13. Azure Security Center for Kubernetes

Kubernetes is a very powerful platform with a lot of configuration options. Configuring your workload the right way and making sure you follow best practices can be difficult. There are industry benchmarks that you can follow to get guidelines for how to deploy your workloads securely, such as the **Center for Internet Security** (**CIS**) Benchmarks for Kubernetes: https://www.cisecurity.org/benchmark/kubernetes/.

Azure Security Center is a unified infrastructure security management platform. It provides continuous security monitoring and alerting for resources in Azure as well as for hybrid workloads. It offers protection for many Azure resources, including Kubernetes clusters. This will allow you to ensure your workloads are configured securely and protected.

Azure Security Center offers two types of protection. First, it monitors your resource configuration and compares it to security best practices, and then gives you actionable recommendations to improve your security posture. Second, it also does threat protection by assessing your workloads and raising alerts when a potential threat is identified. This threat detection capability is part of a feature called Azure Defender within Azure Security Center.

When it comes to monitoring Kubernetes workloads, Azure Security Center can monitor both your cluster configuration as well as the configuration of the workloads running in your cluster. To monitor the configuration of the workloads in your cluster, Azure Security Center uses Microsoft Azure Policy for Kubernetes. This free add-on for **Azure Kubernetes Service** (**AKS**) will enable Azure Security Center to compare the configuration of your workloads against known best practices.

Azure Defender also has specific threat detection capabilities for Kubernetes. It monitors a combination of Kubernetes audit logs, node logs, as well as cluster and workload configuration to identify potential threats. Examples of threats that can be discovered are crypto-miners, the creation of high-privileged roles, or exposing the Kubernetes dashboard.

In this chapter, you'll enable Azure Security Center, Azure Policy for Kubernetes, and Azure Defender for Kubernetes and monitor several sample applications against threats.

In this chapter, we will cover the following topics:

- Azure Security Center for Kubernetes
- Azure Defender for Kubernetes
- Deploying offending workloads
- Analyzing configuration using Azure Secure Score
- Neutralizing threats using Azure Defender

Let's start by setting up Azure Security Center for Kubernetes.

Setting up Azure Security Center for Kubernetes

We'll start this chapter by setting up Azure Security Center for Kubernetes. To enable Azure Security Center for Kubernetes, you need to enable Azure Policy for AKS on your cluster. This will enable Azure Security to monitor your workload configuration. To benefit from Azure Defender's threat protection, you will also need to enable Azure Defender for Kubernetes on your subscription.

Let's get started.

1. Search for your AKS cluster in the Azure search bar, as shown in *Figure 13.1*:

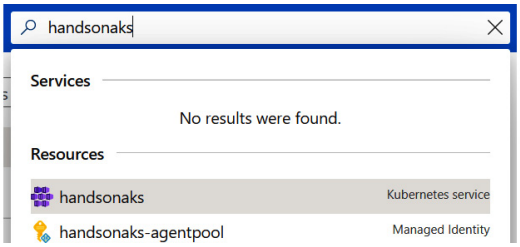

Figure 13.1: Looking for your cluster in the Azure search bar

2. You will now enable Azure Policy for AKS. To enable this, click the Policies button on the left-hand side and on the resulting pane, click on Enable add-on, as shown in *Figure 13.2*:

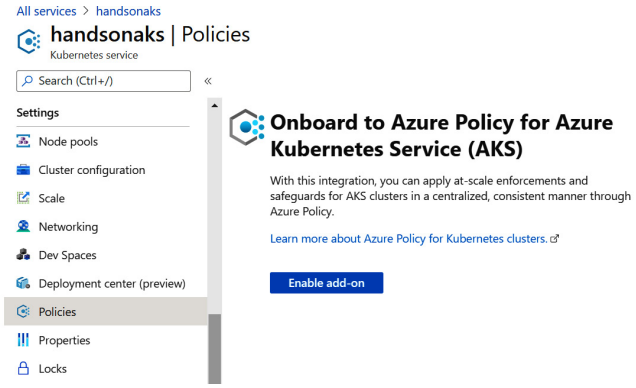

Figure 13.2: Enabling Azure Policy for AKS

Enabling the add-on will take a couple of minutes to complete. After a while, you should see a message saying that the service is now enabled, as shown in *Figure 13.3*:

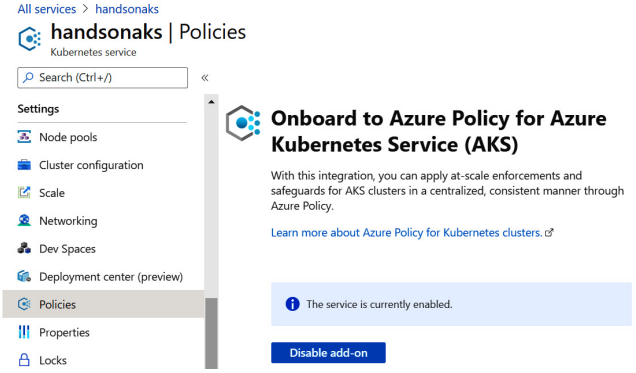

Figure 13.3: Azure Policy for AKS is now enabled

3. This has enabled Azure Policy for AKS. Next, you will enable Azure Defender to get the threat prevention ability from Azure Security Center. To do so, look up **`security center`** in the Azure portal's search bar, as shown in *Figure 13.4*:

Figure 13.4: Searching for security center in the Azure portal search bar

4. If this is the first time you are accessing Azure Security Center in your subscription, you'll be greeted with a message as shown in *Figure 13.5*. To enable Azure Defender, click on Upgrade at the bottom of the screen:

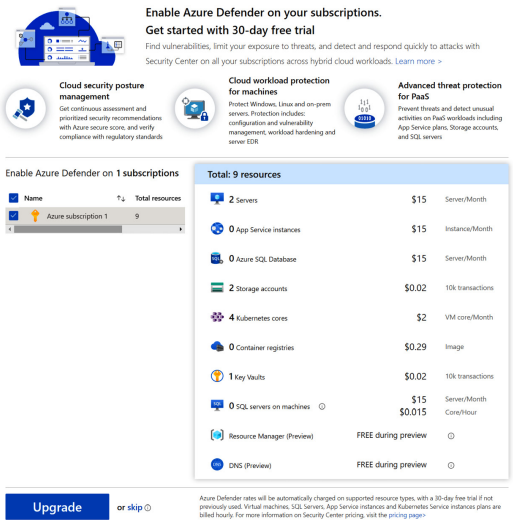

Figure 13.5: Upgrading to Azure Defender

If you're not accessing Azure Security Center in your subscription for the first time, you might not be greeted with this message to enable Azure Defender. To enable it, click on Pricing & settings on the left-hand side and select your subscription, as shown in *Figure 13.6*:

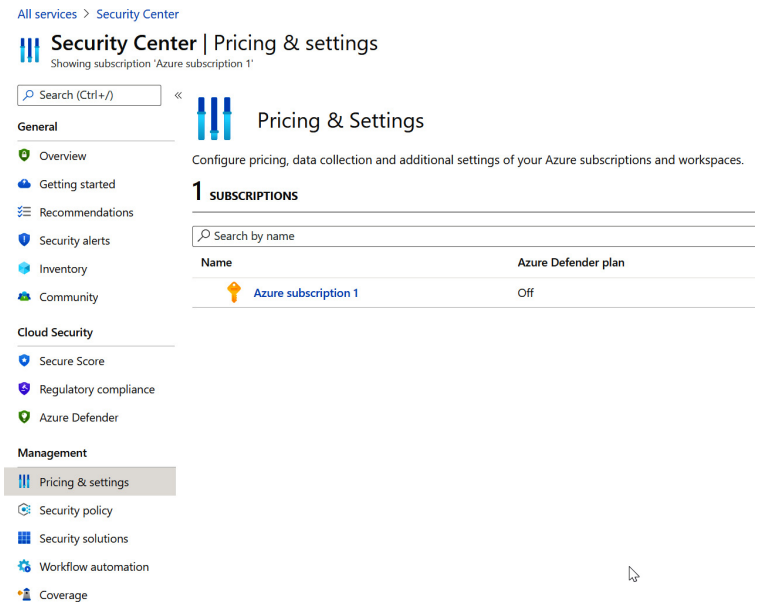

Figure 13.6: Manually upgrading to Azure Defender

In the resulting pane, select the right box with the title Azure Defender on and click the Save button at the top of the screen to enable Azure Defender. Optionally, you could tune which service you want to enable/disable Azure Defender for, as shown in *Figure 13.7*:

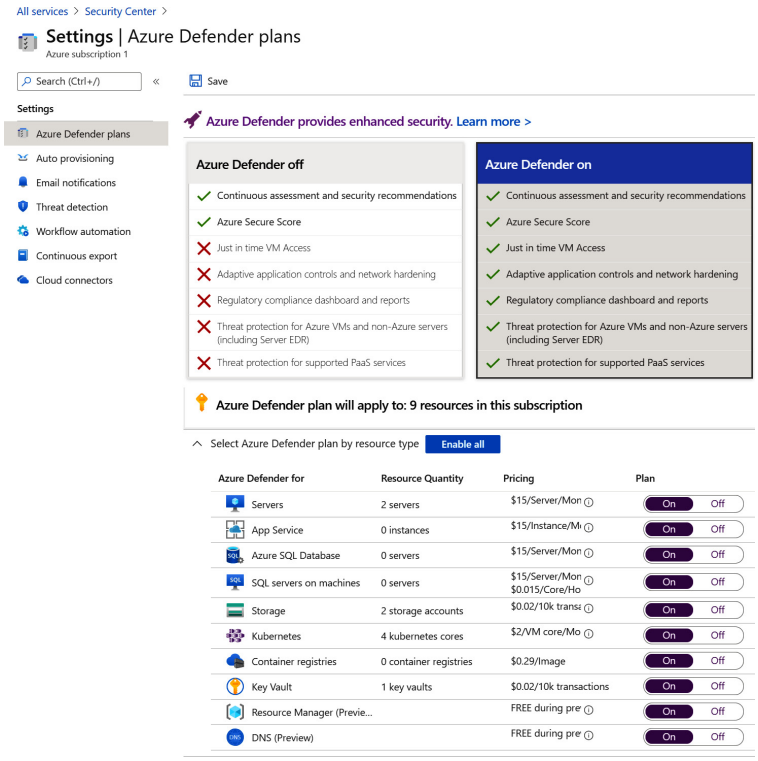

Figure 13.7: Turning on Azure Defender for your subscription

Now that you have enabled Azure Security Center and Azure Defender, it will take up to 30 minutes for the system to configure the default policy and start detections.

While you are waiting for this configuration to become effective, you will deploy several offending workloads on your cluster, which will trigger alerts in Azure Defender.

Deploying offending workloads

To trigger recommendations and threat alerts in Azure Security Center, you will need to have offending workloads deployed on your cluster. In this section, you will deploy a number of workloads to your cluster that are either not configured according to best practices or even contain potentially malicious software such as crypto-miners. Let's have a look at the examples of offending workloads you can find in the code samples for this chapter:

- **`crypto-miner.yaml`**: This file contains a deployment that will create a crypto-miner on your cluster.

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: crypto-miner
5    labels:
6      app: mining
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: mining
12   template:
13     metadata:
14       labels:
15         app: mining
16     spec:
17       containers:
18       - name: mining
19         image: kannix/monero-miner:latest
```

This file is a regular deployment in Kubernetes. As you can see on *line 19*, the container image for this deployment will be a crypto-miner.

Note

Make sure to stop running the crypto-miner as soon as you are done with this chapter, as explained in the Neutralizing threats using Azure Defender section. There is no point in running the crypto-miner any longer than this example requires.

- **`escalation.yaml`**: This file contains a deployment that allows privilege escalations in the container. This means that a process in the container can get access to the host operating system. There are cases where this is desired behavior, but typically you don't want this configuration.

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: escalation
5    labels:
6      app: nginx-escalation
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: nginx-escalation
12   template:
13     metadata:
14       labels:
15         app: nginx-escalation
16     spec:
```

```
17      containers:
18      - name: nginx-escalation
19        image: nginx:alpine
20        securityContext:
21          allowPrivilegeEscalation: true
```

As you can see in the preceding code sample, on *lines 20–21*, you configure the security context of the container with **`securityContext`**. You allow privilege escalation on *line 21*.

- **`host-volume.yaml`**: This file contains a deployment with a directory on the host mounted in the container. This is not recommended because this way, the container can get access to the host.

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: host-volume
5    labels:
6      app: nginx-host-volume
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: nginx-host-volume
12   template:
13     metadata:
14       labels:
15         app: nginx-host-volume
16     spec:
17       containers:
18       - name: nginx-host-volume
19         image: nginx:alpine
```

```
20        volumeMounts:
21        - mountPath: /test-pd
22          name: test-volume
23          readOnly: true
24      volumes:
25      - name: test-volume
26        hostPath:
27          # Directory on host
28          path: /tmp
```

This code sample contains a **`volumeMount`** field and a volume. As you can see in *lines 24–28*, the volume is using **`hostPath`**, meaning it mounts a volume on the node running the container.

- **`role.yaml`**: A role with very broad permissions. It is recommended to approach roles in Kubernetes with the principle of least privilege to ensure permissions are tightly controlled. A role with broad permissions is first and foremost a bad configuration, but worse, could be a sign of compromise on your cluster.

```
1  apiVersion: rbac.authorization.k8s.io/v1
2  kind: ClusterRole
3  metadata:
4    name: super-admin
5  rules:
6  - apiGroups: ["*" ]
7    resources: ["*"]
8    verbs: ["*"]
```

This instance of **`ClusterRole`** gives very broad permissions, as you can see in *lines 6–8*. This configuration gives anybody who gets assigned this role all permissions on all resources on all APIs in Kubernetes.

None of the deployments in these code samples contain resource requests and limits. As explained in *Chapter 3*, *Application deployment on AKS*, it is recommended to configure resource requests and limits, as these can prevent a workload from consuming too many resources.

Finally, you will also deploy the Kubernetes dashboard on a public service. This is also highly discouraged, as this might inadvertently give attackers access to your cluster. You will see how Azure Defender detects this.

Let's start deploying these files.

1. Open Cloud Shell in the Azure portal and navigating to the code samples for this chapter.
2. Once there, execute the following commands to create the offending workloads.
   kubectl create -f crypto-miner.yaml
   kubectl create -f escalation.yaml
   kubectl create -f host-volume.yaml
   kubectl create -f role.yaml
   This will create an output similar to *Figure 13.8*:

```
user@Azure:~/Hands-On-Kubernetes-on-Azure/Chapter13$ kubectl create -f crypto-miner.yaml
deployment.apps/alert-test created
user@Azure:~/Hands-On-Kubernetes-on-Azure/Chapter13$ kubectl create -f escalation.yaml
deployment.apps/escalation created
user@Azure:~/Hands-On-Kubernetes-on-Azure/Chapter13$ kubectl create -f host-volume.yaml
deployment.apps/host-volume created
user@Azure:~/Hands-On-Kubernetes-on-Azure/Chapter13$ kubectl create -f role.yaml
clusterrole.rbac.authorization.k8s.io/super-admin created
```

   Figure 13.8: Creating the offending workload

3. Now, deploy the Kubernetes dashboard using the following command:
   kubectl apply -f
   https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/ai
   This will create an output similar to *Figure 13.9*:

```
user@Azure:~/Hands-On-Kubernetes-on-Azure/Chapter13$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
```

Figure 13.9: Creating the Kubernetes dashboard

4. By default, the Kubernetes dashboard is not exposed through a load balancer. This is also the recommended configuration, because the dashboard gives broad access to your cluster. In this chapter, however, you will create this discouraged configuration to trigger a security alert in Azure Defender. To add a load balancer to the Kubernetes dashboard, use the following command:
   kubectl patch service \
     kubernetes-dashboard -n kubernetes-dashboard \
     -p '{"spec": {"type": "LoadBalancer"}}'
   This will patch the service and turn it into a service of the **`LoadBalancer`** type. Verify that this was patched successfully and get the public IP address of the service using the following command:
   kubectl get service -n kubernetes-dashboard
   This will generate an output similar to *Figure 13.10*:

```
user@Azure:~/Hands-On-Kubernetes-on-Azure/Chapter13$ kubectl get service -n kubernetes-dashboard
NAME                        TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)         AGE
dashboard-metrics-scraper   ClusterIP      10.0.53.98     <none>         8000/TCP        3m30s
kubernetes-dashboard        LoadBalancer   10.0.187.130   20.190.3.178   443:31673/TCP   3m31s
```

Figure 13.10: Getting the public IP of the kubernetes-dashboard service

5. Verify that you can access this service by browsing to **`https://<public IP>`**. Depending

on your browser configuration, you might get a certificate error, which you can bypass by selecting Continue to <public IP> (unsafe), as shown in *Figure 13.11*:

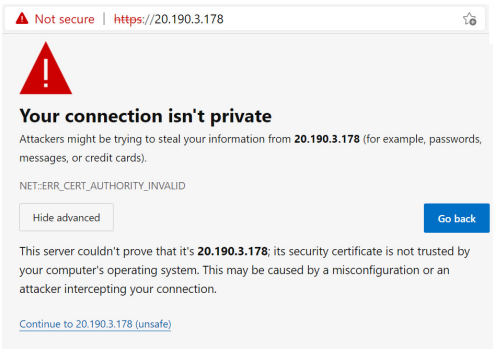

Figure 13.11: Security warning about the certificate on the Kubernetes dashboard service

Once you have continued to the dashboard, you should get a sign-in screen as shown in *Figure 13.12*:

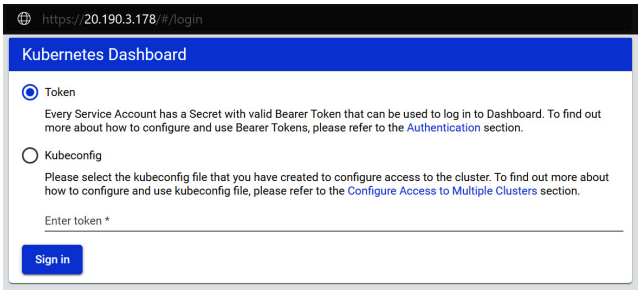

Figure 13.12: The exposed Kubernetes dashboard

You won't sign in to the dashboard here, but if you wish to explore its capabilities, please refer to the Kubernetes documentation at **https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/**.

You now have five offending workloads running on your cluster. Some of these will cause configuration warnings in Azure Security Center; some others will even trigger a security alert. You will

explore those in the next two sections of this chapter.

Analyzing configuration using Azure Secure Score

In the previous section, you created several workloads that are purposefully misconfigured. In this section, you'll review the recommendations in Azure Security Center related to these workloads.

Note

It can take up to 30 minutes after the workloads have been created for recommendations and alerts to show up.

1. After you have created the offending workloads, you will get security recommendations in Azure Security Center. To start, click on Secure Score in the left-hand navigation within Azure Security Center. This will show you a pane similar to *Figure 13.13*:

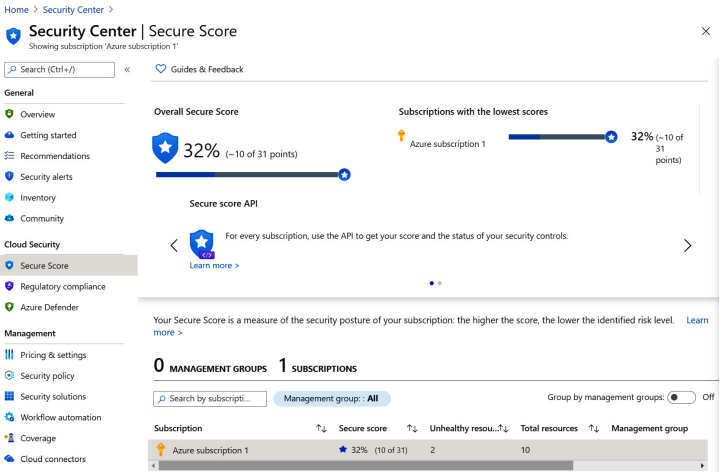

Figure 13.13: Secure Score in Azure Security Center

What you see here is a summary of the security posture of your environment. In the example shown in *Figure 13.13*, you see that the overall secure score was 32%. If you manage multiple Azure subscriptions, this view can give you a quick bird's-eye view of the security configuration of your environment.

2. Let's drill down into the configuration of the Kubernetes cluster by clicking on the Azure subscription 1 subscription at the bottom of *Figure 13.13*. This will bring you to a pane similar to *Figure 13.14*:

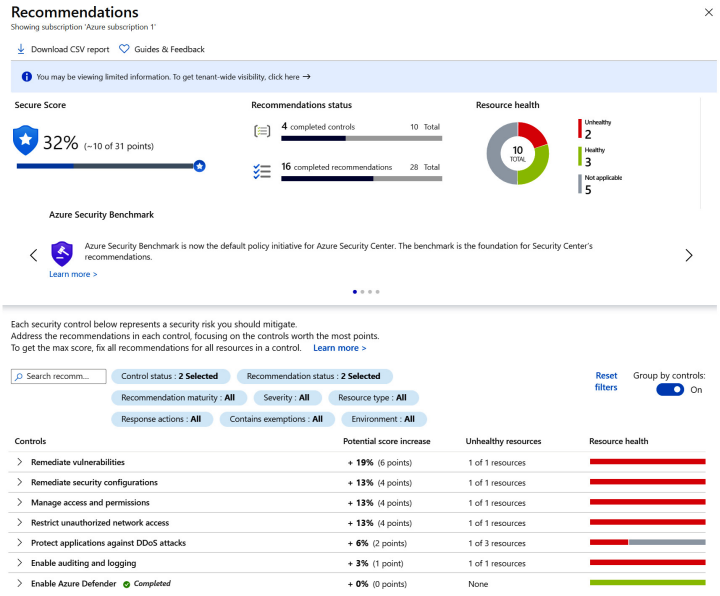

Figure 13.14: Secure Score details for the subscription

This view contains more details about the secure score for this subscription. It again shows you the secure score, as well as the recommendation status and resource health. The bottom of the screen contains more details about the specific recommendations.

3. Tune this screen to get more insight into the Kubernetes recommendations. To do this, disable the Group by controls option on the right-

hand side of the screen, and set the Resource type filter to managed cluster, as shown in *Figure 13.15*:

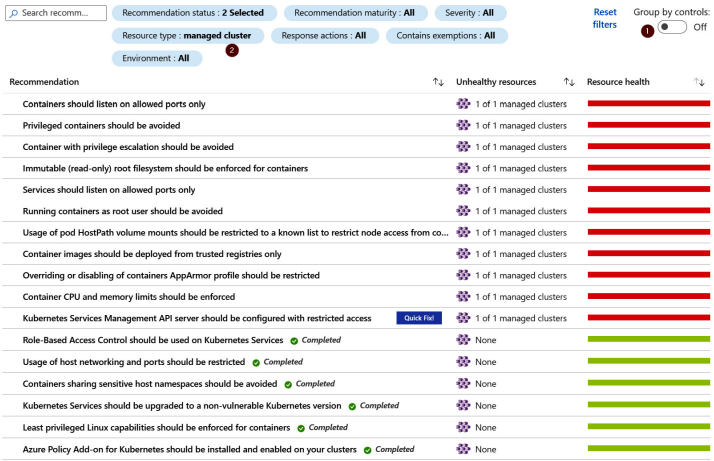

Figure 13.15: Kubernetes recommendations in Azure Security Center

You are now looking at a list of Kubernetes security recommendations recommended by Azure Security Center. The list is too exhaustive to cover completely in this chapter, but if you want to see more details about each recommendation, please refer to the AKS documentation for more detailed descriptions: **https://docs.microsoft.com/azure/aks/policy-reference**.

4. Let's, however, explore a number of the recommendations that were caused by the offending workloads you created earlier. Start by clicking on the recommendation called Container with privilege escalation should be avoided. This will bring you to a view similar to *Figure 13.16*:

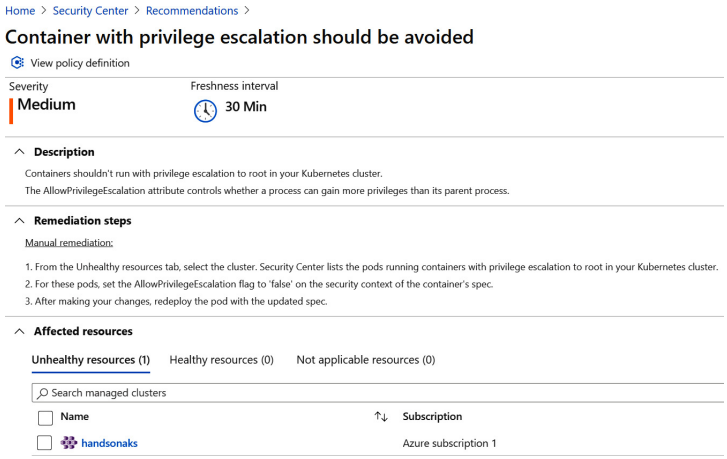

Figure 13.16: Details of the Container with privilege escalation should be avoided recommendation

As you can see, this recommendation contains a description of the recommendation itself, as well as a number of remediation steps to follow this recommendation. It also shows you the affected resource, which in this case is an AKS cluster. If you click on the cluster, you will even get more details about the offending workload, as shown in *Figure 13.17*:

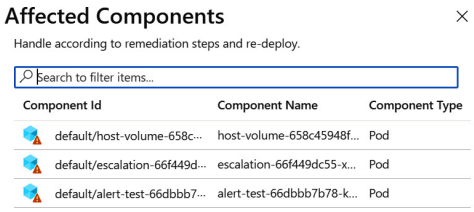

Figure 13.17: Pods affected by the privilege escalation recommendation

In this case, each of the pods you created triggered this recommendation, not just the one where privilege escalation was allowed. This shows you that Kubernetes allows this privilege escalation by default and you should implement the safeguard in all your deployments. This shows you the benefit of a security

monitoring solution such as Azure Security Center, to monitor against potential side effects of the default configuration.

5. Let's apply the suggested remediation to this issue. To solve the privilege escalation, you will need to configure the security context of the container to no longer allow privilege escalation. This can be done by updating each deployment using the following commands:

```
kubectl patch deployment crypto-miner -p '
 {
  "spec": {
   "template": {
    "spec": {
     "containers": [
      {
       "name": "mining",
       "securityContext": {
        "allowPrivilegeEscalation": false
       }
      }
     ]
    }
   }
  }
 }
 '
kubectl patch deployment escalation -p '
 {
  "spec": {
   "template": {
    "spec": {
     "containers": [
      {
       "name": "nginx-escalation",
```

```
          "securityContext": {
            "allowPrivilegeEscalation": false
          }
        }
      ]
    }
  }
}
'

kubectl patch deployment host-volume -p '
{
  "spec": {
    "template": {
      "spec": {
        "containers": [
          {
            "name": "nginx-host-volume",
            "securityContext": {
              "allowPrivilegeEscalation": false
            }
          }
        ]
      }
    }
  }
}
'
```

As you can see in the command, you are patching each deployment. In each patch, you are configuring the **`securityContext`** field and setting the **`allowPrivilegeEscalation`** field to **`false`**.

After you have applied the patch, it will take Azure Security Center up to 30 minutes to refresh the security recommendation. After that time passes, your cluster should show up as a healthy resource for this recommendation, as shown in *Figure 13.18*:

Figure 13.18: Cluster is now healthy for the privilege escalation recommendation

6. Let's investigate another recommendation, namely the one called Usage of pod HostPath volume mounts should be restricted to a known list to restrict node access from compromised containers. Click on this recommendation to be shown more details, as shown in *Figure 13.19*:

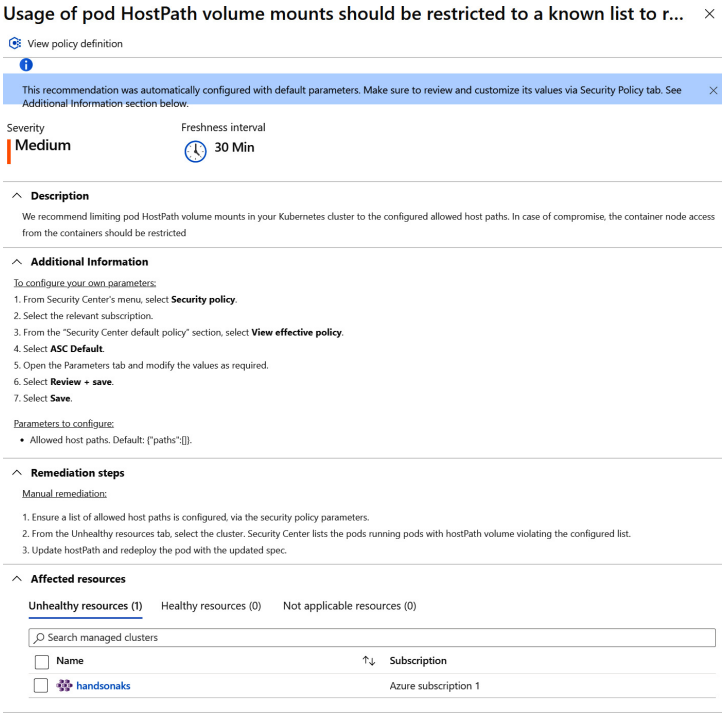

Figure 13.19: More details about the HostPath recommendation

This recommendation shows you similar information to the previous one. However, the policy that triggered this recommendation can be tuned to allow certain **`HostPath`** to be accessed. Let's explore how to edit the policy to allow this.

7. Go back to the Azure Security Center main pane and click on Security policy on the left-hand side. In the resulting pane, click on your subscription and then click on the ASC Default assignment shown in *Figure 13.20*:

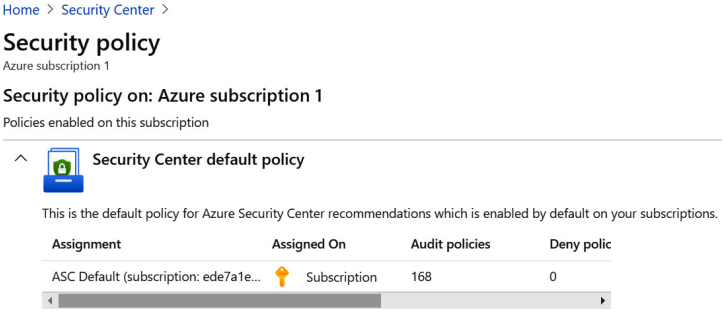

Figure 13.20: ASC default policy assignment

8. In the resulting pane, click on Parameters at the top of the screen. Look for Allowed host paths in the list of parameters and change that parameter to the following:
   { "paths": ["/tmp"]}
   The result is shown in *Figure 13.21*:

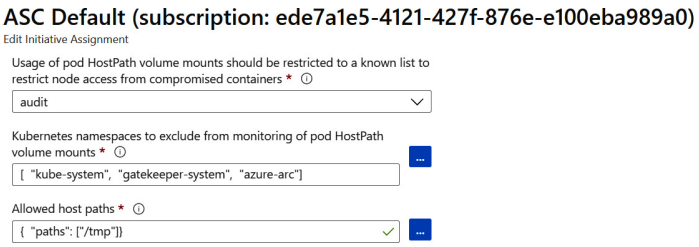

Figure 13.21: Adding a path to the allowed host paths

To apply the changes, click Review + save at the bottom of the screen and then click Save on the final screen:

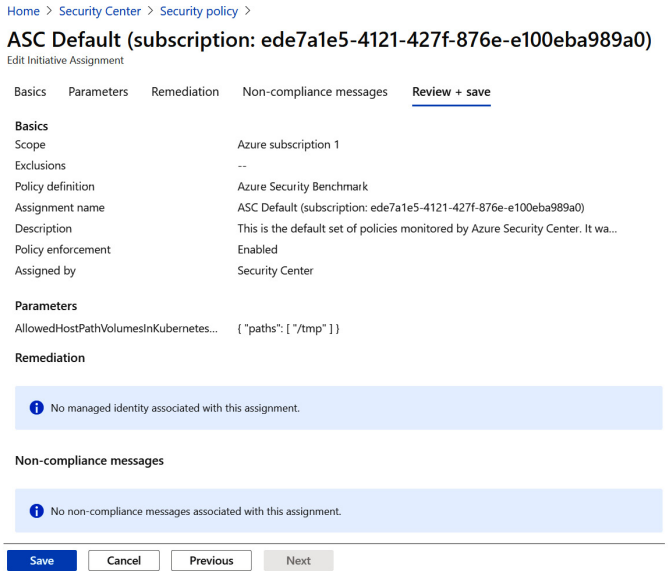

Figure 13.22: Clicking Save to confirm the policy change

It will now take about 30 minutes for the recommendation to be refreshed. After 30 minutes, this recommendation will no longer be

active, since you configured the **`/tmp`** path to be allowed.

9. There's one final recommendation worth highlighting in this list. That is Kubernetes Services Management API server should be configured with restricted access. If you remember, in *Chapter 11*, *Network security in AKS*, you configured authorized IP ranges on your cluster. This is recommended by Microsoft, and also shows up as an Azure Security Center recommendation. Click on that recommendation to get more details, as shown in *Figure 13.23*:

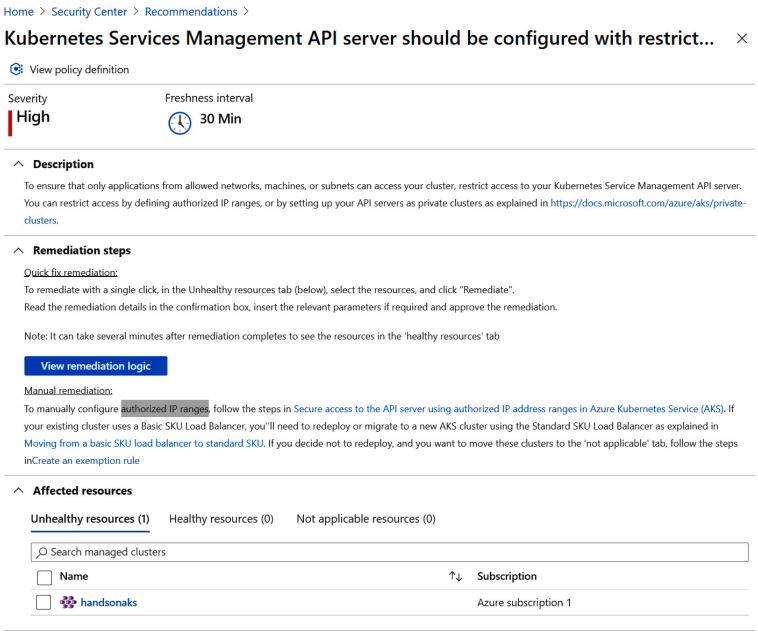

Figure 13.23: Details explaining that authorized IP ranges should be enabled

As you can see, again you get a description of the recommendation and remediation steps. The reason this recommendation is worth highlighting is that it contains a quick fix remediation within Azure Security Center. To quickly remediate this recommendation, select your AKS cluster

and click Remediate at the bottom of the screen, as shown in *Figure 13.24*:

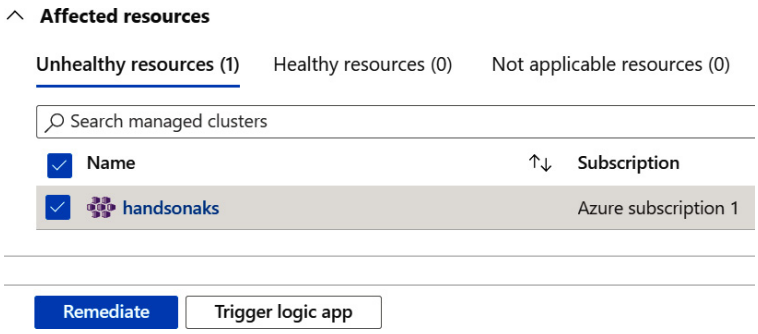

Figure 13.24: Remediating the authorized IP recommendation

This will show you a view similar to *Figure 13.25* where you could input the IP ranges you need to authorize access from.

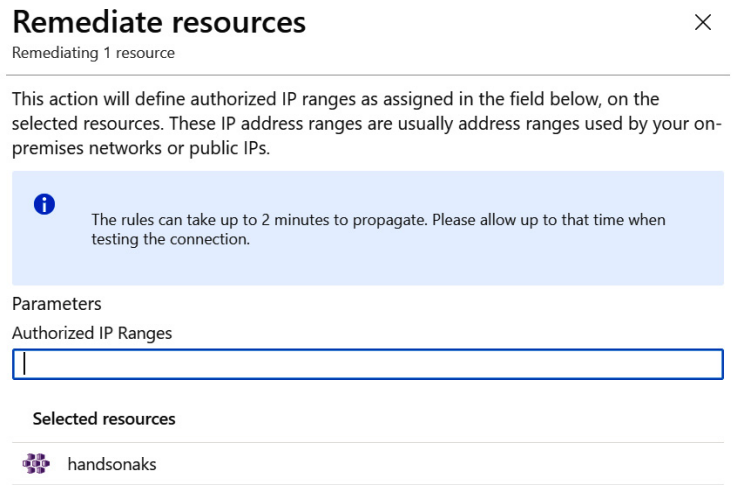

Figure 13.25: Setting up authorized IP ranges through Azure Security Center

You could configure this configuration from within Azure Security Center. Since you'll be using Cloud Shell in the next steps and next chapters and Cloud Shell does not have a fixed IP, it's not recommended to apply the remediation while working through this book. It is, however, worthwhile to show that Azure Security Center

allows you to remediate certain configuration recommendations directly from within Security Center.

You have now explored the recommendations and the secure score in Azure Security Center. If you want to learn more about this, please refer to the Azure documentation, which contains a YAML deployment example fully configured with all the recommendations: **https://docs.microsoft.com/azure/security-center/kubernetes-workload-protections**.

Neutralizing threats using Azure Defender

Now that you've explored the configuration best practices using Azure Security Center and Secure Score, you will explore how to investigate and deal with security alerts and active threats. Some of the workloads you created should have triggered security alerts by now, which you can investigate in Azure Defender.

Specifically, in the *Deploying offending workloads* section, you created three workloads that trigger security alerts in Azure Defender:

- **`crypto-miner.yaml`**: By deploying this file, you created a crypto-miner on your cluster. This crypto-miner will generate two security alerts in Azure Defender as you will see in this section. One alert will be generated by monitoring the Kubernetes cluster itself, while another alert will be generated based on DNS traffic.

- **`role.yaml`**: This file contained a cluster-wide role with very broad permissions. This will generate a security alert in Azure Defender notifying you of the risk.
- Kubernetes dashboard: You also created the Kubernetes dashboard and exposed this publicly. Azure Defender will also generate a security alert based on this.

Let's explore each of these security alerts in details:

1. To start, in Azure Security Center, click on Azure Defender in the left-hand navigation bar. This will open the Azure Defender pane in Azure Security Center. This shows you your coverage, your security alerts, and the advanced protection options, as shown in *Figure 13.26*. In this section, you will focus on the four security alerts that were generated.

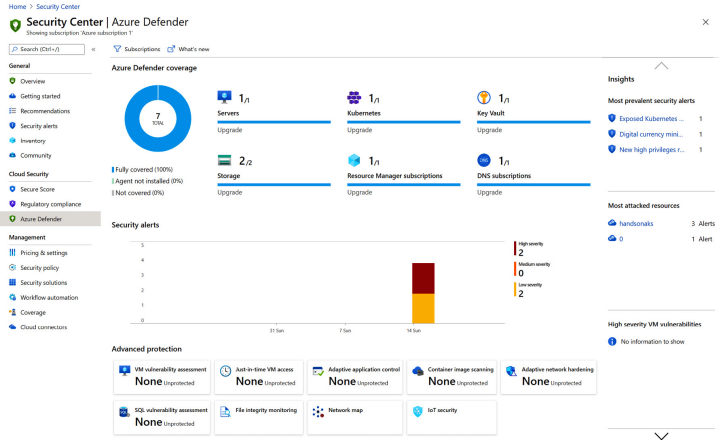

Figure 13.26: Azure Defender - Overview pane

2. To get more details about the security alerts, click anywhere on the Security alerts bar chart in the middle of the screen. That will take you to a new pane, as shown in *Figure 13.27*:

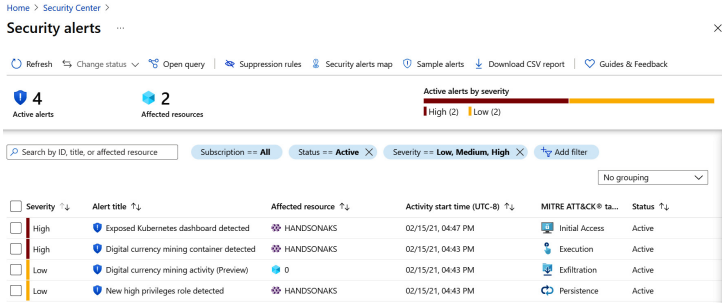

Figure 13.27: Security alerts in Azure Defender

As you can see in *Figure 13.27*, four security alerts have been triggered: one for the exposed Kubernetes dashboard, two for the crypto-miner, and one for the high-privileged roles. Let's explore each one in more detail.

3. Let's start by exploring the Exposed Kubernetes dashboard detected alert. Click on the alert title to get more details. To see all the details, click on View full details in the resulting pane, as shown in *Figure 13.28*:

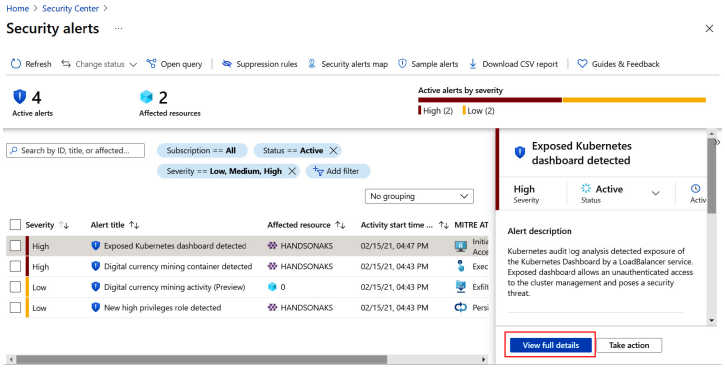

Figure 13.28: Getting full details of the alert

This will take you to a new pane, as shown in *Figure 13.29*:

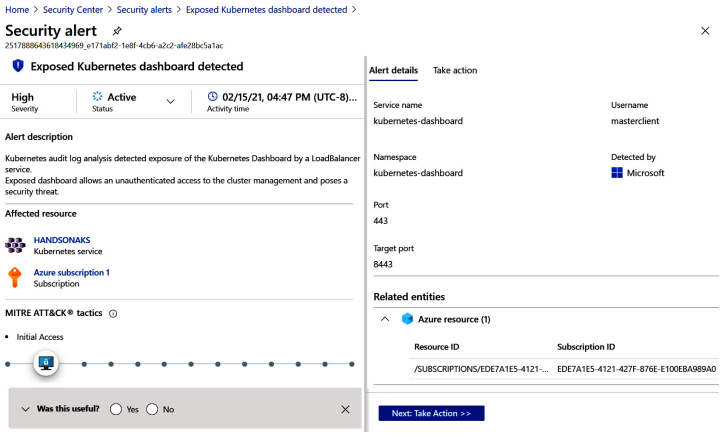

Figure 13.29: Details of the Exposed Kubernetes dashboard detected alert

This shows you a couple of information points. First, it classifies this alert as high severity, marks it as active, and also marks when it was first encountered. Next, you get a description of the alert, which in this case explains that the dashboard should not be exposed publicly. It also shows you the affected resource. Finally, you see which stage of the MITRE ATT&CK® tactics framework this attack targets. The MITRE ATT&CK® tactics framework is a framework describing multiple stages in a cyber-attack. For more information on MITRE ATT&CK® tactics, please refer to **https://attack.mitre.org/versions/v7/**.
On the right-hand side of the screen, you get more details about the alert. This contains the service name, the namespace, the port of the service, the target port exposed on the backend pods, and the affected Azure resource ID and subscription ID. If you click the Next: Take Action >> button at the bottom of the screen, you'll be taken to a new pane, as shown in *Figure 13.30*:

Figure 13.30: Security recommendations on the dashboard alert

In the Take action pane, you get information on how to mitigate the threat and how to prevent future attacks like this. Notice how the Prevent future attacks section contains a link to the security recommendations you reviewed in the previous section.

4. Let's take the suggested action and update the Kubernetes dashboard service so it is no longer of the **`LoadBalancer`** type using the following command. This command will remove the **`nodePort`** that Kubernetes set up to expose the service through the load balancer and change the type of the service back to the **`ClusterIP`** type, which is only available from within the cluster.

```
kubectl patch service \
  kubernetes-dashboard -n kubernetes-dash-
board \
  -p '{
      "spec": {
        "ports": [
```

```
      {
        "nodePort": null,
        "port": 443
      }
    ],
    "type": "ClusterIP"
  }
}'
```

Finally, you see that you can optionally trigger automated responses or suppress similar alerts in the future in case this alert is a false positive.

5. Now that you have mitigated the threat, you can dismiss the alert. That way, others using the same subscription don't see this same alert. To dismiss the alert, click on the status on the left-hand side of the screen, select Dismissed, and click OK, as shown in *Figure 13.31*:

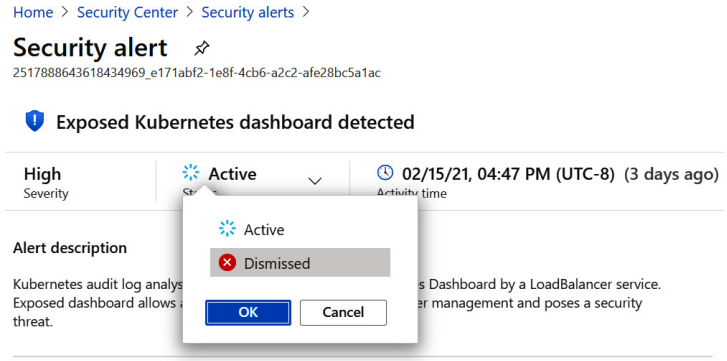

Figure 13.31: Dismissing the dashboard alert

6. Let's move on to the next alert. Close the detail panes for the dashboard alert by pressing the X at the top of the screen. Let's now focus on the first Digital currency mining container detected alert. Select that alert and click on View full details as you did with the previous alert. This will take you to a pane similar to *Figure 13.32*:

Figure 13.32: Details of the Digital currency mining container detected alert

This view contains similar details to the previous alert. As you can see, this alert is part of the Execution phase of the MITRE ATT&CK® tactics framework. On the right-hand side of the pane, you now see the name of the offending container, the image it's using, its namespace, and the name of the offending pod.
If you press the Next: Take Action >> button at the bottom of the screen, you'll be taken to the Take action view on this alert, as shown in *Figure 13.33*:

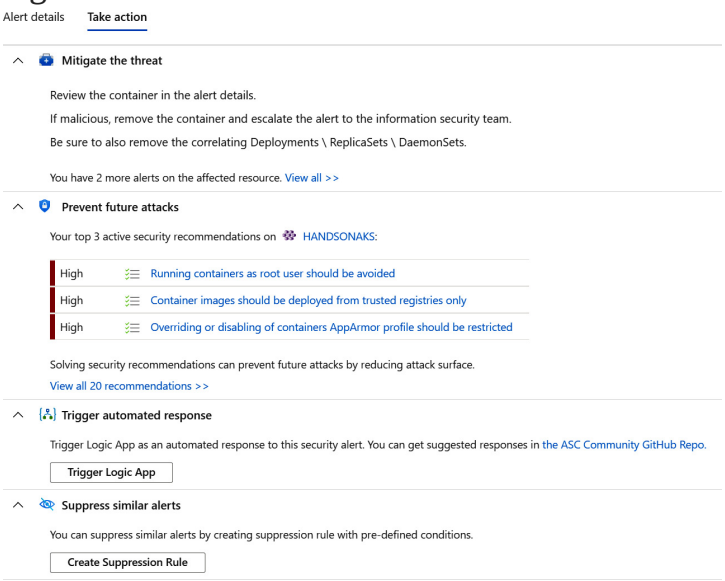

Figure 13.33: Security recommendations on the Digital currency mining container detected alert

Here, again, you see similar details to the previous alert. In the Mitigate the threat section, you get a different description of how to mitigate this ongoing threat. Don't take any mitigating action yet, since you'll need to explore one more alert related to the crypto-miner.

7. To explore that alert, close the detailed pane for the first crypto-miner alert by pressing the X at the top of the screen. Now select the second alert, which is called Digital currency mining activity (Preview). This is actually not a Kubernetes alert, but an alert based on DNS, as you can see in *Figure 13.34*:

Note

This alert will only show up if you enabled Azure Defender for DNS. If you did not enable this, you will not get this alert.

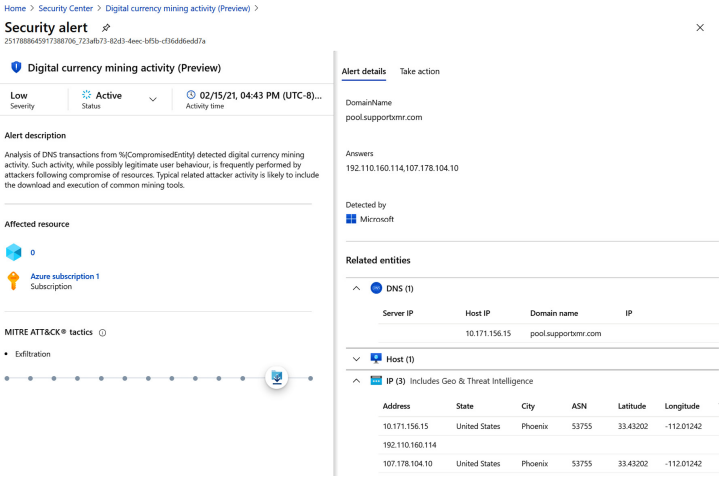

Figure 13.34: Details of the Digital currency mining activity alert

This alert was generated by Azure Defender for DNS. It shows you a number of details about the attack itself. On the right-hand side of the pane, you see more details about the attack, showing you the domain name and the IP addresses used. If you have a look at the Take action pane, you get more information about potential next steps for this attack, as shown in *Figure 13.35*:

Figure 13.35: Security recommendations on the curreny mining alert

Since this is a DNS-based alert, there are limited specifics here on which processes to inspect. However, Azure still provides you with a number of steps to run through to mitigate this threat. As you already know the process that is running this crypto-miner, you can mitigate the threat using the following command:
kubectl delete -f crypto-miner.yaml

8. This will resolve the alert. To actually mark it as resolved, you can dismiss the alert in the Azure portal. To do this, click on the status on the left-hand side of the screen, select the

Dismissed status, and click OK, as shown in *Figure 13.36*:

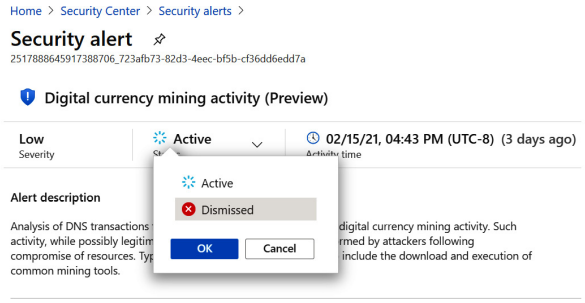

Figure 13.36: Dismissing the digital currency DNS alert

9. From the Security alerts pane, click on the last alert, which is called New high privileges role detected, and click on View full details in the resulting pane. This will bring you to a pane similar to *Figure 13.37*:

Figure 13.37: New high privileges role detected alert

This is a low-severity alert. As with the previous alerts, you get the description, the affected resource, and the phase in the MITRE ATT&CK® tactics framework, which is persistence in this case. This means that this potential attack is used by attackers to gain persistent access to your environment.

On the right-hand side, you also get the alert details, with the role name, the namespace (which in this case is the whole cluster since it is a **`ClusterRole`**), and the rules this role gives access to. If you click the Next: Take Action >> button, you'll get more information about the mitigation as well, as shown in *Figure 13.38*:

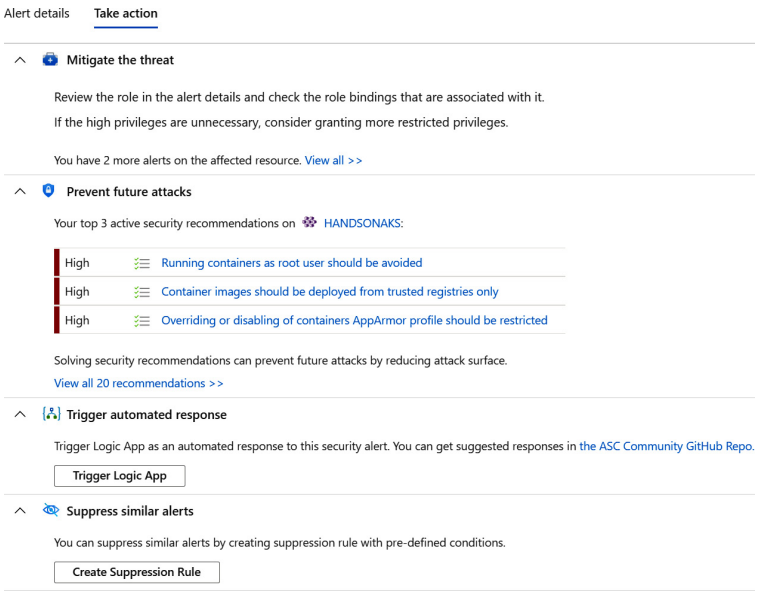

Figure 13.38: Security recommendations on the new high privileges alert

As you can see here, Azure recommends you review the role in the alerts and check any role bindings linked to this role. It is also recommended to grant more restricted privileges than the open permissions as provided in this role. Let's also remove this threat from the cluster using the following command:

```
kubectl delete -f role.yaml
```

This will delete the role from the cluster. You can also dismiss this alert, by clicking on the status on the left-hand side of the screen, selecting the

Dismissed state, and clicking OK, as shown in *Figure 13.39*:

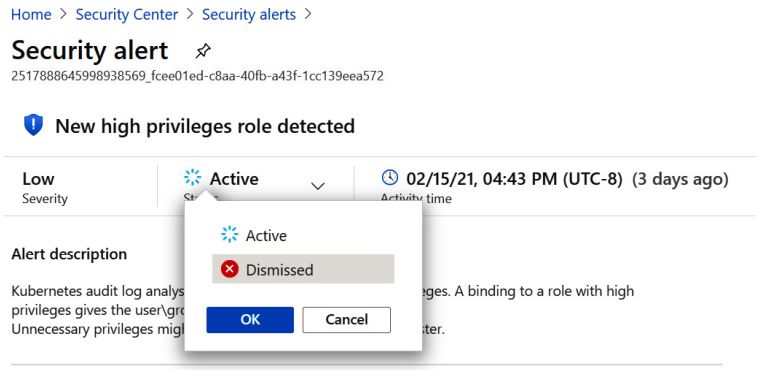

Figure 13.39: Dismissing the New high privileges alert

This covers all the alerts that were generated by the resources you created earlier in this chapter. The resources linked to the alerts were already deleted as part of the remediation, but let's also delete the other resources that were created in this chapter:

kubectl delete -f escalation.yaml

kubectl delete -f host-volume.yaml

kubectl delete -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/dep

And that concludes this chapter.

Summary

In this chapter, you explored Azure Security Center and Azure Defender. Azure Security Center is an infrastructure security monitoring platform. It provides both monitoring of security configuration as well as monitoring of any potential ongoing threats. To monitor workloads in a

Kubernetes cluster, Azure Security Center makes use of Azure Policy for AKS.

To start, you enabled Azure Policy for AKS. You then enabled Azure Security Center and Azure Defender on your subscription.

You then created five harmful workloads on your cluster. Some of those caused configuration recommendations in Azure Security Center. Some others even caused security alerts to be triggered in Azure Defender. You explored four security alerts and followed the mitigation steps recommended to resolve these alerts.