

Ejemplo de configuración de las funciones avanzadas DAP

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Ejemplo OU-basado de la coincidencia](#)

[Ejemplo de la membresía del grupo \(memberOf\)](#)

[CheckAndMsg con la función de encargo](#)

[Contra virus, Anti-Spyware, y ejemplos del Firewall](#)

[Comprobación para la instalación del contra virus](#)

[La comprobación para la instalación del contra virus y la actualización más reciente, y proporciona los mensajes de error](#)

[Comprobación para la instalación del Anti-Spyware](#)

[Comprobación para la instalación del Firewall](#)

[Comprobación para el contra virus, el Anti-Spyware, o la instalación del Firewall](#)

[Termine si ninguna instalación del Spyware](#)

[Marque para saber si hay instalación del contra virus y del Firewall, y valide el contra virus más reciente que la actualización no es mayor de treinta días](#)

[El corresponder con de la expresión normal](#)

[Conecte si el punto final PC tiene cualquier caso del hotfix KB944](#)

[Utilice el script LUA para marcar el OUI de la dirección MAC](#)

[Conecte basado en las primeras tres cartas del nombre de host \(no con diferenciación entre mayúsculas y minúsculas\)](#)

[Conecte si Device.id del punto final PC y del número de serie en el certificado son lo mismo](#)

[Aplique el DAP basado en la exploración del host CSD para la clave de registro del dominio](#)

[Soporte DAP para Windows 7 y CSD 3.5](#)

[Identificación de los iPhones, de los iPads, y de los dispositivos móviles](#)

[Uso del DAP de prevenir la conexión del navegador específico](#)

[Advertencias](#)

[FAQ](#)

[¿Por qué mi script LUA trabaja para algunos usuarios, pero no para todos?](#)

Introducción

Este documento describe las funciones avanzadas de las directivas del acceso dinámico (DAP)

para los VPN de accesos remotos. Usted puede utilizar estas funciones avanzadas cuando usted necesita la flexibilidad adicional para hacer juego por los criterios.

Nota: Consulte [Información Importante sobre Comandos de Debug](#) antes de usar un comando debug.

Prerrequisitos

Requisitos

Cisco recomienda que tenga conocimiento sobre estos temas:

- Una comprensión del DAP básico se requiere. Refiera al [Guía de despliegue de las directivas del Guía de despliegue de las directivas del acceso dinámico ASA 8.x \(DAP\)](#) (documentación del soporte) o del [acceso dinámico ASA 8.x \(DAP\)](#) (comunidad del soporte).
- Una buena comprensión de la programación de Lua es también beneficiosa. Refiera a los materiales programados de Lua disponibles en la red.

Componentes Utilizados

Este documento no se restringe a las versiones de software y hardware específicas, pero requieren al Administrador de dispositivos de seguridad adaptante (ASDM) para completar la configuración.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Antecedentes

Precaución: Utilice las funciones de encargo avanzadas DAP de Lua solamente si la Configuración del GUI del ASDM o la función EVAL no proporciona el comportamiento que corresponde con que usted necesita. En las implementaciones de producto, utilice las funciones avanzadas de Lua con el cuidado extremo y con la dirección de dirigir/Centro de Asistencia Técnica (TAC) de Cisco para evitar cualquier comportamiento involuntario con el DAP.

Si usted utiliza el DAP para los VPN de accesos remotos, usted puede necesitar la flexibilidad adicional para hacer juego por los criterios. Por ejemplo, usted puede aplicar un diverso DAP basado en estos escenarios:

- La unidad organizativa (OU) o el otro nivel de la jerarquía contiene el objeto de usuario.
- Un nombre del grupo (memberOf) se adhiere a la convención para nombres pero tiene

muchas coincidencias posibles, así que usted quiere utilizar a un comodín en los nombres del grupo.

- Usted quiere marcar para saber si hay contra virus, anti-spyware, o paquetes del Firewall en el punto final PC.

1. Utilice el ASDM para crear una expresión lógica para sus criterios de concordancia.

2. Utilice el modo avanzado para crear las funciones de encargo con una expresión lógica y un código de Lua.

Ejemplo OU-basado de la coincidencia

El servidor del Lightweight Directory Access Protocol (LDAP) puede volver muchos atributos que el DAP pueda utilizar en una expresión lógica.

Por un ejemplo de estos atributos, utilice el **comando trace del dap del debug** en la consola adaptante del dispositivo de seguridad (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
```

Un Nombre distintivo (DN) del usuario es un atributo vuelto por el servidor LDAP. El DN identifica implícito donde el objeto de usuario está situado en el directorio. Por ejemplo, si el DN es usuario de CN=Joe, OU=Admins, dc=cisco, dc=com, este usuario está situado en OU=Admins, dc=cisco, dc=com. Si todos los administradores están en este OU (o cualquier envase debajo de este nivel), utilice esta expresión lógica para hacer juego en los criterios:

```
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)()
```

En este ejemplo, la función string.find permite una expresión normal. El \$ en el extremo de la cadena asegura esta cadena al final del campo del distinguishedName.

Utilice el carácter de escape % en su cadena de búsqueda para escapar los caracteres especiales por ejemplo (). ¿% + - *? [^ \$. Por ejemplo, usted puede escapar - el carácter en esta cadena (OU=Admins, dc=my-domain, dc=com\$) tal y como se muestra en de esta cadena (OU=Admins, dc=my%-domain, dc=com\$).

Ejemplo de la membresía del grupo (memberOf)

Usted puede crear una expresión lógica similar, básica para la coincidencia de patrones de la membresía del grupo del Active Directory (AD). Porque los usuarios pueden ser miembros de los múltiples grupos, el DAP analiza la respuesta del servidor LDAP en las entradas separadas y las lleva a cabo en una tabla. En este caso, se requiere una función más avanzada para:

- Compare el campo del memberOf como cadena si el usuario pertenece a solamente un grupo.
- Itere a través de cada campo vuelto del memberOf si los datos devueltos son de tipo "tabla."

Por ejemplo, si un usuario es un miembro de cualquier grupo con quien termina "- stu," él hace juego 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 con la función de encargo

Esta función utiliza el DAP con el conjunto de la acción 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)())
```

Sobre la terminación, visualiza este mensaje:

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

Contra virus, Anti-Spyware, y ejemplos del Firewall

Control de estos funcionamientos de Lua para los atributos relacionados con el contra virus, el anti-spyware, y los paquetes del Firewall en el punto final PC devuelto por la exploración del host del (CSD) del Cisco Secure Desktop.

Comprobación para la instalación del contra virus

Controles de este funcionamiento de la aduana si el CSD detecta cualquier contra virus:

```
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)()
```

La comprobación para la instalación del contra virus y la actualización más reciente, y proporciona los mensajes de error

Este ejemplo demuestra cómo el DAP puede marcar para saber si hay una instalación del contra virus, marcar para saber si hay la actualización más reciente, y notificar al usuario para la corrección. Utiliza una función similar a ésta en la [comprobación para la instalación del contra virus](#).

Fije el Authentication, Authorization, and Accounting (AAA) le atribuye deseo para hacer juego. En el campo avanzado, asegúrese que Y la operación esté seleccionada; en el campo de acción, asegúrese que la opción del terminal esté seleccionada. Si el usuario hace juego los atributos AAA y si la función de Lua vuelve un valor de verdad, se selecciona el DAP, un mensaje aparece que explica porqué el expediente DAP fue visualizado, y se termina la conexión del usuario. Si la función de Lua no vuelve un valor de verdad, el DAP no hace juego y el acceso de los permisos. En el campo de la casilla de mensaje, ingrese el mensaje, “ningún programa de contra virus encontrado, instale por favor el contra virus y el intento otra vez.” Si el usuario tiene un paquete del contra virus y está debajo del umbral de los días de la actualización, no se dan un mensaje según lo indicado por las comillas dobles en la línea 7 de este ejemplo:

```
(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)())
```

Si el usuario tiene contra virus del Norton, pero la actualización más reciente es mayor de 15 días, este mensaje de ejemplo aparece:

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

Si el EVAL no hace juego, va a la función siguiente, las coincidencias, y vuelve un valor de verdad. Puesto que no hay CheckAndMsg asociado a la segunda función, utiliza el mensaje de texto DAP:

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

En resumen, el DAP busca un usuario AAA y el atributo del punto final para hacer juego el DAP. Si el DAP hace juego, terminan al usuario con un mensaje. La coincidencia del punto final es un resultado de un Lua EVAL que vuelva verdad o falso al DAP. Las coincidencias verdaderas y niegan la conexión. Un falso no hace juego y permite la conexión.

1. La primera función en los controles de loop si endpoint.av.xxxxx.exists es igual verdad y si la actualización más reciente es menos que los días configurados. Los usuarios sin el software del contra virus son acceso permitido, porque el usuario AAA hace juego, pero el Lua mira específicamente para endpoint.av.xxxxx.exists = verdad y los días del <= endpoint.av.xxxxx.lastupdate.
2. El segundo loop coge a los usuarios sin el software del contra virus y los bloquea, porque la segunda función mira solamente para endpoint.av.xxxxx.exists NE verdad. Si el usuario que existe el punto final sistema de pesos americano no es igual verdad, la función vuelve un valor de verdad, que significa que él no tiene contra virus. El DAP hace juego y niega la conexión.

Comprobación para la instalación del Anti-Spyware

Controles de este funcionamiento de la aduana si el CSD detecta el 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)()
```

Comprobación para la instalación del Firewall

Controles de este funcionamiento de la aduana si el CSD detecta un 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)()
```

Comprobación para el contra virus, el Anti-Spyware, o la instalación del Firewall

Esta función vuelve un verdad si se encuentra un contra virus, el anti-spyware o un paquete del Firewall:

```
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
```

```

end
end
return false
end
return (check(endpoint.av) or check(endpoint.fw) or check(endpoint.as))
end) ()

```

Termine si ninguna instalación del Spyware

La única diferencia entre esta función y la función en la [comprobación para la instalación del Anti-Spyware](#) es que “no” precede la afirmación.

```

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) ()

```

La comprobación para la instalación del contra virus y del Firewall, y valida el contra virus más reciente que la actualización no es mayor de treinta días

Este ejemplo vuelve verdad si se encuentran un contra virus y un Firewall y si la actualización más reciente del contra virus no es mayor de 30 días:

```

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 el Firewall no tiene un valor del lastupdate a volver, tiene una función separada.

El corresponder con de la expresión normal

Esta sección describe las funciones que utilizan las expresiones del regex para corresponder con

ciertos atributos y determinar la validez del equipo del host. Estas capacidades del regex se han probado y son válidas:

- La muestra de dólar (\$) asegura la cadena de búsqueda al final del valor devuelto.
- El signo de intercalación (^) asegura la cadena de búsqueda al principio del valor devuelto.
- Los caracteres acorchetados, tales como [Aa], hacen juego los caracteres múltiples en una posición específica. Por ejemplo, para hacer juego el ou=Cisco (sin diferenciación entre mayúsculas y minúsculas), [Oo] del [Cc] del [Ss] del [Ii] del [Cc] del uso OU=.
- El período (.) hace juego cualquier solo carácter en esta posición. Por ejemplo, grupo. Coincidencias Group01Users de los usuarios, Group33Users, y así sucesivamente.

Conecte si el punto final PC tiene cualquier caso del hotfix KB944

Esta función utiliza la expresión normal que corresponde con para considerar si la lista del hotfix contiene un modelo. En este ejemplo, el Cisco Secure Desktop vuelve todo el hotfixes en el punto final PC; si hay un caso de KB944, la directiva DAP hace juego y se aplica.

```
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 ejemplo, si el equipo del host tiene el hotfix KB944533 o hotfix KB944653, hace juego la regla.

Utilice el script LUA para marcar el OUI de la dirección MAC

Esta función es similar a la que está descrita adentro [conecta si el punto final PC tiene cualquier caso del hotfix KB944](#). Esta función utiliza una expresión normal para hacer juego de organización el Identificador único (OUI) de la dirección MAC.

En este ejemplo, la dirección MAC comienza con d067.e5XX.XX. Utilice una expresión normal y un código de Lua para hacer juego las máquinas que comienzan con el mismo 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) ()
```

Nota: Una diversa versión de esta función se requiere para marcar polivalente.

Conecte basado en las primeras tres cartas del nombre de host (no con diferenciación entre mayúsculas y minúsculas)

Esta función utiliza las expresiones normales para determinar si las primeras tres cartas del nombre de host son msv (sin diferenciación entre mayúsculas y 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 si Device.id del punto final PC y del número de serie en el certificado son lo mismo

Esta expresión de Lua se significa para conectar si los device.id del punto final PC y del número de serie en el certificado son lo mismo:

```
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: El uso del carácter comodín (*) no trabaja en esta función determinada (el ["*"]endpoint.certificate.user no hace trabajo). Usted debe extraer cada par kilovoltio

individualmente y analizar con él.

Aplique el DAP basado en la exploración del host CSD para la clave de registro del dominio

Este procedimiento proporciona un ejemplo de un Procedimiento de configuración con el ASDM.

1. Localice la clave de registro que sostiene en \ HKEY_LOCAL_MACHINE \ SISTEMA \ CurrentControlSet \ servicios \ Tcpip \ parámetros \ dominio del dominio.
2. Defina el parámetro de la exploración del host para la configuración del registro.
3. Aplique el atributo del punto final del registro a la directiva DAP.
4. Establezca a la sesión de VPN de Secure Sockets Layer (SSL).
5. Verifique la aplicación de políticas DAP vía los debugs DAP.

Soporte DAP para Windows 7 y CSD 3.5

Las Plataformas de Windows 7 se soportan con la versión 3.5 CSD o más adelante. Con la versión de mantenimiento del ASDM 6.2.x y las versiones 6.3.x, usted puede utilizar directamente la interfaz para marcar para saber si hay Windows 7 OS. Con versiones anteriores del ASDM, un script avanzado DAP Lua se requiere para marcar para saber si hay máquinas de Windows 7. En un ASA con la versión 8.x y la versión PRE-beta 3.5 CSD, ingrese esta cadena del script de Lua en el rectángulo avanzado DAP del ASDM para realizar las comprobaciones para Windows 7 máquinas:

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

Identificación de los iPhones, de los iPads, y de los dispositivos móviles

Esta expresión de Lua le deja seguir los dispositivos móviles específicos por sus Identificadores únicos (uid). Usted puede utilizar el DAP para alcanzar esta funcionalidad básica.

Cuando el valor no se puede poner en hard-code y las necesidades de ser leído en el AD, esto llega a ser más difícil. Porque no hay campo específico del UID en el AD, usted puede salvar el valor para un usuario determinado bajo diverso campo. Este ejemplo utiliza el otherHomePhone para salvar el UID.

Para ayudarle a identificar el UID para un iPhone o un iPad, busque la red para una herramienta apropiada.

Una vez que usted identifica el UID, agreguelo al otherHomePhone en la entrada AD para ese usuario.

Del comando del **ldap 255 del debug** y de la prueba de la autenticación del usuario, obtenga el atributo LDAP que es avanzado, que es otherHomePhone.

Permita que el teléfono conecte, después ejecute una traza DAP durante la conexión frustrada para identificar el atributo del punto final que contiene el UID (endpoint.anyconnect.deviceuniqueid).

Esta expresión de Lua puede entonces comparar los dos parámetros:

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

Uso del DAP de prevenir la conexión del navegador específico

Este procedimiento describe cómo utilizar el DAP para prevenir la conexión de un navegador de Chrome:

1. Permiso CSD.
2. Bajo configuración de la exploración del host, utilice el identificador de proceso (PID) y el nombre del proceso para agregar una exploración de proceso.

Usted puede determinar el PID y el nombre del proceso en Windows con el administrador de tareas. Para visualizar el valor PID, abra el **administrador de tareas**, vaya a la pestaña de los **procesos**, haga clic en el menú de la **visión**, después haga clic en las **columnas selectas**. En las columnas selectas o las columnas de proceso selectas de la página de diálogo, haga clic y marque el checkbox para **PID (identificador de proceso)**, y haga clic en **AUTORIZACIÓN**.

En los Macs, usted puede determinar el identificador de proceso con el monitor de actividad. O, en el shell del golpe (que usted puede poder también utilizar en Unix), utilice el **picosegundo** - comando **e** mientras que el proceso se está ejecutando, después para hacer juego los PID al nombre del proceso con el comando de **/proc/ <PID>/cmdline del gato**.

3. Cree una directiva DAP para probar, entre otras cosas, si ese proceso se está ejecutando en la máquina.
4. Pruebe su conexión.

Advertencias

1. El problema con esta solución es que el usuario no puede tener Chrome abierto en su máquina en absoluto. El DAP marca simplemente si ese proceso de Chrome del detalle se está ejecutando, pero no marca para ver si la sesión del clientless fue iniciada por ese proceso o por un cierto otro proceso. El usuario no puede ejecutar tan Chrome en el fondo cuando se intenta una conexión WebVPN.
2. Asuma un escenario donde el usuario utiliza Firefox para abrir a una sesión WebVPN, y el intento de inicio de sesión falla. El usuario recuerda que Chrome todavía se está ejecutando, así que el usuario termina la conexión de Chrome e intenta iniciar sesión otra vez. El login todavía falla porque el CSD tiene que volver a efectuar la exploración del host. Así pues, el usuario debe también cerrar el caso de Firefox que fue utilizado para acceder el WebVPN, después recomienza Firefox. Este proceso puede ser confuso al usuario. Cisco recomienda que usted crea un mensaje de error DAP que diga al usuario terminar Chrome y cerrar al navegador que están utilizando actualmente:

FAQ

Esta sección proporciona una respuesta a una lo más frecuentemente de las preguntas hechas adentro mira la información que se describe en este documento.

¿Por qué mi script LUA trabaja para algunos usuarios, pero no para todos?

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  
  return false  
end)()
```

Este script se diseñó para hacer juego el almacén de certificados y el tema DC que se encuentra en el certificado. Sin embargo, este script se ha probado con las máquinas múltiples y se ha encontrado para trabajar en algunas máquinas, sino para fallar en muchas otras.

La razón que este script trabaja solamente está intermitentemente debido a la manera que hostscan vuelve los valores. Cuando usted ve la traza DAP que no trabaja, usted puede ver que el *subject_dc* vuelve los valores múltiples por el certificado. Usted puede también ver que el valor devuelto más reciente no es *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"
```

Cuando usted ve la traza DAP que trabaja, esto puede ser observada:

```
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"
```

Esto indica que el script LUA trabaja correctamente. Sin embargo, debido a la manera que la postura-evaluación vuelve los valores en algunas máquinas, el script no hace juego. En este caso, los arreglos para el bug Cisco ID [CSCuu85646](#) y [CSCuh67472](#) se requieren.