Configurar IBNS 2.0 para encenações do host único e do Multi-domínio

Índice

Introdução

Este documento descreve como configurar os serviços de rede baseados identidade 2.0 (IBNS) para encenações do host único e do multi-domínio.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Protocolo extensible authentication sobre a rede de área local (EAPoL)
- Protocolo de raio
- Versão 2.0 do Cisco Identity Services Engine

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Correção de programa 2 da versão 2.0 do motor do serviço da identidade de Cisco
- Valor-limite com OS de Windows 7
- Switch Cisco 3750X com IO 15.2(4)E1
- Switch Cisco 3850 com 03.02.03.SE
- Cisco IP Phone 9971

A informação neste documento é criada dos 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.

Configurar

Teoria da configuração

A fim permitir IBNS 2.0, você precisa de executar o comando no modo do privilégio em seu switch Cisco:

```
#authentication display new-style
```

Configurar o switchport para IBNS 2.0 com comandos como mostrado:

```
 access-session host-mode {single-host | multi-domain | multi-auth}
 access-session port-control auto
 dot1x pae authenticator
{mab} service-policy type control subscriber TEST
```

Desvio da autenticação e opcionalmente da autenticação de MAC do dot1x destes comandos enable (MAB) na relação. Quando você segue a sintaxe nova, você usa comandos que começa com acesso-sessão. A finalidade daqueles comandos é mesma que para os comandos que usam a sintaxe antiga (que começa com palavra-chave da **autenticação**). Aplique a serviço-**política** para especificar o **mapa de política** que deve ser usado para a relação.

O mapa de política mencionado acima define o comportamento do interruptor (autenticador) durante a autenticação. Por exemplo, você pode especificar o que deve acontecer em caso da falha de autenticação. Para cada **evento** você pode configurar as ações múltiplas baseadas no tipo do evento combinado no **mapa de classe** configurado sob ele. Como um exemplo, olhe a lista como mostrado (**mapa de política TEST4**). Se o valor-limite do dot1x que está conectado à relação onde esta política é aplicada falha, a seguir a ação definida em **DOT1X_FAILED** está executada. Se você gostaria de especificar o mesmo comportamento para classes como **MAB_FAILED** e **DOT1X_FAILED**, a seguir você pode usar a classe padrão - **mapa de classe sempre.**

```
policy-map type control subscriber TEST4
(...)
 event authentication-failure match-first
  10 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
(...)
  40 class always do-until-failure
   10 terminate mab
   20 terminate dot1x
   30 authentication-restart 60
(...)
```

O mapa de política usado para IBNS 2.0 sempre deve ter o tipo **subscritor do controle**.

Você pode ver a lista de eventos disponíveis desta maneira:

```
Switch(config-event-control-policymap)#event ?
  aaa-available                aaa-available event
  absolute-timeout             absolute timeout event
  agent-found                  agent found event
  authentication-failure       authentication failure event
```

```
  authentication-success         authentication success event
  authorization-failure          authorization failure event
  inactivity-timeout             inactivity timeout event
  session-started                session started event
  tag-added                      tag to apply event
  tag-removed                    tag to remove event
  template-activated             template activated event
  template-activation-failed     template activation failed event
  template-deactivated           template deactivated event
  template-deactivation-failed   template deactivation failed event
  timer-expiry                   timer-expiry event
  violation                      session violation event
```

Na configuração de evento você tem a possibilidade para definir como as classes devem ser avaliadas:

```
Switch(config-event-control-policymap)#event authentication-failure ?
  match-all    Evaluate all the classes
  match-first  Evaluate the first class
```

Você pode definir a opção similar para **mapas de classe**, embora aqui você especifique como as ações devem ser executadas caso que sua classe é combinada:

```
Switch(config-class-control-policymap)#10 class always ?
  do-all            Execute all the actions
  do-until-failure  Execute actions until one of them fails
  do-until-success  Execute actions until one of them is successful
```

A última parte (opcional) da configuração no estilo novo do dot1x é **mapa de classe**. Igualmente deve datilografar o **subscritor do controle** e é usada para combinar o comportamento ou o tráfego específico. Configurar exigências para a avaliação da condição do **mapa de classe**. Você pode especificar que todas as circunstâncias têm que ser combinadas ou toda a circunstância tem que ser combinada ou nenhumas das circunstâncias devem combinar.

```
Switch(config)#class-map type control subscriber ?
  match-all   TRUE if everything matches in the class-map
  match-any   TRUE if anything matches in the class-map
  match-none  TRUE if nothing matches in the class-map
```

Este é exemplo do **mapa de classe** usado combinando a falha de autenticação do dot1x:

```
class-map type control subscriber match-all DOT1X_FAILED
 match method dot1x
 match result-type method dot1x authoritative
```

Para algumas encenações, na maior parte quando o serviço-molde está no uso, você precisa de adicionar a configuração para a mudança da autorização (CoA):

```
aaa server radius dynamic-author
 client 10.48.17.232 server-key cisco
```

Encenação para o host único

Diagrama de Rede

Configurações

Configuração básica do 802.1X exigida para a encenação do host único testada no catalizador 3750X com IO 15.2(4)E1. Encenação testada com o suplicante nativo de Windows e o Cisco AnyConnect.

```
aaa new-model
!
aaa group server radius tests
 server name RAD-1
!
aaa authentication dot1x default group tests
aaa authorization network default group tests
!
dot1x system-auth-control
!
policy-map type control subscriber TEST
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x priority 10
!
interface GigabitEthernet1/0/21
 switchport access vlan 613
 switchport mode access
 access-session host-mode single-host
 access-session port-control auto
 dot1x pae authenticator
 service-policy type control subscriber TEST
!
radius server RAD-1
 address ipv4 10.48.17.232 auth-port 1812 acct-port 1813
 key cisco
```

Encenação para o Multi-domínio

Diagrama de Rede

Configurações

a encenação do Multi-domínio foi testada no catalizador 3850 com os IO 03.02.03.SE devido às exigências PoE (potência sobre Ethernet) para o telefone IP (telefone IP 9971 de Cisoc).

```
aaa new-model
!
aaa group server radius tests
 server name RAD-1
!
aaa authentication dot1x default group tests
aaa authorization network default group tests
!
aaa server radius dynamic-author
 client 10.48.17.232 server-key cisco
!
```

```
dot1x system-auth-control
!
class-map type control subscriber match-all DOT1X
 match method dot1x
!
class-map type control subscriber match-all DOT1X_FAILED
 match method dot1x
 match result-type method dot1x authoritative
!
class-map type control subscriber match-all DOT1X_NO_RESP
 match method dot1x
 match result-type method dot1x agent-not-found
!
class-map type control subscriber match-all MAB
 match method mab
!
class-map type control subscriber match-all MAB_FAILED
 match method mab
 match result-type method mab authoritative
!
policy-map type control subscriber TEST4
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x priority 10
   20 authenticate using mab priority 20
 event authentication-failure match-first
  10 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
  20 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authenticate using dot1x priority 10
  30 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  40 class always do-until-failure
   10 terminate mab
   20 terminate dot1x
   30 authentication-restart 60
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x priority 10
 event authentication-success match-all
  10 class always do-until-failure
   10 activate service-template DEFAULT_LINKSEC_POLICY_SHOULD_SECURE
!
interface GigabitEthernet1/0/1
 switchport access vlan 613
 switchport mode access
 switchport voice vlan 612
 access-session host-mode multi-domain
 access-session port-control auto
 mab
 dot1x pae authenticator
 spanning-tree portfast
 service-policy type control subscriber TEST4
!
radius-server attribute 6 on-for-login-auth
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
radius-server vsa send cisco-nas-port
!
radius server RAD-1
 address ipv4 10.48.17.232 auth-port 1812 acct-port 1813
```

```
key cisco
```

Verificar

Use esta seção para confirmar se a sua configuração funciona corretamente.

Para efeitos de verificação, use estes comandam para alistar sessões de todos os switchports:

```
show access-session
```

Você pode igualmente ver a informação detalhada sobre sessões de um único switchport:

```
show access-session interface [Gi 1/0/1] {detail}
```

Troubleshooting

Esta seção fornece a informação que você pode se usar a fim pesquisar defeitos sua configuração.

A fim pesquisar defeitos problemas relacionados do 802.1X, você pode permitir debuga a mesma maneira que para a sintaxe do 802.1X do estilo antigo:

```
debug mab all
debug dot1x all
debug pre all*
```

* optionaly para debugar-lo pre pode usar somente o evento e/ou a regra para limitar a saída à informação relevante IBNS 2.0.