

Building a Cloud-Based SIEM Home Lab with Azure and Microsoft Sentinel

Image unavailable: Microsoft Azure. <https://oxen.tech/wp-content/uploads/2020/09/microsoftazure-featured.png>



Background

While browsing LinkedIn one day, I saw a post referencing a SOC/SIEM home lab that he created and the SIEM had recorded hundreds of external connection attempts overnight. That example motivated me to build a similar environment in Azure to analyze real-world logs.

Overview

In this lab, I will be utilizing Azure to deploy and configure a public-facing target virtual machine, and forward host and platform logs to a Log Analytics workspace running Microsoft Sentinel.

Lab Architecture and Resources

In [Azure](#), I provisioned a resource group to store my lab resources (Windows 10 virtual machine, network security group, and a virtual network) for the lab environment.

Microsoft Azure

Upgrade

Home > Resource Manager | Resource groups >

Create a resource group ...

BasicsTagsReview + create

Automation Link

Basics

Subscription	Azure subscription 1
Resource group name	SOC-LAB
Region	East US 2

Tags

None

Within that resource group, I created a virtual network for our virtual machine.

Home >

Vnet-soc-lab1Virtual network

Analyze connectivity for my Virtual NetworkRetrieve detailed routing information for troubleshootingEvaluate routing for this virtual network

Search

MoveDeleteRefreshGive feedback

OverviewActivity logAccess control (IAM)TagsDiagnose and solve problemsResource visualizerSettings

Essentials

Resource group (move): BG-SOC-LabLocation (move): East US 2Subscription (move): Azure subscription 1Subscription ID: Tags (edit): Add tags

Address space: 10.0.0.0/16Subnets: 1 subnetDNS servers: Azure provided DNS serviceBGP community string: ConfigureVirtual network ID:

JSON View

Now that the resource group and the virtual network has been created in Azure, I deployed a legacy Windows 10 Pro virtual machine as the target to maximize noise and replicate the behavior of a poorly maintained endpoint.

Exposure and telemetry collection

NARO-WEBPRODEAST1
Virtual machine

Help me copy this VM in any region | Manage this VM with Azure CLI

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

Connect

Networking

Settings

Availability + scale

Size

Availability + scaling

Security

Identity

Microsoft Defender for Cloud

Backup + disaster recovery

Operations

Monitoring

Automation

Help

Add or remove favorites by pressing Ctrl+Shift+F

NARO-WEBPRODEAST1 virtual machine agent status is not ready. Troubleshoot the issue →

Help me copy this VM in any region

Connect | Start | Restart | Stop | Hibernate | Capture | Delete | Refresh | Open in mobile | Feedback | CLI / PS

Essentials [JSON View](#)

Resource group (move) : [RG-SOC-LAB](#)

Status : Running

Location : East US 2 (Zone 2)

Subscription (move) : [Azure subscription 1](#)

Subscription ID : [REDACTED]

Availability zone : 2

Operating system : Windows

Size : Standard B1s (1 vcpu, 1 GiB memory)

Primary NIC public IP : [REDACTED]

Virtual network/subnet : [Vnet-soc-lab1/default](#)

DNS name : [Not configured](#)

Health state : -

Time created : 10/16/2025, 10:14 PM UTC

Tags (edit) : [Add tags](#)

Properties | Monitoring | Capabilities (8) | Recommendations (5) | Tutorials

Virtual machine

Computer name : NARO-WEBPRODEAS

Operating system : Windows

VM generation : V2

VM architecture : x64

Agent status : Not Ready

Agent version : Unknown

Hibernation : Disabled

Host group : -

Host : -

Networking

Public IP address : [REDACTED] (Network interface [naro-webprodeast1744_z2](#))

1 associated public IPs

Public IP address (IPv6) : -

Private IP address : [REDACTED]

Private IP address (IPv6) : -

Virtual network/subnet : [Vnet-soc-lab1/default](#)

DNS name : [Configure](#)

Going back to the resource group, the virtual machine and related resources have populated within our resource group.

Now, to open up our target virtual machine to the public internet, we will access the <VIRTUALMACHINE-NAME>-nsg to edit the firewall rules.

I deleted the default open 3389 (Remote Desktop Protocol) port, and created the worst firewall rule known to man, opening traffic for every port on our virtual machine:



DANGER-AllowAnyCustomAnyInbound

NARO-WEBPRODEAST1-nsg



Source ⓘ

Any

Source port ranges * ⓘ

*

Destination ⓘ

Any

Service ⓘ

Custom

Destination port ranges * ⓘ

*

Protocol

☒ Any

☐ TCP

☐ UDP

☐ ICMPv4

☐ ICMPv6

Action

☒ Allow

☐ Deny

Priority * ⓘ

100

Name

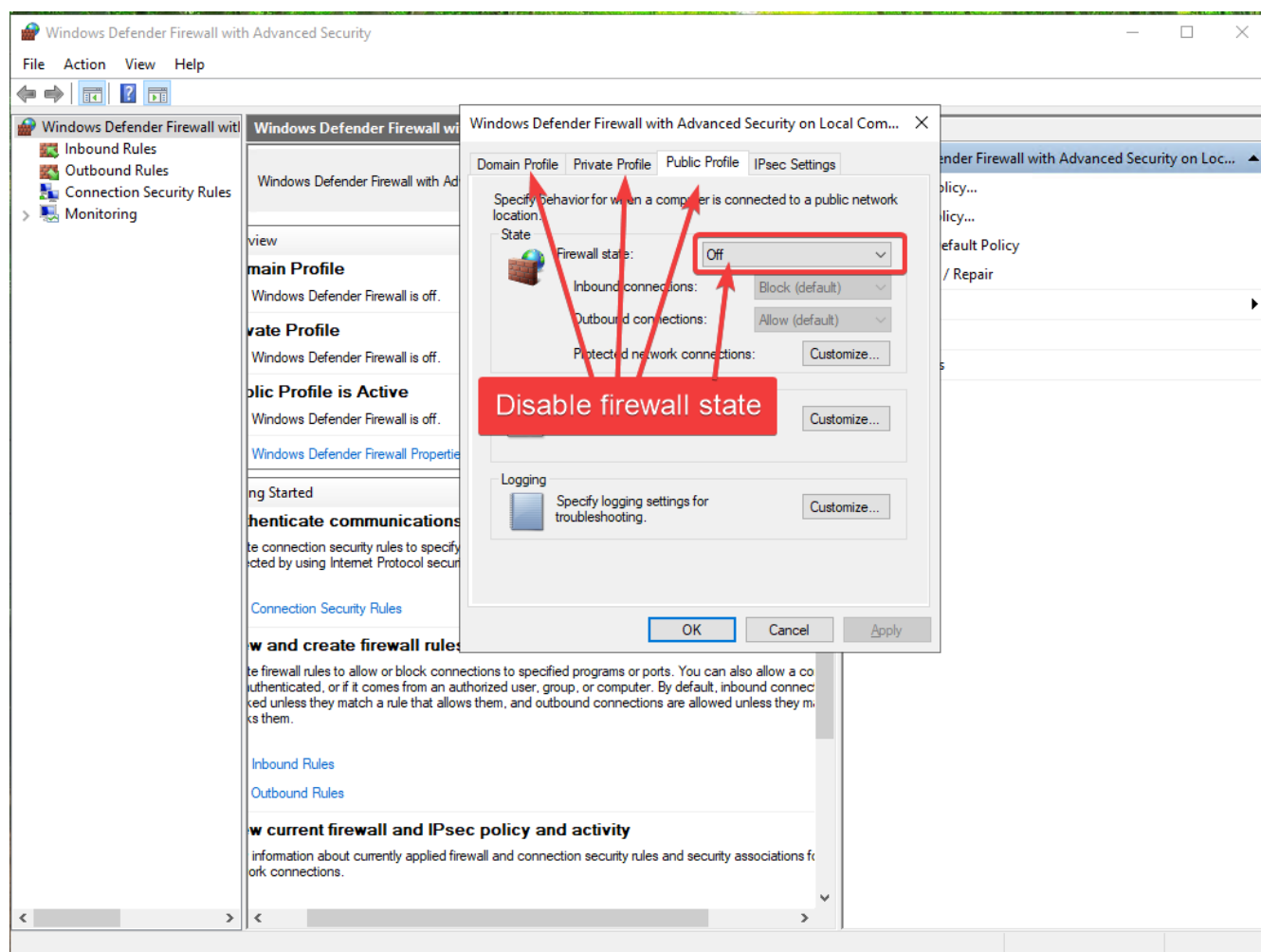
DANGER-AllowAnyCustomAnyInbound

Now that our network security group firewall is allowing traffic on any port, we also need to add a similar rule in the Windows Defender Firewall.

Using Remote Desktop Protocol, I connected to my Windows 10 Virtual Machine through its public IP address assigned on the overview page.

In order to generate the most amount of logs for the purpose of this lab, we will disable our last line of defense, the Windows Defender Firewall.

To access the Windows Defender Firewall, in the start menu, type `wf.msc`. Then, disable on Domain Profile, Private Profile, and Public Profile.



To confirm our virtual machine is reachable to the public internet, I pinged the virtual machines public IP address to confirm it is reachable.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

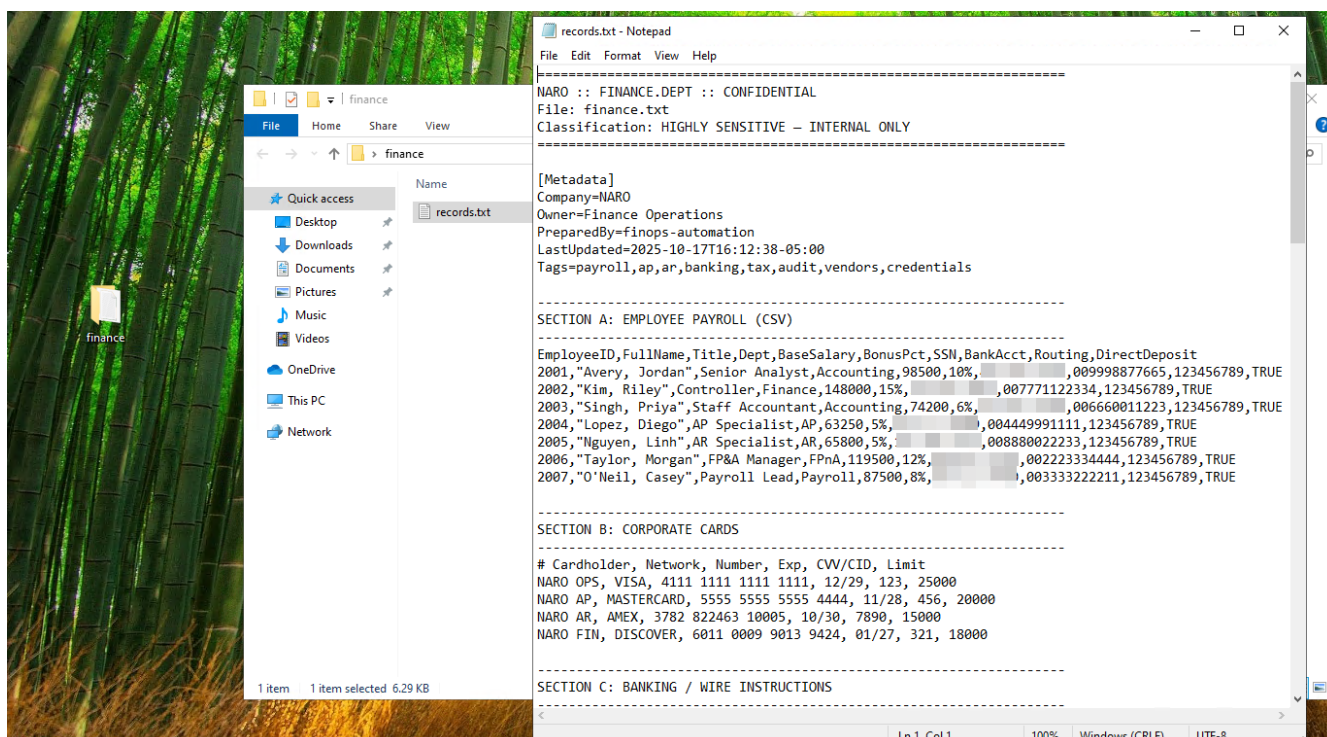
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hunts3c> ping 172.200.181.71

Pinging 172.200.181.71 with 32 bytes of data:
Reply from 172.200.181.71: bytes=32 time=96ms TTL=109
Reply from 172.200.181.71: bytes=32 time=66ms TTL=109
Reply from 172.200.181.71: bytes=32 time=45ms TTL=109
Reply from 172.200.181.71: bytes=32 time=51ms TTL=109

Ping statistics for 172.200.181.71:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 45ms, Maximum = 96ms, Average = 64ms
PS C:\Users\hunts3c>
```

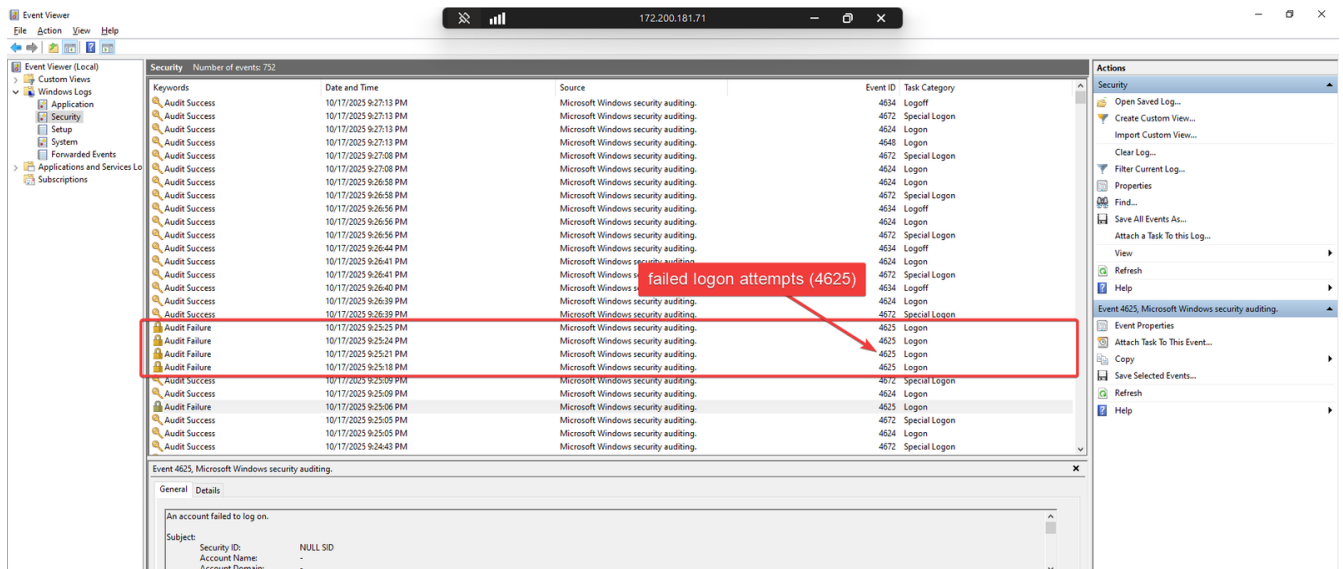
I also felt like adding an enticing honeypot file named `records.txt` filled with dummy data inside a super real finance folder.



Generating and capturing telemetry

Time to start messing around with some logs, to generate some logs I opened up Remote Desktop Connection on my local computer, and failed the logon a few times with a user that doesn't exist in the environment.

Signing back in as our labuser account, and opening the Event Viewer, we can capture our failed logon attempts.



Sentinel Integration & Hunting

Logging has been confirmed, however for the purpose of setting up a SIEM, in Azure I created a Log Analytics workspace, and then added Microsoft Sentinel to that Log Analytics workspace.

After setting up logging and Sentinel Extensions to our Log Analytics workspace, we can begin querying for logs using KQL.

Running a query for Security Alerts, no logs were generated. 😞 So, to let the logs generate for a bit, I took a break and let the virtual machine open.

Observations

After about an hour, by querying for successful logons, unsuccessful logons, process creation, and service

installations I began to notice some password spraying targeting user & administrator accounts from an IP of 102.88.21.216 in Nigeria.

Run Time range : Last 24 hours Show : 1000 results

```
1 SecurityEvent
2 | where EventID == 4624 or EventID == 4625 or EventID == 7045 or EventID == 4688
3 | project TimeGenerated, Account, Computer, IpAddress, Activity, EventID
```

Geolocation data from IP2Location Product DB6, 2025 10 1

IP ADDRESS: 102.88.21.216
COUNTRY: Nigeria
REGION: Lagos
CITY: Lagos

ISP: MTN Nigeria Communication Limited
ORGANIZATION: Not available
LATITUDE: 6.4390
LONGITUDE: 3.3968

Incorrect location? Contact IP2 location view map

Results Chart

TimeGenerated [UTC] ↑↓	Account	Computer	IpAddress	Activity	EventID
> 10/17/2025, 11:06:13.578 PM	\ADMIN	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625
> 10/17/2025, 11:04:45.799 PM	\ADMINISTRATOR	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625
> 10/17/2025, 11:03:16.795 PM	\USER	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625
> 10/17/2025, 11:01:50.446 PM	\ADMIN	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625
> 10/17/2025, 11:00:19.803 PM	\ADMINISTRATOR	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625
> 10/17/2025, 10:58:52.236 PM	\USER	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625
> 10/17/2025, 10:58:01.241 PM	NT AUTHORITY\SYSTEM	NARO-WEBPRODEAS	-	4624 - An account was successf...	4624
> 10/17/2025, 10:58:00.331 PM	NT AUTHORITY\SYSTEM	NARO-WEBPRODEAS	-	4624 - An account was successf...	4624
> 10/17/2025, 10:57:22.530 PM	\ADMIN	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625
> 10/17/2025, 10:56:57.688 PM	NT AUTHORITY\SYSTEM	NARO-WEBPRODEAS	-	4624 - An account was successf...	4624
> 10/17/2025, 10:56:14.250 PM	NT AUTHORITY\SYSTEM	NARO-WEBPRODEAS	-	4624 - An account was successf...	4624
> 10/17/2025, 10:55:49.917 PM	\ADMINISTRATOR	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625
> 10/17/2025, 10:54:15.985 PM	\USER	NARO-WEBPRODEAS	102.88.21.216	4625 - An account failed to log ...	4625

To further identify where the malicious traffic is coming from, I added a watchlist in Azure containing a .csv file with data on IP address and specific location information such as latitude/longitude, and city/country name.

Using a KQL query with our watchlist, searching for the attackers IP address and failed logon events, we can identify the latitude/longitude and city name the malicious trafficking is sourcing from.

Failed/Succes... Failed Successful Localational Save Share Queries hub

Time range: Last 24 hours Show: 1000 results KQL mode

```

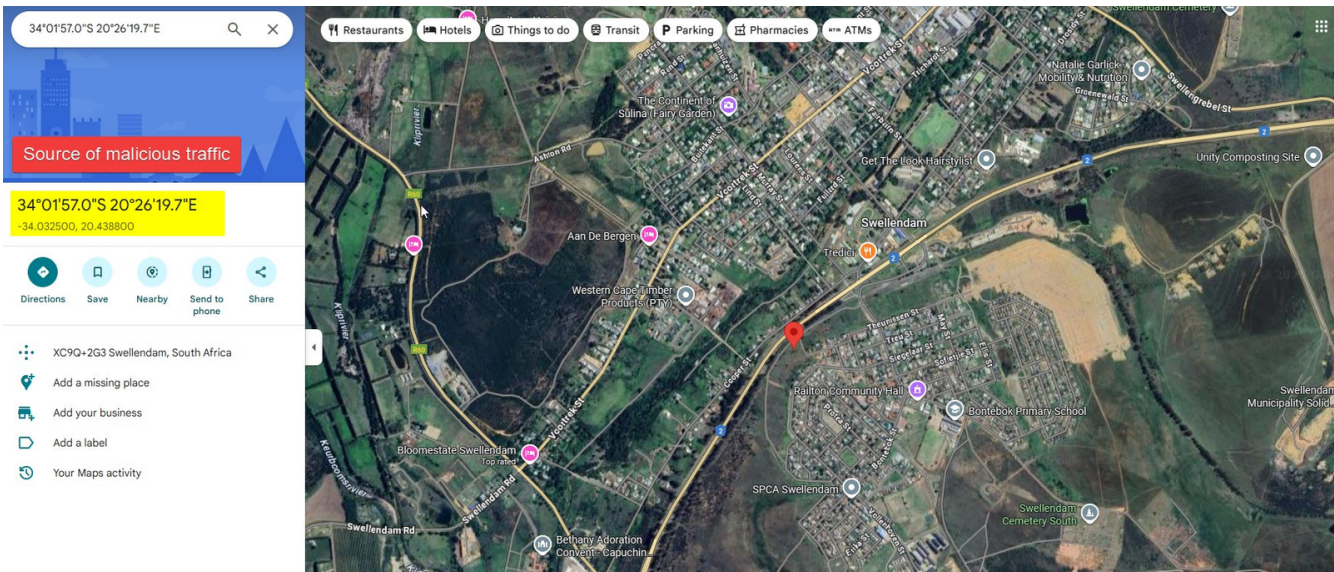
1 let GeoIPDB_FULL = _GetMatchlist("geolp");
2 let WindowsEvents = SecurityEvent
3   | where Ipaddress == "102.88.21.216"
4   | where EventID == 4625
5   | order by TimeGenerated desc
6   | evaluate ipv4_lookup(GeoIPDB_FULL, IpAddress, network);
7 WindowsEvents
8

```

	LastUpdatedTimeUTC [UTC]	SearchKey	cityname	countryname	latitude	longitude	network
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16
I-b10d-4074-834e-aa...	10/17/2025, 11:24:11.326 PM	102.88.0.0/16	Swellendam	South Africa	-34.0325	20.4388	102.88.0.0/16

0s 881ms Display time (UTC+00:00) Query details 1 - 13 of 33

Out of curiosity, I cross-referenced the IP and coordinates to Google Maps to better visualize the origin of the traffic:



I did some research into creating an attack map in Azure, and was able to get that successfully running:

Windows VM Attack Map

law-soclab



How do enterprises leverage Workbooks?

Query recent Azure Monitor workbooks changes

Help me create my first Azure Workbook

Edit Open Auto refresh: Off



Swellendam (South Africa)

42

Since the lab environment has only been deployed for ~2 hours, and malicious activity is currently only coming from a single location, I will leave the virtual machine up for an additional 24 hours and observe attacker behaviours and further validate analytic rules and visualizations.

[Home](#) >

Windows VM Attack Map

law-soclab

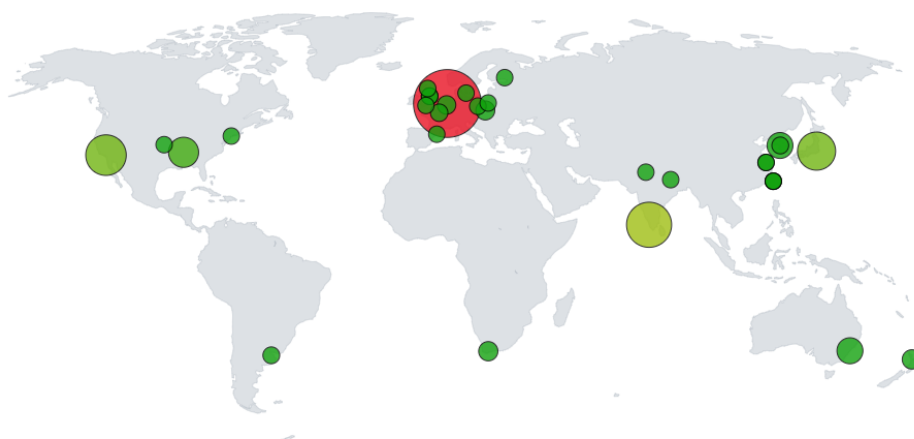


How do enterprises leverage Workbooks?

Help me create my first Azure Workbook

Query recent Azure Monitor workbooks changes

Edit Open Auto refresh: Off



Maarn (Netherlands)

38.5 κ

Kochi (India)

16.2 κ

Glendora (United States)

12.7 κ

Kitaterao (Japan)

11 κ

Memphis (United States)

6.09 κ

Sydney (Australia)

3.92 κ

Gwanak-gu (South Korea)

3.88 κ

Other

2.63 κ

Auckland (New Zealand)

1.02 κ

Swellendam (South Africa)

1 κ

Within approximately 12 hours, the virtual machine has recorded over 90,000 failed authentication attempts, indicating a large scale brute force or credential stuffing attack targeting the system.

1/3 of the traffic coming from Maam, Netherlands.

Learning Points

This lab provided valuable insight into real world attack logs, SIEM configuration, and threat-hunting methodologies within a controller Azure environment. Some key takeaways:

1. Exposure driving visibility

- By intentionally exposing a poorly secured VM to the public internet with nearly no defenses other than a username and password, it became very clear very quickly how quickly automated scanners and brute forcers identify and target open systems. The speed and scale of the activity was very shocking with over 90,000 failed logon attempts in under 12 hours, 1,000 in the first hour.

2. The importance of context

- Integrating external IP watchlists and geolocation data took these raw logs into key intelligence mapping attacker IPs to physical regions across the world really helped visualize the global distribution of attacks.

3. Log Centralization

- It was very nice having one centralized SIEM across the board, having all my logs flow into Microsoft Sentinel shows the importance of SIEMs and log centralization instead of combing across different tools to obtain key information.

4. KQL

- Using Kusto Query Language (KQL), allowed me to query important information, quickly. Using specific queries targeting specific event IDs, and user accounts, alongside IP geolocation data streamlined the process of identifying these brute force attacks.

Next Steps

1. Deploy additional virtual machines under a domain to simulate a small enterprise network and expand log sources to include a variety of systems including Linux systems, database systems, and Azure activity logs.
2. Integrating threat intelligence feeds into Microsoft Sentinel to gain a deeper understanding of detections.
3. Start writing custom detection rules and playbooks for automated response.



hunts.dev

Originally published on hunts.dev

<https://hunts.dev/blog/soclab/>