ASA 및 ACS용 RSA 토큰 서버 및 SDI 프로토콜 사용

목차

소개

이 문서에서는 Cisco ASA(Adaptive Security Appliance) 및 Cisco ACS(Secure Access Control Server)와 통합될 수 있는 RSA Authentication Manager의 문제 해결 절차에 대해 설명합니다.

RSA Authentication Manager는 인증을 위한 OTP(One Time Password)를 제공하는 솔루션입니다.이 비밀번호는 60초마다 변경되며 한 번만 사용할 수 있습니다.하드웨어 및 소프트웨어 토큰을 모두 지원합니다.

사전 요구 사항

요구 사항

Cisco에서는 이러한 주제에 대한 기본적인 지식을 얻을 것을 권장합니다.

- Cisco ASA CLI 컨피그레이션
- Cisco ACS 컨피그레이션

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 버전을 기반으로 합니다.

- Cisco ASA 소프트웨어, 버전 8.4 이상
- Cisco Secure ACS, 버전 5.3 이상

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다.이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다.현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

이론

RADIUS 또는 전용 RSA 프로토콜을 사용하여 RSA 서버에 액세스할 수 있습니다.SDI.ASA와 ACS는 모두 RSA에 액세스하기 위해 두 프로토콜(RADIUS, SDI)을 모두 사용할 수 있습니다.

소프트웨어 토큰을 사용할 때 RSA를 Cisco AnyConnect Secure Mobility Client와 통합할 수 있습니다.이 문서에서는 ASA 및 ACS 통합에만 초점을 맞춥니다.AnyConnect에 대한 자세한 내용은 Cisco AnyConnect Secure Mobility Client 관리자 설명서, 릴리스 3.1의 SDI 인증 사용 섹션을 참조하십시오.

RADIUS를 통한 RSA

RADIUS는 SDI에 비해 한 가지 큰 장점을 가지고 있습니다.RSA에서는 특정 프로파일(ACS의 그룹이라고 함)을 사용자에게 할당할 수 있습니다.이러한 프로파일에는 특정 RADIUS 특성이 정의되어 있습니다.인증에 성공하면 RSA에서 반환된 RADIUS-Accept 메시지에 해당 특성이 포함됩니다.이러한 특성에 따라 ACS는 추가 결정을 합니다.가장 일반적인 시나리오는 RSA의 프로필과 관련된 특정 RADIUS 특성을 ACS의 특정 그룹에 매핑하기 위해 ACS 그룹 매핑을 사용하는 것입니다.이 논리를 사용하면 RSA에서 ACS로 전체 인증 프로세스를 이동하고 RSA에서와 같이 세분화된 논리를 유지할 수 있습니다.

SDI를 통한 RSA

SDI는 RADIUS에 비해 두 가지 주요 장점을 가지고 있습니다.첫 번째는 전체 세션이 암호화된다는 것입니다.두 번째 옵션은 SDI 에이전트에서 제공하는 흥미로운 옵션입니다.인증 또는 권한 부여에 실패했거나 사용자를 찾을 수 없어 오류가 생성되었는지 확인할 수 있습니다.

이 정보는 ACS가 ID를 위해 작업에 사용합니다.예를 들어 "user not found"에 대해 계속 진행하지만 "authentication failed"에 대해서는 거부될 수 있습니다.

RADIUS와 SDI의 차이점이 하나 더 있습니다.ASA와 같은 네트워크 액세스 디바이스가 SDI를 사용하는 경우 ACS는 인증만 수행합니다.RADIUS를 사용하는 경우 ACS는 인증, 권한 부여, 계정 관리(AAA)를 수행합니다. 하지만 큰 차이는 아니다.인증을 위해 SDI를 구성하고 동일한 세션에 대해 어카운팅할 RADIUS를 구성할 수 있습니다.

SDI 프로토콜

기본적으로 SDI는 UDP(User Datagram Protocol) 5500을 사용합니다.SDI는 세션을 암호화하기 위해 RADIUS 키와 유사한 대칭 암호화 키를 사용합니다.이 키는 노드 비밀 파일에 저장되며 모든 SDI 클라이언트에 대해 다릅니다.이 파일은 수동 또는 자동으로 구축됩니다.

> **참고:**ACS/ASA는 수동 구축을 지원하지 않습니다.

자동 배포 노드의 경우 첫 번째 인증 성공 후 비밀 파일이 자동으로 다운로드됩니다.노드 암호는 사용자의 암호 및 기타 정보에서 파생된 키로 암호화됩니다.이렇게 하면 몇 가지 보안 문제가 발생할 수 있으므로 첫 번째 인증이 로컬에서 수행되어야 하며 암호화된 프로토콜(텔넷이 아닌 SSH[Secure Shell])을 사용하여 공격자가 해당 파일을 가로채고 해독할 수 없도록 해야 합니다.

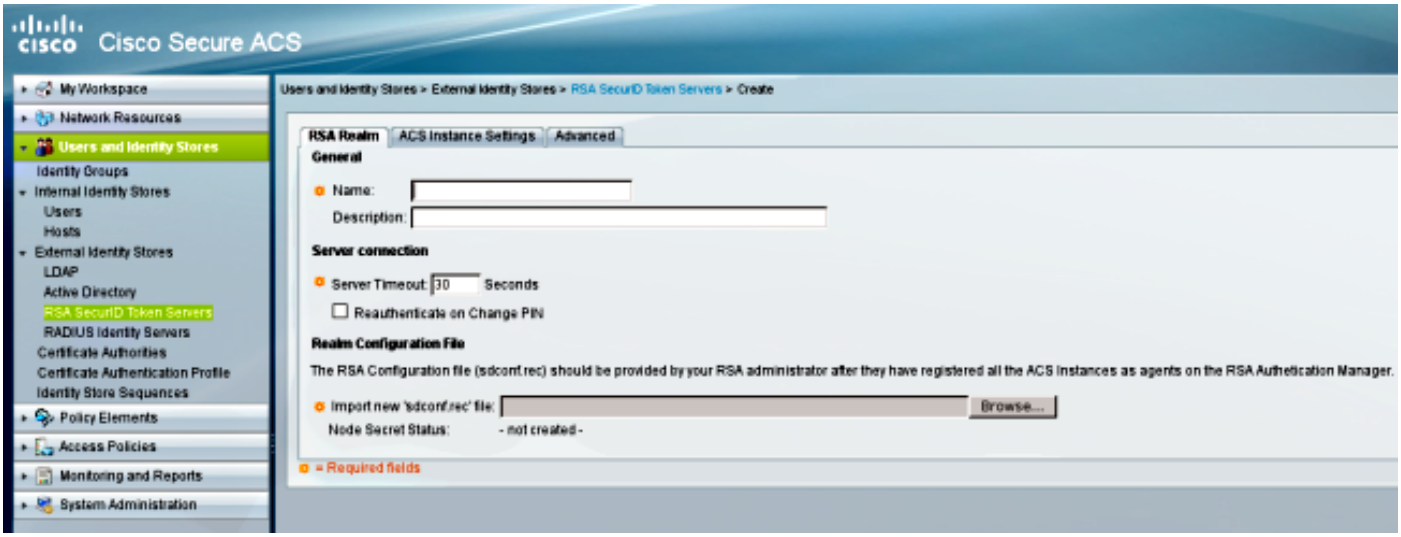
구성

> **참고:**
>
> 이 섹션에 사용된 명령에 대한 자세한 내용을 보려면 Command Lookup Tool(등록된 고객만 해당)을 사용합니다.
>
> Output Interpreter 도구(등록된 고객만 해당)는 특정 **show** 명령을 지원합니다.**show** 명령 출력의 분석을 보려면 [출력 인터프리터 도구]를 사용합니다.
>
> **debug** 명령을 사용하기 전에 디버그 명령에 대한 중요 정보를 참조하십시오.

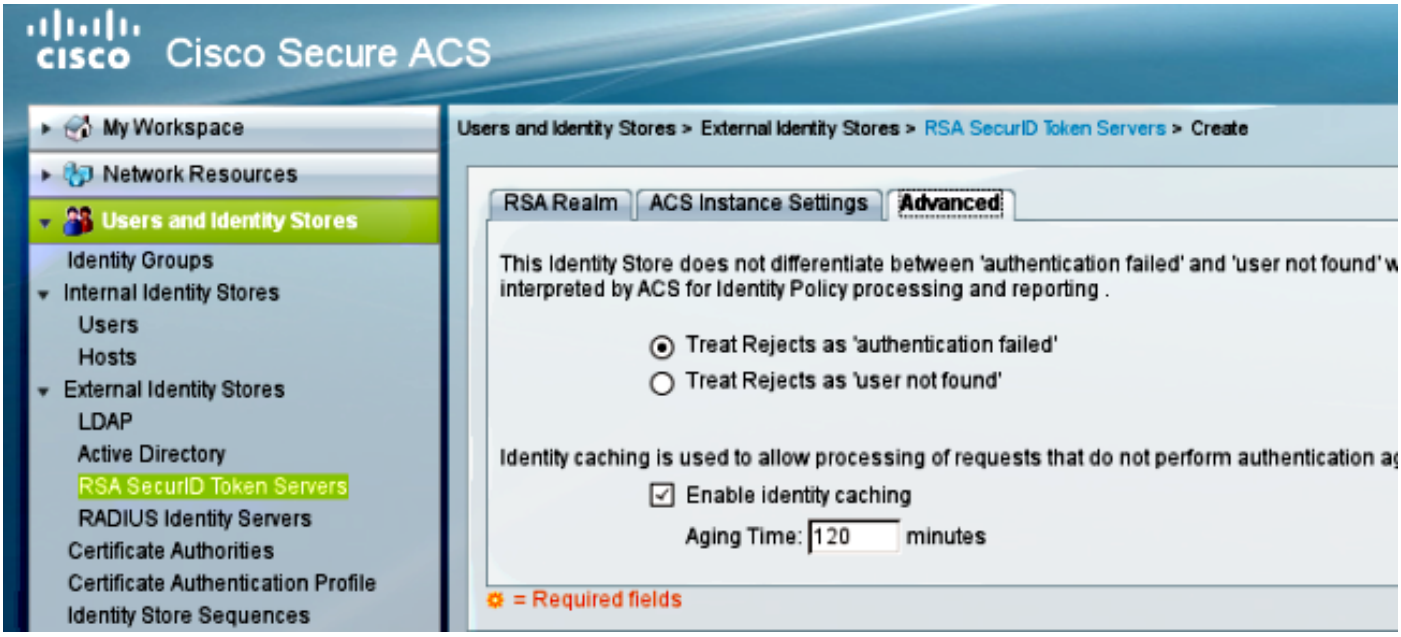
ACS의 SDI

Users and **Identity Stores(사용자 및 ID 저장소) > External Identity Store(외부 ID 저장소) > RSA Secure ID Token Servers(RSA 보안 ID 토큰 서버)에서 구성됩니다**.

RSA에는 ACS용 보조 서버와 같은 여러 복제본 서버가 있습니다.RSA 관리자가 제공한 **sdconf.rec** 파일만 있으면 모든 주소를 여기에 넣을 필요가 없습니다.이 파일에는 기본 RSA 서버의 IP 주소가 포함됩니다.첫 번째 성공 적 인 인증 노드 후 비밀 파일은 모든 RSA 복제본의 IP 주소와 함께 다운로드 됩니다.



"user not found(사용자를 찾을 수 없음)"와 "authentication failure(인증 실패)"를 구별하려면 **Advanced(고급)** 탭에서 설정을 선택합니다.



여러 RSA 서버(기본 및 복제본) 간의 기본 라우팅(로드 밸런싱) 메커니즘을 변경할 수도 있습니다. RSA 관리자가 제공한 **sdopts.rec** 파일로 변경합니다.ACS에서는 Users and Identity Stores(사용자 및 ID 저장소) **> External Identity Store(외부 ID 저장소) > RSA Secure ID Token Servers(RSA 보안 ID 토큰 서버) > ACS Instance Settings(ACS 인스턴스 설정)에 업로드됩니다**.

클러스터 구축의 경우 컨피그레이션을 복제해야 합니다.첫 번째 성공적인 인증 후 각 ACS 노드는 기본 RSA 서버에서 다운로드한 자체 노드 암호를 사용합니다.클러스터의 모든 ACS 노드에 대해 RSA를 구성해야 합니다.

ASA의 SDI

ASA는 sdconf**.rec** 파일의 업로드를 허용하지 않습니다.또한 ACS와 마찬가지로 자동 구축만 가능합니다.기본 RSA 서버를 가리키려면 ASA를 수동으로 구성해야 합니다.비밀번호는 필요하지 않습니다.첫 번째 성공 적 인 인증 노드 후, 비밀 파일 (.sdi 파일 on flash) 이 설치 되고 추가 인증 세션이 보호 됩니다.다른 RSA 서버의 IP 주소도 다운로드됩니다.

예를 들면 다음과 같습니다.

```
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
```

인증이 성공하면 **show aaa-server protocol sdi** 또는 **show aaa-server <aaa-server-group>** 명령은 모든 RSA 서버(둘 이상 있는 경우)를 표시하고 **show run** 명령은 기본 IP 주소만 표시합니다.

```
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
```

문제 해결

이 섹션에서는 컨피그레이션 문제를 해결하는 데 사용할 수 있는 정보를 제공합니다.

RSA에 에이전트 컨피그레이션 없음

새 ASA를 설치하거나 ASA IP 주소를 변경한 후 RSA에서 동일한 변경 사항을 적용하는 경우가 많습니다.RSA에 액세스하는 모든 클라이언트에 대해 RSA의 에이전트 IP 주소를 업데이트해야 합니다.그런 다음 새 노드 암호가 생성됩니다.ACS에도 동일하게 적용됩니다. 특히 보조 노드는 서로 다른 IP 주소를 가지고 있고 RSA는 이들을 신뢰해야 하기 때문입니다.

손상된 암호 노드

ASA 또는 RSA의 비밀 노드 파일이 손상되는 경우가 있습니다.그런 다음 RSA에서 에이전트 컨피그레이션을 제거하고 다시 추가하는 것이 좋습니다.또한 ASA/ACS에서 동일한 프로세스를 수행해야 합니다. 구성을 제거하고 다시 추가해야 합니다.또한 다음 인증에서 새 .sdi 파일이 설치되도록 플래시에서 .sdi 파일을 삭제합니다.자동 노드 암호 구축은 이 작업이 완료되면 수행해야 합니다.

일시 중단 모드의 노드

노드 중 하나가 일시 중단 모드에 있는 경우가 있는데, 이는 해당 서버로부터 응답이 없기 때문입니다.

```
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
```

일시 중단 모드에서는 ASA가 해당 노드에 패킷을 전송하려고 시도하지 않습니다.상태가 **OK** 상태여야 합니다.장애가 발생한 서버가 데드 타이머 후에 다시 활성 모드로 전환됩니다.자세한 내용은 Cisco ASA Series 명령 참조, 9.1 가이드의 reactivation-mode 명령 섹션을 참조하십시오.

이러한 시나리오에서는 해당 서버를 다시 활성 모드로 트리거하려면 해당 그룹에 대한 AAA-server 컨피그레이션을 제거하고 추가하는 것이 좋습니다.

계정 잠김

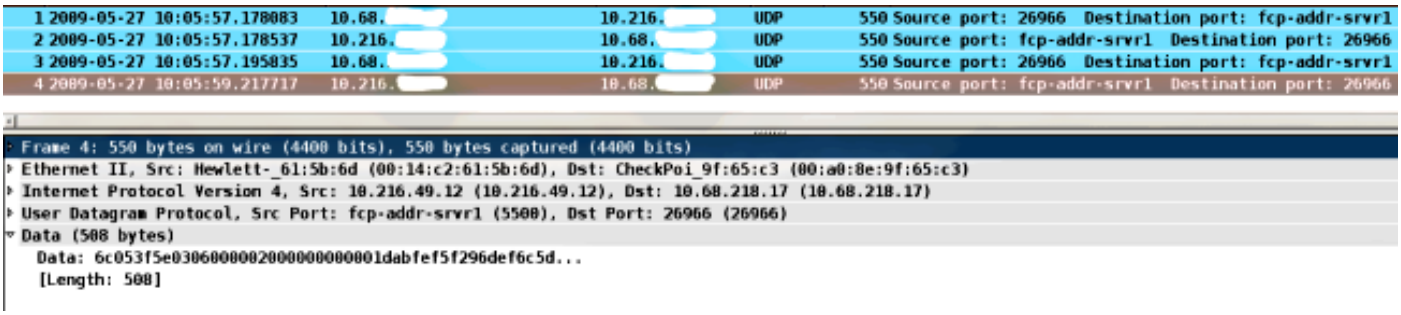
여러 번 재시도한 후 RSA는 어카운트에서 잠길 수 있습니다.RSA에서 보고서를 쉽게 확인할 수 있습니다.ASA/ACS에서 보고서에는 "실패한 인증"만 표시됩니다.

MTU(Maximum Transition Unit) 문제 및 단편화

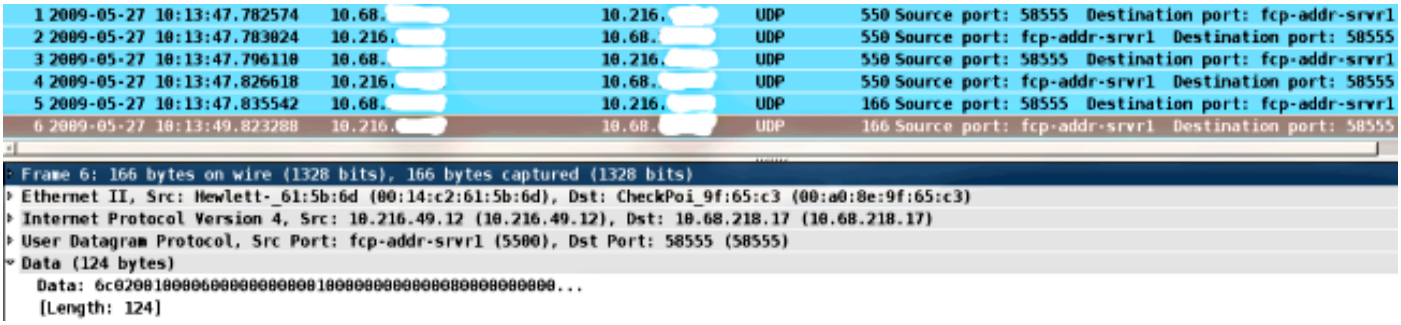
SDI는 MTU 경로 검색이 아닌 전송으로 UDP를 사용합니다.또한 UDP 트래픽에는 기본적으로 DF(Don't Fragment) 비트가 설정되어 있지 않습니다.대규모 패킷의 경우 단편화 문제가 있을 수 있습니다.RSA에서 트래픽을 스니핑하기 쉽습니다(어플라이언스와 가상 머신[VM] 모두 Windows를 사용하고 Wireshark를 사용합니다). ASA/ACS에서 동일한 프로세스를 완료하고 비교합니다.또한 RSA에서 RADIUS 또는 WebAuthentication을 테스트하여 문제를 좁히기 위해 SDI와 비교합니다.

ACS용 패킷 및 디버그

SDI 페이로드는 암호화되므로 캡처 문제를 해결하는 유일한 방법은 응답의 크기를 비교하는 것입니다.200바이트보다 작으면 문제가 발생할 수 있습니다.일반적인 SDI 교환에는 4개의 패킷이 포함되며, 각 패킷은 550바이트이지만 RSA 서버 버전으로 변경될 수 있습니다.



문제가 발생할 경우 일반적으로 교환된 패킷이 4개 이상이고 크기가 작습니다.



또한 ACS 로그는 매우 명확합니다.다음은 ACS의 일반적인 SDI 로그입니다.

```
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
```

관련 정보

- RSA Authentication Manager 리소스
- CLI, 8.4 및 8.6을 사용하는 Cisco ASA 5500 Series 컨피그레이션 가이드의 RSA/SDI 서버 지원 섹션
- Cisco Secure Access Control System 5.4 사용 설명서의 RSA SecurID Server 섹션
- 기술 지원 및 문서 – Cisco Systems