PowerShell로 ONTAP 사이버 소산 생성, 강화 및 검증

NetApp Solutions

NetApp
September 23, 2024

목차

PowerShell로 ONTAP 사이버 소산 생성, 강화 및 검증

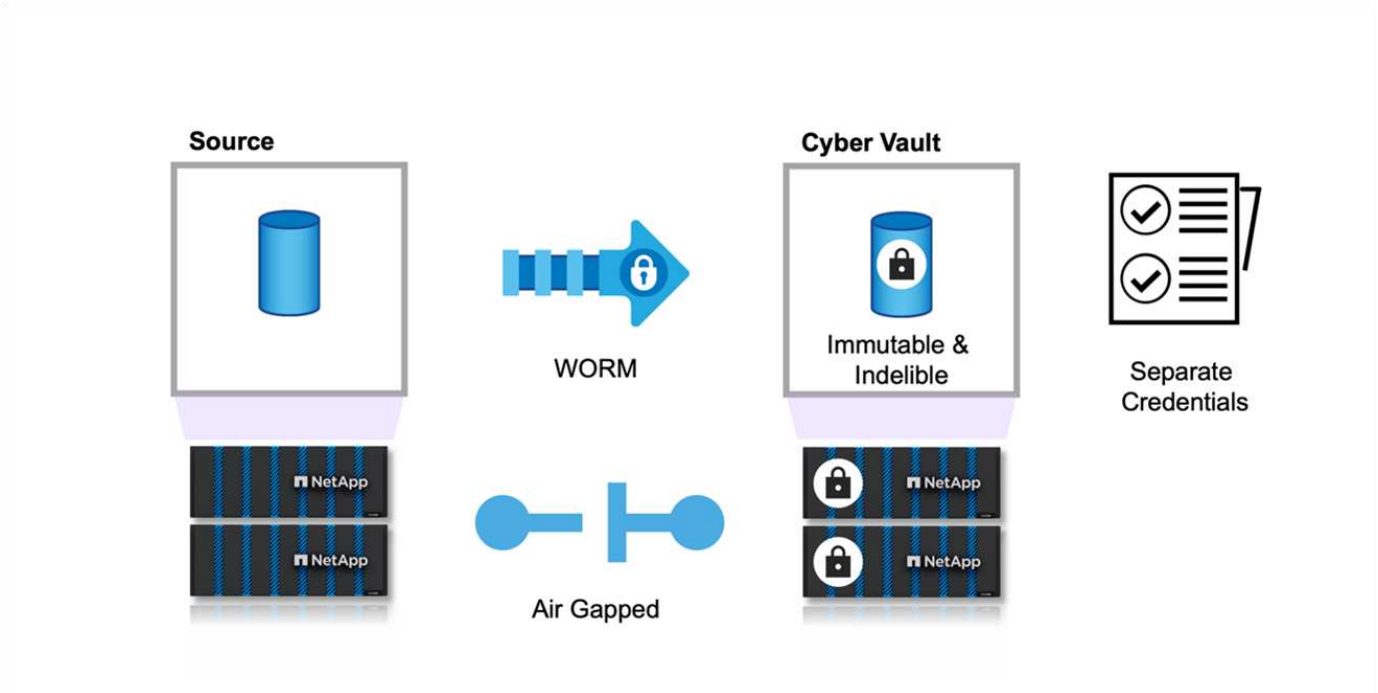
PowerShell을 사용한 ONTAP 사이버 소산 개요

오늘날의 디지털 환경에서 조직의 중요한 데이터 자산을 보호하는 것은 단순한 모범 사례가 아니라 비즈니스에 필수적인 요소입니다. 사이버 위협은 전례 없는 속도로 발전하고 있으며, 기존의 데이터 보호 수단으로는 중요한 정보를 안전하게 보호할 수 없습니다. NetApp의 최첨단 ONTAP 기반 솔루션은 첨단 에어 갭 기술과 강력한 데이터 보호 수단을 결합하여 사이버 위협에 대한 침투할 수 있는 장벽을 만듭니다. 보안 강화 기술로 가장 중요한 데이터를 격리함으로써 사이버 볼트는 공격 표면을 최소화하여 가장 중요한 데이터를 안전하게 보호하고 필요할 때 즉시 사용할 수 있도록 합니다.

사이버 저장소는 방화벽, 네트워킹, 스토리지와 같은 여러 보호 계층으로 구성된 보안 스토리지 시설입니다. 이러한 구성 요소는 중요한 비즈니스 운영에 필요한 중요한 복구 데이터를 보호합니다. 사이버 볼트의 구성 요소는 볼트 정책에 따라 필수 프로덕션 데이터와 정기적으로 동기화되지만 그렇지 않으면 액세스할 수 없습니다. 이러한 고립되고 단절된 설정은 사이버 공격이 프로덕션 환경을 손상시키는 경우 사이버 저장소에서 신뢰할 수 있는 최종 복구를 쉽게 수행할 수 있도록 합니다.

NetApp을 사용하면 네트워크를 구성하고, LIF를 비활성화하고, 방화벽 규칙을 업데이트하고, 시스템을 외부 네트워크 및 인터넷으로부터 격리하여 사이버 레질리언스 공간을 쉽게 생성할 수 있습니다. 이 강력한 접근 방식은 외부 네트워크와 인터넷으로부터 시스템을 효과적으로 분리하여 원격 사이버 공격과 무단 액세스 시도에 대한 탁월한 보호 기능을 제공함으로써 시스템이 네트워크 기반 위협 및 침입에 영향을 받지 않도록 합니다.

SnapLock Compliance 보호와 결합하면 ONTAP 관리자나 NetApp 지원에서도 데이터를 수정하거나 삭제할 수 없습니다. SnapLock는 SEC 및 FINRA 규정에 대한 정기적인 감사를 통해 데이터 복원성이 금융 업계의 엄격한 WORM 및 데이터 보존 규정을 준수하도록 보장합니다. NetApp는 NSA CSfC에서 가장 중요한 데이터를 저장하기 위해 검증된 유일한 엔터프라이즈 스토리지입니다.



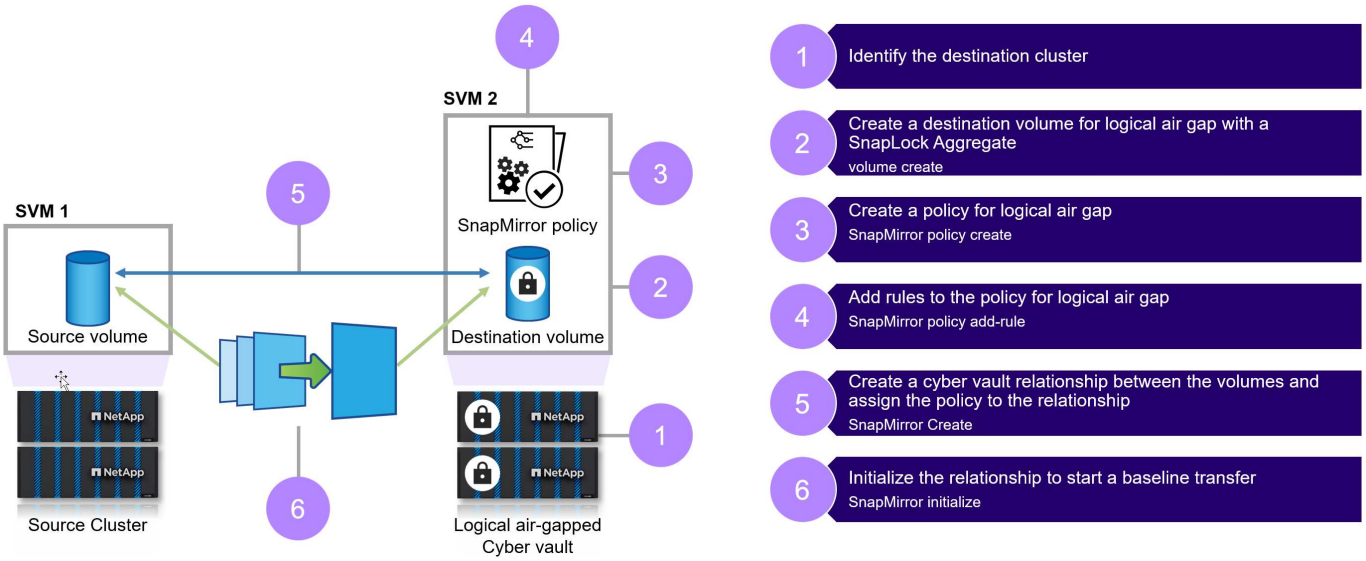
이 문서에서는 온프레미스 ONTAP 스토리지를 위한의 사이버 볼트를 변경 불가능한 스냅샷이 있는 다른 지정된

ONTAP 스토리지에 자동으로 구성하는 NetApp의 사이버 레질리언스 전략으로, 증가하는 사이버 공격으로부터 보호하는 계층을 추가하는 방법을 설명합니다. 이 아키텍처의 일부로 전체 구성은 ONTAP 모범 사례에 따라 적용됩니다. 마지막 섹션에는 공격 발생 시 복구를 수행하기 위한 지침이 나와 있습니다.

동일한 솔루션을 FSx for ONTAP를 사용하여 AWS에서 지정된 사이버 보관소를 생성할 수도 있습니다.

ONTAP 사이버 저장소를 만들기 위한 상위 단계

- 피어링 관계를 생성합니다
  - ONTAP 스토리지를 사용하는 운영 사이트는 지정된 사이버 소산 ONTAP 스토리지와 함께 피어링됩니다
- SnapLock Compliance 볼륨을 생성합니다
- 레이블을 설정할 SnapMirror 관계 및 규칙을 설정합니다
  - SnapMirror 관계 및 적절한 일정이 구성됩니다
- SnapMirror(볼트) 전송을 시작하기 전에 보존 설정을 지정합니다
  - 복제된 데이터에 Retention Lock이 적용되어 내부 데이터나 데이터 오류가 발생하지 않습니다. 이 옵션을 사용하면 보존 기간이 만료되기 전에 데이터를 삭제할 수 없습니다
  - 조직에서는 요구 사항에 따라 이 데이터를 몇 주/몇 개월 동안 보관할 수 있습니다
- 레이블을 기반으로 SnapMirror 관계를 초기화합니다
  - 초기 시드 및 증분 영구 전송은 SnapMirror 일정에 따라 수행됩니다
  - SnapLock Compliance로 데이터가 보호되며(변경 및 삭제가 불가함), 데이터가 복구에 사용 가능합니다
- 엄격한 데이터 전송 제어 구현
  - 사이버 볼트는 프로덕션 사이트의 데이터와 함께 제한된 기간 동안 잠금 해제되며 볼트의 데이터와 동기화됩니다. 전송이 완료되면 연결을 끊고 닫은 후 다시 잠깁니다
- 빠른 복구
  - 운영 사이트가 영향을 받는 경우 사이버 소산의 데이터를 원래 운영 환경 또는 다른 선택된 환경으로 안전하게 복구합니다



솔루션 구성 요소

소스 및 대상 클러스터에서 9.15.1을 실행하는 NetApp ONTAP

ONTAP One: NetApp ONTAP의 올인원 라이선스.

ONTAP One 라이센스에서 사용되는 기능:

- SnapLock 규정 준수
- SnapMirror를 참조하십시오
- 다중 관리 검증
- ONTAP에서 제공하는 모든 강화 기능
- 사이버 보관용으로 별도의 RBAC 자격 증명

All ONTAP 유니파이드 물리적 어레이를 사이버 소산에 사용할 수 있지만, AFF C-Series 용량 기반 플래시 시스템과 FAS 하이브리드 플래시 시스템은 이러한 목적에 가장 비용 효율적인 이상적인 플랫폼입니다. "ONTAP 사이버 소산 크기 조정"사이징 지침은 을 참조하십시오.

PowerShell로 ONTAP 사이버 소산 생성

기존 방법을 사용하는 공기 교환 백업에는 공간을 생성하고 운영 미디어와 보조 미디어를 물리적으로 분리하는 작업이 포함됩니다. 미디어를 오프사이트로 이동하거나 연결을 끊으면 악의적인 사용자가 데이터에 액세스할 수 없습니다. 이렇게 하면 데이터가 보호되지만 복구 시간이 느려질 수 있습니다. SnapLock Compliance를 사용할 경우 물리적인 분리가 필요하지 않습니다. SnapLock Compliance는 보관된 스냅샷 시점, 읽기 전용 복사본을 보호하여 데이터에 빠르게 액세스하고 삭제 또는 삭제로부터 안전하며 수정이나 변경 불가능한 상황에서도 데이터를 안전하게 보호합니다.

필수 구성 요소

이 문서의 다음 섹션에 있는 단계를 시작하기 전에 다음과 같은 사전 요구 사항이 충족되는지 확인하십시오.

- 소스 클러스터에서 ONTAP 9 이상이 실행 중이어야 합니다.
- 소스 및 타겟 애그리게이트는 64비트여야 합니다.
- 소스 및 타겟 클러스터를 내다봐야 합니다.
- 소스 및 타겟 SVM을 피어링해야 한다.
- 클러스터 피어링 암호화가 활성화되어 있는지 확인

ONTAP 사이버 저장소로 데이터 전송을 설정하려면 몇 단계를 거쳐야 합니다. 운영 볼륨에서 생성할 복제본과 복제본을 생성할 시기를 지정하는 스냅샷 정책을 구성하고 레이블을 할당하여 SnapVault에서 전송할 복제본을 지정합니다. 2차에서는 전송할 스냅샷 복사본의 레이블 및 사이버 볼트에 보관해야 할 복사본 수를 지정하는 SnapMirror 정책을 생성해야 합니다. 이러한 정책을 구성한 후 SnapVault 관계를 생성하고 전송 일정을 설정합니다.

이 문서에서는 운영 스토리지 및 지정된 ONTAP 사이버 볼트가 이미 설정 및 구성되어 있다고 가정합니다.

사이버 소산 클러스터는 소스 데이터와 동일하거나 다른 데이터 센터에 있을 수 있습니다.

ONTAP 사이버 저장소를 만드는 단계입니다

1. ONTAP CLI 또는 System Manager를 사용하여 규정 준수 클럭을 초기화합니다.
2. SnapLock Compliance를 사용하도록 설정한 데이터 보호 볼륨을 생성합니다.
3. SnapMirror create 명령을 사용하여 SnapVault 데이터 보호 관계를 생성합니다.
4. 대상 볼륨에 대한 기본 SnapLock Compliance 보존 기간을 설정합니다.

기본 보존은 "최소값으로 설정"입니다. 볼트 대상인 SnapLock 볼륨에 기본 보존 기간이 할당되어 있습니다. 이 기간의 값은 SnapLock Compliance 볼륨의 경우 처음에는 최소 0년, 최대 30년으로 설정됩니다. 각 NetApp 스냅샷 복사본은 처음에 이 기본 보존 기간을 사용하여 커밋됩니다. 보존 기간은 나중에 필요할 경우 연장할 수 있지만 줄일 수는 없습니다.

위 항목에는 수동 단계가 포함되어 있습니다. 보안 전문가는 수동 관리를 방지하기 위해 프로세스를 자동화함으로써 오류 발생 시 큰 이윤을 발생시킬 것을 권고합니다. 다음은 SnapLock Compliance의 사전 요구 사항과 구성 및 클록 초기화를 완전히 자동화하는 코드 조각입니다.

다음은 ONTAP 규정 준수 클록을 초기화하는 PowerShell 코드 예제입니다.

```
function initializeSnapLockComplianceClock {
    try {
        $nodes = Get-NcNode

        $isInitialized = $false
        logMessage -message "Cheking if snaplock compliance clock is
initialized"
        foreach($node in $nodes) {
            $check = Get-NcSnaplockComplianceClock -Node $node.Node
            if ($check.SnaplockComplianceClockSpecified -eq "True") {
                $isInitialized = $true
            }
        }

        if ($isInitialized) {
            logMessage -message "SnapLock Compliance clock already
initialized" -type "SUCCESS"
        } else {
            logMessage -message "Initializing SnapLock compliance clock"
            foreach($node in $nodes) {
                Set-NcSnaplockComplianceClock -Node $node.Node
            }
            logMessage -message "Successfully initialized SnapLock
Compliance clock" -type "SUCCESS"
        }
    } catch {
        handleError -errorMessage $_.Exception.Message
    }
}
```

다음은 ONTAP 사이버 볼트를 구성하는 PowerShell 코드 예제입니다.

```
function configureCyberVault {
    for($i = 0; $i -lt $DESTINATION_VOLUME_NAMES.Length; $i++) {
        try {
            # checking if the volume already exists and is of type
snaplock compliance
            logMessage -message "Checking if SnapLock Compliance volume
$($DESTINATION_VOLUME_NAMES[$i]) already exists in vServer
$DESTINATION_VSERVER"
            $volume = Get-NcVol -Vserver $DESTINATION_VSERVER -Volume
$DESTINATION_VOLUME_NAMES[$i] | Select-Object -Property Name, State,
TotalSize, Aggregate, Vserver, Snaplock | Where-Object { $_.Snaplock.Type
-eq "compliance" }
```

```
            if($volume) {
                $volume
                logMessage -message "SnapLock Compliance volume
$($DESTINATION_VOLUME_NAMES[$i]) already exists in vServer
$DESTINATION_VSERVER" -type "SUCCESS"
            } else {
                # Create SnapLock Compliance volume
                logMessage -message "Creating SnapLock Compliance volume:
$($DESTINATION_VOLUME_NAMES[$i])"
                New-NcVol -Name $DESTINATION_VOLUME_NAMES[$i] -Aggregate
$DESTINATION_AGGREGATE_NAMES[$i] -SnaplockType Compliance -Type DP -Size
$DESTINATION_VOLUME_SIZES[$i] -ErrorAction Stop | Select-Object -Property
Name, State, TotalSize, Aggregate, Vserver
                logMessage -message "Volume $($DESTINATION_VOLUME_NAMES[
$i]) created successfully" -type "SUCCESS"
            }

            # Set SnapLock volume attributes
            logMessage -message "Setting SnapLock volume attributes for
volume: $($DESTINATION_VOLUME_NAMES[$i])"
            Set-NcSnaplockVolAttr -Volume $DESTINATION_VOLUME_NAMES[$i]
-MinimumRetentionPeriod $SNAPLOCK_MIN_RETENTION -MaximumRetentionPeriod
$SNAPLOCK_MAX_RETENTION -ErrorAction Stop | Select-Object -Property Type,
MinimumRetentionPeriod, MaximumRetentionPeriod
            logMessage -message "SnapLock volume attributes set
successfully for volume: $($DESTINATION_VOLUME_NAMES[$i])" -type "SUCCESS"

            # checking snapmirror relationship
            logMessage -message "Checking if SnapMirror relationship
exists between source volume $($SOURCE_VOLUME_NAMES[$i]) and destination
SnapLock Compliance volume $($DESTINATION_VOLUME_NAMES[$i])"
            $snapmirror = Get-NcSnapmirror | Select-Object SourceCluster,
SourceLocation, DestinationCluster, DestinationLocation, Status,
MirrorState | Where-Object { $_.SourceCluster -eq
$SOURCE_ONTAP_CLUSTER_NAME -and $_.SourceLocation -eq "$($SOURCE_VSERVER)
:$($SOURCE_VOLUME_NAMES[$i])" -and $_.DestinationCluster -eq
$DESTINATION_ONTAP_CLUSTER_NAME -and $_.DestinationLocation -eq "
$($DESTINATION_VSERVER):$($DESTINATION_VOLUME_NAMES[$i])" -and ($_.Status
-eq "snapmirrored" -or $_.Status -eq "uninitialized") }
            if($snapmirror) {
                $snapmirror
                logMessage -message "SnapMirror relationship already
exists for volume: $($DESTINATION_VOLUME_NAMES[$i])" -type "SUCCESS"
            } else {
                # Create SnapMirror relationship
                logMessage -message "Creating SnapMirror relationship for
```

```
volume: $($DESTINATION_VOLUME_NAMES[$i])"
                New-NcSnapmirror -SourceCluster $SOURCE_ONTAP_CLUSTER_NAME
-SourceVserver $SOURCE_VSERVER -SourceVolume $SOURCE_VOLUME_NAMES[$i]
-DestinationCluster $DESTINATION_ONTAP_CLUSTER_NAME -DestinationVserver
$DESTINATION_VSERVER -DestinationVolume $DESTINATION_VOLUME_NAMES[$i]
-Policy $SNAPMIRROR_PROTECTION_POLICY -Schedule $SNAPMIRROR_SCHEDULE
-ErrorAction Stop | Select-Object -Property SourceCluster, SourceLocation,
DestinationCluster, DestinationLocation, Status, Policy, Schedule
                logMessage -message "SnapMirror relationship created
successfully for volume: $($DESTINATION_VOLUME_NAMES[$i])" -type "SUCCESS"
            }

        } catch {
            handleError -errorMessage $_.Exception.Message
        }
    }
}
```

1. 위 단계가 완료되면 SnapLock Compliance 및 SnapVault를 사용하는 항공기용 사이버 보관소가 준비됩니다.

스냅샷 데이터를 사이버 저장소로 전송하기 전에 SnapVault 관계를 초기화해야 합니다. 그러나 그 전에 보안 강화를 수행하여 볼트를 보호해야 합니다.

PowerShell로 ONTAP 사이버 소산 강화

ONTAP 사이버 저장소는 기존 솔루션보다 사이버 공격에 대한 복원력을 높입니다. 보안을 강화하기 위해 아키텍처를 설계할 때는 공격의 표면적 영역을 줄이는 방법을 고려해야 합니다. 이는 강화된 암호 정책 구현, RBAC 활성화, 기본 사용자 계정 잠금, 방화벽 구성, 볼트 시스템 변경에 대한 승인 흐름 활용 등 다양한 방법을 통해 달성할 수 있습니다. 또한 네트워크 액세스 프로토콜을 특정 IP 주소에서 제한하면 잠재적인 취약점을 줄이는 데 도움이 될 수 있습니다.

ONTAP는 ONTAP 스토리지를 강화하는 일련의 제어 기능을 제공합니다. 를 사용하여 "ONTAP에 대한 지침 및 구성 설정"조직이 정보 시스템의 기밀성, 무결성 및 가용성에 대해 규정된 보안 목표를 충족할 수 있도록 하십시오.

강화 모범 사례

수동 단계

1. 미리 정의된 사용자 지정 관리 역할을 가진 지정된 사용자를 생성합니다.
2. 네트워크 트래픽을 격리하기 위해 새 IPspace를 생성합니다.
3. 새 IPspace에 상주하는 새 SVM을 생성합니다.
4. 방화벽 라우팅 정책이 올바르게 구성되고 모든 규칙이 정기적으로 감사되고 필요에 따라 업데이트되는지 확인합니다.

ONTAP CLI 또는 자동화 스크립트 사용

1. 관리자 다중 검증(MFA)으로 관리 보호
2. 클러스터 간 표준 데이터 "전송 중"에 대한 암호화를 활성화합니다.
3. 강력한 암호화 암호로 SSH를 보호하고 보안 암호를 적용합니다.
4. 글로벌 FIPS를 활성화합니다.
5. Telnet 및 Remote Shell(RSH)을 비활성화해야 합니다.
6. 기본 관리자 계정을 잠급니다.
7. 데이터 LIF 비활성화 및 원격 액세스 지점 보안
8. 사용하지 않거나 불필요한 프로토콜 및 서비스를 비활성화하고 제거합니다.
9. 네트워크 트래픽을 암호화합니다.
10. 슈퍼유저 및 관리 역할을 설정할 때 최소 권한 원칙을 사용합니다.
11. 허용된 IP 옵션을 사용하여 특정 IP 주소에서 HTTPS 및 SSH를 제한합니다.
12. 전송 스케줄에 따라 복제를 중지하고 재개합니다.

총알 1-4는 격리된 네트워크 지정, IPspace 분리 등과 같은 수동 개입이 필요하며 사전에 수행해야 합니다. 강화 구성에 대한 자세한 내용은 을 "ONTAP 보안 강화 가이드 를 참조하십시오"참조하십시오. 나머지 부분은 손쉽게 자동화하여 구축 및 모니터링할 수 있습니다. 이 조정 방식의 목표는 볼트 컨트롤러를 미래에 대비할 수 있도록 강화 단계를 자동화하는 메커니즘을 제공하는 것입니다. 사이버 볼트 공극이 열린 기간은 가능한 한 짧습니다. SnapVault는 마지막 업데이트 이후 변경 사항만 사이버 저장소로 이동하여 사이버 볼트가 열려 있어야 하는 시간을 최소화하는 점진적 영구 기술을 활용합니다. 워크플로를 더욱 최적화하기 위해 사이버 볼트 열기가 복제 일정과 조정되어 가장 작은 연결 기간을 보장합니다.

다음은 ONTAP 컨트롤러를 강화하기 위한 PowerShell 코드 예제입니다.

```
function removeSvmDataProtocols {
    try {

        # checking NFS service is disabled
        logMessage -message "Checking if NFS service is disabled on
vServer $DESTINATION_VSERVER"
        $nfsService = Get-NcNfsService
        if($nfsService) {
            # Remove NFS
            logMessage -message "Removing NFS protocol on vServer :
$DESTINATION_VSERVER"
            Remove-NcNfsService -VserverContext $DESTINATION_VSERVER
-Confirm:$false
            logMessage -message "NFS protocol removed on vServer :
$DESTINATION_VSERVER" -type "SUCCESS"
        } else {
            logMessage -message "NFS service is disabled on vServer
$DESTINATION_VSERVER" -type "SUCCESS"
        }
```

```
        # checking CIFS/SMB server is disabled
        logMessage -message "Checking if CIFS/SMB server is disabled on
vServer $DESTINATION_VSERVER"
        $cifsServer = Get-NcCifsServer
        if($cifsServer) {
            # Remove SMB/CIFS
            logMessage -message "Removing SMB/CIFS protocol on vServer :
$DESTINATION_VSERVER"
            $domainAdministratorUsername = Read-Host -Prompt "Enter Domain
administrator username"
            $domainAdministratorPassword = Read-Host -Prompt "Enter Domain
administrator password" -AsSecureString
            $plainPassword = [Runtime.InteropServices.Marshal
]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($
domainAdministratorPassword))
            Remove-NcCifsServer -VserverContext $DESTINATION_VSERVER
-AdminUsername $domainAdministratorUsername -AdminPassword $plainPassword
-Confirm:$false -ErrorAction Stop
            logMessage -message "SMB/CIFS protocol removed on vServer :
$DESTINATION_VSERVER" -type "SUCCESS"
        } else {
            logMessage -message "CIFS/SMB server is disabled on vServer
$DESTINATION_VSERVER" -type "SUCCESS"
        }

        # checking iSCSI service is disabled
        logMessage -message "Checking if iSCSI service is disabled on
vServer $DESTINATION_VSERVER"
        $iscsiService = Get-NcIscsiService
        if($iscsiService) {
            # Remove iSCSI
            logMessage -message "Removing iSCSI protocol on vServer :
$DESTINATION_VSERVER"
            Remove-NcIscsiService -VserverContext $DESTINATION_VSERVER
-Confirm:$false
            logMessage -message "iSCSI protocol removed on vServer :
$DESTINATION_VSERVER" -type "SUCCESS"
        } else {
            logMessage -message "iSCSI service is disabled on vServer
$DESTINATION_VSERVER" -type "SUCCESS"
        }

        # checking FCP service is disabled
        logMessage -message "Checking if FCP service is disabled on
vServer $DESTINATION_VSERVER"
```

```
        $fcpService = Get-NcFcpService
        if($fcpService) {
            # Remove FCP
            logMessage -message "Removing FC protocol on vServer :
$DESTINATION_VSERVER"
            Remove-NcFcpService -VserverContext $DESTINATION_VSERVER
-Confirm:$false
            logMessage -message "FC protocol removed on vServer :
$DESTINATION_VSERVER" -type "SUCCESS"
        } else {
            logMessage -message "FCP service is disabled on vServer
$DESTINATION_VSERVER" -type "SUCCESS"
        }

    } catch {
        handleError -errorMessage $_.Exception.Message
    }
}

function disableSvmDataLifs {
    try {
        logMessage -message "Finding all data lifs on vServer :
$DESTINATION_VSERVER"
        $dataLifs = Get-NcNetInterface -Vserver $DESTINATION_VSERVER |
Where-Object { $_.Role -contains "data_core" }
        $dataLifs | Select-Object -Property InterfaceName, OpStatus,
DataProtocols, Vserver, Address

        logMessage -message "Disabling all data lifs on vServer :
$DESTINATION_VSERVER"
        # Disable the filtered data LIFs
        foreach ($lif in $dataLifs) {
            $disableLif = Set-NcNetInterface -Vserver $DESTINATION_VSERVER
-Name $lif.InterfaceName -AdministrativeStatus down -ErrorAction Stop
            $disableLif | Select-Object -Property InterfaceName, OpStatus,
DataProtocols, Vserver, Address
        }
        logMessage -message "Disabled all data lifs on vServer :
$DESTINATION_VSERVER" -type "SUCCESS"

    } catch {
        handleError -errorMessage $_.Exception.Message
    }
}

function configureMultiAdminApproval {
```

```
    try {

        # check if multi admin verification is enabled
        logMessage -message "Checking if multi-admin verification is
enabled"
        $maaConfig = Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP
-Credential $DESTINATION_ONTAP_CREDS -Command "set -privilege advanced;
security multi-admin-verify show"
        if ($maaConfig.Value -match "Enabled" -and $maaConfig.Value -match
"true") {
            $maaConfig
            logMessage -message "Multi-admin verification is configured
and enabled" -type "SUCCESS"
        } else {
            logMessage -message "Setting Multi-admin verification rules"
            # Define the commands to be restricted
            $rules = @(
                "cluster peer delete",
                "vserver peer delete",
                "volume snapshot policy modify",
                "volume snapshot rename",
                "vserver audit modify",
                "vserver audit delete",
                "vserver audit disable"
            )
            foreach($rule in $rules) {
                Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP
-Credential $DESTINATION_ONTAP_CREDS -Command "security multi-admin-verify
rule create -operation `"$rule`""
            }

            logMessage -message "Creating multi admin verification group
for ONTAP Cluster $DESTINATION_ONTAP_CLUSTER_MGMT_IP, Group name :
$MULTI_ADMIN_APPROVAL_GROUP_NAME, Users : $MULTI_ADMIN_APPROVAL_USERS,
Email : $MULTI_ADMIN_APPROVAL_EMAIL"
            Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP
-Credential $DESTINATION_ONTAP_CREDS -Command "security multi-admin-verify
approval-group create -name $MULTI_ADMIN_APPROVAL_GROUP_NAME -approvers
$MULTI_ADMIN_APPROVAL_USERS -email `"$MULTI_ADMIN_APPROVAL_EMAIL`""
            logMessage -message "Created multi admin verification group
for ONTAP Cluster $DESTINATION_ONTAP_CLUSTER_MGMT_IP, Group name :
$MULTI_ADMIN_APPROVAL_GROUP_NAME, Users : $MULTI_ADMIN_APPROVAL_USERS,
Email : $MULTI_ADMIN_APPROVAL_EMAIL" -type "SUCCESS"

            logMessage -message "Enabling multi admin verification group
$MULTI_ADMIN_APPROVAL_GROUP_NAME"
```

```
            Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP
-Credential $DESTINATION_ONTAP_CREDS -Command "security multi-admin-verify
modify -approval-groups $MULTI_ADMIN_APPROVAL_GROUP_NAME -required
-approvers 1 -enabled true"
            logMessage -message "Enabled multi admin verification group
$MULTI_ADMIN_APPROVAL_GROUP_NAME" -type "SUCCESS"

            logMessage -message "Enabling multi admin verification for
ONTAP Cluster $DESTINATION_ONTAP_CLUSTER_MGMT_IP"
            Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP
-Credential $DESTINATION_ONTAP_CREDS -Command "security multi-admin-verify
modify -enabled true"
            logMessage -message "Successfully enabled multi admin
verification for ONTAP Cluster $DESTINATION_ONTAP_CLUSTER_MGMT_IP" -type
"SUCCESS"

            logMessage -message "Enabling multi admin verification for
ONTAP Cluster $DESTINATION_ONTAP_CLUSTER_MGMT_IP"
            Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP
-Credential $DESTINATION_ONTAP_CREDS -Command "security multi-admin-verify
modify -enabled true"
            logMessage -message "Successfully enabled multi admin
verification for ONTAP Cluster $DESTINATION_ONTAP_CLUSTER_MGMT_IP" -type
"SUCCESS"
        }

    } catch {
        handleError -errorMessage $_.Exception.Message
    }
}

function additionalSecurityHardening {
    try {
        $command = "set -privilege advanced -confirmations off;security
protocol modify -application telnet -enabled false;"
        logMessage -message "Disabling Telnet"
        Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP -Credential
$DESTINATION_ONTAP_CREDS -Command $command
        logMessage -message "Disabled Telnet" -type "SUCCESS"

        #$command = "set -privilege advanced -confirmations off;security
config modify -interface SSL -is-fips-enabled true;"
        #logMessage -message "Enabling Global FIPS"
        ##Invoke-SSHCommand -SessionId $sshSession.SessionId -Command
$command -ErrorAction Stop
        #logMessage -message "Enabled Global FIPS" -type "SUCCESS"
```

```
        $command = "set -privilege advanced -confirmations off;network
interface service-policy modify-service -vserver cluster2 -policy default-
management -service management-https -allowed-addresses $ALLOWED_IPS;"
        logMessage -message "Restricting IP addresses $ALLOWED_IPS for
Cluster management HTTPS"
        Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP -Credential
$DESTINATION_ONTAP_CREDS -Command $command
        logMessage -message "Successfully restricted IP addresses
$ALLOWED_IPS for Cluster management HTTPS" -type "SUCCESS"

        #logMessage -message "Checking if audit logs volume audit_logs
exists"
        #$volume = Get-NcVol -Vserver $DESTINATION_VSERVER -Name
audit_logs -ErrorAction Stop

        #if($volume) {
        #    logMessage -message "Volume audit_logs already exists!
Skipping creation"
        #} else {
        #    # Create audit logs volume
        #    logMessage -message "Creating audit logs volume : audit_logs"
        #    New-NcVol -Name audit_logs -Aggregate
$DESTINATION_AGGREGATE_NAME -Size 5g -ErrorAction Stop | Select-Object
-Property Name, State, TotalSize, Aggregate, Vserver
        #    logMessage -message "Volume audit_logs created successfully"
-type "SUCCESS"
        #}

        ## Mount audit logs volume to path /vol/audit_logs
        #logMessage -message "Creating junction path for volume audit_logs
at path /vol/audit_logs for vServer $DESTINATION_VSERVER"
        #Mount-NcVol -VserverContext $DESTINATION_VSERVER -Name audit_logs
-JunctionPath /audit_logs | Select-Object -Property Name, -JunctionPath
        #logMessage -message "Created junction path for volume audit_logs
at path /vol/audit_logs for vServer $DESTINATION_VSERVER" -type "SUCCESS"

        #logMessage -message "Enabling audit logging for vServer
$DESTINATION_VSERVER at path /vol/audit_logs"
        #$command = "set -privilege advanced -confirmations off;vserver
audit create -vserver $DESTINATION_VSERVER -destination /audit_logs
-format xml;"
        #Invoke-SSHCommand -SessionI  $sshSession.SessionId -Command
$command -ErrorAction Stop
        #logMessage -message "Successfully enabled audit logging for
vServer $DESTINATION_VSERVER at path /vol/audit_logs"
```

```
    } catch {
        handleError -errorMessage $_.Exception.Message
    }
}
```

PowerShell로 ONTAP 사이버 소산 검증

강력한 사이버 저장소는 공격자가 상승된 Privileges로 환경에 액세스할 수 있는 자격 증명을 가지고 있더라도 정교한 공격을 견딜 수 있어야 합니다.

규칙이 수립되면 공격자가 볼트 측에서 스냅샷을 삭제하려는 시도(공격자가 침입할 수 있다고 가정함)가 실패합니다. 필요한 제한 사항을 적용하고 시스템을 보호함으로써 모든 강화 설정에도 동일하게 적용됩니다.

일정에 따라 구성을 검증하는 PowerShell 코드 예제입니다.

```
function analyze {

    for($i = 0; $i -lt $DESTINATION_VOLUME_NAMES.Length; $i++) {
        try {
            # checking if volume is of type SnapLock Compliance
            logMessage -message "Checking if SnapLock Compliance volume
$($DESTINATION_VOLUME_NAMES[$i]) exists in vServer $DESTINATION_VSERVER"
            $volume = Get-NcVol -Vserver $DESTINATION_VSERVER -Volume
$DESTINATION_VOLUME_NAMES[$i] | Select-Object -Property Name, State,
TotalSize, Aggregate, Vserver, Snaplock | Where-Object { $_.Snaplock.Type
-eq "compliance" }
            if($volume) {
                $volume
                logMessage -message "SnapLock Compliance volume
$($DESTINATION_VOLUME_NAMES[$i]) exists in vServer $DESTINATION_VSERVER"
-type "SUCCESS"
            } else {
                handleError -errorMessage "SnapLock Compliance volume
$($DESTINATION_VOLUME_NAMES[$i]) does not exist in vServer
$DESTINATION_VSERVER. Recommendation: Run the script with SCRIPT_MODE
`"configure`" to create and configure the cyber vault SnapLock Compliance
volume"
            }

            # checking SnapMirror relationship
            logMessage -message "Checking if SnapMirror relationship
exists between source volume $($SOURCE_VOLUME_NAMES[$i]) and destination
SnapLock Compliance volume $($DESTINATION_VOLUME_NAMES[$i])"
            $snapmirror = Get-NcSnapmirror | Select-Object SourceCluster,
SourceLocation, DestinationCluster, DestinationLocation, Status,
```

```
MirrorState | Where-Object { $_.SourceCluster -eq
$SOURCE_ONTAP_CLUSTER_NAME -and $_.SourceLocation -eq "$($SOURCE_VSERVER)
:$($SOURCE_VOLUME_NAMES[$i])" -and $_.DestinationCluster -eq
$DESTINATION_ONTAP_CLUSTER_NAME -and $_.DestinationLocation -eq "
$($DESTINATION_VSERVER):$($DESTINATION_VOLUME_NAMES[$i])" -and $_.Status
-eq "snapmirrored" }
            if($snapmirror) {
                $snapmirror
                logMessage -message "SnapMirror relationship successfully
configured and in healthy state" -type "SUCCESS"
            } else {
                handleError -errorMessage "SnapMirror relationship does
not exist between the source volume $($SOURCE_VOLUME_NAMES[$i]) and
destination SnapLock Compliance volume $($DESTINATION_VOLUME_NAMES[$i])
(or) SnapMirror status uninitialized/unhealthy. Recommendation: Run the
script with SCRIPT_MODE `"configure`" to create and configure the cyber
vault SnapLock Compliance volume and configure the SnapMirror
relationship"
            }
        }
        catch {
            handleError -errorMessage $_.Exception.Message
        }
    }

    try {

        # checking NFS service is disabled
        logMessage -message "Checking if NFS service is disabled on
vServer $DESTINATION_VSERVER"
        $nfsService = Get-NcNfsService
        if($nfsService) {
            handleError -errorMessage "NFS service running on vServer
$DESTINATION_VSERVER. Recommendation: Run the script with SCRIPT_MODE
`"configure`" to disable NFS on vServer $DESTINATION_VSERVER"
        } else {
            logMessage -message "NFS service is disabled on vServer
$DESTINATION_VSERVER" -type "SUCCESS"
        }

        # checking CIFS/SMB server is disabled
        logMessage -message "Checking if CIFS/SMB server is disabled on
vServer $DESTINATION_VSERVER"
        $cifsServer = Get-NcCifsServer
        if($cifsServer) {
            handleError -errorMessage "CIFS/SMB server running on vServer
```

```
$DESTINATION_VSERVER. Recommendation: Run the script with SCRIPT_MODE
`"configure`" to disable CIFS/SMB on vServer $DESTINATION_VSERVER"
        } else {
            logMessage -message "CIFS/SMB server is disabled on vServer
$DESTINATION_VSERVER" -type "SUCCESS"
        }

        # checking iSCSI service is disabled
        logMessage -message "Checking if iSCSI service is disabled on
vServer $DESTINATION_VSERVER"
        $iscsiService = Get-NcIscsiService
        if($iscsiService) {
            handleError -errorMessage "iSCSI service running on vServer
$DESTINATION_VSERVER. Recommendation: Run the script with SCRIPT_MODE
`"configure`" to disable iSCSI on vServer $DESTINATION_VSERVER"
        } else {
            logMessage -message "iSCSI service is disabled on vServer
$DESTINATION_VSERVER" -type "SUCCESS"
        }

        # checking FCP service is disabled
        logMessage -message "Checking if FCP service is disabled on
vServer $DESTINATION_VSERVER"
        $fcpService = Get-NcFcpService
        if($fcpService) {
            handleError -errorMessage "FCP service running on vServer
$DESTINATION_VSERVER. Recommendation: Run the script with SCRIPT_MODE
`"configure`" to disable FCP on vServer $DESTINATION_VSERVER"
        } else {
            logMessage -message "FCP service is disabled on vServer
$DESTINATION_VSERVER" -type "SUCCESS"
        }

        # checking if all data lifs are disabled on vServer
        logMessage -message "Finding all data lifs on vServer :
$DESTINATION_VSERVER"
        $dataLifs = Get-NcNetInterface -Vserver $DESTINATION_VSERVER |
Where-Object { $_.Role -contains "data_core" }
        $dataLifs | Select-Object -Property InterfaceName, OpStatus,
DataProtocols, Vserver, Address

        logMessage -message "Checking if all data lifs are disabled for
vServer : $DESTINATION_VSERVER"
        # Disable the filtered data LIFs
        foreach ($lif in $dataLifs) {
            $checkLif = Get-NcNetInterface -Vserver $DESTINATION_VSERVER
```

```
-Name $lif.InterfaceName | Where-Object { $_.OpStatus -eq "down" }
            if($checkLif) {
                logMessage -message "Data lif $($lif.InterfaceName)
disabled for vServer $DESTINATION_VSERVER" -type "SUCCESS"
            } else {
                handleError -errorMessage "Data lif $($lif.InterfaceName)
is enabled. Recommendation: Run the script with SCRIPT_MODE `"configure`"
to disable Data lifs for vServer $DESTINATION_VSERVER"
            }
        }
        logMessage -message "All data lifs are disabled for vServer :
$DESTINATION_VSERVER" -type "SUCCESS"

        # check if multi-admin verification is enabled
        logMessage -message "Checking if multi-admin verification is
enabled"
        $maaConfig = Invoke-NcSsh -Name $DESTINATION_ONTAP_CLUSTER_MGMT_IP
-Credential $DESTINATION_ONTAP_CREDS -Command "set -privilege advanced;
security multi-admin-verify show"
        if ($maaConfig.Value -match "Enabled" -and $maaConfig.Value -match
"true") {
            $maaConfig
            logMessage -message "Multi-admin verification is configured
and enabled" -type "SUCCESS"
        } else {
            handleError -errorMessage "Multi-admin verification is not
configured or not enabled. Recommendation: Run the script with SCRIPT_MODE
`"configure`" to enable and configure Multi-admin verification"
        }

        # check if telnet is disabled
        logMessage -message "Checking if telnet is disabled"
        $telnetConfig = Invoke-NcSsh -Name
$DESTINATION_ONTAP_CLUSTER_MGMT_IP -Credential $DESTINATION_ONTAP_CREDS
-Command "set -privilege advanced; security protocol show -application
telnet"
        if ($telnetConfig.Value -match "enabled" -and $telnetConfig.Value
-match "false") {
            logMessage -message "Telnet is disabled" -type "SUCCESS"
        } else {
            handleError -errorMessage "Telnet is enabled. Recommendation:
Run the script with SCRIPT_MODE `"configure`" to disable telnet"
        }

        # check if network https is restricted to allowed IP addresses
        logMessage -message "Checking if HTTPS is restricted to allowed IP
```

```
addresses $ALLOWED_IPS"
        $networkServicePolicy = Invoke-NcSsh -Name
$DESTINATION_ONTAP_CLUSTER_MGMT_IP -Credential $DESTINATION_ONTAP_CREDS
-Command "set -privilege advanced; network interface service-policy show"
        if ($networkServicePolicy.Value -match "management-https:
$($ALLOWED_IPS)") {
            logMessage -message "HTTPS is restricted to allowed IP
addresses $ALLOWED_IPS" -type "SUCCESS"
        } else {
            handleError -errorMessage "HTTPS is not restricted to allowed
IP addresses $ALLOWED_IPS. Recommendation: Run the script with SCRIPT_MODE
`"configure`" to restrict allowed IP addresses for HTTPS management"
        }
    }
    catch {
        handleError -errorMessage $_.Exception.Message
    }
}
```

이 스크린샷은 볼트 컨트롤러에 연결이 없음을 보여 줍니다.

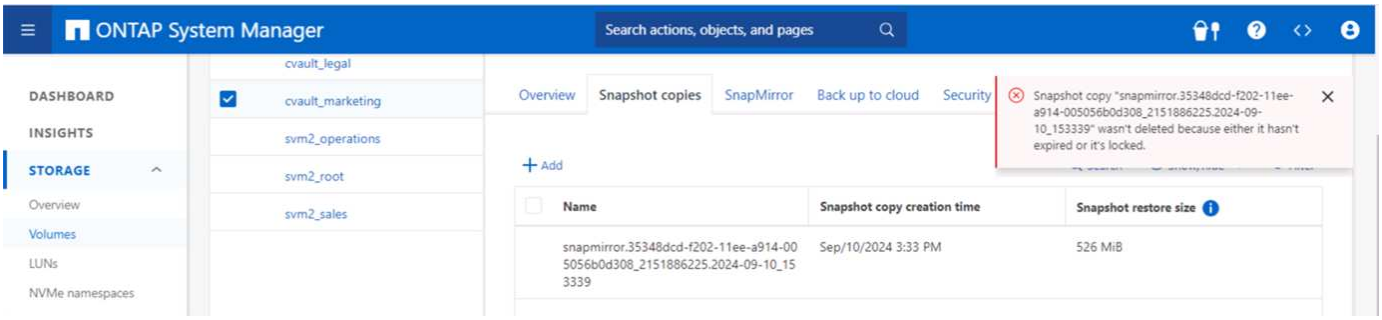
```
cluster2::> network connections  listening show
This table is currently empty.

cluster2::> network connections active show-services
This table is currently empty.

cluster2::> network connections active show-protocols
This table is currently empty.

cluster2::>
```

이 스크린샷은 스냅샷을 변조하는 기능이 없음을 보여 줍니다.



에어 갭 기능을 검증하고 확인하려면 다음 단계를 수행하십시오.

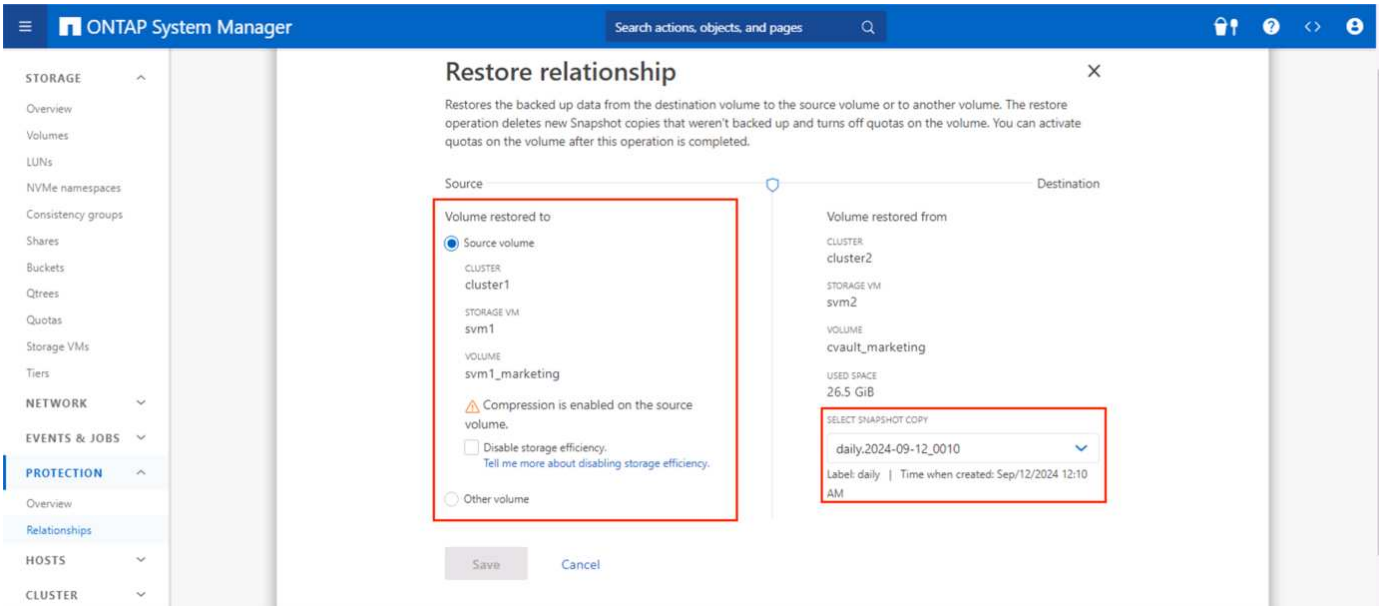
- 네트워크 격리 기능을 테스트하고 데이터가 전송되지 않을 때 연결을 중지하는 기능을 테스트합니다.

- 허용된 IP 주소 이외의 엔터티에서 관리 인터페이스에 액세스할 수 없는지 확인합니다.
- 추가 승인 레이어를 제공하기 위해 다중 관리자 확인이 있는지 확인합니다.
- CLI 및 REST API를 통한 액세스 기능을 검증합니다
- 원본에서 볼트로 전송 작업을 트리거하고 볼트로 저장된 복사본을 수정할 수 없도록 합니다.
- 볼트로 전송된 변경 불가능한 스냅샷 복사본을 삭제해 보십시오.
- 시스템 클록을 변조하여 보존 기간을 수정하십시오.

ONTAP 사이버 소산 데이터 복구

프로덕션 데이터 센터에서 데이터가 파괴되면 사이버 볼트의 데이터를 선택한 환경에 안전하게 복구할 수 있습니다. 물리적으로 에어갭 솔루션과 달리, 에어갭 ONTAP 사이버 소산은 SnapLock Compliance 및 SnapMirror과 같은 네이티브 ONTAP 기능을 사용하여 구축됩니다. 그 결과 복구 프로세스가 빠르고 쉽게 실행됩니다.

랜섬웨어 공격이 발생하고 사이버 보관소로부터 복구해야 할 경우, 사이버 저장소에 포함된 스냅샷 복사본을 사용하여 암호화된 데이터를 복원하므로 복구 프로세스가 간단하고 쉽습니다.



필요한 경우 데이터를 신속하게 검증하고, 격리하고, 복구용 데이터를 분석하기 위해 필요한 경우 데이터를 다시 온라인으로 되돌릴 수 있는 빠른 방법을 제공해야 합니다. SnapLock-type 옵션을 non-SnapLock 유형으로 설정한 FlexClone와 함께 사용하면 이러한 작업을 쉽게 수행할 수 있습니다.

.13.1부터 SnapLock 소산 관계의 대상 SnapLock 볼륨에서 잠긴 스냅샷 복사본을 복원합니다. SnapLock-type 옵션이 "non-SnapLock"로 설정된 FlexClone를 생성하면 ONTAP 9를 즉시 복원할 수 있습니다. 볼륨 클론 생성 작업을 실행할 때 스냅샷 복사본을 "상위 스냅샷"으로 지정합니다. SnapLock 유형으로 FlexClone 볼륨을 생성하는 방법에 대한 자세한 정보 "여기."

사이버 볼트에서 복구 절차를 연습하면 사이버 볼트에 연결하고 데이터를 검색하기 위한 적절한 단계가 수립됩니다. 사이버 공격 이벤트 중 복구를 위해서는 절차를 계획하고 테스트하는 것이 중요합니다.

추가 고려 사항

ONTAP 기반 사이버 볼트를 설계 및 배포할 때 추가로 고려해야 할 사항이 있습니다.

용량 사이징 고려 사항

ONTAP 사이버 소산 대상 볼륨에 필요한 디스크 공간의 양은 다양한 요인에 따라 달라집니다. 그 중 가장 중요한 것은 소스 볼륨의 데이터 변화율입니다. 대상 볼륨의 백업 일정과 스냅샷 일정은 모두 대상 볼륨의 디스크 사용량에 영향을 미치며 소스 볼륨의 변경 속도는 일정하지 않을 가능성이 높습니다. 최종 사용자 또는 애플리케이션 동작의 향후 변경 사항을 수용하는 데 필요한 추가 스토리지 용량 버퍼를 제공하는 것이 좋습니다.

ONTAP에서 보존 기간이 1개월인 관계로 사이징하려면 운영 데이터 세트의 크기, 데이터 변경률(일일 변경률), 데이터 중복 제거 및 압축 절약 효과(해당되는 경우)를 비롯한 여러 요소를 기준으로 스토리지 요구 사항을 계산해야 합니다.

단계별 접근 방식은 다음과 같습니다.

첫 번째 단계는 사이버 볼트로 보호할 소스 볼륨의 크기를 파악하는 것입니다. 이 데이터는 처음에 사이버 볼트 대상에 복제할 기본 데이터 양입니다. 그런 다음 데이터 세트의 일일 변경률을 추정합니다. 이 값은 매일 변경되는 데이터의 비율입니다. 데이터의 동적 특성을 잘 이해하는 것이 중요합니다.

예를 들면 다음과 같습니다.

- 운영 데이터 세트의 크기 = 5TB
- 일일 변경률 = 5%(0.05)
- 중복제거 및 압축 효율성 = 50%(0.50)

이제 계산에 대해 살펴보겠습니다.

- 일일 데이터 변경률 계산:

  ```
  Changed data per day = 5000 * 5% = 250GB
  ```

- 30일 동안 변경된 총 데이터 계산:

  ```
  Total changed data in 30 days = 250 GB * 14 = 3.5TB
  ```

- 필요한 총 스토리지 용량 계산:

  ```
  TOTAL = 5TB + 3.5TB = 8.5TB
  ```

- 중복제거 및 압축 절약 효과 적용:

  ```
  EFFECTIVE = 8.5TB * 50% = 4.25TB
  ```

- 스토리지 요구 사항 요약 *
- 효율성 없음: 30일간의 사이버 소산 데이터를 저장하려면 * 8.5TB * 가 필요합니다.
- 50% 효율성: 중복 제거 및 압축 후에 4.25TB * 의 스토리지가 필요합니다.

Snapshot 복사본에는 메타데이터로 인한 추가 오버헤드가 있을 수 있지만 일반적으로 경미한 수준입니다.

하루에 여러 백업을 수행하는 경우 매일 생성되는 스냅샷 복사본 수로 계산을 조정합니다.

사이징이 미래에 대비하여 확장될 수 있도록 시간에 따른 데이터 증가를 고려하십시오.

운영/소스에 대한 성능 영향

데이터 전송은 풀 작업이므로 운영 스토리지 성능에 미치는 영향은 워크로드, 데이터 볼륨 및 백업 빈도에 따라 달라질 수 있습니다. 하지만 데이터 전송은 데이터 보호 및 백업 작업을 사이버 소산 스토리지 시스템으로 오프로드하도록 설계되었으므로 운영 시스템의 전반적인 성능 영향은 일반적으로 중간 정도이며 관리가 용이합니다. 초기 관계 설정과 첫 번째 전체 백업 중에 상당한 양의 데이터가 운영 시스템에서 사이버 소산 시스템(SnapLock Compliance 볼륨)으로 전송됩니다. 이로 인해 운영 시스템의 네트워크 트래픽과 I/O 로드가 증가할 수 있습니다. 초기 전체 백업이 완료되면 ONTAP는 마지막 백업 이후에 변경된 블록만 추적하고 전송하면 됩니다. 따라서 초기 복제에 비해 입출력 로드가 훨씬 적습니다. 증분 업데이트는 효율적이며 운영 스토리지 성능에 미치는 영향이 최소화됩니다. 볼트 프로세스는 백그라운드에서 실행되므로 운영 시스템의 생산 작업 부하에 대한 간섭 가능성이 줄어듭니다.

- 스토리지 시스템에 추가 로드를 처리할 수 있는 충분한 리소스(CPU, 메모리 및 IOPS)가 있는지 확인하면 성능에 미치는 영향이 줄어듭니다.

구성, 분석, cron 스크립트

NetApp는 다운로드하여 사이버 볼트 관계를 구성, 확인 및 예약하는 데 사용할 수 있는 단일 스크립트를 만들었습니다.

이 스크립트의 기능

- 클러스터 피어링
- SVM 피어링
- DP 볼륨 생성
- SnapMirror 관계 및 초기화
- 사이버 보관소에 사용되는 ONTAP 시스템을 강화합니다
- 전송 일정에 따라 관계를 중지하고 재개합니다
- 보안 설정을 주기적으로 확인하고 이상 징후를 보여 주는 보고서를 생성합니다

이 스크립트를 사용하는 방법

스크립트를 다운로드하고 스크립트를 사용하려면 다음 단계를 따르십시오.

- 관리자 권한으로 Windows PowerShell을 시작합니다.
- 스크립트가 포함된 디렉터리로 이동합니다.
- `.\`필요한 매개 변수와 함께 구문을 사용하여 스크립트를 실행합니다

모든 정보를 입력했는지 확인하십시오. 첫 실행(구성 모드)에서 운영 및 새 사이버 볼트 시스템에 대한 자격 증명을 요구합니다. 그 후 SVM 피어링(존재하지 않는 경우), 시스템 간에 볼륨과 SnapMirror를 생성하여 초기화합니다.

cron 모드를 사용하여 데이터 전송 일시 중지 및 다시 시작을 예약할 수 있습니다.

작동 모드

자동화 스크립트는 실행 모드 - `configure`, `analyze` 를 `cron`제공합니다.

```
if($SCRIPT_MODE -eq "configure") {
        configure
    } elseif ($SCRIPT_MODE -eq "analyze") {
        analyze
    } elseif ($SCRIPT_MODE -eq "cron") {
        runCron
    }
```

- Configure(구성) - 유효성 검사를 수행하고 시스템을 에어 갭(air-gapped)으로 구성합니다.
- 분석 - 자동화된 모니터링 및 보고 기능을 통해 모니터링 그룹에 정보를 보내 이상 징후와 의심스러운 활동이 있는지 확인하고 구성이 표류되지 않도록 합니다.
- cron - 연결이 끊긴 인프라를 사용하기 위해 cron 모드는 LIF를 자동으로 비활성화하고 전송 관계를 중지합니다.

시스템 성능과 데이터 양에 따라 선택한 볼륨의 데이터를 전송하는 데 시간이 걸립니다.

```
./script.ps1 -SOURCE_ONTAP_CLUSTER_MGMT_IP "172.21.166.157"
-SOURCE_ONTAP_CLUSTER_NAME "NTAP915_Src" -SOURCE_VSERVER "svm_NFS"
-SOURCE_VOLUME_NAME "Src_RP_Vol01" -DESTINATION_ONTAP_CLUSTER_MGMT_IP
"172.21.166.159" -DESTINATION_ONTAP_CLUSTER_NAME "NTAP915_Destn"
-DESTINATION_VSERVER "svm_nim_nfs" -DESTINATION_AGGREGATE_NAME
"NTAP915_Destn_01_VM_DISK_1" -DESTINATION_VOLUME_NAME "Dst_RP_Vol01_Vault"
-DESTINATION_VOLUME_SIZE "5g" -SNAPLOCK_MIN_RETENTION "15minutes"
-SNAPLOCK_MAX_RETENTION "30minutes" -SNAPMIRROR_PROTECTION_POLICY
"XDPDefault" -SNAPMIRROR_SCHEDULE "5min" -DESTINATION_CLUSTER_USERNAME
"admin" -DESTINATION_CLUSTER_PASSWORD "PASSWORD123"
```

ONTAP 사이버 소산 PowerShell 솔루션 결론

NetApp은 ONTAP에서 제공하는 강력한 강화 방법론을 통해 공기 흐름을 활용하여 진화하는 사이버 위협에 맞서 복원력을 갖춘 격리된 보안 스토리지 환경을 구축할 수 있도록 지원합니다. 기존 스토리지 인프라의 민첩성과 효율성을 유지하면서 이러한 모든 것이 수행됩니다. 이러한 보안 액세스를 통해 기업은 기존의 인력, 프로세스 및 기술 프레임워크의 변경을 최소화하면서 엄격한 안전 및 가동 시간 목표를 달성할 수 있습니다.

ONTAP 사이버 볼트는 ONTAP의 기본 기능을 사용하므로 보호 향상 및 삭제가 불가능한 데이터 복사본을 만들 수 있습니다. NetApp의 ONTAP 기반 사이버 보관소를 전반적인 보안 환경에 추가하면 다음과 같은 이점이 있습니다.

- 운영 및 백업 네트워크와 분리되어 있는 별도의 환경을 생성하고 이 환경에 대한 사용자 액세스를 제한합니다.