

# Windows 10 Azure SOC Lab Project

By Bear Schneider

## Goal

The goal of this lab is to create a virtual Windows machine that is open to the internet and vulnerable to attackers that may attempt to gain access. Creating this environment will allow me to gain practical experience in observing traffic to open ports, reading logs, using Security Information and Event Management (SIEM) tools, writing custom queries, and gaining some hands-on experience with Microsoft Azure and its various tools.

## Summary

**Creating a Virtual Machine (VM):** We began by setting up a Windows 10 VM in Azure. The VM is set up with Remote Desktop Protocol (RDP) open, making it susceptible to security events such as brute force attacks.

**Deploying Microsoft Sentinel:** Once the VM was spun up, we next deployed Microsoft Sentinel, which acted as the SIEM solution. Sentinel was then added to a Log Analytics workspace, and configurations were made to ensure it could collect security events from the VM.

**Setting Up Data Collection:** We set up a data connector that pulled event logs from the VM, which were then sent to Sentinel for analysis.

**Creating Custom Sentinel Rules:** We then configured a custom Sentinel rule to monitor RDP login attempts, specifically successful sign-ins. The rule triggered alerts when unusual activity occurred, such as non-system accounts logging in via RDP.

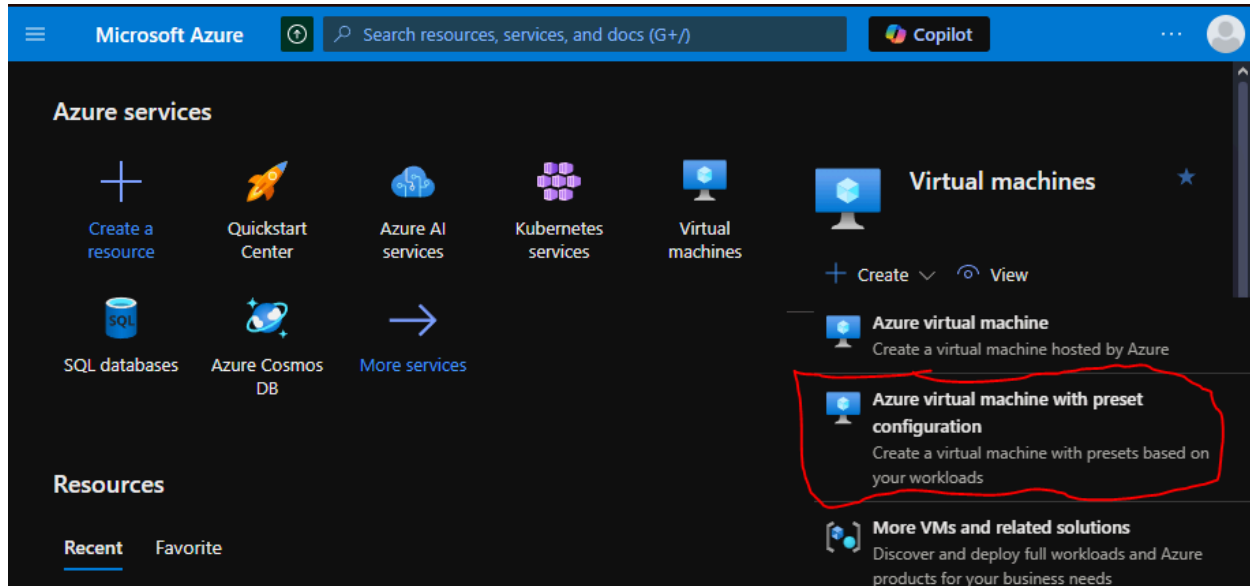
**Testing and Incident Generation:** After configuring the rule, the system was tested by logging in to the VM via RDP. This generated an alert, which could be seen in the Sentinel interface as an incident, demonstrating the system's ability to detect and respond to security events.

**Logging and Investigation:** The VM was left running for a week to collect logs and alerts.

**Incident Response:**

# Process

The process starts by signing up for an Azure free trial, giving \$200 in credit for a month. From here, we deployed a machine using Windows 10 Pro with default configurations.



**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ Azure subscription 1

Resource group \* ⓘ (New) BearVM\_group

[Create new](#)

**Instance details**

Virtual machine name \* ⓘ BearVM ✓

Region \* ⓘ (US) West US

Availability options ⓘ No infrastructure redundancy required

Security type ⓘ Trusted launch virtual machines

[Configure security features](#)

Image \* ⓘ Windows 10 Pro, version 22H2 - x64 Gen2 (free services eligible)

[See all images](#) | [Configure VM generation](#)

VM architecture ⓘ

☐ Arm64

☒ x64

**Administrator account**

Username \* ⓘ  ✓

Password \*  ✓

Confirm password \*  ✓

Public inbound ports \* ⓘ ☐ None ☒ Allow selected ports

Select inbound ports \*  ✓

⚠ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

An important note for this project, as seen in the above screenshot, we are leaving Remote Desktop Protocol (RDP, port 3389) open and allowing that traffic so we can capture login attempts. We are also giving this admin account a short name and insecure password.

RANK	PASSWORD	TIME TO CRACK IT
82	Password@123	2 Minutes

This is listed as the 82nd most common password, takes approximately 2 minutes to crack (couldn't make it too easy), and it just meets Azure's password policy requirements.

This is a new experience. [Please provide feedback](#)

BearVM-vnet / default 0 (Configure)

Public IP address 13.88.22.68 Network security group BearVM-nsg

Private IP address 10.0.0.4 Accelerated networking Disabled

Admin security rules 0 (Configure) Effective security rules 0

---

Rules [Collapse all](#)

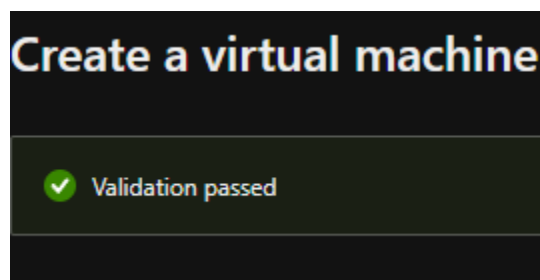
Network security group BearVM-nsg (attached to networkinterface: bearvm386) [+ Create port rule](#)

Impacts 0 subnets, 1 network interfaces

Search rules Source == all Destination == all Protocol == all Action == all

Priority ↑	Name	Port	Protocol	Source
Inbound port rules (4)				
300	RDP	3389	TCP	Any
65000	AllowVnetInBound	Any	Any	VirtualNetwork
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer
65500	DenyAllInBound	Any	Any	Any

Here you can see the public IP address and confirmation that RDP is open.



Upon successful deployment, our next steps were to create a data analytics workspace, and to configure both Microsoft Sentinel and a data connector to allow Sentinel to ingest logs for the VM.

Home > Microsoft Sentinel


# Microsoft Sentinel

Default Directory

[+ Create](#) [Manage view](#) [...](#)

Filter for any field...

Name [↑↓](#)

 Bear-LogAnalytics



# Microsoft Sentinel

Selected workspace: 'bear-loganalytics'


[Search](#)


> General


> Threat management


> Content management


✓ Configuration


 Workspace manager  
(Preview)


 Data connectors

 Analytics

 Summary rules (Preview)

 Watchlist

 Automation

 Settings



## Windows Security Events

Microsoft Provider	 Microsoft Support	 3.0.9 Version
-----------------------	--	---

Description




**Note:** Please refer to the following before installing the solution:

- Review the solution [Release Notes](#)

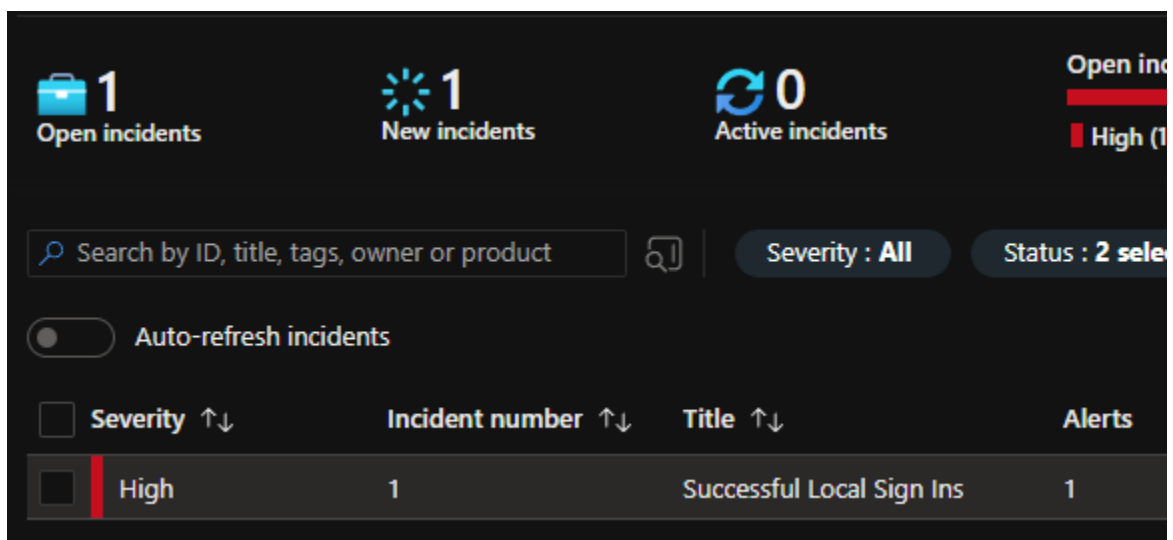
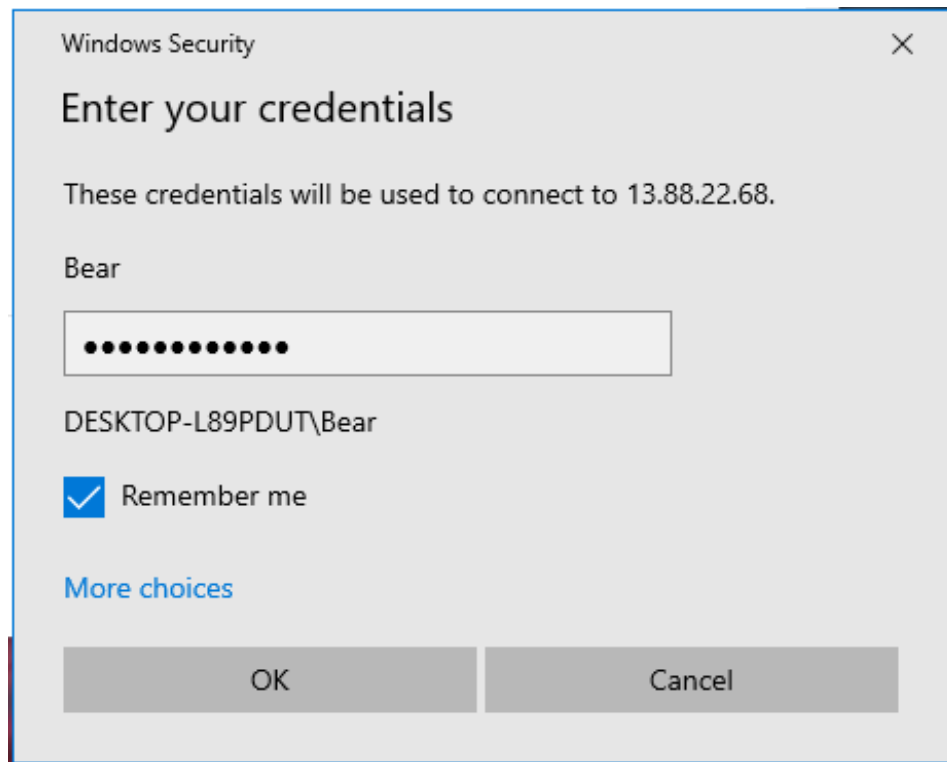
The Windows Security Events solution for Microsoft Sentinel allows you to ingest Security events from your Windows machines using the Windows Agent into Microsoft Sentinel. This solution includes two (2) data connectors to help ingest the logs.

**1. Windows Security Events via AMA** - This data connector helps in ingesting Security Events logs into your Log Analytics Workspace using the new Azure Monitor Agent. Learn more about ingesting using the new Azure Monitor Agent [here](#). **Microsoft recommends using this Data Connector.**

We used Windows Security Events via AMA as our data connector.

Analytics rule details	
Name	Successful Local Sign Ins
Description	
MITRE ATT&CK	 Initial Access
Severity	 High
Status	 Enabled
Analytics rule settings	
Rule query	SecurityEvent   where Activity contains "successful" and Account !contains "system"
Rule frequency	Run query every <b>5 minutes</b>
Rule period	Last <b>5 minutes</b> data
Rule start time	Automatic
Rule threshold	Trigger alert if query returns <b>more than 0</b> results
Event grouping	Group all events into a single alert

Next, we created a rule to send an alert upon successful logins. We exclude any events that come from the system accessing the VM, and later we refined the Account section of the query to only return logs that had accounts that contained “Bear”, as management tools sent false positives, and any non-automated account fields start with “BearVM/”. This query runs every 5 minutes and returns any relevant logs as a single alert.



We logged in to the machine to test the alert system, and the alert did show up on the dashboard.


The next step was to wait and analyze the logs as they came in.

Account	IpAddress ↑		
\BHALL	31.43.185.39	\TAMIRES	45.143.201.131
\BPSC	31.43.185.39	\USER	45.143.201.131
\STELZER	31.43.185.39	\PESSOAL	45.143.201.131
\JAYCN	31.43.185.39	\PLANO	45.143.201.131
\MIMO	31.43.185.39	\USER	45.143.201.131
\KONICA	31.43.185.39	\RH01	45.143.201.131
\DIRECCION	31.43.185.39	\CARMEN	45.143.201.131
\SHAFAEY	31.43.185.39	\MARIA	45.143.201.131
\NAIARA	31.43.185.39	\OAO	45.143.201.131
\LOGISTIC	31.43.185.39	\PROSOFT	45.143.201.131
\ROZNICA	31.43.185.39	\ROBERTO	45.143.201.131
\LPF	31.43.185.39	\MANUEL	45.143.201.131
\GEOVANE	31.43.185.39	\ESTAGIO	45.143.201.131
\OMNICOM	31.43.185.39	\HR	45.143.201.131
\GIANCARLOF	31.43.185.39	\IGOR	45.143.201.131
\PORTARIA	31.43.185.39	\ALBERTO	45.143.201.131
\ADISYON	31.43.185.39	\JAVIER	45.143.201.131

Within a few hours, we had several thousand brute force attempts. These are the top 2 IP's, and a couple guesses from iplocation:













**Geolocation data from** **IP2Location** **Product: DB6, 2024-10-15**

 <b>IP ADDRESS:</b> 31.43.185.39	 <b>ISP:</b> FOP Dmytro Nedilskyi
 <b>COUNTRY:</b> Netherlands 	 <b>ORGANIZATION:</b> Not available
 <b>REGION:</b> Noord-Holland	 <b>LATITUDE:</b> 52.3785
 <b>CITY:</b> Amsterdam	 <b>LONGITUDE:</b> 4.9000










[Incorrect location? Contact IP2Location](#)  [view map](#)


**Geolocation data from** **ipinfo.io** **Product: API, real-time**

 <b>IP ADDRESS:</b> 31.43.185.39	 <b>ISP:</b> Not available
 <b>COUNTRY:</b> Ukraine 	 <b>ORGANIZATION:</b> AS211736 FOP Dmytro Nedilskyi
 <b>REGION:</b> Kyiv City	 <b>LATITUDE:</b> 50.4547
 <b>CITY:</b> Kyiv	 <b>LONGITUDE:</b> 30.5238




[Incorrect location? Contact ipinfo.io](#)  [view map](#)

**Geolocation data from** **ipbase.com** **Product: API, real-time**

 <b>IP ADDRESS:</b> 45.143.201.131	 <b>ISP:</b> TOV E-RISHENNYA
 <b>COUNTRY:</b> Belgium 	 <b>ORGANIZATION:</b> TOV E-RISHENNYA
 <b>REGION:</b> Brussels	 <b>LATITUDE:</b> 50.8504
 <b>CITY:</b> Brussels	 <b>LONGITUDE:</b> 4.3488

[Incorrect location? Contact ipbase.com](#)  [view map](#)

**Geolocation data from** **criminalip.io** **Product: API, real-time**

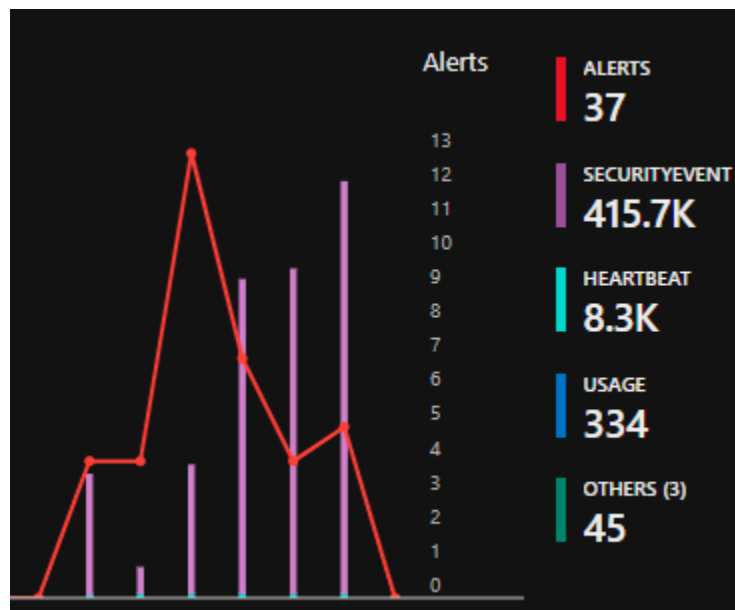
 <b>IP ADDRESS:</b> 45.143.201.131	 <b>ISP:</b> Not available
 <b>COUNTRY:</b> Russia 	 <b>ORGANIZATION:</b> Tov E-rishennya
 <b>REGION:</b> Not available	 <b>LATITUDE:</b> 55.7386
 <b>CITY:</b> Not available	 <b>LONGITUDE:</b> 37.6068

*Personal note: This last site is called "criminalip.io" and was the only source with Russia as the address, I found that humorous*

After two days, we did encounter some successful attempts and were properly alerted.



We left the VM running for a week, and here are the results:



In total, 37 separate events with 57 successful logins from different sources, and 415,000 events in total