RSA-Token-Server und SDI-Protokoll-Verwendung für ASA und ACS

Inhalt

Einführung

In diesem Dokument werden die Fehlerbehebungsverfahren für den RSA Authentication Manager beschrieben, der in die Cisco Adaptive Security Appliance (ASA) und den Cisco Secure Access Control Server (ACS) integriert werden kann.

Der RSA Authentication Manager ist eine Lösung, die das One Time Password (OTP) für die Authentifizierung bereitstellt. Dieses Kennwort wird alle 60 Sekunden geändert und kann nur einmal verwendet werden. Es unterstützt sowohl Hardware- als auch Software-Token.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über grundlegende Kenntnisse in folgenden Bereichen zu verfügen:

- Cisco ASA CLI-Konfiguration
- Cisco ACS-Konfiguration

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Softwareversionen:

- Cisco ASA Software, Version 8.4 und höher
- Cisco Secure ACS, Version 5.3 und höher

Die Informationen in diesem Dokument wurden 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.

Theorie

Der Zugriff auf den RSA-Server erfolgt über RADIUS oder das proprietäre RSA-Protokoll: SDI. Sowohl ASA als auch ACS können beide Protokolle (RADIUS, SDI) verwenden, um auf die RSA zuzugreifen.

Denken Sie daran, dass die RSA bei Verwendung eines Software-Tokens in den Cisco AnyConnect Secure Mobility Client integriert werden kann. Dieses Dokument konzentriert sich ausschließlich auf die Integration von ASA und ACS. Weitere Informationen zu AnyConnect finden Sie im Abschnitt Using SDI Authentication im Administratorhandbuch für den Cisco AnyConnect Secure Mobility Client, Version 3.1.

RSA über RADIUS

RADIUS hat einen großen Vorteil gegenüber SDI. Auf dem RSA können Benutzern spezifische Profile (Gruppen im ACS) zugewiesen werden. Für diese Profile sind spezifische RADIUS-Attribute definiert. Nach erfolgreicher Authentifizierung enthält die vom RSA zurückgegebene RADIUS-Accept-Nachricht diese Attribute. Auf der Grundlage dieser Attribute trifft der ACS zusätzliche Entscheidungen. Das häufigste Szenario ist die Entscheidung, die ACS-Gruppenzuordnung zu verwenden, um bestimmte RADIUS-Attribute, die sich auf das Profil auf der RSA beziehen, einer bestimmten Gruppe auf dem ACS zuzuordnen. Mit dieser Logik ist es möglich, den gesamten Autorisierungsprozess von der RSA in den ACS zu verschieben und dabei wie bei der RSA eine präzise Logik beizubehalten.

RSA über SDI

SDI bietet gegenüber RADIUS zwei Hauptvorteile. Die erste ist, dass die gesamte Sitzung verschlüsselt ist. Die zweite Option sind die interessanten Optionen, die der SDI-Agent bietet: kann ermittelt werden, ob der Fehler erstellt wurde, weil die Authentifizierung oder Autorisierung fehlgeschlagen ist oder der Benutzer nicht gefunden wurde.

Diese Informationen werden vom ACS in Aktion für die Identität verwendet. Beispielsweise könnte er für "user not found" (Benutzer nicht gefunden) fortfahren, aber für "authentication failed" (Authentifizierung fehlgeschlagen) ablehnen.

Zwischen RADIUS und SDI besteht ein weiterer Unterschied. Wenn ein Netzwerkzugriffsgerät wie ASA SDI verwendet, führt der ACS nur eine Authentifizierung durch. Wenn RADIUS verwendet wird, führt der ACS Authentifizierung, Autorisierung, Accounting (AAA) durch. Dies ist jedoch kein großer Unterschied. Es ist möglich, SDI für die Authentifizierung und RADIUS für die Abrechnung derselben Sitzungen zu konfigurieren.

SDI-Protokoll

SDI verwendet standardmäßig User Datagram Protocol (UDP) 5500. SDI verwendet zur Verschlüsselung von Sitzungen einen symmetrischen Verschlüsselungsschlüssel, ähnlich dem RADIUS-Schlüssel. Dieser Schlüssel wird in einer geheimen Knotendatei gespeichert und ist für jeden SDI-Client unterschiedlich. Diese Datei wird manuell oder automatisch bereitgestellt.

> **Hinweis**: ACS/ASA unterstützt keine manuelle Bereitstellung.

Für den automatischen Bereitstellungsknoten wird die geheime Datei nach der ersten erfolgreichen Authentifizierung automatisch heruntergeladen. Der Knotengeheim wird mit einem Schlüssel verschlüsselt, der aus dem Passcode des Benutzers und anderen Informationen abgeleitet wird. Dies führt zu einigen möglichen Sicherheitsproblemen. Daher sollte die erste Authentifizierung lokal erfolgen und das verschlüsselte Protokoll (Secure Shell [SSH], nicht Telnet) verwenden, um sicherzustellen, dass der Angreifer diese Datei nicht abfangen und entschlüsseln kann.

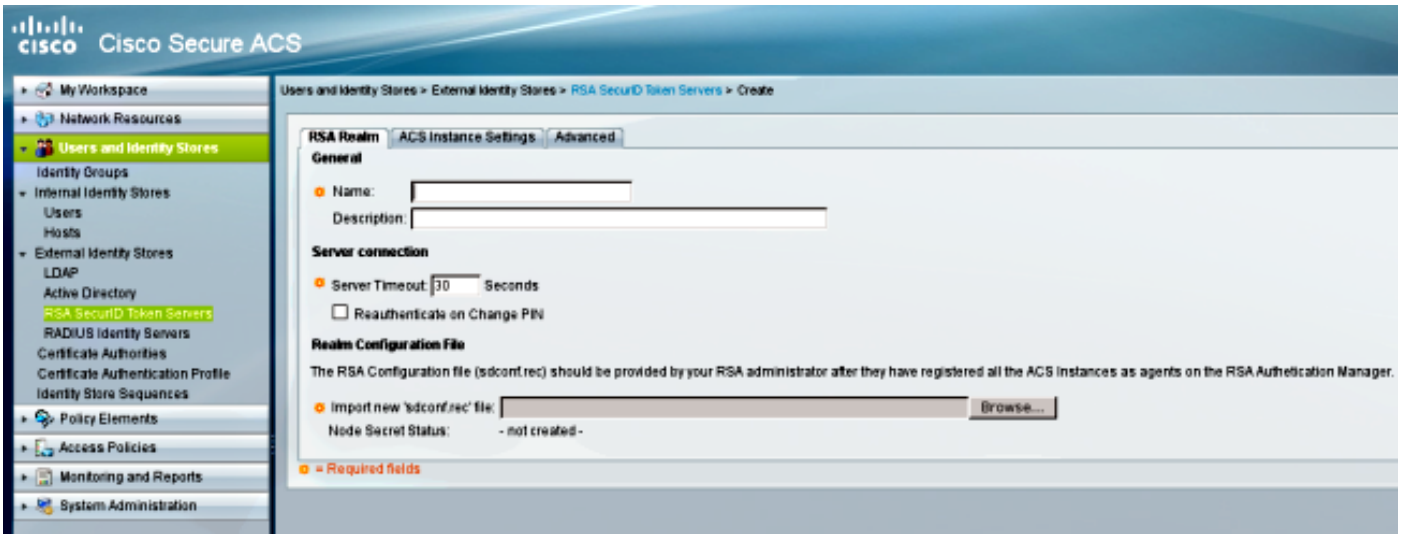
Konfiguration

> **Hinweise**:
>
> Verwenden Sie das Command Lookup Tool (nur registrierte Kunden), um weitere Informationen zu den in diesem Abschnitt verwendeten Befehlen zu erhalten.
>
> Das Output Interpreter Tool (nur registrierte Kunden) unterstützt bestimmte **show**-Befehle. Verwenden Sie das Output Interpreter Tool, um eine Analyse der **Ausgabe des** Befehls **show** anzuzeigen.
>
> Weitere Informationen zu Debug-Befehlen vor der Verwendung von **Debug**-Befehlen finden Sie unter Wichtige Informationen.

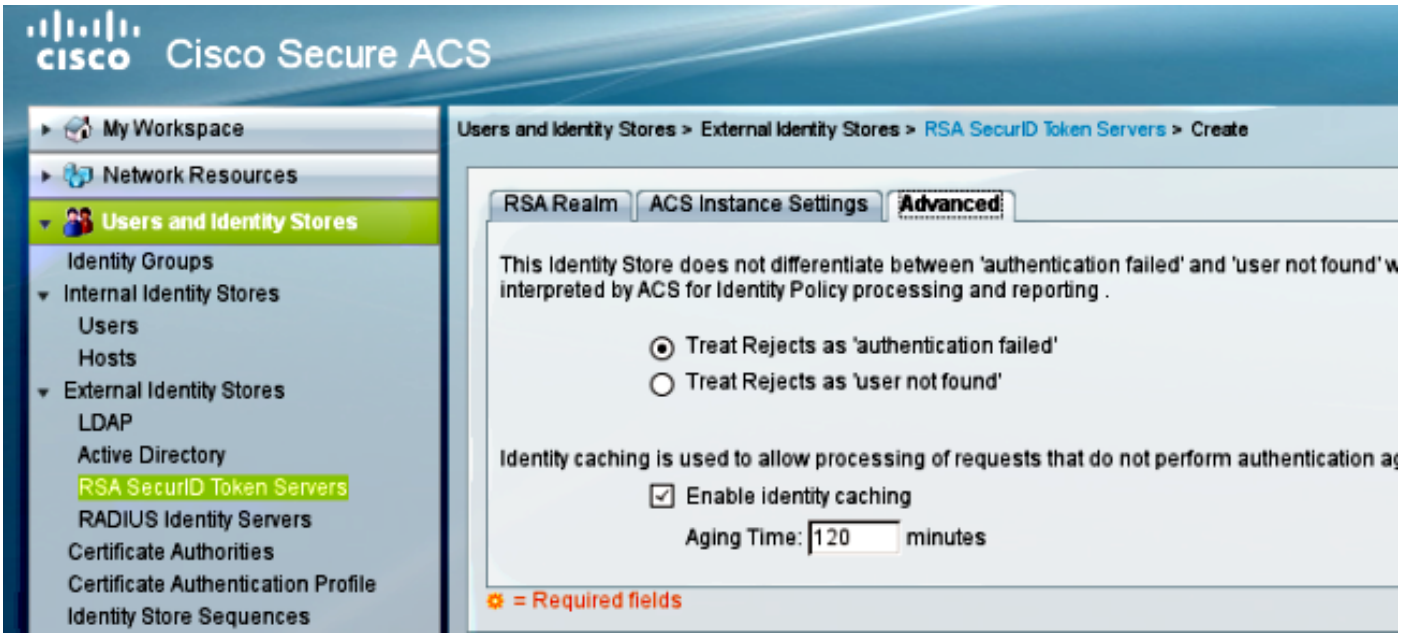
SDI auf ACS

Sie wird in **Benutzer- und Identitätsspeichern > Externer Identitätsspeicher > Sichere RSA-ID-Token-Server** konfiguriert.

Das RSA verfügt über mehrere Replikationsserver, z. B. die sekundären Server für den ACS. Es ist nicht erforderlich, alle Adressen dort anzugeben, nur die **sdconf.rec**-Datei, die vom RSA-Administrator bereitgestellt wird. Diese Datei enthält die IP-Adresse des primären RSA-Servers. Nach dem ersten erfolgreichen Authentifizierungsknoten wird die geheime Datei zusammen mit den IP-Adressen aller RSA-Replikate heruntergeladen.



Um "Benutzer nicht gefunden" von "Authentifizierungsfehler" zu unterscheiden, wählen Sie die Einstellungen auf der Registerkarte **Erweitert**:



Es ist auch möglich, die Standard-Routing-Mechanismen (Load Balancing) zwischen mehreren RSA-Servern (primär und replika) zu ändern. Ändern Sie sie mit der vom RSA-Administrator bereitgestellten Datei **sdopts.rec**. Im ACS wird es in **Benutzer- und Identitätsdaten** hochgeladen **> Externer Identitätsspeicher > Sichere RSA-ID-Token-Server > ACS-Instanzeinstellungen**.

Für die Cluster-Bereitstellung sollte die Konfiguration repliziert werden. Nach der ersten erfolgreichen Authentifizierung verwendet jeder ACS-Knoten seinen eigenen, vom primären RSA-Server heruntergeladenen Knoten-geheim. Es ist wichtig, sich zu erinnern, dass das RSA für alle ACS-Knoten im Cluster konfiguriert wird.

SDI auf ASA

Die ASA lässt das Hochladen der Datei **sdconf.rec** nicht zu. Ebenso wie der ACS ist auch hier nur die automatische Bereitstellung möglich. Die ASA muss manuell konfiguriert werden, um auf den primären RSA-Server zu zeigen. Ein Kennwort ist nicht erforderlich. Nach dem ersten erfolgreichen Authentifizierungsknoten wird die geheime Datei installiert (SDI-Datei im Flash-Speicher) und weitere Authentifizierungssitzungen sind geschützt. Auch die IP-Adresse anderer RSA-Server wird heruntergeladen.

Hier ein Beispiel:

```
aaa-server SDI protocol sdi
aaa-server SDI (backbone) host 1.1.1.1
debug sdi 255
test aaa auth SDI host 1.1.1.1 user test pass 321321321
```

Nach erfolgreicher Authentifizierung zeigt der Befehl **show aaa-server protocol sdi** oder **show aaa-server <aaa-server-group>** alle RSA-Server an (wenn es mehrere Server gibt), während der Befehl **show run** nur die primäre IP-Adresse anzeigt:

```
bsns-asa5510-17# show aaa-server RSA
Server Group:    RSA
Server Protocol: sdi
Server Address:  10.0.0.101
Server port:     5500
Server status:   ACTIVE (admin initiated), Last transaction at
10:13:55 UTC Sat Jul 27 2013
Number of pending requests              0
Average round trip time                 706ms
Number of authentication requests       4
Number of authorization requests        0
Number of accounting requests           0
Number of retransmissions               0
Number of accepts                       1
Number of rejects                       3
Number of challenges                    0
Number of malformed responses           0
Number of bad authenticators            0
Number of timeouts                      0
Number of unrecognized responses        0


SDI Server List:
       Active Address:          10.0.0.101
       Server Address:          10.0.0.101
       Server port:             5500
       Priority:                0
       Proximity:               2
        Status:                  OK
       Number of accepts                     0
       Number of rejects                     0
       Number of bad next token codes        0
       Number of bad new pins sent           0
       Number of retries                     0
       Number of timeouts                    0

       Active Address:          10.0.0.102
       Server Address:          10.0.0.102
       Server port:             5500
       Priority:                8
       Proximity:               2
        Status:                  OK
```

```
        Number of accepts                       1
        Number of rejects                       0
        Number of bad next token codes          0
        Number of bad new pins sent             0
        Number of retries                       0
        Number of timeouts                      0
```

Fehlerbehebung

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

Keine Agentenkonfiguration auf RSA

In vielen Fällen nach der Installation einer neuen ASA oder Änderung der ASA-IP-Adresse ist es leicht zu vergessen, dieselben Änderungen an der RSA vorzunehmen. Die Agent-IP-Adresse auf dem RSA muss für alle Clients aktualisiert werden, die auf das RSA zugreifen. Dann wird der neue Knotengeheim generiert. Das Gleiche gilt für den ACS, insbesondere für sekundäre Knoten, da diese über unterschiedliche IP-Adressen verfügen und ihnen der RSA vertrauen muss.

Beschädigter geheimer Knoten

Manchmal wird die geheime Knotendatei auf der ASA oder der RSA beschädigt. Anschließend empfiehlt es sich, die Agent-Konfiguration auf dem RSA zu entfernen und erneut hinzuzufügen. Sie müssen den gleichen Prozess auch auf der ASA/ACS ausführen - Konfiguration entfernen und erneut hinzufügen. Löschen Sie außerdem die SDI-Datei im Flash-Speicher, sodass bei der nächsten Authentifizierung eine neue SDI-Datei installiert wird. Die automatische, geheime Knotenbereitstellung sollte erfolgen, sobald diese abgeschlossen ist.

Knoten im ausgesetzten Modus

Manchmal befindet sich einer der Knoten im Suspendiermodus, was darauf zurückzuführen ist, dass der Server nicht reagiert:

```
asa# show aaa-server RSA
<.....output ommited"
SDI Server List:
Active Address: 10.0.0.101
Server Address: 10.0.0.101
Server port: 5500
Priority: 0
Proximity: 2
  Status:                  SUSPENDED
```

Im ausgesetzten Modus versucht die ASA nicht, Pakete an diesen Knoten zu senden. dafür muss er einen **OK**-Status haben. Der ausgefallene Server wird nach dem Dead-Timer erneut in den aktiven Modus versetzt. Weitere Informationen finden Sie im Abschnitt zum Reaktivierungsmodus in der Cisco ASA Series Command Reference, 9.1 Guide.

In solchen Szenarien empfiehlt es sich, die AAA-Serverkonfiguration für diese Gruppe zu

entfernen und hinzuzufügen, um den Server erneut in den aktiven Modus zu schalten.

Konto gesperrt

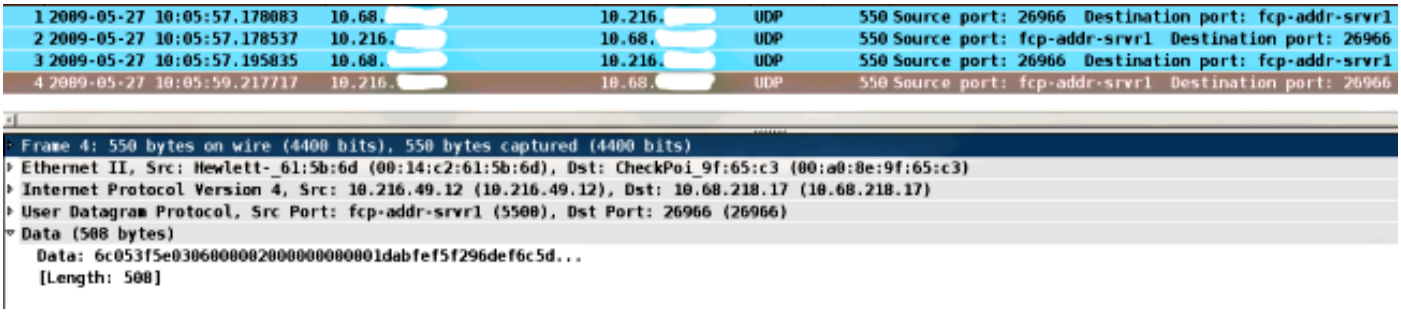
Nach mehreren erneuten Versuchen kann sich das RSA vom Konto abmelden. Es ist auf dem RSA mit Berichten einfach zu überprüfen. Auf ASA/ACS wird in Berichten nur "fehlgeschlagene Authentifizierung" angezeigt.

Maximale Anzahl von Problemen und Fragmentierungen bei Übergangseinheiten (MTU)

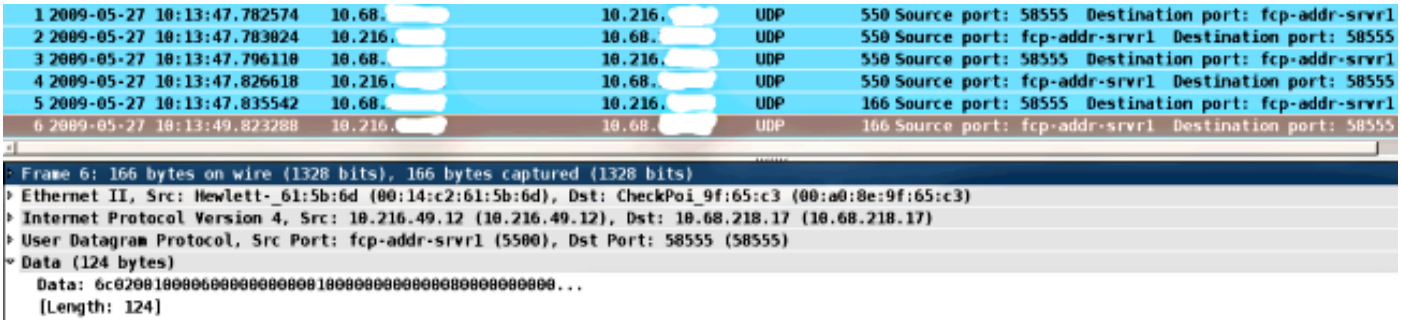
SDI verwendet UDP als Transport, nicht als MTU-Pfaderkennung. Auch für UDP-Datenverkehr ist das DF-Bit (Don't Fragment) nicht standardmäßig festgelegt. Bei größeren Paketen kann es manchmal zu Fragmentierungsproblemen kommen. Es ist einfach, Datenverkehr auf der RSA zu schnüffeln (sowohl die Appliance als auch das virtuelle System [VM] verwenden Windows und Wireshark). Führen Sie denselben Prozess auf ASA/ACS durch, und vergleichen Sie ihn. Testen Sie außerdem RADIUS oder WebAuthentication auf dem RSA, um es mit SDI zu vergleichen (um das Problem einzugrenzen).

Pakete und Debugger für ACS

Da die SDI-Nutzlast verschlüsselt ist, besteht die einzige Möglichkeit zur Fehlerbehebung in einem Vergleich der Größe der Antwort. Wenn sie kleiner als 200 Byte ist, kann ein Problem auftreten. Ein typischer SDI-Austausch umfasst vier Pakete, von denen jedes 550 Byte groß ist, die sich jedoch mit der RSA-Serverversion ändern können:



Im Problemfall werden in der Regel mehr als vier Pakete ausgetauscht und kleinere Größen:



Außerdem sind die ACS-Protokolle ziemlich klar. Nachfolgend sind typische SDI-Protokolle für ACS aufgeführt:

```
EventHandler,11/03/2013,13:47:58:416,DEBUG,3050957712,Stack: 0xa3de560
Calling backRSAIDStore: Method MethodCaller<RSAIDStore, RSAAgentEvent> in
thread:3050957712,EventStack.cpp:242

AuthenSessionState,11/03/2013,13:47:58:416,DEBUG,3050957712,cntx=0000146144,
sesn=acs-01/150591921/1587,user=mickey.mouse,[RSACheckPasscodeState
::onEnterState],RSACheckPasscodeState.cpp:23

EventHandler,11/03/2013,13:47:58:416,DEBUG,3002137488,Stack: 0xa3de560
Calling RSAAgent:Method MethodCaller<RSAAgent, RSAAgentEvent> in thread:
3002137488,EventStack.cpp:204

RSAAgent,11/03/2013,13:47:58:416,DEBUG,3002137488,cntx=0000146144,sesn=
acs-01/150591921/1587,user=mickey.mouse,[RSAAgent::handleCheckPasscode],
RSAAgent.cpp:319

RSASessionHandler,11/03/2013,13:47:58:416,DEBUG,3002137488,[RSASessionHandler::
checkPasscode] call AceCheck,RSASessionHandler.cpp:251

EventHandler,11/03/2013,13:48:00:417,DEBUG,2965347216,Stack: 0xc14bba0
Create newstack, EventStack.cpp:27

EventHandler,11/03/2013,13:48:00:417,DEBUG,3002137488,Stack: 0xc14bba0 Calling
RSAAgent: Method MethodCaller<RSAAgent, RSAServerResponseEvent> in
thread:3002137488,EventStack.cpp:204

RSAAgent,11/03/2013,13:48:00:417,DEBUG,3002137488,cntx=0000146144,sesn=acs-01
/150591921/1587,user=mickey.mouse,[RSAAgent::handleResponse] operation completed
with ACM_OKstatus,RSAAgent.cpp:237

EventHandler,11/03/2013,13:48:00:417,DEBUG,3002137488,Stack: 0xc14bba0
EventStack.cpp:37

EventHandler,11/03/2013,13:48:00:417,DEBUG,3049905040,Stack: 0xa3de560 Calling
back RSAIDStore: Method MethodCaller<RSAIDStore, RSAAgentEvent> in thread:
3049905040,EventStack.cpp:242

AuthenSessionState,11/03/2013,13:48:00:417,DEBUG,3049905040,cntx=0000146144,sesn=
acs-01/150591921/1587,user=mickey.mouse,[RSACheckPasscodeState::onRSAAgentResponse]
Checkpasscode succeeded, Authentication passed,RSACheckPasscodeState.cpp:55
```

Zugehörige Informationen

- RSA Authentication Manager-Ressourcen
- RSA/SDI Server Support im Abschnitt Cisco ASA 5500 Series Configuration Guide unter Verwendung der CLI, 8.4 und 8.6
- RSA SecurID Server im Benutzerhandbuch für Cisco Secure Access Control System 5.4
- Technischer Support und Dokumentation - Cisco Systems