

Exemplo de configuração das funções avançadas DAP

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Exemplo OU-baseado do fósforo](#)

[Exemplo da membrasia do clube \(memberOf\)](#)

[CheckAndMsg com função feita sob encomenda](#)

[Anti-vírus, Anti-spyware, e exemplos do Firewall](#)

[Verifique para ver se há a instalação anti-vírus](#)

[Verifique para ver se há a instalação anti-vírus e a última atualização, e forneça Mensagens de Erro](#)

[Verifique para ver se há a instalação de Anti-spyware](#)

[Verifique para ver se há a instalação de Firewall](#)

[Verifique para ver se há anti-vírus, o Anti-spyware, ou a instalação de Firewall](#)

[Termine se nenhuma instalação do spyware](#)

[Verifique para ver se há a instalação anti-vírus e do Firewall, e valide a última atualização anti-vírus não é maior de trinta dias](#)

[Harmonização da expressão regular](#)

[Conecte se o valor-limite PC tem qualquer exemplo do hotfix KB944](#)

[Script do uso LUA para verificar o OUI do MAC address](#)

[Conecte baseado nas primeiras três letras do hostname \(não diferenciando maiúsculas e minúsculas\)](#)

[Conecte se Device.id do valor-limite PC e do número de série no certificado são os mesmos](#)

[Reforce o DAP baseado na varredura do host CSD para a chave de registro do domínio](#)

[Apoio DAP para Windows 7 e CSD 3.5](#)

[Identificação dos iPhones, dos iPads, e dos dispositivos móveis](#)

[Uso do DAP impedir a conexão pelo navegador específico](#)

[Caveats](#)

[FAQ](#)

[Por que meu LUA passa pelo processo de script o trabalho para alguns usuários, mas não para tudo?](#)

Introdução

Este documento descreve as funções avançadas das políticas do acesso dinâmico (DAP) para acessos remoto VPN. Você pode usar estas funções avançadas quando você precisa a flexibilidade adicional combinar por critérios.

Nota: Consulte [Informações Importantes sobre Comandos de Depuração](#) antes de usar comandos **debug**.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Uma compreensão do DAP básico é exigida. Refira o [guia de distribuição das políticas do guia de distribuição das políticas do acesso dinâmico ASA 8.x \(DAP\)](#) (documentação do apoio) ou do [acesso dinâmico ASA 8.x \(DAP\)](#) (a comunidade do apoio).
- Uma boa compreensão da programação de Lua é igualmente benéfica. Refira os materiais de programação de Lua disponíveis na Web.

Componentes Utilizados

Este documento não é restringido à versão de software e hardware específica, mas o Security Device Manager adaptável (ASDM) é exigido a fim terminar a configuração.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Informações de Apoio

Cuidado: Use funções feitas sob encomenda avançadas DAP de Lua somente se a configuração GUI ASDM ou a função EVAL não fornecem o comportamento que de harmonização você precisa. Nas distribuições do produto, use funções avançadas de Lua com extremo cuidado e com orientação do planejamento de Cisco/centro de assistência técnica (TAC) a fim evitar todo o comportamento sem intenção com DAP.

Se você usa o DAP para acessos remoto VPN, você pode precisar a flexibilidade adicional combinar por critérios. Por exemplo, você pode aplicar um DAP diferente baseado nestas encenações:

- Uma unidade organizacional (OU) ou o outro nível da hierarquia contêm o objeto do usuário.
- Um nome do grupo (memberOf) adere à convenção de nomeação mas tem muitos fósforos possíveis, assim que você quer usar um convite em nomes do grupo.

- Você quer verificar para ver se há anti-vírus, o anti-spyware, ou pacotes de Firewall no valor-limite PC.
1. Use o ASDM a fim criar uma expressão lógica para seus critérios de verificação de repetição de dados.
 2. Use o modo avançado a fim criar funções feitas sob encomenda com uma expressão lógica e um código de Lua.

Exemplo OU-baseado do fósforo

O server do Lightweight Directory Access Protocol (LDAP) pode retornar muitos atributos que o DAP pode usar em uma expressão lógica.

Para um exemplo destes atributos, use o **comando trace do dap debugar no console adaptável da ferramenta de segurança (ASA)**.

```
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
```

Um nome destacado (DN) do usuário é um atributo retornado pelo servidor ldap. O DN identifica implicitamente onde o objeto do usuário é ficado situado no diretório. Por exemplo, se o DN é usuário de CN=Joe, OU=Admins, dc=cisco, dc=com, este usuário é ficado situado em OU=Admins, dc=cisco, dc=com. Se todos os administradores estão neste OU (ou em algum recipiente abaixo deste nível), use esta expressão lógica a fim combinar nos critérios:

```
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)()
```

Neste exemplo, a função string.find permite uma expressão regular. \$ na extremidade da corda ancora esta corda à extremidade do campo do distinguishedName.

Use o caractere de escape % em seu caractere de texto de pesquisa a fim escapar caracteres especiais como (). % + - *? [^ \$. Por exemplo, você pode escapar - o caráter nesta corda (OU=Admins, dc=my-domain, dc=com\$) segundo as indicações desta corda (OU=Admins, dc=my%-domain, dc=com\$).

Exemplo da membrasia do clube (memberOf)

Você pode criar uma expressão lógica similar, básica para a correspondência de padrão da membrasia do clube do diretório ativo (AD). Porque os usuários podem ser membros dos grupos múltiplos, o DAP analisa gramaticalmente a resposta do servidor ldap em entradas separadas e realiza-as em uma tabela. Neste caso, uma função mais avançada é exigida:

- Compare o campo do memberOf como uma corda se o usuário pertence a somente um grupo.
- Itere através de cada campo retornado do memberOf se os dados retornados são do tipo "tabela."

Por exemplo, se um usuário é um membro de qualquer grupo que termine com "- stu," combinam este DAP:

```
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())
```

CheckAndMsg com função feita sob encomenda

Esta função usa o DAP com grupo da ação para terminar:

```
(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())
```

Em cima da terminação, indica esta mensagem:

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

Anti-vírus, Anti-spyware, e exemplos do Firewall

Verificação destas funções de Lua para os atributos relativos a anti-vírus, ao anti-spyware, e aos pacotes do Firewall no valor-limite PC retornado pela varredura do host do Cisco Secure Desktop (CSD).

Verifique para ver se há a instalação anti-vírus

Verificações desta função do costume se o CSD detecta anti-vírus:

```
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())
```

Verifique para ver se há a instalação anti-vírus e a última atualização, e forneça Mensagens de Erro

Este exemplo demonstra como o DAP pode verificar para ver se há uma instalação anti-vírus, verificar para ver se há a última atualização, e notificar o usuário para a remediação. Usa uma função similar àquela na [verificação para a instalação anti-vírus](#).

Ajuste o Authentication, Authorization, and Accounting (AAA) atribui-o desejo para combinar. No campo avançado, assegure-se de que E a operação esteja selecionada; no campo de ação, assegure-se de que a opção da terminação esteja selecionada. Se o usuário combina os atributos AAA e se a função de Lua retorna um valor de verdadeiro, o DAP está selecionado, uma mensagem aparece que explique porque o registro DAP foi indicado, e a conexão do usuário é terminada. Se a função de Lua não retorna um valor de verdadeiro, o DAP não combina e permite o acesso. No campo da caixa de mensagem, incorpore a mensagem, “nenhum programa anti-vírus encontrado, instale por favor anti-vírus e a tentativa outra vez.” Se o usuário tem um pacote anti-vírus e está abaixo do ponto inicial dos dias da atualização, não estão dados uma mensagem como indicado pelas aspas duplas na linha 7 deste exemplo:

```
(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())
```

Se o usuário tem Norton anti-vírus, mas a última atualização é maior de 15 dias, este exemplo de mensagem aparece:

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

Se o EVAL não combina, vai à função seguinte, fósforos, e retorna um valor de verdadeiro. Desde que não há nenhum CheckAndMsg associado com a segunda função, usa o texto de mensagem DAP:

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

Em resumo, o DAP procura um usuário AAA e o atributo do valor-limite a fim combinar o DAP. Se o DAP combina, o usuário está terminado com uma mensagem. O fósforo do valor-limite é um resultado de um Lua EVAL que retorne verdadeiro ou falso ao DAP. Os fósforos verdadeiros e negam a conexão. Um falso não combina e permite a conexão.

1. A primeira função nas verificações de laço se endpoint.av.xxxxx.exists é igual retificar e se a última atualização é menos do que os dias configurados. Os usuários sem software anti-vírus são acesso permitido, porque os fósforos do usuário AAA, mas o Lua olham especificamente para endpoint.av.xxxxx.exists = retificam e dias do <= endpoint.av.xxxxx.lastupdate.
2. O segundo laço trava usuários sem software anti-vírus e obstrui-os, porque a segunda função olha somente para endpoint.av.xxxxx.exists NE verdadeiro. Se o usuário que o valor-limite avoirdupois existe não é igual retificar, a função retorna um valor de verdadeiro, que significa que não tem anti-vírus. O DAP combina e nega a conexão.

Verifique para ver se há a instalação de Anti-spyware

Verificações desta função do costume se o CSD detecta o anti-spyware:

```
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)()
```

Verifique para ver se há a instalação de Firewall

Verificações desta função do costume se o CSD detecta um Firewall:

```
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)()
```

Verifique para ver se há anti-vírus, o Anti-spyware, ou a instalação de Firewall

Esta função retorna um verdadeiro se um anti-vírus, o anti-spyware ou um pacote do Firewall são encontrados:

```
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()
```

Termine se nenhuma instalação do spyware

A única diferença entre esta função e a função na [verificação para a instalação do Anti-spyware](#) é que “não” precede a afirmação.

```
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())
```

Verifique para ver se há a instalação anti-vírus e do Firewall, e valide a última atualização anti-vírus não é maior de trinta dias

Este exemplo retorna verdadeiro se um anti-vírus e um Firewall são encontrados e se a última atualização do anti-vírus não é maior de 30 dias:

```
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())
```

Porque o Firewall não tem um valor do lastupdate a retornar, tem uma função separada.

Harmonização da expressão regular

Esta seção descreve as funções que usam expressões do regex a fim combinar determinados atributos e determinar a validade da máquina host. Estas capacidades do regex foram testadas e são válidas:

- O sinal de dólar (\$) ancora o caractere de texto de pesquisa ao fim do valor exibido.
- O sinal de intercalação (^) ancora o caractere de texto de pesquisa ao começo do valor exibido.
- Os caracteres suportados, tais como o [Aa], combinam caracteres múltiplos em uma posição específica. Por exemplo, a fim combinar o OU=Cisco (não diferenciando maiúsculas e minúsculas), [Oo] do [Cc] do [Ss] do [li] do [Cc] do uso OU=.
- O período (.) combina todo o único caráter nesta posição. Por exemplo, grupo. Fósforos Group01Users dos usuários, Group33Users, e assim por diante.

Conecte se o valor-limite PC tem qualquer exemplo do hotfix KB944

Esta função usa a expressão regular que combina a fim considerar se a lista do hotfix contém um teste padrão. Neste exemplo, o Cisco Secure Desktop retorna todos os hotfix no valor-limite PC; se há um exemplo de KB944, a política DAP combina e está reforçada.

```
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)()
```

Por exemplo, se a máquina host tem o hotfix KB944533 ou o hotfix KB944653, combina a regra.

Script do uso LUA para verificar o OUI do MAC address

Esta função é similar a essa descrita dentro [conecta se o valor-limite PC tem qualquer exemplo do hotfix KB944](#). Esta função usa uma expressão regular a fim combinar de organização o identificador exclusivo (OUI) do MAC address.

Neste exemplo, o MAC address começa com d067.e5XX.XX. Use uma expressão regular e um código de Lua a fim combinar as máquinas que começam com o mesmo OUI MAC.

```
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
end())
```

Nota: Uma versão diferente desta função é exigida para uma verificação multi-avaliada.

Conecte baseado nas primeiras três letras do hostname (não diferenciando maiúsculas e minúsculas)

Esta função usa expressões regulares a fim determinar se as primeiras três letras do hostname são msv (não diferenciando maiúsculas e minúsculas):

```
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())
```

Conecte se Device.id do valor-limite PC e do número de série no certificado são os mesmos

Esta expressão de Lua está significando conectar se os device.id do valor-limite PC e do número de série no certificado são os mesmos:

```
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 ) ()
```

Nota: O uso do caractere wildcard (*) não trabalha nesta função particular (o ["*"]endpoint.certificate.user não faz trabalho). Você deve recuperar cada par quilovolt individualmente e analisá-lo gramaticalmente com ele.

Reforce o DAP baseado na varredura do host CSD para a chave de registro do domínio

Este procedimento fornece um exemplo de um procedimento de configuração o ASDM.

1. Encontre a chave de registro que guarda em do domínio \ HKEY_LOCAL_MACHINE \ SISTEMA \ CurrentControlSet \ serviços \ Tcipip \ parâmetros \ domínio.
2. Defina o parâmetro da varredura do host para a configuração de registro.
3. Aplique o atributo do valor-limite do registro à política DAP.
4. Estabeleça a sessão de VPN do secure sockets layer (SSL).
5. Verifique que o reforço de política DAP através do DAP debuga.

Apoio DAP para Windows 7 e CSD 3.5

As Plataformas de Windows 7 são apoiadas com a liberação 3.5 CSD ou mais atrasado. Com a versão de manutenção ASDM 6.2.x e as liberações 6.3.x, você pode diretamente usar a relação a fim verificar para ver se há o OS de Windows 7. Com liberações mais adiantadas ASDM, um script avançado DAP Lua é exigido a fim verificar para ver se há máquinas de Windows 7. Em um ASA com liberação 8.x e PRE-beta CSD libere 3.5, entram nesta corda do script de Lua na caixa avançada DAP ASDM a fim verificar máquinas de Windows 7:

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

Identificação dos iPhones, dos iPads, e dos dispositivos móveis

Esta expressão de Lua deixa-o seguir dispositivos móveis específicos por seus identificadores exclusivos (UID). Você pode usar o DAP a fim conseguir esta funcionalidade básica.

Quando o valor não pode duro-ser codificado e necessidades de ser lido do AD, este torna-se mais difícil. Porque não há nenhum campo específico UID no AD, você pode armazenar o valor para um usuário particular sob um campo diferente. Este exemplo usa o otherHomePhone para armazenar o UID.

Para ajudá-lo a identificar o UID para um iPhone ou um iPad, procure a Web por uma ferramenta apropriada.

Uma vez que você identifica o UID, adicionar-lo ao otherHomePhone na entrada AD para esse usuário.

Do comando do **ldap 255 debugar** e da autenticação de teste do usuário, obtenha o atributo LDAP que está sendo empurrado, que é `otherHomePhone`.

Permita que o telefone conecte, a seguir execute um traço DAP durante a conexão tentada a fim identificar o atributo do valor-limite que contém o UID (`endpoint.anyconnect.deviceuniqueid`).

Esta expressão de Lua pode então comparar os dois parâmetros:

```
assert(function()  
if (type(aaa.ldap.otherHomePhone) ==type(endpoint.anyconnect.deviceuniqueid)  
then  
return true  
end  
return false  
end)()
```

Uso do DAP impedir a conexão pelo navegador específico

Este procedimento descreve como usar o DAP para impedir a conexão por um navegador de Chrome:

1. Permita o CSD.
2. Sob a configuração da varredura do host, use o processo ID (PID) e o nome de processo a fim adicionar uma varredura do processo.

Você pode determinar o PID e o nome de processo em Windows com o gerenciador de tarefa. A fim indicar o valor PID, para abrir o **gerenciador de tarefa**, vai à aba dos **processos**, clica o menu da **vista**, a seguir clica **colunas seletas**. Nas colunas seletas ou selecione colunas diálogo da página do processo, tiquetaqueie e verifique a caixa de seleção para ver se há **PID (identificador do processo)**, e clique a **APROVAÇÃO**.

Em Mac, você pode determinar o processo ID com o monitor de atividade. Ou, no shell da festança (que você pode enlatar igualmente se usa em Unix), use o **picosegundo** - comando **e** quando o processo for executado, a seguir para combinar os PID ao nome de processo com o comando de **/proc/ <PID>/cmdline do gato**.

3. Crie uma política DAP a fim testar, entre outras coisas, se esse processo está sendo executado na máquina.

4. Teste sua conexão.

Caveats

1. O problema com esta solução é que o usuário não pode ter Chrome aberto em sua máquina de todo. O DAP verifica simplesmente se esse processo de Chrome do detalhe está sendo executado, mas não o verifica para ver se a sessão dos sem clientes foi iniciada por esse processo ou por algum outro processo. Assim o usuário não pode executar Chrome no fundo quando uma conexão VPN da Web é tentada.
2. Supõe uma encenação onde o usuário use Firefox para abrir uma sessão de VPN da Web, e a tentativa de login falha. O usuário recorda que Chrome ainda está sendo executado, assim que o usuário termina a conexão de Chrome e tenta-a entrar outra vez. O início de uma sessão ainda falha porque o CSD tem que tornar a colocar em funcionamento a varredura do host. Assim, o usuário deve igualmente fechar o exemplo de Firefox que foi usado para alcançar o WebVPN, a seguir reinicia Firefox. Este processo pode ser desconcertante ao usuário. Cisco recomenda que você cria um mensagem de falha DAP que diga o usuário para terminar Chrome e para fechar o navegador que se estão usando atualmente:

FAQ

Esta seção dá uma resposta a uma mais frequentemente das perguntas feitas considera dentro a informação que é descrita neste documento.

Por que meu LUA passa pelo processo de script o trabalho para alguns usuários, mas não para tudo?

Considere este script LUA:

```
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  
end  
return false  
end)()
```

Este script é projetado a fim combinar a loja do certificado e o assunto DC que é encontrado no certificado. Contudo, este script foi testado com máquinas múltiplas e encontrado para trabalhar em algumas máquinas, mas para falhar em muitas outro.

A razão que este script trabalha somente é intermitentemente devido à maneira que hostscan retorna os valores. Quando você vê o traço DAP que não trabalha, você pode ver que o *subject_dc* retorna valores múltiplos pelo certificado. Você pode igualmente ver que o último valor retornado não é *Home Depot*.

```
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"
```

Quando você vê o traço DAP que trabalha, este pode ser observado:

```
DAP_TRACE: endpoint.certificate.user["20"] = {}  
DAP_TRACE: endpoint.certificate.user["20"].subject_cn = "JHD0C6"  
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"
```

Isto indica que o script LUA trabalha corretamente. Contudo, devido à maneira que a postura-avaliação retorna os valores em algumas máquinas, o script não combina. Neste caso, os reparos para o Bug da Cisco ID [CSCuu85646](#) e [CSCuh67472](#) são exigidos.