Microsoft

Service Fabric – Security

Microsoft Services

Agenda

- Cluster Security
- Node Application Security

Service Fabric cluster security scenarios

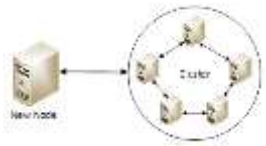
- Node-to-Node security
- Client-to-Node security
- Role-based access control (RBAC)

Azure Key Vault and Cluster security

- Azure Key Vault is used to manage certificates used for cluster security
- Certificates can be for node-to-node or client-to-node security
- Azure resource provider pulls certificates from vault and installs them on the cluster
- Create your key vaults in a separate resource group

Node-to-Node security

- Secures communication between VMs within the cluster
- Ensures that only computers authorized to join the cluster can host applications in the cluster
- Use either Certificate based security or Windows Security
- Changing the security type (certificate vs. Windows) will require cluster redeployment



Node-to-Node security – Certificate based

- Certificate based security
  - X.509 certificates become part of the node-type configuration at cluster creation time
  - Certificate security can be configured via the Azure Portal or ARM templates
  - Can have both a primary and secondary certificate to be used for certificate rollovers
    - Make sure the primary and secondary certs are different than the admin client and read-only client certs
  - Private key files (.pfx) for certificates must be in a special JSON format (Azure Resource Provider requirement)
  - Certificate must be installed in Azure Key Vault

Node-to-Node security – Windows based

- For standalone Windows Server deployments
- Requires either a Windows Server Active Directory Group or an Azure AD Group
- Requires node machines to be joined to the domain
- Using Azure Active Directory
  - Use Azure AD Domain Services (requires a classic virtual network)
  - Use VNet peering to join cluster virtual network to classic virtual network
  - Join cluster machines to the Azure AD DS domain
- Using Windows Server Active Directory
  - Create a Windows domain controller in Azure IaaS or use VPN to connect to on-premises Windows domain
  - Join node machines to the Windows domain

Node-to-Node security – Windows based

Setup of Windows security uses a ClusterConfig.Windows.*.JSON file

```
"security": {
            "ClusterCredentialType": "Windows",
            "ServerCredentialType": "Windows",
            "WindowsIdentities": {
                        "ClusterIdentity" : "[domain\machinegroup]",                    "ClientIdentities":
[{  "Identity": "[domain\username]",                    "IsAdmin": true
                        }]
            }
            }
```

*Represents the type of cluster, ie DevCluster, MultiMachine

Download http://go.microsoft.com/fwlink/?LinkId=730690

Client-to-Node security – Certificate based

- Authenticates clients and secures communication between client and individual cluster nodes
- Only authorized users can access the cluster and apps deployed on the cluster
- Clients are uniquely identified through certificate security credentials
- Certificate security can be configured via the Azure Portal or ARM templates
- Can have both a primary and secondary certificate to be used for certificate rollovers
  - Make sure the primary and secondary certs are different than the admin client and read-only client certs
- Certificate must be installed in Azure Key Vault
- Good for service development and testing, but best practice is to use Azure AD for Client-to-Node security

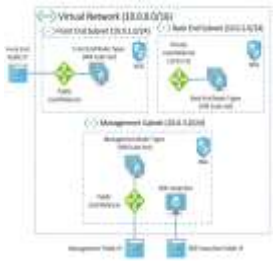
Client-to-node security – Azure AD

- Create certificates for node-to-node security (recommended)
- Certificates will be placed in Azure Key Vault (required)
- Create two Azure AD apps ~ one for Service Fabric Explorer and one for Visual Studio (recommended)
- Assign users to the roles that are supported by Service Fabric: read-only and admin
- https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-creation-via-arm

Role-based Access Control (RBAC)

- Two different access control types for clients
  - Administrators – full access to manage capabilities (read/write)
  - Users – read-only access, query capabilities, resolve applications and services
- Certificate only client-to-node - Specify the two client roles at creation time by providing separate certificates
- Azure AD client-to-node – Setup roles in an Azure AD group
- https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-security-roles

Network Isolation with Service Fabric *(example)*



Ports settings required example topology

Priority	Name	Source	Destination	Service	Action
3900	allowSvcFabPortal	Any	Any	Custom (ANY/19080)	Allow
3910	allowSvcFabClient	Any	Any	Custom (ANY/19000)	Allow
3920	allowSvcFabCluster	VirtualNetwork	Any	Custom(Any/1025-1027)	Allow
3930	allowSvcFabricEphemeral	VirtualNetwork	Any	Custom(ANY/49152/65534)	Allow
3940	allowSvcFabSMB	VirtualNetwork	Any	Custom(ANY/445)	Allow
3950	allowVNetRDP	VirtualNetwork	Any	Custom(ANY/3389)	Allow
3960	allowJumpBoxRDP	Any	10.0.3.4*	Custom(ANY/3389(	Allow
4000	blockAll	Any	Any	Custom(ANY/Any)	Deny

More information…

- https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-security

Service Fabric application security

- Applications can be secured to run under different user accounts
- Service Fabric can secure resources used by applications at deployment time
- Applications run under the account that Fabric.exe runs under
- Applications can also run under LocalUser, NetworkService, LocalService and LocalSystem
- Standalone deployments would use AD domain accounts

Service SetupEntryPoint – ServiceManifest.xml

- SetupEntryPoint – privileged entry point that runs as NetworkService account
- EntryPoint executable is typically a long running service host
- EntryPoint executable runs after SetupEntryPoint

```
<CodePackage Name="Code" Version="1.0.0">
    <SetupEntryPoint>
      <ExeHost>
        <Program>MySetup.bat</Program>
        <WorkingFolder>CodePackage</WorkingFolder>
      </ExeHost>
    </SetupEntryPoint>
    <EntryPoint>
      <ExeHost>
        <Program>MyServiceHost.exe</Program>
      </ExeHost>
    </EntryPoint>
```


Running the startup script as a local system account

- Generally recommended NOT to run a startup script as an administrator
- Recommended to run as LocalSystem
- Setup in the ApplicationManifest.xml file

```
<ServiceManifestImport>
    <ServiceManifestRef ServiceManifestName="MyServiceTypePkg" ServiceManifestVersion="1.0.0" />
    <ConfigOverrides />
    <Policies>
      <RunAsPolicy CodePackageRef="Code" UserRef="SetupLocalSystem" EntryPointType="Setup" />
    </Policies>
  </ServiceManifestImport>
  <Principals>
    <Users>
      <User Name="SetupLocalSystem" AccountType="LocalSystem" />
    </Users>
  </Principals>
</ApplicationManifest>
```


More information…

https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-application-runas-security

Demonstration

Web API Services with OWIN Self-Hosting

Microsoft