在CPAR 8.0上配置自定義指令碼

目錄

簡介

本檔案介紹如何使用指令碼和擴展點自訂Cisco Prime Access Registrar(CPAR)8.0行為。

必要條件

需求

思科建議您瞭解以下主題：

- CPAR 8.0管理

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- CPAR 8.0安裝在CentOS 6.5 64位上

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

CPAR可通過內部和外部指令碼進行修改。指令碼可以用C/C++/Java/TCL編寫。指令碼可用於修改RADIUS、TACACS和DIAMETER資料包的處理。指令碼可以在CPAR的擴展點中引用。擴展點是出現在某些配置元素下並允許引用指令碼的設定/屬性。根據參[考指南](#),CPAR不對自定義指令碼引起的任何資料丟失、損壞等負責。

以下是網路裝置配置下的兩個擴展點的示例

```
[ //localhost/Radius/Clients/piborowi ]
    Name = piborowi
```

```
Description =
Protocol = tacacs-and-radius
IPAddress = 192.168.255.15
SharedSecret = <encrypted>
Type = NAS
Vendor =
IncomingScript~ =                          // Extension point for incomming traffic
OutgoingScript~ =                          // Extension point for outgoing traffic
EnableDynamicAuthorization = FALSE
NetMask =
EnableNotifications = FALSE
EnforceTrafficThrottling = TRUE
```

根據CPAR管理指南，有多個可用的擴展點。可以在以下每個擴展點引用傳入指令碼：

- RADIUS伺服器
- 供應商（直接客戶的）
- 客戶端（單個NAS）
- NAS-Vendor-Behind-the-Proxy
- Client-Behind-the-Proxy
- 遠端伺服器（RADIUS型別）
- 服務

可以在以下每個擴展點引用身份驗證或授權指令碼：

- 群組驗證
- 使用者驗證
- 組授權
- 使用者授權

可以在以下每個擴展點引用傳出指令碼：

- 服務
- Client-Behind-the-Proxy
- NAS-Vendor-Behind-the-Proxy
- 客戶端（單個NAS）
- NAS供應商
- RADIUS伺服器

由於存在多個擴展點，因此瞭解CPAR執行指令碼的順序至關重要。請參閱管理員指南的表7-1，檢視29個可用指令碼/擴展點的順序。

內部指令碼是直接在CPAR CLI(aregcmd)中配置的指令碼。 它不需要任何外部檔案和很多程式設計知識。外部指令碼是儲存在作業系統（CENTOS或RHEL）中的檔案中，並且僅在CPAR CLI中引用。

設定

傳出流量的內部指令碼

在內部指令碼中，可以使用以下修飾符：

1.+rsp: — 為響應新增和屬性

2.-rsp: — 從響應中刪除屬性

3.#rsp: — 用新值替換屬性

4.以上內容可用於請求（請求/輸入資料包和env，即環境字典）。 示例+req:或 — env:

在/Radius/Scripts下新增內部指令碼。配置另外兩個AVP以與訪問接受資料包一起返回：Filter-Id和Vendor-Specific one（用於加入語音域）。

```
--> ls -R

[ //localhost/Radius/Scripts/addattr ]
    Name = addattr
    Description =
    Language = internal
    Statements/
        1. +rsp:Filter-Id=PhoneACL
        2. +rsp:Cisco-AVPair=device-traffic-class=voice

--> ls -R

[ Services/local-users ]
    Name = local-users
    Description =
    Type = local
    IncomingScript~ =
    OutgoingScript~ = addattr
    OutagePolicy~ = RejectAll
    OutageScript~ =
    UserList = Default
    EnableDeviceAccess = True
    DefaultDeviceAccessAction~ = DenyAll
    DeviceAccessRules/
        1. switches
```

使用本地radclient進行測試：

```
--> simple
```

```
p011
--> p011 send
p014
--> p014
Packet: code = Access-Accept, id = 18, length = 64, attributes =
        Filter-Id = PhoneACL
        Cisco-AVPair = device-traffic-class=voice
```

跟蹤：

```
07/31/2019 10:31:26.254: P2363: Running Service local-users's OutgoingScript: addattr
07/31/2019 10:31:26.254: P2363: Internal Script for 1   +rsp:Filter-Id=PhoneACL : Filter-Id =
PhoneACL
07/31/2019 10:31:26.254: P2363: Setting value PhoneACL for attribute Filter-Id
```

```
07/31/2019 10:31:26.254: P2363: Trace of Response Dictionary
07/31/2019 10:31:26.254: P2363: Trace of Access-Request packet
07/31/2019 10:31:26.254: P2363:    identifier = 18
07/31/2019 10:31:26.254: P2363:    length = 30
07/31/2019 10:31:26.254: P2363:    respauth = fb:63:14:3f:c1:fb:ac:03:7d:16:29:61:ba:ef:13:4f
07/31/2019 10:31:26.254: P2363:    Filter-Id = PhoneACL
07/31/2019 10:31:26.254: P2363: Internal Script for 2   +rsp:Cisco-AVPair=device-traffic-
class=voice : Cisco-AVPair = device-traffic-class=voice
07/31/2019 10:31:26.254: P2363: Setting value device-traffic-class=voice for attribute Cisco-
AVPair
07/31/2019 10:31:26.254: P2363: Trace of Response Dictionary
07/31/2019 10:31:26.254: P2363: Trace of Access-Request packet
07/31/2019 10:31:26.254: P2363:    identifier = 18
07/31/2019 10:31:26.254: P2363:    length = 64
07/31/2019 10:31:26.254: P2363:    respauth = fb:63:14:3f:c1:fb:ac:03:7d:16:29:61:ba:ef:13:4f
07/31/2019 10:31:26.254: P2363:    Filter-Id = PhoneACL
07/31/2019 10:31:26.254: P2363:    Cisco-AVPair = device-traffic-class=voice
```

傳入流量的內部指令碼

建立一個新指令碼，將格式user@domain的所有使用者名稱替換為anonymous，並將其作為您使用的服務的傳入指令碼應用。

設定:

```
--> cd /Radius/Scripts

--> add test

--> set language internal

--> cd Statements

--> add 1

--> cd 1

--> set statements "#req:User-Name=~(.*)(@[a-z]+.[a-z]+)~\anonymous"

--> ls -R

[ //localhost/Radius/Scripts/test ]
    Name = test
    Description =
    Language = internal
    Statements/
        1. #env:User-Name=~(.*)~anonymous

--> ls -R /Radius/Services/employee-service/

[ /Radius/Services/employee-service ]
    Name = employee-service
    Description =
    Type = local
    IncomingScript~ = test
    OutgoingScript~ =
    OutagePolicy~ = RejectAll
    OutageScript~ =
    UserList = default
    EnableDeviceAccess = FALSE
```

```
    DefaultDeviceAccessAction~ = DenyAll
```

使用radclient進行測試（請求很可能被拒絕，因為使用者名稱已更改為匿名）：

```
--> simple




p01e

--> p01e
Packet: code = Access-Request, id = 27, length = 72, attributes =
User-Name = <username>@cisco.com
User-Password = <password>
NAS-Identifier = localhost
NAS-Port = 7
--> p01e send
p020
--> p020
Packet: code = Access-Reject, id = 27, length = 35, attributes =
        Reply-Message = Access Denied
```

跟蹤：

在執行員工服務之前，將呼叫三個指令碼。首先，CPAR呼叫*CiscoIncomingScript*，然後呼叫附加到localhost Client/Network Device配置的*ParseServiceHints*。它從資料包中提取使用者名稱並將其放入環境字典中。第二個指令碼：呼叫*test*，並將環境字典中的使用者名稱從<username>更改為anonymous

localhost客戶端：

```
[ //localhost/Radius/Clients/localhost ]
    Name = localhost
    Description =
    Protocol = radius
    IPAddress = 127.0.0.1
    SharedSecret = <encrypted>
    Type = NAS+Proxy
    Vendor = Cisco
    IncomingScript~ = ParseServiceHints
    OutgoingScript~ =
    EnableDynamicAuthorization = FALSE
    NetMask =
    EnableNotifications = FALSE
    EnforceTrafficThrottling = TRUE
```

跟蹤輸出：

```
07/31/2019 11:38:53.522: P2855: PolicyEngine: [SelectPolicy] Successful
07/31/2019 11:38:53.522: P2855: Using Client: localhost
07/31/2019 11:38:53.522: P2855: Using Vendor: Cisco
07/31/2019 11:38:53.522: P2855: Running Vendor Cisco's IncomingScript: CiscoIncomingScript
07/31/2019 11:38:53.522: P2855: Running Client localhost IncomingScript: ParseServiceHints
07/31/2019 11:38:53.522: P2855: Rex: environ->get( "Request-Type" ) -> "Access-Request"
```

```
07/31/2019 11:38:53.522: P2855: Rex: environ->get( "Request-Type" ) -> "Access-Request"
07/31/2019 11:38:53.522: P2855: Rex: environ->get( "User-Name" ) -> "<username>"


07/31/2019 11:38:53.522: P2855: Authenticating and Authorizing with Service employee-service
07/31/2019 11:38:53.522: P2855: Running Service employee-service's IncomingScript: test
07/31/2019 11:38:53.522: P2855: Numbered attribute got for the radius / tacacs packet. ignoring
# User-Name
07/31/2019 11:38:53.523: P2855: Numbered attribute got for the radius / tacacs packet. ignoring
# User-Name
07/31/2019 11:38:53.523: P2855: Numbered attribute got for the radius / tacacs packet. ignoring
# User-Name
07/31/2019 11:38:53.523: P2855: Internal Script for 1   #env:User-Name=~(.*)~anonymous : User-
Name = anonymous
07/31/2019 11:38:53.523: P2855: Setting value anonymous for attribute User-Name
07/31/2019 11:38:53.523: P2855: Trace of Environment Dictionary
07/31/2019 11:38:53.523: P2855:            User-Name = anonymous
07/31/2019 11:38:53.523: P2855:            NAS-Name-And-IPAddress = localhost (127.0.0.1)
07/31/2019 11:38:53.523: P2855:            Authorization-Service = employee-service
07/31/2019 11:38:53.523: P2855:            Source-Port = 51169
07/31/2019 11:38:53.523: P2855:            Authentication-Service = employee-service
07/31/2019 11:38:53.523: P2855:            Trace-Level = 1000
07/31/2019 11:38:53.523: P2855:            Destination-Port = 1812
07/31/2019 11:38:53.523: P2855:            Destination-IP-Address = 127.0.0.1
07/31/2019 11:38:53.523: P2855:            Source-IP-Address = 127.0.0.1
07/31/2019 11:38:53.523: P2855:            Enforce-Traffic-Throttling = TRUE
07/31/2019 11:38:53.523: P2855:            Request-Type = Access-Request
07/31/2019 11:38:53.523: P2855:            Script-Level = 6
07/31/2019 11:38:53.523: P2855:            Provider-Identifier = Default
07/31/2019 11:38:53.523: P2855:            Request-Authenticator =
5f:62:5a:72:0f:7b:a2:2a:9c:06:ba:2e:bd:f4:e4:4b
07/31/2019 11:38:53.523: P2855:            Realm = cisco.com
07/31/2019 11:38:53.523: P2855: Getting User anonymous's UserRecord from UserList Default
07/31/2019 11:38:53.523: P2855: Failed to get User anonymous's UserRecord from UserList Default
07/31/2019 11:38:53.523: P2855: Running Vendor Cisco's OutgoingScript: CiscoOutgoingScript
07/31/2019 11:38:53.523: P2855: Trace of Access-Reject packet
07/31/2019 11:38:53.523: P2855:    identifier = 27
07/31/2019 11:38:53.523: P2855:    length = 35
07/31/2019 11:38:53.523: P2855:    respauth = d3:7d:b3:f6:05:47:2c:66:d9:c0:01:7d:67:d7:93:99
07/31/2019 11:38:53.523: P2855:    Reply-Message = Access Denied
07/31/2019 11:38:53.523: P2855: Sending response to 127.0.0.1
```

建立外部指令碼

將檔案*nadp.tcl*新增到/opt/CSCOar/scripts/radius/tcl/目錄並新增以下內容：

```
[root@piborowi-cpar80-16 tcl]# cat /opt/CSCOar/scripts/radius/tcl/nadip.tcl
proc UpdateNASIP {request response environ} {
$request trace 2 "TCL CUSTOM_SCRIPT Updating NAS IP ADDRESS"
$request trace 2 "Before put: " [ $request get NAS-IP-Address ]
$request put NAS-IP-Address 1.2.3.4
$request trace 2 "After put: " [ $request get NAS-IP-Address ]
}
```

nadip.tcl*的內容按*行說明如下：

行#1過程定義和引數。請求、響應、環境和三個可用詞典，您可以在其中修改會話/資料包資料。

行#2要作為跟蹤級別2列印的指令碼的調試行。

在設#3此值之前，請在請求字典中輸入NAS-IP-Address屬性的內容。

行#4將請求字典中的Nas-IP-Address屬性設定為值1.2.3.4。

行#5再次列印NAS-IP-Address屬性。

在作業系統中建立並儲存指令碼後，配置對指令碼的CPAR引用。將language設定為TCL，filename必須是帶副檔名的確切檔名（本例中為nadip.tcl）。 EntryPoint是檔案中要作為指令碼執行的過程的名稱。引用在服務(incomingScript)下建立CPAR指令碼並使用radclient進行測試。

在追#2中可#3到#5線100、100和1000:

```
--> ls -R /Radius/scripts/nadipaddress/

[ /Radius/Scripts/nadipaddress ]
    Name = nadipaddress
    Description =
    Language = tcl             <<<<<<<<
    Filename = nadip.tcl       <<<<<<<<
    EntryPoint = UpdateNASIP   <<<<<<<<
    InitEntryPoint =
    InitEntryPointArgs =

--> ls -R /Radius/services/employee-service/

[ /Radius/Services/employee-service ]
    Name = employee-service
    Description =
    Type = local
    IncomingScript~ = nadipaddress    <<<<<<<<
    OutgoingScript~ =
    OutagePolicy~ = RejectAll
    OutageScript~ =
    UserList = default
    EnableDeviceAccess = FALSE
    DefaultDeviceAccessAction~ = DenyAll
```

跟蹤：

```
07/31/2019 13:40:53.615: P3490: Running Service employee-service's IncomingScript: nadipaddress
07/31/2019 13:40:53.615: P3490: TCL CUSTOM_SCRIPT Updating NAS IP ADDRESS
07/31/2019 13:40:53.616: P3490:     Tcl: request trace 2 TCL CUSTOM_SCRIPT Updating NAS IP
ADDRESS -> OK
07/31/2019 13:40:53.616: P3490:     Tcl: request get NAS-IP-Address -> <empty>
07/31/2019 13:40:53.616: P3490: Before put:
07/31/2019 13:40:53.616: P3490:     Tcl: request trace 2 Before put:   -> OK
07/31/2019 13:40:53.616: P3490:     Tcl: request put NAS-IP-Address 1.2.3.4 -> OK
07/31/2019 13:40:53.616: P3490:     Tcl: request get NAS-IP-Address -> 1.2.3.4
07/31/2019 13:40:53.616: P3490: After put: 1.2.3.4
07/31/2019 13:40:53.616: P3490:     Tcl: request trace 2 After put:  1.2.3.4 -> OK
```