Konfigurieren von IBNS 2.0 für Single-Host- und Multi-Domain-Szenarien

Inhalt

Einführung
Voraussetzungen
Anforderungen
Verwendete Komponenten
Konfigurieren
Konfigurationstheorie
Szenario für einen einzelnen Host
Netzwerkdiagramm
Konfigurationen
Szenario für mehrere Domänen
Netzwerkdiagramm
Konfigurationen
Überprüfen
Fehlerbehebung

Einführung

In diesem Dokument wird beschrieben, wie Identity Based Networking Services 2.0 (IBNS) für Single-Host- und Multi-Domain-Szenarien konfiguriert wird.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- Extensible Authentication Protocol über Local Area Network (EAPoL)
- Radius-Protokoll
- Cisco Identity Services Engine Version 2.0

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Cisco Identity Service Engine Version 2.0 Patch 2
- Endgerät mit Windows 7-Betriebssystem
- Cisco Switch 3750X mit IOS 15.2(4)E1
- Cisco Switch 3850 mit 03.02.03.SE

- Cisco IP-Telefon 9971

Die Informationen in diesem Dokument werden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Konfigurieren

Konfigurationstheorie

Um IBNS 2.0 zu aktivieren, müssen Sie den Befehl im privilegierten Modus auf Ihrem Cisco Switch ausführen:

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

Konfigurieren Sie den Switch-Port für IBNS 2.0 mit folgenden Befehlen:

```
 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
```

Diese Befehle ermöglichen die 802.1x-Authentifizierung und optional MAB (MAC Authentication Bypass) auf der Schnittstelle. Wenn Sie die neue Syntax befolgen, verwenden Sie Befehle, die mit der Zugriffssitzung beginnen. Der Zweck dieser Befehle ist der gleiche wie für Befehle, die alte Syntax verwenden (beginnend mit dem **authentication-**Schlüsselwort). Wenden Sie **die Dienstrichtlinie** an, um **die Richtlinienzuordnung** anzugeben, die für die Schnittstelle verwendet werden soll.

Die oben erwähnte **Richtlinienzuordnung** definiert das Verhalten des Switches (Authentifizierer) während der Authentifizierung. Sie können z. B. angeben, was bei einem Authentifizierungsfehler geschehen soll. Für jedes **Ereignis** können Sie mehrere Aktionen anhand des Ereignistyps konfigurieren, der in der unter diesem Ereignis konfigurierten **Klassenzuordnung** zugeordnet ist. Schauen Sie sich beispielsweise die Liste an (**policy-map TEST4**). Wenn ein dot1x-Endpunkt, der mit der Schnittstelle verbunden ist, auf die diese Richtlinie angewendet wird, fehlschlägt, wird die in **DOT1X_FAILED** definierte Aktion ausgeführt. Wenn Sie dasselbe Verhalten für Klassen wie **MAB_FAILED** und **DOT1X_FAILED** angeben möchten, können Sie die Standardklasse **class-map immer** verwenden.

```
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
(...)
```

Die für IBNS 2.0 verwendete **Richtlinienzuordnung** muss immer über einen **Subscriber zur** Typsteuerung verfügen.

Sie können die Liste der verfügbaren Veranstaltungen folgendermaßen anzeigen:

```
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
```

In der Konfiguration können Sie definieren, wie Klassen evaluiert werden sollen:

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

Sie können eine ähnliche Option für **Klassenzuordnungen** definieren, wobei Sie jedoch hier angeben, wie Aktionen ausgeführt werden sollen, wenn die Klasse zugeordnet wird:

```
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
```

Der letzte (optionale) Teil der Konfiguration im neuen 802.1x-Format ist **class-map**.Sie sollte auch **Control Subscriber** eingeben und wird verwendet, um bestimmtes Verhalten oder bestimmten Datenverkehr zuzuordnen. Konfigurieren von Anforderungen für die Bedingungsauswertung **der Klassenzuordnung** Sie können angeben, dass alle Bedingungen zugeordnet werden müssen oder eine Bedingung zugeordnet werden muss, oder dass keine der Bedingungen übereinstimmen sollte.

```
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
```

Dies ist ein Beispiel für eine **Klassenzuordnung**, die für den Fehler bei der 802.1x-Authentifizierung verwendet wird:

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

In einigen Szenarien, vor allem wenn eine Service-Vorlage verwendet wird, müssen Sie die Konfiguration für die Änderung der Autorisierung (CoA) hinzufügen:

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

Szenario für einen einzelnen Host

Netzwerkdiagramm

Konfigurationen

Grundlegende 802.1X-Konfiguration erforderlich für Single-Host-Szenario, getestet auf Catalyst 3750X mit IOS 15.2(4)E1. Szenario getestet mit Windows Native Supplicant und 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
```

Szenario für mehrere Domänen

Netzwerkdiagramm

Konfigurationen

Das Multi-Domain-Szenario wurde auf Catalyst 3850 mit IOS 03.02.03.SE aufgrund von PoE (Power over Ethernet)-Anforderungen für IP-Telefone (Cisco IP Phone 9971) getestet.

```
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
```

Überprüfen

In diesem Abschnitt überprüfen Sie, ob Ihre Konfiguration ordnungsgemäß funktioniert.

Verwenden Sie diesen Befehl zur Überprüfung von Sitzungen aller Switch-Ports:

```
show access-session
```

Sie können auch detaillierte Informationen zu Sitzungen von einem einzigen Switch-Port aus anzeigen:

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

Fehlerbehebung

Dieser Abschnitt enthält Informationen, die Sie zur Fehlerbehebung bei Ihrer Konfiguration verwenden können.

Um Probleme im Zusammenhang mit 802.1X zu beheben, können Sie das Debuggen genauso aktivieren wie für die alte 802.1X-Syntax:

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

* Optional können Sie zum Debugging-Pre nur Ereignisse und/oder Regeln verwenden, um die Ausgabe auf relevante Informationen zu IBNS 2.0 zu beschränken.