is logged by ISE:

<#root>

```
12102   Extracted EAP-Response containing EAP-FAST challenge-response and accepting EAP-FAST as negotia
12800   Extracted first TLS record; TLS handshake started

12805   Extracted TLS ClientHello message


12814   Prepared TLS Alert message


12817   TLS handshake failed


12121   Client didn't provide suitable ciphers for anonymous PAC-provisioning


11504   Prepared EAP-Failure
```

```
11003   Returned RADIUS Access-Reject
```

This occurs when NAM is trying to build an authenticated TLS tunnel with its speciphic TLS ciphers - and those are not accepted by ISE which is configured for anonymous TLS tunnel (accepting DH ciphers only)

Troubleshoot

ISE

For detailed logs, Runtime-AAA debugs must be enabled on the corresponding PSN node. Here are a few example logs from prrt-server.log:

Machine PAC generation:

<#root>

```
DEBUG,0x7fd5332fe700,cntx=0001162745,sesn=mgarcarz-ise14/223983918/29245,CPMSessionID=0A3E946D00000FE51
Using IID from PAC request for machine
,EapFastTlv.cpp:1234
DEBUG,0x7fd5332fe700,cntx=0001162745,sesn=mgarcarz-ise14/223983918/29245,CPMSessionID=0A3E946D00000FE51
Adding PAC of type=Machine Authorization
,EapFastProtocol.cpp:3610
DEBUG,0x7fd5332fe700,cntx=0001162745,sesn=mgarcarz-ise14/223983918/29245,CPMSessionID=0A3E946D00000FE51
Generating Pac, Issued PAC type=Machine Authorization with expiration time: Fri Jul  3 10:38:30 2015
```

PAC request approval:

<#root>

```
INFO ,0x7fd5330fc700,cntx=0001162745,sesn=mgarcarz-ise14/223983918/29245,CPMSessionID=0A3E946D00000FE51
PAC request approved for PAC type - Requested PAC type=Machine
,EapFastProtocol.cpp:955
INFO ,0x7fd5330fc700,cntx=0001162745,sesn=mgarcarz-ise14/223983918/29245,CPMSessionID=0A3E946D00000FE51
PAC request approved for PAC type - Requested PAC type=Machine Authorization
,EapFastProtocol.cpp:955
```

PAC validation:

<#root>

DEBUG,0x7fd5330fc700,cntx=0001162499,sesn=mgarcarz-ise14/223983918/29243,CPMSessionID=0A3E946D00000FE51

Authorization PAC is valid

,EapFastProtocol.cpp:3403

Eap,2015-07-03 09:34:39,208,DEBUG,0x7fd5330fc700,cntx=0001162499,sesn=mgarcarz-ise14/223983918/29243,CP

Authorization PAC accepted

,EapFastProtocol.cpp:3430

Example of successful summary for PAC generation:

<#root>

DEBUG,0x7fd5331fd700,cntx=0001162749,sesn=mgarcarz-ise14/223983918/29245,CPMSessionID=0A3E946D00000FE51

Generated PAC of type Tunnel V1A. Generated PAC of type User Authorization. Generated PAC of type Machin

. Success

Example of successful summary for PAC validation:

<#root>

DEBUG,0x7fd5330fc700,cntx=0001162503,sesn=mgarcarz-ise14/223983918/29243,CPMSessionID=0A3E946D00000FE51

PAC type Tunnel V1A. PAC is valid.Skip inner method. Skip inner method. Success

AnyConnect NAM

Example for non EAP-Chaining session, Machine authentication without fast reconnect:

<#root>

EAP: Identity requested
Auth[eap-fast-pac:

machine-auth

]:

Performing full authentication

Auth[eap-fast-pac:

machine-auth

]:

Disabling fast reauthentication

Example of Authorization PAC lookup (machine authentication for non EAP-Chaining session):

<#root>

Looking for matching pac with iid: host/ADMIN-PC2

Requested machine pac was sen

All states of inner method (for MSCHAP) can be verified from these logs:

<#root>

```
 EAP (0) EAP-MSCHAP-V2:
```

State: 0

```
 (eap_auth_mschapv2_c.c 731
 EAP (0) EAP-MSCHAP-V2:
```

State: 2

```
 (eap_auth_mschapv2_c.c 731
 EAP (0) EAP-MSCHAP-V2:
```

State: 1

```
 (eap_auth_mschapv2_c.c 731
 EAP (0) EAP-MSCHAP-V2:
```

State: 4

```
 (eap_auth_mschapv2_c.c 73
```

NAM allows the configuration of the extended logging feature which captures all EAP packets and save them in pcap file. This is especially helpful for Start Before Logon functionality (EAP packets are captured even for authentications which occur before user logon). For feature activation ask your TAC engineer.

References

- **Cisco AnyConnect Secure Mobility Client Administrator Guide, Release 4.0 EAP-FAST configuration**
- **Cisco Identity Services Engine Administrator Guide, Release 1.4 EAP-FAST recommendations**
- **Cisco Identity Services Engine Design Guides**
- **Deploying EAP Chaining with AnyConnect NAM and Cisco ISE**
- **Technical Support & Documentation - Cisco Systems**