

보안 구성 가이드

vRealize Automation 7.3

이 문서는 새 버전으로 교체되기 전까지 나열된 각 제품 버전 및 모든 이후 버전을 지원합니다. 이 문서에 대한 최신 버전을 확인하려면 <http://www.vmware.com/kr/support/pubs>를 참조하십시오.

KO-002535-01

vmware[®]

VMware 웹 사이트 (<http://www.vmware.com/kr/support/>) 에서 최신 기술 문서를 확인할 수 있습니다.
또한 VMware 웹 사이트에서 최신 제품 업데이트를 제공합니다.
이 문서에 대한 의견이 있으면 docfeedback@vmware.com으로 사용자 의견을 보내주십시오.

Copyright © 2015–2017 VMware, Inc. 판권 소유. [저작권 및 상표 정보](#).

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

목차

- 1 보안 구성 5
- 2 업데이트된 정보 7
- 3 vRealize Automation 보안 기준선 개요 9
- 4 설치 미디어의 무결성 확인 11
- 5 VMware 시스템 소프트웨어 인프라 강화 13
 - VMware vSphere[®] 환경 강화 13
 - Infrastructure as a Service 호스트 강화 13
 - Microsoft SQL Server 강화 14
 - Microsoft .NET 강화 14
 - Microsoft IIS(인터넷 정보 서비스) 강화 14
- 6 설치된 소프트웨어 검토 15
- 7 VMware 보안 권고 사항 및 패치 17
- 8 보안 구성 19
 - vRealize Automation 장치 보호 19
 - 루트 암호 변경 19
 - 루트 암호 해시 및 복잡성 확인 20
 - 루트 암호 기록 확인 20
 - 암호 만료 기한 관리 20
 - 보안 셸 및 관리 계정 관리 21
 - 가상 장치 관리 인터페이스 사용자 변경 25
 - 부팅 로더 인증 설정 26
 - NTP 구성 26
 - 전송 중인 vRealize Automation 장치 데이터에 대한 TLS 구성 27
 - 미사용 데이터의 보안 확인 35
 - vRealize Automation 애플리케이션 리소스 구성 36
 - 콘솔 프록시 구성 사용자 지정 38
 - 서버 응답 머릿글 구성 39
 - vRealize Automation 장치 세션 시간 초과 설정 41
 - 불필요한 소프트웨어 관리 41
 - IaaS(Infrastructure as a Service) 구성 요소 보안 44
 - Windows 시간 서비스를 사용하지 않도록 설정 45
 - 전송 중인 Infrastructure as a Service 데이터에 대한 TLS 구성 45
 - TLS 암호 그룹 구성 47

- 호스트 서버 보안 확인 47
- 애플리케이션 리소스 보호 47
- Infrastructure as a Service 호스트 시스템 보안 48

9 호스트 네트워크 보안 구성 51

- VMware 장치에 대한 네트워크 설정 구성 51
 - 사용자의 네트워크 인터페이스 제어 방지 51
 - TCP 백로그 대기열 크기 설정 51
 - 브로드캐스트 주소에 대한 ICMPv4 에코 거부 52
 - IPv4 프록시 ARP 사용 안 함 52
 - IPv4 ICMP Redirect 메시지 거부 53
 - IPv6 ICMP redirect 메시지 거부 53
 - IPv4 Martian 패킷 기록 54
 - IPv4 역방향 경로 필터링 사용 54
 - IPv4 전달 거부 55
 - IPv6 전달 거부 55
 - IPv4 TCP Syncookie 사용 56
 - IPv6 라우터 알람 거부 56
 - IPv6 라우터 요청 거부 57
 - 라우터 요청의 IPv6 라우터 기본 설정 거부 57
 - IPv6 라우터 접두사 거부 58
 - IPv6 라우터 알람 홉 제한 설정 거부 58
 - IPv6 라우터 알람 Autoconf 설정 거부 59
 - IPv6 인접 라우터 요청 거부 60
 - IPv6 최대 주소 제한 60
- Infrastructure as a Service 호스트에 대한 네트워크 설정 구성 61
 - 포트 및 프로토콜 구성 61
 - 사용자 필수 포트 61
 - 관리자 필수 포트 61

10 감사 및 로깅 65

- 색인 67

보안 구성

보안 구성은 사용자가 vRealize Automation 배포의 보안 구성을 평가하고 최적화하는 데 도움이 됩니다.

보안 구성에서는 일반적인 vRealize Automation 환경의 보안 배포에 대한 확인 및 구성에 대해 설명하며 사용자가 보안 구성과 관련하여 충분한 정보를 바탕으로 결정을 내릴 수 있도록 정보와 절차를 제공합니다.

대상 사용자

이 정보는 vRealize Automation 시스템 관리자 및 시스템 보안 유지 보수 및 구성을 담당하는 사용자를 대상으로 합니다.

VMware 기술 자료 용어집

VMware 기술 자료 사이트에서는 새로운 용어를 정리한 용어집을 제공하고 있습니다. VMware 기술 설명서에 사용된 용어에 대한 정의를 보려면 <http://www.vmware.com/support/pubs>를 참조하십시오.

업데이트된 정보

이 보안 구성 가이드는 제품의 각 릴리스에 따라 또는 필요할 때 업데이트됩니다.

이 표에는 보안 구성 가이드의 업데이트 기록이 나와 있습니다.

개정	설명
002535-01	“vRealize Automation 장치 세션 시간 초과 설정,” (41 페이지) 섹션을 업데이트했습니다.
002535-00	최초 릴리스

vRealize Automation 보안 기준선 개요

3

VMware는 vRealize Automation 시스템을 위한 보안 기준선을 확인 및 구성하는 데 도움이 되는 포괄적인 권장 사항을 제공합니다.

VMware에서 지정한 대로 적절한 도구 및 절차를 사용하여 vRealize Automation 시스템을 위한 강화된 보안 기준선 구성을 확인 및 유지 보수합니다. 일부 vRealize Automation 구성 요소는 강화된 상태 또는 부분적으로 강화된 상태로 설치되지만 VMware 보안 권장 사항, 회사 보안 정책 및 알려진 위협 요소 측면에서 각 구성 요소의 구성을 검토 및 확인해야 합니다.

vRealize Automation 보안 태세

vRealize Automation의 보안 태세는 시스템 및 네트워크 구성, 조직 보안 정책 및 보안 모범 사례를 기반으로 전체적 보안 환경을 가정합니다.

vRealize Automation 시스템의 강화를 확인 및 구성하는 경우 VMware 강화 권장 사항에서 제시한 대로 다음 각 영역을 고려합니다.

- 보안 배포
- 보안 구성
- 네트워크 보안

시스템의 보안 강화를 확인하려면 이러한 각 개념 영역과 관련된 VMware 권장 사항 및 로컬 보안 정책을 고려합니다.

시스템 구성 요소

vRealize Automation 시스템의 강화 및 보안 구성을 고려하는 경우 모든 구성 요소를 비롯해 시스템 기능을 지원하기 위한 해당 구성 요소의 상호 작동 방식을 이해해야 합니다.

보안 시스템을 계획 및 구현할 때 다음 구성 요소를 고려합니다.

- vRealize Automation 장치
- IaaS 구성 요소

vRealize Automation 및 구성 요소의 상호 작동 방식을 숙지하려면 VMware vRealize Automation 설명서 센터에서 기초 및 개념을 참조하십시오. 일반 vRealize Automation 배포 및 아키텍처에 대한 자세한 내용은 참조 아키텍처 항목을 참조하십시오.

설치 미디어의 무결성 확인

항상 사용자는 VMware 제품을 설치하기 전에 설치 미디어의 무결성을 확인해야 합니다.

ISO, 오프라인 번들 또는 패치를 다운로드한 후에는 항상 SHA1 해시를 확인하여 다운로드한 파일의 무결성과 신뢰성을 확인합니다. VMware에서 받은 물리적 미디어의 보안 봉인이 파손된 경우 소프트웨어를 VMware에 반환하여 교체 받으십시오.

미디어를 다운로드한 후에는 MD5/SHA1 합계 값을 사용하여 다운로드의 무결성을 확인합니다.

MD5/SHA1 해시 출력을 VMware 웹 사이트에 게시된 값과 비교합니다. SHA1 또는 MD5 해시가 일치해야 합니다.

설치 미디어의 무결성 확인에 대한 자세한 내용은 <http://kb.vmware.com/kb/1537>을 참조하십시오.

VMware 시스템 소프트웨어 인프라 강화

5

강화 프로세스의 일부로 VMware 시스템을 지원하는 배포된 소프트웨어 인프라를 평가하고 VMware 강화 지침을 준수하는지 확인합니다.

완벽하게 강화된 보안 환경을 만들 수 있도록 VMware 시스템을 강화하기 전에 지원 소프트웨어 인프라의 보안 결함을 검토하고 해결합니다. 고려해야 하는 소프트웨어 인프라 요소에는 운영 체제 구성 요소, 지원 소프트웨어 및 데이터베이스 소프트웨어가 포함됩니다. 제조업체의 권장 사항 및 기타 관련된 보안 프로토콜에 따라 이러한 구성 요소와 기타 구성 요소의 보안 문제를 해결합니다.

이 장에서는 다음 주제에 대해 설명합니다.

- “VMware vSphere® 환경 강화,” (13 페이지)
- “Infrastructure as a Service 호스트 강화,” (13 페이지)
- “Microsoft SQL Server 강화,” (14 페이지)
- “Microsoft .NET 강화,” (14 페이지)
- “Microsoft IIS(인터넷 정보 서비스) 강화,” (14 페이지)

VMware vSphere® 환경 강화

VMware vSphere® 환경을 평가하여 적절한 수준의 vSphere 강화 지침이 적용 및 유지되고 있는지 확인합니다.

강화에 대한 자세한 지침은 <http://www.vmware.com/security/hardening-guides.html> 페이지를 참조하십시오.

VMware vSphere® 인프라는 전체적으로 강화된 환경의 일부로 VMware에서 정의한 보안 지침을 준수해야 합니다.

Infrastructure as a Service 호스트 강화

Infrastructure as a Service Microsoft Windows 호스트 시스템이 VMware 지침에 따라 강화되었는지 확인합니다.

해당 Microsoft Windows 강화 및 보안 모범 사례 지침에 나와 있는 권장 사항을 검토하고, 사용 중인 Windows Server 호스트가 적절하게 강화되었는지 확인합니다. 강화 권장 사항을 따르지 않으면 Windows 릴리스에 포함된 안전하지 않은 구성 요소의 알려진 보안 취약성에 노출될 수 있습니다.

사용 중인 버전이 지원되는지 확인하려면 [vRealize Automation 지원 매트릭스](#)를 참조하십시오.

Microsoft 제품 강화 사례의 올바른 지침에 대해서는 Microsoft 벤더에게 문의하십시오.

Microsoft SQL Server 강화

Microsoft SQL Server 데이터베이스가 Microsoft 및 VMware의 보안 지침을 준수하는지 확인합니다.

해당하는 Microsoft SQL Server 강화 및 보안 모범 사례 지침에 제시된 권장 사항을 검토합니다. 설치되어 있는 Microsoft SQL Server 버전과 관련된 모든 Microsoft 보안 공지를 검토합니다. 강화 권장 사항을 따르지 않으면 Microsoft SQL Server 버전에 포함된 안전하지 않은 구성 요소의 알려진 보안 취약성에 노출될 수 있습니다.

사용 중인 Microsoft SQL Server 버전이 지원되는지 확인하려면 [vRealize Automation 지원 매트릭스](#)를 참조하십시오.

Microsoft 제품 강화 사례의 지침에 대해서는 Microsoft 벤더에게 문의하십시오.

Microsoft .NET 강화

Microsoft .NET은 전체적으로 강화된 환경의 일부로 Microsoft 및 VMware에서 제시하는 보안 지침을 준수해야 합니다.

해당하는 .NET 강화 및 보안 모범 사례 지침에 제시된 권장 사항을 검토합니다. 또한 현재 사용 중인 Microsoft SQL Server 버전과 관련된 모든 Microsoft 보안 공지를 검토합니다. 강화 권장 사항을 따르지 않으면 안전하지 않은 Microsoft.NET 구성 요소의 알려진 보안 취약성에 노출될 수 있습니다.

사용 중인 Microsoft .NET 버전이 지원되는지 확인하려면 [vRealize Automation 지원 매트릭스](#)를 참조하십시오.

Microsoft 제품 강화 사례의 지침에 대해서는 Microsoft 벤더에게 문의하십시오.

Microsoft IIS(인터넷 정보 서비스) 강화

Microsoft IIS(인터넷 정보 서비스)가 Microsoft 및 VMware 보안 지침을 모두 준수하는지 확인합니다.

해당하는 Microsoft IIS 강화 및 보안 모범 사례 지침에 제시된 권장 사항을 검토합니다. 또한 현재 사용 중인 IIS 버전과 관련된 모든 Microsoft 보안 공지를 검토합니다. 강화 권장 사항을 따르지 않으면 알려진 보안 취약성에 노출될 수 있습니다.

사용 중인 버전이 지원되는지 확인하려면 [vRealize Automation 지원 매트릭스](#)를 참조하십시오.

Microsoft 제품 강화 사례의 지침에 대해서는 Microsoft 벤더에게 문의하십시오.

설치된 소프트웨어 검토

타사 및 미사용 소프트웨어의 취약성이 인증되지 않은 시스템 액세스 및 가용성 중단 위험을 높이기 때문에 VMware 호스트 시스템에 설치된 모든 소프트웨어를 검토하고 해당 용도를 평가하는 것이 중요합니다.

VMware 호스트 시스템에 시스템의 보안 작업에 필요하지 않은 소프트웨어를 설치하지 마십시오. 미사용 또는 관련 없는 소프트웨어를 제거합니다.

인벤토리 설치 지원되지 않는 소프트웨어

설치된 제품의 VMware 배포 및 인벤토리에 액세스하여 지원되지 않는 관련 없는 소프트웨어가 설치되지 않았는지 확인합니다.

타사 제품을 위한 지원 정책에 대한 자세한 내용은 VMware 지원 문서 (<https://www.vmware.com/support/policies/thirdparty.html>)를 참조하십시오.

타사 소프트웨어 확인

VMware는 테스트 및 확인되지 않은 타사 소프트웨어의 설치를 지원하거나 권장하지 않습니다.

VMware 호스트 시스템에 설치된 보안되지 않거나 패치되지 않거나 인증되지 않은 타사 소프트웨어는 시스템을 인증되지 않은 액세스 및 가용성 중단 위험에 처하게 할 수 있습니다. 지원되지 않는 타사 소프트웨어를 사용해야 하는 경우 타사 벤더에 보안 구성 및 패치 요구 사항을 문의하십시오.

VMware 보안 권고 사항 및 패치

시스템에 대한 최대 보안을 유지 보수하려면 VMware 보안 권고 사항을 따르고 모든 관련 패치를 적용합니다.

VMware는 제품에 대한 보안 권고 사항을 릴리스합니다. 이러한 권고 사항을 모니터링하여 제품이 알려진 위협 요소로부터 보호되고 있는지 확인합니다.

vRealize Automation 설치, 패치 및 업그레이드 기록을 평가하고 릴리스된 VMware 보안 권고 사항을 따르고 적용하는지 확인합니다.

현재 VMware 보안 권고 사항에 대한 자세한 내용은

<http://www.vmware.com/security/advisories/>를 참조하십시오.

보안 구성

시스템 구성에 맞게 vRealize Automation 가상 장치에 대한 보안 설정 및 IaaS(Infrastructure as a Service) 구성 요소를 확인 및 업데이트합니다. 또한 기타 구성 요소 및 애플리케이션의 구성을 확인 및 업데이트합니다.

vRealize Automation 설치 보안 구성에는 각 구성 요소의 개별 구성 및 상호 작동 시 구성이 포함됩니다. 알맞은 보안 기준선을 달성하려면 모든 시스템 구성 요소의 상호 협동 구성을 고려합니다.

이 장에서는 다음 주제에 대해 설명합니다.

- “vRealize Automation 장치 보호,” (19 페이지)
- “IaaS(Infrastructure as a Service) 구성 요소 보안,” (44 페이지)

vRealize Automation 장치 보호

시스템 구성의 필요에 따라 vRealize Automation 장치에 대한 보안 설정을 확인 및 업데이트합니다.

가상 장치 및 해당 호스트 운영 체제에 대한 보안 설정을 구성하십시오. 또한 다른 관련 구성 요소 및 애플리케이션에 대한 구성을 설정하거나 확인하십시오. 경우에 따라 기존 설정을 확인해야 하고 다른 경우에는 적절한 구성을 사용하기 위해 설정을 변경하거나 추가해야 합니다.

루트 암호 변경

적용 가능한 보안 요구 사항을 충족하도록 vRealize Automation 장치의 루트 암호를 변경할 수 있습니다.

vRealize Automation 장치의 루트 암호는 가상 장치 관리 인터페이스를 사용하여 변경합니다. 루트 암호가 조직의 회사 암호 복잡성 요구 사항을 충족하는지 확인하십시오.

프로시저

- 1 vRealize Automation 장치의 가상 장치 관리 인터페이스를 엽니다.
https://vRealizeAppliance-url:5480
- 2 가상 장치 관리 인터페이스에서 **관리** 탭을 선택합니다.
- 3 **관리** 하위 메뉴를 선택합니다.
- 4 **현재 관리자 암호** 텍스트 상자에 기존 암호를 입력합니다.
- 5 **새 관리자 암호** 텍스트 상자에 새 암호를 입력합니다.
- 6 **새 관리자 암호 다시 입력** 텍스트 상자에 새 암호를 입력합니다.
- 7 **설정 저장**을 클릭하여 변경 내용을 저장합니다.

루트 암호 해시 및 복잡성 확인

루트 암호가 조직의 회사 암호 복잡성 요구 사항을 충족하는지 확인하십시오.

루트 암호 복잡성 검증은 루트 사용자가 사용자 계정에 적용되는 pam_cracklib 모듈 암호 복잡성 검사를 우회할 때 필요합니다.

계정 암호는 sha512 해시를 나타내는 \$6\$로 시작해야 합니다. 이는 모든 강화된 장치에 대한 표준 해시입니다.

프로시저

- 1 루트 암호의 해시를 확인하려면 루트로 로그인하고 # more /etc/shadow 명령을 실행합니다.
해시 정보가 표시됩니다.

그림 8-1. 암호 해시 결과

```
vcac148-084-111:~ # more /etc/shadow
bin:!:16332:0:60:7:::
daemon:!:16332:0:60:7:::
haldaemon:!:16332:0:60:7:::
mail:!:15870:::60:::
man:!:16332:0:60:7:::
messagebus:!:16332:0:60:7:::
nobody:!:15870:::60:::
ntp:!:16332:0:60:7:::
polkituser:!:16332:0:60:7:::
postfix:!:16332:0:60:7:::
root:$6$PHxGPY5A$ba8KezK4SS44UEHPfAtgs
P/:16346:0:365:7:::
```

- 2 루트 암호에 sha512 해시가 포함되지 않은 경우 passwd 명령을 실행하여 변경합니다.

모든 강화된 장치는 /etc/pam.d/common-password 파일에서 pw_history 모듈에 대해 enforce_for_root를 사용하도록 설정합니다. 시스템은 기본적으로 마지막 5개 암호를 기억합니다. 각 사용자의 이전 암호는 /etc/security/passwd 파일에 저장됩니다.

루트 암호 기록 확인

암호 기록이 루트 계정에 대해 적용되는지 확인합니다.

모든 강화된 장치는 /etc/pam.d/common-password 파일에서 pw_history 모듈에 대해 enforce_for_root를 사용하도록 설정합니다. 시스템은 기본적으로 마지막 5개 암호를 기억합니다. 각 사용자의 이전 암호는 /etc/security/passwd 파일에 저장됩니다.

프로시저

- 1 다음 명령을 실행합니다.
`cat /etc/pam.d/common-password-vmware.local | grep pam_pwhistory.so`
- 2 enforce_for_root가 반환된 결과에 표시되는지 확인합니다.
`password required pam_pwhistory.so enforce_for_root remember=5 retry=3`

암호 만료 기한 관리

조직의 보안 정책에 따라 모든 계정 암호 만료 기한을 구성합니다.

기본적으로 모든 강화된 VMware 가상 장치 계정의 암호 만료 기한은 60일입니다. 대부분의 강화된 장치에서 루트 계정의 암호 만료 기한은 365일로 설정됩니다. 모범 사례로 모든 계정의 만료 기한이 보안 및 운영 요구 사항 표준 둘 모두를 준수하는지 확인하는 것이 좋습니다.

루트 암호는 만료되면 복구할 수 없습니다. 관리 및 루트 암호가 만료되지 않도록 사이트별 정책을 구현해야 합니다.

프로시저

- 1 가상 장치 시스템에 루트로 로그인한 후 다음 명령을 실행하여 모든 계정의 암호 만료 기한을 확인합니다.

```
# cat /etc/shadow
```

암호 만료 기한은 새도 파일의 다섯 번째 필드입니다(필드는 콜론으로 구분됨). 루트 만료 기한은 일 단위로 설정됩니다.

그림 8-2. 암호 만료 기한 필드

```
vcac148-084-111:~ # more /etc/shadow
bin:!:16332:0:60:7:::
daemon:!:16332:0:60:7:::
haldaemon:!:16332:0:60:7:::
mail:!:15870:::60:::
man:!:16332:0:60:7:::
messagebus:!:16332:0:60:7:::
nobody:!:15870:::60:::
ntp:!:16332:0:60:7:::
polkituser:!:16332:0:60:7:::
postfix:!:16332:0:60:7:::
root:$6$PHxGPY5A$ba8KezK4SS44UEHPfAtgsB6iy1nJYbBh
zRbYR1mvJiX.pJpub0AEpP/:16346:0:365:7:::
sshd:!:16332:0:60:7:::
```

- 2 루트 계정의 만료 기한을 수정하려면 다음 형식으로 명령을 실행합니다.

```
# passwd -x 365 root
```

이 명령에서 365는 암호 만료까지의 일 수를 지정합니다. 같은 명령을 사용하여 원하는 사용자를 수정할 수 있으며, 이 경우 'root'를 특정 계정으로 대체하고 조직의 만료 기한 표준에 맞게 일 수를 바꾸면 됩니다.

보안 셸 및 관리 계정 관리

원격 연결의 경우 강화된 모든 장치에는 SSH(보안 셸) 프로토콜이 포함됩니다. 시스템 보안을 유지하려면 SSH를 필요한 경우에만 사용하고 적절하게 관리해야 합니다.

SSH는 VMware 가상 장치에 대한 원격 연결을 지원하는 대화형 명령줄 환경입니다. 기본적으로 SSH 액세스를 위해서는 높은 권한을 가진 사용자 계정 자격 증명이 필요합니다. 루트 사용자 SSH 작업은 일반적으로 RBAC(역할 기반 액세스 제어) 및 가상 장치의 감사 제어를 생략합니다.

모범 사례로 운영 환경에서는 SSH를 사용 안 함으로 설정하고, 다른 방법으로는 해결할 수 없는 문제를 해결할 때만 활성화합니다. 특정 용도로 필요한 경우에만 조직의 보안 정책에 따라 SSH를 사용 가능한 상태로 둡니다. vRealize Automation 장치에서 SSH는 기본적으로 사용하지 않도록 설정됩니다. vSphere 구성에 따라 OVF(Open Virtualization Format) 템플릿 배포 시 SSH를 사용 또는 사용 안 함으로 설정할 수 있습니다.

시스템에서 SSH가 사용하도록 설정되었는지를 테스트하는 가장 간단한 방법은 SSH를 사용하여 연결을 열어보는 것입니다. 연결이 열리고 자격 증명 요청 메시지가 표시되면 SSH가 사용하도록 설정되고 연결에 사용 가능한 상태입니다.

보안 셸 루트 사용자 계정

VMware 장치에는 미리 구성된 사용자 계정이 포함되어 있지 않기 때문에 기본적으로 루트 계정이 SSH를 사용하여 직접 로그인할 수 있습니다. SSH는 루트 권한으로 가능한 빨리 사용 안 함으로 설정해야 합니다.

거부 없음에 대한 규정 표준을 준수하기 위해 모든 강화된 장치에서 SSH 서버에는 AllowGroups 휠 항목이 미리 구성되어 SSH 액세스가 보조 그룹 휠로 제한됩니다. 책임 분담을 위해 /etc/ssh/sshd_config 파일에서 sshd 같이 다른 그룹을 사용하도록 AllowGroups 휠 항목을 수정할 수 있습니다.

휠 그룹은 슈퍼유저 액세스를 위해 pam_wheel 모듈을 사용하도록 설정되므로 휠 그룹의 구성원은 루트로 su할 수 있으며, 이때 루트 암호가 필요합니다. 그룹을 분리하면 사용자가 SSH를 통해 장치에 연결할 수 있지만 루트로 su할 수 없습니다. 장치 기능이 올바르게 작동하도록 AllowGroups 필드의 다른 항목은 제거하거나 수정하지 마십시오. 값을 변경한 후에는 # service sshd restart 명령을 실행하여 SSH 대본을 다시 시작해야 합니다.

vRealize Automation 장치에서 보안 셸 사용 또는 사용 안 함

문제 해결을 위해서만 vRealize Automation 장치에 SSH(보안 셸)를 사용하도록 설정합니다. 일반적인 운영 작업 중에는 이러한 구성 요소에서 SSH를 사용 안 함으로 설정합니다.

가상 장치 관리 콘솔을 사용하여 vRealize Automation 장치에서 SSH를 사용 또는 사용하지 않도록 설정할 수 있습니다.

프로시저

- 1 vRealize Automation 장치의 VAMI(가상 장치 관리 인터페이스)로 이동합니다.
: <https://vRealizeAppliance url:5480>
- 2 **관리** 탭을 클릭합니다.
- 3 **관리** 하위 메뉴를 클릭합니다.
- 4 **SSH 서비스 사용** 확인란을 선택하여 SSH를 사용하도록 설정하거나 확인란을 선택 취소하여 SSH를 사용하지 않도록 설정합니다.
- 5 **설정 저장**을 클릭하여 변경 내용을 저장합니다.

보안 셸용 로컬 관리자 계정 생성

보안 모범 사례로 가상 장치 호스트 시스템에서 보안 셸(SSH)용 로컬 관리자 계정을 생성하고 구성합니다. 또한 적절한 계정을 생성한 후 루트 SSH 액세스를 제거하십시오.

SSH용 로컬 관리자 계정 또는 보조 wheel 그룹의 구성원을 생성하거나 두 가지 모두를 생성합니다. 직접적인 루트 액세스를 사용하지 않도록 설정하기 전에 인증된 관리자가 AllowGroups를 사용하여 SSH에 액세스할 수 있는지, wheel 그룹을 사용하여 루트로 su할 수 있는지 테스트합니다.

프로시저

- 1 가상 장치에 루트로 로그인하고 적절한 사용자 이름을 사용하여 다음 명령을 실행합니다.

```
# useradd -g users <username> -G wheel -m -d /home/<username>
# passwd <username>
```

Wheel은 ssh 액세스를 위해 AllowGroups에 지정된 그룹입니다. 여러 개의 보조 그룹을 추가하려면 -G wheel,sshd를 사용합니다.

- 2 해당 사용자로 전환하고 새 암호를 제공하여 암호 복잡성 확인을 적용합니다.

```
# su -<username>
# <username>@hostname:~>passwd
```

암호 복잡성이 충족되면 암호가 업데이트됩니다. 암호 복잡성이 충족되지 않으면 암호가 원래 암호로 되돌아가고 암호 명령을 다시 실행해야 합니다.

- SSH에 직접 로그인을 제거하려면 `/etc/ssh/sshd_config` 파일을 수정하여 `(#)PermitRootLogin yes`를 `PermitRootLogin no`로 교체하십시오.

또는 **관리 탭의 관리자 SSH 로그인 사용** 확인란을 선택하거나 선택 취소하여 가상 장치 관리 인터페이스(VAMI)에서 SSH를 사용하거나 사용하지 않도록 설정할 수 있습니다.

후속 작업

루트로 직접 로그인을 사용하지 않도록 설정합니다. 기본적으로 강화된 장치는 콘솔을 통해 루트로 직접 로그인을 허용합니다. 부인 방지에 대한 관리 계정을 생성하고 `su-root wheel` 액세스에 대해 해당 계정을 테스트한 후 `/etc/security` 파일을 루트로 편집하고 `tty1` 항목을 `console`로 교체하여 직접 루트 로그인을 사용하지 않도록 설정합니다.

- 텍스트 편집기에서 `/etc/securetty` 파일을 엽니다.
- `tty1`을 찾아서 `console`로 교체합니다.
- 파일을 저장하고 닫습니다.

보안 셸 액세스 제한

시스템 강화 프로세스의 일부로 모든 VMware 가상 장치 호스트 시스템에서 적절히 `tcp_wrappers` 패키지를 구성하여 SSH(보안 셸) 액세스를 제한합니다. 또한 이러한 장치에 대한 필수 SSH 키 파일 사용 권한을 유지 보수합니다.

모든 VMware 가상 장치에는 TCP 지원 데몬이 `libwrap` 데몬에 액세스할 수 있는 네트워크 서브넷을 제어하도록 허용하는 `tcp_wrappers` 패키지가 포함됩니다. 기본적으로 `/etc/hosts.allow` 파일에는 보안 셸에 대한 모든 액세스를 허용하는 일반 항목 `Sshd: ALL : ALLOW`가 포함됩니다. 이 액세스를 조직에 맞게 제한합니다.

프로시저

- 텍스트 편집기에서 가상 장치 호스트 시스템의 `/etc/hosts.allow` 파일을 엽니다.
- 보안 작업을 위한 관리 네트워크 서브넷과 로컬 호스트 항목만 포함하도록 운영 환경의 일반 항목을 변경합니다.

```
sshd:127.0.0.1 : ALLOW
sshd: [::1] : ALLOW
sshd: 10.0.0.0 :ALLOW
```

이 예에서는 클라이언트가 10.0.0.0 서브넷에서 수행하는 연결 및 모든 로컬 호스트 연결이 허용됩니다.

- 모든 적절한 시스템 ID(예: 호스트 이름, IP 주소, FQDN(정규화된 도메인 이름) 및 루프백)를 추가합니다.
- 파일을 저장하고 닫습니다.

보안 셸 서버 구성 강화

가능한 경우 모든 VMware 장치에는 기본 강화된 구성이 있습니다. 구성 파일에서 글로벌 옵션 섹션의 서버 및 클라이언트 서비스 설정을 검토하여 현재 구성이 적절하게 강화되었는지 확인할 수 있습니다.

가능한 경우 SSH 서버를 관리 서브넷에서만 사용하도록 `/etc/hosts.allow` 파일에서 제한합니다.

프로시저

- 1 VMware 장치에서 `/etc/ssh/sshd_config` 서버 구성 파일을 열고 설정이 올바른지 확인합니다.

설정	상태
서버 대문 프로토콜	Protocol 2
CBC 암호	aes256-ctr 및 aes128-ctr
TCP 포워딩	AllowTCPForwarding no
서버 게이트웨이 포트	Gateway Ports no
X11 포워딩	X11Forwarding no
SSH 서비스	AllowGroups 필드를 사용하여 액세스가 허용된 그룹을 지정합니다. 이 그룹에 적절한 구성원을 추가합니다.
GSSAPI 인증	GSSAPIAuthentication no(사용하지 않는 경우)
Keberos 인증	KeberosAuthentication no(사용하지 않는 경우)
로컬 변수(AcceptEnv 글로벌 옵션)	disabled by commenting out 또는 enabled for LC_* or LANG variables로 설정
터널 구성	PermitTunnel no
네트워크 세션	MaxSessions 1
사용자 동시 연결	루트 및 기타 모든 사용자에게 대해 1로 설정. <code>/etc/security/limits.conf</code> 파일에도 동일한 설정이 구성되어 있어야 합니다.
엄격한 모드 확인	Strict Modes yes
권한 구분	UsePrivilegeSeparation yes
rhosts RSA 인증	RhostsESAAuthentication no
압축	Compression delayed 또는 Compression no
메시지 인증 코드	MACs hmac-sha1
사용자 액세스 제한	PermitUserEnvironment no

- 2 변경 사항을 저장하고 파일을 닫습니다.

보안 셸 클라이언트 구성 강화

시스템 강화 프로세스의 일부로 SSH 클라이언트의 강화를 확인합니다. 이를 위해 가상 장치 호스트 시스템에서 SSH 클라이언트 구성 파일이 VMware 지침에 따라 구성되었는지 검토합니다.

프로시저

- 1 SSH 클라이언트 구성 파일 `/etc/ssh/ssh_config`를 열고 글로벌 옵션 섹션의 설정이 올바른지 확인합니다.

설정	상태
클라이언트 프로토콜	Protocol 2
클라이언트 게이트웨이 포트	Gateway Ports no
GSSAPI 인증	GSSAPIAuthentication no
로컬 변수(SendEnv 글로벌 옵션)	LC_* 또는 LANG 변수만 제공
CBC 암호	aes256-ctr 및 aes128-ctr만
메시지 인증 코드	MACs hmac-sha1 항목에만 사용됨

- 2 변경 사항을 저장하고 파일을 닫습니다.

보안 셸 키 파일 사용 권한 확인

악의적 공격의 가능성을 최소화하려면 가상 장치 호스트 시스템에 대한 중요 SSH 키 파일 사용 권한을 유지 보수합니다.

SSH 구성을 지정하거나 업데이트한 후 항상 다음 SSH 키 파일 사용 권한이 변경되지 않도록 확인합니다.

- `/etc/ssh/*key.pub`에 있는 공용 호스트 키 파일은 루트 사용자가 소유하며 `0644(-rw-r--r--)`로 설정된 사용 권한이 있습니다.
- `/etc/ssh/*key`에 있는 개인 호스트 키 파일은 루트 사용자가 소유하며 `0600(-rw-----)`으로 설정된 사용 권한이 있습니다.

SSH 키 파일 사용 권한 확인

SSH 사용 권한이 공용 키 파일과 개인 키 파일 둘 모두에 적용되는지 확인합니다.

프로시저

- 1 다음 명령을 실행하여 SSH 공용 키 파일을 확인합니다. `ls -l /etc/ssh/*key.pub`
- 2 소유자가 루트이고, 그룹 소유자가 루트이며 파일의 사용 권한이 `0644(-rw-r--r--)`로 설정되었는지 확인합니다.
- 3 다음 명령을 실행하여 모든 문제를 해결합니다.


```
chown root /etc/ssh/*key.pub
chgrp root /etc/ssh/*key.pub
chmod 644 /etc/ssh/*key.pub
```
- 4 다음 명령을 실행하여 SSH 개인 키 파일을 확인합니다. `ls -l /etc/ssh/*key`
- 5 다음 명령을 실행하여 모든 문제를 해결합니다.


```
chown root /etc/ssh/*key
chgrp root /etc/ssh/*key
chmod 644 /etc/ssh/*key
```

가상 장치 관리 인터페이스 사용자 변경

가상 장치 관리 인터페이스에서 사용자를 추가하고 삭제하여 적절한 보안 수준을 생성할 수 있습니다.

가상 장치 관리 인터페이스의 루트 사용자 계정은 인증을 위해 PAM을 사용하기 때문에 PAM에 의해 설정된 클리핑 수준도 적용됩니다. 가상 장치 관리 인터페이스를 적절하게 분리하지 않은 경우에는 공격자가 무차별 암호 대입 공격으로 로그인을 시도할 경우 시스템 루트 계정에 잠금이 설정될 수 있습니다. 또한 루트 계정이 조직에 속한 사용자 두 명 이상의 부인 방지를 제공하는 데 부족하다고 판단될 경우에는 관리 인터페이스의 관리자를 변경하도록 선택할 수 있습니다.

필수 조건

프로시저

- 1 다음 명령을 실행하여 새 사용자를 생성한 후 가상 장치 관리 인터페이스 그룹에 추가합니다.


```
useradd -G vami,root user
```
- 2 사용자의 암호를 생성합니다.


```
passwd user
```

- 3 (선택 사항) 다음 명령을 실행하여 가상 장치 관리 인터페이스에서 루트 액세스를 사용하지 않도록 설정합니다.

```
usermod -R vami root
```

참고 가상 장치 관리 인터페이스에 대한 루트 액세스를 사용하지 않도록 설정하면 [관리] 탭에서 관리자(또는 루트) 암호를 업데이트할 수 있는 기능도 사용할 수 없습니다.

부팅 로더 인증 설정

적절한 보안 수준을 제공하려면 VMware 가상 장치에서 부팅 로더 인증을 구성합니다.

시스템의 부팅 로더에 인증이 필요하지 않은 경우 시스템 콘솔 액세스 권한이 있는 사용자는 시스템 부팅 구성을 변경하거나 시스템을 단일 사용자 또는 유지 보수 모드로 부팅할 수 있으며 이로 인해 서비스 거부 또는 인증되지 않은 시스템 액세스가 발생할 수 있습니다. 부팅 로더 인증은 VMware 가상 장치에서 기본적으로 설정되지 않기 때문에 GRUB 암호를 생성하여 이를 구성해야 합니다.

프로시저

- 1 가상 장치의 `/boot/grub/menu.lst` 파일에서 `password --md5 <password-hash>` 줄을 찾아 부팅 암호가 있는지 확인합니다.
- 2 암호가 없는 경우 가상 장치에서 `# /usr/sbin/grub-md5-crypt` 명령을 실행합니다.
MD5 암호가 생성되고 명령이 md5 해시 출력을 제공합니다.
- 3 `# password --md5 <hash from grub-md5-crypt>` 명령을 실행하여 암호를 `menu.lst` 파일에 추가합니다.

NTP 구성

중요한 시간 소싱의 경우 vRealize Automation 장치에서 호스트 시간 동기화를 사용하지 않도록 설정하고 NTP(네트워크 시간 프로토콜)를 사용합니다.

vRealize Automation 장치의 NTP 데몬은 동기화된 시간 서비스를 제공합니다. NTP는 기본적으로 사용하지 않도록 설정되기 때문에 수동으로 구성해야 합니다. 가능한 경우 정확한 감사 및 로그 유지를 통해 사용자 작업을 추적하고 잠재적인 악성 공격 및 침입을 감지하기 위해 운영 환경에서 NTP를 사용합니다. NTP 보안 알림에 대한 자세한 내용은 NTP 웹 사이트를 참조하십시오.

NTP 구성 파일은 각 장치의 `/etc/` 폴더에 있습니다. vRealize Automation 장치에 대해 NTP 서비스를 사용하도록 설정하고 가상 장치 관리 인터페이스의 **관리** 탭에서 시간 서버를 추가할 수 있습니다.

프로시저

- 1 텍스트 편집기를 사용하여 가상 장치 호스트 시스템에서 `/etc/ntp.conf` 구성 파일을 엽니다.
- 2 파일 소유권을 `root:root`로 설정합니다.
- 3 사용 권한을 `0640`로 설정합니다.
- 4 NTP 서비스에 대한 서비스 거부 증폭 공격의 위험을 완화하려면 `/etc/ntp.conf` 파일을 열고 `restrict` 줄이 파일에 있는지 확인합니다.

```
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1
```

- 5 변경 사항이 있으면 저장하고 파일을 닫습니다.

전송 중인 vRealize Automation 장치 데이터에 대한 TLS 구성

vRealize Automation 배포에서 강력한 TLS 프로토콜을 사용하여 vRealize Automation 장치 구성 요소에 대한 전송 채널을 보호하는지 확인합니다.

성능 고려 사항을 위해 TLS는 일부 애플리케이션 서비스 간의 localhost 연결에 사용되지 않도록 설정되어 있습니다. 심층 방어가 중요한 경우 모든 localhost 통신에 TLS를 사용하도록 설정하십시오.

중요 로드 밸런서에서 TLS를 종료하는 경우 모든 로드 밸런서에서 SSLv2, SSLv3 및 TLS 1.0과 같이 안전하지 않은 프로토콜을 사용하지 않도록 설정하십시오.

Localhost 구성에 TLS 사용

기본적으로 일부 localhost 통신에는 TLS가 사용되지 않습니다. 모든 localhost 연결에 TLS를 사용하도록 설정하여 보안을 강화할 수 있습니다.

프로시저

- 1 SSH를 사용하여 vRealize Automation 장치에 연결합니다.
- 2 다음 명령을 실행하여 vcac 키 저장소에 대한 사용 권한을 설정합니다.


```
chown vcac.pivotal /etc/vcac/vcac.keystore
chmod 640 /etc/vcac/vcac.keystore
```
- 3 HAProxy 구성을 업데이트합니다.
 - a 다음 문자열이 포함된 줄을 찾습니다.


```
server local 127.0.0.1...
```

 그런 다음 해당 줄 끝에 다음을 추가합니다. “ssl verify none 이 섹션에는 다음과 같은 줄도 포함됩니다.

backend-horizon	backend-vro
backend-vra	backend-artifactory
backend-vra-health	
 - b backend-horizon의 포트를 8080에서 8443으로 변경합니다.
- 4 keystorePass의 암호를 가져옵니다.
 - a /etc/vcac/security.properties 파일에서 certificate.store.password 속성을 찾습니다.


```
예: certificate.store.password=s2enc~iom0GXATG+RB8ff7Wdm4Bg==
```
 - b 다음 명령을 사용하여 값을 해독합니다.


```
vcac-config prop-util -d --p VALUE
```

```
예: vcac-config prop-util -d --p s2enc~iom0GXATG+RB8ff7Wdm4Bg==
```
- 5 vRealize Automation 서비스를 구성합니다.
 - a /etc/vcac/server.xml 파일을 엽니다.
 - b Connector 태그에 다음 특성을 추가합니다. 이때 certificate.store.password를 etc/vcac/security.properties에 있는 인증서 저장소 암호 값으로 바꿉니다.


```
scheme="https" secure="true" SSLEnabled="true" sslProtocol="TLS"
keystoreFile="/etc/vcac/vcac.keystore" keyAlias="apache" keystorePass="
certificate.store.password"
```

- 6 vRealize Orchestrator 서비스를 구성합니다.
 - a /etc/vco/app/server.xml 파일을 엽니다.
 - b Connector 태그에 다음 특성을 추가합니다. 이때 certificate.store.password를 etc/vcac/security.properties에 있는 인증서 저장소 암호 값으로 바꿉니다.


```
scheme="https" secure="true" SSLEnabled="true" sslProtocol="TLS"
                    keystoreFile="/etc/vcac/vcac.keystore" keyAlias="apache" keystorePass="
                    certificate.store.password"
```
- 7 vRealize Orchestrator, vRealize Automation 및 haproxy 서비스를 다시 시작합니다.


```
service vcac-server restart
service vco-server restart
service haproxy restart
```
- 8 가상 장치 관리 인터페이스를 구성합니다.
 - a /opt/vmware/share/htdocs/service/café-services/services.py 파일을 엽니다.
 - b conn = httpLib.HTTP() 줄을 conn = httpLib.HTTPS()로 변경하여 보안 기능을 향상시킵니다.

FIPS(Federal Information Processing Standard) 140-2 규격 사용

vRealize Automation 장치는 이제 모든 인바운드 및 아웃바운드 네트워크 트래픽에 대해 TLS를 통한 전송 중 데이터에 OpenSSL의 FIPS(Federal Information Processing Standard) 140-2 인증 버전을 사용합니다.

FIPS 모드는 vRealize Automation 장치 관리 인터페이스에서 사용 또는 사용하지 않도록 설정할 수 있습니다. 루트로 로그인한 상태에서 다음 명령을 사용하여 명령줄에서 FIPS를 구성할 수도 있습니다.

```
vcac-vami fips enable
vcac-vami fips disable
vcac-vami fips status
```

FIPS를 사용하면 포트 443의 인바운드 및 아웃바운드 vRealize Automation 장치 네트워크 트래픽에 FIPS 140-2 규격 암호화가 사용됩니다. FIPS 설정에 관계없이 vRealize Automation은 AES-256을 사용하여 vRealize Automation 장치에 저장된 보안 데이터를 보호합니다.

참고 현재 vRealize Automation은 일부 내부 구성 요소가 인증된 암호화 모듈을 사용하지 않기 때문에 FIPS 규격을 부분적으로만 사용합니다. 인증된 모듈이 아직 구현되지 않은 경우에는 모든 암호화 알고리즘에 AES-256 기반 암호화가 사용됩니다.

참고 다음 절차에서 구성을 변경하면 물리적 시스템이 재부팅됩니다.

프로시저

- 1 vRealize Automation 장치 관리 인터페이스에 루트로 로그인합니다.


```
https://vrealize-automation-appliance-FQDN:5480
```
- 2 **vRA 설정 > 호스트 설정**을 선택합니다.
- 3 오른쪽 위에서 [작업] 제목 아래의 버튼을 클릭하여 FIPS를 사용 또는 사용 안 함으로 설정합니다.
- 4 **예**를 클릭하여 vRealize Automation 장치를 다시 시작합니다.

SSLv3 및 TLS 1.0이 사용하지 않도록 설정되었는지 확인

강화 프로세스의 일부로 배포된 vRealize Automation 장치가 보안 전송 채널을 사용하는지 확인합니다.

필수 조건

“[Localhost 구성에 TLS 사용](#),” (27 페이지)을 완료합니다.

프로시저

- 1 SSLv3이 vRealize Automation 장치의 HAProxy https 처리기에서 사용되지 않도록 설정되었는지 확인합니다.

이 파일 검토	다음 항목이 있는지 확인	표시된 대로 적절한 줄에 있음
/etc/haproxy/conf.d/20-vcac.cfg	no-ssl3	bind 0.0.0.0:443 ssl crt /etc/apache2/server.pem ciphers TLSv1+HIGH:!aNULL:!eNULL:!3DES:!RC4:! CAMELLIA:!DH:!kECDHE:@STRENGTH no-ssl3 no- t1sv10
/etc/haproxy/conf.d/30-vro- config.cfg	no-ssl3	bind :8283 ssl crt /opt/vmware/etc/lighttpd/server.pem ciphers TLSv1+HIGH:!aNULL:!eNULL:!3DES:!RC4:! CAMELLIA:!DH:!kECDHE:@STRENGTH no-ssl3 no- t1sv10

- 2 /opt/vmware/etc/lighttpd/lighttpd.conf 파일을 열고 올바른 사용 안 함 항목이 표시되는지 확인합니다.


```
ssl.use-ssl2 = "disable"
ssl.use-ssl3 = "disable"
```
- 3 vRealize Automation 장치에서 콘솔 프록시에 대해 SSLv3을 사용하지 않도록 설정했는지 확인합니다.
 - a 다음 줄을 추가하거나 수정하여 /etc/vcac/security.properties 파일을 편집합니다.


```
consoleproxy.ssl.server.protocols = TLSv1.2, TLSv1.1
```
 - b 다음 명령을 실행하여 서버를 다시 시작합니다.


```
service vcac-server restart
```
- 4 SSLv3이 vCO 서비스에 대해 사용되지 않도록 설정되었는지 확인합니다.
 - a /etc/vco/app-server/server.xml 파일에서 <Connector> 태그를 찾고 다음 특성을 추가합니다.


```
sslEnabledProtocols = "TLSv1.1,TLSv1.2"
```
 - b 다음 명령을 실행하여 vCO 서비스를 다시 시작합니다.


```
service vco-server restart
```
- 5 SSLv3이 vRealize Automation 서비스에 대해 사용되지 않도록 설정되었는지 확인합니다.
 - a /etc/vcac/server.xml 파일의 <Connector> 태그에 다음 특성을 추가합니다.


```
sslEnabledProtocols = "TLSv1.1,TLSv1.2"
```
 - b 다음 명령을 실행하여 vRealize Automation 서비스를 다시 시작합니다.


```
service vcac-server restart
```

- 6 SSLv3이 RabbitMQ에 대해 사용되지 않도록 설정되었는지 확인합니다.

/etc/rabbitmq/rabbitmq.config 파일을 열고 {versions, ['tlsv1.2', 'tlsv1.1']}이 ssl 및 ssl_options 섹션에 있는지 확인합니다.

```
[
  {ssl, [
    {versions, ['tlsv1.2', 'tlsv1.1']},
    {ciphers, ["AES256-SHA", "AES128-SHA"]}],
  },
  {rabbit, [
    {tcp_listeners, [{"127.0.0.1", 5672}]},
    {frame_max, 262144},
    {ssl_listeners, [5671]},
    {ssl_options, [
      {cacertfile, "/etc/rabbitmq/certs/ca/cacert.pem"},
      {certfile, "/etc/rabbitmq/certs/server/cert.pem"},
      {keyfile, "/etc/rabbitmq/certs/server/key.pem"},
      {versions, ['tlsv1.2', 'tlsv1.1']},
      {ciphers, ["AES256-SHA", "AES128-SHA"]},
      {verify, verify_peer},
      {fail_if_no_peer_cert, false}
    ]},
    {mnesia_table_loading_timeout, 600000},
    {cluster_partition_handling, autoheal},
    {heartbeat, 600}
  ]},
  {kernel, [{net_ticktime, 120}]}
].
```

- 7 다음 명령을 실행하여 RabbitMQ 서버를 다시 시작합니다.

```
# service rabbitmq-server restart
```

- 8 SSLv3이 vIDM 서비스에 대해 사용되지 않도록 설정되었는지 확인합니다.

SSLEnabled="true"가 포함된 커넥터의 각 인스턴스에 대한 /opt/vmware/horizon/workspace/config/server.xml 파일을 열고 다음 줄이 있는지 확인합니다.

```
sslEnabledProtocols="TLSv1.1,TLSv1.2"
```

TLS 1.0 사용 안 함

적용 가능한 vRealize Automation 구성 요소에서 TLS 1.0을 사용하지 않도록 설정합니다.

Lighttpd에서 TLS 1.0을 사용 안 함으로 설정하는 지시문은 없습니다. 아래의 2단계에 설명된 대로 OpenSSL이 TLS 1.0의 암호 그룹을 사용하지 않도록 하여 TLS 1.0 사용에 대한 제한을 부분적으로 최소화할 수 있습니다.

프로시저

- 1 vRealize Automation 장치의 HAProxy https 처리기에서 TLS 1.0을 사용 안 함으로 설정합니다.

- a /etc/haproxy/conf.d/20-vcac.cfg 파일에서 다음 항목 끝에 no-tlsv10을 추가합니다.

```
bind 0.0.0.0:443 ssl crt /etc/apache2/server.pem ciphers TLSv1+HIGH:!aNULL:!eNULL:!3DES:!RC4:!
CAMELLIA:!DH:!kECDHE:@STRENGTH no-sslsv3 no-tlsv10
```

- b /etc/haproxy/conf.d/30-vro-config.cfg 파일에서 다음 항목 끝에 no-tlsv10을 추가합니다.

```
bind :8283 ssl crt /opt/vmware/etc/lighttpd/server.pem ciphers TLSv1+HIGH:!aNULL:!eNULL:!3DES:!
RC4:!CAMELLIA:!DH:!kECDHE:@STRENGTH no-sslsv3 no-tlsv10
```

참고 TLS 1.0을 다시 사용하도록 설정하려면 bind 지시문에서 no-tlsv10을 제거합니다.

- 2 Lighttpd에서 OpenSSL이 TLS 1.0의 암호 그룹을 사용하지 않는지 확인합니다.

- a /opt/vmware/etc/lighttpd/lighttpd.conf 파일의 ssl.cipher-list 줄을 다음과 같이 편집합니다.

```
ssl.cipher-list = "ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-
POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-
AES128-SHA256"
```

- b 다음 명령을 사용하여 lighttpd를 다시 시작합니다.

```
service vami-lighttp restart
```

- 3 vRealize Automation 장치에서 콘솔 프록시에 대해 TLS 1.0을 사용 안 함으로 설정합니다.

- a /etc/vcac/security.properties 파일에서 다음 줄을 추가하거나 수정합니다.

```
consoleproxy.ssl.server.protocols = TLSv1.2, TLSv1.1
```

- b 다음 명령을 실행하여 서버를 다시 시작합니다.

```
service vcac-server restart
```

참고 TLS 1.0을 다시 사용하도록 설정하려면 다음과 같이 TLSv1을 추가한 다음 vcac-server 서비스를 다시 시작합니다.

```
consoleproxy.ssl.server.protocols = TLSv1.2,TLSv1.1, TLSv1
```

- 4 vCO 서비스에 대해 TLS 1.0을 사용 안 함으로 설정합니다.

- a /etc/vco/app/server/server.xml 파일에서 <Connector> 태그를 찾아 다음 특성을 추가합니다.

```
sslEnabledProtocols = "TLSv1.1,TLSv1.2"
```

- b 다음 명령을 실행하여 vCO 서비스를 다시 시작합니다.

```
service vco-server restart
```

- 5 vRealize Automation 서비스에 대해 TLS 1.0을 사용 안 함으로 설정합니다.

- a /etc/vcac/server.xml 파일에서 <Connector> 태그를 찾아 다음 특성을 추가합니다.

```
sslEnabledProtocols = "TLSv1.1,TLSv1.2"
```

- b 다음 명령을 실행하여 vRealize Automation 서비스를 다시 시작합니다.

```
service vcac-server restart
```

참고 TLS 1.0을 다시 사용하도록 설정하려면 sslEnabledProtocols에 TLSv1을 추가합니다.
예: sslEnabledProtocols = "TLSv1.1,TLSv1.2,TLSv1"

6 RabbitMQ에 대해 TLS 1.0을 사용 안 함으로 설정합니다.

- a `/etc/rabbitmq/rabbitmq.config` 파일을 열고 다음 예에 나와 있는 것처럼 `tlsv1.2` 및 `tlsv1.1`이 `ssl` 섹션과 `ssl_options` 섹션에 추가되었는지 확인합니다.

```
[
  {ssl, [
    {versions, ['tlsv1.2', 'tlsv1.1']},
    {ciphers, ["AES256-SHA", "AES128-SHA"]}
  ]},
  {rabbit, [
    {tcp_listeners, [{"127.0.0.1", 5672}]},
    {frame_max, 262144},
    {ssl_listeners, [5671]},
    {ssl_options, [
      {cacertfile, "/etc/rabbitmq/certs/ca/cacert.pem"},
      {certfile, "/etc/rabbitmq/certs/server/cert.pem"},
      {keyfile, "/etc/rabbitmq/certs/server/key.pem"},
      {versions, ['tlsv1.2', 'tlsv1.1']},
      {ciphers, ["AES256-SHA", "AES128-SHA"]},
      {verify, verify_peer},
      {fail_if_no_peer_cert, false}
    ]},
    {mnesia_table_loading_timeout, 600000},
    {cluster_partition_handling, autoheal},
    {heartbeat, 600}
  ]},
  {kernel, [{net_ticktime, 120}]}
].
```

- b 다음 명령을 실행하여 RabbitMQ 서버를 다시 시작합니다.

```
# service rabbitmq-server restart
```

vRealize Automation 구성 요소에 대한 TLS 암호 그룹 구성

최상의 보안을 위해 강력한 암호를 사용하도록 vRealize Automation 구성 요소를 구성해야 합니다.

서버와 브라우저 사이에 협상되는 암호화 암호는 TLS 세션에 사용되는 암호화 강도를 결정합니다.

강력한 암호만 선택되도록 하려면 vRealize Automation 구성 요소에서 약한 암호를 사용하지 않도록 설정합니다. 강력한 암호만 지원하고 충분한 키 크기를 사용하도록 서버를 구성합니다. 또한 모든 암호를 적합한 순서로 구성합니다.

NULL 암호 그룹, aNULL 또는 eNULL과 같이 인증을 제공하지 않는 암호 그룹은 사용하지 않도록 설정합니다. 또한 익명 Diffie-Hellman 키 교환(ADH), 내보내기 수준 암호(EXP, DES를 포함하는 암호), 페이로드 트래픽 암호화에 대한 128비트 미만의 키 크기, 페이로드 트래픽에 대한 해시 메커니즘으로 MD5 사용, IDEA 암호화 그룹 및 RC4 암호화 그룹을 사용하지 않도록 설정합니다. 또한 Diffie-Hellman(DHE) 키 교환을 사용하는 암호 그룹을 사용하지 않도록 설정되었는지 확인합니다.

HA 프록시에 약한 암호 사용 안 함

허용되는 암호 목록을 기준으로 vRealize Automation 장치 HA 프록시 서비스 암호를 검토하고 약한 것으로 생각되는 모든 암호를 사용 안 함으로 설정합니다.

NULL 암호 그룹, aNULL 또는 eNULL과 같이 인증을 제공하지 않는 암호 그룹은 사용하지 않도록 설정합니다. 또한 익명 Diffie-Hellman 키 교환(ADH), 내보내기 수준 암호(EXP, DES를 포함하는 암호), 페이로드 트래픽 암호화에 대한 128비트 미만의 키 크기, 페이로드 트래픽에 대한 해시 메커니즘으로 MD5 사용, IDEA 암호화 그룹 및 RC4 암호화 그룹을 사용하지 않도록 설정합니다.

프로시저

- 1 /etc/haproxy/conf.d/20-vcac.cfg 파일에서 bind 지시문의 암호 항목을 검토하고, 약한 것으로 생각되는 모든 암호를 사용 안 함으로 설정합니다.

```
bind 0.0.0.0:443 ssl crt /etc/apache2/server.pem ciphers TLSv1+HIGH:!aNULL:!eNULL:!3DES:!RC4:!
CAMELLIA:!DH:!kECDHE:@STRENGTH no-ssl3 no-tls10
```

- 2 /etc/haproxy/conf.d/30-vro-config.cfg 파일에서 bind 지시문의 암호 항목을 검토하고, 약한 것으로 생각되는 모든 암호를 사용 안 함으로 설정합니다.

```
bind :8283 ssl crt /opt/vmware/etc/lighttpd/server.pem ciphers TLSv1+HIGH:!aNULL:!eNULL:!3DES:!RC4:!
CAMELLIA:!DH:!kECDHE:@STRENGTH no-ssl3 no-tls10
```

vRealize Automation 장치 vRealize Automation 장치 콘솔 프록시 서비스에서 약한 암호 사용 안 함

허용되는 암호 목록을 기준으로 vRealize Automation 장치 콘솔 프록시 서비스 암호를 검토하고 약한 것으로 생각되는 모든 암호를 사용 안 함으로 설정합니다.

NULL 암호 그룹, aNULL 또는 eNULL과 같이 인증을 제공하지 않는 암호 그룹은 사용하지 않도록 설정합니다. 또한 익명 Diffie-Hellman 키 교환(ADH), 내보내기 수준 암호(EXP, DES를 포함하는 암호), 페이로드 트래픽 암호화에 대한 128비트 미만의 키 크기, 페이로드 트래픽에 대한 해시 메커니즘으로 MD5 사용, IDEA 암호화 그룹 및 RC4 암호화 그룹을 사용하지 않도록 설정합니다.

프로시저

- 1 /etc/vcac/security.properties 파일을 텍스트 편집기에서 엽니다.
- 2 원하지 않는 암호 그룹을 사용 안 함으로 설정하는 줄을 파일에 추가합니다.

다음 줄을 변형하여 사용하십시오.

```
consoleproxy.ssl.ciphers.disallowed=cipher_suite_1, cipher_suite_2, etc
```

예를 들어 AES 128 암호 그룹을 사용 안 함으로 설정하려면 다음 줄을 추가합니다.

```
consoleproxy.ssl.ciphers.disallowed=TLS_DH_DSS_WITH_AES_128_CBC_SHA, TLS_DH_DSS_WITH_AES_256_CBC_SHA,
TLS_DH_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_DSS_WITH_AES_128_CBC_SHA, TLS_DHE_DSS_WITH_AES_256_CBC_SHA,
TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA
```

- 3 다음 명령을 사용하여 서버를 다시 시작합니다.

```
service vcac-server restart
```

vRealize Automation 장치 vCO 서비스에서 약한 암호 사용 안 함

허용되는 암호 목록을 기준으로 vRealize Automation 장치 vCO 서비스 암호를 검토하고 약한 것으로 생각되는 모든 암호를 사용 안 함으로 설정합니다.

NULL 암호 그룹, aNULL 또는 eNULL과 같이 인증을 제공하지 않는 암호 그룹은 사용하지 않도록 설정합니다. 또한 익명 Diffie-Hellman 키 교환(ADH), 내보내기 수준 암호(EXP, DES를 포함하는 암호), 페이로드 트래픽 암호화에 대한 128비트 미만의 키 크기, 페이로드 트래픽에 대한 해시 메커니즘으로 MD5 사용, IDEA 암호화 그룹 및 RC4 암호화 그룹을 사용하지 않도록 설정합니다.

프로시저

- 1 /etc/vco/app/server/server.xml 파일에서 <Connector> 태그를 찾습니다.

- 원하는 암호 그룹을 사용하도록 암호 특성을 편집하거나 추가합니다.

다음 예를 참조하십시오.

```
ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
```

vRealize Automation 장치 RabbitMQ 서비스에서 약한 암호 사용 안 함

허용되는 암호 목록을 기준으로 vRealize Automation 장치 RabbitMQ 서비스 암호를 검토하고 약한 것으로 생각되는 모든 암호를 사용 안 함으로 설정합니다.

NULL 암호 그룹, aNULL 또는 eNULL과 같이 인증을 제공하지 않는 암호 그룹은 사용하지 않도록 설정합니다. 또한 익명 Diffie-Hellman 키 교환(ADH), 내보내기 수준 암호(EXP, DES를 포함하는 암호), 페이로드 트래픽 암호화에 대한 128비트 미만의 키 크기, 페이로드 트래픽에 대한 해시 메커니즘으로 MD5 사용, IDEA 암호화 그룹 및 RC4 암호화 그룹을 사용하지 않도록 설정합니다.

프로시저

- 지원되는 암호 그룹을 평가하기 위해 # /usr/sbin/rabbitmqctl eval 'ssl:cipher_suites().' 명령을 실행합니다.

다음 예에서 반환되는 암호는 지원되는 암호만 나타냅니다. RabbitMQ 서버는 rabbitmq.config 파일에 구성된 경우가 아니면 이러한 암호를 사용하거나 알리지 않습니다.

```
[ "ECDHE-ECDSA-AES256-GCM-SHA384", "ECDHE-RSA-AES256-GCM-SHA384",
  "ECDHE-ECDSA-AES256-SHA384", "ECDHE-RSA-AES256-SHA384",
  "ECDH-ECDSA-AES256-GCM-SHA384", "ECDH-RSA-AES256-GCM-SHA384",
  "ECDH-ECDSA-AES256-SHA384", "ECDH-RSA-AES256-SHA384",
  "DHE-RSA-AES256-GCM-SHA384", "DHE-DSS-AES256-GCM-SHA384",
  "DHE-RSA-AES256-SHA256", "DHE-DSS-AES256-SHA256", "AES256-GCM-SHA384",
  "AES256-SHA256", "ECDHE-ECDSA-AES128-GCM-SHA256",
  "ECDHE-RSA-AES128-GCM-SHA256", "ECDHE-ECDSA-AES128-SHA256",
  "ECDHE-RSA-AES128-SHA256", "ECDH-ECDSA-AES128-GCM-SHA256",
  "ECDH-RSA-AES128-GCM-SHA256", "ECDH-ECDSA-AES128-SHA256",
  "ECDH-RSA-AES128-SHA256", "DHE-RSA-AES128-GCM-SHA256",
  "DHE-DSS-AES128-GCM-SHA256", "DHE-RSA-AES128-SHA256", "DHE-DSS-AES128-SHA256",
  "AES128-GCM-SHA256", "AES128-SHA256", "ECDHE-ECDSA-AES256-SHA",
  "ECDHE-RSA-AES256-SHA", "DHE-RSA-AES256-SHA", "DHE-DSS-AES256-SHA",
  "ECDH-ECDSA-AES256-SHA", "ECDH-RSA-AES256-SHA", "AES256-SHA",
  "ECDHE-ECDSA-DES-CBC3-SHA", "ECDHE-RSA-DES-CBC3-SHA", "EDH-RSA-DES-CBC3-SHA",
  "EDH-DSS-DES-CBC3-SHA", "ECDH-ECDSA-DES-CBC3-SHA", "ECDH-RSA-DES-CBC3-SHA",
  "DES-CBC3-SHA", "ECDHE-ECDSA-AES128-SHA", "ECDHE-RSA-AES128-SHA",
  "DHE-RSA-AES128-SHA", "DHE-DSS-AES128-SHA", "ECDH-ECDSA-AES128-SHA",
  "ECDH-RSA-AES128-SHA", "AES128-SHA" ]
```

- 조직의 보안 요구 사항을 충족하는 지원 암호를 선택합니다.

예를 들어 ECDHE-ECDSA-AES128-GCM-SHA256 & ECDHE-ECDSA-AES256-GCM-SHA384만 허용하려면 /etc/rabbitmq/rabbitmq.config 파일을 검토하고 ssl 및 ssl_options에 다음 줄을 추가합니다.

```
{ciphers, [ "ECDHE-ECDSA-AES128-GCM-SHA256", "ECDHE-ECDSA-AES256-GCM-SHA384" ]}
```

- 다음 명령을 사용하여 RabbitMQ 서버를 다시 시작합니다.

```
service rabbitmq-server restart
```

미사용 데이터의 보안 확인

vRealize Automation에 사용되는 데이터베이스 사용자 및 계정의 보안을 확인합니다.

Postgres 사용자

Postgres Linux 사용자 계정은 Postgres 데이터베이스 슈퍼 사용자 계정 역할에 연결되어 있으며 기본적으로 잠긴 계정입니다. 이는 루트 사용자 계정에서만 액세스할 수 있기 때문에 이 사용자에게 가장 안전한 구성입니다. 이 사용자 계정을 잠금 해제하지 마십시오.

데이터베이스 사용자 계정 역할

기본 Postgres 사용자 계정 역할은 애플리케이션 기능 이외의 용도로 사용되어서는 안 됩니다. 기본이 아닌 데이터베이스 검토 또는 보고 작업을 지원하려면 추가 계정이 생성되어야 하며 암호가 적절히 보호되어야 합니다.

명령줄에서 다음 스크립트를 실행합니다.

```
vcac-vami add-db-user newUsername newPassword
```

그러면 새 사용자가 추가되고 암호가 해당 사용자를 통해 제공됩니다.

참고 이 스크립트는 마스터-슬레이브 HA Postgres 설정이 구성된 경우 마스터 Postgres 데이터베이스에 대해 실행되어야 합니다.

PostgreSQL 클라이언트 인증 구성

vRealize Automation 장치 PostgreSQL 데이터베이스에 대해 로컬 신뢰 인증이 구성되어 있지 않은지 확인합니다. 이 구성은 데이터베이스 슈퍼유저를 포함한 모든 로컬 사용자가 암호 없이 모든 PostgreSQL 사용자로 로그인할 수 있도록 허용합니다.

참고 Postgres 슈퍼유저 계정은 로컬 신뢰로 두어야 합니다.

암호화된 암호를 보내는 md5 인증 방법을 사용하는 것이 좋습니다.

클라이언트 인증 구성 설정은 /storage/db/pgdata/pg_hba.conf 파일에 있습니다.

```
# TYPE DATABASE          USER                ADDRESS             METHOD
# "local" is for Unix domain socket connections only
local    all             postgres            trust
# IPv4 local connections:
#host    all             all                 127.0.0.1/32       md5
hostssl  all             all                 127.0.0.1/32       md5
# IPv6 local connections:
#host    all             all                 ::1/128            md5
hostssl  all             all                 ::1/128            md5

# Allow remote connections for VCAC user.
#host    vcac              vcac                0.0.0.0/0          md5
hostssl  vcac              vcac                0.0.0.0/0          md5
hostssl  vcac              vcac                ::0/0              md5
# Allow remote connections for VCAC replication user.
#host    vcac              vcac_replication   0.0.0.0/0          md5
hostssl  vcac              vcac_replication   0.0.0.0/0          md5
hostssl  vcac              vcac_replication   ::0/0              md5
```

```
# Allow replication connections by a user with the replication privilege.
#host replication          vcac_replication 0.0.0.0/0      md5
hostssl replication       vcac_replication 0.0.0.0/0      md5
hostssl replication       vcac_replication ::0/0      md5

pg_hba.conf 파일을 편집할 경우, 다음 명령을 실행하여 Postgres 서버를 다시 시작해야만 변경 내용이 적용됩니다.

# cd /opt/vmware/vpostgres/9.2/bin
# su postgres
# ./pg_ctl restart -D /storage/db/pgdata/ -m fast
```

vRealize Automation 애플리케이션 리소스 구성

vRealize Automation 애플리케이션 리소스를 검토하고 파일 사용 권한을 제한합니다.

프로시저

- 1 다음 명령을 실행하여 SUID 및 GUID 비트가 설정된 파일이 올바르게 정의되었는지 확인합니다.

```
find / -path /proc -prune -o -type f -perm +6000 -ls
```

다음 목록이 표시되어야 합니다.

```
2197357  24 -rwsr-xr-x  1 polkituser root      23176 Mar 31  2015 /usr/lib/PolicyKit/polkit-set-
default-helper
2197354  16 -rwxr-sr-x  1 root      polkituser  14856 Mar 31  2015 /usr/lib/PolicyKit/polkit-read-
auth-helper
2197353  12 -rwsr-x---  1 root      polkituser  10744 Mar 31  2015 /usr/lib/PolicyKit/polkit-grant-
helper-pam
2197352  20 -rwxr-sr-x  1 root      polkituser  19208 Mar 31  2015 /usr/lib/PolicyKit/polkit-grant-
helper
2197351  20 -rwxr-sr-x  1 root      polkituser  19008 Mar 31  2015 /usr/lib/PolicyKit/polkit-
explicit-grant-helper
2197356  24 -rwxr-sr-x  1 root      polkituser  23160 Mar 31  2015 /usr/lib/PolicyKit/polkit-revoke-
helper
2188203 460 -rws--x--x  1 root      root        465364 Apr 21 22:38 /usr/lib64/ssh/ssh-keysign
2138858  12 -rwxr-sr-x  1 root      tty         10680 May 10  2010 /usr/sbin/utempter
2142482 144 -rwsr-xr-x  1 root      root        142890 Sep 15  2015 /usr/bin/passwd
2142477 164 -rwsr-xr-x  1 root      shadow      161782 Sep 15  2015 /usr/bin/chage
2142467 156 -rwsr-xr-x  1 root      shadow      152850 Sep 15  2015 /usr/bin/chfn
1458298 364 -rwsr-xr-x  1 root      root        365787 Jul 22  2015 /usr/bin/sudo
2142481  64 -rwsr-xr-x  1 root      root        57776 Sep 15  2015 /usr/bin/newgrp
1458249  40 -rwsr-x---  1 root      trusted    40432 Mar 18  2015 /usr/bin/crontab
2142478 148 -rwsr-xr-x  1 root      shadow      146459 Sep 15  2015 /usr/bin/chsh
2142480 156 -rwsr-xr-x  1 root      shadow      152387 Sep 15  2015 /usr/bin/gpasswd
2142479  48 -rwsr-xr-x  1 root      shadow      46967 Sep 15  2015 /usr/bin/expiry
311484  48 -rwsr-x---  1 root      messagebus 47912 Sep 16  2014 /lib64/dbus-1/dbus-daemon-launch-
helper
876574  36 -rwsr-xr-x  1 root      shadow      35688 Apr 10  2014 /sbin/unix_chkpwd
876648  12 -rwsr-xr-x  1 root      shadow      10736 Dec 16  2011 /sbin/unix2_chkpwd
 49308  68 -rwsr-xr-x  1 root      root        63376 May 27  2015 /opt/likewise/bin/ksu
1130552  40 -rwsr-xr-x  1 root      root        40016 Apr 16  2015 /bin/su
1130511  40 -rwsr-xr-x  1 root      root        40048 Apr 15  2011 /bin/ping
1130600 100 -rwsr-xr-x  1 root      root        94808 Mar 11  2015 /bin/mount
1130601  72 -rwsr-xr-x  1 root      root        69240 Mar 11  2015 /bin/umount
1130512  36 -rwsr-xr-x  1 root      root        35792 Apr 15  2011 /bin/ping6
2012 /lib64/dbus-1/dbus-daemon-launch-helper
```

- 2 다음 명령을 실행하여 가상 장치에 있는 모든 파일에 대해 소유자가 있는지 확인합니다.

```
find / -path /proc -prune -o -nouser -o -nogroup
```

- 3 다음 명령을 실행하여 가상 장치에 대한 모든 파일의 사용 권한을 검토하고 world writable 사용 권한이 있는지 확인합니다.

```
find / -name "*" -type f -perm -a+w | xargs ls -ldb
```

- 4 다음 명령을 실행하여 vcac 사용자만 올바른 파일을 소유하는지 확인합니다.

```
find / -name "proc" -prune -o -user vcac -print | egrep -v -e "*/vcac/*" | egrep -v -e "*/vmware-vcac/*"
```

결과가 표시되지 않으면 모든 올바른 파일을 vcac 사용자만 소유하고 있는 것입니다.

- 5 다음 파일에 대해 vcac 사용자만 쓰기 가능한지 확인합니다.

```
/etc/vcac/vcac/security.properties
```

```
/etc/vcac/vcac/solution-users.properties
```

```
/etc/vcac/vcac/sso-admin.properties
```

```
/etc/vcac/vcac/vcac.keystore
```

```
/etc/vcac/vcac/vcac.properties
```

다음 파일과 해당 하위 디렉토리도 확인합니다.

```
/var/log/vcac/*
```

```
/var/lib/vcac/*
```

```
/var/cache/vcac/*
```

- 6 vcac 또는 루트 사용자만 다음 디렉토리와 해당 하위 디렉토리의 올바른 파일을 읽을 수 있는지 확인합니다.

```
/etc/vcac/*
```

```
/var/log/vcac/*
```

```
/var/lib/vcac/*
```

```
/var/cache/vcac/*
```

- 7 다음 디렉토리와 해당 하위 디렉토리에 나와 있는 올바른 파일을 vco 또는 루트 사용자만 소유하고 있는지 확인합니다.

```
/etc/vco/*
```

```
/var/log/vco/*
```

```
/var/lib/vco/*
```

```
/var/cache/vco/*
```

- 8 다음 디렉토리와 해당 하위 디렉토리에 나와 있는 올바른 파일에 대해 vco 또는 루트 사용자만 쓰기 가능한지 확인합니다.

```
/etc/vco/*
```

```
/var/log/vco/*
```

```
/var/lib/vco/*
```

```
/var/cache/vco/*
```

- 9 다음 디렉토리와 해당 하위 디렉토리에 나와 있는 올바른 파일에 대해 vco 또는 루트 사용자만 읽기 가능한지 확인합니다.

```
/etc/vco/*
```

```
/var/log/vco/*
```

```
/var/lib/vco/*
```

```
/var/cache/vco/*
```

콘솔 프록시 구성 사용자 지정

문제 해결 및 조직 업무를 용이하게 하기 위해 vRealize Automation에 대한 원격 콘솔 구성을 사용자 지정할 수 있습니다.

vRealize Automation을 설치, 구성 또는 유지 보수하는 경우 설치에 대한 디버깅 및 문제 해결이 가능하도록 일부 설정을 변경할 수 있습니다. 필요한 용도에 따라 적용 가능한 구성 요소의 보안을 제대로 유지할 수 있도록 변경하는 모든 내용을 목록으로 작성하고 감사를 수행하십시오. 구성 변경에 대한 보안이 올바른지가 확실하지 않으면 운영 환경으로 전환하지 마십시오.

VMware Remote Console 티켓 만료 기한 사용자 지정

VMware Remote Console 연결 설정에 사용되는 원격 콘솔 티켓에 대한 유효 기간을 사용자 지정할 수 있습니다.

사용자가 VMware Remote Console 연결을 생성하는 경우 시스템은 가상 시스템에 특정 연결을 설정하는 일회성 자격 증명을 생성하고 반환합니다. 지정된 기간에 대해 티켓 만료 기한을 분 단위로 설정할 수 있습니다.

프로시저

- 1 `/etc/vcac/security.properties` 파일을 텍스트 편집기에서 엽니다.
- 2 `consoleproxy.ticket.validitySec=30` 형식의 줄을 파일에 추가합니다.
이 줄에서 숫자 값은 티켓이 만료되기까지의 시간(분)을 지정합니다.
- 3 파일을 저장하고 닫습니다.
- 4 `/etc/init.d/vcac-server restart` 명령을 사용하여 `vcac-server`를 다시 시작합니다.

티켓 만료 기한 값이 분 단위의 지정된 시간으로 재설정됩니다.

콘솔 프록시 서버 포트 사용자 지정

VMware Remote Console 콘솔 프록시가 메시지를 수신하는 포트를 사용자 지정할 수 있습니다.

프로시저

- 1 `/etc/vcac/security.properties` 파일을 텍스트 편집기에서 엽니다.
- 2 `consoleproxy.service.port=8445` 형식의 줄을 파일에 추가합니다.
숫자 값은 콘솔 프록시 서비스 포트 번호를 지정하며, 이 경우 8445입니다.
- 3 파일을 저장하고 닫습니다.
- 4 `/etc/init.d/vcac-server restart` 명령을 사용하여 `vcac-server`를 다시 시작합니다.

프록시 서비스 포트가 지정된 포트 번호로 변경됩니다.

X-XSS-Protection 응답 헤더 구성

X-XSS-Protection 응답 헤더를 haproxy 구성 파일에 추가합니다.

프로시저

- 1 편집을 위해 `/etc/haproxy/conf.d/20-vcac.cfg`를 엽니다.
- 2 프론트 엔드 섹션에 다음 줄을 추가합니다.

```
rspdel X-XSS-Protection:W 1;W mode=block
      rspadd X-XSS-Protection:W 1;W mode=block
```

- 3 다음 명령을 사용하여 HAProxy 구성을 다시 로드합니다.

```
/etc/init.d/haproxy reload
```

HTTP Strict Transport Security 응답 헤더 구성

HTTP Strict Transport Security(HSTS) 응답 헤더를 HAProxy 구성에 추가합니다.

프로시저

- 1 편집을 위해 `/etc/haproxy/conf.d/20-vcac.cfg`를 엽니다.
- 2 프론트 엔드 섹션에 다음 줄을 추가합니다.

```
rspdel Strict-Transport-Security:W max-age=31536000
      rspadd Strict-Transport-Security:W max-age=31536000
```

- 3 다음 명령을 사용하여 HAProxy 구성을 다시 로드합니다.

```
/etc/init.d/haproxy reload
```

X-Frame-Options 응답 헤더 구성

X-Frame-Options 응답 헤더는 경우에 따라 두 번 표시될 수 있습니다.

X-Frame-Options 응답 헤더는 vIDM 서비스가 이 헤더를 백엔드는 물론 HAProxy에 추가하기 때문에 두 번 표시될 수 있습니다. 적절한 구성을 통해 두 번 표시되는 것을 막을 수 있습니다.

프로시저

- 1 편집을 위해 `/etc/haproxy/conf.d/20-vcac.cfg`를 엽니다.
- 2 프론트 엔드 섹션에서 다음 줄을 찾습니다.

```
rspadd X-Frame-Options:W SAMEORIGIN
```

- 3 위 단계에서 찾은 줄 앞에 다음 줄을 추가합니다.

```
rspdel X-Frame-Options:W SAMEORIGIN
```

- 4 다음 명령을 사용하여 HAProxy 구성을 다시 로드합니다.

```
/etc/init.d/haproxy reload
```

서버 응답 머리글 구성

보안 모범 사례로 잠재적 공격자가 사용할 수 있는 정보를 제한하도록 vRealize Automation 시스템을 구성합니다.

최대한 시스템이 해당 ID 및 버전에 대해 공유하는 정보 양을 최소화합니다. 해커 및 악의적 작업자는 이 정보를 사용하여 웹 서버 또는 버전에 대한 지정된 공격을 만들 수 있습니다.

Lighttpd 서버 응답 헤더 구성

vRealize Automation 장치 lighttpd 서버에 대한 빈 서버 헤더를 생성하는 것이 가장 좋습니다.

프로시저

- 1 /opt/vmware/etc/lighttpd/lighttpd.conf 파일을 텍스트 편집기에서 엽니다.
- 2 server.tag = " "를 파일에 추가합니다.
- 3 변경 사항을 저장하고 파일을 닫습니다.
- 4 # /opt/vmware/etc/init.d/vami-lighttp restart 명령을 실행하여 lighttpd 서버를 다시 시작합니다.

vRealize Automation 장치에 대한 TCServer 응답 헤더 구성

악성 공격자가 소중한 정보를 입수할 가능성을 제한하기 위해 vRealize Automation 장치와 함께 사용되는 TCServer 응답 헤더에 대한 사용자 지정 빈 서버 헤더를 생성하는 것이 가장 좋습니다.

프로시저

- 1 /etc/vco/app-server/server.xml 파일을 텍스트 편집기에서 엽니다.
- 2 각 <Connector> 요소에 server=" "를 추가합니다.
예: <Connector protocol="HTTP/1.1" server="" />
- 3 변경 사항을 저장하고 파일을 닫습니다.
- 4 다음 명령을 사용하여 서버를 다시 시작합니다.

```
service vco-server restart
```

인터넷 정보 서비스 서버 응답 헤더 구성

악성 공격자가 소중한 정보를 입수할 가능성을 제한하기 위해 Identity Appliance와 함께 사용되는 IIS(인터넷 정보 서비스) 서버에 대한 사용자 지정 빈 서버 헤더를 생성하는 것이 가장 좋습니다.

프로시저

- 1 C:\Windows\System32\inetresrv\urlscan\urlscan.ini 파일을 텍스트 편집기에서 엽니다.
- 2 RemoveServerHeader=0을 검색하여 RemoveServerHeader=1로 변경합니다.
- 3 변경 사항을 저장하고 파일을 닫습니다.
- 4 iisreset 명령을 실행하여 서버를 다시 시작합니다.

후속 작업

IIS 관리자 콘솔의 목록에서 HTTP 응답 헤더를 제거하여 IIS X-Powered By 헤더를 사용하지 않도록 설정합니다.

- 1 IIS 관리자 콘솔을 엽니다.
- 2 HTTP 응답 헤더를 열고 목록에서 제거합니다.
- 3 iisreset 명령을 실행하여 서버를 다시 시작합니다.

vRealize Automation 장치 세션 시간 초과 설정

회사 보안 정책에 따라 vRealize Automation 장치에서 세션 시간 초과 설정을 구성합니다.

사용자 비활성에 대한 vRealize Automation 장치 기본 세션 시간 초과는 30분입니다. 조직의 보안 정책을 준수하도록 이 시간 초과 값을 조정하려면 vRealize Automation 장치 호스트 시스템에서 web.xml 파일을 편집합니다.

프로시저

- 1 텍스트 편집기에서 /usr/lib/vcac/server/webapps/vcac/WEB-INF/web.xml 파일을 엽니다.
- 2 session-config를 찾고 세션 시간 초과 값을 설정합니다. 다음 코드 샘플을 확인합니다.

```
<!-- 30 minutes session expiration time -->
<session-config>
  <session-timeout>30</session-timeout>
  <tracking-mode>COOKIE</tracking-mode>
  <cookie-config>
    <path>/</path>
  </cookie-config>
</session-config>
```

- 3 다음 명령을 실행하여 Apache 서버를 다시 시작합니다.

```
service vcac-server restart
```

불필요한 소프트웨어 관리

보안 위험을 최소화하려면 vRealize Automation 호스트 시스템에서 불필요한 소프트웨어를 제거하거나 구성합니다.

제조업체 권장 사항 및 보안 모범 사례에 따라 제거하지 않는 모든 소프트웨어를 구성하여 보안 위반 가능성을 최소화합니다.

USB 대용량 스토리지 처리기 보안

VMware 가상 장치 호스트 시스템에서 USB 디바이스 처리기로 사용되지 않도록 하려면 USB 대용량 스토리지 처리기를 보안합니다. 잠재적 공격자는 이 처리기를 이용하여 시스템을 손상시킬 수 있습니다.

프로시저

- 1 텍스트 편집기에서 /etc/modprobe.conf.local 파일을 엽니다.
- 2 이 파일에 install usb-storage /bin/true 줄이 표시되는지 확인합니다.
- 3 파일을 저장하고 닫습니다.

Bluetooth 프로토콜 처리기 보안

가상 장치 호스트 시스템에서 Bluetooth 프로토콜 처리기를 보안하여 잠재적 공격자가 이를 이용하지 못하도록 합니다.

Bluetooth 프로토콜을 네트워크 스택에 바인딩하는 것은 불필요하며 이렇게 할 경우 호스트의 공격 표면이 증가할 수 있습니다.

프로시저

- 1 텍스트 편집기에서 /etc/modprobe.conf.local 파일을 엽니다.

- 이 파일에 다음 줄이 표시되는지 확인합니다.

```
install bluetooth /bin/true
```

- 파일을 저장하고 닫습니다.

SCTP(Stream Control Transmission Protocol) 보안

기본적으로 시스템에서 SCTP(Stream Control Transmission Protocol)가 로드되지 못하도록 합니다. 잠재적 공격자는 이 프로토콜을 이용하여 시스템을 손상시킬 수 있습니다.

절대적으로 필요한 경우가 아니면 SCTP(Stream Control Transmission Protocol) 모듈이 로드되지 못하도록 시스템을 구성합니다. SCTP는 미사용 IETF 표준화 전송 계층 프로토콜입니다. 이 프로토콜을 네트워크 스택에 바인딩하면 호스트의 공격 표면이 증가합니다. 권한 없는 로컬 프로세스는 프로토콜로 소켓을 열어 커널이 동적으로 프로토콜 처리기를 로드하게 할 수 있습니다.

프로시저

- 텍스트 편집기에서 `/etc/modprobe.conf.local` 파일을 엽니다.
- 이 파일에 다음 줄이 표시되는지 확인합니다.

```
install sctp /bin/true
```

- 파일을 저장하고 닫습니다.

DCCP(Datagram Congestion Protocol) 보안

시스템 강화 작업의 일부로 기본적으로 DCCP(Datagram Congestion Protocol)가 가상 장치 호스트 시스템에서 로드되지 못하도록 합니다. 잠재적 공격자는 이 프로토콜을 이용하여 시스템을 손상시킬 수 있습니다.

절대적으로 필요한 경우가 아니면 DCCP(Datagram Congestion Protocol) 모듈을 로드하지 않습니다. DCCP는 사용되지 않는 제안된 전송 계층 프로토콜입니다. 이 프로토콜을 네트워크 스택에 바인딩하면 호스트의 공격 표면이 증가합니다. 권한 없는 로컬 프로세스는 프로토콜로 소켓을 열어 커널이 동적으로 프로토콜 처리기를 로드하게 할 수 있습니다.

프로시저

- 텍스트 편집기에서 `/etc/modprobe.conf.local` 파일을 엽니다.
- 이 파일에 DCCP 줄이 표시되는지 확인합니다.

```
install dccp/bin/true
install dccp_ipv4/bin/true
install dccp_ipv6/bin/true
```

- 파일을 저장하고 닫습니다.

네트워크 브리징 보안

기본적으로 시스템에서 네트워크 브리징 모듈이 로드되지 못하도록 합니다. 잠재적 공격자는 이 모듈을 이용하여 시스템을 손상시킬 수 있습니다.

절대적으로 필요한 경우가 아니면 네트워크가 로드하지 못하도록 시스템을 구성합니다. 잠재적 공격자는 이 모듈을 사용하여 네트워크 파티셔닝 및 보안을 우회할 수 있습니다.

프로시저

- 모든 VMware 가상 장치 호스트 시스템에서 다음 명령을 실행합니다.

```
# rmmmod bridge
```

- 텍스트 편집기에서 `/etc/modprobe.conf.local` 파일을 엽니다.

- 이 파일에 다음 줄이 표시되는지 확인합니다.

```
install bridge /bin/false
```

- 파일을 저장하고 닫습니다.

보안 RDS(Reliable Datagram Sockets) 프로토콜

시스템 강화 작업의 일부로 기본적으로 RDS(Reliable Datagram Sockets) 프로토콜이 가상 장치 호스트 시스템에서 로드되지 못하도록 합니다. 잠재적 공격자는 이 프로토콜을 이용하여 시스템을 손상시킬 수 있습니다.

RDS(Reliable Datagram Sockets) 프로토콜을 네트워크 스택에 바인딩하면 호스트의 공격 표면이 증가합니다. 권한 없는 로컬 프로세스는 프로토콜로 소켓을 열어 시스템이 동적으로 프로토콜 처리기를 로드하게 할 수 있습니다.

프로시저

- 텍스트 편집기에서 `/etc/modprobe.conf.local` 파일을 엽니다.
- 이 파일에 `install rds /bin/true` 줄이 표시되는지 확인합니다.
- 파일을 저장하고 닫습니다.

TIPC(Transparent Inter-Process Communication) 프로토콜 보안

시스템 강화 작업의 일부로 기본적으로 TIPC(Transparent Inter-Process Communication) 프로토콜이 가상 장치 호스트 시스템에서 로드되지 못하도록 합니다. 잠재적 공격자는 이 프로토콜을 이용하여 시스템을 손상시킬 수 있습니다.

TIPC(Transparent Inter-Process Communication) 프로토콜을 네트워크 스택에 바인딩하면 호스트의 공격 표면이 증가합니다. 권한 없는 로컬 프로세스는 프로토콜로 소켓을 열어 커널이 동적으로 프로토콜 처리기를 로드하게 할 수 있습니다.

프로시저

- 텍스트 편집기에서 `/etc/modprobe.conf.local` 파일을 엽니다.
- 이 파일에 `install tipc /bin/true` 줄이 표시되는지 확인합니다.
- 파일을 저장하고 닫습니다.

IPX(Internetwork Packet Exchange) 프로토콜 보안

기본적으로 시스템에서 IPX(Internetwork Packet Exchange) 프로토콜이 로드되지 못하도록 합니다. 잠재적 공격자는 이 프로토콜을 이용하여 시스템을 손상시킬 수 있습니다.

절대적으로 필요한 경우가 아니면 IPX(Internetwork Packet Exchange) 프로토콜 모듈을 로드하지 않습니다. IPX 프로토콜은 사용되지 않는 네트워크 계층 프로토콜입니다. 이 프로토콜을 네트워크 스택에 바인딩하면 호스트의 공격 표면이 증가합니다. 권한 없는 로컬 프로세스는 프로토콜로 소켓을 열어 시스템이 동적으로 프로토콜 처리기를 로드하게 할 수 있습니다.

프로시저

- 텍스트 편집기에서 `/etc/modprobe.conf.local` 파일을 엽니다.
- 이 파일에 다음 줄이 표시되는지 확인합니다.


```
install ipx /bin/true
```
- 파일을 저장하고 닫습니다.

Appletalk 프로토콜 보안

기본적으로 시스템에서 Appletalk 프로토콜이 로드되지 못하도록 합니다. 잠재적 공격자는 이 프로토콜을 이용하여 시스템을 손상시킬 수 있습니다.

절대적으로 필요한 경우가 아니면 Appletalk 프로토콜 모듈을 로드하지 않습니다. 이 프로토콜을 네트워크 스택에 바인딩하면 호스트의 공격 표면이 증가합니다. 권한 없는 로컬 프로세스는 프로토콜로 소켓을 열어 시스템이 동적으로 프로토콜 처리기를 로드하게 할 수 있습니다.

프로시저

- 1 텍스트 편집기에서 `/etc/modprobe.conf.local` 파일을 엽니다.
- 2 이 파일에 다음 줄이 표시되는지 확인합니다.

```
install appletalk /bin/true
```

- 3 파일을 저장하고 닫습니다.

DECnet 프로토콜 보안

기본적으로 시스템에서 DECnet 프로토콜이 로드되지 못하도록 합니다. 잠재적 공격자는 이 프로토콜을 이용하여 시스템을 손상시킬 수 있습니다.

절대적으로 필요한 경우가 아니면 DECnet 프로토콜 모듈을 로드하지 않습니다. 이 프로토콜을 네트워크 스택에 바인딩하면 호스트의 공격 표면이 증가합니다. 권한 없는 로컬 프로세스는 프로토콜로 소켓을 열어 시스템이 동적으로 프로토콜 처리기를 로드하게 할 수 있습니다.

프로시저

- 1 텍스트 편집기에서 DECnet 프로토콜 `/etc/modprobe.conf.local` 파일을 엽니다.
- 2 이 파일에 다음 줄이 표시되는지 확인합니다.

```
install decnet /bin/true
```

- 3 파일을 저장하고 닫습니다.

Firewire 모듈 보안

기본적으로 시스템에서 Firewire 모듈이 로드되지 못하도록 합니다. 잠재적 공격자는 이 프로토콜을 이용하여 시스템을 손상시킬 수 있습니다.

절대적으로 필요한 경우가 아니면 Firewire 모듈을 로드하지 않습니다.

프로시저

- 1 텍스트 편집기에서 `/etc/modprobe.conf.local` 파일을 엽니다.
- 2 이 파일에 다음 줄이 표시되는지 확인합니다.

```
install ieee1394 /bin/true
```

- 3 파일을 저장하고 닫습니다.

IaaS(Infrastructure as a Service) 구성 요소 보안

시스템을 강화하는 경우 vRealize Automation IaaS(Infrastructure as a Service) 구성 요소 및 해당 호스트 시스템을 보안하여 잠재적 공격자가 이를 이용하지 못하도록 합니다.

vRealize Automation IaaS(Infrastructure as a Service) 구성 요소 및 해당 구성 요소가 상주하는 호스트에 대한 보안 설정을 구성해야 합니다. 기타 관련 구성 요소 및 애플리케이션의 구성을 설정하거나 확인해야 합니다. 경우에 따라 기존 설정을 확인할 수 있으며 그렇지 않으면 적절한 구성을 위한 설정을 변경하거나 추가해야 합니다.

Windows 시간 서비스를 사용하지 않도록 설정

보안 모범 사례로 vRealize Automation 운영 환경에서는 호스트 시간 동기화보다는 인증된 시간 서버를 사용합니다.

운영 환경에서 사용자 작업을 정확하게 추적하고 발생 가능한 악의적인 공격 및 침입을 감사 및 로깅을 통해 식별하려면 호스트 시간 동기화를 사용 안 함으로 설정하고 인증된 시간 서버를 사용하는 것이 좋습니다.

전송 중인 Infrastructure as a Service 데이터에 대한 TLS 구성

vRealize Automation 배포에서 강력한 TLS 프로토콜을 사용하여 Infrastructure as a Service 구성 요소에 대한 전송 채널을 보호하는지 확인하십시오.

SSL(Secure Sockets Layer) 및 보다 최근에 개발된 TLS(전송 계층 보안)는 서로 다른 시스템 구성 요소 간의 네트워크 통신 중에 시스템 보안을 보장하도록 도와주는 암호화 프로토콜입니다. SSL은 오래된 표준이기 때문에 다수의 SSL 구현 사항이 더 이상 잠재적인 공격에 대해 충분한 보안을 제공하지 못합니다. SSLv2 및 SSLv3을 비롯한 이전 SSL 프로토콜에서 심각한 약점이 확인되었습니다. 이러한 프로토콜은 더 이상 안전한 것으로 간주되지 않습니다.

조직의 보안 정책에 따라서 TLS 1.0도 사용하지 않도록 설정하는 것이 좋습니다.

참고 로드 밸런서에서 TLS를 종료하는 경우, SSLv2, SSLv3은 물론 필요한 경우 TLS 1.0과 같은 약한 프로토콜도 사용하지 않도록 설정하십시오.

인터넷 정보 서비스에서 SSLv3 사용 안 함

보안 모범 사례로 IaaS(Infrastructure as a Service) 호스트 서버 시스템의 IIS(인터넷 정보 서비스)에서 SSLv3을 사용 안 함으로 설정합니다.

프로시저

- 1 관리자 권한으로 Windows 레지스트리 편집기를 실행합니다.
- 2 레지스트리 창에서 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols`로 이동합니다.
- 3 **Protocols**를 마우스 오른쪽 버튼으로 클릭하고 **새로 만들기 > 키**를 선택합니다.
- 4 **SSL 3.0**을 입력합니다.
- 5 탐색 트리에서 새로 만든 **SSL 3.0** 키를 마우스 오른쪽 버튼으로 클릭하고, 팝업 메뉴에서 **새로 만들기 > 키**를 선택한 후 **Client**를 입력합니다.
- 6 탐색 트리에서 새로 만든 **SSL 3.0** 키를 마우스 오른쪽 버튼으로 클릭하고, 팝업 메뉴에서 **새로 만들기 > 키**를 선택한 후 **Server**를 입력합니다.
- 7 탐색 트리에서 SSL 3.0 아래에 있는 **Client**를 마우스 오른쪽 버튼으로 클릭하고 **새로 만들기 > DWORD(32비트) 값**을 선택한 후 **DisabledByDefault**를 입력합니다.
- 8 탐색 트리에서 SSL 3.0 아래에 있는 **Client**를 선택하고, 오른쪽 창의 **DisabledByDefault**를 두 번 클릭한 후 **1**을 입력합니다.
- 9 탐색 트리에서 SSL 3.0 아래에 있는 **Server**를 마우스 오른쪽 버튼으로 클릭하고 **새로 만들기 > DWORD(32비트) 값**을 선택한 후 **Enabled**를 입력합니다.
- 10 탐색 트리에서 SSL 3.0 아래에 있는 **Server**를 선택하고, 오른쪽 창에서 사용하도록 설정된 **DWORD**를 두 번 클릭한 후 **0**을 입력합니다.
- 11 Windows Server를 다시 시작합니다.

IaaS에 대해 TLS 1.0 사용 안 함

보안을 극대화하려면 풀링을 사용하고 TLS 1.0을 사용하지 않도록 IaaS를 구성합니다.

자세한 내용은 Microsoft 기술 자료 문서(<https://support.microsoft.com/en-us/kb/245030>)를 참조하십시오.

프로시저

- 1 웹 소켓 대신 풀링을 사용하도록 IaaS를 구성합니다.
 - a <appSettings> 섹션에 다음 값을 추가하여 Manager Service 구성 파일인 C:\Program Files (x86)\VMware\VCAC\Server\ManagerService.exe.config를 업데이트합니다.


```
<add key="Extensibility.Client.RetrievalMethod" value="Polling"/>
<add key="Extensibility.Client.PollingInterval" value="2000"/>
<add key="Extensibility.Client.PollingMaxEvents" value="128"/>
```
 - b Manager Service(VMware vCloud Automation Center 서비스)를 다시 시작합니다.
- 2 IaaS 서버에 대해 TLS 1.0이 사용 안 함으로 설정되었는지 확인합니다.
 - a 관리자 권한으로 레지스트리 편집기를 실행합니다.
 - b 레지스트리 창에서 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols로 이동합니다.
 - c Protocols를 마우스 오른쪽 버튼으로 클릭하고 **새로 만들기 > 키**를 선택한 후 **TLS 1.0**을 입력합니다.
 - d 탐색 트리에서 방금 만든 TLS 1.0 키를 마우스 오른쪽 버튼으로 클릭하고, 팝업 메뉴에서 **새로 만들기 > 키**를 선택한 후 **Client**를 입력합니다.
 - e 탐색 트리에서 방금 만든 TLS 1.0 키를 마우스 오른쪽 버튼으로 클릭하고, 팝업 메뉴에서 **새로 만들기 > 키**를 선택한 후 **Server**를 입력합니다.
 - f 탐색 트리에서 TLS 1.0 아래에 있는 **Client**를 마우스 오른쪽 버튼으로 클릭하고 **새로 만들기 > DWORD(32비트) 값**을 클릭한 후 **DisabledByDefault**를 입력합니다.
 - g 탐색 트리에서 TLS 1.0 아래에 있는 **Client**를 선택하고, 오른쪽 창의 **DisabledByDefault** DWORD를 두 번 클릭한 후 **1**을 입력합니다.
 - h 탐색 트리에서 TLS 1.0 아래에 있는 **Server**를 마우스 오른쪽 버튼으로 클릭하고 **새로 만들기 > DWORD(32비트) 값**을 선택한 후 **Enabled**를 입력합니다.
 - i 탐색 트리에서 TLS 1.0 아래에 있는 **Server**를 선택하고, 오른쪽 창의 **Enabled** DWORD를 두 번 클릭한 후 **0**을 입력합니다.
 - j Windows Server를 다시 시작합니다.

TLS 암호 그룹 구성

최상의 보안을 위해 강력한 암호를 사용하도록 vRealize Automation 구성 요소를 구성해야 합니다. 서버와 브라우저 사이에 협상되는 암호화 암호는 TLS 세션에 사용되는 암호화 강도를 결정합니다. 강력한 암호만 선택되도록 하려면 vRealize Automation 구성 요소에서 약한 암호를 사용하지 않도록 설정합니다. 강력한 암호만 지원하고 충분한 키 크기를 사용하도록 서버를 구성합니다. 또한 모든 암호를 적합한 순서로 구성합니다.

허용되지 않는 암호 그룹

NULL 암호 그룹, aNULL 또는 eNULL과 같이 인증을 제공하지 않는 암호 그룹은 사용하지 않도록 설정합니다. 또한 익명 Diffie-Hellman 키 교환(ADH), 내보내기 수준 암호(EXP, DES를 포함하는 암호), 페이로드 트래픽 암호화에 대한 128비트 미만의 키 크기, 페이로드 트래픽에 대한 해시 메커니즘으로 MD5 사용, IDEA 암호화 그룹 및 RC4 암호화 그룹을 사용하지 않도록 설정합니다. 또한 Diffie-Hellman(DHE) 키 교환을 사용하는 암호 그룹을 사용하지 않도록 설정되었는지 확인합니다.

호스트 서버 보안 확인

보안 모범 사례로 IaaS(Infrastructure as a Service) 호스트 서버 시스템의 보안 구성을 확인합니다.

Microsoft는 호스트 서버 시스템에서 보안을 확인하는 데 도움이 되는 여러 도구를 제공합니다. 이러한 도구의 가장 적절한 사용에 대한 지침은 Microsoft 벤더에 문의하십시오.

호스트 서버 보안 기준선 확인

MBSA(Microsoft Baseline Security Analyzer)를 실행하여 서버에 최신 업데이트 또는 핫 픽스가 있는지 신속하게 확인합니다. MBSA를 사용하여 Microsoft로부터 누락된 보안 패치를 설치함으로써 Microsoft 보안 권장 사항으로 서버를 최신 상태로 유지할 수 있습니다.

Microsoft 웹 사이트에서 최신 버전의 MBSA 도구를 다운로드합니다.

호스트 서버 보안 구성 확인

Windows SCW(보안 구성 마법사) 및 Microsoft SCM(보안 규정 준수 관리자) 툴킷을 사용하여 호스트 서버가 보안 구성되었는지 확인합니다.

Windows Server의 관리 도구에서 SCW를 실행합니다. 이 도구는 서버의 역할과 설치된 기능(네트워킹, Windows 방화벽 및 레지스트리 설정 등)을 식별할 수 있습니다. 보고서를 Windows Server에 대한 관련 SCM의 최신 강화 지침과 비교합니다. 결과를 기반으로 네트워크 서비스, 계정 설정 및 Windows 방화벽과 같은 각 기능에 대한 보안 설정을 미세 조정하고 설정을 서버에 적용할 수 있습니다.

Microsoft Technet 웹 사이트에서 SCW 도구에 대한 자세한 정보를 찾을 수 있습니다.

애플리케이션 리소스 보호

보안 모범 사례로 모든 관련 Infrastructure as a Service 파일의 사용 권한이 적절한지 확인합니다.

Infrastructure as a Service 설치를 기준으로 Infrastructure as a Service 파일을 검토합니다. 대부분의 경우 모든 폴더의 하위 폴더 및 파일의 설정은 해당 상위 폴더의 설정과 동일해야 합니다.

디렉토리 또는 파일	그룹 또는 사용자	모든 권한		읽기 및 실행		
		수정	수정	읽기	쓰기	쓰기
VMware\VCAC\Agents \<agent_name> \logs	시스템	X	X	X	X	X
	관리자	X	X	X	X	X
	관리자	X	X	X	X	X

불필요한 포트 및 서비스 사용 안 함

호스트 서버의 방화벽에서 열린 포트 목록을 검토합니다. IaaS 구성 요소 또는 중요한 시스템 작업에 필요하지 않은 포트를 모두 차단합니다. [“포트 및 프로토콜 구성,”](#) (61 페이지). 호스트 서버에서 실행 중인 서비스에 대해 감사를 수행하고 필요하지 않은 서비스는 사용하지 않도록 설정합니다.

호스트 네트워크 보안 구성

알려진 보안 위협에 대해 최상의 보호를 제공하려면 모든 VMware 호스트 시스템에 네트워크 인터페이스 및 통신 설정을 구성하십시오.

포괄적인 보안 계획의 일환으로 설정된 보안 지침에 따라 VMware 가상 장치 및 Infrastructure as a Service 구성 요소에 대한 네트워크 인터페이스 보안 설정을 구성하십시오.

이 장에서는 다음 주제에 대해 설명합니다.

- “VMware 장치에 대한 네트워크 설정 구성,” (51 페이지)
- “Infrastructure as a Service 호스트에 대한 네트워크 설정 구성,” (61 페이지)
- “포트 및 프로토콜 구성,” (61 페이지)

VMware 장치에 대한 네트워크 설정 구성

VMware 가상 장치 호스트 시스템이 안전하고 필요한 통신만 지원하도록 하려면 해당 네트워크 통신 설정을 검토하고 편집합니다.

VMware 호스트 시스템의 네트워크 IP 프로토콜 구성을 검사하고 보안 지침에 따라 네트워크 설정을 구성합니다. 필요하지 않은 모든 통신 프로토콜을 사용하지 않도록 설정합니다.

사용자의 네트워크 인터페이스 제어 방지

보안 모범 사례로 VMware 장치 호스트 시스템에서 자신의 작업을 수행하는 데 필요한 시스템 권한만 사용자에게 허용해야 합니다.

네트워크 인터페이스를 조작할 수 있는 권한을 사용자 계정에 허용하면 네트워크 보안 메커니즘 생략 또는 서비스 거부 문제가 발생할 수 있습니다. 네트워크 인터페이스 설정을 변경할 수 있는 기능은 권한 있는 사용자만 사용할 수 있도록 제한해야 합니다.

프로시저

- 1 각 VMware 장치 호스트 시스템에서 다음 명령을 실행합니다.


```
# grep -i '^USERCONTROL=' /etc/sysconfig/network/ifcfg*
```
- 2 각 인터페이스가 NO로 설정되었는지 확인합니다.

TCP 백로그 대기열 크기 설정

악의적 공격에 대한 일정 수준의 방어를 제공하려면 VMware 장치 호스트 시스템에서 기본 TCP 백로그 대기열 크기를 구성합니다.

TCP 백로그 대기열 크기를 적절한 기본 크기로 설정하여 TCP 서비스 거부 공격에 대한 완화를 제공합니다. 권장되는 기본 설정은 1280입니다.

프로시저

- 1 각 VMware 장치 호스트 시스템에서 다음 명령을 실행합니다.

```
# cat /proc/sys/net/ipv4/tcp_max_syn_backlog
```
- 2 텍스트 편집기에서 `/etc/sysctl.conf` 파일을 엽니다.
- 3 다음 항목을 파일에 추가하여 기본 TCP 백로그 대기열 크기를 설정합니다.

```
net.ipv4.tcp_max_syn_backlog=1280
```
- 4 변경 사항을 저장하고 파일을 닫습니다.

브로드캐스트 주소에 대한 ICMPv4 에코 거부

보안 모범 사례로 VMware 장치 호스트 시스템이 ICMP 브로드캐스트 주소 에코 요청을 무시하는지 확인합니다.

브로드캐스트 ICMP(Internet Control Message Protocol) 에코에 대한 응답은 증폭 공격에 대한 공격 벡터를 제공하고 악성 에이전트의 네트워크 매핑을 용이하게 할 수 있습니다. 장치 호스트 시스템이 ICMPv4 에코를 무시하도록 구성되면 해당 공격으로부터 시스템을 보호할 수 있습니다.

프로시저

- 1 VMware 가상 장치 호스트 시스템에서 `# cat /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts` 명령을 실행하여 시스템이 IPv4 브로드캐스트 주소 에코 요청을 거부하는지 확인합니다.
 호스트 시스템이 IPv4 리디렉션을 거부하도록 구성되어 있는 경우, 이 명령은 `/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts`에 대해 0 값을 반환합니다.
- 2 가상 장치 호스트 시스템이 ICMPv4 브로드캐스트 주소 에코 요청을 거부하도록 구성하려면, Windows 호스트 시스템의 `/etc/sysctl.conf` 파일을 텍스트 편집기에서 엽니다.
- 3 `net.ipv4.icmp_echo_ignore_broadcasts=0` 이라고 쓰여 있는 항목을 찾습니다. 이 항목의 값이 0으로 설정되어 있지 않거나 해당 항목이 없는 경우 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.
- 4 변경 사항을 저장하고 파일을 닫습니다.

IPv4 프록시 ARP 사용 안 함

VMware 장치 호스트 시스템에서 필요한 경우가 아니면 무단 정보 공유를 방지하기 위해 IPv4 프록시 ARP를 사용 안 함으로 설정했는지 확인합니다.

IPv4 프록시 ARP를 통해 시스템은 특정 인터페이스에서 다른 인터페이스에 연결된 호스트를 대신하여 ARP 요청에 대해 응답을 보낼 수 있습니다. 필요하지 않은 경우에는 연결된 네트워크 세그먼트 사이의 주소 정보 유출을 방지하기 위해 이 기능을 사용 안 함으로 설정하십시오.

프로시저

- 1 VMware 가상 장치 호스트 시스템에서 `# grep [01] /proc/sys/net/ipv4/conf/*/proxy_arp | grep "default|all"` 명령을 실행하여 IPv4 프록시 ARP가 사용하지 않도록 설정되었는지 확인합니다.
 호스트 시스템에서 IPv6 프록시 ARP가 사용 안 함으로 설정되어 있으면 이 명령이 0을 값으로 반환합니다.

```
/proc/sys/net/ipv4/conf/all/proxy_arp:0
/proc/sys/net/ipv4/conf/default/proxy_arp:0
```

 호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.
- 2 호스트 시스템에 IPv6 프록시 ARP를 구성해야 하면 텍스트 편집기에서 `/etc/sysctl.conf` 파일을 엽니다.

- 3 다음 항목을 확인합니다.

```
net.ipv4.conf.default.proxy_arp=0
net.ipv4.conf.all.proxy_arp=0
```

해당 항목이 없거나 해당 값이 0으로 설정되어 있지 않으면 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.

- 4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

IPv4 ICMP Redirect 메시지 거부

보안 모범 사례로 VMware 가상 장치 호스트 시스템에서 IPv4 ICMP Redirect 메시지를 거부하는지 확인합니다.

라우터는 ICMP Redirect 메시지를 사용하여 대상에 보다 직접적인 경로가 있다는 사실을 호스트에 알려줍니다. 악성 ICMP Redirect 메시지는 메시지 가로채기 공격을 용이하게 만들 수 있습니다. 이러한 메시지는 호스트의 경로 테이블을 수정하며 인증되지 않습니다. 달리 필요한 경우가 아니라면 시스템에서 이러한 메시지를 무시하도록 구성해야 합니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv4/conf/*/accept_redirects | grep "default|all" 명령을 실행하여 시스템이 IPv4 리디렉션 메시지를 거부하는지 확인합니다.

호스트 시스템이 IPv4 리디렉션을 거부하도록 구성되어 있는 경우, 이 명령은 다음을 반환합니다.

```
/proc/sys/net/ipv4/conf/all/accept_redirects:0
/proc/sys/net/ipv4/conf/default/accept_redirects:0
```

- 2 가상 장치 호스트 시스템이 IPv4 리디렉션 메시지를 거부하도록 구성하려면 /etc/sysctl.conf 파일을 텍스트 편집기에서 엽니다.

- 3 net.ipv4.conf로 시작하는 줄의 값을 확인합니다.

다음 항목의 값이 0으로 설정되어 있지 않거나 해당 항목이 없는 경우 항목을 파일에 추가하거나 기존 항목을 이에 맞게 업데이트합니다.

```
net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.default.accept_redirects=0
```

- 4 변경한 사항을 저장하고 파일을 닫습니다.

IPv6 ICMP redirect 메시지 거부

보안 모범 사례로 VMware 가상 장치 호스트 시스템이 IPv6 ICMP redirect 메시지를 거부하는지 확인합니다.

라우터는 ICMP Redirect 메시지를 사용하여 대상에 보다 직접적인 경로가 있다는 사실을 호스트에 알려줍니다. 악성 ICMP Redirect 메시지는 메시지 가로채기 공격을 용이하게 만들 수 있습니다. 이러한 메시지는 호스트의 경로 테이블을 수정하며 인증되지 않습니다. 필요한 경우가 아니면 이러한 메시지를 무시하도록 시스템이 구성되었는지 확인합니다.

프로시저

- 1 VMware 가상 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv6/conf/*/accept_redirects | grep "default|all" 명령을 실행하여 시스템에서 IPv6 리디렉션 메시지를 거부하는지 확인합니다.

IPv6 리디렉션을 거부하도록 호스트 시스템이 구성되어 있으면 이 명령이 다음을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/accept_redirects:0
/proc/sys/net/ipv6/conf/default/accept_redirects:0
```

- 2 IPv4 리디렉션 메시지를 거부하도록 가상 장치 호스트 시스템을 구성하려면 `/etc/sysctl.conf` 파일을 텍스트 편집기에서 엽니다.
- 3 `net.ipv6.conf`로 시작하는 줄의 값을 확인합니다.
파일에서 다음 항목의 값이 0으로 설정되지 않았거나 항목 자체가 없으면 파일에 해당 항목을 추가하거나, 기존 항목을 적절하게 업데이트합니다.

```
net.ipv6.conf.all.accept_redirects=0
net.ipv6.conf.default.accept_redirects=0
```
- 4 변경 사항을 저장하고 파일을 닫습니다.

IPv4 Martian 패킷 기록

보안 모범 사례로 VMware 가상 장치 호스트 시스템이 IPv4 Martian 패킷을 기록하는지 확인합니다.

Martian 패킷에는 시스템에서 잘못된 것으로 알고 있는 주소가 포함됩니다. 잘못된 구성 또는 진행 중인 공격을 식별할 수 있게 이러한 메시지를 기록하도록 호스트 시스템을 구성하십시오.

프로시저

- 1 VMware 장치 호스트 시스템에서 `# grep [01] /proc/sys/net/ipv4/conf/*/log_martians | grep "default|all"` 명령을 실행하여 시스템이 IPv4 Martian 패킷을 기록하는지 확인합니다.
Martian 패킷을 기록하도록 구성된 가상 시스템에서는 다음을 반환합니다.

```
/proc/sys/net/ipv4/conf/all/log_martians:1
/proc/sys/net/ipv4/default/log_martians:1
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.
- 2 IPv4 martian 패킷을 기록하도록 가상 시스템을 구성해야 할 경우에는 텍스트 편집기에서 `/etc/sysctl.conf` 파일을 엽니다.
- 3 `net.ipv4.conf`로 시작하는 줄의 값을 확인합니다.
다음 항목의 값이 1로 설정되지 않았거나 항목 자체가 없으면 파일에 해당 항목을 추가하거나, 기존 항목을 적절하게 업데이트합니다.

```
net.ipv4.conf.all.log_martians=1
net.ipv4.conf.default.log_martians=1
```
- 4 변경 사항을 저장하고 파일을 닫습니다.

IPv4 역방향 경로 필터링 사용

보안 모범 사례로 VMware 가상 장치 호스트 시스템에서 IPv4 역방향 경로 필터링을 사용하는지 확인합니다.

역방향 경로 필터링은 시스템에서 경로가 없는 소스 주소 또는 원본 인터페이스를 가리키지 않는 경로가 있는 소스 주소가 포함된 패킷을 삭제하도록 하여 스푸핑된 소스 주소로부터 시스템을 보호합니다. 가능할 때마다 역방향 경로 필터링을 사용하도록 호스트 시스템을 구성하십시오. 시스템 역할에 따라서 역방향 경로 필터링으로 인해 시스템이 정당한 트래픽을 삭제하는 경우가 있습니다. 이러한 문제가 발생하는 경우에는 더 허용되는 모드를 사용하거나 역방향 경로 필터링을 사용하지 않도록 함께 설정해야 할 수 있습니다.

프로시저

- 1 VMware 가상 장치 호스트 시스템에서 # `grep [01] /proc/sys/net/ipv4/conf/*/rp_filter | grep "default|all"` 명령을 실행하여 IPv4 역방향 경로 필터링을 사용하는지 확인합니다.
가상 시스템에서 IPv4 역방향 경로 필터링을 사용하는 경우 이 명령은 다음을 반환합니다.

```
/proc/sys/net/ipv4/conf/all/rp_filter:1
/proc/sys/net/ipv4/conf/default/re_filter:1
```

가상 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.
- 2 호스트 시스템에 IPv4 역방향 경로 필터링을 구성해야 하는 경우 `/etc/sysctl.conf` 파일을 텍스트 편집기에서 엽니다.
- 3 `net.ipv4.conf`로 시작하는 줄의 값을 확인합니다.
다음 항목의 값이 1로 설정되어 있지 않거나 해당 항목이 없는 경우 항목을 파일에 추가하거나 기존 항목을 이에 맞게 업데이트합니다.

```
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.default.rp_filter=1
```
- 4 변경 사항을 저장하고 파일을 닫습니다.

IPv4 전달 거부

VMware 장치 호스트 시스템이 IPv4 전달을 거부하는지 확인합니다.

시스템이 IP 전달을 위해 구성되어 있지만 지정된 라우터가 아닌 경우, 공격자는 이 시스템을 사용하여 네트워크 디바이스에서 필터링되지 않은 통신 경로를 제공하여 네트워크 보안을 무시할 수 있습니다. 이러한 위험을 방지하기 위해 가상 장치 호스트 시스템이 IPv4 전달을 거부하도록 구성하십시오.

프로시저

- 1 VMware 장치 호스트 시스템에서 # `cat /proc/sys/net/ipv4/ip_forward` 명령을 실행하여 시스템이 IPv4 전달을 거부하는지 확인합니다.
호스트 시스템이 IPv4 전달을 거부하도록 구성되어 있는 경우, 이 명령은 `/proc/sys/net/ipv4/ip_forward`에 대해 0 값을 반환합니다. 가상 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.
- 2 가상 장치 호스트 시스템이 IPv4 전달을 거부하도록 구성하려면 `/etc/sysctl.conf` 파일을 텍스트 편집기에서 엽니다.
- 3 `net.ipv4.ip_forward=0`이라고 쓰여 있는 항목을 찾습니다. 이 항목의 값이 현재 0으로 설정되어 있지 않거나 해당 항목이 없는 경우 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.
- 4 변경 사항이 있으면 저장하고 파일을 닫습니다.

IPv6 전달 거부

보안 모범 사례로 VMware 장치 호스트 시스템이 IPv6 전달을 거부하는지 확인합니다.

시스템이 IP 전달을 위해 구성되어 있지만 지정된 라우터가 아닌 경우, 공격자는 이 시스템을 사용하여 네트워크 디바이스에서 필터링되지 않은 통신 경로를 제공하여 네트워크 보안을 무시할 수 있습니다. 이러한 위험을 방지하기 위해 가상 장치 호스트 시스템이 IPv6 전달을 거부하도록 구성하십시오.

프로시저

- 1 VMware 장치 호스트 시스템에서 # `grep [01] /proc/sys/net/ipv6/conf/*/forwarding | grep "default | all"` 명령을 실행하여 시스템이 IPv6 전달을 거부하는지 확인합니다.

호스트 시스템이 IPv6 전달을 거부하도록 구성되어 있는 경우, 이 명령은 다음을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/forwarding:0
/proc/sys/net/ipv6/conf/default/forwarding:0
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.

- 2 호스트 시스템이 IPv6 전달을 거부하도록 구성하려면 `/etc/sysctl.conf` 파일을 텍스트 편집기에서 엽니다.
- 3 `net.ipv6.conf`로 시작하는 줄의 값을 확인합니다.

다음 항목의 값이 0으로 설정되어 있지 않거나 해당 항목이 없는 경우 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.

```
net.ipv6.conf.all.accept_redirects=0
net.ipv6.conf.default.accept_redirects=0
```

- 4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

IPv4 TCP Syncookie 사용

VMware 장치 호스트 시스템이 IPv4 TCP Syncookie를 사용하는지 확인합니다.

TCP SYN 플러드 공격은 시스템의 TCP 연결 테이블을 SYN_RCVD 상태의 연결로 채워 서비스 거부를 초래할 수 있습니다. Syncookie는 후속 ACK를 수신할 때까지 연결 추적을 방지하여 이니시에이터가 유효한 연결을 시도하고 있으며 플러드 소스가 아님을 확인합니다. 이 기술은 완전한 표준 준수 방식으로 작동하지 않지만 플러드 조건 동안에만 활성화되며 계속해서 유효한 요청을 서비스하는 동안 시스템 방어를 허용합니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # `cat /proc/sys/net/ipv4/tcp_syncookies` 명령을 실행하여 해당 시스템이 IPv4 TCP Syncookie를 사용하는지 확인합니다.

호스트 시스템이 IPv4 포워딩을 거부하도록 구성된 경우 이 명령은 `/proc/sys/net/ipv4/tcp_syncookies`에 대해 값 1을 반환합니다. 가상 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.

- 2 IPv4 TCP Syncookie를 사용하도록 가상 장치를 구성해야 하는 경우 텍스트 편집기에서 `/etc/sysctl.conf`를 엽니다.

- 3 `net.ipv4.tcp_syncookies=1`이라고 쓰여 있는 항목을 찾습니다.

이 항목의 값이 현재 1로 설정되어 있지 않거나 해당 항목이 없는 경우 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.

- 4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

IPv6 라우터 알림 거부

VMware 호스트 시스템이 시스템 운영에 달리 필요하지 않는 한 라우터 알림 및 ICMP Redirect 수락을 거부하는지 확인하십시오.

IPv6를 사용하면 시스템이 네트워크의 정보를 자동으로 사용하여 네트워킹 디바이스를 구성하는 것이 가능합니다. 보안의 관점에서 중요한 구성 정보를 인증되지 않는 방식으로 네트워크에서 수락하기보다는 수동으로 구성하는 것이 더 좋습니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv6/conf/*/accept_ra | grep "default|all" 명령을 실행하여 시스템이 라우터 알림을 거부하는지 확인합니다.

호스트 시스템이 IPv6 라우터 알림을 거부하도록 구성되어 있는 경우, 이 명령은 0 값을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/accept_ra:0
/proc/sys/net/ipv6/conf/default/accept_ra:0
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.

- 2 호스트 시스템이 IPv6 라우터 알림을 거부하도록 구성하려면 /etc/sysctl.conf 파일을 텍스트 편집기에서 엽니다.
- 3 다음 항목을 확인합니다.

```
net.ipv6.conf.all.accept_ra=0
net.ipv6.conf.default.accept_ra=0
```

이러한 항목이 없거나 해당 값이 0으로 설정되어 있지 않으면 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.

- 4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

IPv6 라우터 요청 거부

보안 모범 사례로 VMware 장치 호스트 시스템이 시스템 운영에 달리 필요하지 않는 한 IPv6 라우터 요청을 거부하는지 확인합니다.

라우터 요청 설정은 인터페이스를 활성화할 때 보낼 라우터 요청의 수를 결정합니다. 주소가 정적으로 할당되면 요청을 보낼 필요가 없습니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv6/conf/*/router_solicitations | grep "default|all" 명령을 실행하여 시스템이 IPv6 라우터 요청을 거부하는지 확인합니다.

호스트 시스템이 IPv6 라우터 알림을 거부하도록 구성되어 있는 경우, 이 명령은 다음을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/router_solicitations:0
/proc/sys/net/ipv6/conf/default/router_solicitations:0
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.

- 2 호스트 시스템이 IPv6 라우터 요청을 거부하도록 구성하려면 /etc/sysctl.conf 파일을 텍스트 편집기에서 엽니다.
- 3 다음 항목을 확인합니다.

```
net.ipv6.conf.all.router_solicitations=0
net.ipv6.conf.default.router_solicitations=0
```

해당 항목이 없거나 해당 값이 0으로 설정되어 있지 않으면 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.

- 4 변경 사항이 있으면 저장하고 파일을 닫습니다.

라우터 요청의 IPv6 라우터 기본 설정 거부

VMware 장치 호스트 시스템이 시스템 운영에 달리 필요하지 않는 한 IPv6 라우터 요청을 거부하는지 확인하십시오.

요청 설정의 라우터 기본 설정은 라우터 기본 설정을 결정합니다. 주소가 정적으로 할당되면 요청에 대한 라우터 기본 설정을 수신할 필요가 없습니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_rtr_pref | grep "default|all" 명령을 실행하여 시스템이 IPv6 라우터 요청을 거부하는지 확인합니다.
호스트 시스템이 IPv6 라우터 알림을 거부하도록 구성되어 있는 경우, 이 명령은 다음을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/accept_ra_rtr_pref:0
/proc/sys/net/ipv6/conf/default/accept_ra_rtr_pref:0
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.
- 2 호스트 시스템이 IPv6 라우터 요청을 거부하도록 구성하려면 /etc/sysctl.conf 파일을 텍스트 편집기에서 엽니다.
- 3 다음 항목을 확인합니다.

```
net.ipv6.conf.all.accept_ra_rtr_pref=0
net.ipv6.conf.default.accept_ra_rtr_pref=0
```

해당 항목이 없거나 해당 값이 0으로 설정되어 있지 않으면 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.
- 4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

IPv6 라우터 접두사 거부

VMware 장치 호스트 시스템이 시스템 운영에 달리 필요하지 않는 한 IPv6 라우터 접두사 정보를 거부하는지 확인하십시오.

accept_ra_pinfo 설정은 시스템이 라우터의 접두사 정보를 수락할지 여부를 제어합니다. 주소가 정적으로 할당되면 라우터 접두사 정보를 수신할 필요가 없습니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_pinfo | grep "default|all" 명령을 실행하여 시스템이 IPv6 라우터 접두사 정보를 거부하는지 확인합니다.
호스트 시스템이 IPv6 라우터 알림을 거부하도록 구성되어 있는 경우, 이 명령은 다음을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/accept_ra_pinfo:0
/proc/sys/net/ipv6/conf/default/accept_ra_pinfo:0
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.
- 2 호스트 시스템이 IPv6 라우터 접두사 정보를 거부하도록 구성하려면 /etc/sysctl.conf 파일을 텍스트 편집기에서 엽니다.
- 3 다음 항목을 확인합니다.

```
net.ipv6.conf.all.accept_ra_pinfo=0
net.ipv6.conf.default.accept_ra_pinfo=0
```

해당 항목이 없거나 해당 값이 0으로 설정되어 있지 않으면 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.
- 4 변경 사항이 있으면 저장하고 파일을 닫습니다.

IPv6 라우터 알림 홉 제한 설정 거부

VMware 장치 호스트 시스템이 필요한 경우가 아니면 IPv6 라우터 홉 제한 설정을 거부하는지 확인하십시오.

accept_ra_defrtr 설정은 시스템이 라우터 알림의 홉 제한 설정을 수락할지 여부를 제어합니다. 이 값을 0으로 설정하면 라우터가 송신 패킷에 대한 기본 IPv6 홉 제한을 변경할 수 없습니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_defrtr | grep "default|all" 명령을 실행하여 시스템이 IPv6 라우터 홉 제한 설정을 거부하는지 확인합니다.
호스트 시스템이 IPv6 라우터 홉 제한 설정을 거부하도록 구성되어 있는 경우, 이 명령은 0 값을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/accept_ra_defrtr:0
/proc/sys/net/ipv6/conf/default/accept_ra_defrtr:0
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.
- 2 호스트 시스템이 IPv6 라우터 홉 제한 설정을 거부하도록 구성하려면 /etc/sysctl.conf 파일을 텍스트 편집기에서 엽니다.
- 3 다음 항목을 확인합니다.

```
net.ipv6.conf.all.autoconf=0
net.ipv6.conf.default.autoconf=0
```

해당 항목이 없거나 해당 값이 0으로 설정되어 있지 않으면 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.
- 4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

IPv6 라우터 알림 Autoconf 설정 거부

VMware 장치 호스트 시스템이 필요한 경우를 제외하고 IPv6 라우터 autoconf 설정을 거부하는지 확인하십시오.

autoconf 설정은 라우터 알림으로 인해 시스템이 글로벌 유니캐스트 주소를 인터페이스에 할당할지 여부를 제어합니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv6/conf/*/autoconf | grep "default|all" 명령을 실행하여 시스템이 IPv6 라우터 autoconf 설정을 거부하는지 확인합니다.
호스트 시스템이 IPv6 라우터 autoconf 설정을 거부하도록 구성되어 있는 경우, 이 명령은 0 값을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/autoconf:0
/proc/sys/net/ipv6/conf/default/autoconf:0
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.
- 2 호스트 시스템이 IPv6 라우터 autoconf 설정을 거부하도록 구성하려면 /etc/sysctl.conf 파일을 텍스트 편집기에서 엽니다.
- 3 다음 항목을 확인합니다.

```
net.ipv6.conf.all.autoconf=0
net.ipv6.conf.default.autoconf=0
```

해당 항목이 없거나 해당 값이 0으로 설정되어 있지 않으면 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.
- 4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

IPv6 인접 라우터 요청 거부

VMware 장치 호스트 시스템이 필요한 경우를 제외하고 IPv6 인접 라우터 요청을 거부하는지 확인하십시오.

dad_transmits 설정은 인터페이스를 활성화할 때 원하는 주소가 네트워크에서 고유한지 확인하기 위해 주소(글로벌 또는 링크 로컬)당 보낼 인접 라우터 요청의 수를 결정합니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [01] /proc/sys/net/ipv6/conf/*/dad_transmits | grep "default|all" 명령을 실행하여 시스템이 IPv6 인접 라우터 요청을 거부하는지 확인합니다.

호스트 시스템이 IPv6 인접 라우터 요청을 거부하도록 구성되어 있는 경우, 이 명령은 0 값을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/dad_transmits:0
/proc/sys/net/ipv6/conf/default/dad_transmits:0
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.

- 2 호스트 시스템이 IPv6 인접 라우터 요청을 거부하도록 구성하려면 /etc/sysctl.conf 파일을 텍스트 편집기에서 엽니다.
- 3 다음 항목을 확인합니다.

```
net.ipv6.conf.all.dad_transmits=0
net.ipv6.conf.default.dad_transmits=0
```

해당 항목이 없거나 해당 값이 0으로 설정되어 있지 않으면 항목을 추가하거나 기존 항목을 이에 맞게 업데이트합니다.

- 4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

IPv6 최대 주소 제한

VMware 장치 호스트 시스템이 IPv6 최대 주소 설정을 시스템 작업에 필요한 최소값으로 제한하는지 확인합니다.

최대 주소 설정은 각 인터페이스에 사용할 수 있는 글로벌 유니캐스트 IPv6 주소 수를 결정합니다. 기본 값은 16이지만 시스템에 필요한 정적으로 구성된 글로벌 주소 수로 정확히 설정해야 합니다.

프로시저

- 1 VMware 장치 호스트 시스템에서 # grep [1] /proc/sys/net/ipv6/conf/*/max_addresses | grep "default|all" 명령을 실행하여 해당 시스템이 IPv6 최대 주소를 적절히 제한하는지 확인합니다.

호스트 시스템이 IPv6 최대 주소를 제한하도록 구성된 경우 이 명령은 값 1을 반환합니다.

```
/proc/sys/net/ipv6/conf/all/max_addresses:1
/proc/sys/net/ipv6/conf/default/max_addresses:1
```

호스트 시스템이 제대로 구성된 경우에는 추가 작업이 필요하지 않습니다.

- 2 호스트 시스템에서 IPv6 최대 주소를 구성해야 하는 경우 텍스트 편집기에서 /etc/sysctl.conf 파일을 엽니다.
- 3 다음 항목을 확인합니다.

```
net.ipv6.conf.all.max_addresses=1
net.ipv6.conf.default.max_addresses=1
```

항목이 없거나 해당 값이 1로 설정되지 않은 경우 항목을 추가하거나 기존 항목을 적절히 업데이트합니다.

4 변경한 사항이 있으면 저장하고 파일을 닫습니다.

Infrastructure as a Service 호스트에 대한 네트워크 설정 구성

보안 모범 사례로 VMware 요구 사항 및 지침에 따라 VMware IaaS(Infrastructure as a Service) 구성 요소 호스트 시스템에서 네트워크 통신 설정을 구성합니다.

적절한 보안과 함께 전체 vRealize Automation 기능을 지원하도록 IaaS(Infrastructure as a Service) 호스트 시스템의 네트워크 구성을 구성하십시오.

“IaaS(Infrastructure as a Service) 구성 요소 보안,” (44 페이지) 항목을 참조하십시오.

포트 및 프로토콜 구성

보안 모범 사례로 VMware 지침에 따라 모든 vRealize Automation 장치와 구성 요소의 포트 및 프로토콜을 구성합니다.

중요한 시스템 구성 요소가 운영 환경에서 작동하도록 vRealize Automation 구성 요소의 수신 및 송신 포트를 필요에 맞게 구성합니다. 불필요한 모든 포트와 프로토콜을 사용하지 않도록 설정합니다. vRealize Automation 참조 아키텍처를 참조하십시오.

사용자 필수 포트

보안 모범 사례로 VMware 지침에 따라 vRealize Automation 사용자 포트를 구성합니다.

필수 포트는 보안 네트워크를 통해서만 노출하십시오.

서버	포트
vRealize Automation 장치	443, 8443

관리자 필수 포트

보안 모범 사례로 VMware 지침에 따라 vRealize Automation 관리자 포트를 구성합니다.

필수 포트는 보안 네트워크를 통해서만 노출하십시오.

서버	포트
vRealize Application Services 서버	5480

vRealize Automation 장치 포트

보안 모범 사례로 VMware 권장 사항에 따라 vRealize Automation 장치에 대한 수신 및 송신 포트를 구성합니다.

수신 포트

vRealize Automation 장치에 대한 최소 필수 수신 포트를 구성합니다. 시스템 구성에 필요한 경우 선택적 포트를 구성합니다.

표 9-1. 필요한 최소 수신 포트

포트	프로토콜	설명
443	TCP	vRealize Automation 콘솔 및 API 호출에 액세스
8443	TCP	콘솔 프록시(VMRC)입니다.
5480	TCP	가상 장치 웹 관리 콘솔에 액세스합니다.
5488, 5489	TCP	내부용. 업데이트를 위해 vRealize Automation 장치에서 사용합니다.

표 9-1. 필요한 최소 수신 포트 (계속)

포트	프로토콜	설명
5672	TCP	RabbitMQ 메시징. 참고 vRealize Automation 장치 인스턴스를 클러스터하는 경우 오픈 포트 4369 및 25672를 구성해야 할 수 있습니다.
40002	TCP	vIDM 서비스에 필요합니다. 이는 HA 구성에서 추가하는 경우 기타 vRealize Automation 장치 노드에서의 트래픽을 제외하고 모든 외부 트래픽에 방화벽을 사용합니다.

필요한 경우 선택적 수신 포트를 구성합니다.

표 9-2. 선택적 수신 포트

포트	프로토콜	설명
22	TCP	(선택 사항) SSH입니다. 운영 환경에서 포트 22에서 수신하는 SSH 서비스를 사용하지 않도록 설정하고 포트 22를 닫습니다.
80	TCP	(선택 사항) 443으로 리디렉션됩니다.

송신 포트

필수 송신 포트를 구성합니다.

표 9-3. 필요한 최소 송신 포트

포트	프로토콜	설명
25,587	TCP, UDP	아웃바운드 알림 이메일 전송용 SMTP.
53	TCP, UDP	DNS.
67, 68, 546, 547	TCP, UDP	DHCP.
110, 995	TCP, UDP	인바운드 알림 이메일 수신용 POP.
143, 993	TCP, UDP	인바운드 알림 이메일 수신용 IMAP.
443	TCP	HTTPS를 통한 Infrastructure as a Service Manager Service.

필요한 경우 선택적 송신 포트를 구성합니다.

표 9-4. 선택적 송신 포트

포트	프로토콜	설명
80	TCP	(선택 사항) 소프트웨어 업데이트를 가져오는 데 사용. 업데이트를 별도로 다운로드하고 적용할 수 있습니다.
123	TCP, UDP	(선택 사항) 호스트 시간을 사용하는 대신 NTP에 직접 연결하는 데 사용.

Infrastructure as a Service 포트

보안 모범 사례로, VMware 지침에 따라 IaaS(Infrastructure as a Service) 구성 요소의 수신 및 송신 포트를 구성합니다.

수신 포트

IaaS 구성 요소에 대해 필요한 최소 수신 포트를 구성합니다.

표 9-5. 필요한 최소 수신 포트

구성 요소	포트	프로토콜	설명
Manager Service	443	TCP	HTTPS를 통한 IaaS 구성 요소 및 vRealize Automation 장치와의 통신. 프록시 에이전트가 관리하는 모든 가상화 호스트에도 수신 트래픽을 위한 TCP 포트 443이 열려 있어야 합니다.

송신 포트

IaaS 구성 요소에 대해 필요한 최소 송신 포트를 구성합니다.

표 9-6. 필요한 최소 송신 포트

구성 요소	포트	프로토콜	설명
모두	53	TCP, UDP	DNS.
모두		TCP, UDP	DHCP.
Manager Service	443	TCP	HTTPS를 통한 vRealize Automation 장치와의 통신.
웹 사이트	443	TCP	HTTPS를 통한 Manager Service와의 통신.
Distributed Execution Manager	443	TCP	HTTPS를 통한 Manager Service와의 통신.
프록시 에이전트	443	TCP	HTTPS를 통한 Manager Service 및 가상화 호스트와의 통신
게스트 에이전트	443	TCP	HTTPS를 통한 Manager Service와의 통신.
Manager Service, 웹 사이트	1433	TCP	MSSQL.

필요한 경우 선택적 송신 포트를 구성합니다.

표 9-7. 선택적 송신 포트

구성 요소	포트	프로토콜	설명
모두	123	TCP, UDP	NTP는 선택 사항입니다.

감사 및 로깅

보안 모범 사례로 VMware 권장 사항에 따라 vRealize Automation 시스템에 감사 및 로깅을 설정합니다.

중앙 로그 호스트에 원격 로깅을 사용하면 로그 파일을 위한 안전한 저장소가 제공됩니다. 로그 파일을 중앙 호스트에 모으면 단일 도구를 사용하여 환경을 모니터링할 수 있습니다. 또한 인프라 내 여러 엔티티에 대한 전략적인 공격과 같은 위협 증거에 대해 종합적인 분석 및 검색을 수행할 수 있습니다. 안전한 중앙 집중식 로그 서버에 로깅하면 로그 변조를 방지하는 데 도움이 되고 장기적인 감사 기록도 제공됩니다.

원격 로깅 서버의 보안 확인

공격자가 호스트 시스템의 보안을 침해한 후 로그 파일을 검색하고 변조하여 자신의 흔적을 감추고 몰래 시스템에 대한 제어를 유지하려고 시도하는 경우가 종종 있습니다. 원격 로깅 서버를 적절하게 보호하면 로그 변조를 방지하는 데 도움이 됩니다.

인증된 NTP 서버 사용

모든 호스트 시스템이 적절한 지역화 오프셋을 포함한 동일한 상대 시간 소스를 사용하며, 상대 시간 소스를 협정 세계시(UTC)와 같이 합의된 시간 표준과 연관시킬 수 있는지 확인합니다. 시간 소스에 대한 올바른 접근 방식을 사용하면 적절한 로그 파일을 검토하여 침입자의 활동을 신속하게 추적하고 상관관계를 파악할 수 있습니다. 시간 설정이 정확하지 않으면 로그 파일을 검사하고 연관시켜서 공격을 감지하기 어려워지고 감사가 부정확해질 수 있습니다.

외부 시간 소스의 NTP 서버를 3개 이상 사용하거나 신뢰할 수 있는 네트워크에 3개 이상의 외부 시간 소스로부터 차례로 시간을 가져오는 로컬 NTP 서버를 여러 개 구성합니다.

색인

A

Application Services 장치, 루트 암호 변경 **20**

B

Bluetooth 프로토콜 처리기 **41**

D

DCCP(Datagram Congestion Protocol) **42**

F

FIPS, 사용 **28**

H

HA 프록시, 약한 암호 사용 안 함 **32**

I

IaaS 호스트 **13**

IaaS(Infrastructure as a Service), 보안 **44**

IaaS(Infrastructure as a Service), 호스트 서버 보안 **47**

Infrastructure as a Service

네트워크 설정 구성 **61**

암호 구성 **47**

포트 **62**

Infrastructure as a Service, SSLv3 사용 안 함 **45**

Infrastructure as a Service, 보안 구성 **48**

Infrastructure as a Service, 애플리케이션 리소스 보호 **47**

Infrastructure as a Service, 전송 중인 데이터에 대한 TLS **45**

IPv4, 전달 거부 **55**

IPv4 전달 거부 **55**

IPv4, ICMP redirect 메시지 무시 **53**

IPv4, martian 패킷 기록 **54**

IPv4, TCP Syncookie 사용 **56**

IPv4, 역방향 경로 필터링 **54**

IPv4, 프록시 ARP 사용 안 함 **52**

IPv6

라우터 알람 autoconf 설정 거부 **59**

라우터 알람 거부 **56**

라우터 알람 홉 제한 설정 거부 **58**

라우터 요청의 라우터 기본 설정 거부 **57**

라우팅 요청 거부 **57**

인접 라우터 요청 거부 **60**

전달 거부 **55**

IPv6 라우터 기본 설정 **58**

IPv6 라우터 기본 설정 거부 **58**

IPv6 라우터 알람 Autoconf 설정 거부 **59**

IPv6 라우터 알람 거부 **56**

IPv6 라우터 알람 홉 제한 설정 거부 **58**

IPv6 라우팅 요청 거부 **57**

IPv6 인접 라우터 요청 거부 **60**

IPv6 전달 거부 **55**

IPv6, ICMP redirect 메시지 무시 **53**

L

Lighttpd 서버 응답 헤더 **40**

localhost 통신 **27**

M

Microsoft .NET, 강화 **14**

Microsoft .NET 강화 **14**

Microsoft 인터넷 정보 서비스 강화 **14**

P

PostgreSQL 클라이언트 인증 **35**

R

RabbitMQ 서비스, 약한 암호 사용 안 함 **34**

RDS(Reliable Datagram Sockets) 프로토콜 **43**

S

SSH 키 파일 사용 권한, 확인 **25**

SSLv3 사용 안 함 **45**

T

TCP Syncookie **56**

TCP 백로그 대기열 크기 **51**

tcserver 응답 헤더 **40**

TIPC(Transparent Inter-Process Communication) 프로토콜 **43**

TLS, localhost 통신 **27**

TLS 1.0, IaaS에 대해 사용 안 함 **46**

TLS 1.0, 사용 안 함 **30**

TLS, vRealize Automation 장치 **27**

U

USB 대용량 스토리지 처리기 **41**

V

VAMI, vRealize Automation 장치 사용 **22**

VMware Remote Console, 티켓 만료 기한 **38**

vRealize Appliance
약한 암호 사용 안 함 **33, 34**
포트 **61**

vRealize Appliance, 강력한 프로토콜 **29**

vRealize Appliance, 애플리케이션 리소스 보호 **36**

vRealize Automation, 감사 및 로깅 **65**

vSphere, 강화 **13**

W

Windows 시간 서비스, 사용 안 함 **45**

Windows 시간 서비스 사용 안 함 **45**

ㄱ

가상 시스템

IPv4 프록시 ARP 사용 안 함 **52**

IPv6 ICMP redirect 메시지 무시 **53**

사용자 제어 방지 **51**

가상 시스템, 브로드캐스트 주소에 대한 ICMPv4 에코 거부 **52**

가상 장치

Bluetooth 프로토콜 처리기 **41**

DCCP(Datagram Congestion Protocol) **42**

IPv4 ICMP redirect 메시지 무시 **53**

IPv4 Martian 패킷 기록 **54**

RDS(Reliable Datagram Sockets) 프로토콜 **43**

TIPC(Transparent Inter-Process Communication) 프로토콜 **43**

USB 대용량 스토리지 처리기 **41**

강력한 암호 구성 **32**

네트워크 설정 구성 **51**

네트워크 시간 프로토콜 구성 **26**

보안 계층 구성, vRealize Automation 장치 **27**

보안 구성 **19**

보안 셸 로컬 관리자 계정 **22**

보안 셸 사용 또는 사용 안 함 **22**

부팅 로더 인증 **26**

역방향 경로 필터링 **54**

콘솔 프록시 구성 **38**

가상 장치 관리 인터페이스, 사용자 변경 **25**

감사 **65**

강력한 암호를 사용하도록 vRealize 구성 요소 구성 **32**

강력한 프로토콜, vRealize Appliance **29**
강화, IaaS 호스트 **13**

강화, 개요 **9**

관리 계정 **21**

관리자 포트 **61**

구성

PostgreSQL 클라이언트 인증 **35**

네트워크 보안 및 보안 통신 **51**

암호 만료 기한 **20**

정보 공개 **39**

콘솔 프록시 **38**

구성, 가상 장치 보안 **19**

ㄴ

네트워크 보안 **51**

네트워크 설정 구성, Infrastructure as a Service **61**

네트워크 설정, 가상 장치 **51**

네트워크 시간 프로토콜 구성 **26**

ㄷ

라우터 요청의 IPv6 라우터 기본 설정 거부 **57**

로깅 **65**

루트 암호, Application Services 장치 **20**

루트 암호 기록 확인 **20**

루트 암호 기록, 확인 **20**

ㅂ

보안 구성 **19**

보안 권고 사항 및 패치 **17**

보안 셸 로컬 관리자 계정 **22**

보안 셸 서버 구성 **23**

보안 셸 액세스 제한 **23**

보안 셸 클라이언트 구성 **24**

보안 셸 키 파일 사용 권한 **25**

보안 셸, 관리 **21**

보안 셸, 사용 또는 사용 안 함 **22**

보안 통신 **51**

부팅 로더 인증 **26**

불필요한 소프트웨어 관리 **41**

브로드캐스트 주소에 대한 ICMPv4 에코 거부 **52**

ㅅ

사용자 제어 방지 **51**

사용자 필수 포트 **61**

서버 구성, 보안 셸 **23**

서버 응답 헤더 **40**

설치된 소프트웨어 **15**

세션 시간 초과, 설정 **41**

ㅇ

암호

- HA 프록시 **32**

- 가상 장치 구성 **32**

- 암호 만료 기한, 관리 **20**

- 암호, Infrastructure as a Service **47**

- 애플리케이션 리소스 **47**

- 애플리케이션 리소스 보호, vRealize

- Appliance **36**

- 약한 암호 사용 안 함, vCO 서비스 **33**

- 약한 암호 사용 안 함, vRealize

- Appliance **33**

- 약한 암호 사용 안 함, 콘솔 프록시 서비스 **33**

- 약한 암호를 사용 안 함, RabbitMQ 서비스 **34**

- 업데이트된 정보 **7**

- 인터넷 정보 서비스, 강화 **14**

- 인프라, 보안 권고 사항 및 패치 **17**

- 인프라 강화 **13**

- 인프라 강화, Microsoft SQL Server **14**

- 인프라 강화, vSphere **13**

- 인프라, 강화 **13**

ㅈ

- 정보 공개 **39, 40**

- 정보 공개, 인터넷 정보 서비스 **40**

ㅊ

- 콘솔 프록시 구성 **38**

- 콘솔 프록시 서버 포트 **38**

- 콘솔 프록시 서비스, 약한 암호 사용 안 함 **33**

- 클라이언트 구성, 보안 셀 **24**

ㅌ

- 티켓 만료 기한, VMware Remote Console **38**

ㅍ

- 파일 사용 권한, 보안 셀 **25**

포트

- 관리자 **61**

- 콘솔 프록시 서버 **38**

- 포트 및 프로토콜 **61**

- 포트, Infrastructure as a Service **62**

- 포트, vRealize Appliance **61**

ㅎ

- 호스트 서버 보안 확인 **47**

