

{

"fqdns": "",

"id": "/subscriptions/ed3ca12d-4d7b-452c-8952-839cabcb406a/resourceGroups/learn-1b0ae239-83c8-471c-86a9-ca11b723ead9/providers/Microsoft.Compute/virtualMachines/my-vm",

"location": "westus",

"macAddress": "00-0D-3A-32-2C-3C",

"powerState": "VM running",

"privateIpAddress": "10.0.0.4",

"publicIpAddress": "13.64.58.189",

"resourceGroup": "learn-1b0ae239-83c8-471c-86a9-ca11b723ead9",

"zones": ""

}

{

"autoUpgradeMinorVersion": true,

"enableAutomaticUpgrade": null,

"forceUpdateTag": null,

"id": "/subscriptions/ed3ca12d-4d7b-452c-8952-839cabcb406a/resourceGroups/learn-1b0ae239-83c8-471c-86a9-ca11b723ead9/providers/Microsoft.Compute/virtualMachines/my-vm/extensions/customScript",

"instanceView": null,

"location": "westus",

"name": "customScript",

"protectedSettings": null,

"provisioningState": "Succeeded",

"publisher": "Microsoft.Azure.Extensions",

"resourceGroup": "learn-1b0ae239-83c8-471c-86a9-ca11b723ead9",

"settings": {

"fileUris": [

"https://raw.githubusercontent.com/MicrosoftDocs/mslearn-welcome-to-azure/master/configure-nginx.sh"

]

},

"tags": null,

"type": "Microsoft.Compute/virtualMachines/extensions",

"typeHandlerVersion": "2.1",

"typePropertiesType": "customScript"

}

#!/bin/bash

# Update apt cache.

sudo apt-get update

# Install Nginx.

sudo apt-get install -y nginx

# Set the home page.

echo "<html><body><h2>Welcome to Azure! My name is $(hostname).</h2></body></html>" | sudo tee -a /var/www/html/index.html

[

{

"access": "Allow",

"description": null,

"destinationAddressPrefix": "\*",

"destinationAddressPrefixes": [],

"destinationApplicationSecurityGroups": null,

"destinationPortRange": "22",

"destinationPortRanges": [],

"direction": "Inbound",

"etag": "W/\"146bc2b4-f16e-4a14-b7f1-e7e133a4867b\"",

"id": "/subscriptions/ed3ca12d-4d7b-452c-8952-839cabcb406a/resourceGroups/learn-1b0ae239-83c8-471c-86a9-ca11b723ead9/providers/Microsoft.Network/networkSecurityGroups/my-vmNSG/securityRules/default-allow-ssh",

"name": "default-allow-ssh",

"priority": 1000,

"protocol": "Tcp",

"provisioningState": "Succeeded",

"resourceGroup": "learn-1b0ae239-83c8-471c-86a9-ca11b723ead9",

"sourceAddressPrefix": "\*",

"sourceAddressPrefixes": [],

"sourceApplicationSecurityGroups": null,

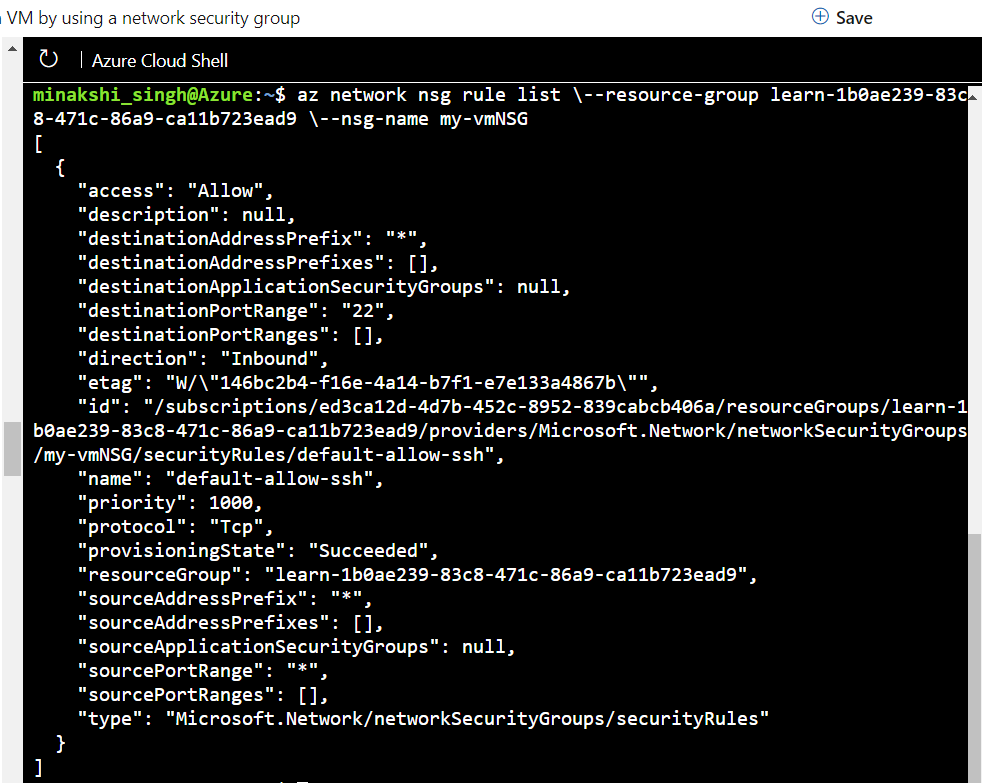
"sourcePortRange": "\*",

"sourcePortRanges": [],

"type": "Microsoft.Network/networkSecurityGroups/securityRules"

}

]



minakshi\_singh@Azure:~$ az network nsg rule create \--resource-group learn-1b0ae239-83c8-471c-86a9-ca11b723ead9 \--nsg-name my-vmNSG \--name allow-http \--protocol tcp \--priority 100 \--destination-port-ranges 80 \--access Allow

{

"access": "Allow",

"description": null,

"destinationAddressPrefix": "\*",

"destinationAddressPrefixes": [],

"destinationApplicationSecurityGroups": null,

"destinationPortRange": "80",

"destinationPortRanges": [],

"direction": "Inbound",

"etag": "W/\"7c4ac754-cd91-4f91-88ab-993523147a3b\"",

"id": "/subscriptions/ed3ca12d-4d7b-452c-8952-839cabcb406a/resourceGroups/learn-1b0ae239-83c8-471c-86a9-ca11b723ead9/providers/Microsoft.Network/networkSecurityGroups/my-vmNSG/securityRules/allow-http",

"name": "allow-http",

"priority": 100,

"protocol": "Tcp",

"provisioningState": "Succeeded",

"resourceGroup": "learn-1b0ae239-83c8-471c-86a9-ca11b723ead9",

"sourceAddressPrefix": "\*",

"sourceAddressPrefixes": [],

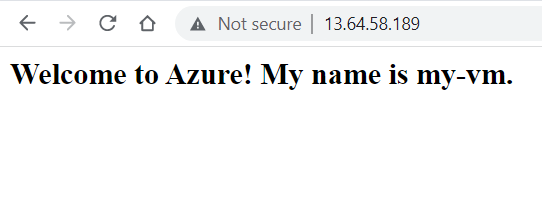
"sourceApplicationSecurityGroups": null,

"sourcePortRange": "\*",

"sourcePortRanges": [],

"type": "Microsoft.Network/networkSecurityGroups/securityRules"

}



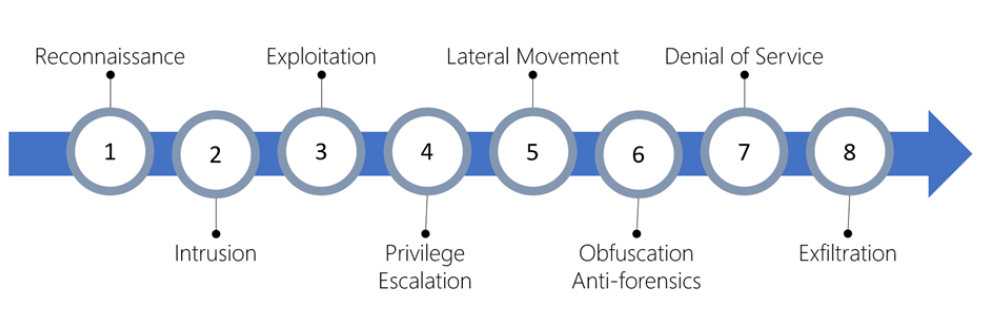
**What is a security alert?**

Alerts are the notifications that Security Center generates when it detects threats on your resources. Security Center prioritizes and lists the alerts, along with the information needed for you to investigate the problem quickly. Security Center also provides recommendations for how you can remediate an attack.

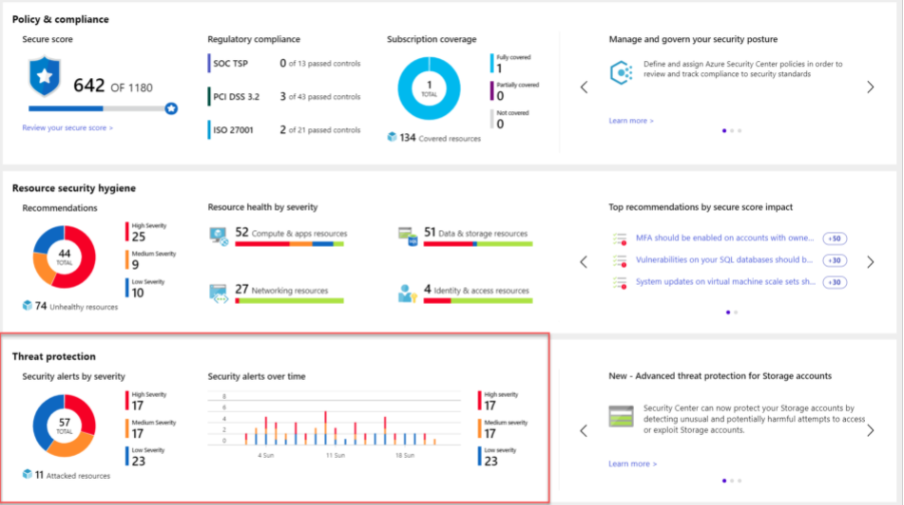
**Alert types and the Cyber Kill Chain**

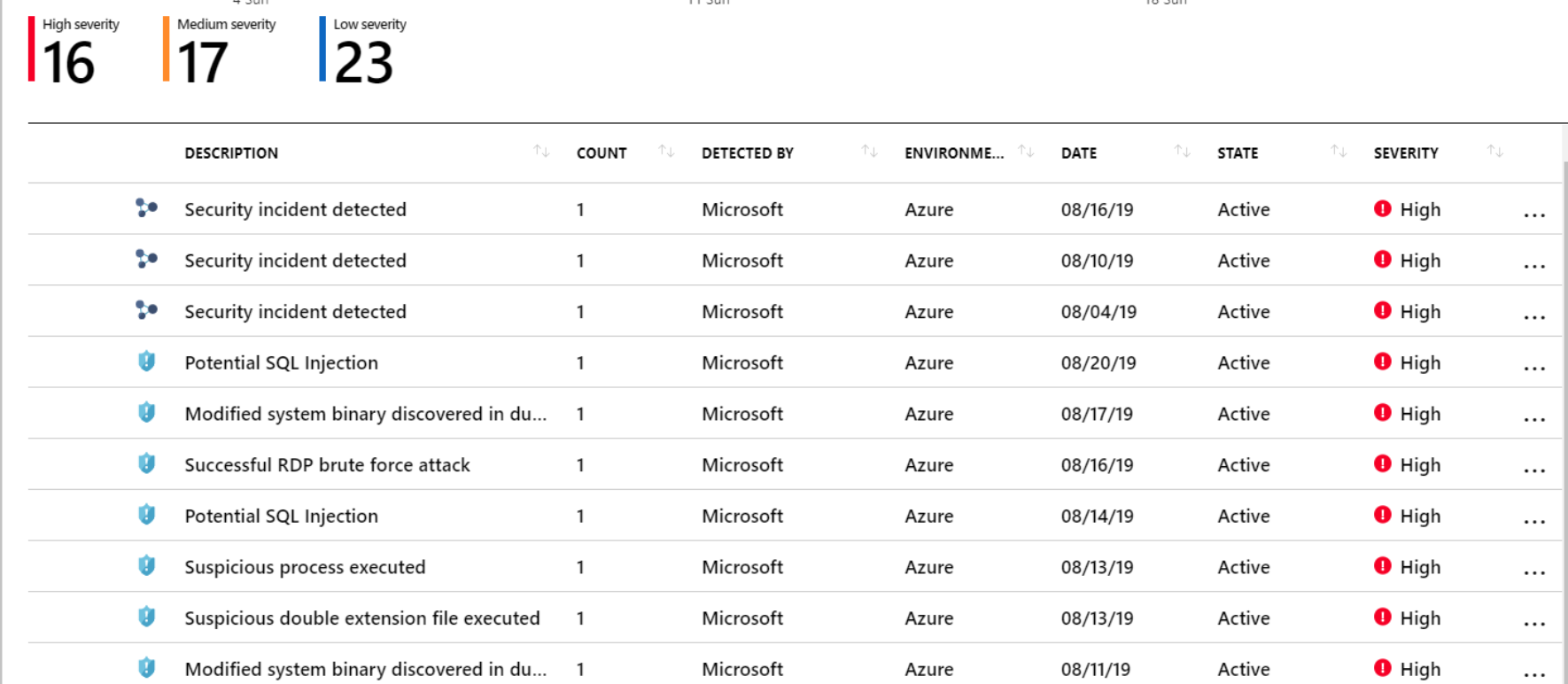
Azure Security Center provides a variety of alerts that align with the stages of the *cyber kill chain*. The cyber kill chain is a series of steps that trace the stages of a cyberattack from the early reconnaissance stages to the exfiltration of data. The kill chain was created by Lockheed Martin and is modeled off a military framework established to identify and engage enemy targets.

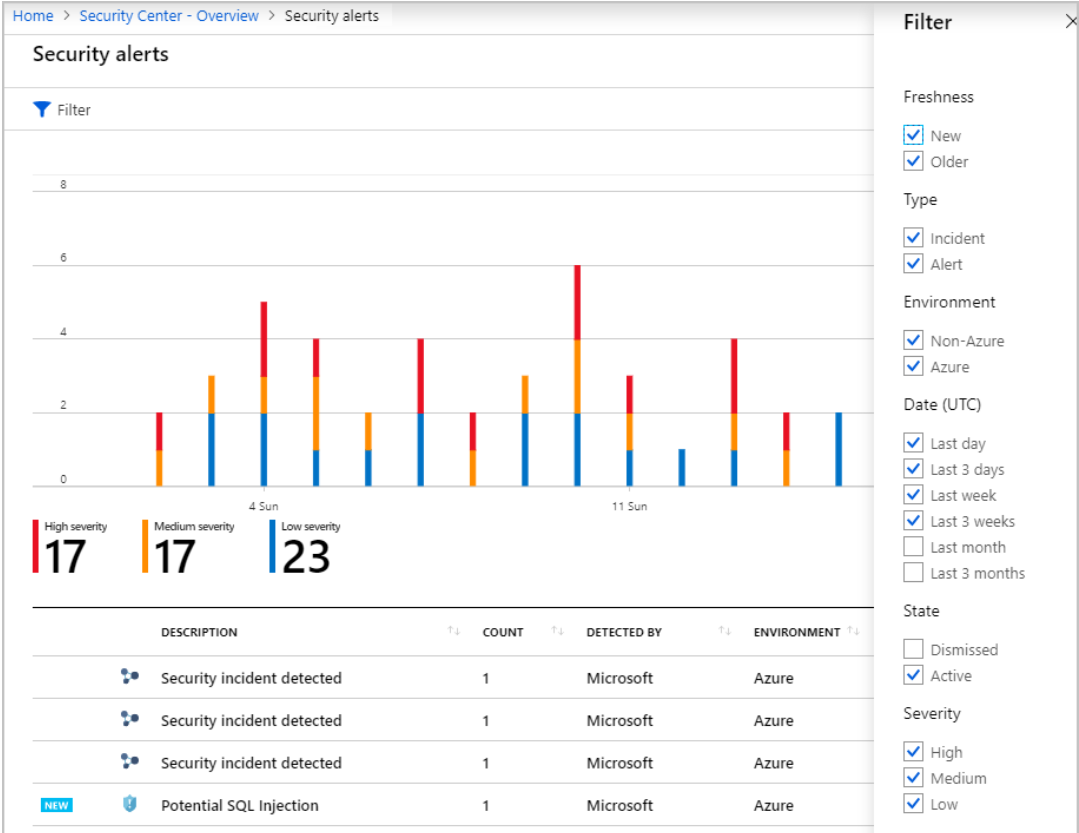
The kill chain consists of eight phases as shown in the below image. Different types of attacks are associated with each stage, and they target various subsystems. All the common attack vectors from brute force logins to viruses and worms trigger activity on the cyber kill chain.

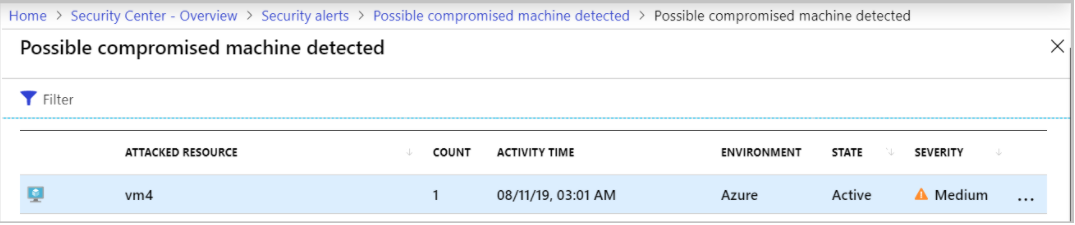


1. **Reconnaissance:** The observation stage where attackers assess your network and services to identify possible targets and techniques to gain entry.
2. **Intrusion:** Attackers use knowledge gained in the reconnaissance phase to get access to a part of your network. This often involves exploring a flaw or security hole.
3. **Exploitation:** This phase involves exploiting vulnerabilities and inserting malicious code onto the system to get more access.
4. **Privilege Escalation:** Attackers often try to gain administrative access to compromised systems so they can get access to more critical data and move into other connected systems.
5. **Lateral Movement:** This is the act of moving laterally to connected servers and gain greater access to potential data.
6. **Obfuscation / Anti-forensics:** To successfully pull off a cyberattack, attackers need to cover their entry. They will often compromise data and clear audit logs to try to prevent detection by any security team.
7. **Denial of Service:** This phase involves disruption of normal access for users and systems to keep the attack from being monitored, tracked, or blocked.
8. **Exfiltration:** The final extraction stage: getting valuable data out of the compromised systems.









After reviewing this information, a security engineer can select an attacked resource to get specific information about:

1. **What happened?** (*Possible compromised machine detected*)
2. **When did it happen?** (*Sunday, August 11 2019 3:01:00 AM*)
3. **What resource was attacked?** (*vm4*)
4. **Where is the resource located?** (*Azure*)
5. **What should you do about it?** (Remediation steps)

**Incident Response Plan (IRP):**

An incident response plan (IRP) allows you to identify and minimize the damage, reduce the cost, and fix the cause of a security attack. A well-designed IRP provides step-by-step instructions for handling an incident and ensures that the security team responds using an established set of procedures, that the right people are involved, and that proper communication channels are informed.

**What is Workflow Automation in Azure Security Center?**

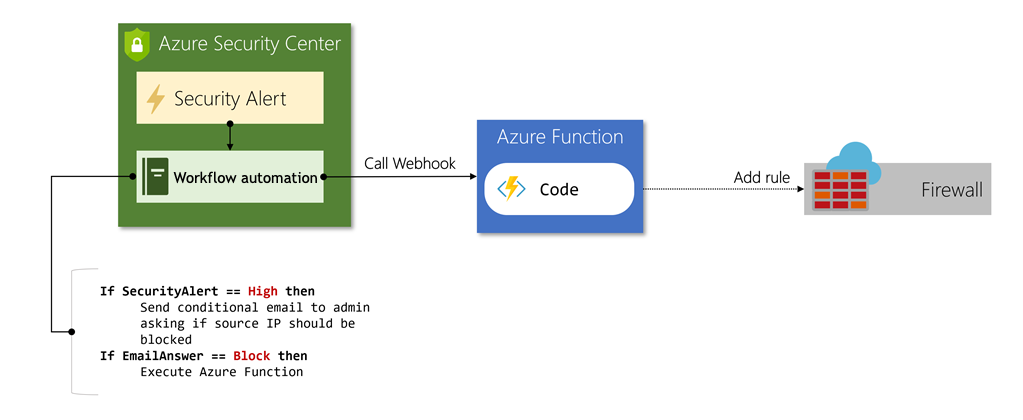
A workflow automation is a collection of grouped procedures that the security response team can execute with a single click in Security Center when a specific alert is detected. These actions are *not* automatically triggered - they require human interaction to run.

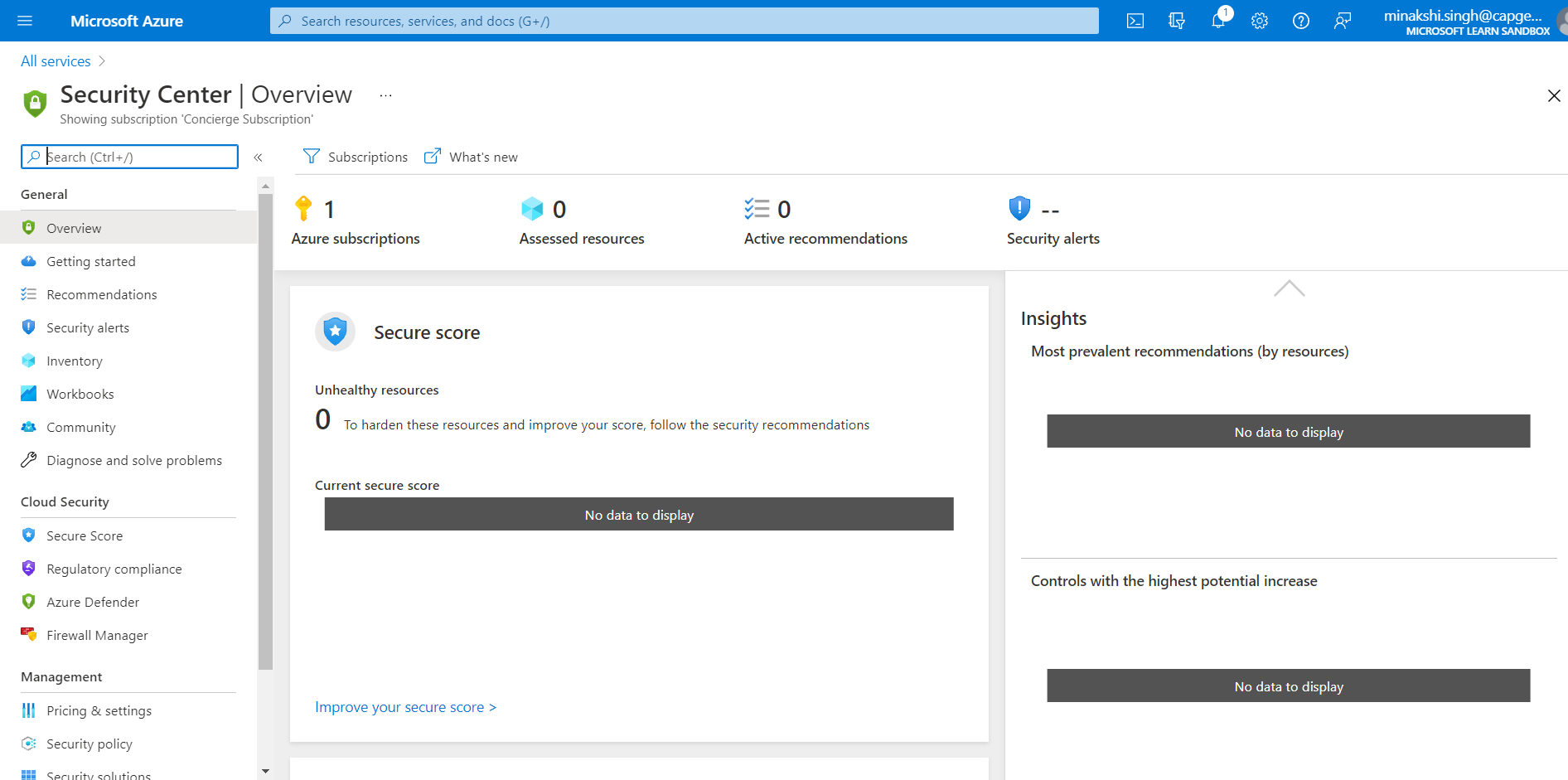
Workflow automations are built on **Azure Logic Apps**. You can easily customize the logic and workflow using the visual workflow designer. You can either start with an existing Logic App or create a new Logic App and use Security Center to trigger it when an alert is generated.

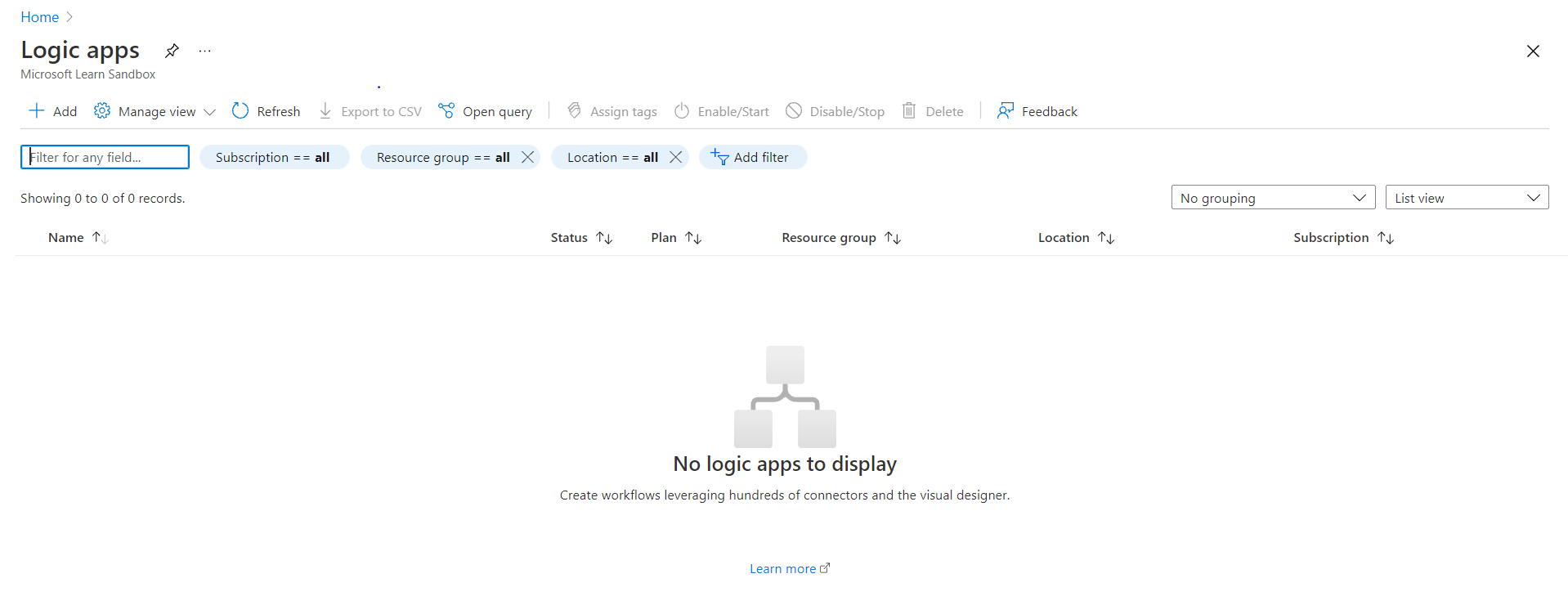
Some of the pre-built actions are:

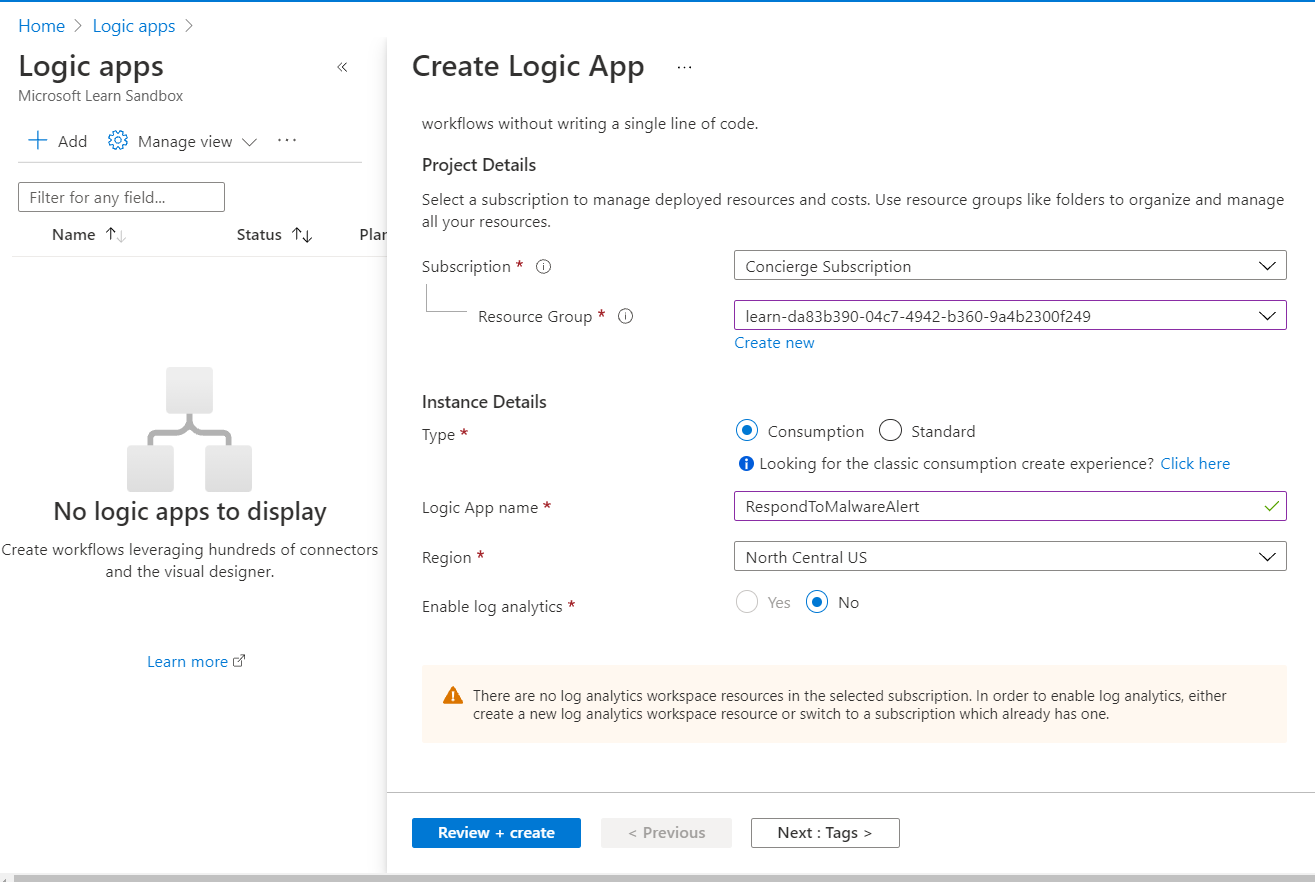
* Create an automated incident report in another system, filling in fields from the active alert
* Email a distribution group with details about the active alert(s)
* Send a notification to a Teams or Slack channel

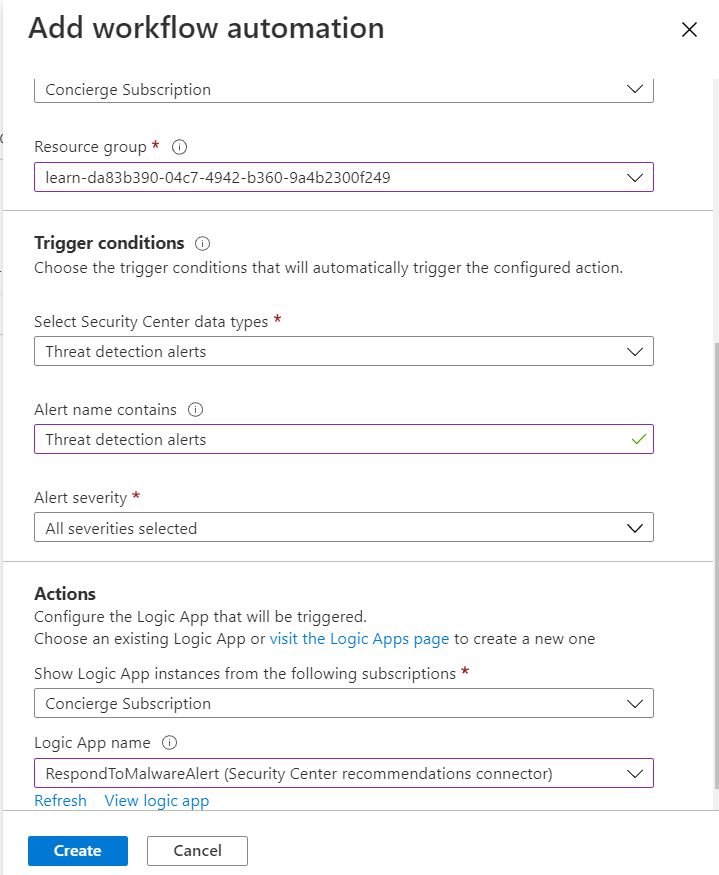
However, because Logic Apps can integrate Azure Functions and webhooks, the possible actions are endless. Imagine a case where a client is misusing your service, and it's generating an alert in Security Center. We can create a custom Azure Function to take the source IP address from the alert and create a rule in our firewall to block all inbound traffic from that address. A visual representation of this might look something like this:





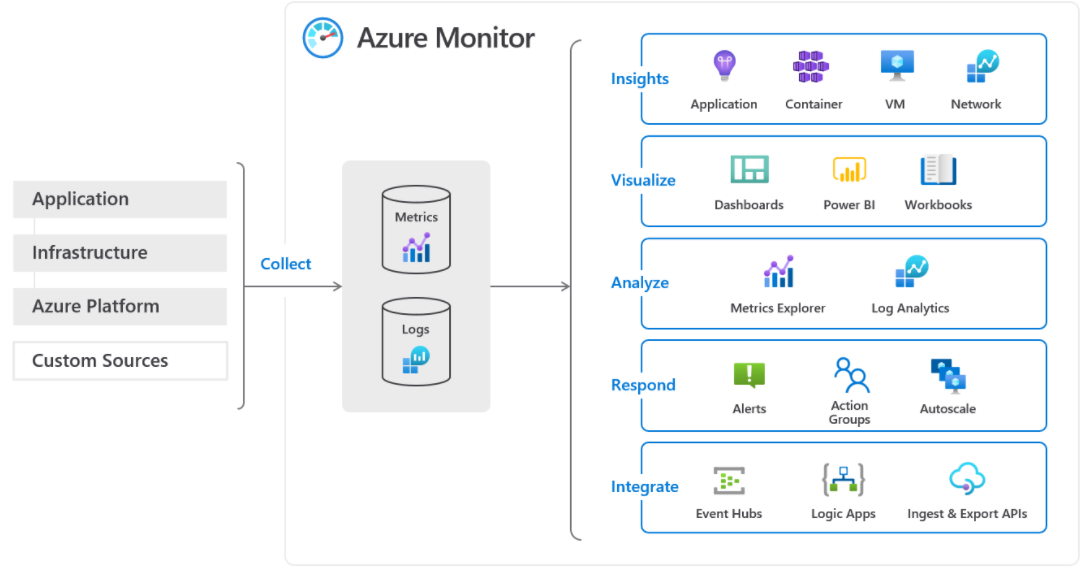






## What is Azure Monitor?

Azure Monitor is a service for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. It helps you maximize the availability and performance of your applications and services. Azure Monitor enables you to detect and diagnose application, infrastructure, and platform issues. You can analyze metrics and logs from monitored resources. And it also supports operational workflows with alerts and automated actions, and enables you to create visualizations such as dashboards and reports.



Azure Monitor starts with collecting telemetry, this data includes application layer data and infrastructure performance data from VM guest operating systems and containers. Additionally, Azure Monitor collects directly from the Azure platform resources, and you can also ingest your own custom data using APIs. The collected data is stored in centralized and fully-managed data stores - Azure Monitor Metrics for numerical time-series values; and Azure Monitor Log Analytics workspaces for resource logs. Metrics are automatically collected and stored for Azure resources, but user configuration is required to send and store resource logs. After the data is collected, you can choose how you consume, analyze, and respond.