DAP усовершенствованный пример конфигурации функций

Содержание

Введение

Этот документ описывает усовершенствованные функции политиков динамического доступа (DAP) для VPN для удаленного доступа. Можно использовать эти усовершенствованные функции при необходимости в дополнительной гибкости для соответствия по критериям.

> **Примечание:** Прежде чем выполнять какие-либо команды отладки , ознакомьтесь с документом "Важные сведения о командах отладки".

Предварительные условия

Требования

Компания Cisco рекомендует предварительно ознакомиться со следующими предметами:

- Понимание основного DAP требуется. См. любой ASA 8.x Руководство по развертыванию политиков динамического доступа (DAP) (поддерживают документацию), или ASA 8.x Руководство по развертыванию политиков динамического доступа (DAP) (сообщество поддержки).
- Хорошее понимание программирования Lua также выгодно. См. Lua программирование имеющихся в сети материалов.

Используемые компоненты

Этот документ не ограничен определенными версиями программного и аппаратного обеспечения, но Менеджер устройств адаптивной безопасности (ASDM) (ASDM) требуется для завершения конфигурации.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Общие сведения

> **Внимание. :** Используйте усовершенствованные пользовательские функции Lua DAP, только если конфигурация GUI ASDM или функция EVAL не предоставляют поведение при сравнении, в котором вы нуждаетесь. В развертываниях на производстве используйте, усовершенствовал функции Lua с большой осторожностью и с руководством от Разработки/центра технической поддержки (TAC) Cisco во избежание любого непреднамеренного поведения с DAP.

При использовании DAP для VPN для удаленного доступа вам, возможно, понадобится дополнительная гибкость для соответствия по критериям. Например, можно применить другой DAP на основе этих сценариев:

- Подразделение (OU) или другой уровень иерархии содержат объект пользователя.

- Имя группы (memberOf) придерживается соглашения о записи имен, но имеет много возможных соответствий, таким образом, вы хотите использовать подстановочный знак на именах групп.
- Вы хотите проверить для антивируса, программы обезвреживания шпионского ПО или пакетов межсетевого экрана на ПК оконечной точки.

1. Используйте ASDM для создания логического выражения для условий соответствия.

2. Используйте усовершенствованный режим для создания пользовательских функций с кодом Lua и логическим выражением.

Основанный на OU пример соответствия

Сервер Протокола LDAP может возвратить много атрибутов, которые DAP может использовать в логическом выражении.

Для примера этих атрибутов используйте **отладку dap** команда **трассировки** на консоли Устройства адаптивной защиты (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
```

Составное имя (DN) пользователя является одним атрибутом, возвращенным Сервером LDAP. DN неявно определяет, где объект пользователя расположен в каталоге. Например, если DN является Пользователем CN=Joe, OU=Admins, dc=cisco, dc=com, этот пользователь расположен в OU=Admins, dc=cisco, dc=com. Если все администраторы находятся в этом OU (или любой контейнер ниже этого уровня), используйте это логическое выражение для соответствия на критериях:

```
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)()
```

В данном примере функция string.find обеспечивает регулярное выражение. $ в конце строки привязывает эту строку до конца distinguishedName поля.

Используйте % символа выхода в своей строке поиска для выхода из специальных символов такой как (). % + - *? [$ ^. Например, можно выйти - символ в этой строке (OU=Admins, dc=my-domain, dc=com$) как показано в этой строке (OU=Admins, dc=my %-домен, dc=com$).

Состав группы (memberOf) пример

Можно создать подобное, основное логическое выражение для совпадения с образцом состава группы Active Directory (AD). Поскольку пользователи могут быть участниками множественных групп, DAP анализирует ответ от Сервера LDAP в отдельные записи и держит их в таблице. В этом случае более усовершенствованная функция требуется чтобы к:

- Сравните memberOf поле как строку, если пользователь принадлежит только одной группе.
- Выполните итерации через каждого, возвратил memberOf поле, если возвращенные данные имеют тип 'таблица'.

Например, если пользователь является участником какой-либо группы, которая заканчивается '-stu', они совпадают с этим 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 с пользовательской функцией

Эта функция использует DAP с набором действия для завершения:

```
(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_threshhold, "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)())
```

После завершения это отображает это сообщение:

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

Антивирус, программа обезвреживания шпионского ПО и

примеры межсетевого экрана

Эти Lua функционируют проверка для атрибутов, отнесенных к антивирусу, программе обезвреживания шпионского ПО и пакетам межсетевого экрана на ПК оконечной точки, возвращенном просмотром хоста Cisco Secure Desktop (CSD).

Проверьте для антивирусной установки

Эта пользовательская функция проверяет, обнаруживает ли CSD какой-либо антивирус:

```
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)()
```

Проверьте для антивирусной установки и последнего обновления, и предоставьте сообщения об ошибках

Данный пример демонстрирует, как DAP может проверить для антивирусной установки, проверьте для последнего обновления и уведомьте пользователя для исправления. Это использует функцию, подобную этому под контролем для Антивирусной Установки.

Установите аутентификацию, авторизацию и учет (AAA), приписывает вас, хотят совпасть. В усовершенствованном поле гарантируйте, что выбрана операция И; в поле действия гарантируйте, что выбрана оконечная опция. Если пользователь совпадает с атрибутами AAA и если функция Lua возвращает значение истинных, DAP выбран, сообщение появляется, который объясняет, почему запись DAP была отображена, и подключение пользователя завершено. Если функция Lua не возвращает значение истинных, DAP не совпадает и разрешает доступ. В поле окна сообщения введите сообщение, "Никакая найденная антивирусная программа, установите антивирус и попробуйте еще раз". Если пользователь имеет антивирусный пакет и ниже дневного порога обновления, им не дают сообщение, как обозначено двойными кавычками в линии 7 из данного примера:

```
(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)())
```

Если у пользователя есть антивирус Norton, но последнее обновление больше, чем 15 дней, этот пример сообщения появляется:

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

Если EVAL не совпадает, это переходит к следующей функции, соответствиям, и возвращает значение истины. С тех пор нет никакого CheckAndMsg, привязанного к второй функции, она использует текстовое сообщение DAP:

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

Таким образом, DAP ищет пользовательский AAA и атрибут оконечной точки для соответствия с DAP. Если DAP совпадает, пользователь завершен с сообщением. Соответствие оконечной точки является результатом EVAL Lua, который возвращает истину или ложь к DAP. Истина соответствия и запрещает соединение. Ложь не совпадает и действительно разрешает соединение.

1. Первая функция в петле проверяет, равен ли endpoint.av.xxxxx.exists истине и если последнее обновление является меньше, чем настроенные дни. Пользователи без антивирусного программного обеспечения являются доступом разрешен, потому что пользовательский AAA совпадает, но Lua в частности ищет endpoint.av.xxxxx.exists = истинный и endpoint.av.xxxxx.lastupdate <= дни.
2. Вторая петля ловит пользователей без антивирусного программного обеспечения и блокирует их, потому что вторая функция только ищет endpoint.av.xxxxx.exists истинный NE. Если пользовательская оконечная точка av существует, не равно истине, функция возвращает значение истины, что означает, что у них нет антивируса. DAP совпадает и запрещает соединение.

Проверьте для установки программы обезвреживания шпионского ПО

Эта пользовательская функция проверяет, обнаруживает ли CSD программу обезвреживания шпионского ПО:

```
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)()
```

Проверьте для установки межсетевого экрана

Эта пользовательская функция проверяет, обнаруживает ли CSD межсетевой экран:

```
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)()
```

Проверьте для антивируса, программы обезвреживания шпионского ПО или установки межсетевого экрана

Если антивирус, программа обезвреживания шпионского ПО или пакет межсетевого экрана найдены, эта функция возвращает true:

```
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)()
```

Оконечный, если никакая установка шпионского ПО

Единственная разница между этой функцией и функцией под контролем для Установки Программы обезвреживания шпионского ПО то, что '`not' precedes` утверждение.

```
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)()
```

Проверьте для установки антивируса и межсетевого экрана и проверьте последнее обновление антивируса, не больше, чем тридцать дней

Данный пример возвращает true, если антивирус и межсетевой экран найдены и если последнее обновление антивируса не больше, чем 30 дней:

```
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)()
```

Поскольку межсетевой экран не имеет значения lastupdate для возврата, он имеет отдельную функцию.

Соответствие регулярного выражения

В этом разделе описываются функции, которые используют выражения regex, чтобы совпасть с определенными атрибутами и определить законность главного компьютера. Эти возможности regex были протестированы и допустимы:

- Знак доллара ($) привязывает строку поиска до конца возвращаемого значения.
- Вставка (^) привязывает строку поиска к началу возвращаемого значения.
- Символы на кронштейнах, такие как [Aa], совпадают со множественными символами в определенной позиции. Например, для соответствия с (нечувствительным к регистру) Ou=Cisco, используйте OU = [Cc][Ii][Ss][Cc][Oo].
- Период (.) совпадает с любым отдельным символом в этой позиции. Например, Группа.. Пользователи совпадают с Group01Users, Group33Users, и так далее.

Подключение, если ПК оконечной точки имеет какой-либо экземпляр исправления KB944

Эта функция использует соответствие регулярного выражения, чтобы видеть, содержит ли список заплаты образец. В данном примере Cisco Secure Desktop возвращает все заплаты на ПК оконечной точки; если существует экземпляр KB944, соответствий политики DAP и принужден.

```
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)()
```

Например, если главный компьютер имеет заплату KB944533 или заплата KB944653, это совпадает с правилом.

Используйте сценарий LUA для проверки OUI MAC-адреса

Если ПК Оконечной точки Имеет Какой-либо Экземпляр исправления KB944, эта функция подобна той, описанной в Подключении. Эта функция использует регулярное выражение для соответствия с организационно уникальным идентификатором (OUI) MAC-адреса.

В данном примере MAC-адрес запускается с d067.e5XX.XX. Используйте регулярное выражение и код Lua для соответствия с машинами, которые запускаются с того же MAC OUI.

```
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)()
```

Примечание: Другая версия этой функции требуется для многозначной проверки.

Подключение на основе первых трех букв от имени хоста (не чувствительный к регистру)

Эта функция использует регулярные выражения, чтобы определить, являются ли первые три буквы от имени хоста (нечувствительным к регистру) msv:

```
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)()
```

Подключение, если Device.id ПК оконечной точки и серийного номера на сертификате являются тем же

Если device.id ПК оконечной точки и серийного номера на сертификате являются тем же, это выражение Lua предназначается, чтобы соединиться:

```
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 ) ()
```

Примечание: Использование подстановочного знака **(*)** не работает в этой определенной функции (endpoint.certificate.user [" * "], не работает). Необходимо получить каждую пару KV индивидуально и синтаксический анализ через них.

Принудите DAP на основе просмотра хоста CSD для доменного ключа реестра

Эта процедура предоставляет примеру процедуры настройки с ASDM.

1. Найдите ключ реестра, который держит домен в \HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Domain.

2. Определите параметр просмотра хоста для настройки реестра.

3. Примените атрибут оконечной точки реестра к политике DAP.

4. Установите сеанс VPN Уровня защищенных сокетов (SSL).

5. Проверьте принудительную политику DAP через отладки DAP.

Поддержка DAP Windows 7 и CSD 3.5

Платформы Windows 7 поддерживаются с Выпуском 3.5 CSD или позже. С ASDM 6.2.x отладочный релиз и 6.3.x версии, можно непосредственно использовать интерфейс для проверки для Windows 7 OS. С более ранними версиями ASDM усовершенствованный DAP сценарий Lua требуется для проверки для машин Windows 7. На ASA с Выпуском 8.x и предбета Выпуском 3.5 CSD, войдите, эта строка сценария Lua в DAP ASDM усовершенствовала коробку для выполнения проверок для машин Windows 7:

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

Идентификация iPhone, iPad и Мобильных устройств

Это выражение Lua позволяет вам отследить определенные мобильные устройства их уникальными идентификаторами (UID). Можно использовать DAP для достижения этих базовых функций.

Когда значение не может быть жестко закодировано и должно быть считано из AD, это становится более трудным. Поскольку нет никакого определенного поля UID в AD, можно сохранить значение для индивидуального пользователя под другим полем. Данный пример использует otherHomePhone для хранения UID.

Чтобы помочь вам определять UID для iPhone или iPad, ищите сеть соответствующее программное средство.

Как только вы определяете UID, добавьте его к otherHomePhone в AD записи для того пользователя.

От **команды debug ldap 255** и от пользовательской тестовой аутентификации, получите выдвигаемый атрибут LDAP, который является otherHomePhone.

Позвольте телефону соединяться, затем выполнять трассировку DAP во время предпринятого соединения для определения атрибута оконечной точки, который содержит UID (endpoint.anyconnect.deviceuniqueid).

Это выражение Lua может тогда сравнить эти два параметра:

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

Использование DAP для предотвращения соединения определенным браузером

Эта процедура описывает, как использовать DAP для предотвращения соединения браузером Chrome:

1. Включите CSD.

2. Под конфигурацией просмотра хоста используйте ID Процесса (PID) и имя процесса для добавления просмотра процесса.

   Можно определить PID и имя процесса на Windows с Менеджером задач. Для отображения PID значения откройте **Менеджера задач**, перейдите к вкладке **Processes**, нажмите **меню View**, затем нажмите **Select Columns**. В Выбрать Columns или диалоговом окне Select Process Page Columns, отметьте и проверьте флажок для **PID**

(Идентификатор процесса) и нажмите OK.

На Macs можно определить ID Процесса с монитором активности. Или, в оболочке удара (который банка можно также использовать в Unix), используйте ps-e команда, в то время как процесс работает, затем совпадите с PIDs к имени процесса с кошкой / процедура / <pid>/cmdline команда.

3. Создайте политику DAP для тестирования, среди прочего, если тот процесс работает на машине.

4. Протестируйте свое соединение.

Предупреждения

1. Проблема с этим решением состоит в том, что у пользователя не может быть Chrome, открытого на его машине вообще. DAP просто проверяет, работает ли тот определенный процесс Chrome, но не проверяет, чтобы видеть, инициировался ли безклиентый сеанс тем процессом или некоторым другим процессом. Когда подключение WebVPN предпринято, таким образом, пользователь не может выполнить Chrome в фоновом режиме.
2. Примите сценарий, где пользователь использует Firefox для открытия сеанса WebVPN и сбоев попытки входа. Пользователь помнит, что Chrome все еще работает, таким образом, пользователь завершает соединение Chrome и попытки войти снова. Вход в систему все еще отказывает, потому что CSD должен повторно выполнить просмотр хоста. Так, пользователь должен также закрыть экземпляр Firefox, который использовался, чтобы обратиться к WebVPN, затем перезапустить Firefox. Этот процесс может сбить с толку пользователю. Cisco рекомендует создать сообщение об ошибках DAP, которое говорит пользователю завершать Chrome и закрывать браузер, который они в настоящее время используют:

Часто задаваемые вопросы

Этот раздел предоставляет ответ на один из большинства часто задаваемых вопросов в отношениях информация, которая описана в этом документе.

Почему мой сценарий LUA работает для некоторых пользователей, но не для всех?

Рассмотрите этот сценарий 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)()
```

Этот сценарий разработан для соответствия с хранилищем сертификата и подчиненным DC, который найден в сертификате. Однако этот сценарий был протестирован со множественными машинами и, как находили, работал на некоторые машины, но сбой на многих других.

Причина, что этот сценарий только работает периодически, из-за способа, которым hostscan возвращает значения. Когда вы просматриваете трассировку DAP, которая не работает, вы видите, что *subject_dc* возвращает множественные значения на сертификат. Можно также видеть, что последнее значение возвратилось, не *homedepot*:

```
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"
```

Когда вы просматриваете трассировку DAP, которая действительно работает, это может наблюдаться:

```
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"
```

Это указывает, что сценарий LUA работает должным образом. Однако из-за способа, которым оценка положения возвращает значения на некоторых машинах, сценарий не совпадает. В этом случае исправления для идентификаторов ошибок Cisco CSCuu85646 и CSCuh67472 требуются.