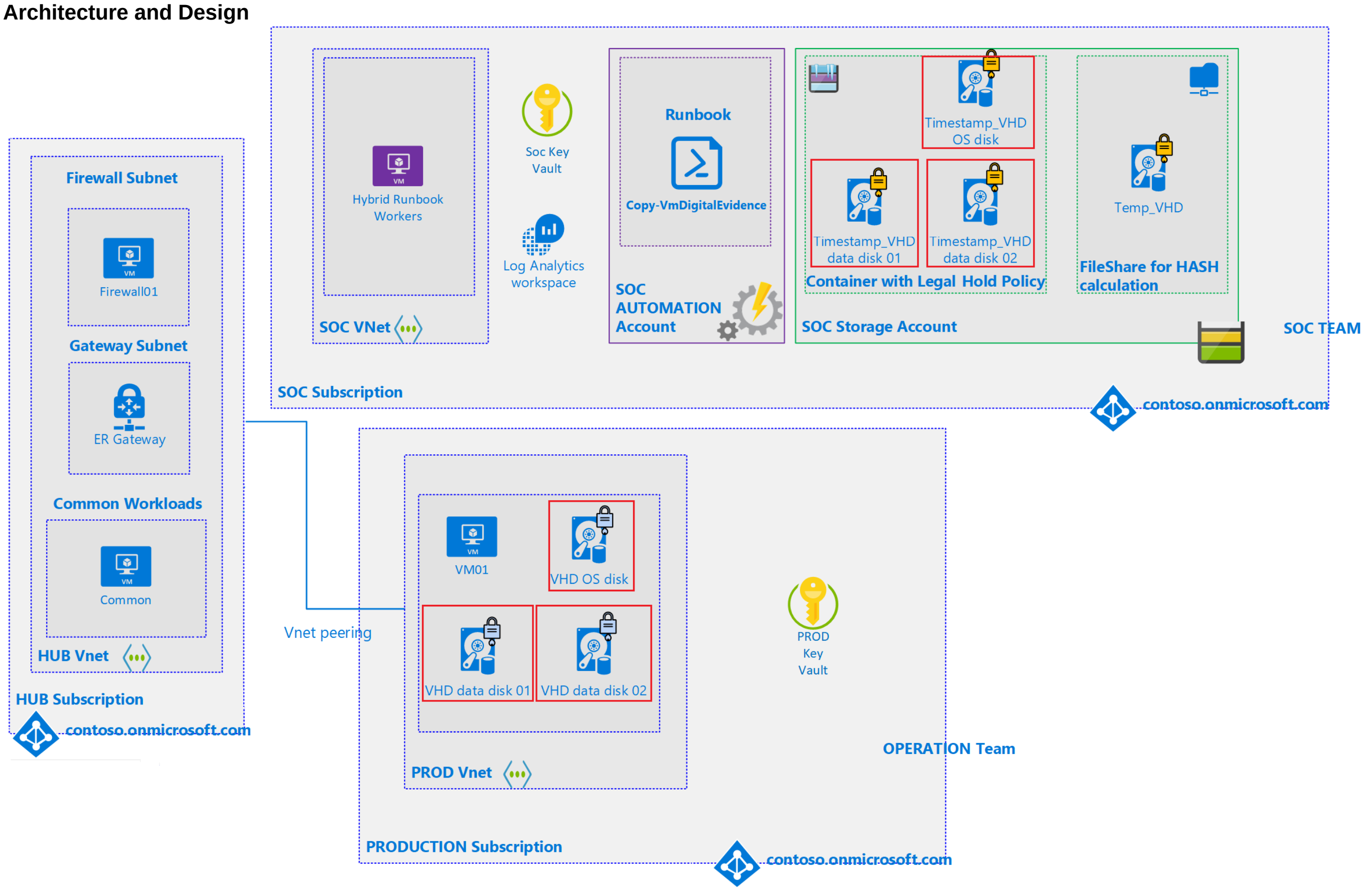
Azure - Forensics

Architecture and Design



Chain of Custody

Storage Container - stores disk snapshots in a container configured as Azure immutable Blob storage. (Write Once, Read Many (WORM). The secure transfer must be enabled.

File Share - Used as a temporary repository for calculating the snapshots’s SHA-256 hash value.

Key Vault- Contains the SHA-256 hash values for the disk snapshot. Every attached disk is protected by BitLocker. To unlock an Azure data disk connected see instructions below

Azure Automation - stores the Hybrid Runbook Workers

Log Analytics - Stores information about SOC Subscription

VM - Windows

Requirements:

- Replace Placeholders
- Adding required code
- Remove output fields - used for debugging only
- Windows Hybrid Runbook Worked
  - The scrip will create disk snapshots for all disks, copying them to immutable SOC storage, and take a SHA-256 hash, and storing the results in your SOC Key Vault.
- https://github.com/mspnp/solution-architectures/blob/master/forensics/Copy-VmDigitalEvidenceWin.ps1

VM - Linux

Requirements:

- Replace Placeholders
- Adding required code
- Remove output fields - used for debugging only
- Windows Hybrid Runbook Worked

    - The scrip will create disk snapshots for all disks, copying them to immutable SOC storage, and take a SHA-256 hash, and storing the results in your SOC Key Vault.
- https://github.com/mspnp/solution-architectures/blob/master/forensics/Copy-VmDigitalEvidence.ps1

Windows disks unlock

- Open the SOC key vault, and search the secret containing the BEK of the disk. The secret is named with the thumbprint of the Copy-VmDigitalEvidence runbook execution
- Copy the BEK string and paste it into the `$bekSecretBase64` variable in the following PowerShell script. Paste the value of the `DiskEncryptionKeyFileName` tag associated to the secret into the `$fileName` variable

```
$bekSecretbase64=""

$fileName=""

$bekFileBytes = [Convert]::FromBase64String($bekSecretbase64)

$path = "C:\BEK\$fileName"

[System.IO.File]::WriteAllBytes($path,$bekFileBytes)

manage-bde -unlock G: -rk $path
```

Linux disks unlock

- Open the SOC key vault, and search the secret containing the BEK of the disk. The secret is named with the thumbprint of the Copy-VmDigitalEvidence runbook execution
- Copy the BEK string and paste it into the `bekSecretBase64` variable in the following bash script

```
#!/bin/bash

bekSecretbase64=""

mountname="datadisk"

device=$(blkid -t TYPE=crypto_LUKS -o device)

passphrase="$(base64 -d <<< $bekSecretbase64)"

echo "Passphrase: " $passphrase

if [ ! -d "${mountname:+$mountname/}" ]; then

mkdir $mountname

fi

cryptsetup open $device $mountname
```

Capture Live Memory From Windows

- Capture the snapshot in the VMWare console with “Take Snapshot” either at the bugcheck screen or if another issue, at the time of the issue.

- Go to the following website: https://labs.vmware.com/flings/vmss2core
    - On the left-hand side, check the Agree and Download box.
    - Change the Dropdown to the appropriate OS (vmss2core-sb-8456865.exe).
    - Click on download.
- Once you have downloaded the file, save it on the C drive to a folder called c:\Snapshot
- Copy the vmss or vmsn/vmem file that you wish to convert to that folder.
- Open an elevated command prompt and run the following command:
    - cd **c:\Snapshot**
        - For VMs OS until Windows 7/2008R2 use: **vmss2core-sb-8456865 –W <snapshot.vmsn/Suspend.vmss> <snapshot.vmem>**

          For VMs OS Windows 8.1/2012 and above use: **vmss2core-sb-8456865 –W8 <snapshot.vmsn/Suspend.vmss> <snapshot.vmem>**
    - Replace the '**<snapshot.vmsn/Suspend.vmss> <snapshot.vmem>'** with the name of the snapshot.
    - This process may take a few minutes depending on the size of the snapshot, but it will create a memory.dmp file in the **c:\snapshots** folder.