

DAP の拡張機能の設定例

目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[OU ベースの照合の例](#)

[グループ メンバーシップ \(memberOf \) の例](#)

[カスタム関数を伴う CheckAndMsg](#)

[アンチウイルス、アンチスパイウェア、およびファイアウォールの例](#)

[アンチウイルスのインストールの確認](#)

[アンチウイルスのインストールと最終更新日を確認してエラー メッセージを表示](#)

[アンチスパイウェアのインストールの確認](#)

[ファイアウォールのインストールの確認](#)

[アンチウイルス、アンチスパイウェア、またはファイアウォールのインストールの確認](#)

[スパイウェアがインストールされていない場合に終了する](#)

[アンチウイルスとファイアウォールのインストールの確認、およびアンチウイルスの最終更新日が 30 日以内であることの検証](#)

[正規表現の照合](#)

[エンドポイント PC にホットフィックス KB944 のいずれかのインスタンスが含まれている場合に接続する](#)

[Lua スクリプトを使用して MAC アドレスの OUI を確認する](#)

[ホスト名の最初の 3 文字 \(大文字と小文字を区別しない \) に基づいて接続する](#)

[エンドポイント PC の device.id と証明書のシリアル番号が同じ場合に接続する](#)

[Domain レジストリ キーの CSD ホスト スキャンに基づいて DAP を適用する](#)

[Windows 7 と CSD 3.5 の DAP サポート](#)

[iPhone、iPad、およびモバイル デバイスの識別](#)

[DAP の使用による特定ブラウザでの接続の回避](#)

[警告](#)

[FAQ](#)

[LUA はなぜ何人かのユーザ向けに、ないすべてのためにはたらきますスクリプトを書きますか。](#)

概要

このドキュメントでは、リモート アクセス VPN のダイナミック アクセス ポリシー (DAP) の拡張機能について説明します。条件による照合にさらに柔軟性が必要な場合に、これらの拡張機能を使用できます。

注: [debug](#) コマンドを使用する前に、『[debug コマンドの重要な情報](#)』を参照してください。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- 基本的な DAP の理解を前提としています。『[ASA 8.x ダイナミック アクセス ポリシー \(DAP\) 導入ガイド](#)』 (サポート ドキュメント) または 『[ASA 8.x ダイナミック アクセス ポリシー \(DAP\) 導入ガイド](#)』 (サポート コミュニティ) を参照してください。
- Lua プログラミングについての十分な理解も役立ちます。Web 上で提供されている Lua プログラミングの資料を参照してください。

使用するコンポーネント

このドキュメントは、特定のソフトウェアやハードウェアのバージョンに限定されるものではありませんが、設定の実行には Adaptive Security Device Manager (ASDM) が必要です。

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たものです。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。ネットワークが稼働中の場合は、コマンドが及ぼす潜在的な影響を十分に理解しておく必要があります。

背景説明

注意： ASDM の GUI 設定または EVAL 関数が必要な照合動作を提供しない場合のみ、DAP の拡張カスタム Lua 機能を使用してください。実稼働環境へ導入する場合は、DAP の意図しない動作を避けるために、拡張 Lua 機能は細心の注意を払って Cisco Engineering/Technical Assistance Center (TAC) の指導に従って使用してください。

リモート アクセス VPN で DAP を使用する場合、条件による照合にさらに柔軟性が必要になる場合があります。たとえば、次のようなシナリオに基づいてさまざまな DAP を適用できます。

- ユーザ オブジェクトを含む組織ユニット (OU) またはその他の階層レベル。
- 命名規則に従ったグループ名 (memberOf) 。ただし一致する可能性がある名前が多いため、グループ名にワイルドカードを使用する。
- エンドポイント PC のアンチウイルス、アンチスパイウェア、またはファイアウォール パッケージの確認。

1. 一致基準の論理式を作成するために ASDM を使用します。

2. 論理式と Lua コードを含むカスタム関数を作成するために拡張モードを使用します。

OU ベースの照合の例

Lightweight Directory Access Protocol (LDAP) サーバは、DAP が論理式の中で使用可能な多くの属性を返すことができます。

このような属性の例を表示するには、適応型セキュリティ アプライアンス (ASA) のコンソールで **debug dap trace** コマンドを使用します。

```
assert(function()  
if ( (type(aaa.ldap.distinguishedName) == "string") and  
(string.find(aaa.ldap.distinguishedName, "OU=Admins,dc=cisco,dc=com$")  
~= nil) ) then  
return true  
end  
return false  
end
```

LDAP サーバが返す属性の 1 つにユーザの識別名 (DN) があります。DN は、ユーザ オブジェクトが存在するディレクトリ内の場所を暗黙に特定します。たとえば、DN が CN=Joe User,OU=Admins,dc=cisco,dc=com である場合、このユーザは OU=Admins,dc=cisco,dc=com に存在します。すべての管理者がこの OU (またはこのレベルの下の任意のコンテナ) に存在する場合、条件に一致させるためには次の論理式を使用します。

```
assert(function()  
if ( (type(aaa.ldap.distinguishedName) == "string") and  
(string.find(aaa.ldap.distinguishedName, "OU=Admins,dc=cisco,dc=com$")  
~= nil) ) then  
return true  
end  
return false  
end)()
```

この例では、string.find 関数で正規表現を使用できます。文字列の最後の \$ は、この文字列を distinguishedName フィールドの末尾に固定します。

() . % + - * ? [^ \$ などの特殊文字を通常の文字として扱うには、検索文字列の中でエスケープ文字 % を使用します。たとえば、文字列 (OU=Admins,dc=my-domain,dc=com\$) 内の文字 - は、(OU=Admins,dc=my%-domain,dc=com\$) のようにエスケープできます。

グループ メンバーシップ (memberOf) の例

Active Directory (AD) のグループ メンバーシップのパターン照合のために、同様の基本的な論理式を作成できます。ユーザが複数のグループのメンバーであることが考えられるため、DAP は LDAP サーバからの応答を解析して個別のエントリにし、テーブル内に保持します。この場合、次を行うためのより高度な関数が必要になります。

- ユーザが 1 つのグループにのみ所属する場合は、memberOf フィールドを文字列として比較する。
- 返されたそれぞれの memberOf フィールドで繰り返し処理し、返されたデータが「table」タイプであるかどうかを確認する。

この例では、ユーザが「-stu」で終わるいずれかのグループのメンバーである場合、この DAP に一致します。

```
assert(function()  
local pattern = "-stu$"  
local attribute = aaa.ldap.memberOf  
if ((type(attribute) == "string") and  
(string.find(attribute, pattern) ~= nil)) then  
return true  
elseif (type(attribute) == "table") then  
local k, v  
for k, v in pairs(attribute) do  
if (string.find(v, pattern) ~= nil) then  
return true  
end  
end  
end  
return false  
end)()
```

カスタム関数を伴う CheckAndMsg

次の関数では、アクションが終了に設定された DAP を使用します。

```
(assert(function()  
local block_connection = true  
local update_threshold = "150000" --this is the value of lastupdate in  
seconds  
for k,v in pairs(endpoint.av) do  
if (CheckAndMsg(EVAL(v.exists, "EQ", "true", "string") and EVAL  
(v.lastupdate, "LT", update_threshold, "integer"),k.." exists; last update is  
"..string.sub((tonumber(v.lastupdate)/86400), 1, 3).. " days",k.." does not exist; last update is  
"..string.sub((tonumber(v.lastupdate)/86400), 1, 3).. " days")) then  
block_connection = false  
end  
end  
return block_connection  
end)())
```

終了した場合は次のメッセージを表示します。

```
Login denied.<AV Name> does not exists; last update is <X> days
```

アンチウイルス、アンチスパイウェア、およびファイアウォールの例

これらの Lua 関数は、Cisco Secure Desktop (CSD) ホスト スキャンによって返される、エンドポイント PC のアンチウイルス、アンチスパイウェア、およびファイアウォール パッケージ関連の属性を確認します。

アンチウイルスのインストールの確認

このカスタム関数は、CSD がアンチウイルスを検出したかどうかを確認します。

```
assert(function()  
for k,v in pairs(endpoint.av) do
```

```
if (EVAL(v.exists, "EQ", "true", "string")) then
return true
end
end
return false
end()
```

アンチウイルスのインストールと最終更新日を確認してエラー メッセージを表示

この例では、DAP によりアンチウイルスのインストールと最終更新日を確認し、ユーザに修復方法を通知する方法について説明します。「[アンチウイルスのインストールの確認](#)」の関数に似た関数を使用します。

照合する認証、許可、およびアカウントिंग (AAA) 属性を設定します。[Advanced] フィールドで [AND] 演算が選択されていることを確認します。[Action] フィールドで [Terminate] オプションが選択されていることを確認します。ユーザが AAA 属性に一致し、Lua 関数が true の値を返す場合は、DAP が選択されて DAP の記録が表示された理由を説明するメッセージが表示され、ユーザの接続は終了します。Lua 関数が true の値を返さない場合は、DAP は一致せず、アクセスは許可されます。メッセージ ボックス フィールドにメッセージ「No anti-virus program found, please install anti-virus and try again.」を入力します。ユーザがアンチウイルス パッケージをインストールしており、更新日数のしきい値に達していない場合は、次の例の 8 行目の二重引用符で示されているように、メッセージは表示されません。

```
(assert(function()
local block_connection = true
local update_days = "15" --days
local av_lastupdate = update_days*86400
for k,v in pairs(endpoint.av) do
if (CheckAndMsg(EVAL(v.exists, "EQ", "true", "string") and EVAL(v.lastupdate, "LT",
av_lastupdate, "integer"),",",k..) exists; but last update is greater than 15 days old. Expecting
under 15 days.)) then
block_connection = false
elseif (EVAL(v.exists, "NE", "true", "string")) then
block_connection = true
end
end
return block_connection
end())
```

ユーザが Norton アンチウイルスをインストールしているが、最終更新日が 15 日間よりも大きい場合は次のサンプル メッセージが表示されます。

```
NortonAV exists; but last update is greater than 15 days old. Expecting under 15 days.
```

EVAL が一致しなかった場合は次の関数に移り、一致して true の値を返します。2 番目の関数には関連付けられた CheckAndMsg がいないため、次の DAP メッセージ テキストが使用されます。

```
No anti-virus program found, please install anti-virus and try again.
```

まとめると、DAP の照合のために DAP はユーザの AAA とエンドポイントの属性を調べます。DAP が一致する場合、ユーザにはメッセージが表示されて接続が終了します。エンドポイントの一致は、DAP に true または false を返す Lua EVAL の結果によります。true の場合は一致し、接続は拒否されます。false の場合は一致せず、接続は許可されます。

1. ループ内の最初の関数では、endpoint.av.xxxxx.exists が true に等しく、最終更新日が設定された日数よりも小さいかを確認します。アンチウイルス ソフトウェアをインストールしていないユーザはアクセスを許可されます。これはユーザの AAA は一致しますが、Lua は具体的には endpoint.av.xxxxx.exists = true and endpoint.av.xxxxx.lastupdate <= days を確認

するためです。

2. 2 番目のループではアンチウイルス ソフトウェアをインストールしていないユーザを検出してブロックします。2 番目の関数では endpoint.av.xxxxx.exists NE true のみを確認するためです。ユーザの endpoint av exists が true と等しくない場合、この関数は true の値を返し、アンチウイルスがないという意味になります。DAP が一致し、接続が拒否されます。

アンチスパイウェアのインストールの確認

このカスタム関数は、CSD がアンチスパイウェアを検出したかどうかを確認します。

```
assert(function()  
for k,v in pairs(endpoint.as) do  
if (EVAL(v.exists, "EQ", "true", "string")) then  
return true  
end  
end  
return false  
end)()
```

ファイアウォールのインストールの確認

このカスタム関数は、CSD がファイアウォールを検出したかどうかを確認します。

```
assert(function()  
for k,v in pairs(endpoint.fw) do  
if (EVAL(v.exists, "EQ", "true", "string")) then  
return true  
end  
end  
return false  
end)()
```

アンチウイルス、アンチスパイウェア、またはファイアウォールのインストールの確認

この関数は、アンチウイルス、アンチスパイウェア、またはファイアウォール パッケージが検出された場合に true を返します。

```
assert(function()  
function check(antix)  
if (type(antix) == "table") then  
for k,v in pairs(antix) do  
if (EVAL(v.exists, "EQ", "true", "string")) then  
return true  
end  
end  
end  
return false  
end  
return (check(endpoint.av) or check(endpoint.fw) or check(endpoint.as))  
end)()
```

スパイウェアがインストールされていない場合に終了する

この関数と「[アンチスパイウェアのインストールの確認](#)」の関数の唯一の違いは、assert の前に

「not」があることです。

```
not assert(function()  
for k,v in pairs(endpoint.as) do  
if (EVAL(v.exists, "EQ", "true", "string")) then  
return true  
end  
end  
return false  
end)()
```

アンチウイルスとファイアウォールのインストールの確認、およびアンチウイルスの最終更新日が 30 日以内であることの検証

この例では、アンチウイルスとファイアウォールが検出され、アンチウイルスの最終更新日が 30 日以内である場合に true を返します。

```
assert(function()  
function checkav(antix)  
if (type(antix) == "table") then  
for k,v in pairs(antix) do  
if (EVAL(v.activescan, "EQ", "ok", "string") and EVAL (v.lastupdate, "LT", "2592000",  
"integer")) then  
return true  
end  
end  
end  
return false  
end  
function checkfw(antix)  
if (type(antix) == "table") then  
for k,v in pairs(antix) do  
if (EVAL(v.enabled, "EQ", "ok", "string")) then  
return true  
end  
end  
end  
return false  
end  
return (checkav(endpoint.av) and checkfw(endpoint.fw))  
end)()
```

ファイアウォールには返す最終更新日の値がないため、別の関数になります。

正規表現の照合

このセクションでは、特定の属性の照合を行ってホスト マシンの妥当性を判断するために正規表現を使用する関数について説明します。次の正規表現の機能はテスト済みであり、有効です。

- ドル記号 (\$) は検索文字列を戻り値の末尾に固定します。
- キャレット (^) は検索文字列を戻り値の先頭に固定します。
- [Aa] のように角カッコで囲まれた文字は、特定の位置の複数の文字と一致します。たとえば、OU=Cisco (大文字と小文字を区別しない) に一致させるには、OU=[Cc][Ii][Ss][Cc][Oo] を使用します。
- ピリオド (.) はその位置の任意の 1 文字と一致します。たとえば、Group..Users は Group01Users、Group33Users などに一致します。

エンドポイント PC にホットフィックス KB944 のいずれかのインスタンスが含まれている場合に接続する

この関数では、ホットフィックスのリストにあるパターンが含まれているかどうかを確認するために正規表現の照合を使用します。この例では、Cisco Secure Desktop はエンドポイント PC のすべてのホットフィックスを返します。KB944 のインスタンスがある場合は、DAP ポリシーが一致し、適用されます。

```
assert(function ()
local pattern = "KB944"
local true_on_match = true
local match = false
for k,v in pairs(endpoint.os.hotfix) do
print(k)
match = string.find(k, pattern)
if (match) then
if (true_on_match) then
return true
else return (false)
end
end
end
end)()
```

たとえば、ホスト マシンにホットフィックス KB944533 やホットフィックス KB944653 が含まれる場合、ルールに一致します。

Lua スクリプトを使用して MAC アドレスの OUI を確認する

この関数は、「[エンドポイント PC にホットフィックス KB944 のいずれかのインスタンスが含まれている場合に接続する](#)」に記載されている関数と似ています。この関数では、MAC アドレスの組織固有識別子 (OUI) を照合するために正規表現を使用します。

この例では、MAC アドレスは d067.e5XX.XX で始まります。正規表現と Lua コードを使用して同じ OUI MAC で始まるマシンを照合します。

```
assert(function ()
local pattern = "^d067\.e5*"
local true_on_match = true

local match = false
for k,v in pairs(endpoint.device.MAC) do
print(k)
match = string.find(k, pattern)
if (match) then
if (true_on_match) then
return true
else return (false)
end
end
end
end)()
```

注: 複数値の確認を行う場合は、この関数の別のバージョンが必要です。

ホスト名の最初の 3 文字 (大文字と小文字を区別しない) に基づいて接続する

この関数では、ホスト名の最初の 3 文字が msv (大文字と小文字を区別しない) であるかどうかを判断するために正規表現を使用します。

```
assert(function()  
local match_pattern = "^[Mm][Ss][Vv]"  
local match_value = endpoint.device.hostname  
if (type(match_value) == "string") then  
if (string.find(match_value, match_pattern) ~= nil) then  
return true  
end  
elseif (type(match_value) == "table") then  
local k,v  
for k,v in pairs(match_value) do  
if (string.find(v, match_pattern) ~= nil) then  
return true  
end  
end  
end  
return false  
end)()
```

エンドポイント PC の device.id と証明書のシリアル番号が同じ場合に接続する

この Lua 式は、エンドポイント PC の device.id と証明書のシリアル番号が同じ場合に接続することを目的としています。

```
assert(function()  
local match_pattern = endpoint.device.id  
for k,v in pairs(endpoint.certificate.user) do  
if (type(v.subject_e) == "string") then  
if (string.find(v.subject_e, match_pattern) ~= nil) then  
return true  
end  
elseif (type(v.subject_e) == "table") then  
local k,v  
for k,v in pairs(v.subject_e) do  
if (string.find(v, match_pattern) ~= nil) then  
return true  
end  
end  
end  
return false  
end ) ()
```

注: ワイルドカードの使用はこの特定の関数で (*) はたつきません (endpoint.certificate.user は [「*」] 作業)。各 KV ペアをそれぞれ取得し、それらによって解析して下さい。

Domain レジストリ キーの CSD ホスト スキャンに基づいて DAP を適用する

この手順では、ASDM による設定手順の例を示します。

1. ドメインを保持するレジストリ キーを
\\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Domain
で見つけます。

2. レジストリ設定用のホスト スキャン パラメータを定義します。
3. レジストリ エンドポイント属性を DAP ポリシーに適用します。
4. Secure Sockets Layer (SSL) の VPN セッションを確立します。
5. DAP デバッグで DAP ポリシーの適用を確認します。

Windows 7 と CSD 3.5 の DAP サポート

Windows 7 プラットフォームは CSD リリース 3.5 以降でサポートされています。ASDM 6.2.x メンテナンス リリースおよび 6.3.x リリースでは、Windows 7 OS の確認のためにこのインターフェイスを直接使用できます。それよりも前の ASDM リリースで Windows 7 マシンを確認するためには、拡張 DAP Lua スクリプトが必要です。ASA リリース 8.x およびベータ以前の CSD リリース 3.5 で Windows 7 マシンを確認するには、次の Lua スクリプト文字列を ASDM DAP の詳細ボックスに入力します。

```
(EVAL(endpoint.os.version,"EQ","Windows 7","string"))
```

iPhone、iPad、およびモバイル デバイスの識別

この Lua 式を使用して、特定のモバイル デバイスをその固有識別子 (UID) で追跡できます。この基本機能の実現に DAP を使用できます。

この値をハードコードできず、AD から読み取る必要がある場合は、より難しくなります。AD には専用の UID フィールドがないため、特定のユーザ用の値を別のフィールドに格納する必要があります。この例では UID の格納に otherHomePhone を使用します。

iPhone や iPad の UID を特定するためには、Web で適切なツールを検索します。

UID を特定したら、それをユーザの AD エントリの otherHomePhone に追加します。

debug ldap 255 コマンドとユーザ テスト認証により、LDAP 属性 otherHomePhone がプッシュされるのを確認します。

電話による接続を許可した後で、UID を含むエンドポイント属性 (endpoint.anyconnect.deviceuniqueid) を特定するために、接続を試みる間の DAP のトレースを実行します。

その後、次の Lua 式によって 2 つのパラメータを比較できます。

```
assert(function()  
if (type(aaa.ldap.otherHomePhone) ==type(endpoint.anyconnect.deviceuniqueid))
```

```
then
return true
end
return false
end()
```

DAP の使用による特定ブラウザでの接続の回避

この手順では、DAP を使用して Chrome ブラウザによる接続を回避する方法について説明します。

1. CSD を有効にします。
2. ホスト スキャン設定で、プロセス ID (PID) とプロセス名を使用してプロセス スキャンを追加します。

Windows では、タスク マネージャを使用して PID とプロセス名を判別できます。PID の値を表示するには、[Task Manager] を開いて [Processes] タブにアクセスし、[View] メニューをクリックして [Select Columns] を選択します。[Select Columns] または [Select Process Page Columns] ダイアログで [PID (Process Identifier)] チェックボックスをオンにして、[OK] をクリックします。

Mac では、アクティビティ モニターでプロセス ID を定義できます。または (Unix でも使用できる) bash シェルで、プロセスの実行中に `ps -e` コマンドを使用し、次に `cat /proc/<PID>/cmdline` コマンドを使用して、PID とプロセス名を照合します。

3. DAP ポリシーを作成し、そのプロセスがマシン上で作動しているかどうかをテストします。
4. 接続をテストします。

警告

1. このソリューションの問題は、ユーザが自身のマシン上で Chrome を開いたままにしておけないことです。DAP は特定の Chrome プロセスが作動しているかどうかを単純にチェックしますが、そのプロセス、または他のプロセスによってクライアントレス セッションが開始されたかどうかはチェックしません。このため、WebVPN 接続を試行しているときに、Chrome をバックグラウンドで実行することはできません。
2. ユーザが Firefox を使用して WebVPN セッションを開くというシナリオでは、ログインし

ようとするとう失敗します。まだ Chrome が作動していることに気付いて、ユーザは Chrome の接続を終了してからもう一度ログインを試みます。この場合も、CSD がホスト スキャンを再実行する必要があるため、ログインは失敗します。したがってユーザは、WebVPN へのアクセスで使用した Firefox のインスタンスも閉じて、Firefox を再起動する必要があります。このプロセスで、ユーザは混乱することがあります。シスコでは、DAP 失敗のメッセージを作成し、Chrome を終了して使用中のブラウザを閉じるようユーザに通知することを推奨しています。

FAQ

このセクションはこの資料に説明があるほとんどの FAQ の 1 つに返事を見なします情報を提供します。

LUA が何人かのユーザ向けに、ないすべてのためにはたらきますスクリプトを書く理由

この LUA スクリプトを考慮して下さい:

```
assert(function()  
    for k,v in pairs(endpoint.certificate.user) do  
        if (v.subject_store == "capi" and v.subject_dc == "homedepot") then  
            return true  
        end  
    end  
    return false  
end)()
```

このスクリプトは認証にあるサブジェクト DC をおおよび証明書ストア一致するために設計されています。ただし、このスクリプトは複数のマシンによってテストされ、いくつかのマシンで動作するために、多くの他で失敗するためにありました。

このスクリプトがただはたらくという原因は hostscan 値を戻す方法が理由で断続的にそうなったものです。はたらかない DAP トレースを表示するとき、*subject_dc* が認証ごとの複数の値を戻すことがわかります。また戻る最後の値がホームデポではないことがわかります:

```
DAP_TRACE: endpoint.policy.location = "CORP-Windows"  
:  
.  
DAP_TRACE: endpoint.certificate.user["6"].subject_store = "capi"  
DAP_TRACE: endpoint.certificate.user["6"].subject_dc = "com"  
DAP_TRACE: endpoint.certificate.user["6"].subject_dc = "homedepot"  
DAP_TRACE: endpoint.certificate.user["6"].subject_dc = "amer"
```

はたらく DAP トレースを表示するとき、これは観察することができます:

```
DAP_TRACE: endpoint.certificate.user["20"] = {}  
DAP_TRACE: endpoint.certificate.user["20"].subject_cn = "JHDOC6"  
DAP_TRACE: endpoint.certificate.user["20"].subject_e = "jimmie_harden@homedepot.com"  
DAP_TRACE: endpoint.certificate.user["20"].subject_ou = "Associates"  
DAP_TRACE: endpoint.certificate.user["20"].subject_store = "capi"  
DAP_TRACE: endpoint.certificate.user["20"].subject_dc = "com"  
DAP_TRACE: endpoint.certificate.user["20"].subject_dc = "homedepot"  
DAP_TRACE: endpoint.certificate.user["20"].issuer_cn = "The Home Depot Remote  
Access Issuing CA v2"
```

これは LUA スクリプトがきちんとはたらくことを示します。ただし、ポスチャ アセスメントがいくつかのマシンの値を戻す方法が原因で、スクリプトは一致する。この場合、Cisco バグ ID [CSCuu85646](#) のための修正がおよび [CSCuh67472](#) は必要となります。