Configurazione di script personalizzati in CPAR 8.0

Sommario

Introduzione

In questo documento viene descritto come personalizzare il comportamento di Cisco Prime Access Registrar (CPAR) 8.0 con l'utilizzo di script e punti di estensione.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Amministrazione CPAR 8.0

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- CPAR 8.0 installato su CentOS 6.5 64 bit

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

CPAR può essere modificato da script interni ed esterni. Gli script possono essere scritti in C/C++/Java/TCL. Gli script possono essere utilizzati per modificare l'elaborazione dei pacchetti RADIUS, TACACS e DIAMETER. È possibile fare riferimento agli script in CPAR nei punti di

estensione. I punti di estensione sono un'impostazione/un attributo visualizzato sotto alcuni elementi di configurazione e che consente di fare riferimento a uno script. Come da guida di riferimento CPAR non è responsabile per qualsiasi perdita di dati, danni, ecc. causati da script personalizzati.

Di seguito è riportato un esempio di due punti di estensione nella configurazione dei dispositivi di rete

```
[ //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
```

Secondo la guida all'amministrazione CPAR, sono disponibili diversi punti di estensione. È possibile fare riferimento a uno script in ingresso in ognuno dei seguenti punti di estensione:

- server RADIUS
- Fornitore (del client immediato)
- Client (NAS individuale)
- NAS-Vendor-Behind-the-Proxy
- Client-Dietro-il-Proxy
- Server remoto (di tipo RADIUS)
- Servizio

Èpossibile fare riferimento a uno script di autenticazione o autorizzazione in ognuno dei seguenti punti di estensione:

- Autenticazione gruppo
- Autenticazione utente
- Autorizzazione gruppo
- Autorizzazione utente

Èpossibile fare riferimento allo script in uscita in ognuno dei seguenti punti di estensione:

- Servizio
- Client-Dietro-il-Proxy
- NAS-Vendor-Behind-the-Proxy
- Client (NAS individuale)
- Fornitore NAS
- server RADIUS

Èfondamentale comprendere l'ordine in cui gli script vengono eseguiti da CPAR, poiché esistono più punti di estensione. Consultare la tabella 7-1 della guida per l'amministratore per verificare l'ordine di 29 punti di script/estensione disponibili.

Uno script interno è uno script configurato direttamente in CLI di CPAR (aregcmd). Non richiede alcun file esterno e molta conoscenza della programmazione. Uno script esterno è uno script memorizzato in un file nel sistema operativo (CENTOS o RHEL) e referenziato in CLI CPAR.

Configurazione

Script Interno Per Traffico In Uscita

Negli script interni è possibile utilizzare i seguenti modificatori:

1. +rsp: - aggiunge e assegna alla risposta

2. -rsp: - rimuove l'attributo dalla risposta

3. #rsp: - sostituisce l'attributo con il nuovo valore

4. può essere utilizzato per req (request/incomming packet and env, che è un dizionario di ambiente). Esempi +richiesta: o -env:

Aggiungere uno script interno in /Radius/Scripts. Configurare due AVP aggiuntivi da restituire con il pacchetto Access-Accept: Filter-Id e uno specifico del fornitore (per l'aggiunta al dominio vocale).

```
--> 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
```

Test con l'utilizzo di radclient locale:

```
--> simple
```

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

Tracce:

```
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
```

Script Interno Per Traffico In Entrata

Creare un nuovo script che sostituisca tutti i nomi utente nel formato user@domain con anonimo e applicarlo come script in ingresso per il servizio utilizzato.

Configurazione:

```
--> 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
```

Test con radclient (la richiesta è probabilmente rifiutata perché il nome utente è stato modificato in anonimo):

```
--> 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
```

Traccia:

Prima dell'esecuzione del servizio per i dipendenti, vengono richiamati tre script. Prima CPAR richiama *CiscoIncomingScript*, quindi *ParseServiceHints* collegato alla configurazione del client/dispositivo di rete localhost. Estrae il nome utente dal pacchetto e lo inserisce nel dizionario dell'ambiente. Secondo script, viene richiamato il *test* e il nome utente nel dizionario dell'ambiente viene modificato da <nomeutente> a anonimo

client localhost:

```
[ //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
```

Output di traccia:

```
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
```

Crea script esterno

Aggiungere un file *nadip.tcl* alla directory /opt/CSCOar/scripts/radius/tcl/ e aggiungere il seguente contenuto:

```
[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 ]
}
```

Contenuto di *nadip.tcl* spiegato riga per riga:

Riga n. 1 Definizione e argomenti della routine. Richiesta, risposta, environ e tre dizionari disponibili in cui è possibile modificare i dati di sessioni/pacchetti.

Riga n. 2 Riga di debug per lo script da stampare come livello di traccia 2.

Riga n. 3 Contenuto dell'attributo NAS-IP-Address nel dizionario delle richieste prima di impostare questo valore.

Riga n. 4 Impostare l'attributo Nas-IP-Address nel dizionario della richiesta sul valore 1.2.3.4.

Riga 5: stampare nuovamente l'attributo indirizzo IP-NAS.

Una volta creato e salvato lo script nel sistema operativo, configurare il riferimento CPAR allo script. Imposta la lingua come TCL, il nome file deve essere esattamente il nome file con estensione (in questo caso è nadip.tcl). EntryPoint è il nome della routine nel file che si desidera eseguire come script. Riferimento creato script CPAR in servizio (incomingScript) e test con radclient.

Le righe #2, #3, #5 possono essere osservate nella traccia:

```
--> 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
```

Traccia:

```
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
```