디바이스 연결

• Secure Device Connector, 1 페이지

Secure Device Connector

Secure Device Connector(SDC)는 Cisco 디바이스가 Security Cloud Control와 통신하도록 허용하는 인텔리전트 프록시입니다. 디바이스 자격 증명을 사용하여 인터넷을 통해 직접 연결할 수 없는 디바이스를 Security Cloud Control에 온보딩하는 경우, 네트워크에서 SDC를 구축하여 디바이스와 Security Cloud Control 간의 통신을 프록시할 수 있습니다. 아니면 원하는 경우 Security Cloud Control에서 외부 인터페이스를 통해 직접 통신을 수신하도록 디바이스를 활성화할 수 있습니다.

ASA(Adaptive Security Appliances), Meraki MXs, Secure Firewall Management Center 디바이스, 일반 SSH 및 iOS 디바이스는 모두 SDC를 사용하여 Security Cloud Control에 온보딩할 수 있습니다. 클라우드 제공 Firewall Management Center 에서 관리하는 Secure Firewall Threat Defense 디바이스는 SDC를 사용한 온보딩이 필요하지 않으며, 프록시를 통한 온보딩을 지원하지 않습니다. 위협 방어 디바이스에 클라우드 제공 Firewall Management Center에 연결하기 위한 적절한 DNS 설정 및 아웃바운드 인터넷 연결성이 있는지 확인합니다. 자세한 내용은 온보딩 개요를 참조하십시오.

참고 Secure Device Connector 및 Secure Event Connector를 생성하려면 슈퍼 관리자 사용자 역할이 필요합니다.

SDC는 매니지드 디바이스에서 실행해야 하는 명령과 매니지드 디바이스로 전송해야 하는 메시지에 대해 Security Cloud Control를 모니터링합니다. SDC는 Security Cloud Control를 대신하여 명령을 실행하고, 매니지드 디바이스를 대신하여 Security Cloud Control에 메시지를 전송하고, 매니지드 디바이스에서 Security Cloud Control로 응답을 반환합니다.

SDC은 AES-128-GCM over HTTPS(TLS 1.3)를 사용하여 서명 및 암호화된 보안 통신 메시지를 통해 Security Cloud Control와 통신합니다. 온보딩된 디바이스 및 서비스에 대한 모든 자격 증명은 브라우저에서 SDC로 직접 암호화되며, AES-128-GCM을 사용하여 저장 상태에서도 암호화됩니다. SDC만 디바이스 자격 증명에 액세스할 수 있습니다. 다른 Security Cloud Control 서비스는 자격 증명에 액세스할 수 없습니다.

SDC와 Security Cloud Control 간의 통신을 허용하는 방법에 대한 자세한 내용은 매니지드 디바이스에 Security Cloud Control 연결, 2 페이지의 내용을 참조하십시오.

SDC는 모든 Ubuntu 인스턴스에 설치할 수 있습니다. 편의를 위해 SDC CLI가 사전 설치되어 있는 강화된 Ubuntu 22 인스턴스용 OVA를 제공합니다. CLI는 가상 머신을 구성하고, 필요한 모든 시스템 패키지를 설치하며, 호스트에서 SDC를 Docker 컨테이너로 부트스트랩하는 데 도움이 됩니다. 또는 자체 Ubuntu 인스턴스(현재 테스트 중인 버전 20~24)를 롤링하고 CLI를 별도로 다운로드할 수도 있습니다.

각 Security Cloud Control 테넌트에는 무제한의 SDC가 있을 수 있습니다. 이러한 SDC는 테넌트 간에 공유되지 않으며 단일 테넌트 전용입니다. 단일 SDC에서 관리할 수 있는 디바이스의 수는 해당 디바이스에 구현된 기능 및 해당 구성 파일의 크기에 따라 달라집니다. 그러나 구축을 계획할 때는 1개의 SDC가 약 500개의 디바이스를 지원할 것으로 예상합니다.

테넌트에 대해 둘 이상의 SDC를 구축하면 다음과 같은 이점도 제공됩니다.

- 성능 저하 없이 Security Cloud Control 테넌트로 더 많은 디바이스를 관리할 수 있습니다.
- 네트워크 내의 격리된 네트워크 세그먼트에 SDC를 구축하고 동일한 Security Cloud Control 테넌트로 해당 세그먼트의 디바이스를 계속 관리할 수 있습니다. 여러 SDC이 없으면 서로 다른 Security Cloud Control 테넌트를 사용하여 격리된 네트워크 세그먼트에서 디바이스를 관리해야 합니다.

단일 호스트에서 여러 SDC를 실행할 수 있으며, 실행하려는 각 SDC에 대해 부트스트랩 절차를 따르십시오. 테넌트의 초기 SDC는 테넌트의 이름과 숫자 1을 통합하며 Security Cloud Control의 **Services(**서비스**)** 페이지에 있는 **Secure Connectors(**보안 커넥터**)** 탭에 표시됩니다. 각 추가 SDC는 순서대로 번호가 매겨집니다.

자세한 내용은 Secure Device Connector 및 Secure Event Connector 실행을 위한 VM 구축, 4 페이지을 참조하십시오.

관련 정보**:**

- Cisco Defense Orchestrator를 Secure Device Connector에 연결
- Secure Device Connector 업데이트, 22 페이지
- Secure Device Connector 제거, 21 페이지

매니지드 디바이스에 **Security Cloud Control** 연결

Security Cloud Control는 클라우드 커넥터 또는 SDC(Secure Device Connector)를 통해 관리하는 디바이스에 연결합니다.

인터넷에서 디바이스에 직접 액세스할 수 있는 경우 클라우드 커넥터를 사용하여 디바이스에 연결해야 합니다. 디바이스를 구성할 수 있는 경우 클라우드 지역의 Security Cloud Control IP 주소에서 포트 443에 대한 인바운드 액세스를 허용합니다.

인터넷에서 디바이스에 액세스할 수 없는 경우 Security Cloud Control가 디바이스와 통신할 수 있도록 네트워크에 온프레미스 SDC를 구축할 수 있습니다.

포트 443(또는 디바이스 관리를 위해 구성한 포트)에서 디바이스 서브넷/IP의 전체 인바운드 액세스를 허용하도록 디바이스를 구성합니다.

다른 모든 디바이스와 서비스는 Security Cloud Control이 클라우드 커넥터를 사용하여 연결하므로 온프레미스 SDC가 필요하지 않습니다. 인바운드 액세스를 허용해야 하는 IP 주소를 확인하려면 다음 섹션을 참조하십시오.

클라우드 커넥터를 통해 디바이스를 **Security Cloud Control**에 연결

클라우드 커넥터를 통해 Security Cloud Control를 디바이스에 직접 연결할 때는 EMEA, 미국 또는 APJ 지역의 다양한 IP 주소에 대해 포트 443(또는 디바이스 관리를 위해 구성한 모든 포트)에서 인바운드 액세스를 허용해야 합니다.

AJP(Asia-Pacific-Japan) 고객이고, https://apj.manage.security.cisco.com에서 Security Cloud Control에 연결하는 경우 다음 IP 주소에서 인바운드 액세스를 허용합니다.

- 54.199.195.111
- 52.199.243.0

Australia(호주**) (AUS)** 지역의 고객이고, https://aus.manage.security.cisco.com에서 Security Cloud Control에 연결하는 경우 다음 IP 주소에서 인바운드 액세스를 허용합니다.

- 13.55.73.159
- 13.238.226.118

유럽**,** 중동 또는 아프리카**(EMEA)** 지역의 고객이 https://eu.manage.security.cisco.com에서 Security Cloud Control에 연결하는 경우 다음 IP 주소에서의 인바운드 액세스를 허용합니다.

- 35.157.12.126
- 35.157.12.15

India(인도**) (IN)** 지역의 고객이고, https://in.manage.security.cisco.com에서 Security Cloud Control에 연결하는 경우 다음 IP 주소에서 인바운드 액세스를 허용합니다.

- 35.154.115.175
- 13.201.213.99

US 지역의 고객이고, https://us.manage.security.cisco.com에서 Security Cloud Control에 연결하는 경우, 다음 IP 주소에서 인바운드 액세스를 허용합니다.

- 52.34.234.2
- 52.36.70.147

SDC에 Security Cloud Control 연결

SDC를 통해 Security Cloud Control를 디바이스에 연결할 때 Security Cloud Control에서 관리하려는 디바이스는 포트 443(또는 디바이스 관리를 위해 구성한 포트)에서 SDC 호스트의 전체 인바운드 액세스를 허용해야 합니다. 이는 관리 액세스 제어 규칙을 사용하여 구성됩니다.

또한 SDC가 구축된 가상 머신이 매니지드 디바이스의 관리 인터페이스에 네트워크로 연결되어 있는지 확인해야 합니다.

Secure Device Connector 및 Secure Event Connector 실행을 위한 VM 구축

디바이스 자격 증명을 사용하여 Security Cloud Control를 디바이스에 연결하는 경우 네트워크에 SDC를 다운로드하고 구축하여 Security Cloud Control와 디바이스 간의 통신을 관리하는 것이 가장 좋습니다. 일반적으로 이러한 디바이스는 경계를 기반으로 하지 않으며 공용 IP 주소가 없거나 외부 인터페이스에 대한 개방형 포트가 있습니다.

SDC는 매니지드 디바이스에서 실행해야 하는 명령과 매니지드 디바이스로 전송해야 하는 메시지에 대해 Security Cloud Control를 모니터링합니다. SDC는 Security Cloud Control를 대신하여 명령을 실행하고, 매니지드 디바이스를 대신하여 Security Cloud Control에 메시지를 전송하고, 매니지드 디바이스에서 Security Cloud Control로 응답을 반환합니다.

단일 SDC에서 관리할 수 있는 디바이스의 수는 해당 디바이스에 구현된 기능 및 해당 구성 파일의 크기에 따라 달라집니다. 그러나 구축을 계획할 때는 1개의 SDC가 약 500개의 디바이스를 지원할 것으로 예상합니다. 자세한 정보는 단일 Security Cloud Control 테넌트에서 여러 SDC 사용을 참조하십시오.

이 절차에서는 Security Cloud Control의 VM 이미지를 사용하여 네트워크에 SDC를 설치하는 방법을 설명합니다. 이는 SDC를 생성하는 권장 방법입니다.

시작하기 전에

- Security Cloud Control는 엄격한 인증서 확인이 필요하며 SDC(Secure Device Connector)와 인터넷 간의 웹 또는 콘텐츠 프록시 검사를 지원하지 않습니다. 프록시 서버를 사용하는 경우 SDC와 Security Cloud Control 간의 트래픽 검사를 비활성화합니다.
- SDC는 TCP 포트 443 또는 디바이스 관리를 위해 구성한 포트에서 인터넷에 대한 전체 아웃바운드 액세스 권한을 가져야 합니다.
- Security Cloud Control에서 관리하는 디바이스는 SDC VM의 IP 주소로부터의 인바운드 트래픽을 허용해야 합니다.
- 매니지드 디바이스에 Security Cloud Control 연결, 2 페이지를 Secure Device Connector에 연결을 검토하여 적절한 네트워크 액세스를 확인합니다.
- 네트워크에서 프록시를 사용하는 경우, 호스트 설정 명령을 실행하기 전에 필요한 모든 세부 정보를 준비합니다. 문제는 잘못된 프록시 설정과 관련이 있습니다. 중요한 세부 정보:
  - IP/호스트 이름.
  - 프록시가 트래픽을 가로채고 자체 인증서를 사용하여 다시 암호화하는지 여부. 이 세부 정보 때문에 SDC VM 설정과 관련된 대부분의 복잡성이 발생합니다.

- 프록시가 트래픽을 가로채는 경우, VM 구성 시 루트 인증서를 준비하십시오. 메시지가 표시되면 붙여넣어 호스트 및 SDC가 프록시에서 생성한 인증서를 신뢰함을 알 수 있도록 할 수 있습니다.
- 프록시가 트래픽을 가로채지 않는다면, 여기에 추가로 필요한 것은 없습니다.

- 다음 항목들은 프록시된 HTTP 및 HTTPS 연결에서 동일할 가능성이 가장 높습니다. 그러나 각 프로토콜마다 다른 프록시를 사용하는 경우, 각각에 대해 다음의 모든 것이 필요합니다.
  - 프록시 IP 주소
  - 사용하는 프록시 포트
  - 프록시 자체에 대한 연결이 HTTPS를 통해 이루어져야 하는지 여부(일반적으로 해당되지 않음). 예를 들어 프록시 주소가 https://proxy.corp.com:80로 나열되어 있으면 Yes로 응답합니다. 나열된 주소가 http://proxy.corp.com:80라면 No로 답해야 합니다. 두 URL 모두 포트 80을 사용하지만 프로토콜이 다르다는 점에 유의합니다.
  - 프록시 인증 세부 정보:
    - 프록시가 인증을 요구하는지 여부(대부분은 요구하지 않음)
    - Yes라면 호스트를 구성할 때 사용할 수 있는 사용자 이름과 비밀번호가 필요합니다.

지원되는 설치

- Security Cloud Control는 vSphere 웹 클라이언트 또는 ESXi 웹 클라이언트를 사용하여 SDC VM OVF 이미지 설치를 지원합니다.
- Security Cloud Control는 vSphere 데스크톱 클라이언트를 사용한 SDC VM OVF 이미지 설치를 지원하지 않습니다.
- Security Cloud Control에서는 자체 Ubuntu 인스턴스에 SDC 설치를 지원합니다. 현재 버전 20LTS부터 24LTS까지 지원됩니다.
- ESXi 5.1 하이퍼바이저.

시스템 요구 사항

- 하나의 SDC를 가진 VM의 시스템 요구 사항:
  - vCPU 2개
  - 2GB 메모리
  - 64GB의 디스크 공간
- 호스트에 추가하는 각 SDC에는 vCPU 1개 및 1GB RAM이 필요합니다.

- Cisco Security Analytics 및 로깅에 사용되는 구성 요소인 SEC 하나를 탑재한 VM의 시스템 요구 사항:
  - vCPU 최소 4개
  - 8GB 메모리
- 호스트에 SEC를 추가할 때마다 해당 리소스를 두 배로 늘려야 합니다. 따라서 VMware ESXi 호스트에 SDC 1개와 SEC 1개를 구성할 경우 요구 사항은 다음과 같습니다.
  - vCPU 6개
  - 10GB 메모리
  - 64GB의 디스크 공간

설치 준비

- 호스트에서 네트워킹을 수동으로 구성하려면 다음 정보를 수집합니다.
  - SDC에 사용할 고정 IP 주소
  - `cdo`사용자(또는 sudo 액세스 권한이 있는 사용자) 및 'sdc' 사용자(Docker를 실행하는 사용자)에 사용할 비밀번호
  - 조직에서 사용하는 DNS 서버의 IP 주소
  - SDC 주소가 있는 네트워크의 게이트웨이 IP 주소
  - 시간/NTP 서버의 FQDN 또는 IP 주소.
- SDC 가상 머신은 보안 패치를 정기적으로 설치하도록 구성되며, 이를 위해서는 포트 80 아웃바운드를 열어야 합니다.

  네트워크에서 아웃바운드 연결에 허용/차단 목록을 사용하는 경우, 보안 업데이트를 적용할 수 있도록 ubuntu.com에 대한 연결을 허용해야 합니다.

참고 우분투는 체크섬으로 업데이트를 보호하며 HTTPS가 아닌 HTTP만 사용합니다. 보안 업데이트를 받으려면 ubuntu.com에 대한 HTTP 연결을 허용해야 합니다.

VM 구축

SDC 및 SEC를 실행하는 데 사용되는 VM을 구축하는 방법에는 두 가지 옵션이 있습니다.

1. 아래 단계에 따라 Security Cloud Control에서 제공하는 VMware 이미지를 다운로드합니다.
2. Ubuntu 20, 22 또는 24를 직접 구축. 자체 Ubuntu 인스턴스를 구축하는 경우, 다음 섹션을 건너뛰고 VM 구성 섹션으로 진행할 수 있습니다.

절차

1. 왼쪽 창에서 **Administration(**관리**)** > **Integration(**통합**)** > **Secure Connectors(**보안 커넥터**)**를 클릭합니다.
2. **Services(**서비스**)** 페이지에서 **Secure Connectors(**보안 커넥터**)** 탭을 선택하고 파란색 플러스 버튼을 클릭한 후 **Secure Device Connector(**보안 디바이스 커넥터**)**를 선택합니다.
3. **Download the SDC VM image(SDC VM** 이미지 다운로드**)**를 클릭합니다. 새 탭에서 열립니다.
4. .zip 파일의 모든 파일을 추출합니다. 다음과 같이 표시됩니다.
   - CDO-SDC-VM-ddd50fa.ovf
   - CDO-SDC-VM-ddd50fa.mf
   - CDO-SDC-VM-ddd50fa-disk1.vmdk
5. vSphere 웹 클라이언트를 사용하여 VMware 서버에 관리자로 로그인합니다.

참고 ESXi 웹 클라이언트를 사용하지 마십시오.

지시에 따라 OVF 템플릿에서 Secure Device Connector 가상 머신을 구축합니다.

6. 설정이 완료되면 SDC VM의 전원을 켭니다.
7. 새 SDC VM의 콘솔을 엽니다.
8. `cdo` 사용자 이름으로 로그인합니다. 기본 암호는 `adm123`입니다.

VM 구성

이제 구축한 VM 이미지의 콘솔을 가져올 수 있으며, 직접 VM을 구축하고 SSH를 활성화한 경우에는 SSH를 통해 접속할 수도 있습니다. SDC 또는 SEC Docker 컨테이너를 실행할 수 있도록 호스트를 준비하려면 구성 스크립트를 실행해야 합니다.

1. Security Cloud Control 제공 VM을 다운로드한 경우, CLI가 이미 설치되어 있으며 2단계로 진행할 수 있습니다. 자체 VM을 구축한 경우 SSH로 연결하고 명령을 실행하여 CLI를 설치합니다.

   ```
   curl -O
   https://s3.us-west-2.amazonaws.com/download.defenseorchestrator.com/sdc-cli/sdc-cli-package-latest.tgz
    && tar -xvf sdc-cli-package-latest.tgz && chmod +x ./install.sh && ./install.sh
   ```

2. 다음 명령어를 실행하여 호스트 구성을 시작합니다.

   ```
   sudo sdc host configure
   ```

3. 비밀번호가 표시되면 Security Cloud Control 제공 VM의 경우 adm123을 입력하거나 자체 VM에 대해 선택한 관리자 비밀번호를 입력합니다.
4. 프롬프트에 따라 `sdc` 사용자를 구성합니다.
5. 네트워크 구성에 대한 프롬프트가 표시되면 다음 중 하나를 선택합니다.

- 이 호스트에 정적 IP를 수동으로 구성: 이 호스트의 IP, 게이트웨이, DNS 서버 등을 지정하고 VM의 시스템 구성에 기록하려는 경우.
- DHCP: VM에 고정 IP를 할당하는 DHCP 서버가 있는 경우.
- 정적 IP는 이미 설정되어 있으며, 현재 네트워크 설정을 변경하고 싶지 않습니다.

6. 프록시 구성에 관한 질문이 표시되면 답변합니다. 이 주제의 상단에 있는 상세 목록을 검토하여 모든 필수 조건 및 잠재적인 프록시 구성 옵션을 확인합니다.
7. 프록시를 구성한 경우, 모든 프록시 설정을 적용하려면 VM을 재부팅하라는 메시지가 표시됩니다. 그렇지 않은 경우 재부팅을 요청받지 않으며 8단계로 진행할 수 있습니다.
8. 사용자 지정 인터넷 접속 테스트 URL 설정. 기본적으로 모든 아웃바운드 연결을 차단하는 경우에만 이 작업을 수행하면 됩니다. 해당 사항이 있다면, 허용 목록에 포함된 공개적으로 접근 가능한 웹 URL(예: https://google.com)을 지정합니다.
9. 최신 보안 패치를 설치합니다. 일부는 운영체제 도구와 Docker 서버가 필요합니다.
10. 프롬프트가 표시되면, 스크립트가 SSH 구성을 강화하도록 할 것인지 여부를 선택합니다.

    VM을 사용하는 경우 계속 진행합니다. 자체 VM을 사용하고 SSH를 직접 구성하는 경우 현재 구성이 변경되지 않도록 이 단계를 건너뛸 수 있습니다.
11. SDC 또는 SEC 및 CLI 자체에 대한 자동 업데이트를 활성화하라는 메시지가 표시되면, 이 작업을 수행하여 버그 수정, 패치 및 새로운 기능을 최신 상태로 유지하는 것이 좋습니다. 정책으로 인해 자동 업데이트를 허용할 수 없는 경우에는 Secure Device Connector 업데이트를참조하십시오.

VM에 Secure Device Connector 구축

디바이스 자격 증명을 사용하여 Security Cloud Control를 디바이스에 연결하는 경우, 네트워크에서 SDC(Secure Device Connector)를 다운로드하고 구축하여 Security Cloud Control와 디바이스 간의 통신을 관리하는 것이 모범 사례입니다. 일반적으로 이러한 디바이스는 경계를 기반으로 하지 않으며 공용 IP 주소가 없거나 외부 인터페이스에 대한 개방형 포트가 있습니다. ASA(Adaptive Security Appliance), FDM 관리 디바이스, FMC(Firepower Management Center) 디바이스는 모두 디바이스 자격 증명을 사용하여 Security Cloud Control에 온보딩할 수 있습니다.

SDC는 매니지드 디바이스에서 실행해야 하는 명령과 매니지드 디바이스로 전송해야 하는 메시지에 대해 Security Cloud Control를 모니터링합니다. SDC는 Security Cloud Control를 대신하여 명령을 실행하고, 매니지드 디바이스를 대신하여 Security Cloud Control에 메시지를 전송하고, 매니지드 디바이스에서 Security Cloud Control로 응답을 반환합니다.

단일 SDC에서 관리할 수 있는 디바이스의 수는 해당 디바이스에 구현된 기능 및 해당 구성 파일의 크기에 따라 달라집니다. 그러나 구축을 계획할 때는 1개의 SDC가 약 500개의 디바이스를 지원할 것으로 예상합니다. 자세한 내용은 단일 Security Cloud Control 테넌트에서 여러 SDC 사용, 23 페이지를 참조하십시오.

이 절차에서는 자체 가상 머신 이미지를 사용하여 네트워크에 SDC를 설치하는 방법을 설명합니다.

참고 SDC를 설치하는 가장 쉽고 신뢰할 수 있는 방법은 Security Cloud Control의 SDC OVA 이미지를 다운로드하여 설치하는 것입니다.

시작하기 전에

- Security Cloud Control는 엄격한 인증서 확인이 필요하며 SDC와 인터넷 간의 웹/콘텐츠 프록시를 지원하지 않습니다.
- SDC가 Security Cloud Control와 통신하려면 TCP 포트 443에서 인터넷에 대한 전체 아웃바운드 액세스 권한이 있어야 합니다.
- SDC를 통해 Security Cloud Control에 접속하는 디바이스는 포트 443에서 SDC로부터의 인바운드 액세스를 허용해야 합니다.
- 네트워킹 지침은 Secure Device Connector를 사용하여 Security Cloud Control에 연결을 검토하십시오.
- vCenter 웹 클라이언트 또는 ESXi 웹 클라이언트와 함께 설치된 VMware ESXi 호스트

참고 vSphere 데스크톱 클라이언트를 사용한 설치는 지원되지 않습니다.

- ESXi 5.1 하이퍼바이저.
- Ubuntu 22.04 및 Ubuntu 24.04 운영 체제.
- SDC만 있는 VM의 시스템 요구 사항:
  - VMware ESXi 호스트에는 CPU 2개가 필요합니다.
  - VMware ESXi 호스트에는 최소 2GB의 메모리가 필요합니다.
  - VMware ESXi에는 프로비저닝 선택에 따라 가상 머신을 지원하기 위해 64GB의 디스크 공간이 필요합니다. 이 값은 필요에 따라 필요한 디스크 공간을 확장할 수 있도록 파티션과 함께 LVM(논리적 볼륨 관리)을 사용한다고 가정합니다.
- VM의 CPU와 메모리를 업데이트한 후 VM의 전원을 켜고 Secure Connector(보안 커넥터) 페이지에 SDC가 "Active(활성)" 상태로 표시되는지 확인합니다.
- 이 절차를 수행하는 사용자는 Linux 환경에서 작업하고 파일 편집을 위해 vi 시각적 편집기를 사용하는 데 익숙해야 합니다.
- CentOS 가상 머신에 온프레미스 SDC를 설치하는 경우 정기적으로 Yum 보안 패치를 설치하는 것이 좋습니다. Yum 구성에 따라 Yum 업데이트를 가져오려면 포트 80 및 443에서 아웃바운드 액세스를 열어야 할 수 있습니다. 또한 업데이트를 예약하려면 yum-cron 또는 crontab을 구성해야 합니다. 보안 운영 팀과 함께 Yum 업데이트를 받기 위해 변경해야 하는 보안 정책이 있는지 확인합니다.

참고 시작하기 전에: 절차의 명령을 복사하여 터미널 창에 붙여넣지 말고 대신 입력하십시오. 일부 명령에는 "n-대시"가 포함되며, 잘라내기 및 붙여넣기 프로세스에서 이러한 명령은 "m-대시"로 적용되어 명령이 실패할 수 있습니다.

프로시저

단계 **1** SDC를 생성할 Security Cloud Control 테넌트에 로그인합니다.

단계 **2** 왼쪽 창에서 **Administration(**관리**)** > **Integration(**통합**)** > **Secure Connectors(**보안 커넥터**)**를 클릭합니다.

단계 **3** **Services(**서비스**)** 페이지에서 **Secure Connectors(**보안 커넥터**)** 탭을 선택하고 파란색 플러스 버튼을 클릭한 후 **Secure Device Connector(**보안 디바이스 커넥터**)**를 선택합니다.

단계 **4** 창의 2단계에서 부트스트랩 데이터를 메모장에 복사합니다.

단계 **5** 최소 SDC에 할당된 다음 RAM 및 디스크 공간을 사용하여 **CentOS 7** 가상 머신을 설치합니다.

- 8GB RAM
- 10GB 디스크 공간

단계 **6** 설치가 완료되면 SDC의 IP 주소, 서브넷 마스크 및 게이트웨이를 지정하는 등의 기본 네트워킹을 구성합니다.

단계 **7** DNS(Domain Name Server) 서버를 구성합니다.

단계 **8** NTP(Network Time Protocol) 서버를 구성합니다.

단계 **9** SDC의 CLI와의 손쉬운 상호 작용을 위해 CentOS에 SSH 서버를 설치합니다.

단계 **10** Yum 업데이트를 실행한 후 **open-vm-tools**, **nettools** 및 **bind-utils** 패키지를 설치합니다.

```
[root@sdc-vm ~]# yum update -y
                [root@sdc-vm ~]# yum install -y open-vm-tools net-tools bind-utils
```

단계 **11** AWS CLI 패키지를 설치합니다. https://docs.aws.amazon.com/cli/latest/userguide/awscli-install-linux.html의 내용을 참조하십시오.

참고
--user 플래그를 사용하지 마십시오.

단계 **12** Docker CE 패키지를 설치합니다. https://docs.docker.com/install/linux/docker-ce/centos/#install-docker-ce의 내용을 참조하십시오.

참고
"저장소를 사용하여 설치" 방법을 사용합니다.

단계 **13** Docker 서비스를 시작하고 부팅 시 시작되도록 활성화합니다.

```
[root@sdc-vm ~]# systemctl start docker
[root@sdc-vm ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multiuser.target.wants/docker.service to
    /usr/lib/systemd/system/docker.service.
```

단계 **14** 두 사용자(cdo 및 sdc)를 생성합니다. 루트 사용자에 직접 액세스하지 않고 관리 작업을 수행하려면 cdo 사용자를 사용합니다. sdc 사용자는 SDC 도커 컨테이너를 실행하도록 지정됩니다.

```
[root@sdc-vm ~]# useradd cdo
[root@sdc-vm ~]# useradd sdc –d /usr/local/cdo
```

단계 **15** sdc 사용자의 비밀번호를 생성합니다.

```
[root@sdc-vm ~]# passwd cdo
Changing password for user cdo.
New password: <type password>
Retype new password: <type password>
passwd: all authentication tokens updated successfully.
```

단계 **16** sdc 사용자를 wheel 그룹에 추가하여 관리(sudo) 권한을 부여합니다.

```
[root@sdc-vm ~]# usermod -aG wheel cdo
[root@sdc-vm ~]#
```

단계 **17** Docker가 설치되면 사용자 그룹이 생성됩니다. CentOS/Docker 버전에 따라 "docker" 또는 "dockerroot"라고 부를 수 있습니다. /etc/group 파일을 확인하여 어떤 그룹이 생성되었는지 확인한 다음 sdc 사용자를 이 그룹에 추가합니다.

```
[root@sdc-vm ~]# grep docker /etc/group
 docker:x:993:
[root@sdc-vm ~]#
[root@sdc-vm ~]# usermod -aG docker sdc
[root@sdc-vm ~]#
```

단계 **18** /etc/docker/daemon.json 파일이 없는 경우 파일을 생성하고 아래 내용을 입력합니다. 생성되면 docker 데몬을 다시 시작합니다.

참고
"group" 키에 입력한 그룹 이름이 이전 단계의 /etc/group 파일에서 찾은 그룹과 일치하는지 확인합니다.

```
[root@sdc-vm ~]# cat /etc/docker/daemon.json
        {
           "live-restore": true,
           "group": "docker"
        }
        [root@sdc-vm ~]# systemctl restart docker
        [root@sdc-vm ~]#
```

단계 **19** 현재 vSphere 콘솔 세션을 사용 중인 경우 SSH로 전환하고 cdo 사용자로 로그인합니다. 로그인한 후에는 sdc 사용자로 변경합니다. 암호를 묻는 메시지가 표시되면 cdo 사용자의 암호를 입력합니다.

```
[CDO@sdc-vm ~]$ sudo su sdc
              [sudo] password for cdo: <type password for cdo user>
              [sdc@sdc-vm ~]$
```

단계 **20** 디렉터리를 **/usr/local/CDO**로 변경합니다.

단계 **21** bootstrapdata라는 새 파일을 생성하고 온프레미스 **Secure Device Connector** 구축 마법사의 2단계에서 가져온 부트스트랩 데이터를 이 파일에 붙여넣습니다. 파일을 **Save(**저장**)**합니다. vi 또는 nano를 사용하여 파일을 생성할 수 있습니다.

단계 **22** 부트스트랩 데이터는 base64로 인코딩됩니다. 이를 디코딩하고 `extractedbootstrapdata`라는 파일로 내보냅니다.

```
[sdc@sdc-vm ~]$ base64 -d /usr/local/ CDO/bootstrapdata > /usr/local/CDO/extractedbootstrapdata
                [sdc@sdc-vm ~]$
```

cat 명령을 실행하여 디코딩된 데이터를 확인합니다. 명령 및 디코딩된 데이터는 다음과 같이 표시됩니다.

```
[sdc@sdc-vm ~]$ cat /usr/local/ CDO/extractedbootstrapdata
                CDO_TOKEN="<token string>"
                CDO_DOMAIN="www.defenseorchestrator.com"
                CDO_TENANT="<tenant-name>"

CDO_BOOTSTRAP_URL="https://www.defenseorchestrator.com/sdc/bootstrap/tenant-name/<tenant-name-SDC>"
```

단계 **23** 다음 명령을 실행하여 디코딩된 부트스트랩 데이터의 섹션을 환경 변수로 내보냅니다.

```
[sdc@sdc-vm ~]$ sed -e 's/^/export /g' extractedbootstrapdata > sdcenv && source sdcenv
                [sdc@sdc-vm ~]$
```

단계 **24** Security Cloud Control에서 부트스트랩 번들을 다운로드합니다.

```
[sdc@sdc-vm ~]$ curl -O -H "Authorization: Bearer $CDO_TOKEN" "$CDO_BOOTSTRAP_URL"
                100 10314 100 10314 0 0 10656 0 --:--:-- --:--:-- --:--:-- 10654
                [sdc@sdc-vm ~]$ ls -l /usr/local/ CDO/*SDC
                -rw-rw-r--. 1 sdc sdc 10314 Jul 23 13:48 /usr/local/CDO/tenant-name-SDC
```

단계 **25** SDC tarball의 압축을 풀고 `bootstrap.sh` 파일을 실행하여 SDC 패키지를 설치합니다.

```
[sdc@sdc-vm ~]$ tar xzvf /usr/local/CDO/tenant-name-SDC
                <snipped – extracted files>
                [sdc@sdc-vm ~]$
                [sdc@sdc-vm ~]$ /usr/local/ CDO/bootstrap/bootstrap.sh
                [2018-07-23 13:54:02] environment properly configured
                download: s3://onprem-sdc/toolkit/prod/toolkit.tar to toolkit/toolkit.tar
                toolkit.sh
                common.sh
                [2018-07-23 13:54:04] startup new container
                Unable to find image 'ciscodefenseorchestrator/sdc_prod:latest' locally
                sha256:d98f17101db10e66db5b5d6afda1c95c29ea0004d9e4315508fd30579b275458: Pulling from
                ciscodefenseorchestrator/sdc_prod
                08d48e6f1cff: Pull complete
                ebbd10b629b1: Pull complete
                d14d580ef2ed: Pull complete
                45421d451ab8: Pull complete
                <snipped – downloads>
                no crontab for sdc
```

이제 SDC가 Security Cloud Control에서 "Active(활성)"로 표시됩니다.

다음에 수행할 작업

-

구축된 호스트에서 보안 디바이스 커넥터 부트스트랩

Procedure

단계 **1** 왼쪽 창에서 **Administration(관리)** > **Integration(통합)** > **Secure Connectors(보안 커넥터)**를 클릭합니다.

단계 **2** **Services(서비스)** 페이지에서 **Secure Connectors(보안 커넥터)** 탭을 선택하고, + 아이콘을 클릭한 후, **Secure Device Connector(보안 디바이스 커넥터)**를 선택합니다.

단계 **3** 창의 2단계에서 부트스트랩 데이터를 메모장에 복사합니다.

단계 **4** 관리자 사용자(일반적으로 `cdo`)와 선택한 비밀번호를 사용하여 VM에 SSH로 연결합니다.

단계 **5** 다음 명령을 입력하여 `sdc` 사용자로 전환합니다.

```
sudo su - sdc
```

단계 **6** 다음 명령을 사용하여 새 SDC를 부트스트랩합니다.

```
sdc bootstrap <paste-your-bootstrap-data-here>
```

단계 **7** 사용하려는 SDC의 버전을 선택합니다.

SDC 버전에는 세 가지 옵션이 있습니다.

- SDC 2024 - 대부분의 사용자가 실행하고자 하는 버전입니다.
- FIPS가 활성화된 SDC 2024 - FedRamp 컴플라이언스가 적용되는 경우 이 버전을 선택합니다.
- SDC 레거시 - 이 버전은 더 이상 기능 업데이트를 수신하지 않으므로 대신 SDC 2024를 실행하는 것이 좋습니다.

단계 **8** CLI는 컨테이너 이미지를 가져와 SDC를 시작합니다. 사용자 인터페이스에서 SDC가 활성화되고 정상 작동하는지 확인할 수 있으며, 호스트에서도 다음 명령을 실행하여 확인할 수 있습니다.

```
sdc show running
```

테넌트에 대한 SDC가 표시됩니다.

Terraform을 사용하여 vSphere에 Secure Device Connector 구축

시작하기 전에

이 절차에서는 vSphere용 Security Cloud Control SDC Terraform 모듈을 Security Cloud Control Terraform 제공자와 함께 사용하여 SDC를 vSphere에 구축하는 방법을 자세히 설명합니다. 이 작업 절차를 수행하기 전에 다음 사전 요구 사항을 검토하십시오.

- vSphere 데이터센터 버전 7 이상을 보유하고 있습니다.
- 데이터센터에 관리자 어카운트가 있으며, 다음 작업을 수행할 수 있는 권한이 있습니다.

- VM 생성
- 폴더 생성
- 콘텐츠 라이브러리 생성
- 콘텐츠 라이브러리에 파일 업로드

- Terraform 정보

프로시저

단계 **1** Security Cloud Control에서 API 전용 사용자를 생성하고 API 토큰을 복사합니다. API 전용 사용자를 생성하는 방법을 알아보려면 API 전용 사용자 생성을 참조하십시오.

단계 **2** Security Cloud Control Terraform 제공자의 지침에 따라 Terraform 저장소에서 Security Cloud ControlTerraform 제공자를 구성합니다.

예제:

```
terraform {
  required_providers {
    cdo = {
      source = "CiscoDevNet/cdo"
      version = "0.7.0"
    }
  }
}

provider "cdo" {
  base_url = “<the CDO URL you use to access CDO>”
  api_token = “<the API Token generated in step 1>”
}
```

단계 **3** Security Cloud Control Terraform 제공자를 사용하여 `cdo_sdc` 리소스를 생성하는 Terraform 코드를 작성합니다. 자세한 내용은 Security Cloud Control-sdc 리소스에 대한 Terraform 레지스트리를 참조하십시오.

예제:

```
Resource “cdo_sdc” “my-sdc” {
  name = “my-sdc-in-vsphere”
}
```

이 리소스의 `bootstrap_data` 속성은 Security Cloud Control 부트스트랩 데이터의 값으로 채워지며 다음 단계에서 `cdo_sdc` Terraform 모듈에 제공됩니다.

단계 **4** Security Cloud Control_sdc Terraform 모듈을 사용하여 vSphere에서 SDC를 생성하는 Terraform 코드를 작성합니다.

예제:

```
data "cdo_tenant" "current" {}

module "vsphere-cdo-sdc" {
  source               = "CiscoDevNet/cdo-sdc/vsphere"
  version              = "1.0.0"
  vsphere_username     = "<replace-with-username-with-admin-privileges>"
  vsphere_password     = "<super-secure-password>"
```

```
  vsphere_server       = "<replace-with-address-of-vsphere-server>"
  datacenter           = "<replace-with-datacenter-name>"
  resource_pool        = "<replace-with-resource-pool-name>"
  cdo_tenant_name      = data.cdo_tenant.current.human_readable_name
  datastore            = "<replace-with-name-of-datastore-to-deploy-vm-in>"
  network              = "<replace-with-name-of-network-to-deploy-vm-in>"
  host                 = "<replace-with-esxi-host-address>"
  allow_unverified_ssl = <boolean; set to true if your vsphere server does not have a valid SSL
certificate>
  ip_address          = "<sdc-vm-ip-address; must be in the subnet of the assigned network for the VM>"

  gateway              = "<replace-with-network-gateway-address>"
  cdo_user_password    = "<replace-with-password-for-cdo-user-in-sdc-vm>"
  root_user_password   = "<replace-with-password-for-root-user-in-sdc-vm>"
  cdo_bootstrap_data   = cdo_sdc.sdc-in-vsphere.bootstrap_data
}
```

생성된 VM에는 두 명의 사용자(`root` 사용자와 `cdo`라는 사용자)가 있으며 VM의 IP 주소는 정적으로 구성됩니다. `cdo_bootstrap_data` 속성에는 `cdo_sdc` 리소스가 생성될 때 생성된 `bootstrap_data` 속성의 값이 지정됩니다.

단계 **5** `terraform 계획` 및 `terraform 적용`을 사용하여 일반적으로 Terraform을 계획하고 적용합니다.

전체 예시는 CiscoDevNet의 Security Cloud Control 자동화 저장소를 참조하십시오.

SDC가 온보딩 상태를 유지한 경우 원격 콘솔을 사용하여 vSphere VM에 연결하고 `cdo` 사용자로 로그인한 후에 다음 명령을 실행합니다.

```
sdc host status
```

판독에 따라 수동으로 다음을 실행해야 할 수 있습니다.

```
sdc host configure
```

참고 Security Cloud Control Terraform 모듈은 Apache 2.0 라이선스에 따라 오픈 소스 소프트웨어로 게시됩니다. 지원이 필요한 경우 GitHub에서 문제를 제출할 수 있습니다.

Terraform 모듈을 사용하여 AWS VPC에 Secure Device Connector 구축

시작하기 전에

AWS VPC에서 SDC를 구축하기 전에 다음 사전 요건을 검토하십시오.

- Security Cloud Control는 엄격한 인증서 확인이 필요하며 SDC와 인터넷 간의 웹/콘텐츠 프록시 검사를 지원하지 않습니다. 프록시 서버를 사용하는 경우 SDC(Secure Device Connector)와 Security Cloud Control 간의 트래픽 검사를 비활성화합니다.
- Secure Device Connector를 사용하여 Security Cloud Control에 연결을 확인하여 커넥터에 대한 적절한 네트워크 액세스를 확인합니다.
- 이 경우 AWS 어카운트, 하나 이상의 서브넷이 있는 AWS VPC 및 AWS Route53 호스팅 영역이 필요합니다.

- Security Cloud Control 부트스트랩 데이터, AWS VPC ID 및 해당 서브넷 ID가 있는지 확인합니다.
- SDC를 구축하는 프라이빗 서브넷에 NAT 게이트웨이가 연결되어 있는지 확인합니다.
- 방화벽 관리 HTTP 인터페이스가 실행 중인 포트에서 NAT 게이트웨이에 연결된 탄력적 IP로 이동하는 트래픽을 엽니다.

프로시저

단계 **1** Terraform 파일에 다음 코드 줄을 추가합니다. 변수에 대한 입력을 수동으로 입력해야 합니다.

```
module "example-sdc" {
  source             = "git::https://github.com/cisco-lockhart/terraform-aws-cdo-sdc.git?ref=v0.0.1"
  env                = "example-env-ci"
  instance_name      = "example-instance-name"
  instance_size      = "r5a.xlarge"
  cdo_bootstrap_data = "<replace-with-cdo-bootstrap-data>"
  vpc_id             = <replace-with-vpc-id>
  subnet_id          = <replace-with-private-subnet-id>
}
```

입력 변수 및 설명 목록은 Secure Device Connector Terraform 모듈을 참조하십시오.

단계 **2** Terraform 코드에서 `instance_id`를 출력으로 등록합니다.

```
output "example_sdc_instance_id" {
  value = module. example-sdc.instance_id
}
```

`instance_id`를 사용하여 AWS Systems Manager 세션 관리자(SSM)를 통한 문제 해결을 위해 SDC 인스턴스에 연결할 수 있습니다. 사용 가능한 출력 목록은 Secure Device Connector Terraform 모듈의 출력을 참조하십시오.

다음에 수행할 작업

모든 SDC 문제 해결의 경우, AWS SSM을 사용하여 SDC 인스턴스에 연결해야 합니다. 인스턴스에 연결하는 방법에 대한 자세한 내용은 AWS Systems Manager 세션 관리자를 참조하십시오. SSH를 사용하여 SDC 인스턴스에 연결하는 포트는 보안상의 이유로 노출되지 않습니다.

참고 Security Cloud Control Terraform 모듈은 Apache 2.0 라이선스에 따라 오픈 소스 소프트웨어로 게시됩니다. 지원이 필요한 경우 GitHub에서 문제를 제출할 수 있습니다.

온프레미스 Secure Device Connector 및 Secure Event Connector를 CentOS 7 가상 머신에서 Ubuntu 가상 머신으로 마이그레이션

Security Cloud Control의 온프레미스 Secure Device Connector(SDC)는 지금까지 CentOS 7 가상 머신에 설치되었습니다. 이제 CentOS 7은 단종되었으며 Security Cloud Control에 의해 더 이상 사용되지 않으므로, 모든 SDC을 CentOS 7에서 Ubuntu 가상 머신으로 마이그레이션할 수 있도록 이 마이그레이션 프로세스를 생성했습니다.

마이그레이션하기 전에

- SDC는 TCP 포트 443에서 인터넷에 대한 전체 아웃바운드 액세스 권한을 가져야 합니다.
- SDC를 실행 중인 Ubuntu 가상 머신은 ASA 및 Cisco IOS 디바이스와 같이 통신하는 디바이스의 관리 인터페이스에 대한 네트워크 액세스 권한이 있어야 합니다.
- 이전 SDC VM의 IP 주소 또는 FQDN이 디바이스에 도달하기 위해 생성된 모든 네트워킹 규칙은 새 SDC VM의 IP 주소 또는 FQDN으로 다시 생성해야 합니다.
- 마이그레이션에는 10~15분이 소요됩니다. 이 시간 동안 디바이스는 계속해서 보안 정책을 적용하고 네트워크 트래픽을 라우팅하지만 SDC를 통해 통신할 수 없습니다.

사전 요건

Secure Device Connector 및 Secure Event Connector를 실행하기 위한 VM 구축에 나와 있는 지침에 따라 새 호스트를 구축합니다.

호스트 구성

SDC 및/또는 SEC를 마이그레이션하는 경우 이 절차를 따르십시오.

1. 여기에서 새 VM 이미지를 다운로드합니다.
2. CDO-SDC_VM.zip 파일의 압축을 풉니다. 다음과 비슷하게 이름이 지정된 3개의 VM 파일이 표시됩니다.
   - CDO-SDC-VM-708cd33-2024-05-30-2031-disk1.vmdk
   - CDO-SDC-VM-708cd33-2024-05-30-2031.mf
   - CDO-SDC-VM-708cd33-2024-05-30-2031.ovf
3. 방금 다운로드한 VM을 구축합니다.
4. 새 VM에 할당된 고정 IP 주소 또는 FQDN을 기록해 둡니다.
5. SSH를 사용하여 새 VM에 `cdo` 사용자로 로그인합니다.
6. 프롬프트에 다음 명령을 입력합니다.

   ```
   sudo sdc host configure
   ```

참고

- 마이그레이션 스크립트의 지시를 잘 따릅니다. 스크립트는 잘 문서화되어 있으며 각 단계를 설명하면서 마이그레이션 프로세스를 안내합니다.
- 마이그레이션 스크립트가 끝나면 SDC가 새 VM으로 마이그레이션되었음을 알리는 메시지가 표시됩니다. SDC는 마이그레이션 후에도 해당 이름을 그대로 유지합니다.

SDC 마이그레이션

절차:

1. SSH를 사용하여 이전(CentOS) SDC에 `cdo` 사용자로 로그인합니다.
2. 명령을 사용하여 CLI를 설치합니다.

```
curl -O
https://s3.us-west-2.amazonaws.com/download.defenseorchestrator.com/sdc-cli/sdc-cli-package-latest.tgz
 && tar -xvf sdc-cli-package-latest.tgz && chmod +x ./install.sh && ./install.sh
```

3. 다음 명령을 실행하고 프롬프트를 따릅니다.

```
sudo sdc migrate now
```

확인:

1. Security Cloud Control 테넌트에 로그인합니다.
2. 마이그레이션한 SDC를 선택하고 **Actions(**작업**)** 창에서 **Request Heartbeat(**하트비트 요청**)**을 클릭합니다.

참고 SDC가 **Active(**활성**)** 상태인지 확인합니다.

SEC 마이그레이션

절차:

1. SSH를 사용하여 이전(CentOS) SDC에 `cdo` 사용자로 로그인합니다.
2. 명령을 사용하여 CLI를 설치합니다.

```
curl -O
https://s3.us-west-2.amazonaws.com/download.defenseorchestrator.com/sdc-cli/sdc-cli-package-latest.tgz
 && tar -xvf sdc-cli-package-latest.tgz && chmod +x ./install.sh && ./install.sh
```

3. 다음 명령을 실행하고 프롬프트를 따릅니다.

```
sudo sdc eventing migrate
```

4. SEC의 새 IP 주소를 가리키도록 디바이스를 구성하거나 이전 호스트를 종료하고 새 호스트에 이전 호스트와 동일한 IP 주소를 할당하여 디바이스를 업데이트할 필요가 없도록 할 수 있습니다.

확인:

SEC 상태에 대한 자세한 내용은 상태 확인을 사용하여 Secure Event Connector의 상태 학습을 참조하십시오.

추가 지침

이전 **SDC**를 다시 시작하지 않음

마이그레이션이 완료된 후에는 원래의 가상 머신에서 이전 SDC를 다시 시작하지 마십시오.

실패한 마이그레이션 되돌리기

어떤 이유로든 마이그레이션이 실패하거나 결과가 예상과 달라 이전 SDC으로 되돌리려면 아래 지침을 따릅니다.

1. 새 VM에 로그인하고 **SDC** 사용자로 전환합니다.
2. 다음 명령을 사용하여 새 VM에서 SDC가 현재 실행 중이 아닌지 확인합니다.

   ```
   docker ps
   ```

3. SDC가 실행 중인 경우 명령을 실행합니다.

   ```
   sdc stop
   ```

4. `docker ps`를 다시 실행하여 SDC 실행이 중지되었는지 확인합니다.
5. 이전 VM에 로그인하고 다음 명령을 실행합니다.

   ```
   sdc migrate revert
   ```

6. 이전 SDC가 활성 상태이고 UI에 표시되면 새 VM으로 돌아가 명령을 실행합니다.

   ```
   sdc delete <your-tenant-name-here>
   ```

7. 브라우저를 완전히 새로 고치고 SDC를 클릭한 다음, 이전 호스트의 IP가 사이드바에 표시되는지 확인합니다.

   이 단계를 수행했음에도 새 IP가 계속 표시된다면, 새 상태 확인을 요청하고 브라우저를 새로 고친 후 다시 확인합니다.

8. SEC 마이그레이션을 되돌리려면 다음 명령을 실행합니다.

   ```
   sdc eventing revert
   ```

보안 디바이스 커넥터의 IP 주소 변경

시작하기 전에

- 이 작업을 수행하려면 관리자여야 합니다.
- SDC는 TCP 포트 443 또는 디바이스 관리를 위해 구성한 포트에서 인터넷에 대한 전체 아웃바운드 액세스 권한을 가져야 합니다.

참고 SDC의 IP 주소를 변경한 후 디바이스를 Security Cloud Control에 다시 등록할 필요가 없습니다.

프로시저

단계 **1** SDC에 대한 SSH 연결을 생성하거나 가상 머신의 콘솔을 열고 **CDO** 사용자로 로그인합니다.

단계 **2** IP 주소를 변경하기 전에 SDC VM의 네트워크 인터페이스 구성 정보를 보려면 다음 명령을 사용합니다.

```
[cdo@localhost ~]$ ip addr
```

단계 **3** 인터페이스의 IP 주소를 변경하려면 다음 명령을 사용하여 호스트 구성을 재시작합니다.

```
[cdo@localhost ~]$ sdc host configure network
```

단계 **4** 프롬프트에 비밀번호를 입력합니다.

단계 **5** 구성 스크립트는 이후 네트워크 설정에 대해 묻고, 새 IP를 포함한 새 구성 파일을 작성한 후 해당 구성을 적용합니다.

참고
이 시점에서 SSH 연결이 끊어집니다.

단계 **6** SDC에 할당한 새 IP 주소를 사용하여 SSH 연결을 만들고 로그인합니다.

단계 **7** SDC는 자동으로 시작되어야 하지만, 그렇지 않은 경우 다음 명령어를 실행합니다.

```
[cdo@localhost ~]$ sudo su - sdc
[cdo@localhost ~]$ sdc start
```

참고
VM의 콘솔에서 이 절차를 수행하는 경우 값이 올바른지 확인하면 연결 상태 테스트가 자동으로 실행되고 상태가 표시됩니다.

단계 **8** Security Cloud Control 사용자 인터페이스를 통해 SDC의 연결을 확인할 수도 있습니다. 이렇게 하려면 Security Cloud Control 애플리케이션을 열고 **Administration(**관리**)** > **Integration(**통합**)** > **Secure Connectors(**보안 커넥터**)** 페이지로 이동합니다.

단계 **9** 페이지를 한 번 새로 고치고 IP 주소를 변경한 보안 커넥터를 선택합니다.

단계 **10** **Actions(**작업**)** 창에서 **Request Heartbeat(**하트비트 요청**)**를 클릭합니다. 하트비트 요청 성공 메시지가 표시되고, 마지막 하트비트에 현재 날짜와 시간이 표시되어야 합니다.

Secure Device Connector 제거

Warning 이 절차에서는 SDC(Secure Device Connector)를 삭제합니다. 이는 되돌릴 수 없습니다. 이 작업을 수행한 후에는 새 SDC를 설치하고 디바이스를 다시 연결할 때까지 해당 SDC에 연결된 디바이스를 관리할 수 없습니다. 디바이스를 다시 연결하려면 다시 연결해야 하는 각 디바이스에 대한 관리자 자격 증명을 다시 입력해야 할 수 있습니다.

테넌트에서 SDC를 제거하려면 다음 절차를 수행합니다.

Procedure

단계 **1** 삭제할 SDC에 연결된 모든 디바이스를 제거합니다.

- **a.** SDC에서 사용하는 모든 디바이스를 식별하려면 동일한 SDC를 사용하여 Security Cloud Control에 연결하는 모든 디바이스 찾기를 참조하십시오.
- **b.** **Security Devices(**보안 디바이스**)** 페이지에서 식별한 모든 디바이스를 선택합니다.
- **c.** Device Actions(디바이스 작업) 창에서 **Remove(**제거**)**를 클릭하고 **OK(**확인**)**를 클릭하여 작업을 확인합니다.

단계 **2** 왼쪽 창에서 **Administration(**관리**)** > **Integration(**통합**)** > **Secure Connectors(**보안 커넥터**)**를 클릭합니다.

단계 **3** **Secure Connectors(**보안 커넥터**)** 탭이 선택된 **Services(**서비스**)** 페이지에서 파란색 플러스 버튼을 클릭한 후 **Secure Device Connector(**보안 디바이스 커넥터**)**를 선택합니다.

단계 **4** Secure Connector(보안 커넥터) 테이블에서 제거할 SDC를 선택합니다. 이제 디바이스 수가 0이어야 합니다.

단계 **5** 작업 창에서 **Remove(**제거**)**를 클릭합니다. 다음 경고가 표시됩니다.

Warning
<sdc_name>을 삭제하려고 합니다. SDC 삭제는 되돌릴 수 없습니다. SDC를 삭제하면 디바이스를 온보딩하거나 다시 온보딩하기 전에 새 SDC를 생성하고 온보딩해야 합니다.

현재 온보딩된 디바이스가 있으므로 SDC를 제거하려면 새 SDC를 설정한 후 해당 디바이스를 다시 연결하고 자격 증명을 다시 제공해야 합니다.

- 질문이나 우려 사항이 있는 경우 **Cancel(**취소**)**을 클릭하고 Security Cloud Control 지원에 문의하십시오.
- 계속하려면 <sdc_name>을 입력란에 입력하고 **OK(**확인**)**를 클릭합니다.

단계 **6** 확인 대화 상자에서 계속 진행하려면 경고 메시지에 나와 있는 SDC의 이름을 입력합니다.

단계 **7** **OK(**확인**)**를 클릭하여 SDC 제거를 확인합니다.

SDC 간에 ASA 이동

Security Cloud Control는 테넌트당 둘 이상의 SDC 사용을 지원합니다. 다음 절차를 사용하여 한 SDC에서 다른 SDC로 관리형 ASA를 이동할 수 있습니다.

프로시저

단계 1 왼쪽 창에서 보안 디바이스를 클릭합니다.

단계 2 **ASA** 탭을 클릭합니다.

단계 3 다른 SDC로 이동하려는 ASA를 선택합니다.

단계 4 **Device Actions(**디바이스 작업**)** 창에서 **Update Credentials(**자격 증명 업데이트**)**를 클릭합니다.

단계 5 Secure Device Connector 버튼을 클릭하고 디바이스를 이동하려는 SDC를 선택합니다.

단계 6 Security Cloud Control가 디바이스에 로그인하는 데 사용하는 관리자 사용자 이름과 암호를 입력하고 **Update(**업데이트**)**를 클릭합니다. 변경되지 않은 경우 관리자 사용자 이름과 암호는 ASA 온보딩에 사용한 것과 동일한 자격 증명입니다. 이러한 변경 사항을 디바이스에 구축할 필요는 없습니다.

참고
모든 ASA가 동일한 자격 증명을 사용하는 경우 한 SDC에서 다른 SDC로 ASA를 대량으로 이동할 수 있습니다. ASA에 다른 자격 증명이 있는 경우 한 번에 하나의 SDC에서 다른 하나로 이동해야 합니다.

보안 디바이스 커넥터 이름 변경

프로시저

단계 1 왼쪽 창에서 **Administration(**관리**)** > **Integration(**통합**)** > **Secure Connectors(**보안 커넥터**)**를 선택합니다.

단계 2 이름을 바꾸려는 SDC를 선택합니다.

단계 3 세부 정보 창에서 SDC 이름 옆에 있는 편집 아이콘 를 클릭합니다.

단계 4 SDC의 이름을 바꿉니다.

Security Devices(보안 디바이스**)** 창의 Secure Device Connector 필터를 포함하여 Security Cloud Control 인터페이스에 SDC 이름이 나타날 때마다 이 새 이름이 나타납니다.

Secure Device Connector 업데이트

이 절차를 문제 해결 툴로 사용합니다. 일반적으로 SDC는 자동으로 업데이트되므로 이 절차를 사용할 필요가 없습니다. 오류가 발생하면 수동 업데이트를 시작해야 할 수 있습니다.

Procedure

단계 **1** SDC에 연결합니다. SSH를 사용하여 연결하거나 하이퍼바이저에서 콘솔 보기를 사용할 수 있습니다.)

단계 **2** 관리자(일반적으로 **cdo**)로 SDC에 로그인합니다.

단계 **3** SDC 도커 컨테이너를 업데이트하려면 SDC 사용자로 전환합니다.

```
[cdo@sdc-vm ~]$ sudo su sdc
                [sudo] password for cdo: <type password for cdo user>
                [sdc@sdc-vm ~]$
```

단계 **4** SDC를 업그레이드합니다.

```
sdc upgrade
```

Note
SDC 가상 머신의 권장 업데이트 및 유지 관리

조직의 내부 IT 보안 및 패치 관리 정책에 따라 Ubuntu 리눅스에서 실행되는 SDC VM을 모니터링하고 업데이트를 적용해야 합니다. SDC VM이 네트워크 환경 내에서 보안을 유지하고 최적으로 작동하도록 관련 보안 패치를 정기적으로 검토하고 적용할 것을 적극 권장합니다.

단일 **Security Cloud Control** 테넌트에서 여러 **SDC** 사용

테넌트에 대해 둘 이상의 SDC를 구축하면 성능 저하 없이 더 많은 디바이스를 관리할 수 있습니다. 단일 SDC에서 관리할 수 있는 디바이스의 수는 해당 디바이스에 구현된 기능 및 해당 구성 파일의 크기에 따라 달라집니다.

테넌트에 SDC를 무제한으로 설치할 수 있습니다. 각 SDC는 하나의 네트워크 세그먼트를 관리할 수 있습니다. 이러한 SDC는 해당 네트워크 세그먼트의 디바이스를 동일한 Security Cloud Control 테넌트에 연결합니다. 여러 SDC가 없으면 서로 다른 Security Cloud Control 테넌트를 사용하여 격리된 네트워크 세그먼트에서 디바이스를 관리해야 합니다.

- 두 번째 또는 후속 SDC를 구축하는 절차는 첫 번째 SDC를 구축할 때와 동일합니다.
- 테넌트의 초기 SDC는 테넌트의 이름과 숫자 1을 통합합니다. 각 추가 SDC는 순서대로 번호가 매겨집니다.

동일한 **SDC**를 사용하는 **Security Cloud Control** 디바이스

동일한 SDC를 사용하여 Security Cloud Control에 연결하는 모든 디바이스를 식별하려면 다음 절차를 수행합니다.

Procedure

단계 **1** 왼쪽 창에서 보안 디바이스.

단계 **2** **Devices(**디바이스**)** 탭을 클릭하여 디바이스를 찾습니다.

단계 **3** 해당 디바이스 탭을 클릭합니다.

단계 **4** 필터 기준이 이미 지정된 경우 **Security Devices(**보안 디바이스**)** 페이지 상단에 있는 **clear(**지우기**)** 버튼을 클릭하여 Security Cloud Control로 관리하는 모든 디바이스 및 서비스를 표시합니다.

단계 **5** 필터 버튼 을 클릭하여 필터 메뉴를 확장합니다.

단계 **6** 필터의 **Secure Device Connector(**보안 디바이스 커넥터**)** 섹션에서 원하는 SDC의 이름을 확인합니다. **Security Devices(**보안 디바이스**)** 페이지에는 필터에서 선택한 SDC를 통해 Security Cloud Control에 연결하는 디바이스만 표시됩니다.

단계 **7** (선택 사항) 필터 메뉴에서 추가 필터를 선택하여 검색을 더욱 구체화합니다.

단계 **8** (선택 사항) 작업이 완료되면 **Security Devices(**보안 디바이스**)** 페이지 상단에 있는 **clear(**지우기**)** 버튼을 클릭하여 Security Cloud Control로 관리하는 모든 디바이스 및 서비스를 표시합니다.

SDC의 오픈소스 및 서드파티 라이선스

========== ==========

*** amqplib ***

amqplib copyright (c) 2013, 2014

Michael Bridgen <mikeb@squaremobius.net>

이 패키지 **"amqplib"**는 **MIT** 라이선스에 따라 라이선스가 부여됩니다**.** 사본은 이 디렉토리의 **LICENSE-MIT** 파일에서 찾거나 다음에서 다운로드할 수 있습니다**.**

http://opensource.org/licenses/MIT

==========

*** async ***

Copyright (c) 2010-2016 Caolan McMahon

본 소프트웨어와 관련 문서 파일**(**이하 "소프트웨어"**)**의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며**,** 이에 대한 비용은 부담하지 않습니다**.** 이는 소프트웨어의 사용**,** 복제**,** 수정**,** 통합**,** 게시**,** 구축**,** 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다**.** 또한**,** 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다**.**

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다**.**

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어 또는 소프트웨어의 사용 또는 기타 취급과 관련하여 계약 행위, 불법 행위 또는 기타 행위 등의 클레임, 손해, 또는 기타 책임에 대해 책임을 지지 않습니다.

==========

*** bluebird ***

MIT 라이선스**(MIT)**

Copyright (c) 2013-2015 Petka Antonov

본 소프트웨어와 관련 문서 파일(이하 "소프트웨어")의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어 또는 소프트웨어의 사용 또는 기타 취급과 관련하여 계약 행위, 불법 행위 또는 기타 행위 등의 클레임, 손해, 또는 기타 책임에 대해 책임을 지지 않습니다.

==========

*** cheerio ***

Copyright (c) 2012 Matt Mueller <mattmuelle@gmail.com>

본 소프트웨어와 관련 문서 파일(이하 "소프트웨어")의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어 또는 소프트웨어의 사용 또는 기타 취급과 관련하여 계약 행위, 불법 행위 또는 기타 행위 등의 클레임, 손해, 또는 기타 책임에 대해 책임을 지지 않습니다.

==========

*** command-line-args ***

MIT 라이선스**(MIT)**

Copyright (c) 2015 Lloyd Brookes <75pound@gmail.com>

본 소프트웨어와 관련 문서 파일(이하 "소프트웨어")의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어 또는 소프트웨어의 사용 또는 기타 취급과 관련하여 계약 행위, 불법 행위 또는 기타 행위 등의 클레임, 손해, 또는 기타 책임에 대해 책임을 지지 않습니다.

==========

*** ip ***

이 소프트웨어는 **MIT** 라이센스에 따라 라이센스가 부여됩니다.

Copyright Fedor Indutny, 2012.

본 소프트웨어와 관련 문서 파일(이하 "소프트웨어")의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어 또는 소프트웨어의 사용 또는 기타 취급과 관련하여 계약 행위, 불법 행위 또는 기타 행위 등의 클레임, 손해, 또는 기타 책임에 대해 책임을 지지 않습니다.

==========

*** json-buffer ***

Copyright (c) 2013 Dominic Tarr

본 소프트웨어와 관련 문서 파일(이하 "소프트웨어")의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다.어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어 또는 소프트웨어의 사용 또는 기타 거래와 관련하여 계약, 불법 행위 등으로 인해 발생한 어떠한 클레임, 손해, 또는 기타 책임에 대해서도 책임을 지지 않습니다.

==========

*** json-stable-stringify ***

이 소프트웨어는 **MIT** 라이선스에 따라 구축됩니다.

본 소프트웨어와 관련 문서 파일(이하 "소프트웨어")의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어 또는 소프트웨어의 사용 또는 기타 취급과 관련하여 계약 행위, 불법 행위 또는 기타 행위 등의 클레임, 손해, 또는 기타 책임에 대해 책임을 지지 않습니다.

==========

*** json-stringify-safe ***

ISC 라이선스

Copyright (c) Isaac Z. Schlueter and Contributors

위의 저작권 고지와 이 허가 고지가 모든 사본에 포함되어 있는 한, 본 소프트웨어를 비용 여부에 상관없이 어떤 목적으로든 사용, 복사, 수정 및/또는 구축할 수 있는 권한이 여기에 부여됩니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자는 본 소프트웨어의 사용 또는 성능과 관련하여 계약, 과실 또는 기타 불법 행위로 인해 발생한 어떠한 직접적, 간접적, 부수적, 특별, 징벌적 또는 결과적 손해(대체 상품 또는 서비스의 조달, 사용, 데이터 또는 이익의 손실, 사업 중단을 포함하되 이에 국한되지 않음)에 대해 책임을 지지 않습니다.

==========

*** lodash ***

Copyright JS Foundation and other contributors <https://js.foundation/>

Underscore.js, copyright Jeremy Ashkenas 기반

DocumentCloud 및 조사 리포터 및 편집자<http://underscorejs.org/>

이 소프트웨어는 많은 개인의 자발적 기여로 구성됩니다. 정확한 기여 내역은 다음에서 제공되는 https://github.com/lodash/lodash 개정 내역을 참조하십시오.

다음 라이센스는 다음을 제외하고 이 소프트웨어의 모든 부분에 적용됩니다.

아래에 문서화:

====

본 소프트웨어와 관련 문서 파일(이하 "소프트웨어")의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어의 사용 또는 성능과 관련하여 계약, 불법 행위 등으로 인해 발생한 어떠한 직접적, 간접적, 부수적, 특별, 우발적 또는 결과적 손해(대체 상품 또는 서비스의 조달, 사용, 데이터 또는 이익의 손실, 사업 중단을 포함하되 이에 국한되지 않음)에 대해 책임을 지지 않습니다.

====

샘플 코드에 대한 저작권 및 관련 권리는 **CC0**을 통해 포기됩니다. 샘플 코드는 문서의 산문 내에 표시되는 모든 소스 코드로 정의됩니다.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

node_modules 및 공급업체 디렉토리에 있는 파일은 자체 라이선스가 있는 이 소프트웨어에서 사용하는 외부에서 유지 관리되는 라이브러리입니다. 용어가 위의 용어와 다를 수 있으므로 해당 용어를 읽어보는 것이 좋습니다.

==========

*** log4js ***

Copyright 2015 Gareth Jones (다른 많은 사람들의 기여 포함**)**

Apache 라이선스 버전 **2.0("**라이선스**")**에 따라 라이선스가 부여됩니다. 라이센스를 준수하지 않는 한 이 파일을 사용할 수 없습니다. 다음에서 라이센스 사본을 얻을 수 있습니다.

http://www.apache.org/licenses/LICENSE-2.0

해당 법률에서 요구하거나 서면으로 동의하지 않는 한 라이선스에 따라 구축된 소프트웨어는 명시적이든 묵시적이든 어떠한 종류의 보증이나 조건 없이 "있는 그대로" 구축됩니다.라이선스에 따른 권한 및 제한 사항을 관리하는 특정 언어는 라이선스를 참조하십시오.

==========

*** mkdirp ***

Copyright 2010 James Halliday(mail@substack.net)

이 프로젝트는 **MIT/X11** 라이선스에 따라 구축되는 무료 소프트웨어입니다.

본 소프트웨어와 관련 문서 파일(이하 "소프트웨어")의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태("있는 그대로")로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어 또는 소프트웨어의 사용 또는 기타 취급과 관련하여 계약 행위, 불법 행위 또는 기타 행위 등의 클레임, 손해, 또는 기타 책임에 대해 책임을 지지 않습니다.

==========

*** node-forge ***

새로운 **BSD** 라이선스(**3**개 조항)

Copyright (c) 2010, Digital Bazar, Inc.

All rights reserved.

다음 조건을 충족하는 경우 수정 여부에 관계없이 소스 및 바이너리 형식으로 재구축 및 사용이 허용됩니다.

***** 소스 코드의 재구축는 위의 저작권 표시, 이 조건 목록 및 다음 면책 조항을 유지해야 합니다.

***** 바이너리 형식의 재구축는 구축와 함께 제공된 설명서 및/또는 기타 자료에 위의 저작권 고지, 이 조건 목록 및 다음 면책 조항을 복제해야 합니다.

*** Digital Bazaar, Inc.**의 이름이나 기여자의 이름은 특정 사전 서면 허가 없이 이 소프트웨어에서 파생된 제품을 보증하거나 홍보하는 데 사용할 수 없습니다.

이 소프트웨어는 저작권자와 기여자에 의해 "있는 그대로" 제공되며, 명시적 또는 묵시적으로 상품성 및 특정 목적에의 적합성을 포함한 모든 보증이 거부됩니다. 어떠한 경우에도 **DIGITAL BAZAAR**는 어떠한 직접적, 간접적, 부수적, 특별, 징벌적 또는 결과적 손해(대체 상품 또는 서비스의 조달, 사용, 데이터 또는 이익의 손실, 사업 중단을 포함하되 이에 국한되지 않음)에 대해 책임을 지지 않습니다. 이러한 손해가 계약, 엄격한 책임, 불법행위(과실 또는 기타 포함)에 대한 어떠한 책임 이론에 의거하여 발생하였는지 여부와 상관없이, 해당 손해의 가능성이 있음을 사전에 고지받았더라도, 이 소프트웨어의 사용으로 인해 발생하는 어떠한 방식의 책임도 지지 않습니다.

==========

*** request ***

Apache License

버전 **2.0, 2004**년 **1**월

http://www.apache.org/licenses/

사용, 복제 및 구축에 대한 약관

1. 정의.

"라이선스"는 이 문서의 섹션 **1**에서 **9**까지 정의된 사용, 복제 및 구축에 대한 조건을 의미합니다.

"라이선스 제공자"는 라이센스를 부여하는 저작권 소유자 또는 저작권 소유자가 승인한 법인을 의미합니다.

"법적 실체"는 행위 실체와 해당 실체를 통제하거나 통제받거나 공통 통제하에 있는 다른 모든 실체의 조합을 의미합니다. 이 정의의 목적상 "통제"는 **(i)** 계약 또는 기타 방식으로 해당 법인의 지시 또는 관리를 유발하는 직간접적인 권한 또는 **(ii) 50%** 또는 더 많은 발행주식, 또는 **(iii)** 해당 법인의 수익적 소유권.

"귀하"(또는 "귀하의")는 계약 또는 기타 방식으로 본 라이센스에 의해 부여된 권한을 행사하는 개인 또는 법인을 의미하거나 **(ii)** 다음 중 **50%** 이상을 소유합니다. 발행주식, 또는 **(iii)** 그러한 법인의 수익적 소유권.

"소스" 형식은 소프트웨어 소스 코드, 문서 소스 및 구성 파일을 포함하되 이에 국한되지 않는 수정을 위한 기본 형식을 의미합니다.

"개체" 형식은 컴파일된 개체 코드, 생성된 문서 및 다른 미디어 유형으로의 변환을 포함하되 이에 국한되지 않는 소스 형식의 기계적 변환 또는 변환으로 인해 발생하는 모든 형식을 의미합니다.

"저작물"은 저작물에 포함되거나 첨부된 저작권 표시(예는 아래 부록에 제공됨)에 표시된 대로 라이센스에 따라 사용 가능한 소스 또는 개체 형식의 저작물을 의미합니다.

"파생 저작물"은 저작물을 기반으로 하는(또는 저작물에서 파생된) 원본 또는 개체 형식의 모든 저작물을 의미하며 편집 편집, 주석, 정교화 또는 기타 편집이 전체적으로 저자의 원본 저작물을 나타냅니다. 본 라이선스에서 파생물은 저작물 및 그 파생물과 분리된 상태를 유지하거나 인터페이스에 단순히 링크(또는 이름으로 연결)되는 저작물은 포함하지 않습니다.

"기여"는 원본 저작물과 저작물에 포함하기 위해 저작권 소유자 또는 저작권 소유자를 대신하여 제출할 권한이 있는 개인 또는 법인이 라이선스 허가자에게 의도적으로 제출된 저작물 또는 파생물에 대한 편집 또는 추가를 포함한 저작물을 의미합니다. 이 정의에서 \"제출\"은 저작물에 대해 논의하고 개선하기 위해 라이선스 허가자 또는 그 대리인에게 전송되는 모든 형태의 전자, 구두 또는 서면 제출을 의미합니다. 여기에는 저작물을 논의하고 개선할 목적으로 라이선스 허가자 또는 그 대리인이 관리하는 전자 메일 목록, 소스 코드 제어 시스템, 문제 추적 시스템을 통한 커뮤니케이션이 포함되며 이에 국한되지 않습니다. 단, 저작권 소유자가 "기고문이 아님"을 명시적으로 표시하거나 서면으로 지정한 커뮤니케이션은 제외됩니다.

"기여자"는 라이선스 제공자 및 라이선스 제공자가 대신하여 기여물을 받아 작업물에 통합한 모든 개인 또는 법인을 의미합니다.

2. 저작권 라이선스 부여. 이 라이선스의 조건에 따라 각 기여자는 이로써 귀하에게 영구적이고 전 세계적이며 비독점적이고 무료이며 로열티가 없고 취소할 수 없는 저작권 라이선스를 부여합니다. 저작물 및 그러한 파생 저작물을 소스 또는 개체 형태로 재라이선스하고 구축합니다.

3. 특허 라이선스 부여. 본 라이선스의 약관에 따라 각 기여자는 귀하에게 저작물의 제작, 사용, 판매 제안, 판매, 가져오기, 기타 방식의 이전을 위한 영구적, 세계적, 비독점적, 요금 및 로열티 무료, 철회 불가능한(본 섹션에 명시된 경우 제외) 특허 라이선스를 부여합니다. 이러한 라이선스는 기여자가 부

여할 수 있는 특허권에만 적용되며, 이는 기여물 단독으로 또는 기여물이 제출된 저작물과의 조합으로 인해 침해될 수 있습니다. 귀하가 저작물 또는 저작물에 통합된 기여가 직접적 또는 기여적 특허 침해를 구성한다고 주장하는 어떤 법인에 대해 특허 소송(소송에서 교차 청구 또는 반소 포함)을 제기하는 경우, 본 라이선스에 따라 귀하에게 부여된 모든 특허 라이선스는 작업은 그러한 소송이 제기된 날짜에 종료됩니다.

4. 재구축. 귀하는 다음 조건을 충족하는 경우 편집 여부에 관계없이 모든 매체와 소스 또는 개체 형식으로 저작물 또는 그 파생 저작물의 사본을 재생산 및 구축할 수 있습니다.

귀하는 저작물 또는 파생 저작물의 다른 수령인에게 본 라이센스의 사본을 제공해야 합니다. 그리고

귀하는 수정된 파일에 귀하가 파일을 변경했음을 알리는 눈에 띄는 통지를 전달해야 합니다. 그리고

파생 저작물의 일부와 관련되지 않은 통지를 제외하고 저작물의 소스 형식에서 가져온 모든 저작권, 특허, 상표 및 귀속 고지를 귀하가 구축하는 파생 저작물의 소스 형식으로 유지해야 합니다. 그리고

저작물이 구축의 일부로 "**NOTICE**" 텍스트 파일을 포함하는 경우 귀하가 구축하는 모든 파생 저작물에는 그러한 **NOTICE** 파일에 포함된 귀속 고지의 읽을 수 있는 사본이 포함되어야 합니다. 다음 위치 중 적어도 하나에 있는 **2**차 저작물**: 2**차 저작물의 일부로 구축되는 **NOTICE** 텍스트 파일 내에서 파생 저작물과 함께 제공되는 경우 소스 양식 또는 문서 내에서; 또는 파생 저작물에 의해 생성된 디스플레이 내에서 그러한 제**3**자 고지가 일반적으로 표시되는 경우. **NOTICE** 파일의 내용은 정보 제공의 목적으로만 사용되며 이에 따라 라이선스가 편집되지 않습니다. 저작물의 **NOTICE** 텍스트와 함께 또는 그 부록으로 구축하는 파생물 내 속성 고지로 인해 라이선스가 변경되지 않는 경우, 이러한 추가적인 속성 고지를 추가할 수 있습니다. 귀하는 귀하의 편집 사항에 자신의 저작권 진술을 추가할 수 있으며 편집 사항의 사용, 복제 또는 구축 또는 그러한 파생 저작물 전체에 대한 추가 또는 다른 라이선스 조건을 제공할 수 있습니다. 그렇지 않으면 저작물은 본 라이선스에 명시된 조건을 준수합니다.

5. 기여물 제출. 귀하가 달리 명시하지 않는 한, 귀하가 저작물에 포함하기 위해 라이선스 허가자에게 의도적으로 제출한 모든 기여물은 추가 약관 없이 본 라이선스의 약관에 따라야 합니다. 위의 내용에도 불구하고 여기의 어떠한 내용도 그러한 기여와 관련하여 귀하가 라이선스 제공자와 체결한 별도의 라이선스 계약 조건을 대체하거나 편집하지 않습니다.

6. 상표. 본 라이선스는 저작물의 출처를 설명하고 **NOTICE** 파일의 내용을 재생산하는 데 합당하고 관례적인 사용에 필요한 경우를 제외하고 라이선스 제공자의 상표명, 상표, 서비스 마크 또는 제품 이름을 사용할 수 있는 권한을 부여하지 않습니다.

7. 보증의 면책조항. 해당 법률에 의해 요구되는 경우나 서면으로 합의된 경우를 제외하고, 라이선서는 작업물(및 각 기여자는 자신의 기여물)을 "있는 그대로" 제공하며, 명시적이거나 묵시적으로 어떠한 종류의 보증이나 조건도 포함하지 않습니다. 이는 제목, 비침해성, 상품성 또는 특정 목적에 대한 적합성에 대한 어떠한 보증이나 조건을 포함합니다. 귀하는 저작물 사용 또는 재구축의 적합성을 판단할 전적인 책임이 있으며 본 라이선스에 따른 귀하의 권한 행사와 관련된 모든 위험을 감수합니다.

8. 책임의 제한. 어떠한 경우에도, 과실(비롯하여 과실포함 책임), 계약 또는 기타 이론에 의한, 해당 법에 의해 요구되는 경우(예: 고의적이고 중대한 과실 행위)나 서면으로 합의된 경우를 제외하고, 어떤 기여자도 본 라이선스로 인한 손해에 대해 법적으로 책임을지지 않습니다. 이는 작업물의 사용 또는 사용 불능으로 인해 발생하는 어떠한 종류의 직접적, 간접적, 특수적, 우발적 또는 결과적 손해(예: 선의의 상실, 작업 중단, 컴퓨터 고장 또는 장애, 그 외 모든 상업적 손해나 손실을 포함)에 대해서도

책임을지지 않습니다. 심지어 해당 기여자에게 그러한 손해의 가능성이 알려져 있더라도 마찬가지입니다.

9. 보증 또는 추가 책임 수락. 저작물 또는 그 파생물을 재구축하는 경우 귀하는 지원, 보증, 면책 또는 본 라이선스와 일치하는 기타 책임 의무 및/또는 권리의 수락을 제공하고 요금을 청구할 수 있습니다. 그러나 이러한 의무를 수락할 때에는 다른 기여자를 대신하여 행동하지 않고 오로지 자신의 명의로, 자신의 책임하에만 행동해야 하며, 해당 보증이나 추가적인 책임을 수락함으로써 발생하는 어떠한 책임에 대해서도 각 기여자를 면책시키고 방어하며 보호하기 위해 보증을 제공하는 경우에만 그렇게 행동할 수 있습니다.

이용 약관의 끝

==========

*** rimraf ***

ISC 라이선스

Copyright (c) Isaac Z. Schlueter and Contributors

위의 저작권 고지와 이 허가 고지가 모든 사본에 포함되어 있는 한, 본 소프트웨어를 비용 여부에 상관없이 어떤 목적으로든 사용, 복사, 수정 및/또는 구축할 수 있는 권한이 여기에 부여됩니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태**("**있는 그대로**")**로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자는 본 소프트웨어의 사용 또는 성능과 관련하여 계약, 과실 또는 기타 불법 행위로 인해 발생한 어떠한 직접적, 간접적, 부수적, 특별, 징벌적 또는 결과적 손해(대체 상품 또는 서비스의 조달, 사용, 데이터 또는 이익의 손실, 사업 중단을 포함하되 이에 국한되지 않음)에 대해 책임을 지지 않습니다.

==========

*** uuid ***

Copyright (c) 2010-2012 Robert Kieffer

MIT 라이선스- http://opensource.org/licenses/mit-license.php

==========

*** validator ***

Copyright (c) 2016 Chris O'Hara <cohara87@gmail.com>

본 소프트웨어와 관련 문서 파일**(**이하 "소프트웨어"**)**의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며, 이에 대한 비용은 부담하지 않습니다. 이는 소프트웨어의 사용, 복제, 수정, 통합, 게시, 구축, 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다. 또한, 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다.

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다.

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태**("**있는 그대로**")**로 제공됩니다. 이에는 상업성, 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다. 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어

의 사용 또는 성능과 관련하여 계약, 불법 행위 등으로 인해 발생한 어떠한 직접적, 간접적, 부수적, 특별, 우발적 또는 결과적 손해(대체 상품 또는 서비스의 조달, 사용, 데이터 또는 이익의 손실, 사업 중단을 포함하되 이에 국한되지 않음)에 대해 책임을 지지 않습니다.

==========

*** when ***

오픈 소스 이니셔티브 **OSI - MIT** 라이선스

http://www.opensource.org/licenses/mit-license.php

Copyright (c) 2011 Brian Cavalier

본 소프트웨어와 관련 문서 파일**(**이하 "소프트웨어"**)**의 사본을 획득한 모든 사람은 본 소프트웨어를 제한 없이 다룰 수 있으며**,** 이에 대한 비용은 부담하지 않습니다**.** 이는 소프트웨어의 사용**,** 복제**,** 수정**,** 통합**,** 게시**,** 구축**,** 하위 라이선스 부여 및 소프트웨어의 사본을 판매하는 권리를 포함한 제한 없는 권리를 제공합니다**.** 또한**,** 소프트웨어를 제공받은 사람에게 이러한 권리를 부여함에 따라 다음의 조건에 따라야 합니다**.**

위의 저작권 표시와 이 허가 표시는 모든 소프트웨어의 사본 또는 실질적인 부분에 포함되어야 합니다**.**

본 소프트웨어는 명시적 또는 묵시적인 어떠한 종류의 보증도 포함되지 않은 상태**("**있는 그대로"**)**로 제공됩니다**.** 이에는 상업성**,** 특정 목적에의 적합성 및 타인의 권리를 침해하지 않음 등의 보증이 포함되지만 이에 한정되지는 않습니다**.** 어떠한 경우에도 저작자 또는 저작권 보유자는 본 소프트웨어의 사용 또는 성능과 관련하여 계약, 불법 행위 등으로 인해 발생한 어떠한 직접적, 간접적, 부수적, 특별, 우발적 또는 결과적 손해(대체 상품 또는 서비스의 조달, 사용, 데이터 또는 이익의 손실, 사업 중단을 포함하되 이에 국한되지 않음)에 대해 책임을 지지 않습니다.==========

번역에 관하여
Cisco는 일부 지역에서 본 콘텐츠의 현지 언어 번역을 제공할 수 있습니다. 이러한 번역은 정보 제공의 목적으로만 제공되며, 불일치가 있는 경우 본 콘텐츠의 영어 버전이 우선합니다.

번역에 관하여
Cisco는 일부 지역에서 본 콘텐츠의 현지 언어 번역을 제공할 수 있습니다. 이러한 번역은 정보 제공의 목적으로만 제공되며, 불일치가 있는 경우 본 콘텐츠의 영어 버전이 우선합니다.