

AKS Pentesting

Lightning Talk (10 min)

Maxime Coquerel - MVP Azure



Speaker

Maxime Coquerel

Director Cloud Security Architecture

CISSP, CCSP, CSSK, Azure Security Engineer Associate

Email : max.coquerel@live.fr

Blog : zigmax.net (since 2012)

Github : <https://github.com/zigmax>

Twitter : [@zig_max](https://twitter.com/zig_max)



Disclaimer

“Any views or opinions expressed in this presentation are those of the presenter and not necessarily represent the view and opinions of my employer, its ownership, management or its employees .“

Thank you!



**CLOUD NATIVE
COMPUTING FOUNDATION**



VOOBAN

Attack Vector | Discovery: Azure Resource Graph

Query 1

```
1 resources
2 | where type == "microsoft.containerservice/managedclusters"
```

Get started **Results** Charts Messages

[Download as CSV](#) [Pin to dashboard](#)

id ↑↓	name ↑↓	type ↑↓	tenantId ↑↓	kind ↑↓	location ↑↓	resourceGroup ↑↓
/subscriptions/0dc934c0-1264-...	aksdemopub	microsoft.containerservic...	4eaa7964-c08c-4ca1-a...		canadacentral	aksdemo-pub
/subscriptions/0dc934c0-1264-...	aksdemopriv	microsoft.containerservic...	4eaa7964-c08c-4ca1-a...		canadacentral	aksdemopriv

```
"agentPoolProfiles": [
  {
    "provisioningState": "Succeeded",
    "type": "VirtualMachineScaleSets",
    "name": "agentpool",
    "count": 2,
    "osType": "Linux",
    "powerState": {
      "code": "Running"
    },
    "vmSize": "Standard_DS2_v2",
    "mode": "System",
    "currentOrchestratorVersion": "1.25.6",
    "orchestratorVersion": "1.25.6",
    "nodeImageVersion": "AKSUbuntu-2204gen2containerd-202304.10.0",
    "enableAutoScaling": true,
```

```
"addonProfiles": {
  "azureKeyvaultSecretsProvider": {
    "enabled": false,
    "config": null
  },
  "azurepolicy": {
    "enabled": false,
    "config": null
  },
```

Attack Vector | Discovery: Azure Resource Graph

```
"securityProfile": {},  
"fqdn": "aksdemopub-dns-pw0np8x8.hcp.canadacentral.azmk8s.io",  
"currentKubernetesVersion": "1.25.6",  
"servicePrincipalProfile": {  
  "clientId": "msi"  
},
```

```
"identityProfile": {  
  "kubenetidentity": {  
    "objectId": "f1f93d0c-54be-4275-b67f-100f1cfd6d9a",  
    "resourceId": "/subscriptions/0dc934c0-1264-4893-8898-  
    "clientId": "4c46b36b-8f47-4280-9b70-058faa7c7f8a"  
  }  
},  
"disableLocalAccounts": false,  
"enableRBAC": true,  
"dnsPrefix": "aksdemopub-dns",  
"autoUpgradeProfile": {  
  "upgradeChannel": "patch"
```

Log	Layer	Description
Resource logs	Azure Resources	Provide insight into operations that were performed within an Azure resource (the <i>data plane</i>). Examples might be getting a secret from a key vault or making a request to a database. The content of resource logs varies by the Azure service and resource type. <i>Resource logs were previously referred to as diagnostic logs.</i>
Activity log	Azure Subscription	Provides insight into the operations on each Azure resource in the subscription from the outside (the <i>management plane</i>) in addition to updates on Service Health events. Use the Activity log to determine the <i>what</i> , <i>who</i> , and <i>when</i> for any write operations (PUT, POST, DELETE) taken on the resources in your subscription. There's a single activity log for each Azure subscription.
Azure Active Directory (Azure AD) logs	Azure Tenant	Contain the history of sign-in activity and audit trail of changes made in Azure AD for a particular tenant.

Attack Vector | API Servers Attacks

The Kubernetes API server is a critical component of the system, and attacks against it can lead to complete system compromise. Common API server attacks include brute force attacks, unauthorized access, and denial-of-service attacks.

AKS API Server is a managed service provided by Microsoft.

- **Private API Server**

```
[→ ~ nslookup aksdemopriv-dns-lsduj049.hcp.canadacentral.azmk8s.io
Server:          24.201.245.77
Address:         24.201.245.77#53

Non-authoritative answer:
Name:   aksdemopriv-dns-lsduj049.hcp.canadacentral.azmk8s.io
Address: 10.224.0.4
```

- **Public API Server**

```
[→ ~ nslookup aksdemopub-dns-pw0np8x8.hcp.canadacentral.azmk8s.io
Server:          24.201.245.77
Address:         24.201.245.77#53

Non-authoritative answer:
Name:   aksdemopub-dns-pw0np8x8.hcp.canadacentral.azmk8s.io
Address: 20.200.67.249
```

Attack Vector | Misconfiguration

One of the most common Kubernetes attack vectors is misconfigurations that can lead to privilege escalation, data leaks, and other vulnerabilities.

Managed Clusters - List Cluster Admin Credentials

Reference

 [Feedback](#)

Service: AKS

API Version: 2023-02-01

Lists the admin credentials of a managed cluster.

HTTP

 Copy

 Try It

```
containerService/managedClusters/{resourceName}/listClusterAdminCredential?api-version=2023-02-01
```

Reference: <https://learn.microsoft.com/en-us/rest/api/aks/managed-clusters/list-cluster-admin-credentials>

Attack Vector | Misconfiguration

One of the most common Kubernetes attack vectors is misconfigurations that can lead to privilege escalation, data leaks, and other vulnerabilities.

```
PS C:\Users\maxime\MicroBurst> Get-AzPasswords -AKS Y -ACR N -AutomationAccounts N -AppServices N -Keys N -CosmosDB N -FunctionApps N -StorageAccounts N
```

```
Write-Verbose "`tGetting the clusterAdmin kubeconfig files for the $currentCluster AKS Cluster"
# For each cluster, get the admin creds
$clusterAdminCreds = ((Invoke-WebRequest -Uri (-join ('https://management.azure.com',$clusterID,'/listClusterAdminCredential?api-version=2021-05-01')) -Verbose $clusterAdminCredFile = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String(((($clusterAdminCreds | ConvertFrom-Json).kubeConfigs).value))

# Add creds to the table
$TempTblCreds.Rows.Add("AKS Cluster Admin ",$currentCluster,"clusterAdmin",$clusterAdminCredFile,"N/A","N/A","N/A","N/A","Kubeconfig-File","N/A",$subName) | Out-Null

Write-Verbose "`tGetting the clusterUser kubeconfig files for the $currentCluster AKS Cluster"
# For each cluster, get the user creds
$clusterUserCreds = ((Invoke-WebRequest -Uri (-join ('https://management.azure.com',$clusterID,'/listClusterUserCredential?api-version=2021-05-01')) -Verbose:$clusterUserCredFile = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String(((($clusterUserCreds | ConvertFrom-Json).kubeConfigs).value))
```

```
Type      : AKS Cluster Admin
Name      : akszigmaxlab
Username  : clusterAdmin
Value     : apiVersion: v1
          clusters:
            - cluster:
                certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUU2VENDQXRHZ0F3SUJBZ01SQU1zCjNG
                O6piZzcySD1NN31MnNzR3d3RFFZSktvWk1odmNOQVFFTEJRQXcKRFRFTE1Ba0dBmVVFQXhNQ1kyRXdJQmNOTWpJd01qSXpNVGt4T1RR
                eVdoZ1BnakExTwpBeU1qTXhPVEkxTKRKYQpNQTb4Q3pBSkNt1ZCQU1UQw10aE1JSUNJjakFOQmdrcWhraUc5dzBCQVFRkFBT0NBZzhB
                TU1JQ0NnS0NBZ0VBc2VnBRR1JERkVYanNZa3JiekJxQmVERzRWR1N3Mk1mQ1ZHNndzRzRjZ0I3MWh1a1M3OHZvVnJLbXhXNmV4K
                SVo0MFpHOXRUAjB2R3JHm9IODIXSFBfazBma3B0aXJUMUt3cmJxSksg3cIwdTJCZD1iZFdNbmR4aD14U21oNwpqVm1BZDZiUnJ0eWfa
                dKRcY21HSYX0c0t5YzNjeEoyNm9XVmv6cDN4cE1nWhhVeEYrMjR3SGZqcDhCSkpaSVhYc1Q5UERweGsvR1h6M1pDQkhubHk4YTVHS655
                Sk12S1ZPZEhhM3RvUkJMSk1FY25YTVN5Wm1XSE12NwozVDIvaE0kGjXL1RtN313S1Zu0GRXMnFRUnNESUJXm1VjZ1TKWEMvTG1JeDBD
                NDVYs3hQNTc0N3BRQjI0Um51M2t4L2k5VApVQitqaGQxSG8rQ1V1cCs0dUZwZ0Q5OENNK3RHZxgzNFRzd31td1BUQ1dLZTQ4eFBPPEVt
                UFNqRWF2Zjd1Yk1rCnJqewJ2VDDdhWfQvVTRERQzhVYm54Wi90RFh40Tz1aThrTudITnh6VzdOUkMvWEVPaDddT01tQ1RNTUhg6UxmSUCk
                MXhPTVRMYmgnN1GtKVFNU1pd0RoNm91S2RmNU9VcFR0Z0gzS0ZrVehJWCtn0V1Jcnhnc1FTWVArTWFSSFFMSwo3V0tHRM0yd1RJkVVRV
                SVF0TW3aj1saC9RL2tLM0xINGNZZWzak15MXVmb09xcVwRVFucTBNR01FckVzMcFICnh5N2Urc1Y5QTcrUFZ5UWkwaS9ybTdsdTKr
                R08ra1I3S3dMYzhVUHZWUJqUmFzVkdBMHV3S21wdFpRY1RmSHIKZw5xwDZ2cU0vbkRnRzBjQnJoQ2t0Nk1EWG9RUUEyV3Vka2VXG5E
                OGvFY0NBd0VBQwFOQ01FQXdEZ11EV1IwUApBUUgVqkFRREfnS2tNQTtHQTFVZEV3RUIvd1FGTUFuNkFmOHdIUUV1EV1IwT0JCWlVGT1N1
                K2pNMhk5OU1RVH2wCmV5TkV6TDc0Y1MrK01BMEdu3FHU01iM0RRRUJd1VBQTRJQ0FRQXkvaTdHQjRxcM8vb0ZSWFpZVU5JanhMMSk
                M43NURvOHFUV0hBb1NUVUZkemhadERQZVYvY1d2dW1G63EvZ255cG9nUuzUSFGZQV15YnQ2Uk1Mz1hKNpDMgoyYWRpSTBoNwVneGxP
                Tk8vRvJ1Uw1QwENqaTVMMDR0VEtxb1dabGNBYXVZOU16N0F60EdXcVRMvXBwa3ZjWkXpEcmp4c3Z1R11ElwmdLL2dMc2sxczF3QmxmMmZ1
                ZFRWnzNrK21EUjg3cFVHYwx1Rme5bmlhWEtPU2V6V1F4TH10REQKdFFUVA4VmxldwRVemF4VDhTVjM5MXi0VjBLQ1NrRnFoUnc1Q2hy
                djZwNzRrMUN0cTNTc11XbVc3VmFNNU15RwpVOXjwb1BVUVM1dD11bys5NVp2bTjQk1ExTkxJcEFaZ1NmZUw0Zz1JTFJjT0x2U1PbUJR
                ZlhhQ1dxSEsxcWswcJfJeDIXSG9pdEVYtnfs0ENPZmJlBkhJUVdYjTF1DbVfnS1pNT0gzZ1RZe1hV6ZmL2FvWnJXQWJtKzNtTnJ3NkgK
                NUw4N2gyd1pvUGFzXng40G5yUytpQkRDQVZvL3pQenZjV3NpM24wWnra2MxNjhucURzTjhzB31rL0hYSnQ5TApaoXZK1hoMwJYUExp
                Q0ZaZ19XeUQxcjZLcXpocnBDQ21mK3fQmInYrDYwzjRXUXBmZjZednJISE9jcHVIQTVPC19kL21BZGJZL0t4dkZvY3FnN1I0a2Z1akZJ
                ejJwT1V0QStl0mc2T3Iwb2FvR1prSEIyV2ZuQ01JaDNPcHU4Vi8KV3Z0y1JTde5tM0Vsd21VdTZidVks5WpwnlQQTf4d2hPbDN2c2itS
                Y1NzaDRtTnFrUw42Ujd10DY10Gp3ZHVSTgpudnFPUMFzZD1V2K3NDIwThc9PQotLS0tLUVORCDBRVJUSUZJQ0FURSB0tLS0tCg==
                server: https://akszigmaxlab-dns-c137eba5.hcp.canadacentral.azmk8s.io:443
                name: akszigmaxlab
            contexts:
            - context:
                cluster: akszigmaxlab
                user: clusterAdmin_aks-demo-max_akszigmaxlab
                name: akszigmaxlab
            current-context: akszigmaxlab
            kind: Config
            preferences: {}
            users:
            - name: clusterAdmin_aks-demo-max_akszigmaxlab
              user:
                client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUZIakNDQXdhZ0F3SUJBZ01SQUYXt3c2Z1d
                5URRPQ3F0NwdkM2I2T3d3RFFZSktvWk1odmNOQVFFTEJRQXcKRFRFTE1Ba0dBmVVFQXhNQ1kyRXdIaGNOTWpJd01qSXpNVGt4T1RR
                eVdoY05Na1F3TwpJek1Ua31OVFF5V2pBdwpNUmN3R1FZRFZRUUtFdZV6ZVh0MFpXMDZiV0Z6ZEdwEwN6RVZNQk1HQTFVRUF4TU1iV0Z6ZEd
                WkYkeHBAVzUwCk1JSUNJjakFOQmdrcWhraUc5dzBCQVFRkFBT0NBZzhBbTU1JQ0NnS0NBZ0VBc2VnVnVjZ1TKWEMvTG1JeDBD
                NDVYs3hQNTc0N3BRQjI0Um51M2t4L2k5VApVQitqaGQxSG8rQ1V1cCs0dUZwZ0Q5OENNK3RHZxgzNFRzd31td1BUQ1dLZTQ4eFBPPEVt
                UFNqRWF2Zjd1Yk1rCnJqewJ2VDDdhWfQvVTRERQzhVYm54Wi90RFh40Tz1aThrTudITnh6VzdOUkMvWEVPaDddT01tQ1RNTUhg6UxmSUCk
                MXhPTVRMYmgnN1GtKVFNU1pd0RoNm91S2RmNU9VcFR0Z0gzS0ZrVehJWCtn0V1Jcnhnc1FTWVArTWFSSFFMSwo3V0tHRM0yd1RJkVVRV
                SVF0TW3aj1saC9RL2tLM0xINGNZZWzak15MXVmb09xcVwRVFucTBNR01FckVzMcFICnh5N2Urc1Y5QTcrUFZ5UWkwaS9ybTdsdTKr
                R08ra1I3S3dMYzhVUHZWUJqUmFzVkdBMHV3S21wdFpRY1RmSHIKZw5xwDZ2cU0vbkRnRzBjQnJoQ2t0Nk1EWG9RUUEyV3Vka2VXG5E
                OGvFY0NBd0VBQwFOQ01FQXdEZ11EV1IwUApBUUgVqkFRREfnS2tNQTtHQTFVZEV3RUIvd1FGTUFuNkFmOHdIUUV1EV1IwT0JCWlVGT1N1
                K2pNMhk5OU1RVH2wCmV5TkV6TDc0Y1MrK01BMEdu3FHU01iM0RRRUJd1VBQTRJQ0FRQXkvaTdHQjRxcM8vb0ZSWFpZVU5JanhMMSk
                M43NURvOHFUV0hBb1NUVUZkemhadERQZVYvY1d2dW1G63EvZ255cG9nUuzUSFGZQV15YnQ2Uk1Mz1hKNpDMgoyYWRpSTBoNwVneGxP
                Tk8vRvJ1Uw1QwENqaTVMMDR0VEtxb1dabGNBYXVZOU16N0F60EdXcVRMvXBwa3ZjWkXpEcmp4c3Z1R11ElwmdLL2dMc2sxczF3QmxmMmZ1
                ZFRWnzNrK21EUjg3cFVHYwx1Rme5bmlhWEtPU2V6V1F4TH10REQKdFFUVA4VmxldwRVemF4VDhTVjM5MXi0VjBLQ1NrRnFoUnc1Q2hy
                djZwNzRrMUN0cTNTc11XbVc3VmFNNU15RwpVOXjwb1BVUVM1dD11bys5NVp2bTjQk1ExTkxJcEFaZ1NmZUw0Zz1JTFJjT0x2U1PbUJR
                ZlhhQ1dxSEsxcWswcJfJeDIXSG9pdEVYtnfs0ENPZmJlBkhJUVdYjTF1DbVfnS1pNT0gzZ1RZe1hV6ZmL2FvWnJXQWJtKzNtTnJ3NkgK
                NUw4N2gyd1pvUGFzXng40G5yUytpQkRDQVZvL3pQenZjV3NpM24wWnra2MxNjhucURzTjhzB31rL0hYSnQ5TApaoXZK1hoMwJYUExp
                Q0ZaZ19XeUQxcjZLcXpocnBDQ21mK3fQmInYrDYwzjRXUXBmZjZednJISE9jcHVIQTVPC19kL21BZGJZL0t4dkZvY3FnN1I0a2Z1akZJ
                ejJwT1V0QStl0mc2T3Iwb2FvR1prSEIyV2ZuQ01JaDNPcHU4Vi8KV3Z0y1JTde5tM0Vsd21VdTZidVks5WpwnlQQTf4d2hPbDN2c2itS
                Y1NzaDRtTnFrUw42Ujd10DY10Gp3ZHVSTgpudnFPUMFzZD1V2K3NDIwThc9PQotLS0tLUVORCDBRVJUSUZJQ0FURSB0tLS0tCg==
```


Attack Vector | Container Vulnerabilities

Kubernetes manages containers, so vulnerabilities in the container images can also be exploited to compromise the entire Kubernetes cluster.

Dashboard > Microsoft Defender for Cloud - Recommendations > Vulnerabilities in Azure Cont

2e7c9245e5fd

Image security health

Image	Total vulnerabilities	Vulnerabilities by severity	
 2e7c9245e5fd	3	High	0
		Medium	3
		Low	0

Digest : sha256:2e7c9245e5fdc21ff0e9a5dad05198d6639efeff7782457da022f

Tags : [2.2.401]

Findings

Search to filter items...

ID	Security Check	Category
91571	Microsoft .NET Core Security Update September 2019	Windows
177338	Debian Security Update for expat (DSA 4530-1)	Debian
177277	Debian Security Update for nghttp2 (DSA 4511-1)	Debian

91571-Microsoft .NET Core Security Update September 2019

Description

.NET Core is a general purpose development platform maintained by Microsoft and the .NET community on GitHub. It is cross-platform, supporting Windows, macOS and Linux, and can be used in device, cloud, and embedded/IoT scenarios.

A denial of service vulnerability exists when .NET Core improperly handles web requests.

Affected versions

- .NET Core 2.1.0 prior to 2.1.13
- .NET Core 2.2.0 prior to 2.2.7

Qid detection logic:Authenticated

The qid looks for sub directories under %programfiles%\dotnet\shared

```
\Microsoft.NETCore.App, %programfiles(x86)%\dotnet\shared
\Microsoft.NETCore.App and checks for vulnerable versions in .version file on windows.
```

General information

ID	91571
Severity	▲ Medium
Type	Vulnerability
Published	9/11/2019, 6:44 AM GMT+3
Patchable	Yes
Cvss 3.0 base score	7.5
CVEs	CVE-2019-1301

Remediation

Microsoft has released an update. Please refer to vendor security advisory [.NET Core CVE-2019-1301](#) for more information.

Attack Vector | Insecure third party software

Kubernetes relies on many third-party components, including plugins, add-ons, and integrations. These components can have vulnerabilities that attackers can exploit to gain access to the Kubernetes cluster.

Malicious admissions Controller

```
controlplane $ kubectl get po -n webhook-demo -w
NAME                                READY   STATUS    RESTARTS   AGE
webhook-server-5f7dcf8d7c-dbkwd     0/1    Pending  0           0s
webhook-server-5f7dcf8d7c-dbkwd     0/1    Pending  0           0s
^Ccontrolplane $ kubectl get po -n webhook-demo -w
NAME                                READY   STATUS    RESTARTS   AGE
webhook-server-5f7dcf8d7c-dbkwd     0/1    Pending  0           5s
webhook-server-5f7dcf8d7c-dbkwd     0/1    Pending  0          17s
webhook-server-5f7dcf8d7c-dbkwd     0/1    ContainerCreating  0           17s
webhook-server-5f7dcf8d7c-dbkwd     1/1    Running   0           33s
^Ccontrolplane $ kubectl run nginx --image nginx
pod/nginx created
controlplane $ kubectl get po -w
NAME    READY   STATUS             RESTARTS   AGE
nginx   0/1    ContainerCreating  0           0s
nginx   0/1    ErrImagePull       0           11s
^Ccontrolplane $ kubectl describe po nginx | grep "Image: "
Image:          rewanthtammana/malicious-image
controlplane $
```

TAMPERING IMAGE

Attack Vector | Network attacks

Kubernetes uses a network to communicate between the various components in the cluster, and attackers can intercept and manipulate network traffic to gain access to sensitive data.

Kubernetes by default **connects all the containers running in the same node** (even if they belong to different namespaces) down to **Layer 2** (ethernet). This allows a malicious containers to perform an **ARP spoofing attack** to the containers on the same node and capture their traffic.

- **ARP Spoofing in pods in the same Node**

- <https://gist.github.com/rbn15/bc054f9a84489dbdfc35d333e3d63c87#file-arpspoof-py>

- **DNS Spoofing**

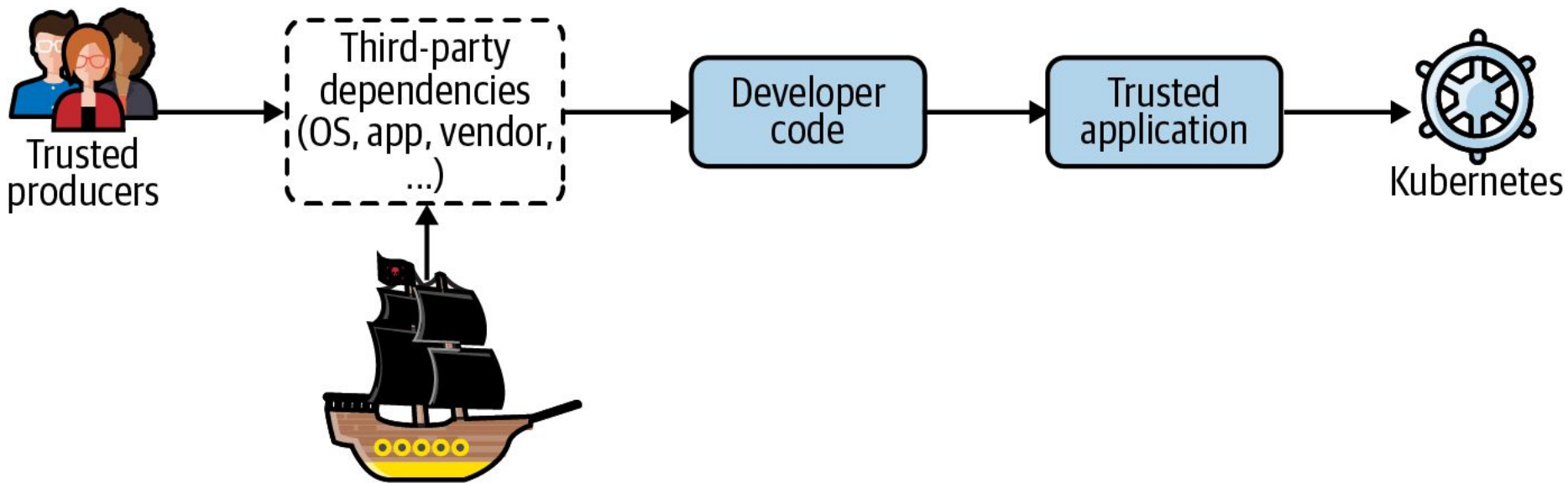
- <https://github.com/danielsagi/kube-dnsspoof/>

- **Capturing Traffic**

- <https://github.com/danielsagi/kube-dnsspoof/>

Attack Vector | Supply chain attacks

Attackers can compromise the supply chain of Kubernetes by inserting malicious code into the codebase or container images.



Attack Vector | cluster and node level attacks

Attackers can target the Kubernetes cluster or individual nodes in the cluster to gain access to sensitive data or disrupt operations.

Basics Node pools **Authentication** Networking Integrations Tags Review + create

Cluster infrastructure
The cluster infrastructure authentication specified is used by Azure Kubernetes Service to manage cloud resources attached to the cluster. This can be either a [service principal](#) or a [system-assigned managed identity](#).

Authentication method Service principal System-assigned managed identity

i The system-assigned managed identity authentication method must be used in order to associate an Azure Container Registry.

Service principal client ID *

Service principal client secret *

On the agent node VMs in the Kubernetes cluster, the service principal credentials are stored in the file **`/etc/kubernetes/azure.json`**

By default, the Service Principal that is assigned to the cluster will get the **ACRPull** role assigned at the **subscription scope level**.

Basics Node pools **Access** Networking Integrations Advanced Tags Review + create

Resource identity ⓘ

System-assigned managed identity

By default, Azure uses a managed identity. To use a service principal, use the CLI.

[Learn more](#) ↗

Attack Vector | cluster and node level attacks

```
root@aks-nodepool1-13572336-vmss000000:/# cat /host/etc/kubernetes/azure.json
{
  "cloud": "AzurePublicCloud",
  "tenantId": "4eaa7964-c08c-4ca1-a75c-4edea4556710",
  "subscriptionId": "0dc934c0-1264-4893-8898-2830b8a7d655",
  "aadClientId": "f4fc0932-878f-401d-aa46-64f9f0c6e19a",
  "aadClientSecret": "~h08Q~EmZGftm
  "resourceGroup": "MC_aksdemosp_aksdemosp_eastus",
  "location": "eastus",
  "vmType": "vmss",
  "subnetName": "aks-subnet",
  "securityGroupName": "aks-agentpool-15939821-nsg",
  "vnetName": "aks-vnet-15939821",
```


Attack Vector | cluster and node level attacks

```
root@aks-agentpool-16631174-vmss000001:/# cat /host/etc/kubernetes/azure.json
{
  "cloud": "AzurePublicCloud",
  "tenantId": "4eaa7964-c08c-4ca1-a75c-4edea4556710",
  "subscriptionId": "0dc934c0-1264-4893-8898-2830b8a7d655",
  "aadClientId": "msi",
  "aadClientSecret": "msi",
  "resourceGroup": "MC_aksdemo-pub_aksdemopub_canadacentral",
  "location": "canadacentral",
  "vmType": "vmss",
  "subnetName": "default",
  "securityGroupName": "aks-agentpool-37147250-nsg",
  "vnetName": "aksdemo-pub-vnet",
```

Attack Vector | cluster and node level attacks

Other considerations

Azure CLI

Azure PowerShell

When using AKS and an Azure AD service principal, consider the following:

- The service principal for Kubernetes is a part of the cluster configuration. However, don't use this identity to deploy the cluster.
- By default, the service principal credentials are valid for one year. You can [update or rotate the service principal credentials](#) at any time.
- Every service principal is associated with an Azure AD application. The service principal for a Kubernetes cluster can be associated with any valid Azure AD application name (for example: <https://www.contoso.org/example>). The URL for the application doesn't have to be a real endpoint.
- When you specify the service principal **Client ID**, use the value of the `ApplicationId`.
- On the agent node VMs in the Kubernetes cluster, the service principal credentials are stored in the file `/etc/kubernetes/azure.json`
- When you delete an AKS cluster that was created by `New-AzAksCluster`, the service principal created automatically isn't deleted.

Reference: <https://learn.microsoft.com/en-us/azure/aks/kubernetes-service-principal>

Attack Vector | Insider Threats

Attackers with insider access, such as employees or contractors, can exploit their privileged access to compromise the Kubernetes cluster.

```
maxime@Azure:~/clouddrive$ az aks command invoke --resource-group aksdemopriv --name aksdemopriv --command "kubectl get pods -n kube-system"
command started at 2023-04-30 23:47:15+00:00, finished at 2023-04-30 23:47:15+00:00 with exitcode=0
```

NAME	READY	STATUS	RESTARTS	AGE
ama-logs-b5n55	2/2	Running	0	89m
ama-logs-rs-757b685cd7-fgtnm	1/1	Running	0	78m
azure-ip-masq-agent-4c95r	1/1	Running	0	89m
azure-npm-qn8x5	1/1	Running	0	88m
cloud-node-manager-xhb66	1/1	Running	0	89m
coredns-75bbfcbc66-gq2x9	1/1	Running	0	88m
coredns-75bbfcbc66-rpxgw	1/1	Running	0	92m
coredns-autoscaler-7d674577fc-k7t12	1/1	Running	0	92m
csi-azuredisk-node-8vwvj	3/3	Running	0	89m
csi-azurefile-node-wcmhb	3/3	Running	0	89m
konnectivity-agent-8cb4d4cf9-24sd2	1/1	Running	0	85m
konnectivity-agent-8cb4d4cf9-fjjmw	1/1	Running	0	85m
kube-proxy-7k4cf	1/1	Running	0	89m
metrics-server-7574bb8d59-hcs2h	2/2	Running	0	76m
metrics-server-7574bb8d59-zj9sv	2/2	Running	0	76m

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Images from a private registry	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account		Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files		
Exposed Dashboard	SSH server running inside container				Access managed identity credential	Instance Metadata API	Writable volume mounts on the host		
Exposed sensitive interfaces	Sidecar injection				Malicious admission controller		Access Kubernetes dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

 = New technique

 = Deprecated technique

Questions / Talks