Creating, hardening and validating a ONTAP cyber vault with PowerShell

NetApp Solutions

NetApp
December 19, 2024

Table of Contents

Creating, hardening and validating a ONTAP cyber vault with PowerShell

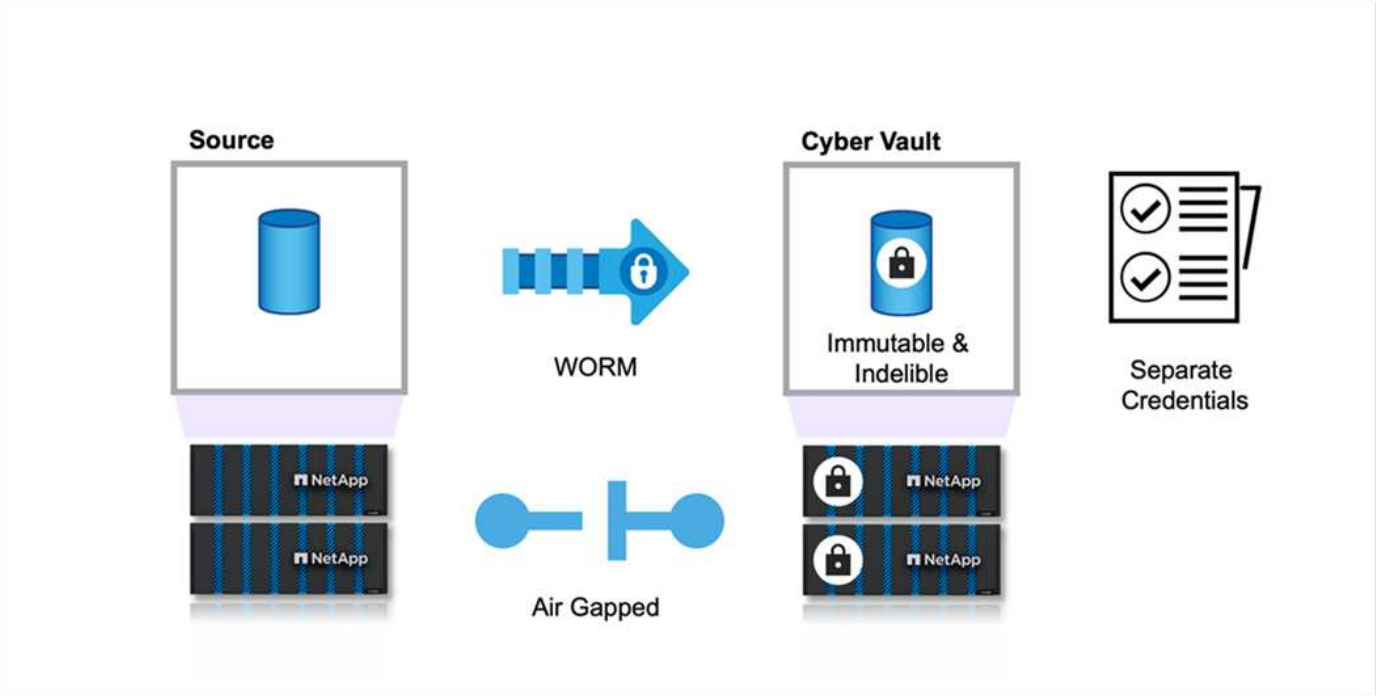
Overview of ONTAP cyber vault with PowerShell

In today's digital landscape, safeguarding an organization's critical data assets is not just a best practice - it is a business imperative. Cyber threats are evolving at an unprecedented pace, and traditional data protection measures are no longer sufficient to keep sensitive information secure. That is where a cyber vault comes in. NetApp's cutting-edge ONTAP based solution combines advanced air-gapping techniques with robust data protection measures to create an impenetrable barrier against cyber threats. By isolating the most valuable data with secure hardening technology, a cyber vault minimizes the attack surface so that the most critical data remains confidential, intact, and readily available when needed.

A cyber vault is a secure storage facility that consists of multiple layers of protection, such as firewalls, networking, and storage. These components safeguard vital recovery data necessary for crucial business operations. The cyber vault's components regularly synchronize with the essential production data based on the vault policy, but otherwise remain inaccessible. This isolated and disconnected setup ensures that in the event of a cyber-attack compromising the production environment, a reliable and final recovery can easily be carried out from the cyber vault.

NetApp enables easy creation of an air-gap for cyber vault by configuring the network, disabling LIFs, updating firewall rules, and isolating the system from external networks and the internet. This robust approach effectively disconnects the system from external networks and the internet, providing unparalleled protection against remote cyber-attacks and unauthorized access attempts, making the system immune to network-based threats and intrusion.

Combining this with SnapLock Compliance protection, data cannot be modified or deleted, not even by ONTAP administrators or NetApp Support. SnapLock is regularly audited against SEC and FINRA regulations, ensuring that data resiliency meets these stringent WORM and data retention regulations of the banking industry. NetApp is the only enterprise storage validated by NSA CSfC to store top-secret data.



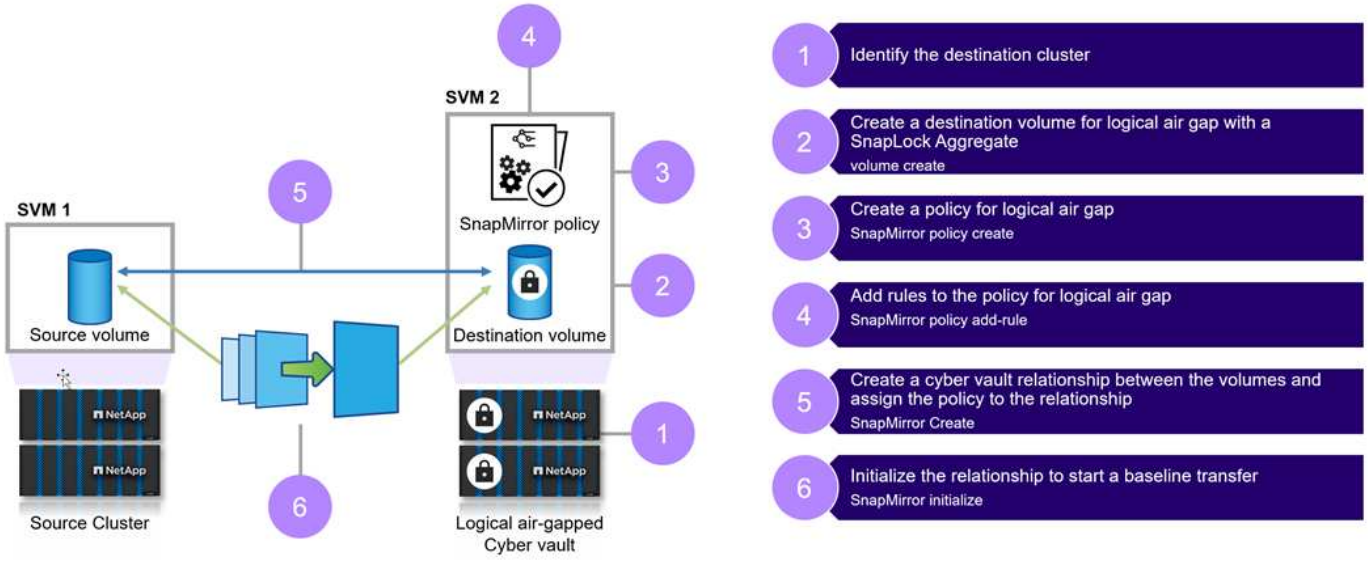
This document describes the automated configuration of NetApp's cyber vault for on-premises ONTAP storage to another designated ONTAP storage with immutable snapshots adding an extra layer of protection from increasing cyber-attacks for rapid recovery. As part of this architecture, the entire configuration is applied as per ONTAP best practices. The last section has instructions for performing a recovery in case of an attack.

The same solution is applicable to create the designated cyber vault in AWS using FSx ONTAP.

High level steps to create a ONTAP cyber vault

- Create peering relationship
  - Production site using ONTAP storage is peered with designated cyber vault ONTAP storage
- Create SnapLock Compliance volume
- Setup SnapMirror relationship and rule to set label
  - SnapMirror relationship and appropriate schedules are configured
- Set retentions prior to initiating the SnapMirror (vault) transfer
  - Retention lock is applied on the copied data, which further prevents the data from any insider or data failure. Using this, the data cannot be deleted before the retention period expires
  - Organizations can keep this data for few weeks/months depending upon their requirements
- Initialize the SnapMirror relationship based on labels
  - Initial seeding and incremental forever transfer happens based on the SnapMirror schedule
  - Data is protected (immutable and indelible) with SnapLock compliance, and that the data is available for recovery
- Implement strict data transfer controls
  - Cyber vault is unlocked for a limited period with data from the production site and is synced with data in the vault. Once the transfer is complete, the connection is disconnected, closed, and locked again
- Quick recovery

- If primary is affected in the production site, the data from the cyber vault is securely recovered to the original production or to another chosen environment



Solution components

NetApp ONTAP running 9.15.1 on source and destination clusters.

ONTAP One: NetApp ONTAP's all-in-one license.

Capabilities used from ONTAP One license:

- SnapLock Compliance
- SnapMirror
- Multi-admin verification
- All hardening capabilities exposed by ONTAP
- Separate RBAC credentials for cyber vault

> ⓘ All ONTAP unified physical arrays can be used for a cyber vault, however AFF C-series capacity based flash systems and FAS hybrid flash systems are the most cost-effective ideal platforms for this purpose. Please consult the ONTAP cyber vault sizing for sizing guidance.

ONTAP cyber vault creation with PowerShell

Air-gapping backups that use traditional methods involve creating space and physically separating the primary and secondary media. By moving the media off-site and/or severing connectivity, bad actors have no access to the data. This protects the data but can lead to slower recovery times. With SnapLock Compliance, physical separation is not required. SnapLock Compliance protects the vaulted snapshot point-in-time, read-only copies, resulting in data that is quickly accessible, safe from deletion or indelible, and safe from modification or immutable.

Pre-requisites

Before starting with the steps in the next section of this document, make sure the following prerequisites are met:

- The source cluster must be running ONTAP 9 or later.
- The source and destination aggregates must be 64-bit.
- The source and destination clusters must be peered.
- The source and destination SVMs must be peered.
- Ensure cluster peering encryption is enabled.

Setting up data transfers to a ONTAP cyber vault requires several steps. On the primary volume, configure a snapshot policy that specifies which copies to create and when to create them by using appropriate schedules and assign labels to specify which copies should be transferred by SnapVault. On the secondary, a SnapMirror policy must be created that specifies the labels of Snapshot copies to be transferred and how many of these copies should be kept on the cyber vault. After configuring these policies, create the SnapVault relationship and establish a transfer schedule.

This document assumes the primary storage and designated ONTAP cyber vault is already setup and configured.

Cyber vault cluster can be in the same or different data center as the source data.

Steps to create a ONTAP cyber vault

1. Use the ONTAP CLI or System Manager to initialize the compliance clock.
2. Create a data protection volume with SnapLock compliance enabled.
3. Use the SnapMirror create command to create SnapVault data protection relationships.
4. Set the default SnapLock Compliance retention period for the destination volume.

Default Retention is "Set to minimum." A SnapLock volume that is a vault destination has a default retention period assigned to it. The value for this period is initially set to a minimum of 0 years and maximum of 100 years (Beginning with ONTAP 9.10.1. For earlier ONTAP releases, the value is 0 - 70.) for SnapLock Compliance volumes. Each NetApp Snapshot copy is committed with this default retention period at first. The retention period can be extended later, if needed, but never shortened. For more information, see Set retention time overview.

The above encompasses manual steps. Security experts advise automating the process to avoid manual management which introduces big margin for error. Below is the code snippet that completely automates the pre-requisites and configuration of SnapLock compliance and initialization of the clock.

Here is a PowerShell code example to initializing the ONTAP compliance clock.

```
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
    }
}
```

Here is a PowerShell code example to configure a ONTAP cyber vault.

```
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
        }
    }
}
```

5. Once the above steps are completed, air-gapped cyber vault using SnapLock Compliance and SnapVault is ready.

Before transferring snapshot data to the cyber vault, the SnapVault relationship must be initialized. However, prior to that, it is necessary to perform security hardening to secure the vault.

ONTAP cyber vault hardening with PowerShell

The ONTAP cyber vault provides better resilience against cyber-attacks compared to traditional solutions. When designing an architecture to enhance security, it is crucial to consider measures to reduce the surface area of attack. This can be achieved through various methods such as implementing hardened password policies, enabling RBAC, locking default user accounts, configuring firewalls, and utilizing approval flows for any changes to the vault system. Furthermore, restricting network access protocols from specific IP address can help to limit potential vulnerabilities.

ONTAP provides a set of controls that allow to harden the ONTAP storage. Use the guidance and configuration settings for ONTAP to help organization meet prescribed security objectives for information system confidentiality, integrity, and availability.

Hardening best practices

Manual steps

1. Create a designated user with pre-defined and custom administrative role.
2. Create a new IPspace to isolate network traffic.
3. Create a new SVM residing in the new IPspace.
4. Ensure firewall routing policies are properly configured and that all rules are regularly audited and updated as needed.

ONTAP CLI or via automation script

1. Protect administration with Multi-Admin Verification (MFA)
2. Enable encryption for standard data "in-flight" between clusters.
3. Secure SSH with strong encryption cipher and enforce secure passwords.
4. Enable global FIPS.
5. Telnet and Remote Shell (RSH) should be disabled.
6. Lock default admin account.
7. Disable data LIFs and secure remote access points.
8. Disable and remove unused or extraneous protocols and services.
9. Encrypt network traffic.
10. Use the principle of least privilege when setting up superuser and administrative roles.
11. Restrict HTTPS and SSH from specific IP address using allowed IP option.
12. Quiesce and resume the replication based on the transfer schedule.

Bullets 1-4 needs manual intervention like designating an isolated network, segregating the IPspace and so on and needs to be performed beforehand. Detailed information to configure the hardening can be found in the ONTAP security hardening guide. The rest can be easily automated for easy deployment and monitoring purposes. The objective of this orchestrated approach is to provide a mechanism to automate the hardening steps to future proof the vault controller. The time frame the cyber vault air-gap is open is as short as possible. SnapVault leverages incremental forever technology, which will only move the changes since the last update into the cyber vault, thereby minimizing the amount of time the cyber vault must stay open. To further optimize the workflow, the cyber vault opening is coordinated with the replication schedule to ensure the smallest connection window.

Here is a PowerShell code example to harden a ONTAP controller.

```
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
    }
}
```

ONTAP cyber vault validation with PowerShell

A robust cyber vault should be able to withstand a sophisticated attack, even when the attacker has credentials to access the environment with elevated privileges.

Once the rules are in place, an attempt (assuming somehow the attacker was able to get in) to delete a snapshot on the vault side will fail. Same applies with all hardening settings by placing on the necessary restrictions and safeguarding the system.

PowerShell code example to validate the configuration on a schedule basis.

```
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,
```

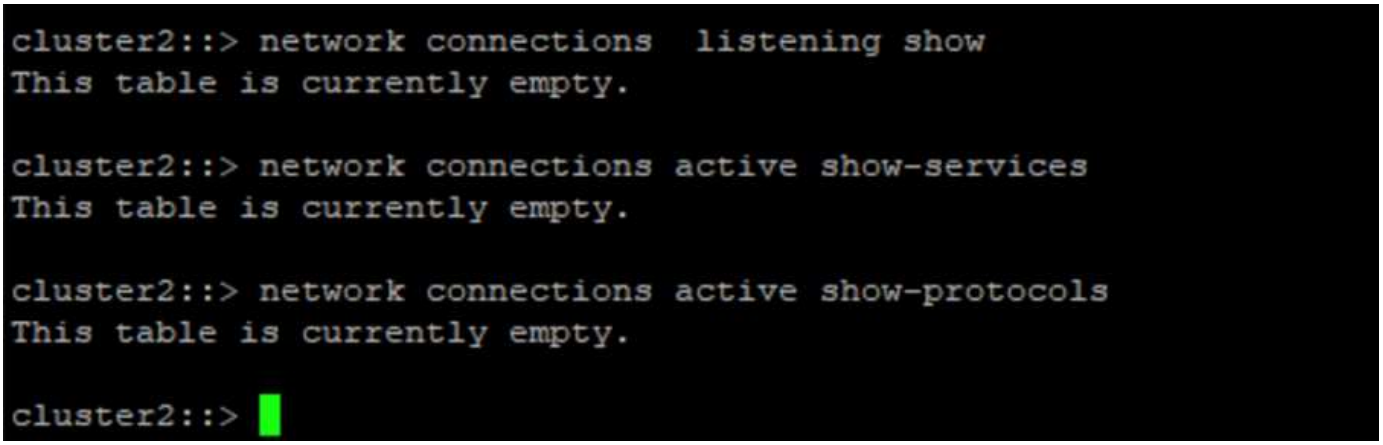
```
MirrorState | Where-Object { $_.SourceCluster -eq
$SOURCE_ONTAP_CLUSTER_NAME -and $_.SourceLocation -eq "$($SOURCE_VSERVER)
:$($SOURCE_VOLUME_NAMES[$i])" -and $_.DestinationCluster -eq
$DESTINATION_ONTAP_CLUSTER_NAME -and $_.DestinationLocation -eq "
$($DESTINATION_VSERVER):$($DESTINATION_VOLUME_NAMES[$i])" -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
```

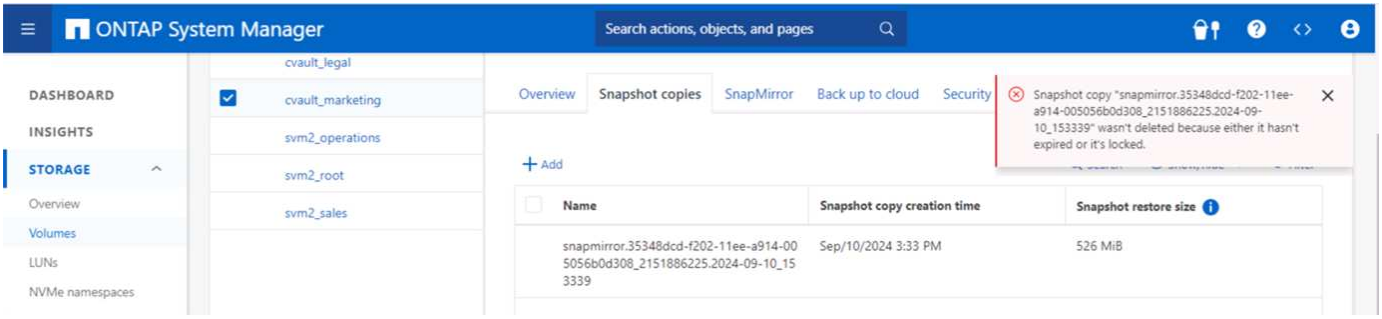
```
-Name $lif.InterfaceName | Where-Object { $_.OpStatus -eq "down" }
            if($checkLif) {
                logMessage -message "Data lif $($lif.InterfaceName)
disabled for vServer $DESTINATION_VSERVER" -type "SUCCESS"
            } else {
                handleError -errorMessage "Data lif $($lif.InterfaceName)
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
    }
}
```

This screenshot shows there are no connections on the vault controller.



This screenshot shows there is no ability to tamper with the snapshots.



To validate and confirm air-gapping functionality, follow the below steps:

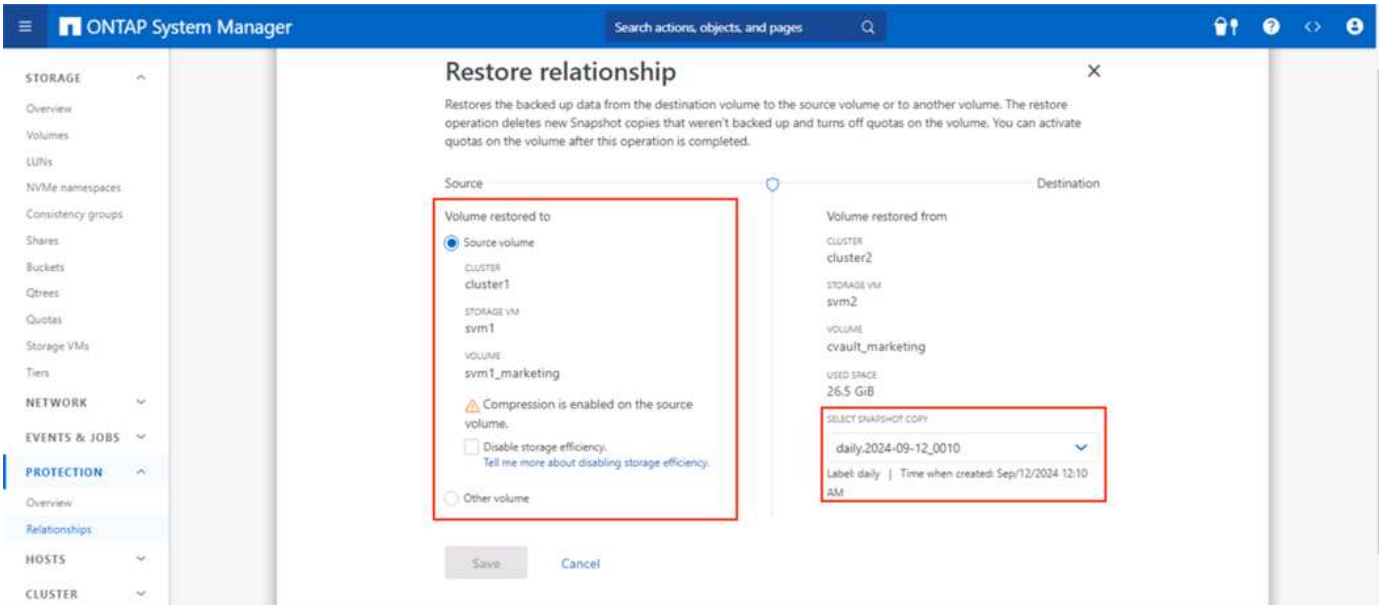
- Test network isolation capabilities, and the ability to quiesce a connection when data is not being transferred.

- Verify the management interface cannot be accessed from any entities apart from the allowed IP addresses.
- Verify Multi-admin verification is in place to provide an additional layer of approval.
- Validate the ability to access via CLI and REST API
- From the source, trigger a transfer operation to vault and ensure the vaulted copy cannot be modified.
- Try to delete the immutable snapshot copies that are transferred to the vault.
- Try to modify the retention period by tampering the system clock.

ONTAP cyber vault data recovery

If data is destroyed in the production datacenter, the data from the cyber vault can be securely recovered to the chosen environment. Unlike a physically air-gapped solution, the air-gapped ONTAP cyber vault is built using native ONTAP features like SnapLock Compliance and SnapMirror. The result is a recovery process that is both fast and easy to execute.

In the event of ransomware attack and need for recovering from the cyber vault, the recovery process is simple and easy as the snapshot copies housed in the cyber vault are used to restore the encrypted data.



If the requirement is to provide a faster method of bringing data back online when necessary to quickly validate, isolate and analyze the data for recovery. This can be easily achieved by using with FlexClone with the snaplock-type option set to non-snaplock type.

> Starting with ONTAP 9.13.1, restore a locked Snapshot copy on the destination SnapLock volume of a SnapLock vault relationship can be instantly restored by creating a FlexClone with the snaplock-type option set to "non-snaplock." When executing the volume clone creation operation, specify the Snapshot copy as the "parent-snapshot." More information about creating a FlexClone volume with a SnapLock type here.

Practicing recovery procedures from the cyber vault will ensure the proper steps are established for connecting to the cyber vault and retrieving data. Planning and testing the procedure is essential for any recovery during a cyber-attack event.

Additional considerations

There are additional considerations when designing and deploying an ONTAP based cyber vault.

Capacity sizing considerations

The amount of disk space required for an ONTAP cyber vault destination volume depends on a variety of factors, the most important of which is the rate of change for data in the source volume. The backup schedule and the Snapshot schedule on the destination volume both affect disk usage on the destination volume, and rate of change on the source volume is not likely to be constant. It is a good idea to provide a buffer of additional storage capacity above that which is required to accommodate future changes in end-user or application behavior.

Sizing a relationship for 1 month of retention in ONTAP requires calculating the storage requirements based on several factors, including the size of the primary dataset, the rate of data change (daily change rate), and the deduplication and compression savings (if applicable).

Here is the step-by-step approach:

The first step is to know the size of the source volume(s) you are protecting with the cyber vault. This is the base amount of data that will initially replicate to the cyber vault destination. Next, estimate the daily change rate for the dataset. This is the percentage of data that changes every day. It is crucial to have a good understanding of how dynamic your data is.

For example:

- Primary dataset size = 5TB
- Daily change rate = 5% (0.05)
- Deduplication and compression efficiency = 50% (0.50)

Now, let us walk through the calculation:

- Calculate the daily data change rate:

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

- Calculate the total changed data for 30 days:

  ```
  Total changed data in 30 days = 250 GB * 30 = 7.5TB
  ```

- Calculate the total storage required:

  ```
  TOTAL = 5TB + 7.5TB = 12.5TB
  ```

- Apply deduplication and compression savings:

  ```
  EFFECTIVE = 12.5TB * 50% = 6.25TB
  ```

Summary of storage needs

- Without efficiency: It would require **12.5TB** to store 30 days of the cyber vault data.
- With 50% efficiency: It would require **6.25TB** of storage after deduplication and compression.

Snapshot copies may have additional overhead due to metadata, but this is usually minor.

If multiple backups are taken per day, adjust the calculation by the number of Snapshot copies taken each day.

Factor in data growth over time to ensure sizing is future proof.

Performance impact on primary / source

Because the data transfer is a pull operation, the impact on primary storage performance can vary depending on the workload, data volume and the frequency of backups. However, the overall performance impact on the primary system is generally moderate and manageable, as data transfer is designed to offload data protection and backup tasks to the cyber vault storage system. During the initial relationship setup and the first full backup, a significant amount of data is transferred from the primary system to the cyber vault system (the SnapLock Compliance volume). This can lead to increased network traffic and I/O load on the primary system. Once the initial full backup is complete, ONTAP only needs to track and transfer blocks that have changed since the last backup. This results in a much smaller I/O load compared to the initial replication. Incremental updates are efficient and have minimal impact on primary storage performance. The vault process runs in the background, which reduces the chances of interference with the primary system's production workloads.

- Ensuring the storage system has enough resources (CPU, memory and IOPs) to handle the additional load mitigates the performance impact.

Configure, analyze, cron script

NetApp has created a single script that can be downloaded and used to configure, verify, and schedule cyber vault relationships.

What this script does

- Cluster peering
- SVM peering
- DP volume creation
- SnapMirror relationship and initialization
- Harden the ONTAP system used for the cyber vault
- Quiesce and resume the relationship based on the transfer schedule
- Validate the security settings periodically and generate a report showing any anomalies

How to use this script

Download the script and to use the script, simply follow the below steps:

- Launch Windows PowerShell as an administrator.

- Navigate to the directory containing the script.
- Execute the script using `.\` syntax along with the required parameters

> Please ensure all information entered. On the first run (configure mode), it will ask for credentials for both, the production and the new cyber vault system. After that, it will create the SVM peering's (if not existent), the volumes and the SnapMirror between the system and initialize them.

> Cron mode can be used to schedule the quiesce and resume of data transfer.

Modes of operation

The automation script provides 3 modes for execution - `configure`, `analyze` and `cron`.

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

- Configure - Performs the validation checks and configures the system as air-gapped.
- Analyze - Automated monitoring and reporting feature to send out information to monitoring groups for anomalies and suspicious activities to ensure the configurations are not drifted.
- Cron - To enable disconnected infrastructure, cron mode automates disabling the LIF and quiesces the transfer relationship.

It will take time to transfer the data in those selected volumes depending on both systems performance and the amount of data.

```
./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 cyber vault PowerShell solution conclusion

By leveraging air-gapping with robust hardening methodologies provided by ONTAP, NetApp enables you to create a secure, isolated storage environment that is resilient against evolving cyber threats. All of this is accomplished while maintaining the agility and efficiency of existing storage infrastructure. This secure access empowers companies to achieve their stringent safety and uptime goals with minimal change to their existing people, process, and technology framework.

ONTAP cyber vault uses native features in ONTAP is an easy approach for additional protection to create immutable and indelible copies of your data. Adding NetApp’s ONTAP based cyber vault to the overall security posture will:

- Create an environment that is separate and disconnected to the production and backup networks and restrict user access to it.

Copyright information

Trademark information